



Intelligent Security API

开发指南

有效期：2024-08-15

在法律允许的最大范围内，本文档是“按照现状”提供，可能存在瑕疵或错误。本公司不对本文档提供任何形式的明示或默示保证，包括但不限于适销性、质量满意度、适合特定目的、不侵犯第三方权利等保证；亦不对使用或是分发本文档导致的任何特殊、附带、偶然或间接的损害进行赔偿，包括但不限于商业利润损失、系统故障、数据或文档丢失产生的损失。

有效期：2024-08-15

1. 概览

1.1 介绍

ISAPI全称为Intelligent Security API(智能安全API)，是基于[HTTP\(Hypertext Transfer Protocol\)](#)的应用层协议，并采用[REST架构\(Representational State Transfer, 表述性状态转移\)](#)，实现安防设备（如摄像机、数字录像机、网络录像机等）与平台或客户端软件之间的通信。

自2013年创建以来，ISAPI已经有11000多个接口，内容包括设备管理、车辆识别、停车场管理、人脸智能、门禁权限管理审讯管控、录像管控等功能，应用于公安、司法、交通、消防、安检、教育等各种行业。

1.2 适用产品

- 门禁设备

- 明眸

DS-K1F600U-D6E、DS-K1F900、DS-K1T240、DS-K1T240-3XF、DS-K1T240-QR、DS-K1T240MG-3XF、DS-K1T6-GS1、DS-K1T6-GS2(D)、DS-K1T6-GS3、DS-K1T601、DS-K1T610、DS-K1T6240、DS-K1T6240MG-3XF、DS-K1T640A、DS-K1T640系列、DS-K1T641、DS-K1T641A、DS-K1T642、DS-K1T643-3XF、DS-K1T671、DS-K1T671A、DS-K1T671B、DS-K1T671BM-3XFTD、DS-K1T671M-3XF、DS-K1T671MI、DS-K1T671TM-3XF、DS-K1T671TM-3XF/TB、DS-K1T671TMI、DS-K1T672、DS-K1T673、DS-K1T673M-3XF、DS-K1T680、DS-K1T690、DS-K1T691、DS-K1T950M、DS-K1T950MF、DS-K1T981系列、DS-K3080、DS-K5012-R(国内标配)单机版、DS-K5013、DS-K5014、DS-K5014B、DS-K5014C、DS-K5022、DS-K5022A、DS-K5032-3XFD、DS-K5032系列、DS-K5603A、DS-K5604A、DS-K5604A-3XF/VI、DS-K5604A-3XF/ZV、DS-K5604A-3XFVIDQ、DS-K5604A-3XFVIGQ、DS-K5604A-VI、DS-K5604A-VI/C、DS-K5604A-Z、DS-K5604B、DS-K5604B-VI、DS-K5604B-Z、DS-K5607、DS-K5671、DS-K5671-3XF/ZU、DS-K5671-W、DS-K5671B、DS-K5680系列、DS-K56A0-3XFD、DS-K56A0系列、DS-KAB5014-RC、DS-KAB6-ST、HIK-FaceACE-P4

1.3 术语和定义

1.3.1 中心 center

一组独立的服务或服务群，包括至少一个设备接入服务，以及一个或多个应用平台。

1.3.2 多重认证 multi-factor authentication

一种人员群组内的成员按照配置的规则刷卡/指纹/人脸/虹膜等认证才能开门的认证方式。

1.3.3 通道设备升级

升级接入到边缘域设备下的边缘节点设备，需要边缘域设备把边缘节点设备的固件升级包下载到边缘域本地（基于HTTP(s)下载），然后升级边缘节点设备。如果是通过ISUP协议接入，那么设备必须支持FTP升级，如果是OTAP协议接入，设备必须支持HTTP(s)升级。

1.3.4 固件识别码

用于设备升级过程中，唯一匹配升级包和设备。只有当升级包的固件识别码和设备固件识别码相同时，才能使用该升级包对设备进行升级。

1.3.5 边缘节点设备

侧重于多维感知数据采集和前端智能处理的设备，一般指代相机等。

1.3.6 监听 listen

平台本地对自身的网络IP和端口开启监听服务，设备产生报警事件后与服务端口建立连接，将消息发送发送给平台监听端口，完成后关闭连接。

1.3.7 设备号码

泛指楼宇号码规则，是为了区分各单元设备，由期号，幢号，单元号，编号组合而成的11位号码。主要用于呼叫对讲过程中设备的唯一标识。

1.3.80 人员类型 user type

人员根据指定区域所具有属性的不同，可分为普通人、访客、黑名单人。在门禁系统中：普通人具有指定区域的长期访问权限；访客具有指定区域的临时访问权限；黑名单人没有指定区域的访问权限。

1.3.9 以人为中心 person-based access control

人员ID作为人员管理的唯一标识，卡片、指纹、人脸作为人员凭证，通过人员ID进行关联，权限也通过人员ID进行关联。

1.3.10 凭证类型 verify type

用于识别指定人员具体是谁的数据信息。如卡片、指纹、人脸都可以作为凭证与指定人员进行关联，当设备检测到相关凭证信息时，即可以通过相关比对算法识别出该凭证关联的人员信息。

1.3.11 子设备

详见：[子设备集成流程](#)

1.3.12 访问权限 access permission

用户设定什么人在什么时候可以对哪些门有开门权限。权限内容包括人员信息、时间信息和门信息。

1.3.13 群组 group

包含一组人员的集合（同一人员可同时所属于不同群组，目前最多同属于4个群组），用于多重认证场景。

1.3.140 萤石云 EZVIZ Cloud Storage

萤石提供的云存储服务。

1.3.15 推模式 pull mode

设备主动将相关信息推送给中心。

1.3.16 监听主机

监听主机为开启监听服务的平台，可以接收设备上报的事件信息。

1.3.17 布防 fortify

客户端主动向设备建立事件上传连接，设备产生事件消息时通过此连接发送给客户端。布防有订阅和非订阅两种：订阅为客户端指定事件列表，设备只发送列表中的事件消息；非订阅向设备为客户端发送设备所有类型的事件消息。

1.3.18 设备升级

升级边缘域设备和边缘节点设备。

1.3.190 新设备号码

为了应对后续扩展各种其他行业设备号码规则差异化，新设备号码标识由行业，设备类型，版本，原设备号码4种因子组成的16位号码。具体组装格式为：行业#设备类型#版本#号码，设备之间信令通信过程中通过传递新设备号码标识来解析设备。

1.3.20 边缘域设备

侧重原始感知数据存储、智能处理、关联分析及快速响应的设备，一般指代NVR等。

1.3.21 设备

包含边缘域设备和边缘节点设备。

1.3.22 拉模式 push mode

中心主动从设备拉取相关信息。

1.3.23 事件 event

设备主动上报的信息。这类信息需要设备实时上报，并被客户端软件或平台（简称平台）及时处理。如果设备网络中断，可以缓存下来，待恢复后再上报。

1.4 符号和缩略语

无

EZVIZ: 萤石（Easy View/Vision）

HiCore: 海康设备内部的应用主程序（hikvision Core）

ISUP: 智能安全向上传输协议（Intelligent Security Uplink Protocol）。

AMS: 报警管理服务器（Alarm Manage Server）。

CMS: 中心管理服务器/设备接入服务器（Central Mangement Server）。

admin: 管理员（administrators）

HTTP: 超文本传输协议（Hypertext Transfer Protocol）

ACS: 门禁控制系统（Access Control System）。是对出入口通道进行管制的系统。主要组成部分包括门禁读卡器、门禁主机、电控锁、出门按钮、门卡、应用软件等。

RTSP: 实时流传输协议（Real Time Streaming Protocol）

ISAPI: 基于HTTP RESTfull的海康设备接入协议（Intelligent Security API）。

APK: Android应用程序包(Android application package)

SDP: 会话描述协议（Session Description Protocol）

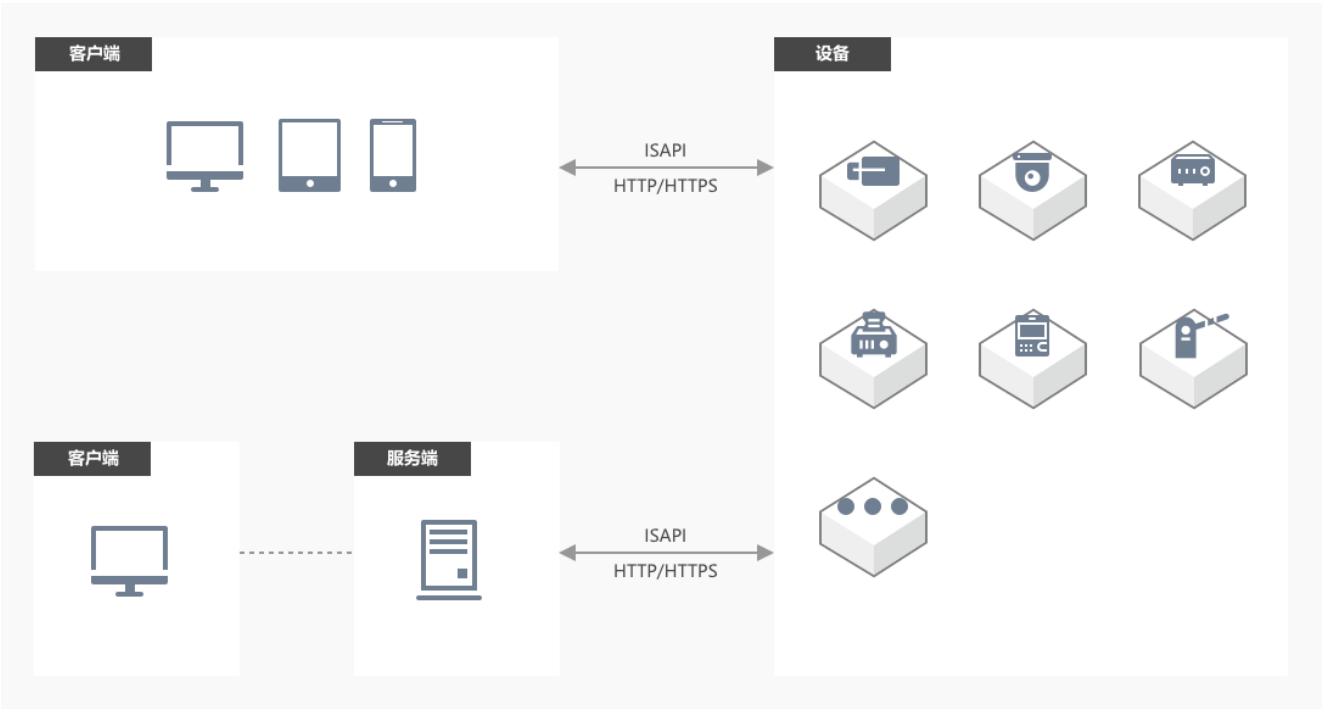
HEOP: 海康嵌入式开放平台（hikvision Embedded open platform）

1.5 更新说明

- 新增人脸库管理功能领域
- 新增日程管理功能领域
- 优化获取日程参数配置能力 URI:/ISAPI/Publish/ScheduleMgr/playSchedule/<scheduleID>/capabilities
- 优化获取素材管理参数能力 URI:/ISAPI/Publish/material/<materialID>/capabilities

2 应用场景

用户通过ISAPI对接设备时，设备做为服务端监听固定端口，用户程序做为客户端主动登录设备进行通信。这要求设备具有固定的IP地址，并且保证客户端的请求能够到达服务端。



3 所属网络模型中的层级

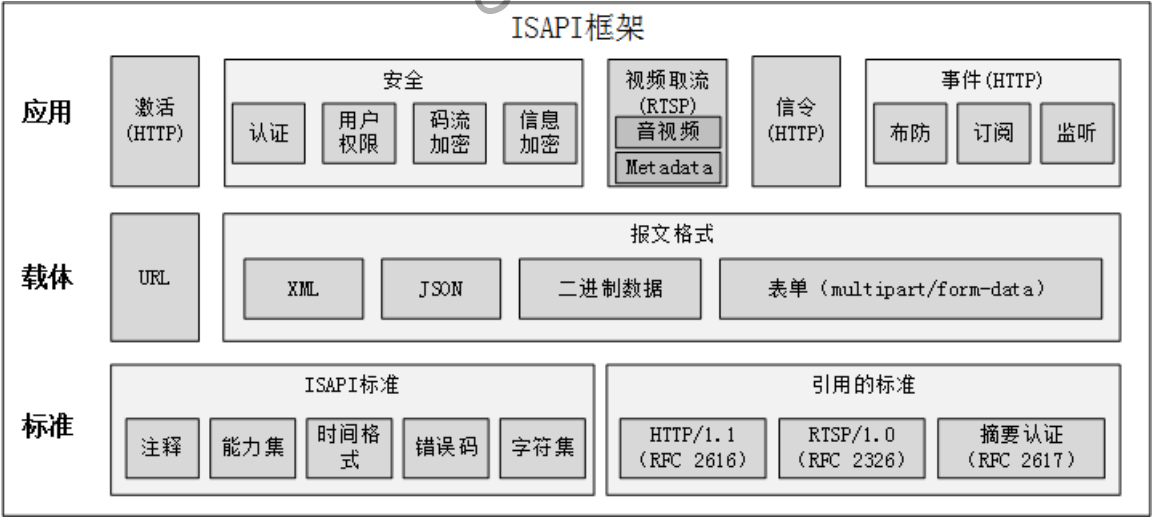
ISAPI是基于HTTP的应用层协议，所以ISAPI继承了HTTP的所有规范和特性。

经常和ISAPI一起使用的协议还有基于多播/组播的SADP（Search Active Device Protocol），用于实现设备发现和激活；基于TCP/UDP的RTSP（Real Time Streaming Protocol），用于实现设备实时预览和录像回放等。

	设备发现	信令交互	媒体流
应用层	SADP	ISAPI (HTTP)	RTSP/RTP
传输层	MCAST	TCP	TCP/UDP
网络层	IP		
网络接口	硬件驱动接口		

4 ISAPI框架

4.1 总览



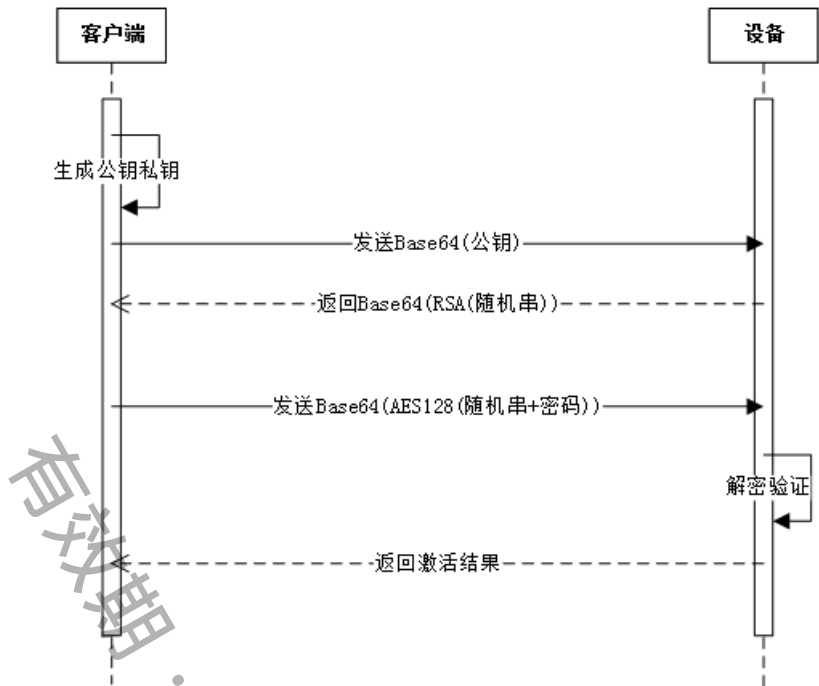
说明：一般来讲，ISAPI指的是基于HTTP标准的通信协议，但是ISAPI通常都是和RTSP一起使用的，所以我们将RTSP标准也纳入了ISAPI体系当中。

并且在RTSP标准上扩展了传输码流附加信息的Metadata方案，用于同步传输视频码流和码流的智能结构化信息。Metadata方案和RTSP标准是兼容的。

4.2 激活

激活的目的是保证客户对设备进行初始化密码设置，且密码串符合安全规则。激活设备后才能使用相关的功能。ISAPI是应用层的通信协议，使用ISAPI激活设备时，必须要知道设备的IP地址，并且保证激活客户端和设备网络的连通性。

设备自带的Web程序支持ISAPI激活方式激活设备。在客户机浏览器地址栏输入设备IP地址，按照激活引导完成激活。集成客户想要用自己的程序激活设备，需要集成ISAPI激活功能，相关流程和API如下。



首先定义两种运算：

- bytesToHexString: 长度为N的字节数组转换为长度为2N的十六进制格式字符串，例如 127,10,23 -> 7f0a17
 - hexStringToBytes: 长度为2N的十六进制格式字符串转换为长度为N的字节数组，例如 7f0a17 -> 127,10,23
1. 客户端生成1024bits的公私钥对，取出公钥中的模数modulus（长度为128字节，若取出长度大于128需去掉最前面的0），后续我们称之为公钥模数。
 2. 公钥模数做bytesToHexString转换得到256字节的公钥字符串，将公钥字符串base64编码后组装到XML消息中发给设备，接口为POST /ISAPI/Security/challenge。
 3. 设备解析请求，base64解码得到256字节的公钥字符串，然后对其做hexStringToBytes转换，得到128字节的公钥模数，使用公钥模数和公钥指数(exponent，固定值为'010001')构造得到完整的公钥。
 4. 设备生成32字节的十六进制随机字符串，调用RSA接口使用公钥对随机字符串进行加密，得到加密数据，加密数据做bytesToHexString转换，再base64编码，回复给客户端。
 5. 客户端base64解码设备响应的字符串后，做hexStringToBytes转换得到加密数据，对加密数据用私钥进行RSA解密后，得到32字节的十六进制随机串，然后做hexStringToBytes转换得到16字节的AES密钥，对32字节随机串前16字节+真实密码进行AES128-ECB（填充方式zeropadding）加密处理，得到的密文做bytesToHexString转换，再base64编码，组装到XML消息中发给设备，接口为PUT /ISAPI/System/activate。注：例如随机串的前16个字符串是aaaabbbbccccdddd，真实密码是Abc12345，那么加密前的数据是aaaabbbbccccddddAbc12345。这么可以保证客户端执行第二步时，是按第一步给的随机串作为秘钥加密的。
 6. 设备base64解码后，做hexStringToBytes转换得到密文，再用AES密钥做AES128-ECB解密后去除前16字节得到真实密码。
 7. 校验密码合法性，设备返回激活结果。

说明：

- 可以通过接口 `GET /SDK/activateStatus` 获取设备激活状态，此接口不需要认证鉴权。
- 海康设备还支持SADP(Search Active Device Protocol，设备发现协议)方式激活，SADP使用链路层通信协议，使用SADP时，无需知道设备的IP地址，但是需要保证设备和运行SADP程序连接在同一个路由器上。SADP还支持设备发现（局域网内）、修改设备密码等功能。我们提供SADP的集成开发包HCSadpSDK，集成包有开发指南、插件以及示例Demo，示例Demo也可以作为简单的SADP工具使用。

4.3 安全机制

4.3.1 认证

客户端程序向设备发送请求时，需要使用摘要认证（详见[RFC 7616](#)）完成身份认证。

目前主流的HTTP请求类库都封装支持了摘要认证。客户程序只需要简单地调用类库接口即可完成摘要认证过程，示例源码如下。

4.3.1.1 C/C++ (libcurl)

```
// #include <curl/curl.h>
// 回调函数
static size_t OnWriteData(void* buffer, size_t size, size_t nmemb, void* lpVoid)
{
    std::string* str = dynamic_cast<std::string*>((std::string *)lpVoid);
    if( NULL == str || NULL == buffer )
    {
        return -1;
    }

    char* pData = (char*)buffer;
    str->append(pData, size * nmemb);
    return nmemb;
}

std::string strUrl = "http://192.168.18.84:80/ISAPI/System/deviceInfo";
std::string strResponseData;
CURL *pCurlHandle = curl_easy_init();
curl_easy_setopt(pCurlHandle, CURLOPT_CUSTOMREQUEST, "GET");
curl_easy_setopt(pCurlHandle, CURLOPT_URL, strUrl.c_str());
// 设置用户名和密码
curl_easy_setopt(pCurlHandle, CURLOPT_USERPWD, "admin:admin12345");
// 设置认证方式为摘要认证
curl_easy_setopt(pCurlHandle, CURLOPT_HTTPAUTH, CURLAUTH_DIGEST);
// 设置回调函数
curl_easy_setopt(pCurlHandle, CURLOPT_WRITEFUNCTION, OnWriteData);
// 设置回调函数的参数, 获取反馈信息
curl_easy_setopt(pCurlHandle, CURLOPT_WRITEDATA, &strResponseData);
// 接收数据时超时设置, 如果5秒内数据未接收完, 直接退出
curl_easy_setopt(pCurlHandle, CURLOPT_TIMEOUT, 5);
// 设置重定向次数, 防止重定向次数太多
curl_easy_setopt(pCurlHandle, CURLOPT_MAXREDIRS, 1);
// 连接超时, 这个数值如果设置太短可能导致数据请求不到就断开了
curl_easy_setopt(pCurlHandle, CURLOPT_CONNECTTIMEOUT, 5);
CURLcode nRet = curl_easy_perform(pCurlHandle);
if (0 == nRet)
{
    // 输出接收的消息
    std::cout << strResponseData << std::endl;
}
curl_easy_cleanup(pCurlHandle);
```

4.3.1.2 C# (WebClient)

```
// using System.Net;
// using System.Net.Security;
try
{
    string strUrl = "http://192.168.18.84:80/ISAPI/System/deviceInfo";
    WebClient client = new WebClient();
    // 设置用户名和密码
    client.Credentials = new NetworkCredential("admin", "admin12345");
    byte[] responseData = client.DownloadData(strUrl);
    string strResponseData = Encoding.UTF8.GetString(responseData);
    // 输出接收的消息
    Console.WriteLine(strResponseData);
}
catch (Exception ex)
{
    Console.WriteLine(ex.Message);
}
```

4.3.1.3 Java (HttpClient)

```
// import org.apache.commons.httpclient.HttpClient;
String url = "http://192.168.18.84:80/ISAPI/System/deviceInfo";
HttpClient client = new HttpClient();
// 设置用户名和密码
UsernamePasswordCredentials creds = new UsernamePasswordCredentials("admin", "admin12345");
client.getState().setCredentials(AuthScope.ANY, creds);
GetMethod method = new GetMethod(url);
method.setDoAuthentication(true);
int statusCode = client.executeMethod(method);
byte[] responseData = method.getResponseBodyAsString().getBytes(method.getResponseCharSet());
String strResponseData = new String(responseData, "utf-8");
method.releaseConnection();
// 输出接收的消息
System.out.println(strResponseData);
```

4.3.1.4 Python (requests)

```
# -*- coding: utf-8 -*-

import requests
request_url = 'http://192.168.18.84:80/ISAPI/System/deviceInfo'
# 设置认证信息
auth = requests.auth.HTTPDigestAuth('admin', 'admin12345')
# 发送请求, 获取响应
response = requests.get(request_url, auth=auth)
# 输出响应内容
print(response.text)
```

4.3.2 用户权限

通过采用三类拥有不同权限用户控制和管理访问。

管理员：该用户类型的账号也称为“admin”，拥有访问所有支持资源的权限。另外该账号需要在任何时间内保持激活状态。

操作员：该用户类型的账号拥有访问通用资源和少量高级资源的权限。

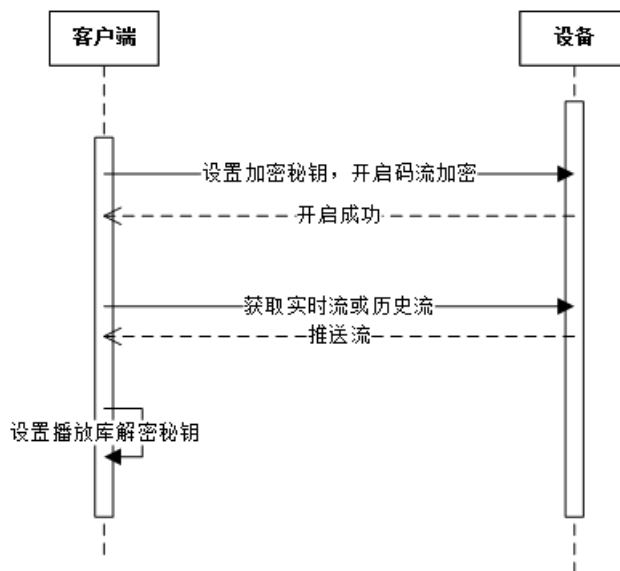
普通用户：该用户类型的账号仅拥有访问通用资源的权限。

4.3.3 码流加密

为了提高来自设备的实时或历史码流的安全性，同时遵循某些国际市场的法律法规，我们提供了基于AES算法的加密功能对码流进行加密。

开启设备码流加密需要先判断设备是否支持码流加密，并且需要设置码流加密密钥、设置播放库解密密钥等操作，交互步骤如下图所示。

说明：开启码流加密之后只能使用海康播放库解码。



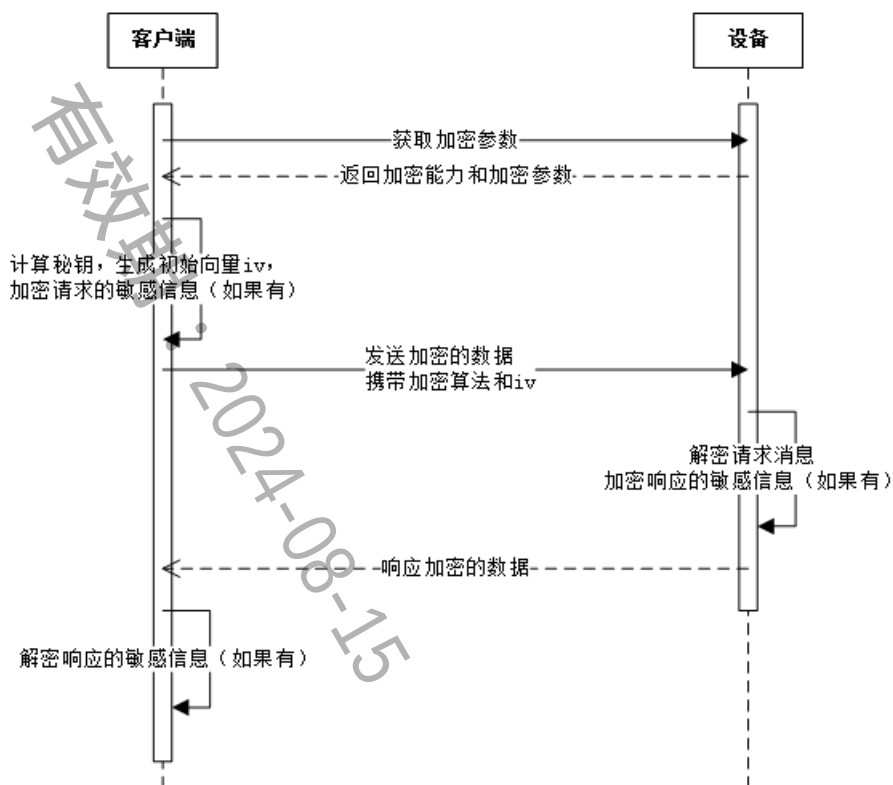
4.3.4 信息加密

在ISAPI集成过程中，部分传输报文中的敏感信息，如用户名、密码等需要进行加密以提高安全性。

有两种方式实现信息加密传输：HTTPS、敏感信息加密（AES加密）。

HTTPS（推荐）：需要先开启设备的HTTPS服务，客户程序通过HTTPS和设备通信，保证信息的安全传输。

敏感信息加密：对报文中的用户名、密码等敏感文本进行加密处理，加密算法的核心算法是AES。交互步骤如下图。



4.4 视频取流

4.4.1 音视频流

ISAPI支持获取和设置设备流媒体相关参数，如视频分辨率、编码格式和码流等。

海康视频设备支持标准的RTSP（Real Time Streaming Protocol, 详见[RFC 7826](#)），客户程序可以使用RTSP向设备取流。

实时取流和录像回放详见功能领域“实时取流”和“回放”。

4.4.2 Metadata

Metadata是海康智能设备产生的智能结构化信息，客户程序在用RTSP向设备取音视频流时设备同步返回。例如描述视频流中的人脸目标框、人脸信息、车辆目标框、车牌号、车辆信息等。客户程序把目标框、人脸信息等内容叠加到视频画面中。

使用Metadata时需要先开启设备的Metadata功能，部分设备支持按类型订阅Metadata，然后使用RTSP向设备取流。详细集成流程见功能领域 **"Metadata管理"**。

4.5 报文解析

4.5.1 报文格式

在使用ISAPI进行通讯交互过程中，请求和响应报文通常采用XML和JSON格式的文本数据，也会有一些如固件包、配置文件二进制格式数据，也存在一次请求中包括多种格式数据的表单格式（multipart/form-data）。

4.5.1.1 XML

对应HTTP请求Headers中的Content-Type通常为application/xml; charset="UTF-8"。

ISAPI请求和响应报文中的XML都为UTF-8编码格式。

在ISAPI中，我们为所有的XML设置默认的命名空间（namespace）<http://www.isapi.org/ver20/XMLSchema>和ISAPI版本号2.0，示例如下。

```
<?xml version="1.0" encoding="UTF-8"?>
<NodeList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <Node>
    <id>1</id>
    <enabled>true</enabled>
    <nodeName>nodeName</nodeName>
    <level>level1</level>
  </Node>
</NodeList>
```

4.5.1.2 JSON

对应HTTP请求Headers中的Content-Type通常为application/json。

ISAPI为了区分XML格式报文和JSON格式报文的接口，在JSON格式报文请求URL中统一增加format=json参数，如<http://192.168.1.1:80/ISAPI/System/Sensor/thermometrySensor?format=json>，没有format=json参数的URL对应的报文格式通常都是XML，但也存在个别例外，请以协议接口定义为准。

ISAPI请求和响应报文中的JSON都为UTF-8编码格式。

4.5.1.3 二进制数据

如固件、配置文件，对应HTTP请求Headers中的Content-Type通常为application/octet-stream。

4.5.1.4 表单（multipart/form-data）

一个ISAPI请求中同时提交多份数据时，如向人脸库中添加人脸记录需要一起提交人员信息和人脸图片，对应HTTP请求Headers中的Content-Type通常为multipart/form-data, boundary=AaB03x，其中boundary是一个变量，用于将整个HTTP Body分割成多个单元，每个单元为一份数据，都有各自的Headers。详细格式说明参考[RFC 1867 \(Form-based File Upload in HTML\)](#)，示例如下，请注意boundary前面和后面的横线--。

```
POST /ISAPI/Intelligent/FDLib/pictureUpload
Content-Type: multipart/form-data; boundary=AaB03x
Content-Length: 56789

--AaB03x
Content-Disposition: form-data; name="PictureUploadData";
Content-Type: text/xml
Content-Length: 1234

<PictureUploadData/>
--AaB03x
Content-Disposition: form-data; name="face_picture"; filename="face_picture.jpg";
Content-Type: image/jpeg
Content-Length: 34567

图片数据
--AaB03x--
```

说明： RFC规范中强烈建议携带一个整体的Headers字段Content-Length，但这不是必须的。而且没有要求每个表单单元Headers中是否应携带Content-Length字段。客户程序和设备程序在解析表格式数据时，都应考虑Headers中没有Content-Length字段的情况。

4.5.2 注释说明

ISAPI请求报文和响应报文的字段说明都以注释形式写在请求示例和响应示例中，如下：

```
<?xml version="1.0" encoding="UTF-8"?>

<NodeList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 节点列表, attr:version{req, string, 协议版本, range:[,]}-->
  <Node>
    <!--ro, opt, object, 节点信息-->
    <id>
      <!--ro, req, int, 节点序号, range:[,], step:, unit:, unitType:-->1
    </id>
    <enabled>
      <!--ro, opt, bool, 使能标志-->true
    </enabled>
    <nodeName>
      <!--ro, req, string, 节点名称, range:[1,32]-->test
    </nodeName>
    <level>
      <!--ro, opt, enum, 级别, subType:string,
      [Level1#等级1,Level2#等级2,Level3#等级3]-->level1
    </level>
  </Node>
</NodeList>
```

```
{
  "name": "test",
  /*ro, req, string, 名称, range:[1,32]*/
  "type": "type1",
  /*ro, req, enum, 类型, subType:string, [type1#类型1,type2#类型2]*/
  "enabled": true,
  /*ro, opt, bool, 使能, desc:xxxxxx*/
  "NodeList": {
    /*opt, object, 节点列表, depend,{$.enabled,eq,true}*/
    "scene": 1,
    /*req, enum, 场景, subType:int, [1#场景1,2#场景2,3#场景3]*/
    "ID": 1
    /*req, int, 编号, range:[1,8], step:, unit:, unitType:*/
  }
}
```

其中关键注释的含义如下表。

符号	含义	说明
ro	读写属性：只读	此字段只支持获取，不支持修改
wo	读写属性：只写	此字段只支持修改，不支持获取
req	选择属性：必选	发送给设备的请求消息中必须有此字段，设备响应消息中必须有此字段
opt	选择属性：可选	发送给设备的请求消息中可选此字段，设备响应消息中可选此字段
dep	选择属性：依赖	当满足特定条件时，此字段有效且必选
object	字段类型：对象	object字段有若干个子字段
list	字段类型：列表	后面的subType表示list的子项类型
subType	字段类型：字符串	后面的range表示字段的最大长度和最小长度
int	字段类型：整数	后面的range表示字段的最大值和最小值
float	字段类型：浮点数	后面的range表示字段的最大值和最小值
bool	字段类型：布尔	取值有true和false
enum	字段类型：枚举	后面的subType表示枚举项是string类型还是int类型，subType后面[]中的内容为枚举选项。
subType	字段子类型	字段类型为list或enum时，表示每一个子对象的类型
desc	字段描述	字段的详细描述

4.5.3 能力集

ISAPI为几乎每个功能、每个接口以及每个字段都设计了能力集。能力集的URL都是以/capabilities结尾（有些接口会带URL参数，格式为/capabilities?format=json&type=xxx）。

ISAPI能力集主要有两种：设备是否支持某功能、接口字段的取值范围。

设备是否支持某功能：一般写为isSupportxxxxxxx的格式，表示是否支持某个功能以及功能相关的一套接口。

```
{
  "isSupportMap": true,
  /*ro, opt, bool, 是否支持电子地图功能, desc:/ISAPI/SDT/Management/map/capabilities?format=json*/
  "isSupportAlgTrainResourceInfo": true,
  /*ro, opt, bool, 是否只获取算法训练平台资源信息, desc:/ISAPI/SDT/algorithmTraining/ResourceInfo?format=json*/
  "isSupportAlgTrainAuthInfo": true,
  /*ro, opt, bool, 是否只获取算法训练平台授权信息, desc:/ISAPI/SDT/algorithmTraining/SoftLock/AuthInfo?format=json*/
  "isSupportAlgTrainNodeList": true,
  /*ro, opt, bool, 是否只获取算法训练平台节点信息, desc:/ISAPI/SDT/algorithmTraining/NodeList?format=json*/
  "isSupportNAS": true
  /*ro, opt, bool, 是否支持NAS挂载和卸载, desc:/ISAPI/SDT/Management/NAS/capabilities?format=json*/
}
```

接口字段的取值范围：表示接口中每个字段的最大值、最小值、最大长度、最小长度以及可选项等。

```

{
  "boolType": {
    /*req, object, bool类型能力示例*/
    "@opt": [true, false]
    /*req, array, 选项, subType: bool*/
  },
  "integerType": {
    /*req, object, integer类型能力示例*/
    "@min": 0,
    /*ro, req, int, 最小值*/
    "@max": 100
    /*ro, req, int, 最大值*/
  },
  "stringType": {
    /*req, object, string类型能力示例*/
    "@min": 0,
    /*ro, req, int, 最小长度*/
    "@max": 32
    /*ro, req, int, 最大长度*/
  },
  "enumType": {
    /*req, object, enum类型能力示例*/
    "@opt": ["enum1", "enum2", "enum3"]
    /*req, array, 选项, subType: string*/
  }
}

```

说明：同一个能力集，不同型号、版本的设备可能返回不同的结果。协议文档中的示例值仅供参考，请以实际设备返回的结果为准。

4.5.4 时间格式

和W3C标准日期和时间格式一样，ISAPI也采用ISO 8601标准时间格式。

具体格式为：YYYY-MM-DDThh:mm:ss.sTZD

YYYY = 四位数年份
 MM = 两位数月份（01 = 一月，02 = 二月，等）
 DD = 月份中的两位数日期（01 到 31）
 hh = 小时的两位数（00 到 23）（不允许上午/下午）
 mm = 两位数的分钟（00 到 59）
 ss = 两位数的秒（00 到 59）
 s = 一个或多个数字表示一秒的小数部分
 TZD = 时区指示符（Z 或 +hh:mm 或 -hh:mm）

示例：2017-08-16T20:17:06.123+08:00 表示本地时间为2017年08月16日下午20时17分06秒123毫秒，本地时间比UTC时间快8小时，时差加号(+)表示数值比UTC时间更大，减号(-)表示数值比UTC时间更小。

开启夏令时后，本地时间和时差相较于UTC时间都会发生变化，对应的数值都要发生变化，关闭夏令时变化相反。

示例：中国1986年实行夏令时，从5月4日凌晨2时整（北京时间），将时钟拨快一小时，当天减少一小时。9月14日凌晨2时整，再将时钟拨回一小时，当天增加一小时。对应时间变化如下。

- 进入夏令时：1986-05-04T02:00:00+08:00 --> 1986-05-04T03:00:00+09:00
- 结束夏令时：1986-09-14T02:00:00+09:00 --> 1986-09-14T01:00:00+08:00

说明：

- 时间中的时差不能单纯地做为时区的判断依据，当进入夏令时后，时差会发生变化，不代表真实的时区。
- TZ格式（UTC时间，例1986-05-03T18:00:00Z）和TD格式（本地时间与时差，例1986-05-04T02:00:00+08:00）都是符合ISO 8601标准的时间格式。在ISAPI中我们推荐使用TD格式，即用户程序发送的消息和设备发送的消息都用TD格式。
- 少量老版本设备会返回TZ格式的时间，为了表示时差信息且向前兼容，我们为其增加了额外的一个timeDiff字段，示例如下。用户程序在解析设备消息中的时间时需要同时支持TD格式和TZ格式。

```

{
  "dateTime": "1986-05-03T18:00:00Z", /*设备时间，TZ格式为UTC时间，TD格式为设备本地时间与UTC时差*/
  "timeDiff": "+08:00" /*可选，本地时间与UTC时间时差，如无此字段，客户程序把dateTime换算成本地时间使用*/
}

```

4.5.5 字符集

为了避免不常用符号导致设备程序和客户端程序发生异常，ISAPI限定了string类型字段有效字符范围，以下描述的是被允许出现在ISAPI string类型字段中的字符集。

- 单字节：英文小写字母a-z、英文大写字母A-Z、阿拉伯数字0-9、特殊符号（见下表）。
- 多字节：采用Unicode字符集的语言文字，UTF-8编码（UTF-8编码是Unicode字符集的一种传输编码方式，详见RFC 2044）。

序号	特殊字符名称	特殊字符	序号	特殊字符名称	特殊字符
1	开括号	(	18	美元符	\$
2	闭括号	)	19	百分号	%
3	加号	+	20	和号	&
4	逗号	,	21	闭单引号	'
5	减号	-	22	星号	*
6	句号	.	23	斜杠	/
7	分号	;	24	小于	<
8	等号	=	25	大于	>
9	电子邮件符号	@	26	问号	?
10	开方括号	[	27	脱字符	^
11	闭方括号	]	28	开单引号	'
12	下划线	_	29	垂线	
13	开花括号	{	30	波浪号	~
14	闭花括号	}	31	双引号	"
15	空格		32	冒号	:
16	叹号	!	33	反斜杠	\
17	井符	#			

下面列举的是一些特定含义字段的字符有效范围描述。

- 用户名：英文小写字母a-z、英文大写字母A-Z、阿拉伯数字0-9、特殊符号表中编号1-30（包括1和30）的字符。
- 密码：英文小写字母a-z、英文大写字母A-Z、阿拉伯数字0-9、特殊符号表中编号1-33（包括1和33）的字符。
- 用于用户界面显示的名称（如设备名称、人员名称、人脸库名称等）：英文小写字母a-z、英文大写字母A-Z、阿拉伯数字0-9、特殊符号表中编号1-15（包括1和15）的字符、多字节的语言文字。
- 无特殊说明的string类型字段默认支持英文小写字母a-z、英文大写字母A-Z、阿拉伯数字0-9、特殊符号表中编号1-15（包括1和15）的字符、多字节的语言文字。

4.5.6 出错处理

当ISAPI请求失败时（HTTP状态码不等于200），设备会返回HTTP状态码和ISAPI错误码。HTTP状态码详情参见RFC 2616 10 Status Code Definitions，ISAPI错误码详见附件

消息示例如下。

```
HTTP/1.1 403 Forbidden
Content-Type: application/json; charset="UTF-8"
Date: Thu, 15 Jul 2021 20:43:30 GMT
Content-Length: 229
Connection: Keep-Alive

{
  "requestURL": "/ISAPI/Event/triggers/notifications/channels/whiteLightAlarm",
  "statusCode": 4,
  "statusString": "Invalid Operation",
  "subStatusCode": "notSupport",
  "errorCode": 1073741825,
  "errorMsg": "notSupport"
}
```

4.6 事件上报

当在设备上设置的规则被触发时，设备会产生事件消息（例如：移动侦测、人脸抓拍、车辆抓拍等）。事件消息通过主动上报的方式发送给客户端，在ISAPI中有三种接收设备事件消息的方式：布防、订阅、监听。

4.6.1 布防

ISAPI布防通过HTTP长连接实现。客户端向设备发送请求GET /ISAPI/Event/notification/alertStream建立布防连接后，双方保持连接不断开，设备产生事件消息时通过此连接发送给客户端。

说明：

- 设备的应答为表单格式（multipart/form-data），表单每个单元对应一个事件或事件的一部分，这取决于事件消息本身是否由多份数据组成，如人脸抓拍事件同时包括JSON格式的人脸信息和人脸图片二进制数据，这会产生多个表单单元。
- 当设备5秒内没产生事件消息时，设备向客户端发送一个心跳消息，用于确认布防的连接状态。

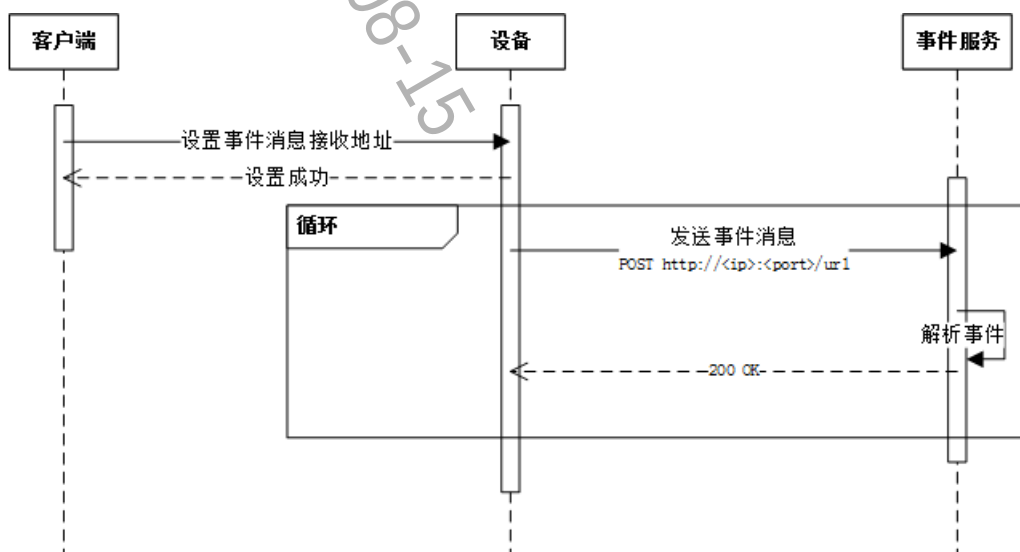
4.6.2 订阅

ISAPI订阅和布防整体上基本一致，区别在于订阅建立连接的接口是POST /ISAPI/Event/notification/subscribeEvent，建立连接时通过请求Body订阅具体的视频通道、事件类型等内容，之后设备则只发送订阅列表中的事件消息给客户端。

集成流程和相关接口见功能领域“事件布防订阅管理”。

4.6.3 监听

监听方式是设备产生事件消息时，主动发送到指定的事件接收地址。需要在设备上配置事件推送地址，并确保地址有效。



说明：

- 客户端和事件服务可以为同一个程序。
- 监听方式设备不产生心跳消息。

设置事件消息接收地址接口见功能领域“HTTP报警主机管理”。

5 用户管理

5.1 功能介绍

设备激活后，可通过admin用户和激活设置的密码登录设备，根据使用需求进行用户管理，包括：

1. admin用户修改密码，用户名不可修改；
2. 添加、修改、删除其他用户，包括用户类型、用户密码、用户名等。非admin用户创建后，即可登录相应账户进行其他操作。

注释：

1. 常见用户类型：
 - 管理员（admin）：拥有访问设备所有支持资源的权限，可使用设备系统所有功能，账户不可删除。
 - 操作员（operator）：具备浏览权限，操作权限由管理员指定，仅管理员可创建。
 - 普通用户（viewer）：仅有浏览权限，没有操作权限，仅管理员可创建。
2. 用户密码：
 - 为了保证用户帐户信息的安全，建议设置的密码长度达到8-16位，且至少由数字、小写字母、大写字母和特殊字符中的两种或两种以上类型组合而成，密码中不能包含用户名。
 - 密码长度小于8位、密码只包含一类字符、密码与用户名一样或密码是用户名的倒写，以上几类密码属于风险密码，为更好保护用户隐私并提升使用安全性，建议将风险密码更改为高强度密码。
 - 密码强度规则如下：
 - a. 如果设置的密码包含三种或三种以上类型（数字、小写字母、大写字母、特殊字符），属于强密码。
 - b. 如果设置的密码为数字和特殊字符组合、小写字母和特殊字符组合、大写字母和特殊字符组合、小写字母和大写字母组合中的一种，属于中密码。
 - c. 如果设置的密码为数字和小写字母组合、数字和大写字母组合，属于弱密码。

5.2 集成流程

1. 客户端获取设备用户管理参数能力：GET /ISAPI/Security/users/<indexID>/capabilities;
2. 客户端添加设备用户：POST /ISAPI/Security/users?security=<security>&iv=<iv>;

注释：

- 仅admin用户可添加其他用户，且需进行admin登录密码验证（<loginPassword>）；
- 新增用户初始状态为未激活，需登录账户并对密码进行修改（PUT /ISAPI/Security/users/<indexID>?security=<security>&iv=<iv>）后，修改成功后才转为已激活状态；
- 新增用户账户处于未激活时，除可进行密码修改外，设备其他功能均不可用，若进行修改密码外的操作，设备返回错误码：0x0020000f;

3. 客户端修改用户信息：PUT /ISAPI/Security/users/<indexID>?security=<security>&iv=<iv>;

注释：

- admin用户修改用户密码时，需进行admin登录密码验证（<loginPassword>），用户密码修改后，账户转为未激活状态。一旦用户退出登录，再次登录需先进行密码修改；
- 非admin用户修改自身密码，账户仍为已激活状态；

4. 客户端删除用户：DELETE /ISAPI/Security/users?loginPassword=<loginPassword>&security=<security>&iv=<iv>;

注释：

o 仅admin用户可删除用户，需进行admin登录密码验证（<loginPassword>）；

5. 客户端获取设备用户信息，包括：用户名、用户激活状态（<userActivationStatus>）等：

获取单个用户信息：GET /ISAPI/Security/users/<indexID>?security=<security>&iv=<iv>;

获取全部用户信息：GET /ISAPI/Security/users?security=<security>&iv=<iv>;

获取在线用户信息：GET /ISAPI/Security/onlineUser。在线用户为当前登录到设备的全部用户，可查看：登录用户名、用户类型、IP地址信息等。

注释：

- 添加、修改、删除用户过程中，若admin登录密码验证出现多次失败，会触发设备锁定机制（同用户登录认证多次失败），锁定前剩余尝试认证次数见响应字段retryTimes。
- 新设置密码不能与原密码相同，若相同返回错误码：0x400010E8。

5.3 异常处理

错误码

statusCode	statusString	subStatusCode	errorCode	errorMsg	中文描述	备注
4	Invalid Operation	cannotSameAsOldPassword	0x400010E8	New password cannot be the same as the old one.	不能与旧密码相同，请更换密码，再进行尝试	
4	Invalid Operation	theAccountIsNotActivated	0x0020000f	The account is not activated	用户账户未激活	
4	Invalid Operation	loginPasswordError	0x4000000C	Incorrect login password.	登录密码错误	
4	Invalid Operation	theAnswerToTheUserSecurityQuestionIsDuplicate	0x4000A0B6	The answer to the user security question is duplicate.	用户安全问题答案重复，请重新设置答案，再进行尝试。	用户安全问题答案重复，请重新设置答案，再进行尝试。

4	Invalid Operation	theAnswerToTheUserSecurityQuestionIsTooShort	0x4000A0B7	The answer to the user security question is too short.	用户安全问题答案过短，请重新设置答案，再行尝试。
4	Invalid Operation	cannotSameAsOldPassword	0x400010E8	New password cannot be the same as the old one.	不能与旧密码相同，请更换密码，再行尝试。
6	Invalid Content	administratorPasswordError	0x60000042	Incorrect administrator password.	请确认密码，若正确的，联系管理员重置用户密码；若管理员密码遗忘的，可设置进行密码重置。

6 时间校时

6.1 功能介绍

时间校时是指针对网络内所有具有时钟的设备进行时钟同步，使网络内所有设备的时钟保持一致，从而使设备能够提供基于统一时间的多种应用。当前支持的校时类型有：NTP校时、手动校时、卫星校时、萤石平台校时、GB28181校时等。当前重点说明下NTP的校时原理：

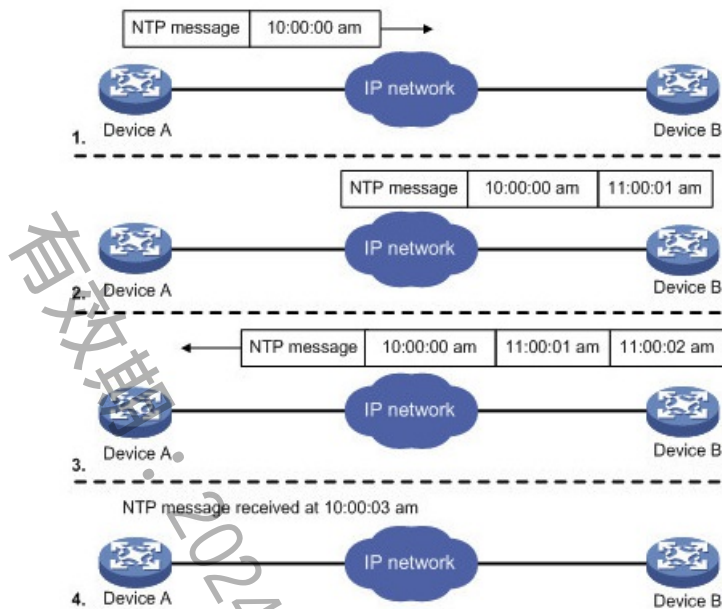
6.1.1 NTP校时

运行NTP的本地系统，既可以接收来自其他时钟源的同步（自身作为客户端），又可以作为时钟源同步其他的时钟（自身作为服务端），并且可以和其他设备互相同步。

NTP的基本工作原理如图所示。Device A和Device B通过网络相连，它们都有自己独立的系统时钟，需要通过NTP实现各自系统时钟的自动同步。为便于理解，作如下假设：

在Device A和Device B的系统时钟同步之前，Device A的时钟设定为10:00:00am，Device B的时钟设定为11:00:00am。Device B作为NTP时间服务器，即Device A将使自己的时钟与Device B的时钟同步。

NTP报文在Device A和Device B之间单向传输所需要的时间为1秒。



系统时钟同步的工作过程如下：

Device A发送一个NTP报文给Device B，该报文带有它离开Device A时的时间戳，该时间戳为10:00:00am（T1）。

当此NTP报文到达Device B时，Device B加上自己的时间戳，该时间戳为11:00:01am（T2）。

当此NTP报文离开Device B时，Device B再加上自己的时间戳，该时间戳为11:00:02am（T3）。

当Device A接收到该响应报文时，Device A的本地时间为10:00:03am（T4）。

至此，Device A已经拥有足够的信息来计算两个重要的参数：

NTP报文的往返时延 $Delay = (T4 - T1) - (T3 - T2) = 2$ 秒。

Device A相对Device B的时间差 $offset = ((T2 - T1) + (T3 - T4)) / 2 = 1$ 小时。

这样，Device A就能够根据这些信息来设定自己的时钟，使之与Device B的时钟同步。

6.2 集成流程

6.2.1 时间校时配置

1. 获取设备校时管理能力

可通过该接口获取到设备当前所支持的校时类型，当前支持的校时类型有：NTP校时、手动校时、卫星校时、萤石平台校时、GB28181校时等。

获取能力:GET /ISAPI/System/time/capabilities。

2. 配置设备校时管理

可通过该接口配置设备当前的校时类型：

NTP校时：请查看NTP校时（客户端）章节。

手动校时：将timeMode配置为manual后，可通过localTime、timeZone字段配置当前设备本地时间。

卫星校时：将timeMode配置为satellite后，可通过satelliteInterval配置卫星校时间隔时间。

萤石校时：将timeMode配置为platform后，设备则通过萤石平台进行校时。

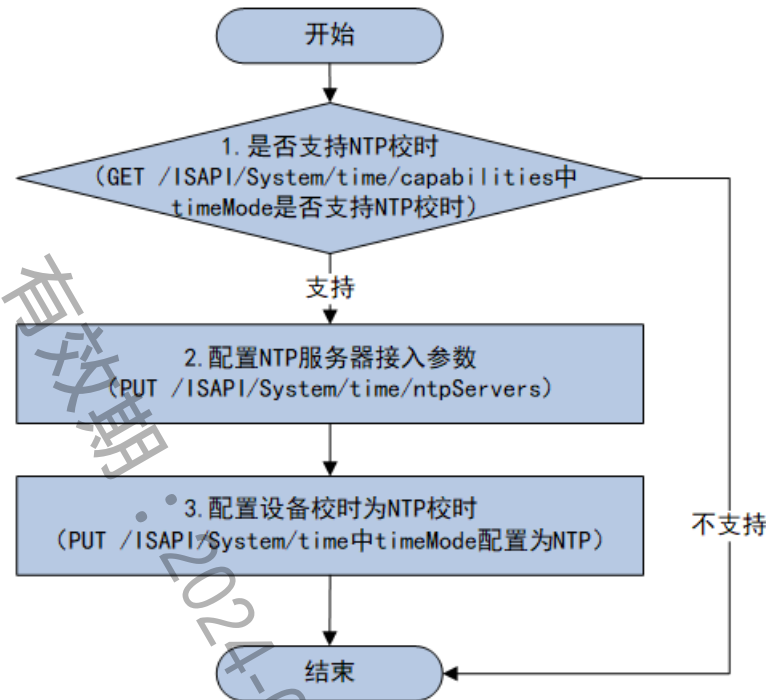
GB28181校时：将timeMode配置为GB28181后，设备则通过GB28181协议进行校时。若存在双国标平台的情况，可通过platformNo 字段选择平台号。platformNo 字段

在/ISAPI/System/Network/SIP/<SIPServerID>中GB28181List 下platformNo所配置,双国标平台唯一标识。

获取设备校时管理参数，GET /ISAPI/System/time； 配置设备校时管理参数， PUT /ISAPI/System/time；

6.2.2 NTP校时（客户端）

运行NTP的本地系统，既可以接收来自其他时钟源的同步（自身作为客户端），又可以作为时钟源同步其他的时钟（自身作为服务端），并且可以和其他设备互相同步。这里为自身作为客户端时的集成流程：



1. 判断设备是否支持是否支持NTP校时 获取能力:GET /ISAPI/System/time/capabilities， timeMode 是否支持NTP校时类型;

2. 配置NTP服务器接入参数

通过该接口配置所需要接入的NTP服务器地址，由该服务器负责时间同步。

获取NTP服务器接入参数能力:GET /ISAPI/System/time/ntpServers/capabilities

配置NTP服务器接入参数:PUT /ISAPI/System/time/ntpServers

获取NTP服务器接入参数:GET /ISAPI/System/time/ntpServers

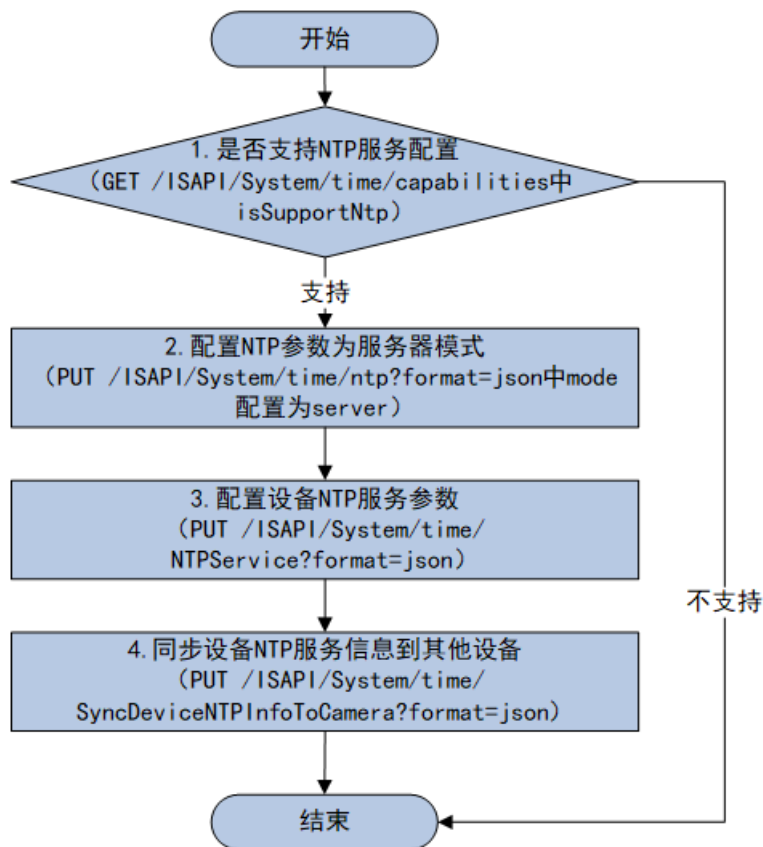
3. 配置设备校时为NTP校时

用户可通过该接口将设备切换为NTP校时，即将timeMode配置为NTP

获取设备校时管理参数， GET /ISAPI/System/time； 配置设备校时管理参数， PUT /ISAPI/System/time；

6.2.2 NTP校时（服务端）

运行NTP的本地系统，既可以接收来自其他时钟源的同步（自身作为客户端），又可以作为时钟源同步其他的时钟（自身作为服务端），并且可以和其他设备互相同步。这里为自身作为服务端时的集成流程：



1. 判断设备是否支持是否支持NTP服务配置 获取系统能力:GET /ISAPI/System/time/capabilities, 返回isSupportNtp表示支持;

2. 配置NTP参数为服务器模式

通过该接口将NTP模式切换为服务器模式, 即mode配置为server。

获取NTP参数为服务器模式能力: GET /ISAPI/System/time/ntp/capabilities?format=json;

配置NTP参数为服务器模式: PUT /ISAPI/System/time/ntp?format=json;

获取NTP参数为服务器模式: GET /ISAPI/System/time/ntp?format=json;

3. 配置设备NTP服务参数

主要为服务地址, 其他设备可以通过该地址接入。

获取能力: GET /ISAPI/System/time/NTPService/capabilities?format=json

配置参数: PUT /ISAPI/System/time/NTPService?format=json

获取参数: GET /ISAPI/System/time/NTPService?format=json

4. 同步设备NTP服务信息到其他设备

可通过该接口将时间信息同步给其他设备。

获取同步设备NTP服务信息到相机能力: GET /ISAPI/System/time/SyncDeviceNTPInfoToCamera/capabilities?format=json

同步设备NTP服务信息到相机: PUT /ISAPI/System/time/SyncDeviceNTPInfoToCamera?format=json

获取同步设备NTP服务信息到相机进度: GET /ISAPI/System/time/SyncDeviceNTPInfoToCamera/Progress?format=json

查询同步设备NTP服务信息到相机结果: POST /ISAPI/System/time/SyncDeviceNTPInfoToCamera/SearchResult?format=json

7 布防与订阅

7.1 功能介绍

客户端主动向设备建立HTTP长连接, 持续接收设备产生的事件消息。

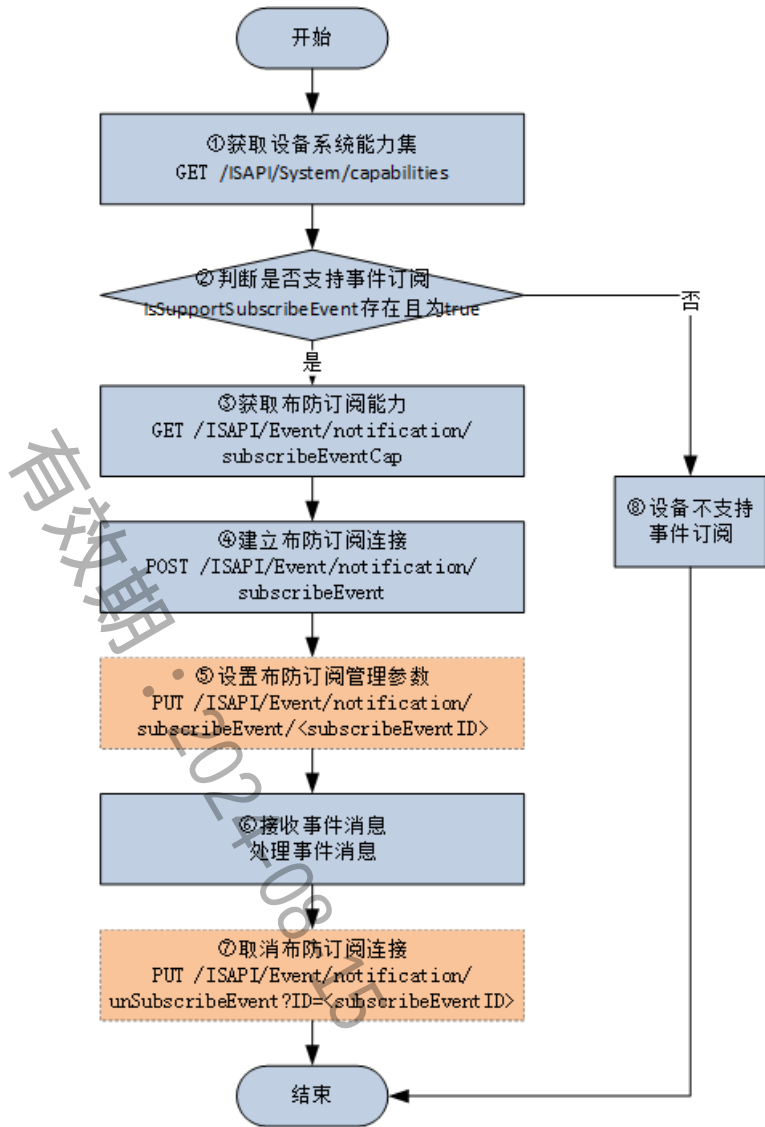
ISAPI事件布防有非订阅和订阅两种方式接收设备产生的事件消息，区别在于非订阅使用HTTP GET方法接收设备所有的事件消息，订阅使用HTTP POST方法只接收订阅列表中的事件消息。

7.2 集成流程

7.2.1 非订阅

1. 建立布防非订阅连接GET /ISAPI/Event/notification/alertStream，客户端需要设置HTTP Headers中Connection: keep-alive；
2. 接收设备发送的事件，通过boundary分割解析事件消息，解析说明详见下面的“事件消息解析”；
3. 不需要接收事件消息时关闭布防连接。

7.2.2 订阅



1. 获取设备系统能力集GET /ISAPI/System/capabilities；
2. 判断是否支持事件订阅isSupportSubscribeEvent存在且为true；
3. 获取布防订阅能力GET /ISAPI/Event/notification/subscribeEventCap；
4. 建立布防订阅连接POST /ISAPI/Event/notification/subscribeEvent客户端需要设置HTTP Headers中Connection: keep-alive；
5. （可选）修改已有布防订阅连接的订阅参数，如果要在当前订阅配置基础上修改订阅参数，需要先获取布防订阅管理参数GET /ISAPI/Event/notification/subscribeEvent/<subscribeEventID>,在获取到的订阅配置基础上修改完成后再调用设置布防订阅管理参数PUT /ISAPI/Event/notification/subscribeEvent/<subscribeEventID>;
6. 接收设备发送的事件，通过boundary分割解析事件消息，解析说明详见下面的“事件消息解析”；
7. （可选）取消布防订阅连接PUT /ISAPI/Event/notification/unSubscribeEvent?ID=<subscribeEventID>，当使

用HTTP方式直接和设备通信时，不需要调用此接口，直接断开布防连接即可。通过萤石、OTAP等方式透传ISAPI事件订阅时，关闭事件订阅时需要调用此接口通知设备释放资源；

8. 设备不支持事件订阅；

说明：通过萤石、OTAP透传ISAPI事件时，只要成功建立了ISAPI布防订阅连接，设备就通过透传链路上报ISAPI事件，并且和萤石、OTAP本身的事件互不影响。比如设备通过OTAP接入平台，平台既订阅了设备OTAP事件中的移动侦测，又通过透传链路订阅了ISAPI事件中的移动侦测，当设备触发移动侦测事件时，平台会同时收到此设备OTAP的移动侦测事件消息和透传ISAPI的移动侦测事件消息。

7.2.3 事件消息解析

客户端向设备建立布防连接之后，设备发送数据的格式为HTTP表单格式（multipart/form-data）。HTTP请求Headers中的Content-Type通常为multipart/form-data，boundary=AaB03x，其中boundary是一个变量，用于将整个HTTP Body分割成多个单元，每个单元为一份数据，都有各自的Headers和body。详细格式说明参考RFC 1867 (Form-based File Upload in HTML)，示例如下，请注意boundary前面和后面的横线--。

```
HTTP/1.1 200 OK
Content-Type: multipart/form-data; boundary=AaB03x
Connection: keep-alive
--AaB03x
Content-Disposition: form-data; name="ANPR.xml"; filename="ANPR.xml";
Content-Type: application/xml
Content-Length: 9

<ANPR/>
--AaB03x
Content-Disposition: form-data; name="licensePlatePicture.jpg"; filename="licensePlatePicture.jpg";
Content-Type: image/jpeg
Content-Length: 14

图片数据
--AaB03x--
```

其中部分关键字的说明如下。

关键字	示例	描述
Content-Type	multipart/form-data; boundary=AaB03x	内容类型，multipart/form-data表示这是一个表单格式的消息。
boundary	AaB03x	表单消息分隔符，--boundary 为一段表单的开始，--boundary- 为整个HTTP表单消息的结束
Content-Disposition	form-data; name="ANPR.xml"; filename="ANPR.xml";	内容描述，form-data表示是一段表单数据
name	"ANPR.xml"	表单名称，此段表单的名称
filename	"ANPR.xml"	文件名称，此段表单为一个文件，此文件的名称
Content-Length	9	内容长度，从下一个\r\n之后开始，到下一个 --boundary 之前的长度

7.3 限制说明

需要注意的是，ISAPI布防（包括订阅和非订阅）使用的是HTTP/HTTPS长连接，由于HTTP的单工特点，在布防连接建立成功之后，设备会持续不断的发送事件消息，此时是不支持客户端在此连接上再发送任何消息给设备。

客户端在超过心跳时间后没再收到任何设备消息时应主动断开布防连接，尝试重新建立新的布防连接。

7.4 示例报文

7.4.1 客户端创建布防订阅

```
POST /ISAPI/Event/notification/subscribeEvent HTTP/1.1
Authorization: Digest username="admin", realm="IP
Camera(C2183)", nonce="4e5468694e7a42694e7a4d364f4449354d7a6b354d54513d", uri="/ISAPI/Event/notification/alertStream", cnonce="3d183a245b8729121ae4ca3d41b90f18
", nc=00000001, qop="auth", response="f2e0728991bb031f83df557a8f185178"
Host: device_ip

<SubscribeEvent/>
```

7.4.2 服务端响应以及推送事件消息

```
HTTP/1.1 200 OK
MIME-Version: 1.0
Connection: close
Content-Type: multipart/mixed; boundary=<frontier>

--<frontier>
Content-Type: application/xml; charset="UTF-8" <!-- 部分报警走json格式，所以要求上层解析的时候，根据Content-Type字段进行区分-->
Content-Length: text_length

<SubscribeEventResponse/>
--<frontier>
Content-Type: application/xml; charset="UTF-8" <!-- 部分报警走json格式，所以要求上层解析的时候，根据Content-Type字段进行区分-->
Content-Length: text_length

<EventNotificationAlert version="2.0" xmlns="http://www.hikvision.com/ver20/XMLSchema">
  <ipAddress>10.17.133.46</ipAddress>
  <portNo>80</portNo>
  <protocol>HTTP</protocol>
  <macAddress>44:19:b6:6d:24:85</macAddress>
  <channelID>1</channelID>
  <dateTime>2017-05-04T11:20:02+08:00</dateTime>
  <activePostCount>0</activePostCount>
  <eventType>heartBeat</eventType>
  <eventState>active</eventState>
  <eventDescription>heartBeat</eventDescription>
</EventNotificationAlert>
--<frontier>
Content-Disposition: form-data; name="Picture_Name"
Content-Type: image/jpeg
Content-Length: image_length
Content-ID: image_ID

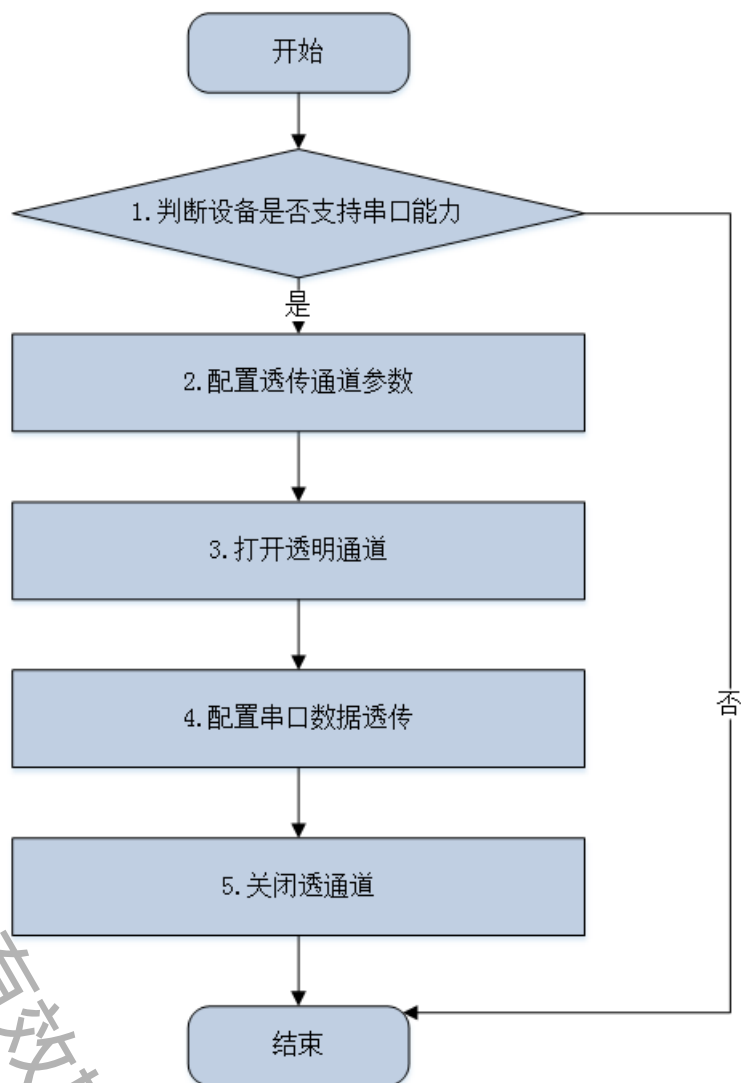
[图片数据]
--<frontier>
```

8 串口数据透传

8.1 功能介绍

将设备外接的RS485、RS422和RS232串口作为透明通道，传输串口数据。

8.2 集成流程



1.判断设备是否支持串口数据传输:

获取设备串口支持透传的能力节点 `GET /ISAPI/System/capabilities`;如果报文返回节点 `SerialCap` 为 `true` 表示设备支持串口相关功能。

2.配置透明通道列表参数，包含如下:

获取指定透明通道参数 `GET /ISAPI/System/Serial/ports/<portID>/Transparent/channels/<channelID>;`

配置指定透明通道参数 `GET /ISAPI/System/Serial/ports/<portID>/Transparent/channels/<channelID>;`

3.打开透明通道: `PUT /ISAPI/System/Serial/ports/<portID>/Transparent/channels/<channelID>/open;`

4.串口数据传输配置，包含如下:

获取串口数据传输: `GET /ISAPI/System/Serial/ports/<portID>/Transparent/channels/<channelID>/transData;`

配置串口数据传输: `PUT /ISAPI/System/Serial/ports/<portID>/Transparent/channels/<channelID>/transData;`

5.关闭透明通道: `PUT /ISAPI/System/Serial/ports/<portID>/Transparent/channels/<channelID>/close;`

9 串口参数控制

9.1 功能介绍

管理串口参数配置。

9.2 集成流程

1.判断设备是否支持串口参数控制:

获取设备串口参数控制的能力节点 `GET /ISAPI/System/capabilities`;如果报文返回节点 `SerialCap` 为 `true` 表示设备支持串口相关功能。

3.全部串口控制参数获取，包含如下:

获取所有串口能力: GET /ISAPI/System/Serial/capabilities;

获取全部串口控制参数: GET /ISAPI/System/Serial/ports?permissionController=<indexID>;

4.单个串口控制参数配置, 包含如下:

获取单个串口控制能力集: GET /ISAPI/System/Serial/ports/<portID>/capabilities;

获取单个串口控制参数: GET /ISAPI/System/Serial/ports/<portID>/deviceInfo?format=json;

配置单个串口控制参数: PUT /ISAPI/System/Serial/ports/<portID>/deviceInfo?format=json;

5.获取单个串口状态: GET /ISAPI/System/Serial/ports/<portID>/status.

10 串口设备管理

9.1 功能介绍

管理设备串口外接的设备信息, 包含两个方面: 一、配置指定串口接入设备的厂商、类型、型号信息; 二、查询指定串口下支持哪些厂商的设备类型或型号接入;

9.2 集成流程

1.判断设备是否支持串口设备信息管理: GET /ISAPI/System/Serial/capabilities;如果报文返回节点<isSupportDeviceInfo>表示设备支持串口设备信息配置。

3.配置串口设备信息:

获取单个串口设备信息参数能力: GET /ISAPI/System/Serial/ports/<portID>/deviceInfo?format=json;

获取单个串口设备信息参数: GET /ISAPI/System/Serial/ports/<portID>/deviceInfo?format=json;

配置单个串口设备信息参数: PUT /ISAPI/System/Serial/ports/<portID>/deviceInfo?format=json;

4.判断设备是否支持串口设备信息关联: GET /ISAPI/System/Serial/capabilities;如果报文返回节点<isSupportSearchDeviceInfoRelations>表示设备支持串口设备信息配置。

5.查询串口设备信息关联:

获取查询单个串口设备信息关联参数能力: GET

/ISAPI/System/Serial/ports/<portID>/searchDeviceInfoRelations/capabilities?format=json;

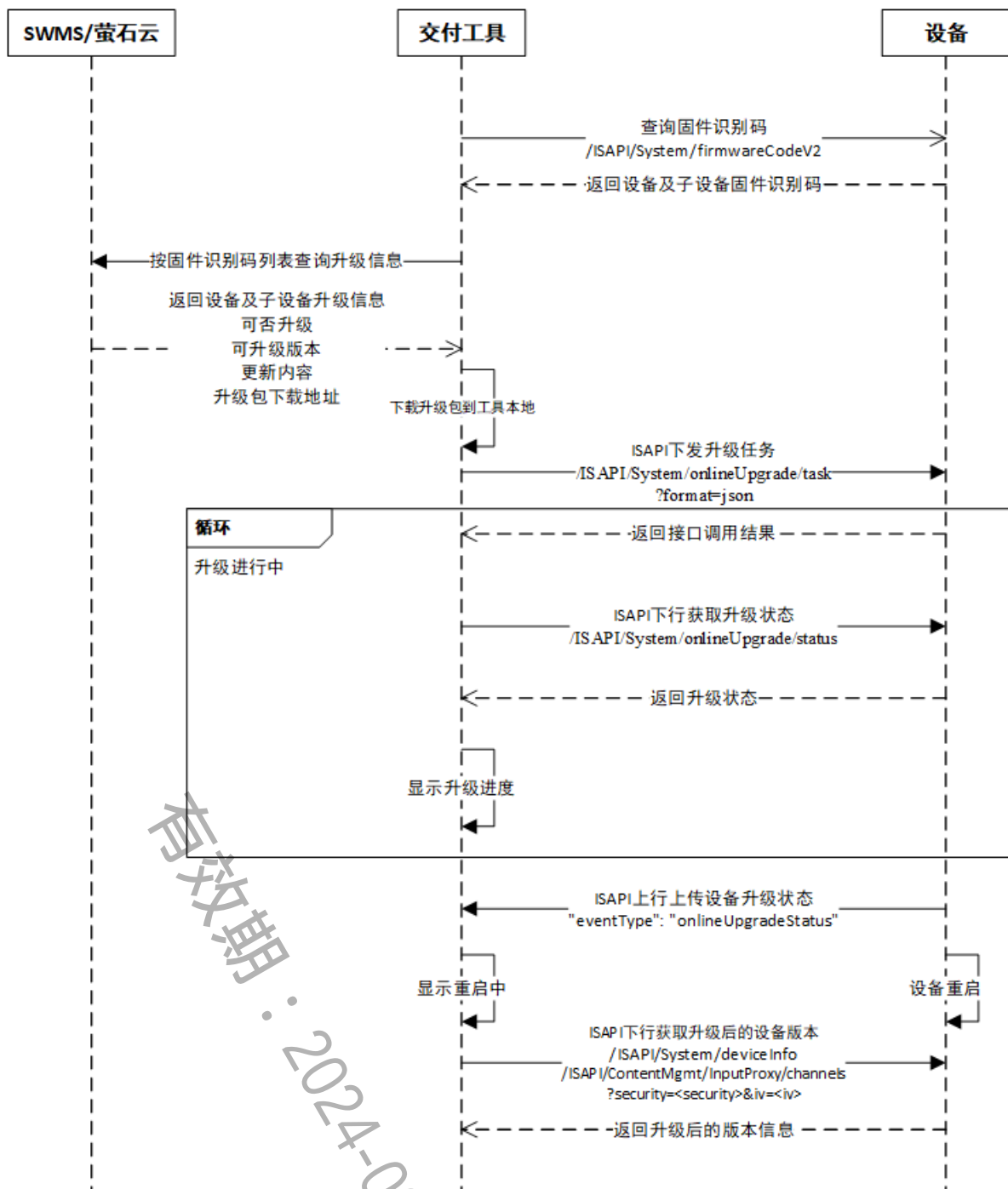
查询单个串口设备信息关联参数: POST /ISAPI/System/Serial/ports/<portID>/searchDeviceInfoRelations?format=json;

11 基于本地化部署交互工具升级

11.1 功能场景介绍

该方案流程适用于ISAPI协议使用的场景, 用于支持设备和子设备升级。在ISAPI协议下, 需要搭建HTTP(s)服务, 用于存储设备升级包, 生成可以下载升级包的URL。如果设备HCNetSDK协议接入的, 可以通过透传ISAPI协议报文的机制满足应用需求。

11.2 集成流程



1.工具向设备查询固件识别码: GET /ISAPI/System/firmwareCodeV2; 2.根据固件识别码在SWMS/萤石云上匹配到设备升级信息,并下载升级包到本地; 下发升级任务到设备: POST /ISAPI/System/onlineUpgrade/task?format=json; 任务中可以通过<channel>字段指定是设备本地升级还是通道设备升级;

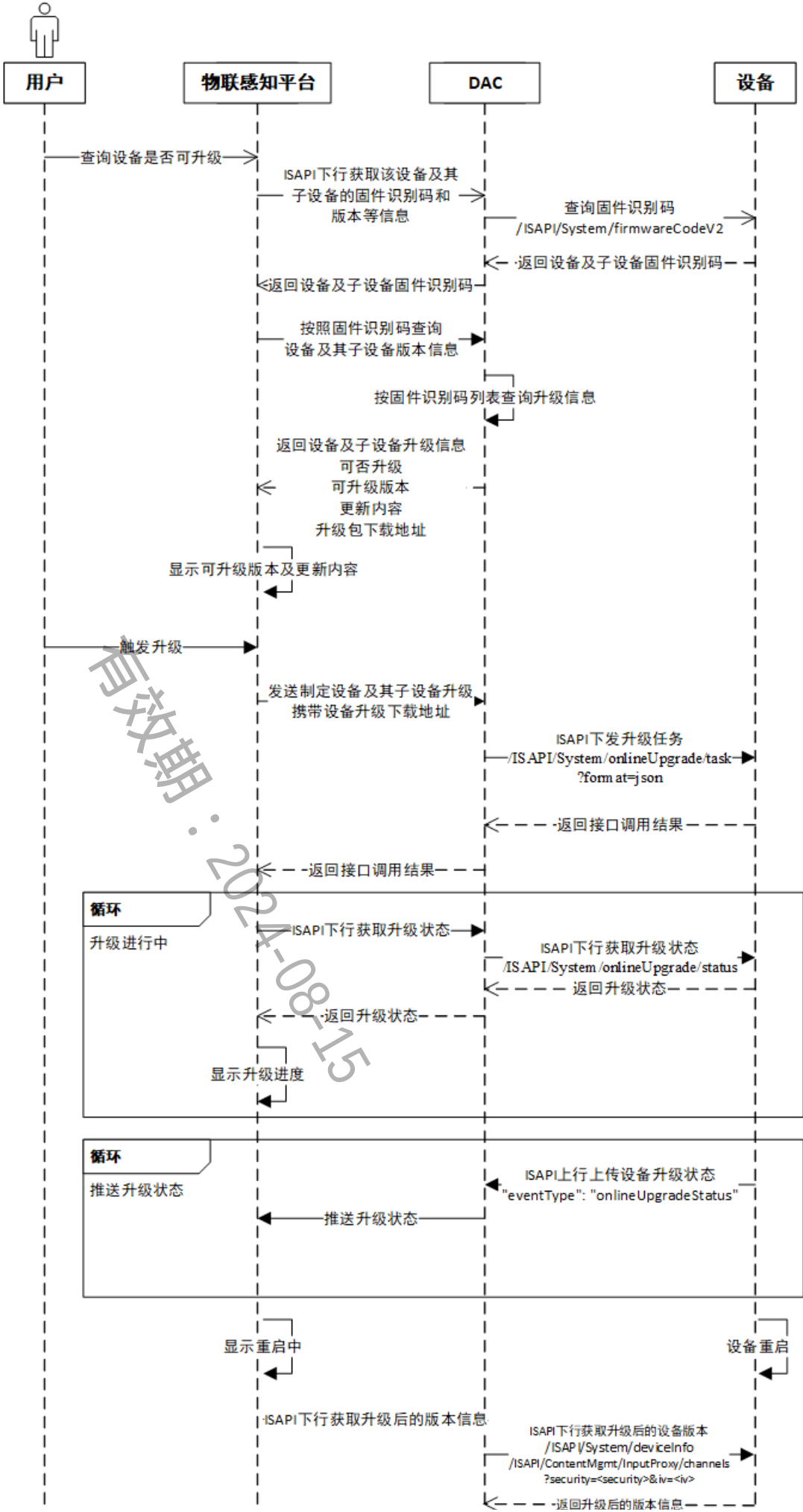
3.设备升级过程中,工具可以实时获取升级进度: GET /ISAPI/System/onlineUpgrade/status; 4.设备升级完成后,上报升级状态"onlineUpgradeStatus",并自动进入重启; 5.升级重启完成后,工具可以重新获取升级后的设备版本信息: 获取设备版本信息: GET /ISAPI/System/deviceInfo; 获取通道设备版本信息: GET /ISAPI/ContentMgmt/InputProxy/channels?security=<security>&iv=<iv>.

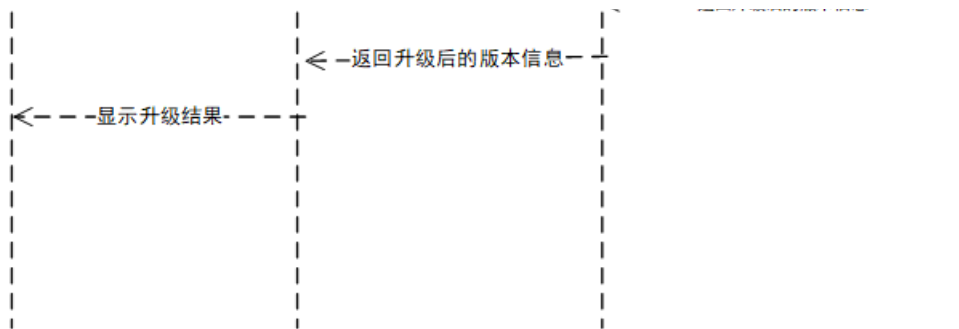
12 基于本地化部署平台升级

12.1 功能场景介绍

该方案流程适用于ISAPI, ISUP协议使用的场景,用于支持设备和子设备升级。在ISAPI协议下,需要搭建HTTP(s)服务或者直接传输二进制升级包文件,用于存储设备升级包,生成可以下载升级包的URL。在ISUP协议下,需要搭建FTP服务,用于存储设备升级包,生成可以下载升级包的FTP地址。如果设备通过HCNetSDK协议接入的,可以通过透传ISAPI协议报文的机制满足应用需求。

12.2 集成流程





1.用户通过平台向设备查询固件识别码: GET /ISAPI/System/firmwareCodeV2; 2.平台根据固件识别码向DAC查询设备升级信息; 如有可升级版本, 则触发升级任务, 通过DAC下发升级任务到设备: POST /ISAPI/System/onlineUpgrade/task?format=json; 任务中可以通过<channel>字段指定是设备本地升级还是通道设备升级;

3.设备升级过程中, 平台可以实时获取升级进度: GET /ISAPI/System/onlineUpgrade/status; 4.设备升级完成后, 上报升级状态"onlineUpgradeStatus", 并自动进入重启; 5.升级重启完成后, 平台可以重新获取升级后的设备版本信息: 获取设备版本信息: GET /ISAPI/System/deviceInfo; 获取通道设备版本信息: GET /ISAPI/ContentMgmt/InputProxy/channels?security=<security>&iv=<iv>.

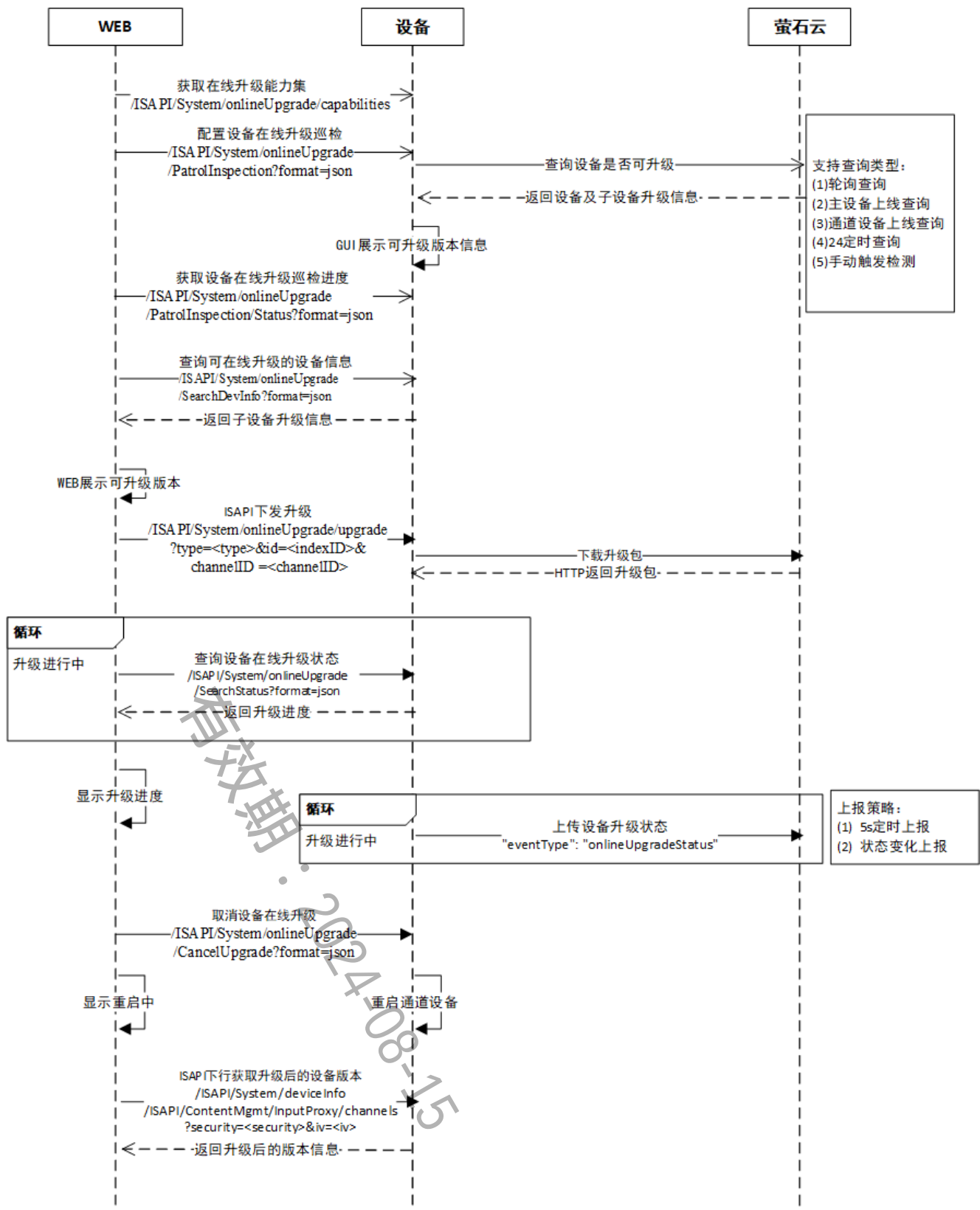
13 基于WEB客户端远程在线升级

11.1 功能场景介绍

该方案用于设备接入WEB, 支持设备和子设备的远程在线升级。

11.2 集成流程

有效期: 2024-08-15



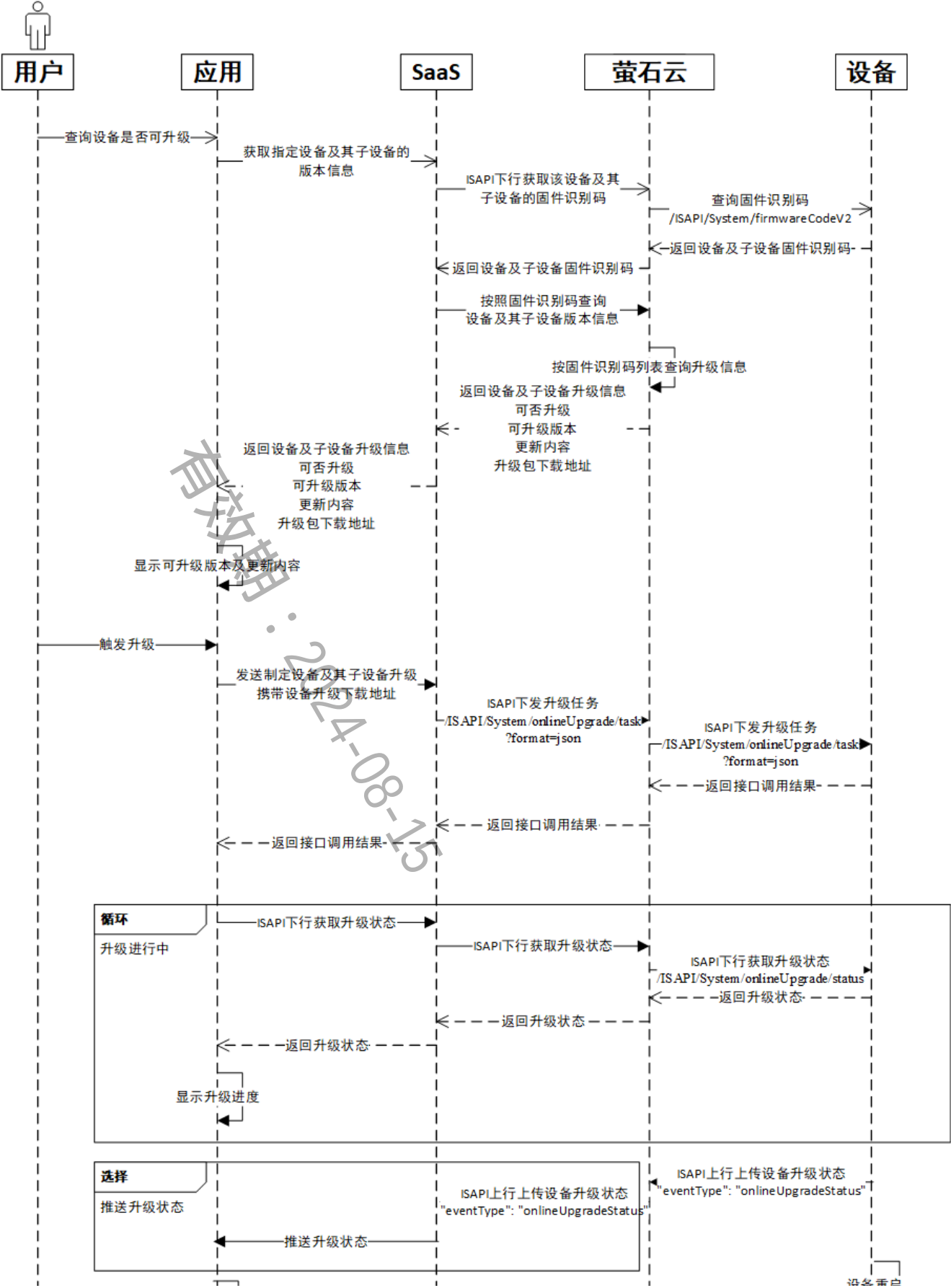
1. 获取设备在线升级能力: GET /ISAPI/System/onlineUpgrade/capabilities; 2. 手动触发设备在线升级巡检: POST /ISAPI/System/onlineUpgrade /PatrolInspection?format=json; 配置后, 设备会向萤石云查询是否有可升级版本; 3. 设备巡检过程中, WEB可以实时获取巡检进度: GET /ISAPI/System/onlineUpgrade /PatrolInspection/Status?format=json; 4. 设备巡检完成后, 查询可在线升级的设备信息: GET /ISAPI/System/onlineUpgrade/SearchDevInfo?format=json; 可以查询到通道设备和子设备的可升级信息; 5. 下发升级命令: PUT /ISAPI/System/onlineUpgrade/upgrade?type=<type>&id=<indexID>&channelID =<channelID>; URL 中<channelID>指定通道设备升级, 无<channelID>表示设备升级; <type>字段指定子设备升级; 6. 设备升级过程中, 可查询设备在线升级状态: GET /ISAPI/System/onlineUpgrade/SearchStatus?format=json; 7. 设备升级过程中, 可取消设备在线升级: PUT /ISAPI/System/onlineUpgrade /CancelUpgrade?format=json ; 8. 升级重启完成后, 可以重新获取升级后的设备版本信息: 获取设备版本信息: GET /ISAPI/System/deviceInfo 获取通道设备版本信息: GET /ISAPI/ContentMgmt/InputProxy/channels?security=<security>&iv=<iv>

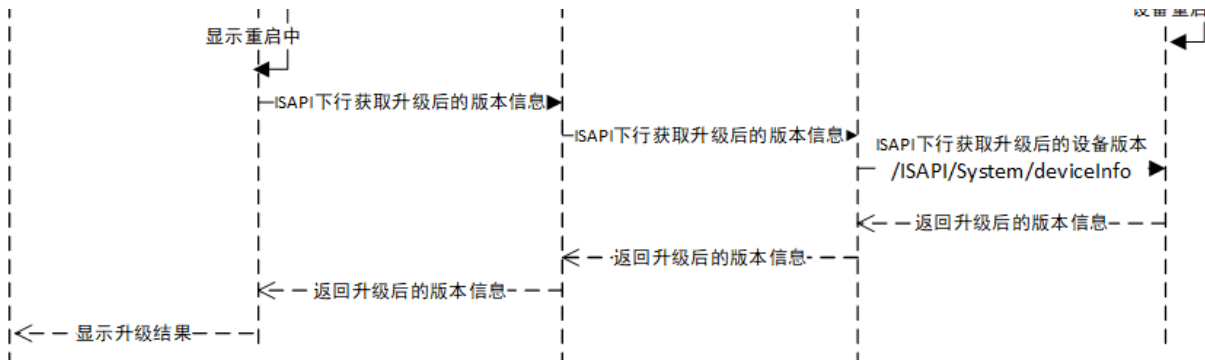
14 基于萤石2.0部署平台升级

11.1 功能场景介绍

该方案由于设备接入萤石云走的是萤石2.0的协议，不支持萤石OTA的升级方案。一般应用于指定版本升级的场景。

11.2 集成流程





1.向设备查询固件识别码: `GET /ISAPI/System/firmwareCodeV2`; 2.根据固件识别码查询设备升级信息; 如有可升级版本, 则下发升级任务到设备: `POST /ISAPI/System/onlineUpgrade/task?format=json`; 任务中可以通过`<channel>`字段指定是设备本地升级还是通道设备升级;
3.设备升级过程中, 平台可以实时获取升级进度: `GET /ISAPI/System/onlineUpgrade/status`; 4.设备升级完成后, 上报升级状态`"onlineUpgradeStatus"`, 并自动进入重启; 5.升级重启完成后, 平台可以重新获取升级后的设备版本信息: 获取设备版本信息: `GET /ISAPI/System/deviceInfo` 获取通道设备版本信息: `GET /ISAPI/ContentMgmt/InputProxy/channels?security=<security>&iv=<iv>`.

15 萤石接入服务参数

15.1 功能介绍

配置萤石接入服务参数, 包括是否开启萤石服务, 服务器地址, 验证码等。

15.2 集成流程

1. 获取萤石接入服务参数能力 `GET /ISAPI/System/Network/EZVIZ/capabilities`
2. 获取萤石接入服务参数 `GET /ISAPI/System/Network/EZVIZ?security=<security>&iv=<iv>`
3. 配置萤石接入服务参数 `PUT /ISAPI/System/Network/EZVIZ?security=<security>&iv=<iv>`

说明: 其中`verificationCode`字段有两个作用, 一个是用户设备添加到萤石APP时, 作为验证码。另一个码流进行加密时, 作为码流加密密钥。

16 萤石云存储配置

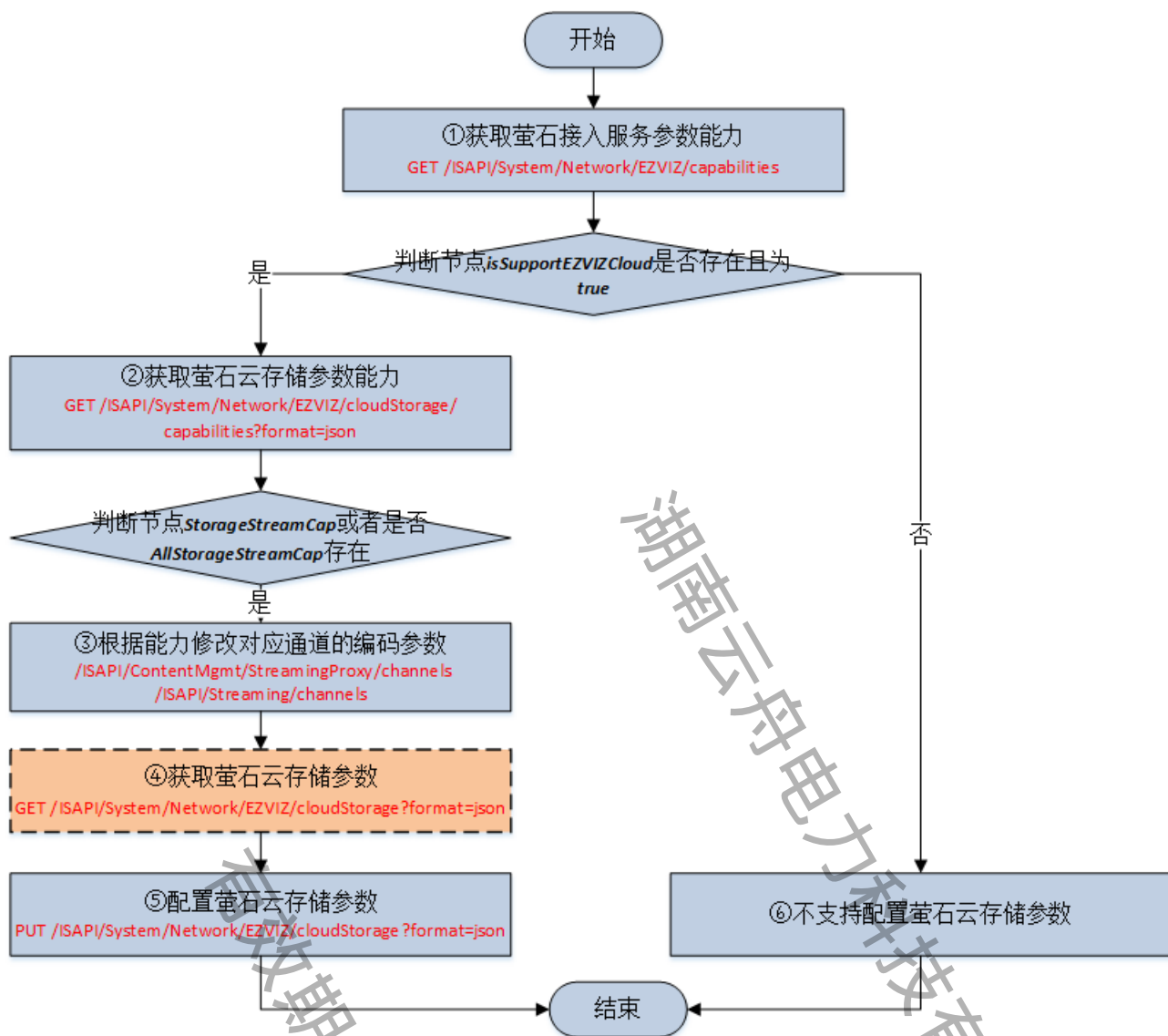
16.1 功能介绍

配置萤石云存储服务参数, 设备根据参数接入对应的萤石云存储服务。

说明:

1. 萤石云服务的基础是萤石接入服务, 注册到萤石服务后, 该服务会自动连接到萤石云服务, 所以云存储不需要配置服务地址等信息。
2. 本配置目前只针对事件录像参数, 报警图片是否上传云存储是报警订阅或者布防时, 指定了图片的上传方式 (如报警订阅:图片格式参数为`/ISAPI/Event/notification/subscribeEvent` 中的`pictureURLType`)

16.2 集成流程



1. 获取萤石接入服务参数能力 GET /ISAPI/System/Network/EZVIZ/capabilities，判断节点isSupportEZVIZCloud存在且为true
2. 获取萤石云存储配置参数能力 GET /ISAPI/System/Network/EZVIZ/cloudStorage/capabilities?format=json
3. （可选）根据步骤2返回的能力修改通道的编码参数
4. （可选）获取萤石云存储参数 GET /ISAPI/System/Network/EZVIZ/cloudStorage?format=json
5. 配置萤石云存储参数 PUT /ISAPI/System/Network/EZVIZ/cloudStorage?format=json 说明：1.该接口是全量配置，配置前需要先获取，确认已开启的通道，防止误操作。2.该接口中的EZVIZCloudStorage.enabled是总开关。当节点为true时，EZVIZCloudStorage.CloudStorage数组有效；当节点为false时，EZVIZCloudStorage.CloudStorage数组无效。
6. 不支持配置萤石云存储参数

说明：产生步骤3原因是设备向萤石云上传视频时需要设备缓存5秒的码流（萤石云服务的限制），可能会让设备的性能不足。设备通过限制编码参数来保证视频流畅性。若不进行此操作，设备上报的码流可能是断断续续的，显示效果不好。

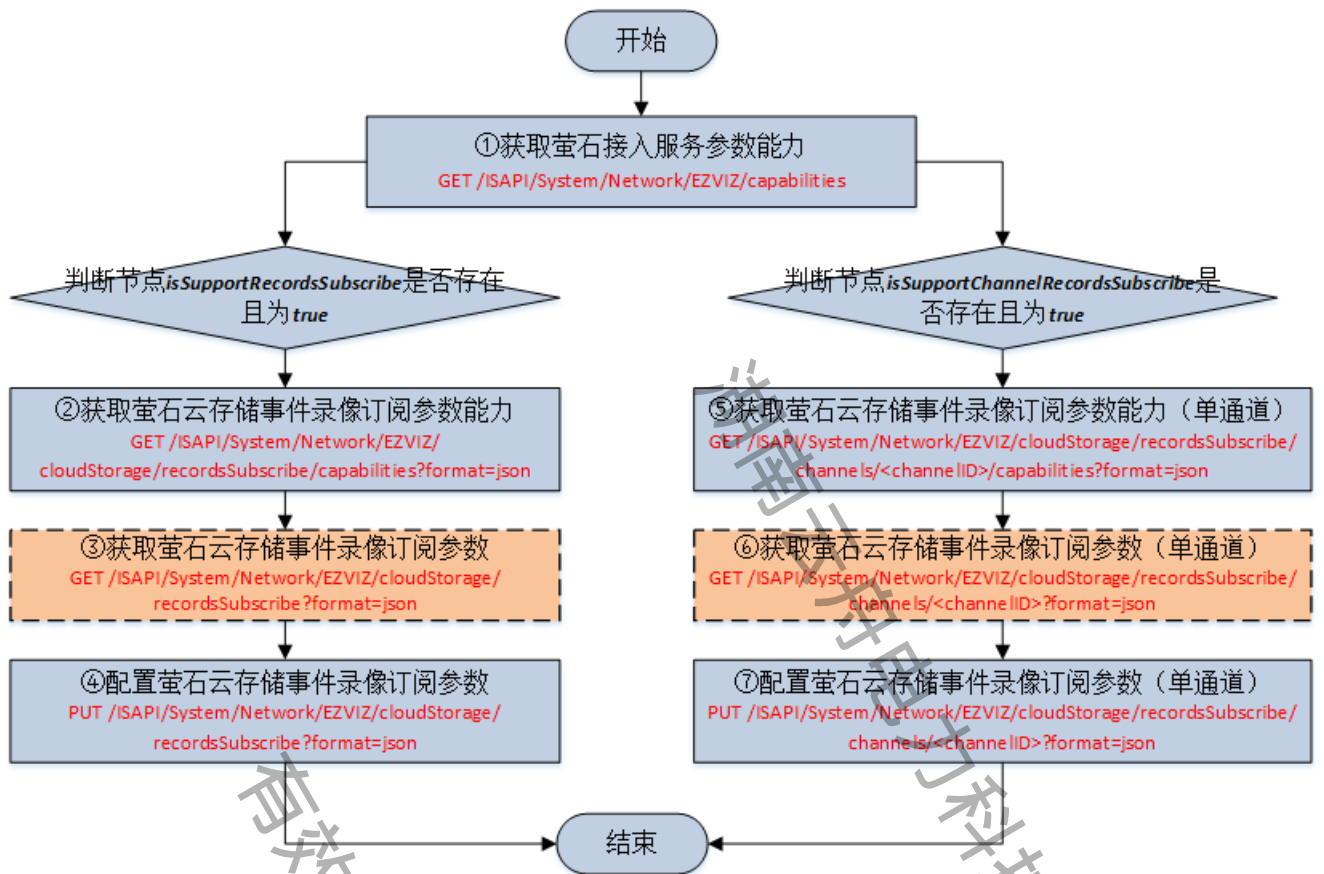
17 萤石云存储事件录像订阅

17.1 功能介绍

支持配置订阅上传到萤石云存储的事件类型（按通道和事件）。

使用场景：从云存储服务角度看，客户如果只需要部分事件录像上传，可以节省云存储空间，理解为相同的2GB空间支持1个客户，现在事件减少了可以支持两个客户。

17.2 集成流程



1. 获取萤石接入服务参数能力 GET /ISAPI/System/Network/EZVIZ/capabilities，判断节点isSupportRecordsSubscribe存在且为true或者判断节点isSupportChannelRecordsSubscribe存在且为true
2. 获取萤石云存储事件录像订阅参数能力 GET /ISAPI/System/Network/EZVIZ/cloudStorage/recordsSubscribe/capabilities?format=json
3. (可选)获取萤石云存储事件录像订阅参数 GET /ISAPI/System/Network/EZVIZ/cloudStorage/recordsSubscribe?format=json
4. 配置萤石云存储事件录像订阅参数 PUT /ISAPI/System/Network/EZVIZ/cloudStorage/recordsSubscribe?format=json
5. 获取萤石云存储事件录像订阅参数能力（单通道） GET /ISAPI/System/Network/EZVIZ/cloudStorage/recordsSubscribe/channels/<channelID>/capabilities?format=json
6. (可选)获取萤石云存储事件录像订阅参数（单通道） GET /ISAPI/System/Network/EZVIZ/cloudStorage/recordsSubscribe/channels/<channelID>?format=json
7. 配置萤石云存储事件录像订阅参数（单通道） PUT /ISAPI/System/Network/EZVIZ/cloudStorage/recordsSubscribe/channels/<channelID>?format=json

18 萤石云存储订阅事件录像统计

18.1 功能介绍

用户订阅需要事件上传云存储后，关注当前事件上传的情况，例如一段时间补录事件个数增长较快，或者某个事件较多，用户可查看事件是否为其想要的，另外补录个数增加较快是否为网络带宽变小了

18.2 集成流程

1. 获取萤石云存储功能能力 `GET /ISAPI/System/Network/EZVIZ/capabilities`，判断节点 `isSupportRecordsSubscribeStatistics` 存在且为 `true`
2. 获取萤石云存储订阅事件录像统计能力 `GET /ISAPI/System/Network/EZVIZ/cloudStorage/recordsSubscribe/statistics/capabilities?format=json`
3. 获取萤石云存储订阅事件录像统计 `POST /ISAPI/System/Network/EZVIZ/cloudStorage/recordsSubscribe/statistics?format=json`

19 萤石物理接触添加设备

19.1 功能介绍

设备在生产和维修过程中，可能会出现重号现象（即两个设备具有同一个设备编码，比如贴错标签、维修操作不规范等）。用户在萤石云APP中添加设备就会发生错误，可能添加了别人的设备，或者自己的设备被别人添加了。为了避免萤石云APP添加错设备，采用物理接触方案验证用户对设备的拥有控制权。

19.2 集成流程

19.2.1 概述

由于设备形态和功能的差异，萤石云APP添加设备的方式有很多种，下面分别按添加方式介绍物理接触集成流程。目前有以下添加方式：

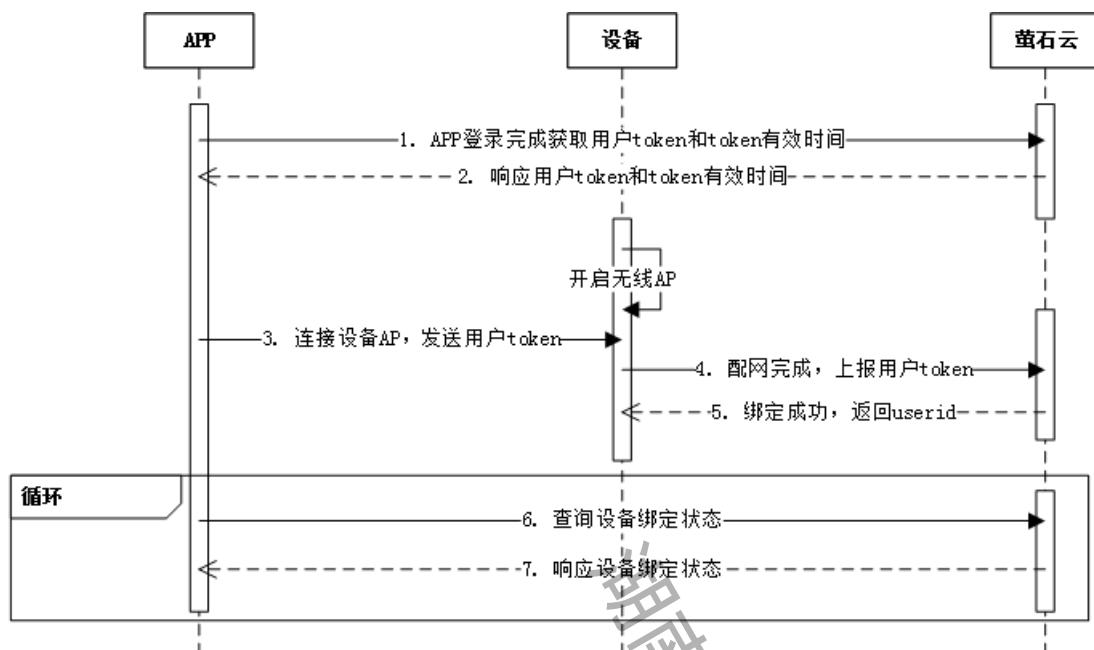
- 无线设备AP
- 有线设备局域网SADP扫描
- 有线设备物理按键接触
- 有线设备APP扫码
- 设备网页手动配置

注：“有线设备APP扫码”添加方式不涉及ISAPI交互，本文中不再介绍相关流程。

通过获取萤石接入服务参数能力 `GET /ISAPI/System/Network/EZVIZ/capabilities` 中返回的 `physicalAccessVerification` 节点判断设备支持的添加方式。

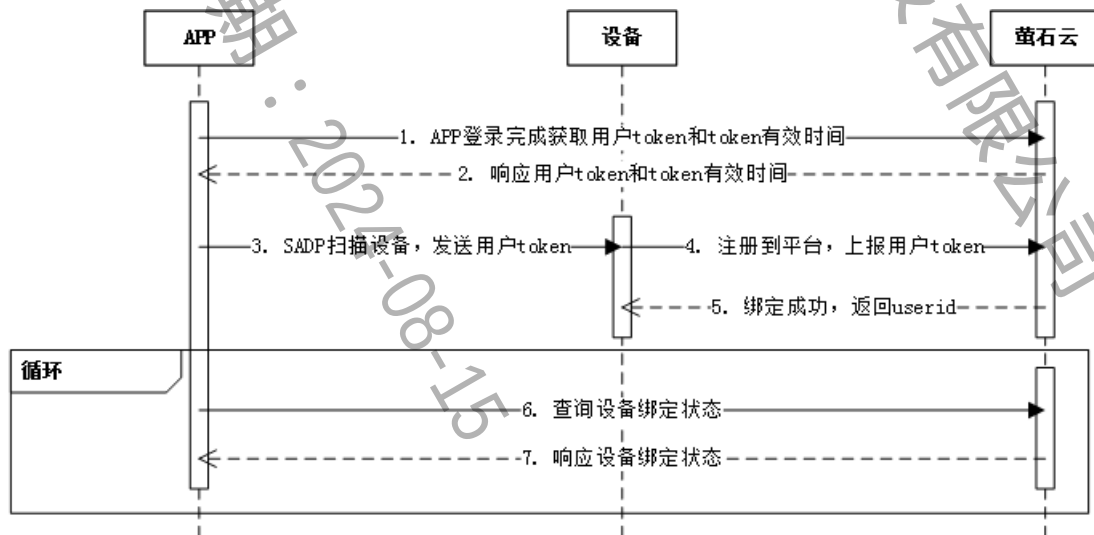
```
<physicalAccessVerification opt="1,2,3,4">
  <!--ro, opt, enum, 设备支持的物理接触式添加方式, subType:string, [1#AP配网传递,2#用户令牌(用户token)绑定,3#物理按键接触,4#扫码绑定(设备token)],
  attr:opt{req, string, 取值范围},
  desc:物理接触式添加—用户告诉平台, 对自己所添加的设备拥有控制权(能触碰到这个设备)
  AP配网传递: 设备和萤石APP通过设备AP热点建立局域网, 萤石APP从萤石平台获取用户token, 下发给设备;
  用户令牌(用户token)绑定: 设备通过有线或者无线连接路由器, 客户端(APP是客户端的一种)与设备在一个局域网内, 客户端从萤石平台获取用户token, 下发给设备, 与方案1的区别就是组网差异;
  物理按键接触: 真实物理接触, 比如在规定时间内要求用户按设备的reset按键, 设备上报一个特定事件, 让平台知道用户在规定的时间内触碰了这个设备;
  扫码绑定(设备token): 设备在联网且未绑定时, 从萤石平台获取有时效的设备token, 结合设备信息生成二维码, 可以扫码添加到萤石账户, 此方法可实现远程添加。-->1
</physicalAccessVerification>
```

19.2.2 无线设备AP



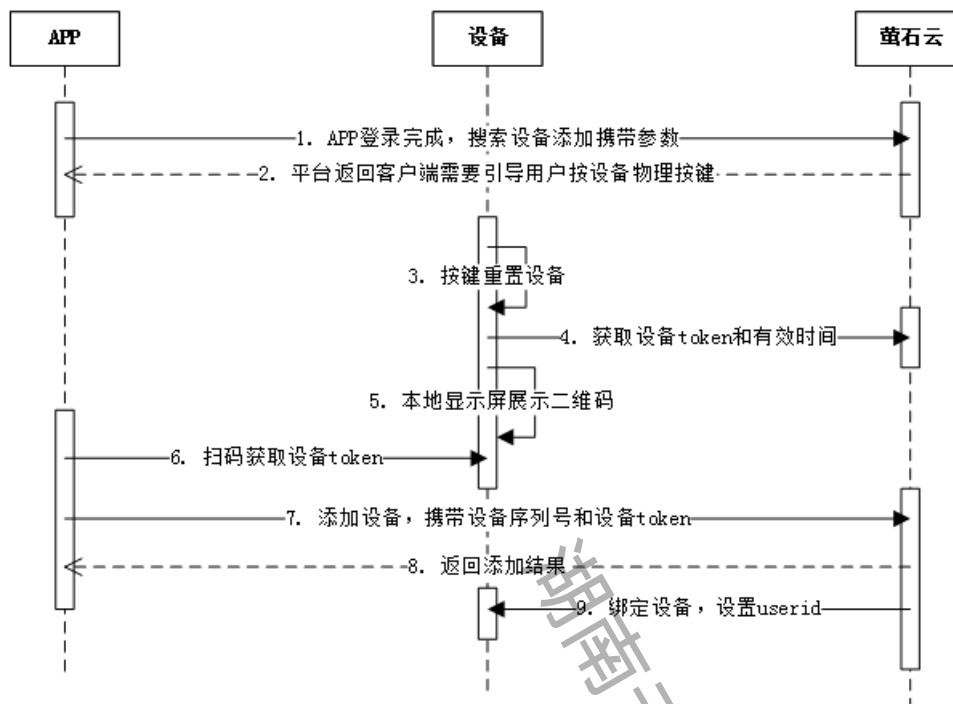
1. APP登录完成获取用户token和token有效时间；
2. 响应用户token和token有效时间；
3. APP连接设备无线AP并发送用户token，发送用户token使用接口PUT
/ISAPI/System/Network/EZVIZ/physicalAccessUserToken?format=json&security=<security>&iv=<iv>;
4. 设备连接WiFi后注册到萤石云平台，上报用户token，完成注册；
5. 绑定成功，平台设置设备userid；
6. APP轮询查询设备绑定状态；
7. 平台响应设备绑定状态，设备已绑定成功，添加完成。

19.2.3 有线设备局域网SADP扫描



1. APP登录完成获取用户token和token有效时间；
2. 平台响应用户token和token有效时间
3. APP通过SADP扫描到设备并激活设备，然后通过ISAPI配置设备注册到萤石云的地址和用户token，其中配置萤石云地址使用接口PUT /ISAPI/System/Network/EZVIZ/，发送用户token使用接口PUT
/ISAPI/System/Network/EZVIZ/physicalAccessUserToken?format=json&security=<security>&iv=<iv>;
4. 设备注册到平台，上报用户token；
5. 绑定成功，平台设置设备userid；
6. APP轮询查询设备绑定状态；
7. 平台响应设备绑定状态，设备已绑定成功，添加完成。

19.2.4 有线设备物理按键接触



1. APP登录完成，搜索设备添加携带参数；
2. 平台返回客户端需要引导用户按设备物理按键；
3. 按键重置设备；
4. 获取设备token和有效时间；
5. 本地显示屏展示二维码；
6. 扫码或网络通信获取设备token，网络通信获取使用接口GET
`/ISAPI/System/Network/EZVIZ/physicalAccessVerification?format=json&security=<security>&iv=<iv>;`
7. 添加设备，携带设备序列号和设备token；
8. 返回添加结果；
9. 绑定设备，设置userid。

19.2.5 设备网页手动配置

除以上几种添加方式外，部分设备可以通过网页手动传递用户token或设备token完成设备添加。其原理与以上几种添加方式相同，差别在于用户token或设备token不是通过网络或扫码传递，而是显示在APP上或设备网页上，人工读取和输入token内容，完成整个添加过程。

设备网页配置用户Token和查看设备Token示例如下（不同设备的网页可能样式不同）。

启用

平台接入方式

萤石云

接入服务器IP

XXXXXXXXXX

☐ 自定义

注册状态

不在线

绑定状态

未知

设备绑定状态需要联网后更新，请以设备联网后更新状态为准

码流加密

码流加密

账号绑定配置

① 验证码绑定

① 扫码绑定

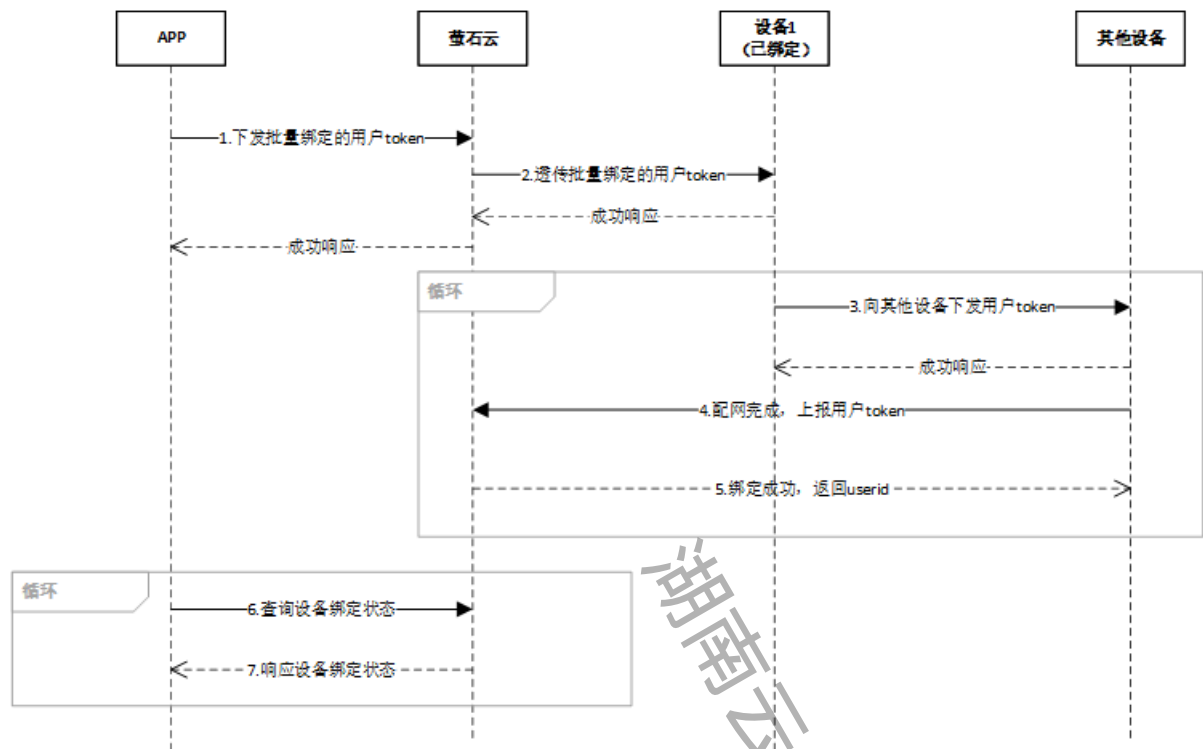
获取

保存

刷新

1. 获取萤石物理确权用户token: GET /ISAPI/System/Network/EZVIZ/physicalAccessUserToken?format=json&security=<security>&iv=<iv>
2. 配置萤石物理确权用户token: PUT /ISAPI/System/Network/EZVIZ/physicalAccessUserToken?format=json&security=<security>&iv=<iv>
3. 获取萤石物理确权设备token: GET /ISAPI/System/Network/EZVIZ/physicalAccessVerification?format=json&security=<security>&iv=<iv>

19.2.6 用户Token整网批量绑定



使用场景：使用局域网中已绑定设备去绑定同一网络中其他设备，例如：app使用一台已绑定的交换机设备，去绑定网络中其他交换机等设备。

1. APP向萤石云发送请求开始批量绑定，使用接口POST
`/ISAPI/System/Network/EZVIZ/ModifyAccessUserTokenCfgList?format=json&security=<security>&iv=<iv>;`
2. 萤石云透传批量绑定报文到已绑定的设备，使用接口同步骤1中接口；
3. 已绑定设备向其他未绑定设备下发用户Token，使用接口PUT
`/ISAPI/System/Network/EZVIZ/physicalAccessUserToken?format=json&security=<security>&iv=<iv>;`
4. 设备注册到平台，上报用户token；
5. 绑定成功，平台设置设备userid；
6. APP轮询查询设备绑定状态；
7. 平台响应设备绑定状态，所有未绑定设备已绑定，请求完成。

20 WIFI网络管理

20.1 功能介绍

用户在一些无线网络场所，将一根网络插在无线路由器上，并设置路由器的工作模式为AP模式，就可以使用到wifi网络。wifi网络管理下的设备就是作为AP使用，提供无线网络供Station站点使用。功能包括网络接口的无线服务器信息管理，以及该无线服务器下接入的设备列表信息。

20.2 集成流程

1.WIFI参数配置，包含如下：

判断设备是否支持WIFI参数，能力节点GET `/ISAPI/System/capabilities`；如果报文返回节点且节点值为true, 表示设备支持无线网，支持WIFI参数配置。

获取WIFI参数配置能力：GET `/ISAPI/System/Network/interfaces/<interfaceID>/wireless/capabilities`；

获取WIFI参数：GET `/ISAPI/System/Network/interfaces/<interfaceID>/wireless`；

配置WIFI参数：PUT `/ISAPI/System/Network/interfaces/<interfaceID>/wireless`；

获取WIFI质量参数：GET `/ISAPI/System/Network/wifiQuality?format=json`；

2.WIFI区域参数配置，包含如下：

判断设备是否支持WIFI区域参数，能力节点GET `/ISAPI/System/capabilities`；如果报文返回节点且节点值为true, 表示设备支持WIFI区域参数配置。

获取WIFI区域参数配置能力: GET /ISAPI/System/Network/wirelessServer/Region/Capabilities;

获取WIFI区域参数: GET /ISAPI/System/Network/wirelessServer/Region;

配置WIFI区域参数: PUT /ISAPI/System/Network/wirelessServer/Region;

3.WIFI连接参数，包含如下:

获取单网口WIFI连接参数能力: GET

/ISAPI/System/Network/interfaces/<interfaceID>/wirelessStatus/capabilities;

获取单网口WIFI连接参数: GET /ISAPI/System/Network/interfaces/<interfaceID>/wirelessStatus;

配置单网口WIFI连接参数: PUT /ISAPI/System/Network/interfaces/<interfaceID>/wirelessStatus;

获取单网口WIFI连接状态: GET /ISAPI/System/Network/interfaces/<interfaceID>/wireless/connectStatus;

4.WIFI传输状态，包含如下:

获取单网口WIFI传输状态能力: GET

/ISAPI/System/Network/interfaces/<interfaceID>/wireless/transmission/capabilities;

获取单网口WIFI传输状态: GET /ISAPI/System/Network/interfaces/<interfaceID>/wireless/transmission;

获取指定网口WIFI传输状态: GET

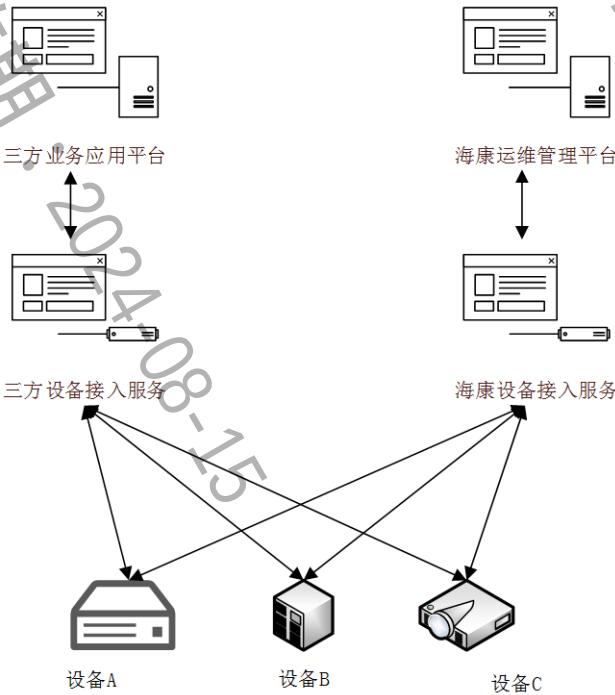
/ISAPI/System/Network/interfaces/<interfaceID>/wireless/transmission/<indexID>;

获取所有网口WIFI传输状态: GET /ISAPI/System/Network/interfaces/<interfaceID>/wireless/transmissionList;

21 ISUP多中心管理

21.1 功能介绍

海康设备在接入物联网时，存在同时接入到多个中心（注意区分中心和平台的概念，一个中心可能包含多个平台）的需求。应用场景如设备需要同时接入到三方集成商业务中心和海康运维管理中心，如下图所示。



过去海康设备接入到中心大多使用海康私有协议、ISAPI协议，这类中心主动登录设备的通信协议，我们称这种通信模式为拉模式。

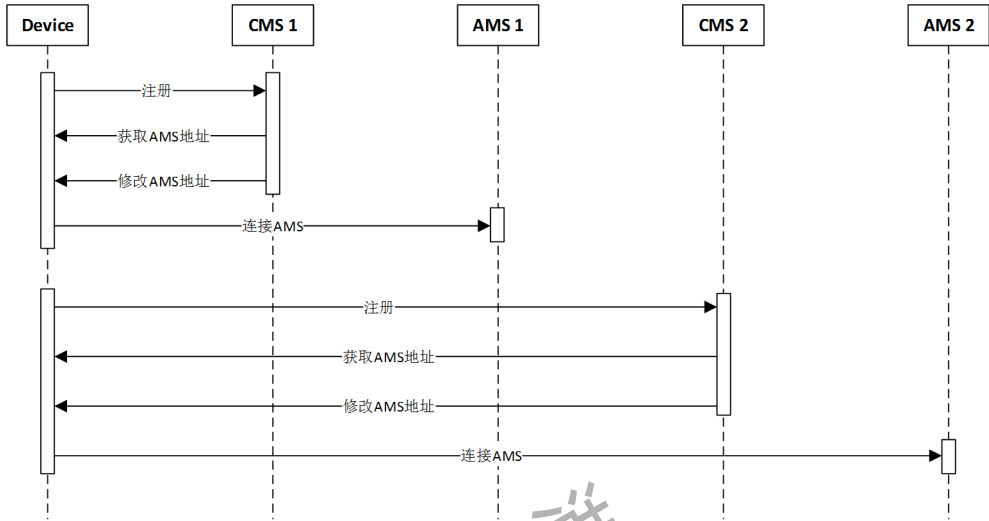
也有一部分设备以主动注册方式接入到中心，我们称这种通信模式为推模式，如：设备接入萤石云中心、设备接入ISUP。

本文主要讲述设备同时接入多个ISUP中心的集成管理流程。

21.2 集成流程

21.2.1 流程示例

海康设备ISUP多中心流程示例如下所示。



21.2.2 CMS地址配置

通过ISAPI协议对设备进行ISUP多中心CMS 1、CMS 2地址配置：

1. 判断是否支持ISUP协议：`GET /ISAPI/System/Network/capabilities`中返回isSupportEhome且为true，代表支持ISUP协议。
2. 判断是否支持多中心ISUP协议：`GET /ISAPI/System/Network/Ehome/capabilities`中返回ISUPID，代表支持多个ISUP中心。
3. 配置ISUP多中心地址：`[GET/PUT] /ISAPI/System/Network/Ehome/<ISUPID>`。

注：

URL中ISUPID表示CMS 1和CMS 2的中心ID，两个中心CMS地址需要分别配置。

21.2.3 AMS地址配置

完成两个中心CMS地址配置后，设备通过ISUP协议主动注册到CMS 1和CMS 2后，CMS 1和CMS 2分别通过ISUP协议对其配置AMS地址：

1. 获取报警地址列表：`GET /ISAPI/Event/notification/httpHosts`，通过与CMS通信链路需返回所有已配置的HTTP/HTTPS中心地址，以及对应链路的ISUP AMS中心地址，不返回其他链路的ISUP AMS中心地址。

举例：

- (1) 如设备已被配置为id=1为CMS 1的AMS 1地址（protocolType为EHome），id=2为CMS 2的AMS 2地址（protocolType为EHome），id=3为HTTP的监听中心地址（protocolType为HTTP）；
- (2) CMS 1获取报警地址列表时，设备返回：id=1的AMS 1地址（protocolType为EHome），id=3的HTTP监听中心地址（protocolType为HTTP）；
- (3) CMS 2获取报警地址列表时，设备返回：id=2的AMS 2地址（protocolType为EHome），id=3的HTTP监听中心地址（protocolType为HTTP）。

2. CMS遍历获取到的报警地址列表：

- (1) 若已有AMS地址（protocolType中有EHome类型），则调用：
`PUT /ISAPI/Event/notification/httpHosts/<ID>`对AMS地址修改；
- (2) 若没有AMS地址（protocolType中无EHome类型），则调用：
`POST /ISAPI/Event/notification/httpHosts`创建AMS地址。

3. 设备根据配置好的AMS地址连接到AMS上。

注：

- (1) 通过ISAPI协议调用：`DELETE /ISAPI/Event/notification/httpHosts`，仅删除HTTP/HTTPS中心，不对ISUP AMS中心进行任何操作；

- (2) 如通过ISUP协议调用：`PUT /ISAPI/Event/notification/httpHosts`对其进行AMS地址配置时，列表中有两个及以上的protocolType为EHome，代表配置参数错误，接口需要报错；
- (3) 如通过ISUP协议调用：`PUT /ISAPI/Event/notification/httpHosts/<ID>`，且protocolType为EHome时，无论URL中为多少，都代表配置该链路的AMS地址。

21.2.4 ISUP信息获取

通过ISAPI协议对设备进ISUP多中心CMS地址和AMS地址获取：

1. 判断是否支持ISUP信息获取：`GET /ISAPI/System/Network/Ehome/capabilities`中返回isSupportISUPServerInfo且为true，代表支持ISUP信息获取。
2. ISUP信息获取：`GET /ISAPI/System/Network/ISUPServerInfo?format=json`，获取设备已配置的CMS、AMS地址列表。

21.3 报文格式与示例

URL：`/ISAPI/System/Network/ISUPServerInfo?format=json`

Method：`GET`

报文：如下

```
{
  "ISUPServerInfoList": [
    {
      "ID": 1, //req, int, 序号, range:[1,2]
      "enabled": true, //req, boolean, 使能
      "CMSInfo": {
        "addressingFormatType": "ipaddress", //opt, string, 地址类型(ipaddress-IP地址, hostname-域名)
        "hostName": "", //opt, string, 域名 (当addressingFormatType为hostname时有效)
        "ipV4Address": "", //opt, string, IPv4地址 (当addressingFormatType为ipaddress时有效)
        "ipV6Address": "", //opt, string, IPv6地址 (当addressingFormatType为ipaddress时有效)
        "portNo": 1, //opt, integer, 端口号, range:[0,65535]
        "deviceID": "", //opt, string, 设备ID, range:[1,64]
        "protocolVersion": "v5.0", //opt, string, 协议版本 (v2.0-v2.0版本,v2.6-v2.6版本,v4.0-v4.0版本,v5.0-v5.0版本)
      },
      "AMSInfo": {
        "addressingFormatType": "ipaddress", //opt, string, 地址类型(ipaddress-IP地址, hostname-域名)
        "hostName": "", //opt, string, 域名 (当addressingFormatType为hostname时有效)
        "ipV4Address": "", //opt, string, IPv4地址 (当addressingFormatType为ipaddress时有效)
        "ipV6Address": "", //opt, string, IPv6地址 (当addressingFormatType为ipaddress时有效)
        "portNo": 1 //opt, integer, 端口号, range:[0,65535]
      }
    }
  ]
}
```

22 监听报警

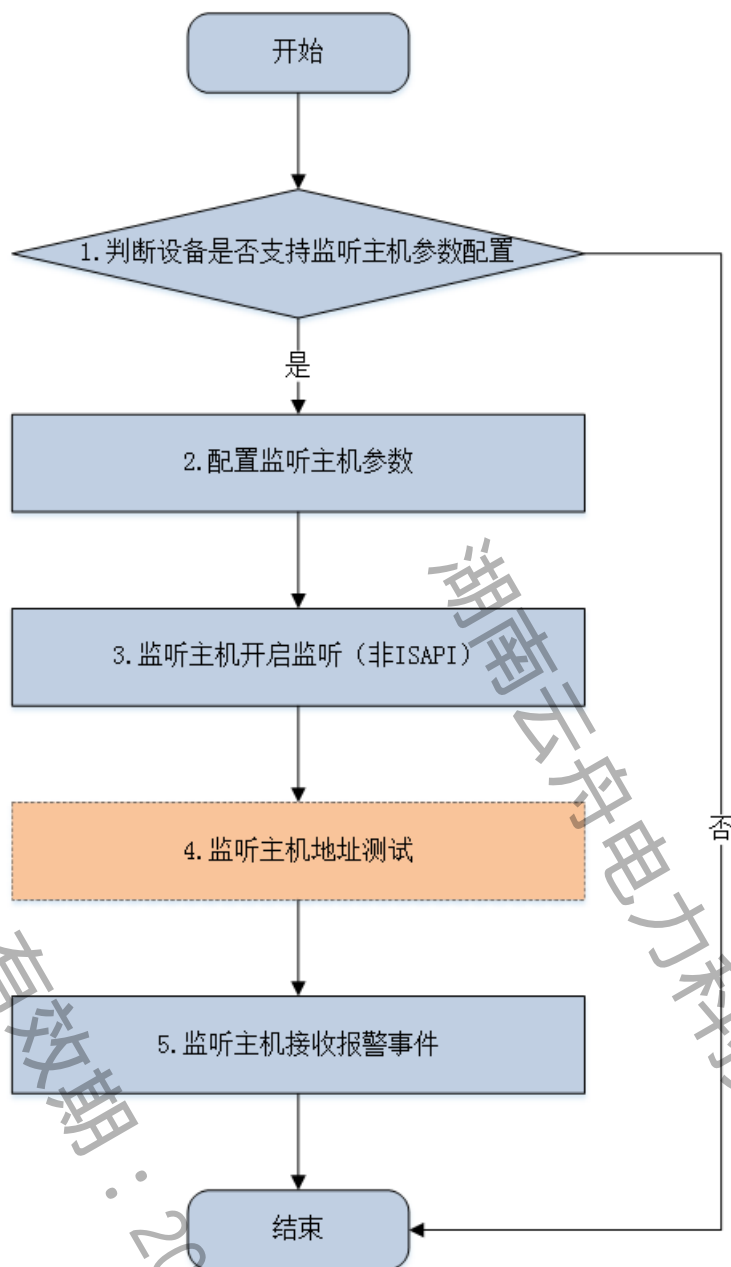
22.1 功能介绍

触发事件时设备主动连接客户端并且上传报警信息，报警接收端同时在设定IP、端口监听并接收数据。需要先在设备端配置报警接收主机的IP和端口，与报警接收端监听的IP、端口一致。

HTTP监听报警支持订阅配置，即添加和修改监听主机时配置需要订阅的报警事件，触发事件后时设备只上报订阅的报警事件。（设备未实现）

22.2 集成流程

22.2.1 监听报警



1. 判断通道是否支持监听主机参数配置：

获取监听主机参数能力 `GET /ISAPI/Event/notification/httpHosts/capabilities`，如果返回成功并且报文内容为 `<HttpHostNotificationCap>`，表示设备支持监听主机参数配置。

2. 配置监听主机参数：

配置全部监听主机参数 `PUT /ISAPI/Event/notification/httpHosts?security=<security>&iv=<iv>`;

获取全部监听主机参数 `GET /ISAPI/Event/notification/httpHosts?security=<security>&iv=<iv>`;

配置单个监听主机参数 `PUT /ISAPI/Event/notification/httpHosts/<hostID>?security=<security>&iv=<iv>`;

获取单个监听主机参数 `GET /ISAPI/Event/notification/httpHosts/<hostID>?security=<security>&iv=<iv>`;

3. 监听主机开启监听：

此步骤需要用户将监听主机开启监听服务，属于TCP/IP协议的网络编程内容，本文不做赘述。

4. （可选）监听主机测试：

平台向设备下发此命令，用于测试监听主机对于设备是否可用 `POST /ISAPI/Event/notification/httpHosts/<hostID>/test`。

5. 监听主机接收报警事件：

当触发事件时设备主动连接客户端并且上传报警信息，监听主机接收报警数据，事件消息详情见事件消息语法。

备注：可通过 /ISAPI/Event/notification/httpHosts/<hostID>/uploadCtrl 协议配置监听的超时时间等参数。

22.2.2 事件消息语法

触发监听报警事件之后，报警分为携带二进制数据（如图片）和不携带二进制数据。

1. 不携带二进制数据：

设备发送数据HTTP请求Headers中的Content-Type通常为application/xml或application/json，示例如下。

设备发送报警报文

```
POST Request_URI HTTP/1.1 <!--Request_URI,来自于URI(POST /ISAPI/Event/notification/httpHosts)的创建-->
Host: data_gateway_ip:port <!--Host: 指定设备访问的http服务器的域名/IP 地址和端口号，此处来自于URI(POST /ISAPI/Event/notification/httpHosts)的创建，根据实际情况填充：-->
Accept-Language: zh-cn
Date: YourDate
Content-Type: application/xml; <!--部分报警走json格式，所以要求上层解析的时候，根据Content-Type字段进行区分-->
Content-Length: text_length
Connection: keep-alive <!--设备端与接收服务端保持长连接，提升传输性能-->

<EventNotificationAlert/>
```

监听主机应答

```
HTTP/1.1 200 OK
Date: YourDate
Connection: close
```

2. 携带二进制数据：

设备发送数据的格式为HTTP表单格式（multipart/form-data）。HTTP请求Headers中的Content-Type通常为multipart/form-data, boundary=<frontier>, 其中boundary是一个变量，用于将整个HTTP Body分割成多个单元，每个单元为一份数据，都有各自的Headers和body。详细格式说明参考RFC 1867 (Form-based File Upload in HTML)，示例如下，请注意boundary前面和后面的横线--。

设备发送报警报文

```
POST Request_URI HTTP/1.1 <!--Request_URI,来自于URI(POST /ISAPI/Event/notification/httpHosts)的创建-->
Host: device_ip:port <!--Host: 指定设备访问的http服务器的域名/IP 地址和端口号，此处来自于URI(POST /ISAPI/Event/notification/httpHosts)的创建，根据实际情况填充：-->
Accept-Language: zh-cn
Date: YourDate
Content-Type: multipart/form-data;boundary=<frontier>
Content-Length: text_length
Connection: keep-alive <!--设备端与接收服务端保持长连接，提升传输性能-->

--<frontier>
Content-Disposition: form-data; name="Event_Type"
Content-Type: text/xml <!--部分报警走json格式，所以要求上层解析的时候，根据Content-Type字段进行区分-->

<EventNotificationAlert/>
--<frontier>
Content-Disposition: form-data; name="Picture_Name"
Content-Length: image_length
Content-Type: image/jpeg

[图片数据]
--<frontier>--
```

监听主机应答

```
HTTP/1.1 200 OK
Date: YourDate
Connection: close
```

其中部分关键字的说明如下。

关键字	示例	描述
Content-Type	multipart/form-data; boundary=frontier	内容类型，multipart/form-data表示这是一个表单格式的消息。
boundary	frontier	表单消息分隔符，--boundary 为一段表单的开始，--boundary-- 为整个HTTP表单消息的结束
Content-Disposition	form-data; name="Picture_Name";	内容描述，form-data表示是一段表单数据
filename	"Picture_Name"	文件名称，此段表单为一个文件，此文件的名称
Content-Length	10	内容长度，从下一个\r\n之后开始，到下一个 --boundary 之前的长度

22.3 异常处理

22.3.1 错误码

statusCode	statusString	subStatusCode	errorCode	errorMsg	中文描述	备注
6	Invalid Content	eventNotSupport	0x60001024		订阅事件不支持	

23 移动网络管理

23.1 功能介绍

该功能可对4G、5G等移动网络进行拨号相关配置.若支持多SIM卡设备，可同时对每个SIM卡进行拨号参数配置，以保证移动网络的正常连接。下面为web配置界面：

HIKVISION®

预览配置智能展示应用仓库

系统网络

基本配置高级配置

视音频图像PTZ事件存储

TCP/IPDDNS端口端口映射无线拨号多播配置

启用

智能网卡切换

SIM卡配置

SIM 1未连接IP地址0.0.0.0

SIM 2未连接IP地址0.0.0.0

切换卡片时需点击保存才会生效

拨号参数

拨号计划

辅助功能

拨号方式自动

网络切换方式自动

下线时间3600秒

电话

接入号码

用户名

密码

APN

MTU1400

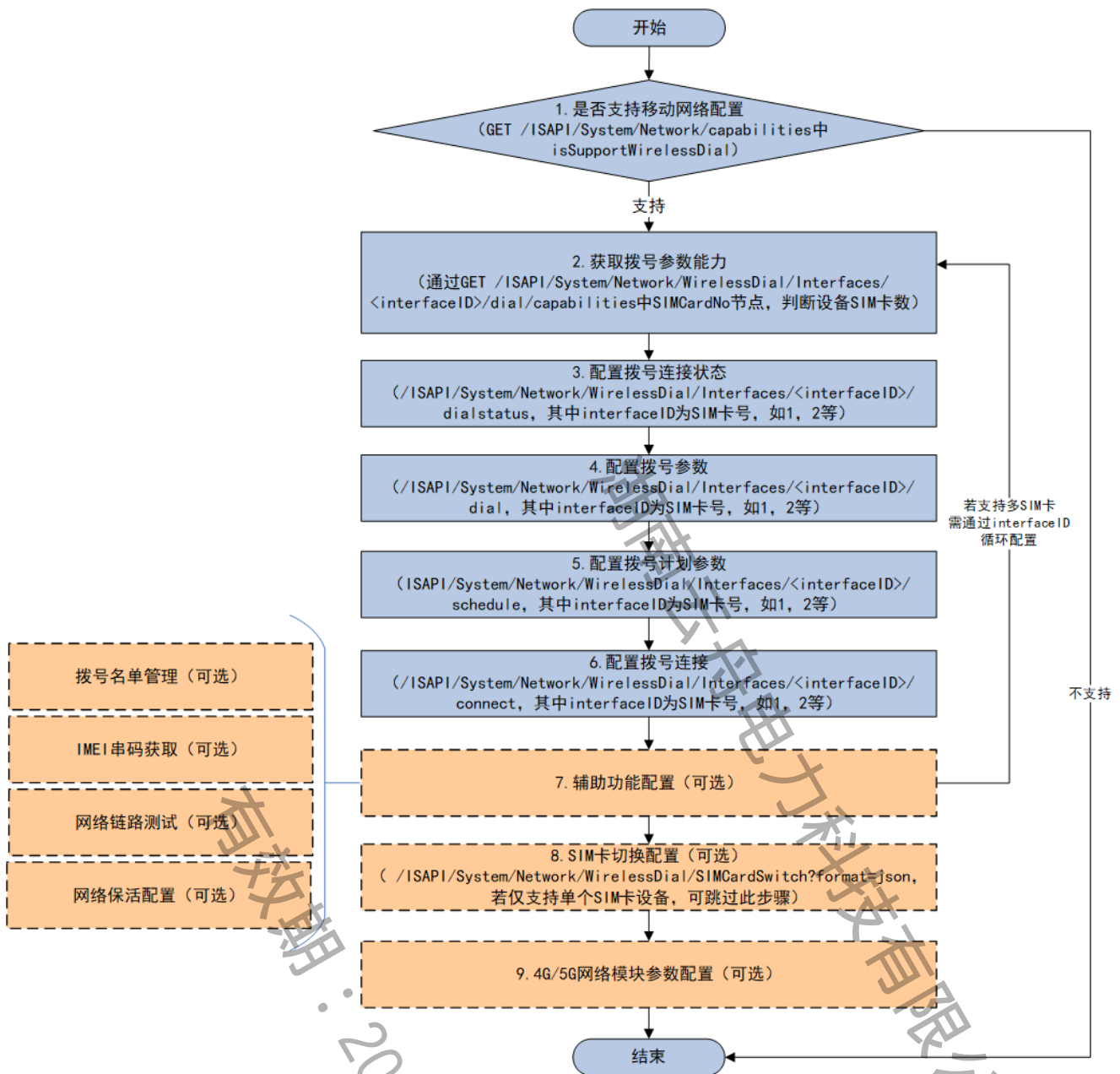
验证协议自动

保存

2024-08-15

海康威视技术有限公司

23.2 集成流程



1. 是否支持移动网络配置 获取能力:GET /ISAPI/System/Network/capabilities, 返回isSupportWirelessDial表示支持;

2. 获取拨号参数能力

第一次获取该能力时, 可通过该接口获取拨号参数能力 (URL中默认将interfaceID=1), 若返回SIMCardNo节点并SIMCardNo返回的值大于1, 则代表支持多个SIM卡, 因此流程中第2步到第6步中interfaceID需要循环配置; 例如SIMCardNo返回为2时需要, 第2步到第7步中interfaceID需要循环配置interfaceID=1和interfaceID=2的参数。

获取能力:GET /ISAPI/System/Network/WirelessDial/Interfaces/<interfaceID>/dial/capabilities

3. 配置拨号连接状态

可通过该接口配置拨号地址参数, 并可以获取当前的拨号状态情况。

获取拨号连接状态, GET /ISAPI/System/Network/WirelessDial/Interfaces/<interfaceID>/dialstatus;

配置拨号连接状态, PUT /ISAPI/System/Network/WirelessDial/Interfaces/<interfaceID>/dialstatus;

4. 配置拨号参数

可通过该接口配置拨号用户名、密码、APN名称、UIM卡序号等参数。

获取拨号连接状态, GET /ISAPI/System/Network/WirelessDial/Interfaces/<interfaceID>/dial;

配置拨号连接状态, PUT /ISAPI/System/Network/WirelessDial/Interfaces/<interfaceID>/dial;

5. 配置拨号计划参数

可通过该接口配置拨号时间周计划, 支持每周7天, 每天8个时段的配置。

获取拨号计划参数, GET /ISAPI/System/Network/WirelessDial/Interfaces/<interfaceID>/schedule;

配置拨号计划参数, PUT /ISAPI/System/Network/WirelessDial/Interfaces/<interfaceID>/schedule;

6.配置拨号连接

可通过该接口拨号链接的使能。

配置单个拨号连接, PUT /ISAPI/System/Network/WirelessDial/Interfaces/<interfaceID>/connect;

7.【可选】辅助功能配置

可按需进行辅助功能的配置, 功能如下:

拨号白名单管理:

获取单个拨号名单信息能力, GET

/ISAPI/System/Network/WirelessDial/Interfaces/<interfaceID>/messageConfig/messageConfigCap;

获取单个拨号名单信息, GET /ISAPI/System/Network/WirelessDial/Interfaces/<interfaceID>/messageConfig;

配置单个拨号名单信息, PUT /ISAPI/System/Network/WirelessDial/Interfaces/<interfaceID>/messageConfig;

IMEI串码获取:

获取设备IMEI串码, PUT /ISAPI/System/deviceInfo/IMEI?format=json;

网络链路测试:

开启网络链路测试, PUT /ISAPI/System/Network/WirelessDial/Interfaces/<interfaceID>/LinkTest/Start?
format=json;

获取网络链路测试结果, GET /ISAPI/System/Network/WirelessDial/Interfaces/<interfaceID>/LinkTest/Result?
format=json;

网络保活配置:

获取网络保活参数能力, GET

/ISAPI/System/Network/WirelessDial/Interfaces/<interfaceID>/keepAlive/capabilities?format=json;

获取网络保活参数, GET /ISAPI/System/Network/WirelessDial/Interfaces/<interfaceID>/keepAlive?
format=json;

配置网络保活参数, PUT /ISAPI/System/Network/WirelessDial/Interfaces/<interfaceID>/keepAlive?
format=json;

测试网络保活参数, POST /ISAPI/System/Network/WirelessDial/Interfaces/<interfaceID>/keepAlive/test?
format=json;

8.【可选】SIM卡切换配置

当第二步中返回SIMCardNo节点并SIMCardNo返回的值大于1时, 可按需进行SIM卡的切换配置。

获取SIM卡切换配置参数能力, GET /ISAPI/System/Network/WirelessDial/SIMCardSwitch/capabilities?
format=json;

获取SIM卡切换配置参数, GET /ISAPI/System/Network/WirelessDial/SIMCardSwitch?format=json;

配置SIM卡切换配置参数, PUT /ISAPI/System/Network/WirelessDial/SIMCardSwitch?format=json;

9.【可选】4G/5G网络模块参数配置

4G模块相关配置:

支持能力, /ISAPI/System/Network/capabilities, 返回isSupport4G表示支持;

获取4G总能力集, GET /ISAPI/System/Network/4G/capabilities?format=json,

获取4G模块参数配置能力, GET /ISAPI/System/Network/4G/mode/capabilities?format=json,

配置4G模块参数, PUT /ISAPI/System/Network/4G/mode?format=json,

获取4G模块参数, GET /ISAPI/System/Network/4G/mode?format=json,

获取4G网络状态, GET /ISAPI/System/Network/4G/status?format=json,

配置4G模块重启, PUT /ISAPI/System/Network/4G/<subModuleID>/reboot?format=json,

5G模块相关配置:

支持能力, /ISAPI/System/Network/capabilities, 返回isSupport5GModule表示支持;

获取5G模块参数配置能力, GET /ISAPI/System/Network/5GModule/capabilities?format=json,

5G模块手动鉴权, PUT /ISAPI/System/Network/5GModule/manualAuth?format=json,

获取5G模块状态, GET /ISAPI/System/Network/5GModule/info?format=json,

5G模块密码找回, PUT /ISAPI/System/Network/5GModule/resetPassword?format=json,

10.【可选】网络模块升级

获取能力, GET /ISAPI/System/Network/capabilities, 返回isSupportNetworkModuleUpgrade表示支持;

导入网络模块升级包, POST /ISAPI/System/Network/WirelessDial/ImportUpgradePackage?format=json,

获取网络模块升级包导入进度,GET /ISAPI/System/Network/WirelessDial/GetUpgradePackageImportStatus?format=json,

开始升级网络模块,POST /ISAPI/System/Network/WirelessDial/StartUpgradeNetworkModule?format=json,

获取网络模块升级状态,GET /ISAPI/System/Network/WirelessDial/GetUpgradeNetworkModuleStatus?format=json.

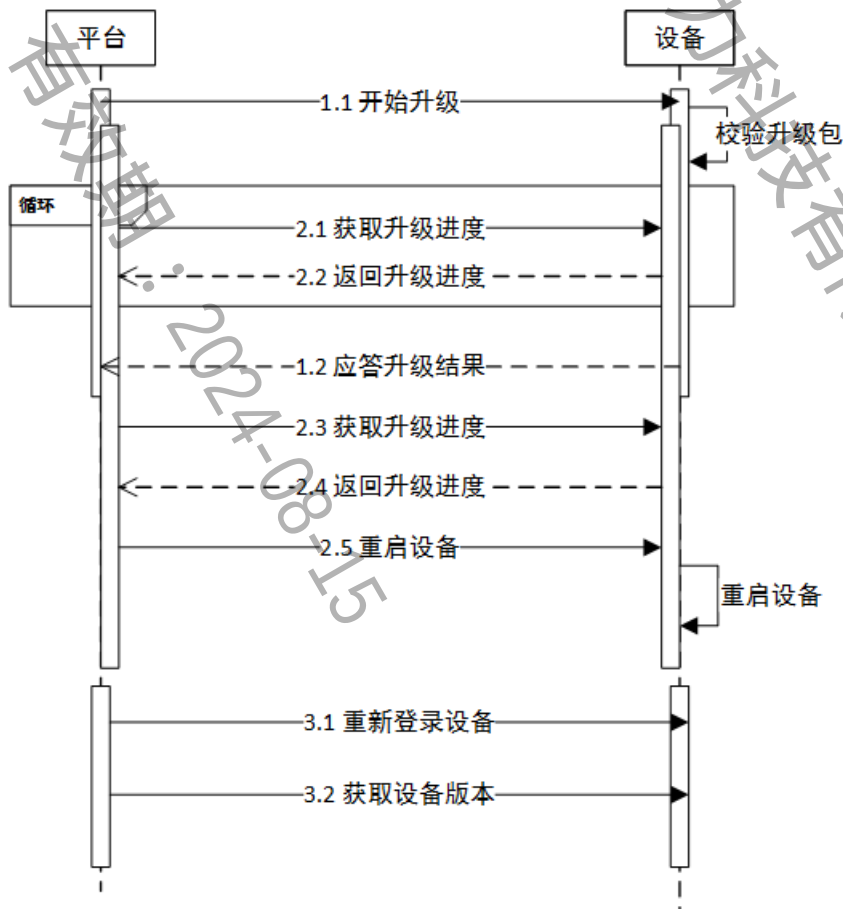
24 设备升级

24.1 功能介绍

局域网下平台/客户端/Web通过ISAPI协议，对设备进行升级。

24.2 集成流程

以平台为例，对设备进行升级的时序图如下：



- 1. 设备升级：
对设备固件升级 POST /ISAPI/System/updateFirmware;
- 2. 获取设备升级的进度：

获取设备升级的进度 GET /ISAPI/System/upgradeStatus;

3. 重启设备：

重启设备 PUT /ISAPI/System/reboot。

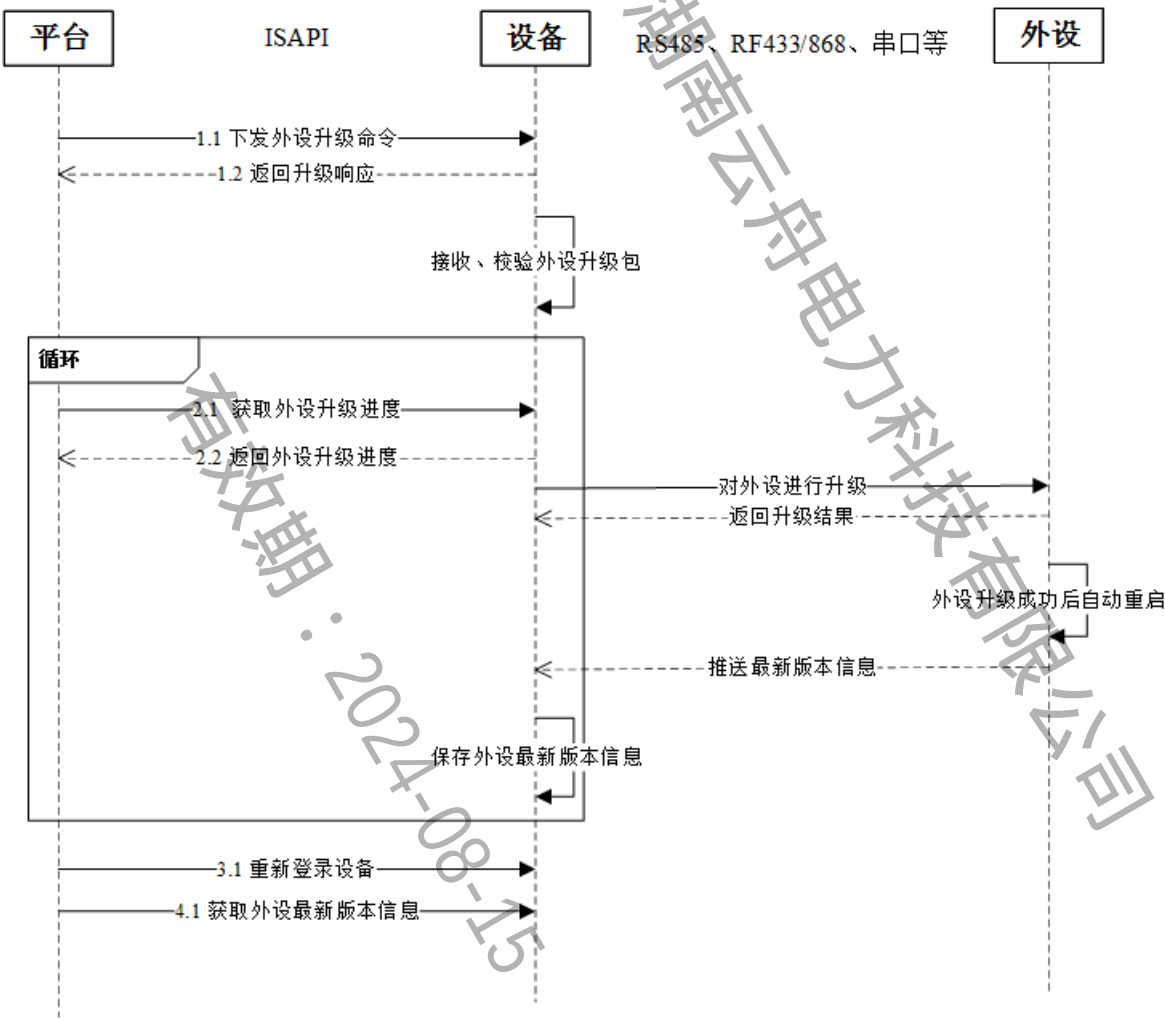
25 设备外设升级

25.1 功能介绍

局域网下平台/客户端/Web通过ISAPI协议，对设备外设进行升级。

25.2 集成流程

以平台为例，对设备外设进行升级的时序图如下：



1. 判断设备是否支持外设模块升级 GET /ISAPI/System/capabilities，isSupportAcsUpdate字段返回为true，表示支持；返回为false或者不返回，表示不支持；
2. 获取外设模块升级能力集 GET /ISAPI/System/AcsUpdate/capabilities，判断设备当前支持升级的外设类型以及编号；
3. 下发外设升级命令 POST /ISAPI/System/updateFirmware?type=<type>&moduleAddress=<moduleAddress>&id=<indexID>，其中URL参数type表示外设类型，moduleAddress表示外设的模块地址（一代混合报警主机专用），indexID表示想要进行升级的外设编号，此时平台会下发外设升级包给到设备；
4. 获取外设升级的进度 GET /ISAPI/System/upgradeStatus?type=<Type>;
5. 重新登录设备；

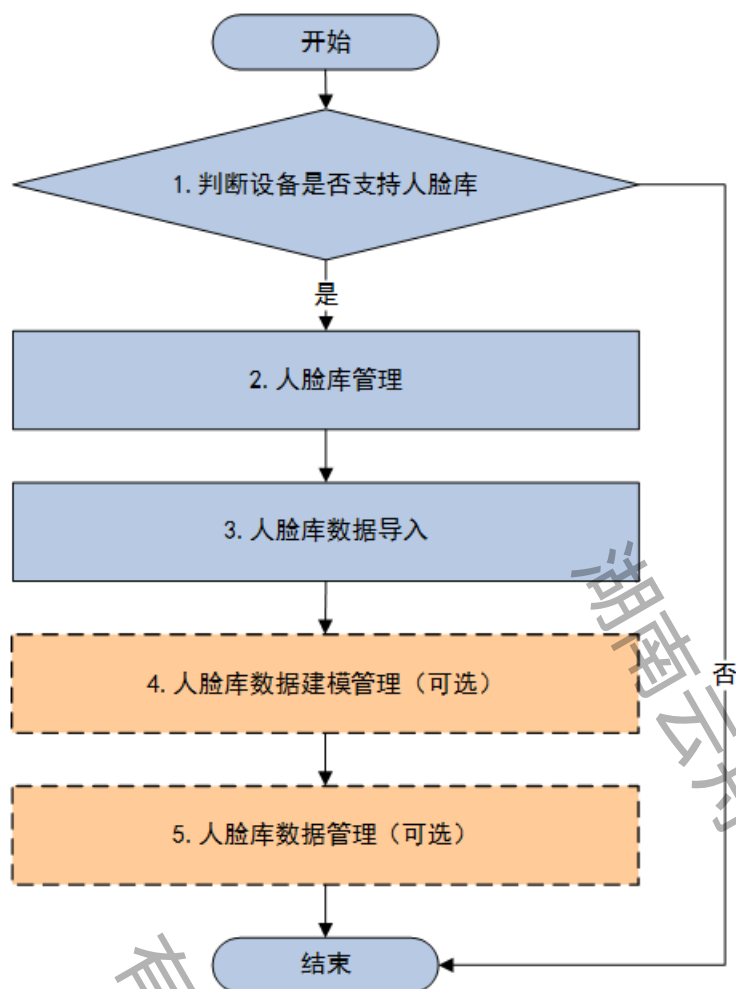
26.2 集成流程

1. 支持预览码流和回放码流应用于不同的场景，可以配置不同的转码压缩参数；
- 2) 转码码流压缩参数的配置，不影响正常视音频中的视频编码参数

27 人脸库管理

27.1 人脸库管理

27.1.2 集成流程



1. 判断设备是否支持人脸库管理：获取人脸库管理能力是否支持GET /ISAPI/Intelligent/FDLib/capabilities。
2. 人脸库管理，包含如下：创建人脸库POST /ISAPI/Intelligent/FDLib；创建成功后，设备会自动生成人脸库ID（即在出参FDLibInfoList中返回FDID），如果用户需自行管理人脸库ID，可在下发入参CreateFDLibList时，按需添加自定义人脸库IDcustomFaceLibID，后续人脸库相关操作可通过该ID替代FDID进行操作。修改人脸库参数PUT /ISAPI/Intelligent/FDLib/<FDID>；若上述步骤中已添加自定义人脸库ID,也可以通过customFaceLibID进行配置PUT /ISAPI/Intelligent/FDLib/<customFaceLibID>?FDType=custom。获取人脸库参数GET /ISAPI/Intelligent/FDLib/<FDID>；自定义人脸库ID相关操作同上。删除人脸库DELETE /ISAPI/Intelligent/FDLib/<FDID>；自定义人脸库ID相关操作同上。
3. 人脸库数据导入：人脸库数据导入并自动建模POST /ISAPI/Intelligent/FDLib/pictureUpload；用户可通过循环下发该接口将人脸数据（其中包含人脸图片及对应人脸附加信息FaceAppendData）导入至对应人脸库中；导入成功后，设备会对该图片做自动建模，但是会存在一定的耗时，建模成功后设备会自动生成人脸图片ID（即在出参MaskInfo中返回PID）；如果用户需自行管理人脸图片ID，可在下发入参PictureUploadData时，按需添加自定义人员IDcustomHumanID，后续人脸图片相关操作可通过该ID替代PID进行操作。人脸库数据导入并发处理POST /ISAPI/Intelligent/FDLib/pictureUpload?type=concurrent；如果用户想提高导入效率，可采用并发处理的方式，此时设备仅仅是保存图片，并不会对图片进行自动建模，后续需进行手动建模的方式对这些图片进行后建模（具体可查看“人脸库数据建模管理步骤”）。注：若为后端超脑设备，哪怕是并发处理情况下，设备依然会在导入数据完成后的空闲时间自动进行建模；但是若用户需要手动强制性建模，依然可以采用建模管理的方式。
4. （可选）人脸库数据建模管理：当用户在人脸库数据导入时做了并发处理操作，则在导入完成后，选择时间对设备进行数据建模管理；接口如下：控制底图统一建模GET /ISAPI/Intelligent/FDLib/manualModeling；查询离散建模控制结果GET /ISAPI/Intelligent/FDLib/manualModelingRangeTask；获取建模进度状态GET /ISAPI/Intelligent/FDLib/manualModeling/progressState。
5. 人脸库数据管理（可选）：可对人脸库进行数据查询、修改以及删除等操作，接口如下：查询人脸图片数据POST /ISAPI/Intelligent/FDLib/FDSearch；修改人脸数据PUT /ISAPI/Intelligent/FDLib/FDModify?format=json,该接口为导入图片二进制方式进行数据修改；修改人脸库中的人脸数据PUT

/ISAPI/Intelligent/FDLib/FDSearch?format=json&FDID=<FDID>&FPID=<FPID>&faceLibType=<faceLibType>,该接口为导入图片URL方式进行数据修改；删除人脸库中的人脸数据PUT
/ISAPI/Intelligent/FDLib/FDSearch/Delete?format=json&FDID=<FDID>&faceLibType=<FDType>。

27.1.3 异常处理

27.1.3.1 错误码

statusCode	statusString	subStatusCode	errorCode	errorMsg	中文描述	备注
1	OK	ok	0x00000001			
2	Device Busy	noMemory	0x20000001		设备内存不足	
4	Invalid Operation	notSupport	0x40000001		设备不支持	
4	Invalid Operation	faceLibDiskFull	0x400020F4		人脸库存储介质满	
4	Invalid Operation	faceLibraryIDNotExis	0x400020FD		人脸库不存在	
4	Invalid Operation	faceLibInitializing	0x40002136		脸库正在初始化，请稍候	
4	Invalid Operation	notMatchFacelib	0x4000a004		未匹配人脸库	
4	Invalid Operation	existSameNameFacelib	0x4000a00a		存在同名人脸库	
4	Invalid Operation	delFacelibFailed	0x4000a00b		删除人脸库失败	
4	Invalid Operation	modifyFacelibFailed	0x4000a00c		修改人脸库失败	
4	Invalid Operation	createFacelibFailed	0x4000a00d		创建人脸库失败	
4	Invalid Operation	facelibAlreadyUsed	0x4000a010		该人脸库已被使用	
6	Invalid Content	faceLibraryDatabaseError	0x60000032		人脸库中数据库版本不匹配	
6	Invalid Content	faceLibraryError	0x60000034		人脸库数据错误	
6	Invalid Content	multipleFaceObjectError	0x60000035		多张人脸目标错误（导入人脸库的图片中存在多个有效目标,且没有坐标区分）	
6	Invalid Content	faceLibPicImportingError	0x6000003A		人脸库导入图片中	
6	Invalid Content	faceLibraryNoData	0x60000069		人脸库搜索无数据	

6	Invalid Content	NoDataForModelingError	0x6000006B		人脸库无待建模数据
6	Invalid Content	customFaceLibIDRepeat	0x6000006C		上层下发重复的自定义人脸库ID
6	Invalid Content	customFaceDataDeleteing	0x600000A1		原自定义人脸库数据正在删除中，上层无法重复创建该ID的自定义人脸库
6	Invalid Content	analysisEnginesFacelibImporting	0x60001005		分析引擎正在导入人脸库
6	Invalid Content	facelibDataProcessing	0x6000100b		人脸库数据正在处理
6	Invalid Content	personInfoExtendValueIsTooLong	0x60001083		人脸库自定义标签长度超限
6	Invalid Content	personInfoExtendValueContainsInvalidChar	0x60001084		人脸库自定义标签含有非法字符

27.1.4 消息示例

人脸图片二进制方式导入至人脸库并自动建模POST /ISAPI/Intelligent/FDLib/pictureUpload 示例如下。

```
POST http://10.16.96.29/ISAPI/Intelligent/FDLib/pictureUpload HTTP/1.1
If-Modified-Since: 0
Accept: */*
X-Requested-With: XMLHttpRequest
Content-Type: multipart/form-data; boundary=-----7e27010d1dba
Referer: http://10.16.96.29/doc/page/config.asp
Accept-Language: zh-CN
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/5.0 (compatible; MSIE 10.0; Windows NT 6.1; WOW64; Trident/6.0)
Connection: Keep-Alive
Content-Length: 903330
DNT: 1
Host: 10.16.96.29
Pragma: no-cache
Cookie: language=zh; WebSession=48b693ca2b7301e3119d; sdMarkMenu=14%3AfaceSnapLib; sdMarkTab_1_0=0%3AsettingBasic

-----7e27010d1dba
Content-Disposition: form-data; name="PictureUploadData"

<?xml version='1.0' encoding='UTF-8'?>
<<PictureUploadData><FDID>55869AFE1E840958120C97054720372</FDID><FaceAppendData><name>20171222103007395_2</name></FaceAppendData></PictureUploadData>
-----7e27010d1dba
Content-Disposition: form-data; name="importImage"; filename="20171222103007395_2.jpeg"
Content-Type: image/jpeg

【图片二进制数据】
-----7e27010d1dba--
```

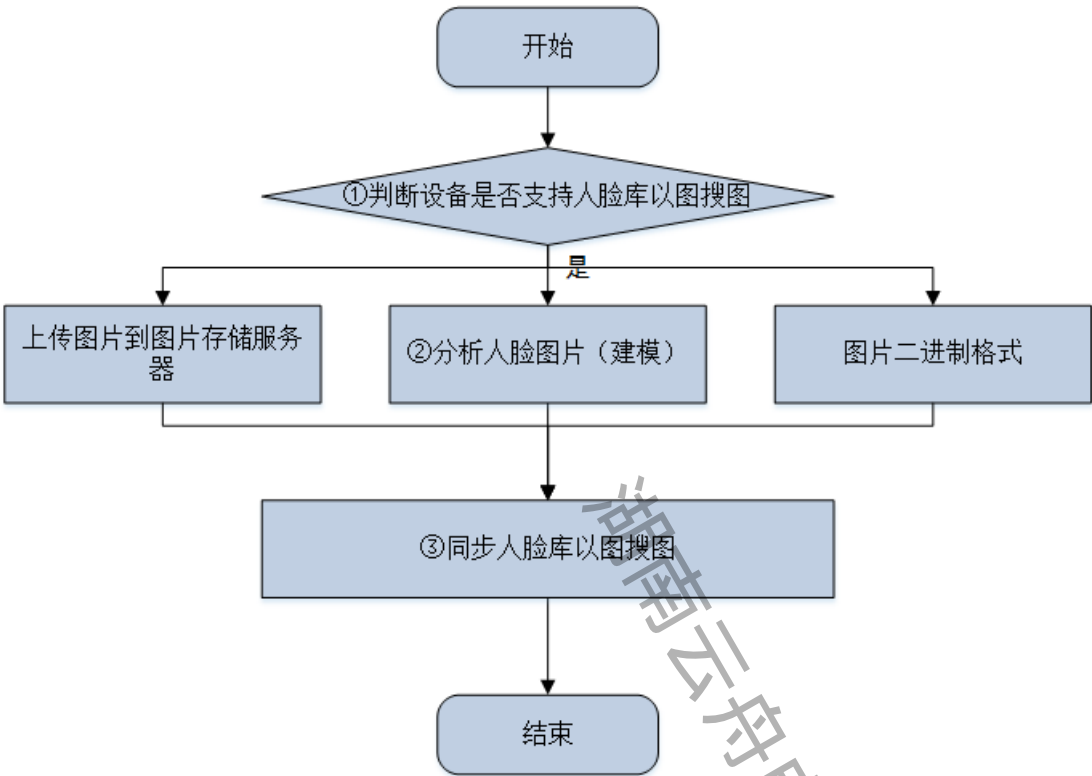
28 人脸库管理

28.1 人脸库以图搜图

28.2.1 功能介绍

人脸库以图搜图，即根据图片搜索人脸库里面符合条件的图片数据。
对于数据中心产品，同步人脸库以图搜图功能查找人脸库中与查询的图片相似的人，主要用于做身份确认。

28.1.2 集成流程



1. 判断设备是否支持同步以图搜图

获取以图搜图能力集GET /ISAPI/Intelligent/FDLib/capabilities?format=json; 如果报文返回isSuportFSSearchByPic为true表示设备支持同步以图搜图。

注：部分支持人脸库以图搜图的设备没有实现获取以图搜图能力集接口，用户直接调用人脸库以图搜图接口POST /ISAPI/Intelligent/FDLib/searchByPic?format=json

2. 搜索图片支持三种方式

URL方式：

需要有图片存储服务，将图片上传后得到图片URL，后续可以通过URL下载图片文件；

模型数据：

获取分析人脸图片能力集 GET /ISAPI/SDT/Face/pictureAnalysis/capabilities;

分析人脸图片（建模） POSTT /ISAPI/SDT/Face/pictureAnalysis，设备建模后返回图片建模数据targetModelData;

二进制图片：

直接读取图片文件，以二进制数据形式。

3. 提交同步人脸库以图搜图

同步人脸库以图搜图 POST /ISAPI/Intelligent/FDLib/searchByPic?format=json;

注：输入参数节点dataType表示三种方式，图片URL对应输入参数节点faceURL，模型数据对应输入参数节点targetModelData，二进制数据方式在输入报文为表单形式，JSON数据后带有二进制图片数据。

二进制图片请求报文示例

```
POST /ISAPI/Intelligent/FDLib/searchByPic?format=json
Host: device_ip:port
Accept-Language: zh-cn
Date: YourDate
Content-Type: multipart/form-data;boundary=<frontier>
Content-Length: text_length
Connection: keep-alive
```

```
--<frontier>
Content-Disposition: form-data; name=""
Content-Type: application/json
```

```
[JSON报文]
--<frontier>
Content-Disposition: form-data; name="Picture_Name"
Content-Length: image_length
Content-Type: image/jpeg
```

```
[图片数据]
--<frontier>--
```

29 实时音频取流集成说明

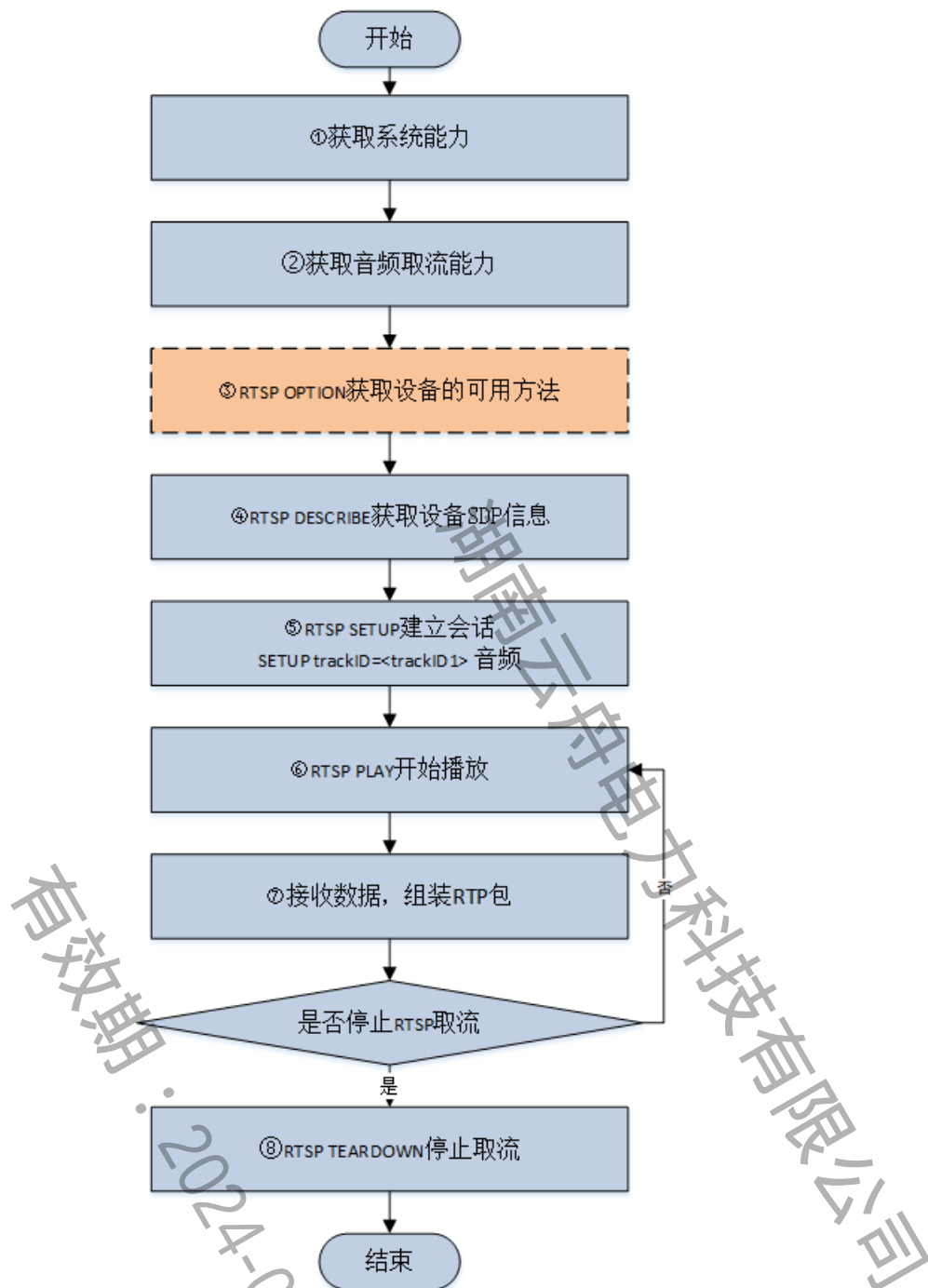
29.1 功能介绍

客户端通过RTSP协议获取设备采集的实时音频数据，实现监听设备端的情况。

29.2 集成流程

有效期：2024-08-15

湖南云舟电力科技有限公司



1. 客户端获取系统能力，GET /ISAPI/System/capabilities，返回节点isSupportRTSPAudio表示设备支持RTSP实时音频取流;
2. 客户端获取音频能力，GET /ISAPI/System/Audio/capabilities,返回节点isSupportRTSPAudio表示设备支持RTSP实时音频取流;
3. （可选）客户端发送RTSP OPTION命令，OPTION rtsp://IP地址/ISAPI/Streaming/audio/<audioID> RTSP/1.0，获取设备支持的RTSP方法;
4. 客户端发送RTSP DESCRIBE命令，DESCRIBE rtsp://IP地址/ISAPI/Streaming/audio/<audioID> RTSP/1.0，在此步骤交互时和设备完成MD5摘要认证;
5. 客户端利用从SDP信息中解析出的trackID，设置RTSP播放参数，trackID=1表示音频;
6. 客户端发送RTSP PLAY命令，设备开始推送音频流，格式为PLAY rtsp://IP地址/ISAPI/Streaming/audio/<audioID> RTSP/1.0;
7. 客户端接收设备发送的RTP包，需要注意RTP有分包的情况，在客户端组装完成后再进行解析;
8. 客户端发送RTSP TEARDOWN命令停止取流.

注释：

- <audioID>表示音频通道号


```
SETUP rtsp://10.21.84.147:554/ISAPI/Streaming/audio/1 trackID=1 RTSP/1.0
CSeq:3
Authorization: Digest username="admin", realm="3521781c29acb312330dd668", nonce="026019333", uri="rtsp://10.21.84.147:554/ISAPI/Streaming/audio/1/",
response="ff343f5ff82deb028dd9b4932cc44201"
Transport:RTP/AVP/TCP;unicast;interleaved=0-1;ssrc=0
User-Agent:NKPlayer-1.00.00.081112
```

设备响应

```
RTSP/1.0 200 OK
Session: 1127293610;timeout=60
Transport: RTP/AVP/TCP;unicast;interleaved=0-1;ssrc=433122aa
CSeq: 3
Accept-Ranges: NPT
Media-Properties: No-Seeking, Time-Progressing, Time-Duration=0
Date: Tue, 17 Nov 2020 02:09:45 GMT
```

29.3.4 客户端发送RTSP PLAY请求

```
PLAY rtsp://10.21.84.147:554/ISAPI/Streaming/audio/1 RTSP/1.0
CSeq:4
Authorization: Digest username="admin", realm="3521781c29acb312330dd668", nonce="026019333", uri="rtsp://10.21.84.147:554/ISAPI/Streaming/audio/1/",
response="24edf8a6ff3ef767f7c49d1c847200bd"
Session:1127293610;timeout=60
Range:npt=0.000000-0.000000
User-Agent:NKPlayer-1.00.00.081112
```

服务端推送音频流数据

```
RTSP/1.0 200 OK
Session: 1127293610
CSeq: 4
Date: Tue, 17 Nov 2020 02:09:45 GMT

$. ....d1.w....c....".T....g....)i.....a....7.S...~J.....X....X.
```

29.3.5 客户端发送RTSP TEARDOWN请求

```
TEARDOWN rtsp://10.21.84.147:554/ISAPI/Streaming/audio/1 RTSP/1.0
CSeq:5
Authorization: Digest username="admin", realm="3521781c29acb312330dd668", nonce="026019333", uri="rtsp://10.21.84.147:554/ISAPI/Streaming/audio/1/",
response="24edf8a6ff3ef767f7c49d1c847200bd"
Session:1127293610;timeout=60
Range:npt=0.000000-0.000000
User-Agent:NKPlayer-1.00.00.081112
```

设备响应

```
RTSP/1.0 200 OK
Session: 1127293610
CSeq: 5
Date: Tue, 17 Nov 2020 02:09:50 GMT
```

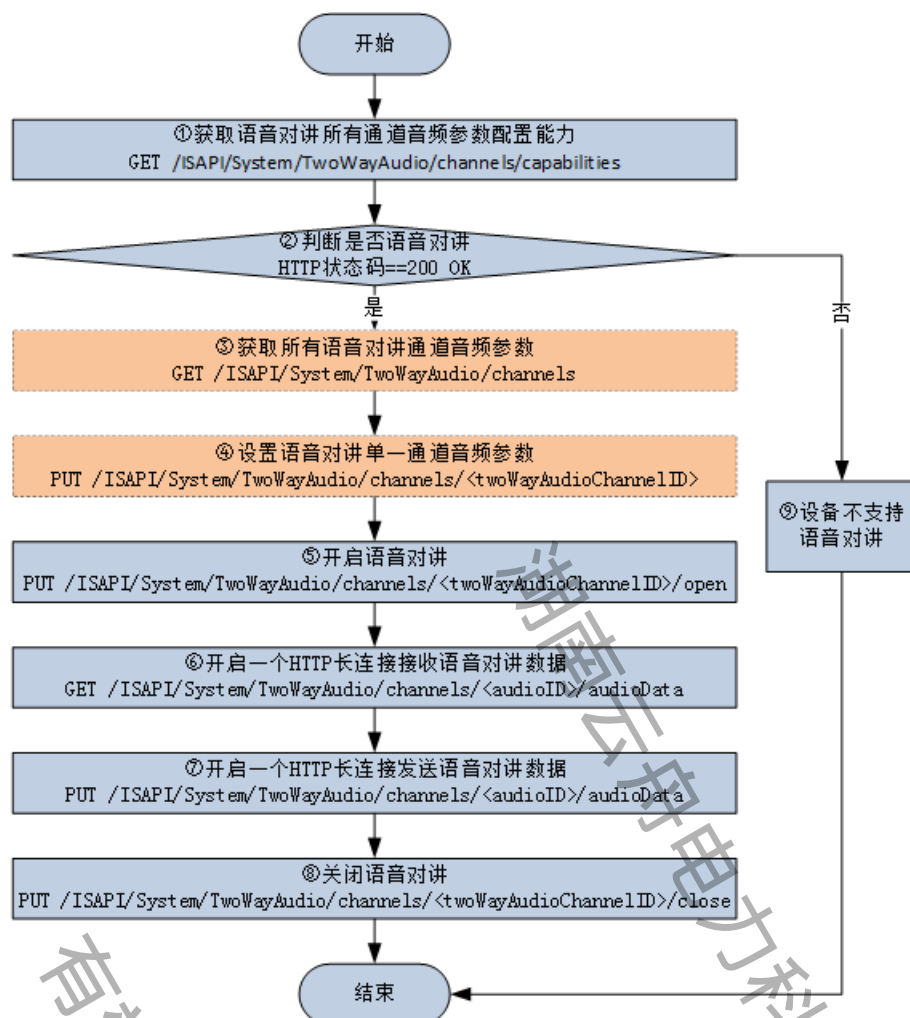
30 语音对讲

30.1 功能介绍

实现客户端与设备的双向音频对讲功能。在使用ISAPI实现语音对讲功能时，客户端需要实现本地音频的采集与编码、设备音频数据的解码。

注：在开始语音对讲之前，需要确认设备音频输入采集方式、音频编码格式和音量等参数是否正确，客户端发送的音频编码格式要和设备音频编码格式一致。

30.2 集成流程



1. 获取语音对讲所有通道音频参数配置能力GET /ISAPI/System/TwoWayAudio/channels/capabilities;
2. 判断是否语音对讲，HTTP状态码==200 OK;
3. 获取所有语音对讲通道音频参数GET /ISAPI/System/TwoWayAudio/channels;
4. 设置语音对讲单一通道音频参数PUT /ISAPI/System/TwoWayAudio/channels/<twoWayAudioChannelID>;
5. 开启语音对讲PUT /ISAPI/System/TwoWayAudio/channels/<twoWayAudioChannelID>/open，开启成功后设备返回sessionId;
6. 开启一个HTTP长连接接收语音对讲数据GET /ISAPI/System/TwoWayAudio/channels/<audioID>/audioData;
7. 开启一个HTTP长连接发送语音对讲数据PUT /ISAPI/System/TwoWayAudio/channels/<audioID>/audioData;
8. 对接结束时客户端关闭步骤6和步骤7的长连接，然后给设备发送消息：关闭语音对讲PUT /ISAPI/System/TwoWayAudio/channels/<twoWayAudioChannelID>/close;
9. 设备不支持语音对讲;

注：

- 步骤4、步骤5和步骤8中的<twoWayAudioChannelID>为设备音频对讲通道号，从1开始。可以从步骤1设备响应消息中的<id>字段解析出。
- 步骤6和步骤7使用长连接交互，请求的HTTP Headers中不要设置Content-Length，需要设置Connection: keep-alive和Content-Type: application/octet-stream。

30.3异常处理

错误码：

statusCode	statusString	subStatusCode	errorCode	errorMsg	中文描述	备注
4	Invalid Operation	twoWayAudioInProgressPleaseWait	0x40002068	Two-way audio in progress...Please wait.	正在语音对讲,请稍后	语音对讲正在进行中,不允许当前操作,请等待语音对讲停止,再进行尝试。

30. 4报文格式与示例

30.4.1 接收语音对讲数据

```
GET /ISAPI/System/TwoWayAudio/channels/1/audioData HTTP/1.1
Host: 10.17.115.128
Connection: keep-alive
Content-Type: application/octet-stream

HTTP/1.1 200 OK
Connection: keep-alive
Content-Type: application/octet-stream

xxxxxxxx设备发送的语音对讲数据xxxxxxxx
```

30.4.2 发送语音对讲数据

```
PUT /ISAPI/System/TwoWayAudio/channels/1/audioData HTTP/1.1
Host: 10.17.115.128
Connection: keep-alive
Content-Type: application/octet-stream

xxxxxxxx客户端发送的语音对讲数据xxxxxxxx
```

31 身份证非授权名单管理

31.1 功能介绍

设备支持导入、清空身份证非授权名单，以及身份证非授权名单核验，用于满足用户认证时核验其是否为身份证非授权名单中的一员，满足市场多样化安全性的应用。如访客在酒店进行入住登记时，需要核验该访客是否为国家通缉人员或具有违法犯罪的案例。

31.2 集成流程

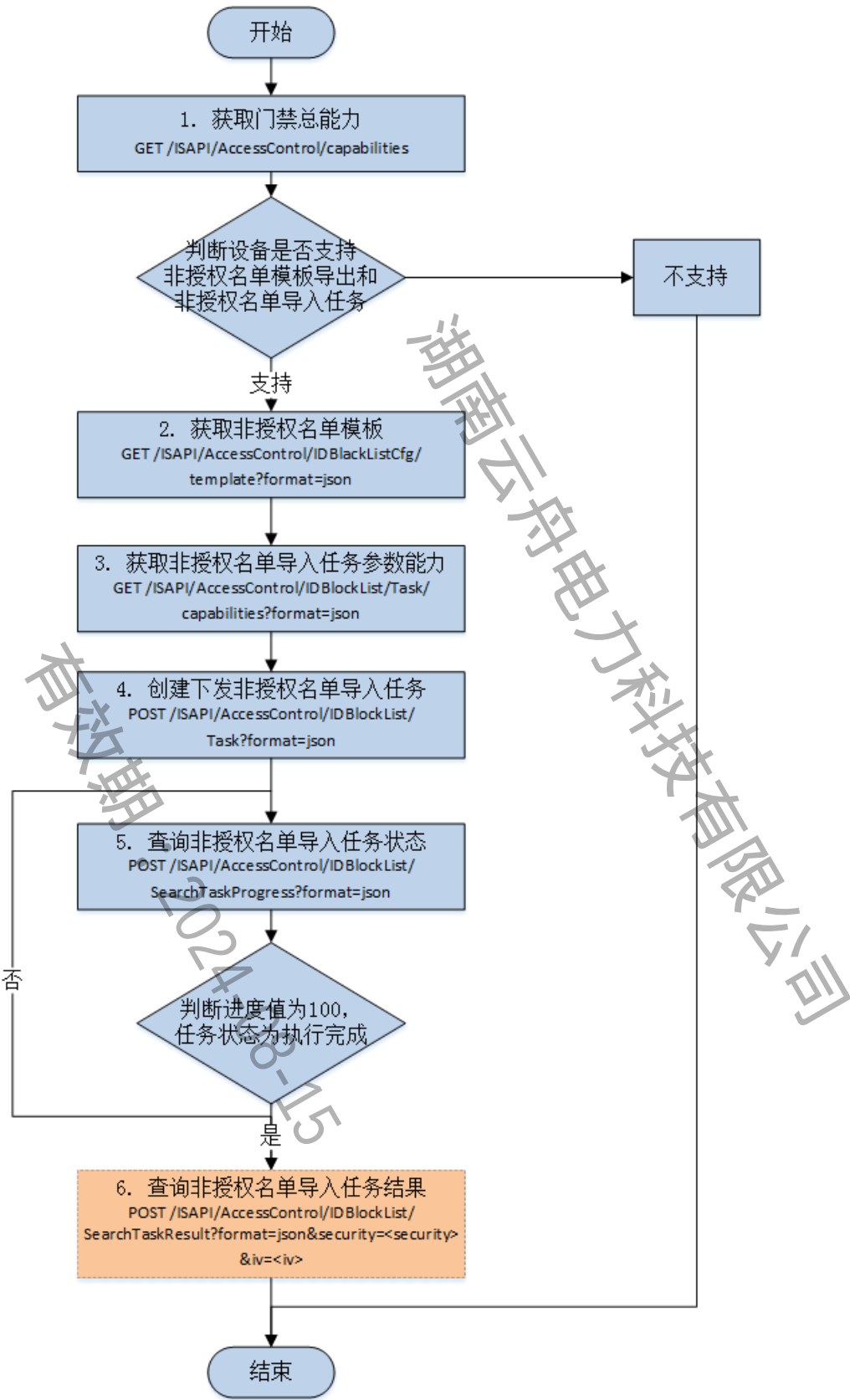
31.2.1 身份证非授权名单下发

31.2.1.1 单个方式下发

- 获取门禁总能力GET /ISAPI/AccessControl/capabilities，通过isSupportIDBlackListTemplate、isSupportIDBlackListCfg字段返回为true，判断设备支持身份证非授权名单模板导出和身份证非授权名单下发功能；
- 调用GET /ISAPI/AccessControl/IDBlackListCfg/template?format=json，获取身份证非授权名单模板；
- 调用GET /ISAPI/AccessControl/IDBlackListCfg/capabilities，获取身份证非授权名单下发功能设备支持的详细参数规格；

4. 循环调用PUT /ISAPI/AccessControl/IDBlackListCfg?security=<security>&iv=<iv>, 下发身份证非授权名单, 对于姓名、住址、身份证号码、民族等敏感信息需要加密传输。

31.2.1.2 任务方式下发



1. 获取门禁总能力GET /ISAPI/AccessControl/capabilities, 通过isSupportIDBlackListTemplate、isSupportIDBlockListTask字段返回为true, 判断设备支持身份证非授权名单模板导出和身份证非授权名单导入任务功能;
2. 调用GET /ISAPI/AccessControl/IDBlackListCfg/template?format=json, 获取身份证非授权名单模板;
3. 调用GET /ISAPI/AccessControl/IDBlockList/Task/capabilities?format=json, 获取身份证非授权名单导入任务功能设备支持的详细参数规格, 包含是否支持创建任务、查询任务状态、查询任务结果等;

4. 调用POST /ISAPI/AccessControl/IDBlockList/Task?format=json，创建下发身份证非授权名单任务；
5. 循环调用POST /ISAPI/AccessControl/IDBlockList/SearchTaskProgress?format=json，查询身份证非授权名单导入任务状态，设备会返回任务进度和任务状态；
6. 待设备任务进度值返回100，任务状态返回执行完成，可调用POST /ISAPI/AccessControl/IDBlockList/SearchTaskResult?format=json&security=<security>&iv=<iv>，查询身份证非授权名单导入任务结果，对于姓名、住址、身份证号码、民族等敏感信息需要加密传输。

31.2.2 身份证非授权名单清空

1. 获取门禁总能力GET /ISAPI/AccessControl/capabilities，通过isSupportIDBlackListClear字段返回为true，判断设备支持身份证非授权名单清空功能；
2. 调用PUT /ISAPI/AccessControl/IDBlackListClear?format=json，清空身份证非授权名单。

31.2.3 身份证非授权名单参数配置

1. 获取门禁总能力GET /ISAPI/AccessControl/capabilities，通过isSupportIDBlackListParamCfg字段返回为true，判断设备支持身份证非授权名单参数配置功能；
2. 调用GET /ISAPI/AccessControl/IDBlackListParamCfg/capabilities?format=json，获取身份证非授权名单参数配置功能设备支持的详细参数规格；
3. 调用GET|PUT /ISAPI/AccessControl/IDBlackListParamCfg?format=json，进行身份证非授权名单的参数获取和配置。

32 身份证授权名单管理

32.1 功能描述

设备支持导入、清空身份证授权名单，以及授权名单核验，用于满足用户认证时核验其是否为身份证授权名单中的一员，满足市场多样化安全性的应用。如公务员考试场景下，每个考场固定分配考生名单，考生来考场签到时，需要核验其是否为本考场的考生。

32.2 集成流程

32.2.1 身份证授权名单下发

1. 获取门禁总能力GET /ISAPI/AccessControl/capabilities，通过isSupportExportIDAllowListTemplateByURL、isSupportExportIDAllowListTemplateByBinary、isSupportIDAllowListTask字段返回为true，判断设备支持身份证授权名单模板导出（URL方式）、身份证授权名单模板导出（二进制方式）和身份证授权名单导入任务功能；
2. 调用GET /ISAPI/AccessControl/IDAllowList/ExportTemplateByBinary?format=json，获取身份证授权名单模板（二进制方式）；或者，调用GET /ISAPI/AccessControl/IDAllowList/ExportTemplateByURL/capabilities?format=json，获取身份证授权名单模板（URL方式）设备支持的详细参数规格，再调用GET /ISAPI/AccessControl/IDAllowList/ExportTemplateByURL?format=json，获取身份证授权名单模板（URL方式）；
3. 调用GET /ISAPI/AccessControl/IDAllowList/Task/capabilities?format=json，获取身份证授权名单导入任务功能设备支持的详细参数规格，包含是否支持创建任务、查询任务状态、查询任务结果等；
4. 调用POST /ISAPI/AccessControl/IDAllowList/Task?format=json，创建下发身份证授权名单任务；
5. 循环调用POST /ISAPI/AccessControl/IDAllowList/SearchTaskProgress?format=json，查询身份证授权名单导入任务状态，设备会返回任务进度和任务状态；
6. 待设备任务进度值返回100，任务状态返回执行完成，可调用POST /ISAPI/AccessControl/IDAllowList/SearchTaskResult?format=json&security=<security>&iv=<iv>，查询身份证授权名单导入任务结果，对于姓名、住址、身份证号码、民族等敏感信息需要加密传输。注意：身份证授权名单下发流程与身份证非授权名单下发（任务方式）流程相同。

32.2.2 身份证授权名单清空

1. 获取门禁总能力GET /ISAPI/AccessControl/capabilities，通过isSupportIDAllowListClear字段返回为true，判

断设备支持身份证授权名单清空功能;

2. 调用PUT /ISAPI/AccessControl/IDAllowListClear?format=json, 清空身份证授权名单。

32.2.3 身份证授权名单参数配置

- 1. 获取门禁总能力GET /ISAPI/AccessControl/capabilities, 通过isSupportIDAllowListParamCfg字段返回为true, 判断设备支持身份证授权名单参数配置功能;
- 2. 调用GET /ISAPI/AccessControl/IDAllowList/ParamCfg/capabilities?format=json, 获取身份证授权名单参数配置功能设备支持的详细参数规格;
- 3. 调用GET|PUT /ISAPI/AccessControl/IDAllowList/ParamCfg?format=json, 进行身份证授权名单的参数获取和配置。

33 身份证软解

33.1 功能描述

使能身份证软解后, 设备可以通过软件解码读取到身份证号、图片等信息; 未使能时设备只能读取到卡片序列号。

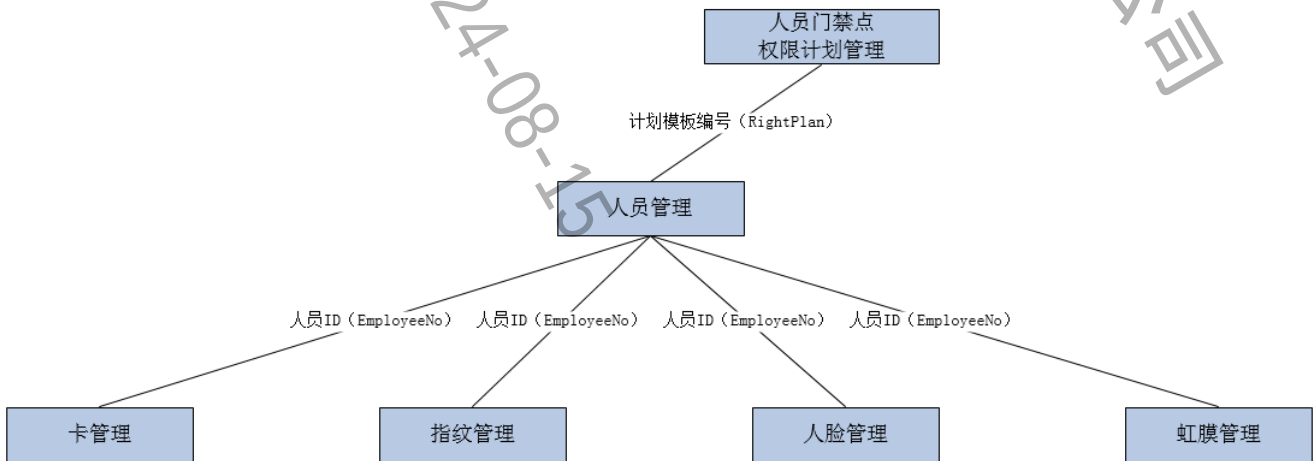
33.2 集成流程

- 1. 获取门禁总能力GET /ISAPI/AccessControl/capabilities, 通过isSupportIDCardSoftSolutionParams字段返回为true, 判断设备支持身份证软解功能;
- 2. 调用GET /ISAPI/AccessControl/IDCardMgr/IDCardSoftSolutionParams/capabilities?format=json, 获取身份证软解功能设备支持的详细参数规格;
- 3. 调用GET|PUT /ISAPI/AccessControl/IDCardMgr/IDCardSoftSolutionParams?format=json, 进行身份证软解的参数获取和配置。

34 人员及凭证管理

34.1 功能介绍

人员及权限管理功能采用以人中心架构, 用于进行人员管理、凭证管理(卡管理、指纹管理、人脸管理、虹膜管理)、人员门禁点权限计划管理, 来控制人员对受控区域内的进出权限, 其整体架构图如下所示。



本文主要讲述人员管理、凭证管理(卡管理、指纹管理、人脸管理、虹膜管理)的集成流程, 人员门禁点权限计划管理集成流程见"人员门禁点权限计划管理"。

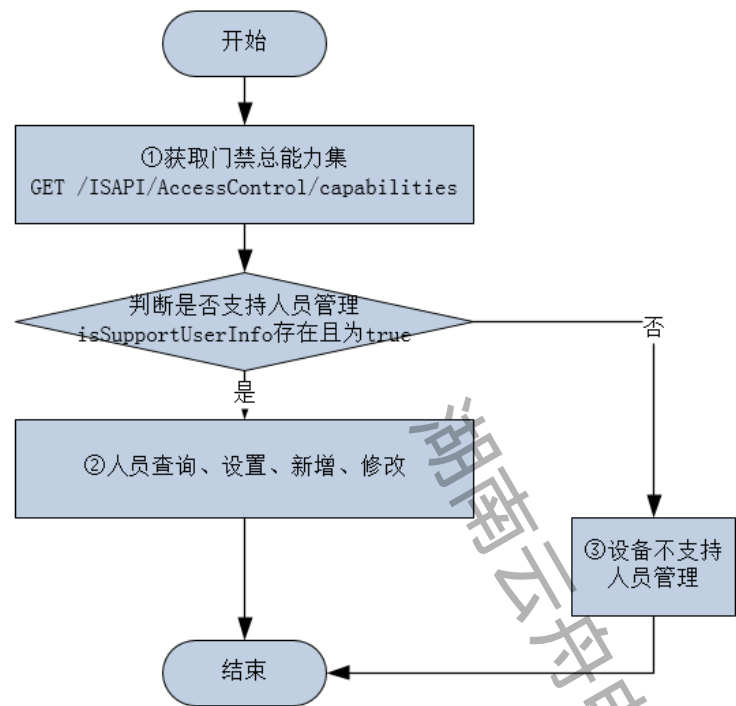
35 人员管理

35.1 功能介绍

人员管理包含人员的查询、设置、新增、修改、删除, 下面针对功能进行详细描述。

35.2 集成流程

35.2.1 判断是否支持人员管理



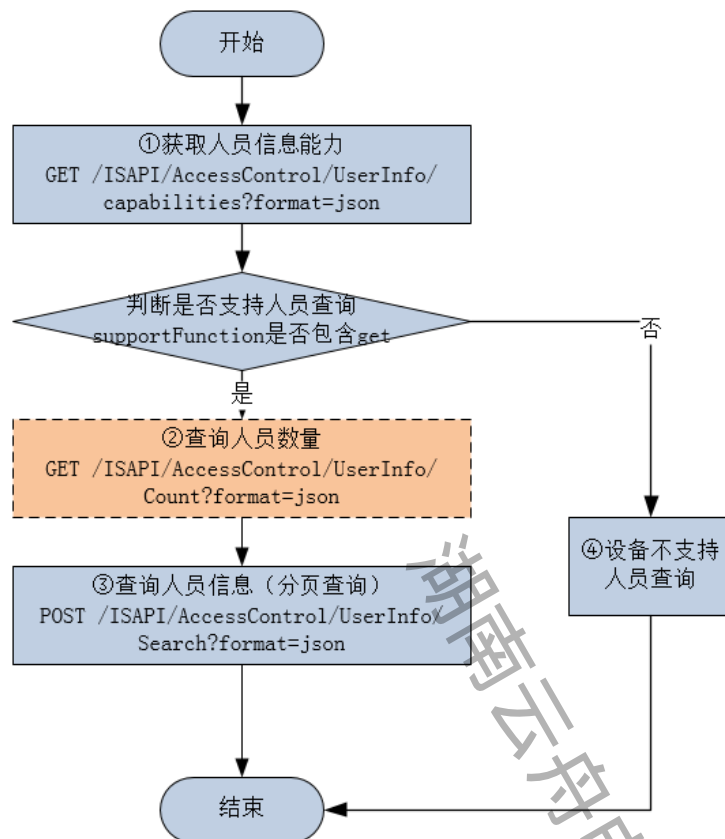
在集成人员管理功能前，首先要确保设备支持人员管理：

1. 判断是否支持人员管理：GET /ISAPI/AccessControl/capabilities 中返回isSupportUserInfo且为true，代表支持人员管理。
2. 进行人员查询、设置、新增、修改。
3. 设备不支持人员管理。

注：

（1）人员ID（EmployeeNo）作为管理人员及凭证的唯一标识，可通过：GET /ISAPI/AccessControl/capabilities中EmployeeNoInfo子节点employeeNo、characterType、isSupportCompress判定设备支持的工号字符串长度以及字符可配范围（一般都要支持32字节，任意字符。但对于门禁主机、经销型门禁产品需要根据这几个字段判断具体支持情况）。

35.2.2 人员查询



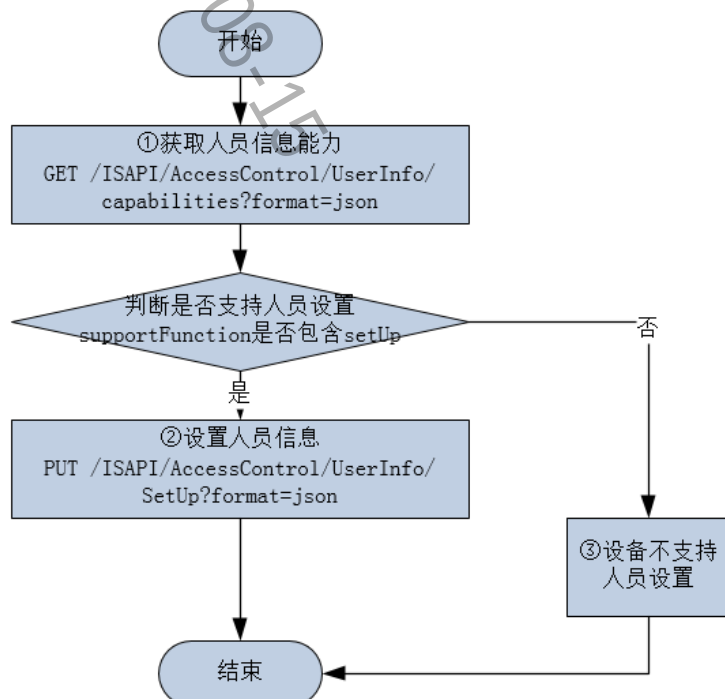
通过人员查询功能，可查询设备中已有的人员数量及人员信息：

1. 判断是否支持人员查询：GET /ISAPI/AccessControl/UserInfo/capabilities?format=json 中 supportFunction 包含 get，代表支持人员查询。
2. 查询人员数量：GET /ISAPI/AccessControl/UserInfo/Count?format=json 中返回的 userNumber，代表设备中已添加的人员数量。
3. 查询人员信息：POST /ISAPI/AccessControl/UserInfo/Search?format=json，通过分页的方式查询人员信息。
4. 设备不支持人员查询。

注：

（1）如需要获取设备支持的最大人员数量，可通过：GET /ISAPI/AccessControl/UserInfo/capabilities?format=json 中返回的 maxRecordNum 知晓。

35.2.3 人员设置



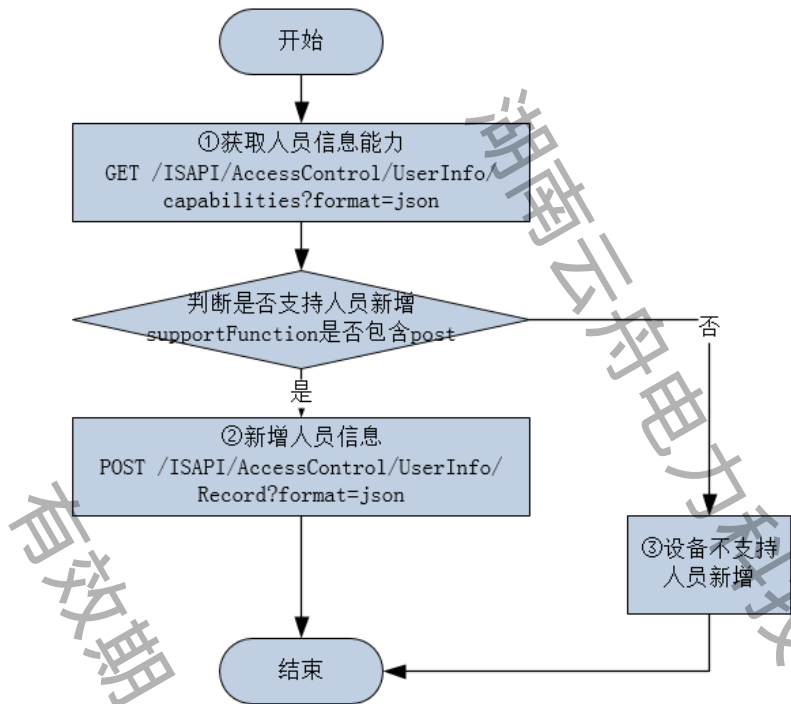
通过人员设置功能，下发人员信息到设备。如设备已存在该人员，则修改该人员信息；如设备未存在该人员，则新增该人员信息：

- 1. 判断是否支持人员设置：GET /ISAPI/AccessControl/UserInfo/capabilities?format=json中supportFunction包含setUp，代表支持人员设置。
- 2. 设置人员信息：PUT /ISAPI/AccessControl/UserInfo/SetUp?format=json。
- 3. 设备不支持人员设置。

注：

（1）根据人员设置中的employeeNo字段来判定设备中是否存在该人员。

35.2.4 人员新增



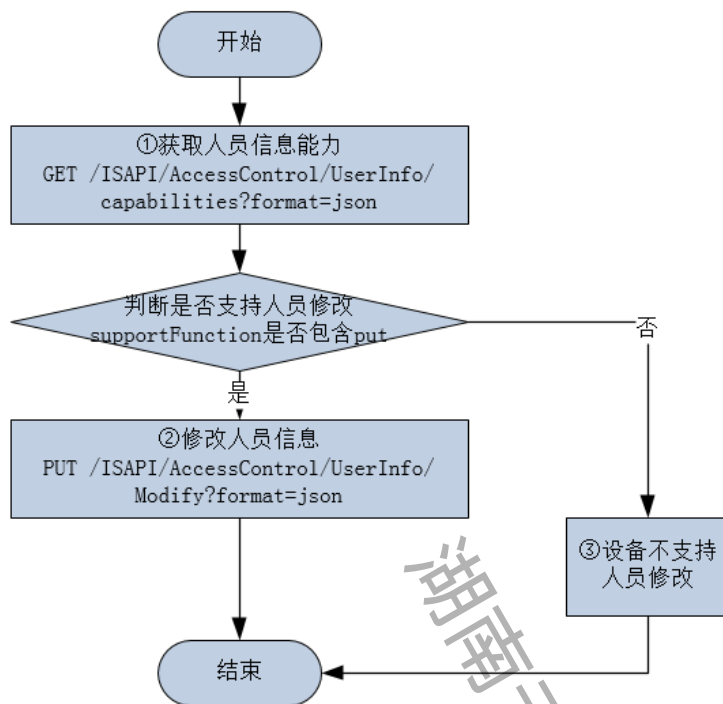
通过人员新增功能，新增人员信息到设备。如设备已存在该人员，则设备报错；如设备未存在该人员，则新增该人员信息：

- 1. 判断是否支持人员新增：GET /ISAPI/AccessControl/UserInfo/capabilities?format=json中supportFunction包含post，代表支持人员新增。
- 2. 新增人员信息：POST /ISAPI/AccessControl/UserInfo/Record?format=json。
- 3. 设备不支持人员新增。

注：

（1）根据人员新增中的employeeNo字段来判定设备中是否存在该人员。

35.2.5 人员修改



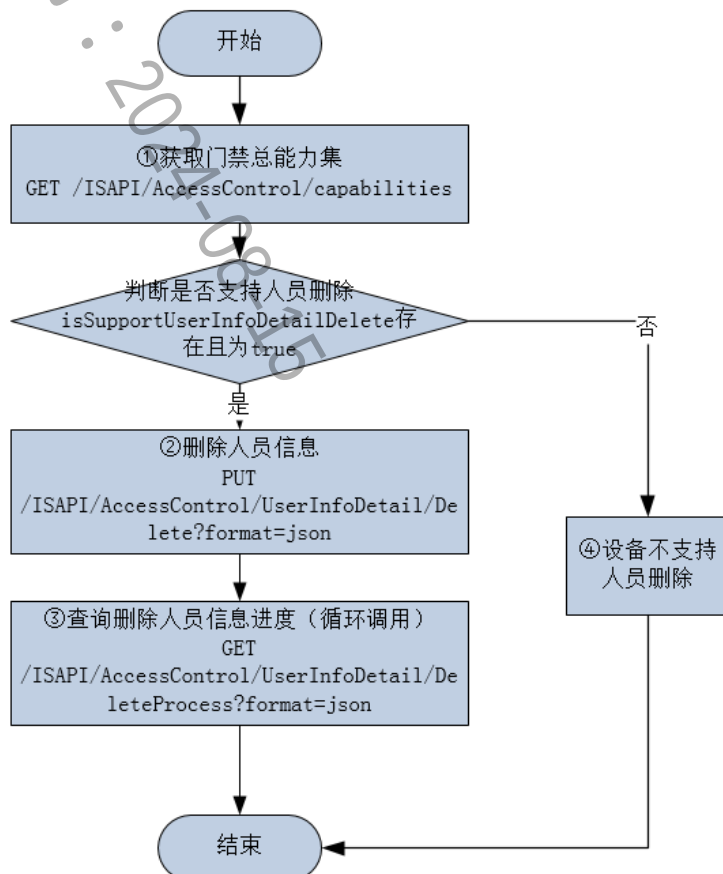
通过人员修改功能，修改人员信息到设备。如设备已存在该人员，则修改该人员信息；如设备未存在该人员，则设备报错：

1. 判断是否支持人员修改：GET /ISAPI/AccessControl/UserInfo/capabilities?format=json 中supportFunction包含put，代表支持人员修改。
2. 修改人员信息：PUT /ISAPI/AccessControl/UserInfo/Modify?format=json。
3. 设备不支持人员修改。

注：

- (1) 根据人员修改中的employeeNo字段来判定设备中是否存在该人员。

35.2.6 人员删除



通过人员删除功能，删除设备的人员信息。删除不存在的人员，设备不报错：

1. 判断是否支持人员删除：GET /ISAPI/AccessControl/capabilities 中返回isSupportUserInfoDetailDelete且为true，代表支持人员删除。
2. 删除人员信息：PUT /ISAPI/AccessControl/UserInfoDetail/Delete?format=json，该接口返回成功后，代表设备已开始执行删除动作，不代表设备已删除成功。
3. 查询删除人员信息进度：GET /ISAPI/AccessControl/UserInfoDetail/DeleteProcess，循环调用该接口，查询设备删除人员信息的进度。
4. 设备不支持人员删除。

注：

（1）在删除人员信息的同时，也会删除其关联（通过人员ID关联）的人员凭证信息（卡信息、指纹信息、人脸信息、虹膜信息）。

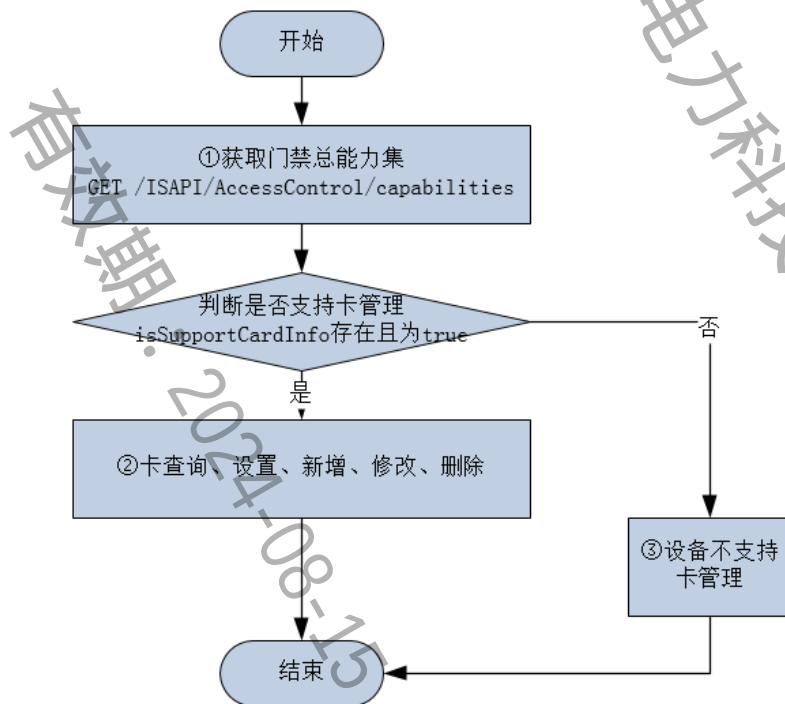
36 卡管理

36.1 功能介绍

卡管理包含卡的查询、设置、新增、修改、删除、采集，下面针对功能进行详细描述。

36.2 集成流程

36.2.1 判断是否支持卡管理



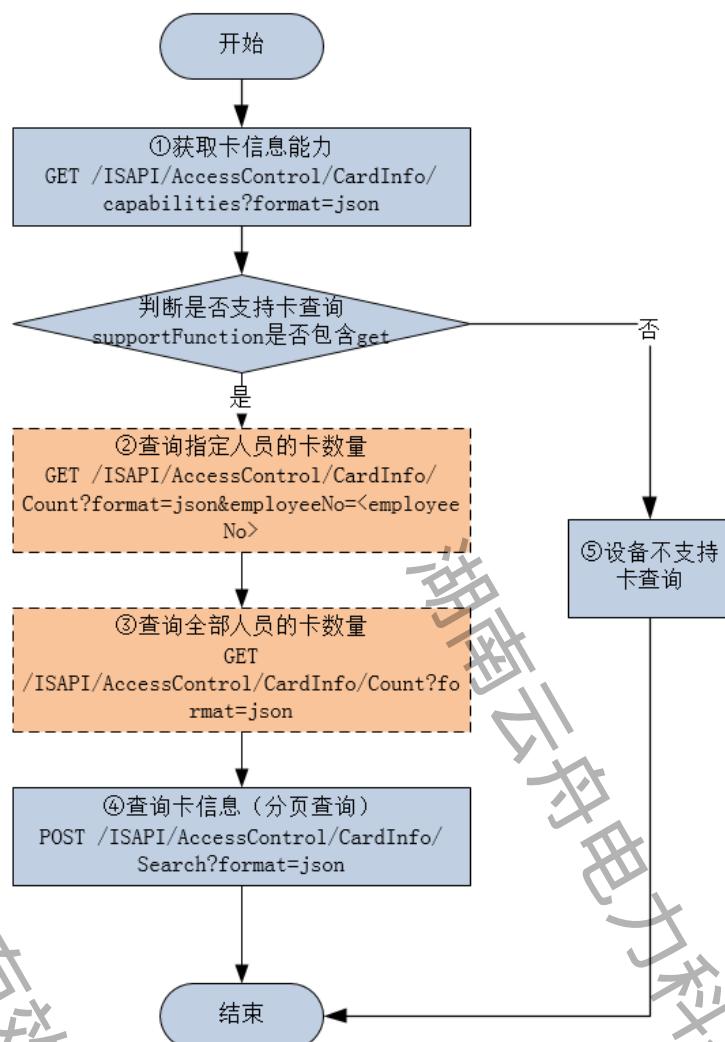
在集成卡管理功能前，首先要确保设备支持卡管理：

1. 判断是否支持卡管理：GET /ISAPI/AccessControl/capabilities 中返回isSupportCardInfo且为true，代表支持卡管理。
2. 进行卡查询、设置、新增、修改、删除。
3. 设备不支持卡管理。

注：

- （1）在设置/新增/修改卡信息到设备前，要保证设备中已下发相关人员信息（根据人员ID关联）到设备中。
- （2）可通过GET /ISAPI/AccessControl/CardInfo/capabilities?format=json中返回的numberPerPerson来知晓每个人支持的最大卡数量（返回255，代表无限制）；如设备未返回该字段，则每个人员至多支持5张卡信息下发。
- （3）可通过[GET/PUT] /ISAPI/AccessControl/CardVerificationRule?format=json 来进行不同字节的卡片管理。

36.2.2 卡查询



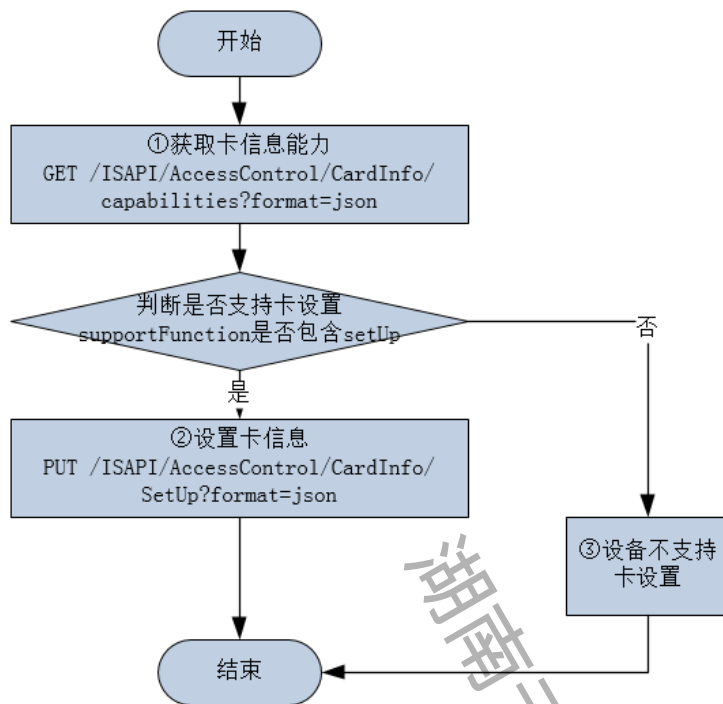
通过卡查询功能，可查询设备中已有的卡数量及卡信息：

1. 判断是否支持卡查询：GET /ISAPI/AccessControl/CardInfo/capabilities?format=json 中 supportFunction 包含 get，代表支持卡查询。
2. 查询指定人员的卡数量：GET /ISAPI/AccessControl/CardInfo/Count?format=json&employeeNo=<employeeNo> 中返回的 cardNumber，代表设备中指定人员已添加的卡数量。
3. 查询全部人员的卡数量：GET /ISAPI/AccessControl/CardInfo/Count?format=json 中返回的 cardNumber，代表设备中全部人员已添加的卡数量。
4. 查询卡信息：POST /ISAPI/AccessControl/CardInfo/Search?format=json，通过分页的方式查询卡信息。
5. 设备不支持卡查询。

注：

(1) 如需要获取设备支持的最大卡数量，可通过：GET /ISAPI/AccessControl/CardInfo/capabilities?format=json 中返回的 maxRecordNum 知晓。

36.2.3 卡设置



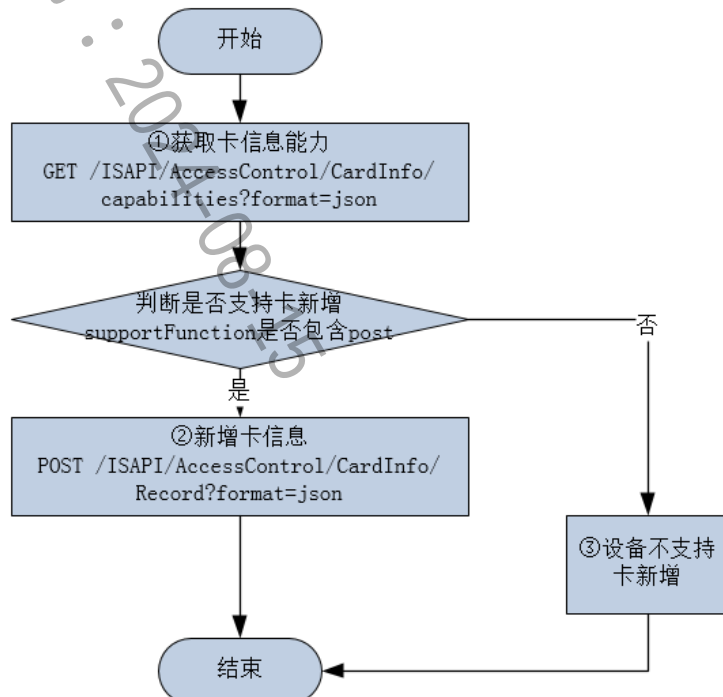
通过卡设置功能，下发卡信息到设备。如设备已存在该卡，则修改该卡信息；如设备未存在该卡，则新增该卡信息：

1. 判断是否支持卡设置：GET /ISAPI/AccessControl/CardInfo/capabilities?format=json 中supportFunction包含setUp，代表支持卡设置。
2. 设置卡信息：PUT /ISAPI/AccessControl/CardInfo/SetUp?format=json。
3. 设备不支持卡设置。

注：

- (1) 根据卡设置中的cardNo字段来判定设备中是否存在该卡。

36.2.4 卡新增



通过卡新增功能，新增卡信息到设备。如设备已存在该卡，则设备报错；如设备未存在该卡，则新增该卡信息：

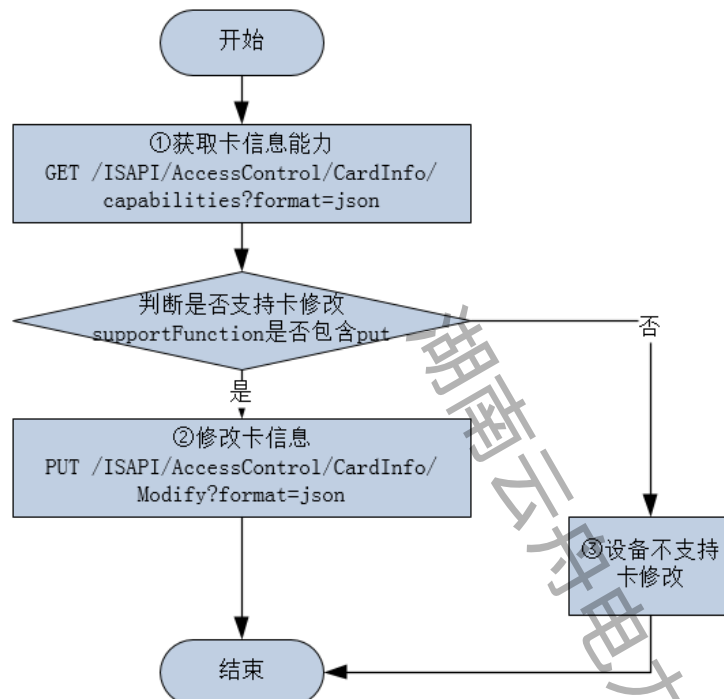
1. 判断是否支持卡新增：GET /ISAPI/AccessControl/CardInfo/capabilities?format=json 中supportFunction包含post，代表支持卡新增。
2. 新增卡信息：POST /ISAPI/AccessControl/CardInfo/Record?format=json。

3. 设备不支持卡新增。

注：

(1) 根据卡新增中的cardNo字段来判定设备中是否存在该卡。

36.2.5 卡修改



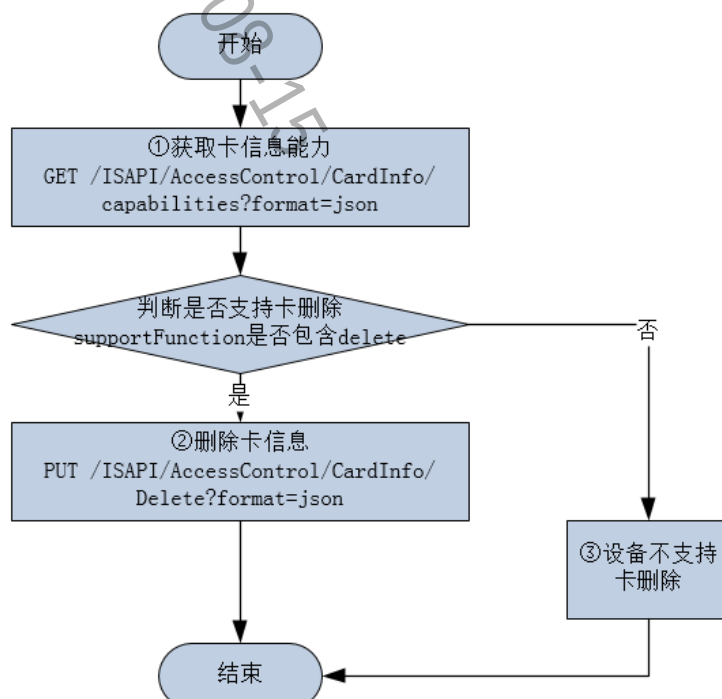
通过卡修改功能，修改卡信息到设备。如设备已存在该卡，则修改该卡信息；如设备未存在该卡，则设备报错：

1. 判断是否支持卡修改：GET /ISAPI/AccessControl/CardInfo/capabilities?format=json中supportFunction包含put，代表支持卡修改。
2. 修改卡信息：PUT /ISAPI/AccessControl/CardInfo/Modify?format=json。
3. 设备不支持卡修改。

注：

(1) 根据卡修改中的cardNo字段来判定设备中是否存在该卡。

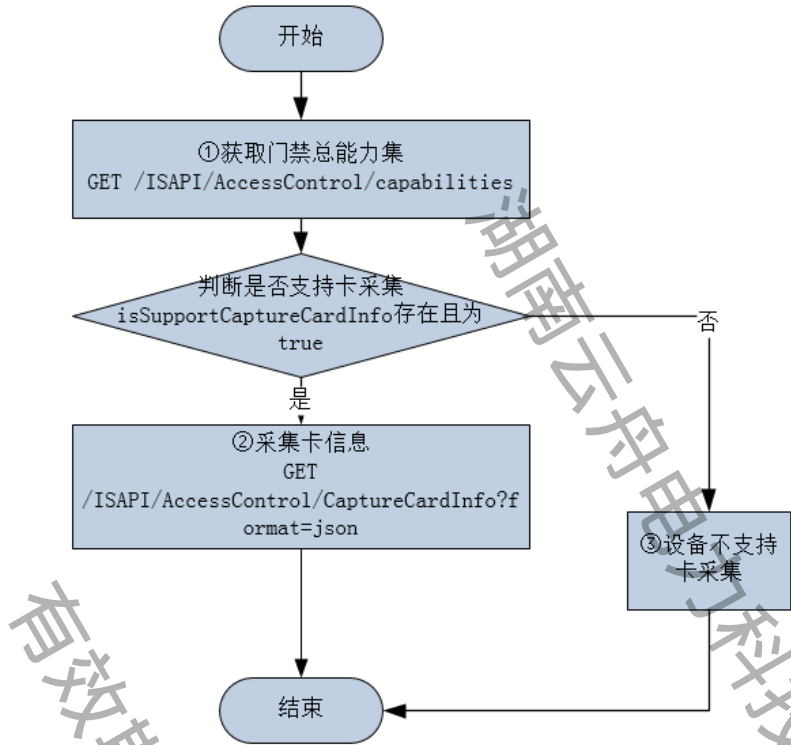
36.2.6 卡删除



通过卡删除功能，删除设备的卡信息。删除不存在的卡，设备不报错：

- 1. 判断是否支持卡删除：GET /ISAPI/AccessControl/CardInfo/capabilities?format=json中supportFunction包含delete，代表支持卡删除。
- 2. 删除卡信息：PUT /ISAPI/AccessControl/CardInfo/Delete?format=json，该接口返回成功后，代表设备已删除成功。
- 3. 设备不支持卡删除。

36.2.7 卡采集



通过卡采集功能，可采集卡号、卡类型等信息：

- 1. 判断是否支持卡采集：GET /ISAPI/AccessControl/capabilities中返回isSupportCaptureCardInfo且为true，代表支持卡采集。
- 2. 采集卡信息：GET /ISAPI/AccessControl/CaptureCardInfo?format=json。
- 3. 设备不支持卡采集。

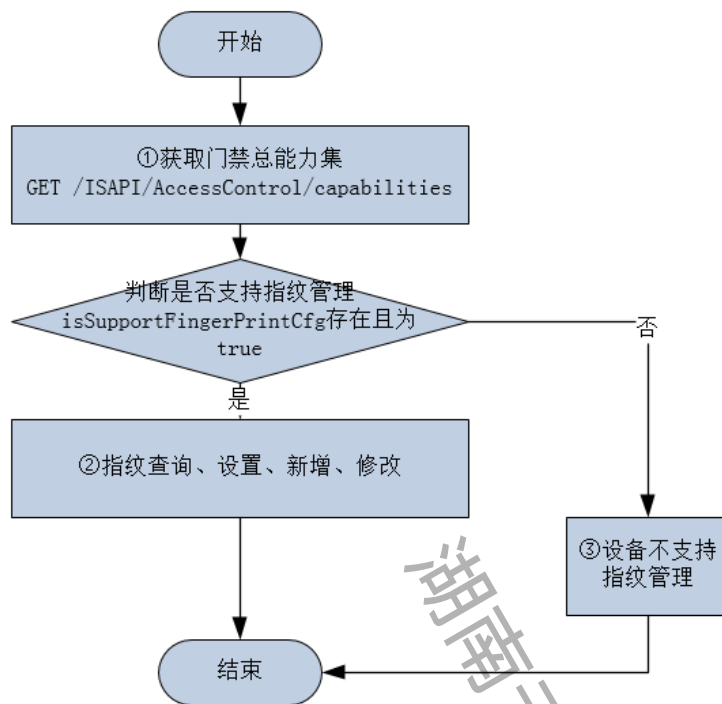
37 指纹管理

37.1 功能介绍

指纹管理包含指纹的查询、设置、删除、采集，下面针对功能进行详细描述。

37.2 集成流程

37.2.1 判断是否支持指纹管理



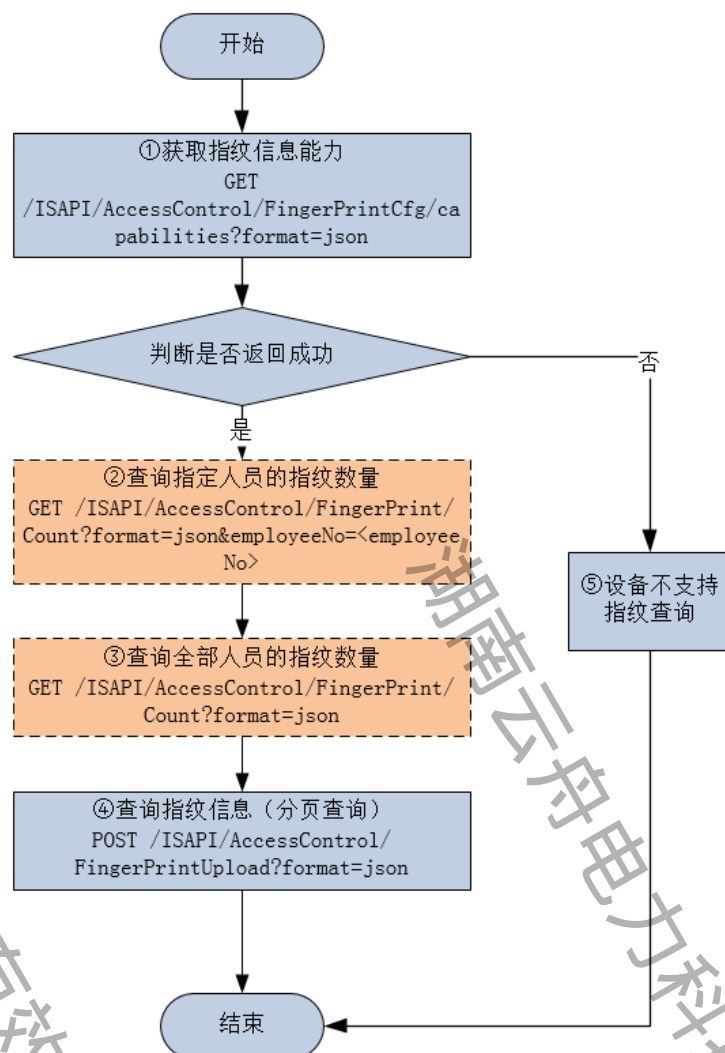
在集成指纹管理功能前，首先要确保设备支持指纹管理：

1. 判断是否支持指纹管理：GET /ISAPI/AccessControl/capabilities中返回isSupportFingerPrintCfg且为true，代表支持指纹管理。
2. 进行指纹查询、设置、新增、修改。
3. 设备不支持指纹管理。

注：

- (1) 在设置指纹信息到设备前，要保证设备中已下发相关人员信息（根据人员ID关联）到设备中。
- (2) 当前设备中，每个人员至多支持10枚指纹信息下发，分别对应人员的10枚手指。

37.2.2 指纹查询



通过指纹查询功能，可查询设备中已有的指纹数量及指纹信息：

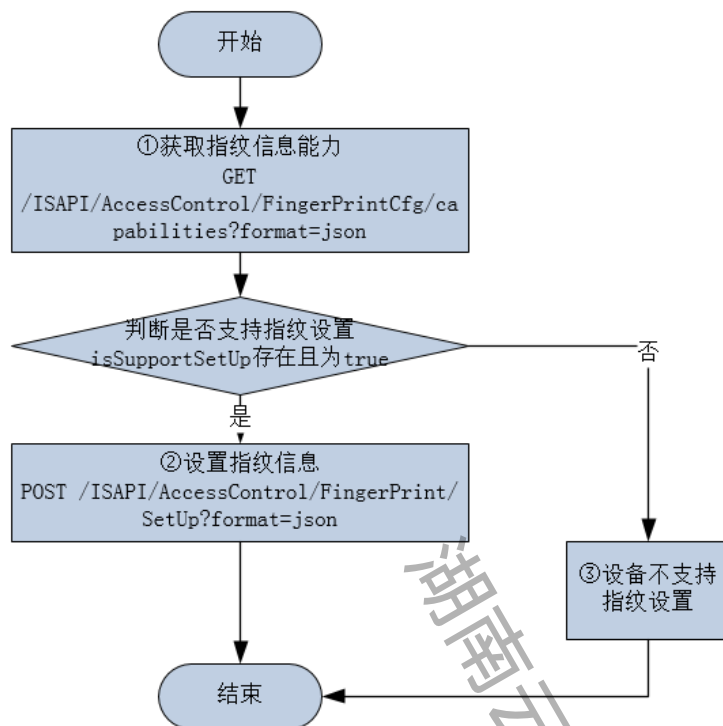
1. 判断是否支持指纹查询：GET /ISAPI/AccessControl/FingerPrintCfg/capabilities?format=json，接口返回成功，代表支持指纹查询。
2. 查询指定人员的指纹数量：GET /ISAPI/AccessControl/FingerPrint/Count?format=json&employeeNo=<employeeNo>中返回的numberOfFP，代表设备中指定人员已添加的指纹数量。
3. 查询全部人员的指纹数量：GET /ISAPI/AccessControl/FingerPrint/Count?format=json中返回的numberOfFP，代表设备中全部人员已添加的指纹数量。
4. 查询指纹信息：POST /ISAPI/AccessControl/FingerPrintUpload?format=json，通过分页的方式查询指纹信息，直到返回的FingerPrintInfo中status字段为NoFP，代表已获取完成。
5. 设备不支持指纹查询。

注：

（1）如需要获取某一读卡器的最大指纹数量，可通过：GET /ISAPI/AccessControl/CardReaderCfg/<cardReaderID>?format=json中返回的fingerPrintCapacity知晓。

（2）如需要获取某一读卡器的已添加指纹数量，可通过：GET /ISAPI/AccessControl/CardReaderCfg/<cardReaderID>?format=json中返回的fingerPrintNum知晓。

37.2.3 指纹设置



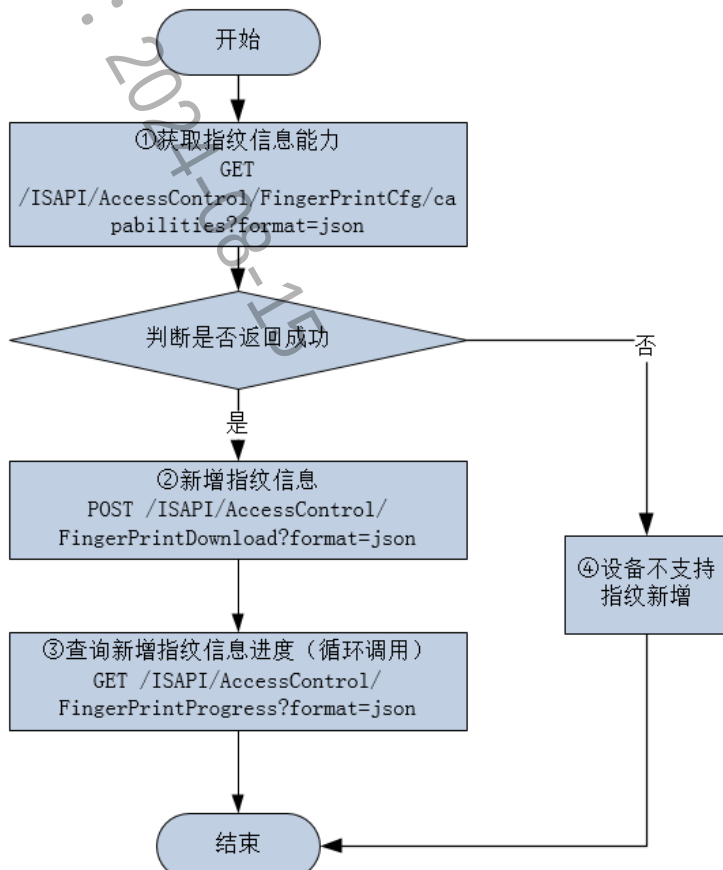
通过指纹设置功能，下发指纹信息到设备。如设备已存在该指纹，则修改该指纹信息；如设备未存在该指纹，则新增该指纹信息：

1. 判断是否支持指纹设置：GET /ISAPI/AccessControl/FingerPrintCfg/capabilities?format=json中返回isSupportSetUp且为true，代表支持指纹设置。
2. 设置指纹信息：POST /ISAPI/AccessControl/FingerPrint/Setup?format=json。
3. 设备不支持指纹设置。

注：

- (1) 根据指纹设置中的employeeNo、fingerPrintID字段来判定设备中是否存在该指纹。

37.2.4 指纹新增



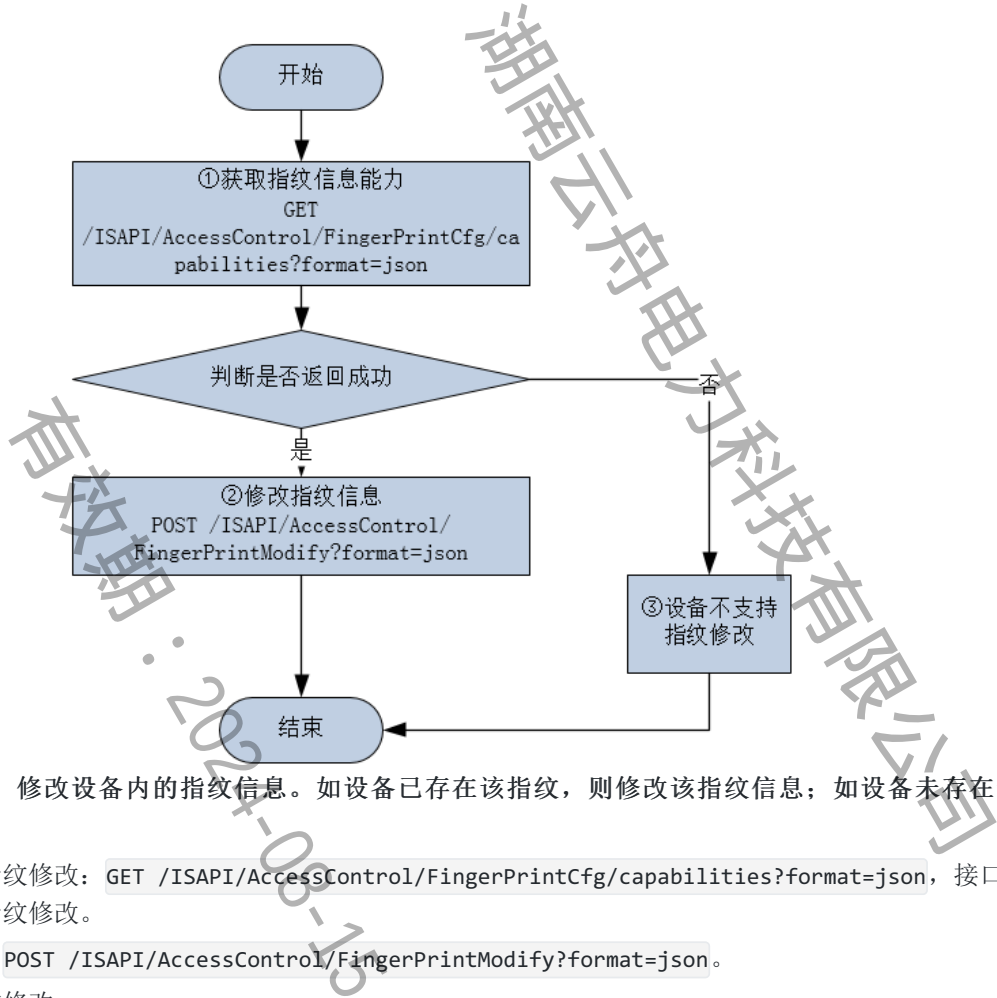
通过指纹新增功能，新增指纹信息到设备。如设备已存在该指纹，则设备报错；如设备未存在该指纹，则新增该指纹信息：

- 1. 判断是否支持指纹新增：GET /ISAPI/AccessControl/FingerPrintCfg/capabilities?format=json，接口返回成功，代表支持指纹新增。
- 2. 新增指纹信息：POST /ISAPI/AccessControl/FingerPrintDownload?format=json，该接口返回成功后，代表设备已开始执行新增指纹动作，不代表设备已新增指纹成功。
- 3. 查询新增指纹信息进度：GET /ISAPI/AccessControl/FingerPrintProgress?format=json，循环调用该接口，查询设备新增指纹信息的进度。
- 4. 设备不支持指纹新增。

注：

- (1) 根据指纹设置中的employeeNo、fingerPrintID字段来判定设备中是否存在该指纹。

37.2.5 指纹修改



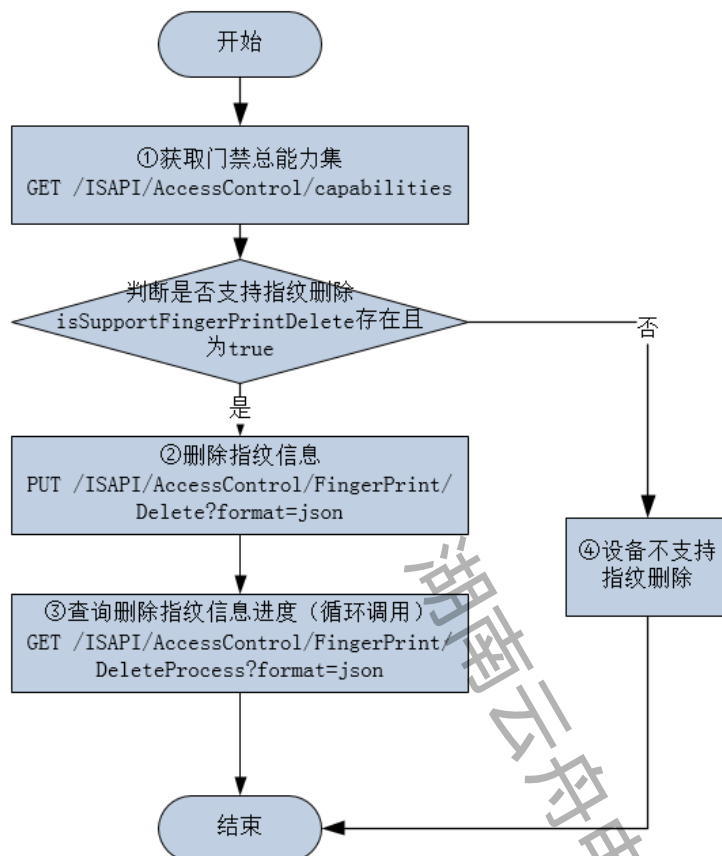
通过指纹修改功能，修改设备内的指纹信息。如设备已存在该指纹，则修改该指纹信息；如设备未存在该指纹，则设备报错：

- 1. 判断是否支持指纹修改：GET /ISAPI/AccessControl/FingerPrintCfg/capabilities?format=json，接口返回成功，代表支持指纹修改。
- 2. 修改指纹信息：POST /ISAPI/AccessControl/FingerPrintModify?format=json。
- 3. 设备不支持指纹修改。

注：

- (1) 根据指纹设置中的employeeNo、fingerPrintID字段来判定设备中是否存在该指纹。
- (2) 修改指纹信息时，仅修改相关指纹参数，不对指纹数据进行修改。

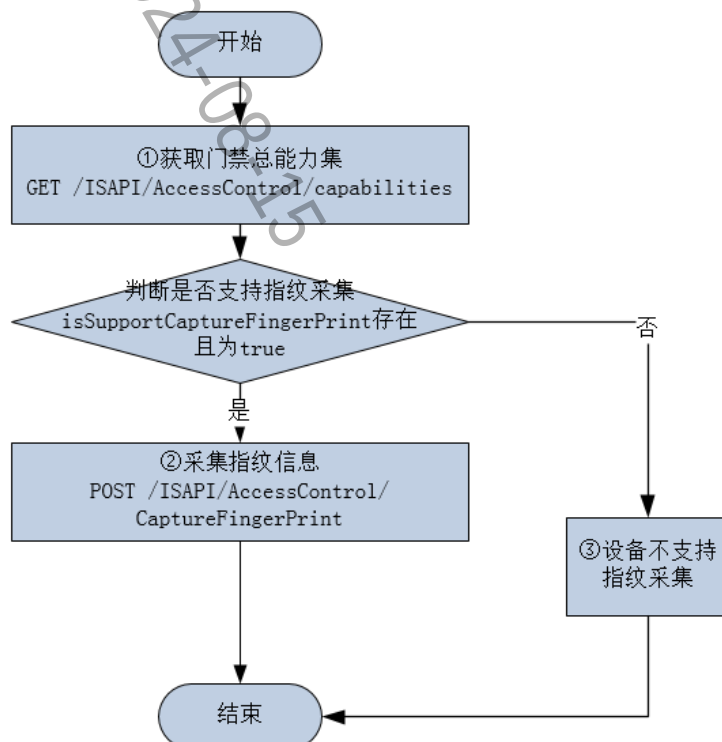
37.2.6 指纹删除



通过指纹删除功能，删除设备的指纹信息。删除不存在的指纹，设备不报错：

1. 判断是否支持指纹删除：GET /ISAPI/AccessControl/capabilities 中返回isSupportFingerPrintDelete且为true，代表支持指纹删除。
2. 删除指纹信息：PUT /ISAPI/AccessControl/FingerPrint/Delete?format=json，该接口返回成功后，代表设备已开始执行删除动作，不代表设备已删除成功。
3. 查询删除指纹信息进度：GET /ISAPI/AccessControl/FingerPrint/DeleteProcess?format=json，循环调用该接口，查询设备删除指纹信息的进度。
4. 设备不支持指纹删除。

37.2.7 指纹采集



通过指纹采集功能，可采集指纹数据、指纹质量等信息：

1. 判断是否支持指纹采集: `GET /ISAPI/AccessControl/capabilities`中返回`isSupportCaptureFingerPrint`且为`true`, 代表支持指纹采集。
2. 采集指纹信息: `POST /ISAPI/AccessControl/CaptureFingerPrint`。
3. 设备不支持指纹采集。

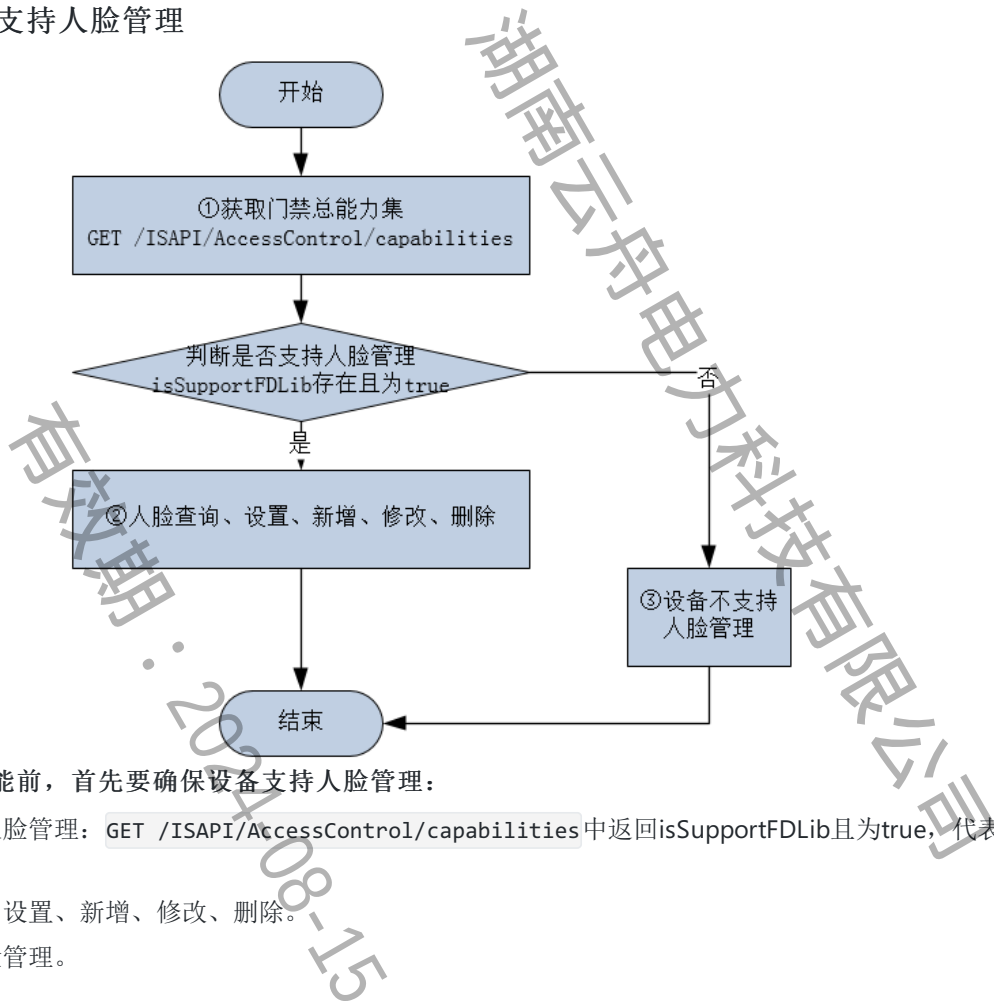
38 人脸管理

38.1 功能介绍

人脸管理包含人脸的查询、设置、新增、修改、删除、采集, 下面针对功能进行详细描述。

38.2 集成流程

38.2.1 判断是否支持人脸管理



在集成人脸管理功能前, 首先要确保设备支持人脸管理:

1. 判断是否支持人脸管理: `GET /ISAPI/AccessControl/capabilities`中返回`isSupportFDLib`且为`true`, 代表支持人脸管理。
2. 进行人脸查询、设置、新增、修改、删除。
3. 设备不支持人脸管理。

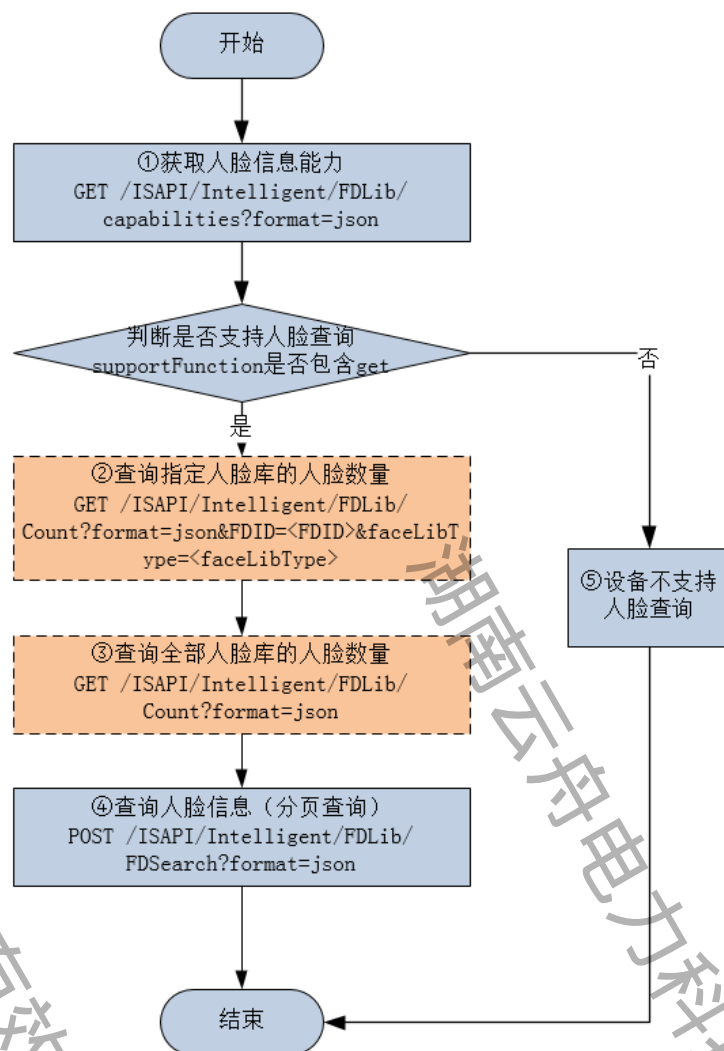
注:

(1) 在设置/新增/修改人脸信息到设备前, 要保证设备中已下发相关人员信息(根据人员ID关联)到设备中, 且需要通过`GET /ISAPI/Intelligent/FDLib?format=json`确定设备中已存在人脸库(如不存在, 则需要通过`POST /ISAPI/Intelligent/FDLib?format=json`进行人脸库创建), 其中可见光人脸库ID(FDID)为1。

(2) 对于支持深度模式(采用红外图片进行人脸比对, `GET /ISAPI/AccessControl/FaceRecognizeMode/capabilities?format=json`中`mode`节点包含`deepMode`)的设备: 人脸库ID(FDID)为2, 下发红外图片到该人脸库, 用红外图片进行人脸比对; 人脸库ID(FDID)为1, 下发可见光图片到该人脸库, 用可见光图片进行设备本地UI界面展示。

(3) 设备深度模式与普通模式切换: `[GET/PUT] /ISAPI/AccessControl/FaceRecognizeMode?format=json`中通过`mode`节点进行切换。

38.2.2 人脸查询



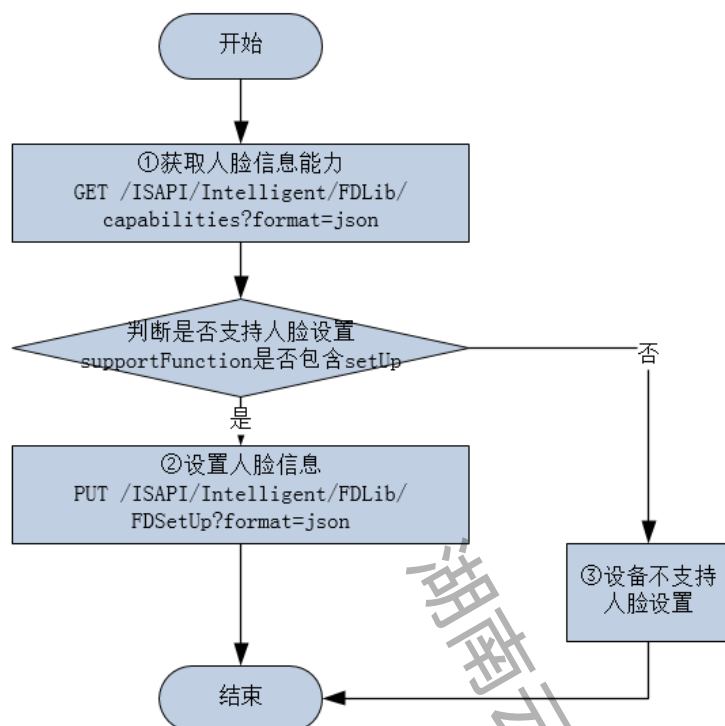
通过人脸查询功能，可查询设备中已有的人脸数量及人脸信息：

1. 判断是否支持人脸查询：GET /ISAPI/Intelligent/FDLib/capabilities?format=json中supportFunction包含get，代表支持人脸查询。
2. 查询指定人脸库的人脸数量：GET /ISAPI/Intelligent/FDLib/Count?format=json&FDID=<FDID>&faceLibType=<faceLibType>中返回的recordDataNumber，代表设备中指定人脸库已添加的人脸数量。
3. 查询全部人脸库的人脸数量：GET /ISAPI/Intelligent/FDLib/Count?format=json中返回的recordDataNumber，代表不同人脸库已添加的人脸数量。
4. 查询人脸信息：POST /ISAPI/Intelligent/FDLib/FDSearch?format=json，通过分页的方式查询人脸信息。
5. 设备不支持人脸查询。

注：

（1）如需要获取设备支持的最大人脸数量，可通过：GET /ISAPI/Intelligent/FDLib/capabilities?format=json中返回的FDRecordDataMaxNum知晓。

38.2.3 人脸设置



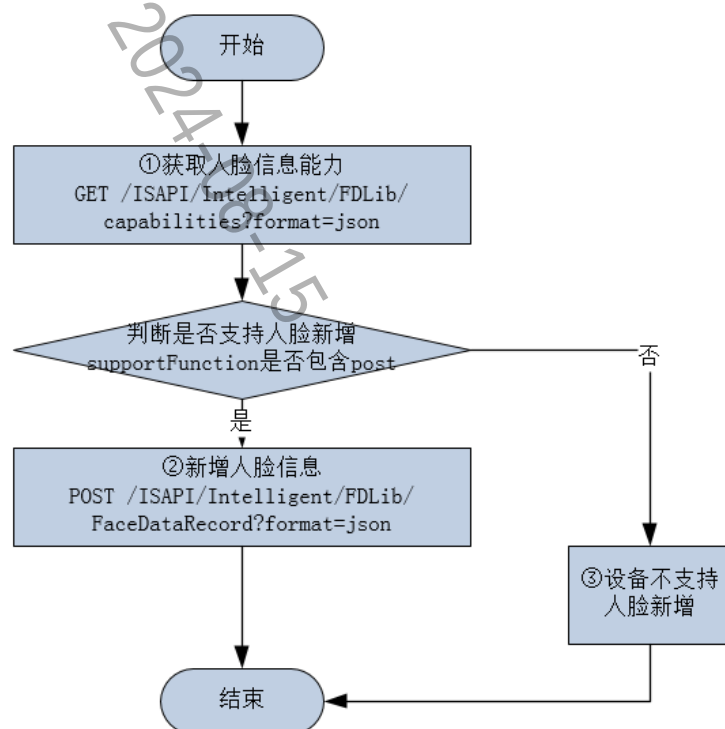
通过人脸设置功能，下发人脸信息到设备。如设备已存在该人脸，则修改该人脸信息；如设备未存在该人脸，则新增该人脸信息：

1. 判断是否支持人脸设置：GET /ISAPI/Intelligent/FDLib/capabilities?format=json中supportFunction包含setUp，代表支持人脸设置。
2. 设置人脸信息：PUT /ISAPI/Intelligent/FDLib/FDSetUp?format=json。
3. 设备不支持人脸设置。

注：

（1）根据人脸设置中的FPID字段来判定设备中是否存在该人脸，并通过人脸管理中的FPID字段和人员管理中的employeeNo字段，来关联人脸和人员信息。

38.2.4 人脸新增



通过人脸新增功能，新增人脸信息到设备。如设备已存在该人脸，则设备报错；如设备未存在该人脸，则新增该人脸信息：

1. 判断是否支持人脸新增：GET /ISAPI/Intelligent/FDLib/capabilities?format=json中supportFunction包含

post，代表支持人脸新增。

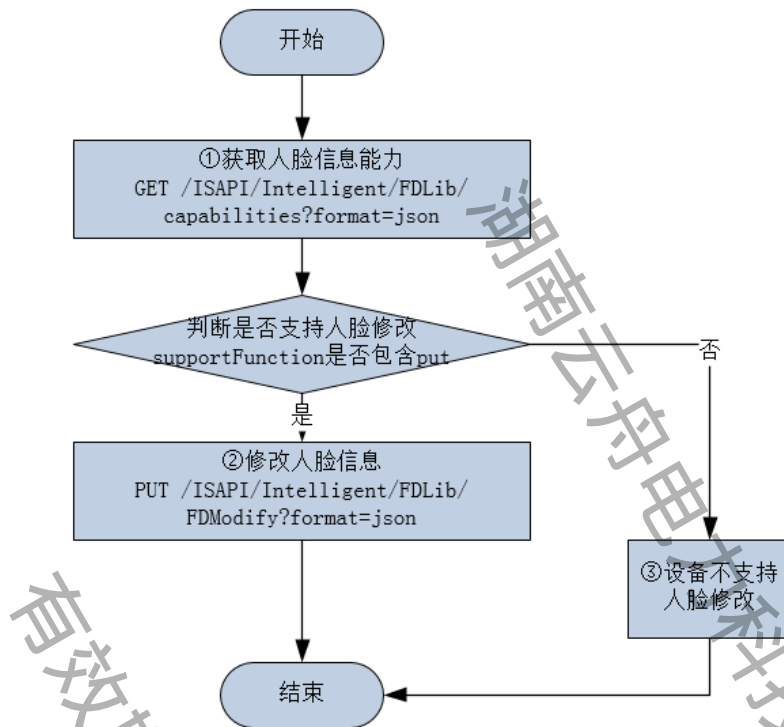
2. 新增人脸信息：POST /ISAPI/Intelligent/FDLib/FaceDataRecord?format=json。

3. 设备不支持人脸新增。

注：

（1）根据人脸新增中的FPID字段来判定设备中是否存在该人脸，并通过人脸管理中的FPID字段和人员管理中的employeeNo字段，来关联人脸和人员信息。

38.2.5 人脸修改



通过人脸修改功能，修改人脸信息到设备。如设备已存在该人脸，则修改该人脸信息；如设备未存在该人脸，则设备报错：

1. 判断是否支持人脸修改：GET /ISAPI/Intelligent/FDLib/capabilities?format=json中supportFunction包含put，代表支持人脸修改。

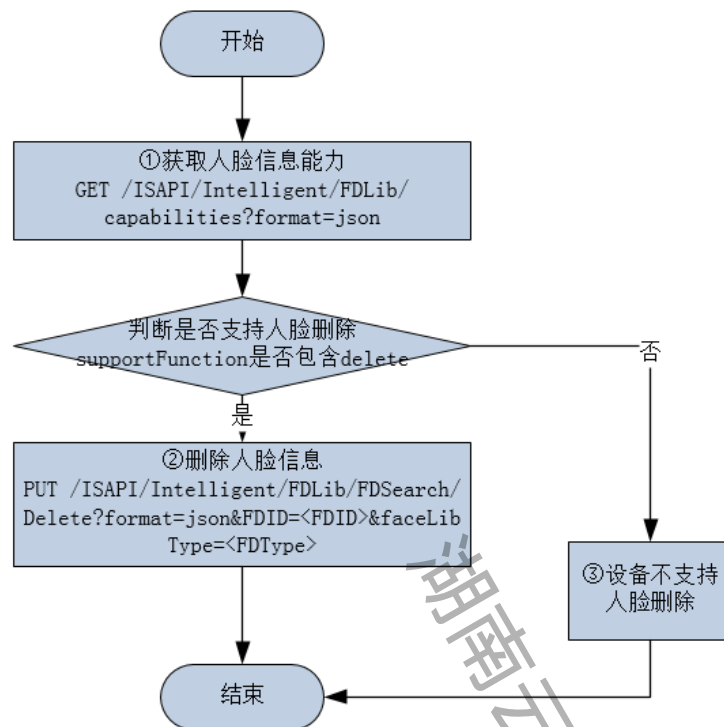
2. 修改人脸信息：PUT /ISAPI/Intelligent/FDLib/FDModify?format=json。

3. 设备不支持人脸修改。

注：

（1）根据人脸修改中的FPID字段来判定设备中是否存在该人脸，并通过人脸管理中的FPID字段和人员管理中的employeeNo字段，来关联人脸和人员信息。

38.2.6 人脸删除



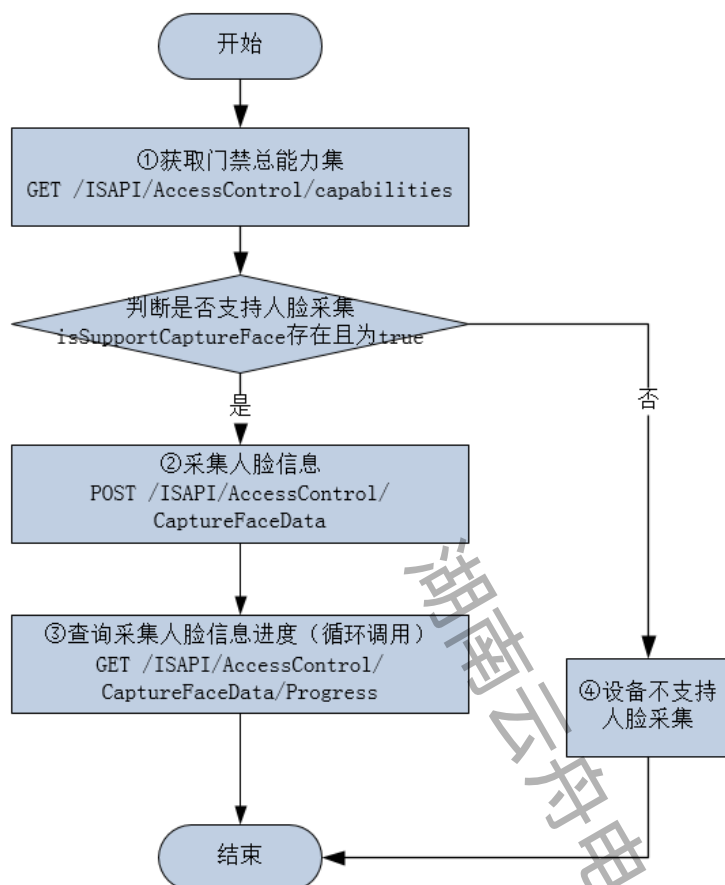
通过人脸删除功能，删除设备的人脸信息。删除不存在的人脸，设备不报错：

1. 判断是否支持人脸删除：GET /ISAPI/Intelligent/FDLib/capabilities?format=json 中 supportFunction 包含 delete，代表支持人脸删除。
2. 删除人脸信息：PUT /ISAPI/Intelligent/FDLib/FDSearch/Delete?format=json&FDID=<FDID>&faceLibType=<FDType>，该接口返回成功后，代表设备已删除成功。
3. 设备不支持人脸删除。

注：

(1) 可通过 DELETE /ISAPI/Intelligent/FDLib?format=json 接口删除设备内的所有人脸库，同时删除人脸库中的所有人脸信息。

38.2.7 人脸采集



通过人脸采集功能，可采集人脸数据、人脸评分等信息：

1. 判断是否支持人脸采集：GET /ISAPI/AccessControl/capabilities 中返回isSupportCaptureFace且为true，代表支持（可见光）人脸采集；返回isSupportCaptureInfraredFace且为true，代表支持（红外）人脸采集。
2. 采集人脸信息：POST /ISAPI/AccessControl/CaptureFaceData，进行人脸采集中：
 - （1）如captureProgress字段返回100，代表采集到人脸，直接解析采集到的人脸图片二进制数据或URL。
 - （2）如captureProgress字段返回0，代表未采集到人脸，需要查询采集人脸信息进度。
3. 查询采集人脸信息进度：GET /ISAPI/AccessControl/CaptureFaceData/Progress，循环调用该接口：
 - （1）直到captureProgress返回100，代表采集成功，可以解析采集到的人脸图片二进制数据或URL，则停止循环调用。
 - （2）或captureProgress返回0的同时isCurRequestOver返回true，代表采集人脸失败，则停止循环调用。
4. 设备不支持人脸采集。

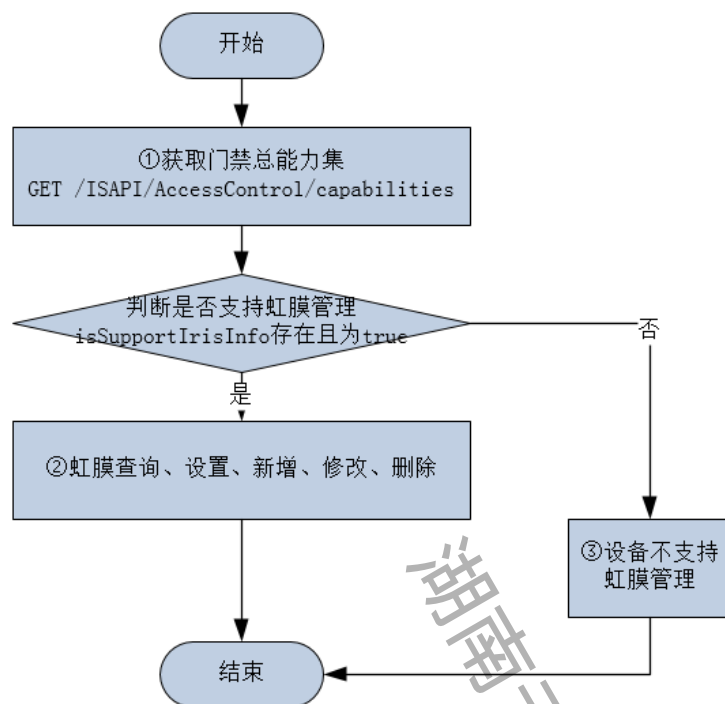
39 虹膜管理

39.1 功能介绍

虹膜管理包含虹膜的查询、设置、新增、修改、删除、采集，下面针对功能进行详细描述。

39.2 集成流程

39.2.1 判断是否支持虹膜管理



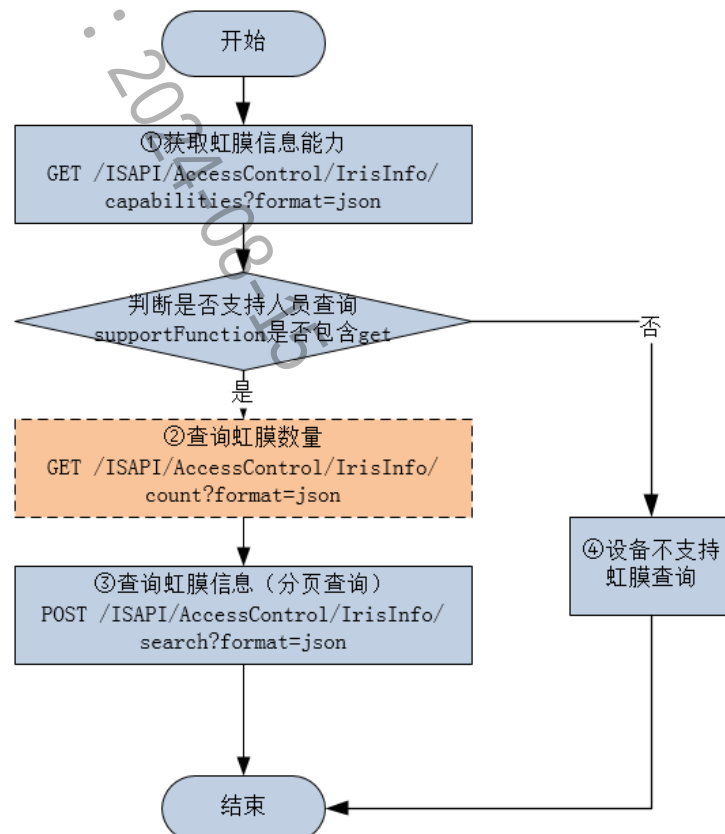
在集成虹膜管理功能前，首先要确保设备支持虹膜管理：

1. 判断是否支持虹膜管理：GET /ISAPI/AccessControl/capabilities中返回isSupportIrisInfo且为true，代表支持虹膜管理。
2. 进行虹膜查询、设置、新增、修改、删除。
3. 设备不支持虹膜管理。

注：

- (1) 在设置/新增/修改虹膜信息到设备前，要保证设备中已下发相关人员信息（根据人员ID关联）到设备中。
- (2) 当前设备中，每个人员至多支持2枚虹膜信息下发，分别对应人员的左眼和右眼。

39.2.2 虹膜查询



通过虹膜查询功能，可查询设备中已有的虹膜数量及虹膜信息：

1. 判断是否支持虹膜查询：GET /ISAPI/AccessControl/IrisInfo/capabilities?format=json中supportFunction包

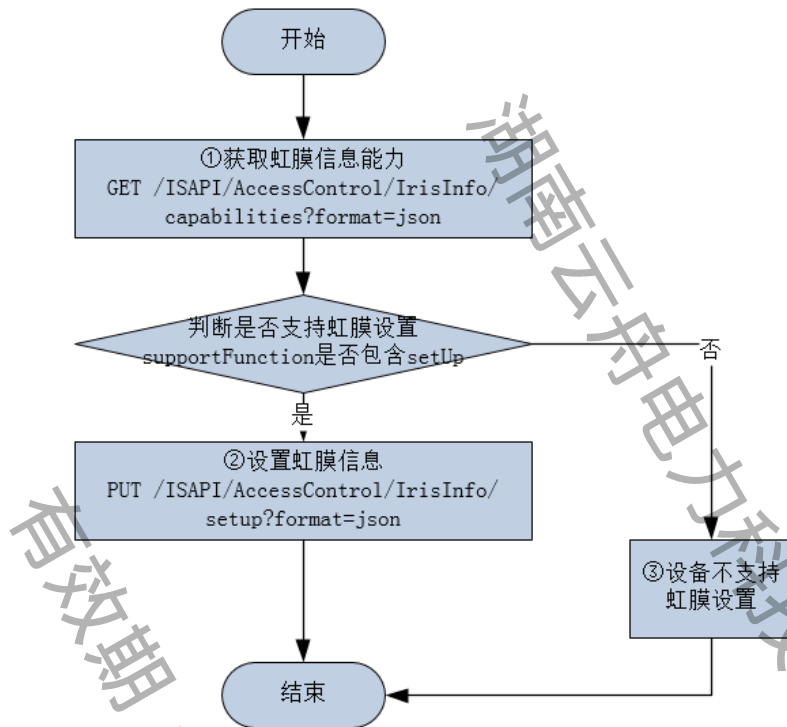
含get，代表支持虹膜查询。

2. 查询虹膜数量：GET /ISAPI/AccessControl/IrisInfo/count?format=json中返回的IrisNumber，代表设备中已添加的虹膜数量。
3. 查询虹膜信息：POST /ISAPI/AccessControl/IrisInfo/search?format=json，通过分页的方式查询虹膜信息。
4. 设备不支持虹膜查询。

注：

（1）如需要获取设备支持的最大虹膜数量，可通过：GET /ISAPI/AccessControl/IrisInfo/capabilities?format=json中返回的maxRecordNum知晓。

39.2.3 虹膜设置



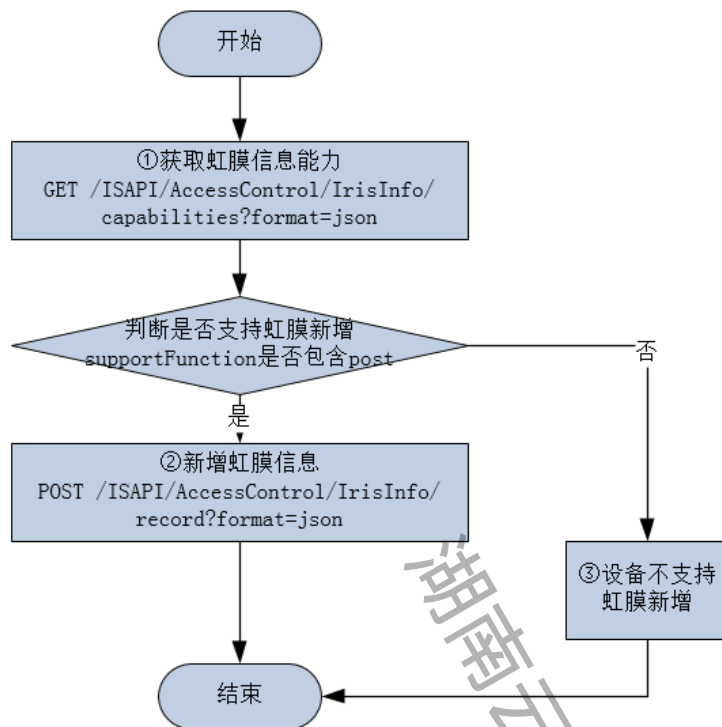
通过虹膜设置功能，下发虹膜信息到设备。如设备已存在该虹膜，则修改该虹膜信息；如设备未存在该虹膜，则新增该虹膜信息：

1. 判断是否支持虹膜设置：GET /ISAPI/AccessControl/IrisInfo/capabilities?format=json中supportFunction包含setUp，代表支持虹膜设置。
2. 设置虹膜信息：PUT /ISAPI/AccessControl/IrisInfo/setup?format=json。
3. 设备不支持虹膜设置。

注：

（1）根据虹膜设置中的employeeNo和id字段来判定设备中是否存在该虹膜。

39.2.4 虹膜新增



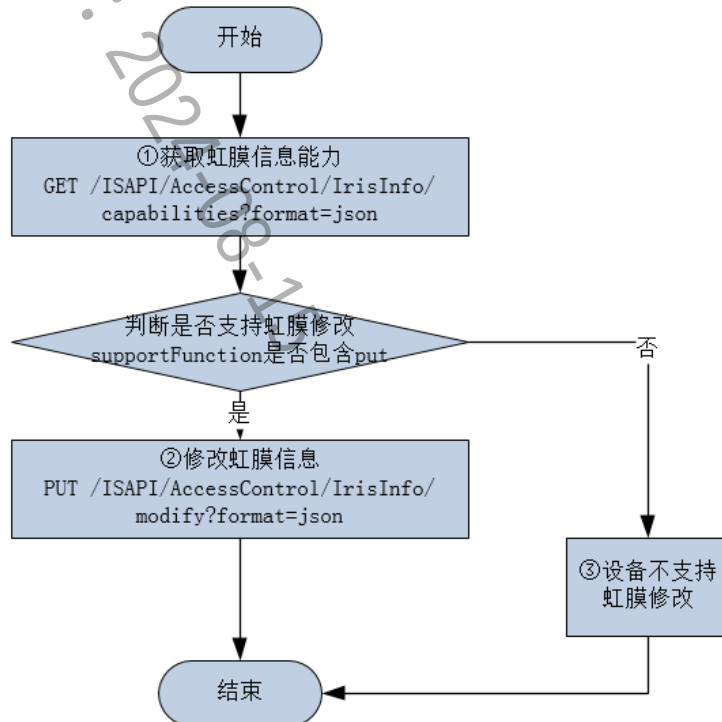
通过虹膜新增功能，新增虹膜信息到设备。如设备已存在该虹膜，则设备报错；如设备未存在该虹膜，则新增该虹膜信息：

1. 判断是否支持虹膜新增：GET /ISAPI/AccessControl/IrisInfo/capabilities?format=json中supportFunction包含post，代表支持虹膜新增。
2. 新增虹膜信息：POST /ISAPI/AccessControl/IrisInfo/record?format=json。
3. 设备不支持虹膜新增。

注：

- (1) 根据虹膜新增中的employeeNo和id字段来判定设备中是否存在该虹膜。

39.2.5 虹膜修改



通过虹膜修改功能，修改虹膜信息到设备。如设备已存在该虹膜，则修改该虹膜信息；如设备未存在该虹膜，则设备报错：

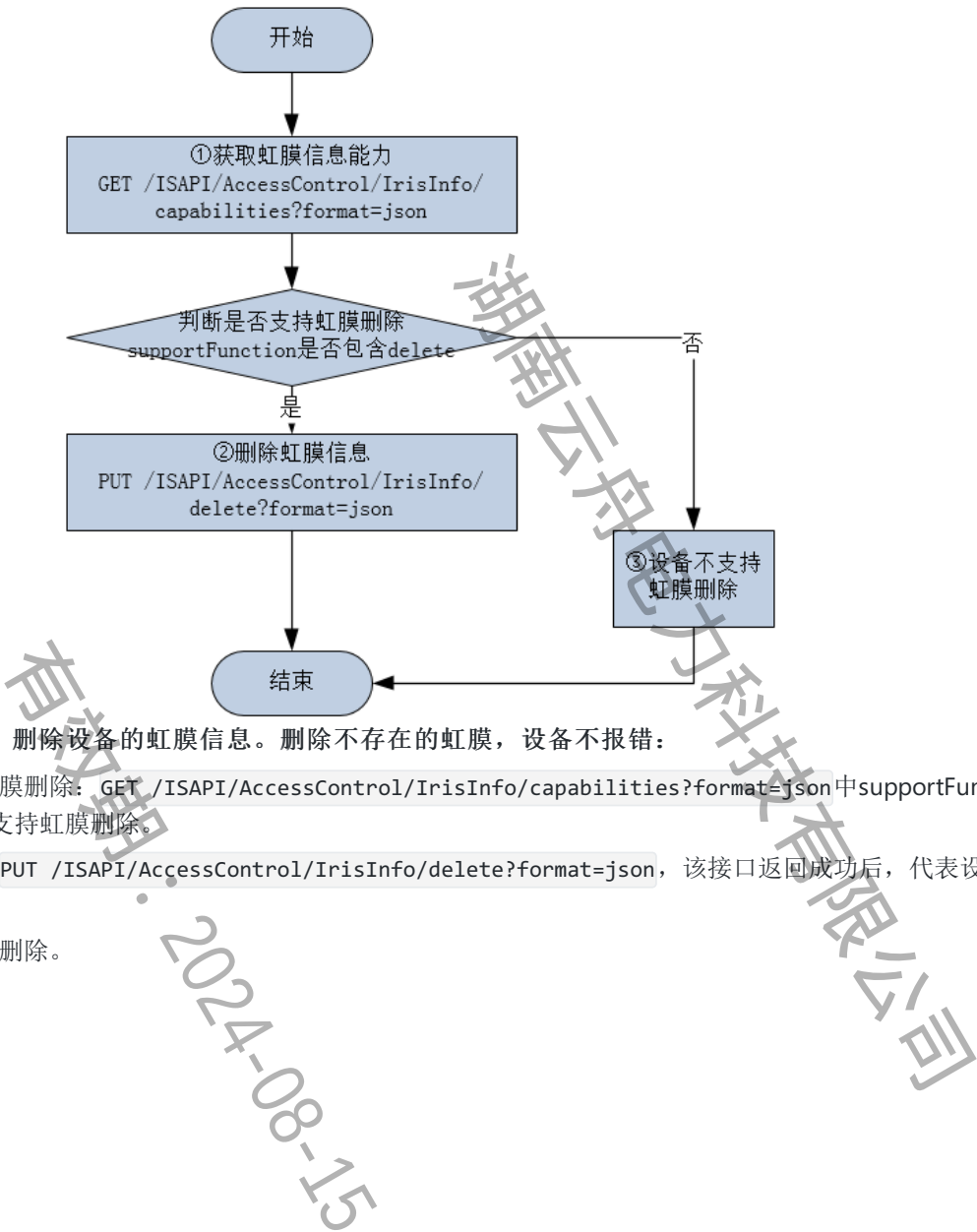
1. 判断是否支持虹膜修改：GET /ISAPI/AccessControl/IrisInfo/capabilities?format=json中supportFunction包含put，代表支持虹膜修改。

- 2. 修改虹膜信息：PUT /ISAPI/AccessControl/IrisInfo/modify?format=json。
- 3. 设备不支持虹膜修改。

注：

- (1) 根据虹膜修改中的employeeNo和id字段来判定设备中是否存在该虹膜。

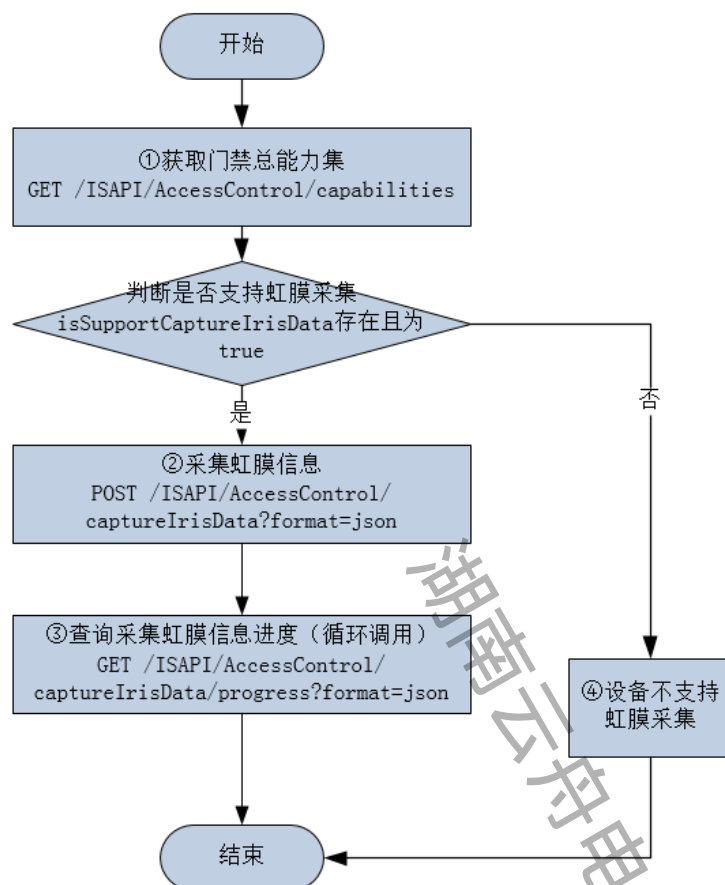
39.2.6 虹膜删除



通过虹膜删除功能，删除设备的虹膜信息。删除不存在的虹膜，设备不报错：

- 1. 判断是否支持虹膜删除：GET /ISAPI/AccessControl/IrisInfo/capabilities?format=json中supportFunction包含delete，代表支持虹膜删除。
- 2. 删除虹膜信息：PUT /ISAPI/AccessControl/IrisInfo/delete?format=json，该接口返回成功后，代表设备已删除成功。
- 3. 设备不支持虹膜删除。

39.2.7 虹膜采集



通过虹膜采集功能，可采集虹膜数据信息：

1. 判断是否支持虹膜采集：GET /ISAPI/AccessControl/capabilities 中返回isSupportCaptureIrisData且为true，代表支持虹膜采集。
2. 采集虹膜信息：POST /ISAPI/AccessControl/captureIrisData?format=json，该接口返回成功后，代表设备已开始执行采集动作。
3. 查询采集虹膜信息进度：GET /ISAPI/AccessControl/captureIrisData/progress?format=json，循环调用该接口，直到captureProgress返回采集成功。
4. 设备不支持虹膜采集。

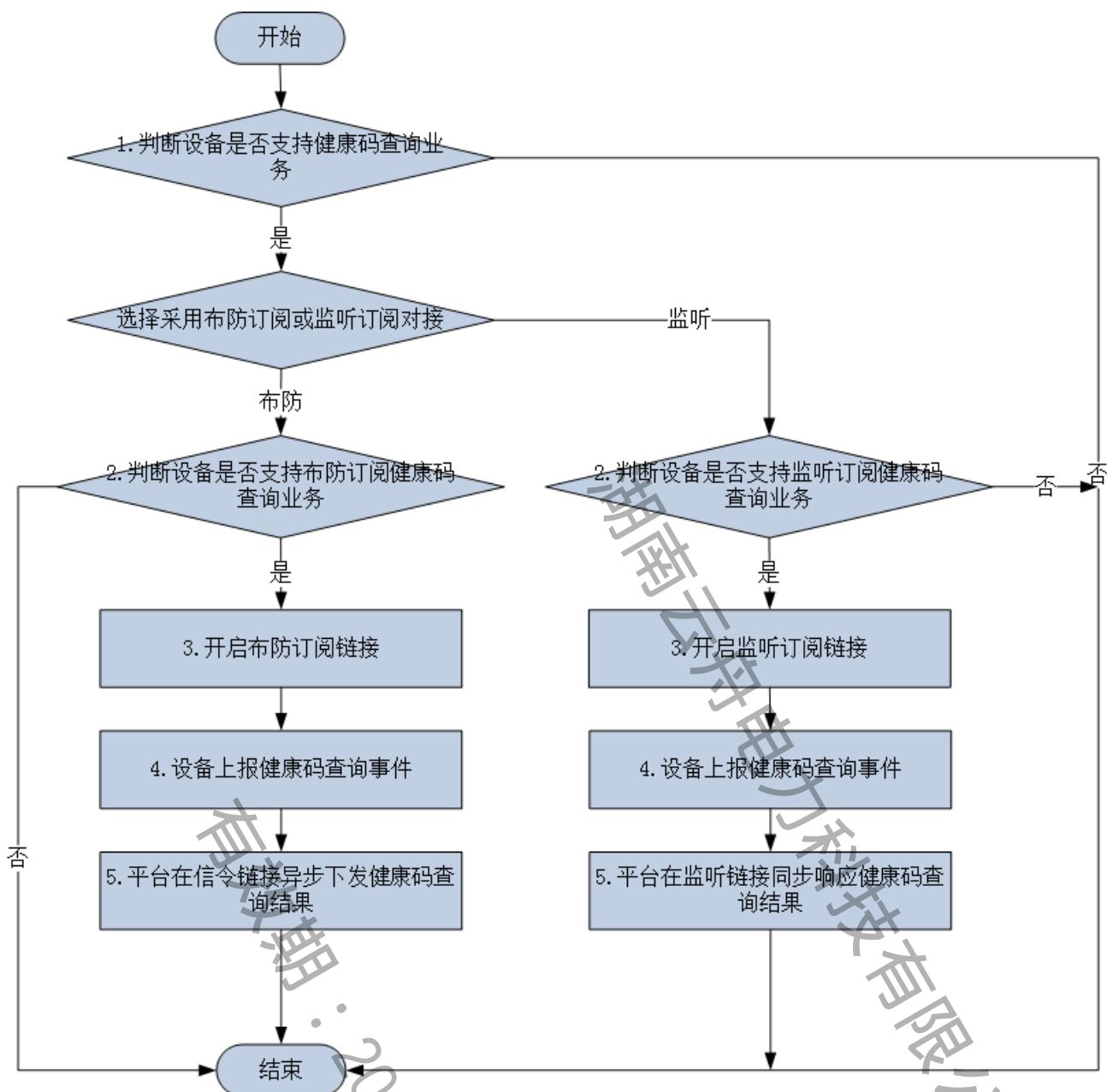
40 健康码查询流程

40.1 功能介绍

监听方式：平台可配置的监听服务器地址并监听订阅（仅支持订阅）健康码查询事件，设备通过http/https监听的方式，提交查询条件到服务器地址，服务器响应查询结果给设备，同时设备将查询结果展示给用户。
布防方式：平台开启布防订阅链接（仅支持订阅），订阅健康码查询事件，设备上报事件后，平台通过ISAPI信令通道下发响应结果给设备，同时设备将查询结果展示给用户。

40.2 集成流程

流程图如下：



ISAPI协议交互流程： 1.判断设备是否支持健康信息查询业务，获取系统能力GET /ISAPI/System/capabilities判断isSupportHealthInfoSyncQuery字段是否为true；若采用布防订阅对接，则步骤2~5如下： 2.判断设备是否支持布防订阅健康信息查询业务，获取布防订阅能力/ISAPI/Event/notification/subscribeEventCap判断type字段是否存在HealthInfoSyncQuery；获取门禁总能力GET /ISAPI/AccessControl/capabilities判断isSupportHealthInfoResult字段是否为true； 3.开启布防订阅链接POST /ISAPI/Event/notification/subscribeEvent?deployID=<deployID>，当订阅事件类型type为HealthInfoSyncQuery或订阅全部事件（eventMode为all）时，设备会上报健康信息查询事件，详见事件布防管理<http://pmp.hikvision.com.cn/#/protocol/infoDetail/91/ISAPI?mdID=ee283eed75294ac48a0412d7d112474f>； 4.设备上报健康信息查询事件eventType: HealthInfoSyncQuery； 5.平台异步下发健康码查询结果/ISAPI/AccessControl/healthInfoResult?format=json，参数能力集见GET /ISAPI/AccessControl/healthInfoResult/capabilities?format=json；若采用监听订阅对接，则步骤2~5如下： 2.判断设备是否支持监听订阅健康信息查询业务，获取监听订阅能力GET /ISAPI/Event/notification/httpHosts/capabilities判断type字段是否存在HealthInfoSyncQuery； 3.平台建立http/https监听连接，当订阅事件类型type为HealthInfoSyncQuery或订阅全部事件（eventMode为all）时，设备会上报健康信息查询事件，详见[http监听管理http://pmp.hikvision.com.cn/#/protocol/infoDetail/91/ISAPI?mdID=214594b629544f1aa36efda3377f76e9](http://pmp.hikvision.com.cn/#/protocol/infoDetail/91/ISAPI?mdID=214594b629544f1aa36efda3377f76e9)； 4.设备上报健康信息查询事件eventType: HealthInfoSyncQuery； 5.平台在监听链接同步响应健康码查询结果，响应报文见4.3.1章节；

40.3 消息示例

40.3.1 健康码查询事件响应结果

```
{
  "checkResult": "success",
  /*req, enum, 核验结果, subType:string, [success#核验成功,failed#核验失败], desc:若平台核验失败,则设备不开门,且上报认证失败事件*/
  "healthCode": 1,
  /*opt, enum, 健康码状态, subType:int, [1#未申领,2#绿码,3#黄码,4#红码,5#无此人员,6#其他错误信息(如接口异常导致查询失败),7#查询健康码超时]*/
  "NADCode": 1,
  /*opt, enum, 核酸检测结果, subType:int, [0#未查询到核酸检测结果,1#核酸检测阴性(代表正常),2#核酸检测阴性(代表确诊),3#核酸检测有效期已过,4#查询核酸结果失败,5#检测结果待复核]*/
  "NADMsg": "test",
  /*opt, string, 核酸检测信息, range:[0,64], desc:空字符串表示查询到核酸检测信息失败*/
  "NADTime": 1,
  /*opt, enum, 核酸检测时间, subType:int, [1#24小时内,2#48小时内,3#48小时外]*/
  "travelCode": 1,
  /*opt, enum, 行程编码, subType:int, [0#14天内一直在当地,1#14天内离开过当地,2#14天内到过疫区,3#其他,4#查询行程信息失败]*/
  "travelInfo": "test",
  /*opt, string, 行程信息, desc:空字符串表示查询到行程信息失败*/
  "vaccineStatus": 1,
  /*opt, enum, 打疫苗状态, subType:int, [0#未打疫苗,1#部分注射疫苗,2#已完成疫苗,3#查询疫苗信息失败,4#已完成加强针疫苗]*/
  "vaccineNum": 1,
  /*opt, int, 打疫苗针数, step:1*/
  "vaccineMsg": "test",
  /*opt, string, 打疫苗信息, range:[0,64], desc:空字符串表示查询到疫苗信息失败*/
  "healthMessage": "test",
  /*opt, string, 自定义健康播报信息, range:[0,15], desc:平台下发该字段后,设备在认证后播报该字段,如“绿码请通行”,该字段优先级高于门禁的自定义语音播报功能*/
  /*
  "ANTCode": 1,
  /*opt, enum, 抗原检测状态, subType:int, [0#未查询到抗原检测结果,1#抗原检测阴性,2#抗原检测阳性,3#抗原检测有效期已过,4#抗原检测无效]*/
  "ANTMsg": "test",
  /*opt, string, 抗原检测信息, range:[0,64], desc:空字符串表示查询到抗原检测信息失败*/
  "selfDefineInfo": "test",
  /*opt, string, 自定义信息, range:[0,128], desc:自定义信息,可填写为场所信息、省份信息等自定义内容*/
  "IDNum": "test",
  /*opt, string, 身份证号, range:[0,18]*/
  "name": "test",
  /*opt, string, 人员姓名, range:[0,128]*/
  "faceID": 1
  /*opt, int, 人脸抓拍识别事件中的人脸ID, desc:desc:此处人脸ID与人脸抓拍识别事件(faceCapture)中的faceId一致,一体机上的明眸设备直接上传身份证信息到平台,一体机上传人脸抓拍识别事件(faceCapture)到平台。平台通过证件照和人脸图片比对,关联人脸ID和健康码信息,并在下发健康码信息时需要将人脸ID发给一体机,用于一体机进行关联*/
}
}
```

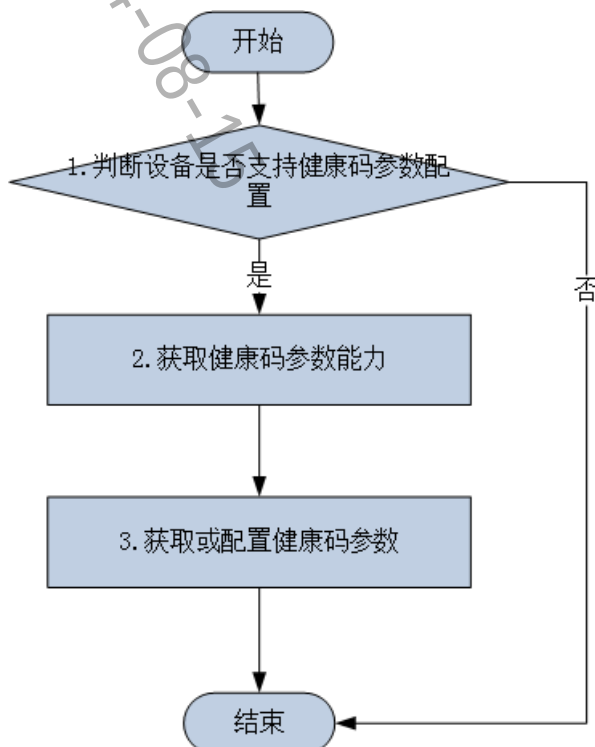
41 健康信息参数配置

41.1 功能介绍

控制设备是否开启健康信息查询功能,设备默认关闭

41.2 集成流程

流程图如下:



ISAPI协议交互流程： 1.判断设备是否支持健康信息参数配置，获取门禁总能力GET
/ISAPI/AccessControl/capabilities判断isSupportHealthCodeCfg字段是否为true； 2.获取健康码参数能力， GET
/ISAPI/AccessControl/healthCodeCfg/capabilities?format=json； 3.获取及配置健康码参数， GET/PUT
/ISAPI/AccessControl/healthCodeCfg?format=json；

42 设备内反潜回参数配置

42.1 功能介绍

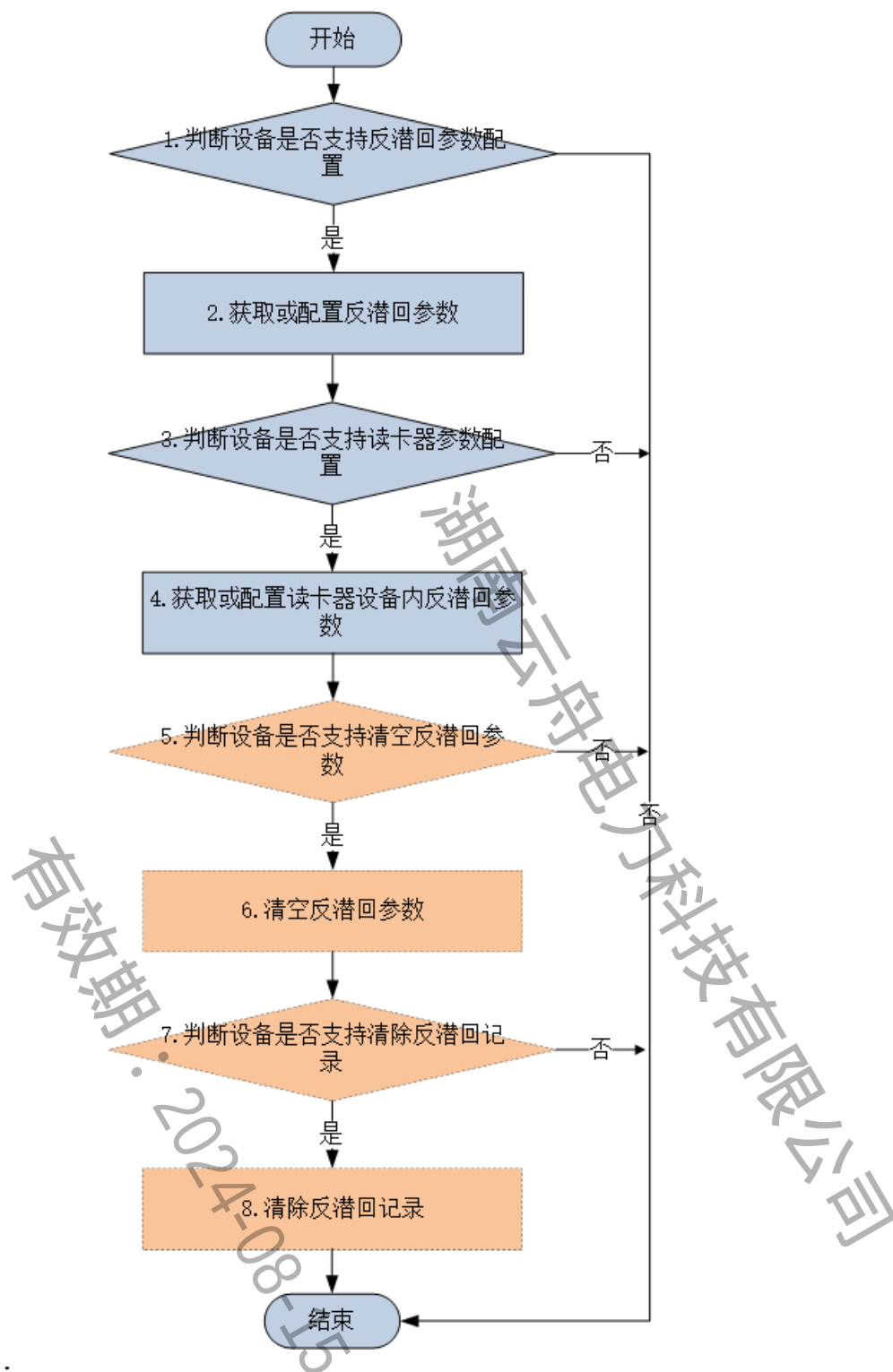
反潜回是一种让持卡人（人脸、指纹等在读卡器支持的情况下同样适用）按照指定区域路线进出的功能。在一个区域，有多个出入口，要求持卡人在进出门时必须在指定门禁点进行刷卡，且限定刷卡次数一次，即持卡者从某个门刷卡进来也需要从某个门刷卡才可以出去，刷卡记录必须一进一出严格对应。单机反潜回可防止一卡多人使用或人员多次进入/退出的问题，主要应用于展会、景点、地铁出入口等出入口相对较少一人一卡的场景。

42.2 集成流程

流程图如下：

有效期：2024-08-15

湖南云舟电力科技有限公司



ISAPI协议交互流程：

1. 判断设备是否支持反潜回参数配置，获取门禁总能力GET /ISAPI/AccessControl/capabilities判断isSupportAntiSneakCfg字段是否为true;
2. 获取反潜回参数，GET /ISAPI/AccessControl/AntiSneakCfg?format=json; 配置反潜回参数，PUT /ISAPI/AccessControl/AntiSneakCfg?format=json，可配置使能项开启反潜回功能，同时可配置反潜回初始读卡器，即人员进门的初始入口，参数能力见GET /ISAPI/AccessControl/AntiSneakCfg/capabilities?format=json;
3. 判断设备是否支持读卡器参数配置，获取门禁总能力GET /ISAPI/AccessControl/capabilities判断isSupportCardReaderAntiSneakCfg字段是否为true;
4. 获取读卡器设备内反潜回参数，GET /ISAPI/AccessControl/CardReaderAntiSneakCfg/<cardReaderID>?format=json; 配置读卡器设备内反潜回参数，PUT /ISAPI/AccessControl/CardReaderAntiSneakCfg/<cardReaderID>?format=json，URL中cardReaderID为读卡器编号，持卡人的进出门路线通过此协议配置读卡器参数中的后续读卡器编号方式确定，这里需要注意的是需将反潜回路线配置成一个环路，否则当一个线路认证完成后再从起点开始认证会失败。举例：读卡器1->读卡器2->读卡器1

卡器3，必须要降读卡器3的后续读卡器配置成读卡器1，否则，当读卡器3认证完之后，再认证读卡器1会失败。

参数能力见GET /ISAPI/AccessControl/CardReaderAntiSneakCfg/capabilities?format=json;

- 判断设备是否支持清空反潜回参数，获取门禁总能力GET /ISAPI/AccessControl/capabilities判断isSupportClearAntiSneakCfg字段是否为true;
- 清空反潜回参数，PUT /ISAPI/AccessControl/ClearAntiSneakCfg?format=json，通过配置反潜回参数为false使设备反潜回功能失效，参数能力见GET /ISAPI/AccessControl/ClearAntiSneakCfg/capabilities?format=json;
- 判断设备是否支持清除反潜回记录，获取门禁总能力GET /ISAPI/AccessControl/capabilities判断isSupportClearAntiSneak字段是否为true;
- 清除反潜回记录，PUT /ISAPI/AccessControl/ClearAntiSneak?format=json，反潜回记录清除支持按人员ID清除，同时支持清除全部的操作，参数能力见GET /ISAPI/AccessControl/ClearAntiSneak/capabilities?format=json。

注：配置之前先清空之前的反潜回配置参数，否则可能有脏数据影响（下述配置协议下发后设备同样需执行清空操作）。

43 设备内反潜回规则重置（附加功能）

43.1 功能介绍

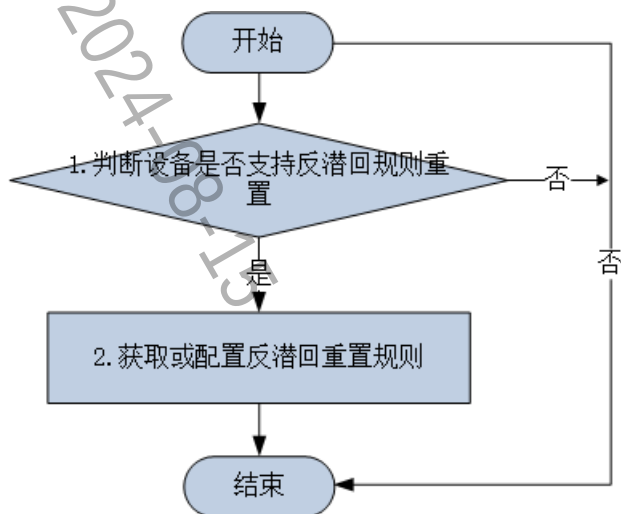
可重置的反潜回规则：

- 按认证时长重置模式。触发反潜的规定时长内，反潜生效，若某一用户按路线刷卡后，在到时间后，自动重置该用户的反潜回标志；
- 按时间点重置模式。到时间点时，自动重置所有用户反潜回标志；
- 无效模式。即不启用重置规则。

使用场景：在无保安的情况下，可以灵活解决一些反潜问题，此功能需设置使用，时间段反潜和时间点反潜互斥，使用者可设置规定时长和规定时间点。

43.2 集成流程

流程图如下：



ISAPI协议交互流程：

- 判断设备是否支持设备内反潜回重置规则功能，获取门禁总能力GET /ISAPI/AccessControl/capabilities判断isSupportAntiPassbackResetRules字段是否为true;
- 获取反潜回重置规则，GET /ISAPI/AccessControl/AntiPassback/resetRules?format=json;配置反潜回重置规则，PUT /ISAPI/AccessControl/AntiPassback/resetRules?format=json，可配置重置模式，参数能力见GET /ISAPI/AccessControl/AntiPassback/resetRules/capabilities?format=json;

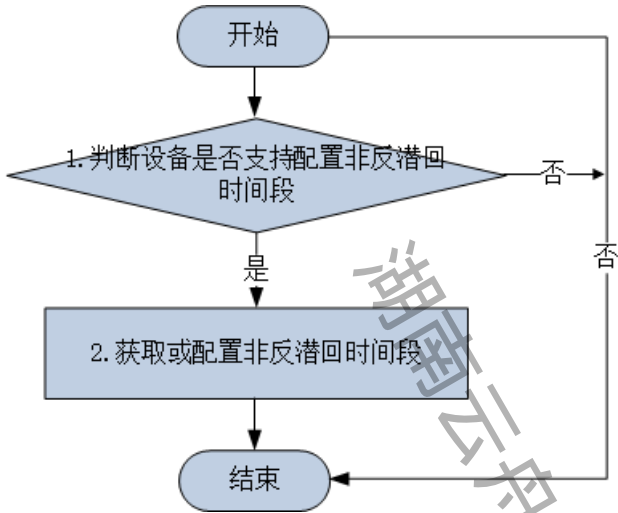
44 非反潜回时间段配置（附加功能）

非反潜时间段：在规定的时段内，不触发反潜。使用场景： 在一些出入高峰期，刷卡人员会跟随其他人员出入，反潜情况经常发生，用户希望在高峰期，反潜失效，以便正常的进出。

44.1 功能介绍

44.2 集成流程

流程图如下：



ISAPI协议交互流程：

1. 判断设备是否支持设备内反潜回时间段功能，获取门禁总能力GET /ISAPI/AccessControl/capabilities判断 isSupportAntiPassbackTimeRange 字段是否为true；
2. 获取非反潜回时间段参数，GET /ISAPI/AccessControl/AntiPassback/timeRange?format=json；配置非反潜回时间段参数，PUT /ISAPI/AccessControl/AntiPassback/timeRange?format=json；通过配置时间段的方式使反潜回功能失效，参数能力见GET /ISAPI/AccessControl/AntiPassback/timeRange/capabilities?format=json；

45 考勤模式配置

45.1 功能描述

设备支持四种考勤模式配置：自动、手动、自动和手动，禁用，不同考勤模式，考勤结果显示和考勤数据不同。

45.2 集成流程

1. 调用门禁总能力GET /ISAPI/AccessControl/capabilities，通过isSupportAttendanceMode字段返回为true，判断设备支持考勤模式配置功能；
2. 调用GET /ISAPI/AccessControl/Configuration/attendanceMode/capabilities?format=json，获取考勤模式配置功能设备支持的详细参数规格；
3. 调用GET|PUT /ISAPI/AccessControl/Configuration/attendanceMode?format=json，进行考勤模式的获取和配置。

46 按键考勤参数配置

46.1 功能描述

设备支持在自动、手动、自动和手动模式下对单个按键进行配置，包含考勤状态和自定义名称。且按键序号1~6，与checkIn-上班，checkOut-下班，breakOut-开始休息，breakIn-结束休息，overtimeIn-开始加班，overtimeOut-结束加班六种考勤状态严格一一对应，若未对应设备需报错。

46.2 集成流程

1. 调用门禁总能力GET /ISAPI/AccessControl/capabilities，通过isSupportKeyCfgAttendance字段返回为true，判断设备支持按键考勤参数配置功能；
2. 调用GET /ISAPI/AccessControl/keyCfg/attendance/capabilities?format=json，获取按键考勤参数配置功能设备支持的详细参数规格；
3. 调用GET /ISAPI/AccessControl/keyCfg/attendance?format=json，获取所有按键考勤参数；调用GET|PUT /ISAPI/AccessControl/keyCfg/<keyID>/attendance?format=json，进行单个按键考勤参数的获取和配置。

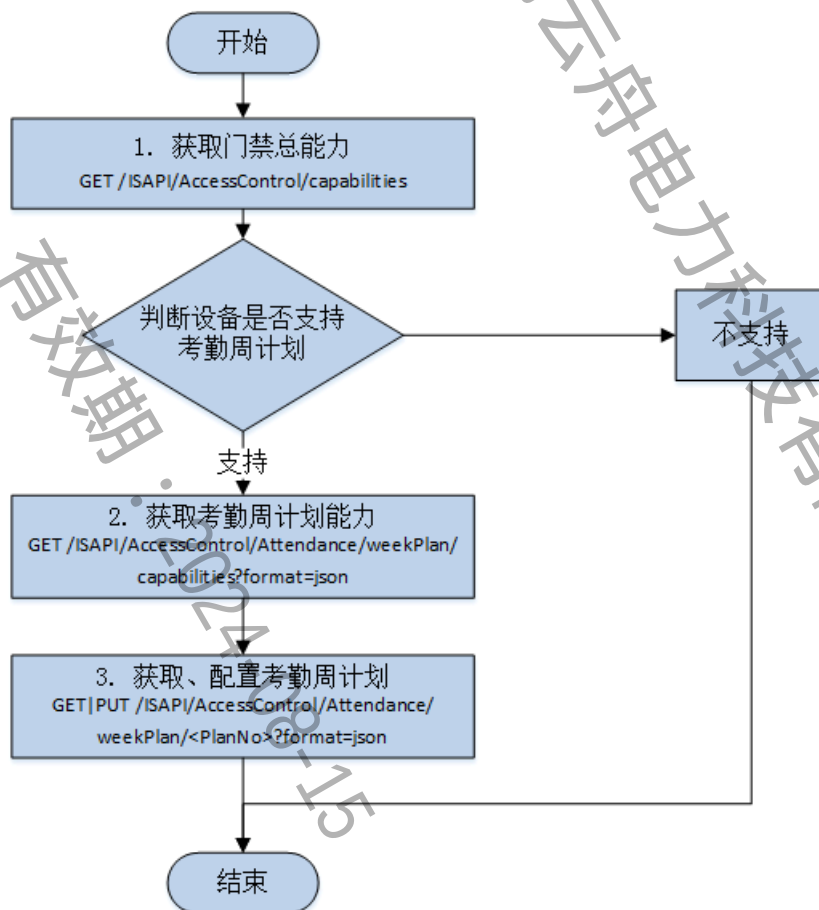
47 考勤计划配置

47.1 功能描述

设备支持按键考勤周计划和计划模板的配置。单个周计划中每天最多支持8个时间段，一周总共56个时间段。上下班、休息、加班每种考勤属性计划模板只能配置1个，且需绑定1个周计划。

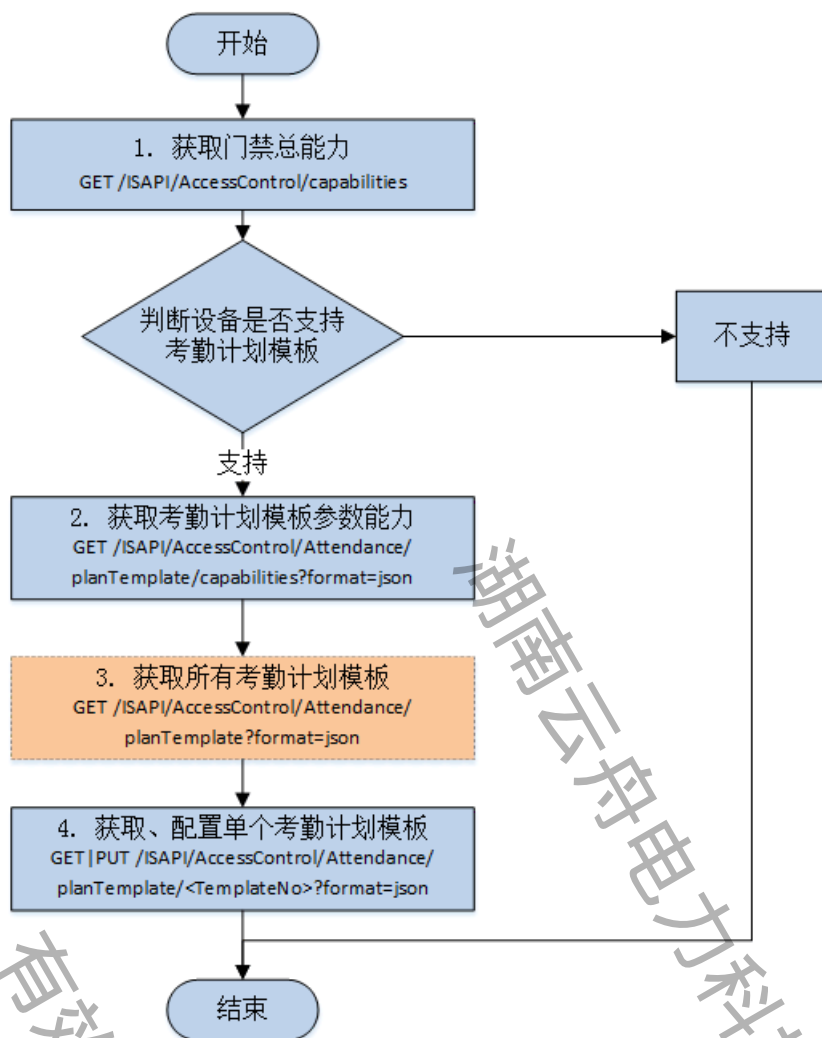
47.2 集成流程

47.2.1 考勤周计划



1. 调用门禁总能力GET /ISAPI/AccessControl/capabilities，通过isSupportAttendanceWeekPlan字段返回为true，判断设备支持考勤周计划配置功能；
2. 调用GET /ISAPI/AccessControl/Attendance/weekPlan/capabilities?format=json，获取考勤周计划配置功能设备支持的详细参数规格；
3. 调用GET|PUT /ISAPI/AccessControl/Attendance/weekPlan/<PlanNo>?format=json，进行考勤周计划的获取和配置。

47.2.2 考勤计划模板



1. 调用门禁总能力GET /ISAPI/AccessControl/capabilities，通过isSupportAttendancePlanTemplate字段返回为true，判断设备支持考勤计划模板配置功能；
2. 调用GET /ISAPI/AccessControl/Attendance/planTemplate/capabilities?format=json，获取考勤计划模板 配置功能设备支持的详细参数规格；
3. 调用GET /ISAPI/AccessControl/Attendance/planTemplate?format=json，获取所有考勤计划模板；
4. 调用GET|PUT /ISAPI/AccessControl/Attendance/planTemplate/<TemplateNo>?format=json，进行单个考勤计划模板的获取和配置。

47.2.3 清空考勤计划

1. 调用门禁总能力GET /ISAPI/AccessControl/capabilities，通过isSupportClearAttendancePlan字段返回为true，判断设备支持清空考勤计划功能；
2. 调用PUT /ISAPI/AccessControl/ClearAttendancePlan?format=json，清空考勤计划。

48 多重认证

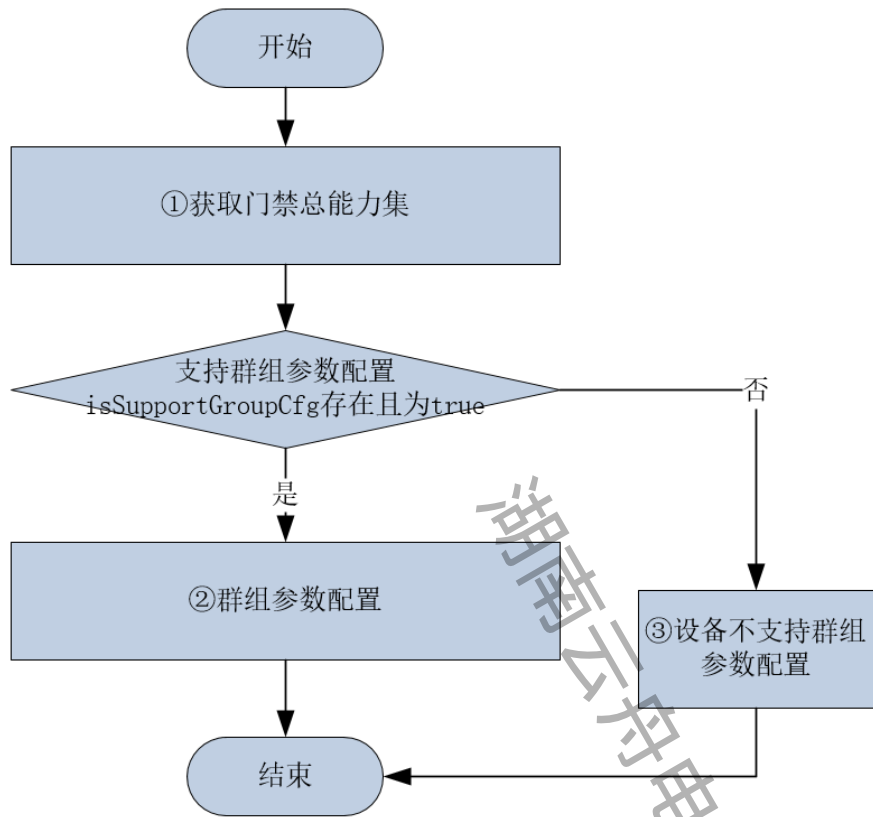
48.1 功能介绍

在一些安全等级较高的场景下，可以给对应的门禁点设置多个不同人员在固定时间内认证成功后门才能被打开。

比如在银行场景中，一般开门都是需要两个或者两个以上的工作人员进行凭证认证（如刷卡/指纹/人脸/虹膜等）后才能开门。即，配置了多重认证的门，需要不同群组内的人员按照配置的人员数量及认证顺序进行认证后才能开门。

48.2 集成流程

48.2.1 群组参数配置



集成流程如下：

- 1. 判断是否支持群组参数配置：GET /ISAPI/AccessControl/capabilities 中isSupportGroupCfg存在且为true，代表支持群组参数配置。
- 2. 群组参数配置：[GET/PUT] /ISAPI/AccessControl/GroupCfg/<groupID>?format=json，配置群组参数。
- 3. 设备不支持群组参数配置。

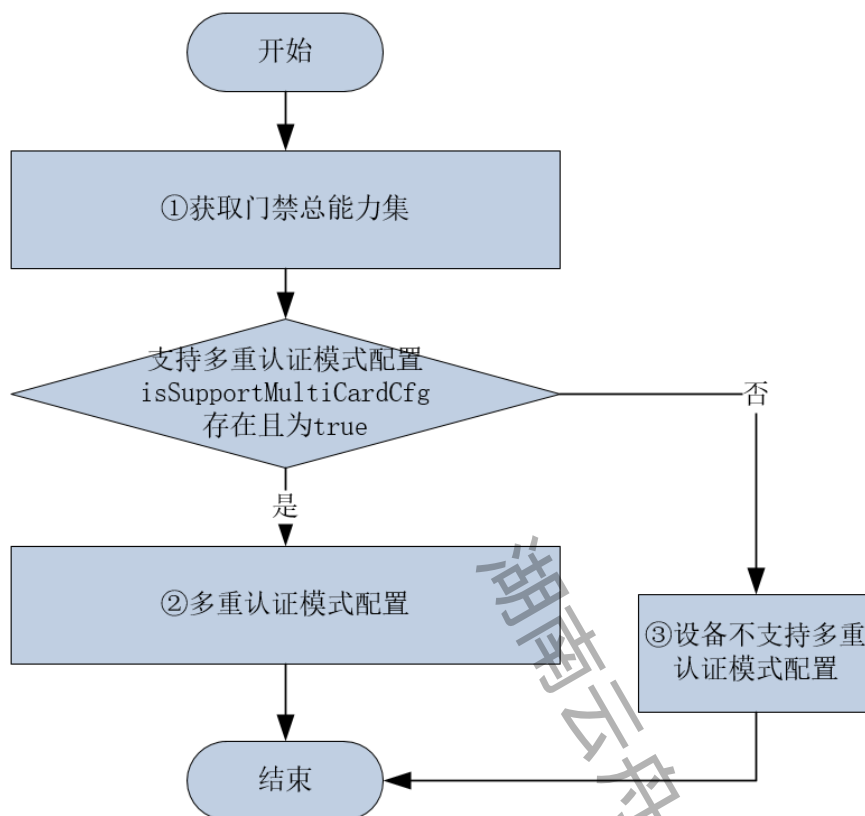
48.2.2 人员添加至群组

配置完群组参数后，通过人员及凭证管理集成流程中人员管理将人员与群组进行关联（通过belongGroup字段），对应接口如下：

- 1. 人员设置：PUT /ISAPI/AccessControl/UserInfo/SetUp?format=json
- 2. 人员新增：POST /ISAPI/AccessControl/UserInfo/Record?format=json
- 3. 人员修改：PUT /ISAPI/AccessControl/UserInfo/Modify?format=json

其中一个人最多可以关联4个群组。

48.2.3 多重认证模式配置



集成流程如下：

1. 判断是否支持多重认证模式配置：`GET /ISAPI/AccessControl/capabilities`中isSupportMultiCardCfg存在且为true，代表支持多重认证模式配置（如支持该配置，则设备一定支持群组参数配置）。
2. 多重认证模式配置：`[GET/PUT] /ISAPI/AccessControl/MultiCardCfg/<doorID>?format=json`，配置多重认证模式。
3. 设备不支持多重认证模式配置。

49 门禁事件存储与查询

49.1 功能描述

设备支持配置门禁事件存储参数、查询门禁事件以及查询门禁事件条数。门禁事件存储支持定期删除旧事件、按指定日期删除旧事件以及循环覆盖三种存储模式。

49.2 集成流程

49.2.1 门禁事件存储参数配置

1. 调用门禁总能力`GET /ISAPI/AccessControl/capabilities`，通过isSupportEventStorageCfg字段返回为true，判断设备支持门禁事件存储参数配置功能；
2. 调用`GET /ISAPI/AccessControl/AcsEvent/StorageCfg/capabilities?format=json`，获取门禁事件存储参数配置功能设备支持的详细参数规格；
3. 调用`GET|PUT /ISAPI/AccessControl/AcsEvent/StorageCfg?format=json`，进行门禁事件存储参数的获取和配置。

49.2.2 查询门禁事件

1. 调用门禁总能力`GET /ISAPI/AccessControl/capabilities`，通过isSupportAcsEvent字段返回为true，判断设备支持查询门禁事件功能；
2. 调用`GET /ISAPI/AccessControl/AcsEvent/capabilities?format=json`，获取查询门禁事件功能设备支持的详细参数规格；

3. 调用`POST /ISAPI/AccessControl/AcsEvent?format=json` 查询门禁事件

3. 调用 `POST /ISAPI/AccessControl/AcsEvent?format=json`，查询门禁事件。

49.2.3 查询门禁事件条数

1. 调用门禁总能力 `GET /ISAPI/AccessControl/capabilities`，通过 `isSupportAcsEventTotalNum` 字段返回为 `true`，判断设备支持查询门禁事件条数功能；
2. 调用 `GET /ISAPI/AccessControl/AcsEventTotalNum/capabilities?format=json`，获取查询门禁事件条数功能设备支持的详细参数规格；
3. 调用 `POST /ISAPI/AccessControl/AcsEventTotalNum?format=json`，查询门禁事件条数。

50 事件及卡号联动管理

50.1 功能描述

设备支持在触发某种操作时关联具体的联动动作，联动方式支持事件、卡号、MAC地址、工号四种。应用场景举例：明眸设备，人员刷脸认证成功，可以开门。

50.2 集成流程

50.2.1 事件及卡号联动参数配置

1. 调用门禁总能力 `GET /ISAPI/AccessControl/capabilities`，通过 `isSupportEventCardLinkageCfg` 字段返回为 `true`，判断设备支持事件及卡号联动参数配置功能；
2. 调用 `GET /ISAPI/AccessControl/EventCardLinkageCfg/capabilities?format=json`，获取事件及卡号联动参数配置功能设备支持的详细参数规格；
3. 调用 `GET|PUT /ISAPI/AccessControl/EventCardLinkageCfg/<ACEID>?format=json`，进行事件及卡号联动参数的获取和配置。

50.2.2 查询事件及卡号联动参数

1. 调用门禁总能力 `GET /ISAPI/AccessControl/capabilities`，通过 `isSupportEventCardLinkageCfgSearch` 字段返回为 `true`，判断设备支持查询事件及卡号联动参数功能；
2. 调用 `GET /ISAPI/AccessControl/EventCardLinkageCfg/search/capabilities?format=json`，获取查询事件及卡号联动参数功能设备支持的详细参数规格；
3. 调用 `POST /ISAPI/AccessControl/EventCardLinkageCfg/search?format=json`，查询事件及卡号联动参数。

50.2.3 删除指定事件及卡号联动参数

1. 调用门禁总能力 `GET /ISAPI/AccessControl/capabilities`，通过 `isSupportEventCardLinkageCfgDelete` 字段返回为 `true`，判断设备支持删除指定事件及卡号联动参数功能；
2. 调用 `PUT /ISAPI/AccessControl/EventCardLinkageCfgDelete?format=json`，删除指定事件及卡号联动参数。

50.2.4 获取事件及卡号联动编号列表

1. 调用门禁总能力 `GET /ISAPI/AccessControl/capabilities`，通过 `isSupportEventCardNoList` 字段返回为 `true`，判断设备支持获取事件及卡号联动编号列表功能；
2. 调用 `GET /ISAPI/AccessControl/EventCardNoList/capabilities?format=json`，获取事件及卡号联动编号列表功能设备支持的详细参数规格；
3. 调用 `GET /ISAPI/AccessControl/EventCardNoList?format=json`，获取事件及卡号联动编号列表。

50.2.5 事件优化

- 1. 调用门禁总能力GET /ISAPI/AccessControl/capabilities，通过isSupportEventOptimizationCfg字段返回为true，判断设备支持事件优化功能；
- 2. 调用GET /ISAPI/AccessControl/EventOptimizationCfg/capabilities?format=json，获取事件优化功能设备支持的详细参数规格；
- 3. 调用GET|PUT /ISAPI/AccessControl/EventOptimizationCfg?format=json，获取和配置事件优化参数。

51 布防信息

51.1 功能描述

设备支持获取当前已布防信息，包含布防侧IP和端口、布防类型、协议类型等信息。

51.2 集成流程

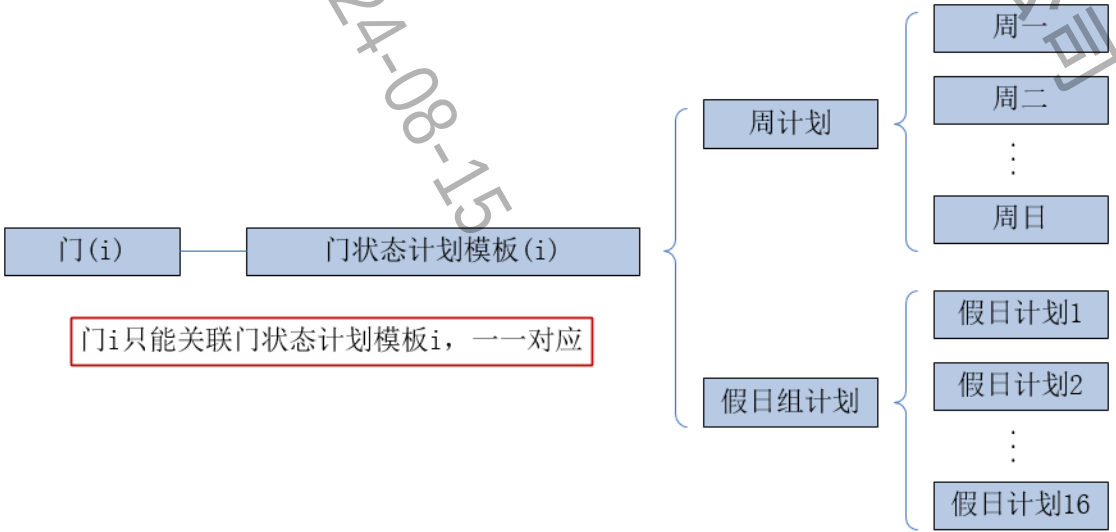
- 1. 调用门禁总能力GET /ISAPI/AccessControl/capabilities，通过isSupportDeployInfo字段返回为true，判断设备支持获取布防信息功能；
- 2. 调用GET /ISAPI/AccessControl/DeployInfo/capabilities，获取布防信息功能设备支持的详细参数规格；
- 3. 调用GET /ISAPI/AccessControl/DeployInfo，获取布防信息。

52 门状态计划管理

52.1 功能介绍

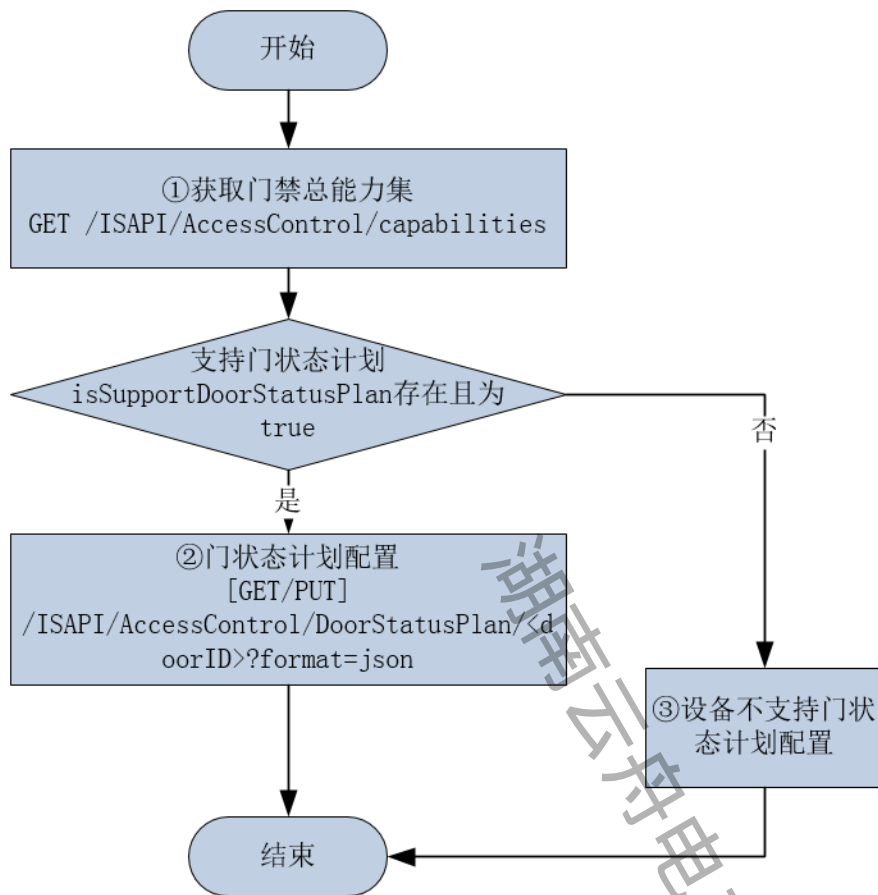
对门禁控制的门或者梯控的楼层，配置关联计划模板，可以配置计划时间段和状态：常开、常闭、休眠、普通状态。门状态计划配置不是必须的，没有配置设备默认所有门都有权限。远程控门的优先级高于门状态计划的优先级，即当门处于常开/常闭/休眠/普通状态时，远程控门操作均能生效。

每一个计划模板可以关联一个周计划和四个假日组计划，假日计划的优先级高于周计划。周计划可以配置周一到周日的时段，每天支持配置8个不同的时段；假日组计划可以关联16个不同的假日计划，每一个假日计划可以配置一个假日的起止日期，日期范围内每天的时段相同（可以配置不同的8个时段）。通过这个计划模板配置，可以实现门的权限管理。



52.2 集成流程

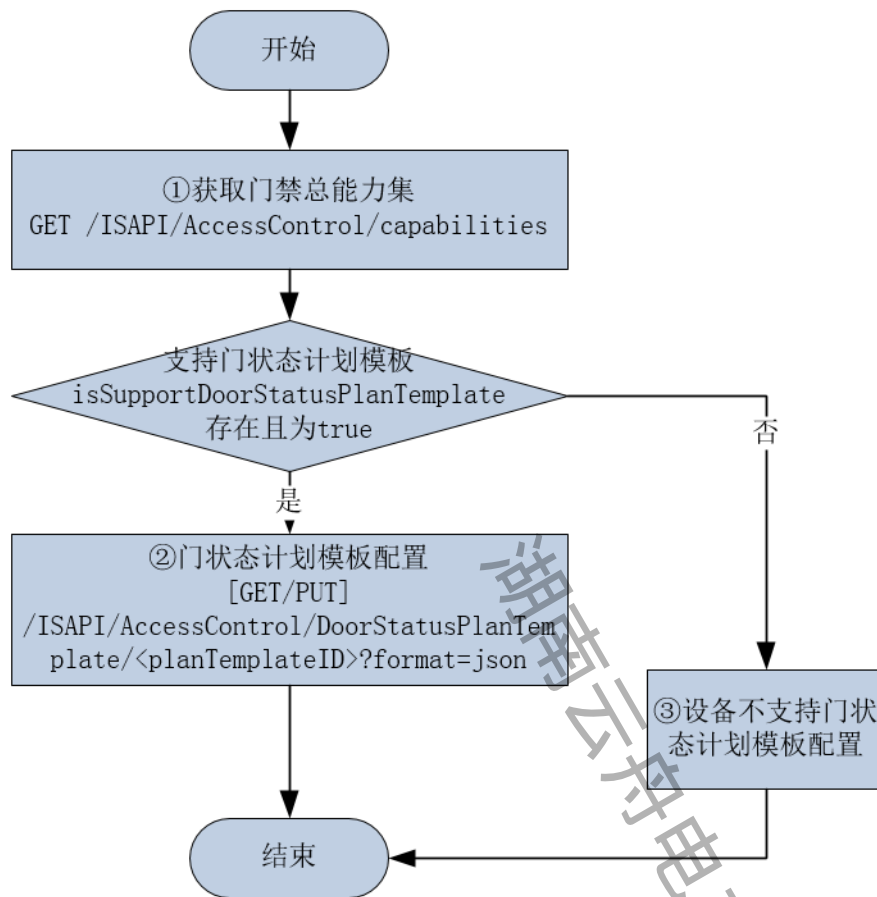
52.2.1 门状态计划配置



集成流程如下：

1. 判断是否支持门状态计划配置：GET /ISAPI/AccessControl/capabilities中isSupportDoorStatusPlan存在且为true，代表支持门状态计划配置（如支持该配置，则设备一定支持门状态计划模板配置）。
2. 门状态计划配置：[GET/PUT] /ISAPI/AccessControl/DoorStatusPlan/<doorID>?format=json，配置门状态计划。
3. 设备不支持门状态计划配置。

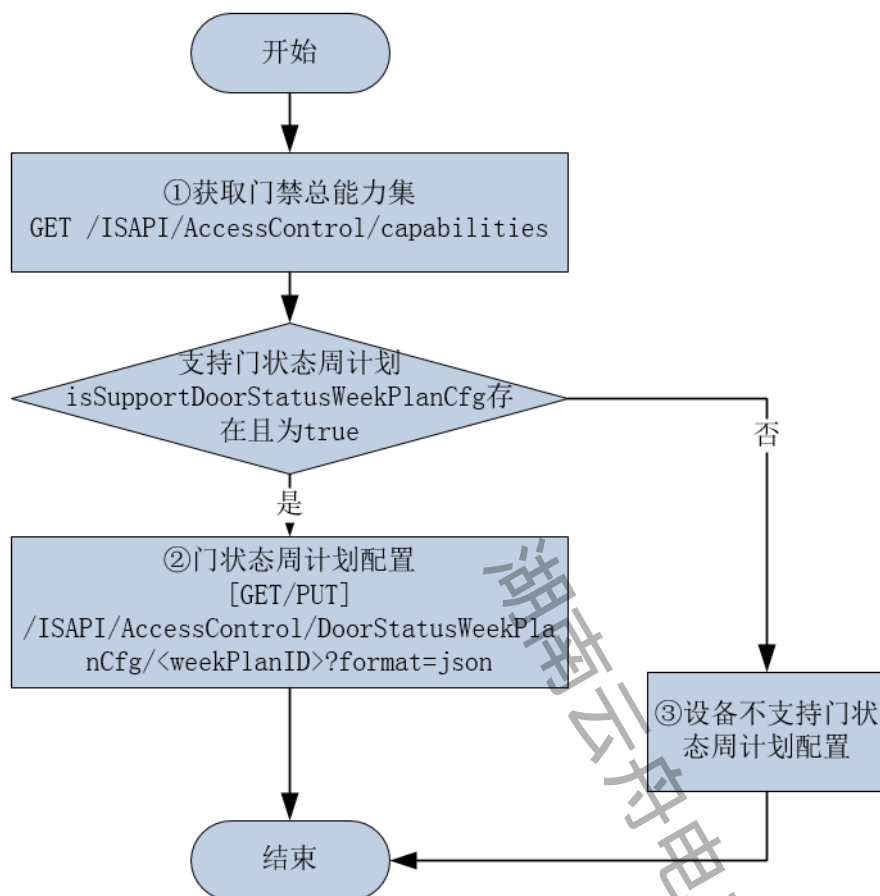
52.2.2 门状态计划模板配置



集成流程如下：

1. 判断是否支持门状态计划模板配置：GET /ISAPI/AccessControl/capabilities中 isSupportDoorStatusPlanTemplate存在且为true，代表支持门状态计划模板配置（如支持该配置，则设备一定支持门状态周计划配置）。
2. 门状态计划模板配置：[GET/PUT] /ISAPI/AccessControl/DoorStatusPlanTemplate/<planTemplateID>?format=json，配置门状态计划模板。
3. 设备不支持门状态计划模板配置。

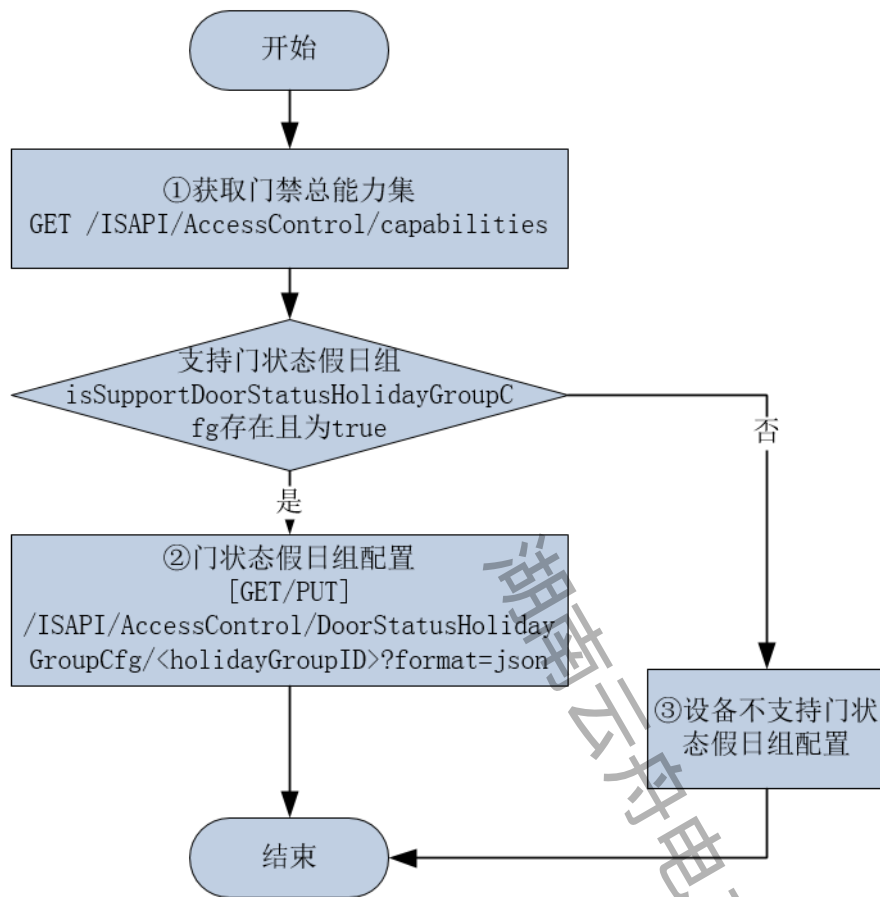
52.2.3 门状态周计划配置



集成流程如下：

1. 判断是否支持门状态周计划配置：GET /ISAPI/AccessControl/capabilities中isSupportDoorStatusWeekPlanCfg存在且为true，代表支持门状态周计划配置。
2. 门状态周计划配置：[GET/PUT] /ISAPI/AccessControl/DoorStatusWeekPlanCfg/<weekPlanID>?format=json，配置门状态周计划。
3. 设备不支持门状态周计划配置。

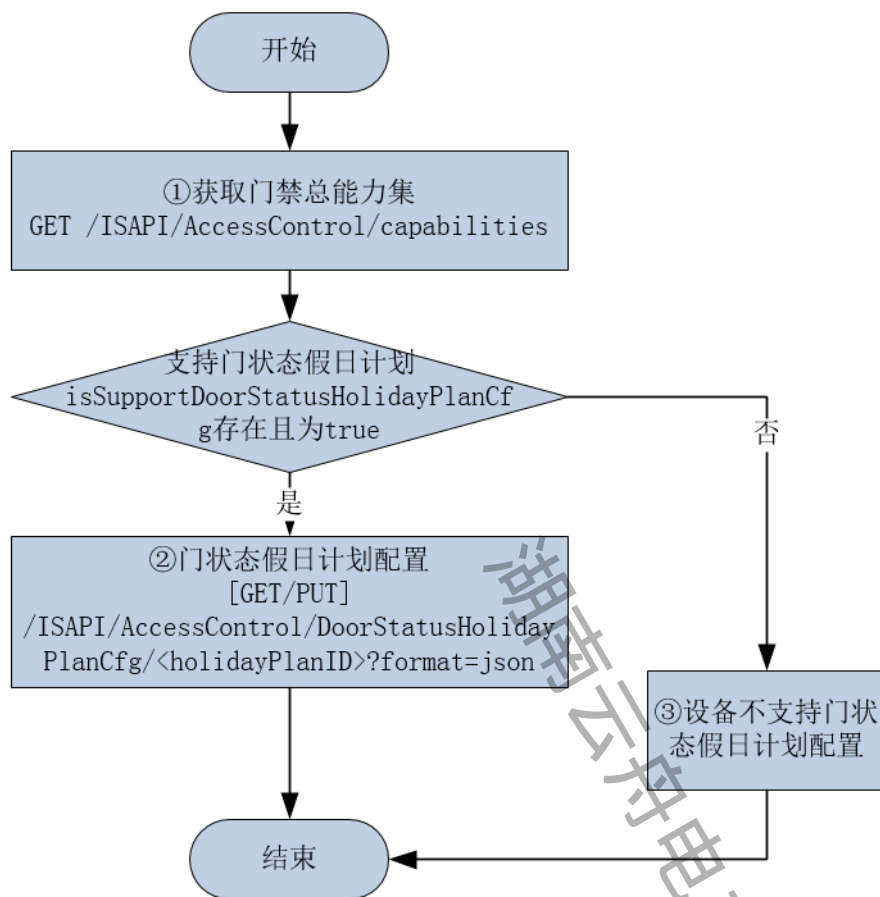
52.2.4 门状态假日组配置



集成流程如下：

1. 判断是否支持门状态假日组配置：GET /ISAPI/AccessControl/capabilities中 isSupportDoorStatusHolidayGroupCfg存在且为true，代表支持门状态假日组配置（如支持该配置，则设备一定支持门状态假日计划配置）。
2. 门状态假日组配置：[GET/PUT] /ISAPI/AccessControl/DoorStatusHolidayGroupCfg/<holidayGroupID>?format=json，配置门状态假日组。
3. 设备不支持门状态假日组配置。

52.2.5 门状态假日计划配置



集成流程如下：

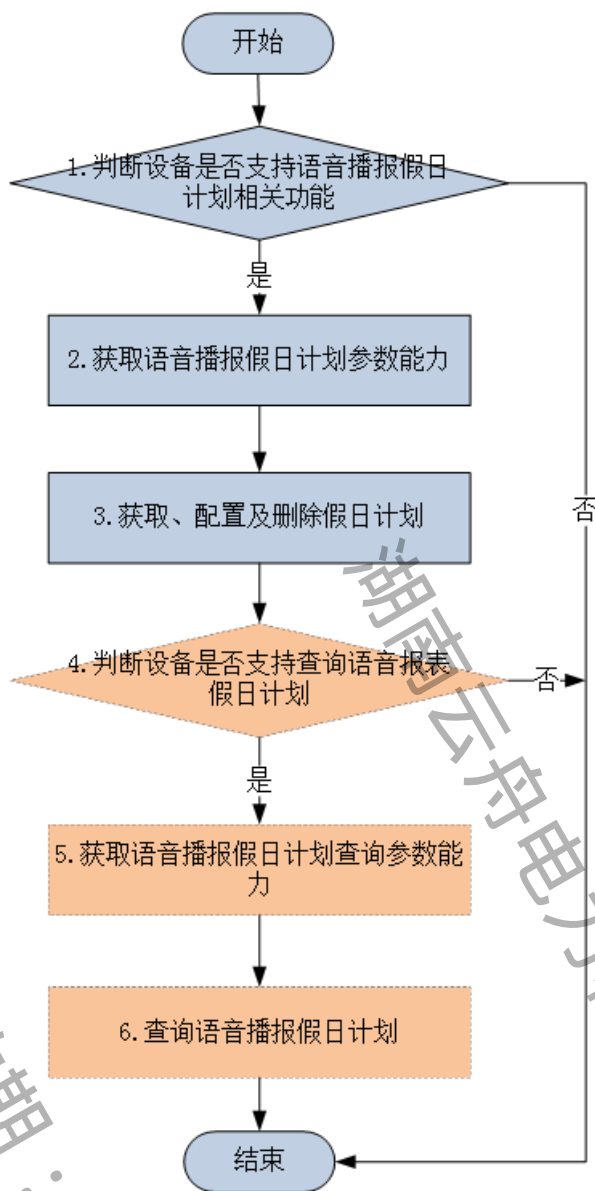
1. 判断是否支持门状态假日计划配置：GET /ISAPI/AccessControl/capabilities中 isSupportDoorStatusHolidayPlanCfg存在且为true，代表支持门状态假日计划配置。
2. 门状态假日计划配置：[GET/PUT] /ISAPI/AccessControl/DoorStatusHolidayPlanCfg/<holidayPlanID>?format=json，配置门状态假日计划。
3. 设备不支持门状态假日计划配置。

53 语音播报假日计划管理流程

53.1 功能介绍

业务集成方可配置在假日时段，设备发生认证成功或失败时，所播放的TTS语音文本或音频文件。

53.2 集成流程



ISAPI协议交互流程： 1.判断设备是否支持语音播报假日计划功能，获取门禁总能力GET /ISAPI/AccessControl/capabilities?format=json判断isSupportTTSTextHolidayPlan，字段是否为true； 2.获取语音播报假日计划参数能力GET /ISAPI/AccessControl/Verification/ttsText/holidayPlan/capabilities?format=json； 3.获取语音播报假日计划GET /ISAPI/AccessControl/Verification/ttsText/holidayPlan/<holidayPlanID>?format=json；配置语音播报假日计划PUT /ISAPI/AccessControl/Verification/ttsText/holidayPlan/<holidayPlanID>?format=json；删除语音播报假日计划DELETE /ISAPI/AccessControl/Verification/ttsText/holidayPlan/<holidayPlanID>?format=json；4.判断设备是否支持查询语音播报假日计划，获取门禁总能力GET /ISAPI/AccessControl/capabilities?format=json判断isSupportTTSTextSearchHolidayPlan，字段是否为true； 5.获取查询语音播报假日计划参数能力GET /ISAPI/AccessControl/Verification/ttsText/searchHolidayPlan/capabilities?format=json； 6.查询语音播报假日计划POST /ISAPI/AccessControl/Verification/ttsText/searchHolidayPlan?format=json；

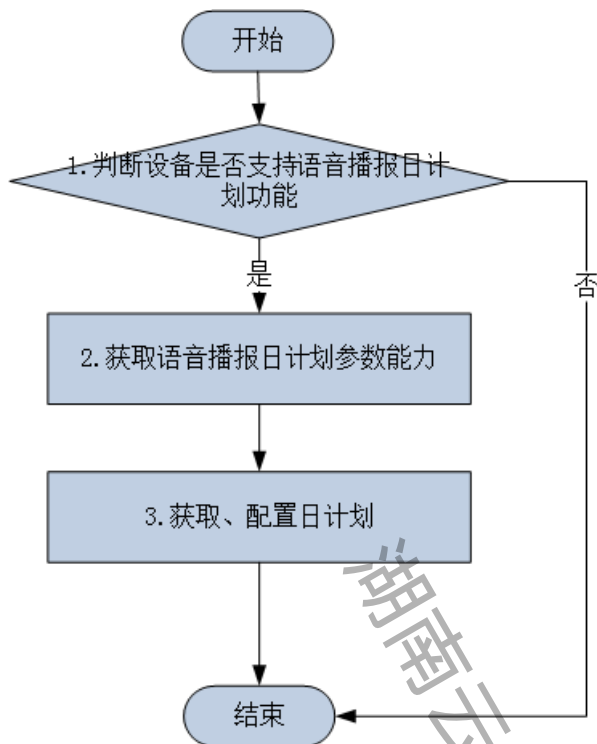
54 语音播报日计划管理流程

54.1 功能介绍

业务集成方可配置在假日时段，设备发生认证成功或失败时，所播放的TTS语音文本或音频文件。当设备在播放时假日计划的优先级高于日计划，当前时段属于假日计划则播放假日计划的音频，当前时段无假日计划则播放日计划的音频。

54.2 集成流程

流程图如下：



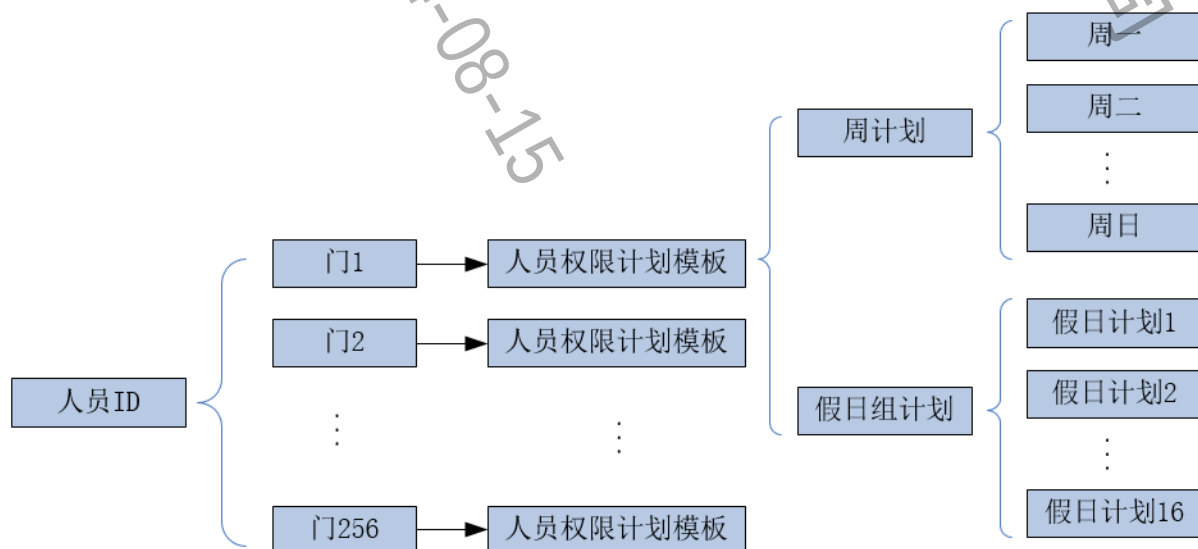
ISAPI协议交互流程： 1.判断设备是否支持语音播报日计划管理功能，获取门禁总能力GET /ISAPI/AccessControl/capabilities?format=json判断isSupportTTSText字段是否为true； 2.获取语音播报日计划参数能力GET /ISAPI/AccessControl/Verification/ttsText/capabilities?format=json； 3.获取及配置语音播报日计划GET/PUT /ISAPI/AccessControl/Verification/ttsText?format=json；

55 人员权限计划管理

55.1 功能介绍

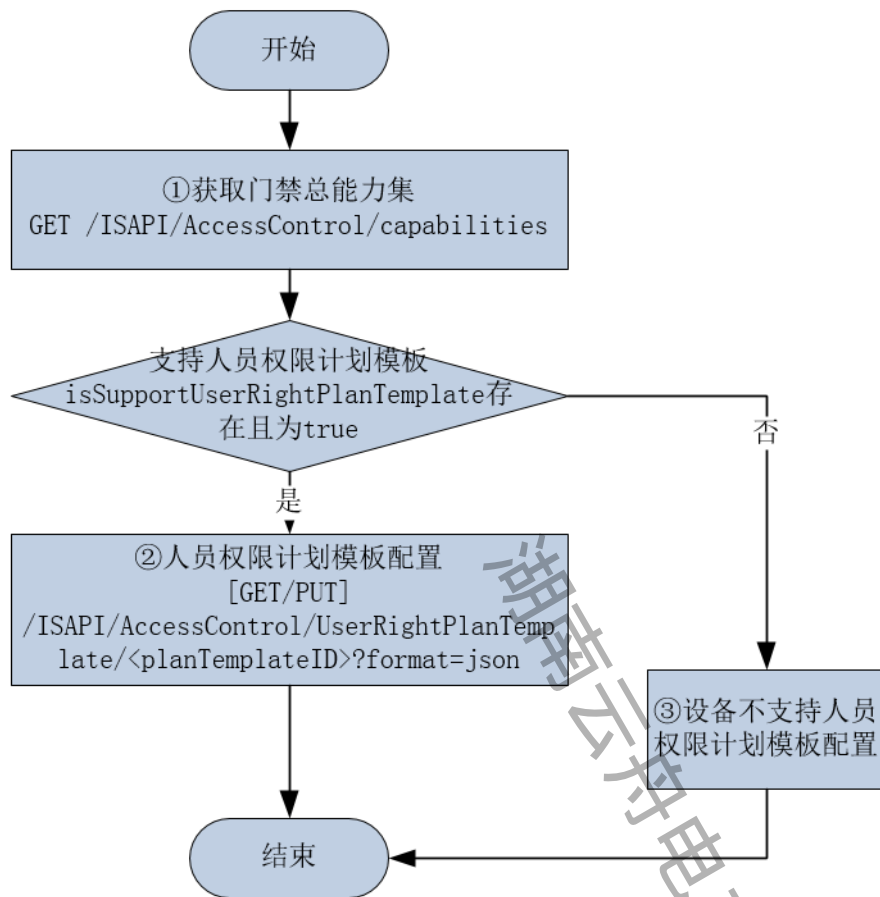
人员下发的时候需要关联门权限和每个门关联的人员权限计划模板，人员下发参考“人员及凭证管理”中“人员管理”的流程说明。人员权限计划是必须配置的，否则下发的人员将没有权限。

每一个计划模板可以关联一个周计划和四个假日组计划，假日计划的优先级高于周计划。周计划可以配置周一到周日的时段，每天支持配置8个不同的时段；假日组计划可以关联16个不同的假日计划，每一个假日计划可以配置一个假日的起止日期，日期范围内每天的时段相同（可以配置不同的8个时段）。通过这个计划模板配置，可以实现人员时间权限管理。



55.2 集成流程

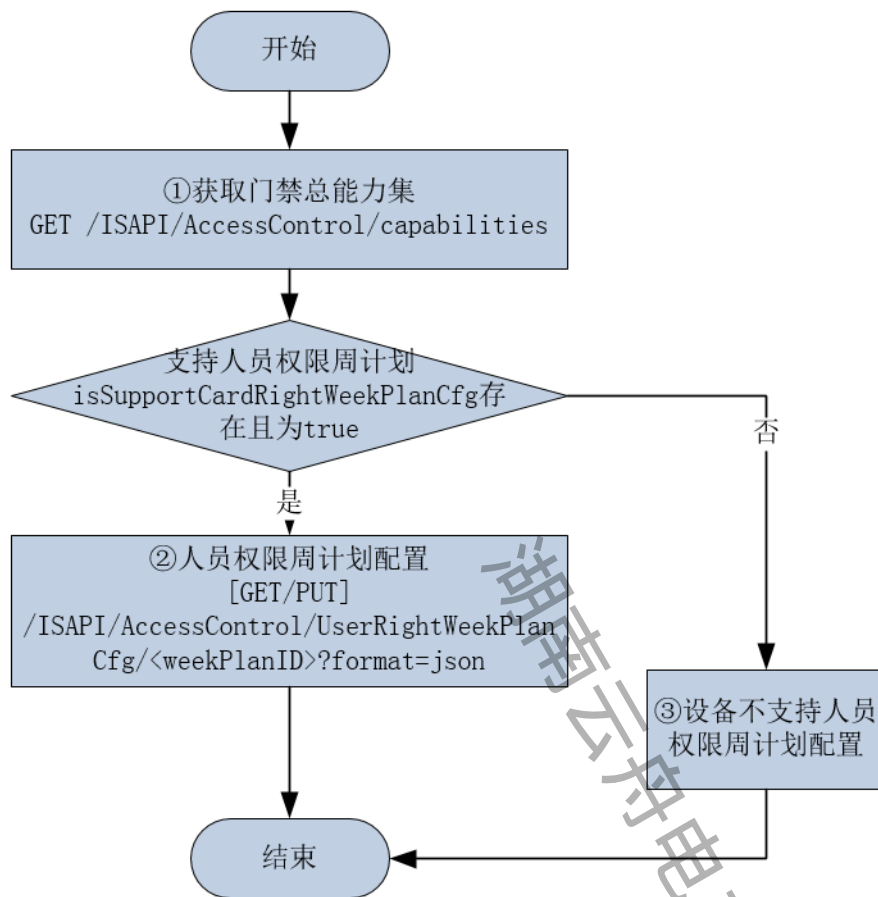
55.2.1 人员权限计划模板配置



集成流程如下：

1. 判断是否支持人员权限计划模板配置：GET /ISAPI/AccessControl/capabilities中 isSupportUserRightPlanTemplate存在且为true，代表支持人员权限计划模板配置（如支持该配置，则设备一定支持人员权限周计划配置）。
2. 人员权限计划模板配置：[GET/PUT] /ISAPI/AccessControl/UserRightPlanTemplate/<planTemplateID>?format=json，配置人员权限计划模板。
3. 设备不支持人员权限计划模板配置。

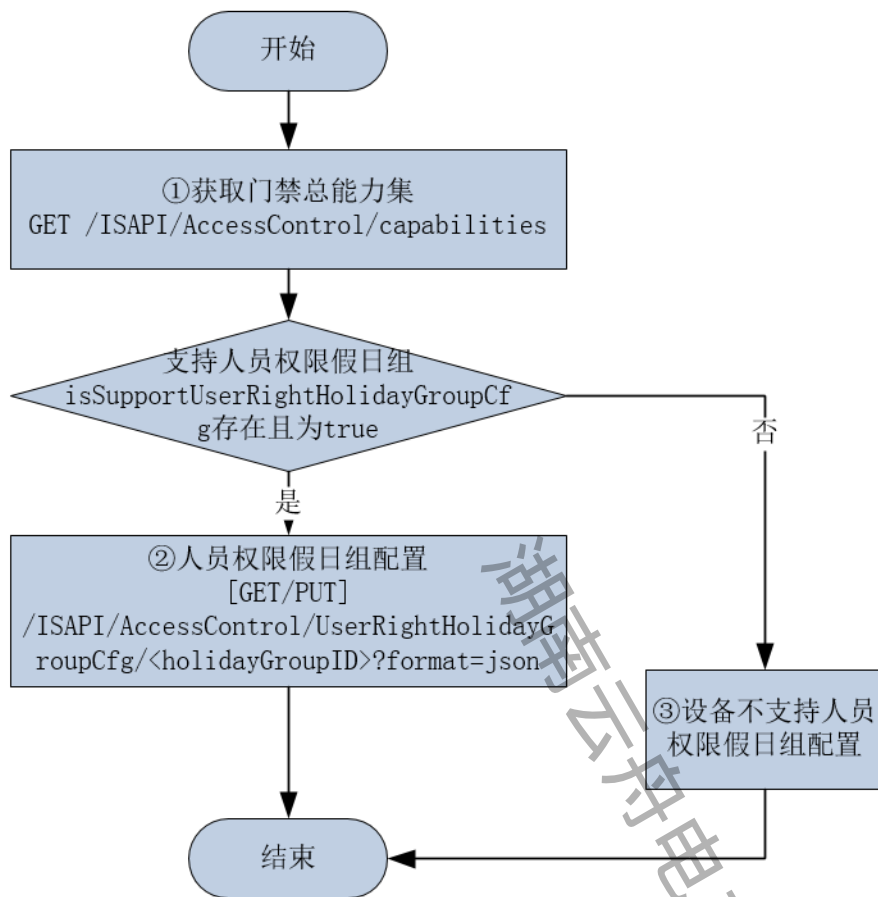
55.2.2 人员权限周计划配置



集成流程如下：

1. 判断是否支持人员权限周计划配置：GET /ISAPI/AccessControl/capabilities中 isSupportCardRightWeekPlanCfg存在且为true，代表支持人员权限周计划配置。
2. 人员权限周计划配置：[GET/PUT] /ISAPI/AccessControl/UserRightWeekPlanCfg/<weekPlanID>?format=json，配置人员权限周计划。
3. 设备不支持人员权限周计划配置。

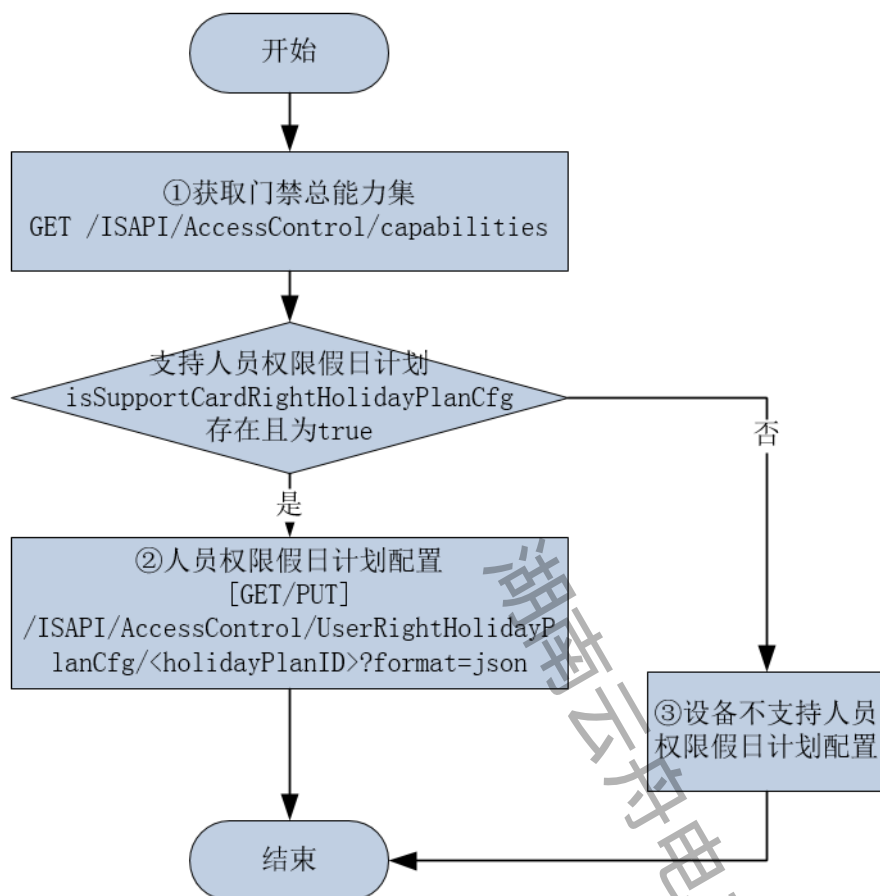
55.2.3 人员权限假日组配置



集成流程如下：

1. 判断是否支持人员权限假日组配置：GET /ISAPI/AccessControl/capabilities中 isSupportUserRightHolidayGroupCfg存在且为true，代表支持人员权限假日组配置（如支持该配置，则设备一定支持人员权限假日计划配置）。
2. 人员权限假日组配置：[GET/PUT] /ISAPI/AccessControl/UserRightHolidayGroupCfg/<holidayGroupID>?format=json，配置人员权限假日组。
3. 设备不支持人员权限假日组配置。

55.2.4 人员权限假日计划配置



集成流程如下：

1. 判断是否支持人员权限假日计划配置：GET /ISAPI/AccessControl/capabilities中 isSupportCardRightHolidayPlanCfg存在且为true，代表支持人员权限假日计划配置。
2. 人员权限假日计划配置：[GET/PUT] /ISAPI/AccessControl/UserRightHolidayPlanCfg/<holidayPlanID>?format=json，配置人员权限假日计划。
3. 设备不支持人员权限假日计划配置。

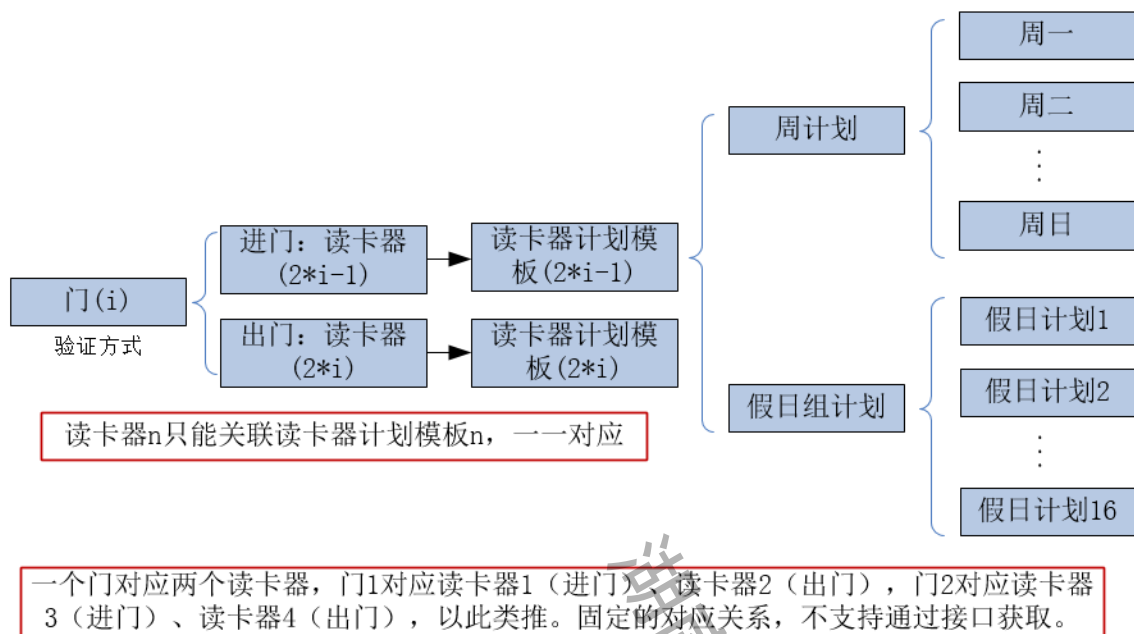
56 认证方式计划管理

56.1 功能介绍

配置门进出的读卡器的计划时间段和认证方式（刷卡、刷脸等认证方式）。一个门对应两个读卡器，门1对应读卡器1（进门）、读卡器2（出门），门2对应读卡器3（进门）、读卡器4（出门），以此类推。读卡器n只能关联读卡器计划模板n，一一对应。

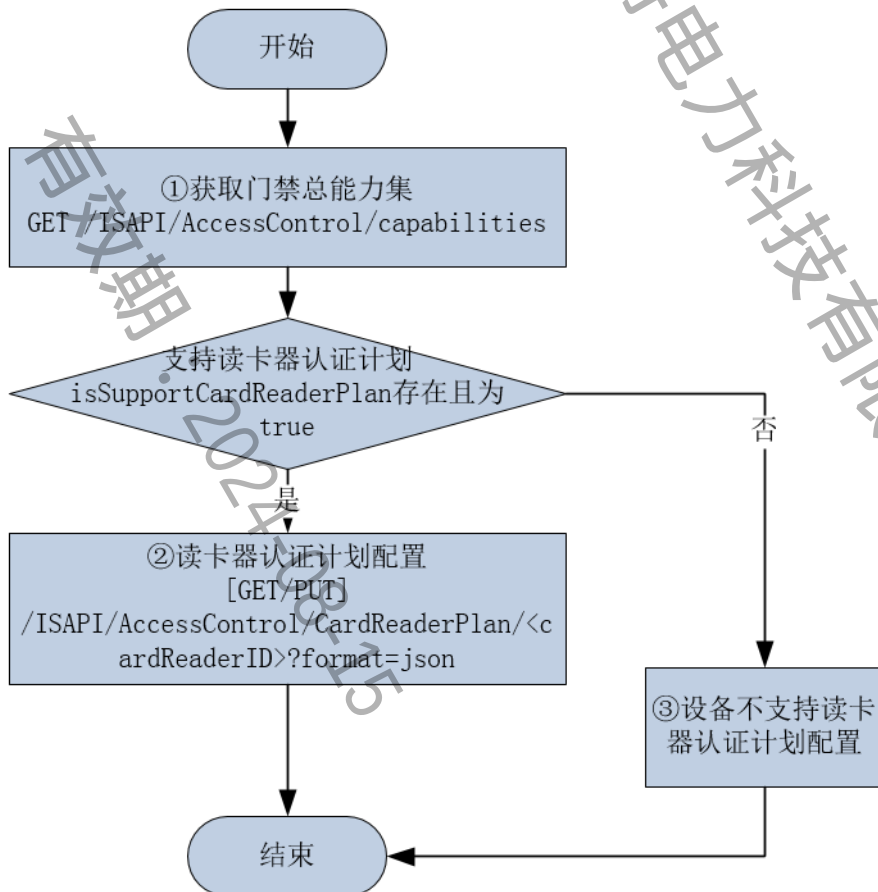
每一个计划模板可以关联一个周计划和四个假日组计划，假日计划的优先级高于周计划。周计划可以配置周一到周日的时段，每天支持配置8个不同的时段；假日组计划可以关联16个不同的假日计划，每一个假日计划可以配置一个假日的起止日期，日期范围内每天的时段相同（可以配置不同的8个时段）。通过这个计划模板配置，可以实现门的权限管理。

在“人员及凭证管理”的“人员管理”功能中，如给人员配置了人员认证方式（userVerifyMode），那么人员认证方式的优先级高于认证方式计划管理，该人员进行认证时，按照人员认证方式的配置生效。



56.2 集成流程

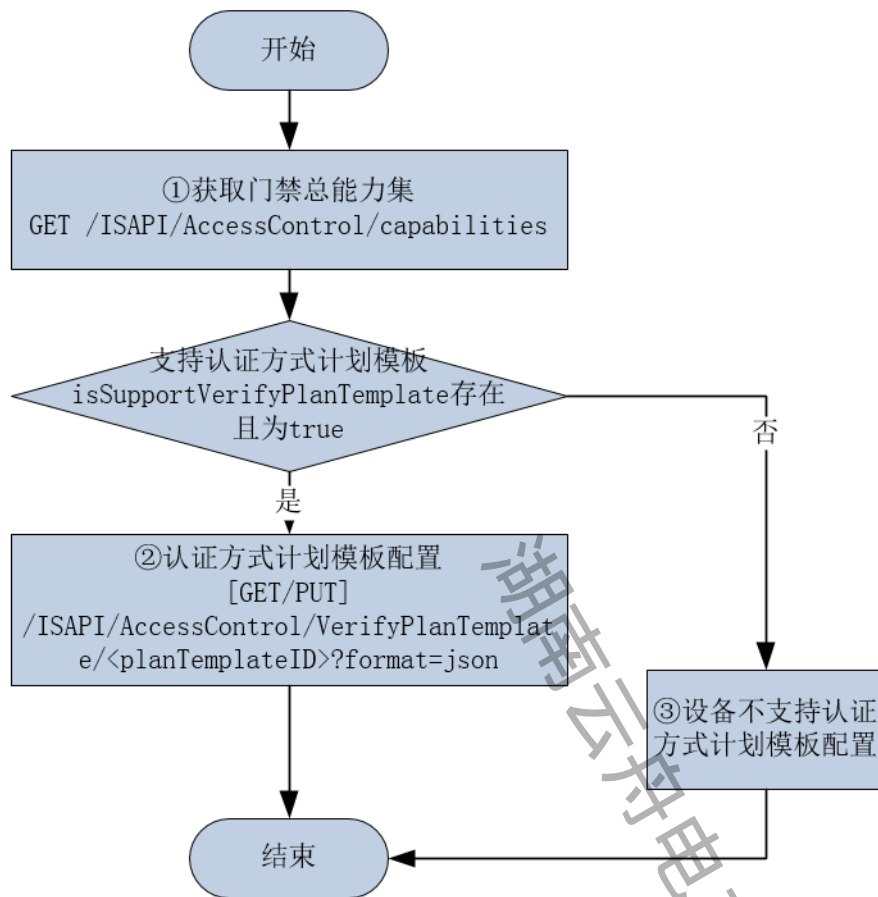
56.2.1 读卡器认证方式计划配置



集成流程如下：

1. 判断是否支持读卡器认证方式计划配置：GET /ISAPI/AccessControl/capabilities 中 isSupportCardReaderPlan 存在且为 true，代表支持读卡器认证方式计划配置（如支持该配置，则设备一定支持认证方式计划模板配置）。
2. 读卡器认证方式计划配置：[GET/PUT] /ISAPI/AccessControl/CardReaderPlan/<cardReaderID>?format=json，配置读卡器认证方式计划（URL 中的 cardReaderID 需要与报文中的 templateNo 编号保持一致）。
3. 设备不支持读卡器认证方式计划配置。

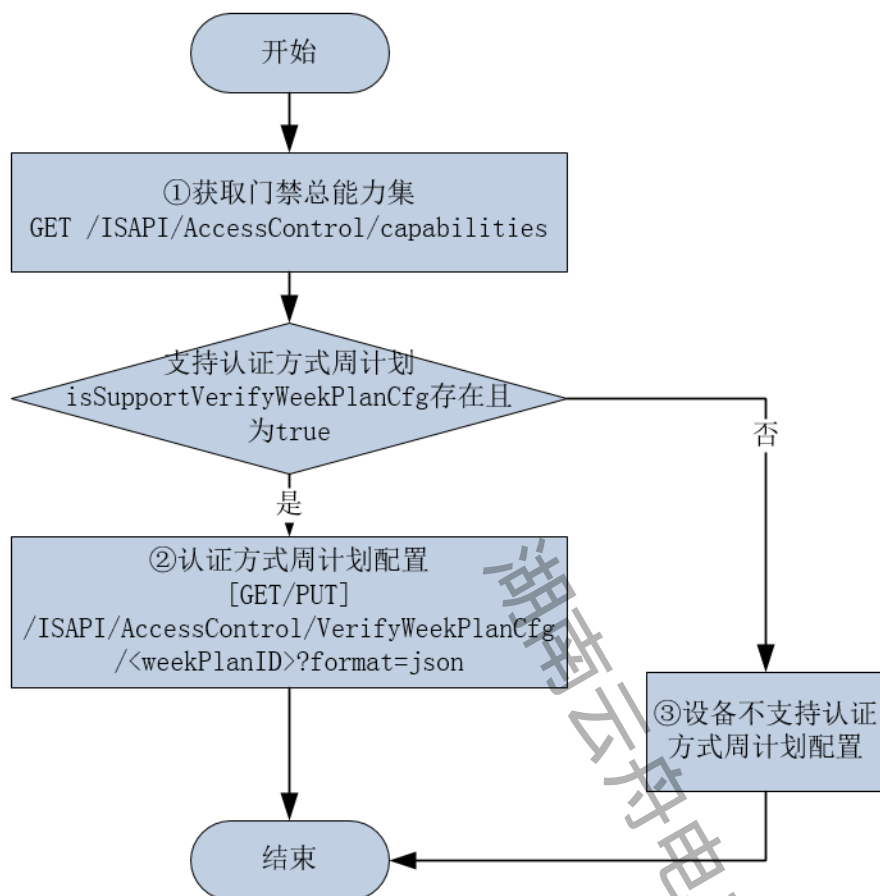
56.2.2 认证方式计划模板配置



集成流程如下：

1. 判断是否支持认证方式计划模板配置：GET /ISAPI/AccessControl/capabilities中isSupportVerifyPlanTemplate存在且为true，代表支持认证方式计划模板配置（如支持该配置，则设备一定支持认证方式周计划配置）。
2. 认证方式计划模板配置：[GET/PUT] /ISAPI/AccessControl/VerifyPlanTemplate/<planTemplateID>?format=json，配置认证方式计划模板。
3. 设备不支持认证方式计划模板配置。

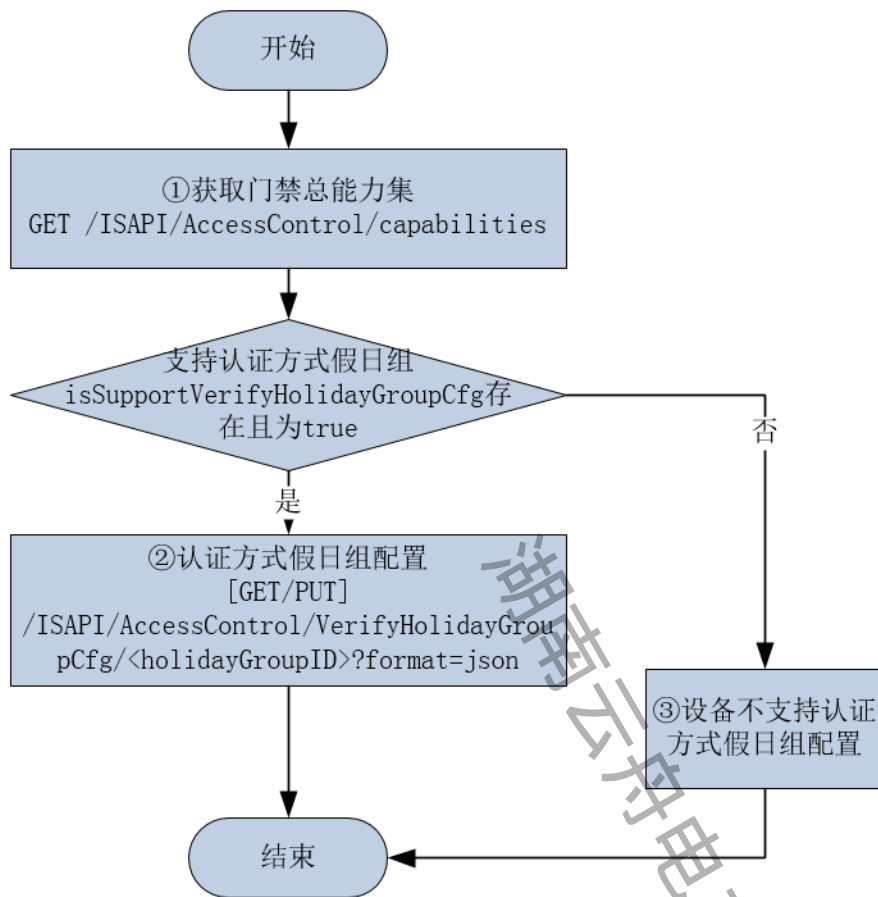
56.2.3 认证方式周计划配置



集成流程如下：

1. 判断是否支持认证方式周计划配置：GET /ISAPI/AccessControl/capabilities中isSupportVerifyWeekPlanCfg存在且为true，代表支持认证方式周计划配置。
2. 认证方式周计划配置：[GET/PUT] /ISAPI/AccessControl/VerifyWeekPlanCfg/<weekPlanID>?format=json，配置认证方式周计划。
3. 设备不支持认证方式周计划配置。

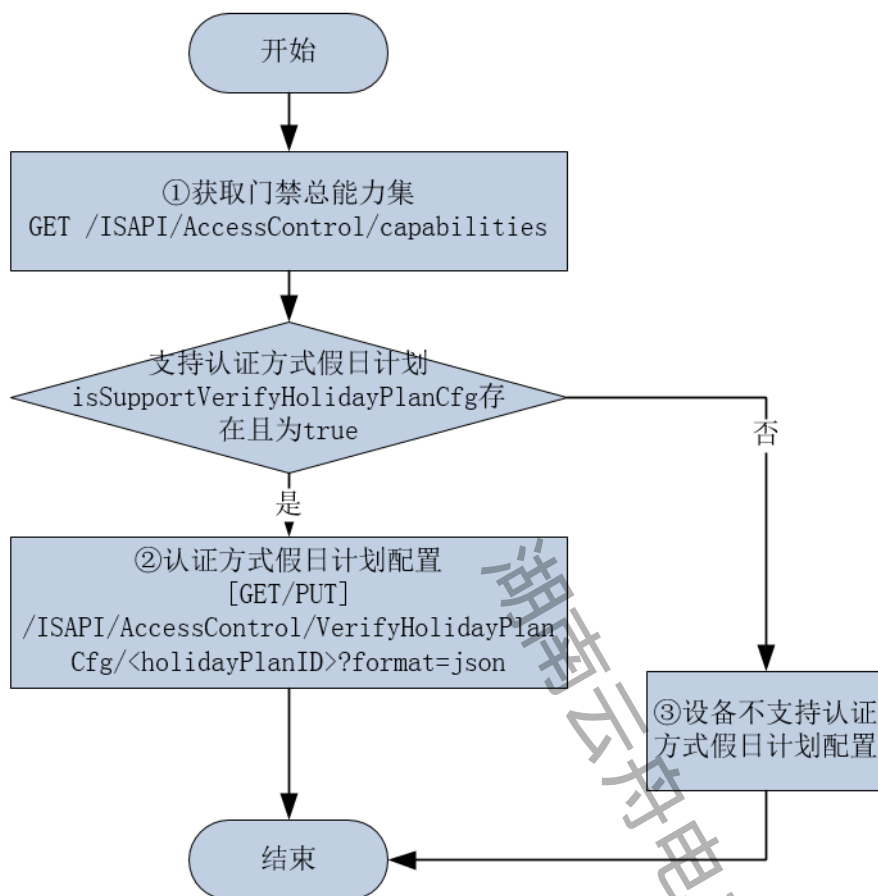
56.2.4 认证方式假日组配置



集成流程如下：

1. 判断是否支持认证方式假日组配置：GET /ISAPI/AccessControl/capabilities中 isSupportVerifyHolidayGroupCfg存在且为true，代表支持认证方式假日组配置（如支持该配置，则设备一定支持认证方式假日计划配置）。
2. 认证方式假日组配置：[GET/PUT] /ISAPI/AccessControl/VerifyHolidayGroupCfg/<holidayGroupID>?format=json，配置认证方式假日组。
3. 设备不支持认证方式假日组配置。

56.2.5 认证方式假日计划配置



集成流程如下：

1. 判断是否支持认证方式假日计划配置：GET /ISAPI/AccessControl/capabilities中 isSupportVerifyHolidayPlanCfg存在且为true，代表支持认证方式假日计划配置。
2. 认证方式假日计划配置：[GET/PUT] /ISAPI/AccessControl/VerifyHolidayPlanCfg/<holidayPlanID>?format=json，配置认证方式假日计划。
3. 设备不支持认证方式假日计划配置。

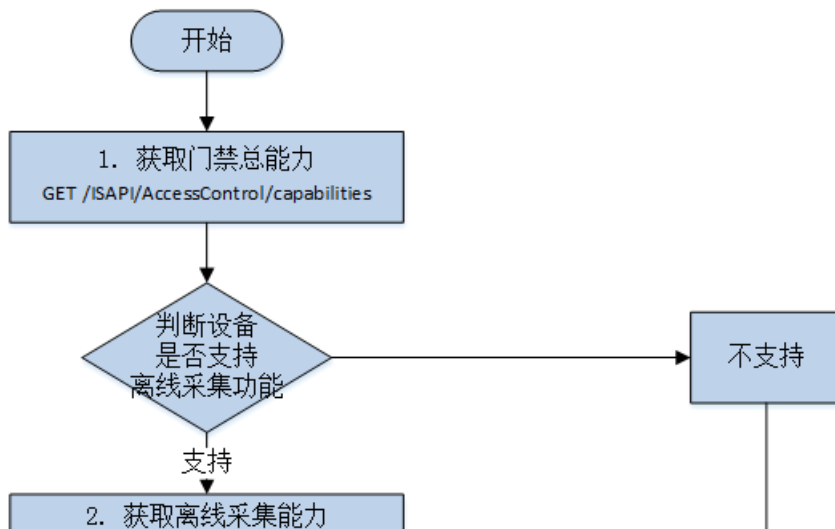
57 离线采集

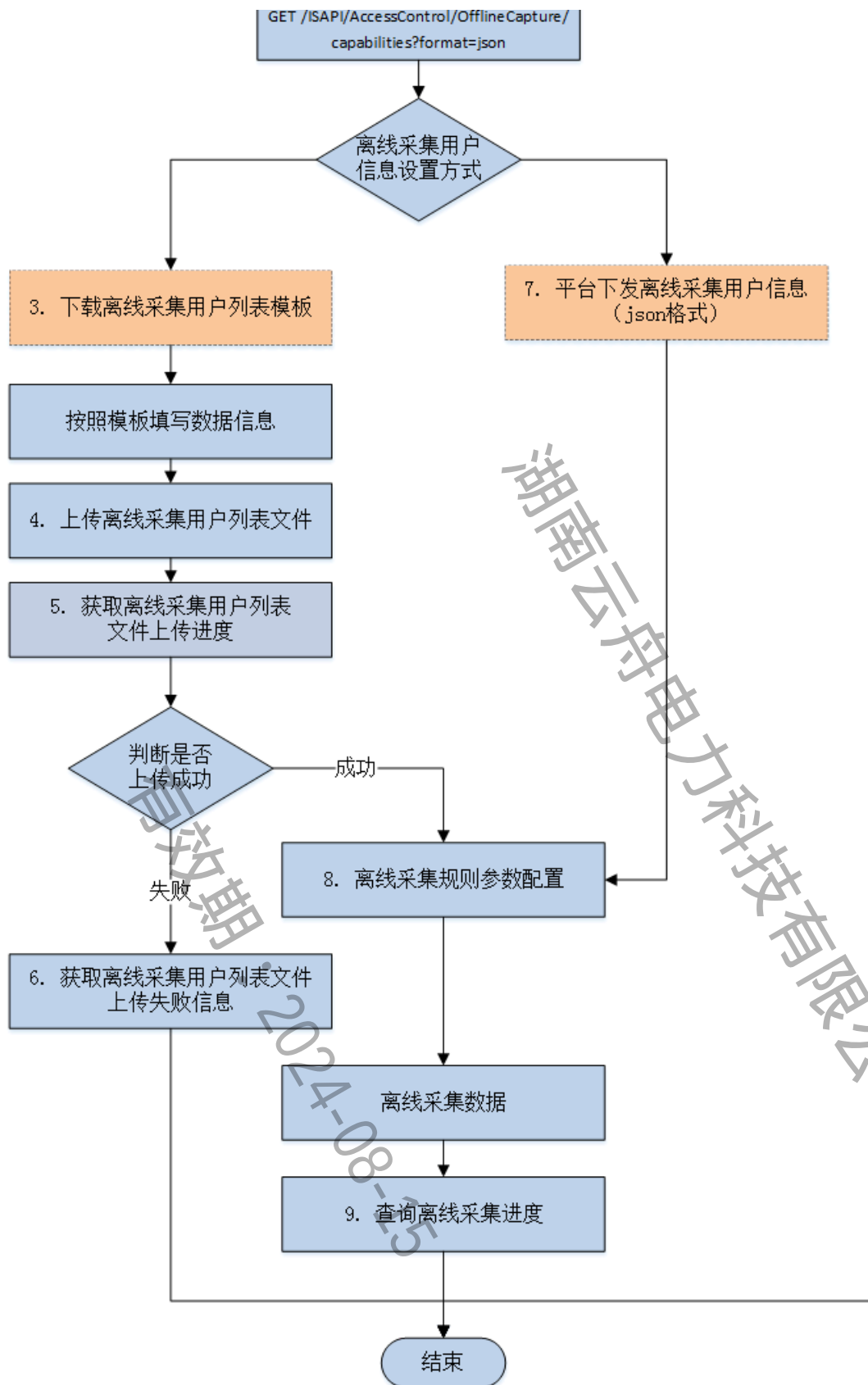
57.1 功能介绍

设备支持导入需要采集的信息的描述，然后本地采集相关数据存储，后续上层平台或客户端通过接口查询采集到的数据，用于满足非实时采集的场景需求，满足市场多变化的应用。

57.2 集成流程

57.2.1 离线采集数据





1. 获取门禁总能力GET /ISAPI/AccessControl/capabilities，通过isSupportOfflineCapture字段返回为true，判断设备支持数据离线采集功能；
2. 获取离线采集能力GET /ISAPI/AccessControl/OfflineCapture/capabilities?format=json，获取设备支持的离线采集功能相关的接口和参数规格；
3. 下载离线采集用户列表模板POST /ISAPI/AccessControl/OfflineCapture/InfoFileTemplateDownload?format=json；
4. 上传离线采集用户列表POST /ISAPI/AccessControl/OfflineCapture/InfoFile?format=json；

5. 获取离线采集用户列表文件上传进度GET /ISAPI/AccessControl/OfflineCapture/InfoFile/progress?format=json;
6. 获取离线采集用户列表文件上传失败信息GET /ISAPI/AccessControl/OfflineCapture/uploadFailedDetails?format=json;
7. 平台下发离线采集用户信息（json格式），新增离线采集用户信息（json格式）POST /ISAPI/AccessControl/OfflineCapture/userInfo?format=json，修改离线采集用户信息（json格式）PUT /ISAPI/AccessControl/OfflineCapture/userInfo?format=json;
8. 离线采集规则参数配置，获取离线采集规则参数GET /ISAPI/AccessControl/OfflineCapture/ruleInfo?format=json，配置离线采集规则参数PUT /ISAPI/AccessControl/OfflineCapture/ruleInfo?format=json;
9. 查询离线采集进度GET /ISAPI/AccessControl/OfflineCapture/progress?format=json。

57.2.2 数据处理

1. 查询离线采集数据POST /ISAPI/AccessControl/OfflineCapture/DataCollections/searchTask?format=json;
2. 下载离线采集数据POST /ISAPI/AccessControl/OfflineCapture/DataCollections/downloadTask?format=json;
3. 导出离线采集数据PUT /ISAPI/AccessControl/OfflineCapture/dataOutput?format=json&security=<security>&iv=<iv>; 获取离线采集数据导出进度GET /ISAPI/AccessControl/OfflineCapture/dataOutput/progress?format=json ;
4. 删除离线采集数据：删除离线采集数据（按采集序号） DELETE /ISAPI/AccessControl/OfflineCapture/DataCollections/<captureNo>?format=json，删除离线采集数据（按条件） POST /ISAPI/AccessControl/OfflineCapture/deleteDataCollections?format=json，删除全部离线采集数据DELETE /ISAPI/AccessControl/OfflineCapture/DataCollections?format=json。

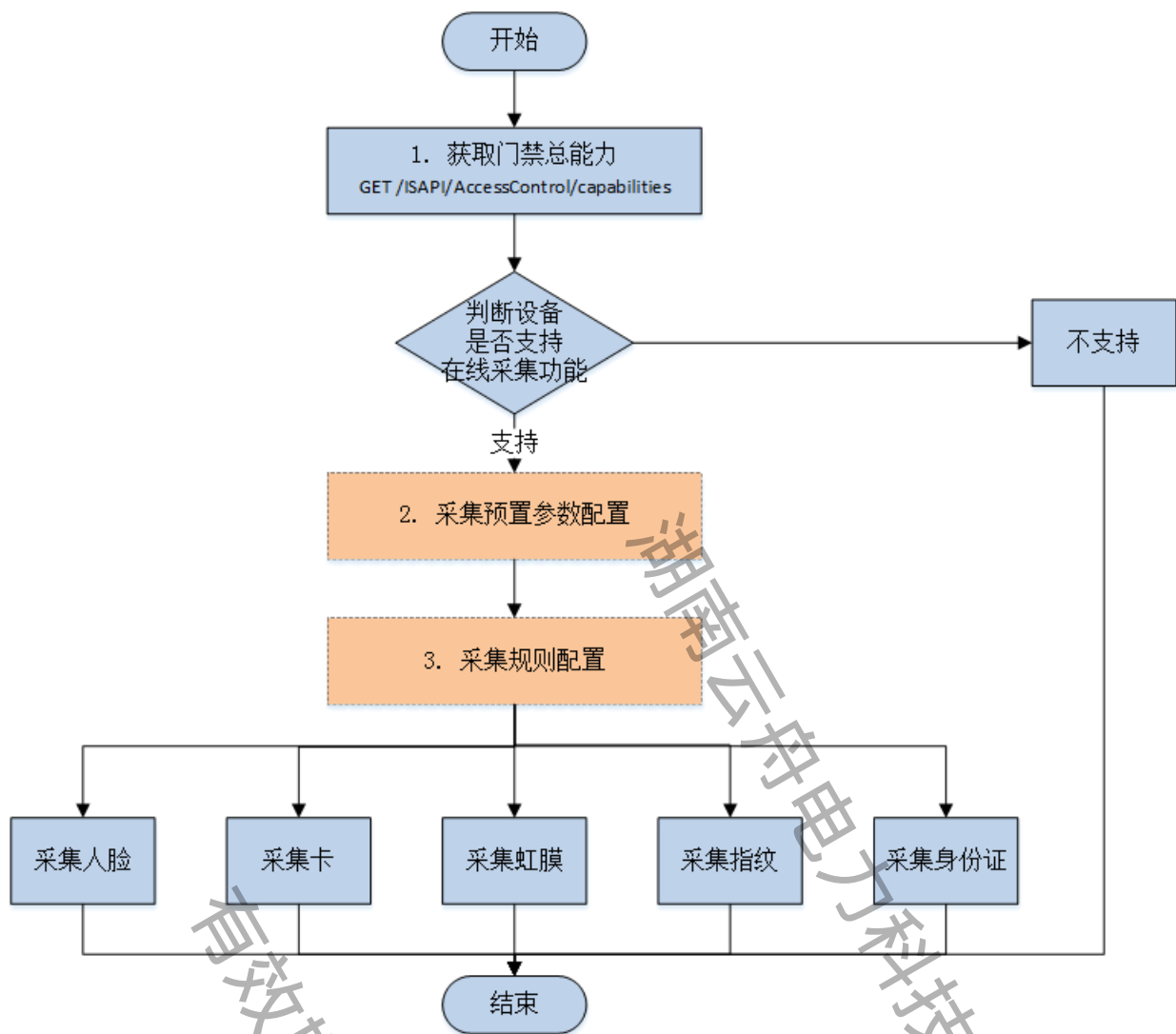
58 在线采集

58.1 功能介绍

设备支持实时采集人脸、卡号、指纹、虹膜、身份证等信息，满足市场多变化的应用。

58.2 集成流程

58.2.1 在线采集总体流程



1. 调用获取门禁总能力 `GET /ISAPI/AccessControl/capabilities`，通过 `isSupportCaptureFace`、`isSupportCaptureInfraredFace`、`isSupportCaptureCardInfo`、`isSupportCaptureIrisData`、`isSupportCaptureFingerPrint`、`isSupportCaptureIDInfo`、`isSupportCapturePresetParam`、`isSupportCaptureRule`、字段返回为true，判断设备是否支持采集人脸、采集红外人脸、采集卡、采集虹膜、采集指纹、采集身份证以及采集预置参数配置、采集规则配置功能；
2. 调用 `GET /ISAPI/AccessControl/CapturePresetParam/capabilities?format=json`，获取设备支持的采集预置参数配置功能的详细参数规格；调用 `GET|PUT /ISAPI/AccessControl/CapturePresetParam?format=json&security=<security>&iv=<iv>`，进行采集预置参数的获取与配置；
3. 调用 `GET /ISAPI/AccessControl/CaptureRule/capabilities?format=json`，获取设备支持的采集规则配置功能的详细参数规格；调用 `GET|PUT /ISAPI/AccessControl/CaptureRule?format=json`，进行采集预置参数的获取与配置；

58.2.2 采集人脸

1. 调用 `GET /ISAPI/AccessControl/CaptureFaceData/capabilities` 获取人脸信息采集能力，得到采集人脸的详细参数规格；调用 `GET /ISAPI/AccessControl/CaptureFaceData/Progress/capabilities` 获取采集人脸信息进度能力集，得到人脸信息进度获取的详细参数规格；
2. 调用 `POST /ISAPI/AccessControl/CaptureFaceData`，采集人脸信息；
3. 循环调用 `GET /ISAPI/AccessControl/CaptureFaceData/Progress?readerID=<readerID>`，获取采集人脸进度，直到进度值为100，上层开始解析人脸URL或数据。

58.2.3 采集卡

1. 调用 `GET /ISAPI/AccessControl/CaptureCardInfo/capabilities?format=json` 获取卡采集能力，得到采集卡的详细参数规格；
2. 调用 `GET /ISAPI/AccessControl/CaptureCardInfo?format=json`，采集卡信息。

58.2.4 采集指纹

1. 调用GET /ISAPI/AccessControl/CaptureFingerPrint/capabilities 获取指纹采集能力，得到采集指纹的详细参数规格；
2. 调用POST /ISAPI/AccessControl/CaptureFingerPrint，采集指纹信息。

58.2.5 采集虹膜

1. 调用GET /ISAPI/AccessControl/captureIrisData/capabilities?format=json 获取虹膜信息采集能力，得到采集虹膜的详细参数规格；
2. 调用POST /ISAPI/AccessControl/captureIrisData?format=json，采集虹膜信息；
3. 循环调用GET /ISAPI/AccessControl/captureIrisData/progress?format=json&security=<security>&iv=<iv>，获取采集虹膜进度，直到采集进度状态为采集成功。

58.2.6 采集身份证

1. 调用GET /ISAPI/AccessControl/CaptureIDInfo/capabilities?format=json 获取身份证采集能力，得到采集身份证的详细参数规格；
2. 调用POST /ISAPI/AccessControl/CaptureIDInfo?format=json&security=<security>&iv=<iv>，采集身份证信息。

59 访客管理

59.1 功能介绍

实现预约访客和非预约访客的访客登记、访客信息同步、访客签离功能。

59.2 集成场景

1. 访客机、门禁设备、平台处于同一个局域网络中。
2. 来访访客分为：预约访客、非预约访客。
3. 访客到访后，对于预约/非预约访客在访客机上填写访客基本信息，并进行黑名单比对，人证比对，比对通过后，将登记信息上传给平台。
4. 平台将登记信息传递给其他访客机，同步访客信息。
5. 平台将访客权限（通过提供给访客的物理卡或二维码及人脸抓拍照片）下发给门禁设备（门禁控制器/指纹一体机/明眸/门口机等），让访客有某个门/闸机的开门权限。
6. 访客离开时，在访客机上进行签离，上传给平台，平台将签离信息同步给其他访客机，同时将访客权限从门禁设备上删除。

注：访客信息同步主要用于园区/小区部署多台访客机的应用场景，用于访客在一台访客机上签到，在其他访客机上可以查询该访客信息或进行签离的场景。同步时间和是否同步可配：默认为不同步；如配置为同步，则默认同步时间为10分钟。

59.3 集成流程

59.3.1 访客机准备

在访客来访前，首先要进行访客机配置：

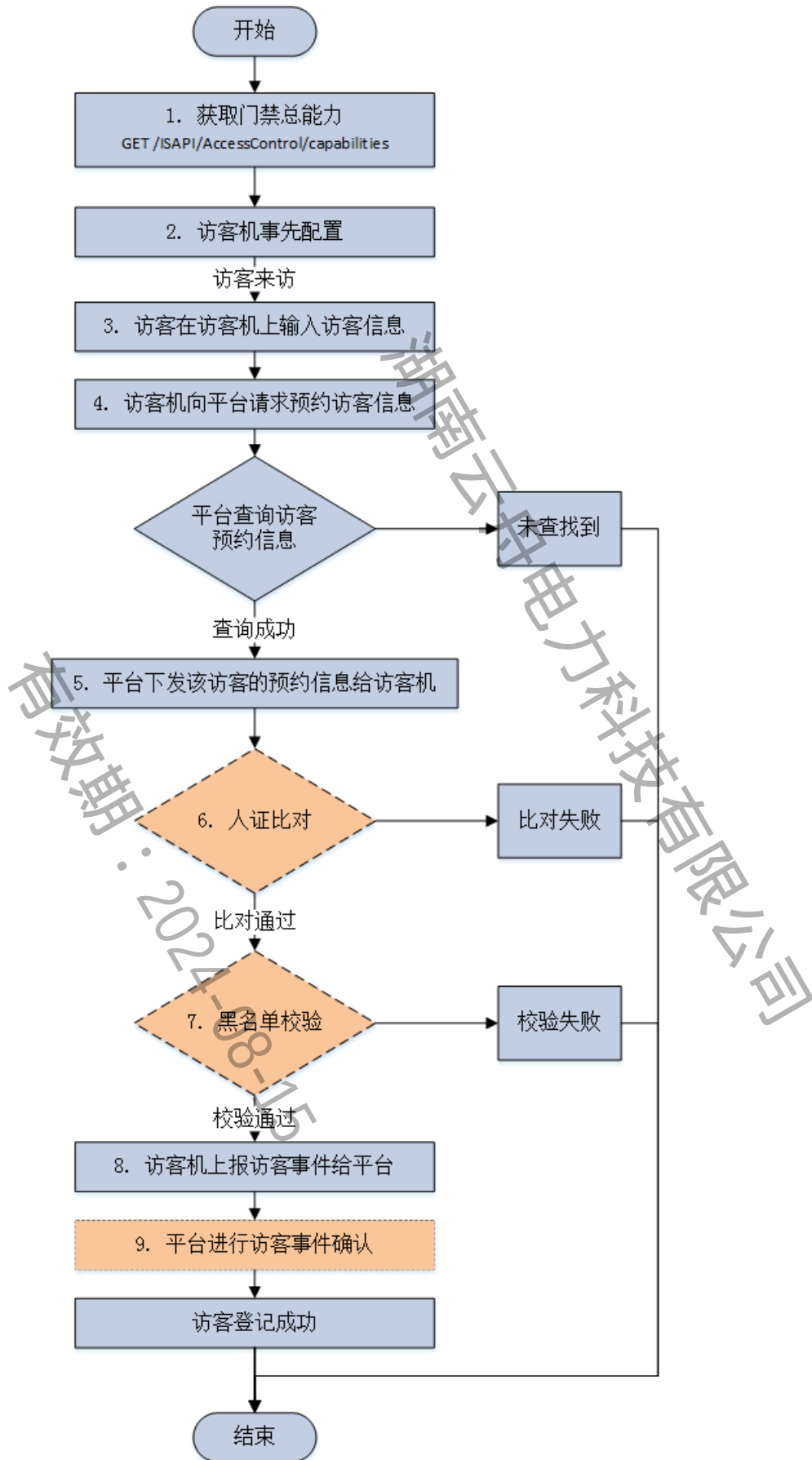
1. 访客机黑名单信息下发（身份证号），以及黑名单校验功能是否开启。调用GET /ISAPI/AccessControl/capabilities接口，根据返回的isSupportIDBlackListCfg、isSupportIDBlackListClear、isSupportIDBlackListParamCfg字段，来判断设备是否支持黑名单下发、清空、配置。如isSupportIDBlackListCfg字段返回true，调用GET /ISAPI/AccessControl/IDBlackListCfg/capabilities获取详细的身份证黑名单能力，调用PUT /ISAPI/AccessControl/IDBlackListCfg下发身份证黑名单；如isSupportIDBlackListClear字段返回true，调

用PUT /ISAPI/AccessControl/IDBlackListClear?format=json用以清空身份证黑名单；如isSupportIDBlackListParamCfg字段返回true，调用GET /ISAPI/AccessControl/IDBlackListParamCfg/capabilities?format=json获取详细的身份证黑名单配置能力，调用PUT /ISAPI/AccessControl/IDBlackListParamCfg?format=json配置身份证黑名单功能是否开启。

2. 访客机是否进行人证比对，以及人证比对的验证方式。调用GET /ISAPI/AccessControl/capabilities，如isSupportCertificateCompareCfg字段返回为true，表示设备支持人证比对参数配置，调用GET /ISAPI/AccessControl/certificateCompareCfg/capabilities?format=json 获取详细的人证比对参数配置能力，调用PUT /ISAPI/AccessControl/certificateCompareCfg?format=json配置人证比对参数，验证方式对应verifyMode字段。
3. 访客凭证是二维码、刷卡、二维码和刷卡，或者没有凭证等。调用GET /ISAPI/AccessControl/capabilities接口，根据返回的isSupportVisitorParamCfg字段，来判断设备是否支持访客参数配置，如isSupportVisitorParamCfg字段返回true，调用GET /ISAPI/AccessControl/visitorParamCfg/capabilities?format=json接口，根据是否返回rightAuthType字段，来判断设备是否支持访客权限凭证类型配置，如能力集中返回了rightAuthType字段，则可以调用PUT /ISAPI/AccessControl/visitorParamCfg?format=json接口，配置rightAuthType字段。
4. 自助访客机刷卡授权请求跳过功能，是否支持员工授权。调用GET /ISAPI/AccessControl/capabilities接口，根据返回的isSupportVisitorParamCfg字段，来判断设备是否支持访客参数配置，如isSupportVisitorParamCfg字段返回true，调用GET /ISAPI/AccessControl/visitorParamCfg/capabilities?format=json接口，根据是否返回skipCertificateCompareEnabled字段，来判断设备是否支持是否允许手动跳过人证比对配置，如能力集中返回了skipCertificateCompareEnabled字段，则可以调用PUT /ISAPI/AccessControl/visitorParamCfg?format=json接口，配置skipCertificateCompareEnabled字段。
5. 访客机是否自动同步请求访客信息（向平台请求，用于多台访客机间同步访客登记、签离信息）。调用GET /ISAPI/AccessControl/capabilities接口，根据返回的isSupportVisitorAdvanceParamCfg字段，来判断设备是否支持访客高级参数配置，如isSupportVisitorAdvanceParamCfg字段返回true，调用GET /ISAPI/AccessControl/visitorAdvanceParamCfg/capabilities?format=json接口，根据是否返回autoSyncInfoEnabled字段，来判断设备是否支持自动同步访客信息，如能力集中返回了autoSyncInfoEnabled字段，则可以调用PUT /ISAPI/AccessControl/visitorAdvanceParamCfg?format=json接口，配置autoSyncInfoEnabled字段。
6. 访客机实时登记、签离事件是否需要确认。调用GET /ISAPI/AccessControl/capabilities接口，根据返回的isSupportVisitorAdvanceParamCfg字段，来判断设备是否支持访客高级参数配置，如isSupportVisitorAdvanceParamCfg字段返回true，调用GET /ISAPI/AccessControl/visitorAdvanceParamCfg/capabilities?format=json接口，根据是否返回currentEventConfirmEnabled字段，来判断设备是否支持开启访客实时登记和签离事件确认功能，如能力集中返回了currentEventConfirmEnabled字段，则可以调用PUT /ISAPI/AccessControl/visitorAdvanceParamCfg?format=json接口，配置currentEventConfirmEnabled字段。
7. 访客机预约功能是否开启。调用GET /ISAPI/AccessControl/capabilities接口，根据返回的isSupportVisitorAdvanceParamCfg字段，来判断设备是否支持访客高级参数配置，如isSupportVisitorAdvanceParamCfg字段返回true，调用GET /ISAPI/AccessControl/visitorAdvanceParamCfg/capabilities?format=json接口，根据是否返回visitorAppointEnabled字段，来判断设备是否支持开启访客预约功能，如能力集中返回了visitorAppointEnabled字段，则可以调用PUT /ISAPI/AccessControl/visitorAdvanceParamCfg?format=json接口，配置visitorAppointEnabled字段。
8. 访客权限组配置。调用GET /ISAPI/AccessControl/capabilities，如isSupportVisitorRightListCfg字段返回为true，表示设备支持访客权限列表配置，调用GET /ISAPI/AccessControl/visitorRightListCfg/capabilities?format=json 获取详细的访客权限列表配置能力，调用PUT /ISAPI/AccessControl/visitorRightListCfg?format=json 配置访客权限列表。
9. 访客机打印凭条信息及展示顺序（二维码用于门禁权限控制，也可以使用刷卡或者人脸）。调用GET /ISAPI/AccessControl/capabilities，如isSupportVisitorReceiptInfoCfg字段返回为true，表示设备支持访客凭条信息配置，调用GET /ISAPI/AccessControl/visitorReceiptInfoCfg/capabilities?format=json 获取详细的访客凭条信息配置能力，调用PUT /ISAPI/AccessControl/visitorReceiptInfoCfg?format=json 访客凭条信息。
10. 访客机UI展示界面。调用GET /ISAPI/AccessControl/capabilities，如isSupportVisitorRegisterInfoCfg字段返回为true，表示设备支持访客登记信息配置，调用GET /ISAPI/AccessControl/visitorRegisterInfoCfg/capabilities?format=json 获取详细的访客登记信息配置能

力，调用PUT /ISAPI/AccessControl/visitorRegisterInfoCfg?format=json 配置访客登记信息。

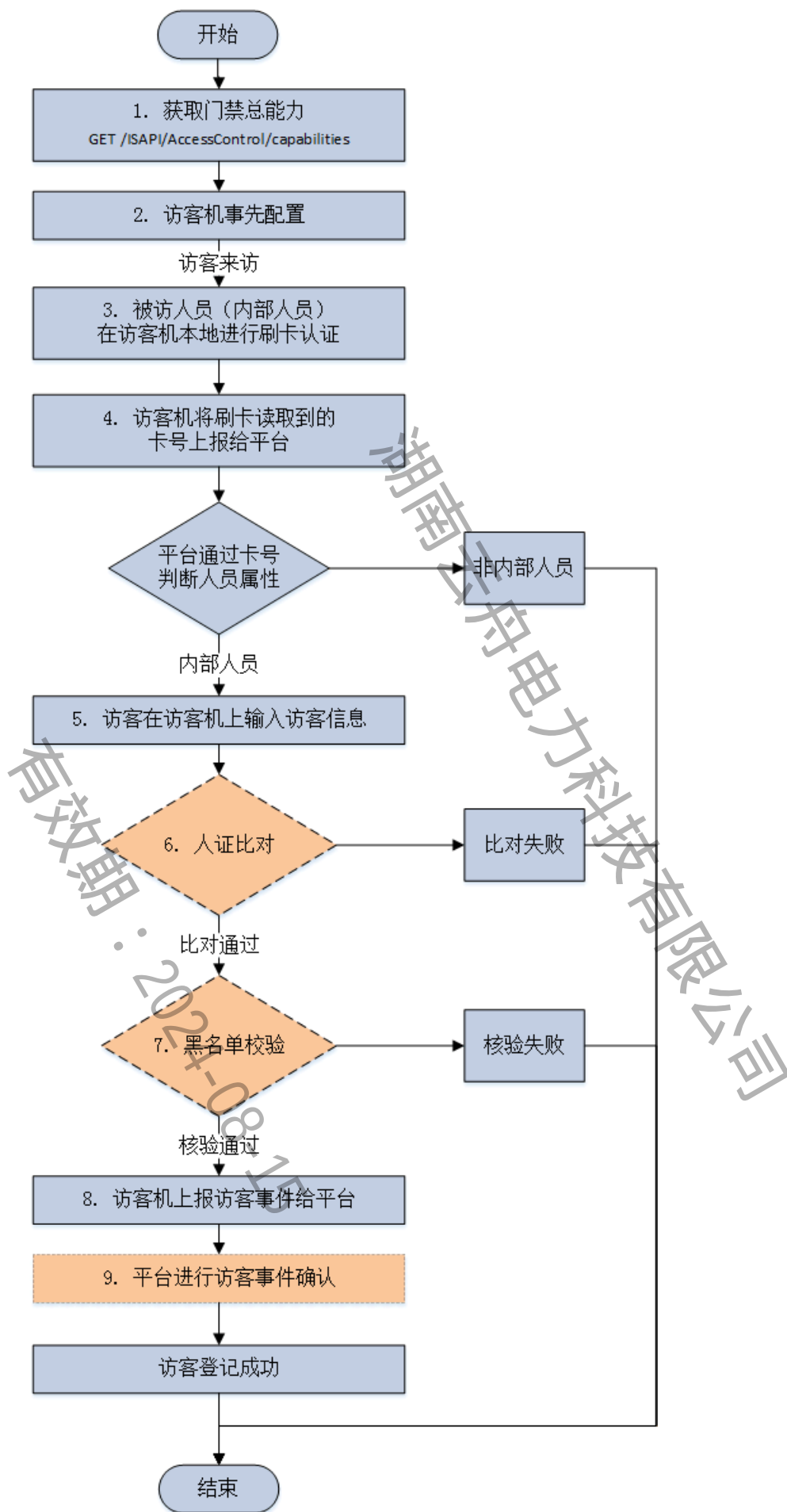
59.3.2 预约访客登记



1. 获取门禁总能力GET /ISAPI/AccessControl/capabilities，通过isSupportVisitorParamCfg、isSupportVisitorAdvanceParamCfg、isSupportVisitorRegisterInfoCfg、isSupportVisitorRightListCfg、isSupportVisitorReceiptInfoCfg、isSupportVisitorRegisterAudio字段返回为true，判断设备支持访客参数配置、访客高级参数配置、访客登记信息配置、访客权限配置、访客凭条信息配置、访客登记提示音配置；

2. 访客机事先配置，参考访客机准备；
3. 访客在访客机上输入访客信息，如认证码/手机号/身份证号；
4. 访客机向平台请求预约访客信息，上报访客信息查询事件 `VisitorInfoSearchEvent`；
5. 平台下发该访客的预约信息给访客机，平台在接入设备后，即会调用 `GET /ISAPI/AccessControl/visitorInfoSearchEventConfirm/capabilities?format=json` 获取访客信息查询事件确认能力，此处直接调用 `PUT /ISAPI/AccessControl/visitorInfoSearchEventConfirm?format=json&security=<security>&iv=<iv>` 进行访客信息查询事件确认；
6. 人证比对，即访客机本地进行访客证件照和人脸抓拍照比对；
7. 黑名单校验，平台事先将黑名单列表下发给访客机，在访客机本地做黑名单校验；
8. 访客机上报访客事件给平台， `VisitorEvent`；
9. 平台进行访客事件确认， 平台在接入设备后，即会调用 `GET /ISAPI/AccessControl/visitorEventConfirm/capabilities?format=json` 获取访客事件确认能力，此处直接调用 `PUT /ISAPI/AccessControl/visitorEventConfirm?format=json&security=<security>&iv=<iv>` 进行访客事件确认。若访客机事先开启访客实时登记和签离事件确认功能，而平台没有进行访客事件确认，则此处登记作为离线登记处理，访客虽登记成功，但是没有访客权限；若访客机事先没有开启访客实时登记和签离事件确认功能，则平台没有进行访客事件确认对访客无影响。

59.3.3 非预约访客登记（自助访客机）



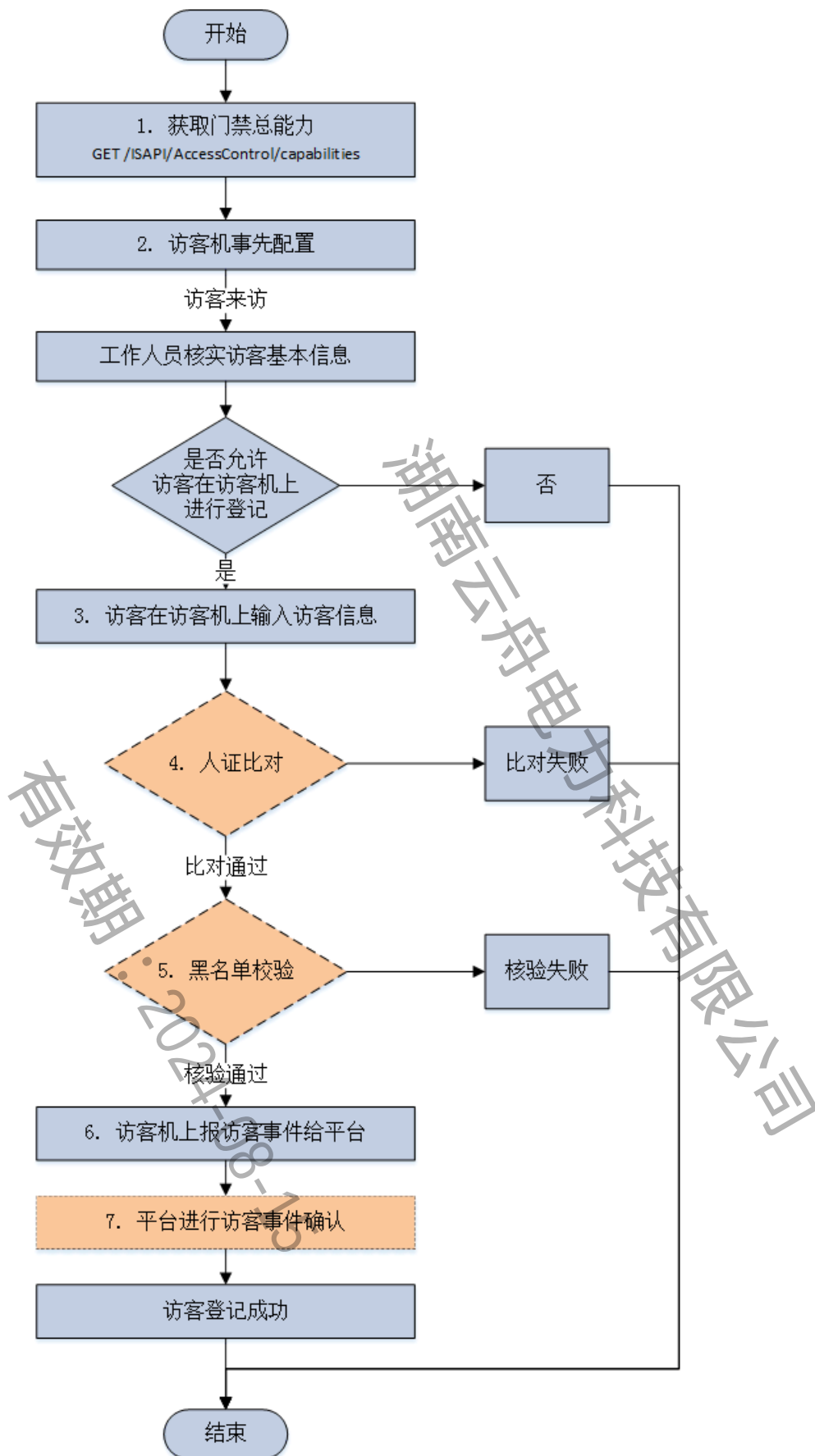
1. 获取门禁总能力，GET /ISAPI/AccessControl/capabilities，通过isSupportVisitorParamCfg、isSupportVisitorAdvanceParamCfg、isSupportVisitorRegisterInfoCfg、isSupportVisitorRightListCfg、isSupportVisitorReceiptInfoCfg、isSupportVisitorRegisterAudio字段返回为true，判断设备支持访客参数配置、访客高级参数配置、访客登记信息配置、访客权限配置、访客凭条信息配置、访客登记提示音配置；

2. 访客机事先配置，参考访客机准备；
3. 被访人员（内部人员）在访客机本地进行刷卡认证；
4. 访客机将刷卡读取到的卡号上报给平台，上报门禁事件 `AccessControllerEvent`；
5. 访客在访客机上输入访客信息，如认证码/手机号/身份证号；
6. 人证比对，即访客机本地进行访客证件照和人脸抓拍照比对；
7. 黑名单校验，平台事先将黑名单列表下发给访客机，在访客机本地做黑名单校验；
8. 访客机上报访客事件给平台， `VisitorEvent`；
9. 平台进行访客事件确认，平台在接入设备后，即会调用 `GET /ISAPI/AccessControl/visitorEventConfirm/capabilities?format=json` 获取访客事件确认能力，此处直接调用 `PUT /ISAPI/AccessControl/visitorEventConfirm?format=json&security=<security>&iv=<iv>` 进行访客事件确认。若访客机事先开启访客实时登记和签离事件确认功能，而平台没有进行访客事件确认，则此处登记作为离线登记处理，访客虽登记成功，但是没有访客权限；若访客机事先没有开启访客实时登记和签离事件确认功能，则平台没有进行访客事件确认对访客无影响。

59.3.4 非预约访客登记（非自助访客机）

湖南云舟电力科技有限公司

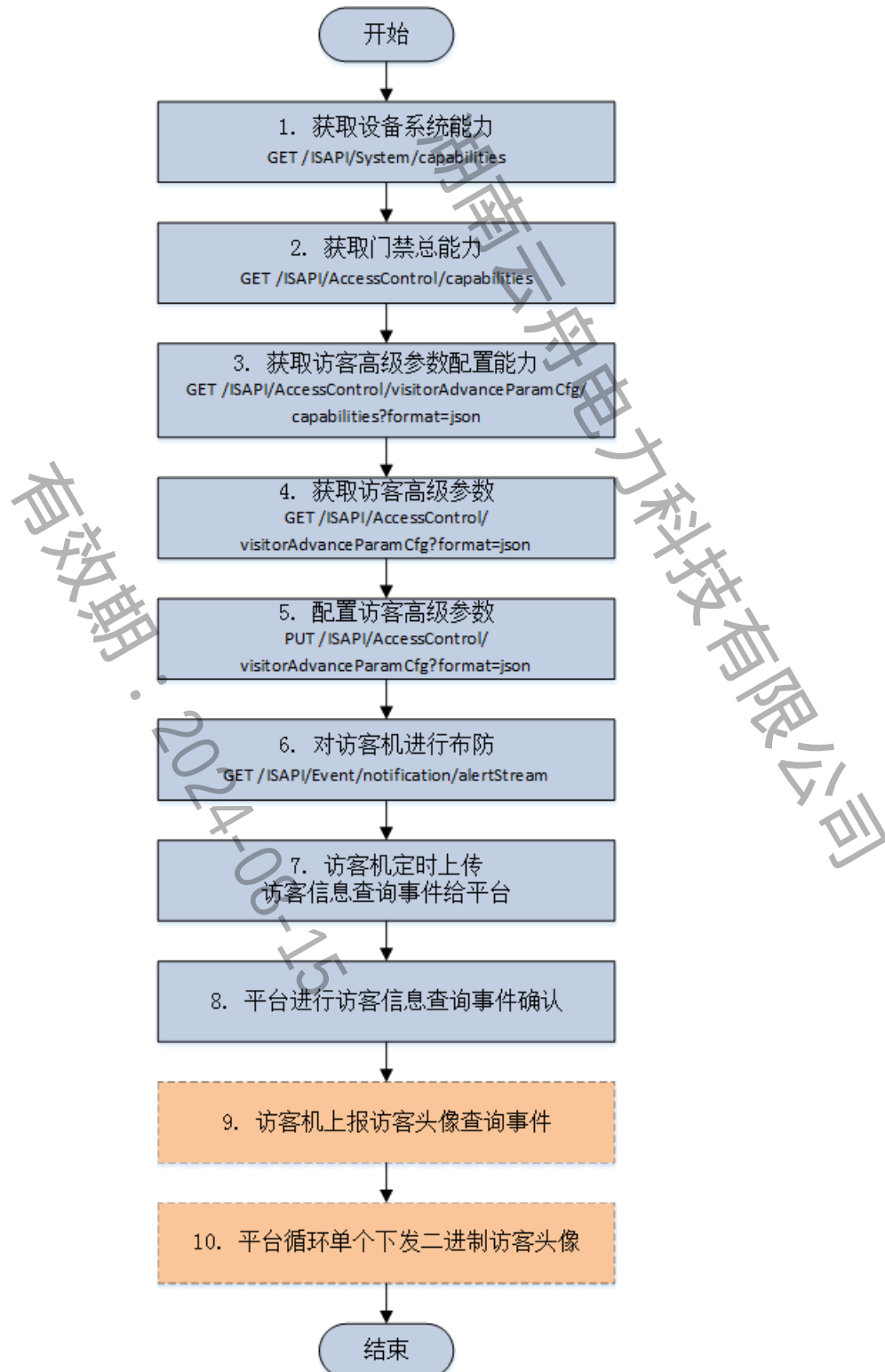
有效期：2024-08-15



1. 获取门禁总能力，GET /ISAPI/AccessControl/capabilities，通过isSupportVisitorParamCfg、isSupportVisitorAdvanceParamCfg、isSupportVisitorRegisterInfoCfg、isSupportVisitorRightListCfg、isSupportVisitorReceiptInfoCfg、isSupportVisitorRegisterAudio字段返回为true，判断设备支持访客参数配置、访客高级参数配置、访客登记信息配置、访客权限配置、访客凭条信息配置、访客登记提示音配置；
2. 访客机事先配置，参考访客机准备；
3. 访客在访客机上输入访客信息，如认证码/手机号/身份证号；
4. 人证比对，即访客机本地进行访客证件照和人脸抓拍拍照比对；

5. 黑名单校验，平台事先将黑名单列表下发给访客机，在访客机本地做黑名单校验；
6. 访客机上报访客事件给平台，`VisitorEvent`；
7. 平台进行访客事件确认，平台在接入设备后，即会调用 `GET /ISAPI/AccessControl/visitorEventConfirm/capabilities?format=json` 获取访客事件确认能力，此处直接调用 `PUT /ISAPI/AccessControl/visitorEventConfirm?format=json&security=<security>&iv=<iv>` 进行访客事件确认。若访客机事先开启访客实时登记和签离事件确认功能，而平台没有进行访客事件确认，则此处登记作为离线登记处理，访客虽登记成功，但是没有访客权限；若访客机事先没有开启访客实时登记和签离事件确认功能，则平台没有进行访客事件确认对访客无影响。

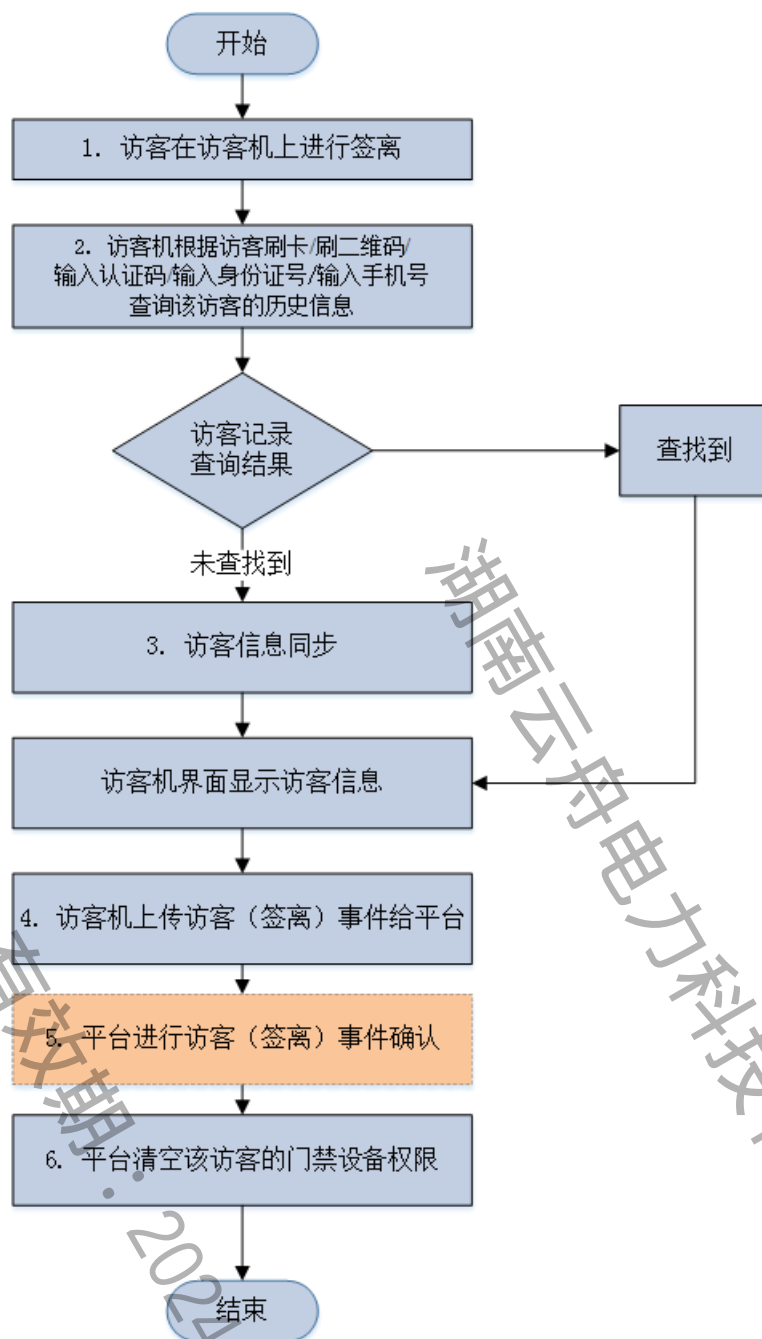
59.3.5 访客信息同步



1. 获取设备系统能力 `GET /ISAPI/System/capabilities`，通过 `isSupportVisitorInfoSearchEvent` 字段返回为 `true`，判断设备支持访客信息查询；通过 `isSupportVisitorPictureSearchEvent` 字段返回为 `true`，判断设备支持访客头像查询事件；

2. 获取门禁总能力GET /ISAPI/AccessControl/capabilities，通过 isSupportVisitorAdvanceParamCfg字段返回为true，判断设备支持访客高级参数配置；通过 isSupportAddVisitorPicture 字段返回为true，判断设备支持访客头像下发；
3. 获取访客高级参数配置能力GET /ISAPI/AccessControl/visitorAdvanceParamCfg/capabilities?format=json，通过autoSyncInfoEnabled、autoSyncInfoTimeInterval字段，判断设备是否支持自动同步访客信息以及同步时间间隔；
4. 获取访客高级参数GET /ISAPI/AccessControl/visitorAdvanceParamCfg?format=json；
5. 配置访客高级参数GET /ISAPI/AccessControl/visitorAdvanceParamCfg?format=json，autoSyncInfoEnabled字段为true，配置autoSyncInfoTimeInterval具体的自动同步访客信息时间间隔；
6. 对访客机进行布防GET /ISAPI/Event/notification/alertStream?deployID=<deployID>，此处以布防非订阅方式为例，URL参数deployID: 0-客户端布防，1-实时布防，不存在时，默认为客户端布防；
7. 访客机定时上传访客信息查询事件给平台，VisitorInfoSearchEvent；
8. 平台进行访客信息查询事件确认，平台在接入设备后，即会调用 GET /ISAPI/AccessControl/visitorInfoSearchEventConfirm/capabilities?format=json获取访客事件确认能力，此处直接调用PUT /ISAPI/AccessControl/visitorInfoSearchEventConfirm?format=json&security=<security>&iv=<iv> 进行访客信息查询事件确认；
9. 访客机上报访客头像查询事件，VisitorPictureSearchEvent，报文中可携带多个访客UUID；
10. 平台循环单个下发二进制访客头像，POST /ISAPI/AccessControl/addVisitorPicture?format=json。

59.3.6 访客签离



1. 访客在访客机上进行签离；
2. 访客机根据访客刷卡/刷二维码/输入认证码/输入身份证号/输入手机号查询该访客的历史信息，为本地查询；
3. 访客信息同步，参考访客信息同步章节；
4. 访客机上传访客（签离）事件给平台，复用VisitorEvent；
5. 平台进行访客（签离）事件确认，平台在接入设备后，即会调用 `GET /ISAPI/AccessControl/visitorEventConfirm/capabilities?format=json` 获取访客事件确认能力，此处直接调用 `PUT /ISAPI/AccessControl/visitorEventConfirm?format=json&security=<security>&iv=<iv>` 进行访客（签离）事件确认。
6. 平台清空该访客的门禁设备权限。

60 通话交互

60.1 功能介绍

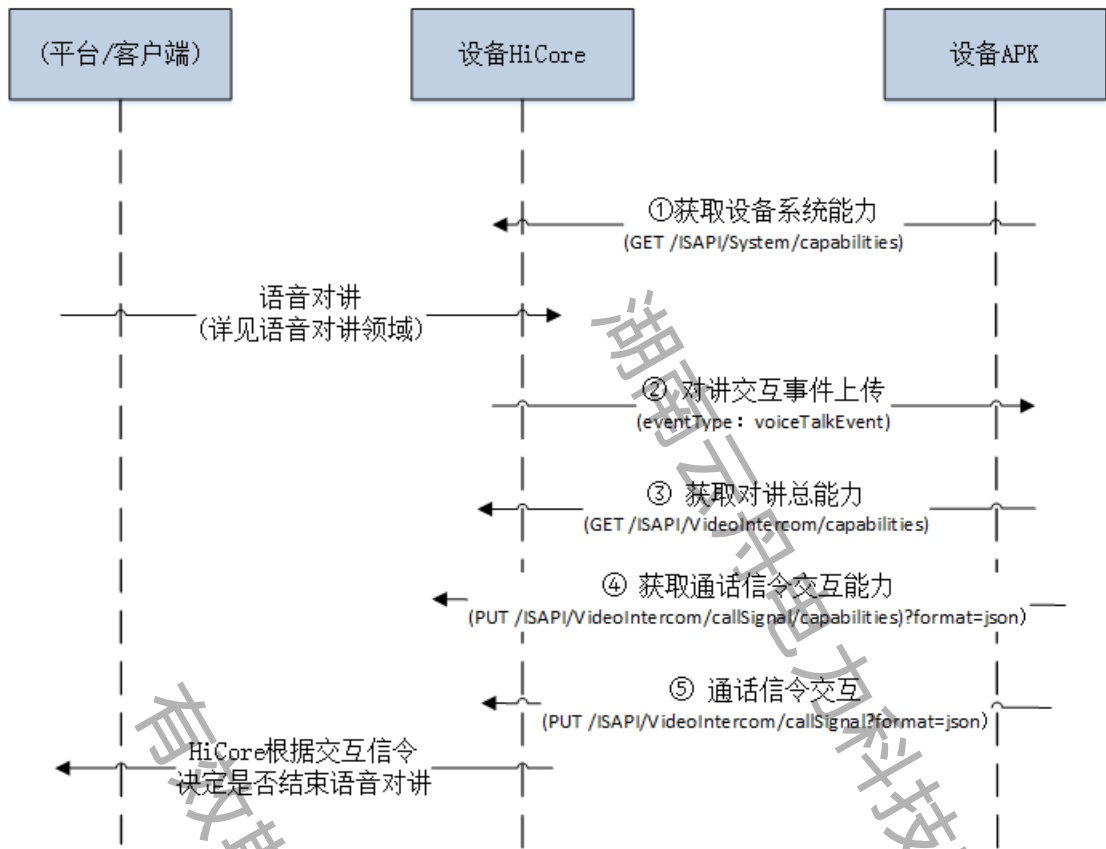
上层平台通过语音对讲(可参考语音对讲流程)与设备进行对讲交互，设备HiCore收到语音对讲后，向设备APK发送对讲交互事件(通过HEOP协议)。设备APK收到对讲交互事件后向设备HiCore发送通话交互信令，反馈通话选项。

通话交互也可应用于设备与设备间的对讲，与平台的区别在与设备间的对讲协议使用的是SIP协议。

设备与设备间交互，如果涉及管理机作为对讲方，对于已经采用了新长号规则的管理机设备，需要配置管理机对于终端设备长号的解析规则，兼容新老长号。

60.2 集成流程

60.2.1 通话交互



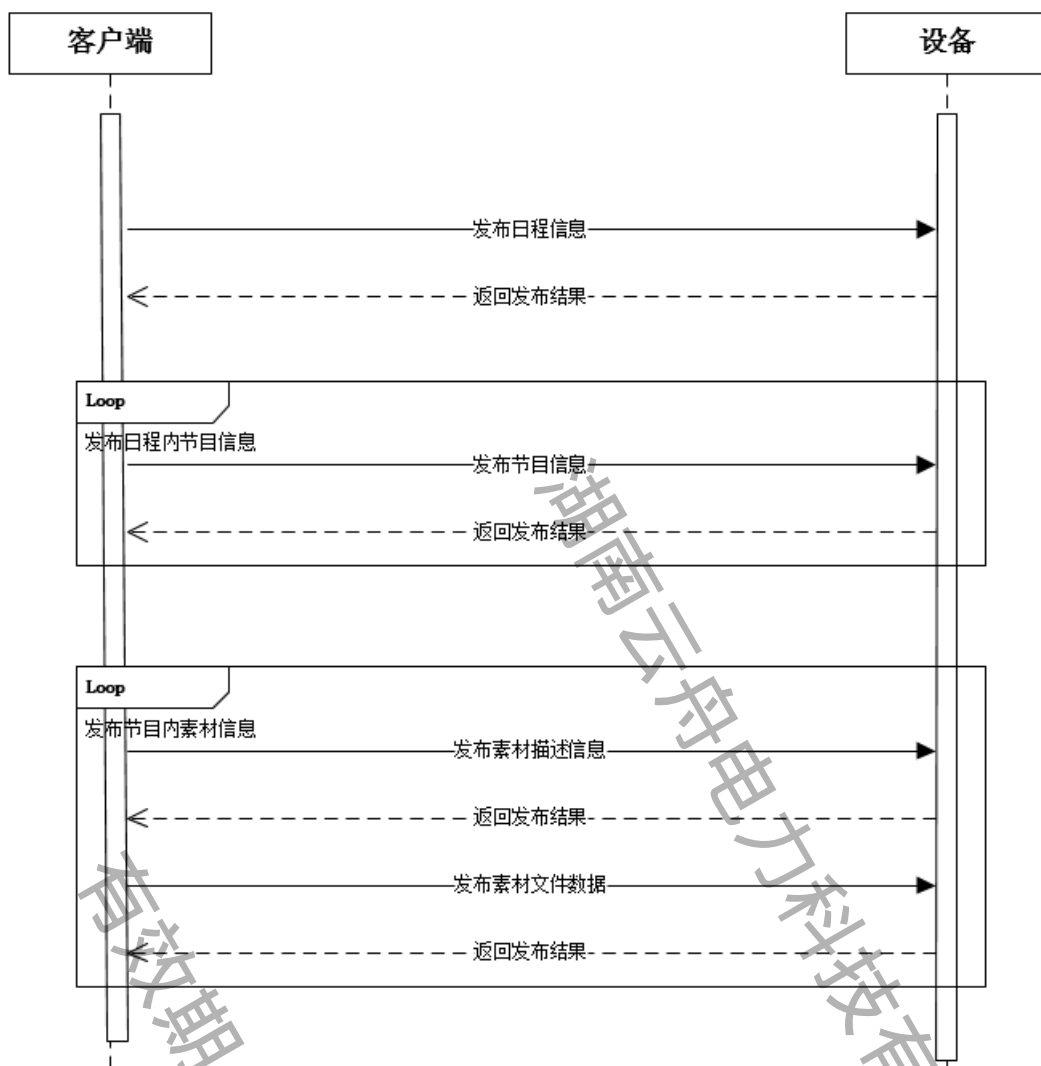
1. 获取系统总能力: `GET /ISAPI/System/capabilities`; 通过返回节点判断设备支持功能情况;
对讲交互事件能力节点: `isSupportVoiceTalkEvent` 返回true表示设备支持对讲交互事件, 返回false或者不返回表示不支持;
2. 对讲交互事件: `eventType: voiceTalkEvent`;
3. 获取对讲总能力: `GET /ISAPI/VideoIntercom/capabilities`; 通过返回节点判断设备支持情况;
通话交互能力节点: `isSupportCallSignal` 返回true表示设备支持通话信令交互, 返回false或者不返回表示不支持;
4. 获取通话信令交互参数能力集: `GET /ISAPI/VideoIntercom/callSignal/capabilities?format=json`;
5. 配置通话信令交互: `PUT /ISAPI/VideoIntercom/callSignal?format=json`;

61 日程发布

61.1 功能介绍

将多个节目，以按日播放、按周播放、自定义播放、轮播播放或垫片播放等方式组合起来，组成日程，并下发到设备。设备在指定的时间会播放指定节目。

61.2 集成流程



1.下发日程通知PUT /ISAPI/Publish/ScheduleMgr/release，通知设备开始接收日程; 3.下发日程参数POST /ISAPI/Publish/ScheduleMgr/playSchedule，通知设备日程中节目的编排时间;

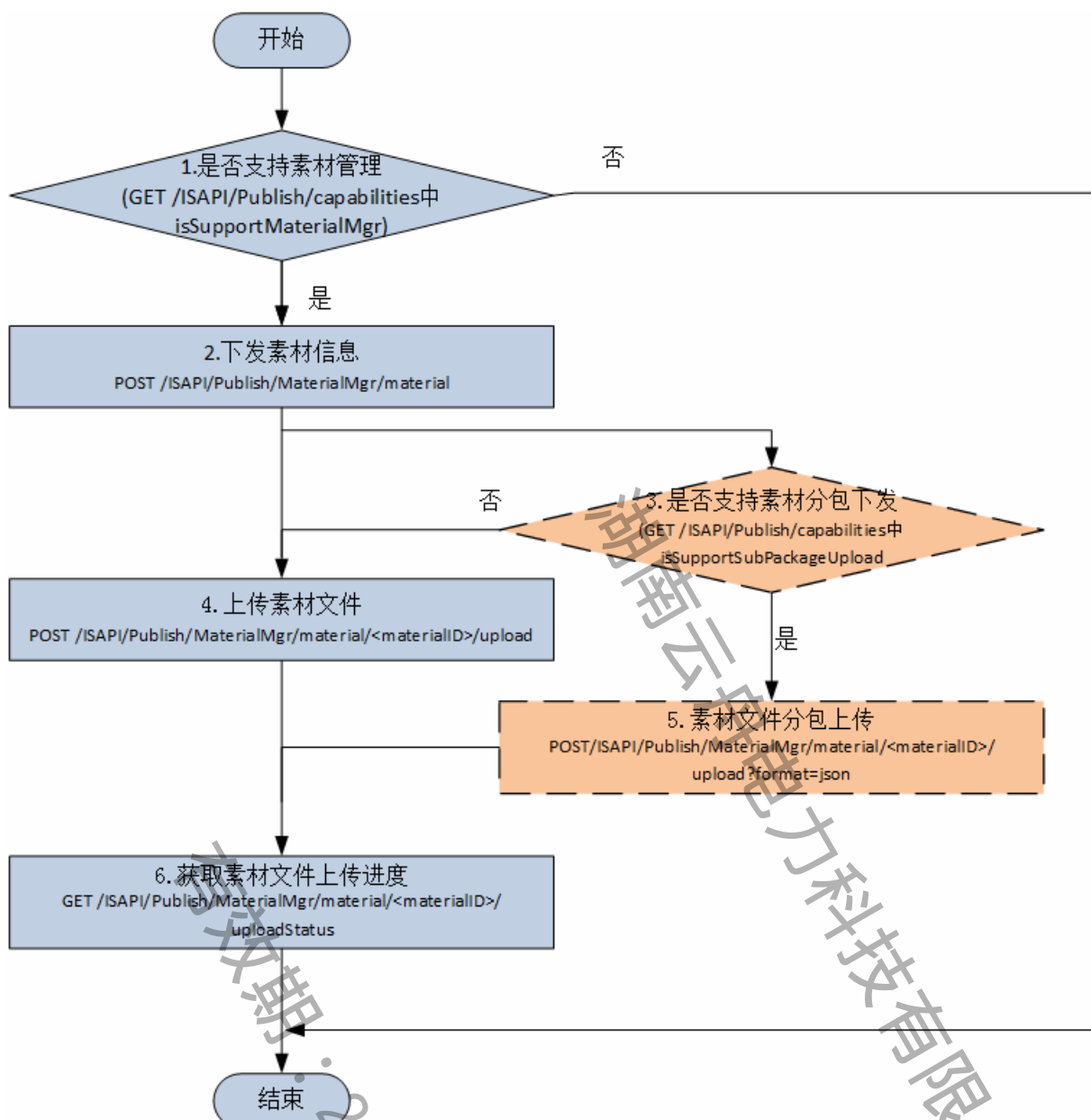
3.客户端向设备循环发布节目和页面信息，包括：下发节目参数POST /ISAPI/Publish/ProgramMgr/program，通知设备节目的名称类型等信息； 4.循环下发页面参数POST /ISAPI/Publish/ProgramMgr/program/<programID>/page，通知设备日程中节目的编排时间; 5.客户端向设备循环发布素材描述信息POST /ISAPI/Publish/MaterialMgr/material.如果用户将素材文件直接上传到设备，则需要下面的 上传素材文件的操作。如果用户的素材文件直接存储在存储服务器上，在报文staticMaterialurl中带上存储URL信息，则不需要下面的操作。 6.上传素材文件 POST /ISAPI/Publish/MaterialMgr/material/<materialID>/upload

62 静态素材上传

62.1 功能介绍

将静态素材文件上传到设备上。

62.2 集成流程



1.判断设备是否支持素材管理GET /ISAPI/Publish/capabilities，返回isSupportMaterialMgr表示支持；2.下发素材信息POST /ISAPI/Publish/MaterialMgr/material，包括素材的名称、描述、素材类型、文件格式等信息。其中输入报文中的素材索引<id>，必须要传；设备成功响应的报文中如果也返回素材索引，则以设备返回的素材索引为准。如果设备不返回，则以上层传入的素材索引为准。

3.判断设备是否支持素材分包下发GET /ISAPI/Publish/capabilities，返回isSupportSubPackageUpload表示支持，当设备支持分包上传，并且上传的素材文件超过2G时，建议采用素材文件分包上传。4.上传素材文件POST /ISAPI/Publish/MaterialMgr/material/<materialID>/upload，URL中的materialID和步骤3中的素材索引一致。5.素材文件分包上传POST/ISAPI/Publish/MaterialMgr/material/<materialID>/upload?format=json，URL中的materialID和步骤3中的素材索引一致。6.获取素材文件上传进度GET /ISAPI/Publish/MaterialMgr/material/<materialID>/uploadStatus。

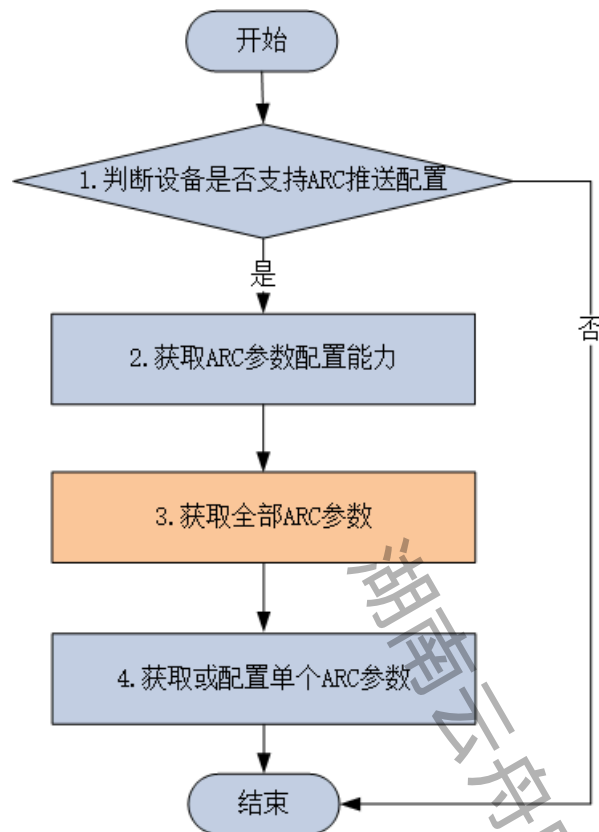
63 ARC管理

63.1 功能介绍

配置ARC参数进入使能状态，并配置ARC推送参数（订阅事件），当订阅的事件发生报警了，对应的报警信息便可推送至ARC平台。目前最多支持4个接警中心，默认编号1和3为主路线，2和4可以设置为1和3的备用路线，若主路线事件推送正常，便不会推送至备用路线，若主路线推送失败，便会将报警信息推送至备用路线。

63.2 ARC参数配置

流程图如下：

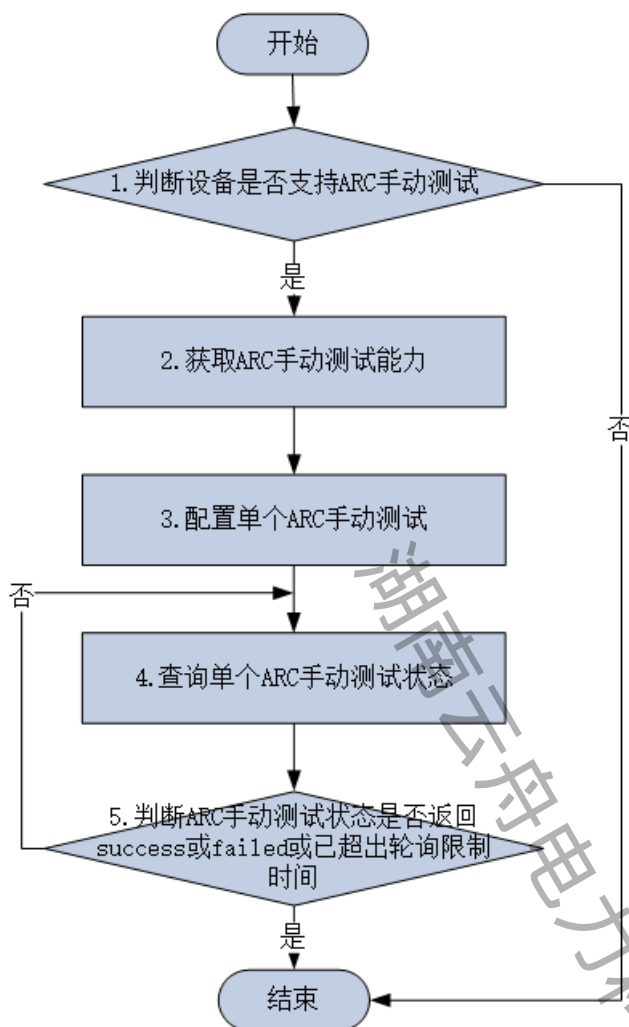


ISAPI协议交互流程： 1.判断设备是否支持是否支持ARC推送配置，获取报警主机配置能力GET /ISAPI/SecurityCP/Configuration/capabilities?format=json判断isSptARC字段是否为true； 2.获取ARC参数配置能力，GET /ISAPI/SecurityCP/Configuration/ARC/capabilities?format=json； 3.获取全部ARC参数，GET /ISAPI/SecurityCP/Configuration/ARC?format=json&security=<security>&iv=<iv>，若设备支持多个ARC，则调用此协议获取，报文中的userName、password字段需进行加密处理； 4.获取或配置单个ARC参数，GET /ISAPI/SecurityCP/Configuration/ARC?format=json&security=<security>&iv=<iv>，报文中的userName、password字段需进行加密处理；配置单个ARC参数，PUT /ISAPI/SecurityCP/Configuration/ARC/<indexID>?format=json&security=<security>&iv=<iv>，其中indexID表示ARC的编号。

63.3 ARC手动测试

完成ARC参数配置后，可对配置好的参数进行手动测试，验证所配置的ARC参数是否正确，链路是否正常。

流程图如下：



ISAPI协议交互流程： 1.判断设备是否支持是否支持ARC手动测试，获取报警主机配置能力GET /ISAPI/SecurityCP/Configuration/capabilities?format=json判断isSptARCManualTest字段是否为true； 2.获取ARC手动测试能力，GET /ISAPI/SecurityCP/Configuration/ARC/manualTest/capabilities?format=json; 3.配置单个ARC手动测试，PUT /ISAPI/SecurityCP/Configuration/ARC/manualTest?format=json，输入报文填入所需要进行测试的ARC编号；

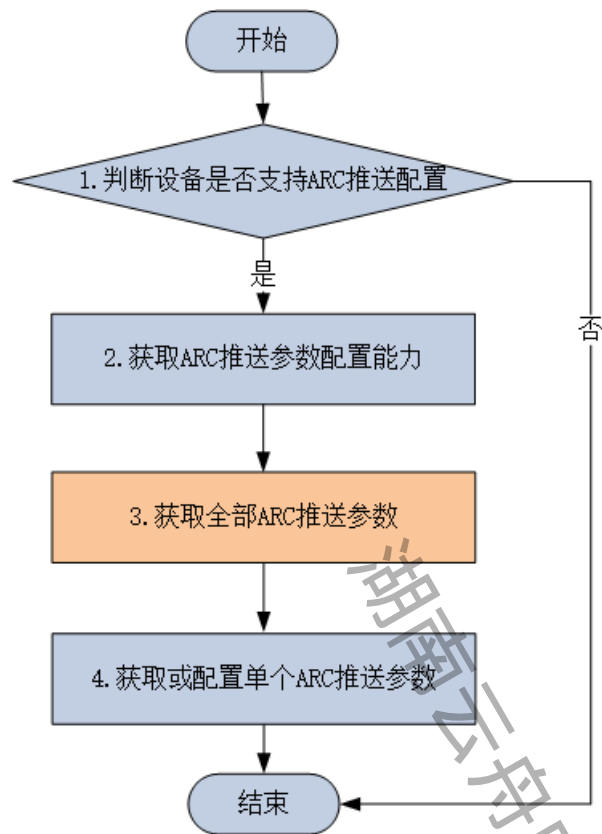
4.查询单个ARC手动测试状态，POST /ISAPI/SecurityCP/Configuration/ARC/manualTest/status?format=json,同样输入报文填入所需要进行状态查询的ARC编号；

5.判断ARC手动测试状态是否返回 success或failed或已超出设备定义的轮询限制时间，返回success测试成功，返回failed测试失败，如果返回processing且未超出轮询限制时间，则进入第4步骤。

63.4 ARC推送参数配置

配置ARC的推送参数，推送项包含报警&防拆事件，生命安全事件，系统状态事件，操作事件，紧急报警事件，医疗报警事件，燃气报警事件等，使能相应项，若发生报警便会推送。

流程图如下：



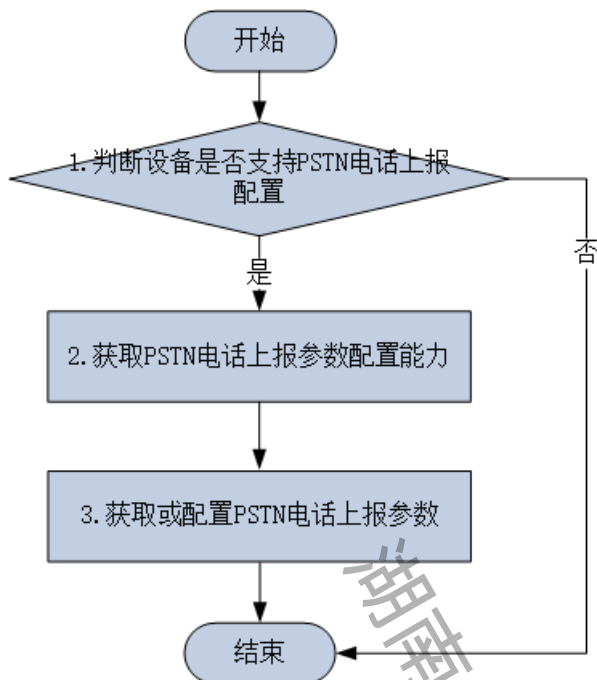
ISAPI协议交互流程：

- 1.判断设备是否支持是否支持ARC推送配置，获取报警主机配置能力GET /ISAPI/SecurityCP/Configuration/capabilities?format=json判断isSptARC字段是否为true； 2.获取ARC推送参数配置能力， GET /ISAPI/SecurityCP/Configuration/messageSendARC/capabilities?format=json；
- 3.获取全部ARC推送参数， GET /ISAPI/SecurityCP/Configuration/ARC?format=json&security=<security>&iv=<iv>，若设备支持多个ARC，则调用此协议获取；
- 4.获取或配置单个ARC推送参数， GET /ISAPI/SecurityCP/Configuration/messageSendARC?id=<indexID>&format=json；配置单个ARC参数， PUT /ISAPI/SecurityCP/Configuration/messageSendARC?id=<indexID>&format=json，其中indexID表示ARC的编号。

64 PSTN电话上报参数管理

若设备支持PSTN，可采用拨号的方式实现报警信息上报，支持配置拨号对象为接警中心或个人。

流程图如下：



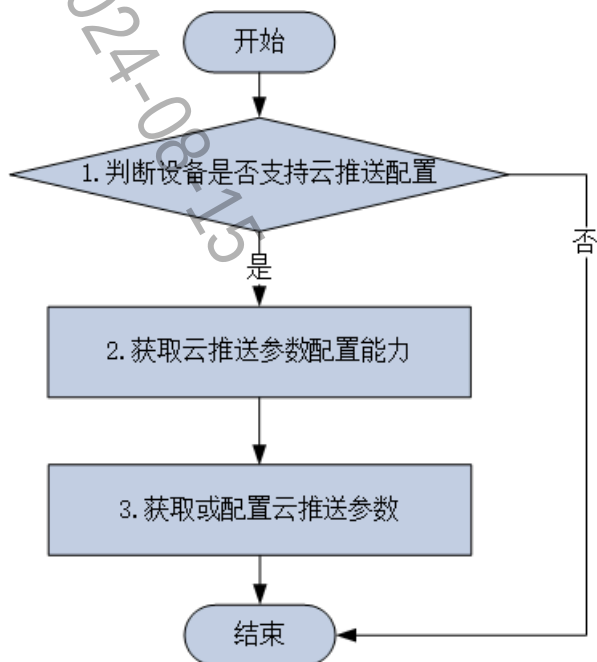
ISAPI协议交互流程:

1.判断设备是否支持PSTN电话上报配置，获取报警主机配置能力GET /ISAPI/SecurityCP/Configuration/capabilities?format=json判断isSptPSTNCfg字段是否为true； 2.获取PSTN电话上报参数配置能力，GET /ISAPI/SecurityCP/Configuration/PSTNCfg/capabilities?format=json； 3.获取全部PSTN电话上报参数，GET /ISAPI/SecurityCP/Configuration/PSTNCfg?format=json； 配置单个PSTN电话上报参数，PUT /ISAPI/SecurityCP/Configuration/PSTNCfg/<indexID>?format=json，可配置PSTN链路是否使能，上报中心的号码，通信协议（目前支持CID）以及传输方式（目前有0#DTMF 5/S,1#DTMF 10/S），其中indexID表示PSTN的编号。

65 云推送参数管理

配置APP云推送参数，推送项包含报警&防拆事件，生命安全事件，系统状态事件，操作事件，紧急报警事件，医疗报警事件，燃气报警事件等，使能相应项，若发生报警便会推送至APP。

流程图如下：



ISAPI协议交互流程:

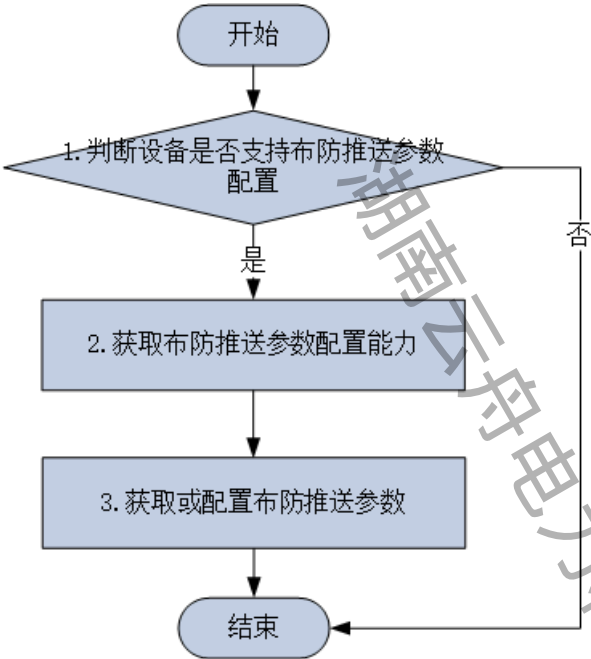
1.判断设备是否支持云推送配置，获取报警主机配置能力GET /ISAPI/SecurityCP/Configuration/capabilities?format=json判断isSptCloud字段是否为true；

- 2.获取云推送参数配置能力，GET /ISAPI/SecurityCP/Configuration/messageSendCloud/capabilities?format=json;
- 3.获取云推送参数，GET /ISAPI/SecurityCP/Configuration/messageSendCloud?format=json; 配置推送参数，PUT /ISAPI/SecurityCP/Configuration/messageSendCloud?format=json。

66 布防推送参数管理

配置布防推送推送参数，推送项包含报警&防拆事件，生命安全事件，系统状态事件，操作事件，紧急报警事件，医疗报警事件，燃气报警事件等。如IVMS-4200，布防后，使能相应项，若发生报警便会推送至IVMS-4200。

流程图如下：



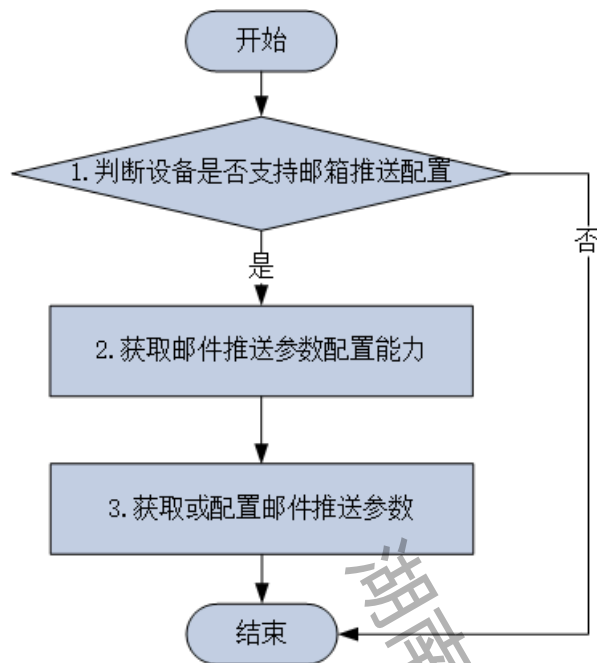
ISAPI协议交互流程：

- 1.判断设备是否支持布防推送参数配置，获取报警主机配置能力GET /ISAPI/SecurityCP/Configuration/capabilities?format=json判断isSptDirect字段是否为true;
- 2.获取布防推送参数配置能力，GET /ISAPI/SecurityCP/Configuration/messageSendDirect/capabilities?format=json;
- 3.获取布防推送参数，GET /ISAPI/SecurityCP/Configuration/messageSendDirect?format=json; 配置布防推送参数，PUT /ISAPI/SecurityCP/Configuration/messageSendDirect?format=json。

67 邮件推送参数管理

完成邮箱参数配置后，若设备支持邮件推送，使能相应报警项（支持报警&防拆事件推送，PC断开事件推送），若发生报警便会推送邮件至邮箱。

流程图如下：



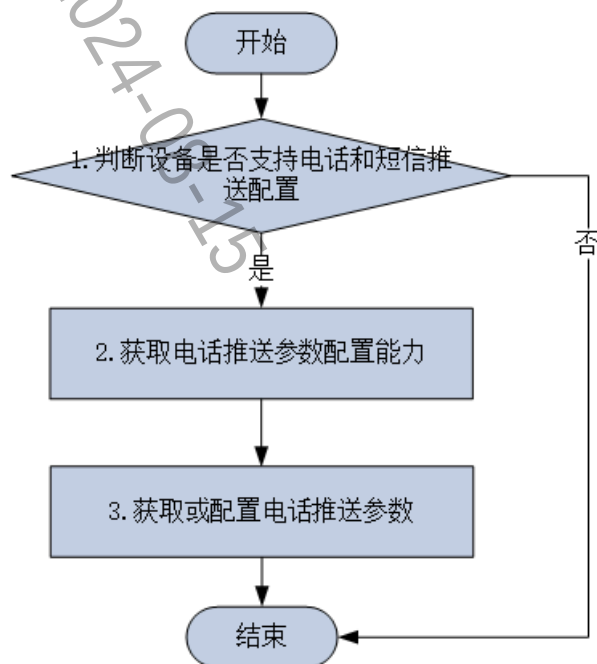
ISAPI协议交互流程：

- 1.判断设备是否支持邮箱推送配置，获取报警主机配置能力GET /ISAPI/SecurityCP/Configuration/capabilities?format=json判断iisSptMail字段是否为true;
- 2.获取邮件推送参数配置能力，GET /ISAPI/SecurityCP/Configuration/messageSendMail/capabilities?format=json;
- 3.获取全部邮件推送参数，GET /ISAPI/SecurityCP/Configuration/PSTNCfg?format=json；配置单个邮件推送参数，PUT /ISAPI/SecurityCP/Configuration/messageSendMail/<mailID>?format=json，URL中mailID表示邮箱索引。

68 电话推送参数管理（弃用，电话推送高级参数配置替之）

配置电话推送参数，推送项包括报警&防拆事件推送，生命安全事件推送，系统状态事件推送，操作事件推送，支持短信和电话提示。

流程图如下：



ISAPI协议交互流程：

- 1.判断设备是否支持电话和短信推送配置，获取报警主机配置能力GET /ISAPI/SecurityCP/Configuration/capabilities?format=json判断isSptPhone字段是否为true;

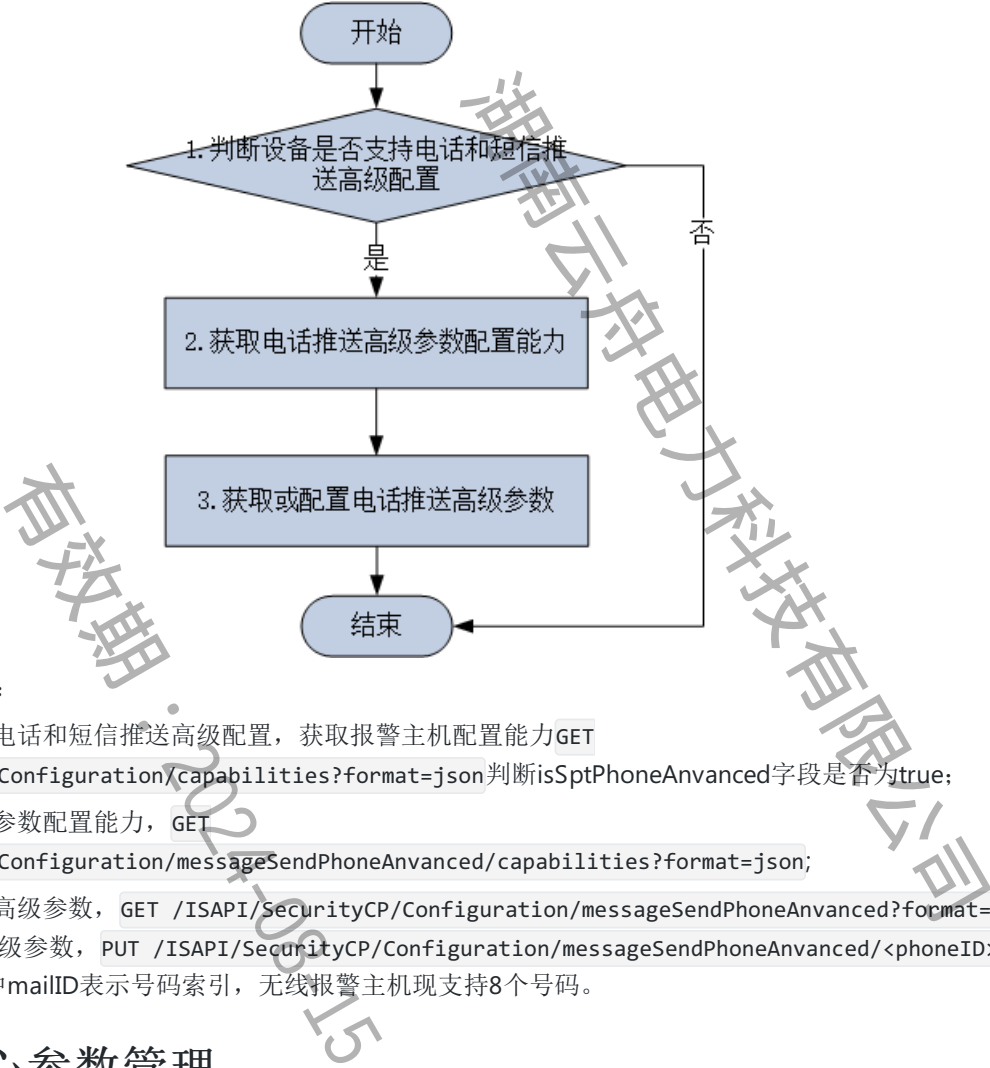
2.获取电话推送参数配置能力，GET /ISAPI/SecurityCP/Configuration/messageSendPhone/capabilities?format=json;

3.获取全部电话推送参数，GET /ISAPI/SecurityCP/Configuration/messageSendPhone?format=json; 获取全部电话推送参数，PUT /ISAPI/SecurityCP/Configuration/messageSendPhone/<phoneID>?format=json，URL中phoneID表示号码索引。

10 电话推送高级参数管理

配置电话推送高级参数，在原来电话推送参数管理的基础上，新增了紧急报警事件，医疗报警事件，燃气报警事件等项，电话短信时间段过滤参数配置，支持对每个号码进行布防，撤防，消警权限配置。

流程图如下：



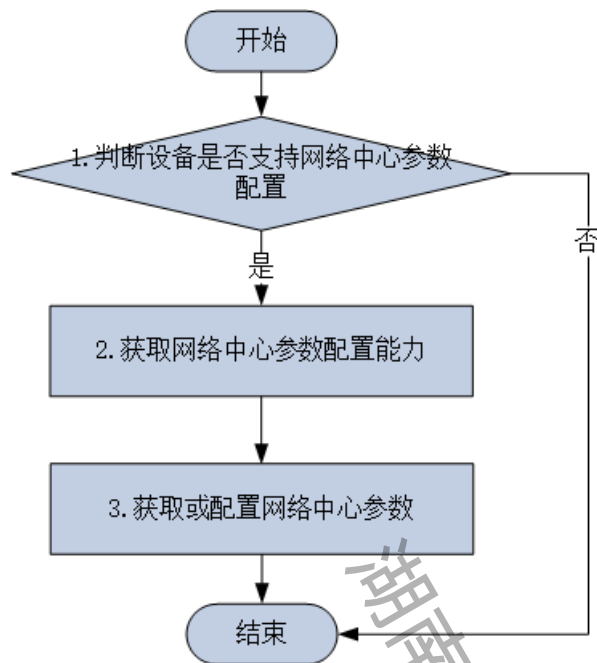
ISAPI协议交互流程：

- 1.判断设备是否支持电话和短信推送高级配置，获取报警主机配置能力GET /ISAPI/SecurityCP/Configuration/capabilities?format=json判断isSptPhoneAnvanced字段是否为true;
- 2.获取电话推送高级参数配置能力，GET /ISAPI/SecurityCP/Configuration/messageSendPhoneAnvanced/capabilities?format=json;
- 3.获取全部电话推送高级参数，GET /ISAPI/SecurityCP/Configuration/messageSendPhoneAnvanced?format=json; 配置单个电话推送高级参数，PUT /ISAPI/SecurityCP/Configuration/messageSendPhoneAnvanced/<phoneID>?format=json，URL中mailID表示号码索引，无线报警主机现支持8个号码。

11 网络中心参数管理

网络中心参数配置，可实现对设备内每一个网卡，分别配置多个网络中心参数，即网卡与网络中心可以是一对多的关系，目前支持私有协议,NAL2300协议,Ehome协议。

流程图如下：



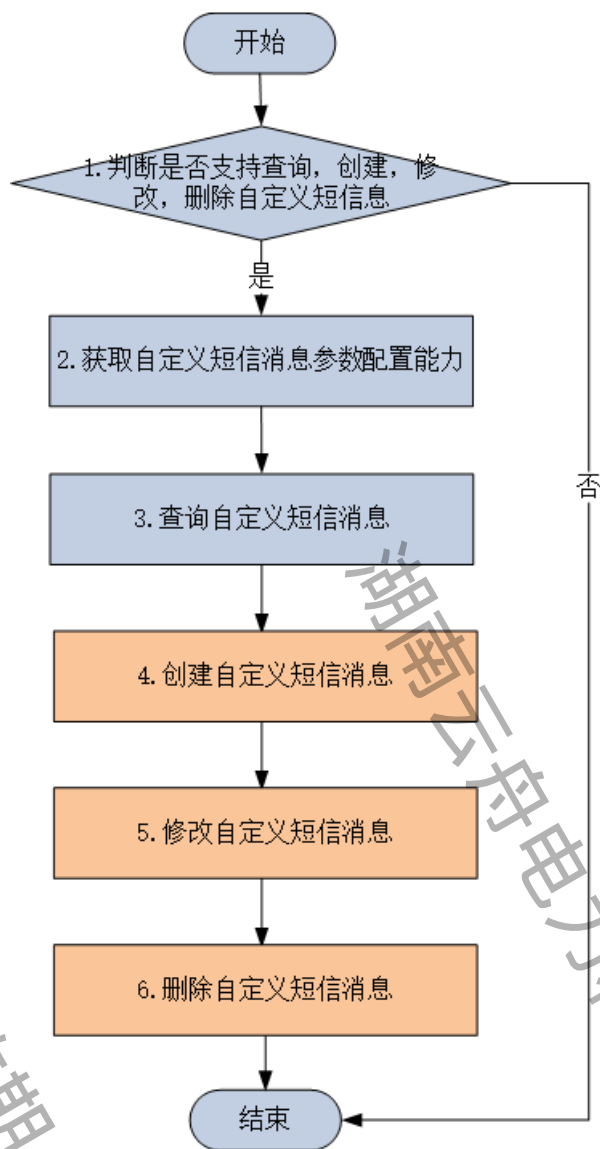
ISAPI协议交互流程：

- 1.判断设备是否支持电话和短信推送高级配置，获取报警主机配置能力GET /ISAPI/SecurityCP/Configuration/capabilities?format=json判断isSptNetCfg字段是否为true;
- 2.获取网络中心参数配置能力，GET /ISAPI/SecurityCP/NetCfg/capabilities?format=json;
- 3.获取网络中心参数，GET /ISAPI/SecurityCP/NetCfg/<interfaceID>?format=json；配置网络中心参数，PUT /ISAPI/SecurityCP/NetCfg/<interfaceID>?format=json，URL中interfaceID表示网卡编号，1为主网卡，2为扩展网卡。

12 自定义短信息管理

电话推送形式支持自定义语音（自定义语音集成流程见“报警主机自定义语音”类目，目前暂未完成编写）和自定义短信导入。

流程图如下：



ISAPI协议交互流程:

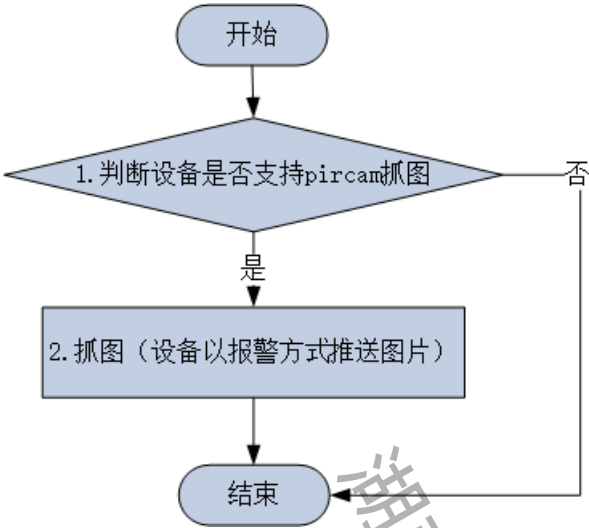
- 1.判断设备是否支持查询、创建、修改、删除自定义短信息，获取报警主机配置能力GET /ISAPI/SecurityCP/Configuration/capabilities?format=json 分别判断 isSupportSearchCustomAudioList, isSupportAddCustomMessage, isSupportModifyCustomMessage, isSupportDeleteCustomMessage 字段是否为true;
- 2.获取自定义短信消息参数配置能力，GET /ISAPI/SecurityCP/customMessage/capabilities?format=json;
- 3.获取全部自定义短信消息，GET /ISAPI/SecurityCP/customMessage/searchCustomMessageList?format=json&security=<security>&iv=<iv>，支持customMessageContent字段加密处理;
- 4.创建单个自定义短信消息（可选），POST /ISAPI/SecurityCP/customMessage/addCustomMessage?format=json&security=<security>&iv=<iv>;
- 5.修改单个自定义短信消息（可选），PUT /ISAPI/SecurityCP/customMessage/modifyCustomMessage?format=json&security=<security>&iv=<iv>;
- 6.删除单个自定义短信消息（可选），POST /ISAPI/SecurityCP/customMessage/deleteCustomMessage?format=json。

13 指定防区抓图

目前pircam针对防区防区的主动抓图，有两种方式，1.下发抓图命令，图片以报警cid方式上传图片URL；2，异步抓图，首先下发抓图命令，成功后循环调用获取协议，获取图片URL。

13.1 抓图（设备以报警方式推送图片）

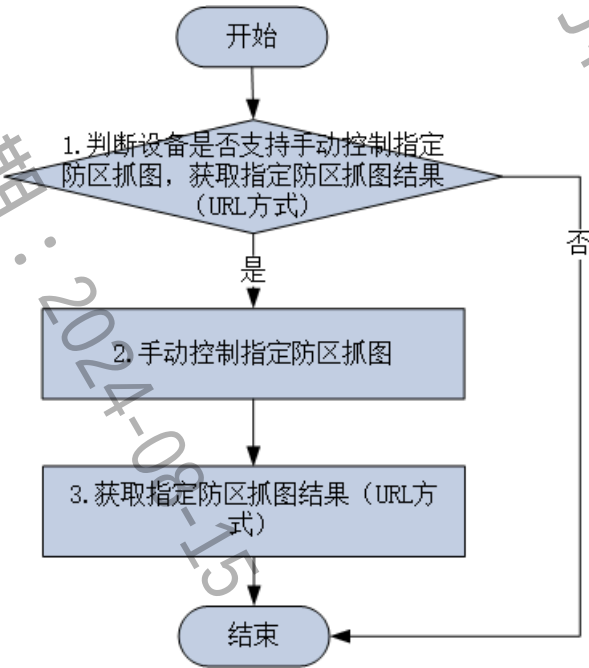
ircam手动抓图触发，设备如果支持，返回成功响应，否则，返回不支持的错误，设备端产生的抓图通过报警上传（CID事件）的方式主动推送（当前仅支持萤石推送，不支持ISAPI协议推送）。



ISAPI协议交互流程：

- 1.判断设备是否支持pircam抓图，获取报警主机配置能力GET /ISAPI/SecurityCP/Configuration/capabilities?format=json分别判断isSptPircamCapture字段是否为true;
- 2.抓图（设备以报警方式推送图片），GET /ISAPI/Streaming/channels/<zoneID>/picture/devicePush，zoneID表示防区号;

13.2 抓图（URL方式异步返回）



ISAPI协议交互流程：

- 1.判断设备是否支持手动控制指定防区抓图，获取指定防区抓图结果（URL方式），获取报警主机总能力GET /ISAPI/SecurityCP/capabilities?format=json分别判断isSptManualControlCapture，isSptGetPictureByURL字段是否为true;
- 2.手动控制指定防区抓图，PUT /ISAPI/SecurityCP/Zone/<zoneID>/Capture/ManualControlCapture?format=json，zoneID表示防区号;
- 3.获取指定防区抓图结果（URL方式），GET /ISAPI/SecurityCP/Zone/<zoneID>/Capture/GetPictureByURL?format=json。

69 功能接口

69.1 信息管理

69.1.1 配置设备信息参数

Request URL

PUT /ISAPI/System/deviceInfo

查询参数

无

请求报文

```
<?xml version="1.0" encoding="UTF-8"?>

<DeviceInfo xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 设备信息, attr:version{opt, string, 协议版本}-->
  <deviceName>
    <!--req, string, 设备名称, range:[1,128]-->test
  </deviceName>
  <deviceId>
    <!--req, string, 设备编号, range:[1,128]-->test
  </deviceId>
  <deviceDescription>
    <!--opt, string, 设备描述, range:[1,128]-->test
  </deviceDescription>
  <deviceLocation>
    <!--opt, string, 设备位置, range:[1,128]-->hangzhou
  </deviceLocation>
  <deviceStatus>
    <!--ro, opt, enum, 设备状态, subType:string, [normal#正常,abnormal#异常]-->normal
  </deviceStatus>
  <DetailAbnormalStatus>
    <!--ro, opt, object, 详细异常状态, desc: 仅当deviceStatus为abnormal时有效-->
    <hardDiskFull>
      <!--ro, opt, bool, 盘满-->true
    </hardDiskFull>
    <hardDiskError>
      <!--ro, opt, bool, 盘错-->true
    </hardDiskError>
    <ethernetBroken>
      <!--ro, opt, bool, 网线断-->true
    </ethernetBroken>
    <ipaddrConflict>
      <!--ro, opt, bool, IP地址冲突-->true
    </ipaddrConflict>
    <illegalAccess>
      <!--ro, opt, bool, 非法访问-->true
    </illegalAccess>
    <recordError>
      <!--ro, opt, bool, 录像异常-->true
    </recordError>
    <raidLogicDiskError>
      <!--ro, opt, bool, 阵列虚拟磁盘异常-->true
    </raidLogicDiskError>
    <spareWorkDeviceError>
      <!--ro, opt, bool, 热备监控的工作机异常-->true
    </spareWorkDeviceError>
  </DetailAbnormalStatus>
  <systemContact>
    <!--ro, opt, string, 生产商, range:[1,32]-->STD-CGI
  </systemContact>
  <model>
    <!--ro, req, string, 设备型号, range:[1,64]-->iDS-9632NX-I8/X
  </model>
  <serialNumber>
    <!--ro, req, string, 长序列号, range:[1,48]-->iDS-9632NX-I8/X1620181209CCRRC77605411WCVU
  </serialNumber>
  <macAddress>
    <!--ro, req, string, 物理地址, range:[1,64]-->44:47:cc:c8:d9:e4
  </macAddress>
  <firmwareVersion>
    <!--ro, req, string, 主控版本号, range:[1,64]-->V4.1.40
  </firmwareVersion>
  <firmwareReleasedDate>
    <!--ro, opt, string, 主控版本日期-->2019-11-01
  </firmwareReleasedDate>
  <encoderVersion>
    <!--ro, opt, string, 编码版本, range:[1,32]-->V7.3
  </encoderVersion>
  <encoderReleasedDate>
    <!--ro, opt, string, 编码版本日期-->2019-11-02
  </encoderReleasedDate>
  <bootVersion>
    <!--ro, opt, string, boot版本, range:[1,16]-->V1.3.4
  </bootVersion>
  <bootReleasedDate>
    <!--ro, opt, string, boot发布日期-->2019-11-03
  </bootReleasedDate>
</DeviceInfo>
```

```
</bootReleasedDate>
<panelVersion>
  <!--ro, opt, string, 面板版本, range:[1,32]-->V1.0
</panelVersion>
<hardwareVersion>
  <!--ro, opt, string, 硬件版本, range:[1,16]-->0x0
</hardwareVersion>
<decoderVersion>
  <!--ro, opt, string, 解码器版本, range:[1,32]-->V1.0
</decoderVersion>
<decoderReleasedDate>
  <!--ro, opt, string, 解码器版本日期-->2019-01-01
</decoderReleasedDate>
<softwareVersion>
  <!--ro, opt, string, 软件版本, range:[1,32]-->V1.23
</softwareVersion>
<capacity>
  <!--ro, opt, int, 设备容量, range:[1,10240], desc:单位 MB-->1
</capacity>
<usedCapacity>
  <!--ro, opt, int, 设备已使用容量, range:[1,10240], desc:单位 MB-->1
</usedCapacity>
<deviceType>
  <!--ro, req, enum, 设备类型, subType:string, [Blade#刀锋, BladePS#治安刀锋, CVR#CVR, DM#深思, DMbehavior#行为分析设备, DVR#硬盘录像机, DVS#编码器, DockStation#
  采集站, FA#人脸实时比对报警服务器, FD#抓拍检测服务器, FS#人脸静态数据服务器, HAWK#猎鹰服务器, HMCP#模型比对服务器, HybirdNVR#混合型硬盘录像机, IPCamera#网络摄像
  机, IPDome#网络球机, IPZoom#一体化摄像机, InfoReleaseSys#信息发布系统, InfoTerminal#信息发布终端, MCU#MCU, NVR#NVR, PHA#一代混合报警主机, PURE#纯分析服务
  器, RFID#RFID, SipServer#Sip服务器, Switch#交换机, LightFace#轻量级脸谱, Visitor#访客机, PWA#一代无线报警主机, PHAPro#二代混合报警主机, ACS#门禁设备, VIS#对讲设
  备, FacePaymentTerminal#消费机, InteractiveTerminal#智能交互终端, Cabinet#智能柜, VoiceSpeaker#音柱音箱, PowerAmplifier#功放, FireControlMatrix#消防矩
  阵, PagingMicrophone#寻呼话筒, ElevatorControl#梯控主机, conferencePlat#会议平板, AIOT#智能物联网主机(Artificial Intelligence & Internet of Things), OPCA#光学理
  化分析装置(Optical physical and chemical analyzer), NetworkReceiver#网络接警机, PanicAlarmPanel#一键式紧急报警求助设备, bodyCamera#执法记录仪, videoMatrix#视频
  综合平台, EmbeddedCentralController#物联中控主机, networkMic#网络麦克风, IPA#模拟和音频产品, PersonnelChannel#人员通道, MediaComponentGateway#媒体组件网
  关, securityRadar#态势感知雷达]-->Blade
</deviceType>
<telecontrolID>
  <!--opt, int, 遥控器ID, range:[1,255]-->1
</telecontrolID>
<supportBeep>
  <!--ro, opt, bool, 是否支持蜂鸣器-->true
</supportBeep>
<supportVideoLoss>
  <!--ro, opt, bool, 是否支持视频丢失-->true
</supportVideoLoss>
<firmwareVersionInfo>
  <!--ro, opt, string, 主控版本信息, range:[1,32]-->test
</firmwareVersionInfo>
<actualFloorNum>
  <!--ro, opt, int, 实际层数, range:[1,128]-->1
</actualFloorNum>
<localZoneNum>
  <!--ro, opt, int, 本地防区数, range:[0,16]-->1
</localZoneNum>
<alarmOutNum>
  <!--ro, opt, int, 报警时输出数, range:[0,16]-->1
</alarmOutNum>
<distanceResolution>
  <!--ro, opt, float, 距离分辨率, range:[0.000,0.999]-->0.000
</distanceResolution>
<angleResolution>
  <!--ro, opt, float, 角分辨率, range:[0.000,0.999]-->0.000
</angleResolution>
<speedResolution>
  <!--ro, opt, float, 速度分辨率, range:[0.000,0.999]-->0.000
</speedResolution>
<detectDistance>
  <!--ro, opt, float, 检测距离, range:[0.000,0.999]-->0.000
</detectDistance>
<relayNum>
  <!--ro, opt, int, 本地继电器数(case)数量, range:[0,16]-->1
</relayNum>
<electroLockNum>
  <!--ro, opt, int, 本地电锁数, range:[0,16]-->1
</electroLockNum>
<sirenNum>
  <!--ro, opt, int, 本地警号数, range:[0,16]-->1
</sirenNum>
<alarmLamp>
  <!--ro, opt, int, 警灯数, range:[0,16]-->1
</alarmLamp>
<RS485Num>
  <!--ro, opt, int, 本地485数, range:[0,16]-->1
</RS485Num>
<radarVersion>
  <!--ro, opt, string, 雷达版本, range:[1,32]-->test
</radarVersion>
<cameraModuleVersion>
  <!--ro, opt, string, 相机模组版本, range:[1,32]-->test
</cameraModuleVersion>
<mainversion>
  <!--ro, opt, int, 主版本号, range:[1,255]-->1
</mainversion>
<subversion>
  <!--ro, opt, int, 子版本号, range:[1,255]-->1
</subversion>
<unmadedevision>
```

```

<!--ro, opt, int, 升级版本号, range:[1,255]-->1
</upgradeversion>
<customizeversion>
<!--ro, opt, int, 定制版本号, range:[1,255]-->1
</customizeversion>
<companyName>
<!--ro, opt, string, 生产公司简称, range:[1,32]-->test
</companyName>
<copyright>
<!--ro, opt, string, 版权信息, range:[1,32]-->test
</copyright>
<systemName>
<!--ro, opt, enum, 存储系统名称, subType:string, [storageManagement#存储管理系统,distributedStorageManagement#分布式存储管理系统]-->storageManagement
</systemName>
<systemStatus>
<!--ro, opt, enum, 系统状态, subType:string, [configured#已配置,unConfigured#未配置]-->configured
</systemStatus>
<isLeaderDevice>
<!--ro, opt, bool, 是否是资源IP对应设备-->true
</isLeaderDevice>
<clusterVersion>
<!--ro, opt, string, 系统集群版本, range:[1,32], desc:当前isLeaderDevice字段返回true时有效-->test
</clusterVersion>
<manufacturer>
<!--ro, opt, enum, 厂商信息, subType:string, [hikvision#海康设备,#中性设备,Hostone#皓视通设备], desc:用于OEM设备区分;海康设备必须返回"hikvision", 中性设备返回"", 皓视通设备返回Hostone, 其他OEM/厂商名称需要找SDK组申请-->hikvision
</manufacturer>
<centralStorageVersion>
<!--ro, opt, string, 中心存储版本, range:[1,16]-->test
</centralStorageVersion>
<powerOnMode>
<!--ro, opt, enum, 启动模式, subType:string, [button#按键启动,adapter#接入电源适配器启动]-->button
</powerOnMode>
<customizedInfo>
<!--ro, opt, string, 定制项目单号, range:[1,32], desc:基线设备节点返回空, 定制设备返回定制项目单号-->test
</customizedInfo>
<verificationCode>
<!--ro, opt, string, 设备验证码-->test
</verificationCode>
<supportUrl>
<!--ro, opt, string, 服务门户网站-->test
</supportUrl>
<subSerialNumber>
<!--ro, opt, string, 子序列号-->test
</subSerialNumber>
<languageType opt="chinese,english,spanish,portuguese,italian,french,russian,turkish,greek,czech,brazilianPortuguese">
<!--ro, opt, enum, 语言类型, subType:string, [chinese#中文,english#英语,spanish#西班牙语,portuguese#葡萄牙语,italian#意大利语,french#法语,russian#俄语,turkish#土耳其语,greek#希腊语,czech#捷克语,brazilianPortuguese#巴西葡萄牙语,slovenian#斯洛文尼亚语,swedish#瑞典语,norwegian#挪威语,romanian#罗马尼亚语,danish#丹麦语,german#德语,polish#波兰语,dutch#荷兰语,hungarian#匈牙利语,slovak#斯洛伐克语,serbian#塞尔维亚语,southAmericanSpanish#南美西班牙语,ukrainian#乌克兰语,croatian#克罗地亚语,irish#爱尔兰语,bulgarian#保加利亚语,hebrew#希伯来语,thai#泰语,indonesian#印尼语,arabic#阿拉伯语,traditionalChinese#繁体中文,lithuanian#立陶宛语,anglicism#英式英语], attr:opt{req, string, 取值范围}-->chinese
</languageType>
<DockStation>
<!--ro, opt, object, 采集站信息配置-->
<Platform>
<!--ro, opt, object, 采集站对接平台信息配置-->
<type>
<!--ro, req, enum, 采集站对接平台, subType:string, [none#无,9533#9533,8618#8618,ISAPI#ISAPI]-->none
</type>
<ip>
<!--ro, opt, string, IP地址, range:[1,32]-->test
</ip>
<port>
<!--ro, opt, int, 通信端口, range:[1,65535]-->1
</port>
<userName>
<!--ro, req, string, 用户名, range:[1,32]-->test
</userName>
<password>
<!--ro, req, string, 密码, range:[1,16]-->test
</password>
</Platform>
<centralStorageBackupEnabled>
<!--ro, opt, bool, 中心存储备份开启使能-->true
</centralStorageBackupEnabled>
</DockStation>
<webVersion>
<!--ro, opt, string, Web版本号, range:[1,32]-->test
</webVersion>
<deviceRFProgramVersion>
<!--ro, opt, string, 设备射频程序版本号, range:[1,32]-->test
</deviceRFProgramVersion>
<securityModuleSerialNo>
<!--ro, opt, string, 安全模块序列号, range:[1,32]-->test
</securityModuleSerialNo>
<securityModuleVersion>
<!--ro, opt, string, 安全模块版本号, range:[1,32]-->test
</securityModuleVersion>
<securityChipVersion>
<!--ro, opt, string, 安全芯片版本号, range:[1,32]-->test
</securityChipVersion>
<securityModuleKeyVersion>
<!--ro, opt, string, 安全模块秘钥版本号, range:[1,32]-->test

```

```
</securityModuleKeyVersion>
<UIDLampRecognition>
  <!--ro, opt, object, uid灯识别设备信息-->
  <enabled>
    <!--ro, opt, bool, 使能配置-->true
  </enabled>
</UIDLampRecognition>
<confDeviceIdPrefix>
  <!--ro, opt, bool, 是否会议带设备名称做前缀-->true
</confDeviceIdPrefix>
<OEMCode>
  <!--ro, opt, enum, 厂商OME代码, subType:int, [1#海康设备,0#中性设备], desc:海康设备返回1, 中性设备返回0, 其他厂商根据SDK定义返回-->1
</OEMCode>
<simpleAlgorithmVersion>
  <!--ro, opt, string, 单算法版本, desc:智慧消防分析仪设备信息中返回这个字段表明分析仪只支持一种识别算法, 出厂时软件已写死并且不支持录像存储、回放、数据库
查询、人脸库、实时预览等功能-->test
</simpleAlgorithmVersion>
<bootTime>
  <!--ro, opt, datetime, 系统启动时间-->1970-01-01T00:00+08:00
</bootTime>
<intelligentAnalysisEngineModel>
  <!--ro, opt, string, 引擎型号, range:[1,32]-->test
</intelligentAnalysisEngineModel>
<marketType>
  <!--ro, opt, enum, 市场类型, subType:int, [0#无效(未知类型),1#经销型,2#行业型]-->0
</marketType>
<algorithmVersion>
  <!--ro, opt, string, HCN算法版本, desc:该数值的字典由研究院统一提供,用于算法模型的匹配-->test
</algorithmVersion>
<firmware>
  <!--ro, opt, string, 固件版本, desc:该数值的字典由研究院统一提供,用于算法模型的匹配-->test
</firmware>
<engineList>
  <!--ro, opt, object, 设备计算能力列表, desc:以GPU芯片个数来区分,返回engine的编号,例如有3个芯片,编号分别为1、5、11,则返回的值为1、5、11-->
  <engine>
    <!--ro, opt, int, 返回的engine的编号-->1
  </engine>
</engineList>
<platform>
  <!--ro, opt, enum, 平台类型, subType:int, [1#TX1,2#P4,3#3559,4#3519,5#3516]-->1
</platform>
<platformName>
  <!--ro, opt, string, 算法运行平台对外展现名称, desc:"K81,TX1,KT,H5,H7,K82,G3",产品自定义,该字段仅用于直接展现TX1/K81-后端NVR,KT-数据中心,H5-前端,H7-前
端,K82-后端NVR,G3-前端-->test
</platformName>
<touchScreenVersionInfo>
  <!--ro, opt, string, 触摸屏组件版本-->test
</touchScreenVersionInfo>
<protocolFileURL>
  <!--ro, opt, string, 协议须知文件URL, range:[1,32]-->test
</protocolFileURL>
<recycleRecordEnabled>
  <!--ro, opt, bool, 是否循环录像使能-->>false
</recycleRecordEnabled>
<decordChannelNums>
  <!--ro, opt, int, 解码通道数量-->0
</decordChannelNums>
<VGANums>
  <!--ro, opt, int, VGA口个数-->0
</VGANums>
<USBNums>
  <!--ro, opt, int, usb口个数-->0
</USBNums>
<auxoutNums>
  <!--ro, opt, int, 辅助口个数-->0
</auxoutNums>
<expansionBoardVersion>
  <!--ro, opt, string, 扩展板版本信息, range:[1,32]-->test
</expansionBoardVersion>
<initWizzardDisplay>
  <!--ro, opt, bool, 是否显示初始化向导, desc:仅用于(集成了向导的应用,如本地GUI,海康互联APP)显示配置。默认为true,当设备本地GUI配置完向导之后,状态改为
false,海康互联APP每次(重新)添加设备获取一次状态,判断是否显示向导界面-->true
</initWizzardDisplay>
<beaconID>
  <!--ro, opt, string, (ETC的扩展单元)设备射频程序版本号, range:[0,32], desc:默认填写当前值-->test
</beaconID>
<isResetDeviceLanguage>
  <!--ro, opt, bool, 是否重置设备语言, desc:在对主机进行切换语言升级时(仅Admin、Installer有权限),当置为true,表示HC和Web按照切换后的语言显示设备信息;
若置为false,表示HC和Web仍按照切换前的语言显示设备信息。-->>false
</isResetDeviceLanguage>
<dispalyNum>
  <!--ro, opt, int, 设备屏幕个数-->0
</dispalyNum>
<bspVersion>
  <!--ro, opt, string, BSP软件版本-->test
</bspVersion>
<dspVersion>
  <!--ro, opt, string, DSP软件版本-->test
</dspVersion>
<localUIVersion>
  <!--ro, opt, string, 本地UI版本-->test
</localUIVersion>
<detectorType>
```

```
<!--ro, opt, enum, 检测器类型, subType:string, [HUA#主板阵列检测器]-->HUA
</detectorType>
<wiegandOutNum>
  <!--ro, opt, int, 韦根输出口个数, range:[0,1]-->1
</wiegandOutNum>
<ChipVersionInfoList>
  <!--ro, opt, object, 芯片版本信息列表-->
  <ChipVersionInfo>
    <!--ro, opt, object, 芯片版本信息-->
    <ID>
      <!--ro, req, int, 芯片编号, range:[0,1], desc:不同类型的芯片编号可以重复,用户可根据芯片类型+芯片编号区分-->1
    </ID>
    <type>
      <!--ro, req, enum, 芯片类型, subType:string, [MCU#MCU芯片,FPGA#FPGA芯片,MSTAR#MSTAR芯片]-->MCU
    </type>
    <firmwareVersion>
      <!--ro, req, string, 芯片固件版本, range:[0,64]-->1.0.0
    </firmwareVersion>
  </ChipVersionInfo>
</ChipVersionInfoList>
<personBagLinkAlgoEngineVersion>
  <!--ro, opt, string, 人包关联模块的引擎版本, range:[0,64], desc:安检分析仪中人包关联模块的算法引擎版本-->1.0.0
</personBagLinkAlgoEngineVersion>
<BIOSVersion>
  <!--ro, opt, string, BIOS版本, range:[0,16]-->test
</BIOSVersion>
<contactInformation>
  <!--ro, opt, string, 联系方式, range:[0,64]-->test
</contactInformation>
<materialScanAlgorithmVersion>
  <!--opt, string, 物料扫描算法版本, range:[0,64], desc:成像雷达主机用于物料扫描的算法版本-->test
</materialScanAlgorithmVersion>
</DeviceInfo>
```

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 响应消息, attr:version{ro, req, string, 版本}-->
  <requestURL>
    <!--ro, req, string, 请求的URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, 状态码, subType:int, [0#OK,1#OK,2#Device Busy,3#Device Error,4#Invalid Operation,5#Invalid XML Format,6#Invalid XML Content,7#Reboot Required]-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, 状态信息, subType:string, [OK#成功,Device Busy#设备忙,Device Error#设备异常,Invalid Operation#无效的操作,Invalid XML Format#无效的XML格式,Invalid XML Content#无效的XML内容,Reboot#设备重启]-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, 详细错误码英文描述, desc:详细错误码的英文描述-->OK
  </subStatusCode>
</ResponseStatus>
```

69.1.2 获取设备信息参数

Request URL

GET /ISAPI/System/deviceInfo

查询参数

无

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<DeviceInfo xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 设备信息, attr:version{opt, string, 协议版本}-->
  <deviceName>
    <!--ro, req, string, 设备名称, range:[1,128]-->test
  </deviceName>
  <deviceID>
    <!--ro, req, string, 设备编号, range:[1,128]-->test
  </deviceID>
  <deviceDescription>
    <!--ro, opt, string, 设备描述, range:[1,128]-->test
  </deviceDescription>
  <deviceLocation>
```

```
<!--ro, opt, string, 设备位置, range:[1,128]-->hangzhou
</deviceLocation>
<deviceStatus>
  <!--ro, opt, enum, 设备状态, subType:string, [normal#正常,abnormal#异常]-->normal
</deviceStatus>
<DetailAbnormalStatus>
  <!--ro, opt, object, 详细异常状态, desc:仅当deviceStatus为abnormal时有效-->
  <hardDiskFull>
    <!--ro, opt, bool, 盘满-->true
  </hardDiskFull>
  <hardDiskError>
    <!--ro, opt, bool, 盘错-->true
  </hardDiskError>
  <ethernetBroken>
    <!--ro, opt, bool, 网线断-->true
  </ethernetBroken>
  <ipaddrConflict>
    <!--ro, opt, bool, IP地址冲突-->true
  </ipaddrConflict>
  <illegalAccess>
    <!--ro, opt, bool, 非法访问-->true
  </illegalAccess>
  <recordError>
    <!--ro, opt, bool, 录像异常-->true
  </recordError>
  <raidLogicDiskError>
    <!--ro, opt, bool, 阵列虚拟磁盘异常-->true
  </raidLogicDiskError>
  <spareWorkDeviceError>
    <!--ro, opt, bool, 热备监控的工作机异常-->true
  </spareWorkDeviceError>
</DetailAbnormalStatus>
<systemContact>
  <!--ro, opt, string, 生产商, range:[1,32]-->STD-CGI
</systemContact>
<model>
  <!--ro, req, string, 设备型号, range:[1,64]-->IDS-9632NX-I8/X
</model>
<serialNumber>
  <!--ro, req, string, 长序列号, range:[1,48]-->IDS-9632NX-I8/X1620181209CCRR76705411WCVU
</serialNumber>
<macAddress>
  <!--ro, req, string, 物理地址, range:[1,64]-->44:47:cc:c8:d9:e4
</macAddress>
<firmwareVersion>
  <!--ro, req, string, 主控版本号, range:[1,64]-->V4.1.40
</firmwareVersion>
<firmwareReleasedDate>
  <!--ro, opt, string, 主控版本日期-->2019-11-01
</firmwareReleasedDate>
<encoderVersion>
  <!--ro, opt, string, 编码版本, range:[1,32]-->V7.3
</encoderVersion>
<encoderReleasedDate>
  <!--ro, opt, string, 编码版本日期-->2019-11-02
</encoderReleasedDate>
<bootVersion>
  <!--ro, opt, string, boot版本, range:[1,16]-->V1.3.4
</bootVersion>
<bootReleasedDate>
  <!--ro, opt, string, boot发布日期-->2019-11-03
</bootReleasedDate>
<panelVersion>
  <!--ro, opt, string, 面板版本, range:[1,32]-->V1.0
</panelVersion>
<hardwareVersion>
  <!--ro, opt, string, 硬件版本, range:[1,16]-->0x0
</hardwareVersion>
<decoderVersion>
  <!--ro, opt, string, 解码器版本, range:[1,32]-->V1.0
</decoderVersion>
<decoderReleasedDate>
  <!--ro, opt, string, 解码器版本日期-->2019-01-01
</decoderReleasedDate>
<softwareVersion>
  <!--ro, opt, string, 软件版本, range:[1,32]-->V1.23
</softwareVersion>
<capacity>
  <!--ro, opt, int, 设备容量, range:[1,10240], desc:单位 MB-->1
</capacity>
<usedCapacity>
  <!--ro, opt, int, 设备已使用容量, range:[1,10240], desc:单位 MB-->1
</usedCapacity>
<deviceType>
  <!--ro, req, enum, 设备类型, subType:string, [Blade#刀锋,BladePS#治安刀锋,CVR#CVR,DM#深思,DMbehavior#行为分析设备,DVR#硬盘录像机,DVS#编码器,DockStation#
采集站,FA#人脸实时比对报警服务器,FD#抓拍检测服务器,FS#人脸静态数据服务器,HAWK#猎鹰服务器,HMCP#模型比对服务器,HybirdNVR#混合型硬盘录像机,IPCamera#网络摄像
机,IPDome#网络球机,IPZoom#一体化摄像机,InfoReleaseSys#信息发布系统,InfoTerminal#信息发布终端,MCU#MCU,NVR#NVR,PHA#一代混合报警主机,PURE#纯分析服务
器,RFID#RFID,SipServer#Sip服务器,Switch#交换机,LightFace#轻量级脸谱,Visitor#访客机,PWA#一代无线报警主机,PHAPro#二代混合报警主机,ACS#门禁设备,VIS#对讲设
备,FacePaymentTerminal#消费机,InteractiveTerminal#智能交互终端,Cabinet#智能柜,VoiceSpeaker#音柱音箱,PowerAmplifier#功放,FireControlMatrix#消防矩
阵,PagingMicrophone#寻呼话筒,ElevatorControl#梯控主机,conferencePlat#会议平板,AIOT#智能物联网主机(Artificial Intelligence & Internet of Things),OPCA#光学理
化分析装置(Optical physical and chemical analyzer),NetworkReceiver#网络接警机,PanicAlarmPanel#一键式紧急报警求助设备,bodyCamera#执法记录仪,videoMatrix#视频
综合平台,EmbeddedCentralControler#物联中控主机,networkMic#网络麦克风,IPA#模拟和音频产品,PersonnelChannel#人员通道,MediaComponentGateway#媒体组件网
关,ImagingRadarPanel#成像雷达主机,TrafficTerminalServer#交通终端服务器,DecodeStitchControl#解码拼控设备,PDA#PDA扫码设备,securityRadar#态势感知雷达]-->Blade
</deviceType>
```

```
</uequipmentType>
<subDeviceType>
  <!--ro, opt, enum, 设备子类型, subType:string, [accessControlTerminal#门禁一体机 (deviceType为ACS时有效), attendanceCheckDevice#考勤机 (deviceType为ACS时有效), multiChannelAccessController#多通道门禁主机 (deviceType为ACS时有效), personAndIdCardDevice#人证设备 (deviceType为ACS时有效), doorStation#门口机 (deviceType为VIS时有效), mainStation#管理机 (deviceType为VIS时有效)]-->accessControlTerminal
</subDeviceType>
<telecontrolID>
  <!--ro, opt, int, 遥控器ID, range:[1,255]-->1
</telecontrolID>
<supportBeep>
  <!--ro, opt, bool, 是否支持蜂鸣器-->true
</supportBeep>
<supportVideoLoss>
  <!--ro, opt, bool, 是否支持视频丢失-->true
</supportVideoLoss>
<firmwareVersionInfo>
  <!--ro, opt, string, 主控版本信息, range:[1,32]-->B-R-H5-0
</firmwareVersionInfo>
<actualFloorNum>
  <!--ro, opt, int, 实际层数, range:[1,128]-->1
</actualFloorNum>
<localZoneNum>
  <!--ro, opt, int, 本地防区数, range:[0,16]-->1
</localZoneNum>
<alarmOutNum>
  <!--ro, opt, int, 报警时输出数, range:[0,16]-->1
</alarmOutNum>
<alarmInNum>
  <!--ro, opt, int, 报警时输入数, range:[0,16]-->1
</alarmInNum>
<distanceResolution>
  <!--ro, opt, float, 距离分辨率, range:[0.000,0.999]-->0.000
</distanceResolution>
<angleResolution>
  <!--ro, opt, float, 角分辨率, range:[0.000,0.999]-->0.000
</angleResolution>
<speedResolution>
  <!--ro, opt, float, 速度分辨率, range:[0.000,0.999]-->0.000
</speedResolution>
<detectDistance>
  <!--ro, opt, float, 检测距离, range:[0.000,0.999]-->0.000
</detectDistance>
<relayNum>
  <!--ro, opt, int, 本地继电器数 (case) 数量, range:[0,16]-->1
</relayNum>
<electroLockNum>
  <!--ro, opt, int, 本地电锁数, range:[0,16]-->1
</electroLockNum>
<sirenNum>
  <!--ro, opt, int, 本地警号数, range:[0,16]-->1
</sirenNum>
<alarmLamp>
  <!--ro, opt, int, 警灯数, range:[0,16]-->1
</alarmLamp>
<RS485Num>
  <!--ro, opt, int, 本地485数, range:[0,16]-->1
</RS485Num>
<radarVersion>
  <!--ro, opt, string, 雷达版本, range:[1,32]-->test
</radarVersion>
<cameraModuleVersion>
  <!--ro, opt, string, 相机模组版本, range:[1,32]-->test
</cameraModuleVersion>
<mainversion>
  <!--ro, opt, int, 主版本号, range:[1,255]-->1
</mainversion>
<subversion>
  <!--ro, opt, int, 子版本号, range:[1,255]-->1
</subversion>
<upgradeversion>
  <!--ro, opt, int, 升级版本号, range:[1,255]-->1
</upgradeversion>
<customizeversion>
  <!--ro, opt, int, 定制版本号, range:[1,255]-->1
</customizeversion>
<companyName>
  <!--ro, opt, string, 生产公司简称, range:[1,32]-->test
</companyName>
<copyright>
  <!--ro, opt, string, 版权信息, range:[1,32]-->test
</copyright>
<systemName>
  <!--ro, opt, enum, 存储系统名称, subType:string, [storageManagement#存储管理系统, distributedStorageManagement#分布式存储管理系统]-->storageManagement
</systemName>
<systemStatus>
  <!--ro, opt, enum, 系统状态, subType:string, [configured#已配置, unConfigured#未配置]-->configured
</systemStatus>
<isLeaderDevice>
  <!--ro, opt, bool, 是否是资源IP对应设备-->true
</isLeaderDevice>
<clusterVersion>
  <!--ro, opt, string, 系统集群版本, range:[1,32], desc:当前isLeaderDevice字段返回true时有效-->test
</clusterVersion>
<manufacturer>
```

```
<!--ro, opt, enum, 厂商信息, subType:string, [hikvision#海康设备,#中性设备,Hostone#皓视通设备], desc:用于OEM设备区分;海康设备必须返回"hikvision", 中性设备返回""，皓视通设备返回Hostone，其他OEM/厂商名称需要找SDK组申请-->hikvision
</manufacturer>
<centralStorageVersion>
  <!--ro, opt, string, 中心存储版本, range:[1,16]-->test
</centralStorageVersion>
<powerOnMode>
  <!--ro, opt, enum, 启动模式, subType:string, [button#按键启动,adapter#接入电源适配器启动]-->button
</powerOnMode>
<customizedInfo>
  <!--ro, opt, string, 定制项目单号, range:[1,32], desc:基线设备节点返回空，定制设备返回定制项目单号-->test
</customizedInfo>
<verificationCode>
  <!--ro, opt, string, 设备验证码-->test
</verificationCode>
<supportUrl>
  <!--ro, opt, string, 服务门户网站-->test
</supportUrl>
<subSerialNumber>
  <!--ro, opt, string, 子序列号-->test
</subSerialNumber>
<languageType opt="chinese,english,spanish,portuguese,italian,french,russian,turkish,greek,czech,brazilianPortuguese">
  <!--ro, opt, enum, 语言类型, subType:string, [chinese#中文,english#英语,spanish#西班牙语,portuguese#葡萄牙语,italian#意大利语,french#法语,russian#俄语,turkish#土耳其语,greek#希腊语,czech#捷克语,brazilianPortuguese#巴西葡萄牙语,slovenian#斯洛文尼亚语,swedish#瑞典语,norwegian#挪威语,romanian#罗马尼亚语,danish#丹麦语,german#德语,polish#波兰语,dutch#荷兰语,hungarian#匈牙利语,slovak#斯洛伐克语,serbian#塞尔维亚语,southAmericanSpanish#南美西班牙语,ukrainian#乌克兰语,croatian#克罗地亚语,irish#爱尔兰语,bulgarian#保加利亚语,hebrew#希伯来语,thai#泰语,indonesian#印尼语,arabic#阿拉伯语,traditionalChinese#繁体中文,lithuanian#立陶宛语,anglicism#英式英语], attr:opt{req, string, 取值范围}-->chinese
</languageType>
<DockStation>
  <!--ro, opt, object, 采集站信息配置-->
  <Platform>
    <!--ro, opt, object, 采集站对接平台信息配置-->
    <type>
      <!--ro, req, enum, 采集站对接平台, subType:string, [none#无,9533#9533,8618#8618,ISAPI#ISAPI]-->none
    </type>
    <ip>
      <!--ro, opt, string, IP地址, range:[1,32]-->test
    </ip>
    <port>
      <!--ro, opt, int, 通信端口, range:[1,65535]-->1
    </port>
    <userName>
      <!--ro, req, string, 用户名, range:[1,32]-->test
    </userName>
  </Platform>
  <centralStorageBackupEnabled>
    <!--ro, opt, bool, 中心存储备份开启使能-->true
  </centralStorageBackupEnabled>
</DockStation>
<webVersion>
  <!--ro, opt, string, Web版本号, range:[1,32]-->test
</webVersion>
<deviceRFProgramVersion>
  <!--ro, opt, string, 设备射频程序版本号, range:[1,32]-->test
</deviceRFProgramVersion>
<securityModuleSerialNo>
  <!--ro, opt, string, 安全模块序列号, range:[1,32]-->test
</securityModuleSerialNo>
<securityModuleVersion>
  <!--ro, opt, string, 安全模块版本号, range:[1,32]-->test
</securityModuleVersion>
<securityChipVersion>
  <!--ro, opt, string, 安全芯片版本号, range:[1,32]-->test
</securityChipVersion>
<securityModuleKeyVersion>
  <!--ro, opt, string, 安全模块密钥版本号, range:[1,32]-->test
</securityModuleKeyVersion>
<UIDLampRecognition>
  <!--ro, opt, object, uid灯识别设备信息-->
  <enabled>
    <!--ro, opt, bool, 使能配置-->true
  </enabled>
</UIDLampRecognition>
<confDeviceIdPrefix>
  <!--ro, opt, bool, 是否会议带设备名称做前缀-->true
</confDeviceIdPrefix>
<OEMCode>
  <!--ro, opt, enum, 厂商OEM代码, subType:int, [1#海康设备,0#中性设备], desc:海康设备返回1，中性设备返回0，其他厂商根据SDK定义返回-->1
</OEMCode>
<simpleAlgorithmVersion>
  <!--ro, opt, string, 单算法版本, desc:智慧消防分析仪设备信息中返回这个字段表明分析仪只支持一种识别算法，出厂时软件已写死并且不支持录像存储、回放、数据库查询、人脸库、实时预览等功能-->test
</simpleAlgorithmVersion>
<bootTime>
  <!--ro, opt, datetime, 系统启动时间, desc:ISO8601 time （TD格式：本地时间与时差）-->1970-01-01T08:00:00+08:00
</bootTime>
<intelligentAnalysisEngineModel>
  <!--ro, opt, string, 引擎型号, range:[1,32]-->test
</intelligentAnalysisEngineModel>
<marketType>
  <!--ro, opt, enum, 市场类型, subType:int, [0#无效（未知类型）,1#经销型,2#行业型]-->0
</marketType>
<algorithmVersion>
```

```
<!--ro, opt, string, HCNW算法版本, desc: 该数值的字典由研究院统一提供,用于算法模型的匹配-->test
</algorithmVersion>
<firmware>
  <!--ro, opt, string, 固件版本, desc: 该数值的字典由研究院统一提供,用于算法模型的匹配-->test
</firmware>
<engineList>
  <!--ro, opt, object, 设备计算能力列表, desc: 以GPU芯片个数来区分,返回engine的编号,例如有3个芯片,编号分别为1、5、11,则返回的值为1、5、11-->
  <engine>
    <!--ro, opt, int, 返回的engine的编号-->1
  </engine>
</engineList>
<platform>
  <!--ro, opt, enum, 平台类型, subType:int, [1#TX1,2#P4,3#3559,4#3519,5#3516]-->1
</platform>
<platformName>
  <!--ro, opt, string, 算法运行平台对外展现名称, desc: "K81, TX1, KT, H5, H7, K82, G3", 产品自定义, 该字段仅用于直接展现TX1/K81-后端NVR, KT-数据中心, H5-前端, H7-前
端, K82-后端NVR, G3-前端-->test
</platformName>
<touchScreenVersionInfo>
  <!--ro, opt, string, 触摸屏组件版本-->test
</touchScreenVersionInfo>
<protocolFileURL>
  <!--ro, opt, string, 协议须知文件URL, range:[1,32]-->test
</protocolFileURL>
<recycleRecordEnabled>
  <!--ro, opt, bool, 是否循环录像使能-->>false
</recycleRecordEnabled>
<decordChannelNums>
  <!--ro, opt, int, 解码通道数量-->0
</decordChannelNums>
<VGAEnums>
  <!--ro, opt, int, VGA口个数-->0
</VGAEnums>
<USBNums>
  <!--ro, opt, int, usb口个数-->0
</USBNums>
<auxoutNums>
  <!--ro, opt, int, 辅助口个数-->0
</auxoutNums>
<expansionBoardVersion>
  <!--ro, opt, string, 扩展板版本信息, range:[1,32]-->test
</expansionBoardVersion>
<initWizzardDisplay>
  <!--ro, opt, bool, 是否显示初始化向导, desc: 仅用于(集成了向导的应用, 如本地GUI, 海康互联APP)显示配置。默认为true, 当设备本地GUI配置完向导之后, 状态改为
false, 海康互联APP每次(重新)添加设备获取一次状态, 判断是否显示向导界面-->true
</initWizzardDisplay>
<beaconID>
  <!--ro, opt, string, (ETC的扩展单元)设备射频程序版本号, range:[0,32], desc: 默认填写当前值-->test
</beaconID>
<isResetDeviceLanguage>
  <!--ro, opt, bool, 是否重置设备语言, desc: 在对主机进行切换语言升级时(仅Admin、Installer有限权), 当置为true, 表示HC和Web按照切换后的语言显示设备信息;
若置为false, 表示HC和Web仍按照切换前的语言显示设备信息。-->>false
</isResetDeviceLanguage>
<dispalyNum>
  <!--ro, opt, int, 设备屏幕个数-->0
</dispalyNum>
<bspVersion>
  <!--ro, opt, string, BSP软件版本-->test
</bspVersion>
<dspVersion>
  <!--ro, opt, string, DSP软件版本-->test
</dspVersion>
<localUIVersion>
  <!--ro, opt, string, 本地UI版本-->test
</localUIVersion>
<OPCASubType>
  <!--ro, opt, enum, 光学理化分析装置(Optical physical and chemical analyzer)子类型, subType:string, [detector#检测器,pump#泵箱,columnOven#柱温
箱,autoSampler#自动进样器], desc:pump泵箱, autoSampler自动进样器, detector检测器, columnOven柱温-->detector
</OPCASubType>
<wiegandOutNum>
  <!--ro, opt, int, 韦根输出口个数, range:[0,1]-->1
</wiegandOutNum>
<ChipVersionInfoList>
  <!--ro, opt, object, 芯片版本信息列表-->
  <ChipVersionInfo>
    <!--ro, opt, object, 芯片版本信息-->
    <ID>
      <!--ro, req, int, 芯片编号, range:[0,1], desc: 不同类型的芯片编号可以重复, 用户可根据芯片类型+芯片编号区分-->1
    </ID>
    <type>
      <!--ro, req, enum, 芯片类型, subType:string, [MCU#MCU芯片,FPGA#FPGA芯片,MSTAR#MSTAR芯片]-->MCU
    </type>
    <firmwareVersion>
      <!--ro, req, string, 芯片固件版本, range:[0,64]-->1.0.0
    </firmwareVersion>
  </ChipVersionInfo>
</ChipVersionInfoList>
<personBagLinkAlgoEngineVersion>
  <!--ro, opt, string, 人包关联模块的引擎版本, range:[0,64], desc: 安检分析仪中人包关联模块的算法引擎版本-->1.0.0
</personBagLinkAlgoEngineVersion>
<BIOSVersion>
  <!--ro, opt, string, BIOS版本, range:[0,16]-->V1.3.4
</BIOSVersion>
<contactInformation>
```

```

    <!--ro, opt, string, 联系方式, range:[0,64]-->test
  </contactInformation>
  <temperatureModuleVersionInfo>
    <!--ro, opt, string, 测温组件版本-->test
  </temperatureModuleVersionInfo>
  <pedestrianWarningModuleVersion>
    <!--ro, opt, object, 行人预警桩设备模块信息, desc: 当为行人预警桩时才返回-->
    <pedestrianWarningMCUVersion>
      <!--ro, opt, string, MCU(行人预警桩单片机控制板)程序版本, range:[0,32]-->V1.3.4
    </pedestrianWarningMCUVersion>
    <pedestrianWarningRadarVersion>
      <!--ro, opt, string, 行人预警雷达程序版本, range:[0,32]-->V1.3.4
    </pedestrianWarningRadarVersion>
    <pedestrianRangingModuleVersion>
      <!--ro, opt, string, 行人测距模组程序版本, range:[0,32]-->V1.3.4
    </pedestrianRangingModuleVersion>
  </pedestrianWarningModuleVersion>
  <encryptionModel>
    <!--ro, opt, string, 设备加密型号值, range:[0,16], desc: 加密型号是设备固化无法修改的, 目前就是16进制的数字-->test
  </encryptionModel>
  <UWBVersion>
    <!--ro, opt, string, UWB(测距基站模块)版本, range:[1,32]-->V1.3.0
  </UWBVersion>
  <audioBoard>
    <!--ro, opt, object, 音频板信息-->
    <audioBoardModel>
      <!--ro, opt, string, 音频板型号, range:[0,32], desc: 目前仅返回C8,R8-->C8
    </audioBoardModel>
    <audioBoardVersion>
      <!--ro, opt, string, 音频板版本, range:[0,32]-->V1.0.0 build 211210
    </audioBoardVersion>
  </audioBoard>
  <materialScanAlgorithmVersion>
    <!--ro, opt, string, 物料扫描算法版本, range:[0,64], desc: 成像雷达主机用于物料aozhou 扫描的算法版本-->test
  </materialScanAlgorithmVersion>
  <regionVersion>
    <!--ro, opt, enum, 区域版本, subType:string, [basic#基线版本, australia#澳洲版本]-->basic
  </regionVersion>
  <productionDate>
    <!--ro, opt, date, 生产日期-->2022-04-02
  </productionDate>
  <wifiModuleMACAddress>
    <!--ro, opt, string, WiFi模块MAC地址, range:[1,48]-->44:47:cc:c8:d9:e4
  </wifiModuleMACAddress>
  <displayInterFaceSize>
    <!--ro, opt, float, 显示界面尺寸, range:[0.0,16.0], unit:in, unitType:长度, desc: 对讲设备的显示界面对角线长度-->0
  </displayInterFaceSize>
  <releaseRegion>
    <!--ro, opt, enum, 发布区域, subType:string, [China#中国, abroad#海外], desc: 当前仅在HEOP中使用, 用于区分设备是国内还是海外-->China
  </releaseRegion>
  <shortSerialNumber>
    <!--ro, req, string, 短序列号, range:[1,9]-->test
  </shortSerialNumber>
</DeviceInfo>

```

69.1.3 获取设备信息参数配置能力

Request URL

GET /ISAPI/System/deviceInfo/capabilities

查询参数

无

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>
<DeviceInfo xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 设备信息, attr:version{opt, string, 协议版本}-->
  <deviceName min="1" max="128">
    <!--ro, req, string, 设备名称, range:[1,128], attr:min{opt, string, 最小值},max{opt, string, 最大值}-->test
  </deviceName>
  <model min="1" max="64">
    <!--ro, opt, string, 设备型号, range:[1,64], attr:min{opt, string, 最小值},max{opt, string, 最大值}-->iDS-9632NX-I8/X
  </model>
  <serialNumber min="1" max="48">
    <!--ro, opt, string, 长序列号, range:[1,48], attr:min{opt, string, 最小值},max{opt, string, 最大值}-->iDS-9632NX-I8/X1620181209CCRR77605411WCVU
  </serialNumber>
  <macAddress min="1" max="64">
    <!--ro, opt, string, 物理地址, range:[1,64], attr:min{opt, string, 最小值},max{opt, string, 最大值}-->44:47:cc:c8:d9:e4
  </macAddress>
  <firmwareVersion min="1" max="64">
    <!--ro, opt, string, 主控版本号, range:[1,64], attr:min{opt, string, 最小值},max{opt, string, 最大值}-->V4.1.40
  </firmwareVersion>
  <firmwareReleaseDate min="1" max="64">
    <!--ro, opt, string, 主控版本日期, range:[1,64], attr:min{opt, string, 最小值},max{opt, string, 最大值}-->build 181129
  </firmwareReleaseDate>
  <deviceType
    opt="IPCamera,IPDome,DVR,HybirdNVR,NVR,DVS,IPZoom,CVR,DockStation,RFID,InfoReleaseSys,MCU,SipServer,PHA,PWA,InfoTerminal,Switch,Visitor,PHAPro,ACS,VIS,FacePaymentTerminal,InteractiveTerminal,Cabinet,VoiceSpeaker,PowerAmplifier,FireControlMatrix,PagingMicrophone,ElevatorControl,conferencePlat,OPCA,NetworkReceiver,PanicAlarmPanel,videoMatrix,EmbeddedCentralController,networkMic,PersonnelChannel,MediaComponentGateway,ImagingRadarPanel,TrafficTerminalServer,securityRadar">
    <!--ro, opt, string, 设备类型, attr:opt{opt, string, 取值范围}, desc:由于一代的无线报警主机（Hub主机）已经没有维护项目了，后续只会有二代的无线报警主机（Ax Pro 系列），所以无线报警主机可以延续使用PWA。BLade#刀锋, BLadePS#治安刀锋, CVR#CVR, DM#深潜, DMbehavior#行为分析设备, DVR#硬盘录像机, DVS#编码器, DockStation#采集站, FA#人脸实时比对报警服务器, FD#抓拍检测服务器, FS#人脸静态数据服务器, HAWK#猎鹰服务器, HMC#模型比对服务器, HybridNVR#混合型硬盘录像机, IPCamera#网络摄像机, IPDome#网络球机, IPZoom#一体化摄像机, InfoReleaseSys#信息发布系统, InfoTerminal#信息发布终端, MCU#MCU, NVR#NVR, PHA#一代混合报警主机, PURE#纯分析服务器, RFID#RFID, SipServer#Sip服务器, Switch#交换机, LightFace#轻量级脸谱, Visitor#访客机, PWA#一代无线报警主机, PHAPro#二代混合报警主机, ACS#门禁设备, VIS#对讲设备, FacePaymentTerminal#消费机, InteractiveTerminal#智能交互终端, Cabinet#智能柜, VoiceSpeaker#音柱音箱, PowerAmplifier#功放, FireControlMatrix#消防矩阵, PagingMicrophone#寻呼话筒, ElevatorControl#梯控主机, conferencePlat#会议平板, AIOT#智能物联网主机(Artificial Intelligence & Internet of Things), OPCA#光学理化分析装置(Optical physical and chemical analyzer), NetworkReceiver#网络接警机, PanicAlarmPanel#一键式紧急报警求助设备, bodyCamera#执法记录仪, videoMatrix#视频综合平台, EmbeddedCentralController#物联中控主机, networkMic#网络麦克风, IPA#模拟和音频产品, MediaComponentGateway#媒体组件网关, ImagingRadarPanel#成像雷达主机, TrafficTerminalServer#交通终端服务器, securityRadar#态势感知雷达, PersonnelChannel#人员通道-->IPCamera
  </deviceType>
  <alarmOutNum min="0" max="16">
    <!--ro, opt, int, 报警时输出数, range:[0,16], attr:min{opt, string, 最小值},max{opt, string, 最大值}-->1
  </alarmOutNum>
  <webVersion>
    <!--ro, opt, string, Web版本号, range:[1,32]-->1.0
  </webVersion>
  <marketType opt="0,1,2">
    <!--ro, opt, enum, 市场类型, subtype:int, [0#无效（未知类型）,1#经销型,2#行业型], attr:opt{req, string, 取值范围}-->0
  </marketType>
</DeviceInfo>
```

69.2 用户管理

69.2.1 配置设备安全问题

Request URL

PUT /ISAPI/Security/questionConfiguration?security=<security>&iv=<iv>

查询参数

参数名称	参数类型	描述
security	string	取值范围：0,1,2； security=0预留出来，表示不加密，设备返回的能力集中不需要返回0； security = 1表示AES128加密； security=2表示AES256加密。报文中涉及加密字段为：报文中子节点。
iv	string	表示AES加密时的初始化向量参数，固定为16字节长度，在security=1或者2时为必须参数;

请求报文

```
<?xml version="1.0" encoding="UTF-8"?>

<SecurityQuestion xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, 安全问题, attr:version{req, string, 协议版本}-->
  <QuestionList>
    <!--req, array, 问题列表, subType:object-->
    <Question>
      <!--req, object, 单个问题-->
      <id>
        <!--req, int, 表示设备支持的安全问题序号, desc:设备保证唯一性-->1
      </id>
      <answer>
        <!--wo, opt, string, 表示安全问题的答案, desc:GET时不应该返回, PUT时可写, 网传加密 先BASE64后AES128CBC的加密方式-->test
      </answer>
      <mark>
        <!--req, bool, 表示安全问题的标记, desc:标记用户设置过的安全问题; 配置时只处理选中位当前设备配置安全问题支持3个 (一个都不能少) -->true
      </mark>
    </Question>
  </QuestionList>
  <password>
    <!--wo, opt, string, 用户登录密码, desc:CBC,用户登录密码, 该字段为只写字段, 并支持敏感信息加密-->test
  </password>
</SecurityQuestion>
```

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 响应消息, attr:version{ro, req, string, 版本}-->
  <requestURL>
    <!--ro, req, string, 请求的URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, 状态码, subType:int, [0#OK,1#OK,2#Device Busy,3#Device Error,4#Invalid Operation,5#Invalid XML Format,6#Invalid XML Content,7#Reboot Required]-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, 状态信息, subType:string, [OK#成功,Device Busy#设备忙,Device Error#设备异常,Invalid Operation#无效的操作,Invalid XML Format#无效的XML格式,Invalid XML Content#无效的XML内容,Reboot#设备重启]-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, 详细错误码英文描述, desc:详细错误码的英文描述-->OK
  </subStatusCode>
</ResponseStatus>
```

69.2.2 获取检测用户名密码匹配

Request URL

GET /ISAPI/Security/userCheck

查询参数

无

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<userCheck xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 用户名密码匹配, attr:version{req, string, 协议版本}-->
  <statusValue>
    <!--ro, req, enum, 校验结果状态, subType:int, [200#200,401#401]-->200
  </statusValue>
  <statusString>
    <!--ro, opt, enum, 校验结果状态字符信息, subType:string, [OK#OK,Unauthorized#Unauthorized]-->OK
  </statusString>
  <isDefaultPassword>
    <!--ro, opt, bool, 是否是默认密码-->true
  </isDefaultPassword>
  <isRiskPassword>
    <!--ro, opt, bool, 是否是风险密码-->true
  </isRiskPassword>
  <isActivated>
    <!--ro, opt, bool, 是否已激活-->true
  </isActivated>
  <residualValidity>
    <!--ro, opt, int, 密码剩余有效天数, desc: 返回负值, 表示密码已经超期使用, 例如"-3"表示密码已经超期使用3天-->1
  </residualValidity>
  <lockStatus>
    <!--ro, opt, enum, 锁定状态, subType:string, [unlock#未锁定,locked#已锁定]-->unlock
  </lockStatus>
  <unlockTime>
    <!--ro, opt, int, 解锁剩余时间, unit:s, unitType:时间-->1
  </unlockTime>
  <retryLoginTime>
    <!--ro, opt, int, 重试次数-->1
  </retryLoginTime>
</userCheck>
```

69.2.3 获取单个用户本地权限

Request URL

GET /ISAPI/Security/UserPermission/<indexID>/localPermission

查询参数

参数名称	参数类型	描述
indexID	string	

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<localPermission xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 本地权限节点, attr:version{req, string, 协议版本}-->
  <backup>
    <!--ro, opt, bool, 是否开启备份(所有通道)权限, desc:true为开启,false为关闭-->true
  </backup>
  <record>
    <!--ro, opt, bool, 是否开启录像(所有通道)权限, desc:true为开启,false为关闭-->true
  </record>
  <playBack>
    <!--ro, opt, bool, 是否开启回放(所有通道)权限, desc:true为开启,false为关闭-->true
  </playBack>
  <preview>
    <!--ro, opt, bool, 是否开启预览(所有通道)权限, desc:true为开启,false为关闭-->true
  </preview>
  <videoChannelPermissionList>
    <!--ro, opt, object, 视频通道权限配置列表-->
    <videoChannelPermission>
      <!--ro, opt, object, 单一视频通道权限配置-->
      <id>
        <!--ro, req, int, 视频通道ID, range:[1,32]-->test
      </id>
      <playBack>
        <!--ro, opt, bool, 是否开启当前通道回放权限, desc:true为开启,false为关闭-->true
      </playBack>
      <preview>
        <!--ro, opt, bool, 是否开启当前通道预览权限, desc:true为开启,false为关闭-->true
      </preview>
      <record>
        <!--ro, opt, bool, 是否开启当前通道录像权限, desc:true为开启,false为关闭-->true
      </record>
      <backup>
        <!--ro, opt, bool, 是否开启当前通道备份权限, desc:true为开启,false为关闭-->true
      </backup>
      <playBackDoubleVerification>
        <!--ro, opt, bool, 是否开启当前通道回放二次认证权限, desc:true为开启,false为关闭-->true
      </playBackDoubleVerification>
      <backupDoubleVerification>
        <!--ro, opt, bool, 是否开启当前通道备份二次认证权限, desc:true为开启,false为关闭-->true
      </backupDoubleVerification>
    </videoChannelPermission>
  </videoChannelPermissionList>
  <ptzControl>
    <!--ro, req, bool, 是否开启云台控制(所有通道)权限, desc:true为开启,false为关闭-->true
  </ptzControl>
  <ptzChannelPermissionList>
    <!--ro, opt, object, 视频通道云台控制权限列表-->
    <ptzChannelPermission>
      <!--ro, req, object, 单一视频通道云台控制权限-->
      <id>
        <!--ro, req, int, 视频通道ID, range:[1,32]-->test
      </id>
      <ptzControl>
        <!--ro, opt, bool, 是否开启当前通道云台控制权限, desc:true为开启,false为关闭-->true
      </ptzControl>
    </ptzChannelPermission>
  </ptzChannelPermissionList>
  <logOrStateCheck>
    <!--ro, opt, bool, 是否开启日志和设备状态查看权限, desc:true为开启,false为关闭-->true
  </logOrStateCheck>
  <parameterConfig>
    <!--ro, opt, bool, 是否开启设备远程参数设置权限, desc:true为开启,false为关闭-->true
  </parameterConfig>
  <restartOrShutdown>
    <!--ro, opt, bool, 是否开启重启或关机权限, desc:true为开启,false为关闭-->true
  </restartOrShutdown>
  <upgrade>
    <!--ro, opt, bool, 是否开启升级权限, desc:true为开启,false为关闭-->true
  </upgrade>
  <manageChannel>
    <!--ro, opt, bool, 是否开启通道管理权限, desc:true为开启,false为关闭-->true
  </manageChannel>
  <AIModelManagement>
    <!--ro, opt, bool, 是否开启AI模型包操作权限, desc:true为开启,false为关闭-->true
  </AIModelManagement>
  <AITaskManagement>
    <!--ro, opt, bool, 是否开启AI任务管理操作权限, desc:true为开启,false为关闭-->true
  </AITaskManagement>
</localPermission>
```

69.2.4 配置单个用户本地权限

Request URL

PUT /ISAPI/Security/UserPermission/<indexID>/localPermission

查询参数

参数名称	参数类型	描述
indexID	string	

请求报文

```
<?xml version="1.0" encoding="UTF-8"?>

<localPermission xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, 本地权限节点, attr:version{req, string, 协议版本}-->
  <backup>
    <!--opt, bool, 是否开启备份(所有通道)权限, desc:true为开启,false为关闭-->true
  </backup>
  <record>
    <!--opt, bool, 是否开启录像(所有通道)权限, desc:true为开启,false为关闭-->true
  </record>
  <playBack>
    <!--opt, bool, 是否开启回放(所有通道)权限, desc:true为开启,false为关闭-->true
  </playBack>
  <preview>
    <!--opt, bool, 是否开启预览(所有通道)权限, desc:true为开启,false为关闭-->true
  </preview>
  <videoChannelPermissionList>
    <!--opt, object, 视频通道权限配置列表-->
    <videoChannelPermission>
      <!--opt, object, 单一视频通道权限配置-->
      <id>
        <!--req, int, 视频通道ID, range:[1,32]-->test
      </id>
      <playBack>
        <!--opt, bool, 是否开启当前通道回放权限, desc:true为开启,false为关闭-->true
      </playBack>
      <preview>
        <!--opt, bool, 是否开启当前通道预览权限, desc:true为开启,false为关闭-->true
      </preview>
      <record>
        <!--opt, bool, 是否开启当前通道录像权限, desc:true为开启,false为关闭-->true
      </record>
      <backup>
        <!--opt, bool, 是否开启当前通道备份权限, desc:true为开启,false为关闭-->true
      </backup>
      <playBackDoubleVerification>
        <!--opt, bool, 是否开启当前通道回放二次认证权限, desc:true为开启,false为关闭-->true
      </playBackDoubleVerification>
      <backupDoubleVerification>
        <!--opt, bool, 是否开启当前通道备份二次认证权限, desc:true为开启,false为关闭-->true
      </backupDoubleVerification>
    </videoChannelPermission>
  </videoChannelPermissionList>
  <ptzControl>
    <!--req, bool, 是否开启云台控制(所有通道)权限, desc:true为开启,false为关闭-->true
  </ptzControl>
  <ptzChannelPermissionList>
    <!--opt, object, 视频通道云台控制权限列表-->
    <ptzChannelPermission>
      <!--req, object, 单一视频通道云台控制权限-->
      <id>
        <!--req, int, 视频通道ID, range:[1,32]-->test
      </id>
      <ptzControl>
        <!--opt, bool, 是否开启当前通道云台控制权限, desc:true为开启,false为关闭-->true
      </ptzControl>
    </ptzChannelPermission>
  </ptzChannelPermissionList>
  <logOrStateCheck>
    <!--opt, bool, 是否开启日志和设备状态查看权限, desc:true为开启,false为关闭-->true
  </logOrStateCheck>
  <parameterConfig>
    <!--opt, bool, 是否开启设备远程参数设置权限, desc:true为开启,false为关闭-->true
  </parameterConfig>
  <restartOrShutdown>
    <!--opt, bool, 是否开启重启或关机权限, desc:true为开启,false为关闭-->true
  </restartOrShutdown>
  <upgrade>
    <!--opt, bool, 是否开启升级权限, desc:true为开启,false为关闭-->true
  </upgrade>
  <manageChannel>
    <!--opt, bool, 是否开启通道管理权限, desc:true为开启,false为关闭-->true
  </manageChannel>
  <AIModelManagement>
    <!--opt, bool, 是否开启AI模型包操作权限, desc:true为开启,false为关闭-->true
  </AIModelManagement>
  <AITaskManagement>
    <!--opt, bool, 是否开启AI任务管理操作权限, desc:true为开启,false为关闭-->true
  </AITaskManagement>
</localPermission>
```

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 响应消息, attr:version{ro, req, string, 版本}-->
  <requestURL>
    <!--ro, req, string, 请求的URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, 状态码, subType:int, [0#OK,1#OK,2#Device Busy,3#Device Error,4#Invalid Operation,5#Invalid XML Format,6#Invalid XML Content,7#Reboot Required]-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, 状态信息, subType:string, [OK#成功,Device Busy#设备忙,Device Error#设备异常,Invalid Operation#无效的操作,Invalid XML Format#无效的XML格式,Invalid XML Content#无效的XML内容,Reboot#设备重启]-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, 详细错误码英文描述, desc:详细错误码的英文描述-->OK
  </subStatusCode>
</ResponseStatus>
```

69.2.5 获取安全邮箱对应的二维码

Request URL

GET /ISAPI/Security/email/qrCode?format=json&security=<security>&iv=<iv>

查询参数

参数名称	参数类型	描述
security	string	取值范围：0,1,2；security=0预留出来，表示不加密，设备返回的能力集中不需要返回0；security = 1表示AES128加密；security=2表示AES256加密。报文中涉及加密字段为：SecurityInformation下的子节点emailAddress，qrCode；节点serverEmailAddress
iv	string	表示AES加密时的初始化向量参数，固定为16字节长度，在security=1或者2时为必须参数；

请求报文

无

响应报文

```
{
  "SecurityEmailQrCode": {
    /*ro, req, object, 安全邮箱二维码信息*/
    "SecurityInformation": [
      /*ro, req, array, 邮安全信息, subType:object*/
      {
        "emailAddress": "test",
        /*ro, req, string, 邮箱地址, desc:敏感信息加密*/
        "qrCode": "test"
        /*ro, req, string, 二维码, desc:敏感信息加密, 最大长度1024字节*/
      }
    ],
    "serverEmailAddress": "test",
    /*ro, opt, string, 公司服务邮箱地址, desc:敏感信息加密*/
    "isDisclaimerDisplay": true,
    /*ro, opt, bool, 是否显示免责声明, desc:标配设备由于返回的公司服务邮箱地址为海康服务器，经法务沟通，用户邮箱发送至海康服务器，触犯欧洲GDPR隐私保护法规。需要在重置前（海康设备）提示免责声明*/
    "isSupportEZVIZAppScan": true,
    /*ro, opt, bool, 是否支持萤石App扫码, desc:最新的设备支持将二维码通过萤石App扫码获取验证码*/
    "EZVIZAppQrCode": "test",
    /*ro, opt, string, 萤石App二维码, range:[0,1024], desc:仅在isSupportEZVIZAppScan为true时返回，敏感信息加密节*/
    "officialAccountQrCode": "test"
    /*ro, opt, string, 海康威视客户服务公众账号二维码*/
  }
}
```

69.2.6 配置单个用户信息

Request URL

PUT /ISAPI/Security/users/<indexID>?security=<security>&iv=<iv>

查询参数

参数名称	参数类型	描述
indexID	string	
security	string	没有security表示数据不加密；security=1,表示报文中的phoneNum、emailAddress、password、duressPassword、keypadPassword、loginPassword字段AES128加密；security=2,表示报文中的phoneNum、emailAddress、password、duressPassword、keypadPassword、loginPassword字段AES256加密；
iv	string	iv表示初始化向量，在security=1或者2时为必须参数;长度限制为32

请求报文

```
<?xml version="1.0" encoding="UTF-8"?>

<User xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, 用户, attr:version{req, string, 协议版本}-->
  <id>
    <!--req, int, 用户参数序号, range:[1,16], desc: 目前支持1~16-->1
  </id>
  <enabled>
    <!--opt, bool, 用户使能标志-->true
  </enabled>
  <userName>
    <!--req, string, 用户名, range:[1,32]-->test
  </userName>
  <phoneNum>
    <!--wo, opt, string, 电话号码, range:[1,11], desc: 需要敏感信息加密-->test
  </phoneNum>
  <emailAddress>
    <!--wo, opt, string, 邮件地址, range:[1,32], desc: 需要敏感信息加密-->test
  </emailAddress>
  <password>
    <!--wo, req, string, 用户登录密码, range:[1,16], desc: 需进行敏感信息加密-->test
  </password>
  <duressPassword>
    <!--wo, opt, string, 挟持密码, range:[1,16], desc: 需要敏感信息加密-->test
  </duressPassword>
  <keypadPassword>
    <!--wo, opt, string, 键盘密码, range:[1,16], desc: 需要敏感信息加密, 当前通过外设键盘对报警主机进行布防撤防时需要输入该密码-->test
  </keypadPassword>
  <singleKeypadEnable>
    <!--opt, bool, 单次键盘密码使能-->true
  </singleKeypadEnable>
  <invalidKeypadPassword>
    <!--opt, bool, 当前键盘密码是否无效, desc: singleKeypadEnable配置后有效-->true
  </invalidKeypadPassword>
  <userOperateType>
    <!--opt, enum, 用户操作类型, subType:int, [1#网络用户, 2#键盘用户, 3#网络用户+键盘用户],
    desc: 1、当用户配置为网络用户（userOperateType为1或不填）时password有效, 当方法为POST时password必填, 否则可选
    2、当用户配置为键盘用户（userOperateType传递为2）时keypadPassword有效, 当方法为POST时keypadPassword必填, 否则可选
    3、当用户配置为即是网络用户又是键盘用户（userOperateType传递为3）时password和keypadPassword都有效, 当方法为POST时password和keypadPassword必填, 否则可选-->1
  </userOperateType>
  <bondIpAddressList>
    <!--opt, object, 绑定用户的socket IP地址列表-->
    <bondIpAddress>
      <!--opt, object, 绑定用户的socket IP地址-->
      <id>
        <!--req, int, 绑定IP的序号, range:[1,16], desc: 从1开始-->1
      </id>
      <ipAddress>
        <!--opt, string, 绑定的IPv4地址, range:[1,32]-->test
      </ipAddress>
      <ipv6Address>
        <!--opt, string, 绑定的IPv6地址, range:[1,32]-->test
      </ipv6Address>
    </bondIpAddress>
  </bondIpAddressList>
  <bondMacAddressList>
    <!--opt, object, 绑定用户的mac地址列表-->
    <bondMacAddress>
      <!--opt, object, 绑定用户的mac地址-->
      <id>
        <!--req, int, 绑定mac地址的序号, range:[1,16], desc: 从1开始-->1
      </id>
      <macAddress>
        <!--opt, string, 绑定的mac地址, range:[1,32]-->test
      </macAddress>
    </bondMacAddress>
  </bondMacAddressList>
  <userLevel>
    <!--opt, enum, 用户级别, subType:string, [Administrator#管理员, Operator#操作员, Viewer#观察员, installer#安装员, manufacturer#制造者, single#单用户, installerAdmin#安装管理员, installEmployee#安装雇员]-->Administrator
  </userLevel>
</attribute>
```

```
<!--opt, object, 用户特殊属性配置-->
<inherent>
  <!--opt, bool, 是否可被删除, desc:true代表着该用户不可删除,false或者不支持该字段代表着该用户可被删除-->true
</inherent>
</attribute>
<loginPassword>
  <!--req, string, 确认密码, range:[1,16], desc:新增用户, 修改用户信息的时候, 都需要确认密码(确认密码就是用户登录密码); 确认密码在HTTP传输必须强制敏感信息加密; 在HTTPS下可以不用单独加密, HCNetsDK内部默认加密; ISUP和萤石透传的时候, 默认链路加密; -->King726014
</loginPassword>
<cardList>
  <!--opt, object, 该用户所拥有的卡列表-->
  <card>
    <!--opt, object, 卡-->
    <id>
      <!--opt, int, 编号, range:[1,16]-->1
    </id>
    <name>
      <!--opt, string, 名称, range:[1,32]-->test
    </name>
  </card>
</cardList>
<remoteCtrlList>
  <!--opt, object, 该用户所拥有的遥控器列表-->
  <remoteCtrl>
    <!--opt, object, 遥控器-->
    <id>
      <!--opt, int, 编号, range:[1,16]-->1
    </id>
    <name>
      <!--opt, string, 名称, range:[1,32]-->test
    </name>
  </remoteCtrl>
</remoteCtrlList>
<passwordPeriodPrompt>
  <!--opt, object, 密码有效期提醒配置, desc:当密码超过所配置的有效期时, 在登录时会给出密钥过期的提示, 但是不修改密码也不会影响密码的使用-->
  <passwordPeriodType>
    <!--req, enum, 密码有效期提醒类型, subType:string, [permanent#永久生效,byDate#按天数生效], desc:永久生效则该密码永远不会提示密码过期;按天数生效则到密码有效时间时会给出密钥过期的提示, 但是不修改密码也不会影响密码的使用-->permanent
  </passwordPeriodType>
  <passwordPeriodbyDate>
    <!--opt, int, 密码按天数生效有效时间, range:[1,1200], unit:天, unitType:时间, dep:and,{$.User.PasswordPeriodPrompt.passwordPeriodType,eq,byDate}-->90
  </passwordPeriodbyDate>
</PasswordPeriodPrompt>
</User>
```

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 响应消息, attr:version{ro, req, string, 版本}-->
  <requestURL>
    <!--ro, req, string, 请求的URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, 状态码, subType:int, [0#OK,1#OK,2#Device Busy,3#Device Error,4#Invalid Operation,5#Invalid XML Format,6#Invalid XML Content,7#Reboot Required]-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, 状态信息, subType:string, [OK#成功,Device Busy#设备忙,Device Error#设备异常,Invalid Operation#无效的操作,Invalid XML Format#无效的XML格式,Invalid XML Content#无效的XML内容,Reboot#设备重启]-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, 详细错误码英文描述, desc:详细错误码的英文描述-->OK
  </subStatusCode>
  <lockStatus>
    <!--ro, opt, enum, 锁定状态, subType:string, [Lock#锁定,unlock#未锁定], desc:LoginPassword校验失败时返回-->lock
  </lockStatus>
  <retryTimes>
    <!--ro, opt, int, 剩余尝试次数, range:[0,6], desc:LoginPassword校验失败时返回-->1
  </retryTimes>
</ResponseStatus>
```

69.2.7 配置单个用户远程权限

Request URL

PUT /ISAPI/Security/UserPermission/<indexID>/remotePermission

查询参数

参数名称	参数类型	描述
indexID	string	

请求报文

```
<?xml version="1.0" encoding="UTF-8"?>

<remotePermission xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, 远程权限节点, attr:version{req, string, 协议版本}-->
  <record>
    <!--opt, bool, 是否开启录像（所有通道）权限, desc:true为开启,false为关闭-->true
  </record>
  <playBack>
    <!--opt, bool, 是否开启回放（所有通道）权限, desc:true为开启,false为关闭-->true
  </playBack>
  <preview>
    <!--opt, bool, 是否开启预览（所有通道）权限, desc:true为开启,false为关闭-->true
  </preview>
  <videoChannelPermissionList>
    <!--opt, object, 视频通道权限配置列表-->
    <videoChannelPermission>
      <!--opt, object, 单一视频通道权限配置-->
      <id>
        <!--req, int, 视频通道ID, range:[1,32]-->test
      </id>
      <preview>
        <!--opt, bool, 是否开启当前通道预览权限, desc:true为开启,false为关闭-->true
      </preview>
      <playBack>
        <!--opt, bool, 是否开启当前通道回放权限, desc:true为开启,false为关闭-->true
      </playBack>
      <record>
        <!--opt, bool, 是否开启当前通道录像权限, desc:true为开启,false为关闭-->true
      </record>
      <playBackDoubleVerification>
        <!--opt, bool, 是否开启当前通道回放二次认证权限, desc:true为开启,false为关闭-->true
      </playBackDoubleVerification>
    </videoChannelPermission>
  </videoChannelPermissionList>
  <ptzControl>
    <!--opt, bool, 是否开启云台控制（所有通道）权限, desc:true为开启,false为关闭-->true
  </ptzControl>
  <ptzChannelPermissionList>
    <!--opt, object, 视频通道云台控制权限列表-->
    <ptzChannelPermission>
      <!--opt, object, 单一视频通道云台控制权限-->
      <id>
        <!--req, int, 视频通道ID, range:[1,32], desc:/ISAPI/PTZCtrl/channels/ID-->test
      </id>
      <ptzControl>
        <!--opt, bool, 是否开启当前通道云台控制权限, desc:true为开启,false为关闭-->true
      </ptzControl>
    </ptzChannelPermission>
  </ptzChannelPermissionList>
  <logOrStateCheck>
    <!--opt, bool, 是否开启日志和设备状态查看权限-->true
  </logOrStateCheck>
  <parameterConfig>
    <!--opt, bool, 是否开启设备远程参数设置权限-->true
  </parameterConfig>
  <restartOrShutdown>
    <!--opt, bool, 是否开启重启或关机权限-->true
  </restartOrShutdown>
  <upgrade>
    <!--opt, bool, 是否开启升级权限-->true
  </upgrade>
  <voiceTalk>
    <!--opt, bool, 是否开启语音对讲（所有通道）权限-->true
  </voiceTalk>
  <transParentChannel>
    <!--opt, bool, 是否开启透明通道（所有通道）权限, desc:也即控制串口-->true
  </transParentChannel>
  <controlLocalOut>
    <!--opt, bool, 是否开启控制本地输出权限-->true
  </controlLocalOut>
  <manageChannel>
    <!--opt, bool, 是否开启通道管理权限-->true
  </manageChannel>
  <alarmOutOrUpload>
    <!--opt, bool, 是否开启报警上报和输出权限-->true
  </alarmOutOrUpload>
  <IRAIID>
    <!--opt, object, Ir Raid快速权限配置-->
    <rapidConfiguration>
      <!--opt, bool, 是否开启Ir Raid快速配置权限-->true
    </rapidConfiguration>
    <reset>
      <!--opt, bool, 是否开启Ir Raid重设配置权限-->true
    </reset>
  </IRAIID>
  <LEDConfig>
    <!--opt, bool, 是否允许用户修改LED配置-->true
  </LEDConfig>
  <AIModelManagement>
    <!--opt, bool, 是否开启AI模型包操作权限-->true
  </AIModelManagement>
</remotePermission>
```

```
</AIModelManagement>
<AITaskManagement>
  <!--opt, bool, 是否开启AI任务管理操作权限-->true
</AITaskManagement>
<materialReview>
  <!--opt, bool, 素材审核权限, desc:信息发布-->true
</materialReview>
<programReview>
  <!--opt, bool, 节目审核权限, desc:信息发布-->true
</programReview>
<uploadMaterial>
  <!--opt, bool, 上传素材, desc:信息发布-->true
</uploadMaterial>
<addProgram>
  <!--opt, bool, 节目编辑权限, desc:增, 删, 改-->true
</addProgram>
<addSchedule>
  <!--opt, bool, 日程编辑权限, desc:增, 删, 改-->true
</addSchedule>
<programPlay>
  <!--opt, bool, 信息播放权限-->true
</programPlay>
<terminalMgr>
  <!--opt, bool, 终端管理权限, desc:对终端的背光, Logo灯等参数设置-->true
</terminalMgr>
<shopSmart>
  <!--opt, bool, 商业智能权限-->true
</shopSmart>
<materialReplace>
  <!--opt, bool, 素材替换权限-->true
</materialReplace>
<scheduleRelease>
  <!--opt, bool, 日程发布权限-->true
</scheduleRelease>
<scheduleReview>
  <!--opt, bool, 日程审核-->true
</scheduleReview>
<addTerminal>
  <!--opt, bool, 终端添加-->true
</addTerminal>
<terminalDispatch>
  <!--opt, bool, 终端分配权限, desc:将该组织下的终端分配给子组织-->true
</terminalDispatch>
<addUser>
  <!--opt, bool, 用户编辑权限, desc:增, 删, 改-->true
</addUser>
<addOrg>
  <!--opt, bool, 组织编辑权限, desc:增, 删, 改-->true
</addOrg>
<logMgr>
  <!--opt, bool, 日志管理-->true
</logMgr>
<dataUpdate>
  <!--opt, bool, 数据更新, desc:叫号数据-->true
</dataUpdate>
<insertMessage>
  <!--opt, bool, 插播消息权限-->true
</insertMessage>
<notReview>
  <!--opt, bool, 无需审核权限-->true
</notReview>
<MgrTerminal>
  <!--opt, object, 管理的终端或终端组-->
  <TerminalNoList>
    <!--opt, object, 终端号列表-->
    <terminalNo>
      <!--opt, int, 终端号, range:[1,10]-->1
    </terminalNo>
  </TerminalNoList>
  <TerminalGroupNoList>
    <!--opt, object, 终端组列表-->
    <terminalGroupNo>
      <!--opt, int, 终端组号, range:[1,10]-->1
    </terminalGroupNo>
  </TerminalGroupNoList>
</MgrTerminal>
<subSysOrZoneArm>
  <!--opt, bool, 子系统/防区布防权限-->true
</subSysOrZoneArm>
<subSysOrZoneDisarm>
  <!--opt, bool, 子系统/防区撤防权限-->true
</subSysOrZoneDisarm>
<subSysOrZoneClearArm>
  <!--opt, bool, 子系统/防区消警权限-->true
</subSysOrZoneClearArm>
<zoneBypass>
  <!--opt, bool, 防区旁路权限-->true
</zoneBypass>
<zoneBypassRecover>
  <!--opt, bool, 防区旁路恢复权限-->true
</zoneBypassRecover>
<subSystemList>
  <!--opt, object, 子系统列表, desc:当用户添加或者配置operator时可关联子系统-->
```

```
<subSystem>
  <!--opt, int, 子系统, range:[1,16]-->1
</subSystem>
</subSystemList>
<factoryReset>
  <!--opt, bool, 恢复默认参数-->true
</factoryReset>
<arm>
  <!--opt, bool, 布防-->true
</arm>
<disarm>
  <!--opt, bool, 撤防-->true
</disarm>
<LPListAudit>
  <!--opt, bool, 黑白名单操作权限,
desc:有这个权限的用户可以导入/导出/编辑/删除/查询黑白名单
对应接口如下:
车牌黑白名单查询:/ISAPI/Traffic/channels/<ID>/searchLPListAudit
导入/导出车牌黑白名单数据:/ISAPI/Traffic/channels/<ID>/LicensePlateAuditData
车牌名单添加修改:/ISAPI/Traffic/channels/<channelID>/LicensePlateAuditData/record?format=json
车牌黑白名单数据删除: /ISAPI/Traffic/channels/<channelID>/LicensePlateAuditData/delete?format=json-->true
</LPListAudit>
<applicationWarehouse>
  <!--opt, bool, 应用仓库管理权限, desc:具备管理权限, 用户可进行导入、删除、启动HEOP APP等操作-->true
</applicationWarehouse>
<falseAlarmMaterialLibMgr>
  <!--opt, bool, 误报分析素材库管理权限-->true
</falseAlarmMaterialLibMgr>
</remotePermission>
```

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 响应消息, attr:version{ro, req, string, 版本}-->
  <requestURL>
    <!--ro, req, string, 请求的URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, 状态码, subtype:int, [0#OK,1#OK,2#Device Busy,3#Device Error,4#Invalid Operation,5#Invalid XML Format,6#Invalid XML Content,7#Reboot Required]-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, 状态信息, subtype:string, [OK#成功,Device Busy#设备忙,Device Error#设备异常,Invalid Operation#无效的操作,Invalid XML Format#无效的XML格式,Invalid XML Content#无效的XML内容,Reboot#设备重启]-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, 详细错误码英文描述, desc:详细错误码的英文描述-->OK
  </subStatusCode>
</ResponseStatus>
```

69.2.8 获取单个用户远程权限

Request URL

GET /ISAPI/Security/UserPermission/<indexID>/remotePermission

查询参数

参数名称	参数类型	描述
indexID	string	

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<remotePermission xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 远程权限节点, attr:version{req, string, 协议版本}-->
  <record>
    <!--ro, opt, bool, 是否开启录像（所有通道）权限, desc:true为开启,false为关闭-->true
  </record>
  <playBack>
    <!--ro, opt, bool, 是否开启回放（所有通道）权限, desc:true为开启,false为关闭-->true
  </playBack>
  <preview>
    <!--ro, opt, bool, 是否开启预览（所有通道）权限, desc:true为开启,false为关闭-->true
  </preview>
  <videoChannelPermissionList>
```

```
<!--ro, opt, object, 视频通道权限配置列表-->
<videoChannelPermission>
  <!--ro, opt, object, 单一视频通道权限配置-->
  <id>
    <!--ro, req, int, 对应视频通道ID, range:[1,32]-->test
  </id>
  <preview>
    <!--ro, opt, bool, 是否开启当前通道预览权限, desc:true为开启,false为关闭-->true
  </preview>
  <playBack>
    <!--ro, opt, bool, 是否开启当前通道回放权限, desc:true为开启,false为关闭-->true
  </playBack>
  <record>
    <!--ro, opt, bool, 是否开启当前通道录像权限, desc:true为开启,false为关闭-->true
  </record>
  <playBackDoubleVerification>
    <!--ro, opt, bool, 是否开启当前通道回放二次认证权限, desc:true为开启,false为关闭-->true
  </playBackDoubleVerification>
</videoChannelPermission>
</videoChannelPermissionList>
<ptzControl>
  <!--ro, opt, bool, 是否开启云台控制（所有通道）权限, desc:true为开启,false为关闭-->true
</ptzControl>
<ptzChannelPermissionList>
  <!--ro, opt, object, 视频通道云台控制权限列表-->
  <ptzChannelPermission>
    <!--ro, opt, object, 单一视频通道云台控制权限-->
    <id>
      <!--ro, req, int, 视频通道ID, range:[1,32], desc:/ISAPI/PTZCtrl/channels/ID-->test
    </id>
    <ptzControl>
      <!--ro, opt, bool, 是否开启当前通道云台控制权限, desc:true为开启,false为关闭-->true
    </ptzControl>
  </ptzChannelPermission>
</ptzChannelPermissionList>
<logOrStateCheck>
  <!--ro, opt, bool, 是否开启日志和设备状态查看权限-->true
</logOrStateCheck>
<parameterConfig>
  <!--ro, opt, bool, 是否开启设备远程参数设置权限-->true
</parameterConfig>
<restartOrShutdown>
  <!--ro, opt, bool, 是否开启重启或关机权限-->true
</restartOrShutdown>
<upgrade>
  <!--ro, opt, bool, 是否开启升级权限-->true
</upgrade>
<voiceTalk>
  <!--ro, opt, bool, 是否开启语音对讲（所有通道）权限-->true
</voiceTalk>
<transparentChannel>
  <!--ro, opt, bool, 是否开启透明通道（所有通道）权限, desc:也即控制串口-->true
</transparentChannel>
<controlLocalOut>
  <!--ro, opt, bool, 是否开启控制本地输出权限-->true
</controlLocalOut>
<manageChannel>
  <!--ro, opt, bool, 是否开启通道管理权限-->true
</manageChannel>
<alarmOutOrUpload>
  <!--ro, opt, bool, 是否开启报警上报和输出权限-->true
</alarmOutOrUpload>
<IRAID>
  <!--ro, opt, object, Iraid快速权限配置-->
  <rapidConfiguration>
    <!--ro, opt, bool, 是否开启Iraid快速配置权限-->true
  </rapidConfiguration>
  <reset>
    <!--ro, opt, bool, 是否开启Iraid重设配置权限-->true
  </reset>
</IRAID>
<LEDConfig>
  <!--ro, opt, bool, 是否允许用户修改LED配置-->true
</LEDConfig>
<AIModelManagement>
  <!--ro, opt, bool, 是否开启AI模型包操作权限-->true
</AIModelManagement>
<AITaskManagement>
  <!--ro, opt, bool, 是否开启AI任务管理操作权限-->true
</AITaskManagement>
<materialReview>
  <!--ro, opt, bool, 素材审核权限, desc:信息发布-->true
</materialReview>
<programReview>
  <!--ro, opt, bool, 节目审核权限, desc:信息发布-->true
</programReview>
<uploadMaterial>
  <!--ro, opt, bool, 上传素材, desc:信息发布-->true
</uploadMaterial>
<addProgram>
  <!--ro, opt, bool, 节目编辑权限, desc:增, 删, 改-->true
</addProgram>
<addSchedule>
```

```
<!--ro, opt, bool, 口控编程权限, desc:增, 删, 改-->true
</addSchedule>
<programPlay>
  <!--ro, opt, bool, 信息播放权限-->true
</programPlay>
<terminalMgr>
  <!--ro, opt, bool, 终端管理权限, desc:对终端的背光, Logo灯等参数设置-->true
</terminalMgr>
<shopSmart>
  <!--ro, opt, bool, 商业智能权限-->true
</shopSmart>
<materialReplace>
  <!--ro, opt, bool, 素材替换权限-->true
</materialReplace>
<scheduleRelease>
  <!--ro, opt, bool, 日程发布权限-->true
</scheduleRelease>
<scheduleReview>
  <!--ro, opt, bool, 日程审核-->true
</scheduleReview>
<addTerminal>
  <!--ro, opt, bool, 终端添加-->true
</addTerminal>
<terminalDispatch>
  <!--ro, opt, bool, 终端分配权限, desc:将该组织下的终端分配给子组织-->true
</terminalDispatch>
<addUser>
  <!--ro, opt, bool, 用户编辑权限, desc:增, 删, 改-->true
</addUser>
<addOrg>
  <!--ro, opt, bool, 组织编辑权限, desc:增, 删, 改-->true
</addOrg>
<logMgr>
  <!--ro, opt, bool, 日志管理-->true
</logMgr>
<dataUpdate>
  <!--ro, opt, bool, 数据更新, desc:叫号数据-->true
</dataUpdate>
<insertMessage>
  <!--ro, opt, bool, 插播消息权限-->true
</insertMessage>
<notReview>
  <!--ro, opt, bool, 无需审核权限-->true
</notReview>
<MgrTerminal>
  <!--ro, opt, object, 管理的终端或终端组-->
  <TerminalNoList>
    <!--ro, opt, object, 终端号列表-->
    <terminalNo>
      <!--ro, opt, int, 终端号, range:[1,10]-->1
    </terminalNo>
  </TerminalNoList>
  <TerminalGroupNoList>
    <!--ro, opt, object, 终端组列表-->
    <terminalGroupNo>
      <!--ro, opt, int, 终端组号, range:[1,10]-->1
    </terminalGroupNo>
  </TerminalGroupNoList>
</MgrTerminal>
<subSysOrZoneArm>
  <!--ro, opt, bool, 子系统/防区布防权限-->true
</subSysOrZoneArm>
<subSysOrZoneDisarm>
  <!--ro, opt, bool, 子系统/防区撤防权限-->true
</subSysOrZoneDisarm>
<subSysOrZoneClearArm>
  <!--ro, opt, bool, 子系统/防区消警权限-->true
</subSysOrZoneClearArm>
<zoneBypass>
  <!--ro, opt, bool, 防区旁路权限-->true
</zoneBypass>
<zoneBypassRecover>
  <!--ro, opt, bool, 防区旁路恢复权限-->true
</zoneBypassRecover>
<subSystemList>
  <!--ro, opt, object, 子系统列表, desc:当用户添加或者配置operator时可关联子系统-->
  <subSystem>
    <!--ro, opt, int, 子系统, range:[1,16]-->1
  </subSystem>
</subSystemList>
<factoryReset>
  <!--ro, opt, bool, 恢复默认参数-->true
</factoryReset>
<arm>
  <!--ro, opt, bool, 布防-->true
</arm>
<disarm>
  <!--ro, opt, bool, 撤防-->true
</disarm>
<accessControl>
  <!--ro, opt, bool, 门禁控制-->true
</accessControl>
<LPListAudit>
  <!--ro, opt, bool, 黑白名单操作权限,
```

desc: 有这个权限的用户可以导入/导出/编辑/删除/查询黑白名单
对应接口如下:
车牌黑白名单查询: /ISAPI/Traffic/channels/<ID>/searchLPListAudit
导入/导出车牌黑白名单数据: /ISAPI/Traffic/channels/<ID>/LicensePlateAuditData
车牌名单添加修改: /ISAPI/Traffic/channels/<channelID>/LicensePlateAuditData/record?format=json
车牌黑白名单数据删除: /ISAPI/Traffic/channels/<channelID>/LicensePlateAuditData/delete?format=json-->true

```
</LPListAudit>
<applicationWarehouse>
  <!--ro, opt, bool, 应用仓库管理权限, desc: 不支持不返回。具备管理权限, 用户可进行导入、删除、启动HEOP APP等操作-->true
</applicationWarehouse>
<falseAlarmMaterialLibMgr>
  <!--ro, opt, bool, 误报分析素材库管理权限-->true
</falseAlarmMaterialLibMgr>
</remotePermission>
```

69.2.9 获取单个用户远程权限能力

Request URL

GET /ISAPI/Security/UserPermission/<indexID>/remotePermission/capabilities

查询参数

参数名称	参数类型	描述
indexID	string	

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<RemotePermission xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 用户权限能力, attr:version{req, string, 协议版本}-->
  <record>
    <!--ro, opt, bool, 是否开启录像（所有通道）权限, desc:true为开启,false为关闭-->true
  </record>
  <playBack>
    <!--ro, opt, bool, 是否开启回放（所有通道）权限, desc:true为开启,false为关闭-->true
  </playBack>
  <preview>
    <!--ro, opt, bool, 是否开启预览（所有通道）权限, desc:true为开启,false为关闭-->true
  </preview>
  <videoChannelPermissionList>
    <!--ro, opt, object, 视频通道权限配置列表-->
    <videoChannelPermission>
      <!--ro, opt, object, 单一视频通道权限配置-->
      <id>
        <!--ro, req, int, 视频通道ID, range:[1,32]-->test
      </id>
      <preview>
        <!--ro, opt, bool, 是否开启当前通道预览权限, desc:true为开启,false为关闭-->true
      </preview>
      <playBack>
        <!--ro, opt, bool, 是否开启当前通道回放权限, desc:true为开启,false为关闭-->true
      </playBack>
      <record>
        <!--ro, opt, bool, 是否开启当前通道录像权限, desc:true为开启,false为关闭-->true
      </record>
      <playBackDoubleVerification>
        <!--ro, opt, bool, 是否开启当前通道回放二次认证权限, desc:true为开启,false为关闭-->true
      </playBackDoubleVerification>
    </videoChannelPermission>
  </videoChannelPermissionList>
  <ptzControl>
    <!--ro, opt, bool, 是否开启云台控制（所有通道）权限, desc:true为开启,false为关闭-->true
  </ptzControl>
  <ptzChannelPermissionList>
    <!--ro, opt, object, 视频通道云台控制权限列表-->
    <ptzChannelPermission>
      <!--ro, opt, object, 单一视频通道云台控制权限-->
      <id>
        <!--ro, req, int, 视频通道ID, range:[1,32], desc:/ISAPI/PTZCtrl/channels/ID-->test
      </id>
      <ptzControl>
        <!--ro, opt, bool, 是否开启当前通道云台控制权限, desc:true为开启,false为关闭-->true
      </ptzControl>
    </ptzChannelPermission>
  </ptzChannelPermissionList>
  <logOrStateCheck>
    <!--ro, opt, bool, 是否开启日志和设备状态查看权限-->true
  </logOrStateCheck>
  <parameterConfig>
```

```
<!--ro, opt, bool, 是否开启设备远程参数读写权限-->true
</parameterConfig>
<restartOrShutdown>
  <!--ro, opt, bool, 是否开启重启或关机权限-->true
</restartOrShutdown>
<upgrade>
  <!--ro, opt, bool, 是否开启升级权限-->true
</upgrade>
<voiceTalk>
  <!--ro, opt, bool, 是否开启语音对讲（所有通道）权限-->true
</voiceTalk>
<transParentChannel>
  <!--ro, opt, bool, 是否开启透明通道（所有通道）权限, desc: 也即控制串口-->true
</transParentChannel>
<contorlLocalOut>
  <!--ro, opt, bool, 是否开启控制本地输出权限-->true
</contorlLocalOut>
<manageChannel>
  <!--ro, opt, bool, 是否开启通道管理权限-->true
</manageChannel>
<alarmOutOrUpload>
  <!--ro, opt, bool, 是否开启报警上报和输出权限-->true
</alarmOutOrUpload>
<IRAID>
  <!--ro, opt, object, Iraid快速权限配置-->
  <rapidConfiguration>
    <!--ro, opt, bool, 是否开启Iraid快速配置权限-->true
  </rapidConfiguration>
  <reset>
    <!--ro, opt, bool, 是否开启Iraid重置配置权限-->true
  </reset>
</IRAID>
<LEDConfig>
  <!--ro, opt, bool, 是否允许用户修改LED配置-->true
</LEDConfig>
<AIModelManagement>
  <!--ro, opt, bool, 是否开启AI模型包操作权限-->true
</AIModelManagement>
<AITaskManagement>
  <!--ro, opt, bool, 是否开启AI任务管理操作权限-->true
</AITaskManagement>
<materialReview>
  <!--ro, opt, bool, 素材审核权限, desc: 信息发布-->true
</materialReview>
<programReview>
  <!--ro, opt, bool, 节目审核权限, desc: 信息发布-->true
</programReview>
<uploadMaterial>
  <!--ro, opt, bool, 上传素材, desc: 信息发布-->true
</uploadMaterial>
<addProgram>
  <!--ro, opt, bool, 节目编辑权限, desc: 增, 删, 改-->true
</addProgram>
<addSchedule>
  <!--ro, opt, bool, 日程编辑权限, desc: 增, 删, 改-->true
</addSchedule>
<programPlay>
  <!--ro, opt, bool, 信息播放权限-->true
</programPlay>
<terminalMgr>
  <!--ro, opt, bool, 终端管理权限, desc: 对终端的背光, logo灯等参数设置-->true
</terminalMgr>
<shopSmart>
  <!--ro, opt, bool, 商业智能权限-->true
</shopSmart>
<materialReplace>
  <!--ro, opt, bool, 素材替换权限-->true
</materialReplace>
<scheduleRelease>
  <!--ro, opt, bool, 日程发布权限-->true
</scheduleRelease>
<scheduleReview>
  <!--ro, opt, bool, 日程审核-->true
</scheduleReview>
<addTerminal>
  <!--ro, opt, bool, 终端添加-->true
</addTerminal>
<terminalDispatch>
  <!--ro, opt, bool, 终端分配权限, desc: 将该组织下的终端分配给子组织-->true
</terminalDispatch>
<addUser>
  <!--ro, opt, bool, 用户编辑权限, desc: 增, 删, 改-->true
</addUser>
<addOrg>
  <!--ro, opt, bool, 组织编辑权限, desc: 增, 删, 改-->true
</addOrg>
<logMgr>
  <!--ro, opt, bool, 日志管理-->true
</logMgr>
<dataUpdate>
  <!--ro, opt, bool, 数据更新, desc: 叫号数据-->true
</dataUpdate>
<insertMessage>
  <!--ro, opt, bool, 插播消息权限-->true
```

```

</insertMessage>
<notReview>
  <!--ro, opt, bool, 无需审核权限-->true
</notReview>
<MgrTerminal>
  <!--ro, opt, object, 管理的终端或终端组-->
  <TerminalNoList size="10">
    <!--ro, opt, object, 终端号列表, attr:size{req, int, 数组元素总数}-->
    <terminalNo>
      <!--ro, opt, int, 终端号, range:[1,10]-->1
    </terminalNo>
  </TerminalNoList>
  <TerminalGroupNoList size="10">
    <!--ro, opt, object, 终端组列表, attr:size{req, int, 数组元素总数}-->
    <terminalGroupNo>
      <!--ro, opt, int, 终端组号, range:[1,10]-->1
    </terminalGroupNo>
  </TerminalGroupNoList>
</MgrTerminal>
<LPListAudit>
  <!--ro, opt, bool, 黑白名单操作权限,
desc:有这个权限的用户可以导入/导出/编辑/删除/查询黑白名单
对应接口如下:
车牌黑白名单查询:/ISAPI/Traffic/channels/<ID>/searchLPListAudit
导入/导出车牌黑白名单数据:/ISAPI/Traffic/channels/<ID>/LicensePlateAuditData
车牌名单添加修改:/ISAPI/Traffic/channels/<channelID>/LicensePlateAuditData/record?format=json
车牌黑白名单数据删除:/ISAPI/Traffic/channels/<channelID>/LicensePlateAuditData/delete?format=json-->true
  </LPListAudit>
  <applicationWarehouse opt="true,false">
    <!--ro, opt, bool, 应用仓库管理权限, attr:opt{req, string, 取值范围}, desc:不支持不返回。具备管理权限, 用户可进行导入、删除、启动HEOP APP等操作-->true
  </applicationWarehouse>
  <falseAlarmMaterialLibMgr opt="true,false">
    <!--ro, opt, bool, 误报分析素材库管理权限, attr:opt{req, string, 取值范围}-->true
  </falseAlarmMaterialLibMgr>
</RemotePermission>

```

69.2.10 获取单个用户信息

Request URL

GET /ISAPI/Security/users/<indexID>?security=<security>&iv=<iv>

查询参数

参数名称	参数类型	描述
indexID	string	
security	string	没有security表示数据不加密; security=1,表示报文中的phoneNum、emailAddress、password、duressPassword、keypadPassword、loginPassword字段AES128加密; security=2,表示报文中的phoneNum、emailAddress、password、duressPassword、keypadPassword、loginPassword字段AES256加密;
iv	string	iv表示初始化向量, 在security=1或者2时为必须参数;长度限制为32

请求报文

无

响应报文

```

<?xml version="1.0" encoding="UTF-8"?>
<User xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 用户, attr:version{req, string, 协议版本}-->
  <id>
    <!--ro, req, int, 用户参数序号-->1
  </id>
  <enabled>
    <!--ro, opt, bool, 用户使能标志-->true
  </enabled>
  <userName>
    <!--ro, req, string, 用户名, range:[1,32]-->test
  </userName>
  <phoneNum>
    <!--ro, opt, string, 电话号码, range:[1,11], desc:需要敏感信息加密-->test
  </phoneNum>
  <emailAddress>
    <!--ro, opt, string, 邮件地址, range:[1,32], desc:需要敏感信息加密-->test
  </emailAddress>
  <duressPassword>

```

```
<!--ro, opt, string, 扶持密码, range:[1,16], desc:需要敏感信息加密-->test
</duressPassword>
<keypadPassword>
  <!--ro, opt, string, 键盘密码, range:[1,16], desc:需要敏感信息加密,当前通过外设键盘对报警主机进行布防撤防时需要输入该密码-->test
</keypadPassword>
<singleKeypadEnable>
  <!--ro, opt, bool, 单次键盘密码使能-->true
</singleKeypadEnable>
<invalidKeypadPassword>
  <!--ro, opt, bool, 当前键盘密码是否无效, desc:singleKeypadEnable配置后有效-->true
</invalidKeypadPassword>
<userOperateType>
  <!--ro, opt, enum, 用户操作类型, subType:int, [1#网络用户,2#键盘用户,3#网络用户+键盘用户],
desc:1、当用户配置为网络用户（userOperateType为1或不填）时password有效,当方法为POST时password必填, 否则可选
2、当用户配置为键盘用户（userOperateType传递为2）时keypadPassword有效,当方法为POST时keypadPassword必填, 否则可选
3、当用户配置为即是网络用户又是键盘用户（userOperateType传递为3）时password和keypadPassword都有效,当方法为POST时password和keypadPassword必填, 否则可选-->1
</userOperateType>
<bondIpAddressList>
  <!--ro, opt, object, 绑定用户的socket IP地址列表-->
  <bondIpAddress>
    <!--ro, opt, object, 绑定用户的socket IP地址-->
    <id>
      <!--ro, req, int, 绑定IP的序号, range:[1,16], desc:从1开始-->1
    </id>
    <ipAddress>
      <!--ro, opt, string, 绑定的IPv4地址, range:[1,32]-->test
    </ipAddress>
    <ipv6Address>
      <!--ro, opt, string, 绑定的IPv6地址, range:[1,128]-->test
    </ipv6Address>
  </bondIpAddress>
</bondIpAddressList>
<bondMacAddressList>
  <!--ro, opt, object, 绑定用户的mac地址列表-->
  <bondMacAddress>
    <!--ro, opt, object, 绑定用户的mac地址-->
    <id>
      <!--ro, req, int, 绑定mac地址的序号, range:[1,16], desc:从1开始-->1
    </id>
    <macAddress>
      <!--ro, opt, string, 绑定的mac地址, range:[1,32]-->test
    </macAddress>
  </bondMacAddress>
</bondMacAddressList>
<userLevel>
  <!--ro, opt, enum, 用户级别, subType:string, [Administrator#管理员,Operator#操作员,Viewer#观察员,installer#安装员,manufacturer#制造者,single#单用
户,installerAdmin#安装管理员,installEmployee#安装雇员,Employee#普通员工（考勤）,DepartmentManager#部门主管（考勤）]-->Administrator
</userLevel>
<attribute>
  <!--ro, opt, object, 用户特殊属性配置-->
  <inherent>
    <!--ro, opt, bool, 是否可被删除, desc:true代表着该用户不可删除,false或者不支持该字段代表着该用户可被删除-->true
  </inherent>
</attribute>
<cardList>
  <!--ro, opt, object, 该用户所拥有的卡列表-->
  <card>
    <!--ro, opt, object, 卡-->
    <id>
      <!--ro, opt, int, 编号, range:[1,16]-->1
    </id>
    <name>
      <!--ro, opt, string, 名称, range:[1,32]-->test
    </name>
  </card>
</cardList>
<remoteCtrlList>
  <!--ro, opt, object, 该用户所拥有的遥控器列表-->
  <remoteCtrl>
    <!--ro, opt, object, 遥控器-->
    <id>
      <!--ro, opt, int, 编号, range:[1,16]-->1
    </id>
    <name>
      <!--ro, opt, string, 名称, range:[1,32]-->test
    </name>
  </remoteCtrl>
</remoteCtrlList>
<userNo>
  <!--ro, opt, int, 用户ID对应的界面展示编号, desc:该编号用于App和Web做界面展示, 以及上报ARC, 针对所有用户类型都有效, 实现方式为500+id-->501
</userNo>
<adminType>
  <!--ro, opt, enum, admin类型, subType:string, [Cloud#云端admin,LAN#局域网admin], desc:用于混合报警主机区分该admin用户是云端创建的, 还是本地创建的, 并在
App和Web做界面展示-->Cloud
</adminType>
<installerType>
  <!--ro, opt, enum, installer类型, subType:string, [Cloud#云端installer,LAN#局域网installer], desc:用于混合报警主机区分该installer用户是云端创建的, 还是
本地创建, 并在App和Web做界面展示-->Cloud
</installerType>
<PasswordPeriodPrompt>
  <!--ro, opt, object, 密码有效期提醒配置, desc:当密码超过所配置的有效期时, 在登录时会给出密钥过期的提示, 但是不修改密码也不会影响密码的使用-->
  <passwordPeriodType>
    <!--ro, req, enum, 密码有效期提醒类型, subType:string, [permanent#永久生效,byDate#按天数生效], desc:永久生效则该密码永远不会提示密码过期;按天数生效则
到密码有效时间时会给出密钥过期的提示. 但是不修改密码也不会影响密码的使用-->permanent
```

```

</passwordPeriodType>
<passwordPeriodbyDate>
  <!--ro, opt, int, 密码按天数生效有效时间, range:[1,1200], unit:天, unitType:时间, dep:and,{$.User.PasswordPeriodPrompt.passwordPeriodType,eq,byDate}-->
</passwordPeriodbyDate>
</PasswordPeriodPrompt>
<userActivationStatus>
  <!--ro, opt, bool, 用户激活状态, desc:不返回默认为true.true#已激活,false#未激活-->true
</userActivationStatus>
<invalidDuressPassword>
  <!--ro, opt, bool, 当前挟持密码是否无效, desc:若未配置挟持密码, 该字段为true-->true
</invalidDuressPassword>
</User>

```

69.2.11 获取单个用户参数配置能力

Request URL

GET /ISAPI/Security/users/<indexID>/capabilities

查询参数

参数名称	参数类型	描述
indexID	string	--

请求报文

无

响应报文

```

<?xml version="1.0" encoding="UTF-8"?>

<User xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 用户, attr:version{req, string, 协议版本}-->
  <id min="1" max="64">
    <!--ro, req, int, 用户参数序号, range:[1,64], attr:min{req, int, 最小值},max{req, int, 最大值}-->1
  </id>
  <enabled opt="true,false">
    <!--ro, opt, bool, 用户使能标志, attr:opt{req, string, 取值范围}-->true
  </enabled>
  <userName min="1" max="32">
    <!--ro, req, string, 用户名, range:[1,32], attr:min{req, int, 最小值},max{req, int, 最大值}-->test
  </userName>
  <phoneNum min="1" max="11">
    <!--ro, opt, string, 电话号码, range:[1,11], attr:min{req, int, 最小值},max{req, int, 最大值}, desc:可替换userName, 用于登录, 需要敏感信息加密-->test
  </phoneNum>
  <emailAddress min="1" max="32">
    <!--ro, opt, string, 邮件地址, range:[1,32], attr:min{req, int, 最小值},max{req, int, 最大值}, desc:可替换userName, 用于登录, 需要敏感信息加密-->test
  </emailAddress>
  <oldPassword min="1" max="16">
    <!--ro, opt, string, 用户登录之前的旧密码, range:[1,16], attr:min{req, int, 最小值},max{req, int, 最大值}, desc:设备不返回密码内容-->null
  </oldPassword>
  <password min="1" max="16">
    <!--ro, req, string, 当前用户登录密码, range:[1,16], attr:min{req, int, 最小值},max{req, int, 最大值}, desc:设备不返回密码内容-->null
  </password>
  <duressPassword min="1" max="16">
    <!--ro, opt, string, 挟持密码, range:[1,16], attr:min{req, int, 最小值},max{req, int, 最大值}, desc:设备不返回密码内容-->null
  </duressPassword>
  <keypadPassword min="1" max="16">
    <!--ro, opt, string, 键盘密码, range:[1,16], attr:min{req, int, 最小值},max{req, int, 最大值}, desc:设备不返回密码内容-->null
  </keypadPassword>
  <singleKeypadEnable opt="true,false">
    <!--ro, opt, bool, 单次键盘密码使能, attr:opt{req, string, 取值范围}-->true
  </singleKeypadEnable>
  <invalidKeypadPassword opt="true,false">
    <!--ro, opt, bool, 当前键盘密码是否无效, attr:opt{req, string, 取值范围}, desc:singleKeypadEnable配置后有效-->true
  </invalidKeypadPassword>
  <userOperateType opt="1,2,3">
    <!--ro, opt, enum, 用户操作类型, subType:int, [1#网络用户(云端创建的operator),2#键盘用户(局域网(本地)创建的operator),3#网络用户+键盘用户(云端分享的operator)], attr:opt{req, string, 取值范围}, desc:云端分享的operator可以进行云端、Web、本地(指的是通过键盘、遥控器、卡片登录)登录, 云端创建的operator和局域网创建的operator都只能通过本地登录.-->1
  </userOperateType>
  <bondIpAddressList>
    <!--ro, opt, array, 绑定用户的socket IP地址列表, subType:object-->
    <bondIpAddress>
      <!--ro, opt, object, 绑定用户的socket IP地址-->
      <id>
        <!--ro, req, int, 绑定IP的序号, range:[1,16]-->1
      </id>
      <ipAddress>
        <!--ro, opt, string, 绑定的IPv4地址, range:[1,32]-->test
      </ipAddress>
      <ipv6Address>
        <!--ro, opt, string, 绑定的IPv6地址, range:[1,32]-->test
      </ipv6Address>
    </bondIpAddress>
  </bondIpAddressList>

```

```

</ipv6Address>
</bondIpAddress>
</bondIpAddressList>
<bondMacAddressList>
  <!--ro, opt, array, 绑定用户的mac地址列表, subType:object-->
  <bondMacAddress>
    <!--ro, opt, object, 绑定用户的mac地址-->
    <id>
      <!--ro, req, int, 绑定mac地址的序号, range:[1,16]-->1
    </id>
    <macAddress>
      <!--ro, opt, string, 绑定的mac地址, range:[1,32]-->test
    </macAddress>
  </bondMacAddress>
</bondMacAddressList>
<userLevel opt="Administrator,Operator,Viewer,installer,manufacturer,single,installerAdmin,installEmployee,Employee,DepartmentManager">
  <!--ro, opt, enum, 用户级别, subType:string, [Administrator#管理员,Operator#操作员,Viewer#Viewer（设备端暂未使用）,installer#安装商,manufacturer#制造商,single#单用户（设备端暂未使用）,installerAdmin#安装管理员,installEmployee#安装雇员,Employee#普通员工（考勤）,DepartmentManager#部门主管（考勤）], attr:opt{req, string, 取值范围}-->Administrator
</userLevel>
<attribute>
  <!--ro, opt, object, 用户特殊属性配置-->
  <inherent>
    <!--ro, opt, bool, 是否可被删除, desc:true代表着该用户不可删除,false或者不支持该字段代表着该用户可被删除-->true
  </inherent>
</attribute>
<belongUserId>
  <!--ro, opt, int, 所属上级用户的id, range:[0,16], desc:如果上级没有用户,该值为0-->1
</belongUserId>
<belongOrgNo>
  <!--ro, opt, int, 所属组织编号, range:[1,16]-->1
</belongOrgNo>
<cardList>
  <!--ro, opt, object, 该用户所拥有的卡列表-->
  <card>
    <!--ro, opt, object, 卡-->
    <id>
      <!--ro, opt, int, 编号, range:[1,16]-->1
    </id>
    <name>
      <!--ro, opt, string, 名称, range:[1,32]-->test
    </name>
  </card>
</cardList>
<remoteCtrlList>
  <!--ro, opt, array, 该用户所拥有的遥控器列表, subType:object-->
  <remoteCtrl>
    <!--ro, opt, object, 遥控器-->
    <id>
      <!--ro, opt, int, 编号, range:[1,16]-->1
    </id>
    <name>
      <!--ro, opt, string, 名称, range:[1,32]-->test
    </name>
  </remoteCtrl>
</remoteCtrlList>
<userNo min="501" max="null">
  <!--ro, opt, int, 用户ID对应的界面展示编号, attr:min{req, int, 最小值},max{req, int, 最大值}, desc:该编号用于App和Web做界面展示,以及上报ARC,针对所有用户类型都有效,实现方式为500+id.由于某些ARC不支持通过用户名来区分用户,所以新增了userNo字段.使用场景:针对于公寓场景,很多时候几家住户会共用一台报警主机,并签订ARC服务,其中只有一户当中的一个会作为管理员;此时,ARC需要在添加其他不同住户的User时,将主机的User ID和实际每户的APP使用者关联起来,方便出问题能及时联系到对应的用户;所有用户在创建自己的APP账户时,也需要知道自己在ARC是几号用户,用做关联.-->1
</userNo>
<adminType opt="Cloud,LAN">
  <!--ro, opt, enum, admin类型, subType:string, [Cloud#云端admin,LAN#局域网admin], attr:opt{req, string, 取值范围}, desc:用于混合报警主机区分该admin用户是云端创建的,还是本地创建的,并在App和Web做界面展示-->Cloud
</adminType>
<installerType opt="Cloud,LAN">
  <!--ro, opt, enum, installer类型, subType:string, [Cloud#云端installer,LAN#局域网installer], attr:opt{req, string, 取值范围}, desc:用于混合报警主机区分该installer用户是云端创建的,还是本地创建,并在App和Web做界面展示-->Cloud
</installerType>
<PasswordPeriodPrompt>
  <!--ro, opt, object, 密码有效期提醒配置, desc:当密码超过所配置的有效期时,在登录时会给出密钥过期的提示,但是不修改密码也不会影响密码的使用-->
  <passwordPeriodType opt="permanent,byDate" def="permanent">
    <!--ro, req, enum, 密码有效期提醒类型, subType:string, [permanent#永久生效,byDate#按天数生效], attr:opt{req, string, 取值范围},def{req, string, 默认值}, desc:永久生效则该密码永远不会提示密码过期;按天数生效则到密码有效时间时会给出密钥过期的提示,但是不修改密码也不会影响密码的使用-->permanent
  </passwordPeriodType>
  <passwordPeriodbyDate min="1" max="1200" def="90">
    <!--ro, opt, int, 密码按天数生效有效时间, range:[1,1200], unit:天, unitType:时间, dep:and,{$.User.PasswordPeriodPrompt.passwordPeriodType,eq,byDate}, attr:min{req, int, 最小值},max{req, int, 最大值},def{req, int, 默认值}-->90
  </passwordPeriodbyDate>
</PasswordPeriodPrompt>
<userActivationStatus opt="true,false">
  <!--ro, opt, bool, 用户激活状态, attr:opt{req, string, 取值范围}, desc:不返回默认为true.true#已激活,false#未激活-->true
</userActivationStatus>
<invalidDuressPassword opt="true,false">
  <!--ro, opt, bool, 当前挟持密码是否无效, attr:opt{req, string, 取值范围}-->true
</invalidDuressPassword>
<userID min="1" max="64">
  <!--ro, opt, string, 用户编号, range:[1,64], attr:min{req, int, 最小值},max{req, int, 最大值}, desc:UUID,用户唯一标识-->test
</userID>
</User>

```

69.2.12 获取全部用户信息

Request URL

GET /ISAPI/Security/users?security=<security>&iv=<iv>

查询参数

参数名称	参数类型	描述
security	string	没有security表示数据不加密；security=1,表示报文中的password、keypadPassword和loginPassword字段AES128加密；security=2,表示报文中的password、keypadPassword和loginPassword字段AES256加密；
iv	string	iv表示初始化向量，在security=1或者2时为必须参数;

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<UserList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 用户列表, attr:version{req, string, 协议版本}-->
  <User>
    <!--ro, opt, object, 用户-->
    <id>
      <!--ro, req, int, 用户参数序号, range:[1,64]-->1
    </id>
    <enabled>
      <!--ro, opt, bool, 用户使能标志-->true
    </enabled>
    <userName>
      <!--ro, req, string, 用户名, range:[1,32]-->test
    </userName>
    <phoneNumber>
      <!--ro, opt, string, 电话号码, range:[1,11], desc:需要敏感信息加密-->test
    </phoneNumber>
    <emailAddress>
      <!--ro, opt, string, 邮件地址, range:[1,32], desc:需要敏感信息加密-->test
    </emailAddress>
    <duressPassword>
      <!--ro, opt, string, 挟持密码, range:[1,16], desc:需要敏感信息加密-->test
    </duressPassword>
    <keypadPassword>
      <!--ro, opt, string, 键盘密码, range:[1,16], desc:需要敏感信息加密,当前通过外设键盘对报警主机进行布防撤防时需要输入该密码-->test
    </keypadPassword>
    <singleKeypadEnable>
      <!--ro, opt, bool, 单次键盘密码使能-->true
    </singleKeypadEnable>
    <invalidKeypadPassword>
      <!--ro, opt, bool, 当前键盘密码是否无效, desc:singleKeypadEnable配置后有效-->true
    </invalidKeypadPassword>
    <userOperateType>
      <!--ro, opt, enum, 用户操作类型, subType:int, [1#网络用户, 2#键盘用户, 3#网络用户+键盘用户],
      desc:1. 当用户配置为网络用户（userOperateType为1或不填）时password有效,当方法为POST时password必填，否则可选
      2. 当用户配置为键盘用户（userOperateType传递为2）时keypadPassword有效,当方法为POST时keypadPassword必填，否则可选
      3. 当用户配置为即是网络用户又是键盘用户（userOperateType传递为3）时password和keypadPassword都有效,当方法为POST时password和keypadPassword必填，否则可选-->1
    </userOperateType>
    <bondIpAddressList>
      <!--ro, opt, object, 绑定用户的socket IP地址列表-->
      <bondIpAddress>
        <!--ro, opt, object, 绑定用户的socket IP地址-->
        <id>
          <!--ro, req, int, 绑定IP的序号, range:[1,16], desc:从1开始-->1
        </id>
        <ipAddress>
          <!--ro, opt, string, 绑定的IPv4地址, range:[1,32]-->test
        </ipAddress>
        <ipv6Address>
          <!--ro, opt, string, 绑定的IPv6地址, range:[1,128]-->test
        </ipv6Address>
      </bondIpAddress>
    </bondIpAddressList>
    <bondMacAddressList>
      <!--ro, opt, object, 绑定用户的mac地址列表-->
      <bondMacAddress>
        <!--ro, opt, object, 绑定用户的mac地址-->
        <id>
          <!--ro, req, int, 绑定mac地址的序号, range:[1,16], desc:从1开始-->1
        </id>
        <macAddress>
          <!--ro, opt, string, 绑定的mac地址, range:[1,32]-->test
        </macAddress>
      </bondMacAddress>
    </bondMacAddressList>
  </User>
</UserList>
```

```

</macAddress>
</bondMacAddress>
</bondMacAddressList>
<userLevel>
  <!--ro, opt, enum, 用户级别, subType:string, [Administrator#管理员,Operator#操作员,Viewer#观察员,installer#安装员,manufacturer#制造者,single#单用户,installerAdmin#安装管理员,installEmployee#安装雇员,Employee#普通员工（考勤）,DepartmentManager#部门主管（考勤）]-->Administrator
</userLevel>
<attribute>
  <!--ro, opt, object, 用户特殊属性配置-->
<inherit>
  <!--ro, opt, bool, 是否可被删除, desc:true代表着该用户不可删除,false或者不支持该字段代表着该用户可被删除-->true
</inherit>
</attribute>
<cardList>
  <!--ro, opt, object, 该用户所拥有的卡列表-->
<card>
  <!--ro, opt, object, 卡-->
  <id>
    <!--ro, opt, int, 编号, range:[1,16]-->1
  </id>
  <name>
    <!--ro, opt, string, 名称, range:[1,32]-->test
  </name>
</card>
</cardList>
<remoteCtrlList>
  <!--ro, opt, object, 该用户所拥有的遥控器列表-->
<remoteCtrl>
  <!--ro, opt, object, 遥控器-->
  <id>
    <!--ro, opt, int, 编号, range:[1,16]-->1
  </id>
  <name>
    <!--ro, opt, string, 名称, range:[1,32]-->test
  </name>
</remoteCtrl>
</remoteCtrlList>
<userNo>
  <!--ro, opt, int, 用户ID对应的界面展示编号, desc:该编号用于App和Web做界面展示, 以及上报ARC, 针对所有用户类型都有效, 实现方式为500+id-->501
</userNo>
<adminType>
  <!--ro, opt, enum, admin类型, subType:string, [Cloud#云端admin,LAN#局域网admin], desc:用于混合报警主机区分该admin用户是云端创建的, 还是在App和Web做界面展示-->Cloud
</adminType>
<installerType>
  <!--ro, opt, enum, installer类型, subType:string, [Cloud#云端installer,LAN#局域网installer], desc:用于混合报警主机区分该installer用户是云端创建的, 还是本地创建, 并在App和Web做界面展示-->Cloud
</installerType>
<PasswordPeriodPrompt>
  <!--ro, opt, object, 密码有效期提醒配置, desc:当密码超过所配置的有效时期, 在登录时会给出密码过期的提示, 但是不修改密码也不会影响密码的使用-->
<passwordPeriodType>
  <!--ro, req, enum, 密码有效期提醒类型, subType:string, [permanent#永久生效,byDate#按天数生效], desc:永久生效则该密码永远不会提示密码过期;按天数生效则到密码有效时间时会给出密码过期的提示, 但是不修改密码也不会影响密码的使用-->permanent
</passwordPeriodType>
<passwordPeriodbyDate>
  <!--ro, opt, int, 密码按天数生效有效时间, range:[1,1200], unit:天, unitType:时间, dep:and, {$.User.PasswordPeriodPrompt.passwordPeriodType,eq,byDate}-->90
</passwordPeriodbyDate>
</PasswordPeriodPrompt>
<userActivationStatus>
  <!--ro, opt, bool, 用户激活状态, desc:不返回默认为true.true#已激活,false#未激活-->true
</userActivationStatus>
<invalidDuressPassword>
  <!--ro, opt, bool, 当前挟持密码是否无效, desc:若未配置挟持密码, 该字段为true-->true
</invalidDuressPassword>
<userID>
  <!--ro, opt, string, 用户编号, range:[1,64], desc:UUID,用户唯一标识-->test
</userID>
</User>
</UserList>

```

69.2.13 配置全部用户信息

Request URL

PUT /ISAPI/Security/users?security=<security>&iv=<iv>

查询参数

参数名称	参数类型	描述
security	string	没有security表示数据不加密； security=1,表示报文中的password、keypadPassword和loginPassword字段AES128加密； security=2,表示报文中的password、keypadPassword和loginPassword字段AES256加密；
iv	string	iv表示初始化向量，在security=1或者2时为必须参数,随机字符串，长度限制为32；

请求报文

湖南云舟电力科技有限公司
有效期：2024-08-15

```

<?xml version="1.0" encoding="UTF-8"?>

<UserList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, 用户列表, attr:version{req, string, 协议版本}-->
  <User>
    <!--opt, object, 用户-->
    <id>
      <!--req, int, 用户参数序号, range:[1,64]-->1
    </id>
    <enabled>
      <!--opt, bool, 用户使能标志-->true
    </enabled>
    <userName>
      <!--req, string, 用户名, range:[1,32]-->test
    </userName>
    <password>
      <!--wo, req, string, 用户登录密码, range:[1,16], desc:需进行敏感信息加密-->test
    </password>
    <keypadPassword>
      <!--wo, opt, string, 键盘密码, range:[1,16], desc:需要敏感信息加密,当前通过外设键盘对报警主机进行布防撤防时需要输入该密码-->test
    </keypadPassword>
    <userOperateType>
      <!--opt, enum, 用户操作类型, subType:int, [1#网络用户,2#键盘用户,3#网络用户+键盘用户],
      desc:1. 当用户配置为网络用户 (userOperateType为1或不填) 时password有效,当方法为POST时password必填, 否则可选
      2. 当用户配置为键盘用户 (userOperateType传递为2) 时keypadPassword有效,当方法为POST时keypadPassword必填, 否则可选
      3. 当用户配置为即是网络用户又是键盘用户 (userOperateType传递为3) 时password和keypadPassword都有效,当方法为POST时password和keypadPassword必填, 否则可选-->1
    </userOperateType>
    <bondIpAddressList>
      <!--opt, object, 绑定用户的socket IP地址列表-->
      <bondIpAddress>
        <!--opt, object, 绑定用户的socket IP地址-->
        <id>
          <!--req, int, 绑定IP的序号, range:[1,16], desc:从1开始-->1
        </id>
        <ipAddress>
          <!--opt, string, 绑定的IPv4地址, range:[1,32]-->test
        </ipAddress>
        <ipv6Address>
          <!--opt, string, 绑定的IPv6地址, range:[1,32]-->test
        </ipv6Address>
      </bondIpAddress>
    </bondIpAddressList>
    <bondMacAddressList>
      <!--opt, object, 绑定用户的mac地址列表-->
      <bondMacAddress>
        <!--opt, object, 绑定用户的mac地址-->
        <id>
          <!--req, int, 绑定mac地址的序号, range:[1,16], desc:从1开始-->1
        </id>
        <macAddress>
          <!--opt, string, 绑定的mac地址, range:[1,32]-->test
        </macAddress>
      </bondMacAddress>
    </bondMacAddressList>
    <userLevel>
      <!--opt, enum, 用户级别, subType:string, [Administrator#管理员,Operator#操作员,Viewer#观察员,installer#安装员,manufacturer#制造者,single#单用户,installerAdmin#安装管理员,installEmployee#安装雇员]-->Administrator
    </userLevel>
    <attribute>
      <!--opt, object, 用户特殊属性配置-->
      <inherent>
        <!--opt, bool, 是否可被删除, desc:true代表着该用户不可删除,false或者不支持该字段代表着该用户可被删除-->true
      </inherent>
    </attribute>
    <loginPassword>
      <!--req, string, 确认密码, range:[1,16], desc:新增用户, 修改用户信息的时候, 都会需要确认密码 (确认密码就是用户登录密码); 确认密码在HTTP传输必须强制敏感信息加密; 在HTTPS下可以不用单独加密, HCNetSDK内部默认加密; ISUP和萤石透传的时候, 默认链路加密.-->King726014
    </loginPassword>
    <PasswordPeriodPrompt>
      <!--opt, object, 密码有效期提醒配置, desc:当密码超过所配置的有效时期, 在登录时会给出密钥过期的提示,但是不修改密码也不会影响密码的使用-->
      <passwordPeriodType>
        <!--req, enum, 密码有效期提醒类型, subType:string, [permanent#永久生效,byDate#按天数生效], desc:永久生效则该密码永远不会提示密码过期;按天数生效则到密码有效时间时会给出密钥过期的提示,但是不修改密码也不会影响密码的使用-->permanent
      </passwordPeriodType>
      <passwordPeriodbyDate>
        <!--opt, int, 密码按天数生效有效时间, range:[1,1200], unit:天, unitType:时间, dep:and,{$.User.PasswordPeriodPrompt.passwordPeriodType,eq,byDate}-->90
      </passwordPeriodbyDate>
    </PasswordPeriodPrompt>
    <userID>
      <!--opt, string, 用户编号, range:[1,64], desc:UUID,用户唯一标识-->test
    </userID>
  </User>
</UserList>

```

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 响应消息, attr:version{ro, req, string, 版本}-->
  <requestURL>
    <!--ro, req, string, 请求的URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, 状态码, subType:int, [0#OK,1#OK,2#Device Busy,3#Device Error,4#Invalid Operation,5#Invalid XML Format,6#Invalid XML Content,7#Reboot Required]-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, 状态信息, subType:string, [0#成功,Device Busy#设备忙,Device Error#设备异常,Invalid Operation#无效的操作,Invalid XML Format#无效的XML格式,Invalid XML Content#无效的XML内容,Reboot#设备重启]-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, 详细错误码英文描述, desc:详细错误码的英文描述-->OK
  </subStatusCode>
  <lockStatus>
    <!--ro, opt, enum, 锁定状态, subType:string, [Lock#锁定,unLock#未锁定], desc:LoginPassword校验失败时返回-->lock
  </lockStatus>
  <retryTimes>
    <!--ro, opt, int, 剩余尝试次数, range:[0,6], desc:LoginPassword校验失败时返回-->1
  </retryTimes>
</ResponseStatus>
```

69.2.14 配置全部用户权限

Request URL

PUT /ISAPI/Security/UserPermission

查询参数

无

请求报文

```
<?xml version="1.0" encoding="UTF-8"?>

<UserPermissionList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, 用户权限列表, attr:version{req, string, 协议版本}-->
  <UserPermission>
    <!--req, object, 用户权限-->
    <id>
      <!--req, string, 索引值, range:[1,32]-->test
    </id>
    <userID>
      <!--req, string, 用户ID, range:[1,32]-->test
    </userID>
    <userType>
      <!--req, enum, 用户类型, subType:string, [admin#系统用户,operator#操作员,viewer#观察员,installer#安装员,manufacturer#制造者,single#单用户]-->admin
    </userType>
    <localPermission>
      <!--opt, object, 操作计划, dep:or,{$.UserPermissionList.UserPermission.LocalPermission.operationScheduleEnabled,eq,true}-->
      <backup>
        <!--opt, bool, 是否开启备份(所有通道)权限, desc:true为开启,false为关闭-->true
      </backup>
      <record>
        <!--opt, bool, 是否开启录像(所有通道)权限, desc:true为开启,false为关闭-->true
      </record>
      <playBack>
        <!--opt, bool, 是否开启回放(所有通道)权限, desc:true为开启,false为关闭-->true
      </playBack>
      <preview>
        <!--opt, bool, 是否开启预览(所有通道)权限, desc:true为开启,false为关闭-->true
      </preview>
      <videoChannelPermissionList>
        <!--opt, object, 视频通道权限配置列表-->
        <videoChannelPermission>
          <!--opt, object, 单一视频通道权限配置-->
          <id>
            <!--req, string, 视频通道ID, range:[1,32]-->test
          </id>
          <playBack>
            <!--opt, bool, 是否开启当前通道回放权限, desc:true为开启,false为关闭-->true
          </playBack>
          <preview>
            <!--opt, bool, 是否开启当前通道预览权限, desc:true为开启,false为关闭-->true
          </preview>
          <record>
            <!--opt, bool, 是否开启当前通道录像权限, desc:true为开启,false为关闭-->true
          </record>
          <backup>
            <!--opt, bool, 是否开启当前通道备份权限, desc:true为开启,false为关闭-->true
          </backup>
          <playBackDoubleVerification>
```

```
<!--opt, bool, 是否开启当前通道回放二次认证权限, desc:true为开启,false为关闭-->true
</playBackDoubleVerification>
<backupDoubleVerification>
  <!--opt, bool, 是否开启当前通道备份二次认证权限, desc:true为开启,false为关闭-->true
  </backupDoubleVerification>
</videoChannelPermission>
</videoChannelPermissionList>
<ptzControl>
  <!--req, bool, 是否开启云台控制(所有通道)权限, desc:true为开启,false为关闭-->true
</ptzControl>
<ptzChannelPermissionList>
  <!--opt, object, 视频通道云台控制权限列表-->
  <ptzChannelPermission>
    <!--req, object, 单一视频通道云台控制权限-->
    <id>
      <!--req, string, 视频通道ID, range:[1,32]-->test
    </id>
    <ptzControl>
      <!--opt, bool, 是否开启当前通道云台控制权限, desc:true为开启,false为关闭-->true
    </ptzControl>
  </ptzChannelPermission>
</ptzChannelPermissionList>
<logOrStateCheck>
  <!--opt, bool, 是否开启日志和设备状态查看权限, desc:true为开启,false为关闭-->true
</logOrStateCheck>
<parameterConfig>
  <!--opt, bool, 是否开启设备远程参数设置权限, desc:true为开启,false为关闭-->true
</parameterConfig>
<restartOrShutdown>
  <!--opt, bool, 是否开启重启或关机权限, desc:true为开启,false为关闭-->true
</restartOrShutdown>
<upgrade>
  <!--opt, bool, 是否开启升级权限, desc:true为开启,false为关闭-->true
</upgrade>
<manageChannel>
  <!--opt, bool, 是否开启通道管理权限, desc:true为开启,false为关闭-->true
</manageChannel>
<AIModelManagement>
  <!--opt, bool, 是否开启AI模型包操作权限, desc:true为开启,false为关闭-->true
</AIModelManagement>
<AITaskManagement>
  <!--opt, bool, 是否开启AI任务管理操作权限, desc:true为开启,false为关闭-->true
</AITaskManagement>
<subSysOrZoneArm>
  <!--opt, bool, 子系统/防区布防权限-->true
</subSysOrZoneArm>
<subSysOrZoneDisarm>
  <!--opt, bool, 子系统/防区撤防权限-->true
</subSysOrZoneDisarm>
<operateOutput>
  <!--opt, bool, 操作继电器权限-->true
</operateOutput>
<subSystemList>
  <!--opt, object, 子系统列表, desc:当用户添加或者配置operator时可关联-->
  <subSystem>
    <!--opt, int, 子系统, range:[1,64]-->1
  </subSystem>
</subSystemList>
<localFileMgr>
  <!--opt, bool, 本地文件管理权限, desc:指对本地文件的增删改查权限-->true
</localFileMgr>
<LEDConfig>
  <!--opt, bool, 是否允许用户修改LED配置-->true
</LEDConfig>
<logMgr>
  <!--opt, bool, 日志管理权限-->true
</logMgr>
<TIPConfig>
  <!--opt, bool, TIP配置权限-->true
</TIPConfig>
<securityFunctionCfg>
  <!--opt, bool, 安检功能配置权限, desc:针对安检类目的权限配置-->true
</securityFunctionCfg>
<networkCfg>
  <!--opt, bool, 网络配置权限, desc:包括无线和有线网络-->true
</networkCfg>
<ISUPAccess>
  <!--opt, bool, ISUP接入配置权限, desc:指设备接入ISUP的权限配置,对应ISUP接入领域-->true
</ISUPAccess>
<timeMgr>
  <!--opt, bool, 时间管理权限, desc:指系统时间和时区管理的权限,对应时间管理领域-->true
</timeMgr>
<HDDMgr>
  <!--opt, bool, 硬盘管理权限, desc:指硬盘管理的权限,对应硬盘管理领域-->true
</HDDMgr>
<securitySystemMaintenance>
  <!--opt, bool, 安检机系统维护权限, desc:包含探测器测试、射线源测试、操作键盘测试、系统诊断、维护记录、其他检测（主控板、IO、射线源灯）-->true
</securitySystemMaintenance>
<securityCheckingPicExport>
  <!--opt, bool, 安检过包检测图片导出权限-->true
</securityCheckingPicExport>
<securityCheckingPicDelete>
  <!--opt, bool, 安检过包检测图片删除权限-->true
</securityCheckingPicDelete>
```

```

<operationScheduleList>
  <!--opt, array, 操作计划列表, subType:object, dep:or,{$.UserPermissionList.UserPermission.LocalPermission.operationScheduleEnabled,eq,true}-->
  <operationSchedule>
    <!--opt, object, 操作计划-->
    <operationScheduleEnabled>
      <!--opt, bool, 操作计划使能-->true
    </operationScheduleEnabled>
    <operationDetailItem>
      <!--opt, enum, 操作具体项, subType:string, [all#所有操作,arm#布防,disarm#撤防,operateOutput#继电器操作]-->arm
    </operationDetailItem>
    <timeRange>
      <!--opt, object, 时间范围-->
      <beginTime>
        <!--opt, time, 开始时间-->00:00:00
      </beginTime>
      <endTime>
        <!--opt, time, 结束时间-->00:00:00
      </endTime>
    </timeRange>
  </operationSchedule>
</operationScheduleList>
</localPermission>
<remotePermission>
  <!--opt, object, 远程权限节点-->
  <record>
    <!--opt, bool, 是否开启录像（所有通道）权限, desc:true为开启,false为关闭-->true
  </record>
  <playBack>
    <!--opt, bool, 是否开启回放（所有通道）权限, desc:true为开启,false为关闭-->true
  </playBack>
  <preview>
    <!--opt, bool, 是否开启预览（所有通道）权限, desc:true为开启,false为关闭-->true
  </preview>
  <videoChannelPermissionList>
    <!--opt, object, 视频通道权限配置列表-->
    <videoChannelPermission>
      <!--opt, object, 单一视频通道权限配置-->
      <id>
        <!--req, string, 视频通道ID, range:[1,32]-->test
      </id>
      <preview>
        <!--opt, bool, 是否开启当前通道预览权限, desc:true为开启,false为关闭-->true
      </preview>
      <playBack>
        <!--opt, bool, 是否开启当前通道回放权限, desc:true为开启,false为关闭-->true
      </playBack>
      <record>
        <!--opt, bool, 是否开启当前通道录像权限, desc:true为开启,false为关闭-->true
      </record>
      <playBackDoubleVerification>
        <!--opt, bool, 是否开启当前通道回放二次认证权限, desc:true为开启,false为关闭-->true
      </playBackDoubleVerification>
    </videoChannelPermission>
  </videoChannelPermissionList>
  <ptzControl>
    <!--opt, bool, 是否开启云台控制（所有通道）权限, desc:true为开启,false为关闭-->true
  </ptzControl>
  <ptzChannelPermissionList>
    <!--opt, object, 视频通道云台控制权限列表-->
    <ptzChannelPermission>
      <!--opt, object, 单一视频通道云台控制权限-->
      <id>
        <!--req, string, 视频通道ID, range:[1,32], desc:/ISAPI/PTZCtrl/channels/ID-->test
      </id>
      <ptzControl>
        <!--opt, bool, 是否开启当前通道云台控制权限, desc:true为开启,false为关闭-->true
      </ptzControl>
    </ptzChannelPermission>
  </ptzChannelPermissionList>
  <logOrStateCheck>
    <!--opt, bool, 是否开启日志和设备状态查看权限-->true
  </logOrStateCheck>
  <parameterConfig>
    <!--opt, bool, 是否开启设备远程参数设置权限-->true
  </parameterConfig>
  <restartOrShutdown>
    <!--opt, bool, 是否开启重启或关机权限-->true
  </restartOrShutdown>
  <upgrade>
    <!--opt, bool, 是否开启升级权限-->true
  </upgrade>
  <upgradeNotShow>
    <!--opt, bool, 升级不展示, desc:为确保兼容, 设备端admin,installer用户升级权限也支持, 该字段返回true的话, 上层界面隐藏admin和installer升级权限的相关
    界面, 前后端暂没有installer,admin等用户,EN认证过程用web进行操作要求升级不能和installer和admin进行绑定,手机app也会用到该协议, 设备如果不返回升级权限字段那么
    手机app就没有升级功, 因此增加该字段-->true
  </upgradeNotShow>
  <voiceTalk>
    <!--opt, bool, 是否开启语音对讲（所有通道）权限-->true
  </voiceTalk>
  <transParentChannel>
    <!--opt, bool, 是否开启透明通道（所有通道）权限, desc:也即控制串口-->true
  </transParentChannel>
  <contorlLocalOut>

```

```
<!--opt, bool, 是否开启控制本地输出权限-->true
</controlLocalOut>
<manageChannel>
  <!--opt, bool, 是否开启通道管理权限-->true
</manageChannel>
<alarmOutOrUpload>
  <!--opt, bool, 是否开启报警上报和输出权限-->true
</alarmOutOrUpload>
<IRAID>
  <!--opt, object, Iraid快速权限配置-->
  <rapidConfiguration>
    <!--opt, bool, 是否开启Iraid快速配置权限-->true
  </rapidConfiguration>
  <reset>
    <!--opt, bool, 是否开启Iraid重设配置权限-->true
  </reset>
</IRAID>
<LEDConfig>
  <!--opt, bool, 是否允许用户修改LED配置-->true
</LEDConfig>
<AIModelManagement>
  <!--opt, bool, 是否开启AI模型包操作权限-->true
</AIModelManagement>
<AITaskManagement>
  <!--opt, bool, 是否开启AI任务管理操作权限-->true
</AITaskManagement>
<materialReview>
  <!--opt, bool, 素材审核权限, desc:信息发布-->true
</materialReview>
<programReview>
  <!--opt, bool, 节目审核权限, desc:信息发布-->true
</programReview>
<uploadMaterial>
  <!--opt, bool, 上传素材, desc:信息发布-->true
</uploadMaterial>
<addProgram>
  <!--opt, bool, 节目编辑权限, desc:增, 删, 改-->true
</addProgram>
<addSchedule>
  <!--opt, bool, 日程编辑权限, desc:增, 删, 改-->true
</addSchedule>
<programPlay>
  <!--opt, bool, 信息播放权限-->true
</programPlay>
<terminalMgr>
  <!--opt, bool, 终端管理权限, desc:对终端的背光, Logo灯等参数设置-->true
</terminalMgr>
<shopSmart>
  <!--opt, bool, 商业智能权限-->true
</shopSmart>
<materialReplace>
  <!--opt, bool, 素材替换权限-->true
</materialReplace>
<scheduleRelease>
  <!--opt, bool, 日程发布权限-->true
</scheduleRelease>
<scheduleReview>
  <!--opt, bool, 日程审核-->true
</scheduleReview>
<addTerminal>
  <!--opt, bool, 终端添加-->true
</addTerminal>
<terminalDispatch>
  <!--opt, bool, 终端分配权限, desc:将该组织下的终端分配给于组织-->true
</terminalDispatch>
<addUser>
  <!--opt, bool, 用户编辑权限, desc:增, 删, 改-->true
</addUser>
<addOrg>
  <!--opt, bool, 组织编辑权限, desc:增, 删, 改-->true
</addOrg>
<logMgr>
  <!--opt, bool, 日志管理-->true
</logMgr>
<dataUpdate>
  <!--opt, bool, 数据更新, desc:叫号数据-->true
</dataUpdate>
<insertMessage>
  <!--opt, bool, 插播消息权限-->true
</insertMessage>
<notReview>
  <!--opt, bool, 无需审核权限-->true
</notReview>
<MgrTerminal>
  <!--opt, object, 管理的终端或终端组-->
  <TerminalNoList>
    <!--opt, object, 终端号列表-->
    <terminalNo>
      <!--opt, int, 终端号, range:[1,10]-->1
    </terminalNo>
  </TerminalNoList>
  <TerminalGroupNoList>
    <!--opt, object, 终端组列表-->
    <terminalGroupNo>
```

湖南云舟电力科技有限公司
2024-08-15

```

<!--opt, int, 终端组号, range:[1,10]-->1
</terminalGroupNo>
</TerminalGroupNoList>
</MgrTerminal>
<subSysOrZoneArm>
  <!--opt, bool, 子系统/防区布防权限-->true
</subSysOrZoneArm>
<subSysOrZoneDisarm>
  <!--opt, bool, 子系统/防区撤防权限-->true
</subSysOrZoneDisarm>
<subSysOrZoneClearArm>
  <!--opt, bool, 子系统/防区消警权限-->true
</subSysOrZoneClearArm>
<zoneBypass>
  <!--opt, bool, 防区旁路权限-->true
</zoneBypass>
<zoneBypassRecover>
  <!--opt, bool, 防区旁路恢复权限-->true
</zoneBypassRecover>
<operateOutput>
  <!--opt, bool, 操作继电器权限-->true
</operateOutput>
<subSystemList>
  <!--opt, object, 子系统列表, desc:当用户添加或者配置operator时可关联子系统-->
  <subSystem>
    <!--opt, int, 子系统, range:[1,64]-->1
  </subSystem>
</subSystemList>
<factoryReset>
  <!--opt, bool, 恢复默认参数-->true
</factoryReset>
<arm>
  <!--opt, bool, 布防-->true
</arm>
<disarm>
  <!--opt, bool, 撤防-->true
</disarm>
<LPListAudit>
  <!--opt, bool, 黑白名单操作权限,
desc:有这个权限的用户可以导入/导出/编辑/删除/查询黑白名单
对应接口如下:
车牌黑白名单查询:/ISAPI/Traffic/channels/<ID>/searchLPListAudit
导入/导出车牌黑白名单数据:/ISAPI/Traffic/channels/<ID>/LicensePlateAuditData
车牌名单添加修改:/ISAPI/Traffic/channels/<channelID>/LicensePlateAuditData/record?format=json
车牌黑白名单数据删除:/ISAPI/Traffic/channels/<channelID>/LicensePlateAuditData/delete?format=json-->true
  </LPListAudit>
<applicationWarehouse>
  <!--opt, bool, 应用仓库管理权限, desc:不支持不返回。具备管理权限, 用户可进行导入、删除、启动HEOP APP等操作-->true
</applicationWarehouse>
<subSystemManagement>
  <!--opt, bool, 子系统管理权限, desc:包含子系统管理相关的所有权限-->true
</subSystemManagement>
<userOfModifyKeypadPasswordList>
  <!--opt, array, 允许修改键盘密码的用户列表, subType:object-->
  <userOfModifyKeypadPassword>
    <!--opt, object, 允许修改键盘密码的用户-->
  </subUserType>
  <!--opt, enum, 子用户类型, subType:string, [cCloudAdmin#云端admin, LocalAdmin#本地admin, installerAdmin#安装管理员, installEmployee#安装雇员, LocalInstaller#本地installer, cCloudOperator#云端操作员, LocalOperator#本地操作员]-->cCloudAdmin
  </subUserType>
</userOfModifyKeypadPassword>
</userOfModifyKeypadPasswordList>
<userOfModifyUserPermissionList>
  <!--opt, array, 允许修改用户权限的用户列表, subType:object-->
  <userOfModifyUserPermission>
    <!--opt, object, 允许修改用户权限的用户-->
  </subUserType>
  <!--opt, enum, 子用户类型, subType:string, [cCloudAdmin#云端admin, LocalAdmin#本地admin, installerAdmin#安装管理员, installEmployee#安装雇员, LocalInstaller#本地installer, cCloudOperator#云端操作员, LocalOperator#本地操作员]-->cCloudAdmin
  </subUserType>
</userOfModifyUserPermission>
</userOfModifyUserPermissionList>
<falseAlarmMaterialLibMgr>
  <!--opt, bool, 误报分析素材库管理权限-->true
</falseAlarmMaterialLibMgr>
<TIPConfig>
  <!--opt, bool, TIP配置权限-->true
</TIPConfig>
<networkCfg>
  <!--opt, bool, 网络配置权限, desc:包括无线和有线网络-->true
</networkCfg>
<ISUPAccess>
  <!--opt, bool, ISUP接入配置权限, desc:指设备接入ISUP的权限配置, 对应ISUP接入领域-->true
</ISUPAccess>
<timeMgr>
  <!--opt, bool, 时间管理权限, desc:指系统时间和时区管理的权限, 对应时间管理领域-->true
</timeMgr>
<HDDMgr>
  <!--opt, bool, 硬盘管理权限, desc:指硬盘管理的权限, 对应硬盘管理领域-->true
</HDDMgr>
</remotePermission>
<passwordValidity>
  <!--opt, int, 密码有效期, range:[0,365], desc:仅管理员用户可以修改, 单位: 天, [0, 365]-->1

```

```
</passwordValidity>
<subUserType>
  <!--opt, enum, 子用户类型, subType:string, [cLoudAdmin#云端admin, LocalAdmin#本地admin, installerAdmin#安装管理员, installerEmployee#安装雇员, LocalInstaller#本地installer, cLoudOperator#云端操作员, LocalOperator#本地操作员]-->cLoudAdmin
</subUserType>
</UserPermission>
</UserPermissionList>
```

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 响应消息, attr:version{ro, req, string, 版本}-->
  <requestURL>
    <!--ro, req, string, 请求的URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, 状态码, subType:int, [0#OK,1#OK,2#Device Busy,3#Device Error,4#Invalid Operation,5#Invalid XML Format,6#Invalid XML Content,7#Reboot Required,8#Batch Operation]-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, 状态信息, subType:string, [OK#成功,Device Busy#设备忙,Device Error#设备异常,Invalid Operation#无效的操作,Invalid XML Format#无效的XML格式,Invalid XML Content#无效的XML内容,Reboot#设备重启]-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, 详细错误码英文描述, desc:详细错误码的英文描述-->OK
  </subStatusCode>
  <AttachInfo>
    <!--ro, opt, object, 批量操作返回状态附加信息, desc:statusCode为8 (Batch Operation#批量操作)时返回-->
    <StatusList>
      <!--ro, req, object, 状态信息列表-->
      <Status>
        <!--ro, req, object, 状态信息-->
        <index>
          <!--ro, req, int, 编号, desc:用于识别批量添加时,添加的List列表中的每一个资源信息,与批量添加的资源参数一一对应-->0
        </index>
        <statusCode>
          <!--ro, req, enum, 状态码, subType:int, [0#OK,1#OK,2#Device Busy,3#Device Error,4#Invalid Operation,5#Invalid Format,6#Invalid Content,7#Reboot Required], desc:这里只表示子系统操作的状态,返回方式与上面一致-->0
        </statusCode>
        <statusString>
          <!--ro, req, enum, 状态信息, subType:string, [OK#成功,Device Busy#设备忙,Device Error#设备异常,Invalid Operation#无效的操作,Invalid Format#无效的格式,Invalid Content#无效的内容,Reboot#设备重启], desc:这里只表示子系统操作的状态,返回方式与上面一致-->OK
        </statusString>
        <subStatusCode>
          <!--ro, req, string, 详细错误码英文描述, range:[1,64], desc:详细错误码的英文描述,这里只表示子系统操作的状态,返回方式与上面一致-->ok
        </subStatusCode>
      </Status>
    </StatusList>
  </AttachInfo>
</ResponseStatus>
```

69.2.15 获取全部用户权限

Request URL

GET /ISAPI/Security/UserPermission

查询参数

无

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<UserPermissionList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, array, 用户权限列表, subType:object, attr:version{req, string, 协议版本}-->
  <UserPermission>
    <!--ro, req, object, 用户权限-->
    <id>
      <!--ro, req, string, 索引值, range:[1,32]-->test
    </id>
    <userID>
      <!--ro, req, string, 用户ID, range:[1,32]-->test
    </userID>
    <userType>
      <!--ro, req, enum, 用户类型, subType:string, [admin#系统用户,operator#操作员,viewer#观察员,installer#安装员,manufacturer#制造者,single#单用户]-->admin
    </userType>
    <localPermission>
      <!--ro, opt, object, 本地权限节点-->
    </localPermission>
  </UserPermission>
</UserPermissionList>
```

```
<!--ro, opt, bool, 是否开启备份(所有通道)权限, desc:true为开启,false为关闭-->true
</backup>
<record>
<!--ro, opt, bool, 是否开启录像(所有通道)权限, desc:true为开启,false为关闭-->true
</record>
<playBack>
<!--ro, opt, bool, 是否开启回放(所有通道)权限, desc:true为开启,false为关闭-->true
</playBack>
<preview>
<!--ro, opt, bool, 是否开启预览(所有通道)权限, desc:true为开启,false为关闭-->true
</preview>
<videoChannelPermissionList>
<!--ro, opt, object, 视频通道权限配置列表-->
<videoChannelPermission>
<!--ro, opt, object, 单一视频通道权限配置-->
<id>
<!--ro, req, string, 视频通道ID, range:[1,32]-->test
</id>
<playBack>
<!--ro, opt, bool, 是否开启当前通道回放权限, desc:true为开启,false为关闭-->true
</playBack>
<preview>
<!--ro, opt, bool, 是否开启当前通道预览权限, desc:true为开启,false为关闭-->true
</preview>
<record>
<!--ro, opt, bool, 是否开启当前通道录像权限, desc:true为开启,false为关闭-->true
</record>
<backup>
<!--ro, opt, bool, 是否开启当前通道备份权限, desc:true为开启,false为关闭-->true
</backup>
<playBackDoubleVerification>
<!--ro, opt, bool, 是否开启当前通道回放二次认证权限, desc:true为开启,false为关闭-->true
</playBackDoubleVerification>
<backupDoubleVerification>
<!--ro, opt, bool, 是否开启当前通道备份二次认证权限, desc:true为开启,false为关闭-->true
</backupDoubleVerification>
</videoChannelPermission>
</videoChannelPermissionList>
<ptzControl>
<!--ro, req, bool, 是否开启云台控制(所有通道)权限, desc:true为开启,false为关闭-->true
</ptzControl>
<ptzChannelPermissionList>
<!--ro, opt, object, 视频通道云台控制权限列表-->
<ptzChannelPermission>
<!--ro, req, object, 单一视频通道云台控制权限-->
<id>
<!--ro, req, string, 视频通道ID, range:[1,32]-->test
</id>
<ptzControl>
<!--ro, opt, bool, 是否开启当前通道云台控制权限, desc:true为开启,false为关闭-->true
</ptzControl>
</ptzChannelPermission>
</ptzChannelPermissionList>
<logOrStateCheck>
<!--ro, opt, bool, 是否开启日志和设备状态查看权限, desc:true为开启,false为关闭-->true
</logOrStateCheck>
<parameterConfig>
<!--ro, opt, bool, 是否开启设备远程参数设置权限, desc:true为开启,false为关闭-->true
</parameterConfig>
<restartOrShutdown>
<!--ro, opt, bool, 是否开启重启或关机权限, desc:true为开启,false为关闭-->true
</restartOrShutdown>
<upgrade>
<!--ro, opt, bool, 是否开启升级权限, desc:true为开启,false为关闭-->true
</upgrade>
<manageChannel>
<!--ro, opt, bool, 是否开启通道管理权限, desc:true为开启,false为关闭-->true
</manageChannel>
<AIModelManagement>
<!--ro, opt, bool, 是否开启AI模型包操作权限, desc:true为开启,false为关闭-->true
</AIModelManagement>
<AITaskManagement>
<!--ro, opt, bool, 是否开启AI任务管理操作权限, desc:true为开启,false为关闭-->true
</AITaskManagement>
<subSysOrZoneArm>
<!--ro, opt, bool, 子系统/防区布防权限-->true
</subSysOrZoneArm>
<subSysOrZoneDisarm>
<!--ro, opt, bool, 子系统/防区撤防权限-->true
</subSysOrZoneDisarm>
<operateOutput>
<!--ro, opt, bool, 操作继电器权限-->true
</operateOutput>
<subSystemList>
<!--ro, opt, object, 子系统列表, desc:当用户添加或者配置operator时可关联-->
<subSystem>
<!--ro, opt, int, 子系统, range:[1,64]-->1
</subSystem>
</subSystemList>
<localFileMgr>
<!--ro, opt, bool, 本地文件管理权限, desc:指对本地文件的增删改查权限-->true
</localFileMgr>
```

```
<LEDConfig>
  <!--ro, opt, bool, 是否允许用户修改LED配置-->true
</LEDConfig>
<logMgr>
  <!--ro, opt, bool, 日志管理权限-->true
</logMgr>
<TIPConfig>
  <!--ro, opt, bool, TIP配置权限-->true
</TIPConfig>
<securityFunctionCfg>
  <!--ro, opt, bool, 安检功能配置权限, desc: 针对安检类目的权限配置-->true
</securityFunctionCfg>
<networkCfg>
  <!--ro, opt, bool, 网络配置权限, desc: 包括无线和有线网络-->true
</networkCfg>
<ISUPAccess>
  <!--ro, opt, bool, ISUP接入配置权限, desc: 指设备接入ISUP的权限配置, 对应ISUP接入领域-->true
</ISUPAccess>
<timeMgr>
  <!--ro, opt, bool, 时间管理权限, desc: 指系统时间和时区管理的权限, 对应时间管理领域-->true
</timeMgr>
<HDDMgr>
  <!--ro, opt, bool, 硬盘管理权限, desc: 指硬盘管理的权限, 对应硬盘管理领域-->true
</HDDMgr>
<securitySystemMaintenance>
  <!--ro, opt, bool, 安检机系统维护权限, desc: 包含探测器测试、射线源测试、操作键盘测试、系统诊断、维护记录、其他检测（主控板、IO、射线源灯）-->true
</securitySystemMaintenance>
<securityCheckingPicExport>
  <!--ro, opt, bool, 安检过包检测图片导出权限-->true
</securityCheckingPicExport>
<securityCheckingPicDelete>
  <!--ro, opt, bool, 安检过包检测图片删除权限-->true
</securityCheckingPicDelete>
<operationScheduleList>
  <!--ro, opt, array, 操作计划列表, subType:object, dep:or,{$.UserPermissionList.UserPermission.LocalPermission.operationScheduleEnabled,eg,true}-->
  <operationSchedule>
    <!--ro, opt, object, 操作计划-->
    <operationScheduleEnabled>
      <!--ro, opt, bool, 操作计划使能-->true
    </operationScheduleEnabled>
    <operationDetailItem>
      <!--ro, opt, enum, 操作具体项, subType:string, [all#所有操作,arm#布防,disarm#撤防,operateOutput#继电器操作]-->arm
    </operationDetailItem>
    <timeRange>
      <!--ro, opt, object, 时间范围-->
      <beginTime>
        <!--ro, opt, time, 开始时间-->00:00:00
      </beginTime>
      <endTime>
        <!--ro, opt, time, 结束时间-->00:00:00
      </endTime>
    </timeRange>
  </operationSchedule>
</operationScheduleList>
</localPermission>
<remotePermission>
  <!--ro, opt, object, 远程权限节点-->
  <record>
    <!--ro, opt, bool, 是否开启录像（所有通道）权限, desc:true为开启,false为关闭-->true
  </record>
  <playBack>
    <!--ro, opt, bool, 是否开启回放（所有通道）权限, desc:true为开启,false为关闭-->true
  </playBack>
  <preview>
    <!--ro, opt, bool, 是否开启预览（所有通道）权限, desc:true为开启,false为关闭-->true
  </preview>
  <videoChannelPermissionList>
    <!--ro, opt, object, 视频通道权限配置列表-->
    <videoChannelPermission>
      <!--ro, opt, object, 单一视频通道权限配置-->
      <id>
        <!--ro, req, string, 视频通道ID, range:[1,32]-->test
      </id>
      <preview>
        <!--ro, opt, bool, 是否开启当前通道预览权限, desc:true为开启,false为关闭-->true
      </preview>
      <playBack>
        <!--ro, opt, bool, 是否开启当前通道回放权限, desc:true为开启,false为关闭-->true
      </playBack>
      <record>
        <!--ro, opt, bool, 是否开启当前通道录像权限, desc:true为开启,false为关闭-->true
      </record>
      <playBackDoubleVerification>
        <!--ro, opt, bool, 是否开启当前通道回放二次认证权限, desc:true为开启,false为关闭-->true
      </playBackDoubleVerification>
    </videoChannelPermission>
  </videoChannelPermissionList>
  <ptzControl>
    <!--ro, opt, bool, 是否开启云台控制（所有通道）权限, desc:true为开启,false为关闭-->true
  </ptzControl>
  <ptzChannelPermissionList>
    <!--ro, opt, object, 视频通道云台控制权限列表-->
    <ptzChannelPermission>
      <!--ro, opt, object, 单一视频通道云台控制权限-->
```

```
<!--ro, opt, boolean, 是否开启设备升级权限-->
<id>
<!--ro, req, string, 视频通道ID, range:[1,32], desc:/ISAPI/PTZCtrl/channels/ID-->test
</id>
<ptzControl>
<!--ro, opt, bool, 是否开启当前通道云台控制权限, desc:true为开启,false为关闭-->true
</ptzControl>
</ptzChannelPermission>
</ptzChannelPermissionList>
<logOrStateCheck>
<!--ro, opt, bool, 是否开启日志和设备状态查看权限-->true
</logOrStateCheck>
<parameterConfig>
<!--ro, opt, bool, 是否开启设备远程参数设置权限-->true
</parameterConfig>
<restartOrShutdown>
<!--ro, opt, bool, 是否开启重启或关机权限-->true
</restartOrShutdown>
<upgrade>
<!--ro, opt, bool, 是否开启升级权限-->true
</upgrade>
<upgradeNotShow>
<!--ro, opt, bool, 升级不展示, desc:为确保兼容, 设备端admin,installer用户升级权限也支持, 该字段返回true的话, 上层界面隐藏admin和installer升级权限的
相关界面, 前后端暂没有installer,admin等用户,EN认证过程用web进行操作要求升级不能和installer和admin进行绑定, 手机app也会用到该协议, 设备如果不返回升级权限字段
那么手机app就没有升级功, 因此增加该字段-->true
</upgradeNotShow>
<voiceTalk>
<!--ro, opt, bool, 是否开启语音对讲(所有通道)权限-->true
</voiceTalk>
<transParentChannel>
<!--ro, opt, bool, 是否开启透明通道(所有通道)权限, desc:也即控制串口-->true
</transParentChannel>
<contorlLocalOut>
<!--ro, opt, bool, 是否开启控制本地输出权限-->true
</contorlLocalOut>
<manageChannel>
<!--ro, opt, bool, 是否开启通道管理权限-->true
</manageChannel>
<alarmOutOrUpload>
<!--ro, opt, bool, 是否开启报警上报和输出权限-->true
</alarmOutOrUpload>
<IRAID>
<!--ro, opt, object, Iraid快速权限配置-->
<rapidConfiguration>
<!--ro, opt, bool, 是否开启Iraid快速配置权限-->true
</rapidConfiguration>
<reset>
<!--ro, opt, bool, 是否开启Iraid重设配置权限-->true
</reset>
</IRAID>
<LEDConfig>
<!--ro, opt, bool, 是否允许用户修改LED配置-->true
</LEDConfig>
<AIModelManagement>
<!--ro, opt, bool, 是否开启AI模型包操作权限-->true
</AIModelManagement>
<AITaskManagement>
<!--ro, opt, bool, 是否开启AI任务管理操作权限-->true
</AITaskManagement>
<materialReview>
<!--ro, opt, bool, 素材审核权限, desc:信息发布-->true
</materialReview>
<programReview>
<!--ro, opt, bool, 节目审核权限, desc:信息发布-->true
</programReview>
<uploadMaterial>
<!--ro, opt, bool, 上传素材, desc:信息发布-->true
</uploadMaterial>
<addProgram>
<!--ro, opt, bool, 节目编辑权限, desc:增, 删, 改-->true
</addProgram>
<addSchedule>
<!--ro, opt, bool, 日程编辑权限, desc:增, 删, 改-->true
</addSchedule>
<programPlay>
<!--ro, opt, bool, 信息播放权限-->true
</programPlay>
<terminalMgr>
<!--ro, opt, bool, 终端管理权限, desc:对终端的背光, Logo灯等参数设置-->true
</terminalMgr>
<shopSmart>
<!--ro, opt, bool, 商业智能权限-->true
</shopSmart>
<materialReplace>
<!--ro, opt, bool, 素材替换权限-->true
</materialReplace>
<scheduleRelease>
<!--ro, opt, bool, 日程发布权限-->true
</scheduleRelease>
<scheduleReview>
<!--ro, opt, bool, 日程审核-->true
</scheduleReview>
<addTerminal>
<!--ro, opt, bool, 终端添加-->true
```

```

</addTerminal>
<terminalDispatch>
  <!--ro, opt, bool, 终端分配权限, desc:将该组织下的终端分配给子组织-->true
</terminalDispatch>
<addUser>
  <!--ro, opt, bool, 用户编辑权限, desc:增, 删, 改-->true
</addUser>
<addOrg>
  <!--ro, opt, bool, 组织编辑权限, desc:增, 删, 改-->true
</addOrg>
<logMgr>
  <!--ro, opt, bool, 日志管理-->true
</logMgr>
<dataUpdate>
  <!--ro, opt, bool, 数据更新, desc:叫号数据-->true
</dataUpdate>
<insertMessage>
  <!--ro, opt, bool, 插播消息权限-->true
</insertMessage>
<notReview>
  <!--ro, opt, bool, 无需审核权限-->true
</notReview>
<MgrTerminal>
  <!--ro, opt, object, 管理的终端或终端组-->
  <TerminalNoList>
    <!--ro, opt, object, 终端号列表-->
    <terminalNo>
      <!--ro, opt, int, 终端号, range:[1,10]-->1
    </terminalNo>
  </TerminalNoList>
  <TerminalGroupNoList>
    <!--ro, opt, object, 终端组列表-->
    <terminalGroupNo>
      <!--ro, opt, int, 终端组号, range:[1,10]-->1
    </terminalGroupNo>
  </TerminalGroupNoList>
</MgrTerminal>
<subSysOrZoneArm>
  <!--ro, opt, bool, 子系统/防区布防权限-->true
</subSysOrZoneArm>
<subSysOrZoneDisarm>
  <!--ro, opt, bool, 子系统/防区撤防权限-->true
</subSysOrZoneDisarm>
<subSysOrZoneClearArm>
  <!--ro, opt, bool, 子系统/防区消警权限-->true
</subSysOrZoneClearArm>
<zoneBypass>
  <!--ro, opt, bool, 防区旁路权限-->true
</zoneBypass>
<zoneBypassRecover>
  <!--ro, opt, bool, 防区旁路恢复权限-->true
</zoneBypassRecover>
<operateOutput>
  <!--ro, opt, bool, 操作继电器权限-->true
</operateOutput>
<subSystemList>
  <!--ro, opt, object, 子系统列表, desc:当用户添加或者配置operator时可关联子系统-->
  <subSystem>
    <!--ro, opt, int, 子系统, range:[1,64]-->1
  </subSystem>
</subSystemList>
<factoryReset>
  <!--ro, opt, bool, 恢复默认参数-->true
</factoryReset>
<arm>
  <!--ro, opt, bool, 布防-->true
</arm>
<disarm>
  <!--ro, opt, bool, 撤防-->true
</disarm>
<LPListAudit>
  <!--ro, opt, bool, 黑白名单操作权限,

```

desc:有这个权限的用户可以导入/导出/编辑/删除/查询黑白名单
对应接口如下:

车牌黑白名单查询: /ISAPI/Traffic/channels/<ID>/searchLPListAudit

导入/导出车牌黑白名单数据: /ISAPI/Traffic/channels/<ID>/LicensePlateAuditData

车牌名单添加修改: /ISAPI/Traffic/channels/<channelID>/LicensePlateAuditData/record?format=json

车牌黑白名单数据删除: /ISAPI/Traffic/channels/<channelID>/LicensePlateAuditData/delete?format=json-->true

```

</LPListAudit>
<applicationWarehouse>
  <!--ro, opt, bool, 应用仓库管理权限, desc:不支持不返回。具备管理权限, 用户可进行导入、删除、启动HEOP APP等操作-->true
</applicationWarehouse>
<subSystemManagement>
  <!--ro, opt, bool, 子系统管理权限, desc:包含子系统管理相关的所有权限-->true
</subSystemManagement>
<userOfModifyKeypadPasswordList>
  <!--ro, opt, array, 允许修改键盘密码的用户列表, subType:object-->
  <userOfModifyKeypadPassword>
    <!--ro, opt, object, 允许修改键盘密码的用户-->
    <subUserType>
      <!--ro, opt, enum, 子用户类型, subType:string, [cCloudAdmin#云端admin, LocalAdmin#本地admin, installerAdmin#安装管理员, installerEmployee#安装雇员, LocalInstaller#本地installer, CloudOperator#云端操作员, LocalOperator#本地操作员]-->cCloudAdmin
    </subUserType>
  </userOfModifyKeypadPassword>
</userOfModifyKeypadPasswordList>

```

```
</userModifyKeypadPassword>
</userOfModifyKeypadPasswordList>
<userOfModifyUserPermissionList>
  <!--ro, opt, array, 允许修改用户权限的用户列表, subType:object-->
  <userOfModifyUserPermission>
    <!--ro, opt, object, 允许修改用户权限的用户-->
    <subUserType>
      <!--ro, opt, enum, 子用户类型, subType:string, [cCloudAdmin#云端admin, LocalAdmin#本地admin, installerAdmin#安装管理员, installEmployee#安装雇员, LocalInstaller#本地installer, cCloudOperator#云端操作员, LocalOperator#本地操作员]-->cCloudAdmin
    </subUserType>
    </userOfModifyUserPermission>
  </userOfModifyUserPermissionList>
  <falseAlarmMaterialLibMgr>
    <!--ro, opt, bool, 误报分析素材库管理权限-->true
  </falseAlarmMaterialLibMgr>
  <TIPConfig>
    <!--ro, opt, bool, TIP配置权限-->true
  </TIPConfig>
  <networkCfg>
    <!--ro, opt, bool, 网络配置权限, desc:包括无线和有线网络-->true
  </networkCfg>
  <ISUPAccess>
    <!--ro, opt, bool, ISUP接入配置权限, desc:指设备接入ISUP的权限配置, 对应ISUP接入领域-->true
  </ISUPAccess>
  <timeMgr>
    <!--ro, opt, bool, 时间管理权限, desc:指系统时间和时区管理的权限, 对应时间管理领域-->true
  </timeMgr>
  <HDDMgr>
    <!--ro, opt, bool, 硬盘管理权限, desc:指硬盘管理的权限, 对应硬盘管理领域-->true
  </HDDMgr>
  </remotePermission>
  <passwordValidity>
    <!--ro, opt, int, 密码有效期, range:[0,365], desc:仅管理员用户可以修改, 单位: 天, [0, 365]-->1
  </passwordValidity>
  <subUserType>
    <!--ro, opt, enum, 子用户类型, subType:string, [cCloudAdmin#云端admin, LocalAdmin#本地admin, installerAdmin#安装管理员, installEmployee#安装雇员, LocalInstaller#本地installer, cCloudOperator#云端操作员, LocalOperator#本地操作员]-->cCloudAdmin
  </subUserType>
  </UserPermission>
</UserPermissionList>
```

69.2.16 配置单个用户权限

Request URL

PUT /ISAPI/Security/UserPermission/<indexID>

查询参数

参数名称	参数类型	描述
indexID	string	

请求报文

```
<?xml version="1.0" encoding="UTF-8"?>

<UserPermission xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, 用户权限, attr:version{req, string, 协议版本}-->
  <id>
    <!--req, string, 索引值, range:[1,32]-->test
  </id>
  <userID>
    <!--req, string, 用户ID, range:[1,32]-->test
  </userID>
  <userType>
    <!--req, enum, 用户类型, subType:string, [admin#系统用户, operator#操作员, viewer#观察者, installer#安装员, manufacturer#制造者, single#单用户]-->admin
  </userType>
  <localPermission>
    <!--opt, object, 本地权限节点-->
    <backup>
      <!--opt, bool, 是否开启备份(所有通道)权限, desc:true为开启, false为关闭-->true
    </backup>
    <record>
      <!--opt, bool, 是否开启录像(所有通道)权限, desc:true为开启, false为关闭-->true
    </record>
    <playBack>
      <!--opt, bool, 是否开启回放(所有通道)权限, desc:true为开启, false为关闭-->true
    </playBack>
    <preview>
      <!--opt, bool, 是否开启预览(所有通道)权限, desc:true为开启, false为关闭-->true
    </preview>
    <videoChannelPermissionList>
      <!--opt, object, 视频通道权限配置列表-->
      <videoChannelPermission>
        <!--opt, object, 单一视频通道权限配置-->
        <id>
```

```
<!--req, string, 视频通道ID, range:[1,32]-->test
</id>
<playBack>
  <!--opt, bool, 是否开启当前通道回放权限, desc:true为开启,false为关闭-->true
</playBack>
<preview>
  <!--opt, bool, 是否开启当前通道预览权限, desc:true为开启,false为关闭-->true
</preview>
<record>
  <!--opt, bool, 是否开启当前通道录像权限, desc:true为开启,false为关闭-->true
</record>
<backup>
  <!--opt, bool, 是否开启当前通道备份权限, desc:true为开启,false为关闭-->true
</backup>
<playBackDoubleVerification>
  <!--opt, bool, 是否开启当前通道回放二次认证权限, desc:true为开启,false为关闭-->true
</playBackDoubleVerification>
<backupDoubleVerification>
  <!--opt, bool, 是否开启当前通道备份二次认证权限, desc:true为开启,false为关闭-->true
</backupDoubleVerification>
</videoChannelPermission>
</videoChannelPermissionList>
<ptzControl>
  <!--req, bool, 是否开启云台控制(所有通道)权限, desc:true为开启,false为关闭-->true
</ptzControl>
<ptzChannelPermissionList>
  <!--opt, object, 视频通道云台控制权限列表-->
  <ptzChannelPermission>
    <!--req, object, 单一视频通道云台控制权限-->
    <id>
      <!--req, string, 视频通道ID, range:[1,32]-->test
    </id>
    <ptzControl>
      <!--opt, bool, 是否开启当前通道云台控制权限, desc:true为开启,false为关闭-->true
    </ptzControl>
  </ptzChannelPermission>
</ptzChannelPermissionList>
<logOrStateCheck>
  <!--opt, bool, 是否开启日志和设备状态查看权限, desc:true为开启,false为关闭-->true
</logOrStateCheck>
<parameterConfig>
  <!--opt, bool, 是否开启设备远程参数设置权限, desc:true为开启,false为关闭-->true
</parameterConfig>
<restartOrShutdown>
  <!--opt, bool, 是否开启重启或关机权限, desc:true为开启,false为关闭-->true
</restartOrShutdown>
<upgrade>
  <!--opt, bool, 是否开启升级权限, desc:true为开启,false为关闭-->true
</upgrade>
<upgradeNotShow>
  <!--opt, bool, 升级不展示, desc:为确保兼容, 设备端admin, installer用户升级权限也支持, 该字段返回true的话, 上层界面隐藏admin和installer升级权限的相关界面, 前后端曾没有installer, admin等用户, 认证过程用web进行操作要求升级不能和installer和admin进行绑定, 手机app也会用到该协议, 设备如果不返回升级权限字段那么手机app就没有升级功-->true
</upgradeNotShow>
<manageChannel>
  <!--opt, bool, 是否开启通道管理权限, desc:true为开启,false为关闭-->true
</manageChannel>
<AIModelManagement>
  <!--opt, bool, 是否开启AI模型包操作权限, desc:true为开启,false为关闭-->true
</AIModelManagement>
<AITaskManagement>
  <!--opt, bool, 是否开启AI任务管理操作权限, desc:true为开启,false为关闭-->true
</AITaskManagement>
<subSysOrZoneArm>
  <!--opt, bool, 子系统/防区布防权限-->true
</subSysOrZoneArm>
<subSysOrZoneDisarm>
  <!--opt, bool, 子系统/防区撤防权限-->true
</subSysOrZoneDisarm>
<operateOutput>
  <!--opt, bool, 操作继电器权限-->true
</operateOutput>
<supportLinkageSubSystemList>
  <!--opt, object, 已使能子系统列表-->
  <subSystem>
    <!--opt, int, 已使能子系统, range:[1,64]-->1
  </subSystem>
</supportLinkageSubSystemList>
<subSystemList>
  <!--opt, object, 子系统列表, desc:当用户添加或者配置operator时可关联-->
  <subSystem>
    <!--opt, int, 子系统, range:[1,64]-->1
  </subSystem>
</subSystemList>
<localFileMgr>
  <!--opt, bool, 本地文件管理权限, desc:针对本地文件的增删改查权限-->true
</localFileMgr>
<LEDConfig>
  <!--opt, bool, 是否允许用户修改LED配置-->true
</LEDConfig>
<logMgr>
  <!--opt, bool, 日志管理权限-->true
</logMgr>
<TIPConfie>
```

```
<!--opt, bool, TIP配置权限-->true
</TIPConfig>
<securityFunctionCfg>
  <!--opt, bool, 安检功能配置权限, desc: 针对安检类目的权限配置-->true
</securityFunctionCfg>
<networkCfg>
  <!--opt, bool, 网络配置权限, desc: 包括无线和有线网络-->true
</networkCfg>
<ISUPAccess>
  <!--opt, bool, ISUP接入配置权限, desc: 指设备接入ISUP的权限配置, 对应ISUP接入领域-->true
</ISUPAccess>
<timeMgr>
  <!--opt, bool, 时间管理权限, desc: 指系统时间和时区管理的权限, 对应时间管理领域-->true
</timeMgr>
<HDDMgr>
  <!--opt, bool, 硬盘管理权限, desc: 指硬盘管理的权限, 对应硬盘管理领域-->true
</HDDMgr>
<securitySystemMaintenance>
  <!--opt, bool, 安检机系统维护权限, desc: 包含探测器测试、射线源测试、操作键盘测试、系统诊断、维护记录、其他检测（主控板、IO、射线源灯）-->true
</securitySystemMaintenance>
<securityCheckingPicExport>
  <!--opt, bool, 安检过包检测图片导出权限-->true
</securityCheckingPicExport>
<securityCheckingPicDelete>
  <!--opt, bool, 安检过包检测图片删除权限-->true
</securityCheckingPicDelete>
<operationScheduleList>
  <!--opt, array, 操作计划列表, subType:object, dep:or,{$.UserPermissionList.UserPermission.LocalPermission.operationScheduleEnabled,eq,true}-->
<operationSchedule>
  <!--opt, object, 操作计划-->
  <operationScheduleEnabled>
    <!--opt, bool, 操作计划使能-->true
  </operationScheduleEnabled>
  <operationDetailItem>
    <!--opt, enum, 操作具体项, subType:string, [all#所有操作,arm#布防,disarm#撤防,operateOutput#继电器操作]-->arm
  </operationDetailItem>
  <timeRange>
    <!--opt, object, 时间范围-->
    <beginTime>
      <!--opt, time, 开始时间-->00:00:00
    </beginTime>
    <endTime>
      <!--opt, time, 结束时间-->00:00:00
    </endTime>
  </timeRange>
</operationSchedule>
</operationScheduleList>
</localPermission>
<remotePermission>
  <!--opt, object, 远程权限节点-->
  <record>
    <!--opt, bool, 是否开启录像（所有通道）权限, desc:true为开启,false为关闭-->true
  </record>
  <playBack>
    <!--opt, bool, 是否开启回放（所有通道）权限, desc:true为开启,false为关闭-->true
  </playBack>
  <preview>
    <!--opt, bool, 是否开启预览（所有通道）权限, desc:true为开启,false为关闭-->true
  </preview>
  <videoChannelPermissionList>
    <!--opt, object, 视频通道权限配置列表-->
    <videoChannelPermission>
      <!--opt, object, 单一视频通道权限配置-->
      <id>
        <!--req, string, 对应视频通道ID, range:[1,32]-->test
      </id>
      <preview>
        <!--opt, bool, 是否开启当前通道预览权限, desc:true为开启,false为关闭-->true
      </preview>
      <playBack>
        <!--opt, bool, 是否开启当前通道回放权限, desc:true为开启,false为关闭-->true
      </playBack>
      <record>
        <!--opt, bool, 是否开启当前通道录像权限, desc:true为开启,false为关闭-->true
      </record>
      <playBackDoubleVerification>
        <!--opt, bool, 是否开启当前通道回放二次认证权限, desc:true为开启,false为关闭-->true
      </playBackDoubleVerification>
    </videoChannelPermission>
  </videoChannelPermissionList>
  <ptzControl>
    <!--opt, bool, 是否开启云台控制（所有通道）权限, desc:true为开启,false为关闭-->true
  </ptzControl>
  <ptzChannelPermissionList>
    <!--opt, object, 视频通道云台控制权限列表-->
    <ptzChannelPermission>
      <!--opt, object, 单一视频通道云台控制权限-->
      <id>
        <!--req, string, 视频通道ID, range:[1,32], desc:/ISAPI/PTZCtrl/channels/ID-->test
      </id>
      <ptzControl>
        <!--opt, bool, 是否开启当前通道云台控制权限, desc:true为开启,false为关闭-->true
      </ptzControl>
```

```
</ptzChannelPermission>
</ptzChannelPermissionList>
<logOrStateCheck>
  <!--opt, bool, 是否开启日志和设备状态查看权限-->true
</logOrStateCheck>
<parameterConfig>
  <!--opt, bool, 是否开启设备远程参数设置权限-->true
</parameterConfig>
<restartOrShutdown>
  <!--opt, bool, 是否开启重启或关机权限-->true
</restartOrShutdown>
<upgrade>
  <!--opt, bool, 是否开启升级权限-->true
</upgrade>
<upgradeNotShow>
  <!--opt, bool, 升级不展示, desc:为确保兼容, 设备端admin, installer用户升级权限也支持, 该字段返回true的话, 上层界面隐藏admin和installer升级权限的相关界面, 前后端暂时没有installer, admin等用户, EN认证过程用web进行操作要求升级不能和installer和admin进行绑定, 手机app也会用到该协议, 设备如果不返回升级权限字段那么手机app就没有升级功-->true
</upgradeNotShow>
<voiceTalk>
  <!--opt, bool, 是否开启语音对讲(所有通道)权限-->true
</voiceTalk>
<transParentChannel>
  <!--opt, bool, 是否开启透明通道(所有通道)权限, desc:也即控制串口-->true
</transParentChannel>
<contorlLocalOut>
  <!--opt, bool, 是否开启控制本地输出权限-->true
</contorlLocalOut>
<manageChannel>
  <!--opt, bool, 是否开启通道管理权限-->true
</manageChannel>
<alarmOutOrUpload>
  <!--opt, bool, 是否开启报警上报和输出权限-->true
</alarmOutOrUpload>
<IRAID>
  <!--opt, object, Iraid快速权限配置-->
  <rapidConfiguration>
    <!--opt, bool, 是否开启Iraid快速配置权限-->true
  </rapidConfiguration>
  <reset>
    <!--opt, bool, 是否开启Iraid重设配置权限-->true
  </reset>
</IRAID>
<LEDConfig>
  <!--opt, bool, 是否允许用户修改LED配置-->true
</LEDConfig>
<AIModelManagement>
  <!--opt, bool, 是否开启AI模型包操作权限-->true
</AIModelManagement>
<AITaskManagement>
  <!--opt, bool, 是否开启AI任务管理操作权限-->true
</AITaskManagement>
<materialReview>
  <!--opt, bool, 素材审核权限, desc:信息发布-->true
</materialReview>
<programReview>
  <!--opt, bool, 节目审核权限, desc:信息发布-->true
</programReview>
<uploadMaterial>
  <!--opt, bool, 上传素材, desc:信息发布-->true
</uploadMaterial>
<addProgram>
  <!--opt, bool, 节目编辑权限, desc:增, 删, 改-->true
</addProgram>
<addSchedule>
  <!--opt, bool, 日程编辑权限, desc:增, 删, 改-->true
</addSchedule>
<programPlay>
  <!--opt, bool, 信息播放权限-->true
</programPlay>
<terminalMgr>
  <!--opt, bool, 终端管理权限, desc:对终端的背光, Logo灯等参数设置-->true
</terminalMgr>
<shopSmart>
  <!--opt, bool, 商业智能权限-->true
</shopSmart>
<materialReplace>
  <!--opt, bool, 素材替换权限-->true
</materialReplace>
<scheduleRelease>
  <!--opt, bool, 日程发布权限-->true
</scheduleRelease>
<scheduleReview>
  <!--opt, bool, 日程审核-->true
</scheduleReview>
<addTerminal>
  <!--opt, bool, 终端添加-->true
</addTerminal>
<terminalDispatch>
  <!--opt, bool, 终端分配权限, desc:将该组织下的终端分配给子组织-->true
</terminalDispatch>
<addUser>
  <!--opt, bool, 用户编辑权限, desc:增, 删, 改-->true
</addUser>
```

```

</addOrg>
<!--opt, bool, 组织编辑权限, desc:增, 删, 改-->true
</addOrg>
<logMgr>
<!--opt, bool, 日志管理-->true
</logMgr>
<dataUpdate>
<!--opt, bool, 数据更新, desc:叫号数据-->true
</dataUpdate>
<insertMessage>
<!--opt, bool, 插播消息权限-->true
</insertMessage>
<notReview>
<!--opt, bool, 无需审核权限-->true
</notReview>
<MgrTerminal>
<!--opt, object, 管理的终端或终端组-->
<TerminalNoList size="10">
<!--opt, object, 终端号列表, attr:size{req, int, 数组元素总数}-->
<terminalNo>
<!--opt, int, 终端号, range:[1,10]-->1
</terminalNo>
</TerminalNoList>
<TerminalGroupNoList size="10">
<!--opt, object, 终端组列表, attr:size{req, int, 数组元素总数}-->
<terminalGroupNo>
<!--opt, int, 终端组号, range:[1,10]-->1
</terminalGroupNo>
</TerminalGroupNoList>
</MgrTerminal>
<subSysOrZoneArm>
<!--opt, bool, 子系统/防区布防权限-->true
</subSysOrZoneArm>
<subSysOrZoneDisarm>
<!--opt, bool, 子系统/防区撤防权限-->true
</subSysOrZoneDisarm>
<subSysOrZoneClearArm>
<!--opt, bool, 子系统/防区消警权限-->true
</subSysOrZoneClearArm>
<zoneBypass>
<!--opt, bool, 防区旁路权限-->true
</zoneBypass>
<zoneBypassRecover>
<!--opt, bool, 防区旁路恢复权限-->true
</zoneBypassRecover>
<operateOutput>
<!--opt, bool, 操作继电器权限-->true
</operateOutput>
<supportLinkageSubSystemList>
<!--opt, object, 已使能子系统列表-->
<subSystem>
<!--opt, int, 已使能子系统, range:[1,64]-->1
</subSystem>
</supportLinkageSubSystemList>
<subSystemList>
<!--opt, object, 子系统列表, desc:当用户添加或者配置operator时可关联子系统-->
<subSystem>
<!--opt, int, 子系统, range:[1,64]-->1
</subSystem>
</subSystemList>
<factoryReset>
<!--opt, bool, 恢复默认参数-->true
</factoryReset>
<arm>
<!--opt, bool, 布防-->true
</arm>
<disarm>
<!--opt, bool, 撤防-->true
</disarm>
<accessControl>
<!--opt, bool, 门禁控制-->true
</accessControl>
<restoreTamper>
<!--opt, bool, 恢复防拆配置, desc:使用描述见: /ISAPI/Security/UserPermission/capabilities-->true
</restoreTamper>
<restoreConfirmedAlarm>
<!--opt, bool, 恢复确认报警配置, desc:使用描述见: /ISAPI/Security/UserPermission/capabilities-->true
</restoreConfirmedAlarm>
<LPListAudit>
<!--opt, bool, 黑白名单操作权限,
desc:有这个权限的用户可以导入/导出/编辑/删除/查询黑白名单
对应接口如下:
车牌黑白名单查询:/ISAPI/Traffic/channels/<ID>/searchLPListAudit
导入/导出车牌黑白名单数据:/ISAPI/Traffic/channels/<ID>/LicensePlateAuditData
车牌名单添加修改/ISAPI/Traffic/channels/<channelID>/LicensePlateAuditData/record?format=json
车牌黑白名单数据删除: /ISAPI/Traffic/channels/<channelID>/LicensePlateAuditData/delete?format=json-->true
</LPListAudit>
<applicationWarehouse>
<!--opt, bool, 应用仓库管理权限, desc:不支持不返回。具备管理权限, 用户可进行导入、删除、启动HEOP APP等操作-->true
</applicationWarehouse>
<subSystemManagement>
<!--opt, bool, 子系统管理权限, desc:包含子系统管理相关的所有权限-->true
</subSystemManagement>

```

```

<userOfModifyKeypadPasswordList>
  <!--opt, array, 允许修改键盘密码的用户列表, subType:object-->
  <userOfModifyKeypadPassword>
    <!--opt, object, 允许修改键盘密码的用户-->
    <subUserType>
      <!--opt, enum, 子用户类型, subType:string, [cCloudAdmin#云端admin, LocalAdmin#本地admin, installerAdmin#安装管理员, installEmployee#安装雇员, LocalInstaller#本地installer, cCloudOperator#云端操作员, LocalOperator#本地操作员]-->cCloudAdmin
    </subUserType>
  </userOfModifyKeypadPassword>
</userOfModifyKeypadPasswordList>
<userOfModifyUserPermissionList>
  <!--opt, array, 允许修改用户权限的用户列表, subType:object-->
  <userOfModifyUserPermission>
    <!--opt, object, 允许修改用户权限的用户-->
    <subUserType>
      <!--opt, enum, 子用户类型, subType:string, [cCloudAdmin#云端admin, LocalAdmin#本地admin, installerAdmin#安装管理员, installEmployee#安装雇员, LocalInstaller#本地installer, cCloudOperator#云端操作员, LocalOperator#本地操作员]-->cCloudAdmin
    </subUserType>
  </userOfModifyUserPermission>
</userOfModifyUserPermissionList>
<falseAlarmMaterialLibMgr>
  <!--opt, bool, 误报分析素材库管理权限-->true
</falseAlarmMaterialLibMgr>
<TIPConfig>
  <!--opt, bool, TIP配置权限-->true
</TIPConfig>
<networkCfg>
  <!--opt, bool, 网络配置权限, desc:包括无线和有线网络-->true
</networkCfg>
<ISUPAccess>
  <!--opt, bool, ISUP接入配置权限, desc:指设备接入ISUP的权限配置, 对应ISUP接入领域-->true
</ISUPAccess>
<timeMgr>
  <!--opt, bool, 时间管理权限, desc:指系统时间和时区管理的权限, 对应时间管理领域-->true
</timeMgr>
<HDDMgr>
  <!--opt, bool, 硬盘管理权限, desc:指硬盘管理的权限, 对应硬盘管理领域-->true
</HDDMgr>
</remotePermission>
<passwordValidity>
  <!--opt, int, 密码有效期, range:[0,365], desc:仅管理员用户可以修改, 单位: 天, [0, 365]-->1
</passwordValidity>
<subUserType>
  <!--opt, enum, 子用户类型, subType:string, [cCloudAdmin#云端admin, LocalAdmin#本地admin, installerAdmin#安装管理员, installEmployee#安装雇员, LocalInstaller#本地installer, cCloudOperator#云端操作员, LocalOperator#本地操作员]-->cCloudAdmin
</subUserType>
</UserPermission>

```

响应报文

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 响应消息, attr:version{ro, req, string, 版本}-->
  <requestURL>
    <!--ro, req, string, 请求的URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, 状态码, subType:int, [0#OK,1#OK,2#Device Busy,3#Device Error,4#Invalid Operation,5#Invalid XML Format,6#Invalid XML Content,7#Reboot Required]-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, 状态信息, subType:string, [OK#成功,Device Busy#设备忙,Device Error#设备异常,Invalid Operation#无效的操作,Invalid XML Format#无效的XML格式,Invalid XML Content#无效的XML内容,Reboot#设备重启]-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, 详细错误码英文描述, desc:详细错误码的英文描述-->OK
  </subStatusCode>
</ResponseStatus>

```

69.2.17 获取单个用户权限

Request URL

GET /ISAPI/Security/UserPermission/<indexID>

查询参数

参数名称	参数类型	描述
indexID	string	

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<UserPermission xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 用户权限, attr:version{req, string, 协议版本}-->
  <id>
    <!--ro, req, string, 索引值, range:[1,32]-->test
  </id>
  <userID>
    <!--ro, req, string, 用户ID, range:[1,32]-->test
  </userID>
  <userType>
    <!--ro, req, enum, 用户类型, subType:string, [admin#系统用户,operator#操作员,viewer#观察员,installer#安装员,manufacturer#制造者,single#单用户]-->admin
  </userType>
  <localPermission>
    <!--ro, opt, object, 本地权限节点-->
    <backup>
      <!--ro, opt, bool, 是否开启备份(所有通道)权限, desc:true为开启,false为关闭-->true
    </backup>
    <record>
      <!--ro, opt, bool, 是否开启录像(所有通道)权限, desc:true为开启,false为关闭-->true
    </record>
    <playBack>
      <!--ro, opt, bool, 是否开启回放(所有通道)权限, desc:true为开启,false为关闭-->true
    </playBack>
    <preview>
      <!--ro, opt, bool, 是否开启预览(所有通道)权限, desc:true为开启,false为关闭-->true
    </preview>
    <videoChannelPermissionList>
      <!--ro, opt, object, 视频通道权限配置列表-->
      <videoChannelPermission>
        <!--ro, opt, object, 单一视频通道权限配置-->
        <id>
          <!--ro, req, string, 视频通道ID, range:[1,32]-->test
        </id>
        <playBack>
          <!--ro, opt, bool, 是否开启当前通道回放权限, desc:true为开启,false为关闭-->true
        </playBack>
        <preview>
          <!--ro, opt, bool, 是否开启当前通道预览权限, desc:true为开启,false为关闭-->true
        </preview>
        <record>
          <!--ro, opt, bool, 是否开启当前通道录像权限, desc:true为开启,false为关闭-->true
        </record>
        <backup>
          <!--ro, opt, bool, 是否开启当前通道备份权限, desc:true为开启,false为关闭-->true
        </backup>
        <playBackDoubleVerification>
          <!--ro, opt, bool, 是否开启当前通道回放二次认证权限, desc:true为开启,false为关闭-->true
        </playBackDoubleVerification>
        <backupDoubleVerification>
          <!--ro, opt, bool, 是否开启当前通道备份二次认证权限, desc:true为开启,false为关闭-->true
        </backupDoubleVerification>
      </videoChannelPermission>
    </videoChannelPermissionList>
    <ptzControl>
      <!--ro, req, bool, 是否开启云台控制(所有通道)权限, desc:true为开启,false为关闭-->true
    </ptzControl>
    <ptzChannelPermissionList>
      <!--ro, opt, object, 视频通道云台控制权限列表-->
      <ptzChannelPermission>
        <!--ro, req, object, 单一视频通道云台控制权限-->
        <id>
          <!--ro, req, string, 视频通道ID, range:[1,32]-->test
        </id>
        <ptzControl>
          <!--ro, opt, bool, 是否开启当前通道云台控制权限, desc:true为开启,false为关闭-->true
        </ptzControl>
      </ptzChannelPermission>
    </ptzChannelPermissionList>
    <logOrStateCheck>
      <!--ro, opt, bool, 是否开启日志和设备状态查看权限, desc:true为开启,false为关闭-->true
    </logOrStateCheck>
    <parameterConfig>
      <!--ro, opt, bool, 是否开启设备远程参数设置权限, desc:true为开启,false为关闭-->true
    </parameterConfig>
    <restartOrShutdown>
      <!--ro, opt, bool, 是否开启重启或关机权限, desc:true为开启,false为关闭-->true
    </restartOrShutdown>
    <upgrade>
      <!--ro, opt, bool, 是否开启升级权限, desc:true为开启,false为关闭-->true
    </upgrade>
    <upgradeNotShow>
      <!--ro, opt, bool, 升级不展示, desc:为确保兼容, 设备端admin,installer用户升级权限也支持, 该字段返回true的话, 上层界面隐藏admin和installer升级权限的相关界面, 前后端暂时没有installer,admin等用户,EN认证过程用web进行操作要求升级不能和installer和admin进行绑定, 手机app也会用到该协议, 设备如果不返回升级权限字段那么手机app就没有升级功-->true
    </upgradeNotShow>
    <manageChannel>
      <!--ro, opt, bool, 是否开启通道管理权限, desc:true为开启,false为关闭-->true
    </manageChannel>
  </localPermission>

```

```
</manageChannel>
<AIModelManagement>
  <!--ro, opt, bool, 是否开启AI模型包操作权限, desc:true为开启,false为关闭-->true
</AIModelManagement>
<AITaskManagement>
  <!--ro, opt, bool, 是否开启AI任务管理操作权限, desc:true为开启,false为关闭-->true
</AITaskManagement>
<subSysOrZoneArm>
  <!--ro, opt, bool, 子系统/防区布防权限-->true
</subSysOrZoneArm>
<subSysOrZoneDisarm>
  <!--ro, opt, bool, 子系统/防区撤防权限-->true
</subSysOrZoneDisarm>
<operateOutput>
  <!--ro, opt, bool, 操作继电器权限-->true
</operateOutput>
<supportLinkageSubSystemList>
  <!--ro, opt, object, 已使能子系统列表-->
  <subSystem>
    <!--ro, opt, int, 已使能子系统, range:[1,64]-->1
  </subSystem>
</supportLinkageSubSystemList>
<subSystemList>
  <!--ro, opt, object, 子系统列表, desc:当用户添加或者配置operator时可关联-->
  <subSystem>
    <!--ro, opt, int, 子系统, range:[1,64]-->1
  </subSystem>
</subSystemList>
<localFileMgr>
  <!--ro, opt, bool, 本地文件管理权限, desc:指对本地文件的增删改查权限-->true
</localFileMgr>
<LEDConfig>
  <!--ro, opt, bool, 是否允许用户修改LED配置-->true
</LEDConfig>
<logMgr>
  <!--ro, opt, bool, 日志管理权限-->true
</logMgr>
<TIPConfig>
  <!--ro, opt, bool, TIP配置权限-->true
</TIPConfig>
<securityFunctionCfg>
  <!--ro, opt, bool, 安检功能配置权限, desc:针对安检类目的权限配置-->true
</securityFunctionCfg>
<networkCfg>
  <!--ro, opt, bool, 网络配置权限, desc:包括无线和有线网络-->true
</networkCfg>
<ISUPAccess>
  <!--ro, opt, bool, ISUP接入配置权限, desc:指设备接入ISUP的权限配置, 对应ISUP接入领域-->true
</ISUPAccess>
<timeMgr>
  <!--ro, opt, bool, 时间管理权限, desc:指系统时间和时区管理的权限, 对应时间管理领域-->true
</timeMgr>
<HDDMgr>
  <!--ro, opt, bool, 硬盘管理权限, desc:指硬盘管理的权限, 对应硬盘管理领域-->true
</HDDMgr>
<securitySystemMaintenance>
  <!--ro, opt, bool, 安检机系统维护权限, desc:包含探测器测试、射线源测试、操作键盘测试、系统诊断、维护记录、其他检测（主控板、IO、射线源灯）-->true
</securitySystemMaintenance>
<securityCheckingPicExport>
  <!--ro, opt, bool, 安检过包检测图片导出权限-->true
</securityCheckingPicExport>
<securityCheckingPicDelete>
  <!--ro, opt, bool, 安检过包检测图片删除权限-->true
</securityCheckingPicDelete>
<operationScheduleList>
  <!--ro, opt, array, 操作计划列表, subType:object, dep:or, [{UserPermissionList.UserPermission.LocalPermission.operationScheduleEnabled,eq,true}]-->
  <operationSchedule>
    <!--ro, opt, object, 操作计划-->
    <operationScheduleEnabled>
      <!--ro, opt, bool, 操作计划使能-->true
    </operationScheduleEnabled>
    <operationDetailItem>
      <!--ro, opt, enum, 操作具体项, subType:string, [all#所有操作,arm#布防,disarm#撤防,operateOutput#继电器操作]-->arm
    </operationDetailItem>
    <timeRange>
      <!--ro, opt, object, 时间范围-->
      <beginTime>
        <!--ro, opt, time, 开始时间-->00:00:00
      </beginTime>
      <endTime>
        <!--ro, opt, time, 结束时间-->00:00:00
      </endTime>
    </timeRange>
  </operationSchedule>
</operationScheduleList>
</localPermission>
<remotePermission>
  <!--ro, opt, object, 远程权限节点-->
  <record>
    <!--ro, opt, bool, 是否开启录像（所有通道）权限, desc:true为开启,false为关闭-->true
  </record>
  <playBack>
    <!--ro, opt, bool, 是否开启回放（所有通道）权限, desc:true为开启,false为关闭-->true
```

```
</playBack>
<preview>
  <!--ro, opt, bool, 是否开启预览（所有通道）权限, desc:true为开启,false为关闭-->true
</preview>
<videoChannelPermissionList>
  <!--ro, opt, object, 视频通道权限配置列表-->
  <videoChannelPermission>
    <!--ro, opt, object, 单一视频通道权限配置-->
    <id>
      <!--ro, req, string, 对应视频通道ID, range:[1,32]-->test
    </id>
    <preview>
      <!--ro, opt, bool, 是否开启当前通道预览权限, desc:true为开启,false为关闭-->true
    </preview>
    <playBack>
      <!--ro, opt, bool, 是否开启当前通道回放权限, desc:true为开启,false为关闭-->true
    </playBack>
    <record>
      <!--ro, opt, bool, 是否开启当前通道录像权限, desc:true为开启,false为关闭-->true
    </record>
    <playBackDoubleVerification>
      <!--ro, opt, bool, 是否开启当前通道回放二次认证权限, desc:true为开启,false为关闭-->true
    </playBackDoubleVerification>
  </videoChannelPermission>
</videoChannelPermissionList>
<ptzControl>
  <!--ro, opt, bool, 是否开启云台控制（所有通道）权限, desc:true为开启,false为关闭-->true
</ptzControl>
<ptzChannelPermissionList>
  <!--ro, opt, object, 视频通道云台控制权限列表-->
  <ptzChannelPermission>
    <!--ro, opt, object, 单一视频通道云台控制权限-->
    <id>
      <!--ro, req, string, 视频通道ID, range:[1,32], desc:/ISAPI/PTZCtrl/channels/ID-->test
    </id>
    <ptzControl>
      <!--ro, opt, bool, 是否开启当前通道云台控制权限, desc:true为开启,false为关闭-->true
    </ptzControl>
  </ptzChannelPermission>
</ptzChannelPermissionList>
<logOrStateCheck>
  <!--ro, opt, bool, 是否开启日志和设备状态查看权限-->true
</logOrStateCheck>
<parameterConfig>
  <!--ro, opt, bool, 是否开启设备远程参数设置权限-->true
</parameterConfig>
<restartOrShutdown>
  <!--ro, opt, bool, 是否开启重启或关机权限-->true
</restartOrShutdown>
<upgrade>
  <!--ro, opt, bool, 是否开启升级权限-->true
</upgrade>
<upgradeNotShow>
  <!--ro, opt, bool, 升级不展示, desc:为确保兼容, 设备端admin, installer用户升级权限也支持, 该字段返回true的话, 上层界面隐藏admin和installer升级权限的相关界面, 前后端暂没有installer, admin等用户, EN认证过程用web进行操作要求升级不能和installer和admin进行绑定, 手机app也会用到该协议, 设备如果不返回升级权限字段那么手机app就没有升级功-->true
</upgradeNotShow>
<voiceTalk>
  <!--ro, opt, bool, 是否开启语音对讲（所有通道）权限-->true
</voiceTalk>
<transparentChannel>
  <!--ro, opt, bool, 是否开启透明通道（所有通道）权限, desc:也即控制串口-->true
</transparentChannel>
<contorlLocalOut>
  <!--ro, opt, bool, 是否开启控制本地输出权限-->true
</contorlLocalOut>
<manageChannel>
  <!--ro, opt, bool, 是否开启通道管理权限-->true
</manageChannel>
<alarmOutOrUpload>
  <!--ro, opt, bool, 是否开启报警上报和输出权限-->true
</alarmOutOrUpload>
<IRAID>
  <!--ro, opt, object, Iraid快速权限配置-->
  <rapidConfiguration>
    <!--ro, opt, bool, 是否开启Iraid快速配置权限-->true
  </rapidConfiguration>
  <reset>
    <!--ro, opt, bool, 是否开启Iraid重置配置权限-->true
  </reset>
</IRAID>
<LEDConfig>
  <!--ro, opt, bool, 是否允许用户修改LED配置-->true
</LEDConfig>
<AIModelManagement>
  <!--ro, opt, bool, 是否开启AI模型包操作权限-->true
</AIModelManagement>
<AITaskManagement>
  <!--ro, opt, bool, 是否开启AI任务管理操作权限-->true
</AITaskManagement>
<materialReview>
  <!--ro, opt, bool, 素材审核权限, desc:信息发布-->true
</materialReview>
<programReview>
```

```
<!--ro, opt, bool, 节目审核权限, desc:信息发布-->true
</programReview>
<uploadMaterial>
  <!--ro, opt, bool, 上传素材, desc:信息发布-->true
</uploadMaterial>
<addProgram>
  <!--ro, opt, bool, 节目编辑权限, desc:增, 删, 改-->true
</addProgram>
<addSchedule>
  <!--ro, opt, bool, 日程编辑权限, desc:增, 删, 改-->true
</addSchedule>
<programPlay>
  <!--ro, opt, bool, 信息播放权限-->true
</programPlay>
<terminalMgr>
  <!--ro, opt, bool, 终端管理权限, desc:对终端的背光, Logo灯等参数设置-->true
</terminalMgr>
<shopSmart>
  <!--ro, opt, bool, 商业智能权限-->true
</shopSmart>
<materialReplace>
  <!--ro, opt, bool, 素材替换权限-->true
</materialReplace>
<scheduleRelease>
  <!--ro, opt, bool, 日程发布权限-->true
</scheduleRelease>
<scheduleReview>
  <!--ro, opt, bool, 日程审核-->true
</scheduleReview>
<addTerminal>
  <!--ro, opt, bool, 终端添加-->true
</addTerminal>
<terminalDispatch>
  <!--ro, opt, bool, 终端分配权限, desc:将该组织下的终端分配给子组织-->true
</terminalDispatch>
<addUser>
  <!--ro, opt, bool, 用户编辑权限, desc:增, 删, 改-->true
</addUser>
<addOrg>
  <!--ro, opt, bool, 组织编辑权限, desc:增, 删, 改-->true
</addOrg>
<logMgr>
  <!--ro, opt, bool, 日志管理-->true
</logMgr>
<dataUpdate>
  <!--ro, opt, bool, 数据更新, desc:叫号数据-->true
</dataUpdate>
<insertMessage>
  <!--ro, opt, bool, 插播消息权限-->true
</insertMessage>
<notReview>
  <!--ro, opt, bool, 无需审核权限-->true
</notReview>
<MgrTerminal>
  <!--ro, opt, object, 管理的终端或终端组-->
  <TerminalNoList size="10">
    <!--ro, opt, object, 终端号列表, attr:size(req, int, 数组元素总数)-->
    <terminalNo>
      <!--ro, opt, int, 终端号, range:[1,10]-->1
    </terminalNo>
  </TerminalNoList>
  <TerminalGroupNoList size="10">
    <!--ro, opt, object, 终端组列表, attr:size(req, int, 数组元素总数)-->
    <terminalGroupNo>
      <!--ro, opt, int, 终端组号, range:[1,10]-->1
    </terminalGroupNo>
  </TerminalGroupNoList>
</MgrTerminal>
<subSysOrZoneArm>
  <!--ro, opt, bool, 子系统/防区布防权限-->true
</subSysOrZoneArm>
<subSysOrZoneDisarm>
  <!--ro, opt, bool, 子系统/防区撤防权限-->true
</subSysOrZoneDisarm>
<subSysOrZoneClearArm>
  <!--ro, opt, bool, 子系统/防区消警权限-->true
</subSysOrZoneClearArm>
<zoneBypass>
  <!--ro, opt, bool, 防区旁路权限-->true
</zoneBypass>
<zoneBypassRecover>
  <!--ro, opt, bool, 防区旁路恢复权限-->true
</zoneBypassRecover>
<operateOutput>
  <!--ro, opt, bool, 操作继电器权限-->true
</operateOutput>
<supportLinkageSubSystemList>
  <!--ro, opt, object, 已使能子系统列表-->
  <subSystem>
    <!--ro, opt, int, 已使能子系统, range:[1,64]-->1
  </subSystem>
</supportLinkageSubSystemList>
<subSystemList>
```

```

<!--ro, opt, object, 子系统列表, desc:当用户添加或者配置operator时可关联子系统-->
<subSystem>
  <!--ro, opt, int, 子系统, range:[1,64]-->1
</subSystem>
</subSystemList>
<factoryReset>
  <!--ro, opt, bool, 恢复默认参数-->true
</factoryReset>
<arm>
  <!--ro, opt, bool, 布防-->true
</arm>
<disarm>
  <!--ro, opt, bool, 撤防-->true
</disarm>
<accessControl>
  <!--ro, opt, bool, 门禁控制-->true
</accessControl>
<restoreTamper>
  <!--ro, opt, bool, 恢复防拆配置, desc:使用描述见: /ISAPI/Security/UserPermission/capabilities-->true
</restoreTamper>
<restoreConfirmedAlarm>
  <!--ro, opt, bool, 恢复确认报警配置, desc:使用描述见: /ISAPI/Security/UserPermission/capabilities-->true
</restoreConfirmedAlarm>
<LPListAudit>
  <!--ro, opt, bool, 黑白名单操作权限,
desc:有这个权限的用户可以导入/导出/编辑/删除/查询黑白名单
对应接口如下:
车牌黑白名单查询:/ISAPI/Traffic/channels/<ID>/searchLPListAudit
导入/导出车牌黑白名单数据:/ISAPI/Traffic/channels/<ID>/LicensePlateAuditData
车牌名单添加修改/ISAPI/Traffic/channels/<channelID>/LicensePlateAuditData/record?format=json
车牌黑白名单数据删除: /ISAPI/Traffic/channels/<channelID>/LicensePlateAuditData/delete?format=json-->true
</LPListAudit>
<applicationWarehouse>
  <!--ro, opt, bool, 应用仓库管理权限, desc:不支持不返回。具备管理权限, 用户可进行导入、删除、启动HEOP APP等操作-->true
</applicationWarehouse>
<subSystemManagement>
  <!--ro, opt, bool, 子系统管理权限, desc:包含子系统管理相关的所有权限-->true
</subSystemManagement>
<userOfModifyKeypadPasswordList>
  <!--ro, opt, array, 允许修改键盘密码的用户列表, subType:object-->
<userOfModifyKeypadPassword>
  <!--ro, opt, object, 允许修改键盘密码的用户-->
  <subUserType>
    <!--ro, opt, enum, 子用户类型, subType:string, [cCloudAdmin#云端admin, LocalAdmin#本地admin, installerAdmin#安装管理员, installEmployee#安装雇员, LocalInstaller#本地installer, cCloudOperator#云端操作员, LocalOperator#本地操作员]-->cCloudAdmin
  </subUserType>
</userOfModifyKeypadPassword>
</userOfModifyKeypadPasswordList>
<userOfModifyUserPermissionList>
  <!--ro, opt, array, 允许修改用户权限的用户列表, subType:object-->
<userOfModifyUserPermission>
  <!--ro, opt, object, 允许修改用户权限的用户-->
  <subUserType>
    <!--ro, opt, enum, 子用户类型, subType:string, [cCloudAdmin#云端admin, LocalAdmin#本地admin, installerAdmin#安装管理员, installEmployee#安装雇员, LocalInstaller#本地installer, cCloudOperator#云端操作员, LocalOperator#本地操作员]-->cCloudAdmin
  </subUserType>
</userOfModifyUserPermission>
</userOfModifyUserPermissionList>
<falseAlarmMaterialLibMgr>
  <!--ro, opt, bool, 误报分析素材库管理权限-->true
</falseAlarmMaterialLibMgr>
<TIPConfig>
  <!--ro, opt, bool, TIP配置权限-->true
</TIPConfig>
<networkCfg>
  <!--ro, opt, bool, 网络配置权限, desc:包括无线和有线网络-->true
</networkCfg>
<ISUPAccess>
  <!--ro, opt, bool, ISUP接入配置权限, desc:指设备接入ISUP的权限配置, 对应ISUP接入领域-->true
</ISUPAccess>
<timeMgr>
  <!--ro, opt, bool, 时间管理权限, desc:指系统时间和时区管理的权限, 对应时间管理领域-->true
</timeMgr>
<HDDMgr>
  <!--ro, opt, bool, 硬盘管理权限, desc:指硬盘管理的权限, 对应硬盘管理领域-->true
</HDDMgr>
</remotePermission>
<passwordValidity>
  <!--ro, opt, int, 密码有效期, range:[0,365], desc:仅管理员用户可以修改, 单位: 天, [0, 365]-->1
</passwordValidity>
<subUserType>
  <!--ro, opt, enum, 子用户类型, subType:string, [cCloudAdmin#云端admin, LocalAdmin#本地admin, installerAdmin#安装管理员, installEmployee#安装雇员, LocalInstaller#本地installer, cCloudOperator#云端操作员, LocalOperator#本地操作员]-->cCloudAdmin
</subUserType>
</UserPermission>

```

69.2.18 获取全部用户权限能力

Request URL

GET /ISAPI/Security/UserPermission/capabilities

查询参数

无

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>
<UserPermissionList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 用户权限列表, attr:version{req, string, 协议版本}-->
  <UserPermission>
    <!--ro, req, object, 用户权限, desc:仅管理员用户可以修改-->
    <id min="1" max="32">
      <!--ro, req, string, 权限id, range:[1,32], attr:min{req, int, 最小值},max{req, int, 最大值}-->test
    </id>
    <userID min="1" max="32">
      <!--ro, req, string, 用户ID, range:[1,32], attr:min{req, int, 最小值},max{req, int, 最大值}-->test
    </userID>
    <userType opt="admin,operator,viewer,installer,manufacturer,single">
      <!--ro, req, enum, 用户类型, subtype:string, [admin#系统用户,operator#操作员,viewer#观察员,installer#安装员,manufacturer#制造者,single#单用户],
      attr:opt{req, string, 取值范围}-->admin
    </userType>
    <remotePermission>
      <!--ro, opt, object, 远程权限-->
      <record opt="true,false">
        <!--ro, opt, bool, 远程手动录像, attr:opt{req, string, 取值范围}-->true
      </record>
      <playBack opt="true,false">
        <!--ro, opt, bool, 远程回放, attr:opt{req, string, 取值范围}-->true
      </playBack>
      <preview opt="true,false">
        <!--ro, opt, bool, 远程预览, attr:opt{req, string, 取值范围}-->true
      </preview>
      <videoChannelPermissionList>
        <!--ro, opt, object, 视频通道权限配置列表-->
        <videoChannelPermission>
          <!--ro, opt, object, 单一视频通道权限配置-->
          <id min="1" max="32">
            <!--ro, req, string, 视频通道ID, range:[1,32], attr:min{req, int, 最小值},max{req, int, 最大值}-->test
          </id>
          <preview opt="true,false">
            <!--ro, opt, bool, 远程预览, attr:opt{req, string, 取值范围}-->true
          </preview>
          <playBack opt="true,false">
            <!--ro, opt, bool, 远程回放, attr:opt{req, string, 取值范围}-->true
          </playBack>
        </videoChannelPermission>
      </videoChannelPermissionList>
      <logOrStateCheck opt="true,false">
        <!--ro, opt, bool, 远程查看日志, attr:opt{req, string, 取值范围}-->true
      </logOrStateCheck>
      <parameterConfig opt="true,false">
        <!--ro, opt, bool, 远程设置参数, attr:opt{req, string, 取值范围}-->true
      </parameterConfig>
      <restartOrShutdown opt="true,false">
        <!--ro, opt, bool, 远程关机或重启, attr:opt{req, string, 取值范围}-->true
      </restartOrShutdown>
      <upgrade opt="true,false">
        <!--ro, opt, bool, 远程升级或格式化, attr:opt{req, string, 取值范围}-->true
      </upgrade>
      <voiceTalk opt="true,false">
        <!--ro, opt, bool, 远程语音对讲, attr:opt{req, string, 取值范围}-->true
      </voiceTalk>
      <transParentChannel opt="true,false">
        <!--ro, opt, bool, 远程控制串口, attr:opt{req, string, 取值范围}-->true
      </transParentChannel>
      <contorlLocalOut opt="true,false">
        <!--ro, opt, bool, 远程控制本地输出, attr:opt{req, string, 取值范围}-->true
      </contorlLocalOut>
      <alarmOutOrUpload opt="true,false">
        <!--ro, opt, bool, 远程请求报警上传或报警输出, attr:opt{req, string, 取值范围}-->true
      </alarmOutOrUpload>
    </remotePermission>
  </UserPermission>
</UserPermissionList>
```

69.3 时间管理

69.3.1 获取全部NTP服务器参数

Request URL

GET /ISAPI/System/time/ntpServers

查询参数

无

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<NTPServerList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, array, NTP服务器信息列表, subType:object, attr:version{req, string, 协议版本}-->
  <NTPServer>
    <!--ro, opt, object, NTP服务器信息-->
    <id>
      <!--ro, req, string, 索引-->1
    </id>
    <addressingFormatType>
      <!--ro, req, enum, NTP服务器地址类型, subType:string, [ipaddress#ip地址,hostname#域名]-->hostname
    </addressingFormatType>
    <hostName>
      <!--ro, opt, string, NTP服务器域名, range:[1,64]-->hik12345
    </hostName>
    <ipAddress>
      <!--ro, opt, string, IPV4地址, range:[1,32]-->192.168.1.112
    </ipAddress>
    <ipv6Address>
      <!--ro, opt, string, IPV6地址, range:[1,128]-->1030:C9B4:FF12:48AA:1A2B
    </ipv6Address>
    <portNo>
      <!--ro, opt, int, 端口号, range:[1,65535], desc:默认端口号123-->123
    </portNo>
    <synchronizeInterval>
      <!--ro, opt, int, 校时间隔, range:[1,10000], unit:min, unitType:时间-->1440
    </synchronizeInterval>
    <enabled>
      <!--ro, opt, bool, 使能, desc:默认不使能-->>false
    </enabled>
  </NTPServer>
</NTPServerList>
```

69.3.2 获取单个NTP服务器参数

Request URL

GET /ISAPI/System/time/ntpServers/<NTPServerID>

查询参数

参数名称	参数类型	描述
NTPServerID	string	NTP服务器编号

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<NTPServer xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, NTP服务器信息, attr:version{req, string, 协议版本}-->
  <id>
    <!--ro, req, string, ID-->1
  </id>
  <addressingFormatType>
    <!--ro, req, enum, NTP服务器IP地址类型, subType:string, [ipaddress#ip地址,hostname#域名]-->hostname
  </addressingFormatType>
  <hostName>
    <!--ro, opt, string, NTP服务器域名, range:[1,64]-->hik12345
  </hostName>
  <ipAddress>
    <!--ro, opt, string, IPV4地址, range:[1,32]-->192.168.1.112
  </ipAddress>
  <ipv6Address>
    <!--ro, opt, string, IPV6地址, range:[1,128]-->1030:C9B4:FF12:48AA:1A2B
  </ipv6Address>
  <portNo>
    <!--ro, opt, int, 端口号, range:[1,65535], step:1, desc:端口号-->1
  </portNo>
  <synchronizeInterval>
    <!--ro, opt, int, 校时间隔, range:[1,10800], step:1, unit:min, unitType:时间, desc:校时间隔,minutes-->1440
  </synchronizeInterval>
  <enabled>
    <!--ro, opt, bool, 使能, desc:默认不使能-->>false
  </enabled>
</NTPServer>
```

69.3.3 获取单个NTP服务器参数配置能力

Request URL

GET /ISAPI/System/time/ntpServers/<NTPServerID>/capabilities

查询参数

参数名称	参数类型	描述
NTPServerID	string	

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<NTPServer xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, NTP服务器信息, attr:version{req, string, 协议版本}-->
  <id min="1" max="1">
    <!--ro, req, string, ID, range:[0,1], attr:min{req, int, 最小值},max{req, int, 最大值}-->1
  </id>
  <addressingFormatType opt="ipaddress,hostname">
    <!--ro, req, enum, NTP服务器IP地址类型, subType:string, [ipaddress#ip地址,hostname#域名], attr:opt{req, string, 取值范围}, desc:ipaddress,hostname-->hostname
  </addressingFormatType>
  <hostName min="0" max="68">
    <!--ro, opt, string, NTP服务器域名地址, range:[0,68], attr:min{req, int, 最小值},max{req, int, 最大值}-->time.windows.com
  </hostName>
  <ipAddress min="0" max="32">
    <!--ro, opt, string, IPV4地址, range:[0,32], attr:min{req, int, 最小值},max{req, int, 最大值}, desc:IPV4地址-->192.168.1.112
  </ipAddress>
  <ipv6Address min="0" max="128">
    <!--ro, opt, string, IPV6地址, range:[0,128], attr:min{req, int, 最小值},max{req, int, 最大值}, desc:ipv6地址-->1030::C9B4:FF12:48AA:1A2B
  </ipv6Address>
  <portNo min="0" max="65535">
    <!--ro, opt, int, 端口号, range:[0,65535], attr:min{req, int, 最小值},max{req, int, 最大值}, desc:端口号-->123
  </portNo>
  <synchronizeInterval min="0" max="10800">
    <!--ro, opt, int, 校时间隔, range:[0,10800], unit:min, unitType:时间, attr:min{req, int, 最小值},max{req, int, 最大值}, desc:校时间隔,minutes-->1440
  </synchronizeInterval>
  <enabled opt="true,false">
    <!--ro, opt, bool, 使能, attr:opt{req, string, 取值范围}, desc:true-使能, false-不使能-->true
  </enabled>
</NTPServer>
```

69.3.4 配置单个NTP服务器参数

Request URL

PUT /ISAPI/System/time/ntpServers/<NTPServerID>

查询参数

参数名称	参数类型	描述
NTPServerID	string	NTP服务器编号

请求报文

```
<?xml version="1.0" encoding="UTF-8"?>

<NTPServer xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, NTP服务器信息, attr:version{req, string, 协议版本}-->
  <id>
    <!--req, string, ID-->1
  </id>
  <addressingFormatType>
    <!--req, enum, NTP服务器IP地址类型, subType:string, [ipaddress#ip地址,hostname#域名]-->hostname
  </addressingFormatType>
  <hostName>
    <!--opt, string, NTP服务器域名, range:[1,64]-->hik12345
  </hostName>
  <ipAddress>
    <!--opt, string, IPV4地址, range:[1,32]-->192.168.1.112
  </ipAddress>
  <ipv6Address>
    <!--opt, string, IPV6地址, range:[1,128]-->1030:C9B4:FF12:48AA:1A2B
  </ipv6Address>
  <portNo>
    <!--opt, int, 端口号, range:[1,65535], step:1, desc:端口号-->1
  </portNo>
  <synchronizeInterval>
    <!--opt, int, 校时间隔, range:[1,10800], step:1, unit:min, unitType:时间, desc:校时间隔,minutes-->1440
  </synchronizeInterval>
  <enabled>
    <!--opt, bool, 使能, desc:默认不使能-->>false
  </enabled>
</NTPServer>
```

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 响应消息, attr:version{ro, req, string, 版本}-->
  <requestURL>
    <!--ro, req, string, 请求的URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, 状态码, subType:int, [0#OK,1#OK,2#Device Busy,3#Device Error,4#Invalid Operation,5#Invalid XML Format,6#Invalid XML Content,7#Reboot Required]-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, 状态信息, subType:string, [OK#成功,Device Busy#设备忙,Device Error#设备异常,Invalid Operation#无效的操作,Invalid XML Format#无效的XML格式,Invalid XML Content#无效的XML内容,Reboot#设备重启]-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, 详细错误码英文描述, desc:详细错误码的英文描述-->OK
  </subStatusCode>
</ResponseStatus>
```

69.3.5 获取设备校时管理参数

Request URL

GET /ISAPI/System/time

查询参数

无

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<Time xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, 时间管理, attr:version{opt, string, 协议版本}-->
  <timeMode>
    <!--ro, req, enum, 校时方式, subType:string, [NTP#NTP校时,manual#手动校时,satellite#卫星校时,platform#平台校时,NONE#不允校时或无校时源,GB28181#GB28181校时]-->NTP
  </timeMode>
  <localTime>
    <!--ro, opt, string, 本地时间, range:[0,256], dep:and,{$.Time.timeMode,eq,manual}-->2019-02-28T10:50:44+08:00
  </localTime>
  <timeZone>
    <!--ro, opt, string, 时区, range:[0,256], dep:and,{$.Time.timeMode,eq,manual},{$.Time.timeMode,eq,NTP}-->CST-8:00:00DST00:30:00,M4.1.0/02:00:00,M10.5.0/02:00:00
  </timeZone>
  <satelliteInterval>
    <!--ro, opt, int, 卫星校时间隔, step:1, unit:min, unitType:时间, desc:单位: 分钟-->60
  </satelliteInterval>
  <isSummerTime>
    <!--ro, opt, bool, 当前设备返回时间是否为夏令时-->true
  </isSummerTime>
  <platformType>
    <!--ro, opt, enum, 平台类型, subType:string, [EZVIZ#萤石平台], dep:and,{$.Time.timeMode,eq,platform}, desc:仅当timeMode选为platform时存在, EZVIZ-萤石平台(本字段需要和萤石平台配置中萤石校时字段 (/ISAPI/System/Network/EZVIZ 中<enabLedTiming>填为true) 赋值保持一致-->EZVIZ
  </platformType>
  <platformNo>
    <!--ro, opt, int, 平台号, range:[1,2], dep:and,{$.Time.timeMode,eq,GB28181}, desc:/ISAPI/System/Network/SIP/<SIPServerID>中GB28181List下platformNo所配置, 双国标平台唯一标识, dep:and,{$.Time.timeMode,eq,GB28181}, -->1
  </platformNo>
</Time>
```

69.3.6 配置设备校时管理参数

Request URL

PUT /ISAPI/System/time

查询参数

无

请求报文

```
<?xml version="1.0" encoding="UTF-8"?>

<Time xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, 时间管理, attr:version{opt, string, 协议版本}-->
  <timeMode>
    <!--req, enum, 校时方式, subType:string, [NTP#NTP校时,manual#手动校时,satellite#卫星校时,platform#平台校时,NONE#不允校时或无校时源,GB28181#GB28181校时]->NTP
  </timeMode>
  <localTime>
    <!--opt, string, 本地时间, range:[0,256], dep:and,{$.Time.timeMode,eq,manual}-->2019-02-28T10:50:44+08:00
  </localTime>
  <timeZone>
    <!--opt, string, 时区, range:[0,256], dep:and,{$.Time.timeMode,eq,manual},{$.Time.timeMode,eq,NTP}-->CST-8:00:00DST00:30:00,M4.1.0/02:00:00,M10.5.0/02:00:00
  </timeZone>
  <satelliteInterval>
    <!--opt, int, 卫星校时间隔, step:1, unit:min, unitType:时间, desc:单位: 分钟-->60
  </satelliteInterval>
  <isSummerTime>
    <!--opt, bool, 当前设备返回时间是否为夏令时-->true
  </isSummerTime>
  <platformType>
    <!--opt, enum, 平台类型, subType:string, [EZVIZ#萤石平台], dep:and,{$.Time.timeMode,eq,platform}, desc:仅当timeMode选为platform时存在, EZVIZ-萤石平台(本字段需要和萤石平台配置中萤石校时字段 (/ISAPI/System/Network/EZVIZ 中<enabLedTiming>填为true) 赋值保持一致-->EZVIZ
  </platformType>
  <platformNo>
    <!--opt, int, 平台号, range:[1,2], dep:and,{$.Time.timeMode,eq,GB28181}, desc:/ISAPI/System/Network/SIP/<SIPServerID>中GB28181List下platformNo所配置, 双国标平台唯一标识, dep:and,{$.Time.timeMode,eq,GB28181}, -->1
  </platformNo>
</Time>
```

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 响应消息, attr:version{ro, req, string, 版本}-->
  <requestURL>
    <!--ro, req, string, 请求的URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, 状态码, subType:int, [0#OK,1#OK,2#Device Busy,3#Device Error,4#Invalid Operation,5#Invalid XML Format,6#Invalid XML Content,7#Reboot Required]-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, 状态信息, subType:string, [OK#成功,Device Busy#设备忙,Device Error#设备异常,Invalid Operation#无效的操作,Invalid XML Format#无效的XML格式,Invalid XML Content#无效的XML内容,Reboot#设备重启]-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, 详细错误码英文描述, desc: 详细错误码的英文描述-->OK
  </subStatusCode>
  <FailedNodeInfoList>
    <!--ro, opt, object, 节点失败信息列表, desc: 对于中心分析集群手动校时时,如果存在校时失败的分析节点,返回此结构-->
    <FailedNodeInfo>
      <!--ro, opt, object, 节点失败信息-->
      <nodeID>
        <!--ro, req, string, 节点ID, range:[0,64]-->test
      </nodeID>
      <nodeIP>
        <!--ro, req, string, 节点ID, range:[0,20]-->test
      </nodeIP>
      <reason>
        <!--ro, opt, string, 节点校时失败原因, range:[0,128]-->test
      </reason>
    </FailedNodeInfo>
  </FailedNodeInfoList>
</ResponseStatus>
```

69.3.7 获取设备校时管理能力

Request URL

GET /ISAPI/System/time/capabilities

查询参数

无

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<Time xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, 时间管理能力集, attr:version{opt, string, 协议版本}-->
  <timeMode opt="NTP,manual,satellite,SDK,28181,ONVIF,ALL(任何支持的校时方式都允许校时),NONE(不允校时或无校时源),platform">
    <!--ro, req, enum, 校时方式, subType:string, [NTP#NTP校时,manual#手动校时,satellite#卫星校时,platform#平台校时,NONE#不允校时或无校时源,GB28181#GB28181校时], attr:opt{opt, string, 取值范围}-->NTP
  </timeMode>
  <localTime min="0" max="256">
    <!--ro, opt, string, 本地时间, range:[0,256], attr:min{opt, string, 最小值},max{opt, string, 最大值}-->test
  </localTime>
  <timeZone min="0" max="256">
    <!--ro, opt, string, 时区, range:[0,256], attr:min{opt, string, 最小值},max{opt, string, 最大值}-->test
  </timeZone>
  <satelliteInterval min="0" max="3600">
    <!--ro, opt, int, 卫星校时间隔, step:1, unit:min, unitType:时间, attr:min{opt, string, 最小值},max{opt, string, 最大值}, desc:单位: 分钟-->60
  </satelliteInterval>
  <timeType opt="local,UTC">
    <!--ro, opt, enum, 时间类型, subType:string, [local#本地时间,UTC#UTC时间], attr:opt{opt, string, 取值范围}-->local
  </timeType>
  <platformType opt="EZVIZ">
    <!--ro, opt, enum, 平台类型, subType:string, [EZVIZ#萤石平台], dep:and,{$.Time.timeMode,eq,platform}, attr:opt{opt, string, 取值范围}-->EZVIZ
  </platformType>
  <platformNo min="1" max="2">
    <!--ro, opt, int, 平台号, range:[1,2], dep:and,{$.Time.timeMode,eq,GB28181}, attr:min{req, int, 最小值},max{req, int, 最大值}, desc:/ISAPI/System/Network/SIP/<SIPServerID>中GB28181List下platformNo所配置,双国标平台唯一标识, dep:and,{$.Time.timeMode,eq,GB28181},-->1
  </platformNo>
  <isSupportHistoryTime>
    <!--ro, opt, bool, 历史校时列表支持能力, desc:历史校时列表支持能力,功能对应的接口: /ISAPI/System/time/historyInfo?format=json-->true
  </isSupportHistoryTime>
  <isSupportTimeFilter>
    <!--ro, opt, bool, 校时过滤支持能力, desc:功能对应的接口能力: /ISAPI/System/time/filter/capabilities?format=json-->true
  </isSupportTimeFilter>
  <displayFormat opt="MM/dd/yyyy hh:mm,mm,dd-MM-yyyy hh,MM-dd-yyyy hh:mm,yyyy-MM-dd hh:mm">
    <!--ro, opt, enum, 时间显示格式, subType:string, [MM/dd/yyyy hh:mm#MM/dd/yyyy hh:mm,dd-MM-yyyy hh:mm#dd-MM-yyyy hh:mm,MM-dd-yyyy hh:mm#MM-dd-yyyy hh:mm,yyyy-MM-dd hh:mm#yyyy-MM-dd hh:mm], attr:opt{req, string, 取值范围}, desc:返回此节点表示支持时间显示格式配置,对应/ISAPI/System/time/timeType?format=json中的displayFormat节点-->MM/dd/yyyy hh:mm
  </displayFormat>
  <isSupportSyncDeviceNTPInfoToCamera>
    <!--ro, opt, bool, 同步设备NTP服务信息到相机能力, desc:功能对应的接口能力: /ISAPI/System/time/SyncDeviceNTPInfoToCamera/capabilities?format=json-->true
  </isSupportSyncDeviceNTPInfoToCamera>
  <isSupportNTPService>
    <!--ro, opt, bool, 是否支持设备NTP服务, desc:功能对应的接口能力: /ISAPI/System/time/NTPService/capabilities?format=json-->true
  </isSupportNTPService>
</Time>
```

69.3.8 配置手动校时

Request URL

PUT /ISAPI/System/time/localTime

查询参数

无

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 响应消息, attr:version{ro, req, string, 版本}-->
  <requestURL>
    <!--ro, req, string, 请求的URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, 状态码, subType:int, [0#OK,1#OK,2#Device Busy,3#Device Error,4#Invalid Operation,5#Invalid XML Format,6#Invalid XML Content,7#Reboot Required]-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, 状态信息, subType:string, [OK#成功,Device Busy#设备忙,Device Error#设备异常,Invalid Operation#无效的操作,Invalid XML Format#无效的XML格式,Invalid XML Content#无效的XML内容,Reboot#设备重启]-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, 详细错误码英文描述, desc:详细错误码的英文描述-->OK
  </subStatusCode>
</ResponseStatus>
```

69.3.9 获取时区

Request URL

GET /ISAPI/System/time/timeZone

查询参数

无

请求报文

无

响应报文

无

69.3.10 配置时区

Request URL

PUT /ISAPI/System/time/timeZone

查询参数

无

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 响应消息, attr:version{ro, req, string, 版本}-->
  <requestURL>
    <!--ro, req, string, 请求的URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, 状态码, subType:int, [0#OK,1#OK,2#Device Busy,3#Device Error,4#Invalid Operation,5#Invalid XML Format,6#Invalid XML Content,7#Reboot Required]-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, 状态信息, subType:string, [OK#成功,Device Busy#设备忙,Device Error#设备异常,Invalid Operation#无效的操作,Invalid XML Format#无效的XML格式,Invalid XML Content#无效的XML内容,Reboot#设备重启]-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, 详细错误码英文描述, desc:详细错误码的英文描述-->OK
  </subStatusCode>
</ResponseStatus>
```

69.3.11 获取全部NTP服务器参数配置能力

Request URL

GET /ISAPI/System/time/ntpServers/capabilities

查询参数

无

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<NTPServerList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, array, NTP服务信息列表, subType:object, attr:version{opt, string, 协议版本}-->
  <NTPServer>
    <!--ro, opt, object, NTP服务器信息-->
    <id>
      <!--ro, req, string, 索引-->1
    </id>
    <addressingFormatType opt="ipaddress,hostname">
      <!--ro, req, enum, NTP服务器地址类型, subType:string, [ipaddress#ip地址,hostname#域名], attr:opt{req, string, 取值范围}-->hostname
    </addressingFormatType>
    <hostName min="1" max="64">
      <!--ro, opt, string, NTP服务器域名, range:[1,64], attr:min{req, int, 最小值},max{req, int, 最大值}-->hik12345
    </hostName>
    <ipAddress min="1" max="32">
      <!--ro, opt, string, IPV4地址, range:[1,32], attr:min{req, int, 最小值},max{req, int, 最大值}, desc:IPV4地址-->192.168.1.112
    </ipAddress>
    <ipv6Address min="1" max="128">
      <!--ro, opt, string, IPV6地址, range:[1,128], attr:min{req, int, 最小值},max{req, int, 最大值}, desc:ipv6地址-->1030:C9B4:FF12:48AA:1A2B
    </ipv6Address>
    <portNo min="1" max="65535">
      <!--ro, opt, int, 端口号, range:[1,65535], attr:min{req, int, 最小值},max{req, int, 最大值}, desc:默认为 123-->123
    </portNo>
    <synchronizeInterval min="1" max="10800">
      <!--ro, opt, int, 校时间隔, unit:min, unitType:时间, attr:min{req, int, 最小值},max{req, int, 最大值}-->1440
    </synchronizeInterval>
    <enabled opt="true,false">
      <!--ro, opt, bool, 使能, attr:opt{req, string, 取值范围}, desc:true-使能, false-不使能-->true
    </enabled>
  </NTPServer>
</NTPServerList>
```

69.3.12 配置全部NTP服务器参数

Request URL

PUT /ISAPI/System/time/ntpServers

查询参数

无

请求报文

```
<?xml version="1.0" encoding="UTF-8"?>

<NTPServerList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, array, NTP服务信息列表, subType:object, attr:version{opt, string, 协议版本}-->
  <NTPServer>
    <!--opt, object, NTP服务器信息-->
    <id>
      <!--req, string, 索引-->1
    </id>
    <addressingFormatType>
      <!--req, enum, NTP服务器地址类型, subType:string, [ipaddress#ip地址,hostname#域名]-->hostname
    </addressingFormatType>
    <hostName>
      <!--opt, string, NTP服务器域名, range:[1,64]-->hik12345
    </hostName>
    <ipAddress>
      <!--opt, string, IPV4地址, range:[1,32]-->192.168.1.112
    </ipAddress>
    <ipv6Address>
      <!--opt, string, IPV6地址, range:[1,128]-->1030:C9B4:FF12:48AA:1A2B
    </ipv6Address>
    <portNo>
      <!--opt, int, 端口号, range:[1,65535], desc:默认端口号123-->123
    </portNo>
    <synchronizeInterval>
      <!--opt, int, 校时间隔, range:[1,10800], unit:min, unitType:时间-->1440
    </synchronizeInterval>
    <enabled>
      <!--opt, bool, 使能, desc:默认不使能-->>false
    </enabled>
  </NTPServer>
</NTPServerList>
```

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 响应消息, attr:version{ro, req, string, 版本}-->
  <requestURL>
    <!--ro, req, string, 请求的URL-->/ISAPI/xxxx
  </requestURL>
  <statusCode>
    <!--ro, req, enum, 状态码, subType:int, [0#OK,1#OK,2#Device Busy,3#Device Error,4#Invalid Operation,5#Invalid XML Format,6#Invalid XML Content,7#Reboot Required]-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, 状态信息, subType:string, [OK#成功,Device Busy#设备忙,Device Error#设备异常,Invalid Operation#无效的操作,Invalid XML Format#无效的XML格式,Invalid XML Content#无效的XML内容,Reboot#设备重启]-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, 详细错误码英文描述, desc:详细错误码的英文描述-->OK
  </subStatusCode>
</ResponseStatus>
```

69.3.13 获取手动校时

Request URL

GET /ISAPI/System/time/localTime

查询参数

无

请求报文

无

响应报文

无

69.4 日志管理

69.4.1 查询日志

Request URL

POST /ISAPI/ContentMgmt/logSearch

查询参数

无

请求报文

```
<?xml version="1.0" encoding="UTF-8"?>

<CMSearchDescription xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, 日志查询, attr:version{opt, string, 协议版本}-->
  <searchID>
    <!--req, string, 搜索ID, range:[1,32], desc:UUID/GUID. 搜索记录唯一标识,用来确认上层客户端是否为同一个(倘若是一个,则设备记录内存,下次搜索加快速度)-->test
  </searchID>
  <trackIDList>
    <!--opt, array, 通道列表, subType:object-->
    <trackID>
      <!--req, int, 通道号, desc:通道号-->1
    </trackID>
  </trackIDList>
  <metaId>
    <!--req, enum, 日志类型, subType:string, [Log.std-cgi.com/Alarm/methaneConcentrationException#危险气体浓度异常事件(默认甲烷),Log.std-cgi.com/Alarm/methaneLightIntensityException#危险气体光强异常事件(默认甲烷),Log.std-cgi.com/Alarm/fishingShipDetection#渔船检测], desc:{Log.std-cgi.com/<majorType>/<minorType>},<majorType>-日志主类型,<minorType>-日志次类型,详细日志类型定义参见备注-->log.std-cgi.com/Alarm/methaneConcentrationException
  </metaId>
  <timeSpanList>
    <!--opt, array, 搜索时间列表, subType:object, desc:搜索时间列表-->
    <timeSpan>
      <!--opt, object, 搜索时间, desc:搜索时间-->
      <startTime>
        <!--req, datetime, 开始时间, desc:ISO8601 time (TZ格式: UTC时间)-->1970-01-01T00:00:00Z
      </startTime>
      <endTime>
        <!--req, datetime, 结束时间, desc:ISO8601 time (TZ格式: UTC时间)-->1970-01-01T01:00:00Z
      </endTime>
    </timeSpan>
  </timeSpanList>
  <searchResultPostion>
    <!--opt, int, 查询结果起始位置, desc:表示查询结果在结果列表中的起始位置(该字段存在拼写错误,请各产线均按照错误拼写实现)-->100
  </searchResultPostion>
  <maxResults>
    <!--opt, int, 本次查询条数, range:[0,100]-->0
  </maxResults>
  <onlySmart>
    <!--opt, bool, 是否只搜索带smart信息的日志, desc:默认false-->>false
  </onlySmart>
  <logLevel>
    <!--opt, enum, 日志等级, subType:string, [emergency#emergency,alert#alert,critical#critical,error#error,warning#warning,notice#notice,information#information,debug#debug]-->emergency
  </logLevel>
  <alarmLevel>
    <!--opt, enum, 告警等级, subType:string, [high#high,middle#middle,lower#lower,minor#minor]-->high
  </alarmLevel>
</CMSearchDescription>
```

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<CMSearchResult xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 日志查询结果, attr:version{opt, string, 协议版本}-->
  <searchID>
    <!--ro, req, string, 搜索ID, range:[1,32], desc:UUID/GUID. 搜索记录唯一标识,用来确认上层客户端是否为同一个(倘若是一个,则设备记录内存,下次搜索加快速度)-->test
  </searchID>
  <responseStatus>
    <!--ro, req, bool, 查询状态-->true
  </responseStatus>
  <responseStatusStrg>
    <!--ro, req, enum, 查询状态字符串表示, subType:string, [OK#查询结束,NO MATCHES#查询失败,MORE#还有数据等待查询], desc:responseStatus = true, responseStatusStrg = OK, 查询结束-->OK
  </responseStatusStrg>
  <totalMatches>
    <!--ro, opt, int, 符合条件总条数-->12
  </totalMatches>
  <numOfMatches>
    <!--ro, opt, int, 本次返回的记录条数, range:[0,100]-->24
  </numOfMatches>
  <matchList>
    <!--ro, opt, array, 匹配列表, subType:object-->
    <searchMatchItem>
      <!--ro, opt, object, 搜索匹配项-->
      <logDescriptor>
        <!--ro, opt, object, 日志描述-->
        <metaId>
          <!--ro, req, string, 搜索日志类型, desc:Log.std-cgi.com/<majorType>/<minorType>, <majorType>-日志主类型,<minorType>-日志次类型,详细日志类型定义参见备注.-->log.std-cgi.com/Alarm/24HzZoneAlarm
        </metaId>
        <startDateTime>
          <!--ro, req, datetime, 日志时间, desc:ISO8601 time (TZ格式: UTC时间)-->1970-01-01T00:30:00Z
        </startDateTime>
        <localID>
          <!--ro, opt, string, 通道号-->1
        </localID>
      </logDescriptor>
    </searchMatchItem>
  </matchList>
```

```

<paraType>
  <!--ro, opt, string, 参数类型-->test
</paraType>
<userName>
  <!--ro, opt, string, 用户名-->Admin
</userName>
<infoContent>
  <!--ro, opt, string, 日志信息内容-->test
</infoContent>
<logInfo>
  <!--ro, opt, object, 日志信息-->
  <OpenDoorRecord>
    <!--ro, opt, object, 可视对讲开锁记录-->
    <type>
      <!--ro, req, enum, 开锁类型, subType:string, [password#密码开锁,hiJack#劫持开锁,card#刷卡开锁,resident#住户开锁,center#管理中心开锁]-->password
    </type>
  </OpenDoorRecord>
  <VisAlarmRecord>
    <!--ro, opt, object, 可视对讲报警记录-->
    <type>
      <!--ro, req, enum, 报警类型, subType:string, [zone#防区报警,dismantLe#防拆报警,hiJack#劫持报警,passwordErr#密码错误报警,doorNotOpen#门没开报警,doorNotClose#门没关报警,SOS#SOS报警,callReq#设备请求呼叫报警,smartLockHiJackFingerPrint#智能锁劫持指纹报警,smartLockHiJackPassword#智能锁劫持密码报警,smartLockBreaking#智能锁撬门报警,smartLockBeLocked#智能锁门锁定报警,smartLockLowBattery#智能锁电量不足报警]-->zone
    </type>
  </VisAlarmRecord>
</logInfo>
<ipAddress>
  <!--ro, opt, string, 设备ip地址, range:[1,32]-->10.20.30.10
</ipAddress>
<object>
  <!--ro, opt, enum, 操作对象, subType:string, [network#网络,keypad#键盘,remoteCtrl#遥控器,card#卡片]-->network
</object>
<params>
  <!--ro, opt, string, 日志参数, desc:参数 (防区号等)-->test
</params>
<seq>
  <!--ro, opt, string, 设备序列号, range:[1,32]-->test
</seq>
<additionInformation>
  <!--ro, opt, string, 附加信息, range:[1,128], desc:日志附加描述-->test
</additionInformation>
<panelUser>
  <!--ro, opt, string, 操作面板的用户名, range:[0,16]-->admin
</panelUser>
<diskNumber>
  <!--ro, opt, int, 硬盘号, range:[0,10000]-->0
</diskNumber>
<alarmInPort>
  <!--ro, opt, int, 报警输入端口, range:[0,1000]-->0
</alarmInPort>
<alarmOutPort>
  <!--ro, opt, int, 报警输出端口, range:[0,10000]-->0
</alarmOutPort>
<remoteHostIPAddress>
  <!--ro, opt, string, 远程主机IP地址, range:[1,32]-->10.12.25.23
</remoteHostIPAddress>
<logLevel>
  <!--ro, opt, enum, 日志等级, subType:string, [emergency#emergency>alert#alert,critical#critical,error#error,warning#warning,notice#notice,information#information,debug#debug]-->emergency
</logLevel>
<alarmLevel>
  <!--ro, opt, enum, 告警等级, subType:string, [high#high,middle#middle,lower#lower,minor#minor]-->high
</alarmLevel>
<moduleName>
  <!--ro, opt, string, 模块名称, range:[1,32]-->test
</moduleName>
</logDescriptor>
</searchMatchItem>
</matchList>
</CMSearchResult>

```

69.4.2 配置接入syslog参数

Request URL

PUT /ISAPI/System/logServer

查询参数

无

请求报文

```
<?xml version="1.0" encoding="UTF-8"?>

<LogServer xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--opt, object, 日志服务, attr:version{req, string, 协议版本}-->
  <enabled>
    <!--req, bool, 日志存储服务器使能-->true
  </enabled>
  <addressingFormatType>
    <!--req, enum, 服务器地址类型, subType:string, [ipaddress#IP地址,hostname#主机名称]-->ipaddress
  </addressingFormatType>
  <hostName>
    <!--opt, string, 主机名称-->test
  </hostName>
  <ipAddress>
    <!--opt, string, IP地址-->test
  </ipAddress>
  <ipv6Address>
    <!--opt, string, IPV6地址-->test
  </ipv6Address>
  <portNo>
    <!--opt, int, 服务器端口号-->1
  </portNo>
  <transmissionEncryption>
    <!--opt, bool, 传输加密-->true
  </transmissionEncryption>
  <checkCertificate>
    <!--opt, bool, 证书校验使能-->true
  </checkCertificate>
  <uploadInterval>
    <!--opt, int, 日志上传周期, range:[1,24], desc:单位小时, 默认1小时-->1
  </uploadInterval>
  <protocolType>
    <!--opt, enum, 日志上传协议类型, subType:string, [tcp#tcp协议,udp#udp协议], desc:默认采用TCP方式-->tcp
  </protocolType>
  <sysLogCacheSize>
    <!--opt, int, 系统日志缓存大小, range:[10,3000]-->1000
  </sysLogCacheSize>
  <sysLogLevel>
    <!--opt, enum, 系统日志等级, subType:string,
    [emergency#emergency,alert#alert,critical#critical,error#error,warning#warning,notice#notice,information#information,debug#debug]-->emergency
  </sysLogLevel>
  <serverLogLevel>
    <!--opt, enum, 服务器日志等级, subType:string,
    [emergency#emergency,alert#alert,critical#critical,error#error,warning#warning,notice#notice,information#information,debug#debug]-->emergency
  </serverLogLevel>
  <alarmLogCacheSize>
    <!--opt, int, 告警日志缓存大小, range:[10,3000]-->1000
  </alarmLogCacheSize>
</LogServer>
```

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 响应消息, attr:version{ro, req, string, 版本}-->
  <requestURL>
    <!--ro, req, string, 请求的URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, 状态码, subType:int, [0#OK,1#OK,2#Device Busy,3#Device Error,4#Invalid Operation,5#Invalid XML Format,6#Invalid XML Content,7#Reboot Required]-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, 状态信息, subType:string, [OK#成功,Device Busy#设备忙,Device Error#设备异常,Invalid Operation#无效的操作,Invalid XML Format#无效的XML格式,Invalid XML Content#无效的XML内容,Reboot#设备重启]-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, 详细错误码英文描述, desc:详细错误码的英文描述-->OK
  </subStatusCode>
</ResponseStatus>
```

69.4.3 获取接入syslog参数

Request URL

GET /ISAPI/System/logServer

查询参数

无

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<LogServer xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, 日志服务, attr:version{req, string, 协议版本}-->
  <enabled>
    <!--ro, req, bool, 日志存储服务器使能-->true
  </enabled>
  <addressingFormatType>
    <!--ro, req, enum, 服务器地址类型, subType:string, [ipaddress#IP地址,hostname#主机名称]-->ipaddress
  </addressingFormatType>
  <hostName>
    <!--ro, opt, string, 主机名称-->test
  </hostName>
  <ipAddress>
    <!--ro, opt, string, IP地址-->test
  </ipAddress>
  <ipv6Address>
    <!--ro, opt, string, IPV6地址-->test
  </ipv6Address>
  <portNo>
    <!--ro, opt, int, 服务器端口号-->1
  </portNo>
  <transmissionEncryption>
    <!--ro, opt, bool, 传输加密-->true
  </transmissionEncryption>
  <checkCertificate>
    <!--ro, opt, bool, 证书校验使能-->true
  </checkCertificate>
  <uploadInterval>
    <!--ro, opt, int, 日志上传周期, range:[1,24], desc:单位小时, 默认1小时-->1
  </uploadInterval>
  <protocolType>
    <!--ro, opt, enum, 日志上传协议类型, subType:string, [tcp#tcp协议,udp#udp协议], desc:默认采用TCP方式-->tcp
  </protocolType>
  <sysLogCacheSize>
    <!--ro, opt, int, 系统日志缓存大小, range:[10,3000]-->1000
  </sysLogCacheSize>
  <sysLogLevel>
    <!--ro, opt, enum, 系统日志等级, subType:string,
    [emergency#emergency,alert#alert,critical#critical,error#error,warning#warning,notice#notice,information#information,debug#debug]-->emergency
  </sysLogLevel>
  <serverLogLevel>
    <!--ro, opt, enum, 服务器日志等级, subType:string,
    [emergency#emergency,alert#alert,critical#critical,error#error,warning#warning,notice#notice,information#information,debug#debug]-->emergency
  </serverLogLevel>
  <alarmLogCacheSize>
    <!--ro, opt, int, 告警日志缓存大小, range:[10,3000]-->1000
  </alarmLogCacheSize>
</LogServer>
```

69.4.4 获取接入syslog参数能力

Request URL

GET /ISAPI/System/logServer/capabilities

查询参数

无

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<LogServerCap xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, 日志服务能力, attr:version{req, string, 协议版本}-->
  <enabled opt="true,false">
    <!--ro, opt, string, 使能, attr:opt{req, string, 取值范围}-->test
  </enabled>
  <addressingFormatType opt="ipaddress,hostname">
    <!--ro, req, string, 地址类型, attr:opt{req, string, 地址类型}-->test
  </addressingFormatType>
  <hostName min="1" max="10">
    <!--ro, opt, string, 域名, attr:min{req, int, 最小值},max{req, int, 最大值}-->test
  </hostName>
  <ipAddress min="1" max="10">
    <!--ro, opt, string, ip地址, attr:min{req, int, 最小值},max{req, int, 最大值}-->test
  </ipAddress>
  <ipv6Address min="1" max="10">
    <!--ro, opt, string, ipv6地址, attr:min{req, int, 最小值},max{req, int, 最大值}-->test
  </ipv6Address>
  <portNo min="1" max="10">
    <!--ro, opt, string, 端口, attr:min{req, int, 最小值},max{req, int, 最大值}-->1
  </portNo>
  <transmissionEncryption opt="true,false">
    <!--ro, opt, string, 传输加密, attr:opt{req, bool, 取值范围}-->true
  </transmissionEncryption>
  <checkCertificate opt="true,false">
    <!--ro, opt, string, 证书校验使能, attr:opt{req, bool, 取值范围}-->true
  </checkCertificate>
  <uploadInterval min="1" max="24" def="1">
    <!--ro, opt, string, 日志上传周期, attr:min{req, int, 最小值},max{req, int, 最大值},def{req, int, 默认值}-->1
  </uploadInterval>
  <protocolType opt="tcp,udp">
    <!--ro, opt, enum, 日志上传协议类型, subType:string, [tcp#tcp协议,udp#udp协议], attr:opt{req, string, 取值范围}, desc:默认采用TCP方式-->tcp
  </protocolType>
  <sysLogCacheSize min="10" max="3000" def="1000">
    <!--ro, opt, string, 系统日志缓存大小, range:[10,3000], attr:min{req, int, 最小值},max{req, int, 最大值},def{req, int, 默认值}-->1000
  </sysLogCacheSize>
  <sysLogLevel opt="emergency,alert,critical,error,warning,notice,information,debug">
    <!--ro, opt, enum, 系统日志等级, subType:string,
    [emergency#emergency,alert#alert,critical#critical,error#error,warning#warning,notice#notice,information#information,debug#debug], attr:opt{req, string, 取
    值范围}-->emergency
  </sysLogLevel>
  <serverLogLevel opt="emergency,alert,critical,error,warning,notice,information,debug">
    <!--ro, opt, enum, 服务器日志等级, subType:string,
    [emergency#emergency,alert#alert,critical#critical,error#error,warning#warning,notice#notice,information#information,debug#debug], attr:opt{req, string, 取
    值范围}-->emergency
  </serverLogLevel>
  <alarmLogCacheSize min="10" max="3000" def="1000">
    <!--ro, opt, string, 告警日志缓存大小, range:[10,3000], attr:min{req, int, 最小值},max{req, int, 最大值},def{req, int, 默认值}-->1000
  </alarmLogCacheSize>
</LogServerCap>
```

69.5 事件订阅

69.5.1 建立布防订阅连接

Request URL

POST /ISAPI/Event/notification/subscribeEvent?deployID=<deployID>

查询参数

参数名称	参数类型	描述
deployID	string	0-客户端布防, 1-实时布防, deployID参数不存在时, 表示客户端布防

请求报文

```
<?xml version="1.0" encoding="UTF-8"?>

<SubscribeEvent xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, 该节点表示所有含图片事件图片上传方式, attr:version{req, string, 协议版本}-->
  <heartbeat>
    <!--opt, int, 心跳间隔时间, range:[1,180], unit:s, unitType:时间-->1
  </heartbeat>
  <channelMode>
    <!--opt, enum, 订阅通道类型, subType:string, [all#所有通道订阅,List#按具体通道号列表进行订阅]-->list
  </channelMode>
  <eventMode>
    <!--req, enum, 事件订阅模式, subType:string, [all#订阅设备的所有通道下目标的事件,List#按具体事件类型进行订阅]-->list
  </eventMode>
  <EventList>
    <!--opt, array, 订阅事件类型列表, subType:object, desc:当eventMode为List时生效-->
  </EventList>
```

```
    <!--opt, object, 单一事件订阅-->
    <type>
    <!--req, enum, 事件类型, subType:string, [ADAS#高级辅助驾驶报警,ADASAlarm#高级驾驶辅助报警,AID#交通事件检测,ANPR#车牌识别,AccessControllerEvent#门禁主机检测事件,CDSStatus#光盘刻录状态上报,DBD#驾驶行为报警,GPSUpload#GPS信息上传,HFPD#高频人员报警,IO#信号量报警,IOTD#非视频设备检测事件,LES#物流扫描事件,LFPD#低频人员检测,PALMismatch#视频制式不匹配,PIR#PIR,PeopleCounting#客流量统计,PeopleNumChange#人数异常,Standup#人员站立检测,TMA#测温报警,TMPA#测温预警,VMD#移动侦测,abnormalAcceleration#设备加速度异常报警,abnormalDriving#异常驾驶行为,advReachHeight#折线攀高检测,alarmResult#人脸比对,attendance#考勤事件,attendedBaggage#物品拿取,audioAbnormal#声强突变检测,audioException#音频异常侦测,behaviorResult#行为分析结果上报,blindSpotDetection#盲区监测报警,cardMatch#认证比对,changedStatus#状态变化上报,collision#碰撞,containerDetection#集装箱检测,crowdSituationAnalysis#人群态势分析,databaseException#数据库异常,defocus#虚焦侦测,diskUnformat#磁盘未格式化,diskError#硬盘错误,diskFull#硬盘满,driverConditionMonitor#驾驶员状态监测报警,emergencyAlarm#紧急报警,faceCapture#人脸抓拍识别,faceSnapModeling#人脸抓拍建模,facedetection#人脸侦测,failDown#人员倒地（使用错误的拼写,faultAlarm#故障告警,fielddetection#区域入侵,fireDetection#火点检测,fireEscapeDetection#消防通道检测,flowOverrun#流量超限,framesPeopleCounting#区域人数统计,getUp#起身检测,group#人员聚集,hdBadBlock#硬盘坏块,hdImpact#硬盘冲击,heatmap#热度图报警,highHDTemperature#硬盘温度过高,highTempAlarm#设备温度过高报警,hotSpare#热备异常,iLLAccess#非法访问,ipcTransferAbnormal#IPC迁移失败,ipconflict#IP地址冲突,keyPersonGetUp#重点人员起身检测,leavePosition#离岗检测,linedetection#越界侦测（穿越警戒线）,listSyncException#列表（名单）同步异常,Loitering#徘徊侦测,LowHDTemperature#硬盘温度过低,mixedTargetDetection#混合目标检测,modelError#模型错误,nicBroken#网络断开,nodeOffline#掉线异常,nonPoliceIntrusion#非警察人员入侵,overSpeed#超速报警,overTimeTarry#超时滞留检测,parking#停车侦测,peopleNumChange#人数变化,peopleNumCounting#人数统计（区别于区域人数统计）,personAbnormalAlarm#人员身份异常报警,personDensityDetection#人员密度,personQueueCounting#排队人数检测,personQueueDetection#排队检测,personQueueRealTime#人员排队检测实时数据上报,personQueueTime#排队时长检测,pLayCellphone#玩手机检测,pocException#视频异常,poe#POE供电异常,policeAbsent#警察不在场,radarAlarm#雷达告警,radarFieldDetection#雷达区域入侵,radarLineDetection#雷达越界侦测,radarPerimeterRule#雷达规则数据,radarTargetDetection#雷达目标数据,radarVideoDetection#雷达目标检测,raidException#阵列异常,rapidMove#快速移动,reachHeight#攀高检测,recordCycleAbnormal#录像周期不足,recordException#录像异常,regionEntrance#进入区域,regionExiting#离开区域,retention#人员滞留事件,roLLover#侧翻,running#人员奔跑,safetyHelmetDetection#安全帽检测,scenechangedetection#场景变更,sensorAlarm#角速度加速度报警,severeHDFailure#硬盘严重故障,shellterAlarm#遮挡报警,shipsDetection#船只检测,sitQuietly#静坐检测,smokeAndFireDetection#烟火检测,smokeDetection#烟雾检测,softIO#报警软IO,spacingChange#间距异常,sysStorFull#集群系统存储满异常,takingElevatorDetection#乘梯检测（电瓶车进电梯检测）,targetCapture#目标抓拍,temperature#温差报警,thermometry#温度报警,thirdPartyException#第三方异常,toiletTarry#如厕超时检测,tollCodeInfo#二维码信息上报,tossing#抛物,unattendedBaggage#物品遗留,vehicleMatchResult#黑白名单车辆报警上传,vehicleRecogResult#车辆图片二次识别,versionAbnormal#集群版本异常,videoException#视频异常,videoLoss#视频丢失,violationAlarm#违规告警,violentMotion#剧烈运动,yardTarry#放风场滞留检测,CertificateCaptureEvent#证件抓拍比对事件,UncertificateCompareEvent#无证比对事件,ConsumptionAndTransactionRecordEvent#消费和交易记录事件,ConsumptionEvent#消费事件,TransactionRecordEvent#交易记录事件,TFS#交通取证,SetMealQuery#消费套餐查询事件,ConsumptionStatusQuery#消费状态查询,certificateRevocation#证书失效,humanBodyComparison#人体比对,regionTargetNumberCounting#区域目标统计,cabinetChargingQuery#智能柜计费查询请求], desc: 参考事件类型列表eventType-->mixedTargetDetection
    </type>
    <minorAlarm>
    <!--opt, string, 报警次类型（参考事件上传宏定义, desc:参考事件上传宏定义;type为AccessControllerEvent该字段必填-->0x400,0x401,0x402,0x403
    <minorAlarm>
    <minorException>
    <!--opt, string, 异常次类型, desc:参考事件上传宏定义;type为AccessControllerEvent该字段必填-->0x400,0x401,0x402,0x403
    <minorException>
    <minorOperation>
    <!--opt, string, 操作次类型, desc:参考事件上传宏定义;type为AccessControllerEvent该字段必填-->0x400,0x401,0x402,0x403
    <minorOperation>
    <minorEvent>
    <!--opt, string, 事件次类型, desc:参考事件上传宏定义;type为AccessControllerEvent该字段必填-->0x01,0x02,0x03,0x04
    <minorEvent>
    <pictureURLType>
    <!--opt, enum, 报警图片格式, subType:string, [binary#二进制,LocalURL#设备本地URL,cCloudStorageURL#云存储URL]->CloudStorageURL
    </pictureURLType>
    <channels>
    <!--opt, string, 事件关联的通道信息, desc:支持复选;用英文逗号隔开-->1,2,3,4
    <channels>
    <ConferenceIDList>
    <!--opt, array, 视频会议会议ID列表, subType:object, desc:不包含该字段表示订阅所有会议;此处理解为某个事件支持上报的会议号-->
    <ConferenceID>
    <!--opt, string, 订阅的视频会议ID号, _range:[1,32]-->test
    </ConferenceID>
    <ConferenceIDList>
    <uploadAllTarget>
    <!--opt, bool, 是否上传所有检测目标, desc:周界防范中用于NVR对IPC配置, 以便于NVR和获取到IPC检测到的所有目标,默认为false-->>false
    </uploadAllTarget>
    <uploadMEFData>
    <!--opt, bool, 是否上传事件关联的电围数据, desc:默认为true. 电围数据上报关联智能事件上传, 由于电围模块采集数据间隔几分钟一次, 为使关联的电围数据中包含某手机号信息, 关联的电围数据需前向偏移一定时间, 智能事件要在达到偏移时间后才能上报, 由此事件会较触发时间延迟, 布防订阅时关闭电围数据上传, 一旦智能事件（人脸抓拍识别事件、混合目标检测事件等）触发, 设备立即上报智能事件（不含电围数据）-->>true
    </uploadMEFData>
    <minorType>
    <!--opt, enum, 事件子类型, subType:string, [success#成功,failure#失败],
    desc:1、当type为CertificateCaptureEvent或UncertificateCompareEvent时有效。
    2、对于高铁站人员安检和酒店客人办理入住场景, 真实存在只需要订阅成功或者失败某一类的事件。
    3、当该字段不下发时, 代表订阅成功和失败两种事件。-->success
    <minorType>
    </Event>
    </EventList>
    <channels>
    <!--opt, string, 通道统一布控, desc:如果该字段下发;那么Event中channels不做生效。如果布防全部通道;则不下发这个节点。布防部分通道时;列出通道号;支持复选;用英文逗号隔开。-->1,2,3,4
    <channels>
    <pictureURLType>
    <!--opt, enum, 报警图片格式统一配置, subType:string, [binary#二进制,LocalURL#设备本地URL,cCloudStorageURL#云存储URL], desc:该节点表示所有含图片事件图片上传方;如果该字段下发;那么<Event>中<pictureURLType>不做生效。如果不下发这个节点;设备采用能力集返回的图片默认上传方式上报;前端设备默认为二进制;后端设备默认为设备本地URL。-->CloudStorageURL
    </pictureURLType>
    <ChangedUploadSub>
    <!--opt, object, 设备状态变化上报信息订阅-->
    <interval>
    <!--opt, int, 布防guid的生命周期, range:[1,300], unit:s, unitType:时间, desc:默认5分钟;布防guid的生命周期;该间隔期间未过来重连;设备认为是新的一路布防;重新生成guid-->300
    </interval>
    <StatusSub>
    <!--opt, object, 状态信息-->
    <all>
    <!--opt, bool, 是否订阅所有通道、磁盘、能力变化状态-->>false
    </all>
    <channel>
    <!--opt, bool, 订阅通道状态, desc:all为true时不必传入-->true
    </channel>
```

```
<hd>
  <!--opt, bool, 订阅磁盘状态, desc:all为true时不必传入-->true
</hd>
<capability>
  <!--opt, bool, 订阅能力集变化状态, desc:all为true时不必传入-->true
</capability>
<ChanStatus>
  <!--ro, opt, object, 变化上报中通道状态订阅, desc:当channel为true时有效,无此节点表示所有都上报。场景:如录像状态上报太频繁,平台更关注设备在线状态和布防状态,所以想要取消对录像状态变化上报的订阅-->
  <online>
    <!--ro, opt, bool, 订阅通道在线状态, desc:无此节点表示不要该报警-->true
  </online>
  <record>
    <!--ro, opt, bool, 订阅是否在录像, desc:无此节点表示不要该报警-->true
  </record>
  <recordStatus>
    <!--ro, opt, bool, 订阅是录像状态, desc:无此节点表示不要该报警-->true
  </recordStatus>
  <signal>
    <!--ro, opt, bool, 订阅信号状态, desc:无此节点表示不要该报警-->true
  </signal>
  <arming>
    <!--ro, opt, bool, 订阅NVR对前端设备的布防状态, desc:无此节点表示不要该报警-->true
  </arming>
</ChanStatus>
</StatusSub>
</ChangedUploadSub>
<identityKey>
  <!--opt, string, 订阅连接交互口令, range:[1,64], desc:mickey超脑产品引入,支持针对指定导入的人脸库(导入中携带identityKey口令)的比对结果进行布控-->test
</identityKey>
<ConferenceIDList>
  <!--opt, array, 视频会议会议ID列表, subType:object-->
  <ConferenceID>
    <!--opt, string, 订阅的视频会议ID号, range:[1,32]-->test
  </ConferenceID>
</ConferenceIDList>
<subscribeEventID>
  <!--opt, string, 自定义订阅ID号;平台下发时确保其唯一性;下发该字段后设备默认开启全事件断网续传, range:[1,128], desc:建立布防订阅成功(POST /ISAPI/Event/notification/subscribeEvent)时;下发该字段后;可替代SubscribeEventResponse中返回的id字段使用-->test
</subscribeEventID>
<level>
  <!--opt, enum, 布防等级, subType:string, [high#高,middle#中,low#低], desc:不填默认为high-->high
</level>
<middleLevelPictureEnabeld>
  <!--opt, bool, 中等级布防上传图片使能, desc:没有此节点默认为true-->true
</middleLevelPictureEnabeld>
</SubscribeEvent>
```

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<SubscribeEventResponse xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 订阅结果, attr:version{req, string, 协议版本}-->
  <id>
    <!--ro, req, int, 订阅ID;由设备返回并确保唯一性, range:[1,64]-->1
  </id>
  <FailedEventList>
    <!--ro, opt, array, 订阅失败的事件列表, subType:object, desc:在订阅失败的时候,必须返回;上层通过判断<FailedEventList>是否存在;判断设备是否全部订阅成功-->
    <Event>
      <!--ro, opt, object, 订阅失败事件;报警类型名称从能力集获取-->
      <type>
        <!--ro, req, enum, 订阅失败事件类型, subType:string, [ADAS#高级辅助驾驶报警,ADASAlarm#高级驾驶辅助报警,AID#交通事件检测,ANPR#车牌识别,AccessControllerEvent#门禁主机检测事件,CDsStatus#光盘刻录状态上报,DBD#驾驶行为报警,GPSToLoad#GPS信息上传,HFPD#高频人员报警,IO#信号量报警,IOTD#非视频设备检测事件,LES#物流扫描事件,LFPD#低频人员检测,PALMismatch#视频制式不匹配,PIR#PIR,PeopleCounting#客流量统计,PeopleNumChange#人数异常,Standup#人员站立检测,TMA#测温报警,TMPA#测温预警,VMD#移动侦测,abnormalAcceleration#设备加速度异常报警,abnormalDriving#异常驾驶行为,advReachHeight#折线攀高检测,alarmResult#人脸比对,attendance#考勤事件,attendedBaggage#物品拿取,audioAbnormal#声强突变检测,audioException#音频异常侦测,behaviorResult#行为分析结果上报,blindSpotDetection#盲区监测报警,cardMatch#认证比对,changedStatus#状态变化上报,collision#碰撞,containerDetection#集装箱检测,crowdSituationAnalysis#人群态势分析,databaseException#数据库异常,defocus#虚焦侦测,diskUnformat#磁盘未格式化,diskerror#硬盘错误,diskfull#硬盘满,driverConditionMonitor#驾驶员状态监测报警,emergencyAlarm#紧急报警,faceCapture#人脸抓拍识别,faceSnapModeling#人脸抓拍建模,facedetection#人脸侦测,failDown#人员倒地（使用错误的拼写,faultAlarm#故障告警,fieldDetection#区域入侵,fireDetection#火点检测,fireEscapeDetection#消防通道检测,flowOverrun#流量超限,framesPeopleCounting#区域人数统计,getUp#起身检测,group#人员聚集,hdBadBlock#硬盘坏块,hdImpact#硬盘冲击,heatmap#热度图报警,highTemperature#硬盘温度过高,highTempAlarm#设备温度过高报警,hotSpare#热备异常,iLAccess#非法访问,ipcTransferAbnormal#IPC迁移失败,ipConflict#IP地址冲突,keyPersonGetUp#重点人员起身检测,leavePosition#离岗检测,linedetection#越界侦测（穿越警戒线）,listSyncException#列表（名单）同步异常,loitering#徘徊侦测,LowTemperature#硬盘温度过低,mixedTargetDetection#混合目标检测,modelError#模型错误,nicbroken#网络断开,nodeOffline#掉线异常,nonPoliceIntrusion#非警察人员入侵,overSpeed#超速报警,overtimeTarry#超时滞留检测,parking#停车侦测,peopleNumChange#人数变化,peopleNumCounting#人数统计（区别于区域人数统计）,personAbnormalAlarm#人员身份异常报警,personDensityDetection#人员密度,personQueueCounting#排队人数检测,personQueueDetection#排队检测,personQueueRealTime#人员排队检测实时数据上报,personQueueTime#排队时长检测,playCellPhone#玩手机检测,pocException#视频异常,poe#POE供电异常,policeAbsent#警察不在场,radarAlarm#雷达告警,radarFieldDetection#雷达区域入侵,radarLineDetection#雷达越界侦测,radarPerimeterRule#雷达规则数据,radarTargetDetection#雷达目标数据,radarVideoDetection#雷达目标检测,raidException#阵列异常,rapidMove#快速移动,reachHeight#攀高检测,recordCycleAbnormal#录像周期不足,recordException#录像异常,regionEntrance#进入区域,regionExiting#离开区域,retention#人员滞留事件,rollOver#侧翻,running#人员奔跑,safetyHelmetDetection#安全帽检测,scenechangedetection#场景变更,sensorAlarm#角速度加速度报警,severeFailure#硬盘严重故障,shelterAlarm#遮挡报警,shipsDetection#船只检测,sitQuietly#静坐检测,smokeAndFireDetection#烟火检测,smokeDetection#烟雾检测,softIO#报警软IO,spacingChange#间距异常,sysStorFull#集群系统存储满异常,takingElevatorDetection#乘梯检测（电瓶车进电梯检测）,targetCapture#目标抓拍,temperature#温差报警,thermometry#温度报警,thirdPartyException#第三方异常,toiletTarry#如厕超时检测,tollCodeInfo#二维码信息上报,tossing#抛物,unattendedBaggage#物品遗留,vehicleMatchResult#黑白名单车辆报警上传,vehicleRcogResult#车辆图片二次识别,versionAbnormal#集群版本异常,videoException#视频异常,videoLoss#视频丢失,violationAlarm#违规告警,violentMotion#剧烈运动,yardTarry#放风场滞留检测,CertificateCaptureEvent#证件抓拍比对事件,UncertificateCompareEvent#无证比对事件], desc:对于门禁、身份证事件,如果四种次类型有任意一种订阅失败,则认为订阅门禁、身份证事件失败. 返回方式如下:如果报警次类型成功,但异常、操作、事件次类型失败,那么minorAlarm的值为空,minorException、minorOperation、minorEvent的值为失败事件的类型值-->ADAS
      </type>
      <minorAlarm>
        <!--ro, opt, string, 报警次类型, desc:参考事件上传宏定义, type为AccessControllerEvent, IDCardInfoEvent该字段必填-->0x400,0x401,0x402,0x403
      </minorAlarm>
      <minorException>
        <!--ro, opt, string, 异常次类型, desc:参考事件上传宏定义, type为AccessControllerEvent, IDCardInfoEvent该字段必填-->0x400,0x401,0x402,0x403
      </minorException>
      <minorOperation>
        <!--ro, opt, string, 操作次类型, desc:参考事件上传宏定义, type为AccessControllerEvent, IDCardInfoEvent该字段必填-->0x400,0x401,0x402,0x403
      </minorOperation>
      <minorEvent>
        <!--ro, opt, string, 事件次类型, desc:参考事件上传宏定义, type为AccessControllerEvent, IDCardInfoEvent该字段必填-->0x01,0x02,0x03,0x04
      </minorEvent>
      <pictureURLType>
        <!--ro, opt, enum, 报警图片格式, subType:string, [binary#二进制,LocalURL#设备本地URL,cloudStorageURL#云存储URL], desc:不进行订阅,图片数据会按照设备默认行为;可以在事件信息中区分图片的传输方式-->binary
      </pictureURLType>
      <channels>
        <!--ro, opt, string, 1, desc:支持复选,用英文逗号隔开-->1,2,3,4
      </channels>
      <subStatusCode>
        <!--ro, req, enum, 订阅失败的错误码, subType:string, [activeNumMax#已达License允许接入最大设备数日,LicenseExpired#License过期], desc:和ISAPI协议错误保持一致;可参考错误码字典-->activeNumMax
      </subStatusCode>
      <minorType>
        <!--ro, opt, enum, 事件子类型, subType:string, [success#成功,failure#失败], desc:1、当type为CertificateCaptureEvent或UncertificateCompareEvent时有效。2、对于高铁站人员安检和酒店客人办理入住场景,真实存在只需要订阅成功或失败某一类的事件。3、当该字段不返回时,代表订阅成功和失败两种事件。-->success
      </minorType>
    </Event>
  </FailedEventList>
</SubscribeEventResponse>
```

69.5.2 设置布防订阅管理参数

Request URL

PUT /ISAPI/Event/notification/subscribeEvent/<subscribeEventID>

查询参数

参数名称	参数类型	描述
subscribeEventID	string	建立布防订阅成功(POST /ISAPI/Event/notification/subscribeEvent)时，设备返回SubscribeEventResponse中id字段。若建立布防订阅成功(POST /ISAPI/Event/notification/subscribeEvent)时，下发了subscribeEventID字段可替代SubscribeEventResponse中id字段使用。

请求报文

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<SubscribeEvent xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
```

```
<!--req, object, 该节点表示所有含图片事件图片上传方式, attr:version{req, string, 协议版本}-->
```

```
<heartbeat>
```

```
<!--opt, int, 心跳间隔时间, range:[1,180], unit:s, unitType:时间-->1
```

```
</heartbeat>
```

```
<channelMode>
```

```
<!--opt, enum, 订阅通道类型, subType:string, [aLL#所有通道订阅,List#按具体通道号列表进行订阅]-->list
```

```
</channelMode>
```

```
<eventMode>
```

```
<!--req, enum, 事件订阅模式, subType:string, [aLL#订阅设备的所有通道下目标的事件,List#按具体事件类型进行订阅]-->list
```

```
</eventMode>
```

```
<eventList>
```

```
<!--opt, array, 订阅事件类型列表, subType:object, desc: 当eventMode为List时生效-->
```

```
<Event>
```

```
<!--opt, object, 单一事件订阅-->
```

```
<type>
```

```
<!--req, enum, 事件类型, subType:string, [ADAS#高级辅助驾驶报警,ADASALarm#高级驾驶辅助报警,AID#交通事件检测,ANPR#车牌识别,AccessControlLerEvent#门禁主机检测事件,CDsStatus#光盘刻录状态上报,DBD#驾驶行为报警,GPSPUpload#GPS信息上传,HFPD#高频人员报警,IO#信号量报警,IOTD#非视频设备检测事件,LES#物流扫描事件,LFPD#低频人员检测,PALMismatch#视频制式不匹配,PIR#PIR,PeopleCounting#客流量统计,PeopleNumChange#人数异常,Standup#人员站立检测,TMA#测温报警,TMPA#测温预警,WMD#移动侦测,abnormalAcceleration#设备加速度异常报警,abnormalDriving#异常驾驶行为,advReachHeight#折线攀高检测,alarmResult#人脸比对,attendance#考勤事件,attendedBaggage#物品拿取,audioAbnormal#声强突变检测,audioexception#音频异常侦测,behaviorResult#行为分析结果上报,blindSpotDetection#盲区监测报警,cardMatch#认证比对,changedStatus#状态变化上报,collision#碰撞,containerDetection#集装箱检测,crowdSituationAnalysis#人群态势分析,databaseException#数据库异常,defocus#虚焦侦测,diskUnformat#磁盘未格式化,diskerror#硬盘错误,diskfull#硬盘满,driverConditionMonitor#驾驶员状态监测报警,emergencyAlarm#紧急报警,faceCapture#人脸抓拍识别,facesnapModeling#人脸抓拍建模,facedetection#人脸侦测,failDown#人员倒地(使用错误的拼写,faultAlarm#故障告警,fieldDetection#区域入侵,fireDetection#火点检测,fireEscapeDetection#消防通道检测,flowOverrun#流量超限,framesPeopleCounting#区域人数统计,getUp#起身检测,group#人员聚集,hdBadBlock#硬盘坏块,hdImpact#硬盘冲击,heatmap#热度图报警,highHDTemperature#硬盘温度过高,highTempAlarm#设备温度过高报警,hotSpare#热备异常,iLLaccess#非法访问,ipcTransferAbnormal#IPC迁移失败,ipConflict#IP地址冲突,keyPersonGetUp#重点人员起身检测,leavePosition#离岗检测,linedetection#越界侦测(穿越警戒线),ListSyncException#列表(名单)同步异常,Loitering#徘徊侦测,LowHDTemperature#硬盘温度过低,mixedTargetDetection#混合目标检测,modelError#模型错误,nicbroken#网络断开,nodeOffline#离线异常,nonPoliceIntrusion#非警察人员入侵,overSpeed#超速报警,overtimeTarry#超时滞留检测,parking#停车侦测,peopleNumChange#人数变化,peopleNumCounting#人数统计(区别于区域人数统计),personAbnormalAlarm#人员身份异常报警,personDensityDetection#人员密度,personQueueCounting#排队人数检测,personQueueDetection#排队检测,personQueueRealTime#人员排队检测实时数据上报,personQueueTime#排队时长检测,playCellphone#玩手机检测,pocException#视频异常,poe#POE供电异常,policeAbsent#警察不在场,radarAlarm#雷达告警,radarFieldDetection#雷达区域入侵,radarLineDetection#雷达越界侦测,radarPerimeterRule#雷达规则数据,radarTargetDetection#雷达目标数据,radarVideoDetection#雷视目标检测,raidException#阵列异常,rapidMove#快速移动,reachHeight#攀高检测,recordCycleAbnormal#录像周期不足,recordException#录像异常,regionEntrance#进入区域,regionExiting#离开区域,retention#人员滞留事件,rollOver#侧翻,running#人员奔跑,safetyHelmetsDetection#安全帽检测,sceneChangedetection#场景变更,sensorAlarm#角速度加速度报警,severeHDFailure#硬盘严重故障,shelttenAlarm#遮挡报警,shipsDetection#船只检测,sitQuietly#静坐检测,smokeAndFireDetection#烟火检测,smokeDetection#烟雾检测,softIO#报警软IO,spacingChange#间距异常,sysStorFull#集群系统存储满异常,takingElevatorDetection#乘梯检测(电瓶车进电梯检测),targetCapture#目标抓拍,temperature#温差报警,thermometry#温度报警,thirdPartyException#第三方异常,toiletTarry#如厕超时检测,tollCodeInfo#二维码信息上报,tossing#抛物,unattendedBaggage#物品遗留,vehicleMatchResult#黑白名单车辆报警上传,vehicleRecogResult#车辆图片二次识别,versionAbnormal#集群版本异常,videoException#视频异常,videoLoss#视频丢失,violationAlarm#违规告警,violentMotion#剧烈运动,yardTarry#放风场滞留检测,AccessControlLerEvent#门禁事件,IDCardInfoEvent#门禁身份证事件,FaceTemperatureMeasurementEvent#门禁测温事件,QRCODEEvent#门禁二维码事件,CertificateCaptureEvent#证件抓拍比对事件,UncertificateCompareEvent#无证比对事件,ConsumptionAndTransactionRecordEvent#消费和交易记录事件,ConsumptionEvent#消费事件,TransactionRecordEvent#交易记录事件,TFS#交通取证,SetMealQuery#消费套餐查询,ConsumptionStatusQuery#消费状态查询,certificateRevocation#证书失效,humanBodyComparison#人体比对,regionTargetNumberCounting#区域目标统计,paymentQRCodeQuery#查询三方支付二维码,ConsumptionQuery#本地消费查询事件,topUpRequest#充值请求,topUpStatusQuery#查询充值状态,topUpEvent#充值事件,LogOffRefundStatusQuery#查询人员注销退款状态,LogOffRefundRequest#人员注销退款请求], desc: 参考事件类型列表eventType-->mixedTargetDetection
```

```
</type>
```

```
<minorAlarm>
```

```
<!--opt, string, 报警次类型, desc: 参考事件上传宏定义;type为AccessControlLerEvent. IDCardInfoEvent该字段必填-->0x400,0x401,0x402,0x403
```

```
<minorAlarm>
```

```
<minorException>
```

```
<!--opt, string, 异常次类型, desc: 参考事件上传宏定义;type为AccessControlLerEvent. IDCardInfoEvent该字段必填-->0x400,0x401,0x402,0x403
```

```
<minorException>
```

```
<minorOperation>
```

```
<!--opt, string, 操作次类型, desc: 参考事件上传宏定义;type为AccessControlLerEvent. IDCardInfoEvent该字段必填-->0x400,0x401,0x402,0x403
```

```
<minorOperation>
```

```
<minorEvent>
```

```
<!--opt, string, 事件次类型, desc: 参考事件上传宏定义;type为AccessControlLerEvent. IDCardInfoEvent该字段必填-->0x01,0x02,0x03,0x04
```

```
</minorEvent>
```

```
<pictureURLType>
```

```
<!--opt, enum, 报警图片格式, subType:string, [binary#二进制,localURL#设备本地URL,cloudStorageURL#云存储URL]-->cloudStorageURL
```

```
</pictureURLType>
```

```
<channels>
```

```
<!--opt, string, 事件关联的通道信息, desc: 支持复选;用英文逗号隔开-->1,2,3,4
```

```
</channels>
```

```
<conferenceIDList>
```

```
<!--opt, array, 视频会议会议ID列表, subType:object, desc: 不包含该字段表示订阅所有会议;此处理解为某个事件支持上报的会议号-->
```

```
<conferenceID>
```

```
<!--opt, string, 订阅的视频会议ID号, range:[1,32]-->test
```

```
</conferenceID>
```

```
</conferenceIDList>
```

```
<uploadAllTarget>
```

```
<!--opt, bool, 是否上传所有检测目标, desc: 周组防范中用于NVR对IPC配置, 以便于NVR和获取到IPC检测到的所有目标, 默认为false-->>false
```

```
</uploadAllTarget>
```

```
<uploadMEFData>
```

```
<!--opt, bool, 是否上传事件关联的电围数据, desc: 默认为true. 电围数据上报关联智能事件上传, 由于电围模块采集数据间隔几分钟一次, 为使关联的电围数据中包
```

含某手机号信息，关联的电围数据需前后偏移一定时间，智能事件要在达到偏移时间后才能上报，由此事件会较触发时间延迟，布防订阅时关闭电围数据上传，一旦智能事件（人脸抓拍识别事件、混合目标检测事件等）触发，设备立即上报智能事件（不含电围数据）-->true

```
</uploadMEFData>
<minorType>
  <!--opt, enum, 事件子类型, subType:string, [success#成功,failed#失败],
desc:1. 当type为CertificateCaptureEvent或UncertificateCompareEvent时有效。
2、对于高铁站人员安检和酒店客人办理入住场景，真实存在只需要订阅成功或者失败某一类的事件。
3、当该字段不下发时，代表订阅成功和失败两种事件。-->success
</minorType>
</Event>
</EventList>
<channels>
  <!--opt, string, 通道统一布控, desc:如果该字段下发;那么Event中channels不做生效。如果布防全部通道;则不下发这个节点。布防部分通道时;列出通道号;支持复选;用英文逗号隔开.-->1,2,3,4
</channels>
<pictureURLType>
  <!--opt, enum, 报警图片格式统一配置, subType:string, [binary#二进制,LocalURL#设备本地URL,cloudStorageURL#云存储URL], desc:该节点表示所有含图片事件图片上传方;如果该字段下发;那么<Event>中<pictureURLType>不做生效.如果不下发这个节点;设备采用能力集返回的图片默认上传方式上报;前端设备默认为二进制;后端设备默认为设备本地URL.-->cloudStorageURL
</pictureURLType>
<ChangedUploadSub>
  <!--opt, object, 设备状态变化上报信息订阅-->
<interval>
  <!--opt, int, 布防guid的生命周期, range:[1,300], unit:s, unitType:时间, desc:默认5分钟;布防guid的生命周期;该间隔期间未过来重连;设备认为是新的一路布防;重新生成guid-->300
</interval>
<StatusSub>
  <!--opt, object, 状态信息-->
<all>
  <!--opt, bool, 是否订阅所有通道、磁盘、能力变化状态-->>false
</all>
<channel>
  <!--opt, bool, 订阅通道状态, desc:all为true时不必传入-->true
</channel>
<hd>
  <!--opt, bool, 订阅磁盘状态, desc:all为true时不必传入-->true
</hd>
<capability>
  <!--opt, bool, 订阅能力集变化状态, desc:all为true时不必传入-->true
</capability>
<ChanStatus>
  <!--ro, opt, object, 变化上报中通道状态订阅, desc:当channel为true时有效,无此节点表示所有都上报。场景:如录像状态上报太频繁,平台更关注设备在线状态和布防状态,所以想要取消对录像状态变化上报的订阅-->
<online>
  <!--ro, opt, bool, 订阅通道在线状态, desc:无此节点表示不要该报警-->true
</online>
<record>
  <!--ro, opt, bool, 订阅是否在录像, desc:无此节点表示不要该报警-->true
</record>
<recordStatus>
  <!--ro, opt, bool, 订阅是录像状态, desc:无此节点表示不要该报警-->true
</recordStatus>
<signal>
  <!--ro, opt, bool, 订阅信号状态, desc:无此节点表示不要该报警-->true
</signal>
<arming>
  <!--ro, opt, bool, 订阅NVR对前端设备的布防状态, desc:无此节点表示不要该报警-->true
</arming>
</ChanStatus>
</StatusSub>
</ChangedUploadSub>
<identityKey>
  <!--opt, string, 订阅连接交互口令, range:[1,64], desc:mickey通脑产品引入,支持针对指定导入的人脸库(导入中携带identityKey口令)的比对结果进行布控-->test
</identityKey>
<ConferenceIDList>
  <!--opt, array, 视频会议会议ID列表, subType:object-->
<ConferenceID>
  <!--opt, string, 订阅的视频会议ID号, range:[1,32]-->test
</ConferenceID>
</ConferenceIDList>
<level>
  <!--opt, enum, 布防等级, subType:string, [high#高,middle#中,low#低], desc:不填默认为high-->high
</level>
<middleLevelPictureEnabeld>
  <!--opt, bool, 中等级布防上传图片使能, desc:没有此节点默认为true-->true
</middleLevelPictureEnabeld>
</SubscribeEvent>
```

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 响应消息, attr:version{ro, req, string, 版本}-->
  <requestURL>
    <!--ro, req, string, 请求的URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, 状态码, subType:int, [0#OK,1#OK,2#Device Busy,3#Device Error,4#Invalid Operation,5#Invalid XML Format,6#Invalid XML Content,7#Reboot Required]-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, 状态信息, subType:string, [OK#成功,Device Busy#设备忙,Device Error#设备异常,Invalid Operation#无效的操作,Invalid XML Format#无效的XML格式,Invalid XML Content#无效的XML内容,Reboot#设备重启]-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, 详细错误码英文描述, desc:详细错误码的英文描述-->OK
  </subStatusCode>
</ResponseStatus>
```

69.5.3 获取布防订阅管理参数

Request URL

GET /ISAPI/Event/notification/subscribeEvent/<subscribeEventID>

查询参数

参数名称	参数类型	描述
subscribeEventID	string	为建立布防订阅成功(POST /ISAPI/Event/notification/subscribeEvent)时，设备返回SubscribeEventResponse中id字段。若建立布防订阅成功(POST /ISAPI/Event/notification/subscribeEvent)时，下发了subscribeEventID字段可替代SubscribeEventResponse中id字段使用。

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<SubscribeEvent xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 该节点表示所有含图片事件图片上传方式, attr:version{req, string, 协议版本}-->
  <heartbeat>
    <!--ro, opt, int, 心跳间隔时间, range:[1,180], unit:s, unitType:时间-->1
  </heartbeat>
  <channelMode>
    <!--ro, opt, enum, 订阅通道类型, subType:string, [all#所有通道订阅,List#按具体通道号列表进行订阅]-->list
  </channelMode>
  <eventMode>
    <!--ro, req, enum, 事件订阅模式, subType:string, [all#订阅设备的所有通道下目标的事件,List#按具体事件类型进行订阅]-->list
  </eventMode>
  <eventList>
    <!--ro, opt, array, 订阅事件类型列表, subType:object, desc:当eventMode为List时生效-->
    <Event>
      <!--ro, opt, object, 单一事件订阅-->
      <type>
        <!--ro, req, enum, 事件类型, subType:string, [ADAS#高级辅助驾驶报警,ADASAlarm#高级驾驶辅助报警,AID#交通事件检测,ANPR#车牌识别,AccessControllerEvent#门禁主机检测事件,CDsStatus#光盘刻录状态上报,DBD#驾驶行为报警,GPSUpload#GPS信息上传,HFPD#高频人员报警,IO#信号量报警,IOTD#非视频设备检测事件,LES#物流扫描事件,LFPD#低频人员检测,PALMismatch#视频制式不匹配,PIR#PIR,PeopleCounting#客流量统计,PeopleNumChange#人数异常,Standup#人员站立检测,TMA#测温报警,TMPA#测温预警,VMD#移动侦测,abnormalAcceleration#设备加速度异常报警,abnormalDriving#异常驾驶行为,advReachHeight#折线攀高检测,alarmResult#人脸比对,attendance#考勤事件,attendedBaggage#物品拿取,audioAbnormal#声强突变检测,audioException#音频异常检测,behaviorResult#行为分析结果上报,blindSpotDetection#盲区监测报警,cardMatch#认证比对,changedStatus#状态变化上报,collision#碰撞,containerDetection#集装箱检测,crowdSituationAnalysis#人群态势分析,databaseException#数据库异常,defocus#虚焦侦测,diskUnformat#磁盘未格式化,diskError#硬盘错误,diskFull#硬盘满,driverConditionMonitor#驾驶员状态监测报警,emergencyAlarm#紧急报警,faceCapture#人脸抓拍识别,faceSnapModeling#人脸抓拍建模,facedetection#人脸侦测,failDown#人员倒地(使用错误的拼写,faultAlarm#故障告警,fieldDetection#区域入侵,fireDetection#火点检测,fireEscapeDetection#消防通道检测,fLowOverrun#流量超限,framesPeopleCounting#区域人数统计,getUp#起身检测,group#人员聚集,hdBADBlock#硬盘坏块,hdImpact#硬盘冲击,heatmap#热度图报警,highHDTemperature#硬盘温度过高,highTempAlarm#设备温度过高报警,hotSpare#热备异常,iLLAccess#非法访问,ipcTransferAbnormal#IPC迁移失败,ipconflict#IP地址冲突,keyPersonGetUp#重点人员起身检测,leavePosition#离岗检测,linedetection#越界侦测(穿越警戒线),ListSyncException#列表(名单)同步异常,Loitering#徘徊侦测,LowHDTemperature#硬盘温度过低,mixedTargetDetection#混合目标检测,modelError#模型错误,nicBroken#网络断开,nodeOffline#掉线异常,nonPoliceIntrusion#非警察人员入侵,overSpeed#超速报警,overTimeTarry#超时滞留检测,parking#停车侦测,peopleNumChange#人数变化,peopleNumCounting#人数统计(区别于区域人数统计),personAbnormalAlarm#人员身份异常报警,personDensityDetection#人员密度,personQueueCounting#排队人数检测,personQueueDetection#排队检测,personQueueRealTime#人员排队检测实时数据上报,personQueueTime#排队时长检测,playCellPhone#玩手机检测,pocException#视频异常,poe#POE供电异常,policeAbsent#警察不在场,radarAlarm#雷达告警,radarFieldDetection#雷达区域入侵,radarLineDetection#雷达越界侦测,radarPerimeterRule#雷达规则数据,radarTargetDetection#雷达目标数据,radarVideoDetection#雷达视频目标检测,raidException#阵列异常,rapidMove#快速移动,reachHeight#攀高检测,recordCycleAbnormal#录像周期不足,recordException#录像异常,regionEntrance#进入区域,regionExiting#离开区域,retention#人员滞留事件,rollOver#侧翻,running#人员奔跑,safetyHelmetDetection#安全帽检测,scenechangedetection#场景变更,sensorAlarm#角速度加速度报警,severeHDFailure#硬盘严重故障,shelterAlarm#遮挡报警,shipsDetection#船只检测,sitQuietly#静坐检测,smokeAndFireDetection#烟火检测,smokeDetection#烟雾检测,softIO#报警软IO,spacingChange#间距异常,sysStorageFull#集群系统存储满异常,takingElevatorDetection#乘梯检测(电瓶车进电梯检测),targetCapture#目标抓拍,temperature#温差报警,thermometry#温度报警,thirdPartyException#第三方异常,tolLetTarry#如厕超时检测,tolLCodeInfo#二维码信息上报,tossing#抛物,unattendedBaggage#物品遗留,vehicleMatchResult#黑白名单车辆报警上
```

传,vehicleRcogResult#车辆图片二次识别,versionAbnormal#集群版本异常,videoException#视频异常,videoLoss#视频丢失,violationAlarm#违规告警,violentMotion#剧烈运动,yardTarry#放风场滞留检测,AccessControllerEvent#门禁事件,IDCardInfoEvent#门禁身份证事件,FaceTemperatureMeasurementEvent#门禁测温事件,QRCodeEvent#门禁二维码事件,CertificateCaptureEvent#证件抓拍比对事件,UncertificateCompareEvent#无证比对事件,ConsumptionAndTransactionRecordEvent#消费和交易记录事件,ConsumptionEvent#消费事件,TransactionRecordEvent#交易记录事件,TFS#交通取证,SetMealQuery#套餐消费查询,ConsumptionStatusQuery#消费状态查询,certificateRevocation#证书失效,humanBodyComparison#人体比对,regionTargetNumberCounting#区域目标统计,paymentQRCodeQuery#查询三方支付二维码,ConsumptionQuery#本地消费查询事件,topUpRequest#充值请求,topUpStatusQuery#查询充值状态,topUpEvent#充值事件,LogOffRefundStatusQuery#查询人员注销退款状态,LogOffRefundRequest#人员注销退款请求], desc:参考事件类型列表eventType-->mixedTargetDetection

```
</type>
<minorAlarm>
  <!--ro, opt, string, 报警次类型, desc:参考事件上传宏定义;type为AccessControllerEvent, IDCardInfoEvent 该字段必填-->0x400,0x401,0x402,0x403
</minorAlarm>
<minorException>
  <!--ro, opt, string, 异常次类型, desc:参考事件上传宏定义;type为AccessControllerEvent, IDCardInfoEvent 该字段必填-->0x400,0x401,0x402,0x403
</minorException>
<minorOperation>
  <!--ro, opt, string, 操作次类型, desc:参考事件上传宏定义;type为AccessControllerEvent, IDCardInfoEvent 该字段必填-->0x400,0x401,0x402,0x403
</minorOperation>
<minorEvent>
  <!--ro, opt, string, 事件次类型, desc:参考事件上传宏定义;type为AccessControllerEvent, IDCardInfoEvent 该字段必填-->0x01,0x02,0x03,0x04
</minorEvent>
<pictureURLType>
  <!--ro, opt, enum, 报警图片格式, subType:string, [binary#二进制,LocalURL#设备本地URL,cCloudStorageURL#云存储URL]-->cCloudStorageURL
</pictureURLType>
<channels>
  <!--ro, opt, string, 事件关联的通道信息, desc:支持复选;用英文逗号隔开-->1,2,3,4
</channels>
<ConferenceIDList>
  <!--ro, opt, array, 视频会议会议ID列表, subType:object, desc:不包含该字段表示订阅所有会议;此处理解为某个事件支持上报的会议号-->
  <ConferenceID>
    <!--ro, opt, string, 订阅的视频会议ID号, range:[1,32]-->text
  </ConferenceID>
</ConferenceIDList>
<uploadAllTarget>
  <!--ro, opt, bool, 是否上传所有检测目标, desc:周组防范中用于NVR对IPC配置, 以便于NVR和获取到IPC检测到的所有目标,默认为false-->>false
</uploadAllTarget>
<uploadMEFData>
  <!--ro, opt, bool, 是否上传事件关联的电围数据, desc:默认为true. 电围数据上报关联智能事件上传, 由于电围模块采集数据间隔几分钟一次, 为使关联的电围数据中包含某手机号信息, 关联的电围数据需前后偏移一定时间, 智能事件要在达到偏移时间后才能上报, 由此事件会较触发时间延迟, 布防订阅时关闭电围数据上传, 一旦智能事件(人脸抓拍识别事件、混合目标检测事件等)触发, 设备立即上报智能事件(不含电围数据)-->>true
</uploadMEFData>
<minorType>
  <!--ro, opt, enum, 事件子类型, subType:string, [success#成功,failed#失败],
desc:1. 当type为CertificateCaptureEvent或UncertificateCompareEvent时有效。
2. 对于高铁站人员安检和酒店客人为理入住场景, 真实存在只需要订阅成功或者失败某一类的事件。
3. 当该字段不下发时, 代表订阅成功和失败两种事件。-->success
</minorType>
</Event>
</EventList>
<channels>
  <!--ro, opt, string, 通道统一布控, desc:如果该字段下发;那么Event中channels不做生效。如果布防全部通道;则不下发这个节点。布防部分通道时;列出通道号;支持复选;用英文逗号隔开。-->1,2,3,4
</channels>
<pictureURLType>
  <!--ro, opt, enum, 报警图片格式统一配置, subType:string, [binary#二进制,LocalURL#设备本地URL,cCloudStorageURL#云存储URL], desc:该节点表示所有含图片事件图片上传方;如果该字段下发;那么<Event>中<pictureURLType>不做生效。如果不下发这个节点;设备采用能力集返回的图片默认上传方式上报;前端设备默认为二进制;后端设备默认为设备本地URL.-->cCloudStorageURL
</pictureURLType>
<ChangedUploadSub>
  <!--ro, opt, object, 设备状态变化上报信息订阅-->
<interval>
  <!--ro, opt, int, 布防guid的生命周期, range:[1,300], unit:s, unitType:时间, desc:默认5分钟;布防guid的生命周期;该间隔期间未过来重连;设备认为是新的一路布防;重新生成guid-->300
</interval>
<StatusSub>
  <!--ro, opt, object, 状态信息-->
  <all>
    <!--ro, opt, bool, 是否订阅所有通道、磁盘、能力变化状态-->>false
  </all>
  <channel>
    <!--ro, opt, bool, 订阅通道状态, desc:all为true时不必传入-->>true
  </channel>
  <hd>
    <!--ro, opt, bool, 订阅磁盘状态, desc:all为true时不必传入-->>true
  </hd>
  <capability>
    <!--ro, opt, bool, 订阅能力集变化状态, desc:all为true时不必传入-->>true
  </capability>
  <ChanStatus>
    <!--ro, opt, object, 变化上报中通道状态订阅, desc:当channel为true时有效,无此节点表示所有都上报。场景:如录像状态上报太频繁,平台更关注设备在线状态和布防状态, 所以想要取消对录像状态变化上报的订阅-->
    <online>
      <!--ro, opt, bool, 订阅通道在线状态, desc:无此节点表示不要该报警-->>true
    </online>
    <record>
      <!--ro, opt, bool, 订阅是否在录像, desc:无此节点表示不要该报警-->>true
    </record>
    <recordStatus>
      <!--ro, opt, bool, 订阅是录像状态, desc:无此节点表示不要该报警-->>true
    </recordStatus>
    <signal>
      <!--ro, opt, bool, 订阅信号状态, desc:无此节点表示不要该报警-->>true
    </signal>
    <arming>
      <!--ro, opt, bool, 订阅NVR对前端设备的布防状态, desc:无此节点表示不要该报警-->>true
    </arming>
```

```
</arming>
</ChanStatus>
</StatusSub>
</ChangedUploadSub>
<identityKey>
  <!--ro, opt, string, 订阅连接交互口令, range:[1,64], desc:mickey超脑产品引入, 支持针对指定导入的人脸库（导入中携带identityKey口令）的比对结果进行布控-->test
</identityKey>
<ConferenceIDList>
  <!--ro, opt, array, 视频会议会议ID列表, subType:object-->
  <ConferenceID>
    <!--ro, opt, string, 订阅的视频会议ID号, range:[1,32]-->test
  </ConferenceID>
</ConferenceIDList>
<level>
  <!--ro, opt, enum, 布防等级, subType:string, [high#高,middle#中,low#低], desc:不填默认为high-->high
</level>
<middleLevelPictureEnabeld>
  <!--ro, opt, bool, 中等级布防上传图片使能, desc:没有此节点默认为true-->true
</middleLevelPictureEnabeld>
</SubscribeEvent>
```

69.5.4 获取布防订阅能力

Request URL

GET /ISAPI/Event/notification/subscribeEventCap

查询参数

无

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>
<SubscribeEventCap xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 该节点表示所有含图片事件图片上传方式, attr:version{req, string, 协议版本}-->
  <heartbeat min="1" max="180">
    <!--ro, opt, int, 心跳间隔时间, range:[1,180], unit:s, unitType:时间, attr:min{req, int, 最小值},max{req, int, 最大值}-->1
  </heartbeat>
  <channelMode opt="all,list">
    <!--ro, opt, enum, 订阅通道类型, subType:string, [all#所有通道订阅,List#按具体通道号列表进行订阅], attr:opt{req, string, 取值范围}-->list
  </channelMode>
  <eventMode opt="all,list">
    <!--ro, req, enum, 事件订阅模式, subType:string, [all#订阅设备的所有通道下目标的事件,List#按具体事件类型进行订阅], attr:opt{req, string, 取值范围}-->list
  </eventMode>
  <EventList>
    <!--ro, opt, array, 订阅事件类型列表, subType:object, desc:当eventMode为List时生效-->
    <Event>
      <!--ro, opt, object, 单一事件订阅-->
      <type>
        <!--ro, req, enum, 事件类型, subType:string, [ADAS#高级辅助驾驶报警,ADASALarm#高级驾驶辅助报警,AID#交通事件检测,ANPR#车牌识别,AccessControllerEvent#门禁主机检测事件,CdsStatus#光盘刻录状态上报,DBD#驾驶行为报警,GPSUpload#GPS信息上传,HFPD#高频人员报警,IO#信号量报警,IOTD#非视频设备检测事件,LES#物流扫描事件,LFPD#低频人员检测,PALMismatch#视频制式不匹配,PIR#PIR,PeopleCounting#客流量统计,PeopleNumChange#人数异常,Standup#人员站立检测,TMA#测温报警,TMPA#测温预警,VMD#移动侦测,abnormalAcceleration#设备加速度异常报警,abnormalDriving#异常驾驶行为,advReachHeight#折线攀高检测,alarmResult#人脸比对,attendance#考勤事件,attendedBaggage#物品拿取,audioAbnormal#声强突变检测,audioException#音频异常侦测,behaviorResult#行为分析结果上报,blindSpotDetection#盲区监测报警,cardMatch#认证比对,changedStatus#状态变化上报,collision#碰撞,containerDetection#集装箱检测,crowdSituationAnalysis#人群态势分析,databaseException#数据库异常,defocus#虚焦侦测,diskUnformat#磁盘未格式化,diskerror#硬盘错误,diskFull#硬盘满,driverConditionMonitor#驾驶员状态监测报警,emergencyAlarm#紧急报警,faceCapture#人脸抓拍识别,faceSnapModeling#人脸抓拍建模,facedetection#人脸侦测,failDown#人员倒地（使用错误的拼写,faultAlarm#故障告警,fielddetection#区域入侵,fireDetection#火点检测,fireEscapeDetection#消防通道检测,flowOverrun#流量超限,framesPeopleCounting#区域人数统计,getUp#起身检测,group#人员聚集,hdBadBlock#硬盘坏块,hdImpact#硬盘冲击,heatmap#热度图报警,highHDTemperature#硬盘温度过高,highTempAlarm#设备温度过高报警,hotSpare#热备异常,iLLAccess#非法访问,ipcTransferAbnormal#IPC迁移失败,ipconflict#IP地址冲突,keyPersonGetUp#重点人员起身检测,leavePosition#离岗检测,linedetection#越界侦测（穿越警戒线）,ListSyncException#列表（名单）同步异常,Loitering#徘徊侦测,LowHDTemperature#硬盘温度过低,mixedTargetDetection#混合目标检测,modelError#模型错误,nicbroken#网络断开,nodeOffline#掉线异常,nonPoliceIntrusion#非警察人员入侵,overSpeed#超速报警,overtimeTarry#超时滞留检测,parking#停车侦测,peopleNumChange#人数变化,peopleNumCounting#人数统计（区别于区域人数统计）,personAbnormalAlarm#人员身份异常报警,personDensityDetection#人员密度,personQueueCounting#排队人数检测,personQueueDetection#排队检测,personQueueRealTime#人员排队检测实时数据上报,personQueueTime#排队时长检测,pLayCellphone#玩手机检测,pocException#视频异常,poe#POE供电异常,policeAbsent#警察不在场,radarAlarm#雷达告警,radarFieldDetection#雷达区域入侵,radarLineDetection#雷达越界侦测,radarPerimeterRule#雷达规则数据,radarTargetDetection#雷达目标数据,radarVideoDetection#雷视目标检测,raidException#阵列异常,rapidMove#快速移动,reachHeight#攀高检测,recordCycleAbnormal#录像周期不足,recordException#录像异常,regionEntrance#进入区域,regionExiting#离开区域,retention#人员滞留事件,rollOver#侧翻,running#人员奔跑,safetyHelmetDetection#安全帽检测,scenechangedetection#场景变更,sensorAlarm#角速度加速度报警,severeHdFailure#硬盘严重故障,shelterAlarm#遮挡报警,shipsDetection#船只检测,sitQuietly#静坐检测,smokeAndFireDetection#烟火检测,smokeDetection#烟雾检测,softIO#报警软IO,spacingChange#间距异常,sysStorFull#集群系统存储满异常,takingElevatorDetection#乘梯检测（电瓶车进电梯检测）,targetCapture#目标抓拍,temperature#温差报警,thermometry#温度报警,thirdPartyException#第三方异常,tolLetTarry#如厕超时检测,tollCodeInfo#二维码信息上报,tossing#抛物,unattendedBaggage#物品遗留,vehicleMatchResult#黑白名单车辆报警上传,vehicleRecogResult#车辆图片二次识别,versionAbnormal#集群版本异常,videoException#视频异常,videoLoss#视频丢失,violationAlarm#违规告警,violentMotion#剧烈运动,yardTarry#放风场滞留检测,AccessControllerEvent#门禁事件,IDCardInfoEvent#门禁身份证事件,FaceTemperatureMeasurementEvent#门禁测温事件,QRCodeEvent#门禁二维码事件,CertificateCaptureEvent#证件抓拍比对事件,UncertificateCompareEvent#无证比对事件,ConsumptionAndTransactionRecordEvent#消费和交易记录事件,ConsumptionEvent#消费事件,TransactionRecordEvent#交易记录事件,HealthInfoSyncQuery#健康信息查询事件,TFS#交通取证,SetMealQuery#消费套餐查询,ConsumptionStatusQuery#消费状态查询,certificateRevocation#证书失效,humanBodyComparison#人体比对,regionTargetNumberCounting#区域目标统计,cabinetChargingQuery#智能柜计费查询请求,paymentQRCodeQuery#查询第三方支付二维码,ConsumptionQuery#本地消费查询事件,topUpRequest#充值请求,topUpStatusQuery#查询充值状态,topUpEvent#充值事件,LogOffRefundStatusQuery#查询人员注销退款状态,LogOffRefundRequest#人员注销退款请求,personQueueTimingStatistics#人数统计定时上报], desc:参考事件类型列表eventType-->mixedTargetDetection
      </type>
    </Event>
  </EventList>
</SubscribeEventCap>
```

```
<minorAlarm opt="0x400,0x401,0x402,0x403">
  <!--ro, opt, string, 报警次类型, attr:opt{req, string, 取值范围}, desc:type为AccessControlllerEvent, IDCardInfoEvent 该字段必填-->0x400,0x401
</minorAlarm>
<minorException opt="0x400,0x401,0x402,0x403">
  <!--ro, opt, string, 异常次类型, attr:opt{req, string, 取值范围}, desc:type为AccessControlllerEvent, IDCardInfoEvent 该字段必填-->0x400,0x401
</minorException>
<minorOperation opt="0x400,0x401,0x402,0x403">
  <!--ro, opt, string, 操作次类型, attr:opt{req, string, 取值范围}, desc:type为AccessControlllerEvent, IDCardInfoEvent 该字段必填-->0x400,0x401
</minorOperation>
<minorEvent opt="0x01,0x02,0x03,0x04">
  <!--ro, opt, string, 事件次类型, attr:opt{req, string, 取值范围}, desc:type为AccessControlllerEvent, IDCardInfoEvent 该字段必填-->0x400,0x401
</minorEvent>
<pictureURLType opt="binary,localURL,cloudStorageURL" def="cloudStorageURL">
  <!--ro, opt, enum, 报警图片格式, subType:string, [binary#二进制,LocalURL#设备本地URL,cloudStorageURL#云存储URL], attr:opt{req, string, 取值范围},def{req, string, 默认值}->cloudStorageURL
</pictureURLType>
<channels>
  <!--ro, opt, string, 事件关联的通道信息, desc:支持复选;用英文逗号隔开-->1,2,3,4
</channels>
<ConferenceIDList size="1">
  <!--ro, opt, array, 视频会议会议ID列表, subType:object, attr:size{req, int, 数组元素总数}, desc:不包含该字段表示订阅所有会议;此处理解为某个事件支持上报的会议号-->
    <ConferenceID min="1" max="32">
      <!--ro, opt, string, 订阅的视频会议ID号, range:[1,32], attr:min{req, int, 最小值},max{req, int, 最大值}->test
    </ConferenceID>
  </ConferenceIDList>
<uploadAllTarget opt="true,false">
  <!--ro, opt, bool, 是否上传所有检测目标, attr:opt{req, string, 取值范围}, desc:周姐防淹中用于NVR对IPC配置, 以便于NVR和获取到IPC检测到的所有目标,默认为false-->>false
</uploadAllTarget>
<uploadMEFData opt="true,false">
  <!--ro, opt, bool, 是否上传事件关联的电围数据, attr:opt{req, string, 取值范围}, desc:默认为true.电围数据上报关联智能事件上传, 由于电围模块采集数据间隔几分钟一次, 为使关联的电围数据中包含某手机号信息, 关联的电围数据需前后偏移一定时间, 智能事件要在达到偏移时间后才能上报, 由此事件会较触发时间延迟, 布防订阅时关闭电围数据上传, 一旦智能事件(人脸抓拍识别事件、混合目标检测事件等)触发, 设备立即上报智能事件(不含电围数据)-->true
</uploadMEFData>
<minorType opt="success,failed">
  <!--ro, opt, string, 事件子类型, attr:opt{req, string, 取值范围},
desc:1. 当type为CertificateCaptureEvent或UncertificateCompareEvent时有效。
2. 对于高铁站人员安检和酒店客人办理入住场景, 真实存在只需要订阅成功或者失败某一类的事件。-->success
</minorType>
</Event>
</EventList>
<channels>
  <!--ro, opt, string, 通道统一布控, desc:如果该字段下发;那么Event中channels不做生效。如果布防全部通道;则不下发这个节点。布防部分通道时;列出通道号;支持复选;用英文逗号隔开.-->1,2,3,4
</channels>
<pictureURLType opt="binary,localURL,cloudStorageURL" def="cloudStorageURL">
  <!--ro, opt, enum, 报警图片格式统一配置, subType:string, [binary#二进制,LocalURL#设备本地URL,cloudStorageURL#云存储URL], attr:opt{req, string, 取值范围},def{req, string, 默认值}, desc:该节点表示所有含图片事件图片上传方;如果该字段下发;那么<Event>中<pictureURLType>不做生效.如果不下发这个节点;设备采用能力集返回的图片默认上传方式上报;前端设备默认为一进制;后端设备默认为设备本地URL.-->cloudStorageURL
</pictureURLType>
<ChangedUploadSub>
  <!--ro, opt, object, 设备状态变化上报信息订阅-->
    <interval>
      <!--ro, opt, int, 布防guid的生命周期, range:[1,300], unit:s, unitType:时间, desc:默认5分钟;布防guid的生命周期;该间隔期间未过来重连;设备认为是新的一路布防;重新生成guid-->300
    </interval>
    <StatusSub>
      <!--ro, opt, bool, 状态信息-->true
      <all>
        <!--ro, opt, bool, 是否订阅所有通道、磁盘、能力变化状态-->>false
      </all>
      <channel>
        <!--ro, opt, bool, 订阅通道状态, desc:all为true时不必传入-->true
      </channel>
      <hd>
        <!--ro, opt, bool, 订阅磁盘状态, desc:all为true时不必传入-->true
      </hd>
      <capability>
        <!--ro, opt, bool, 订阅能力集变化状态, desc:all为true时不必传入-->true
      </capability>
    </StatusSub>
    <ChanStatus>
      <!--ro, opt, object, 变化上报中通道状态订阅, desc:当channel为true时有效,无此节点表示所有都上报。场景:如录像状态上报太频繁, 平台更关注设备在线状态和布防状态, 所以想要取消对录像状态变化上报的订阅-->
        <online opt="true,false">
          <!--ro, opt, bool, 订阅通道在线状态, attr:opt{req, string, 取值范围}, desc:无此节点表示不要该报警-->true
        </online>
        <record opt="true,false">
          <!--ro, opt, bool, 订阅是否在录像, attr:opt{req, string, 取值范围}, desc:无此节点表示不要该报警-->true
        </record>
        <recordStatus opt="true,false">
          <!--ro, opt, bool, 订阅是录像状态, attr:opt{req, string, 取值范围}, desc:无此节点表示不要该报警-->true
        </recordStatus>
        <signal opt="true,false">
          <!--ro, opt, bool, 订阅信号状态, attr:opt{req, string, 取值范围}, desc:无此节点表示不要该报警-->true
        </signal>
        <arming opt="true,false">
          <!--ro, opt, bool, 订阅NVR对前端设备的布防状态, attr:opt{req, string, 取值范围}, desc:无此节点表示不要该报警-->true
        </arming>
      </ChanStatus>
    </StatusSub>
  </ChangedUploadSub>
  <identityKey max="64">
    <!--ro, opt, string, 订阅连接交互口令, range:[1,64], attr:max{req, int, 最大值}, desc:micru超脑产品引入, 支持针对指定导入的人脸库(导入中携带
```

```

identityKey口令)的比对结果进行布控-->test
</identityKey>
<ConferenceIDList size="1">
  <!--ro, opt, array, 视频会议会议ID列表, subType:object, attr:size{req, int, 数组元素总数}-->
  <ConferenceID min="1" max="32">
    <!--ro, opt, string, 订阅的视频会议ID号, range:[1,32], attr:min{req, int, 最小值},max{req, int, 最大值}-->test
  </ConferenceID>
</ConferenceIDList>
<isSupportModifySubscribeEvent>
  <!--ro, opt, bool, 是否支持布防订阅管理, desc:对应接口/ISAPI/Event/notification/subscribeEvent/<ID>-->true
</isSupportModifySubscribeEvent>
<subscribeEventID min="1" max="128">
  <!--ro, opt, string, 自定义订阅ID号;平台下发时确保其唯一性;下发该字段后设备默认开启全事件断网续传, range:[1,128], attr:min{req, int, 最小值},max{req, int, 最大值}, desc:建立布防订阅成功(POST /ISAPI/Event/notification/subscribeEvent)时;下发该字段后;可替代SubscribeEventResponse中返回的id字段使用-->test
</subscribeEventID>
<level opt="high,middle,low">
  <!--ro, opt, enum, 布防等级, subType:string, [high#高,middle#中,low#低], attr:opt{req, string, 取值范围}, desc:不填默认为high-->high
</level>
<middleLevelPictureEnabeld opt="true,false">
  <!--ro, opt, bool, 中等级布防上传图片使能, attr:opt{req, string, 取值范围}, desc:没有此节点默认为true-->true
</middleLevelPictureEnabeld>
<deployID opt="1,2">
  <!--ro, opt, enum, 布防类型, subType:int, [0#客户端布防,1#实时布防], attr:opt{req, string, 取值范围}-->0
</deployID>
</SubscribeEventCap>

```

69.5.5 事件订阅心跳上传

事件标识: heartBeat

```

<?xml version="1.0" encoding="UTF-8"?>
<EventNotificationAlert xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, 报警消息体, attr:version{opt, string, 未定义}-->
  <ipAddress>
    <!--ro, req, string, 报警设备IPv4地址-->172.6.64.7
  </ipAddress>
  <ipv6Address>
    <!--ro, opt, string, 报警设备IPv6地址-->1080:0:0:0:8:800:200C:417A
  </ipv6Address>
  <portNo>
    <!--ro, opt, int, 报警设备端口号-->80
  </portNo>
  <protocol>
    <!--ro, opt, enum, 传输通信协议类型, subType:string, [HTTP#HTTP,HTTPS#HTTPS,EHome#EHome], desc:传输通信协议HTTP/HTTPS/EHome(萤石透传ISAPI的时候赋值HTTP;ISUP透传ISAPI的时候赋值EHome)-->HTTP
  </protocol>
  <macAddress>
    <!--ro, opt, string, MAC地址-->01:17:24:45:D9:F4
  </macAddress>
  <channelID>
    <!--ro, opt, int, 触发报警的设备通道号, desc:触发的视频通道号-->1
  </channelID>
  <dateTime>
    <!--ro, req, datetime, 报警触发时间-->2004-05-03T17:30:08+08:00
  </dateTime>
  <activePostCount>
    <!--ro, opt, int, 同一个报警已经上传的次数, desc:事件触发频次-->1
  </activePostCount>
  <eventType>
    <!--ro, req, string, 事件类型-->heartBeat
  </eventType>
  <eventState>
    <!--ro, req, enum, 事件状态, subType:string, [active#有效事件,inactive#无效事件], desc:针对持续性事件-->active
  </eventState>
  <eventDescription>
    <!--ro, req, string, 事件描述-->heartBeat
  </eventDescription>
  <channelName>
    <!--ro, opt, string, 通道名称, range:[1,64]-->test
  </channelName>
  <deviceID>
    <!--ro, opt, string, 设备ID, desc:EHome报警中需返回, 例如test0123(EHome2.0、EHome4.0、ISUP5.0)-->12345
  </deviceID>
</EventNotificationAlert>

```

69.6 系统维护

69.6.1 获取设备系统能力(json格式)

Request URL

GET /ISAPI/System/capabilities?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "statusCode": 1,
  /*ro, opt, int, 状态码, desc:无法用1表示时（1表示成功且无特殊状态）必须返回*/
  "statusString": "OK",
  /*ro, opt, string, 状态描述, range:[1,64], desc:无法用OK表示时（OK表示成功且无特殊状态）必须返回*/
  "subStatusCode": "ok",
  /*ro, opt, string, 子状态码, range:[1,64], desc:无法用ok表示（ok表示成功且无特殊状态）必须返回*/
  "errorCode": 1,
  /*ro, req, int, 错误码, desc:当statusCode不为1时，与subStatusCode对应*/
  "errorMsg": "ok",
  /*ro, req, string, 错误信息, desc:当statusCode不为1时，解释信息在协议约束中，允许设备在后续的版本迭代中，进行优化丰富提升（不限制死）*/
  "isSupportBlackFDControl": true,
  /*ro, opt, bool, 是否支持名单管理*/
  "isSupportExecuteControl": true,
  /*ro, opt, bool, 是否支持布控管理*/
  "isSupportEventUpload": true,
  /*ro, opt, bool, 是否支持事件上报*/
  "isSupportSDKServer": true,
  /*ro, opt, bool, 是否支持SDK服务配置*/
  "isSupportAI": true,
  /*ro, opt, bool, 是否支持AI算法相关功能, desc:表示应用是否支持AI算法管理、AI任务分析、AI事件上报,web也用于控制UI的显示和隐藏,boolean类型,智能应用服务器V1.2.0增加*/
  "isSupportBehavior": true,
  /*ro, opt, bool, 是否支持行为分析算法相关功能, desc:表示应用是否支持行为分析算法管理、任务分析、事件上报,web也用于控制UI的显示和隐藏,boolean类型,行为分析4.0增加*/
  "isSupportAlgorithm": true,
  /*ro, opt, bool, 是否支持算法仓库, desc:本节点为后补节点(智能应用服务器V1.2.0),用于web显示/隐藏算法仓库,本节点不存在需要获取/ISAPI/Intelligent/algorithm/capabilities?format=json判断设备是否真正支持算法仓库*/
  "isSupportHttpHosts": true,
  /*ro, opt, bool, 是否支持报警主机配置, desc:对应接口/ISAPI/Event/notification/httpHosts/capabilities?format=json,深思3.3.10和脸谱1.4.0后补能力,只给web显示功能用(内部字段)*/
  "isSupportDST": true,
  /*ro, opt, bool, 是否支持夏令时*/
  "isSupportEhome": true,
  /*ro, opt, bool, 是否支持ISUP功能*/
}
```

69.6.2 获取视频能力

Request URL

GET /ISAPI/System/Video/capabilities

查询参数

无

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<VideoCap xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 视频能力, attr:version{req, string, 协议版本}-->
  <videoInputPortNums>
    <!--ro, opt, int, 视频输入端口个数-->0
  </videoInputPortNums>
  <videoOutputPortNums>
    <!--ro, opt, int, 视频输出端口个数-->0
  </videoOutputPortNums>
  <menuNums>
    <!--ro, opt, int, 支持显示本地菜单数目-->0
  </menuNums>
  <externalChannelNum>
    <!--ro, opt, bool, 外扩模拟通道个数-->true
  </externalChannelNum>
  <isSupportHeatmap>
    <!--ro, opt, bool, 是否支持热度图-->true
  </isSupportHeatmap>
  <isSupportCounting>
    <!--ro, opt, bool, 是否支持人数统计-->true
  </isSupportCounting>
  <countingType>
```

```

    <!--ro, opt, enum, 统计类型, subType:string, [human#表示的是客流统计,object#表示的是过线计数],
desc:human-表示的是客流统计; object-表示的是过线计数;
都支持以逗号隔开, eg:"human,object"-->human
</countingType>
<isSupportPicture>
    <!--ro, opt, bool, 是否支持OSD图片叠加, desc:对应url:/ISAPI/System/Video/inputs/channels/<channelID>/image/picture-->true
</isSupportPicture>
<isSupportPreviewSwitch>
    <!--ro, opt, bool, 是否支持预览切换-->true
</isSupportPreviewSwitch>
<isSupportRecodStatus>
    <!--ro, opt, bool, 是否支持录像状态查询-->true
</isSupportRecodStatus>
<isSupportPrivacyMask>
    <!--ro, opt, bool, 是否支持隐私遮蔽功能-->true
</isSupportPrivacyMask>
<isSupportBinocularPreviewSwitch>
    <!--ro, opt, bool, 是否支持双目预览切换-->true
</isSupportBinocularPreviewSwitch>
<isSupportCalibCheck>
    <!--ro, opt, bool, 是否支持标定检验-->true
</isSupportCalibCheck>
<isSupportPIP>
    <!--ro, opt, bool, 是否支持画中画, desc:PIP(picture in picture, 画中画)-->true
</isSupportPIP>
<isSupportFocusVideoMode>
    <!--ro, opt, bool, 是否支持聚焦视频模式, desc:用于安装时调试不同模式的对焦-->true
</isSupportFocusVideoMode>
<isSupportExternalChannel>
    <!--ro, opt, bool, 是否支持外扩模拟通道-->true
</isSupportExternalChannel>
<isSupportMultiChannelCounting>
    <!--ro, opt, bool, 是否支持多通道客流量查询-->true
</isSupportMultiChannelCounting>
<isSupportCountingCollection>
    <!--ro, opt, bool, 是否支持客流量补录-->true
</isSupportCountingCollection>
<isSupportHeatmapCollection>
    <!--ro, opt, bool, 是否支持热度图数据补录-->true
</isSupportHeatmapCollection>
<channelFlexible opt="name,enable,online,linknum">
    <!--ro, opt, enum, 灵活获取通道信息的条件能力, subType:string, [name#名称,enable#使能,online#在线状态,linknum#连接数目], attr:opt{req, string, 取值范围}, desc:对应ISAPI/System/Video/inputs/channels?format=json中的InputParam内容-->name
</channelFlexible>
<isSupportOutputsResource>
    <!--ro, opt, bool, 是否支持设备输出使能配置, desc:对应/ISAPI/System/Video/outputs/resource?format=json-->true
</isSupportOutputsResource>
<OSDLanguage opt="GBK,EUC-KR,Hebrew" def="GBK">
    <!--ro, opt, enum, OSD语言选择, subType:string, [GBK#中文,EUC-KR#韩文,Hebrew#希伯来语], attr:opt{req, string, 取值范围}, def{req, string, 默认值}-->GBK
</OSDLanguage>
<isSupportMixedChannel>
    <!--ro, opt, bool, 是否支持模式通道合并-->true
</isSupportMixedChannel>
<isSupportMixedChannelStatus>
    <!--ro, opt, bool, 是否支持模式通道状态合并-->true
</isSupportMixedChannelStatus>
<isSupportOutputCourseware>
    <!--ro, opt, bool, 是否支持课件输出配置, desc:/ISAPI/System/Video/outputs/courseware/capabilities?format=json-->true
</isSupportOutputCourseware>
<isSupportVideoInputMode>
    <!--ro, opt, bool, 是否支持模拟通道模式配置功能, desc:/ISAPI/System/Video/inputs/mode/capabilities-->true
</isSupportVideoInputMode>
<isSupportVideoOutputMode>
    <!--ro, opt, bool, 是否支持视频输出模式配置, desc:/ISAPI/System/Video/outputs/mode/capabilities?format=json-->true
</isSupportVideoOutputMode>
<PreviewMode>
    <!--ro, opt, object, 双路输出时web显示画中画, desc:isSupportVideoOutputMode节点返回true时,该功能支持-->
<PIPType>
    <!--ro, opt, object, 画中画模式显示-->
<mainScreenChannelID>
    <!--ro, opt, int, 画中画显示主预览画面通道号-->0
</mainScreenChannelID>
<subScreenChannelID>
    <!--ro, opt, int, 画中画显示次预览画面通道号-->0
</subScreenChannelID>
</PIPType>
</PreviewMode>
<isSupportViewTag>
    <!--ro, opt, bool, 是否支持视图标签, desc:不支持不返回-->true
</isSupportViewTag>
<isSupportMenuStatus>
    <!--ro, opt, bool, 是否支持菜单状态获取, desc:对应接口: /ISAPI/System/Video/Menu/<indexID>/status?format=json,indexID的范围来自于VideoCap.menuNums-->true
</isSupportMenuStatus>
</VideoCap>

```

69.6.3 导入设备配置文件

Request URL

POST /ISAPI/System/configurationData?security= <security> &iv= <iv> &secretkey= <secretkey>

查询参数

参数名称	参数类型	描述
security	string	没有security表示数据不加密； security=1,表示secretkey字段AES128加密； security=2,表示secretkey字段AES256加密；
iv	string	AES加密时的初始化向量参数， 在security=1或者2时为必须参数;
secretkey	string	密钥， security对其生效。

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 响应消息, attr:version{ro, req, string, 版本}-->
  <requestURL>
    <!--ro, req, string, 请求的URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, 状态码, subType:int, [0#OK,1#OK,2#Device Busy,3#Device Error,4#Invalid Operation,5#Invalid XML Format,6#Invalid XML Content,7#Reboot Required]-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, 状态信息, subType:string, [OK#成功,Device Busy#设备忙,Device Error#设备异常,Invalid Operation#无效的操作,Invalid XML Format#无效的XML格式,Invalid XML Content#无效的XML内容,Reboot#设备重启]-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, 详细错误码英文描述, desc:详细错误码的英文描述-->OK
  </subStatusCode>
</ResponseStatus>
```

69.6.4 导出设备配置文件

Request URL

GET /ISAPI/System/configurationData?security= <security> &iv= <iv> &secretkey= <secretkey>

查询参数

参数名称	参数类型	描述
security	string	没有security表示数据不加密； security=1,表示secretkey字段AES128加密； security=2,表示secretkey字段AES256加密；
iv	string	AES加密时的初始化向量参数， 在security=1或者2时为必须参数;
secretkey	string	密钥， security对其生效。

请求报文

无

响应报文

无

69.6.5 获取系统安全能力

Request URL

GET /ISAPI/Security/capabilities

查询参数

无

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>
<SecurityCap xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 系统安全能力, attr:version{req, string, 协议版本}-->
  <supportUserNums>
    <!--ro, opt, int, 支持用户最大数-->0
  </supportUserNums>
  <userBondIpNums>
    <!--ro, opt, int, 支持绑定IP最大数-->0
  </userBondIpNums>
  <userBondMacNums>
    <!--ro, opt, int, 支持绑定MAC地址最大数-->0
  </userBondMacNums>
  <isSupCertificate>
    <!--ro, opt, bool, 是否支持认证-->true
  </isSupCertificate>
  <issupIllegalLoginLock>
    <!--ro, opt, bool, 是否支持登陆锁定-->true
  </issupIllegalLoginLock>
  <isSupportOnlineUser>
    <!--ro, opt, bool, 是否支持在线用户配置-->true
  </isSupportOnlineUser>
  <isSupportAnonymous>
    <!--ro, opt, bool, 是否支持匿名登陆-->true
  </isSupportAnonymous>
  <isSupportStreamEncryption>
    <!--ro, opt, bool, 是否支持码流加密-->true
  </isSupportStreamEncryption>
  <securityVersion opt="1,2">
    <!--ro, opt, int, 表示加密能力集, attr:opt{req, string, 取值范围}, desc:每一个版本包括两个方面: 加密算法及加密的节点范围。目前1表示AES128加密, 2表示AES256加密, 加密范围在各个协议中描述-->1
  </securityVersion>
  <keyIterateNum>
    <!--ro, opt, int, 秘钥迭代次数, dep:or,{$.SecurityCap.securityVersion,eg,1},{$.SecurityCap.securityVersion,eg,2}, desc:depend on securityVersion,取值一般是100~1000-->100
  </keyIterateNum>
  <isSupportUserCheck>
    <!--ro, opt, bool, 是否支持用户参数修改(修改/增加/删除)时进行登录密码的校验, dep:or,{$.SecurityCap.securityVersion,eg,0},{$.SecurityCap.securityVersion,eg,1}, desc:增加能力,表示是否支持用户参数修改(修改/增加/删除)时进行登录密码的校验,此功能只在支持敏感信息加密的版本上有效,即depend on securityVersion-->true
  </isSupportUserCheck>
  <isSupportGUIDFileDataExport>
    <!--ro, opt, bool, 表示是否支持设备GUID文件导出, desc:对应的配置接口:/ISAPI/Security/GUIDFileData-->true
  </isSupportGUIDFileDataExport>
  <isSupportSecurityQuestionConfig>
    <!--ro, opt, bool, 表示是否支持回答安全问题, desc:对应的配置接口:/ISAPI/Security/questionConfiguration-->true
  </isSupportSecurityQuestionConfig>
  <isSupportGetOnlineUserListSC>
    <!--ro, opt, bool, 是否支持在线用户列表查询, desc:对应的配置接口:/ISAPI/Security/onLineUser-->true
  </isSupportGetOnlineUserListSC>
  <SecurityLimits>
    <!--ro, opt, object, 参数安全限制能力-->
    <LoginPasswordLenLimit min="1" max="16">
      <!--ro, opt, string, 用户登陆密码长度限制, attr:min{req, int, 最小值},max{req, int, 最大值}-->1
    </LoginPasswordLenLimit>
    <SecurityAnswerLenLimit min="1" max="128">
      <!--ro, opt, string, 安全问题答案长度限制, attr:min{req, int, 最小值},max{req, int, 最大值}-->1
    </SecurityAnswerLenLimit>
  </SecurityLimits>
  <RSAKeyLength opt="512,1024,2048" def="2048">
    <!--ro, opt, enum, HTTPS证书长度, subType:int, [512#512,1024#1024,2048#2048], attr:opt{req, string, 取值范围},def{req, string, 默认值}-->2048
  </RSAKeyLength>
  <isSupportONVIFUserManagement>
    <!--ro, opt, bool, ONVIF协议用户管理-->true
  </isSupportONVIFUserManagement>
  <WebCertificateCap>
    <!--ro, opt, object, HTTP认证能力, desc:该节点不返回则表示设备默认支持basic,digest认证-->
    <certificateType opt="basic,digest,digest/basic">
      <!--ro, req, enum, 证书类型, subType:string, [basic#basic认证,digest#digest认证,digest/basic#digest/basic认证], attr:opt{req, string, 取值范围}-->basic
    </certificateType>
  </WebCertificateCap>
  <isSupportConfigFileImport>
    <!--ro, opt, bool, 是否支持配置文件安全导入, desc:true支持返回,不支持不返回-->true
  </isSupportConfigFileImport>
  <isSupportConfigFileExport>
    <!--ro, opt, bool, 是否支持配置文件安全导出, desc:true支持返回,不支持不返回-->true
  </isSupportConfigFileExport>
  <cfgFileSecretKeyLenLimit min="0" max="16">
    <!--ro, opt, int, 配置文件校验密码长度限制, attr:min{req, int, 最小值},max{req, int, 最大值}-->1
  </cfgFileSecretKeyLenLimit>
  <isIrreversible>
    <!--ro, opt, bool, 设备是否支持密码不可逆存储, desc:不支持则代表设备端会存用户信息的明文密码,支持则代表设备端密码散列存储-->true
  </isIrreversible>
  <salt>
    <!--ro, opt, string, 用于用户登录时所使用的具体盐值-->test
  </salt>
```

```
</isSupportDeviceCertificatesManagement>
  <!--ro, opt, bool, 是否支持设备证书管理, desc: 该节点不返回表示不支持-->true
</isSupportDeviceCertificatesManagement>
<isSupportSecurityEmail>
  <!--ro, opt, bool, 是否支持安全邮箱配置, desc: 不返回表示不支持-->true
</isSupportSecurityEmail>
<isSupportEncryptCertificate>
  <!--ro, opt, bool, 是否支持证书加密, desc: /ISAPI/Security/deviceCertificate-->true
</isSupportEncryptCertificate>
<isSupportCertificateCustomID>
  <!--ro, opt, bool, 是否支持证书配置使用用户自定义ID-->true
</isSupportCertificateCustomID>
</SecurityCap>
```

69.6.6 配置设备语言

Request URL

PUT /ISAPI/System/DeviceLanguage

查询参数

无

请求报文

```
<?xml version="1.0" encoding="UTF-8"?>

<DeviceLanguage xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, 设备语言, attr:version{req, string, 协议版本}-->
  <language>
    <!--req, enum, 语言, subType:string, [SimChinese#简体中文,TraChinese#繁体中文,English#英文,Russian#俄语,Bulgarian#保加利亚语,Hungarian#匈牙利语,Greek#希腊语,German#德语,Italian#意大利语,Czech#捷克语,Slovakia#斯洛伐克语,French#法语,Polish#波兰语,Dutch#荷兰语,Portuguese#葡萄牙语,Spanish#西班牙语,Romanian#罗马尼亚语,Turkish#土耳其语,Japanese#日语,Danish#丹麦语,Swedish#瑞典语,Norwegian#挪威语,Finnish#芬兰语,Korean#韩语,Thai#泰语,Estonia#爱沙尼亚语,Vietnamese#越南语,Hebrew#希伯来语,Latvian#拉脱维亚语,Arabic#阿拉伯语,Slovenian#斯洛文尼亚语,Croatian#克罗地亚语,Lithuanian#立陶宛语,Serbian#塞尔维亚语,BrazilianPortuguese#巴西葡萄牙语,Indonesian#印尼语,Ukrainian#乌克兰语,EURSpanish#西班牙语(欧洲),Sovenian#Sovenian,Uzbek#乌兹别克语,Kazak#哈萨克语,Kirghiz#吉尔吉斯语,Farsi#波斯语,Azerbaidzhan#阿塞拜疆语,Burmes#缅甸语,Mongolian#蒙古语,Anglicism#英式英语]-->SimChinese
  </language>
</DeviceLanguage>
```

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 响应消息, attr:version{ro, req, string, 版本}-->
  <requestURL>
    <!--ro, req, string, 请求的URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, 状态码, subType:int, [0#OK,1#OK,2#Device Busy,3#Device Error,4#Invalid Operation,5#Invalid XML Format,6#Invalid XML Content,7#Reboot Required]-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, 状态信息, subType:string, [OK#成功,Device Busy#设备忙,Device Error#设备异常,Invalid Operation#无效的操作,Invalid XML Format#无效的XML格式,Invalid XML Content#无效的XML内容,Reboot#设备重启]-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, 详细错误码英文描述, desc: 详细错误码的英文描述-->OK
  </subStatusCode>
</ResponseStatus>
```

69.6.7 获取设备语言

Request URL

GET /ISAPI/System/DeviceLanguage

查询参数

无

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<DeviceLanguage xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 设备语言, attr:version{req, string, 协议版本}-->
  <language>
    <!--ro, req, enum, 语言, subType:string, [SimChinese#简体中文,TraChinese#繁体中文,English#英文,Russian#俄语,Bulgarian#保加利亚语,Hungarian#匈牙利语,Greek#希腊语,German#德语,Italian#意大利语,Czech#捷克语,Slovakia#斯洛伐克语,French#法语,Polish#波兰语,Dutch#荷兰语,Portuguese#葡萄牙语,Spanish#西班牙语,Romanian#罗马尼亚语,Turkish#土耳其语,Japanese#日语,Danish#丹麦语,Swedish#瑞典语,Norwegian#挪威语,Finnish#芬兰语,Korean#韩语,Estonia#爱沙尼亚语,Vietnamese#越南语,Hebrew#希伯来语,Latvian#拉脱维亚语,Arabic#阿拉伯语,Slovenian#斯洛文尼亚语,Croatian#克罗地亚语,Lithuanian#立陶宛语,Serbian#塞尔维亚语,BrazilianPortuguese#巴西葡萄牙语,Indonesian#印尼语,Ukrainian#乌克兰语,EURSpanish#西班牙语（欧洲）,Sovenian#Sovenian,Uzbek#乌兹别克语,Kazak#哈萨克语,Kirghiz#吉尔吉斯语,Farsi#波斯语,Azerbaidzhan#阿塞拜疆语,Burmese#缅甸语,Mongolian#蒙古语,Anglicism#英式英语]-->SimChinese
  </language>
</DeviceLanguage>
```

69.6.8 获取设备语言配置能力

Request URL

GET /ISAPI/System/DeviceLanguage/capabilities

查询参数

无

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<DeviceLanguage xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, 设备语言配置, attr:version{req, string, 协议版本}-->
  <language
    opt="SimChinese,TraChinese,English,Russian,Bulgarian,Hungarian,Greek,German,Italian,Czech,Slovakia,French,Polish,Dutch,Portuguese,Spanish,Romanian,Turkish,Japanese,Danish,Swedish,Norwegian,Finnish,Korean,Thai,Estonia,Vietnamese,Hebrew,Latvian,Arabic,Sovenian,Croatian,Lithuanian,Serbian,BrazilianPortuguese,Indonesian,Ukrainian,EURSpanish,Uzbek,Kazak,Kirghiz,Farsi,Azerbaidzhan,Burmese,Mongolian,Anglicism">
    <!--ro, req, enum, 语言, subType:string, [SimChinese#简体中文,TraChinese#繁体中文,English#英文,Russian#俄语,Bulgarian#保加利亚语,Hungarian#匈牙利语,Greek#希腊语,German#德语,Italian#意大利语,Czech#捷克语,Slovakia#斯洛伐克语,French#法语,Polish#波兰语,Dutch#荷兰语,Portuguese#葡萄牙语,Spanish#西班牙语,Romanian#罗马尼亚语,Turkish#土耳其语,Japanese#日语,Danish#丹麦语,Swedish#瑞典语,Norwegian#挪威语,Finnish#芬兰语,Korean#韩语,Thai#泰语,Estonia#爱沙尼亚语,Vietnamese#越南语,Hebrew#希伯来语,Latvian#拉脱维亚语,Arabic#阿拉伯语,Slovenian#斯洛文尼亚语,Croatian#克罗地亚语,Lithuanian#立陶宛语,Serbian#塞尔维亚语,BrazilianPortuguese#巴西葡萄牙语,Indonesian#印尼语,Ukrainian#乌克兰语,EURSpanish#西班牙语（欧洲）,Sovenian#Sovenian,Uzbek#乌兹别克语,Kazak#哈萨克语,Kirghiz#吉尔吉斯语,Farsi#波斯语,Azerbaidzhan#阿塞拜疆语,Burmese#缅甸语,Mongolian#蒙古语,Anglicism#英式英语], attr:opt{req, string, 取值范围}-->SimChinese
  </language>
  <upgradeFirmwareEnabled>
    <!--ro, opt, bool, 升级固件启用-->true
  </upgradeFirmwareEnabled>
</DeviceLanguage>
```

69.6.9 获取SSH参数

Request URL

GET /ISAPI/System/Network/ssh

查询参数

无

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<SSH xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, SSH参数, attr:version{opt, string, 协议版本}-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <port>
    <!--ro, opt, int, SSH端口号-->22
  </port>
  <serviceClosurePeriod>
    <!--ro, opt, int, SSH服务开启期限, range:[0,30], unit:d, unitType:时间-->30
  </serviceClosurePeriod>
</SSH>
```

69.6.10 配置SSH参数

Request URL

PUT /ISAPI/System/Network/ssh

查询参数

无

请求报文

```
<?xml version="1.0" encoding="UTF-8"?>

<SSH xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--wo, opt, object, SSH参数, attr:version{opt, string, 协议版本}-->
  <enabled>
    <!--wo, req, bool, 使能-->true
  </enabled>
  <port>
    <!--wo, opt, int, SSH端口号-->22
  </port>
  <serviceClosurePeriod>
    <!--wo, opt, int, SSH服务开启期限, range:[0,30], unit:d, unitType:时间-->30
  </serviceClosurePeriod>
</SSH>
```

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 响应消息, attr:version{ro, req, string, 版本}-->
  <requestURL>
    <!--ro, req, string, 请求的URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, 状态码, subType:int, [0#OK,1#OK,2#Device Busy,3#Device Error,4#Invalid Operation,5#Invalid XML Format,6#Invalid XML Content,7#Reboot Required]-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, 状态信息, subType:string, [OK#成功,Device Busy#设备忙,Device Error#设备异常,Invalid Operation#无效的操作,Invalid XML Format#无效的XML格式,Invalid XML Content#无效的XML内容,Reboot#设备重启]-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, 详细错误码英文描述, desc: 详细错误码的英文描述-->OK
  </subStatusCode>
</ResponseStatus>
```

69.6.11 获取SSH配置能力

Request URL

GET /ISAPI/System/Network/ssh/capabilities

查询参数

无

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<SSH xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, SSH参数配置能力, attr:version{opt, string, 协议版本}-->
  <enabled opt="true,false">
    <!--ro, req, bool, 使能, attr:opt{opt, string, 取值范围}-->true
  </enabled>
  <port min="1" max="65535">
    <!--ro, opt, int, SSH端口号, range:[1,65535], attr:min{req, int, 最小值},max{req, int, 最大值}-->22
  </port>
  <serviceClosurePeriod min="0" max="30">
    <!--ro, opt, int, SSH服务开启期限, range:[0,30], unit:d, unitType:时间, attr:min{req, int, 最小值},max{req, int, 最大值}-->30
  </serviceClosurePeriod>
</SSH>
```

69.6.12 导出设备配置文件

Request URL

GET /ISAPI/System/configurationData?security=<security>&iv=<iv>&secretkey=<secretkey>

查询参数

参数名称	参数类型	描述
security	string	没有security表示数据不加密；security=1,表示secretkey字段AES128加密；security=2,表示secretkey字段AES256加密；
iv	string	AES加密时的初始化向量参数，在security=1或者2时为必须参数;
secretkey	string	文件密钥，security对其生效。用户在导出文件时需要设置一个文件密钥，在用户导入文件时，需输入所设置的密钥才可导入成功。

请求报文

无

响应报文

无

69.6.13 获取网络服务能力

Request URL

GET /ISAPI/System/Network/capabilities

查询参数

无

请求报文

无

响应报文

湖南云舟电力科技有限公司
有效期：2024-08-15

```

<?xml version="1.0" encoding="UTF-8"?>
<NetworkCap xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 获取网络服务能力, attr:version{opt, string, 协议版本}-->
  <isSupportWireless>
    <!--ro, req, bool, 是否支持无线网-->true
  </isSupportWireless>
  <isSupportPPPoE>
    <!--ro, req, bool, 是否支持以太网点对点协议-->true
  </isSupportPPPoE>
  <isSupportBond>
    <!--ro, req, bool, 是否支持网卡绑定-->true
  </isSupportBond>
  <isSupport802_1x>
    <!--ro, req, bool, 是否支持802.1x协议-->true
  </isSupport802_1x>
  <isSupportNtp>
    <!--ro, opt, bool, 是否支持网络时间协议-->true
  </isSupportNtp>
  <isSupportFtp>
    <!--ro, opt, bool, 是否支持文件传输协议-->true
  </isSupportFtp>
  <isSupportUpnp>
    <!--ro, opt, bool, 是否支持通用即用即插协议-->true
  </isSupportUpnp>
  <isSupportDdns>
    <!--ro, opt, bool, 是否支持动态域名服务-->true
  </isSupportDdns>
  <isSupportHttps>
    <!--ro, opt, bool, 是否支持超文本传输安全协议-->true
  </isSupportHttps>
  <SnmpCap>
    <!--ro, opt, object, 简单网络管理协议能力-->
    <isSupport>
      <!--ro, req, bool, 是否支持简单网络管理协议-->true
    </isSupport>
  </SnmpCap>
  <isSupportExtNetCfg>
    <!--ro, opt, bool, 是否支持配置扩展的网络参数-->true
  </isSupportExtNetCfg>
  <isSupportSSH>
    <!--ro, opt, bool, 是否支持SSH-->true
  </isSupportSSH>
  <isSupportEZVIZ>
    <!--ro, opt, bool, 是否支持萤石协议-->true
  </isSupportEZVIZ>
  <isSupportEhome>
    <!--ro, opt, bool, 是否支持EHome (ISUP) 协议-->true
  </isSupportEhome>
  <isSupportWirelessDial>
    <!--ro, opt, bool, 是否支持无线拨号协议-->true
  </isSupportWirelessDial>
  <isSupportWirelessServer>
    <!--ro, opt, bool, 是否这次会无线服务器-->true
  </isSupportWirelessServer>
  <isSupportWebSocket>
    <!--ro, opt, bool, 是否支持WebSocket-->true
  </isSupportWebSocket>
  <isSupportEZVIZQRcode>
    <!--ro, opt, bool, 是否支持萤石二维码获取, desc:/ISAPI/System/Network/EZVIZ/QRCode?format=json-->true
  </isSupportEZVIZQRcode>
  <isSupportEZVIZUnbind>
    <!--ro, opt, bool, 是否支持萤石解绑功能, desc:/ISAPI/System/Network/EZVIZ/unbind?format=json-->true
  </isSupportEZVIZUnbind>
</NetworkCap>

```

69.6.14 获取设备存储能力

Request URL

GET /ISAPI/ContentMgmt/capabilities

查询参数

无

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>
<RacmCap xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 设备存储能力, attr:version{req, string, 协议版本}-->
  <isSupportZeroChan>
    <!--ro, opt, bool, 是否支持零通道, desc: 对应/ISAPI/ContentMgmt/ZeroVideo/channels-->true
  </isSupportZeroChan>
  <isSupportLogConfig>
    <!--ro, opt, bool, 是否支持日志配置, desc: 对应/ISAPI/ContentMgmt/LogConfig/capabilities-->true
  </isSupportLogConfig>
  <isSupportGetBackupDevice>
    <!--ro, opt, bool, 是否支持获取备份设备, desc: 对应/ISAPI/ContentMgmt/Storage/backup/device?format=json-->true
  </isSupportGetBackupDevice>
  <isSupportAutoBackup>
    <!--ro, opt, bool, 是否支持自动备份, desc: 对应/ISAPI/ContentMgmt/Storage/backup/auto/capabilities?format=json-->true
  </isSupportAutoBackup>
  <defaultAdminPortEnabled>
    <!--ro, opt, bool, 默认通信端口使能, desc: 对应/ISAPI/ContentMgmt/InputProxy/channels/<ID> 本协议中<defaultAdminPortEnabled>字段是否支持-->true
  </defaultAdminPortEnabled>
  <isSupportLogDataPackage>
    <!--ro, opt, bool, 是否支持日志文件导出, desc: 对应url:/ISAPI/ContentMgmt/LogSearch/dataPackage-->true
  </isSupportLogDataPackage>
  <isSupportLogSearch>
    <!--ro, opt, bool, 设备是否支持日志查询-->true
  </isSupportLogSearch>
</RacmCap>
```

69.6.15 获取设备系统能力

Request URL

GET /ISAPI/System/capabilities

查询参数

无

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>
<DeviceCap xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 设备系统能力, attr:version{opt, string, 版本信息}-->
  <isSupportPreview>
    <!--ro, opt, bool, 是否支持预览-->true
  </isSupportPreview>
  <isSupportReboot>
    <!--ro, opt, bool, 是否支持重启, desc: 交通产线必填且为true-->true
  </isSupportReboot>
  <isSupportFactoryReset>
    <!--ro, opt, bool, 是否支持恢复出厂设置, desc: 交通产线必填且为true-->true
  </isSupportFactoryReset>
  <isSupportUpdatefirmware>
    <!--ro, opt, bool, 是否支持设备升级, desc: 交通产线必填且为true-->true
  </isSupportUpdatefirmware>
  <isSupportDeviceInfo>
    <!--ro, opt, bool, 是否支持获取设备信息, desc: 交通产线必填且为true-->true
  </isSupportDeviceInfo>
  <isSupportTime>
    <!--ro, opt, bool, 表示是否支持时间配置, desc: 交通产线必填且为true, 其他产线使用的是isSupportTimeCap-->true
  </isSupportTime>
  <SysCap>
    <!--ro, opt, object, 系统能力-->
    <isSupportDst>
      <!--ro, opt, bool, 是否支持夏令时-->true
    </isSupportDst>
  </SysCap>
  <NetworkCap>
    <!--ro, opt, object, 网络能力, desc: 接口:/ISAPI/System/Network/capabilities-->
    <isSupportWireless>
      <!--ro, req, bool, 是否支持无线网-->true
    </isSupportWireless>
    <isSupportPPPoE>
      <!--ro, req, bool, 是否支持以太网点对点协议-->true
    </isSupportPPPoE>
    <isSupportBond>
      <!--ro, req, bool, 是否支持网卡绑定-->true
    </isSupportBond>
    <isSupport802_1x>
      <!--ro, req, bool, 是否支持802.1x协议-->true
    </isSupport802_1x>
    <isSupportNtp>
      <!--ro, opt, bool, 是否支持网络时间协议-->true
    </isSupportNtp>
    <isSupportFtp>
      <!--ro, opt, bool, 是否支持文件传输协议-->true
    </isSupportFtp>
  </NetworkCap>
</DeviceCap>
```

```
</isSupportFtp>
<isSupportUpnp>
  <!--ro, opt, bool, 是否支持通用即用即插协议-->true
</isSupportUpnp>
<isSupportDdns>
  <!--ro, opt, bool, 是否支持动态域名服务-->true
</isSupportDdns>
<isSupportHttps>
  <!--ro, opt, bool, 是否支持超文本传输安全协议-->true
</isSupportHttps>
<SnmpCap>
  <!--ro, opt, object, 简单网络管理协议能力-->
  <isSupport>
    <!--ro, req, bool, 是否支持简单网络管理协议-->true
  </isSupport>
</SnmpCap>
<isSupportExtNetCfg>
  <!--ro, opt, bool, whether supports configuring extended network parameter-->true
</isSupportExtNetCfg>
<isSupportSSH opt="true,false">
  <!--ro, opt, bool, 是否支持SSH, attr:opt{opt, string, 取值范围}-->true
</isSupportSSH>
<isSupportEZVIZ>
  <!--ro, opt, bool, 是否支持萤石协议-->true
</isSupportEZVIZ>
<isSupportEhome>
  <!--ro, opt, bool, ehome (ISUP) 服务器配置-->true
</isSupportEhome>
<isSupportWirelessDial>
  <!--ro, opt, bool, 是否支持无线拨号协议-->true
</isSupportWirelessDial>
<isSupportWirelessServer>
  <!--ro, opt, bool, 是否支持无线服务器-->true
</isSupportWirelessServer>
<isSupportWebSocket>
  <!--ro, opt, bool, 是否支持WebSocket-->true
</isSupportWebSocket>
<isSupportEZVIZQRcode>
  <!--ro, opt, bool, 是否支持萤石二维码获取, desc:对应控制接口: /ISAPI/System/Network/EZVIZ/QRCode?format=json-->true
</isSupportEZVIZQRcode>
<isSupportEZVIZUnbind>
  <!--ro, opt, bool, 是否支持萤石解绑功能, desc:/ISAPI/System/Network/EZVIZ/unbind?format=json-->true
</isSupportEZVIZUnbind>
</NetworkCap>
<SerialCap>
  <!--ro, opt, object, 设备支持的485串口号范围-->
  <rs485PortNums>
    <!--ro, opt, int, 设备支持的rs485的串口最大数量-->true
  </rs485PortNums>
</SerialCap>
<VideoCap>
  <!--ro, opt, object, 视频编码能力, desc:接口:/ISAPI/System/Video/capabilities-->
  <videoInputPortNums>
    <!--ro, opt, int, 视频输入端口个数-->true
  </videoInputPortNums>
  <videoOutputPortNums>
    <!--ro, opt, int, 视频输出端口个数-->true
  </videoOutputPortNums>
  <isSupportHeatmap>
    <!--ro, opt, bool, 是否支持热度图-->true
  </isSupportHeatmap>
  <isSupportCounting>
    <!--ro, opt, bool, 是否支持人数统计-->true
  </isSupportCounting>
  <isSupportPreviewSwitch>
    <!--ro, opt, bool, 是否支持预览切换-->true
  </isSupportPreviewSwitch>
  <isSupportRecodStatus>
    <!--ro, opt, bool, 是否支持录像状态查询-->true
  </isSupportRecodStatus>
  <isSupportPrivacyMask>
    <!--ro, opt, bool, 是否支持隐私遮蔽功能-->true
  </isSupportPrivacyMask>
  <isSupportBinocularPreviewSwitch>
    <!--ro, opt, bool, 是否支持双目预览切换-->true
  </isSupportBinocularPreviewSwitch>
  <isSupportCalibCheck>
    <!--ro, opt, bool, 是否支持标定检验-->true
  </isSupportCalibCheck>
  <isSupportPIP>
    <!--ro, opt, bool, 是否支持PIP-->true
  </isSupportPIP>
</VideoCap>
<AudioCap>
  <!--ro, opt, object, 音频编码能力, desc:接口:/ISAPI/System/Audio/capabilities-->
  <audioInputNums>
    <!--ro, req, int, 音频输入个数-->1
  </audioInputNums>
  <audioOutputNums>
    <!--ro, req, int, 音频输出个数-->1
  </audioOutputNums>
  <mixAudioInSet>
    <!--ro, opt, bool, 待确认-->true
  </mixAudioInSet>
```

```

</mixAudioOutSet>
<mixAudioOutSet>
  <!--ro, opt, bool, 待确认-->true
</mixAudioOutSet>
<isSupportAudioMixing>
  <!--ro, opt, bool, 是否支持音频混音-->true
</isSupportAudioMixing>
</AudioCap>
<isSupportExternalDevice>
  <!--ro, opt, bool, 是否支持设备接入外设功能(true-支持, desc:接口:/ISAPI/System/externalDevice/capabilities)-->true
</isSupportExternalDevice>
<isSupportSubscribeEvent>
  <!--ro, opt, bool, 是否支持事件订阅, desc:接口:/ISAPI/Event/notification/subscribeEventCap-->true
</isSupportSubscribeEvent>
<isSupportTimeCap>
  <!--ro, opt, bool, 是否支持时间配置, desc:接口:/ISAPI/System/time/capabilities-->true
</isSupportTimeCap>
<isSupportPostUpdateFirmware>
  <!--ro, opt, bool, 是否支持POST方法升级固件-->true
</isSupportPostUpdateFirmware>
</SysCap>
<voicetalkNums>
  <!--ro, opt, int, 语音对讲通道数目-->2
</voicetalkNums>
<isSupportSnapshot>
  <!--ro, opt, bool, 是否支持抓图-->true
</isSupportSnapshot>
<SecurityCap>
  <!--ro, opt, object, 表示加密能力集-->
  <supportUserNums>
    <!--ro, opt, int, 支持用户最大数-->1
  </supportUserNums>
  <userBondIpNums>
    <!--ro, opt, int, 支持绑定IP最大数-->1
  </userBondIpNums>
  <userBondMacNums>
    <!--ro, opt, int, 支持绑定MAC地址最大数-->1
  </userBondMacNums>
  <isSupCertificate>
    <!--ro, opt, bool, 是否支持认证-->true
  </isSupCertificate>
  <issupIllegalLoginLock>
    <!--ro, opt, bool, 是否支持登陆锁定-->true
  </issupIllegalLoginLock>
  <isSupportOnlineUser>
    <!--ro, opt, bool, 是否支持在线用户配置-->true
  </isSupportOnlineUser>
  <isSupportAnonymous>
    <!--ro, opt, bool, 是否支持匿名登陆-->true
  </isSupportAnonymous>
  <isSupportStreamEncryption>
    <!--ro, opt, bool, 是否支持码流加密-->true
  </isSupportStreamEncryption>
  <securityVersion opt="1,2">
    <!--ro, opt, string, 加密能力, attr:opt{opt, string, 取值范围}, desc:每一个版本包括两个方面: 加密算法及加密的节点范围。目前1表示AES128加密,2表示AES256
    加密,加密范围在各个协议中描述-->test
  </securityVersion>
  <keyIterateNum>
    <!--ro, opt, int, 迭代次数, desc:取值一般是100-1000-->100
  </keyIterateNum>
  <isSupportUserCheck>
    <!--ro, opt, bool, 是否支持用户参数修改(修改/增加/删除)时进行登录密码的校验, desc:表示是否支持用户参数修改(修改/增加/删除)时进行登录密码的校验,此功能
    只在支持敏感信息加密的版本上有效,即depend on securityVersion-->true
  </isSupportUserCheck>
  <isSupportGUIDFileDataExport>
    <!--ro, opt, bool, 表示是否支持设备GUID文件导出-->true
  </isSupportGUIDFileDataExport>
  <isSupportSecurityQuestionConfig>
    <!--ro, opt, bool, 表示是否支持回答安全问题-->true
  </isSupportSecurityQuestionConfig>
  <isSupportGetOnlineUserListSC>
    <!--ro, opt, bool, 是否支持在线用户列表查询-->true
  </isSupportGetOnlineUserListSC>
  <SecurityLimits>
    <!--ro, opt, object, 参数安全限制能力-->
    <LoginPasswordLenLimit min="1" max="16">
      <!--ro, opt, string, 用户登陆密码长度限制, attr:min{opt, string, 最小值},max{opt, string, 最大值}-->test
    </LoginPasswordLenLimit>
    <SecurityAnswerLenLimit min="1" max="128">
      <!--ro, opt, string, 安全问题答案长度限制, attr:min{opt, string, 最小值},max{opt, string, 最大值}-->test
    </SecurityAnswerLenLimit>
  </SecurityLimits>
  <RSAKeyLength opt="512,1024,2048" def="2048">
    <!--ro, opt, enum, HTTPS证书长度, subType:string, [512#512,1024#1024,2048#2048], attr:opt{opt, string, 取值范围},def{opt, string, 默认值}-->2048
  </RSAKeyLength>
  <isSupportONVIFUserManagement>
    <!--ro, opt, bool, ONVIF协议用户管理-->true
  </isSupportONVIFUserManagement>
  <WebCertificateCap>
    <!--ro, opt, object, HTTP认证能力-->
    <CertificateType opt="basic,digest,digest/basic">
      <!--ro, req, string, 证书类型: basic认证, attr:opt{opt, string, 取值范围}-->test
    </CertificateType>
  </WebCertificateCap>

```

```

<isSupportConfigFileImport>
  <!--ro, opt, bool, 是否支持配置文件安全导入-->true
</isSupportConfigFileImport>
<isSupportConfigFileExport>
  <!--ro, opt, bool, 是否支持配置文件安全导出-->true
</isSupportConfigFileExport>
<cfgFileSecretKeyLenLimit min="0" max="16">
  <!--ro, opt, string, 配置文件校验密钥长度限制, attr:min{opt, string, 最小值},max{opt, string, 最大值}-->0
</cfgFileSecretKeyLenLimit>
<salt>
  <!--ro, opt, string, 用于用户登录时所使用的具体盐值-->test
</salt>
<isSupportDeviceCertificatesManagement>
  <!--ro, opt, bool, 是否支持设备证书管理-->true
</isSupportDeviceCertificatesManagement>
<isSupportSecurityEmail>
  <!--ro, opt, bool, 是否支持安全邮箱配置-->true
</isSupportSecurityEmail>
<isSupportEncryptCertificate>
  <!--ro, opt, bool, 是否支持证书加密, desc:/ISAPI/Security/deviceCertificate-->true
</isSupportEncryptCertificate>
<isSupportCertificateCustomID>
  <!--ro, opt, bool, 是否支持证书配置使用用户自定义ID-->true
</isSupportCertificateCustomID>
</SecurityCap>
<EventCap>
  <!--ro, opt, object, 事件能力-->
</EventCap>
<RacmCap>
  <!--ro, opt, object, 图片搜索之前是一个界面-->
<SecurityLog>
  <!--ro, opt, object, 设备安全日志能力-->
  <isSupportLogServer>
    <!--ro, opt, bool, 是否支持安全审计日志服务器配置-->true
  </isSupportLogServer>
</SecurityLog>
</RacmCap>
<isSupportEhome>
  <!--ro, opt, bool, 是否支持ISUP功能-->true
</isSupportEhome>
<isSupportFirmwareVersionInfo>
  <!--ro, opt, bool, 是否支持显示固件版本信息-->true
</isSupportFirmwareVersionInfo>
<VideoIntercomCap>
  <!--ro, opt, object, 可视对讲能力, desc:接口:/ISAPI/VideoIntercom/capabilities-->
</VideoIntercomCap>
<isSupportAcsUpdate>
  <!--ro, opt, bool, 是否支持外设模块升级（支持时返回true, desc:接口:/ISAPI/System/AcsUpdate/capabilities-->true
</isSupportAcsUpdate>
<isSupportEncryption>
  <!--ro, opt, bool, 码流加密能力-->true
</isSupportEncryption>
<supportImageChannel opt="1,2,3,4">
  <!--ro, opt, string, 是否支持图像通道配置, attr:opt{req, string, 取值范围},
desc:当前设备所支持的图像通道配置(图像参数接口: /ISAPI/Image/channels/<channelID>),若不返回该节点图像通道默认与编码通道(/ISAPI/Streaming/channels)保持一致。
  该节点主要用于图像通道与实际码流通道不一致的情况:如实验考图像拼接设备,该设备存在四个sensor却拼接为一组编码通道进行输出,但是每个sensor都支持图像参数配置,图像通道1默认仅对sensor1的图像生效-->test
</supportImageChannel>
<isSupportBluetooth>
  <!--ro, opt, bool, 是否支持蓝牙, desc:接口:/ISAPI/System/Bluetooth/capabilities-->true
</isSupportBluetooth>
<isSupportPictureServer>
  <!--ro, opt, bool, 是否支持图片存储服务器配置, desc:接口:/ISAPI/System/PictureServer/capabilities?format=json-->true
</isSupportPictureServer>
<isSupportAlgorithmsInfo>
  <!--ro, opt, bool, 是否支持设备算法版本获取-->true
</isSupportAlgorithmsInfo>
<isSupportVisitorEvent>
  <!--ro, opt, bool, 是否支持访客事件, desc:对应事件上报报文中eventType:VisitorEvent-->true
</isSupportVisitorEvent>
<isSupportVisitorInfoSearchEvent>
  <!--ro, opt, bool, 是否支持访客信息查询事件, desc:对应事件上报报文中eventType:VisitorInfoSearchEvent-->true
</isSupportVisitorInfoSearchEvent>
<isSupportAccessControlCap>
  <!--ro, opt, bool, 门禁总能力, desc:接口: /ISAPI/AccessControl/capabilities-->true
</isSupportAccessControlCap>
<isSupportClientProxyWEB>
  <!--ro, opt, bool, 是否支持WEB页面跳转-->true
</isSupportClientProxyWEB>
<WEBLocation>
  <!--ro, opt, string, WEB页面放置位置, range:[1,32], desc:Local-设备本地, remote-远程（不返回该节点时,默认WEB页面放置设备本地）-->local
</WEBLocation>
<deviceId>
  <!--ro, opt, string, 设备ID, range:[1,32], desc:如果WEBLocation为remote,则需要返回设备唯一的设备ID,用来标示设备唯一对应的web包,这个deviceId有Web配置包统一管理,一款设备多型号返回相同-->test
</deviceId>
<isSupportUncertificateCompareEvent>
  <!--ro, opt, bool, 是否支持无证比对事件, desc:接口: eventType:UncertificateCompareEvent,不支持不返回-->true
</isSupportUncertificateCompareEvent>
<isSupportSetupStructure>
  <!--ro, opt, bool, 是否支持架设结构参数, desc:/ISAPI/System/setupStructure?format=json-->true
</isSupportSetupStructure>

```

```
<isSupportEZVIZParameterTest>
  <!--ro, opt, bool, 是否支持萤石参数测试, desc: 对应URL:/ISAPI/System/Network/EZVIZ/parameterTest?format=json-->true
</isSupportEZVIZParameterTest>
</DeviceCap>
```

69.6.16 设备重启

Request URL

PUT /ISAPI/System/reboot

查询参数

无

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 响应消息, attr:version{ro, req, string, 版本}-->
  <requestURL>
    <!--ro, req, string, 请求的URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, 状态码, subType:int, [0#OK,1#OK,2#Device Busy,3#Device Error,4#Invalid Operation,5#Invalid XML Format,6#Invalid XML Content,7#Reboot Required]-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, 状态信息, subType:string, [OK#成功,Device Busy#设备忙,Device Error#设备异常,Invalid Operation#无效的操作,Invalid XML Format#无效的XML格式,Invalid XML Content#无效的XML内容,Reboot#设备重启]-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, 详细错误码英文描述, desc:详细错误码的英文描述-->OK
  </subStatusCode>
</ResponseStatus>
```

69.6.17 配置设备恢复默认参数

Request URL

PUT /ISAPI/System/factoryReset?mode=<mode>

查询参数

参数名称	参数类型	描述
mode	enum	

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 响应消息, attr:version{ro, req, string, 版本}-->
  <requestURL>
    <!--ro, req, string, 请求的URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, 状态码, subType:int, [0#OK,1#OK,2#Device Busy,3#Device Error,4#Invalid Operation,5#Invalid XML Format,6#Invalid XML Content,7#Reboot Required]-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, 状态信息, subType:string, [OK#成功,Device Busy#设备忙,Device Error#设备异常,Invalid Operation#无效的操作,Invalid XML Format#无效的XML格式,Invalid XML Content#无效的XML内容,Reboot#设备重启]-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, 详细错误码英文描述, desc:详细错误码的英文描述-->OK
  </subStatusCode>
</ResponseStatus>
```

69.6.18 导入设备配置文件（不推荐）

Request URL

PUT /ISAPI/System/configurationData?security=<security>&iv=<iv>&secretkey=<secretkey>

查询参数

参数名称	参数类型	描述
security	string	没有security表示数据不加密； security=1,表示secretkey字段AES128加密； security=2,表示secretkey字段AES256加密；
iv	string	AES加密时的初始化向量参数，在security=1或者2时为必须参数;
secretkey	string	密钥，security对其生效。

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 响应消息, attr:version{ro, req, string, 版本}-->
  <requestURL>
    <!--ro, req, string, 请求的URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, 状态码, subType:int, [0#OK,1#OK,2#Device Busy,3#Device Error,4#Invalid Operation,5#Invalid XML Format,6#Invalid XML Content,7#Reboot Required]-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, 状态信息, subType:string, [OK#成功,Device Busy#设备忙,Device Error#设备异常,Invalid Operation#无效的操作,Invalid XML Format#无效的XML格式,Invalid XML Content#无效的XML内容,Reboot#设备重启]-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, 详细错误码英文描述, desc:详细错误码的英文描述-->OK
  </subStatusCode>
</ResponseStatus>
```

69.7 串口配置

69.7.1 获取全部串口控制参数

Request URL

GET /ISAPI/System/Serial/ports?permissionController=<indexID>

查询参数

参数名称	参数类型	描述
indexID	string	0-主权限控制器，1-从权限控制器。该参数不存在时，表示获取主权限控制器串口信息（即设备串口信息）

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<SerialPortList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, SerialPortList, attr:version{req, string, 协议版本}-->
  <SerialPort>
    <!--ro, opt, object, 端口-->
    <id>
      <!--ro, req, int, 串口号-->0
    </id>
    <enabled>
```

```
<!--ro, opt, bool, 使能-->true
</enabled>
<encryptKey>
<!--ro, opt, string, 用于串口通信之间加密的密钥-->test
</encryptKey>
<serialNumber>
<!--ro, opt, int, 序列号-->0
</serialNumber>
<serialPortType>
<!--ro, opt, enum, 串口类型, subType:string, [RS485#RS485,RS422#RS422,RS232#RS232]-->RS485
</serialPortType>
<serialAddress>
<!--ro, opt, int, 串口地址, desc: 只有串口类型为RS485时有效-->1
</serialAddress>
<duplexMode>
<!--ro, opt, enum, 双工模式, subType:string, [half#半双工,full#全双工]-->half
</duplexMode>
<direction>
<!--ro, opt, enum, 单向的, subType:string, [monodirectional#单向通讯,bidirectional#双向通讯]-->monodirectional
</direction>
<baudRate>
<!--ro, opt, enum, 波特率, subType:int, [2400#2400,4800#4800,9600#9600,19200#19200,38400#38400,57600#57600,115200#115200]-->2400
</baudRate>
<dataBits>
<!--ro, opt, int, 数据位-->6
</dataBits>
<parityType>
<!--ro, opt, enum, 奇偶校验类型, subType:string, [none#无校验位,even#偶校验,odd#奇校验,mark#校验位始终为1,space#校验位始终为0]-->none
</parityType>
<stopBits>
<!--ro, opt, enum, 停止位, subType:string, [1#1,1.5#1.5,2#2]-->1
</stopBits>
<workMode>
<!--ro, opt, enum, 工作模式, subType:string, [console#控制台,transparent#透传,audiomix#混音,screenCtrl#屏幕控制,ptzCtrl#PTZ控制,keyboard#键盘,matrix#
矩阵]-->console
</workMode>
<flowCtrl>
<!--ro, opt, enum, 串口号流控制, subType:string, [none#禁用串口号流控制,software#软件串口号流控制,hardware#硬件串口号流控制]-->none
</flowCtrl>
<rs485WorkMode>
<!--ro, opt, enum, rs485的工作模式, subType:string, [Led#外接LED屏,sensor#外接传感器,CaptureTrigger#外接触发设备进行触发抓拍数据传输,transparent#透明
传输,GPS#外接GPS模块,ModbusRTU#外接RTU模块,supplementLight#白光补光灯], dep:or,{$.SerialPort.serialPortType,eq,RS485}-->sensor
</rs485WorkMode>
<copyToAll>
<!--ro, opt, bool, 是否复制当前串口配置给所有串口-->true
</copyToAll>
<audiomixerProtocolType>
<!--ro, opt, enum, HIKVISION, subType:string, [HIKVISION#HIKVISION,DS-ZH101S#DS-ZH101S]-->HIKVISION
</audiomixerProtocolType>
<deviceName>
<!--ro, opt, string, 设备名称-->test
</deviceName>
<deviceType>
<!--ro, opt, int, 设备类型-->1
</deviceType>
<deviceProtocol>
<!--ro, opt, int, 设备协议-->1
</deviceProtocol>
<mode>
<!--ro, opt, enum, 工作模式, subType:string, [readerMode#下行接读卡器,clientMode#上行输出至4200客户端,externMode#下行接门禁模块,stairControl#上行输出
至门禁主机,accessControlHost#下行接梯控主机,disabled#禁用,cardReceiver#收卡器,QRCodeReader#二维码读卡器,disabled#关闭,custom#自定义]-->readerMode
</mode>
<outputDataType>
<!--ro, opt, enum, 输出数据类型, subType:string, [cardNo#卡号,employeeNo#工号], dep:and,{$.SerialPort.mode,eq,accessControlHost}-->cardNo
</outputDataType>
<isVariable>
<!--ro, opt, bool, 是否可变的-->true
</isVariable>
<PTZInfo>
<!--ro, opt, object, 云台信息-->
<bindChannelNo>
<!--ro, req, int, 绑定通道号-->0
</bindChannelNo>
<PTZProtocolList>
<!--ro, req, object, PTZ协议列表-->
<PTZProtocol>
<!--ro, opt, object, PTZ协议-->
<id>
<!--ro, req, int, 协议索引-->0
</id>
<protoDesc>
<!--ro, req, string, 协议描述-->test
</protoDesc>
</PTZProtocol>
</PTZProtocolList>
</PTZInfo>
<ScreenCtrlInfo>
<!--ro, opt, object, ScreenCtrlInfo, desc: 当workMode包含screenCtrl时有效-->
<ScreenCtrlProtoList>
<!--ro, req, object, ScreenCtrlProtoList-->
<ScreenCtrlProto>
<!--ro, opt, object, ScreenCtrlProto-->
<id>
```

```
<!--ro, req, int, 协议索引-->0
</id>
<protoDesc>
  <!--ro, req, string, 协议描述-->test
</protoDesc>
</ScreenCtrlProto>
</ScreenCtrlProtoList>
</ScreenCtrlInfo>
<MatrixCtrlInfo>
  <!--ro, opt, object, MatrixCtrlInfo-->
  <MatrixCtrlProtoList>
    <!--ro, req, object, MatrixCtrlProtoList-->
    <MatrixCtrlProto>
      <!--ro, opt, object, MatrixCtrlProto-->
      <id>
        <!--ro, req, int, 协议索引-->0
      </id>
      <protoDesc>
        <!--ro, req, string, 协议描述-->test
      </protoDesc>
    </MatrixCtrlProto>
  </MatrixCtrlProtoList>
</MatrixCtrlInfo>
<KeyboardCtrlInfo>
  <!--ro, opt, object, KeyboardCtrlInfo-->
  <KeyboardCtrlProtoList>
    <!--ro, req, object, KeyboardCtrlProtoList-->
    <KeyboardCtrlProto>
      <!--ro, opt, object, KeyboardCtrlProto-->
      <id>
        <!--ro, req, int, 协议索引-->0
      </id>
      <protoDesc>
        <!--ro, req, string, 协议描述-->test
      </protoDesc>
    </KeyboardCtrlProto>
  </KeyboardCtrlProtoList>
</KeyboardCtrlInfo>
<ControlAddress>
  <!--ro, opt, object, 控制（解码器）地址-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <address>
    <!--ro, req, int, 地址, range:[0,255]-->1
  </address>
</ControlAddress>
<SensorCtrlList>
  <!--ro, opt, object, 外接传感器控制信息, desc:rs485WorkMode 支持sensor时返回-->
  <Sensor>
    <!--ro, opt, object, 传感器-->
    <id>
      <!--ro, req, int, 索引, range:[1,3]-->1
    </id>
    <name>
      <!--ro, req, string, 传感器名称, range:[1,32], desc:visibilitySensor-能见度传感器, fiveElementWeatherSensor-气象五要素传感器, roadConditionSensor-路面状态传感器-->visibilitySensor
    </name>
    <enabled>
      <!--ro, req, bool, 使能, desc:开启后, 传感器定时上传事件.例如: 开启能见度传感器后, 上报事件roadVisibilityAnalysis-->>false
    </enabled>
  </Sensor>
</SensorCtrlList>
<stairsControl>
  <!--ro, opt, object, 梯控参数, dep:and,{$.SerialPort.mode,eq,stairsControl}-->
  <outputDataType>
    <!--ro, opt, enum, 门口机对梯控主机的输出数据类型, subType:string, [fFloorNumber#楼层号（默认）,cardNo#卡号]-->fFloorNumber
  </outputDataType>
</stairsControl>
<comMode>
  <!--ro, opt, enum, 通信模式, subType:string, [active#主站模式,passive#从站模式]-->active
</comMode>
<ModbusRTU>
  <!--ro, opt, object, 外接RTU模块参数, dep:and,{$.SerialPortList.SerialPort.rs485WorkMode,eq,ModbusRTU}-->
  <ModbusRTUBaudRate>
    <!--ro, opt, enum, 波特率, subType:string, [1200#1200,2400#2400,4800#4800,9600#9600,19200#19200]-->9600
  </ModbusRTUBaudRate>
  <ModbusRTUDataBits>
    <!--ro, opt, enum, 数据位, subType:int, [5#5,6#6,7#7,8#8]-->6
  </ModbusRTUDataBits>
  <ModbusRTUParityType>
    <!--ro, opt, enum, 奇偶校验类型, subType:string, [none#无校验位,even#偶校验,odd#奇校验]-->none
  </ModbusRTUParityType>
  <ModbusRTUStopBits>
    <!--ro, opt, enum, 停止位, subType:string, [1#1,2#2]-->1
  </ModbusRTUStopBits>
  <ModbusRTUFlowCtrl>
    <!--ro, opt, enum, 串口流控制, subType:string, [none#禁用串口流控制,software#软件串口流控制,hardware#硬件串口流控制]-->none
  </ModbusRTUFlowCtrl>
  <decoderAddress>
    <!--ro, opt, int, 解码器地址, range:[0,255]-->0
  </decoderAddress>
  <online>
    <!--ro, opt, bool, 在线状态-->true
```

```
</online>
</ModbusRTU>
<controlProtocol>
  <!--ro, opt, enum, 控制（解码器）协议, subType:string, [HIKVISION#HIKVISION, PELCO-D#PELCO-D, PELCO-P#PELCO-P]-->HIKVISION
</controlProtocol>
<accessName>
  <!--ro, opt, string, 串口接口接入名称, range:[1,32]-->test
</accessName>
<serialPortUse>
  <!--ro, opt, enum, 串口用途, subType:string, [acquAndCtrl#采集与控制, swipe#刷卡, strongElectric#强电网]-->acquAndCtrl
</serialPortUse>
<serialProtocol>
  <!--ro, opt, enum, 串口协议, subType:string, [FECbus#FECbus, GB26875#GB26875]-->GB26875
</serialProtocol>
<SupplementLightList>
  <!--ro, opt, array, 外接白光灯控制信息, subType:object, dep:and,{$.SerialPort.rs485WorkMode, eq, supplementLight}-->
  <SupplementLight>
    <!--ro, opt, object, 外接白光灯-->
    <SupplementLightID>
      <!--ro, req, int, 补光灯ID, range:[1,5], desc:通过/ISAPI/System/externalDevice/supplementLightStrategy?format=json配置-->1
    </SupplementLightID>
  </SupplementLight>
</SupplementLightList>
<cardReaderType>
  <!--ro, opt, enum, 刷卡器类型, subType:string, [K1108AM#K1108AM型号], desc:只读-->K1108AM
</cardReaderType>
<strongElectricType>
  <!--ro, opt, enum, 强电类型, subType:string, [19M04-RO/T#19M04-RO/T型号, 19M08-RO/T#19M08-RO/T型号], desc:只读-->19M04-RO/T
</strongElectricType>
</SerialPort>
</SerialPortList>
```

69.7.2 配置单个串口控制参数

Request URL

PUT /ISAPI/System/Serial/ports/<portID>?permissionController=<indexID>

查询参数

参数名称	参数类型	描述
portID	string	
indexID	string	0-主权限控制器，1-从权限控制器。该参数不存在时，表示配置主权限控制器串口信息（即设备串口信息）

请求报文

```
<?xml version="1.0" encoding="UTF-8"?>
<SerialPort xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--opt, object, 端口, attr:version{req, string, 协议版本}-->
  <id>
    <!--req, int, 端口号-->0
  </id>
  <enabled>
    <!--opt, bool, 使能-->>true
  </enabled>
  <encryptKey>
    <!--opt, string, 用于串口通信之间加密的密钥-->test
  </encryptKey>
  <serialNumber>
    <!--opt, int, 序列号-->0
  </serialNumber>
  <serialPortType>
    <!--opt, enum, 串口类型, subType:string, [RS485#RS485, RS422#RS422, RS232#RS232]-->RS485
  </serialPortType>
  <serialAddress>
    <!--opt, int, 串口地址, desc:只有串口类型为RS485时有效-->1
  </serialAddress>
  <duplexMode>
    <!--opt, enum, 双工模式, subType:string, [half#半双工, full#全双工]-->half
  </duplexMode>
  <direction>
    <!--opt, enum, 单向的, subType:string, [monodirectional#单向通讯, bidirectional#双向通讯]-->monodirectional
  </direction>
  <baudRate>
    <!--opt, enum, 波特率, subType:int, [2400#2400, 4800#4800, 9600#9600, 19200#19200, 38400#38400, 57600#57600, 115200#115200]-->2400
  </baudRate>
  <dataBits>
    <!--opt, int, 数据位-->6
  </dataBits>
  <parityType>
```

```
<!--opt, enum, 奇偶校验类型, subType:string, [none#无校验位,even#偶校验,odd#奇校验,mark#校验位始终为1,space#校验位始终为0]-->none
</parityType>
<stopBits>
  <!--opt, string, 停止位-->1
</stopBits>
<workMode>
  <!--opt, enum, 工作模式, subType:string, [console#控制台,transparent#透传,audiomixe#混音,screenCtrl#屏幕控制,ptzCtrl#PTZ控制,keyboard#键盘,matrix#矩阵]-
->console
  </workMode>
  <flowCtrl>
    <!--opt, enum, 串口流控制, subType:string, [none#禁用串口流控制,software#软件串口流控制,hardware#硬件串口流控制]-->none
  </flowCtrl>
  <rs485WorkMode>
    <!--opt, enum, rs485的工作模式, subType:string, [Led#外接LED屏,sensor#外接传感器,CaptureTrigger#外接触发设备进行触发抓拍数据传输,transparent#透明传
输,GPS#外接GPS模块,ModbusRTU#外接RTU模块,supplementLight#白光补光灯,columnSpeaker#音柱模式,commModeUsing433MHz#433模块], dep:or,
{$.SerialPort.serialPortType,eq,RS485}-->sensor
  </rs485WorkMode>
  <copyToAll>
    <!--opt, bool, 是否复制当前串口配置给所有串口-->true
  </copyToAll>
  <audiomixerProtocolType>
    <!--opt, enum, HIKVISION, subType:string, [HIKVISION#HIKVISION,DS-ZH101S#DS-ZH101S]-->HIKVISION
  </audiomixerProtocolType>
  <deviceName>
    <!--ro, opt, string, 设备名称-->test
  </deviceName>
  <deviceType>
    <!--ro, opt, int, 设备类型-->1
  </deviceType>
  <deviceProtocol>
    <!--ro, opt, int, 设备协议-->1
  </deviceProtocol>
  <mode>
    <!--opt, enum, 工作模式, subType:string, [readerMode#下行接读卡器,clientMode#上行输出至4200客户端,externMode#下行接门禁模块,stairsControl#上行输出至门禁
主机,accessControlHost#下行接梯控主机,cardReceiver#收卡器,QRCoderReader#二维码读卡器,disabled#禁用,custom#自定义]-->readerMode
  </mode>
  <outputDataType>
    <!--opt, enum, 输出数据类型, subType:string, [cardNo#卡号,employeeNo#工号], dep:and,{$.SerialPort.mode,eq,accessControlHost}-->cardNo
  </outputDataType>
  <isVariable>
    <!--opt, bool, 是可变的-->true
  </isVariable>
  <PTZInfo>
    <!--opt, object, 云台信息-->
    <bindChannelNo>
      <!--req, int, 绑定通道号-->0
    </bindChannelNo>
    <PTZProtocolList>
      <!--req, object, PTZ协议列表-->
      <PTZProtocol>
        <!--opt, object, PTZ协议-->
        <id>
          <!--req, int, 协议索引-->0
        </id>
        <protoDesc>
          <!--req, string, 协议描述-->test
        </protoDesc>
      </PTZProtocol>
    </PTZProtocolList>
  </PTZInfo>
  <ScreenCtrlInfo>
    <!--opt, object, ScreenCtrlInfo, desc: 当workMode包含screenCtrl时有效-->
    <ScreenCtrlProtoList>
      <!--req, object, ScreenCtrlProtoList-->
      <ScreenCtrlProto>
        <!--opt, object, ScreenCtrlProto-->
        <id>
          <!--req, int, 协议索引-->0
        </id>
        <protoDesc>
          <!--req, string, 协议描述-->test
        </protoDesc>
      </ScreenCtrlProto>
    </ScreenCtrlProtoList>
  </ScreenCtrlInfo>
  <MatrixCtrlInfo>
    <!--opt, object, MatrixCtrlInfo-->
    <MatrixCtrlProtoList>
      <!--req, object, MatrixCtrlProtoList-->
      <MatrixCtrlProto>
        <!--opt, object, MatrixCtrlProto-->
        <id>
          <!--req, int, 协议索引-->0
        </id>
        <protoDesc>
          <!--req, string, 协议描述-->test
        </protoDesc>
      </MatrixCtrlProto>
    </MatrixCtrlProtoList>
  </MatrixCtrlInfo>
  <KeyBoardCtrlInfo>
    <!--opt, object, KeyBoardCtrlInfo-->
    <KeyBoardCtrlProtoList>
```

```

<!--req, object, KeyBoardCtrlProtoList-->
<KeyBoardCtrlProto>
  <!--opt, object, KeyBoardCtrlProto-->
  <id>
    <!--req, int, 协议索引-->0
  </id>
  <protoDesc>
    <!--req, string, 协议描述-->test
  </protoDesc>
</KeyBoardCtrlProto>
</KeyBoardCtrlProtoList>
</KeyBoardCtrlInfo>
<ControlAddress>
  <!--opt, object, 控制(解码器)地址-->
  <enabled>
    <!--req, bool, 使能-->true
  </enabled>
  <address>
    <!--req, int, 地址, range:[0,255]-->1
  </address>
</ControlAddress>
<SensorCtrlList>
  <!--opt, object, 外接传感器控制信息, desc:rs485WorkMode 支持sensor时返回-->
  <Sensor>
    <!--opt, object, 传感器-->
    <id>
      <!--req, int, 索引, range:[1,3]-->1
    </id>
    <name>
      <!--req, string, 传感器名称, range:[1,32], desc:visibilitySensor-能见度传感器, fiveElementWeatherSensor-气象五要素传感器, roadConditionSensor-路面状态传感器-->visibilitySensor
    </name>
    <enabled>
      <!--req, bool, 使能, desc:开启后, 传感器定时上传事件.例如: 开启能见度传感器后, 上报事件roadVisibilityAnalysis-->>false
    </enabled>
  </Sensor>
</SensorCtrlList>
<stairsControl>
  <!--opt, object, 梯控参数, dep:and,{$.SerialPort.mode,eq,stairsControl}-->
  <outputDataType>
    <!--opt, enum, 门口机对梯控主机的输出数据类型, subType:string, [fFloorNumber#楼层号(默认),cardNo#卡号]-->fFloorNumber
  </outputDataType>
</stairsControl>
<comMode>
  <!--opt, enum, 通信模式, subType:string, [active#主站模式,passive#从站模式]-->active
</comMode>
<ModbusRTU>
  <!--opt, object, 外接RTU模块参数, dep:and,{$.SerialPort.rs485WorkMode,eq,ModbusRTU}-->
  <ModbusRTUBaudRate>
    <!--opt, enum, 波特率, subType:string, [1200#1200,2400#2400,4800#4800,9600#9600,19200#19200]-->9600
  </ModbusRTUBaudRate>
  <ModbusRTUDataBits>
    <!--opt, enum, 数据位, subType:int, [5#5,6#6,7#7,8#8]-->6
  </ModbusRTUDataBits>
  <ModbusRTUParityType>
    <!--opt, enum, 奇偶校验类型, subType:string, [none#无校验位,even#偶校验,odd#奇校验]-->none
  </ModbusRTUParityType>
  <ModbusRTUStopBits>
    <!--opt, enum, 停止位, subType:string, [1#1,2#2]-->1
  </ModbusRTUStopBits>
  <ModbusRTUFlowCtrl>
    <!--opt, enum, 串口号流控制, subType:string, [none#禁用串口号流控制,software#软件串口号流控制,hardware#硬件串口号流控制]-->none
  </ModbusRTUFlowCtrl>
  <decoderAddress>
    <!--opt, int, 解码器地址, range:[0,255]-->0
  </decoderAddress>
</ModbusRTU>
<controlProtocol>
  <!--opt, string, 控制(解码器)协议-->HIKVISION
</controlProtocol>
<accessName>
  <!--opt, string, 串口接口接入名称, range:[1,32]-->test
</accessName>
<serialPortUse>
  <!--opt, enum, 串口用途, subType:string, [acquAndCtrl#采集与控制,swipe#刷卡,strongElectric#强电组网]-->acquAndCtrl
</serialPortUse>
<serialProtocol>
  <!--opt, enum, 串口协议, subType:string, [FECbus#FECbus,GB26875#GB26875]-->GB26875
</serialProtocol>
<SupplementLightList>
  <!--opt, array, 外接白光灯控制信息, subType:object, dep:and,{$.SerialPort.rs485WorkMode,eq,supplementLight}-->
  <SupplementLight>
    <!--opt, object, 外接白光灯-->
    <SupplementLightID>
      <!--req, int, 补光灯ID, range:[1,5], desc:通过/ISAPI/System/externalDevice/supplementLightStrategy?format=json配置-->1
    </SupplementLightID>
  </SupplementLight>
</SupplementLightList>
<cardReaderType>
  <!--opt, enum, 刷卡器类型, subType:string, [K1108AM#K1108AM型号], desc:只读-->K1108AM
</cardReaderType>
<strongElectricType>
  <!--opt, enum, 强电类型, subType:string, [19M04-RO/T#19M04-RO/T型号,19M08-RO/T#19M08-RO/T型号], desc:只读-->19M04-RO/T
  </strongElectricType>

```

```
</strongElectricType>
</SerialPort>
```

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 响应消息, attr:version{ro, req, string, 版本}-->
  <requestURL>
    <!--ro, req, string, 请求的URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, 状态码, subType:int, [0#OK,1#OK,2#Device Busy,3#Device Error,4#Invalid Operation,5#Invalid XML Format,6#Invalid XML Content,7#Reboot Required]-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, 状态信息, subType:string, [OK#成功,Device Busy#设备忙,Device Error#设备异常,Invalid Operation#无效的操作,Invalid XML Format#无效的XML格式,Invalid XML Content#无效的XML内容,Reboot#设备重启]-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, 详细错误码英文描述, desc: 详细错误码的英文描述-->OK
  </subStatusCode>
</ResponseStatus>
```

69.7.3 获取单个串口控制参数

Request URL

GET /ISAPI/System/Serial/ports/<portID>?permissionController=<indexID>

查询参数

参数名称	参数类型	描述
portID	string	
indexID	string	0-主权限控制器，1-从权限控制器。该参数不存在时，表示获取主权限控制器串口信息（即设备串口信息）。

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<SerialPort xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, 端口, attr:version{req, string, 协议版本}-->
  <id>
    <!--ro, req, int, 串口号-->0
  </id>
  <enabled>
    <!--ro, opt, bool, 使能-->true
  </enabled>
  <encryptKey>
    <!--ro, opt, string, 用于串口通信之间加密的密钥-->test
  </encryptKey>
  <serialNumber>
    <!--ro, opt, int, 序列号-->0
  </serialNumber>
  <serialPortType>
    <!--ro, opt, enum, 串口类型, subType:string, [RS485#RS485,RS422#RS422,RS232#RS232]-->RS485
  </serialPortType>
  <serialAddress>
    <!--ro, opt, int, 串口地址, desc: 只有串口类型为RS485时有效-->1
  </serialAddress>
  <duplexMode>
    <!--ro, opt, enum, 双工模式, subType:string, [half#半双工,full#全双工]-->half
  </duplexMode>
  <direction>
    <!--ro, opt, enum, 单向的, subType:string, [monodirectional#单向通讯,bidirectional#双向通讯]-->monodirectional
  </direction>
  <baudRate>
    <!--ro, opt, enum, 波特率, subType:int, [2400#2400,4800#4800,9600#9600,19200#19200,38400#38400,57600#57600,115200#115200]-->2400
  </baudRate>
  <dataBits>
    <!--ro, opt, int, 数据位-->6
  </dataBits>
  <parityType>
```

```
<!--ro, opt, enum, 奇偶校验类型, subType:string, [none#无校验位,even#偶校验,odd#奇校验,mark#校验位始终为1,space#校验位始终为0]-->none
</parityType>
<stopBits>
  <!--ro, opt, enum, 停止位, subType:string, [1#1,1.5#1.5,2#2]-->1
</stopBits>
<workMode>
  <!--ro, opt, enum, 工作模式, subType:string, [console#控制台,transparent#透传,audiomixe#混音,screenCtrl#屏幕控制,ptzCtrl#PTZ控制,keyboard#键盘,matrix#矩阵,audioMixers#混音台]-->console
</workMode>
<flowCtrl>
  <!--ro, opt, enum, 串口流控制, subType:string, [none#禁用串口流控制,software#软件串口流控制,hardware#硬件串口流控制]-->none
</flowCtrl>
<rs485WorkMode>
  <!--ro, opt, enum, rs485的工作模式, subType:string, [Led#外接LED屏,sensor#外接传感器,CaptureTrigger#外接触发设备进行触发抓拍数据传输,transparent#透明传输,GPS#外接GPS模块,ModbusRTU#外接RTU模块,supplementLight#白光补光灯,columnSpeaker#音柱模式,commModeUsing433Mhz#433通信], dep:or,{$.SerialPort.serialPortType,eq,RS485}-->sensor
</rs485WorkMode>
<copyToAll>
  <!--ro, opt, bool, 是否复制当前串口配置给所有串口-->true
</copyToAll>
<audiomixerProtocolType>
  <!--ro, opt, enum, HIKVISION, subType:string, [HIKVISION#HIKVISION,DS-ZH101S#DS-ZH101S]-->HIKVISION
</audiomixerProtocolType>
<deviceName>
  <!--ro, opt, string, 设备名称-->test
</deviceName>
<deviceType>
  <!--ro, opt, int, 设备类型-->1
</deviceType>
<deviceProtocol>
  <!--ro, opt, int, 设备协议-->1
</deviceProtocol>
<mode>
  <!--ro, opt, enum, 工作模式, subType:string, [readerMode#下行接读卡器,clientMode#上行输出至4200客户端,externMode#下行接门禁模块,stairsControl#上行输出至门禁主机,accessControlHost#下行接梯控主机,disabLed#禁用,cardReceiver#读卡器,QRCodeReader#二维码读卡器,disabLed#关闭,custom#自定义]-->readerMode
</mode>
<outputDataType>
  <!--ro, opt, enum, 输出数据类型, subType:string, [cardNo#卡号,employeeNo#工号], dep:and,{$.SerialPort.mode,eq,accessControlHost}-->cardNo
</outputDataType>
<isVariable>
  <!--ro, opt, bool, 是可变的-->true
</isVariable>
<PTZInfo>
  <!--ro, opt, object, 云台信息-->
  <bindChannelNo>
    <!--ro, req, int, 绑定通道号-->0
  </bindChannelNo>
  <PTZProtocolList>
    <!--ro, req, object, PTZ协议列表-->
    <PTZProtocol>
      <!--ro, opt, object, PTZ协议-->
      <id>
        <!--ro, req, int, 协议索引-->0
      </id>
      <protoDesc>
        <!--ro, req, string, 协议描述-->test
      </protoDesc>
    </PTZProtocol>
  </PTZProtocolList>
</PTZInfo>
<ScreenCtrlInfo>
  <!--ro, opt, object, ScreenCtrlInfo, desc:当workMode包含screenCtrl时有效-->
  <ScreenCtrlProtoList>
    <!--ro, req, object, ScreenCtrlProtoList-->
    <ScreenCtrlProto>
      <!--ro, opt, object, ScreenCtrlProto-->
      <id>
        <!--ro, req, int, 协议索引-->0
      </id>
      <protoDesc>
        <!--ro, req, string, 协议描述-->test
      </protoDesc>
    </ScreenCtrlProto>
  </ScreenCtrlProtoList>
</ScreenCtrlInfo>
<MatrixCtrlInfo>
  <!--ro, opt, object, MatrixCtrlInfo-->
  <MatrixCtrlProtoList>
    <!--ro, req, object, MatrixCtrlProtoList-->
    <MatrixCtrlProto>
      <!--ro, opt, object, MatrixCtrlProto-->
      <id>
        <!--ro, req, int, 协议索引-->0
      </id>
      <protoDesc>
        <!--ro, req, string, 协议描述-->test
      </protoDesc>
    </MatrixCtrlProto>
  </MatrixCtrlProtoList>
</MatrixCtrlInfo>
<KeyBoardCtrlInfo>
  <!--ro, opt, object, KeyBoardCtrlInfo-->
  <KeyBoardCtrlProtoList>
    <!--ro, req, object, KeyBoardCtrlProtoList-->
```

```
<!--ro, req, object, KeyboardCtrlProtoList-->
<KeyboardCtrlProto>
  <!--ro, opt, object, KeyboardCtrlProto-->
  <id>
    <!--ro, req, int, 协议索引-->0
  </id>
  <protoDesc>
    <!--ro, req, string, 协议描述-->test
  </protoDesc>
</KeyboardCtrlProto>
</KeyboardCtrlProtoList>
</KeyboardCtrlInfo>
<ControlAddress>
  <!--ro, opt, object, 控制（解码器）地址-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <address>
    <!--ro, req, int, 地址, range:[0,255]-->0
  </address>
</ControlAddress>
<SensorCtrlList>
  <!--ro, opt, object, 外接传感器控制信息, desc:rs485WorkMode 支持sensor时返回-->
  <Sensor>
    <!--ro, opt, object, 传感器-->
    <id>
      <!--ro, req, int, 索引, range:[1,3]-->1
    </id>
    <name>
      <!--ro, req, string, 传感器名称, range:[1,32], desc:visibilitySensor-能见度传感器, fiveElementWeatherSensor-气象五要素传感器, roadConditionSensor-路面状态传感器-->visibilitySensor
    </name>
    <enabled>
      <!--ro, req, bool, 使能, desc:开启后, 传感器定时上传事件.例如: 开启能见度传感器后, 上报事件roadVisibilityAnalysis-->>false
    </enabled>
  </Sensor>
</SensorCtrlList>
<stairsControl>
  <!--ro, opt, object, 梯控参数, dep:and,{$.SerialPort.mode,eq,stairsControl}-->
  <outputDataType>
    <!--ro, opt, enum, 门口机对梯控主机的输出数据类型, subType:string, [fFloorNumber#楼层号（默认）,cardNo#卡号]-->fFloorNumber
  </outputDataType>
</stairsControl>
<comMode>
  <!--ro, opt, enum, 通信模式, subType:string, [active#主站模式,passive#从站模式]-->active
</comMode>
<ModbusRTU>
  <!--ro, opt, object, 外接RTU模块参数, dep:and,{$.SerialPort.rs485WorkMode,eq,ModbusRTU}-->
  <ModbusRTUBaudRate>
    <!--ro, opt, enum, 波特率, subType:string, [1200#1200,2400#2400,4800#4800,9600#9600,19200#19200]-->9600
  </ModbusRTUBaudRate>
  <ModbusRTUDataBits>
    <!--ro, opt, enum, 数据位, subType:int, [5#5,6#6,7#7,8#8]-->6
  </ModbusRTUDataBits>
  <ModbusRTUParityType>
    <!--ro, opt, enum, 奇偶校验类型, subType:string, [none#无校验位,even#偶校验,odd#奇校验]-->none
  </ModbusRTUParityType>
  <ModbusRTUStopBits>
    <!--ro, opt, enum, 停止位, subType:string, [1#1,2#2]-->1
  </ModbusRTUStopBits>
  <ModbusRTUFlowCtrl>
    <!--ro, opt, enum, 串口流控制, subType:string, [none#禁用串口流控制,software#软件串口流控制,hardware#硬件串口流控制]-->none
  </ModbusRTUFlowCtrl>
  <decoderAddress>
    <!--ro, opt, int, 解码器地址, range:[0,255]-->0
  </decoderAddress>
  <online>
    <!--ro, opt, bool, 在线状态-->true
  </online>
</ModbusRTU>
<controlProtocol>
  <!--ro, opt, enum, 控制（解码器）协议, subType:string, [HIKVISION#HIKVISION,PELCO-D#PELCO-D,PELCO-P#PELCO-P]-->HIKVISION
</controlProtocol>
<accessName>
  <!--ro, opt, string, 串口接口接入名称, range:[1,32]-->test
</accessName>
<serialPortUse>
  <!--ro, opt, enum, 串口用途, subType:string, [acquAndCtrl#采集与控制,swipe#刷卡,strongElectric#强电组网]-->acquAndCtrl
</serialPortUse>
<serialProtocol>
  <!--ro, opt, enum, 串口协议, subType:string, [FECbus#FECbus,GB26875#GB26875]-->GB26875
</serialProtocol>
<SupplementLightList>
  <!--ro, opt, array, 外接白光灯控制信息, subType:object, dep:and,{$.SerialPort.rs485WorkMode,eq,supplementLight}-->
  <SupplementLight>
    <!--ro, opt, object, 外接白光灯-->
    <SupplementLightID>
      <!--ro, req, int, 补光灯ID, range:[1,5], desc:通过/ISAPI/System/externalDevice/supplementLightStrategy?format=json配置-->1
    </SupplementLightID>
  </SupplementLight>
</SupplementLightList>
<cardReaderType>
  <!--ro, opt, enum, 刷卡器类型, subType:string, [K1108AM#K1108AM型号], desc:只读-->K1108AM
</cardReaderType>
```

```
<strongElectricType>
  <!--ro, opt, enum, 强电类型, subType:string, [19M04-RO/T#19M04-RO/T型号,19M08-RO/T#19M08-RO/T型号], desc:只读-->19M04-RO/T
</strongElectricType>
</SerialPort>
```

69.7.4 获取串口能力集

Request URL

GET /ISAPI/System/Serial/capabilities

查询参数

无

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<SerialCap xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, 设备支持的485串口号范围, attr:version{req, string, 协议版本}-->
  <rs485PortNums>
    <!--ro, opt, int, 设备支持的rs485的串口最大数量-->0
  </rs485PortNums>
  <supportRS232Config>
    <!--ro, opt, bool, 设备是否支持的rs232串口相关参数配置-->true
  </supportRS232Config>
  <rs422PortNums>
    <!--ro, opt, int, 设备支持的rs422的串口最大数量-->0
  </rs422PortNums>
  <rs232PortNums>
    <!--ro, opt, int, 设备支持的rs232的串口最大数量-->0
  </rs232PortNums>
  <rs485WorkMode opt="Led, CaptureTrigger,transparent">
    <!--ro, opt, string, 设备支持的rs485的工作模式: Led-外接LED屏, attr:opt{req, string, 取值范围}-->test
  </rs485WorkMode>
  <rs232SerialNumber opt="1,2,3">
    <!--ro, opt, int, 设备支持的232串口号范围, attr:opt{req, string, 取值范围}-->1
  </rs232SerialNumber>
  <rs485SerialNumber opt="4,5,6">
    <!--ro, opt, int, 设备支持的485串口号范围, attr:opt{req, string, 取值范围}-->1
  </rs485SerialNumber>
  <isSupportAuthenticationService>
    <!--ro, opt, bool, 设备是否支持串口认证服务-->true
  </isSupportAuthenticationService>
  <isSupportDeviceInfo>
    <!--ro, opt, bool, 设备是否支持串口设备信息配置-->true
  </isSupportDeviceInfo>
  <isSupportSearchDeviceInfoRelations>
    <!--ro, opt, bool, 设备是否支持查询串口设备信息关联-->true
  </isSupportSearchDeviceInfoRelations>
  <isSupportSerialPortWorkMode>
    <!--ro, opt, bool, 是否支持设备串口模式, desc:GET/PUT /ISAPI/System/SerialPortMgr/SerialPortWorkMode?format=json-->true
  </isSupportSerialPortWorkMode>
</SerialCap>
```

69.7.5 获取单个串口控制能力集

Request URL

GET /ISAPI/System/Serial/ports/<portID>/capabilities

查询参数

参数名称	参数类型	描述
portID	string	

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>
<SerialPort xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, 端口, attr:version{req, string, 协议版本}-->
  <id>
    <!--ro, req, int, id-->0
  </id>
  <enabled opt="true,false">
    <!--ro, opt, bool, 使能, attr:opt{req, string, 取值范围}-->true
  </enabled>
  <serialPortType opt="RS485,RS422,RS232">
    <!--ro, opt, string, 串口类型, attr:opt{req, string, 取值范围}-->RS485
  </serialPortType>
  <serialAddress min="0" max="10">
    <!--ro, opt, int, 串口地址, attr:min{req, int, 最小值},max{req, int, 最大值}, desc:只有串口类型为RS485时有效-->1
  </serialAddress>
  <duplexMode opt="half,full">
    <!--ro, opt, string, 双工模式, attr:opt{req, string, 取值范围}-->half
  </duplexMode>
  <direction opt="monodirectional,bidirectional">
    <!--ro, opt, string, 单向的, attr:opt{req, string, 取值范围}-->monodirectional
  </direction>
  <baudRate opt="1200,2400,4800,9600,19200,38400,57600,115200">
    <!--ro, opt, int, 波特率, attr:opt{req, string, 取值范围}-->1200
  </baudRate>
  <dataBits opt="6,7,8">
    <!--ro, opt, int, 数据位, attr:opt{req, string, 取值范围}-->6
  </dataBits>
  <parityType opt="none,even,odd,mark,space">
    <!--ro, opt, string, 奇偶校验类型, attr:opt{req, string, 取值范围}-->none
  </parityType>
  <stopBits opt="1,1.5,2">
    <!--ro, opt, string, 停止位, attr:opt{req, string, 取值范围}-->1
  </stopBits>
  <flowCtrl opt="none,software,hardware">
    <!--ro, opt, string, flowCtrl, attr:opt{req, string, 取值范围}-->none
  </flowCtrl>
  <deviceName min="0" max="32">
    <!--ro, opt, string, 设备名称, attr:min{req, int, 最小值},max{req, int, 最大值}-->test
  </deviceName>
  <mode opt="readerMode,clientMode,externMode,stairsControl,accessControlHost,disabled,custom,cardReceiver,QRCodeReader">
    <!--ro, opt, string, 工作模式, attr:opt{req, string, 取值范围}-->readerMode
  </mode>
  <outputDataType opt="cardNo,employeeNo,auto">
    <!--ro, opt, string, 输出数据类型, attr:opt{req, string, 取值范围}, desc:if < mode >= accessControlHost, xs:string, ;工作模式是accessControlHost时有效-->cardNo
  </outputDataType>
  <stairsControl>
    <!--ro, opt, object, 梯控参数-->
    <outputDataType opt="floorNumber,cardNo">
      <!--ro, opt, enum, 门口机对梯控主机的输出数据类型, subType:string, [floorNumber#楼层号（默认）,cardNo#卡号], attr:opt{req, string, 取值范围}-->floorNumber
    </outputDataType>
  </stairsControl>
</SerialPort>
```

69.7.6 获取单个串口控制参数

Request URL

GET /ISAPI/System/Serial/ports/<portID>?permissionController=<indexID>

查询参数

参数名称	参数类型	描述
portID	string	
indexID	string	0-主权限控制器，1-从权限控制器。该参数不存在时，表示获取主权限控制器串口信息（即设备串口信息）

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>
<SerialPort xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, 端口, attr:version{req, string, 协议版本}-->
  <id>
    <!--ro, req, int, 串口号-->0
```

```
</id>
<enabled>
  <!--ro, opt, bool, 使能-->true
</enabled>
<encryptKey>
  <!--ro, opt, string, 用于串口通信之间加密的密钥-->test
</encryptKey>
<serialNumber>
  <!--ro, opt, int, 序列号-->0
</serialNumber>
<serialPortType>
  <!--ro, opt, enum, 串口类型, subType:string, [RS485#RS485,RS422#RS422,RS232#RS232]-->RS485
</serialPortType>
<serialAddress>
  <!--ro, opt, int, 串口地址, desc:只有串口类型为RS485时有效-->1
</serialAddress>
<duplexMode>
  <!--ro, opt, enum, 双工模式, subType:string, [half#半双工,full#全双工]-->half
</duplexMode>
<direction>
  <!--ro, opt, enum, 单向的, subType:string, [monodirectional#单向通讯,bidirectional#双向通讯]-->monodirectional
</direction>
<baudRate>
  <!--ro, opt, enum, 波特率, subType:int, [2400#2400,4800#4800,9600#9600,19200#19200,38400#38400,57600#57600,115200#115200]-->2400
</baudRate>
<dataBits>
  <!--ro, opt, int, 数据位-->6
</dataBits>
<parityType>
  <!--ro, opt, enum, 奇偶校验类型, subType:string, [none#无校验位,even#偶校验,odd#奇校验,mark#校验位始终为1,space#校验位始终为0]-->none
</parityType>
<stopBits>
  <!--ro, opt, enum, 停止位, subType:string, [1#1,1.5#1.5,2#2]-->1
</stopBits>
<workMode>
  <!--ro, opt, enum, 工作模式, subType:string, [console#控制台,transparent#透传,audiomix#混音,screenCtrl#屏幕控制,ptzCtrl#PTZ控制,keyboard#键盘,matrix#矩阵,audioMixers#混音台]-->console
</workMode>
<flowCtrl>
  <!--ro, opt, enum, 串口流控制, subType:string, [none#禁用串口流控制,software#软件串口流控制,hardware#硬件串口流控制]-->none
</flowCtrl>
<rs485WorkMode>
  <!--ro, opt, enum, rs485的工作模式, subType:string, [Led#外接LED屏,sensor#外接传感器,CaptureTrigger#外接触发设备进行触发抓拍数据传输,transparent#透明传输,GPS#外接GPS模块,ModbusRTU#外接RTU模块,supplementLight#白光补光灯,columnSpeaker#音柱模式,commModeUsing433MHz#433通信], dep:or, {$.SerialPort.serialPortType,eq,RS485}-->sensor
</rs485WorkMode>
<copyToAll>
  <!--ro, opt, bool, 是否复制当前串口配置给所有串口-->true
</copyToAll>
<audiomixerProtocolType>
  <!--ro, opt, enum, HIKVISION, subType:string, [HIKVISION#HIKVISION,DS-ZH101S#DS-ZH101S]-->HIKVISION
</audiomixerProtocolType>
<deviceName>
  <!--ro, opt, string, 设备名称-->test
</deviceName>
<deviceType>
  <!--ro, opt, int, 设备类型-->1
</deviceType>
<deviceProtocol>
  <!--ro, opt, int, 设备协议-->1
</deviceProtocol>
<mode>
  <!--ro, opt, enum, 工作模式, subType:string, [readerMode#下行读卡器,clientMode#上行输出至4200客户端,externMode#下行接门控模块,stairsControl#下行接楼梯控主机,accessControlHost#上行输出至门禁主机,cardReceiver#读卡器,QRCoder#二维码读卡器,disabLed#禁用,custom#自定义]-->readerMode
</mode>
<outputDataType>
  <!--ro, opt, enum, 输出数据类型, subType:string, [cardNo#卡号,employeeNo#工号,auto#自动], dep:and,{$.SerialPort.mode,eq,accessControlHost}-->cardNo
</outputDataType>
<isVariable>
  <!--ro, opt, bool, 是可变的-->true
</isVariable>
<PTZInfo>
  <!--ro, opt, object, 云台信息-->
  <bindChannelNo>
    <!--ro, req, int, 绑定通道号-->0
  </bindChannelNo>
  <PTZProtocolList>
    <!--ro, req, object, PTZ协议列表-->
    <PTZProtocol>
      <!--ro, opt, object, PTZ协议-->
      <id>
        <!--ro, req, int, 协议索引-->0
      </id>
      <protoDesc>
        <!--ro, req, string, 协议描述-->test
      </protoDesc>
    </PTZProtocol>
  </PTZProtocolList>
</PTZInfo>
<ScreenCtrlInfo>
  <!--ro, opt, object, ScreenCtrlInfo, desc:当workMode包含screenCtrl时有效-->
  <ScreenCtrlProtoList>
    <!--ro, req, object, ScreenCtrlProtoList-->
  </ScreenCtrlProtoList>
```

```

<ScreenCtrlProto>
  <!--ro, opt, object, ScreenCtrlProto-->
  <id>
    <!--ro, req, int, 协议索引-->0
  </id>
  <protoDesc>
    <!--ro, req, string, 协议描述-->test
  </protoDesc>
</ScreenCtrlProto>
</ScreenCtrlProtoList>
<ScreenCtrlInfo>
<MatrixCtrlInfo>
  <!--ro, opt, object, MatrixCtrlInfo-->
  <MatrixCtrlProtoList>
    <!--ro, req, object, MatrixCtrlProtoList-->
    <MatrixCtrlProto>
      <!--ro, opt, object, MatrixCtrlProto-->
      <id>
        <!--ro, req, int, 协议索引-->0
      </id>
      <protoDesc>
        <!--ro, req, string, 协议描述-->test
      </protoDesc>
    </MatrixCtrlProto>
  </MatrixCtrlProtoList>
</MatrixCtrlInfo>
<KeyboardCtrlInfo>
  <!--ro, opt, object, KeyboardCtrlInfo-->
  <KeyboardCtrlProtoList>
    <!--ro, req, object, KeyboardCtrlProtoList-->
    <KeyboardCtrlProto>
      <!--ro, opt, object, KeyboardCtrlProto-->
      <id>
        <!--ro, req, int, 协议索引-->0
      </id>
      <protoDesc>
        <!--ro, req, string, 协议描述-->test
      </protoDesc>
    </KeyboardCtrlProto>
  </KeyboardCtrlProtoList>
</KeyboardCtrlInfo>
<ControlAddress>
  <!--ro, opt, object, 控制(解码器)地址-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <address>
    <!--ro, req, int, 地址, range:[0,255]-->0
  </address>
</ControlAddress>
<SensorCtrlList>
  <!--ro, opt, object, 外接传感器控制信息, desc:rs485WorkMode 支持sensor时返回-->
  <Sensor>
    <!--ro, opt, object, 传感器-->
    <id>
      <!--ro, req, int, 索引, range:[1,3]-->1
    </id>
    <name>
      <!--ro, req, string, 传感器名称, range:[1,32], desc:visibilitySensor-能见度传感器, fiveElementWeatherSensor-气象五要素传感器, roadConditionSensor-路面状态传感器-->visibilitySensor
    </name>
    <enabled>
      <!--ro, req, bool, 使能, desc:开启后, 传感器定时上传事件, 例如: 开启能见度传感器后, 上报事件roadVisibilityAnalysis-->>false
    </enabled>
  </Sensor>
</SensorCtrlList>
<stairsControl>
  <!--ro, opt, object, 梯控参数, dep:and,{$.SerialPort.mode,eq,stairsControl}-->
  <outputDataType>
    <!--ro, opt, enum, 门口机对梯控主机的输出数据类型, subType:string, [fFloorNumber#楼层号(默认),cardNo#卡号]-->fFloorNumber
  </outputDataType>
</stairsControl>
<comMode>
  <!--ro, opt, enum, 通信模式, subType:string, [active#主站模式,passive#从站模式]-->active
</comMode>
<ModbusRTU>
  <!--ro, opt, object, 外接RTU模块参数, dep:and,{$.SerialPort.rs485WorkMode,eq,ModbusRTU}-->
  <ModbusRTUBaudRate>
    <!--ro, opt, enum, 波特率, subType:string, [1200#1200,2400#2400,4800#4800,9600#9600,19200#19200]-->9600
  </ModbusRTUBaudRate>
  <ModbusRTUDataBits>
    <!--ro, opt, enum, 数据位, subType:int, [5#5,6#6,7#7,8#8]-->6
  </ModbusRTUDataBits>
  <ModbusRTUParityType>
    <!--ro, opt, enum, 奇偶校验类型, subType:string, [none#无校验位,even#偶校验,odd#奇校验]-->none
  </ModbusRTUParityType>
  <ModbusRTUStopBits>
    <!--ro, opt, enum, 停止位, subType:string, [1#1,2#2]-->1
  </ModbusRTUStopBits>
  <ModbusRTUFlowCtrl>
    <!--ro, opt, enum, 串口流控制, subType:string, [none#禁用串口流控制,software#软件串口流控制,hardware#硬件串口流控制]-->none
  </ModbusRTUFlowCtrl>
  <decoderAddress>
    <!--ro, opt, int, 解码器地址, range:[0,255]-->0

```

```
</decoderAddress>
<online>
  <!--ro, opt, bool, 在线状态-->true
</online>
</ModbusRTU>
<controlProtocol>
  <!--ro, opt, enum, 控制（解码器）协议, subType:string, [HIKVISION#HIKVISION, PELCO-D#PELCO-D, PELCO-P#PELCO-P]-->HIKVISION
</controlProtocol>
<accessName>
  <!--ro, opt, string, 串口接口接入名称, range:[1,32]-->test
</accessName>
<serialPortUse>
  <!--ro, opt, enum, 串口用途, subType:string, [acquAndCtrl#采集与控制, swipe#刷卡, strongElectric#强电组网]-->acquAndCtrl
</serialPortUse>
<serialProtocol>
  <!--ro, opt, enum, 串口协议, subType:string, [FECbus#FECbus, GB26875#GB26875]-->GB26875
</serialProtocol>
<SupplementLightList>
  <!--ro, opt, array, 外接白光灯控制信息, subType:object, dep:and,{$.SerialPort.rs485WorkMode, eq, supplementLight}-->
  <SupplementLight>
    <!--ro, opt, object, 外接白光灯-->
    <SupplementLightID>
      <!--ro, req, int, 补光灯ID, range:[1,5], desc:通过/ISAPI/System/externalDevice/supplementLightStrategy?format=json配置-->1
    </SupplementLightID>
  </SupplementLight>
</SupplementLightList>
<cardReaderType>
  <!--ro, opt, enum, 刷卡器类型, subType:string, [K1108AM#K1108AM型号], desc:只读-->K1108AM
</cardReaderType>
<strongElectricType>
  <!--ro, opt, enum, 强电类型, subType:string, [19M04-RO/T#19M04-RO/T型号, 19M08-RO/T#19M08-RO/T型号], desc:只读-->19M04-RO/T
</strongElectricType>
</SerialPort>
```

69.7.7 获取全部串口控制参数

Request URL

GET /ISAPI/System/Serial/ports?permissionController= <indexID>

查询参数

参数名称	参数类型	描述
indexID	string	0-主权限控制器，1-从权限控制器。该参数不存在时，表示获取主权限控制器串口信息（即设备串口信息）

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<SerialPortList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, SerialPortList, attr:version{req, string, 协议版本}-->
  <SerialPort>
    <!--ro, opt, object, 端口-->
    <id>
      <!--ro, req, int, 串口号-->0
    </id>
    <enabled>
      <!--ro, opt, bool, 使能-->true
    </enabled>
    <encryptKey>
      <!--ro, opt, string, 用于串口通信之间加密的密钥-->test
    </encryptKey>
    <serialNumber>
      <!--ro, opt, int, 序列号-->0
    </serialNumber>
    <serialPortType>
      <!--ro, opt, enum, 串口类型, subType:string, [RS485#RS485, RS422#RS422, RS232#RS232]-->RS485
    </serialPortType>
    <serialAddress>
      <!--ro, opt, int, 串口地址, desc:只有串口类型为RS485时有效-->1
    </serialAddress>
    <duplexMode>
      <!--ro, opt, enum, 双工模式, subType:string, [half#半双工, full#全双工]-->half
    </duplexMode>
    <direction>
      <!--ro, opt, enum, 单向的, subType:string, [monodirectional#单向通讯, bidirectional#双向通讯]-->monodirectional
    </direction>
    <baudRate>
```

```
<!--ro, opt, enum, 波特率, subType:int, [2400#2400,4800#4800,9600#9600,19200#19200,38400#38400,57600#57600,115200#115200]-->2400
</baudRate>
<dataBits>
<!--ro, opt, int, 数据位-->6
</dataBits>
<parityType>
<!--ro, opt, enum, 奇偶校验类型, subType:string, [none#无校验位,even#偶校验,odd#奇校验,mark#校验位始终为1,space#校验位始终为0]-->none
</parityType>
<stopBits>
<!--ro, opt, enum, 停止位, subType:string, [1#1,1.5#1.5,2#2]-->1
</stopBits>
<workMode>
<!--ro, opt, enum, 工作模式, subType:string, [console#控制台,transparent#透传,audiomix#混音,screenCtrl#屏幕控制,ptzCtrl#PTZ控制,keyboard#键盘,matrix#
矩阵]-->console
</workMode>
<flowCtrl>
<!--ro, opt, enum, 串口号控制, subType:string, [none#禁用串口号控制,software#软件串口号控制,hardware#硬件串口号控制]-->none
</flowCtrl>
<rs485WorkMode>
<!--ro, opt, enum, rs485的工作模式, subType:string, [Led#外接LED屏,sensor#外接传感器,CaptureTrigger#外接触发设备进行触发抓拍数据传输,transparent#透明
传输,GPS#外接GPS模块,ModbusRTU#外接RTU模块,supplementLight#白光补光灯], dep:or,{$.SerialPort.serialPortType,eq,RS485}-->sensor
</rs485WorkMode>
<copyToAll>
<!--ro, opt, bool, 是否复制当前串口配置给所有串口-->true
</copyToAll>
<audiomixerProtocolType>
<!--ro, opt, enum, HIKVISION, subType:string, [HIKVISION#HIKVISION,DS-ZH101S#DS-ZH101S]-->HIKVISION
</audiomixerProtocolType>
<deviceName>
<!--ro, opt, string, 设备名称-->test
</deviceName>
<deviceType>
<!--ro, opt, int, 设备类型-->1
</deviceType>
<deviceProtocol>
<!--ro, opt, int, 设备协议-->1
</deviceProtocol>
<mode>
<!--ro, opt, enum, 工作模式, subType:string, [readerMode#下行接读卡器,clientMode#上行输出至4200客户端,externMode#下行接门禁模块,stairsControl#上行输出
至门禁主机,accessControlHost#下行接梯控主机,disabled#禁用,cardReceiver#读卡器,QRCoderReader#二维码读卡器,disabled#关闭,custom#自定义]-->readerMode
</mode>
<outputDataType>
<!--ro, opt, enum, 输出数据类型, subType:string, [cardNo#卡号,empLoyeeNo#工号,auto#自动], dep:and,{$.SerialPort.mode,eq,accessControlHost}-->cardNo
</outputDataType>
<isVariable>
<!--ro, opt, bool, 是否可变-->true
</isVariable>
<PTZInfo>
<!--ro, opt, object, 云台信息-->
<bindChannelNo>
<!--ro, req, int, 绑定通道号-->0
</bindChannelNo>
<PTZProtocolList>
<!--ro, req, object, PTZ协议列表-->
<PTZProtocol>
<!--ro, opt, object, PTZ协议-->
<id>
<!--ro, req, int, 协议索引-->0
</id>
<protoDesc>
<!--ro, req, string, 协议描述-->test
</protoDesc>
</PTZProtocol>
</PTZProtocolList>
</PTZInfo>
<ScreenCtrlInfo>
<!--ro, opt, object, ScreenCtrlInfo, desc:当workMode包含screenCtrl时有效-->
<ScreenCtrlProtoList>
<!--ro, req, object, ScreenCtrlProtoList-->
<ScreenCtrlProto>
<!--ro, opt, object, ScreenCtrlProto-->
<id>
<!--ro, req, int, 协议索引-->0
</id>
<protoDesc>
<!--ro, req, string, 协议描述-->test
</protoDesc>
</ScreenCtrlProto>
</ScreenCtrlProtoList>
</ScreenCtrlInfo>
<MatrixCtrlInfo>
<!--ro, opt, object, MatrixCtrlInfo-->
<MatrixCtrlProtoList>
<!--ro, req, object, MatrixCtrlProtoList-->
<MatrixCtrlProto>
<!--ro, opt, object, MatrixCtrlProto-->
<id>
<!--ro, req, int, 协议索引-->0
</id>
<protoDesc>
<!--ro, req, string, 协议描述-->test
</protoDesc>
</MatrixCtrlProto>
</MatrixCtrlInfo>
```

```

</MatrixCtrlInfo>
<KeyboardCtrlInfo>
  <!--ro, opt, object, KeyboardCtrlInfo-->
  <KeyboardCtrlProtoList>
    <!--ro, req, object, KeyboardCtrlProtoList-->
    <KeyboardCtrlProto>
      <!--ro, opt, object, KeyboardCtrlProto-->
      <id>
        <!--ro, req, int, 协议索引-->0
      </id>
      <protoDesc>
        <!--ro, req, string, 协议描述-->test
      </protoDesc>
    </KeyboardCtrlProto>
  </KeyboardCtrlProtoList>
</KeyboardCtrlInfo>
<ControlAddress>
  <!--ro, opt, object, 控制（解码器）地址-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <address>
    <!--ro, req, int, 地址, range:[0,255]-->1
  </address>
</ControlAddress>
<SensorCtrlList>
  <!--ro, opt, object, 外接传感器控制信息, desc:rs485WorkMode 支持sensor时返回-->
  <Sensor>
    <!--ro, opt, object, 传感器-->
    <id>
      <!--ro, req, int, 索引, range:[1,3]-->1
    </id>
    <name>
      <!--ro, req, string, 传感器名称, range:[1,32], desc:visibilitySensor-能见度传感器, fiveElementWeatherSensor-气象五要素传感器, roadConditionSensor-路面状态传感器-->visibilitySensor
    </name>
    <enabled>
      <!--ro, req, bool, 使能, desc:开启后, 传感器定时上传事件.例如: 开启能见度传感器后, 上报事件roadVisibilityAnalysis-->>false
    </enabled>
  </Sensor>
</SensorCtrlList>
<stairsControl>
  <!--ro, opt, object, 梯控参数, dep:and,{$.SerialPort.mode,eq,stairsControl}-->
  <outputDataType>
    <!--ro, opt, enum, 门口机对梯控主机的输出数据类型, subType:string, [fFloorNumber#楼层号（默认）,cardNo#卡号]-->fFloorNumber
  </outputDataType>
</stairsControl>
<comMode>
  <!--ro, opt, enum, 通信模式, subType:string, [active#主站模式,passive#从站模式]-->active
</comMode>
<ModbusRTU>
  <!--ro, opt, object, 外接RTU模块参数, dep:and,{$.SerialPortList.SerialPort.rs485WorkMode,eq,ModbusRTU}-->
  <ModbusRTUBaudRate>
    <!--ro, opt, enum, 波特率, subType:string, [1200#1200,2400#2400,4800#4800,9600#9600,19200#19200]-->9600
  </ModbusRTUBaudRate>
  <ModbusRTUDataBits>
    <!--ro, opt, enum, 数据位, subType:int, [5#5,6#6,7#7,8#8]-->6
  </ModbusRTUDataBits>
  <ModbusRTUParityType>
    <!--ro, opt, enum, 奇偶校验类型, subType:string, [none#无校验位,even#偶校验,odd#奇校验]-->none
  </ModbusRTUParityType>
  <ModbusRTUStopBits>
    <!--ro, opt, enum, 停止位, subType:string, [1#1,2#2]-->1
  </ModbusRTUStopBits>
  <ModbusRTUFlowCtrl>
    <!--ro, opt, enum, 串口流控制, subType:string, [none#禁用串口流控制,software#软件串口流控制,hardware#硬件串口流控制]-->none
  </ModbusRTUFlowCtrl>
  <decoderAddress>
    <!--ro, opt, int, 解码器地址, range:[0,255]-->0
  </decoderAddress>
  <online>
    <!--ro, opt, bool, 在线状态-->true
  </online>
</ModbusRTU>
<controlProtocol>
  <!--ro, opt, enum, 控制（解码器）协议, subType:string, [HIKVISION#HIKVISION,PELCO-D#PELCO-D,PELCO-P#PELCO-P]-->HIKVISION
</controlProtocol>
<accessName>
  <!--ro, opt, string, 串口接口接入名称, range:[1,32]-->test
</accessName>
<serialPortUse>
  <!--ro, opt, enum, 串口用途, subType:string, [acquAndCtrl#采集与控制,swipe#刷卡,strongElectric#强电组网]-->acquAndCtrl
</serialPortUse>
<serialProtocol>
  <!--ro, opt, enum, 串口协议, subType:string, [FECbus#FECbus,GB26875#GB26875]-->GB26875
</serialProtocol>
<SupplementLightList>
  <!--ro, opt, array, 外接白光灯控制信息, subType:object, dep:and,{$.SerialPort.rs485WorkMode,eq,supplementLight}-->
  <SupplementLight>
    <!--ro, opt, object, 外接白光灯-->
    <SupplementLightID>
      <!--ro, req, int, 补光灯ID, range:[1,5], desc:通过/ISAPI/System/externalDevice/supplementLightStrategy?format=json配置-->1
    </SupplementLightID>
  </SupplementLight>

```

```
</SupplementLight>
</SupplementLightList>
<cardReaderType>
  <!--ro, opt, enum, 刷卡器类型, subType:string, [K1108AM#K1108AM型号], desc:只读-->K1108AM
</cardReaderType>
<strongElectricType>
  <!--ro, opt, enum, 强电类型, subType:string, [19M04-RO/T#19M04-RO/T型号,19M08-RO/T#19M08-RO/T型号], desc:只读-->19M04-RO/T
</strongElectricType>
</SerialPort>
</SerialPortList>
```

69.7.8 配置单个串口控制参数

Request URL

PUT /ISAPI/System/Serial/ports/<portID>?permissionController=<indexID>

查询参数

参数名称	参数类型	描述
portID	string	
indexID	string	0-主权限控制器，1-从权限控制器。该参数不存在时，表示配置主权限控制器串口信息（即设备串口信息）

请求报文

```
<?xml version="1.0" encoding="UTF-8"?>

<SerialPort xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--opt, object, 端口, attr:version{req, string, 协议版本}-->
  <id>
    <!--req, int, 串口号-->0
  </id>
  <enabled>
    <!--opt, bool, 使能-->>true
  </enabled>
  <encryptKey>
    <!--opt, string, 用于串口通信之间加密的秘钥-->test
  </encryptKey>
  <serialNumber>
    <!--opt, int, 序列号-->0
  </serialNumber>
  <serialPortType>
    <!--opt, enum, 串口类型, subType:string, [RS485#RS485,RS422#RS422,RS232#RS232]-->RS485
  </serialPortType>
  <serialAddress>
    <!--opt, int, 串口地址, desc:只有串口类型为RS485时有效-->1
  </serialAddress>
  <duplexMode>
    <!--opt, enum, 双工模式, subType:string, [half#半双工,full#全双工]-->half
  </duplexMode>
  <direction>
    <!--opt, enum, 单向的, subType:string, [monodirectional#单向通讯,bidirectional#双向通讯]-->monodirectional
  </direction>
  <baudRate>
    <!--opt, enum, 波特率, subType:int, [2400#2400,4800#4800,9600#9600,19200#19200,38400#38400,57600#57600,115200#115200]-->2400
  </baudRate>
  <dataBits>
    <!--opt, int, 数据位-->6
  </dataBits>
  <parityType>
    <!--opt, enum, 奇偶校验类型, subType:string, [none#无校验位,even#偶校验,odd#奇校验,mark#校验位始终为1,space#校验位始终为0]-->none
  </parityType>
  <stopBits>
    <!--opt, string, 停止位-->1
  </stopBits>
  <workMode>
    <!--opt, enum, 工作模式, subType:string, [console#控制台,transparent#透传,audiomixe#混音,screenCtrl#屏幕控制,ptzCtrl#PTZ控制,keyboard#键盘,matrix#矩阵]->console
  </workMode>
  <flowCtrl>
    <!--opt, enum, 串口号流控制, subType:string, [none#禁用串口号流控制,software#软件串口号流控制,hardware#硬件串口号流控制]-->none
  </flowCtrl>
  <rs485WorkMode>
    <!--opt, enum, rs485的工作模式, subType:string, [Led#外接LED屏,sensor#外接传感器,CaptureTrigger#外接触发设备进行触发抓拍数据传输,transparent#透明传输,GPS#外接GPS模块,ModbusRTU#外接RTU模块,supplementLight#白光补光灯,columnSpeaker#音柱模式,commModeUsing433MHz#433模块], dep:or,
    {$.SerialPort.serialPortType,eq,RS485}-->sensor
  </rs485WorkMode>
  <copyToAll>
    <!--opt, bool, 是否复制当前串口配置给所有串口-->true
  </copyToAll>
  <audiomixerProtocolTune>
```

```
<!--opt, enum, HIKVISION, subType:string, [HIKVISION#HIKVISION,DS-ZH101S#DS-ZH101S]-->HIKVISION
</audiomixerProtocolType>
<deviceName>
  <!--ro, opt, string, 设备名称-->test
</deviceName>
<deviceType>
  <!--ro, opt, int, 设备类型-->1
</deviceType>
<deviceProtocol>
  <!--ro, opt, int, 设备协议-->1
</deviceProtocol>
<mode>
  <!--opt, enum, 工作模式, subType:string, [readerMode#下行接读卡器,clientMode#上行输出至4200客户端,externMode#下行接门禁模块,stairControl#下行接梯控主机,accessControlHost#上行输出至门禁主机,cardReceiver#收卡器,QRCodeReader#二维码读卡器,disabed#禁用,custom#自定义]-->readerMode
</mode>
<outputDataType>
  <!--opt, enum, 输出数据类型, subType:string, [cardNo#卡号,employeeNo#工号,auto#自动], dep:and,{$.SerialPort.mode,eq,accessControlHost}, desc:配置成自动模式后, 输出数据类型取决于上行门禁主机的认证方式.-->cardNo
</outputDataType>
<isVariable>
  <!--opt, bool, 是可变的-->true
</isVariable>
<PTZInfo>
  <!--opt, object, 云台信息-->
  <bindChannelNo>
    <!--req, int, 绑定通道号-->0
  </bindChannelNo>
  <PTZProtocolList>
    <!--req, object, PTZ协议列表-->
    <PTZProtocol>
      <!--opt, object, PTZ协议-->
      <id>
        <!--req, int, 协议索引-->0
      </id>
      <protoDesc>
        <!--req, string, 协议描述-->test
      </protoDesc>
    </PTZProtocol>
  </PTZProtocolList>
</PTZInfo>
<ScreenCtrlInfo>
  <!--opt, object, ScreenCtrlInfo, desc:当workMode包含screenCtrl时有效-->
  <ScreenCtrlProtoList>
    <!--req, object, ScreenCtrlProtoList-->
    <ScreenCtrlProto>
      <!--opt, object, ScreenCtrlProto-->
      <id>
        <!--req, int, 协议索引-->0
      </id>
      <protoDesc>
        <!--req, string, 协议描述-->test
      </protoDesc>
    </ScreenCtrlProto>
  </ScreenCtrlProtoList>
</ScreenCtrlInfo>
<MatrixCtrlInfo>
  <!--opt, object, MatrixCtrlInfo-->
  <MatrixCtrlProtoList>
    <!--req, object, MatrixCtrlProtoList-->
    <MatrixCtrlProto>
      <!--opt, object, MatrixCtrlProto-->
      <id>
        <!--req, int, 协议索引-->0
      </id>
      <protoDesc>
        <!--req, string, 协议描述-->test
      </protoDesc>
    </MatrixCtrlProto>
  </MatrixCtrlProtoList>
</MatrixCtrlInfo>
<KeyBoardCtrlInfo>
  <!--opt, object, KeyBoardCtrlInfo-->
  <KeyBoardCtrlProtoList>
    <!--req, object, KeyBoardCtrlProtoList-->
    <KeyBoardCtrlProto>
      <!--opt, object, KeyBoardCtrlProto-->
      <id>
        <!--req, int, 协议索引-->0
      </id>
      <protoDesc>
        <!--req, string, 协议描述-->test
      </protoDesc>
    </KeyBoardCtrlProto>
  </KeyBoardCtrlProtoList>
</KeyBoardCtrlInfo>
<ControlAddress>
  <!--opt, object, 控制(解码器)地址-->
  <enabled>
    <!--req, bool, 使能-->true
  </enabled>
  <address>
    <!--req, int, 地址, range:[0,255]-->1
  </address>
```

```

</ControlAddress>
<SensorCtrlList>
  <!--opt, object, 外接传感器控制信息, desc:rs485WorkMode 支持sensor时返回-->
  <Sensor>
    <!--opt, object, 传感器-->
    <id>
      <!--req, int, 索引, range:[1,3]-->1
    </id>
    <name>
      <!--req, string, 传感器名称, range:[1,32], desc:visibilitySensor~能见度传感器, fiveElementWeatherSensor~气象五要素传感器, roadConditionSensor~路面状态传感器-->visibilitySensor
    </name>
    <enabled>
      <!--req, bool, 使能, desc:开启后, 传感器定时上传事件.例如: 开启能见度传感器后, 上报事件roadVisibilityAnalysis-->>false
    </enabled>
  </Sensor>
</SensorCtrlList>
<stairsControl>
  <!--opt, object, 梯控参数, dep:and,{$.SerialPort.mode,eq,stairsControl}-->
  <outputDataType>
    <!--opt, enum, 门口机对梯控主机的输出数据类型, subType:string, [fFloorNumber#楼层号(默认),cardNo#卡号]-->fFloorNumber
  </outputDataType>
</stairsControl>
<comMode>
  <!--opt, enum, 通信模式, subType:string, [active#主站模式,passive#从站模式]-->active
</comMode>
<ModbusRTU>
  <!--opt, object, 外接RTU模块参数, dep:and,{$.SerialPort.rs485WorkMode,eq,ModbusRTU}-->
  <ModbusRTUBaudRate>
    <!--opt, enum, 波特率, subType:string, [1200#1200,2400#2400,4800#4800,9600#9600,19200#19200]-->9600
  </ModbusRTUBaudRate>
  <ModbusRTUDataBits>
    <!--opt, enum, 数据位, subType:int, [5#5,6#6,7#7,8#8]-->6
  </ModbusRTUDataBits>
  <ModbusRTUParityType>
    <!--opt, enum, 奇偶校验类型, subType:string, [none#无校验位,even#偶校验,odd#奇校验]-->none
  </ModbusRTUParityType>
  <ModbusRTUStopBits>
    <!--opt, enum, 停止位, subType:string, [1#1,2#2]-->1
  </ModbusRTUStopBits>
  <ModbusRTUFlowCtrl>
    <!--opt, enum, 串口流控制, subType:string, [none#禁用串口流控制,software#软件串口流控制,hardware#硬件串口流控制]-->none
  </ModbusRTUFlowCtrl>
  <decoderAddress>
    <!--opt, int, 解码器地址, range:[0,255]-->0
  </decoderAddress>
</ModbusRTU>
<controlProtocol>
  <!--opt, string, 控制(解码器)协议-->HIKVISION
</controlProtocol>
<accessName>
  <!--opt, string, 串口接口接入名称, range:[1,32]-->test
</accessName>
<serialPortUse>
  <!--opt, enum, 串口用途, subType:string, [acquAndCtrl#采集与控制,swipe#刷卡,strongElectric#强电组网]-->acquAndCtrl
</serialPortUse>
<serialProtocol>
  <!--opt, enum, 串口协议, subType:string, [FECbus#FECbus,GB26875#GB26875]-->GB26875
</serialProtocol>
<SupplementLightList>
  <!--opt, array, 外接白光灯控制信息, subType:object, dep:and,{$.SerialPort.rs485WorkMode,eq,supplementLight}-->
  <SupplementLight>
    <!--opt, object, 外接白光灯-->
    <SupplementLightID>
      <!--req, int, 补光灯ID, range:[1,5], desc:通过/ISAPI/System/externalDevice/supplementLightStrategy?format=json配置-->1
    </SupplementLightID>
  </SupplementLight>
</SupplementLightList>
<cardReaderType>
  <!--opt, enum, 刷卡器类型, subType:string, [K1108AM#K1108AM型号], desc:只读-->K1108AM
</cardReaderType>
<strongElectricType>
  <!--opt, enum, 强电类型, subType:string, [19M04-RO/T#19M04-RO/T型号,19M08-RO/T#19M08-RO/T型号], desc:只读-->19M04-RO/T
</strongElectricType>
</SerialPort>

```

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 响应消息, attr:version{ro, req, string, 版本}-->
  <requestURL>
    <!--ro, req, string, 请求的URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, 状态码, subType:int, [0#OK,1#OK,2#Device Busy,3#Device Error,4#Invalid Operation,5#Invalid XML Format,6#Invalid XML Content,7#Reboot Required]-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, 状态信息, subType:string, [OK#成功,Device Busy#设备忙,Device Error#设备异常,Invalid Operation#无效的操作,Invalid XML Format#无效的XML格式,Invalid XML Content#无效的XML内容,Reboot#设备重启]-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, 详细错误码英文描述, desc:详细错误码的英文描述-->OK
  </subStatusCode>
</ResponseStatus>
```

69.8 协议接入(S)

69.8.1 获取设备接入协议列表能力

Request URL

GET /ISAPI/Security/adminAccesses/capabilities

查询参数

无

请求报文

无

响应报文

湖南云舟电力科技有限公司
有效期：2024-08-15

```

<?xml version="1.0" encoding="UTF-8"?>

<AdminAccessProtocolList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, array, 设备接入协议列表对象, subType:object, attr:version{req, string, 协议版本}-->
  <AdminAccessProtocol>
    <!--ro, req, object, 设备接入协议-->
    <id>
      <!--ro, req, string, 序号ID-->test
    </id>
    <enabled opt="true,false">
      <!--ro, opt, bool, 使能, attr:opt{req, string, 取值范围}-->true
    </enabled>
    <protocol opt="HTTP,HTTPS,SDK,RTSP,DEV_MANAGE,WebSocket,WebSocketS,SDK_OVER_TLS,MATRIX_GATEWAY,IOT,GB35114,104Proto,SRTP,ModbusTCP">
      <!--ro, req, string, HTTPS, attr:opt{req, string, 协议类型}-->GB35114
    </protocol>
    <redirectToHttps opt="true,false">
      <!--ro, opt, bool, 是否开启https重定向, attr:opt{req, string, 取值范围}-->true
    </redirectToHttps>
    <streamOverTls opt="true,false">
      <!--ro, opt, bool, 当protocol选择为SDK_OVER_TLS时, attr:opt{req, string, 取值范围}-->true
    </streamOverTls>
    <portNo min="1" max="65535" default="8443">
      <!--ro, req, int, 当前协议对应的端口号;协议类型通过protocol字段获取, range:[1,65535], attr:min{req, int, 最小值},max{req, int, 最大值},default{opt, int, 默认值}-->1
    </portNo>
    <TLS1_1Enable opt="true,false">
      <!--ro, opt, bool, TLS 1.1 版本使能, attr:opt{req, string, 取值范围}-->true
    </TLS1_1Enable>
    <TLS1_2Enable opt="true,false">
      <!--ro, opt, bool, TLS 1.2 版本使能, attr:opt{req, string, 取值范围}-->true
    </TLS1_2Enable>
    <TLS1_3Enable opt="true,false">
      <!--ro, opt, bool, TLS 1.3 版本使能, attr:opt{req, string, 取值范围}-->true
    </TLS1_3Enable>
    <TLS1_1SecurityPrompt opt="Only when the video output mode is Panorama, the device supports the VCA functions except Monitoring">
      <!--ro, opt, string, TLS1.1安全风险提示, attr:opt{req, string, 协议类型},
desc:prompt1中文提示语:仅在全景图像模式下支持非普通监控资源外的智能功能。
prompt1英文提示语:Only when the video output mode is Panorama, the device supports the VCA functions except Monitoring.*/-->Only when the video output mode
is Panorama, the device supports the VCA functions except Monitoring
    </TLS1_1SecurityPrompt>
    <connectTime min="1" max="999" def="10">
      <!--ro, opt, int, 登录用户连接超时时间, range:[1,999], step:1, unit:min, unitType:时间, attr:min{req, int, 最小值},max{req, int, 最大值},def{req, int, 默认值}, desc:缺省配置是10分钟-->10
    </connectTime>
  </AdminAccessProtocol>
</AdminAccessProtocolList>

```

69.8.2 获取设备所有协议安全参数

Request URL

GET /ISAPI/Security/adminAccesses

查询参数

无

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<AdminAccessProtocolList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, array, 设备接入协议列表, subType:object, attr:version{req, string, 协议版本}-->
  <AdminAccessProtocol>
    <!--ro, opt, object, 设备接入协议-->
    <id>
      <!--ro, req, string, 序号ID-->1
    </id>
    <enabled>
      <!--ro, opt, bool, 使能-->true
    </enabled>
    <protocol>
      <!--ro, req, enum, 协议类型, subType:string, [HTTP#HTTP,HTTPS#HTTPS,SDK#SDK私有协议等于
DEV_MANAGE,RTSP#RTSP,DEV_MANAGE#SDK,WebSocket#WebSocket,WebSocketS#WebSockets,SDK_OVER_TLS#SDK私有协议TLS链路加密,MATRIX_GATEWAY#矩阵接入网关协议,IOT#物联网
协议,GB35114#公安监控国标35114协议,104Proto#IEC104标准通讯协议,Bonjour#Bonjour,SRTP#RTSP码流传输加密,modbusTCP#modbusTCP],
desc:默认开启协议: HTTP,SDK,RTSP,DEV_MANAGE
默认关闭协议: HTTPS,WebSocket,WebSockets,
SDK_OVER_TLS,MATRIX_GATEWAY,IOT,GB35114,104Proto,SRTP-->HTTP
    </protocol>
    <redirectToHttps>
      <!--ro, opt, bool, 重定向, desc:在protocol为HTTPS时有效,是否开启https重定向.-->true
    </redirectToHttps>
    <streamOverTls>
      <!--ro, opt, bool, 码流是否加密, desc:当protocol选择为SDK_OVER_TLS时,码流是否启用TLS链路加密-->true
    </streamOverTls>
    <portNo>
      <!--ro, req, int, 端口号, range:[1,65535], desc:当前协议对应的端口号;协议类型通过protocol字段获取-->80
    </portNo>
    <TLS1_1Enable>
      <!--ro, opt, bool, TLS 1.1 版本使能, dep:and,{$.AdminAccessProtocolList.AdminAccessProtocol.protocol,eq,HTTPS},
{$.AdminAccessProtocolList.AdminAccessProtocol.protocol,eq,WebSocket},{$.AdminAccessProtocolList.AdminAccessProtocol.protocol,eq,WebSockets},
{$.AdminAccessProtocolList.AdminAccessProtocol.protocol,eq,SDK_OVER_TLS}-->true
    </TLS1_1Enable>
    <TLS1_2Enable>
      <!--ro, opt, bool, TLS 1.2 版本使能, dep:and,{$.AdminAccessProtocolList.AdminAccessProtocol.protocol,eq,HTTPS},
{$.AdminAccessProtocolList.AdminAccessProtocol.protocol,eq,WebSocket},{$.AdminAccessProtocolList.AdminAccessProtocol.protocol,eq,WebSockets},
{$.AdminAccessProtocolList.AdminAccessProtocol.protocol,eq,SDK_OVER_TLS}-->true
    </TLS1_2Enable>
    <TLS1_3Enable>
      <!--ro, opt, bool, TLS 1.3 版本使能, dep:and,{$.AdminAccessProtocolList.AdminAccessProtocol.protocol,eq,HTTPS},
{$.AdminAccessProtocolList.AdminAccessProtocol.protocol,eq,WebSocket},{$.AdminAccessProtocolList.AdminAccessProtocol.protocol,eq,WebSockets},
{$.AdminAccessProtocolList.AdminAccessProtocol.protocol,eq,SDK_OVER_TLS}-->true
    </TLS1_3Enable>
    <connectTime>
      <!--ro, opt, int, 登录用户连接超时时间, range:[1,999], step:1, unit:min, unitType:时间, desc:缺省配置是10分钟-->10
    </connectTime>
  </AdminAccessProtocol>
</AdminAccessProtocolList>
```

69.8.3 配置设备所有协议安全参数

Request URL

PUT /ISAPI/Security/adminAccesses

查询参数

无

请求报文

```
<?xml version="1.0" encoding="UTF-8"?>

<AdminAccessProtocolList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, array, 设备接入协议列表, subType:object, attr:version{req, string, 协议版本}-->
  <AdminAccessProtocol>
    <!--opt, object, 设备接入协议-->
    <id>
      <!--req, string, 序号ID-->1
    </id>
    <enabled>
      <!--opt, bool, 使能-->true
    </enabled>
    <protocol>
      <!--req, enum, 协议类型, subType:string, [HTTP#HTTP,HTTPS#HTTPS,SDK#SDK私有协议等于
DEV_MANAGE,RTSP#RTSP,DEV_MANAGE#SDK,WebSocket#WebSocket,WebSocketS#WebSocketS,SDK_OVER_TLS#SDK私有协议TLS链路加密,MATRIX_GATEWAY#矩阵接入网关协议,IOT#物联网
协议,GB35114#公安监控国标35114协议,104Proto#IEC104标准通讯协议,Bonjour#Bonjour,SRTP#RTSP码流传输加密,modbusTCP#modbusTCP],
desc:默认开启协议: HTTP,SDK,RTSP,DEV_MANAGE
默认关闭协议: HTTPS,WebSocket,WebSocketS,
SDK_OVER_TLS,MATRIX_GATEWAY,IOT,GB35114,104Proto,SRTP-->HTTP
    </protocol>
    <redirectToHttps>
      <!--opt, bool, 重定向, desc:在protocol为HTTPS时有效,是否开启https重定向.-->true
    </redirectToHttps>
    <streamOverTls>
      <!--opt, bool, 码流是否加密, desc:当protocol选择为SDK_OVER_TLS时,码流是否启用TLS链路加密-->true
    </streamOverTls>
    <portNo>
      <!--req, int, 端口号, range:[1,65535], desc:当前协议对应的端口号;协议类型通过protocol字段获取-->80
    </portNo>
    <TLS1_1Enable>
      <!--opt, bool, TLS 1.1 版本使能, dep:and,{$.AdminAccessProtocolList.AdminAccessProtocol.protocol,eq,HTTPS},
{$.AdminAccessProtocolList.AdminAccessProtocol.protocol,eq,WebSocket},{$.AdminAccessProtocolList.AdminAccessProtocol.protocol,eq,WebSocketS},
{$.AdminAccessProtocolList.AdminAccessProtocol.protocol,eq,SDK_OVER_TLS}-->true
    </TLS1_1Enable>
    <TLS1_2Enable>
      <!--opt, bool, TLS 1.2 版本使能, dep:and,{$.AdminAccessProtocolList.AdminAccessProtocol.protocol,eq,HTTPS},
{$.AdminAccessProtocolList.AdminAccessProtocol.protocol,eq,WebSocket},{$.AdminAccessProtocolList.AdminAccessProtocol.protocol,eq,WebSocketS},
{$.AdminAccessProtocolList.AdminAccessProtocol.protocol,eq,SDK_OVER_TLS}-->true
    </TLS1_2Enable>
    <TLS1_3Enable>
      <!--opt, bool, TLS 1.3 版本使能, dep:and,{$.AdminAccessProtocolList.AdminAccessProtocol.protocol,eq,HTTPS},
{$.AdminAccessProtocolList.AdminAccessProtocol.protocol,eq,WebSocket},{$.AdminAccessProtocolList.AdminAccessProtocol.protocol,eq,WebSocketS},
{$.AdminAccessProtocolList.AdminAccessProtocol.protocol,eq,SDK_OVER_TLS}-->true
    </TLS1_3Enable>
    <connectTime>
      <!--opt, int, 登录用户连接超时时间, range:[1,999], step:1, unit:min, unitType:时间, desc:缺省配置是10分钟-->10
    </connectTime>
  </AdminAccessProtocol>
</AdminAccessProtocolList>
```

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 响应消息, attr:version{ro, req, string, 版本}-->
  <requestURL>
    <!--ro, req, string, 请求的URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, 状态码, subType:int, [0#OK,1#OK,2#Device Busy,3#Device Error,4#Invalid Operation,5#Invalid XML Format,6#Invalid XML Content,7#Reboot
Required]-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, 状态信息, subType:string, [OK#成功,Device Busy#设备忙,Device Error#设备异常,Invalid Operation#无效的操作,Invalid XML Format#无效的XML
格式,Invalid XML Content#无效的XML内容,Reboot#设备重启]-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, 详细错误码英文描述, desc:详细错误码的英文描述-->OK
  </subStatusCode>
</ResponseStatus>
```

69.8.4 获取设备单个协议安全参数

Request URL

GET /ISAPI/Security/adminAccesses/<indexID>

查询参数

参数名称	参数类型	描述
indexID	string	协议类型序号（可以通过批量接口获取）

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<AdminAccessProtocol xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 设备接入协议, attr:version{req, string, 协议版本}-->
  <id>
    <!--ro, req, string, 序号ID-->1
  </id>
  <enabled>
    <!--ro, opt, bool, 使能-->true
  </enabled>
  <protocol>
    <!--ro, req, enum, 协议类型, subType:string, [HTTP#HTTP,HTTPS#HTTPS,SDK#SDK私有协议等于
DEV_MANAGE, RTSP#RTSP,DEV_MANAGE#SDK,WebSocket#WebSocket,WebSocketS#WebSockets, SDK_OVER_TLS#SDK私有协议TLS链路加密,MATRIX_GATEWAY#矩阵接入网关协议,IOT#物联网
协议,GB35114#公安监控国标35114协议,104Proto#IEC104标准通讯协议,Bonjour#Bonjour,SRTP#RTSP码流传输加密,modbusTCP#modbusTCP],
desc:默认开启协议: HTTP,SDK,RTSP,DEV_MANAGE
默认关闭协议: HTTPS,WebSocket,WebSockets,
SDK_OVER_TLS,MATRIX_GATEWAY,IOT,GB35114,104Proto,SRTP-->HTTP
  </protocol>
  <redirectToHttps>
    <!--ro, opt, bool, 重定向, desc: 在protocol为HTTPS时有效,是否开启https重定向. -->true
  </redirectToHttps>
  <streamOverTls>
    <!--ro, opt, bool, 码流是否加密, desc: 当protocol选择为SDK_OVER_TLS时,码流是否启用TLS链路加密-->true
  </streamOverTls>
  <portNo>
    <!--ro, req, int, 端口号, range:[1,65535], desc: 当前协议对应的端口号;协议类型通过protocol字段获取-->80
  </portNo>
  <TLS1_1Enable>
    <!--ro, opt, bool, TLS 1.1 版本使能, dep:and,{$.AdminAccessProtocolList.AdminAccessProtocol.protocol,eq,HTTPS},
{$.AdminAccessProtocolList.AdminAccessProtocol.protocol,eq,WebSocket},{$.AdminAccessProtocolList.AdminAccessProtocol.protocol,eq,WebSockets},
{$.AdminAccessProtocolList.AdminAccessProtocol.protocol,eq,SDK_OVER_TLS}-->true
  </TLS1_1Enable>
  <TLS1_2Enable>
    <!--ro, opt, bool, TLS 1.2 版本使能, dep:and,{$.AdminAccessProtocolList.AdminAccessProtocol.protocol,eq,HTTPS},
{$.AdminAccessProtocolList.AdminAccessProtocol.protocol,eq,WebSocket},{$.AdminAccessProtocolList.AdminAccessProtocol.protocol,eq,WebSockets},
{$.AdminAccessProtocolList.AdminAccessProtocol.protocol,eq,SDK_OVER_TLS}-->true
  </TLS1_2Enable>
  <TLS1_3Enable>
    <!--ro, opt, bool, TLS 1.3 版本使能, dep:and,{$.AdminAccessProtocolList.AdminAccessProtocol.protocol,eq,HTTPS},
{$.AdminAccessProtocolList.AdminAccessProtocol.protocol,eq,WebSocket},{$.AdminAccessProtocolList.AdminAccessProtocol.protocol,eq,WebSockets},
{$.AdminAccessProtocolList.AdminAccessProtocol.protocol,eq,SDK_OVER_TLS}-->true
  </TLS1_3Enable>
</AdminAccessProtocol>
```

69.8.5 配置设备单个协议安全参数

Request URL

PUT /ISAPI/Security/adminAccesses/<indexID>

查询参数

参数名称	参数类型	描述
indexID	string	协议类型序号（可以通过批量接口获取）

请求报文

```
<?xml version="1.0" encoding="UTF-8"?>

<AdminAccessProtocol xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, 设备接入协议, attr:version{req, string, 协议版本}-->
  <id>
    <!--req, string, 序号ID-->1
  </id>
  <enabled>
    <!--opt, bool, 使能-->true
  </enabled>
  <protocol>
    <!--req, enum, 协议类型, subType:string, [HTTP#HTTP,HTTPS#HTTPS,SDK#SDK私有协议等于
DEV_MANAGE,RTSP#RTSP,DEV_MANAGE#SDK,WebSocket#WebSocket,WebSocketS#WebSocketS,SDK_OVER_TLS#SDK私有协议TLS链路加密,MATRIX_GATEWAY#矩阵接入网关协议,IOT#物联网
协议,GB35114#公安监控国标35114协议,104Proto#IEC104标准通讯协议,Bonjour#Bonjour,SRTP#RTSP码流传输加密,modbusTCP#modbusTCP],
desc:默认开启协议: HTTP,SDK,RTSP,DEV_MANAGE
默认关闭协议: HTTPS,WebSocket,WebSocketS,
SDK_OVER_TLS,MATRIX_GATEWAY,IOT,GB35114,104Proto,SRTP-->HTTP
  </protocol>
  <redirectToHttps>
    <!--opt, bool, 重定向, desc:在protocol为HTTPS时有效,是否开启https重定向.-->true
  </redirectToHttps>
  <streamOverTls>
    <!--opt, bool, 码流是否加密, desc:当protocol选择为SDK_OVER_TLS时,码流是否启用TLS链路加密-->true
  </streamOverTls>
  <portNo>
    <!--req, int, 端口号, range:[1,65535], desc:当前协议对应的端口号;协议类型通过protocol字段获取-->80
  </portNo>
  <TLS1_1Enable>
    <!--opt, bool, TLS 1.1 版本使能, dep:and,{$.AdminAccessProtocolList.AdminAccessProtocol.protocol,eq,HTTPS},
{$.AdminAccessProtocolList.AdminAccessProtocol.protocol,eq,WebSocket},{$.AdminAccessProtocolList.AdminAccessProtocol.protocol,eq,WebSocketS},
{$.AdminAccessProtocolList.AdminAccessProtocol.protocol,eq,SDK_OVER_TLS}-->true
  </TLS1_1Enable>
  <TLS1_2Enable>
    <!--opt, bool, TLS 1.2 版本使能, dep:and,{$.AdminAccessProtocolList.AdminAccessProtocol.protocol,eq,HTTPS},
{$.AdminAccessProtocolList.AdminAccessProtocol.protocol,eq,WebSocket},{$.AdminAccessProtocolList.AdminAccessProtocol.protocol,eq,WebSocketS},
{$.AdminAccessProtocolList.AdminAccessProtocol.protocol,eq,SDK_OVER_TLS}-->true
  </TLS1_2Enable>
  <TLS1_3Enable>
    <!--opt, bool, TLS 1.3 版本使能, dep:and,{$.AdminAccessProtocolList.AdminAccessProtocol.protocol,eq,HTTPS},
{$.AdminAccessProtocolList.AdminAccessProtocol.protocol,eq,WebSocket},{$.AdminAccessProtocolList.AdminAccessProtocol.protocol,eq,WebSocketS},
{$.AdminAccessProtocolList.AdminAccessProtocol.protocol,eq,SDK_OVER_TLS}-->true
  </TLS1_3Enable>
</AdminAccessProtocol>
```

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 响应消息, attr:version{ro, req, string, 版本}-->
  <requestURL>
    <!--ro, req, string, 请求的URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, 状态码, subType:int, [0#OK,1#OK,2#Device Busy,3#Device Error,4#Invalid Operation,5#Invalid XML Format,6#Invalid XML Content,7#Reboot
Required]-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, 状态信息, subType:string, [OK#成功,Device Busy#设备忙,Device Error#设备异常,Invalid Operation#无效的操作,Invalid XML Format#无效的XML
格式,Invalid XML Content#无效的XML内容,Reboot#设备重启]-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, 详细错误码英文描述, desc:详细错误码的英文描述-->OK
  </subStatusCode>
</ResponseStatus>
```

69.9 在线升级管理(S)

69.9.1 获取在线升级状态

Request URL

GET /ISAPI/System/onlineUpgrade/status

查询参数

无

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<OnlineUpgradeStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, 在线升级状态, attr:version{req, string, 协议版本}-->
  <status>
    <!--ro, req, enum, 在线升级状态, subType:string, [notUpgrade#未升级,upgrading#升级中,successful#成功升级,languageMismatch#语言不匹配,writeFlashError#写Flash错误,packageTypeMismatch#升级包类型不匹配,packageVersionMismatch#升级包版本不匹配,netUnreachable#网络断开,unknownError#未知错误,engineVersionMismatch#升级包引擎版本不匹配]-->notUpgrade
  </status>
  <percent>
    <!--ro, req, int, 当前升级进度百分比, range:[0,100]-->1
  </percent>
  <onlineUpdateDevInfoList>
    <!--ro, opt, object, 在线升级设备信息列表, desc:指的是子设备、子模块或子通道的升级状态信息.-->
    <onlineUpdateDevInfo>
      <!--ro, opt, object, 在线升级设备信息-->
      <moduleType>
        <!--ro, req, enum, 模块类型, subType:string, [keypad#键盘,cardReader#读卡器,siren#警号,reLay#继电器,transmitter#Transmitter,remoteCtrl#遥控器,peater#中继器,detector#探测器]-->keypad
      </moduleType>
      <moduleID>
        <!--ro, req, int, 模块编号, range:[1,128], step:1-->1
      </moduleID>
      <model>
        <!--ro, opt, string, 型号, range:[1,32]-->test
      </model>
      <status>
        <!--ro, opt, enum, 升级状态, subType:string, [allowance#可升级（未触发升级命令）,ready#待升级（已触发升级命令,upgrading#升级中,success#升级成功,failed#升级失败]-->allowance
      </status>
      <percent>
        <!--ro, opt, int, 当前升级进度百分比, range:[0,100]-->1
      </percent>
    </onlineUpdateDevInfo>
  </onlineUpdateDevInfoList>
</OnlineUpgradeStatus>
```

69.9.2 配置开始下载升级包

Request URL

PUT /ISAPI/System/onlineUpgrade/downloadPackage?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "statusCode": 1,
  /*ro, opt, int, 状态码, desc:无法用1表示时（1表示成功且无特殊状态）必须返回*/
  "statusString": "OK",
  /*ro, opt, string, 状态描述, range:[1,64], desc:无法用OK表示时（OK表示成功且无特殊状态）必须返回*/
  "subStatusCode": "ok",
  /*ro, opt, string, 子状态码, range:[1,64], desc:无法用ok表示（ok表示成功且无特殊状态）必须返回*/
  "errorCode": 1,
  /*ro, opt, int, 错误码, desc:当statusCode不为1时，与subStatusCode对应*/
  "errorMsg": "ok"
  /*ro, opt, string, 错误信息, desc:当statusCode不为1时，必须返回,解释信息在协议约束中，允许设备在后续的版本迭代中，进行优化丰富提升（不限制死）*/
}
```

69.9.3 获取在线升级下载升级包进度

Request URL

GET /ISAPI/System/onlineUpgrade/downloadPackage/status?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "DownloadPackageStatus": {
    /*ro, req, object, 下载升级包状态*/
    "status": "notDownload",
    /*ro, req, enum, 下载状态, subType:string, [notDownload#未下载,downloading#下载中,pause#暂停,finish#完成,incorrectPackage#升级包格式不正确,hdOperationFailed#硬盘操作失败,downloadFailed#网络下载失败(下载过程中出现的错误都归于此)]*/
    "total": 0.0,
    /*ro, opt, float, 升级包总大小, desc:保留一位小数 单位MB*/
    "remain": 0.0,
    /*ro, opt, float, 剩余容量, desc:保留一位小数 单位MB*/
    "speed": 0.0,
    /*ro, opt, float, 下载速度, desc:保留一位小数 单位MB*/
    "remainTime": 0,
    /*ro, opt, int, 估计剩余时间, unit:s, unitType:时间*/
    "progress": 0
    /*ro, req, int, 进度, range:[0,100]*/
  }
}
```

69.9.4 获取新版本信息

Request URL

GET /ISAPI/System/onlineUpgrade/version?check= <check>

查询参数

参数名称	参数类型	描述
check	string	check为false（默认），设备不需要与服务器交互，只需要返回周期性检查的结果;check为true,设备需要与服务器交互并返回最新结果

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<OnlineUpgradeVersion xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, 新版本信息, attr:version{req, string, 协议版本}-->
  <newVersionAvailable>
    <!--ro, req, bool, 新版本是否已经生效-->true
  </newVersionAvailable>
  <newVersion>
    <!--ro, opt, string, 新升级包版本号信息-->test
  </newVersion>
  <changeLog>
    <!--ro, opt, string, 新升级包版本更新日志信-->test
  </changeLog>
</OnlineUpgradeVersion>
```

69.9.5 配置设备使用已有升级包升级

Request URL

PUT /ISAPI/System/onlineUpgrade/upgradeWithoutDownload?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "statusCode": 1,
  /*ro, opt, int, 状态码, desc:无法用1表示时（1表示成功且无特殊状态）必须返回*/
  "statusString": "OK",
  /*ro, opt, string, 状态描述, range:[1,64], desc:无法用OK表示时（OK表示成功且无特殊状态）必须返回*/
  "subStatusCode": "ok",
  /*ro, opt, string, 子状态码, range:[1,64], desc:无法用ok表示（ok表示成功且无特殊状态）必须返回*/
  "errorCode": 1,
  /*ro, opt, int, 错误码, desc:当statusCode不为1时，与subStatusCode对应*/
  "errorMsg": "ok"
  /*ro, opt, string, 错误信息, desc:当statusCode不为1时，必须返回,解释信息在协议约束中，允许设备在后续的版本迭代中，进行优化丰富提升（不限制死）*/
}
```

69.9.6 配置设备在线升级

Request URL

PUT /ISAPI/System/onlineUpgrade/upgrade?type= <type> &id= <indexID> &channelID= <channelID>

查询参数

参数名称	参数类型	描述
type	string	设备类型type: 1、报警设备: keypad-键盘, siren-警号, repeater-中继器, detector-探测器, transmitter-Transmitter外设, remoteCtrl-遥控器; relay-继电器, cardReader#读卡器
indexID	string	设备编号id: 从1开始例外: 防区和继电器id: 从0开始
channelID	string	通道号

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 响应消息, attr:version{ro, req, string, 版本}-->
  <requestURL>
    <!--ro, req, string, 请求的URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, 状态码, subType:int, [0#OK,1#OK,2#Device Busy,3#Device Error,4#Invalid Operation,5#Invalid XML Format,6#Invalid XML Content,7#Reboot Required]-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, 状态信息, subType:string, [OK#成功,Device Busy#设备忙,Device Error#设备异常,Invalid Operation#无效的操作,Invalid XML Format#无效的XML格式,Invalid XML Content#无效的XML内容,Reboot#设备重启]-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, 详细错误码英文描述, desc:详细错误码的英文描述-->OK
  </subStatusCode>
</ResponseStatus>
```

69.9.7 获取在线升级能力

Request URL

GET /ISAPI/System/onlineUpgrade/capabilities

查询参数

无

请求报文

无

响应报文

```

<?xml version="1.0" encoding="UTF-8"?>

<OnlineUpgradeCap xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, 在线升级, attr:version{req, string, 协议版本}-->
  <firmwareNum max="10">
    <!--ro, req, int, 可在线升级的包数量, attr:max{req, int, 可在线升级的包数量最大值}-->1
  </firmwareNum>
  <firmwareCode max="256">
    <!--ro, req, int, 识别码最大长度, attr:max{req, int, 识别码最大长度最大值}-->1
  </firmwareCode>
  <firmwareVersion max="64">
    <!--ro, req, int, 版本最大长度, attr:max{req, int, 版本最大长度最大值}-->1
  </firmwareVersion>
  <firmwareCodeNumOnce max="10">
    <!--ro, req, int, 单次可获取的识别码的最大个数, attr:max{req, int, 单次可获取的识别码的最大个数最大值}-->1
  </firmwareCodeNumOnce>
  <upgradePercent min="0" max="100">
    <!--ro, req, int, 升级进度百分比能力, attr:min{req, int, 升级进度百分比能力最小值},max{req, int, 升级进度百分比能力最大值}-->1
  </upgradePercent>
  <Version>
    <!--ro, req, object, 升级包版本信息-->
    <newVersion max="64">
      <!--ro, req, string, 新升级包版本号信息, attr:max{req, int, 新升级包版本号信息长度最大值}-->test
    </newVersion>
    <changeLog max="64">
      <!--ro, req, string, 新升级包版本日志信息, attr:max{req, int, 新升级包版本日志信息长度最大值}-->test
    </changeLog>
  </Version>
  <rebootAfterUpgrade>
    <!--ro, opt, enum, 设备重启后是否更新, subType:string, [auto#自动更新,manual#手动更新]-->auto
  </rebootAfterUpgrade>
  <DeviceParameter>
    <!--ro, opt, object, 在线升级参数-->
    <isSupportAutoDownloadPackage>
      <!--ro, opt, bool, 自动下载升级包是否支持-->true
    </isSupportAutoDownloadPackage>
    <notSupportAutoUpgrade>
      <!--ro, opt, bool, 自动下载升级包并升级是否不再支持, desc:对应url: PUT /ISAPI/System/onlineUpgrade/upgrade-->true
    </notSupportAutoUpgrade>
    <isSupportTimingUpgrade>
      <!--ro, opt, bool, 定时升级是否支持-->true
    </isSupportTimingUpgrade>
  </DeviceParameter>
  <ManualDownloadPackage>
    <!--ro, opt, object, 手动下载升级包-->
    <supportOperation opt="start,cancel,pause,resume">
      <!--ro, opt, enum, 支持的操作, subType:string, [start#开始,cancel#取消,pause#中断,resume#继续], attr:opt{req, string, 支持的操作取值范围}-->start
    </supportOperation>
  </ManualDownloadPackage>
  <isSupportIgnoreCurrentVersion>
    <!--ro, opt, bool, 是否支持忽略当前版本-->true
  </isSupportIgnoreCurrentVersion>
  <UpgradePackageTask>
    <!--ro, opt, object, 是否支持下发升级任务, desc:返回本节点表示支持/ISAPI/System/onlineUpgrade/task?format=json、/ISAPI/System/onlineUpgrade/status/<ID>->
  </UpgradePackageTask>
  <isSupportUpgradeModules>
    <!--ro, opt, bool, 设备是否支持升级子模块-->true
  </isSupportUpgradeModules>
  <isSupportUpgradeChannels>
    <!--ro, opt, bool, 设备是否支持升级子通道-->true
  </isSupportUpgradeChannels>
  <UpgradePackageTask>
  <isSupportPatrolInspection>
    <!--ro, opt, bool, 设备是否支持巡检-->true
  </isSupportPatrolInspection>
  <isSupportBatchUpgrade>
    <!--ro, opt, bool, 设备是否支持批量在线升级-->true
  </isSupportBatchUpgrade>
  <isSupportCancelUpgrade>
    <!--ro, opt, bool, 设备是否支持取消在线升级-->true
  </isSupportCancelUpgrade>
  <isSupportPackagePause>
    <!--ro, opt, bool, 设备是否支持暂停升级包下载, desc:/ISAPI/System/onlineUpgrade/PackagePause?format=json-->true
  </isSupportPackagePause>
  <isSupportPackageResume>
    <!--ro, opt, bool, 设备是否支持继续升级包下载, desc:/ISAPI/System/onlineUpgrade/PackageResume?format=json-->true
  </isSupportPackageResume>
</OnlineUpgradeCap>

```

69.10 蓝牙网络配置

69.10.1 获取蓝牙参数能力集

Request URL

GET /ISAPI/System/Bluetooth/capabilities

查询参数

无

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<BluetoothCap xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 能力, attr:version{req, string, 协议版本}-->
  <isSupportIbeacon>
    <!--ro, opt, bool, 是否支持Ibeacon蓝牙参数配置-->true
  </isSupportIbeacon>
  <isSupportDeviceCfg>
    <!--ro, opt, bool, 是否支持蓝牙参数配置-->true
  </isSupportDeviceCfg>
  <isSupportDeviceConnection>
    <!--ro, opt, bool, 是否支持控制蓝牙设备连接-->true
  </isSupportDeviceConnection>
  <isSupportDeviceSearchResult>
    <!--ro, opt, bool, 是否支持获取蓝牙设备搜索结果-->true
  </isSupportDeviceSearchResult>
  <isSupportAllowList>
    <!--ro, opt, bool, 是否支持蓝牙网络白名单管理, desc: 接口能力/ISAPI/System/Bluetooth/allowList/capabilities?format=json-->true
  </isSupportAllowList>
</BluetoothCap>
```

69.10.2 获取蓝牙设备参数能力

Request URL

GET /ISAPI/System/Bluetooth/deviceCfg/capabilities?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "enabled": {
    /*ro, req, object, 蓝牙启用使用取值范围*/
    "@opt": [true, false]
    /*ro, req, array, 枚举, subType:bool*/
  },
  "version": {
    /*ro, opt, object, 本机蓝牙模块版本字符串长度取值范围*/
    "@min": 0,
    /*ro, req, int, 最小长度, range:[0,32]*/
    "@max": 32
    /*ro, req, int, 最大长度, range:[0,32]*/
  },
  "name": {
    /*ro, opt, object, 本机蓝牙设备名称字符串长度取值范围（按设备支持最大返回）*/
    "@min": 0,
    /*ro, req, int, 最小长度, range:[0,32]*/
    "@max": 32
    /*ro, req, int, 最大长度, range:[0,32]*/
  },
  "status": {
    /*ro, opt, object, 本机蓝牙连接状态取值范围*/
    "@opt": ["connected", "notConnected"]
    /*ro, req, array, 枚举, subType:string*/
  },
  "forcedDisconnectEnabled": {
    /*ro, opt, object, 蓝牙强制断开使用取值范围*/
    "@opt": [true, false]
    /*ro, req, array, 枚举, subType:bool*/
  },
  "pairingPassword": {
    /*ro, opt, object, 蓝牙配对密码字符串长度取值范围*/
    "@min": 1,
    /*ro, req, int, 最小长度, range:[1,32]*/
    "@max": 32
    /*ro, req, int, 最大长度, range:[1,32]*/
  },
  "transmitPower": {
    /*ro, opt, object, 发射功率*/
    "@min": 1,
    /*ro, req, int, 最小值, range:[1,12]*/
    "@max": 12
    /*ro, req, int, 最大值, range:[1,12]*/
  },
  "readerID": {
    /*ro, opt, object, 读卡器编号, desc:表示获取及配置协议URL中支持readerID参数*/
    "@min": 1,
    /*ro, req, int, 最小值*/
    "@max": 8
    /*ro, req, int, 最大值*/
  },
  "requireReaderID": true,
  /*ro, opt, bool, 请求URL中是否必须包含readerID,
  desc:字段不存在默认为false.
  对于智能主机仅支持URL按readerID配置*/
  "connectTimeOut": {
    /*ro, opt, object, 连接超时时间取值范围, desc:表示蓝牙连接的超时时间,单位s*/
    "@min": 0,
    /*ro, opt, int, 最小值*/
    "@max": 60
    /*ro, opt, int, 最大值*/
  },
  "autoDisconnectEnabled": {
    /*ro, opt, object, 连接超时自动断开使用取值范围, desc:true:超时自动断开,false:超时不自动断开*/
    "@opt": [true, false]
    /*ro, opt, array, 连接超时自动断开使用取值枚举, subType:bool*/
  }
}
```

69.10.3 获取蓝牙设备参数

Request URL

GET /ISAPI/System/Bluetooth/deviceCfg?format=json&readerID=<readerID>

查询参数

参数名称	参数类型	描述
readerID	string	智能主机存在多路读卡器，需要支持按ID配置

请求报文

无

响应报文

```
{
  "enabled": true,
  /*ro, req, bool, 蓝牙启用使能*/
  "version": "test",
  /*ro, opt, string, 蓝牙模块版本, range:[1,32]*/
  "name": "test",
  /*ro, opt, string, 蓝牙名称, range:[0,30]*/
  "status": "connected",
  /*ro, opt, enum, 蓝牙连接状态, subType:string, [connected#已连接,notConnected#未连接]*/
  "connectedDeviceName": ["", ""],
  /*ro, opt, array, 已连接的蓝牙设备名称, subType:string*/
  "transmitPower": 1,
  /*ro, opt, int, 发射功率等级, range:[1,12]*/
  "connectTimeout": 1,
  /*ro, opt, int, 连接超时时间, range:[0,60], desc:表示蓝牙连接的超时时间,单位s*/
  "autoDisconnectEnabled": true
  /*ro, opt, bool, 连接超时自动断开使能, desc:true:超时自动断开,false:超时不自动断开*/
}
```

69.11 接入EZVIZ云协议

69.11.1 获取萤石二维码

Request URL

GET /ISAPI/System/Network/EZVIZ/QRCode?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "EZVIZQRCode": {
    /*ro, opt, object, 萤石二维码*/
    "QRCode": "test"
    /*ro, req, string, 萤石二维码字符串*/
  }
}
```

69.11.2 获取萤石接入服务参数

Request URL

GET /ISAPI/System/Network/EZVIZ?security=<security> &iv=<iv>

查询参数

参数名称	参数类型	描述
security	string	取值范围[1,2],security=1,表示报文中的verificationCode和operateCode字段AES128加密security=2,表示报文中的verificationCode和operateCode字段AES256加密
iv	string	iv表示初始化向量, 在security=1或者2时为必须参数

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<EZVIZ xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 萤石服务参数, attr:version{opt, string, 协议版本}-->
  <enabled>
    <!--ro, req, bool, 使能, desc:默认关闭-->false
  </enabled>
  <registerStatus>
    <!--ro, opt, bool, 注册状态-->true
  </registerStatus>
```

```

</registerStatus>
<redirect>
  <!--ro, opt, bool, 是否允许设备重定向服务器地址, desc:是否允许设备重定向服务器地址,注意:false-表示允许, true-表示不允许-->true
</redirect>
<serverAddress>
  <!--ro, opt, object, 服务器地址-->
  <addressingFormatType>
    <!--ro, req, enum, ipAddress, subType:string, [ipAddress#IP地址,hostname#域名]-->ipAddress
  </addressingFormatType>
  <hostName>
    <!--ro, opt, string, 主机域名-->test
  </hostName>
  <ipAddress>
    <!--ro, opt, string, IPV4地址-->172.6.64.7
  </ipAddress>
  <ipv6Address>
    <!--ro, opt, string, IPV6地址-->1080:0:0:0:8:800:200C:417A
  </ipv6Address>
</serverAddress>
<verificationCode>
  <!--ro, opt, string, 操作码,
desc:字段作用: 设备添加到萤石账户的密码, 也是码流&录像加密密码。
设备赋值:
1、如果设备中的接入萤石配置默认关闭, 设备是没有操作码的, 需要第一次开启萤石的时候配置。
2、如果设备中的接入萤石配置默认开启, 操作码和ISAP码(验证码)保持一致。-->123456789TESTtest
</verificationCode>
<offlineStatus>
  <!--ro, opt, enum, 设备离线具体状态, subType:string, [secretKeyInvalid#萤石注册验证码无效]-->secretKeyInvalid
</offlineStatus>
<enabledTiming>
  <!--ro, opt, bool, 启用萤石平台校时, desc:本字段需和设备时间配置中的萤石校时字段 (/ISAPI/System/time中的platformType填为EZVIZ) 赋值保持一致-->true
</enabledTiming>
<version>
  <!--ro, opt, string, 版本信息, desc:格式Vx.x.xbuildYYMMDD-->test
</version>
<operateCode>
  <!--ro, opt, string, 操作码字段弃用, desc:操作码字段弃用, 统一使用verificationCode字段-->test
</operateCode>
<network>
  <!--ro, opt, enum, 网络类型, subType:int, [0#无意义,1#自动,2#有线网络优先,3#有线网络(默认),4#3G/4G/GPRS]-->3
</network>
<convergenceCloudEnabled>
  <!--ro, opt, bool, 设备互联互通(萤石)功能使能-->>false
</convergenceCloudEnabled>
<streamEncryptEnabled>
  <!--ro, opt, bool, 萤石云码流加密开关, desc:当值为true, 或streamEncryptEnabled字段不存在时表示打开, 当值为false时表示关闭-->true
</streamEncryptEnabled>
<bindStatus>
  <!--ro, opt, enum, 萤石云账号状态, subType:string, [bind#已绑定,unbind#已解绑,unknown#未知], desc:unknown为设备还未联网并取到最新状态, 能力中有unknown表示设备支持离线方式添加, 并且设备在初始化状态未联网之前都保持unknown状态-->bind
</bindStatus>
<upISAPIStatus>
  <!--ro, opt, enum, 设备当前的支持萤石上行透传ISAPI的状态, subType:string, [open#打开,close#关闭], desc:设备当前的支持萤石上行透传ISAPI的状态, "open,close",指的是设备与萤石之间的是否打开上传链路,该能力是动态的,open-链路打开,close-链路关闭,GET /ISAPI/System/Network/EZVIZ/capabilities 中 isSupportUpISAPI 返回true表示支持该状态获取-->open
</upISAPIStatus>
<offlineCode>
  <!--ro, opt, string, 注册离线时返回的设备状态码;该节点不支持PUT方法;(该状态码由设备侧维护用于控件提示;并提供给设备侧定位问题所用), desc:当 registerStatus=false时才会返回。-->0xFFFFFFFF
</offlineCode>
<periodicTestEnabled>
  <!--ro, opt, bool, 周期检测使能, desc:使能后, 可以设置时间间隔, 报警主机会每隔一定时间给萤石平台发一个Test事件, 用于测试双方交互链路是否断开, 类似于心跳检测 (videoloss)。-->true
</periodicTestEnabled>
<periodicTestTime>
  <!--ro, opt, int, 周期检测时间, range:[10,86400], desc:单位 秒-->10
</periodicTestTime>
<networkPriorityEnabled>
  <!--ro, opt, bool, 网络优先级使能-->true
</networkPriorityEnabled>
<NetworkPriorityList>
  <!--ro, opt, object, 网络优先级列表, dep:and,{$.EZVIZ.networkPriorityEnabled,eq,true}, desc:不同网络类型注册到萤石的优先级-->
  <NetworkPriority>
    <!--ro, opt, object, 网络优先级-->
    <networkType>
      <!--ro, opt, enum, 网络类型, subType:int, [1#有线网络,2#有线网络2,3#无线网络,4#4G网络,5#5G网络]-->1
    </networkType>
    <priority>
      <!--ro, opt, int, 优先级, range:[1,10], desc:值越小, 优先级越高-->1
    </priority>
  </NetworkPriority>
</NetworkPriorityList>
<protocolVersion>
  <!--ro, opt, enum, 协议版本, subType:string, [v2.0#萤石v2.0,v3.0#萤石v3.0]-->v2.0
</protocolVersion>
</EZVIZ>

```

69.11.3 配置萤石接入服务参数

Request URL

PUT /ISAPI/System/Network/EZVIZ?security=<security>&iv=<iv>

查询参数

参数名称	参数类型	描述
security	string	取值范围[1,2]没有security表示数据不加密security=1,表示报文中的verificationCode和operateCode字段AES128加密security=2,表示报文中的verificationCode和operateCode字段AES256加密
iv	string	iv表示初始化向量，在security=1或者2时为必须参数

请求报文

```
<?xml version="1.0" encoding="UTF-8"?>

<EZVIZ xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, 萤石服务参数, attr:version{opt, string, 协议版本}-->
  <enabled>
    <!--req, bool, 使能, desc:默认关闭-->false
  </enabled>
  <registerStatus>
    <!--ro, opt, bool, 注册状态-->true
  </registerStatus>
  <redirect>
    <!--opt, bool, 是否允许设备重定向服务器地址, desc:是否允许设备重定向服务器地址,注意:false=表示允许, true=表示不允许-->true
  </redirect>
  <serverAddress>
    <!--opt, object, 服务器地址-->
    <addressingFormatType>
      <!--req, enum, ipAddress, subType:string, [ipaddress#IP地址,hostname#域名]-->ipaddress
    </addressingFormatType>
    <hostName>
      <!--opt, string, 主机域名-->test
    </hostName>
    <ipAddress>
      <!--opt, string, IPV4地址-->172.6.64.7
    </ipAddress>
    <ipv6Address>
      <!--opt, string, IPV6地址-->1080:0:0:0:8:800:200C:417A
    </ipv6Address>
    </serverAddress>
  <verificationCode>
    <!--ro, opt, string, 操作码,
    desc:字段作用: 设备添加到萤石账户的密码,也是码流&录像加密密码。
    设备赋值:
    1、如果设备中的接入萤石配置默认关闭,设备是没有操作码的,需要第一次开启萤石的时候配置。
    2、如果设备中的接入萤石配置默认开启,操作码和SAP码(验证码)保持一致。-->123456789TESTtest
  </verificationCode>
  <offlineStatus>
    <!--ro, opt, enum, 设备离线具体状态, subType:string, [secretKeyInvalid#萤石注册验证码无效]-->secretKeyInvalid
  </offlineStatus>
  <enabledTiming>
    <!--opt, bool, 启用萤石平台校时, desc:本字段需和设备时间配置中的萤石校时字段 (/ISAPI/System/time中的platformType填为EZVIZ) 赋值保持一致-->true
  </enabledTiming>
  <version>
    <!--opt, string, 版本信息, desc:格式Vx.x.xbuildYYMMDD-->test
  </version>
  <operateCode>
    <!--opt, string, 操作码字段弃用, desc:操作码字段弃用,统一使用verificationCode字段-->test
  </operateCode>
  <network>
    <!--opt, enum, 网络类型, subType:int, [0#无意义,1#自动,2#有线网络优先,3#有线网络(默认),4#3G/4G/GPRS]-->3
  </network>
  <convergenceCloudEnabled>
    <!--opt, bool, 设备互联互通(萤石)功能使能-->false
  </convergenceCloudEnabled>
  <streamEncryptEnabled>
    <!--opt, bool, 萤石云码流加密开关, desc:当值为true,或streamEncryptEnabled字段不存在时表示打开,当值为false时表示关闭-->true
  </streamEncryptEnabled>
  <upISAPIStatus>
    <!--ro, opt, enum, 设备当前的支持萤石上行透传ISAPI的状态, subType:string, [open#打开,close#关闭], desc:设备当前的支持萤石上行透传ISAPI的状态, "open,close",指的是设备与萤石之间的是否打开上传链路,该能力是动态的,open-链路打开,close-链路关闭,GET /ISAPI/System/Network/EZVIZ/capabilities 中 isSupportUpISAPI 返回true表示支持该状态获取-->open
  </upISAPIStatus>
  <bindStatus>
    <!--ro, opt, enum, 设备当前设备的萤石绑定状态, subType:string, [bind#已绑定,unbind#已解绑], desc:当返回bind状态时,可通过/ISAPI/System/Network/EZVIZ/unbind接口进行解绑-->bind
  </bindStatus>
  <periodicTestEnabled>
    <!--opt, bool, 周期检测使能, desc:使能后,可以设置时间间隔,报警主机会每隔一定时间给萤石平台发一个Test事件,用于测试双方交互链路是否断开,类似于心跳检测(video loss)。-->true
  </periodicTestEnabled>
  <periodicTestTime>
    <!--opt, int, 周期检测时间, range:[10,86400], desc:单位 秒-->10
  </periodicTestTime>
  <networkPriorityEnabled>
```

```
<!--opt, 0001, 网络优先级使能-->true
</networkPriorityEnabled>
<NetworkPriorityList>
<!--opt, object, 网络优先级列表, dep:and,{$.EZVIZ.networkPriorityEnabled,eq,true}, desc:不同网络类型注册到萤石的优先级-->
<NetworkPriority>
<!--opt, object, 网络优先级-->
<networkType>
<!--opt, enum, 网络类型, subType:int, [1#有线网络,2#有线网络2,3#无线网络,4#4G网络,5#5G网络]-->1
</networkType>
<priority>
<!--opt, int, 优先级, range:[1,10], desc:值越小, 优先级越高-->1
</priority>
</NetworkPriority>
</NetworkPriorityList>
<protocolVersion>
<!--opt, enum, 协议版本, subType:string, [v2.0#萤石v2.0,v3.0#萤石v3.0]-->v2.0
</protocolVersion>
<streamEncryptionPassword>
<!--opt, string, 码流加密密码, range:[8,16], desc:只能修改, 获取时不返回此字段-->test
</streamEncryptionPassword>
</EZVIZ>
```

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
<!--ro, req, object, 响应消息, attr:version{ro, req, string, 版本}-->
<requestURL>
<!--ro, req, string, 请求的URL-->null
</requestURL>
<statusCode>
<!--ro, req, enum, 状态码, subType:int, [0#OK,1#OK,2#Device Busy,3#Device Error,4#Invalid Operation,5#Invalid XML Format,6#Invalid XML Content,7#Reboot Required]-->0
</statusCode>
<statusString>
<!--ro, req, enum, 状态信息, subType:string, [OK#成功,Device Busy#设备忙,Device Error#设备异常,Invalid Operation#无效的操作,Invalid XML Format#无效的XML格式,Invalid XML Content#无效的XML内容,Reboot#设备重启]-->OK
</statusString>
<subStatusCode>
<!--ro, req, string, 详细错误码英文描述, desc:详细错误码的英文描述-->OK
</subStatusCode>
</ResponseStatus>
```

69.11.4 获取萤石接入服务参数能力

Request URL

GET /ISAPI/System/Network/EZVIZ/capabilities

查询参数

无

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<EZVIZ xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
<!--ro, req, object, 萤石服务能力集, attr:version{opt, string, 协议版本}-->
<enabled opt="true,false">
<!--ro, req, bool, 使能, attr:opt{opt, string, 取值范围}, desc:默认关闭-->>false
</enabled>
<registerStatus opt="true,false">
<!--ro, opt, bool, 注册状态, attr:opt{opt, string, 取值范围}-->true
</registerStatus>
<redirect opt="true,false">
<!--ro, opt, bool, 是否允许设备重定向服务器地址, attr:opt{opt, string, 取值范围}, desc:是否允许设备重定向服务器地址,注意:false-表示允许, true-表示不允许-->true
</redirect>
<serverAddress>
<!--ro, opt, object, 服务器地址-->
<addressingFormatType opt="ipaddress,hostname">
<!--ro, req, enum, 地址类型, subType:string, [ipaddress#IP地址,hostname#域名], attr:opt{req, string, 取值范围}, desc:ipaddress-IP地址,hostname-域名-->ipaddress
</addressingFormatType>
<hostname min="1" max="128">
<!--ro, opt, string, 域名, attr:min{opt, string, 最小值},max{opt, string, 最大值}-->www.baidu.com
</hostname>
<ipAddress min="null" max="null">
<!--ro, opt, string, IPV4地址, attr:min{opt, string, 最小值},max{opt, string, 最大值}-->172.6.64.7
</ipAddress>
```

```

    <ipv6Address min="null" max="null">
      <!--ro, opt, string, IPV6地址, attr:min{opt, string, 最小值},max{opt, string, 最大值}-->1080:0:0:0:8:800:200C:417A
    </ipv6Address>
  </serverAddress>
  <verificationCode min="null" max="null">
    <!--ro, opt, string, 验证码, attr:min{opt, string, 最小值},max{opt, string, 最大值}-->test
  </verificationCode>
  <offlineStatus opt="secretKeyInvalid,networkError,notInAllowList,unknown">
    <!--ro, opt, string, 设备离线具体状态, attr:opt{opt, string, 取值范围}, desc:secretKeyInvalid-萤石注册验证码无效,networkError-网络异常,notInAllowList-未授权,unknown-未知-->test
  </offlineStatus>
  <network opt="1,2,3,4">
    <!--ro, opt, enum, 网络类型, subType:int, [0#无意义,1#自动,2#有线网络优先,3#有线网络（默认）,4#3G/4G/GPRS], attr:opt{req, string, 取值范围}-->3
  </network>
  <streamEncryptEnabled opt="true,false">
    <!--ro, opt, bool, 萤石云码流加密开关, attr:opt{opt, string, 取值范围}, desc:或streamEncryptEnabled字段不存在时表示打开, 当值为false时表示关闭-->true
  </streamEncryptEnabled>
  <bindStatus opt="bind,unbind,unknown">
    <!--ro, opt, enum, 萤石云账号状态, subType:string, [bind#已绑定,unbind#已解绑,unknown#未知], attr:opt{req, string, 取值范围}, desc:获取当前设备的萤石绑定状态,bind为已绑定,unbind为已解绑。unknown为设备还未联网获取到最新状态,能力中有unknown表示设备支持离线方式添加,并且设备在初始化状态未联网之前都保持unknown状态-->bind
  </bindStatus>
  <isSupportEZVIZQRCode>
    <!--ro, opt, bool, 是否支持获取萤石二维码, desc:是否支持获取萤石二维码（GET /ISAPI/System/Network/EZVIZ/QRCode?format=json）-->true
  </isSupportEZVIZQRCode>
  <isSupportUpISAPI>
    <!--ro, opt, bool, 设备是否支持萤石上行透传ISAPI报文-->true
  </isSupportUpISAPI>
  <isSupportDownISAPI>
    <!--ro, opt, bool, 设备是否支持萤石下行透传ISAPI报文-->true
  </isSupportDownISAPI>
  <convergenceCloudEnabled opt="true,false">
    <!--ro, opt, bool, 设备互联互通（萤石）功能使能, attr:opt{opt, string, 可选项}-->false
  </convergenceCloudEnabled>
  <offlineCode min="0" max="16">
    <!--ro, opt, string, 注册离线时返回的设备状态码;该节点不支持PUT方法;(该状态码由设备侧维护用于控件提示;并提供给设备侧定位问题所用), attr:min{opt, string, 最小值},max{opt, string, 最大值}, desc:当registerStatus=false时才会返回.-->0xFFFFFFFF
  </offlineCode>
  <isSupportEZVIZCloud>
    <!--ro, opt, bool, 是否支持萤石云存储配置, desc:对应配置接口:ISAPI/System/Network/EZVIZ/cloudStorage?format=json-->true
  </isSupportEZVIZCloud>
  <periodicTestEnabled opt="true,false">
    <!--ro, opt, bool, 周期检测使能, attr:opt{req, string, 取值范围}, desc:使能后, 可以设置时间间隔, 报警主机每隔一定时间给萤石平台发一个Test事件, 用于测试双方交互链路是否断开, 类似于心跳检测（videoLoss）。-->true
  </periodicTestEnabled>
  <periodicTestTime min="10" max="86400">
    <!--ro, opt, int, 周期检测时间, attr:min{req, int, 最小值},max{req, int, 最大值}, desc:单位 秒-->5
  </periodicTestTime>
  <CloudStorage>
    <!--ro, opt, object, 云存储相关配置, desc:当isSupportEZVIZCloud为true时-->
    <isSupportRecordsSubscribe>
      <!--ro, opt, bool, 是否支持萤石云云存储事件录像订阅参数, desc:/ISAPI/System/Network/EZVIZ/cloudStorage/recordsSubscribe/capabilities?format=json-->true
    </isSupportRecordsSubscribe>
    <isSupportChannelRecordsSubscribe>
      <!--ro, opt, bool, 是否支持单通道配置萤石云云存储事件录像订阅参数, desc:/ISAPI/System/Network/EZVIZ/cloudStorage/recordsSubscribe/channelIds/channelID/capabilities?format=json-->true
    </isSupportChannelRecordsSubscribe>
    <isSupportRecordsSubscribeStatistics>
      <!--ro, opt, bool, 是否支持实时统计, desc:该功能依赖与云存储事件录像订阅功能,对应URL:/ISAPI/System/Network/EZVIZ/cloudStorage/recordsSubscribe/statistics/capabilities?format=json-->true
    </isSupportRecordsSubscribeStatistics>
  </CloudStorage>
  <physicalAccessVerification opt="1,2,3,4">
    <!--ro, opt, enum, 设备支持的物理接触式添加方式, subType:string, [1#AP配网传递,2#用户令牌（用户token）绑定,3#物理按键接触,4#扫码绑定（设备token）], attr:opt{req, string, 取值范围}, desc:物理接触式添加—用户告诉平台, 对自己所添加的设备拥有控制权（能触碰到这个设备）AP配网传递: 设备和萤石APP通过设备AP热点建立局域网, 萤石APP从萤石平台获取用户token, 下发给设备; 用户令牌（用户token）绑定: 设备通过有线或者无线连接路由器, 客户端（APP是客户端的一种）与设备在一个局域网内, 客户端从萤石平台获取用户token, 下发给设备, 与方案1的区别就是组网差异; 物理按键接触: 真实物理接触, 比如在规定时间内要求用户按设备的reset按键, 设备上报一个特定事件, 让平台知道用户在规定的时间内触碰了这个设备; 扫码绑定（设备token）: 设备在联网且未绑定时, 从萤石平台获取有时效的设备token, 结合设备信息生成二维码, 可以扫码添加到萤石账户, 此方法可实现远程添加.-->1
  </physicalAccessVerification>
  <userToken min="0" max="16">
    <!--ro, opt, string, 设备token, attr:min{req, int, 最小值},max{req, int, 最大值}, desc:对应url: GET /ISAPI/System/Network/EZVIZ/physicalAccessVerification?format=json&security=<security>&iv=<iv>-->test
  </userToken>
  <isSupportUserTokenUpdate opt="true,false">
    <!--ro, opt, bool, 是否支持强制刷新设备从萤石云获取的token, attr:opt{req, string, 取值范围}, desc:设备从萤石云获取的Token是有保活周期的, 如果失效了, 可通过刷新接口更新PUT /ISAPI/System/Network/EZVIZ/physicalAccessVerification/userTokenUpdate?format=json-->true
  </isSupportUserTokenUpdate>
  <networkPriorityEnabled opt="true,false">
    <!--ro, opt, bool, 网络优先级使能, attr:opt{req, string, 取值范围}-->true
  </networkPriorityEnabled>
  <networkPriorityList size="5">
    <!--ro, opt, object, 网络优先级列表, attr:size{req, int, 数组元素总数}, desc:不同网络类型注册到萤石的优先级-->
  </networkPriorityList>
  <networkType opt="1,2,3,4,5">
    <!--ro, opt, enum, 网络类型, subType:int, [1#有线网络,2#有线网络2,3#无线网络,4#4G网络,5#5G网络], attr:opt{req, string, 取值范围}-->1
  </networkType>
  <priority min="1" max="10">
    <!--ro, opt, int, 优先级, range:[1,10], attr:min{req, int, 最小值},max{req, int, 最大值}, desc:值越小, 优先级越高-->1
  </priority>

```

```
</priority>
</NetworkPriority>
</NetworkPriorityList>
<protocolVersion opt="v2.0,v3.0" def="v2.0">
  <!--ro, opt, enum, 协议版本, subtype:string, [v2.0#萤石v2.0,v3.0#萤石v3.0], attr:opt{req, string, 取值范围},def{req, string, 默认值}-->v2.0
</protocolVersion>
<physicalAccessUserToken min="0" max="16">
  <!--ro, opt, string, 用户token, attr:min{req, int, 最小值},max{req, int, 最大值}, desc:对应url GET /ISAPI/System/Network/EZVIZ/physicalAccessUserToken?
format=json和PUT /ISAPI/System/Network/EZVIZ/physicalAccessUserToken?format=json-->test
</physicalAccessUserToken>
<streamEncryptionPassword min="8" max="16">
  <!--ro, opt, string, 码流加密密码, range:[8,16], attr:min{req, int, 最小值},max{req, int, 最大值}, desc:只能修改, 获取时不返回此字段, 需要敏感字段加密,
对应PUT /ISAPI/System/Network/EZVIZ?security=<security>&iv=<iv>中的streamEncryptionPassword参数-->test
</streamEncryptionPassword>
<isSupportAccessUserTokenCfgList>
  <!--ro, opt, bool, 是否支持萤石用户token批量配置, desc:/ISAPI/System/Network/EZVIZ/AccessUserTokenCfgList/capabilities?format=json-->true
</isSupportAccessUserTokenCfgList>
</EZVIZ>
```

69.12 WIFI网络配置

69.12.1 获取单个网络列表

Request URL

GET /ISAPI/System/Network/interfaces/<interfaceID>/wireless/accessPointList/<pointsID>

查询参数

参数名称	参数类型	描述
interfaceID	string	
pointsID	string	

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<accessPoint xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, 接入点, attr:version{req, string, 协议版本}-->
  <id>
    <!--ro, req, int, 索引-->0
  </id>
  <networkMode>
    <!--ro, opt, enum, 网络模式, subType:string, [infrastructure#基础设施,adhoc#临时安排]-->infrastructure
  </networkMode>
  <channel>
    <!--ro, opt, string, 通道号-->test
  </channel>
  <ssid>
    <!--ro, req, string, SSID-->test
  </ssid>
  <speed>
    <!--ro, opt, int, 速率, unit:Mbps, unitType:带宽速度-->0
  </speed>
  <signalStrength>
    <!--ro, opt, int, 信号强度-->0
  </signalStrength>
  <securityMode>
    <!--ro, req, enum, 安全模式, subType:string, [disable#未启用,WEP#WEP,WPA-personal#WPA-personal,WPA2-personal#WPA2-personal,WPA-RADIUS#WPA-RADIUS,WPA-enterprise#WPA-enterprise,WPA2-enterprise#WPA2-enterprise,WPA#WPA,WPA2-PSK#WPA2-PSK]-->disable
  </securityMode>
  <connected>
    <!--ro, opt, bool, 是否连接-->true
  </connected>
  <connecting>
    <!--ro, opt, bool, wifi是否处于连接中, desc:上面的connected应该是表示wifi已连接还是没有连接, 这个字段表示是否在连接中-->true
  </connecting>
  <ipAddress>
    <!--ro, opt, string, ipv4地址-->test
  </ipAddress>
  <ipv6Address>
    <!--ro, opt, string, ipv6地址-->test
  </ipv6Address>
  <subnetMask>
    <!--ro, opt, string, 子网掩码, desc:针对IPv4-->test
  </subnetMask>
  <defaultGateway>
    <!--ro, opt, string, 默认网关-->test
  </defaultGateway>
</accessPoint>
```

69.12.2 获取全部网络列表

Request URL

GET /ISAPI/System/Network/interfaces/<interfaceID>/wireless/accessPointList

查询参数

参数名称	参数类型	描述
interfaceID	string	

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<accessPointList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, array, array, subType:object, attr:version{req, string, 协议版本}-->
  <accessPoint>
    <!--ro, opt, object, 接入点-->
    <id>
      <!--ro, req, int, 索引-->0
    </id>
    <networkMode>
      <!--ro, opt, enum, 网络模式, subType:string, [infrastructure#基础设施,adhoc#临时安排]-->infrastructure
    </networkMode>
    <channel>
      <!--ro, opt, string, 通道号-->test
    </channel>
    <ssid>
      <!--ro, req, string, SSID-->test
    </ssid>
    <speed>
      <!--ro, opt, int, 速率, unit:Mbps, unitType:带宽速度-->0
    </speed>
    <signalStrength>
      <!--ro, opt, int, 信号强度-->0
    </signalStrength>
    <securityMode>
      <!--ro, req, enum, 安全模式, subType:string, [disable#未启用,WEP#WEP,WPA-personal#WPA-personal,WPA2-personal#WPA2-personal,WPA-RADIUS#WPA-RADIUS,WPA-enterprise#WPA-enterprise,WPA2-enterprise#WPA2-enterprise,WPA#WPA,WPA2-PSK#WPA2-PSK]-->disable
    </securityMode>
    <connected>
      <!--ro, opt, bool, 是否连接-->true
    </connected>
    <connecting>
      <!--ro, opt, bool, wifi是否处于连接中, desc:上面的connected应该是表示wifi已连接还是没有连接, 这个字段表示是否在连接中-->true
    </connecting>
    <ipAddress>
      <!--ro, opt, string, ipv4地址-->test
    </ipAddress>
    <ipv6Address>
      <!--ro, opt, string, ipv6地址-->test
    </ipv6Address>
    <subnetMask>
      <!--ro, opt, string, 子网掩码, desc:针对IPV4-->test
    </subnetMask>
    <defaultGateway>
      <!--ro, opt, string, 默认网关-->test
    </defaultGateway>
  </accessPoint>
</accessPointList>
```

69.12.3 获取单网口WIFI连接状态

Request URL

GET /ISAPI/System/Network/interfaces/<interfaceID>/wireless/connectStatus

查询参数

参数名称	参数类型	描述
interfaceID	string	

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<WirelessConnectStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, 无线连接状态, attr:version{req, string, 协议版本}-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <status>
    <!--ro, opt, enum, wifi连接状态, subType:string, [fail#连接失败,succes#连接成功,connecting#连接中,nomodule#无模块]-->fail
  </status>
  <ssid>
    <!--ro, opt, string, ssid信息-->test
  </ssid>
  <signalStrength>
    <!--ro, opt, int, 信号强度, range:[0,100]-->1
  </signalStrength>
  <ipAddress>
    <!--ro, opt, string, ip地址-->test
  </ipAddress>
</WirelessConnectStatus>
```

69.13 ISUP协议接入

69.13.1 配置ISUP(EHome)单中心服务参数

Request URL

PUT /ISAPI/System/Network/Ehome?security= <security> &iv= <iv> ¢erID= <centerID>

查询参数

参数名称	参数类型	描述
security	string	没有security表示数据不加密； security=1,表示报文中的key字段AES128加密； security=2,表示报文中的key字段AES256加密；
iv	string	iv表示初始化向量， 在security=1或者2时为必须参数;
centerID	string	centerID代表中心组序号;可通过/ISAPI/SecurityCP/ReportCenterCfg/?format=json接口配置中心组序号。centerID不存在时表示1

请求报文

```
<?xml version="1.0" encoding="UTF-8"?>

<Ehome xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--opt, object, EHome服务配置, attr:version{req, string, 协议版本}-->
  <enabled>
    <!--opt, bool, 服务器使能, desc:默认关闭-->false
  </enabled>
  <addressingFormatType>
    <!--req, enum, 地址类型, subType:string, [ipaddress#IP地址,hostname#主机名]-->ipaddress
  </addressingFormatType>
  <hostName>
    <!--opt, string, 主机名;当addressingFormatType=hostname时生效, range:[1,128]-->time.windows.com
  </hostName>
  <ipAddress>
    <!--opt, string, IPV4地址, range:[1,16], desc:IPV4地址-->192.168.1.112
  </ipAddress>
  <ipv6Address>
    <!--opt, string, IPV6地址, range:[1,128], desc:ipv6地址-->1030::C9B4:FF12:48AA:1A2B
  </ipv6Address>
  <portNo>
    <!--opt, int, 端口号, range:[1,65535], desc:端口号-->74
  </portNo>
  <deviceID>
    <!--req, string, ISUP设备序列号, range:[1,64]-->test1234
  </deviceID>
  <registerStatus>
    <!--ro, opt, bool, 注册状态:true为已注册,false为未注册-->true
  </registerStatus>
  <ServerAddressList>
    <!--ro, opt, array, 服务器信息列表, subType:object-->
    <ServerAddress>
      <!--ro, opt, object, 服务器信息-->
      <id>
        <!--ro, req, enum, 服务器类型, subType:string, [alarm#报警,picture#图片,ntp#ntp校时,backup#备份,enforce#取证,bayonet#卡口]-->alarm
      </id>
      <ipAddress>
        <!--ro, req, string, 服务器IP地址, range:[1,16]-->192.168.1.113
      </ipAddress>
      <portNo>
        <!--ro, req, int, 服务器端口, range:[1,65535]-->76
      </portNo>
    </ServerAddress>
  </ServerAddressList>
  <key>
    <!--ro, opt, string, 加密封钥;ISUP#0、OTAP注册时需使用, range:[1,64], dep:and,{$.Ehome.protocolVersion,eq,OTAP},{$.Ehome.protocolVersion,eq,v5.0}-->test
  </key>
  <version>
    <!--ro, opt, string, 版本号, range:[1,64]-->2.3.0.4
  </version>
  <voiceDeviceType>
    <!--opt, enum, 语音对讲设备类型, subType:string, [bluetooth#蓝牙设备,client#客户端,local#本机（本机音频口插线接入音频）]-->bluetooth
  </voiceDeviceType>
  <protocolVersion>
    <!--opt, enum, ISUP协议版本号, subType:string, [v2.0#v2.0,v2.6#v2.6,v4.0#v4.0,v5.0#v5.0,OTAP#OTAP], desc:若设备不返回该节点;则以<version>字段为准-->v5.0
  </protocolVersion>
  <network>
    <!--opt, enum, 网络, subType:string, [0#无意义,1#自动,2#有线网络优先,3#有线网络,4#3G/4G/GPRS,5#有线网络网卡1,6#有线网络网卡2]-->1
  </network>
</Ehome>
```

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 响应消息, attr:version{ro, req, string, 版本}-->
  <requestURL>
    <!--ro, req, string, 请求的URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, 状态码, subType:int, [0#OK,1#OK,2#Device Busy,3#Device Error,4#Invalid Operation,5#Invalid XML Format,6#Invalid XML Content,7#Reboot Required]-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, 状态信息, subType:string, [OK#成功,Device Busy#设备忙,Device Error#设备异常,Invalid Operation#无效的操作,Invalid XML Format#无效的XML格式,Invalid XML Content#无效的XML内容,Reboot#设备重启]-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, 详细错误码英文描述, desc:详细错误码的英文描述-->OK
  </subStatusCode>
</ResponseStatus>
```

69.13.2 获取ISUP(EHome)服务参数能力

Request URL

GET /ISAPI/System/Network/Ehome/capabilities

查询参数

无

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<Ehome xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, ehome服务器配置能力, attr:version{req, string, 协议版本}, desc:该能力为新增能力; 原来老设备不会返回该能力. 兼容方案: 对于不会返回该能力的设备; 控件只判断网络参数总能力中的<isSupportEhome>节点即可; 若设备返回该能力; 需判断该参数能力; -->
  <enabled opt="true,false">
    <!--ro, opt, bool, 使能, attr:opt{req, string, 取值范围}, desc:默认关闭-->>false
  </enabled>
  <id min="1" max="1">
    <!--ro, opt, string, 中心id, attr:min{req, int, 最小值},max{req, int, 最大值}-->test
  </id>
  <addressingFormatType opt="ipaddress,hostname">
    <!--ro, req, string, 地址类型, attr:opt{req, string, 取值范围}-->ipaddress
  </addressingFormatType>
  <GPRSAddressingFormatType opt="ipaddress">
    <!--ro, opt, string, GPRS地址类型;如不返回该字段;则根据addressingFormatType判断地址类型.代表不同的网络所配置的类型是不需要区分的, attr:opt{req, string, 取值范围}-->ipaddress
  </GPRSAddressingFormatType>
  <hostName min="1" max="128">
    <!--ro, opt, string, 主机名, attr:min{req, int, 最小值},max{req, int, 最大值}-->time.windows.com
  </hostName>
  <ipAddress min="1" max="16">
    <!--ro, opt, string, IPV4地址, attr:min{req, int, 最小值},max{req, int, 最大值}-->192.168.1.112
  </ipAddress>
  <ipv6Address min="1" max="128">
    <!--ro, opt, string, IPV6地址, attr:min{req, int, 最小值},max{req, int, 最大值}-->1030::C9B4:FF12:48AA:1A2B
  </ipv6Address>
  <portNo min="1" max="65535">
    <!--ro, opt, int, 端口号, attr:min{req, int, 最小值},max{req, int, 最大值}-->74
  </portNo>
  <deviceID min="1" max="64">
    <!--ro, req, string, ISUP设备序列号, attr:min{req, int, 最小值},max{req, int, 最大值}-->test1234
  </deviceID>
  <registerStatus opt="true,false">
    <!--ro, opt, bool, 注册状态, attr:opt{req, string, 取值范围}-->true
  </registerStatus>
  <ServerAddressList>
    <!--ro, opt, array, 服务器信息列表, subType:object-->
    <ServerAddress>
      <!--ro, opt, object, 服务器信息-->
      <id>
        <!--ro, req, enum, 服务器类型, subType:string, [alarm#报警,picture#图片,ntp#ntp校时,backup#备份,enforce#取证,bayonet#卡口]-->alarm
      </id>
      <ipAddress>
        <!--ro, req, string, 服务器IP地址-->192.168.1.113
      </ipAddress>
      <portNo>
        <!--ro, req, int, 服务器端口-->76
      </portNo>
    </ServerAddress>
  </ServerAddressList>
  <key min="1" max="64">
    <!--ro, opt, string, 加密密钥;ISUP5.0、OTAP注册时需使用, dep:and,{$.Ehome.protocolVersion,eq,OTAP},{$.Ehome.protocolVersion,eq,v5.0}, attr:min{req, int, 最小值},max{req, int, 最大值}, desc:不返回key内容,只返回属性-->>null
  </key>
  <version min="1" max="64">
    <!--ro, opt, string, 版本, attr:min{req, int, 最小值},max{req, int, 最大值}-->test
  </version>
  <voiceDeviceType opt="bluetooth,client,local">
    <!--ro, opt, string, 语音对讲设备类型: bluetooth-蓝牙设备; client-客户端; local-本地, attr:opt{req, string, 取值范围}-->test
  </voiceDeviceType>
  <protocolVersion opt="v2.0,v2.6,v4.0,v5.0,OTAP">
    <!--ro, opt, string, 协议版本, attr:opt{req, string, 取值范围}-->v5.0
  </protocolVersion>
  <network opt="0,1,2,3,4,5,6">
    <!--ro, opt, int, 网络类型, attr:opt{req, string, 取值范围}, desc:0-无, 1-自动, 2-有线网络优先, 3-有线网络, 4-3G/4G/GPRS, 5-有线网络网卡1, 6-有线网络网卡2-->1
  </network>
  <pictureServerID min="1" max="64">
    <!--ro, opt, int, 关联图片服务器id(/ISAPI/System/PictureServer), attr:min{req, int, 最小值},max{req, int, 最大值}-->1
  </pictureServerID>
  <ISUPID min="1" max="64">
    <!--ro, opt, int, ISUP多平台ID, attr:min{req, int, 最小值},max{req, int, 最大值}-->1
  </ISUPID>
  <isSupportTest>
    <!--ro, opt, bool, 是否支持ISUP(EHome)服务器测试-->true
  </isSupportTest>
  <periodicTestEnabled opt="true,false">
    <!--ro, opt, bool, 周期检测使能, attr:opt{req, string, 取值范围}, desc:使能后, 可以设置时间间隔. 报警主机每间隔一定时间给TSUP服务器发一个Test事件. 用于
```

```
<!--ro, opt, bool, 测试双方交互链路是否断开, 类似于心跳检测 (videoLoss)。-->true
</periodicTestEnabled>
<periodicTestTime min="10" max="86400">
  <!--ro, opt, int, 周期检测时间, attr:min{req, int, 最小值},max{req, int, 最大值}, desc:单位 秒, 当periodicTestEnabled为true时有效-->50
</periodicTestTime>
<isSupportISUPServerInfo>
  <!--ro, opt, bool, 是否支持获取ISUP(EHome)服务信息, desc: 对应URL: /ISAPI/System/Network/ISUPServerInfo?format=json-->true
</isSupportISUPServerInfo>
<isSupportMulticenterTest>
  <!--ro, opt, bool, 是否支持ISUP(EHome)多中心指定服务连接状态, desc:/ISAPI/System/Network/EHome/<ISUPID>/test?security=<security>&iv=<iv>-->true
</isSupportMulticenterTest>
</EHome>
```

69.13.3 获取ISUP(EHome)单中心服务参数

Request URL

GET /ISAPI/System/Network/Ehome?security= <security> &iv= <iv> ¢erID= <centerID>

查询参数

参数名称	参数类型	描述
security	string	没有security表示数据不加密； security=1,表示报文中的key字段AES128加密； security=2,表示报文中的key字段AES256加密；
iv	string	iv表示初始化向量，在security=1或者2时为必须参数;
centerID	string	centerID代表中心组序号;可通过/ISAPI/SecurityCP/ReportCenterCfg/?format=json接口配置中心组序号。centerID不存在表示centerID为1

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<Ehome xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, Ehome服务配置, attr:version{req, string, 协议版本}-->
  <enabled>
    <!--ro, opt, bool, 服务器使能, desc:默认关闭-->false
  </enabled>
  <addressingFormatType>
    <!--ro, req, enum, 地址类型, subType:string, [ipaddress#IP地址,hostname#主机名]-->ipaddress
  </addressingFormatType>
  <hostName>
    <!--ro, opt, string, 主机名;当addressingFormatType=hostname时生效, range:[1,128]-->time.windows.com
  </hostName>
  <ipAddress>
    <!--ro, opt, string, IPV4地址, range:[1,16], desc:IPV4地址-->192.168.1.112
  </ipAddress>
  <ipv6Address>
    <!--ro, opt, string, IPV6地址, range:[1,128], desc:ipv6地址-->1030::C9B4:FF12:48AA:1A2B
  </ipv6Address>
  <portNo>
    <!--ro, opt, int, 端口号, range:[1,65535], desc:端口号-->74
  </portNo>
  <deviceID>
    <!--ro, req, string, ISUP设备序列号, range:[1,64]-->test1234
  </deviceID>
  <registerStatus>
    <!--ro, opt, bool, 注册状态;true为已注册;false为未注册-->true
  </registerStatus>
  <ServerAddressList>
    <!--ro, opt, array, 服务器信息列表, subType:object-->
    <ServerAddress>
      <!--ro, opt, object, 服务器信息-->
      <id>
        <!--ro, req, enum, 服务器类型, subType:string, [alarm#报警,picture#图片,ntp#ntp校时,backup#备份,enforce#取证,bayonet#卡口]-->alarm
      </id>
      <ipAddress>
        <!--ro, req, string, 服务器IP地址, range:[1,16]-->192.168.1.113
      </ipAddress>
      <portNo>
        <!--ro, req, int, 服务器端口, range:[1,65535]-->76
      </portNo>
    </ServerAddress>
  </ServerAddressList>
  <version>
    <!--ro, opt, string, 版本号, range:[1,64]-->2.3.0.4
  </version>
  <voiceDeviceType>
    <!--ro, opt, enum, 语音对讲设备类型, subType:string, [bluetooth#蓝牙设备,client#客户端,local#本机（本机音频口插线接入音频）]-->bluetooth
  </voiceDeviceType>
  <protocolVersion>
    <!--ro, opt, enum, ISUP协议版本号, subType:string, [v2.0#v2.0,v2.6#v2.6,v4.0#v4.0,v5.0#v5.0,OTAP#OTAP], desc:若设备不返回该节点,则以<version>字段为准-->v5.0
  </protocolVersion>
  <network>
    <!--ro, opt, enum, 网络, subType:string, [0#无意义,1#自动,2#有线网络优先,3#有线网络,4#3G/4G/GPRS,5#有线网络网卡1,6#有线网络网卡2]-->1
  </network>
</Ehome>
```

69.14 武汉云存储协议接入

69.14.1 配置图片服务器参数

Request URL

PUT /ISAPI/System/PictureServer?format=json&security= <security> &iv= <iv>

查询参数

参数名称	参数类型	描述
security	enum	
iv	string	iv表示初始化向量，在security=1或者2时为必须参数;

请求报文

```

{
  "PictureServerInformation": {
    /*req, object, 图片服务器信息*/
    "pictureServerType": "cloudStorage",
    /*req, enum, 图片服务器类型, subType:string, [tomcat#tomcat,VRB#VRB,cloudStorage#云存储,KMS#KMS,null#表示图片服务器全部关闭]*/
    "addressingFormatType": "ipaddress",
    /*req, enum, 图片服务器地址格式类型, subType:string, [ipaddress#IP地址,hostname#域名]*/
    "hostName": "pc3-netsdk",
    /*opt, string, 图片服务器域名*/
    "ipv4Address": "10.17.133.46",
    /*opt, string, 图片服务器ipv4地址*/
    "ipv6Address": "2001:A304:6101:1::E0:F726:4E58",
    /*opt, string, 图片服务器ipv6地址*/
    "portNo": 1024,
    /*req, int, 图片服务器端口, range:[1024,65535]*/
    "underlyingProtocol": "http",
    /*opt, enum, 通信协议, subType:string, [http#http,https#https]*/
    "cloudStorage": {
      /*opt, object, 云存储服务器参数配置, dep:and,{$.PictureServerInformation.pictureServerType,eq,cloudStorage}*/
      "cloudManageHttpPort": 6120,
      /*opt, int, 云存储服务器中心管理Http端口（即云存储提供的图片和智能结构化数据URL下载端口），range:[1024,65535]*/
      "cloudTransDataPort": 9001,
      /*opt, int, 云存储服务器数据传输端口, range:[1024,65535]*/
      "cloudCmdPort": 6011,
      /*opt, int, 云存储服务器信令端口（客户端与云存储信令网关间的通信端口），range:[1024,65535]*/
      "cloudHeartBeatPort": 9999,
      /*opt, int, 云存储服务器心跳端口, range:[1024,65535]*/
      "cloudStorageHttpPort": 8083,
      /*opt, int, 云存储服务器存储http端口, range:[1024,65535]*/
      "cloudUsername": "username",
      /*opt, string, 云存储服务器用户名, range:[0,32]*/
      "cloudPassword": "12345",
      /*opt, string, 云存储服务器用户名*/
      "cloudPoolId": 1,
      /*opt, int, 云存储存储池ID（整型格式），range:[1,4294967295], desc:若上层没有配置设备默认填1*/
      "cloudProtocolVersion": "V2.0",
      /*req, string, 云存储服务器协议版本, range:[0,32]*/
      "cloudAccessKey": "test",
      /*opt, string, 云存储服务器access_key, range:[0,64], dep:and,{$.PictureServerInformation.cloudStorage.cloudProtocolVersion,eq,V2.0}*/
      "cloudSecretKey": "test",
      /*opt, string, 云存储服务器secret_key, range:[0,64], dep:and,{$.PictureServerInformation.cloudStorage.cloudProtocolVersion,eq,V2.0}*/
      "resourceID": "test",
      /*opt, string, 资源池ID（字符串格式），range:[1,256], dep:and,{$.PictureServerInformation.cloudStorage.cloudProtocolVersion,eq,V2.0},
      desc:与cloudPoolId功能相同，该资源池ID支持字符串；
      当设备支持该节点时可以通过该节点对池ID进行赋值，无需重复对cloudPoolId进行赋值；
      当设备不支持该节点时则需通过cloudPoolId进行赋值；
      若resourceID、cloudPoolId同时赋值时，以cloudPoolId字段为准*/
    }
  }
}

```

响应报文

```

{
  "requestURL": "test",
  /*ro, opt, string, URI*/
  "statusCode": "test",
  /*ro, opt, string, 状态码*/
  "statusString": "test",
  /*ro, opt, string, 状态描述*/
  "subStatusCode": "test",
  /*ro, opt, string, 子状态码*/
  "errorCode": 1,
  /*ro, req, int, 错误码*/
  "errorMsg": "ok",
  /*ro, req, string, 错误信息*/
}

```

69.14.2 获取图片服务器参数能力

Request URL

GET /ISAPI/System/PictureServer/capabilities?format=json

查询参数

无

请求报文

无

响应报文

```

{
  "PictureServerInformation": {
    /*ro, opt, object, 图片服务器信息*/
    "pictureServerType": {
      /*ro, opt, object, 图片服务器类型*/
      "@opt": "tomcat,VRB,cloudStorage,KMS,null",
      /*ro, opt, string, 可选值*/
      "#text": "cloudStorage"
      /*ro, req, string, 图片服务器类型*/
    },
    "addressingFormatType": {
      /*ro, opt, object, 图片服务器地址格式类型*/
      "@opt": "ipaddress,hostname",
      /*ro, opt, string, 可选值*/
      "#text": "ipaddress"
      /*ro, req, string, 图片服务器地址格式类型*/
    },
    "hostName": {
      /*ro, opt, object, 图片服务器域名*/
      "@min": 0,
      /*ro, opt, int, 最小值*/
      "@max": 64,
      /*ro, opt, int, 最大值*/
      "#text": "pc3-netsdk"
      /*ro, opt, string, 图片服务器域名*/
    },
    "ipv4Address": {
      /*ro, opt, object, 图片服务器ipv4地址*/
      "@min": 0,
      /*ro, opt, int, 最小值*/
      "@max": 64,
      /*ro, opt, int, 最大值*/
      "#text": "10.17.133.46"
      /*ro, opt, string, 图片服务器ipv4地址*/
    },
    "ipv6Address": {
      /*ro, opt, object, 图片服务器ipv6地址*/
      "@min": 0,
      /*ro, opt, int, 最小值*/
      "@max": 128,
      /*ro, opt, int, 最大值*/
      "#text": "2001:A304:6101:1::E0:F726:4E58"
      /*ro, opt, string, 图片服务器ipv6地址*/
    },
    "portNo": {
      /*ro, opt, object, 图片服务器端口*/
      "@min": 1024,
      /*ro, opt, int, 最小值*/
      "@max": 65535,
      /*ro, opt, int, 最大值*/
      "#text": 8000
      /*ro, req, int, 图片服务器端口*/
    },
    "underlyProtocol": {
      /*ro, opt, object, 通信协议*/
      "@opt": "http,https"
      /*ro, opt, string, 通信协议*/
    },
    "cloudStorage": {
      /*ro, opt, object, 云存储服务器参数配置*/
      "cloudManageHttpPort": {
        /*ro, opt, object, 云存储服务器中心管理Http端口*/
        "@min": 1024,
        /*ro, opt, int, 最小值*/
        "@max": 65535,
        /*ro, opt, int, 最大值*/
        "#text": 8009
        /*ro, opt, int, 云存储服务器中心管理Http端口*/
      },
      "cloudTransDataPort": {
        /*ro, opt, object, 云存储服务器数据传输端口*/
        "@min": 1024,
        /*ro, opt, int, 最小值*/
        "@max": 65535,
        /*ro, opt, int, 最大值*/
        "#text": 9001
        /*ro, opt, int, 云存储服务器数据传输端口*/
      },
      "cloudCmdPort": {
        /*ro, opt, object, 云存储服务器信令端口*/
        "@min": 1024,
        /*ro, opt, int, 最小值*/
        "@max": 65535,
        /*ro, opt, int, 最大值*/
        "#text": 8008
        /*ro, opt, int, 云存储服务器信令端口*/
      },
      "cloudHeartBeatPort": {
        /*ro, opt, object, 云存储服务器心跳端口*/
        "@min": 1024,
        /*ro, opt, int, 最小值*/
        "@max": 65535,
        /*ro, opt, int, 最大值*/
        "#text": 9999

```

```

/*ro, opt, int, 云存储服务器心跳端口*/
},
"cloudStorageHttpPort": {
/*ro, opt, object, 云存储服务器存储http端口*/
"@min": 1024,
/*ro, opt, int, 最小值*/
"@max": 65535,
/*ro, opt, int, 最大值*/
"#text": 8083
/*ro, opt, int, 云存储服务器存储http端口*/
},
"cloudUsername": {
/*ro, opt, object, 云存储服务器用户名*/
"@min": 0,
/*ro, opt, int, 最小值*/
"@max": 32,
/*ro, opt, int, 最大值*/
"#text": "username"
/*ro, opt, string, 云存储服务器用户名*/
},
"cloudPassword": {
/*ro, opt, object, 云存储服务器用户名*/
"@min": 0,
/*ro, opt, int, 最小值*/
"@max": 32,
/*ro, opt, int, 最大值*/
},
"cloudPoolId": {
/*ro, opt, object, 云存储存储池ID（整型格式）*/
"@min": 1,
/*ro, opt, int, 最小值*/
"@max": 4294967295,
/*ro, opt, float, 最大值*/
"#text": 1
/*ro, opt, int, 云存储存储池ID*/
},
"cloudProtocolVersion": {
/*ro, opt, object, 云存储服务器协议版本*/
"@min": 0,
/*ro, opt, int, 最小值*/
"@max": 32,
/*ro, opt, int, 最大值*/
"#text": "V2.0"
/*ro, req, string, 云存储服务器协议版本*/
},
"cloudAccessKey": {
/*ro, opt, object, 云存储服务器access_key*/
"@min": 0,
/*ro, opt, int, 最小值*/
"@max": 64,
/*ro, opt, int, 最大值*/
},
"cloudSecretKey": {
/*ro, opt, object, 云存储服务器secret_key*/
"@min": 0,
/*ro, opt, int, 最小值*/
"@max": 64,
/*ro, opt, int, 最大值*/
},
"resourceID": {
/*ro, opt, object, 云存储存储池ID（字符串格式），dep:and,{$.PictureServerInformation.cloudStorage.cloudProtocolVersion,eq,V2.0},
desc:与cloudPoolId功能相同，该资源池ID支持字符串；
当设备支持该节点时则可以通过该节点对池ID进行赋值，无需重复对cloudPoolId进行赋值；
当设备不支持该节点时则需通过cloudPoolId进行赋值；
若resourceID、cloudPoolId同时赋值时，以cloudPoolId字段为准*/
"@min": 1,
/*ro, opt, int, 最小字符长度*/
"@max": 256,
/*ro, opt, int, 最大字符长度*/
}
},
"pictureServerID": {
/*ro, opt, object, 图片服务器id*/
"@min": 1,
/*ro, opt, int, 最小值*/
"@max": 2,
/*ro, opt, int, 最大值*/
"#text": 1
/*ro, opt, int, 图片服务器id(/ISAPI/System/PictureServer)*/
}
}
}

```

69.14.3 获取图片服务器参数

Request URL

GET /ISAPI/System/PictureServer?format=json&security=<security>&iv=<iv>

查询参数

参数名称	参数类型	描述
security	enum	
iv	string	iv表示初始化向量，在security=1或者2时为必须参数;

请求报文

无

响应报文

```
{
  "PictureServerInformation": {
    /*ro, req, object, 图片服务器信息*/
    "pictureServerType": "cloudStorage",
    /*ro, req, enum, 图片服务器类型, subType:string, [tomcat#tomcat,VRB#VRB,cloudStorage#云存储,KMS#KMS,null#表示图片服务器全部关闭]*/
    "addressingFormatType": "ipaddress",
    /*ro, req, enum, 图片服务器地址格式类型, subType:string, [ipaddress#IP地址,hostname#域名]*/
    "hostName": "pc3-netsdk",
    /*ro, opt, string, 图片服务器域名*/
    "ipv4Address": "10.17.133.46",
    /*ro, opt, string, 图片服务器ipv4地址*/
    "ipv6Address": "2001:A304:6101:1::E0:F726:4E58",
    /*ro, opt, string, 图片服务器ipv6地址*/
    "portNo": 1024,
    /*ro, req, int, 图片服务器端口, range:[1024,65535]*/
    "underlyingProtocol": "http",
    /*ro, opt, enum, 通信协议, subType:string, [http#http,https#https]*/
    "cloudStorage": {
      /*ro, opt, object, 云存储服务参数配置, dep:and,{$.PictureServerInformation.pictureServerType,eq,cloudStorage}*/
      "cloudManageHttpPort": 6120,
      /*ro, opt, int, 云存储服务管理中心管理Http端口（即云存储提供的图片和智能结构化数据URL下载端口），range:[1024,65535]*/
      "cloudTransDataPort": 9001,
      /*ro, opt, int, 云存储服务数据传输端口, range:[1024,65535]*/
      "cloudCmdPort": 6011,
      /*ro, opt, int, 云存储服务信令端口（客户端与云存储信令网关间的通信端口），range:[1024,65535]*/
      "cloudHeartBeatPort": 9999,
      /*ro, opt, int, 云存储服务心跳端口, range:[1024,65535]*/
      "cloudStorageHttpPort": 8083,
      /*ro, opt, int, 云存储服务存储http端口, range:[1024,65535]*/
      "cloudUsername": "username",
      /*ro, opt, string, 云存储服务用户名, range:[0,32]*/
      "cloudPassword": "12345",
      /*ro, opt, string, 云存储服务用户名*/
      "cloudPoolId": 1,
      /*ro, opt, int, 云存储存储池ID（整型格式），range:[1,4294967295], desc:若上层没有配置设备端默认填1*/
      "cloudProtocolVersion": "V2.0",
      /*ro, req, string, 云存储服务协议版本, range:[0,32]*/
      "cloudAccessKey": "test",
      /*ro, opt, string, 云存储服务access_key, range:[0,64], dep:and,{$.PictureServerInformation.cloudStorage.cloudProtocolVersion,eq,V2.0}*/
      "cloudSecretKey": "test",
      /*ro, opt, string, 云存储服务secret_key, range:[0,64], dep:and,{$.PictureServerInformation.cloudStorage.cloudProtocolVersion,eq,V2.0}*/
      "resourceID": "test"
      /*ro, opt, string, 云存储存储池ID（字符串格式），range:[1,256], dep:and,{$.PictureServerInformation.cloudStorage.cloudProtocolVersion,eq,V2.0},
      desc:与cloudPoolId功能相同，该资源池ID支持字符串；
      当设备支持该节点时则可以通过该节点对池ID进行赋值，无需重复对cloudPoolId进行赋值；
      当设备不支持该节点时则需通过cloudPoolId进行赋值；
      若resourceID、cloudPoolId同时赋值时，以cloudPoolId字段为准*/
    }
  }
}
```

69.15 设备监听管理

69.15.1 获取报警上传控制

Request URL

GET /ISAPI/Event/notification/httpHosts/<hostID>/uploadCtrl

查询参数

参数名称	参数类型	描述
hostID	string	

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<HttpHostUploadCtrl xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, http主机上传控制, attr:version{req, string, 协议版本}-->
  <postDataTypes>
    <!--ro, opt, string, 卡口数据类型字典索引-->1,2,3
  </postDataTypes>
  <epoliceDataTypes>
    <!--ro, opt, string, 电警数据类型字典索引-->1,2,3
  </epoliceDataTypes>
  <eventDataTypes>
    <!--ro, opt, string, 事件数据类型字典索引-->1,2,3
  </eventDataTypes>
  <evidenceDataTypes>
    <!--ro, opt, string, 取证数据类型字典索引-->1,2,3
  </evidenceDataTypes>
  <uploadInterval min="1" max="2000">
    <!--ro, req, int, 上传间隔, range:[1,2000], unit:ms, unitType:时间, attr:min{req, int, 最小值},max{req, int, 最大值}, desc:默认50-->50
  </uploadInterval>
  <uploadTimeout min="1" max="6000">
    <!--ro, req, int, 上传超时时间, range:[1,6000], unit:ms, unitType:时间, attr:min{req, int, 最小值},max{req, int, 最大值}, desc:默认5000-->5000
  </uploadTimeout>
  <noPlateUploadEnabled>
    <!--ro, req, bool, 无车牌数据上传使能-->true
  </noPlateUploadEnabled>
  <historyEnabled>
    <!--ro, req, bool, 历史数据上传使能-->true
  </historyEnabled>
  <uploadTimePlan>
    <!--ro, opt, object, 上传时间计划-->
    <enabled>
      <!--ro, req, bool, 使能-->true
    </enabled>
    <timeList>
      <!--ro, opt, object, 时间列表-->
      <timeSpan>
        <!--ro, opt, object, 时间-->
        <timeId min="0" max="1">
          <!--ro, req, int, 时间id, range:[0,1], attr:min{req, int, 最小值},max{req, int, 最大值}-->1
        </timeId>
        <startHour min="0" max="23">
          <!--ro, opt, int, 开始小时, range:[0,23], attr:min{req, int, 最小值},max{req, int, 最大值}-->1
        </startHour>
        <startMinute min="0" max="59">
          <!--ro, opt, int, 开始分钟, range:[0,59], attr:min{req, int, 最小值},max{req, int, 最大值}-->1
        </startMinute>
        <endHour min="0" max="23">
          <!--ro, opt, int, 结束小时, range:[0,23], attr:min{req, int, 最小值},max{req, int, 最大值}-->1
        </endHour>
        <endMinute min="0" max="59">
          <!--ro, opt, int, 结束分钟, range:[0,59], attr:min{req, int, 最小值},max{req, int, 最大值}-->1
        </endMinute>
      </timeSpan>
    </timeList>
  </uploadTimePlan>
</HttpHostUploadCtrl>
```

69.15.2 配置单个监听主机参数

Request URL

PUT /ISAPI/Event/notification/httpHosts/<hostID>?security=<security>&iv=<iv>

查询参数

参数名称	参数类型	描述
hostID	string	通过POST /ISAPI/Event/notification/httpHosts?security=&iv=添加单个报警主机，成功会返回报警主机id
security	string	security=1,表示报文中的userName和密码字段AES128加密 security=2,表示报文中的userName和密码字段AES256加密
iv	string	iv表示初始化向量，在security=1或者2时为必须参数;

请求报文

<?xml version="1.0" encoding="UTF-8"?>

```
<HttpNotification xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, 监听主机, attr:version{req, string, 协议版本}-->
  <id>
    <!--req, string, 监听ID, range:[1,10]-->test
  </id>
  <url>
    <!--req, string, URL-->test
  </url>
  <protocolType>
    <!--req, enum, 协议类型, subType:string, [HTTP#HTTP协议,HTTPS#HTTPS协议,EHome#EHome协议]-->HTTP
  </protocolType>
  <parameterFormatType>
    <!--req, enum, 参数格式类型, subType:string, [JSON#JSON格式,XML#XML格式]-->JSON
  </parameterFormatType>
  <addressingFormatType>
    <!--req, enum, 地址类型, subType:string, [hostname#hostname,ipaddress#ipaddress]-->hostname
  </addressingFormatType>
  <hostName>
    <!--opt, string, 主机名-->test
  </hostName>
  <ipAddress>
    <!--opt, string, IP地址-->test
  </ipAddress>
  <ipv6Address>
    <!--opt, string, ipv6地址-->test
  </ipv6Address>
  <portNo>
    <!--opt, int, 端口号-->1
  </portNo>
  <userName>
    <!--opt, string, 用户名-->test
  </userName>
  <password>
    <!--opt, string, 密码-->test
  </password>
  <httpAuthenticationMethod>
    <!--req, enum, 认证方法, subType:string, [MD5digest#MD5,none#无,base64#base64]-->MD5digest
  </httpAuthenticationMethod>
  <ANPR>
    <!--opt, object, ANPR-->
    <detectioUploadPicturesType>
      <!--opt, enum, 侦测上传图片类型控制, subType:string, [all#所有,LicensePlatePicture#车牌图片,detectionPicture#检测图片]-->all
    </detectioUploadPicturesType>
  </ANPR>
  <Extensions>
    <!--opt, object, 范围-->
    <intervalBetweenEvents>
      <!--opt, int, 事件间隔时间-->1
    </intervalBetweenEvents>
  </Extensions>
  <uploadImagesDataType>
    <!--opt, enum, 图片数据类型, subType:string, [URL#URL,binary#二进制], desc:默认是二进制,在支持了云存储之后,配置成URL,才会生效-->URL
  </uploadImagesDataType>
  <httpBroken>
    <!--opt, bool, 断网续传使能, desc:开启后实现监听的断网续传功能,对所有事件有效-->true
  </httpBroken>
  <SubscribeEvent>
    <!--opt, object, 该节点表示所有含图片事件图片上传方式-->
    <heartbeat>
      <!--opt, int, 心跳间隔时间-->30
    </heartbeat>
    <eventMode>
      <!--req, enum, 事件模式, subType:string, [all#表示所有报警均需要上报,List#表示指定报警才需要上报]-->all
    </eventMode>
    <EventList>
      <!--opt, array, 指定报警上报的方式, subType:object-->
      <Event>
        <!--opt, object, 事件关联的通道信息-->
        <type>
          <!--req, enum, 事件类型, subType:string, [ADAS#高级辅助驾驶报警,ADASAlarm#高级驾驶辅助报警,AID#交通事件检测,ANPR#车牌识别,AccessControlerEvent#门禁主机检测事件,CDsStatus#光盘刻录状态上报,DBD#驾驶行为报警,GPSUpload#GPS信息上传,HFPD#高频人员报警,IO#信号量报警,IOTD#非视频设备检测事件,LES#物流扫描事件,LFPD#低频人员检测,PALMismatch#视频制式不匹配,PIR#PIR,PeopleCounting#客流量统计,PeopleNumChange#人数异常,Standup#人员站立检测,TMA#测温报警,TMPA#测温预警,VMD#移动侦测,abnormalAcceleration#设备加速度异常报警,abnormalDriving#异常驾驶行为,advReachHeight#折线攀高检测,alarmResult#人脸比对,attendance#考勤事件,attendedBaggage#物品拿取,audioAbnormal#声强突变检测,audioException#音频异常侦测,behaviorResult#行为分析结果上报,blindSpotDetection#盲区监测报警,cardMatch#认证比对,changedStatus#状态变化上报,colLision#碰撞,containerDetection#集装箱检测,crowdSituationAnalysis#人群态势分析,databaseException#数据库异常,defocus#虚焦侦测,diskUnformat#磁盘未格式化,diskerror#硬盘错误,diskfull#硬盘满,driverConditionMonitor#驾驶员状态监测报警,emergencyAlarm#紧急报警,faceCapture#人脸抓拍识别,faceSnapModeling#人脸抓拍建模,facedetection#人脸侦测,failDown#人员倒地(使用错误的拼写,faultAlarm#故障告警,fieldDetection#区域入侵,fireDetection#火点检测,fireEscapeDetection#消防通道检测,flowOverrun#流量超限,framesPeopleCounting#区域人数统计,getUp#起身检测,group#人员聚集,hdBadBlock#硬盘坏块,hdImpact#硬盘冲击,heatmap#热度图报警,highHDTemperature#硬盘温度过高,highTempAlarm#设备温度过高报警,hotSpare#热备异常,iLLAccess#非法访问,ipcTransferAbnormal#IPC迁移失败,ipconflict#IP地址冲突,keyPersonGetUp#重点人员起身检测,LeavePosition#离岗检测,linedetection#越界侦测(穿越警戒线),listSyncException#列表(名单)同步异常,Loitering#徘徊侦测,LowHDTemperature#硬盘温度过低,mixedTargetDetection#混合目标检测,modelError#模型错误,microbroken#网络断开,nodeOffline#掉线异常,nonPoliceIntrusion#非警察人员入侵,overSpeed#超速报警,overTimeTarry#超时滞留检测,parking#停车侦测,peopleNumChange#人数变化,peopleNumCounting#人数统计(区别于区域人数统计),personAbnormalAlarm#人员身份异常报警,personDensityDetection#人员密度,personQueueCounting#排队人数检测,personQueueDetection#排队检测,personQueueRealTime#人员排队检测实时数据上报,personQueueTime#排队时长检测,playCellphone#玩手机检测,pocException#视频异常,poe#POE供电异常,policeAbsent#警察不在场,radarAlarm#雷达告警,radarFieldDetection#雷达区域入侵,radarLineDetection#雷达越界侦测,radarPerimeterRule#雷达规则数据,radarTargetDetection#雷达目标数据,radarVideoDetection#雷视目标检测,raidException#阵列异常,rapidMove#快速移动,reachHeight#攀高检测,recordCycleAbnormal#录像周期不足,recordException#录像异常,regionEntrance#进入区域,regionExiting#离开区域,retention#人员滞留事件,rollOver#侧翻,running#人员奔跑,safetyHelmetDetection#安全帽检测,scenechangedetection#场景变更,sensorAlarm#角速度加速度报警,severeHDFailure#硬盘严重故障,shelfAlarm#遮挡报警,shipsDetection#船只检测,sitQuietly#静坐检测,smokeAndFireDetection#烟火检测,smokeDetection#烟雾检测,softIO#报警软IO,spacingChange#间距异常,sysStorFull#集群系统存储满异常,takingElevatorDetection#乘梯检测(电瓶车进电梯检测),targetCapture#目标抓拍,temperature#温差报警,thermometry#温度报警,thirdPartyException#第三方异常,tolLetTarry#如厕超时检测,tollCodeInfo#二维码信息上报,tossing#抛物,unattendedBaggage#物品遗留,vehicleMatchResult#黑白名单车辆报警上
```

传,vehicleRecogResult#车辆图片二次识别,versionAbnormal#集群版本异常,videoException#视频异常,videoLoss#视频丢失,violationAlarm#违规告警,violentMotion#剧烈运动,yardTarry#放风场滞留检测,AccessControllerEvent#门禁事件,IDCardInfoEvent#门禁身份证事件,FaceTemperatureMeasurementEvent#门禁测温事件,QRCodeEvent#门禁二维码事件,CertificateCaptureEvent#证件抓拍比对事件,UncertificateCompareEvent#无证比对事件,ConsumptionAndTransactionRecordEvent#消费和交易记录事件,ConsumptionEvent#消费事件,TransactionRecordEvent#交易记录事件,SetMealQuery#套餐消费查询,ConsumptionStatusQuery#消费状态查询,humanBodyComparison#人体比对,regionTargetNumberCounting#区域目标统计], desc: 参考事件类型列表eventType-->mixedTargetDetection

```
</type>
<minorAlarm>
  <!--opt, string, 报警次类型, desc: 参考事件上传宏定义;type为AccessControllerEvent, IDCardInfoEvent 该字段必填-->0x400,0x401,0x402,0x403
</minorAlarm>
<minorException>
  <!--opt, string, 异常次类型, desc: 参考事件上传宏定义;type为AccessControllerEvent, IDCardInfoEvent 该字段必填-->0x400,0x401,0x402,0x403
</minorException>
<minorOperation>
  <!--opt, string, 操作次类型, desc: 参考事件上传宏定义;type为AccessControllerEvent, IDCardInfoEvent 该字段必填-->0x400,0x401,0x402,0x403
</minorOperation>
<minorEvent>
  <!--opt, string, 事件次类型, desc: 参考事件上传宏定义;type为AccessControllerEvent, IDCardInfoEvent 该字段必填-->0x01,0x02,0x03,0x04
</minorEvent>
<pictureURLType>
  <!--opt, enum, 指定事件的报警图片格式, subType:string, [binary#二进制,LocalURL#设备本地URL,cCloudStorageURL#云存储URL,EZVIZURL#萤石URL]-->binary
</pictureURLType>
<channels>
  <!--opt, string, 监听指定通道号列表的事件, desc: 如果监听全部通道, 则不下发这个节点; 监听部分通道时, 列出通道号, 用逗号隔开-->1,2,3,4
</channels>
</Event>
</EventList>
<channels>
  <!--opt, string, 监听指定通道号列表, desc: 如果监听全部通道, 则不下发这个节点; 监听部分通道时, 列出通道号, 用逗号隔开-->1,2,3,4
</channels>
<pictureURLType>
  <!--opt, enum, 报警图片格式, subType:string, [binary#二进制,LocalURL#设备本地URL,cCloudStorageURL#云存储URL,EZVIZURL#萤石URL], desc: 该节点表示所有含图片事件图片上传方式, 如果该字段下发, 那么<Event>中<pictureURLType>不做生效; 如果不下发这个节点, 设备采用能力集返回的图片默认上传方式上报, 前端设备默认为二进制, 后端设备默认为设备本地URL-->binary
</pictureURLType>
<ChangedUploadSub>
  <!--opt, object, 订阅报文-->
  <interval>
    <!--opt, int, 布防guid的生命周期, desc: 布防guid的生命周期, 该间隔期间未过来重连, 设备认为是新的一路布防, 重新生成guid-->5
  </interval>
  <StatusSub>
    <!--opt, object, 子状态-->
    <all>
      <!--opt, bool, 是否订阅所有-->true
    </all>
    <channel>
      <!--opt, bool, 订阅通道状态, desc: all为true时不必传-->true
    </channel>
    <hd>
      <!--opt, bool, 订阅磁盘状态, desc: all为true时不必传-->true
    </hd>
    <capability>
      <!--opt, bool, 订阅能力集变化状态, desc: all为true时不必传-->true
    </capability>
  </StatusSub>
</ChangedUploadSub>
</SubscribeEvent>
<PackingSpaceRecognition>
  <!--opt, object, 当前报警主机下车位检测监听事件控制参数, desc: 对应事件为PackingSpaceRecognition-->
  <uploadPicturesType>
    <!--req, enum, 上传图片类型控制, subType:string, [all#所有,picturesTypes#指定图片类型上传,notUpload#不上上传图片]-->all
  </uploadPicturesType>
  <PicturesTypes>
    <!--opt, array, 指定图片类型上传列表, subType:object, dep:and,[$.HttpHostNotification.PackingSpaceRecognition.uploadPicturesType,eq,picturesTypes]-->
    <picturesType>
      <!--opt, enum, 上传图片类型, subType:string, [backgroundImage#背景大图,plateImage#车牌小图]-->backgroundImage
    </picturesType>
  </PicturesTypes>
</PackingSpaceRecognition>
</HttpHostNotification>
```

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 响应消息, attr:version{ro, req, string, 版本}-->
  <requestURL>
    <!--ro, req, string, 请求的URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, 状态码, subType:int, [0#OK,1#OK,2#Device Busy,3#Device Error,4#Invalid Operation,5#Invalid XML Format,6#Invalid XML Content,7#Reboot Required]-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, 状态信息, subType:string, [OK#成功,Device Busy#设备忙,Device Error#设备异常,Invalid Operation#无效的操作,Invalid XML Format#无效的XML格式,Invalid XML Content#无效的XML内容,Reboot#设备重启]-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, 详细错误码英文描述, desc:详细错误码的英文描述-->OK
  </subStatusCode>
</ResponseStatus>
```

69.15.3 获取全部监听主机参数

Request URL

GET /ISAPI/Event/notification/httpHosts?security=<security> &iv=<iv>

查询参数

参数名称	参数类型	描述
security	string	security=1,表示报文中的userName和password字段AES128加密 security=2,表示报文中的userName和password字段AES256加密
iv	string	iv表示初始化向量，在security=1或者2时为必须参数;

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<HttpHostNotificationList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, array, 监听主机列表, subType:object, attr:version{req, string, 协议版本}-->
  <HttpHostNotification>
    <!--ro, opt, object, 监听主机-->
    <id>
      <!--ro, req, string, 订阅ID, range:[1,10]-->test
    </id>
    <url>
      <!--ro, req, string, URL-->test
    </url>
    <protocolType>
      <!--ro, req, enum, 协议类型, subType:string, [HTTP#HTTP协议,HTTPS#HTTPS协议,EHome#EHome协议]-->HTTP
    </protocolType>
    <parameterFormatType>
      <!--ro, req, enum, 参数格式类型, subType:string, [JSON#JSON格式,XML#XML格式]-->JSON
    </parameterFormatType>
    <addressingFormatType>
      <!--ro, req, enum, 地址类型, subType:string, [hostname#hostname,ipaddress#ipaddress]-->hostname
    </addressingFormatType>
    <hostName>
      <!--ro, opt, string, 主机名-->test
    </hostName>
    <ipAddress>
      <!--ro, opt, string, IP地址-->test
    </ipAddress>
    <ipv6Address>
      <!--ro, opt, string, ipv6地址-->test
    </ipv6Address>
    <portNo>
      <!--ro, opt, int, 端口号-->1
    </portNo>
    <userName>
      <!--ro, opt, string, 用户名-->test
    </userName>
    <password>
      <!--ro, opt, string, 密码-->test
    </password>
    <httpAuthenticationMethod>
      <!--ro, req, enum, 认证方法, subType:string, [MD5digest#MD5,none#无,base64#base64]-->MD5digest
    </httpAuthenticationMethod>
  </HttpHostNotification>
</HttpHostNotificationList>
```

```
</httpAuthenticationMethod>
<ANPR>
  <!--ro, opt, object, ANPR-->
  <detectioUploadPicturesType>
    <!--ro, opt, enum, 侦测上传图片类型控制, subType:string, [aLL#所有,LicensePlatePicture#车牌图片,detectionPicture#检测图片]-->all
  </detectioUploadPicturesType>
</ANPR>
<Extensions>
  <!--ro, opt, object, 范围-->
  <intervalBetweenEvents>
    <!--ro, opt, int, 事件间隔时间-->1
  </intervalBetweenEvents>
</Extensions>
<uploadImagesDataType>
  <!--ro, opt, enum, 图片数据类型, subType:string, [URL#URL,binary#二进制], desc:默认是二进制,在支持了云存储之后,配置成URL,才会生效-->URL
</uploadImagesDataType>
<httpBroken>
  <!--ro, opt, bool, 断网续传使能, desc:开启后实现监听的断网续传功能,对所有事件有效-->true
</httpBroken>
<SubscribeEvent>
  <!--ro, opt, object, 该节点表示所有含图片事件图片上传方式-->
  <heartbeat>
    <!--ro, opt, int, 心跳间隔时间-->30
  </heartbeat>
  <eventMode>
    <!--ro, req, enum, 事件模式, subType:string, [aLL#表示所有报警均需要上报,list#表示指定报警才需要上报]-->all
  </eventMode>
  <eventList>
    <!--ro, opt, array, 指定报警上报的方式, subType:object-->
    <Event>
      <!--ro, opt, object, 事件关联的通道信息-->
      <type>
        <!--ro, req, enum, 事件类型, subType:string, [ADAS#高级辅助驾驶报警,ADASAlarm#高级驾驶辅助报警,AID#交通事件检测,ANPR#车牌识别,AccessControlLerEvent#门禁主机检测事件,CdsStatus#光盘刻录状态上报,DBD#驾驶行为报警,GPSUpLoad#GPS信息上传,HFPD#高频人员报警,IO#信号量报警,IOTD#非视频设备检测事件,LES#物流扫描事件,LFPD#低频人员检测,PALMismatch#视频制式不匹配,PIR#PIR,PeopleCounting#客流量统计,PeopleNumChange#人数异常,Standup#人员站立检测,TMA#测温报警,TMPA#测温预警,VMD#移动侦测,abnormalAcceleration#设备加速度异常报警,abnormalDriving#异常驾驶行为,advReachHeight#折线攀高检测,alarmResult#人脸比对,attendance#考勤事件,attendedBaggage#物品拿取,audioAbnormal#声强突变检测,audioException#音频异常侦测,behaviorResult#行为分析结果上报,blindSpotDetection#盲区监测报警,cardMatch#认证比对,changedStatus#状态变化上报,collision#碰撞,containerDetection#集装箱检测,crowdSituationAnalysis#人群态势分析,databaseException#数据库异常,defocus#虚焦侦测,diskUnformat#磁盘未格式化,diskerror#硬盘错误,diskfull#硬盘满,driverConditionMonitor#驾驶员状态监测报警,emergencyAlarm#紧急报警,faceCapture#人脸抓拍识别,faceSnapModeling#人脸抓拍建模,facedetection#人脸侦测,failDown#人员倒地（使用错误的拼写,faultAlarm#故障告警,fielddetection#区域入侵,firedetection#火点检测,fireEscapeDetection#消防通道检测,fLowOverrun#流量超限,framesPeopleCounting#区域人数统计,getUp#起身检测,group#人员聚集,hdBadBlock#硬盘坏块,hdImpact#硬盘冲击,heatmap#热度图报警,highHDTemperature#硬盘温度过高,highTempAlarm#设备温度过高报警,hotSpare#热备异常,iLLAccess#非法访问,ipcTransferAbnormal#IPC迁移失败,ipconflict#IP地址冲突,keyPersonGetUp#重点人员起身检测,LeavePosition#离岗检测,Linedetection#越界侦测（穿越警戒线）,ListsSyncException#列表（名单）同步异常,Loitering#徘徊侦测,LowHDTemperature#硬盘温度过低,mixedTargetDetection#混合目标检测,modelError#模型错误,nicbroken#网络断开,nodeOffline#掉线异常,nonPoliceIntrusion#非警察人员入侵,overSpeed#超速报警,overtimeTarry#超时滞留检测,parking#停车侦测,peopleNumChange#人数变化,peopleNumCounting#人数统计（区别于区域人数统计）,personAbnormalAlarm#人员身份异常报警,personDensityDetection#人员密度,personQueueCounting#排队人数检测,personQueueDetection#排队检测,personQueueRealTime#人员排队检测实时数据上报,personQueueTime#排队时长检测,playCellphone#玩手机检测,pocException#视频异常,poe#POE供电异常,policeAbsent#警察不在场,radarAlarm#雷达告警,radarFieldDetection#雷达区域入侵,radarLineDetection#雷达越界侦测,radarPerimeterRule#雷达规则数据,radarTargetDetection#雷达目标数据,radarVideoDetection#雷达目标检测,raidException#阵列异常,rapidMove#快速移动,reachHeight#攀高检测,recordCycleAbnormal#录像周期不足,recordException#录像异常,regionEntrance#进入区域,regionExiting#离开区域,retention#人员滞留事件,rollOver#侧翻,running#人员奔跑,safetyHelmetDetection#安全帽检测,scenechangedetection#场景变更,sensorAlarm#角速度加速度报警,severeHDFailure#硬盘严重故障,shelterAlarm#遮挡报警,shipsDetection#船只检测,sitQuietly#静坐检测,smokeAndFireDetection#烟火检测,smokeDetection#烟雾检测,softIO#报警软IO,spacingChange#间距异常,sysStorFull#集群系统存储满异常,takingElevatorDetection#乘梯检测（电瓶车进电梯检测）,targetCapture#目标抓拍,temperature#温度报警,thermometry#温度报警,thirdPartyException#第三方异常,tolLetTarry#如厕超时检测,tollCodeInfo#二维码信息上报,tossing#抛物,unattendedBaggage#物品遗留,vehicleMatchResult#黑白名单车辆报警上传,vehicleRecogResult#车辆图片二次识别,versionAbnormal#集群版本异常,videoException#视频异常,videoLoss#视频丢失,violationAlarm#违规报警,violentMotion#剧烈运动,yardTarry#放风场滞留检测,AccessControlLerEvent#门禁事件,IDCardInfoEvent#门禁身份证事件,FaceTemperatureMeasurementEvent#门禁测温事件,QRCODEEvent#门禁二维码事件,CertificateCaptureEvent#证件抓拍比对事件,UncertificateCompareEvent#无证比对事件,ConsumptionAndTransactionRecordEvent#消费和交易记录事件,ConsumptionEvent#消费事件,TransactionRecordEvent#交易记录事件,SetMealQuery#消费套餐查询,ConsumptionStatusQuery#消费状态查询,humanBodyComparison#人体比对,regionTargetNumberCounting#区域目标统计], desc:参考事件类型列表eventType-->mixedTargetDetection
      </type>
      <minorAlarm>
        <!--ro, opt, string, 报警次类型, desc:参考事件上传宏定义;type为AccessControlLerEvent, IDCardInfoEvent 该字段必填-->0x400,0x401,0x402,0x403
      </minorAlarm>
      <minorException>
        <!--ro, opt, string, 异常次类型, desc:参考事件上传宏定义;type为AccessControlLerEvent, IDCardInfoEvent 该字段必填-->0x400,0x401,0x402,0x403
      </minorException>
      <minorOperation>
        <!--ro, opt, string, 操作次类型, desc:参考事件上传宏定义;type为AccessControlLerEvent, IDCardInfoEvent 该字段必填-->0x400,0x401,0x402,0x403
      </minorOperation>
      <minorEvent>
        <!--ro, opt, string, 事件次类型, desc:参考事件上传宏定义;type为AccessControlLerEvent, IDCardInfoEvent 该字段必填-->0x01,0x02,0x03,0x04
      </minorEvent>
      <pictureURLType>
        <!--ro, opt, enum, 指定事件的报警图片格式, subType:string, [binary#二进制,LocalURL#设备本地URL,cldStorageURL#云存储URL,EZVIZURL#萤石URL]-->binary
    </Event>
  </eventList>
  <channels>
    <!--ro, opt, string, 监听指定通道号列表, desc:如果监听全部通道,则不下发这个节点:监听部分通道时,列出通道号,用逗号隔开-->1,2,3,4
  </channels>
  <Event>
  </EventList>
  <channels>
    <!--ro, opt, string, 监听指定通道号列表, desc:如果监听全部通道,则不下发这个节点:监听部分通道时,列出通道号,用逗号隔开-->1,2,3,4
  </channels>
  <pictureURLType>
    <!--ro, opt, enum, 报警图片格式, subType:string, [binary#二进制,LocalURL#设备本地URL,cldStorageURL#云存储URL,EZVIZURL#萤石URL], desc:该节点表示所有含图片事件图片上传方式,如果该字段下发,那么<Events>中<pictureURLType>不做生效:如果不下发这个节点,设备采用能力集返回的图片默认上传方式上报,前端设备默认为二进制,后端设备默认为设备本地UR-->binary
  </pictureURLType>
  <changedUploadSub>
    <!--ro, opt, object, 订阅报文-->
  </changedUploadSub>
  <interval>
    <!--ro, opt, int, 布防guid的生命周期, desc:布防guid的生命周期,该间隔期间未过来重连,设备认为是新的一路布防,重新生成guid-->5
  </interval>
```

```

</interval>
<StatusSub>
  <!--ro, opt, object, 子状态-->
  <all>
    <!--ro, opt, bool, 是否订阅所有-->true
  </all>
  <channel>
    <!--ro, opt, bool, 订阅通道状态, desc:all为true时不必传-->true
  </channel>
  <hd>
    <!--ro, opt, bool, 订阅磁盘状态, desc:all为true时不必传-->true
  </hd>
  <capability>
    <!--ro, opt, bool, 订阅能力集变化状态, desc:all为true时不必传-->true
  </capability>
</StatusSub>
</ChangedUploadSub>
</SubscribeEvent>
<PackingSpaceRecognition>
  <!--ro, req, object, 当前报警主机下车位检测监听事件控制参数, desc:对应事件为PackingSpaceRecognition-->
  <uploadPicturesType>
    <!--ro, opt, enum, 上传图片类型控制, subType:string, [all#所有,picturesTypes#指定图片类型上传,notUpload#不上传图片]-->all
  </uploadPicturesType>
  <PicturesTypes>
    <!--ro, opt, array, 指定图片类型上传列表, subType:object, dep:and,
    {$.HttpHostNotificationList[*].HttpHostNotification.uploadPicturesType,eq,picturesTypes}-->
    <picturesType>
      <!--ro, opt, enum, 上传图片类型, subType:string, [backgroundImage#背景大图,plateImage#车牌小图]-->backgroundImage
    </picturesType>
  </PicturesTypes>
</PackingSpaceRecognition>
</HttpHostNotification>
</HttpHostNotificationList>

```

69.15.4 配置全部监听主机参数

Request URL

PUT /ISAPI/Event/notification/httpHosts?security=<security>&iv=<iv>

查询参数

参数名称	参数类型	描述
security	string	security=1,表示报文中的userName和密码字段AES128加密 security=2,表示报文中的userName和密码字段AES256加密
iv	string	iv表示初始化向量，在security=1或者2时为必须参数;

请求报文

```

<?xml version="1.0" encoding="UTF-8"?>

<HttpHostNotificationList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, array, 监听主机列表, subType:object, attr:version(req, string, 协议版本)-->
  <HttpHostNotification>
    <!--opt, object, 监听主机-->
    <id>
      <!--req, string, id-->test
    </id>
    <url>
      <!--req, string, URL-->test
    </url>
    <protocolType>
      <!--req, enum, 协议类型, subType:string, [HTTP#HTTP协议,HTTPS#HTTPS协议,EHome#EHome协议]-->HTTP
    </protocolType>
    <parameterFormatType>
      <!--req, enum, 参数格式类型, subType:string, [JSON#JSON格式,XML#XML格式]-->JSON
    </parameterFormatType>
    <addressingFormatType>
      <!--req, enum, 地址类型, subType:string, [hostname#hostname,ipaddress#ipaddress]-->hostname
    </addressingFormatType>
    <hostName>
      <!--opt, string, 主机名-->test
    </hostName>
    <ipAddress>
      <!--opt, string, IP地址-->test
    </ipAddress>
    <ipv6Address>
      <!--opt, string, ipv6地址-->test
    </ipv6Address>
    <portNo>
      <!--opt, int, 端口号-->1
    </portNo>
  </HttpHostNotification>
</HttpHostNotificationList>

```

```
<userName>
  <!--opt, string, 用户名-->test
</userName>
<password>
  <!--opt, string, 密码-->test
</password>
<httpAuthenticationMethod>
  <!--req, enum, 认证方法, subType:string, [MD5digest#MD5,none#无,base64#base64]-->MD5digest
</httpAuthenticationMethod>
<ANPR>
  <!--opt, object, ANPR-->
  <detectioUploadPicturesType>
    <!--opt, enum, 侦测上传图片类型控制, subType:string, [all#所有,licensePlatePicture#车牌图片,detectionPicture#检测图片]-->all
  </detectioUploadPicturesType>
</ANPR>
<Extensions>
  <!--opt, object, 范围-->
  <intervalBetweenEvents>
    <!--opt, int, 事件间隔时间-->1
  </intervalBetweenEvents>
</Extensions>
<uploadImagesDataType>
  <!--opt, enum, 图片数据类型, subType:string, [URL#URL,binary#二进制], desc:默认是二进制,在支持了云存储之后,配置成URL,才会生效-->URL
</uploadImagesDataType>
<httpBroken>
  <!--opt, bool, 断网续传使能, desc:开启后实现监听的断网续传功能,对所有事件有效-->true
</httpBroken>
<SubscribeEvent>
  <!--opt, object, 该节点表示所有含图片事件图片上传方式-->
  <heartbeat>
    <!--opt, int, 心跳间隔时间-->30
  </heartbeat>
  <eventMode>
    <!--req, enum, 事件模式, subType:string, [all#表示所有报警均需上报,List#表示指定报警才需要上报]-->all
  </eventMode>
  <EventList>
    <!--opt, array, 指定报警上报的方式, subType:object-->
    <Event>
      <!--opt, object, 事件关联的通道信息-->
      <type>
        <!--req, enum, 事件类型, subType:string, [ADAS#高级辅助驾驶报警,ADASAlarm#高级驾驶辅助报警,AID#交通事件检测,ANPR#车牌识别,AccessControlerEvent#
门禁主机检测事件,CdsStatus#光盘刻录状态上报,DBD#驾驶行为报警,GPSUpload#GPS信息上传,HFPD#高频人员报警,IO#信号量报警,IOTD#非视频设备检测事件,LES#物流扫描事
件,LFPD#低频人员检测,PALMismatch#视频制式不匹配,PIR#PIR,PeopleNumChange#人数异常,Standup#人员站立检测,TMA#测温报警,TMPA#测温预
警,WMD#移动侦测,abnormalAcceleration#设备加速度异常报警,abnormalDriving#异常驾驶行为,advReachHeight#折线攀高检测,alarmResult#人脸比对,attendance#考勤事
件,attendedBaggage#物品拿取,audioAbnormal#声强突变检测,audioException#音频异常侦测,behaviorResult#行为分析结果上报,blindSpotDetection#盲区监测报
警,cardMatch#认证比对,changedStatus#状态变化上报,collision#碰撞,containerDetection#集装箱检测,crowdSituationAnalysis#人群态势分析,databaseException#数据库异
常,defocus#虚焦侦测,diskUnformat#磁盘未格式化,diskerror#硬盘错误,diskfull#硬盘满,driverConditionMonitor#驾驶员状态监测报警,emergencyAlarm#紧急报
警,faceCapture#人脸抓拍识别,faceSnapModeling#人脸抓拍建模,facedetection#人脸侦测,failDown#人员倒地（使用错误的拼写,faultAlarm#故障告警,fielddetection#区域入
侵,fireDetection#火点检测,fireEscapeDetection#消防通道检测,fLowOverrun#流量超限,framesPeopleCounting#区域人数统计,getUp#起身检测,group#人员聚集,hadBadBlock#
硬盘坏块,hdImpact#硬盘冲击,heatmap#热图报警,highHDTemperature#硬盘温度过高,highTempAlarm#设备温度过高报警,hotSpare#热备异常,iLLAccess#非法访
问,ipcTransferAbnormal#IPC迁移失败,ipconFlict#IP地址冲突,keyPersonGetUp#重点人员起身检测,leavePosition#离岗检测,linedetection#越界侦测（穿越警戒
线）,listsSyncException#列表（名单）同步异常,Loitering#徘徊侦测,LowHDTemperature#硬盘温度过低,mixedTargetDetection#混合目标检测,modelError#模型错
误,nicbroken#网络断开,nodeOffline#掉线异常,nonPoliceIntrusion#非警察人员入侵,overSpeed#超速报警,overTimeTarry#超时滞留检测,parking#停车侦测,peopleNumChange#
人数变化,peopleNumCounting#人数统计（区别于区域人数统计）,personAbnormalAlarm#人员身份异常报警,personDensityDetection#人员密度,personQueueCounting#排队人数
检测,personQueueDetection#排队检测,personQueueRealTime#人员排队检测实时数据上报,personQueueTime#排队时长检测,playCellphone#玩手机检测,poeException#视频异
常,poe#POE供电异常,policeAbsent#警察不在场,radarAlarm#雷达告警,radarFieldDetection#雷达区域入侵,radarLineDetection#雷达越界侦测,radarPerimeterRule#雷达规则
数据,radarTargetDetection#雷达目标数据,radarVideoDetection#雷达目标检测,raidException#阵列异常,rapidMove#快速移动,reachHeight#攀高检测,recordCycleAbnormal#
录像周期不足,recordException#录像异常,regionEntrance#进入区域,regionExiting#离开区域,retention#人员滞留事件,rollOver#侧翻,running#人员奔
跑,safetyHelmetDetection#安全帽检测,scenechangedetection#场景变更,sensorAlarm#角速度加速度报警,severeHDFailure#硬盘严重故障,shelterAlarm#遮挡报
警,shipsDetection#船只检测,sitQuietly#静坐检测,smokeAndFireDetection#烟火检测,smokeDetection#烟雾检测,softIO#报警软IO,spacingChange#间距异常,sysStarFull#集
群系统存储满异常,takingElevatorDetection#乘梯检测（电瓶车进电梯检测）,targetCapture#目标抓拍,temperature#温差报警,thermometry#温度报警,thirdPartyException#
第三方异常,totLetTarry#如厕超时检测,tolLCodeInfo#二维码信息上报,tossing#抛物,unattendedBaggage#物品遗留,vehicleMatchResult#黑白名单车辆报警上
传,vehicleRecogResult#车辆图片二次识别,versionAbnormal#集群版本异常,videoException#视频异常,videoLoss#视频丢失,violationAlarm#违规告警,violentMotion#剧烈运
动,yardTarry#放风场滞留检测,AccessControlerEvent#门禁事件,IDCardInfoEvent#门禁身份证事件,FaceTemperatureMeasurementEvent#门禁测温事件,QRCodeEvent#门禁二维
码事件,CertificateCaptureEvent#证件抓拍比对事件,UncertificateCompareEvent#无证比对事件,ConsumptionAndTransactionRecordEvent#消费和交易记录事
件,ConsumptionEvent#消费事件,TransactionRecordEvent#交易记录事件,SetMealQuery#消费套餐查询,ConsumptionStatusQuery#消费状态查询,humanBodyComparison#人体比
对,regionTargetNumberCounting#区域目标统计], desc:参考事件类型列表eventType-->mixedTargetDetection
      </type>
      <minorAlarm>
        <!--opt, string, 报警次类型, desc:参考事件上传宏定义;type为AccessControlerEvent, IDCardInfoEvent该字段必填-->0x400,0x401,0x402,0x403
      </minorAlarm>
      <minorException>
        <!--opt, string, 异常次类型, desc:参考事件上传宏定义;type为AccessControlerEvent, IDCardInfoEvent该字段必填-->0x400,0x401,0x402,0x403
      </minorException>
      <minorOperation>
        <!--opt, string, 操作次类型, desc:参考事件上传宏定义;type为AccessControlerEvent, IDCardInfoEvent该字段必填-->0x400,0x401,0x402,0x403
      </minorOperation>
      <minorEvent>
        <!--opt, string, 事件次类型, desc:参考事件上传宏定义;type为AccessControlerEvent, IDCardInfoEvent该字段必填-->0x01,0x02,0x03,0x04
      </minorEvent>
      <pictureURLType>
        <!--opt, enum, 指定事件的报警图片格式, subType:string, [binary#二进制,localURL#设备本地URL,cloudStorageURL#云存储URL,EZVIZURL#萤石URL]-->binary
      </pictureURLType>
      <channels>
        <!--opt, string, 监听指定通道号列表的事件, desc:如果监听全部通道,则不下发这个节点;监听部分通道时,列出通道号,用逗号隔开-->1,2,3,4
      </channels>
    </Event>
  </EventList>
  <channels>
    <!--opt, string, 监听指定通道号列表, desc:如果监听全部通道,则不下发这个节点;监听部分通道时,列出通道号,用逗号隔开-->1,2,3,4
  </channels>
  <pictureURLType>
    <!--opt, enum, 报警图片格式, subType:string, [binary#二进制,localURL#设备本地URL,cloudStorageURL#云存储URL,EZVIZURL#萤石URL], desc:该节点表示所有含
图片事件图片上传方式.如果该字段不发.那么<Event>中<pictureURLType>不做生效;如果不下发这个节点.设备采用能力集返回的图片默认上传方式上报.前端设备默认为二进制.

```

```

后端设备默认为设备本地UR-->binary
</pictureURLType>
<ChangedUploadSub>
  <!--opt, object, 订阅报文-->
  <interval>
    <!--opt, int, 布防guid的生命周期, desc:布防guid的生命周期,该间隔期间未过来重连,设备认为是新的一路布防,重新生成guid-->5
  </interval>
  <StatusSub>
    <!--opt, object, 子状态-->
    <all>
      <!--opt, bool, 是否订阅所有-->true
    </all>
    <channel>
      <!--opt, bool, 订阅通道状态, desc:all为true时不必传-->true
    </channel>
    <hd>
      <!--opt, bool, 订阅磁盘状态, desc:all为true时不必传-->true
    </hd>
    <capability>
      <!--opt, bool, 订阅能力集变化状态, desc:all为true时不必传-->true
    </capability>
  </StatusSub>
</ChangedUploadSub>
</SubscribeEvent>
<PackingSpaceRecognition>
  <!--opt, object, 当前报警主机下车位检测监听事件控制参数, desc:对应事件为PackingSpaceRecognition-->
  <uploadPicturesType>
    <!--req, enum, 上传图片类型控制, subType:string, [all#所有,picturesTypes#指定图片类型上传,notUpload#不上传图片]-->all
  </uploadPicturesType>
  <PicturesTypes>
    <!--opt, array, 指定图片类型上传列表, subType:object, dep:and,
    {$.HostNameNotificationList[*].HostNameNotification.PackingSpaceRecognition.uploadPicturesType.eq,picturesTypes}-->
    <picturesType>
      <!--opt, enum, 上传图片类型, subType:string, [backgroundImage#背景大图,plateImage#车牌小图]-->backgroundImage
    </picturesType>
  </PicturesTypes>
</PackingSpaceRecognition>
</HostNameNotification>
</HostNameNotificationList>

```

响应报文

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 响应消息, attr:version{ro, req, string, 版本}-->
  <requestURL>
    <!--ro, req, string, 请求的URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, 状态码, subType:int, [0#OK,1#OK,2#Device Busy,3#Device Error,4#Invalid Operation,5#Invalid XML Format,6#Invalid XML Content,7#Reboot Required]-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, 状态信息, subType:string, [OK#成功,Device Busy#设备忙,Device Error#设备异常,Invalid Operation#无效的操作,Invalid XML Format#无效的XML格式,Invalid XML Content#无效的XML内容,Reboot#设备重启]-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, 详细错误码英文描述, desc:详细错误码的英文描述-->OK
  </subStatusCode>
</ResponseStatus>

```

69.15.5 获取单个监听主机参数

Request URL

GET /ISAPI/Event/notification/httpHosts/<hostID>?security=<security>&iv=<iv>

查询参数

参数名称	参数类型	描述
hostID	string	通过POST /ISAPI/Event/notification/httpHosts?security=&iv=添加单个报警主机，成功会返回报警主机id
security	string	security=1,表示报文中的userName和password字段AES128加密 security=2,表示报文中的userName和password字段AES256加密
iv	string	iv表示初始化向量，在security=1或者2时为必须参数;

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<HttpHostNotification xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 监听主机, attr:version{req, string, 协议版本}-->
  <id>
    <!--ro, req, string, 监听ID, range:[1,10]-->test
  </id>
  <url>
    <!--ro, req, string, URL-->test
  </url>
  <protocolType>
    <!--ro, req, enum, 协议类型, subType:string, [HTTP#HTTP协议,HTTPS#HTTPS协议,EHome#EHome协议]-->HTTP
  </protocolType>
  <parameterFormatType>
    <!--ro, req, enum, 参数格式类型, subType:string, [JSON#JSON格式,XML#XML格式]-->JSON
  </parameterFormatType>
  <addressingFormatType>
    <!--ro, req, enum, 地址类型, subType:string, [hostname#hostname,ipaddress#ipaddress]-->hostname
  </addressingFormatType>
  <hostName>
    <!--ro, opt, string, 主机名-->test
  </hostName>
  <ipAddress>
    <!--ro, opt, string, IP地址-->test
  </ipAddress>
  <ipv6Address>
    <!--ro, opt, string, ipv6地址-->test
  </ipv6Address>
  <portNo>
    <!--ro, opt, int, 端口号-->1
  </portNo>
  <userName>
    <!--ro, opt, string, 用户名-->test
  </userName>
  <httpAuthenticationMethod>
    <!--ro, req, enum, 认证方法, subType:string, [MD5digest#MD5,none#无,base64#base64]-->MD5digest
  </httpAuthenticationMethod>
  <ANPR>
    <!--ro, opt, object, ANPR-->
    <detectioUploadPicturesType>
      <!--ro, opt, enum, 侦测上传图片类型控制, subType:string, [aLL#所有,LicensePlatePicture#车牌图片,detectionPicture#检测图片]-->aLL
    </detectioUploadPicturesType>
  </ANPR>
  <Extensions>
    <!--ro, opt, object, 范围-->
    <intervalBetweenEvents>
      <!--ro, opt, int, 事件间隔时间-->1
    </intervalBetweenEvents>
  </Extensions>
  <uploadImagesDataType>
    <!--ro, opt, enum, 图片数据类型, subType:string, [URL#URL,binary#二进制], desc:默认是二进制,在支持了云存储之后,配置成URL,才会生效-->URL
  </uploadImagesDataType>
  <httpBroken>
    <!--ro, opt, bool, 断网续传使能, desc:开启后实现监听的断网续传功能,对所有事件有效-->true
  </httpBroken>
  <SubscribeEvent>
    <!--ro, opt, object, 该节点表示所有含图片事件图片上传方式-->
    <heartbeat>
      <!--ro, opt, int, 心跳间隔时间-->30
    </heartbeat>
    <eventMode>
      <!--ro, req, enum, 事件模式, subType:string, [aLL#表示所有报警均需要上报,List#表示指定报警才需要上报]-->aLL
    </eventMode>
    <EventList>
      <!--ro, opt, array, 指定报警上报的方式, subType:object-->
      <Event>
        <!--ro, opt, object, 事件关联的通道信息-->
        <type>
          <!--ro, req, enum, 事件类型, subType:string, [ADAS#高级辅助驾驶报警,ADASAlarm#高级驾驶辅助报警,AID#交通事件检测,ANPR#车牌识别,AccessControllerEvent#门禁主机检测事件,CdsStatus#光盘刻录状态上报,DBD#驾驶行为报警,GPSUpload#GPS信息上传,HFPD#高频人员报警,IO#信号量报警,IOTD#非视频设备检测事件,LES#物流扫描事件,LFPD#低频人员检测,PALMismatch#视频制式不匹配,PIR#PIR,PeopLeCounting#客流量统计,PeopLeNumChange#人数异常,Standup#人员站立检测,TMA#测温报警,TMPA#测温预警,VMD#移动侦测,abnormalAcceleration#设备加速度异常报警,abnormalDriving#异常驾驶行为,advReachHeight#折线攀高检测,alarmResult#人脸比对,attendance#考勤事件,attendedBaggage#物品拿取,audioAbnormal#声强突变检测,audioexception#音频异常侦测,behaviorResult#行为分析结果上报,blindSpotDetection#盲区监测报警,cardMatch#认证比对,changedStatus#状态变化上报,collision#碰撞,containerDetection#集装箱检测,crowdSituationAnalysis#人群态势分析,databaseException#数据库异常,defocus#虚焦侦测,diskUnformat#磁盘未格式化,diskerror#硬盘错误,diskfull#硬盘满,driverConditionMonitor#驾驶员状态监测报警,emergencyAlarm#紧急报警,faceCapture#人脸抓拍识别,faceSnapModeling#人脸抓拍建模,facedetection#人脸侦测,failDown#人员倒地(使用错误的拼写,faultAlarm#故障告警,fieldDetection#区域入侵,fireDetection#火点检测,fireEscapeDetection#消防通道检测,flowOverrun#流量超限,framesPeopleCounting#区域人数统计,getUp#起身检测,group#人员聚集,hdBadBlock#硬盘坏块,hdImpact#硬盘冲击,heatmap#热度图报警,highHDTemperature#硬盘温度过高,highTempAlarm#设备温度过高报警,hotSpare#热备异常,iLlAccess#非法访问,ipcTransferAbnormal#IPC迁移失败,ipconflict#IP地址冲突,keyPersonGetUp#重点人员起身检测,LeavePosition#离岗检测,Linedetection#越界侦测(穿越警戒线),ListSyncException#列表(名单)同步异常,Loitering#徘徊侦测,LowHDTemperature#硬盘温度过低,mixedTargetDetection#混合目标检测,modelError#模型错误,nicbroke#网络断开,nodeOffline#掉线异常,nonPoliceIntrusion#非警察人员入侵,overSpeed#超速报警,overtimeTarry#超时滞留检测,parking#停车侦测,peopLeNumChange#人数变化,peopLeNumCounting#人数统计(区别于区域人数统计),personAbnormalAlarm#人员身份异常报警,personDensityDetection#人员密度,personQueueCounting#排队人数检测,personQueueDetection#排队检测,personQueueRealTime#人员排队检测实时数据上报,personQueueTime#排队时长检测,pLayCellphone#玩手机检测,pocException#视频异常,poe#POE供电异常,policeAbsent#警察不在场,radarAlarm#雷达告警,radarFieldDetection#雷达区域入侵,radarLineDetection#雷达越界侦测,radarPerimeterRule#雷达规则数据,radarTargetDetection#雷达目标数据,radarVideoDetection#雷视目标检测,raidException#阵列异常,rapidMove#快速移动,reachHeight#攀高检测,recordCycleAbnormal#录像周期异常,recordEvention#录像异常,regionEnter#进入区域,regionExit#离开区域,retention#人员滞留事件,rollEvent#侧翻,running#人员奔
```

水浸/漏水检测, fireException#火灾异常, regionNumberCounting#区域目标统计, desc: 参考事件类型列表 eventType-->mixedTargetDetection
 跑, safetyHelmetDetection#安全帽检测, scenechangedetection#场景变更, sensorAlarm#角速度加速度报警, severeHdFailure#硬盘严重故障, shelterAlarm#遮挡报警, shipsDetection#船只检测, sitQuietly#静坐检测, smokeAndFireDetection#烟火检测, smokeDetection#烟雾检测, softIO#报警软IO, spacingChange#间距异常, sysStorageFull#集群系统存储满异常, takingElevatorDetection#乘梯检测 (电瓶车进电梯检测), targetCapture#目标抓拍, temperature#温差报警, thermometer#温度报警, thirdPartyException#第三方异常, tollETarrry#如厕超时检测, tollCodeInfo#二维码信息上报, tossing#抛物, unattendedBaggage#物品遗留, vehicleMatchResult#黑白名单车辆报警上传, vehicleRecognResult#车辆图片二次识别, versionAbnormal#集群版本异常, videoException#视频异常, videoLoss#视频丢失, violationAlarm#违规告警, violentMotion#剧烈运动, yardTarrry#放风场滞留检测, AccessControlLerEvent#门禁事件, IDCardInfoEvent#门禁身份证事件, FaceTemperatureMeasurementEvent#门禁测温事件, QRCodeEvent#门禁二维码事件, CertificateCaptureEvent#证件抓拍比对事件, UncertificateCompareEvent#无证比对事件, ConsumptionAndTransactionRecordEvent#消费和交易记录事件, ConsumptionEvent#消费事件, TransactionRecordEvent#交易记录事件, SetMealQuery#消费套餐查询, ConsumptionStatusQuery#消费状态查询, humanBodyComparison#人体比对, regionTargetNumberCounting#区域目标统计, desc: 参考事件类型列表 eventType-->mixedTargetDetection

```

</type>
<minorAlarm>
  <!--ro, opt, string, 报警次类型, desc: 参考事件上传宏定义; type为AccessControlLerEvent, IDCardInfoEvent 该字段必填-->0x400, 0x401, 0x402, 0x403
</minorAlarm>
<minorException>
  <!--ro, opt, string, 异常次类型, desc: 参考事件上传宏定义; type为AccessControlLerEvent, IDCardInfoEvent 该字段必填-->0x400, 0x401, 0x402, 0x403
</minorException>
<minorOperation>
  <!--ro, opt, string, 操作次类型, desc: 参考事件上传宏定义; type为AccessControlLerEvent, IDCardInfoEvent 该字段必填-->0x400, 0x401, 0x402, 0x403
</minorOperation>
<minorEvent>
  <!--ro, opt, string, 事件次类型, desc: 参考事件上传宏定义; type为AccessControlLerEvent, IDCardInfoEvent 该字段必填-->0x01, 0x02, 0x03, 0x04
</minorEvent>
<pictureURLType>
  <!--ro, opt, enum, 指定事件的报警图片格式, subType:string, [binary#二进制, LocalURL#设备本地URL, cloudStorageURL#云存储URL, EZVIZURL#萤石URL]-->binary
</pictureURLType>
<channels>
  <!--ro, opt, string, 监听指定通道号列表的事件, desc: 如果监听全部通道, 则不下发这个节点; 监听部分通道时, 列出通道号, 用逗号隔开-->1, 2, 3, 4
</channels>
<Event>
</EventList>
<channels>
  <!--ro, opt, string, 监听指定通道号列表, desc: 如果监听全部通道, 则不下发这个节点; 监听部分通道时, 列出通道号, 用逗号隔开-->1, 2, 3, 4
</channels>
<pictureURLType>
  <!--ro, opt, enum, 报警图片格式, subType:string, [binary#二进制, LocalURL#设备本地URL, cloudStorageURL#云存储URL, EZVIZURL#萤石URL], desc: 该节点表示所有含图片事件图片上传方式, 如果该字段下发, 那么<Event>中<pictureURLType>不做生效; 如果不下发这个节点, 设备采用能力集返回的图片默认上传方式上报, 前端设备默认为二进制, 后端设备默认为设备本地URL-->binary
</pictureURLType>
<ChangedUploadSub>
  <!--ro, opt, object, 订阅报文-->
<interval>
  <!--ro, opt, int, 布防guid的生命周期, desc: 布防guid的生命周期, 该间隔期间未过来重连, 设备认为是新的一路布防, 重新生成guid-->5
</interval>
<StatusSub>
  <!--ro, opt, object, 子状态-->
<all>
  <!--ro, opt, bool, 是否订阅所有-->true
</all>
<channel>
  <!--ro, opt, bool, 订阅通道状态, desc: all为true时不必传-->true
</channel>
<hd>
  <!--ro, opt, bool, 订阅磁盘状态, desc: all为true时不必传-->true
</hd>
<capability>
  <!--ro, opt, bool, 订阅能力集变化状态, desc: all为true时不必传-->true
</capability>
</StatusSub>
</ChangedUploadSub>
</SubscribeEvent>
<PackingSpaceRecognition>
  <!--ro, opt, object, 当前报警主机下车位检测监听事件控制参数, desc: 对应事件为PackingSpaceRecognition-->
<uploadPicturesType>
  <!--ro, req, enum, 上传图片类型控制, subType:string, [all#所有, picturesTypes#指定图片类型上传, notUpload#不上上传图片]-->all
</uploadPicturesType>
<PicturesTypes>
  <!--ro, opt, array, 指定图片类型上传列表, subType:object, dep:and, {$$.HttpHostNotification.PackingSpaceRecognition.uploadPicturesType, eq, picturesTypes}-->
<picturesType>
  <!--ro, opt, enum, 上传图片类型, subType:string, [backgroundImage#背景大图, plateImage#车牌小图]-->backgroundImage
</picturesType>
</PicturesTypes>
</PackingSpaceRecognition>
</HttpHostNotification>
  
```

69.15.6 配置单个监听主机参数

Request URL

PUT /ISAPI/Event/notification/httpHosts/<hostID>?security=<security>&iv=<iv>

查询参数

参数名称	参数类型	描述
hostID	string	通过POST /ISAPI/Event/notification/httpHosts?security=&iv=添加单个报警主机, 成功会返回报警主机id
security	string	security=1,表示报文中的userName和password字段AES128加密 security=2,表示报文中的userName和password字段AES256加密
iv	string	iv表示初始化向量, 在security=1或者2时为必须参数;

请求报文

```
<?xml version="1.0" encoding="UTF-8"?>

<HttpHostNotification xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, 监听主机, attr:version{req, string, 协议版本}-->
  <id>
    <!--req, string, 监听ID, range:[1,10]-->test
  </id>
  <url>
    <!--req, string, URL-->test
  </url>
  <protocolType>
    <!--req, enum, 协议类型, subType:string, [HTTP#HTTP协议,HTTPS#HTTPS协议,EHome#EHome协议]-->HTTP
  </protocolType>
  <parameterFormatType>
    <!--req, enum, 参数格式类型, subType:string, [JSON#JSON格式,XML#XML格式]-->JSON
  </parameterFormatType>
  <addressingFormatType>
    <!--req, enum, 地址类型, subType:string, [hostname#hostname,ipaddress#ipaddress]-->hostname
  </addressingFormatType>
  <hostName>
    <!--opt, string, 主机名-->test
  </hostName>
  <ipAddress>
    <!--opt, string, IP地址-->test
  </ipAddress>
  <ipv6Address>
    <!--opt, string, ipv6地址-->test
  </ipv6Address>
  <portNo>
    <!--opt, int, 端口号-->1
  </portNo>
  <userName>
    <!--opt, string, 用户名, dep:and,{$.HttpHostNotification.httpAuthenticationMethod,eq,MD5digest}-->test
  </userName>
  <password>
    <!--opt, string, 密码, dep:and,{$.HttpHostNotification.httpAuthenticationMethod,eq,MD5digest}-->test
  </password>
  <httpAuthenticationMethod>
    <!--req, enum, 认证方法, subType:string, [MD5digest#MD5,none#无,base64#base64]-->MD5digest
  </httpAuthenticationMethod>
  <ANPR>
    <!--opt, object, ANPR-->
    <detectioUploadPicturesType>
      <!--opt, enum, 侦测上传图片类型控制, subType:string, [all#所有,LicensePlatePicture#车牌图片,detectionPicture#检测图片]-->all
    </detectioUploadPicturesType>
  </ANPR>
  <Extensions>
    <!--opt, object, 范围-->
    <intervalBetweenEvents>
      <!--opt, int, 事件间隔时间-->1
    </intervalBetweenEvents>
  </Extensions>
  <uploadImagesDataType>
    <!--opt, enum, 图片数据类型, subType:string, [URL#URL,binary#二进制], desc:默认是二进制,在支持了云存储之后,配置成URL,才会生效-->URL
  </uploadImagesDataType>
  <httpBroken>
    <!--opt, bool, 断网续传使能, desc:开启后实现监听的断网续传功能,对所有事件有效-->true
  </httpBroken>
  <SubscribeEvent>
    <!--opt, object, 该节点表示所有含图片事件图片上传方式-->
    <heartbeat>
      <!--opt, int, 心跳间隔时间-->30
    </heartbeat>
    <eventMode>
      <!--req, enum, 事件模式, subType:string, [all#表示所有报警均需要上报,List#表示指定报警才需要上报]-->all
    </eventMode>
    <EventList>
      <!--opt, array, 指定报警上报的方式, subType:object-->
      <Event>
        <!--opt, object, 事件关联的通道信息-->
        <type>
          <!--req, enum, 事件类型, subType:string, [ADAS#高级辅助驾驶报警,ADASAlarm#高级驾驶辅助报警,AID#交通事件检测,ANPR#车牌识别,AccessControlLerEvent#门禁主机检测事件,CdsStatus#光盘刻录状态上报,DBD#驾驶行为报警,GPSUpload#GPS信息上传,HFPD#高频人员报警,IO#信号量报警,IOTD#非视频设备检测事件,LES#物流扫描事
```

件, LFPD#低频人员检测, PALMismatch#视频制式不匹配, PIR#PIR, PeopleCounting#客流量统计, PeopleNumChange#人数异常, Standup#人员站立检测, TMA#测温报警, TMPA#测温预警, VMD#移动侦测, abnormalAcceleration#设备加速度异常报警, abnormalDriving#异常驾驶行为, advReachHeight#折线攀高检测, alarmResult#人脸比对, attendance#考勤事件, attendedBaggage#物品拿取, audioAbnormal#声强突变检测, audioException#音频异常侦测, behaviorResult#行为分析结果上报, blindSpotDetection#盲区监测报警, cardMatch#认证比对, changedStatus#状态变化上报, collision#碰撞, containerDetection#集装箱检测, crowdSituationAnalysis#人群态势分析, databaseException#数据库异常, defocus#虚焦侦测, diskUnformat#磁盘未格式化, diskerror#硬盘错误, diskfull#硬盘满, driverConditionMonitor#驾驶员状态监测报警, emergencyAlarm#紧急报警, faceCapture#人脸抓拍识别, faceSnapModeling#人脸抓拍建模, facedetection#人脸侦测, failDown#人员倒地 (使用错误的拼写, faultAlarm#故障告警, fieldDetection#区域入侵, fireDetection#火点检测, fireEscapeDetection#消防通道检测, fLowOverrun#流量超限, framesPeopleCounting#区域人数统计, getUp#起身检测, group#人员聚集, hdBadBlock#硬盘坏块, hdImpact#硬盘冲击, heatmap#热度图报警, highHDTemperature#硬盘温度过高, highTempAlarm#设备温度过高报警, hotSpare#热备异常, iLLAccess#非法访问, ipcTransferAbnormal#IPC迁移失败, ipconflict#IP地址冲突, keyPersonGetUp#重点人员起身检测, LeavePosition#离岗检测, linedetection#越界侦测 (穿越警戒线), ListsyncException#列表 (名单) 同步异常, Loitering#徘徊侦测, LowHDTemperature#硬盘温度过低, mixedTargetDetection#混合目标检测, modeError#模型错误, nicbroken#网络断开, nodeOffline#掉线异常, nonPoliceIntrusion#非警察人员入侵, overSpeed#超速报警, overtimeTarry#超时滞留检测, parking#停车侦测, peopleNumChange#人数变化, peopleNumCounting#人数统计 (区别于区域人数统计), personAbnormalAlarm#人员身份异常报警, personDensityDetection#人员密度, personQueueCounting#排队人数检测, personQueueDetection#排队检测, personQueueRealTime#人员排队检测实时数据上报, personQueueTime#排队时长检测, playCellPhone#玩手机检测, pocException#视频异常, poe#POE供电异常, policeAbsent#警察不在场, radarAlarm#雷达告警, radarFieldDetection#雷达区域入侵, radarLineDetection#雷达越界侦测, radarPerimeterRule#雷达规则数据, radarTargetDetection#雷达目标数据, radarVideoDetection#雷达目标检测, raidException#阵列异常, rapidMove#快速移动, reachHeight#攀高检测, recordCycleAbnormal#录像周期不足, recordException#录像异常, regionEntrance#进入区域, regionExiting#离开区域, retention#人员滞留事件, rollover#侧翻, running#人员奔跑, safetyHelmetDetection#安全帽检测, scenechangedetection#场景变更, sensorAlarm#角速度加速度报警, severeHDFailure#硬盘严重故障, shelterAlarm#遮挡报警, shipsDetection#船只检测, sitQuietly#静坐检测, smokeAndFireDetection#烟火检测, smokeDetection#烟雾检测, softIO#报警软IO, spacingChange#间距异常, sysStorFull#集群系统存储满异常, takingElevatorDetection#乘梯检测 (电瓶车进电梯检测), targetCapture#目标抓拍, temperature#温差报警, thermometry#温度报警, thirdPartyException#第三方异常, toiletTarry#如厕超时检测, tollCodeInfo#二维码信息上报, tossing#抛物, unattendedBaggage#物品遗留, vehicleMatchResult#黑白名单车辆报警上传, vehicleRcogResult#车辆图片二次识别, versionAbnormal#集群版本异常, videoException#视频异常, videoLoss#视频丢失, violationAlarm#违规告警, violentMotion#剧烈运动, yardTarry#放风场滞留检测, AccessControllerEvent#门禁事件, IDCardInfoEvent#门禁身份证事件, FaceTemperatureMeasurementEvent#门禁测温事件, QRCodeEvent#门禁二维码事件, CertificateCaptureEvent#证件抓拍比对事件, UncertificateCompareEvent#无证比对事件, ConsumptionAndTransactionRecordEvent#消费和交易记录事件, ConsumptionEvent#消费事件, TransactionRecordEvent#交易记录事件, SetMealQuery#套餐消费查询, ConsumptionStatusQuery#消费状态查询, humanBodyComparison#人体比对, regionTargetNumberCounting#区域目标统计, paymentQRCodeQuery#查询三方支付二维码, ConsumptionQuery#本地消费查询事件, topUpRequest#充值请求, topUpStatusQuery#查询充值状态, topUpEvent#充值事件, LogOffRefundStatusQuery#查询人员注销退款状态, LogOffRefundRequest#人员注销退款请求], desc:参考事件类型列表eventType-->mixedTargetDetection

```
</type>
<minorAlarm>
  <!--opt, string, 报警次类型, desc:参考事件上传宏定义;type为AccessControlLerEvent, IDCardInfoEvent 该字段必填-->0x400,0x401,0x402,0x403
</minorAlarm>
<minorException>
  <!--opt, string, 异常次类型, desc:参考事件上传宏定义;type为AccessControlLerEvent, IDCardInfoEvent 该字段必填-->0x400,0x401,0x402,0x403
</minorException>
<minorOperation>
  <!--opt, string, 操作次类型, desc:参考事件上传宏定义;type为AccessControlLerEvent, IDCardInfoEvent 该字段必填-->0x400,0x401,0x402,0x403
</minorOperation>
<minorEvent>
  <!--opt, string, 事件次类型, desc:参考事件上传宏定义;type为AccessControlLerEvent, IDCardInfoEvent 该字段必填-->0x01,0x02,0x03,0x04
</minorEvent>
<pictureURLType>
  <!--opt, enum, 指定事件的报警图片格式, subType:string, [binary#二进制, LocalURL#设备本地URL, cloudStorageURL#云存储URL, EZVIZURL#萤石URL]-->binary
</pictureURLType>
<channels>
  <!--opt, string, 监听指定通道号列表的事件, desc:如果监听全部通道, 则不下发这个节点; 监听部分通道时, 列出通道号, 用逗号隔开-->1,2,3,4
</channels>
</Event>
</EventList>
<channels>
  <!--opt, string, 监听指定通道号列表, desc:如果监听全部通道, 则不下发这个节点; 监听部分通道时, 列出通道号, 用逗号隔开-->1,2,3,4
</channels>
<pictureURLType>
  <!--opt, enum, 报警图片格式, subType:string, [binary#二进制, LocalURL#设备本地URL, cloudStorageURL#云存储URL, EZVIZURL#萤石URL], desc:该节点表示所有含图片事件图片上传方式, 如果该字段下发, 那么<Event>中<pictureURLType>不做生效; 如果不下发这个节点, 设备采用能力集返回的图片默认上传方式上报, 前端设备默认为二进制, 后端设备默认为设备本地URL-->binary
</pictureURLType>
<ChangedUploadSub>
  <!--opt, object, 订阅报文-->
  <interval>
    <!--opt, int, 布防guid的生命周期, desc:布防guid的生命周期, 该间隔期间未过来重连, 设备认为是新的一路布防, 重新生成guid-->5
  </interval>
  <StatusSub>
    <!--opt, object, 子状态-->
    <all>
      <!--opt, bool, 是否订阅所有-->true
    </all>
    <channel>
      <!--opt, bool, 订阅通道状态, desc:all为true时不必传-->true
    </channel>
    <hd>
      <!--opt, bool, 订阅磁盘状态, desc:all为true时不必传-->true
    </hd>
    <capability>
      <!--opt, bool, 订阅能力集变化状态, desc:all为true时不必传-->true
    </capability>
  </StatusSub>
</ChangedUploadSub>
</SubscribeEvent>
<PackingSpaceRecognition>
  <!--opt, object, 当前报警主机下车位检测监听事件控制参数, desc:对应事件为PackingSpaceRecognition-->
  <uploadPicturesType>
    <!--req, enum, 上传图片类型控制, subType:string, [all#所有, picturesTypes#指定图片类型上传, notUpload#不上上传图片]-->all
  </uploadPicturesType>
  <PicturesTypes>
    <!--opt, array, 指定图片类型上传列表, subType:object, dep:and,{$.HttpRequest.Notification.PackingSpaceRecognition.uploadPicturesType,eq,picturesTypes}-->
    <picturesType>
      <!--opt, enum, 上传图片类型, subType:string, [backgroundImage#背景大图, plateImage#车牌小图]-->backgroundImage
    </picturesType>
  </PicturesTypes>
</PackingSpaceRecognition>
</HttpRequestNotification>
```

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 响应消息, attr:version{ro, req, string, 版本}-->
  <requestURL>
    <!--ro, req, string, 请求的URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, 状态码, subType:int, [0#OK,1#OK,2#Device Busy,3#Device Error,4#Invalid Operation,5#Invalid XML Format,6#Invalid XML Content,7#Reboot Required]-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, 状态信息, subType:string, [OK#成功,Device Busy#设备忙,Device Error#设备异常,Invalid Operation#无效的操作,Invalid XML Format#无效的XML格式,Invalid XML Content#无效的XML内容,Reboot#设备重启]-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, 详细错误码英文描述, desc:详细错误码的英文描述-->OK
  </subStatusCode>
</ResponseStatus>
```

69.15.7 获取单个监听主机参数

Request URL

GET /ISAPI/Event/notification/httpHosts/<hostID>?security=<security>&iv=<iv>

查询参数

参数名称	参数类型	描述
hostID	string	通过POST /ISAPI/Event/notification/httpHosts?security=&iv=添加单个报警主机，成功会返回报警主机id
security	string	security=1,表示报文中的userName和password字段AES128加密 security=2,表示报文中的userName和password字段AES256加密
iv	string	iv表示初始化向量，在security=1或者2时为必须参数;

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<HttpHostNotification xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 监听主机, attr:version{req, string, 协议版本}-->
  <id>
    <!--ro, req, string, 监听ID, range:[1,10]-->test
  </id>
  <url>
    <!--ro, req, string, URL-->test
  </url>
  <protocolType>
    <!--ro, req, enum, 协议类型, subType:string, [HTTP#HTTP协议,HTTPS#HTTPS协议,EHome#EHome协议]-->HTTP
  </protocolType>
  <parameterFormatType>
    <!--ro, req, enum, 参数格式类型, subType:string, [JSON#JSON格式,XML#XML格式]-->JSON
  </parameterFormatType>
  <addressingFormatType>
    <!--ro, req, enum, 地址类型, subType:string, [hostname#hostname,ipaddress#ipaddress]-->hostname
  </addressingFormatType>
  <hostName>
    <!--ro, opt, string, 主机名-->test
  </hostName>
  <ipAddress>
    <!--ro, opt, string, IP地址-->test
  </ipAddress>
  <ipv6Address>
    <!--ro, opt, string, ipv6地址-->test
  </ipv6Address>
  <portNo>
    <!--ro, opt, int, 端口号-->1
  </portNo>
  <userName>
    <!--ro, opt, string, 用户名, dep:and,{$.HttpHostNotification.httpAuthenticationMethod,eq,MD5digest}-->test
  </userName>
  <httpAuthenticationMethod>
```

```
<!--ro, req, enum, 认证方法, subType:string, [MD5digest#MD5,none#无,base64#base64]-->MD5digest
</httpAuthenticationMethod>
<ANPR>
<!--ro, opt, object, ANPR-->
<!--ro, opt, enum, 侦测上传图片类型控制, subType:string, [all#所有,LicensePlatePicture#车牌图片,detectionPicture#检测图片]-->all
</detectionUploadPicturesType>
</ANPR>
<Extensions>
<!--ro, opt, object, 范围-->
<intervalBetweenEvents>
<!--ro, opt, int, 事件间隔时间-->1
</intervalBetweenEvents>
</Extensions>
<uploadImagesDataType>
<!--ro, opt, enum, 图片数据类型, subType:string, [URL#URL,binary#二进制], desc:默认是二进制,在支持了云存储之后,配置成URL,才会生效-->URL
</uploadImagesDataType>
<httpBroken>
<!--ro, opt, bool, 断网续传使能, desc:开启后实现监听的断网续传功能,对所有事件有效-->true
</httpBroken>
<SubscribeEvent>
<!--ro, opt, object, 该节点表示所有含图片事件图片上传方式-->
<heartbeat>
<!--ro, opt, int, 心跳间隔时间-->30
</heartbeat>
<eventMode>
<!--ro, req, enum, 事件模式, subType:string, [all#表示所有报警均需要上报,list#表示指定报警才需要上报]-->all
</eventMode>
<eventList>
<!--ro, opt, array, 指定报警上报的方式, subType:object-->
<Event>
<!--ro, opt, object, 事件关联的通道信息-->
<type>
<!--ro, req, enum, 事件类型, subType:string, [ADAS#高级辅助驾驶报警,ADASAlarm#高级驾驶辅助报警,AID#交通事件检测,ANPR#车牌识别,AccessControlerEvent#门禁主机检测事件,CDsStatus#光盘刻录状态上报,DBD#驾驶行为报警,GPSupLoad#GPS信息上传,HFPD#高频人员报警,IO#信号量报警,IOTD#非视频设备检测事件,LES#物流扫描事件,LFPD#低频人员检测,PALMismatch#视频制式不匹配,PIR#PIR,PeopLeCounting#客流量统计,PeopLeNumChange#人数异常,Standup#人员站立检测,TMA#测温报警,TMPA#测温预警,VMD#移动侦测,abnormalACceleration#设备加速度异常报警,abnormalDriving#异常驾驶行为,advReachHeight#折线攀高检测,alarmResult#人脸比对,attendance#考勤事件,attendedBaggage#物品拿取,audioAbnormal#声强突变检测,audioException#音频异常侦测,behaviorResult#行为分析结果上报,blindSpotDetection#盲区监测报警,cardMatch#认证比对,changedStatus#状态变化上报,collision#碰撞,containerDetection#集装箱检测,crowdSituationAnalysis#人群态势分析,databaseException#数据库异常,defocus#虚焦侦测,diskUnformat#磁盘未格式化,diskerror#硬盘错误,diskfull#硬盘满,driverConditionMonitor#驾驶员状态监测报警,emergencyAlarm#紧急报警,faceCapture#人脸抓拍识别,faceSnapModeling#人脸抓拍建模,facedetection#人脸侦测,failDown#人员倒地(使用错误的拼写,faultAlarm#故障告警,fieldDetection#区域入侵,firedetection#火点检测,fireEscapeDetection#消防通道检测,fLowOverrun#流量超限,framesPeopleCounting#区域人数统计,getUp#起身检测,group#人员聚集,hdBadBlock#硬盘坏块,hdImpact#硬盘冲击,heatmap#热度图报警,highDTemperature#硬盘温度过高,highTempAlarm#设备温度过高报警,hotSpare#热备异常,iLLAccess#非法访问,ipcTransferAbnormal#IPC迁移失败,ipconflict#IP地址冲突,keyPersonGetUp#重点人员起身检测,LeavePosition#离岗检测,linedetection#越界侦测(穿越警戒线),ListSyncException#列表(名单)同步异常,Loitering#徘徊侦测,LowDTemperature#硬盘温度过低,mixedTargetDetection#混合目标检测,modelError#模型错误,nicbroken#网络断开,nodeOffLine#掉线异常,nonPoliceIntrusion#非警察人员入侵,overSpeed#超速报警,overTimeTarry#超时滞留检测,parking#停车侦测,peopLeNumChange#人数变化,peopleNumCounting#人数统计(区别于区域人数统计),personAbnormalAlarm#人员身份异常报警,personDensityDetection#人员密度,personQueueCounting#排队人数检测,personQueueDetection#排队检测,personQueueRealTime#人员排队检测实时数据上报,personQueueTime#排队时长检测,playCellPhone#玩手机检测,pocException#视频异常,poe#POE供电异常,policeAbsent#警察不在场,radarAlarm#雷达告警,radarFieldDetection#雷达区域入侵,radarLineDetection#雷达越界侦测,radarPerimeterRule#雷达规则数据,radarTargetDetection#雷达目标数据,radarVideoDetection#雷达视频目标检测,raidException#阵列异常,rapidMove#快速移动,reachHeight#攀高检测,recordCycleAbnormal#录像周期不足,recordException#录像异常,regionEntrance#进入区域,regionExiting#离开区域,retention#人员滞留事件,rollOver#侧翻,running#人员奔跑,safetyHelmetDetection#安全帽检测,scenechangedetection#场景变更,sensorAlarm#角速度加速度报警,severeHDFailure#硬盘严重故障,shelterAlarm#遮挡报警,shipsDetection#船只检测,sitQuietly#静坐检测,smokeAndFireDetection#烟火检测,smokeDetection#烟雾检测,softIO#报警软IO,spacingChange#间距异常,sysStorFull#集群系统存储满异常,takingElevatorDetection#乘梯检测(电瓶车进电梯检测),targetCapture#目标抓拍,temperature#温度报警,thermometry#温度报警,thirdPartyException#第三方异常,toiletTarry#如厕超时检测,tollCodeInfo#二维码信息上报,tossing#抛物,unattendedBaggage#物品遗留,vehicleMatchResult#黑白名单车辆报警上传,vehicleRcogResult#车辆图片二次识别,versionAbnormal#集群版本异常,videoException#视频异常,videoLoss#视频丢失,violationAlarm#违规告警,violentMotion#剧烈运动,yardTarry#放风场滞留检测,AccessControlerEvent#门禁事件,IDCardInfoEvent#门禁身份证事件,FaceTemperatureMeasurementEvent#门禁测温事件,QRCodeEvent#门禁二维码事件,CertificateCaptureEvent#证件抓拍比对事件,UnCertificateCompareEvent#无证比对事件,ConsumptionAndTransactionRecordEvent#消费和交易记录事件,ConsumptionEvent#消费事件,TransactionRecordEvent#交易记录事件,SetMealQuery#消费套餐查询,ConsumptionStatusQuery#消费状态查询,humanBodyComparison#人体比对,regionTargetNumberCounting#区域目标统计,paymentQRCodeQuery#查询第三方支付二维码,ConsumptionQuery#本地消费查询事件,topUpRequest#充值请求,topUpStatusQuery#查询充值状态,topUpEvent#充值事件,LogOffRefundStatusQuery#查询人员注销退款状态,LogOffRefundRequest#人员注销退款请求], desc:参考事件类型列表eventType-->mixedTargetDetection
</type>
<minorAlarm>
<!--ro, opt, string, 报警次类型, desc:参考事件上传宏定义,type为AccessControlerEvent, IDCardInfoEvent 该字段必填-->0x400,0x401,0x402,0x403
</minorAlarm>
<minorException>
<!--ro, opt, string, 异常次类型, desc:参考事件上传宏定义,type为AccessControlerEvent, IDCardInfoEvent 该字段必填-->0x400,0x401,0x402,0x403
</minorException>
<minorOperation>
<!--ro, opt, string, 操作次类型, desc:参考事件上传宏定义,type为AccessControlerEvent, IDCardInfoEvent 该字段必填-->0x400,0x401,0x402,0x403
</minorOperation>
<minorEvent>
<!--ro, opt, string, 事件次类型, desc:参考事件上传宏定义,type为AccessControlerEvent, IDCardInfoEvent 该字段必填-->0x01,0x02,0x03,0x04
</minorEvent>
<pictureURLType>
<!--ro, opt, enum, 指定事件的报警图片格式, subType:string, [binary#二进制,LocalURL#设备本地URL,cloudStorageURL#云存储URL,EZVIZURL#萤石URL]-->binary
</pictureURLType>
<channels>
<!--ro, opt, string, 监听指定通道号列表的事件, desc:如果监听全部通道,则不下发这个节点:监听部分通道时,列出通道号,用逗号隔开-->1,2,3,4
</channels>
</Event>
</EventList>
<channels>
<!--ro, opt, string, 监听指定通道号列表, desc:如果监听全部通道,则不下发这个节点:监听部分通道时,列出通道号,用逗号隔开-->1,2,3,4
</channels>
<pictureURLType>
<!--ro, opt, enum, 报警图片格式, subType:string, [binary#二进制,LocalURL#设备本地URL,cloudStorageURL#云存储URL,EZVIZURL#萤石URL], desc:该节点表示所有含图片事件图片上传方式,如果该字段下发,那么<Event>中<pictureURLType>不做生效:如果不下发这个节点,设备采用能力集返回的图片默认上传方式上报,前端设备默认为二进制,后端设备默认为设备本地URL-->binary
</pictureURLType>
<ChangedUploadSub>
```

```

<!--ro, opt, object, 订阅报文-->
<interval>
  <!--ro, opt, int, 布防guid的生命周期, desc:布防guid的生命周期,该间隔期间未过来重连,设备认为是新的一路布防,重新生成guid-->5
</interval>
<StatusSub>
  <!--ro, opt, object, 子状态-->
  <all>
    <!--ro, opt, bool, 是否订阅所有-->true
  </all>
  <channel>
    <!--ro, opt, bool, 订阅通道状态, desc:all为true时不必传-->true
  </channel>
  <hd>
    <!--ro, opt, bool, 订阅磁盘状态, desc:all为true时不必传-->true
  </hd>
  <capability>
    <!--ro, opt, bool, 订阅能力集变化状态, desc:all为true时不必传-->true
  </capability>
</StatusSub>
</ChangedUploadSub>
</SubscribeEvent>
<PackingSpaceRecognition>
  <!--ro, opt, object, 当前报警主机下车位检测监听事件控制参数, desc:对应事件为PackingSpaceRecognition-->
  <uploadPicturesType>
    <!--ro, req, enum, 上传图片类型控制, subType:string, [all#所有,picturesTypes#指定图片类型上传,notUpload#不上传图片]-->all
  </uploadPicturesType>
  <PicturesTypes>
    <!--ro, opt, array, 指定图片类型上传列表, subType:object, dep:and,
    {$.HttpHostNotification.PackingSpaceRecognition.uploadPicturesType,eq,picturesTypes}-->
  <picturesType>
    <!--ro, opt, enum, 上传图片类型, subType:string, [backgroundImage#背景大图,plateImage#车牌小图]-->backgroundImage
  </picturesType>
</PicturesTypes>
</PackingSpaceRecognition>
</HttpHostNotification>

```

69.15.8 配置全部监听主机参数

Request URL

PUT /ISAPI/Event/notification/httpHosts?security=<security>&iv=<iv>

查询参数

参数名称	参数类型	描述
security	string	security=1,表示报文中的userName和密码字段AES128加密 security=2,表示报文中的userName和密码字段AES256加密
iv	string	iv表示初始化向量，在security=1或者2时为必须参数;

请求报文

```

<?xml version="1.0" encoding="UTF-8"?>

<HttpHostNotificationList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, array, 监听主机列表, subType:object, attr:version{req, string, 协议版本}-->
  <HttpHostNotification>
    <!--opt, object, 监听主机-->
    <id>
      <!--req, string, id-->test
    </id>
    <url>
      <!--req, string, URL-->test
    </url>
    <protocolType>
      <!--req, enum, 协议类型, subType:string, [HTTP#HTTP协议,HTTPS#HTTPS协议,EHome#EHome协议]-->HTTP
    </protocolType>
    <parameterFormatType>
      <!--req, enum, 参数格式类型, subType:string, [JSON#JSON格式,XML#XML格式]-->JSON
    </parameterFormatType>
    <addressingFormatType>
      <!--req, enum, 地址类型, subType:string, [hostname#hostname,ipaddress#ipaddress]-->hostname
    </addressingFormatType>
    <hostName>
      <!--opt, string, 主机名-->test
    </hostName>
    <ipAddress>
      <!--opt, string, IP地址-->test
    </ipAddress>
    <ipv6Address>
      <!--opt, string, ipv6地址-->test
    </ipv6Address>
    <portNo>

```

```
<!--opt, int, 端口号-->1
</portNo>
<userName>
<!--opt, string, 用户名, dep:and,{$.HttpHostNotification.httpAuthenticationMethod,eq,MD5digest}-->test
</userName>
<password>
<!--opt, string, 密码, dep:and,{$.HttpHostNotification.httpAuthenticationMethod,eq,MD5digest}-->test
</password>
<httpAuthenticationMethod>
<!--req, enum, 认证方法, subType:string, [MD5digest#MD5,none#无,base64#base64]-->MD5digest
</httpAuthenticationMethod>
<ANPR>
<!--opt, object, ANPR-->
<uploadPicturesType>
<!--opt, enum, 侦测上传图片类型控制, subType:string, [all#所有,LicensePlatePicture#车牌图片,detectionPicture#检测图片]-->all
</uploadPicturesType>
</ANPR>
<Extensions>
<!--opt, object, 范围-->
<intervalBetweenEvents>
<!--opt, int, 事件间隔时间-->1
</intervalBetweenEvents>
</Extensions>
<uploadImagesDataType>
<!--opt, enum, 图片数据类型, subType:string, [URL#URL,binary#二进制], desc:默认是二进制,在支持了云存储之后,配置成URL,才会生效-->URL
</uploadImagesDataType>
<httpBroken>
<!--opt, bool, 断网续传使能, desc:开启后实现监听的断网续传功能,对所有事件有效-->true
</httpBroken>
<SubscribeEvent>
<!--opt, object, 该节点表示所有含图片事件图片上传方式-->
<heartbeat>
<!--opt, int, 心跳间隔时间-->30
</heartbeat>
<eventMode>
<!--req, enum, 事件模式, subType:string, [all#表示所有报警均需要上报,List#表示指定报警才需要上报]-->all
</eventMode>
<EventList>
<!--opt, array, 指定报警上报的方式, subType:object-->
<Event>
<!--opt, object, 事件关联的通道信息-->
<type>
<!--req, enum, 事件类型, subType:string, [ADAS#高级辅助驾驶报警,ADASAlarm#高级驾驶辅助报警,AID#交通事件检测,ANPR#车牌识别,AccessControlLerEvent#门禁主机检测事件,CDSStatus#车道偏离状态上报,DBD#驾驶行为报警,GPSUpload#GPS信息上传,HFPD#高频人员报警,IO#信号量报警,IOTD#非视频设备检测事件,LES#物流扫描事件,LFPD#低频人员检测,PALMismatch#视频帧式不匹配,PIR#PIR,PeopleCounting#客流量统计,PeopleNumChange#人数异常,Standup#人员站立检测,TMA#测温报警,TMPA#测温预警,VM#移动侦测,abnormalAcceleration#设备加速度异常报警,abnormalDriving#异常驾驶行为,advReachHeight#折线攀高检测,alarmResult#人脸比对,attendance#考勤事件,attendedBaggage#物品拿取,audioAbnormal#声强突变检测,audioException#音频异常侦测,behaviorResult#行为分析结果上报,blindSpotDetection#盲区监测报警,cardMatch#认证比对,changedStatus#状态变化上报,collision#碰撞,containerDetection#集装箱检测,crowdSituationAnalysis#人群态势分析,databaseException#数据库异常,defocus#虚焦侦测,diskUnformat#磁盘未格式化,diskError#硬盘错误,diskFull#硬盘满,driverConditionMonitor#驾驶员状态监测报警,emergencyAlarm#紧急报警,faceCapture#人脸抓拍识别,faceSnapModeling#人脸抓拍建模,facedetection#人脸侦测,failDown#人员倒地(使用错误的拼写,faultAlarm#故障告警,fieldDetection#区域入侵,fireDetection#火点检测,fireEscapeDetection#消防通道检测,flowOverrun#流量超限,framesPeopleCounting#区域人数统计,getUp#起身检测,group#人员聚集,hdBadBlock#硬盘坏块,hdImpact#硬盘冲击,heatmap#热度图报警,highHDTemperature#硬盘温度过高,highTempAlarm#设备温度过高报警,hotSpare#热备异常,ilLAccess#非法访问,ipcTransferAbnormal#IPC迁移失败,ipConflict#IP地址冲突,keyPersonGetUp#重点人员起身检测,LeavePosition#离岗检测,Linedetection#越界侦测(穿越警戒线),ListSyncException#列表(名单)同步异常,Loitering#徘徊侦测,LowHDTemperature#硬盘温度过低,mixedTargetDetection#混合目标检测,modelError#模型错误,nicBroken#网络断开,nodeOffline#掉线异常,nonPoliceIntrusion#非警察人员入侵,overSpeed#超速报警,overtimeTarry#超时滞留检测,parking#停车侦测,peopleNumChange#人数变化,peopleNumCounting#人数统计(区别于区域人数统计),personAbnormalAlarm#人员身份异常报警,personDensityDetection#人员密度,personQueueCounting#排队人数检测,personQueueDetection#排队检测,personQueueRealTime#人员排队检测实时数据上报,personQueueTime#排队时长检测,pLAYCellPhone#玩手机检测,pocException#视频异常,poe#POE供电异常,policeAbsent#警察不在场,radarAlarm#雷达告警,radarFieldDetection#雷达区域入侵,radarLineDetection#雷达越界侦测,radarPerimeterRule#雷达规则数据,radarTargetDetection#雷达目标数据,radarVideoDetection#雷达目标检测,raidException#阵列异常,rapidMove#快速移动,reachHeight#攀高检测,recordCycleAbnormal#录像周期不足,recordException#录像异常,regionEntrance#进入区域,regionExiting#离开区域,retention#人员滞留事件,rollOver#侧翻,running#人员奔跑,safetyHelmetDetection#安全帽检测,scenechangedetection#场景变更,sensorAlarm#角速度加速度报警,severeHDFailure#硬盘严重故障,shelLateralArm#遮挡报警,shipsDetection#船只检测,sitQuietly#静坐检测,smokeAndFireDetection#烟火检测,smokeDetection#烟雾检测,softIO#报警软IO,spacingChange#间距异常,sysStarFull#集群系统存储满异常,takingElevatorDetection#乘梯检测(电瓶车进电梯检测),targetCapture#目标抓拍,temperature#温差报警,thermometry#温度报警,thirdPartyException#第三方异常,toiletTarry#厕所超时检测,tollCodeInfo#二维码信息上报,tossing#抛物,unattendedBaggage#物品遗留,vehicleMatchResult#黑白名单车辆报警上传,vehicleRecogResult#车辆图片二次识别,versionAbnormal#集群版本异常,videoException#视频异常,videoLoss#视频丢失,violationAlarm#违规告警,violentMotion#剧烈运动,yardTarry#放风场滞留检测,AccessControlLerEvent#门禁事件,IDCardInfoEvent#门禁身份证事件,FaceTemperatureMeasurementEvent#门禁测温事件,QRCodeEvent#门禁二维码事件,CertificateConsumptionEvent#证件抓拍比对事件,UncertificateCompareEvent#无证证件事件,ConsumptionAndTransactionRecordEvent#消费和交易记录事件,ConsumptionEvent#消费事件,TransactionRecordEvent#交易记录事件,SetMealQuery#消费套餐查询,ConsumptionStatusQuery#消费状态查询,humanBodyComparison#人体比对,regionTargetNumberCounting#区域目标统计,paymentQRCodeQuery#查询第三方支付二维码,ConsumptionQuery#本地消费查询事件,topUpRequest#充值请求,topUpStatusQuery#查询充值状态,topUpEvent#充值事件,LogOffRefundStatusQuery#查询人员注销退款状态,LogOffRefundRequest#人员注销退款请求], desc:参考事件类型列表eventType-->mixedTargetDetection
</type>
<minorAlarm>
<!--opt, string, 报警次类型, desc:参考事件上传宏定义;type为AccessControlLerEvent, IDCardInfoEvent该字段必填-->0x400,0x401,0x402,0x403
</minorAlarm>
<minorException>
<!--opt, string, 异常次类型, desc:参考事件上传宏定义;type为AccessControlLerEvent, IDCardInfoEvent该字段必填-->0x400,0x401,0x402,0x403
</minorException>
<minorOperation>
<!--opt, string, 操作次类型, desc:参考事件上传宏定义;type为AccessControlLerEvent, IDCardInfoEvent该字段必填-->0x400,0x401,0x402,0x403
</minorOperation>
<minorEvent>
<!--opt, string, 事件次类型, desc:参考事件上传宏定义;type为AccessControlLerEvent, IDCardInfoEvent该字段必填-->0x01,0x02,0x03,0x04
</minorEvent>
<pictureURLType>
<!--opt, enum, 指定事件的报警图片格式, subType:string, [binary#二进制,localURL#设备本地URL,cLOUDStorageURL#云存储URL,EZVIZURL#萤石URL]-->binary
</pictureURLType>
<channels>
<!--opt, string, 监听指定通道号列表的事件, desc:如果监听全部通道,则不下发这个节点;监听部分通道时,列出通道号,用逗号隔开-->1,2,3,4
</channels>
</Event>
</EventList>
<channels>
```

```

<!--opt, string, 监听指定通道号列表, desc:如果监听全部通道, 则不下发这个节点; 监听部分通道时, 列出通道号, 用逗号隔开-->1,2,3,4
</channels>
<pictureURLType>
  <!--opt, enum, 报警图片格式, subType:string, [binary#二进制, LocalURL#设备本地URL, cloudStorageURL#云存储URL, EZVIZURL#萤石URL], desc:该节点表示所有含
  图片事件图片上传方式, 如果该字段下发, 那么<Event>中<pictureURLType>不做生效; 如果不下发这个节点, 设备采用能力集返回的图片默认上传方式上报, 前端设备默认为二进制,
  后端设备默认为设备本地URL-->binary
</pictureURLType>
<ChangedUploadSub>
  <!--opt, object, 订阅报文-->
  <interval>
    <!--opt, int, 布防guid的生命周期, desc:布防guid的生命周期, 该间隔期间未过来重连, 设备认为是新的一路布防, 重新生成guid-->5
  </interval>
  <StatusSub>
    <!--opt, object, 子状态-->
    <all>
      <!--opt, bool, 是否订阅所有-->true
    </all>
    <channel>
      <!--opt, bool, 订阅通道状态, desc:all为true时不必传-->true
    </channel>
    <hd>
      <!--opt, bool, 订阅磁盘状态, desc:all为true时不必传-->true
    </hd>
    <capability>
      <!--opt, bool, 订阅能力集变化状态, desc:all为true时不必传-->true
    </capability>
  </StatusSub>
</ChangedUploadSub>
</SubscribeEvent>
<PackingSpaceRecognition>
  <!--opt, object, 当前报警主机下车位检测监听事件控制参数, desc:对应事件为PackingSpaceRecognition-->
  <uploadPictureType>
    <!--req, enum, 上传图片类型控制, subType:string, [all#所有, picturesTypes#指定图片类型上传, notUpload#不上上传图片]-->all
  </uploadPictureType>
  <PicturesTypes>
    <!--opt, array, 指定图片类型上传列表, subType:object, dep:and,
    {$.HttpHostNotificationList[*].HttpHostNotification.PackingSpaceRecognition.uploadPictureType,eq,picturesTypes}-->
  <picturesType>
    <!--opt, enum, 上传图片类型, subType:string, [backgroundImage#背景大图, plateImage#车牌小图]-->backgroundImage
  </picturesType>
</PicturesTypes>
</PackingSpaceRecognition>
</HttpHostNotification>
</HttpHostNotificationList>

```

响应报文

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 响应消息, attr:version{ro, req, string, 版本}-->
  <requestURL>
    <!--ro, req, string, 请求的URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, 状态码, subType:int, [0#OK,1#OK,2#Device Busy,3#Device Error,4#Invalid Operation,5#Invalid XML Format,6#Invalid XML Content,7#Reboot
    Required]-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, 状态信息, subType:string, [OK#成功,Device Busy#设备忙,Device Error#设备异常,Invalid Operation#无效的操作,Invalid XML Format#无效的XML
    格式,Invalid XML Content#无效的XML内容,Reboot#设备重启]-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, 详细错误码英文描述, desc:详细错误码的英文描述-->OK
  </subStatusCode>
</ResponseStatus>

```

69.15.9 获取全部监听主机参数

Request URL

GET /ISAPI/Event/notification/httpHosts?security=<security>&iv=<iv>

查询参数

参数名称	参数类型	描述
security	string	security=1,表示报文中的userName和password字段AES128加密 security=2,表示报文中的userName和password字段AES256加密
iv	string	iv表示初始化向量, 在security=1或者2时为必须参数;

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<HttpHostNotificationList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, array, 监听主机列表, subType:object, attr:version{req, string, 协议版本}-->
  <HttpHostNotification>
    <!--ro, opt, object, 监听主机-->
    <id>
      <!--ro, req, string, 订阅ID, range:[1,10]-->test
    </id>
    <url>
      <!--ro, req, string, URL-->test
    </url>
    <protocolType>
      <!--ro, req, enum, 协议类型, subType:string, [HTTP#HTTP协议,HTTPS#HTTPS协议,EHome#EHome 协议]-->HTTP
    </protocolType>
    <parameterFormatType>
      <!--ro, req, enum, 参数格式类型, subType:string, [JSON#JSON格式,XML#XML 格式]-->JSON
    </parameterFormatType>
    <addressingFormatType>
      <!--ro, req, enum, 地址类型, subType:string, [hostname#hostname,ipaddress#ipaddress]-->hostname
    </addressingFormatType>
    <hostName>
      <!--ro, opt, string, 主机名-->test
    </hostName>
    <ipAddress>
      <!--ro, opt, string, IP地址-->test
    </ipAddress>
    <ipv6Address>
      <!--ro, opt, string, ipv6地址-->test
    </ipv6Address>
    <portNo>
      <!--ro, opt, int, 端口号-->1
    </portNo>
    <userName>
      <!--ro, opt, string, 用户名, dep:and,{$.HttpHostNotification.httpAuthenticationMethod,eq,MD5digest}-->test
    </userName>
    <httpAuthenticationMethod>
      <!--ro, req, enum, 认证方法, subType:string, [MD5digest#MD5,none#无,base64#base64]-->MD5digest
    </httpAuthenticationMethod>
    <ANPR>
      <!--ro, opt, object, ANPR-->
      <uploadPicturesType>
        <!--ro, opt, enum, 侦测上传图片类型控制, subType:string, [all#所有,licensePlatePicture#车牌图片,detectionPicture#检测图片]-->all
      </uploadPicturesType>
    </ANPR>
    <Extensions>
      <!--ro, opt, object, 范围-->
      <intervalBetweenEvents>
        <!--ro, opt, int, 事件间隔时间-->1
      </intervalBetweenEvents>
    </Extensions>
    <uploadImagesDataType>
      <!--ro, opt, enum, 图片数据类型, subType:string, [URL#URL,binary#二进制], desc:默认是二进制,在支持了云存储之后,配置成URL,才会生效-->URL
    </uploadImagesDataType>
    <httpBroken>
      <!--ro, opt, bool, 断网续传使能, desc:开启后实现监听的断网续传功能,对所有事件有效-->true
    </httpBroken>
    <SubscribeEvent>
      <!--ro, opt, object, 该节点表示所有含图片事件图片上传方式-->
      <heartbeat>
        <!--ro, opt, int, 心跳间隔时间-->30
      </heartbeat>
      <eventMode>
        <!--ro, req, enum, 事件模式, subType:string, [all#表示所有报警均需要上报,List#表示指定报警才需要上报]-->all
      </eventMode>
      <EventList>
        <!--ro, opt, array, 指定报警上报的方式, subType:object-->
        <Event>
          <!--ro, opt, object, 事件关联的通道信息-->
          <type>
            <!--ro, req, enum, 事件类型, subType:string, [ADAS#高级辅助驾驶报警,ADASALarm#高级驾驶辅助报警,AID#交通事件检测,ANPR#车牌识别,AccessControlLerEvent#门禁主机检测事件,CdsStatus#光盘刻录状态上报,DBD#驾驶行为报警,GPSUpLoad#GPS信息上传,HFPD#高频人员报警,IO#信号量报警,IOTD#非视频设备检测事件,LES#物流扫描事件,LFPD#低频人员检测,PALMismatch#视频制式不匹配,PIR#PIR,PeopleCounting#客流量统计,PeopleNumChange#人数异常,Standup#人员站立检测,TMA#测温报警,TMPA#测温预警,VMD#移动侦测,abnormalAcceleration#设备加速度异常报警,abnormalDriving#异常驾驶行为,advReachHeight#折线攀高检测,alarmResult#人脸比对,attendance#考勤事件,attendedBaggage#物品拿取,audioAbnormal#声强突变检测,audioException#音频异常侦测,behaviorResult#行为分析结果上报,blindSpotDetection#盲区监测报警,cardMatch#认证比对,changedStatus#状态变化上报,collision#碰撞,containerDetection#集装箱检测,crowdSituationAnalysis#人群态势分析,databaseException#数据库异常,defocus#虚焦侦测,diskUnformat#磁盘未格式化,diskerror#硬盘错误,diskfull#硬盘满,driverConditionMonitor#驾驶员状态监测报警,emergencyAlarm#紧急报警,faceCapture#人脸抓拍识别,faceSnapModeling#人脸抓拍建模,facedetection#人脸侦测,failDown#人员倒地（使用错误的拼写,faultAlarm#故障告警,fieldDetection#区域入侵,fireDetection#火点检测,fireEscapeDetection#消防通道检测,flowOverrun#流量超限,framesPeopleCounting#区域人数统计,getUp#起身检测,group#人员聚集,hdBadBlock#硬盘坏块,hdImpact#硬盘冲击,heatmap#热度图报警,highHDTemperature#硬盘温度过高,highTempAlarm#设备温度过高报警,hotSpare#热备异常,iLlAccess#非法访问,ipcTransferAbnormal#IPC迁移失败,ipConflict#IP地址冲突,keyPersonGetUp#重点人员起身检测,LeavePosition#离岗检测,Linedetection#越界侦测（穿越警戒线）,ListSyncException#列表（名单）同步异常,Loitering#徘徊侦测,LowHDTemperature#硬盘温度过低,mixedTargetDetection#混合目标检测,modelError#模型错误,nicbroken#网络断开,nodeOffline#掉线异常,nonPoliceIntrusion#非警察人员入侵,overSpeed#超速报警,overTimeTarry#超时滞留检测,parking#停车侦测,peopleNumChange#人数变化,peopleNumCounting#人数统计（区别于区域人数统计）,personAbnormalALarm#人员身份异常报警,personDensityDetection#人员密度,personQueueCounting#排队人数检测,personQueueDetection#排队检测,personQueueRealTime#人员排队检测实时数据上报,personQueueTime#排队时长检测,playCellPhone#玩手机检测,pocException#视频异常,radarBDR#雷达异常,radarBlindSpot#警戒不在场,radarAlarm#雷达告警,radarFieldDetection#雷达入侵入侵,radarLineDetection#雷达越界侦测,radarPerimeterRule#雷达规则
```

```
数据, radarTargetDetection#雷达目标数据, radarVideoDetection#雷视目标检测, raidException#阵列异常, rapidMove#快速移动, reachHeight#攀高检测, recordCycleAbnormal#
录像周期不足, recordException#录像异常, regionEntrance#进入区域, regionExiting#离开区域, retention#人员滞留事件, rollover#侧翻, running#人员奔
跑, safetyHelmetDetection#安全帽检测, scenechangedetection#场景变更, sensorAlarm#角速度加速度报警, severeHdFailure#硬盘严重故障, shelterAlarm#遮挡报
警, shipsDetection#船只检测, sitQuietly#静坐检测, smokeAndFireDetection#烟火检测, smokeDetection#烟雾检测, softIO#报警软IO, spacingChange#间距异常, sysStorFull#集
群系统存储满异常, takingElevatorDetection#乘梯检测(电瓶车进电梯检测), targetCapture#目标抓拍, temperature#温差报警, thermometry#温度报警, thirdPartyException#
第三方异常, toiletTarry#如厕超时检测, tollCodeInfo#二维码信息上报, tossing#抛物, unattendedBaggage#物品遗留, vehicleMatchResult#黑白名单车辆报警上
传, vehicleRcogResult#车辆图片二次识别, versionAbnormal#集群版本异常, videoException#视频异常, videoLoss#视频丢失, violationAlarm#违规警告, violentMotion#剧烈运
动, yardTarry#放风场滞留检测, AccessControllerEvent#门禁事件, IDCardInfoEvent#门禁身份证事件, FaceTemperatureMeasurementEvent#门禁测温事件, QRCodeEvent#门禁二维
码事件, CertificateCaptureEvent#证件抓拍比对事件, UncertificateCompareEvent#无证比对事件, ConsumptionAndTransactionRecordEvent#消费和交易记录事
件, ConsumptionEvent#消费事件, TransactionRecordEvent#交易记录事件, SetMealQuery#消费套餐查询, ConsumptionStatusQuery#消费状态查询, humanBodyComparison#人体比
对, regionTargetNumberCounting#区域目标统计, paymentQRCodeQuery#查询三方支付二维码, ConsumptionQuery#本地消费查询事件, topUpRequest#充值请求, topUpStatusQuery#查
询充值状态, topUpEvent#充值事件, LogOffRefundStatusQuery#查询人员注销退款状态, LogOffRefundRequest#人员注销退款请求], desc:参考事件类型列表eventType--
>mixedTargetDetection
</type>
<minorAlarm>
  <!--ro, opt, string, 报警次类型, desc:参考事件上传宏定义;type为AccessControllerEvent, IDCardInfoEvent该字段必填-->0x400,0x401,0x402,0x403
</minorAlarm>
<minorException>
  <!--ro, opt, string, 异常次类型, desc:参考事件上传宏定义;type为AccessControllerEvent, IDCardInfoEvent该字段必填-->0x400,0x401,0x402,0x403
</minorException>
<minorOperation>
  <!--ro, opt, string, 操作次类型, desc:参考事件上传宏定义;type为AccessControllerEvent, IDCardInfoEvent该字段必填-->0x400,0x401,0x402,0x403
</minorOperation>
<minorEvent>
  <!--ro, opt, string, 事件次类型, desc:参考事件上传宏定义;type为AccessControllerEvent, IDCardInfoEvent该字段必填-->0x01,0x02,0x03,0x04
</minorEvent>
<pictureURLType>
  <!--ro, opt, enum, 指定事件的报警图片格式, subType:string, [binary#二进制, LocalURL#设备本地URL, cloudStorageURL#云存储URL, EZVIZURL#萤石URL]--
>binary
</pictureURLType>
<channels>
  <!--ro, opt, string, 监听指定通道号列表的事件, desc:如果监听全部通道, 则不下发这个节点; 监听部分通道时, 列出通道号, 用逗号隔开-->1,2,3,4
</channels>
</Event>
</EventList>
<channels>
  <!--ro, opt, string, 监听指定通道号列表, desc:如果监听全部通道, 则不下发这个节点; 监听部分通道时, 列出通道号, 用逗号隔开-->1,2,3,4
</channels>
<pictureURLType>
  <!--ro, opt, enum, 报警图片格式, subType:string, [binary#二进制, LocalURL#设备本地URL, cloudStorageURL#云存储URL, EZVIZURL#萤石URL], desc:该节点表示所
有含图片事件图片上传方式, 如果该字段下发, 那么<Event>中<pictureURLType>不做生效; 如果不下发这个节点, 设备采用能力集返回的图片默认上传方式上报, 前端设备默认为二
进制, 后端设备默认为设备本地URL-->binary
</pictureURLType>
<ChangedUploadSub>
  <!--ro, opt, object, 订阅博文-->
<interval>
  <!--ro, opt, int, 布防guid的生命周期, desc:布防guid的生命周期, 该间隔期间未过来重连, 设备认为是新的一路布防, 重新生成guid-->5
</interval>
<StatusSub>
  <!--ro, opt, object, 子状态-->
<all>
  <!--ro, opt, bool, 是否订阅所有-->true
</all>
<channel>
  <!--ro, opt, bool, 订阅通道状态, desc:all为true时不必传-->true
</channel>
<hd>
  <!--ro, opt, bool, 订阅磁盘状态, desc:all为true时不必传-->true
</hd>
<capability>
  <!--ro, opt, bool, 订阅能力集变化状态, desc:all为true时不必传-->true
</capability>
</StatusSub>
</ChangedUploadSub>
</SubscribeEvent>
<PackingSpaceRecognition>
  <!--ro, req, object, 当前报警主机下车位检测监听事件控制参数, desc:对应事件为PackingSpaceRecognition-->
<uploadPicturesType>
  <!--ro, opt, enum, 上传图片类型控制, subType:string, [all#所有, picturesTypes#指定图片类型上传, notUpload#不上上传图片]-->all
</uploadPicturesType>
<PicturesTypes>
  <!--ro, opt, array, 指定图片类型上传列表, subType:object, dep:and,
{$.HttpHostNotificationList[*].HttpHostNotification.PackingSpaceRecognition.uploadPicturesType,eq,picturesTypes}-->
<picturesType>
  <!--ro, opt, enum, 上传图片类型, subType:string, [backgroundImage#背景大图, pLateImage#车牌小图]-->backgroundImage
</picturesType>
</PicturesTypes>
</PackingSpaceRecognition>
</HttpHostNotification>
</HttpHostNotificationList>
```

69.15.10 获取监听主机参数能力

Request URL

GET /ISAPI/Event/notification/httpHosts/capabilities

查询参数

无

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<HttpHostNotificationCap xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 事件类型订阅和事件通道订阅的能力节点, attr:version{req, string, 协议版本}-->
  <hostNumber>
    <!--ro, req, int, 支持的监听主机个数-->0
  </hostNumber>
  <urlLen max="10">
    <!--ro, req, string, url长度, attr:max{req, int, 最大值}-->test
  </urlLen>
  <url>
    <!--ro, opt, string, url-->test
  </url>
  <protocolType opt="HTTP,HTTPS,EHome">
    <!--ro, req, enum, 协议类型, subType:string, [HTTP#HTTP,HTTPS#HTTPS,EHome#EHome], attr:opt{req, string, 取值范围}-->HTTP
  </protocolType>
  <parameterFormatType opt="XML,querystring,JSON">
    <!--ro, req, enum, 报警上报的参数类型 (该字段无实际作用, subType:string, [XML#XML格式,querystring#querystring,JSON#JSON格式], attr:opt{req, string, 取值范围}-->XML
  </parameterFormatType>
  <addressingFormatType opt="ipaddress,hostname">
    <!--ro, req, enum, 地址格式类型, subType:string, [ipaddress#ip地址,hostname#域名], attr:opt{req, string, 取值范围}-->ipaddress
  </addressingFormatType>
  <hostName min="1" max="64">
    <!--ro, opt, string, 域名, attr:min{req, int, 最小值},max{req, int, 最大值}, desc:当addressingFormatType为hostname时有效-->test
  </hostName>
  <ipAddress opt="ipv4,ipv6">
    <!--ro, opt, string, ip地址, attr:opt{req, string, 取值范围}, desc:当addressingFormatType为ipaddress时有效-->test
  </ipAddress>
  <portNo min="0" max="10">
    <!--ro, opt, int, 端口, range:[0,65535], attr:min{req, int, 最小值},max{req, int, 最大值}-->1
  </portNo>
  <userNameLen min="0" max="10">
    <!--ro, opt, int, 用户名长度, attr:min{req, int, 最小值},max{req, int, 最大值}-->1
  </userNameLen>
  <passwordLen min="0" max="10">
    <!--ro, opt, int, 密码长度, attr:min{req, int, 最小值},max{req, int, 最大值}-->1
  </passwordLen>
  <httpAuthenticationMethod opt="MD5digest,none,base64">
    <!--ro, req, enum, http认证方式, subType:string, [MD5digest#摘要,none#无,base64#base64], attr:opt{req, string, 取值范围}-->MD5digest
  </httpAuthenticationMethod>
  <ANPR>
    <!--ro, opt, object, ANPR-->
    <detectUploadPicturesType opt="test">
      <!--ro, opt, string, 侦测上传图片类型控制, attr:opt{req, string, 取值范围}-->test
    </detectUploadPicturesType>
    <alarmHttpPushProtocol opt="baseline,custom">
      <!--ro, opt, string, 报警http推协议, attr:opt{req, string, 取值范围}-->baseline
    </alarmHttpPushProtocol>
  </ANPR>
  <HttpHostUploadCtrl>
    <!--ro, opt, object, 报警上传控制参数能力-->
    <postDataTypes>
      <!--ro, opt, string, 卡口数据类型字典索引-->test
    </postDataTypes>
    <epoliceDataTypes>
      <!--ro, opt, string, 电警数据类型字典索引-->test
    </epoliceDataTypes>
    <eventDataTypes>
      <!--ro, opt, string, 事件数据类型字典索引-->test
    </eventDataTypes>
    <evidenceDataTypes>
      <!--ro, opt, string, 取证数据类型字典索引-->test
    </evidenceDataTypes>
    <uploadInterval min="1" max="2000">
      <!--ro, req, int, 上传间隔(1~2000ms)默认50, attr:min{req, int, 最小值},max{req, int, 最大值}-->0
    </uploadInterval>
    <uploadTimeout min="1" max="6000">
      <!--ro, req, int, 上传超时时间(1~6000ms)默认5000, attr:min{req, int, 最小值},max{req, int, 最大值}-->0
    </uploadTimeout>
    <noPlateUploadEnabled opt="true,false">
      <!--ro, req, bool, 无车牌数据上传使能, attr:opt{req, string, 取值范围}-->true
    </noPlateUploadEnabled>
    <historyEnabled opt="true,false">
      <!--ro, req, bool, 历史数据上传使能, attr:opt{req, string, 取值范围}-->true
    </historyEnabled>
    <uploadTimePlan>
      <!--ro, opt, object, 上传事件计划-->
      <enabled opt="true,false">
        <!--ro, req, bool, 使能, attr:opt{req, string, 取值范围}-->true
      </enabled>
      <timeList>
        <!--ro, opt, object, 时间列表-->
      </timeList>
    </uploadTimePlan>
  </HttpHostUploadCtrl>

```

```
<time>span>
  <!--ro, opt, object, 时间-->
  <timeId min="0" max="1">
    <!--ro, req, int, 时间编号, attr:min{req, int, 最小值},max{req, int, 最大值}-->0
  </timeId>
  <startHour min="0" max="23">
    <!--ro, opt, int, 开始小时, attr:min{req, int, 最小值},max{req, int, 最大值}-->0
  </startHour>
  <startMinute min="0" max="59">
    <!--ro, opt, int, 开始分钟, attr:min{req, int, 最小值},max{req, int, 最大值}-->0
  </startMinute>
  <endHour min="0" max="23">
    <!--ro, opt, int, 结束小时, attr:min{req, int, 最小值},max{req, int, 最大值}-->0
  </endHour>
  <endMinute min="0" max="59">
    <!--ro, opt, int, 结束分钟, attr:min{req, int, 最小值},max{req, int, 最大值}-->0
  </endMinute>
</timeSpan>
</timeList>
</uploadTimePlan>
</HttpHostUploadCtrl>
<Extensions>
  <!--ro, opt, object, 扩展-->
  <intervalBetweenEvents min="0" max="10">
    <!--ro, opt, int, 事件间隔, attr:min{req, int, 最小值},max{req, int, 最大值}-->0
  </intervalBetweenEvents>
</Extensions>
<uploadImagesDataType opt="URL,binary">
  <!--ro, opt, enum, 上传的图片类型, subType:string, [URL#url,binary#二进制数据], attr:opt{req, string, 取值范围}, desc:默认是二进制,在支持了云存储之后,配置成URL,才会生效-->URL
  </uploadImagesDataType>
  <httpBroken opt="true,false" def="true">
    <!--ro, opt, bool, 断网续传使能, attr:opt{req, string, 取值范围},def{req, string, 默认值}-->true
  </httpBroken>
  <SubscribeEventCap>
    <!--ro, opt, string, 订阅能力-->test
    <heartbeat min="0" max="10">
      <!--ro, opt, int, 心跳间隔时间, unit:s, unitType:时间, attr:min{req, int, 最小值},max{req, int, 最大值}-->1
    </heartbeat>
    <channelMode opt="all,list">
      <!--ro, opt, enum, 通道模式, subType:string, [all#表示设备不支持区分通道进行订阅,list#表示设备支持区分通道订阅], attr:opt{req, string, 取值范围}, desc:如果channelMode&eventMode均返回all,那么实际等同于设备不支持事件类型和事件通道的订阅-->all
    </channelMode>
    <eventMode opt="all,list">
      <!--ro, opt, enum, 事件模式, subType:string, [all#表示订阅设备的所有通道、目标的事件,list#表示设备支持区分事件类型、通道、目标订阅], attr:opt{req, string, 取值范围}-->all
    </eventMode>
    <EventList>
      <!--ro, opt, object, 指定报警上报的方式-->
      <Event>
        <!--ro, opt, string, 事件订阅信息-->test
      </Event>
    </EventList>
    <type>
      <!--ro, req, enum, 参考事件类型列表, subType:string, [AccessControllerEvent#门禁事件,IDCardInfoEvent#门禁身份证事件,FaceTemperatureMeasurementEvent#门禁测温事件,QRCodeEvent#门禁二维码事件,CertificateCaptureEvent#证件抓拍比对事件,UncertificateCompareEvent#无证比对事件,ConsumptionAndTransactionRecordEvent#消费和交易记录事件,ConsumptionEvent#消费事件,TransactionRecordEvent#交易记录事件,HealthInfoSyncQuery#健康信息查询事件,SetMealQuery#消费套餐查询,ConsumptionStatusQuery#消费状态查询,humanBodyComparison#人体比对,regionTargetNumberCounting#区域目标统计,cabinetChargingQuery#智能柜计费查询请求,paymentQRCodeQuery#查询第三方支付二维码,ConsumptionQuery#本地消费查询事件,topUpRequest#充值请求,topUpStatusQuery#查询充值状态,topUpEvent#充值事件,LogOffRefundStatusQuery#查询人员注销退款状态,LogOffRefundRequest#人员注销退款请求]-->AccessControllerEvent
    </type>
    <minorAlarm opt="0x400,0x401,0x402,0x403">
      <!--ro, opt, string, 报警次类型, attr:opt{req, string, 取值范围}, desc:type为AccessControllerEvent, IDCardInfoEvent该字段必填-->0x400,0x401
    </minorAlarm>
    <minorException opt="0x400,0x401,0x402,0x403">
      <!--ro, opt, string, 异常次类型, attr:opt{req, string, 取值范围}, desc:type为AccessControllerEvent, IDCardInfoEvent该字段必填-->0x400,0x401
    </minorException>
    <minorOperation opt="0x400,0x401,0x402,0x403">
      <!--ro, opt, string, 操作次类型, attr:opt{req, string, 取值范围}, desc:type为AccessControllerEvent, IDCardInfoEvent该字段必填-->0x400,0x401
    </minorOperation>
    <minorEvent opt="0x01,0x02,0x03,0x04">
      <!--ro, opt, string, 事件次类型, attr:opt{req, string, 取值范围}, desc:type为AccessControllerEvent, IDCardInfoEvent该字段必填-->0x400,0x401
    </minorEvent>
    <pictureURLType opt="binary,localURL,cloudStorageURL,EZVIZURL" def="cloudStorageURL">
      <!--ro, opt, enum, 报警图片格式, subType:string, [binary#二进制,localURL#设备本地URL,cloudStorageURL#云存储URL,EZVIZURL#萤石URL], attr:opt{req, string, 取值范围},def{req, string, 默认值}-->cloudStorageURL
    </pictureURLType>
  </Event>
</EventList>
<pictureURLType opt="binary,localURL,cloudStorageURL,EZVIZURL" def="cloudStorageURL">
  <!--ro, opt, enum, 该节点表示所有含图片事件图片上传方式, subType:string, [binary#二进制,localURL#设备本地URL,cloudStorageURL#云存储URL,EZVIZURL#萤石URL], attr:opt{req, string, 取值范围},def{req, string, 默认值}, desc:如果该字段下发,那么<Event>中<pictureURLType>不做生效; 如果不下发这个节点,设备采用能力集返回的图片默认上传方式上报,前端设备默认为二进制,后端设备默认为设备本地URL-->cloudStorageURL
</pictureURLType>
<ChangedUploadSub>
  <!--ro, opt, int, 变化上报-->0
  <interval>
    <!--ro, opt, int, 布防guid的生命周期, desc:默认5分钟,布防guid的生命周期,该间隔期间未过来重连,设备认为是新的一路布防,重新生成guid-->1
  </interval>
  <StatusSub>
    <!--ro, opt, bool, 状态订阅-->true
    <all>
      <!--ro, opt, bool, 是否订阅所有-->true
    </all>
    <channel>
      <!--ro, opt, bool, 订阅通道状态, desc:all为true时不必传入-->true
    </channel>
  </StatusSub>
</ChangedUploadSub>
```

```
</channel>
<hd>
  <!--ro, opt, bool, 订阅磁盘状态, desc:all为true时不必传入-->true
</hd>
<capability>
  <!--ro, opt, bool, 订阅能力集变化状态, desc:all为true时不必传入-->true
</capability>
</StatusSub>
</ChangedUploadSub>
</SubscribeEventCap>
<PackingSpaceRecognition>
  <!--ro, opt, object, 当前报警主机下车位检测监听事件控制参数, desc:对应事件为PackingSpaceRecognition;当设备不支持该节点时,默认为所有图片上传-->
  <uploadPicturesType opt="all,picturesTypes,notUpload" def="all">
    <!--ro, req, enum, 上传图片类型控制, subType:string, [all#所有,picturesTypes#指定图片类型上传,notUpload#不上传图片], attr:opt{req, string, 取值范围},def{req, string, 默认值}-->all
  </uploadPicturesType>
  <PicturesTypes size="2">
    <!--ro, opt, array, 指定图片类型上传列表, subType:object, dep:and,
    {$.HttpHostNotificationList[*].HttpHostNotification.PackingSpaceRecognition.uploadPicturesType,eq,picturesTypes}, attr:size{req, int, 数组元素总数}-->
    <picturesType opt="backgroundImage,plateImage">
      <!--ro, opt, enum, 上传图片类型, subType:string, [backgroundImage#背景大图,plateImage#车牌小图], attr:opt{req, string, 取值范围}-->backgroundImage
    </picturesType>
  </PicturesTypes>
</PackingSpaceRecognition>
</HttpHostNotificationCap>
```

69.15.11 配置报警上传控制

Request URL

PUT /ISAPI/Event/notification/httpHosts/<hostID>/uploadCtrl

查询参数

参数名称	参数类型	描述
hostID	string	

请求报文

```

<?xml version="1.0" encoding="UTF-8"?>

<HttpHostUploadCtrl xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, http主机上传控制, attr:version{req, string, 协议版本}-->
  <postDataTypes>
    <!--opt, string, 卡口数据类型字典索引-->1,2,3
  </postDataTypes>
  <epoliceDataTypes>
    <!--opt, string, 电警数据类型字典索引-->1,2,3
  </epoliceDataTypes>
  <eventDataTypes>
    <!--opt, string, 事件数据类型字典索引-->1,2,3
  </eventDataTypes>
  <evidenceDataTypes>
    <!--opt, string, 取证数据类型字典索引-->1,2,3
  </evidenceDataTypes>
  <uploadInterval min="1" max="2000">
    <!--req, int, 上传间隔, range:[1,2000], unit:ms, unitType:时间, attr:min{req, int, 最小值},max{req, int, 最大值}, desc:默认50-->50
  </uploadInterval>
  <uploadTimeout min="1" max="6000">
    <!--req, int, 上传超时时间, range:[1,6000], unit:ms, unitType:时间, attr:min{req, int, 最小值},max{req, int, 最大值}, desc:默认5000-->5000
  </uploadTimeout>
  <noPlateUploadEnabled opt="true,false">
    <!--req, bool, 无车牌数据上传使能, attr:opt{req, string, 取值范围}-->true
  </noPlateUploadEnabled>
  <historyEnabled opt="true,false">
    <!--req, bool, 历史数据上传使能, attr:opt{req, string, 取值范围}-->true
  </historyEnabled>
  <uploadTimePlan>
    <!--opt, object, 上传时间计划-->
    <enabled opt="true,false">
      <!--req, bool, 使能, attr:opt{req, string, 取值范围}-->true
    </enabled>
    <timeList>
      <!--opt, object, 时间列表-->
      <timeSpan>
        <!--opt, object, 时间-->
        <timeId min="0" max="1">
          <!--req, int, 时间id, range:[0,1], attr:min{req, int, 最小值},max{req, int, 最大值}-->1
        </timeId>
        <startHour min="0" max="23">
          <!--opt, int, 开始小时, range:[0,23], attr:min{req, int, 最小值},max{req, int, 最大值}-->1
        </startHour>
        <startMinute min="0" max="59">
          <!--opt, int, 开始分钟, range:[0,59], attr:min{req, int, 最小值},max{req, int, 最大值}-->1
        </startMinute>
        <endHour min="0" max="23">
          <!--opt, int, 结束小时, range:[0,23], attr:min{req, int, 最小值},max{req, int, 最大值}-->1
        </endHour>
        <endMinute min="0" max="59">
          <!--opt, int, 结束分钟, range:[0,59], attr:min{req, int, 最小值},max{req, int, 最大值}-->1
        </endMinute>
      </timeSpan>
    </timeList>
  </uploadTimePlan>
</HttpHostUploadCtrl>

```

响应报文

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 响应消息, attr:version{ro, req, string, 版本}-->
  <requestURL>
    <!--ro, req, string, 请求的URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, 状态码, subType:int, [0#OK,1#OK,2#Device Busy,3#Device Error,4#Invalid Operation,5#Invalid XML Format,6#Invalid XML Content,7#Reboot Required]-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, 状态信息, subType:string, [OK#成功,Device Busy#设备忙,Device Error#设备异常,Invalid Operation#无效的操作,Invalid XML Format#无效的XML格式,Invalid XML Content#无效的XML内容,Reboot#设备重启]-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, 详细错误码英文描述, desc:详细错误码的英文描述-->OK
  </subStatusCode>
</ResponseStatus>

```

69.15.12 删除全部监听主机配置参数

Request URL

DELETE /ISAPI/Event/notification/httpHosts

查询参数

无

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 响应消息, attr:version{ro, req, string, 版本}-->
  <requestURL>
    <!--ro, req, string, 请求的URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, 状态码, subType:int, [0#OK,1#OK,2#Device Busy,3#Device Error,4#Invalid Operation,5#Invalid XML Format,6#Invalid XML Content,7#Reboot Required]-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, 状态信息, subType:string, [OK#成功,Device Busy#设备忙,Device Error#设备异常,Invalid Operation#无效的操作,Invalid XML Format#无效的XML格式,Invalid XML Content#无效的XML内容,Reboot#设备重启]-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, 详细错误码英文描述, desc:详细错误码的英文描述-->OK
  </subStatusCode>
</ResponseStatus>
```

69.15.13 删除单个监听主机配置参数

Request URL

DELETE /ISAPI/Event/notification/httpHosts/<hostID>

查询参数

参数名称	参数类型	描述
hostID	string	通过POST /ISAPI/Event/notification/httpHosts?security=&iv=添加单个报警主机，成功会返回报警主机id

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 响应消息, attr:version{ro, req, string, 版本}-->
  <requestURL>
    <!--ro, req, string, 请求的URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, 状态码, subType:int, [0#OK,1#OK,2#Device Busy,3#Device Error,4#Invalid Operation,5#Invalid XML Format,6#Invalid XML Content,7#Reboot Required]-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, 状态信息, subType:string, [OK#成功,Device Busy#设备忙,Device Error#设备异常,Invalid Operation#无效的操作,Invalid XML Format#无效的XML格式,Invalid XML Content#无效的XML内容,Reboot#设备重启]-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, 详细错误码英文描述, desc:详细错误码的英文描述-->OK
  </subStatusCode>
</ResponseStatus>
```

69.16 移动网络配置

69.16.1 配置单个拨号参数

Request URL

PUT /ISAPI/System/Network/WirelessDial/Interfaces/<interfaceID>/dial?security=<security>&iv=<iv>

查询参数

参数名称	参数类型	描述
interfaceID	string	
security	string	security=1,表示报文中的UIMCardNum、DialNum、Username、Password、pinCode字段AES128加密;security=2,表示报文中的UIMCardNum、DialNum、Username、Password、pinCode字段AES256加密
iv	string	随机字符串，长度限制为32；iv表示初始化向量，在security=1或者2时为必须参数;

请求报文

```
<?xml version="1.0" encoding="UTF-8"?>

<Dial xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--opt, object, 拨号, attr:version{req, string, 协议版本}-->
  <enabled>
    <!--req, bool, 开启-->true
  </enabled>
  <DialMethod>
    <!--req, enum, 拨号方式, subType:string, [auto#自动,manual#手动]-->auto
  </DialMethod>
  <SwitchMethod>
    <!--req, enum, 切换方式, subType:string, [auto#自动,4GFirst#4G优先,3GFirst#3G优先,manualto2G#手动切为2G,manualto3G#手动切为3G,manualto4G#手动切为4G,5GFirst#5G优先,manualto5G#手动切为5G,manualto5GSA#手动切为5G SA组网,manualto5GNSA#手动切为5G NSA组网]-->auto
  </SwitchMethod>
  <OfflineTime>
    <!--opt, int, 离线时间-->0
  </OfflineTime>
  <UIMCardNum>
    <!--opt, string, UIM卡序号-->test
  </UIMCardNum>
  <DialNum>
    <!--opt, string, 拨号序号-->test
  </DialNum>
  <Username>
    <!--opt, string, 用户名-->test
  </Username>
  <Password>
    <!--opt, string, 密码-->test
  </Password>
  <APNname>
    <!--opt, string, APN名称-->test
  </APNname>
  <MTU>
    <!--opt, int, MTU-->0
  </MTU>
  <VerifyProto>
    <!--req, enum, 认证协议, subType:string, [auto#自动,CHAP#CHAP,PAP#PAP,none#无认证]-->auto
  </VerifyProto>
  <DefaultParam>
    <!--opt, object, 默认参数-->
    <DialNum>
      <!--opt, string, 拨号序号-->test
    </DialNum>
    <Username>
      <!--opt, string, 用户名-->test
    </Username>
    <Password>
      <!--opt, string, 密码-->test
    </Password>
    <APNname>
      <!--opt, string, APN名称-->test
    </APNname>
    <VerifyProto>
      <!--req, enum, 认证协议, subType:string, [auto#自动,CHAP#CHAP,PAP#PAP]-->auto
    </VerifyProto>
  </DefaultParam>
  <netAPN>
    <!--opt, string, 私有专网APN配置-->test
  </netAPN>
  <enabled4G>
    <!--opt, bool, 开启4G-->true
  </enabled4G>
  <enabledDNS>
    <!--opt, bool, 是否开启DNS手动配置-->true
  </enabledDNS>
  <Flow>
    <!--opt, object, 流量配置-->
    <limitEnabled>
      <!--opt, bool, 流量限额使能-->true
    </limitEnabled>
    <consumeFlow>
```

```
<!--opt, float, 使用流量, unit:MB, unitType:信息量-->0.000
</consumeFlow>
<threshold>
  <!--opt, int, 限额, unit:MB, unitType:信息量-->1
</threshold>
</Flow>
<pinCode>
  <!--opt, string, PIN码-->test
</pinCode>
<SIMCardNo>
  <!--opt, int, SIM卡的个数-->0
</SIMCardNo>
<voiceAndSMSEnabled>
  <!--opt, bool, 是否开启SIM卡语音和短信功能-->true
</voiceAndSMSEnabled>
<SIMCardType>
  <!--opt, enum, SIM卡类型, subType:string, [normalCard#普通卡,specialCard#特殊卡], desc:特殊卡: 如广电卡、白卡等需特殊处理的卡-->normalCard
</SIMCardType>
<DialStaticRouteList>
  <!--ro, opt, array, 拨号静态路由列表扩展, subType:object, desc:拨号静态路由支持最多四个路由配置-->
  <DialStaticRoute>
    <!--ro, opt, object, 拨号路由-->
    <ipAddress>
      <!--ro, opt, string, IP地址-->test
    </ipAddress>
    <subnetMask>
      <!--ro, opt, string, 子网掩码-->test
    </subnetMask>
  </DialStaticRoute>
</DialStaticRouteList>
</Dial>
```

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 响应消息, attr:version{ro, req, string, 版本}-->
  <requestURL>
    <!--ro, req, string, 请求的URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, 状态码, subType:int, [0#OK,1#OK,2#Device Busy,3#Device Error,4#Invalid Operation,5#Invalid XML Format,6#Invalid XML Content,7#Reboot Required]-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, 状态信息, subType:string, [OK#成功,Device Busy#设备忙,Device Error#设备异常,Invalid Operation#无效的操作,Invalid XML Format#无效的XML格式,Invalid XML Content#无效的XML内容,Reboot#设备重启]-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, 详细错误码英文描述, desc:详细错误码的英文描述-->OK
  </subStatusCode>
</ResponseStatus>
```

69.16.2 获取单个拨号参数

Request URL

GET /ISAPI/System/Network/WirelessDial/Interfaces/<interfaceID>/dial?security=<security>&iv=<iv>

查询参数

参数名称	参数类型	描述
interfaceID	string	--
security	string	security=1,表示报文中的UIMCardNum、DialNum、Username、Password、pinCode字段AES128加密;security=2,表示报文中的UIMCardNum、DialNum、Username、Password、pinCode字段AES256加密
iv	string	随机字符串, 长度限制为32; iv表示初始化向量, 在security=1或者2时为必须参数;

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<Dial xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, 拨号, attr:version{req, string, 协议版本}-->
  <enabled>
    <!--ro, req, bool, 开启-->true
  </enabled>
  <DialMethod>
    <!--ro, req, enum, 拨号方式, subType:string, [auto#自动,manual#手动]-->auto
  </DialMethod>
  <SwitchMethod>
    <!--ro, req, enum, 切换方式, subType:string, [auto#自动,4GFirst#4G优先,3GFirst#3G优先,manualto2G#手动切为2G,manualto3G#手动切为3G,manualto4G#手动切为4G,5GFirst#5G优先,manualto5G#手动切为5G,manualto5GSA#手动切为5G SA组网,manualto5GNSA#手动切为5G NSA组网]-->auto
  </SwitchMethod>
  <OfflineTime>
    <!--ro, opt, int, 离线时间-->0
  </OfflineTime>
  <UIMCardNum>
    <!--ro, opt, string, UIM卡序号-->test
  </UIMCardNum>
  <DialNum>
    <!--ro, opt, string, 拨号序号-->test
  </DialNum>
  <Username>
    <!--ro, opt, string, 用户名-->test
  </Username>
  <Password>
    <!--ro, opt, string, 密码-->test
  </Password>
  <APNname>
    <!--ro, opt, string, APN名称-->test
  </APNname>
  <MTU>
    <!--ro, opt, int, MTU-->0
  </MTU>
  <VerifyProto>
    <!--ro, req, enum, 认证协议, subType:string, [auto#自动,CHAP#CHAP,PAP#PAP,none#无认证]-->auto
  </VerifyProto>
  <DefaultParam>
    <!--ro, opt, object, 默认参数-->
    <DialNum>
      <!--ro, opt, string, 拨号序号-->test
    </DialNum>
    <Username>
      <!--ro, opt, string, 用户名-->test
    </Username>
    <Password>
      <!--ro, opt, string, 密码-->test
    </Password>
    <APNname>
      <!--ro, opt, string, APN名称-->test
    </APNname>
    <VerifyProto>
      <!--ro, req, enum, 认证协议, subType:string, [auto#自动,CHAP#CHAP,PAP#PAP]-->auto
    </VerifyProto>
  </DefaultParam>
  <netAPN>
    <!--ro, opt, string, 私有专网APN配置-->test
  </netAPN>
  <enabled4G>
    <!--ro, opt, bool, 开启4G-->true
  </enabled4G>
  <enabledDNS>
    <!--ro, opt, bool, 是否开启DNS手动配置-->true
  </enabledDNS>
  <Flow>
    <!--ro, opt, object, 流量配置-->
    <limitEnabled>
      <!--ro, opt, bool, 流量限额使能-->true
    </limitEnabled>
    <consumeFlow>
      <!--ro, opt, float, 使用流量, unit:MB, unitType:信息量-->0.000
    </consumeFlow>
    <threshold>
      <!--ro, opt, int, 限额, unit:MB, unitType:信息量-->1
    </threshold>
  </Flow>
  <pinCode>
    <!--ro, opt, string, PIN码-->test
  </pinCode>
  <SIMCardNo>
    <!--ro, opt, int, SIM卡的个数-->0
  </SIMCardNo>
  <voiceAndSMSEnabled>
    <!--ro, opt, bool, 是否开启SIM卡语音和短信功能-->true
  </voiceAndSMSEnabled>
  <SIMCardType>
    <!--ro, opt, enum, SIM卡类型, subType:string, [normalCard#普通卡,specialCard#特殊卡], desc:特殊卡: 如广电卡、白卡等需特殊处理的卡-->normalCard
  </SIMCardType>
  <DialStaticRouteList>
    <!--ro, opt, array, 拨号静态路由列表扩展, subType:object, desc:拨号静态路由支持最多四个路由配置-->
    <DialStaticRoute>
      <!--ro, opt, object, 拨号路由-->
      <ipAddress>
        <!--ro, opt, string, IP地址-->test
      </ipAddress>
    </DialStaticRoute>
  </DialStaticRouteList>
</Dial>
```

```
</ipAddress>
<subnetMask>
  <!--ro, opt, string, 子网掩码-->test
</subnetMask>
</DialStaticRoute>
</DialStaticRouteList>
</Dial>
```

69.16.3 获取单个拨号参数能力

Request URL

GET /ISAPI/System/Network/WirelessDial/Interfaces/<interfaceID>/dial/capabilities

查询参数

参数名称	参数类型	描述
interfaceID	string	

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<Dial xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, 拨号, attr:version{req, string, 协议版本}-->
  <enabled opt="true,false" def="false">
    <!--ro, req, bool, 开启, attr:opt{req, string, 取值范围},def{req, string, 默认值}-->true
  </enabled>
  <DialMethod opt="auto>manual">
    <!--ro, req, enum, 拨号方式, subType:string, [auto#自动>manual#手动], attr:opt{req, string, 取值范围}-->auto
  </DialMethod>
  <SwitchMethod opt="auto,4Gfirst,3Gfirst>manualto2G>manualto3G>manualto4G,cableFirst,5Gfirst>manualto5G>manualto5GSA>manualto5GNSA">
    <!--ro, req, enum, 切换方式, subType:string, [auto#自动,4Gfirst#4G优先,3Gfirst#3G优先>manualto2G#手动切为2G>manualto3G#手动切为3G>manualto4G#手动切为4G,5Gfirst#5G优先>manualto5G#手动切为5G,adjust#老模块自动模式的时候从高频网络切到低频它模块自己就切不回来了而adjust模式可以帮助模块切换上去>manualto5GSA#手动切为5G SA组网>manualto5GNSA#手动切为5G NSA组网], attr:opt{req, string, 取值范围}-->auto
  </SwitchMethod>
  <OfflineTime min="30" max="65535">
    <!--ro, opt, int, 离线时间, attr:min{req, int, 最小值},max{req, int, 最大值}-->0
  </OfflineTime>
  <UIMCardNum min="1" max="32">
    <!--ro, opt, string, UIM卡序号, attr:min{req, int, 最小值},max{req, int, 最大值}-->test
  </UIMCardNum>
  <DialNum min="1" max="32">
    <!--ro, opt, string, 拨号序号, attr:min{req, int, 最小值},max{req, int, 最大值}-->test
  </DialNum>
  <Username min="1" max="32">
    <!--ro, opt, string, 用户名, attr:min{req, int, 最小值},max{req, int, 最大值}-->test
  </Username>
  <Password min="1" max="32">
    <!--ro, opt, string, 密码, attr:min{req, int, 最小值},max{req, int, 最大值}, desc:不返回密码内容-->null
  </Password>
  <APNname min="1" max="32">
    <!--ro, opt, string, APN名称, attr:min{req, int, 最小值},max{req, int, 最大值}-->test
  </APNname>
  <MTU min="100" max="1500">
    <!--ro, opt, int, MTU, attr:min{req, int, 最小值},max{req, int, 最大值}-->0
  </MTU>
  <VerifyProto opt="auto,CHAP,PAP,none">
    <!--ro, req, enum, 认证协议, subType:string, [auto#自动,CHAP#CHAP,PAP#PAP,none#无认证], attr:opt{req, string, 取值范围}-->auto
  </VerifyProto>
  <DefaultParam>
    <!--ro, opt, object, 默认参数-->
    <DialNum min="1" max="32">
      <!--ro, opt, string, 拨号序号, attr:min{req, int, 最小值},max{req, int, 最大值}-->test
    </DialNum>
    <Username min="1" max="32">
      <!--ro, opt, string, 用户名, attr:min{req, int, 最小值},max{req, int, 最大值}-->test
    </Username>
    <Password min="1" max="32">
      <!--ro, opt, string, 密码, attr:min{req, int, 最小值},max{req, int, 最大值}, desc:不返回密码内容-->null
    </Password>
    <APNname min="1" max="32">
      <!--ro, opt, string, APN名称, attr:min{req, int, 最小值},max{req, int, 最大值}-->test
    </APNname>
    <VerifyProto opt="auto,CHAP,PAP">
      <!--ro, req, enum, 认证协议, subType:string, [auto#自动,CHAP#CHAP,PAP#PAP], attr:opt{req, string, 取值范围}-->auto
    </VerifyProto>
  </DefaultParam>
  <netAPN min="0" max="32">
    <!--ro, opt, string, 私有专网APN配置, attr:min{req, int, 最小值},max{req, int, 最大值}-->test
  </netAPN>
  <enabled4G opt="true,false" def="false">
```

```
<enabled4G opt="true,false" def="false">
  <!--ro, opt, bool, 开启4G, attr:opt{req, string, 取值范围},def{req, string, 默认值}-->true
</enabled4G>
<isSupportNetworkKeepAlive>
  <!--ro, opt, bool, 是否支持网络保活, desc:/ISAPI/System/Network/WirelessDial/Interfaces/<ID>/keepAlive?format=json-->true
</isSupportNetworkKeepAlive>
<enabledDNS opt="true,false" def="false">
  <!--ro, opt, bool, 是否开启DNS手动配置, attr:opt{req, string, 取值范围},def{req, string, 默认值}-->true
</enabledDNS>
<Flow>
  <!--ro, opt, object, 流量配置-->
  <limitEnabled opt="true,false">
    <!--ro, opt, bool, 流量限额使能, attr:opt{req, string, 取值范围}-->true
  </limitEnabled>
  <consumeFlow min="0.0" max="9999999.9">
    <!--ro, opt, float, 支持显示已用流量, attr:min{req, float, 最小值},max{req, float, 最大值}-->0.000
  </consumeFlow>
  <threshold min="0" max="999999">
    <!--ro, opt, int, 限额, unit:MB, unitType:信息量, attr:min{req, int, 最小值},max{req, int, 最大值}-->1
  </threshold>
</Flow>
<pinCode min="0" max="10">
  <!--ro, opt, string, PIN码, attr:min{req, int, 最小值},max{req, int, 最大值}-->test
</pinCode>
<SIMCardNo min="0" max="16">
  <!--ro, opt, int, SIM卡的个数, attr:min{req, int, 最小值},max{req, int, 最大值}-->0
</SIMCardNo>
<isSupportRecamp>
  <!--ro, opt, bool, 是否支持重新驻网, desc:对应uri:/ISAPI/System/Network/WirelessDial/Interfaces/<interfaceID>/dial/recamp?format=json-->true
</isSupportRecamp>
<voiceAndSMSEnabled opt="true,false" def="false">
  <!--ro, opt, bool, 是否开启SIM卡语音和短信功能, attr:opt{req, string, 取值范围},def{req, string, 默认值}-->true
</voiceAndSMSEnabled>
<SIMCardType opt="normalCard,specialCard">
  <!--ro, opt, enum, SIM卡类型, subType:string, [normalCard#普通卡,specialCard#特殊卡], attr:opt{req, string, 取值范围}, desc:特殊卡: 如广电卡、白卡等需特殊处理的卡-->normalCard
</SIMCardType>
<DialStaticRouteList size="4">
  <!--ro, opt, array, 拨号静态路由列表扩展, subType:object, attr:size{req, int, 数组元素总数}, desc:拨号静态路由支持最多四个路由配置-->
  <DialStaticRoute>
    <!--ro, opt, object, 拨号路由-->
    <ipAddress min="0" max="32">
      <!--ro, opt, string, IP地址, attr:min{req, int, 最小值},max{req, int, 最大值}-->test
    </ipAddress>
    <subnetMask min="0" max="32">
      <!--ro, opt, string, 子网掩码, attr:min{req, int, 最小值},max{req, int, 最大值}-->test
    </subnetMask>
  </DialStaticRoute>
</DialStaticRouteList>
</Dial>
```

参数名称	参数值	参数类型(Content-Type)	Content-ID	文件名称(filename)	描述
Dial	[报文内容]	application/xml	--	--	--

说明： 此协议通过表单格式传输，表单框架详情说明见章节4.5.1.4，如下实例：

```
--<frontier>
Content-Disposition: form-data; name=参数名称;filename=文件名称
Content-Type: 参数类型
Content-Length: ****
Content-ID: 内容ID
参数值
```

- 参数名称：表单元Headers中Content-Disposition的name属性，表示此表单元的名称
- 参数类型（Content-Type）：表单元Headers中Content-Type属性
- 文件名称（filename）：表单元Headers中Content-Disposition的filename属性，仅在本表单元传输数据为文件类型存在，表示表单元Body的文件名
- 参数值：表单元的body内容

69.17 IP地址配置

69.17.1 配置IP地址参数

Request URL

PUT /ISAPI/System/Network/interfaces/<interfaceID>/ipAddress

查询参数

参数名称	参数类型	描述
interfaceID	string	

请求报文

有效期：2024-08-15

湖南云舟电力科技有限公司

<?xml version="1.0" encoding="UTF-8"?>

```
<IPAddress xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, IP地址参数, attr:version{req, string, 协议版本}-->
  <ipVersion>
    <!--req, enum, IP地址版本, subType:string, [v4#IPv4,v6#IPv6,dual#IPv4和IPv6都支持]-->v4
  </ipVersion>
  <addressingType>
    <!--req, enum, 地址类型, subType:string, [static#静态,dynamic#动态,api#自动IP寻址]-->static
  </addressingType>
  <ipAddress>
    <!--opt, string, IPv4地址-->test
  </ipAddress>
  <subnetMask>
    <!--opt, string, IPv4子网掩码-->test
  </subnetMask>
  <ipv6Address>
    <!--opt, string, IPv6地址-->test
  </ipv6Address>
  <bitMask>
    <!--opt, string, IPv6子网掩码-->test
  </bitMask>
  <DefaultGateway>
    <!--opt, object, 默认网关-->
    <ipAddress>
      <!--opt, string, IPv4地址-->test
    </ipAddress>
    <ipv6Address>
      <!--opt, string, IPv6地址-->test
    </ipv6Address>
  </DefaultGateway>
  <PrimaryDNS>
    <!--opt, object, 首选DNS-->
    <ipAddress>
      <!--opt, string, IPv4地址-->test
    </ipAddress>
    <ipv6Address>
      <!--opt, string, IPv6地址-->test
    </ipv6Address>
  </PrimaryDNS>
  <SecondaryDNS>
    <!--opt, object, 备用DNS-->
    <ipAddress>
      <!--opt, string, IPv4地址-->test
    </ipAddress>
    <ipv6Address>
      <!--opt, string, IPv6地址-->test
    </ipv6Address>
  </SecondaryDNS>
  <DNSEnable>
    <!--opt, bool, DNS自动分配-->true
  </DNSEnable>
  <Ipv6Mode>
    <!--opt, object, IPv6模式-->
    <ipV6AddressingType>
      <!--opt, enum, IPv6地址类型, subType:string, [ra#路由公告>manual#手动,dhcp#动态主机配置]-->ra
    </ipV6AddressingType>
    <ipV6AddressList>
      <!--opt, array, IPv6地址列表, subType:object-->
      <v6Address>
        <!--opt, object, IPv6地址-->
        <id>
          <!--opt, string, IPv6地址索引-->test
        </id>
        <type>
          <!--opt, enum, IPv6地址类型, subType:string, [ra#路由公告>manual#手动,dhcp#动态主机配置]-->ra
        </type>
        <address>
          <!--opt, string, IPv6地址-->test
        </address>
        <bitMask>
          <!--opt, string, IPv6子网掩码-->test
        </bitMask>
      </v6Address>
    </ipV6AddressList>
  </Ipv6Mode>
  <DoubleNetRoute>
    <!--opt, object, 双网口路由-->
    <ipAddress>
      <!--opt, string, IP地址-->test
    </ipAddress>
    <subnetMask>
      <!--opt, string, 子网掩码-->test
    </subnetMask>
  </DoubleNetRoute>
  <LANConfigMode>
    <!--opt, enum, LAN配置模式, subType:string, [manual#手动,auto#自动], desc:路由器进行配置的时候需要选择使用自动还是手动模式, ipv4的自动模式下就是设备默认的LAN口所在网卡的IP和子网掩码信息, 手动模式下, 就需要配置相关的LAN所在的网卡信息, LAN网卡信息需要和DHCP的网段信息相互校验, 保证其在同一个网段-->>manual
  </LANConfigMode>
</IPAddress>
```

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 响应消息, attr:version{ro, req, string, 版本}-->
  <requestURL>
    <!--ro, req, string, 请求的URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, 状态码, subType:int, [0#OK,1#OK,2#Device Busy,3#Device Error,4#Invalid Operation,5#Invalid XML Format,6#Invalid XML Content,7#Reboot Required]-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, 状态信息, subType:string, [OK#成功,Device Busy#设备忙,Device Error#设备异常,Invalid Operation#无效的操作,Invalid XML Format#无效的XML格式,Invalid XML Content#无效的XML内容,Reboot#设备重启]-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, 详细错误码英文描述, desc:详细错误码的英文描述-->OK
  </subStatusCode>
</ResponseStatus>
```

69.17.2 获取IP地址参数

Request URL

GET /ISAPI/System/Network/interfaces/<interfaceID>/ipAddress

查询参数

参数名称	参数类型	描述
interfaceID	string	

请求报文

无

响应报文

湖南云舟电力科技有限公司
有效期：2024-08-15

<?xml version="1.0" encoding="UTF-8"?>

```
<IPAddress xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, IP地址参数, attr:version{req, string, 协议版本}-->
  <ipVersion>
    <!--ro, req, enum, IP地址版本, subType:string, [v4#IPv4,v6#IPv6,dual#IPv4和IPv6都支持]-->v4
  </ipVersion>
  <addressingType>
    <!--ro, req, enum, 地址类型, subType:string, [static#静态,dynamic#动态,apipa#自动IP寻址]-->static
  </addressingType>
  <ipAddress>
    <!--ro, opt, string, IPv4地址-->test
  </ipAddress>
  <subnetMask>
    <!--ro, opt, string, IPv4子网掩码-->test
  </subnetMask>
  <ipv6Address>
    <!--ro, opt, string, IPv6地址-->test
  </ipv6Address>
  <bitMask>
    <!--ro, opt, string, IPv6子网掩码-->test
  </bitMask>
  <DefaultGateway>
    <!--ro, opt, object, 默认网关-->
    <ipAddress>
      <!--ro, opt, string, IPv4地址-->test
    </ipAddress>
    <ipv6Address>
      <!--ro, opt, string, IPv6地址-->test
    </ipv6Address>
  </DefaultGateway>
  <PrimaryDNS>
    <!--ro, opt, object, 首选DNS-->
    <ipAddress>
      <!--ro, opt, string, IPv4地址-->test
    </ipAddress>
    <ipv6Address>
      <!--ro, opt, string, IPv6地址-->test
    </ipv6Address>
  </PrimaryDNS>
  <SecondaryDNS>
    <!--ro, opt, object, 备用DNS-->
    <ipAddress>
      <!--ro, opt, string, IPv4地址-->test
    </ipAddress>
    <ipv6Address>
      <!--ro, opt, string, IPv6地址-->test
    </ipv6Address>
  </SecondaryDNS>
  <DNSEnable>
    <!--ro, opt, bool, DNS自动分配-->true
  </DNSEnable>
  <Ipv6Mode>
    <!--ro, opt, object, IPv6模式-->
    <ipV6AddressingType>
      <!--ro, opt, enum, IPv6地址类型, subType:string, [ra#路由公告>manual#手动,dhcp#动态主机配置]-->ra
    </ipV6AddressingType>
    <ipV6AddressList>
      <!--ro, opt, array, IPv6地址列表, subType:object-->
      <v6Address>
        <!--ro, opt, object, IPv6地址-->
        <id>
          <!--ro, opt, string, IPv6地址索引-->test
        </id>
        <type>
          <!--ro, opt, enum, IPv6地址类型, subType:string, [ra#路由公告>manual#手动,dhcp#动态主机配置]-->ra
        </type>
        <address>
          <!--ro, opt, string, IPv6地址-->test
        </address>
        <bitMask>
          <!--ro, opt, string, IPv6子网掩码-->test
        </bitMask>
      </v6Address>
    </ipV6AddressList>
  </Ipv6Mode>
  <DoubleNetRoute>
    <!--ro, opt, object, 双网口路由-->
    <ipAddress>
      <!--ro, opt, string, IP地址-->test
    </ipAddress>
    <subnetMask>
      <!--ro, opt, string, 子网掩码-->test
    </subnetMask>
  </DoubleNetRoute>
  <LANConfigMode>
    <!--ro, opt, enum, LAN配置模式, subType:string, [manual#手动,auto#自动], desc:路由器进行配置的时候需要选择使用自动还是手动模式, ipv4的自动模式下就是设备默认的LAN口所在网卡的IP和子网掩码信息, 手动模式下, 就需要配置相关的LAN所在的网卡信息, LAN网卡信息需要和DHCP的网段信息相互校验, 保证其在同一个网段-->>manual
  </LANConfigMode>
</IPAddress>
```

69.17.3 获取单个TCPIP参数

Request URL

GET /ISAPI/System/Network/interfaces/<interfaceID>

查询参数

参数名称	参数类型	描述
interfaceID	string	

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<NetworkInterface xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, IP地址参数, attr:version{req, string, 协议版本}-->
  <id>
    <!--ro, req, string, 网卡索引-->test
  </id>
  <IPAddress>
    <!--ro, req, object, IP地址参数-->
    <ipVersion>
      <!--ro, req, enum, IP地址版本, subType:string, [v4#IPv4,v6#IPv6,dual#IPv4和IPv6都支持]-->v4
    </ipVersion>
    <addressingType>
      <!--ro, req, enum, 地址类型, subType:string, [static#静态,dynamic#动态,apipa#自动IP寻址]-->static
    </addressingType>
    <ipAddress>
      <!--ro, opt, string, IPv4地址-->test
    </ipAddress>
    <subnetMask>
      <!--ro, opt, string, IPv4子网掩码-->test
    </subnetMask>
    <ipv6Address>
      <!--ro, opt, string, IPv6地址-->test
    </ipv6Address>
    <bitMask>
      <!--ro, opt, string, IPv6子网掩码-->test
    </bitMask>
    <DefaultGateway>
      <!--ro, opt, object, 默认网关-->
      <ipAddress>
        <!--ro, opt, string, IPv4地址-->test
      </ipAddress>
      <ipv6Address>
        <!--ro, opt, string, IPv6地址-->test
      </ipv6Address>
    </DefaultGateway>
    <PrimaryDNS>
      <!--ro, opt, object, 首选DNS-->
      <ipAddress>
        <!--ro, opt, string, IPv4地址-->test
      </ipAddress>
      <ipv6Address>
        <!--ro, opt, string, IPv6地址-->test
      </ipv6Address>
    </PrimaryDNS>
    <SecondaryDNS>
      <!--ro, opt, object, 备用DNS-->
      <ipAddress>
        <!--ro, opt, string, IPv4地址-->test
      </ipAddress>
      <ipv6Address>
        <!--ro, opt, string, IPv6地址-->test
      </ipv6Address>
    </SecondaryDNS>
    <DNSEnable>
      <!--ro, opt, bool, DNS自动分配-->true
    </DNSEnable>
    <IPv6Mode>
      <!--ro, opt, object, IPv6模式-->
      <ipV6AddressingType>
        <!--ro, opt, enum, IPv6地址类型, subType:string, [ra#路由公告,manual#手动,dhcp#动态主机配置]-->ra
      </ipV6AddressingType>
      <ipV6AddressList>
        <!--ro, opt, array, IPv6地址列表, subType:object-->
        <v6Address>
          <!--ro, opt, object, IPv6地址-->
          <id>
            <!--ro, opt, string, IPv6地址索引-->test
          </id>
          <type>
            <!--ro, opt, enum, IPv6地址类型, subType:string, [ra#路由公告,manual#手动,dhcp#动态主机配置]-->ra
          </type>
        </v6Address>
      </ipV6AddressList>
    </IPv6Mode>
  </IPAddress>
</NetworkInterface>
```

```
<!--ro, opt, enum, IPV6地址类型, subType:string, [IPv6路由出口,manual#手动,auto#自动]-->v6
</type>
<address>
  <!--ro, opt, string, IPV6地址-->test
</address>
<bitMask>
  <!--ro, opt, string, IPV6子网掩码-->test
</bitMask>
</v6Address>
</IPv6AddressList>
</IPv6Mode>
<DoubleNetRoute>
  <!--ro, opt, object, 双网口路由, desc:双网隔离模式下, 若支持多个路由配置时, 其第一组路由由地址通过该节点进行配置-->
  <ipAddress>
    <!--ro, opt, string, IP地址-->test
  </ipAddress>
  <subnetMask>
    <!--ro, opt, string, 子网掩码-->test
  </subnetMask>
</DoubleNetRoute>
<LANConfigMode>
  <!--ro, opt, enum, LAN配置模式, subType:string, [manual#手动,auto#自动], desc:路由器进行配置的时候需要选择使用自动还是手动模式, ipv4的自动模式下就是设备默认的LAN口所在网卡的IP和子网掩码信息, 手动模式下, 就需要配置相关的LAN所在的网卡信息, LAN网卡信息需要和DHCP的网段信息相互校验, 保证其在同一个网段-->manual
</LANConfigMode>
<DoubleNetRouteList>
  <!--ro, opt, array, 双网口路由列表扩展, subType:object, desc:双网隔离模式下, 支持最多四个路由配置;第一个路由器地址需通过DoubleNetRoute配置, 剩余扩展路由可通过本节点配置-->
  <DoubleNetRoute>
    <!--ro, opt, object, 双网口路由-->
    <ipAddress>
      <!--ro, opt, string, IP地址-->test
    </ipAddress>
    <subnetMask>
      <!--ro, opt, string, 子网掩码-->test
    </subnetMask>
  </DoubleNetRoute>
</DoubleNetRouteList>
</IPAddress>
<Wireless>
  <!--ro, opt, object, WiFi参数配置-->
  <enabled>
    <!--ro, req, bool, 无线使能开关-->true
  </enabled>
  <wirelessNetworkMode>
    <!--ro, opt, enum, 无线网络工作模式, subType:string, [infrastructure#无固定设施网络,adhoc#点对点网络]-->infrastructure
  </wirelessNetworkMode>
  <channel>
    <!--ro, opt, string, 通道索引-->test
  </channel>
  <ssid>
    <!--ro, opt, string, 服务集标识-->test
  </ssid>
  <wmmEnabled>
    <!--ro, opt, bool, 无线网多媒体使能-->true
  </wmmEnabled>
  <WirelessSecurity>
    <!--ro, opt, object, 无线安全-->
    <securityMode>
      <!--ro, opt, enum, 安全模式, subType:string, [disable#关闭,WEP#有线等效保密,WPA-personal#WPA个人安全接入,WPA2-personal#WPA2个人安全接入,WPA-RADIUS#WPA+RADIUS服务器,WPA-enterprise#WPA企业安全接入,WPA2-enterprise#WPA2企业安全接入]-->disable
    </securityMode>
    <WEP>
      <!--ro, opt, object, 有线等效保密, dep:or,{$.NetworkInterface.Wireless.WirelessSecurity.securityMode,eq,WEP}-->
      <authenticationType>
        <!--ro, req, enum, 认证类型, subType:string, [open#开启验证,sharedkey#共享密钥,auto#自动验证]-->open
      </authenticationType>
      <defaultTransmitKeyIndex>
        <!--ro, req, int, 默认传输密钥个数-->1
      </defaultTransmitKeyIndex>
      <wepKeyLength>
        <!--ro, opt, int, 密钥长度, range:[64,128]-->1
      </wepKeyLength>
      <EncryptionKeyList>
        <!--ro, opt, array, 密钥列表, subType:object-->
        <encryptionKey>
          <!--ro, req, string, 密钥-->test
        </encryptionKey>
      </EncryptionKeyList>
    </WEP>
    <WPA>
      <!--ro, opt, object, WPA个人安全接入, dep:or,{$.NetworkInterface.Wireless.WirelessSecurity.securityMode,eq,WPA}-->
      <algorithmType>
        <!--ro, req, enum, 算法类型, subType:string, [TKIP#TKIP,AES#AES,TKIP/AES#TKIP/AES]-->TKIP
      </algorithmType>
      <sharedKey>
        <!--ro, req, string, 共享密钥-->test
      </sharedKey>
      <wpaKeyLength>
        <!--ro, req, int, 密钥长度, range:[8,63]-->1
      </wpaKeyLength>
    </WPA>
  </WirelessSecurity>
</workScene>
```

```
<!--ro, opt, enum, 工作场景模式, subType:string, [computerRoom#机房端,monitorTerminal#摄像机端,centralTerminal#中心端]-->computerRoom
</workScene>
<protocol>
  <!--ro, opt, enum, 协议类型, subType:string, [802.11ac#802.11ac]-->802.11ac
</protocol>
<protocolRealTime>
  <!--ro, opt, string, 实时模式-->test
</protocolRealTime>
<hideSsid>
  <!--ro, opt, bool, 隐藏SSID-->true
</hideSsid>
<ChannelConfig>
  <!--ro, opt, object, 信道配置-->
  <width>
    <!--ro, opt, enum, 带宽, subType:string, [auto#auto,20#20,40#40,80#80]-->auto
  </width>
  <autoWidth>
    <!--ro, opt, string, 带宽-->test
  </autoWidth>
  <channel>
    <!--ro, opt, string, 信道-->test
  </channel>
  <autoChannel>
    <!--ro, opt, string, 自动信道-->test
  </autoChannel>
  <transmitPower>
    <!--ro, opt, enum, 功率, subType:int, [9#9,12#12,15#15,18#18,21#21,24#24,27#27]-->9
  </transmitPower>
  <transmitPowerRealTime>
    <!--ro, opt, int, 实时功率-->1
  </transmitPowerRealTime>
  <countryID>
    <!--ro, opt, int, 国家码-->1
  </countryID>
</ChannelConfig>
<rate>
  <!--ro, opt, enum, 频率, subType:string, [2.4GHz#2.4GHz,5GHz#5GHz,auto#auto]-->2.4GHz
</rate>
<dialUpGroupNo>
  <!--ro, opt, string, SSID拨码组号-->test
</dialUpGroupNo>
<baseNoisevalue>
  <!--ro, opt, int, 底噪值, step:1-->1
</baseNoisevalue>
</Wireless>
<Discovery>
  <!--ro, opt, object, 网络发现配置-->
  <UPnP>
    <!--ro, req, object, 通用即插即用-->
    <enabled>
      <!--ro, req, bool, 通用即用即插协议使能-->true
    </enabled>
  </UPnP>
  <Zeroconf>
    <!--ro, opt, object, Zeroconf协议-->
    <enabled>
      <!--ro, req, bool, Zeroconf协议使能-->true
    </enabled>
  </Zeroconf>
</Discovery>
<Link>
  <!--ro, opt, object, 设备单个网卡连接信息-->
  <MACAddress>
    <!--ro, req, string, MAC地址-->test
  </MACAddress>
  <autoNegotiation>
    <!--ro, req, bool, 自适应使能-->true
  </autoNegotiation>
  <speed>
    <!--ro, req, enum, 链接速度, subType:int, [10#10M,100#100M,1000#1000M]-->10
  </speed>
  <duplex>
    <!--ro, req, enum, 双工模式, subType:string, [half#half,full#full]-->half
  </duplex>
  <MTU>
    <!--ro, req, int, MTU-->1
  </MTU>
  <linkMode>
    <!--ro, opt, enum, 连接模式, subType:string, [optical#光口模式,electrical#电口模式]-->optical
  </linkMode>
</Link>
<defaultConnection>
  <!--ro, opt, bool, 是否为默认网络连接, desc:当存在多个网卡时需要改节点-->true
</defaultConnection>
<ActiveMulticast>
  <!--ro, opt, object, 主动多播-->
  <enabled>
    <!--ro, req, bool, 使能开关-->true
  </enabled>
  <streamID>
    <!--ro, req, enum, 码流索引, subType:string, [main#main]-->main
  </streamID>
  <ipV4Address>
```

```
<!--ro, opt, string, IPV4地址-->test
</ipV4Address>
<ipV6Address>
  <!--ro, opt, string, IPV6地址-->test
</ipV6Address>
<port>
  <!--ro, opt, int, 端口号-->1
</port>
</ActiveMulticast>
<macAddress>
  <!--ro, opt, string, 物理地址-->test
</macAddress>
<EthernetPortList>
  <!--ro, opt, array, 以太网端口信息列表, subType:object-->
  <EthernetPort>
    <!--ro, opt, object, 以太网端口信息-->
    <id>
      <!--ro, req, int, 端口索引, range:[1,4]-->1
    </id>
    <MACAddress>
      <!--ro, req, string, 物理地址-->test
    </MACAddress>
    <status>
      <!--ro, opt, enum, 连接状态, subType:string, [connected#已连接,disconnect#已断开]-->connected
    </status>
    <speed>
      <!--ro, req, enum, 速率, subType:string, [10#10M,100#100M,1000#1000M,10000#10000M]-->10
    </speed>
  </EthernetPort>
</EthernetPortList>
<Extensions>
  <!--ro, opt, object, 扩展信息-->
  <NetworkCardType>
    <!--ro, opt, enum, 网卡类型, subType:string, [normal#网卡,internal#内网卡,external#外网卡,main#主网卡]-->normal
  </NetworkCardType>
</Extensions>
<interfaceName>
  <!--ro, opt, string, 网卡名称-->test
</interfaceName>
<enabled>
  <!--ro, opt, bool, 网卡使能开关, desc:门禁微校认证6301设备支持对网卡启用、关闭, 启用之后方可进行配置参数, 为兼容各产线, 默认为true.-->true
</enabled>
</NetworkInterface>
```

69.17.4 配置单个TCPIP参数

Request URL

PUT /ISAPI/System/Network/interfaces/<interfaceID>

查询参数

参数名称	参数类型	描述
interfaceID	string	

请求报文

```
<?xml version="1.0" encoding="UTF-8"?>

<NetworkInterface xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, IP地址参数, attr:version{req, string, 协议版本}-->
  <id>
    <!--req, string, 网卡索引-->test
  </id>
  <IPAddress>
    <!--req, object, IP地址参数-->
    <ipVersion>
      <!--req, enum, IP地址版本, subType:string, [v4#IPv4,v6#IPv6,dual#IPv4和IPv6都支持]-->v4
    </ipVersion>
    <addressingType>
      <!--req, enum, 地址类型, subType:string, [static#静态,dynamic#动态,apiipa#自动IP寻址]-->static
    </addressingType>
    <ipAddress>
      <!--opt, string, IPV4地址-->test
    </ipAddress>
    <subnetMask>
      <!--opt, string, IPV4子网掩码-->test
    </subnetMask>
    <ipV6Address>
      <!--opt, string, IPV6地址-->test
    </ipV6Address>
    <bitMask>
      <!--opt, string, IPV6子网掩码-->test
    </bitMask>
    <DefaultGateway>
      <!--opt, object, 默认网关-->
```

```

<ipAddress>
  <!--opt, string, IPV4地址-->test
</ipAddress>
<ipv6Address>
  <!--opt, string, IPV6地址-->test
</ipv6Address>
</DefaultGateway>
<PrimaryDNS>
  <!--opt, object, 首选DNS-->
  <ipAddress>
    <!--opt, string, IPV4地址-->test
  </ipAddress>
  <ipv6Address>
    <!--opt, string, IPV6地址-->test
  </ipv6Address>
</PrimaryDNS>
<SecondaryDNS>
  <!--opt, object, 备用DNS-->
  <ipAddress>
    <!--opt, string, IPV4地址-->test
  </ipAddress>
  <ipv6Address>
    <!--opt, string, IPV6地址-->test
  </ipv6Address>
</SecondaryDNS>
<DNSEnable>
  <!--opt, bool, DNS自动分配-->true
</DNSEnable>
<Ipv6Mode>
  <!--opt, object, IPV6模式-->
  <ipv6AddressingType>
    <!--opt, enum, IPV6地址类型, subType:string, [ra#路由公告,manual#手动,dhcp#动态主机配置]-->ra
  </ipv6AddressingType>
  <ipv6AddressList>
    <!--opt, array, IPV6地址列表, subType:object-->
    <v6Address>
      <!--opt, object, IPV6地址-->
      <id>
        <!--opt, string, IPV6地址索引-->test
      </id>
      <type>
        <!--opt, enum, IPV6地址类型, subType:string, [ra#路由公告,manual#手动,dhcp#动态主机配置]-->ra
      </type>
      <address>
        <!--opt, string, IPV6地址-->test
      </address>
      <bitMask>
        <!--opt, string, IPV6子网掩码-->test
      </bitMask>
    </v6Address>
  </ipv6AddressList>
</Ipv6Mode>
<DoubleNetRoute>
  <!--opt, object, 双网口路由, desc:双网隔离模式下, 若支持多个路由配置时, 其第一组路由地址通过该节点进行配置-->
  <ipAddress>
    <!--opt, string, IP地址-->test
  </ipAddress>
  <subnetMask>
    <!--opt, string, 子网掩码-->test
  </subnetMask>
</DoubleNetRoute>
<LANConfigMode>
  <!--opt, enum, LAN配置模式, subType:string, [manual#手动,auto#自动], desc:路由器进行配置的时候需要选择使用自动还是手动模式, ipv4的自动模式下就是设备默认LAN口所在网卡的IP和子网掩码信息, 手动模式下, 就需要配置相关的LAN所在的网卡信息, LAN网卡信息需要和DHCP的网段信息相互校验, 保证其在同一个网段-->manual
</LANConfigMode>
<DoubleNetRouteList>
  <!--opt, array, 双网口路由列表扩展, subType:object, desc:双网隔离模式下, 支持最多四个路由配置;第一个路由器地址需通过DoubleNetRoute配置, 剩余扩展路由可通过本节点配置-->
  <DoubleNetRoute>
    <!--opt, object, 双网口路由-->
    <ipAddress>
      <!--opt, string, IP地址-->test
    </ipAddress>
    <subnetMask>
      <!--opt, string, 子网掩码-->test
    </subnetMask>
  </DoubleNetRoute>
</DoubleNetRouteList>
</IPAddress>
<Wireless>
  <!--opt, object, WiFi参数配置-->
  <enabled>
    <!--req, bool, 无线使能开关-->true
  </enabled>
  <wirelessNetworkMode>
    <!--opt, enum, 无线网络工作模式, subType:string, [infrastructure#无固定设施网络,adhoc#点对点网络]-->infrastructure
  </wirelessNetworkMode>
  <channel>
    <!--opt, string, 通道索引-->test
  </channel>
  <ssid>
    <!--opt, string, 服务集标识-->test
  </ssid>
</wmmEnabled>

```

```
<!--opt, bool, 无线网多媒体使能-->true
</wmmEnabled>
<WirelessSecurity>
  <!--opt, object, 无线安全-->
  <securityMode>
    <!--opt, enum, 安全模式, subType:string, [disabLe#关闭,WEP#有线等效保密,WPA-personal#WPA个人安全接入,WPA2-personal#WPA2个人安全接入,WPA-
RADIUS#WPA+RADIUS服务器,WPA-enterprise#WPA企业安全接入,WPA2-enterprise#WPA2企业安全接入]-->disabLe
  </securityMode>
  <WEP>
    <!--opt, object, 有线等效保密, dep:or,{$.NetworkInterface.Wireless.WirelessSecurity.securityMode,eq,WEP}-->
    <authenticationType>
      <!--req, enum, 认证类型, subType:string, [open#开启验证,sharedkey#共享密钥,auto#自动验证]-->open
    </authenticationType>
    <defaultTransmitKeyIndex>
      <!--req, int, 默认传输密钥个数-->1
    </defaultTransmitKeyIndex>
    <wepKeyLength>
      <!--opt, int, 密钥长度, range:[64,128]-->1
    </wepKeyLength>
    <EncryptionKeyList>
      <!--opt, array, 密钥列表, subType:object-->
      <encryptionKey>
        <!--req, string, 密钥-->test
      </encryptionKey>
    </EncryptionKeyList>
  </WEP>
  <WPA>
    <!--opt, object, WPA个人安全接入, dep:or,{$.NetworkInterface.Wireless.WirelessSecurity.securityMode,eq,WPA}-->
    <algorithmType>
      <!--req, enum, 算法类型, subType:string, [TKIP#TKIP,AES#AES,TKIP/AES#TKIP/AES]-->TKIP
    </algorithmType>
    <sharedKey>
      <!--req, string, 共享密钥-->test
    </sharedKey>
    <wpaKeyLength>
      <!--req, int, 密钥长度, range:[8,63]-->1
    </wpaKeyLength>
  </WPA>
</WirelessSecurity>
<workScene>
  <!--opt, enum, 工作场景模式, subType:string, [computerRoom#机房端,monitorTerminal#摄像机端,centralTerminal#中心端]-->computerRoom
</workScene>
<protocol>
  <!--opt, enum, 协议类型, subType:string, [802.11ac#802.11ac]-->802.11ac
</protocol>
<protocolRealTime>
  <!--opt, string, 实时模式-->test
</protocolRealTime>
<hideSsid>
  <!--opt, bool, 隐藏SSID-->true
</hideSsid>
<ChannelConfig>
  <!--opt, object, 信道配置-->
  <width>
    <!--opt, enum, 带宽, subType:string, [auto#auto,20#20,40#40,80#80]-->auto
  </width>
  <autoWidth>
    <!--ro, opt, string, 带宽-->test
  </autoWidth>
  <channel>
    <!--opt, string, 信道-->test
  </channel>
  <autoChannel>
    <!--ro, opt, string, 自动信道-->test
  </autoChannel>
  <transmitPower>
    <!--opt, enum, 功率, subType:int, [9#9,12#12,15#15,18#18,21#21,24#24,27#27]-->9
  </transmitPower>
  <transmitPowerRealTime>
    <!--opt, int, 实时功率-->1
  </transmitPowerRealTime>
  <countryID>
    <!--opt, int, 国家码-->1
  </countryID>
</ChannelConfig>
<rate>
  <!--opt, enum, 频率, subType:string, [2.4GHz#2.4GHz,5GHz#5GHz,auto#auto]-->2.4GHz
</rate>
<dialUpGroupNo>
  <!--opt, string, SSID拨码组号-->test
</dialUpGroupNo>
<baseNoisevalue>
  <!--ro, opt, int, 底噪值, step:1-->1
</baseNoisevalue>
</Wireless>
<Discovery>
  <!--opt, object, 网络发现配置-->
  <UPnP>
    <!--req, object, 通用即插即用-->
    <enabled>
      <!--req, bool, 通用即插即用协议使能-->true
    </enabled>
  </UPnP>
```

```
<Zeroconf>
  <!--opt, object, Zeroconf协议-->
  <enabled>
    <!--req, bool, Zeroconf协议使能-->true
  </enabled>
</Zeroconf>
</Discovery>
<Link>
  <!--opt, object, 设备单个网卡连接信息-->
  <MACAddress>
    <!--req, string, MAC地址-->test
  </MACAddress>
  <autoNegotiation>
    <!--req, bool, 自适应使能-->true
  </autoNegotiation>
  <speed>
    <!--req, enum, 链接速度, subType:int, [10#10M,100#100M,1000#1000M]-->10
  </speed>
  <duplex>
    <!--req, enum, 双工模式, subType:string, [half#half,full#full]-->half
  </duplex>
  <MTU>
    <!--req, int, MTU-->1
  </MTU>
  <linkMode>
    <!--opt, enum, 连接模式, subType:string, [optical#光口模式,electrical#电口模式]-->optical
  </linkMode>
</Link>
<defaultConnection>
  <!--opt, bool, 是否为默认网络连接, desc:当存在多个网卡时需要改节点-->true
</defaultConnection>
<ActiveMulticast>
  <!--opt, object, 主动多播-->
  <enabled>
    <!--req, bool, 使能开关-->true
  </enabled>
  <streamID>
    <!--req, enum, 码流索引, subType:string, [main#main]-->main
  </streamID>
  <ipV4Address>
    <!--opt, string, IPV4地址-->test
  </ipV4Address>
  <ipV6Address>
    <!--opt, string, IPV6地址-->test
  </ipV6Address>
  <port>
    <!--opt, int, 端口号-->1
  </port>
</ActiveMulticast>
<macAddress>
  <!--opt, string, 物理地址-->test
</macAddress>
<EthernetPortList>
  <!--opt, array, 以太网端口信息列表, subType:object-->
  <EthernetPort>
    <!--opt, object, 以太网端口信息-->
    <id>
      <!--req, int, 端口索引, range:[1,4]-->1
    </id>
    <MACAddress>
      <!--req, string, 物理地址-->test
    </MACAddress>
    <status>
      <!--opt, enum, 连接状态, subType:string, [connected#已连接,disconnect#已断开]-->connected
    </status>
    <speed>
      <!--req, enum, 速率, subType:string, [10#10M,100#100M,1000#1000M,10000#10000M]-->10
    </speed>
  </EthernetPort>
</EthernetPortList>
<Extensions>
  <!--opt, object, 扩展信息-->
  <NetworkCardType>
    <!--opt, enum, 网卡类型, subType:string, [normal#网卡,internal#内网卡,external#外网卡,main#主网卡]-->normal
  </NetworkCardType>
</Extensions>
<interfaceName>
  <!--opt, string, 网卡名称-->test
</interfaceName>
<enabled>
  <!--opt, bool, 网卡使能开关, desc:门禁微校认证6301设备支持对网卡启用、关闭, 启用之后方可进行配置参数, 为兼容各产线, 默认为true. -->true
</enabled>
</NetworkInterface>
```

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 响应消息, attr:version{ro, req, string, 版本}-->
  <requestURL>
    <!--ro, req, string, 请求的URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, 状态码, subType:int, [0#OK,1#OK,2#Device Busy,3#Device Error,4#Invalid Operation,5#Invalid XML Format,6#Invalid XML Content,7#Reboot Required]-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, 状态信息, subType:string, [OK#成功,Device Busy#设备忙,Device Error#设备异常,Invalid Operation#无效的操作,Invalid XML Format#无效的XML格式,Invalid XML Content#无效的XML内容,Reboot#设备重启]-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, 详细错误码英文描述, desc:详细错误码的英文描述-->OK
  </subStatusCode>
</ResponseStatus>
```

69.17.5 获取全部TCPIP参数

Request URL

GET /ISAPI/System/Network/interfaces

查询参数

无

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<NetworkInterfaceList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, array, IP地址参数列表, subType:object, attr:version{req, string, 协议版本}-->
  <NetworkInterface>
    <!--ro, req, object, IP地址参数-->
    <id>
      <!--ro, req, string, 网卡索引-->test
    </id>
    <IPAddress>
      <!--ro, req, object, IP地址参数-->
      <ipVersion>
        <!--ro, req, enum, IP地址版本, subType:string, [v4#IPv4,v6#IPv6,dual#IPv4和IPv6都支持]-->v4
      </ipVersion>
      <addressingType>
        <!--ro, req, enum, 地址类型, subType:string, [static#静态,dynamic#动态,api#自动IP寻址]-->static
      </addressingType>
      <ipAddress>
        <!--ro, opt, string, IPv4地址-->test
      </ipAddress>
      <subnetMask>
        <!--ro, opt, string, IPv4子网掩码-->test
      </subnetMask>
      <ipv6Address>
        <!--ro, opt, string, IPv6地址-->test
      </ipv6Address>
      <bitMask>
        <!--ro, opt, string, IPv6子网掩码-->test
      </bitMask>
      <DefaultGateway>
        <!--ro, opt, object, 默认网关-->
        <ipAddress>
          <!--ro, opt, string, IPv4地址-->test
        </ipAddress>
        <ipv6Address>
          <!--ro, opt, string, IPv6地址-->test
        </ipv6Address>
      </DefaultGateway>
      <PrimaryDNS>
        <!--ro, opt, object, 首选DNS-->
        <ipAddress>
          <!--ro, opt, string, IPv4地址-->test
        </ipAddress>
        <ipv6Address>
          <!--ro, opt, string, IPv6地址-->test
        </ipv6Address>
      </PrimaryDNS>
      <SecondaryDNS>
        <!--ro, opt, object, 备用DNS-->
        <ipAddress>
          <!--ro, opt, string, IPv4地址-->test
        </ipAddress>
      </SecondaryDNS>
    </NetworkInterface>
  </NetworkInterfaceList>
```

```

</ipAddress>
<ipv6Address>
  <!--ro, opt, string, IPV6地址-->test
</ipv6Address>
</SecondaryDNS>
<DNSEnable>
  <!--ro, opt, bool, DNS自动分配-->true
</DNSEnable>
<IPv6Mode>
  <!--ro, opt, object, IPV6模式-->
  <ipv6AddressingType>
    <!--ro, opt, enum, IPV6地址类型, subType:string, [ra#路由公告,manual#手动,dhcp#动态主机配置]-->ra
  </ipv6AddressingType>
  <ipv6AddressList>
    <!--ro, opt, array, IPV6地址列表, subType:object-->
    <v6Address>
      <!--ro, opt, object, IPV6地址-->
      <id>
        <!--ro, opt, string, IPV6地址索引-->test
      </id>
      <type>
        <!--ro, opt, enum, IPV6地址类型, subType:string, [ra#路由公告,manual#手动,dhcp#动态主机配置]-->ra
      </type>
      <address>
        <!--ro, opt, string, IPV6地址-->test
      </address>
      <bitMask>
        <!--ro, opt, string, IPV6子网掩码-->test
      </bitMask>
    </v6Address>
  </ipv6AddressList>
</IPv6Mode>
<DoubleNetRoute>
  <!--ro, opt, object, 双网口路由, desc:双网隔离模式下, 若支持多个路由配置时, 其第一组路由地址通过该节点进行配置-->
  <ipAddress>
    <!--ro, opt, string, IP地址-->test
  </ipAddress>
  <subnetMask>
    <!--ro, opt, string, 子网掩码-->test
  </subnetMask>
</DoubleNetRoute>
<LANConfigMode>
  <!--ro, opt, enum, LAN配置模式, subType:string, [manual#手动,auto#自动], desc:路由器进行配置的时候需要选择使用自动还是手动模式, ipv4的自动模式下就是设备默认的LAN口所在网卡的IP和子网掩码信息, 手动模式下, 就需要配置相关的LAN所在的网卡信息, LAN网卡信息需要和DHCP的网段信息相互校验, 保证其在同一个网段-->manual
</LANConfigMode>
<DoubleNetRouteList>
  <!--ro, opt, array, 双网口路由列表扩展, subType:object, desc:双网隔离模式下, 支持最多四个路由配置;第一个路由器地址需通过DoubleNetRoute配置, 剩余扩展路由可通过本节点配置-->
  <DoubleNetRoute>
    <!--ro, opt, object, 双网口路由-->
    <ipAddress>
      <!--ro, opt, string, IP地址-->test
    </ipAddress>
    <subnetMask>
      <!--ro, opt, string, 子网掩码-->test
    </subnetMask>
  </DoubleNetRoute>
</DoubleNetRouteList>
</IPAddress>
<Wireless>
  <!--ro, opt, object, WiFi参数配置-->
  <enabled>
    <!--ro, req, bool, 无线使能开关-->true
  </enabled>
  <wirelessNetworkMode>
    <!--ro, opt, enum, 无线网络工作模式, subType:string, [infrastructure#无固定设施网络,adhoc#点对点网络]-->infrastructure
  </wirelessNetworkMode>
  <channel>
    <!--ro, opt, string, 通道索引-->test
  </channel>
  <ssid>
    <!--ro, opt, string, 服务集标识-->test
  </ssid>
  <wmmEnabled>
    <!--ro, opt, bool, 无线网多媒体使能-->true
  </wmmEnabled>
  <WirelessSecurity>
    <!--ro, opt, object, 无线安全-->
    <securityMode>
      <!--ro, opt, enum, 安全模式, subType:string, [disable#关闭,WEP#有线等效保密,WPA-personal#WPA个人安全接入,WPA2-personal#WPA2个人安全接入,WPA-RADIUS#WPA+RADIUS服务器,WPA-enterprise#WPA企业安全接入,WPA2-enterprise#WPA2企业安全接入]-->disable
    </securityMode>
    <WEP>
      <!--ro, opt, object, 有线等效保密, dep:or,{$.NetworkInterface.Wireless.WirelessSecurity.securityMode,eq,WEP}-->
      <authenticationType>
        <!--ro, req, enum, 认证类型, subType:string, [open#开启验证,sharedkey#共享密钥,auto#自动验证]-->open
      </authenticationType>
      <defaultTransmitKeyIndex>
        <!--ro, req, int, 默认传输密钥个数-->1
      </defaultTransmitKeyIndex>
      <wepKeyLength>
        <!--ro, opt, int, 密钥长度, range:[64,128]-->1
      </wepKeyLength>
    </WEP>
  </WirelessSecurity>

```

```

</wpaEncryptionKeyList>
<EncryptionKeyList>
  <!--ro, opt, array, 密钥列表, subType:object-->
  <encryptionKey>
    <!--ro, req, string, 密钥-->test
  </encryptionKey>
</EncryptionKeyList>
</WEP>
<WPA>
  <!--ro, opt, object, WPA 个人安全接入, dep:or,{$.NetworkInterface.Wireless.WirelessSecurity.securityMode,eq,WPA}-->
  <algorithmType>
    <!--ro, req, enum, 算法类型, subType:string, [TKIP#TKIP,AES#AES,TKIP/AES#TKIP/AES]-->TKIP
  </algorithmType>
  <sharedKey>
    <!--ro, req, string, 共享密钥-->test
  </sharedKey>
  <wpaKeyLength>
    <!--ro, req, int, 密钥长度, range:[8,63]-->1
  </wpaKeyLength>
</WPA>
</WirelessSecurity>
<workScene>
  <!--ro, opt, enum, 工作场景模式, subType:string, [computerRoom#机房端,monitorTerminal#摄像终端,centralTerminal#中心端]-->computerRoom
</workScene>
<protocol>
  <!--ro, opt, enum, 协议类型, subType:string, [802.11ac#802.11ac]-->802.11ac
</protocol>
<protocolRealTime>
  <!--ro, opt, string, 实时模式-->test
</protocolRealTime>
<hideSsid>
  <!--ro, opt, bool, 隐藏SSID-->true
</hideSsid>
<ChannelConfig>
  <!--ro, opt, object, 信道配置-->
  <width>
    <!--ro, opt, enum, 带宽, subType:string, [auto#auto,20#20,40#40,80#80]-->auto
  </width>
  <autoWidth>
    <!--ro, opt, string, 带宽-->test
  </autoWidth>
  <channel>
    <!--ro, opt, string, 信道-->test
  </channel>
  <autoChannel>
    <!--ro, opt, string, 自动信道-->test
  </autoChannel>
  <transmitPower>
    <!--ro, opt, enum, 功率, subType:int, [9#9,12#12,15#15,18#18,21#21,24#24,27#27]-->9
  </transmitPower>
  <transmitPowerRealTime>
    <!--ro, opt, int, 实时功率-->1
  </transmitPowerRealTime>
  <countryID>
    <!--ro, opt, int, 国家码-->1
  </countryID>
</ChannelConfig>
<rate>
  <!--ro, opt, enum, 频率, subType:string, [2.4GHz#2.4GHz,5GHz#5GHz,auto#auto]-->2.4GHz
</rate>
<dialUpGroupNo>
  <!--ro, opt, string, SSID拨码组号-->test
</dialUpGroupNo>
<baseNoisevalue>
  <!--ro, opt, int, 底噪值, step:1-->1
</baseNoisevalue>
</Wireless>
<Discovery>
  <!--ro, opt, object, 网络发现配置-->
  <UPnP>
    <!--ro, req, object, 通用即插即用-->
    <enabled>
      <!--ro, req, bool, 通用即用即插协议使能-->true
    </enabled>
  </UPnP>
  <Zeroconf>
    <!--ro, opt, object, Zeroconf协议-->
    <enabled>
      <!--ro, req, bool, Zeroconf协议使能-->true
    </enabled>
  </Zeroconf>
</Discovery>
<Link>
  <!--ro, opt, object, 设备单个网卡连接信息-->
  <MACAddress>
    <!--ro, req, string, MAC地址-->test
  </MACAddress>
  <autoNegotiation>
    <!--ro, req, bool, 自适应使能-->true
  </autoNegotiation>
  <speed>
    <!--ro, req, enum, 链接速度, subType:int, [10#10M,100#100M,1000#1000M]-->10
  </speed>
  <duplex>

```

```

    <!--ro, req, enum, 双工模式, subType:string, [half#half,full#full]-->half
</duplex>
<MTU>
    <!--ro, req, int, MTU-->1
</MTU>
<linkMode>
    <!--ro, opt, enum, 连接模式, subType:string, [optical#光口模式,electrical#电口模式]-->optical
</linkMode>
</Link>
<defaultConnection>
    <!--ro, opt, bool, 是否为默认网络连接, desc:当存在多个网卡时需要改节点-->true
</defaultConnection>
<ActiveMulticast>
    <!--ro, opt, object, 主动多播-->
    <enabled>
        <!--ro, req, bool, 使能开关-->true
    </enabled>
    <streamID>
        <!--ro, req, enum, 码流索引, subType:string, [main#main]-->main
    </streamID>
    <ipV4Address>
        <!--ro, opt, string, IPV4地址-->test
    </ipV4Address>
    <ipV6Address>
        <!--ro, opt, string, IPV6地址-->test
    </ipV6Address>
    <port>
        <!--ro, opt, int, 端口号-->1
    </port>
</ActiveMulticast>
<macAddress>
    <!--ro, opt, string, 物理地址-->test
</macAddress>
<EthernetPortList>
    <!--ro, opt, array, 以太网端口信息列表, subType:object-->
    <EthernetPort>
        <!--ro, opt, object, 以太网端口信息-->
        <id>
            <!--ro, req, int, 端口索引, range:[1,4]-->1
        </id>
        <MACAddress>
            <!--ro, req, string, 物理地址-->test
        </MACAddress>
        <status>
            <!--ro, opt, enum, 连接状态, subType:string, [connected#已连接,disconnect#已断开]-->connected
        </status>
        <speed>
            <!--ro, req, enum, 速率, subType:string, [10#10M,100#100M,1000#1000M,10000#10000M]-->10
        </speed>
    </EthernetPort>
</EthernetPortList>
<Extensions>
    <!--ro, opt, object, 扩展信息-->
    <NetworkCardType>
        <!--ro, opt, enum, 网卡类型, subType:string, [normal#网卡,internal#内网卡,external#外网卡,main#主网卡]-->normal
    </NetworkCardType>
</Extensions>
<interfaceName>
    <!--ro, opt, string, 网卡名称-->test
</interfaceName>
<enabled>
    <!--ro, opt, bool, 网卡使能开关, desc:门禁微校认证6301设备支持对网卡启用、关闭, 启用之后方可进行配置参数, 为兼容各产线, 默认为true.-->true
</enabled>
</NetworkInterface>
</NetworkInterfaceList>

```

69.17.6 获取单个TCPIP参数配置能力

Request URL

GET /ISAPI/System/Network/interfaces/<interfaceID>/capabilities

查询参数

参数名称	参数类型	描述
interfaceID	string	

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>
<NetworkInterface xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 网络, attr:version{req, string, 协议版本}-->
  <id>
    <!--ro, req, string, id号-->1
  </id>
  <IPAddress>
    <!--ro, req, object, ip地址-->
    <ipVersion opt="v4,v6,dual">
      <!--ro, req, enum, ip版本, subType:string, [v4#v4,v6#v6,dual#dual], attr:opt{req, string, 取值范围}-->v4
    </ipVersion>
    <addressingType opt="static,dynamic,apipa">
      <!--ro, req, enum, 地址类型, subType:string, [static#静态,dynamic#动态,apipa#apipa], attr:opt{req, string, 取值范围}-->static
    </addressingType>
    <ipAddress>
      <!--ro, opt, string, ip地址-->test
    </ipAddress>
    <subnetMask>
      <!--ro, opt, string, 子网掩码-->test
    </subnetMask>
    <DefaultGateway>
      <!--ro, opt, object, 默认网关-->
      <ipAddress>
        <!--ro, opt, string, ip地址-->test
      </ipAddress>
    </DefaultGateway>
    <PrimaryDNS>
      <!--ro, opt, object, 初级DNS-->
      <ipAddress>
        <!--ro, opt, string, ip地址-->test
      </ipAddress>
    </PrimaryDNS>
    <SecondaryDNS>
      <!--ro, opt, object, 辅助DNS-->
      <ipAddress>
        <!--ro, opt, string, ip地址-->test
      </ipAddress>
    </SecondaryDNS>
    <DNSEnable opt="true,false">
      <!--ro, opt, bool, DNS自动分配使能, attr:opt{req, string, 取值范围}-->true
    </DNSEnable>
  </IPAddress>
  <Link>
    <!--ro, opt, object, 连接-->
    <MACAddress>
      <!--ro, req, string, MAC地址-->test
    </MACAddress>
    <autoNegotiation opt="true,false">
      <!--ro, req, bool, 自动协商, attr:opt{req, string, 取值范围}-->true
    </autoNegotiation>
    <speed opt="10,100,1000">
      <!--ro, req, int, 速度, attr:opt{req, string, 取值范围}-->1
    </speed>
    <duplex opt="half,full">
      <!--ro, req, enum, 复式, subType:string, [half#一半,full#完整], attr:opt{req, string, 取值范围}-->half
    </duplex>
    <MTU min="1" max="10">
      <!--ro, req, int, 最大传输单元, attr:min{req, int, 最小值},max{req, int, 最大值}-->1
    </MTU>
  </Link>
  <macAddress min="1" max="10">
    <!--ro, opt, string, mac地址, attr:min{req, int, 最小值},max{req, int, 最大值}-->test
  </macAddress>
</NetworkInterface>
```

69.17.7 配置设备单个网卡连接信息

Request URL

PUT /ISAPI/System/Network/interfaces/<interfaceID>/link

查询参数

参数名称	参数类型	描述
interfaceID	string	

请求报文

```
<?xml version="1.0" encoding="UTF-8"?>

<Link xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, 设备单个网卡连接信息, attr:version{req, string, 协议版本}-->
  <MACAddress>
    <!--req, string, MAC地址-->test
  </MACAddress>
  <autoNegotiation>
    <!--req, bool, 自适应使能-->true
  </autoNegotiation>
  <speed>
    <!--opt, enum, 链接速度, subType:int, [10#10M,100#100M,1000#1000M]-->10
  </speed>
  <duplex>
    <!--opt, enum, 双工模式, subType:string, [half#half,full#full]-->half
  </duplex>
  <MTU>
    <!--req, int, MTU-->1
  </MTU>
  <linkMode>
    <!--opt, enum, 连接模式, subType:string, [optical#光口模式,electrical#电口模式]-->optical
  </linkMode>
</Link>
```

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 响应消息, attr:version{ro, req, string, 版本}-->
  <requestURL>
    <!--ro, req, string, 请求的URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, 状态码, subType:int, [0#OK,1#OK,2#Device Busy,3#Device Error,4#Invalid Operation,5#Invalid XML Format,6#Invalid XML Content,7#Reboot Required]-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, 状态信息, subType:string, [OK#成功,Device Busy#设备忙,Device Error#设备异常,Invalid Operation#无效的操作,Invalid XML Format#无效的XML格式,Invalid XML Content#无效的XML内容,Reboot#设备重启]-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, 详细错误码英文描述, desc: 详细错误码的英文描述-->OK
  </subStatusCode>
</ResponseStatus>
```

69.17.8 获取设备单个网卡连接信息

Request URL

GET /ISAPI/System/Network/interfaces/<interfaceID>/link

查询参数

参数名称	参数类型	描述
interfaceID	string	

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<Link xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 设备单个网卡连接信息, attr:version{req, string, 协议版本}-->
  <MACAddress>
    <!--ro, req, string, MAC地址-->test
  </MACAddress>
  <autoNegotiation>
    <!--ro, req, bool, 自适应使能-->true
  </autoNegotiation>
  <speed>
    <!--ro, opt, enum, 链接速度, subType:int, [10#10M,100#100M,1000#1000M]-->10
  </speed>
  <duplex>
    <!--ro, opt, enum, 双工模式, subType:string, [half#half,full#full]-->half
  </duplex>
  <MTU>
    <!--ro, req, int, MTU-->1
  </MTU>
  <linkMode>
    <!--ro, opt, enum, 连接模式, subType:string, [optical#光口模式,electrical#电口模式]-->optical
  </linkMode>
</Link>
```

69.18 相机的通用架设标定

69.18.1 获取相机架设结构参数能力

Request URL

GET /ISAPI/System/setupStructure/capabilities?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "tilt": {
    /*ro, opt, object, 倾斜角度*/
    "@opt": [0, 15]
    /*ro, req, array, 取值范围, subType:int*/
  }
}
```

69.18.2 配置相机架设结构参数

Request URL

PUT /ISAPI/System/setupStructure?format=json

查询参数

无

请求报文

```
{
  "tilt": 0
  /*opt, enum, 倾斜角度, subType:int, [0#0度,15#15度]*/
}
```

响应报文

```
{
  "statusCode": 1,
  /*ro, opt, int, 状态码, desc:无法用1表示时（1表示成功且无特殊状态）必须返回*/
  "statusString": "OK",
  /*ro, opt, string, 状态描述, range:[1,64], desc:无法用OK表示时（OK表示成功且无特殊状态）必须返回*/
  "subStatusCode": "ok",
  /*ro, opt, string, 子状态码, range:[1,64], desc:无法用ok表示（ok表示成功且无特殊状态）必须返回*/
  "errorCode": 1,
  /*ro, opt, int, 错误码, desc:当statusCode不为1时，与subStatusCode对应*/
  "errorMsg": "ok"
  /*ro, opt, string, 错误信息, desc:当statusCode不为1时，必须返回,解释信息在协议约束中，允许设备在后续的版本迭代中，进行优化丰富提升（不限制死）*/
}
```

69.18.3 获取相机架设结构参数

Request URL

GET /ISAPI/System/setupStructure?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "tilt": 0
  /*ro, opt, enum, 倾斜角度, subType:int, [0#0度,15#15度]*/
}
```

69.19 设备接入协议安全配置

69.19.1 获取安全配置能力

Request URL

GET /ISAPI/Security/CommuMode/capabilities?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "CommuMode": {
    /*ro, req, object, 私有协议安全模式*/
    "mode": {
      /*ro, req, object, 私有协议安全模式等级*/
      "@opt": "compatibilityMode,safeMode"
      /*ro, req, string, 可选项, desc:compatibilityMode兼容模式,safeMode安全模式*/
    }
  }
}
```

69.19.2 获取安全配置

Request URL

GET /ISAPI/Security/CommuMode?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "CommuMode": {
    /*ro, req, object, 私有协议安全模式*/
    "mode": "safeMode"
    /*ro, req, enum, 私有协议安全模式等级, subType:string, [compatibilityMode#兼容模式,safeMode#安全模式]*/
  }
}
```

69.19.3 配置安全参数

Request URL

PUT /ISAPI/Security/CommuMode?format=json

查询参数

无

请求报文

```
{
  "CommuMode": {
    /*req, object, 私有协议安全模式*/
    "mode": "safeMode"
    /*req, enum, 私有协议安全模式等级, subType:string, [compatibilityMode#兼容模式,safeMode#安全模式]*/
  }
}
```

响应报文

```
{
  "statusCode": 1,
  /*ro, opt, int, 状态码, desc:无法用1表示时（1表示成功且无特殊状态）必须返回*/
  "statusString": "OK",
  /*ro, opt, string, 状态描述, range:[1,64], desc:无法用OK表示时（OK表示成功且无特殊状态）必须返回*/
  "subStatusCode": "ok",
  /*ro, opt, string, 子状态码, range:[1,64], desc:无法用ok表示（ok表示成功且无特殊状态）必须返回*/
  "errorCode": 1,
  /*ro, opt, int, 错误码, desc:当statusCode不为1时，与subStatusCode对应*/
  "errorMsg": "ok"
  /*ro, opt, string, 错误信息, desc:当statusCode不为1时，必须返回,解释信息在协议约束中，允许设备在后续的版本迭代中，进行优化丰富提升（不限制死）*/
}
```

69.20 升级管理（ZF）

69.20.1 设备升级

Request URL

POST /ISAPI/System/updateFirmware?type=<type>&moduleAddress=<moduleAddress>&id=<indexID>

查询参数

参数名称	参数类型	描述
type	string	设备类型：1、报警设备：keypad-键盘，wirelessRecv-无线接收模块，R3WirelessRecv-R3无线接收模块，RXWirelessRecv-RX无线接收模块，wiredZone-有线防区模块，sirenIndoor-警号[室内]、sirenOutdoor-警号[室外]，sirenAudio-警号[语音对讲]，repeater-中继器，single-单路输出模块（弱电继电器），wallSwitch-墙装开关，smartPlug-智能插座，detector-探测器，transmitter-Transmitter外设，remoteCtrl-遥控器，networkZoneModule-网络防区模块，电锁-electricLock，加热（升温）模块-heatingModule，wiredOutput-有线输出模块。2、门禁设备：cardReader-485读卡器、FPMModule-指纹模组、securityModule-安全模块、extendModule-(IO)扩展模块、channelController-通道控制器、IRModule-红外模块、lampModule-指示灯模块、elevatorController-梯控分控制器、FPAlgorithmProgram-读卡器指纹算法程序、MCU-单片机程序升级、temperatureModule-测温模块、faceModule-人脸识别模块、touchScreenModule-触摸屏模块、bluetoothModule-蓝牙模块、netReader-网络读卡器、userInterfaceBoard-用户接口板、subPermissionController-从权限控制器，QRCodeModule-二维码模组。3、智能柜设备：lockControlBoard-锁控板（格口箱的格口锁控制板，id不下发表示升级所有锁控板）。4、接收卡-ledReceiverCard5、子系统-subsystem（媒体网关新增）6、雷视设备：warningScreenFirmware-预警屏固件升级，warningScreenFont-预警屏字库升级，warningScreenVoice-预警屏语音升级
moduleAddress	string	模块地址：一代混合报警主机专用模块地址可通过调用获取单个或全部防区、外设参数的协议来得到。举例：键盘：GET /ISAPI/SecurityCP/Configuration/outputs?format=json继电器：GET /ISAPI/SecurityCP/Configuration/outputs?format=json警号：GET /ISAPI/SecurityCP/Configuration/wirelessSiren?format=json输出模块：GET /ISAPI/SecurityCP/Configuration/outputModules?format=json扩展模块：GET /ISAPI/SecurityCP/Configuration/extensionModule?format=json防区：GET /ISAPI/SecurityCP/Configuration/zones?format=json
indexID	string	设备编号：1、二代混合主机与无线主机保持一致，用id进行升级2、门禁设备采用id升级3、智能柜设备采用id升级

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 响应消息, attr:version{ro, req, string, 版本}-->
  <requestURL>
    <!--ro, req, string, 请求的URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, 状态码, subType:int, [0#OK,1#OK,2#Device Busy,3#Device Error,4#Invalid Operation,5#Invalid XML Format,6#Invalid XML Content,7#Reboot Required]-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, 状态信息, subType:string, [OK#成功,Device Busy#设备忙,Device Error#设备异常,Invalid Operation#无效的操作,Invalid XML Format#无效的XML格式,Invalid XML Content#无效的XML内容,Reboot#设备重启]-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, 详细错误码英文描述, desc: 详细错误码的英文描述-->OK
  </subStatusCode>
</ResponseStatus>
```

状态码	状态描述	子状态码	错误码	描述
3	Device Error	theDeviceIsOffline	0x30008007	设备离线
3	Device Error	deviceIsForwardedByTheRepeater	0x30008008	设备通信被中继器转发
3	Device Error	lowBattery	0x30008009	电量较低
3	Device Error	theSignalIsUnstable	0x3000800A	当前信号不稳定

69.20.2 获取设备升级的进度

Request URL

GET /ISAPI/System/upgradeStatus?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "requestURL": "/ISAPI/Streaming/channels/1",
  /*ro, opt, string, 请求URL*/
  "statusCode": "test",
  /*ro, req, string, 状态码*/
  "statusString": "test",
  /*ro, req, string, 状态描述*/
  "subStatusCode": "test",
  /*ro, req, string, 子状态码*/
  "errorCode": 1,
  /*ro, opt, int, 当statusCode不为1时*/
  "errorMsg": "ok",
  /*ro, opt, string, 当statusCode不为1时*/
  "upgrading": "TRUE",
  /*ro, opt, string, 是否正在升级*/
  "percent": 22,
  /*ro, opt, int, 升级进度*/
  "idList": [
    /*ro, opt, array, id列表, subType:object*/
    {
      "id": "test",
      /*ro, req, string, 分析单元ID, desc:若支持type字段, 则此处填写格式如"1", 标识指定的外设编号*/
      "type": "electricGenie",
      /*ro, opt, enum, 外设类型, subType:string, [electricGenie#用电精灵], desc:/ISAPI/System/AcsUpdate/capabilities的能力报文中支持batchType时, 支持
      此节点*/
      "percent": 22,
      /*ro, opt, int, 升级进度*/
      "status": "backingUp"
      /*ro, opt, enum, 升级状态, subType:string, [backingUp#升级备份中,replacing#升级替换中,failed#升级失败,successful#升级完成,rebooting#升级完成(重
      启中) */
    }
  ]
}
```

69.20.3 获取外设模块升级能力集

Request URL

GET /ISAPI/System/AcsUpdate/capabilities

查询参数

无

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<AcsUpdate xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 外设模块升级能力集, attr:version{req, string, 协议版本}-->
  <type
```

```
opt="cardReader,FPMModule,securityModule,extendModule,channelController,IRModule,lampModule,elevatorController,FPAlgorithmProgram,uboot,keypad,wirelessRecv,wiredZone,sirenIndoor,sirenOutdoor,sirenAudio,repeater,bluetoothModule,single,wallSwitch,smartPlug,detector,transmitter,remoteCtrl,subModule,dispModule,netReader,lockControlBoard,networkZoneModule,userInterfaceBoard,subPermissionController,electricLock,heatingModule,RS485Module,QRCModule,R3WirelessRecv,RXWirelessRecv,wiredOutput,electricGenie">
```

```
<!--ro, opt, enum, 升级类型（外设类型），subType:string, [cardReader#485读卡器,FPMModule#指纹模组,securityModule#安全模块,extendModule#(IO)扩展模块,channelController#通道控制器,IRModule#红外模块,LampModule#指示灯模块,elevatorController#梯控分控制器,FPAlgorithmProgram#读卡器指纹算法程序,uboot#uboot升级,keypad#键盘,wirelessRecv#无线接收模块,wiredZone#有线防区模块,sirenIndoor#警号室内,sirenOutdoor#警号室外,sirenAudio#警号语音对讲,repeater#中继器,bluetoothModule#蓝牙模块,single#单路输出模块,wallSwitch#墙装开关,smartPlug#智能插座,detector#探测器,transmitter#Transmitter外设,remoteCtrl#遥控器,subModule#子模块,dispModule#显示模块,netReader#网络读卡器,MCU#单片机程序升级,faceModule#人脸模块,touchScreenModule#触摸屏模块,temperatureModule#测温模块,lockControlBoard#锁控板,networkZoneModule#网络防区模块,userInterfaceBoard#用户接口板,subPermissionController#从权限控制器,electricLock#电锁,heatingModule#加热模块,RS485Module#RS485模块,QRCModule#二维码模组,electricGenie#用电精灵], attr:opt{req, string, 取值范围}-->cardReader</type>
```

```
<batchType opt="electricGenie">
```

```
<!--ro, opt, enum, 批量升级类型（外设类型），subType:string, [electricGenie#用电精灵], attr:opt{req, string, 取值范围}, desc:支持批量升级的外设类型，支持该字段后，可通过：
```

- 1./ISAPI/System/updateFirmware?type=<type>下发指定外设类型的升级包
- 2./ISAPI/System/upgradeStatus/startUpgrade?format=json指定外设类型的哪些编号进行升级
- 3./ISAPI/System/upgradeStatus?format=json获取外设的升级进度-->electricGenie

```
</batchType>
```

```
<cardReaderNo min="1" max="10" opt="1,4">
```

```
<!--ro, opt, int, 读卡器编号范围（当type字段返回opt中存在cardReader时有效），attr:min{opt, int, 最小值, step:1},max{opt, int, 最大值, step:1},opt{opt, string, 闸机产品仅支持编号1和4, desc:闸机产品min和max属性不需返回}-->1
```

```
</cardReaderNo>
```

```
<FPMModuleNo min="1" max="10">
```

```
<!--ro, opt, int, 指纹模组编号范围（当type字段返回opt中存在FPMModule时有效），attr:min{req, int, 最小值},max{req, int, 最大值}-->1
```

```
</FPMModuleNo>
```

```
<securityModuleNo min="1" max="10">
```

```
<!--ro, opt, int, 安全模块编号范围（当type字段返回opt中存在securityModule时有效），attr:min{req, int, 最小值},max{req, int, 最大值}-->1
```

```
</securityModuleNo>
```

```
<extendModuleNo min="1" max="10">
```

```
<!--ro, opt, int, (IO)扩展模块编号范围（当type字段返回opt中存在extendModule时有效），attr:min{req, int, 最小值},max{req, int, 最大值}-->1
```

```
</extendModuleNo>
```

```
<channelControllerNo min="1" max="10">
```

```
<!--ro, opt, int, 通道控制器编号范围（当type字段返回opt中存在channelController时有效），attr:min{req, int, 最小值},max{req, int, 最大值}-->1
```

```
</channelControllerNo>
```

```
<IRModuleNo min="1" max="10">
```

```
<!--ro, opt, int, 红外模块编号范围（当type字段返回opt中存在IRModule时有效），attr:min{req, int, 最小值},max{req, int, 最大值}-->1
```

```
</IRModuleNo>
```

```
<lampModuleNo min="1" max="10">
```

```
<!--ro, opt, int, 指示灯模块编号范围（当type字段返回opt中存在LampModule时有效），attr:min{req, int, 最小值},max{req, int, 最大值}-->1
```

```
</lampModuleNo>
```

```
<elevatorControllerNo min="1" max="10">
```

```
<!--ro, opt, int, 梯控分控制器编号范围（当type字段返回opt中存在elevatorController时有效），attr:min{req, int, 最小值},max{req, int, 最大值}-->1
```

```
</elevatorControllerNo>
```

```
<FPAlgorithmProgramNo min="1" max="10">
```

```
<!--ro, opt, int, 读卡器指纹算法程序编号范围（当type字段返回opt中存在FPAlgorithmProgram时有效），attr:min{req, int, 最小值},max{req, int, 最大值}-->1
```

```
</FPAlgorithmProgramNo>
```

```
<faceModuleNo min="1" max="10">
```

```
<!--ro, opt, int, 人脸识别模块编号范围（当type字段返回opt中存在faceModule时有效），attr:min{req, int, 最小值},max{req, int, 最大值}-->1
```

```
</faceModuleNo>
```

```
<touchScreenModuleNo min="1" max="10">
```

```
<!--ro, opt, int, 触摸屏模块编号范围（当type字段返回opt中存在touchScreenModule时有效），attr:min{req, int, 最小值},max{req, int, 最大值}-->1
```

```
</touchScreenModuleNo>
```

```
<temperatureModuleNo min="1" max="10">
```

```
<!--ro, opt, int, 测温模块编号范围（当type字段返回opt中存在temperatureModule时有效），attr:min{req, int, 最小值},max{req, int, 最大值}-->1
```

```
</temperatureModuleNo>
```

```
<keypadAddress opt="1,3,5">
```

```
<!--ro, opt, int, 键盘模块地址范围（当type字段返回opt中存在keypad时有效），attr:opt{req, string, 取值范围}-->1
```

```
</keypadAddress>
```

```
<wirelessRecvAddress opt="1,3,5">
```

```
<!--ro, opt, int, 无线接收模块范围（当type字段返回opt中存在wirelessRecv时有效），attr:opt{req, string, 取值范围}-->1
```

```
</wirelessRecvAddress>
```

```
<wiredZoneAddress opt="1,3,5">
```

```
<!--ro, opt, int, 有线防区模块范围（当type字段返回opt中存在wiredZone时有效），attr:opt{req, string, 取值范围}-->1
```

```
</wiredZoneAddress>
```

```
<sirenIndoorNo opt="1,3,5">
```

```
<!--ro, opt, int, 警号[室内]编号范围（当type字段返回opt中存在sirenIndoor时有效），attr:opt{req, string, 取值范围}-->1
```

```
</sirenIndoorNo>
```

```
<sirenOutdoorNo opt="1,3,5">
```

```
<!--ro, opt, int, 警号[室外]编号范围（当type字段返回opt中存在sirenOutdoor时有效），attr:opt{req, string, 取值范围}-->1
```

```
</sirenOutdoorNo>
```

```
<repeaterNo opt="1,3,5">
```

```
<!--ro, opt, int, 中继器编号范围（当type字段返回opt中存在repeater时有效），attr:opt{req, string, 取值范围}-->1
```

```
</repeaterNo>
```

```
<subModuleNo opt="1,3,5">
```

```
<!--ro, opt, int, 子模块编号范围（当type字段返回opt中存在subModule时有效），attr:opt{req, string, 取值范围}-->1
```

```
</subModuleNo>
```

```
<dispModuleNo opt="1,3,5">
```

```
<!--ro, opt, int, 显示模块编号范围（当type字段返回opt中存在dispModule时有效），attr:opt{req, string, 取值范围}-->1
```

```
</dispModuleNo>
```

```
<single opt="1,3,5">
```

```
<!--ro, opt, int, 单路输出模块（弱电继电器）编号范围（当type字段返回opt中存在single时有效），attr:opt{req, string, 取值范围}-->1
```

```
</single>
```

```
<wallSwitch opt="1,3,5">
```

```
<!--ro, opt, int, 墙装开关编号范围（当type字段返回opt中存在wallSwitch时有效），attr:opt{req, string, 取值范围}-->1
```

```
</wallSwitch>
```

```
<smartPlug opt="1,3,5">
```

```
<!--ro, opt, int, 智能插座编号范围（当type字段返回opt中存在smartPlug时有效），attr:opt{req, string, 取值范围}-->1
```

```
</smartPlug>
```

```
<zoneNo opt="1,3,5">
```

```
<!--ro, opt, int, 防区编号范围（当type字段返回opt中存在detector时有效），attr:opt{req, string, 取值范围}, desc:通过防区号找到对应的探测器。-->1
```

```
</zoneNo>
```

```
<transmitterNo opt="1,3,5">
```

```
<!--ro, opt, int, Transmitter编号范围（当type字段返回opt中存在transmitter时有效），attr:opt{req, string, 取值范围}-->1
</transmitterNo>
<remoteCtrlNo opt="1,3,5">
  <!--ro, opt, int, 遥控器编号范围（当type字段返回opt中存在remoteCtrl时有效），attr:opt{req, string, 取值范围}-->1
</remoteCtrlNo>
<netReaderNo min="1" max="10">
  <!--ro, opt, int, 网络读卡器编号范围（当type字段返回opt中存在netReader时有效），attr:min{req, int, 最小值},max{req, int, 最大值}-->1
</netReaderNo>
<lockControlBoardLNo opt="1,2,3,4,5,6,7,8">
  <!--ro, opt, int, 格口箱左侧编号范围（当type字段返回opt中存在时有效），attr:opt{req, string, 取值范围}-->1
</lockControlBoardLNo>
<lockControlBoardRNo opt="1,2,3,4,5,6,7,8">
  <!--ro, opt, int, 格口箱右侧编号范围（当type字段返回opt中存在时有效），attr:opt{req, string, 取值范围}-->1
</lockControlBoardRNo>
<networkZoneModuleNo opt="1,3,5">
  <!--ro, opt, int, 网络防区模块编号范围（当type字段返回opt中存在networkZoneModule时有效），attr:opt{req, string, 取值范围}-->1
</networkZoneModuleNo>
<userInterfaceBoardNo opt="0,1">
  <!--ro, opt, enum, 用户接口板编号范围, subType:string, [0#主用户接口板,1#从用户接口板], attr:opt{req, string, 取值范围}-->0
</userInterfaceBoardNo>
<electricLockNo opt="1,3,5">
  <!--ro, opt, int, 电锁编号范围（当type字段返回opt中存在electricLock时有效），attr:opt{req, string, 取值范围}-->1
</electricLockNo>
<heatingModuleNo opt="1,3,5">
  <!--ro, opt, int, 加热模块编号范围（当type字段返回opt中存在heatingModule时有效），attr:opt{req, string, 取值范围}, desc:加热模块, 当设备处于低温环境运行时, 加热模块进行工作, 给设备升温。-->1
</heatingModuleNo>
<sirenAudioNo opt="1,3,5">
  <!--ro, opt, int, 警号[对讲]编号范围（当type字段返回opt中存在sirenAudio时有效），attr:opt{req, string, 取值范围}-->1
</sirenAudioNo>
<QRCodeModuleNo min="1" max="10">
  <!--ro, opt, int, 二维码模组编号范围（当type字段返回opt中存在QRCodeModule时有效），attr:min{req, int, 最小值},max{req, int, 最大值}-->1
</QRCodeModuleNo>
<R3WirelessRecvNo min="1" max="54">
  <!--ro, opt, int, R3无线接收模块编号范围（当type字段返回opt中存在R3WireLessRecv时有效），attr:min{req, int, 最小值},max{req, int, 最大值}-->1
</R3WirelessRecvNo>
<RXWirelessRecvNo min="1" max="54">
  <!--ro, opt, int, RX无线接收模块编号范围（当type字段返回opt中存在RXWireLessRecv时有效），attr:min{req, int, 最小值},max{req, int, 最大值}-->1
</RXWirelessRecvNo>
<wiredOutputNo min="1" max="54">
  <!--ro, opt, int, 有线输出模块编号范围（当type字段返回opt中存在wiredOutput时有效），attr:min{req, int, 最小值},max{req, int, 最大值}-->1
</wiredOutputNo>
<electricGenieNo opt="1,2,3,4">
  <!--ro, opt, int, 用电精灵编号范围（当type或batchType字段返回opt中存在electricGenie时有效），attr:opt{req, string, 取值范围}-->1
</electricGenieNo>
</AcsUpdate>
```

69.20.4 获取设备升级状态

Request URL

GET /ISAPI/System/upgradeStatus?type=<Type>

查询参数

参数名称	参数类型	描述
Type	string	usbmic-阵列麦克风风模块

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>
<upgradeStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 升级结果状态, attr:version{req, string, 协议版本}-->
  <upgrading>
    <!--ro, req, bool, 升级状态-->true
  </upgrading>
  <percent>
    <!--ro, req, int, 升级进度, range:[0,100]-->1
  </percent>
</upgradeStatus>
```

69.21 视频编码

69.21.1 获取多通道编码参数

Request URL

GET /ISAPI/Streaming/channels

查询参数

无

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<StreamingChannelList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, array, 通道列表, subType:object, attr:version{opt, string, 协议版本}-->
  <StreamingChannel>
    <!--ro, opt, object, 每个通道的配置信息-->
    <id>
      <!--ro, req, string, 通道ID-->test
    </id>
    <channelName>
      <!--ro, req, string, 通道名称-->test
    </channelName>
    <enabled>
      <!--ro, req, bool, 使能-->true
    </enabled>
    <Transport>
      <!--ro, req, object, 传输配置-->
      <rtspPortNo>
        <!--ro, opt, int, rtsp端口号, range:[0,65535]-->554
      </rtspPortNo>
      <maxPacketSize>
        <!--ro, opt, int, 最大包大小-->1
      </maxPacketSize>
      <audioPacketLength>
        <!--ro, opt, int, 音频包长度-->1
      </audioPacketLength>
      <audioInboundPacketLength>
        <!--ro, opt, int, 音频输入包长度-->1
      </audioInboundPacketLength>
      <audioInboundPortNo>
        <!--ro, opt, int, 音频输入端口号-->1
      </audioInboundPortNo>
      <videoSourcePortNo>
        <!--ro, opt, int, 视频源端口号-->1
      </videoSourcePortNo>
      <audioSourcePortNo>
        <!--ro, opt, int, 音频源端口号-->1
      </audioSourcePortNo>
      <ControlProtocolList>
        <!--ro, req, array, 控制协议列表, subType:object-->
        <ControlProtocol>
          <!--ro, opt, object, 控制协议-->
          <streamingTransport>
            <!--ro, req, enum, 码流传输协议, subType:string, [HTTP#HTTP,RTSP#RTSP,SHHTTP#SHHTTP]-->RTSP
          </streamingTransport>
        </ControlProtocol>
      </ControlProtocolList>
      <Unicast>
        <!--ro, opt, object, 单播配置-->
        <enabled>
          <!--ro, req, bool, 使能-->true
        </enabled>
        <interfaceID>
          <!--ro, opt, string, 网卡ID-->test
        </interfaceID>
        <rtpTransportType>
          <!--ro, opt, enum, RTP传输类型, subType:string, [RTP/UDP#RTP over UDP,RTP/TCP#RTP over TCP]-->RTP/UDP
        </rtpTransportType>
      </Unicast>
      <Multicast>
        <!--ro, opt, object, 多播配置-->
        <enabled>
          <!--ro, req, bool, 使能-->true
        </enabled>
        <userTriggerThreshold>
          <!--ro, opt, int, 用户触发间隔（具体含义待确认）-->1
        </userTriggerThreshold>
        <destIPAddress>
          <!--ro, opt, string, ipv4目的地址-->test
        </destIPAddress>
        <videoDestPortNo>
          <!--ro, opt, int, 视频目的端口-->1
        </videoDestPortNo>
        <audioDestPortNo>
          <!--ro, opt, int, 音频目的端口-->1
        </audioDestPortNo>
        <destIPv6Address>
          <!--ro, opt, string, ipv6目的地址-->test
        </destIPv6Address>
      </Multicast>
    </Transport>
  </StreamingChannel>
</StreamingChannelList>
```

```

<ttl>
  <!--ro, opt, int, 域名解析在DNS服务器中存留时间-->1
</ttl>
<activeMulticastEnabled>
  <!--ro, opt, bool, 主动多播使能-->true
</activeMulticastEnabled>
<packagingFormat>
  <!--ro, opt, enum, 封装格式, subType:string, [RTP#RTP,PS#PS]-->RTP
</packagingFormat>
<fecInfo>
  <!--ro, opt, object, FEC数据占用额外带宽百分比-->
  <fecRatio>
    <!--ro, req, int, FEC多播端口, range:[0,100], desc:默认由设备指定端口-->1
  </fecRatio>
  <fecDestPortNo>
    <!--ro, opt, int, FEC目的端口号-->1
  </fecDestPortNo>
</fecInfo>
</Multicast>
<Security>
  <!--ro, opt, object, 安全配置-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <certificateType>
    <!--ro, req, enum, RTSP认证方式, subType:string, [digest#digest,digest/basic#digest/basic]-->digest
  </certificateType>
  <SecurityAlgorithm>
    <!--ro, opt, object, 安全算法-->
    <algorithmType>
      <!--ro, opt, enum, 算法类型, subType:string, [MD5#MD5,SHA256#SHA256,MD5/SHA256#MD5/SHA256]-->MD5
    </algorithmType>
  </SecurityAlgorithm>
</Security>
<SRTPMulticast>
  <!--ro, opt, object, SRTP多播配置-->
  <SRTPVideoDestPortNo>
    <!--ro, opt, int, SRTP视频目的端口号-->1
  </SRTPVideoDestPortNo>
  <SRTPAudioDestPortNo>
    <!--ro, opt, int, SRTP音频目的端口号-->1
  </SRTPAudioDestPortNo>
</SRTPMulticast>
</Transport>
<Video>
  <!--ro, opt, object, 视频配置-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <videoInputChannelID>
    <!--ro, opt, string, 模拟通道ID-->test
  </videoInputChannelID>
  <dynVideoInputChannelID>
    <!--ro, opt, string, 数字通道ID-->test
  </dynVideoInputChannelID>
  <videoCodecType>
    <!--ro, req, enum, 视频编码类型, subType:string,
[UNKNOWN#UNKNOWN,MPEG4#MPEG4,MJPEG#MJPEG,3GP#3GP,H.264#H.264,HK.264#HK.264,MPNG#MPNG,SVAC#SVAC,H.265#H.265,H.264BP#H.264BP,H.264HP#H.264HP,H.264SVC#H.264SVC,
,MJPEG2#MJPEG2]-->H.264
  </videoCodecType>
  <videoScanType>
    <!--ro, opt, enum, 视频扫描类型, subType:string, [progressive#逐行扫描,interlaced#隔行扫描]-->progressive
  </videoScanType>
  <videoResolutionWidth>
    <!--ro, req, int, 分辨率宽-->1
  </videoResolutionWidth>
  <videoResolutionHeight>
    <!--ro, req, int, 分辨率高-->1
  </videoResolutionHeight>
  <videoResolutionName>
    <!--ro, opt, enum, 分辨率名称, subType:string, [3MP#3MP,5MP#5MP,none#无]-->3MP
  </videoResolutionName>
  <videoDiffResolutionList>
    <!--ro, opt, array, 分辨率列表, subType:object-->
    <videoDiffResolution>
      <!--ro, opt, object, 分辨率-->
      <videoCodecType>
        <!--ro, req, enum, 视频编码类型, subType:string, [MPEG4#MPEG4,MJPEG#MJPEG,3GP#3GP,H.264#H.264,HK.264#HK.264,MPNG#MPNG,SVAC#SVAC,H.265#H.265]--
>H.264
      </videoCodecType>
      <videoResolutionWidth>
        <!--ro, req, int, 分辨率宽-->1
      </videoResolutionWidth>
      <videoResolutionHeight>
        <!--ro, req, int, 分辨率高-->1
      </videoResolutionHeight>
    </videoDiffResolution>
  </videoDiffResolutionList>
  <videoPositionX>
    <!--ro, opt, int, 视频x坐标-->1
  </videoPositionX>
  <videoPositionY>
    <!--ro, opt, int, 视频y坐标-->1
  </videoPositionY>

```

```

</videoQualityControlType>
<videoQualityControlType>
  <!--ro, opt, enum, 视频码率类型, subType:string, [CBR#定码率,VBR#变码率]-->CBR
</videoQualityControlType>
<constantBitRate>
  <!--ro, opt, int, 定码率-->1
</constantBitRate>
<fixedQuality>
  <!--ro, opt, enum, 图像质量, subType:int, [1#Lowest,20#Lower,40#Low,60#medium,80#higher,100#highest]-->1
</fixedQuality>
<vbrUpperCap>
  <!--ro, opt, int, 码流上限-->1
</vbrUpperCap>
<vbrLowerCap>
  <!--ro, opt, int, 码流下限-->1
</vbrLowerCap>
<maxFrameRate>
  <!--ro, req, int, 视频帧率, desc: 帧率x100, 如22帧下发2200,1/4帧下发25-->1
</maxFrameRate>
<keyFrameInterval>
  <!--ro, opt, int, I帧间隔, unit:ms, unitType:时间, desc:为转化为毫秒时间单位的值-->1
</keyFrameInterval>
<rotationDegree>
  <!--ro, opt, int, 旋转度数-->1
</rotationDegree>
<mirrorEnabled>
  <!--ro, opt, bool, 镜面使能-->true
</mirrorEnabled>
<snapshotImageType>
  <!--ro, opt, enum, 抓拍图片格式, subType:string, [JPEG#JPEG,GIF#GIF,PNG#PNG]-->JPEG
</snapshotImageType>
<Mpeg4Profile>
  <!--ro, opt, enum, Mpeg4编码复杂度, subType:string, [SP#SP,ASP#ASP]-->SP
</Mpeg4Profile>
<H264Profile>
  <!--ro, opt, enum, H264编码复杂度, subType:string, [Baseline#Baseline,Main#Main,High#High,Extended#Extended]-->Baseline
</H264Profile>
<SVACProfile>
  <!--ro, opt, enum, SVAC编码复杂度, subType:string, [Baseline#Baseline,Main#Main,High#High,Extended#Extended]-->Baseline
</SVACProfile>
<GovLength>
  <!--ro, opt, int, I帧间隔, desc:为帧的实际值-->1
</GovLength>
<SVC>
  <!--ro, opt, object, 可分级视频编码功能-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <SVCMode>
    <!--ro, opt, enum, 模式选择, subType:string, [manual#manual,auto#auto]-->manual
  </SVCMode>
  <protectLevel>
    <!--ro, opt, enum, 保护等级, subType:string, [all#全等级,nuclear#核心等级]-->all
  </protectLevel>
</SVC>
<smoothing>
  <!--ro, opt, int, 码流平滑度[1-100], range:[1,100], desc:1等级表示清晰, 100表示平滑-->1
</smoothing>
<H265Profile>
  <!--ro, opt, enum, H265编码复杂度, subType:string, [Baseline#Baseline,Main#Main,High#High,Extended#Extended]-->Baseline
</H265Profile>
<SmartCodec>
  <!--ro, opt, object, smart编码-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
</SmartCodec>
<vbrAverageCap>
  <!--ro, opt, int, 平均码率, desc:依赖于SmartCodec使能打开-->1
</vbrAverageCap>
<IntelligentInfoDisplayMethod>
  <!--ro, opt, enum, 智能信息显示方式, subType:string, [player#显示,non-player#不显示]-->player
</IntelligentInfoDisplayMethod>
<LBREnabled>
  <!--ro, opt, bool, 低码率使能-->true
</LBREnabled>
<minimumResolutionSupportedBySmartCode>
  <!--ro, opt, string, smart编码(包含smart264和smart265)支持的最小分辨率-->640*512
</minimumResolutionSupportedBySmartCode>
<videoInputType>
  <!--ro, opt, enum, 视频输入方式, subType:string, [SDI#SDI方式,HDMI#HDMI方式,VGA#VGA方式,AUTO#自动方式]-->SDI
</videoInputType>
<BPFrameInterval>
  <!--ro, opt, enum, P帧间隔, subType:string, [BBP#BBP帧,BP#BP帧,P#单P帧,auto#自动(和源一致)], desc:单P帧(2006-08-11 增加单P帧的配置接口, 可以改善实时流延时问题)-->auto
</BPFrameInterval>
<DepthMapEnable>
  <!--ro, opt, bool, 深度图使能, desc:如果开启后, 第二通道的子码流(双目)的视频参数都不能配置, 默认输出 960*540 的深度图-->false
</DepthMapEnable>
</Video>
<Audio>
  <!--ro, opt, object, 音频配置-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>

```

```
</enabled>
<audioInputChannelID>
  <!--ro, req, string, 数字通道ID-->test
</audioInputChannelID>
<audioCompressionType>
  <!--ro, req, enum, 音频编码类型, subType:string,
[UNKNOWN#UNKNOWN,G.711aLaw#G.711aLaw,G.711uLaw#G.711uLaw,G.726#G.726,G.729#G.729,G.729a#G.729a,G.729b#G.729b,PCM#PCM,MP3#MP3,AC3#AC3,AAC#AAC,ADPCM#ADPCM,MP2
L2#MP2L2,G722#G722,G.719#G.719,AAC-LC#AAC-LC,AAC-LD#AAC-LD,Opus#Opus,G.728#G.728,G.722.1#G.722.1,G.722.1.C#G.722.1.C]-->UNKNOWN
</audioCompressionType>
<audioInboundCompressionType>
  <!--ro, opt, enum, 音频输入编码类型, subType:string,
[UNKNOWN#UNKNOWN,G.711aLaw#G.711aLaw,G.711uLaw#G.711uLaw,G.726#G.726,G.729#G.729,G.729a#G.729a,G.729b#G.729b,PCM#PCM,MP3#MP3,AC3#AC3,AAC#AAC,ADPCM#ADPCM,MP2
L2#MP2L2,G722#G722,G.719#G.719,AAC-LC#AAC-LC,AAC-LD#AAC-LD,Opus#Opus,G.728#G.728,G.722.1#G.722.1,G.722.1.C#G.722.1.C]-->UNKNOWN
</audioInboundCompressionType>
<audioBitRate>
  <!--ro, opt, int, 音频编码比特率-->1
</audioBitRate>
<audioSamplingRate>
  <!--ro, opt, float, 音频采样率-->0.0
</audioSamplingRate>
<audioResolution>
  <!--ro, opt, int, 音频码率-->1
</audioResolution>
<VoiceChanger>
  <!--ro, opt, object, 语音转换-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <level>
    <!--ro, req, int, 级别, range:[-12,12]-->1
  </level>
</VoiceChanger>
<audioInputType>
  <!--ro, opt, enum, 音频输入方式, subType:string, [AudioIn#表示同时支持VGA和HDMI方式,HDMI#(HDMI方式)]-->AudioIn
</audioInputType>
</Audio>
<enableCABAC>
  <!--ro, opt, bool, 码流压缩性能提升使能-->true
</enableCABAC>
<subStreamRecStatus>
  <!--ro, opt, bool, 子码流录像状态-->true
</subStreamRecStatus>
<isSupportRefreshFrame>
  <!--ro, opt, bool, 是否支持重刷帧, desc:当smart264开启时-->true
</isSupportRefreshFrame>
<isSupportBareDataOverlay>
  <!--ro, opt, bool, 是否支持裸数据叠加-->true
</isSupportBareDataOverlay>
<isSupportRTCPCfg>
  <!--ro, opt, bool, 是否支持RTCP配置-->true
</isSupportRTCPCfg>
<customStreamEnable>
  <!--ro, opt, bool, 自定义码流是否开启-->true
</customStreamEnable>
</StreamingChannel>
</StreamingChannelList>
```

69.21.2 配置多通道编码参数

Request URL

PUT /ISAPI/Streaming/channels

查询参数

无

请求报文

```
<?xml version="1.0" encoding="UTF-8"?>

<StreamingChannelList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--opt, array, 通道列表, subType:object, attr:version{opt, string, 协议版本}-->
  <StreamingChannel>
    <!--opt, object, 每个通道的配置信息-->
    <id>
      <!--req, string, 通道ID-->test
    </id>
    <channelName>
      <!--req, string, 通道名称-->test
    </channelName>
    <enabled>
      <!--req, bool, 使能-->true
    </enabled>
    <Transport>
      <!--req, object, 传输配置-->
      <rtspPortNo>
        <!--opt, int, rtsp端口号, range:[0,65535]-->554
      </rtspPortNo>
```

```
<maxPacketSize>
  <!--opt, int, 最大包大小-->1
</maxPacketSize>
<audioPacketLength>
  <!--opt, int, 音频包长度-->1
</audioPacketLength>
<audioInboundPacketLength>
  <!--opt, int, 音频输入包长度-->1
</audioInboundPacketLength>
<audioInboundPortNo>
  <!--opt, int, 音频输入端口号-->1
</audioInboundPortNo>
<videoSourcePortNo>
  <!--opt, int, 视频源端口号-->1
</videoSourcePortNo>
<audioSourcePortNo>
  <!--opt, int, 音频源端口号-->1
</audioSourcePortNo>
<ControlProtocolList>
  <!--req, array, 控制协议列表, subType:object-->
  <ControlProtocol>
    <!--opt, object, 控制协议-->
    <streamingTransport>
      <!--req, enum, 码流传输协议, subType:string, [HTTP#HTTP,RTSP#RTSP,SHTTP#SHTTP]-->RTSP
    </streamingTransport>
  </ControlProtocol>
</ControlProtocolList>
<Unicast>
  <!--opt, object, 单播配置-->
  <enabled>
    <!--req, bool, 使能-->true
  </enabled>
  <interfaceID>
    <!--opt, string, 网卡ID-->test
  </interfaceID>
  <rtpTransportType>
    <!--opt, enum, RTP传输类型, subType:string, [RTP/UDP#RTP over UDP,RTP/TCP#RTP over TCP]-->RTP/UDP
  </rtpTransportType>
</Unicast>
<Multicast>
  <!--opt, object, 多播配置-->
  <enabled>
    <!--req, bool, 使能-->true
  </enabled>
  <userTriggerThreshold>
    <!--opt, int, 用户触发间隔（具体含义待确认）-->1
  </userTriggerThreshold>
  <destIPAddress>
    <!--opt, string, ipv4目的地址-->test
  </destIPAddress>
  <videoDestPortNo>
    <!--opt, int, 视频目的端口-->1
  </videoDestPortNo>
  <audioDestPortNo>
    <!--opt, int, 音频目的端口-->1
  </audioDestPortNo>
  <destIPv6Address>
    <!--opt, string, ipv6目的地址-->test
  </destIPv6Address>
  <tTl>
    <!--opt, int, 域名解析在DNS服务器中存留时间-->1
  </tTl>
  <activeMutilcastEnabled>
    <!--opt, bool, 主动多播使能-->true
  </activeMutilcastEnabled>
  <packagingFormat>
    <!--opt, enum, 封装格式, subType:string, [RTP#RTP,PS#PS]-->RTP
  </packagingFormat>
  <FecInfo>
    <!--opt, object, FEC数据占用额外带宽百分比-->
    <fecRatio>
      <!--req, int, FEC多播端口, range:[0,100], desc:默认由设备指定端口-->1
    </fecRatio>
    <fecDestPortNo>
      <!--opt, int, FEC目的端口号-->1
    </fecDestPortNo>
  </FecInfo>
</Multicast>
<Security>
  <!--opt, object, 安全配置-->
  <enabled>
    <!--req, bool, 使能-->true
  </enabled>
  <certificateType>
    <!--req, enum, RTSP认证方式, subType:string, [digest#digest,digest/baisc#digest/baisc]-->digest
  </certificateType>
  <SecurityAlgorithm>
    <!--opt, object, 安全算法-->
    <algorithmType>
      <!--opt, enum, 算法类型, subType:string, [MD5#MD5,SHA256#SHA256,MD5/SHA256#MD5/SHA256]-->MD5
    </algorithmType>
  </SecurityAlgorithm>
</Security>
```

```
<SRTPMulticast>
  <!--opt, object, SRTP多播配置-->
  <SRTPVideoDestPortNo>
    <!--opt, int, SRTP视频目的端口号-->1
  </SRTPVideoDestPortNo>
  <SRTPAudioDestPortNo>
    <!--opt, int, SRTP音频目的端口号-->1
  </SRTPAudioDestPortNo>
</SRTPMulticast>
</Transport>
<Video>
  <!--opt, object, 视频配置-->
  <enabled>
    <!--req, bool, 使能-->true
  </enabled>
  <videoInputChannelID>
    <!--opt, string, 模拟通道ID-->test
  </videoInputChannelID>
  <dynVideoInputChannelID>
    <!--opt, string, 数字通道ID-->test
  </dynVideoInputChannelID>
  <videoCodecType>
    <!--req, enum, 视频编码类型, subType:string,
[UNKNOWN#UNKNOWN,MPEG4#MPEG4,MJPEG#MJPEG,3GP#3GP,H.264#H.264,HK.264#HK.264,MPEG#MPEG,SVAC#SVAC,H.265#H.265,H.264BP#H.264BP,H.264HP#H.264HP,H.264SVC#H.264SVC,MJPEG2#MJPEG2]-->H.264
  </videoCodecType>
  <videoScanType>
    <!--opt, enum, 视频扫描类型, subType:string, [progressive#逐行扫描,interLaced#隔行扫描]-->progressive
  </videoScanType>
  <videoResolutionWidth>
    <!--req, int, 分辨率宽-->1
  </videoResolutionWidth>
  <videoResolutionHeight>
    <!--req, int, 分辨率高-->1
  </videoResolutionHeight>
  <videoResolutionName>
    <!--opt, enum, 分辨率名称, subType:string, [3MP#3MP,5MP#5MP,none#无]-->3MP
  </videoResolutionName>
  <videoDiffResolutionList>
    <!--opt, array, 分辨率列表, subType:object-->
    <videoDiffResolution>
      <!--opt, object, 分辨率-->
      <videoCodecType>
        <!--req, enum, 视频编码类型, subType:string, [MPEG4#MPEG4,MJPEG#MJPEG,3GP#3GP,H.264#H.264,HK.264#HK.264,MPEG#MPEG,SVAC#SVAC,H.265#H.265]-->H.264
      </videoCodecType>
      <videoResolutionWidth>
        <!--req, int, 分辨率宽-->1
      </videoResolutionWidth>
      <videoResolutionHeight>
        <!--req, int, 分辨率高-->1
      </videoResolutionHeight>
    </videoDiffResolution>
  </videoDiffResolutionList>
  <videoPositionX>
    <!--opt, int, 视频x坐标-->1
  </videoPositionX>
  <videoPositionY>
    <!--opt, int, 视频y坐标-->1
  </videoPositionY>
  <videoQualityControlType>
    <!--opt, enum, 视频码率类型, subType:string, [CBR#定码率,VBR#变码率]-->CBR
  </videoQualityControlType>
  <constantBitRate>
    <!--opt, int, 定码率-->1
  </constantBitRate>
  <fixedQuality>
    <!--opt, enum, 图像质量, subType:int, [1#Lowest,20#Lower,40#Low,60#medium,80#higher,100#highest]-->1
  </fixedQuality>
  <vbrUpperCap>
    <!--opt, int, 码流上限-->1
  </vbrUpperCap>
  <vbrLowerCap>
    <!--opt, int, 码流下限-->1
  </vbrLowerCap>
  <maxFrameRate>
    <!--req, int, 视频帧率, desc: 帧率x100, 如22帧下发2200,1/4帧下发25-->1
  </maxFrameRate>
  <keyFrameInterval>
    <!--opt, int, I帧间隔, unit:ms, unitType:时间, desc:为转化为毫秒时间单位的值-->1
  </keyFrameInterval>
  <rotationDegree>
    <!--opt, int, 旋转度数-->1
  </rotationDegree>
  <mirrorEnabled>
    <!--opt, bool, 镜面使能-->true
  </mirrorEnabled>
  <snapshotImageType>
    <!--opt, enum, 抓拍图片格式, subType:string, [JPEG#JPEG,GIF#GIF,PNG#PNG]-->JPEG
  </snapshotImageType>
  <Mpeg4Profile>
    <!--opt, enum, Mpeg4编码复杂度, subType:string, [SP#SP,ASP#ASP]-->SP
  </Mpeg4Profile>
  <H264Profile>
    <!--opt, enum, H264编码复杂度, subType:string, [Baseline#Baseline,Main#Main,High#High,Extended#Extended]-->Baseline
  </H264Profile>
```

```
<!--opt, enum, H264编码复杂度, subType:string, [Baseline#Baseline,Main#Main,High#High,Extended#Extended]-->Baseline
</H264Profile>
<SVACProfile>
  <!--opt, enum, SVAC编码复杂度, subType:string, [Baseline#Baseline,Main#Main,High#High,Extended#Extended]-->Baseline
</SVACProfile>
<GovLength>
  <!--opt, int, I帧间隔, desc:为帧的实际值-->1
</GovLength>
<SVC>
  <!--opt, object, 可分级视频编码功能-->
  <enabled>
  <!--req, bool, 使能-->true
</enabled>
<SVCMode>
  <!--opt, enum, 模式选择, subType:string, [manual#manual,auto#auto]-->manual
</SVCMode>
<protectLevel>
  <!--opt, enum, 保护等级, subType:string, [all#全等级,nuclear#核心等级]-->all
</protectLevel>
</SVC>
<smoothing>
  <!--opt, int, 码流平滑度[1-100], range:[1,100], desc:1等级表示清晰, 100表示平滑-->1
</smoothing>
<H265Profile>
  <!--opt, enum, H265编码复杂度, subType:string, [Baseline#Baseline,Main#Main,High#High,Extended#Extended]-->Baseline
</H265Profile>
<SmartCodec>
  <!--opt, object, smart编码-->
  <enabled>
  <!--req, bool, 使能-->true
</enabled>
</SmartCodec>
<vbrAverageCap>
  <!--opt, int, 平均码率, desc:依赖于SmartCodec使能打开-->1
</vbrAverageCap>
<IntelligentInfoDisplayMethod>
  <!--opt, enum, 智能信息显示方式, subType:string, [player#显示,non-player#不显示]-->player
</IntelligentInfoDisplayMethod>
<LBREnabled>
  <!--opt, bool, 低码率使能-->true
</LBREnabled>
<minimumResolutionSupportedBySmartCode>
  <!--opt, string, smart编码（包含smart264和smart265）支持的最小分辨率-->640*512
</minimumResolutionSupportedBySmartCode>
<videoInputType>
  <!--opt, enum, 视频输入方式, subType:string, [SDI#SDI方式,HDMI#HDMI方式,VGA#VGA方式,AUTO#自动方式]-->SDI
</videoInputType>
<BPFrameInterval>
  <!--opt, enum, P帧间隔, subType:string, [BBP#BBP帧,BP#BP帧,P#单P帧,auto#自动(和源一致)], desc:单P帧(2006-08-11 增加单P帧的配置接口, 可以改善实时流延
时间问题)-->auto
</BPFrameInterval>
<DepthMapEnable>
  <!--opt, bool, 深度图使能, desc:如果开启后, 第二通道的子码流(双目)的视频参数都不能配置, 默认输出 960*540 的深度图-->false
</DepthMapEnable>
</Video>
<Audio>
  <!--opt, object, 音频配置-->
  <enabled>
  <!--req, bool, 使能-->true
</enabled>
<audioInputChannelID>
  <!--req, string, 数字通道ID-->test
</audioInputChannelID>
<audioCompressionType>
  <!--req, enum, 音频编码类型, subType:string,
[UNKNOWN#UNKNOWN, G.711aLaw#G.711aLaw, G.711uLaw#G.711uLaw, G.726#G.726, G.729#G.729, G.729a#G.729a, G.729b#G.729b, PCM#PCM, MP3#MP3, AC3#AC3, AAC#AAC, ADPCM#ADPCM, MP2
L2#MP2L2, G722#G722, G.719#G.719, AAC-LC#AAC-LC, AAC-LD#AAC-LD, Opus#Opus, G.728#G.728, G.722.1#G.722.1, G.722.1.C#G.722.1.C]-->UNKNOWN
</audioCompressionType>
<audioInboundCompressionType>
  <!--opt, enum, 音频输入编码类型, subType:string,
[UNKNOWN#UNKNOWN, G.711aLaw#G.711aLaw, G.711uLaw#G.711uLaw, G.726#G.726, G.729#G.729, G.729a#G.729a, G.729b#G.729b, PCM#PCM, MP3#MP3, AC3#AC3, AAC#AAC, ADPCM#ADPCM, MP2
L2#MP2L2, G722#G722, G.719#G.719, AAC-LC#AAC-LC, AAC-LD#AAC-LD, Opus#Opus, G.728#G.728, G.722.1#G.722.1, G.722.1.C#G.722.1.C]-->UNKNOWN
</audioInboundCompressionType>
<audioBitRate>
  <!--opt, int, 音频编码比特率-->1
</audioBitRate>
<audioSamplingRate>
  <!--opt, float, 音频采样率-->0.0
</audioSamplingRate>
<audioResolution>
  <!--opt, int, 音频码率-->1
</audioResolution>
<VoiceChanger>
  <!--opt, object, 语音转换-->
  <enabled>
  <!--req, bool, 使能-->true
</enabled>
<level>
  <!--req, int, 级别, range:[-12,12]-->1
</level>
</VoiceChanger>
<audioInputType>
  <!--opt, enum, 音频输入方式, subType:string, [AudioIn#表示同时支持VGA和HDMI方式,HDMI#仅HDMI方式]-->AudioIn
</audioInputType>
```

```
</Audio>
<enableCABAC>
  <!--opt, bool, 码流压缩性能提升使能-->true
</enableCABAC>
<subStreamRecStatus>
  <!--ro, opt, bool, 子码流录像状态-->true
</subStreamRecStatus>
<isSupportRefreshFrame>
  <!--opt, bool, 是否支持重刷帧, desc: 当smart264开启时-->true
</isSupportRefreshFrame>
<isSupportBareDataOverLay>
  <!--opt, bool, 是否支持裸数据叠加-->true
</isSupportBareDataOverLay>
<isSupportRTCPCfg>
  <!--opt, bool, 是否支持RTCP配置-->true
</isSupportRTCPCfg>
<customStreamEnable>
  <!--opt, bool, 自定义码流是否开启-->true
</customStreamEnable>
</StreamingChannel>
</StreamingChannelList>
```

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 响应消息, attr:version{ro, req, string, 版本}-->
  <requestURL>
    <!--ro, req, string, 请求的URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, 状态码, subType:int, [0#OK,1#OK,2#Device Busy,3#Device Error,4#Invalid Operation,5#Invalid XML Format,6#Invalid XML Content,7#Reboot
    Required]-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, 状态信息, subType:string, [OK#成功,Device Busy#设备忙,Device Error#设备异常,Invalid Operation#无效的操作,Invalid XML Format#无效的XML
    格式,Invalid XML Content#无效的XML内容,Reboot#设备重启]-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, 详细错误码英文描述, desc: 详细错误码的英文描述-->OK
  </subStatusCode>
</ResponseStatus>
```

69.21.3 获取指定通道编码参数能力

Request URL

GET /ISAPI/Streaming/channels/<trackStreamID>/capabilities

查询参数

参数名称	参数类型	描述
trackStreamID	string	定义为: channel*100+streamType (1-mianstream, 2-substream依次类推); 例如101表示通道1的主码流, 302表示通道3的子码流

请求报文

无

响应报文

```

<?xml version="1.0" encoding="UTF-8"?>
<StreamingChannel xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 编码参数能力, attr:version{opt, string, 版本信息}-->
  <id opt="1,2">
    <!--ro, req, string, 通道ID, attr:opt{opt, string, 范围}-->1
  </id>
  <channelName min="0" max="0">
    <!--ro, req, string, 通道名称, attr:min{opt, string, 最小值},max{opt, string, 最大值}-->name1
  </channelName>
  <enabled opt="true,false" def="true">
    <!--ro, req, bool, 使能, attr:opt{opt, string, 范围},def{opt, string, 默认值}-->true
  </enabled>
  <Transport>
    <!--ro, req, object, 传输配置-->
    <ControlProtocolList>
      <!--ro, req, array, 控制协议列表, subType:object-->
      <ControlProtocol>
        <!--ro, opt, object, 控制协议-->
        <streamingTransport opt="HTTP,RTSP,SHTTP">
          <!--ro, req, enum, 码流传输协议, subType:string, [HTTP#HTTP,RTSP#RTSP,SHTTP#SHTTP], attr:opt{req, string, 取值范围}-->HTTP
        </streamingTransport>
      </ControlProtocol>
    </ControlProtocolList>
  </Transport>
  <Video>
    <!--ro, opt, object, 视频编码参数能力-->
    <enabled opt="true">
      <!--ro, req, bool, 编码使能;视频流和复合流时该置为true, attr:opt{opt, string, 范围}-->true
    </enabled>
    <videoInputChannelID opt="6">
      <!--ro, opt, string, 模拟通道ID, attr:opt{opt, string, 范围}-->1
    </videoInputChannelID>
    <videoCodecType opt="H.264,H.265,MPEG2">
      <!--ro, req, enum, 视频编码类型, subType:string, [H.264#H264,H.265#H265,MPEG2#MPEG2], attr:opt{opt, string, 范围}-->H.264
    </videoCodecType>
    <videoResolutionWidth opt="352,704,1280,640,1920,3840,1536,3072,1440,720">
      <!--ro, req, enum, 分辨率宽, subType:int, [352#352,704#704,1280#1280,640#640,1920#1920,3840#3840], attr:opt{opt, string, 范围}-->1920
    </videoResolutionWidth>
    <videoResolutionHeight opt="288,576,720,512,1080,2160,432,864,540">
      <!--ro, req, enum, 分辨率高, subType:int, [288#288,576#576,720#720,512#512,1080#1080,536#536], attr:opt{opt, string, 范围}-->1080
    </videoResolutionHeight>
    <videoQualityControlType opt="CBR,VBR,MVBR,SVBR">
      <!--ro, opt, enum, 码率类型, subType:string, [CBR#定码率,VBR#变码率,MVBR#MVBR,SVBR#SVBR], attr:opt{opt, string, 范围}-->CBR
    </videoQualityControlType>
    <constantBitRate min="32" max="16384">
      <!--ro, opt, int, 定码率, unit:kpbs, attr:min{opt, string, 范围},max{opt, string, 范围}-->6144
    </constantBitRate>
    <fixedQuality opt="1,20,30,45,60,75,90,100">
      <!--ro, opt, enum, 图像质量, subType:int, [1#Lowest,20#Lower,40#Low,60#medium,80#higher,100#highest], attr:opt{opt, string, 范围}-->1
    </fixedQuality>
    <vbrUpperCap min="32" max="16384">
      <!--ro, opt, int, 码流上限, range:[32,16384], attr:min{opt, string, 范围},max{opt, string, 范围}-->16384
    </vbrUpperCap>
    <vbrLowerCap opt="32">
      <!--ro, opt, enum, 码流下限, subType:int, [32#32], attr:opt{opt, string, 范围}-->32
    </vbrLowerCap>
    <maxFrameRate opt="0,5000,2500,2200,2000,1800,1600,1500,1200,1000,800,600,400,200,100,50,25,12,6">
      <!--ro, req, enum, 最大帧率, subType:int,
      [0#0,5000#5000,2500#2500,2200#2200,2000#2000,1800#1800,1600#1600,1500#1500,1200#1200,1000#1000,800#800,600#600,400#400,200#200,100#100,50#50,25#25,12#12,6#6
      ], attr:opt{opt, string, 范围},
      desc:1. 0表示全帧率, P制25fps, N制30fps。
      2. 像50fps, 60fps这种是高帧率。-->5000
    </maxFrameRate>
    <keyFrameInterval min="40" max="16000">
      <!--ro, opt, int, I帧间隔;以milliseconds为单位, range:[40,16000], unit:ms, unitType:时间, attr:min{opt, string, 最小值},max{opt, string, 最大值},
      desc:为转化为毫秒时间单位的值-->40
    </keyFrameInterval>
    <GovLength min="1" max="400">
      <!--ro, opt, int, I帧间隔, range:[1,400], attr:min{opt, string, 最小值},max{opt, string, 最大值}, desc:为帧的实际值-->1
    </GovLength>
  </Video>
</StreamingChannel>

```

69.22 视频画面调整

69.22.1 获取指定通道视频制式参数

Request URL

GET /ISAPI/Image/channels/<channelID>/powerLineFrequency

查询参数

参数名称	参数类型	描述
channelID	string	--

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<powerLineFrequency xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 视频制式, attr:version{req, string, 协议版本}-->
  <powerLineFrequencyMode>
    <!--ro, opt, enum, 视频制式模式, subType:string, [50hz#50hz,60hz#60hz]-->50hz
  </powerLineFrequencyMode>
</powerLineFrequency>
```

69.22.2 配置指定通道视频制式参数

Request URL

PUT /ISAPI/Image/channels/<channelID>/powerLineFrequency

查询参数

参数名称	参数类型	描述
channelID	string	--

请求报文

```
<?xml version="1.0" encoding="UTF-8"?>

<powerLineFrequency xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, 视频制式, attr:version{req, string, 协议版本}-->
  <powerLineFrequencyMode>
    <!--opt, enum, 视频制式模式, subType:string, [50hz#50hz,60hz#60hz]-->50hz
  </powerLineFrequencyMode>
</powerLineFrequency>
```

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 响应消息, attr:version{ro, req, string, 版本}-->
  <requestURL>
    <!--ro, req, string, 请求的URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, 状态码, subType:int, [0#OK,1#OK,2#Device Busy,3#Device Error,4#Invalid Operation,5#Invalid XML Format,6#Invalid XML Content,7#Reboot Required]-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, 状态信息, subType:string, [OK#成功,Device Busy#设备忙,Device Error#设备异常,Invalid Operation#无效的操作,Invalid XML Format#无效的XML格式,Invalid XML Content#无效的XML内容,Reboot#设备重启]-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, 详细错误码英文描述, desc: 详细错误码的英文描述-->OK
  </subStatusCode>
</ResponseStatus>
```

69.22.3 获取指定热成像通道融合标定能力

Request URL

GET /ISAPI/Thermal/channels/<channelID>/fusionCalibration/capabilities

查询参数

参数名称	参数类型	描述
channelID	string	通道号

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<FusionCalibrationCap xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, 热成像融合标定能力, attr:version{req, string, 协议版本}-->
  <distanceLevel min="1" max="10" default="4">
    <!--ro, opt, int, 距离档位, attr:min{req, int, 最小值},max{req, int, 最大值},default{opt, string, 默认值}-->1
  </distanceLevel>
  <ScaleRatio>
    <!--ro, opt, object, 缩放率-->
    <horizontal min="1" max="128" default="4">
      <!--ro, opt, int, 水平, attr:min{req, int, 最小值},max{req, int, 最大值},default{opt, string, 默认值}-->1
    </horizontal>
    <vertical min="1" max="128" default="4">
      <!--ro, opt, int, 垂直, attr:min{req, int, 最小值},max{req, int, 最大值},default{opt, string, 默认值}-->1
    </vertical>
  </ScaleRatio>
  <Bias>
    <!--ro, opt, object, 偏置-->
    <horizontal min="1" max="1" default="4">
      <!--ro, opt, int, 水平, attr:min{req, int, 最小值},max{req, int, 最大值},default{opt, string, 默认值}-->1
    </horizontal>
    <vertical min="1" max="1" default="4">
      <!--ro, opt, int, 垂直, attr:min{req, int, 最小值},max{req, int, 最大值},default{opt, string, 默认值}-->1
    </vertical>
  </Bias>
  <zoomRatio min="100" max="1000" default="100">
    <!--ro, opt, int, 变倍倍率, attr:min{req, int, 最小值},max{req, int, 最大值},default{opt, string, 默认值}-->1
  </zoomRatio>
  <croppingMode opt="center,floating">
    <!--ro, opt, enum, 裁剪模式, subType:string, [center#中心裁剪,fFloating#浮动裁剪], attr:opt{req, string, 取值范围}-->center
  </croppingMode>
  <verticalCroppingOffset min="-212" max="212">
    <!--ro, opt, int, 垂直裁剪偏移量, range:[-212,212], attr:min{req, int, 最小值},max{req, int, 最大值}-->1
  </verticalCroppingOffset>
  <horizontalCroppingOffset min="-212" max="212">
    <!--ro, opt, int, 水平裁剪偏移量, range:[-212,212], attr:min{req, int, 最小值},max{req, int, 最大值}-->1
  </horizontalCroppingOffset>
  <scaleNormalization min="1" max="10000" default="1000">
    <!--ro, opt, int, 缩放率归一化数值, range:[1,10000], unitType:无, attr:min{req, int, 最小值},max{req, int, 最大值},default{opt, int, 默认值}, desc:缩放率归一化数值, 若有这个节点, 则缩放率的归一化值为此节点值, 否则是默认1000-->1000
  </scaleNormalization>
</FusionCalibrationCap>
```

69.22.4 获取指定热成像通道融合标定参数

Request URL

GET /ISAPI/Thermal/channels/<channelID>/fusionCalibration

查询参数

参数名称	参数类型	描述
channelID	string	通道号

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<FusionCalibration xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, 热成像融合标定, attr:version{req, string, 协议版本}-->
  <distanceLevel>
    <!--ro, opt, int, 距离档位-->1
  </distanceLevel>
  <ScaleRatio>
    <!--ro, opt, object, 缩放率-->
    <horizontal>
      <!--ro, opt, int, 水平-->1
    </horizontal>
    <vertical>
      <!--ro, opt, int, 垂直-->1
    </vertical>
  </ScaleRatio>
  <Bias>
    <!--ro, opt, object, 偏置-->
    <horizontal>
      <!--ro, opt, int, 水平-->1
    </horizontal>
    <vertical>
      <!--ro, opt, int, 垂直-->1
    </vertical>
  </Bias>
  <zoomRatio>
    <!--ro, opt, int, 变倍倍率-->1
  </zoomRatio>
  <croppingMode>
    <!--ro, opt, enum, 裁剪模式, subType:string, [center#中心裁剪,floating#浮动裁剪]-->center
  </croppingMode>
  <verticalCroppingOffset>
    <!--ro, opt, int, 垂直裁剪偏移量, range:[-212,212]-->1
  </verticalCroppingOffset>
  <horizontalCroppingOffset>
    <!--ro, opt, int, 水平裁剪偏移量, range:[-212,212]-->1
  </horizontalCroppingOffset>
  <scaleNormalization>
    <!--ro, opt, int, 缩放率归一化数值, range:[1,10000], unitType:无, desc:缩放率归一化数值, 若有这个节点, 则缩放率的归一化值为此节点值, 否则是默认1000-->1000
  </scaleNormalization>
</FusionCalibration>
```

69.22.5 获取指定通道镜像参数

Request URL

GET /ISAPI/Image/channels/<channelID>/imageFlip

查询参数

参数名称	参数类型	描述
channelID	string	

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<ImageFlip xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, 镜像参数, attr:version{req, string, 协议版本}-->
  <enabled>
    <!--ro, opt, bool, 使能-->true
  </enabled>
  <ImageFlipStyle>
    <!--ro, opt, enum, 镜像类型, subType:string, [LEFTRIGHT#LEFTRIGHT,UPDOWN#UPDOWN,CENTER#CENTER,AUTO#AUTO]-->LEFTRIGHT
  </ImageFlipStyle>
  <flipAngle>
    <!--ro, opt, enum, 顺时针旋转角度, subType:string, [90#90,180#180,270#270]-->90
  </flipAngle>
</ImageFlip>
```

69.22.6 配置指定通道镜像参数

Request URL

PUT /ISAPI/Image/channels/<channelID>/imageFlip

查询参数

参数名称	参数类型	描述
channelID	string	

请求报文

```
<?xml version="1.0" encoding="UTF-8"?>

<ImageFlip xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--opt, object, 镜像参数, attr:version{req, string, 协议版本}-->
  <enabled>
    <!--opt, bool, 使能-->true
  </enabled>
  <ImageFlipStyle>
    <!--opt, enum, 镜像类型, subType:string, [LEFTRIGHT#LEFTRIGHT,UPDOWN#UPDOWN,CENTER#CENTER,AUTO#AUTO]-->LEFTRIGHT
  </ImageFlipStyle>
  <flipAngle>
    <!--opt, enum, 顺时针旋转角度, subType:string, [90#90,180#180,270#270]-->90
  </flipAngle>
</ImageFlip>
```

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 响应消息, attr:version{ro, req, string, 版本}-->
  <requestURL>
    <!--ro, req, string, 请求的URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, 状态码, subType:int, [0#OK,1#OK,2#Device Busy,3#Device Error,4#Invalid Operation,5#Invalid XML Format,6#Invalid XML Content,7#Reboot Required]-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, 状态信息, subType:string, [OK#成功,Device Busy#设备忙,Device Error#设备异常,Invalid Operation#无效的操作,Invalid XML Format#无效的XML格式,Invalid XML Content#无效的XML内容,Reboot#设备重启]-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, 详细错误码英文描述, desc: 详细错误码的英文描述-->OK
  </subStatusCode>
</ResponseStatus>
```

69.22.7 配置指定热成像通道融合标定参数

Request URL

PUT /ISAPI/Thermal/channels/<channelID>/fusionCalibration

查询参数

参数名称	参数类型	描述
channelID	string	通道号

请求报文

```
<?xml version="1.0" encoding="UTF-8"?>

<FusionCalibration xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--opt, object, 热成像融合标定, attr:version{req, string, 协议版本}-->
  <distanceLevel>
    <!--opt, int, 距离档位-->1
  </distanceLevel>
  <ScaleRatio>
    <!--opt, object, 缩放率-->
    <horizontal>
      <!--opt, int, 水平-->1
    </horizontal>
    <vertical>
      <!--opt, int, 垂直-->1
    </vertical>
  </ScaleRatio>
  <Bias>
    <!--opt, object, 偏置-->
    <horizontal>
      <!--opt, int, 水平-->1
    </horizontal>
    <vertical>
      <!--opt, int, 垂直-->1
    </vertical>
  </Bias>
  <zoomRatio>
    <!--opt, int, 变倍倍率-->1
  </zoomRatio>
  <croppingMode>
    <!--opt, enum, 裁剪模式, subType:string, [center#中心裁剪,floating#浮动裁剪]-->center
  </croppingMode>
  <verticalCroppingOffset>
    <!--opt, int, 垂直裁剪偏移量, range: [-212,212]-->1
  </verticalCroppingOffset>
  <horizontalCroppingOffset>
    <!--opt, int, 水平裁剪偏移量, range: [-212,212]-->1
  </horizontalCroppingOffset>
  <scaleNormalization>
    <!--opt, int, 缩放率归一化数值, range: [1,10000], unitType:无, desc:缩放率归一化数值, 若有这个节点, 则缩放率的归一化值为该节点值, 否则是默认1000-->1000
  </scaleNormalization>
</FusionCalibration>
```

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 响应消息, attr:version{ro, req, string, 版本}-->
  <requestURL>
    <!--ro, req, string, 请求的URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, 状态码, subType:int, [0#OK,1#OK,2#Device Busy,3#Device Error,4#Invalid Operation,5#Invalid XML Format,6#Invalid XML Content,7#Reboot Required]-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, 状态信息, subType:string, [OK#成功,Device Busy#设备忙,Device Error#设备异常,Invalid Operation#无效的操作,Invalid XML Format#无效的XML格式,Invalid XML Content#无效的XML内容,Reboot#设备重启]-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, 详细错误码英文描述, desc:详细错误码的英文描述-->OK
  </subStatusCode>
</ResponseStatus>
```

69.23 视频本地输入

69.23.1 获取所有视频输入参数

Request URL

GET /ISAPI/System/Video/inputs

查询参数

无

请求报文

无

响应报文

```

<?xml version="1.0" encoding="UTF-8"?>

<VideoInput xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, 视频输入, attr:version{req, string, 协议版本}-->
  <VideoInputChannelList>
    <!--ro, opt, array, 视频输入通道列表, subType:object-->
    <VideoInputChannel>
      <!--ro, opt, object, 视频输入通道-->
      <id>
        <!--ro, req, string, 索引-->test
      </id>
      <inputPort>
        <!--ro, req, string, 通道号-->test
      </inputPort>
      <videoInputEnabled>
        <!--ro, opt, bool, 使能-->true
      </videoInputEnabled>
      <name>
        <!--ro, opt, string, 通道名称-->test
      </name>
      <videoFormat>
        <!--ro, opt, enum, 视频制式, subType:string, [PAL#PAL,NTSC#NTSC]-->PAL
      </videoFormat>
      <portType>
        <!--ro, opt, enum, 端口类型, subType:string, [SDI#SDI,OPT#OPT,VGA#VGA,HDMI#HDMI,YPbPr#YPbPr,DVI#DVI,HDMI/HDbase-T#HDMI/HDbase-T]-->SDI
      </portType>
      <resDesc>
        <!--ro, opt, enum, 模拟通道状态描述, subType:string, [NO VIDEO#无信号,NTSC#N制,PAL#P制,1080P30 Lite#1080P30 Lite,4MP30 Lite#4MP30 Lite,TVI/CVBS
CHANNEL#TVI/CVBS CHANNEL,CVBS CHANNEL#CVBS CHANNEL,TVI 3MP CHANNEL#TVI 3MP CHANNEL,AHD 1080P CHANNEL#AHD 1080P CHANNEL,LANGUAGE MISMATCH#LANGUAGE MISMATCH]-
->NO VIDEO
      </resDesc>
      <channelType>
        <!--ro, opt, enum, 通道类型, subType:string, [trial#审讯通道,demonstration#示证通道,silence#无声通道]-->trial
      </channelType>
      <status>
        <!--ro, opt, enum, 通道状态, subType:string, [online#在线,offline#离线]-->online
      </status>
      <recordStatus>
        <!--ro, opt, enum, 录像状态, subType:string, [recording#录像中,pausing#暂停录像,notRecording#为录像]-->recording
      </recordStatus>
      <channelDescription>
        <!--ro, opt, string, 通道描述, desc:对通道号的细化描述, 如: 细节、全景等-->panorama
      </channelDescription>
    </VideoInputChannel>
  </VideoInputChannelList>
</VideoInput>

```

69.23.2 获取所有视频输入通道参数

Request URL

GET /ISAPI/System/Video/inputs/channels

查询参数

无

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<VideoInputChannelList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, array, 视频输入通道, subType:object, attr:version{req, string, 协议版本}-->
  <VideoInputChannel>
    <!--ro, opt, object, 视频输入通道-->
    <id>
      <!--ro, req, string, 索引-->test
    </id>
    <inputPort>
      <!--ro, req, string, 通道号-->test
    </inputPort>
    <videoInputEnabled>
      <!--ro, opt, bool, 使能-->true
    </videoInputEnabled>
    <name>
      <!--ro, opt, string, 通道名称-->test
    </name>
    <videoFormat>
      <!--ro, opt, enum, 视频制式, subType:string, [PAL#PAL,NTSC#NTSC]-->PAL
    </videoFormat>
    <portType>
      <!--ro, opt, enum, 端口类型, subType:string, [SDI#SDI,OPT#OPT,VGA#VGA,HDMI#HDMI,YPbPr#YPbPr,DVI#DVI,HDMI/Hdbase-T#HDMI/Hdbase-T]-->SDI
    </portType>
    <resDesc>
      <!--ro, opt, enum, 模拟通道状态描述, subType:string, [NO VIDEO#无信号,NTSC#N制,PAL#P制,1080P30 Lite#1080P30 Lite,4MP30 Lite#4MP30 Lite,TVI/CVBS CHANNEL#TVI/CVBS CHANNEL,CVBS CHANNEL#CVBS CHANNEL,TVI 3MP CHANNEL#TVI 3MP CHANNEL,AHD 1080P CHANNEL#AHD 1080P CHANNEL,LANGUAGE MISMATCH#LANGUAGE MISMATCH]->NO VIDEO
    </resDesc>
    <channelType>
      <!--ro, opt, enum, 通道类型, subType:string, [trial#审讯通道,demonstration#示证通道,silence#无声通道,courseware#课件通道]-->trial
    </channelType>
    <status>
      <!--ro, opt, enum, 通道状态, subType:string, [online#在线,offline#离线]-->online
    </status>
    <recordStatus>
      <!--ro, opt, enum, 录像状态, subType:string, [recording#录像中,pausing#暂停录像,notRecording#为录像]-->recording
    </recordStatus>
    <channelDescription>
      <!--ro, opt, string, 通道描述, range:[1,64], desc:对通道号的细化描述, 如: 细节、全景等-->panorama
    </channelDescription>
  </VideoInputChannel>
</VideoInputChannelList>
```

69.23.3 配置单个视频输入通道参数

Request URL

PUT /ISAPI/System/Video/inputs/channels/<channelID>

查询参数

参数名称	参数类型	描述
channelID	string	通道号

请求报文

```
<?xml version="1.0" encoding="UTF-8"?>

<VideoInputChannel xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--opt, object, 视频输入通道, attr:version{req, string, 协议版本}-->
  <id>
    <!--req, string, 索引-->test
  </id>
  <inputPort>
    <!--req, string, 通道号-->test
  </inputPort>
  <videoInputEnabled>
    <!--opt, bool, 使能-->true
  </videoInputEnabled>
  <name>
    <!--opt, string, 通道名称-->test
  </name>
  <videoFormat>
    <!--opt, enum, 视频制式, subType:string, [PAL#PAL,NTSC#NTSC]-->PAL
  </videoFormat>
  <portType>
    <!--opt, enum, 端口类型, subType:string, [SDI#SDI,OPT#OPT,VGA#VGA,HDMI#HDMI,YPbPr#YPbPr,DVI#DVI,HDMI/Hdbase-T#HDMI/Hdbase-T]-->SDI
  </portType>
  <resDesc>
    <!--ro, opt, enum, 模拟通道状态描述, subType:string, [NO VIDEO#无信号,NTSC#N制,PAL#P制,1080P30 Lite#1080P30 Lite,4MP30 Lite#4MP30 Lite,TVI/CVBS CHANNEL#TVI/CVBS CHANNEL,CVBS CHANNEL#CVBS CHANNEL,TVI 3MP CHANNEL#TVI 3MP CHANNEL,AHD 1080P CHANNEL#AHD 1080P CHANNEL,LANGUAGE MISMATCH#LANGUAGE MISMATCH]->NO VIDEO
  </resDesc>
  <channelType>
    <!--opt, enum, 通道类型, subType:string, [trial#审讯通道,demonstration#示证通道,silence#无声通道]-->trial
  </channelType>
  <status>
    <!--opt, enum, 通道状态, subType:string, [online#在线,offline#离线]-->online
  </status>
  <recordStatus>
    <!--opt, enum, 录像状态, subType:string, [recording#录像中,pausing#暂停录像,notRecording#为录像]-->recording
  </recordStatus>
</VideoInputChannel>
```

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 响应消息, attr:version{ro, req, string, 版本}-->
  <requestURL>
    <!--ro, req, string, 请求的URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, 状态码, subType:int, [0#OK,1#OK,2#Device Busy,3#Device Error,4#Invalid Operation,5#Invalid XML Format,6#Invalid XML Content,7#Reboot Required]-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, 状态信息, subType:string, [OK#成功,Device Busy#设备忙,Device Error#设备异常,Invalid Operation#无效的操作,Invalid XML Format#无效的XML格式,Invalid XML Content#无效的XML内容,Reboot#设备重启]-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, 详细错误码英文描述, desc:详细错误码的英文描述-->OK
  </subStatusCode>
</ResponseStatus>
```

69.23.4 获取单个视频输入通道参数

Request URL

GET /ISAPI/System/Video/inputs/channels/<channelID>

查询参数

参数名称	参数类型	描述
channelID	string	通道号

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<VideoInputChannel xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, 视频输入通道, attr:version{req, string, 协议版本}-->
  <id>
    <!--ro, req, string, 索引-->test
  </id>
  <inputPort>
    <!--ro, req, string, 通道号-->test
  </inputPort>
  <videoInputEnabled>
    <!--ro, opt, bool, 使能-->true
  </videoInputEnabled>
  <name>
    <!--ro, opt, string, 通道名称-->test
  </name>
  <videoFormat>
    <!--ro, opt, enum, 视频制式, subType:string, [PAL#PAL,NTSC#NTSC]-->PAL
  </videoFormat>
  <portType>
    <!--ro, opt, enum, 端口类型, subType:string, [SDI#SDI,OPT#OPT,VGA#VGA,HDMI#HDMI,YPbPr#YPbPr,DVI#DVI,HDMI/Hdbase-T#HDMI/Hdbase-T]-->SDI
  </portType>
  <resDesc>
    <!--ro, opt, enum, 模拟通道状态描述, subType:string, [NO VIDEO#无信号,NTSC#N制,PAL#P制,1080P30 Lite#1080P30 Lite,4MP30 Lite#4MP30 Lite,TVI/CVBS CHANNEL#TVI/CVBS CHANNEL,CVBS CHANNEL#CVBS CHANNEL,TVI 3MP CHANNEL#TVI 3MP CHANNEL,AHD 1080P CHANNEL#AHD 1080P CHANNEL,LANGUAGE MISMATCH#LANGUAGE MISMATCH]->NO VIDEO
  </resDesc>
  <channelType>
    <!--ro, opt, enum, 通道类型, subType:string, [trial#审讯通道,demonstration#示证通道,silence#无声通道,courseware#课件通道]-->trial
  </channelType>
  <status>
    <!--ro, opt, enum, 通道状态, subType:string, [online#在线,offline#离线]-->online
  </status>
  <recordStatus>
    <!--ro, opt, enum, 录像状态, subType:string, [recording#录像中,pausing#暂停录像,notRecording#未录像]-->recording
  </recordStatus>
  <channelDescription>
    <!--ro, opt, string, 通道描述, range:[1,64], desc:对通道号的细化描述, 如: 细节、全景等-->panorama
  </channelDescription>
</VideoInputChannel>
```

69.24 图像调节

69.24.1 获取指定通道图像调节（自动）参数

Request URL

GET /ISAPI/Image/channels/<channelID>/color

查询参数

参数名称	参数类型	描述
channelID	string	通道编号

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<Color xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, Color, attr:version{opt, string, 版本号}-->
  <brightnessLevel>
    <!--ro, opt, int, 亮度-->24
  </brightnessLevel>
  <contrastLevel>
    <!--ro, opt, int, 对比度-->22
  </contrastLevel>
  <saturationLevel>
    <!--ro, opt, int, 饱和度, dep:and,{$.Color.nightMode,eq,true}-->33
  </saturationLevel>
  <hueLevel>
    <!--ro, opt, int, 色调-->55
  </hueLevel>
  <grayScale>
    <!--ro, opt, object, 灰度-->
    <grayScaleMode>
      <!--ro, opt, enum, 灰度模式, subType:string, [indoor#室内,outdoor#户外]-->outdoor
    </grayScaleMode>
  </grayScale>
  <nightMode>
    <!--ro, opt, bool, 夜间黑白模式使能-->true
  </nightMode>
  <redBrightnessLevel>
    <!--ro, opt, int, 红外路亮度, range:[0,100]-->0
  </redBrightnessLevel>
  <sharpnessLevel>
    <!--ro, opt, int, 锐度, range:[0,100]-->0
  </sharpnessLevel>
</Color>
```

69.24.2 配置指定通道图像调节（自动）参数

Request URL

PUT /ISAPI/Image/channels/<channelID>/color

查询参数

参数名称	参数类型	描述
channelID	string	通道编号

请求报文

```
<?xml version="1.0" encoding="UTF-8"?>

<Color xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--opt, object, Color, attr:version{opt, string, 版本号}-->
  <brightnessLevel>
    <!--opt, int, 亮度-->24
  </brightnessLevel>
  <contrastLevel>
    <!--opt, int, 对比度-->22
  </contrastLevel>
  <saturationLevel>
    <!--opt, int, 饱和度, dep:and,{$.Color.nightMode,eq,true}-->33
  </saturationLevel>
  <hueLevel>
    <!--opt, int, 色调-->55
  </hueLevel>
  <grayScale>
    <!--opt, object, 灰度-->
    <grayScaleMode>
      <!--opt, enum, 灰度模式, subType:string, [indoor#室内,outdoor#户外]-->outdoor
    </grayScaleMode>
  </grayScale>
  <nightMode>
    <!--opt, bool, 夜间黑白模式使能-->true
  </nightMode>
  <redBrightnessLevel>
    <!--opt, int, 红外路亮度, range:[0,100]-->0
  </redBrightnessLevel>
  <sharpnessLevel>
    <!--opt, int, 锐度, range:[0,100]-->0
  </sharpnessLevel>
</Color>
```

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 响应消息, attr:version{ro, req, string, 版本}-->
  <requestURL>
    <!--ro, req, string, 请求的URL-->/ISAPI/xxxx
  </requestURL>
  <statusCode>
    <!--ro, req, enum, 状态码, subType:int, [0#OK,1#OK,2#Device Busy,3#Device Error,4#Invalid Operation,5#Invalid XML Format,6#Invalid XML Content,7#Reboot Required]-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, 状态信息, subType:string, [OK#成功,Device Busy#设备忙,Device Error#设备异常,Invalid Operation#无效的操作,Invalid XML Format#无效的XML格式,Invalid XML Content#无效的XML内容,Reboot#设备重启]-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, 详细错误码英文描述, desc:详细错误码的英文描述-->OK
  </subStatusCode>
</ResponseStatus>
```

69.24.3 获取指定通道图像锐度控制（自动）参数

Request URL

GET /ISAPI/Image/channels/<channelID>/sharpness

查询参数

参数名称	参数类型	描述
channelID	string	

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<Sharpness xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, 锐度, attr:version{opt, string, 版本号}-->
  <SharpnessMode>
    <!--ro, opt, enum, 锐度模式, subType:string, [manual#手动,auto#自动]-->manual
  </SharpnessMode>
  <SharpnessLevel>
    <!--ro, req, int, 锐度级别-->1
  </SharpnessLevel>
</Sharpness>
```

69.24.4 配置指定通道图像锐度控制（自动）参数

Request URL

PUT /ISAPI/Image/channels/<channelID>/sharpness

查询参数

参数名称	参数类型	描述
channelID	string	

请求报文

```
<?xml version="1.0" encoding="UTF-8"?>

<Sharpness xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--opt, object, 锐度, attr:version{opt, string, 版本号}-->
  <SharpnessMode>
    <!--opt, enum, 锐度模式, subType:string, [manual#手动,auto#自动]-->manual
  </SharpnessMode>
  <SharpnessLevel>
    <!--req, int, 锐度级别-->1
  </SharpnessLevel>
</Sharpness>
```

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 响应消息, attr:version{ro, req, string, 版本}-->
  <requestURL>
    <!--ro, req, string, 请求的URL-->/ISAPI/xxxx
  </requestURL>
  <statusCode>
    <!--ro, req, enum, 状态码, subType:int, [0#OK,1#OK,2#Device Busy,3#Device Error,4#Invalid Operation,5#Invalid XML Format,6#Invalid XML Content,7#Reboot Required]-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, 状态信息, subType:string, [OK#成功,Device Busy#设备忙,Device Error#设备异常,Invalid Operation#无效的操作,Invalid XML Format#无效的XML格式,Invalid XML Content#无效的XML内容,Reboot#设备重启]-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, 详细错误码英文描述, desc:详细错误码的英文描述-->OK
  </subStatusCode>
</ResponseStatus>
```

69.24.5 重置指定通道视频图像参数

Request URL

PUT /ISAPI/Image/channels/<channelID>/restore

查询参数

参数名称	参数类型	描述
channelID	string	通道编号

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 响应消息, attr:version{ro, req, string, 版本}-->
  <requestURL>
    <!--ro, req, string, 请求的URL-->/ISAPI/xxxx
  </requestURL>
  <statusCode>
    <!--ro, req, enum, 状态码, subType:int, [0#OK,1#OK,2#Device Busy,3#Device Error,4#Invalid Operation,5#Invalid XML Format,6#Invalid XML Content,7#Reboot Required]-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, 状态信息, subType:string, [OK#成功,Device Busy#设备忙,Device Error#设备异常,Invalid Operation#无效的操作,Invalid XML Format#无效的XML格式,Invalid XML Content#无效的XML内容,Reboot#设备重启]-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, 详细错误码英文描述, desc:详细错误码的英文描述-->OK
  </subStatusCode>
</ResponseStatus>
```

69.24.6 获取所有通道视频图像模式参数

Request URL

GET /ISAPI/Image/channels/imageModes

查询参数

无

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<ImageModeList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, array, 图像模式配置, subType:object, attr:version{req, string, 协议版本}-->
  <ImageMode>
    <!--ro, opt, object, 图像模式-->
    <type>
      <!--ro, req, enum, 类型, subType:string, [standard#标准,indoor#室内,outdoor#室外,dimLight#暗灯,bright#明亮,soft#柔和,vivid#鲜艳,custom#自定义]-->
      >standard
    </type>
    <recommendation>
      <!--ro, req, object, 推荐-->
      <brightnessLevel>
        <!--ro, opt, int, 亮度, range:[0,100]-->1
      </brightnessLevel>
      <contrastLevel>
        <!--ro, opt, int, 对比度, range:[0,100]-->1
      </contrastLevel>
      <sharpnessLevel>
        <!--ro, opt, int, 锐度, range:[0,100]-->1
      </sharpnessLevel>
      <saturationLevel>
        <!--ro, opt, int, 饱和度, range:[0,100]-->1
      </saturationLevel>
      <hueLevel>
        <!--ro, opt, int, 色调, range:[0,100]-->1
      </hueLevel>
      <deNoiseLevel>
        <!--ro, opt, int, 抗噪, range:[0,100]-->1
      </deNoiseLevel>
    </recommendation>
  </ImageMode>
</ImageModeList>
```

69.24.7 获取指定通道视频图模式参数

Request URL

GET /ISAPI/Image/channels/imageModes/<channelID>

查询参数

参数名称	参数类型	描述
channelID	string	

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<ImageMode xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, 图像模式, attr:version{req, string, 协议版本}-->
  <type>
    <!--ro, req, enum, 类型, subType:string, [standard#标准,indoor#室内,outdoor#室外,dimLight#暗灯,bright#明亮,soft#柔和,vivid#鲜艳,custom#自定义]-->
    >standard
  </type>
  <recommendation>
    <!--ro, req, object, 推荐-->
    <brightnessLevel>
      <!--ro, opt, int, 亮度, range:[0,100]-->1
    </brightnessLevel>
    <contrastLevel>
      <!--ro, opt, int, 对比度, range:[0,100]-->1
    </contrastLevel>
    <sharpnessLevel>
      <!--ro, opt, int, 锐度, range:[0,100]-->1
    </sharpnessLevel>
    <saturationLevel>
      <!--ro, opt, int, 饱和度, range:[0,100]-->1
    </saturationLevel>
    <hueLevel>
      <!--ro, opt, int, 色调, range:[0,100]-->1
    </hueLevel>
    <deNoiseLevel>
      <!--ro, opt, int, 抗噪, range:[0,100]-->1
    </deNoiseLevel>
  </recommendation>
</ImageMode>
```

69.24.8 配置指定通道图像处理参数

Request URL

PUT /ISAPI/Image/channels/<channelID>

查询参数

参数名称	参数类型	描述
channelID	string	视频通道号

请求报文

```
<?xml version="1.0" encoding="UTF-8"?>

<ImageChannel xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, 图像信号处理参数, attr:version{req, string, 协议版本}-->
  <id>
    <!--req, int, 序号ID-->1
  </id>
  <enabled>
    <!--req, bool, 使能-->true
  </enabled>
  <videoInputID>
    <!--req, int, 视频通道号ID-->1
  </videoInputID>
  <Focusconfiguration>
    <!--opt, object, 普通聚焦-->
    <focusStyle>
      <!--req, enum, 聚焦类型, subType:string, [AUTO#自动,MANUAL#手动,SEMIAUTOMATIC#半自动]-->AUTO
    </focusStyle>
    <focusLimited>
      <!--opt, enum, 聚焦限制, subType:int, [10#10,50#50,100#100,150#150,300#300,600#600,1000#1000,2000#2000,5000#5000,20000#20000,50000#50000,65535#65535]->10
    </focusLimited>
    <focusPosition>
      <!--opt, int, 焦点位置, range:[1,10]-->1
    </focusPosition>
    <focusSpeed>
      <!--opt, int, 聚焦速度, range:[1,3]-->1
    </focusSpeed>
    <focusSensitivity>
      <!--opt, int, 聚焦灵敏度, range:[0,2]-->1
    </focusSensitivity>
    <temperatureChangeAdaptEnabled>
      <!--opt, bool, 温度变化适应使能-->true
    </temperatureChangeAdaptEnabled>
    <relativeFocusPos>
      <!--opt, int, 相对焦点位置, range:[0,4294967295]-->1
    </relativeFocusPos>
    <lensType>
      <!--opt, enum, 聚焦镜头类型, subType:string, [1140#1140,1050#1050]-->1140
    </lensType>
    <highTemperaturePriority>
      <!--opt, bool, 高温优先模式使能-->true
    </highTemperaturePriority>
    <focusRange>
      <!--opt, enum, 聚焦调节范围, subType:string, [min#最小,medium#适中,max#最大]-->max
    </focusRange>
    <focusDistanceMode>
      <!--opt, enum, 聚焦距离模式, subType:string, [compatible#兼容模式], desc:compatible-兼容模式, 在球罩设备上调整聚焦限制距离focusLimited, 仍然虚焦的情况, 选择兼容模式, focusLimited参数失效, 设备释放镜头最大聚焦能力, 使镜头获得更大的焦距范围, 解决球罩设备近距聚焦效果差的问题。开启该功能, 聚焦速度会变慢, 非球罩设备不建议使用。-->compatible
    </focusDistanceMode>
  </Focusconfiguration>
  <LensInitialization>
    <!--opt, object, 镜头初始化-->
    <enabled>
      <!--req, bool, 使能-->true
    </enabled>
  </LensInitialization>
  <ImageFlip>
    <!--opt, object, 图像翻转-->
    <enabled>
      <!--req, bool, 使能-->true
    </enabled>
    <ImageFlipStyle>
      <!--opt, enum, 镜像类型, subType:string, [LEFTRIGHT#LEFTRIGHT,UPDOWN#UPDOWN,CENTER#CENTER,AUTO#AUTO]-->LEFTRIGHT
    </ImageFlipStyle>
    <flipAngle>
      <!--opt, enum, 旋转角度, subType:int, [90#90,180#180,270#270]-->90
    </flipAngle>
  </ImageFlip>
  <ImageFreeze>
    <!--opt, object, 图像冻结-->
    <enabled>
      <!--req, bool, 使能-->true
    </enabled>
```

```
</ImageFreeze>
<proportionalpan>
  <!--opt, object, 比例盘-->
  <enabled>
    <!--req, bool, 使能-->true
  </enabled>
</proportionalpan>
<WDR>
  <!--opt, object, 宽动态(自动)能力-->
  <mode>
    <!--req, enum, open, subType:string, [open#开启,close#关闭,auto#自动,SDR#SDR]-->open
  </mode>
  <WDRLevel>
    <!--opt, int, 宽动态等级, range:[1,100]-->1
  </WDRLevel>
  <WDRContrastLevel>
    <!--opt, int, 宽动态对比度, range:[1,100]-->1
  </WDRContrastLevel>
  <WDRLevel1>
    <!--opt, int, 第二类宽动态等级, range:[1,100]-->1
  </WDRLevel1>
</WDR>
<BLC>
  <!--opt, object, 背光补偿区域-->
  <enabled>
    <!--req, bool, 使能-->true
  </enabled>
  <BLCMode>
    <!--opt, enum, 背光补偿模式, subType:string, [UP#UP,DOWN#DOWN,LEFT#LEFT,RIGHT#RIGHT,CENTER#CENTER,MULTI-AREA#MULTI-AREA,Region#Region,AUTO#AUTO]-->UP
  </BLCMode>
  <BLCLevel>
    <!--opt, int, 背光补偿等级, range:[0,10]-->0
  </BLCLevel>
  <BLCRegionList>
    <!--opt, array, 背光补偿区域列表, subType:object-->
    <BLCRegion>
      <!--opt, object, 背光补偿区域-->
      <id>
        <!--req, int, 区域编号, range:[1,10]-->1
      </id>
      <RegionCoordinatesList>
        <!--req, array, 坐标点列表, subType:object, range:[0,4], desc:矩形-->
        <RegionCoordinates>
          <!--opt, object, 坐标点, desc:屏幕左下角为坐标原点-->
          <positionX>
            <!--req, int, x坐标, range:[0,704]-->1
          </positionX>
          <positionY>
            <!--req, int, y坐标, range:[0,576]-->1
          </positionY>
        </RegionCoordinates>
      </RegionCoordinatesList>
    </BLCRegion>
  </BLCRegionList>
</BLC>
<NoiseReduce>
  <!--opt, object, 降噪参数-->
  <mode>
    <!--req, enum, 模式, subType:string, [close#关闭,general#常规,advanced#高级]-->close
  </mode>
  <GeneralMode>
    <!--opt, object, 模式-->
    <generalLevel>
      <!--req, int, 等级, range:[0,100]-->1
    </generalLevel>
  </GeneralMode>
  <AdvancedMode>
    <!--opt, object, 高级模式-->
    <FrameNoiseReduceLevel>
      <!--req, int, 框架噪声降阶, range:[0,100]-->1
    </FrameNoiseReduceLevel>
    <InterFrameNoiseReduceLevel>
      <!--req, int, 帧间噪声降维, range:[0,100]-->1
    </InterFrameNoiseReduceLevel>
    <secondarySpatialNoiseReduceLevel>
      <!--opt, int, 二级降噪空域等级, range:[0,100]-->1
    </secondarySpatialNoiseReduceLevel>
    <secondaryTemporalNoiseReduceLevel>
      <!--opt, int, 二级降噪时域等级, range:[0,100]-->1
    </secondaryTemporalNoiseReduceLevel>
  </AdvancedMode>
</NoiseReduce>
<ImageEnhancement>
  <!--opt, object, 图像增强-->
  <enabled>
    <!--req, bool, 使能-->true
  </enabled>
  <ImageEnhancementLevel>
    <!--opt, int, 图像增强等级, range:[0,10]-->0
  </ImageEnhancementLevel>
</ImageEnhancement>
<DSS>
  <!--opt, object, DSS-->
```

```
<enabled>
  <!--req, bool, 使能-->true
</enabled>
<DSSLevel>
  <!--opt, enum, 低照度等级, subType:string,
[*1.25#*1.25,*1.5#*1.5,*2#*2,*3#*3,*4#*4,*5#*5,*6#*6,*8#*8,*12#*12,*16#*16,*24#*24,*32#*32,*48#*48,*64#*64,*96#*96,*128#*128,*256#*256,*512#*512,auto#auto]-
->*1.25
</DSSLevel>
<mode>
  <!--opt, enum, 模式, subType:string, [frame#帧模式,Level#等级模式], desc:无mode节点默认为frame模式-->frame
</mode>
<slowShutterLevel>
  <!--opt, int, 慢快门等级, range:[1,6]-->1
</slowShutterLevel>
<DSS>
<WhiteBalance>
  <!--opt, object, 白平衡-->
  <WhiteBalanceStyle>
    <!--opt, enum, 白平衡类型, subType:string, [auto#自动,auto1#自动1,auto2#自动2>manual#手动,indoor#室内,outdoor#室外,autotrace#自动跟踪,once#单
次,sodiumLight#苏打光,mercuryLight#水银灯,customColorTemperature#自定义色温]-->auto
  </WhiteBalanceStyle>
  <WhiteBalanceRed>
    <!--opt, int, 白平衡红, range:[0,100]-->1
  </WhiteBalanceRed>
  <WhiteBalanceBlue>
    <!--opt, int, 白平衡蓝, range:[0,100]-->1
  </WhiteBalanceBlue>
  <WhiteBalanceLevel>
    <!--opt, int, 白平衡等级, range:[0,100], desc:<WhiteBalanceStyle>为auto,auto1,auto2时有效-->1
  </WhiteBalanceLevel>
  <customColorTemperatureLevel>
    <!--opt, int, 自定义色温等级, range:[2700,7500], dep:and,{$.WhiteBalance,eq,customColorTemperature}-->1
  </customColorTemperatureLevel>
</WhiteBalance>
<Exposure>
  <!--opt, object, 曝光模式-->
  <ExposureType>
    <!--req, enum, 镜头类型, subType:string, [auto#auto,IrisFirst#IrisFirst,ShutterFirst#ShutterFirst,gainFirst#gainFirst>manual#manual,pIris#pIris,T5280-
PQ1#T5280-PQ1,T5289-PQ1#T5289-PQ1,T1140-PQ1#T1140-PQ1,T2712-PQ1#T2712-PQ1,HV1250P-MPIR#HV1250P-MPIR,pc#pc,pIris-General#pIris-General,HZ1140P-
8MPIR#HZ1140P-8MPIR,HZ1135P-8MPGSA#HZ1135P-8MPGSA,MV1550P-12MPIR#MV1550P-12MPIR,MZ1545P-12MP#MZ1545P-12MP,HZ0618P-4MPGSA#HZ0618P-4MPGSA,MZ1050P-
12MP#MZ1050P-12MP]-->auto
  </ExposureType>
  <autoIrisLevel>
    <!--opt, int, 光圈自动级别, range:[0,100]-->1
  </autoIrisLevel>
  <OverexposeSuppress>
    <!--opt, object, 过度曝光抑制-->
    <enabled>
      <!--req, bool, 使能-->true
    </enabled>
    <Type>
      <!--opt, enum, 过度曝光抑制类型, subType:string, [AUTO#自动,MANUAL#手动]-->AUTO
    </Type>
    <DistanceLevel>
      <!--opt, int, 焦距, range:[1,100]-->1
    </DistanceLevel>
    <shortIRDistanceLevel>
      <!--opt, int, IR短距离级别, range:[1,100]-->1
    </shortIRDistanceLevel>
    <longIRDistanceLevel>
      <!--opt, int, IR长距离级别, range:[1,100]-->1
    </longIRDistanceLevel>
    <supplementLightIntensity1>
      <!--opt, int, 补充曝光强度1, range:[1,100]-->1
    </supplementLightIntensity1>
    <supplementLightIntensity2>
      <!--opt, int, 补充曝光强度2, range:[1,100]-->1
    </supplementLightIntensity2>
    <supplementLightIntensity3>
      <!--opt, int, 补充曝光强度3, range:[1,100]-->1
    </supplementLightIntensity3>
    <supplementLightIntensity4>
      <!--opt, int, 补充曝光强度4, range:[1,100]-->1
    </supplementLightIntensity4>
  </OverexposeSuppress>
  <pIris>
    <!--opt, object, 光圈-->
    <pIrisType>
      <!--opt, enum, 光圈类型, subType:string, [AUTO#AUTO,MANUAL#MANUAL]-->MANUAL
    </pIrisType>
    <IrisLevel>
      <!--opt, int, 光圈级别, range:[1,100]-->1
    </IrisLevel>
  </pIris>
  <faceExposure>
    <!--opt, object, 人脸曝光-->
    <enabled>
      <!--opt, bool, 使能-->true
    </enabled>
    <sensitivity>
      <!--opt, int, 灵敏度, range:[0,100]-->0
    </sensitivity>
  </faceExposure>
  <PTriGeneral>
```

```
<!--opt, object, pIris-General配置-->
<irisLevel>
  <!--opt, int, 通用镜头下的光圈等级, range:[1,100]-->1
</irisLevel>
</IrisGeneral>
</Exposure>
<Sharpness>
  <!--opt, object, 锐度-->
  <SharpnessMode>
    <!--opt, enum, 锐度模式, subType:string, [manual#手动,auto#自动]-->manual
  </SharpnessMode>
  <SharpnessLevel>
    <!--req, int, 锐度级别, range:[0,100]-->1
  </SharpnessLevel>
</Sharpness>
<gammaCorrection>
  <!--opt, object, 伽玛校正-->
  <gammaCorrectionEnabled>
    <!--opt, bool, 伽玛校正使能-->true
  </gammaCorrectionEnabled>
  <gammaCorrectionLevel>
    <!--opt, int, 伽玛校正级别, range:[0,100]-->1
  </gammaCorrectionLevel>
</gammaCorrection>
<Shutter>
  <!--opt, object, 快门配置-->
  <ShutterLevel>
    <!--opt, enum, 曝光等级, subType:string,
[1/1#1/1,1/2#1/2,1/3#1/3,1/6#1/6,1/12#1/12,1/25#1/25,1/50#1/50,1/75#1/75,1/100#1/100,1/120#1/120,1/125#1/125,1/150#1/150,1/175#1/175], desc:depends on
<ExposureType>-->1/1
  </ShutterLevel>
  <maxShutterLevelLimit>
    <!--opt, enum, 最大曝光等级, subType:string,
[1/1#1/1,1/2#1/2,1/3#1/3,1/6#1/6,1/12#1/12,1/25#1/25,1/50#1/50,1/75#1/75,1/100#1/100,1/120#1/120,1/125#1/125,1/150#1/150,1/175#1/175]-->1/1
  </maxShutterLevelLimit>
  <minShutterLevelLimit>
    <!--opt, enum, 最小曝光等级, subType:string,
[1/1#1/1,1/2#1/2,1/3#1/3,1/6#1/6,1/12#1/12,1/25#1/25,1/50#1/50,1/75#1/75,1/100#1/100,1/120#1/120,1/125#1/125,1/150#1/150,1/175#1/175]-->1/1
  </minShutterLevelLimit>
</Shutter>
<powerLineFrequency>
  <!--opt, object, 视频制式-->
  <powerLineFrequencyMode>
    <!--opt, enum, 视频制式模式, subType:string, [50hz#50hz,60hz#60hz]-->50hz
  </powerLineFrequencyMode>
</powerLineFrequency>
<Gain>
  <!--opt, object, 增益-->
  <GainLevel>
    <!--opt, string, 增益等级, range:[0,10]-->test
  </GainLevel>
  <GainLimit>
    <!--opt, enum, 增益, subType:string,
[0#0,400#400,900#900,1300#1300,1800#1800,2200#2200,2600#2600,3100#3100,3500#3500,4000#4000,4400#4400,4800#4800,5300#5300,5700#5700,6200#6200,6600#6600]-->0
  </GainLimit>
</Gain>
<Color>
  <!--opt, object, 颜色-->
  <brightnessLevel>
    <!--opt, int, 亮度, range:[0,100]-->0
  </brightnessLevel>
  <contrastLevel>
    <!--opt, int, 对比度, range:[0,100]-->0
  </contrastLevel>
  <saturationLevel>
    <!--opt, int, 饱和度, range:[0,100]-->0
  </saturationLevel>
  <hueLevel>
    <!--opt, int, 色调, range:[0,100]-->0
  </hueLevel>
  <grayScale>
    <!--opt, object, 灰度-->
    <grayScaleMode>
      <!--opt, enum, 灰度模式, subType:string, [indoor#室内,outdoor#室外]-->indoor
    </grayScaleMode>
  </grayScale>
  <nightMode>
    <!--opt, bool, 夜间黑白模式使能-->true
  </nightMode>
  <sharpnessLevel>
    <!--opt, int, 锐度, range:[0,100]-->0
  </sharpnessLevel>
</Color>
<IrcutFilter>
  <!--opt, object, IrcutFilter配置-->
  <IrcutFilterType>
    <!--opt, enum, IrcutFilter类型, subType:string,
[auto#auto,day#day,night#night,schedule#schedule,eventTrigger#eventTrigger,videoAuto#videoAuto,darkFighterX#darkFighterX,darkFighterXAuto#darkFighterXAuto,d
arkFighterXSchedule#darkFighterXSchedule]-->auto
  </IrcutFilterType>
  <dayToNightFilterLevel>
    <!--opt, enum, 日夜切换级别, subType:string, [low#低,normal#中,high#高]-->low
  </dayToNightFilterLevel>
```

```
<dayToNightFilterTime>
  <!--opt, int, 日夜切换时间, range:[0,10]-->1
</dayToNightFilterTime>
<nightToDayFilterLevel>
  <!--opt, enum, 夜日切换级别, subType:string, [Low#低,norml#中,high#高]-->low
</nightToDayFilterLevel>
<nightToDayFilterTime>
  <!--opt, int, 夜日切换时间, range:[0,10]-->1
</nightToDayFilterTime>
<Schedule>
  <!--opt, object, 定时-->
  <scheduleType>
    <!--req, enum, 定时类型, subType:string, [day#白天,night#晚上]-->day
  </scheduleType>
  <TimeRange>
    <!--req, object, 时间段-->
    <beginTime>
      <!--req, time, 开始时间-->00:00:00+08:00
    </beginTime>
    <endTime>
      <!--req, time, 结束时间-->00:00:00+08:00
    </endTime>
  </TimeRange>
</Schedule>
<EventTrigger>
  <!--opt, object, 事件触发-->
  <eventType>
    <!--req, enum, 事件类型, subType:string, [IO#IO事件,VMD#VMD事件]-->IO
  </eventType>
  <IrcutFilterAction>
    <!--req, enum, IrcutFilter操作, subType:string, [day#白天,night#晚上]-->day
  </IrcutFilterAction>
</EventTrigger>
</IrcutFilter>
<Scene>
  <!--opt, object, 场景-->
  <mode>
    <!--req, enum, 模式, subType:string, [indoor#户内,outdoor#户外,standard#标准,dimLight#昏暗]-->indoor
  </mode>
</Scene>
<PTZ>
  <!--opt, object, PTZ-->
  <enabled>
    <!--opt, bool, 使能-->true
  </enabled>
</PTZ>
<EPTZ>
  <!--opt, object, EPTZ-->
  <enabled>
    <!--req, bool, 使能-->true
  </enabled>
  <mode>
    <!--opt, enum, 模式, subType:string, [cruise#巡航(云台可以支持预置点、巡航功能、ZOOM和8个方向),autoTrack#autoTrack-自动跟踪]-->cruise
  </mode>
</EPTZ>
<EIS>
  <!--opt, object, 电子防抖-->
  <enabled>
    <!--req, bool, 使能-->true
  </enabled>
  <EISLevel>
    <!--opt, int, EIS等级, range:[0,10]-->0
  </EISLevel>
</EIS>
<HLC>
  <!--opt, object, 强光抑制-->
  <enabled>
    <!--req, bool, 使能-->true
  </enabled>
  <HLCLevel>
    <!--opt, int, HLC等级, range:[0,10]-->0
  </HLCLevel>
</HLC>
<ZoomLimit>
  <!--opt, object, 缩放限制-->
  <ZoomLimitRatio>
    <!--opt, enum, 缩放比例, subType:int, [4#4,8#8,16#16,32#32,64#64]-->4
  </ZoomLimitRatio>
</ZoomLimit>
<corridor>
  <!--opt, object, 镜头走廊模式-->
  <enabled>
    <!--req, bool, 使能-->true
  </enabled>
</corridor>
<Dehaze>
  <!--opt, object, 雾透-->
  <DehazeMode>
    <!--opt, enum, 雾透模式, subType:string, [open#打开,close#关闭,auto#自动]-->open
  </DehazeMode>
  <DehazeLevel>
    <!--opt, int, 雾透等级, range:[0,10]-->0
  </DehazeLevel>
</Dehaze>
```

```
</enable>
<ImageMode>
  <!--opt, enum, 图像模式, subType:string, [standard#标准,indoor#室内,outdoor#室外,dimLight#昏暗]-->standard
</ImageMode>
<enableImageLossDetection>
  <!--opt, bool, 图像丢失检测使能-->true
</enableImageLossDetection>
<CaptureMode>
  <!--opt, object, 抓拍模式-->
  <mode>
    <!--req, enum, 模式, subType:string, [close#关闭,1920*1080@30fps#1920*1080@30fps]-->close
  </mode>
</CaptureMode>
<IrLight>
  <!--opt, object, 亮度限制-->
  <mode>
    <!--req, enum, 模式, subType:string, [auto#自动>manual#手动]-->auto
  </mode>
  <brightnessLevel>
    <!--opt, int, 亮度等级, range:[0,100]-->1
  </brightnessLevel>
  <sensitivityLevel>
    <!--opt, int, 灵敏度, range:[0,100]-->1
  </sensitivityLevel>
  <brightnessLimit>
    <!--opt, int, 亮度限制, range:[0,100]-->1
  </brightnessLimit>
  <triggerMode>
    <!--opt, enum, 触发模式, subType:string, [camera#相机,photosensitive#光敏]-->camera
  </triggerMode>
</IrLight>
<LensDistortionCorrection>
  <!--opt, object, 镜头畸变校正-->
  <enabled>
    <!--req, bool, 使能-->true
  </enabled>
  <level>
    <!--opt, enum, 等级, subType:string, [Level1#Level1,Level2#Level2,Level3#Level3,custom#自定义]-->level1
  </level>
  <accurateLevel>
    <!--opt, int, 畸变校正强度, range:[0,100]-->0
  </accurateLevel>
  <zoomedInDistantViewLevel>
    <!--opt, int, 远端放大, range:[0,100], desc:畸变校正开启时生效-->0
  </zoomedInDistantViewLevel>
  <horizontalFOV>
    <!--opt, int, 水平视场角, range:[0,100]-->0
  </horizontalFOV>
  <verticalFOV>
    <!--opt, int, 垂直视场角, range:[0,100]-->0
  </verticalFOV>
</LensDistortionCorrection>
<SupplementLight>
  <!--opt, object, 内置补光灯管理参数能力-->
  <mode>
    <!--opt, enum, 补光灯调节模式, subType:string, [schedule#白光定时,off#白光关闭,on#白光开启,auto#自动>manual#手动]-->auto
  </mode>
  <Schedule>
    <!--opt, object, 白光灯定时生效时间配置;当<Mode>= schedule时生效;所配置时间为白天的时间-->
    <TimeRange>
      <!--req, object, 白天时段;当天剩余时段为夜晚时段-->
      <beginTime>
        <!--req, time, 开始时间-->00:08:00+08:00
      </beginTime>
      <endTime>
        <!--req, time, 结束时间-->00:18:00+08:00
      </endTime>
    </TimeRange>
  </Schedule>
  <brightnessLimit>
    <!--opt, int, 白光补光灯亮度限制, range:[0,100], desc:当<Mode>= on时;可以调节该限制;为白光补光灯亮度最大值;实际补光灯亮度为自动调节 (注: mode里面的自动为补光灯开光自动;自动调节模式需查看mixedLightBrightnessRegulatMode节点) -->80
  </brightnessLimit>
  <supplementLightMode>
    <!--opt, enum, 补光灯模式, subType:string, [mixed#混合模式,whitelight#白光模式,close#关闭,colorVwWhitelight#全彩白光,irLight#红外模式,dualLight#双补光,eventIntelligence#事件智能补光,multiAzimuthMixed#多方位混合补光,multiAzimuthWhitelight#多方位白光模式,multiAzimuthIrLight#多方位红外模式,panoramicWhitelight#全景白光,panoramicIrLight#全景红外],
    desc:eventIntelligence为事件智能补光;解释如下:1.在满足补光灯功能生效条件下;当没有产生报警;设备仅红外灯开启;2.产生报警后自动关闭红外灯;开启白光灯持续1min;在持续时间内如继续收到报警;这一状态保持至最后一个报警发生后1min结束。multiAzimuthMixed多方位混合补光模式可以通过左右边各一组补光灯 (一组补光灯包含远中近白光红外补光灯) 对目标进行多方位角度的补光;不同于之前mixed混合补光模式仅支持一组补光灯单方位进行补光。multiAzimuthWhitelight多方位白光模式、multiAzimuthIrLight多方位红外模式与多方位混光情况相同。
    panoramcWhitelight、panoramicIrLight全景白光、红外模式当前仅四目全景相机支持,当前下相机无方位的概念,由四个补光灯围绕相机部署。-->mixed
  </supplementLightMode>
  <irLightBrightness>
    <!--opt, int, 红外补光灯亮度, range:[0,100], desc:当<supplementLightMode>= mixed;且<mixedLightBrightnessRegulatMode>= manual时生效-->80
  </irLightBrightness>
  <mixedLightBrightnessRegulatMode>
    <!--opt, enum, 混合补光灯亮度调节模式, subType:string, [manual#手动,auto#自动], desc:该字段仅在<supplementLightMode>= mixed时生效;手动模式下用户需手动<irLightBrightness>红外补光灯亮度参数以及<brightnessLimit>白光补光灯亮度;自动模式下用户无需配置亮度相关参数;由设备自动调节。-->auto
  </mixedLightBrightnessRegulatMode>
  <highIrLightBrightness>
    <!--opt, int, 远光红外光亮度配置, range:[0,100], desc:参数由mixed#混合模式、whitelight#白光模式、colorVwWhitelight#全彩白光,irLight#红外模式、dualLight#双补光模式公用-->80
  </highIrLightBrightness>
```

```

<highWhiteLightBrightness>
  <!--opt, int, 远光白光亮度配置, range:[0,100], desc:参数由mixed#混合模式、whiteLight#白光模式、colorVuWhiteLight#全彩白光,irLight#红外模式、dualLight#
  双补光模式公用-->80
</highWhiteLightBrightness>
<lowIrLightBrightness>
  <!--opt, int, 近光红外光亮度配置, range:[0,100], desc:参数由mixed#混合模式、whiteLight#白光模式、colorVuWhiteLight#全彩白光,irLight#红外模式、
  dualLight#双补光模式公用-->80
</lowIrLightBrightness>
<lowWhiteLightBrightness>
  <!--opt, int, 近光白光亮度配置, range:[0,100], desc:参数由mixed#混合模式、whiteLight#白光模式、colorVuWhiteLight#全彩白光,irLight#红外模式、dualLight#
  双补光模式公用-->80
</lowWhiteLightBrightness>
<whiteLightBrightness>
  <!--opt, int, 光亮度配置(不分远近光灯时控制), range:[0,100], desc:参数由mixed#混合模式、whiteLight#白光模式、colorVuWhiteLight#全彩白光,irLight#红外模
  式、dualLight#双补光模式、mode=auto 公用-->80
</whiteLightBrightness>
<whiteLightModeCfg>
  <!--opt, object, 白光模式参数配置, desc:当supplementLightMode == whiteLight生效-->
  <highWhiteLightBrightness>
    <!--opt, int, 远光白光亮度配置, range:[0,100]-->80
  </highWhiteLightBrightness>
  <lowWhiteLightBrightness>
    <!--opt, int, 近光白光亮度配置, range:[0,100]-->80
  </lowWhiteLightBrightness>
  <whiteLightBrightness>
    <!--opt, int, 白光亮度配置(不分远近光灯时控制), range:[0,100]-->80
  </whiteLightBrightness>
</whiteLightModeCfg>
<isAutoModeBrightnessCfg>
  <!--opt, bool, 是否支持自动模式下亮度调节;支持时亮度节点下发为自动调节的亮度最大值-->true
</isAutoModeBrightnessCfg>
<irLightbrightnessLimit>
  <!--opt, int, 红外灯亮度限制, range:[0,100], desc:当<Mode>== auto时且supplementLightMode==mixed模式下生效.-->80
</irLightbrightnessLimit>
<whiteLightbrightnessLimit>
  <!--opt, int, 白光灯亮度限制, range:[0,100], desc:当<Mode>== auto时且supplementLightMode==mixed模式下生效.-->80
</whiteLightbrightnessLimit>
<isNotSupportMixedWhiteLight>
  <!--opt, bool, 该节点返回true表示不支持混光模式下白光灯配置;支持配置则不返回该节点-->true
</isNotSupportMixedWhiteLight>
<EventIntelligenceModeCfg>
  <!--opt, object, 事件智能补光模式参数配置, desc:当supplementLightMode == eventIntelligence生效-->
  <brightnessRegulatMode>
    <!--opt, enum, 补光灯亮度调节模式, subType:string, [manual#手动,auto#自动], desc:手动模式下用户需手动配置补光灯亮度;自动模式下用户无需配置亮度相关
    参数;由设备自动调节.-->auto
  </brightnessRegulatMode>
  <whiteLightBrightness>
    <!--opt, int, 白光亮度配置(不分远近光灯时控制), range:[0,100]-->80
  </whiteLightBrightness>
  <irLightBrightness>
    <!--opt, int, 红外亮度配置(不分远近光灯时控制), range:[0,100]-->80
  </irLightBrightness>
  <BrightnessCtrlList>
    <!--opt, array, 远近光亮度控制参数, subType:object-->
    <BrightnessCtrl>
      <!--opt, object, 远近光亮度控制-->
      <lightType>
        <!--opt, enum, 补光灯类型, subType:string, [highWhite#远光白光补光灯,lowWhite#近光白光补光灯,highIrLight#远光红外补光灯,lowIrLight#近光红外补光
        灯]-->highWhite
      </lightType>
      <brightness>
        <!--opt, int, 补光灯亮度配置, range:[0,100]-->80
      </brightness>
    </BrightnessCtrl>
  </BrightnessCtrlList>
</EventIntelligenceModeCfg>
<MultiAzimuthMixedModeCfg>
  <!--opt, object, 多方位混合模式参数配置, desc:当supplementLightMode == multiAzimuthMixed生效-->
  <brightnessRegulatMode>
    <!--opt, enum, 补光灯亮度调节模式, subType:string, [manual#手动,auto#自动], desc:手动模式下用户需手动配置各组补光灯的亮度(LightCtrlList);自动模式
    下用户无需配置亮度相关参数;由设备自动调节.-->auto
  </brightnessRegulatMode>
  <LightCtrlList>
    <!--opt, array, 补光灯组控制参数, subType:object-->
    <LightCtrl>
      <!--opt, object, 单个补光灯控制参数-->
      <azimuthType>
        <!--opt, enum, 补光灯所在的方位类型, subType:string, [Left#左侧方位补光灯,right#右侧方位补光灯]-->left
      </azimuthType>
      <lightType>
        <!--opt, enum, 补光灯类型, subType:string, [highWhite#远光白光补光灯,midWhite#中光白光补光灯,lowWhite#近光白光补光灯,highIrLight#远光红外补光
        灯,midIrLight#中光红外补光灯,lowIrLight#近光红外补光灯]-->highWhite
      </lightType>
      <brightness>
        <!--opt, int, 补光灯亮度配置, range:[0,100]-->80
      </brightness>
    </LightCtrl>
  </LightCtrlList>
</MultiAzimuthMixedModeCfg>
<MultiAzimuthWhiteLightCfg>
  <!--opt, object, 多方位白光模式参数配置, desc:当supplementLightMode == multiAzimuthWhiteLight生效-->
  <brightnessRegulatMode>
    <!--opt, enum, 补光灯亮度调节模式, subType:string, [manual#手动,auto#自动], desc:手动模式下用户需手动配置各组补光灯的亮度(LightCtrlList);自动模式
    下用户无需配置亮度相关参数;由设备自动调节.-->auto
  </brightnessRegulatMode>
  <LightCtrlList>
    <!--opt, array, 补光灯组控制参数, subType:object-->
    <LightCtrl>
      <!--opt, object, 单个补光灯控制参数-->
      <azimuthType>
        <!--opt, enum, 补光灯所在的方位类型, subType:string, [Left#左侧方位补光灯,right#右侧方位补光灯]-->left
      </azimuthType>
      <lightType>
        <!--opt, enum, 补光灯类型, subType:string, [highWhite#远光白光补光灯,midWhite#中光白光补光灯,lowWhite#近光白光补光灯,highIrLight#远光红外补光
        灯,midIrLight#中光红外补光灯,lowIrLight#近光红外补光灯]-->highWhite
      </lightType>
      <brightness>
        <!--opt, int, 补光灯亮度配置, range:[0,100]-->80
      </brightness>
    </LightCtrl>
  </LightCtrlList>
</MultiAzimuthWhiteLightCfg>

```

下用户无需配置亮度相关参数;由设备自动调节.-->auto

```
</brightnessRegulatMode>
<LightCtrlList>
  <!--opt, array, 补光灯组控制参数, subType:object-->
  <LightCtrl>
    <!--opt, object, 单个补光灯控制参数-->
    <azimuthType>
      <!--opt, enum, 补光灯所在的方位类型, subType:string, [Left#左侧方位补光灯,right#右侧方位补光灯]-->left
    </azimuthType>
    <lightType>
      <!--opt, enum, 补光灯类型, subType:string, [highWhite#远光白光补光灯,midWhite#中光白光补光灯,lowWhite#近光白光补光灯,white#普通白光补光(不区分远中近光)]-->highWhite
    </lightType>
    <brightness>
      <!--opt, int, 补光灯亮度配置, range:[0,100]-->80
    </brightness>
  </LightCtrl>
</LightCtrlList>
</MultiAzimuthWhiteLightCfg>
<MultiAzimuthIrLightCfg>
  <!--opt, object, 多方位红外模式参数配置, desc:当supplementLightMode == multiAzimuthIrLight生效-->
  <brightnessRegulatMode>
    <!--opt, enum, 补光灯亮度调节模式, subType:string, [manual#手动,auto#自动], desc:手动模式下用户需手动配置各组补光灯的亮度（LightCtrlList）；自动模式下用户无需配置亮度相关参数;由设备自动调节.-->auto
  </brightnessRegulatMode>
  <LightCtrlList>
    <!--opt, array, 补光灯组控制参数, subType:object-->
    <LightCtrl>
      <!--opt, object, 单个补光灯控制参数-->
      <azimuthType>
        <!--opt, enum, 补光灯所在的方位类型, subType:string, [Left#左侧方位补光灯,right#右侧方位补光灯]-->left
      </azimuthType>
      <lightType>
        <!--opt, enum, 补光灯类型, subType:string, [highIrLight#远光红外补光灯,midIrLight#中光红外补光灯,lowIrLight#近光红外补光灯,irLight#普通红外补光灯(不区分远中近光)]-->highIrLight
      </lightType>
      <brightness>
        <!--opt, int, 补光灯亮度配置, range:[0,100]-->80
      </brightness>
    </LightCtrl>
  </LightCtrlList>
</MultiAzimuthIrLightCfg>
<PanoramicWhiteLightCfg>
  <!--opt, object, 全景白光模式参数配置, desc:当supplementLightMode == panoramicWhiteLight生效-->
  <brightnessRegulatMode>
    <!--opt, enum, 补光灯亮度调节模式, subType:string, [manual#手动,auto#自动], desc:手动模式下用户需手动配置各组补光灯的亮度（LightCtrlList）；自动模式下用户无需配置亮度相关参数;由设备自动调节.-->auto
  </brightnessRegulatMode>
  <LightCtrlList>
    <!--opt, array, 补光灯组控制参数, subType:object-->
    <LightCtrl>
      <!--opt, object, 单个补光灯控制参数-->
      <lightID>
        <!--opt, int, 补光灯序号, range:[1,4]-->1
      </lightID>
      <brightness>
        <!--opt, int, 补光灯亮度配置, range:[0,100]-->80
      </brightness>
    </LightCtrl>
  </LightCtrlList>
</PanoramicWhiteLightCfg>
<PanoramicIrLightCfg>
  <!--opt, object, 全景红外模式参数配置, desc:当supplementLightMode == panoramicIrLightCfg生效-->
  <brightnessRegulatMode>
    <!--opt, enum, 补光灯亮度调节模式, subType:string, [manual#手动,auto#自动], desc:手动模式下用户需手动配置各组补光灯的亮度（LightCtrlList）；自动模式下用户无需配置亮度相关参数;由设备自动调节.-->auto
  </brightnessRegulatMode>
  <LightCtrlList>
    <!--opt, array, 补光灯组控制参数, subType:object-->
    <LightCtrl>
      <!--opt, object, 单个补光灯控制参数-->
      <lightID>
        <!--opt, int, 补光灯序号, range:[1,4]-->1
      </lightID>
      <brightness>
        <!--opt, int, 补光灯亮度配置, range:[0,100]-->80
      </brightness>
    </LightCtrl>
  </LightCtrlList>
</PanoramicIrLightCfg>
<whiteLightEnhanceEnabled>
  <!--opt, bool, 白光亮度增强使能(不区分远光灯时控制), desc:开启增强使能后, 白光会进入强电流模式, 亮度会增强十倍, 同时也可通过$.SupplementLight.whiteLightBrightness参数去控制强电流模式下的灯光亮度。该参数不区分补光灯模式及白光灯类型.-->true
</whiteLightEnhanceEnabled>
<LightManualCtrlList>
  <!--opt, array, 补光灯组手动控制参数, subType:object, dep:and,{$.ImageChannel.SupplementLight.mode,eq>manual}, desc:与supplementLightMode无关-->
  <LightCtrl>
    <!--opt, object, 单个补光灯控制参数-->
    <lightType>
      <!--opt, enum, 补光灯类型, subType:string, [highIrLight#远光红外补光灯,midIrLight#中光红外补光灯,lowIrLight#近光红外补光灯,irLight#普通红外补光灯(不区分远中近光),highWhite#远光白光补光灯,midWhite#中光白光补光灯,lowWhite#近光白光补光灯,white#普通红外补光灯(不区分远中近光)]-->highIrLight
    </lightType>
    <brightness>
      <!--opt, int, 补光灯亮度配置, range:[0,100]-->80
    </brightness>
```

```
</brightness>
</LightCtrl>
</LightManualCtrlList>
</SupplementLight>
<OpticalDehaze>
  <!--opt, object, 光学雾透-->
  <enabled>
    <!--opt, bool, 使能-->true
  </enabled>
</OpticalDehaze>
<ManualRanging>
  <!--opt, object, 手动测距-->
  <enabled>
    <!--opt, bool, 使能-->true
  </enabled>
</ManualRanging>
<OIS>
  <!--opt, object, 光学防抖-->
  <mode>
    <!--req, enum, 模式, subType:string, [close#关闭,ordinary#普通,experts#专家]-->close
  </mode>
  <OISLevel>
    <!--opt, enum, 光学防抖等级, subType:string, [high#高,middle#中,low#低]-->high
  </OISLevel>
  <OISSensitivity>
    <!--opt, string, 光学防抖灵敏度-->test
  </OISSensitivity>
</OIS>
<isSupportLaserSpotManual>
  <!--opt, bool, 是否支持手动激光点-->true
</isSupportLaserSpotManual>
<isSupportLaserSpotAdjustment>
  <!--opt, bool, 是否支持激光点调整-->true
</isSupportLaserSpotAdjustment>
<DigitalZoom>
  <!--opt, object, 数字缩放-->
  <ZoomRatio>
    <!--opt, enum, 缩放比例, subType:string, [1x#1x,2x#2x,4x#4x,8x#8x,16x#16x,32x#32x]-->1x
  </ZoomRatio>
</DigitalZoom>
<Palettes>
  <!--opt, object, 热成像调色板配置-->
  <mode>
    <!--opt, enum, 模式, subType:string, [WhiteHot#白热,BlackHot#黑热,Fusion1#融合1,Rainbow#彩虹,Fusion2#融合2,Ironbow1#铁红1,Ironbow2#铁红2,Sepia#深褐色,Color1#色彩1,Color2#色彩2,IceFire#冰火,Rain#雨,RedHot#红热,GreenHot#绿热,DeepBlue#深蓝,Color3#色彩3]-->WhiteHot
  </mode>
  <ColorateTarget>
    <!--opt, object, 目标凸显, desc:dep mode = "WhiteHot"-->
  <ColorateTargetModelList>
    <!--opt, array, 目标凸显模式列表, subType:object-->
    <ColorateTargetMode>
      <!--opt, object, 目标凸显模式-->
      <id>
        <!--req, int, 序号, desc:从1开始-->0
      </id>
      <id>
      </id>
      <mode>
        <!--req, enum, 模式, subType:string, [colorateHotArea#大于(才凸显),colorateIntervalArea#区间(才凸显),colorateColdArea#小于(才凸显)]-->
      </mode>
      <enabled>
        <!--req, bool, 使能-->true
      </enabled>
      <TemperatureLimit>
        <!--req, object, 边界温度-->
        <minTemperature>
          <!--opt, float, 最低温度, range:[-20,150]-->0.000
        </minTemperature>
        <maxTemperature>
          <!--opt, float, 最高温度, range:[-20,150]-->0.000
        </maxTemperature>
      </TemperatureLimit>
      <Color>
        <!--req, object, 区域颜色-->
        <R>
          <!--req, int, R, range:[0,255]-->0
        </R>
        <G>
          <!--req, int, G, range:[0,255]-->0
        </G>
        <B>
          <!--req, int, B, range:[0,255]-->0
        </B>
      </Color>
    </ColorateTargetMode>
  </ColorateTargetModelList>
</ColorateTarget>
</Palettes>
<ExposureSync>
  <!--opt, object, 曝光同步-->
  <enabled>
    <!--req, bool, 使能-->true
  </enabled>
</ExposureSync>
```

```
<BrightnessSuddenChangeSuppression>
  <!--opt, object, 亮度突变抑制-->
  <enabled>
    <!--req, bool, 使能-->true
  </enabled>
</BrightnessSuddenChangeSuppression>
<AGCCap>
  <!--opt, object, AGC能力-->
  <AGCMode>
    <!--opt, enum, AGC模式, subType:string, [self_Adaptive#自适应,histogram#直方图,Linear#线性]-->self_Adaptive
  </AGCMode>
</AGCCap>
<RegionalImageEnhancementCap>
  <!--opt, object, 区域图像增强能力-->
  <mode>
    <!--req, enum, 模式, subType:string,
[off#off,customArea#customArea,up_60%#up_60%,down_60%#down_60%,Left_60%#Left_60%,right_60%#right_60%,center_50%#center_50%,center_75%#center_75%]-->off
  </mode>
  <CustomAreaParam>
    <!--opt, object, 自定义区域参数, desc:dep <mode>="customArea"-->
  <RegionCoordinatesList>
    <!--opt, array, 区域坐标列表, subType:object, range:[3,10]-->
  <RegionCoordinates>
    <!--opt, object, 区域坐标, desc:屏幕左上角为坐标原点-->
    <positionX>
      <!--req, int, x坐标, range:[0,1000]-->0
    </positionX>
    <positionY>
      <!--req, int, y坐标, range:[0,1000]-->0
    </positionY>
  </RegionCoordinates>
</RegionCoordinatesList>
</CustomAreaParam>
<DefaultAreaParamList>
  <!--opt, array, 默认区域参数列表, subType:object, desc:dep <mode>="up_60%,down_60%,Left_60%,right_60%,center_50%,center_75%"-->
  <DefaultAreaParam>
    <!--opt, object, 默认区域参数-->
    <mode>
      <!--req, enum, 模式, subType:string,
[up_60%#up_60%,down_60%#down_60%,Left_60%#Left_60%,right_60%#right_60%,center_50%#center_50%,center_75%#center_75%]-->up_60%
    </mode>
    <RegionCoordinatesList>
      <!--opt, array, 区域坐标列表, subType:object, range:[3,10]-->
    <RegionCoordinates>
      <!--opt, object, 区域坐标, desc:屏幕左上角为坐标原点-->
      <positionX>
        <!--req, int, x坐标, range:[0,1000]-->0
      </positionX>
      <positionY>
        <!--req, int, y坐标, range:[0,1000]-->0
      </positionY>
    </RegionCoordinates>
  </RegionCoordinatesList>
</DefaultAreaParam>
</DefaultAreaParamList>
</RegionalImageEnhancementCap>
<faceExposure>
  <!--opt, object, 人脸曝光-->
  <faceExposureEnabled>
    <!--req, bool, 人脸曝光使能-->true
  </faceExposureEnabled>
  <faceExposureSensitivity>
    <!--req, int, 人脸曝光灵敏度, range:[0,100]-->0
  </faceExposureSensitivity>
</faceExposure>
<lightHal>
  <!--opt, object, 信号灯光晕抑制-->
  <enabled>
    <!--req, bool, 使能-->true
  </enabled>
  <halLevel>
    <!--req, int, 光晕抑制等级, range:[0,100]-->0
  </halLevel>
  <halRange>
    <!--req, int, 光晕抑制范围, range:[0,2064]-->0
  </halRange>
</lightHal>
<TargetEnhancement>
  <!--opt, object, 图像目标增强配置-->
  <enabled>
    <!--req, bool, 使能-->true
  </enabled>
</TargetEnhancement>
<LaserLight>
  <!--opt, object, 激光补光-->
  <mode>
    <!--opt, enum, 模式, subType:string, [auto#自动>manual#手动]-->auto
  </mode>
  <compensationRegionMode>
    <!--opt, enum, 激光补光区域, subType:string, [center#中心,adaptive#适应,full_Screen#全屏]-->center
  </compensationRegionMode>
  <compensationLevel>
    <!--opt, enum, 激光补光强度, subType:string, [Low#低,medium#中,high#高]-->low
  </compensationLevel>
```

```
</ComponentControl>
<LaserColor>
  <!--opt, object, 激光颜色, desc: 依赖<mode>为manual-->
  <R>
    <!--req, int, R(红色)值, range:[0,100]-->0
  </R>
  <G>
    <!--req, int, G(绿色)值, range:[0,100]-->0
  </G>
  <B>
    <!--req, int, B(蓝色)值, range:[0,100]-->0
  </B>
</LaserColor>
</LaserLight>
<ExternalSupplementLight>
  <!--opt, object, 外接补光灯参数配置能力-->
  <mode>
    <!--opt, enum, 模式, subType:string, [close#关闭,oneMixedLight#一组混合补光]-->close
  </mode>
  <BrightnessList>
    <!--opt, array, 亮度列表, subType:object-->
    <Brightness>
      <!--opt, object, 亮度-->
      <id>
        <!--opt, int, 每组补光灯的id-->0
      </id>
      <irLightBrightness>
        <!--opt, int, 红外灯亮度, range:[0,100]-->0
      </irLightBrightness>
      <whiteLightBrightness>
        <!--opt, int, 白光灯亮度, range:[0,100]-->0
      </whiteLightBrightness>
    </Brightness>
  </BrightnessList>
</ExternalSupplementLight>
<TempRange>
  <!--opt, object, 温度范围-->
  <mode>
    <!--req, enum, 温度范围模式, subType:string, [automatic#自动模式>manual#手动模式]-->automatic
  </mode>
  <temperatureUpperLimit>
    <!--opt, float, 上限温度, range:[-20,550], unit:℃, unitType:温度, desc:精确到小数点后一位,手动温度范围模式下有效,上限温度需大于下限温度-->0.000
  </temperatureUpperLimit>
  <temperatureLowerLimit>
    <!--opt, float, 下限温度, range:[-20,550], unit:℃, unitType:温度, desc:精确到小数点后一位,手动温度范围模式下有效,下限温度需小于上限温度-->0.000
  </temperatureLowerLimit>
</TempRange>
<Beauty>
  <!--opt, object, 美颜能力-->
  <enabled>
    <!--req, bool, 使能-->true
  </enabled>
  <whiteningStrength>
    <!--opt, int, 美白强度, range:[0,10]-->1
  </whiteningStrength>
  <skinSmoothingStrength>
    <!--opt, int, 磨皮强度, range:[0,10]-->1
  </skinSmoothingStrength>
</Beauty>
<MExposure>
  <!--opt, object, 手动曝光(工厂标定)-->
  <enabled>
    <!--opt, bool, 使能-->true
  </enabled>
  <exposureLevel>
    <!--opt, string, 曝光等级, range:[0,1000000]-->test
  </exposureLevel>
  <gainLevel>
    <!--opt, int, 增益等级, range:[0,100]-->0
  </gainLevel>
</MExposure>
<SensorRGBCtrl>
  <!--opt, object, SensorRGB控制-->
  <sensorID>
    <!--opt, int, SensorID, range:[0,1]-->0
  </sensorID>
</SensorRGBCtrl>
<YPbPrCalibration>
  <!--opt, object, 色差标定-->
  <mode>
    <!--opt, int, 模式, range:[0,2]-->1
  </mode>
</YPbPrCalibration>
<intrusionTriggeredLightingCfg>
  <!--opt, object, 随影补光-->
  <enabled>
    <!--req, bool, 使能-->true
  </enabled>
</intrusionTriggeredLightingCfg>
<shotProCfg>
  <!--opt, object, 幕影-->
  <shotProMode>
    <!--req, enum, 幕影模式, subType:string, [close#关闭,general#通用,enhancement#增强]-->general
  </shotProMode>
</shotProCfg>
```

```
</shotProCfg>
</ImageChannel>
```

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 响应消息, attr:version{ro, req, string, 版本}-->
  <requestURL>
    <!--ro, req, string, 请求的URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, 状态码, subType:int, [0#OK,1#OK,2#Device Busy,3#Device Error,4#Invalid Operation,5#Invalid XML Format,6#Invalid XML Content,7#Reboot Required]-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, 状态信息, subType:string, [OK#成功,Device Busy#设备忙,Device Error#设备异常,Invalid Operation#无效的操作,Invalid XML Format#无效的XML格式,Invalid XML Content#无效的XML内容,Reboot#设备重启]-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, 详细错误码英文描述, desc:详细错误码的英文描述-->OK
  </subStatusCode>
</ResponseStatus>
```

69.24.9 获取指定通道图像处理参数

Request URL

GET /ISAPI/Image/channels/<channelID>

查询参数

参数名称	参数类型	描述
channelID	string	视频通道号

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<ImageChannel xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 图像信号处理参数, attr:version{req, string, 协议版本}-->
  <id>
    <!--ro, req, int, 序号ID-->1
  </id>
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <videoInputID>
    <!--ro, req, int, 视频通道号ID-->1
  </videoInputID>
  <Focusconfiguration>
    <!--ro, opt, object, 普通聚焦-->
    <focusStyle>
      <!--ro, req, enum, 聚焦类型, subType:string, [AUTO#自动,MANUAL#手动,SEMIAUTOMATIC#半自动]-->AUTO
    </focusStyle>
    <focusLimited>
      <!--ro, opt, enum, 聚焦限制, subType:int,
      [10#10,50#50,100#100,150#150,300#300,600#600,1000#1000,2000#2000,5000#5000,20000#20000,50000#50000,65535#65535]-->10
    </focusLimited>
    <focusPosition>
      <!--ro, opt, int, 焦点位置, range:[1,10]-->1
    </focusPosition>
    <focusSpeed>
      <!--ro, opt, int, 聚焦速度, range:[1,3]-->1
    </focusSpeed>
    <focusSensitivity>
      <!--ro, opt, int, 聚焦灵敏度, range:[0,2]-->1
    </focusSensitivity>
    <temperatureChangeAdaptEnabled>
      <!--ro, opt, bool, 温度变化适应使能-->true
    </temperatureChangeAdaptEnabled>
    <relativeFocusPos>
      <!--ro, opt, int, 相对焦点位置, range:[0,4294967295]-->1
    </relativeFocusPos>
    <lensType>
      <!--ro, opt, enum, 聚焦镜头类型, subType:string, [1140#1140,1050#1050]-->1140
    </lensType>
    <highTemperaturePriority>
```

```
<!--ro, opt, bool, 高温优先模式使能-->true
</highTemperaturePriority>
<focusRange>
  <!--ro, opt, enum, 聚焦调节范围, subType:string, [min#最小,medium#适中,max#最大]-->max
</focusRange>
<foucusDistanceMode>
  <!--ro, opt, enum, 聚焦距离模式, subType:string, [compatible#兼容模式], desc:compatible-兼容模式, 在球罩设备上调整聚焦限制距离focusLimited,仍然虚焦的情况, 选择兼容模式, focusLimited参数失效, 设备释放镜头最大聚焦能力, 使镜头获得更大的焦距范围, 解决球罩设备近物距聚焦效果差的问题。开启该功能, 聚焦速度会变慢, 非球罩设备不建议使用。-->compatible
</foucusDistanceMode>
</Focusconfiguration>
<LensInitialization>
  <!--ro, opt, object, 镜头初始化-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
</LensInitialization>
<ImageFlip>
  <!--ro, opt, object, 图像翻转-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <ImageFlipStyle>
    <!--ro, opt, enum, 镜像类型, subType:string, [LEFTRIGHT#LEFTRIGHT,UPDOWN#UPDOWN,CENTER#CENTER,AUTO#AUTO]-->LEFTRIGHT
  </ImageFlipStyle>
  <flipAngle>
    <!--ro, opt, enum, 旋转角度, subType:int, [90#90,180#180,270#270]-->90
  </flipAngle>
</ImageFlip>
<ImageFreeze>
  <!--ro, opt, object, 图像冻结-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
</ImageFreeze>
<proportionalpan>
  <!--ro, opt, object, 比例盘-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
</proportionalpan>
<WDR>
  <!--ro, opt, object, 宽动态(自动)能力-->
  <mode>
    <!--ro, req, enum, open, subType:string, [open#开启,close#关闭,auto#自动,SDR#SDR]-->open
  </mode>
  <WDRLevel>
    <!--ro, opt, int, 宽动态等级, range:[1,100]-->1
  </WDRLevel>
  <WDRContrastLevel>
    <!--ro, opt, int, 宽动态对比度, range:[1,100]-->1
  </WDRContrastLevel>
  <WDRLevel1>
    <!--ro, opt, int, 第二类宽动态等级, range:[1,100]-->1
  </WDRLevel1>
</WDR>
<BLC>
  <!--ro, opt, object, 背光补偿区域-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <BLCMode>
    <!--ro, opt, enum, 背光补偿模式, subType:string, [UP#UP,DOWN#DOWN,LEFT#LEFT,RIGHT#RIGHT,CENTER#CENTER,MULTI-AREA#MULTI-AREA,Region#Region,AUTO#AUTO]-->UP
  </BLCMode>
  <BLCLevel>
    <!--ro, opt, int, 背光补偿等级, range:[0,10]-->0
  </BLCLevel>
  <BLCRegionList>
    <!--ro, opt, array, 背光补偿区域列表, subType:object-->
    <BLCRegion>
      <!--ro, opt, object, 背光补偿区域-->
      <id>
        <!--ro, req, int, 区域编号, range:[1,10]-->1
      </id>
      <RegionCoordinatesList>
        <!--ro, req, array, 坐标点列表, subType:object, range:[0,4], desc:矩形-->
        <RegionCoordinates>
          <!--ro, opt, object, 坐标点, desc:屏幕左下角为坐标原点-->
          <positionX>
            <!--ro, req, int, x坐标, range:[0,704]-->1
          </positionX>
          <positionY>
            <!--ro, req, int, y坐标, range:[0,576]-->1
          </positionY>
        </RegionCoordinates>
      </RegionCoordinatesList>
    </BLCRegion>
  </BLCRegionList>
</BLC>
<NoiseReduce>
  <!--ro, opt, object, 降噪参数-->
  <mode>
```

```
<!--ro, req, enum, 模式, subType:string, [close#关闭,general#常规,advanced#高级]-->close
</mode>
<GeneralMode>
  <!--ro, opt, object, 模式-->
  <generalLevel>
    <!--ro, req, int, 等级, range:[0,100]-->1
  </generalLevel>
</GeneralMode>
<AdvancedMode>
  <!--ro, opt, object, 高级模式-->
  <FrameNoiseReduceLevel>
    <!--ro, req, int, 框架噪声降阶, range:[0,100]-->1
  </FrameNoiseReduceLevel>
  <InterFrameNoiseReduceLevel>
    <!--ro, req, int, 帧间噪声降维, range:[0,100]-->1
  </InterFrameNoiseReduceLevel>
  <secondarySpatialNoiseReduceLevel>
    <!--ro, opt, int, 二级降噪空域等级, range:[0,100]-->1
  </secondarySpatialNoiseReduceLevel>
  <secondaryTemporalNoiseReduceLevel>
    <!--ro, opt, int, 二级降噪时域等级, range:[0,100]-->1
  </secondaryTemporalNoiseReduceLevel>
</AdvancedMode>
</NoiseReduce>
<ImageEnhancement>
  <!--ro, opt, object, 图像增强-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <ImageEnhancementLevel>
    <!--ro, opt, int, 图像增强等级, range:[0,10]-->0
  </ImageEnhancementLevel>
</ImageEnhancement>
<DSS>
  <!--ro, opt, object, DSS-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <DSSLevel>
    <!--ro, opt, enum, 低照度等级, subType:string,
[*1.25#*1.25,*1.5#*1.5,*2#*2,*3#*3,*4#*4,*5#*5,*6#*6,*8#*8,*12#*12,*16#*16,*24#*24,*32#*32,*48#*48,*64#*64,*96#*96,*128#*128,*256#*256,*512#*512,auto#auto]-
->*1.25
  </DSSLevel>
  <mode>
    <!--ro, opt, enum, 模式, subType:string, [frame#帧模式,Level#等级模式], desc:无mode节点默认为frame模式-->frame
  </mode>
  <slowShutterLevel>
    <!--ro, opt, int, 慢快门等级, range:[1,6]-->1
  </slowShutterLevel>
</DSS>
<WhiteBalance>
  <!--ro, opt, object, 白平衡-->
  <WhiteBalanceStyle>
    <!--ro, opt, enum, 白平衡类型, subType:string, [auto#自动,auto1#自动1,auto2#自动2>manual#手动,indoor#室内,outdoor#室外,autotrace#自动跟踪,once#单
次,sodiumLight#苏打光,mercuryLight#水银灯,customColorTemperature#自定义色温]-->auto
  </WhiteBalanceStyle>
  <WhiteBalanceRed>
    <!--ro, opt, int, 白平衡红, range:[0,100]-->1
  </WhiteBalanceRed>
  <WhiteBalanceBlue>
    <!--ro, opt, int, 白平衡蓝, range:[0,100]-->1
  </WhiteBalanceBlue>
  <WhiteBalanceLevel>
    <!--ro, opt, int, 白平衡等级, range:[0,100], desc:<WhiteBalanceStyle>为auto,auto1,auto2时有效-->1
  </WhiteBalanceLevel>
  <customColorTemperatureLevel>
    <!--ro, opt, int, 自定义色温等级, range:[2700,7500]-->1
  </customColorTemperatureLevel>
</WhiteBalance>
<Exposure>
  <!--ro, opt, object, 曝光模式-->
  <ExposureType>
    <!--ro, req, enum, 镜头类型, subType:string,
[auto#auto,IrisFirst#IrisFirst,ShutterFirst#ShutterFirst,gainFirst#gainFirst>manual#manual,pIris#pIris,T5280-PQ1#T5280-PQ1,T5289-PQ1#T5289-PQ1,T1140-
PQ1#T1140-PQ1,T2712-PQ1#T2712-PQ1,HV1250P-MPIR#HV1250P-MPIR,pc#pc,pIris-General#pIris-General,HZ1140P-8MPIRA#HZ1140P-8MPIRA,HZ1135P-8MPSGA#HZ1135P-
8MPSGA,MV1550P-12MPIR#MV1550P-12MPIR,MZ1545P-12MP#MZ1545P-12MP,HZ0618P-4MPSG#HZ0618P-4MPSG,MZ1050P-12MP#MZ1050P-12MP]-->auto
  </ExposureType>
  <autoIrisLevel>
    <!--ro, opt, int, 光圈自动级别, range:[0,100]-->1
  </autoIrisLevel>
  <OverexposeSuppress>
    <!--ro, opt, object, 过度曝光抑制-->
    <enabled>
      <!--ro, req, bool, 使能-->true
    </enabled>
  </OverexposeSuppress>
  <Type>
    <!--ro, opt, enum, 过度曝光抑制类型, subType:string, [AUTO#自动,MANUAL#手动]-->AUTO
  </Type>
  <DistanceLevel>
    <!--ro, opt, int, 焦距, range:[1,100]-->1
  </DistanceLevel>
  <shortIRDistanceLevel>
    <!--ro, opt, int, IR短距离级别, range:[1,100]-->1
  </shortIRDistanceLevel>
```

```
</longIRDistanceLevel>
<!--ro, opt, int, IR长距离级别, range:[1,100]-->1
</longIRDistanceLevel>
<supplementLightIntensity1>
<!--ro, opt, int, 补充曝光强度1, range:[1,100]-->1
</supplementLightIntensity1>
<supplementLightIntensity2>
<!--ro, opt, int, 补充曝光强度2, range:[1,100]-->1
</supplementLightIntensity2>
<supplementLightIntensity3>
<!--ro, opt, int, 补充曝光强度3, range:[1,100]-->1
</supplementLightIntensity3>
<supplementLightIntensity4>
<!--ro, opt, int, 补充曝光强度4, range:[1,100]-->1
</supplementLightIntensity4>
</OverexposeSuppress>
<pIris>
<!--ro, opt, object, 光圈-->
<pIrisType>
<!--ro, opt, enum, 光圈类型, subType:string, [AUTO#AUTO,MANUAL#MANUAL]-->MANUAL
</pIrisType>
<IrisLevel>
<!--ro, opt, int, 光圈级别, range:[1,100]-->1
</IrisLevel>
</pIris>
<faceExposure>
<!--ro, opt, object, 人脸曝光-->
<enabled>
<!--ro, opt, bool, 使能-->true
</enabled>
<sensitivity>
<!--ro, opt, int, 灵敏度, range:[0,100]-->0
</sensitivity>
</faceExposure>
<PIrisGeneral>
<!--ro, opt, object, pIris-General配置-->
<irisLevel>
<!--ro, opt, int, 通用镜头下的光圈等级, range:[1,100]-->1
</irisLevel>
</PIrisGeneral>
</Exposure>
<Sharpness>
<!--ro, opt, object, 锐度-->
<SharpnessMode>
<!--ro, opt, enum, 锐度模式, subType:string, [manual#手动,auto#自动]-->manual
</SharpnessMode>
<SharpnessLevel>
<!--ro, req, int, 锐度级别, range:[0,100]-->1
</SharpnessLevel>
</Sharpness>
<gammaCorrection>
<!--ro, opt, object, 伽玛校正-->
<gammaCorrectionEnabled>
<!--ro, opt, bool, 伽玛校正使能-->true
</gammaCorrectionEnabled>
<gammaCorrectionLevel>
<!--ro, opt, int, 伽玛校正级别, range:[0,100]-->1
</gammaCorrectionLevel>
</gammaCorrection>
<Shutter>
<!--ro, opt, object, 快门配置-->
<ShutterLevel>
<!--ro, opt, enum, 曝光等级, subType:string,
[1/1#1/1,1/2#1/2,1/3#1/3,1/6#1/6,1/12#1/12,1/25#1/25,1/50#1/50,1/75#1/75,1/100#1/100,1/120#1/120,1/125#1/125,1/150#1/150,1/175#1/175], desc:depends on
<ExposureType>-->1/1
</ShutterLevel>
<maxShutterLevelLimit>
<!--ro, opt, enum, 最大曝光等级, subType:string,
[1/1#1/1,1/2#1/2,1/3#1/3,1/6#1/6,1/12#1/12,1/25#1/25,1/50#1/50,1/75#1/75,1/100#1/100,1/120#1/120,1/125#1/125,1/150#1/150,1/175#1/175]-->1/1
</maxShutterLevelLimit>
<minShutterLevelLimit>
<!--ro, opt, enum, 最小曝光等级, subType:string,
[1/1#1/1,1/2#1/2,1/3#1/3,1/6#1/6,1/12#1/12,1/25#1/25,1/50#1/50,1/75#1/75,1/100#1/100,1/120#1/120,1/125#1/125,1/150#1/150,1/175#1/175]-->1/1
</minShutterLevelLimit>
</Shutter>
<powerLineFrequency>
<!--ro, opt, object, 视频制式-->
<powerLineFrequencyMode>
<!--ro, opt, enum, 视频制式模式, subType:string, [50hz#50hz,60hz#60hz]-->50hz
</powerLineFrequencyMode>
</powerLineFrequency>
<Gain>
<!--ro, opt, object, 增益-->
<GainLevel>
<!--ro, opt, string, 增益等级, range:[0,10]-->test
</GainLevel>
<GainLimit>
<!--ro, opt, enum, 增益, subType:string,
[0#0,400#400,900#900,1300#1300,1800#1800,2200#2200,2600#2600,3100#3100,3500#3500,4000#4000,4400#4400,4800#4800,5300#5300,5700#5700,6200#6200,6600#6600]-->0
</GainLimit>
</Gain>
<Color>
<!--ro, opt, object, 颜色-->
```

```

<brightnessLevel>
  <!--ro, opt, int, 亮度, range:[0,100]-->0
</brightnessLevel>
<contrastLevel>
  <!--ro, opt, int, 对比度, range:[0,100]-->0
</contrastLevel>
<saturationLevel>
  <!--ro, opt, int, 饱和度, range:[0,100]-->0
</saturationLevel>
<hueLevel>
  <!--ro, opt, int, 色调, range:[0,100]-->0
</hueLevel>
<grayScale>
  <!--ro, opt, object, 灰度-->
  <grayScaleMode>
    <!--ro, opt, enum, 灰度模式, subType:string, [indoor#室内,outdoor#室外]-->indoor
  </grayScaleMode>
</grayScale>
<nightMode>
  <!--ro, opt, bool, 夜间黑白模式使能-->true
</nightMode>
<sharpnessLevel>
  <!--ro, opt, int, 锐度, range:[0,100]-->0
</sharpnessLevel>
</Color>
<IrcutFilter>
  <!--ro, opt, object, IrcutFilter配置-->
  <IrcutFilterType>
    <!--ro, opt, enum, IrcutFilter类型, subType:string,
[auto#auto,day#day,night#night,schedule#schedule,eventTrigger#eventTrigger,videoAuto#videoAuto,darkFighterX#darkFighterX,darkFighterXAuto#darkFighterXAuto,d
arkFighterXSchedule#darkFighterXSchedule]-->auto
  </IrcutFilterType>
  <dayToNightFilterLevel>
    <!--ro, opt, enum, 日夜切换级别, subType:string, [Low#低,normal#中,high#高]-->low
  </dayToNightFilterLevel>
  <dayToNightFilterTime>
    <!--ro, opt, int, 日夜切换时间, range:[0,10]-->1
  </dayToNightFilterTime>
  <nightToDayFilterLevel>
    <!--ro, opt, enum, 夜日切换级别, subType:string, [Low#低,normal#中,high#高]-->low
  </nightToDayFilterLevel>
  <nightToDayFilterTime>
    <!--ro, opt, int, 夜日切换时间, range:[0,10]-->1
  </nightToDayFilterTime>
  <Schedule>
    <!--ro, opt, object, 定时-->
    <scheduleType>
      <!--ro, req, enum, 定时类型, subType:string, [day#白天,night#晚上]-->day
    </scheduleType>
    <TimeRange>
      <!--ro, req, object, 时间段-->
      <beginTime>
        <!--ro, req, time, 开始时间-->00:00:00+08:00
      </beginTime>
      <endTime>
        <!--ro, req, time, 结束时间-->00:00:00+08:00
      </endTime>
    </TimeRange>
  </Schedule>
  <EventTrigger>
    <!--ro, opt, object, 事件触发-->
    <eventType>
      <!--ro, req, enum, 事件类型, subType:string, [IO#IO事件,VMD#VMD事件]-->IO
    </eventType>
    <IrcutFilterAction>
      <!--ro, req, enum, IrcutFilter操作, subType:string, [day#白天,night#晚上]-->day
    </IrcutFilterAction>
  </EventTrigger>
</IrcutFilter>
<Scene>
  <!--ro, opt, object, 场景-->
  <mode>
    <!--ro, req, enum, 模式, subType:string, [indoor#户内,outdoor#户外,standard#标准,dimLight#昏暗]-->indoor
  </mode>
</Scene>
<PTZ>
  <!--ro, opt, object, PTZ-->
  <enabled>
    <!--ro, opt, bool, 使能-->true
  </enabled>
</PTZ>
<EPTZ>
  <!--ro, opt, object, EPTZ-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <mode>
    <!--ro, opt, enum, 模式, subType:string, [cruise#巡航(云台可以支持预置点、巡航功能、ZOOM和8个方向),autoTrack#autoTrack-自动跟踪]-->cruise
  </mode>
</EPTZ>
<EIS>
  <!--ro, opt, object, 电子防抖-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>

```

```
<!--ro, req, bool, 使能-->true
</enabled>
<EISLevel>
  <!--ro, opt, int, EIS等级, range:[0,10]-->0
</EISLevel>
</EIS>
<HLC>
  <!--ro, opt, object, 强光抑制-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <HLCLevel>
    <!--ro, opt, int, HLC等级, range:[0,10]-->0
  </HLCLevel>
</HLC>
<ZoomLimit>
  <!--ro, opt, object, 缩放限制-->
  <ZoomLimitRatio>
    <!--ro, opt, enum, 缩放比例, subType:int, [4#4,8#8,16#16,32#32,64#64]-->4
  </ZoomLimitRatio>
</ZoomLimit>
<corridor>
  <!--ro, opt, object, 镜头走廊模式-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
</corridor>
<Dehaze>
  <!--ro, opt, object, 雾透-->
  <DehazeMode>
    <!--ro, opt, enum, 雾透模式, subType:string, [open#打开,close#关闭,auto#自动]-->open
  </DehazeMode>
  <DehazeLevel>
    <!--ro, opt, int, 雾透等级, range:[0,10]-->0
  </DehazeLevel>
</Dehaze>
<ImageMode>
  <!--ro, opt, enum, 图像模式, subType:string, [standard#标准,indoor#室内,outdoor#室外,dimLight#昏暗]-->standard
</ImageMode>
<enableImageLossDetection>
  <!--ro, opt, bool, 图像丢失检测使能-->true
</enableImageLossDetection>
<CaptureMode>
  <!--ro, opt, object, 抓拍模式-->
  <mode>
    <!--ro, req, enum, 模式, subType:string, [close#关闭,1920*1080@30fps#1920*1080@30fps]-->close
  </mode>
</CaptureMode>
<IrLight>
  <!--ro, opt, object, 亮度限制-->
  <mode>
    <!--ro, req, enum, 模式, subType:string, [auto#自动>manual#手动]-->auto
  </mode>
  <brightnessLevel>
    <!--ro, opt, int, 亮度等级, range:[0,100]-->1
  </brightnessLevel>
  <sensitivityLevel>
    <!--ro, opt, int, 灵敏度, range:[0,100]-->1
  </sensitivityLevel>
  <brightnessLimit>
    <!--ro, opt, int, 亮度限制, range:[0,100]-->1
  </brightnessLimit>
  <triggerMode>
    <!--ro, opt, enum, 触发模式, subType:string, [camera#相机,photosensitive#光敏]-->camera
  </triggerMode>
</IrLight>
<LensDistortionCorrection>
  <!--ro, opt, object, 镜头畸变校正-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <level>
    <!--ro, opt, enum, 等级, subType:string, [Level1#Level1,Level2#Level2,Level3#Level3,custom#自定义]-->level1
  </level>
  <accurateLevel>
    <!--ro, opt, int, 畸变校正强度, range:[0,100]-->0
  </accurateLevel>
  <zoomedInDistantViewLevel>
    <!--ro, opt, int, 远端放大, range:[0,100], desc:畸变校正开启时生效-->0
  </zoomedInDistantViewLevel>
  <horizontalFOV>
    <!--ro, opt, int, 水平视场角, range:[0,100]-->0
  </horizontalFOV>
  <verticalFOV>
    <!--ro, opt, int, 垂直视场角, range:[0,100]-->0
  </verticalFOV>
</LensDistortionCorrection>
<SupplementLight>
  <!--ro, opt, object, 内置补光灯管理参数能力-->
  <mode>
    <!--ro, opt, enum, 补光灯调节模式, subType:string, [schedule#白光定时,off#白光关闭,on#白光开启,auto#自动>manual#手动]-->auto
  </mode>
  <Schedule>
    <!--ro, opt, object, 白光灯定时生效时间配置;当<Mode>= schedule时生效;所配置时间为白天的时间-->
```

```
<TimeRange>
  <!--ro, req, object, 白天时段;当天剩余时段为夜晚时段-->
  <beginTime>
    <!--ro, req, time, 开始时间-->00:08:00+08:00
  </beginTime>
  <endTime>
    <!--ro, req, time, 结束时间-->00:18:00+08:00
  </endTime>
</TimeRange>
</Schedule>
<brightnessLimit>
  <!--ro, opt, int, 白光补光灯亮度限制, range:[0,100], desc:当<Mode>= on时;可以调节该限制;为白光补光灯亮度最大值;实际补光灯亮度为自动调节（注：mode里面的自动为补光灯开关自动;自动调节模式需查看mixedLightBrightnessRegulateMode节点）-->80
</brightnessLimit>
<supplementLightMode>
  <!--ro, opt, enum, 补光灯模式, subType:string, [mixed#混合模式,whiteLight#白光模式,close#关闭,colorVulWhiteLight#全彩白光,irLight#红外模式,dualLight#双补光,eventIntelligence#事件智能补光,multiAzimuthMixed#多方位混合补光,multiAzimuthWhiteLight#多方位白光模式,multiAzimuthIrLight#多方位红外模式,panoramicWhiteLight#全景白光,panoramicIrLight#全景红外],
  desc:eventIntelligence为事件智能补光;解释如下:1.在满足补光灯功能生效条件下;当没有产生报警;设备仅红外灯开启;2.产生报警后自动关闭红外灯;开启白光灯持续1min;在持续时间内如继续收到报警;这一状态保持至最后一个报警发生后1min结束。multiAzimuthMixed多方位混合模式可以通过左右边各一组补光灯（一组补光灯包含远中近白光红外补光灯）对目标进行多方位角度的补光;不同于之前mixed混合补光模式仅支持一组补光灯单方位进行补光。multiAzimuthWhiteLight多方位白光模式、multiAzimuthIrLight多方位红外模式与多方位混光情况相同。
  panoramicWhiteLight、panoramicIrLight全景白光、红外模式当前仅四目全景相机支持,当前下相机无方位的概念,由四个补光灯围绕相机部署。-->mixed
  </supplementLightMode>
  <irLightBrightness>
    <!--ro, opt, int, 红外补光灯亮度, range:[0,100], desc:当<supplementLightMode>= mixed;且<mixedLightBrightnessRegulateMode>= manual时生效-->80
  </irLightBrightness>
  <mixedLightBrightnessRegulateMode>
    <!--ro, opt, enum, 混合补光灯亮度调节模式, subType:string, [manual#手动,auto#自动], desc:该字段仅在<supplementLightMode>= mixed时生效;手动模式下用户需手动<irLightBrightness>红外补光灯亮度参数以及<brightnessLimit>白光补光灯亮度;自动模式下用户无需配置亮度相关参数;由设备自动调节。-->auto
  </mixedLightBrightnessRegulateMode>
  <highIrLightBrightness>
    <!--ro, opt, int, 远光红外光亮度配置, range:[0,100], desc:参数由mixed#混合模式、whiteLight#白光模式、colorVulWhiteLight#全彩白光,irLight#红外模式、dualLight#双补光模式公用-->80
  </highIrLightBrightness>
  <highWhiteLightBrightness>
    <!--ro, opt, int, 远光白光亮度配置, range:[0,100], desc:参数由mixed#混合模式、whiteLight#白光模式、colorVulWhiteLight#全彩白光,irLight#红外模式、dualLight#双补光模式公用-->80
  </highWhiteLightBrightness>
  <lowIrLightBrightness>
    <!--ro, opt, int, 近光红外光亮度配置, range:[0,100], desc:参数由mixed#混合模式、whiteLight#白光模式、colorVulWhiteLight#全彩白光,irLight#红外模式、dualLight#双补光模式公用-->80
  </lowIrLightBrightness>
  <lowWhiteLightBrightness>
    <!--ro, opt, int, 近光白光亮度配置, range:[0,100], desc:参数由mixed#混合模式、whiteLight#白光模式、colorVulWhiteLight#全彩白光,irLight#红外模式、dualLight#双补光模式公用-->80
  </lowWhiteLightBrightness>
  <whiteLightBrightness>
    <!--ro, opt, int, 光亮度配置(不分远近光灯时控制), range:[0,100], desc:参数由mixed#混合模式、whiteLight#白光模式、colorVulWhiteLight#全彩白光,irLight#红外模式、dualLight#双补光模式、mode=auto 公用-->80
  </whiteLightBrightness>
  <whiteLightModeCfg>
    <!--ro, opt, object, 白光模式参数配置, desc:当supplementLightMode == whiteLight生效-->
    <highWhiteLightBrightness>
      <!--ro, opt, int, 远光白光亮度配置, range:[0,100]-->80
    </highWhiteLightBrightness>
    <lowWhiteLightBrightness>
      <!--ro, opt, int, 近光白光亮度配置, range:[0,100]-->80
    </lowWhiteLightBrightness>
    <whiteLightBrightness>
      <!--ro, opt, int, 白光亮度配置(不分远近光灯时控制), range:[0,100]-->80
    </whiteLightBrightness>
  </whiteLightModeCfg>
  <isAutoModeBrightnessCfg>
    <!--ro, opt, bool, 是否支持自动模式下亮度调节;支持时亮度调节点下发为自动调节的亮度最大值-->true
  </isAutoModeBrightnessCfg>
  <irLightbrightnessLimit>
    <!--ro, opt, int, 红外灯亮度限制, range:[0,100], desc:当<Mode>== auto时且supplementLightMode==mixed模式下生效。-->80
  </irLightbrightnessLimit>
  <whiteLightbrightnessLimit>
    <!--ro, opt, int, 白光灯亮度限制, range:[0,100], desc:当<Mode>== auto时且supplementLightMode==mixed模式下生效。-->80
  </whiteLightbrightnessLimit>
  <isNotSupportMixedWhiteLight>
    <!--ro, opt, bool, 该节点返回true表示不支持混光模式下白光灯配置;支持配置则不返回该节点-->true
  </isNotSupportMixedWhiteLight>
  <EventIntelligenceModeCfg>
    <!--ro, opt, object, 事件智能补光模式参数配置, desc:当supplementLightMode == eventIntelligence生效-->
    <brightnessRegulateMode>
      <!--ro, opt, enum, 补光灯亮度调节模式, subType:string, [manual#手动,auto#自动], desc:手动模式下用户需手动配置补光灯亮度;自动模式下用户无需配置亮度相关参数;由设备自动调节。-->auto
    </brightnessRegulateMode>
    <whiteLightBrightness>
      <!--ro, opt, int, 白光亮度配置(不分远近光灯时控制), range:[0,100]-->80
    </whiteLightBrightness>
    <irLightBrightness>
      <!--ro, opt, int, 红外亮度配置(不分远近光灯时控制), range:[0,100]-->80
    </irLightBrightness>
    <BrightnessCtrlList>
      <!--ro, opt, array, 远近光亮度控制参数, subType:object-->
      <BrightnessCtrl>
        <!--ro, opt, object, 远近光亮度控制-->
      </BrightnessCtrl>
    </BrightnessCtrlList>
    <lightType>
      <!--ro, opt, enum, 补光灯类型, subType:string, [highWhite#远光白光补光灯,lowWhite#近光白光补光灯,highIrLight#远光红外补光灯,lowIrLight#近光红外补光灯]-->
    </lightType>
  </EventIntelligenceModeCfg>
```

```

补光灯J-->highWhite
  </LightType>
  <brightness>
    <!--ro, opt, int, 补光灯亮度配置, range:[0,100]-->80
  </brightness>
</BrightnessCtrl>
</BrightnessCtrlList>
</EventIntelligenceModeCfg>
<MultiAzimuthMixedModeCfg>
  <!--ro, opt, object, 多方位混合模式参数配置, desc: 当supplementLightMode == multiAzimuthMixed生效-->
  <brightnessRegulatMode>
    <!--ro, opt, enum, 补光灯亮度调节模式, subType:string, [manual#手动,auto#自动], desc: 手动模式下用户需手动配置各组补光灯的亮度（LightCtrlList）；自动模式下用户无需配置亮度相关参数;由设备自动调节.-->auto
  </brightnessRegulatMode>
  <LightCtrlList>
    <!--ro, opt, array, 补光灯组控制参数, subType:object-->
    <LightCtrl>
      <!--ro, opt, object, 单个补光灯控制参数-->
      <azimuthType>
        <!--ro, opt, enum, 补光灯所在的方位类型, subType:string, [left#左侧方位补光灯,right#右侧方位补光灯]-->left
      </azimuthType>
      <lightType>
        <!--ro, opt, enum, 补光灯类型, subType:string, [highWhite#远光白光补光灯,midWhite#中光白光补光灯,lowWhite#近光白光补光灯,highIrLight#远光红外补光灯,midIrLight#中光红外补光灯,lowIrLight#近光红外补光灯]-->highWhite
      </lightType>
      <brightness>
        <!--ro, opt, int, 补光灯亮度配置, range:[0,100]-->80
      </brightness>
    </LightCtrl>
  </LightCtrlList>
</MultiAzimuthMixedModeCfg>
<MultiAzimuthWhiteLightCfg>
  <!--ro, opt, object, 多方位白光模式参数配置, desc: 当supplementLightMode == multiAzimuthWhiteLight生效-->
  <brightnessRegulatMode>
    <!--ro, opt, enum, 补光灯亮度调节模式, subType:string, [manual#手动,auto#自动], desc: 手动模式下用户需手动配置各组补光灯的亮度（LightCtrlList）；自动模式下用户无需配置亮度相关参数;由设备自动调节.-->auto
  </brightnessRegulatMode>
  <LightCtrlList>
    <!--ro, opt, array, 补光灯组控制参数, subType:object-->
    <LightCtrl>
      <!--ro, opt, object, 单个补光灯控制参数-->
      <azimuthType>
        <!--ro, opt, enum, 补光灯所在的方位类型, subType:string, [left#左侧方位补光灯,right#右侧方位补光灯]-->left
      </azimuthType>
      <lightType>
        <!--ro, opt, enum, 补光灯类型, subType:string, [highWhite#远光白光补光灯,midWhite#中光白光补光灯,lowWhite#近光白光补光灯,white#普通白光补光(不分远中近光)]-->highWhite
      </lightType>
      <brightness>
        <!--ro, opt, int, 补光灯亮度配置, range:[0,100]-->80
      </brightness>
    </LightCtrl>
  </LightCtrlList>
</MultiAzimuthWhiteLightCfg>
<MultiAzimuthIrLightCfg>
  <!--ro, opt, object, 多方位红外模式参数配置, desc: 当supplementLightMode == multiAzimuthIrLight生效-->
  <brightnessRegulatMode>
    <!--ro, opt, enum, 补光灯亮度调节模式, subType:string, [manual#手动,auto#自动], desc: 手动模式下用户需手动配置各组补光灯的亮度（LightCtrlList）；自动模式下用户无需配置亮度相关参数;由设备自动调节.-->auto
  </brightnessRegulatMode>
  <LightCtrlList>
    <!--ro, opt, array, 补光灯组控制参数, subType:object-->
    <LightCtrl>
      <!--ro, opt, object, 单个补光灯控制参数-->
      <azimuthType>
        <!--ro, opt, enum, 补光灯所在的方位类型, subType:string, [left#左侧方位补光灯,right#右侧方位补光灯]-->left
      </azimuthType>
      <lightType>
        <!--ro, opt, enum, 补光灯类型, subType:string, [highIrLight#远光红外补光灯,midIrLight#中光红外补光灯,lowIrLight#近光红外补光灯,irLight#普通红外补光灯(不分远中近光)]-->highIrLight
      </lightType>
      <brightness>
        <!--ro, opt, int, 补光灯亮度配置, range:[0,100]-->80
      </brightness>
    </LightCtrl>
  </LightCtrlList>
</MultiAzimuthIrLightCfg>
<PanoramicWhiteLightCfg>
  <!--ro, opt, object, 全景白光模式参数配置, desc: 当supplementLightMode == panoramicWhiteLight生效-->
  <brightnessRegulatMode>
    <!--ro, opt, enum, 补光灯亮度调节模式, subType:string, [manual#手动,auto#自动], desc: 手动模式下用户需手动配置各组补光灯的亮度（LightCtrlList）；自动模式下用户无需配置亮度相关参数;由设备自动调节.-->auto
  </brightnessRegulatMode>
  <LightCtrlList>
    <!--ro, opt, array, 补光灯组控制参数, subType:object-->
    <LightCtrl>
      <!--ro, opt, object, 单个补光灯控制参数-->
      <lightID>
        <!--ro, opt, int, 补光灯序号, range:[1,4]-->1
      </lightID>
      <brightness>
        <!--ro, opt, int, 补光灯亮度配置, range:[0,100]-->80
      </brightness>
    </LightCtrl>

```

```
</LightCtrlList>
</PanoramicWhiteLightCfg>
<PanoramicIrLightCfg>
  <!--ro, opt, object, 全景红外模式参数配置, desc:当supplementLightMode == panoramicIrLightCfg生效-->
  <brightnessRegulatMode>
    <!--ro, opt, enum, 补光灯亮度调节模式, subType:string, [manual#手动,auto#自动], desc:手动模式下用户需手动配置各组补光灯的亮度（LightCtrlList）；自动模式下用户无需配置亮度相关参数;由设备自动调节.-->auto
  </brightnessRegulatMode>
  <LightCtrlList>
    <!--ro, opt, array, 补光灯组控制参数, subType:object-->
    <LightCtrl>
      <!--ro, opt, object, 单个补光灯控制参数-->
      <lightID>
        <!--ro, opt, int, 补光灯序号, range:[1,4]-->1
      </lightID>
      <brightness>
        <!--ro, opt, int, 补光灯亮度配置, range:[0,100]-->80
      </brightness>
    </LightCtrl>
  </LightCtrlList>
</PanoramicIrLightCfg>
<whiteLightEnhanceEnabled>
  <!--ro, opt, bool, 白光灯亮度增强使能(不分远近光灯时控制), desc:开启增强使能后,白光灯会进入强电流模式,亮度会增强十倍,同时也可通过$.SupplementLight.whiteLightBrightness参数去控制强电流模式下的灯光亮度。该参数不区分补光灯模式及白光灯类型.-->true
</whiteLightEnhanceEnabled>
<LightManualCtrlList>
  <!--ro, opt, array, 补光灯组手动控制参数, subType:object, dep:and,{$.ImageChannel.SupplementLight.mode,eq,manual}, desc:与supplementLightMode无关-->
  <LightCtrl>
    <!--ro, opt, object, 单个补光灯控制参数-->
    <lightType>
      <!--ro, opt, enum, 补光灯类型, subType:string, [highIrLight#远光红外补光灯,midIrLight#中光红外补光灯,lowIrLight#近光红外补光灯,irLight#普通红外补光灯(不区分远中近光),highWhite#远光白光补光灯,midWhite#中光白光补光灯,lowWhite#近光白光补光灯,white#普通红外补光灯(不区分远中近光)]-->highIrLight
    </lightType>
    <brightness>
      <!--ro, opt, int, 补光灯亮度配置, range:[0,100]-->80
    </brightness>
  </LightCtrl>
</LightManualCtrlList>
</SupplementLight>
<OpticalDehaze>
  <!--ro, opt, object, 光学雾透-->
  <enabled>
    <!--ro, opt, bool, 使能-->true
  </enabled>
</OpticalDehaze>
<ManualRanging>
  <!--ro, opt, object, 手动测距-->
  <enabled>
    <!--ro, opt, bool, 使能-->true
  </enabled>
</ManualRanging>
<OIS>
  <!--ro, opt, object, 光学防抖-->
  <mode>
    <!--ro, req, enum, 模式, subType:string, [close#关闭,ordinary#普通,experts#专家]-->close
  </mode>
  <OISLevel>
    <!--ro, opt, enum, 光学防抖等级, subType:string, [high#高,middle#中,low#低]-->high
  </OISLevel>
  <OISSensitivity>
    <!--ro, opt, string, 光学防抖灵敏度-->test
  </OISSensitivity>
</OIS>
<isSupportLaserSpotManual>
  <!--ro, opt, bool, 是否支持手动激光点-->true
</isSupportLaserSpotManual>
<isSupportLaserSpotAdjustment>
  <!--ro, opt, bool, 是否支持激光点调整-->true
</isSupportLaserSpotAdjustment>
<DigitalZoom>
  <!--ro, opt, object, 数字缩放-->
  <ZoomRatio>
    <!--ro, opt, enum, 缩放比例, subType:string, [1x#1x,2x#2x,4x#4x,8x#8x,16x#16x,32x#32x]-->1x
  </ZoomRatio>
</DigitalZoom>
<Palettes>
  <!--ro, opt, object, 热成像调色板配置-->
  <mode>
    <!--ro, opt, enum, 模式, subType:string, [WhiteHot#白热,BlackHot#黑热,Fusion1#融合1,Rainbow#彩虹,Fusion2#融合2,Ironbow1#铁红1,Ironbow2#铁红2,Sepia#深褐色,Color1#色彩1,Color2#色彩2,IceFire#冰火,Rain#雨,RedHot#红热,GreenHot#绿热,DeepBlue#深蓝,Color3#色彩3]-->WhiteHot
  </mode>
  <ColorateTarget>
    <!--ro, opt, object, 目标凸显, desc:dep mode = "WhiteHot"-->
    <ColorateTargetModelList>
      <!--ro, opt, array, 目标凸显模式列表, subType:object-->
      <ColorateTargetMode>
        <!--ro, opt, object, 目标凸显模式-->
        <id>
          <!--ro, req, int, 序号, desc:从1开始-->0
        </id>
        <mode>
          <!--ro, req, enum, 模式, subType:string, [colorateHotArea#大于（才凸显）,colorateIntervalArea#区间（才凸显）,colorateColdArea#小于（才凸显）]-->colorateHotArea
        </mode>
      </ColorateTargetMode>
    </ColorateTargetModelList>
  </ColorateTarget>
</Palettes>
```

```

</mode>
<enabled>
  <!--ro, req, bool, 使能-->true
</enabled>
<TemperatureLimit>
  <!--ro, req, object, 边界温度-->
  <minTemperature>
    <!--ro, opt, float, 最低温度, range: [-20,150]-->0.000
  </minTemperature>
  <maxTemperature>
    <!--ro, opt, float, 最高温度, range: [-20,150]-->0.000
  </maxTemperature>
</TemperatureLimit>
<Color>
  <!--ro, req, object, 区域颜色-->
  <R>
    <!--ro, req, int, R, range: [0,255]-->0
  </R>
  <G>
    <!--ro, req, int, G, range: [0,255]-->0
  </G>
  <B>
    <!--ro, req, int, B, range: [0,255]-->0
  </B>
</Color>
</ColorateTargetMode>
</ColorateTargetModelList>
</ColorateTarget>
</Palettes>
<ExposureSync>
  <!--ro, opt, object, 曝光同步-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
</ExposureSync>
<BrightnessSuddenChangeSuppression>
  <!--ro, opt, object, 亮度突变抑制-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
</BrightnessSuddenChangeSuppression>
<AGCCap>
  <!--ro, opt, object, AGC能力-->
  <AGCMode>
    <!--ro, opt, enum, AGC模式, subType:string, [self_Adaptive#自适应,histogram#直方图,linear#线性]-->self_Adaptive
  </AGCMode>
</AGCCap>
<RegionalImageEnhancementCap>
  <!--ro, opt, object, 区域图像增强能力-->
  <mode>
    <!--ro, req, enum, 模式, subType:string,
[off#off,customArea#customArea,up_60%#up_60%,down_60%#down_60%,Left_60%#Left_60%,right_60%#right_60%,center_50%#center_50%,center_75%#center_75%]-->off
  </mode>
  <CustomAreaParam>
    <!--ro, opt, object, 自定义区域参数, desc:dep <mode>="customArea"-->
  <RegionCoordinatesList>
    <!--ro, opt, array, 区域坐标列表, subType:object, range:[3,10]-->
  <RegionCoordinates>
    <!--ro, opt, object, 区域坐标, desc:屏幕左上角为坐标原点-->
    <positionX>
      <!--ro, req, int, x坐标, range:[0,1000]-->0
    </positionX>
    <positionY>
      <!--ro, req, int, y坐标, range:[0,1000]-->0
    </positionY>
  </RegionCoordinates>
</RegionCoordinatesList>
</CustomAreaParam>
<DefaultAreaParamList>
  <!--ro, opt, array, 默认区域参数列表, subType:object, desc:dep <mode>="up_60%,down_60%,Left_60%,right_60%,center_50%,center_75%"-->
  <DefaultAreaParam>
    <!--ro, opt, object, 默认区域参数-->
  <mode>
    <!--ro, req, enum, 模式, subType:string,
[up_60%#up_60%,down_60%#down_60%,Left_60%#Left_60%,right_60%#right_60%,center_50%#center_50%,center_75%#center_75%]-->up_60%
  </mode>
  <RegionCoordinatesList>
    <!--ro, opt, array, 区域坐标列表, subType:object, range:[3,10]-->
  <RegionCoordinates>
    <!--ro, opt, object, 区域坐标, desc:屏幕左上角为坐标原点-->
    <positionX>
      <!--ro, req, int, x坐标, range:[0,1000]-->0
    </positionX>
    <positionY>
      <!--ro, req, int, y坐标, range:[0,1000]-->0
    </positionY>
  </RegionCoordinates>
</RegionCoordinatesList>
</DefaultAreaParam>
</DefaultAreaParamList>
</RegionalImageEnhancementCap>
<faceExposure>
  <!--ro, opt, object, 人脸曝光-->
  <faceExposureEnabled>

```

```
</faceExposureSensitivity>
<!--ro, req, bool, 人脸曝光使能-->true
</faceExposureEnabled>
<faceExposureSensitivity>
<!--ro, req, int, 人脸曝光灵敏度, range:[0,100]-->0
</faceExposureSensitivity>
</faceExposure>
<LightHal>
<!--ro, opt, object, 信号灯光晕抑制-->
<enabled>
<!--ro, req, bool, 使能-->true
</enabled>
<halLevel>
<!--ro, req, int, 光晕抑制等级, range:[0,100]-->0
</halLevel>
<halRange>
<!--ro, req, int, 光晕抑制范围, range:[0,2064]-->0
</halRange>
</LightHal>
<TargetEnhancement>
<!--ro, opt, object, 图像目标增强配置-->
<enabled>
<!--ro, req, bool, 使能-->true
</enabled>
</TargetEnhancement>
<LaserLight>
<!--ro, opt, object, 激光补光-->
<mode>
<!--ro, opt, enum, 模式, subType:string, [auto#自动,manual#手动]-->auto
</mode>
<compensationRegionMode>
<!--ro, opt, enum, 激光补光区域, subType:string, [center#中心,adaptive#适应,full_Screen#全屏]-->center
</compensationRegionMode>
<compensationLevel>
<!--ro, opt, enum, 激光补光强度, subType:string, [Low#低,medium#中,high#高]-->low
</compensationLevel>
<LaserColor>
<!--ro, opt, object, 激光颜色, desc:依赖<mode>为manual-->
<R>
<!--ro, req, int, R(红色)值, range:[0,100]-->0
</R>
<G>
<!--ro, req, int, G(绿色)值, range:[0,100]-->0
</G>
<B>
<!--ro, req, int, B(蓝色)值, range:[0,100]-->0
</B>
</LaserColor>
</LaserLight>
<ExternalSupplementLight>
<!--ro, opt, object, 外接补光灯参数配置-->
<mode>
<!--ro, opt, enum, 模式, subType:string, [close#关闭,oneMixedLight#一组混合补光]-->close
</mode>
<BrightnessList>
<!--ro, opt, array, 亮度列表, subType:object-->
<Brightness>
<!--ro, opt, object, 亮度-->
<id>
<!--ro, opt, int, 每组补光灯的id-->0
</id>
<irLightBrightness>
<!--ro, opt, int, 红外灯亮度, range:[0,100]-->0
</irLightBrightness>
<whiteLightBrightness>
<!--ro, opt, int, 白光灯亮度, range:[0,100]-->0
</whiteLightBrightness>
</Brightness>
</BrightnessList>
</ExternalSupplementLight>
<TempRange>
<!--ro, req, object, 温度范围-->
<mode>
<!--ro, req, enum, 温度范围模式, subType:string, [automatic#自动模式,manual#手动模式]-->automatic
</mode>
<temperatureUpperLimit>
<!--ro, opt, float, 上限温度, range:[-20,550], unit:℃, unitType:温度, desc:精确到小数点后一位,手动温度范围模式下有效,上限温度需大于下限温度-->0.000
</temperatureUpperLimit>
<temperatureLowerLimit>
<!--ro, opt, float, 下限温度, range:[-20,550], unit:℃, unitType:温度, desc:精确到小数点后一位,手动温度范围模式下有效,下限温度需小于上限温度-->0.000
</temperatureLowerLimit>
</TempRange>
<Beauty>
<!--ro, opt, object, 美颜能力-->
<enabled>
<!--ro, req, bool, 使能-->true
</enabled>
<whiteningStrength>
<!--ro, opt, int, 美白强度, range:[0,10]-->1
</whiteningStrength>
<skinSmoothingStrength>
<!--ro, opt, int, 磨皮强度, range:[0,10]-->1
</skinSmoothingStrength>
</Beauty>
```

```
<MEposure>
  <!--ro, opt, object, 手动曝光（工厂标定）-->
  <enabled>
    <!--ro, opt, bool, 使能-->true
  </enabled>
  <exposureLevel>
    <!--ro, opt, string, 曝光等级, range:[0,1000000]-->test
  </exposureLevel>
  <gainLevel>
    <!--ro, opt, int, 增益等级, range:[0,100]-->0
  </gainLevel>
</MEposure>
<SensorRGBCtrl>
  <!--ro, opt, object, SensorRGB控制-->
  <sensorID>
    <!--ro, opt, int, SensorID, range:[0,1]-->0
  </sensorID>
</SensorRGBCtrl>
<YPbPrCalibration>
  <!--ro, opt, object, 色差标定-->
  <mode>
    <!--ro, opt, int, 模式, range:[0,2]-->1
  </mode>
</YPbPrCalibration>
<intrusionTriggeredLightingCfg>
  <!--ro, opt, object, 随影补光-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
</intrusionTriggeredLightingCfg>
<shotProCfg>
  <!--ro, opt, object, 幕影-->
  <shotProMode>
    <!--ro, req, enum, 幕影模式, subType:string, [cLose#关闭,general#通用,enhancement#增强]-->general
  </shotProMode>
</shotProCfg>
</ImageChannel>
```

69.24.10 获取所有通道图像处理参数

Request URL

GET /ISAPI/Image/channels

查询参数

无

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>
<ImageChannelList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, array, 图像信号处理参数列表, subType:object, attr:version(req, string, 协议版本)-->
  <ImageChannel>
    <!--ro, req, object, 图像信号处理参数-->
    <id>
      <!--ro, req, int, 序号ID-->1
    </id>
    <enabled>
      <!--ro, req, bool, 使能-->true
    </enabled>
    <videoInputID>
      <!--ro, req, int, 视频通道号ID-->1
    </videoInputID>
    <FocusConfiguration>
      <!--ro, opt, object, 普通聚焦-->
      <focusStyle>
        <!--ro, req, enum, 聚焦类型, subType:string, [AUTO#自动,MANUAL#手动,SEMIAUTOMATIC#半自动]-->AUTO
      </focusStyle>
      <focusLimited>
        <!--ro, opt, enum, 聚焦限制, subType:int,
[10#10,50#50,100#100,150#150,300#300,600#600,1000#1000,2000#2000,5000#5000,20000#20000,50000#50000,65535#65535]-->10
      </focusLimited>
      <focusPosition>
        <!--ro, opt, int, 焦点位置, range:[1,10]-->1
      </focusPosition>
      <focusSpeed>
        <!--ro, opt, int, 聚焦速度, range:[1,3]-->1
      </focusSpeed>
      <focusSensitivity>
        <!--ro, opt, int, 聚焦灵敏度, range:[0,2]-->1
      </focusSensitivity>
      <temperatureChangeAdaptEnabled>
        <!--ro, opt, bool, 温度变化适应使能-->true
      </temperatureChangeAdaptEnabled>
    </FocusConfiguration>
  </ImageChannel>
</ImageChannelList>
```

```
<!--ro, opt, bool, 温度变化自适应-->true
</temperatureChangeAdaptEnabled>
<relativeFocusPos>
  <!--ro, opt, int, 相对焦点位置, range:[0,4294967295]-->1
</relativeFocusPos>
<lensType>
  <!--ro, opt, enum, 聚焦镜头类型, subType:string, [1140#1140,1050#1050]-->1140
</lensType>
<highTemperaturePriority>
  <!--ro, opt, bool, 高温优先模式使能-->true
</highTemperaturePriority>
<focusRange>
  <!--ro, opt, enum, 聚焦调节范围, subType:string, [min#最小,medium#适中,max#最大]-->max
</focusRange>
<focusDistanceMode>
  <!--ro, opt, enum, 聚焦距离模式, subType:string, [compatible#兼容模式], desc:compatible-兼容模式, 在球罩设备上调整聚焦限制距离focusLimited,仍然虚焦
的情况, 选择兼容模式, focusLimited参数失效, 设备释放镜头最大聚焦能力, 使镜头获得更大的焦距范围, 解决球罩设备近物距聚焦效果差的问题。开启该功能, 聚焦速度会变
慢, 非球罩设备不建议使用。-->compatible
</focusDistanceMode>
</FocusConfiguration>
<LensInitialization>
  <!--ro, opt, object, 镜头初始化-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
</LensInitialization>
<ImageFlip>
  <!--ro, opt, object, 图像翻转-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
<ImageFlipStyle>
  <!--ro, opt, enum, 镜像类型, subType:string, [LEFTRIGHT#LEFTRIGHT,UPDOWN#UPDOWN,CENTER#CENTER,AUTO#AUTO]-->LEFTRIGHT
</ImageFlipStyle>
<flipAngle>
  <!--ro, opt, enum, 旋转角度, subType:int, [90#90,180#180,270#270]-->90
</flipAngle>
</ImageFlip>
<ImageFreeze>
  <!--ro, opt, object, 图像冻结-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
</ImageFreeze>
<proportionalpan>
  <!--ro, opt, object, 比例盘-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
</proportionalpan>
<WDR>
  <!--ro, opt, object, 宽动态(自动)能力-->
  <mode>
    <!--ro, req, enum, open, subType:string, [open#开启,close#关闭,auto#自动,SDR#SDR]-->open
  </mode>
  <WDRLevel>
    <!--ro, opt, int, 宽动态等级, range:[1,100]-->1
  </WDRLevel>
  <WDRContrastLevel>
    <!--ro, opt, int, 宽动态对比度, range:[1,100]-->1
  </WDRContrastLevel>
  <WDRLevel1>
    <!--ro, opt, int, 第二类宽动态等级, range:[1,100]-->1
  </WDRLevel1>
</WDR>
<BLC>
  <!--ro, opt, object, 背光补偿区域-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <BLCMode>
    <!--ro, opt, enum, 背光补偿模式, subType:string, [UP#UP,DOWN#DOWN,LEFT#LEFT,RIGHT#RIGHT,CENTER#CENTER,MULTI-AREA#MULTI-
AREA,Region#Region,AUTO#AUTO]-->UP
  </BLCMode>
  <BLCLevel>
    <!--ro, opt, int, 背光补偿等级, range:[0,10]-->0
  </BLCLevel>
  <BLCRegionList>
    <!--ro, opt, array, 背光补偿区域列表, subType:object-->
    <BLCRegion>
      <!--ro, opt, object, 背光补偿区域-->
      <id>
        <!--ro, req, int, 区域编号, range:[1,10]-->1
      </id>
      <RegionCoordinatesList>
        <!--ro, req, array, 坐标点列表, subType:object, range:[0,4], desc:矩形-->
        <RegionCoordinates>
          <!--ro, opt, object, 坐标点, desc:屏幕左下角为坐标原点-->
          <positionX>
            <!--ro, req, int, x坐标, range:[0,704]-->1
          </positionX>
          <positionY>
            <!--ro, req, int, y坐标, range:[0,576]-->1
          </positionY>
```

```
</RegionCoordinates>
</RegionCoordinatesList>
</BLCRegion>
</BLCRegionList>
</BLC>
<NoiseReduce>
  <!--ro, opt, object, 降噪参数-->
  <mode>
    <!--ro, req, enum, 模式, subType:string, [close#关闭,general#常规,advanced#高级]-->close
  </mode>
  <GeneralMode>
    <!--ro, opt, object, 模式-->
    <generalLevel>
      <!--ro, req, int, 等级, range:[0,100]-->1
    </generalLevel>
  </GeneralMode>
  <AdvancedMode>
    <!--ro, opt, object, 高级模式-->
    <FrameNoiseReduceLevel>
      <!--ro, req, int, 框架噪声降阶, range:[0,100]-->1
    </FrameNoiseReduceLevel>
    <InterFrameNoiseReduceLevel>
      <!--ro, req, int, 帧间噪声降维, range:[0,100]-->1
    </InterFrameNoiseReduceLevel>
    <secondarySpatialNoiseReduceLevel>
      <!--ro, opt, int, 二级降噪空域等级, range:[0,100]-->1
    </secondarySpatialNoiseReduceLevel>
    <secondaryTemporalNoiseReduceLevel>
      <!--ro, opt, int, 二级降噪时域等级, range:[0,100]-->1
    </secondaryTemporalNoiseReduceLevel>
  </AdvancedMode>
</NoiseReduce>
<ImageEnhancement>
  <!--ro, opt, object, 图像增强-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <ImageEnhancementLevel>
    <!--ro, opt, int, 图像增强等级, range:[0,10]-->0
  </ImageEnhancementLevel>
</ImageEnhancement>
<DSS>
  <!--ro, opt, object, DSS-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <DSSLevel>
    <!--ro, opt, enum, 低照度等级, subType:string,
[*1.25#1.25,*1.5#1.5,*2#2,*3#3,*4#4,*5#5,*6#6,*8#8,*12#12,*16#16,*24#24,*32#32,*48#48,*64#64,*96#96,*128#128,*256#256,*512#512,auto#auto]-
->*1.25
  </DSSLevel>
  <mode>
    <!--ro, opt, enum, 模式, subType:string, [frame#帧模式,Level#等级模式], desc:无mode节点默认为frame模式-->frame
  </mode>
  <slowShutterLevel>
    <!--ro, opt, int, 慢快门等级, range:[1,6]-->1
  </slowShutterLevel>
</DSS>
<WhiteBalance>
  <!--ro, opt, object, 白平衡-->
  <WhiteBalanceStyle>
    <!--ro, opt, enum, 白平衡类型, subType:string, [auto#自动,auto1#自动1,auto2#自动2>manual#手动,indoor#室内,outdoor#室外,autotrace#自动跟踪,once#单
次,sodiumLight#苏打光,mercuryLight#水银灯,customColorTemperature#自定义色温]-->auto
  </WhiteBalanceStyle>
  <WhiteBalanceRed>
    <!--ro, opt, int, 白平衡红, range:[0,100]-->1
  </WhiteBalanceRed>
  <WhiteBalanceBlue>
    <!--ro, opt, int, 白平衡蓝, range:[0,100]-->1
  </WhiteBalanceBlue>
  <WhiteBalanceLevel>
    <!--ro, opt, int, 白平衡等级, range:[0,100], desc:<WhiteBalanceStyle>为auto,auto1,auto2时有效-->1
  </WhiteBalanceLevel>
  <customColorTemperatureLevel>
    <!--ro, opt, int, 自定义色温等级, range:[2700,7500]-->1
  </customColorTemperatureLevel>
</WhiteBalance>
<Exposure>
  <!--ro, opt, object, 曝光模式-->
  <ExposureType>
    <!--ro, req, enum, 镜头类型, subType:string,
[auto#auto,IrisFirst#IrisFirst,ShutterFirst#ShutterFirst,gainFirst#gainFirst>manual#manual,pIris#pIris,T5280-PQ1#T5280-PQ1,T5289-PQ1#T5289-PQ1,T1140-
PQ1#T1140-PQ1,T2712-PQ1#T2712-PQ1,HV1250P-MPIR#HV1250P-MPIR,pc#pc,pIris-General#pIris-General,HZ1140P-8MPIRA#HZ1140P-8MPIRA,HZ1135P-8MPSGA#HZ1135P-
8MPSGA,MV1550P-12MPIR#MV1550P-12MPIR,MZ1545P-12MP#MZ1545P-12MP,HZ0618P-4MPSG#HZ0618P-4MPSG,MZ1050P-12MP#MZ1050P-12MP]-->auto
  </ExposureType>
  <autoIrisLevel>
    <!--ro, opt, int, 光圈自动级别, range:[0,100]-->1
  </autoIrisLevel>
  <OverexposeSuppress>
    <!--ro, opt, object, 过度曝光抑制-->
    <enabled>
      <!--ro, req, bool, 使能-->true
    </enabled>
```

```

<type>
  <!--ro, opt, enum, 过度曝光抑制类型, subType:string, [AUTO#自动,MANUAL#手动]-->AUTO
</Type>
<DistanceLevel>
  <!--ro, opt, int, 焦距, range:[1,100]-->1
</DistanceLevel>
<shortIRDistanceLevel>
  <!--ro, opt, int, IR短距离级别, range:[1,100]-->1
</shortIRDistanceLevel>
<longIRDistanceLevel>
  <!--ro, opt, int, IR长距离级别, range:[1,100]-->1
</longIRDistanceLevel>
<supplementLightIntensity1>
  <!--ro, opt, int, 补充曝光强度1, range:[1,100]-->1
</supplementLightIntensity1>
<supplementLightIntensity2>
  <!--ro, opt, int, 补充曝光强度2, range:[1,100]-->1
</supplementLightIntensity2>
<supplementLightIntensity3>
  <!--ro, opt, int, 补充曝光强度3, range:[1,100]-->1
</supplementLightIntensity3>
<supplementLightIntensity4>
  <!--ro, opt, int, 补充曝光强度4, range:[1,100]-->1
</supplementLightIntensity4>
</OverexposeSuppress>
<pIris>
  <!--ro, opt, object, 光圈-->
  <pIrisType>
    <!--ro, opt, enum, 光圈类型, subType:string, [AUTO#AUTO,MANUAL#MANUAL]-->MANUAL
  </pIrisType>
  <IrisLevel>
    <!--ro, opt, int, 光圈级别, range:[1,100]-->1
  </IrisLevel>
</pIris>
<faceExposure>
  <!--ro, opt, object, 人脸曝光-->
  <enabled>
    <!--ro, opt, bool, 使能-->true
  </enabled>
  <sensitivity>
    <!--ro, opt, int, 灵敏度, range:[0,100]-->0
  </sensitivity>
</faceExposure>
<PIrisGeneral>
  <!--ro, opt, object, pIris-General配置-->
  <irisLevel>
    <!--ro, opt, int, 通用镜头下的光圈等级, range:[1,100]-->1
  </irisLevel>
</PIrisGeneral>
</Exposure>
<Sharpness>
  <!--ro, opt, object, 锐度-->
  <SharpnessMode>
    <!--ro, opt, enum, 锐度模式, subType:string, [manual#手动,auto#自动]-->manual
  </SharpnessMode>
  <SharpnessLevel>
    <!--ro, req, int, 锐度级别, range:[0,100]-->1
  </SharpnessLevel>
</Sharpness>
<gammaCorrection>
  <!--ro, opt, object, 伽玛校正-->
  <gammaCorrectionEnabled>
    <!--ro, opt, bool, 伽玛校正使能-->true
  </gammaCorrectionEnabled>
  <gammaCorrectionLevel>
    <!--ro, opt, int, 伽玛校正级别, range:[0,100]-->1
  </gammaCorrectionLevel>
</gammaCorrection>
<Shutter>
  <!--ro, opt, object, 快门配置-->
  <ShutterLevel>
    <!--ro, opt, enum, 曝光等级, subType:string,
[1/1#1/1,1/2#1/2,1/3#1/3,1/6#1/6,1/12#1/12,1/25#1/25,1/50#1/50,1/75#1/75,1/100#1/100,1/120#1/120,1/125#1/125,1/150#1/150,1/175#1/175], desc:depends on
<ExposureType>-->1/1
  </ShutterLevel>
  <maxShutterLevelLimit>
    <!--ro, opt, enum, 最大曝光等级, subType:string,
[1/1#1/1,1/2#1/2,1/3#1/3,1/6#1/6,1/12#1/12,1/25#1/25,1/50#1/50,1/75#1/75,1/100#1/100,1/120#1/120,1/125#1/125,1/150#1/150,1/175#1/175]-->1/1
  </maxShutterLevelLimit>
  <minShutterLevelLimit>
    <!--ro, opt, enum, 最小曝光等级, subType:string,
[1/1#1/1,1/2#1/2,1/3#1/3,1/6#1/6,1/12#1/12,1/25#1/25,1/50#1/50,1/75#1/75,1/100#1/100,1/120#1/120,1/125#1/125,1/150#1/150,1/175#1/175]-->1/1
  </minShutterLevelLimit>
</Shutter>
<powerLineFrequency>
  <!--ro, opt, object, 视频制式-->
  <powerLineFrequencyMode>
    <!--ro, opt, enum, 视频制式模式, subType:string, [50hz#50hz,60hz#60hz]-->50hz
  </powerLineFrequencyMode>
</powerLineFrequency>
<Gain>
  <!--ro, opt, object, 增益-->
  <GainLevel>
    <!--ro, opt, string, 增益等级, range:[0,10]-->test
  </GainLevel>

```

```
</GainLevel>
<GainLimit>
  <!--ro, opt, enum, 增益, subType:string,
[0#0,400#400,900#900,1300#1300,1800#1800,2200#2200,2600#2600,3100#3100,3500#3500,4000#4000,4400#4400,4800#4800,5300#5300,5700#5700,6200#6200,6600#6600]-->0
</GainLimit>
</Gain>
<Color>
  <!--ro, opt, object, 颜色-->
  <brightnessLevel>
    <!--ro, opt, int, 亮度, range:[0,100]-->0
  </brightnessLevel>
  <contrastLevel>
    <!--ro, opt, int, 对比度, range:[0,100]-->0
  </contrastLevel>
  <saturationLevel>
    <!--ro, opt, int, 饱和度, range:[0,100]-->0
  </saturationLevel>
  <hueLevel>
    <!--ro, opt, int, 色调, range:[0,100]-->0
  </hueLevel>
  <grayScale>
    <!--ro, opt, object, 灰度-->
    <grayScaleMode>
      <!--ro, opt, enum, 灰度模式, subType:string, [indoor#室内,outdoor#室外]-->indoor
    </grayScaleMode>
  </grayScale>
  <nightMode>
    <!--ro, opt, bool, 夜间黑白模式使能-->true
  </nightMode>
  <sharpnessLevel>
    <!--ro, opt, int, 锐度, range:[0,100]-->0
  </sharpnessLevel>
</Color>
<IrcutFilter>
  <!--ro, opt, object, IrcutFilter配置-->
  <IrcutFilterType>
    <!--ro, opt, enum, IrcutFilter类型, subType:string,
[auto#auto,day#day,night#night,schedule#schedule,eventTrigger#eventTrigger,videoAuto#videoAuto,darkFighterX#darkFighterX,darkFighterXAuto#darkFighterXAuto,d
arkFighterXSchedule#darkFighterXSchedule]-->auto
  </IrcutFilterType>
  <dayToNightFilterLevel>
    <!--ro, opt, enum, 日夜切换级别, subType:string, [Low#低,normal#中,high#高]-->low
  </dayToNightFilterLevel>
  <dayToNightFilterTime>
    <!--ro, opt, int, 日夜切换时间, range:[0,10]-->1
  </dayToNightFilterTime>
  <nightToDayFilterLevel>
    <!--ro, opt, enum, 夜日切换级别, subType:string, [Low#低,normal#中,high#高]-->low
  </nightToDayFilterLevel>
  <nightToDayFilterTime>
    <!--ro, opt, int, 夜日切换时间, range:[0,10]-->1
  </nightToDayFilterTime>
  <Schedule>
    <!--ro, opt, object, 定时-->
    <scheduleType>
      <!--ro, req, enum, 定时类型, subType:string, [day#白天,night#晚上]-->day
    </scheduleType>
    <TimeRange>
      <!--ro, req, object, 时间段-->
      <beginTime>
        <!--ro, req, time, 开始时间-->00:00:00+08:00
      </beginTime>
      <endTime>
        <!--ro, req, time, 结束时间-->00:00:00+08:00
      </endTime>
    </TimeRange>
  </Schedule>
  <EventTrigger>
    <!--ro, opt, object, 事件触发-->
    <eventType>
      <!--ro, req, enum, 事件类型, subType:string, [IO#IO事件,VMD#VMD事件]-->IO
    </eventType>
    <IrcutFilterAction>
      <!--ro, req, enum, IrcutFilter操作, subType:string, [day#白天,night#晚上]-->day
    </IrcutFilterAction>
  </EventTrigger>
</IrcutFilter>
<Scene>
  <!--ro, opt, object, 场景-->
  <mode>
    <!--ro, req, enum, 模式, subType:string, [indoor#户内,outdoor#户外,standard#标准,dimLight#昏暗]-->indoor
  </mode>
</Scene>
<PTZ>
  <!--ro, opt, object, PTZ-->
  <enabled>
    <!--ro, opt, bool, 使能-->true
  </enabled>
</PTZ>
<EPTZ>
  <!--ro, opt, object, EPTZ-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
</EPTZ>
```

```
</enabled>
<mode>
  <!--ro, opt, enum, 模式, subType:string, [cruise#巡航（云台可以支持预置点、巡航功能、ZOOM和8个方向）,autoTrack#autoTrack-自动跟踪]-->cruise
</mode>
</EPTZ>
<EIS>
  <!--ro, opt, object, 电子防抖-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <EISLevel>
    <!--ro, opt, int, EIS等级, range:[0,10]-->0
  </EISLevel>
</EIS>
<HLC>
  <!--ro, opt, object, 强光抑制-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <HLCLevel>
    <!--ro, opt, int, HLC等级, range:[0,10]-->0
  </HLCLevel>
</HLC>
<ZoomLimit>
  <!--ro, opt, object, 缩放限制-->
  <ZoomLimitRatio>
    <!--ro, opt, enum, 缩放比例, subType:int, [4#4,8#8,16#16,32#32,64#64]-->4
  </ZoomLimitRatio>
</ZoomLimit>
<corridor>
  <!--ro, opt, object, 镜头走廊模式-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
</corridor>
<Dehaze>
  <!--ro, opt, object, 雾透-->
  <DehazeMode>
    <!--ro, opt, enum, 雾透模式, subType:string, [open#打开,close#关闭,auto#自动]-->open
  </DehazeMode>
  <DehazeLevel>
    <!--ro, opt, int, 雾透等级, range:[0,10]-->0
  </DehazeLevel>
</Dehaze>
<ImageMode>
  <!--ro, opt, enum, 图像模式, subType:string, [standard#标准,indoor#室内,outdoor#室外,dimLight#昏暗]-->standard
</ImageMode>
<enableImageLossDetection>
  <!--ro, opt, bool, 图像丢失检测使能-->true
</enableImageLossDetection>
<CaptureMode>
  <!--ro, opt, object, 抓拍模式-->
  <mode>
    <!--ro, req, enum, 模式, subType:string, [close#关闭,1920*1080@30fps#1920*1080@30fps]-->close
  </mode>
</CaptureMode>
<IrLight>
  <!--ro, opt, object, 亮度限制-->
  <mode>
    <!--ro, req, enum, 模式, subType:string, [auto#自动>manual#手动]-->auto
  </mode>
  <brightnessLevel>
    <!--ro, opt, int, 亮度等级, range:[0,100]-->1
  </brightnessLevel>
  <sensitivityLevel>
    <!--ro, opt, int, 灵敏度, range:[0,100]-->1
  </sensitivityLevel>
  <brightnessLimit>
    <!--ro, opt, int, 亮度限制, range:[0,100]-->1
  </brightnessLimit>
  <triggerMode>
    <!--ro, opt, enum, 触发模式, subType:string, [camera#相机,photosensitive#光敏]-->camera
  </triggerMode>
</IrLight>
<LensDistortionCorrection>
  <!--ro, opt, object, 镜头畸变校正-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <level>
    <!--ro, opt, enum, 等级, subType:string, [level1#level1,level2#level2,level3#level3,custom#自定义]-->level1
  </level>
  <accurateLevel>
    <!--ro, opt, int, 畸变校正强度, range:[0,100]-->0
  </accurateLevel>
  <zoomedInDistantViewLevel>
    <!--ro, opt, int, 远端放大, range:[0,100], desc:畸变校正开启时生效-->0
  </zoomedInDistantViewLevel>
  <horizontalFOV>
    <!--ro, opt, int, 水平视场角, range:[0,100]-->0
  </horizontalFOV>
  <verticalFOV>
    <!--ro, opt, int, 垂直视场角, range:[0,100]-->0
  </verticalFOV>
```

```

</LensDistortionCorrection>
<SupplementLight>
  <!--ro, opt, object, 内置补光灯管理参数能力-->
  <mode>
    <!--ro, opt, enum, 补光灯调节模式, subType:string, [schedule#白光定时,off#白光关闭,on#白光开启,auto#自动>manual#手动]-->auto
  </mode>
  <Schedule>
    <!--ro, opt, object, 白光灯定时生效时间配置;当<Mode>= schedule时生效;所配置时间为白天的时间-->
    <TimeRange>
      <!--ro, req, object, 白天时段;当天剩余时段为夜晚时段-->
      <beginTime>
        <!--ro, req, time, 开始时间-->00:08:00+08:00
      </beginTime>
      <endTime>
        <!--ro, req, time, 结束时间-->00:18:00+08:00
      </endTime>
    </TimeRange>
  </Schedule>
  <brightnessLimit>
    <!--ro, opt, int, 白光补光灯亮度限制, range:[0,100], desc:当<Mode>= on时;可以调节该限制;为白光补光灯亮度最大值;实际补光灯亮度为自动调节（注：mode里面的自动为补光灯开关自动;自动调节模式需查看mixedLightBrightnessRegulatMode节点）-->80
  </brightnessLimit>
  <supplementLightMode>
    <!--ro, opt, enum, 补光灯模式, subType:string, [mixed#混合模式,whiteLight#白光模式,close#关闭,colorVuWhiteLight#全彩白光,irLight#红外模式,dualLight#双补光,eventIntelligence#事件智能补光,multiAzimuthMixed#多方位混合补光,multiAzimuthWhiteLight#多方位白光模式,multiAzimuthIrLight#多方位红外模式,panoramicWhiteLight#全景白光,panoramicIrLight#全景红外],
    desc:eventIntelligence为事件智能补光;解释如下:1.在满足补光灯功能生效条件下;当没有产生报警;设备仅红外灯开启;2.产生报警后自动关闭红外灯;开启白光灯持续1min;在持续时间内如继续收到报警;这一状态保持至最后一个报警发生后1min结束。multiAzimuthMixed多方位混合模式可以通过左右边各一组补光灯（一组补光灯包含远中近白光红外补光灯）对目标进行多方位角度的补光;不同于之前mixed混合补光模式仅支持一组补光灯单方位进行补光。multiAzimuthWhiteLight多方位白光模式、multiAzimuthIrLight多方位红外模式与多方位混光情况相同。
    panoramicWhiteLight、panoramicIrLight全景白光、红外模式当前仅四目全景相机支持,当前下相机无方位的概念,由四个补光灯围绕相机部署。-->mixed
  </supplementLightMode>
  <irLightBrightness>
    <!--ro, opt, int, 红外补光灯亮度, range:[0,100], desc:当<supplementLightMode>= mixed;且<mixedLightBrightnessRegulatMode>= manual时生效-->80
  </irLightBrightness>
  <mixedLightBrightnessRegulatMode>
    <!--ro, opt, enum, 混合补光灯亮度调节模式, subType:string, [manual#手动,auto#自动], desc:该字段仅在<supplementLightMode>= mixed时生效;手动模式下用户需手动<irLightBrightness>红外补光灯亮度参数以及<brightnessLimit>白光补光灯亮度;自动模式下用户无需配置亮度相关参数;由设备自动调节。-->auto
  </mixedLightBrightnessRegulatMode>
  <highIrLightBrightness>
    <!--ro, opt, int, 远光红外光亮度配置, range:[0,100], desc:参数由mixed#混合模式、whiteLight#白光模式、colorVuWhiteLight#全彩白光,irLight#红外模式、dualLight#双补光模式公用-->80
  </highIrLightBrightness>
  <highWhiteLightBrightness>
    <!--ro, opt, int, 远光白光亮度配置, range:[0,100], desc:参数由mixed#混合模式、whiteLight#白光模式、colorVuWhiteLight#全彩白光,irLight#红外模式、dualLight#双补光模式公用-->80
  </highWhiteLightBrightness>
  <lowIrLightBrightness>
    <!--ro, opt, int, 近光红外光亮度配置, range:[0,100], desc:参数由mixed#混合模式、whiteLight#白光模式、colorVuWhiteLight#全彩白光,irLight#红外模式、dualLight#双补光模式公用-->80
  </lowIrLightBrightness>
  <lowWhiteLightBrightness>
    <!--ro, opt, int, 近光白光亮度配置, range:[0,100], desc:参数由mixed#混合模式、whiteLight#白光模式、colorVuWhiteLight#全彩白光,irLight#红外模式、dualLight#双补光模式公用-->80
  </lowWhiteLightBrightness>
  <whiteLightBrightness>
    <!--ro, opt, int, 光亮度配置(不分远近光灯时控制), range:[0,100], desc:参数由mixed#混合模式、whiteLight#白光模式、colorVuWhiteLight#全彩白光,irLight#红外模式、dualLight#双补光模式、mode=auto 公用-->80
  </whiteLightBrightness>
  <whiteLightModeCfg>
    <!--ro, opt, object, 白光模式参数配置, desc:当supplementLightMode == whiteLight生效-->
    <highWhiteLightBrightness>
      <!--ro, opt, int, 远光白光亮度配置, range:[0,100]-->80
    </highWhiteLightBrightness>
    <lowWhiteLightBrightness>
      <!--ro, opt, int, 近光白光亮度配置, range:[0,100]-->80
    </lowWhiteLightBrightness>
    <whiteLightBrightness>
      <!--ro, opt, int, 白光亮度配置(不分远近光灯时控制), range:[0,100]-->80
    </whiteLightBrightness>
  </whiteLightModeCfg>
  <isAutoModeBrightnessCfg>
    <!--ro, opt, bool, 是否支持自动模式下亮度调节;支持时亮度节点下发为自动调节的亮度最大值-->true
  </isAutoModeBrightnessCfg>
  <irLightbrightnessLimit>
    <!--ro, opt, int, 红外灯亮度限制, range:[0,100], desc:当<Mode>== auto时且supplementLightMode==mixed模式下生效。-->80
  </irLightbrightnessLimit>
  <whiteLightbrightnessLimit>
    <!--ro, opt, int, 白光灯亮度限制, range:[0,100], desc:当<Mode>== auto时且supplementLightMode==mixed模式下生效。-->80
  </whiteLightbrightnessLimit>
  <isNotSupportMixedWhiteLight>
    <!--ro, opt, bool, 该节点返回true表示不支持混光模式下白光灯配置;支持配置则不返回该节点-->true
  </isNotSupportMixedWhiteLight>
  <EventIntelligenceModeCfg>
    <!--ro, opt, object, 事件智能补光模式参数配置, desc:当supplementLightMode == eventIntelligence生效-->
    <brightnessRegulatMode>
      <!--ro, opt, enum, 补光灯亮度调节模式, subType:string, [manual#手动,auto#自动], desc:手动模式下用户需手动<irLightBrightness>红外补光灯亮度参数以及<whiteLightBrightness>白光补光灯亮度;自动模式下用户无需配置亮度相关参数;由设备自动调节。-->auto
    </brightnessRegulatMode>
    <whiteLightBrightness>
      <!--ro, opt, int, 白光亮度配置(不分远近光灯时控制), range:[0,100]-->80
    </whiteLightBrightness>
    <irLightBrightness>

```

```
<!--ro, opt, int, 红外亮度配置(不分远近光灯时控制), range:[0,100]-->80
</irLightBrightness>
<BrightnessCtrlList>
  <!--ro, opt, array, 远近光亮度控制参数, subType:object-->
  <BrightnessCtrl>
    <!--ro, opt, object, 远近光亮度控制-->
    <lightType>
      <!--ro, opt, enum, 补光灯类型, subType:string, [highWhite#远光白光补光灯,lowWhite#近光白光补光灯,highIrLight#远光红外补光灯,lowIrLight#近光红外补光灯]-->highWhite
    </lightType>
    <brightness>
      <!--ro, opt, int, 补光灯亮度配置, range:[0,100]-->80
    </brightness>
  </BrightnessCtrl>
</BrightnessCtrlList>
</EventIntelligenceModeCfg>
<MultiAzimuthMixedModeCfg>
  <!--ro, opt, object, 多方位混合模式参数配置, desc:当supplementLightMode == multiAzimuthMixed生效-->
  <brightnessRegulatMode>
    <!--ro, opt, enum, 补光灯亮度调节模式, subType:string, [manual#手动,auto#自动], desc:手动模式下用户需手动配置各组补光灯的亮度(LightCtrlList): 自动模式下用户无需配置亮度相关参数;由设备自动调节.-->auto
  </brightnessRegulatMode>
  <LightCtrlList>
    <!--ro, opt, array, 补光灯组控制参数, subType:object-->
    <LightCtrl>
      <!--ro, opt, object, 单个补光灯控制参数-->
      <azimuthType>
        <!--ro, opt, enum, 补光灯所在的方位类型, subType:string, [left#左侧方位补光灯,right#右侧方位补光灯]-->left
      </azimuthType>
      <lightType>
        <!--ro, opt, enum, 补光灯类型, subType:string, [highWhite#远光白光补光灯,midWhite#中光白光补光灯,lowWhite#近光白光补光灯,highIrLight#远光红外补光灯,midIrLight#中光红外补光灯,lowIrLight#近光红外补光灯]-->highWhite
      </lightType>
      <brightness>
        <!--ro, opt, int, 补光灯亮度配置, range:[0,100]-->80
      </brightness>
    </LightCtrl>
  </LightCtrlList>
</MultiAzimuthMixedModeCfg>
<MultiAzimuthWhiteLightCfg>
  <!--ro, opt, object, 多方位白光模式参数配置, desc:当supplementLightMode == multiAzimuthWhiteLight生效-->
  <brightnessRegulatMode>
    <!--ro, opt, enum, 补光灯亮度调节模式, subType:string, [manual#手动,auto#自动], desc:手动模式下用户需手动配置各组补光灯的亮度(LightCtrlList): 自动模式下用户无需配置亮度相关参数;由设备自动调节.-->auto
  </brightnessRegulatMode>
  <LightCtrlList>
    <!--ro, opt, array, 补光灯组控制参数, subType:object-->
    <LightCtrl>
      <!--ro, opt, object, 单个补光灯控制参数-->
      <azimuthType>
        <!--ro, opt, enum, 补光灯所在的方位类型, subType:string, [left#左侧方位补光灯,right#右侧方位补光灯]-->left
      </azimuthType>
      <lightType>
        <!--ro, opt, enum, 补光灯类型, subType:string, [highWhite#远光白光补光灯,midWhite#中光白光补光灯,lowWhite#近光白光补光灯,white#普通白光补光灯(不区分远中近光)]-->highWhite
      </lightType>
      <brightness>
        <!--ro, opt, int, 补光灯亮度配置, range:[0,100]-->80
      </brightness>
    </LightCtrl>
  </LightCtrlList>
</MultiAzimuthWhiteLightCfg>
<MultiAzimuthIrLightCfg>
  <!--ro, opt, object, 多方位红外模式参数配置, desc:当supplementLightMode == multiAzimuthIrLight生效-->
  <brightnessRegulatMode>
    <!--ro, opt, enum, 补光灯亮度调节模式, subType:string, [manual#手动,auto#自动], desc:手动模式下用户需手动配置各组补光灯的亮度(LightCtrlList): 自动模式下用户无需配置亮度相关参数;由设备自动调节.-->auto
  </brightnessRegulatMode>
  <LightCtrlList>
    <!--ro, opt, array, 补光灯组控制参数, subType:object-->
    <LightCtrl>
      <!--ro, opt, object, 单个补光灯控制参数-->
      <azimuthType>
        <!--ro, opt, enum, 补光灯所在的方位类型, subType:string, [left#左侧方位补光灯,right#右侧方位补光灯]-->left
      </azimuthType>
      <lightType>
        <!--ro, opt, enum, 补光灯类型, subType:string, [highIrLight#远光红外补光灯,midIrLight#中光红外补光灯,lowIrLight#近光红外补光灯,irLight#普通红外补光灯(不区分远中近光)]-->highIrLight
      </lightType>
      <brightness>
        <!--ro, opt, int, 补光灯亮度配置, range:[0,100]-->80
      </brightness>
    </LightCtrl>
  </LightCtrlList>
</MultiAzimuthIrLightCfg>
<PanoramicWhiteLightCfg>
  <!--ro, opt, object, 全景白光模式参数配置, desc:当supplementLightMode == panoramicWhiteLight生效-->
  <brightnessRegulatMode>
    <!--ro, opt, enum, 补光灯亮度调节模式, subType:string, [manual#手动,auto#自动], desc:手动模式下用户需手动配置各组补光灯的亮度(LightCtrlList): 自动模式下用户无需配置亮度相关参数;由设备自动调节.-->auto
  </brightnessRegulatMode>
  <LightCtrlList>
    <!--ro, opt, array, 补光灯组控制参数, subType:object-->
    <LightCtrl>
```

```
<!--ro, opt, object, 单个补光灯控制参数-->
<lightID>
  <!--ro, opt, int, 补光灯序号, range:[1,4]-->1
</lightID>
<brightness>
  <!--ro, opt, int, 补光灯亮度配置, range:[0,100]-->80
</brightness>
</LightCtrl>
</LightCtrlList>
</PanoramicWhiteLightCfg>
<PanoramicIrLightCfg>
  <!--ro, opt, object, 全景红外模式参数配置, desc: 当supplementLightMode == panoramicIrLightCfg生效-->
  <brightnessRegulatMode>
    <!--ro, opt, enum, 补光灯亮度调节模式, subType:string, [manual#手动,auto#自动], desc: 手动模式下用户需手动配置各组补光灯的亮度 (LightCtrlList): 自动模式下用户无需配置亮度相关参数; 由设备自动调节.-->auto
  </brightnessRegulatMode>
  <LightCtrlList>
    <!--ro, opt, array, 补光灯组控制参数, subType:object-->
    <LightCtrl>
      <!--ro, opt, object, 单个补光灯控制参数-->
      <lightID>
        <!--ro, opt, int, 补光灯序号, range:[1,4]-->1
      </lightID>
      <brightness>
        <!--ro, opt, int, 补光灯亮度配置, range:[0,100]-->80
      </brightness>
      </LightCtrl>
    </LightCtrlList>
  </PanoramicIrLightCfg>
  <whiteLightEnhanceEnabled>
    <!--ro, opt, bool, 白光灯亮度增强使能(不分远近光灯时控制), desc: 开启增强使能后, 白光灯会进入强电流模式, 亮度会增强十倍, 同时也可通过$.SupplementLight.whiteLightBrightness参数去控制强电流模式下的灯光亮度。该参数不区分补光灯模式及白光灯类型.-->true
  </whiteLightEnhanceEnabled>
  <LightManualCtrlList>
    <!--ro, opt, array, 补光灯组手动控制参数, subType:object, dep:and,[$.ImageChannelList[*].ImageChannel.SupplementLight.mode,eq,manual], desc: 与supplementLightMode无关-->
    <LightCtrl>
      <!--ro, opt, object, 单个补光灯控制参数-->
      <lightType>
        <!--ro, opt, enum, 补光灯类型, subType:string, [highIrLight#远光红外补光灯,midIrLight#中光红外补光灯,lowIrLight#近光红外补光灯,irLight#普通红外补光灯(不区分远中近光),highWhite#远光白光补光灯,midWhite#中光白光补光灯,lowWhite#近光白光补光灯,white#普通红外补光灯(不区分远中近光)]-->highIrLight
      </lightType>
      <brightness>
        <!--ro, opt, int, 补光灯亮度配置, range:[0,100]-->80
      </brightness>
      </LightCtrl>
    </LightManualCtrlList>
  </SupplementLight>
<OpticalDehaze>
  <!--ro, opt, object, 光学雾透-->
  <enabled>
    <!--ro, opt, bool, 使能-->true
  </enabled>
</OpticalDehaze>
<ManualRanging>
  <!--ro, opt, object, 手动测距-->
  <enabled>
    <!--ro, opt, bool, 使能-->true
  </enabled>
</ManualRanging>
<OIS>
  <!--ro, opt, object, 光学防抖-->
  <mode>
    <!--ro, req, enum, 模式, subType:string, [close#关闭,ordinary#普通,experts#专家]-->close
  </mode>
  <OISLevel>
    <!--ro, opt, enum, 光学防抖等级, subType:string, [high#高,middle#中,low#低]-->high
  </OISLevel>
  <OISSensitivity>
    <!--ro, opt, string, 光学防抖灵敏度-->test
  </OISSensitivity>
</OIS>
<isSupportLaserSpotManual>
  <!--ro, opt, bool, 是否支持手动激光光点-->true
</isSupportLaserSpotManual>
<isSupportLaserSpotAdjustment>
  <!--ro, opt, bool, 是否支持激光光点调整-->true
</isSupportLaserSpotAdjustment>
<DigitalZoom>
  <!--ro, opt, object, 数字缩放-->
  <ZoomRatio>
    <!--ro, opt, enum, 缩放比例, subType:string, [1x#1x,2x#2x,4x#4x,8x#8x,16x#16x,32x#32x]-->1x
  </ZoomRatio>
</DigitalZoom>
<Palettes>
  <!--ro, opt, object, 热成像调色板配置-->
  <mode>
    <!--ro, opt, enum, 模式, subType:string, [WhiteHot#白热,BlackHot#黑热,Fusion1#融合1,Rainbow#彩虹,Fusion2#融合2,Ironbow1#铁红1,Ironbow2#铁红2,Sepia#深褐色,Color1#色彩1,Color2#色彩2,IceFire#冰火,Rain#雨,RedHot#红热,GreenHot#绿热,DeepBlue#深蓝,Color3#色彩3]-->WhiteHot
  </mode>
  <ColorateTarget>
    <!--ro, opt, object, 目标凸显, desc:dep mode = "WhiteHot"-->
    <ColorateTargetModelList>
```



```
<!--ro, opt, object, 自定义区域参数, desc:dep <mode>= customArea-->
<RegionCoordinatesList>
<!--ro, opt, array, 区域坐标列表, subType:object, range:[3,10]-->
<RegionCoordinates>
<!--ro, opt, object, 区域坐标, desc:屏幕左上角为坐标原点-->
<positionX>
<!--ro, req, int, x坐标, range:[0,1000]-->0
</positionX>
<positionY>
<!--ro, req, int, y坐标, range:[0,1000]-->0
</positionY>
</RegionCoordinates>
</RegionCoordinatesList>
</CustomAreaParam>
<DefaultAreaParamList>
<!--ro, opt, array, 默认区域参数列表, subType:object, desc:dep <mode>="up_60%,down_60%,left_60%,right_60%,center_50%,center_75%"-->
<DefaultAreaParam>
<!--ro, opt, object, 默认区域参数-->
<mode>
<!--ro, req, enum, 模式, subType:string,
[up_60%#up_60%,down_60%#down_60%,left_60%#left_60%,right_60%#right_60%,center_50%#center_50%,center_75%#center_75%]-->up_60%
</mode>
<RegionCoordinatesList>
<!--ro, opt, array, 区域坐标列表, subType:object, range:[3,10]-->
<RegionCoordinates>
<!--ro, opt, object, 区域坐标, desc:屏幕左上角为坐标原点-->
<positionX>
<!--ro, req, int, x坐标, range:[0,1000]-->0
</positionX>
<positionY>
<!--ro, req, int, y坐标, range:[0,1000]-->0
</positionY>
</RegionCoordinates>
</RegionCoordinatesList>
</DefaultAreaParam>
</DefaultAreaParamList>
</RegionalImageEnhancementCap>
<faceExposure>
<!--ro, opt, object, 人脸曝光-->
<faceExposureEnabled>
<!--ro, req, bool, 人脸曝光使能-->true
</faceExposureEnabled>
<faceExposureSensitivity>
<!--ro, req, int, 人脸曝光灵敏度, range:[0,100]-->0
</faceExposureSensitivity>
</faceExposure>
<lightHal>
<!--ro, opt, object, 信号灯光量抑制-->
<enabled>
<!--ro, req, bool, 使能-->true
</enabled>
<halLevel>
<!--ro, req, int, 光量抑制等级, range:[0,100]-->0
</halLevel>
<halRange>
<!--ro, req, int, 光量抑制范围, range:[0,2064]-->0
</halRange>
</lightHal>
<TargetEnhancementCap>
<!--ro, opt, object, 图像目标增强配置-->
<enabled>
<!--ro, req, bool, 使能-->true
</enabled>
</TargetEnhancementCap>
<LaserLight>
<!--ro, opt, object, 激光补光-->
<mode>
<!--ro, opt, enum, 模式, subType:string, [auto#自动,manual#手动]-->auto
</mode>
<compensationRegionMode>
<!--ro, opt, enum, 激光补光区域, subType:string, [center#中心,adaptive#适应,full_Screen#全屏]-->center
</compensationRegionMode>
<compensationLevel>
<!--ro, opt, enum, 激光补光强度, subType:string, [low#低,medium#中,high#高]-->low
</compensationLevel>
<LaserColor>
<!--ro, opt, object, 激光颜色, desc:依赖<mode>为manual-->
<R>
<!--ro, req, int, R(红色)值, range:[0,100]-->0
</R>
<G>
<!--ro, req, int, G(绿色)值, range:[0,100]-->0
</G>
<B>
<!--ro, req, int, B(蓝色)值, range:[0,100]-->0
</B>
</LaserColor>
</LaserLight>
<ExternalSupplementLightCap>
<!--ro, opt, object, 外接补光灯参数配置能力-->
<mode>
<!--ro, opt, enum, 模式, subType:string, [close#关闭,oneMixedLight#一组混合补光]-->close
</mode>
<BrightnessList>
```

```
<!--ro, opt, array, 亮度列表, subType:object-->
<Brightness>
  <!--ro, opt, object, 亮度-->
  <id>
    <!--ro, opt, int, 每组补光灯的id-->0
  </id>
  <irLightBrightness>
    <!--ro, opt, int, 红外灯亮度, range:[0,100]-->0
  </irLightBrightness>
  <whiteLightBrightness>
    <!--ro, opt, int, 白光灯亮度, range:[0,100]-->0
  </whiteLightBrightness>
</Brightness>
</BrightnessList>
</ExternalSupplementLightCap>
<TempRange>
  <!--ro, opt, object, 温度范围-->
  <mode>
    <!--ro, req, enum, 温度范围模式, subType:string, [automatic#自动模式,manual#手动模式]-->automatic
  </mode>
  <temperatureUpperLimit>
    <!--ro, opt, float, 上限温度, range:[-20,550], unit:℃, unitType:温度, desc:精确到小数点后一位,手动温度范围模式下有效,上限温度需大于下限温度-->0.000
  </temperatureUpperLimit>
  <temperatureLowerLimit>
    <!--ro, opt, float, 下限温度, range:[-20,550], unit:℃, unitType:温度, desc:精确到小数点后一位,手动温度范围模式下有效,下限温度需小于上限温度-->0.000
  </temperatureLowerLimit>
</TempRange>
<BeautyCap>
  <!--ro, opt, object, 美颜能力-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <whiteningStrength>
    <!--ro, opt, int, 美白强度, range:[0,10]-->1
  </whiteningStrength>
  <skinSmoothingStrength>
    <!--ro, opt, int, 磨皮强度, range:[0,10]-->1
  </skinSmoothingStrength>
</BeautyCap>
<MExposure>
  <!--ro, opt, object, 手动曝光(工厂标定)-->
  <enabled>
    <!--ro, opt, bool, 使能-->true
  </enabled>
  <exposureLevel>
    <!--ro, opt, string, 曝光等级, range:[0,1000000]-->test
  </exposureLevel>
  <gainLevel>
    <!--ro, opt, int, 增益等级, range:[0,100]-->0
  </gainLevel>
</MExposure>
<SensorRGBCtrl>
  <!--ro, opt, object, SensorRGB控制-->
  <sensorID>
    <!--ro, opt, int, SensorID, range:[0,1]-->0
  </sensorID>
</SensorRGBCtrl>
<YPbPrCalibration>
  <!--ro, opt, object, 色差标定-->
  <mode>
    <!--ro, opt, int, 模式, range:[0,2]-->1
  </mode>
</YPbPrCalibration>
<intrusionTriggeredLightingCfg>
  <!--ro, opt, object, 随影补光-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
</intrusionTriggeredLightingCfg>
<shotProCfg>
  <!--ro, opt, object, 幕影-->
  <shotProMode>
    <!--ro, req, enum, 幕影模式, subType:string, [close#关闭,general#通用,enhancement#增强]-->general
  </shotProMode>
</shotProCfg>
</ImageChannel>
</ImageChannellist>
```

69.24.11 配置所有通道图像处理参数

Request URL

PUT /ISAPI/Image/channels

查询参数

无

请求报文

<?xml version="1.0" encoding="UTF-8"?>

<ImageChannellist xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">

<!--req, array, 图像信号处理参数列表, subType:object, attr:version{req, string, 协议版本}-->

<ImageChannel>

<!--req, object, 图像信号处理参数-->

<id>

<!--req, int, 序号ID-->1

</id>

<enabled>

<!--req, bool, 使能-->true

</enabled>

<videoInputID>

<!--req, int, 视频通道号ID-->1

</videoInputID>

<FocusConfiguration>

<!--opt, object, 普通聚焦-->

<focusStyle>

<!--req, enum, 聚焦类型, subType:string, [AUTO#自动,MANUAL#手动,SEMIAUTOMATIC#半自动]-->AUTO

</focusStyle>

<focusLimited>

<!--opt, enum, 聚焦限制, subType:int,

[10#10,50#50,100#100,150#150,300#300,600#600,1000#1000,2000#2000,5000#5000,20000#20000,50000#50000,65535#65535]-->10

</focusLimited>

<focusPosition>

<!--opt, int, 焦点位置, range:[1,10]-->1

</focusPosition>

<focusSpeed>

<!--opt, int, 聚焦速度, range:[1,3]-->1

</focusSpeed>

<focusSensitivity>

<!--opt, int, 聚焦灵敏度, range:[0,2]-->1

</focusSensitivity>

<temperatureChangeAdaptEnabled>

<!--opt, bool, 温度变化适应使能-->true

</temperatureChangeAdaptEnabled>

<relativeFocusPos>

<!--opt, int, 相对焦点位置, range:[0,4294967295]-->1

</relativeFocusPos>

<lensType>

<!--opt, enum, 聚焦镜头类型, subType:string, [1140#1140,1050#1050]-->1140

</lensType>

<highTemperaturePriority>

<!--opt, bool, 高温优先模式使能-->true

</highTemperaturePriority>

<focusRange>

<!--opt, enum, 聚焦调节范围, subType:string, [min#最小,medium#适中,max#最大]-->max

</focusRange>

<focusDistanceMode>

<!--opt, enum, 聚焦距离模式, subType:string, [compatible#兼容模式], desc:compatible-兼容模式, 在球罩设备上调整聚焦限制距离focusLimited,仍然虚焦的情况, 选择兼容模式, focusLimited参数失效, 设备释放镜头最大聚焦能力, 使镜头获得更大的焦距范围, 解决球罩设备近物距聚焦效果差的问题. 开启该功能, 聚焦速度会变慢, 非球罩设备不建议使用. -->compatible

</focusDistanceMode>

</FocusConfiguration>

<LensInitialization>

<!--opt, object, 镜头初始化-->

<enabled>

<!--req, bool, 使能-->true

</enabled>

</LensInitialization>

<ImageFlip>

<!--opt, object, 图像翻转-->

<enabled>

<!--req, bool, 使能-->true

</enabled>

<ImageFlipStyle>

<!--opt, enum, 镜像类型, subType:string, [LEFT#LEFT,UPDOWN#UPDOWN,CENTER#CENTER,AUTO#AUTO]-->LEFT

</ImageFlipStyle>

<flipAngle>

<!--opt, enum, 旋转角度, subType:int, [90#90,180#180,270#270]-->90

</flipAngle>

</ImageFlip>

<ImageFreeze>

<!--opt, object, 图像冻结-->

<enabled>

<!--req, bool, 使能-->true

</enabled>

</ImageFreeze>

<proportionalpan>

<!--opt, object, 比例盘-->

<enabled>

<!--req, bool, 使能-->true

</enabled>

</proportionalpan>

<WDR>

<!--opt, object, 宽动态(自动)能力-->

<mode>

<!--req, enum, open, subType:string, [open#开启,close#关闭,auto#自动,SDR#SDR]-->open

</mode>

<WDRLevel>

<!--opt, int, 宽动态等级, range:[1,100]-->1

</WDRLevel>

```

<WDRContrastLevel>
  <!--opt, int, 宽动态对比度, range:[1,100]-->1
</WDRContrastLevel>
<WDRLevel1>
  <!--opt, int, 第二类宽动态等级, range:[1,100]-->1
</WDRLevel1>
</WDR>
<BLC>
  <!--opt, object, 背光补偿区域-->
  <enabled>
    <!--req, bool, 使能-->true
  </enabled>
  <BLCMode>
    <!--opt, enum, 背光补偿模式, subType:string, [UP#UP,DOWN#DOWN,LEFT#LEFT,RIGHT#RIGHT,CENTER#CENTER,MULTI-AREA#MULTI-AREA,Region#Region,AUTO#AUTO]-->UP
  </BLCMode>
  <BLCLevel>
    <!--opt, int, 背光补偿等级, range:[0,10]-->0
  </BLCLevel>
  <BLCRegionList>
    <!--opt, array, 背光补偿区域列表, subType:object-->
    <BLCRegion>
      <!--opt, object, 背光补偿区域-->
      <id>
        <!--req, int, 区域编号, range:[1,10]-->1
      </id>
      <RegionCoordinatesList>
        <!--req, array, 坐标点列表, subType:object, range:[0,4], desc:矩形-->
        <RegionCoordinates>
          <!--opt, object, 坐标点, desc:屏幕左下角为坐标原点-->
          <positionX>
            <!--req, int, x坐标, range:[0,704]-->1
          </positionX>
          <positionY>
            <!--req, int, y坐标, range:[0,576]-->1
          </positionY>
        </RegionCoordinates>
      </RegionCoordinatesList>
    </BLCRegion>
  </BLCRegionList>
</BLC>
<NoiseReduce>
  <!--opt, object, 降噪参数-->
  <mode>
    <!--req, enum, 模式, subType:string, [close#关闭,general#常规,advanced#高级]-->close
  </mode>
  <GeneralMode>
    <!--opt, object, 模式-->
    <generalLevel>
      <!--req, int, 等级, range:[0,100]-->1
    </generalLevel>
  </GeneralMode>
  <AdvancedMode>
    <!--opt, object, 高级模式-->
    <FrameNoiseReduceLevel>
      <!--req, int, 框架噪声降阶, range:[0,100]-->1
    </FrameNoiseReduceLevel>
    <InterFrameNoiseReduceLevel>
      <!--req, int, 帧间噪声降维, range:[0,100]-->1
    </InterFrameNoiseReduceLevel>
    <secondarySpatialNoiseReduceLevel>
      <!--opt, int, 二级降噪空域等级, range:[0,100]-->1
    </secondarySpatialNoiseReduceLevel>
    <secondaryTemporalNoiseReduceLevel>
      <!--opt, int, 二级降噪时域等级, range:[0,100]-->1
    </secondaryTemporalNoiseReduceLevel>
  </AdvancedMode>
</NoiseReduce>
<ImageEnhancement>
  <!--opt, object, 图像增强-->
  <enabled>
    <!--req, bool, 使能-->true
  </enabled>
  <ImageEnhancementLevel>
    <!--opt, int, 图像增强等级, range:[0,10]-->0
  </ImageEnhancementLevel>
</ImageEnhancement>
<DSS>
  <!--opt, object, DSS-->
  <enabled>
    <!--req, bool, 使能-->true
  </enabled>
  <DSSLevel>
    <!--opt, enum, 低照度等级, subType:string,
    [*1.25#*1.25,*1.5#*1.5,*2#*2,*3#*3,*4#*4,*5#*5,*6#*6,*8#*8,*12#*12,*16#*16,*24#*24,*32#*32,*48#*48,*64#*64,*96#*96,*128#*128,*256#*256,*512#*512,auto#auto]->*1.25
  </DSSLevel>
  <mode>
    <!--opt, enum, 模式, subType:string, [frame#帧模式,Level#等级模式], desc:无mode节点默认为frame模式-->frame
  </mode>
  <slowShutterLevel>
    <!--opt, int, 慢快门等级, range:[1,6]-->1
  </slowShutterLevel>
</DSS>

```

```
</uds>
<WhiteBalance>
  <!--opt, object, 白平衡-->
  <WhiteBalanceStyle>
    <!--opt, enum, 白平衡类型, subType:string, [auto#自动,auto1#自动1,auto2#自动2>manual#手动,indoor#室内,outdoor#室外,autotrace#自动跟踪,onece#单
    次,sodiumLight#苏打光,mercuryLight#水银灯,customColorTemperature#自定义色温]-->auto
  </WhiteBalanceStyle>
  <WhiteBalanceRed>
    <!--opt, int, 白平衡红, range:[0,100]-->1
  </WhiteBalanceRed>
  <WhiteBalanceBlue>
    <!--opt, int, 白平衡蓝, range:[0,100]-->1
  </WhiteBalanceBlue>
  <WhiteBalanceLevel>
    <!--opt, int, 白平衡等级, range:[0,100], desc:<WhiteBalanceStyle>为auto,auto1,auto2时有效-->1
  </WhiteBalanceLevel>
  <customColorTemperatureLevel>
    <!--opt, int, 自定义色温等级, range:[2700,7500]-->1
  </customColorTemperatureLevel>
</WhiteBalance>
<Exposure>
  <!--opt, object, 曝光模式-->
  <ExposureType>
    <!--req, enum, 镜头类型, subType:string,
    [auto#auto,IrisFirst#IrisFirst,ShutterFirst#ShutterFirst,gainFirst#gainFirst>manual#manual,pIris#pIris,T5280-PQ1#T5280-PQ1,T5289-PQ1#T5289-PQ1,T1140-
    PQ1#T1140-PQ1,T2712-PQ1#T2712-PQ1,HV1250P-MPIR#HV1250P-MPIR,pc#pc,pIris-General#pIris-General,HZ1140P-8MPIRA#HZ1140P-8MPIRA,HZ1135P-8MPSGA#HZ1135P-
    8MPSGA,MV1550P-12MPIR#MV1550P-12MPIR,MZ1545P-12MP#MZ1545P-12MP,HZ0618P-4MPSG#HZ0618P-4MPSG,MZ1050P-12MP#MZ1050P-12MP]-->auto
  </ExposureType>
  <autoIrisLevel>
    <!--opt, int, 光圈自动级别, range:[0,100]-->1
  </autoIrisLevel>
  <OverexposeSuppress>
    <!--opt, object, 过度曝光抑制-->
    <enabled>
      <!--req, bool, 使能-->true
    </enabled>
    <Type>
      <!--opt, enum, 过度曝光抑制类型, subType:string, [AUTO#自动,MANUAL#手动]-->AUTO
    </Type>
    <DistanceLevel>
      <!--opt, int, 焦距, range:[1,100]-->1
    </DistanceLevel>
    <shortIRDistanceLevel>
      <!--opt, int, IR短距离级别, range:[1,100]-->1
    </shortIRDistanceLevel>
    <longIRDistanceLevel>
      <!--opt, int, IR长距离级别, range:[1,100]-->1
    </longIRDistanceLevel>
    <supplementLightIntensity1>
      <!--opt, int, 补充曝光强度1, range:[1,100]-->1
    </supplementLightIntensity1>
    <supplementLightIntensity2>
      <!--opt, int, 补充曝光强度2, range:[1,100]-->1
    </supplementLightIntensity2>
    <supplementLightIntensity3>
      <!--opt, int, 补充曝光强度3, range:[1,100]-->1
    </supplementLightIntensity3>
    <supplementLightIntensity4>
      <!--opt, int, 补充曝光强度4, range:[1,100]-->1
    </supplementLightIntensity4>
  </OverexposeSuppress>
  <pIris>
    <!--opt, object, 光圈-->
    <pIrisType>
      <!--opt, enum, 光圈类型, subType:string, [AUTO#AUTO,MANUAL#MANUAL]-->MANUAL
    </pIrisType>
    <IrisLevel>
      <!--opt, int, 光圈级别, range:[1,100]-->1
    </IrisLevel>
  </pIris>
  <faceExposure>
    <!--opt, object, 人脸曝光-->
    <enabled>
      <!--opt, bool, 使能-->true
    </enabled>
    <sensitivity>
      <!--opt, int, 灵敏度, range:[0,100]-->0
    </sensitivity>
  </faceExposure>
  <PIrisGeneral>
    <!--opt, object, pIris-General配置-->
    <irisLevel>
      <!--opt, int, 通用镜头下的光圈等级, range:[1,100]-->1
    </irisLevel>
  </PIrisGeneral>
</Exposure>
<Sharpness>
  <!--opt, object, 锐度-->
  <SharpnessMode>
    <!--opt, enum, 锐度模式, subType:string, [manual#手动,auto#自动]-->manual
  </SharpnessMode>
  <SharpnessLevel>
    <!--req, int, 锐度级别, range:[0,100]-->1
  </SharpnessLevel>
```

```
</Sharpness>
<gammaCorrection>
  <!--opt, object, 伽玛校正-->
  <gammaCorrectionEnabled>
    <!--opt, bool, 伽玛校正使能-->true
  </gammaCorrectionEnabled>
  <gammaCorrectionLevel>
    <!--opt, int, 伽玛校正级别, range:[0,100]-->1
  </gammaCorrectionLevel>
</gammaCorrection>
<Shutter>
  <!--opt, object, 快门配置-->
  <ShutterLevel>
    <!--opt, enum, 曝光等级, subType:string,
[1/1#1/1,1/2#1/2,1/3#1/3,1/6#1/6,1/12#1/12,1/25#1/25,1/50#1/50,1/75#1/75,1/100#1/100,1/120#1/120,1/125#1/125,1/150#1/150,1/175#1/175], desc:depends on
<ExposureType>-->1/1
  </ShutterLevel>
  <maxShutterLevelLimit>
    <!--opt, enum, 最大曝光等级, subType:string,
[1/1#1/1,1/2#1/2,1/3#1/3,1/6#1/6,1/12#1/12,1/25#1/25,1/50#1/50,1/75#1/75,1/100#1/100,1/120#1/120,1/125#1/125,1/150#1/150,1/175#1/175]-->1/1
  </maxShutterLevelLimit>
  <minShutterLevelLimit>
    <!--opt, enum, 最小曝光等级, subType:string,
[1/1#1/1,1/2#1/2,1/3#1/3,1/6#1/6,1/12#1/12,1/25#1/25,1/50#1/50,1/75#1/75,1/100#1/100,1/120#1/120,1/125#1/125,1/150#1/150,1/175#1/175]-->1/1
  </minShutterLevelLimit>
</Shutter>
<powerLineFrequency>
  <!--opt, object, 视频制式-->
  <powerLineFrequencyMode>
    <!--opt, enum, 视频制式模式, subType:string, [50hz#50hz,60hz#60hz]-->50hz
  </powerLineFrequencyMode>
</powerLineFrequency>
<Gain>
  <!--opt, object, 增益-->
  <GainLevel>
    <!--opt, string, 增益等级, range:[0,10]-->test
  </GainLevel>
  <GainLimit>
    <!--opt, enum, 增益, subType:string,
[0#0,400#400,900#900,1300#1300,1800#1800,2200#2200,2600#2600,3100#3100,3500#3500,4000#4000,4400#4400,4800#4800,5300#5300,5700#5700,6200#6200,6600#6600]-->0
  </GainLimit>
</Gain>
<Color>
  <!--opt, object, 颜色-->
  <brightnessLevel>
    <!--opt, int, 亮度, range:[0,100]-->0
  </brightnessLevel>
  <contrastLevel>
    <!--opt, int, 对比度, range:[0,100]-->0
  </contrastLevel>
  <saturationLevel>
    <!--opt, int, 饱和度, range:[0,100]-->0
  </saturationLevel>
  <hueLevel>
    <!--opt, int, 色调, range:[0,100]-->0
  </hueLevel>
  <grayScale>
    <!--opt, object, 灰度-->
    <grayScaleMode>
      <!--opt, enum, 灰度模式, subType:string, [indoor#室内,outdoor#室外]-->indoor
    </grayScaleMode>
  </grayScale>
  <nightMode>
    <!--opt, bool, 夜间黑白模式使能-->true
  </nightMode>
  <sharpnessLevel>
    <!--opt, int, 锐度, range:[0,100]-->0
  </sharpnessLevel>
</Color>
<IrcutFilter>
  <!--opt, object, IrcutFilter配置-->
  <IrcutFilterType>
    <!--opt, enum, IrcutFilter类型, subType:string,
[auto#auto,day#day,night#night,schedule#schedule,eventTrigger#eventTrigger,videoAuto#videoAuto,darkFighterX#darkFighterX,darkFighterXAuto#darkFighterXAuto,d
arkFighterXSchedLe#darkFighterXSchedule]-->auto
  </IrcutFilterType>
  <dayToNightFilterLevel>
    <!--opt, enum, 日夜切换级别, subType:string, [Low#低,normal#中,high#高]-->low
  </dayToNightFilterLevel>
  <dayToNightFilterTime>
    <!--opt, int, 日夜切换时间, range:[0,10]-->1
  </dayToNightFilterTime>
  <nightToDayFilterLevel>
    <!--opt, enum, 夜日切换级别, subType:string, [Low#低,normal#中,high#高]-->low
  </nightToDayFilterLevel>
  <nightToDayFilterTime>
    <!--opt, int, 夜日切换时间, range:[0,10]-->1
  </nightToDayFilterTime>
  <Schedule>
    <!--opt, object, 定时-->
    <scheduleType>
      <!--req, enum, 定时类型, subType:string, [day#白天,night#晚上]-->day
    </scheduleType>
  </Schedule>
-->
```

```
<TimeRange>
  <!--req, object, 时间段-->
  <beginTime>
    <!--req, time, 开始时间-->00:00:00+08:00
  </beginTime>
  <endTime>
    <!--req, time, 结束时间-->00:00:00+08:00
  </endTime>
</TimeRange>
</Schedule>
<EventTrigger>
  <!--opt, object, 事件触发-->
  <eventType>
    <!--req, enum, 事件类型, subType:string, [IO#IO事件,VMD#VMD事件]-->IO
  </eventType>
  <IrcutFilterAction>
    <!--req, enum, IrcutFilter操作, subType:string, [day#白天,night#晚上]-->day
  </IrcutFilterAction>
</EventTrigger>
</IrcutFilter>
<Scene>
  <!--opt, object, 场景-->
  <mode>
    <!--req, enum, 模式, subType:string, [indoor#户内,outdoor#户外,standard#标准,dimLight#昏暗]-->indoor
  </mode>
</Scene>
<PTZ>
  <!--opt, object, PTZ-->
  <enabled>
    <!--opt, bool, 使能-->true
  </enabled>
</PTZ>
<EPTZ>
  <!--opt, object, EPTZ-->
  <enabled>
    <!--req, bool, 使能-->true
  </enabled>
  <mode>
    <!--opt, enum, 模式, subType:string, [cruise#巡航（云台可以支持预置点、巡航功能、ZOOM和8个方向）,autoTrack#autoTrack-自动跟踪]-->cruise
  </mode>
</EPTZ>
<EIS>
  <!--opt, object, 电子防抖-->
  <enabled>
    <!--req, bool, 使能-->true
  </enabled>
  <EISLevel>
    <!--opt, int, EIS等级, range:[0,10]-->0
  </EISLevel>
</EIS>
<HLC>
  <!--opt, object, 强光抑制-->
  <enabled>
    <!--req, bool, 使能-->true
  </enabled>
  <HLCLevel>
    <!--opt, int, HLC等级, range:[0,10]-->0
  </HLCLevel>
</HLC>
<ZoomLimit>
  <!--opt, object, 缩放限制-->
  <ZoomLimitRatio>
    <!--opt, enum, 缩放比例, subType:int, [4#4,8#8,16#16,32#32,64#64]-->4
  </ZoomLimitRatio>
</ZoomLimit>
<corridor>
  <!--opt, object, 镜头走廊模式-->
  <enabled>
    <!--req, bool, 使能-->true
  </enabled>
</corridor>
<Dehaze>
  <!--opt, object, 雾透-->
  <DehazeMode>
    <!--opt, enum, 雾透模式, subType:string, [open#打开,close#关闭,auto#自动]-->open
  </DehazeMode>
  <DehazeLevel>
    <!--opt, int, 雾透等级, range:[0,10]-->0
  </DehazeLevel>
</Dehaze>
<ImageMode>
  <!--opt, enum, 图像模式, subType:string, [standard#标准,indoor#室内,outdoor#室外,dimLight#昏暗]-->standard
</ImageMode>
<enableImageLossDetection>
  <!--opt, bool, 图像丢失检测使能-->true
</enableImageLossDetection>
<CaptureMode>
  <!--opt, object, 抓拍模式-->
  <mode>
    <!--req, enum, 模式, subType:string, [close#关闭,1920*1080@30fps#1920*1080@30fps]-->close
  </mode>
</CaptureMode>
<IrLight>
  <!--opt, object, 亮度限制-->
```

```
<mode>
  <!--req, enum, 模式, subType:string, [auto#自动,manual#手动]-->auto
</mode>
<brightnessLevel>
  <!--opt, int, 亮度等级, range:[0,100]-->1
</brightnessLevel>
<sensitivityLevel>
  <!--opt, int, 灵敏度, range:[0,100]-->1
</sensitivityLevel>
<brightnessLimit>
  <!--opt, int, 亮度限制, range:[0,100]-->1
</brightnessLimit>
<triggerMode>
  <!--opt, enum, 触发模式, subType:string, [camera#相机,photosensitive#光敏]-->camera
</triggerMode>
<IrLight>
<LensDistortionCorrection>
  <!--opt, object, 镜头畸变校正-->
  <enabled>
    <!--req, bool, 使能-->true
  </enabled>
  <level>
    <!--opt, enum, 等级, subType:string, [Level1#Level1,Level2#Level2,Level3#Level3,custom#自定义]-->level1
  </level>
  <accurateLevel>
    <!--opt, int, 畸变矫正强度, range:[0,100]-->0
  </accurateLevel>
  <zoomedInDistantViewLevel>
    <!--opt, int, 远端放大, range:[0,100], desc:畸变校正开启时生效-->0
  </zoomedInDistantViewLevel>
  <horizontalFOV>
    <!--opt, int, 水平视场角, range:[0,100]-->0
  </horizontalFOV>
  <verticalFOV>
    <!--opt, int, 垂直视场角, range:[0,100]-->0
  </verticalFOV>
</LensDistortionCorrection>
<SupplementLight>
  <!--opt, object, 内置补光灯管理参数能力-->
  <mode>
    <!--opt, enum, 补光灯调节模式, subType:string, [schedule#白光定时,off#白光关闭,on#白光开启,auto#自动,manual#手动]-->auto
  </mode>
  <Schedule>
    <!--opt, object, 白光定时生效时间配置;当<Mode>= schedule时生效;所配置时间为白天的时间-->
    <TimeRange>
      <!--req, object, 白天时段;当天剩余时段为夜晚时段-->
      <beginTime>
        <!--req, time, 开始时间-->00:08:00+08:00
      </beginTime>
      <endTime>
        <!--req, time, 结束时间-->00:18:00+08:00
      </endTime>
    </TimeRange>
  </Schedule>
  <brightnessLimit>
    <!--opt, int, 白光补光灯亮度限制, range:[0,100], desc:当<Mode>= on时;可以调节该限制;为白光补光灯亮度最大值;实际补光灯亮度为自动调节(注: mode里面的自动为补光灯开光自动;自动调节模式需查看mixedLightBrightnessRegulateMode节点)-->80
  </brightnessLimit>
  <supplementLightMode>
    <!--opt, enum, 补光灯模式, subType:string, [mixed#混合模式,whiteLight#白光模式,close#关闭,colorVwWhiteLight#全彩白光,irLight#红外模式,dualLight#双补光,eventIntelligence#事件智能补光,multiAzimuthMixed#多方位混合补光,multiAzimuthWhiteLight#多方位白光模式,multiAzimuthIrLight#多方位红外模式,panoramicWhiteLight#全景白光,panoramicIrLight#全景红外],
    desc:eventIntelligence为事件智能补光;解释如下:1. 在满足补光灯功能生效条件下;当没有产生报警;设备仅红外灯开启;2. 产生报警后自动关闭红外灯; 开启白光灯持续1min;
    在持续时间内如继续收到报警;这一状态保持至最后一个报警发生后1min结束。multiAzimuthMixed多方位混合模式可以通过左右边各一组补光灯(一组补光灯包含远中近白光红外补光灯)对目标进行多方位角度的补光;不同于之前mixed混合补光模式仅支持一组补光灯单方位进行补光。multiAzimuthWhiteLight多方位白光模式、multiAzimuthIrLight多方位红外模式与多方位混光情况相同。
    panoramicWhiteLight、panoramicIrLight全景白光、红外模式当前仅四目全景相机支持, 当前下相机无方位的概念,由四个补光灯围绕相机部署。-->mixed
  </supplementLightMode>
  <irLightBrightness>
    <!--opt, int, 红外补光灯亮度, range:[0,100], desc:当<supplementLightMode>= mixed;且<mixedLightBrightnessRegulateMode>= manual时生效-->80
  </irLightBrightness>
  <mixedLightBrightnessRegulateMode>
    <!--opt, enum, 混合补光灯亮度调节模式, subType:string, [manual#手动,auto#自动], desc:该字段仅在<supplementLightMode>= mixed时生效;手动模式下用户需手动<irLightBrightness>红外补光灯亮度参数以及<brightnessLimit>白光补光灯亮度;自动模式下用户无需配置亮度相关参数;由设备自动调节。-->auto
  </mixedLightBrightnessRegulateMode>
  <highIrLightBrightness>
    <!--opt, int, 远光红外光亮度配置, range:[0,100], desc:参数由mixed#混合模式、whiteLight#白光模式、colorVwWhiteLight#全彩白光,irLight#红外模式、dualLight#双补光模式公用-->80
  </highIrLightBrightness>
  <highWhiteLightBrightness>
    <!--opt, int, 远光白光亮度配置, range:[0,100], desc:参数由mixed#混合模式、whiteLight#白光模式、colorVwWhiteLight#全彩白光,irLight#红外模式、dualLight#双补光模式公用-->80
  </highWhiteLightBrightness>
  <lowIrLightBrightness>
    <!--opt, int, 近光红外光亮度配置, range:[0,100], desc:参数由mixed#混合模式、whiteLight#白光模式、colorVwWhiteLight#全彩白光,irLight#红外模式、dualLight#双补光模式公用-->80
  </lowIrLightBrightness>
  <lowWhiteLightBrightness>
    <!--opt, int, 近光白光亮度配置, range:[0,100], desc:参数由mixed#混合模式、whiteLight#白光模式、colorVwWhiteLight#全彩白光,irLight#红外模式、dualLight#双补光模式公用-->80
  </lowWhiteLightBrightness>
  <whiteLightBrightness>
    <!--opt, int, 光亮度配置(不分远近光灯时控制), range:[0,100], desc:参数由mixed#混合模式、whiteLight#白光模式、colorVwWhiteLight#全彩白光,irLight#红外
```

模式、dualLight#双补光模式、mode=auto 公用-->80

```
</whiteLightBrightness>
<whiteLightModeCfg>
  <!--opt, object, 白光模式参数配置, desc: 当supplementLightMode == whiteLight生效-->
  <highWhiteLightBrightness>
    <!--opt, int, 远光白光亮度配置, range:[0,100]-->80
  </highWhiteLightBrightness>
  <lowWhiteLightBrightness>
    <!--opt, int, 近光白光亮度配置, range:[0,100]-->80
  </lowWhiteLightBrightness>
  <whiteLightBrightness>
    <!--opt, int, 白光亮度配置(不分远近光灯时控制), range:[0,100]-->80
  </whiteLightBrightness>
</whiteLightModeCfg>
<isAutoModeBrightnessCfg>
  <!--opt, bool, 是否支持自动模式下亮度调节;支持时亮度节点下发为自动调节的亮度最大值-->true
</isAutoModeBrightnessCfg>
<irLightbrightnessLimit>
  <!--opt, int, 红外灯亮度限制, range:[0,100], desc: 当<Mode>== auto时且supplementLightMode==mixed模式下生效.-->80
</irLightbrightnessLimit>
<whiteLightbrightnessLimit>
  <!--opt, int, 白光灯亮度限制, range:[0,100], desc: 当<Mode>== auto时且supplementLightMode==mixed模式下生效.-->80
</whiteLightbrightnessLimit>
<isNotSupportMixedWhiteLight>
  <!--opt, bool, 该节点返回true表示不支持混光模式下白光灯配置;支持配置则不返回该节点-->true
</isNotSupportMixedWhiteLight>
<EventIntelligenceModeCfg>
  <!--opt, object, 事件智能补光模式参数配置, desc: 当supplementLightMode == eventIntelligence生效-->
  <brightnessRegulatMode>
    <!--opt, enum, 补光灯亮度调节模式, subType:string, [manual#手动,auto#自动], desc: 手动模式下用户需手动配置补光灯亮度; 自动模式下用户无需配置亮度相关参数;由设备自动调节.-->auto
  </brightnessRegulatMode>
  <whiteLightBrightness>
    <!--opt, int, 白光亮度配置(不分远近光灯时控制), range:[0,100]-->80
  </whiteLightBrightness>
  <irLightBrightness>
    <!--opt, int, 红外亮度配置(不分远近光灯时控制), range:[0,100]-->80
  </irLightBrightness>
  <BrightnessCtrlList>
    <!--opt, array, 远近光亮度控制参数, subType:object-->
    <BrightnessCtrl>
      <!--opt, object, 远近光亮度控制-->
      <lightType>
        <!--opt, enum, 补光灯类型, subType:string, [highWhite#远光白光补光灯,lowWhite#近光白光补光灯,highIrLight#远光红外补光灯,lowIrLight#近光红外补光灯]-->highWhite
      </lightType>
      <brightness>
        <!--opt, int, 补光灯亮度配置, range:[0,100]-->80
      </brightness>
      <BrightnessCtrl>
      </BrightnessCtrl>
    </BrightnessCtrlList>
  </EventIntelligenceModeCfg>
  <MultiAzimuthMixedModeCfg>
    <!--opt, object, 多方位混合模式参数配置, desc: 当supplementLightMode == multiAzimuthMixed生效-->
    <brightnessRegulatMode>
      <!--opt, enum, 补光灯亮度调节模式, subType:string, [manual#手动,auto#自动], desc: 手动模式下用户需手动配置各组补光灯的亮度 (LightCtrlList); 自动模式下用户无需配置亮度相关参数;由设备自动调节.-->auto
    </brightnessRegulatMode>
    <LightCtrlList>
      <!--opt, array, 补光灯组控制参数, subType:object-->
      <LightCtrl>
        <!--opt, object, 单个补光灯控制参数-->
        <azimuthType>
          <!--opt, enum, 补光灯所在的方位类型, subType:string, [left#左侧方位补光灯,right#右侧方位补光灯]-->left
        </azimuthType>
        <lightType>
          <!--opt, enum, 补光灯类型, subType:string, [highWhite#远光白光补光灯,midWhite#中光白光补光灯,lowWhite#近光白光补光灯,highIrLight#远光红外补光灯,midIrLight#中光红外补光灯,lowIrLight#近光红外补光灯]-->highWhite
        </lightType>
        <brightness>
          <!--opt, int, 补光灯亮度配置, range:[0,100]-->80
        </brightness>
        <LightCtrl>
        </LightCtrl>
      </LightCtrlList>
    </MultiAzimuthMixedModeCfg>
    <MultiAzimuthWhiteLightCfg>
      <!--opt, object, 多方位白光模式参数配置, desc: 当supplementLightMode == multiAzimuthWhiteLight生效-->
      <brightnessRegulatMode>
        <!--opt, enum, 补光灯亮度调节模式, subType:string, [manual#手动,auto#自动], desc: 手动模式下用户需手动配置各组补光灯的亮度 (LightCtrlList); 自动模式下用户无需配置亮度相关参数;由设备自动调节.-->auto
      </brightnessRegulatMode>
      <LightCtrlList>
        <!--opt, array, 补光灯组控制参数, subType:object-->
        <LightCtrl>
          <!--opt, object, 单个补光灯控制参数-->
          <azimuthType>
            <!--opt, enum, 补光灯所在的方位类型, subType:string, [left#左侧方位补光灯,right#右侧方位补光灯]-->left
          </azimuthType>
          <lightType>
            <!--opt, enum, 补光灯类型, subType:string, [highWhite#远光白光补光灯,midWhite#中光白光补光灯,lowWhite#近光白光补光灯,white#普通白光补光(不区分远中近光)]-->highWhite
          </lightType>
          <brightness>
            <!--opt, int, 补光灯亮度配置, range:[0,100]-->80
          </brightness>
```

```

        </brightness>
    </LightCtrl>
</LightCtrlList>
</MultiAzimuthWhiteLightCfg>
<MultiAzimuthIrLightCfg>
    <!--opt, object, 多方位红外模式参数配置, desc: 当supplementLightMode == multiAzimuthIrLight生效-->
    <brightnessRegulatMode>
        <!--opt, enum, 补光灯亮度调节模式, subType:string, [manual#手动,auto#自动], desc: 手动模式下用户需手动配置各组补光灯的亮度（LightCtrlList）；自动模式下用户无需配置亮度相关参数；由设备自动调节.-->auto
    </brightnessRegulatMode>
    <LightCtrlList>
        <!--opt, array, 补光灯组控制参数, subType:object-->
        <LightCtrl>
            <!--opt, object, 单个补光灯控制参数-->
            <azimuthType>
                <!--opt, enum, 补光灯所在的方位类型, subType:string, [left#左侧方位补光灯,right#右侧方位补光灯]-->left
            </azimuthType>
            <lightType>
                <!--opt, enum, 补光灯类型, subType:string, [highIrLight#远光红外补光灯,midIrLight#中光红外补光灯,lowIrLight#近光红外补光灯,irLight#普通红外补光灯(不区分远中近光)]-->highIrLight
            </lightType>
            <brightness>
                <!--opt, int, 补光灯亮度配置, range:[0,100]-->80
            </brightness>
        </LightCtrl>
    </LightCtrlList>
</MultiAzimuthIrLightCfg>
<PanoramicWhiteLightCfg>
    <!--opt, object, 全景白光模式参数配置, desc: 当supplementLightMode == panoramicWhiteLight生效-->
    <brightnessRegulatMode>
        <!--opt, enum, 补光灯亮度调节模式, subType:string, [manual#手动,auto#自动], desc: 手动模式下用户需手动配置各组补光灯的亮度（LightCtrlList）；自动模式下用户无需配置亮度相关参数；由设备自动调节.-->auto
    </brightnessRegulatMode>
    <LightCtrlList>
        <!--opt, array, 补光灯组控制参数, subType:object-->
        <LightCtrl>
            <!--opt, object, 单个补光灯控制参数-->
            <lightID>
                <!--opt, int, 补光灯序号, range:[1,4]-->1
            </lightID>
            <brightness>
                <!--opt, int, 补光灯亮度配置, range:[0,100]-->80
            </brightness>
        </LightCtrl>
    </LightCtrlList>
</PanoramicWhiteLightCfg>
<PanoramicIrLightCfg>
    <!--opt, object, 全景红外模式参数配置, desc: 当supplementLightMode == panoramicIrLightCfg生效-->
    <brightnessRegulatMode>
        <!--opt, enum, 补光灯亮度调节模式, subType:string, [manual#手动,auto#自动], desc: 手动模式下用户需手动配置各组补光灯的亮度（LightCtrlList）；自动模式下用户无需配置亮度相关参数；由设备自动调节.-->auto
    </brightnessRegulatMode>
    <LightCtrlList>
        <!--opt, array, 补光灯组控制参数, subType:object-->
        <LightCtrl>
            <!--opt, object, 单个补光灯控制参数-->
            <lightID>
                <!--opt, int, 补光灯序号, range:[1,4]-->1
            </lightID>
            <brightness>
                <!--opt, int, 补光灯亮度配置, range:[0,100]-->80
            </brightness>
        </LightCtrl>
    </LightCtrlList>
</PanoramicIrLightCfg>
<whiteLightEnhanceEnabled>
    <!--opt, bool, 白光灯亮度增强使能(不分远近光灯时控制), desc: 开启增强使能后, 白光灯会进入强电流模式, 亮度会增强十倍, 同时也可通过$.SupplementLight.whiteLightBrightness参数去控制强电流模式下的灯光亮度。该参数不区分补光灯模式及白光灯类型.-->>true
</whiteLightEnhanceEnabled>
<LightManualCtrlList>
    <!--opt, array, 补光灯组手动控制参数, subType:object, dep:and,{$.ImageChannelList[*].ImageChannel.SupplementLight.mode,eq>manual}, desc: 与supplementLightMode无关-->
    <LightCtrl>
        <!--opt, object, 单个补光灯控制参数-->
        <lightType>
            <!--opt, enum, 补光灯类型, subType:string, [highIrLight#远光红外补光灯,midIrLight#中光红外补光灯,lowIrLight#近光红外补光灯,irLight#普通红外补光灯(不区分远中近光),highWhite#远光白光补光灯,midWhite#中光白光补光灯,lowWhite#近光白光补光灯,white#普通红外补光灯(不区分远中近光)]-->highIrLight
        </lightType>
        <brightness>
            <!--opt, int, 补光灯亮度配置, range:[0,100]-->80
        </brightness>
    </LightCtrl>
</LightManualCtrlList>
</SupplementLight>
<OpticalDehaze>
    <!--opt, object, 光学雾透-->
    <enabled>
        <!--opt, bool, 使能-->true
    </enabled>
</OpticalDehaze>
<ManualRanging>
    <!--opt, object, 手动测距-->
    <enabled>

```

```
<!--opt, bool, 使能-->true
</enabled>
</ManualRanging>
<OIS>
  <!--opt, object, 光学防抖-->
  <mode>
    <!--req, enum, 模式, subType:string, [close#关闭,ordinary#普通,experts#专家]-->close
  </mode>
  <OISLevel>
    <!--opt, enum, 光学防抖等级, subType:string, [high#高,middle#中,low#低]-->high
  </OISLevel>
  <OISSensitivity>
    <!--opt, string, 光学防抖灵敏度-->test
  </OISSensitivity>
</OIS>
<isSupportLaserSpotManual>
  <!--opt, bool, 是否支持手动激光光点-->true
</isSupportLaserSpotManual>
<isSupportLaserSpotAdjustment>
  <!--opt, bool, 是否支持激光光点调整-->true
</isSupportLaserSpotAdjustment>
<DigitalZoom>
  <!--opt, object, 数字缩放-->
  <ZoomRatio>
    <!--opt, enum, 缩放比例, subType:string, [1x#1x,2x#2x,4x#4x,8x#8x,16x#16x,32x#32x]-->1x
  </ZoomRatio>
</DigitalZoom>
<Palettes>
  <!--opt, object, 热成像调色板配置-->
  <mode>
    <!--opt, enum, 模式, subType:string, [WhiteHot#白热,BlackHot#黑热,Fusion1#融合1,Rainbow#彩虹,Fusion2#融合2,Ironbow1#铁红1,Ironbow2#铁红2,Sepia#深褐色,Color1#色彩1,Color2#色彩2,IceFire#冰火,Rain#雨,RedHot#红热,GreenHot#绿热,DeepBlue#深蓝,Color3#色彩3]-->WhiteHot
  </mode>
  <ColorateTarget>
    <!--opt, object, 目标凸显, desc:dep mode = "WhiteHot"-->
    <ColorateTargetModelList>
      <!--opt, array, 目标凸显模式列表, subType:object-->
      <ColorateTargetMode>
        <!--opt, object, 目标凸显模式-->
        <id>
          <!--req, int, 序号, desc:从1开始-->0
        </id>
        <mode>
          <!--req, enum, 模式, subType:string, [colorateHotArea#大于(才凸显),colorateIntervalArea#区间(才凸显),colorateColdArea#小于(才凸显)]-->
        </mode>
        <enabled>
          <!--req, bool, 使能-->true
        </enabled>
        <TemperatureLimit>
          <!--req, object, 边界温度-->
          <minTemperature>
            <!--opt, float, 最低温度, range:[-20,150]-->0.000
          </minTemperature>
          <maxTemperature>
            <!--opt, float, 最高温度, range:[-20,150]-->0.000
          </maxTemperature>
        </TemperatureLimit>
        <Color>
          <!--req, object, 区域颜色-->
          <R>
            <!--req, int, R, range:[0,255]-->0
          </R>
          <G>
            <!--req, int, G, range:[0,255]-->0
          </G>
          <B>
            <!--req, int, B, range:[0,255]-->0
          </B>
        </Color>
      </ColorateTargetMode>
    </ColorateTargetModelList>
  </ColorateTarget>
</Palettes>
<ExposureSync>
  <!--opt, object, 镜头曝光同步-->
  <enabled>
    <!--req, bool, 使能-->true
  </enabled>
<GapList>
  <!--opt, object, 拼接缝列表-->
  <Gap>
    <!--opt, object, 拼接缝-->
    <gapID>
      <!--req, int, 拼接缝ID, range:[1,16]-->0
    </gapID>
    <RectList>
      <!--opt, object, 区域列表-->
      <Rect>
        <!--opt, object, 区域-->
        <rectID>
          <!--req, int, 区域ID, range:[1,16]-->0
        </rectID>
      </Rect>
    </RectList>
  </Gap>
</GapList>
```

```

<neqint>
  <!--req, int, 高度, range:[0,1000]-->0
</height>
<width>
  <!--req, int, 宽度, range:[0,1000]-->0
</width>
<x>
  <!--req, int, 左上角原点, range:[0,1000]-->0
</x>
<y>
  <!--req, int, 左上角原点, range:[0,1000]-->0
</y>
</Rect>
</RectList>
</Gap>
</GapList>
</ExposureSync>
<BrightnessSuddenChangeSuppressionCap>
  <!--opt, object, 亮度突变抑制能力-->
  <enabled>
    <!--req, bool, 使能-->true
  </enabled>
</BrightnessSuddenChangeSuppressionCap>
<AGCCap>
  <!--opt, object, AGC能力-->
  <AGCMode>
    <!--opt, enum, AGC模式, subType:string, [self_Adaptive#自适应,histogram#直方图,linear#线性]-->self_Adaptive
  </AGCMode>
</AGCCap>
<RegionalImageEnhancementCap>
  <!--opt, object, 区域图像增强能力-->
  <mode>
    <!--req, enum, 模式, subType:string,
[off#off,customArea#customArea,up_60%#up_60%,down_60%#down_60%,Left_60%#Left_60%,right_60%#right_60%,center_50%#center_50%,center_75%#center_75%]-->off
  </mode>
  <CustomAreaParam>
    <!--opt, object, 自定义区域参数, desc:dep <mode>="customArea"-->
  <RegionCoordinatesList>
    <!--opt, array, 区域坐标列表, subType:object, range:[3,10]-->
    <RegionCoordinates>
      <!--opt, object, 区域坐标, desc:屏幕左上角为坐标原点-->
      <positionX>
        <!--req, int, x坐标, range:[0,1000]-->0
      </positionX>
      <positionY>
        <!--req, int, y坐标, range:[0,1000]-->0
      </positionY>
    </RegionCoordinates>
  </RegionCoordinatesList>
</CustomAreaParam>
<DefaultAreaParamList>
  <!--opt, array, 默认区域参数列表, subType:object, desc:dep <mode>="up_60%,down_60%,Left_60%,right_60%,center_50%,center_75%"-->
  <DefaultAreaParam>
    <!--opt, object, 默认区域参数-->
    <mode>
      <!--req, enum, 模式, subType:string,
[up_60%#up_60%,down_60%#down_60%,Left_60%#Left_60%,right_60%#right_60%,center_50%#center_50%,center_75%#center_75%]-->up_60%
    </mode>
    <RegionCoordinatesList>
      <!--opt, array, 区域坐标列表, subType:object, range:[3,10]-->
      <RegionCoordinates>
        <!--opt, object, 区域坐标, desc:屏幕左上角为坐标原点-->
        <positionX>
          <!--req, int, x坐标, range:[0,1000]-->0
        </positionX>
        <positionY>
          <!--req, int, y坐标, range:[0,1000]-->0
        </positionY>
      </RegionCoordinates>
    </RegionCoordinatesList>
  </DefaultAreaParam>
</DefaultAreaParamList>
</RegionalImageEnhancementCap>
<faceExposure>
  <!--opt, object, 人脸曝光-->
  <faceExposureEnabled>
    <!--req, bool, 人脸曝光使能-->true
  </faceExposureEnabled>
  <faceExposureSensitivity>
    <!--req, int, 人脸曝光灵敏度, range:[0,100]-->0
  </faceExposureSensitivity>
</faceExposure>
<lightHal>
  <!--opt, object, 信号灯光晕抑制-->
  <enabled>
    <!--req, bool, 使能-->true
  </enabled>
  <halLevel>
    <!--req, int, 光晕抑制等级, range:[0,100]-->0
  </halLevel>
  <halRange>
    <!--req, int, 光晕抑制范围, range:[0,2064]-->0
  </halRange>
</lightHal>

```

```

<TargetEnhancementCap>
  <!--opt, object, 图像目标增强配置-->
  <enabled>
    <!--req, bool, 使能-->true
  </enabled>
</TargetEnhancementCap>
<LaserLight>
  <!--opt, object, 激光补光-->
  <mode>
    <!--opt, enum, 模式, subType:string, [auto#自动,manual#手动]-->auto
  </mode>
  <compensationRegionMode>
    <!--opt, enum, 激光补光区域, subType:string, [center#中心,adaptive#适应,full_Screen#全屏]-->center
  </compensationRegionMode>
  <compensationLevel>
    <!--opt, enum, 激光补光强度, subType:string, [Low#低,medium#中,high#高]-->low
  </compensationLevel>
  <LaserColor>
    <!--opt, object, 激光颜色, desc:依赖<mode>为manual-->
    <R>
      <!--req, int, R(红色)值, range:[0,100]-->0
    </R>
    <G>
      <!--req, int, G(绿色)值, range:[0,100]-->0
    </G>
    <B>
      <!--req, int, B(蓝色)值, range:[0,100]-->0
    </B>
  </LaserColor>
</LaserLight>
<ExternalSupplementLightCap>
  <!--opt, object, 外接补光灯参数配置能力-->
  <mode>
    <!--opt, enum, 模式, subType:string, [close#关闭,oneMixedLight#一组混合补光]-->close
  </mode>
  <BrightnessList>
    <!--opt, array, 亮度列表, subType:object-->
    <Brightness>
      <!--opt, object, 亮度-->
      <id>
        <!--opt, int, 每组补光灯的id-->0
      </id>
      <irLightBrightness>
        <!--opt, int, 红外灯亮度, range:[0,100]-->0
      </irLightBrightness>
      <whiteLightBrightness>
        <!--opt, int, 白光灯亮度, range:[0,100]-->0
      </whiteLightBrightness>
    </Brightness>
  </BrightnessList>
</ExternalSupplementLightCap>
<TempRange>
  <!--opt, object, 温度范围-->
  <mode>
    <!--req, enum, 温度范围模式, subType:string, [automatic#自动模式,manual#手动模式]-->automatic
  </mode>
  <temperatureUpperLimit>
    <!--opt, float, 上限温度, range:[-20,550], unit:C, unitType:温度, desc:精确到小数点后一位,手动温度范围模式下有效,上限温度需大于下限温度-->0.000
  </temperatureUpperLimit>
  <temperatureLowerLimit>
    <!--opt, float, 下限温度, range:[-20,550], unit:C, unitType:温度, desc:精确到小数点后一位,手动温度范围模式下有效,下限温度需小于上限温度-->0.000
  </temperatureLowerLimit>
</TempRange>
<BeautyCap>
  <!--opt, object, 美颜能力-->
  <enabled>
    <!--req, bool, 使能-->true
  </enabled>
  <whiteningStrength>
    <!--opt, int, 美白强度, range:[0,10]-->1
  </whiteningStrength>
  <skinSmoothingStrength>
    <!--opt, int, 磨皮强度, range:[0,10]-->1
  </skinSmoothingStrength>
</BeautyCap>
<MExposure>
  <!--opt, object, 手动曝光(工厂标定)-->
  <enabled>
    <!--opt, bool, 使能-->true
  </enabled>
  <exposureLevel>
    <!--opt, string, 曝光等级, range:[0,1000000]-->test
  </exposureLevel>
  <gainLevel>
    <!--opt, int, 增益等级, range:[0,100]-->0
  </gainLevel>
</MExposure>
<SensorRGBCtrl>
  <!--opt, object, SensorRGB控制-->
  <sensorID>
    <!--opt, int, SensorID, range:[0,1]-->0
  </sensorID>
</SensorRGBCtrl>

```

```
<YPbPrCalibration>
  <!--opt, object, 色差标定-->
  <mode>
    <!--opt, int, 模式, range:[0,2]-->1
  </mode>
</YPbPrCalibration>
<intrusionTriggeredLightingCfg>
  <!--opt, object, 随影补光-->
  <enabled>
    <!--req, bool, 使能-->true
  </enabled>
</intrusionTriggeredLightingCfg>
<shotProCfg>
  <!--opt, object, 幕影-->
  <shotProMode>
    <!--req, enum, 幕影模式, subType:string, [cLose#关闭,general#通用,enhancement#增强]-->general
  </shotProMode>
</shotProCfg>
</ImageChannel>
</ImageChannelList>
```

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 响应消息, attr:version{ro, string, 版本}-->
  <requestURL>
    <!--ro, req, string, 请求的URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, 状态码, subType:int, [0#OK,1#OK,2#Device Busy,3#Device Error,4#Invalid Operation,5#Invalid XML Format,6#Invalid XML Content,7#Reboot Required]-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, 状态信息, subType:string, [OK#成功,Device Busy#设备忙,Device Error#设备异常,Invalid Operation#无效的操作,Invalid XML Format#无效的XML格式,Invalid XML Content#无效的XML内容,Reboot#设备重启]-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, 详细错误码英文描述, desc:详细错误码的英文描述-->OK
  </subStatusCode>
</ResponseStatus>
```

69.24.12 获取通道图像参数能力

Request URL

GET /ISAPI/Image/channels/<channelID>/capabilities

查询参数

参数名称	参数类型	描述
channelID	string	视频通道号

请求报文

无

响应报文

```

<?xml version="1.0" encoding="UTF-8"?>
<ImageChannel xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 与ISAPI/Image/channels/<ID>/tempRange/capabilities中的xml内容一致, attr:version{req, string, 协议版本}-->
  <id>
    <!--ro, req, int, 序号ID-->1
  </id>
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <videoInputID>
    <!--ro, req, int, 视频通道号ID-->1
  </videoInputID>
  <ImageFlip>
    <!--ro, opt, object, 图像翻转-->
    <enabled opt="true,false">
      <!--ro, req, bool, 使能, attr:opt{req, string, 取值范围}-->true
    </enabled>
  </ImageFlip>
  <WDR>
    <!--ro, opt, object, 宽动态(自动)能力-->
    <mode opt="open,close,auto,SDR">
      <!--ro, req, enum, open, subType:string, [[open#开启,close#关闭,auto#自动,SDR#SDR], attr:opt{req, string, 取值范围, desc:SDR宽动态(super SDR)是指基于支持单帧曝光, 通过增益得到两帧图像输出, 并由平台进行合成的宽动态技术, 解决真宽动态运动模糊问题]-->[open
    </mode>
  </WDR>
  <Sharpness>
    <!--ro, opt, object, 锐度-->
    <SharpnessLevel min="0" max="100">
      <!--ro, req, int, 锐度级别, attr:min{req, int, 最小值},max{req, int, 最大值}-->1
    </SharpnessLevel>
  </Sharpness>
  <powerLineFrequency>
    <!--ro, opt, object, 电力线频率-->
    <powerLineFrequencyMode opt="50hz,60hz">
      <!--ro, opt, string, 模式, attr:opt{req, string, 取值范围}-->test
    </powerLineFrequencyMode>
  </powerLineFrequency>
  <Color>
    <!--ro, opt, object, 颜色-->
    <brightnessLevel min="0" max="10">
      <!--ro, opt, int, 亮度, attr:min{req, int, 最小值},max{req, int, 最大值}-->0
    </brightnessLevel>
    <contrastLevel min="0" max="10">
      <!--ro, opt, int, 对比度, attr:min{req, int, 最小值},max{req, int, 最大值}-->0
    </contrastLevel>
    <saturationLevel min="0" max="10">
      <!--ro, opt, int, 饱和度, attr:min{req, int, 最小值},max{req, int, 最大值}-->0
    </saturationLevel>
  </Color>
  <SupplementLight>
    <!--ro, opt, object, 内置补光灯管理参数能力-->
    <mode opt="schedule,off,on,auto,manual">
      <!--ro, opt, enum, 补光灯调节模式, subType:string, [schedule#白光定时,off#白光关闭,on#白光开启,auto#自动>manual#手动], attr:opt{req, string, 取值范围}-->auto
    </mode>
    <Schedule>
      <!--ro, opt, object, 白光灯定时生效时间配置;当<Mode>= schedule时生效;所配置时间为白天的时间-->
      <TimeRange>
        <!--ro, req, object, 白天时段;当天剩余时段为夜晚时段-->
        <beginTime>
          <!--ro, req, time, 开始时间-->00:08:00+08:00
        </beginTime>
        <endTime>
          <!--ro, req, time, 结束时间-->00:18:00+08:00
        </endTime>
      </TimeRange>
    </Schedule>
    <brightnessLimit min="0" max="100">
      <!--ro, opt, int, 白光补光灯亮度限制, attr:min{req, int, 最小值},max{req, int, 最大值}, desc:当<Mode>= on时;可以调节该限制;为白光补光灯亮度最大值;实际补光灯亮度为自动调节 (注: mode里面的自动为补光灯开光自动;自动调节模式需查看mixedLightBrightnessRegulatMode节点) -->80
    </brightnessLimit>
  </SupplementLight>
  <BeautyCap>
    <!--ro, opt, object, 美颜能力-->
    <enabled opt="true,false">
      <!--ro, req, bool, 使能, attr:opt{req, string, 取值范围}-->true
    </enabled>
    <whiteningStrength min="0" max="10">
      <!--ro, opt, int, 美白强度, range:[0,10], attr:min{req, int, 最小值},max{req, int, 最大值}-->1
    </whiteningStrength>
    <skinSmoothingStrength min="0" max="10">
      <!--ro, opt, int, 磨皮强度, range:[0,10], attr:min{req, int, 最小值},max{req, int, 最大值}-->1
    </skinSmoothingStrength>
  </BeautyCap>
</ImageChannel>

```

69.24.13 获取全部通道视频图像处理能力

Request URL

GET /ISAPI/Image/channels/capabilities

查询参数

无

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>
<ImageChannel xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 视频图像能力, attr:version{req, string, 协议版本}-->
  <id>
    <!--ro, req, int, 索引-->0
  </id>
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <videoInputID>
    <!--ro, req, int, 视频输入通道号-->0
  </videoInputID>
  <WDR>
    <!--ro, opt, object, 宽动态配置参数-->
  </WDR>
  <Sharpness>
    <!--ro, opt, object, 锐度-->
    <SharpnessLevel min="0" max="100">
      <!--ro, req, int, 锐度级别, attr:min{req, int, 最小值},max{req, int, 最大值}-->1
    </SharpnessLevel>
  </Sharpness>
  <powerLineFrequency>
    <!--ro, opt, object, 视频制式-->
    <powerLineFrequencyMode opt="50hz,60hz">
      <!--ro, opt, enum, 视频制式模式, subType:string, [50hz#50hz,60hz#60hz], attr:opt{req, string, 取值范围}-->50hz
    </powerLineFrequencyMode>
  </powerLineFrequency>
  <Color>
    <!--ro, opt, object, Color-->
    <brightnessLevel min="0" max="100">
      <!--ro, opt, int, 亮度, attr:min{req, int, 最小值},max{req, int, 最大值}-->24
    </brightnessLevel>
    <contrastLevel min="0" max="100">
      <!--ro, opt, int, 对比度, attr:min{req, int, 最小值},max{req, int, 最大值}-->22
    </contrastLevel>
    <saturationLevel min="0" max="100">
      <!--ro, opt, int, 饱和度, dep:and,{${Color.nightMode,eq,true}}, attr:min{req, int, 最小值},max{req, int, 最大值}-->33
    </saturationLevel>
  </Color>
  <SupplementLight>
    <!--ro, opt, object, 内置补光灯管理参数-->
    <mode opt="schedule,off,on,auto">
      <!--ro, opt, enum, 白光补光灯调节模式, subType:string, [schedule#定时,off#关闭,on#开启,auto#自动], attr:opt{req, string, 取值范围}-->auto
    </mode>
    <Schedule>
      <!--ro, opt, object, 白光灯定时生效时间配置;当<Mode>= schedule时生效;所配置时间为白天的时间-->
      <TimeRange>
        <!--ro, req, object, 白天时段;当天剩余时段为夜晚时段-->
        <beginTime>
          <!--ro, req, time, 开始时间-->00:08:00+08:00
        </beginTime>
        <endTime>
          <!--ro, req, time, 结束时间-->00:18:00+08:00
        </endTime>
      </TimeRange>
    </Schedule>
    <brightnessLimit min="0" max="100">
      <!--ro, opt, int, 白光补光灯亮度限制, attr:min{req, int, 最小值},max{req, int, 最大值}, desc:当<Mode>= on时;可以调节该限制;为白光补光灯亮度最大值;实际补光灯亮度为自动调节 (注: mode里面的自动为补光灯开光自动;自动调节模式需查看mixedLightBrightnessReguLatMode节点) -->80
    </brightnessLimit>
  </SupplementLight>
  <BeautyCap>
    <!--ro, opt, object, 美颜参数-->
    <enabled opt="true,false">
      <!--ro, req, bool, 使能, attr:opt{req, string, 取值范围}-->true
    </enabled>
    <whiteningStrength min="0" max="10">
      <!--ro, opt, int, 美白强度, range:[0,10], attr:min{req, int, 最小值},max{req, int, 最大值}-->0
    </whiteningStrength>
    <skinSmoothingStrength min="0" max="10">
      <!--ro, opt, int, 磨皮强度, range:[0,10], attr:min{req, int, 最小值},max{req, int, 最大值}-->0
    </skinSmoothingStrength>
  </BeautyCap>
</ImageChannel>
```

69.25 图像调整

69.25.1 获取双光融合参数能力

Request URL

GET /ISAPI/Image/dualLightFusion/capabilities?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "enabled": {
    /*ro, opt, object, 双光融合使能*/
    "@opt": [true, false]
    /*ro, req, array, 取值范围, subType:bool*/
  },
  "switchingThreshold": {
    /*ro, opt, object, 双光融合切换阈值*/
    "@min": 0,
    /*ro, req, int, 取值范围, range:[0,7]*/
    "@max": 7
    /*ro, req, int, 取值范围, range:[0,7], step:1*/
  },
  "channelID": {
    /*ro, opt, object, 视频通道编号, desc:支持该字段时, /ISAPI/Image/dualLightFusion?format=json&channelID=1,的channelID字段必填*/
    "@min": 1,
    /*ro, opt, int, 取值范围, range:[0,7]*/
    "@max": 8,
    /*ro, opt, int, 取值范围, range:[0,7], step:1*/
    "@opt": [1, 4]
    /*ro, opt, array, 枚举值, subType:int, desc:闸机可仅返回@opt, 表示仅支持1和4通道*/
  }
}
```

69.25.2 配置双光融合参数

Request URL

PUT /ISAPI/Image/dualLightFusion?format=json&channelID=<channelID>

查询参数

参数名称	参数类型	描述
channelID	string	当支持channelID时, 该字段必填

请求报文

```
{
  "enabled": true,
  /*opt, bool, 可见光与红外光融合使能, desc:低照度情况下, 融合可见光与红外光, 提升预览效果*/
  "switchingThreshold": 2
  /*opt, int, 双光融合切换阈值, range:[0,7], step:1*/
}
```

响应报文

```
{
  "statusCode": 1,
  /*ro, opt, int, 状态码, desc:无法用1表示时 (1表示成功且无特殊状态) 必须返回*/
  "statusString": "OK",
  /*ro, opt, string, 状态描述, range:[1,64], desc:无法用OK表示时 (OK表示成功且无特殊状态) 必须返回*/
  "subStatusCode": "ok",
  /*ro, opt, string, 子状态码, range:[1,64], desc:无法用ok表示 (ok表示成功且无特殊状态) 必须返回*/
  "errorCode": 1,
  /*ro, opt, int, 错误码, desc:当statusCode不为1时, 与subStatusCode对应*/
  "errorMsg": "ok"
  /*ro, opt, string, 错误信息, desc:当statusCode不为1时, 必须返回,解释信息在协议约束中, 允许设备在后续的版本迭代中, 进行优化丰富提升 (不限制死)*/
}
```

69.25.3 获取双光融合参数

Request URL

GET /ISAPI/Image/dualLightFusion?format=json&channelID=<channelID>

查询参数

参数名称	参数类型	描述
channelID	string	支持该字段时，该字段必填

请求报文

无

响应报文

```
{
  "enabled": true,
  /*ro, opt, bool, 可见光与红外光融合使能, desc:低照度情况下，融合可见光与红外光，提升预览效果*/
  "switchingThreshold": 2
  /*ro, opt, int, 双光融合切换阈值, range:[0,7], step:1*/
}
```

69.25.4 获取指定通道美颜参数

Request URL

GET /ISAPI/Image/channels/<channelID>/beauty

查询参数

参数名称	参数类型	描述
channelID	string	--

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<Beauty xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 美颜能力, attr:version{req, string, 协议版本}-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <whiteningStrength>
    <!--ro, opt, int, 美白强度, range:[0,10]-->1
  </whiteningStrength>
  <skinSmoothingStrength>
    <!--ro, opt, int, 磨皮强度, range:[0,10]-->1
  </skinSmoothingStrength>
</Beauty>
```

69.25.5 配置指定通道美颜参数

Request URL

PUT /ISAPI/Image/channels/<channelID>/beauty

查询参数

参数名称	参数类型	描述
channelID	string	--

请求报文

```
<?xml version="1.0" encoding="UTF-8"?>

<Beauty xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, 美颜能力, attr:version{req, string, 协议版本}-->
  <enabled>
    <!--req, bool, 使能-->true
  </enabled>
  <whiteningStrength>
    <!--opt, int, 美白强度, range:[0,10]-->1
  </whiteningStrength>
  <skinSmoothingStrength>
    <!--opt, int, 磨皮强度, range:[0,10]-->1
  </skinSmoothingStrength>
</Beauty>
```

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 响应消息, attr:version{ro, req, string, 版本}-->
  <requestURL>
    <!--ro, req, string, 请求的URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, 状态码, subType:int, [0#OK,1#OK,2#Device Busy,3#Device Error,4#Invalid Operation,5#Invalid XML Format,6#Invalid XML Content,7#Reboot Required]-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, 状态信息, subType:string, [OK#成功,Device Busy#设备忙,Device Error#设备异常,Invalid Operation#无效的操作,Invalid XML Format#无效的XML格式,Invalid XML Content#无效的XML内容,Reboot#设备重启]-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, 详细错误码英文描述, desc:详细错误码的英文描述-->OK
  </subStatusCode>
</ResponseStatus>
```

69.25.6 获取指定通道美颜参数能力

Request URL

GET /ISAPI/Image/channels/<channelID>/beauty/capabilities

查询参数

参数名称	参数类型	描述
channelID	string	--

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<BeautyCap xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 美颜能力, attr:version{req, string, 协议版本}-->
  <enabled opt="true,false">
    <!--ro, req, bool, 使能, attr:opt{req, string, 取值范围}-->true
  </enabled>
  <whiteningStrength min="0" max="10">
    <!--ro, opt, int, 美白强度, range:[0,10], attr:min{req, int, 最小值},max{req, int, 最大值}-->1
  </whiteningStrength>
  <skinSmoothingStrength min="0" max="10">
    <!--ro, opt, int, 磨皮强度, range:[0,10], attr:min{req, int, 最小值},max{req, int, 最大值}-->1
  </skinSmoothingStrength>
</BeautyCap>
```

69.26 图像补光

69.26.1 配置指定通道图像补光控制参数

Request URL

PUT /ISAPI/Image/channels/<channelID>/supplementLight

查询参数

参数名称	参数类型	描述
channelID	string	通道号

请求报文

```
<?xml version="1.0" encoding="UTF-8"?>

<SupplementLight xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, 内置补光灯管理参数, attr:version{req, string, 协议版本}-->
  <mode>
    <!--opt, enum, 补光灯调节模式, subType:string, [schedule#白光定时,off#白光关闭,on#白光开启,auto#自动>manual#手动]-->auto
  </mode>
  <Schedule>
    <!--opt, object, 白光灯定时生效时间配置, desc:当<Mode>= schedule或<mixedLightBrightnessRegulatMode>=schedule时生效;所配置时间为白天的时间-->
    <TimeRange>
      <!--req, object, 白天时段, desc:当天剩余时段为夜晚时段-->
      <beginTime>
        <!--req, time, 开始时间-->00:08:00+08:00
      </beginTime>
      <endTime>
        <!--req, time, 结束时间-->00:18:00+08:00
      </endTime>
    </TimeRange>
  </Schedule>
  <brightnessLimit>
    <!--opt, int, 白光补光灯亮度限制, desc:当<Mode>= on时;可以调节该限制;为白光补光灯亮度最大值;实际补光灯亮度为自动调节（注：mode里面的自动为补光灯开光自动;自动调节模式需查看mixedLightBrightnessRegulatMode节点）-->80
  </brightnessLimit>
  <supplementLightMode>
    <!--opt, enum, 补光灯模式, subType:string, [mixed#混合模式,whiteLight#白光模式,close#关闭,colorVuWhiteLight#全彩白光,irLight#红外模式,dualLight#双补光,eventIntelligence#事件智能补光,multiAzimuthMixed#多方位混合补光,multiAzimuthWhiteLight#多方位白光模式,multiAzimuthIrLight#多方位红外模式,panoramicWhiteLight#全景白光模式,panoramicIrLight#全景红外模式],
    desc:eventIntelligence为事件智能补光;解释如下:1. 在满足补光灯功能生效条件下;当没有产生报警;设备仅红外灯开启;2.产生报警后自动关闭红外灯;开启白光灯持续1min;在持续时间内如继续收到报警;这一状态保持至最后一个报警发生后1min结束。multiAzimuthMixed多方位混合模式可以通过左右边各一组补光灯（一组补光灯包含远中近白光红外补光灯）对目标进行多方位角度的补光;不同于之前mixed混合补光模式仅支持一组补光灯单方位进行补光。multiAzimuthWhiteLight多方位白光模式、multiAzimuthIrLight多方位红外模式与多方位混光情况相同
    panoramicWhiteLight、panoramicIrLight全景白光、红外模式当前仅四目全景相机支持, 当前下相机无方位的概念,由四个补光灯围绕相机部署。-->mixed
  </supplementLightMode>
  <irLightBrightness>
    <!--opt, int, 红外补光灯亮度, range:[0,100], desc:当<supplementLightMode>= mixed;且<mixedLightBrightnessRegulatMode>= manual时生效; 或当<supplementLightMode>= irLight;且<mixedLightBrightnessRegulatMode>= auto时生效-->80
  </irLightBrightness>
  <mixedLightBrightnessRegulatMode>
    <!--opt, enum, 补光灯亮度调节模式, subType:string, [manual#手动,auto#自动,schedule#白光定时,on#白光开启,off#白光关闭], desc:该字段在<supplementLightMode>= mixed或whiteLight或irLight、colorVuWhiteLight时生效; 手动模式下用户需手动<irLightBrightness>红外补光灯亮度参数以及<brightnessLimit>白光补光灯亮度; 自动模式下用户无需配置亮度相关参数;由设备自动调节。-->auto
  </mixedLightBrightnessRegulatMode>
  <highIrLightBrightness>
    <!--opt, int, 远光红外光亮度配置, range:[0,100], desc:参数由mixed#混合模式、whiteLight#白光模式、colorVuWhiteLight#全彩白光,irLight#红外模式、dualLight#双补光模式公用-->80
  </highIrLightBrightness>
  <highWhiteLightBrightness>
    <!--opt, int, 远光白光亮度配置, range:[0,100], desc:参数由mixed#混合模式、whiteLight#白光模式、colorVuWhiteLight#全彩白光,irLight#红外模式、dualLight#双补光模式公用-->80
  </highWhiteLightBrightness>
  <lowIrLightBrightness>
    <!--opt, int, 近光红外光亮度配置, range:[0,100], desc:参数由mixed#混合模式、whiteLight#白光模式、colorVuWhiteLight#全彩白光,irLight#红外模式、dualLight#双补光模式公用-->80
  </lowIrLightBrightness>
  <lowWhiteLightBrightness>
    <!--opt, int, 近光白光亮度配置, range:[0,100], desc:参数由mixed#混合模式、whiteLight#白光模式、colorVuWhiteLight#全彩白光,irLight#红外模式、dualLight#双补光模式公用-->80
  </lowWhiteLightBrightness>
  <whiteLightBrightness>
    <!--opt, int, 光亮度配置(不分远近光灯时控制), range:[0,100], desc:参数由mixed#混合模式、whiteLight#白光模式（且mixedLightBrightnessRegulatMode为manual、auto、schedule、on时有效）、colorVuWhiteLight#全彩白光,irLight#红外模式、dualLight#双补光模式、mode=auto 公用-->80
  </whiteLightBrightness>
  <WhiteLightModeCfg>
    <!--opt, object, 白光模式参数配置, desc:当supplementLightMode == whiteLight生效-->
    <highWhiteLightBrightness>
      <!--opt, int, 远光白光亮度配置, range:[0,100]-->80
    </highWhiteLightBrightness>
    <lowWhiteLightBrightness>
      <!--opt, int, 近光白光亮度配置, range:[0,100]-->80
    </lowWhiteLightBrightness>
    <whiteLightBrightness>
      <!--opt, int, 白光亮度配置(不分远近光灯时控制), range:[0,100]-->80
    </whiteLightBrightness>
  </WhiteLightModeCfg>
  <irLightbrightnessLimit>
    <!--opt, int, 红外灯亮度限制, range:[0,100], desc:当<Mode>== auto时且supplementLightMode==mixed模式下生效。-->80
  </irLightbrightnessLimit>
  <whiteLightbrightnessLimit>
    <!--opt, int, 白光灯亮度限制, range:[0,100], desc:当<Mode>== auto时且supplementLightMode==mixed模式下生效。-->80
  </whiteLightbrightnessLimit>
  <EventIntelligenceModeCfg>
    <!--opt, object, 事件智能补光模式参数配置, desc:当supplementLightMode == eventIntelligence生效-->
```

```
    <!--opt, enum, 补光灯亮度调节模式, subType:string, [manual#手动,auto#自动], desc: 手动模式下用户需手动配置补光灯亮度; 自动模式下用户无需配置亮度相关参数; 由设备自动调节.-->auto
  </brightnessRegulatMode>
  <brightness>
    <!--opt, int, 白光亮度配置(不分远近光灯时控制), range:[0,100]-->80
  </brightness>
  <irLightBrightness>
    <!--opt, int, 红外亮度配置(不分远近光灯时控制), range:[0,100]-->80
  </irLightBrightness>
  <BrightnessCtrlList>
    <!--opt, array, 远近光亮度控制参数, subType:object-->
    <BrightnessCtrl>
      <!--opt, object, 远近光亮度控制-->
      <lightType>
        <!--opt, enum, 补光灯类型, subType:string, [highWhite#远光白光补光灯,lowWhite#近光白光补光灯,highIrLight#远光红外补光灯,lowIrLight#近光红外补光灯]-->highWhite
      </lightType>
      <brightness>
        <!--opt, int, 补光灯亮度配置, range:[0,100]-->80
      </brightness>
    </BrightnessCtrl>
  </BrightnessCtrlList>
</EventIntelligenceModeCfg>
<MultiAzimuthMixedModeCfg>
  <!--opt, object, 多方位混合模式参数配置, desc: 当supplementLightMode == multiAzimuthMixed生效-->
  <brightnessRegulatMode>
    <!--opt, enum, 补光灯亮度调节模式, subType:string, [manual#手动,auto#自动], desc: 手动模式下用户需手动配置各组补光灯的亮度 (LightCtrlList); 自动模式下用户无需配置亮度相关参数; 由设备自动调节.-->auto
  </brightnessRegulatMode>
  <LightCtrlList>
    <!--opt, array, 补光灯组控制参数, subType:object-->
    <LightCtrl>
      <!--opt, object, 单个补光灯控制参数-->
      <azimuthType>
        <!--opt, enum, 补光灯所在的方位类型, subType:string, [left#左侧方位补光灯,right#右侧方位补光灯]-->left
      </azimuthType>
      <lightType>
        <!--opt, enum, 补光灯类型, subType:string, [highWhite#远光白光补光灯,midWhite#中光白光补光灯,lowWhite#近光白光补光灯,highIrLight#远光红外补光灯,midIrLight#中光红外补光灯,lowIrLight#近光红外补光灯]-->highWhite
      </lightType>
      <brightness>
        <!--opt, int, 补光灯亮度配置, range:[0,100]-->80
      </brightness>
    </LightCtrl>
  </LightCtrlList>
  <regulatSensitivityType>
    <!--opt, enum, 自动调节灵敏度类型, subType:string, [fast#快,medium#中,slow#慢], dep:and,
  </regulatSensitivityType>
  <MultiAzimuthMixedModeCfg>
  <MultiAzimuthWhiteLightCfg>
    <!--opt, object, 多方位白光模式参数配置, desc: 当supplementLightMode == multiAzimuthWhiteLight生效-->
    <brightnessRegulatMode>
      <!--opt, enum, 补光灯亮度调节模式, subType:string, [manual#手动,auto#自动], desc: 手动模式下用户需手动配置各组补光灯的亮度 (LightCtrlList); 自动模式下用户无需配置亮度相关参数; 由设备自动调节.-->auto
    </brightnessRegulatMode>
    <LightCtrlList>
      <!--opt, array, 补光灯组控制参数, subType:object-->
      <LightCtrl>
        <!--opt, object, 单个补光灯控制参数-->
        <azimuthType>
          <!--opt, enum, 补光灯所在的方位类型, subType:string, [left#左侧方位补光灯,right#右侧方位补光灯]-->left
        </azimuthType>
        <lightType>
          <!--opt, enum, 补光灯类型, subType:string, [highWhite#远光白光补光灯,midWhite#中光白光补光灯,lowWhite#近光白光补光灯,white#普通白光补光灯(不区分远中近光)]-->highWhite
        </lightType>
        <brightness>
          <!--opt, int, 补光灯亮度配置, range:[0,100]-->80
        </brightness>
      </LightCtrl>
    </LightCtrlList>
    <regulatSensitivityType>
      <!--opt, enum, 自动调节灵敏度类型, subType:string, [fast#快,medium#中,slow#慢], dep:and,
    </regulatSensitivityType>
    <MultiAzimuthWhiteLightCfg>
    <MultiAzimuthIrLightCfg>
      <!--opt, object, 多方位红外模式参数配置, desc: 当supplementLightMode == multiAzimuthIrLight生效-->
      <brightnessRegulatMode>
        <!--opt, enum, 补光灯亮度调节模式, subType:string, [manual#手动,auto#自动], desc: 手动模式下用户需手动配置各组补光灯的亮度 (LightCtrlList); 自动模式下用户无需配置亮度相关参数; 由设备自动调节.-->auto
      </brightnessRegulatMode>
      <LightCtrlList>
        <!--opt, array, 补光灯组控制参数, subType:object-->
        <LightCtrl>
          <!--opt, object, 单个补光灯控制参数-->
          <azimuthType>
            <!--opt, enum, 补光灯所在的方位类型, subType:string, [left#左侧方位补光灯,right#右侧方位补光灯]-->left
          </azimuthType>
          <lightType>
            <!--opt, enum, 补光灯类型, subType:string, [highIrLight#远光红外补光灯,midIrLight#中光红外补光灯,lowIrLight#近光红外补光灯,irLight#普通红外补光灯
```

```

(不区分远中近光)]-->highIrLight
    </lightType>
    <brightness>
        <!--opt, int, 补光灯亮度配置, range:[0,100]-->80
    </brightness>
    </LightCtrl>
</LightCtrlList>
</MultiAzimuthIrLightCfg>
<PanoramicWhiteLightCfg>
    <!--opt, object, 全景白光模式参数配置, desc:当supplementLightMode == panoramicWhiteLight生效-->
    <brightnessRegulatMode>
        <!--opt, enum, 补光灯亮度调节模式, subType:string, [manual#手动,auto#自动], desc:手动模式下用户需手动配置各组补光灯的亮度 (LightCtrlList); 自动模式下
        用户无需配置亮度相关参数;由设备自动调节.-->auto
    </brightnessRegulatMode>
    <LightCtrlList>
        <!--opt, array, 补光灯组控制参数, subType:object-->
        <LightCtrl>
            <!--opt, object, 单个补光灯控制参数-->
            <lightID>
                <!--opt, int, 补光灯序号, range:[1,4]-->1
            </lightID>
            <brightness>
                <!--opt, int, 补光灯亮度配置, range:[0,100]-->80
            </brightness>
        </LightCtrl>
    </LightCtrlList>
</PanoramicWhiteLightCfg>
<PanoramicIrLightCfg>
    <!--opt, object, 全景红外模式参数配置, desc:当supplementLightMode == PanoramicIrLightCfg生效-->
    <brightnessRegulatMode>
        <!--opt, enum, 补光灯亮度调节模式, subType:string, [manual#手动,auto#自动], desc:手动模式下用户需手动配置各组补光灯的亮度 (LightCtrlList); 自动模式下
        用户无需配置亮度相关参数;由设备自动调节.-->auto
    </brightnessRegulatMode>
    <LightCtrlList>
        <!--opt, array, 补光灯组控制参数, subType:object-->
        <LightCtrl>
            <!--opt, object, 单个补光灯控制参数-->
            <lightID>
                <!--opt, int, 补光灯序号, range:[1,4]-->1
            </lightID>
            <brightness>
                <!--opt, int, 补光灯亮度配置, range:[0,100]-->80
            </brightness>
        </LightCtrl>
    </LightCtrlList>
</PanoramicIrLightCfg>
<whiteLightEnhanceEnabled>
    <!--opt, bool, 白光亮度增强使能(不分远近光灯时控制), desc:手动开启增强使能后, 白光灯会进入强电流模式, 亮度会增强十倍, 同时也可通过
    $.SupplementLight.whiteLightBrightness参数去控制强电流模式下的灯光亮度。该参数不区分补光灯模式及白光灯类型-->true
</whiteLightEnhanceEnabled>
<LightManualCtrlList>
    <!--opt, array, 补光灯组手动控制参数, subType:object, dep:and,{$.SupplementLight.mode,eq,manual}, desc:与supplementLightMode无关-->
    <LightCtrl>
        <!--opt, object, 单个补光灯控制参数-->
        <lightType>
            <!--opt, enum, 补光灯类型, subType:string, [highIrLight#远光红外补光灯,midIrLight#中光红外补光灯,lowIrLight#近光红外补光灯,trLight#普通红外补光灯(不
            区分远中近光),highWhite#远光白光补光灯,midWhite#中光白光补光灯,lowWhite#近光白光补光灯,white#普通红外补光灯(不区分远中近光)]-->highIrLight
        </lightType>
        <lightID>
            <!--opt, int, 补光灯序号, range:[1,4]-->1
        </lightID>
        <enabled>
            <!--req, bool, 使能-->true
        </enabled>
        <brightness>
            <!--opt, int, 补光灯亮度配置, range:[0,100]-->80
        </brightness>
    </LightCtrl>
</LightManualCtrlList>
</SupplementLight>

```

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 响应消息, attr:version{ro, req, string, 版本}-->
  <requestURL>
    <!--ro, req, string, 请求的URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, 状态码, subType:int, [0#OK,1#OK,2#Device Busy,3#Device Error,4#Invalid Operation,5#Invalid XML Format,6#Invalid XML Content,7#Reboot Required]-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, 状态信息, subType:string, [OK#成功,Device Busy#设备忙,Device Error#设备异常,Invalid Operation#无效的操作,Invalid XML Format#无效的XML格式,Invalid XML Content#无效的XML内容,Reboot#设备重启]-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, 详细错误码英文描述, desc: 详细错误码的英文描述-->OK
  </subStatusCode>
</ResponseStatus>
```

69.26.2 获取指定通道图像补光控制参数

Request URL

GET /ISAPI/Image/channels/<channelID>/supplementLight

查询参数

参数名称	参数类型	描述
channelID	string	通道号

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<SupplementLight xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 内置补光灯管理参数, attr:version{req, string, 协议版本}-->
  <mode>
    <!--ro, opt, enum, 补光灯调节模式, subType:string, [schedule#白光定时,off#白光关闭,on#白光开启,auto#自动>manual#手动]-->auto
  </mode>
  <Schedule>
    <!--ro, opt, object, 白光灯定时生效时间配置, desc: 当<Mode>= schedule或mixedLightBrightnessRegulatMode=schedule时生效;所配置时间为白天的时间-->
    <TimeRange>
      <!--ro, req, object, 白天时段, desc: 当天剩余时段为夜晚时段-->
      <beginTime>
        <!--ro, req, time, 开始时间-->00:08:00+08:00
      </beginTime>
      <endTime>
        <!--ro, req, time, 结束时间-->00:18:00+08:00
      </endTime>
    </TimeRange>
  </Schedule>
  <brightnessLimit>
    <!--ro, opt, int, 白光补光灯亮度限制, desc: 当<Mode>= on时;可以调节该限制;为白光补光灯亮度最大值;实际补光灯亮度为自动调节（注: mode里面的自动为补光灯开关自动;自动调节模式需查看mixedLightBrightnessRegulatMode节点）-->80
  </brightnessLimit>
  <supplementLightMode>
    <!--ro, opt, enum, 补光灯模式, subType:string, [mixed#混合模式,whiteLight#白光模式,close#关闭,colorVuWhiteLight#全彩白光,irLight#红外模式,dualLight#双补光,eventIntelligence#事件智能补光,multiAzimuthMixed#多方位混合补光,multiAzimuthWhiteLight#多方位白光模式,multiAzimuthIrLight#多方位红外模式,panoramicWhiteLight#全景白光模式,panoramicIrLight#全景红外模式],
    desc:eventIntelligence为事件智能补光;解释如下:1.在满足补光灯功能生效条件下;当没有产生报警;设备仅红外灯开启;2.产生报警后自动关闭红外灯;开启白光灯持续1min;在持续时间内如继续收到报警;这一状态保持至最后一个报警发生后1min结束。multiAzimuthMixed多方位混合模式可以通过左右边各一组补光灯（一组补光灯包含远中近白光红外补光灯）对目标进行多方位角度的补光;不同于之前mixed混合补光模式仅支持一组补光灯单方位进行补光。multiAzimuthWhiteLight多方位白光模式、multiAzimuthIrLight多方位红外模式与多方位混光情况相同。
    panoramicWhiteLight、panoramicIrLight全景白光、红外模式当前仅四目全景相机支持,当前下相机无方位的概念,由四个补光灯围绕相机部署。-->mixed
  </supplementLightMode>
  <irLightBrightness>
    <!--ro, opt, int, 红外补光灯亮度, range:[0,100], desc: 当<supplementLightMode>= mixed;且<mixedLightBrightnessRegulatMode>= manual时生效;或当<supplementLightMode>= irLight;且<mixedLightBrightnessRegulatMode>= auto时生效-->80
  </irLightBrightness>
  <mixedLightBrightnessRegulatMode>
    <!--ro, opt, enum, 补光灯亮度调节模式, subType:string, [manual#手动,auto#自动,schedule#白光定时,on#白光开启,off#白光关闭], desc: 该字段在<supplementLightMode>= mixed或whiteLight或irLight、colorVuWhiteLight时生效;手动模式下用户需手动<irLightBrightness>红外补光灯亮度参数以及<brightnessLimit>白光补光灯亮度;自动模式下用户无需配置亮度相关参数;由设备自动调节。-->auto
  </mixedLightBrightnessRegulatMode>
  <highIrLightBrightness>
    <!--ro, opt, int, 远光红外光亮度配置, range:[0,100], desc: 参数由mixed#混合模式、whiteLight#白光模式、colorVuWhiteLight#全彩白光,irLight#红外模式、dualLight#双补光模式公用-->80
  </highIrLightBrightness>
  <highWhiteLightBrightness>
```

```
<!--ro, opt, int, 远光白光亮度配置, range:[0,100], desc:参数由mixed#混合模式、whiteLight#白光模式、colorVuWhiteLight#全彩白光,irLight#红外模式、
dualLight#双补光模式公用-->80
</highWhiteLightBrightness>
<!--ro, opt, int, 近光红外光亮度配置, range:[0,100], desc:参数由mixed#混合模式、whiteLight#白光模式、colorVuWhiteLight#全彩白光,irLight#红外模式、
dualLight#双补光模式公用-->80
</lowIrLightBrightness>
<!--ro, opt, int, 近光白光亮度配置, range:[0,100], desc:参数由mixed#混合模式、whiteLight#白光模式、colorVuWhiteLight#全彩白光,irLight#红外模式、
dualLight#双补光模式公用-->80
</lowWhiteLightBrightness>
<!--ro, opt, int, 光度配置(不分远近光灯时控制), range:[0,100], desc:参数由mixed#混合模式、whiteLight#白光模式（且mixedLightBrightnessRegulatMode为
manual、auto、schedule、on时有效）、colorVuWhiteLight#全彩白光,irLight#红外模式、dualLight#双补光模式、mode=auto 公用-->80
</whiteLightBrightness>
<!--ro, opt, object, 白光模式参数配置, desc:当supplementLightMode == whiteLight生效-->
<highWhiteLightBrightness>
<!--ro, opt, int, 远光白光亮度配置, range:[0,100]-->80
</highWhiteLightBrightness>
<lowWhiteLightBrightness>
<!--ro, opt, int, 近光白光亮度配置, range:[0,100]-->80
</lowWhiteLightBrightness>
<whiteLightBrightness>
<!--ro, opt, int, 白光亮度配置(不分远近光灯时控制), range:[0,100]-->80
</whiteLightBrightness>
</WhiteLightModeCfg>
<irLightbrightnessLimit>
<!--ro, opt, int, 红外灯亮度限制, range:[0,100], desc:当<Mode>== auto时且supplementLightMode==mixed模式下生效.-->80
</irLightbrightnessLimit>
<whiteLightbrightnessLimit>
<!--ro, opt, int, 白光灯亮度限制, range:[0,100], desc:当<Mode>== auto时且supplementLightMode==mixed模式下生效.-->80
</whiteLightbrightnessLimit>
<EventIntelligenceModeCfg>
<!--ro, opt, object, 事件智能补光模式参数配置, desc:当supplementLightMode == eventIntelligence生效-->
<brightnessRegulatMode>
<!--ro, opt, enum, 补光灯亮度调节模式, subType:string, [manual#手动,auto#自动], desc:手动模式下用户需手动<irLightBrightness>红外补光灯亮度参数以及
<whiteLightBrightness>白光补光灯亮度; 自动模式下用户无需配置亮度相关参数;由设备自动调节.-->auto
</brightnessRegulatMode>
<whiteLightBrightness>
<!--ro, opt, int, 白光亮度配置(不分远近光灯时控制), range:[0,100]-->80
</whiteLightBrightness>
<irLightBrightness>
<!--ro, opt, int, 红外亮度配置(不分远近光灯时控制), range:[0,100]-->80
</irLightBrightness>
<BrightnessCtrlList>
<!--ro, opt, array, 远近光亮度控制参数, subType:object-->
<BrightnessCtrl>
<!--ro, opt, object, 远近光亮度控制-->
<lightType>
<!--ro, opt, enum, 补光灯类型, subType:string, [highWhite#远光白光补光灯,lowWhite#近光白光补光灯,highIrLight#远光红外补光灯,lowIrLight#近光红外补
光灯]-->highWhite
</lightType>
<brightness>
<!--ro, opt, int, 补光灯亮度配置, range:[0,100]-->80
</brightness>
</BrightnessCtrl>
</BrightnessCtrlList>
</EventIntelligenceModeCfg>
<MultiAzimuthMixedModeCfg>
<!--ro, opt, object, 多方位混合模式参数配置, desc:当supplementLightMode == multiAzimuthMixed生效-->
<brightnessRegulatMode>
<!--ro, opt, enum, 补光灯亮度调节模式, subType:string, [manual#手动,auto#自动], desc:手动模式下用户需手动配置各组补光灯的亮度（LightCtrlList）; 自动模
式下用户无需配置亮度相关参数;由设备自动调节.-->auto
</brightnessRegulatMode>
<LightCtrlList>
<!--ro, opt, array, 补光灯组控制参数, subType:object-->
<LightCtrl>
<!--ro, opt, object, 单个补光灯控制参数-->
<azimuthType>
<!--ro, opt, enum, 补光灯所在的方位类型, subType:string, [Left#左侧方位补光灯,right#右侧方位补光灯]-->left
</azimuthType>
<lightType>
<!--ro, opt, enum, 补光灯类型, subType:string, [highWhite#远光白光补光灯,midWhite#中光白光补光灯,lowWhite#近光白光补光灯,highIrLight#远光红外补光
灯,midIrLight#中光红外补光灯,lowIrLight#近光红外补光灯]-->highWhite
</lightType>
<brightness>
<!--ro, opt, int, 补光灯亮度配置, range:[0,100]-->80
</brightness>
</LightCtrl>
</LightCtrlList>
<regulatSensitivityType>
<!--ro, opt, enum, 自动调节灵敏度类型, subType:string, [fast#快,medium#中,sLow#慢], dep:and,
{$.SupplementLight.MultiAzimuthMixedModeCfg.brightnessRegulatMode,eq,auto}-->fast
</regulatSensitivityType>
</MultiAzimuthMixedModeCfg>
<MultiAzimuthWhiteLightCfg>
<!--ro, opt, object, 多方位白光模式参数配置, desc:当supplementLightMode == multiAzimuthWhiteLight生效-->
<brightnessRegulatMode>
<!--ro, opt, enum, 补光灯亮度调节模式, subType:string, [manual#手动,auto#自动], desc:手动模式下用户需手动配置各组补光灯的亮度（LightCtrlList）; 自动模
式下用户无需配置亮度相关参数;由设备自动调节.-->auto
</brightnessRegulatMode>
<LightCtrlList>
<!--ro, opt, array, 补光灯组控制参数, subType:object-->
```

```

<!--ro, opt, array, 补光灯亮度调节参数, subType:object-->
<LightCtrl>
<!--ro, opt, object, 单个补光灯控制参数-->
<azimuthType>
<!--ro, opt, enum, 补光灯所在的方位类型, subType:string, [left#左侧方位补光灯,right#右侧方位补光灯]-->left
</azimuthType>
<lightType>
<!--ro, opt, enum, 补光灯类型, subType:string, [highWhite#远光白光补光灯,midWhite#中光白光补光灯,lowWhite#近光白光补光灯,white#普通白光补光(不区分远中近光)]-->highWhite
</lightType>
<brightness>
<!--ro, opt, int, 补光灯亮度配置, range:[0,100]-->80
</brightness>
</LightCtrl>
</LightCtrlList>
<regulatSensitivityType>
<!--ro, opt, enum, 自动调节灵敏度类型, subType:string, [fast#快,medium#中,sLow#慢], dep:and,
{$.SupplementLight.MultiAzimuthWhiteLightCfg.brightnessRegulatMode,eq,auto}-->fast
</regulatSensitivityType>
</MultiAzimuthWhiteLightCfg>
<MultiAzimuthIrLightCfg>
<!--ro, opt, object, 多方位红外模式参数配置, desc:当supplementLightMode == multiAzimuthIrLight生效-->
<brightnessRegulatMode>
<!--ro, opt, enum, 补光灯亮度调节模式, subType:string, [manual#手动,auto#自动], desc:手动模式下用户需手动配置各组补光灯的亮度（LightCtrlList）；自动模式下用户无需配置亮度相关参数;由设备自动调节.-->auto
</brightnessRegulatMode>
<LightCtrlList>
<!--ro, opt, array, 补光灯组控制参数, subType:object-->
<LightCtrl>
<!--ro, opt, object, 单个补光灯控制参数-->
<azimuthType>
<!--ro, opt, enum, 补光灯所在的方位类型, subType:string, [left#左侧方位补光灯,right#右侧方位补光灯]-->left
</azimuthType>
<lightType>
<!--ro, opt, enum, 补光灯类型, subType:string, [highIrLight#远光红外补光灯,midIrLight#中光红外补光灯,lowIrLight#近光红外补光灯,irLight#普通红外补光灯(不区分远中近光)]-->highIrLight
</lightType>
<brightness>
<!--ro, opt, int, 补光灯亮度配置, range:[0,100]-->80
</brightness>
</LightCtrl>
</LightCtrlList>
</MultiAzimuthIrLightCfg>
<PanoramicWhiteLightCfg>
<!--ro, opt, object, 全景白光模式参数配置, desc:当supplementLightMode == panoramicWhiteLight生效-->
<brightnessRegulatMode>
<!--ro, opt, enum, 补光灯亮度调节模式, subType:string, [manual#手动,auto#自动], desc:手动模式下用户需手动配置各组补光灯的亮度（LightCtrlList）；自动模式下用户无需配置亮度相关参数;由设备自动调节.-->auto
</brightnessRegulatMode>
<LightCtrlList>
<!--ro, opt, array, 补光灯组控制参数, subType:object-->
<LightCtrl>
<!--ro, opt, object, 单个补光灯控制参数-->
<lightID>
<!--ro, opt, int, 补光灯序号, range:[1,4]-->1
</lightID>
<brightness>
<!--ro, opt, int, 补光灯亮度配置, range:[0,100]-->80
</brightness>
</LightCtrl>
</LightCtrlList>
</PanoramicWhiteLightCfg>
<PanoramicIrLightCfg>
<!--ro, opt, object, 全景红外模式参数配置, desc:当supplementLightMode == PanoramicIrLightCfg生效-->
<brightnessRegulatMode>
<!--ro, opt, enum, 补光灯亮度调节模式, subType:string, [manual#手动,auto#自动], desc:手动模式下用户需手动配置各组补光灯的亮度（LightCtrlList）；自动模式下用户无需配置亮度相关参数;由设备自动调节.-->auto
</brightnessRegulatMode>
<LightCtrlList>
<!--ro, opt, array, 补光灯组控制参数, subType:object-->
<LightCtrl>
<!--ro, opt, object, 单个补光灯控制参数-->
<lightID>
<!--ro, opt, int, 补光灯序号, range:[1,4]-->1
</lightID>
<brightness>
<!--ro, opt, int, 补光灯亮度配置, range:[0,100]-->80
</brightness>
</LightCtrl>
</LightCtrlList>
</PanoramicIrLightCfg>
<whiteLightEnhanceEnabled>
<!--ro, opt, bool, 白光灯亮度增强使能(不分远近光灯时控制), desc:手动开启开启增强使能后, 白光灯会进入强电流模式, 亮度会增强十倍, 同时也可通过$.SupplementLight.whiteLightBrightness参数去控制强电流模式下的灯光亮度。该参数不区分补光灯模式及白光灯类型-->true
</whiteLightEnhanceEnabled>
<LightManualCtrlList>
<!--ro, opt, array, 补光灯组手动控制参数, subType:object, dep:and,{$.SupplementLight.mode,eq>manual}, desc:与supplementLightMode无关-->
<LightCtrl>
<!--ro, opt, object, 单个补光灯控制参数-->
<lightType>
<!--ro, opt, enum, 补光灯类型, subType:string, [highIrLight#远光红外补光灯,midIrLight#中光红外补光灯,lowIrLight#近光红外补光灯,irLight#普通红外补光灯(不区分远中近光),highWhite#远光白光补光灯,midWhite#中光白光补光灯,lowWhite#近光白光补光灯,white#普通红外补光灯(不区分远中近光)]-->highIrLight
</lightType>
<lightID>

```

```
<!--ro, opt, int, 补光灯序号, range:[1,4]-->1
</lightID>
<enabled>
<!--ro, req, bool, 使能-->true
</enabled>
<brightness>
<!--ro, opt, int, 补光灯亮度配置, range:[0,100]-->80
</brightness>
</LightCtrl>
</LightManualCtrlList>
</SupplementLight>
```

69.26.3 获取指定通道图像补光控制能力

Request URL

GET /ISAPI/Image/channels/<channelID>/supplementLight/capabilities

查询参数

参数名称	参数类型	描述
channelID	string	通道号

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>
<SupplementLight xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 内置补光灯管理参数, attr:version{req, string, 协议版本}-->
  <mode opt="schedule,off,on,auto,manual">
    <!--ro, opt, enum, 补光灯调节模式, subType:string, [schedule#白光定时,off#白光关闭,on#白光开启,auto#自动,manual#手动], attr:opt{req, string, 取值范围},
    desc:与supplementLightMode互斥-->auto
  </mode>
  <Schedule>
    <!--ro, opt, object, 白光灯定时生效时间配置, desc:当<Mode>= schedule或mixedLightBrightnessReguLatMode=schedule时生效;所配置时间为白天的时间-->
    <TimeRange>
      <!--ro, req, object, 白天时段, desc:当天剩余时段为夜晚时段-->
      <beginTime>
        <!--ro, req, time, 开始时间-->00:08:00+08:00
      </beginTime>
      <endTime>
        <!--ro, req, time, 结束时间-->00:18:00+08:00
      </endTime>
    </TimeRange>
  </Schedule>
  <brightnessLimit min="0" max="100">
    <!--ro, opt, int, 白光补光亮度限制, attr:min{req, int, 最小值},max{req, int, 最大值}, desc:当<Mode>= on时;可以调节该限制;为白光补光亮度最大值;实际补
    光亮度为自动调节 (注: mode里面的自动为补光灯开关自动;自动调节模式需查看mixedLightBrightnessReguLatMode节点) -->80
  </brightnessLimit>
</SupplementLight>
```

69.27 存储策略

69.27.1 获取当前存储模式配置能力

Request URL

GET /ISAPI/ContentMgmt/Storage/capabilities

查询参数

无

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>
<storage xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 存储信息, attr:version{req, string, 协议版本}-->
  <hddList size="10">
    <!--ro, opt, object, 硬盘信息列表, attr:size{req, int, 数组元素总数}-->
    <hdd>
```

```

<!--ro, opt, object, 硬盘信息-->
<id>
  <!--ro, req, string, 硬盘号-->test
</id>
<hddName>
  <!--ro, req, string, 硬盘名称-->test
</hddName>
<hddPath>
  <!--ro, opt, string, 硬盘路径-->test
</hddPath>
<hddType>
  <!--ro, req, enum, 硬盘类型, subType:string, [IDE#IDE,SATA#SATA,eSATA#eSATA,NFS#NFS,iSCSI#iSCSI,Virtual Disk#Virtual Disk]-->IDE
</hddType>
<status>
  <!--ro, req, enum, 硬盘状态, subType:string, [ok#正常,unformatted#未格式化,error#错误,idle#闲置,mismatch#不匹配,offline#网络硬盘未连接,smartFailed#smart检测失败,repairing#硬盘正在修复,formatting#硬盘正在格式化,notexist#本地硬盘不存在,unRecordHostFormatted#未在录播主机中格式化,maintaining#维护中 (CVR磐石323项目增加),synching#同步中,syncError#同步异常,unloaded#已卸载 (未加载),sizeMismatch#硬盘不匹配容量超限,unsupported#该硬盘设备不支持], desc:ok-正常,unformatted-未格式化,error-错误,idle-闲置,mismatch-不匹配,offline-网络硬盘未连接,smartFailed-smart检测失败,repairing-硬盘正在修复,formatting-硬盘正在格式化,notexist-本地硬盘不存在,unRecordHostFormatted-未在录播主机中格式化,maintaining-维护中, CVR磐石323项目增加,维护进度为maintainingPercent字段-->ok
</status>
<capacity>
  <!--ro, req, float, 容量, unit:MB, unitType:信息量-->50.000
</capacity>
<freeSpace>
  <!--ro, req, float, 剩余空间, unit:MB, unitType:信息量-->20.000
</freeSpace>
<property>
  <!--ro, req, enum, 硬盘属性, subType:string, [RW#RW,RO#RO,Redund#Redund]-->RW
</property>
<group>
  <!--ro, opt, string, 硬盘所在盘组号-->test
</group>
<dataModeList>
  <!--ro, opt, array, 数据模式列表, subType:object-->
  <dataMode>
    <!--ro, opt, object, 数据模式-->
    <type>
      <!--ro, req, enum, 应用方式类型, subType:string, [recordStorage#recordStorage,pictureCloudStorage#pictureCloudStorage,fileStorage#fileStorage]-->recordStorage
      </type>
      <occupancyRate>
        <!--ro, req, int, 磁盘占用率, range:[0,100], desc:表示对应应用方式占用了多少磁盘空间-->0
      </occupancyRate>
      <dataMode>
      </dataMode>
      <dataModeList>
      </dataModeList>
      <formatType>
        <!--ro, opt, enum, 磁盘文件类型, subType:string, [FAT32#FAT32,EXT4#EXT4], desc:FAT32,EXT4-->FAT32
      </formatType>
      <ipAddress>
        <!--ro, opt, string, IPV4, desc:硬盘所在集群中cs的IPV4地址-->172.6.64.7
      </ipAddress>
      <ipv6Address>
        <!--ro, opt, string, IPV6, desc:硬盘所在集群中cs的IPV6地址-->1080:0:0:0:8:800:200C:417A
      </ipv6Address>
      <encryptionStatus>
        <!--ro, opt, enum, 加密状态, subType:string, [unencrypted#未加密,encrypted#已加密,verifyFailed#校验失败], desc:unencrypted-未加密,encrypted-已加密,verifyFailed-校验失败-->unencrypted
      </encryptionStatus>
      <logStorageEnable>
        <!--ro, opt, bool, 当前磁盘是否是日志存储盘-->true
      </logStorageEnable>
      <installationTime>
        <!--ro, opt, time, 存储卡安装时间-->12:05:00+08:00
      </installationTime>
      <hddModel>
        <!--ro, opt, string, 硬盘型号-->test
      </hddModel>
      <hddSerialNumber>
        <!--ro, opt, string, 硬盘序列号-->test
      </hddSerialNumber>
      <HDDDataType opt="trainingData">
        <!--ro, opt, enum, 磁盘数据类型, subType:string, [trainingData#训练数据盘], attr:opt(req, string, 取值范围), desc:自学习超脑将HDD的第一块硬盘当成训练的数据盘 (算法训练任务的数据集、测试集等数据), 客户端只在算法训练的的硬盘管理界面显示此硬盘, 与普通HDD硬盘分开管理-->trainingData
      </HDDDataType>
      </hdd>
    </hddList>
  </nasList size="10">
  <!--ro, opt, object, nas信息列表, attr:size(req, int, 数组元素总数)-->
  <nas>
    <!--ro, opt, object, nas信息-->
    <id>
      <!--ro, req, string, 索引-->test
    </id>
    <addressingFormatType>
      <!--ro, req, enum, 地址类型, subType:string, [ipaddress#ip地址,hostname#主机名]-->ipaddress
    </addressingFormatType>
    <hostName>
      <!--ro, opt, string, 主机名, dep:and,{$.nasList.nas.addressingFormatType,eq,hostname}, desc:仅当addressingFormatType为hostname时存在-->test
    </hostName>
    <ipAddress>
      <!--ro, opt, string, ipv4地址, dep:and,{$.nasList.nas.addressingFormatType,eq,ipaddress}, desc:仅当addressingFormatType为ipaddress时存在-->192.168.230.1
    </ipAddress>
    <ipv6Address>
      <!--ro, opt, string, ipv6地址, dep:and,{$.nasList.nas.addressingFormatType,eq,ipv6address}, desc:仅当addressingFormatType为ipv6address时存在-->1080:0:0:0:8:800:200C:417A
    </ipv6Address>
  </nas>
</nasList>

```

```

<ipv6Address>
  <!--ro, opt, string, ipv6地址, dep:and,{$.nasList.nas.addressingFormatType,eq,ipaddress}, desc: 仅当addressingFormatType为ipaddress时存在-->
  <CD:910A:2222:5498:8475:1111:3900:2020
</ipv6Address>
<portNo>
  <!--ro, req, int, 端口号-->1
</portNo>
<userName>
  <!--ro, opt, string, 用户名-->test
</userName>
<password>
  <!--ro, opt, string, 密码-->test
</password>
<nasType>
  <!--ro, req, enum, nas盘类型, subType:string, [NFS#NFS,iSCSI#iSCSI,iSCSI @#iSCSI @#]-->NFS
</nasType>
<path>
  <!--ro, req, string, nas盘路径-->test
</path>
<status>
  <!--ro, req, enum, 当前状态, subType:string, [ok#正常,unformatted#未格式化,error#错误,idle#闲置,mismatch#不匹配,offline#网络硬盘未连接,smartFailed#smart检测失败,reparing#硬盘正在修复,formatting#硬盘正在格式化,notexist#本地硬盘不存在]-->ok
</status>
<capacity>
  <!--ro, req, float, 容量, desc:单位 MB-->0.000
</capacity>
<freeSpace>
  <!--ro, req, float, 剩余空间, desc:单位 MB-->0.000
</freeSpace>
<property>
  <!--ro, req, enum, 属性, subType:string, [RW#读写,RO#只读,redund#冗余,RDD#RDD]-->RW
</property>
<group>
  <!--ro, opt, string, 所属盘组号-->test
</group>
</nas>
</nasList>
<workMode opt="group,quota,extract">
  <!--ro, opt, enum, 工作模式, subType:string, [group#盘组,quota#配额,extract#抽帧,auto#自动], attr:opt{req, string, 取值范围}-->group
</workMode>
<supportDataReconstruction opt="all,record">
  <!--ro, opt, enum, 支持的数据重建类型, subType:string, [all#全部,record#录像], attr:opt{req, string, 取值范围}-->all
</supportDataReconstruction>
<isSupportSSDManagement>
  <!--ro, opt, bool, 是否支持SSD管理, desc:对应/ISAPI/ContentMgmt/storage/ssd/<ID>/capabilities?format=json-->true
</isSupportSSDManagement>
<ssdList size="10">
  <!--ro, opt, object, ssd信息列表, attr:size{req, int, 数组元素总数}-->
  <ssd>
    <!--ro, opt, object, ssd信息-->
    <id>
      <!--ro, req, string, 索引-->test
    </id>
    <ssdName>
      <!--ro, req, string, 固态硬盘名称-->test
    </ssdName>
    <version>
      <!--ro, opt, string, 固态硬盘固件版本-->test
    </version>
  </ssd>
</ssdList>
<minHddCapacity>
  <!--ro, opt, string, 设备支持的硬盘容量最小限制带单位, desc:eg."8GB"-->test
</minHddCapacity>
<maxHddCapacity>
  <!--ro, opt, string, 设备支持的硬盘容量最大限制带单位, desc:eg."16TB"-->test
</maxHddCapacity>
<isSupportCapacity>
  <!--ro, opt, bool, 是否支持总存储空间信息查询, desc:对应URL:/ISAPI/ContentMgmt/Storage/capacity?format=json-->true
</isSupportCapacity>
<isSupportpoolConfig>
  <!--ro, opt, bool, 支持资源池配置-->true
</isSupportpoolConfig>
<isSupportRAIDDisk>
  <!--ro, opt, bool, 是否支持所有磁盘信息配置, desc:/ISAPI/ContentMgmt/Storage/raid/RAIDDisk/capabilities?format=json-->true
</isSupportRAIDDisk>
</storage>

```

69.27.2 获取当前存储模式参数

Request URL

GET /ISAPI/ContentMgmt/Storage

查询参数

无

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<storage xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 存储信息, attr:version{req, string, 协议版本}-->
  <hddList size="10">
    <!--ro, opt, object, 硬盘信息列表, attr:size{req, int, 数组元素总数}-->
    <hdd>
      <!--ro, opt, object, 硬盘信息-->
      <id>
        <!--ro, req, string, 硬盘号-->test
      </id>
      <hddName>
        <!--ro, req, string, 硬盘名称-->test
      </hddName>
      <hddPath>
        <!--ro, opt, string, 硬盘路径-->test
      </hddPath>
      <hddType>
        <!--ro, req, string, 硬盘类型, desc:IDE,SATA,eSATA,NFS,iSCSI,Virtual Disk,NTFS-->test
      </hddType>
      <status>
        <!--ro, req, enum, 硬盘状态, subType:string, [ok#正常,unformatted#未格式化,error#错误,idle#闲置,mismatch#不匹配,offline#网络硬盘未连接,smartFailed#smart检测失败,reparing#硬盘正在修复,formatting#硬盘正在格式化,notexist#本地硬盘不存在,unRecordHostFormatted#未在录播主机中格式化,maintaining#维护中(CVR磐石323项目增加),synchronizing#同步中,syncError#同步异常,unloaded#已卸载(未加载)], desc:ok-正常,unformatted-未格式化,error-错误,idle-闲置,mismatch-不匹配,offline-网络硬盘未连接,smartFailed-smart检测失败,reparing-硬盘正在修复,formatting-硬盘正在格式化,notexist-本地硬盘不存在,unRecordHostFormatted-未在录播主机中格式化,maintaining-维护中, CVR磐石323项目增加,维护进度为maintainingPercent字段-->ok
      </status>
      <capacity>
        <!--ro, req, float, 容量, unit:MB, unitType:信息量-->50.000
      </capacity>
      <freeSpace>
        <!--ro, req, float, 剩余空间, unit:MB, unitType:信息量-->20.000
      </freeSpace>
      <property>
        <!--ro, req, string, 硬盘属性, desc:RW-读写,RO-只读,Redund-冗余-->test
      </property>
      <group>
        <!--ro, opt, string, 硬盘所在盘组号-->test
      </group>
      <DataModeList>
        <!--ro, opt, array, 数据模式列表, subType:object-->
        <DataMode>
          <!--ro, opt, object, 数据模式-->
          <type>
            <!--ro, req, string, 应用方式类型, desc:recordStorage-录像存储, pictureCloudStorage-图片云存储,fileStorage-文件存储,只能单选-->test
          </type>
          <occupancyRate>
            <!--ro, req, int, 磁盘占用率, range:[0,100], desc:表示对应应用方式占用了多少磁盘空间-->0
          </occupancyRate>
        </DataMode>
      </DataModeList>
      <formatType>
        <!--ro, opt, enum, 磁盘文件类型, subType:string, [FAT32#FAT32,EXT4#EXT4], desc:FAT32,EXT4-->FAT32
      </formatType>
      <ipAddress>
        <!--ro, opt, string, IPV4, desc:硬盘所在集群中cs的IPV4地址-->172.6.64.7
      </ipAddress>
      <ipV6Address>
        <!--ro, opt, string, IPV6, desc:硬盘所在集群中cs的IPV6地址-->1080:0:0:0:8:800:200C:417A
      </ipV6Address>
      <encryptionStatus>
        <!--ro, opt, enum, 加密状态, subType:string, [unencrypted#未加密,encrypted#已加密,verifyFailed#校验失败], desc:unencrypted-未加密,encrypted-已加密,verifyFailed-校验失败-->unencrypted
      </encryptionStatus>
      <logStorageEnable>
        <!--ro, opt, bool, 当前磁盘是否是日志存储盘-->true
      </logStorageEnable>
      <installationTime>
        <!--ro, opt, time, 存储卡安装时间-->12:05:00+08:00
      </installationTime>
      <hddModel>
        <!--ro, opt, string, 硬盘型号-->test
      </hddModel>
      <hddSerialNumber>
        <!--ro, opt, string, 硬盘序列号-->test
      </hddSerialNumber>
      <HDDDataType>
        <!--ro, opt, enum, 磁盘数据类型, subType:string, [trainingData#训练数据盘], desc:自学习超脑将HDD的第一块硬盘当成训练的数据盘(算法训练任务的数据集、测试集等数据), 客户端只在算法训练的的硬盘管理界面显示此硬盘, 与普通HDD硬盘分开管理-->trainingData
      </HDDDataType>
    </hdd>
  </hddList>
  <nasList size="10">
    <!--ro, opt, object, nas信息列表, attr:size{req, int, 数组元素总数}-->
    <nas>
      <!--ro, opt, object, nas信息-->
      <id>
        <!--ro, req, string, 索引-->test
      </id>
    </nas>
  </nasList>

```

```

</id>
<addressingFormatType>
  <!--ro, req, enum, 地址类型, subType:string, [ipaddress#ip地址,hostname#主机名]-->ipaddress
</addressingFormatType>
<hostName>
  <!--ro, opt, string, 主机名, dep:and,{$.nasList.nas.addressingFormatType,eq,hostname}, desc:仅当addressingFormatType为hostname时存在-->test
</hostName>
<ipAddress>
  <!--ro, opt, string, ipv4地址, dep:and,{$.nasList.nas.addressingFormatType,eq,ipaddress}, desc:仅当addressingFormatType为ipaddress时存在--
>192.168.230.1
</ipAddress>
<ipv6Address>
  <!--ro, opt, string, ipv6地址, dep:and,{$.nasList.nas.addressingFormatType,eq,ipaddress}, desc:仅当addressingFormatType为ipaddress时存在--
>CCD:910A:2222:5498:8475:1111:3900:2020
</ipv6Address>
<portNo>
  <!--ro, req, int, 端口号-->1
</portNo>
<userName>
  <!--ro, opt, string, 用户名-->test
</userName>
<nasType>
  <!--ro, req, enum, nas盘类型, subType:string, [NFS#NFS,iSCSI#iSCSI]-->NFS
</nasType>
<path>
  <!--ro, req, string, nas盘路径-->test
</path>
<status>
  <!--ro, req, enum, 当前状态, subType:string, [ok#正常,unformatted#未格式化,error#错误,idle#闲置,mismatch#不匹配,offline#网络硬盘未连接,smartFailed#smart检测失败,reparing#硬盘正在修复,formatting#硬盘正在格式化,notexist#本地硬盘不存在]-->ok
</status>
<capacity>
  <!--ro, req, float, 容量, desc:单位 MB-->0.000
</capacity>
<freeSpace>
  <!--ro, req, float, 剩余空间, desc:单位 MB-->0.000
</freeSpace>
<property>
  <!--ro, req, enum, 属性, subType:string, [RW#读写,RO#只读,redund#冗余]-->RW
</property>
<group>
  <!--ro, opt, string, 所属盘组号-->test
</group>
</nas>
</nasList>
<workMode opt="group,quota,extract">
  <!--ro, opt, enum, 工作模式, subType:string, [group#盘组,quota#配额,extract#抽帧,auto#自动], attr:opt{req, string, 取值范围}-->group
</workMode>
<ssdList size="10">
  <!--ro, opt, object, ssd信息列表, attr:size{req, int, 数组元素总数}-->
  <ssd>
    <!--ro, opt, object, ssd信息-->
    <id>
      <!--ro, req, string, 索引-->test
    </id>
    <ssdName>
      <!--ro, req, string, 固态硬盘名称-->test
    </ssdName>
    <version>
      <!--ro, opt, string, 固态硬盘固件版本-->test
    </version>
  </ssd>
</ssdList>
</storage>

```

69.27.3 配置当前存储模式参数

Request URL

PUT /ISAPI/ContentMgmt/Storage

查询参数

无

请求报文

```

<?xml version="1.0" encoding="UTF-8"?>

<storage xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, 存储信息, attr:version{req, string, 协议版本}-->
  <hddList size="10">
    <!--opt, object, 硬盘信息列表, attr:size{req, int, 数组元素总数}-->
    <hdd>
      <!--opt, object, 硬盘信息-->
      <id>
        <!--req, string, 硬盘号-->test
      </id>
      <hddName>

```

```
<!--name-->
<!--req, string, 硬盘名称-->test
</hddName>
<hddPath>
<!--opt, string, 硬盘路径-->test
</hddPath>
<hddType>
<!--req, string, 硬盘类型, desc:IDE,SATA,eSATA,NFS,iSCSI,Virtual Disk,NTFS-->test
</hddType>
<status>
<!--req, enum, 硬盘状态, subType:string, [ok#正常,unformatted#未格式化,error#错误,idle#闲置,mismatch#不匹配,offline#网络硬盘未连接,smartFailed#smart
检测失败,reparing#硬盘正在修复,formatting#硬盘正在格式化,notexist#本地硬盘不存在,unRecordHostFormatted#未在录播主机中格式化,maintaining#维护中 (CVR磐石323项
目增加),synching#同步中,syncError#同步异常,unloaded#已卸载 (未加载)], desc:ok-正常,unformatted-未格式化,error-错误,idle-闲置,mismatch-不匹配,offline-网络
硬盘未连接,smartFailed-smart检测失败,reparing-硬盘正在修复,formatting-硬盘正在格式化,notexist-本地硬盘不存在,unRecordHostFormatted-未在录播主机中格式
化,maintaining-维护中, CVR磐石323项目增加,维护进度为maintainingPercent字段-->ok
</status>
<capacity>
<!--req, float, 容量, unit:MB, unitType:信息量-->50.000
</capacity>
<freeSpace>
<!--req, float, 剩余空间, unit:MB, unitType:信息量-->20.000
</freeSpace>
<property>
<!--req, string, 硬盘属性, desc:RW-读写,R0-只读,Redund-冗余-->test
</property>
<group>
<!--opt, string, 硬盘所在盘组号-->test
</group>
<DataModelList>
<!--opt, array, 数据模式列表, subType:object-->
<DataMode>
<!--opt, object, 数据模式-->
<type>
<!--req, string, 应用方式类型, desc:recordStorage-录像存储, pictureCloudStorage-图片云存储,fileStorage-文件存储,只能单选-->test
</type>
<occupancyRate>
<!--req, int, 磁盘占用率, range:[0,100], desc:表示对应应用方式占用了多少磁盘空间-->0
</occupancyRate>
</DataMode>
</DataModelList>
<formatType>
<!--opt, enum, 磁盘文件类型, subType:string, [FAT32#FAT32,EXT4#EXT4], desc:FAT32,EXT4-->FAT32
</formatType>
<ipAddress>
<!--opt, string, IPV4, desc:硬盘所在集群中cs的IPV4地址-->172.6.64.7
</ipAddress>
<ipv6Address>
<!--opt, string, IPV6, desc:硬盘所在集群中cs的IPV6地址-->1080:0:0:0:8:800:200C:417A
</ipv6Address>
<encryptionStatus>
<!--opt, enum, 加密状态, subType:string, [unencrypted#未加密,encrypted#已加密,verifyFailed#校验失败], desc:unencrypted-未加密,encrypted-已加
密,verifyFailed-校验失败-->unencrypted
</encryptionStatus>
<logStorageEnable>
<!--opt, bool, 当前磁盘是否是日志存储盘-->true
</logStorageEnable>
<installationTime>
<!--opt, time, 存储卡安装时间-->12:05:00+08:00
</installationTime>
<hddModel>
<!--opt, string, 硬盘型号-->test
</hddModel>
<hddSerialNumber>
<!--opt, string, 硬盘序列号-->test
</hddSerialNumber>
</hdd>
</hddList>
<nasList size="10">
<!--opt, object, nas信息列表, attr:size(req, int, 数组元素总数)-->
<nas>
<!--opt, object, nas信息-->
<id>
<!--req, string, 索引-->test
</id>
<addressingFormatType>
<!--req, enum, 地址类型, subType:string, [ipaddress#ip地址,hostname#主机名]-->ipaddress
</addressingFormatType>
<hostName>
<!--opt, string, 主机名, dep:and,{$.nasList.nas.addressingFormatType,eq,hostname}, desc:仅当addressingFormatType为hostname时存在-->test
</hostName>
<ipAddress>
<!--opt, string, ipv4地址, dep:and,{$.nasList.nas.addressingFormatType,eq,ipaddress}, desc:仅当addressingFormatType为ipaddress时存在-->192.168.230.1
</ipAddress>
<ipv6Address>
<!--opt, string, ipv6地址, dep:and,{$.nasList.nas.addressingFormatType,eq,ipaddress}, desc:仅当addressingFormatType为ipaddress时存在--
>CDCC:910A:2222:5498:8475:1111:3900:2020
</ipv6Address>
<portNo>
<!--req, int, 端口号-->1
</portNo>
<userName>
<!--opt, string, 用户名-->test
</userName>
<password>
```

```

    <!--opt, string, 密码-->test
  </password>
  <nasType>
    <!--req, enum, nas盘类型, subType:string, [NFS#NFS,iSCSI#iSCSI]-->NFS
  </nasType>
  <path>
    <!--req, string, nas盘路径-->test
  </path>
  <status>
    <!--req, enum, 当前状态, subType:string, [ok#正常,unformatted#未格式化,error#错误,idle#闲置,mismatch#不匹配,offline#网络硬盘未连接,smartFailed#smart
检测失败,repairing#硬盘正在修复,formatting#硬盘正在格式化,notexist#本地硬盘不存在]-->ok
  </status>
  <capacity>
    <!--req, float, 容量, desc:单位 MB-->0.000
  </capacity>
  <freeSpace>
    <!--req, float, 剩余空间, desc:单位 MB-->0.000
  </freeSpace>
  <property>
    <!--req, enum, 属性, subType:string, [RW#读写,RO#只读,redund#冗余]-->RW
  </property>
  <group>
    <!--opt, string, 所属盘组号-->test
  </group>
</nas>
</nasList>
<workMode opt="group,quota,extract">
  <!--opt, enum, 工作模式, subType:string, [group#盘组,quota#配额,extract#抽帧,auto#自动], attr:opt{req, string, 取值范围}-->group
</workMode>
<ssdList size="10">
  <!--opt, object, ssd信息列表, attr:size{req, int, 数组元素总数}-->
  <ssd>
    <!--opt, object, ssd信息-->
    <id>
      <!--req, string, 索引-->test
    </id>
    <ssdName>
      <!--req, string, 固态硬盘名称-->test
    </ssdName>
    <version>
      <!--opt, string, 固态硬盘固件版本-->test
    </version>
  </ssd>
</ssdList>
</storage>

```

响应报文

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 响应消息, attr:version{ro, req, string, 版本}-->
  <requestURL>
    <!--ro, req, string, 请求的URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, 状态码, subType:int, [0#OK,1#OK,2#Device Busy,3#Device Error,4#Invalid Operation,5#Invalid XML Format,6#Invalid XML Content,7#Reboot
Required]-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, 状态信息, subType:string, [OK#成功,Device Busy#设备忙,Device Error#设备异常,Invalid Operation#无效的操作,Invalid XML Format#无效的XML
格式,Invalid XML Content#无效的XML内容,Reboot#设备重启]-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, 详细错误码英文描述, desc:详细错误码的英文描述-->OK
  </subStatusCode>
</ResponseStatus>

```

69.28 人脸库管理

69.28.1 获取人脸比对库全部配置

Request URL

GET /ISAPI/Intelligent/FDLib?security=<security>&iv=<iv>&identityKey=<identityKey>

查询参数

参数名称	参数类型	描述
security	string	没有security表示数据不加密； security=1,表示报文中的userName、password和loginPassword字段AES128加密； security=2,表示报文中的userName、password和loginPassword字段AES256加密；
iv	string	iv表示初始化向量，在security=1或者2时为必须参数;随机字符串，长度限制为32
identityKey	string	最大长度 64字符

请求报文

无

响应报文

湖南云舟电力科技有限公司
有效期：2024-08-15

<?xml version="1.0" encoding="UTF-8"?>

```
<FDLibBaseCfgList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, array, 人脸库信息列表, subType:object, attr:version{req, string, 协议版本}-->
  <FDLibBaseCfg>
    <!--ro, opt, object, 人脸库信息-->
    <id>
      <!--ro, opt, int, 表示List中子项个数;从1开始赋值;依次增加, range:[1,15]-->1
    </id>
    <FDID>
      <!--ro, opt, string, 表示人脸库唯一ID-->test
    </FDID>
    <name>
      <!--ro, req, string, 人脸库名称, range:[1,64]-->faceLib1
    </name>
    <thresholdValue>
      <!--ro, opt, int, 检测阈值, range:[0,100]-->50
    </thresholdValue>
    <customInfo>
      <!--ro, opt, string, 人脸库附加信息, range:[1,128]-->test
    </customInfo>
    <customFaceLibID>
      <!--ro, opt, string, 自定义人脸库ID, range:[1,64]-->test
    </customFaceLibID>
    <faceLibType>
      <!--ro, opt, enum, 人脸库类型, subType:string, [ordinary#普通库,private#私密库,stranger#陌生库,allowable#允许库,privacyMask#隐私遮蔽库]-->ordinary
    </faceLibType>
    <autoUpdate>
      <!--ro, opt, bool, 人脸库自动更新库中底图使能, desc:若能力集中autoUpdate只支持true,表示人脸库自动更新功能默认开启,不支持关闭-->true
    </autoUpdate>
    <qualityThreshold>
      <!--ro, opt, int, 图片质量阈值, range:[0,100]-->50
    </qualityThreshold>
    <importFDThreshold>
      <!--ro, opt, int, 入库相似度阈值, range:[0,100]-->40
    </importFDThreshold>
    <Age>
      <!--ro, opt, object, 人脸库聚类老化参数配置;当<autoUpdate>=true时生效-->
      <judgementCycleType>
        <!--ro, req, enum, 人脸库聚类数据老化判断周期时间类型, subType:string, [day#天,week#周,month#月,duration#按时长判断], desc:天:晚上0点,周:周日0点 月:月初0点,默认:month-->day
      </judgementCycleType>
      <statisticalCycleType>
        <!--ro, req, enum, 人脸库聚类比对成功统计周期时间类型能力, subType:string, [day#天,week#周,month#月]-->month
      </statisticalCycleType>
      <statisticalCycle>
        <!--ro, req, int, 比对成功的统计周期, range:[0,99999]-->1
      </statisticalCycle>
      <minFMCount>
        <!--ro, req, int, 统计周期内最少人脸比对成功次数, range:[0,99999]-->1
      </minFMCount>
      <FMCountEnabled>
        <!--ro, opt, bool, 是否启用人脸比对成功总次数-->true
      </FMCountEnabled>
      <FMCount>
        <!--ro, opt, int, 人脸比对成功总次数, range:[0,99999]-->1
      </FMCount>
      <backgroundPicThreshold>
        <!--ro, opt, int, 底图替换相似度阈值, range:[1,100]-->90
      </backgroundPicThreshold>
      <judgementCycle>
        <!--ro, opt, int, 人脸库聚类数据老化判断周期时间范围, range:[1,365], unit:d, unitType:时间, dep:and,
        {$.FDLibCap.CreateFDLibList[*].CreateFDLib.Age.judgementCycleType,eq,duration}-->30
      </judgementCycle>
    </Age>
    <FaceCounting>
      <!--ro, opt, object, 人脸库比对客流统计去重配置-->
      <enabled>
        <!--ro, req, bool, 人脸客流去重是否开启-->true
      </enabled>
      <uploadType>
        <!--ro, req, enum, 人脸客流去重开启后的上报模式, subType:string, [out_repetition#去重时间段外上报,noUpload#去重时间段外不上报]-->noUpload
      </uploadType>
    </FaceCounting>
    <faceMonitor>
      <!--ro, opt, bool, 是否是智能人脸布控的人脸库-->true
    </faceMonitor>
    <monitorFaceNum>
      <!--ro, opt, int, 布控人脸支持的人脸最大个数-->1
    </monitorFaceNum>
    <customNoConfig>
      <!--ro, opt, bool, 人员布控模式下, desc:返回为true-自定义库web不可配置,协议可以配置,web限制不可配置,不支持不返回-->true
    </customNoConfig>
    <uploadID>
      <!--ro, opt, string, 自学习人脸库人脸比对布防订阅上传策略ID;平台自己确保唯一性, range:[1,128], dep:and,
      {$.FDLibBaseCfgList.FDLibBaseCfg.autoUpdate,eq,true}, desc:autoUpdate=true时有效,因为自学习人脸库报警,对于不同的平台来说,报警结果是不一样的,因为A平台有这个人脸库,所以比对成功,对于B平台没有此人脸库,则应该属于比对失败,所以需要新增布防订阅上传策略ID, -->test
    </uploadID>
  </FDLibBaseCfg>
</FDLibBaseCfgList>
```

69.28.2 配置人脸比对库全部参数

Request URL

PUT /ISAPI/Intelligent/FDLib?security=<security>&iv=<iv>&identityKey=<identityKey>

查询参数

参数名称	参数类型	描述
security	string	没有security表示数据不加密； security=1,表示报文中的userName、password和loginPassword字段AES128加密； security=2,表示报文中的userName、password和loginPassword字段AES256加密；
iv	string	iv表示初始化向量，在security=1或者2时为必须参数;;随机字符串，长度限制为32
identityKey	string	最大长度 64字符

请求报文

湖南云舟电力科技有限公司
有效期：2024-08-15

<?xml version="1.0" encoding="UTF-8"?>

```
<FdlLibBaseCfgList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--opt, array, 人脸库信息列表, subType:object, attr:version{req, string, 协议版本}-->
  <FdlLibBaseCfg>
    <!--opt, object, 人脸库信息-->
    <id>
      <!--opt, int, 表示List中子项个数;从1开始赋值;依次增加, range:[1,15]-->1
    </id>
    <FDID>
      <!--opt, string, 表示人脸库唯一ID-->test
    </FDID>
    <name>
      <!--req, string, 人脸库名称, range:[1,64]-->faceLib1
    </name>
    <thresholdValue>
      <!--opt, int, 检测阈值, range:[0,100]-->50
    </thresholdValue>
    <customInfo>
      <!--opt, string, 人脸库附加信息, range:[1,128]-->test
    </customInfo>
    <customFaceLibID>
      <!--opt, string, 自定义人脸库ID, range:[1,64]-->test
    </customFaceLibID>
    <faceLibType>
      <!--ro, opt, enum, 人脸库类型, subType:string, [ordinary#普通库,private#私密库,stranger#陌生人库,allowable#允许库,privacyMask#隐私遮蔽库]-->allowable
    </faceLibType>
    <autoUpdate>
      <!--opt, bool, 人脸库自动更新库中底图使能, desc:若能力集中autoUpdate只支持true,表示人脸库自动更新功能默认开启, 不支持关闭-->true
    </autoUpdate>
    <qualityThreshold>
      <!--opt, int, 图片质量阈值, range:[0,100]-->50
    </qualityThreshold>
    <importFDThreshold>
      <!--opt, int, 入库相似度阈值, range:[0,100]-->40
    </importFDThreshold>
    <Age>
      <!--opt, object, 人脸库聚类老化参数配置;当<autoUpdate>=true时生效-->
      <judgementCycleType>
        <!--req, enum, 人脸库聚类数据老化判断周期时间类型, subType:string, [day#天,week#周,month#月,duration#按时长判断], desc:天:晚上0点,周:周日0点 月:月初0点,默认:month-->day
      </judgementCycleType>
      <statisticalCycleType>
        <!--ro, req, enum, 人脸库聚类比对成功统计周期时间类型能力, subType:string, [day#天,week#周,month#月]-->month
      </statisticalCycleType>
      <statisticalCycle>
        <!--req, int, 比对成功的统计周期, range:[0,99999]-->1
      </statisticalCycle>
      <minFMCCount>
        <!--req, int, 统计周期内最少人脸比对成功次数, range:[0,99999]-->1
      </minFMCCount>
      <FMCCountEnabled>
        <!--opt, bool, 是否启用人脸比对成功总次数-->true
      </FMCCountEnabled>
      <FMCCount>
        <!--opt, int, 人脸比对成功总次数, range:[0,99999]-->1
      </FMCCount>
      <backgroundPicThreshold>
        <!--opt, int, 底图替换相似度阈值, range:[1,100]-->90
      </backgroundPicThreshold>
      <judgementCycle>
        <!--opt, int, 人脸库聚类数据老化判断周期时间范围, range:[1,365], unit:d, unitType:时间, dep:and,
        {$.FdlLibCap.CreateFdlLibList[*].CreateFdlLib.Age.judgementCycleType,eq,duration}-->30
      </judgementCycle>
    </Age>
    <FaceCounting>
      <!--opt, object, 人脸库比对客流统计去重配置-->
      <enabled>
        <!--req, bool, 人脸客流去重是否开启-->true
      </enabled>
      <uploadType>
        <!--ro, req, enum, 人脸客流去重开启后的上报模式, subType:string, [out_repetition#去重时间段外上报,noUpload#去重时间段外不上报]-->noUpload
      </uploadType>
    </FaceCounting>
    <faceMonitor>
      <!--ro, opt, bool, 是否是智能人脸布控的人脸库-->true
    </faceMonitor>
    <monitorFaceNum>
      <!--ro, opt, int, 布控人脸支持的人脸最大个数-->1
    </monitorFaceNum>
    <customNoConfig>
      <!--ro, opt, bool, 人员布控模式下, desc:返回为true-自定义库web不可配置,协议可以配置,web限制不可配置,不支持不返回-->true
    </customNoConfig>
    <uploadID>
      <!--opt, string, 自学习人脸库人脸比对布防订阅上传策略ID;平台自己确保唯一性, range:[1,128], dep:and,
      {$.FdlLibBaseCfgList.FdlLibBaseCfg.autoUpdate,eq,true}, desc:autoUpdate=true时有效,因为自学习人脸库报警,对于不同的平台来说,报警结果是不一样的,因为A平台有这
      个人脸库, 所以比对成功,对于B平台没有此人脸库,则应该属于比对失败,所以需要新增布防订阅上传策略ID, -->test
    </uploadID>
  </FdlLibBaseCfg>
</FdlLibBaseCfgList>
```

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 响应消息, attr:version{ro, req, string, 版本}-->
  <requestURL>
    <!--ro, req, string, 请求的URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, 状态码, subType:int, [0#OK,1#OK,2#Device Busy,3#Device Error,4#Invalid Operation,5#Invalid XML Format,6#Invalid XML Content,7#Reboot Required]-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, 状态信息, subType:string, [OK#成功,Device Busy#设备忙,Device Error#设备异常,Invalid Operation#无效的操作,Invalid XML Format#无效的XML格式,Invalid XML Content#无效的XML内容,Reboot#设备重启]-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, 详细错误码英文描述, desc:详细错误码的英文描述-->OK
  </subStatusCode>
</ResponseStatus>
```

69.28.3 修改人脸库中的人脸数据

Request URL

PUT /ISAPI/Intelligent/FDLib/FDSearch?format=json&FDID=<FDID>&FPID=<FPID>&faceLibType=<faceLibType>

查询参数

参数名称	参数类型	描述
FDID	string	最大长度为63字节;人脸库ID. 创建人脸库: POST /ISAPI/Intelligent/FDLib?format=json 成功时返回
FPID	string	最大长度为63字节;人脸记录ID,添加人脸记录: POST /ISAPI/Intelligent/FDLib/FaceDataRecord?format=json成功时返回
faceLibType	string	人脸库类型: blackFD: 名单库,staticFD: 静态库

请求报文

```

{
  "faceURL": "test",
  /*opt, string, 人脸图片URL方式上传时输入的图片存储URL, range:[0,256]*/
  "name": "张三",
  /*req, string, 人脸图片对应的人员姓名, range:[0,96]*/
  "gender": "male",
  /*opt, enum, 人脸图片对应的人员性别, subType:string, [male#男,female#女,unknown#未知]*/
  "bornTime": "1970-01-01+08:00",
  /*req, date, 人脸图片对应的人员出生日期*/
  "city": "130100",
  /*opt, string, 人脸图片对应的人员出生城市编号(详见全国各省份城市列表)*/
  "certificateType": "officerID",
  /*opt, enum, 证件类型, subType:string, [officerID#军官证,ID#身份证]*/
  "certificateNumber": "test",
  /*opt, string, 证件号*/
  "caseInfo": "test",
  /*opt, string, 备注信息, range:[0,192], desc:dep if type == blackFD*/
  "tag": "aa,bb,cc,dd",
  /*opt, string, 自定义标签, range:[0,195], desc:单个标签最大48字节,(最多4个标签, 英文逗号隔开, 最大长度为195字节), dep if faceLibType == blackFD*/
  "address": "test",
  /*opt, string, 人员住址, range:[0,192], desc:dep if faceLibType == staticFD*/
  "customInfo": "test",
  /*opt, string, 自定义信息, range:[0,192], desc:dep if faceLibType == staticFD*/
  "modelData": "test",
  /*opt, string, 目标模型数据, desc:传输过程中针对二进制非建模数据进行base64的加密处理*/
  "rowKey": "test",
  /*opt, string, 人脸记录数据库主键, range:[0,64], desc:根据rowKey进行查询,能提高查找效率*/
  "transfer": true,
  /*opt, bool, 是否转存,
  desc:PBG对接,需要设备开启微视云安全认证,为了兼容原有功能,对部分协议进行变更,
  涉及到名单添加和名单修改2个接口,协议中增加是否转存的字段,不填默认不转存,如果是web操作,需要将此字段写成需要转存,智能应用服务器1.1.0增加*/
  "PicFeaturePoints": [
    /*opt, array, 图片特征点下发, subType:object,
    desc:若设备仅支持3种类型的特征点,平台下发3种以上特征点时,设备不能报错
    背景: 不同设备人脸评分算法不一致,导致平台评分通过的人脸下发至设备评分不通过, 无法建模
    解决方法: 由上层平台采用统一的人脸评分以及特征点提取算法计算出图片的特征点并下发至设备,有了特征点可以保证设备建模成功,兼容不同设备人脸评分算法的差异
    特征点下发方式: 特征点数据编码至人脸图片数据下发、特征点数据以json参数格式随人脸图片url下发
    设备获取特征点优先级从高到低: 1.图片内携带特征点 2.json格式特征点*/
    {
      "featurePointType": "face",
      /*req, enum, 特征点类型, subType:string, [face#人脸,leftEye#左眼,rightEye#右眼,leftMouthCorner#左嘴角,rightMouthCorner#右嘴角,nose#鼻子]*/
      "coordinatePoint": {
        /*opt, object, 区域, desc:坐标原点位于左上方*/
        "x": 1,
        /*req, int, x坐标, range:[0,1000], desc:坐标进行归一化处理,归至0-1000*/
        "y": 1,
        /*req, int, y坐标, range:[0,1000], desc:坐标进行归一化处理,归至0-1000*/
        "width": 1,
        /*opt, int, 宽度, range:[0,1000], desc:当featurePointType为face时必须填*/
        "height": 1
        /*opt, int, 高度, range:[0,1000], desc:当featurePointType为face时必须填*/
      }
    }
  ],
  "faceType": "normalFace",
  /*opt, enum, 人脸类型, subType:string, [normalFace#普通人脸(默认),patrolFace#巡更人脸,hijackFace#胁迫人脸,superFace#超级人脸]*/
  "saveFacePic": true,
  /*opt, bool, 是否保存人脸底图*/
  "leaderPermission": [1, 2, 3, 4]
  /*opt, array, 首次认证权限, subType:int, range:[1,4],
  desc:数组为空表示清空人员首次认证权限
  表示人员具有1,2,3,4门编号的首次认证权限*/
}

```

响应报文

```

{
  "statusCode": 1,
  /*ro, opt, int, 状态码, desc:无法用1表示时(1表示成功且无特殊状态)必须返回*/
  "statusString": "OK",
  /*ro, opt, string, 状态描述, range:[1,64], desc:无法用OK表示时(OK表示成功且无特殊状态)必须返回*/
  "subStatusCode": "ok",
  /*ro, opt, string, 子状态码, range:[1,64], desc:无法用ok表示(ok表示成功且无特殊状态)必须返回*/
  "errorCode": 1,
  /*ro, req, int, 错误码, desc:当statusCode不为1时, 与subStatusCode对应*/
  "errorMsg": "ok"
  /*ro, req, string, 错误信息, desc:当statusCode不为1时, 解释信息在协议约束中, 允许设备在后续的版本迭代中, 进行优化丰富提升(不限制死)*/
}

```

69.28.4 获取人脸库总能力

Request URL

GET /ISAPI/Intelligent/FDLib/capabilities

查询参数

无

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<FDLibCap xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 人脸库能力对象, attr:version{opt, string, xml版本}-->
  <CreateFDLibList size="15">
    <!--ro, req, array, 人脸库能力对象数组, subType:object, attr:size{opt, string, 人脸库能力对象数组大小}-->
    <CreateFDLib>
      <!--ro, req, object, 人脸库能力对象-->
      <id max="15" min="1">
        <!--ro, req, int, 人脸库个数范围能力, range:[1,15], attr:max{opt, string, 最大值},min{opt, string, 最小值}-->10
      </id>
      <name max="64" min="1">
        <!--ro, req, string, 人脸库名称长度能力, range:[1,64], attr:max{opt, string, 字符串长度最大值},min{opt, string, 字符串长度最小值}-->64
      </name>
      <thresholdValue max="100" min="0" def="60">
        <!--ro, opt, int, 人脸库检测阈值范围能力, range:[0,100], attr:max{opt, string, 最大值},min{opt, string, 最小值},def{req, int, 默认值}, desc: 阈值越大检测准确率越低-->50
      </thresholdValue>
      <customInfo max="128" min="1">
        <!--ro, opt, string, 人脸库自定义描述长度能力, range:[1,128], attr:max{opt, string, 字符串长度最大值},min{opt, string, 字符串长度最小值}-->128
      </customInfo>
      <faceDataMax max="150000" min="1">
        <!--ro, opt, int, 人脸库最大支持人脸底图张数能力, range:[1,150000], attr:max{opt, string, 最大值},min{opt, string, 最小值}-->30000
      </faceDataMax>
      <customFaceLibID max="64" min="1">
        <!--ro, opt, string, 人脸库自定义ID长度能力, range:[1,64], attr:max{opt, string, 字符串长度最大值},min{opt, string, 字符串长度最小值}-->64
      </customFaceLibID>
      <PersonInfoExtendList size="128">
        <!--ro, opt, array, 人脸底图扩展信息对象数组能力, subType:object, attr:size{opt, string, 数组大小}-->
        <PersonInfoExtend>
          <!--ro, opt, object, 人脸底图扩展信息对象能力-->
          <id max="15000" min="1">
            <!--ro, req, int, 扩展信息序号范围能力, range:[1,15000], attr:max{opt, string, 最大值},min{opt, string, 最小值}-->30000
          </id>
          <enable opt="true,false">
            <!--ro, req, bool, 扩展信息使能能力, attr:opt{opt, string, 可选值}-->true
          </enable>
          <name max="128" min="1">
            <!--ro, opt, string, 扩展信息名称长度能力, range:[1,128], attr:max{opt, string, 字符串长度最大值},min{opt, string, 字符串长度最小值}-->128
          </name>
          <value max="256" min="1">
            <!--ro, opt, string, 扩展信息内容长度能力, range:[1,256], attr:max{opt, string, 字符串长度最大值},min{opt, string, 字符串长度最小值}-->256
          </value>
        </PersonInfoExtend>
      </PersonInfoExtendList>
      <identityKey max="64" min="0">
        <!--ro, opt, string, 人脸私密库密钥描述长度, range:[1,64], attr:max{opt, string, 字符串长度最大值},min{opt, string, 字符串长度最小值}-->64
      </identityKey>
      <autoUpdate opt="true,false">
        <!--ro, opt, bool, 人脸库聚类自动更新底图数据使能能力, attr:opt{opt, string, 可选值}, desc: 自动更新只支持true时, 表示人脸库聚类自动更新功能默认开启, 不支持关闭. -->true
      </autoUpdate>
      <qualityThreshold max="100" min="0">
        <!--ro, opt, int, 人脸库聚类底图图片质量范围能力, range:[0,100], attr:max{opt, string, 最大值},min{opt, string, 最小值}, desc: 当<autoUpdate>=true时生效-->50
      </qualityThreshold>
      <importFDThreshold max="100" min="0">
        <!--ro, opt, int, 人脸库聚类自动入库相似度阈值范围能力, range:[0,100], attr:max{opt, string, 最大值},min{opt, string, 最小值}, desc: 当<autoUpdate>=true时生效-->40
      </importFDThreshold>
      <Age>
        <!--ro, opt, object, 人脸库聚类老化参数配置能力-->
        <judgementCycleType opt="day,week,month,duration">
          <!--ro, req, enum, 人脸库聚类数据老化判断周期时间类型能力, subType:string, [day#天,week#周,month#月,duration#时长], attr:opt{opt, string, 可选值}, desc: day、month、week: 支持按照天、周、月设置 (天: 23h/59m/59s, 周: 周日23h/59m/59s; 月: 每月1号 23h/59m/59s)。duration-按天数老化-->month
        </judgementCycleType>
        <statisticalCycleType opt="day,week,month">
          <!--ro, req, enum, 人脸库聚类比对成功统计周期时间类型能力, subType:string, [day#天,week#周,month#月], attr:opt{opt, string, 可选值}-->month
        </statisticalCycleType>
        <statisticalCycle max="99999" min="0">
          <!--ro, opt, int, 人脸库聚类比对成功统计周期时间范围能力, range:[0,99999], attr:max{opt, string, 最大值},min{opt, string, 最小值}, desc: 依赖于statisticalCycleType-->30
        </statisticalCycle>
        <minFMCount max="99999" min="0">
          <!--ro, req, int, 统计周期内最少人脸比对成功次数能力, range:[0,99999], attr:max{opt, string, 最大值},min{opt, string, 最小值}-->30
        </minFMCount>
        <FMCountEnabled opt="true,false">
          <!--ro, opt, bool, 人脸库聚类比对成功总次数限制使能能力, attr:opt{opt, string, 可选值}-->true
        </FMCountEnabled>
        <FMCount max="99999" min="0">
          <!--ro, opt, int, 人脸比对成功总次数范围能力, range:[0,99999], attr:max{opt, string, 最大值},min{opt, string, 最小值}-->100
        </FMCount>
      </Age>
    </CreateFDLib>
  </CreateFDLibList>
</FDLibCap>
```

```

</FMCount>
<backgroundPicThreshold min="1" max="100">
  <!--ro, opt, int, 底图替换相似度阈值, range:[1,100], attr:min{req, int, 最小值},max{req, int, 最大值}-->90
</backgroundPicThreshold>
<judgementCycle max="365" min="1">
  <!--ro, opt, int, 人脸库聚类数据老化判断周期时间范围, range:[1,365], unit:d, unitType:时间, dep:and,
{$.FDLibCap.CreateFDLibList[*]}.CreateFDLib.Age.judgementCycleType,eq,duration}, attr:max{opt, string, 最大值},min{opt, string, 最小值}-->30
</judgementCycle>
</Age>
<FaceCounting>
  <!--ro, req, object, 人脸库比对客流统计去重对象能力-->
  <enabled opt="true,false">
    <!--ro, req, bool, 人脸客流去重使能能力, attr:opt{opt, string, 可选值}-->true
  </enabled>
  <uploadType opt="out_repetition,noUpload">
    <!--ro, req, enum, 人脸客流去重开启后的上报模式能力, subType:string, [out_repetition#去重时间段外上报,noUpload#去重时间段外不上报], attr:opt{opt,
string, 取值范围}-->noUpload
  </uploadType>
</FaceCounting>
<strangerLibNum max="1" min="2">
  <!--ro, opt, int, 陌生人脸库数目范围能力, range:[1,2], attr:max{opt, string, 最大值},min{opt, string, 最小值}, desc:陌生人脸库数目,注: 可创建的人脸库
数目= CreateFDLibList节点的size属性值- strangerLibNum节点的max属性值-->1
</strangerLibNum>
<faceLibType opt="privacyMask">
  <!--ro, opt, enum, 创建指定人脸库类型, subType:string, [ordinary#普通库,privacyMask#隐私遮蔽库], attr:opt{req, string, 取值范围}, desc:默认支持创建
普通库-->ordinary
</faceLibType>
<userType opt="normal,visitor">
  <!--ro, opt, enum, 人员类型, subType:string, [normal#普通人,visitor#访客], attr:opt{req, string, 取值范围}-->normal
</userType>
<isGreenChannelPlate>
  <!--ro, opt, bool, 是否为绿色通道车牌相关人员, desc:默认非绿色通道,适用于安检人车核验一体机产品进行人车核验时使用-->true
</isGreenChannelPlate>
</CreateFDLib>
<CreateFDLibList>
<FaceAppendData>
  <!--ro, opt, object, 人脸库底图数据扩展对象能力-->
  <bornTime>
    <!--ro, opt, date, 人员出生日期-->1970-01-01+08:00
  </bornTime>
  <name max="64" min="1">
    <!--ro, opt, string, 人员姓名描述长度, range:[1,64], attr:max{opt, string, 最大值},min{opt, string, 最小值}-->64
  </name>
  <sex opt="male,female">
    <!--ro, opt, enum, 人员性别, subType:string, [male#男,female#女], attr:opt{opt, string, 取值范围}-->male
  </sex>
  <province>
    <!--ro, opt, string, 省份-->01
  </province>
  <city>
    <!--ro, opt, string, 城市-->01
  </city>
  <certificateType opt="officerID,ID,passportID,other">
    <!--ro, opt, enum, 证件类型能力, subType:string, [officerID#军官证,ID#身份证,passportID#护照,other#其他], attr:opt{opt, string, 取值范围}-->officerID
  </certificateType>
  <certificateNumber max="32" min="1">
    <!--ro, opt, string, 证件号长度能力, range:[1,128], attr:max{opt, string, 最大值},min{opt, string, 最小值}-->32
  </certificateNumber>
  <PersonInfoExtendList size="128">
    <!--ro, opt, array, 人脸底图扩展信息对象数组能力, subType:object, attr:size{opt, string, 数组元素总数}, desc:返回即表示支持对应
URI(/ISAPI/Intelligent/FDLib/pictureUpload)-->
    <PersonInfoExtend>
      <!--ro, opt, object, 人脸底图扩展信息对象能力-->
      <id max="15000" min="1">
        <!--ro, req, int, 扩展信息序号范围能力, range:[1,15000], attr:max{opt, string, 最大值},min{opt, string, 最小值}-->30000
      </id>
      <enable opt="true,false">
        <!--ro, req, bool, 扩展信息使能能力, attr:opt{opt, string, 取值范围}-->true
      </enable>
      <name max="128" min="1">
        <!--ro, opt, string, 扩展信息名称长度能力, range:[1,128], attr:max{opt, string, 最大值},min{opt, string, 最小值}-->128
      </name>
      <value max="256" min="1">
        <!--ro, opt, string, 扩展信息内容长度能力, range:[1,256], attr:max{opt, string, 最大值},min{opt, string, 最小值}-->256
      </value>
    </PersonInfoExtend>
  </PersonInfoExtendList>
  <isSupportPhoneNumber>
    <!--ro, opt, bool, 是否支持电话号码能力-->true
  </isSupportPhoneNumber>
  <isSupportImagesURLImport>
    <!--ro, opt, bool, 人脸库底图数据异步导入(通过URI)能力-->true
  </isSupportImagesURLImport>
  <customHumanID max="128" min="1">
    <!--ro, opt, string, 自定义人脸底图人员ID能力, range:[1,128], attr:max{opt, string, 最大值},min{opt, string, 最小值}-->64
  </customHumanID>
  <faceDescribe max="128" min="1">
    <!--ro, opt, string, 人脸底图描述信息能力, range:[1,128], attr:max{opt, string, 最大值},min{opt, string, 最小值}-->128
  </faceDescribe>
  <RegionCoordinatesList size="4">
    <!--ro, opt, array, 人脸目标, subType:object, attr:size{req, int, 数组元素总数}-->
    <RegionCoordinates>
      <!--ro, req, object, 人脸目标区域-->
      <positionX min="0" max="1000">

```

```

<!--ro, req, int, X坐标, attr:min{req, int, 最小值},max{req, int, 最大值}-->1
</positionX>
<positionY min="0" max="1000">
<!--ro, req, int, Y坐标, attr:min{req, int, 最小值},max{req, int, 最大值}-->1
</positionY>
</RegionCoordinates>
</RegionCoordinatesList>
<IMSIList size="5">
<!--ro, opt, array, IMSI列表, subType:object, attr:size{req, int, 数组元素总数}, desc:若人脸库类型为允许库, 必须绑定至少一个IMSI号码, 最多5个-->
<IMSI min="1" max="15">
<!--ro, opt, string, IMSI号码, range:[1,15], attr:min{req, int, 最小值},max{req, int, 最大值}-->123456789012345
</IMSI>
</IMSIList>
<plateNo min="0" max="32">
<!--ro, opt, string, 车牌号码, range:[0,32], attr:min{req, int, 最小值},max{req, int, 最大值}-->test
</plateNo>
</FaceAppendData>
<FDSearchDescription>
<!--ro, opt, object, 人脸库搜索对象能力-->
<searchID max="63" min="1">
<!--ro, opt, string, 搜索ID, attr:max{opt, string, 最大值},min{opt, string, 最小值},
desc:搜索记录唯一标识,用来确认上层客户端是否为同一个(倘若为同一个,则设备记录内存,下次搜索加快速度);
min/max按照设备实际能力返回,不返回默认:min=1 max=63-->63
</searchID>
<searchResultPosition>
<!--ro, opt, int, 搜索起始位置, desc:该节点未返回,确认<isSupportStandardSearch>存在并且为true,默认从0开始,未返回则默认从1开始-->1
</searchResultPosition>
<maxResults>
<!--ro, opt, int, 本次查询条数, desc:单次搜索范围为1-500条,如果超过设备实际支持的单次最大查询条数,设备默认返回单次最大查询条数结果;该节点未返回默认是
1-500条-->30
</maxResults>
<FDID>
<!--ro, opt, string, 人脸库ID-->1
</FDID>
<startTime>
<!--ro, opt, datetime, 人脸库人员出生开始时间-->1970-01-01T00:00:00+08:00
</startTime>
<endTime>
<!--ro, opt, datetime, 人脸库人员出生结束时间-->1970-01-01T00:00:00+08:00
</endTime>
<name max="64" min="1">
<!--ro, opt, string, 人员姓名, attr:max{opt, string, 最大值},min{opt, string, 最小值}-->64
</name>
<sex opt="male,female">
<!--ro, opt, enum, 人员性别, subType:string, [male#男,female#女], attr:opt{opt, string, 取值范围}-->male
</sex>
<province>
<!--ro, opt, string, 省份-->01
</province>
<city>
<!--ro, opt, string, 城市-->01
</city>
<certificateType opt="officerID,ID,passportID,other">
<!--ro, opt, enum, 证件类型能力, subType:string, [officerID#军官证,ID#身份证,passportID#护照,other#其他], attr:opt{opt, string, 取值范围}-->officerID
</certificateType>
<certificateNumber max="32" min="1">
<!--ro, opt, string, 证件号长度能力, range:[1,128], attr:max{opt, string, 最大值},min{opt, string, 最小值}-->32
</certificateNumber>
<FaceModeCap>
<!--ro, opt, object, 以图索图对象能力-->
<FaceMode>
<!--ro, opt, object, 人脸模型-->
<similarity max="100.0" min="0.0">
<!--ro, opt, float, 相似度, attr:max{opt, string, 最大值},min{opt, string, 最小值}-->50.0
</similarity>
</FaceMode>
<maxResultNum>
<!--ro, opt, int, 最大返回结果数目-->30
</maxResultNum>
</FaceModeCap>
<modelStatus opt="modeling,unmodeled,modelingFailed">
<!--ro, opt, enum, 人脸库底图建模状态, subType:string, [modeling#已建模,unmodeled#未建模,modelingFailed#建模失败], attr:opt{opt, string, 可选值}--
>modeling
</modelStatus>
<modelingStatus opt="success,failed,none">
<!--ro, opt, enum, 人脸库底图建模结果, subType:string, [success#建模成功,failed#建模失败,none#无], attr:opt{opt, string, 可选值}-->success
</modelingStatus>
<customFaceLibID max="64" min="0">
<!--ro, opt, string, 人脸库自定义ID长度能力, range:[1,64], attr:max{opt, string, 字符串长度最大值},min{opt, string, 字符串长度最小值}-->64
</customFaceLibID>
<identityKey max="64" min="0">
<!--ro, opt, string, 人脸库私密库密钥长度能力, attr:max{opt, string, 字符串长度最大值},min{opt, string, 字符串长度最小值}-->64
</identityKey>
<occurrences max="30" min="1">
<!--ro, opt, int, 人员频次能力范围, attr:max{opt, string, 最大值},min{opt, string, 最小值}-->30
</occurrences>
<occurrencesSearchType opt="greaterThanOrEqual,lessThan,equal">
<!--ro, opt, enum, 频次检索方式, subType:string, [greaterThanOrEqual#大于等于,lessThan#小于,equal#等于], attr:opt{opt, string, 可选值}, desc:当
occurrences 字段返回时,该字段才能返回-->greaterThanOrEqual
</occurrencesSearchType>
<faceScore min="-1" max="100">
<!--ro, opt, int, 人脸评分, range:[-1,100], attr:min{opt, string, 最小值},max{opt, string, 最大值}, desc:有效值[0-100],-1表示评分未知(未对该人脸评
分)-->-1

```

```
</faceScore>
<isSupportFDIDList>
  <!--ro, opt, array, 是否支持多库以图搜图, subType:object, desc:对应/ISAPI/Intelligent/FDLib/FDSearch中支持FDIDList-->
</isSupportFDIDList>
<FaceScoreRange>
  <!--ro, opt, object, 人脸评分范围-->
  <min min="0" max="100">
    <!--ro, req, int, 人脸评分范围最小值, attr:min{req, int, 最小值},max{req, int, 最大值}-->1
  </min>
  <max min="0" max="100">
    <!--ro, req, int, 人脸评分范围最大值, attr:min{req, int, 最小值},max{req, int, 最大值}-->1
  </max>
</FaceScoreRange>
<pictureStatus opt="normal,abnormal">
  <!--ro, opt, string, 人脸图片状态, attr:opt{req, string, 取值范围}, desc:正常人脸, 异常人脸 (建模失败+人脸评分为低+图片url下载失败的,此搜索结果是设备
根据当前算法识别效果提供的评判标准,异常图片可能影响人脸识别和比对的效果,如果为异常图片的,建议更换图片, 后期随着算法准确率的提升,此判断标准可能发生变化,因
此此判断标准结果仅供参考-->test
</pictureStatus>
<sortord opt="similarity,time,contrastSuccessTime,count">
  <!--ro, opt, enum, 排序主键, subType:string, [count#比对成功次数,contrastSuccessTime#比对成功时间,time#入库时间,similarity#相似度排序], attr:opt{req,
string, 取值范围}, desc:"similarity",相似度排序,能力是后期加的, 之前的设备可能不会返回该节点-->count
</sortord>
<order opt="DESC,ASC">
  <!--ro, opt, string, 搜索结果排序类型, attr:opt{req, string, 取值范围}, desc:DESC:顺序,最晚入库最先展示; ASC: 逆序,最早入库最先展示;默认:DESC-->test
</order>
<importStartTime>
  <!--ro, opt, datetime, 入库开始时间-->1970-01-01T00:00:00+08:00
</importStartTime>
<importEndTime>
  <!--ro, opt, datetime, 入库结束时间-->1970-01-01T00:00:00+08:00
</importEndTime>
<IMSIList size="5">
  <!--ro, opt, array, IMSI列表(输出), subType:object, attr:size{req, int, 数组元素总数}, desc:若人脸库类型为允许库, 必须绑定至少一个IMSI号码, 最多5个-->
<IMSIItem>
  <!--ro, opt, object, IMSI采集信息-->
  <IMSI min="1" max="15">
    <!--ro, opt, string, IMSI号码, range:[1,15], attr:min{req, int, 最小值},max{req, int, 最大值}-->123456789012345
  </IMSI>
  <count min="1">
    <!--ro, opt, int, 比中次数, attr:min{req, int, 最小值}-->10
  </count>
  <lastFMTIME>
    <!--ro, opt, datetime, 最近比对成功时间-->1970-01-01T00:00:00+08:00
  </lastFMTIME>
</IMSIItem>
</IMSIList>
<lastFMTIME>
  <!--ro, opt, datetime, 最近比对成功时间(输出)-->1970-01-01T00:00:00+08:00
</lastFMTIME>
<faceContrastSuccessStartTime>
  <!--ro, opt, datetime, 人脸比对成功开始时间, desc:支持确认某段时间内监控场景有哪些人出现过。配合使用的查询条件: 人脸库ID(FDID, 包括: 动态库、允许库、
静态库)、排序类型 (sortord)、排序顺序 (order) -->1970-01-01T00:00:00+08:00
</faceContrastSuccessStartTime>
<faceContrastSuccessEndTime>
  <!--ro, opt, datetime, 人脸比对成功结束时间, desc:支持确认某段时间内监控场景有哪些人出现过。配合使用的查询条件: 人脸库ID(FDID, 包括: 动态库、允许库、
静态库)、排序类型 (sortord)、排序顺序 (order) -->1970-01-01T00:00:00+08:00
</faceContrastSuccessEndTime>
<updateType opt="full,latest">
  <!--ro, opt, enum, 人脸信息更新类型, attr:opt{opt, string, 可选值}-->true
</updateType>
</FDSearchDescription>
<isSupportAnalysisFace>
  <!--ro, opt, bool, 是否支持分析人脸-->true
</isSupportAnalysisFace>
<isSupportFDLibImpEmp opt="true,false">
  <!--ro, opt, bool, 是否支持人脸库导入导出, attr:opt{opt, string, 可选值}-->true
</isSupportFDLibImpEmp>
<isSupportFCSearch opt="true,false">
  <!--ro, opt, bool, 是否支持人脸抓拍库检索, attr:opt{opt, string, 可选值}-->true
</isSupportFCSearch>
<FDLibType opt="faceContrast,blacklistFaceContrast,whitelistFaceContrast">
  <!--ro, opt, enum, 人脸比对类型, subType:string, [faceContrast#人脸比对,blacklistFaceContrast#黑名单人脸比对,whitelistFaceContrast#白名单人脸比对],
attr:opt{opt, string, 可选值}-->faceContrast
</FDLibType>
<isSupportFDLibEachImport>
  <!--ro, opt, bool, 人脸库相互导入底图数据-->true
</isSupportFDLibEachImport>
<FCSearchDescription>
  <!--ro, opt, object, 人脸抓拍库搜索对象能力-->
  <searchID max="63" min="1">
    <!--ro, opt, string, 搜索ID, attr:max{opt, string, 字符串长度最大值},min{opt, string, 字符串长度最小值}-->63
  </searchID>
  <searchResultPosition>
    <!--ro, opt, int, 搜索起始位置;该节点未返回-->1
  </searchResultPosition>
  <maxResults>
    <!--ro, opt, int, 本次查询条数-->30
  </maxResults>
  <snapStartTime>
    <!--ro, opt, datetime, ISO8601 time-->1970-01-01T00:00:00+08:00
  </snapStartTime>
  <snapEndTime>
    <!--ro, opt, datetime, ISO8601 time-->1970-01-01T00:00:00+08:00
  </snapEndTime>
```

```
</shapeName>
<faceMatchInfoEnable opt="true,false">
  <!--ro, opt, bool, 检索比对成功数据, attr:opt{opt, string, 可选项}-->true
</faceMatchInfoEnable>
<monitorName max="1" min="2">
  <!--ro, opt, string, 监控点名称, attr:max{opt, string, 字符串长度最大值},min{opt, string, 字符串长度最小值}-->1
</monitorName>
<similarity max="100.0" min="0.0">
  <!--ro, opt, float, 相似度, attr:max{opt, string, 最大值},min{opt, string, 最小值}-->50.0
</similarity>
<eventType opt="unlimit,faceContrast,whiteFaceContrast,faceCapture,faceContrastFailure">
  <!--ro, opt, enum, 人脸抓拍事件类型, subType:string, [unlimit#不限,faceContrast#人脸比对,whiteFaceContrast#白名单人脸比对（即陌生人报警）,faceCapture#人脸抓拍,faceContrastFailure#比对失败], attr:opt{opt, string, 取值范围}-->unlimit
</eventType>
<faceModeCap>
  <!--ro, opt, object, 以图索图对象能力-->
  <faceMode>
    <!--ro, opt, object, 人脸模型-->
    <similarity max="100.0" min="0.0">
      <!--ro, opt, float, 相似度, range:[0.0,100.0], attr:max{opt, string, 最大值},min{opt, string, 最小值}-->50.0
    </similarity>
  </faceMode>
  <maxResultNum>
    <!--ro, opt, int, 最大返回结果数目, desc:抓拍库的容量可能非常大,检索时间较长,对于远程检索时更加明显,为了提高检索性能,限制检索的样本数和检索的结果数-->30
  </maxResultNum>
</faceModeCap>
<pictureSearchCap>
  <!--ro, req, object, 支持按图片ID检索时返回-->
  <maxSearchPictureNum max="1" min="1">
    <!--ro, opt, int, 最大支持检索图片数目, attr:max{opt, string, 最大值},min{opt, string, 最小值}-->50
  </maxSearchPictureNum>
</pictureSearchCap>
<isSupportMultiChannelSearch>
  <!--ro, opt, bool, 是否支持多通道检索-->true
</isSupportMultiChannelSearch>
<isSupportTotalSearchResult>
  <!--ro, opt, bool, 是否支持设备单次检索条数限制-->true
</isSupportTotalSearchResult>
<thermometryUnit opt="celsius,fahrenheit,kelvin">
  <!--ro, opt, enum, 测温单位, subType:string, [celsius#摄氏度,fahrenheit#华氏度,kelvin#开尔文], attr:opt{req, string, 取值范围}-->celsius
</thermometryUnit>
<currTemperature min="-20.0" max="150.0℃">
  <!--ro, opt, float, 人脸温度, range:[-20.0,150.0], attr:min{req, int, 最小值},max{req, int, 最大值}, desc:与thermometryUnit测温单位配合使用-->-20.0
</currTemperature>
<temperatureCondition opt="normal,abnormal,all,above">
  <!--ro, opt, enum, 人脸测温温度状况, subType:string, [normal#正常,abnormal#异常,all#全部,above#温度高于], attr:opt{req, string, 取值范围}, desc:温度高于用于指定高于具体温度值搜索, 温度值参数temperatureValue-->normal
</temperatureCondition>
<temperatureValue>
  <!--ro, opt, float, 温度值, range:[-20,105], unit:℃, unitType:温度, dep:and,{$.FDLibCap.FCSearchDescription.temperatureCondition,eq,above}-->37.2
</temperatureValue>
<IMSIList size="5">
  <!--ro, opt, array, IMSI列表(输出), subType:object, attr:size{req, int, 数组元素总数}, desc:若人脸库类型为允许库, 必须绑定至少一个IMSI号码, 最多5个-->
  <IMSI min="1" max="15">
    <!--ro, opt, string, IMSI号码, range:[4,15], attr:min{req, int, 最小值},max{req, int, 最大值}-->123456789012345
  </IMSI>
</IMSIList>
</FCSearchDescription>
<isSupportFDSearchDataPackage>
  <!--ro, opt, bool, 是否支持人脸库搜索数据打包-->true
</isSupportFDSearchDataPackage>
<isSupportFCSearchDataPackage>
  <!--ro, opt, bool, 是否支持人脸检索结果打包-->true
</isSupportFCSearchDataPackage>
<maxFCSearchDataPackageNum>
  <!--ro, opt, int, 最大支持人脸检索结果打包数目-->50
</maxFCSearchDataPackageNum>
<isSupportFDExecuteControl>
  <!--ro, opt, bool, 是否支持人脸库布控关系-->true
</isSupportFDExecuteControl>
<isSupportManualModeling>
  <!--ro, opt, bool, 是否支持建模控制-->true
</isSupportManualModeling>
<isSupportSingleManualModeling>
  <!--ro, opt, bool, 是否支持单一建模-->true
</isSupportSingleManualModeling>
<isSupportRangeManualModeling>
  <!--ro, opt, bool, 是否支持整体建模-->true
</isSupportRangeManualModeling>
<isSupportSingleManualModelingSearch>
  <!--ro, opt, bool, 是否支持单一建模查询-->true
</isSupportSingleManualModelingSearch>
<isSupportRangeManualModelingSearch>
  <!--ro, opt, bool, 是否支持整体建模控制-->true
</isSupportRangeManualModelingSearch>
<isSupportModelingStatusSearch>
  <!--ro, opt, bool, 是否支持建模状态条件查询-->true
</isSupportModelingStatusSearch>
<Reboot>
  <!--ro, opt, object, 需要提示重启设备生效的具体参数内容-->
  <thresholdValue>
    <!--ro, opt, bool, 人脸库阈值修改后需要设备重启生效-->true
  </thresholdValue>
```

```

<delFaceLib>
  <!--ro, opt, bool, 删除人脸库后需要重启设备生效-->true
</delFaceLib>
</Reboot>
<isSupportCoverImportpictures>
  <!--ro, opt, bool, 表示是否支持图片覆盖式导入-->true
</isSupportCoverImportpictures>
<isSupportCoverImportFaceLib>
  <!--ro, opt, bool, 表示是否支持人脸库覆盖式导入-->true
</isSupportCoverImportFaceLib>
<isSupportConcurrentImportpictures>
  <!--ro, opt, bool, 表示是否支持图片并发式导入-->true
</isSupportConcurrentImportpictures>
<isSupportSurplusCapacity>
  <!--ro, opt, bool, 表示是否支持指定人脸库中剩余人脸图片张数查询-->true
</isSupportSurplusCapacity>
<isSupportFDlibProgress>
  <!--ro, opt, bool, 表示是否支持人脸库导入导出状态查询-->true
</isSupportFDlibProgress>
<isSupportFDlibPhotoWall>
  <!--ro, opt, bool, 表示是否支持4200业务显示照片能力-->true
</isSupportFDlibPhotoWall>
<isSupportFDCoverInfo>
  <!--ro, opt, bool, 表示是否支持获取指定人脸库导入会覆盖的人脸图片张数-->true
</isSupportFDCoverInfo>
<isSupportStandardSearch>
  <!--ro, opt, bool, 该节点用于给4200兼容前后协议实现不一致,
desc:当前前端实现和协议不一致,后续修改如下:
  responseStatus都返回为true,matchList和matchElement首字母都大写,和协议保持一致。
  包括人脸库搜索/ISAPI/Intelligent/FDlib/FDSearch,人脸比对搜索/ISAPI/Intelligent/FDlib/FCSearch,人脸模型状态搜索,searchResultPosition修改为从0开始--
>true
  </isSupportStandardSearch>
  <isSupportFaceDataExport>
    <!--ro, opt, bool, 表示支持人脸库中数据是否支持导出-->true
  </isSupportFaceDataExport>
  <isSupportNewlyPictureUpload>
    <!--ro, opt, bool, 是否支持导入图片到人脸库, desc:用于设备端兼容大图片导入-->true
  </isSupportNewlyPictureUpload>
  <isSupportFCSearchNormal>
    <!--ro, opt, bool, 是否支持人脸数据全部检索, desc:用于4200采用ISAPI协议搜索人脸结果,普通人脸全部检索,以前如果不带搜索条件,即默认搜索所有时,会走
NET_DVR_SmartSearchPicture接口获取图片-->true
  </isSupportFCSearchNormal>
  <FDLibEachImportCap>
    <!--ro, opt, object, 人脸库相互导入能力-->
  <isSupportImportLib>
    <!--ro, opt, bool, 是否支持整个库导入-->true
  </isSupportImportLib>
  <isSupportImportLibPicture>
    <!--ro, opt, bool, 是否支持人脸库指定图片导入-->true
  </isSupportImportLibPicture>
  </FDLibEachImportCap>
  <isSupportPrompt>
    <!--ro, opt, bool, 是否支持提示语自定义, desc:如果支持,对应支持URL(/ISAPI/Intelligent/FDlib/prompt/capabilities?format=json),再通过该URL可以判断具体支持的功能-->true
  </isSupportPrompt>
  <isSupportCustomHumanID>
    <!--ro, opt, bool, 表示是否支持集成客户自定义人员ID-->true
  </isSupportCustomHumanID>
  <isSupportCustomFaceLibID>
    <!--ro, opt, bool, 表示是否支持集成客户自定义人脸库ID-->true
  </isSupportCustomFaceLibID>
  <isSupportFCSearchAttendancePackage>
    <!--ro, opt, bool, 人脸比对签到文件导出-->true
  </isSupportFCSearchAttendancePackage>
  <isSupportFCSearchJsonFormat>
    <!--ro, opt, bool, 支持人脸比对报警信息查询json格式能力, desc:如果支持,对应支持URL(/ISAPI/Intelligent/FDlib/FCSearch/capabilities?format=json),再通过该
URL可以判断具体支持的功能-->true
  </isSupportFCSearchJsonFormat>
  <isSupportFCSearchDataPackageJsonFormat>
    <!--ro, opt, bool, 支持人脸比对报警信息打包json格式能力, desc:如果支持,对应支持URL(POST /ISAPI/Intelligent/FDlib/FCSearch/dataPackage?format=json)--
>true
  </isSupportFCSearchDataPackageJsonFormat>
  <isSupportFDCapacity>
    <!--ro, opt, bool, 表示是否支持获取人脸库容量-->true
  </isSupportFDCapacity>
  <SupportUploadPictureType opt="binary,url">
    <!--ro, opt, enum, 表示支持的导入图片类型, subType:string, [binary#二进制,url#url], attr:opt{opt, string, 取值范围}, desc:不返回,默认为二进制-->binary
  </SupportUploadPictureType>
  <isSupportNoSaveUploadPicture>
    <!--ro, opt, bool, 支持不保存上传原图的操作-->true
  </isSupportNoSaveUploadPicture>
  <isSupportUploadModeling>
    <!--ro, opt, bool, 支持直接上传图片建模数据-->true
  </isSupportUploadModeling>
  <SupportUploadPictureUrlType opt="ipAddress,domain">
    <!--ro, opt, enum, 表示支持的以URL方式导入图片到人脸库URL类型, subType:string, [ipAddress#ip地址,domain#域名], attr:opt{opt, string, 取值范围}, desc:不
返回,默认为ip地址方式-->ipAddress
  </SupportUploadPictureUrlType>
  <isSptFDlibBatchFileManager>
    <!--ro, opt, bool, 支持人脸数据文件批量管理-->true
  </isSptFDlibBatchFileManager>
  <importImageSize max="300" min="1">
    <!--ro, opt, int, 支持导入JPG/JPEG图片的大小, unit:KB, unitType:信息量, attr:max{opt, string, 最大值},min{opt, string, 最小值}-->30
  </importImageSize>

```

```

</import imageSize>
<isSupportUpdateHumanID>
  <!--ro, opt, bool, 是否支持更新自定义人员ID, desc:/ISAPI/Intelligent/FDLib/<FDID>/updateHumanId?format=json-->true
</isSupportUpdateHumanID>
<FDModelingStatusSearchDescription>
  <!--ro, opt, object, 任务ID-->
  <searchID max="63" min="1">
    <!--ro, opt, string, 搜索ID, attr:max{opt, string, searchID, range:[0,64], desc:无},min{opt, string, 最小值},
desc: 搜索记录唯一标识, 用来确认上层客户端是否为同一个(倘若为同一个, 则设备记录内存, 下次搜索加快速度);
min/max按照设备实际能力返回, 不返回默认:min=1 max=63-->searchIDtest
  </searchID>
  <searchResultPosition>
    <!--ro, opt, int, 搜索起始位置; 该节点未返回, desc: 该节点未返回, 确认<isSupportStandardSearch>存在并且为"true", 默认从0开始, 未返回则默认从1开始-->30
  </searchResultPosition>
  <maxResults>
    <!--ro, opt, int, 本次查询条数, desc: 单次搜索范围为1-500条, 如果超过设备实际支持的单次最大查询条数, 设备默认返回单次最大查询条数结果; 该节点未返回默认是
1-500条-->30
  </maxResults>
  <FDIDLList>
    <!--ro, req, array, 设备支持人脸库中建模数据的查询的人脸库ID, subType:object-->
  </FDIDLList>
  <taskID max="64" min="1">
    <!--ro, opt, string, 任务ID, attr:max{opt, string, 最大值},min{opt, string, 最小值}, desc: 设备优先实现
taskID, URL(/ISAPI/Intelligent/FDLib/manualModelingRangeTask 或者/ISAPI/Intelligent/FDLib/manualModeling?FDID=&&PID=), 该节点返回-->64
  </taskID>
  <FDModelingStatusSearchDescription>
  <StrangerLibFaceAppendData>
    <!--ro, opt, object, 陌生人库支持人脸属性信息-->
    <PersonInfoExtendList size="128">
      <!--ro, opt, array, 人脸底图扩展信息对象数组能力, subType:object, attr:size{opt, string, 数组元素总数}, desc: 返回即表示支持对应
URI(/ISAPI/Intelligent/FDLib/pictureUpload)-->
    <PersonInfoExtend>
      <!--ro, opt, object, 人脸底图扩展信息对象能力-->
      <id max="15000" min="1">
        <!--ro, req, int, 扩展信息序号范围能力, range:[1,15000], attr:max{opt, string, 最大值},min{opt, string, 最小值}-->30000
      </id>
      <enable opt="true,false">
        <!--ro, req, bool, 扩展信息使能能力, attr:opt{opt, string, 取值范围}-->true
      </enable>
      <name max="128" min="1">
        <!--ro, opt, string, 扩展信息名称长度能力, range:[1,128], attr:max{opt, string, 最大值},min{opt, string, 最小值}-->128
      </name>
      <value max="256" min="1">
        <!--ro, opt, string, 扩展信息内容长度能力, range:[1,256], attr:max{opt, string, 最大值},min{opt, string, 最小值}-->256
      </value>
    </PersonInfoExtend>
  </PersonInfoExtendList>
</StrangerLibFaceAppendData>
<StrangerLibFDSearchDescription>
  <!--ro, opt, object, 陌生人人脸库对象能力-->
  <FDID>
    <!--ro, opt, string, 人脸比对库唯一标识ID-->1
  </FDID>
  <PersonInfoExtendList size="128">
    <!--ro, opt, array, 人脸底图扩展信息对象数组能力, subType:object, attr:size{opt, string, 数组元素总数}, desc: 返回即表示支持对应
URI(/ISAPI/Intelligent/FDLib/pictureUpload)-->
    <PersonInfoExtend>
      <!--ro, opt, object, 人脸底图扩展信息对象能力-->
      <id max="15000" min="1">
        <!--ro, req, int, 扩展信息序号范围能力, range:[1,15000], attr:max{opt, string, 最大值},min{opt, string, 最小值}-->30000
      </id>
      <enable opt="true,false">
        <!--ro, req, bool, 扩展信息使能能力, attr:opt{opt, string, 取值范围}-->true
      </enable>
      <name max="128" min="1">
        <!--ro, opt, string, 扩展信息名称长度能力, range:[1,128], attr:max{opt, string, 最大值},min{opt, string, 最小值}-->128
      </name>
      <value max="256" min="1">
        <!--ro, opt, string, 扩展信息内容长度能力, range:[1,256], attr:max{opt, string, 最大值},min{opt, string, 最小值}-->256
      </value>
    </PersonInfoExtend>
  </PersonInfoExtendList>
  <occurrences max="30" min="1">
    <!--ro, opt, int, 人员频次能力范围, attr:max{opt, string, 最大值},min{opt, string, 最小值}-->30
  </occurrences>
  <occurrencesSearchType opt="greaterThanOrEqual,lessThan,equal">
    <!--ro, opt, enum, 频次检索方式, subType:string, [greaterThanOrEqual#大于等于,lessThan#小于,equal#等于], attr:opt{opt, string, 取值范围}, desc: 当
occurrences 字段返回时, 该字段才能返回-->greaterThanOrEqual
  </occurrencesSearchType>
</StrangerLibFDSearchDescription>
<isSupportOccurrencesData>
  <!--ro, opt, bool, 支持直接获取图片频次信息, desc: 对应协议/ISAPI/Intelligent/FDLib/<FDID>/picture/<PID>/occurrenceData?format=json-->true
</isSupportOccurrencesData>
<isSupportCopyFaceRecord>
  <!--ro, opt, bool, 是否支持人脸记录复制-->true
</isSupportCopyFaceRecord>
<isSupportConferenceSmartCheck>
  <!--ro, opt, bool, 是否支持人脸签到任务配置-->true
</isSupportConferenceSmartCheck>
<isSupportControlSmartCheck>
  <!--ro, opt, bool, 是否支持人脸签到任务控制-->true
</isSupportControlSmartCheck>
<isSupportSmartCheckStatus>
  <!--ro, opt, bool, 是否支持人脸签到状态获取-->true

```

```

</isSupportSmartCheckStatus>
<isSupportSmartClockIn>
  <!--ro, opt, bool, 是否支持人脸考勤任务配置-->true
</isSupportSmartClockIn>
<isSupportControlSmartClockIn>
  <!--ro, opt, bool, 是否支持人脸考勤任务控制-->true
</isSupportControlSmartClockIn>
<isSupportSmartClockInStatus>
  <!--ro, opt, bool, 是否支持人脸考勤状态获取-->true
</isSupportSmartClockInStatus>
<isSupportSmartClockInTemplate>
  <!--ro, opt, bool, 是否支持人脸考勤模板配置-->true
</isSupportSmartClockInTemplate>
<isSupportFaceLibFormat>
  <!--ro, opt, bool, 支持人脸库格式化, desc:对应URL(/ISAPI/Intelligent/FDLib/format?format=json)-->true
</isSupportFaceLibFormat>
<isSupportSurplusCapacityALL>
  <!--ro, opt, bool, 是否支持全部人脸库容量查询, desc:对应url为/ISAPI/Intelligent/FDLib/surplusCapacity?format=json-->true
</isSupportSurplusCapacityALL>
<importImageSize min="1" max="300">
  <!--ro, opt, int, 支持导入JPG/JPEG图片的大小, range:[10,100], attr:min{opt, string, 最小图片大小},max{opt, string, 最大图片大小}-->10
</importImageSize>
<AsyncImportCap>
  <!--ro, opt, object, 单个数据文件大小范围-->
  <singleFileMaxFaceNum max="100">
    <!--ro, opt, int, 单个文件的最大人脸数量, attr:max{opt, string, 单个文件的最大人脸数量}-->100
  </singleFileMaxFaceNum>
  <singleFileMaxSize max="100">
    <!--ro, opt, int, 单个数据文件大小范围, unit:KB, unitType:信息量, attr:max{opt, int, 单个数据文件大小范围}, desc:单位:kb-->100
  </singleFileMaxSize>
</AsyncImportCap>
<isSupportAsyncImportData>
  <!--ro, opt, bool, 是否支持人脸库异步导入数据, desc:对应URL (/ISAPI/Intelligent/FDLib/asyncImportDatas/capabilities?format=json) -->true
</isSupportAsyncImportData>
<isSupportTaskStatusSearch>
  <!--ro, opt, bool, 是否支持人脸库异步导入任务状态查询, desc:对应URL (/ISAPI/Intelligent/FDLib/asyncImportDatasTasks?format=json) -->true
</isSupportTaskStatusSearch>
<SyncDeletePicNum max="10">
  <!--ro, opt, int, 支持单次同步删除人脸图片数量, attr:max{opt, int, 单次同步删除图片最大数量}-->1
</SyncDeletePicNum>
<isSupportPICCertification>
  <!--ro, opt, bool, 是否支持云存储(非武汉云)URI鉴URI权限鉴权, desc:对应URL (/ISAPI/Intelligent/FDLib/PICCertification?format=json) -->true
</isSupportPICCertification>
<deleteOriginalPicEnabled opt="true,false">
  <!--ro, opt, bool, 导入后不保存原始图片使能, attr:opt{opt, string, 可选项}-->true
</deleteOriginalPicEnabled>
<isSupportFaceScore>
  <!--ro, opt, bool, 是否支持人脸评分详情查询, desc:对应接口/ISAPI/Intelligent/FDLib/<FDID>/picture/<PID>/faceScore?format=json-->true
</isSupportFaceScore>
<ImportDetails>
  <!--ro, opt, object, 人脸库整库导入能力, desc:对应URL /ISAPI/Intelligent/FDLib/<FDID>/importDetails-->
  <checkCode min="0" max="16">
    <!--ro, req, string, 文件头加密密码(只存在第一包数据中), attr:min{req, int, 最小值},max{req, int, 最大值}-->1
  </checkCode>
  <concurrentRequest>
    <!--ro, opt, int, 并发请求数-->1
  </concurrentRequest>
  <chunkSize min="1024" max="2048">
    <!--ro, opt, int, 每个分包的大小, unit:KB, unitType:信息量, attr:min{req, int, 最小值},max{req, int, 最大值, unit:KB, unitType:信息量}-->1
  </chunkSize>
</ImportDetails>
<ExportDetails>
  <!--ro, opt, object, 人脸库整库导出能力, desc:对应URL /ISAPI/Intelligent/FDLib/<FDID>/exportDetails-->
  <checkCode min="0" max="16">
    <!--ro, req, string, 文件加密密码, attr:min{req, int, 最小值},max{req, int, 最大值}-->1
  </checkCode>
  <GUID min="1" max="36">
    <!--ro, req, string, 导出时的唯一标识GUID, attr:min{req, int, 最小值},max{req, int, 最大值}-->1
  </GUID>
  <concurrentRequest>
    <!--ro, opt, int, 并发请求数-->1
  </concurrentRequest>
  <loginPassword min="8" max="16">
    <!--ro, req, string, 登录密码, attr:min{req, int, 最小值},max{req, int, 最大值}, desc:只返回节点和属性, 不返回节点取值内容。用于人脸库导出二次认证, 协议校验安全升级, 字段必传。-->null
  </loginPassword>
</ExportDetails>
<isSupportSyncFaceContrastResult>
  <!--ro, opt, bool, 是否支持同步人脸比对结果, desc:POST /ISAPI/Intelligent/FDLib/SyncFaceContrastResult?format=json-->true
</isSupportSyncFaceContrastResult>
</FDLibCap>

```

69.28.5 配置人脸库中的人脸数据

Request URL

PUT /ISAPI/Intelligent/FDLib/FDSetUp?format=json

查询参数

无

请求报文

```
{
  "faceURL": "http",
  /*opt, string, 图片url*/
  "faceLibType": "blackFD",
  /*req, enum, 人脸库类型, subType:string, [blackFD#名单库,staticFD#静态库], desc:必填, 人脸库类型: blackFD-名单库, staticFD-静态库, string类型, 最大长度为32*/
  "FDID": "test",
  /*req, string, 人脸库ID, desc:最大长度为63字节,多个人脸库用逗号隔开*/
  "FPID": " ",
  /*opt, string, 人脸记录ID, desc:可选, 人脸记录ID, string类型, 如果外部传入,最长63字节, 字母数字组合, 需要保证唯一性; 如果外部不传则由设备自动生成(与非视频工号(人员ID)字段一致)*/
  "deleteFP": true,
  /*opt, bool, 是否删除该人脸, desc:可选, boolean, 是否删除该人脸, true-是(只有删除该人脸时,才填写该字段; 新增或修改人脸时,不填写该字段)*/
  "modelData": "test",
  /*opt, string, 目标模型数据, desc:可选, 目标模型数据, 字符串类型, 传输过程中针对二进制非建模数据进行base64的加密处理*/
  "PicFeaturePoints": [
  /*opt, array, 特征图片坐标, subType:object,
  desc:图片特征点下发,若设备仅支持3种类型的特征点,平台下发3种以上特征点时,设备不能报错
  背景: 不同设备人脸评分算法不一致,导致平台评分通过的人脸下发至设备评分不通过,无法建模
  解决方法: 由上层平台采用统一的人脸评分以及特征点提取算法计算出图片的特征点并下发至设备,有了特征点可以保证设备建模成功,兼容不同设备人脸评分算法的差异
  性
  特征点下发方式: 特征点数据编码至人脸图片数据下发、特征点数据以json参数格式随人脸图片url下发
  设备获取特征点优先级从高到低: 1.图片内携带特征点 2.json格式特征点*/
  {
    "featurePointType": "face",
    /*req, enum, 特征点类型, subType:string, [face#人脸,leftEye#左眼,rightEye#右眼,leftMouthCorner#左嘴角,rightMouthCorner#右嘴角,nose#鼻子], desc:
    必填,string,特征点类型:face-人脸,leftEye-左眼,rightEye-右眼,leftMouthCorner-左嘴角,rightMouthCorner-右嘴角,nose-鼻子*/
    "coordinatePoint": {
      /*opt, object, 点坐标*/
      "x": 1,
      /*req, int, x坐标, range:[0,1000], desc:坐标进行归一化处理*/
      "y": 1,
      /*req, int, y坐标, range:[0,1000], desc:坐标进行归一化处理*/
      "width": 1,
      /*opt, int, 宽度, range:[0,1000], desc:当featurePointType为face时必须必填*/
      "height": 1
      /*opt, int, 高度, range:[0,1000], desc:当featurePointType为face时必须必填*/
    }
  }
],
  "faceType": "normalFace",
  /*opt, enum, 人脸类型, subType:string, [normalFace#普通人脸(默认),patrolFace#巡更人脸,hijackFace#胁迫人脸,superFace#超级人脸]*/
  "saveFacePic": true,
  /*opt, bool, 是否保存人脸底图*/
  "leaderPermission": [1, 2, 3, 4]
  /*opt, array, 首次认证权限, subType:int, range:[1,4],
  desc:数组为空表示清空人员首次认证权限
  表示人员具有1,2,3,4门编号的首次认证权限*/
}
```

参数名称	参数值	参数类型(Content-Type)	Content-ID	文件名称(filename)	描述
faceURL	[报文内容]	application/json	--	--	--
img	[图片二进制数据]	image/jpeg		faceImage.jpg	人脸图片

说明： 此协议通过表单格式传输， 表单框架详情说明见章节4.5.1.4， 如下实例：

```
--<frontier>
Content-Disposition: form-data; name=参数名称;filename=文件名称
Content-Type: 参数类型
Content-Length: ****
Content-ID: 内容ID
参数值
```

- 参数名称： 表单元Headers中Content-Disposition的name属性， 表示此表单元的名称
- 参数类型（Content-Type）： 表单元Headers中Content-Type属性
- 文件名称（filename）： 表单元Headers中Content-Disposition的filename属性， 仅在本表单元传输数据为文件类型存在， 表示表单元Body的文件名
- 参数值： 表单元的body内容

响应报文

```
{
  "statusCode": 1,
  /*ro, opt, int, 状态码, desc:无法用1表示时（1表示成功且无特殊状态）必须返回*/
  "statusString": "OK",
  /*ro, opt, string, 状态描述, range:[1,64], desc:无法用OK表示时（OK表示成功且无特殊状态）必须返回*/
  "subStatusCode": "ok",
  /*ro, opt, string, 子状态码, range:[1,64], desc:无法用ok表示（ok表示成功且无特殊状态）必须返回*/
  "errorCode": 1,
  /*ro, req, int, 错误码, desc:当statusCode不为1时，与subStatusCode对应*/
  "errorMsg": "ok"
  /*ro, req, string, 错误信息, desc:当statusCode不为1时，解释信息在协议约束中，允许设备在后续的版本迭代中，进行优化丰富提升（不限制死）*/
}
```

69.28.6 添加人脸数据

Request URL

POST /ISAPI/Intelligent/FDLib/FaceDataRecord?format=json

查询参数

无

请求报文

有效期：2024-08-15

湖南云舟电力科技有限公司

```
{
  "faceURL": "test",
  /*opt, string, 人脸url*/
  "faceLibType": "blackFD",
  /*req, enum, 人脸库类型, subType:string, [blackFD#名单库,staticFD#静态库], desc:人脸库类型: blackFD-名单库, staticFD-静态库, string类型, 最大长度为32*/
  "FDID": "test",
  /*req, string, 人脸库ID, desc:人脸库ID, string类型, 最大长度为63字节,多个人脸库用逗号隔开;*/
  "FPID": " test",
  /*opt, string, 人脸记录ID, desc:人脸记录ID（与非视频工号（人员ID）字段一致）, string类型, 最大长度为63字节*/
  "name": "张三",
  /*req, string, 人脸图片对应的人员姓名, desc:最大长度为96字节*/
  "gender": "male",
  /*opt, enum, 人脸图片对应的人员性别, subType:string, [male#男,female#女,unknown#未知], desc:最大长度为32*/
  "bornTime": "2004-05-03",
  /*req, string, 人脸图片对应的人员出生日期, desc:最大长度为20*/
  "city": "130100",
  /*opt, string, 人脸图片对应的人员出生城市编号, desc:人脸图片对应的人员出生城市编号(详见:全国各省份城市列表), string类型, 最大长度为32*/
  "certificateType": "officerID",
  /*opt, enum, 证件类型, subType:string, [officerID#军官证,ID#身份证,other#其他], desc:最大长度为10*/
  "certificateNumber": "test",
  /*opt, string, 证件号, desc:string类型, 最大长度为32字节（由数字和字母 "0-9a-zA-Z"组成）*/
  "caseInfo": "test",
  /*opt, string, 备注信息, desc:string类型, 最大长度为192字节, {dep if type == blackFD}*/
  "tag": "aa,bb,cc,dd",
  /*opt, string, 自定义标签, desc:自定义标签,单个标签最大48字节,(最多4个标签, 英文逗号隔开, 最大长度为195字节), string类型, , {dep if faceLibType == blackFD}*/
  "address": "test",
  /*opt, string, 人员住址, desc:最大长度为192字节, {dep if faceLibType == staticFD }*/
  "customInfo": "test",
  /*opt, string, 自定义信息, desc:最大长度为192字节, {dep if faceLibType == staticFD }*/
  "modelData": "test",
  /*opt, string, 目标模型数据, desc:目标模型数据, 字符串类型, 传输过程中针对二进制非建模数据进行base64的加密处理*/
  "transfer": true,
  /*opt, bool, PBG对接,
  desc:boolean类型, PBG对接, 需要设备开启微视云安全认证, 为了兼容原有功能, 对部分协议进行变更,
  涉及到名单添加和名单修改2个接口, 协议中增加是否转存的字段, 不填默认不转存, 如果是web操作, 需要将此字段写成需要转存, 智能应用服务器1.1.0增加*/
  "operateType": "byTerminal",
  /*opt, enum, 操作类型, subType:string, [byTerminal#按终端操作]*/
  "terminalNoList": [1],
  /*opt, array, type为byTerminal, subType:int, desc:type为byTerminal,byTerminal0ng时必须填,终端ID列表（目前仅支持单个终端）*/
  "PicFeaturePoints": [
    /*opt, array, 图片特征点坐标, subType:object*/
    {
      "featurePointType": "Face",
      /*req, enum, 特征点类型, subType:string, [face#人脸,leftEye#左眼,rightEye#右眼,leftMouthCorner#左嘴角,rightMouthCorner#右嘴角,nose#鼻子], desc:
      特征点类型:face-人脸,leftEye-左眼,rightEye-右眼,leftMouthCorner-左嘴角,rightMouthCorner-右嘴角,nose-鼻子*/
      "coordinatePoint": {
        /*opt, object, 坐标点, desc:坐标原点位于左上方*/
        "x": 1,
        /*req, int, x坐标, range:[0,1000], desc:坐标进行归一化处理*/
        "y": 1,
        /*req, int, y坐标, range:[0,1000], desc:坐标进行归一化处理*/
        "width": 1,
        /*opt, int, 宽度, range:[0,1000], desc:当featurePointType为face时必须填*/
        "height": 1
        /*opt, int, 高度, range:[0,1000], desc:当featurePointType为face时必须填*/
      }
    }
  ],
  "faceType": "normalFace",
  /*opt, enum, 人脸类型, subType:string, [normalFace#普通人脸（默认）,patrolFace#巡更人脸,hijackFace#胁迫人脸,superFace#超级人脸]*/
  "saveFacePic": true,
  /*opt, bool, 是否保存人脸底图*/
  "leaderPermission": [1, 2, 3, 4]
  /*opt, array, 首次认证权限, subType:int, range:[1,4],
  desc:数组为空表示清空人员首次认证权限
  表示人员具有1,2,3,4门编号的首次认证权限*/
}
```

参数名称	参数值	参数类型(Content-Type)	Content-ID	文件名称(filename)	描述
faceURL	[报文内容]	application/json	--	--	--
img	[图片二进制数据]	image/jpeg		facePic.jpg	人脸图片

说明： 此协议通过表单格式传输，表单框架详情说明见章节4.5.1.4，如下实例：

```
--<frontier>
Content-Disposition: form-data; name=参数名称;filename=文件名称
Content-Type: 参数类型
Content-Length: ****
Content-ID: 内容ID
参数值
```

- 参数名称：表单单元Headers中Content-Disposition的name属性，表示此表单单元的名称
- 参数类型（Content-Type）：表单单元Headers中Content-Type属性
- 文件名称（filename）：表单单元Headers中Content-Disposition的filename属性，仅在本表单单元传输数据为文件类型存在，表示表单单元Body的文件名
- 参数值：表单单元的body内容

响应报文

```
{
  "requestURL": "test",
  /*ro, opt, string, 请求URL*/
  "statusCode": 1,
  /*ro, req, int, 状态码*/
  "statusString": "test",
  /*ro, req, string, 状态描述*/
  "subStatusCode": "test",
  /*ro, req, string, 子状态码*/
  "errorCode": 1,
  /*ro, opt, int, 错误码, desc:当statusCode不为1时*/
  "errorMsg": "ok",
  /*ro, opt, string, 错误描述, desc:当statusCode不为1时*/
  "FPID": "test",
  /*ro, opt, string, 人脸记录ID, desc:人脸数据添加成功之后返回的人脸记录ID, 具有唯一性, string类型,最大长度为63字节, {dep if errcode == 1 && errMsg == ok}*/
  "rowKey": "test",
  /*ro, opt, string, 人脸记录rowKey, desc:人脸数据添加成功之后返回的人脸记录rowKey, 具有唯一性, string类型,最大长度为64字节, {dep if errcode == 1 && errMsg == ok}*/
}
```

69.28.7 修改人脸数据

Request URL

PUT /ISAPI/Intelligent/FDLib/FDModify?format=json

查询参数

无

请求报文

```
{
  "faceURL": "test",
  /*opt, string, 人脸url*/
  "faceLibType": "blackFD",
  /*req, enum, 人脸库类型, subType:string, [blackFD#名单库,staticFD#静态库]*/
  "FDID": "test",
  /*req, string, 人脸库ID, desc:最大长度为63字节,多个人脸库用逗号隔开*/
  "FPID": " ",
  /*opt, string, 人脸记录ID*/
  "name": "test",
  /*opt, string, 姓名*/
  "gender": "any",
  /*req, enum, 性别, subType:string, [any#条件不限（搜索的时候使用）,female#女,male#男,unknown#未知]*/
  "bornTime": "2004-05-03",
  /*req, time, 人脸图片对应的人员出生日期, desc:人脸图片对应的人员出生日期 格式应该是年月日,必填, 人脸图片对应的人员出生日期*/
  "city": "130100",
  /*opt, string, 城市编码, desc:出生城市编号(详见:全国各省份城市列表)*/
  "certificateType": "ID",
  /*req, enum, 证件类型, subType:string, [ID#身份证,LawyerCertificate#律师证,birthCertificate#出生证明,householdRegister#户口簿,judgeCertificate#法官证,officerID#军官证,other#其他,passport#护照,policeID#警官证,pressCard#记者证,prosecutorCard#检察官证,studentIDCard#学生证,temporaryIDCard#临时身份证,temporaryResidencePermit#居住证,workCard#工作证]*/
  "certificateNumber": "test",
  /*opt, string, 证件号*/
  "caseInfo": "test",
  /*opt, string, 备注信息, range:[0,192]*/
  "tag": "aa,bb,cc,dd",
  /*opt, string, 自定义标签*/
  "address": "test",
  /*opt, string, 人员住址*/
  "customInfo": "test",
  /*opt, string, 自定义信息*/
  "modelData": "test",
  /*opt, string, 目标模型数据*/
  "operateType": "byTerminal",
  /*opt, enum, 操作类型, subType:string, [byTerminal#按终端操作]*/
  "terminalNoList": [1],
  /*opt, array, 终端ID列表, subType:int, desc:byTerminalOrg时必填,终端ID列表（目前仅支持单个终端）*/
  "PicFeaturePoints": [
    /*opt, array, 特征图片坐标, subType:object*/
    {
      "featurePointType": "Face",
      /*req, enum, 特征点类型, subType:string, [face#人脸,leftEye#左眼,rightEye#右眼,leftMouthCorner#左嘴角,rightMouthCorner#右嘴角,nose#鼻子]*/
      "coordinatePoint": {
        /*opt, object, 点坐标*/
        "x": 1,
        /*req, int, x坐标, range:[0,1000], desc:坐标进行归一化处理*/
        "y": 1,
        /*req, int, y坐标, range:[0,1000], desc:坐标进行归一化处理*/
        "width": 1,
        /*opt, int, 宽度, range:[0,1000], desc:当featurePointType为face时必填*/
        "height": 1
        /*opt, int, 高度, range:[0,1000], desc:当featurePointType为face时必填*/
      }
    }
  ],
  "faceType": "normalFace",
  /*opt, enum, 人脸类型, subType:string, [normalFace#普通人脸（默认）,patrolFace#巡更人脸,hiJackFace#胁迫人脸,superFace#超级人脸]*/
  "saveFacePic": true,
  /*opt, bool, 是否保存人脸底图*/
  "leaderPermission": [1, 2, 3, 4]
  /*opt, array, 首次认证权限, subType:int, range:[1,4],
  desc:数组为空表示清空人员首次认证权限
  表示人员具有1,2,3,4门编号的首次认证权限*/
}
```

参数名称	参数值	参数类型(Content-Type)	Content-ID	文件名称(filename)	描述
faceURL	[报文内容]	application/json	--	--	--
img	[图片二进制数据]	image/jpeg		faceImage.jpg	人脸图片

说明： 此协议通过表格格式传输， 表单框架详情说明见章节4.5.1.4， 如下实例：

```
--<frontier>
Content-Disposition: form-data; name=参数名称;filename=文件名称
Content-Type: 参数类型
Content-Length: ****
Content-ID: 内容ID
参数值
```

- 参数名称： 表单单元Headers中Content-Disposition的name属性， 表示此表单单元的名称
- 参数类型（Content-Type）： 表单单元Headers中Content-Type属性

- 文件名称（filename）：表单单元Headers中Content-Disposition的filename属性，仅在本表单单元传输数据为文件类型存在，表示表单单元Body的文件名
- 参数值：表单单元的body内容

响应报文

```
{
  "statusCode": 1,
  /*ro, opt, int, 状态码, desc:无法用1表示时（1表示成功且无特殊状态）必须返回*/
  "statusString": "OK",
  /*ro, opt, string, 状态描述, range:[1,64], desc:无法用OK表示时（OK表示成功且无特殊状态）必须返回*/
  "subStatusCode": "ok",
  /*ro, opt, string, 子状态码, range:[1,64], desc:无法用ok表示（ok表示成功且无特殊状态）必须返回*/
  "errorCode": 1,
  /*ro, opt, int, 错误码, desc:当statusCode不为1时，与subStatusCode对应*/
  "errorMsg": "ok"
  /*ro, opt, string, 错误信息, desc:当statusCode不为1时，必须返回,解释信息在协议约束中，允许设备在后续的版本迭代中，进行优化丰富提升（不限制死）*/
}
```

69.29 人脸库管理（待优化）

69.29.1 创建人脸库信息

Request URL

POST /ISAPI/Intelligent/FDLib?format=json

查询参数

无

请求报文

```
{
  "faceLibType": "blackFD,staticFD",
  /*req, string, 人脸库类型, desc:人脸库类型: infraredFD -红外人脸库, blackFD-名单库, staticFD- 静态库, 多选时用逗号隔开, string类型, 最大长度为3*/
  "name": "test",
  /*req, string, 人脸库的名称, range:[0,48], desc:string类型, 最大长度为48字节, 名单库名称不能重复, 重复会提示错误*/
  "customInfo": "test",
  /*opt, string, 自定义信息, range:[0,192], desc:string类型, 最大长度为192字节, 该自定义信息用于标示数据的属性(或者唯一性)*/
  "libArmingType": "armingLib",
  /*opt, enum, 库布控类型, subType:string, [armingLib#布控库,nonArmingLib#非布控库], desc:老设备不支持该字段,默认是布控库,即新建的所有库都运行进行布控*/
  "libAttribute": "blackList",
  /*opt, enum, 库属性类型, subType:string, [general#普通库,blackList#黑名单库,VIP#VIP库,passerby#路人库(路人库不允许删除)]*/
  "FDID": "test"
  /*opt, string, 人脸库ID, desc:有本参数表示平台创建人脸库自己指定ID,若平台不下发本参数,由设备生成返回,同类型的人脸库ID具有唯一性, string类型, 最大长度为63字节*/
}
```

响应报文

```
{
  "requestURL": "test",
  /*ro, opt, string, 请求URL*/
  "statusCode": 1,
  /*ro, req, int, 状态码, desc:无法用1表示时（1表示成功且无特殊状态）必须返回*/
  "statusString": "OK",
  /*ro, req, string, 状态描述, range:[1,64], desc:无法用ok表示时（ok表示成功且无特殊状态）必须返回*/
  "subStatusCode": "ok",
  /*ro, req, string, 子状态码, range:[1,64], desc:无法用ok表示（ok表示成功且无特殊状态）必须返回*/
  "errorCode": 1,
  /*ro, opt, int, 错误码, desc:当statusCode不为1时，与subStatusCode对应*/
  "errorMsg": "ok",
  /*ro, opt, string, 错误信息, desc:当statusCode不为1时，解释信息在协议约束中，允许设备在后续的版本迭代中，进行优化丰富提升（不限制死）*/
  "FDID": "test"
  /*ro, opt, string, 人脸库ID, desc:平台创建人脸库自己指定ID,本参数返回相同ID, 平台不下发ID时，设备创建人脸库成功之后生成人脸库ID, 同类型的人脸库ID具有唯一性, string类型, 最大长度为63字节*/
}
```

69.29.2 获取人脸库中人脸记录总数

Request URL

GET /ISAPI/Intelligent/FDLib/Count?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "requestURL": "http://10.7.52.31:8080/kms/services/rest/dataInfoService/downloadFile",
  /*ro, opt, string, 请求URL*/
  "statusCode": 1,
  /*ro, opt, int, 状态码, desc:无法用1表示时（1表示成功且无特殊状态）必须返回*/
  "statusString": "OK",
  /*ro, opt, string, 状态描述, range:[1,64], desc:无法用OK表示时（OK表示成功且无特殊状态）必须返回*/
  "subStatusCode": "ok",
  /*ro, opt, string, 子状态码, range:[1,64], desc:无法用ok表示（ok表示成功且无特殊状态）必须返回*/
  "errorCode": 1,
  /*ro, req, int, 错误码, desc:当statusCode不为1时，与subStatusCode对应*/
  "errorMsg": "ok",
  /*ro, req, string, 错误信息, desc:当statusCode不为1时，解释信息在协议约束中，允许设备在后续的版本迭代中，进行优化丰富提升（不限制死）*/
  "FDRecordDataInfo": [
    /*ro, opt, array, 人脸库记录信息, subType:object, desc:人脸库记录信息, {dep if errcode==1&&errMsg==ok}*/
    {
      "FDID": "test",
      /*ro, opt, string, 人脸库ID, desc:人脸库ID, string类型, 最大长度为63字节*/
      "faceLibType": "blackFD",
      /*ro, opt, enum, 人脸库类型, subType:string, [blackFD#名单库,staticFD#静态库], desc:人脸库类型: blackFD-名单库, staticFD-静态库, string类型, 最大长度为32*/
      "name": "test",
      /*ro, opt, string, 人脸库的名称, desc:人脸库的名称, string类型, 最大长度为48字节*/
      "recordDataNumber": 123,
      /*ro, opt, int, 记录数据条数*/
      "libArmingType": "armingLib",
      /*ro, opt, enum, 库布控类型, subType:string, [armingLib#布控库,nonArmingLib#非布控库], desc:可选, 库布控类型:"armingLib:布控库, nonArmingLib:非布控库",老设备不支持该字段, 默认是布控库,即新建的所有库都运行进行布控,stin*/
      "libAttribute": "general",
      /*ro, opt, enum, 库属性类型, subType:string, [general#普通库,blackList#黑名单库,VIP#VIP库,passerby#路人库], desc:可选, 库属性类型:"general:普通库,blackList:黑名单库,VIP:VIP库, passerby:路人库,string*/
      "personnelFileEnabled": true
      /*ro, opt, bool, 是否启用人员档案配置*/
    }
  ],
  "totalRecordDataNumber": 500000,
  /*ro, opt, int, 所有记录数据条数, desc:所有记录数据条数, integer32类型*/
  "FDCapacity": {
    /*ro, opt, object, 人脸库容量信息*/
    "total": 0.000,
    /*ro, req, float, 总容量, desc:单位GB*/
    "use": 0.000,
    /*ro, req, float, 已用容量, desc:单位GB*/
    "remain": 0.000,
    /*ro, req, float, 剩余容量, desc:单位GB*/
    "maxRecordDataNumber": 0,
    /*ro, opt, int, 最大记录数据条数*/
    "useRecordDataNumber": 0,
    /*ro, opt, int, 已记录数据条数*/
    "remainRecordDataNumber": 0
    /*ro, opt, int, 剩余记录数据条数*/
  }
}
```

69.29.3 获取人脸库能力集

Request URL

GET /ISAPI/Intelligent/FDLib/capabilities?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "requestURL": "test",
  /*ro, opt, string, 请求URL*/
  "statusCode": 1,
  /*ro, req, int, 状态码*/
  "statusString": "test",
  /*ro, req, string, 状态描述*/
  "subStatusCode": "test",
  /*ro, req, string, 子状态码*/
}
```

```

/*ro, req, string, 人脸库名称*/
"errorCode": 1,
/*ro, opt, int, 当statusCode不为1时*/
"errorMsg": "ok",
/*ro, opt, string, 当statusCode不为1时*/
"FDNameMaxLen": 64,
/*ro, req, int, 人脸库最大名称长度*/
"customInfoMaxLen": 192,
/*ro, req, int, 自定义信息最大长度*/
"FDMaxNum": 3,
/*ro, req, int, 最大人脸库个数*/
"FDRecordDataMaxNum": 12345,
/*ro, req, int, 人脸库支持最大人脸记录数量*/
"supportFDFunction": "post,delete,put,get,setUp",
/*ro, req, string, 支持人脸库的增删改查、总数查询功能, desc:post-创建, delete-删除, put-修改, get-查询, setUp-设置*/
"isSupportFDSearch": true,
/*ro, req, bool, 是否支持人脸库搜索*/
"isSupportFDSearchDataPackage": true,
/*ro, req, bool, 是否支持人脸库搜索数据打包*/
"isSupportFDSearchByPic": true,
/*ro, req, bool, 是否支持人脸库以图搜图*/
"isSupportFDSearchByPicGenerate": true,
/*ro, req, bool, 是否支持人脸库以图搜图结果导出*/
"isSupportFDSearchDuplicate": true,
/*ro, req, bool, 是否支持人脸库查重*/
"isSupportFDSearchDuplicateGenerate": true,
/*ro, req, bool, 是否支持人脸库查重结果导出*/
"isSupportFCSearch": true,
/*ro, req, bool, 是否支持人脸库报警搜索*/
"isSupportFCSearchDataPackage": true,
/*ro, req, bool, 是否支持人脸库报警搜索数据打包*/
"isSupportFDExecuteControl": true,
/*ro, req, bool, 是否支持人脸库布控关系*/
"generateMaxNum": 1234,
/*ro, req, int, 人脸库支持的最大导出条数*/
"faceLibType": "blackFD,staticFD",
/*ro, opt, string, 人脸库类型*/
"modelMaxNum": 1000,
/*ro, opt, int, 查询的最大结果数量*/
"isSupportModelData": true,
/*ro, opt, bool, 是否支持模型数据下发*/
"isSupportFDLibArmingType": true,
/*ro, opt, bool, 是否支持人脸库布控类型*/
"isSupportFDLibSearch": true,
/*ro, opt, bool, 是否支持人脸库查询*/
"FDArmingRecordDataMaxNum": 12345,
/*ro, opt, int, 布控库库最大支持人脸记录数量*/
"isSupportControlPersonRecordByHumanId": true,
/*ro, opt, bool, 是否支持通过 humanId 修改和删除人脸记录*/
"isSupportControlPersonRecordByRowKey": true,
/*ro, opt, bool, 是否支持通过 rowKey 修改和删除人脸记录*/
"isSupportFaceLibRebuildCfg": true,
/*ro, opt, bool, 是否支持人脸库重新创建信息配置*/
"isSupportFDMove": true,
/*ro, opt, bool, 是否支持批量移动人脸库中的人脸数据, desc:/ISAPI/Intelligent/FDLib/FDMove/capabilities?format=json*/
"faceURLLen": 1024,
/*ro, opt, int, 人脸图片URL的最大长度, desc:人脸图片URL的最大长度 (如不返回该字段, 则设备默认支持人脸图片URL长度为256字节, 如返回该字段, 则该字段的值设备需要支持大于等于256) */
"isSupportArmingLibCfg": true,
/*ro, opt, bool, 是否支持布控库参数配置, desc:/ISAPI/Intelligent/FDLib/armingLibCfg/capabilities?format=json*/
"isSupportModelTransformation": true,
/*ro, opt, bool, 是否支持人脸库模型转换, desc:/ISAPI/Intelligent/FDLib/model/transformation/capabilities?format=json*/
"featurePointTypeList": ["face", "leftEye", "rightEye", "leftMouthCorner", "rightMouthCorner", "nose"],
/*ro, opt, array, 人脸图片特征点类型支持范围, subType:string*/
"isSupportPresort": true,
/*ro, opt, bool, 是否支持预分类配置, desc:对应接口/ISAPI/Intelligent/FDLib/presort/capabilities?format=json, 深思3.3.10和脸谱1.4.0后补能力, 只给web显示功能用(内部字段)*/
"libAttribute": {
/*ro, opt, object, 库属性类型, desc:"general:普通库,blackList:黑名单库,VIP:VIP库, passerby:路人库(路人库不允许删除)*/
"@opt": ["general", "blackList", "VIP", "passerby"]
/*ro, opt, array, 可选值, subType:string*/
},
"faceType": {
/*ro, opt, object, 人脸类型*/
"@opt": ["normalFace", "patrolFace", "hijackFace", "superFace"]
/*ro, opt, array, 枚举, subType:string*/
},
"saveFacePic": {
/*ro, opt, object, 是否保存人脸底图*/
"@opt": [true, false]
/*ro, opt, array, 枚举, subType:bool*/
},
"leaderPermission": {
/*ro, opt, object, 首次认证权限*/
"@size": 4,
/*ro, opt, int, 数组元素最大个数, range:[1,4]*/
"@min": 1,
/*ro, opt, int, 元素数值范围, range:[1,4]*/
"@max": 4
/*ro, opt, int, 元素数值范围, range:[1,4]*/
}
}
}

```

69.29.4 获取指定人脸库信息

Request URL

GET /ISAPI/Intelligent/FDLib?format=json&FDID=<FDID>&faceLibType=<faceLibType>&terminalNo=<terminalNo>

查询参数

参数名称	参数类型	描述
FDID	string	--
faceLibType	string	blackFD-名单库,staticFD-静态库
terminalNo	string	int,大于等于1,终端编号

请求报文

无

响应报文

```
{
  "requestURL": "test",
  /*ro, opt, string, 请求URL*/
  "statusCode": 1,
  /*ro, req, int, 状态码, desc:无法用1表示时（1表示成功且无特殊状态）必须返回*/
  "statusString": "OK",
  /*ro, req, string, 状态描述, range:[1,64], desc:无法用ok表示时（ok表示成功且无特殊状态）必须返回*/
  "subStatusCode": "ok",
  /*ro, req, string, 子状态码, range:[1,64], desc:无法用ok表示（ok表示成功且无特殊状态）必须返回*/
  "errorCode": 1,
  /*ro, opt, int, op, desc:当statusCode不为1时，与subStatusCode对应*/
  "errorMsg": "ok",
  /*ro, opt, string, 错误信息, desc:当statusCode不为1时，解释信息在协议约束中，允许设备在后续的版本迭代中，进行优化丰富提升（不限制死）*/
  "faceLibType": "blackFD",
  /*ro, opt, enum, 人脸库类型, subType:string, [blackFD#名单库,staticFD#静态库]*/
  "name": "test",
  /*ro, opt, string, 人脸库的名称, range:[0,48]*/
  "customInfo": "test",
  /*ro, opt, string, 自定义信息, range:[0,192]*/
  "libArmingType": "armingLib",
  /*ro, opt, enum, 库布控类型, subType:string, [armingLib#布控库,nonArmingLib#非布控库], desc:老设备不支持该字段,默认是布控库,即新建的所有库都运行进行布控*/
  /*
  "libAttribute": "blackList",
  /*ro, opt, enum, 库属性类型, subType:string, [blackList#黑名单库,VIP#VIP库,passerby#路人库（路人库不允许删除）]*/
  "personnelFileEnabled": true
  /*ro, opt, bool, 是否启用人员档案配置*/
}

```

69.29.5 删除全部人脸库信息

Request URL

DELETE /ISAPI/Intelligent/FDLib?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "statusCode": 1,
  /*ro, opt, int, 状态码, desc:无法用1表示时（1表示成功且无特殊状态）必须返回*/
  "statusString": "OK",
  /*ro, opt, string, 状态描述, range:[1,64], desc:无法用OK表示时（OK表示成功且无特殊状态）必须返回*/
  "subStatusCode": "ok",
  /*ro, opt, string, 子状态码, range:[1,64], desc:无法用ok表示（ok表示成功且无特殊状态）必须返回*/
  "errorCode": 1,
  /*ro, opt, int, 错误码, desc:当statusCode不为1时，与subStatusCode对应*/
  "errorMsg": "ok",
  /*ro, opt, string, 错误信息, desc:当statusCode不为1时，必须返回,解释信息在协议约束中，允许设备在后续的版本迭代中，进行优化丰富提升（不限制死）*/
}

```

69.29.6 删除人脸库中的人脸数据

Request URL

PUT /ISAPI/Intelligent/FDLib/FDSearch/Delete?format=json&FDID= <FDID> &faceLibType= <FDType>

查询参数

参数名称	参数类型	描述
FDID	string	最大长度为63字节人脸库ID. 创建人脸库：POST /ISAPI/Intelligent/FDLib?format=json 成功时返回
FDType	enum	人脸库类型 blackFD：名单库,staticFD：静态库

请求报文

```
{
  "FPID": [
    /*opt, array, 人脸记录ID列表, subType:object*/
    {
      "value": "test",
      /*req, string, 人脸记录ID, range:[0,63]*/
      "rowKey": "test"
      /*opt, string, 人脸记录数据库主键, range:[0,64], desc:可选,人脸记录数据库主键,根据rowKey进行查询,能提高查找效率,string,64字节 在轻量级产品中是必
      填字段*/
    }
  ],
  "operateType": "byTerminal",
  /*opt, enum, 操作类型, subType:string, [byTerminal#按终端操作]*/
  "terminalNoList": [1]
  /*opt, array, 终端ID列表, subType:int, desc:可选,array,type为byTerminal,byTerminalOrg时必填,终端ID列表（目前仅支持单个终端）*/
}
```

响应报文

```
{
  "statusCode": 1,
  /*ro, opt, int, 状态码, desc:无法用1表示时（1表示成功且无特殊状态）必须返回*/
  "statusString": "OK",
  /*ro, opt, string, 状态描述, range:[1,64], desc:无法用OK表示时（OK表示成功且无特殊状态）必须返回*/
  "subStatusCode": "ok",
  /*ro, opt, string, 子状态码, range:[1,64], desc:无法用ok表示（ok表示成功且无特殊状态）必须返回*/
  "errorCode": 1,
  /*ro, opt, int, 错误码, desc:当statusCode不为1时, 与subStatusCode对应*/
  "errorMsg": "ok"
  /*ro, opt, string, 错误信息, desc:当statusCode不为1时, 必须返回,解释信息在协议约束中, 允许设备在后续的版本迭代中, 进行优化丰富提升（不限制死）*/
}
```

69.30 音频本地输入

69.30.1 获取所有音频通道参数

Request URL

GET /ISAPI/System/Audio/channels

查询参数

无

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<AudioChannelList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, array, 音频通道列表, subType:object, attr:version{req, string, 协议版本}-->
  <AudioChannel>
    <!--ro, opt, object, 音频通道-->
    <id>
      <!--ro, req, string, 音频通道索引-->test
    </id>
    <enabled>
      <!--ro, req, bool, 音频使能-->true
    </enabled>
    <type opt="audioIn,audioOut">
      <!--ro, opt, enum, 音频通道管理, subType:string, [audioIn#音频输入,audioOut#音频输出], attr:opt{req, string, 取值范围}-->audioIn
    </type>
    <name>
      <!--ro, opt, string, 音频通道名称-->test
    </name>
  </AudioChannel>
</AudioChannelList>
```

69.30.2 获取单通道音频参数

Request URL

GET /ISAPI/System/Audio/channels/<audioID>

查询参数

参数名称	参数类型	描述
audioID	string	

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<AudioChannel xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, 音频通道, attr:version{req, string, 协议版本}-->
  <id>
    <!--ro, req, string, 音频通道索引-->test
  </id>
  <enabled>
    <!--ro, req, bool, 音频使能-->true
  </enabled>
  <type opt="audioIn,audioOut">
    <!--ro, opt, enum, 音频通道类型, subType:string, [audioIn#音频输入,audioOut#音频输出], attr:opt{req, string, 取值范围}-->audioIn
  </type>
  <name>
    <!--ro, opt, string, 音频通道名称-->test
  </name>
</AudioChannel>
```

69.30.3 获取音频能力

Request URL

GET /ISAPI/System/Audio/capabilities

查询参数

无

请求报文

无

响应报文

```

<?xml version="1.0" encoding="UTF-8"?>

<AudioCap xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, 音频能力, attr:version{req, string, 协议版本}-->
  <audioInputNums>
    <!--ro, req, int, 音频输入个数-->1
  </audioInputNums>
  <audioOutputNums>
    <!--ro, req, int, 音频输出个数-->1
  </audioOutputNums>
  <mixAudioInSet>
    <!--ro, opt, bool, 是否支持混音输入参数配置, desc: 对应接口: NET_DVR_MIX_AUDIOIN_CFG-->true
  </mixAudioInSet>
  <mixAudioOutSet>
    <!--ro, opt, bool, 是否支持混音输出参数配置, desc: 对应接口: NET_DVR_MIX_AUDIOOUT_CFG-->true
  </mixAudioOutSet>
  <isSupportAudioMixing>
    <!--ro, opt, bool, 是否支持音频混音, desc:/ISAPI/System/Audio/Mixing/capabilities-->true
  </isSupportAudioMixing>
  <isSupportRemoveFrequencyPoint>
    <!--ro, opt, bool, 是否支持啸叫频点清除-->true
  </isSupportRemoveFrequencyPoint>
  <isSupportAudioInConfig>
    <!--ro, opt, bool, 是否支持全部音量输入配置, desc: 对应/ISAPI/System/Audio/AudioIn/capabilities-->true
  </isSupportAudioInConfig>
  <muteDuringPanTilt opt="MicIn">
    <!--ro, opt, string, 是否支持运动时消音, attr:opt{req, string, 取值范围}, desc: 目前只支持MicIn,/ISAPI/System/TwoWayAudio/channels/ID的muteDuringPanTilt
节点配置-->test
  </muteDuringPanTilt>
  <isSupportAudioOutConfig>
    <!--ro, opt, bool, 是否支持全部音量输出配置, desc: 对应/ISAPI/System/Audio/AudioOut/capabilities?format=json-->true
  </isSupportAudioOutConfig>
  <isSupportSAIP>
    <!--ro, opt, bool, 是否支持独立音轨通道接入设备-->true
  </isSupportSAIP>
  <SAIPChannelsNum>
    <!--ro, opt, int, 支持独立音轨通道接入的设备数量, dep:and,{$.AudioCap.isSupportSAIP,eq,true}-->1
  </SAIPChannelsNum>
  <isSupportDataStorage>
    <!--ro, opt, bool, 是否支持音频文件上传, desc: 对应url: /ISAPI/System/Audio/dataStorage?format=json-->true
  </isSupportDataStorage>
  <isSupportDataRelease>
    <!--ro, opt, bool, 是否支持音频文件发布, desc: 对应url: /ISAPI/System/Audio/dataRelease?format=json-->true
  </isSupportDataRelease>
  <isSupportAudioPlan>
    <!--ro, opt, bool, 是否支持音频计划方案, desc: 对应url: /ISAPI/System/Audio/plan?format=json-->true
  </isSupportAudioPlan>
  <isSupportAudioTask>
    <!--ro, opt, bool, 是否支持音频计划任务, desc: 对应url /ISAPI/System/Audio/pPlan/<PlanID>/task?format=json-->true
  </isSupportAudioTask>
  <isSupportaudioCollection>
    <!--ro, opt, bool, 是否支持音频收集, desc: 对应url:/ISAPI/System/Audio/audioCollection/capabilities?format=json, 维护功能, 用于采集音频, 便于音频问题分析-->true
  </isSupportaudioCollection>
  <isSupportWirelessPickUp>
    <!--ro, opt, bool, 是否支持无线拾音器设备接入, desc: 对应url:/ISAPI/System/Audio/wirelessPickUp/capabilities?format=json-->true
  </isSupportWirelessPickUp>
  <audioOutputBroadcastNums opt="1,2,3,4">
    <!--ro, opt, int, 音频广播输出个数, attr:opt{opt, string, 取值范围}, desc: 当前仅广播音响设备支持本节点。返回opt="1,2,3,4"时输出口以此为准, 不返回
opt="1,2,3,4"时, 输出口默认从1开始。音响设备内部会按业务类型在硬件上划分输出口, 有广播输出口和对讲输出口(音频输出口)。由于输出口的起始编号均为1开始, 所以需要
返回具体的广播输出口支持个数。返回该节点表示支持音频广播输出音量配置, 不返回表示不支持, 对应URI:/ISAPI/System/Audio/AudioOut/broadcast/capabilities?
format=json-->0
  </audioOutputBroadcastNums>
  <isSupportAudioOutMixing>
    <!--ro, opt, bool, 是否支持输出音频混音, desc: 对应url: /ISAPI/System/Audio/AudioOut/mixing/capabilities?format=json-->true
  </isSupportAudioOutMixing>
  <isSupportRTSPAUDIO>
    <!--ro, opt, bool, 是否支持实时监听RTSP音频, desc: rtsp uri:/ISAPI/Streaming/audio/<audioID>, audioID- 音频通道-->true
  </isSupportRTSPAUDIO>
  <AudioBoard>
    <!--ro, opt, object, 是否支持音频板相关配置-->
    <isSupportNetworkParamCfg>
      <!--ro, opt, bool, 是否支持音频板网络参数配置, desc:/ISAPI/System/Audio/AudioBoard/NetworkParamCfg/capabilities?format=json-->true
    </isSupportNetworkParamCfg>
  </AudioBoard>
  <isSupportRTSPAUDIOPlaybackByFile>
    <!--ro, opt, bool, 是否支持按照文件回放RTSP音频, desc: RTSP取流URL, 来自于/ISAPI/ContentMgmt/recordAudio/search?format=json返回的playbackURI-->true
  </isSupportRTSPAUDIOPlaybackByFile>
</AudioCap>

```

69.30.4 配置指定音频输入通道参数

Request URL

PUT /ISAPI/System/Audio/AudioIn/channels/<audioID>

查询参数

参数名称	参数类型	描述
audioID	string	

请求报文

```
<?xml version="1.0" encoding="UTF-8"?>

<AudioIn xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--opt, object, 单通道音频输入参数, attr:version{req, string, 协议版本}-->
  <id>
    <!--req, string, 索引-->test
  </id>
  <MixAudioIn>
    <!--opt, object, 混音参数-->
    <enabled>
      <!--req, bool, 使能-->true
    </enabled>
    <audioInputType>
      <!--opt, enum, 音频输入类型, subType:string, [macIn#麦克风输入,LineIn#线路输入]-->macIn
    </audioInputType>
    <highPassFilter>
      <!--opt, int, 高通滤波, range:[0,127], unit:Hz, unitType:频率-->1
    </highPassFilter>
    <noiseMargin>
      <!--opt, int, 噪声门限, range:[0,127], desc:单位: dB (分贝) -->1
    </noiseMargin>
    <AutoLimitWave>
      <!--opt, object, 自动限波控制参数-->
      <FBCEnable>
        <!--req, bool, 虎啸抑制-->true
      </FBCEnable>
      <mode>
        <!--req, enum, 模式, subType:string, [fast#快,general#一般,slow#慢]-->fast
      </mode>
      <filterQValue>
        <!--req, string, 自适应滤波器Q值, desc:单位: Oct-->1
      </filterQValue>
      <staticFilterNum>
        <!--req, int, 静态滤波器数量, range:[0,12]-->1
      </staticFilterNum>
    </AutoLimitWave>
    <DSPPParameter>
      <!--opt, object, 数字信号处理芯片参数-->
      <FBCEnable>
        <!--req, bool, 虎啸抑制-->true
      </FBCEnable>
      <volume>
        <!--req, int, 输入音量, range:[0,255]-->1
      </volume>
    </DSPPParameter>
  </MixAudioIn>
  <AudioInVolumelist>
    <!--opt, object, 输入声音列表-->
    <AudioInVlome>
      <!--opt, object, 输入声音-->
      <type>
        <!--req, enum, 类型, subType:string, [audioInput#音频输入,audioOutput#音频输出,audioEncode#音频编码]-->audioOutput
      </type>
      <volume>
        <!--req, int, 声音-->1
      </volume>
    </AudioInVlome>
  </AudioInVolumelist>
  <automaticGainControl>
    <!--opt, int, 自动增益调节配置, desc:针对每个通道MIC1-MIC16-->1
  </automaticGainControl>
  <energyThreshold>
    <!--opt, int, 能量门限, desc:针对每个通道MIC1-MIC16-->1
  </energyThreshold>
  <PhantomPower>
    <!--opt, object, 幻象电源配置-->
    <enabled>
      <!--req, bool, 是否开启幻象电源-->true
    </enabled>
  </PhantomPower>
  <audioInCodingFormat>
    <!--opt, enum, 音频输入编码格式, subType:string, [G.711U#G.711U,G.711A#G.711A,OPUS#OPUS,AAC#AAC,PCM#PCM], desc:默认输入编码格式根据设备实际情况实现, eg. 当前对讲产品中音箱实现为默认G.711U-->G.711U
  </audioInCodingFormat>
  <audioInModeEnabled>
    <!--opt, bool, 音频输入方式选择使能, desc:开启后设备根据用户自定义选择的输入方式(audioInMode)对进行音频采集;关闭后用户不可配置输入方式, 由设备自动检测输入方式进行音频采集。默认关闭-->true
  </audioInModeEnabled>
  <audioInMode>
    <!--opt, enum, 音频输入方式, subType:string, [LeftMic#左麦克风,rightMic#右麦克风,bluetooth#蓝牙,LineIn#线路输入,earphone#耳机输入,6.5mm#6.5mm输入,RCALeft#RCA(莲花头)左声道输入,RCARight#RCA(莲花头)右声道输入,embeddedAudio#内嵌音频], dep:and,{$.AudioIn.audioInModeEnable,eq,true}, desc:左右麦克风为方位输入,输入的音频流经DSP处理后传输给扬声器,使声音可优化;线路输入直接将音频流传输给扬声器,不进行声音优化;蓝牙输入为配置该通道为蓝牙芯片接入-->leftMic
  </audioInMode>
  <audioInSensitivity>
```

```
<!--opt, enum, 音频输入灵敏度, subType:string, [high#高,mid#中,Low#低], dep:and,{$.AudioIn.audioInMode,eq,LineIn}, desc:表示音频档位; 当音频输入方式为线路输入(audioInMode=LineIn)时可选有效,默认高-->high
</audioInSensitivity>
<AGCEnabled>
  <!--opt, bool, 音频自动增益使能, desc:开启之后, 使放大电路的增益自动地随信号强度而调整.-->true
</AGCEnabled>
<noiseReductionEnabled>
  <!--opt, bool, 音频降噪使能, desc:开启之后可以极大消除音频中的多种噪声-->true
</noiseReductionEnabled>
<AECEnable>
  <!--opt, bool, 回声消除-->true
</AECEnable>
<audioSamplingRate>
  <!--opt, int, 音频采样率, desc:单位:kHz-->0
</audioSamplingRate>
<audioBitRate>
  <!--opt, int, 音频码率, desc:0#32Kbps,1#64Kbps,2#128Kbps,3#256Kbps,4#512Kbps-->0
</audioBitRate>
<noiseReductionLevel>
  <!--opt, int, 降噪等级, range:[0,6], desc:0表示关闭降噪-->0
</noiseReductionLevel>
<AGCGain>
  <!--opt, int, AGC增益, range:[0,100], step:1, unit:%, unitType:级差-->50
</AGCGain>
</AudioIn>
```

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 响应消息, attr:version{ro, req, string, 版本}-->
  <requestURL>
    <!--ro, req, string, 请求的URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, 状态码, subType:int, [0#OK,1#OK,2#Device Busy,3#Device Error,4#Invalid Operation,5#Invalid XML Format,6#Invalid XML Content,7#Reboot Required]-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, 状态信息, subType:string, [OK#成功,Device Busy#设备忙,Device Error#设备异常,Invalid Operation#无效的操作,Invalid XML Format#无效的XML格式,Invalid XML Content#无效的XML内容,Reboot#设备重启]-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, 详细错误码英文描述, desc:详细错误码的英文描述-->OK
  </subStatusCode>
</ResponseStatus>
```

69.30.5 获取指定音频输入通道参数

Request URL

GET /ISAPI/System/Audio/AudioIn/channels/<audioID>

查询参数

参数名称	参数类型	描述
audioID	string	

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<AudioIn xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, 单通道音频输入参数, attr:version{req, string, 协议版本}-->
  <id>
    <!--ro, req, string, 索引-->test
  </id>
  <MixAudioIn>
    <!--ro, opt, object, 混音参数-->
    <enabled>
      <!--ro, req, bool, 使能-->true
    </enabled>
    <audioInputType>
      <!--ro, opt, enum, 音频输入类型, subType:string, [macIn#麦克风输入,LineIn#线路输入]-->macIn
    </audioInputType>
    <highPassFilter>
      <!--ro, opt, int, 高通滤波, range:[0,127], unit:Hz, unitType:频率-->1
    </highPassFilter>
  </MixAudioIn>
</AudioIn>
```

```

</highPassFilter>
<noiseMargin>
  <!--ro, opt, int, 噪声门限, range:[0,127], desc:单位: dB (分贝)-->1
</noiseMargin>
<AutoLimitWave>
  <!--ro, opt, object, 自动限波控制参数-->
  <FBCEnable>
    <!--ro, req, bool, 虎啸抑制-->true
  </FBCEnable>
  <mode>
    <!--ro, req, enum, 模式, subType:string, [fast#快,general#一般,slow#慢]-->fast
  </mode>
  <filterQValue>
    <!--ro, req, string, 自适应滤波器Q值, desc:单位: Oct-->1
  </filterQValue>
  <staticFilterNum>
    <!--ro, req, int, 静态滤波器数量, range:[0,12]-->1
  </staticFilterNum>
</AutoLimitWave>
<DSPPParameter>
  <!--ro, opt, object, 数字信号处理芯片参数-->
  <FBCEnable>
    <!--ro, req, bool, 虎啸抑制-->true
  </FBCEnable>
  <volume>
    <!--ro, req, int, 输入音量, range:[0,255]-->1
  </volume>
</DSPPParameter>
</MixAudioIn>
<AudioInVolumelist>
  <!--ro, opt, object, 输入声音列表-->
  <AudioInVlome>
    <!--ro, opt, object, 输入声音-->
    <type>
      <!--ro, req, enum, 类型, subType:string, [audioInput#音频输入,audioOutput#音频输出,audioEncode#音频编码]-->audioOutput
    </type>
    <volume>
      <!--ro, req, int, 声音-->1
    </volume>
  </AudioInVlome>
</AudioInVolumelist>
<automaticGainControl>
  <!--ro, opt, int, 自动增益调节配置, desc:针对每个通道MIC1~MIC16-->1
</automaticGainControl>
<energyThreshold>
  <!--ro, opt, int, 能量门限, desc:针对每个通道MIC1~MIC16-->1
</energyThreshold>
<PhantomPower>
  <!--ro, opt, object, 幻象电源配置-->
  <enabled>
    <!--ro, req, bool, 是否开启幻象电源-->true
  </enabled>
</PhantomPower>
<audioInCodingFormat>
  <!--ro, opt, enum, 音频输入编码格式, subType:string, [G.711U#G.711U,G.711A#G.711A,OPUS#OPUS,AAC#AAC,PCM#PCM], desc:默认输入编码格式根据设备实际情况实现, eg.当前对讲产品中音箱实现为默认G.711U-->G.711U
</audioInCodingFormat>
<audioInModeEnabled>
  <!--ro, opt, bool, 音频输入方式选择使能, desc:开启后设备根据用户自定义选择的输入方式(audioInMode)对进行音频采集;关闭后用户不可配置输入方式,由设备自动检测输入方式进行音频采集。默认关闭-->true
</audioInModeEnabled>
<audioInMode>
  <!--ro, opt, enum, 音频输入方式, subType:string, [LeftMic#左麦克风,rightMic#右麦克风,bluetooth#蓝牙,lineIn#线路输入,earphone#耳机输入,6.5mm#6.5mm输入,RCALeft#RCA(莲花头)左声道输入,RCARight#RCA(莲花头)右声道输入,embeddedAudio#内嵌音频], dep:and,{$.AudioIn.audioInModeEnable,eq,true}, desc:左右麦克风为方位输入,输入的音频流经DSP处理后传输给扬声器,使声音可优化;线路输入直接将音频流传输给扬声器,不进行声音优化;蓝牙输入为配置该通道为蓝牙芯片接入-->leftMic
</audioInMode>
<audioInSensitivity>
  <!--ro, opt, enum, 音频输入灵敏度, subType:string, [high#高,mid#中,low#低], dep:and,{$.AudioIn.audioInMode,eq,lineIn}, desc:表示音频档位:当音频输入方式为线路输入(audioInMode=lineIn)时可选有效,默认高-->high
</audioInSensitivity>
<AGCEnable>
  <!--ro, opt, bool, 音频自动增益使能, desc:开启之后,使放大电路的增益自动地随信号强度而调整.-->true
</AGCEnable>
<noiseReductionEnabled>
  <!--ro, opt, bool, 音频降噪使能, desc:开启之后可以极大消除音频中的多种噪声-->true
</noiseReductionEnabled>
<AECEnable>
  <!--ro, opt, bool, 回声消除-->true
</AECEnable>
<audioSamplingRate>
  <!--ro, opt, int, 音频采样率, desc:单位:kHz-->0
</audioSamplingRate>
<audioBitRate>
  <!--ro, opt, int, 音频码率, desc:0#32Kbps,1#64Kbps,2#128Kbps,3#256Kbps,4#512Kbps-->0
</audioBitRate>
<noiseReductionLevel>
  <!--ro, opt, int, 降噪等级, range:[0,6], desc:0表示关闭降噪-->0
</noiseReductionLevel>
<AGCGain>
  <!--ro, opt, int, AGC增益, range:[0,100], step:1, unit:%, unitType:级差-->50
</AGCGain>
</AudioIn>

```

69.30.6 获取指定音频输入通道参数能力

Request URL

GET /ISAPI/System/Audio/AudioIn/channels/<audioID>/capabilities

查询参数

参数名称	参数类型	描述
audioID	string	音频通道

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<AudioInCap xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, 单通道音频输入能力集, attr:version{req, string, 协议版本}-->
  <id>
    <!--ro, req, string, 索引-->test
  </id>
  <MixAudioIn>
    <!--ro, opt, object, 混音参数-->
    <enabled opt="true,false">
      <!--ro, req, bool, 使能, attr:opt{req, string, 取值范围}-->true
    </enabled>
    <audioInputType opt="micIn, lineIn">
      <!--ro, opt, enum, 音频输入类型, subType:string, [macIn#麦克风输入,LineIn#线路输入], attr:opt{req, string, 取值范围}-->macIn
    </audioInputType>
    <highPassFilter
      opt="0,8,16,24,31,39,47,55,63,71,79,87,94,102,110,118,126,134,142,150,157,165,173,181,189,197,205,213,220,228,236,244,252,260,268,276,283,291,299,307,315,32
      3,331,339,346,354,362,370,378,386,394,402,409,417,425,433,441,449,457,465,472,480,488,496,504,512,520,528,536,543,551,559,567,575,583,591,598,606,614,622,63
      0,638,646,654,661,669,677,685,693,701,709,717,724,732,740,748,756,764,772,780,787,795,803,811,819,827,835,843,850,858,866,874,882,890,898,906,913,921,929,93
      7,945,953,961,969,976,984,992,1000" def="30">
      <!--ro, opt, int, 高通滤波, range:[0,127], unit:Hz, unitType:频率, attr:opt{req, string, 取值范围},def{req, string, 默认值}-->1
    </highPassFilter>
    <noiseMargin opt="1000,660,657,653,650,647,643,640,637,633,-
      630,627,623,620,617,613,610,607,603,600,597,593,590,587,583,580,577,573,570,567,563,560,557,553,550,547,543,540,537,533,530,527,523,520,517,513,510,507,503,
      500,497,493,490,487,483,480,477,473,470,467,463,460,457,453,450,447,443,440,437,433,430,427,423,420,417,413,410,407,403,400,397,393,390,387,383,380,377,373,
      370,367,363,360,357,353,350,347,343,340,337,333,330,327,323,320,317,313,310,307,303,300,297,293,290,287,283,280,277,273,270,267,263,260,257,253,250,247,243,
      240" def="30">
      <!--ro, opt, int, 噪声门限, range:[0,127], attr:opt{req, string, 取值范围},def{req, string, 默认值}, desc:单位: dB (分贝)-->1
    </noiseMargin>
    <AutoLimitWave>
      <!--ro, opt, object, 自动限波控制参数-->
      <FBCEnable opt="true,false">
        <!--ro, req, bool, 虎啸抑制, attr:opt{req, string, 取值范围}-->true
      </FBCEnable>
      <mode opt="fast,general,slow" def="general">
        <!--ro, req, enum, 模式, subType:string, [fast#快,general#一般,slow#慢], attr:opt{req, string, 取值范围},def{req, string, 默认值}-->fast
      </mode>
      <filterQValue opt="40,10" def="40">
        <!--ro, req, int, 自适应滤波器Q值, attr:opt{req, string, 取值范围},def{req, string, 默认值}, desc:单位: Oct-->1
      </filterQValue>
      <staticFilterNum min="0" max="12" def="0">
        <!--ro, req, int, 静态滤波器数量, range:[0,12], attr:min{req, int, 最小值},max{req, int, 最大值},def{req, int, 默认值}-->1
      </staticFilterNum>
    </AutoLimitWave>
    <DSPPParameter>
      <!--ro, opt, object, 数字信号处理芯片参数-->
      <FBCEnable>
        <!--ro, req, bool, 虎啸抑制-->true
      </FBCEnable>
      <volume min="1" max="10">
        <!--ro, req, int, 输入音量, range:[0,255], attr:min{req, int, 最小值},max{req, int, 最大值}-->1
      </volume>
    </DSPPParameter>
  </MixAudioIn>
  <AudioInVolumelist>
    <!--ro, opt, object, 输入声音列表-->
    <AudioInVlome>
      <!--ro, opt, object, 输入声音-->
      <type opt="audioInput,audioOutput,audioEncode">
        <!--ro, req, enum, 类型, subType:string, [audioInput#音频输入,audioOutput#音频输出,audioEncode#音频编码], attr:opt{req, string, 取值范围}-->
      </type>
    </AudioInVlome>
  </AudioInVolumelist>
  <automaticGainControl min="0" max="12" default="12">
    <!--ro, opt, int, 自动增益调节配置, attr:min{req, int, 最小值},max{req, int, 最大值},default{opt, string, 默认值}, desc:针对每个通道MIC1~MIC16-->1
  </automaticGainControl>
  <noiseThreshold min="00" max="0" default="55">
```

```
<energyThreshold min="0" max="0" default="-33">
  <!--ro, opt, int, 能量门限, attr:min{req, int, 最小值},max{req, int, 最大值},default{opt, string, 默认值}, desc:针对每个通道MIC1-MIC16-->1
</energyThreshold>
<PhantomPower>
  <!--ro, opt, object, 幻象电源配置-->
  <enabled opt="true,false" def="false">
    <!--ro, req, bool, 是否开启幻象电源, attr:opt{req, string, 取值范围},def{opt, string, 默认值}-->true
  </enabled>
</PhantomPower>
<audioInCodingFormat opt="G.711U,G.711A,OPUS,PCM,AAC" def="G.711U">
  <!--ro, opt, enum, 音频输入编码格式, subType:string, [G.711U#G.711U,G.711A#G.711A,OPUS#OPUS,AAC#AAC,PCM#PCM], attr:opt{req, string, 取值范围},def{req, string, 默认值}, desc:默认输入编码格式根据设备实际情况实现,eg.当前对讲产品中音箱实现为默认G.711U-->G.711U
</audioInCodingFormat>
<audioInModeEnabled>
  <!--ro, opt, bool, 音频输入方式选择使能, desc:开启后设备根据用户自定义选择的输入方式(audioInMode)对进行音频采集;关闭后用户不可配置输入方式,由设备自动检测输入方式进行音频采集.默认关闭-->true
</audioInModeEnabled>
<audioInMode opt="leftMic,rightMic,bluetooth,lineIn,earphone,6.5mm,RCAleft,RCAright,embeddedAudio,externalInput">
  <!--ro, opt, enum, 音频输入方式, subType:string, [leftMic#左麦克风,rightMic#右麦克风,bluetooth#蓝牙,lineIn#线路输入,earphone#耳机输入,6.5mm#6.5mm输入,RCAleft#RCA(莲花头)左声道输入,RCAright#RCA(莲花头)右声道输入,externalInput#外部输入], dep:and,{$.AudioInCap.audioInModeEnable,eq,true}, attr:opt{req, string, 取值范围}, desc:左右麦克风为方位输入,输入的音频流经过DSP处理后传输给扬声器,使声音可优化;线路输入直接将音频流传输给扬声器,不进行声音优化;蓝牙输入为配置该通道为蓝牙芯片接入,耳机和6.5mm输入在硬件上有差别;RCA为莲花头输入,RCA分左声道和右声道分别是RCAleft和RCAright-->leftMic
</audioInMode>
<audioInSensitivity opt="high,mid,low">
  <!--ro, opt, enum, 音频输入灵敏度, subType:string, [high#高,mid#中,low#低], dep:and,{$.AudioInCap.audioInMode,eq,lineIn}, attr:opt{req, string, 取值范围}, desc:表示档位.当音频输入方式为线路输入(audioInMode=lineIn)时可选有效,默认高-->high
</audioInSensitivity>
<AGCEnabled opt="true,false" def="true">
  <!--ro, opt, bool, 音频自动增益使能, attr:opt{req, string, 取值范围},def{req, string, 默认值}, desc:开启之后,使放大电路的增益自动地随信号强度而调整.-->true
</AGCEnabled>
<noiseReductionEnabled opt="true,false">
  <!--ro, opt, bool, 音频降噪使能, attr:opt{req, string, 取值范围}, desc:开启之后可以极大消除音频中的多种噪声-->true
</noiseReductionEnabled>
<AECEnable opt="true,false">
  <!--ro, opt, bool, 回声消除, attr:opt{req, string, 取值范围}-->true
</AECEnable>
<audioSamplingRate opt="16,32" def="16">
  <!--ro, opt, int, 音频采样率, attr:opt{req, string, 取值范围},def{req, string, 默认值}, desc:单位:kHz-->0
</audioSamplingRate>
<audioBitRate opt="1,2,3,4,5" def="0">
  <!--ro, opt, int, 音频码率, attr:opt{req, string, 取值范围},def{req, string, 默认值}, desc:0#32Kbps,1#64Kbps,2#128Kbps,3#256Kbps,4#512Kbps-->0
</audioBitRate>
<isSupportAudioSceneSettings>
  <!--ro, opt, bool, 是否支持音频场景高级参数配置, desc:对应uri:/ISAPI/System/Audio/AudioIn/channels/<audioID>/sceneSettings/capabilities?format=json-->true
</isSupportAudioSceneSettings>
<isSupportSoundSourcePosition>
  <!--ro, opt, bool, 是否支持音频分离声源方位信息, desc:对应uri:/ISAPI/System/Audio/AudioIn/channels/<audioID>/soundSourcePosition/capabilities?format=json-->true
</isSupportSoundSourcePosition>
<noiseReductionLevel min="0" max="6">
  <!--ro, opt, int, 降噪等级, range:[0,6], attr:min{req, int, 最小值},max{req, int, 最大值}, desc:0表示关闭降噪-->0
</noiseReductionLevel>
<AGCGain min="0" max="100" def="50">
  <!--ro, opt, int, AGC增益, range:[0,100], step:1, unit:%, unitType:级差, attr:min{req, int, 最小值},max{req, int, 最大值},def{req, int, 默认值}-->50
</AGCGain>
</AudioInCap>
```

69.31 音频本地输出

69.31.1 配置指定音频输出通道参数

Request URL

PUT /ISAPI/System/Audio/AudioOut/channels/<audioID>

查询参数

参数名称	参数类型	描述
audioID	string	

请求报文

```
<?xml version="1.0" encoding="UTF-8"?>

<AudioOut xmlns="http://www.isapi.org/ver20/XMLSchema" version="20">
  <!--opt, object, 单通道音频输出参数, attr:version{req, string, 协议版本}-->
  <id>
    <!--req, string, 索引-->test
  </id>
  <MixAudioOut>
    <!--opt, object, 混音参数-->
    <enabled>
      <!--req, bool, 使能-->true
    </enabled>
  </MixAudioOut>
</AudioOut>
```

```

</enabled>
<modulatorEnable>
  <!--req, bool, 是否启用相位调制器-->true
</modulatorEnable>
<postFilter>
  <!--req, bool, Post Filter开关-->true
</postFilter>
<limitPressure>
  <!--req, bool, 输出压限开关-->true
</limitPressure>
<modulatorValue>
  <!--req, int, 相位调制量, range:[0,127], desc:默认10-->1
</modulatorValue>
<triggerTime>
  <!--opt, int, 压限触发时间, range:[1,872], unit:ms, unitType:时间, desc:默认为4-->1
</triggerTime>
<freeTime>
  <!--opt, int, 压限释放时间, range:[26,46062], unit:ms, unitType:时间, desc:默认为42-->26
</freeTime>
<compressThreshold>
  <!--req, int, 压缩门限-->1
</compressThreshold>
<compressMode>
  <!--req, enum, 压缩模式, subType:string, [soft#软压缩,hard#硬压缩]-->soft
</compressMode>
<compressRate>
  <!--req, int, 压缩率, range:[0,127]-->1
</compressRate>
<recoveryGain>
  <!--req, int, 回复增益-->1
</recoveryGain>
<outputGain>
  <!--req, int, 输出增益-->1
</outputGain>
<outputMute>
  <!--opt, bool, 输出静音-->true
</outputMute>
<soundQualityHandling>
  <!--opt, bool, 音质处理-->true
</soundQualityHandling>
<outputVolume>
  <!--opt, int, 输出音量, range:[0,100]-->1
</outputVolume>
</MixAudioOut>
<AudioOutVolumelist>
  <!--opt, object, 输出音频声音列表-->
  <AudioOutVlome>
    <!--opt, object, 输出音频声音-->
    <type>
      <!--req, enum, 类型, subType:string, [audioOutput#音频输出,audioEncode#音频编码,specific#特定通道]-->audioOutput
    </type>
    <volume>
      <!--req, int, 系统音量, desc:用于表示设备音频输出出口的系统音量-->1
    </volume>
    <id>
      <!--opt, string, 音频输入通道索引, dep:and,{$.AudioOutCap.AudioOutVolumelist.AudioOutVlome.type,eq,specific}-->test
    </id>
    <name>
      <!--opt, string, 音频输入通道名称, dep:and,{$.AudioOutCap.AudioOutVolumelist.AudioOutVlome.type,eq,specific}-->test
    </name>
    <talkVolume>
      <!--opt, int, 对讲音量, range:[0,100], desc:音量百分比,仅作为对讲音量单独使用,在对讲业务时生效。当系统音量volume节点为0时,对讲音量不可用-->0
    </talkVolume>
    <trebleVolume>
      <!--opt, int, 高音音量, range:[0,100]-->0
    </trebleVolume>
    <bassVolume>
      <!--opt, int, 低音音量, range:[0,100]-->0
    </bassVolume>
    <balanceVolume>
      <!--opt, int, 左右平衡音量, range:[0,100]-->0
    </balanceVolume>
  </AudioOutVlome>
</AudioOutVolumelist>
<audioOutputInterfaceType>
  <!--opt, enum, 音频输出接口类型, subType:string, [RCA#莲花头输出], desc:用于区分输出接口的类型, RCA输出LR声道同源。1、对讲产品中,原有的音频输出通道和RCA音频输出通道在硬件上存在差异(音质以及输出电路),需要进行区分。2、在音箱产品的定压功放模块中分为单普通输出通道+RCA输出模块 和 双普通输出通道两种模块(具体使用模块由设备自身决定)。因此需要指明一个通道作为RCA通道-->RCA
</audioOutputInterfaceType>
<audioOutputInterfaceMode>
  <!--opt, enum, 音频输出接口配置模式, subType:string, [auto#自适应>manual#手动], desc:用于区分手动配置音频输出类型, 还是设备自适应使用输出。-->auto
</audioOutputInterfaceMode>
<audioBitRate>
  <!--opt, int, 音频编码比特率, range:[32,196], unit:kbps, unitType:带宽速度-->32
</audioBitRate>
</AudioOut>

```

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 响应消息, attr:version{ro, req, string, 版本}-->
  <requestURL>
    <!--ro, req, string, 请求的URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, 状态码, subType:int, [0#OK,1#OK,2#Device Busy,3#Device Error,4#Invalid Operation,5#Invalid XML Format,6#Invalid XML Content,7#Reboot Required]-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, 状态信息, subType:string, [OK#成功,Device Busy#设备忙,Device Error#设备异常,Invalid Operation#无效的操作,Invalid XML Format#无效的XML格式,Invalid XML Content#无效的XML内容,Reboot#设备重启]-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, 详细错误码英文描述, desc:详细错误码的英文描述-->OK
  </subStatusCode>
</ResponseStatus>
```

69.31.2 获取指定音频输出通道参数

Request URL

GET /ISAPI/System/Audio/AudioOut/channels/<audioID>

查询参数

参数名称	参数类型	描述
audioID	string	

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<AudioOut xmlns="http://www.isapi.org/ver20/XMLSchema" version="20">
  <!--ro, opt, object, 单通道音频输出, attr:version{req, string, 协议版本}-->
  <id>
    <!--ro, req, string, 索引-->test
  </id>
  <MixAudioOut>
    <!--ro, opt, object, 混音参数-->
    <enabled>
      <!--ro, req, bool, 使能-->true
    </enabled>
    <modulatorEnbale>
      <!--ro, req, bool, 是否启用相位调制器-->true
    </modulatorEnbale>
    <postFilter>
      <!--ro, req, bool, Post Filter开关-->true
    </postFilter>
    <limitPressure>
      <!--ro, req, bool, 输出压限开关-->true
    </limitPressure>
    <modulatorValue>
      <!--ro, req, int, 相位调制量, range:[0,127], desc:默认10-->1
    </modulatorValue>
    <triggerTime>
      <!--ro, opt, int, 压限触发时间, range:[1,872], unit:ms, unitType:时间, desc:默认为4-->1
    </triggerTime>
    <freeTime>
      <!--ro, opt, int, 压限释放时间, range:[26,46062], unit:ms, unitType:时间, desc:默认为42-->26
    </freeTime>
    <compressThreshold>
      <!--ro, req, int, 压缩门限-->1
    </compressThreshold>
    <compressMode>
      <!--ro, req, enum, 压缩模式, subType:string, [soft#软压缩,hard#硬压缩]-->soft
    </compressMode>
    <compressRate>
      <!--ro, req, int, 压缩率, range:[0,127]-->1
    </compressRate>
    <recoveryGain>
      <!--ro, req, int, 回复增益-->1
    </recoveryGain>
    <outputGain>
      <!--ro, req, int, 输出增益-->1
    </outputGain>
    <outputMute>
      <!--ro, opt, bool, 输出静音-->true
    </outputMute>
  </MixAudioOut>
</AudioOut>
```

```

</outputMute>
<soundQualityHandling>
  <!--ro, opt, bool, 音质处理-->true
</soundQualityHandling>
<outputVolume>
  <!--ro, opt, int, 输出音量, range:[0,100]-->1
</outputVolume>
</MixAudioOut>
<AudioOutVolumelist>
  <!--ro, opt, object, 输出音频声音列表-->
<AudioOutVlome>
  <!--ro, opt, object, 输出音频声音-->
  <type>
    <!--ro, req, enum, 类型, subType:string, [audioOutput#音频输出,audioEncode#音频编码,specific#特定通道]-->audioOutput
  </type>
  <volume>
    <!--ro, req, int, 系统音量, desc:用于表示设备音频输出/输入的系统音量-->1
  </volume>
  <id>
    <!--ro, opt, string, 音频输入通道索引, dep:and,{$.AudioOutCap.AudioOutVolumelist.AudioOutVlome.type,eq,specific}-->test
  </id>
  <name>
    <!--ro, opt, string, 音频输入通道名称, dep:and,{$.AudioOutCap.AudioOutVolumelist.AudioOutVlome.type,eq,specific}-->test
  </name>
  <talkVolume>
    <!--ro, opt, int, 对讲音量, range:[0,100], desc:音量百分比,仅作为对讲音量单独使用,在对讲业务时生效。当系统音量volume节点为0时,对讲音量不可用-->0
  </talkVolume>
  <trebleVolume>
    <!--ro, opt, int, 高音音量, range:[0,100]-->0
  </trebleVolume>
  <bassVolume>
    <!--ro, opt, int, 低音音量, range:[0,100]-->0
  </bassVolume>
  <balanceVolume>
    <!--ro, opt, int, 左右平衡音量, range:[0,100]-->0
  </balanceVolume>
</AudioOutVlome>
</AudioOutVolumelist>
<audioOutputInterFaceType>
  <!--ro, opt, enum, 音频输出接口类型, subType:string, [RCA#莲花头输出], desc:用于区分音频输出接口类型,RCA输出LR声道同源。1、对讲产品中,原有的音频输出通道和RCA音频输出通道在硬件上存在差异(音质以及输出电路),需要进行区分。2、在音箱产品的定压功放模块中分为单普通输出通道+RCA输出模块 和 双普通输出通道两种模块(具体使用模块由设备自身决定)。因此需要指明一个通道作为RCA通道-->RCA
</audioOutputInterFaceType>
<audioOutputInterFaceMode>
  <!--ro, opt, enum, 音频输出接口配置模式, subType:string, [auto#自适应>manual#手动], desc:用于区分手动配置音频输出类型,还是设备自适应使用输出接口。-->auto
</audioOutputInterFaceMode>
<audioBitRate>
  <!--ro, opt, int, 音频编码比特率, range:[32,196], unit:kbps, unitType:带宽速度-->32
</audioBitRate>
</AudioOut>

```

69.31.3 获取指定音频输出通道参数能力

Request URL

GET /ISAPI/System/Audio/AudioOut/channels/<audioID>/capabilities

查询参数

参数名称	参数类型	描述
audioID	string	

请求报文

无

响应报文

```

<?xml version="1.0" encoding="UTF-8"?>

<AudioOutCap xmlns="http://www.isapi.org/ver20/XMLSchema" version="20">
  <!--ro, opt, object, 单通道音频输出能力集, attr:version{req, string, 协议版本}-->
  <id>
    <!--ro, req, string, 索引-->test
  </id>
  <MixAudioOut>
    <!--ro, opt, object, 混音参数-->
    <enabled opt="true,false">
      <!--ro, req, bool, 使能, attr:opt{req, string, 取值范围}-->true
    </enabled>
    <modulatorEnbale opt="true,false">
      <!--ro, req, bool, 是否启用相位调制器, attr:opt{req, string, 取值范围}-->true
    </modulatorEnbale>
    <postFilter opt="false,true" default="true">
      <!--ro, req, bool, Post Filter开关, attr:opt{req, string, 取值范围},default{opt, string, 范围}-->true
    </postFilter>

```

```

<!--ro, req, bool, 输出压限开关, attr:opt{req, string, 取值范围},default{opt, string, 范围}-->true
</limitPressure>
<modulatorValue min="0" max="127" default="10">
  <!--ro, req, int, 相位调制量, range:[0,127], attr:min{req, int, 最小值},max{req, int, 最大值},default{opt, string, 默认值}, desc:默认10-->1
</modulatorValue>
<triggerTime>
  <!--ro, opt, int, 压限触发时间, range:[1,872], unit:ms, unitType:时间, desc:默认为4-->1
</triggerTime>
<freeTime>
  <!--ro, opt, int, 压限释放时间, range:[26,46062], unit:ms, unitType:时间, desc:默认为42-->26
</freeTime>
<compressThreshold opt="1,2,4,8" default="2">
  <!--ro, req, int, 压缩门限, attr:opt{req, string, 取值范围},default{opt, string, 默认值}-->1
</compressThreshold>
<compressMode opt="soft,hard" default="soft">
  <!--ro, req, enum, 压缩模式, subType:string, [soft#软压缩,hard#硬压缩], attr:opt{req, string, 取值范围},default{opt, string, 默认值}-->soft
</compressMode>
<compressRate min="0" max="127">
  <!--ro, req, int, 压缩率, attr:min{req, int, 最小值},max{req, int, 最大值}-->1
</compressRate>
<recoveryGain opt="1,2,4,8" default="2">
  <!--ro, req, int, 回复增益, attr:opt{req, string, 取值范围},default{opt, string, 默认值}-->1
</recoveryGain>
<outputGain def="1">
  <!--ro, req, int, 输出增益, attr:def{opt, string, 默认值}-->1
</outputGain>
<outputMute opt="true,false" def="true">
  <!--ro, opt, bool, 输出静音, attr:opt{req, string, 取值范围},def{req, string, 默认值}-->true
</outputMute>
<soundQualityHandling opt="true,false">
  <!--ro, opt, bool, 音质处理, attr:opt{req, string, 取值范围}-->true
</soundQualityHandling>
<outputVolume>
  <!--ro, opt, int, 输出音量, range:[0,100]-->1
</outputVolume>
</MixAudioOut>
<AudioOutVolumelist>
  <!--ro, opt, object, 输出音频声音列表-->
  <AudioOutVlome>
    <!--ro, opt, object, 输出音频声音-->
    <type opt="audioOutput,audioEncode,specific">
      <!--ro, req, enum, 类型, subType:string, [audioOutput#音频输出,audioEncode#音频编码,specific#特定通道], attr:opt{req, string, 取值范围}-->
    </type>
    <volume min="0" max="127" def="50">
      <!--ro, req, int, 系统音量, attr:min{opt, int, 最小值},max{opt, int, 最大值},def{opt, int, 默认值}, desc:用于表示设备音频输出出口的系统音量-->1
    </volume>
    <id>
      <!--ro, opt, string, 音频输入通道索引, dep:and,{$.AudioOutCap.AudioOutVolumelist.AudioOutVlome.type,eq,specific}-->test
    </id>
    <name>
      <!--ro, opt, string, 音频输入通道名称, dep:and,{$.AudioOutCap.AudioOutVolumelist.AudioOutVlome.type,eq,specific}-->test
    </name>
    <talkVolume min="0" max="15" def="5">
      <!--ro, opt, int, 对讲音量, range:[0,100], attr:min{req, int, 最小值},max{req, int, 最大值},def{opt, int, 默认值}, desc:音量百分比,仅作为对讲音量单独使用,在对讲业务时生效。当系统音量volume节点为0时,对讲音量不可用-->0
    </talkVolume>
    <trebleVolume min="0" max="100">
      <!--ro, opt, int, 高音音量, range:[0,100], attr:min{req, int, 最小值},max{req, int, 最大值}-->0
    </trebleVolume>
    <bassVolume min="0" max="100">
      <!--ro, opt, int, 低音音量, range:[0,100], attr:min{req, int, 最小值},max{req, int, 最大值}-->0
    </bassVolume>
    <balanceVolume min="0" max="100">
      <!--ro, opt, int, 左右平衡音量, range:[0,100], attr:min{req, int, 最小值},max{req, int, 最大值}-->0
    </balanceVolume>
  </AudioOutVlome>
</AudioOutVolumelist>
<audioOutputInterFaceType opt="RCA">
  <!--ro, opt, enum, 音频输出口类型, subType:string, [RCA#莲花头输出], attr:opt{req, string, 取值范围}, desc:用于区分音频输出口类型,RCA输出LR声道同源。1、对讲产品中,原有的音频输出通道和RCA音频输出通道在硬件上存在差异(音质以及输出电路),需要进行区分。2、在音箱产品的定压功放模块中分为单普通输出通道+RCA输出模块和 双普通输出通道两种模块(具体使用模块由设备自身决定)。因此需要指明一个通道作为RCA通道-->RCA
</audioOutputInterFaceType>
<audioOutputInterFaceMode opt="auto">
  <!--ro, opt, enum, 音频输出口配置模式, subType:string, [auto#自适应>manual#手动], attr:opt{req, string, 取值范围}, desc:用于区分手动配置音频输出类型,还是设备自适应使用输出口。-->auto
</audioOutputInterFaceMode>
<audioBitRate min="32" max="196">
  <!--ro, opt, int, 音频编码比特率, range:[32,196], step:1, unit:kbps, unitType:带宽速度, attr:min{req, int, 最小值},max{req, int, 最大值}-->32
</audioBitRate>
</AudioOutCap>

```

69.32 语音对讲

69.32.1 关闭语音对讲

Request URL

PUT /ISAPI/System/TwoWayAudio/channels/<twoWayAudioChannelID>/close

查询参数

参数名称	参数类型	描述
twoWayAudioChannelID	string	语音对讲通道ID

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 响应消息, attr:version{ro, req, string, 版本}-->
  <requestURL>
    <!--ro, req, string, 请求的URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, 状态码, subType:int, [0#OK,1#OK,2#Device Busy,3#Device Error,4#Invalid Operation,5#Invalid XML Format,6#Invalid XML Content,7#Reboot Required]-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, 状态信息, subType:string, [OK#成功,Device Busy#设备忙,Device Error#设备异常,Invalid Operation#无效的操作,Invalid XML Format#无效的XML格式,Invalid XML Content#无效的XML内容,Reboot#设备重启]-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, 详细错误码英文描述, desc:详细错误码的英文描述-->OK
  </subStatusCode>
</ResponseStatus>
```

69.32.2 开启语音对讲

Request URL

PUT /ISAPI/System/TwoWayAudio/channels/<twoWayAudioChannelID>/open

查询参数

参数名称	参数类型	描述
twoWayAudioChannelID	string	语音对讲通道号

请求报文

```
<?xml version="1.0" encoding="UTF-8"?>

<TwoWayAudio xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--opt, object, 本次对讲参数, attr:version{req, string, 协议版本}-->
  <audioLevel>
    <!--opt, int, 音频播放级别, range:[0,15], desc:0~15级别从低到高-->0
  </audioLevel>
  <microphoneVolume>
    <!--opt, int, 输出音量-->100
  </microphoneVolume>
  <audioCompressionType>
    <!--req, enum, 音频输出编码类型, subType:string,
    [G.711aLaw#G.711aLaw, G.711uLaw#G.711uLaw, G.726#G.726, G.729#G.729, G.729a#G.729a, G.729b#G.729b, PCM#PCM, MP3#MP3, AC3#AC3, AAC#AAC, ADPCM#ADPCM, MP2L2#MP2L2, Opus#Opus, G.722.1#G.722.1]-->G.711aLaw
  </audioCompressionType>
  <audioSamplingRate>
    <!--opt, float, 音频采样率, unit:kHz-->48.00
  </audioSamplingRate>
</TwoWayAudio>
```

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<TwoWayAudioSession xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, 开启语音对讲成功时返回的结果, attr:version{opt, string, 协议版本}-->
  <sessionId>
    <!--ro, req, string, 语音对讲链路ID-->1
  </sessionId>
</TwoWayAudioSession>
```

69.32.3 获取所有语音对讲通道音频参数

Request URL

GET /ISAPI/System/TwoWayAudio/channels

查询参数

无

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<TwoWayAudioChannelList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, array, 语音对讲通道音频参数配置, subType:object, attr:version(opt, string, 版本)-->
  <TwoWayAudioChannel>
    <!--ro, opt, object, 语音对讲通道音频配置-->
    <id>
      <!--ro, req, string, 音频通道ID-->1
    </id>
    <enabled>
      <!--ro, req, bool, 使能开关-->true
    </enabled>
    <audioCompressionType>
      <!--ro, req, enum, 音频输出编码类型, subType:string,
      [G.711aLaw#G.711aLaw,G.711uLaw#G.711uLaw,G.726#G.726,G.729#G.729,G.729a#G.729a,G.729b#G.729b,PCM#PCM,MP3#MP3,AC3#AC3,AAC#AAC,ADPCM#ADPCM,MP2L2#MP2L2,Opus#Opus]-->G.711aLaw
    </audioCompressionType>
    <audioInboundCompressionType>
      <!--ro, opt, enum, 音频输入编码类型, subType:string,
      [G.711aLaw#G.711aLaw,G.711uLaw#G.711uLaw,G.726#G.726,G.729#G.729,G.729a#G.729a,G.729b#G.729b,PCM#PCM,MP3#MP3,AC3#AC3,AAC#AAC,ADPCM#ADPCM]-->G.711aLaw
    </audioInboundCompressionType>
    <speakerVolume>
      <!--ro, opt, int, 输入音量, range:[1,100]-->100
    </speakerVolume>
    <microphoneVolume>
      <!--ro, opt, int, 输出音量, range:[1,100]-->100
    </microphoneVolume>
    <noisereduce>
      <!--ro, opt, bool, 环境噪声过滤使能-->true
    </noisereduce>
    <audioBitRate>
      <!--ro, opt, int, 音频帧率, unit:kbs-->100
    </audioBitRate>
    <audioInputType>
      <!--ro, opt, enum, 音频输入类型, subType:string, [MicIn#麦克风搜集,LineIn#音频线输入,selfAdaptive#自适应(设备插入状态自行配置mic_in还是Linein),wirelessPickUp#无线拾音器]-->MicIn
    </audioInputType>
    <associateVideoInputs>
      <!--ro, opt, object, 关联视频通道-->
      <enabled>
        <!--ro, req, bool, 关联通道使能-->true
      </enabled>
      <videoInputChannelList>
        <!--ro, req, array, 关联视频通道列表, subType:object-->
        <videoInputChannelID>
          <!--ro, opt, string, 关联的视频通道-->1
        </videoInputChannelID>
      </videoInputChannelList>
    </associateVideoInputs>
    <audioSamplingRate>
      <!--ro, opt, float, 音频采样率, unit:kHz-->48.00
    </audioSamplingRate>
    <lineOutForbidden>
      <!--ro, opt, bool, 是否不支持音频输出;该节点为反向能力节点;设备不返回该节点或该节点返回false时表示设备支持音频输出;设备返回true时表示设备不支持音频输出-->true
    </lineOutForbidden>
    <muteDuringPanTilt>
      <!--ro, opt, bool, 运动时消音使能-->true
    </muteDuringPanTilt>
    <audioOutputType>
      <!--ro, opt, enum, 音频输出类型, subType:string, [Close#关闭,LineOut#音频线输出,Speaker#扬声器输出,selfAdaptive#自适应(设备插入状态自行配置LineOut还是Speaker),LineOut_Speaker#音频线和扬声器同时输出]-->LineOut
    </audioOutputType>
    <matrixPickUp>
      <!--ro, opt, bool, 是否使用有线矩阵拾音器 不支持有线矩阵拾音器不返回该节点 有线矩阵拾音器依赖于音频输入类型选择为Linein-->true
    </matrixPickUp>
  </TwoWayAudioChannel>
</TwoWayAudioChannelList>
```

69.32.4 获取指定语音对讲通道音频参数能力

Request URL

GET /ISAPI/System/TwoWayAudio/channels/<twoWayAudioChannelID>/capabilities

查询参数

参数名称	参数类型	描述
twoWayAudioChannelID	string	语音对讲通道号

请求报文

无

响应报文

湖南云舟电力科技有限公司
有效期：2024-08-15

```

<?xml version="1.0" encoding="UTF-8"?>

<TwoWayAudioChannel xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, 语音对讲通道音频配置能力, attr:version{opt, string, 版本号}-->
  <id>
    <!--ro, req, string, 音频通道ID-->1
  </id>
  <enabled>
    <!--ro, req, bool, 使能开关-->true
  </enabled>
  <audioCompressionType opt="G.711alaw,G.711ulaw,G.726,G.729,G.729a,G.729b,PCM,MP3,AC3,AAC,ADPCM,MP2L2,Opus,G.722.1">
    <!--ro, req, enum, 音频输出编码类型, subType:string,
    [G.711alaw#G.711alaw,G.711ulaw#G.711ulaw,G.726#G.726,G.729#G.729,G.729a#G.729a,G.729b#G.729b,PCM#PCM,MP3#MP3,AC3#AC3,AAC#AAC,ADPCM#ADPCM,MP2L2#MP2L2,Opus#Opus,G.722.1#G.722.1], attr:opt{opt, string, 范围}-->G.711alaw
  </audioCompressionType>
  <audioInboundCompressionType opt="G.711alaw,G.711ulaw,G.726,G.729,G.729a,G.729b,PCM,MP3,AC3,AAC,ADPCM">
    <!--ro, req, enum, 音频输入编码类型, subType:string,
    [G.711alaw#G.711alaw,G.711ulaw#G.711ulaw,G.726#G.726,G.729#G.729,G.729a#G.729a,G.729b#G.729b,PCM#PCM,MP3#MP3,AC3#AC3,AAC#AAC,ADPCM#ADPCM], attr:opt{opt, string, 范围}-->G.711alaw
  </audioInboundCompressionType>
  <speakerVolume min="0" max="100">
    <!--ro, opt, int, 输入音量, attr:min{opt, int, 最小值},max{opt, int, 最大值}-->100
  </speakerVolume>
  <microphoneVolume min="0" max="100">
    <!--ro, opt, int, 输出音量, attr:min{opt, int, 最小值},max{opt, int, 最大值}-->100
  </microphoneVolume>
  <noisereduce opt="true,false">
    <!--ro, opt, bool, 环境噪声过滤使能, attr:opt{opt, string, 范围}-->true
  </noisereduce>
  <audioBitRate min="1" max="100">
    <!--ro, opt, int, 音频帧率, unit:kbs, attr:min{opt, int, 最小值},max{opt, int, 最大值}-->100
  </audioBitRate>
  <audioInputType opt="MicIn,LineIn,selfAdaptive,wirelessPickUp">
    <!--ro, opt, enum, 音频输入类型, subType:string, [MicIn#麦克风搜集,LineIn#音频线输入,selfAdaptive#自适应(设备插入状态自行配置mic in还是Line in),wirelessPickUp#无线拾音器], attr:opt{opt, string, 范围}-->MicIn
  </audioInputType>
  <associateVideoInputs>
    <!--ro, opt, object, 关联视频通道-->
    <enabled opt="true,false">
      <!--ro, req, bool, 关联通道使能, attr:opt{opt, string, 范围}-->true
    </enabled>
    <videoInputChannelList>
      <!--ro, req, array, 关联视频通道列表, subType:object-->
      <videoInputChannelID min="1" max="2048">
        <!--ro, opt, string, 关联的视频通道, attr:min{opt, string, 最小值},max{opt, string, 最大值}-->1
      </videoInputChannelID>
    </videoInputChannelList>
  </associateVideoInputs>
  <audioSamplingRate min="1.00" max="48.00">
    <!--ro, opt, float, 音频采样率, unit:khz, attr:min{opt, float, 最小值},max{opt, float, 最大值}-->48.00
  </audioSamplingRate>
  <lineOutForbidden opt="true,false">
    <!--ro, opt, bool, 是否不支持音频输出;该节点为双向能力节点;设备不返回该节点或该节点返回false时表示设备支持音频输出;设备返回true时表示设备不支持音频输出, attr:opt{opt, string, 范围}-->true
  </lineOutForbidden>
  <muteDuringPanTilt opt="true,false">
    <!--ro, opt, bool, 运动时消音, attr:opt{opt, string, 运动时消音使能}-->true
  </muteDuringPanTilt>
  <audioOutputType opt="Close,LineOut,Speaker,selfAdaptive,LineOut_Speaker" def="LineOut">
    <!--ro, opt, enum, 音频输出类型, subType:string, [Close#关闭,LineOut#音频线输出,Speaker#扬声器输出,selfAdaptive#自适应(设备插入状态自行配置LineOut还是Speaker),LineOut_Speaker#音频线和扬声器同时输出], attr:opt{opt, string, 范围},def{opt, string, 默认值}-->LineOut
  </audioOutputType>
  <isSupportEchoCancellation>
    <!--ro, opt, bool, 是否支持回声消除功能-->true
  </isSupportEchoCancellation>
  <timeOut min="10" max="600">
    <!--ro, opt, int, 语音对讲时超时时间, attr:min{req, int, 最小值},max{req, int, 最大值}, desc:Web向雷达设备主动发起单向语音对讲时, 由于杂音的存在, 即使Web端人员没有说话, 也会有杂音数据传输到雷达设备端, 此时, 若配置了该字段, 则到达一定时间后, 设备端主动断开连接, 无论Web端人员是否说话结束.-->300
  </timeOut>
  <matrixPickUp opt="true,false">
    <!--ro, opt, bool, 是否使用有线矩阵拾音器 不支持有线矩阵拾音器不返回该节点 有线矩阵拾音器依赖于音频输入类型选择为Linein, attr:opt{opt, string, 是否使用海康矩阵式拾音器}-->false
  </matrixPickUp>
  <isSupportRestore>
    <!--ro, opt, bool, 是否支持恢复语音对讲通道音频参数, desc:对应接口/ISAPI/System/TwoWayAudio/channels/<twoWayAudioChannelID>/restore?format=json-->true
  </isSupportRestore>
</TwoWayAudioChannel>

```

69.32.5 获取所有语音对讲通道音频参数能力

Request URL

GET /ISAPI/System/TwoWayAudio/channels/capabilities

查询参数

无

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<TwoWayAudioChannelList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 语音对讲所有通道音频配置能力, attr:version{opt, string, 协议版本}-->
  <TwoWayAudioChannel>
    <!--ro, opt, object, 语音对讲通道音频配置能力-->
    <id>
      <!--ro, req, string, 音频通道ID-->1
    </id>
    <enabled>
      <!--ro, req, bool, 使能开关-->true
    </enabled>
    <audioCompressionType opt="G.711alaw,G.711ulaw,G.726,G.729,G.729a,G.729b,PCM,MP3,AC3,AAC,ADPCM,MP2L2,Opus">
      <!--ro, req, enum, 音频输出编码类型, subType:string,
      [G.711aLaw#G.711aLaw,G.711uLaw#G.711uLaw,G.726#G.726,G.729#G.729,G.729a#G.729a,G.729b#G.729b,PCM#PCM,MP3#MP3,AC3#AC3,AAC#AAC,ADPCM#ADPCM,MP2L2#MP2L2,Opus#Opus], attr:opt{opt, string, 取值范围}-->G.711alaw
    </audioCompressionType>
    <audioInboundCompressionType opt="G.711alaw,G.711ulaw,G.726,G.729,G.729a,G.729b,PCM,MP3,AC3,AAC,ADPCM">
      <!--ro, opt, enum, 音频输入编码类型, subType:string,
      [G.711aLaw#G.711aLaw,G.711uLaw#G.711uLaw,G.726#G.726,G.729#G.729,G.729a#G.729a,G.729b#G.729b,PCM#PCM,MP3#MP3,AC3#AC3,AAC#AAC,ADPCM#ADPCM], attr:opt{opt, string, 取值范围}-->G.711alaw
    </audioInboundCompressionType>
    <speakerVolume min="0" max="100">
      <!--ro, opt, int, 输入音量, attr:min{opt, int, 最小值},max{opt, int, 最大值}-->100
    </speakerVolume>
    <microphoneVolume min="0" max="100">
      <!--ro, opt, int, 输出音量, attr:min{opt, int, 最小值},max{opt, int, 最大值}-->100
    </microphoneVolume>
    <noisereduce opt="true,false">
      <!--ro, opt, bool, 环境噪声过滤使能, attr:opt{opt, string, 取值范围}-->true
    </noisereduce>
    <audioBitRate min="1" max="100">
      <!--ro, opt, int, 音频频率, unit:kbs, attr:min{opt, string, 最小值},max{opt, string, 最大值}-->100
    </audioBitRate>
    <audioInputType opt="MicIn,LineIn,selfAdaptive,wirelessPickUp">
      <!--ro, opt, enum, 音频输入类型, subType:string, [MicIn#麦克风搜集,LineIn#音频线输入,selfAdaptive#自适应(设备插入状态自行配置mic in还是Line in),wirelessPickUp#无线拾音器], attr:opt{opt, string, 取值范围}-->MicIn
    </audioInputType>
    <associateVideoInputs>
      <!--ro, opt, object, 关联视频通道-->
      <enabled opt="true,false">
        <!--ro, req, bool, 关联通道使能, attr:opt{opt, string, 取值范围}-->true
      </enabled>
      <videoInputChannelList>
        <!--ro, req, array, 关联视频通道列表, subType:object-->
        <videoInputChannelID min="1" max="2048">
          <!--ro, opt, string, 关联的视频通道, attr:min{opt, string, 最小值},max{opt, string, 最大值}-->1
        </videoInputChannelID>
      </videoInputChannelList>
    </associateVideoInputs>
    <audioSamplingRate min="1.00" max="48.00">
      <!--ro, opt, float, 音频采样率, unit:khz, attr:min{opt, float, 最小值},max{opt, float, 最大值}-->48.00
    </audioSamplingRate>
    <lineOutForbidden opt="true,false">
      <!--ro, opt, bool, 是否不支持音频输出;该节点为反向能力节点,设备不返回该节点或该节点返回false时表示设备支持音频输出;设备返回true时表示设备不支持音频输出, attr:opt{opt, string, 取值范围}-->true
    </lineOutForbidden>
    <muteDuringPanTilt opt="true,false">
      <!--ro, opt, bool, 运动时静音, attr:opt{opt, string, 运动时静音使能}-->true
    </muteDuringPanTilt>
    <audioOutputType opt="Close,LineOut,Speaker,selfAdaptive,LineOut_Speaker" def="LineOut">
      <!--ro, opt, enum, 音频输出类型, subType:string, [Close#关闭,LineOut#音频线输出,Speaker#扬声器输出,selfAdaptive#自适应,LineOut_Speaker#音频线和扬声器同时输出], attr:opt{opt, string, 取值范围},def{opt, string, 默认值}-->LineOut
    </audioOutputType>
    <isSupportEchoCancellation>
      <!--ro, opt, bool, 是否支持回声消除功能-->true
    </isSupportEchoCancellation>
    <matrixPickUp opt="true,false">
      <!--ro, opt, bool, 是否使用有线矩阵拾音器 不支持有线矩阵拾音器不返回该节点 有线矩阵拾音器依赖于音频输入类型选择为Linein, attr:opt{opt, string, 是否使用有线矩阵拾音器}-->false
    </matrixPickUp>
    <isSupportRestore>
      <!--ro, opt, bool, 是否支持恢复语音对讲通道音频参数, desc:对应接口/ISAPI/System/TwoWayAudio/channels/<twoWayAudioChannelID>/restore?format=json-->true
    </isSupportRestore>
  </TwoWayAudioChannel>
</TwoWayAudioChannelList>
```

69.32.6 获取指定语音对讲通道音频参数

Request URL

GET /ISAPI/System/TwoWayAudio/channels/<twoWayAudioChannelID>

查询参数

参数名称	参数类型	描述
twoWayAudioChannelID	string	语音对讲通道号

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<TwoWayAudioChannel xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, 语音对讲通道音频配置, attr:version{opt, string, 协议版本}-->
  <id>
    <!--ro, req, string, 音频通道ID-->1
  </id>
  <enabled>
    <!--ro, req, bool, 使能开关-->true
  </enabled>
  <audioCompressionType>
    <!--ro, req, enum, 音频输出编码类型, subType:string,
    [G.711aLaw#G.711aLaw, G.711uLaw#G.711uLaw, G.726#G.726, G.729#G.729, G.729a#G.729a, G.729b#G.729b, PCM#PCM, MP3#MP3, AC3#AC3, AAC#AAC, ADPCM#ADPCM, MP2L2#MP2L2, Opus#Opus, G.722.1#G.722.1]-->G.711aLaw
  </audioCompressionType>
  <audioInboundCompressionType>
    <!--ro, opt, enum, 音频输入编码类型, subType:string,
    [G.711aLaw#G.711aLaw, G.711uLaw#G.711uLaw, G.726#G.726, G.729#G.729, G.729a#G.729a, G.729b#G.729b, PCM#PCM, MP3#MP3, AC3#AC3, AAC#AAC, ADPCM#ADPCM]-->G.711aLaw
  </audioInboundCompressionType>
  <speakerVolume>
    <!--ro, opt, int, 输入音量, range:[1,100]-->100
  </speakerVolume>
  <microphoneVolume>
    <!--ro, opt, int, 输出音量, range:[1,100]-->100
  </microphoneVolume>
  <noisereduce>
    <!--ro, opt, bool, 环境噪声过滤使能-->true
  </noisereduce>
  <audioBitRate>
    <!--ro, opt, int, 音频帧率, unit:kbs-->100
  </audioBitRate>
  <audioInputType>
    <!--ro, opt, enum, 音频输入类型, subType:string, [MicIn#麦克风搜集, LineIn#音频线输入, selfAdaptive#自适应(设备插入状态自行配置mic in还是Line in), wirelessPickUp#无线拾音器]-->MicIn
  </audioInputType>
  <associateVideoInputs>
    <!--ro, opt, object, 关联视频通道-->
    <enabled>
      <!--ro, req, bool, 关联通道使能-->true
    </enabled>
    <videoInputChannelList>
      <!--ro, req, array, 关联视频通道列表, subType:object-->
      <videoInputChannelID>
        <!--ro, opt, string, 关联的视频通道-->1
      </videoInputChannelID>
    </videoInputChannelList>
    </associateVideoInputs>
  <audioSamplingRate>
    <!--ro, opt, float, 音频采样率, unit:kHz-->48.00
  </audioSamplingRate>
  <lineOutForbidden>
    <!--ro, opt, bool, 是否不支持音频输出;该节点为反向能力节点;设备不返回该节点或该节点返回false时表示设备支持音频输出;设备返回true时表示设备不支持音频输出-->true
  </lineOutForbidden>
  <micInForbidden>
    <!--ro, opt, bool, 是否不支持音频输入;该节点为反向能力节点;设备不返回该节点或该节点返回false时表示设备支持音频输入;设备返回true时表示设备不支持音频输入-->true
  </micInForbidden>
  <muteDuringPanTilt>
    <!--ro, opt, bool, 运动时消音-->true
  </muteDuringPanTilt>
  <audioOutputType>
    <!--ro, opt, enum, 音频输出类型, subType:string, [Close#关闭, LineOut#音频线输出, Speaker#扬声器输出, selfAdaptive#自适应(设备插入状态自行配置LineOut还是Speaker), LineOut_Speaker#音频线和扬声器同时输出]->LineOut
  </audioOutputType>
  <timeOut>
    <!--ro, opt, int, 语音对讲时超时时间, range:[10,600], unit:s, unitType:时间, desc:Web向雷达设备主动发起单向语音对讲时, 由于杂音的存在, 即使Web端人员没有说话, 也会有杂音数据传输到雷达设备端, 此时, 若配置了该字段, 则到达一定时间后, 设备端主动断开连接, 无论Web端人员是否说话结束.-->10
  </timeOut>
  <matrixPickUp>
    <!--ro, opt, bool, 是否使用有线矩阵拾音器 有线矩阵拾音器依赖于音频输入类型选择为Linein-->true
  </matrixPickUp>
</TwoWayAudioChannel>
```

69.33 身份证管理

69.33.1 配置身份证黑名单下发

Request URL

PUT /ISAPI/AccessControl/IDBlackListCfg?security=<security>&iv=<iv>

查询参数

参数名称	参数类型	描述
security	string	没有security表示数据不加密；security=1，表示报文中的name、addr、IDNum、nation字段AES128加密；security=2，表示报文中的name、addr、IDNum、nation字段AES256加密
iv	string	iv表示初始化向量，在security=1或者2时为必须参数;随机字符串，长度限制为32

请求报文

```
<?xml version="1.0" encoding="UTF-8"?>

<IDBlackListCfg xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, 身份证黑名单, attr:version{req, string, 协议版本}-->
  <blackListValid>
    <!--req, enum, 身份证黑名单是否有效, subType:int, [0#无效,1#有效], desc:用于按身份证号码删除身份证黑名单, 该字段为0时代表删除-->1
  </blackListValid>
  <IDCardInfo>
    <!--opt, object, 身份证信息-->
    <name>
      <!--opt, string, 姓名, range:[1,32]-->test
    </name>
    <birth>
      <!--opt, date, 出生日期-->1990-02-24
    </birth>
    <addr>
      <!--opt, string, 住址, range:[1,32]-->test
    </addr>
    <IDNum>
      <!--ro, req, string, 身份证号码-->542137199002241111
    </IDNum>
    <issuingAuthority>
      <!--opt, string, 签发机关, range:[1,32]-->test
    </issuingAuthority>
    <startDate>
      <!--opt, date, 有效开始日期-->1970-01-01
    </startDate>
    <endDate>
      <!--opt, date, 有效截止日期, dep:and,{$.IDBlackListCfg.IDCardInfo.termOfValidity,ue,true}-->1977-01-01
    </endDate>
    <termOfValidity>
      <!--opt, bool, 是否长期有效, desc:false-否, true-是(有效截止日期无效)-->true
    </termOfValidity>
    <sex>
      <!--opt, enum, 性别, subType:string, [male#男,female#女]-->male
    </sex>
    <nation>
      <!--opt, enum, 民族, subType:int, [1#汉,2#蒙古,3#回,4#藏,5#维吾尔,6#苗,7#彝,8#壮,9#布依,10#朝鲜,11#满,12#侗,13#瑶,14#白,15#土家,16#哈尼,17#哈萨克,18#傣,19#黎,20#傈僳,21#佤,22#侳,23#高山,24#拉祜,25#水,26#东乡,27#纳西,28#景颇,29#柯尔克孜,30#土,31#达斡尔,32#仫佬,33#羌,34#布朗,35#撒拉,36#毛南,37#仫佬,38#锡伯,39#阿昌,40#普米,41#塔吉克,42#怒,43#乌孜别克,44#俄罗斯,45#鄂温克,46#德昂,47#保安,48#裕固,49#京,50#塔塔尔,51#独龙,52#鄂伦春,53#赫哲,54#门巴,55#珞巴,56#基诺,57#未知]-->1
    </nation>
  </IDCardInfo>
</IDBlackListCfg>
```

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 响应消息, attr:version{ro, req, string, 版本}-->
  <requestURL>
    <!--ro, req, string, 请求的URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, 状态码, subType:int, [0#OK,1#OK,2#Device Busy,3#Device Error,4#Invalid Operation,5#Invalid XML Format,6#Invalid XML Content,7#Reboot Required]-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, 状态信息, subType:string, [OK#成功,Device Busy#设备忙,Device Error#设备异常,Invalid Operation#无效的操作,Invalid XML Format#无效的XML格式,Invalid XML Content#无效的XML内容,Reboot#设备重启]-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, 详细错误码英文描述, desc:详细错误码的英文描述-->OK
  </subStatusCode>
</ResponseStatus>
```

69.33.2 查询身份证事件

Request URL

POST /ISAPI/AccessControl/IDCardInfoEvent?format=json

查询参数

无

请求报文

```
{
  "IDCardInfoEventCond": {
    /*req, object, 身份证事件查询条件*/
    "searchID": "test",
    /*req, string, 搜索记录唯一标识, desc:用来确认上层客户端是否为同一个(倘若为同一个,设备记录内存,下次搜索加快速度)*/
    "searchResultPosition": 0,
    /*req, int, 查询结果在结果列表中的起始位置, desc:当记录条数很多时,一次查询不能获取所有的记录,下一次查询时指定位置可以查询后面的记录(若设备支持的最大totalMatches为M个,但是当前设备已存储的totalMatches为N个(N<=M),则该字段的合法范围为0~N-1)*/
    "maxResults": 30,
    /*req, int, 本次协议调用可获取的最大记录数, desc:如maxResults值大于设备能力集返回的范围,则设备按照能力集最大值返回,设备不进行报错*/
    "majorEventType": 1,
    /*req, int, 主类型, desc:参考事件上传宏定义, 0-全部(此处传递的是10进制数据,不能传递16进制,如传递1代表0x1(报警主类型))*/
    "subEventType": 1024,
    /*req, int, 次类型, dep:and,{$.IDCardInfoEventCond.majorEventType,eq,任意值}, desc:参考事件上传宏定义, 0-全部(此处传递的是10进制数据,不能传递16进制,如传递1024代表0x400(防区短路报警次类型))*/
    "major": 1,
    /*opt, int, 主类型, desc:部分老设备不支持majorEventType,仅支持major(需要根据能力集区分,新设备统一使用majorEventType)*/
    "minor": 1024,
    /*opt, int, 次类型, dep:and,{$.IDCardInfoEventCond.majorEventType,eq,任意值}, desc:部分老设备不支持subEventType,仅支持minor(需要根据能力集区分,新设备统一使用subEventType)*/
    "startTime": "1970-01-01T00:00:00+08:00",
    /*opt, datetime, 开始时间(UTC时间)*/
    "endTime": "1970-01-01T00:00:00+08:00",
    /*opt, datetime, 结束时间(UTC时间)*/
    "picEnable": true,
    /*opt, bool, 是否带图片*/
    "beginSerialNo": 1,
    /*opt, int, 起始流水号*/
    "endSerialNo": 1,
    /*opt, int, 结束流水号*/
    "isAbnormalTemperature": true,
    /*opt, bool, 人脸温度是否异常*/
  }
}
```

响应报文

```
{
  "IDCardInfoEvent": {
    /*ro, req, object, 身份证事件*/
    "searchID": "test",
    /*ro, req, string, 搜索记录唯一标识, desc:用来确认上层客户端是否为同一个(倘若为同一个,设备记录内存,下次搜索加快速度)*/
    "responseStatusStrg": "OK",
    /*ro, req, enum, 查询状态字符串描述, subType:string, [OK#查询结束,MORE#还有数据等待查询,NO_MATCH#没有匹配数据]*/
    "numOfMatches": 1,
    /*ro, req, int, 本次返回的记录条数*/
    "totalMatches": 1,
    /*ro, req, int, 符合条件的记录总条数*/
    "InfoList": [
      /*ro, opt, array, 事件信息列表, subType:object*/
      {
        "deviceName": "test",
        /*ro, opt, string, 设备名称*/
      }
    ]
  }
}
```

```

/*ro, opt, string, 设备名称*/
"majorEventType": 1,
/*ro, req, int, 报警主类型, desc:参考宏定义（此处传递的是10进制数据, 不能传递16进制, 如传递1代表0x1（报警主类型））*/
"subEventType": 1,
/*ro, req, int, 报警次类型, dep:and,{$.IDCardInfoEvent.InfoList[*].majorEventType,eq,任意值}, desc:参考宏定义（此处传递的是10进制数据, 不能传递16进制, 如传递1024代表0x400（防区短路报警次类型））*/
"inductiveEventType": "test",
/*ro, opt, string, 归纳事件类型, desc:后端设备定义, 门禁设备暂时用不到*/
"netUser": "test",
/*ro, opt, string, 网络操作的用户名*/
"remoteHostAddr": "test",
/*ro, opt, string, 远程主机地址*/
"cardType": 1,
/*ro, opt, enum, 卡类型, subType:int, [1#普通卡,2#残疾人卡,3#黑名单卡,4#巡更卡,5#胁迫卡,6#超级卡,7#来宾卡,8#解除卡]*/
"cardReaderNo": 1,
/*ro, opt, int, 读卡器编号*/
"doorNo": 1,
/*ro, opt, int, 门编号（楼层编号）*/
"deviceNo": 1,
/*ro, opt, int, 设备编号*/
"serialNo": 1,
/*ro, opt, int, 事件流水号*/
"QRCodeInfo": "test",
/*ro, opt, string, 二维码信息*/
"thermometryUnit": "celsius",
/*ro, opt, enum, 测温单位, subType:string, [celsius#摄氏度（默认,fahrenheit#华氏度,kelvin#开尔文）*/
"currTemperature": 1.0,
/*ro, opt, float, 人脸温度, desc:精确到小数点后一位*/
"isAbnormalTemperature": true,
/*ro, opt, bool, 人脸测温是否温度异常, desc:true-是, false-否*/
"RegionCoordinates": {
/*ro, opt, object, 人脸温度坐标*/
"positionX": 1,
/*ro, opt, int, X坐标 , range:[0,1000], desc:归一化坐标0-1000*/
"positionY": 1
/*ro, opt, int, Y坐标 , range:[0,1000], desc:归一化坐标0-1000*/
},
"mask": "unknown",
/*ro, opt, enum, 是否戴口罩, subType:string, [unknown#未知,yes#戴口罩,no#不戴口罩]*/
"frontSerialNo": 1,
/*ro, opt, int, 上一条事件流水号, desc:若设备没返回该字段, 平台根据serialNo判断是否丢失事件; 若设备返回该字段, 平台根据该字段和serialNo字段共同判断是否丢失事件,主要用于解决报警订阅后导致serialNo不连续的情况*/
"IDCardInfo": {
/*ro, opt, object, 身份证信息*/
"name": "test",
/*ro, opt, string, 姓名*/
"sex": "male",
/*ro, opt, enum, 性别, subType:string, [male#男,female#女]*/
"birth": "1970-01-01",
/*ro, opt, date, 出生日期*/
"addr": "test",
/*ro, opt, string, 住址*/
"IDCardNo": "test",
/*ro, opt, string, 身份证号码*/
"issuingAuthority": "test",
/*ro, opt, string, 签发机关*/
"startDate": "test",
/*ro, opt, string, 有效期起始时间*/
"endDate": "test",
/*ro, opt, string, 有效期截止时间*/
"nation": 1,
/*ro, opt, int, 民族, range:[1,57], desc:1-"汉", 2-"蒙古", 3-"回",4-"藏",5-"维吾尔",6-"苗",7-"彝",8-"壮",9-"布依",10-"朝鲜",11-"满",12-"侗",13-"瑶",14-"白",15-"土家",16-"哈尼",17-"哈萨克",18-"傣",19-"黎",20-"傈僳",21-"佤",22-"畲",23-"高山",24-"拉祜",25-"水",26-"东乡",27-"纳西",28-"景颇",29-"柯尔克孜",30-"土",31-"达斡尔",32-"仫佬",33-"羌",34-"布朗",35-"撒拉",36-"毛南",37-"仡佬",38-"锡伯",39-"阿昌",40-"普米",41-"塔吉克",42-"怒",43-"乌孜别克",44-"俄罗斯",45-"鄂温克",46-"德昂",47-"保安",48-"裕固",49-"京",50-"塔塔尔",51-"独龙",52-"鄂伦春",53-"赫哲",54-"门巴",55-"珞巴",56-"基诺",57及之后为未知*/
"isLongTermEffective": false
/*ro, opt, bool, 是否长期有效*/
},
"capturePicUrl": "test",
/*ro, opt, string, 抓拍图片url*/
"IDCardPic": "test",
/*ro, opt, string, 身份证图片url*/
"visibleLightPicUrl": "test",
/*ro, opt, string, 可见光图片url*/
"thermalPicUrl": "test",
/*ro, opt, string, 热成像图片url*/
"helmet": "unknown",
/*ro, opt, enum, 是否戴安全帽, subType:string, [unknown#未知,yes#戴安全帽,no#不戴安全帽]*/
"dateTime": "1970-01-01T00:00:00+08:00",
/*ro, req, datetime, 报警触发时间（UTC时间）, desc:最大长度为32*/
"HealthInfo": {
/*ro, opt, object, 健康信息*/
"healthCode": 1,
/*ro, opt, enum, 健康码状态, subType:int, [0#未请求,1#未申领,2#绿码,3#黄码,4#红码,5#无此人员,6#其他错误信息（如接口异常导致查询失败）,7#查询健康码超时]*/
"NAIDCode": 1,
/*ro, opt, enum, 核酸检测结果, subType:int, [0#未查询到核酸检测结果,1#核酸检测阴性（代表正常）,2#核酸检测阳性（代表确诊）,3#核酸检测有效期已过,4#查询核酸结果失败]*/
"NAIDMsg": "test",
/*ro, opt, string, 核酸检测信息, range:[0,64], desc:空字符串表示查询到核酸检测信息失败*/
"travelCode": 1,
/*ro, opt, enum, 行程编码, subType:int, [0#14天内一直在当地,1#14天内离开过当地,2#14天内到过疫区,3#其他,4#查询行程信息失败]*/
"travelInfo": "test",
/*ro, opt, string, 行程信息, desc:空字符串表示查询到行程信息失败*/

```

```

        "vaccineStatus": 1,
        /*ro, opt, enum, 打疫苗状态, subType:int, [0#未打疫苗,1#部分注射疫苗,2#已完成疫苗,3#查询疫苗信息失败,4#已完成加强针疫苗]*/
        "vaccineNum": 1,
        /*ro, opt, int, 打疫苗针数, step:1*/
        "vaccineMsg": "test",
        /*ro, opt, string, 打疫苗信息, range:[0,64], desc:空字符串表示查询到疫苗信息失败*/
        "ANTCode": 1,
        /*ro, opt, enum, 抗原检测状态, subType:int, [0#未查询到抗原检测结果,1#抗原检测阴性,2#抗原检测阳性,3#抗原检测有效期已过,4#抗原检测无效]*/
        "ANTMsg": "test"
        /*ro, opt, string, 抗原检测信息, range:[0,64], desc:空字符串表示查询到抗原检测信息失败*/
    },
    "FaceRect": {
        /*ro, opt, object, 人脸矩形框, desc:屏幕左上角为坐标原点*/
        "height": 1.000,
        /*ro, req, float, 高度, range:[0.000,1.000]*/
        "width": 1.000,
        /*ro, req, float, 宽度, range:[0.000,1.000]*/
        "x": 0.000,
        /*ro, req, float, 区域左上角顶点的水平坐标, range:[0.000,1.000]*/
        "y": 0.000
        /*ro, req, float, 区域左上角顶点的垂直坐标, range:[0.000,1.000]*/
    },
    "currentAuthenticationTimes": 1,
    /*ro, opt, int, 当前权限计划时间段已认证次数, range:[0,255], step:1, desc:若认证通过, 认证次数加1*/
    "allowAuthenticationTimes": 1
    /*ro, opt, int, 当前权限计划时间段允许的认证次数, range:[0,255], step:1*/
}
}
}
}

```

69.33.3 获取查询身份证事件参数能力

Request URL

GET /ISAPI/AccessControl/IDCardInfoEvent/capabilities?format=json

查询参数

无

请求报文

无

响应报文

有效期：2024-08-15

湖南云舟电力科技有限公司

```

{
  "IDCardInfoEventCap": {
    /*ro, req, object, 身份证事件主动获取能力*/
    "IDCardInfoEventCond": {
      /*ro, opt, object, 身份证事件主动获取条件*/
      "searchID": {
        /*ro, req, object, 搜索记录唯一标识, desc:用来确认上层客户端是否为同一个（倘若为同一个，设备记录内存，下次搜索加快速度）*/
        "@min": 1,
        /*ro, opt, int, 最小值*/
        "@max": 1
        /*ro, opt, int, 最大值*/
      },
      "searchResultPosition": {
        /*ro, req, object, 查询结果在结果列表中的起始位置, desc:当记录条数很多时，一次查询不能获取所有的记录，下一次查询时指定位置可以查询后面的记录（若设备支持的最大totalMatches为M个，但是当前设备已存储的totalMatches为N个（N<=M），则该字段的合法范围为0-N-1）*/
        "@min": 1,
        /*ro, opt, int, 最小值*/
        "@max": 1
        /*ro, opt, int, 最大值*/
      },
      "maxResults": {
        /*ro, req, object, 本次协议调用可获取的最大记录数, desc:如maxResults值大于设备能力集返回的范围，则设备按照能力集最大值返回，设备不进行报错*/
        "@min": 1,
        /*ro, opt, int, 最小值*/
        "@max": 1
        /*ro, opt, int, 最大值*/
      },
      "major": {
        /*ro, req, object, 主类型, desc:参考事件上传宏定义（此处传递的是10进制数据，不能传递16进制）：0-全部，1-报警主类型，2-异常主类型，3-操作主类型，5-事件主类型*/
        "@opt": [0, 1, 2, 3, 5]
        /*ro, req, array, 可选项, subType:int*/
      },
      "minorAlarm": {
        /*ro, req, object, 报警次类型, desc:参考事件上传宏定义（此处传递的是10进制数据，不能传递16进制），对应配置中minor节点的能力*/
        "@opt": [1024, 1025, 1026, 1027]
        /*ro, req, array, 可选项, subType:int*/
      },
      "minorException": {
        /*ro, req, object, 异常次类型, desc:参考事件上传宏定义（此处传递的是10进制数据，不能传递16进制），对应配置中minor节点的能力*/
        "@opt": [39, 58, 59, 1024]
        /*ro, req, array, 可选项, subType:int*/
      },
      "minorOperation": {
        /*ro, req, object, 操作次类型, desc:参考事件上传宏定义（此处传递的是10进制数据，不能传递16进制），对应配置中minor节点的能力*/
        "@opt": [80, 90, 112, 113]
        /*ro, req, array, 可选项, subType:int*/
      },
      "minorEvent": {
        /*ro, req, object, 事件次类型, desc:参考事件上传宏定义（此处传递的是10进制数据，不能传递16进制），对应配置中minor节点的能力*/
        "@opt": [1, 2, 3, 4]
        /*ro, req, array, 可选项, subType:int*/
      },
      "startTime": "1970-01-01T00:00:00+08:00",
        /*ro, opt, datetime, 开始时间（UTC时间）*/
      "endTime": "1970-01-01T00:00:00+08:00",
        /*ro, opt, datetime, 结束时间（UTC时间）*/
      "picEnable": {
        /*ro, opt, object, 是否带图片, desc:false-不带图片，true-带图片（1.配置为false，所有符合条件的事件不带图片上报。2.配置为true，所有符合条件的事件如无联动图片，则上报事件本身；符合条件的事件如有联动图片，则上报事件+联动图片。3.该字段如未配置，则设备默认按照true返回。）*/
        "@opt": [true, false]
        /*ro, opt, array, 可选项, subType:bool*/
      },
      "beginSerialNo": {
        /*ro, opt, object, 起始流水号*/
        "@min": 1,
        /*ro, opt, int, 最小值*/
        "@max": 1
        /*ro, opt, int, 最大值*/
      },
      "endSerialNo": {
        /*ro, opt, object, 结束流水号*/
        "@min": 1,
        /*ro, opt, int, 最小值*/
        "@max": 1
        /*ro, opt, int, 最大值*/
      },
    },
  },
}

```

69.33.4 获取身份证黑名单模板

Request URL

GET /ISAPI/AccessControl/IDBlackListCfg/template?format=json

查询参数

无
请求报文
无
响应报文
无

69.33.5 配置身份证黑名单清空

Request URL

PUT /ISAPI/AccessControl/IDBlackListClear?format=json

查询参数

无
请求报文
无
响应报文

```
{
  "statusCode": 1,
  /*ro, opt, int, 状态码, desc:无法用1表示时（1表示成功且无特殊状态）必须返回*/
  "statusString": "OK",
  /*ro, opt, string, 状态描述, range:[1,64], desc:无法用OK表示时（OK表示成功且无特殊状态）必须返回*/
  "subStatusCode": "ok",
  /*ro, opt, string, 子状态码, range:[1,64], desc:无法用ok表示（ok表示成功且无特殊状态）必须返回*/
  "errorCode": 1,
  /*ro, opt, int, 错误码, desc:当statusCode不为1时，与subStatusCode对应*/
  "errorMsg": "ok"
  /*ro, opt, string, 错误信息, desc:当statusCode不为1时，必须返回,解释信息在协议约束中，允许设备在后续的版本迭代中，进行优化丰富提升（不限制死）*/
}
```

69.33.6 获取身份证黑名单参数配置能力

Request URL

GET /ISAPI/AccessControl/IDBlackListParamCfg/capabilities?format=json

查询参数

无
请求报文
无
响应报文

```
{
  "IDBlackListParamCfgCap": {
    /*ro, req, object, 身份证黑名单参数配置能力*/
    "enabled": {
      /*ro, opt, object, 使能*/
      "@opt": [true, false]
      /*ro, opt, array, 可选项, subType:bool*/
    }
  }
}
```

69.33.7 配置身份证黑名单参数

Request URL

PUT /ISAPI/AccessControl/IDBlackListParamCfg?format=json

查询参数

无
请求报文

```
{
  "IDBlackListParamCfg": {
    /*req, object, 身份证黑名单参数配置*/
    "enabled": true
    /*opt, bool, 使能*/
  }
}
```

响应报文

```
{
  "statusCode": 1,
  /*ro, opt, int, 状态码, desc:无法用1表示时（1表示成功且无特殊状态）必须返回*/
  "statusString": "OK",
  /*ro, opt, string, 状态描述, range:[1,64], desc:无法用OK表示时（OK表示成功且无特殊状态）必须返回*/
  "subStatusCode": "ok",
  /*ro, opt, string, 子状态码, range:[1,64], desc:无法用ok表示（ok表示成功且无特殊状态）必须返回*/
  "errorCode": 1,
  /*ro, opt, int, 错误码, desc:当statusCode不为1时，与subStatusCode对应*/
  "errorMsg": "ok"
  /*ro, opt, string, 错误信息, desc:当statusCode不为1时，必须返回,解释信息在协议约束中，允许设备在后续的版本迭代中，进行优化丰富提升（不限制死）*/
}
```

69.33.8 获取身份证黑名单参数

Request URL

GET /ISAPI/AccessControl/IDBlackListParamCfg?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "IDBlackListParamCfg": {
    /*ro, req, object, 身份证黑名单参数配置*/
    "enabled": true
    /*ro, opt, bool, 使能*/
  }
}
```

69.33.9 获取身份证黑名单下发能力

Request URL

GET /ISAPI/AccessControl/IDBlackListCfg/capabilities

查询参数

无

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<IDBlackListCfg xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, 身份证黑名单, attr:version{req, string, 协议版本}-->
  <blackListValid opt="0,1">
    <!--ro, req, enum, 身份证黑名单是否有效, subType:int, [0#无效,1#有效], attr:opt{req, string, 取值范围}, desc:用于按身份证号码删除身份证黑名单, 该字段为0时代表删除-->1
  </blackListValid>
  <IDCardInfo>
    <!--ro, opt, object, 身份证信息-->
    <name min="1" max="10">
      <!--ro, opt, string, 姓名, range:[1,32], attr:min{req, int, 姓名长度最小值, desc:姓名长度最小值},max{req, int, 姓名长度最大值, desc:姓名长度最大值}-->test
    </name>
    <birth>
      <!--ro, opt, date, 出生日期-->1970-01-01
    </birth>
    <addr min="1" max="10">
      <!--ro, opt, string, 住址, range:[1,32], attr:min{req, int, 住址最小长度},max{req, int, 住址最大长度}-->test
    </addr>
    <IDNum min="18" max="18">
      <!--ro, req, const, 身份证号码, attr:min{req, int, 身份证号码最小长度},max{req, int, 身份证号码最大长度}-->542137199002241111
    </IDNum>
    <issuingAuthority min="1" max="10">
      <!--ro, opt, string, 签发机关, attr:min{req, int, 签发机关最小长度},max{req, int, 签发机关最大长度}-->test
    </issuingAuthority>
    <startDate>
      <!--ro, opt, date, 有效开始日期-->1970-01-01
    </startDate>
    <endDate>
      <!--ro, opt, date, 有效截止日期, dep:and,{$.IDBlackListCfg.IDCardInfo.termOfValidity,ue,true}-->1970-01-01
    </endDate>
    <termOfValidity opt="true,false">
      <!--ro, opt, bool, 是否长期有效, attr:opt{req, string, 取值范围}, desc:false-否, true-是(有效截止日期无效)-->true
    </termOfValidity>
    <sex opt="male,female">
      <!--ro, opt, enum, 性别, subType:string, [male#男,female#女], attr:opt{req, string, 性别取值范围, desc:male-男, female-女}-->male
    </sex>
    <nation
      opt="1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57">
      <!--ro, opt, enum, 民族, subType:int, [1#汉,2#蒙古,3#回,4#藏,5#维吾尔,6#苗,7#彝,8#壮,9#布依,10#朝鲜,11#满,12#侗,13#瑶,14#白,15#土家,16#哈尼,17#哈萨克,18#傣,19#黎,20#傈僳,21#佤,22#侬,23#高山,24#拉祜,25#水,26#东乡,27#纳西,28#景颇,29#柯尔克孜,30#土,31#达斡尔,32#仡佬,33#羌,34#布朗,35#撒拉,36#毛南,37#仫佬,38#锡伯,39#阿昌,40#普米,41#塔吉克,42#怒,43#乌孜别克,44#俄罗斯,45#鄂温克,46#德昂,47#保安,48#裕固,49#京,50#塔塔尔,51#独龙,52#鄂伦春,53#赫哲,54#门巴,55#珞巴,56#基诺,57#未知], attr:opt{req, string, 取值范围}, desc:由于之前门禁产品线楼宇对讲、明眸设备实现该字段的时候,是按照min、max属性返回的,是不合理的,为了兼容不同产线设备,因此上层解析时,优先解析opt,其次解析min、max;且后续设备实现该协议时,一律按照opt属性实现-->1
    </nation>
  </IDCardInfo>
</IDBlackListCfg>
```

69.33.10 身份证事件

事件标识: IDCardInfoEvent

```
{
  "ipAddress": "172.6.64.7",
  /*ro, req, string, 报警设备IPv4地址*/
  "ipv6Address": "1080:0:0:0:8:800:200C:417A",
  /*ro, opt, string, 报警设备IPv6地址*/
  "portNo": 80,
  /*ro, opt, int, 报警设备端口号*/
  "protocol": "HTTP",
  /*ro, opt, enum, 传输通信协议类型, subType:string, [HTTP#HTTP,HTTPS#HTTPS,EHome#EHome], desc:传输通信协议HTTP/HTTPS/EHome(萤石透传ISAPI的时候赋值HTTP;ISUP透传ISAPI的时候赋值EHome) protocolType作为历史遗留由平台兼容*/
  "macAddress": "01:17:24:45:D9:F4",
  /*ro, opt, string, MAC地址*/
  "channelID": 1,
  /*ro, opt, int, 触发报警的设备通道号, desc:触发的视频通道号(1、在SDK透传ISAPI协议的时候,上传的是私有协议对应的视频通道号;(2、在萤石透传ISAPI协议的时候,上传的是萤石协议对应的视频通道号;(3、在ISUP透传ISAPI协议的时候,上传的是ISUP协议对应的视频通道号;*/
  "dateTime": "2004-05-03T17:30:08+08:00",
  /*ro, req, datetime, 报警触发时间*/
  "activePostCount": 1,
  /*ro, opt, int, 同一个报警已经上传的次数, desc:事件触发频次脉冲事件 定义:事件持续触发(按照设备的检测频率),例如:移动检测.瞬时事件 定义:区分目标,一个目标触发一次,例如:人脸识别activePostCount 在脉冲事件 类型触发的时候,用于区分是否是同一触发源触发的事件.例如:移动检测,按照设备检测频率会一直上传;如果触发源发生了变化,这个时候计数就可以重新开始了.这个可以作为事件触发频次的方式来集成:*/
  "eventType": "IDCardInfoEvent",
  /*ro, req, string, 事件类型, desc:IDCardInfoEvent-身份证事件*/
  "eventState": "active",
  /*ro, req, enum, 事件状态, subType:string, [active#有效事件,inactive#无效事件], desc:针对持续性事件active - 表示有效事件(开始 或者 无过程状态也使用该字段); inactive - 表示无效事件(结束): remark:在心跳类型下, 该字段赋值(表示心跳数据,10s上传一次); */
  "eventDescription": "IDCard Info Event",
  /*ro, req, string, 事件描述*/
  "deviceID": "test0123",
  /*ro, opt, string, 即PUID, desc:ehome报警中需返回,例如test0123.*/
  "IDCardInfoEvent": {
    /*ro, opt, object, 身份证事件*/
    "deviceName": "test",
    /*ro, opt, string, 设备名称, range:[1,32]*/
```

```

"majorEventType": 1,
/*ro, req, int, 报警主类型, desc:参考宏定义（此处传递的是10进制数据，不能传递16进制，如传递1代表0x1（报警主类型））*/
"subEventType": 1,
/*ro, req, int, 报警次类型, dep:and,{$.IDCardInfoEvent.majorEventType,eq,任意值}, desc:参考宏定义（此处传递的是10进制数据，不能传递16进制，如传递
1024代表0x400（防区短路报警次类型），0x69-人证比对通过，0x70-人证比对失败）*/
"inductiveEventType": "authenticated",
/*ro, opt, enum, 归纳事件类型, subType:string, [authenticated#认证通过,authenticationFailed#认证失败,openingDoor#开门动作,closingDoor#关门动
作,doorException#门异常,remoteOperation#远程操作,timeSynchronization#校时事件,deviceException#设备异常事件,deviceRecovered#设备恢复正常事件,alarmTriggered#
报警事件,alarmRecovered#报警恢复事件,callCenter#呼叫中心], desc:（后端设备定义，门禁设备暂时用不到）*/
"netUser": "test",
/*ro, opt, string, 网络操作的用户名, range:[1,32]*/
"remoteHostAddr": "test",
/*ro, opt, string, 远程主机地址*/
"cardType": 1,
/*ro, opt, enum, 卡类型, subType:int, [1#普通卡,2#残疾人卡,3#黑名单卡,4#巡更卡,5#胁迫卡,6#超级卡,7#来宾卡,8#解除卡]*/
"cardReaderNo": 1,
/*ro, opt, int, 读卡器编号*/
"doorNo": 1,
/*ro, opt, int, 门编号（楼层编号）*/
"deviceNo": 1,
/*ro, opt, int, 设备编号*/
"serialNo": 1,
/*ro, opt, int, 事件流水号*/
"currentEvent": true,
/*ro, opt, bool, 是否为实时事件, desc:ture-是（实时事件），false-否（离线事件）*/
"QRCodeInfo": "test",
/*ro, opt, string, 二维码信息, range:[1,64]*/
"thermometryUnit": "celsius",
/*ro, opt, enum, 测温单位, subType:string, [celsius#摄氏度（默认）,fahrenheit#华氏度,kelvin#开尔文], desc:默认摄氏度*/
"currTemperature": 36.1,
/*ro, opt, float, 人脸温度, desc:精确到小数点后一位*/
"isAbnomalTemperature": true,
/*ro, opt, bool, 人脸测温是否温度异常*/
"RegionCoordinates": {
/*ro, opt, object, 人脸温度坐标*/
"positionX": 0,
/*ro, opt, int, X坐标, range:[0,1000], desc:归一化坐标0-1000*/
"positionY": 0
/*ro, opt, int, Y坐标, range:[0,1000], desc:归一化坐标0-1000*/
},
"remoteCheck": true,
/*ro, opt, bool, 是否需要远程核验, desc:true-需要, false-不需要, 默认false*/
"mask": "unknown",
/*ro, opt, enum, 是否戴口罩, subType:string, [unknown#未知,yes#戴口罩,no#不戴口罩]*/
"frontSerialNo": 1,
/*ro, opt, int, 上一条事件流水号, desc:若设备没返回该字段，平台根据serialNo判断是否丢失事件；若设备返回该字段，平台根据该字段和serialNo字段共同判断
是否丢失事件（主要用于解决报警订阅后导致serialNo不连续的情况）*/
"IDCardInfo": {
/*ro, opt, object, 身份证信息*/
"name": "test",
/*ro, opt, string, 姓名, range:[1,128]*/
"sex": "male",
/*ro, opt, enum, 性别, subType:string, [male#男,female#女]*/
"birth": "1970-01-01",
/*ro, opt, date, 出生日期*/
"addr": "test",
/*ro, opt, string, 住址*/
"IDCardNo": "test",
/*ro, opt, string, 身份证号码*/
"issuingAuthority": "test",
/*ro, opt, string, 签发机关*/
"startDate": "test",
/*ro, opt, string, 有效期起始时间*/
"endDate": "test",
/*ro, opt, string, 有效期截止时间*/
"nation": 1,
/*ro, opt, int, 民族, range:[1,57], desc:1-"汉", 2-"蒙古", 3-"回", 4-"藏", 5-"维吾尔", 6-"苗", 7-"彝", 8-"壮", 9-"布依", 10-"朝鲜", 11-"满", 12-"侗", 13-
"瑶", 14-"白", 15-"土家", 16-"哈尼", 17-"哈萨克", 18-"傣", 19-"黎", 20-"傈僳", 21-"佤", 22-"畲", 23-"高山", 24-"拉祜", 25-"水", 26-"东乡", 27-"纳西", 28-"景颇", 29-"柯尔克
孜", 30-"土", 31-"达斡尔", 32-"仫佬", 33-"羌", 34-"布朗", 35-"撒拉", 36-"毛南", 37-"仡佬", 38-"锡伯", 39-"阿昌", 40-"普米", 41-"塔吉克", 42-"怒", 43-"乌孜别克", 44-"俄罗
斯", 45-"鄂温克", 46-"德昂", 47-"保安", 48-"裕固", 49-"京", 50-"塔塔尔", 51-"独龙", 52-"鄂伦春", 53-"赫哲", 54-"门巴", 55-"珞巴", 56-"基诺", 57及之后为未知*/
"isLongTermEffective": false
/*ro, opt, bool, 是否长期有效*/
},
"pictureURL": "test",
/*ro, opt, string, 抓拍图片URL, range:[1,256]*/
"IDCardPicURL": "test",
/*ro, opt, string, 身份证图片URL, range:[1,256]*/
"certificatePicURL": "test",
/*ro, opt, string, 证件抓拍图片URL, range:[1,256]*/
"contentId": "Picture",
/*ro, opt, string, 图片contentID, desc:直传二进制图片数据时使用,与报文Content-ID对应*/
"IDCardContentId": "IDCardPicture",
/*ro, opt, string, 身份证图片contentID, desc:直传二进制图片数据时使用,与报文Content-ID对应*/
"certificateContentId": "CertificatePicture",
/*ro, opt, string, 证件抓拍图片contentID, desc:直传二进制图片数据时使用,与报文Content-ID对应*/
"visibleLightURL": "test",
/*ro, opt, string, 热成像相机可见光图片URL, range:[0,256]*/
"thermalURL": "test",
/*ro, opt, string, 热成像图片URL, range:[0,256]*/
"picturesNumber": 1,
/*ro, opt, int, 图片数量, range:[0,4], desc:没图片时，该字段为0或不返回*/
"helmet": "unknown",
/*ro, opt, enum, 是否戴安全帽, subType:string, [unknown#未知,yes#戴安全帽,no#不戴安全帽]*/

```

```

    "HealthInfo": {
      /*ro, opt, object, 健康信息*/
      "healthCode": 1,
      /*ro, opt, enum, 健康码状态, subType:int, [0#未请求,1#未申领,2#绿码,3#黄码,4#红码,5#无此人员,6#其他错误信息（如接口异常导致查询失败）,7#查询健康码超时]*/
      "NADCode": 1,
      /*ro, opt, enum, 核酸检测结果, subType:int, [0#未查询到核酸检测结果,1#核酸检测阴性（代表正常）,2#核酸检测阳性（代表确诊）,3#核酸检测有效期已过,4#查询核酸结果失败]*/
      "NADMsg": "test",
      /*ro, opt, string, 核酸检测信息, range:[0,64], desc:空字符串表示查询到核酸检测信息失败*/
      "travelCode": 1,
      /*ro, opt, enum, 行程编码, subType:int, [0#14天内一直在当地,1#14天内离开过当地,2#14天内到过疫区,3#其他,4#查询行程信息失败]*/
      "travelInfo": "test",
      /*ro, opt, string, 行程信息, desc:空字符串表示查询到行程信息失败*/
      "vaccineStatus": 1,
      /*ro, opt, enum, 打疫苗状态, subType:int, [0#未打疫苗,1#部分注射疫苗,2#已完成疫苗,3#查询疫苗信息失败,4#已完成加强针疫苗]*/
      "vaccineNum": 1,
      /*ro, opt, int, 打疫苗针数, step:1*/
      "vaccineMsg": "test",
      /*ro, opt, string, 打疫苗信息, range:[0,64], desc:空字符串表示查询到疫苗信息失败*/
      "ANTCode": 1,
      /*ro, opt, enum, 抗原检测状态, subType:int, [0#未查询到抗原检测结果,1#抗原检测阴性,2#抗原检测阳性,3#抗原检测有效期已过,4#抗原检测无效]*/
      "ANTMsg": "test"
      /*ro, opt, string, 抗原检测信息, range:[0,64], desc:空字符串表示查询到抗原检测信息失败*/
    },
    "PhysicalInfo": {
      /*ro, opt, object, 身体信息, desc:明眸对接体脂称, 认证通过后明眸获取体脂称信息*/
      "weight": 7000,
      /*ro, opt, int, 体重, unit:kg, unitType:质量, 由于测量值存在小数点, 实际值为体重(kg)*100*/
      "height": 18000
      /*ro, opt, int, 身高, unit:cm, unitType:长度, 由于测量值存在小数点, 实际值为身高(cm)*100*/
    },
    "FaceRect": {
      /*ro, opt, object, 人脸矩形框, desc:屏幕左上角为坐标原点*/
      "height": 1.000,
      /*ro, req, float, 高度, range:[0.000,1.000]*/
      "width": 1.000,
      /*ro, req, float, 宽度, range:[0.000,1.000]*/
      "x": 0.000,
      /*ro, req, float, 区域左上角顶点的水平坐标, range:[0.000,1.000]*/
      "y": 0.000
      /*ro, req, float, 区域左上角顶点的垂直坐标, range:[0.000,1.000]*/
    },
    "currentAuthenticationTimes": 1,
    /*ro, opt, int, 当前权限计划时间段已认证次数, range:[0,255], step:1, desc:若认证通过, 认证次数加1*/
    "allowAuthenticationTimes": 1,
    /*ro, opt, int, 当前权限计划时间段允许的认证次数, range:[0,255], step:1*/
    "protocolFileAndSignPicURL": "test",
    /*ro, opt, string, 隐私声明+用户签名图片URL, range:[1,256]*/
    "protocolFileEnabled": true,
    /*ro, opt, bool, 隐私声明使能*/
    "isAgreePrivacyStatement": true
    /*ro, opt, bool, 是否同意隐私声明*/
  }
}

```

参数名称	参数值	参数类型 (Content-Type)	Content-ID	文件名称(filename)	描述
IDCardInfoEvent	[报文内容]	application/json	--	--	--
IDCardPic	[图片二进制数据]	image/jpeg	IDCardPicImage	IDCardPic.jpg	身份证图片数据
CertificatePicture	[图片二进制数据]	image/jpeg	CertificatePicture	CertificatePicture.jpg	证件抓拍图片数据
VisibleLight	[图片二进制数据]	image/jpeg	visibleLight_image	VisibleLight.jpg	热成像相机可见光图片数据
Thermal	[图片二进制数据]	image/jpeg	thermal_image	Thermal.jpg	热成像图片数据
ProtocolFileAndSignPic	[图片二进制数据]	image/jpeg		ProtocolFileAndSignPic.jpg	隐私声明+用户签名图片数据

说明：此协议通过表单格式传输，表单框架详情说明见章节4.5.1.4，如下实例：

```
--<frontier>
Content-Disposition: form-data; name=参数名称;filename=文件名称
Content-Type: 参数类型
Content-Length: ****
Content-ID: 内容ID
参数值
```

- 参数名称：表单单元Headers中Content-Disposition的name属性，表示此表单单元的名称
- 参数类型（Content-Type）：表单单元Headers中Content-Type属性
- 文件名称（filename）：表单单元Headers中Content-Disposition的filename属性，仅在本表单单元传输数据为文件类型存在，表示表单单元Body的文件名
- 参数值：表单单元的body内容

69.34 门禁点单元管理

69.34.1 获取门锁掉电开关状态配置

Request URL

GET /ISAPI/AccessControl/Configuration/lockType?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "LockType": {
    /*ro, opt, object, 状态*/
    "status": "alwaysOpen"
    /*ro, opt, enum, 掉电异常阶段锁状态, subType:string, [alwaysOpen#常开,alwaysClose#常闭]*/
  }
}
```

69.34.2 配置门锁掉电开关状态参数

Request URL

PUT /ISAPI/AccessControl/Configuration/lockType?format=json

查询参数

无

请求报文

```
{
  "LockType": {
    /*opt, object, 状态*/
    "status": "alwaysOpen"
    /*req, enum, 掉电异常阶段锁状态, subType:string, [alwaysOpen#常开,alwaysClose#常闭]*/
  }
}
```

响应报文

```
{
  "requestURL": "test",
  /*ro, opt, string, URI*/
  "statusCode": "test",
  /*ro, opt, string, 状态码*/
  "statusString": "test",
  /*ro, opt, string, 状态描述*/
  "subStatusCode": "test",
  /*ro, opt, string, 子状态码*/
  "errorCode": 1,
  /*ro, req, int, 错误码*/
  "errorMsg": "ok",
  /*ro, req, string, 错误信息*/
}
```

69.34.3 获取门参数能力

Request URL

GET /ISAPI/AccessControl/Door/param/<doorID>/capabilities

查询参数

参数名称	参数类型	描述
doorID	string	门编号

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<DoorParam xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 门参数, attr:version{req, string, 协议版本}-->
  <doorNo min="1" max="10">
    <!--ro, opt, int, 门编号, attr:min{req, int, 最小值},max{req, int, 最大值}-->1
  </doorNo>
  <doorName min="1" max="32">
    <!--ro, opt, string, 门名称, attr:min{req, int, 最小值},max{req, int, 最大值}-->test
  </doorName>
  <magneticType opt="alwaysClose,alwaysOpen">
    <!--ro, opt, enum, 门磁类型, subType:string, [alwaysClose#常闭,alwaysOpen#常开], attr:opt{req, string, 取值范围}-->alwaysClose
  </magneticType>
  <openButtonType opt="alwaysClose,alwaysOpen">
    <!--ro, opt, enum, 开门按钮类型, subType:string, [alwaysClose#常闭,alwaysOpen#常开], attr:opt{req, string, 取值范围}-->alwaysClose
  </openButtonType>
</DoorParam>
```

```
</openButtonType>
<openDuration min="1" max="255">
  <!--ro, opt, int, 开门持续时间（楼层继电器动作时间），attr:min{req, int, 最小值},max{req, int, 最大值}-->1
</openDuration>
<disabledOpenDuration min="1" max="255">
  <!--ro, opt, int, 残疾人卡开门持续时间（关门延迟持续时间），attr:min{req, int, 最小值},max{req, int, 最大值}-->1
</disabledOpenDuration>
<magneticAlarmTimeout min="0" max="255">
  <!--ro, opt, int, 门磁检测超时报警时间, range:[0,255], unit:s, unitType:时间, attr:min{req, int, 最小值},max{req, int, 最大值}, desc:0表示不报警-->1
</magneticAlarmTimeout>
<enableDoorLock opt="true,false">
  <!--ro, opt, bool, 是否启用闭门回锁, attr:opt{req, string, 取值范围}-->true
</enableDoorLock>
<enableLeaderCard opt="true,false">
  <!--ro, opt, bool, 是否启用首卡常开功能, attr:opt{req, string, 取值范围}-->true
</enableLeaderCard>
<leaderCardMode opt="disable,alwaysOpen,authorize">
  <!--ro, opt, enum, 首卡模式, subType:string, [disable#不启用,alwaysOpen#首卡常开,authorize#首卡授权], attr:opt{req, string, 取值范围}, desc:authorize-首卡授权（使用了此节点, 则enableLeaderCard节点无效）-->disable
</leaderCardMode>
<leaderCardOpenDuration min="1" max="1440">
  <!--ro, opt, int, 首卡常开持续时间, attr:min{req, int, 最小值},max{req, int, 最大值}-->1
</leaderCardOpenDuration>
<stressPassword min="1" max="8">
  <!--ro, opt, string, 胁迫密码, attr:min{req, int, 最小值},max{req, int, 最大值}, desc:最大8个字节采用Base64进行编码后传输-->test
</stressPassword>
<superPassword min="1" max="8">
  <!--ro, opt, string, 超级密码, attr:min{req, int, 最小值},max{req, int, 最大值}, desc:最大8个字节采用Base64进行编码后传输-->test
</superPassword>
<unlockPassword min="1" max="8">
  <!--ro, opt, string, 解锁密码, attr:min{req, int, 最小值},max{req, int, 最大值}, desc:最大8个字节采用Base64进行编码后传输-->test
</unlockPassword>
<useLocalController opt="true,false">
  <!--ro, opt, bool, 是否连接在就地控制器上, attr:opt{req, string, 取值范围}-->true
</useLocalController>
<localControllerID min="0" max="64">
  <!--ro, opt, int, 就地控制器序号, range:[0,64], attr:min{req, int, 最小值},max{req, int, 最大值}, desc:0代表未注册-->1
</localControllerID>
<localControllerDoorNumber min="0" max="4">
  <!--ro, opt, int, 就地控制器的门编号, range:[0,4], attr:min{req, int, 最小值},max{req, int, 最大值}, desc:0代表未注册-->1
</localControllerDoorNumber>
<localControllerStatus opt="0,1,2,3,4,5,6,7,8,9">
  <!--ro, opt, enum, 就地控制器在线状态, subType:int, [0#离线,1#网络在线,2#环路1上的RS485串口1,3#环路1上的RS485串口2,4#环路2上的RS485串口1,5#环路2上的RS485串口2,6#环路3上的RS485串口1,7#环路3上的RS485串口2,8#环路4上的RS485串口1,9#环路4上的RS485串口2], attr:opt{req, string, 取值范围}-->1
</localControllerStatus>
<lockInputCheck opt="true,false">
  <!--ro, opt, bool, 是否启用门锁输入检测, attr:opt{req, string, 取值范围}-->true
</lockInputCheck>
<lockInputType opt="alwaysClose,alwaysOpen">
  <!--ro, opt, enum, 门锁输入类型, subType:string, [alwaysClose#常闭,alwaysOpen#常开], attr:opt{req, string, 取值范围}, desc:默认常闭-->alwaysClose
</lockInputType>
<doorTerminalMode opt="preventCutAndShort,preventCutAndShort,common">
  <!--ro, opt, enum, 门相关端子工作模式, subType:string, [preventCutAndShort#防剪防短,common#普通], attr:opt{req, string, 取值范围}, desc:默认防剪防短-->preventCutAndShort
</doorTerminalMode>
<openButton opt="true,false">
  <!--ro, opt, bool, 是否启用开门按钮, attr:opt{req, string, 取值范围}, desc:默认是-->true
</openButton>
<ladderControlDelayTime min="1" max="255">
  <!--ro, opt, int, 梯控访客延迟时间, attr:min{req, int, 最小值},max{req, int, 最大值}-->1
</ladderControlDelayTime>
<Leader>
  <!--ro, opt, object, 首人配置-->
  <continuousVerificationTimes min="1" max="10">
    <!--ro, opt, int, 连续认证次数, range:[1,10], attr:min{req, int, 最小值},max{req, int, 最大值}, desc:规定时间内同个用户连续进行首人认证, 次数满足后触发首人功能-->1
  </continuousVerificationTimes>
  <continuousVerificationDuration min="1" max="10">
    <!--ro, opt, int, 连续认证间隔, range:[5,60], unit:s, unitType:时间, attr:min{req, int, 最小值},max{req, int, 最大值}, desc:规定时间内同个用户连续进行首人认证, 次数满足后触发首人功能-->20
  </continuousVerificationDuration>
  <effectiveTimeEnabled opt="true,false">
    <!--ro, req, bool, 生效时间点使能, attr:opt{req, string, 取值范围}, desc:默认不启用-->true
  </effectiveTimeEnabled>
  <dayBeginTime>
    <!--ro, opt, time, 一天的开始时间, dep:and,{$.DoorParam.Leader.effectiveTimeEnabled,eq,true}, desc:00:00:00~24:00:00, 一天的开始时间不能在生效时间段范围内
    1.首人常开: 若dayBeginTime为3点, 表示一天从3点开始, 3点结束, 假设beginEffectiveTime为8点, endEffectiveTime为20点. 若在3点到8点之间完成首人认证, 则8点触发首人常开; 若8点到20点之间第一次完成首人认证, 则即时触发首人常开
    如再次完成首人认证, 则即时关闭首人常开, 如一直未再次完成首人认证, 则20点自动关闭首人常开; 若在20点到3点之间完成首人认证, 则无效
    2.首人认证同理
    特殊情况: 如果endEffectiveTime与LeaderCardOpenDuration均配置, 则取两者的最小值作为结束生效时间点-->00:00:00
  </dayBeginTime>
  <beginEffectiveTime>
    <!--ro, opt, time, 开始生效时间点, dep:and,{$.DoorParam.Leader.effectiveTimeEnabled,eq,true}, desc:00:00:00~24:00:00-->00:00:00
  </beginEffectiveTime>
  <endEffectiveTime>
    <!--ro, opt, time, 结束生效时间点, dep:and,{$.DoorParam.Leader.effectiveTimeEnabled,eq,true}, desc:00:00:00~24:00:00-->00:00:00
  </endEffectiveTime>
</Leader>
<verificationPassOpenDoor opt="true,false">
  <!--ro, opt, bool, 认证通过开门, attr:opt{req, string, 取值范围}, desc:信发门牌使用,默认为false,门牌内部通过电磁继电器与门锁连接,电磁继电器开门时会发出“喀”的一声,对于没有门锁的门,可以关闭开门使能,避免产生声响-->true
```

```
</verificationPassOpenDoor>
<relayReverseEnabled opt="true,false" def="false">
  <!--ro, opt, bool, 继电器反转使能, attr:opt{req, string, 取值范围},def{req, bool, 默认值}, desc:继电器反转表现为门锁的接入方式,默认不反转-->true
</relayReverseEnabled>
</DoorParam>
```

69.34.4 获取门参数

Request URL

GET /ISAPI/AccessControl/Door/param/<doorID>?security=<security>&iv=<iv>

查询参数

参数名称	参数类型	描述
doorID	string	门编号（楼层编号），从1开始
security	string	没有security表示数据不加密；security=1,表示报文中的stressPassword, superPassword, unlockPassword字段AES128加密；security=2,表示报文中的stressPassword, superPassword, unlockPassword字段AES256加密；
iv	string	iv表示初始化向量，在security=1或者2时为必须参数；

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<DoorParam xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 门参数, attr:version{req, string, 协议版本}-->
  <doorName>
    <!--ro, opt, string, 门名称-->test
  </doorName>
  <magneticType>
    <!--ro, opt, enum, 门磁类型, subType:string, [alwaysClose#常闭,alwaysOpen#常开]-->alwaysClose
  </magneticType>
  <openButtonType>
    <!--ro, opt, enum, 开门按钮类型, subType:string, [alwaysClose#常闭,alwaysOpen#常开]-->alwaysClose
  </openButtonType>
  <openDuration>
    <!--ro, opt, int, 开门持续时间, range:[1,255], unit:s, unitType:时间-->1
  </openDuration>
  <disabledOpenDuration>
    <!--ro, opt, int, 残疾人卡开门持续时间（关门延迟持续时间），range:[1,255], unit:s, unitType:时间-->1
  </disabledOpenDuration>
  <magneticAlarmTimeout>
    <!--ro, opt, int, 门磁检测超时报警时间, range:[0,255], unit:s, unitType:时间, desc:0表示不报警-->1
  </magneticAlarmTimeout>
  <enableDoorLock>
    <!--ro, opt, bool, 是否启用闭门回锁-->true
  </enableDoorLock>
  <enableLeaderCard>
    <!--ro, opt, bool, 是否启用首卡常开功能-->true
  </enableLeaderCard>
  <leaderCardMode>
    <!--ro, opt, enum, 首卡模式, subType:string, [disable#不启用,alwaysOpen#首卡常开,authorize#首卡授权], desc:authorize-首卡授权（使用了此节点，则enableLeaderCard节点无效）-->disable
  </leaderCardMode>
  <leaderCardOpenDuration>
    <!--ro, opt, int, 首卡常开持续时间, range:[0,1440], unit:min, unitType:时间, dep:and,{$.DoorParam.LeaderCardMode,eq,alwaysOpen}-->1
  </leaderCardOpenDuration>
  <stressPassword>
    <!--ro, opt, string, 胁迫密码, desc:最大8个字节采用Base64进行编码后传输-->test
  </stressPassword>
  <superPassword>
    <!--ro, opt, string, 超级密码, desc:最大8个字节采用Base64进行编码后传输-->test
  </superPassword>
  <unlockPassword>
    <!--ro, opt, string, 解锁密码, desc:最大8个字节采用Base64进行编码后传输-->test
  </unlockPassword>
  <useLocalController>
    <!--ro, opt, bool, 是否连接在就地控制器上-->true
  </useLocalController>
  <localControllerID>
    <!--ro, opt, int, 就地控制器序号, range:[0,64], desc:0代表未注册-->1
  </localControllerID>
  <localControllerDoorNumber>
    <!--ro, opt, int, 就地控制器的门编号, range:[0,4], desc:0代表未注册-->1
  </localControllerDoorNumber>
</DoorParam>
```

```
</localControllerDoorNumber>
<localControllerStatus>
  <!--ro, opt, enum, 就地控制器在线状态, subType:int, [0#离线,1#网络在线,2#环路1上的RS485串口1,3#环路1上的RS485串口2,4#环路2上的RS485串口1,5#环路2上的RS485串口2,6#环路3上的RS485串口1,7#环路3上的RS485串口2,8#环路4上的RS485串口1,9#环路4上的RS485串口2]-->1
</localControllerStatus>
<lockInputCheck>
  <!--ro, opt, bool, 是否启用门锁输入检测-->true
</lockInputCheck>
<lockInputType>
  <!--ro, opt, enum, 门锁输入类型, subType:string, [alwaysClose#常闭,alwaysOpen#常开], desc:默认常闭-->alwaysClose
</lockInputType>
<doorTerminalMode>
  <!--ro, opt, enum, 门相关端子工作模式, subType:string, [preventCutAndShort#防剪防短,common#普通], desc:默认防剪防短-->preventCutAndShort
</doorTerminalMode>
<openButton>
  <!--ro, opt, bool, 是否启用开门按钮-->true
</openButton>
<ladderControlDelayTime>
  <!--ro, opt, int, 梯控访客延迟时间, range:[1,255], unit:min, unitType:时间-->1
</ladderControlDelayTime>
<Leader>
  <!--ro, opt, object, 首人配置-->
  <continuousVerificationTimes>
    <!--ro, opt, int, 连续认证次数, range:[1,10], desc:规定时间内同个用户连续进行首人认证, 次数满足后触发首人功能-->1
  </continuousVerificationTimes>
  <continuousVerificationDuration>
    <!--ro, opt, int, 连续认证间隔, range:[5,60], unit:s, unitType:时间, desc:规定时间内同个用户连续进行首人认证, 次数满足后触发首人功能-->20
  </continuousVerificationDuration>
  <effectiveTimeEnabled>
    <!--ro, req, bool, 生效时间点使能, desc:默认不启用-->true
  </effectiveTimeEnabled>
  <dayBeginTime>
    <!--ro, opt, time, 一天的开始时间, dep:and,{$.DoorParam.Leader.effectiveTimeEnabled,eq,true}, desc:00:00:00-24:00:00, 一天的开始时间不能在生效时间段范围内
1.首人常开: 若dayBeginTime为3点, 表示一天从3点开始, 3点结束, 假设beginEffectiveTime为8点, endEffectiveTime为20点. 若在3点到8点之间完成首人认证, 则8点触发首人常开; 若8点到20点之间第一次完成首人认证, 则即时触发首人常开
如再次完成首人认证, 则即时关闭首人常开, 则20点自动关闭首人常开; 若在20点到3点之间完成首人认证, 则无效
2.首人认证同理
特殊情况: 如果endEffectiveTime与LeaderCardOpenDuration均配置, 则取两者的最小值作为结束生效时间点-->00:00:00
  </dayBeginTime>
  <beginEffectiveTime>
    <!--ro, opt, time, 开始生效时间点, dep:and,{$.DoorParam.Leader.effectiveTimeEnabled,eq,true}, desc:00:00:00-24:00:00-->00:00:00
  </beginEffectiveTime>
  <endEffectiveTime>
    <!--ro, opt, time, 结束生效时间点, dep:and,{$.DoorParam.Leader.effectiveTimeEnabled,eq,true}, desc:00:00:00-24:00:00-->00:00:00
  </endEffectiveTime>
</Leader>
<verificationPassOpenDoor>
  <!--ro, opt, bool, 认证通过开门, desc:触发门牌使用,默认为false,门牌内部通过电磁继电器与门锁连接,电磁继电器开门时会发出"嗒"的一声,对于没有门锁的门,可以关闭开门使能,避免产生声响-->true
</verificationPassOpenDoor>
<relayReverseEnabled>
  <!--ro, opt, bool, 继电器反转使能, desc:继电器反转表现为门锁的接入方式,默认不反转-->true
</relayReverseEnabled>
</DoorParam>
```

69.34.5 配置门参数

Request URL

PUT /ISAPI/AccessControl/Door/param/<doorID>?security=<security>&iv=<iv>

查询参数

参数名称	参数类型	描述
doorID	string	门编号（楼层编号），从1开始
security	string	没有security表示数据不加密；security=1,表示报文中的stressPassword, superPassword, unlockPassword字段AES128加密；security=2,表示报文中的stressPassword, superPassword, unlockPassword字段AES256加密；
iv	string	iv表示初始化向量，在security=1或者2时为必须参数;

请求报文

```
<?xml version="1.0" encoding="UTF-8"?>
<DoorParam xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, 门参数, attr:version{req, string, 协议版本}-->
  <doorName>
    <!--opt, string, 门名称-->test
  </doorName>
```

```

</doorParam>
<magneticType>
  <!--opt, enum, 门磁类型, subType:string, [alwaysClose#常闭,alwaysOpen#常开]-->alwaysClose
</magneticType>
<openButtonType>
  <!--opt, enum, 开门按钮类型, subType:string, [alwaysClose#常闭,alwaysOpen#常开]-->alwaysClose
</openButtonType>
<openDuration>
  <!--opt, int, 开门持续时间, range:[1,255], unit:s, unitType:时间-->1
</openDuration>
<disabledOpenDuration>
  <!--opt, int, 残疾人卡开门持续时间（关门延迟持续时间）, range:[1,255], unit:s, unitType:时间-->1
</disabledOpenDuration>
<magneticAlarmTimeout>
  <!--opt, int, 门磁检测超时报警时间, range:[0,255], unit:s, unitType:时间, desc:0表示不报警-->1
</magneticAlarmTimeout>
<enableDoorLock>
  <!--opt, bool, 是否启用闭门回锁-->true
</enableDoorLock>
<enableLeaderCard>
  <!--opt, bool, 是否启用首卡常开功能-->true
</enableLeaderCard>
<leaderCardMode>
  <!--opt, enum, 首卡模式, subType:string, [disable#不启用,alwaysOpen#首卡常开,authorize#首卡授权], desc:authorize-首卡授权（使用了此节点，则enableLeaderCard节点无效）-->disable
</leaderCardMode>
<leaderCardOpenDuration>
  <!--opt, int, 首卡常开持续时间, range:[0,1440], unit:min, unitType:时间, dep:and,{$.DoorParam.LeaderCardMode,eq,alwaysOpen}-->1
</leaderCardOpenDuration>
<stressPassword>
  <!--opt, string, 胁迫密码, desc:最大8个字节采用Base64进行编码后传输-->test
</stressPassword>
<superPassword>
  <!--opt, string, 超级密码, desc:最大8个字节采用Base64进行编码后传输-->test
</superPassword>
<unlockPassword>
  <!--opt, string, 解锁密码, desc:最大8个字节采用Base64进行编码后传输-->test
</unlockPassword>
<useLocalController>
  <!--opt, bool, 是否连接在就地控制器上-->true
</useLocalController>
<localControllerID>
  <!--opt, int, 就地控制器序号, range:[0,64], desc:0代表未注册-->1
</localControllerID>
<localControllerDoorNumber>
  <!--opt, int, 就地控制器的门编号, range:[0,4], desc:0代表未注册-->1
</localControllerDoorNumber>
<localControllerStatus>
  <!--opt, enum, 就地控制器在线状态, subType:int, [0#离线,1#网络在线,2#环路1上的RS485串口1,3#环路1上的RS485串口2,4#环路2上的RS485串口1,5#环路2上的RS485串口2,6#环路3上的RS485串口1,7#环路3上的RS485串口2,8#环路4上的RS485串口1,9#环路4上的RS485串口2]-->1
</localControllerStatus>
<lockInputCheck>
  <!--opt, bool, 是否启用门锁输入检测-->true
</lockInputCheck>
<lockInputType>
  <!--opt, enum, 门锁输入类型, subType:string, [alwaysClose#常闭,alwaysOpen#常开], desc:默认常闭-->alwaysClose
</lockInputType>
<doorTerminalMode>
  <!--opt, enum, 门相关端子工作模式, subType:string, [preventCutAndShort#防剪防短,common#普通], desc:默认防剪防短-->preventCutAndShort
</doorTerminalMode>
<openButton>
  <!--opt, bool, 是否启用开门按钮-->true
</openButton>
<ladderControlDelayTime>
  <!--opt, int, 梯控访客延迟时间, range:[1,255], unit:min, unitType:时间-->1
</ladderControlDelayTime>
<Leader>
  <!--opt, object, 首人配置-->
  <continuousVerificationTimes>
    <!--opt, int, 连续认证次数, range:[1,10], desc:规定时间内同个用户连续进行首人认证, 次数满足后触发首人功能-->1
  </continuousVerificationTimes>
  <continuousVerificationDuration>
    <!--opt, int, 连续认证间隔, range:[5,60], unit:s, unitType:时间, desc:规定时间内同个用户连续进行首人认证, 次数满足后触发首人功能-->20
  </continuousVerificationDuration>
  <effectiveTimeEnabled>
    <!--req, bool, 生效时间点使能, desc:默认不启用-->true
  </effectiveTimeEnabled>
  <dayBeginTime>
    <!--opt, time, 一天的开始时间, dep:and,{$.DoorParam.Leader.effectiveTimeEnabled,eq,true},
desc:00:00:00~24:00:00, 一天的开始时间不能在生效时间段范围内
1.首人常开: 若dayBeginTime为3点, 表示一天从3点开始, 3点结束, 假设beginEffectiveTime为8点, endEffectiveTime为20点. 若在3点到8点之间完成首人认证, 则8点触发首人常开; 若8点到20点之间第一次完成首人认证, 则即时触发首人常开
如再次完成首人认证, 则即时关闭首人常开, 如一直未再次完成首人认证, 则20点自动关闭首人常开; 若在20点到3点之间完成首人认证, 则无效
2.首人认证同理
特殊情况: 如果endEffectiveTime与LeaderCardOpenDuration均配置, 则取两者的最小值作为结束生效时间点-->00:00:00
    </dayBeginTime>
  <beginEffectiveTime>
    <!--opt, time, 开始生效时间点, dep:and,{$.DoorParam.Leader.effectiveTimeEnabled,eq,true}, desc:00:00:00~24:00:00-->00:00:00
  </beginEffectiveTime>
  <endEffectiveTime>
    <!--opt, time, 结束生效时间点, dep:and,{$.DoorParam.Leader.effectiveTimeEnabled,eq,true}, desc:00:00:00~24:00:00-->00:00:00
  </endEffectiveTime>
</Leader>
<verificationPassOpenDoor>

```

```
<!--opt, bool, 认证通过开门, desc:信发门牌使用,默认为false,门牌内部通过电磁继电器与门锁连接,电磁继电器开门时会发出"喀"的一声,对于没有门锁的门,可以关闭开门使能,避免产生声响-->true
</verificationPassOpenDoor>
<relayReverseEnabled>
  <!--opt, bool, 继电器反转使能, desc:继电器反转表现为门锁的接入方式,默认不反转-->true
</relayReverseEnabled>
</DoorParam>
```

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 响应消息, attr:version{ro, req, string, 版本}-->
  <requestURL>
    <!--ro, req, string, 请求的URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, 状态码, subType:int, [0#OK,1#OK,2#Device Busy,3#Device Error,4#Invalid Operation,5#Invalid XML Format,6#Invalid XML Content,7#Reboot Required]-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, 状态信息, subType:string, [OK#成功,Device Busy#设备忙,Device Error#设备异常,Invalid Operation#无效的操作,Invalid XML Format#无效的XML格式,Invalid XML Content#无效的XML内容,Reboot#设备重启]-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, 详细错误码英文描述, desc:详细错误码的英文描述-->OK
  </subStatusCode>
</ResponseStatus>
```

69.34.6 获取门控安全模块状态能力

Request URL

GET /ISAPI/AccessControl/DoorSecurityModule/moduleStatus/capabilities

查询参数

无

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<ModuleStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, 模块状态, attr:version{req, string, 协议版本}-->
  <securityModuleNo min="1" max="256">
    <!--ro, req, string, 门控安全模块编号, attr:min{req, int, 最小值},max{req, int, 最大值}-->test
  </securityModuleNo>
  <onlineStatus opt="0,1">
    <!--ro, req, enum, 在线状态, subType:int, [0#不在线,1#在线], attr:opt{req, string, 取值范围}-->1
  </onlineStatus>
  <desmanteIStatus opt="0,1">
    <!--ro, req, enum, 防拆状态, subType:int, [0#关闭,1#防拆], attr:opt{req, string, 取值范围}-->1
  </desmanteIStatus>
</ModuleStatus>
```

69.34.7 获取门控安全模块状态

Request URL

GET /ISAPI/AccessControl/DoorSecurityModule/moduleStatus

查询参数

无

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<ModuleStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, 模块状态, attr:version{req, string, 协议版本}-->
  <securityModuleNo min="1" max="256">
    <!--ro, req, string, 门控安全模块编号, attr:min{req, int, 最小值},max{req, int, 最大值}-->test
  </securityModuleNo>
  <onlineStatus opt="0,1">
    <!--ro, req, enum, 在线状态, subType:int, [0#不在线,1#在线], attr:opt{req, string, 取值范围}-->1
  </onlineStatus>
  <desmantelStatus opt="0,1">
    <!--ro, req, enum, 防拆状态, subType:int, [0#关闭,1#防拆], attr:opt{req, string, 取值范围}-->1
  </desmantelStatus>
</ModuleStatus>
```

69.34.8 获取门锁掉电开关状态配置能力

Request URL

GET /ISAPI/AccessControl/Configuration/lockType/capabilities?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "LockTypeCap": {
    /*ro, req, object, 门锁掉电开关状态配置能力*/
    "status": {
      /*ro, opt, object, 掉电异常阶段锁状态*/
      "@opt": ["alwaysOpen", "alwaysClose"]
      /*ro, req, array, 可选项, subType:string*/
    }
  }
}
```

69.35 抓拍参数管理(S).

69.35.1 获取抓拍参数

Request URL

GET /ISAPI/AccessControl/SnapConfig

查询参数

无

请求报文

无

响应报文

```

<?xml version="1.0" encoding="UTF-8"?>

<SnapConfig xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, 抓拍参数, attr:version{req, string, 协议版本}-->
  <snapTimes>
    <!--ro, req, int, 线圈抓拍次数, range:[0,5]-->1
  </snapTimes>
  <snapWaitTime>
    <!--ro, req, int, 抓拍等待时间, range:[0,6000]-->1
  </snapWaitTime>
  <intervalTimeList size="4">
    <!--ro, req, array, 间隔列表, subType:object, attr:size{req, int, 数组元素总数}-->
    <intervalTime>
      <!--ro, req, int, 连拍时间间隔, range:[0,6000]-->1
    </intervalTime>
  </intervalTimeList>
  <JPEGParam>
    <!--ro, opt, object, 图片参数-->
    <pictureSize>
      <!--ro, req, string, 图片分辨率,
      desc://0-CIF, 1-QCIF, 2-D1, 3-UXGA(1600x1200),4-SVGA(800x600),5-HD720p(1280x720), 6-VGA, 7-XVGA, 8-HD900p, 9-HD1080, 10-2560*1920,
      //11-1600*304, 12-2048*1536, 13-2448*2048,14-2448*1280, 15-2448*800, 16-XGA(1024*768), 17-SXGA(1280*1024),18-WD1(960*576/960*480), 19-1080i,20-
      576*576,
      //21-1536*1536, 22-1920*1920, 23-320*240, 24-720*720, 25-1024*768,26-1280*1280, 27-1600*600,28-2048*768,29-160*120,75-336*256,
      //78-384*256,79-384*216,80-320*256,82-320*192,83-512*384,127-480*272,128-512*272, 161-288*320, 162-144*176,163-480*640,164-240*320,165-120*160,166-
      576*720,167-720*1280,
      //168-576*960, 180-180*240,181-360*480,182-540*720,183-720*960,184-960*1280,185-1080*1440, 186-1920*1080, 187-640*3600xff-Auto(使用当前码流分辨率)--
      >0xff
    </pictureSize>
    <pictureQuality>
      <!--ro, req, enum, 图片质量系数, subType:string, [best#best,better#better,general#general]-->best
    </pictureQuality>
    <picSize>
      <!--ro, opt, enum, 图片大小, subType:int, [0#不限制,1#40K,2#80K,3#120K,4#200K], desc:默认为不限制-->0
    </picSize>
  </JPEGParam>
</SnapConfig>

```

69.35.2 配置抓拍参数

Request URL

PUT /ISAPI/AccessControl/SnapConfig

查询参数

无

请求报文

```
<?xml version="1.0" encoding="UTF-8"?>

<SnapConfig xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, 抓拍参数, attr:version{req, string, 协议版本}-->
  <snapTimes>
    <!--req, int, 线圈抓拍次数, range:[0,5]-->1
  </snapTimes>
  <snapWaitTime>
    <!--req, int, 抓拍等待时间, range:[0,6000]-->1
  </snapWaitTime>
  <intervalTimeList size="4">
    <!--req, array, 间隔列表, subType:object, attr:size{req, int, 数组元素总数}-->
    <intervalTime>
      <!--req, int, 连拍时间间隔, range:[0,6000]-->1
    </intervalTime>
  </intervalTimeList>
  <JPEGParam>
    <!--opt, object, 图片参数-->
    <pictureSize>
      <!--req, string, 图片分辨率,
      desc://0-CIF, 1-QCIF, 2-D1, 3-UXGA(1600x1200),4-SVGA(800x600),5-HD720p(1280x720), 6-VGA, 7-XVGA, 8-HD900p, 9-HD1080, 10-2560*1920,
      //11-1600*304, 12-2048*1536, 13-2448*2048,14-2448*1200, 15-2448*800, 16-XGA(1024*768), 17-SXGA(1280*1024),18-WD1(960*576/960*480), 19-1080i,20-
      576*576,
      //21-1536*1536, 22-1920*1920, 23-320*240, 24-720*720, 25-1024*768,26-1280*1200, 27-1600*600,28-2048*768,29-160*120,75-336*256,
      //78-384*256,79-384*216,80-320*256,82-320*192,83-512*384,127-480*272,128-512*272, 161-288*320, 162-144*176,163-480*640,164-240*320,165-120*160,166-
      576*720,167-720*1280,
      //168-576*960, 180-180*240,181-360*480,182-540*720,183-720*960,184-960*1280,185-1080*1440, 186-1920*1080, 187-640*360, 0xff-Auto(使用当前码流分辨
      率)-->0xff
    </pictureSize>
    <pictureQuality>
      <!--req, enum, 图片质量系数, subType:string, [best#best,better#better,general#general]-->best
    </pictureQuality>
    <picSize>
      <!--opt, enum, 图片大小, subType:int, [0#不限制,1#40K,2#80K,3#120K,4#200K], desc:默认为不限制-->0
    </picSize>
  </JPEGParam>
</SnapConfig>
```

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 响应消息, attr:version{ro, req, string, 版本}-->
  <requestURL>
    <!--ro, req, string, 请求的URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, 状态码, subType:int, [0#OK,1#OK,2#Device Busy,3#Device Error,4#Invalid Operation,5#Invalid XML Format,6#Invalid XML Content,7#Reboot
    Required]-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, 状态信息, subType:string, [OK#成功,Device Busy#设备忙,Device Error#设备异常,Invalid Operation#无效的操作,Invalid XML Format#无效的XML
    格式,Invalid XML Content#无效的XML内容,Reboot#设备重启]-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, 详细错误码英文描述, desc:详细错误码的英文描述-->OK
  </subStatusCode>
</ResponseStatus>
```

69.35.3 获取抓拍参数能力

Request URL

GET /ISAPI/AccessControl/SnapConfig/capabilities

查询参数

无

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<SnapConfig xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 抓拍参数, attr:version{req, string, 协议版本}-->
  <snapTimes min="0" max="5">
    <!--ro, req, int, 线圈抓拍次数, range:[0,5], attr:min{req, int, 最小值},max{req, int, 最大值}-->1
  </snapTimes>
  <snapWaitTime min="0" max="6000">
    <!--ro, req, int, 抓拍等待时间, range:[0,6000], attr:min{req, int, 最小值},max{req, int, 最大值}, desc:此字段未使用-->1
  </snapWaitTime>
  <intervalTimeList size="4">
    <!--ro, req, array, 间隔列表, subType:object, attr:size{req, int, 数组元素总数}-->
    <intervalTime min="0" max="6000">
      <!--ro, req, int, 连拍时间间隔, range:[0,6000], attr:min{req, int, 最小值},max{req, int, 最大值}-->1
    </intervalTime>
  </intervalTimeList>
  <JPEGParam>
    <!--ro, opt, object, 图片参数-->
    <pictureSize
      opt="0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,75,78,79,80,82,127,128,161,162,163,164,165,166,167,168,180,181,182,183,184,185,0xFF">
      <!--ro, req, string, 图片分辨率, attr:opt{req, string, 取值范围},
      desc://0-CIF, 1-QCIF, 2-D1, 3-UXGA(1600x1200),4-SVGA(800x600),5-HD720p(1280x720), 6-VGA, 7-XVGA, 8-HD900p, 9-HD1080, 10-2560*1920,
      //11-1600*304, 12-2048*1536, 13-2448*2048,14-2448*1280, 15-2448*800, 16-XGA(1024*768), 17-SXGA(1280*1024),18-HD1(960*576/960*480), 19-1080i,20-576*576,
      //21-1536*1536, 22-1920*1920, 23-320*240, 24-720*720, 25-1024*768,26-1280*1280, 27-1600*600,28-2048*768,29-160*120,75-336*256,
      //78-384*256,79-384*216,80-320*256,82-320*192,83-512*384,127-480*272,128-512*272, 161-288*320, 162-144*176,163-480*640,164-240*320,165-120*160,166-576*720,167-720*1280,
      //168-576*960, 180-180*240,181-360*480,182-540*720,183-720*960,184-960*1280,185-1080*1440. 0xFF-Auto(使用当前码流分辨率)-->0xFF
    </pictureSize>
    <pictureQuality opt="best, better, general">
      <!--ro, req, enum, 图片质量系数, subType:string, [best#best,better#better,general#general], attr:opt{req, string, 取值范围}-->best
    </pictureQuality>
    <picSize opt="0,1,2,3,4">
      <!--ro, opt, enum, 图片大小, subType:int, [0#不限制,1#40K,2#80K,3#120K,4#200K], attr:opt{req, string, 取值范围}-->1
    </picSize>
  </JPEGParam>
  <isSupportSetSnapConfig>
    <!--ro, opt, bool, 是否支持抓拍参数配置, desc:支持时返回true, 不支持时不返回该字段或返回false-->true
  </isSupportSetSnapConfig>
</SnapConfig>
```

69.36 访问控制自定义语音管理

69.36.1 导入自定义语音文件

Request URL

POST /ISAPI/AccessControl/customAudio/addCustomAudio?format=json

查询参数

无

请求报文

```
{
  "customAudioType": "callCenter",
  /*req, enum, 自定义语音文件类型, subType:string, [callCenter#正在呼叫中心,centerBusy#对方正忙,centerRefused#对方拒绝本次呼叫,centerOverTime#对方无人接听,swipeCard#请刷卡,thanks#谢谢,callAgain#请稍后再试,verifyFailed#认证失败,verifySuccess#认证成功,doorOpened#门已打开,wearSafetyHelmet#请佩戴安全帽,wearMask#请佩戴口罩,abnormalTemperature#体温异常]*/
  "filePathType": "URL",
  /*req, enum, 文件路径类型, subType:string, [simpleStorage#简单存储,url#URL,localPath#本地存储,binary#二进制]*/
  "customAudioURL": "test"
  /*opt, string, 自定义语音文件URL, range:[1,256], desc:当filePathType为URL时必须*/
}
```

参数名称	参数值	参数类型(Content-Type)	Content-ID	文件名称(filename)	描述
CustomAudioInfo	[报文内容]	application/json	--	--	--
File_Name	[音频文件]	audio/wav		设备未实现	音频文件

说明： 此协议通过表单格式传输，表单框架详情说明见章节4.5.1.4，如下实例：

```
--<frontier>
Content-Disposition: form-data; name=参数名称;filename=文件名称
Content-Type: 参数类型
Content-Length: ****
Content-ID: 内容ID
参数值
```

- 参数名称：表单单元Headers中Content-Disposition的name属性，表示此表单单元的名称
- 参数类型（Content-Type）：表单单元Headers中Content-Type属性
- 文件名称（filename）：表单单元Headers中Content-Disposition的filename属性，仅在本表单单元传输数据为文件类型存在，表示表单单元Body的文件名
- 参数值：表单单元的body内容

响应报文

```
{
  "statusCode": 1,
  /*ro, opt, int, 状态码, desc:无法用1表示时（1表示成功且无特殊状态）必须返回*/
  "statusString": "OK",
  /*ro, opt, string, 状态描述, range:[1,64], desc:无法用OK表示时（OK表示成功且无特殊状态）必须返回*/
  "subStatusCode": "ok",
  /*ro, opt, string, 子状态码, range:[1,64], desc:无法用ok表示（ok表示成功且无特殊状态）必须返回*/
  "errorCode": 1,
  /*ro, opt, int, 错误码, desc:当statusCode不为1时，与subStatusCode对应*/
  "errorMsg": "ok"
  /*ro, opt, string, 错误信息, desc:当statusCode不为1时，必须返回,解释信息在协议约束中，允许设备在后续的版本迭代中，进行优化丰富提升（不限制死）*/
}
```

69.36.2 获取自定义语音配置能力

Request URL

GET /ISAPI/AccessControl/customAudio/capabilities?format=json

查询参数

无

请求报文

无

响应报文

```

{
  "customAudioType": {
    /*ro, req, object, 自定义语音文件类型*/
    "@opt": ["callCenter", "centerBusy", "centerRefused", "centerOverTime", "swipeCard", "thanks", "callAgain", "verifyFailed", "verifySuccess",
    "doorOpened", "wearSafetyHelmet", "wearMask", "abnormalTemperature"]
    /*ro, opt, array, 可选项, subType:string, desc:callCenter-正在呼叫中心, centerBusy-对方正忙, 请稍后呼叫, centerRefused-对方拒绝本次呼
    叫, centerOverTime-对方无人接听, 请稍后呼叫, swipeCard-请刷卡, thanks-谢谢, callAgain-请稍后再试, verifyFailed-认证失败, verifySuccess-认证成功, doorOpened-
    门已打开, wearSafetyHelmet-请佩戴安全帽, wearMask-请佩戴口罩, abnormalTemperature-体温异常*/
  },
  "filePathType": {
    /*ro, req, object, 文件路径类型, desc:simpleStorage-简单存储, URL-URL, localPath-本地存储, binary-二进制*/
    "@opt": ["simpleStorage", "URL", "localPath", "binary"]
    /*ro, opt, array, 可选项, subType:string*/
  },
  "customAudioURL": {
    /*ro, opt, object, 自定义语音文件URL, desc:当filePathType为URL时必须*/
    "@min": 1,
    /*ro, opt, int, 自定义语音文件URL长度最小值*/
    "@max": 256
    /*ro, opt, int, 自定义语音文件URL长度最大值*/
  },
  "customAudioFormat": {
    /*ro, opt, object, 自定义语音文件格式, desc:wav-wav格式*/
    "@opt": ["wav"]
    /*ro, opt, array, 可选项, subType:string*/
  },
  "customAudioSize": {
    /*ro, opt, object, 自定义语音文件大小, desc:单位 KB*/
    "@min": 0,
    /*ro, opt, int, 最小值*/
    "@max": 512
    /*ro, opt, int, 最大值*/
  },
  "customAudioSearchType": {
    /*ro, opt, object, 自定义语音文件状态查询类型*/
    "@size": 13,
    /*ro, opt, int, 查询类型最大个数, desc:该值随着自定义语音文件类型的增加而增大*/
    "@opt": ["all", "callCenter", "centerBusy", "centerRefused", "centerOverTime", "swipeCard", "thanks", "callAgain", "verifyFailed", "verifySuccess",
    "doorOpened", "wearSafetyHelmet", "wearMask", "abnormalTemperature"]
    /*ro, opt, array, 可选项, subType:string, desc:all-查询全部类型, callCenter-正在呼叫中心, centerBusy-对方正忙, 请稍后呼叫, centerRefused-对方拒绝本
    次呼叫, centerOverTime-对方无人接听, 请稍后呼叫, swipeCard-请刷卡, thanks-谢谢, callAgain-请稍后再试, verifyFailed-认证失败, verifySuccess-认证成功,
    doorOpened-门已打开, wearSafetyHelmet-请佩戴安全帽, wearMask-请佩戴口罩, abnormalTemperature-体温异常*/
  }
}

```

69.36.3 删除自定义语音文件

Request URL

POST /ISAPI/AccessControl/customAudio/deleteCustomAudio?format=json

查询参数

无

请求报文

```

{
  "customAudioType": "callCenter"
  /*req, enum, 自定义语音文件类型, subType:string, [callCenter#正在呼叫中心, centerBusy#对方正忙, centerRefused#对方拒绝本次呼叫, centerOverTime#对方无人接
  听, swipeCard#请刷卡, thanks#谢谢, callAgain#请稍后再试, verifyFailed#认证失败, verifySuccess#认证成功, doorOpened#门已打开, wearSafetyHelmet#请佩戴安全
  帽, wearMask#请佩戴口罩, abnormalTemperature#体温异常]*/
}

```

响应报文

```

{
  "statusCode": 1,
  /*ro, opt, int, 状态码, desc:无法用1表示时（1表示成功且无特殊状态）必须返回*/
  "statusString": "OK",
  /*ro, opt, string, 状态描述, range:[1,64], desc:无法用OK表示时（OK表示成功且无特殊状态）必须返回*/
  "subStatusCode": "ok",
  /*ro, opt, string, 子状态码, range:[1,64], desc:无法用ok表示（ok表示成功且无特殊状态）必须返回*/
  "errorCode": 1,
  /*ro, opt, int, 错误码, desc:当statusCode不为1时, 与subStatusCode对应*/
  "errorMsg": "ok"
  /*ro, opt, string, 错误信息, desc:当statusCode不为1时, 必须返回,解释信息在协议约束中, 允许设备在后续的版本迭代中, 进行优化丰富提升（不限制死）*/
}

```

69.36.4 查询指定自定义语音文件下发状态

Request URL

POST /ISAPI/AccessControl/customAudio/searchCustomAudioStatus?format=json

查询参数

无

请求报文

```
{
  "customAudioSearchType": ["callCenter", "centerBusy"]
  /*req, array, 自定义语音文件查询类型, subType:string, desc:all-查询全部类型, callCenter#正在呼叫中心,centerBusy#对方正忙,centerRefused#对方拒绝本次呼叫,centerOverTime#对方无人接听,swipeCard#请刷卡,thanks#谢谢,callAgain#请稍后再试,verifyFailed#认证失败,verifySuccess#认证成功,doorOpened#门已打开,wearSafetyHelmet#请佩戴安全帽,wearMask#请佩戴口罩,abnormalTemperature#体温异常*/
}
```

响应报文

```
{
  "CustomAudioStatusList": [
    /*ro, req, array, 自定义语音文件状态列表, subType:object*/
    {
      "customAudioType": "callCenter",
      /*ro, req, enum, 自定义语音文件类型, subType:string, [callCenter#正在呼叫中心,centerBusy#对方正忙,centerRefused#对方拒绝本次呼叫,centerOverTime#对方无人接听,swipeCard#请刷卡,thanks#谢谢,callAgain#请稍后再试,verifyFailed#认证失败,verifySuccess#认证成功,doorOpened#门已打开,wearSafetyHelmet#请佩戴安全帽,wearMask#请佩戴口罩,abnormalTemperature#体温异常*/
      "status": "normal",
      /*ro, req, enum, 状态, subType:string, [normal#已下发,abnormal#未下发]*/
    }
  ]
}
```

69.37 人员及凭证管理

69.37.1 获取虹膜信息参数能力

Request URL

GET /ISAPI/AccessControl/IrisInfo/capabilities?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "IrisInfoCap": {
    /*ro, req, object, 能力*/
    "supportFunction": {
      /*ro, req, object, 支持虹膜的相关功能, desc:post-创建,delete-删除,put-修改,get-批量查询,count-虹膜数量查询,setup-设置,singleGet-单个查询*/
      "@opt": "post,delete,put,get,count,setup,singleGet"
      /*ro, req, string, 取值范围*/
    },
    "IrisInfoSearchCond": {
      /*ro, req, object, 虹膜搜索条件*/
      "searchID": {
        /*ro, req, object, 搜索记录唯一标识*/
        "@min": 1,
        /*ro, req, int, 取值范围*/
        "@max": 1
        /*ro, req, int, 取值范围*/
      },
      "maxResults": {
        /*ro, req, object, 本次协议调用可获取的最大记录数*/
        "@min": 1,
        /*ro, req, int, 取值范围*/
        "@max": 1
        /*ro, req, int, 取值范围*/
      },
      "EmployeeNoList": {
        /*ro, opt, object, 工号列表, desc:当该字段为空或不存在时,表示查询所有人员*/
        "@size": 1,
        /*ro, req, int, list支持的最大元素个数*/
        "employeeNo": {
          /*ro, opt, object, 工号 (人员ID) */
          "@min": 1,
```

```

        /*ro, req, int, 取值范围*/
        "@max": 1
        /*ro, req, int, 取值范围*/
    }
},
"IrisInfoDelCond": {
/*ro, opt, object, 删除条件*/
"EmployeeNoList": {
/*ro, opt, object, 工号列表*/
"@size": 1,
/*ro, req, int, list支持的最大元素个数*/
"employeeNo": {
/*ro, opt, object, 工号（人员ID）*/
"@min": 1,
/*ro, req, int, 取值范围*/
"@max": 1
/*ro, req, int, 取值范围*/
},
"irisIdList": {
/*ro, opt, object, 虹膜序号*/
"@size": 2,
/*ro, req, int, 数组支持的最大个数*/
"@min": 1,
/*ro, req, int, 数组中序号的取值范围*/
"@max": 1
/*ro, req, int, 数组中序号的取值范围*/
}
}
},
"employeeNo": {
/*ro, opt, object, 工号（人员ID）*/
"@min": 1,
/*ro, req, int, 取值范围*/
"@max": 1
/*ro, req, int, 取值范围*/
},
"eyesType": {
/*ro, req, object, 采集的眼睛类型, desc:单目下发时,设备处理IrisData中的第一组数据;双目下发时,IrisData两组数据均成功才成功,如果存在失败,则返回哪张失
败*/
"@opt": ["monocular", "binocular"]
/*ro, req, array, 取值范围, subType:string*/
},
"irisPicDataType": {
/*ro, req, object, 虹膜图片数据传输方式*/
"@opt": ["url", "binary"]
/*ro, req, array, 取值范围, subType:string*/
},
"IrisData": {
/*ro, req, object, 虹膜数据*/
"id": {
/*ro, req, object, 虹膜数据序号, desc:与响应结果中AttachInfo节点内的id一一对应,AttachInfo节点内的id为string类型是为了兼容其他业务存在字符串的情
况,本协议中使用"1"表示即可;id仅支持1和2,否则失败*/
"@min": 1,
/*ro, req, int, 取值范围*/
"@max": 1
/*ro, req, int, 取值范围*/
},
"irisModelData": {
/*ro, opt, object, 虹膜模板数据, desc:当下发模板+图片时,以模板数据为准,且不查重,设备不判断模板有效性*/
"@min": 1,
/*ro, req, int, 取值范围*/
"@max": 1
/*ro, req, int, 取值范围*/
},
"iriPicData": {
/*ro, opt, object, 虹膜图片数据, desc:当irisPicDataType为url时,此处为url字符串;当irisPicDataType为binary时,此处与图片的Content-ID严格对应*/
"@min": 1,
/*ro, req, int, 取值范围*/
"@max": 1
/*ro, req, int, 取值范围*/
}
},
"picFormat": "bmp",
/*ro, opt, enum, 设备支持的虹膜图片格式类型, subType:string, [bmp#bmp格式]*/
"maxRecordNum": 10000
/*ro, req, int, 支持的最大记录条数（虹膜数量）*/
}
}
}

```

69.37.2 获取指纹信息参数能力

Request URL

GET /ISAPI/AccessControl/FingerPrintCfg/capabilities?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "FingerPrintCfg": {
    /*ro, opt, object, 指纹配置*/
    "employeeNo": {
      /*ro, req, object, 指纹关联的工号*/
      "@min": 1,
      /*ro, opt, int, 最小值*/
      "@max": 32
      /*ro, opt, int, 最大值*/
    },
    "enableCardReader": {
      /*ro, req, object, 需要下发指纹的读卡器*/
      "@min": 1,
      /*ro, opt, int, 最小值*/
      "@max": 512
      /*ro, opt, int, 最大值*/
    },
    "fingerPrintID": {
      /*ro, req, object, 手指编号*/
      "@min": 1,
      /*ro, opt, int, 最小值*/
      "@max": 10
      /*ro, opt, int, 最大值*/
    },
    "fingerType": {
      /*ro, req, object, 指纹类型*/
      "@opt": "normalFP"
      /*ro, opt, enum, 指纹类型, subType:string, [normalFP#普通指纹,hijackFP#胁迫指纹,patrolFP#巡更指纹,superFP#超级指纹,dismissingFP#解除指纹]*/
    },
    "leaderFP": {
      /*ro, opt, object, 是否有首次认证功能*/
      "@min": 1,
      /*ro, opt, int, 最小值*/
      "@max": 32
      /*ro, opt, int, 最大值*/
    },
    "checkEmployeeNo": "true,false",
    /*ro, opt, string, 设备是否进行工号（人员ID）存在校验*/
    "StatusList": {
      /*ro, opt, object, 状态*/
      "id": {
        /*ro, opt, object, 读卡器编号*/
        "@min": 1,
        /*ro, opt, int, 最小值*/
        "@max": 512
        /*ro, opt, int, 最大值*/
      },
      "cardReaderRecvStatus": {
        /*ro, opt, object, 指纹读卡器状态*/
        "@opt": "0,1,2,3,4,5,6,7,8,10"
        /*ro, opt, string, 指纹读卡器状态*/
      },
      "errorMsg": {
        /*ro, opt, object, 下发错误信息*/
        "@min": 1,
        /*ro, opt, int, 最小值*/
        "@max": 32
        /*ro, opt, int, 最大值*/
      }
    },
    "totalStatus": {
      /*ro, req, object, 下发总的状态*/
      "@opt": "0,1"
      /*ro, opt, string, 下发总的状态: 0-当前指纹未下完所有读卡器*/
    },
    "isSupportFingerCover": true,
    /*ro, opt, bool, 指纹下发是否支持指纹覆盖（相同人员ID*/
    "isSupportSetUp": true,
    /*ro, opt, bool, 是否支持指纹参数设置（支持时: 返回true*/
    "maxRecordNum": 1,
    /*ro, opt, int, 主机指纹容量, range:[0,10000], desc:指纹支持存储在主机时, 使用该字段返回指纹容量*/
    "isSupportCount": true
    /*ro, opt, bool, 是否支持查询所有人员的指纹数量, desc:针对智能主机, 指纹可存储在主机, 上层查询主机的已存在指纹数
    量/ISAPI/AccessControl/FingerPrint/Count?format=json*/
  }
}
```

69.37.3 配置指纹信息

Request URL

POST /ISAPI/AccessControl/FingerPrint/SetUp?format=json&security=<security>&iv=<iv>

查询参数

参数名称	参数类型	描述
security	string	没有security表示数据不加密； security=1,表示报文中的employeeNo、fingerData字段AES128加密； security=2,表示报文中的employeeNo、fingerData字段AES256加密； fingerData字段是否支持加密见门禁总能力的isSupportFingerDataEncryption字段
iv	string	iv表示初始化向量，在security=1或者2时为必须参数;

请求报文

```
{
  "FingerPrintCfg": {
    /*opt, object, 指纹配置*/
    "employeeNo": "test",
    /*req, string, 指纹关联的工号, range:[1,32]*/
    "enableCardReader": [1, 3, 5],
    /*req, array, 需要下发指纹的读卡器, subType:int, desc:[1,3,5]代表需要下发给读卡器1、读卡器3、读卡器5*/
    "fingerPrintID": 1,
    /*req, int, 手指编号, range:[1,10]*/
    "deleteFingerPrint": true,
    /*opt, bool, 是否删除该指纹, desc:true-是（只有删除该指纹时，才填写该字段；新增或修改指纹时，不填写该字段）*/
    "fingerType": "normalFP",
    /*req, enum, 指纹类型, subType:string, [normalFP#普通指纹,hijackFP#胁迫指纹,patrolFP#巡更指纹,superFP#超级指纹,dismissingFP#解除指纹]*/
    "fingerData": "test",
    /*opt, string, 指纹数据, desc:base64处理, 如果需要加密，则指纹数据仅做一次base64处理，直接进行aes加密*/
    "leaderFP": [1, 3, 5],
    /*opt, array, 是否有首次认证功能, subType:int, range:[1,5], desc:（[1,3,5]代表指纹对于门1、门3、门5有首次认证功能）*/
    "checkEmployeeNo": true
    /*opt, bool, 设备是否进行工号（人员ID）存在校验*/
  }
}
```

响应报文

```
{
  "FingerPrintStatus": {
    /*ro, opt, object, 指纹状态*/
    "status": "success",
    /*ro, opt, enum, 状态, subType:string, [success#成功,failure#失败], desc:（对于实际下发指纹数据到读卡器时，不会返回该字段。如果仅是修改指纹参数（不涉及到指纹数据下发到读卡器）或删除指纹，才会返回该字段）*/
    "StatusList": [
      /*ro, opt, array, 状态列表, subType:object, desc:（对于实际下发指纹数据到读卡器时，才会返回该列表。如果仅是修改指纹参数（不涉及到指纹数据下发到读卡器）或删除指纹，则不会返回该列表）*/
      {
        "id": 1,
        /*ro, opt, int, 读卡器编号*/
        "cardReaderRecvStatus": 1,
        /*ro, opt, enum, 指纹读卡器状态, subType:int, [0#失败,1#成功,2#该指纹模组不在线,3#重试或指纹质量差,4#内存已满,5#已存在该指纹,7#非法指纹ID,8#该指纹模组无需配置,10#指纹读卡器版本过低]*/
        "errorMsg": "test"
        /*ro, opt, string, 下发错误信息, range:[1,32], desc:当cardReaderRecvStatus为5时，表示已存在指纹对应的工号（人员ID）*/
      }
    ]
  }
}
```

69.37.4 获取清空设备图片能力

Request URL

GET /ISAPI/AccessControl/ClearPictureCfg/capabilities?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "ClearPictureCfgCap": {
    /*ro, req, object, 清空设备图片能力节点*/
    "ClearFlags": {
      /*ro, opt, object, 清空标志*/
      "facePicture": {
        /*ro, opt, object, 清空人脸底图*/
        "@opt": [true, false]
        /*ro, req, array, 取值范围, subType:bool*/
      },
      "capOrVerifyPicture": {
        /*ro, opt, object, 清空认证或抓拍图片*/
        "@opt": [true, false]
        /*ro, req, array, 取值范围, subType:bool*/
      },
      "irisPicture": {
        /*ro, opt, object, 清空虹膜底图*/
        "@opt": [true, false]
        /*ro, req, array, 取值范围, subType:bool*/
      }
    }
  }
}
```

69.37.5 获取人员信息及权限删除能力

Request URL

GET /ISAPI/AccessControl/UserInfoDetail/Delete/capabilities?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "UserInfoDetail": {
    /*ro, opt, object, 用户信息*/
    "mode": {
      /*ro, req, object, 删除模式*/
      "@opt": "all,byEmployeeNo"
      /*ro, opt, string, 删除模式, desc:all-删除所有,byEmployeeNo-编号*/
    },
    "EmployeeNoList": {
      /*ro, opt, object, 工号列表, desc:当mode为byEmployeeNo时, 该字段有效*/
      "maxSize": 50,
      /*ro, opt, int, 最大数量*/
      "employeeNo": {
        /*ro, opt, object, 工号*/
        "@min": 1,
        /*ro, opt, int, 最大值*/
        "@max": 32
        /*ro, opt, int, 最小值*/
      }
    }
  }
}
```

69.37.6 添加卡信息

Request URL

POST /ISAPI/AccessControl/CardInfo/Record?format=json&security=<security>&iv=<iv>

查询参数

参数名称	参数类型	描述
security	string	没有security表示数据不加密； security=1,表示报文中的employeeNo、cardNo字段AES128加密； security=2,表示报文中的employeeNo、cardNo字段AES256加密；
iv	string	iv表示初始化向量，在security=1或者2时为必须参数；

请求报文

```
{
  "CardInfo": {
    /*req, object, 卡信息*/
    "employeeNo": "test",
    /*req, string, 工号（人员ID）*/
    "cardNo": "1234567890",
    /*req, string, 卡号*/
    "cardType": "normalCard",
    /*opt, enum, 卡类型, subType:string, [normalCard#普通卡,patrolCard#巡更卡,hijackCard#胁迫卡,superCard#超级卡,dismissingCard#解除卡,emergencyCard#应
急管理卡（用于授权临时卡权限）*/
    "leaderCard": "1,3,5",
    /*opt, string, 是否有首次认证功能, desc:（表示该卡对于门1、门3、门5有首次认证功能）*/
    "checkCardNo": true,
    /*opt, bool, 设备是否进行卡重复添加校验,
    desc:如果不配置该字段, 则设备默认进行卡重复校验
    如果确认设备端不存在任何卡信息, 可将其置为false, 则设备不进行重复校验, 这样会加快下发速度; 如果不确认, 则不建议配置该字段*/
    "checkEmployeeNo": true,
    /*opt, bool, 设备是否进行工号存在校验,
    desc:如果不配置该字段, 则设备默认进行工号存在校验
    该字段配置为false时, 设备不进行工号（人员ID）存在校验（可加快设备下发卡的速度）; 该字段配置为true或不配置该字段时, 设备进行工号（人员ID）存在校验,
    如不对处理速度有强烈要求, 建议使用该方式*/
    "operateType": "byTerminal",
    /*opt, string, 操作类型*/
    "terminalNoList": [1]
    /*opt, array, 终端ID列表, subType:int, desc:type为byTerminal,byTerminal0ng时必须,终端ID列表（目前仅支持单个终端*/
  }
}
```

响应报文

```
{
  "statusCode": 1,
  /*ro, opt, int, 状态码, desc:无法用1表示时（1表示成功且无特殊状态）必须返回*/
  "statusString": "OK",
  /*ro, opt, string, 状态描述, range:[1,64], desc:无法用OK表示时（OK表示成功且无特殊状态）必须返回*/
  "subStatusCode": "ok",
  /*ro, opt, string, 子状态码, range:[1,64], desc:无法用ok表示（ok表示成功且无特殊状态）必须返回*/
  "errorCode": 1,
  /*ro, opt, int, 错误码, desc:当statusCode不为1时, 与subStatusCode对应*/
  "errorMsg": "ok"
  /*ro, opt, string, 错误信息, desc:当statusCode不为1时, 必须返回,解释信息在协议约束中, 允许设备在后续的版本迭代中, 进行优化丰富提升（不限制死）*/
}
```

69.37.7 删除人员信息

Request URL

PUT /ISAPI/AccessControl/UserInfo/Delete?format=json&security=<security> &iv=<iv>

查询参数

参数名称	参数类型	描述
security	string	没有security表示数据不加密; security=1,表示报文中的employeeNo字段AES128加密; security=2,表示报文中的employeeNo字段AES256加密;
iv	string	iv表示初始化向量, 在security=1或者2时为必须参数;

请求报文

```
{
  "UserInfoDelCond": {
    /*req, object, 删除条件*/
    "EmployeeNoList": [
      /*opt, array, 人员ID列表, subType:object*/
      {
        "employeeNo": "test"
        /*req, string, 工号（人员ID），range:[1,32]*/
      }
    ],
    "operateType": "byTerminal",
    /*opt, string, 操作类型*/
    "terminalNoList": [1]
    /*opt, array, 终端ID列表, subType:int, desc:type为byTerminal,byTerminalOrg时必须填,终端ID列表（目前仅支持单个终端）*/
  }
}
```

响应报文

```
{
  "statusCode": 1,
  /*ro, opt, int, 状态码, desc:无法用1表示时（1表示成功且无特殊状态）必须返回*/
  "statusString": "OK",
  /*ro, opt, string, 状态描述, range:[1,64], desc:无法用OK表示时（OK表示成功且无特殊状态）必须返回*/
  "subStatusCode": "ok",
  /*ro, opt, string, 子状态码, range:[1,64], desc:无法用ok表示（ok表示成功且无特殊状态）必须返回*/
  "errorCode": 1,
  /*ro, opt, int, 错误码, desc:当statusCode不为1时，与subStatusCode对应*/
  "errorMsg": "ok"
  /*ro, opt, string, 错误信息, desc:当statusCode不为1时，必须返回,解释信息在协议约束中，允许设备在后续的版本迭代中，进行优化丰富提升（不限制死）*/
}
```

69.37.8 获取人员数据异步下发能力集

Request URL

GET /ISAPI/AccessControl/UserInfo/asyncImportDatasTasks/capabilities?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "AsyncImportDdatasCap": {
    /*ro, req, object, 人员数据异步下发能力节点*/
    "taskID": {
      /*ro, req, object, 下发任务ID字符串长度, desc:最大长度64字符*/
      "@min": 1,
      /*ro, opt, int, 最小值*/
      "@max": 64
      /*ro, opt, int, 最大值*/
    },
    "taskNum": 1,
    /*ro, req, int, 设备同时支持的任务数目*/
    "URL": {
      /*ro, req, object, 数据文件的URL长度, desc:最大长度256字符*/
      "@min": 1,
      /*ro, opt, int, 最小值*/
      "@max": 256
      /*ro, opt, int, 最大值*/
    },
    "singleFileMaxSize": 1,
    /*ro, req, int, 单个人员信息数据文件大小范围, desc:单位: KB*/
    "employeeNo": {
      /*ro, opt, object, 工号（人员ID）*/
      "@min": 1,
      /*ro, opt, int, 最小值*/
      "@max": 32
      /*ro, opt, int, 最大值*/
    },
    "deleteUser": {
      /*ro, opt, object, 删除人员信息*/
      "@opt": [true, false]
      /*ro, opt, array, 可选项, subType:bool*/
    },
    "name": {
      /*ro, opt, object, 姓名*/
      "@min": 1,

```

```

/*ro, opt, int, 姓名长度最小值*/
"@max": 32
/*ro, opt, int, 姓名长度最大值*/
},
"userType": {
/*ro, opt, object, 人员类型, desc:normal-普通人（主人）， visitor-来宾（访客）， blacklist-黑名单人*/
"@opt": ["normal", "visitor", "blacklist"]
/*ro, opt, array, 可选项, subType:string*/
},
"Valid": {
/*ro, opt, object, 有效时间段*/
"enable": {
/*ro, opt, object, 使能有效期, desc:false-不使能, true-使能*/
"@opt": [true, false]
/*ro, opt, array, 可选项, subType:bool*/
},
"beginTime": "1970-01-01T00:00:00+08:00",
/*ro, opt, string, 有效期起始时间*/
"endTime": "2037-12-31T23:59:59+08:00"
/*ro, opt, string, 有效期结束时间*/
},
},
"password": {
/*ro, opt, object, 密码*/
"@min": 8,
/*ro, opt, int, 密码长度最小值*/
"@max": 16
/*ro, opt, int, 密码长度最大值*/
},
"RightPlan": {
/*ro, opt, object, 权限计划*/
"doorNo": {
/*ro, opt, object, 门编号*/
"@min": 1,
/*ro, opt, int, 最小值*/
"@max": 1
/*ro, opt, int, 最大值*/
},
"planTemplateNo": {
/*ro, opt, object, 计划模板编号范围, desc:即每个编号值的范围*/
"@min": 1,
/*ro, opt, int, 最小值*/
"@max": 1
/*ro, opt, int, 最大值*/
},
"planTemplateNum": {
/*ro, opt, object, 计划模板编号可配置的数目范围, desc:即可配置多少个*/
"@min": 0,
/*ro, opt, int, 最小值*/
"@max": 1
/*ro, opt, int, 最大值*/
}
},
"localUIRight": {
/*ro, opt, object, 是否具有设备本地UI访问权限, desc:true-有权限, false-无权限*/
"@opt": [true, false]
/*ro, opt, array, 可选项, subType:bool*/
},
"userVerifyMode": {
/*ro, opt, object, 人员验证方式,
desc:人员验证方式的优先级高于读卡器验证方式。
cardAndPw-刷卡+密码, card-刷卡, cardOrPw-刷卡或密码, fp-指纹, fpAndPw-指纹+密码, fpOrCard-指纹或刷卡, fpAndCard-指纹+刷卡, fpAndCardAndPw-指纹+刷卡+密码, faceOrfpOrCardOrPw-人脸或指纹或刷卡或密码, faceAndfp-人脸+指纹, faceAndPw-人脸+密码, faceAndCard-人脸+刷卡, face-人脸, employeeNoAndPw-工号+密码, fpOrPw-指纹或密码, employeeNoAndFp-工号+指纹, employeeNoAndFpAndPw-工号+指纹+密码, faceAndFpAndCard-人脸+指纹+刷卡, faceAndPwAndFp-人脸+密码+指纹, employeeNoAndFace-工号+人脸, faceOrfaceAndCard-人脸或人脸+刷卡, fpOrface-指纹或人脸, cardOrfaceOrPw-刷卡或人脸或密码, cardOrFace-刷卡或人脸, cardOrFaceOrFp-刷卡或人脸或指纹*/
"@opt": ["cardAndPw", "card", "cardOrPw", "fp", "fpAndPw", "fpOrCard", "fpAndCard", "fpAndCardAndPw", "faceOrfpOrCardOrPw", "faceAndFp", "faceAndPw", "faceAndCard", "face", "employeeNoAndPw", "fpOrPw", "employeeNoAndFp", "employeeNoAndFpAndPw", "faceAndFpAndCard", "faceAndPwAndFp", "employeeNoAndFace", "faceOrfaceAndCard", "fpOrface", "cardOrfaceOrPw", "cardOrFace", "cardOrFaceOrFp"]
/*ro, opt, array, 可选项, subType:string*/
},
},
"AccessControl": {
/*ro, opt, object, 门禁信息*/
"doorRightNum": {
/*ro, opt, object, 门权限可配数目*/
"@min": 0,
/*ro, opt, int, 最小值*/
"@max": 1
/*ro, opt, int, 最大值*/
},
"closeDelayEnabled": {
/*ro, opt, object, 是否关门延迟, desc:true-是, false-否*/
"@opt": [true, false]
/*ro, opt, array, 可选项, subType:bool*/
},
},
"belongGroupNo": {
/*ro, opt, object, 所属群组编号范围, desc:即每个编号值的范围*/
"@min": 1,
/*ro, opt, int, 最小值*/
"@max": 1
/*ro, opt, int, 最大值*/
},
"belongGroupNum": {
/*ro, opt, object, 所属群组编号可配置的数目范围, desc:即可配置多少个*/
"@min": 0

```

```

        /*ro, opt, int, 最小值*/
        "@max": 1
        /*ro, opt, int, 最大值*/
    },
    "maxOpenDoorTime": {
        /*ro, opt, object, 最大认证次数, desc:0为无次数限制（默认为0）*/
        "@min": 0,
        /*ro, opt, int, 最小值*/
        "@max": 1
        /*ro, opt, int, 最大值*/
    },
    "openDoorTime": {
        /*ro, opt, object, 已认证次数*/
        "@min": 1,
        /*ro, opt, int, 最小值*/
        "@max": 1
        /*ro, opt, int, 最大值*/
    },
    "roomNumber": {
        /*ro, opt, object, 房间号*/
        "@min": 1,
        /*ro, opt, int, 最小值*/
        "@max": 1
        /*ro, opt, int, 最大值*/
    },
    "floorNumber": {
        /*ro, opt, object, 层号*/
        "@min": 1,
        /*ro, opt, int, 最小值*/
        "@max": 1
        /*ro, opt, int, 最大值*/
    }
},
"FaceInfo": {
    /*ro, opt, object, 人脸信息*/
    "deleteAllFace": {
        /*ro, opt, object, 删除所有人脸信息*/
        "@opt": [true, false]
        /*ro, opt, array, 可选项, subType:bool*/
    },
    "List": {
        /*ro, opt, object, 列表*/
        "FDID": "test",
        /*ro, opt, string, 人脸库ID*/
        "faceID": {
            /*ro, opt, object, 人脸编号*/
            "@min": 1,
            /*ro, opt, int, 最小值*/
            "@max": 1
            /*ro, opt, int, 最大值*/
        },
        "deleteFace": {
            /*ro, opt, object, 删除人脸信息*/
            "@opt": [true, false]
            /*ro, opt, array, 可选项, subType:bool*/
        },
        "isSupportModelData": true
        /*ro, opt, bool, 是否支持人脸建模数据下发, desc:支持时返回true, 不支持时不返回该字段*/
    }
},
"CardInfo": {
    /*ro, opt, object, 卡信息*/
    "deleteAllCard": {
        /*ro, opt, object, 删除所有卡信息*/
        "@opt": [true, false]
        /*ro, opt, array, 可选项, subType:bool*/
    },
    "List": {
        /*ro, opt, object, 列表*/
        "cardNo": {
            /*ro, opt, object, 卡号*/
            "@min": 1,
            /*ro, opt, int, 最小值*/
            "@max": 32
            /*ro, opt, int, 最大值*/
        },
        "deleteCard": {
            /*ro, opt, object, 删除卡信息*/
            "@opt": [true, false]
            /*ro, opt, array, 可选项, subType:bool*/
        },
        "cardType": {
            /*ro, opt, object, 卡类型, desc:normalCard-普通卡, patrolCard-巡更卡, hijackCard-胁迫卡, superCard-超级卡, dismissingCard-解除
卡, emergencyCard-应急管理卡（用于授权临时卡权限, 本身不能开门）*/
            "@opt": ["normalCard", "patrolCard", "hijackCard", "superCard", "dismissingCard", "emergencyCard"]
            /*ro, opt, array, 可选项, subType:string*/
        },
        "AccessControl": {
            /*ro, opt, object, 门禁信息*/
            "leaderCardNum": {
                /*ro, opt, object, 首次认证功能可配置的门的数目*/
                "@min": 0,
                /*ro, opt, int, 最小值*/
            }
        }
    }
}

```



```

enum : post, delete, put, get, setup
/*ro, req, string, 可选项*/
},
"CardInfoSearchCond": {
/*ro, opt, object, 查找条件*/
"searchID": {
/*ro, opt, object, 查找ID*/
"@min": 1,
/*ro, req, int, 最小值*/
"@max": 36
/*ro, req, int, 最大值*/
},
"maxResults": {
/*ro, req, object, 本次协议调用可获取的最大记录数*/
"@min": 1,
/*ro, req, int, 最小值*/
"@max": 30
/*ro, req, int, 最大值*/
},
"EmployeeNoList": {
/*ro, opt, object, 员工编号列表*/
"maxSize": 56,
/*ro, req, int, 最大值*/
"employeeNo": {
/*ro, opt, object, 工号（人员ID）*/
"@min": 1,
/*ro, req, int, 最小值*/
"@max": 32
/*ro, req, int, 最大值*/
}
},
"CardNoList": {
/*ro, opt, object, 卡号列表*/
"maxSize": 56,
/*ro, req, int, 最大值*/
"cardNo": {
/*ro, opt, object, 卡号*/
"@min": 1,
/*ro, req, int, 最小值*/
"@max": 32
/*ro, req, int, 最大值*/
}
}
},
"CardInfoDelCond": {
/*ro, opt, object, 删除条件*/
"EmployeeNoList": {
/*ro, opt, object, 人员ID列表*/
"maxSize": 56,
/*ro, req, int, 最大值*/
"employeeNo": {
/*ro, opt, object, 工号（人员ID）*/
"@min": 1,
/*ro, req, int, 最小值*/
"@max": 32
/*ro, req, int, 最大值*/
}
},
"CardNoList": {
/*ro, opt, object, 卡号列表*/
"maxSize": 56,
/*ro, req, int, 最大值*/
"cardNo": {
/*ro, opt, object, 卡号*/
"@min": 1,
/*ro, req, int, 最小值*/
"@max": 32
/*ro, req, int, 最大值*/
}
}
},
"cardNo": {
/*ro, req, object, 卡号*/
"@min": 1,
/*ro, req, int, 最小值*/
"@max": 32
/*ro, req, int, 最大值*/
},
"employeeNo": {
/*ro, req, object, 工号（人员ID）*/
"@min": 1,
/*ro, req, int, 最小值*/
"@max": 32
/*ro, req, int, 最大值*/
},
"cardType": {
/*ro, req, object, 卡类型*/
"@opt": "normalCard, patrolCard, hijackCard, superCard, dismissingCard, emergencyCard"
/*ro, req, string, 可选项*/
},
"leaderCard": {
/*ro, opt, object, 是否有首次认证功能*/
"@min": 1,
/*ro, req, int, 最小值*/

```

```
        "@max": 32
        /*ro, req, int, 最大值*/
    },
    "checkCardNo": "true,false",
    /*ro, opt, string, 设备是否进行卡重复添加校验, desc:false-不校验, true-校验（如果不配置该字段, 则设备默认进行卡重复校验。注：该字段用于加快下发卡信息, 一般不建议配置该字段）*/
    "checkEmployeeNo": "true,false",
    /*ro, opt, string, 设备是否进行工号（人员ID）存在校验, desc:false-不校验, true-校验（如果不配置该字段, 则设备默认进行工号（人员ID）存在校验）（该字段配置为false时, 设备不进行工号（人员ID）存在校验（可加快设备下发卡的速度）；该字段配置为true或不配置该字段时, 设备进行工号（人员ID）存在校验, 如不对处理速度有强烈要求, 建议使用该方式）*/
    "maxRecordNum": 10000,
    /*ro, opt, int, 支持的最大记录条数（卡数量）, desc:该字段对于平台集成十分重要, 虽定义成可选, 但建议设备实现*/
    "numberPerPerson": 50
    /*ro, opt, enum, 每人的卡数量限制, subType:int, [50#50张卡], desc:默认一人五张卡*/
}
}
```

69.37.10 删除卡信息

Request URL

PUT /ISAPI/AccessControl/CardInfo/Delete?format=json&security=<security>&iv=<iv>

查询参数

参数名称	参数类型	描述
security	string	没有security表示数据不加密；security=1,表示报文中的employeeNo、cardNo字段AES128加密；security=2,表示报文中的employeeNo、cardNo字段AES256加密；
iv	string	iv表示初始化向量，在security=1或者2时为必须参数;

请求报文

```
{
  "CardInfoDelCond": {
    /*req, object, 删除卡条件信息*/
    "EmployeeNoList": [
      /*opt, array, 人员ID列表, subType:object*/
      {
        "employeeNo": "test"
        /*opt, string, 工号（人员ID）*/
      }
    ],
    "CardNoList": [
      /*opt, array, 卡号列表, subType:object, desc:与人员ID列表互斥, 两者选一*/
      {
        "cardNo": "1234567890"
        /*opt, string, 卡号*/
      }
    ],
    "operateType": "byTerminal",
    /*opt, enum, 操作类型, subType:string, [byTerminal#按终端操作]*/
    "terminalNoList": [1]
    /*opt, array, type为byTerminal, subType:int*/
  }
}
```

响应报文

```
{
  "statusCode": 1,
  /*ro, opt, int, 状态码, desc:无法用1表示时（1表示成功且无特殊状态）必须返回*/
  "statusString": "OK",
  /*ro, opt, string, 状态描述, range:[1,64], desc:无法用OK表示时（OK表示成功且无特殊状态）必须返回*/
  "subStatusCode": "ok",
  /*ro, opt, string, 子状态码, range:[1,64], desc:无法用ok表示（ok表示成功且无特殊状态）必须返回*/
  "errorCode": 1,
  /*ro, opt, int, 错误码, desc:当statusCode不为1时, 与subStatusCode对应*/
  "errorMsg": "ok"
  /*ro, opt, string, 错误信息, desc:当statusCode不为1时, 必须返回,解释信息在协议约束中, 允许设备在后续的版本迭代中, 进行优化丰富提升（不限制死）*/
}
```

69.37.11 获取人员数量

Request URL

GET /ISAPI/AccessControl/UserInfo/Count?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "UserInfoCount": {
    /*ro, req, object, 用户数量*/
    "userNumber": 100,
    /*ro, req, int, 已存在的人员总数量*/
    "bindFaceUserNumber": 100,
    /*ro, opt, int, 已绑定人脸的人员数量*/
    "bindFingerprintUserNumber": 100,
    /*ro, opt, int, 已绑定指纹的人员数量*/
    "bindCardUserNumber": 100,
    /*ro, opt, int, 已绑定卡片的人员数量*/
    "bindIrisUserNumber": 100,
    /*ro, opt, int, 已绑定虹膜的人员数量*/
  }
}
```

69.37.12 修改人员信息

Request URL

PUT /ISAPI/AccessControl/UserInfo/Modify?format=json&security= <security> &iv= <iv>

查询参数

参数名称	参数类型	描述
security	string	没有security表示数据不加密； security=1,表示报文中的employeeNo、name、password、userName、loginPassword和dynamicCode字段AES128加密； security=2,表示报文中的employeeNo、name、password、userName、loginPassword和dynamicCode字段AES256加密；
iv	string	iv表示初始化向量，在security=1或者2时为必须参数;

请求报文

```
{
  "UserInfo": {
    /*req, object, 人员信息*/
    "employeeNo": "test",
    /*req, string, 工号（人员ID）*/
    "name": "test",
    /*opt, string, 姓名*/
    "userType": "normal",
    /*opt, enum, 人员类型, subType:string, [normal#普通人（主人）,visitor#来宾（访客）,blackList#黑名单人,maintenance#维护人员（包括保洁、维修人员等）,patient#病患], desc:维护人员需要在任意时间都能进入房间*/
    "closeDelayEnabled": true,
    /*opt, bool, 是否关门延迟*/
    "Valid": {
      /*opt, object, Valid*/
      "enable": true,
      /*opt, bool, 使能有效期, desc:enable不使能代表长期有效*/
      "beginTime": "2017-08-01T17:30:08+08:00",
      /*opt, datetime, 有效期起始时间, desc:timeType字段不存在或为local时, beginTime为设备本地时间, 如： 2017-08-01T17:30:08; timeType字段为UTC时, beginTime为UTC时间, 如： 2017-08-01T17:30:08+08:00*/
      "endTime": "2017-08-01T17:30:08+08:00",
      /*opt, datetime, 有效期结束时间（timeType字段不存在或为local时, desc:timeType字段不存在或为local时, beginTime为设备本地时间, 如： 2017-08-01T17:30:08; timeType字段为UTC时, beginTime为UTC时间, 如： 2017-08-01T17:30:08+08:00*/
      "timeType": "local"
    },
    /*opt, enum, 时间类型, subType:string, [local#设备本地时间,UTC#UTC时间]*/
  },
  "belongGroup": "1,3,5",
  /*opt, string, 所属群组*/
  "password": "123456",
  /*opt, string, 密码*/
  "doorRight": "1,3",
  /*opt, string, 门权限*/
  "RightPlan": [
    /*opt, array, 门权限计划（锁权限计划）, subType:object*/
    {
      "doorNo": 1,
    }
  ]
}
```

```

        /*opt, int, 门编号 (锁ID) */
        "planTemplateNo": "1,3,5"
        /*opt, string, 计划模板编号,
        desc: 同个门不同计划模板采用权限或的方式处理
        默认计划模板编号
        65535-7*24小时生效
        65534-周一到周五24小时生效
        65533-周六周日24小时生效*/
    }
},
"maxOpenDoorTime": 0,
/*opt, int, 最大认证次数, desc: 0为无次数限制*/
"roomNumber": 123,
/*opt, int, 房间号*/
"floorNumber": 1,
/*opt, int, 层号*/
"doubleLockRight": true,
/*opt, bool, 反锁开门权限*/
"localUIRight": true,
/*opt, bool, 是否具有设备本地UI访问权限*/
"userVerifyMode": "card",
/*opt, enum, 人员验证方式, subType:string, [cardAndPw#刷卡+密码, card#刷卡, cardOrPw#刷卡或密码, fp#指纹, fpAndPw#指纹+密码, fpOrCard#指纹或刷
卡, fpAndCard#指纹+刷卡, fpAndCardAndPw#指纹+刷卡+密码, faceOrFpOrCardOrPw#人脸或指纹或刷卡或密码, faceAndFp#人脸+指纹, faceAndPw#人脸+密码, faceAndCard#人脸+刷
卡, face#人脸, employeeNoAndPw#工号+密码, fpOrPw#指纹或密码, employeeNoAndFp#工号+指纹, employeeNoAndFpAndPw#工号+指纹+密码, faceAndFpAndCard#人脸+指纹+刷
卡, faceAndPwAndFp#人脸+密码+指纹, employeeNoAndFace#工号+人脸, faceOrFaceAndCard#人脸或人脸+刷卡, fpOrFace#指纹或人脸, cardOrFaceOrPw#刷卡或人脸或密码, iris#虹
膜, faceOrFpOrCardOrPwOrIris#人脸或指纹或刷卡或密码或虹膜, faceOrCardOrPwOrIris#人脸或刷卡或密码或虹膜, cardOrFace#刷卡或人脸, cardOrFaceOrFp#刷卡或人脸或指
纹, faceOrPw#人脸或密码, employeeNoAndFaceAndPw#工号+人脸+密码, cardOrFaceOrFaceAndCard#刷卡或人脸或人脸+刷卡, faceOrFpOrPw#人脸或指纹或密
码, cardOrFpOrFaceOrIris#刷卡或指纹或人脸或虹膜, fpOrFaceOrIrisOrPw#指纹或人脸或虹膜或密码, cardOrFpOrIrisOrPw#刷卡或指纹或虹膜或密码, cardOrIrisOrPw#刷卡或虹膜
或密码, cardAndIris#刷卡+虹膜, fpAndIris#指纹+虹膜, faceAndIris#人脸+虹膜, irisAndPw#虹膜+密码, cardAndIrisAndPw#刷卡+虹膜+密码, faceAndIrisAndPw#人脸+虹膜+密
码, cardAndFaceAndIris#刷卡+人脸+虹膜, Pw#密码],
desc: 人员验证方式的优先级高于读卡器验证方式。
若为空表示清空人员验证方式*/
"checkUser": true,
/*opt, bool, 设备是否进行人员重复添加校验, desc: false-不校验, true-校验 (如果不配置该字段, 则设备默认进行人员重复校验) (如果确认设备端不存在任何人
员信息, 可将其置为false, 则设备不进行重复校验, 这样会加快下发速度; 如果不确认, 则不建议配置该字段)*/
"gender": "male",
/*opt, enum, 人脸图片对应的人员性别, subType:string, [male#男, female#女, unknown#未知], desc: male-男, female-女, unknown-未知*/
"PersonInfoExtends": [
/*opt, array, 人员信息扩展, subType:object,
desc: 用于配置人员在设备UI界面的扩展信息显示
明眸当前仅支持一个value, 不支持id字段, 用于作为工号显示
智能主机按标准实现*/
{
    "id": 1,
    /*opt, int, 人员信息扩展序号, range:[1,32],
    desc: 与/ISAPI/AccessControl/personInfoExtendName?format=json中的id对应, 用于关联value的name信息
    id不存在时, 根据数组顺序默认从1开始*/
    "value": "test"
    /*opt, string, 人员信息扩展内容*/
}
],
"dynamicCode": "123456",
/*opt, string, 动态权限码*/
"callNumbers": ["1-1-1-401"],
/*opt, array, 呼叫号码列表, subType:string,
desc: roomNumber的扩展, 支持多房间号, 两者同时存在时, 优先解析callNumbers
国内规则:
1-1-1-层号+房间号。高三位表示层号, 范围1-999; 低两位表示房间号, 范围1-99。eg: 如层号为1, 房间号为1, 则为1-1-1-101; 如层号为222, 房间号为4, 则为1-1-
1-22240。
海外规则:
1-1-1-房间号。房间号的范围1-9999。eg: 如房间号为111, 则为1-1-1-111。*/
"floorNumbers": [1, 2],
/*opt, array, 层号列表, subType:int,
desc: floorNumber的扩展, 支持多层, 两者同时存在时, 优先解析floorNumbers。要求floorNumbers与callNumbers的数组元素个数一一对应。
国内eg:
callNumbers为["1-1-1-101", "1-1-1-302"], 则floorNumbers为[1, 3];
callNumbers为["1-1-1-101", "1-1-1-102"], 则floorNumbers为[1, 1];
海外eg:
callNumbers为["1-1-1-1", "1-1-1-99"], 由于callNumbers未包含层号, 因此floorNumbers仅需要元素个数与callNumbers个数一一对应即可, 若floorNumbers为[A, B],
则表示A层的1号房间和B层的99号房间*/
"groupId": 1,
/*opt, int, 本地考勤组织编号*/
"localAtndPlanTemplateId": 1,
/*opt, int, 本地考勤计划模板*/
"operateType": "byTerminal",
/*opt, string, 操作类型, desc: byTerminal: 按终端操作*/
"terminalNoList": [1],
/*opt, array, 终端ID列表, subType:int, desc: type为byTerminal, byTerminalOrg时必须填, 终端ID列表 (目前仅支持单个终端)*/
"groupName": "test",
/*opt, string, 组织架构, range:[1,64]*/
"age": 0,
/*opt, int, 年龄, range:[0,120]*/
"PatientInfos": {
/*opt, object, 患者信息, desc: userType==patient时必须有效*/
    "deviceId": "test",
    /*opt, string, 设备编号 (对应编号配置中编号)*/
    "admissionTime": "1970-01-01T00:00:00+08:00",
    /*opt, datetime, 入院时间*/
    "chargeNurse": "test",
    /*opt, string, 责任护士, range:[0,32]*/
    "chargeDoctor": "test",
    /*opt, string, 责任医师, range:[0,32]*/
    "nursingLevel": "tertiary",

```

```
/*opt, enum, 护理等级, subType:string, [unknown#未知,tertiary#三级护理,secondary#二级护理,primary#一级护理,special#特级护理]*/
"doctorsAdvice": "test",
/*opt, string, 医嘱注意事项, range:[0,128]*/
"allergicHistory": "test"
/*opt, string, 过敏情况, range:[0,128]*/
},
"ESDType": "handAndFoot",
/*opt, enum, ESD检测类型, subType:string, [handAndFoot#手脚都检测,no#不检测,hand#检测手,foot#检测脚], desc:默认为no*/
"userLevel": "Employee",
/*opt, enum, 用户级别, subType:string, [Employee#普通员工,DepartmentManager#部门主管]*/
"userName": "test",
/*opt, string, 登录账号, range:[1,32], desc:数字、小写字母、大写字母、特殊字符（敏感信息加密）登录账号与登录密码字段必须同时下发*/
"loginPassword": "test",
/*opt, string, 登录密码, range:[8,16], desc:数字、小写字母、大写字母、特殊字符两种及以上组合（敏感信息加密）登录账号与登录密码字段必须同时下发*/
"deviceID": "test"
/*opt, string, 设备编号（对应编号配置中编号*/
}
}
```

响应报文

```
{
  "statusCode": 1,
  /*ro, opt, int, 状态码, desc:无法用1表示时（1表示成功且无特殊状态）必须返回*/
  "statusString": "OK",
  /*ro, opt, string, 状态描述, range:[1,64], desc:无法用OK表示时（OK表示成功且无特殊状态）必须返回*/
  "subStatusCode": "ok",
  /*ro, opt, string, 子状态码, range:[1,64], desc:无法用ok表示（ok表示成功且无特殊状态）必须返回*/
  "errorCode": 1,
  /*ro, opt, int, 错误码, desc:当statusCode不为1时, 与subStatusCode对应*/
  "errorMsg": "ok"
  /*ro, opt, string, 错误信息, desc:当statusCode不为1时, 必须返回,解释信息在协议约束中, 允许设备在后续的版本迭代中, 进行优化丰富提升（不限制死）*/
}
```

69.37.13 获取单一任务状态（人员图片异步下发）

Request URL

GET /ISAPI/AccessControl/UserPic/asyncImportDatasTasks/<taskID>/status?format=json

查询参数

参数名称	参数类型	描述
taskID	string	

请求报文

无

响应报文

```
{
  "AsyncImportDatasTask": {
    /*ro, req, object, 人员图片异步下发任务*/
    "taskID": "test",
    /*ro, req, string, 下发任务ID, range:[1,64]*/
    "URL": "test",
    /*ro, opt, string, 数据文件的URL, range:[1,256]*/
    "status": 1,
    /*ro, req, enum, 任务执行状态, subType:int, [0#未执行,1#执行中,2#执行完成,3#未找到该任务]*/
    "totalNum": 1,
    /*ro, opt, int, 导入人员图片总数, desc:该节点依赖于status为2-执行完成*/
    "successNum": 1,
    /*ro, opt, int, 人员图片存储成功数目, desc:该节点依赖于status为2-执行完成*/
    "failedNum": 1
    /*ro, opt, int, 人员图片存储失败数目, desc:该节点依赖于status为2-执行完成*/
  }
}
```

69.37.14 配置人员信息及权限删除

Request URL

PUT /ISAPI/AccessControl/UserInfoDetail/Delete?format=json

查询参数

无

请求报文

```
{
  "UserInfoDetail": {
    /*opt, object, 用户信息*/
    "mode": "all",
    /*req, enum, 删除模式, subType:string, [all#删除所有,byEmployeeNo#按工号]*/
    "EmployeeNoList": [
      /*opt, array, 人员ID列表, subType:object*/
      {
        "employeeNo": "test"
        /*opt, string, 工号*/
      }
    ],
    "operateType": "byTerminal",
    /*opt, enum, 操作类型, subType:string, [byTerminal#按终端操作,byOrg#按组织操作,byTerminalOrg#按终端组织操作]*/
    "terminalNoList": [1, 2, 3, 4],
    /*opt, array, 终端列表, subType:int, dep:and,{$.UserInfoDetail.operateType,eq,byTerminal}*/
    "orgNoList": [1, 2, 3, 4]
    /*opt, array, 组织列表, subType:int, dep:or,{$.UserInfoDetail.operateType,eq,byOrg},{$.UserInfoDetail.operateType,eq,byTerminalOrg}*/
  }
}
```

响应报文

```
{
  "statusCode": 1,
  /*ro, opt, int, 状态码, desc:无法用1表示时（1表示成功且无特殊状态）必须返回*/
  "statusString": "OK",
  /*ro, opt, string, 状态描述, range:[1,64], desc:无法用OK表示时（OK表示成功且无特殊状态）必须返回*/
  "subStatusCode": "ok",
  /*ro, opt, string, 子状态码, range:[1,64], desc:无法用ok表示（ok表示成功且无特殊状态）必须返回*/
  "errorCode": 1,
  /*ro, opt, int, 错误码, desc:当statusCode不为1时，与subStatusCode对应*/
  "errorMsg": "ok"
  /*ro, opt, string, 错误信息, desc:当statusCode不为1时，必须返回,解释信息在协议约束中，允许设备在后续的版本迭代中，进行优化丰富提升（不限制死）*/
}
```

69.37.15 添加人员信息

Request URL

POST /ISAPI/AccessControl/UserInfo/Record?format=json&security= <security> &iv= <iv>

查询参数

参数名称	参数类型	描述
security	string	没有security表示数据不加密； security=1,表示报文中的employeeNo、name、password、userName、loginPassword和dynamicCode字段AES128加密； security=2,表示报文中的employeeNo、name、password、userName、loginPassword和dynamicCode字段AES256加密；
iv	string	iv表示初始化向量，在security=1或者2时为必须参数;

请求报文

```
{
  "UserInfo": {
    /*req, object, 人员信息*/
    "employeeNo": "test",
    /*req, string, 工号（人员ID）*/
    "name": "test",
    /*opt, string, 姓名*/
    "userType": "normal",
    /*req, enum, 人员类型, subType:string, [normal#普通人（主人）,visitor#来宾（访客）,blackList#黑名单人,maintenance#维护人员（包括保洁、维修人员等）,patient#病患], desc:维护人员需要在任意时间都能进入房间*/
    "closeDelayEnabled": true,
    /*opt, bool, 是否关门延迟*/
    "Valid": {
      /*req, object, Valid*/
      "enable": true,
      /*req, bool, 使能有效期, desc:enable不使能代表长期有效*/
      "beginTime": "2017-08-01T17:30:08+08:00",
      /*req, datetime, 有效期起始时间, desc:timeType字段不存在或为local时, beginTime为设备本地时间, 如: 2017-08-01T17:30:08; timeType字段为UTC时, beginTime为UTC时间, 如: 2017-08-01T17:30:08+08:00*/
      "endTime": "2017-08-01T17:30:08+08:00",
      /*req, datetime, 有效期结束时间（timeType字段不存在或为local时, desc:timeType字段不存在或为local时, beginTime为设备本地时间, 如: 2017-08-01T17:30:08; timeType字段为UTC时, beginTime为UTC时间, 如: 2017-08-01T17:30:08+08:00*/
      "timeType": "local"
    }
  }
}
```

```

        type: "enum",
        /*opt, enum, 时间类型, subType:string, [local#设备本地时间,UTC#UTC时间]*/
    },
    "belongGroup": "1,3,5",
    /*opt, string, 所属群组*/
    "password": "123456",
    /*opt, string, 凭证密码*/
    "doorRight": "1,3",
    /*opt, string, 门权限*/
    "RightPlan": [
    /*opt, array, 门权限计划（锁权限计划），subType:object*/
    {
        "doorNo": 1,
        /*opt, int, 门编号（锁ID）*/
        "planTemplateNo": "1,3,5",
        /*opt, string, 计划模板编号,
        desc:同个门不同计划模板采用权限或的方式处理
        默认计划模板编号
        65535-7*24小时生效
        65534-周一到周五24小时生效
        65533-周六周日24小时生效*/
    }
    ],
    "maxOpenDoorTime": 0,
    /*opt, int, 最大认证次数, desc:0为无次数限制*/
    "roomNumber": 123,
    /*opt, int, 房间号*/
    "floorNumber": 1,
    /*opt, int, 层号*/
    "doubleLockRight": true,
    /*opt, bool, 反锁开门权限*/
    "localUIRight": true,
    /*opt, bool, 是否具有设备本地UI访问权限*/
    "userVerifyMode": "card",
    /*opt, enum, 人员验证方式, subType:string, [cardAndPw#刷卡+密码, card#刷卡, cardOrPw#刷卡或密码, fp#指纹, fpAndPw#指纹+密码, fpOrCard#指纹或刷
    卡, fpAndCard#指纹+刷卡, fpAndCardAndPw#指纹+刷卡+密码, faceOrFpOrCardOrPw#人脸或指纹或刷卡或密码, faceAndFp#人脸+指纹, faceAndPw#人脸+密码, faceAndCard#人脸+刷
    卡, face#人脸, employeeNoAndPw#工号+密码, fpOrPw#指纹或密码, employeeNoAndFp#工号+指纹, employeeNoAndFpAndPw#工号+指纹+密码, faceAndFpAndCard#人脸+指纹+刷
    卡, faceAndPwAndFp#人脸+密码+指纹, employeeNoAndFace#工号+人脸, faceOrfaceAndCard#人脸或人脸+刷卡, fpOrface#指纹或人脸, cardOrfaceOrPw#刷卡或人脸或密码, iris#虹
    膜, faceOrFpOrCardOrPwOrIris#人脸或指纹或刷卡或密码或虹膜, faceOrCardOrPwOrIris#人脸或刷卡或密码或虹膜, cardOrFace#刷卡或人脸, cardOrFaceOrFp#刷卡或人脸或指
    纹, faceOrPw#人脸或密码, employeeNoAndFaceAndPw#工号+人脸+密码, cardOrFaceOrFaceAndCard#刷卡或人脸或人脸+刷卡, faceOrFpOrPw#人脸或指纹或密
    码, cardOrFpOrFaceOrIris#刷卡或指纹或人脸或虹膜, fpOrFaceOrIrisOrPw#指纹或人脸或虹膜或密码, cardOrFpOrIrisOrPw#刷卡或指纹或虹膜或密码, cardOrIrisOrPw#刷卡或虹膜
    或密码, cardAndIris#刷卡+虹膜, fpAndIris#指纹+虹膜, faceAndIris#人脸+虹膜, irisAndPw#虹膜+密码, cardAndIrisAndPw#刷卡+虹膜+密码, faceAndIrisAndPw#人脸+虹膜+密
    码, cardAndFaceAndIris#刷卡+人脸+虹膜, Pw#密码],
    desc:人员验证方式的优先级高于读卡器验证方式
    若为空表示人员验证方式无效*/
    "checkUser": true,
    /*opt, bool, 设备是否进行人员重复添加校验, desc:false-不校验, true-校验（如果不配置该字段, 则设备默认进行人员重复校验）（如果确认设备端不存在任何人
    员信息, 可将其置为false, 则设备不进行重复校验, 这样会加快下发速度; 如果不确认, 则不建议配置该字段）*/
    "gender": "male",
    /*opt, enum, 人脸图片对应的人员性别, subType:string, [male#男, female#女, unknown#未知], desc:male-男, female-女, unknown-未知*/
    "PersonInfoExtends": [
    /*opt, array, 人员信息扩展, subType:object,
    desc:用于配置人员在设备UI界面的扩展信息显示
    明眸当前仅支持一个value, 不支持id字段, 用于作为工号显示
    智能主机按标准实现*/
    {
        "id": 1,
        /*opt, int, 人员信息扩展序号, range:[1,3],
        desc:与/ISAPI/AccessControl/personInfoExtendName?format=json中的id对应, 用于关联value的name信息
        id不存在时, 根据数组顺序默认从1开始*/
        "value": "test",
        /*opt, string, 人员信息扩展内容*/
    }
    ],
    "operateType": "byTerminal",
    /*opt, string, 操作类型, desc:byTerminal:按终端操作*/
    "terminalNoList": [1],
    /*opt, array, 终端ID列表, subType:int, desc:type为byTerminal, byTerminalOr时必须必填, 终端ID列表（目前仅支持单个终端）*/
    "dynamicCode": "123456",
    /*opt, string, 动态权限码*/
    "callNumbers": ["1-1-1-401"],
    /*opt, array, 呼叫号码列表, subType:string,
    desc:roomNumber的扩展, 支持多房间号, 两者同时存在时, 优先解析callNumbers
    国内规则:
    1-1-1-层号+房间号。高三位表示层号, 范围1~999; 低两位表示房间号, 范围1~99。eg: 如层号为1, 房间号为1, 则为1-1-1-101; 如层号为222, 房间号为4, 则为1-1-
    1-22204。
    海外规则:
    1-1-1-房间号。房间号的范围1~9999。eg: 如房间号为111, 则为1-1-1-111。*/
    "floorNumbers": [1, 2],
    /*opt, array, 层号列表, subType:int,
    desc:floorNumber的扩展, 支持多层, 两者同时存在时, 优先解析floorNumbers。要求floorNumbers与callNumbers的数组元素个数一一对应。
    国内eg:
    callNumbers为["1-1-1-101", "1-1-1-302"], 则floorNumbers为[1, 3];
    callNumbers为["1-1-1-101", "1-1-1-102"], 则floorNumbers为[1, 1];
    海外eg:
    callNumbers为["1-1-1-1", "1-1-1-99"], 由于callNumbers未包含层号, 因此floorNumbers仅需要元素个数与callNumbers个数一一对应即可, 若floorNumbers为[A, B],
    则表示A层的1号房间和B层的99号房间*/
    "groupId": 1,
    /*opt, int, 本地考勤组织编号*/
    "localAtndPlanTemplateId": 1,
    /*opt, int, 本地考勤计划模板*/
    "groupName": "test",
    /*opt, string, 组织架构, range:[1,64]*/
    "age": 0,

```

```
/*opt, int, 年龄, range:[0,120]*/
"PatientInfos": {
/*opt, object, 患者信息, desc:userType==patient时必须有效*/
  "deviceID": "test",
  /*opt, string, 设备编号（对应编号配置中编号*/
  "admissionTime": "1970-01-01T00:00:00+08:00",
  /*opt, datetime, 入院时间*/
  "chargeNurse": "test",
  /*opt, string, 责任护士, range:[0,32]*/
  "chargeDoctor": "test",
  /*opt, string, 责任医师, range:[0,32]*/
  "nursingLevel": "tertiary",
  /*opt, enum, 护理等级, subType:string, [unknown#未知,tertiary#三级护理,secondary#二级护理,primary#一级护理,special#特级护理]*/
  "doctorsAdvice": "test",
  /*opt, string, 医嘱注意事项, range:[0,128]*/
  "allergicHistory": "test"
  /*opt, string, 过敏情况, range:[0,128]*/
},
"ESDType": "handAndFoot",
/*opt, enum, ESD检测类型, subType:string, [handAndFoot#手脚都检测,no#不检测,hand#检测手,foot#检测脚], desc:默认为no*/
"userLevel": "Employee",
/*opt, enum, 用户级别, subType:string, [Employee#普通员工,DepartmentManager#部门主管]*/
"userName": "test",
/*opt, string, 登录账号, range:[1,32], desc:数字、小写字母、大写字母、特殊字符（敏感信息加密）登录账号与登录密码字段必须同时下发*/
"loginPassword": "test",
/*opt, string, 登录密码, range:[8,16], desc:数字、小写字母、大写字母、特殊字符两种及以上组合（敏感信息加密）登录账号与登录密码字段必须同时下发*/
"deviceID": "test"
/*opt, string, 设备编号（对应编号配置中编号*/
}
}
```

响应报文

```
{
  "statusCode": 1,
  /*ro, opt, int, 状态码, desc:无法用1表示时（1表示成功且无特殊状态）必须返回*/
  "statusString": "OK",
  /*ro, opt, string, 状态描述, range:[1,64], desc:无法用OK表示时（OK表示成功且无特殊状态）必须返回*/
  "subStatusCode": "ok",
  /*ro, opt, string, 子状态码, range:[1,64], desc:无法用ok表示（ok表示成功且无特殊状态）必须返回*/
  "errorCode": 1,
  /*ro, opt, int, 错误码, desc:当statusCode不为1时, 与subStatusCode对应*/
  "errorMsg": "ok"
  /*ro, opt, string, 错误信息, desc:当statusCode不为1时, 必须返回,解释信息在协议约束中, 允许设备在后续的版本迭代中, 进行优化丰富提升（不限制死）*/
}
```

69.37.16 设置人员信息

Request URL

PUT /ISAPI/AccessControl/UserInfo/SetUp?format=json&security=<security>&iv=<iv>

查询参数

参数名称	参数类型	描述
security	string	没有security表示数据不加密; security=1,表示报文中的employeeNo、name、password、userName、loginPassword和dynamicCode字段AES128加密; security=2,表示报文中的employeeNo、name、password、userName、loginPassword和dynamicCode字段AES256加密;
iv	string	iv表示初始化向量, 在security=1或者2时为必须参数;

请求报文

```
{
  "UserInfo": {
    /*req, object, 人员信息*/
    "employeeNo": "test",
    /*req, string, 工号（人员ID）*/
    "deleteUser": true,
    /*opt, bool, 是否删除该人员*/
    "name": "test",
    /*opt, string, 姓名*/
    "userType": "normal",
    /*req, enum, 人员类型, subType:string, [normal#普通人（主人）,visitor#来宾（访客）,blackList#黑名单人,maintenance#维护人员（包括保洁、维修人员等）,patient#病患], dep:and,{$.UserInfo.deleteUser,eq,false},{$.UserInfo.deleteUser,eq,不存在}, desc:维护人员需要在任意时间都能进入房间*/
    "closeDelayEnabled": true,
    /*opt, bool, 是否关门延迟, desc:true-是, false-否*/
    "Valid": {
      /*req, object, 有效期参数, dep:and,{$.UserInfo.deleteUser,eq,false},{$.UserInfo.deleteUser,eq,不存在}, desc:（enable不使能代表长期有效）（有效时间跨
```

度为1970年1月1日0点0分0秒~2037年12月31日23点59分59秒）*/

```
"enable": true,
/*req, bool, 使能有效期, desc:enable不使能代表长期有效*/
"beginTime": "test",
/*req, string, 有效期起始时间, desc:timeType字段不存在或为local时, beginTime为设备本地时间, 如: 2017-08-01T17:30:08; timeType字段为UTC
时, beginTime为UTC时间, 如: 2017-08-01T17:30:08+08:00*/
"endTime": "test",
/*req, string, 有效期结束时间, desc:timeType字段不存在或为local时, endTime为设备本地时间, 如: 2017-08-01T17:30:08; timeType字段为UTC时, endTime
为UTC时间, 如: 2017-08-01T17:30:08+08:00*/
"timeType": "local"
/*opt, enum, 时间类型, subType:string, [local#设备本地时间,UTC#UTC时间]*/
},
"belongGroup": "1,3,5",
/*opt, string, 所属群组*/
"password": "123456",
/*opt, string, 密码*/
"doorRight": "1,3",
/*opt, string, 门权限, desc:代表对门1、门3有权限*/
"RightPlan": [
/*opt, array, 门权限计划, subType:object*/
{
    "doorNo": 1,
    /*opt, int, 门编号（锁ID）*/
    "planTemplateNo": "1,3,5"
    /*opt, string, 计划模板编号,
    desc:同个门不同计划模板采用权限或的方式处理
    默认计划模板编号
    65535-7*24小时生效
    65534-周一到周五24小时生效
    65533-周六周日24小时生效*/
}
],
"maxOpenDoorTime": 0,
/*opt, int, 最大认证次数*/
"roomNumber": 123,
/*opt, int, 房间号*/
"floorNumber": 1,
/*opt, int, 层号*/
"doubleLockRight": true,
/*opt, bool, 反锁开门权限, desc:true-有权限, false-无权限*/
"localUIRight": true,
/*opt, bool, 是否具有设备本地UI访问权限, desc:true-有权限, false-无权限*/
"userVerifyMode": "card",
/*opt, enum, 人员验证方式, subType:string, [cardAndPw#刷卡+密码,card#刷卡,cardOrPw#刷卡或密码,fp#指纹,fpAndPw#指纹+密码,fpOrCard#指纹或刷
卡,fpAndCard#指纹+刷卡,fpAndCardAndPw#指纹+刷卡+密码,faceOrFpOrCardOrPw#人脸或指纹或刷卡或密码,faceAndFp#人脸+指纹,faceAndPw#人脸+密码,faceAndCard#人脸+刷
卡,face#人脸,employeeNoAndPw#工号+密码,fpOrPw#指纹或密码,employeeNoAndFp#工号+指纹,employeeNoAndFpAndPw#工号+指纹+密码,faceAndFpAndCard#人脸+指纹+刷
卡,faceAndPwAndFp#人脸+密码+指纹,employeeNoAndFace#工号+人脸,faceOrFaceAndCard#人脸或人脸+刷卡,fpOrFace#指纹或人脸,cardOrFaceOrPw#刷卡或人脸或密码,iris#虹
膜,faceOrFpOrCardOrPwOrIris#人脸或指纹或刷卡或密码或虹膜,faceOrCardOrPwOrIris#人脸或刷卡或密码或虹膜,cardOrFace#刷卡或人脸,cardOrFaceOrFp#刷卡或人脸或指
纹,faceOrPw#人脸或密码,employeeNoAndFaceAndPw#工号+人脸+密码,cardOrFaceOrFaceAndCard#刷卡或人脸或人脸+刷卡,faceOrFpOrPw#人脸或指纹或密
码,cardOrFpOrFaceOrIris#刷卡或指纹或人脸或虹膜,fpOrFaceOrIrisOrPw#指纹或人脸或虹膜或密码,cardOrFpOrIrisOrPw#刷卡或指纹或虹膜或密码,cardOrIrisOrPw#刷卡或虹膜
或密码,cardAndIris#刷卡+虹膜,fpAndIris#指纹+虹膜,faceAndIris#人脸+虹膜,irisAndPw#虹膜+密码,cardAndIrisAndPw#刷卡+虹膜+密码,faceAndIrisAndPw#人脸+虹膜+密
码,cardAndFaceAndIris#刷卡+人脸+虹膜,Pw#密码],
desc:人员验证方式的优先级高于读卡器验证方式
若为空表示人员验证方式无效*/
"checkUser": true,
/*opt, bool, 设备是否进行人员重复添加校验, desc:false-不校验, true-校验（如果不配置该字段, 则设备默认进行人员重复校验）（如果确认设备端不存在任何人
员信息, 可将其置为false, 则设备不进行重复校验, 这样会加快下发速度; 如果不确认, 则不建议配置该字段）*/
"gender": "male",
/*opt, enum, 人脸图片对应的人员性别, subType:string, [male#男,female#女,unknown#未知]*/
"PersonInfoExtends": [
/*opt, array, 人员信息扩展, subType:object,
desc:用于配置人员在设备UI界面的扩展信息显示
明眸当前仅支持一个value, 不支持id字段, 用于作为工号显示
智能主机按标准实现*/
{
    "id": 1,
    /*opt, int, 人员信息扩展序号, range:[1,32],
    desc:与/ISAPI/AccessControl/personInfoExtendName?format=json中的id对应, 用于关联value的name信息
    id不存在时, 根据数组顺序默认从1开始*/
    "value": "test"
    /*opt, string, 人员信息扩展内容*/
}
],
"dynamicCode": "123456",
/*opt, string, 动态权限码*/
"callNumbers": ["1-1-1-401"],
/*opt, array, 呼叫号码列表, subType:string,
desc:roomNumber的扩展, 支持多房间号, 两者同时存在时, 优先解析callNumbers
国内规则:
1-1-1-层号+房间号。高三位表示层号, 范围1~999; 低两位表示房间号, 范围1~99。eg: 如层号为1, 房间号为1, 则为1-1-1-101; 如层号为222, 房间号为4, 则
为1-1-1-22204。
海外规则:
1-1-1-房间号。房间号的范围1-9999。eg: 如房间号为111, 则为1-1-1-111。*/
"floorNumbers": [1, 2],
/*opt, array, 层号列表, subType:int,
desc:floorNumber的扩展, 支持多层, 两者同时存在时, 优先解析floorNumbers。要求floorNumbers与callNumbers的数组元素个数一一对应。
国内eg:
callNumbers为["1-1-1-101","1-1-1-302"], 则floorNumbers为[1,3];
callNumbers为["1-1-1-101","1-1-1-102"], 则floorNumbers为[1,1];
海外eg:
callNumbers为["1-1-1-1","1-1-1-99"], 由于callNumbers未包含层号, 因此floorNumbers仅需要元素个数与callNumbers个数一一对应即可, 若floorNumbers为[A,B],
则表示A层的1号房间和B层的99号房间*/
"groupId": 1,
/*opt, int, 本地考勤组织编号*/
```

```

    "localAtndPlanTemplateId": 1,
    /*opt, int, 本地考勤计划模板*/
    "groupName": "test",
    /*opt, string, 组织架构, range:[1,64]*/
    "ESDType": "handAndFoot",
    /*opt, enum, ESD检测类型 , subType:string, [handAndFoot#手脚都检测,no#不检测,hand#检测手,foot#检测脚], desc:默认为no*/
    "userLevel": "Employee",
    /*opt, enum, 用户级别, subType:string, [Employee#普通员工,DepartmentManager#部门主管]*/
    "userName": "test",
    /*opt, string, 登录账号, range:[1,32], desc:数字、小写字母、大写字母、特殊字符（敏感信息加密）登录账号与登录密码字段必须同时下发*/
    "loginPassword": "test"
    /*opt, string, 登录密码, range:[8,16], desc:数字、小写字母、大写字母、特殊字符两种及以上组合（敏感信息加密）登录账号与登录密码字段必须同时下发*/
  }
}

```

响应报文

```

{
  "statusCode": 1,
  /*ro, opt, int, 状态码, desc:无法用1表示时（1表示成功且无特殊状态）必须返回*/
  "statusString": "OK",
  /*ro, opt, string, 状态描述, range:[1,64], desc:无法用OK表示时（OK表示成功且无特殊状态）必须返回*/
  "subStatusCode": "ok",
  /*ro, opt, string, 子状态码, range:[1,64], desc:无法用ok表示（ok表示成功且无特殊状态）必须返回*/
  "errorCode": 1,
  /*ro, opt, int, 错误码, desc:当statusCode不为1时, 与subStatusCode对应*/
  "errorMsg": "ok"
  /*ro, opt, string, 错误信息, desc:当statusCode不为1时, 必须返回,解释信息在协议约束中, 允许设备在后续的版本迭代中, 进行优化丰富提升（不限制死）*/
}

```

69.37.17 修改卡信息

Request URL

PUT /ISAPI/AccessControl/CardInfo/Modify?format=json&security= <security> &iv=<iv>

查询参数

参数名称	参数类型	描述
security	string	没有security表示数据不加密； security=1,表示报文中的employeeNo、 cardNo字段AES128加密； security=2,表示报文中的employeeNo、 cardNo字段AES256加密；
iv	string	iv表示初始化向量，在security=1或者2时为必须参数;

请求报文

```

{
  "CardInfo": {
    /*req, object, 卡信息*/
    "employeeNo": "test",
    /*req, string, 工号（人员ID）*/
    "cardNo": "1234567890",
    /*req, string, 卡号*/
    "cardType": "normalCard",
    /*opt, enum, 卡类型, subType:string, [normalCard#普通卡,patrolCard#巡更卡,hijackCard#胁迫卡,superCard#超级卡,dismissingCard#解除卡,emergencyCard#应急管理卡（用于授权临时卡权限）*/
    "leaderCard": "1,3,5",
    /*opt, string, 是否有首次认证功能, desc:（表示该卡对于门1、门3、门5有首次认证功能）*/
    "operateType": "byTerminal",
    /*opt, string, 操作类型*/
    "terminalNoList": [1]
    /*opt, array, 终端ID列表, subType:int, desc:type为byTerminal,byTerminal0ng时必须填,终端ID列表（目前仅支持单个终端*/
  }
}

```

响应报文

```
{
  "statusCode": 1,
  /*ro, opt, int, 状态码, desc:无法用1表示时（1表示成功且无特殊状态）必须返回*/
  "statusString": "OK",
  /*ro, opt, string, 状态描述, range:[1,64], desc:无法用OK表示时（OK表示成功且无特殊状态）必须返回*/
  "subStatusCode": "ok",
  /*ro, opt, string, 子状态码, range:[1,64], desc:无法用ok表示（ok表示成功且无特殊状态）必须返回*/
  "errorCode": 1,
  /*ro, opt, int, 错误码, desc:当statusCode不为1时，与subStatusCode对应*/
  "errorMsg": "ok"
  /*ro, opt, string, 错误信息, desc:当statusCode不为1时，必须返回,解释信息在协议约束中，允许设备在后续的版本迭代中，进行优化丰富提升（不限制死）*/
}
```

69.37.18 设置卡信息

Request URL

PUT /ISAPI/AccessControl/CardInfo/SetUp?format=json&security=<security>&iv=<iv>

查询参数

参数名称	参数类型	描述
security	string	没有security表示数据不加密；security=1,表示报文中的employeeNo、cardNo字段AES128加密；security=2,表示报文中的employeeNo、cardNo字段AES256加密；
iv	string	--

请求报文

```
{
  "CardInfo": {
    /*opt, object, 卡信息*/
    "employeeNo": "test",
    /*req, string, 工号（人员ID）*/
    "cardNo": "1234567890",
    /*req, string, 卡号, desc:（新增和修改卡号时，该节点必填；如果删除单个人员卡号，该节点必填；如果删除所有人员卡号，该节点可选）*/
    "deleteCard": true,
    /*opt, bool, 是否删除该卡, desc:true-是（只有删除该卡时，才填写该字段；新增或修改卡时，不填写该字段）*/
    "cardType": "normalCard",
    /*req, enum, 卡类型, subType:string, [normalCard#普通卡,patrolCard#巡更卡,hijackCard#胁迫卡,superCard#超级卡,dismissingCard#解除卡,emergencyCard#应急管理卡]*/
    "leaderCard": "1,3,5",
    /*opt, string, 是否有首次认证功能, desc:（表示该卡对于门1、门3、门5有首次认证功能）*/
    "checkCardNo": true,
    /*opt, bool, 设备是否进行卡重复添加校验, desc:false-不校验, true-校验（如果不配置该字段，则设备默认进行卡重复校验）（如果确认设备端不存在任何卡信息，可将其置为false，则设备不进行重复校验，这样会加快下发速度；如果不确认，则不建议配置该字段）*/
    "checkEmployeeNo": true,
    /*opt, bool, 设备是否进行工号（人员ID）存在校验, desc:false-不校验, true-校验（如果不配置该字段，则设备默认进行工号（人员ID）存在校验）（该字段配置为false时，设备不进行工号（人员ID）存在校验（可加快设备下发卡的速度）；该字段配置为true或不配置该字段时，设备进行工号（人员ID）存在校验，如不对处理速度有强烈要求，建议使用该方式）*/
  }
}
```

响应报文

```
{
  "statusCode": 1,
  /*ro, opt, int, 状态码, desc:无法用1表示时（1表示成功且无特殊状态）必须返回*/
  "statusString": "OK",
  /*ro, opt, string, 状态描述, range:[1,64], desc:无法用OK表示时（OK表示成功且无特殊状态）必须返回*/
  "subStatusCode": "ok",
  /*ro, opt, string, 子状态码, range:[1,64], desc:无法用ok表示（ok表示成功且无特殊状态）必须返回*/
  "errorCode": 1,
  /*ro, opt, int, 错误码, desc:当statusCode不为1时，与subStatusCode对应*/
  "errorMsg": "ok"
  /*ro, opt, string, 错误信息, desc:当statusCode不为1时，必须返回,解释信息在协议约束中，允许设备在后续的版本迭代中，进行优化丰富提升（不限制死）*/
}
```

69.37.19 修改虹膜信息

Request URL

PUT /ISAPI/AccessControl/IrisInfo/modify?format=json&security=<security>&iv=<iv>

查询参数

参数名称	参数类型	描述
security	string	没有security表示数据不加密； security=1，表示报文中的employeeNo,irisModelData字段AES128加密； security=2，表示报文中的employeeNo,irisModelData字段AES256加密
iv	string	iv表示初始化向量，在security=1或者2时为必须参数

请求报文

```
{
  "IrisInfo": {
    /*req, object, 虹膜信息, desc:虹膜的单双目下发成功与否，与设备的单双目认证模式无关*/
    "employeeNo": "test",
    /*req, string, 工号*/
    "irisPicDataType": "url",
    /*req, enum, 虹膜图片数据传输方式, subType:string, [url#存储服务器url,binary#二进制数据],
    desc:url时，报文采用非表单方式
    binary时，报文采用表单方式*/
    "IrisData": [
      /*req, array, 虹膜数据, subType:object, desc:模板数据和图片数据必须有其一，否则失败；下发两个虹膜数据时，支持一个成功一个失败的情况，此时响应报文总状态
      为失败，子状态中返回每个虹膜数据的成功失败情况*/
      {
        "id": 1,
        /*req, int, 虹膜数据序号, range:[1,2], desc:与响应结果中AttachInfo节点内的id一一对应,AttachInfo节点内的id为string类型是为了兼容其他业务存在
        字符串的情况,本协议中使用"1"表示即可;id仅支持1和2,否则失败*/
        "irisModelData": "test",
        /*opt, string, 虹膜模板数据,
        desc:必须base处理，防止模型数据出现'\0'字符。
        当下发模板+图片时，以模板数据为准，且不查重，设备不判断模板有效性。
        加密时进行两次base64处理后，再做aes加密*/
        "irisPicData": "test"
        /*opt, string, 虹膜图片数据, desc:当irisPicDataType为url时，此处为url字符串；当irisPicDataType为binary时，此处无用途*/
      }
    ]
  }
}
```

参数名称	参数值	参数类型 (Content-Type)	Content-ID	文件名称 (filename)	描述
IrisInfo	[报文内容]	application/json	--	--	--
设备未实现	[图片二进制数据]	application/x-bmp		设备未实现	name和filename设备不做校验，不区分左右眼虹膜图片顺序，兼容image/bmp
设备未实现	[图片二进制数据]	application/x-bmp		设备未实现	name和filename设备不做校验，不区分左右眼虹膜图片顺序，兼容image/bmp

说明： 此协议通过表格格式传输， 表单框架详情说明见章节4.5.1.4， 如下实例：

```
--<frontier>
Content-Disposition: form-data; name=参数名称;filename=文件名称
Content-Type: 参数类型
Content-Length: ****
Content-ID: 内容ID
参数值
```

- 参数名称：表单元Headers中Content-Disposition的name属性，表示此表单元的名称
- 参数类型（Content-Type）：表单元Headers中Content-Type属性
- 文件名称（filename）：表单元Headers中Content-Disposition的filename属性，仅在本表单元传输数据为文件类型存在，表示表单元Body的文件名
- 参数值：表单元的body内容

响应报文

```

{
  "statusCode": 1,
  /*ro, opt, int, 状态码, desc:无法用1表示时（1表示成功且无特殊状态）必须返回*/
  "statusString": "OK",
  /*ro, opt, string, 状态描述, range:[1,64], desc:无法用OK表示时（OK表示成功且无特殊状态）必须返回*/
  "subStatusCode": "ok",
  /*ro, opt, string, 子状态码, range:[1,64], desc:无法用ok表示（ok表示成功且无特殊状态）必须返回*/
  "errorCode": 1,
  /*ro, opt, int, 错误码, desc:当statusCode不为1时，与subStatusCode对应*/
  "errorMsg": "ok",
  /*ro, opt, string, 错误信息, desc:当statusCode不为1时，必须返回,解释信息在协议约束中，允许设备在后续的版本迭代中，进行优化丰富提升（不限制死）*/
  "id": 1,
  /*ro, opt, int, id信息, desc:一般是在添加操作时返回(也有例外，依情况而定)，批量操作时不生效*/
  "AttachInfo": {
    /*ro, opt, object, 批量操作返回状态附加信息, desc:statusCode为8（Batch Operation#批量操作）时返回*/
    "StatusList": [
      /*ro, opt, array, 状态信息列表, subType:object*/
      {
        "Status": {
          /*ro, opt, object, 状态信息*/
          "index": 1,
          /*ro, req, int, id编号*/
          "statusCode": 1,
          /*ro, opt, int, 状态码, desc:无法用1表示时（1表示成功且无特殊状态）必须返回*/
          "statusString": "OK",
          /*ro, opt, string, 状态描述, range:[1,64], desc:无法用OK表示时（OK表示成功且无特殊状态）必须返回*/
          "subStatusCode": "ok",
          /*ro, opt, string, 子状态码, range:[1,64], desc:无法用ok表示（ok表示成功且无特殊状态）必须返回*/
          "errorCode": 1,
          /*ro, opt, int, 错误码, desc:当statusCode不为1时，与subStatusCode对应*/
          "errorMsg": "ok",
          /*ro, opt, string, 错误信息, desc:当statusCode不为1时，必须返回,解释信息在协议约束中，允许设备在后续的版本迭代中，进行优化丰富提升（不限制死）*/
          "id": "test"
          /*ro, opt, string, 资源的唯一识别号, desc:用于识别添加成功的资源的唯一识别号,由设备生成分配的ID,批量添加资源时使用*/
        }
      }
    ]
  }
}

```

69.37.20 添加虹膜信息

Request URL

POST /ISAPI/AccessControl/IrisInfo/record?format=json&security=<security>&iv=<iv>

查询参数

参数名称	参数类型	描述
security	string	没有security表示数据不加密； security=1，表示报文中的employeeNo，irisModelData字段AES128加密； security=2，表示报文中的employeeNo，irisModelData字段AES256加密
iv	string	iv表示初始化向量，在security=1或者2时为必须参数

请求报文

```
{
  "IrisInfo": {
    /*req, object, 虹膜信息, desc:虹膜的单双目下发成功与否, 与设备的单双目认证模式无关*/
    "employeeNo": "test",
    /*req, string, 工号*/
    "irisPicDataType": "url",
    /*req, enum, 虹膜图片数据传输方式, subType:string, [url#存储服务器url,binary#二进制数据],
    desc:url时, 报文采用非表单方式
    binary时, 报文采用表单方式*/
    "IrisData": [
      /*req, array, 虹膜数据, subType:object, desc:模板数据和图片数据必须有其一,否则失败;下发两个虹膜数据时,支持一个成功一个失败的情况,此时响应报文总状态
      为失败,子状态中返回每个虹膜数据的成功失败情况*/
      {
        "id": 1,
        /*req, int, 虹膜数据序号, range:[1,2], desc:与响应结果中AttachInfo节点内的id一一对应,AttachInfo节点内的id为string类型是为了兼容其他业务存在
        字符串的情况,本协议中使用"1"表示即可;id仅支持1和2,否则失败*/
        "irisModelData": "test",
        /*opt, string, 虹膜模板数据,
        desc:必须base处理,防止模型数据出现'\0'字符。
        当下发模板+图片时,以模板数据为准,且不查重,设备不判断模板有效性*/
        "irisPicData": "test"
        /*opt, string, 虹膜图片数据, desc:当irisPicDataType为url时,此处为url字符串;当irisPicDataType为binary时,此处无用途*/
      }
    ]
  }
}
```

参数名称	参数值	参数类型 (Content-Type)	Content-ID	文件名称 (filename)	描述
IrisInfo	[报文内容]	application/json	--	--	--
设备未实现	[图片二进制数据]	application/x-bmp		设备未实现	name和filename设备不做校验, 不区分左右眼虹膜图片顺序, 兼容image/bmp
设备未实现	[图片二进制数据]	application/x-bmp		设备未实现	name和filename设备不做校验, 不区分左右眼虹膜图片顺序, 兼容image/bmp

说明： 此协议通过表格式传输，表单框架详情说明见章节4.5.1.4，如下实例：

```
--<frontier>
Content-Disposition: form-data; name=参数名称;filename=文件名称
Content-Type: 参数类型
Content-Length: ****
Content-ID: 内容ID
参数值
```

- 参数名称：表单单元Headers中Content-Disposition的name属性，表示此表单单元的名称
- 参数类型（Content-Type）： 表单单元Headers中Content-Type属性
- 文件名称（filename）： 表单单元Headers中Content-Disposition的filename属性，仅在本表单单元传输数据为文件类型存在，表示表单单元Body的文件名
- 参数值： 表单单元的body内容

响应报文

```

{
  "statusCode": 1,
  /*ro, opt, int, 状态码, desc:无法用1表示时（1表示成功且无特殊状态）必须返回*/
  "statusString": "OK",
  /*ro, opt, string, 状态描述, range:[1,64], desc:无法用OK表示时（OK表示成功且无特殊状态）必须返回*/
  "subStatusCode": "ok",
  /*ro, opt, string, 子状态码, range:[1,64], desc:无法用ok表示（ok表示成功且无特殊状态）必须返回*/
  "errorCode": 1,
  /*ro, opt, int, 错误码, desc:当statusCode不为1时，与subStatusCode对应*/
  "errorMsg": "ok",
  /*ro, opt, string, 错误信息, desc:当statusCode不为1时，必须返回,解释信息在协议约束中，允许设备在后续的版本迭代中，进行优化丰富提升（不限制死）*/
  "id": 1,
  /*ro, opt, int, id信息, desc:一般是在添加操作时返回(也有例外，依情况而定)，批量操作时不生效*/
  "AttachInfo": {
    /*ro, opt, object, 批量操作返回状态附加信息, desc:statusCode为8（Batch Operation#批量操作）时返回*/
    "StatusList": [
      /*ro, opt, array, 状态信息列表, subType:object*/
      {
        "Status": {
          /*ro, opt, object, 状态信息*/
          "index": 1,
          /*ro, req, int, id编号*/
          "statusCode": 1,
          /*ro, opt, int, 状态码, desc:无法用1表示时（1表示成功且无特殊状态）必须返回*/
          "statusString": "OK",
          /*ro, opt, string, 状态描述, range:[1,64], desc:无法用OK表示时（OK表示成功且无特殊状态）必须返回*/
          "subStatusCode": "ok",
          /*ro, opt, string, 子状态码, range:[1,64], desc:无法用ok表示（ok表示成功且无特殊状态）必须返回*/
          "errorCode": 1,
          /*ro, opt, int, 错误码, desc:当statusCode不为1时，与subStatusCode对应*/
          "errorMsg": "ok",
          /*ro, opt, string, 错误信息, desc:当statusCode不为1时，必须返回,解释信息在协议约束中，允许设备在后续的版本迭代中，进行优化丰富提升（不限制死）*/
          "id": "test"
          /*ro, opt, string, 资源的唯一识别号, desc:用于识别添加成功的资源的唯一识别号,由设备生成分配的ID,批量添加资源时使用*/
        }
      }
    ]
  }
}

```

69.37.21 查询单个虹膜信息

Request URL

POST /ISAPI/AccessControl/IrisInfo/singleSearch?format=json&security=<security>&iv=<iv>

查询参数

参数名称	参数类型	描述
security	string	没有security表示数据不加密； security=1，表示报文中的employeeNo,irisModelData字段AES128加密； security=2，表示报文中的employeeNo,irisModelData字段AES256加密
iv	string	iv表示初始化向量，在security=1或者2时为必须参数

请求报文

```

{
  "IrisInfoSingleSearch": {
    /*req, object, 查询条件*/
    "employeeNo": "test",
    /*req, string, 工号（人员ID）*/
    "searchResultPosition": 0
  }
}

```

响应报文

```
{
  "IrisInfo": {
    /*ro, req, object, 虹膜信息, desc:虹膜的单双目下发成功与否, 与设备的单双目认证模式无关*/
    "employeeNo": "test",
    /*ro, req, string, 工号*/
    "irisPicDataType": "url",
    /*ro, req, enum, 虹膜图片数据传输方式, subType:string, [url#存储服务器url,binary#二进制数据],
    desc:url时, 报文采用非表单方式
    binary时, 报文采用表单方式*/
    "IrisData": [
      /*ro, req, array, 虹膜数据, subType:object, desc:模板数据和图片数据必须有其一,否则失败;下发两个虹膜数据时,支持一个成功一个失败的情况,此时响应报文总
      状态为失败,子状态中返回每个虹膜数据的成功失败情况*/
      {
        "id": 1,
        /*ro, req, int, 虹膜数据序号, range:[1,2], desc:与响应结果中AttachInfo节点内的id一一对应,AttachInfo节点内的id为string类型是为了兼容其他业务
        存在字符串的情况,本协议中使用"1"表示即可;id仅支持1和2,否则失败*/
        "irisModelData": "test",
        /*ro, opt, string, 虹膜模板数据,
        desc:必须base处理,防止模型数据出现'\0'字符。
        当下发模板+图片时,以模板数据为准,且不查重,设备不判断模板有效性*/
        "irisPicData": "test"
        /*ro, opt, string, 虹膜图片数据, desc:当irisPicDataType为url时,此处为url字符串;当irisPicDataType为binary时,此处与图片的Content-ID严格对应*/
      }
    ]
  }
}
```

参数名称	参数值	参数类型 (Content-Type)	Content-ID	文件名称 (filename)	描述
设备未实现	[报文内容]	application/json	--	--	--
设备未实现	[图片二进制数据]	application/x-bmp		设备未实现	name和filename设备不做校验,不区分左右眼虹膜图片顺序,兼容image/bmp
设备未实现	[图片二进制数据]	application/x-bmp		设备未实现	name和filename设备不做校验,不区分左右眼虹膜图片顺序,兼容image/bmp

说明：此协议通过表单元格式传输，表单框架详情说明见章节4.5.1.4，如下实例：

```
--<frontier>
Content-Disposition: form-data; name=参数名称;filename=文件名称
Content-Type: 参数类型
Content-Length: ****
Content-ID: 内容ID
参数值
```

- 参数名称：表单元Headers中Content-Disposition的name属性，表示此表单元的名称
- 参数类型（Content-Type）：表单元Headers中Content-Type属性
- 文件名称（filename）：表单元Headers中Content-Disposition的filename属性，仅在本表单元传输数据为文件类型存在，表示表单元Body的文件名
- 参数值：表单元的body内容

69.37.22 配置虹膜信息

Request URL

PUT /ISAPI/AccessControl/IrisInfo/setup?format=json&security=<security>&iv=<iv>

查询参数

参数名称	参数类型	描述
security	string	没有security表示数据不加密； security=1，表示报文中的employeeNo,irisModelData字段AES128加密； security=2，表示报文中的employeeNo,irisModelData字段AES256加密
iv	string	iv表示初始化向量，在security=1或者2时为必须参数

请求报文

```
{
  "IrisInfo": {
    /*req, object, 虹膜信息, desc:虹膜的单双目下发成功与否, 与设备的单双目认证模式无关*/
    "employeeNo": "test",
    /*req, string, 工号*/
    "irisPicDataType": "url",
    /*req, enum, 虹膜图片数据传输方式, subType:string, [url#存储服务器url,binary#二进制数据],
    desc:url时, 报文采用非表单方式
    binary时, 报文采用表单方式*/
    "IrisData": [
      /*req, array, 虹膜数据, subType:object, desc:模板数据和图片数据必须有其一,否则失败;下发两个虹膜数据时,支持一个成功一个失败的情况,此时响应报文总状态
      为失败,子状态中返回每个虹膜数据的成功失败情况*/
      {
        "id": 1,
        /*req, int, 虹膜数据序号, range:[1,2], desc:与响应结果中AttachInfo节点内的id一一对应,AttachInfo节点内的id为string类型是为了兼容其他业务存在
        字符串的情况,本协议中使用"1"表示即可;id仅支持1和2,否则失败*/
        "irisModelData": "test",
        /*opt, string, 虹膜模板数据,
        desc:必须base处理,防止模型数据出现'\0'字符。
        当下发模板+图片时,以模板数据为准,且不查重,设备不判断模板有效性。
        加密时进行两次base64处理后,再做aes加密*/
        "irisPicData": "test"
        /*opt, string, 虹膜图片数据, desc:当irisPicDataType为url时,此处为url字符串;当irisPicDataType为binary时,此处无用途*/
      }
    ]
  }
}
```

参数名称	参数值	参数类型 (Content-Type)	Content-ID	文件名称 (filename)	描述
IrisInfo	[报文内容]	application/json	--	--	--
设备未实现	[图片二进制数据]	application/x-bmp		设备未实现	name和filename设备不做校验,不区分左右眼虹膜图片顺序,兼容image/bmp
设备未实现	[图片二进制数据]	application/x-bmp		设备未实现	name和filename设备不做校验,不区分左右眼虹膜图片顺序,兼容image/bmp

说明：此协议通过表格式传输，表单框架详情说明见章节4.5.1.4，如下实例：

```
--<frontier>
Content-Disposition: form-data; name=参数名称; filename=文件名称
Content-Type: 参数类型
Content-Length: ****
Content-ID: 内容ID
参数值
```

- 参数名称：表单单元Headers中Content-Disposition的name属性，表示此表单单元的名称
- 参数类型（Content-Type）：表单单元Headers中Content-Type属性
- 文件名称（filename）：表单单元Headers中Content-Disposition的filename属性，仅在本表单单元传输数据为文件类型存在，表示表单单元Body的文件名
- 参数值：表单单元的body内容

响应报文

```
{
  "statusCode": 1,
  /*ro, opt, int, 状态码, desc:无法用1表示时（1表示成功且无特殊状态）必须返回*/
  "statusString": "OK",
  /*ro, opt, string, 状态描述, range:[1,64], desc:无法用OK表示时（OK表示成功且无特殊状态）必须返回*/
  "subStatusCode": "ok",
  /*ro, opt, string, 子状态码, range:[1,64], desc:无法用ok表示（ok表示成功且无特殊状态）必须返回*/
  "errorCode": 1,
  /*ro, opt, int, 错误码, desc:当statusCode不为1时，与subStatusCode对应*/
  "errorMsg": "ok",
  /*ro, opt, string, 错误信息, desc:当statusCode不为1时，必须返回,解释信息在协议约束中，允许设备在后续的版本迭代中，进行优化丰富提升（不限制死）*/
  "id": 1,
  /*ro, opt, int, id信息, desc:一般是在添加操作时返回(也有例外，依情况而定)，批量操作时不生效*/
  "AttachInfo": {
    /*ro, opt, object, 批量操作返回状态附加信息, desc:statusCode为8（Batch Operation#批量操作）时返回*/
    "StatusList": [
      /*ro, opt, array, 状态信息列表, subType:object*/
      {
        "Status": {
          /*ro, opt, object, 状态信息*/
          "index": 1,
          /*ro, req, int, id编号*/
          "statusCode": 1,
          /*ro, opt, int, 状态码, desc:无法用1表示时（1表示成功且无特殊状态）必须返回*/
          "statusString": "OK",
          /*ro, opt, string, 状态描述, range:[1,64], desc:无法用OK表示时（OK表示成功且无特殊状态）必须返回*/
          "subStatusCode": "ok",
          /*ro, opt, string, 子状态码, range:[1,64], desc:无法用ok表示（ok表示成功且无特殊状态）必须返回*/
          "errorCode": 1,
          /*ro, opt, int, 错误码, desc:当statusCode不为1时，与subStatusCode对应*/
          "errorMsg": "ok",
          /*ro, opt, string, 错误信息, desc:当statusCode不为1时，必须返回,解释信息在协议约束中，允许设备在后续的版本迭代中，进行优化丰富提升（不限制死）*/
          "id": "test"
          /*ro, opt, string, 资源的唯一识别号, desc:用于识别添加成功的资源的唯一识别号,由设备生成分配的ID,批量添加资源时使用*/
        }
      }
    ]
  }
}
```

69.37.23 查询人员信息

Request URL

POST /ISAPI/AccessControl/UserInfo/Search?format=json&security=<security>&iv=<iv>

查询参数

参数名称	参数类型	描述
security	string	没有security表示数据不加密； security=1,表示报文中的employeeNo、name、password、userName和dynamicCode字段AES128加密； security=2,表示报文中的employeeNo、name、password、userName和dynamicCode字段AES256加密；
iv	string	iv表示初始化向量，在security=1或者2时为必须参数;

请求报文

```
{
  "UserInfoSearchCond": {
    /*req, object, 查询条件*/
    "searchID": "123",
    /*req, string, 搜索记录唯一标识, desc:搜索记录唯一标识,用来确认上层客户端是否为同一个(倘若是一个,设备记录内存,下次搜索加快速度),string类型*/
    "searchResultPosition": 0,
    /*req, int, 查询结果在结果列表中的起始位置, desc:当记录条数很多时,一次查询不能获取所有的记录,下一次查询时指定位置可以查询后面的记录(若设备支持的最大totalMatches为M个,但是当前设备已存储的totalMatches为N个(N<=M),则该字段的合法范围为0~N-1)*/
    "maxResults": 30,
    /*req, int, 本次协议调用可获取的最大记录数, desc:如maxResults值大于设备能力集返回的范围,则设备按照能力集最大值返回,设备不进行报错*/
    "EmployeeNoList": [
      /*opt, array, 人员ID列表, subType:object*/
      {
        "employeeNo": "test"
        /*req, string, 工号*/
      }
    ],
    "fuzzySearch": "test",
    /*opt, string, 模糊查询关键字*/
    "groupIdList": [1, 2, 3, 4],
    /*opt, array, 本地考勤组织编号查询范围, subType:int*/
    "groupNameList": ["name1", "name2"],
    /*opt, array, 本地考勤组织名称查询范围, subType:string*/
    "arrangeType": "personal",
    /*opt, enum, 排班类型, subType:string, [personal#个人排班,group#部门排班,all#所有已排班,none#所有未排班], desc:该字段存在时表示查询处于个人排班、部门排班、已排班状态的人*/
    "userType": "normal",
    /*opt, enum, 人员类型, subType:string, [normal#普通人(主人),visitor#来宾(访客),blackList#黑名单人,patient#患者,maintenance#维护人员], desc:维护人员需要在任意时间都能进入房间*/
    "deviceIDList": [1, 2],
    /*opt, array, 要查询的设备编号(床位)列表, subType:int*/
    "hasFace": true,
    /*opt, bool, 已录入人脸*/
    "hasCard": true,
    /*opt, bool, 已录入卡*/
    "hasFingerprint": true,
    /*opt, bool, 已录入指纹*/
    "hasIris": true,
    /*opt, bool, 已录入虹膜*/
    "userLevel": "Employee",
    /*opt, enum, 用户级别, subType:string, [Employee#普通员工,DepartmentManager#部门主管]*/
    "meetingID": "test"
    /*opt, string, 会议编号UUID, range:[1,32], desc:UI可根据会议ID查询该会议下的人员信息,然后根据会议签到状态进行过滤,在会议签到历史页面展示出正常签到、迟到的人员信息。*/
  }
}
```

响应报文

```
{
  "UserInfoSearch": {
    /*ro, req, object, 查询结果*/
    "searchID": "test",
    /*ro, req, string, 搜索记录唯一标识*/
    "responseStatusStr": "OK",
    /*ro, req, enum, 查询状态字符串描述, subType:string, [OK#查询结束,MORE#还有数据等待查询,NO MATCH#没有匹配数据], desc:OK-查询结束,MORE-还有数据等待查询,NO MATCH-没有匹配数据*/
    "numOfMatches": 1,
    /*ro, req, int, 本次返回的记录条数*/
    "totalMatches": 1,
    /*ro, req, int, 符合条件的记录总条数*/
    "UserInfo": [
      /*ro, opt, array, 人员信息, subType:object*/
      {
        "employeeNo": "test",
        /*ro, req, string, 工号(人员ID)*/
        "name": "test",
        /*ro, opt, string, 姓名*/
        "userType": "normal",
        /*ro, req, enum, 人员类型, subType:string, [normal#普通人(主人),visitor#来宾(访客),blackList#黑名单人,patient#患者,maintenance#维护人员(包括保洁、维修人员等)], desc:维护人员需要在任意时间都能进入房间*/
        "closeDelayEnabled": true,
        /*ro, opt, bool, 是否关门延迟*/
        "Valid": {
          /*ro, opt, object, 有效期参数*/
          "enable": true,
          /*ro, req, bool, 使能有效期*/
          "beginTime": "test",
          /*ro, req, string, 有效期起始时间, desc:timeType字段不存在或为local时*/
          "endTime": "test",
          /*ro, req, string, 有效期结束时间, desc:timeType字段不存在或为local时*/
          "timeType": "test"
          /*ro, opt, string, 时间类型*/
        },
        "belongGroup": "1,3,5",
        /*ro, opt, string, 所属群组*/
        "password": "123456",
        /*ro, opt, string, 密码*/
        "doorRight": "1,3",

```

```

/*ro, opt, string, 门权限, desc: (代表对门1、门3有权限) (锁权限*/
"RightPlan": [
/*ro, opt, array, 门权限计划, subType:object*/
{
  "doorNo": 1,
  /*ro, opt, int, 门编号 (锁ID) */
  "planTemplateNo": "1,3,5"
  /*ro, opt, string, 计划模板编号,
  desc: 同个门不同计划模板采用权限或的方式处理
  默认计划模板编号
  65535-7*24小时生效
  65534-周一到周五24小时生效
  65533-周六周日24小时生效*/
}
],
"maxOpenDoorTime": 0,
/*ro, opt, int, 最大认证次数*/
"openDoorTime": 0,
/*ro, opt, int, 已认证次数*/
"roomNumber": 123,
/*ro, opt, int, 房间号*/
"floorNumber": 1,
/*ro, opt, int, 层号*/
"doubleLockRight": true,
/*ro, opt, bool, 反锁开门权限, desc:true-有权限, false-无权限*/
"localUIRight": true,
/*ro, opt, bool, 是否具有设备本地UI访问权限, desc:true-有权限, false-无权限*/
"isLocalOperator": true,
/*ro, opt, bool, 是否为本地操作员,
desc:1. 消费机6301项目新增本字段。本地操作员能够进入设备后台, 但仅有消费模式和消费统计的配置权限, 权限小于管理员。
2. 本字段仅在HEOP中供UI使用。
3. 本地操作员, 人员类型userType依旧为normal, 但本字段需赋值为true, 默认为false。
4. 本字段与localUIRight互斥, 两者不可同时为true。*/
"userVerifyMode": "card",
/*ro, opt, enum, 人员验证方式, subType:string, [cardAndPw#刷卡+密码, card#刷卡, cardOrPw#刷卡或密码, fp#指纹, fpAndPw#指纹+密码, fpOrCard#指纹或
刷卡, fpAndCard#指纹+刷卡, fpAndCardAndPw#指纹+刷卡+密码, faceOrFpOrCardOrPw#人脸或指纹或刷卡或密码, faceAndFp#人脸+指纹, faceAndPw#人脸+密码, faceAndCard#人脸+刷
卡, face#人脸, employeeNoAndPw#工号+密码, fpOrPw#指纹或密码, employeeNoAndFp#工号+指纹, employeeNoAndFpAndPw#工号+指纹+密码, faceAndFpAndCard#人脸+指纹+刷
卡, faceAndPwAndFp#人脸+密码+指纹, employeeNoAndFace#工号+人脸, faceOrfaceAndCard#人脸或人脸+刷卡, fpOrface#指纹或人脸, cardOrfaceOrPw#刷卡或人脸或密码, iris#虹
膜, faceOrFpOrCardOrPwOrIris#人脸或指纹或刷卡或密码或虹膜, faceOrCardOrPwOrIris#人脸或刷卡或密码或虹膜, cardOrFace#刷卡或人脸, cardOrFaceOrFp#刷卡或人脸或指
纹, faceOrPw#人脸或密码, employeeNoAndFaceAndPw#工号+人脸+密码, cardOrFaceOrFaceAndCard#刷卡或人脸或人脸+刷卡, faceOrFpOrPw#人脸或指纹或密
码, cardOrFpOrFaceOrIris#刷卡或指纹或人脸或虹膜, fpOrFaceOrIrisOrPw#指纹或人脸或虹膜或密码, cardOrFpOrIrisOrPw#刷卡或指纹或虹膜或密码, cardOrIrisOrPw#刷卡或虹膜
或密码, cardAndIris#刷卡+虹膜, fpAndIris#指纹+虹膜, faceAndIris#人脸+虹膜, irisAndPw#虹膜+密码, cardAndIrisAndPw#刷卡+虹膜+密码, faceAndIrisAndPw#人脸+虹膜+密
码, cardAndFaceAndIris#刷卡+人脸+虹膜, Pw#密码], desc: 人员验证方式的优先级高于读卡器验证方式*/
"dynamicCode": "123456",
/*ro, opt, string, 动态权限码*/
"callNumbers": ["1-1-1-401"],
/*ro, opt, array, 呼叫号码列表, subType:string,
desc: roomNumber的扩展, 支持多房间号, 两者同时存在时, 优先解析callNumbers
国内规则:
1-1-1-层号+房间号。高三位表示层号, 范围1~999; 低两位表示房间号, 范围1~99。eg: 如层号为1, 房间号为1, 则为1-1-1-1-101; 如层号为222, 房
间号为4, 则为1-1-1-22204。
海外规则:
1-1-1-房间号。房间号的范围1~9999。eg: 如房间号为111, 则为1-1-1-111。*/
"floorNumbers": [1, 2],
/*ro, opt, array, 层号列表, subType:int,
desc: floorNumber的扩展, 支持多层, 两者同时存在时, 优先解析floorNumbers。要求floorNumbers与callNumbers的数组元素个数一一对应。
国内eg:
callNumbers为["1-1-1-101", "1-1-1-302"], 则floorNumbers为[1, 3];
callNumbers为["1-1-1-101", "1-1-1-102"], 则floorNumbers为[1, 1];
海外eg:
callNumbers为["1-1-1-1", "1-1-1-99"], 由于callNumbers未包含层号, 因此floorNumbers仅需要元素个数与callNumbers个数一一对应即可, 若
floorNumbers为[A, B], 则表示A层的1号房间和B层的99号房间*/
"groupId": 1,
/*ro, opt, int, 本地考勤组织编号*/
"localAtndPlanTemplateId": 1,
/*ro, opt, int, 本地考勤计划模板*/
"numOfFace": 0,
/*ro, opt, int, 关联人脸数量, desc:不返回不支持*/
"numOfFP": 0,
/*ro, opt, int, 关联指纹数量, desc:不返回不支持*/
"numOfCard": 0,
/*ro, opt, int, 关联卡数量, desc:不返回不支持*/
"numOfIris": 0,
/*ro, opt, int, 关联虹膜数量, desc:不返回不支持*/
"gender": "male",
/*ro, opt, enum, 人脸图片对应的人员性别, subType:string, [male#男, female#女, unknown#未知]*/
"PersonInfoExtends": [
/*ro, opt, array, 人员信息扩展, subType:object,
desc: 用于配置人员在设备UI界面的扩展信息显示
明眸当前仅支持一个value, 不支持id字段, 用于作为工号显示
智能主机按标准实现*/
{
  "id": 1,
  /*ro, opt, int, 人员信息扩展序号, range:[1,32],
  desc: 与/ISAPI/AccessControl/personInfoExtendName?format=json中的id对应, 用于关联value的name信息
  id不存在时, 根据数组顺序默认从1开始*/
  "value": "test"
  /*ro, opt, string, 人员信息扩展内容*/
}
],
"groupName": "test",
/*ro, opt, string, 组织架构, range:[1,64]*/
"age": 0,
/*ro, opt, int, 年龄, range:[0,120]*/

```

```
/*ro, opt, string, 设备编号 (对应编号配置中编号*/
"PatientInfos": {
/*ro, opt, object, 患者信息, desc:userType==patient时必须有效*/
"deviceID": "test",
/*ro, opt, string, 设备编号 (对应编号配置中编号*/
"admissionTime": "1970-01-01T00:00+08:00",
/*ro, opt, datetime, 入院时间*/
"chargeNurse": "test",
/*ro, opt, string, 责任护士, range:[0,32]*/
"chargeDoctor": "test",
/*ro, opt, string, 责任医师, range:[0,32]*/
"nursingLevel": "tertiary",
/*ro, opt, enum, 护理等级, subType:string, [unknown#未知,tertiary#三级护理,secondary#二级护理,primary#一级护理,special#特级护理]*/
"doctorsAdvice": "test",
/*ro, opt, string, 医嘱注意事项, range:[0,128]*/
"allergicHistory": "test"
/*ro, opt, string, 过敏情况, range:[0,128]*/
},
"TromboneRule": {
/*ro, opt, object, 长号规则信息*/
"industryType": "buildings",
/*ro, opt, enum, 行业(场景)类型, subType:string, [buildings#楼宇,prison#监所,medicalTreatment#医疗,broadcasting#广播], desc:人员关联设
备的行业类型;设备新长号的命名规则为: HI#KD#V10#001001001001003格式。其中HI表示行业,KD表示设备类型,V10表示设备的私有SIP版本号*/
"unitType": "indoor",
/*ro, opt, enum, 设备类型, subType:string, [indoor#室内机,villa#别墅,门口机,confirm#二次确认机,outdoor#门口机,fence#围墙机,doorbell#门铃
机,manage#管理机,acs#门禁设备,wardStation#探视分机,bedheadExtension#床头分机,bedsideExtension#床旁分机,terminal#终端,netAudio#网络音响,interactive#交互终
端,amplifier#功放设备], desc:人员关联设备的设备类型;设备新长号的命名规则为: HI#KD#V10#001001001001003格式。其中HI表示行业,KD表示设备类型,V10表示设备的私有
SIP版本号*/
"SIPVersion": "V10"
/*ro, opt, string, SIP(指私有SIP)版本, range:[0,32], desc:人员关联设备的私有SIP版本号,设备新长号的命名规则为: HI#KD#V10#001001001001003
格式。其中HI表示行业,KD表示设备类型,V10表示设备的私有SIP版本号*/
},
"ESDType": "handAndFoot",
/*ro, opt, enum, ESD检测类型 , subType:string, [handAndFoot#手脚都检测,no#不检测,hand#检测手,foot#检测脚], desc:默认为no*/
"userLevel": "Employee",
/*ro, opt, enum, 用户级别, subType:string, [Employee#普通员工,DepartmentManager#部门主管]*/
"userName": "test",
/*ro, opt, string, 登录账号, range:[1,32], desc:数字、小写字母、大写字母、特殊字符(敏感信息加密)*/
"faceURL": "test",
/*ro, opt, string, 人脸图片URL,
desc:8000端口透传及80端口时, 此处返回设备本地存储URL:
ISUP及萤石链路采用存储服务的URL: */
"LocalAttendance": {
/*ro, opt, object, 本地考勤信息*/
"templateName": "test"
/*ro, opt, string, 模板/排班名称, desc:该人员当前所属的排班名称*/
},
"meetingSignInStatus": "normal",
/*ro, opt, enum, 会议签到状态, subType:string, [normal#正常签到,late#迟到,notSignIn#未签到]*/
"deviceID": "test"
/*ro, opt, string, 设备编号 (对应编号配置中编号*/
}
}
}
```

69.37.24 获取人员信息参数能力

Request URL

GET /ISAPI/AccessControl/UserInfo/capabilities?format=json&terminalNo=<terminalNo>

查询参数

参数名称	参数类型	描述
terminalNo	string	终端编号、仅对信息发布产品使用

请求报文

无

响应报文

```
{
  "UserInfo": {
    /*ro, req, object, 人员信息*/
    "supportFunction": {
      /*ro, req, object, 支持人员的增删改查、总数查询功能, desc:post-创建, delete-删除, put-修改, get-查询, setUp-设置 (对应
URL: /ISAPI/AccessControl/UserInfo/SetUp?format=json) */
      "@opt": "post,delete,put,get,setUp"
      /*ro, opt, string, 操作, desc:post,delete,put,get,setUp*/
    },
    "UserInfoSearchCond": {
      /*ro, opt, object, 查找条件*/
      "maxResults": {
        /*ro, req, object, 本次协议调用可获取的最大记录数*/
      }
    }
  }
}
```

```

        "@min": 1,
        /*ro, opt, int, 最小值*/
        "@max": 100
        /*ro, opt, int, 最大值*/
    },
    "EmployeeNoList": {
        /*ro, opt, object, 人员ID列表*/
        "maxSize": 56,
        /*ro, opt, int, 人员最大数目*/
        "employeeNo": {
            /*ro, opt, object, 工号（人员ID）*/
            "@min": 1,
            /*ro, opt, int, 最小值*/
            "@max": 32
            /*ro, opt, int, 最大值*/
        }
    },
    "fuzzySearch": {
        /*ro, opt, object, 模糊查询关键字*/
        "@min": 1,
        /*ro, opt, int, 最小值*/
        "@max": 32
        /*ro, opt, int, 最大值*/
    },
    "isSupportNumOfFace": true,
    /*ro, opt, bool, 查询时是否支持关联人脸数量*/
    "isSupportNumOfFP": true,
    /*ro, opt, bool, 查询时是否支持关联指纹数量*/
    "isSupportNumOfCard": true,
    /*ro, opt, bool, 查询时是否支持关联卡数量*/
},
"UserInfoDelCond": {
    /*ro, opt, object, 删除条件*/
    "EmployeeNoList": {
        /*ro, opt, object, 人员ID列表*/
        "maxSize": 56,
        /*ro, opt, int, 人员最大数目*/
        "employeeNo": {
            /*ro, opt, object, 工号（人员ID）*/
            "@min": 1,
            /*ro, opt, int, 最小值*/
            "@max": 32
            /*ro, opt, int, 最大值*/
        }
    }
},
"employeeNo": {
    /*ro, req, object, 工号（人员ID）*/
    "@min": 1,
    /*ro, opt, int, 最小值*/
    "@max": 32
    /*ro, opt, int, 最大值*/
},
"name": {
    /*ro, opt, object, 姓名*/
    "@min": 1,
    /*ro, opt, int, 最小值*/
    "@max": 128
    /*ro, opt, int, 最大值*/
},
"userType": {
    /*ro, req, object, 人员类型*/
    "@opt": "normal,visitor,blackList"
    /*ro, opt, string, 取值范围*/
},
"closeDelayEnabled": "true,false",
/*ro, opt, string, 是否关门延迟*/
"Valid": {
    /*ro, req, object, 有效期参数（enable不能代表长期有效）*/
    "enable": "true,false",
    /*ro, req, string, 使能有效期*/
    "beginTime": {
        /*ro, req, object, 有效期起始时间（该时间为设备本地时间还是UTC时间时间根据timeType确定）*/
        "@min": 1,
        /*ro, opt, int, 最小值*/
        "@max": 32
        /*ro, opt, int, 最大值*/
    },
    "endTime": {
        /*ro, req, object, 有效期结束时间（该时间为设备本地时间还是UTC时间时间根据timeType确定）*/
        "@min": 1,
        /*ro, opt, int, 最小值*/
        "@max": 32
        /*ro, opt, int, 最大值*/
    },
    "timeRangeBegin": "1970-01-01T00:00:00",
    /*ro, opt, string, beginTime和endTime可配置的起始时间点, desc:能力集如不返回该字段, 则设备默认的可配起始时间点为1970-01-01T00:00:00*/
    "timeRangeEnd": "2037-12-31T23:59:59",
    /*ro, opt, string, beginTime和endTime可配置的结束时间点, desc:能力集如不返回该字段, 则设备默认的可配起始时间点为1970-01-01T00:00:00*/
    "timeType": {
        /*ro, opt, object, 时间类型*/
        "@opt": "local,UTC"
        /*ro, opt, string, 取值范围*/
    }
}

```

湖南云舟电力科技有限公司

2024-08-15

```

    },
    "maxBelongGroup": 4,
    /*ro, opt, int, 一个人员最多属于几个群组的数目*/
    "belongGroup": {
    /*ro, opt, object, 所属群组*/
        "@min": 1,
        /*ro, opt, int, 最小值*/
        "@max": 32
        /*ro, opt, int, 最大值*/
    },
    "password": {
    /*ro, opt, object, 密码*/
        "@min": 1,
        /*ro, opt, int, 最小值*/
        "@max": 32
        /*ro, opt, int, 最大值*/
    },
    "doorRight": {
    /*ro, opt, object, 门权限（代表对门1、门3有权限）（锁权限*/
        "@min": 1,
        /*ro, opt, int, 最小值*/
        "@max": 32
        /*ro, opt, int, 最大值*/
    },
    "RightPlan": {
    /*ro, opt, object, 门权限计划（锁权限计划）*/
        "maxSize": 32,
        /*ro, opt, int, 最大计划数目*/
        "doorNo": {
        /*ro, opt, object, 门编号（锁ID）*/
            "@min": 1,
            /*ro, opt, int, 最小值*/
            "@max": 32
            /*ro, opt, int, 最大值*/
        },
        "maxPlanTemplate": 4,
        /*ro, opt, int, 一个门最多可配计划模板的数目*/
        "planTemplateNo": {
        /*ro, opt, object, 计划模板编号*/
            "@min": 1,
            /*ro, opt, int, 最小值*/
            "@max": 32,
            /*ro, opt, int, 最大值*/
        }
    },
    "maxOpenDoorTime": {
    /*ro, opt, object, 最大认证次数*/
        "@min": 0,
        /*ro, opt, int, 最小值*/
        "@max": 100
        /*ro, opt, int, 最大值*/
    },
    "openDoorTime": {
    /*ro, opt, object, 已认证次数*/
        "@min": 0,
        /*ro, opt, int, 最小值*/
        "@max": 100
        /*ro, opt, int, 最大值*/
    },
    "roomNumber": {
    /*ro, opt, object, 房间号*/
        "@min": 0,
        /*ro, opt, int, 最小值*/
        "@max": 100
        /*ro, opt, int, 最大值*/
    },
    "floorNumber": {
    /*ro, opt, object, 层号*/
        "@min": 0,
        /*ro, opt, int, 最小值*/
        "@max": 100
        /*ro, opt, int, 最大值*/
    },
    "localUIRight": "true,false",
    /*ro, opt, string, 是否具有设备本地UI访问权限*/
    "userVerifyMode": {
    /*ro, opt, object, 人员验证方式（人员验证方式的优先级高于读卡器验证方式）*/
        "@opt":
        "cardAndPw,card,cardOrPw,fp,fpAndPw,fpOrCard,fpAndCard,fpAndCardAndPw,faceOrFpOrCardOrPw,faceAndFp,faceAndPw,faceAndCard,face,employeeNoAndPw,fpOrPw,employeeNoAndFp,employeeNoAndFpAndPw,faceAndFpAndCard,faceAndPwAndFp,employeeNoAndFace,faceOrFaceAndCard,fpOrface,cardOrfaceOrPw,cardOrFace,cardOrFaceOrFp,cardOrFpOrPw,faceOrPw,employeeNoAndFaceAndPw,cardOrFaceOrFaceAndCard,iris,faceOrFpOrCardOrPwOrIris,faceOrCardOrPwOrIris,cardOrFpOrFaceOrIris,fpOrFaceOrIrisOrPw,cardOrFpOrIrisOrPw,cardOrIrisOrPw,cardAndIris,faceAndIris,irisAndPw,cardAndIrisAndPw,faceAndIrisAndPw,cardAndFaceAndIris,Pw"
        /*ro, opt, string, 取值范围, desc:cardAndPw-刷卡+密码,card-刷卡,cardOrPw-刷卡或密码,fp-指纹,fpAndPw-指纹+密码,fpOrCard-指纹或刷卡,fpAndCard-指纹+刷卡,fpAndCardAndPw-指纹+刷卡+密码,faceOrFpOrCardOrPw-人脸或指纹或刷卡或密码,faceAndFp-人脸+指纹,faceAndPw-人脸+密码,faceAndCard-人脸+刷卡,face-人脸,employeeNoAndPw-工号+密码,fpOrPw-指纹或密码,employeeNoAndFp-工号+指纹,employeeNoAndFpAndPw-工号+指纹+密码,faceAndFpAndCard-人脸+指纹+刷卡,faceAndPwAndFp-人脸+密码+指纹,employeeNoAndFace-工号+人脸,faceOrFaceAndCard-人脸或人脸+刷卡,fpOrface-指纹或人脸,cardOrfaceOrPw-刷卡或人脸或密码,cardOrFace-刷卡或人脸,cardOrFaceOrFp-刷卡或人脸或指纹,cardOrFpOrPw-刷卡或指纹或密码,faceOrPw-人脸或密码,employeeNoAndFaceAndPw-工号+人脸+密码,cardOrFaceOrFaceAndCard-刷卡或人脸或人脸+刷卡,iris-虹膜,faceOrFpOrCardOrPwOrIris-人脸或指纹或卡或密码或虹膜,faceOrCardOrPwOrIris-人脸或卡或指纹或虹膜,cardOrFpOrFaceOrIris-刷卡或指纹或人脸或虹膜,fpOrFaceOrIrisOrPw-指纹或人脸或虹膜或密码,cardOrFpOrIrisOrPw-刷卡或指纹或虹膜或密码,cardOrIrisOrPw-刷卡或虹膜或密码,cardAndIris-刷卡+虹膜,fpAndIris-指纹+虹膜,faceAndIris-人脸+虹膜,irisAndPw-虹膜+密码,cardAndIrisAndPw-刷卡+虹膜+密码,faceAndIrisAndPw-人脸+虹膜+密码,cardAndFaceAndIris-刷卡+人脸+虹膜,Pw-密码*/
    },
    "maxRecordNum": 0,

```

```

/*ro, opt, int, 支持的最大记录条数（人员数量）（该字段对于平台集成十分重要*/
"gender": {
/*ro, opt, object, 人脸图片对应的人员性别*/
"@opt": "male,female,unknown"
/*ro, opt, string, 取值范围*/
},
"PersonInfoExtends": {
/*ro, opt, object, 人员扩展字段*/
"maxSize": 3,
/*ro, req, int, 支持多少个扩展字段: */
"value": {
/*ro, opt, object, 人员信息扩展内容*/
"@min": 0,
/*ro, opt, int, 最小值*/
"@max": 100
/*ro, opt, int, 最大值*/
}
},
"callNumbers": {
/*ro, opt, object, 呼叫号码列表*/
"maxSize": 0,
/*ro, opt, int, 数组元素个数限制*/
"@min": 0,
/*ro, opt, int, 最小值*/
"@max": 100
/*ro, opt, int, 最大值*/
},
"floorNumbers": {
/*ro, opt, object, 层号列表*/
"maxSize": 0,
/*ro, opt, int, 数组元素个数限制*/
"@min": 0,
/*ro, opt, int, 最小值*/
"@max": 100
/*ro, opt, int, 最大值*/
},
"purePwdVerifyEnable": true,
/*ro, opt, bool, 纯密码开门,
desc:true-设备支持纯密码开门（人员信息中的-password字段），不返回本字段-不支持纯密码开门
*纯密码方案:
*a认证方式中的"或密码"为人员密码
*a设备不对人员密码的重复性进行校验，需要上层平台自行保证密码的唯一性
*a设备本地不能对人员密码进行操作（增删改查）*/
}
}

```

69.37.25 获取指纹删除能力

Request URL

GET /ISAPI/AccessControl/FingerPrint/Delete/capabilities?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "FingerPrintDelete": {
    /*ro, opt, object, 删除能力*/
    "mode": {
      /*ro, req, object, 模式*/
      "@opt": "byEmployeeNo,byCardReader"
      /*ro, opt, string, 删除方式, desc:byEmployeeNo-按工号（人员ID）删除, byCardReader-按读卡器删除*/
    },
    "EmployeeNoDetail": {
      /*ro, opt, object, 按工号（人员ID）删除信息（mode为byEmployeeNo有效）*/
      "employeeNo": {
        /*ro, opt, object, 指纹关联的工号*/
        "@min": 1,
        /*ro, opt, int, 最小值*/
        "@max": 32
        /*ro, opt, int, 最大值*/
      },
      "enableCardReader": {
        /*ro, opt, object, 删除指纹的读卡器*/
        "@min": 1,
        /*ro, opt, int, 最小值*/
        "@max": 512
        /*ro, opt, int, 最大值*/
      },
      "fingerPrintID": {
        /*ro, opt, object, 需要删除的手指编号*/
        "@min": 1,
        /*ro, opt, int, 最小值*/
        "@max": 10
        /*ro, opt, int, 最大值*/
      }
    },
    "CardReaderDetail": {
      /*ro, opt, object, 按读卡器删除信息（mode为byCardReader有效）*/
      "cardReaderNo": {
        /*ro, opt, object, 指纹读卡器编号*/
        "@min": 1,
        /*ro, opt, int, 最小值*/
        "@max": 512
        /*ro, opt, int, 最大值*/
      },
      "clearAllCard": "true,false",
      /*ro, opt, string, 是否删除所有卡的指纹信息: false-否（按工号删除）*/
      "employeeNo": {
        /*ro, opt, object, 指纹关联的工号*/
        "@min": 1,
        /*ro, opt, int, 最小值*/
        "@max": 32
        /*ro, opt, int, 最大值*/
      }
    }
  }
}
```

69.37.26 删除虹膜信息

Request URL

PUT /ISAPI/AccessControl/IrisInfo/delete?format=json&security=<security>&iv=<iv>

查询参数

参数名称	参数类型	描述
security	string	没有security表示数据不加密； security=1，表示报文中的employeeNo字段AES128加密； security=2，表示报文中的employeeNo字段AES256加密
iv	string	iv表示初始化向量，在security=1或者2时为必须参数

请求报文

```
{
  "IrisInfoDelCond": {
    /*opt, object, 删除条件*/
    "EmployeeNoList": [
      /*opt, array, 工号列表, subType:object, desc:字段为空表示删除设备所有虹膜*/
      {
        "employeeNo": "test",
        /*opt, string, 工号*/
        "irisIdList": [1, 2]
        /*opt, array, 虹膜序号, subType:int,
        desc:表示要删除的某工号下的虹膜序号
        仅支持1和2,当且仅当employeeNo存在时有效,当该字段不存在或数组为空时表示删除当前工号下所有的虹膜数据*/
      }
    ]
  }
}
```

响应报文

```
{
  "statusCode": 1,
  /*ro, opt, int, 状态码, desc:无法用1表示时（1表示成功且无特殊状态）必须返回*/
  "statusString": "OK",
  /*ro, opt, string, 状态描述, range:[1,64], desc:无法用OK表示时（OK表示成功且无特殊状态）必须返回*/
  "subStatusCode": "ok",
  /*ro, opt, string, 子状态码, range:[1,64], desc:无法用ok表示（ok表示成功且无特殊状态）必须返回*/
  "errorCode": 1,
  /*ro, opt, int, 错误码, desc:当statusCode不为1时, 与subStatusCode对应*/
  "errorMsg": "ok"
  /*ro, opt, string, 错误信息, desc:当statusCode不为1时, 必须返回,解释信息在协议约束中, 允许设备在后续的版本迭代中, 进行优化丰富提升（不限制死）*/
}
```

69.37.27 获取指纹删除进度

Request URL

GET /ISAPI/AccessControl/FingerPrint/DeleteProcess?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "FingerPrintDeleteProcess": {
    /*ro, req, object, 未定义*/
    "status": "processing"
    /*ro, req, enum, 状态, subType:string, [processing#处理中,success#成功,failure#失败]*/
  }
}
```

69.37.28 获取全部工号的卡数量

Request URL

GET /ISAPI/AccessControl/CardInfo/Count?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "CardInfoCount": {
    /*ro, req, object, 卡数量*/
    "cardNumber": 100
    /*ro, req, int, 卡数量*/
  }
}
```

69.37.29 获取指纹参数下发进度

Request URL

GET /ISAPI/AccessControl/FingerPrintProgress?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "FingerPrintStatus": {
    /*ro, opt, object, 指纹状态*/
    "StatusList": [
      /*ro, opt, array, 状态列表, subType:object*/
      {
        "id": 1,
        /*ro, opt, int, 读卡器编号*/
        "cardReaderRecvStatus": 1,
        /*ro, opt, enum, 指纹读卡器状态, subType:int, [0#失败,1#成功,2#该指纹模组不在线,3#重试或指纹质量差,4#内存已满,5#已存在该指纹,6#已存在该指纹ID,7#非法指纹ID,8#该指纹模组无需配置,10#指纹读卡器版本过低（无法支持工号）】*/
        "errorMsg": "test"
        /*ro, opt, string, 下发错误信息, range:[1,32], desc:当cardReaderRecvStatus为5时，表示已存在指纹对应的工号（人员ID）*/
      }
    ],
    "totalStatus": 1
    /*ro, req, enum, 下发总的状态, subType:int, [0#当前指纹未下完所有读卡器,1#已下完所有读卡器]*/
  }
}
```

69.37.30 获取人员图片异步下发能力集

Request URL

GET /ISAPI/AccessControl/UserPic/asynclImportDataTasks/capabilities?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "AsyncImportDatasCap": {
    /*ro, opt, object, 人员图片异步下发能力节点*/
    "taskID": {
      /*ro, opt, object, 下发任务ID字符串长度*/
      "@min": 1,
      /*ro, opt, int, 最小值*/
      "@max": 64
      /*ro, req, int, 最大值*/
    },
    "taskNum": 1,
    /*ro, req, int, 设备同时支持的任务数目*/
    "URL": {
      /*ro, opt, object, 数据文件的URL长度*/
      "@min": 1,
      /*ro, opt, int, 最小值*/
      "@max": 256
      /*ro, req, int, 最大值*/
    },
    "singleFileMaxSize": 1,
    /*ro, req, int, 单个人员图片数据文件大小范围, desc:单位: KB, range:[,], step:, unit:, unitType:*/
    "employeeNo": {
      /*ro, opt, object, 工号（人员ID）*/
      "@min": 1,
      /*ro, opt, int, 最小值*/
      "@max": 32
      /*ro, opt, int, 最大值*/
    },
    "FDID": "test",
    /*ro, opt, string, 人脸库ID*/
    "deleteFacePic": {
      /*ro, opt, object, 删除人脸图片*/
      "@opt": [true, false]
      /*ro, opt, array, 可选项, subType:bool*/
    },
    "picURL": {
      /*ro, opt, object, 人脸图片URL*/
      "@min": 1,
      /*ro, opt, int, 最小值*/
      "@max": 1
      /*ro, opt, int, 最大值*/
    }
  }
}
```

69.37.31 删除单一任务（人员图片异步下发）

Request URL

DELETE /ISAPI/AccessControl/UserPic/asyncImportDatasTasks/<taskID>?format=json

查询参数

参数名称	参数类型	描述
taskID	string	

请求报文

无

响应报文

```
{
  "statusCode": 1,
  /*ro, opt, int, 状态码, desc:无法用1表示时（1表示成功且无特殊状态）必须返回*/
  "statusString": "OK",
  /*ro, opt, string, 状态描述, range:[1,64], desc:无法用OK表示时（OK表示成功且无特殊状态）必须返回*/
  "subStatusCode": "ok",
  /*ro, opt, string, 子状态码, range:[1,64], desc:无法用ok表示（ok表示成功且无特殊状态）必须返回*/
  "errorCode": 1,
  /*ro, opt, int, 错误码, desc:当statusCode不为1时，与subStatusCode对应*/
  "errorMsg": "ok",
  /*ro, opt, string, 错误信息, desc:当statusCode不为1时，必须返回,解释信息在协议约束中，允许设备在后续的版本迭代中，进行优化丰富提升（不限制死）*/
}
```

69.37.32 获取全部任务状态（人员图片异步下发）

Request URL

GET /ISAPI/AccessControl/UserPic/asyncImportDatasTasks/status?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "TasksStatusList": [
    /*ro, req, array, 任务状态列表, subType:object*/
    {
      "AsyncImportDatasTask": {
        /*ro, opt, object, 人员图片异步下发任务*/
        "taskID": "test",
        /*ro, req, string, 下发任务ID, range:[1,64]*/
        "URL": "test",
        /*ro, opt, string, 数据文件的URL, range:[1,256]*/
        "status": 1,
        /*ro, req, enum, 任务执行状态, subType:int, [0#未执行,1#执行中,2#执行完成,3#未找到该任务]*/
        "totalNum": 1,
        /*ro, opt, int, 导入人员图片总数, desc:该节点依赖于status为1-执行完成*/
        "successNum": 1,
        /*ro, opt, int, 人员图片存储成功数目, desc:该节点依赖于status为1-执行完成*/
        "failedNum": 1
        /*ro, opt, int, 人员图片存储失败数目, desc:该节点依赖于status为1-执行完成*/
      }
    }
  ]
}
```

69.37.33 删除单一任务（人员数据异步下发）

Request URL

DELETE /ISAPI/AccessControl/UserInfo/asyncImportDatasTasks/<taskID>?format=json

查询参数

参数名称	参数类型	描述
taskID	string	--

请求报文

无

响应报文

```
{
  "statusCode": 1,
  /*ro, opt, int, 状态码, desc:无法用1表示时（1表示成功且无特殊状态）必须返回*/
  "statusString": "OK",
  /*ro, opt, string, 状态描述, range:[1,64], desc:无法用OK表示时（OK表示成功且无特殊状态）必须返回*/
  "subStatusCode": "ok",
  /*ro, opt, string, 子状态码, range:[1,64], desc:无法用ok表示（ok表示成功且无特殊状态）必须返回*/
  "errorCode": 1,
  /*ro, opt, int, 错误码, desc:当statusCode不为1时，与subStatusCode对应*/
  "errorMsg": "ok"
  /*ro, opt, string, 错误信息, desc:当statusCode不为1时，必须返回,解释信息在协议约束中，允许设备在后续的版本迭代中，进行优化丰富提升（不限制死）*/
}
```

69.37.34 获取单一任务状态（人员数据异步下发）

Request URL

GET /ISAPI/AccessControl/UserInfo/asyncImportDatasTasks/<taskID>/status?format=json

查询参数

参数名称	参数类型	描述
taskID	string	

请求报文

无

响应报文

```
{
  "AsyncImportDatasTask": {
    /*ro, req, object, 人员数据异步导入任务*/
    "taskID": "test",
    /*ro, req, string, 下发任务ID, range:[1,64]*/
    "URL": "test",
    /*ro, opt, string, 数据文件的URL, range:[1,256]*/
    "status": 0,
    /*ro, req, enum, 任务执行状态, subType:int, [0#未执行,1#权限生效中,2#权限持久化中（临时权限已生效）,3#权限持久化完成,4#未找到该任务]*/
    "totalNum": 1,
    /*ro, opt, int, 导入人员数据总数, desc:该节点依赖于status为3-权限持久化完成时返回*/
    "successNum": 1,
    /*ro, opt, int, 人员数据存储成功数目, desc:该节点依赖于status为3-权限持久化完成时返回*/
    "failedNum": 1
    /*ro, opt, int, 人员数据存储失败数目, desc:该节点依赖于status为3-权限持久化完成时返回*/
  }
}
```

69.37.35 查询卡信息

Request URL

POST /ISAPI/AccessControl/CardInfo/Search?format=json&security=<security>&iv=<iv>

查询参数

参数名称	参数类型	描述
security	string	没有security表示数据不加密；security=1,表示报文中的employeeNo、cardNo字段AES128加密；security=2,表示报文中的employeeNo、cardNo字段AES256加密；
iv	string	iv表示初始化向量，在security=1或者2时为必须参数；

请求报文

```
{
  "CardInfoSearchCond": {
    /*req, object, 卡查询条件*/
    "searchID": "test",
    /*req, string, 搜索记录唯一标识, desc:用来确认上层客户端是否为同一个(倘若为同一个,设备记录内存,下次搜索加快速度)*/
    "searchResultPosition": 0,
    /*req, int, 查询结果在结果列表中的起始位置,
    desc:当记录条数很多时,一次查询不能获取所有的记录,下一次查询时指定位置可以查询后面的记录。
    若设备支持的最大totalMatches为M个,但是当前设备已存储的totalMatches为N个（N<=M），则该字段的合法范围为0-N-1*/
    "maxResults": 30,
    /*req, int, 本次协议调用可获取的最大记录数, desc:如maxResults值大于设备能力集返回的范围,则设备按照能力集最大值返回,设备不进行报错*/
    "EmployeeNoList": [
      /*opt, array, 人员ID列表, subType:object*/
      {
        "employeeNo": "test"
        /*opt, string, 工号（人员ID）*/
      }
    ],
    "CardNoList": [
      /*opt, array, 卡号列表, subType:object, desc:与人员ID列表互斥,两者选一*/
      {
        "cardNo": "1234567890"
        /*opt, string, 卡号*/
      }
    ]
  }
}
```

响应报文

```
{
  "CardInfoSearch": {
    /*ro, opt, object, 卡查询信息*/
    "searchID": "test",
    /*ro, req, string, 搜索记录唯一标识*/
    "responseStatusStrg": "OK",
    /*ro, req, enum, 查询状态字符串描述, subType:string, [OK#查询结束,MORE#还有数据等待查询,NO_MATCH#没有匹配数据]*/
    "numOfMatches": 1,
    /*ro, req, int, 本次返回的记录条数*/
    "totalMatches": 1,
    /*ro, req, int, 符合条件的记录总条数*/
    "CardInfo": [
      /*ro, opt, array, 卡信息, subType:object*/
      {
        "employeeNo": "test",
        /*ro, req, string, 工号（人员ID）*/
        "cardNo": "test",
        /*ro, req, string, 卡号*/
        "cardType": "normalCard",
        /*ro, req, enum, 卡类型, [normalCard#普通卡,patrolCard#巡更卡,hijackCard#胁迫卡,superCard#超级卡,dismissingCard#解除卡,emergencyCard#应急管理卡], desc:（用于授权临时卡权限，本身不能开门）*/
        "leaderCard": "1,3,5"
        /*ro, opt, string, 是否有首次认证功能, desc:（表示该卡对于门1、门3、门5有首次认证功能）*/
      }
    ]
  }
}
```

69.37.36 添加指纹信息

Request URL

POST /ISAPI/AccessControl/FingerPrintDownload?format=json&security=<security> &iv=<iv>

查询参数

参数名称	参数类型	描述
security	string	没有security表示数据不加密； security=1,表示报文中的employeeNo、fingerData字段AES128加密； security=2,表示报文中的employeeNo、fingerData字段AES256加密； fingerData字段是否支持加密 见门禁总能力的isSupportFingerDataEncryption字段
iv	string	iv表示初始化向量，在security=1或者2时为必须参数;

请求报文

```
{
  "FingerPrintCfg": {
    /*opt, object, 指纹配置*/
    "employeeNo": "test",
    /*req, string, 指纹关联的工号（人员ID）， range:[1,32]*/
    "enableCardReader": [1, 3, 5],
    /*req, array, 需要下发指纹的读卡器, subType:int*/
    "fingerPrintID": 1,
    /*req, int, 手指编号, range:[1,10]*/
    "fingerType": "normalFP",
    /*req, enum, 指纹类型, subType:string, [normalFP#普通指纹,hijackFP#胁迫指纹,patrolFP#巡更指纹,superFP#超级指纹,dismissingFP#解除指纹]*/
    "fingerData": "test",
    /*req, string, 指纹数据, desc:（需要Base64编码）， 如果需要加密，则指纹数据仅做一次base64处理，直接进行aes加密*/
    "leaderFP": [1, 3, 5],
    /*opt, array, 是否有首次认证功能, subType:int, desc:[1,3,5]代表指纹对于门1、门3、门5有首次认证功能*/
    "checkEmployeeNo": true
    /*opt, bool, 设备是否进行工号（人员ID）存在校验, desc:false-不校验, true-校验（如果不配置该字段，则设备默认进行工号（人员ID）存在校验）（该字段配置为false时，设备不进行工号（人员ID）存在校验（可加快设备下发卡的速度）；该字段配置为true或不配置该字段时，设备进行工号（人员ID）存在校验，如不对处理速度有强烈要求，建议使用该方式）*/
  }
}
```

响应报文

```
{
  "statusCode": 1,
  /*ro, opt, int, 状态码, desc:无法用1表示时（1表示成功且无特殊状态）必须返回*/
  "statusString": "OK",
  /*ro, opt, string, 状态描述, range:[1,64], desc:无法用OK表示时（OK表示成功且无特殊状态）必须返回*/
  "subStatusCode": "ok",
  /*ro, opt, string, 子状态码, range:[1,64], desc:无法用ok表示（ok表示成功且无特殊状态）必须返回*/
  "errorCode": 1,
  /*ro, opt, int, 错误码, desc:当statusCode不为1时，与subStatusCode对应*/
  "errorMsg": "ok"
  /*ro, opt, string, 错误信息, desc:当statusCode不为1时，必须返回,解释信息在协议约束中，允许设备在后续的版本迭代中，进行优化丰富提升（不限制死）*/
}
```

69.37.37 修改指纹信息

Request URL

POST /ISAPI/AccessControl/FingerPrintModify?format=json&security=<security>&iv=<iv>

查询参数

参数名称	参数类型	描述
security	string	没有security表示数据不加密；security=1,表示报文中的employeeNo字段AES128加密；security=2,表示报文中的employeeNo字段AES256加密
iv	string	iv表示初始化向量，在security=1或者2时为必须参数;

请求报文

```
{
  "FingerPrintModify": {
    /*opt, object, 指纹修改*/
    "employeeNo": "test",
    /*req, string, 指纹关联的工号（人员ID），range:[1,32]*/
    "cardReaderNo": 1,
    /*req, int, 读卡器编号*/
    "fingerPrintID": 1,
    /*req, int, 手指编号, range:[1,10]*/
    "fingerType": "normalFP",
    /*req, enum, 指纹类型, subType:string, [normalFP#普通指纹,hijackFP#胁迫指纹,patrolFP#巡更指纹,superFP#超级指纹,dismissingFP#解除指纹]*/
    "leaderFP": [1, 3, 5]
    /*opt, array, 是否有首次认证功能, subType:int, desc:（[1,3,5]代表指纹对于门1、门3、门5有首次认证功能）（该字段不添加，则保持原首次认证功能不变）*/
  }
}
```

响应报文

```
{
  "statusCode": 1,
  /*ro, opt, int, 状态码, desc:无法用1表示时（1表示成功且无特殊状态）必须返回*/
  "statusString": "OK",
  /*ro, opt, string, 状态描述, range:[1,64], desc:无法用OK表示时（OK表示成功且无特殊状态）必须返回*/
  "subStatusCode": "ok",
  /*ro, opt, string, 子状态码, range:[1,64], desc:无法用ok表示（ok表示成功且无特殊状态）必须返回*/
  "errorCode": 1,
  /*ro, opt, int, 错误码, desc:当statusCode不为1时，与subStatusCode对应*/
  "errorMsg": "ok"
  /*ro, opt, string, 错误信息, desc:当statusCode不为1时，必须返回,解释信息在协议约束中，允许设备在后续的版本迭代中，进行优化丰富提升（不限制死）*/
}
```

69.37.38 查询指纹信息

Request URL

POST /ISAPI/AccessControl/FingerPrintUpload?format=json&security=<security>&iv=<iv>

查询参数

参数名称	参数类型	描述
security	string	没有security表示数据不加密； security=1,表示报文中的employeeNo、fingerData字段AES128加密； security=2,表示报文中的employeeNo、fingerData字段AES256加密； fingerData字段是否支持加密见门禁总能力的isSupportFingerDataEncryption字段
iv	string	iv表示初始化向量，在security=1或者2时为必须参数;

请求报文

```
{
  "FingerPrintCond": {
    /*opt, object, 搜索条件*/
    "searchID": "test",
    /*req, string, 搜索记录唯一标识, range:[1,32], desc:用来确认上层客户端是否为同一个(倘若是一个,设备记录内存,下次搜索加快速度)*/
    "employeeNo": "test",
    /*req, string, 指纹关联的工号(人员ID), range:[1,32]*/
    "cardReaderNo": 1,
    /*opt, int, 读卡器编号*/
    "fingerPrintID": 1
    /*opt, int, 手指编号, range:[1,10]*/
  }
}
```

响应报文

```
{
  "FingerPrintInfo": {
    /*ro, opt, object, 指纹信息*/
    "searchID": "test",
    /*ro, req, string, 搜索记录唯一标识*/
    "status": "OK",
    /*ro, req, enum, 状态, subType:string, [OK#存在指纹,NoFP#不存在指纹]*/
    "FingerPrintList": [
      /*ro, opt, array, 指纹列表, subType:object*/
      {
        "cardReaderNo": 1,
        /*ro, req, int, 读卡器编号, range:[1,10]*/
        "fingerPrintID": 1,
        /*ro, req, int, 手指编号*/
        "fingerType": "normalFP",
        /*ro, req, enum, 指纹类型, subType:string, [normalFP#普通指纹,hiJackFP#胁迫指纹,patrolFP#巡更指纹,superFP#超级指纹,dismissingFP#解除指纹]*/
        "fingerData": "test",
        /*ro, req, string, 指纹数据*/
        "leaderFP": [1, 3, 5]
        /*ro, opt, array, 是否有首次认证功能, subType:int, desc: ([1,3,5]代表指纹对于门1、门3、门5有首次认证功能)*/
      }
    ]
  }
}
```

69.37.39 删除指纹信息

Request URL

PUT /ISAPI/AccessControl/FingerPrint/Delete?format=json&security= <security> &iv= <iv>

查询参数

参数名称	参数类型	描述
security	string	没有security表示数据不加密； security=1,表示报文中的employeeNo字段AES128加密； security=2,表示报文中的employeeNo字段AES256加密
iv	string	iv表示初始化向量，在security=1或者2时为必须参数;

请求报文

```
{
  "FingerPrintDelete": {
    /*req, object, 删除配置*/
    "mode": "byEmployeeNo",
    /*req, enum, 删除方式, subType:string, [byEmployeeNo#按工号（人员ID）删除,byCardReader#按读卡器删除], desc:（byEmployeeNo-按工号（人员ID）删除*/
    "EmployeeNoDetail": {
      /*opt, object, 按工号（人员ID）删除信息, dep:and,{$.FingerPrintDelete.mode,eq,byEmployeeNo}, desc:（mode为byEmployeeNo有效）*/
      "employeeNo": "test",
      /*opt, string, 指纹关联的工号（人员ID）*/
      "enableCardReader": [1, 3, 5],
      /*opt, array, 删除指纹的读卡器, subType:int*/
      "fingerPrintID": [1, 3, 5]
      /*opt, array, 需要删除的手指编号, subType:int*/
    },
    "CardReaderDetail": {
      /*opt, object, 按读卡器删除信息, dep:and,{$.FingerPrintDelete.mode,eq,byCardReader}, desc:（mode为byCardReader有效）*/
      "cardReaderNo": 1,
      /*opt, int, 指纹读卡器编号*/
      "clearAllCard": false,
      /*opt, bool, 是否删除所有卡的指纹信息, desc:false-否（按工号删除）*/
      "employeeNo": "test"
      /*opt, string, 指纹关联的工号（人员ID）*/
    }
  }
}
```

响应报文

```
{
  "statusCode": 1,
  /*ro, opt, int, 状态码, desc:无法用1表示时（1表示成功且无特殊状态）必须返回*/
  "statusString": "OK",
  /*ro, opt, string, 状态描述, range:[1,64], desc:无法用OK表示时（OK表示成功且无特殊状态）必须返回*/
  "subStatusCode": "ok",
  /*ro, opt, string, 子状态码, range:[1,64], desc:无法用ok表示（ok表示成功且无特殊状态）必须返回*/
  "errorCode": 1,
  /*ro, opt, int, 错误码, desc:当statusCode不为1时，与subStatusCode对应*/
  "errorMsg": "ok"
  /*ro, opt, string, 错误信息, desc:当statusCode不为1时，必须返回,解释信息在协议约束中，允许设备在后续的版本迭代中，进行优化丰富提升（不限制死）*/
}
```

69.37.40 获取全部任务状态（人员数据异步下发）

Request URL

GET /ISAPI/AccessControl/UserInfo/asyncImportDatasTasks/status?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "TasksStatusList": [
    /*ro, req, array, 任务状态列表, subType:object*/
    {
      "AsyncImportDatasTask": {
        /*ro, opt, object, 人员数据异步下发任务*/
        "taskId": "test",
        /*ro, req, string, 下发任务ID, range:[1,64]*/
        "URL": "test",
        /*ro, opt, string, 数据文件的URL, range:[1,256]*/
        "status": 0,
        /*ro, req, enum, 任务执行状态, subType:int, [0#未执行,1#权限生效中,2#权限持久化中（临时权限已生效）,3#权限持久化完成,4#未找到该任务]*/
        "totalNum": 1,
        /*ro, opt, int, 导入人员数据总数, desc:该节点依赖于status为3-权限持久化完成时返回*/
        "successNum": 1,
        /*ro, opt, int, 人员数据存储成功数目, desc:该节点依赖于status为3-权限持久化完成时返回*/
        "failedNum": 1,
        /*ro, opt, int, 人员数据存储失败数目, desc:该节点依赖于status为3-权限持久化完成时返回*/
        "taskMissingReason": "deviceShutDown"
        /*ro, opt, enum, 任务丢失的原因, subType:string, [deviceShutDown#设备断电,meetingAPPExit#会议应用退出（信发门牌使用）], dep:and,
        {$.TasksStatusList[*].AsyncImportDatasTask.status,eq,4}, desc:该节点依赖于status为4-未找到该任务时返回*/
      }
    }
  ]
}
```

69.37.41 查询虹膜信息

Request URL

POST /ISAPI/AccessControl/IrisInfo/search?format=json&security=<security>&iv=<iv>

查询参数

参数名称	参数类型	描述
security	string	没有security表示数据不加密； security=1，表示报文中的employeeNo,irisModelData字段AES128加密； security=2，表示报文中的employeeNo,irisModelData字段AES256加密
iv	string	iv表示初始化向量，在security=1或者2时为必须参数

请求报文

```
{
  "IrisInfoSearchCond": {
    /*req, object, 搜索条件*/
    "searchID": "test",
    /*req, string, 搜索记录唯一标识, range:[,], desc:用来确认上层客户端是否为同一个(倘若是一个,设备记录内存,下次搜索加快速度)*/
    "searchResultPosition": 0,
    /*req, int, 查询结果在结果列表中的起始位置, range:[0,65535], step:, unit:, unitType:,
    desc:当记录条数很多时,一次查询不能获取所有的记录,下一次查询时指定位置可以查询后面的记录
    //(若设备支持的最大totalMatches为M个,但是当前设备已存储的totalMatches为N个(N<=M),则该字段的合法范围为0~N-1)*/
    "maxResults": 30,
    /*req, int, 本次协议调用可获取的最大记录数, range:[,], step:, unit:, unitType:, desc:如maxResults值大于设备能力集返回的范围,则设备按照能力集最大值
    返回,设备不进行报错*/
    "EmployeeNoList": [
      /*opt, array, 工号列表, subType:object, range:[,], desc:当该字段为空或不存在时,表示查询所有人员*/
      {
        "employeeNo": "test"
      }
    ]
  }
}
```

响应报文

```
{
  "IrisInfoSearch": {
    /*ro, opt, object, 搜索结果*/
    "searchID": "test",
    /*ro, req, string, 搜索记录唯一标识, range:[,]*/
    "responseStatusStrg": "OK",
    /*ro, req, enum, 查询状态字符串描述, subType:string, [OK#查询结束,MORE#还有数据等待查询,NO MATCH#没有匹配数据]*/
    "numOfMatches": 1,
    /*ro, req, int, 本次返回的记录条数, range:[,], step:, unit:, unitType:*/
    "totalMatches": 1,
    /*ro, req, int, 符合条件的记录总条数, range:[,], step:, unit:, unitType:*/
    "IrisInfo": [
      /*ro, req, array, 虹膜信息, subType:object, range:[,]*/
      {
        "employeeNo": "test",
        /*ro, req, string, 工号, range:[,]*/
        "IrisData": [
          /*ro, req, array, 虹膜数据, subType:object, range:[,]*/
          {
            "id": 1,
            /*ro, req, int, 虹膜数据序号, range:[,], step:, unit:, unitType:, desc:仅支持1和2*/
            "irisModelData": "test",
            /*ro, opt, string, 虹膜模板数据, range:[,],
            desc:必须base处理,防止模型数据出现'\0'字符。
            加密时进行两次base64处理后,再做aes加密*/
            "irisPicData": "test"
            /*ro, opt, string, 虹膜图片url数据, range:[,], desc:由于是查询接口可能有多个人员,采用表单传输二进制不便于处理,因此此处必须为url*/
          }
        ]
      }
    ]
  }
}
```

69.37.42 查询虹膜数量

Request URL

GET /ISAPI/AccessControl/IrisInfo/count?format=json&employeeNo=<employeeNo>&security=<security>&iv=<iv>

查询参数

参数名称	参数类型	描述
employeeNo	string	--
security	string	没有security表示数据不加密； security=1，表示URL后的employeeNo字段AES128加密； security=2，表示URL后的employeeNo字段AES256加密
iv	string	iv表示初始化向量，在security=1或者2时为必须参数

请求报文

无

响应报文

```
{
  "IrisInfoCount": {
    /*ro, opt, object, 虹膜信息数量*/
    "IrisNumber": 100
  }
  /*ro, req, int, 虹膜数量, desc:请求参数没有employeeNo时表示当前设备内已有的虹膜数量,有employeeNo时表示指定工号下已有的虹膜数量*/
}
```

69.38 健康码管理

69.38.1 配置健康码显示参数

Request URL

PUT /ISAPI/AccessControl/showHealthCodeCfg?format=json

查询参数

无

请求报文

```
{
  "showHealthCode": true,
  /*req, bool, 是否显示健康码, desc:true-显示, false-不显示*/
  "showNADCode": true,
  /*opt, bool, 是否显示核酸检测结果, desc:true-显示, false-不显示*/
  "showTravelCode": true,
  /*opt, bool, 是否显示行程编码, desc:true-显示, false-不显示*/
  "showPromptMessage": true,
  /*opt, bool, 是否显示自定义提示信息, desc:该字段默认为false*/
  "promptMessage": "text",
  /*opt, string, 自定义提示信息, range:[1,108], dep:and,{$.showPromptMessage,eq,true}, desc:在获取健康码信息失败时UI显示自定义提示信息文本内容*/
  "showANTInfo": true,
  /*opt, bool, 是否显示抗原信息, desc:true-显示, false-不显示*/
  "showVaccineInfo": true,
  /*opt, bool, 是否显示疫苗信息, desc:true-显示, false-不显示*/
  "selfDefineInfo": "test"
  /*opt, string, 自定义信息, range:[0,128],
  desc:自定义信息, 可填写为场所信息、省份信息等自定义内容
  若/ISAPI/AccessControl/healthInfoResult?format=json&security=<security>&iv=<iv>接口中下发了该信息, 则以下发的为准; 若该接口未返回自定义信息, 则以本接口
  配置的信息为准*/
}
```

响应报文

```
{
  "statusCode": 1,
  /*ro, opt, int, 状态码, desc:无法用1表示时（1表示成功且无特殊状态）必须返回*/
  "statusString": "OK",
  /*ro, opt, string, 状态描述, range:[1,64], desc:无法用OK表示时（OK表示成功且无特殊状态）必须返回*/
  "subStatusCode": "ok",
  /*ro, opt, string, 子状态码, range:[1,64], desc:无法用ok表示（ok表示成功且无特殊状态）必须返回*/
  "errorCode": 1,
  /*ro, opt, int, 错误码, desc:当statusCode不为1时，与subStatusCode对应*/
  "errorMsg": "ok"
  /*ro, opt, string, 错误信息, desc:当statusCode不为1时，必须返回,解释信息在协议约束中，允许设备在后续的版本迭代中，进行优化丰富提升（不限制死）*/
}
```

69.38.2 获取健康码显示参数

Request URL

GET /ISAPI/AccessControl/showHealthCodeCfg?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "showHealthCode": true,
  /*ro, req, bool, 是否显示健康码, desc:true-显示, false-不显示*/
  "showNADCode": true,
  /*ro, opt, bool, 是否显示核酸检测结果, desc:true-显示, false-不显示*/
  "showTravelCode": true,
  /*ro, opt, bool, 是否显示行程编码, desc:true-显示, false-不显示*/
  "showPromptMessage": true,
  /*ro, opt, bool, 是否显示自定义提示信息, desc:该字段默认为false*/
  "promptMessage": "text",
  /*ro, opt, string, 自定义提示信息, range:[1,108], dep:and,{$.showPromptMessage,eq,true}, desc:在获取健康码信息失败时UI显示自定义提示信息文本内容*/
  "showANTIInfo": true,
  /*ro, opt, bool, 是否显示抗原信息, desc:true-显示, false-不显示*/
  "showVaccineInfo": true,
  /*ro, opt, bool, 是否显示疫苗信息, desc:true-显示, false-不显示*/
  "selfDefineInfo": "test"
  /*ro, opt, string, 自定义信息, range:[0,128],
  desc:自定义信息, 可填写为场所信息、省份信息等自定义内容
  若/ISAPI/AccessControl/healthInfoResult?format=json&security=<security>&iv=<iv>接口中下发了该信息, 则以下发的为准; 若该接口未返回自定义信息, 则以本接口
  配置的信息为准*/
}
```

69.38.3 获取健康码显示参数配置能力

Request URL

GET /ISAPI/AccessControl/showHealthCodeCfg/capabilities?format=json

查询参数

无

请求报文

无

响应报文

```

{
  "showHealthCode": {
    /*ro, req, object, 是否显示健康码信息, desc:true-显示, false-不显示*/
    "@opt": [true, false]
    /*ro, req, array, 取值范围, subType:bool*/
  },
  "showNADCode": {
    /*ro, opt, object, 是否显示核酸检测结果, desc:true-显示, false-不显示*/
    "@opt": [true, false]
    /*ro, req, array, 取值范围, subType:bool*/
  },
  "showTravelCode": {
    /*ro, opt, object, 是否显示行程编码, desc:true-显示, false-不显示*/
    "@opt": [true, false]
    /*ro, req, array, 取值范围, subType:bool*/
  },
  "showPromptMessage": {
    /*ro, opt, object, 是否显示自定义提示信息, desc:该字段默认为true*/
    "@opt": [true, false]
    /*ro, req, array, 取值范围, subType:bool*/
  },
  "promptMessage": {
    /*ro, opt, object, 自定义提示信息*/
    "@min": 1,
    /*ro, req, int, 最小值, step:1*/
    "@max": 108
    /*ro, req, int, 最大值, step:1*/
  },
  "showANTInfo": {
    /*ro, opt, object, 是否显示抗原信息, desc:true-显示, false-不显示*/
    "@opt": [true, false]
    /*ro, req, array, 取值范围, subType:bool*/
  },
  "showVaccineInfo": {
    /*ro, opt, object, 是否显示疫苗信息, desc:true-显示, false-不显示*/
    "@opt": [true, false]
    /*ro, req, array, 取值范围, subType:bool*/
  },
  "selfDefineInfo": {
    /*ro, opt, object, 自定义信息*/
    "@min": 0,
    /*ro, req, int, 取值范围*/
    "@max": 128
    /*ro, req, int, 取值范围*/
  }
}

```

69.38.4 配置健康码参数

Request URL

PUT /ISAPI/AccessControl/healthCodeCfg?format=json

查询参数

无

请求报文

```

{
  "enabled": true,
  /*req, bool, 使能*/
  "serverAddress": "test",
  /*opt, string, 健康码服务器地址, range:[1,128], desc:配置该地址表示上层平台采用私有协议对接健康码查询业务*/
  "lockEnabled": true,
  /*opt, bool, 锁定使能, desc:当检测到某个人健康信息异常时, 设备锁定, 不允许后面的人继续通行; 待设备解锁后, 才能继续通行; 默认false-不使能*/
  "unlockUserType": "admin",
  /*opt, enum, 解锁权限人员类型, subType:string, [admin#管理员,all#所有用户（可触摸设备屏幕的任意人员）], desc:当对应拥有解锁权限的人员对设备进行认证解锁
后, 设备从锁定状态变成解锁状态后, 才允许后面的人继续通行*/
  "lockRule": {
    /*opt, object, 锁定规则*/
    "healthCode": {
      /*opt, object, 健康码锁定规则*/
      "abnormalLock": true,
      /*opt, bool, 健康码异常时锁定, desc:健康码异常包括: 未申领, 黄码, 红码, 无此人员*/
      "searchFailLock": true
      /*opt, bool, 健康码查询失败时锁定, desc:健康码查询失败包括: 其他错误信息（如接口异常导致查询失败）, 查询健康码超时*/
    },
    "NADCode": {
      /*opt, object, 核酸检测锁定规则*/
      "abnormalLock": true,
      /*opt, bool, 核酸检测异常时锁定, desc:核酸检测异常包括: 未查询到核酸检测结果, 核酸检测阳性（代表确诊）, 检测结果待复核*/
      "searchFailLock": true,
      /*opt, bool, 核酸检测查询失败时锁定, desc:核酸检测查询失败包括: 查询核酸结果失败*/
      "timeoutLock": 1
      /*opt, enum, 核酸检测有效期已过锁定, subType:int, [1#不锁定,2#超过24小时锁定,3#超过48小时锁定]*/
    },
    "ANTCode": {
      /*opt, object, 抗原检测锁定规则*/
      "abnormalLock": true,
      /*opt, bool, 抗原检测异常时锁定, desc:抗原检测异常包括: 未查询到抗原检测结果, 抗原检测阳性（代表确诊）, 抗原检测无效*/
      "searchFailLock": true,
      /*opt, bool, 抗原检测查询失败时锁定, desc:抗原检测查询失败包括: 查询抗原结果失败*/
      "timeoutLock": 1
      /*opt, enum, 抗原检测有效期已过锁定, subType:int, [1#不锁定,2#超过24小时锁定,3#超过48小时锁定]*/
    },
    "travelCode": {
      /*opt, object, 行程编码锁定规则*/
      "epidemicAreaLock": true,
      /*opt, bool, 行程编码到过疫区时锁定, desc:行程编码到过疫区包括14天内到过疫区*/
      "leaveLocalLock": true,
      /*opt, bool, 行程编码离开当地时锁定, desc:行程编码离开当地包括: 14天内离开过当地*/
      "searchFailLock": true
      /*opt, bool, 行程编码查询失败时锁定, desc:行程编码查询失败包括: 查询行程信息失败*/
    }
  },
  "protocolSwitchEnabled": false
  /*opt, bool, 协议切换使能, dep:or,{enabled,eq,true}, desc:1、默认为false。2、当为true时, 表示设备使用三方（对外）协议处理健康码查询业务。*/
}

```

响应报文

```

{
  "statusCode": 1,
  /*ro, opt, int, 状态码, desc:无法用1表示时（1表示成功且无特殊状态）必须返回*/
  "statusString": "OK",
  /*ro, opt, string, 状态描述, range:[1,64], desc:无法用OK表示时（OK表示成功且无特殊状态）必须返回*/
  "subStatusCode": "ok",
  /*ro, opt, string, 子状态码, range:[1,64], desc:无法用ok表示（ok表示成功且无特殊状态）必须返回*/
  "errorCode": 1,
  /*ro, opt, int, 错误码, desc:当statusCode不为1时, 与subStatusCode对应*/
  "errorMsg": "ok"
  /*ro, opt, string, 错误信息, desc:当statusCode不为1时, 必须返回,解释信息在协议约束中, 允许设备在后续的版本迭代中, 进行优化丰富提升（不限制死）*/
}

```

69.38.5 获取健康码参数

Request URL

GET /ISAPI/AccessControl/healthCodeCfg?format=json

查询参数

无

请求报文

无

响应报文

```

{
  "enabled": true,
  /*ro, req, bool, 使能*/
  "serverAddress": "test",
  /*ro, opt, string, 健康码服务器地址, range:[1,128], desc:配置该地址表示上层平台采用私有协议对接健康码查询业务*/
  "lockEnabled": true,
  /*ro, opt, bool, 锁定使能, desc:当检测到某个人健康信息异常时, 设备锁定, 不允许后面的人继续通行; 待设备解锁后, 才能继续通行; 默认false-不使能*/
  "unlockUserType": "admin",
  /*ro, opt, enum, 解锁权限人员类型, subType:string, [admin#管理员,all#所有用户（可触摸设备屏幕的任意人员）], desc:当对应拥有解锁权限的人员对设备进行认证
  解锁后, 设备从锁定状态变成解锁状态后, 才允许后面的人继续通行*/
  "lockRule": {
    /*ro, opt, object, 锁定规则*/
    "healthCode": {
      /*ro, opt, object, 健康码锁定规则*/
      "abnormalLock": true,
      /*ro, opt, bool, 健康码异常时锁定, desc:健康码异常包括: 未申领, 黄码, 红码, 无此人员*/
      "searchFailLock": true
      /*ro, opt, bool, 健康码查询失败时锁定, desc:健康码查询失败包括: 其他错误信息（如接口异常导致查询失败）, 查询健康码超时*/
    },
    "NADCode": {
      /*ro, opt, object, 核酸检测锁定规则*/
      "abnormalLock": true,
      /*ro, opt, bool, 核酸检测异常时锁定, desc:核酸检测异常包括: 未查询到核酸检测结果, 核酸检测阳性（代表确诊）, 检测结果待复核*/
      "searchFailLock": true,
      /*ro, opt, bool, 核酸检测查询失败时锁定, desc:核酸检测查询失败包括: 查询核酸结果失败*/
      "timeoutLock": 1
      /*ro, opt, enum, 核酸检测有效期已过锁定, subType:int, [1#不锁定,2#超过24小时锁定,3#超过48小时锁定]*/
    },
    "ANTCode": {
      /*ro, opt, object, 抗原检测锁定规则*/
      "abnormalLock": true,
      /*ro, opt, bool, 抗原检测异常时锁定, desc:抗原检测异常包括: 未查询到抗原检测结果, 抗原检测阳性（代表确诊）, 抗原检测无效*/
      "searchFailLock": true,
      /*ro, opt, bool, 抗原检测查询失败时锁定, desc:抗原检测查询失败包括: 查询抗原结果失败*/
      "timeoutLock": 1
      /*ro, opt, enum, 抗原检测有效期已过锁定, subType:int, [1#不锁定,2#超过24小时锁定,3#超过48小时锁定]*/
    },
    "travelCode": {
      /*ro, opt, object, 行程编码锁定规则*/
      "epidemicAreaLock": true,
      /*ro, opt, bool, 行程编码到过疫区时锁定, desc:行程编码到过疫区包括14天内到过疫区*/
      "leaveLocalLock": true,
      /*ro, opt, bool, 行程编码离开当地时锁定, desc:行程编码离开当地包括: 14天内离开过当地*/
      "searchFailLock": true
      /*ro, opt, bool, 行程编码查询失败时锁定, desc:行程编码查询失败包括: 查询行程信息失败*/
    }
  },
  "protocolSwitchEnabled": false
  /*ro, opt, bool, 协议切换使能, depton,{renabled,eq,true}, desc:1-默认为false. 2- 当为true时, 表示设备使用三方（对外）协议处理健康码查询业务。*/
}

```

69.38.6 获取健康码参数配置能力

Request URL

GET /ISAPI/AccessControl/healthCodeCfg/capabilities?format=json

查询参数

无

请求报文

无

响应报文

```

{
  "enabled": {
    /*ro, req, object, 使能, desc:true-开启, false-不开启*/
    "@opt": [true, false]
    /*ro, opt, array, 可选项, subType:bool*/
  },
  "serverAddress": {
    /*ro, opt, object, 健康码服务器地址, desc:只是一个字符串地址, 不支持单独的ip和端口配置*/
    "@min": 1,
    /*ro, opt, int, 健康码服务器地址长度最小值*/
    "@max": 128
    /*ro, opt, int, 健康码服务器地址长度最大值*/
  },
  "lockEnabled": {
    /*ro, opt, object, 锁定使能, desc:当检测到某个人健康信息异常时, 设备锁定, 不允许后面的人继续通行; 待设备解锁后, 才能继续通行; 默认false-不使能*/
    "@opt": [true, false]
    /*ro, opt, array, 可选项, subType:bool*/
  },
  "unlockUserType": {
    /*ro, opt, object, 解锁权限人员类型, desc:当对应拥有解锁权限的人员对设备进行认证解锁后, 设备从锁定状态变成解锁状态后, 才允许后面的人继续通行*/
  }
}

```

```

    "@opt": ["admin", "all"]
    /*ro, opt, array, 可选项, subType:string*/
  },
  "lockRule": {
    /*ro, opt, object, 锁定规则*/
    "healthCode": {
      /*ro, opt, object, 健康码锁定规则*/
      "abnormalLock": {
        /*ro, opt, object, 健康码异常时锁定, desc:健康码异常包括: 未申领, 黄码, 红码, 无此人员*/
        "@opt": [true, false]
        /*ro, opt, array, 可选项, subType:bool*/
      },
      "searchFailLock": {
        /*ro, opt, object, 健康码查询失败时锁定, desc:健康码查询失败包括: 其他错误信息(如接口异常导致查询失败), 查询健康码超时*/
        "@opt": [true, false]
        /*ro, opt, array, 可选项, subType:bool*/
      }
    }
  },
  "NADCode": {
    /*ro, opt, object, 核酸检测锁定规则*/
    "abnormalLock": {
      /*ro, opt, object, 核酸检测异常时锁定, desc:核酸检测异常包括: 未查询到核酸检测结果, 核酸检测阳性(代表确诊), 检测结果待复核*/
      "@opt": [true, false]
      /*ro, opt, array, 可选项, subType:bool*/
    },
    "searchFailLock": {
      /*ro, opt, object, 核酸检测查询失败时锁定, desc:核酸检测查询失败包括: 查询核酸结果失败*/
      "@opt": [true, false]
      /*ro, opt, array, 可选项, subType:bool*/
    },
    "timeoutLock": {
      /*ro, opt, object, 核酸检测有效期已过锁定, desc:1#不锁定,2#超过24小时锁定,3#超过48小时锁定*/
      "@opt": [1, 2, 3]
      /*ro, opt, array, 可选项, subType:int*/
    }
  },
  "ANTCode": {
    /*ro, opt, object, 抗原检测锁定规则*/
    "abnormalLock": {
      /*ro, opt, object, 抗原检测异常时锁定, desc:抗原检测异常包括: 未查询到抗原检测结果, 抗原检测阳性(代表确诊), 抗原检测无效*/
      "@opt": [true, false]
      /*ro, opt, array, 可选项, subType:bool*/
    },
    "searchFailLock": {
      /*ro, opt, object, 抗原检测查询失败时锁定, desc:抗原检测查询失败包括: 查询抗原结果失败*/
      "@opt": [true, false]
      /*ro, opt, array, 可选项, subType:bool*/
    },
    "timeoutLock": {
      /*ro, opt, object, 抗原检测有效期已过锁定, desc:1#不锁定,2#超过24小时锁定,3#超过48小时锁定*/
      "@opt": [1, 2, 3]
      /*ro, opt, array, 可选项, subType:int*/
    }
  },
  "travelCode": {
    /*ro, opt, object, 行程编码锁定规则*/
    "epidemicAreaLock": {
      /*ro, opt, object, 行程编码到过疫区时锁定, desc:行程编码到过疫区包括14天内到过疫区*/
      "@opt": [true, false]
      /*ro, opt, array, 可选项, subType:bool*/
    },
    "leaveLocalLock": {
      /*ro, opt, object, 行程编码离开当地时锁定, desc:行程编码离开当地包括: 14天内离开过当地*/
      "@opt": [true, false]
      /*ro, opt, array, 可选项, subType:bool*/
    },
    "searchFailLock": {
      /*ro, opt, object, 行程编码查询失败时锁定, desc:行程编码查询失败包括: 查询行程信息失败*/
      "@opt": [true, false]
      /*ro, opt, array, 可选项, subType:bool*/
    }
  }
},
"protocolSwitchEnabled": {
  /*ro, opt, object, 协议切换使能*/
  "@opt": [true, false]
  /*ro, req, array, 可选项, subType:bool*/
}
}
}

```

69.39 设备内反潜回

69.39.1 配置读卡器反潜回参数

Request URL

PUT /ISAPI/AccessControl/CardReaderAntiSneakCfg/<cardReaderID>?format=json

查询参数

参数名称	参数类型	描述
cardReaderID	string	读卡器编号ID

请求报文

```
{
  "CardReaderAntiSneakCfg": {
    /*req, object, 读卡器反潜回参数*/
    "enable": true,
    /*req, bool, 是否加入反潜回路路径, desc:true-加入, false-不加入*/
    "followUpCardReader": [2, 3, 4]
    /*opt, array, 后续读卡器编号, subType:int, desc:[2,3,4]代表该读卡器后续可刷读卡器2或读卡器3或读卡器4*/
  }
}
```

响应报文

```
{
  "requestURL": "test",
  /*ro, opt, string, URI*/
  "statusCode": 1,
  /*ro, opt, int, 状态码, desc:无法用1表示时（1表示成功且无特殊状态）必须返回*/
  "statusString": "OK",
  /*ro, opt, string, 状态描述, range:[1,64], desc:无法用OK表示时（OK表示成功且无特殊状态）必须返回*/
  "subStatusCode": "ok",
  /*ro, opt, string, 子状态码, range:[1,64], desc:无法用ok表示（ok表示成功且无特殊状态）必须返回*/
  "errorCode": 1,
  /*ro, opt, int, 错误码, desc:当statusCode不为1时，与subStatusCode对应*/
  "errorMsg": "ok"
  /*ro, opt, string, 错误信息, desc:当statusCode不为1时，解释信息在协议约束中，允许设备在后续的版本迭代中，进行优化丰富提升（不限制死）*/
}
```

69.39.2 获取读卡器反潜回参数配置能力

Request URL

GET /ISAPI/AccessControl/CardReaderAntiSneakCfg/capabilities?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "CardReaderAntiSneakCfg": {
    /*ro, req, object, 读卡器反潜回能力*/
    "cardReaderNo": {
      /*ro, opt, object, 读卡器编号*/
      "@min": 1,
      /*ro, opt, int, 最小值*/
      "@max": 512,
      /*ro, opt, int, 最大值*/
      "@opt": [1, 4]
      /*ro, opt, array, 枚举值范围, subType:int, desc:闸机产品仅支持1和4，不需要返回min和max属性*/
    },
    "enable": "true,false",
    /*ro, req, string, 是否加入反潜回路路径*/
    "followUpCardReader": {
      /*ro, opt, object, 后续读卡器编号*/
      "@min": 1,
      /*ro, opt, int, 最小值*/
      "@max": 512,
      /*ro, opt, int, 最大值*/
      "@opt": [1, 4]
      /*ro, opt, array, 枚举值范围, subType:int, desc:闸机产品仅支持1和4，不需要返回min和max属性*/
    }
  }
}
```

69.39.3 获取清空反潜回参数能力

Request URL

GET /ISAPI/AccessControl/ClearAntiSneakCfg/capabilities?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "ClearAntiSneakCfg": {
    /*ro, req, object, 反潜回能力*/
    "ClearFlags": {
      /*ro, req, object, 清空反潜回配置*/
      "antiSneak": "true,false"
      /*ro, req, string, 是否清空反潜回*/
    }
  }
}
```

69.39.4 获取反潜回参数配置能力

Request URL

GET /ISAPI/AccessControl/AntiSneakCfg/capabilities?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "AntiSneakCfg": {
    /*ro, req, object, 反潜配置能力*/
    "enable": "true,false",
    /*ro, req, string, 使能反潜回*/
    "startCardReaderNo": {
      /*ro, opt, object, 反潜回起始读卡器编号, desc:反潜回起始读卡器编号, 为0表示不关注起始读卡器编号*/
      "@min": 1,
      /*ro, opt, int, 最小值*/
      "@max": 4,
      /*ro, opt, int, 最大值*/
      "@opt": [1, 4]
      /*ro, opt, array, 枚举值范围, subType:int, desc:闸机产品仅支持1和4, 不需要返回min和max属性*/
    }
  }
}
```

69.39.5 配置清空反潜回参数

Request URL

PUT /ISAPI/AccessControl/ClearAntiSneakCfg?format=json

查询参数

无

请求报文

```
{
  "ClearAntiSneakCfg": {
    /*req, object, 清空反潜回配置*/
    "ClearFlags": {
      /*req, object, 清空反潜回标志*/
      "antiSneak": true
      /*req, bool, 反潜回参数是否使能*/
    }
  }
}
```

响应报文

```
{
  "requestURL": "test",
  /*ro, opt, string, URI*/
  "statusCode": 1,
  /*ro, opt, int, 状态码, desc:无法用1表示时（1表示成功且无特殊状态）必须返回*/
  "statusString": "OK",
  /*ro, opt, string, 状态描述, range:[1,64], desc:无法用OK表示时（OK表示成功且无特殊状态）必须返回*/
  "subStatusCode": "ok",
  /*ro, opt, string, 子状态码, range:[1,64], desc:无法用ok表示（ok表示成功且无特殊状态）必须返回*/
  "errorCode": 1,
  /*ro, opt, int, 错误码, desc:当statusCode不为1时，与subStatusCode对应*/
  "errorMsg": "ok"
  /*ro, opt, string, 错误信息, desc:当statusCode不为1时，解释信息在协议约束中，允许设备在后续的版本迭代中，进行优化丰富提升（不限制死）*/
}
```

69.39.6 配置反潜回参数

Request URL

PUT /ISAPI/AccessControl/AntiSneakCfg?format=json

查询参数

无

请求报文

```
{
  "AntiSneakCfg": {
    /*req, object, 反潜回参数*/
    "enable": true,
    /*req, bool, 使能反潜回*/
    "startCardReaderNo": 1
    /*opt, int, 反潜回起始读卡器编号, desc:反潜回起始读卡器编号，为0表示不关注起始读卡器编号*/
  }
}
```

响应报文

```
{
  "requestURL": "test",
  /*ro, opt, string, URI*/
  "statusCode": 1,
  /*ro, opt, int, 状态码, desc:无法用1表示时（1表示成功且无特殊状态）必须返回*/
  "statusString": "OK",
  /*ro, opt, string, 状态描述, range:[1,64], desc:无法用OK表示时（OK表示成功且无特殊状态）必须返回*/
  "subStatusCode": "ok",
  /*ro, opt, string, 子状态码, range:[1,64], desc:无法用ok表示（ok表示成功且无特殊状态）必须返回*/
  "errorCode": 1,
  /*ro, opt, int, 错误码, desc:当statusCode不为1时，与subStatusCode对应*/
  "errorMsg": "ok"
  /*ro, opt, string, 错误信息, desc:当statusCode不为1时，解释信息在协议约束中，允许设备在后续的版本迭代中，进行优化丰富提升（不限制死）*/
}
```

69.39.7 获取读卡器反潜回参数

Request URL

GET /ISAPI/AccessControl/CardReaderAntiSneakCfg/<cardReaderID>?format=json

查询参数

参数名称	参数类型	描述
cardReaderID	string	读卡器编号ID

请求报文

无

响应报文

```
{
  "CardReaderAntiSneakCfg": {
    /*ro, req, object, 读卡器反潜回参数*/
    "enable": true,
    /*ro, req, bool, 是否加入反潜回路路径, desc:true-加入, false-不加入*/
    "followUpCardReader": [2, 3, 4]
    /*ro, opt, array, 后续读卡器编号, subType:int, desc:[2,3,4]代表该读卡器后续可刷读卡器2或读卡器3或读卡器4*/
  }
}
```

69.39.8 获取反潜回参数

Request URL

GET /ISAPI/AccessControl/AntiSneakCfg?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "AntiSneakCfg": {
    /*ro, req, object, 反潜回参数*/
    "enable": true,
    /*ro, req, bool, 使能反潜回*/
    "startCardReaderNo": 1
    /*ro, opt, int, 反潜回起始读卡器编号, desc:反潜回起始读卡器编号, 为0表示不关注起始读卡器编号*/
  }
}
```

69.40 口罩检测

69.40.1 配置口罩检测参数

Request URL

PUT /ISAPI/AccessControl/maskDetection?format=json

查询参数

无

请求报文

```
{
  "MaskDetection": {
    /*req, object, 口罩检测*/
    "enable": true,
    /*opt, bool, 是否开启口罩检测*/
    "noMaskStrategy": "noTipsAndOpenDoor"
    /*opt, enum, 检测到未带口罩策略, subType:string, [noTipsAndOpenDoor#不提示且开门,tipsAndOpenDoor#提示且开门（默认）,tipsAndNotOpenDoor#提示且不开门,noTipsAndConsume#不提示且允许消费,tipsAndConsume#提示且允许消费,tipsAndNotConsume#提示且不允许消费,tipsAndNotRegister#提示且不允许访客登记], desc:enable为true时, 该字段生效*/
  }
}
```

响应报文

```
{
  "statusCode": 1,
  /*ro, opt, int, 状态码, desc:无法用1表示时（1表示成功且无特殊状态）必须返回*/
  "statusString": "OK",
  /*ro, opt, string, 状态描述, range:[1,64], desc:无法用OK表示时（OK表示成功且无特殊状态）必须返回*/
  "subStatusCode": "ok",
  /*ro, opt, string, 子状态码, range:[1,64], desc:无法用ok表示（ok表示成功且无特殊状态）必须返回*/
  "errorCode": 1,
  /*ro, opt, int, 错误码, desc:当statusCode不为1时，与subStatusCode对应*/
  "errorMsg": "ok"
  /*ro, opt, string, 错误信息, desc:当statusCode不为1时，必须返回,解释信息在协议约束中，允许设备在后续的版本迭代中，进行优化丰富提升（不限制死）*/
}
```

69.40.2 获取口罩检测参数

Request URL

GET /ISAPI/AccessControl/maskDetection?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "MaskDetection": {
    /*ro, req, object, 口罩检测*/
    "enable": true,
    /*ro, opt, bool, 是否开启口罩检测*/
    "noMaskStrategy": "noTipsAndOpenDoor"
    /*ro, opt, enum, 检测到未戴口罩策略, subType:string, [noTipsAndOpenDoor#不提示且开门,tipsAndOpenDoor#提示且开门（默认）,tipsAndNotOpenDoor#提示且不
    开门,noTipsAndConsume#不提示且允许消费（默认）,tipsAndConsume#提示且允许消费,tipsAndNotConsume#提示且不允许消费,tipsAndNotRegister#提示且不允许访客登记],
    desc:enable为true时，该字段生效*/
  }
}
```

69.40.3 获取口罩检测参数能力

Request URL

GET /ISAPI/AccessControl/maskDetection/capabilities?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "MaskDetectionCap": {
    /*ro, req, object, 口罩检测能力节点*/
    "enable": {
      /*ro, opt, object, 是否开启口罩检测, desc:true-开启, false-不开启（该字段默认值会随着市场策略的变更而发生变更）*/
      "@opt": [true, false]
      /*ro, req, array, 可选项, subType:bool*/
    },
    "noMaskStrategy": {
      /*ro, opt, object, 检测到未戴口罩策略, desc:enable为true时，该字段生效*/
      "@opt": ["noTipsAndOpenDoor", "tipsAndOpenDoor", "tipsAndNotOpenDoor", "noTipsAndConsume", "tipsAndConsume", "tipsAndNotConsume",
      "tipsAndNotRegister"]
      /*ro, req, array, 可选项, subType:string*/
    }
  }
}
```

69.41 门禁主机管理(S)

69.41.1 获取门禁主机工作状态能力

Request URL

GET /ISAPI/AccessControl/AcsWorkStatus/capabilities?format=json

查询参数

无

请求报文

无

响应报文

湖南云舟电力科技有限公司
有效期：2024-08-15

```

{
  "AcsWorkStatus": {
    /*ro, req, object, 门禁主机工作状态*/
    "doorLockStatus": {
      /*ro, opt, object, 门锁状态*/
      "@opt": "0,1,2,3,4"
      /*ro, req, string, 门锁状态（继电器开合状态）*/
    },
    "doorStatus": {
      /*ro, opt, object, 门状态*/
      "@opt": "1,2,3,4"
      /*ro, req, string, 门状态（楼层状态）*/
    },
    "magneticStatus": {
      /*ro, opt, object, 门磁状态*/
      "@opt": "0,1,2,3,4"
      /*ro, req, string, 门磁状态*/
    },
    "caseStatus": {
      /*ro, opt, object, 事件触发器状态*/
      "@min": 1,
      /*ro, req, int, 事件触发器状态, step:1*/
      "@max": 1
      /*ro, req, int, 事件触发器状态, step:1*/
    },
    "antiSneakStatus": {
      /*ro, opt, object, 反潜回状态*/
      "@opt": "close,open"
      /*ro, req, string, 反潜回状态*/
    },
    "hostAntiDismantleStatus": {
      /*ro, opt, object, 主机防拆状态*/
      "@opt": "close,open"
      /*ro, req, string, 主机防拆状态*/
    },
    "cardReaderOnlineStatus": {
      /*ro, opt, object, 485识别单元在线状态*/
      "@min": 1,
      /*ro, req, int, 最小值, range:[1,8], step:1*/
      "@max": 1
      /*ro, req, int, 最大值, range:[1,8], step:1*/
    },
    "cardReaderAntiDismantleStatus": {
      /*ro, opt, object, 读卡器防拆状态*/
      "@min": 1,
      /*ro, req, int, 读卡器防拆状态, step:1*/
      "@max": 1
      /*ro, req, int, 读卡器防拆状态, step:1*/
    },
    "cardReaderVerifyMode": {
      /*ro, opt, object, 读卡器当前验证方式,
      desc:1-休眠, 2-刷卡+密码, 3-刷卡, 4-刷卡或密码, 5-指纹, 6-指纹+密码, 7-指纹或刷卡, 8-指纹+刷卡, 9-指纹+刷卡+密码, 10-人脸或指纹或刷卡或密码, 11-人脸+指纹, 12-人脸+密码, 13-人脸+刷卡, 14-人脸,
      15-工号+密码, 16-指纹或密码, 17-工号+指纹, 18-工号+指纹+密码, 19-人脸+指纹+刷卡, 20-人脸+密码+指纹, 21-工号+人脸, 22-人脸或人脸+刷卡, 23-指纹或人脸, 24-刷卡或人脸或密码, 25-人脸或刷卡, 26-人脸或刷卡或指纹, 27-刷卡或指纹或密码,
      28-人脸或密码, 29-工号+人脸+密码, 30-刷卡或人脸或人脸+刷卡, 31-人脸或指纹或密码, 32-虹膜, 33-人脸或指纹或刷卡或密码或虹膜, 34-人脸或刷卡或密码或虹膜*/
      "@opt": "1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34"
      /*ro, req, string, 读卡器当前验证方式*/
    },
    "alarmOutStatus": {
      /*ro, opt, object, 报警输出口状态*/
      "@min": 1,
      /*ro, req, int, 报警输出口状态, step:1*/
      "@max": 1
      /*ro, req, int, 报警输出口状态, step:1*/
    },
    "cardNum": {
      /*ro, opt, object, 已添加的卡数量*/
      "@min": 1,
      /*ro, req, int, 已添加的卡数量, step:1*/
      "@max": 1
      /*ro, req, int, 已添加的卡数量, step:1*/
    },
  },
}

```

69.41.2 获取门禁主机工作状态

Request URL

GET /ISAPI/AccessControl/AcsWorkStatus?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "AcsWorkStatus": {
    /*ro, req, object, 门禁工作状态*/
    "doorLockStatus": [1, 2, 1, 2],
    /*ro, opt, enumarray, 门锁状态（继电器开合状态），subType:int, [0#正常关,1#正常开,2#短路报警,3#断路报警,4#异常报警], desc:门锁状态（继电器开合状态），0-正常关, 1-正常开, 2-短路报警, 3-断路报警, 4-异常报警（[1,2,1,2]代表：门锁1正常开, 门锁2短路报警, 门锁3正常开, 门锁4短路报警）*/
    "doorStatus": [1, 2, 1, 2],
    /*ro, opt, enumarray, 门状态（楼层状态），subType:int, [1#休眠,2#常开状态（自由）,3#常闭状态（禁用）,4#普通状态（受控）], desc:门状态（楼层状态），1-休眠, 2-常开状态（自由）, 3-常闭状态（禁用）, 4-普通状态（受控）（[1,2,1,2]代表：门1休眠, 门2常开状态, 门3休眠, 门4常开状态）*/
    "magneticStatus": [1, 2, 1, 2],
    /*ro, opt, enumarray, 门磁状态, subType:int, [0#正常关,1#正常开,2#短路报警,3#断路报警,4#异常报警], desc:门磁状态, 0-正常关, 1-正常开, 2-短路报警, 3-断路报警, 4-异常报警（[1,2,1,2]代表：门磁1正常开, 门磁2短路报警, 门磁3正常开, 门磁4短路报警）*/
    "caseStatus": [1, 3, 5],
    /*ro, opt, array, 事件触发器状态, subType:int, desc:事件触发器状态（[1,3,5]代表：事件触发器1、事件触发器3、事件触发器5有输入）*/
    "batteryVoltage": 50,
    /*ro, opt, int, 蓄电池电压值, desc:蓄电池电压值, 实际值乘10, 单位：伏特*/
    "batteryLowVoltage": false,
    /*ro, opt, bool, 蓄电池是否处于低压状态, desc:蓄电池是否处于低压状态, false-否, true-是*/
    "powerSupplyStatus": "ACPowerSupply",
    /*ro, opt, enum, 设备供电状态, subType:string, [ACPowerSupply#交流电供电,BatteryPowerSupply#蓄电池供电]*/
    "multiDoorInterlockStatus": "close",
    /*ro, opt, enum, 多门互锁状态, subType:string, [close#关闭,open#开启]*/
    "antiSneakStatus": "open",
    /*ro, opt, enum, 反潜回状态, subType:string, [close#关闭,open#开启]*/
    "hostAntiDismantleStatus": "open",
    /*ro, opt, enum, 主机防拆状态, subType:string, [close#关闭,open#开启]*/
    "indicatorLightStatus": "online",
    /*ro, opt, enum, 指示灯状态, subType:string, [offline#掉线,online#在线]*/
    "cardReaderOnlineStatus": [1, 3, 5],
    /*ro, opt, array, 485识别单元状态, subType:int, desc:485读卡器（485识别单元）在线状态（[1,3,5]代表：485读卡器1、3、5在线）*/
    "netReaderOnlineStatus": [1, 3, 5],
    /*ro, opt, array, 网络识别单元状态, subType:int, desc:网络识别单元在线状态（[1,3,5]代表：网络识别单元1、3、5在线）*/
    "POEPortList": [
      /*ro, opt, array, 主机的POE端口信息, subType:object, range:[1,8]*/
      {
        "port": 1,
        /*ro, req, int, POE物理网络端口, range:[1,8]*/
        "readerID": [1, 2, 3],
        /*ro, opt, array, 识别单元ID, subType:int, desc:表示识别单元ID 1,2,3关联在POE端口1下*/
      }
    ],
    "cardReaderAntiDismantleStatus": [1, 3, 5],
    /*ro, opt, array, 读卡器防拆状态, subType:int, desc:读卡器防拆状态（[1,3,5]代表：读卡器1、读卡器3、读卡器5防拆功能开启）*/
    "cardReaderVerifyMode": [3, 5, 3, 5],
    /*ro, opt, enumarray, 读卡器当前验证方式, subType:int, [1#休眠,2#刷卡+密码,3#刷卡,4#刷卡或密码,5#指纹,6#指纹+密码,7#指纹或刷卡,8#指纹+刷卡,9#指纹+刷卡+密码,10#人脸或指纹或刷卡或密码,11#人脸+指纹,12#人脸+密码,13#人脸+刷卡,14#人脸,15#工号+密码,16#指纹或密码,17#工号+指纹,18#工号+指纹+密码,19#人脸+指纹+刷卡,20#人脸+密码+指纹,21#工号+人脸,22#人脸或人脸+刷卡,23#指纹或人脸,24#刷卡或人脸或密码,25#人脸或刷卡,26#人脸或刷卡或指纹,27#刷卡或指纹或密码,28#人脸或密码,29#工号+人脸+密码,30#刷卡或人脸或人脸+刷卡,31#人脸或指纹或密码,32#虹膜,33#人脸或指纹或刷卡或密码或虹膜,34#人脸或刷卡或密码或虹膜], desc:（[3,5,3,5]代表：读卡器1当前验证方式为刷卡, 读卡器2当前验证方式为指纹, 读卡器3当前验证方式为刷卡, 读卡器4当前验证方式为指纹）*/
    "setupAlarmStatus": [1, 3, 5],
    /*ro, opt, array, 报警输入口布防状态, subType:int, desc:报警输入口布防状态（[1,3,5]代表：报警输入口1、报警输入口3、报警输入口5处于布防状态）*/
    "alarmInStatus": [1, 3, 5],
    /*ro, opt, array, 报警输入口报警状态, subType:int, desc:报警输入口报警状态（[1,3,5]代表：报警输入口1、报警输入口3、报警输入口5有报警）*/
    "alarmOutStatus": [1, 3, 5],
    /*ro, opt, array, 报警输出状态, subType:int, desc:报警输出状态（[1,3,5]代表：报警输出1、报警输出3、报警输出5有报警）*/
    "cardNum": 3,
    /*ro, opt, int, 已添加的卡数量, desc:已添加的卡数量*/
    "fireAlarmStatus": "normal",
    /*ro, opt, enum, 消防报警状态显示, subType:string, [normal#正常,shortCircuit#短路报警,brokenCircuit#断开报警]*/
    "batteryChargeStatus": "charging",
    /*ro, opt, enum, 电池充电状态, subType:string, [charging#充电中,uncharged#未充电]*/
    "masterChannelControllerStatus": "online",
    /*ro, opt, enum, 主通道控制器在线状态, subType:string, [offline#掉线,online#在线]*/
    "slaveChannelControllerStatus": "online",
    /*ro, opt, enum, 从通道控制器在线状态, subType:string, [offline#掉线,online#在线]*/
    "antiSneakServerStatus": "normal",
    /*ro, opt, enum, 反潜回服务器状态, subType:string, [disable#未启用,norma1#正常,disconnect#断开]*/
    "netStatus": "connect",
    /*ro, opt, enum, 网络状态, subType:string, [connect#网络正常连接,disconnect#网络断开连接,ipConflict#网络IP冲突], desc:单网卡时返回, 指有线网络*/
    "InterfaceStatusList": [
      /*ro, opt, array, 多网卡状态, subType:object, desc:多网卡时返回*/
      {
        "id": 1,
        /*ro, opt, int, 网卡编号*/
        "netType": 1,
        /*ro, opt, enum, 网卡类型, subType:int, [1#有线网络,2#WiFi]*/
        "netStatus": "connect",
        /*ro, opt, enum, 网络状态, subType:string, [connect#网络正常连接,disconnect#网络断开连接,ipConflict#网络IP冲突]*/
      }
    ],
    "signalStatus": "noSignal",
    /*ro, opt, enum, 信号状态, subType:string, [noSignal#无信号,2G#2G信号,3G#3G信号,4G#4G信号,5G#5G信号]*/
    "signalStrength": 0,
    /*ro, opt, int, 移动网络信号强度, range:[0,100], step:1, dep:and,{$.AcsWorkStatus.signalStatus,ue,noSignal}, desc:依赖于signalStatus字段, 当该字段不为noSignal时有效*/
    "sipStatus": "connect",
    /*ro, opt, enum, SIP状态, subType:string, [connect#SIP正常连接,disconnect#SIP断开连接,unregistered#SIP未注册]*/
    "ezvizStatus": "unregistered".
  }
}
```

```

/*ro, opt, enum, EZVIZ状态, subType:string, [unregistered#设备未注册EZVIZ,notAdd#设备未添加到EZVIZ,connect#设备正常连接到EZVIZ]*/
"voipStatus": "unregistered",
/*ro, opt, enum, VOIP状态, subType:string, [unregistered#VOIP未注册,connect#VOIP正常连接,connecting#连接中,noModule#无模块,notEnabled#未启用]*/
"wifiStatus": "connect",
/*ro, opt, enum, WIFI状态, subType:string, [connect#WIFI正常连接,disconnect#WIFI断开连接]*/
"TFCardStatus": "mounted",
/*ro, opt, enum, TF卡状态, subType:string, [mounted#已挂载,unmounted#未挂载]*/
"acrossHostInterlockStatus": "disable",
/*ro, opt, enum, 跨主机多门互锁状态, subType:string, [disable#未启用,normal#正常,disconnect#断开]*/
"faceReceivingStatus": "receiving",
/*ro, opt, enum, 人脸接收状态, subType:string, [receiving#接收中,received#接收完毕]*/
"QRCodeReaderStatusList": [
/*ro, opt, array, 二维码阅读器状态列表, subType:object*/
{
    "id": 1,
/*ro, opt, int, 二维码阅读器编号*/
    "QRCodeReaderStatus": "online"
/*ro, opt, enum, 二维码阅读器状态, subType:string, [online#在线,offline#离线]*/
},
],
"GooseneckStatus": "mounted",
/*ro, opt, enum, 鹅颈插拔状态, subType:string, [mounted#已挂载,unmounted#未挂载]*/
"wifiSignalStrength": 0,
/*ro, opt, int, wifi信号强度, range:[0,100], step:1, dep:and,{$.AcsWorkStatus.wifiStatus,eq,connect}, desc:当wifiStatus为connect时可选有效*/
"thermopileStatus": "connect",
/*ro, opt, enum, 热电堆连接状态, subType:string, [connect#热电堆正常连接,disconnect#热电堆断开连接]*/
"modelStatus": "processing",
/*ro, opt, enum, 人脸建模状态, subType:string, [processing#建模中,finish#已完成], desc:设备刚启动时, 存在人脸数据还未建模时返回*/
"ISUPRegisterStatus": "online",
/*ro, opt, enum, ISUP注册状态, subType:string, [online#在线,offline#离线]*/
"newSipStatus": "connect"
/*ro, opt, enum, 新长号的SIP注册状态, subType:string, [connect#SIP正常连接,disconnect#SIP断开连接,unregistered#SIP未注册], desc:安卓管理机上支持展示两个长号的SIP注册状态*/
}
}
}

```

69.41.3 获取门禁主机参数能力

Request URL

GET /ISAPI/AccessControl/AcsCfg/capabilities?format=json

查询参数

无

请求报文

无

响应报文

```

{
  "AcsCfg": {
    /*ro, req, object, 门禁主机参数配置*/
    "voicePrompt": "true,false",
    /*ro, opt, string, 是否启用语音提示, desc:false-不启用, true-启用*/
    "uploadCapPic": "true,false",
    /*ro, opt, string, 联动抓拍是否上传图片, desc:false-不上传, true-上传*/
    "saveCapPic": "true,false",
    /*ro, opt, string, 是否保存抓拍图片, desc:false-不保存, true-保存*/
    "enable3G4G": "true,false",
    /*ro, opt, string, 3G4G使能, desc:false-不使能, true-使能*/
    "showPicture": "true,false",
    /*ro, opt, string, 是否显示认证照片, desc:false-不显示, true-显示*/
    "showEmployeeNo": "true,false",
    /*ro, opt, string, 是否显示认证工号, desc:false-不显示, true-显示*/
    "showName": "true,false",
    /*ro, opt, string, 是否显示认证姓名, desc:false-不显示, true-显示*/
    "desensitiseEmployeeNo": {
      /*ro, opt, object, 本地UI显示时是否进行认证工号脱敏*/
      "@opt": [true, false]
      /*ro, opt, array, 取值范围, subType:bool*/
    },
    "desensitiseName": {
      /*ro, opt, object, 本地UI显示时是否进行认证姓名脱敏*/
      "@opt": [true, false]
      /*ro, opt, array, 取值范围, subType:bool*/
    },
    "thermalEnabled": {
      /*ro, opt, object, 测温使能*/
      "@opt": [true, false]
      /*ro, opt, array, 可选项, subType:bool*/
    },
    "thermalMode": {
      /*ro, opt, object, 测温模型*/
      "@opt": [true, false]
      /*ro, opt, array, 可选项, subType:bool*/
    },
    "thermalPictureEnabled": {
      /*ro, opt, object, 测温图片使能, desc:仅测温模式可见光图片上传使能（用于控制抓拍图片和热成像可见光图片是否上传）， true-开启, false-关闭（默认）*/
      "@opt": [true, false]
      /*ro, opt, array, 仅测温模式可见光图片上传使能（用于控制抓拍图片和热成像可见光图片是否上传）， subType:bool*/
    },
    "highestThermalThreshold": {
      /*ro, opt, object, 温度阈值上限, desc:单位: 摄氏度, 精度: 小数点后一位*/
      "@min": 1.0,
      /*ro, opt, float, 最小值*/
      "@max": 2.0
      /*ro, opt, float, 最大值*/
    },
    "lowestThermalThreshold": {
      /*ro, opt, object, 温度阈值下限, desc:单位: 摄氏度, 精度: 小数点后一位*/
      "@min": 1.0,
      /*ro, opt, float, 最小值*/
      "@max": 2.0
      /*ro, opt, float, 最大值*/
    },
    "uploadVerificationPic": "true,false",
    /*ro, opt, string, 认证是否上传图片, desc:false-不上传, true-上传*/
    "saveVerificationPic": "true,false",
    /*ro, opt, string, 是否保存认证图片, desc:false-不保存, true-保存*/
    "saveFacePic": "true,false",
    /*ro, opt, string, 是否保存人脸底图, desc:false-不保存, true-保存*/
    "thermalUnit": {
      /*ro, opt, object, 测温单位配置, desc:celsius-摄氏度（默认）> fahrenheit-华氏度*/
      "@opt": ["celsius", "fahrenheit"]
      /*ro, opt, array, 可选项, subType:string*/
    },
    "highestThermalThresholdF": {
      /*ro, opt, object, 温度阈值上限, desc:单位: 华氏度, 精度: 小数点后一位*/
      "@min": 1.0,
      /*ro, opt, float, 最小值*/
      "@max": 1.0
      /*ro, opt, float, 最大值*/
    },
    "lowestThermalThresholdF": {
      /*ro, opt, object, 温度阈值下限, desc:单位: 华氏度, 精度: 小数点后一位*/
      "@min": 1.0,
      /*ro, opt, float, 最小值*/
      "@max": 1.0
      /*ro, opt, float, 最大值*/
    },
    "thermalCompensation": {
      /*ro, opt, object, 温度补偿*/
      "@min": -99.9,
      /*ro, opt, float, 最小值*/
      "@max": 99.9
      /*ro, opt, float, 最大值*/
    }
  },
}
}

```

69.41.4 配置门禁主机参数

Request URL

PUT /ISAPI/AccessControl/AcsCfg?format=json

查询参数

无

请求报文

```
{
  "AcsCfg": {
    /*req, object, 门禁主机参数配置*/
    "RS485Backup": true,
    /*opt, bool, 是否启用下行RS485通信备份功能: false-不启用*/
    "showCapPic": true,
    /*opt, bool, 是否显示抓拍图片*/
    "showUserInfo": true,
    /*opt, bool, 是否显示用户信息*/
    "overlayUserInfo": true,
    /*opt, bool, 是否叠加用户信息*/
    "voicePrompt": true,
    /*opt, bool, 是否启用语音提示*/
    "uploadCapPic": true,
    /*opt, bool, 联动抓拍是否上传图片*/
    "saveCapPic": true,
    /*opt, bool, 是否保存抓拍图片*/
    "inputCardNo": true,
    /*opt, bool, 是否是否允许按键输入卡号*/
    "enableWifiDetect": true,
    /*opt, bool, 是否启动wifi探针*/
    "enable3G4G": true,
    /*opt, bool, 3G4G使能*/
    "protocol": "Private",
    /*opt, enum, 读卡器通信协议类型, subType:string, [Private#私有协议,OSDP#OSDP协议]*/
    "enableCaptureCertificate": true,
    /*opt, bool, 是否启动证件抓拍, desc:false-不启动, true-启动, 无该字段不支持*/
    "showPicture": true,
    /*opt, bool, 是否显示认证照片*/
    "showEmployeeNo": true,
    /*opt, bool, 是否显示认证工号*/
    "showName": true,
    /*opt, bool, 是否显示认证姓名*/
    "desensitiseEmployeeNo": true,
    /*opt, bool, 本地UI显示时是否进行认证工号脱敏, dep:or,{$.AcsCfg.showEmployeeNo,eq,true}*/
    "desensitiseName": true,
    /*opt, bool, 本地UI显示时是否进行认证姓名脱敏, dep:or,{$.AcsCfg.showName,eq,true}*/
    "thermalEnabled": true,
    /*opt, bool, 测温开关使能*/
    "thermalMode": true,
    /*opt, bool, 是否开启仅测温模式*/
    "thermalPictureEnabled": true,
    /*opt, bool, 仅测温模式可见光图片上传使能（用于控制抓拍图片和热成像可见光图片是否上传）*/
    "thermalIp": "192.168.1.1",
    /*opt, string, 热成像测温仪ip, desc:门禁设备采用和卡片机定制的协议集成的, 只需要一个ip地址即可添加到网络; 对于安检门设备, 专门用一个网口连接热成像测温仪, 故不需要配置*/
    "highestThermalThreshold": 37.3,
    /*opt, float, 温度阈值上限*/
    "lowestThermalThreshold": 38.5,
    /*opt, float, 温度阈值下限*/
    "thermalDoorEnabled": false,
    /*opt, bool, 温度阈值开门使能, desc:超过温度阈值上限(highestThermalThreshold)/低于温度阈值下限(lowestThermalThreshold)后是否开门, true-开门, false-不开门(默认)*/
    "QRCodeEnabled": false,
    /*opt, bool, 二维码功能使能*/
    "remoteCheckDoorEnabled": false,
    /*opt, bool, 远程检验是否控制门的开关*/
    "checkChannelType": "Ezviz",
    /*opt, enum, 检验通道类型, subType:string, [Ezviz#萤石通道,ISUP#ISUP(原EHome)通道,ISAPI#ISAPI通道,PrivateSDK#私有SDK通道,ISAPIListen#ISAPI监听通道], dep:or,{$.AcsCfg.remoteCheckDoorEnabled,eq,true}, desc:当remoteCheckDoorEnabled字段为true时, 该字段生效*/
    "channelIp": "test",
    /*opt, string, 检验通道ip, desc:当checkChannelType字段为PrivateSDK时, 该字段生效*/
    "needDeviceCheck": true,
    /*opt, bool, 远程检验是否需要设备校验, dep:or,{$.AcsCfg.remoteCheckDoorEnabled,eq,true}*/
    "remoteCheckTimeout": 5,
    /*opt, int, 远程检验超时时间, range:[1,10], unit:s, unitType:时间, dep:or,{$.AcsCfg.remoteCheckDoorEnabled,eq,true}, desc:无论是布防链路的异步返回核验结果, 还是监听链路的同步返回结果, 设备均需遵循这个超时时间.*/
    "remoteCheckVerifyMode": 1,
    /*opt, enum, 仅需要远程检验的验证方式, subType:int, [1#人证比对,2#刷卡,3#人证比对或刷卡,4#刷卡,5#人证比对或刷卡,6#刷卡或刷卡,7#人证比对或刷卡或刷卡,8#二维码], dep:or,{$.AcsCfg.remoteCheckDoorEnabled,eq,true}, desc:示例: 假设配置为刷卡, 则表示刷卡时需要远程检验, 但是不需要本地检验; 而其他的认证方式需要本地检验+远程检验.*/
    "offlineDevCheckOpenDoorEnabled": false,
    /*opt, bool, 离线远程检验时是否支持以设备本地认证结果开门, dep:or,{$.AcsCfg.remoteCheckDoorEnabled,eq,true},
    desc:1. true-支持, false-不支持, 默认为false.
    2. 存在特殊情况, 远程检验开启了, 设备上报告事件后等待平台响应检验结果, 但是平台和设备间通信异常, 设备收不到检验结果, 此时人员又需要着急进出门.*/
    "uploadVerificationPic": true,
    /*opt, bool, 认证是否上传图片*/
    "uploadVerificationPicType": 0,
    /*opt, enum, 认证上传图片类型, subType:int, [0#人脸抠图,1#全景图], desc:不下发该字段, 默认为全景图.*/
```

```

        "saveVerificationPic": true,
        /*opt, bool, 是否保存认证图片*/
        "saveFacePic": true,
        /*opt, bool, 是否保存人脸底图*/
        "thermalUnit": "celsius",
        /*opt, enum, 测温单位配置, subType:string, [celsius#摄氏度,fahrenheit#华氏度]*/
        "highestThermalThreshold": 1.0,
        /*opt, float, 华氏度温度阈值上限, desc:单位: 华氏度, 精度: 小数点后一位。用于判断超出该阈值门禁设备是否开门,温度字段若未0表明不启用该字段, 需增加注
释*/
        "lowestThermalThreshold": 1.0,
        /*opt, float, 华氏度温度阈值下限, desc:单位: 华氏度, 精度: 小数点后一位。用于判断超出该阈值门禁设备是否开门,温度字段若未0表明不启用该字段, 需增加注
释*/
        "enable5G": true,
        /*opt, bool, 5G使能*/
        "thermalCompensation": 1.0,
        /*opt, float, 温度补偿, desc:单位根据thermalUnit字段确定, 若thermalUnit字段不存在认为单位为摄氏度*/
        "externalCardReaderEnabled": true,
        /*opt, bool, 使能副读卡器作为主读卡器的使用, desc:该字段默认为false, 为true时, 用户可以在副读卡器和主读卡器上同时完成组合认证, 即将副读卡器作为扩展
模块使用*/
        "combinationAuthenticationTimeout": 1,
        /*opt, int, 组合认证超时时间, range:[1,20], step:1, unit:s, unitType:时间, desc:组合认证时, 单步认证达到超时时间后认为组合认证超时*/
        "combinationAuthenticationLimitOrder": true,
        /*opt, bool, 组合认证限制顺序, desc:为true时, 限制组合认证顺序, 为false则可以使用任一有效凭证进行首次认证, 认证通过后, 后续认证顺序按刷卡>指纹>人脸>
密码顺序进行, 该字段默认为true*/
        "passwordEnabled": true,
        /*opt, bool, 密码使能, desc:设备是否开启密码认证功能, 若关闭该功能, 则无法使用密码进行认证*/
        "showGender": true,
        /*opt, bool, 是否显示性别*/
        "showSignInTime": true,
        /*opt, bool, 是否显示签到时间*/
        "showsCustomInfo": true,
        /*opt, bool, 是否显示自定义字段*/
        "showMobileWebQRCode": true,
        /*opt, bool, 是否显示手机web二维码*/
        "fireAlarmInputType": "alwaysOpen",
        /*opt, enum, 消防输入类型, subType:string, [alwaysClose#常闭,alwaysOpen#常开]*/
        "buzzerEnabled": true
        /*opt, bool, 蜂鸣器使能, desc:使能开启后, 若未启用语音提示, 则认证成功快速蜂鸣两次, 认证失败慢速蜂鸣3次*/
    }
}

```

响应报文

```

{
    "requestURL": "test",
    /*ro, opt, string, URI*/
    "statusCode": "test",
    /*ro, opt, string, 状态码*/
    "statusString": "test",
    /*ro, opt, string, 状态描述*/
    "subStatusCode": "test",
    /*ro, opt, string, 子状态码*/
    "errorCode": 1,
    /*ro, req, int, 错误码*/
    "errorMsg": "ok"
    /*ro, req, string, 错误信息*/
}

```

69.41.5 获取门禁主机参数

Request URL

GET /ISAPI/AccessControl/AcsCfg?format=json

查询参数

无

请求报文

无

响应报文

```

{
    "AcsCfg": {
        /*ro, req, object, 门禁主机参数配置*/
        "RS485Backup": true,
        /*ro, opt, bool, 是否启用下行RS485通信备份功能: false-不启用*/
        "showCapPic": true,
        /*ro, opt, bool, 是否显示抓拍图片*/
        "showUserInfo": true,
        /*ro, opt, bool, 是否显示用户信息*/
        "overlayUserInfo": true,
        /*ro, opt, bool, 是否叠加用户信息*/
    }
}

```

```
"voicePrompt": true,
/*ro, opt, bool, 是否启用语音提示*/
"uploadCapPic": true,
/*ro, opt, bool, 联动抓拍是否上传图片*/
"saveCapPic": true,
/*ro, opt, bool, 是否保存抓拍图片*/
"inputCardNo": true,
/*ro, opt, bool, 是否是否允许按键输入卡号*/
"enableWifiDetect": true,
/*ro, opt, bool, 是否启动wifi探针*/
"enable3G4G": true,
/*ro, opt, bool, 3G4G使能*/
"protocol": "Private",
/*ro, opt, enum, 读卡器通信协议类型, subType:string, [Private#私有协议,OSDP#OSDP协议]*/
"enableCaptureCertificate": true,
/*ro, opt, bool, 是否启动证件抓拍, desc:false-不启动, true-启动, 无该字段不支持*/
"showPicture": true,
/*ro, opt, bool, 是否显示认证照片*/
"showEmployeeNo": true,
/*ro, opt, bool, 是否显示认证工号*/
"showName": true,
/*ro, opt, bool, 是否显示认证姓名*/
"desensitiseEmployeeNo": true,
/*ro, opt, bool, 本地UI显示时是否进行认证工号脱敏, dep:or,{$.AcsCfg.showEmployeeNo,eq,true}*/
"desensitiseName": true,
/*ro, opt, bool, 本地UI显示时是否进行认证姓名脱敏, dep:or,{$.AcsCfg.showName,eq,true}*/
"thermalEnabled": true,
/*ro, opt, bool, 测温开关使能*/
"thermalMode": true,
/*ro, opt, bool, 是否开启仅测温模式*/
"thermalPictureEnabled": true,
/*ro, opt, bool, 仅测温模式可见光图片上传使能（用于控制抓拍图片和热成像可见光图片是否上传）*/
"thermalIp": "192.168.1.1",
/*ro, opt, string, 热成像测温仪ip, desc:门禁设备采用和卡片机定制的协议集成的, 只需要一个ip地址即可添加到网络; 对于安检门设备, 专门用一个网口连接热成像测温仪, 故不需要配置*/
"highestThermalThreshold": 37.3,
/*ro, opt, float, 温度阈值上限*/
"lowestThermalThreshold": 38.5,
/*ro, opt, float, 温度阈值下限*/
"thermalDoorEnabled": false,
/*ro, opt, bool, 温度阈值开门使能, desc:超过温度阈值上限（highestThermalThreshold）/低于温度阈值下限（lowestThermalThreshold）后是否开门, true-开门, false-不开门（默认）*/
"QRCodeEnabled": false,
/*ro, opt, bool, 二维码功能使能*/
"remoteCheckDoorEnabled": false,
/*ro, opt, bool, 远程校验是否控制门的开关*/
"checkChannelType": "Ezviz",
/*ro, opt, enum, 校验通道类型, subType:string, [Ezviz#萤石通道,ISUP#ISUP（原EHome）通道,ISAPI#ISAPI通道,PrivateSDK#私有SDK通道,ISAPIListen#ISAPI监听通道], dep:or,{$.AcsCfg.remoteCheckDoorEnabled,eq,true}, desc:当remoteCheckDoorEnabled字段为true时, 该字段生效*/
"channelIp": "test",
/*ro, opt, string, 校验通道ip, desc:当checkChannelType字段为PrivateSDK时, 该字段生效*/
"needDeviceCheck": true,
/*ro, opt, bool, 远程校验是否需要设备校验, dep:or,{$.AcsCfg.remoteCheckDoorEnabled,eq,true}, desc:默认为true*/
"remoteCheckTimeout": 5,
/*ro, opt, int, 远程校验超时时间, range:[1,10], unit:s, unitType:时间, dep:or,{$.AcsCfg.remoteCheckDoorEnabled,eq,true}, desc:无论是布防链路的异步返回校验结果, 还是监听链路的同步返回结果, 设备均需要遵循这个超时时间。*/
"remoteCheckVerifyMode": 1,
/*ro, opt, enum, 仅需要远程校验的验证方式, subType:int, [1#人证比对,2#刷脸,3#人证比对或刷脸,4#刷卡,5#人证比对或刷卡,6#刷脸或刷卡,7#人证比对或刷卡或刷脸,8#二维码], dep:or,{$.AcsCfg.remoteCheckDoorEnabled,eq,true}*/
"offlineDevCheckOpenDoorEnabled": false,
/*ro, opt, bool, 离线远程校验时是否支持以设备本地认证结果开门, dep:or,{$.AcsCfg.remoteCheckDoorEnabled,eq,true},
desc:1. true-支持, false-不支持, 默认为false.
2、存在特殊情况, 远程校验开启了, 设备上报完事件后等待平台响应校验结果, 但是平台和设备间通信异常, 设备收不到校验结果, 此时人员又需要着急进出口。*/
"uploadVerificationPic": true,
/*ro, opt, bool, 认证是否上传图片*/
"uploadVerificationPicType": 0,
/*ro, opt, enum, 认证时上传图片类型, subType:int, [0#人脸抠图,1#全景图], desc:不下发该字段, 默认为全景图。*/
"saveVerificationPic": true,
/*ro, opt, bool, 是否保存认证图片*/
"saveFacePic": true,
/*ro, opt, bool, 是否保存人脸底图*/
"thermalUnit": "celsius",
/*ro, opt, enum, 测温单位配置, subType:string, [celsius#摄氏度,fahrenheit#华氏度]*/
"highestThermalThresholdF": 1.0,
/*ro, opt, float, 华氏度温度阈值上限, desc:单位: 华氏度, 精度: 小数点后一位。用于判断超出该阈值门禁设备是否开门,温度字段若未0表明不启用该字段, 需增加注释*/
"lowestThermalThresholdF": 1.0,
/*ro, opt, float, 华氏度温度阈值下限, desc:单位: 华氏度, 精度: 小数点后一位。用于判断超出该阈值门禁设备是否开门,温度字段若未0表明不启用该字段, 需增加注释*/
"enable5G": true,
/*ro, opt, bool, 5G使能*/
"thermalCompensation": 1.0,
/*ro, opt, float, 温度补偿, desc:单位根据thermalUnit字段确定, 若thermalUnit字段不存在认为单位为摄氏度*/
"externalCardReaderEnabled": true,
/*ro, opt, bool, 使能副读卡器作为主读卡器的使用, desc:该字段默认为false, 为true时, 用户可以在副读卡器和主读卡器上同时完成组合认证, 即将副读卡器作为扩展模块使用*/
"combinationAuthenticationTimeout": 1,
/*ro, opt, int, 组合认证超时时间, range:[1,20], step:1, unit:s, unitType:时间, desc:组合认证时, 单步认证达到超时时间后认为组合认证超时*/
"combinationAuthenticationLimitOrder": true,
/*ro, opt, bool, 组合认证限制顺序, desc:为true时, 限制组合认证顺序, 为false则可以使用任一有效凭证进行首次认证, 认证通过后, 后续认证顺序按刷卡>指纹>人脸>密码顺序进行, 该字段默认为true*/
"passwordEnabled": true,
/*ro, opt, bool, 密码使能, desc:设备是否开启密码认证功能, 若关闭该功能, 则无法使用密码进行认证*/
"showGender": true,
```

```

/*ro, opt, bool, 是否显示性别*/
"showSignInTime": true,
/*ro, opt, bool, 是否显示签到时间*/
"showsCustomInfo": true,
/*ro, opt, bool, 是否显示自定义字段*/
"showMobileWebQRCode": true,
/*ro, opt, bool, 是否显示手机web二维码*/
"fireAlarmInputType": "alwaysOpen",
/*ro, opt, enum, 消防输入类型, subType:string, [alwaysClose#常闭,alwaysOpen#常开]*/
"buzzerEnabled": true
/*ro, opt, bool, 蜂鸣器使能, desc:使能开启后, 若未启用语音提示, 则认证成功快速蜂鸣两次, 认证失败慢速蜂鸣3次*/
}
}

```

69.42 楼宇按键考勤

69.42.1 获取考勤计划模板参数能力

Request URL

GET /ISAPI/AccessControl/Attendance/planTemplate/capabilities?format=json

查询参数

无

请求报文

无

响应报文

```

{
  "AttendancePlanTemplateCap": {
    /*ro, req, object, 考勤计划模板能力*/
    "templateNo": {
      /*ro, opt, object, 计划模板编号*/
      "@min": 1,
      /*ro, opt, int, 最小值*/
      "@max": 16
      /*ro, opt, int, 最大值*/
    },
    "property": {
      /*ro, req, object, 考勤属性, desc:check-上下班 break-休息 overtime-加班*/
      "@opt": ["check", "break", "overtime"]
      /*ro, opt, array, 可选项, subType:String*/
    },
    "enable": {
      /*ro, opt, object, 是否启用, desc:true-启用, false-不启用*/
      "@opt": [true, false]
      /*ro, opt, array, 可选项, subType:bool*/
    },
    "templateName": {
      /*ro, opt, object, 模板名称*/
      "@min": 1,
      /*ro, opt, int, 最小值*/
      "@max": 32
      /*ro, opt, int, 最大值*/
    },
    "weekPlanNo": {
      /*ro, opt, object, 周计划编号*/
      "@min": 1,
      /*ro, opt, int, 最小值*/
      "@max": 16
      /*ro, opt, int, 最大值*/
    },
    "holidayGroupNo": {
      /*ro, opt, object, 假日组编号*/
      "@min": 1,
      /*ro, opt, int, 最小值*/
      "@max": 16
      /*ro, opt, int, 最大值*/
    }
  }
}

```

69.42.2 获取所有考勤计划模板

Request URL

GET /ISAPI/AccessControl/Attendance/planTemplate?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "AttendancePlanTemplateList": [
    /*ro, req, array, 考勤计划模板列表, subType:object*/
    {
      "templateNo": 1,
      /*ro, req, int, 计划模板编号, range:[1,16]*/
      "enable": true,
      /*ro, req, bool, 是否启用, desc:true-启用, false-不启用*/
      "property": "check",
      /*ro, opt, enum, 考勤属性, subType:string, [check#上下班,break#休息,overtime#加班]*/
      "templateName": "test",
      /*ro, req, string, 模板名称, range:[1,32]*/
      "weekPlanNo": 1
      /*ro, req, int, 周计划编号, range:[1,5]*/
    }
  ]
}
```

69.42.3 获取按键考勤参数能力

Request URL

GET /ISAPI/AccessControl/keyCfg/attendance/capabilities?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "AttendanceCap": {
    /*ro, req, object, 考勤参数配置能力节点*/
    "id": {
      /*ro, req, object, 按键编号*/
      "@min": 0,
      /*ro, opt, int, 最小值*/
      "@max": 0
      /*ro, opt, int, 最大值*/
    },
    "enable": {
      /*ro, req, object, 使能*/
      "@opt": [true, false]
      /*ro, opt, array, 可选项, subType:bool*/
    },
    "label": {
      /*ro, opt, object, 自定义名称*/
      "@min": 0,
      /*ro, opt, int, 最小值*/
      "@max": 0
      /*ro, opt, int, 最大值*/
    },
    "attendanceStatus": {
      /*ro, opt, object, 考勤状态, desc:checkIn-上班, checkOut-下班, breakOut-开始休息, breakIn-结束休息, overtimeIn-开始加班, overtimeOut-结束加班*/
      "@opt": ["checkIn", "checkOut", "breakOut", "breakIn", "overtimeIn", "overtimeOut"]
      /*ro, opt, array, 可选项, subType:string*/
    }
  }
}
```

69.42.4 获取考勤模式参数能力

Request URL

GET /ISAPI/AccessControl/Configuration/attendanceMode/capabilities?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "AttendanceMode": {
    /*ro, req, object, 考勤模式*/
    "mode": {
      /*ro, opt, object, 考勤规则模式, desc:disable-禁用, manual-手动, auto-自动, manualAndAuto-手动和自动*/
      "@opt": ["disable", "manual", "auto", "manualAndAuto"]
      /*ro, opt, array, 可选项, subType:string*/
    },
    "attendanceStatusTime": {
      /*ro, opt, object, 考勤状态持续时间, desc:单位: s, mode为manual或manualAndAuto该字段生效*/
      "@min": 0,
      /*ro, opt, int, 最小值*/
      "@max": 0
      /*ro, opt, int, 最大值*/
    },
    "reqAttendanceStatus": {
      /*ro, opt, object, 是否必选考勤状态*/
      "@opt": [true, false]
      /*ro, opt, array, 可选项, subType:bool*/
    }
  }
}
```

69.42.5 配置考勤模式

Request URL

PUT /ISAPI/AccessControl/Configuration/attendanceMode?format=json

查询参数

无

请求报文

```
{
  "AttendanceMode": {
    /*req, object, 考勤模式*/
    "mode": "disable",
    /*opt, enum, 考勤规则模式, subType:string, [disable#禁用,manual#手动,auto#自动,manualAndAuto#手动和自动]*/
    "attendanceStatusTime": 0,
    /*opt, int, 考勤状态持续时间, desc:单位: s, mode为manual或manualAndAuto该字段生效*/
    "reqAttendanceStatus": true
    /*opt, bool, 是否必选考勤状态*/
  }
}
```

响应报文

```
{
  "statusCode": 1,
  /*ro, opt, int, 状态码, desc:无法用1表示时（1表示成功且无特殊状态）必须返回*/
  "statusString": "OK",
  /*ro, opt, string, 状态描述, range:[1,64], desc:无法用OK表示时（OK表示成功且无特殊状态）必须返回*/
  "subStatusCode": "ok",
  /*ro, opt, string, 子状态码, range:[1,64], desc:无法用ok表示（ok表示成功且无特殊状态）必须返回*/
  "errorCode": 1,
  /*ro, opt, int, 错误码, desc:当statusCode不为1时，与subStatusCode对应*/
  "errorMsg": "ok"
  /*ro, opt, string, 错误信息, desc:当statusCode不为1时，必须返回,解释信息在协议约束中，允许设备在后续的版本迭代中，进行优化丰富提升（不限制死）*/
}
```

69.42.6 获取考勤模式

Request URL

GET /ISAPI/AccessControl/Configuration/attendanceMode?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "AttendanceMode": {
    /*ro, req, object, 考勤模式*/
    "mode": "disable",
    /*ro, opt, enum, 考勤规则模式, subType:string, [disable#禁用,manual#手动,auto#自动,manualAndAuto#手动和自动]*/
    "attendanceStatusTime": 0,
    /*ro, opt, int, 考勤状态持续时间, desc:单位: s, mode为manual或manualAndAuto该字段生效*/
    "reqAttendanceStatus": true
    /*ro, opt, bool, 是否必选考勤状态*/
  }
}
```

69.42.7 获取考勤周计划能力

Request URL

GET /ISAPI/AccessControl/Attendance/weekPlan/capabilities?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "AttendanceWeekPlanCap": {
    /*ro, req, object, 考勤周计划能力节点*/
    "planNo": {
      /*ro, opt, object, 周计划编号*/
      "@min": 1,
      /*ro, opt, int, 最小值*/
      "@max": 16
      /*ro, opt, int, 最大值*/
    },
    "enable": {
      /*ro, opt, object, 使能*/
      "@opt": [true, false]
      /*ro, opt, array, 可选项, subType:bool*/
    },
    "WeekPlanCfg": {
      /*ro, opt, object, 周计划参数配置*/
      "maxSize": 5,
      /*ro, opt, int, 最大周计划数*/
      "id": {
        /*ro, opt, object, 时间段编号*/
        "@min": 1,
        /*ro, opt, int, 最小值*/
        "@max": 8
        /*ro, opt, int, 最大值*/
      },
      "week": {
        /*ro, opt, object, 星期, desc:Monday-星期一, Tuesday-星期二, Wednesday-星期三, Thursday-星期四, Friday-星期五, Saturday-星期六, Sunday-星期日*/
        "@opt": ["Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday"]
        /*ro, opt, array, 可选项, subType:string*/
      },
      "enable": {
        /*ro, opt, object, 使能*/
        "@opt": [true, false]
        /*ro, opt, array, 可选项, subType:bool*/
      },
      "TimeSegment": {
        /*ro, opt, object, 时间段*/
        "beginTime": "10:10:00",
        /*ro, opt, time, 开始时间点（设备本地时间）*/
        "endTime": "12:10:00",
        /*ro, opt, time, 结束时间点（设备本地时间）*/
        "validUnit": "second"
        /*ro, opt, enum, 时间点精度, subType:string, [hour#小时,minute#分钟,second#秒], desc:不返回该字段, 则代表精度为分钟*/
      }
    }
  }
}
```

69.42.8 配置考勤周计划

Request URL

PUT /ISAPI/AccessControl/Attendance/weekPlan/<PlanNo>?format=json

查询参数

参数名称	参数类型	描述
PlanNo	string	

请求报文

```
{
  "AttendanceWeekPlan": {
    /*req, object, 考勤周计划*/
    "enable": true,
    /*req, bool, 使能*/
    "WeekPlanCfg": [
      /*req, array, 周计划参数配置, subType:object*/
      {
        "id": 1,
        /*req, int, 时间段编号*/
        "week": "Monday",
        /*req, enum, 星期, subType:string, [Monday#星期一,Tuesday#星期二,Wednesday#星期三,Thursday#星期四,Friday#星期五,Saturday#星期六,Sunday#星期日]*/
        "enable": true,
        /*req, bool, 使能*/
        "TimeSegment": {
          /*req, object, 时间段*/
          "beginTime": "10:10:00",
          /*req, time, 开始时间点（设备本地时间）*/
          "endTime": "12:10:00"
          /*req, time, 结束时间点（设备本地时间）*/
        }
      }
    ]
  }
}
```

响应报文

```
{
  "statusCode": 1,
  /*ro, opt, int, 状态码, desc:无法用1表示时（1表示成功且无特殊状态）必须返回*/
  "statusString": "OK",
  /*ro, opt, string, 状态描述, range:[1,64], desc:无法用OK表示时（OK表示成功且无特殊状态）必须返回*/
  "subStatusCode": "ok",
  /*ro, opt, string, 子状态码, range:[1,64], desc:无法用ok表示（ok表示成功且无特殊状态）必须返回*/
  "errorCode": 1,
  /*ro, opt, int, 错误码, desc:当statusCode不为1时，与subStatusCode对应*/
  "errorMsg": "ok"
  /*ro, opt, string, 错误信息, desc:当statusCode不为1时，必须返回,解释信息在协议约束中，允许设备在后续的版本迭代中，进行优化丰富提升（不限制死）*/
}
```

69.42.9 获取考勤周计划

Request URL

GET /ISAPI/AccessControl/Attendance/weekPlan/<PlanNo>?format=json

查询参数

参数名称	参数类型	描述
PlanNo	string	--

请求报文

无

响应报文

```
{
  "AttendanceWeekPlan": {
    /*ro, req, object, 考勤周计划*/
    "enable": true,
    /*ro, req, bool, 使能*/
    "WeekPlanCfg": [
      /*ro, req, array, 周计划参数配置, subType:object*/
      {
        "id": 1,
        /*ro, req, int, 时间段编号*/
        "week": "Monday",
        /*ro, req, enum, 星期, subType:string, [Monday#星期一,Tuesday#星期二,Wednesday#星期三,Thursday#星期四,Friday#星期五,Saturday#星期六,Sunday#
星期日]*/
        "enable": true,
        /*ro, req, bool, 使能*/
        "TimeSegment": {
          /*ro, req, object, 时间段*/
          "beginTime": "10:10:00",
          /*ro, req, time, 开始时间点（设备本地时间）*/
          "endTime": "12:10:00"
          /*ro, req, time, 结束时间点（设备本地时间）*/
        }
      }
    ]
  }
}
```

69.42.10 配置单个考勤计划模板

Request URL

PUT /ISAPI/AccessControl/Attendance/planTemplate/<TemplateNo>?format=json

查询参数

参数名称	参数类型	描述
TemplateNo	string	--

请求报文

```
{
  "AttendancePlanTemplate": {
    /*req, object, 考勤计划模板*/
    "enable": true,
    /*req, bool, 是否启用, desc:true-启用, false-不启用*/
    "property": "check",
    /*req, enum, 考勤属性, subType:string, [check#上下班,break#休息,overtime#加班（每种属性模板只配置一个）]*/
    "templateName": "test",
    /*req, string, 模板名称, range:[1,32]*/
    "weekPlanNo": 1
    /*req, int, 周计划编号, range:[1,5]*/
  }
}
```

响应报文

```
{
  "statusCode": 1,
  /*ro, opt, int, 状态码, desc:无法用1表示时（1表示成功且无特殊状态）必须返回*/
  "statusString": "OK",
  /*ro, opt, string, 状态描述, range:[1,64], desc:无法用OK表示时（OK表示成功且无特殊状态）必须返回*/
  "subStatusCode": "ok",
  /*ro, opt, string, 子状态码, range:[1,64], desc:无法用ok表示（ok表示成功且无特殊状态）必须返回*/
  "errorCode": 1,
  /*ro, opt, int, 错误码, desc:当statusCode不为1时，与subStatusCode对应*/
  "errorMsg": "ok"
  /*ro, opt, string, 错误信息, desc:当statusCode不为1时，必须返回,解释信息在协议约束中，允许设备在后续的版本迭代中，进行优化丰富提升（不限制死）*/
}
```

69.42.11 获取单个考勤计划模板

Request URL

GET /ISAPI/AccessControl/Attendance/planTemplate/<TemplateNo>?format=json

查询参数

参数名称	参数类型	描述
TemplateNo	string	--

请求报文

无

响应报文

```
{
  "AttendancePlanTemplate": {
    /*ro, req, object, 考勤计划模板*/
    "enable": true,
    /*ro, req, bool, 是否启用, desc:true-启用, false-不启用*/
    "property": "check",
    /*ro, req, enum, 考勤属性, subType:string, [check#上下班,break#休息,overtime#加班（每种属性模板只配置一个）]*/
    "templateName": "test",
    /*ro, req, string, 模板名称, range:[1,32]*/
    "weekPlanNo": 1
    /*ro, req, int, 周计划编号, range:[1,5]*/
  }
}
```

69.42.12 获取单个按键考勤参数

Request URL

GET /ISAPI/AccessControl/keyCfg/<keyID>/attendance?format=json

查询参数

参数名称	参数类型	描述
keyID	string	

请求报文

无

响应报文

```
{
  "Attendance": {
    /*ro, req, object, 考勤*/
    "enable": true,
    /*ro, req, bool, 类型boolean*/
    "attendanceStatus": "checkIn",
    /*ro, opt, enum, 考勤状态, subType:string, [checkIn#上班,checkOut#下班,breakOut#开始休息,breakIn#结束休息,overtimeIn#开始加班,overtimeOut#结束加班]*/
    "label": "test"
    /*ro, opt, string, 自定义名称, range:[1,32]*/
  }
}
```

69.42.13 配置单个按键考勤参数

Request URL

PUT /ISAPI/AccessControl/keyCfg/<keyID>/attendance?format=json

查询参数

参数名称	参数类型	描述
keyID	string	

请求报文

```
{
  "Attendance": {
    /*req, object, 考勤*/
    "enable": true,
    /*req, bool, 类型boolean*/
    "attendanceStatus": "checkIn",
    /*opt, enum, 考勤状态, subType:string, [checkIn#上班,checkOut#下班,breakOut#开始休息,breakIn#结束休息,overtimeIn#开始加班,overtimeOut#结束加班]*/
    "label": "test"
    /*opt, string, 自定义名称, range:[1,32]*/
  }
}
```

响应报文

```
{
  "statusCode": 1,
  /*ro, opt, int, 状态码, desc:无法用1表示时（1表示成功且无特殊状态）必须返回*/
  "statusString": "OK",
  /*ro, opt, string, 状态描述, range:[1,64], desc:无法用OK表示时（OK表示成功且无特殊状态）必须返回*/
  "subStatusCode": "ok",
  /*ro, opt, string, 子状态码, range:[1,64], desc:无法用ok表示（ok表示成功且无特殊状态）必须返回*/
  "errorCode": 1,
  /*ro, opt, int, 错误码, desc:当statusCode不为1时，与subStatusCode对应*/
  "errorMsg": "ok"
  /*ro, opt, string, 错误信息, desc:当statusCode不为1时，必须返回,解释信息在协议约束中，允许设备在后续的版本迭代中，进行优化丰富提升（不限制死）*/
}
```

69.43 多重认证

69.43.1 清空群组参数

Request URL

PUT /ISAPI/AccessControl/ClearGroupCfg?format=json

查询参数

无

请求报文

```
{
  "ClearGroupCfg": {
    /*opt, object, 群组参数*/
    "ClearFlags": {
      /*opt, object, 清空标记*/
      "groupCfg": true
      /*req, bool, 群组参数*/
    }
  }
}
```

响应报文

```
{
  "requestURL": "test",
  /*ro, opt, string, URI*/
  "statusCode": "test",
  /*ro, opt, string, 状态码*/
  "statusString": "test",
  /*ro, opt, string, 状态描述*/
  "subStatusCode": "test",
  /*ro, opt, string, 子状态码*/
  "errorCode": 1,
  /*ro, req, int, 错误码*/
  "errorMsg": "ok"
  /*ro, req, string, 错误信息*/
}
```

69.43.2 获取清空群组参数能力

Request URL

GET /ISAPI/AccessControl/ClearGroupCfg/capabilities?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "ClearGroupCfg": {
    /*ro, opt, object, 清空群组参数配置能力*/
    "ClearFlags": {
      /*ro, opt, object, 清空标记*/
      "groupCfg": "true,false"
      /*ro, req, string, 群组参数*/
    }
  }
}
```

69.43.3 获取群组参数

Request URL

GET /ISAPI/AccessControl/GroupCfg/<groupID>?format=json

查询参数

参数名称	参数类型	描述
groupID	string	代表群组编号，从1开始

请求报文

无

响应报文

```
{
  "GroupCfg": {
    /*ro, opt, object, 群组参数*/
    "enable": true,
    /*ro, req, bool, 是否启用*/
    "ValidPeriodCfg": {
      /*ro, opt, object, 群组有效期参数*/
      "enable": true,
      /*ro, req, bool, 使能有效期*/
      "beginTime": "1970-01-01T00:00:00+08:00",
      /*ro, req, datetime, 有效期起始时间 (UTC时间) */
      "endTime": "1970-01-01T00:00:00+08:00"
      /*ro, req, datetime, 有效期结束时间 (UTC时间) */
    },
    "groupName": "test"
    /*ro, opt, string, 群组名称*/
  }
}
```

69.43.4 配置群组参数

Request URL

PUT /ISAPI/AccessControl/GroupCfg/<groupID>?format=json

查询参数

参数名称	参数类型	描述
groupID	string	代表群组编号，从1开始

请求报文

```
{
  "GroupCfg": {
    /*opt, object, 群组参数*/
    "enable": true,
    /*req, bool, 是否启用*/
    "ValidPeriodCfg": {
      /*opt, object, 群组有效期参数*/
      "enable": true,
      /*req, bool, 使能有效期*/
      "beginTime": "1970-01-01T00:00:00+08:00",
      /*req, datetime, 有效期起始时间（UTC时间）*/
      "endTime": "1970-01-01T00:00:00+08:00"
      /*req, datetime, 有效期结束时间（UTC时间）*/
    },
    "groupName": "test"
    /*opt, string, 群组名称*/
  }
}
```

响应报文

```
{
  "requestURL": "test",
  /*ro, opt, string, URI*/
  "statusCode": "test",
  /*ro, opt, string, 状态码*/
  "statusString": "test",
  /*ro, opt, string, 状态描述*/
  "subStatusCode": "test",
  /*ro, opt, string, 子状态码*/
  "errorCode": 1,
  /*ro, req, int, 错误码*/
  "errorMsg": "ok"
  /*ro, req, string, 错误信息*/
}
```

69.43.5 获取多重认证模式参数配置能力

Request URL

GET /ISAPI/AccessControl/MultiCardCfg/capabilities?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "MultiCardCfg": {
    /*ro, opt, object, 多重认证参数*/
    "doorNo": {
      /*ro, opt, object, 门编号*/
      "@min": 1,
      /*ro, opt, int, 最小值*/
      "@max": 256
      /*ro, opt, int, 最大值*/
    },
    "enable": "true,false",
    /*ro, req, string, 是否启用多重认证功能*/
    "swipeIntervalTimeout": {
      /*ro, opt, object, 刷卡（认证）间隔超时时间*/
      "@min": 1,
      /*ro, opt, int, 最小值*/
      "@max": 255
      /*ro, opt, int, 最大值*/
    },
  },
  "GroupCfg": {
    /*ro, opt, object, 多重认证参数*/
    "maxSize": 20,
    /*ro, opt, int, 最大值*/
    "id": {
      /*ro, opt, object, 多重认证编号*/
      "@min": 1,
      /*ro, opt, int, 最小值*/
      "@max": 20
      /*ro, opt, int, 最大值*/
    },
    "enable": "true,false",
    /*ro, opt, string, 是否启用该多重认证*/
    "enableOfflineVerifyMode": "true,false",
    /*ro, opt, string, 是否启用主机离线时验证方式（超级密码代替远程开门）*/
    "templateNo": {
      /*ro, opt, object, 启用多重认证的计划模板编号*/
      "@min": 1,
      /*ro, opt, int, 最小值*/
      "@max": 20
      /*ro, opt, int, 最大值*/
    },
  },
  "GroupCombination": {
    /*ro, opt, object, 多重认证组合参数*/
    "maxSize": 8,
    /*ro, opt, int, 最大值*/
    "enable": "true,false",
    /*ro, opt, string, 是否启用该多重认证组合*/
    "memberNum": {
      /*ro, opt, object, 刷卡成员数量*/
      "@min": 1,
      /*ro, opt, int, 最小值*/
      "@max": 20
      /*ro, opt, int, 最大值*/
    },
  },
  "sequenceNo": {
    /*ro, opt, object, 多重认证组合刷卡次序号*/
    "@min": 1,
    /*ro, opt, int, 最小值*/
    "@max": 8
    /*ro, opt, int, 最大值*/
  },
  "groupNo": {
    /*ro, opt, object, 群组编号*/
    "@min": 1,
    /*ro, opt, int, 最小值*/
    "@max": 20
    /*ro, opt, int, 最大值*/
  },
}
}
```

69.43.6 配置多重认证模式参数

Request URL

PUT /ISAPI/AccessControl/MultiCardCfg/<doorID>?format=json

查询参数

参数名称	参数类型	描述
doorID	string	门编号

请求报文

```
{
  "MultiCardCfg": {
    /*opt, object, 多重卡参数*/
    "enable": true,
    /*req, bool, 是否启用多重认证功能*/
    "swipeIntervalTimeout": 10,
    /*opt, int, 刷卡（认证）间隔超时时间*/
    "GroupCfg": [
      /*opt, array, 多重认证参数, subType:object*/
      {
        "id": 1,
        /*opt, int, 多重认证编号*/
        "enable": true,
        /*opt, bool, 是否启用该多重认证*/
        "enableOfflineVerifyMode": true,
        /*opt, bool, 是否启用主机离线时验证方式（超级密码代替远程开门）*/
        "templateNo": 1,
        /*opt, int, 启用多重认证的计划模板编号*/
        "GroupCombination": [
          /*opt, array, 多重认证组合参数, subType:object*/
          {
            "enable": true,
            /*opt, bool, 是否启用该多重认证组合*/
            "memberNum": 3,
            /*opt, int, 刷卡成员数量*/
            "sequenceNo": 1,
            /*opt, int, 多重认证组合刷卡次序号*/
            "groupNo": 1
            /*opt, int, 群组编号, desc:65535表示远程开门, 65534表示超级密码*/
          }
        ]
      }
    ]
  }
}
```

响应报文

```
{
  "requestURL": "test",
  /*ro, opt, string, URI*/
  "statusCode": "test",
  /*ro, opt, string, 状态码*/
  "statusString": "test",
  /*ro, opt, string, 状态描述*/
  "subStatusCode": "test",
  /*ro, opt, string, 子状态码*/
  "errorCode": 1,
  /*ro, req, int, 错误码*/
  "errorMsg": "ok",
  /*ro, req, string, 错误信息*/
}
```

69.43.7 获取多重认证模式参数

Request URL

GET /ISAPI/AccessControl/MultiCardCfg/<doorID>?format=json

查询参数

参数名称	参数类型	描述
doorID	string	门编号

请求报文

无

响应报文

```
{
  "MultiCardCfg": {
    /*ro, opt, object, 多重卡参数*/
    "enable": true,
    /*ro, req, bool, 是否启用多重认证功能*/
    "swipeIntervalTimeout": 10,
    /*ro, opt, int, 刷卡（认证）间隔超时时间*/
    "GroupCfg": [
      /*ro, opt, array, 多重认证参数, subType:object*/
      {
        "id": 1,
        /*ro, opt, int, 多重认证编号*/
        "enable": true,
        /*ro, opt, bool, 是否启用该多重认证*/
        "enableOfflineVerifyMode": true,
        /*ro, opt, bool, 是否启用主机离线时验证方式（超级密码代替远程开门）*/
        "templateNo": 1,
        /*ro, opt, int, 启用多重认证的计划模板编号*/
        "GroupCombination": [
          /*ro, opt, array, 多重认证组合参数, subType:object*/
          {
            "enable": true,
            /*ro, opt, bool, 是否启用该多重认证组合*/
            "memberNum": 3,
            /*ro, opt, int, 刷卡成员数量*/
            "sequenceNo": 1,
            /*ro, opt, int, 多重认证组合刷卡次序号*/
            "groupNo": 1
            /*ro, opt, int, 群组编号, desc:65535表示远程开门, 65534表示超级密码*/
          }
        ]
      }
    ]
  }
}
```

69.43.8 获取群组参数配置能力

Request URL

GET /ISAPI/AccessControl/GroupCfg/capabilities?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "GroupCfg": {
    /*ro, opt, object, 群组参数*/
    "groupNo": {
      /*ro, opt, object, 群组编号*/
      "@min": 1,
      /*ro, opt, int, 最小值*/
      "@max": 1
      /*ro, opt, int, 最大值*/
    },
    "enable": "true,false",
    /*ro, req, string, 是否启用*/
    "ValidPeriodCfg": {
      /*ro, req, object, 使能有效期参数*/
      "enable": "true,false",
      /*ro, req, string, 使能有效期*/
      "beginTime": {
        /*ro, req, object, 有效期起始时间（UTC时间）*/
        "@min": 1,
        /*ro, opt, int, 最小值*/
        "@max": 32
        /*ro, opt, int, 最大值*/
      },
      "endTime": {
        /*ro, req, object, 有效期结束时间（UTC时间）*/
        "@min": 1,
        /*ro, opt, int, 最小值*/
        "@max": 32
        /*ro, opt, int, 最大值*/
      }
    },
    "groupName": {
      /*ro, opt, object, 群组名称*/
      "@min": 1,
      /*ro, opt, int, 最小值*/
      "@max": 32
      /*ro, opt, int, 最大值*/
    }
  }
}
```

69.44 门禁蓝牙管理

69.44.1 配置蓝牙加密信息

Request URL

PUT /ISAPI/AccessControl/bluetoothEncryptionInfo?format=json

查询参数

无

请求报文

```
{
  "encryptType": "AES128_CBC",
  /*req, enum, 加密类型, subType:string, [AES128_CBC#AES128_CBC-AES128加密]*/
  "authData": "00112233445566778899aabbccddeeff",
  /*req, string, 加密使用的认证信息, range:[1,32],
  desc:1.32字节16进制数据;
  2.该字段需要进行敏感信息加密*/
  "vector": "00112233445566778899aabbccddeeff"
  /*req, string, 加密使用的初始化向量, range:[1,32],
  desc:1.32字节16进制数据;
  2.该字段需要进行敏感信息加密*/
}
```

响应报文

```
{
  "statusCode": 1,
  /*ro, opt, int, 状态码, desc:无法用1表示时（1表示成功且无特殊状态）必须返回*/
  "statusString": "OK",
  /*ro, opt, string, 状态描述, range:[1,64], desc:无法用OK表示时（OK表示成功且无特殊状态）必须返回*/
  "subStatusCode": "ok",
  /*ro, opt, string, 子状态码, range:[1,64], desc:无法用ok表示（ok表示成功且无特殊状态）必须返回*/
  "errorCode": 1,
  /*ro, opt, int, 错误码, desc:当statusCode不为1时，与subStatusCode对应*/
  "errorMsg": "ok"
  /*ro, opt, string, 错误信息, desc:当statusCode不为1时，必须返回,解释信息在协议约束中，允许设备在后续的版本迭代中，进行优化丰富提升（不限制死）*/
}
```

69.44.2 获取蓝牙加密信息

Request URL

GET /ISAPI/AccessControl/bluetoothEncryptionInfo?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "encryptType": "AES128_CBC",
  /*ro, req, enum, 加密类型, subType:string, [AES128_CBC#AES128_CBC-AES128加密]*/
  "authData": "00112233445566778899aabbccddeeff",
  /*ro, req, string, 加密使用的认证信息, range:[1,32],
  desc:1.32字节16进制数据:
  2.该字段需要进行敏感信息加密*/
  "vector": "00112233445566778899aabbccddeeff",
  /*ro, req, string, 加密使用的初始化向量, range:[1,32],
  desc:1.32字节16进制数据:
  2.该字段需要进行敏感信息加密*/
  "loopCount": 100,
  /*ro, req, int, 生成密钥的循环次数*/
  "employeeNo": "123"
  /*ro, opt, string, 工号（人员ID），desc:该字段需要进行敏感信息加密*/
}
```

69.44.3 获取蓝牙加密信息能力

Request URL

GET /ISAPI/AccessControl/bluetoothEncryptionInfo/capabilities?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "encryptType": {
    /*ro, opt, object, 加密类型*/
    "@opt": ["AES128_CBC"]
    /*ro, opt, array, 可选项, subType:string, desc: AES128_CBC: AES128_CBC-AES128加密, CBC模式*/
  },
  "authData": {
    /*ro, opt, object, 加密使用的认证信息, desc: 32字节16进制数据*/
    "@min": 1,
    /*ro, opt, int, 最小值*/
    "@max": 32
    /*ro, opt, int, 最大值*/
  },
  "vector": {
    /*ro, opt, object, 加密使用的初始化向量, desc: 32字节16进制数据*/
    "@min": 1,
    /*ro, opt, int, 最小值*/
    "@max": 32
    /*ro, opt, int, 最大值*/
  },
  "loopCount": 100,
  /*ro, opt, int, 生成密钥的循环次数*/
  "employeeNo": {
    /*ro, opt, object, 工号（人员ID）*/
    "@min": 1,
    /*ro, opt, int, 最小值*/
    "@max": 32
    /*ro, opt, int, 最大值*/
  }
}
```

69.44.4 配置蓝牙加密版本

Request URL

PUT /ISAPI/AccessControl/bluetoothEncryptionVersion?format=json

查询参数

无

请求报文

```
{
  "version": "V2.0"
  /*req, enum, 加密版本, subType:string, [V2.0#新升级密钥版本（默认）, V1.0#原固定密钥版本]*/
}
```

响应报文

```
{
  "statusCode": 1,
  /*ro, opt, int, 状态码, desc: 无法用1表示时（1表示成功且无特殊状态）必须返回*/
  "statusString": "OK",
  /*ro, opt, string, 状态描述, range: [1, 64], desc: 无法用OK表示时（OK表示成功且无特殊状态）必须返回*/
  "subStatusCode": "ok",
  /*ro, opt, string, 子状态码, range: [1, 64], desc: 无法用ok表示（ok表示成功且无特殊状态）必须返回*/
  "errorCode": 1,
  /*ro, opt, int, 错误码, desc: 当statusCode不为1时，与subStatusCode对应*/
  "errorMsg": "ok"
  /*ro, opt, string, 错误信息, desc: 当statusCode不为1时，必须返回, 解释信息在协议约束中，允许设备在后续的版本迭代中，进行优化丰富提升（不限制死）*/
}
```

69.44.5 获取蓝牙加密版本

Request URL

GET /ISAPI/AccessControl/bluetoothEncryptionVersion?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "version": "V2.0"
  /*ro, req, enum, 加密版本, subType:string, [V2.0#新升级密钥版本（默认）,V1.0#原固定密钥版本]*/
}
```

69.44.6 获取蓝牙加密版本能力

Request URL

GET /ISAPI/AccessControl/bluetoothEncryptionVersion/capabilities?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "version": {
    /*ro, opt, object, 加密版本*/
    "@opt": ["V1.0", "V2.0"]
    /*ro, opt, array, 可选项, subType:string, desc:V2.0-新升级密钥版本（默认）, V1.0-原固定密钥版本*/
  }
}
```

69.44.7 配置门禁蓝牙参数

Request URL

PUT /ISAPI/AccessControl/bluetooth?format=json

查询参数

无

请求报文

```
{
  "openDoorEnabled": true
  /*opt, bool, 蓝牙开门使能*/
}
```

响应报文

```
{
  "statusCode": 1,
  /*ro, opt, int, 状态码, desc:无法用1表示时（1表示成功且无特殊状态）必须返回*/
  "statusString": "OK",
  /*ro, opt, string, 状态描述, range:[1,64], desc:无法用OK表示时（OK表示成功且无特殊状态）必须返回*/
  "subStatusCode": "ok",
  /*ro, opt, string, 子状态码, range:[1,64], desc:无法用ok表示（ok表示成功且无特殊状态）必须返回*/
  "errorCode": 1,
  /*ro, opt, int, 错误码, desc:当statusCode不为1时, 与subStatusCode对应*/
  "errorMsg": "ok"
  /*ro, opt, string, 错误信息, desc:当statusCode不为1时, 必须返回,解释信息在协议约束中, 允许设备在后续的版本迭代中, 进行优化丰富提升（不限制死）*/
}
```

69.44.8 获取门禁蓝牙参数

Request URL

GET /ISAPI/AccessControl/bluetooth?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "openDoorEnabled": true
  /*ro, opt, bool, 蓝牙开门使能*/
}
```

69.44.9 获取门禁蓝牙参数能力

Request URL

GET /ISAPI/AccessControl/bluetooth/capabilities?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "openDoorEnabled": {
    /*ro, opt, object, 蓝牙开门使能*/
    "@opt": [true, false]
    /*ro, req, array, 取值范围, subType:bool*/
  }
}
```

69.45 门禁事件管理(S)

69.45.1 配置清空事件及卡号联动参数

Request URL

PUT /ISAPI/AccessControl/ClearEventCardLinkageCfg?format=json

查询参数

无

请求报文

```
{
  "ClearEventCardLinkageCfg": {
    /*req, object, 清空事件及卡号联动参数*/
    "ClearFlags": {
      /*opt, object, 清空标识*/
      "eventCardLinkage": true
      /*req, bool, 事件及卡号联动参数*/
    }
  }
}
```

响应报文

```
{
  "statusCode": 1,
  /*ro, opt, int, 状态码, desc:无法用1表示时（1表示成功且无特殊状态）必须返回*/
  "statusString": "OK",
  /*ro, opt, string, 状态描述, range:[1,64], desc:无法用OK表示时（OK表示成功且无特殊状态）必须返回*/
  "subStatusCode": "ok",
  /*ro, opt, string, 子状态码, range:[1,64], desc:无法用ok表示（ok表示成功且无特殊状态）必须返回*/
  "errorCode": 1,
  /*ro, opt, int, 错误码, desc:当statusCode不为1时，与subStatusCode对应*/
  "errorMsg": "ok"
  /*ro, opt, string, 错误信息, desc:当statusCode不为1时，必须返回,解释信息在协议约束中，允许设备在后续的版本迭代中，进行优化丰富提升（不限制死）*/
}
```

69.45.2 获取门禁事件存储参数能力

Request URL

GET /ISAPI/AccessControl/AcsEvent/StorageCfg/capabilities?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "EventStorageCfgCap": {
    /*ro, req, object, 事件存储配置能力*/
    "mode": {
      /*ro, req, object, 事件存储方式, desc:regular-定期删除旧事件, time-按指定时间删除旧事件 cycle-循环覆盖*/
      "@opt": ["regular", "time", "cycle"]
      /*ro, opt, array, 可选项, subType:string*/
    },
    "checkTime": {
      /*ro, opt, object, 检查时间点, desc:依赖time模式, 最大长度为32, 删除此时间点之前的事件*/
      "@min": 0,
      /*ro, opt, int, 最小值*/
      "@max": 0
      /*ro, opt, int, 最大值*/
    },
    "period": {
      /*ro, opt, object, 删除旧事件的周期, desc:依赖regular模式, 单位 分钟*/
      "@min": 10,
      /*ro, opt, int, 最小值*/
      "@max": 10
      /*ro, opt, int, 最大值*/
    }
  }
}
```

69.45.3 配置事件及卡号联动参数

Request URL

PUT /ISAPI/AccessControl/EventCardLinkageCfg/<ACEID>?format=json

查询参数

参数名称	参数类型	描述
ACEID	string	

请求报文

```
{
  "EventCardLinkageCfg": {
    /*req, object, 事件及卡号联动参数*/
    "proMode": "event",
    /*req, enum, 联动方式, subType:string, [event#事件, card#卡号, mac#MAC地址, employee#工号（人员ID）]*/
    "EventLinkageInfo": {
      /*opt, object, 事件及卡号联动信息, desc:proMode为event时有效*/
      "mainEventType": 0,
      /*opt, enum, 事件主类型, subType:int, [0#设备事件, 1#报警输入事件, 2#门事件, 3#读卡器事件]*/
      "subEventType": 54
      /*opt, int, 事件次类型, desc:54#消防矩阵事件(EVENT_ACS_FIREMATRIX_EVENT)*/
    },
    "CardNoLinkageInfo": {
      /*opt, object, 卡号联动信息, desc:proMode为card时有效*/
      "cardNo": "test"
      /*opt, string, 卡号*/
    },
    "MacAddrLinkageInfo": {
      /*opt, object, 物理MAC地址连接信息, desc:proMode为mac时有效*/
      "MACAddr": "test"
      /*opt, string, 物理MAC地址*/
    },
    "EmployeeInfo": {
      /*opt, object, 工号信息, desc:proMode为employee时有效*/
      "employeeNo": "test"
      /*opt, string, 工号（人员ID）*/
    },
    "eventSourceID": 1,
    /*opt, int, 事件源ID, desc:proMode为event时有效, 当主类型为设备事件（mainEventType为0）时无效; 当主类型是门事件（mainEventType为2）时为门编号; 当主类型为读卡器事件（mainEventType为3）时, 为读卡器ID; 当为报警输入事件（mainEventType为1）时为防区报警输入ID或事件报警输入ID。65535表示联动全部*/
    "alarmout": [1, 3, 5],
    /*opt, array, 关联的报警输出号, subType:int, desc:[1,3,5]代表关联报警输出1、报警输出3、报警输出5、用于本地报警输出*/
    "ReaderAlarmout": [
      /*opt, array, 联动读卡器的报警输出号, subType:object, desc:用于支持多路读卡器报警输出 当前用于智能主机*/
    ]
  }
}
```

```
/*opt, array, 联动读卡器的报警输出号, subType:int, desc:[1,3,5]代表关联报警输出1、报警输出3、报警输出5*/
{
    "readerID": 1,
    /*opt, int, 读卡器编号*/
    "alarmOut": [1, 3, 5],
    /*opt, array, 关联的报警输出号, subType:int, desc:[1,3,5]代表关联报警输出1、报警输出3、报警输出5*/
    "alarmOutClose": [1, 3, 5]
    /*opt, array, 关联报警输出关闭, subType:int, desc:[1,3,5]代表关联报警输出1、报警输出3、报警输出5的关闭*/
}
},
"openDoor": [1, 3, 5],
/*opt, array, 是否联动开门, subType:int, desc:[1,3,5]代表联动门1、门3、门5开门*/
"closeDoor": [1, 3, 5],
/*opt, array, 是否联动关门, subType:int, desc:[1,3,5]代表联动门1、门3、门5关门*/
"alwaysOpen": [1, 3, 5],
/*opt, array, 是否联动常开, subType:int, desc:[1,3,5]代表联动门1、门3、门5常开*/
"alwaysClose": [1, 3, 5],
/*opt, array, 是否联动常闭, subType:int, desc:[1,3,5]代表联动门1、门3、门5常闭*/
"mainDevBuzzer": true,
/*opt, bool, 主机蜂鸣器, desc:false-不联动输出, true-联动输出*/
"capturePic": true,
/*opt, bool, 是否联动抓拍, desc:false-不联动抓拍, true-联动抓拍, desc:主要用于一体机, 仅支持一路读卡器抓拍*/
"readerCapturePic": [1, 2],
/*opt, array, 联动读卡器抓拍, subType:int, range:[0,8], desc:[1,2]代表联动读卡器1,2抓拍。用于支持多路读卡器抓拍, 当前适用于智能主机*/
"recordVideo": true,
/*opt, bool, 是否联动录像, desc:false-不联动录像, true-联动录像, desc:主要用于一体机, 仅支持一路读卡器录像*/
"readerRecordVideo": [1, 2],
/*opt, array, 联动读卡器录像, subType:int, range:[0,8], desc:[1,2]代表联动读卡器1,2录像。用于支持多路读卡器录像, 当前适用于智能主机*/
"mainDevStopBuzzer": true,
/*opt, bool, 主机停止蜂鸣, desc:false-不联动停止, true-联动停止*/
"audioSourceType": "TTS",
/*opt, enum, 音频源类型, subType:string, [TTS#TTS语音, customFile#自定义语音, none#取消联动]*/
"audioDisplayTTSLanguage": "SimChinese",
/*opt, enum, 联动语音播报TTS语言类型, subType:string, [SimChinese#简体中文, English#英文], dep:and, {$.EventCardLinkageCfg.audioSourceType, eq, TTS},
desc:默认简体中文*/
"audioDisplayTTS": "test",
/*opt, string, 联动语音播报TTS内容, range:[1,64], dep:and, {$.EventCardLinkageCfg.audioSourceType, eq, TTS}*/
"customAudioID": 1,
/*opt, int, 联动的自定义音频文件ID, range:[1,65535], dep:and, {$.EventCardLinkageCfg.audioSourceType, eq, customFile}*/
"audioDisplayMode": "close",
/*opt, enum, 联动语音播放模式, subType:string, [close#关闭, single#单次播放, loop#循环播放]*/
"readerBuzzer": [1, 3, 5],
/*opt, array, 联动读卡器蜂鸣器, subType:int, desc:[1,3,5]代表联动读卡器1、读卡器3、读卡器5的蜂鸣器*/
"alarmOutClose": [1, 3, 5],
/*opt, array, 关联报警输出关闭, subType:int, desc:[1,3,5]代表关联报警输出1、报警输出3、报警输出5的关闭*/
"alarmInSetup": [1, 3, 5],
/*opt, array, 关联防区布防, subType:int, desc:[1,3,5]代表关联防区1、防区3、防区5布防*/
"alarmInClose": [1, 3, 5],
/*opt, array, 关联防区撤防, subType:int, desc:[1,3,5]代表关联防区1、防区3、防区5撤防*/
"readerStopBuzzer": [1, 3, 5],
/*opt, array, 联动读卡器停止蜂鸣, subType:int, desc:[1,3,5]代表联动读卡器1、读卡器3、读卡器5停止蜂鸣*/
"duration": 30,
/*opt, int, 联动输出信号持续时间, range:[0,30], unit:s, unitType:时间, desc:默认持续时间为15*/
"eventCardLinkageName": "test",
/*opt, string, 联动配置名称, range:[0,128]*/
"FireMatrixEventSourceList": [
/*opt, array, 消防矩阵事件来源列表, subType:object, dep:and, {$.EventCardLinkageCfg.EventLinkageInfo.subEventType, eq, 52}, desc:事件类型为消防矩阵事件
时才会返回该节点*/
{
    "IoTChannelID": 1,
    /*req, int, IoT通道号, desc:消防矩阵添加到管理机上后对应的IoT通道号*/
    "channels": [1, 2, 3]
    /*opt, array, 消防矩阵上的通道号数组, subType:int, range:[0,20], desc:一般接入探测器等设备*/
}
},
"Operations": {
/*opt, object, 广播联动操作列表, desc:web或应用下发给管理机的配置,支持不同终端播放不同音频。当前仅web和管理机支持*/
"eventSourceIDList": [1, 2],
/*req, array, 产生报警事件的终端ID列表, subType:int, desc:支持多个报警源关联多个终端*/
"TerminalsInfo": [
/*opt, array, 终端列表, subType:object, range:[0,1024]*/
{
    "terminalID": 1,
    /*req, int, 终端ID(由平台管理保证唯一性), range:[1,1024]*/
    "ACPChannels": [1, 2]
    /*opt, array, 终端设备下音频接入点通道号, subType:int, range:[0,4], desc:无该节点表示无音频接入点通道*/
}
},
"audioSource": "audioFile",
/*req, enum, 音频来源, subType:string, [audioFile#音频文件, speechSynthesis#语音合成]*/
"materialIdList": [1, 2],
/*opt, array, 素材ID列表, subType:int, range:[1,128], desc:音源类型为文件时必须有效*/
"speechSynthesisContent": "test",
/*opt, string, 语音合成内容, range:[0,4096], desc:当音频来源为语音合成时必须有效*/
"voiceType": "male",
/*opt, enum, 语音类型, subType:string, [male#男声, female#女声], desc:默认男声*/
"audioLevel": 0,
/*req, int, 音频播放级别(对于一个终端内多个音频在同一时间段内播放, range:[0,15], step:1, desc:数字越高级别越大,默认为0*/
"audioVolume": 5,
/*opt, int, 播放音量, range:[0,10], desc:表示音量档位*/
"playMode": "order",
/*opt, enum, 播放模式, subType:string, [order#顺序播放一遍, duration#指定播放持续时间], desc:默认顺序播放*/
"duration": 30
/*opt, int, 播放持续时间, range:[0,3000], unit:s, unitType:时间*/
}
```

```
}  
}  
}
```

响应报文

```
{  
  "requestURL": "test",  
  /*ro, opt, string, URI*/  
  "statusCode": "test",  
  /*ro, opt, string, 状态码*/  
  "statusString": "test",  
  /*ro, opt, string, 状态描述*/  
  "subStatusCode": "test",  
  /*ro, opt, string, 子状态码*/  
  "errorCode": 1,  
  /*ro, req, int, 错误码*/  
  "errorMsg": "ok",  
  /*ro, req, string, 错误信息*/  
}
```

69.45.4 查询门禁设备事件总条数

Request URL

POST /ISAPI/AccessControl/AcsEventTotalNum?format=json

查询参数

无

请求报文

```
{  
  "AcsEventTotalNumCond": {  
    /*req, object, 门禁设备事件总条数*/  
    "major": 1,  
    /*req, int, 主类型, desc:参考事件上传宏定义, 0-全部 (此处传递的是10进制数据, 不能传递16进制, 如传递1代表0x1 (报警主类型)) */  
    "minor": 1024,  
    /*req, int, 次类型, step:1,  
    desc:参考事件上传宏定义, 0-全部 (此处传递的是10进制数据, 不能传递16进制, 如传递1024代表0x400 (防区短路报警次类型)) */  
    当主次类型均为0时, 表示查询所有事件*/  
    "startTime": "1970-01-01+08:00",  
    /*opt, date, 开始时间 (UTC时间) */  
    "endTime": "1970-01-01+08:00",  
    /*opt, date, 结束时间 (UTC时间) */  
    "cardNo": "test",  
    /*opt, string, 卡号*/  
    "name": "test",  
    /*opt, string, 持卡人姓名*/  
    "picEnable": true,  
    /*opt, bool, 是否带图片, desc:false-不带图片, true-带图片*/  
    "beginSerialNo": 1,  
    /*opt, int, 起始流水号*/  
    "endSerialNo": 100,  
    /*opt, int, 结束流水号*/  
    "employeeNoString": "test",  
    /*opt, string, 工号 (人员ID), range:[1,32]*/  
  }  
}
```

响应报文

```
{  
  "AcsEventTotalNum": {  
    /*ro, req, object, 门禁设备事件总条数*/  
    "totalNum": 1,  
    /*ro, req, int, 满足搜索条件的事件总条数*/  
    "existedEventNum": 1  
    /*ro, opt, int, 已存在的事件总条数, desc:包含门禁事件、测温事件、二维码事件、身份证事件等设备内所有事件数目, 不随搜索条件变动*/  
  }  
}
```

69.45.5 获取事件及卡号联动参数

Request URL

GET /ISAPI/AccessControl/EventCardLinkageCfg/<ACEID>?format=json

查询参数

参数名称	参数类型	描述
ACEID	string	

请求报文

无

响应报文

```
{
  "EventCardLinkageCfg": {
    /*ro, req, object, 事件及卡号联动参数*/
    "proMode": "event",
    /*ro, req, enum, 联动方式, subType:string, [event#事件,card#卡号,mac#MAC地址,employee#工号（人员ID）]*/
    "EventLinkageInfo": {
      /*ro, opt, object, 事件及卡号联动信息, desc:proMode为event时有效*/
      "mainEventType": 0,
      /*ro, opt, enum, 事件主类型, subType:int, [0#设备事件,1#报警输入事件,2#门事件,3#读卡器事件]*/
      "subEventType": 54
      /*ro, opt, int, 事件次类型, desc:54#消防矩阵事件(EVENT_ACS_FIREMATRIX_EVENT)*/
    },
    "CardNoLinkageInfo": {
      /*ro, opt, object, 卡号联动信息, desc:proMode为card时有效*/
      "cardNo": "test"
      /*ro, opt, string, 卡号*/
    },
    "MacAddrLinkageInfo": {
      /*ro, opt, object, 物理MAC地址连接信息, desc:proMode为mac时有效*/
      "MACAddr": "test"
      /*ro, opt, string, 物理MAC地址*/
    },
    "EmployeeInfo": {
      /*ro, opt, object, 工号信息, desc:proMode为employee时有效*/
      "employeeNo": "test"
      /*ro, opt, string, 工号（人员ID）*/
    },
    "eventSourceID": 1,
    /*ro, opt, int, 事件源ID, desc:proMode为event时有效, 当主类型为设备事件（mainEventType为0）时无效; 当主类型是门事件（mainEventType为2）时为门编号; 当主类型为读卡器事件（mainEventType为3）时, 为读卡器ID; 当为报警输入事件（mainEventType为1）时为防区报警输入ID或事件报警输入ID。65535表示联动全部*/
    "alarmOut": [1, 3, 5],
    /*ro, opt, array, 关联的报警输出号, subType:int, desc:[1,3,5]代表关联报警输出1、报警输出3、报警输出5、用于本地报警输出*/
    "ReaderAlarmOut": [
      /*ro, opt, array, 联动读卡器的报警输出号, subType:object, desc:用于支持多路读卡器报警输出, 当前适用于智能主机*/
      {
        "readerID": 1,
        /*ro, opt, int, 读卡器编号*/
        "alarmOut": [1, 3, 5],
        /*ro, opt, array, 关联的报警输出号, subType:int, desc:[1,3,5]代表关联报警输出1、报警输出3、报警输出5*/
        "alarmOutClose": [1, 3, 5]
        /*ro, opt, array, 关联报警输出关闭, subType:int, desc:[1,3,5]代表关联报警输出1、报警输出3、报警输出5的关闭*/
      }
    ],
    "openDoor": [1, 3, 5],
    /*ro, opt, array, 是否联动开门, subType:int, desc:[1,3,5]代表联动门1、门3、门5开门*/
    "closeDoor": [1, 3, 5],
    /*ro, opt, array, 是否联动关门, subType:int, desc:[1,3,5]代表联动门1、门3、门5关门*/
    "alwaysOpen": [1, 3, 5],
    /*ro, opt, array, 是否联动常开, subType:int, desc:[1,3,5]代表联动门1、门3、门5常开*/
    "alwaysClose": [1, 3, 5],
    /*ro, opt, array, 是否联动常闭, subType:int, desc:[1,3,5]代表联动门1、门3、门5常闭*/
    "mainDevBuzzer": true,
    /*ro, opt, bool, 主机蜂鸣器, desc:false-不联动输出, true-联动输出*/
    "capturePic": true,
    /*ro, opt, bool, 是否联动抓拍, desc:false-不联动抓拍, true-联动抓拍, desc:主要用于一体机, 仅支持一路读卡器抓拍*/
    "readerCapturePic": [1, 2],
    /*ro, opt, array, 联动读卡器抓拍, subType:int, range:[0,8], desc:[1,2]代表联动读卡器1,2抓拍。用于支持多路读卡器抓拍, 当前适用于智能主机*/
    "recordVideo": true,
    /*ro, opt, bool, 是否联动录像, desc:false-不联动录像, true-联动录像, desc:主要用于一体机, 仅支持一路读卡器录像*/
    "readerRecordVideo": [1, 2],
    /*ro, opt, array, 联动读卡器录像, subType:int, range:[0,8], desc:[1,2]代表联动读卡器1,2录像。用于支持多路读卡器录像, 当前适用于智能主机*/
    "mainDevStopBuzzer": true,
    /*ro, opt, bool, 主机停止蜂鸣, desc:false-不联动停止, true-联动停止*/
    "audioSourceType": "TTS",
    /*ro, opt, enum, 音频源类型, subType:string, [TTS#TTS语音,customFile#自定义语音,none#取消联动]*/
    "audioDisplayTTSLanguage": "SimChinese",
    /*ro, opt, enum, 联动语音播报TTS语言类型, subType:string, [SimChinese#简体中文,English#英文], dep:and,
    {$.EventCardLinkageCfg.audioSourceType,eq,TTS}, desc:默认简体中文*/
    "audioDisplayTTS": "test",
    /*ro, opt, string, 联动语音播报TTS内容, range:[1,64], dep:and,{$.EventCardLinkageCfg.audioSourceType,eq,TTS}*/
    "customAudioID": 1,
    /*ro, opt, int, 联动的自定义音频文件ID, range:[1,65535], dep:and,{$.EventCardLinkageCfg.audioSourceType,eq,customFile}*/
    "audioDisplayMode": "close",
    /*ro, opt, enum, 联动语音播放模式, subType:string, [close#关闭,single#单次播放,loop#循环播放]*/
    "readerBuzzer": [1, 3, 5],
    /*ro, opt, array, 联动读卡器蜂鸣器, subType:int, desc:[1,3,5]代表联动读卡器1、读卡器3、读卡器5的蜂鸣器*/
    "alarmOutClose": [1, 3, 5],
    /*ro, opt, array, 联动报警输出关闭, subType:int, desc:[1,3,5]代表关联报警输出1、报警输出3、报警输出5的关闭*/
  }
}
```


Request URL

POST /ISAPI/AccessControl/AcsEvent?format=json

查询参数

无

请求报文

```
{
  "AcsEventCond": {
    /*req, object, 门禁设备事件*/
    "searchID": "test",
    /*req, string, 搜索记录唯一标识, desc:用来确认上层客户端是否为同一个（倘若是同一个，设备记录内存，下次搜索加快速度）*/
    "searchResultPosition": 0,
    /*req, int, 查询结果在结果列表中的起始位置；当记录条数很多时，desc:当记录条数很多时，一次查询不能获取所有的记录，下一次查询时指定位置可以查询后面的记录（若设备支持的最大totalMatches为M个，但是当前设备已存储的totalMatches为N个（N<=M），则该字段的合法范围为0~N-1）*/
    "maxResults": 30,
    /*req, int, 本次协议调用可获取的最大记录数（如maxResults值大于设备能力集返回的范围，desc:如maxResults值大于设备能力集返回的范围，则设备按照能力集最大值返回，设备不进行报错*/
    "major": 1,
    /*req, int, 主类型, desc:参考事件上传宏定义，0-全部（此处传递的是10进制数据，不能传递16进制，如传递1代表0x1（报警主类型））*/
    "minor": 1024,
    /*req, int, 次类型, desc:参考事件上传宏定义，0-全部（此处传递的是10进制数据，不能传递16进制，如传递1024代表0x400（防区短路报警次类型））*/
    "startTime": "1970-01-01T00:00:00+08:00",
    /*opt, datetime, 开始时间（UTC时间）*/
    "endTime": "1970-01-01T00:00:00+08:00",
    /*opt, datetime, 结束时间（UTC时间）*/
    "cardNo": "test",
    /*opt, string, 卡号*/
    "name": "test",
    /*opt, string, 持卡人姓名*/
    "videoChannel": 1,
    /*opt, int, 视频通道号, range:[1,86400], desc:考勤超脑增加*/
    "picEnable": true,
    /*opt, bool, 是否带图片, desc:false-不带图片，true-带图片（1.配置为false，所有符合条件的事件不带图片上报。2.配置为true，所有符合条件的事件如无联动图片，则上报事件本身；符合条件的事件如有联动图片，则上报事件+联动图片。3.该字段如未配置，则设备默认按照true返回。）*/
    "beginSerialNo": 1,
    /*opt, int, 起始流水号*/
    "endSerialNo": 1,
    /*opt, int, 结束流水号*/
    "employeeNoString": "test",
    /*opt, string, 工号（人员ID）*/
    "timeReverseOrder": true,
    /*opt, bool, 是否按时间逆序返回事件, desc:true-是，不存在或者false-否*/
    "isAbnormalTemperature": true,
    /*opt, bool, 人脸温度是否异常*/
    "temperatureSearchCond": "all",
    /*opt, enum, 温度搜索条件, subType:string, [all#携带温度信息的事件,normal#温度信息正常的事件,abnormal#温度信息异常的事件], desc:当与isAbnormalTemperature字段同时存在时，isAbnormalTemperature字段无效*/
    "isAttendanceInfo": true,
    /*opt, bool, 是否为考勤记录信息, desc:HEOP协议新增字段，如果为true，该查询仅需返回事件的主类型、次类型、工号、姓名、时间信息*/
    "hasRecordInfo": true,
    /*opt, bool, 是否存在录像信息*/
  }
}
```

响应报文

```
{
  "AcsEvent": {
    /*ro, req, object, 门禁设备事件*/
    "searchID": "test",
    /*ro, req, string, 搜索记录唯一标识：用来确认上层客户端是否为同一个（倘若是同一个*/
    "responseStatusStrg": "OK",
    /*ro, req, string, 查询状态字符串描述*/
    "numOfMatches": 1,
    /*ro, req, int, 本次返回的记录条数*/
    "totalMatches": 1,
    /*ro, req, int, 符合条件的记录总条数*/
    "InfoList": [
    /*ro, opt, array, 信息列表, subType:object*/
    {
      "major": 1,
      /*ro, req, int, 报警主类型*/
      "minor": 1,
      /*ro, req, int, 报警次类型*/
      "time": "2016-12-12T17:30:08+08:00",
      /*ro, req, string, 时间（UTC时间）*/
      "netUser": "test",
      /*ro, opt, string, 网络操作的用户名*/
      "remoteHostAddr": "test",
      /*ro, opt, string, 远程主机地址*/
      "videoChannel": 1,
      /*ro, opt, int, 视频通道号, range:[1,86400], desc:考勤超脑增加*/
      "cardNo": "test",
      /*ro, opt, string, 卡号*/
      "name": "test",
      /*ro, opt, string, 持卡人姓名*/
    }
  ]
}
```

```

"cardType": 1,
/*ro, opt, enum, 卡类型, subType:int, [1#普通卡,2#残疾人卡,3#黑名单卡,4#巡更卡,5#胁迫卡,6#超级卡,7#来宾卡,8#解除卡]*/
"whiteListNo": 1,
/*ro, opt, int, 白名单单号*/
"reportChannel": 1,
/*ro, opt, int, 报告上传通道*/
"cardReaderKind": 1,
/*ro, opt, int, 读卡器属于哪一类: 1-IC读卡器*/
"cardReaderNo": 1,
/*ro, opt, int, 读卡器编号*/
"doorNo": 1,
/*ro, opt, int, 门编号(楼层编号)*/
"verifyNo": 1,
/*ro, opt, int, 多重卡认证序号*/
"alarmInNo": 1,
/*ro, opt, int, 报警输入号*/
"alarmOutNo": 1,
/*ro, opt, int, 报警输出号*/
"caseSensorNo": 1,
/*ro, opt, int, 事件触发器编号*/
"RS485No": 1,
/*ro, opt, int, RS485通道号*/
"multiCardGroupNo": 1,
/*ro, opt, int, 群组编号*/
"accessChannel": 1,
/*ro, opt, int, 人员通道号*/
"deviceNo": 1,
/*ro, opt, int, 设备编号*/
"distractControlNo": 1,
/*ro, opt, int, 分控器编号*/
"employeeNoString": "test",
/*ro, opt, string, 工号(人员ID)*/
"localControllerID": 1,
/*ro, opt, int, 就地控制器编号*/
"InternetAccess": 1,
/*ro, opt, int, 网口ID*/
"type": 1,
/*ro, opt, int, 防区类型, desc:0-即时防区, 1-24小时防区, 2-延时防区, 3-内部防区, 4-钥匙防区, 5-火警防区, 6-周界防区, 7-24小时无声防区, 8-24
小时辅助防区, 9-24小时震动防区, 10-门禁紧急开门防区, 11-门禁紧急关门防区, 255-无*/
"MACAddr": "test",
/*ro, opt, string, 物理地址*/
"swipeCardType": 1,
/*ro, opt, enum, 刷卡类型, subType:int, [0#无效,1#二维码]*/
"serialNo": 1,
/*ro, opt, int, 事件流水号*/
"channelControllerID": 1,
/*ro, opt, int, 通道控制器ID*/
"channelControllerLampID": 1,
/*ro, opt, int, 通道控制器灯板ID, range:[1,255]*/
"channelControllerIRAdaptorID": 1,
/*ro, opt, int, 通道控制红外转接板ID, range:[1,255]*/
"channelControllerIREmitterID": 1,
/*ro, opt, int, 通道控制红外发射ID, range:[1,255]*/
"userType": "normal",
/*ro, opt, string, 人员类型*/
"currentVerifyMode": "cardAndPw",
/*ro, opt, enum, 读卡器当前验证方式, subType:string, [cardAndPw#刷卡+密码, card#刷卡, cardOrPw#刷卡或密码, fp#指纹, fpAndPw#指纹+密码, fpOrCard#
指纹或刷卡, fpAndCard#指纹+刷卡, fpAndCardAndPw#指纹+刷卡+密码, faceOrFpOrCardOrPw#人脸或指纹或刷卡或密码, faceAndFp#人脸+指纹, faceAndPw#人脸+密码, faceAndCard#
人脸+刷卡, face#人脸, employeeNoAndPw#工号+密码, fpOrPw#指纹或密码, employeeNoAndFp#工号+指纹, employeeNoAndFpAndPw#工号+指纹+密码, faceAndFpAndCard#人脸+指纹+刷
卡, faceAndPwAndFp#人脸+密码+指纹, employeeNoAndFace#工号+人脸, faceOrfaceAndCard#人脸或人脸+刷卡, fpOrface#指纹或人脸, cardOrfaceOrPw#刷卡或人脸或密
码, faceOrFpOrCardOrPwOrIris#人脸或指纹或刷卡或密码或虹膜, faceOrCardOrPwOrIris#人脸或刷卡或密码或虹膜, sleep#休眠, invalid#无效]*/
"QRCodeInfo": "test",
/*ro, opt, string, 二维码信息*/
"thermometryUnit": "celsius",
/*ro, opt, enum, 测温单位, subType:string, [celsius#摄氏度(默认), fahrenheit#华氏度, kelvin#开尔文]*/
"currTemperature": 36.5,
/*ro, opt, float, 人脸温度(精确到小数点后一位)*/
"isAbnormalTemperature": true,
/*ro, opt, bool, 人脸测温是否温度异常(true-是)*/
"RegionCoordinates": {
/*ro, opt, object, 人脸温度坐标*/
"positionX": 254,
/*ro, opt, int, X坐标 归一化坐标0-1000*/
"positionY": 133,
/*ro, opt, int, Y坐标 归一化坐标0-1000*/
},
"mask": "unknown",
/*ro, opt, enum, 是否戴口罩, subType:string, [unknown#未知]*/
"pictureURL": "test",
/*ro, opt, string, 图片URL*/
"filename": "picture1",
/*ro, opt, string, 文件名称, desc:该名称需要与后面的filename保持一致, 并且如果一次返回多张图片, 每一张图片的filename要保证唯一性, 不能与其他
图片相同*/
"attendanceStatus": "undefined",
/*ro, opt, enum, 考勤状态, subType:string, [undefined#未定义, checkIn#上班, checkOut#下班, breakOut#开始休息, breakIn#结束休息, overtimeIn#开始加
班, overtimeOut#结束加班]*/
"label": "test",
/*ro, opt, string, 自定义考勤名称*/
"statusValue": 1,
/*ro, opt, int, 状态值*/
"helmet": "unknown",
/*ro, opt, enum, 是否戴安全帽, subType:string, [unknown#未知, yes#戴安全帽, no#不戴安全帽]*/
"visibleLightPicUrl": "test",
/*ro, opt, string, 可见光图片url*/

```

```

        "thermalPicUrl": "test",
        /*ro, opt, string, 热成像图片url*/
        "appType": "attendance",
        /*ro, opt, enum, 应用类型, subType:string, [attendance#考勤应用,signIn#签到应用（信息发布产品使用）]*/
        "HealthInfo": {
        /*ro, opt, object, 健康信息*/
            "healthCode": 1,
            /*ro, opt, enum, 健康码状态, subType:int, [0#未请求,1#未申领,2#绿码,3#黄码,4#红码,5#无此人员,6#其他错误信息（如接口异常导致查询失败）,7#
            查询健康码超时]*/
            "NADCode": 1,
            /*ro, opt, enum, 核酸检测结果, subType:int, [0#未查询到核酸检测结果,1#核酸检测阴性（代表正常）,2#核酸检测阴性（代表确诊）,3#核酸检测有有效
            期已过,4#查询检测结果失败]*/
            "NADMsg": "test",
            /*ro, opt, string, 核酸检测信息, range:[0,64], desc:空字符串表示查询到核酸检测信息失败*/
            "NADTime": 1,
            /*ro, opt, enum, 核酸检测时间, subType:int, [1#24小时内,2#48小时内,3#48小时外]*/
            "travelCode": 1,
            /*ro, opt, enum, 行程编码, subType:int, [0#14天内一直在当地,1#14天内离开过当地,2#14天内到过疫区,3#其他,4#查询行程信息失败]*/
            "travelInfo": "test",
            /*ro, opt, string, 行程信息, desc:空字符串表示查询到行程信息失败*/
            "vaccineStatus": 1,
            /*ro, opt, enum, 打疫苗状态, subType:int, [0#未打疫苗,1#部分注射疫苗,2#已完成疫苗,3#查询疫苗信息失败,4#已完成加强针疫苗]*/
            "vaccineNum": 1,
            /*ro, opt, int, 打疫苗针数, step:1*/
            "vaccineMsg": "test",
            /*ro, opt, string, 打疫苗信息, range:[0,64], desc:空字符串表示查询到疫苗信息失败*/
            "ANTCode": 1,
            /*ro, opt, enum, 抗原检测状态, subType:int, [0#未查询到抗原检测结果,1#抗原检测阴性,2#抗原检测阳性,3#抗原检测有效期已过,4#抗原检测无效]*/
            "ANTMsg": "test"
            /*ro, opt, string, 抗原检测信息, range:[0,64], desc:空字符串表示查询到抗原检测信息失败*/
        },
        "meetingID": "test",
        /*ro, opt, string, 唯一标识一个会议计划*/
        "PersonInfoExtends": [
        /*ro, opt, array, 人员信息扩展, subType:object,
        desc:人员在设备UI界面的扩展信息显示
        明眸当前仅支持一个value, 不支持id字段, 用于作为工号显示
        智能主机按标准实现*/
            {
                "id": 1,
                /*ro, opt, int, 人员信息扩展序号, range:[1,32],
                desc:与/ISAPI/AccessControl/personInfoExtendName?format=json中的id对应, 用于关联value的name信息
                id不存在时, 根据数组顺序默认从1开始*/
                "value": "test"
                /*ro, opt, string, 人员信息扩展内容*/
            }
        ],
        "name": "test",
        /*ro, opt, string, 姓名, desc:HEOP新增字段, 姓名*/
        "FaceRect": {
        /*ro, opt, object, 人脸矩形框, desc:屏幕左上角为坐标原点*/
            "height": 1.000,
            /*ro, req, float, 高度, range:[0.000,1.000]*/
            "width": 1.000,
            /*ro, req, float, 宽度, range:[0.000,1.000]*/
            "x": 0.000,
            /*ro, req, float, 区域左上角顶点的水平坐标, range:[0.000,1.000]*/
            "y": 0.000
            /*ro, req, float, 区域左上角顶点的垂直坐标, range:[0.000,1.000]*/
        },
        "RecordInfo": {
        /*ro, opt, object, 事件联动录像信息*/
            "recordStatus": "recording",
            /*ro, req, enum, 录像状态, subType:string, [recording#录像中,completed#录像已完成]*/
            "startTime": "1970-01-01T00:00:00+08:00",
            /*ro, opt, datetime, 录像开始时间, dep:or,{$.AcsEvent.InfoList[*].RecordInfo.recordStatus,eq,completed}*/
            "endTime": "1970-01-01T00:00:00+08:00"
            /*ro, opt, datetime, 录像结束时间, dep:or,{$.AcsEvent.InfoList[*].RecordInfo.recordStatus,eq,completed}*/
        },
        "currentAuthenticationTimes": 1,
        /*ro, opt, int, 当前权限计划时间段已认证次数, range:[0,255], step:1, desc:若认证通过, 认证次数加1*/
        "allowAuthenticationTimes": 1
        /*ro, opt, int, 当前权限计划时间段允许的认证次数, range:[0,255], step:1*/
    }
}
}
}

```

69.45.7 获取查询门禁事件参数能力

Request URL

GET /ISAPI/AccessControl/AcsEvent/capabilities?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "AcsEvent": {
    /*ro, req, object, 门禁设备事件*/
    "AcsEventCond": {
      /*ro, opt, object, 查找条件*/
      "searchID": {
        /*ro, req, object, 搜索记录唯一标识: 用来确认上层客户端是否为同一个 (倘若为同一个*/
        "@min": 1,
        /*ro, req, int, 最小值*/
        "@max": 1
        /*ro, req, int, 最大值*/
      },
      "searchResultPosition": {
        /*ro, req, object, 查询结果在结果列表中的起始位置: 当记录条数很多时*/
        "@min": 1,
        /*ro, req, int, 最小值*/
        "@max": 1
        /*ro, req, int, 最大值*/
      },
      "maxResults": {
        /*ro, req, object, 本次协议调用可获取的最大记录数*/
        "@min": 1,
        /*ro, req, int, 最小值*/
        "@max": 1
        /*ro, req, int, 最大值*/
      },
      "major": {
        /*ro, opt, object, 专业*/
        "@opt": "0,1,2,3,5"
        /*ro, req, string, 主类型*/
      },
      "minorAlarm": {
        /*ro, opt, object, 小调*/
        "@opt": "1024,1025,1026,1027..."
        /*ro, req, string, 报警次类型*/
      },
      "minorException": {
        /*ro, opt, object, 轻微例外*/
        "@opt": "39,58,59,1024..."
        /*ro, req, string, 异常次类型*/
      },
      "minorOperation": {
        /*ro, opt, object, 轻微操作*/
        "@opt": "80,90,112,113..."
        /*ro, req, string, 操作次类型*/
      },
      "minorEvent": {
        /*ro, opt, object, 轻微事件*/
        "@opt": "1,2,3,4..."
        /*ro, req, string, 事件次类型*/
      },
      "startTime": {
        /*ro, opt, object, 开始时间*/
        "@min": 0,
        /*ro, req, int, 字符串最小长度, range:[0,32]*/
        "@max": 32
        /*ro, req, int, 字符串最大长度, range:[0,32]*/
      },
      "endTime": {
        /*ro, opt, object, 结束时间*/
        "@min": 0,
        /*ro, req, int, 字符串最小长度, range:[0,32]*/
        "@max": 32
        /*ro, req, int, 字符串最大长度, range:[0,32]*/
      },
      "cardNo": {
        /*ro, opt, object, 卡号*/
        "@min": 1,
        /*ro, req, int, 卡号*/
        "@max": 1
        /*ro, req, int, 卡号*/
      },
      "name": {
        /*ro, opt, object, 持卡人姓名*/
        "@min": 1,
        /*ro, req, int, 持卡人姓名*/
        "@max": 1
        /*ro, req, int, 持卡人姓名*/
      },
      "picEnable": "true,false",
      /*ro, opt, string, 是否带图片: false-不带图片*/
      "beginSerialNo": {
        /*ro, opt, object, 起始流水号*/
        "@min": 1,
        /*ro, req, int, 起始流水号*/
        "@max": 1
        /*ro, req, int, 起始流水号*/
      },
    },
  },
}
```

```

    "endSerialNo": {
/*ro, opt, object, 结束流水号*/
        "@min": 1,
/*ro, req, int, 起始流水号*/
        "@max": 1
/*ro, req, int, 结束流水号*/
    },
    "employeeNoString": {
/*ro, opt, object, 工号*/
        "@min": 1,
/*ro, req, int, 工号（人员ID）*/
        "@max": 1
/*ro, req, int, 工号（人员ID）*/
    },
},
"InfoList": {
/*ro, opt, object, 信息列表*/
    "maxSize": 10,
/*ro, opt, int, 最大值*/
    "time": {
/*ro, opt, object, 时间（UTC时间）*/
        "@min": 0,
/*ro, req, int, 字符串最小长度, range:[0,32]*/
        "@max": 32
/*ro, req, int, 字符串最大长度, range:[0,32]*/
    },
    "netUser": {
/*ro, opt, object, 网络操作的用户名*/
        "@min": 1,
/*ro, req, int, 网络操作的用户名*/
        "@max": 1
/*ro, req, int, 网络操作的用户名*/
    },
    "remoteHostAddr": {
/*ro, opt, object, 远程主机地址*/
        "@min": 1,
/*ro, req, int, 远程主机地址*/
        "@max": 1
/*ro, req, int, 远程主机地址*/
    },
    "cardNo": {
/*ro, opt, object, 卡号*/
        "@min": 1,
/*ro, req, int, 卡号*/
        "@max": 1
/*ro, req, int, 卡号*/
    },
    "cardType": {
/*ro, opt, object, 卡类型*/
        "@opt": "1,2,3,4,5,6,7,8"
/*ro, req, string, 卡类型: 1-普通卡*/
    },
    "reportChannel": {
/*ro, opt, object, 报告上传通道*/
        "@opt": "1,2,3"
/*ro, req, string, 报告上传通道: 1-布网上传*/
    },
    "cardReaderKind": {
/*ro, opt, object, 读卡器属于哪一类*/
        "@opt": "1,2,3,4"
/*ro, req, string, 读卡器属于哪一类: 1-IC读卡器*/
    },
    "cardReaderNo": {
/*ro, opt, object, 读卡器编号*/
        "@min": 1,
/*ro, req, int, 读卡器编号*/
        "@max": 1
/*ro, req, int, 读卡器编号*/
    },
    "doorNo": {
/*ro, opt, object, 门编号（楼层编号）*/
        "@min": 1,
/*ro, req, int, 门编号（楼层编号）*/
        "@max": 1
/*ro, req, int, 门编号（楼层编号）*/
    },
    "verifyNo": {
/*ro, opt, object, 多重卡认证序号*/
        "@min": 1,
/*ro, req, int, 多重卡认证序号*/
        "@max": 1
/*ro, req, int, 多重卡认证序号*/
    },
    "caseSensorNo": {
/*ro, opt, object, 事件触发器编号*/
        "@min": 1,
/*ro, req, int, 事件触发器编号*/
        "@max": 1
/*ro, req, int, 事件触发器编号*/
    },
    "RS485No": {
/*ro, opt, object, RS485通道号*/
        "@min": 1,

```

湖南云舟电力科技有限公司

```

/*ro, req, int, RS485通道号*/
"@max": 1
/*ro, req, int, RS485通道号*/
},
"multiCardGroupNo": {
/*ro, opt, object, 群组编号*/
"@min": 1,
/*ro, req, int, 群组编号*/
"@max": 1
/*ro, req, int, 群组编号*/
},
"deviceNo": {
/*ro, opt, object, 设备编号*/
"@min": 1,
/*ro, req, int, 设备编号*/
"@max": 1
/*ro, req, int, 设备编号*/
},
"employeeNo": {
/*ro, opt, object, 工号（人员ID）*/
"@min": 1,
/*ro, req, int, 工号（人员ID）*/
"@max": 1
/*ro, req, int, 工号（人员ID）*/
},
"InternetAccess": {
/*ro, opt, object, 网口ID*/
"@min": 1,
/*ro, req, int, 网口ID: 1-上行网口1*/
"@max": 1
/*ro, req, int, 网口ID: 1-上行网口1*/
},
"MACAddr": {
/*ro, opt, object, 物理地址*/
"@min": 1,
/*ro, req, int, 物理地址*/
"@max": 1
/*ro, req, int, 物理地址*/
},
"swipeCardType": {
/*ro, opt, object, 刷卡类型*/
"@opt": "0,1"
/*ro, req, string, 刷卡类型: 0-无效*/
},
"serialNo": {
/*ro, opt, object, 事件流水号*/
"@min": 1,
/*ro, req, int, 事件流水号*/
"@max": 1
/*ro, req, int, 事件流水号*/
},
"userType": {
/*ro, opt, object, 人员类型*/
"@opt": "normal,visitor,blackList,administrators"
/*ro, req, string, 人员类型*/
},
"currentVerifyMode": {
/*ro, opt, object, 读卡器当前验证方式*/
"@opt":
"cardAndPw,card,cardOrPw,fp,fpAndPw,fpOrCard,fpAndCard,fpAndCardAndPw,faceOrFpOrCardOrPw,faceAndFp,faceAndPw,faceAndCard,face,employeeNoAndPw,fpOrPw,employeeNoAndFp,employeeNoAndFpAndPw,faceAndFpAndCard,faceAndPwAndFp,employeeNoAndFace,faceOrfaceAndCard,fpOrface,cardOrfaceOrPw,iris,faceOrFpOrCardOrPwOrIris,faceOrCardOrPwOrIris"
/*ro, req, string, 读卡器当前验证方式*/
},
"thermometryUnit": {
/*ro, opt, object, 测温单位*/
"@opt": ["celsius", "fahrenheit", "kelvin"]
/*ro, req, array, 测温单位（celsius-摄氏度（默认）,subType:string*/
},
"currTemperature": {
/*ro, opt, object, 人脸温度*/
"@min": 1,
/*ro, req, int, 人脸温度（精确到小数点后一位）*/
"@max": 1
/*ro, req, int, 人脸温度（精确到小数点后一位）*/
},
"isAbnormalTemperature": {
/*ro, opt, object, 人脸测温是否温度异常*/
"@opt": [true, false]
/*ro, req, array, 人脸测温是否温度异常（true-是,subType:bool*/
},
"RegionCoordinates": {
/*ro, opt, object, 人脸温度坐标*/
"positionX": {
/*ro, opt, object, X坐标*/
"@min": 0,
/*ro, req, int, X坐标 归一化坐标0-1000*/
"@max": 1000
/*ro, req, int, X坐标 归一化坐标0-1000*/
},
"positionY": {
/*ro, opt, object, Y坐标*/
"@min": 0,
/*ro, req, int, Y坐标 归一化坐标0-1000*/
}
}

```

湖南云舟电力科技有限公司

```

        "max": 1000
        /*ro, req, int, Y坐标 归一化坐标0-1000*/
    }
},
"mask": {
    /*ro, opt, object, 是否戴口罩*/
    "@opt": ["unknown", "yes", "no"]
    /*ro, req, array, 是否戴口罩: unknown-未知, subType:string*/
},
"pictureURL": {
    /*ro, opt, object, 抓拍图片URL*/
    "@min": 1,
    /*ro, req, int, URL长度最小值*/
    "@max": 1
    /*ro, req, int, URL长度最大值*/
},
"attendanceStatus": {
    /*ro, opt, object, 考勤状态, desc:undefined-未定义, checkIn-上班, checkOut-下班, breakOut-开始休息, breakIn-结束休息, overtimeIn-开始加班, overtimeOut-结束加班*/
    "@opt": "undefined,checkIn,checkOut,breakOut,breakIn,overtimeIn,overtimeOut"
    /*ro, req, string, 可选项*/
},
"label": {
    /*ro, opt, object, 自定义考勤名称*/
    "@min": 1,
    /*ro, req, int, 自定义考勤名称长度最小值*/
    "@max": 1
    /*ro, req, int, 自定义考勤名称长度最大值*/
},
"statusValue": {
    /*ro, opt, object, 状态值*/
    "@min": 1,
    /*ro, req, int, 最小值*/
    "@max": 1
    /*ro, req, int, 最大值*/
},
"helmet": {
    /*ro, opt, object, 是否戴安全帽, desc:unknown-未知, yes-戴安全帽, no-不戴安全帽*/
    "@opt": "unknown,yes,no"
    /*ro, req, string, 可选项*/
},
"thermalPicUrl": {
    /*ro, opt, object, 热成像图片url*/
    "@min": 1,
    /*ro, req, int, 热成像图片url长度最小值*/
    "@max": 1
    /*ro, req, int, 热成像图片url长度最大值*/
},
"HealthInfo": {
    /*ro, opt, object, 健康信息*/
    "healthCode": {
        /*ro, opt, object, 健康码状态*/
        "@opt": [0, 1, 2, 3, 4, 5, 6, 7]
        /*ro, req, array, 可选项, subType:int, desc:0#未申领,1#未申领,2#绿码,3#黄码,4#红码,5#无此人员,6#其他错误信息(如接口异常导致查询失败),7#查询健康码超时*/
    },
    "NADCode": {
        /*ro, opt, object, 核酸检测结果, desc:0#未查询到核酸检测结果,1#核酸检测阴性(代表正常),2#核酸检测阳性(代表确诊),3#核酸检测有效期已过*/
        "@opt": [0, 1, 2, 3, 4]
        /*ro, req, array, 可选项, subType:int*/
    },
    "travelCode": {
        /*ro, opt, object, 行程编码, desc:0#14天内一直在当地,1#14天内离开过当地,2#14天内到过疫区,3#其他*/
        "@opt": [0, 1, 2, 3, 4]
        /*ro, req, array, 可选项, subType:int*/
    },
    "travelInfo": {
        /*ro, opt, object, 行程信息*/
        "@min": 0,
        /*ro, req, int, 取值范围, step:1*/
        "@max": 15
        /*ro, req, int, 取值范围, step:1*/
    },
    "vaccineStatus": {
        /*ro, opt, object, 打疫苗状态, desc:0#未打疫苗,1#打过疫苗*/
        "@opt": [0, 1, 2, 3, 4]
        /*ro, req, array, 可选项, subType:int*/
    },
    },
},
}
}
}

```

69.45.8 获取事件及卡号联动参数能力

Request URL

GET /ISAPI/AccessControl/EventCardLinkageCfg/capabilities?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "EventCardLinkageCfg": {
    /*ro, req, object, 事件及卡号联动参数*/
    "eventID": {
      /*ro, opt, object, 事件ID*/
      "@min": 1,
      /*ro, opt, int, 事件ID最小值*/
      "@max": 1
      /*ro, opt, int, 事件ID最大值*/
    },
    "proMode": {
      /*ro, req, object, 联动方式*/
      "@opt": "event,card,mac,employee"
      /*ro, opt, string, 联动方式*/
    },
    "EventLinkageInfo": {
      /*ro, opt, object, 事件联动信息*/
      "mainEventType": {
        /*ro, opt, object, 事件主类型*/
        "@opt": "0,1,2,3"
        /*ro, opt, string, 事件主类型*/
      },
      "devSubEventType": {
        /*ro, opt, object, 设备事件次类型*/
        "@opt": "0,1,2,3,54..."
        /*ro, opt, string, 设备事件次类型*/
      },
      "alarmSubEventType": {
        /*ro, opt, object, 报警输入事件次类型*/
        "@opt": "0,1,2,3,52..."
        /*ro, opt, string, 报警输入事件次类型*/
      },
      "doorSubEventType": {
        /*ro, opt, object, 门事件次类型*/
        "@opt": "0,1,2,3..."
        /*ro, opt, string, 门事件次类型*/
      },
      "cardReaderSubEventType": {
        /*ro, opt, object, 读卡器事件次类型*/
        "@opt": "0,1,2,3..."
        /*ro, opt, string, 读卡器事件次类型*/
      }
    },
    "CardNoLinkageInfo": {
      /*ro, opt, object, 卡号联动信息*/
      "cardNo": {
        /*ro, opt, object, 卡号*/
        "@min": 1,
        /*ro, opt, int, 卡号最小值*/
        "@max": 32
        /*ro, opt, int, 卡号最大值*/
      }
    },
    "EmployeeInfo": {
      /*ro, opt, object, 工号（人员ID）*/
      "employeeNo": {
        /*ro, opt, object, 工号（人员ID）*/
        "@min": 1,
        /*ro, opt, int, 工号（人员ID）最小值*/
        "@max": 32
        /*ro, opt, int, 工号（人员ID）最大值*/
      }
    },
    "eventSourceID": {
      /*ro, opt, object, 事件源ID*/
      "@min": 1,
      /*ro, opt, int, 事件源ID最小值*/
      "@max": 1
      /*ro, opt, int, 事件源ID最大值*/
    },
    "alarmout": {
      /*ro, opt, object, 关联的报警输出号*/
      "@min": 1,
      /*ro, opt, int, 关联的报警输出号最小值*/
      "@max": 1
      /*ro, opt, int, 关联的报警输出号最大值*/
    },
    "openDoor": {
      /*ro, opt, object, 是否联动开门*/
      "@min": 1,
      /*ro, opt, int, 是否联动开门最小值*/
      "@max": 1
      /*ro, opt, int, 是否联动开门最大值*/
    }
  }
}
```

```
        "@max": 1,
        /*ro, opt, int, 是否联动开门最大值*/
    },
    "closeDoor": {
        /*ro, opt, object, 是否联动关门*/
        "@min": 1,
        /*ro, opt, int, 是否联动关门最小值*/
        "@max": 1,
        /*ro, opt, int, 是否联动关门最大值*/
    },
    "alwaysOpen": {
        /*ro, opt, object, 是否联动常开*/
        "@min": 1,
        /*ro, opt, int, 是否联动常开最小值*/
        "@max": 1,
        /*ro, opt, int, 是否联动常开最大值*/
    },
    "alwaysClose": {
        /*ro, opt, object, 是否联动常闭*/
        "@min": 1,
        /*ro, opt, int, 是否联动常闭最小值*/
        "@max": 1,
        /*ro, opt, int, 是否联动常闭最大值*/
    },
    "capturePic": "true,false",
    /*ro, opt, string, 是否联动抓拍*/
    "alarmOutClose": {
        /*ro, opt, object, 关联报警输出关闭*/
        "@min": 1,
        /*ro, opt, int, 关联报警输出关闭最小值*/
        "@max": 1,
        /*ro, opt, int, 关联报警输出关闭最大值*/
    },
    "purePwdVerifyEnable": true,
    /*ro, opt, bool, 纯密码开门,
    desc:1、true-设备支持纯密码开门（人员信息中的-password字段），不返回本字段-不支持纯密码开门。
    2、纯密码方案：
    （1）认证方式中的"或密码"为人员密码。
    （2）设备不对人员密码的重复性进行校验，需要上层平台自行保证密码的唯一性。
    （3）设备本地不能对人员密码进行操作（增删改查）。*/
}
}
```

69.45.9 获取门禁设备事件总条数能力

Request URL

GET /ISAPI/AccessControl/AcsEventTotalNum/capabilities?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "AcsEvent": {
    /*ro, opt, object, 门禁设备事件*/
    "AcsEventTotalNumCond": {
      /*ro, opt, object, 查找条件*/
      "major": {
        /*ro, req, object, 主类型*/
        "@opt": "0,1,2,3,5"
        /*ro, opt, string, 主类型*/
      },
      "minorAlarm": {
        /*ro, req, object, 报警次类型*/
        "@opt": "1024,1025,1026,1027"
        /*ro, opt, string, 报警次类型*/
      },
      "minorException": {
        /*ro, req, object, 异常次类型*/
        "@opt": "39,58,59,1024"
        /*ro, opt, string, 异常次类型*/
      },
      "minorOperation": {
        /*ro, req, object, 操作次类型*/
        "@opt": "80,90,112,113"
        /*ro, opt, string, 操作次类型*/
      },
      "minorEvent": {
        /*ro, opt, object, 事件次类型*/
        "@opt": "1,2,3,4"
        /*ro, opt, string, 事件次类型*/
      },
    },
  },
}
```

```

    "startTime": {
/*ro, opt, object, 开始时间*/
        "@min": 1,
/*ro, opt, int, 开始时间（UTC时间）*/
        "@max": 1
/*ro, opt, int, 结束时间（UTC时间）*/
    },
    "endTime": {
/*ro, opt, object, 结束时间*/
        "@min": 1,
/*ro, opt, int, 开始时间（UTC时间）*/
        "@max": 1
/*ro, opt, int, 结束时间（UTC时间）*/
    },
    "cardNo": {
/*ro, opt, object, 卡号*/
        "@min": 1,
/*ro, opt, int, 卡号*/
        "@max": 32
/*ro, opt, int, 卡号*/
    },
    "name": {
/*ro, opt, object, 持卡人姓名*/
        "@min": 1,
/*ro, opt, int, 持卡人姓名*/
        "@max": 32
/*ro, opt, int, 持卡人姓名*/
    },
    "picEnable": "true,false",
/*ro, opt, string, 是否带图片: false-不带图片*/
    "beginSerialNo": {
/*ro, opt, object, 起始流水号*/
        "@min": 1,
/*ro, opt, int, 起始流水号*/
        "@max": 1
/*ro, opt, int, 起始流水号*/
    },
    "endSerialNo": {
/*ro, opt, object, 结束流水号*/
        "@min": 1,
/*ro, opt, int, 结束流水号*/
        "@max": 1
/*ro, opt, int, 结束流水号*/
    },
    "employeeNoString": {
/*ro, opt, object, 工号（人员ID）*/
        "@min": 1,
/*ro, opt, int, 工号（人员ID）*/
        "@max": 32
/*ro, opt, int, 工号（人员ID）*/
    }
},
    "totalNum": {
/*ro, req, object, 满足搜索条件的事件总条数*/
        "@min": 1,
/*ro, opt, int, 满足搜索条件的事件总条数*/
        "@max": 1
/*ro, opt, int, 满足搜索条件的事件总条数*/
    }
}
}
}

```

69.45.10 获取清空事件及卡号联动参数能力

Request URL

GET /ISAPI/AccessControl/ClearEventCardLinkageCfg/capabilities?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "ClearEventCardLinkageCfg": {
    /*ro, opt, object, 清空事件及卡号联动参数*/
    "ClearFlags": {
      /*ro, opt, object, 清空标志*/
      "eventCardLinkage": "true,false"
      /*ro, req, string, 事件及卡号联动参数*/
    }
  }
}
```

69.45.11 配置事件优化

Request URL

PUT /ISAPI/AccessControl/EventOptimizationCfg?format=json

查询参数

无

请求报文

```
{
  "EventOptimizationCfg": {
    /*opt, object, 事件优化配置*/
    "enable": true,
    /*opt, bool, 是否启用事件优化*/
    "isCombinedLinkageEvents": true
    /*opt, bool, 是否启用联动事件合并*/
  }
}
```

响应报文

```
{
  "requestURL": "test",
  /*ro, opt, string, URI*/
  "statusCode": "test",
  /*ro, opt, string, 状态码*/
  "statusString": "test",
  /*ro, opt, string, 状态描述*/
  "subStatusCode": "test",
  /*ro, opt, string, 子状态码*/
  "errorCode": 1,
  /*ro, req, int, 错误码*/
  "errorMsg": "ok"
  /*ro, req, string, 错误信息*/
}
```

69.45.12 获取事件优化配置能力

Request URL

GET /ISAPI/AccessControl/EventOptimizationCfg/capabilities?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "EventOptimizationCfg": {
    /*ro, opt, object, 事件优化配置*/
    "enable": "true,false",
    /*ro, opt, string, 是否启用事件优化*/
    "isCombinedLinkageEvents": "true,false"
    /*ro, opt, string, 是否启用联动事件合并*/
  }
}
```

69.45.13 获取事件优化

Request URL

GET /ISAPI/AccessControl/EventOptimizationCfg?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "EventOptimizationCfg": {
    /*ro, opt, object, 事件优化配置*/
    "enable": true,
    /*ro, opt, bool, 是否启用事件优化*/
    "isCombinedLinkageEvents": true
    /*ro, opt, bool, 是否启用联动事件合并*/
  }
}
```

69.45.14 配置门禁事件存储参数

Request URL

PUT /ISAPI/AccessControl/AcsEvent/StorageCfg?format=json

查询参数

无

请求报文

```
{
  "EventStorageCfg": {
    /*wo, req, object, 事件存储配置*/
    "mode": "regular",
    /*wo, req, enum, 事件存储方式, subtype:string, [regular#定期删除旧事件,time#按指定时间删除旧事件,cycle#循环覆盖]*/
    "checkTime": "1970-01-01 00:00:00",
    /*wo, opt, string, 检查时间点 当存储方式为time时必须*/
    "period": 10
    /*wo, opt, int, 删除旧事件周期 当存储方式为regular时必须, unit:min, unitType:时间, desc:门禁产品支持的最小间隔是10分钟, 即10/20/30/40*/
  }
}
```

响应报文

```
{
  "statusCode": 1,
  /*ro, opt, int, 状态码, desc:无法用1表示时（1表示成功且无特殊状态）必须返回*/
  "statusString": "OK",
  /*ro, opt, string, 状态描述, range:[1,64], desc:无法用OK表示时（OK表示成功且无特殊状态）必须返回*/
  "subStatusCode": "ok",
  /*ro, opt, string, 子状态码, range:[1,64], desc:无法用ok表示（ok表示成功且无特殊状态）必须返回*/
  "errorCode": 1,
  /*ro, opt, int, 错误码, desc:当statusCode不为1时, 与subStatusCode对应*/
  "errorMsg": "ok"
  /*ro, opt, string, 错误信息, desc:当statusCode不为1时, 必须返回,解释信息在协议约束中, 允许设备在后续的版本迭代中, 进行优化丰富提升（不限制死）*/
}
```

69.45.15 获取门禁事件存储参数

Request URL

GET /ISAPI/AccessControl/AcsEvent/StorageCfg?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "EventStorageCfg": {
    /*ro, req, object, 事件存储配置*/
    "mode": "regular",
    /*ro, req, enum, 事件存储方式, subType:string, [regular#定期删除旧事件,time#按指定时间删除旧事件,cycle#循环覆盖]*/
    "checkTime": "1970-01-01 00:00:00",
    /*ro, opt, string, 检查时间点 当存储方式为time时必须*/
    "period": 10
    /*ro, opt, int, 删除旧事件周期 当存储方式为regular时必须, unit:min, unitType:时间, desc:单位分钟*/
  }
}
```

69.45.16 获取布防信息

Request URL

GET /ISAPI/AccessControl/DeployInfo

查询参数

无

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<DeployInfo xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, 布防信息, attr:version{req, string, 协议版本}-->
  <DeployList size="5">
    <!--ro, opt, array, 布防列表, subType:object, attr:size{req, int, 数组元素总数}-->
    <Content>
      <!--ro, opt, object, 报文-->
      <deployNo>
        <!--ro, req, int, 布防编号-->1
      </deployNo>
      <deployType>
        <!--ro, req, enum, 布防类型, subType:int, [0#客户端布防,1#实时布防,2#ISAPI客户端布防,3#ISAPI实时布防]-->1
      </deployType>
      <protocolType>
        <!--ro, opt, enum, 协议类型, subType:string, [HTTP#HTTP,HTTPS#HTTPS], dep:or,{$.DeployInfo.DeployList[*].Content.deployType,eq,2},
        {$.DeployInfo.DeployList[*].Content.deployType,eq,3}-->HTTP
      </protocolType>
      <ipAddr>
        <!--ro, req, string, IP地址-->test
      </ipAddr>
      <port>
        <!--ro, opt, int, 端口号, range:[1,65535]-->1
      </port>
      <eventType>
        <!--ro, opt, enum, 事件上报类型, subType:string, [AccessController#门禁事件,Consumer#交易事件,AccessControllerAndConsumer#门禁和交易事
        件,DepositRetrieveEvent#智能柜物品存取事件,DepositTimeoutEvent#智能柜物品寄存超时事件,CabinetStatusEvent#智能柜状态变化事件,CabinetAuthorityExpired#智能柜权
        限过期事件]-->AccessController
      </eventType>
    </Content>
  </DeployList>
</DeployInfo>
```

69.45.17 获取布防信息能力

Request URL

GET /ISAPI/AccessControl/DeployInfo/capabilities

查询参数

无

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<DeployInfo xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, 布防信息, attr:version{req, string, 协议版本}-->
  <DeployList size="5">
    <!--ro, opt, array, 布防列表, subType:object, attr:size{req, int, 数组元素总数}-->
    <Content>
      <!--ro, opt, object, 报文-->
      <deployNo min="1" max="10">
        <!--ro, req, int, 布防编号, attr:min{req, int, 最小值},max{req, int, 最大值}-->1
      </deployNo>
      <deployType opt="0,1,2,3">
        <!--ro, req, int, 布防类型, attr:opt{req, string, 取值范围}, desc:0-客户端布防,1-实时布防,2-ISAPI客户端布防,3-ISAPI实时布防-->1
      </deployType>
      <protocolType opt="HTTP,HTTPS">
        <!--ro, opt, enum, 协议类型, subType:string, [HTTP#HTTP,HTTPS#HTTPS], attr:opt{req, string, 取值范围}-->HTTP
      </protocolType>
      <ipAddr min="1" max="10">
        <!--ro, req, string, IP地址, attr:min{req, int, 最小值},max{req, int, 最大值}-->test
      </ipAddr>
      <port min="0" max="10">
        <!--ro, opt, int, 端口号, range:[1,65535], attr:min{req, int, 最小值},max{req, int, 最大值}-->1
      </port>
      <eventType opt="AccessController,Consumer,AccessControllerAndConsumer">
        <!--ro, opt, enum, 事件上报类型, subType:string, [AccessControlLer#门禁事件,Consumer#交易事件,AccessControlLerAndConsumer#门禁和交易事件,DepositRetrieveEvent#智能柜物品存取事件,DepositTimeoutEvent#智能柜物品寄存超时事件,CabinetStatusEvent#智能柜状态变化事件,CabinetAuthorityExpired#智能柜权限过期事件], attr:opt{req, string, 取值范围}-->AccessController
      </eventType>
    </Content>
  </DeployList>
</DeployInfo>
```

69.45.18 门禁事件

事件标识: AccessControllerEvent

```
{
  "ipAddress": "172.6.64.7",
  /*ro, req, string, 报警设备IPv4地址*/
  "ipv6Address": "1080:0:0:0:8:800:200C:417A",
  /*ro, opt, string, 报警设备IPv6地址*/
  "portNo": 80,
  /*ro, opt, int, 报警设备端口号*/
  "protocol": "HTTP",
  /*ro, opt, enum, 传输通信协议类型, subType:string, [HTTP#HTTP,HTTPS#HTTPS,EHome#EHome], desc:传输通信协议HTTP/HTTPS/EHome（萤石透传ISAPI的时候赋值HTTP; ISUP透传ISAPI的时候赋值EHome）protocolType作为历史遗留由平台兼容*/
  "macAddress": "01:17:24:45:D9:F4",
  /*ro, opt, string, MAC地址*/
  "channelID": 1,
  /*ro, opt, int, 触发报警的设备通道号, desc:触发的视频通道号（1、在SDK透传ISAPI协议的时候,上传的是私有协议对应的视频通道号;（2、在萤石透传ISAPI协议的时候,上传的是萤石协议对应的视频通道号;（3、在ISUP透传ISAPI协议的时候,上传的是ISUP协议对应的视频通道号;*/
  "dateTime": "2004-05-03T17:30:08+08:00",
  /*ro, req, datetime, 报警触发时间*/
  "activePostCount": 1,
  /*ro, req, int, 同一个报警已经上传的次数, desc:事件触发频次脉冲事件 定义:事件持续触发（按照设备的检测频率）,例如:移动检测。瞬时事件 定义:区分目标,一个目标触发一次,例如:人脸识别activePostCount 在脉冲事件 类型触发的时候,用于区分是否是同一触发源触发的事件。例如:移动检测,按照设备检测频率会一直上传;如果触发源发生了变化,这个时候计数就可以重新开始了。这个可以作为事件触发频次的方式来集成;*/
  "eventType": "AccessControllerEvent",
  /*ro, req, string, 事件类型, desc:AccessControllerEvent-门禁事件上传报警*/
  "eventState": "active",
  /*ro, req, enum, 事件状态, subType:string, [active#有效事件,inactive#无效事件], desc:针对持续性事件active - 表示有效事件（开始 或者 无过程状态也使用该字）; inactive - 表示无效事件（结束）; remark:在心跳类型下,该字段赋值（表示心跳数据,10s上传一次）;*/
  "eventDescription": "AccessControllerEvent",
  /*ro, req, string, 事件描述, desc:AccessControllerEvent-门禁事件上传报警*/
  "deviceID": "test0123",
  /*ro, opt, string, 即PUID, desc:在ISUP协议接入透传ISAPI事件信息中必须返回*/
  "AccessControllerEvent": {
    /*ro, req, object, 门禁事件信息*/
    "deviceName": "test",
    /*ro, opt, string, 网络操作的用户名*/
    "remoteHostAddr": "test",
    /*ro, opt, string, 远程主机地址*/
    "cardNo": "test",
    /*ro, opt, string, 卡号*/
    "cardType": 1,
    /*ro, opt, enum, 卡类型, subType:int, [1#普通卡,2#残疾人卡,3#黑名单卡,4#巡更卡,5#胁迫卡,6#超级卡,7#来宾卡,8#解除卡]*/
    "name": "test",
    /*ro, opt, string, 人员姓名*/
  }
```

```
"sex": "male",
/*ro, opt, enum, 性别, subType:string, [male#男,female#女]*/
"whiteListNo": 1,
/*ro, opt, int, 白名单单号*/
"reportChannel": 1,
/*ro, opt, enum, 报告上传通道, subType:int, [1#SDK布防上传,2#中心组1上传,3#中心组2上传,4#ISAPI布防上传,5#萤石上传]*/
"cardReaderKind": 1,
/*ro, opt, enum, 读卡器属于哪一类, subType:int, [1#IC读卡器,2#身份证读卡器,3#二维码读卡器,4#指纹头]*/
"cardReaderNo": 1,
/*ro, opt, int, 读卡器编号, step:1,
desc:当开启了副读卡器使能为主读卡器的功能后,副读卡器与主读卡器逻辑上认为是读卡器1
/ISAPI/AccessControl/AcsCfg?format=json的externalCardReaderEnabled字段*/
"doorNo": 1,
/*ro, opt, int, 门编号(楼层编号)*/
"verifyNo": 1,
/*ro, opt, int, 多重卡认证序号*/
"alarmInNo": 1,
/*ro, opt, int, 报警输入号*/
"alarmOutNo": 1,
/*ro, opt, int, 报警输出号*/
"caseSensorNo": 1,
/*ro, opt, int, 事件触发器编号*/
"RS485No": 1,
/*ro, opt, int, RS485通道号*/
"multiCardGroupNo": 1,
/*ro, opt, int, 群组编号*/
"accessChannel": 1,
/*ro, opt, int, 人员通道号*/
"deviceNo": 1,
/*ro, opt, int, 设备编号*/
"distractControlNo": 1,
/*ro, opt, int, 分控器编号*/
"employeeNo": 1,
/*ro, opt, int, 工号(人员ID)*/
"employeeNoString": "test",
/*ro, opt, string, 工号(人员ID), desc:对于设备来说,如果使用了工号(人员ID)字段,employeeNoString一定要传递,如果employeeNoString可转换为
employeeNo,那么该字段也要传递;对于上层平台或客户端来说,优先解析employeeNoString字段,如该字段为空,再考虑解析employeeNo字段*/
"employeeName": "test",
/*ro, opt, string, 人员名称, desc:该字段仅传显业务部信息发布项目使用,增加时报文上面的name字段丢失,导致多增加了人员名称字段,目前要求信息发布设备支持
两个name字段均上传*/
"localControllerID": 1,
/*ro, opt, int, 就地控制器编号, desc:0-门禁主机,1-64代表就地控制器*/
"InternetAccess": 1,
/*ro, opt, enum, 网口ID, subType:int, [1#上行网口1,2#上行网口2,3#下行网口1]*/
"type": 1,
/*ro, opt, enum, 防区类型, subType:int, [0#即时防区,1#24小时防区,2#延时防区,3#内部防区,4#钥匙防区,5#火警防区,6#周界防区,7#24小时无声防区,8#24小时辅助防区,9#24小时震动防区,10#门禁紧急开门防区,11#门禁紧急关门防区,255#无]*/
"MACAddr": "test",
/*ro, opt, string, 物理地址*/
"swipeCardType": 1,
/*ro, opt, enum, 刷卡类型, subType:int, [0#无效,1#二维码]*/
"serialNo": 1,
/*ro, opt, int, 事件流水号, range:[1,100000], desc:从1开始,递增加1,达到设备支持的最大值后循环覆盖*/
"channelControllerID": 1,
/*ro, opt, enum, 通道控制器ID, subType:int, [1#主通道控制器,2#从通道控制器]*/
"channelControllerLampID": 1,
/*ro, opt, int, 通道控制器灯板ID, range:[1,255]*/
"channelControllerIRAdaptorID": 1,
/*ro, opt, int, 通道控制器红外转接板ID, range:[1,255]*/
"channelControllerIREmitterID": 1,
/*ro, opt, int, 通道控制器红外对射ID, range:[1,255]*/
"userType": "normal",
/*ro, opt, enum, 人员类型, subType:string, [normal#普通人(主人),visitor#来宾(访客),blackList#黑名单人,administrators#管理员]*/
"currentVerifyMode": "cardAndPw",
/*ro, opt, enum, 读卡器当前验证方式, subType:string, [cardAndPw#刷卡+密码,card#刷卡,cardOrPw#刷卡或密码,fp#指纹,fpAndPw#指纹+密码,fpOrCard#指纹或刷卡,fpAndCard#指纹+刷卡,fpAndCardAndPw#指纹+刷卡+密码,faceOrFpOrCardOrPw#人脸或指纹或刷卡或密码,faceAndFp#人脸+指纹,faceAndPw#人脸+密码,faceAndCard#人脸+刷卡,face#人脸,employeeNoAndPw#工号+密码,fpOrPw#指纹或密码,employeeNoAndFp#工号+指纹,employeeNoAndFpAndPw#工号+指纹+密码,faceAndFpAndCard#人脸+指纹+刷卡,faceAndPwAndFp#人脸+密码+指纹,employeeNoAndFace#工号+人脸,faceOrFaceAndCard#人脸或人脸+刷卡,fpOrface#指纹或人脸,cardOrfaceOrPw#刷卡或人脸或密码,iris#虹膜,faceOrFpOrCardOrPwOrIris#人脸或指纹或刷卡或密码或虹膜,faceOrCardOrPwOrIris#人脸或刷卡或密码或虹膜], desc:人员验证方式的优先级高于读卡器验证方式*/
"currentEvent": true,
/*ro, opt, bool, 是否为实时事件*/
"QRCodeInfo": "test",
/*ro, opt, string, 二维码信息*/
"thermometryResult": "success",
/*ro, opt, enum, 测温结果, subType:string, [success#成功(默认),fail#失败]*/
"thermometryUnit": "celsius",
/*ro, opt, enum, 测温单位, subType:string, [celsius#摄氏度(默认),fahrenheit#华氏度,kelvin#开尔文]*/
"currTemperature": 36.1,
/*ro, opt, float, 人脸温度(精确到小数点后一位)*/
"isAbnormalTemperature": true,
/*ro, opt, bool, 人脸测温是否温度异常*/
"RegionCoordinates": {
/*ro, opt, object, 人脸温度坐标*/
"positionX": 0,
/*ro, opt, int, X坐标 归一化坐标0-1000, range:[0,1000]*/
"positionY": 0,
/*ro, opt, int, Y坐标 归一化坐标0-1000, range:[0,1000]*/
},
"remoteCheck": true,
/*ro, opt, bool, 是否需要远程核验(默认false)*/
"mask": "unknown",
/*ro, opt, enum, 是否戴口罩, subType:string, [unknown#未知,yes#戴口罩,no#不戴口罩]*/
"frontSerialNo": 1,
```

```

/*ro, opt, int, 上一条事件流水号, desc:右设备没返回该子段, 平台根据serialNo判断是否丢失事件; 右设备返回该子段, 平台根据该子段和serialNo字段共同判断是否丢失事件(主要用于解决报警订阅后导致serialNo不连续的情况)*/
"attendanceStatus": "checkIn",
/*ro, opt, enum, 考勤状态, subType:string, [checkIn#上班,checkOut#下班,breakOut#开始休息,breakIn#结束休息,overtimeIn#开始加班,overtimeOut#结束加班]*/
"label": "test",
/*ro, opt, string, 自定义考勤名称*/
"statusValue": 1,
/*ro, opt, int, 状态值*/
"pictureURL": "test",
/*ro, opt, string, 抓拍图片URL, range:[0,256]*/
"visibleLightURL": "test",
/*ro, opt, string, 热成像相机可见光图片URL, range:[0,256]*/
"thermalURL": "test",
/*ro, opt, string, 热成像图片URL, range:[0,256]*/
"faceBasemapURL": "test",
/*ro, opt, string, 人脸底图URL, range:[0,256], desc:仅HEOP使用, 用于人员认证成功后显示人脸图片。*/
"picturesNumber": 1,
/*ro, opt, int, 图片数量*/
"unlockType": "password",
/*ro, opt, enum, 开锁类型, subType:string, [password#密码开锁,hijacking#劫持开锁,card#刷卡开锁,householder#户主开锁,centerplatform#中心平台开锁,bluetooth#蓝牙开锁,qrcode#二维码开锁,face#人脸开锁,fingerprint#指纹开锁], desc:对应类型(MINOR_UNLOCK_RECORD)时返回*/
"classroomId": "test",
/*ro, opt, string, 教室UUID*/
"classroomName": "test",
/*ro, opt, string, 教室名称*/
"analysisModule": "signageApp",
/*ro, opt, enum, 分析模块, subType:string, [signageApp#班牌APP,faceSDK#人脸SDK], desc:本字段不返回默认采用班牌APP上报*/
"customInfo": "test",
/*ro, opt, string, 自定义信息*/
"helmet": "unknown",
/*ro, opt, enum, 是否戴安全帽, subType:string, [unknown#未知,yes#戴安全帽,no#不戴安全帽]*/
"purePwdVerifyEnable": true,
/*ro, opt, bool, 是否支持纯密码开门(人员信息中的-password字段), desc:*纯密码方案:
*a认证方式中的"或密码"为人员密码
*b设备不对人员密码的重复性进行校验, 需要上层平台自行保证密码的唯一性
*c设备本地不能对人员密码进行操作(增删改查)*/
"appType": "attendance",
/*ro, opt, enum, 应用类型(信发产品使用), subType:string, [attendance#考勤应用,signIn#签到应用]*/
"HealthInfo": {
/*ro, opt, object, 健康信息*/
"healthCode": 1,
/*ro, opt, enum, 健康码状态, subType:int, [0#未请求,1#未申领,2#绿码,3#黄码,4#红码,5#无此人员,6#其他错误信息(如接口异常导致查询失败),7#查询健康码超时]*/
"NADCode": 1,
/*ro, opt, enum, 核酸检测结果, subType:int, [0#未查询到核酸检测结果,1#核酸检测阴性(代表正常),2#核酸检测阳性(代表确诊),3#核酸检测有效期已过,4#查询核酸结果失败]*/
"NADMsg": "test",
/*ro, opt, string, 核酸检测信息, range:[0,64], desc:空字符串表示查询到核酸检测信息失败*/
"NADTime": 1,
/*ro, opt, enum, 核酸检测时间, subType:int, [1#24小时内,2#48小时内,3#48小时外]*/
"travelCode": 1,
/*ro, opt, enum, 行程编码, subType:int, [0#14天内一直在当地,1#14天内离开过当地,2#14天内到过疫区,3#其他,4#查询行程信息失败]*/
"travelInfo": "test",
/*ro, opt, string, 行程信息, desc:空字符串表示查询到行程信息失败*/
"vaccineStatus": 1,
/*ro, opt, enum, 打疫苗状态, subType:int, [0#未打疫苗,1#部分注射疫苗,2#已完成疫苗,3#查询疫苗信息失败,4#已完成加强针疫苗]*/
"vaccineNum": 1,
/*ro, opt, int, 打疫苗针数, step:1*/
"vaccineMsg": "test",
/*ro, opt, string, 打疫苗信息, range:[0,64], desc:空字符串表示查询到疫苗信息失败*/
"ANTCode": 1,
/*ro, opt, enum, 抗原检测状态, subType:int, [0#未查询到抗原检测结果,1#抗原检测阴性,2#抗原检测阳性,3#抗原检测有效期已过,4#抗原检测无效]*/
"ANTMsg": "test",
/*ro, opt, string, 抗原检测信息, range:[0,64], desc:空字符串表示查询到抗原检测信息失败*/
},
"PhysicalInfo": {
/*ro, opt, object, 身体信息, desc:明眸对接体脂称, 认证通过后明眸获取体脂称信息*/
"weight": 7000,
/*ro, opt, int, 体重, unit:kg, unitType:质量, 由于测量值存在小数点, 实际值为体重(kg)*100*/
"height": 18000,
/*ro, opt, int, 身高, unit:cm, unitType:长度, 由于测量值存在小数点, 实际值为身高(cm)*100*/
},
"meetingID": "test",
/*ro, opt, string, 会议编号UUID, range:[1,32]*/
"PersonInfoExtends": [
/*ro, opt, array, 人员信息扩展, subType:object, desc:人员在设备UI界面的扩展信息显示
明眸当前仅支持一个value, 不支持id字段, 用于作为工号显示
智能主机按标准实现*/
{
"id": 1,
/*ro, opt, int, 人员信息扩展序号, range:[1,32], desc:与/ISAPI/AccessControl/personInfoExtendName?format=json中的id对应, 用于关联value的name信息
id不存在时, 根据数组顺序默认从1开始*/
"value": "test",
/*ro, opt, string, 人员信息扩展内容*/
}
],
"customPrompt": "test",
/*ro, opt, string, 自定义提示语, range:[1,128], desc:认证成功或失败或为陌生人时, 上报自定义的显示提示语到UI上进行展示*/
"FaceRect": {
/*ro, opt, object, 人脸矩形框, desc:屏幕左上角为坐标原点*/

```

```
"height": 1.000,
/*ro, req, float, 高度, range:[0.000,1.000]*/
"width": 1.000,
/*ro, req, float, 宽度, range:[0.000,1.000]*/
"x": 0.000,
/*ro, req, float, 区域左上角顶点的水平坐标, range:[0.000,1.000]*/
"y": 0.000
/*ro, req, float, 区域左上角顶点的垂直坐标, range:[0.000,1.000]*/
},
"faceSimilarity": 90,
/*ro, opt, int, 人脸相似度, range:[0,100], desc:抓拍照与设备本地存储的人脸图片进行比对。*/
"faceRecognitionDistance": 0.1,
/*ro, opt, float, 人脸识别距离, unit:m, unitType:长度, desc:保留小数点后1位*/
"eyesDistance": 20,
/*ro, opt, int, 眼间距, range:[0,100], step:1, desc:指人脸识别时, 双眼间距在设备屏幕上与屏幕宽度的比值*/
"faceRecognitionFailedReason": "attackBlacklist",
/*ro, opt, enum, 人脸验证失败原因, subType:string, [attackBlacklist#攻击黑名单],
desc:攻击黑名单: 人脸验证时, 开启真人检测后, 一直拿图片认证。(会有极小概率认证通过)。
仅次类型0x4c有效。*/
"currentAuthenticationTimes": 1,
/*ro, opt, int, 当前权限计划时间段已认证次数, range:[0,255], step:1, desc:若认证通过, 认证次数加1*/
"allowAuthenticationTimes": 1,
/*ro, opt, int, 当前权限计划时间段允许的认证次数, range:[0,255], step:1*/
"LocalAttendanceData": {
/*ro, opt, object, 本地考勤数据, desc:认证成功后, hicare推送给UI当前打卡人员的考勤信息*/
"attendanceResult": [
/*ro, opt, array, 本地考勤结果, subType:object*/
{
"date": "1970-01-01",
/*ro, opt, date, 日期*/
"week": 1,
/*ro, opt, enum, 星期, subType:int, [1#星期一,2#星期二,3#星期三,4#星期四,5#星期五,6#星期六,7#星期日]*/
"personalAttendanceStatus": "normal"
/*ro, opt, enum, 个人考勤状态, subType:string, [normal#正常,late#迟到,earlyLeave#早退,absence#缺勤,absenteeism#旷工,lateAndEarlyLeave#迟
到且早退,attendancing#考勤中]*/
}
]
},
"hasRecord": true
/*ro, opt, bool, 是否存在录像, desc:若存在录像, 可以通过查询门禁事件接口来查询录像*/
},
"URLCertificationType": "digest"
/*ro, opt, enum, 图片url认证方式, subType:string, [no#无,digest#摘要认证], desc:no-无 (这个是针对武汉云存储协议); digest-摘要认证 (这个针对设备本地存储
返回URL的方式, 设备例如NVR/DVR)*/
}
```

参数名称	参数值	参数类型(Content-Type)	Content-ID	文件名称(filename)	描述
AccessControllerEvent	[报文内容]	application/json	--	--	--
Picture	[图片二进制数据]	image/jpeg	pictureImage	Picture.jpg	抓拍图片数据
VisibleLight	[图片二进制数据]	image/jpeg	visibleLight_image	VisibleLight.jpg	热成像相机可见光图片数据
Thermal	[图片二进制数据]	image/jpeg	thermal_image	Thermal.jpg	热成像图片数据

说明： 此协议通过表格式传输，表单框架详细说明见章节4.5.1.4，如下实例：

```
--<frontier>
Content-Disposition: form-data; name=参数名称;filename=文件名称
Content-Type: 参数类型
Content-Length: ****
Content-ID: 内容ID
参数值
```

- 参数名称：表单单元Headers中Content-Disposition的name属性，表示此表单单元的名称
- 参数类型（Content-Type）： 表单单元Headers中Content-Type属性
- 文件名称（filename）： 表单单元Headers中Content-Disposition的filename属性，仅在本表单单元传输数据为文件类型存在，表示表单单元Body的文件名
- 参数值： 表单单元的body内容

69.46 门禁点状态计划管理

69.46.1 配置门状态周计划参数

Request URL

PUT /ISAPI/AccessControl/DoorStatusWeekPlanCfg/<weekPlanID>?format=json

查询参数

参数名称	参数类型	描述
weekPlanID	string	周计划编号，从1开始，设备支持的最大值从能力集中获取

请求报文

```
{
  "DoorStatusWeekPlanCfg": {
    /*opt, object, 门状态周计划*/
    "enable": true,
    /*req, bool, 门状态周计划参数使能, desc:true-使能, false-不使能*/
    "WeekPlanCfg": [
      /*req, array, 周计划参数, subType:object*/
      {
        "week": "Monday",
        /*req, enum, 星期, subType:string, [Monday#星期一,Tuesday#星期二,Wednesday#星期三,Thursday#星期四,Friday#星期五,Saturday#星期六,Sunday#星期日]*/
        "id": 1,
        /*req, int, 时间段编号, range:[1,8]*/
        "enable": true,
        /*req, bool, 周计划使能*/
        "doorStatus": "remainClosed",
        /*req, enum, 门状态计划, subType:string, [remainOpen#常开,remainClosed#常闭,normal#普通,sleep#休眠,invalid#无效,induction#感应模式,barrierFree#无障碍模式],
        desc:感应模式: 当对应方向感应到人员通行时, 才打开; 当前只支持红外感应开门
        无障碍模式: 门翼保持开启, 用户通行时如果未认证权限门翼会关闭且进行报警*/
        "TimeSegment": {
          /*req, object, 时间节点*/
          "beginTime": "00:00:00",
          /*req, time, 开始时间点, desc:设备本地时间*/
          "endTime": "10:00:00"
          /*req, time, 结束时间点, desc:设备本地时间*/
        }
      }
    ]
  }
}
```

响应报文

```
{
  "statusCode": 1,
  /*ro, opt, int, 状态码, desc:无法用1表示时（1表示成功且无特殊状态）必须返回*/
  "statusString": "OK",
  /*ro, opt, string, 状态描述, range:[1,64], desc:无法用OK表示时（OK表示成功且无特殊状态）必须返回*/
  "subStatusCode": "ok",
  /*ro, opt, string, 子状态码, range:[1,64], desc:无法用ok表示（ok表示成功且无特殊状态）必须返回*/
  "errorCode": 1,
  /*ro, opt, int, 错误码, desc:当statusCode不为1时，与subStatusCode对应*/
  "errorMsg": "ok"
  /*ro, opt, string, 错误信息, desc:当statusCode不为1时，必须返回,解释信息在协议约束中，允许设备在后续的版本迭代中，进行优化丰富提升（不限制死）*/
}
```

69.46.2 获取门状态假日计划参数

Request URL

GET /ISAPI/AccessControl/DoorStatusHolidayPlanCfg/<holidayPlanID>?format=json

查询参数

参数名称	参数类型	描述
holidayPlanID	string	假日计划编号，从1开始，设备支持的最大值从能力集中获取

请求报文

无

响应报文

```

{
  "DoorStatusHolidayPlanCfg": {
    /*ro, req, object, 门状态假日计划参数*/
    "enable": true,
    /*ro, req, bool, 门状态假日计划参数使能, desc:true-使能, false-不使能*/
    "beginDate": "2017-10-01",
    /*ro, req, date, 假日开始日期*/
    "endDate": "2017-10-08",
    /*ro, req, date, 假日结束日期*/
    "HolidayPlanCfg": [
      /*ro, req, array, 假日计划参数, subType:object*/
      {
        "id": 1,
        /*ro, req, int, 时间段编号, range:[1,8]*/
        "enable": true,
        /*ro, req, bool, 假日计划参数使能, desc:true-使能, false-不使能*/
        "doorStatus": "remainClosed",
        /*ro, req, enum, 门状态计划, subType:string, [remainOpen#常开,remainClosed#常闭,normal#普通,sleep#休眠,invalid#无效,induction#感应模式,barrierFree#无障碍模式],
        desc:感应模式: 当对应方向感应到人员通行时, 才打开; 当前只支持红外感应开门
        无障碍模式: 门翼保持开启, 用户通行时如果未认证权限门翼会关闭且进行报警*/
        "TimeSegment": {
          /*ro, opt, object, 时间节点*/
          "beginTime": "00:00:00",
          /*ro, req, time, 开始时间点, desc:设备本地时间*/
          "endTime": "10:00:00"
          /*ro, req, time, 结束时间点, desc:设备本地时间*/
        }
      }
    ]
  }
}

```

69.46.3 获取门状态周计划参数配置能力

Request URL

GET /ISAPI/AccessControl/DoorStatusWeekPlanCfg/capabilities?format=json

查询参数

无

请求报文

无

响应报文

```

{
  "DoorStatusWeekPlanCfg": {
    /*ro, opt, object, 门状态周计划参数*/
    "planNo": {
      /*ro, opt, object, 周计划编号*/
      "@min": 1,
      /*ro, opt, int, 周计划编号最小值*/
      "@max": 16
      /*ro, opt, int, 周计划编号最大值*/
    },
    "enable": "true,false",
    /*ro, opt, string, 门状态周计划参数使能, desc:true-使能, false-不使能*/
    "WeekPlanCfg": {
      /*ro, opt, object, 周计划*/
      "maxSize": 56,
      /*ro, opt, int, 周计划最大值*/
      "week": {
        /*ro, opt, object, 周名称枚举*/
        "@opt": "Monday,Tuesday,Wednesday,Thursday,Friday,Saturday,Sunday"
        /*ro, opt, string, 周名称枚举值, desc:Monday-星期一, Tuesday-星期二, Wednesday-星期三, Thursday-星期四, Friday-星期五, Saturday-星期六, Sunday-星期日*/
      },
      "id": {
        /*ro, opt, object, 周计划编号*/
        "@min": 1,
        /*ro, opt, int, 周计划编号最小值*/
        "@max": 8
        /*ro, opt, int, 周计划编号最大值*/
      },
      "enable": "true,false",
      /*ro, opt, string, 周计划使能: , desc:true-使能, false-不使能*/
      "doorStatus": {
        /*ro, opt, object, 门状态计划*/
        "@opt": "remainOpen,remainClosed,normal,sleep,invalid,induction,barrierFree"
        /*ro, opt, string, 门状态计划枚举值, desc:remainOpen-常开, remainClosed-常闭, normal-普通, sleep-休眠, invalid-无效,induction-感应模式,barrierFree-无障碍模式*/
      },
      "TimeSegment": {
        /*ro, opt, object, 时间节点*/
        "beginTime": "00:00:00",
        /*ro, opt, time, 开始时间点, desc:设备本地时间*/
        "endTime": "10:00:00",
        /*ro, opt, time, 结束时间点, desc:设备本地时间*/
        "validUnit": "minute"
        /*ro, opt, enum, 时间点精度, subType:string, [hour#小时,minute#分钟,second#秒], desc:不返回该字段, 则代表精度为分钟*/
      }
    }
  }
}

```

69.46.4 获取门状态假日计划参数配置能力

Request URL

GET /ISAPI/AccessControl/DoorStatusHolidayPlanCfg/capabilities?format=json

查询参数

无

请求报文

无

响应报文

```

{
  "DoorStatusHolidayPlanCfg": {
    /*ro, opt, object, 门状态假日计划*/
    "planNo": {
      /*ro, opt, object, 假日计划编号*/
      "@min": 1,
      /*ro, opt, int, 假日计划编号最小值*/
      "@max": 16
      /*ro, opt, int, 假日计划编号最大值*/
    },
    "enable": "true,false",
    /*ro, opt, string, 门状态假日计划使能, desc:true-使能, false-不使能*/
    "beginDate": "1970-01-01",
    /*ro, opt, date, 假日开始日期*/
    "endDate": "1971-01-01",
    /*ro, opt, date, 假日结束日期*/
    "HolidayPlanCfg": {
      /*ro, opt, object, 假日计划参数*/
      "maxSize": 8,
      /*ro, opt, int, 假日计划参数最大值*/
      "id": {
        /*ro, opt, object, 时间段编号*/
        "@min": 1,
        /*ro, opt, int, 时间段编号最小值*/
        "@max": 8
        /*ro, opt, int, 时间段编号最大值*/
      },
      "enable": "true,false",
      /*ro, opt, string, 假日计划参数使能, desc:true-使能, false-不使能*/
      "doorStatus": {
        /*ro, opt, object, 门状态计划*/
        "@opt": "remainOpen,remainClosed,normal,sleep,invlid,induction,barrierFree"
        /*ro, opt, string, 门状态计划枚举值, desc:remainOpen-常开, remainClosed-常闭, normal-普通, sleep-休眠, invlid-无效,induction#感应模式,barrierFree#无障碍模式*/
      },
      "TimeSegment": {
        /*ro, opt, object, 时间节点*/
        "beginTime": "00:00:00",
        /*ro, opt, time, 开始时间点, desc:设备本地时间*/
        "endTime": "00:00:00",
        /*ro, opt, time, 结束时间点, desc:设备本地时间*/
        "validUnit": "minute"
        /*ro, opt, enum, 时间点精度, subType:string, [hour#小时,minute#分钟,second#秒], desc:不返回该字段, 则代表精度为分钟*/
      }
    }
  }
}

```

69.46.5 获取门状态计划模板参数配置能力

Request URL

GET /ISAPI/AccessControl/DoorStatusPlanTemplate/capabilities?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "DoorStatusPlanTemplate": {
    /*ro, opt, object, 计划模板*/
    "templateNo": {
      /*ro, opt, object, 计划模板编号*/
      "@min": 1,
      /*ro, opt, int, 计划模板编号最小值*/
      "@max": 16
      /*ro, opt, int, 计划模板编号最大值*/
    },
    "enable": "true,false",
    /*ro, opt, string, 是否启用, desc:true-启用, false-不启用*/
    "templateName": {
      /*ro, opt, object, 模板名称长度*/
      "@min": 1,
      /*ro, opt, int, 模板名称长度最小值*/
      "@max": 32
      /*ro, opt, int, 模板名称长度最大值*/
    },
    "weekPlanNo": {
      /*ro, opt, object, 周计划编号*/
      "@min": 1,
      /*ro, opt, int, 周计划编号最小值*/
      "@max": 16
      /*ro, opt, int, 周计划编号最大值*/
    },
    "holidayGroupNo": {
      /*ro, opt, object, 假日组编号*/
      "@min": 1,
      /*ro, opt, int, 假日组编号最小值*/
      "@max": 16
      /*ro, opt, int, 假日组编号最大值*/
    }
  }
}
```

69.46.6 获取门状态周计划参数

Request URL

GET /ISAPI/AccessControl/DoorStatusWeekPlanCfg/<weekPlanID>?format=json

查询参数

参数名称	参数类型	描述
weekPlanID	string	周计划编号，从1开始，设备支持的最大值从能力集中获取

请求报文

无

响应报文

```
{
  "DoorStatusWeekPlanCfg": {
    /*ro, opt, object, 门状态周计划*/
    "enable": true,
    /*ro, req, bool, 门状态周计划参数使能, desc:true-使能, false-不使能*/
    "WeekPlanCfg": [
      /*ro, req, array, 周计划参数, subType:object*/
      {
        "week": "Monday",
        /*ro, req, enum, 星期, subType:string, [Monday#星期一,Tuesday#星期二,Wednesday#星期三,Thursday#星期四,Friday#星期五,Saturday#星期六,Sunday#星期日]*/
        "id": 1,
        /*ro, req, int, 时间段编号, range:[1,8]*/
        "enable": true,
        /*ro, req, bool, 周计划使能*/
        "doorStatus": "remainClosed",
        /*ro, req, enum, 门状态计划, subType:string, [remainOpen#常开,remainClosed#常闭,normal#普通,sleep#休眠,invalid#无效,induction#感应模式,barrierFree#无障碍模式],
        desc:感应模式: 当对应方向感应到人员通行时, 才打开; 当前只支持红外感应开门
          无障碍模式: 门翼保持开启, 用户通行时如果未认证权限门翼会关闭且进行报警*/
        "TimeSegment": {
          /*ro, req, object, 时间节点*/
          "beginTime": "00:00:00",
          /*ro, req, time, 开始时间点, desc:设备本地时间*/
          "endTime": "10:00:00"
          /*ro, req, time, 结束时间点, desc:设备本地时间*/
        }
      }
    ]
  }
}
```

69.46.7 配置门状态假日计划参数

Request URL

PUT /ISAPI/AccessControl/DoorStatusHolidayPlanCfg/<holidayPlanID>?format=json

查询参数

参数名称	参数类型	描述
holidayPlanID	string	假日计划编号，从1开始，设备支持的最大值从能力集中获取

请求报文

```
{
  "DoorStatusHolidayPlanCfg": {
    /*ro, req, object, 门状态假日计划参数*/
    "enable": true,
    /*ro, req, bool, 门状态假日计划参数使能, desc:true-使能, false-不使能*/
    "beginDate": "2017-10-01",
    /*ro, req, date, 假日开始日期*/
    "endDate": "2017-10-08",
    /*ro, req, date, 假日结束日期*/
    "HolidayPlanCfg": [
      /*ro, req, array, 假日计划参数, subType:object*/
      {
        "id": 1,
        /*ro, req, int, 时间段编号, range:[1,8]*/
        "enable": true,
        /*ro, req, bool, 假日计划参数使能, desc:true-使能, false-不使能*/
        "doorStatus": "remainClosed",
        /*ro, req, enum, 门状态计划, subType:string, [remainOpen#常开,remainClosed#常闭,normal#普通,sleep#休眠,invalid#无效,induction#感应模式,barrierFree#无障碍模式],
        desc:感应模式: 当对应方向感应到人员通行时, 才打开; 当前只支持红外感应开门
          无障碍模式: 门翼保持开启, 用户通行时如果未认证权限门翼会关闭且进行报警*/
        "TimeSegment": {
          /*ro, opt, object, 时间节点*/
          "beginTime": "00:00:00",
          /*ro, req, time, 开始时间点, desc:设备本地时间*/
          "endTime": "10:00:00"
          /*ro, req, time, 结束时间点, desc:设备本地时间*/
        }
      }
    ]
  }
}
```

响应报文

```
{
  "statusCode": 1,
  /*ro, opt, int, 状态码, desc:无法用1表示时（1表示成功且无特殊状态）必须返回*/
  "statusString": "OK",
  /*ro, opt, string, 状态描述, range:[1,64], desc:无法用OK表示时（OK表示成功且无特殊状态）必须返回*/
  "subStatusCode": "ok",
  /*ro, opt, string, 子状态码, range:[1,64], desc:无法用ok表示（ok表示成功且无特殊状态）必须返回*/
  "errorCode": 1,
  /*ro, opt, int, 错误码, desc:当statusCode不为1时，与subStatusCode对应*/
  "errorMsg": "ok"
  /*ro, opt, string, 错误信息, desc:当statusCode不为1时，必须返回,解释信息在协议约束中，允许设备在后续的版本迭代中，进行优化丰富提升（不限制死）*/
}
```

69.46.8 配置门状态计划模板参数

Request URL

PUT /ISAPI/AccessControl/DoorStatusPlanTemplate/<planTemplateID>?format=json

查询参数

参数名称	参数类型	描述
planTemplateID	string	计划模板编号，从1开始，设备支持的最大值从能力集中获取

请求报文

```
{
  "DoorStatusPlanTemplate": {
    /*ro, opt, object, 门状态计划模板*/
    "enable": true,
    /*ro, req, bool, 是否启用, desc:true-启用, false-不启用*/
    "templateName": "test",
    /*ro, req, string, 模板名称*/
    "weekPlanNo": 1,
    /*ro, req, int, 周计划编号*/
    "holidayGroupNo": "1,3,5"
    /*ro, req, string, 假日组编号, desc:该字段配置可配置空值, 即"holidayPlanNo" : ""*/
  }
}
```

响应报文

```
{
  "statusCode": 1,
  /*ro, opt, int, 状态码, desc:无法用1表示时（1表示成功且无特殊状态）必须返回*/
  "statusString": "OK",
  /*ro, opt, string, 状态描述, range:[1,64], desc:无法用OK表示时（OK表示成功且无特殊状态）必须返回*/
  "subStatusCode": "ok",
  /*ro, opt, string, 子状态码, range:[1,64], desc:无法用ok表示（ok表示成功且无特殊状态）必须返回*/
  "errorCode": 1,
  /*ro, opt, int, 错误码, desc:当statusCode不为1时，与subStatusCode对应*/
  "errorMsg": "ok"
  /*ro, opt, string, 错误信息, desc:当statusCode不为1时，必须返回,解释信息在协议约束中，允许设备在后续的版本迭代中，进行优化丰富提升（不限制死）*/
}
```

69.46.9 获取门状态计划模板参数

Request URL

GET /ISAPI/AccessControl/DoorStatusPlanTemplate/<planTemplateID>?format=json

查询参数

参数名称	参数类型	描述
planTemplateID	string	计划模板编号，从1开始，设备支持的最大值从能力集中获取

请求报文

无

响应报文

```
{
  "DoorStatusPlanTemplate": {
    /*ro, opt, object, 门状态计划模板*/
    "enable": true,
    /*ro, req, bool, 是否启用, desc:true-启用, false-不启用*/
    "templateName": "test",
    /*ro, req, string, 模板名称*/
    "weekPlanNo": 1,
    /*ro, req, int, 周计划编号*/
    "holidayGroupNo": "1,3,5"
    /*ro, req, string, 假日组编号, desc:该字段配置可配置空值, 即"holidayPlanNo" : ""*/
  }
}
```

69.46.10 获取门状态计划参数配置能力

Request URL

GET /ISAPI/AccessControl/DoorStatusPlan/capabilities?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "DoorStatusPlan": {
    /*ro, opt, object, 门状态计划*/
    "doorNo": {
      /*ro, opt, object, 门编号*/
      "@min": 1,
      /*ro, opt, int, 门编号最小值*/
      "@max": 4
      /*ro, opt, int, 门编号最大值*/
    },
    "templateNo": {
      /*ro, opt, object, 计划模板编号*/
      "@min": 1,
      /*ro, opt, int, 计划模板编号最小值*/
      "@max": 16
      /*ro, opt, int, 计划模板编号最大值*/
    }
  }
}
```

69.46.11 获取门状态假日组参数配置能力

Request URL

GET /ISAPI/AccessControl/DoorStatusHolidayGroupCfg/capabilities?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "DoorStatusHolidayGroupCfg": {
    /*ro, opt, object, 假日组编号节点*/
    "groupNo": {
      /*ro, opt, object, 假日组编号*/
      "@min": 1,
      /*ro, opt, int, 假日组编号最小值*/
      "@max": 16
      /*ro, opt, int, 假日组编号最大值*/
    },
    "enable": "true,false",
    /*ro, opt, string, 是否启用, desc:true-启用*/
    "groupName": {
      /*ro, opt, object, 假日组名称长度*/
      "@min": 1,
      /*ro, opt, int, 假日组名称长度最小值*/
      "@max": 32
      /*ro, opt, int, 假日组名称长度最大值*/
    },
    "holidayPlanNo": {
      /*ro, opt, object, 假日组计划编号*/
      "@min": 1,
      /*ro, opt, int, 假日组计划编号最小值*/
      "@max": 16
      /*ro, opt, int, 假日组计划编号最大值*/
    }
  }
}
```

69.46.12 获取门状态假日组参数

Request URL

GET /ISAPI/AccessControl/DoorStatusHolidayGroupCfg/<holidayGroupID>?format=json

查询参数

参数名称	参数类型	描述
holidayGroupID	string	假日组编号，从1开始，设备支持的最大值从能力集中获取

请求报文

无

响应报文

```
{
  "DoorStatusHolidayGroupCfg": {
    /*ro, opt, object, 门状态假日组*/
    "enable": true,
    /*ro, req, bool, 是否启用: true-启用*/
    "groupName": "test",
    /*ro, req, string, 假日组名称*/
    "holidayPlanNo": "1,3,5"
    /*ro, req, string, 假日组计划编号*/
  }
}
```

69.46.13 配置门状态计划参数

Request URL

PUT /ISAPI/AccessControl/DoorStatusPlan/<doorID>?format=json

查询参数

参数名称	参数类型	描述
doorID	string	门编号，从1开始，设备支持的最大值从能力集中获取

请求报文

```
{
  "DoorStatusPlan": {
    /*req, object, 计划模板*/
    "templateNo": 1
    /*req, int, 计划模板编号, desc:0表示取消关联, 恢复默认状态（普通状态）*/
  }
}
```

响应报文

```
{
  "statusCode": 1,
  /*ro, opt, int, 状态码, desc:无法用1表示时（1表示成功且无特殊状态）必须返回*/
  "statusString": "OK",
  /*ro, opt, string, 状态描述, range:[1,64], desc:无法用OK表示时（OK表示成功且无特殊状态）必须返回*/
  "subStatusCode": "ok",
  /*ro, opt, string, 子状态码, range:[1,64], desc:无法用ok表示（ok表示成功且无特殊状态）必须返回*/
  "errorCode": 1,
  /*ro, opt, int, 错误码, desc:当statusCode不为1时, 与subStatusCode对应*/
  "errorMsg": "ok"
  /*ro, opt, string, 错误信息, desc:当statusCode不为1时, 必须返回,解释信息在协议约束中, 允许设备在后续的版本迭代中, 进行优化丰富提升（不限制死）*/
}
```

69.46.14 获取门状态计划参数

Request URL

GET /ISAPI/AccessControl/DoorStatusPlan/<doorID>?format=json

查询参数

参数名称	参数类型	描述
doorID	string	门编号，从1开始，设备支持的最大值从能力集中获取

请求报文

无

响应报文

```
{
  "DoorStatusPlan": {
    /*ro, req, object, 计划模板*/
    "templateNo": 1
    /*ro, req, int, 计划模板编号, desc:0表示取消关联, 恢复默认状态（普通状态）*/
  }
}
```

69.46.15 配置清空门禁计划参数

Request URL

PUT /ISAPI/AccessControl/ClearPlansCfg?format=json

查询参数

无

请求报文

```
{
  "ClearPlansCfg": {
    /*opt, object, 清空门禁计划参数节点*/
    "ClearFlags": {
      /*opt, object, 清空门禁计划参数节点*/
      "doorStatusWeekPlan": true,
      /*opt, bool, 门状态周计划*/
      "cardReaderWeekPlan": true,
      /*opt, bool, 读卡器验证周计划*/
      "userRightWeekPlan": true,
      /*opt, bool, 人员权限周计划*/
      "doorStatusHolidayPlan": true,
      /*opt, bool, 门状态假日计划参数*/
      "cardReaderHolidayPlan": true,
      /*opt, bool, 读卡器假日计划参数*/
      "userRightHolidayPlan": true,
      /*opt, bool, 人员权限假日计划参数*/
      "doorStatusHolidayGroup": true,
      /*opt, bool, 门状态假日组参数*/
      "cardReaderHolidayGroup": true,
      /*opt, bool, 读卡器验证方式假日组参数*/
      "userRightHolidayGroup": true,
      /*opt, bool, 人员权限假日组参数*/
      "doorStatusTemplate": true,
      /*opt, bool, 门状态计划模板参数*/
      "cardReaderTemplate": true,
      /*opt, bool, 读卡器验证方式计划模板参数*/
      "userRightTemplate": true,
      /*opt, bool, 人员权限计划模板参数*/
    }
  }
}
```

响应报文

```
{
  "statusCode": 1,
  /*ro, opt, int, 状态码, desc:无法用1表示时（1表示成功且无特殊状态）必须返回*/
  "statusString": "OK",
  /*ro, opt, string, 状态描述, range:[1,64], desc:无法用OK表示时（OK表示成功且无特殊状态）必须返回*/
  "subStatusCode": "ok",
  /*ro, opt, string, 子状态码, range:[1,64], desc:无法用ok表示（ok表示成功且无特殊状态）必须返回*/
  "errorCode": 1,
  /*ro, opt, int, 错误码, desc:当statusCode不为1时, 与subStatusCode对应*/
  "errorMsg": "ok",
  /*ro, opt, string, 错误信息, desc:当statusCode不为1时, 必须返回,解释信息在协议约束中, 允许设备在后续的版本迭代中, 进行优化丰富提升（不限制死）*/
}
```

69.46.16 获取清空门禁计划参数能力

Request URL

GET /ISAPI/AccessControl/ClearPlansCfg/capabilities?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "ClearPlansCfg": {
    /*ro, req, object, 清空门禁计划参数能力节点*/
    "ClearFlags": {
      /*ro, opt, object, 清空门禁计划参数能力节点*/
      "doorStatusWeekPlan": "true,false",
      /*ro, opt, string, 门状态周计划*/
      "cardReaderWeekPlan": "true,false",
      /*ro, opt, string, 读卡器验证周计划*/
      "userRightWeekPlan": "true,false",
      /*ro, opt, string, 人员权限周计划*/
      "doorStatusHolidayPlan": "true,false",
      /*ro, opt, string, 门状态假日计划参数*/
      "cardReaderHolidayPlan": "true,false",
      /*ro, opt, string, 读卡器假日计划参数*/
      "userRightHolidayPlan": "true,false",
      /*ro, opt, string, 人员权限假日计划参数*/
      "doorStatusHolidayGroup": "true,false",
      /*ro, opt, string, 门状态假日组参数*/
      "cardReaderHolidayGroup": "true,false",
      /*ro, opt, string, 读卡器验证方式假日组参数*/
      "userRightHolidayGroup": "true,false",
      /*ro, opt, string, 人员权限假日组参数*/
      "doorStatusTemplate": "true,false",
      /*ro, opt, string, 门状态计划模板参数*/
      "cardReaderTemplate": "true,false",
      /*ro, opt, string, 读卡器验证方式计划模板参数*/
      "userRightTemplate": "true,false",
      /*ro, opt, string, 人员权限计划模板参数*/
    }
  }
}
```

69.46.17 配置门状态假日组参数

Request URL

PUT /ISAPI/AccessControl/DoorStatusHolidayGroupCfg/<holidayGroupID>?format=json

查询参数

参数名称	参数类型	描述
holidayGroupID	string	假日组编号，从1开始，设备支持的最大值从能力集中获取

请求报文

```
{
  "DoorStatusHolidayGroupCfg": {
    /*opt, object, 门状态假日组参数*/
    "enable": true,
    /*req, bool, 是否启用, desc:true-启用*/
    "groupName": "test",
    /*req, string, 假日组名称*/
    "holidayPlanNo": "1,3,5"
    /*opt, string, 假日组计划编号*/
  }
}
```

响应报文

```
{
  "statusCode": 1,
  /*ro, opt, int, 状态码, desc:无法用1表示时（1表示成功且无特殊状态）必须返回*/
  "statusString": "OK",
  /*ro, opt, string, 状态描述, range:[1,64], desc:无法用OK表示时（OK表示成功且无特殊状态）必须返回*/
  "subStatusCode": "ok",
  /*ro, opt, string, 子状态码, range:[1,64], desc:无法用ok表示（ok表示成功且无特殊状态）必须返回*/
  "errorCode": 1,
  /*ro, opt, int, 错误码, desc:当statusCode不为1时，与subStatusCode对应*/
  "errorMsg": "ok"
  /*ro, opt, string, 错误信息, desc:当statusCode不为1时，必须返回,解释信息在协议约束中，允许设备在后续的版本迭代中，进行优化丰富提升（不限制死）*/
}
```

69.47 凭证识别单元管理

69.47.1 NFC功能启用配置能力

Request URL

GET /ISAPI/AccessControl/Configuration/NFCCfg/capabilities?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "NFCCfgCap": {
    /*ro, opt, object, NFC功能启用配置*/
    "enable": "true,false"
    /*ro, req, string, 是否启用NFC功能, desc:true-启用, false-不启用（默认）*/
  }
}
```

69.47.2 获取卡号认证规则参数能力

Request URL

GET /ISAPI/AccessControl/CardVerificationRule/capabilities?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "CardVerificationRuleCap": {
    /*ro, req, object, 卡号认证模式配置能力节点*/
    "cardNoLenMode": {
      /*ro, opt, object, 卡号比对长度模式, desc:切换后设备需要检验卡号兼容性情况*/
      "@opt": ["full", "4Bytes", "3Bytes", "wiegand27", "wiegand35", "Corporate1000_35", "Corporate1000_48", "H10302_37", "H10304_37",
        "wiegand_26CSN", "H103130_32CSN", "wiegand_56CSN", "wiegand_58"]
      /*ro, opt, array, 可选项, subType:string*/
    },
    "CardVerificationRuleRes": {
      /*ro, opt, object, 卡号认证模式配置*/
      "checkStatus": {
        /*ro, opt, object, 配置结果状态, desc:continue-需要稍后查询切换结果 ok-切换成功 duplicate-存在重复数据,切换失败*/
        "@opt": ["continue", "ok", "duplicate"]
        /*ro, opt, array, 可选项, subType:string*/
      },
      "progress": {
        /*ro, opt, object, 切换进度百分比, desc:0-100 100表示完成*/
        "@min": 0,
        /*ro, opt, int, 最小值*/
        "@max": 0
        /*ro, opt, int, 最大值*/
      }
    },
    "reverseCardNoEnabled": {
      /*ro, opt, object, 卡号翻转使能*/
      "@opt": [true, false]
      /*ro, opt, array, 枚举, subType:bool, range:[1,2]*/
    }
  }
}
```

69.47.3 配置韦根参数

Request URL

PUT /ISAPI/AccessControl/WiegandCfg/wiegandNo/<wiegandID>

查询参数

参数名称	参数类型	描述
wiegandID	string	

请求报文

```
<?xml version="1.0" encoding="UTF-8"?>

<WiegandCfg xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--wo, req, object, 韦根参数, attr:version{req, string, 协议版本}-->
  <communicateDirection>
    <!--wo, req, enum, 通信方向, subType:string, [receive#接收,send#发送]-->receive
  </communicateDirection>
  <wiegandMode>
    <!--wo, opt, enum, 输出方向下的韦根模式, subType:string, [wiegand26#韦根26,wiegand34#韦根34,wiegand27#韦根27,wiegand35#韦根35,Corporate1000_35#韦根Corporate 1000 35bits,Corporate1000_48#韦根Corporate 1000 48bits,H10302_37#韦根H10302 37bits,H10304_37#韦根H10304 37bits,wiegand_26CSN#韦根26 CSN,H103130_32CSN#韦根H103130 32bits CSN,wiegand_56CSN#韦根56 CSN,wiegand_58#韦根58], dep:or,{$.WiegandCfg.communicateDirection,eq,send}-->wiegand26
  </wiegandMode>
  <inputWiegandMode>
    <!--wo, opt, enum, 输入方向下的韦根模式, subType:string, [wiegand26#韦根26,wiegand34#韦根34,wiegand27#韦根27,wiegand35#韦根35,Corporate1000_35#韦根Corporate 1000 35bits,Corporate1000_48#韦根Corporate 1000 48bits,H10302_37#韦根H10302 37bits,H10304_37#韦根H10304 37bits,wiegand_26CSN#韦根26 CSN,H103130_32CSN#韦根H103130 32bits CSN,wiegand_56CSN#韦根56 CSN,wiegand_58#韦根58], dep:or,{$.WiegandCfg.communicateDirection,eq,receive}-->wiegand26
  </inputWiegandMode>
  <signalInterval>
    <!--wo, opt, int, 韦根信号发送间隔, range:[1,20], desc:单位 ms-->1
  </signalInterval>
  <enable>
    <!--wo, opt, bool, 使能开关-->true
  </enable>
  <pulseDuration>
    <!--wo, opt, int, 脉冲宽度, range:[1,10], desc:单位 us-->1
  </pulseDuration>
  <facilityCodeEnabled>
    <!--opt, bool, facilityCode使能-->true
  </facilityCodeEnabled>
  <facilityCode>
    <!--opt, int, facilityCode, range:[0,65535], dep:and,{$.WiegandCfg.facilityCodeEnabled,eq,true}, desc:当wiegandMode为输出时, facilityCode可配置, 且生效 (用于设备输出卡号+facilityCode至三方设备进行校验) 当wiegandMode为输入时, facilityCode可配置, 但不生效 (即设备接收韦根数据时, 不对输入的facilityCode进行校验) -->1
  </facilityCode>
  <dataType>
    <!--opt, enum, 数据类型, subType:string, [employeeNo#工号,cardNo#卡号], dep:or,{$.WiegandCfg.wiegandMode,eq,send}, desc:主机韦根口输出工号或卡号至三方主机-->employeeNo
  </dataType>
</WiegandCfg>
```

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 响应消息, attr:version{ro, req, string, 版本}-->
  <requestURL>
    <!--ro, req, string, 请求的URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, 状态码, subType:int, [0#OK,1#OK,2#Device Busy,3#Device Error,4#Invalid Operation,5#Invalid XML Format,6#Invalid XML Content,7#Reboot Required]-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, 状态信息, subType:string, [OK#成功,Device Busy#设备忙,Device Error#设备异常,Invalid Operation#无效的操作,Invalid XML Format#无效的XML格式,Invalid XML Content#无效的XML内容,Reboot#设备重启]-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, 详细错误码英文描述, desc:详细错误码的英文描述-->OK
  </subStatusCode>
</ResponseStatus>
```

69.47.4 获取韦根参数

Request URL

GET /ISAPI/AccessControl/WiegandCfg/wiegandNo/<wiegandID>

查询参数

参数名称	参数类型	描述
wiegandID	string	

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<WiegandCfg xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 韦根参数, attr:version{req, string, 协议版本}-->
  <communicateDirection>
    <!--ro, req, enum, 通信方向, subType:string, [receive#接收,send#发送]-->receive
  </communicateDirection>
  <wiegandMode>
    <!--ro, opt, enum, 输出方向下的韦根模式, subType:string, [wiegand26#韦根26,wiegand34#韦根34,wiegand27#韦根27,wiegand35#韦根35,Corporate1000_35#韦根Corporate 1000 35bits,Corporate1000_48#韦根Corporate 1000 48bits,H10302_37#韦根H10302 37bits,H10304_37#韦根H10304 37bits,wiegand_26CSN#韦根26 CSN,H103130_32CSN#韦根H103130 32bits CSN,wiegand_56CSN#韦根56 CSN,wiegand_58#韦根58], dep:or,{$.WiegandCfg.communicateDirection,eq,send}-->wiegand26
  </wiegandMode>
  <inputWiegandMode>
    <!--ro, opt, enum, 输入方向下的韦根模式, subType:string, [wiegand26#韦根26,wiegand34#韦根34,wiegand27#韦根27,wiegand35#韦根35,Corporate1000_35#韦根Corporate 1000 35bits,Corporate1000_48#韦根Corporate 1000 48bits,H10302_37#韦根H10302 37bits,H10304_37#韦根H10304 37bits,wiegand_26CSN#韦根26 CSN,H103130_32CSN#韦根H103130 32bits CSN,wiegand_56CSN#韦根56 CSN,wiegand_58#韦根58], dep:or,{$.WiegandCfg.communicateDirection,eq,receive}-->wiegand26
  </inputWiegandMode>
  <signalInterval>
    <!--ro, opt, int, 韦根信号发送间隔, range:[1,20], desc:单位 ms-->1
  </signalInterval>
  <enable>
    <!--ro, opt, bool, 使能开关-->true
  </enable>
  <pulseDuration>
    <!--ro, opt, int, 脉冲宽度, range:[1,10], desc:单位 us-->1
  </pulseDuration>
  <facilityCodeEnabled>
    <!--ro, opt, bool, facilityCode使能-->true
  </facilityCodeEnabled>
  <facilityCode>
    <!--ro, opt, int, facilityCode, range:[0,65535], dep:and,{$.WiegandCfg.facilityCodeEnabled,eq,true}, desc:当wiegandMode为输出时, facilityCode可配置, 且生效 (用于设备输出卡号+facilityCode至三方设备进行校验) 当wiegandMode为输入时, facilityCode可配置, 但不生效 (即设备接收韦根数据时, 不对输入的facilityCode进行校验)-->1
  </facilityCode>
  <dataType>
    <!--ro, opt, enum, 数据类型, subType:string, [employeeNo#工号,cardNo#卡号], dep:or,{$.WiegandCfg.wiegandMode,eq,send}, desc:主机韦根口输出工号或卡号至三方主机-->employeeNo
  </dataType>
</WiegandCfg>
```

69.47.5 获取识别单元的虹膜识别参数

Request URL

GET /ISAPI/AccessControl/IrisCfg/<readerID>?format=json

查询参数

参数名称	参数类型	描述
readerID	string	--

请求报文

无

响应报文

```
{
  "IrisCfg": {
    /*ro, opt, object, 虹膜参数配置*/
    "irisAntiSpoofingEnabled": true,
    /*ro, opt, bool, 虹膜活体（抗欺骗）检测是否开启, desc:活体检测配置开时，需要对虹膜进行活体检测，只有活体才能通过，配置关时，照片可以通过*/
    "irisAntiSpoofingLevel": "high",
    /*ro, opt, enum, 虹膜活体（抗欺骗）检测等级, subType:string, [high#高,medium#中,low#低]*/
    "irisMatchThresholdN": 1,
    /*ro, opt, int, 虹膜1: N比对阈值, range:[0,100], desc:可以配置模板比对的阈值，阈值限制为100以内整数，只有相似度不低于阈值时，认为比对通过*/
    "irisRecognizeInterval": 1,
    /*ro, opt, int, 虹膜连续识别间隔, range:[0,255]*/
    "irisVerifyMode": "monocular",
    /*ro, opt, enum, 虹膜认证模式, subType:string, [monocular#单目,binocular#双目,auto#自动模式],
    desc:双目时必须两个虹膜比对均通过才认为认证通过;单目时若只提取到一个虹膜模板,则直接使用当前模板,若同时提取到两个模板,则选择相似度更高
    双目模式下如人员仅录入单眼,则认证一定失败;单目模式下如人员录入双眼,则任意眼比对成功即认证通过
    自动模式下先使用irisMatchThresholdN阈值比对单眼, 若单眼未通过才会比对双眼, 此时使用binocularIrisMatchThresholdN双眼阈值，通常双眼阈值会低于
    irisMatchThresholdN阈值*/
    "binocularIrisMatchThresholdN": 1,
    /*ro, opt, int, 自动模式下双眼虹膜1: N比对阈值, range:[0,100], dep:and,{$.IrisCfg.irisVerifyMode,eq,auto}, desc:自动模式下当双眼阈值都超过该值时认为
    认证通过*/
    "irisRecognizeEnabledInSimilarFace": true,
    /*ro, opt, bool, 相似人脸情况下虹膜识别开关, desc:人脸识别后，设备发现库内有两个符合条件的，然后提示切虹膜（双胞胎场景）*/
    "monocularIrisMatchThreshold1": 1,
    /*ro, opt, int, 虹膜单眼1: 1比对阈值, range:[0,100]*/
    "binocularIrisMatchThreshold1": 1,
    /*ro, opt, int, 虹膜双眼1: 1比对阈值, range:[0,100]*/
    "irisLiveDetAntiAttackEnabled": true,
    /*ro, opt, bool, 虹膜活体检测防攻击使能*/
    "irisRecognizeTimeOut": 1
    /*ro, opt, int, 虹膜识别超时时间, range:[1,20]*/
  }
}
```

69.47.6 配置识别单元的虹膜识别参数

Request URL

PUT /ISAPI/AccessControl/IrisCfg/<readerID>?format=json

查询参数

参数名称	参数类型	描述
readerID	string	

请求报文

```
{
  "IrisCfg": {
    /*opt, object, 虹膜参数配置*/
    "irisAntiSpoofingEnabled": true,
    /*opt, bool, 虹膜活体（抗欺骗）检测是否开启, desc:活体检测配置开时，需要对虹膜进行活体检测，只有活体才能通过，配置关时，照片可以通过*/
    "irisAntiSpoofingLevel": "high",
    /*opt, enum, 虹膜活体（抗欺骗）检测等级, subType:string, [high#高,medium#中,low#低]*/
    "irisMatchThresholdN": 1,
    /*opt, int, 虹膜1: N比对阈值, range:[0,100], desc:可以配置模板比对的阈值，阈值限制为100以内整数，只有相似度不低于阈值时，认为比对通过*/
    "irisRecognizeInterval": 1,
    /*opt, int, 虹膜连续识别间隔, range:[0,255]*/
    "irisVerifyMode": "monocular",
    /*opt, enum, 虹膜认证模式, subType:string, [monocular#单目,binocular#双目,auto#自动模式],
    desc:双目时必须两个虹膜比对均通过才认为认证通过;单目时若只提取到一个虹膜模板,则直接使用当前模板,若同时提取到两个模板,则选择相似度更高
    双目模式下如人员仅录入单眼,则认证一定失败;单目模式下如人员录入双眼,则任意眼比对成功即认证通过
    自动模式下先使用irisMatchThresholdN阈值比对单眼, 若单眼未通过才会比对双眼, 此时使用binocularIrisMatchThresholdN双眼阈值，通常双眼阈值会低于
    irisMatchThresholdN阈值*/
    "binocularIrisMatchThresholdN": 1,
    /*opt, int, 自动模式下双眼虹膜1: N比对阈值, range:[0,100], dep:and,{$.IrisCfg.irisVerifyMode,eq,auto}, desc:自动模式下当双眼阈值都超过该值时认为认证
    通过*/
    "irisRecognizeEnabledInSimilarFace": true,
    /*opt, bool, 相似人脸情况下虹膜识别开关, desc:人脸识别后，设备发现库内有两个符合条件的，然后提示切虹膜（双胞胎场景）*/
    "monocularIrisMatchThreshold1": 1,
    /*ro, opt, int, 虹膜单眼1: 1比对阈值, range:[0,100]*/
    "binocularIrisMatchThreshold1": 1,
    /*ro, opt, int, 虹膜双眼1: 1比对阈值, range:[0,100]*/
    "irisLiveDetAntiAttackEnabled": true,
    /*ro, opt, bool, 虹膜活体检测防攻击使能*/
    "irisRecognizeTimeOut": 1
    /*ro, opt, int, 虹膜识别超时时间, range:[1,20]*/
  }
}
```

响应报文

```
{
  "statusCode": 1,
  /*ro, opt, int, 状态码, desc:无法用1表示时（1表示成功且无特殊状态）必须返回*/
  "statusString": "OK",
  /*ro, opt, string, 状态描述, range:[1,64], desc:无法用OK表示时（OK表示成功且无特殊状态）必须返回*/
  "subStatusCode": "ok",
  /*ro, opt, string, 子状态码, range:[1,64], desc:无法用ok表示（ok表示成功且无特殊状态）必须返回*/
  "errorCode": 1,
  /*ro, opt, int, 错误码, desc:当statusCode不为1时，与subStatusCode对应*/
  "errorMsg": "ok"
  /*ro, opt, string, 错误信息, desc:当statusCode不为1时，必须返回,解释信息在协议约束中，允许设备在后续的版本迭代中，进行优化丰富提升（不限制死）*/
}
```

69.47.7 获取识别单元的虹膜识别参数能力

Request URL

GET /ISAPI/AccessControl/IrisCfg/capabilities?format=json

查询参数

无

请求报文

无

响应报文

有效期：2024-08-15

湖南云舟电力科技有限公司

```

{
  "IrisCfgCap": {
    /*ro, opt, object, 虹膜参数能力*/
    "cardReaderNo": {
      /*ro, opt, object, 识别单元编号*/
      "@min": 1,
      /*ro, req, int, 最小值*/
      "@max": 8
      /*ro, req, int, 最大值*/
    },
    "irisAntiSpoofingEnabled": {
      /*ro, opt, object, 虹膜活体（抗欺骗）检测是否开启*/
      "@opt": [true, false]
      /*ro, req, array, 取值范围, subType:bool*/
    },
    "irisAntiSpoofingLevel": {
      /*ro, opt, object, 虹膜活体（抗欺骗）检测等级*/
      "@opt": ["high", "medium", "low"]
      /*ro, req, array, 取值范围, subType:string*/
    },
    "irisMatchThresholdN": {
      /*ro, opt, object, 虹膜1: N比对阈值*/
      "@min": 0,
      /*ro, req, int, 取值范围*/
      "@max": 100
      /*ro, req, int, 取值范围*/
    },
    "irisRecognizeInterval": {
      /*ro, opt, object, 虹膜连续识别间隔*/
      "@min": 0,
      /*ro, req, int, 取值范围*/
      "@max": 255
      /*ro, req, int, 取值范围*/
    },
    "irisVerifyMode": {
      /*ro, opt, object, 虹膜认证模式*/
      "@opt": ["monocular", "binocular", "auto"]
      /*ro, req, array, 取值范围, subType:string*/
    },
    "binocularIrisMatchThresholdN": {
      /*ro, opt, object, 虹膜1: N比对阈值*/
      "@min": 0,
      /*ro, req, int, 取值范围*/
      "@max": 100
      /*ro, req, int, 取值范围*/
    },
    "irisRecognizeEnabledInSimilarFace": {
      /*ro, opt, object, 相似人脸情况下虹膜识别开关*/
      "@opt": [true, false]
      /*ro, req, array, 取值范围, subType:bool*/
    },
    "monocularIrisMatchThreshold1": {
      /*ro, opt, object, 虹膜单眼1: 1比对阈值*/
      "@min": 0,
      /*ro, req, int, 取值范围*/
      "@max": 100
      /*ro, req, int, 取值范围*/
    },
    "binocularIrisMatchThreshold1": {
      /*ro, opt, object, 虹膜双眼1: 1比对阈值*/
      "@min": 0,
      /*ro, req, int, 取值范围*/
      "@max": 100
      /*ro, req, int, 取值范围*/
    },
    "irisLiveDetAntiAttackEnabled": {
      /*ro, opt, object, 虹膜活体检测防攻击使能*/
      "@opt": [true, false]
      /*ro, req, array, 取值范围, subType:bool*/
    },
    "irisRecognizeTimeOut": {
      /*ro, opt, object, 虹膜识别超时时间*/
      "@min": 1,
      /*ro, req, int, 取值范围*/
      "@max": 20
      /*ro, req, int, 取值范围*/
    }
  }
}

```

69.47.8 获取识别单元参数

Request URL

GET /ISAPI/AccessControl/CardReaderCfg/<cardReaderID>?format=json

查询参数

参数名称	参数类型	描述
cardReaderID	string	

请求报文

无

响应报文

```
{
  "CardReaderCfg": {
    /*ro, req, object, 读卡器信息*/
    "enable": true,
    /*ro, req, bool, 使能, desc:true-使能, false-不使能*/
    "okLedPolarity": "cathode",
    /*ro, opt, enum, OK LED极性, subType:string, [cathode#阴极,anode#阳极], desc:cathode-阴极, anode-阳极*/
    "errorLedPolarity": "cathode",
    /*ro, opt, enum, Error LED极性, subType:string, [cathode#阴极,anode#阳极], desc:cathode-阴极, anode-阳极*/
    "buzzerPolarity": "cathode",
    /*ro, opt, enum, 蜂鸣器极性, subType:string, [cathode#阴极,anode#阳极], desc:cathode-阴极, anode-阳极*/
    "swipeInterval": 1,
    /*ro, opt, int, 重复认证间隔时间, unit:s, unitType:时间, desc:指纹、卡、人脸等认证都生效*/
    "pressTimeout": 1,
    /*ro, opt, int, 按键超时时间, unit:s, unitType:时间*/
    "enableFailAlarm": true,
    /*ro, opt, bool, 是否启用认证失败超次报警*/
    "maxReadCardFailNum": 1,
    /*ro, opt, int, 最大认证失败次数*/
    "enableTamperCheck": true,
    /*ro, opt, bool, 是否支持防拆检测*/
    "offlineCheckTime": 1,
    /*ro, opt, int, 掉线检测时间, unit:s, unitType:时间*/
    "fingerPrintCheckLevel": 1,
    /*ro, opt, enum, 指纹识别等级, subType:int, [1#1/10误认率,2#1/100误认率,3#1/1000误认率,4#1/10000误认率,5#1/100000误认率,6#1/1000000误认率,7#1/10000000误认率,8#1/100000000误认率,9#3/100误认率,10#3/1000误认率,11#3/10000误认率,12#3/100000误认率,13#3/1000000误认率,14#3/10000000误认率,15#3/100000000误认率,16#Automatic Normal,17#Automatic Secure,18#Automatic More Secure]*/
    "useLocalController": true,
    /*ro, opt, bool, 是否连接在就地控制器上*/
    "localControllerID": 1,
    /*ro, opt, int, 就地控制器序号, range:[0,64], dep:and,{$.CardReaderCfg.localControllerID,eq,true}, desc:useLocalController为true时有效,0代表未注册*/
    "localControllerReaderID": 1,
    /*ro, opt, int, 就地控制器的读卡器ID, dep:and,{$.CardReaderCfg.localControllerID,eq,true}, desc:useLocalController为true时有效,0代表未注册*/
    "cardReaderChannel": 1,
    /*ro, opt, enum, 读卡器通信通道号, subType:int, [0#韦根或离线,1#RS485A,2#RS485B], dep:and,{$.CardReaderCfg.localControllerID,eq,true}, desc:useLocalController为true时有效*/
    "fingerPrintImageQuality": 1,
    /*ro, opt, enum, 指纹图像质量, subType:int, [1#低质量(V1),2#中等质量(V1),3#高质量(V1),4#最高质量(V1),5#低质量(V2),6#中等质量(V2),7#高质量(V2),8#最高质量(V2)]*/
    "fingerPrintContrastTimeOut": 1,
    /*ro, opt, enum, 指纹对比超时时间, subType:int, [1#1s,2#2s,3#3s,4#4s,5#5s,6#6s,7#7s,8#8s,9#9s,10#10s,11#12s,12#12s,13#13s,14#14s,15#15s,16#16s,17#17s,18#18s,19#19s,20#20s,255#无限制]*/
    "fingerPrintRecognizeInterval": 1,
    /*ro, opt, enum, 指纹连续识别间隔, subType:int, [1#1s,2#2s,3#3s,4#4s,5#5s,6#6s,7#7s,8#8s,9#9s,10#10s,255#无延迟]*/
    "fingerPrintMatchFastMode": 1,
    /*ro, opt, enum, 指纹匹配快速模式, subType:int, [1#快速模式1,2#快速模式2,3#快速模式3,4#快速模式4,5#快速模式5,255#自动]*/
    "fingerPrintModuleSensitive": 1,
    /*ro, opt, enum, 指纹模组灵敏度, subType:int, [1#灵敏度级别1,2#灵敏度级别2,3#灵敏度级别3,4#灵敏度级别4,5#灵敏度级别5,6#灵敏度级别6,7#灵敏度级别7,8#灵敏度级别8]*/
    "fingerPrintModuleLightCondition": "outdoor",
    /*ro, opt, enum, 指纹模组光线条件, subType:string, [outdoor#室外,indoor#室内], desc:outdoor-室外, indoor-室内*/
    "faceMatchThresholdN": 1,
    /*ro, opt, int, 人脸1: N比对阈值, range:[0,100]*/
    "faceQuality": 1,
    /*ro, opt, int, 人脸质量, range:[0,100]*/
    "faceRecognizeTimeOut": 1,
    /*ro, opt, enum, 人脸识别超时时间, subType:int, [1#1s,2#2s,3#3s,4#4s,5#5s,6#6s,7#7s,8#8s,9#9s,10#10s,11#12s,12#12s,13#13s,14#14s,15#15s,16#16s,17#17s,18#18s,19#19s,20#20s,255#无限制]*/
    "faceRecognizeInterval": 1,
    /*ro, opt, enum, 人脸连续识别间隔, subType:int, [1#1s,2#2s,3#3s,4#4s,5#5s,6#6s,7#7s,8#8s,9#9s,10#10s,255#无延迟]*/
    "cardReaderFunction": ["fingerPrint", "face", "card"],
    /*ro, opt, enumarray, 读卡器种类, subType:string, [fingerPrint#指纹,face#人脸,fingerVein#静脉,card#卡], desc:["fingerPrint", "face"]代表读卡器同时支持指纹和人脸*/
    "cardReaderDescription": "Wiegand\u000485Offline",
    /*ro, opt, string, 读卡器描述, desc:当读卡器为韦根读卡器或读卡器不在线时, 该字符串为: Wiegand485Offline*/
    "faceImageSensitometry": 1,
    /*ro, opt, int, 人脸图像曝光度, range:[0,65535]*/
    "livingBodyDetect": true,
    /*ro, opt, bool, 真人检测使能*/
    "faceMatchThresholdI": 1,
    /*ro, opt, int, 人脸1: I比对阈值, range:[0,100]*/
    "buzzerTime": 1,
    /*ro, opt, int, 蜂鸣时间, range:[0,59999], unit:s, unitType:时间, desc:0-代表长鸣*/
    "faceMatch1SecurityLevel": 1,
    /*ro, opt, enum, 人脸1: 1识别安全等级, subType:int, [1#一般,2#较强,3#极强]*/
    "faceMatchNSecurityLevel": 1,
    /*ro, opt, enum, 人脸1: N识别安全等级, subType:int, [1#一般,2#较强,3#极强]*/
    "envirMode": "other",
    /*ro, opt, enum, 人脸识别环境模式, subType:string, [other#其他,indoor#室内]*/
    "livingBodyDetect": true
  }
}
```

“test”,
指纹算法库版本, range:[1,32]*/
: “test”,
读卡器版本, range:[1,32]*/
lo”: true,
号翻转使能*/
vals”: 0,
认证间隔时间, desc: (每个人单独计算间隔,区别于swipeInterval), 单位: 秒*/
holdN”: 1,
人脸1VN比对阈值, range:[0,100]*/
hold1”: 1,
人脸1V1比对阈值, range:[0,100]*/
”: “low”,
脸移动侦测级别, subType:string, [low#低,medium#中,height#高]*/
1”,
示模式, subType:string, [concise#简洁模式,normal#正常模式(默认),advertising#广告模式,meeting#会议模式,selfDefine#自定义
正常模式,简洁模式是指仅显示认证结果,不显示工号、姓名等信息以及预览信息;普通模式即正常明眸认证界面;广告模式即上下分屏,
义签到界面;自定义模式即用户自定义UI的主界面、认证成功、认证失败界面,使用信息发布协议下发自定义布局*/
true,
否启用息屏*/
1
屏时间, step:1, unit:s, unitType:时间*/

单元参数

ctrl/CardReaderCfg/<cardReaderID>?format=json

类型	描述

Request URL

查询参数

请求报文

```
认率,8#1/100000000误认率,9#3/100误认率,10#3/1000误认率,11#3/10000误认率,12#3/100000误认率,13#3/1000000误认率,14#3/10000000误认率,15#3/100000000误认率,16#Automatic Normal,17#Automatic Secure,18#Automatic More Secure]*/
"useLocalController": true,
/*opt, bool, 是否连接在就地控制器上*/
"localControllerID": 1,
/*opt, int, 就地控制器序号, range:[0,64], dep:and,{$.CardReaderCfg.localControllerID,eq,true}, desc:useLocalController为true时有效,0代表未注册*/
"localControllerReaderID": 1,
/*opt, int, 就地控制器的读卡器ID, dep:and,{$.CardReaderCfg.localControllerID,eq,true}, desc:useLocalController为true时有效,0代表未注册*/
"cardReaderChannel": 1,
/*opt, enum, 读卡器通信通道号, subType:int, [0#韦根或离线,1#RS485A,2#RS485B], dep:and,{$.CardReaderCfg.localControllerID,eq,true},
desc:useLocalController为true时有效*/
"fingerPrintImageQuality": 1,
/*opt, enum, 指纹图像质量, subType:int, [1#低质量(V1),2#中等质量(V1),3#高质量(V1),4#最高质量(V1),5#低质量(V2),6#中等质量(V2),7#高质量(V2),8#最高质量(V2)]*/
"fingerPrintContrastTimeOut": 1,
/*opt, enum, 指纹对比超时时间, subType:int,
[1#1s,2#2s,3#3s,4#4s,5#5s,6#6s,7#7s,8#8s,9#9s,10#10s,11#12s,12#12s,13#13s,14#14s,15#15s,16#16s,17#17s,18#18s,19#19s,20#20s,255#无限制]*/
"fingerPrintRecognizeInterval": 1,
/*opt, enum, 指纹连续识别间隔, subType:int, [1#1s,2#2s,3#3s,4#4s,5#5s,6#6s,7#7s,8#8s,9#9s,10#10s,255#无延迟]*/
"fingerPrintMatchFastMode": 1,
/*opt, enum, 指纹匹配快速模式, subType:int, [1#快速模式1,2#快速模式2,3#快速模式3,4#快速模式4,5#快速模式5,255#自动]*/
"fingerPrintModuleSensitive": 1,
/*opt, enum, 指纹模组灵敏度, subType:int, [1#灵敏度级别1,2#灵敏度级别2,3#灵敏度级别3,4#灵敏度级别4,5#灵敏度级别5,6#灵敏度级别6,7#灵敏度级别7,8#灵敏度级别8]*/
"fingerPrintModuleLightCondition": "outdoor",
/*opt, enum, 指纹模组光线条件, subType:string, [outdoor#室外,indoor#室内], desc:outdoor-室外, indoor-室内*/
"faceMatchThresholdN": 1,
/*opt, int, 人脸1: N比对阈值, range:[0,100], desc:普通模式和深度模式下的阈值均采用该字段, 设备存储两种模式下的参数*/
"faceQuality": 1,
/*opt, int, 人脸质量, range:[0,100]*/
"faceRecognizeTimeOut": 1,
/*opt, enum, 人脸识别超时时间, subType:int,
[1#1s,2#2s,3#3s,4#4s,5#5s,6#6s,7#7s,8#8s,9#9s,10#10s,11#12s,12#12s,13#13s,14#14s,15#15s,16#16s,17#17s,18#18s,19#19s,20#20s,255#无限制]*/
"faceRecognizeInterval": 1,
/*opt, enum, 人脸连续识别间隔, subType:int, [1#1s,2#2s,3#3s,4#4s,5#5s,6#6s,7#7s,8#8s,9#9s,10#10s,255#无延迟]*/
"cardReaderFunction": ["fingerPrint", "face", "fingerVein", "iris", "card"],
/*opt, enumarray, 读卡器种类, subType:string, [fingerPrint#指纹,face#人脸,fingerVein#静脉,iris#虹膜,card#卡], desc:["fingerPrint", "face"]代表读卡器同时支持指纹和人脸*/
"cardReaderDescription": "WiegandU000485Offline",
/*opt, string, 读卡器描述, desc:当读卡器为韦根读卡器或读卡器不在线时, 该字符串为: Wiegand485Offline*/
"faceImageSensitometry": 1,
/*opt, int, 人脸图像曝光度, range:[0,65535]*/
"livingBodyDetect": true,
/*opt, bool, 真人检测使能*/
"faceMatchThresholdI": 1,
/*opt, int, 人脸1: I比对阈值, range:[0,100], desc:普通模式和深度模式下的阈值均采用该字段, 设备存储两种模式下的参数*/
"buzzerTime": 1,
/*opt, int, 蜂鸣时间, range:[0,59999], unit:s, unitType:时间, desc:0-代表长鸣*/
"faceMatchISecurityLevel": 1,
/*opt, enum, 人脸1: I识别安全等级, subType:int, [1#一般,2#较强,3#极强]*/
"faceMatchNSecurityLevel": 1,
/*opt, enum, 人脸1: N识别安全等级, subType:int, [1#一般,2#较强,3#极强]*/
"envirMode": "indoor",
/*opt, enum, 人脸识别环境模式, subType:string, [other#其他,indoor#室内]*/
"liveDetLevelSet": "low",
/*opt, enum, 活体检测阈值等级设置, subType:string, [low#低,middle#中,high#高], desc:low-低, middle-中, high-高*/
"liveDetAntiAttackCntLimit": 1,
/*opt, int, 活体检测防攻击次数, range:[1,255], desc:客户端、设备统一次数限制, 根据能力限制*/
"enableLiveDetAntiAttack": true,
/*opt, bool, 活体检测防攻击使能*/
"supportDelFPByID": true,
/*opt, bool, 读卡器是否支持按手指ID删除指纹, desc:false-不支持, true-支持*/
"fingerPrintCapacity": 1,
/*opt, int, 指纹容量*/
"fingerPrintNum": 1,
/*opt, int, 已存在指纹数量*/
"defaultVerifyMode": "cardAndPw",
/*opt, enum, 读卡器默认验证方式, subType:string, [cardAndPw#刷卡+密码,card#刷卡,cardOrPw#刷卡或密码,fp#指纹,fpAndPw#指纹+密码,fpOrCard#指纹或刷卡,fpAndCard#指纹+刷卡,fpAndCardAndPw#指纹+刷卡+密码,faceOrFpOrCardOrPw#人脸或指纹或刷卡或密码,faceAndFp#人脸+指纹,faceAndPw#人脸+密码,faceAndCard#人脸+刷卡,faceAndPwAndFp#人脸+密码+指纹,employeeNoAndFp#工号+指纹,employeeNoAndFpAndPw#工号+指纹+密码,faceAndFpAndCard#人脸+指纹+刷卡,faceAndPwAndFp#人脸+密码+指纹,employeeNoAndFace#工号+人脸,faceOrFaceAndCard#人脸或人脸+刷卡,fpOrFace#指纹或人脸,cardOrFaceOrPw#刷卡或人脸或密码,iris#虹膜,faceOrFpOrCardOrPwOrIris#人脸或指纹或刷卡或密码或虹膜,faceOrCardOrPwOrIris#人脸或刷卡或密码或虹膜,cardOrFpOrFaceOrIris#刷卡或指纹或人脸或虹膜,fpOrFaceOrIrisOrPw#指纹或人脸或虹膜或密码,cardOrFpOrIrisOrPw#刷卡或指纹或虹膜或密码,cardOrIrisOrPw#刷卡或虹膜或密码,cardAndIris#刷卡+虹膜,fpAndIris#指纹+虹膜,faceAndIris#人脸+虹膜,irisAndPw#虹膜+密码,cardAndIrisAndPw#刷卡+虹膜+密码,faceAndIrisAndPw#人脸+虹膜+密码,cardAndFaceAndIris#刷卡+人脸+虹膜,Pw#密码], desc:出厂默认*/
"faceRecognizeEnable": 1,
/*opt, enum, 人脸识别, subType:int, [1#开启,2#关闭,3#多人脸考勤]*/
"FPAlgorithmVersion": "test",
/*opt, string, 指纹算法库版本, range:[1,32]*/
"cardReaderVersion": "test",
/*opt, string, 读卡器版本, range:[1,32]*/
"enableReverseCardNo": true,
/*opt, bool, 卡号翻转使能*/
"independSwipeIntervals": 0,
/*opt, int, 人员认证间隔时间, desc:(每个人单独计算间隔,区别于swipeInterval), 单位: 秒*/
"maskFaceMatchThresholdN": 1,
/*opt, int, 口罩人脸1VN比对阈值, range:[0,100], desc:普通模式和深度模式下的阈值均采用该字段, 设备存储两种模式下的参数*/
"maskFaceMatchThresholdI": 1,
/*opt, int, 口罩人脸1VI比对阈值, range:[0,100], desc:普通模式和深度模式下的阈值均采用该字段, 设备存储两种模式下的参数*/
"faceMotionDetLevel": "low",
/*opt, enum, 人脸移动侦测级别, subType:string, [low#低,medium#中,height#高]*/
"showMode": "normal",
/*opt, enum, 显示模式, subType:string, [concise#简洁模式,normal#正常模式(默认),advertising#广告模式,meeting#会议模式,selfDefine#自定义模式], desc:无显示模式字段, 默认在正常模式, 简洁模式是指仅显示认证结果, 不显示工号、姓名等信息以及预览信息; 普通模式即正常明眸认证界面; 广告模式即上下分屏, 上屏广告, 下
```

屏认证：会议模式即会议签到界面；自定义模式即用户自定义UI的主界面、认证成功、认证失败界面，使用信息发布协议下发自定义布局*/

```
"enableScreenOff": true,
/*opt, bool, 是否启用息屏*/
"screenOffTimeout": 1
/*opt, int, 息屏时间, step:1, unit:s, unitType:时间*/
}
}
```

响应报文

```
{
  "statusCode": 1,
  /*ro, opt, int, 状态码, desc:无法用1表示时（1表示成功且无特殊状态）必须返回*/
  "statusString": "OK",
  /*ro, opt, string, 状态描述, range:[1,64], desc:无法用OK表示时（OK表示成功且无特殊状态）必须返回*/
  "subStatusCode": "ok",
  /*ro, opt, string, 子状态码, range:[1,64], desc:无法用ok表示（ok表示成功且无特殊状态）必须返回*/
  "errorCode": 1,
  /*ro, opt, int, 错误码, desc:当statusCode不为1时，与subStatusCode对应*/
  "errorMsg": "ok"
  /*ro, opt, string, 错误信息, desc:当statusCode不为1时，必须返回,解释信息在协议约束中，允许设备在后续的版本迭代中，进行优化丰富提升（不限制死）*/
}
```

69.47.10 获取识别单元参数能力

Request URL

GET /ISAPI/AccessControl/CardReaderCfg/capabilities?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "CardReaderCfg": {
    /*ro, req, object, 读卡器参数配置*/
    "cardReaderNo": {
      /*ro, opt, object, 读卡器编号, desc:读卡器编号*/
      "@min": 1,
      /*ro, opt, int, 最小值*/
      "@max": 512,
      /*ro, opt, int, 最大值*/
    },
    "enable": "true,false",
    /*ro, opt, string, 使能, desc:使能: true-使能, false-不使能*/
    "okLedPolarity": {
      /*ro, opt, object, OK LED极性*/
      "@opt": "cathode,anode"
      /*ro, req, string, 选项, desc:cathode-阴极, anode-阳极*/
    },
    "errorLedPolarity": {
      /*ro, opt, object, Error LED极性*/
      "@opt": "cathode,anode"
      /*ro, req, string, 选项, desc:cathode-阴极, anode-阳极*/
    },
    "swipeInterval": {
      /*ro, opt, object, 重复认证间隔时间, desc:指纹、卡、人脸等认证都生效, 单位: 秒*/
      "@min": 1,
      /*ro, req, int, 最小值*/
      "@max": 255
      /*ro, req, int, 最大值*/
    },
    "pressTimeout": {
      /*ro, opt, object, 按键超时时间, desc:单位: 秒*/
      "@min": 1,
      /*ro, req, int, 最小值*/
      "@max": 255
      /*ro, req, int, 最大值*/
    },
    "enableFailAlarm": "true,false",
    /*ro, opt, string, 是否启用认证失败超次报警*/
    "maxReadCardFailNum": {
      /*ro, opt, object, 最大认证失败次数*/
      "@min": 1,
      /*ro, req, int, 最小值*/
      "@max": 255
      /*ro, req, int, 最大值*/
    },
    "enableTamperCheck": "true,false",
    /*ro, opt, string, 是否启用非法拆卸检测*/
  }
}
```

```

/*ro, opt, string, 是否支持防攻击检测*/
"offlineCheckTime": {
/*ro, opt, object, 掉线检测时间, desc:单位: 秒*/
"@min": 1,
/*ro, req, int, 最小值*/
"@max": 255
/*ro, req, int, 最大值*/
},
"fingerPrintCheckLevel": {
/*ro, opt, object, 指纹识别等级*/
"@opt": "1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18"
/*ro, req, string, 取值范围*/
},
"faceMatchThresholdN": {
/*ro, opt, object, 人脸识别1N阈值, desc:范围0-100*/
"@min": 0,
/*ro, req, int, 最小值*/
"@max": 100
/*ro, req, int, 最大值*/
},
"faceQuality": {
/*ro, opt, object, 人脸质量, desc:范围0-100*/
"@min": 0,
/*ro, req, int, 最小值*/
"@max": 100
/*ro, req, int, 最大值*/
},
"faceRecognizeTimeOut": {
/*ro, opt, object, 人脸识别超时时间, desc:范围1-20代表: 1s-20s, 能力集中的0代表无限制*/
"@min": 0,
/*ro, req, int, 最小值*/
"@max": 20
/*ro, req, int, 最大值*/
},
"faceRecognizeInterval": {
/*ro, opt, object, 人脸连续识别间隔, desc:范围1-10代表: 1s-10s, 能力集中的0代表无延迟*/
"@min": 0,
/*ro, req, int, 最小值*/
"@max": 10
/*ro, req, int, 最大值*/
},
"cardReaderFunction": {
/*ro, opt, object, 读卡器种类*/
"@opt": "fingerPrint,face,fingerVein,iris,card"
/*ro, req, string, 选项, desc:fingerPrint-指纹, face-人脸, fingerVein-指静脉, iris-虹膜*/
},
"cardReaderDescription": {
/*ro, opt, object, 读卡器描述, desc:当读卡器为韦根读卡器或读卡器不在线时, 该字符串为: wiegand4850ffline*/
"@min": 1,
/*ro, req, int, 最小值*/
"@max": 16
/*ro, req, int, 最大值*/
},
"faceImageSensitometry": {
/*ro, opt, object, 人脸图像曝光度, desc:范围0-65535*/
"@min": 0,
/*ro, req, int, 最小值*/
"@max": 65535
/*ro, req, int, 最大值*/
},
"livingBodyDetect": "true,false",
/*ro, opt, string, 真人检测使能*/
"faceMatchThreshold1": {
/*ro, opt, object, 人脸1V1比对阈值, desc:范围0-100*/
"@min": 0,
/*ro, req, int, 最小值*/
"@max": 100
/*ro, req, int, 最大值*/
},
"faceMatch1SecurityLevel": {
/*ro, opt, object, 人脸识别11安全等级*/
"@opt": "1,2,3"
/*ro, req, string, 选项, desc:1-一般, 2-较强, 3-极强*/
},
"envirMode": {
/*ro, opt, object, 人脸识别环境模式*/
"@opt": "indoor,other"
/*ro, req, string, 选项, desc:indoor-室内, other-其他*/
},
"liveDetLevelSet": {
/*ro, opt, object, 活体检测阈值等级设置*/
"@opt": "low,middle,high"
/*ro, req, string, 选项, desc:low-低, middle-中, high-高*/
},
"liveDetAntiAttackCntLimit": {
/*ro, opt, object, 活体检测防攻击次数, desc:1-255次(客户端、设备统一次数限制, 根据能力级限制)*/
"@min": 1,
/*ro, req, int, 最小值*/
"@max": 255
/*ro, req, int, 最大值*/
},
"enableLiveDetAntiAttack": "true,false",
/*ro, opt, string, 活体检测防攻击使能, desc:活体检测防攻击使能*/
"fingerPrintCapacity": {

```

```

/*ro, opt, object, 指纹容量*/
"@min": 1,
/*ro, req, int, 最小值*/
"@max": 100
/*ro, req, int, 最大值*/
},
"fingerPrintNum": {
/*ro, opt, object, 已存在指纹数量*/
"@min": 1,
/*ro, req, int, 最小值*/
"@max": 100
/*ro, req, int, 最大值*/
},
"faceRecognizeEnable": {
/*ro, opt, object, 人脸识别*/
"@opt": "1,2,3"
/*ro, req, string, 选项, desc:1-开启, 2-关闭, 3-多人脸*/
},
"enableReverseCardNo": "true,false",
/*ro, opt, string, 卡号翻转使能*/
"independSwipeIntervals": {
/*ro, opt, object, 人员认证间隔时间, desc: (每个人单独计算间隔, 区别于swipeInterval), 单位: 秒*/
"@min": 1,
/*ro, req, int, 最小值*/
"@max": 1
/*ro, req, int, 最大值*/
},
"maskFaceMatchThresholdN": {
/*ro, opt, object, 口罩人脸1VN比对阈值, desc:范围0-100*/
"@min": 0,
/*ro, req, int, 最小值*/
"@max": 100
/*ro, req, int, 最大值*/
},
"maskFaceMatchThreshold1": {
/*ro, opt, object, 口罩人脸1V1比对阈值, desc:范围0-100*/
"@min": 0,
/*ro, req, int, 最小值*/
"@max": 100
/*ro, req, int, 最大值*/
},
},
}
}

```

69.47.11 配置智能身份识别终端参数

Request URL

PUT /ISAPI/AccessControl/IdentityTerminal

查询参数

无

请求报文

```

<?xml version="1.0" encoding="UTF-8"?>

<IdentityTerminal xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, 智能身份识别终端参数, attr:version{req, string, 协议版本}-->
  <terminalMode>
    <!--req, enum, 终端模式, subType:string, [authMode#认证模式,registerMode#登记模式]-->authMode
  </terminalMode>
  <idCardReader>
    <!--req, enum, 身份证阅读器, subType:string, [iDR210#iDR210,DS-K1F110-I,DS-K1F110-I,DS-K1F110-B#DS-K1F110-B,DS-K1F110-AB#DS-K1F110-AB,none#none,DS-K1F1001-I(USB)#DS-K1F1001-I(USB),DS-K1F1002-I(USB)#DS-K1F1002-I(USB)]-->iDR210
  </idCardReader>
  <camera>
    <!--req, enum, 摄像头, subType:string, [C270#C270,DS-2CS5432B-S#DS-2CS5432B-S]-->C270
  </camera>
  <fingerPrintModule>
    <!--req, enum, 指纹模组, subType:string, [ALIWARD#ALIWARD,HikModuLe#HikModuLe]-->ALIWARD
  </fingerPrintModule>
  <videoStorageTime>
    <!--req, int, 视频存储时间, range:[0,10], desc:单位: 秒-->1
  </videoStorageTime>
  <faceContrastThreshold>
    <!--req, int, 人脸比对阈值, range:[0,100]-->1
  </faceContrastThreshold>
  <twoDimensionCode>
    <!--req, enum, 二维码识别使能, subType:string, [enable#enable,disable#disable]-->enable
  </twoDimensionCode>
  <blackListCheck>
    <!--req, enum, 黑名单核验使能, subType:string, [enable#enable,disable#disable]-->enable
  </blackListCheck>
  <idCardCheckCenter>
    <!--req, enum, 身份证核验中心配置, subType:string, [Local#设备本地比对,server#远程服务器比对]-->local
  </idCardCheckCenter>
  <faceAlgorithm>

```

```
<!--req, enum, 人脸算法, subType:string, [DeepLearn#研究院算法,Tradition#第三方算法]-->DeepLearn
</faceAlgorithm>
<comNo>
  <!--req, int, COM号, range:[1,9]-->1
</comNo>
<memoryLearning>
  <!--req, enum, 记忆学习功能使能, subType:string, [enable#enable,disable#disable]-->enable
</memoryLearning>
<saveCertifiedImage>
  <!--req, enum, 保存认证图片使能, subType:string, [enable#enable,disable#disable]-->enable
</saveCertifiedImage>
<MCUVersion>
  <!--opt, string, MCU版本-->test
</MCUVersion>
<usbOutput>
  <!--opt, enum, 利旧身份证读卡器USB输出使能, subType:string, [enable#enable,disable#disable]-->enable
</usbOutput>
<serialOutput>
  <!--opt, enum, 利旧身份证读卡器串口输出使能, subType:string, [enable#enable,disable#disable]-->enable
</serialOutput>
<readInfoOfCard>
  <!--opt, enum, CPU卡读取内容配置, subType:string, [serialNo#读取序列号,file#读取文件]-->serialNo
</readInfoOfCard>
<workMode>
  <!--opt, enum, 认证工作模式, subType:string, [passMode#直通模式,accessControlMode#门禁模式], desc:直通模式下支持人证比对, 门禁模式下不支持人证比对(可做卡认证、人员认证等)-->passMode
</workMode>
<ecoMode>
  <!--opt, object, eco模式-->
  <eco>
    <!--opt, enum, 环保模式使能, subType:string, [enable#enable,disable#disable]-->enable
  </eco>
  <faceMatchThreshold1>
    <!--req, int, 环保模式人脸比对阈值1V1, range:[0,100]-->1
  </faceMatchThreshold1>
  <faceMatchThresholdN>
    <!--req, int, 环保模式人脸比对阈值1VN, range:[0,100]-->1
  </faceMatchThresholdN>
  <changeThreshold>
    <!--opt, int, 环保模式切换阈值, desc:"0,1,2,3,4,5,6,7,8"-->0
  </changeThreshold>
  <maskFaceMatchThresholdN>
    <!--opt, int, 环保模式人脸比对(口罩人脸&正常底图)阈值1VN, range:[0,100]-->1
  </maskFaceMatchThresholdN>
  <maskFaceMatchThreshold1>
    <!--opt, int, 环保模式人脸比对(口罩人脸&正常底图)阈值1V1, range:[0,100]-->1
  </maskFaceMatchThreshold1>
  <alwaysInfrared>
    <!--opt, bool, 总是开启红外识别使能, desc: 存在该字段, 则changeThreshold字段值为8不再代表总是开启红外识别-->true
  </alwaysInfrared>
  <ageFaceMatchList>
    <!--opt, array, 环保模式年龄段属性检测列表, subType:object, range:[0,5], desc:前提是已开启年龄段属性检测, 见PUT /ISAPI/AccessControl/FaceCompareCond, ageFaceMatchEnabled为true-->
    <ageFaceMatch>
      <!--opt, object, 环保模式年龄段属性检测-->
      <ageLevel>
        <!--opt, enum, 年龄段等级, subType:int, [0#儿童 0-6岁,1#少年 7-17岁,2#青年 18-40岁,3#中年 41-65岁,4#老年 >=66]-->1
      </ageLevel>
      <ageFaceMatchThreshold1>
        <!--opt, int, 环保模式年龄段属性人脸比对阈值1V1, range:[0,100]-->1
      </ageFaceMatchThreshold1>
      <ageFaceMatchThresholdN>
        <!--opt, int, 环保模式年龄段属性人脸比对阈值1VN, range:[0,100]-->1
      </ageFaceMatchThresholdN>
    </ageFaceMatch>
  </ageFaceMatchList>
</ecoMode>
<readCardRule>
  <!--opt, enum, 刷卡卡号规则, subType:string, [wiegand26#wiegand26,wiegand34#wiegand34]-->wiegand26
</readCardRule>
<enableScreenOff>
  <!--opt, bool, 是否启用息屏-->true
</enableScreenOff>
<screenOffTimeout>
  <!--opt, int, 息屏时间, desc:单位秒-->1
</screenOffTimeout>
<enableScreensaver>
  <!--opt, bool, 是否启用屏保功能-->true
</enableScreensaver>
<faceModuleVersion>
  <!--opt, string, 人脸识别组件版本, range:[1,32]-->test
</faceModuleVersion>
<showMode>
  <!--opt, enum, 显示模式, subType:string, [concise#简洁模式,normal#正常模式(默认),advertising#广告模式,meeting#会议模式,selfDefine#自定义模式,boxStatus#格口状态模式], desc:无显示模式字段,默认在正常模式,简洁模式是指仅显示认证结果,不显示工号、姓名等信息以及预览信息;普通模式即正常明眸认证界面;广告模式即上下分屏,上屏广告,下屏认证;会议模式即会议签到界面;自定义模式即用用户自定义UI的主界面、认证成功、认证失败界面,使用信息发布协议下发自定义布局;格口状态模式是智能柜在待机界面时展示格口状态-->concise
</showMode>
<popupPreviewWindow>
  <!--opt, bool, 是否弹出预览窗口, dep:or,{$.IdentityTerminal.showMode,eq,advertising}-->true
</popupPreviewWindow>
<needDeviceCheck>
  <!--opt, bool, 直通模式是否需要设备校验, dep:or,{$.IdentityTerminal.workMode,eq,passMode}-->true
</needDeviceCheck>
```

```
</needDeviceCheck>
</IdentityTerminal>
```

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 响应状态, attr:version{req, string, 协议版本}-->
  <requestURL>
    <!--ro, req, string, 请求URL, desc:请求的URL-->test
  </requestURL>
  <statusCode>
    <!--ro, req, enum, 状态码, subType:int, [0#OK,1#OK,2#Device Busy,3#Device Error,4#Invalid Operation,5#Invalid XML Format,6#Invalid XML Content,7#Reboot Required], desc:0和1-OK, 2-Device Busy, 3-Device Error, 4-Invalid Operation, 5-Invalid XML Format, 6-Invalid XML Content; 7-Reboot Required-->1
  </statusCode>
  <statusString>
    <!--ro, req, enum, 状态描述, subType:string, [OK#成功,Device Busy#设备繁忙,Device Error#设备出错,Invalid Operation#非法操作,Invalid XML Format#非法XML,Invalid XML Content#非法XML内容,Reboot#重启], desc:OK,Device Busy,Device Error,Invalid Operation,Invalid XML Format,Invalid XML Content,Reboot-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, 子状态码, desc:详细错误码的英文描述, 可描述的很详细-->test
  </subStatusCode>
</ResponseStatus>
```

69.47.12 获取智能身份识别终端参数

Request URL

GET /ISAPI/AccessControl/IdentityTerminal

查询参数

无

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<IdentityTerminal xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 智能身份识别终端参数, attr:version{req, string, 协议版本}-->
  <terminalMode>
    <!--ro, req, enum, 终端模式, subType:string, [authMode#认证模式,registerMode#登记模式]-->authMode
  </terminalMode>
  <idCardReader>
    <!--ro, req, enum, 身份证阅读器, subType:string, [iDR210#iDR210,DS-K1F110-I#DS-K1F110-I,DS-K1F1110-B#DS-K1F1110-B,DS-K1F1110-AB#DS-K1F1110-AB,none#none,DS-K1F1001-I(USB)#DS-K1F1001-I(USB),DS-K1F1002-I(USB)#DS-K1F1002-I(USB)]-->iDR210
  </idCardReader>
  <camera>
    <!--ro, req, enum, 摄像头, subType:string, [C270#C270,DS-2C55432B-S#DS-2C55432B-S]-->C270
  </camera>
  <fingerPrintModule>
    <!--ro, req, enum, 指纹模组, subType:string, [ALIWARD#ALIWARD,HikModule#HikModule]-->ALIWARD
  </fingerPrintModule>
  <videoStorageTime>
    <!--ro, req, int, 视频存储时间, range:[0,10], desc:单位: 秒-->1
  </videoStorageTime>
  <faceContrastThreshold>
    <!--ro, req, int, 人脸比对阈值, range:[0,100]-->1
  </faceContrastThreshold>
  <twoDimensionCode>
    <!--ro, req, enum, 二维码识别使能, subType:string, [enable#enable,disable#disable]-->enable
  </twoDimensionCode>
  <blackListCheck>
    <!--ro, req, enum, 黑名单核验使能, subType:string, [enable#enable,disable#disable]-->enable
  </blackListCheck>
  <idCardCheckCenter>
    <!--ro, req, enum, 身份证核验中心配置, subType:string, [Local#设备本地比对,server#远程服务器比对]-->local
  </idCardCheckCenter>
  <faceAlgorithm>
    <!--ro, req, enum, 人脸算法, subType:string, [DeepLearn#研究院算法,Tradition#第三方算法]-->DeepLearn
  </faceAlgorithm>
  <comNo>
    <!--ro, req, int, COM号, range:[1,9]-->1
  </comNo>
  <memoryLearning>
    <!--ro, req, enum, 记忆学习功能使能, subType:string, [enable#enable,disable#disable]-->enable
  </memoryLearning>
  <saveCertifiedImage>
    <!--ro, req, enum, 保存认证图片使能, subType:string, [enable#enable,disable#disable]-->enable
  </saveCertifiedImage>
  <MCUVersion>
```

```

<!--ro, opt, string, MCU版本-->test
</MCUVersion>
<usbOutput>
  <!--ro, opt, enum, 利旧身份证读卡器USB输出使能, subType:string, [enable#enable,disable#disable]-->enable
</usbOutput>
<serialOutput>
  <!--ro, opt, enum, 利旧身份证读卡器串口输出使能, subType:string, [enable#enable,disable#disable]-->enable
</serialOutput>
<readInfoOfCard>
  <!--ro, opt, enum, CPU卡读取内容配置, subType:string, [serialNo#读取序列号,file#读取文件]-->serialNo
</readInfoOfCard>
<workMode>
  <!--ro, opt, enum, 认证工作模式, subType:string, [passMode#直通模式,accessControlMode#门禁模式], desc:直通模式下支持人证比对, 门禁模式下不支持人证比对
(可做卡认证、人员认证等)-->passMode
</workMode>
<ecoMode>
  <!--ro, opt, object, eco模式-->
  <eco>
    <!--ro, opt, enum, 环保模式使能, subType:string, [enable#enable,disable#disable]-->enable
  </eco>
  <faceMatchThreshold1>
    <!--ro, req, int, 环保模式人脸比对阈值1V1, range:[0,100]-->1
  </faceMatchThreshold1>
  <faceMatchThresholdN>
    <!--ro, req, int, 环保模式人脸比对阈值1VN, range:[0,100]-->1
  </faceMatchThresholdN>
  <changeThreshold>
    <!--ro, opt, int, 环保模式切换阈值, desc:"0,1,2,3,4,5,6,7,8"-->0
  </changeThreshold>
  <maskFaceMatchThresholdN>
    <!--ro, opt, int, 环保模式人脸比对(口罩人脸&正常底图)阈值1VN, range:[0,100]-->0
  </maskFaceMatchThresholdN>
  <maskFaceMatchThreshold1>
    <!--ro, opt, int, 环保模式人脸比对(口罩人脸&正常底图)阈值1V1, range:[0,100]-->0
  </maskFaceMatchThreshold1>
  <alwaysInfrared>
    <!--ro, opt, bool, 总是开启红外识别使能, desc:存在该字段, 则changeThreshold字段值为8不再代表总是开启红外识别-->true
  </alwaysInfrared>
  <ageFaceMatchList>
    <!--ro, opt, array, 环保模式年龄段属性检测列表, subType:object, range:[0,5]-->
    <ageFaceMatch>
      <!--ro, opt, object, 环保模式年龄段属性检测-->
      <ageLevel>
        <!--ro, opt, enum, 年龄段等级, subType:int, [0#儿童 0-6岁,1#少年 7-17岁,2#青年 18-40岁,3#中年 41-65岁,4#老年 >=66]-->1
      </ageLevel>
      <ageFaceMatchThreshold1>
        <!--ro, opt, int, 环保模式年龄段属性人脸比对阈值1V1, range:[0,100]-->1
      </ageFaceMatchThreshold1>
      <ageFaceMatchThresholdN>
        <!--ro, opt, int, 环保模式年龄段属性人脸比对阈值1VN, range:[0,100]-->1
      </ageFaceMatchThresholdN>
    </ageFaceMatch>
  </ageFaceMatchList>
</ecoMode>
<readCardRule>
  <!--ro, opt, enum, 刷卡卡号规则, subType:string, [wiegand26#wiegand26,wiegand34#wiegand34]-->wiegand26
</readCardRule>
<enableScreenOff>
  <!--ro, opt, bool, 是否启用息屏-->true
</enableScreenOff>
<screenOffTimeout>
  <!--ro, opt, int, 息屏时间, desc:单位秒-->1
</screenOffTimeout>
<enableScreensaver>
  <!--ro, opt, bool, 是否启用屏保功能-->true
</enableScreensaver>
<faceModuleVersion>
  <!--ro, opt, string, 人脸识别组件版本, range:[1,32]-->test
</faceModuleVersion>
<showMode>
  <!--ro, opt, enum, 显示模式, subType:string, [concise#简洁模式,normal#正常模式(默认),advertising#广告模式,meeting#会议模式,selfDefine#自定义模
式,boxStatus#格口状态模式], desc:无显示模式字段,默认在正常模式,简洁模式是指仅显示认证结果,不显示工号、姓名等信息以及预览信息;普通模式即正常明眸认证界
面;广告模式即上下分屏,上屏广告,下屏认证;会议模式即会议签到界面;自定义模式即用户自定义UI的主界面、认证成功、认证失败界面,使用信息发布协议下发自定义布
局;格口状态模式是智能柜在待机界面时展示格口状态-->concise
</showMode>
<popupPreviewWindow>
  <!--ro, opt, bool, 是否弹出预览窗口, dep:or,{$.IdentityTerminal.showMode,eq,advertising}-->true
</popupPreviewWindow>
<needDeviceCheck>
  <!--ro, opt, bool, 直通模式是否需要设备校验, dep:or,{$.IdentityTerminal.workMode,eq,passMode}-->true
</needDeviceCheck>
</IdentityTerminal>

```

69.47.13 获取智能身份识别终端参数配置能力

Request URL

GET /ISAPI/AccessControl/IdentityTerminal/capabilities

查询参数

无

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<IdentityTerminal xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 智能身份识别终端参数, attr:version{req, string, 协议版本}-->
  <terminalMode opt="authMode,registerMode">
    <!--ro, req, enum, 终端模式, subType:string, [authMode#认证模式,registerMode#登记模式], attr:opt{req, string, 取值范围}-->authMode
  </terminalMode>
  <idCardReader opt="iDR210,DS-K1F110-I,DS-K1F1110-B,DS-K1F1110-AB,none ">
    <!--ro, req, enum, 身份证阅读器, subType:string, [iDR210#iDR210,DS-K1F110-I#DS-K1F110-I,DS-K1F1110-B#DS-K1F1110-B,DS-K1F1110-AB#DS-K1F1110-AB,none#none,DS-K1F1001-I(USB)#DS-K1F1001-I(USB),DS-K1F1002-I(USB)#DS-K1F1002-I(USB)], attr:opt{req, string, 取值范围}-->iDR210
  </idCardReader>
  <camera opt="C270,DS-2CS5432B-S">
    <!--ro, req, enum, 摄像头, subType:string, [C270#C270,DS-2CS5432B-S#DS-2CS5432B-S], attr:opt{req, string, 取值范围}-->C270
  </camera>
  <fingerPrintModule opt="ALIWARD,HikModule">
    <!--ro, req, enum, 指纹模组, subType:string, [ALIWARD#ALIWARD,HikModule#HikModule], attr:opt{req, string, 取值范围}-->ALIWARD
  </fingerPrintModule>
  <videoStorageTime min="0" max="10">
    <!--ro, req, int, 视频存储时间, range:[0,10], attr:min{req, int, 最小值},max{req, int, 最大值}, desc:(单位: 秒)-->1
  </videoStorageTime>
  <faceContrastThreshold min="0" max="100">
    <!--ro, req, int, 人脸比对阈值, range:[0,100], attr:min{req, int, 最小值},max{req, int, 最大值}-->1
  </faceContrastThreshold>
  <twoDimensionCode opt="enable,disable">
    <!--ro, req, enum, 二维码识别使能, subType:string, [enable#enable,disable#disable], attr:opt{req, string, 取值范围}-->enable
  </twoDimensionCode>
  <blackListCheck opt="enable,disable">
    <!--ro, req, enum, 黑名单核验使能, subType:object, [enable#enable,disable#disable], attr:opt{req, string, 取值范围}-->enable
  </blackListCheck>
  <idCardCheckCenter opt="local,server">
    <!--ro, req, enum, 身份证核验中心配置, subType:string, [Local#设备本地比对,server#远程服务器比对], attr:opt{req, string, 取值范围}-->local
  </idCardCheckCenter>
  <faceAlgorithm opt="DeepLearn,Tradition">
    <!--ro, req, enum, 人脸算法, subType:string, [DeepLearn#研究院算法,Tradition#第三方算法], attr:opt{req, string, 取值范围}-->DeepLearn
  </faceAlgorithm>
  <comNo min="1" max="9">
    <!--ro, req, int, COM号, range:[1,9], attr:min{req, int, 最小值},max{req, int, 最大值}-->1
  </comNo>
  <memoryLearning opt="enable,disable">
    <!--ro, req, enum, 记忆学习功能使能, subType:object, [enable#enable,disable#disable], attr:opt{req, string, 取值范围}-->enable
  </memoryLearning>
  <saveCertifiedImage opt="enable,disable">
    <!--ro, req, enum, 保存认证图片使能, subType:string, [enable#enable,disable#disable], attr:opt{req, string, 取值范围}-->enable
  </saveCertifiedImage>
  <MCUVersion min="1" max="10">
    <!--ro, opt, string, MCU版本, attr:min{req, int, 最小值},max{req, int, 最大值}-->test
  </MCUVersion>
  <usbOutput opt="enable,disable">
    <!--ro, req, enum, 利旧身份证读卡器USB输出使能, subType:string, [enable#启用,disable#禁用], attr:opt{req, string, 取值范围}-->enable
  </usbOutput>
  <serialOutput opt="enable,disable">
    <!--ro, req, enum, 利旧身份证读卡器串口输出使能, subType:string, [enable#启用,disable#禁用], attr:opt{req, string, 取值范围}-->enable
  </serialOutput>
  <readInfoOfCard opt="serialNo,file">
    <!--ro, opt, enum, CPU卡读取内容配置, subType:string, [serialNo#读取序列号,file#读取文件], attr:opt{req, string, 取值范围}-->serialNo
  </readInfoOfCard>
  <workMode opt="passMode,accessControlMode">
    <!--ro, opt, enum, 认证工作模式, subType:string, [passMode#直通模式,accessControlMode#门禁模式], attr:opt{req, string, 取值范围}, desc:直通模式下支持人证比对, 门禁模式下不支持人证比对(可做卡认证、人员认证等)-->passMode
  </workMode>
  <ecoMode>
    <!--ro, opt, object, eco模式-->
    <eco opt="enable,disable">
      <!--ro, opt, enum, 环保模式使能, subType:string, [enable#启用,disable#禁用], attr:opt{req, string, 取值范围}-->enable
    </eco>
    <faceMatchThreshold1 min="0" max="100">
      <!--ro, req, int, 环保模式人脸比对阈值1V1, range:[0,100], attr:min{req, int, 最小值},max{req, int, 最大值}-->1
    </faceMatchThreshold1>
    <faceMatchThresholdN min="0" max="100">
      <!--ro, req, int, 环保模式人脸比对阈值1VN, range:[0,100], attr:min{req, int, 最小值},max{req, int, 最大值}-->1
    </faceMatchThresholdN>
    <changeThreshold min="1" max="10">
      <!--ro, opt, int, 环保模式切换阈值, attr:min{req, int, 最小值},max{req, int, 最大值}, desc:"0-8"-->0
    </changeThreshold>
    <maskFaceMatchThresholdN min="0" max="100">
      <!--ro, opt, int, 环保模式人脸比对(口罩人脸&正常底图)阈值1VN, range:[0,100], attr:min{req, int, 最小值},max{req, int, 最大值}-->1
    </maskFaceMatchThresholdN>
    <maskFaceMatchThreshold1 min="0" max="100">
      <!--ro, opt, int, 环保模式人脸比对(口罩人脸&正常底图)阈值1V1, range:[0,100], attr:min{req, int, 最小值},max{req, int, 最大值}-->1
    </maskFaceMatchThreshold1>
    <alwaysInfrared opt="true,false">
```

```

<!--ro, opt, bool, 总是开启红外识别使能, attr:opt{req, string, 取值范围}, desc: 存在该字段, 则changeThreshold字段值为8个再代表总是开启红外识别-->true
</alwaysInfrared>
<ageFaceMatchList size="5">
  <!--ro, opt, array, 环保模式年龄段属性检测列表, subType:object, range:[0,5], attr:size{req, int, 数组元素总数}-->
  <ageFaceMatch>
    <!--ro, opt, object, 环保模式年龄段属性检测-->
    <ageLevel opt="0,1,2,3,4">
      <!--ro, opt, enum, 年龄段等级, subType:int, [0#儿童 0-6岁,1#少年 7-17岁,2#青年 18-40岁,3#中年 41-65岁,4#老年 >=66], attr:opt{req, string, 取值范围}-->1
    </ageLevel>
    <ageFaceMatchThreshold1 min="0" max="100">
      <!--ro, opt, int, 环保模式年龄段属性人脸比对阈值1V1, range:[0,100], attr:min{req, int, 最小值},max{req, int, 最大值}-->1
    </ageFaceMatchThreshold1>
    <ageFaceMatchThresholdN min="0" max="100">
      <!--ro, opt, int, 环保模式年龄段属性人脸比对阈值1VN, range:[0,100], attr:min{req, int, 最小值},max{req, int, 最大值}-->1
    </ageFaceMatchThresholdN>
  </ageFaceMatch>
</ageFaceMatchList>
</ecoMode>
<readCardRule opt="wiegand26,wiegand34">
  <!--ro, opt, enum, 刷卡卡号规则, subType:string, [wiegand26#wiegand26,wiegand34#wiegand34], attr:opt{req, string, 取值范围}, desc:后续弃用, 使用卡号认证模式配置接口进行切换-->wiegand26
</readCardRule>
<enableScreenOff opt="true,false">
  <!--ro, opt, bool, 是否启用息屏, attr:opt{req, string, 取值范围}-->true
</enableScreenOff>
<screenOffTimeout min="0" max="10">
  <!--ro, opt, int, 息屏时间, attr:min{req, int, 最小值},max{req, int, 最大值}, desc:单位秒-->1
</screenOffTimeout>
<enableScreensaver opt="true,false">
  <!--ro, opt, bool, 是否启用屏保功能, attr:opt{req, string, 取值范围}-->true
</enableScreensaver>
<faceModuleVersion min="0" max="10">
  <!--ro, opt, string, 人脸识别组件版本, attr:min{req, int, 最小值},max{req, int, 最大值}-->test
</faceModuleVersion>
<showMode opt="concise,normal,advertising,meeting,selfDefine,boxStatus">
  <!--ro, opt, enum, 显示模式, subType:string, [concise#简洁模式,normal#正常模式(默认),advertising#广告模式,meeting#会议模式,selfDefine#自定义模式,boxStatus#格口状态模式], attr:opt{req, string, 取值范围}, desc:无显示模式字段, 默认在正常模式, 简洁模式是指仅显示认证结果, 不显示工号、姓名等信息以及预览信息; 普通模式即正常明眸认证界面; 广告模式即上下分屏, 上屏广告, 下屏认证; 会议模式即会议签到界面; 自定义模式即用户自定义UI的主界面、认证成功、认证失败界面, 使用信息发布协议下发自定义布局-->concise
</showMode>
<popUpPreviewWindow opt="true,false">
  <!--ro, opt, bool, 是否弹出预览窗口, dep:or,{$.IdentityTerminal.showMode,eq,advertising}, attr:opt{req, string, 取值范围}-->true
</popUpPreviewWindow>
<needDeviceCheck opt="true,false">
  <!--ro, opt, bool, 直通模式是否需要设备校验, dep:or,{$.IdentityTerminal.workMode,eq,passMode}, attr:opt{req, string, 取值范围}-->true
</needDeviceCheck>
</IdentityTerminal>

```

69.47.14 配置人脸识别参数

Request URL

PUT /ISAPI/AccessControl/FaceCompareCond

查询参数

无

请求报文

```
<?xml version="1.0" encoding="UTF-8"?>

<FaceCompareCond xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, 人脸识别参数, attr:version{opt, string, 协议版本}-->
  <faceWidthLowerLimit>
    <!--opt, int, 人脸宽度阈值, range:[0,100], desc:当待测人脸宽度超过该阈值, 直接忽略后面的限制条件, 将执行人脸比对。优先级最高-->50
  </faceWidthLowerLimit>
  <pitch>
    <!--opt, int, 人脸上下仰角, range:[0,90], unitType:立体角, desc:越小越好-->0
  </pitch>
  <yaw>
    <!--opt, int, 人脸左右偏转角, range:[0,90], unitType:立体角, desc:越小越好-->0
  </yaw>
  <width>
    <!--opt, int, 人脸宽度, range:[0,100]-->50
  </width>
  <height>
    <!--opt, int, 人脸高度, range:[0,100]-->50
  </height>
  <leftBorder>
    <!--opt, int, 人脸左边界限, range:[0,100]-->50
  </leftBorder>
  <rightBorder>
    <!--opt, int, 人脸右边界限, range:[0,100]-->50
  </rightBorder>
  <upBorder>
    <!--opt, int, 人脸上边界限, range:[0,100]-->50
  </upBorder>
  <bottomBorder>
    <!--opt, int, 人脸下边界限, range:[0,100]-->50
  </bottomBorder>
  <interorbitalDistance>
    <!--opt, int, 人脸眼间距, range:[0,100]-->50
  </interorbitalDistance>
  <faceScore>
    <!--opt, int, 人脸评分, range:[0,100], desc:待测人脸大于该值合法-->50
  </faceScore>
  <maxDistance>
    <!--opt, enum, 识别最大距离, subType:string, [0.5#0.5m,1#1m,1.5#1.5m,2#2m,auto#自动]-->0.5
  </maxDistance>
  <similarity>
    <!--opt, float, 人脸比对相似度-->50
  </similarity>
  <antiFake>
    <!--opt, int, 人脸防假参数-->50
  </antiFake>
  <identifyType>
    <!--opt, enum, 人脸识别类型, subType:string, [highest#相似度最高,single#单个超过相似度,multiple#多个相似度], desc:相似度最高为对整个人脸库中所人脸进行比对, 选择相似度最高值返回;单个超过相似度返回第一个相似度超过阈值;多个相似度则返回前30个相似度超过阈值的人脸, 默认为相似度最高-->highest
  </identifyType>
  <chooseType>
    <!--opt, enum, 人脸选择类型, subType:string, [middle#最中间,biggest#最大的,all#全部], desc:最中间则为仅识别单幅画面中最中间的人脸;最大的则为仅识别单幅画面中最大的人脸;全部为识别单幅画面中所有的人脸, 默认为全部-->all
  </chooseType>
  <enabled>
    <!--opt, enum, 人脸识别使能, subType:string, [singleFace#单人脸识别,close#关闭人脸识别,multiFace#多人脸识别], desc:默认单人脸识别, 用于使能整个设备的人脸识别功能, 针对所有读卡器均生效-->singleFace
  </enabled>
  <faceScoreEnabled>
    <!--opt, bool, 人脸评分使能, desc:人证比对或人脸采集时, 筛选质量高的图片, 与faceScore字段配合使用-->true
  </faceScoreEnabled>
  <ageFaceMatchEnabled>
    <!--opt, bool, 年龄段属性检测使能-->true
  </ageFaceMatchEnabled>
  <ageFaceMatchList>
    <!--opt, array, 年龄段属性检测列表, subType:object, range:[0,5], dep:or,{$.FaceCompareCond.ageFaceMatchEnabled,eq,true}-->
    <ageFaceMatch>
      <!--opt, object, 年龄段属性检测-->
      <ageLevel>
        <!--opt, enum, 年龄段等级, subType:int, [0#儿童 0-6岁,1#少年 7-17岁,2#青年 18-40岁,3#中年 41-65岁,4#老年 >=66]-->1
      </ageLevel>
      <ageFaceMatchThreshold1>
        <!--opt, int, 年龄段属性人脸比对阈值1V1, range:[0,100]-->1
      </ageFaceMatchThreshold1>
      <ageFaceMatchThresholdN>
        <!--opt, int, 年龄段属性人脸比对阈值1VN, range:[0,100]-->1
      </ageFaceMatchThresholdN>
      </ageFaceMatch>
    </ageFaceMatchList>
  </FaceCompareCond>
```

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 响应消息, attr:version{ro, req, string, 版本}-->
  <requestURL>
    <!--ro, req, string, 请求的URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, 状态码, subType:int, [0#OK,1#OK,2#Device Busy,3#Device Error,4#Invalid Operation,5#Invalid XML Format,6#Invalid XML Content,7#Reboot Required]-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, 状态信息, subType:string, [OK#成功,Device Busy#设备忙,Device Error#设备异常,Invalid Operation#无效的操作,Invalid XML Format#无效的XML格式,Invalid XML Content#无效的XML内容,Reboot#设备重启]-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, 详细错误码英文描述, desc:详细错误码的英文描述-->OK
  </subStatusCode>
</ResponseStatus>
```

69.47.15 获取人脸识别参数

Request URL

GET /ISAPI/AccessControl/FaceCompareCond

查询参数

无

请求报文

无

响应报文

湖南云舟电力科技有限公司
有效期：2024-08-15

```
<?xml version="1.0" encoding="UTF-8"?>

<FaceCompareCond xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 人脸识别参数, attr:version{opt, string, 协议版本}-->
  <faceWidthLowerLimit>
    <!--ro, opt, int, 人脸宽度阈值, range:[0,100], desc:当待测人脸宽度超过该阈值, 直接忽略后面的限制条件, 将执行人脸比对。优先级最高-->50
  </faceWidthLowerLimit>
  <pitch>
    <!--ro, opt, int, 人脸上下仰角, range:[0,90], unitType:立体角, desc:越小越好-->0
  </pitch>
  <yaw>
    <!--ro, opt, int, 人脸左右偏转角, range:[0,90], unitType:立体角, desc:越小越好-->0
  </yaw>
  <width>
    <!--ro, opt, int, 人脸宽度, range:[0,100]-->50
  </width>
  <height>
    <!--ro, opt, int, 人脸高度, range:[0,100]-->50
  </height>
  <leftBorder>
    <!--ro, opt, int, 人脸左边界限, range:[0,100]-->50
  </leftBorder>
  <rightBorder>
    <!--ro, opt, int, 人脸右边界限, range:[0,100]-->50
  </rightBorder>
  <upBorder>
    <!--ro, opt, int, 人脸上边界限, range:[0,100]-->50
  </upBorder>
  <bottomBorder>
    <!--ro, opt, int, 人脸下边界限, range:[0,100]-->50
  </bottomBorder>
  <interorbitalDistance>
    <!--ro, opt, int, 人脸眼间距, range:[0,100]-->50
  </interorbitalDistance>
  <faceScore>
    <!--ro, opt, int, 人脸评分, range:[0,100], desc:待测人脸大于该值合法-->50
  </faceScore>
  <maxDistance>
    <!--ro, opt, enum, 识别最大距离, subType:string, [0.5#0.5m,1#1m,1.5#1.5m,2#2m,auto#自动]-->0.5
  </maxDistance>
  <similarity>
    <!--ro, opt, float, 人脸比对相似度-->50
  </similarity>
  <antiFake>
    <!--ro, opt, int, 人脸防假参数-->50
  </antiFake>
  <identifyType>
    <!--ro, opt, enum, 人脸识别类型, subType:string, [highest#相似度最高,single#单个超过相似度,multiple#多个相似度], desc:相似度最高为对整个人脸库中所人脸进行比对, 选择相似度最高值返回;单个超过相似度返回第一个相似度超过阈值;多个相似度则返回前30个相似度超过阈值的人脸, 默认为相似度最高-->highest
  </identifyType>
  <chooseType>
    <!--ro, opt, enum, 人脸选择类型, subType:string, [middle#最中间,biggest#最大的,all#全部], desc:最中间则为仅识别单幅画面中最中间的人脸;最大的则为仅识别单幅画面中最大的人脸;全部为识别单幅画面中所有的人脸, 默认为全部-->all
  </chooseType>
  <enabled>
    <!--ro, opt, enum, 人脸识别使能, subType:string, [singleFace#单人脸识别,close#关闭人脸识别,multiFace#多人脸识别], desc:默认单人脸识别。用于使能整个设备的人脸识别功能, 针对所有读卡器均生效-->singleFace
  </enabled>
  <faceScoreEnabled>
    <!--ro, opt, bool, 人脸评分使能, desc:人证比对或人脸采集时, 筛选质量高的图片, 与faceScore字段配合使用-->true
  </faceScoreEnabled>
  <ageFaceMatchEnabled>
    <!--ro, opt, bool, 年龄段属性检测使能-->true
  </ageFaceMatchEnabled>
  <ageFaceMatchList>
    <!--ro, opt, array, 年龄段属性检测列表, subType:object, range:[0,5], dep:or,{$.FaceCompareCond.ageFaceMatchEnabled,eq,true}-->
    <ageFaceMatch>
      <!--ro, opt, object, 年龄段属性检测-->
      <ageLevel>
        <!--ro, opt, enum, 年龄段等级, subType:int, [0#儿童 0-6岁,1#少年 7-17岁,2#青年 18-40岁,3#中年 41-65岁,4#老年 >=66]-->1
      </ageLevel>
      <ageFaceMatchThreshold1>
        <!--ro, opt, int, 年龄段属性人脸比对阈值1V1, range:[0,100]-->1
      </ageFaceMatchThreshold1>
      <ageFaceMatchThresholdN>
        <!--ro, opt, int, 年龄段属性人脸比对阈值1VN, range:[0,100]-->1
      </ageFaceMatchThresholdN>
      </ageFaceMatch>
    </ageFaceMatchList>
  </FaceCompareCond>

```

69.47.16 获取人脸识别参数配置能力

Request URL

GET /ISAPI/AccessControl/FaceCompareCond/capabilities

查询参数

无

请求报文

无

响应报文

有效期：2024-08-15

湖南云舟电力科技有限公司

```

<?xml version="1.0" encoding="UTF-8"?>

<FaceCompareCond xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 人脸识别参数配置能力, attr:version{req, string, 协议版本}-->
  <faceWidthLowerLimit min="1" max="10">
    <!--ro, opt, int, 人脸宽度阈值, range:[0,100], attr:min{req, int, 最小值},max{req, int, 最大值}, desc:当待测人脸宽度超过该阈值, 直接忽略后面的限制条件, 将执行人脸比对。优先级最高-->1
  </faceWidthLowerLimit>
  <pitch min="1" max="10">
    <!--ro, opt, int, 人脸上下仰角, range:[0,90], attr:min{req, int, 最小值},max{req, int, 最大值}, desc:越小越好-->1
  </pitch>
  <yaw min="1" max="10">
    <!--ro, opt, int, 人脸左右偏转角, range:[0,90], attr:min{req, int, 最小值},max{req, int, 最大值}, desc:越小越好-->1
  </yaw>
  <width min="1" max="10">
    <!--ro, opt, int, 人脸宽度, range:[0,100], attr:min{req, int, 最小值},max{req, int, 最大值}-->1
  </width>
  <height min="1" max="10">
    <!--ro, opt, int, 人脸高度, range:[0,100], attr:min{req, int, 最小值},max{req, int, 最大值}-->1
  </height>
  <leftBorder min="1" max="10">
    <!--ro, opt, int, 人脸左边界限, range:[0,100], attr:min{req, int, 最小值},max{req, int, 最大值}-->1
  </leftBorder>
  <rightBorder min="1" max="10">
    <!--ro, opt, int, 人脸右边界限, range:[0,100], attr:min{req, int, 最小值},max{req, int, 最大值}-->1
  </rightBorder>
  <upBorder min="1" max="10">
    <!--ro, opt, int, 人脸上边界限, range:[0,100], attr:min{req, int, 最小值},max{req, int, 最大值}-->1
  </upBorder>
  <bottomBorder min="1" max="10">
    <!--ro, opt, int, 人脸下边界限, range:[0,100], attr:min{req, int, 最小值},max{req, int, 最大值}-->1
  </bottomBorder>
  <interorbitalDistance min="1" max="10">
    <!--ro, opt, int, 人脸眼间距, range:[0,100], attr:min{req, int, 最小值},max{req, int, 最大值}-->1
  </interorbitalDistance>
  <faceScore min="1" max="10">
    <!--ro, opt, int, 人脸评分, range:[0,100], attr:min{req, int, 最小值},max{req, int, 最大值}, desc:待测人脸大于该值合法-->1
  </faceScore>
  <maxDistance opt="0.5,1,1.5,2,auto">
    <!--ro, opt, string, 识别最大距离, attr:opt{req, string, 取值范围}, desc:单位m, 该字段优先级高于眼间距-->test
  </maxDistance>
  <similarity min="0.0" max="1.0">
    <!--ro, opt, float, 人脸比对相似度, attr:min{req, int, 最小值},max{req, int, 最大值}-->0.000
  </similarity>
  <antiFake min="1" max="10">
    <!--ro, opt, int, 人脸防假参数, attr:min{req, int, 最小值},max{req, int, 最大值}-->1
  </antiFake>
  <identifyType opt="highest,single,multipl">
    <!--ro, opt, enum, 人脸识别类型, subType:string, [highest#相似度最高,single#单个超过相似度,multiple#多个相似度], attr:opt{req, string, 取值范围}, desc:相似度最高为对整个人脸库中所人脸进行比对, 选择相似度最高值返回;单个超过相似度返回第一个相似度超过阈值;多个相似度则返回前30个相似度超过阈值的人脸, 默认为相似度最高-->highest
  </identifyType>
  <chooseType opt="middle,biggest,all">
    <!--ro, opt, enum, 人脸选择类型, subType:string, [middle#最中间,biggest#最大的,all#全部], attr:opt{req, string, 取值范围}, desc:最中间则为仅识别单幅画面中最中间的人脸;最大的则为仅识别单幅画面中最大的人脸;全部为识别单幅画面中所有的人脸, 默认为全部-->middle
  </chooseType>
  <enabled opt="singleFace,close,multiFace">
    <!--ro, opt, enum, 人脸识别使能, subType:string, [singleFace#单人脸识别,close#关闭人脸识别,multiFace#多人脸识别], attr:opt{req, string, 取值范围}, desc:默认单人脸识别, 用于使能整个设备的人脸识别功能, 针对所有摄像头均生效-->singleFace
  </enabled>
  <faceScoreEnabled opt="true,false">
    <!--ro, opt, bool, 人脸评分使能, attr:opt{req, string, 取值范围}, desc:人证比对或人脸采集时, 筛选质量高的图片, 与faceScore字段配合使用-->true
  </faceScoreEnabled>
  <ageFaceMatchEnabled opt="true,false">
    <!--ro, opt, bool, 年龄段属性检测使能, attr:opt{req, string, 取值范围}-->true
  </ageFaceMatchEnabled>
  <ageFaceMatchList size="5">
    <!--ro, opt, array, 年龄段属性检测列表, subType:object, range:[0,5], dep:or,{$.FaceCompareCond.ageFaceMatchEnabled,eq,true}, attr:size{req, int, 数组元素总数}-->
    <ageFaceMatch>
      <!--ro, opt, object, 年龄段属性检测-->
      <ageLevel opt="0,1,2,3,4">
        <!--ro, opt, enum, 年龄段等级, subType:int, [0#儿童 0-6岁,1#少年 7-17岁,2#青年 18-40岁,3#中年 41-65岁,4#老年 >=66], attr:opt{req, string, 取值范围}-->1
      </ageLevel>
      <ageFaceMatchThreshold1 min="0" max="100">
        <!--ro, opt, int, 年龄段属性人脸比对阈值1V1, range:[0,100], attr:min{req, int, 最小值},max{req, int, 最大值}-->1
      </ageFaceMatchThreshold1>
      <ageFaceMatchThresholdN min="0" max="100">
        <!--ro, opt, int, 年龄段属性人脸比对阈值1VN, range:[0,100], attr:min{req, int, 最小值},max{req, int, 最大值}-->1
      </ageFaceMatchThresholdN>
      </ageFaceMatch>
    </ageFaceMatchList>
  </FaceCompareCond>

```

69.47.17 获取韦根参数能力

Request URL

GET /ISAPI/AccessControl/WiegandCfg/capabilities

查询参数

无

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<WiegandCfg xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 韦根配置, attr:version{req, string, 协议版本}-->
  <wiegandNo min="1" max="4" opt="1,4">
    <!--ro, req, int, 韦根接口编号, range:[0,4], step:1, attr:min{opt, int, 最小值},max{opt, int, 最大值},opt{opt, string, 取值范围}, desc:闸机产品仅返回opt
    即可, 韦根编号仅支持1和4-->1
  </wiegandNo>
  <communicateDirection opt="receive,send">
    <!--ro, req, enum, 通信方向, subType:string, [receive#接收,send#发送], attr:opt{req, string, 取值范围}-->receive
  </communicateDirection>
  <wiegandMode
  opt="wiegand26,wiegand34,wiegand27,wiegand35,Corporate1000_35,Corporate1000_48,H10302_37,H10304_37,wiegand_26CSN,H103130_32CSN,wiegand_56CSN,wiegand_58">
    <!--ro, opt, enum, 输出方向下的韦根模式, subType:string, [wiegand26#韦根26,wiegand34#韦根34,wiegand27#韦根27,wiegand35#韦根35,Corporate1000_35#韦根
    Corporate 1000 35bits,Corporate1000_48#韦根Corporate 1000 48bits,H10302_37#韦根H10302 37bits,H10304_37#韦根H10304 37bits,wiegand_26CSN#韦根26
    CSN,H103130_32CSN#韦根H103130 32bits CSN,wiegand_56CSN#韦根56 CSN,wiegand_58#韦根58], attr:opt{req, string, 取值范围}-->wiegand26
  </wiegandMode>
  <inputWiegandMode
  opt="wiegand26,wiegand34,wiegand27,wiegand35,Corporate1000_35,Corporate1000_48,H10302_37,H10304_37,wiegand_26CSN,H103130_32CSN,wiegand_56CSN,wiegand_58">
    <!--ro, opt, enum, 输入方向下的韦根模式, subType:string, [wiegand26#韦根26,wiegand34#韦根34,wiegand27#韦根27,wiegand35#韦根35,Corporate1000_35#韦根
    Corporate 1000 35bits,Corporate1000_48#韦根Corporate 1000 48bits,H10302_37#韦根H10302 37bits,H10304_37#韦根H10304 37bits,wiegand_26CSN#韦根26
    CSN,H103130_32CSN#韦根H103130 32bits CSN,wiegand_56CSN#韦根56 CSN,wiegand_58#韦根58], dep:or,{$.WiegandCfg.communicateDirection,eq,receive}, attr:opt{req,
    string, 取值范围}-->wiegand26
  </inputWiegandMode>
  <signalInterval min="1" max="20">
    <!--ro, opt, int, 韦根信号发送间隔, range:[1,20], unitType:时间, attr:min{req, int, 最小值},max{req, int, 最大值}, desc:单位 ms-->1
  </signalInterval>
  <enable opt="true,false">
    <!--ro, opt, bool, 使能开关, attr:opt{req, string, 取值范围}-->true
  </enable>
  <pulseDuration min="1" max="10">
    <!--ro, opt, int, 脉冲宽度, range:[1,10], attr:min{req, int, 最小值},max{req, int, 最大值}, desc:单位us-->1
  </pulseDuration>
  <facilityCodeEnabled opt="true,false">
    <!--ro, opt, bool, facilityCode使能, attr:opt{req, string, 取值范围}-->true
  </facilityCodeEnabled>
  <facilityCode min="0" max="65535">
    <!--ro, opt, int, facilityCode, range:[0,65535], dep:and,{$.WiegandCfg.facilityCodeEnabled,eq,true}, attr:min{req, int, 最小值, range:
    [0,65535]},max{req, int, 最大值, range:[0,65535]},
    desc:当wiegandMode为输出时, facilityCode可配置, 且生效 (用于设备输出卡号+facilityCode至三方设备进行校验)
    当wiegandMode为输入时, facilityCode可配置, 但不生效 (即设备接收韦根数据时, 不对输入的facilityCode进行校验) -->1
  </facilityCode>
  <dataType opt="employeeNo,cardNo">
    <!--ro, opt, enum, 数据类型, subType:string, [employeeNo#工号,cardNo#卡号], dep:or,{$.WiegandCfg.wiegandMode,eq,send}, attr:opt{req, string, 取值范围},
    desc:主机韦根口输出工号或卡号至三方主机-->employeeNo
  </dataType>
</WiegandCfg>
```

69.47.18 获取人脸识别模式参数配置能力

Request URL

GET /ISAPI/AccessControl/FaceRecognizeMode/capabilities?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "FaceRecognizeMode": {
    /*ro, opt, object, 人脸识别模式*/
    "mode": {
      /*ro, req, object, 模式*/
      "@opt": "normalMode"
      /*ro, opt, enum, 模式选项, subType:string, [normalMode#普通模式,deepMode#深度模式], desc:normalMode-普通模式  deepMode-深度模式*/
    }
  }
}
```

69.47.19 设置NFC功能启用配置

Request URL

PUT /ISAPI/AccessControl/Configuration/NFCCfg?format=json

查询参数

无

请求报文

```
{
  "NFCCfg": {
    /*req, object, NFC功能启用配置*/
    "enable": true
    /*req, bool, 是否启用NFC功能, desc:true-启用, false-不启用（默认）*/
  }
}
```

响应报文

```
{
  "requestURL": "test",
  /*ro, opt, string, URI*/
  "statusCode": 1,
  /*ro, opt, int, 状态码, desc:无法用1表示时（1表示成功且无特殊状态），否则必须返回*/
  "statusString": "test",
  /*ro, opt, string, 状态描述, desc:无法用ok表示时（ok表示成功且无特殊状态），否则必须返回*/
  "subStatusCode": "test",
  /*ro, opt, string, 子状态码, desc:无法用ok表示（ok表示成功且无特殊状态），否则必须返回*/
  "errorCode": 1,
  /*ro, req, int, 错误码, desc:当statusCode不为1时，与subStatusCode对应*/
  "errorMsg": "ok"
  /*ro, req, string, 错误信息, desc:对errorCode的解释，当statusCode不为1时，解释信息在协议约束中，允许设备在后续的版本迭代中，进行优化丰富提升（不限制死）*/
}
```

69.47.20 获取NFC功能启用配置

Request URL

GET /ISAPI/AccessControl/Configuration/NFCCfg?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "NFCCfg": {
    /*ro, req, object, NFC功能启用配置*/
    "enable": true
    /*ro, req, bool, 是否启用NFC功能, desc:true-启用, false-不启用（默认）*/
  }
}
```

69.47.21 配置人脸识别模式参数

Request URL

PUT /ISAPI/AccessControl/FaceRecognizeMode?format=json

查询参数

无

请求报文

```
{
  "FaceRecognizeMode": {
    /*opt, object, 人脸识别模式*/
    "mode": "normalMode"
    /*req, enum, 人脸识别模式, subType:string, [normalMode#普通模式,deepMode#深度模式], desc:normalMode-普通模式    deepMode-深度模式*/
  }
}
```

响应报文

```
{
  "requestURL": "/ISAPI/Intelligent/FDLib/asyncImportDatas?format=json",
  /*ro, opt, string, 请求URL*/
  "statusCode": 1,
  /*ro, opt, int, 状态码, desc:无法用1表示时（1表示成功且无特殊状态）必须返回*/
  "statusString": "OK",
  /*ro, opt, string, 状态描述, range:[1,64], desc:无法用OK表示时（OK表示成功且无特殊状态）必须返回*/
  "subStatusCode": "ok",
  /*ro, opt, string, 子状态码, range:[1,64], desc:无法用ok表示（ok表示成功且无特殊状态）必须返回*/
  "errorCode": 1,
  /*ro, opt, int, 错误码, desc:当statusCode不为1时，与subStatusCode对应*/
  "errorMsg": "ok"
  /*ro, opt, string, 错误信息, desc:当statusCode不为1时，解释信息在协议约束中，允许设备在后续的版本迭代中，进行优化丰富提升（不限制死）*/
}
```

69.47.22 获取人脸识别模式参数

Request URL

GET /ISAPI/AccessControl/FaceRecognizeMode?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "FaceRecognizeMode": {
    /*ro, opt, object, 人脸识别模式*/
    "mode": "normalMode"
    /*ro, req, enum, 人脸识别模式, subType:string, [normalMode#普通模式,deepMode#深度模式], desc:normalMode-普通模式    deepMode-深度模式*/
  }
}
```

69.47.23 M1卡加密验证功能配置能力

Request URL

GET /ISAPI/AccessControl/M1CardEncryptCfg/capabilities

查询参数

无

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<M1CardEncryptCfg xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, M1卡加密验证功能配置, attr:version{req, string, 协议版本}-->
  <enable opt="true,false">
    <!--ro, req, bool, 使能, attr:opt{req, string, 取值范围}-->true
  </enable>
  <sectionID min="0" max="100">
    <!--ro, req, int, 扇区ID, range:[0,100], attr:min{req, int, 最小值},max{req, int, 最大值}-->1
  </sectionID>
</M1CardEncryptCfg>
```

69.47.24 获取M1卡加密验证功能配置

Request URL

GET /ISAPI/AccessControl/M1CardEncryptCfg

查询参数

无

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<M1CardEncryptCfg xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, M1卡加密验证功能配置, attr:version{req, string, 协议版本}-->
  <enable>
    <!--ro, req, bool, 使能-->true
  </enable>
  <sectionID>
    <!--ro, req, int, 扇区ID, desc: 每次只配置一个扇区, M1卡从0开始-->1
  </sectionID>
</M1CardEncryptCfg>
```

69.47.25 设置M1卡加密验证功能配置

Request URL

PUT /ISAPI/AccessControl/M1CardEncryptCfg

查询参数

无

请求报文

```
<?xml version="1.0" encoding="UTF-8"?>

<M1CardEncryptCfg xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, M1卡加密验证功能配置, attr:version{req, string, 协议版本}-->
  <enable>
    <!--req, bool, 使能-->true
  </enable>
  <sectionID>
    <!--req, int, 扇区ID, desc: 每次只配置一个扇区, M1卡从0开始-->1
  </sectionID>
</M1CardEncryptCfg>
```

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 响应状态, attr:version{req, string, 协议版本}-->
  <requestURL>
    <!--ro, req, string, 请求URL, desc:请求的URL-->test
  </requestURL>
  <statusCode>
    <!--ro, req, enum, 状态码, subType:string, [0#OK,1#OK,2#Device Busy,3#Device Error,4#Invalid Operation,5#Invalid XML Format,6#Invalid XML Content,7#Reboot Required], desc:0和1-OK, 2-Device Busy, 3-Device Error, 4-Invalid Operation, 5-Invalid XML Format, 6-Invalid XML Content; 7-Reboot Required-->1
  </statusCode>
  <statusString>
    <!--ro, req, enum, 状态描述, subType:string, [OK#成功,Device Busy#设备繁忙,Device Error#设备出错,Invalid Operation#非法操作,Invalid XML Format#非法XML,Invalid XML Content#非法XML内容,Reboot#重启], desc:OK,Device Busy,Device Error,Invalid Operation,Invalid XML Format,Invalid XML Content,Reboot-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, 子状态码, desc:详细错误码的英文描述, 可描述的很详细-->test
  </subStatusCode>
</ResponseStatus>
```

69.47.26 获取主机的虹膜识别参数

Request URL

GET /ISAPI/AccessControl/irisRecognition?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "saveIrisPic": true
  /*ro, opt, bool, 是否保存虹膜底图*/
}
```

69.47.27 获取主机的虹膜识别参数能力

Request URL

GET /ISAPI/AccessControl/irisRecognition/capabilities?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "saveIrisPic": {
    /*ro, opt, object, 是否保存虹膜底图*/
    "@opt": [true, false]
    /*ro, req, array, 取值范围, subType:bool*/
  }
}
```

69.47.28 获取射频卡识别启用配置

Request URL

GET /ISAPI/AccessControl/Configuration/RFCardCfg?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "RFCardCfg": [
    /*ro, req, array, 射频卡识别启用配置, subType:object*/
    {
      "cardType": "EMCard",
      /*ro, req, enum, 卡类型, subType:string, [EMCard#EM卡,M1Card#M1卡,CPUCard#CPU卡,IDCard#ID卡,FelicaCard#Felica卡], desc:
["EMCard","M1Card","CPUCard","IDCard"]*/
      "enabled": true
      /*ro, req, bool, 使能, desc:[true,false]*/
    }
  ]
}
```

69.47.29 设置射频卡识别启用配置

Request URL

PUT /ISAPI/AccessControl/Configuration/RFCardCfg?format=json

查询参数

无

请求报文

```
{
  "RFCardCfg": [
    /*req, array, 射频卡识别启用配置, subType:object*/
    {
      "cardType": "EMCard",
      /*req, enum, 卡类型, subType:string, [EMCard#EM卡,M1Card#M1卡,CPUCard#CPU卡,IDCard#ID卡,FelicaCard#Felica卡]*/
      "enabled": true
      /*req, bool, 使能, desc:[true,false]*/
    }
  ]
}
```

响应报文

```
{
  "requestURL": "test",
  /*ro, opt, string, URI*/
  "statusCode": 1,
  /*ro, opt, int, 状态码, desc:无法用1表示时（1表示成功且无特殊状态），否则必须返回*/
  "statusString": "test",
  /*ro, opt, string, 状态描述, desc:无法用ok表示时（ok表示成功且无特殊状态），否则必须返回*/
  "subStatusCode": "test",
  /*ro, opt, string, 子状态码, desc:无法用ok表示（ok表示成功且无特殊状态），否则必须返回*/
  "errorCode": 1,
  /*ro, req, int, 错误码, desc:当statusCode不为1时，与subStatusCode对应*/
  "errorMsg": "ok"
  /*ro, req, string, 错误信息, desc:对errorCode的解释,当statusCode不为1时，解释信息在协议约束中，允许设备在后续的版本迭代中，进行优化丰富提升（不限制死）*/
}
```

69.47.30 射频卡识别启用配置能力

Request URL

GET /ISAPI/AccessControl/Configuration/RFCardCfg/capabilities?format=json

查询参数

无

请求报文

无

响应报文

```

{
  "RFCardCfgCap": {
    /*ro, req, object, 射频卡识别启用配置能力*/
    "cardType": {
      /*ro, opt, object, 射频卡类型, desc:EMCard--EM卡, M1Card--M1卡, CPUCard--CPU卡, IDCard--身份证*/
      "@opt": ["EMCard", "M1Card", "CPUCard", "IDCard", "FelicaCard"]
      /*ro, req, array, 可选项, subType:string*/
    },
    "enabled": {
      /*ro, opt, object, 使能*/
      "@opt": [true, false]
      /*ro, req, array, 可选项, subType:bool*/
    }
  }
}

```

69.47.31 配置卡号认证规则

Request URL

PUT /ISAPI/AccessControl/CardVerificationRule?format=json

查询参数

无

请求报文

```

{
  "CardVerificationRule": {
    /*req, object, 卡号认证模式配置*/
    "cardNoLenMode": "full",
    /*req, enum, 卡号认证长度模式, subType:string, [full#full,3Bytes#3Bytes(wiegand26),4Bytes#4Bytes(wiegand34),wiegand27#韦根27,wiegand35#韦根35,Corporate1000_35#韦根Corporate 1000 35bits,Corporate1000_48#韦根Corporate 1000 48bits,H10302_37#韦根H10302 37bits,H10304_37#韦根H10304 37bits,wiegand_26CSN#韦根26 CSN,H103130_32CSN#韦根H103130 32bits CSN,wiegand_56CSN#韦根56 CSN,wiegand_58#韦根58], desc:切换后设备需要检验卡号兼容性情况*/
    "reverseCardNoEnabled": true
    /*opt, bool, 卡号翻转使能, desc:将主机中的卡号做翻转, 以适配不同厂商的翻转规则*/
  }
}

```

响应报文

```

{
  "statusCode": 1,
  /*ro, opt, int, 状态码, desc:无法用1表示时(1表示成功且无特殊状态)必须返回*/
  "statusString": "OK",
  /*ro, opt, string, 状态描述, range:[1,64], desc:无法用OK表示时(OK表示成功且无特殊状态)必须返回*/
  "subStatusCode": "ok",
  /*ro, opt, string, 子状态码, range:[1,64], desc:无法用ok表示(ok表示成功且无特殊状态)必须返回*/
  "errorCode": 1,
  /*ro, opt, int, 错误码, desc:当statusCode不为1时,与subStatusCode对应*/
  "errorMsg": "ok"
  /*ro, opt, string, 错误信息, desc:当statusCode不为1时,必须返回,解释信息在协议约束中,允许设备在后续的版本迭代中,进行优化丰富提升(不限制死)*/
}

```

69.47.32 获取卡号认证规则

Request URL

GET /ISAPI/AccessControl/CardVerificationRule?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "CardVerificationRule": {
    /*ro, req, object, 卡号认证模式配置*/
    "cardNoLenMode": "full",
    /*ro, req, enum, 卡号认证长度模式, subType:string, [full#full,3Bytes#3Bytes (wiegand26),4Bytes#4Bytes (wiegand34),wiegand27#韦根27,wiegand35#韦根
35,Corporate1000_35#韦根Corporate 1000 35bits,Corporate1000_48#韦根Corporate 1000 48bits,H10302_37#韦根H10302 37bits,H10304_37#韦根H10304
37bits,wiegand_26CSN#韦根26 CSN,H103130_32CSN#韦根H103130 32bits CSN,wiegand_56CSN#韦根56 CSN,wiegand_58#韦根58], desc:切换后设备需要检验卡号兼容性情况*/
    "reverseCardNoEnabled": true
    /*ro, opt, bool, 卡号翻转使能, desc:将主机中的卡号做翻转, 以适应不同厂商的翻转规则*/
  }
}
```

69.47.33 获取卡号认证规则配置进度

Request URL

GET /ISAPI/AccessControl/CardVerificationRule/progress?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "CardVerificationRuleRes": {
    /*ro, req, object, 卡号认证模式配置*/
    "checkStatus": "continue",
    /*ro, opt, enum, 配置结果状态, subType:string, [continue#需要稍后查询切换结果,ok#切换成功,duplicate#存在重复数据,failed#操作失败]*/
    "progress": 100
    /*ro, opt, int, 切换进度百分比, range:[0,100], desc:100表示完成*/
  }
}
```

69.48 权限认证语音播报

69.48.1 配置语音播报日计划参数

Request URL

PUT /ISAPI/AccessControl/Verification/ttsText?format=json

查询参数

无

请求报文

```

{
  "TTSText": {
    /*req, object, 语音播报文本*/
    "enable": true,
    /*req, bool, 使能, desc:true-使能配置, false-不使能配置*/
    "prefix": "name",
    /*opt, enum, 前缀, subType:string, [name#姓名（播报用户姓名）,lastname#姓+称呼（播报用户姓加称呼）,none#无], desc:默认无, 填空字符串也表示无*/
    "temperatureBroadcastEnabled": false,
    /*opt, bool, 温度播报使能,
    desc:默认为false
    开启后, 播报顺序为: 前缀+温度+认证结果*/
    "Success": [
    /*opt, array, 成功, subType:object, range:[1,8]*/
    {
      "TimeSegment": {
        /*opt, object, 时间段, desc:默认整天*/
        "beginTime": "00:00:00",
        /*req, string, 开始时间点, range:[1,32], desc:00:00:00~24:00:00*/
        "endTime": "00:00:00"
        /*req, string, 结束时间点, range:[1,32], desc:00:00:00~24:00:00*/
      },
      "language": "SimChinese",
      /*opt, enum, 语言, subType:string, [SimChinese#简体中文,TraChinese#繁体中文,English#英文], desc:默认简体中文*/
      "audioSourceType": "TTS",
      /*opt, enum, 音频源来下, subType:string, [TTS#TTS语音,customFile#自定义语音],
      desc:字段不存在时, 表示配置TTS语音, text字段为必填
      字段存在且为customFile时, text字段为可选*/
      "text": "test",
      /*req, string, 文本, range:[1,32], dep:and,{$.TTSText.Success[*].audioSourceType,eq,TTS}, desc:audioSourceType为customFile时, 该字段可选*/
      "customFileID": 1
      /*opt, int, 自定义音频文件编号, dep:or,{$.TTSText.Success[*].audioSourceType,eq,customFile}*/
    }
  ],
  "Failure": [
    /*opt, array, 失败, subType:object*/
    {
      "TimeSegment": {
        /*opt, object, 时间段, desc:默认整天*/
        "beginTime": "00:00:00",
        /*req, string, 开始时间点, range:[1,32], desc:00:00:00~24:00:00*/
        "endTime": "00:00:00"
        /*req, string, 结束时间点, range:[1,32], desc:00:00:00~24:00:00*/
      },
      "language": "SimChinese",
      /*opt, enum, 语言, subType:string, [SimChinese#简体中文,TraChinese#繁体中文,English#英文], desc:默认简体中文*/
      "audioSourceType": "TTS",
      /*opt, enum, 音频源类型, subType:string, [TTS#TTS语音,customFile#自定义语音],
      desc:字段不存在时, 表示配置TTS语音, text字段为必填
      字段存在且为customFile时, text字段为可选*/
      "text": "test",
      /*req, string, 文本, range:[1,32], dep:and,{$.TTSText.Success[*].audioSourceType,eq,TTS}, desc:audioSourceType为customFile时, 该字段可选*/
      "customFileID": 1
      /*opt, int, 自定义音频文件编号, dep:or,{$.TTSText.Success[*].audioSourceType,eq,customFile}*/
    }
  ]
}
}

```

响应报文

```

{
  "statusCode": 1,
  /*ro, opt, int, 状态码, desc:无法用1表示时（1表示成功且无特殊状态）必须返回*/
  "statusString": "OK",
  /*ro, opt, string, 状态描述, range:[1,64], desc:无法用OK表示时（OK表示成功且无特殊状态）必须返回*/
  "subStatusCode": "ok",
  /*ro, opt, string, 子状态码, range:[1,64], desc:无法用ok表示（ok表示成功且无特殊状态）必须返回*/
  "errorCode": 1,
  /*ro, opt, int, 错误码, desc:当statusCode不为1时, 与subStatusCode对应*/
  "errorMsg": "ok"
  /*ro, opt, string, 错误信息, desc:当statusCode不为1时, 必须返回,解释信息在协议约束中, 允许设备在后续的版本迭代中, 进行优化丰富提升（不限制死）*/
}

```

69.48.2 获取语音播报日计划参数

Request URL

GET /ISAPI/AccessControl/Verification/ttsText?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "TTSText": {
    /*ro, req, object, 语音播报文本*/
    "enable": true,
    /*ro, req, bool, 使能, desc:true-使能配置, false-不使能配置*/
    "prefix": "name",
    /*ro, opt, enum, 前缀, subType:string, [name#姓名（播报用户姓名）,lastname#姓+称呼（播报用户姓加称呼）,none#无], desc:默认无, 填空字符串也表示无*/
    "temperatureBroadcastEnabled": false,
    /*ro, opt, bool, 温度播报使能,
    desc:默认为false
    开启后, 播报顺序为: 前缀+温度+认证结果*/
    "Success": [
    /*ro, opt, array, 成功, subType:object, range:[1,8]*/
      {
        "TimeSegment": {
          /*ro, opt, object, 时间段, desc:默认整天*/
          "beginTime": "00:00:00",
          /*ro, req, string, 开始时间点, range:[1,32], desc:00:00:00~24:00:00*/
          "endTime": "00:00:00",
          /*ro, req, string, 结束时间点, range:[1,32], desc:00:00:00~24:00:00*/
        },
        "language": "SimChinese",
        /*ro, opt, enum, 语言, subType:string, [SimChinese#简体中文,TraChinese#繁体中文,English#英文], desc:默认简体中文*/
        "audioSourceType": "TTS",
        /*ro, opt, enum, 音频源类型, subType:string, [TTS#TTS语音,customFile#自定义语音],
        desc:字段不存在时, 表示配置TTS语音, text字段为必填
        字段存在且为customFile时, text字段为可选*/
        "text": "test",
        /*ro, req, string, 文本, range:[1,32], dep:and,{$.TTSText.Success[*].audioSourceType,eq,TTS}, desc:audioSourceType为customFile时, 该字段可选

        "customFileID": 1
        /*ro, opt, int, 自定义音频文件变红, dep:or,{$.TTSText.Success[*].audioSourceType,eq,customFile}*/
      }
    ],
    "Failure": [
    /*ro, opt, array, 失败, subType:object*/
      {
        "TimeSegment": {
          /*ro, opt, object, 时间段, desc:默认整天*/
          "beginTime": "00:00:00",
          /*ro, req, string, 开始时间点, range:[1,32], desc:00:00:00~24:00:00*/
          "endTime": "00:00:00",
          /*ro, req, string, 结束时间点, range:[1,32], desc:00:00:00~24:00:00*/
        },
        "language": "SimChinese",
        /*ro, opt, enum, 语言, subType:string, [SimChinese#简体中文,TraChinese#繁体中文,English#英文], desc:默认简体中文*/
        "audioSourceType": "TTS",
        /*ro, opt, enum, 音频源类型, subType:string, [TTS#TTS语音,customFile#自定义语音],
        desc:字段不存在时, 表示配置TTS语音, text字段为必填
        字段存在且为customFile时, text字段为可选*/
        "text": "test",
        /*ro, req, string, 文本, range:[1,32], dep:and,{$.TTSText.Success[*].audioSourceType,eq,TTS}, desc:audioSourceType为customFile时, 该字段可选

        "customFileID": 1
        /*ro, opt, int, 自定义音频文件变红, dep:or,{$.TTSText.Success[*].audioSourceType,eq,customFile}*/
      }
    ]
  }
}
```

69.48.3 获取语音播报日计划参数能力

Request URL

GET /ISAPI/AccessControl/Verification/ttsText/capabilities?format=json

查询参数

无

请求报文

无

响应报文

```

{
  "TTSTextCap": {
    /*ro, req, object, 语音播报文本配置能力*/
    "enable": [true, false],
    /*ro, req, array, 使能, subType:bool, desc:true-使能, false-不使能*/
    "prefix": ["name", "lastname", "none"],
    /*ro, opt, array, 前缀, subType:string, desc:name-姓名（播报用户姓名） lastname-姓+称呼（播报用户姓加称呼）， none-无，默认无，填空字符串也表示无*/
    "temperatureBroadcastEnabled": {
      /*ro, opt, object, 温度播报使能*/
      "@opt": [true, false]
      /*ro, opt, array, 可选项, subType:bool*/
    },
    "Success": {
      /*ro, opt, object, 成功*/
      "maxSize": 4,
      /*ro, opt, int, 最大次数, range:[1,8]*/
      "TimeSegment": {
        /*ro, opt, object, 时间段*/
        "beginTime": "00:00:00",
        /*ro, opt, time, 开始时间点, desc:00:00:00~24:00:00*/
        "endTime": "00:00:00",
        /*ro, opt, string, 结束时间点, desc:00:00:00~24:00:00*/
        "validUnit": "minute"
        /*ro, opt, enum, 时间点精度, subType:string, [hour#小时,minute#分钟,second#秒], desc:不返回该字段，则代表精度为分钟*/
      },
      "language": {
        /*ro, opt, object, 语言*/
        "@opt": ["SimChinese", "TraChinese", "English"]
        /*ro, opt, array, 可选项, subType:string*/
      },
      "audioSourceType": {
        /*ro, opt, object, 音频源类型*/
        "@opt": ["TTS", "customFile"]
        /*ro, opt, array, 取值范围, subType:string*/
      },
      "text": {
        /*ro, req, object, 文本*/
        "@min": 1,
        /*ro, opt, int, 取值范围, range:[1,128]*/
        "@max": 128
        /*ro, opt, int, 取值范围, range:[1,128]*/
      },
      "customFileID": {
        /*ro, opt, object, 自定义音频文件编号*/
        "@min": 1,
        /*ro, opt, int, 取值范围*/
        "@max": 128
        /*ro, opt, int, 取值范围*/
      }
    },
    "Failure": {
      /*ro, opt, object, 失败*/
      "maxSize": 4,
      /*ro, opt, int, 最大次数, range:[1,8]*/
      "TimeSegment": {
        /*ro, opt, object, 时间段*/
        "beginTime": "00:00:00",
        /*ro, opt, time, 开始时间点, desc:00:00:00~24:00:00*/
        "endTime": "00:00:00",
        /*ro, opt, string, 结束时间点, desc:00:00:00~24:00:00*/
        "validUnit": "minute"
        /*ro, opt, enum, 时间点精度, subType:string, [hour#小时,minute#分钟,second#秒], desc:不返回该字段，则代表精度为分钟*/
      },
      "language": {
        /*ro, opt, object, 语言*/
        "@opt": ["SimChinese", "TraChinese", "English"]
        /*ro, opt, array, 可选项, subType:string*/
      },
      "audioSourceType": {
        /*ro, opt, object, 音频源类型*/
        "@opt": ["TTS", "customFile"]
        /*ro, opt, array, 取值范围, subType:string*/
      },
      "text": {
        /*ro, req, object, 文本*/
        "@min": 1,
        /*ro, opt, int, 取值范围, range:[1,128]*/
        "@max": 128
        /*ro, opt, int, 取值范围, range:[1,128]*/
      },
      "customFileID": {
        /*ro, opt, object, 自定义音频文件编号*/
        "@min": 1,
        /*ro, opt, int, 取值范围*/
        "@max": 128
        /*ro, opt, int, 取值范围*/
      }
    }
  }
}
}

```

69.49 访问控制安全帽检测

69.49.1 配置安全帽参数

Request URL

PUT /ISAPI/AccessControl/Configuration/safetyHelmetDetection?format=json

查询参数

无

请求报文

```
{
  "SafetyHelmetDetection": {
    /*req, object, 安全帽检测*/
    "enable": true,
    /*opt, bool, 是否开启安全帽检测, desc:默认false*/
    "noHelmetStrategy": "normal"
    /*opt, enum, 未带安全帽是否开门, subType:string, [normal#提示且开门（默认）, forbidden#提示且不开门, noTipsAndOpenDoor#无提示且开门]*/
  }
}
```

响应报文

```
{
  "statusCode": 1,
  /*ro, opt, int, 状态码, desc:无法用1表示时（1表示成功且无特殊状态）必须返回*/
  "statusString": "OK",
  /*ro, opt, string, 状态描述, range:[1,64], desc:无法用OK表示时（OK表示成功且无特殊状态）必须返回*/
  "subStatusCode": "ok",
  /*ro, opt, string, 子状态码, range:[1,64], desc:无法用ok表示（ok表示成功且无特殊状态）必须返回*/
  "errorCode": 1,
  /*ro, opt, int, 错误码, desc:当statusCode不为1时，与subStatusCode对应*/
  "errorMsg": "ok"
  /*ro, opt, string, 错误信息, desc:当statusCode不为1时，必须返回,解释信息在协议约束中，允许设备在后续的版本迭代中，进行优化丰富提升（不限制死）*/
}
```

69.49.2 获取安全帽参数

Request URL

GET /ISAPI/AccessControl/Configuration/safetyHelmetDetection?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "SafetyHelmetDetection": {
    /*ro, req, object, 安全帽检测*/
    "enable": true,
    /*ro, opt, bool, 是否开启安全帽检测, desc:默认false*/
    "noHelmetStrategy": "normal"
    /*ro, opt, enum, 未带安全帽是否开门, subType:string, [normal#提示且开门（默认）, forbidden#提示且不开门, noTipsAndOpenDoor#无提示且开门]*/
  }
}
```

69.49.3 获取安全帽参数能力

Request URL

GET /ISAPI/AccessControl/Configuration/safetyHelmetDetection/capabilities?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "SafetyHelmetDetectionCap": {
    /*ro, req, object, 安全帽检测能力*/
    "enable": {
      /*ro, opt, object, 是否开启安全帽检测*/
      "@opt": [true, false]
      /*ro, opt, array, 可选项, subType:bool*/
    },
    "noHelmetStrategy": {
      /*ro, opt, object, 未带安全帽是否开门, desc:normal#提示且开门（默认）,forbidden#提示且不开门,noTipsAndOpenDoor#无提示且开门*/
      "@opt": ["normal", "forbidden", "noTipsAndOpenDoor"]
      /*ro, opt, array, 可选项, subType:string*/
    }
  }
}
```

69.50 温度检测

69.50.1 配置测温参数

Request URL

PUT /ISAPI/AccessControl/temperatureMeasureCfg?format=json

查询参数

无

请求报文

```
{
  "showTemperatureInfo": true,
  /*opt, bool, 是否显示温度信息, desc:true-显示, false-不显示*/
  "saveThermalPicture": true,
  /*opt, bool, 是否保存热成像图片, desc:true-保存, false-不保存*/
  "uploadThermalPicture": true,
  /*opt, bool, 是否上传热成像图片, desc:true-上传, false-不上传*/
  "lowTemperatureEnabled": true,
  /*opt, bool, 低温环境测温使能,
  desc:1. 该字段用于现场低温环境下控制测温结果的范围, 增强用户体验。
  2. 低温环境配置开启, 低于36度的人脸温度会映射到36度以上; 低温环境配置关闭后, 低于36度的人脸温度不会映射。*/
  "highTemperatureEnabled": true,
  /*opt, bool, 高温环境测温使能, desc:解决测温功能在温度偏高时有误差, 设备在测温结果高于特定阈值时进行映射转换, 称为高温模式。*/
  "showTemperatureMeasureInterface": true,
  /*opt, bool, 是否显示设备本地测温界面, desc:true-显示, false-不显示*/
  "distance": 0.3,
  /*opt, float, 人脸测温距离, range:[0.3,2.0], unit:m, unitType:长度, desc:精确到小数点后1位*/
  "temperatureMeasureMode": "auto"
  /*opt, enum, 测温模式, subType:string, [auto#自动,manual#手动], desc:auto模式下, 设备测温功能一直开启, 处于持续测温状态; manual模式下, 手动触发一次, 设备测温1次。*/
}
```

响应报文

```
{
  "statusCode": 1,
  /*ro, opt, int, 状态码, desc:无法用1表示时（1表示成功且无特殊状态）必须返回*/
  "statusString": "OK",
  /*ro, opt, string, 状态描述, range:[1,64], desc:无法用OK表示时（OK表示成功且无特殊状态）必须返回*/
  "subStatusCode": "ok",
  /*ro, opt, string, 子状态码, range:[1,64], desc:无法用ok表示（ok表示成功且无特殊状态）必须返回*/
  "errorCode": 1,
  /*ro, opt, int, 错误码, desc:当statusCode不为1时, 与subStatusCode对应*/
  "errorMsg": "ok"
  /*ro, opt, string, 错误信息, desc:当statusCode不为1时, 必须返回,解释信息在协议约束中, 允许设备在后续的版本迭代中, 进行优化丰富提升（不限制死）*/
}
```

69.50.2 获取测温参数

Request URL

GET /ISAPI/AccessControl/temperatureMeasureCfg?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "showTemperatureInfo": true,
  /*ro, opt, bool, 是否显示温度信息, desc:true-显示, false-不显示*/
  "saveThermalPicture": true,
  /*ro, opt, bool, 是否保存热成像图片, desc:true-保存, false-不保存*/
  "uploadThermalPicture": true,
  /*ro, opt, bool, 是否上传热成像图片, desc:true-上传, false-不上传*/
  "lowTemperatureEnabled": true,
  /*ro, opt, bool, 低温环境测温使能,
  desc:1、该字段用于现场低温环境下控制测温结果的范围, 增强用户体验。
  2、低温环境配置开启, 低于36度的人脸温度会映射到36度以上; 低温环境配置关闭后, 低于36度的人脸温度不会映射。*/
  "highTemperatureEnabled": true,
  /*ro, opt, bool, 高温环境测温使能, desc:解决测温功能在温度偏高时有误差, 设备在测温结果高于特定阈值时进行映射转换, 称为高温模式。*/
  "showTemperatureMeasureInterface": true,
  /*ro, opt, bool, 是否显示设备本地测温界面, desc:true-显示, false-不显示*/
  "distance": 0.3,
  /*ro, opt, float, 人脸测温距离, range:[0.3,2.0], unit:m, unitType:长度, desc:精确到小数点后1位*/
  "temperatureMeasureMode": "auto",
  /*ro, opt, enum, 测温模式, subType:string, [auto#自动,manual#手动], desc:auto模式下, 设备测温功能一直开启, 处于持续测温状态; manual模式下, 手动触发一次, 设备测温1次。*/
  "thermometryModuleStatus": "online"
  /*ro, opt, enum, 测温模组状态, subType:string, [online#在线,offline#离线], desc:只读属性*/
}
```

69.50.3 获取测温参数配置能力

Request URL

GET /ISAPI/AccessControl/temperatureMeasureCfg/capabilities?format=json

查询参数

无

请求报文

无

响应报文

```

{
  "showTemperatureInfo": {
    /*ro, opt, object, 是否显示温度信息, desc:true-显示, false-不显示*/
    "@opt": [true, false]
    /*ro, req, array, 可选项, subType:bool*/
  },
  "saveThermalPicture": {
    /*ro, opt, object, 是否保存热成像图片, desc:true-保存, false-不保存*/
    "@opt": [true, false]
    /*ro, req, array, 可选项, subType:bool*/
  },
  "uploadThermalPicture": {
    /*ro, opt, object, 是否上传热成像图片, desc:true-上传, false-不上传*/
    "@opt": [true, false]
    /*ro, req, array, 可选项, subType:bool*/
  },
  "lowTemperatureEnabled": {
    /*ro, opt, object, 低温环境测温使能*/
    "@opt": [true, false]
    /*ro, req, array, 可选项, subType:bool, desc:true-开启, false-关闭*/
  },
  "highTemperatureEnabled": {
    /*ro, opt, object, 高温环境测温使能*/
    "@opt": [true, false]
    /*ro, req, array, 可选项, subType:bool*/
  },
  "showTemperatureMeasureInterface": {
    /*ro, opt, object, 是否显示设备本地测温界面*/
    "@opt": [true, false]
    /*ro, req, array, 可选项, subType:bool, desc:true-显示, false-不显示*/
  },
  "distance": {
    /*ro, opt, object, 人脸测温距离,
    desc:精确到小数点后1位, 单位 米
    安卓访客机项目, 人证比对后自动进行测温*/
    "@min": 0.3,
    /*ro, req, float, 最小值*/
    "@max": 2.0
    /*ro, req, float, 最大值*/
  },
  "temperatureMeasureMode": {
    /*ro, opt, object, 测温模式*/
    "@opt": ["auto", "manual"]
    /*ro, opt, array, 可选项, subType:string, desc:auto-自动, manual-手动。*/
  },
  "thermometryModuleStatus": {
    /*ro, opt, object, 测温模块状态*/
    "@opt": ["online", "offline"]
    /*ro, req, array, 取值范围, subType:string*/
  }
}

```

69.50.4 人脸测温事件

事件标识: **FaceTemperatureMeasurementEvent**

```

{
  "ipAddress": "172.6.64.7",
  /*ro, req, string, 报警设备IPv4地址*/
  "ipv6Address": "1080:0:0:0:8:800:200C:417A",
  /*ro, opt, string, 报警设备IPv6地址*/
  "portNo": 80,
  /*ro, opt, int, 报警设备端口号*/
  "protocol": "HTTP",
  /*ro, opt, string, 协议类型, desc:HTTP或HTTPS*/
  "macAddress": "01:17:24:45:D9:F4",
  /*ro, opt, string, MAC地址*/
  "channelID": 1,
  /*ro, opt, int, 触发报警的设备通道号*/
  "dateTime": "2016-12-12T17:30:08+08:00",
  /*ro, req, string, 报警触发时间*/
  "activePostCount": 1,
  /*ro, req, int, 同一个报警已经上传的次数*/
  "eventType": "FaceTemperatureMeasurementEvent",
  /*ro, req, string, 触发的事件类型, desc:FaceTemperatureMeasurementEvent-人脸测温事件*/
  "eventState": "active",
  /*ro, req, enum, 事件触发状态, subType:string, [active#触发,inactive#未触发]*/
  "eventDescription": "Face Temperature Measurement Event",
  /*ro, req, string, 事件描述, desc:人脸测温事件(Face Temperature Measurement Event)*/
  "deviceId": "test0123",
  /*ro, opt, string, 即PUID, desc:在ISUP协议接入透传ISAPI事件信息中必须返回*/
  "FaceTemperatureMeasurementEvent": {
    /*ro, req, object, FaceTemperatureMeasurementEvent*/
    "deviceName": "test设备",
    /*ro, opt, string, 设备名称*/
    "serialNo": 1,
    /*ro, opt, int, 事件流水号*/
    "currentEvent": true,
    /*ro, opt, bool, 是否为实时事件, desc:true-是(实时事件), false-否(离线事件)*/
    "thermometryUnit": "celsius",
    /*ro, req, enum, 测温单位, subType:string, [celsius#摄氏度,fahrenheit#华氏度,kelvin#开尔文], desc:默认摄氏度*/
    "currTemperature": 36.1,
    /*ro, req, float, 人脸温度, desc:精确到小数点后一位*/
    "isAbnormalTemperature": true,
    /*ro, opt, bool, 人脸测温是否温度异常, desc:true-是, false-否*/
    "RegionCoordinates": {
      /*ro, opt, object, 人脸温度坐标*/
      "positionX": 1,
      /*ro, opt, int, X坐标, range:[0,10000], desc:归一化坐标0-1000*/
      "positionY": 1,
      /*ro, opt, int, Y坐标, desc:归一化坐标0-1000*/
    },
    "remoteCheck": false,
    /*ro, opt, bool, 是否需要远程校验, desc:true-需要, false-不需要(默认)*/
    "mask": "unknown",
    /*ro, opt, enum, 是否戴口罩, subType:string, [unknown#未知,yes#戴口罩,no#不戴口罩]*/
    "visibleLightURL": "xxxxxxxx",
    /*ro, opt, string, 热成像相机可见光图片URL*/
    "thermalURL": "xxxxxxxx",
    /*ro, opt, string, 热成像图片URL*/
    "pictureURL": "test",
    /*ro, opt, string, 抓拍图片URL*/
    "helmet": "unknown",
    /*ro, opt, enum, 是否戴安全帽, subType:string, [unknown#未知,yes#戴安全帽,no#不戴安全帽]*/
    "visibleLightContentID": "visible_light_image",
    /*ro, opt, string, 热成像相机可见光图片ID(与报文中Content-ID字段一致)*/
    "thermalContentID": "thermal_image",
    /*ro, opt, string, 热成像相机可见光图片ID(与报文中Content-ID字段一致)*/
    "pictureContentID": "picture_image",
    /*ro, opt, string, 抓拍图片ID(与报文中Content-ID字段一致)*/
    "FaceRect": {
      /*ro, opt, object, 人脸矩形框, desc:屏幕左上角为坐标原点*/
      "height": 1.000,
      /*ro, req, float, 高度, range:[0.000,1.000]*/
      "width": 1.000,
      /*ro, req, float, 宽度, range:[0.000,1.000]*/
      "x": 0.000,
      /*ro, req, float, 区域左上角顶点的水平坐标, range:[0.000,1.000]*/
      "y": 0.000,
      /*ro, req, float, 区域左上角顶点的垂直坐标, range:[0.000,1.000]*/
    },
    "UUID": "test"
  },
  /*ro, opt, string, 事件的UUID, range:[0,64], desc:用于在HEOP接口中删除对应事件的消息记录*/
}
}

```

参数名称	参数值	参数类型 (Content-Type)	Content-ID	文件名称 (filename)	描述
FaceTemperatureMeasurementEvent	[报文内容]	application/json	--	--	--
VisibleLight	[图片二进制数据]	image/jpeg	visibleLight_image	VisibleLight.jpg	热成像相机可见光图片数据
Thermal	[图片二进制数据]	image/jpeg	thermal_image	Thermal.jpg	热成像图片数据
Picture	[图片二进制数据]	image/jpeg	picture_image	Picture.jpg	抓拍图片数据

说明： 此协议通过表单格式传输，表单框架详细说明见章节4.5.1.4，如下实例：

```
--<frontier>
Content-Disposition: form-data; name=参数名称;filename=文件名称
Content-Type: 参数类型
Content-Length: ****
Content-ID: 内容ID
参数值
```

- 参数名称：表单单元Headers中Content-Disposition的name属性，表示此表单单元的名称
- 参数类型（Content-Type）： 表单单元Headers中Content-Type属性
- 文件名称（filename）： 表单单元Headers中Content-Disposition的filename属性，仅在本表单单元传输数据为文件类型存在，表示表单单元Body的文件名
- 参数值： 表单单元的body内容

69.51 远程控门

69.51.1 远程控门

Request URL

PUT /ISAPI/AccessControl/RemoteControl/door/<doorID>

查询参数

参数名称	参数类型	描述
doorID	string	门编号，65535代表所有门

请求报文

```
<?xml version="1.0" encoding="UTF-8"?>

<RemoteControlDoor xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, 远程控门, attr:version{req, string, 协议版本}-->
  <cmd>
    <!--req, enum, 命令, subType:string, [open#开门,close#关门（受控）,alwaysOpen#常开（自由）,alwaysClose#常关（禁用）,visitorCallLadder#访客呼梯,householdCallLadder#住户呼梯,resume#恢复普通]-->open
  </cmd>
  <password>
    <!--opt, string, 开门密码, range:[8,16], desc:该字段仅在萤石链路下生效，萤石链路下，password与employeeNo至少下发其中一个，如果同时下发，以employeeNo为准，不对password进行校验-->test
  </password>
  <employeeNo>
    <!--no, opt, string, 工号, range:[0,32], desc:该字段仅在萤石链路下生效，萤石链路下，password与employeeNo至少下发其中一个，如果同时下发，以employeeNo为准，不对password进行校验-->test
  </employeeNo>
  <channelNo>
    <!--opt, int, 设备编号(通道号), range:[0,256], step:1-->1
  </channelNo>
  <controlType>
    <!--opt, enum, 开锁场景类型, subType:string, [monitor#监视,calling#通话]-->monitor
  </controlType>
</RemoteControlDoor>
```

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 响应消息, attr:version{ro, req, string, 版本}-->
  <requestURL>
    <!--ro, req, string, 请求的URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, 状态码, subType:int, [0#OK,1#OK,2#Device Busy,3#Device Error,4#Invalid Operation,5#Invalid XML Format,6#Invalid XML Content,7#Reboot Required]-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, 状态信息, subType:string, [OK#成功,Device Busy#设备忙,Device Error#设备异常,Invalid Operation#无效的操作,Invalid XML Format#无效的XML格式,Invalid XML Content#无效的XML内容,Reboot#设备重启]-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, 详细错误码英文描述, desc:详细错误码的英文描述-->OK
  </subStatusCode>
</ResponseStatus>
```

69.51.2 获取远程控门能力集

Request URL

GET /ISAPI/AccessControl/RemoteControl/door/capabilities

查询参数

无

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<RemoteControlDoor xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 远程控门, attr:version{req, float, 协议版本}-->
  <doorNo min="1" max="10">
    <!--ro, opt, int, 门编号范围, attr:min{req, int, 最小值},max{req, int, 最大值}-->1
  </doorNo>
  <cmd opt="open,close,alwaysOpen,alwaysClose,visitorCallLadder,householdCallLadder,resume">
    <!--ro, req, enum, 命令, subType:string, [open#开门,close#关门(受控),alwaysOpen#常开(自由),alwaysClose#常关(禁用),visitorCallLadder#访客呼梯,householdCallLadder#住户呼梯,resume#恢复普通], attr:opt{req, string, 取值范围}-->open
  </cmd>
  <password min="8" max="16">
    <!--ro, opt, string, 开门密码, range:[8,16], attr:min{req, int, 最小值, range:[8,16]},max{req, int, 最大值, range:[8,16]}-->test
  </password>
  <employeeNo min="0" max="32">
    <!--ro, opt, string, 工号, range:[0,32], attr:min{req, int, 最小值, range:[0,32]},max{req, int, 最大值, range:[0,32]}-->test
  </employeeNo>
  <channelNo min="0" max="10">
    <!--ro, opt, int, 设备编号(通道号), attr:min{req, int, 最小值},max{req, int, 最大值}-->0
  </channelNo>
  <controlType opt="monitor,calling">
    <!--ro, opt, enum, 开锁场景类型, subType:string, [monitor#监视,calling#通话], attr:opt{req, string, 取值范围}-->monitor
  </controlType>
</RemoteControlDoor>
```

69.52 访问控制提示语管理(S)

69.52.1 配置自定义提示语

Request URL

PUT /ISAPI/AccessControl/customPrompt?format=json

查询参数

无

请求报文

```

{
  "enabled": true,
  /*req, bool, 自定义提示语使能*/
  "PromptList": [
    /*opt, array, 提示语列表, subType:object, range:[0,3], desc:若对应的提示语类型未下发,则表示不修改该类型的提示语*/
    {
      "promptType": "stranger",
      /*req, enum, 提示语类型, subType:string, [stranger#陌生人,authenticationSuccess#认证成功,authenticationFailed#认证失败]*/
      "promptContent": "test"
      /*opt, string, 提示语内容, range:[1,16]*/
    }
  ]
}

```

响应报文

```

{
  "statusCode": 1,
  /*ro, opt, int, 状态码, desc:无法用1表示时（1表示成功且无特殊状态）必须返回*/
  "statusString": "OK",
  /*ro, opt, string, 状态描述, range:[1,64], desc:无法用OK表示时（OK表示成功且无特殊状态）必须返回*/
  "subStatusCode": "ok",
  /*ro, opt, string, 子状态码, range:[1,64], desc:无法用ok表示（ok表示成功且无特殊状态）必须返回*/
  "errorCode": 1,
  /*ro, opt, int, 错误码, desc:当statusCode不为1时, 与subStatusCode对应*/
  "errorMsg": "ok"
  /*ro, opt, string, 错误信息, desc:当statusCode不为1时, 必须返回,解释信息在协议约束中, 允许设备在后续的版本迭代中, 进行优化丰富提升（不限制死）*/
}

```

69.52.2 获取自定义提示语

Request URL

GET /ISAPI/AccessControl/customPrompt?format=json

查询参数

无

请求报文

无

响应报文

```

{
  "enabled": true,
  /*ro, req, bool, 自定义提示语使能*/
  "PromptList": [
    /*ro, opt, array, 提示语列表, subType:object, range:[0,3], desc:若对应的提示语类型未下发,则表示不修改该类型的提示语*/
    {
      "promptType": "stranger",
      /*ro, req, enum, 提示语类型, subType:string, [stranger#陌生人,authenticationSuccess#认证成功,authenticationFailed#认证失败]*/
      "promptContent": "test"
      /*ro, opt, string, 提示语内容, range:[1,16]*/
    }
  ]
}

```

69.52.3 获取自定义提示语参数能力

Request URL

GET /ISAPI/AccessControl/customPrompt/capabilities?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "enabled": {
    /*ro, req, object, 自定义提示语使能*/
    "@opt": [true, false]
    /*ro, req, array, 取值范围, subType:bool*/
  },
  "PromptList": {
    /*ro, opt, object, 提示语列表*/
    "@size": 3,
    /*ro, req, int, 提示语列表的数组大小范围*/
    "promptType": {
      /*ro, req, object, 提示语类型*/
      "@opt": ["stranger", "authenticationSuccess", "authenticationFailed"]
      /*ro, req, array, 取值范围, subType:string*/
    },
    "promptContent": {
      /*ro, opt, object, 提示语内容*/
      "@min": 1,
      /*ro, req, int, 取值范围*/
      "@max": 16
      /*ro, req, int, 取值范围*/
    }
  }
}
```

69.53 人员门禁点权限计划管理

69.53.1 获取人员权限假日组参数配置能力

Request URL

GET /ISAPI/AccessControl/UserRightHolidayGroupCfg/capabilities?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "UserRightHolidayGroupCfg": {
    /*ro, req, object, 人员权限假日组参数配置*/
    "groupNo": {
      /*ro, opt, object, 假日组编号*/
      "@min": 1,
      /*ro, opt, int, 最小值*/
      "@max": 16
      /*ro, opt, int, 最大值*/
    },
    "enable": "true,false",
    /*ro, opt, string, 是否启用, desc:true-启用, false-不启用*/
    "groupName": {
      /*ro, opt, object, 假日组名称*/
      "@min": 1,
      /*ro, opt, int, 最小值*/
      "@max": 32
      /*ro, opt, int, 最大值*/
    },
    "holidayPlanNo": {
      /*ro, opt, object, 假日组计划编号*/
      "@min": 1,
      /*ro, opt, int, 最小值*/
      "@max": 16
      /*ro, opt, int, 最大值*/
    }
  }
}
```

69.53.2 获取人员权限计划模板参数配置能力

Request URL

GET /ISAPI/AccessControl/UserRightPlanTemplate/capabilities?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "UserRightPlanTemplate": {
    /*ro, opt, object, 人员权限计划模板*/
    "templateNo": {
      /*ro, opt, object, 计划模板编号*/
      "@min": 1,
      /*ro, opt, int, 最小值*/
      "@max": 16
      /*ro, opt, int, 最大值*/
    },
    "enable": "true,false",
    /*ro, opt, string, 是否启用, desc:true-启用, false-不启用*/
    "templateName": {
      /*ro, opt, object, 模板名称*/
      "@min": 1,
      /*ro, opt, int, 最小值*/
      "@max": 32
      /*ro, opt, int, 最大值*/
    },
    "weekPlanNo": {
      /*ro, opt, object, 周计划编号*/
      "@min": 1,
      /*ro, opt, int, 最小值*/
      "@max": 16
      /*ro, opt, int, 最大值*/
    },
    "holidayGroupNo": {
      /*ro, opt, object, 假日组编号*/
      "@min": 1,
      /*ro, opt, int, 最小值*/
      "@max": 16
      /*ro, opt, int, 最大值*/
    }
  }
}
```

69.53.3 获取人员权限周计划参数

Request URL

GET /ISAPI/AccessControl/UserRightWeekPlanCfg/<weekPlanID>?format=json

查询参数

参数名称	参数类型	描述
weekPlanID	string	周计划编号，从1开始，设备支持的最大值从能力集中获取

请求报文

无

响应报文

```
{
  "UserRightWeekPlanCfg": {
    /*ro, opt, object, 人员权限周计划参数配置*/
    "enable": true,
    /*ro, req, bool, 使能, desc:true-使能, false-不使能*/
    "WeekPlanCfg": [
      /*ro, req, array, 周计划参数, subType:object*/
      {
        "week": "Monday",
        /*ro, req, enum, 星期, subType:string, [Monday#星期一,Tuesday#星期二,Wednesday#星期三,Thursday#星期四,Friday#星期五,Saturday#星期六,Sunday#
星期日]*/
        "id": 1,
        /*ro, req, int, 时间段编号, range:[1,8], desc:范围1-8*/
        "enable": true,
        /*ro, req, bool, 使能, desc:true-使能, false-不使能*/
        "TimeSegment": {
          /*ro, req, object, 时间段*/
          "beginTime": "10:10:00",
          /*ro, req, string, 开始时间点, desc:设备本地时间*/
          "endTime": "12:10:00"
          /*ro, req, string, 结束时间点, desc:设备本地时间*/
        },
        "authenticationTimesEnabled": true,
        /*ro, opt, bool, 认证次数使能, desc:true-使能, false-不使能, 默认为false*/
        "authenticationTimes": 10
        /*ro, opt, int, 认证次数, range:[1,255], step:1, desc:处于当前有效期, 认证使能且认证次数未超, 则计划模板验证通过*/
      }
    ]
  }
}
```

69.53.4 获取人员权限周计划参数配置能力

Request URL

GET /ISAPI/AccessControl/UserRightWeekPlanCfg/capabilities?format=json

查询参数

无

请求报文

无

响应报文

```

{
  "UserRightWeekPlanCfg": {
    /*ro, opt, object, 人员权限周计划参数配置*/
    "planNo": {
      /*ro, opt, object, 周计划编号*/
      "@min": 1,
      /*ro, opt, int, 最小值*/
      "@max": 16
      /*ro, opt, int, 最大值*/
    },
    "enable": "true,false",
    /*ro, opt, string, 使能, desc:true-使能, false-不使能*/
    "WeekPlanCfg": {
      /*ro, opt, object, 周计划参数*/
      "maxSize": 56,
      /*ro, opt, int, 最大值*/
      "week": {
        /*ro, opt, object, 星期*/
        "@opt": "Monday"
        /*ro, opt, enum, 星期, subType:string, [Monday#星期一,Tuesday#星期二,Wednesday#星期三,Thursday#星期四,Friday#星期五,Saturday#星期六,Sunday#
星期日]*/
      },
      "id": {
        /*ro, opt, object, 编号*/
        "@min": 1,
        /*ro, opt, int, 最小值*/
        "@max": 8
        /*ro, opt, int, 最大值*/
      },
      "enable": "true,false",
      /*ro, opt, string, 使能, desc:true-使能, false-不使能*/
      "TimeSegment": {
        /*ro, opt, object, 时间段*/
        "beginTime": "test",
        /*ro, opt, string, 开始时间点, desc:设备本地时间*/
        "endTime": "test",
        /*ro, opt, string, 结束时间点, desc:设备本地时间*/
        "validUnit": "minute"
        /*ro, opt, enum, 时间点精度, subType:string, [hour#小时,minute#分钟,second#秒], desc:不返回该字段, 则代表精度为分钟*/
      },
      "authenticationTimesEnabled": {
        /*ro, opt, object, 认证次数使能*/
        "@opt": [true, false]
        /*ro, req, array, 可选项, subType:bool*/
      },
      "authenticationTimes": {
        /*ro, opt, object, 认证次数*/
        "@min": 1,
        /*ro, req, int, 最小值, range:[1,255], step:1*/
        "@max": 255
        /*ro, req, int, 最大值, range:[1,255], step:1*/
      }
    }
  }
}

```

69.53.5 获取人员权限假日计划参数配置能力

Request URL

GET /ISAPI/AccessControl/UserRightHolidayPlanCfg/capabilities?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "UserRightHolidayPlanCfg": {
    /*ro, req, object, 人员权限假日计划参数*/
    "planNo": {
      /*ro, opt, object, 假日计划编号*/
      "@min": 1,
      /*ro, opt, int, 最小值*/
      "@max": 16
      /*ro, opt, int, 最大值*/
    },
    "enable": "true,false",
    /*ro, opt, string, 使能, desc:true-使能, false-不使能*/
    "beginDate": "1970-01-01",
    /*ro, opt, date, 假日开始日期, desc:设备本地时间*/
    "endDate": "1970-01-01",
    /*ro, opt, date, 假日结束日期, desc:设备本地时间*/
    "HolidayPlanCfg": {
      /*ro, opt, object, 假日计划参数*/
      "maxSize": 8,
      /*ro, opt, int, 最大值*/
      "id": {
        /*ro, opt, object, 时间段编号*/
        "@min": 1,
        /*ro, opt, int, 最小值*/
        "@max": 8
        /*ro, opt, int, 最大值*/
      },
      "enable": "true,false",
      /*ro, opt, string, 使能, desc:true-使能, false-不使能*/
      "TimeSegment": {
        /*ro, opt, object, 时间段*/
        "beginTime": "00:00:00",
        /*ro, opt, time, 开始时间点, desc:设备本地时间*/
        "endTime": "00:00:00",
        /*ro, opt, time, 结束时间点, desc:设备本地时间*/
        "validUnit": "minute"
        /*ro, opt, enum, 时间点精度, subType:string, [hour#小时,minute#分钟,second#秒], desc:不返回该字段, 则代表精度为分钟*/
      },
      "authenticationTimesEnabled": {
        /*ro, opt, object, 认证次数使能*/
        "@opt": [true, false]
        /*ro, req, array, 可选项, subType:bool*/
      },
      "authenticationTimes": {
        /*ro, opt, object, 认证次数*/
        "@min": 1,
        /*ro, req, int, 最小值, range:[1,255], step:1*/
        "@max": 255
        /*ro, req, int, 最大值, range:[1,255], step:1*/
      }
    }
  }
}
```

69.53.6 获取人员权限假日计划参数

Request URL

GET /ISAPI/AccessControl/UserRightHolidayPlanCfg/<holidayPlanID>?format=json

查询参数

参数名称	参数类型	描述
holidayPlanID	string	假日计划编号，从1开始，设备支持的最大值从能力集中获取

请求报文

无

响应报文

```
{
  "UserRightHolidayPlanCfg": {
    /*ro, req, object, 人员权限假日计划参数配置*/
    "enable": true,
    /*ro, req, bool, 使能, desc:true-使能, false-不使能*/
    "beginDate": "1970-01-01",
    /*ro, req, date, 假日开始日期, desc:设备本地时间*/
    "endDate": "1970-01-01",
    /*ro, req, date, 假日结束日期, desc:设备本地时间*/
    "HolidayPlanCfg": [
      /*ro, req, array, 假日计划参数, subType:object*/
      {
        "id": 1,
        /*ro, req, int, 时间段编号, range:[1,8], desc:范围1-8*/
        "enable": true,
        /*ro, req, bool, 使能, desc:true-使能, false-不使能*/
        "TimeSegment": {
          /*ro, req, object, 时间段*/
          "beginTime": "00:00:00",
          /*ro, req, time, 开始时间点, desc:设备本地时间*/
          "endTime": "00:00:00"
          /*ro, req, time, 结束时间点, desc:设备本地时间*/
        },
        "authenticationTimesEnabled": true,
        /*ro, opt, bool, 认证次数使能, desc:true-使能, false-不使能, 默认为false*/
        "authenticationTimes": 10
        /*ro, opt, int, 认证次数, range:[1,255], step:1, desc:处于当前有效期, 认证使能且认证次数未超, 则计划模板验证通过*/
      }
    ]
  }
}
```

69.53.7 配置人员权限周计划参数

Request URL

PUT /ISAPI/AccessControl/UserRightWeekPlanCfg/<weekPlanID>?format=json

查询参数

参数名称	参数类型	描述
weekPlanID	string	周计划编号，从1开始，设备支持的最大值从能力集中获取

请求报文

```
{
  "UserRightWeekPlanCfg": {
    /*opt, object, 人员权限周计划参数配置*/
    "enable": true,
    /*req, bool, 使能, desc:true-使能, false-不使能*/
    "WeekPlanCfg": [
      /*req, array, 周计划参数, subType:object*/
      {
        "week": "Monday",
        /*req, enum, 星期, subType:string, [Monday#星期一,Tuesday#星期二,Wednesday#星期三,Thursday#星期四,Friday#星期五,Saturday#星期六,Sunday#星期日], desc:Monday-星期一, Tuesday-星期二, Wednesday-星期三, Thursday-星期四, Friday-星期五, Saturday-星期六, Sunday-星期日*/
        "id": 1,
        /*req, int, 时间段编号, range:[1,8], desc:范围1-8*/
        "enable": true,
        /*req, bool, 使能, desc:true-使能, false-不使能*/
        "TimeSegment": {
          /*req, object, 时间段*/
          "beginTime": "10:10:00",
          /*req, string, 开始时间点, desc:设备本地时间*/
          "endTime": "12:10:00"
          /*req, string, 结束时间点, desc:设备本地时间*/
        },
        "authenticationTimesEnabled": true,
        /*opt, bool, 认证次数使能, desc:true-使能, false-不使能, 默认为false*/
        "authenticationTimes": 10
        /*opt, int, 认证次数, range:[1,255], step:1, desc:处于当前有效期, 认证使能且认证次数未超, 则计划模板验证通过*/
      }
    ]
  }
}
```

响应报文

```
{
  "requestURL": "test",
  /*ro, opt, string, URI*/
  "statusCode": "test",
  /*ro, opt, string, 状态码, desc:无法用1表示时（1表示成功且无特殊状态），否则必须返回*/
  "statusString": "test",
  /*ro, opt, string, 状态描述, desc:无法用ok表示时（ok表示成功且无特殊状态），否则必须返回*/
  "subStatusCode": "test",
  /*ro, opt, string, 子状态码, desc:无法用ok表示（ok表示成功且无特殊状态），否则必须返回*/
  "errorCode": 1,
  /*ro, opt, int, 错误码, desc:当statusCode不为1时，与subStatusCode对应*/
  "errorMsg": "ok"
  /*ro, opt, string, 错误信息, desc:对errorCode的解释,当statusCode不为1时，解释信息在协议约束中，允许设备在后续的版本迭代中，进行优化丰富提升（不限制死）*/
}
```

69.53.8 配置人员权限假日计划参数

Request URL

PUT /ISAPI/AccessControl/UserRightHolidayPlanCfg/<holidayPlanID>?format=json

查询参数

参数名称	参数类型	描述
holidayPlanID	string	假日计划编号，从1开始，设备支持的最大值从能力集中获取

请求报文

```
{
  "UserRightHolidayPlanCfg": {
    /*req, object, 人员权限假日计划参数*/
    "enable": true,
    /*req, bool, 使能, desc:true-使能, false-不使能*/
    "beginDate": "1970-01-01",
    /*req, date, 假日开始日期, desc:设备本地时间*/
    "endDate": "1970-01-01",
    /*req, date, 假日结束日期, desc:设备本地时间*/
    "HolidayPlanCfg": [
      /*req, array, 假日计划参数, subType:object*/
      {
        "id": 1,
        /*req, int, 时间段编号, range:[1,8], desc:范围1-8*/
        "enable": true,
        /*req, bool, 使能, desc:true-使能, false-不使能*/
        "TimeSegment": {
          /*opt, object, 时间段*/
          "beginTime": "00:00:00",
          /*req, time, 开始时间点, desc:设备本地时间*/
          "endTime": "00:00:00",
          /*req, time, 结束时间点, desc:设备本地时间*/
        },
        "authenticationTimesEnabled": true,
        /*opt, bool, 认证次数使能, desc:true-使能, false-不使能, 默认为false*/
        "authenticationTimes": 10,
        /*opt, int, 认证次数, range:[1,255], step:1, desc:处于当前有效期，认证使能且认证次数未超，则计划模板验证通过*/
      }
    ]
  }
}
```

响应报文

```
{
  "requestURL": "test",
  /*ro, opt, string, URI*/
  "statusCode": "test",
  /*ro, opt, string, 状态码, desc:无法用1表示时（1表示成功且无特殊状态），否则必须返回*/
  "statusString": "test",
  /*ro, opt, string, 状态描述, desc:无法用ok表示时（ok表示成功且无特殊状态），否则必须返回*/
  "subStatusCode": "test",
  /*ro, opt, string, 子状态码, desc:无法用ok表示（ok表示成功且无特殊状态），否则必须返回*/
  "errorCode": 1,
  /*ro, opt, int, 错误码, desc:当statusCode不为1时，与subStatusCode对应*/
  "errorMsg": "ok"
  /*ro, opt, string, 错误信息, desc:对errorCode的解释,当statusCode不为1时，解释信息在协议约束中，允许设备在后续的版本迭代中，进行优化丰富提升（不限制死）*/
}
```

69.53.9 获取人员权限计划模板参数

Request URL

GET /ISAPI/AccessControl/UserRightPlanTemplate/<planTemplateID>?format=json

查询参数

参数名称	参数类型	描述
planTemplateID	string	计划模板编号，从1开始，设备支持的最大值从能力集中获取

请求报文

无

响应报文

```
{
  "UserRightPlanTemplate": {
    /*ro, req, object, 人员权限计划模板参数配置*/
    "enable": true,
    /*ro, req, bool, 是否启用, desc:true-启用, false-不启用*/
    "templateName": "test",
    /*ro, req, string, 模板名称*/
    "weekPlanNo": 1,
    /*ro, req, int, 周计划编号*/
    "holidayGroupNo": "1,3,5"
    /*ro, req, string, 假日组编号, desc:该字段配置可配置空值, 即"holidayPlanNo" : ""*/
  }
}
```

69.53.10 获取人员权限假日组参数

Request URL

GET /ISAPI/AccessControl/UserRightHolidayGroupCfg/<holidayGroupID>?format=json

查询参数

参数名称	参数类型	描述
holidayGroupID	string	从1开始，设备支持的最大值从能力集中获取

请求报文

无

响应报文

```
{
  "UserRightHolidayGroupCfg": {
    /*ro, req, object, 人员权限假日组参数配置*/
    "enable": true,
    /*ro, req, bool, 是否启用, desc:true-启用, false-不启用*/
    "groupName": "test",
    /*ro, req, string, 假日组名称*/
    "holidayPlanNo": "1,3,5"
    /*ro, req, string, 假日组计划编号, desc:该字段配置可配置空值, 即"holidayPlanNo" : ""*/
  }
}
```

69.53.11 配置人员权限假日组参数

Request URL

PUT /ISAPI/AccessControl/UserRightHolidayGroupCfg/<holidayGroupID>?format=json

查询参数

参数名称	参数类型	描述
holidayGroupID	string	假日组编号，从1开始，设备支持的最大值从能力集中获取

请求报文

```
{
  "UserRightHolidayGroupCfg": {
    /*req, object, 人员权限假日组参数配置*/
    "enable": true,
    /*req, bool, 是否启用, desc:true-启用, false-不启用*/
    "groupName": "test",
    /*req, string, 假日组名称*/
    "holidayPlanNo": "1,3,5",
    /*req, string, 假日组计划编号, desc:该字段配置可配置空值, 即"holidayPlanNo" : ""*/
    "operateType": "byTerminal",
    /*opt, enum, 操作类型, subType:string, [byTerminal#按终端操作,byOrg#按组织操作,byTerminalOrg#按终端组织操作], desc:byTerminal:按终端操作*/
    "terminalNoList": [1, 2, 3, 4],
    /*opt, array, 终端ID列表, subType:int, desc:type为byTerminal,byTerminalOrg时必须填*/
    "orgNoList": [1, 2, 3, 4]
    /*opt, array, 组织ID列表, subType:int, desc:type为byOrg,byTerminalOrg时必须填*/
  }
}
```

响应报文

```
{
  "requestURL": "test",
  /*ro, opt, string, URI*/
  "statusCode": "test",
  /*ro, opt, string, 状态码, desc:无法用1表示时（1表示成功且无特殊状态），否则必须返回*/
  "statusString": "test",
  /*ro, opt, string, 状态描述, desc:无法用ok表示时（ok表示成功且无特殊状态），否则必须返回*/
  "subStatusCode": "test",
  /*ro, opt, string, 子状态码, desc:无法用ok表示（ok表示成功且无特殊状态），否则必须返回*/
  "errorCode": 1,
  /*ro, req, int, 错误码, desc:当statusCode不为1时, 与subStatusCode对应*/
  "errorMsg": "ok"
  /*ro, req, string, 错误信息, desc:对errorCode的解释,当statusCode不为1时, 解释信息在协议约束中, 允许设备在后续的版本迭代中, 进行优化丰富提升（不限制死*/
}
```

69.53.12 配置人员权限计划模板参数

Request URL

PUT /ISAPI/AccessControl/UserRightPlanTemplate/<planTemplateID>?format=json

查询参数

参数名称	参数类型	描述
planTemplateID	string	模板编号，从1开始，设备支持的最大值从能力集中获取

请求报文

```
{
  "UserRightPlanTemplate": {
    /*req, object, 人员权限计划模板*/
    "enable": true,
    /*req, bool, 是否启用, desc:true-启用, false-不启用*/
    "templateName": "test",
    /*req, string, 模板名称*/
    "weekPlanNo": 1,
    /*req, int, 周计划编号*/
    "holidayGroupNo": "1,3,5",
    /*req, string, 假日组编号, desc:该字段配置可配置空值, 即"holidayPlanNo" : ""*/
    "operateType": "byTerminal",
    /*opt, enum, 操作类型, subType:string, [byTerminal#按终端操作,byOrg#按组织操作,byTerminalOrg#按终端组织操作]*/
    "terminalNoList": [1, 2, 3, 4],
    /*opt, array, 终端ID列表, subType:int, desc:type为byTerminal,byTerminalOrg时必须填*/
    "orgNoList": [1, 2, 3, 4]
    /*opt, array, 组织ID列表, subType:int, desc:type为byOrg,byTerminalOrg时必须填*/
  }
}
```

响应报文

```
{
  "requestURL": "test",
  /*ro, opt, string, URI*/
  "statusCode": "test",
  /*ro, opt, string, 状态码, desc:无法用1表示时（1表示成功且无特殊状态），否则必须返回*/
  "statusString": "test",
  /*ro, opt, string, 状态描述, desc:无法用ok表示时（ok表示成功且无特殊状态），否则必须返回*/
  "subStatusCode": "test",
  /*ro, opt, string, 子状态码, desc:无法用ok表示（ok表示成功且无特殊状态），否则必须返回*/
  "errorCode": 1,
  /*ro, req, int, 错误码, desc:当statusCode不为1时，与subStatusCode对应*/
  "errorMsg": "ok"
  /*ro, req, string, 错误信息, desc:对errorCode的解释,当statusCode不为1时，解释信息在协议约束中，允许设备在后续的版本迭代中，进行优化丰富提升（不限制死）*/
}
```

69.54 认证方式计划管理

69.54.1 获取读卡器验证方式假日组参数配置能力

Request URL

GET /ISAPI/AccessControl/VerifyHolidayGroupCfg/capabilities?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "VerifyHolidayGroupCfg": {
    /*ro, opt, object, 读卡器验证方式假日组参数*/
    "groupNo": {
      /*ro, opt, object, 假日组编号*/
      "@min": 1,
      /*ro, opt, int, 假日组编号最小值*/
      "@max": 16
      /*ro, opt, int, 假日组编号最大值*/
    },
    "enable": "true,false",
    /*ro, opt, string, 是否启用, desc:true-启用*/
    "groupName": {
      /*ro, opt, object, 假日组名称长度*/
      "@min": 1,
      /*ro, opt, int, 假日组名称长度最小值*/
      "@max": 32
      /*ro, opt, int, 假日组名称长度最大值*/
    },
    "holidayPlanNo": {
      /*ro, opt, object, 假日组计划编号*/
      "@min": 1,
      /*ro, opt, int, 假日组计划编号最小值*/
      "@max": 16
      /*ro, opt, int, 假日组计划编号最大值*/
    }
  }
}
```

69.54.2 获取读卡器验证方式计划模板参数配置能力

Request URL

GET /ISAPI/AccessControl/VerifyPlanTemplate/capabilities?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "VerifyPlanTemplate": {
    /*ro, opt, object, 读卡器验证方式计划模板参数*/
    "templateNo": {
      /*ro, opt, object, 计划模板编号*/
      "@min": 1,
      /*ro, opt, int, 计划模板编号最小值*/
      "@max": 16
      /*ro, opt, int, 计划模板编号最大值*/
    },
    "enable": "true,false",
    /*ro, opt, string, 是否启用, desc:true-启用, false-不启用*/
    "templateName": {
      /*ro, opt, object, 模板名称长度*/
      "@min": 1,
      /*ro, opt, int, 模板名称长度最小值*/
      "@max": 32
      /*ro, opt, int, 模板名称长度最大值*/
    },
    "weekPlanNo": {
      /*ro, opt, object, 周计划编号*/
      "@min": 1,
      /*ro, opt, int, 周计划编号最小值*/
      "@max": 16
      /*ro, opt, int, 周计划编号最大值*/
    },
    "holidayGroupNo": {
      /*ro, opt, object, 假日组编号*/
      "@min": 1,
      /*ro, opt, int, 假日组编号最小值*/
      "@max": 16
      /*ro, opt, int, 假日组编号最大值*/
    }
  }
}
```

69.54.3 获取读卡器验证计划参数

Request URL

GET /ISAPI/AccessControl/CardReaderPlan/<cardReaderID>?format=json

查询参数

参数名称	参数类型	描述
cardReaderID	string	读卡器编号，从1开始，设备支持的最大值从能力集中获取

请求报文

无

响应报文

```
{
  "CardReaderPlan": {
    /*ro, req, object, 计划模板结构*/
    "templateNo": 1
    /*ro, req, int, 计划模板编号, desc:为0表示取消关联, 恢复默认状态*/
  }
}
```

69.54.4 配置读卡器验证计划参数

Request URL

PUT /ISAPI/AccessControl/CardReaderPlan/<cardReaderID>?format=json

查询参数

参数名称	参数类型	描述
cardReaderID	string	读卡器编号，从1开始，设备支持的最大值从能力集中获取

请求报文

```
{
  "CardReaderPlan": {
    /*req, object, 计划模板结构*/
    "templateNo": 1
    /*req, int, 计划模板编号, desc:为0表示取消关联, */
  }
}
```

响应报文

```
{
  "statusCode": 1,
  /*ro, opt, int, 状态码, desc:无法用1表示时（1表示成功且无特殊状态）必须返回*/
  "statusString": "OK",
  /*ro, opt, string, 状态描述, range:[1,64], desc:无法用OK表示时（OK表示成功且无特殊状态）必须返回*/
  "subStatusCode": "ok",
  /*ro, opt, string, 子状态码, range:[1,64], desc:无法用ok表示（ok表示成功且无特殊状态）必须返回*/
  "errorCode": 1,
  /*ro, opt, int, 错误码, desc:当statusCode不为1时，与subStatusCode对应*/
  "errorMsg": "ok"
  /*ro, opt, string, 错误信息, desc:当statusCode不为1时，必须返回,解释信息在协议约束中，允许设备在后续的版本迭代中，进行优化丰富提升（不限制死）*/
}
```

69.54.5 获取读卡器验证计划参数配置能力

Request URL

GET /ISAPI/AccessControl/CardReaderPlan/capabilities?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "CardReaderPlan": {
    /*ro, opt, object, 读卡器编号节点*/
    "cardReaderNo": {
      /*ro, opt, object, 读卡器编号*/
      "@min": 1,
      /*ro, opt, int, 读卡器编号最小值*/
      "@max": 4,
      /*ro, opt, int, 读卡器编号最大值*/
      "@opt": [1, 4]
      /*ro, opt, array, 枚举值范围, subType:int, desc:闸机产品仅支持1和4，不需要返回min和max属性*/
    },
    "templateNo": {
      /*ro, opt, object, 计划模板编号节点*/
      "@min": 1,
      /*ro, opt, int, 计划模板编号最小值*/
      "@max": 16
      /*ro, opt, int, 计划模板编号最大值*/
    }
  }
}
```

69.54.6 获取读卡器验证方式假日计划参数

Request URL

GET /ISAPI/AccessControl/VerifyHolidayPlanCfg/<holidayPlanID>?format=json

查询参数

参数名称	参数类型	描述
holidayPlanID	string	假日计划编号，从1开始，设备支持的最大值从能力集中获取

请求报文

无

响应报文

```
{
  "VerifyHolidayPlanCfg": {
    /*ro, opt, object, 读卡器验证方式假日计划参数*/
    "enable": true,
    /*ro, req, bool, 读卡器验证方式假日计划使能, desc:true-使能, false-不使能*/
    "beginDate": "1970-01-01",
    /*ro, req, date, 假日开始日期, desc:设备本地时间*/
    "endDate": "1970-01-02",
    /*ro, req, date, 假日结束日期, desc:设备本地时间*/
    "HolidayPlanCfg": [
      /*ro, req, array, 假日计划, subType:object*/
      {
        "id": 1,
        /*ro, req, int, 时间段编号, range:[1,8]*/
        "enable": true,
        /*ro, req, bool, 假日计划使能, desc:true-使能, false-不使能*/
        "verifyMode": "cardAndPw",
        /*ro, req, enum, 验证方式, subType:string, [cardAndPw#刷卡+密码, card#刷卡, cardOrPw#刷卡或密码, fp#指纹, fpAndPw#指纹+密码, fpOrCard#指纹或刷卡, fpAndCard#指纹+刷卡, fpAndCardAndPw#指纹+刷卡+密码, faceOrFpOrCardOrPw#人脸或指纹或刷卡或密码, faceAndFp#人脸+指纹, faceAndPw#人脸+密码, faceAndCard#人脸+刷卡, face#人脸, employeeNoAndPw#工号+密码, fpOrPw#指纹或密码, employeeNoAndFp#工号+指纹, employeeNoAndFpAndPw#工号+指纹+密码, faceAndFpAndCard#人脸+指纹+刷卡, faceAndPwAndFp#人脸+密码+指纹, employeeNoAndFace#工号+人脸, faceOrfaceAndCard#人脸或人脸+刷卡, fpOrface#指纹或人脸, cardOrfaceOrPw#刷卡或人脸或密码, cardOrFace#刷卡或人脸, cardOrFaceOrFp#刷卡或人脸或指纹, faceOrPw#人脸或密码, employeeNoAndFaceAndPw#工号+人脸+密码, cardOrFaceOrFaceAndCard#刷卡或人脸或人脸+刷卡, faceOrFpOrPw#人脸或指纹或密码, iris#虹膜, faceOrFpOrCardOrPwOrIris#人脸或指纹或刷卡或密码或虹膜, faceOrCardOrPwOrIris#人脸或刷卡或密码或虹膜, sleep#休眠, invalid#无效, cardOrFpOrFaceOrIris#刷卡或指纹或人脸或虹膜, fpOrFaceOrIrisOrPw#指纹或人脸或虹膜或密码, cardOrFpOrIrisOrPw#刷卡或指纹或虹膜或密码, cardOrIrisOrPw#刷卡或虹膜或密码, cardAndIris#刷卡+虹膜, fpAndIris#指纹+虹膜, faceAndIris#人脸+虹膜, irisAndPw#虹膜+密码, cardAndIrisAndPw#刷卡+虹膜+密码, faceAndIrisAndPw#人脸+虹膜+密码, cardAndFaceAndIris#刷卡+人脸+虹膜, Pw#密码]*/
        "TimeSegment": {
          /*ro, opt, object, 时间节点*/
          "beginTime": "00:00:00",
          /*ro, req, time, 开始时间点, desc:设备本地时间*/
          "endTime": "10:00:00"
          /*ro, req, time, 结束时间点, desc:设备本地时间*/
        }
      }
    ]
  }
}
```

69.54.7 配置读卡器验证方式假日计划参数

Request URL

PUT /ISAPI/AccessControl/VerifyHolidayPlanCfg/<holidayPlanID>?format=json

查询参数

参数名称	参数类型	描述
holidayPlanID	string	假日计划编号，从1开始，设备支持的最大值从能力集中获取

请求报文

```
{
  "VerifyHolidayPlanCfg": {
    /*opt, object, 读卡器验证方式假日计划参数*/
    "enable": true,
    /*req, bool, 读卡器验证方式假日计划使能, desc:true-使能, false-不使能*/
    "beginDate": "1970-01-01",
    /*req, date, 假日开始日期, desc:设备本地时间*/
    "endDate": "1970-01-02",
    /*req, date, 假日结束日期, desc:设备本地时间*/
    "HolidayPlanCfg": [
      /*req, array, 假日计划, subType:object*/
      {
        "id": 1,
        /*req, int, 时间段编号, range:[1,8]*/
        "enable": true,
        /*req, bool, 假日计划使能, desc:true-使能, false-不使能*/
        "verifyMode": "cardAndPw",
        /*req, enum, 验证方式, subType:string, [cardAndPw#刷卡+密码, card#刷卡, cardOrPw#刷卡或密码, fp#指纹, fpAndPw#指纹+密码, fpOrCard#指纹或刷卡, fpAndCard#指纹+刷卡, fpAndCardAndPw#指纹+刷卡+密码, faceOrFpOrCardOrPw#人脸或指纹或刷卡或密码, faceAndFp#人脸+指纹, faceAndPw#人脸+密码, faceAndCard#人脸+刷卡, faceAndPwAndFp#人脸+密码+指纹, employeeNoAndFace#工号+人脸, faceOrfaceAndCard#人脸或人脸+刷卡, fpOrface#指纹或人脸, cardOrfaceOrPw#刷卡或人脸或密码, cardOrFace#刷卡或人脸, cardOrFaceOrFp#刷卡或人脸或指纹, faceOrPw#人脸或密码, employeeNoAndFaceAndPw#工号+人脸+密码, cardOrFaceOrFaceAndCard#刷卡或人脸或人脸+刷卡, faceOrFpOrPw#人脸或指纹或密码, iris#虹膜, faceOrFpOrCardOrPwOrIris#人脸或指纹或刷卡或密码或虹膜, faceOrCardOrPwOrIris#人脸或刷卡或密码或虹膜, sleep#睡眠, invalid#无效, cardOrFpOrFaceOrIris#刷卡或指纹或人脸或虹膜, fpOrFaceOrIrisOrPw#指纹或人脸或虹膜或密码, cardOrFpOrIrisOrPw#刷卡或指纹或虹膜或密码, cardOrIrisOrPw#刷卡或虹膜或密码, cardAndIris#刷卡+虹膜, fpAndIris#指纹+虹膜, faceAndIris#人脸+虹膜, irisAndPw#虹膜+密码, cardAndIrisAndPw#刷卡+虹膜+密码, faceAndIrisAndPw#人脸+虹膜+密码, cardAndFaceAndIris#刷卡+人脸+虹膜, Pw#密码]*/
        "TimeSegment": {
          /*opt, object, 时间节点*/
          "beginTime": "00:00:00",
          /*req, time, 开始时间点, desc:设备本地时间*/
          "endTime": "10:00:00",
          /*req, time, 结束时间点, desc:设备本地时间*/
        }
      }
    ]
  }
}
```

响应报文

```
{
  "statusCode": 1,
  /*ro, opt, int, 状态码, desc:无法用1表示时（1表示成功且无特殊状态）必须返回*/
  "statusString": "OK",
  /*ro, opt, string, 状态描述, range:[1,64], desc:无法用OK表示时（OK表示成功且无特殊状态）必须返回*/
  "subStatusCode": "ok",
  /*ro, opt, string, 子状态码, range:[1,64], desc:无法用ok表示（ok表示成功且无特殊状态）必须返回*/
  "errorCode": 1,
  /*ro, opt, int, 错误码, desc:当statusCode不为1时，与subStatusCode对应*/
  "errorMsg": "ok",
  /*ro, opt, string, 错误信息, desc:当statusCode不为1时，必须返回,解释信息在协议约束中，允许设备在后续的版本迭代中，进行优化丰富提升（不限制死）*/
}
```

69.54.8 获取读卡器验证方式周计划参数

Request URL

GET /ISAPI/AccessControl/VerifyWeekPlanCfg/<weekPlanID>?format=json

查询参数

参数名称	参数类型	描述
weekPlanID	string	周计划编号，从1开始，设备支持的最大值从能力集中获取

请求报文

无

响应报文

```
{
  "VerifyWeekPlanCfg": {
    /*ro, opt, object, 读卡器验证方式周计划*/
    "enable": true,
    /*ro, req, bool, 读卡器验证方式周计划参数使能, desc:true-使能, false-不使能*/
    "WeekPlanCfg": [
      /*ro, req, array, 周计划参数, subType:object*/
      {
        "week": "Monday",
        /*ro, req, enum, 星期, subType:string, [Monday#星期一,Tuesday#星期二,Wednesday#星期三,Thursday#星期四,Friday#星期五,Saturday#星期六,Sunday#星期日]*/
        "id": 1,
        /*ro, req, int, 时间段编号, range:[1,8]*/
        "enable": true,
        /*ro, req, bool, 周计划使能*/
        "verifyMode": "cardAndPw",
        /*ro, req, enum, 验证方式, subType:string, [cardAndPw#刷卡+密码,card#刷卡,cardOrPw#刷卡或密码,fp#指纹,fpAndPw#指纹+密码,fpOrCard#指纹或刷卡,fpAndCard#指纹+刷卡,fpAndCardAndPw#指纹+刷卡+密码,faceOrFpOrCardOrPw#人脸或指纹或刷卡或密码,faceAndFp#人脸+指纹,faceAndPw#人脸+密码,faceAndCard#人脸+刷卡,face#人脸,employeeNoAndPw#工号+密码,fpOrPw#指纹或密码,employeeNoAndFp#工号+指纹,employeeNoAndFpAndPw#工号+指纹+密码,faceAndFpAndCard#人脸+指纹+刷卡,faceAndPwAndFp#人脸+密码+指纹,employeeNoAndFace#工号+人脸,faceOrfaceAndCard#人脸或人脸+刷卡,fpOrface#指纹或人脸,cardOrfaceOrPw#刷卡或人脸或密码,cardOrFace#刷卡或人脸,cardOrFaceOrFp#刷卡或人脸或指纹,faceOrPw#人脸或密码,employeeNoAndFaceAndPw#工号+人脸+密码,cardOrFaceOrFaceAndCard#刷卡或人脸或人脸+刷卡,faceOrFpOrPw#人脸或指纹或密码,iris#虹膜,faceOrFpOrCardOrPwOrIris#人脸或指纹或刷卡或密码或虹膜,faceOrCardOrPwOrIris#人脸或刷卡或密码或虹膜,sleep#休眠,invalid#无效,cardOrFpOrFaceOrIris#刷卡或指纹或人脸或虹膜,fpOrFaceOrIrisOrPw#指纹或人脸或虹膜或密码,cardOrFpOrIrisOrPw#刷卡或指纹或虹膜或密码,cardOrIrisOrPw#刷卡或虹膜或密码,cardAndIris#刷卡+虹膜,fpAndIris#指纹+虹膜,faceAndIris#人脸+虹膜,irisAndPw#虹膜+密码,cardAndIrisAndPw#刷卡+虹膜+密码,faceAndIrisAndPw#人脸+虹膜+密码,cardAndFaceAndIris#刷卡+人脸+虹膜,Pw#密码]*/
        "TimeSegment": {
          /*ro, req, object, 时间节点*/
          "beginTime": "00:00:00",
          /*ro, req, time, 开始时间点, desc:设备本地时间*/
          "endTime": "10:00:00"
          /*ro, req, time, 结束时间点, desc:设备本地时间*/
        }
      }
    ]
  }
}
```

69.54.9 设置读卡器验证方式周计划参数

Request URL

PUT /ISAPI/AccessControl/VerifyWeekPlanCfg/<weekPlanID>?format=json

查询参数

参数名称	参数类型	描述
weekPlanID	string	周计划编号，从1开始，设备支持的最大值从能力集中获取

请求报文

```

{
  "VerifyWeekPlanCfg": {
    /*opt, object, 读卡器验证方式周计划*/
    "enable": true,
    /*req, bool, 读卡器验证方式周计划参数使能, desc:true-使能, false-不使能*/
    "WeekPlanCfg": [
      /*req, array, 周计划参数, subType:object*/
      {
        "week": "Monday",
        /*req, enum, 星期, subType:string, [Monday#星期一,Tuesday#星期二,Wednesday#星期三,Thursday#星期四,Friday#星期五,Saturday#星期六,Sunday#星期日]*/
        "id": 1,
        /*req, int, 时间段编号, range:[1,8]*/
        "enable": true,
        /*req, bool, 周计划使能*/
        "verifyMode": "cardAndPw",
        /*req, enum, 验证方式, subType:string, [cardAndPw#刷卡+密码,card#刷卡,cardOrPw#刷卡或密码,fp#指纹,fpAndPw#指纹+密码,fpOrCard#指纹或刷卡,fpAndCard#指纹+刷卡,fpAndCardAndPw#指纹+刷卡+密码,faceOrFpOrCardOrPw#人脸或指纹或刷卡或密码,faceAndFp#人脸+指纹,faceAndPw#人脸+密码,faceAndCard#人脸+刷卡,face#人脸,employeeNoAndPw#工号+密码,fpOrPw#指纹或密码,employeeNoAndFp#工号+指纹,employeeNoAndFpAndPw#工号+指纹+密码,faceAndFpAndCard#人脸+指纹+刷卡,faceAndPwAndFp#人脸+密码+指纹,employeeNoAndFace#工号+人脸,faceOrfaceAndCard#人脸或人脸+刷卡,fpOrface#指纹或人脸,cardOrfaceOrPw#刷卡或人脸或密码,cardOrFace#刷卡或人脸,cardOrFaceOrFp#刷卡或人脸或指纹,faceOrPw#人脸或密码,employeeNoAndFaceAndPw#工号+人脸+密码,cardOrFaceOrFaceAndCard#刷卡或人脸或人脸+刷卡,faceOrFpOrPw#人脸或指纹或密码,iris#虹膜,faceOrFpOrCardOrPwOrIris#人脸或指纹或刷卡或密码或虹膜,faceOrCardOrPwOrIris#人脸或刷卡或密码或虹膜,sleep#睡眠,invalid#无效,cardOrFpOrFaceOrIris#刷卡或指纹或人脸或虹膜,fpOrFaceOrIrisOrPw#指纹或人脸或虹膜或密码,cardOrFpOrIrisOrPw#刷卡或指纹或虹膜或密码,cardOrIrisOrPw#刷卡或虹膜或密码,cardAndIris#刷卡+虹膜,fpAndIris#指纹+虹膜,faceAndIris#人脸+虹膜,irisAndPw#虹膜+密码,cardAndIrisAndPw#刷卡+虹膜+密码,faceAndIrisAndPw#人脸+虹膜+密码,cardAndFaceAndIris#刷卡+人脸+虹膜,Pw#密码]*/
        "TimeSegment": {
          /*req, object, 时间节点*/
          "beginTime": "00:00:00",
          /*req, time, 开始时间点, desc:设备本地时间*/
          "endTime": "10:00:00"
          /*req, time, 结束时间点, desc:设备本地时间*/
        }
      }
    ]
  }
}

```

响应报文

```

{
  "statusCode": 1,
  /*ro, opt, int, 状态码, desc:无法用1表示时（1表示成功且无特殊状态）必须返回*/
  "statusString": "OK",
  /*ro, opt, string, 状态描述, range:[1,64], desc:无法用OK表示时（OK表示成功且无特殊状态）必须返回*/
  "subStatusCode": "ok",
  /*ro, opt, string, 子状态码, range:[1,64], desc:无法用ok表示（ok表示成功且无特殊状态）必须返回*/
  "errorCode": 1,
  /*ro, opt, int, 错误码, desc:当statusCode不为1时, 与subStatusCode对应*/
  "errorMsg": "ok"
  /*ro, opt, string, 错误信息, desc:当statusCode不为1时, 必须返回,解释信息在协议约束中, 允许设备在后续的版本迭代中, 进行优化丰富提升（不限制死）*/
}

```

69.54.10 获取读卡器验证方式假日计划能力

Request URL

GET /ISAPI/AccessControl/VerifyHolidayPlanCfg/capabilities?format=json

查询参数

无

请求报文

无

响应报文

```

{
  "VerifyHolidayPlanCfg": {
    /*ro, opt, object, 读卡器验证方式假日计划*/
    "planNo": {
      /*ro, opt, object, 假日计划编号*/
      "@min": 1,
      /*ro, opt, int, 假日计划编号最小值*/
      "@max": 16,
      /*ro, opt, int, 假日计划编号最大值*/
    },
    "enable": "true,false",
    /*ro, opt, string, 读卡器验证方式假日计划使能, desc:true-使能, false-不使能*/
    "beginDate": "1970-01-01",
    /*ro, opt, date, 假日开始日期, desc:设备本地时间*/
    "endDate": "1970-01-02",
    /*ro, opt, date, 假日结束日期, desc:设备本地时间*/
    "HolidayPlanCfg": {
      /*ro, opt, object, 假日计划*/
      "maxSize": 8,
      /*ro, opt, int, 假日计划最大值*/
      "id": {
        /*ro, opt, object, 时间段编号*/
        "@min": 1,
        /*ro, opt, int, 时间段编号最小值*/
        "@max": 8,
        /*ro, opt, int, 时间段编号最大值*/
      },
      "enable": "true,false",
      /*ro, opt, string, 假日计划使能, desc:true-使能, false-不使能*/
      "verifyMode": {
        /*ro, opt, object, 验证方式*/
        "@opt":
          "cardAndPw,card,cardOrPw,fp,fpAndPw,fpOrCard,fpAndCard,fpAndCardAndPw,faceOrFpOrCardOrPw,faceAndFp,faceAndPw,faceAndCard,face,employeeNoAndPw,fpOrPw,employe
          eNoAndFp,employeeNoAndFpAndPw,faceAndFpAndCard,faceAndPwAndFp,employeeNoAndFace,faceOrFaceAndCard,fpOrFace,cardOrFaceOrPw,cardOrFace,cardOrFaceOrFp,cardOrFp
          OrPw,faceOrPw,employeeNoAndFaceAndPw,cardOrFaceOrFaceAndCard,faceOrFpOrPw,iris,faceOrFpOrCardOrPwOrIris,faceOrCardOrPwOrIris,sleep,invalid,cardOrFpOrFaceOrI
          ris,fpOrFaceOrIrisOrPw,cardOrFpOrIrisOrPw,cardOrIrisOrPw,cardAndIris,fpAndIris,faceAndIris,irisAndPw,cardAndIrisAndPw,faceAndIrisAndPw,cardAndFaceAndIris,Pw
          "
          /*ro, opt, string, 验证方式枚举值, desc:cardAndPw-刷卡+密码,card-刷卡,cardOrPw-刷卡或密码,fp-指纹,fpAndPw-指纹+密码,fpOrCard-指纹或刷
          卡,fpAndCard-指纹+刷卡,fpAndCardAndPw-指纹+刷卡+密码,faceOrFpOrCardOrPw-人脸或指纹或刷卡或密码,faceAndFp-人脸+指纹,faceAndPw-人脸+密码,faceAndCard-人脸+刷
          卡,face-人脸,employeeNoAndPw-工号+密码,fpOrPw-指纹或密码,employeeNoAndFp-工号+指纹,employeeNoAndFpAndPw-工号+指纹+密码,faceAndFpAndCard-人脸+指纹+刷
          卡,faceAndPwAndFp-人脸+密码+指纹,employeeNoAndFace-工号+人脸,faceOrFaceAndCard-人脸或人脸+刷卡,fpOrFace-指纹或人脸,cardOrFaceOrPw-刷卡或人脸或密
          码,cardOrFace-刷卡或人脸,cardOrFaceOrFp-刷卡或人脸或指纹,cardOrFpOrPw-刷卡或指纹或密码,faceOrPw-人脸或密码,employeeNoAndFaceAndPw-工号+人脸+密
          码,cardOrFaceOrFaceAndCard-刷卡或人脸或人脸+刷卡,faceOrFpOrPw#人脸或指纹或密码,iris-虹膜,faceOrFpOrCardOrPwOrIris-人脸或指纹或卡或密码或虹
          膜,faceOrCardOrPwOrIris-人脸或卡或指纹或虹膜,sleep-休眠,invalid-无效,cardOrFpOrFaceOrIris-刷卡或指纹或人脸或虹膜,fpOrFaceOrIrisOrPw-指纹或人脸或虹膜或密
          码,cardOrFpOrIrisOrPw-刷卡或指纹或虹膜或密码,cardOrIrisOrPw-刷卡或虹膜或密码,cardAndIris-刷卡+虹膜,fpAndIris-指纹+虹膜,faceAndIris-人脸+虹膜,irisAndPw-虹膜
          +密码,cardAndIrisAndPw-刷卡+虹膜+密码,faceAndIrisAndPw-人脸+虹膜+密码,cardAndFaceAndIris-刷卡+人脸+虹膜,Pw-密码*/
      },
      "TimeSegment": {
        /*ro, opt, object, 时间节点*/
        "beginTime": "00:00:00",
        /*ro, opt, time, 开始时间点, desc:设备本地时间*/
        "endTime": "10:00:00",
        /*ro, opt, time, 结束时间点, desc:设备本地时间*/
        "validUnit": "minute",
        /*ro, opt, enum, 时间点精度, subType:string, [hour#小时,minute#分钟,second#秒], desc:不返回该字段, 则代表精度为分钟*/
      }
    },
    "purePwdVerifyEnable": true
  }
  /*ro, opt, bool, 纯密码开门,
  desc:1、true-设备支持纯密码开门（人员信息中的-password字段），不返回本字段-不支持纯密码开门。
  2、纯密码方案：
  （1）认证方式中的"或密码"为人员密码。
  （2）设备不对人员密码的重复性进行校验，需要上层平台自行保证密码的唯一性。
  （3）设备本地不能对人员密码进行操作（增删改查）。*/
}
}

```

69.54.11 获取读卡器验证方式周计划参数配置能力

Request URL

GET /ISAPI/AccessControl/VerifyWeekPlanCfg/capabilities?format=json

查询参数

无

请求报文

无

响应报文

```

{
  "VerifyWeekPlanCfg": {
    /*ro, opt, object, 读卡器验证方式周计划参数*/
    "planNo": {
      /*ro, opt, object, 周计划编号*/
      "@min": 1,
      /*ro, opt, int, 周计划编号最小值*/
      "@max": 16
      /*ro, opt, int, 周计划编号最大值*/
    },
    "enable": "true,false",
    /*ro, opt, string, 读卡器验证方式周计划参数使能, desc:true-使能, false-不使能*/
    "WeekPlanCfg": {
      /*ro, opt, object, 周计划参数*/
      "maxSize": 56,
      /*ro, opt, int, 周计划参数最大值*/
      "week": {
        /*ro, opt, object, 星期*/
        "@opt": "Monday,Tuesday,Wednesday,Thursday,Friday,Saturday,Sunday"
        /*ro, opt, string, 星期枚举值, desc:Monday-星期一,Tuesday-星期二,Wednesday-星期三,Thursday-星期四,Friday-星期五,Saturday-星期六,Sunday-星期日*/
      },
      "id": {
        /*ro, opt, object, 周计划编号*/
        "@min": 1,
        /*ro, opt, int, 周计划编号最小值*/
        "@max": 8
        /*ro, opt, int, 周计划编号最大值*/
      },
      "enable": "true,false",
      /*ro, opt, string, 周计划编号使能, desc:true-使能, false-不使能*/
      "verifyMode": {
        /*ro, opt, object, 验证方式*/
        "@opt":
          "cardAndPw,card,cardOrPw,fp,fpAndPw,fpOrCard,fpAndCard,fpAndCardAndPw,faceOrFpOrCardOrPw,faceAndFp,faceAndPw,faceAndCard,face,employeeNoAndPw,fpOrPw,employeeNoAndFp,employeeNoAndFpAndPw,faceAndFpAndCard,faceAndPwAndFp,employeeNoAndFace,faceOrFaceAndCard,fpOrface,cardOrfaceOrPw,cardOrFace,cardOrFaceOrFp,cardOrFpOrPw,faceOrPw,employeeNoAndFaceAndPw,cardOrFaceOrFaceAndCard,faceOrFpOrPw,iris,faceOrFpOrCardOrPwOrIris,faceOrCardOrPwOrIris,cardOrFpOrFaceOrIris,fpOrFaceOrIrisOrPw,cardOrFpOrIrisOrPw,cardOrIrisOrPw,cardAndIris,fpAndIris,faceAndIris,irisAndPw,cardAndIrisAndPw,faceAndIrisAndPw,cardAndFaceAndIris,Pw,sleep,invalid"
          /*ro, opt, string, 验证方式枚举值, desc:cardAndPw-刷卡+密码,card-刷卡,cardOrPw-刷卡或密码,fp-指纹,fpAndPw-指纹+密码,fpOrCard-指纹或刷卡,fpAndCard-指纹+刷卡,fpAndCardAndPw-指纹+刷卡+密码,faceOrFpOrCardOrPw-人脸或指纹或刷卡或密码,faceAndFp-人脸+指纹,faceAndPw-人脸+密码,faceAndCard-人脸+刷卡,face-人脸,employeeNoAndPw-工号+密码,fpOrPw-指纹或密码,employeeNoAndFp-工号+指纹,employeeNoAndFpAndPw-工号+指纹+密码,faceAndFpAndCard-人脸+指纹+刷卡,faceAndPwAndFp-人脸+密码+指纹,employeeNoAndFace-工号+人脸,faceOrfaceAndCard-人脸或人脸+刷卡,fpOrface-指纹或人脸,cardOrfaceOrPw-刷卡或人脸或密码,cardOrFace-刷卡或人脸,cardOrFaceOrFp-刷卡或人脸或指纹,cardOrFpOrPw-刷卡或指纹或密码,faceOrPw-人脸或密码,employeeNoAndFaceAndPw-工号+人脸+密码,cardOrFaceOrFaceAndCard-刷卡或人脸或人脸+刷卡,faceOrFpOrPw#人脸或指纹或密码,iris-虹膜,faceOrFpOrCardOrPwOrIris-人脸或指纹或卡或密码或虹膜,faceOrCardOrPwOrIris-人脸或卡或指纹或虹膜,cardOrFpOrFaceOrIris-刷卡或指纹或人脸或虹膜,fpOrFaceOrIrisOrPw-指纹或人脸或虹膜或密码,cardOrFpOrIrisOrPw-刷卡或指纹或虹膜或密码,cardOrIrisOrPw-刷卡或虹膜或密码,cardAndIris-刷卡+虹膜,fpAndIris-指纹+虹膜,faceAndIris-人脸+虹膜,irisAndPw-虹膜+密码,cardAndIrisAndPw-刷卡+虹膜+密码,faceAndIrisAndPw-人脸+虹膜+密码,cardAndFaceAndIris-刷卡+人脸+虹膜,Pw-密码,sleep-休眠,invalid-无效*/
      },
      "TimeSegment": {
        /*ro, opt, object, 时间节点*/
        "beginTime": "00:00:00",
        /*ro, opt, time, 开始时间点, desc:设备本地时间*/
        "endTime": "10:00:00",
        /*ro, opt, time, 结束时间点, desc:设备本地时间*/
        "validUnit": "minute"
        /*ro, opt, enum, 时间点精度, subType:string, [hour#小时,minute#分钟,second#秒], desc:不返回该字段, 则代表精度为分钟*/
      }
    },
    "purePwVerifyEnable": true
    /*ro, opt, bool, 纯密码开门,
    desc:1-true-设备支持纯密码开门(人员信息中的-password字段), 不返回本字段-不支持纯密码开门。
    2、纯密码方案:
    (1) 认证方式中的"或密码"为人员密码。
    (2) 设备不对人员密码的重复性进行校验, 需要上层平台自行保证密码的唯一性。
    (3) 设备本地不能对人员密码进行操作(增删改查)。*/
  }
}

```

69.54.12 配置读卡器验证方式假日组参数

Request URL

PUT /ISAPI/AccessControl/VerifyHolidayGroupCfg/<holidayGroupID>?format=json

查询参数

参数名称	参数类型	描述
holidayGroupID	string	假日组编号, 从1开始, 设备支持的最大值从能力集中获取

请求报文

```
{
  "VerifyHolidayGroupCfg": {
    /*opt, object, 读卡器验证方式假日组参数*/
    "enable": true,
    /*req, bool, 是否启用, desc:true-启用, false-不启用*/
    "groupName": "test",
    /*req, string, 假日组名称*/
    "holidayPlanNo": "1,3,5"
    /*opt, string, 假日组计划编号, desc:该字段配置可配置空值, 即"holidayPlanNo" : ""*/
  }
}
```

响应报文

```
{
  "statusCode": 1,
  /*ro, opt, int, 状态码, desc:无法用1表示时（1表示成功且无特殊状态）必须返回*/
  "statusString": "OK",
  /*ro, opt, string, 状态描述, range:[1,64], desc:无法用OK表示时（OK表示成功且无特殊状态）必须返回*/
  "subStatusCode": "ok",
  /*ro, opt, string, 子状态码, range:[1,64], desc:无法用ok表示（ok表示成功且无特殊状态）必须返回*/
  "errorCode": 1,
  /*ro, opt, int, 错误码, desc:当statusCode不为1时, 与subStatusCode对应*/
  "errorMsg": "ok"
  /*ro, opt, string, 错误信息, desc:当statusCode不为1时, 必须返回,解释信息在协议约束中, 允许设备在后续的版本迭代中, 进行优化丰富提升（不限制死）*/
}
```

69.54.13 设置读卡器验证方式计划模板参数

Request URL

PUT /ISAPI/AccessControl/VerifyPlanTemplate/<planTemplateID>?format=json

查询参数

参数名称	参数类型	描述
planTemplateID	string	计划模板编号, 从1开始, 设备支持的最大值从能力集中获取

请求报文

```
{
  "VerifyPlanTemplate": {
    /*opt, object, 读卡器验证方式计划模板*/
    "enable": true,
    /*req, bool, 是否启用, desc:true-启用, false-不启用*/
    "templateName": "test",
    /*req, string, 模板名称*/
    "weekPlanNo": 1,
    /*req, int, 周计划编号*/
    "holidayGroupNo": "1,3,5"
    /*req, string, 假日组编号, desc:该字段配置可配置空值, 即"holidayGroupNo" : ""*/
  }
}
```

响应报文

```
{
  "statusCode": 1,
  /*ro, opt, int, 状态码, desc:无法用1表示时（1表示成功且无特殊状态）必须返回*/
  "statusString": "OK",
  /*ro, opt, string, 状态描述, range:[1,64], desc:无法用OK表示时（OK表示成功且无特殊状态）必须返回*/
  "subStatusCode": "ok",
  /*ro, opt, string, 子状态码, range:[1,64], desc:无法用ok表示（ok表示成功且无特殊状态）必须返回*/
  "errorCode": 1,
  /*ro, opt, int, 错误码, desc:当statusCode不为1时, 与subStatusCode对应*/
  "errorMsg": "ok"
  /*ro, opt, string, 错误信息, desc:当statusCode不为1时, 必须返回,解释信息在协议约束中, 允许设备在后续的版本迭代中, 进行优化丰富提升（不限制死）*/
}
```

69.54.14 获取读卡器验证方式计划模板参数

Request URL

GET /ISAPI/AccessControl/VerifyPlanTemplate/<planTemplateID>?format=json

查询参数

参数名称	参数类型	描述
planTemplateID	string	计划模板编号，从1开始，设备支持的最大值从能力集中获取

请求报文

无

响应报文

```
{
  "VerifyPlanTemplate": {
    /*ro, opt, object, 读卡器验证方式计划模板*/
    "enable": true,
    /*ro, req, bool, 是否启用, desc:true-启用, false-不启用*/
    "templateName": "test",
    /*ro, req, string, 模板名称*/
    "weekPlanNo": 1,
    /*ro, req, int, 周计划编号*/
    "holidayGroupNo": "1,3,5"
    /*ro, req, string, 假日组编号, desc:该字段配置可配置空值, 即"holidayGroupNo" : ""*/
  }
}
```

69.54.15 获取读卡器验证方式假日组参数

Request URL

GET /ISAPI/AccessControl/VerifyHolidayGroupCfg/<holidayGroupID>?format=json

查询参数

参数名称	参数类型	描述
holidayGroupID	string	假日组编号，从1开始，设备支持的最大值从能力集中获取

请求报文

无

响应报文

```
{
  "VerifyHolidayGroupCfg": {
    /*ro, opt, object, 读卡器验证方式假日组参数*/
    "enable": true,
    /*ro, req, bool, 是否启用, desc:true-启用, false-不启用*/
    "groupName": "test",
    /*ro, req, string, 假日组名称*/
    "holidayPlanNo": "1,3,5"
    /*ro, opt, string, 假日组计划编号, desc:该字段配置可配置空值, 即"holidayPlanNo" : ""*/
  }
}
```

69.55 离线采集

69.55.1 获取离线采集能力

Request URL

GET /ISAPI/AccessControl/OfflineCapture/capabilities?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "OfflineCaptureCap": {
```

```

/*ro, req, object, 能力*/
"isSupportDownloadOfflineCaptureInfoTemplate": true,
/*ro, opt, bool, 是否支持离线用户列表模板下载*/
"isSupportUploadOfflineCaptureInfo": true,
/*ro, opt, bool, 是否支持离线用户列表文件上传*/
"isSupportDownloadCaptureData": true,
/*ro, opt, bool, 是否支持下载采集数据*/
"isSupportDeleteAllData": true,
/*ro, opt, bool, 是否支持删除全部采集数据*/
"isSupportDeleteTheData": true,
/*ro, opt, bool, 是否支持删除指定采集数据*/
"SearchTask": {
/*ro, opt, object, 搜索任务能力*/
"supportFunction": {
/*ro, opt, object, 支持的方法*/
"@opt": ["put", "get", "delete", "post"]
/*ro, req, array, 枚举, subType:string*/
},
"searchID": {
/*ro, opt, object, 搜索记录唯一标识*/
"@min": 0,
/*ro, req, int, 最小长度*/
"@max": 0
/*ro, req, int, 最大长度*/
},
},
"maxResults": {
/*ro, opt, object, 最大结果个数*/
"@min": 0,
/*ro, req, int, 最小值*/
"@max": 0
/*ro, req, int, 最大值*/
},
},
"captureNoList": {
/*ro, opt, object, 采集序号列表*/
"maxSize": 1,
/*ro, req, int, 列表个数*/
"@min": 0,
/*ro, req, int, 最小值*/
"@max": 0
/*ro, req, int, 最大值*/
},
},
"searchType": {
/*ro, opt, object, 采集序号*/
"@opt": ["new", "modified"]
/*ro, req, array, 枚举, subType:string*/
},
},
"DataCollections": {
/*ro, opt, object, 查询到的匹配数据信息*/
"maxSize": 0,
/*ro, req, int, 数据最大个数*/
"captureNo": {
/*ro, opt, object, 采集序号*/
"@min": 0,
/*ro, req, int, 最小值*/
"@max": 0
/*ro, req, int, 最大值*/
},
},
"name": {
/*ro, opt, object, 姓名*/
"@min": 0,
/*ro, req, int, 最小长度*/
"@max": 0
/*ro, req, int, 最大长度*/
},
},
"employeeNo": {
/*ro, opt, object, 工号*/
"@min": 0,
/*ro, req, int, 最小长度*/
"@max": 0
/*ro, req, int, 最大长度*/
},
},
"CardNoList": {
/*ro, opt, object, 采集序号列表*/
"maxSize": 0,
/*ro, req, int, 数组最大个数*/
"cardNo": {
/*ro, opt, object, 卡号*/
"@min": 0,
/*ro, req, int, 最小长度*/
"@max": 0
/*ro, req, int, 最大长度*/
},
},
"cardType": {
/*ro, opt, object, 卡类型*/
},
},
"IDCardNo": {
/*ro, opt, object, 身份证号*/
"@min": 0,
/*ro, req, int, 最小长度*/
"@max": 0
/*ro, req, int, 最大长度*/
},
},
"FingerprintList": {
/*ro, opt, object, 指纹列表*/
}

```

湖南云舟电力科技有限公司

2024-08-15

```
/*ro, opt, object, 指纹列表*/
"fingerprintID": {
/*ro, opt, object, 指纹编号*/
"@min": 0,
/*ro, req, int, 最小值*/
"@max": 0
/*ro, req, int, 最大值*/
},
"fingerprint": {
/*ro, opt, object, 指纹信息*/
"@min": 0,
/*ro, req, int, 最小长度*/
"@max": 0
/*ro, req, int, 最大长度*/
}
},
"FaceFeature": {
/*ro, opt, object, 人脸特征*/
"isSupportFaceRegion": true,
/*ro, opt, bool, 是否支持人脸特征区域*/
"isSupportCommonPoint": true
/*ro, opt, bool, 是否支持特征点坐标*/
},
"isSupportRiskMark": true,
/*ro, opt, bool, 是否支持风险数据标记*/
"dataType": {
/*ro, opt, object, 数据类型*/
"@opt": ["new", "modified", "normal"]
/*ro, req, array, 枚举, subType:string*/
},
"IdentityInfo": {
/*ro, opt, object, 身份证信息*/
"chnName": {
/*ro, opt, object, 中文姓名*/
"@min": 0,
/*ro, req, int, 最小长度*/
"@max": 0
/*ro, req, int, 最大长度*/
},
"enName": {
/*ro, opt, object, 英文姓名*/
"@min": 0,
/*ro, req, int, 最小长度*/
"@max": 0
/*ro, req, int, 最大长度*/
},
"sex": {
/*ro, opt, object, 性别*/
"@opt": ["1", "2", "0"]
/*ro, req, array, 枚举, subType:string*/
},
"birth": {
/*ro, opt, object, 出生日期*/
"@min": 0,
/*ro, req, int, 最小长度*/
"@max": 0
/*ro, req, int, 最大长度*/
},
"addr": {
/*ro, opt, object, 住址*/
"@min": 0,
/*ro, req, int, 最小长度*/
"@max": 0
/*ro, req, int, 最大长度*/
},
"IDCardNo": {
/*ro, opt, object, 身份证号码*/
"@min": 0,
/*ro, req, int, 最小长度*/
"@max": 0
/*ro, req, int, 最大长度*/
},
"issuingAuthority": {
/*ro, opt, object, 签发机关*/
"@min": 0,
/*ro, req, int, 最小长度*/
"@max": 0
/*ro, req, int, 最大长度*/
},
"startDate": {
/*ro, opt, object, 有效期起始时间*/
"@min": 0,
/*ro, req, int, 最小长度*/
"@max": 0
/*ro, req, int, 最大长度*/
},
"endDate": {
/*ro, opt, object, 有效期截止时间*/
"@min": 0,
/*ro, req, int, 最小长度*/
"@max": 0
/*ro, req, int, 最大长度*/
},
"nation": {
```

```

/*ro, opt, object, 民族*/
    "@min": 1,
    /*ro, req, int, 最小值, range:[1,56]*/
    "@max": 56
    /*ro, req, int, 最大值, range:[1,56]*/
},
"passNo": {
    /*ro, opt, object, 通行证号码*/
    "@min": 0,
    /*ro, req, int, 最小长度*/
    "@max": 0
    /*ro, req, int, 最大长度*/
},
"issueNumber": {
    /*ro, opt, object, 签发次数*/
    "@min": 0,
    /*ro, req, int, 最小长度*/
    "@max": 0
    /*ro, req, int, 最大长度*/
},
"certificateType": {
    /*ro, opt, object, 证件类型识别*/
},
"permanentResidenceCardNo": {
    /*ro, opt, object, 永久居留证号码*/
    "@min": 0,
    /*ro, req, int, 最小长度*/
    "@max": 0
    /*ro, req, int, 最大长度*/
},
"nationalityOrAreaCode": {
    /*ro, opt, object, 国籍或地区代码*/
    "@min": 0,
    /*ro, req, int, 最小长度*/
    "@max": 0
    /*ro, req, int, 最大长度*/
},
"version": {
    /*ro, opt, object, 证件版本号*/
    "@min": 0,
    /*ro, req, int, 最小长度*/
    "@max": 0
    /*ro, req, int, 最大长度*/
},
"receivingAuthorityCode": {
    /*ro, opt, object, 受理机关代码*/
    "@min": 0,
    /*ro, req, int, 最小长度*/
    "@max": 0
    /*ro, req, int, 最大长度*/
},
"FingerprintList": {
    /*ro, opt, object, 指纹信息列表*/
    "maxSize": 0,
    /*ro, req, int, 数组最大个数*/
    "fingerprint": {
        /*ro, opt, object, 指纹信息*/
        "@min": 0,
        /*ro, req, int, 最小长度*/
        "@max": 0
        /*ro, req, int, 最大长度*/
    }
},
"pic": {
    /*ro, opt, object, 证件相片信息*/
    "@min": 0,
    /*ro, req, int, 最小长度*/
    "@max": 0
    /*ro, req, int, 最大长度*/
}
},
"CardIssueStatus": {
    /*ro, opt, object, 指纹人脸信息发行状态列表*/
    "@size": 0,
    /*ro, req, int, 数组元素个数能力*/
    "face": {
        /*ro, opt, object, 人脸发卡状态*/
        "@opt": [true, false]
        /*ro, req, array, 枚举, subType=bool*/
    },
    "fingerprint1": {
        /*ro, opt, object, 指纹1发卡状态*/
        "@opt": [true, false]
        /*ro, req, array, 枚举, subType=bool*/
    },
    "fingerprint2": {
        /*ro, opt, object, 指纹2发卡状态*/
        "@opt": [true, false]
        /*ro, req, array, 枚举, subType=bool*/
    }
}
}
}

```

湖南云舟电力科技有限公司

```

"RuleInfo": {
/*ro, opt, object, 规则列表*/
  "reqAdminRights": [true, false],
/*ro, opt, array, 是否需要管理员权限, subType:bool*/
  "enableCardNoLenAuto": [true, false],
/*ro, opt, array, 使能卡序列号长度自适应, subType:bool*/
  "maxSize": 0,
/*ro, opt, int, 规则列表最大个数*/
  "supportFunction": {
/*ro, opt, object, 支持的方法*/
    "@opt": ["put", "get", "delete", "post"]
/*ro, req, array, 枚举, subType:string*/
  },
  "dataType": {
/*ro, opt, object, 数据类型*/
    "@opt": ["name", "employeeNo", "IDCardNo", "IDCardSerialNo", "IDCardDetails", "card", "fingerprint", "face"]
/*ro, req, array, 枚举, subType:string*/
  },
  "enable": [true, false],
/*ro, opt, array, 是否采集及显示, subType:bool*/
  "uniqueCheck": [true, false],
/*ro, opt, array, 是否唯一性校验, subType:bool*/
  "len": [
/*ro, opt, array, 按照数据类型返回能力列表, subType:object*/
    {
      "dataType": "name",
/*ro, req, enum, 数据类型, subType:string, [name#姓名, employeeNo#工号, card#卡]*/
      "@min": 0,
/*ro, req, int, 最小长度*/
      "@max": 0
/*ro, req, int, 最大长度*/
    }
  ],
  "num": [
/*ro, opt, array, 按照数据类型返回能力列表, subType:object*/
    {
      "dataType": "card",
/*ro, req, enum, 数据类型, subType:string, [card#卡, fingerprint#指纹]*/
      "@min": 0,
/*ro, req, int, 最小值*/
      "@max": 0
/*ro, req, int, 最大值*/
    }
  ],
  "fingerprintIDs": {
/*ro, opt, object, 指纹编号*/
    "@min": 0,
/*ro, req, int, 最小值*/
    "@max": 0
/*ro, req, int, 最大值*/
  },
  "isLocalStorage": {
/*ro, opt, object, 设备本地是否存储指纹和人脸信息*/
    "@opt": [true, false]
/*ro, req, array, 枚举, subType:bool*/
  }
},
"CaptureProgress": {
/*ro, opt, object, 采集进度*/
  "supportFunction": {
/*ro, opt, object, 支持的方法*/
    "@opt": ["put", "get", "delete", "post"]
/*ro, req, array, 枚举, subType:string*/
  },
  "reqCaptureNum": {
/*ro, opt, object, 待采集人员总数量*/
    "@min": 0,
/*ro, req, int, 最小值*/
    "@max": 0
/*ro, req, int, 最大值*/
  },
  "partiallyCaptureNum": {
/*ro, opt, object, 已部分采集人员数量*/
    "@min": 0,
/*ro, req, int, 最小值*/
    "@max": 0
/*ro, req, int, 最大值*/
  },
  "reqFaceNum": {
/*ro, opt, object, 待采集人脸数量*/
    "@min": 0,
/*ro, req, int, 最小值*/
    "@max": 0
/*ro, req, int, 最大值*/
  },
  "faceNum": {
/*ro, opt, object, 已采集人脸数量*/
    "@min": 0,
/*ro, req, int, 最小值*/
    "@max": 0
/*ro, req, int, 最大值*/
  },
  "reqFingerprintNum": {
/*ro, opt, object, 待采集指纹数量*/

```

```

        "@min": 0,
        /*ro, req, int, 最小值*/
        "@max": 0
        /*ro, req, int, 最大值*/
    },
    "fingerprintNum": {
        /*ro, opt, object, 已采集指纹数量*/
        "@min": 0,
        /*ro, req, int, 最小值*/
        "@max": 0
        /*ro, req, int, 最大值*/
    },
    "reqCardNum": {
        /*ro, opt, object, 待采集卡数量*/
        "@min": 0,
        /*ro, req, int, 最小值*/
        "@max": 0
        /*ro, req, int, 最大值*/
    },
    "cardNum": {
        /*ro, opt, object, 已采集卡数量*/
        "@min": 0,
        /*ro, req, int, 最小值*/
        "@max": 0
        /*ro, req, int, 最大值*/
    },
    "reqIDCardNum": {
        /*ro, opt, object, 待采集身份证数量*/
        "@min": 0,
        /*ro, req, int, 最小值*/
        "@max": 0
        /*ro, req, int, 最大值*/
    },
    "IDCardNum": {
        /*ro, opt, object, 已采集身份证数量*/
        "@min": 0,
        /*ro, req, int, 最小值*/
        "@max": 0
        /*ro, req, int, 最大值*/
    },
    },
    "DataOutput": {
        /*ro, opt, object, 数据导出*/
        "supportFunction": {
            /*ro, opt, object, 支持的方法*/
            "@opt": ["put", "get", "delete", "post"]
            /*ro, req, array, 枚举, subType:string*/
        },
        "password": {
            /*ro, opt, object, 导出密码*/
            "@min": 0,
            /*ro, req, int, 最小长度*/
            "@max": 0
            /*ro, req, int, 最大长度*/
        },
        "type": {
            /*ro, opt, object, 导出方式*/
            "@opt": "USB"
            /*ro, req, string, 枚举*/
        },
        "progress": {
            /*ro, opt, object, 导出进度*/
            "@min": 0,
            /*ro, req, int, 最小值*/
            "@max": 0
            /*ro, req, int, 最大值*/
        }
    },
    "UploadFailedDetails": {
        /*ro, opt, object, 上传失败信息*/
        "description": {
            /*ro, opt, object, 详细描述字符串*/
            "@min": 0,
            /*ro, req, int, 最小长度*/
            "@max": 0
            /*ro, req, int, 最大长度*/
        }
    },
    },
}
}

```

69.55.2 获取离线采集用户列表文件上传失败信息

Request URL

GET /ISAPI/AccessControl/OfflineCapture/uploadFailedDetails?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "UploadFailedDetails ": {
    /*ro, req, object, 失败详细信息*/
    "description": "test"
    /*ro, req, string, 描述上传失败细节*/
  }
}
```

69.55.3 获取离线采集规则参数

Request URL

GET /ISAPI/AccessControl/OfflineCapture/ruleInfo?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "RuleInfo": {
    /*ro, req, object, 规则*/
    "reqAdminRights": true,
    /*ro, req, bool, 是否需要管理员权限,
    desc:操作员可以采集数据
    管理员可以查看后台采集数据
    管理员权限采集时需要输入密码进行采集, 退出采集界面后需要重新输入密码*/
    "enableCardNoLenAuto": true,
    /*ro, opt, bool, 卡序列号长度自适应, desc:优先级高于len字段, 卡号长度自适应为true后, 规则列表中的卡号规则数据长度无效*/
    "RuleList": [
      /*ro, opt, array, 规则列表, subType:object, desc:不同数据采集的规则, 支持增量下发*/
      {
        "dataType": "name",
        /*ro, req, enum, 数据类型, subType:string, [name#姓名,employeeNo#工号,IDCardNo#身份证号,IDCardSerialNo#身份证序列号,IDCardDetails#身份证详细信息,card#卡,fingerprint#指纹,face#人脸]*/
        "enable": true,
        /*ro, req, bool, 是否采集及显示*/
        "uniqueCheck": true,
        /*ro, opt, bool, 唯一性校验(默认不需要), desc:数据类型为name时有效, 其他类型均为只读的可选参数, 不支持不返回*/
        "len": 128,
        /*ro, opt, int, 数据长度, desc:if datatype=="name", "employeeNo"或者"card"有效, 姓名长度, 默认128; 其他类型均为只读的可选参数,不支持不返回*/
        "num": 2,
        /*ro, opt, int, 采集数量, desc:if datatype=="fingerprint"或者"card"有效*/
        "fingerprintIDs": [1, 2]
        /*ro, opt, array, 需要采集的指纹ID列表, subType:int, desc:if datatype=="fingerprint", 需要采集的指纹ID列表*/
      }
    ],
    "enableLocalIssueCard": true,
    /*ro, opt, bool, 是否使能本地生物发卡, desc:采集人脸和指纹后, 信息存入生物卡中*/
    "isLocalStorage": false
    /*ro, opt, bool, 设备本地是否存储指纹和人脸信息, desc:隐私功能, 针对海外场景, 采集指纹和人脸数据后, 是否进行本地存储*/
  }
}
```

69.55.4 配置离线采集规则参数

Request URL

PUT /ISAPI/AccessControl/OfflineCapture/ruleInfo?format=json

查询参数

无

请求报文

```
{
  "RuleInfo": {
    /*req, object, 规则*/
    "reqAdminRights": true,
    /*req, bool, 是否需要管理员权限,
    desc:操作人员可以采集数据
    管理员可以查看后台采集数据
    管理员权限采集时需要输入密码进行采集, 退出采集界面后需要重新输入密码*/
    "enableCardNoLenAuto": true,
    /*opt, bool, 卡序列号长度自适应, desc:优先级高于len字段, 卡号长度自适应为true后, 规则列表中的卡号规则数据长度无效*/
    "RuleList": [
      /*opt, array, 规则列表, subType:object, desc:不同数据采集的规则, 支持增量下发*/
      {
        "dataType": "name",
        /*req, enum, 数据类型, subType:string, [name#姓名,employeeNo#工号,IDCardNo#身份证号,IDCardSerialNo#身份证序列号,IDCardDetails#身份证详细信息,card#卡,fingerprint#指纹,face#人脸]*/
        "enable": true,
        /*req, bool, 是否采集及显示*/
        "uniqueCheck": true,
        /*opt, bool, 唯一性校验（默认不需要）, desc:数据类型为name时有效, 其他类型均为只读的可选参数, 不支持不返回*/
        "len": 128,
        /*opt, int, 数据长度, desc:if datatype=="name", "employeeNo"或者"card"有效, 姓名长度, 默认128; 其他类型均为只读的可选参数,不支持不返回*/
        "num": 2,
        /*opt, int, 采集数量, desc:if datatype=="fingerprint"或者"card"有效*/
        "fingerprintIDs": [1, 2]
        /*opt, array, 需要采集的指纹ID列表, subType:int, desc:if datatype=="fingerprint", 需要采集的指纹ID列表*/
      }
    ],
    "enableLocalIssueCard": true,
    /*opt, bool, 是否使能本地生物发卡, desc:采集人脸和指纹后, 信息存入生物卡中*/
    "isLocalStorage": false
    /*opt, bool, 设备本地是否存储指纹和人脸信息, desc:隐私功能, 针对海外场景, 采集指纹和人脸数据后, 是否进行本地存储*/
  }
}
```

响应报文

```
{
  "statusCode": 1,
  /*ro, opt, int, 状态码, desc:无法用1表示时（1表示成功且无特殊状态）必须返回*/
  "statusString": "OK",
  /*ro, opt, string, 状态描述, range:[1,64], desc:无法用OK表示时（OK表示成功且无特殊状态）必须返回*/
  "subStatusCode": "ok",
  /*ro, opt, string, 子状态码, range:[1,64], desc:无法用ok表示（ok表示成功且无特殊状态）必须返回*/
  "errorCode": 1,
  /*ro, opt, int, 错误码, desc:当statusCode不为1时, 与subStatusCode对应*/
  "errorMsg": "ok"
  /*ro, opt, string, 错误信息, desc:当statusCode不为1时, 必须返回,解释信息在协议约束中, 允许设备在后续的版本迭代中, 进行优化丰富提升（不限制死）*/
}
```

69.55.5 查询离线采集进度

Request URL

GET /ISAPI/AccessControl/OfflineCapture/progress?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "CaptureProgress": {
    /*ro, req, object, 采集数据*/
    "reqCaptureNum": 1,
    /*ro, opt, int, 待采集人员总数量*/
    "completelyCaptureNum": 1,
    /*ro, opt, int, 已采集完整的人员数量*/
    "partiallyCaptureNum": 1,
    /*ro, opt, int, 已部分采集人员数量*/
    "reqFaceNum": 1,
    /*ro, opt, int, 待采集人脸数量*/
    "faceNum": 1,
    /*ro, opt, int, 已采集人脸数量*/
    "reqFingerprintNum": 1,
    /*ro, opt, int, 待采集指纹数量*/
    "fingerprintNum": 1,
    /*ro, opt, int, 已采集指纹数量*/
    "reqCardNum": 1,
    /*ro, opt, int, 待采集卡数量*/
    "cardNum": 1,
    /*ro, opt, int, 已采集卡数量*/
    "reqIDCardNum": 1,
    /*ro, opt, int, 待采集身份证数量*/
    "IDCardNum": 1,
    /*ro, opt, int, 已采集身份证数量*/
    "reqIssueNum": 1,
    /*ro, opt, int, 待生物发卡人员数量*/
    "IssuedNum": 1,
    /*ro, opt, int, 已生物发卡人员数量*/
  }
}
```

69.55.6 获取离线采集数据导出进度

Request URL

GET /ISAPI/AccessControl/OfflineCapture/dataOutput/progress?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "DataOutputProgress": {
    /*ro, req, object, 进度*/
    "progress": 1
    /*ro, req, int, 导出进度, range:[1,100]*/
  }
}
```

69.56 在线采集

69.56.1 采集虹膜信息

Request URL

POST /ISAPI/AccessControl/captureIrisData?format=json

查询参数

无

请求报文

```
{
  "CaptureIrisData": {
    /*req, object, 参数, desc:当第一次采集未完成,第二次采集信令下发时,设备会返回状态-设备正忙,等待第一次采集完成后,才可以第二次采集,即阻塞采集方式,和人脸采集保持一致*/
    "eyesType": "monocular"
    /*req, enum, 采集的眼睛类型, subType:string, [monocular#单目,binocular#双目,auto#自动模式], desc:自动模式下设备自动识别采集单眼还是双眼(举例:如用户只有一只眼睛的场景,此模式下会自动采集单眼)*/
  }
}
```

响应报文

```
{
  "statusCode": 1,
  /*ro, opt, int, 状态码, desc:无法用1表示时（1表示成功且无特殊状态）必须返回*/
  "statusString": "OK",
  /*ro, opt, string, 状态描述, range:[1,64], desc:无法用OK表示时（OK表示成功且无特殊状态）必须返回*/
  "subStatusCode": "ok",
  /*ro, opt, string, 子状态码, range:[1,64], desc:无法用ok表示（ok表示成功且无特殊状态）必须返回*/
  "errorCode": 1,
  /*ro, opt, int, 错误码, desc:当statusCode不为1时，与subStatusCode对应*/
  "errorMsg": "ok"
  /*ro, opt, string, 错误信息, desc:当statusCode不为1时，必须返回,解释信息在协议约束中，允许设备在后续的版本迭代中，进行优化丰富提升（不限制死）*/
}
```

69.56.2 获取采集指纹信息能力

Request URL

GET /ISAPI/AccessControl/CaptureFingerPrint/capabilities

查询参数

无

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<CaptureFingerPrint xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 采集指纹信息, attr:version{req, string, 协议版本}-->
  <CaptureFingerPrintCond>
    <!--ro, req, object, 指纹采集条件-->
    <fingerNo min="1" max="10">
      <!--ro, opt, int, 手指编号, range:[1,10], attr:min{req, int, 最小值},max{req, int, 最大值}-->1
    </fingerNo>
  </CaptureFingerPrintCond>
  <fingerData min="1" max="768">
    <!--ro, opt, string, 指纹数据, range:[1,768], attr:min{req, int, 最小值},max{req, int, 最大值}-->test
  </fingerData>
  <fingerNo min="1" max="10">
    <!--ro, opt, int, 手指编号, range:[1,10], attr:min{req, int, 最小值},max{req, int, 最大值}-->1
  </fingerNo>
  <fingerPrintQuality min="1" max="100">
    <!--ro, opt, int, 指纹质量, range:[1,100], attr:min{req, int, 最小值},max{req, int, 最大值}-->1
  </fingerPrintQuality>
</CaptureFingerPrint>
```

69.56.3 采集指纹信息

Request URL

POST /ISAPI/AccessControl/CaptureFingerPrint

查询参数

无

请求报文

```
<?xml version="1.0" encoding="UTF-8"?>

<CaptureFingerPrintCond xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--opt, object, 采集指纹信息条件, attr:version{req, string, 协议版本}-->
  <fingerNo>
    <!--req, int, 手指编号, range:[1,10]-->1
  </fingerNo>
</CaptureFingerPrintCond>
```

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<CaptureFingerPrint xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 指纹数据, attr:version{req, string, 协议版本}-->
  <fingerData>
    <!--ro, opt, string, 指纹数据, range:[1,768], desc:指纹数据 (需要Base64编码)-->test
  </fingerData>
  <fingerNo>
    <!--ro, req, int, 手指编号, range:[1,10]-->1
  </fingerNo>
  <fingerPrintQuality>
    <!--ro, req, int, 指纹质量, range:[1,100]-->1
  </fingerPrintQuality>
</CaptureFingerPrint>
```

69.56.4 获取采集卡信息参数能力

Request URL

GET /ISAPI/AccessControl/CaptureCardInfo/capabilities?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "CardInfoCap": {
    /*ro, req, object, 能力*/
    "cardNo": {
      /*ro, opt, object, 卡号*/
      "@min": 1,
      /*ro, req, int, 最小长度, range:[1,32]*/
      "@max": 32
      /*ro, req, int, 最大长度, range:[1,32]*/
    },
    "cardType": ["TypeA_M1", "TypeA_CPU", "TypeB", "ID_125K", "FelicaCard", "DesfireCard"],
    /*ro, opt, array, 卡类型, subType:string, range:[1,6]*/
    "readerID": {
      /*ro, opt, object, 识别单元编号*/
      "@min": 1,
      /*ro, req, int, 取值范围*/
      "@max": 8
      /*ro, req, int, 取值范围*/
    }
  }
}
```

69.56.5 获取人脸信息采集能力

Request URL

GET /ISAPI/AccessControl/CaptureFaceData/capabilities

查询参数

无

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<CaptureFaceData xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, 人脸信息采集能力, attr:version{req, string, 协议版本}-->
  <CaptureFaceDataCond>
    <!--ro, req, bool, 是否采集红外-->true
    <captureInfrared opt="true,false">
      <!--ro, opt, bool, 是否采集红外, attr:opt{req, string, 取值范围}-->true
    </captureInfrared>
    <dataType opt="url,binary">
      <!--ro, opt, enum, 采集图片数据类型, subType:string, [url#url,binary#二进制], attr:opt{req, string, 取值范围}, desc:默认url,采集图片数据类型-->url
    </dataType>
  </CaptureFaceDataCond>
  <faceDataUrl min="1" max="768">
    <!--ro, opt, string, 人脸数据URL, range:[1,768], attr:min{req, int, 最小值},max{req, int, 最大值}-->1
  </faceDataUrl>
  <captureProgress min="1" max="10">
    <!--ro, req, int, 采集进度, range:[1,10], attr:min{req, int, 最小值},max{req, int, 最大值}-->1
  </captureProgress>
  <infraredFaceDataUrl min="1" max="100">
    <!--ro, req, string, 红外图片URL, range:[1,100], attr:min{req, int, 最小值},max{req, int, 最大值}-->test
  </infraredFaceDataUrl>
  <modelData min="1" max="10">
    <!--ro, opt, string, 人脸建模数据 (base64编码), attr:min{req, int, 最小值},max{req, int, 最大值}-->test
  </modelData>
  <score min="0" max="100">
    <!--ro, opt, int, 人脸评分, range:[0,100], attr:min{req, int, 最小值},max{req, int, 最大值}-->80
  </score>
  <thermometryUnit opt="celsius,fahrenheit,kelvin">
    <!--ro, opt, string, 测温单位 (celsius-摄氏度 (默认), attr:opt{req, string, 取值范围}-->test
  </thermometryUnit>
  <currTemperature min="0" max="10">
    <!--ro, opt, float, 人脸温度 (精确到小数点后一位), attr:min{req, int, 最小值},max{req, int, 最大值}-->0.000
  </currTemperature>
  <visibleLightURL min="0" max="10">
    <!--ro, opt, string, 热成像相机可见光图片URL, attr:min{req, int, 最小值},max{req, int, 最大值}-->test
  </visibleLightURL>
  <thermalURL min="0" max="10">
    <!--ro, opt, string, 热成像图片URL, attr:min{req, int, 最小值},max{req, int, 最大值}-->test
  </thermalURL>
  <readerID min="1" max="8">
    <!--ro, opt, int, 识别单元编号, range:[1,8], attr:min{req, int, 最小值},max{req, int, 最大值}-->1
  </readerID>
  <requireReaderID>
    <!--ro, opt, bool, 请求URL中是否必须包含readerID,
    desc:字段不存在默认为false。
    对于智能主机仅支持URL按readerID配置-->true
  </requireReaderID>
</CaptureFaceData>
```

69.56.6 获取采集人脸信息进度能力集

Request URL

GET /ISAPI/AccessControl/CaptureFaceData/Progress/capabilities

查询参数

无

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<CaptureFaceData xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 采集人脸信息能力, attr:version{req, string, 协议版本}-->
  <faceDataUrl min="0" max="10">
    <!--ro, opt, string, 人脸数据URL, range:[1,32], attr:min{req, int, 最小值},max{req, int, 最大值}, desc:无该节点代表无人脸-->test
  </faceDataUrl>
  <captureProgress min="0" max="100">
    <!--ro, req, int, 采集进度, range:[0,100], attr:min{req, int, 最小值},max{req, int, 最大值}, desc:0-未采集到人脸, 100-采集到人脸（只有在进度为100时, 才解析人脸URL）-->1
  </captureProgress>
  <isCurRequestOver opt="true,false">
    <!--ro, opt, bool, 当前采集的请求是否完成, attr:opt{req, string, 取值范围}-->true
  </isCurRequestOver>
  <infraredFaceDataUrl min="0" max="10">
    <!--ro, opt, string, 红外人脸数据URL, range:[1,32], attr:min{req, int, 最小值},max{req, int, 最大值}, desc:无该节点代表无红外人脸-->test
  </infraredFaceDataUrl>
  <readerID min="1" max="8">
    <!--ro, opt, int, 识别单元编号, range:[1,8], attr:min{req, int, 最小值},max{req, int, 最大值}-->1
  </readerID>
  <requireReaderID>
    <!--ro, opt, bool, 请求URL中是否必须包含readerID,
    desc:字段不存在默认为false。
    对于智能主机仅支持URL按readerID配置-->true
  </requireReaderID>
</CaptureFaceData>
```

69.56.7 获取采集人脸进度

Request URL

GET /ISAPI/AccessControl/CaptureFaceData/Progress?readerID=<readerID>

查询参数

参数名称	参数类型	描述
readerID	string	智能主机支持按读卡器采集人脸

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<CaptureFaceData xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, 人脸数据, attr:version{req, string, 协议版本}-->
  <faceDataUrl>
    <!--ro, opt, string, 人脸数据URL, range:[1,32]-->test
  </faceDataUrl>
  <captureProgress>
    <!--ro, req, int, 采集进度, range:[0,100], desc:只有在进度为100时,才解析人脸URL或数据-->100
  </captureProgress>
  <isCurRequestOver>
    <!--ro, opt, bool, 当前采集的请求是否完成, desc:true-表示采集操作完成 否则返回false,当收到true时,采集操作已经完成,无需再重复获取进度,如果未采集到人脸,可重新发起采集操作-->true
  </isCurRequestOver>
  <infraredFaceDataUrl>
    <!--ro, opt, string, 红外人脸数据URL, range:[1,32], desc:无该节点代表无红外人脸-->test
  </infraredFaceDataUrl>
</CaptureFaceData>
```

69.56.8 采集人脸信息

Request URL

POST /ISAPI/AccessControl/CaptureFaceData

查询参数

无

请求报文

```
<?xml version="1.0" encoding="UTF-8"?>

<CaptureFaceDataCond xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, 采集条件, attr:version{req, string, 协议版本}-->
  <captureInfrared>
    <!--opt, bool, 采集模式, desc:true-同时采集红外图片 false-不采集红外图片,默认不采集红外-->true
  </captureInfrared>
  <dataType>
    <!--opt, enum, 采集图片数据类型, subType:string, [url#url,binary#二进制数据], desc:默认url,采集图片数据类型-->url
  </dataType>
  <readerID>
    <!--opt, int, 识别单元编号, range:[1,8], desc:字段不存在时, 表示对编号1采集, 当前仅智能主机支持多读卡器采集-->1
  </readerID>
</CaptureFaceDataCond>
```

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<CaptureFaceData xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 人脸数据采集条件, attr:version{req, string, 协议版本}-->
  <CaptureFaceDataCond>
    <!--ro, opt, object, 人脸数据条件-->
    <captureInfrared opt="true,false">
      <!--ro, opt, bool, 是否采集红外, attr:opt{req, string, 取值范围}-->true
    </captureInfrared>
    <dataType opt="url,binary">
      <!--ro, opt, enum, 采集图片数据类型, subType:string, [url#url,binary#二进制], attr:opt{req, string, 取值范围}-->url
    </dataType>
  </CaptureFaceDataCond>
  <faceDataURL min="1" max="768">
    <!--ro, opt, string, 人脸数据URL, attr:min{req, int, 最小值},max{req, int, 最大值}-->test
  </faceDataURL>
  <captureProgress min="1" max="10">
    <!--ro, opt, int, 采集进度, range:[0,100], attr:min{req, int, 最小值},max{req, int, 最大值}-->1
  </captureProgress>
  <infraredFaceDataURL min="1" max="100">
    <!--ro, opt, string, 红外图片URL, attr:min{req, int, 最小值},max{req, int, 最大值}-->test
  </infraredFaceDataURL>
  <modelData>
    <!--ro, opt, string, 人脸建模数据 (base64编码) -->test
  </modelData>
  <score>
    <!--ro, opt, int, 人脸评分, range:[0,100]-->0
  </score>
  <thermometryUnit>
    <!--ro, opt, string, 测温单位 (celsius-摄氏度 (默认) -->test
  </thermometryUnit>
  <currTemperature>
    <!--ro, opt, float, 人脸温度 (精确到小数点后一位) -->0.0
  </currTemperature>
  <visibleLightURL>
    <!--ro, opt, string, 热成像相机可见光图片URL-->test
  </visibleLightURL>
  <thermalURL>
    <!--ro, opt, string, 热成像图片URL-->test
  </thermalURL>
</CaptureFaceData>
```

参数名称	参数值	参数类型(Content-Type)	Content-ID	文件名称(filename)	描述
设备未实现	[报文内容]	application/xml	--	--	--
FaceData	[图片二进制数据]	image/jpeg		FaceDataac	可见光图片
InfraredFaceData	[图片二进制数据]	image/jpeg		InfraredFaceData.jpg	红外图片

说明：此协议通过表单格式传输，表单框架详情说明见章节4.5.1.4，如下实例：

```
--<frontier>
Content-Disposition: form-data; name=参数名称;filename=文件名称
Content-Type: 参数类型
Content-Length: ****
Content-ID: 内容ID
参数值
```

- 参数名称：表单单元Headers中Content-Disposition的name属性，表示此表单单元的名称
- 参数类型（Content-Type）：表单单元Headers中Content-Type属性
- 文件名称（filename）：表单单元Headers中Content-Disposition的filename属性，仅在本表单单元传输数据为文

件类型存在，表示表单单元Body的文件名

- 参数值：表单单元的body内容

69.56.9 获取在线采集预置参数

Request URL

GET /ISAPI/AccessControl/CapturePresetParam?format=json&security= <security> &iv= <iv>

查询参数

参数名称	参数类型	描述
security	string	没有security表示数据不加密； security=1，表示报文中的name字段AES128加密； security=2，表示报文中的name字段AES256加密
iv	string	随机字符串，长度限制为32； iv表示初始化向量，在security=1或者2时为必须参数

请求报文

无

响应报文

```
{
  "CapturePreset": {
    /*ro, req, object, 在线采集预置参数*/
    "name": "test"
    /*ro, opt, string, 姓名, range:[1,128]*/
  }
}
```

69.56.10 配置在线采集预置参数

Request URL

PUT /ISAPI/AccessControl/CapturePresetParam?format=json&security= <security> &iv= <iv>

查询参数

参数名称	参数类型	描述
security	string	没有security表示数据不加密； security=1，表示报文中的name字段AES128加密； security=2，表示报文中的name字段AES256加密
iv	string	随机字符串，长度限制为32； iv表示初始化向量，在security=1或者2时为必须参数

请求报文

```
{
  "CapturePreset": {
    /*req, object, 在线采集预置参数*/
    "name": "test"
    /*opt, string, 姓名, range:[1,128]*/
  }
}
```

响应报文

```
{
  "statusCode": 1,
  /*ro, opt, int, 状态码, desc:无法用1表示时（1表示成功且无特殊状态）必须返回*/
  "statusString": "OK",
  /*ro, opt, string, 状态描述, range:[1,64], desc:无法用OK表示时（OK表示成功且无特殊状态）必须返回*/
  "subStatusCode": "ok",
  /*ro, opt, string, 子状态码, range:[1,64], desc:无法用ok表示（ok表示成功且无特殊状态）必须返回*/
  "errorCode": 1,
  /*ro, opt, int, 错误码, desc:当statusCode不为1时，与subStatusCode对应*/
  "errorMsg": "ok"
  /*ro, opt, string, 错误信息, desc:当statusCode不为1时，必须返回,解释信息在协议约束中，允许设备在后续的版本迭代中，进行优化丰富提升（不限制死）*/
}
```

69.56.11 获取在线采集预置参数能力

Request URL

GET /ISAPI/AccessControl/CapturePresetParam/capabilities?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "CapturePresetCap": {
    /*ro, req, object, 能力*/
    "name": {
      /*ro, opt, object, 姓名*/
      "@min": 1,
      /*ro, opt, int, 最小长度, range:[1,128]*/
      "@max": 128
      /*ro, opt, int, 最大长度, range:[1,128]*/
    }
  }
}
```

69.56.12 获取在线采集规则参数

Request URL

GET /ISAPI/AccessControl/CaptureRule?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "CaptureRule": {
    /*ro, req, object, 在线采集规则*/
    "enableCardNoLenAuto": true,
    /*ro, opt, bool, 卡序列号长度自适应*/
    "cardNoLen": 4,
    /*ro, opt, enum, 卡序列号长度, subType:int, [3#3字节,4#4字节,7#7字节,10#10字节]*/
    "cardTimeout": 1000
    /*ro, opt, int, 设备采集卡超时参数, range:[1,5000], unit:ms, unitType:时间*/
  }
}
```

69.56.13 配置在线采集规则参数

Request URL

PUT /ISAPI/AccessControl/CaptureRule?format=json

查询参数

无

请求报文

```
{
  "CaptureRule": {
    /*req, object, 在线采集规则*/
    "enableCardNoLenAuto": true,
    /*opt, bool, 卡序列号长度自适应*/
    "cardNoLen": 4,
    /*opt, enum, 卡序列号长度, subType:int, [3#3字节,4#4字节,7#7字节,10#10字节]*/
    "cardTimeout": 1000
    /*opt, int, 设备采集卡超时参数, range:[1,5000], unit:ms, unitType:时间*/
  }
}
```

响应报文

```
{
  "statusCode": 1,
  /*ro, opt, int, 状态码, desc:无法用1表示时（1表示成功且无特殊状态）必须返回*/
  "statusString": "OK",
  /*ro, opt, string, 状态描述, range:[1,64], desc:无法用OK表示时（OK表示成功且无特殊状态）必须返回*/
  "subStatusCode": "ok",
  /*ro, opt, string, 子状态码, range:[1,64], desc:无法用ok表示（ok表示成功且无特殊状态）必须返回*/
  "errorCode": 1,
  /*ro, opt, int, 错误码, desc:当statusCode不为1时，与subStatusCode对应*/
  "errorMsg": "ok"
  /*ro, opt, string, 错误信息, desc:当statusCode不为1时，必须返回,解释信息在协议约束中，允许设备在后续的版本迭代中，进行优化丰富提升（不限制死）*/
}
```

69.56.14 获取在线采集规则参数能力

Request URL

GET /ISAPI/AccessControl/CaptureRule/capabilities?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "CaptureRuleCap": {
    /*ro, req, object, 能力*/
    "enableCardNoLenAuto": [true, false],
    /*ro, opt, array, 使能卡序列号长度自适应, subType:bool*/
    "cardNoLen": {
      /*ro, opt, object, 卡序列号长度*/
      "@opt": [3, 4, 7, 10]
      /*ro, req, array, 枚举, subType:int*/
    },
    "cardTimeout": {
      /*ro, opt, object, 设备采集卡超时参数单位ms*/
      "@min": 0,
      /*ro, req, int, 最小值*/
      "@max": 5000
      /*ro, req, int, 最大值*/
    }
  }
}
```

69.56.15 获取采集身份证信息参数能力

Request URL

GET /ISAPI/AccessControl/CaptureIDInfo/capabilities?format=json

查询参数

无

请求报文

无

```
{
  "IdentityInfoCap": {
    /*ro, req, object, 能力*/
    "IdentityInfoCond": {},
    /*ro, opt, object, 身份证采集条件*/
    "chnName": {
      /*ro, opt, object, 中文姓名*/
      "@min": 0,
      /*ro, req, int, 最小长度*/
      "@max": 0
      /*ro, req, int, 最大长度*/
    },
    "enName": {
      /*ro, opt, object, 英文姓名*/
      "@min": 0,
      /*ro, req, int, 最小长度*/
      "@max": 0
      /*ro, req, int, 最大长度*/
    },
    "sex": {
      /*ro, opt, object, 性别*/
      "@opt": ["male", "female"]
      /*ro, req, array, 枚举, subType:string*/
    },
    "birth": {
      /*ro, opt, object, 出生日期*/
      "@min": 0,
      /*ro, req, int, 最小长度*/
      "@max": 0
      /*ro, req, int, 最大长度*/
    },
    "addr": {
      /*ro, opt, object, 住址*/
      "@min": 0,
      /*ro, req, int, 最小长度*/
      "@max": 0
      /*ro, req, int, 最大长度*/
    },
    "IDCardNo": {
      /*ro, opt, object, 身份证号码*/
      "@min": 0,
      /*ro, req, int, 最小长度*/
      "@max": 0
      /*ro, req, int, 最大长度*/
    },
    "issuingAuthority": {
      /*ro, opt, object, 签发机关*/
      "@min": 0,
      /*ro, req, int, 最小长度*/
      "@max": 0
      /*ro, req, int, 最大长度*/
    },
    "startDate": {
      /*ro, opt, object, 有效期起始时间*/
      "@min": 0,
      /*ro, req, int, 最小长度*/
      "@max": 0
      /*ro, req, int, 最大长度*/
    },
    "endDate": {
      /*ro, opt, object, 有效期截止时间*/
      "@min": 0,
      /*ro, req, int, 最小长度*/
      "@max": 0
      /*ro, req, int, 最大长度*/
    },
    "nation": {
      /*ro, opt, object, 民族*/
      "@min": 1,
      /*ro, req, int, 最小值, range:[1,56]*/
      "@max": 56
      /*ro, req, int, 最大值, range:[1,56]*/
    },
    "passNo": {
      /*ro, opt, object, 通行证号码*/
      "@min": 0,
      /*ro, req, int, 最小长度*/
      "@max": 0
      /*ro, req, int, 最大长度*/
    },
    "issueNumber": {
      /*ro, opt, object, 签发次数*/
      "@min": 0,
      /*ro, req, int, 最小长度*/
      "@max": 0
      /*ro, req, int, 最大长度*/
    },
    "certificateType": {
      /*ro, opt, object, 证件类型识别*/
      "@min": 0,
```

```

        /*ro, req, int, 最小长度*/
        "@max": 0
        /*ro, req, int, 最大长度*/
    },
    "permanentResidenceCardNo": {
        /*ro, opt, object, 永久居留证号码*/
        "@min": 0,
        /*ro, req, int, 最小长度*/
        "@max": 0
        /*ro, req, int, 最大长度*/
    },
    "nationalityOrAreaCode": {
        /*ro, opt, object, 国籍或地区代码*/
        "@min": 0,
        /*ro, req, int, 最小长度*/
        "@max": 0
        /*ro, req, int, 最大长度*/
    },
    "version": {
        /*ro, opt, object, 证件版本号*/
        "@min": 0,
        /*ro, req, int, 最小长度*/
        "@max": 0
        /*ro, req, int, 最大长度*/
    },
    "receivingAuthorityCode": {
        /*ro, opt, object, 受理机关代码*/
        "@min": 0,
        /*ro, req, int, 最小长度*/
        "@max": 0
        /*ro, req, int, 最大长度*/
    },
    "FingerprintList": {
        /*ro, opt, object, 指纹信息列表*/
        "maxSize": 0,
        /*ro, req, int, 数组最大个数*/
        "fingerprint": {
            /*ro, opt, object, 指纹信息*/
            "@min": 0,
            /*ro, req, int, 最小长度*/
            "@max": 0
            /*ro, req, int, 最大长度*/
        }
    },
    "pic": {
        /*ro, opt, object, 证件相片信息*/
        "@min": 0,
        /*ro, req, int, 最小长度*/
        "@max": 0
        /*ro, req, int, 最大长度*/
    }
}
}
}

```

69.57 访客凭条信息管理

69.57.1 配置访客凭条信息

Request URL

PUT /ISAPI/AccessControl/visitorReceiptInfoCfg?format=json

查询参数

无

请求报文

```
{
  "VisitorReceiptInfoCfg": {
    /*req, object, 访客凭条信息配置*/
    "VisitorReceiptInfoList": [
      /*opt, array, 访客凭条信息列表, subType:object, desc:只是规定了打印项的顺序, 不涉及打印项的内容*/
      {
        "id": 1,
        /*req, int, 顺序号, desc:从1开始*/
        "enabled": true,
        /*req, bool, 是否打印使能*/
        "content": "QRCode"
        /*opt, enum, 凭条内容, subType:string, [QRCode#权限二维码,picture#访客照片,name#访客姓名,IDNum#身份证号,address#访客地址,time#有效日期,receptionist#接待人员,receptionistDepartment#接待人员部门,mattersNeedAttention#注意事项,reason#来访事由,visitTime#实际来访时间,appointmentLeaveTime#预约离开时间,visitorRightList#访客权限组列表]*/
      }
    ],
    "mattersNeedAttention": "test"
    /*opt, string, 注意事项, range:[0,64], desc:平台可自定义注意事项的内容, 比如禁止吸烟、禁止拍照等。*/
  }
}
```

响应报文

```
{
  "statusCode": 1,
  /*ro, opt, int, 状态码, desc:无法用1表示时（1表示成功且无特殊状态）必须返回*/
  "statusString": "OK",
  /*ro, opt, string, 状态描述, range:[1,64], desc:无法用OK表示时（OK表示成功且无特殊状态）必须返回*/
  "subStatusCode": "ok",
  /*ro, opt, string, 子状态码, range:[1,64], desc:无法用ok表示（ok表示成功且无特殊状态）必须返回*/
  "errorCode": 1,
  /*ro, opt, int, 错误码, desc:当statusCode不为1时, 与subStatusCode对应*/
  "errorMsg": "ok"
  /*ro, opt, string, 错误信息, desc:当statusCode不为1时, 必须返回,解释信息在协议约束中, 允许设备在后续的版本迭代中, 进行优化丰富提升（不限制死）*/
}
```

69.57.2 获取访客凭条信息配置

Request URL

GET /ISAPI/AccessControl/visitorReceiptInfoCfg?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "VisitorReceiptInfoCfg": {
    /*ro, req, object, 访客凭条信息配置*/
    "VisitorReceiptInfoList": [
      /*ro, opt, array, 访客凭条信息列表, subType:object, desc:只是规定了打印项的顺序, 不涉及打印项的内容*/
      {
        "id": 1,
        /*ro, req, int, 顺序号, desc:从1开始*/
        "enabled": true,
        /*ro, req, bool, 是否打印使能*/
        "content": "QRCode"
        /*ro, opt, enum, 凭条内容, subType:string, [QRCode#权限二维码,picture#访客照片,name#访客姓名,IDNum#身份证号,address#访客地址,time#有效日期,receptionist#接待人员,receptionistDepartment#接待人员部门,mattersNeedAttention#注意事项,reason#来访事由,visitTime#实际来访时间,appointmentLeaveTime#预约离开时间,visitorRightList#访客权限组列表]*/
      }
    ],
    "mattersNeedAttention": "test"
    /*ro, opt, string, 注意事项, range:[0,64], desc:平台可自定义注意事项的内容, 比如禁止吸烟、禁止拍照等。*/
  }
}
```

69.57.3 获取访客凭条信息配置能力

Request URL

GET /ISAPI/AccessControl/visitorReceiptInfoCfg/capabilities?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "VisitorReceiptInfoCfgCap": {
    /*ro, req, object, 访客凭条信息配置能力*/
    "VisitorReceiptInfoList": {
      /*ro, opt, object, 访客凭条信息列表*/
      "id": {
        /*ro, req, object, 顺序号范围, desc:从1开始*/
        "@min": 1,
        /*ro, req, int, 最小值, range:[1,32]*/
        "@max": 16
        /*ro, req, int, 最大值, range:[1,32]*/
      },
      "enabled": {
        /*ro, req, object, 是否打印使能, desc:true-打印（默认）， false-不打印*/
        "@opt": [true, false]
        /*ro, req, array, 可选项, subType:bool*/
      },
      "content": {
        /*ro, opt, object, 凭条内容*/
        "@opt": ["QRCode", "picture", "name", "IDNum", "address", "time", "receptionist", "receptionistDepartment", "mattersNeedAttention",
          "reason", "visitTime", "appointmentLeaveTime", "visitorRightList"]
        /*ro, req, array, 可选项, subType:string, desc:枚举值含义见参数获取、配置协议*/
      }
    },
    "mattersNeedAttention": {
      /*ro, opt, object, 注意事项*/
      "@min": 0,
      /*ro, req, int, 最小值, range:[0,64]*/
      "@max": 64
      /*ro, req, int, 最大值, range:[0,64]*/
    }
  }
}
```

69.58 访客登记

69.58.1 配置访客参数

Request URL

PUT /ISAPI/AccessControl/visitorParamCfg?format=json

查询参数

无

请求报文

```
{
  "VisitorParamCfg": {
    /*req, object, 访客参数配置*/
    "fillInInfoEnabled": true,
    /*opt, bool, 是否允许在访客界面手动填写访客个人信息*/
    "rightAuthType": "QRCode",
    /*opt, enum, 权限凭证类型, subType:string, [QRCode#二维码（默认）, card#卡片, QRCodeAndCard#二维码+卡片, none#无]*/
    "certificatePictureEnabled": true,
    /*opt, bool, 访客凭条是否使用证件照片*/
    "noAppointmentRegisterEnabled": true,
    /*opt, bool, 非预约访客是否允许直接登记*/
    "visitorInfoMemoryEnabled": true,
    /*opt, bool, 访客记忆功能是否启用*/
    "autoSignOffEnabled": true,
    /*opt, bool, 自动签离是否启用*/
    "skipCertificateCompareEnabled": true,
    /*opt, bool, 是否允许手动跳过人证比对*/
    "authCodeLength": 4,
    /*opt, enum, 访客预约认证码长度, subType:int, [4#4位, 5#5位, 6#6位]*/
    "abnormalTempRegistrationEnabled": true
    /*opt, bool, 访客测温异常是否结束访客登记, desc:true-是, false-否*/
  }
}
```

响应报文

```
{
  "statusCode": 1,
  /*ro, opt, int, 状态码, desc:无法用1表示时（1表示成功且无特殊状态）必须返回*/
  "statusString": "OK",
  /*ro, opt, string, 状态描述, range:[1,64], desc:无法用OK表示时（OK表示成功且无特殊状态）必须返回*/
  "subStatusCode": "ok",
  /*ro, opt, string, 子状态码, range:[1,64], desc:无法用ok表示（ok表示成功且无特殊状态）必须返回*/
  "errorCode": 1,
  /*ro, opt, int, 错误码, desc:当statusCode不为1时，与subStatusCode对应*/
  "errorMsg": "ok"
  /*ro, opt, string, 错误信息, desc:当statusCode不为1时，必须返回,解释信息在协议约束中，允许设备在后续的版本迭代中，进行优化丰富提升（不限制死）*/
}
```

69.58.2 获取访客参数

Request URL

GET /ISAPI/AccessControl/visitorParamCfg?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "VisitorParamCfg": {
    /*ro, req, object, 访客参数配置*/
    "fillInInfoEnabled": true,
    /*ro, opt, bool, 是否允许在访客界面手动填写访客个人信息*/
    "rightAuthType": "QRCode",
    /*ro, opt, enum, 权限凭证类型, subType:string, [QRCode#二维码（默认）,card#卡片,QRCodeAndCard#二维码+卡片,none#无]*/
    "certificatePictureEnabled": true,
    /*ro, opt, bool, 访客凭条是否使用证件照片*/
    "noAppointmentRegisterEnabled": true,
    /*ro, opt, bool, 非预约访客是否允许直接登记*/
    "visitorInfoMemoryEnabled": true,
    /*ro, opt, bool, 访客记忆功能是否启用*/
    "autoSignOffEnabled": true,
    /*ro, opt, bool, 自动签离是否启用*/
    "skipCertificateCompareEnabled": true,
    /*ro, opt, bool, 是否允许手动跳过人证比对*/
    "authCodeLength": 4,
    /*ro, opt, enum, 访客预约认证码长度, subType:int, [4#4位,5#5位,6#6位]*/
    "abnormalTempRegistrationEnabled": true,
    /*ro, opt, bool, 访客测温异常是否结束访客登记, desc:true-是, false-否*/
  }
}
```

69.58.3 获取访客参数配置能力

Request URL

GET /ISAPI/AccessControl/visitorParamCfg/capabilities?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "VisitorParamCfgCap": {
    /*ro, req, object, 访客参数配置能看*/
    "fillInInfoEnabled": {
      /*ro, opt, object, 是否允许在访客界面手动填写访客个人信息, desc:true-允许（默认）， false-不允许（即只能通过读取证件获取相关信息，不能手动填写）*/
      "@opt": [true, false]
      /*ro, opt, array, 可选项, subType:bool*/
    },
    "rightAuthType": {
      /*ro, opt, object, 权限凭证类型, desc:QRCode-二维码（默认）， card-卡片, QRCodeAndCard-二维码+卡片, none-无*/
      "@opt": ["QRCode", "card", "QRCodeAndCard", "none"]
      /*ro, opt, array, 可选项, subType:string*/
    },
    "certificatePictureEnabled": {
      /*ro, opt, object, 访客凭条是否使用证件照片, desc:true-使用（默认。如该访客已刷证件，则使用证件照片；如该访客未刷证件，则使用抓拍照片）， false-不使用（直接使用抓拍照片）*/
      "@opt": [true, false]
      /*ro, opt, array, 可选项, subType:bool*/
    },
    "noAppointmentRegisterEnabled": {
      /*ro, opt, object, 非预约访客是否允许直接登记, desc:针对自助访客机（true-允许（默认）， false-不允许（先要在访客机进行公司内部员工信息校验，待平台将员工信息校验通过后，才能进行非预约访客登记））*/
      "@opt": [true, false]
      /*ro, opt, array, 可选项, subType:bool*/
    },
    "visitorInfoMemoryEnabled": {
      /*ro, opt, object, 访客记忆功能是否启用, desc:true-启用（默认。在访客登记过程中，如发现设备内已存储过该访客的历史信息，则该访客在本次登记时自动补充相关信息）， false-不启用*/
      "@opt": [true, false]
      /*ro, opt, array, 可选项, subType:bool*/
    },
    "autoSignOffEnabled": {
      /*ro, opt, object, 自动签离是否启用, desc:true-启用（默认。在到达访客离开时间点时，若访客没有签离，则在访客离开时间当天24点，自动签离）， false-不启用*/
      "@opt": [true, false]
      /*ro, opt, array, 可选项, subType:bool*/
    },
    "skipCertificateCompareEnabled": {
      /*ro, opt, object, 是否允许手动跳过人证比对, desc:针对非自助访客机（true-允许（在开启了人证比对功能后，设备界面展示跳过按钮，如访客没带证件或人证比对失败，工作人员可人工核验并跳过人证比对，进行访客登记）， false-不允许（默认））*/
      "@opt": [true, false]
      /*ro, opt, array, 可选项, subType:bool*/
    },
    "authCodeLength": {
      /*ro, opt, object, 访客预约认证码长度, desc:支持4-6位纯数字，默认4位*/
      "@opt": [4, 5, 6]
      /*ro, opt, array, 可选项, subType:int*/
    },
    "abnormalTempRegistrationEnabled": {
      /*ro, opt, object, 访客测温异常是否结束访客登记*/
      "@opt": [true, false]
      /*ro, opt, array, 可选项, subType:bool, desc:true-是, false-否*/
    }
  }
}
```

69.58.4 配置无证比对参数

Request URL

PUT /ISAPI/AccessControl/uncertificateCompareCfg?format=json&security= <security>&iv= <iv>

查询参数

参数名称	参数类型	描述
security	string	没有security表示数据不加密； security=1,表示报文中的appKey、appSecret、bizCode字段AES128加密； security=2,表示报文中的appKey、appSecret、bizCode字段AES256加密；
iv	string	随机字符串，长度限制为32； iv表示初始化向量，在security=1或者2时为必须参数;iv举例： FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF

请求报文

```
{
  "enabled": true,
  /*req, bool, 无证比对使能*/
  "verifyMode": "nameAndIDCardAndFace",
  /*opt, enum, 验证模式, subType:string, [nameAndIDCardAndFace#姓名+身份证+抓拍照（人脸）]*/
  "appKey": "test",
  /*opt, string, 萤石应用账号, range:[1,64],
  desc:1、在无证比对开始前，需要先设置appKey、appSecret参数，且appKey、appSecret是用户向萤石申请的，用于标识一台设备，利用appKey、appSecret可以获得
accessToken，作为无证比对的一个参数。
2、由于无证比对功能是萤石平台对接公安系统，是收费业务，收费要通过账号密码唯一标识。
3、appKey仅由字母、数字组成，且字母区分大小写。*/
  "appSecret": "test",
  /*wo, opt, string, 萤石应用密码, range:[1,64], desc:appSecret仅由字母、数字组成，且字母区分大小写。*/
  "bizCode": "test",
  /*opt, string, 萤石商务应用码, range:[1,64],
  desc:1、不同的萤石应用账号对应的bizCode不一样，且不同的萤石商务应用对应的bizCode也不一样，该字段用于区分不同的萤石业务，设备往萤石请求的URL中会携带该信
息，也需要敏感信息加密。
2、请求URL示例：api/component/certificate/alp/ccp?bizCode=xxxx&accessToken=xxxx*/
  "compareThreshold": 80
  /*opt, int, 比对阈值, range:[0,100], desc:事先设定阈值，当公安平台经萤石返回给设备的相似度（抓拍人脸与身份证照片）低于阈值时，无证比对失败*/
}
```

响应报文

```
{
  "statusCode": 1,
  /*ro, opt, int, 状态码, desc:无法用1表示时（1表示成功且无特殊状态）必须返回*/
  "statusString": "OK",
  /*ro, opt, string, 状态描述, range:[1,64], desc:无法用OK表示时（OK表示成功且无特殊状态）必须返回*/
  "subStatusCode": "OK",
  /*ro, opt, string, 子状态码, range:[1,64], desc:无法用ok表示（ok表示成功且无特殊状态）必须返回*/
  "errorCode": 1,
  /*ro, opt, int, 错误码, desc:当statusCode不为1时，与subStatusCode对应*/
  "errorMsg": "ok"
  /*ro, opt, string, 错误信息, desc:当statusCode不为1时，必须返回,解释信息在协议约束中，允许设备在后续的版本迭代中，进行优化丰富提升（不限制死）*/
}
```

69.58.5 获取无证比对参数

Request URL

GET /ISAPI/AccessControl/uncertificateCompareCfg?format=json&security=<security>&iv=<iv>

查询参数

参数名称	参数类型	描述
security	string	没有security表示数据不加密； security=1,表示报文中的appKey、appSecret、bizCode字段AES128加密； security=2,表示报文中的appKey、appSecret、bizCode字段AES256加密；
iv	string	随机字符串，长度限制为32； iv表示初始化向量，在security=1或者2时为必须参数; iv举例：FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF

请求报文

无

响应报文

```

{
  "enabled": true,
  /*ro, req, bool, 无证比对使能*/
  "verifyMode": "nameAndIDCardAndFace",
  /*ro, opt, enum, 验证模式, subType:string, [nameAndIDCardAndFace#姓名+身份证+抓拍照（人脸）]*/
  "appKey": "test",
  /*ro, opt, string, 萤石应用账号, range:[1,64],
  desc:1、在无证比对开始前，需要先设置appKey、appSecret参数，且appKey、appSecret是用户向萤石申请的，用于标识一台设备，利用appKey、appSecret可以获取到
accessToken，作为无证比对的一个参数。
2、由于无证比对功能是萤石平台对接公安系统，是收费业务，收费要通过账号密码唯一标识。
3、appKey仅由字母、数字组成，且字母区分大小写。*/
  "bizCode": "test",
  /*ro, opt, string, 萤石商务应用码, range:[1,64],
  desc:1、不同的萤石应用账号对应的bizCode不一样，且不同的萤石商务应用对应的bizCode也不一样，该字段用于区分不同的萤石业务，设备往萤石请求的URL中会携带该信
息，也需要敏感信息加密。
2、请求URL示例: api/component/certificate/alp/ccp?bizCode=xxxxx&accessToken=xxxxx*/
  "compareThreshold": 80
  /*ro, opt, int, 比对阈值, range:[0,100], desc:事先设定阈值，当公安平台经萤石返回给设备的相似度（抓拍人脸与身份证照片）低于阈值时，无证比对失败*/
}

```

69.58.6 获取无证比对参数配置能力

Request URL

GET /ISAPI/AccessControl/uncertificateCompareCfg/capabilities?format=json

查询参数

无

请求报文

无

响应报文

```

{
  "enabled": {
    /*ro, req, object, 无证比对使能*/
    "@opt": [true, false]
    /*ro, req, array, 可选项, subType:bool, desc:true-开启, false-关闭*/
  },
  "verifyMode": {
    /*ro, opt, object, 验证模式*/
    "@opt": ["nameAndIDCardAndFace"]
    /*ro, req, array, 可选项, subType:string, desc:nameAndIDCardAndFace-姓名+身份证+抓拍照（人脸）*/
  },
  "appKey": {
    /*ro, opt, object, 萤石应用账号,
    desc:1、在无证比对开始前，需要先设置appKey、appSecret参数，且appKey、appSecret是用户向萤石申请的，用于标识一台设备，利用appKey、appSecret可以获取到
accessToken，作为无证比对的一个参数。
2、由于无证比对功能是萤石平台对接公安系统，是收费业务，收费要通过账号密码唯一标识。
3、appKey仅由字母、数字组成，且字母区分大小写。*/
    "@min": 1,
    /*ro, req, int, 最小值, range:[1,64]*/
    "@max": 64
    /*ro, opt, int, 最大值, range:[1,64]*/
  },
  "appSecret": {
    /*ro, opt, object, 萤石应用密码, desc:appSecret仅由字母、数字组成，且字母区分大小写。*/
    "@min": 1,
    /*ro, opt, int, 最小值, range:[1,64]*/
    "@max": 64
    /*ro, opt, int, 最大值, range:[1,64]*/
  },
  "bizCode": {
    /*ro, opt, object, 萤石商务应用码,
    desc:1、不同的萤石应用账号对应的bizCode不一样，且不同的萤石商务应用对应的bizCode也不一样，该字段用于区分不同的萤石业务，设备往萤石请求的URL中会携带该信
息，也需要敏感信息加密。
2、请求URL示例: api/component/certificate/alp/ccp?bizCode=xxxxx&accessToken=xxxxx*/
    "@min": 1,
    /*ro, opt, int, 最小值, range:[1,64]*/
    "@max": 64
    /*ro, opt, int, 最大值, range:[1,64]*/
  },
  "compareThreshold": {
    /*ro, opt, object, 比对阈值*/
    "@min": 0,
    /*ro, opt, int, 最小值, range:[0,100]*/
    "@max": 100
    /*ro, opt, int, 最大值, range:[0,100]*/
  }
}

```

69.58.7 查询无证比对事件

Request URL

POST /ISAPI/AccessControl/searchUncertificateCompareEvent?format=json

查询参数

无

请求报文

```
{
  "searchID": "test",
  /*req, string, 搜索记录唯一标识, range:[1,32], desc:用来确认上层客户端是否为同一个（倘若是同一个，设备记录内存，下次搜索加快速度）*/
  "searchResultPosition": 1,
  /*req, int, 查询结果在结果列表中的起始位置, desc:从1开始，当记录条数很多时，一次查询不能获取所有的记录，下一次查询时指定位置可以查询后面的记录（若设备支持的最大totalMatches为M个，但是当前设备已存储的totalMatches为N个（N<=M），则该字段的合法范围为1~N）*/
  "maxResults": 30,
  /*req, int, 本次协议调用可获取的最大记录数, range:[1,32], desc:如maxResults值大于设备能力集返回的范围，则设备按照能力集最大值返回，设备不进行报错*/
  "type": "success",
  /*opt, enum, 事件类型, subType:string, [success#成功,failure#失败], desc:不下发该字段，代表查询全部（即成功和失败两种事件）*/
  "startTime": "1970-01-01T00:00:00+08:00",
  /*opt, datetime, 开始时间（UTC时间）*/
  "endTime": "1970-01-01T00:00:00+08:00",
  /*opt, datetime, 结束时间（UTC时间）*/
  "picEnable": true,
  /*opt, bool, 是否带图片, desc:false-不带图片，true-带图片（1.配置为false，所有符合条件的事件不带图片上报。2.配置为true，所有符合条件的事件如无联动图片，则上报事件本身；符合条件的事件如有联动图片，则上报事件+联动图片。3.该字段如未配置，则设备默认按照true返回。）*/
  "beginSerialNo": 1,
  /*opt, int, 起始流水号*/
  "endSerialNo": 1,
  /*opt, int, 结束流水号*/
}
```

响应报文

```
{
  "responseStatusStrg": "OK",
  /*ro, req, enum, 查询状态, subType:string, [OK#查询结束,MORE#还有数据等待查询,NO_MATCH#没有匹配数据]*/
  "numOfMatches": 1,
  /*ro, req, int, 本次返回的记录条数, range:[1,32]*/
  "totalMatches": 1,
  /*ro, req, int, 符合条件的记录总条数*/
  "MatchResults": [
    /*ro, opt, array, 搜索结果, subType:object, range:[0,10]*/
    {
      "IDCardInfo": {
        /*ro, opt, object, 身份证信息*/
        "name": "test",
        /*ro, opt, string, 姓名, range:[1,64]*/
        "IDCardNo": "test",
        /*ro, opt, string, 身份证号码, range:[1,30]*/
      },
      "serialNo": 1,
      /*ro, opt, int, 事件流水号, range:[1,3000000000]*/
      "thermometryUnit": "celsius",
      /*ro, opt, enum, 测温单位, subType:string, [celsius#摄氏度（默认）,fahrenheit#华氏度,kelvin#开尔文]*/
      "currTemperature": 36.1,
      /*ro, opt, float, 人脸温度, desc:精确到小数点后一位*/
      "mask": "unknown",
      /*ro, opt, enum, 是否戴口罩, subType:string, [unknown#未知,yes#戴口罩,no#不戴口罩]*/
      "similarity": 90,
      /*ro, opt, int, 相似度, range:[0,100], desc:公安平台存有每个人的身份证照片，该相似度指的是设备上传的人脸抓拍照与公安系统存有的身份证照片的相似度。*/
      "result": "success",
      /*ro, opt, enum, 比对结果, subType:string, [success#成功,failure#失败], desc:黑名单列表是在设备本地维护，当萤石返回给设备是比对成功，设备需要做黑名单校验（前提是黑名单校验功能已开启），如果是黑名单里面的人员，会上报无证比对成功事件，同时上报身份证黑名单事件。*/
      "reason": "incorrectIDCardInfo",
      /*ro, opt, enum, 比对失败原因, subType:string, [incorrectIDCardInfo#输入的身份证信息有误,IDCardInfoAndFacePicMismatch#输入的身份证信息与抓拍人像不匹配,IDCardNotExist#公安系统无此身份证信息,numberOfRequestsExceedLimit#超过最大请求次数,requestTimedOut#请求超时,unknownError#未知错误,IDCardAndFacePicLowSimilarity#身份证照片和抓拍人脸相似度不够],
      desc:1、当result为failure时有效。
      2、IDCardInfoAndFacePicMismatch，是身份证和错误人脸比对；IDCardAndFacePicLowSimilarity，是身份证和本人人脸比对，但是相似度太低（比如：人脸图像位置不对或模糊的情况）。*/
      "pictureURL": "test12345",
      /*ro, opt, string, 人脸抓拍图片URL, range:[1,256]*/
      "dateTime": "1970-01-01T00:00:00+08:00",
      /*ro, req, datetime, 报警触发时间（UTC时间），desc:最大长度为32*/
    }
  ]
}
```

69.58.8 获取查询无证比对事件参数能力

Request URL

GET /ISAPI/AccessControl/searchUncertificateCompareEvent/capabilities?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "UncertificateCompareEventCond": {
    /*ro, opt, object, 无证比对事件查询条件*/
    "searchID": {
      /*ro, req, object, 搜索记录唯一标识, desc:用来确认上层客户端是否为同一个（倘若是一个，设备记录内存，下次搜索加快速度）*/
      "@min": 1,
      /*ro, opt, int, 最小值*/
      "@max": 1
      /*ro, opt, int, 最大值*/
    },
    "searchResultPosition": {
      /*ro, req, object, 查询结果在结果列表中的起始位置, desc:从1开始，当记录条数很多时，一次查询不能获取所有的记录，下一次查询时指定位置可以查询后面的记录（若设备支持的最大totalMatches为M个，但是当前设备已存储的totalMatches为N个（N<=M），则该字段的合法范围为1~N）*/
      "@min": 1,
      /*ro, opt, int, 最小值*/
      "@max": 1
      /*ro, opt, int, 最大值*/
    },
    "maxResults": {
      /*ro, req, object, 本次协议调用可获取的最大记录数, desc:如maxResults值大于设备能力集返回的范围，则设备按照能力集最大值返回，设备不进行报错*/
      "@min": 1,
      /*ro, opt, int, 最小值*/
      "@max": 1
      /*ro, opt, int, 最大值*/
    },
    "type": {
      /*ro, opt, object, 事件类型*/
      "@opt": ["success", "failed"]
      /*ro, opt, array, 可选项, subType:string,
      desc:succcess-成功, failed-失败
      不下发该字段，代表查询全部（即成功和失败两种）*/
    },
    "startTime": "1970-01-01T00:00:00+08:00",
    /*ro, opt, datetime, 开始时间（UTC时间）*/
    "endTime": "1970-01-01T00:00:00+08:00",
    /*ro, opt, datetime, 结束时间（UTC时间）*/
    "picEnable": {
      /*ro, opt, object, 是否带图片, desc:false-不带图片, true-带图片（1.配置为false，所有符合条件的事件不带图片上报。2.配置为true，所有符合条件的事件如无联动图片，则上报事件本身；符合条件的事件如有联动图片，则上报事件+联动图片。3.该字段如未配置，则设备默认按照true返回。）*/
      "@opt": [true, false]
      /*ro, opt, array, 可选项, subType:bool*/
    },
    "beginSerialNo": {
      /*ro, opt, object, 起始流水号*/
      "@min": 1,
      /*ro, opt, int, 最小值*/
      "@max": 1
      /*ro, opt, int, 最大值*/
    },
    "endSerialNo": {
      /*ro, opt, object, 结束流水号*/
      "@min": 1,
      /*ro, opt, int, 最小值*/
      "@max": 1
      /*ro, opt, int, 最大值*/
    }
  },
  "MatchResults": {
    /*ro, opt, object, 事件信息*/
    "IDCardInfo": {
      /*ro, opt, object, 身份证信息*/
      "name": {
        /*ro, opt, object, 姓名*/
        "@min": 1,
        /*ro, opt, int, 最小值*/
        "@max": 64
        /*ro, opt, int, 最大值*/
      },
      "IDCardNo": {
        /*ro, opt, object, 身份证号码*/
        "@min": 1,
        /*ro, opt, int, 最小值*/
        "@max": 30
        /*ro, opt, int, 最大值*/
      }
    },
    "serialNo": {
```

```

/*ro, opt, object, 事件流水号*/
"@min": 1,
/*ro, opt, int, 最小值*/
"@max": 1
/*ro, opt, int, 最大值*/
},
"thermometryUnit": {
/*ro, opt, object, 测温单位*/
"@opt": ["celsius", "fahrenheit", "kelvin"]
/*ro, opt, array, 可选项, subType:string, desc:celsius#摄氏度（默认）,fahrenheit#华氏度,kelvin#开尔文*/
},
"currTemperature": {
/*ro, opt, object, 人脸温度, desc:精确到小数点后一位*/
"@min": 1.0,
/*ro, opt, float, 最小值*/
"@max": 50.0
/*ro, opt, float, 最大值*/
},
"mask": {
/*ro, opt, object, 是否戴口罩*/
"@opt": ["unknown", "yes", "no"]
/*ro, opt, array, 可选项, subType:string, desc:unknown#未知,yes#戴口罩,no#不戴口罩*/
},
"similarity": {
/*ro, opt, object, 相似度*/
"@min": 0,
/*ro, opt, int, 最小值*/
"@max": 100
/*ro, opt, int, 最大值*/
},
"result": {
/*ro, opt, object, 结果*/
"@opt": ["success", "failed"]
/*ro, opt, array, 可选项, subType:string, desc:succcess-成功, failed-失败*/
},
"reason": {
/*ro, opt, object, 失败原因*/
"@opt": ["incorrectIDCardInfo", "IDCardInfoAndFacePicMismatch", "IDCardNotExist", "numberOfRequestsExceedLimit", "requestTimedOut",
"unknownError", "IDCardAndFacePicLowSimilarity"]
/*ro, opt, array, 可选项, subType:string, desc:incorrectIDCardInfo#输入的身份证信息有误, IDCardInfoAndFacePicMismatch#输入的身份证信息与抓拍人像
不匹配, IDCardNotExist#公安系统无此身份证信息,numberOfRequestsExceedLimit#超过最大请求次数, requestTimedOut#请求超时, unknownError#未知错
误, IDCardAndFacePicLowSimilarity#身份证照片和抓拍人脸相似度不够*/
},
"pictureURL": {
/*ro, opt, object, 人脸抓拍图片url*/
"@min": 1,
/*ro, opt, int, 最小值*/
"@max": 256
/*ro, opt, int, 最大值*/
},
"dateTime": "1970-01-01T00:00:00+08:00"
/*ro, opt, datetime, 报警触发时间（UTC时间），desc:最大长度为32*/
}
}

```

69.58.9 获取访客信息查询事件确认能力

Request URL

GET /ISAPI/AccessControl/visitorInfoSearchEventConfirm/capabilities?format=json

查询参数

无

请求报文

无

响应报文

```

{
  "VisitorInfoSearchEventConfirmCap": {
    /*ro, req, object, 访客信息查询事件确认能力*/
    "searchID": {
      /*ro, opt, object, 搜索记录唯一标识长度*/
      "@min": 1,
      /*ro, opt, int, 最小值, range:[1,32]*/
      "@max": 32
      /*ro, opt, int, 最大值, range:[1,32]*/
    },
    "responseStatusStrg": {
      /*ro, opt, object, 查询状态字符串描述*/
      "@opt": ["MORE", "NO MATCH", "OK"]
      /*ro, opt, array, 可选项, subType:string*/
    },
    "numOfMatches": {
      /*ro, opt, object, 本次返回的记录条数数目*/
      "@min": 1
    }
  }
}

```

```

    @min": 1,
    /*ro, opt, int, 最小值, range:[1,100]*/
    "max": 100
    /*ro, opt, int, 最大值, range:[1,100]*/
  },
  "totalMatches": {
    /*ro, opt, object, 符合条件的记录总条数数目*/
    "@min": 1,
    /*ro, opt, int, 最小值, range:[1,100]*/
    "max": 100
    /*ro, opt, int, 最大值, range:[1,100]*/
  },
  "VisitorInfo": {
    /*ro, opt, object, 访客信息*/
    "uuid": {
      /*ro, opt, object, UUID, desc:32字节（由设备生成，全局唯一标识符，访客一次登记/签离/预约唯一）*/
      "@min": 1,
      /*ro, opt, int, 最小值, range:[1,32]*/
      "max": 32
      /*ro, opt, int, 最大值, range:[1,32]*/
    },
    "name": {
      /*ro, opt, object, 姓名长度*/
      "@min": 1,
      /*ro, opt, int, 最小值, range:[1,64]*/
      "max": 64
      /*ro, opt, int, 最大值, range:[1,64]*/
    },
    "certificateType": {
      /*ro, opt, object, 证件类型, desc:ID-二代身份证, thirdID-三代身份证, HK&Macao&TW-港澳台, permanentResidentCard-绿卡, passport-护照, temporaryCard-临时卡, driverLicense-驾驶证*/
      "@opt": ["ID", "thirdID", "HK&Macao&TW", "permanentResidentCard", "passport", "temporaryCard", "driverLicense"]
      /*ro, opt, array, 可选项, subType:string*/
    },
    "IDNum": {
      /*ro, opt, object, 证件号长度*/
      "@min": 1,
      /*ro, opt, int, 最小值, range:[1,128]*/
      "max": 128
      /*ro, opt, int, 最大值, range:[1,128]*/
    },
    "authCode": {
      /*ro, opt, object, 验证码长度*/
      "@opt": [4, 5, 6]
      /*ro, opt, array, 可选项, subType:int, range:[4,6]*/
    },
    "phoneNo": {
      /*ro, opt, object, 手机号长度*/
      "@min": 1,
      /*ro, opt, int, 最小值, range:[1,12]*/
      "max": 12
      /*ro, opt, int, 最大值, range:[1,12]*/
    },
    "cardNo": {
      /*ro, opt, object, 卡号长度*/
      "@min": 1,
      /*ro, opt, int, 最小值, range:[1,32]*/
      "max": 32
      /*ro, opt, int, 最大值, range:[1,32]*/
    },
    "ORCodeEnabled": {
      /*ro, opt, object, 是否生成二维码*/
      "@opt": [true, false]
      /*ro, opt, array, 可选项, subType:bool*/
    },
    "registerType": {
      /*ro, opt, object, 登记类型, desc:appointment-预约, noAppointment-非预约, goTogether-预约随行*/
      "@opt": ["appointment", "noAppointment", "goTogether"]
      /*ro, opt, array, 可选项, subType:string*/
    },
    "registerStatus": {
      /*ro, opt, object, 人员登记状态, desc:registered-已登记, notRegister-未登记, signOff-已签离, overdueAutoSignOff-超期自动签离, overdueNotRegister-超期未登记（预约访客预约当天没有进行登记，登记作废）*/
      "@opt": ["registered", "notRegister", "signOff", "overdueAutoSignOff", "overdueNotRegister"]
      /*ro, opt, array, 可选项, subType:string*/
    },
    "reason": {
      /*ro, opt, object, 来访事由字符串长度*/
      "@min": 1,
      /*ro, opt, int, 最小值, range:[1,120]*/
      "max": 120,
      /*ro, opt, int, 最大值, range:[1,120]*/
    },
    "isSupportVisitTime": true,
    /*ro, opt, bool, 是否支持实际来访时间*/
    "isSupportLeaveTime": true,
    /*ro, opt, bool, 是否支持实际离开时间*/
    "isSupportAppointmentVisitTime": true,
    /*ro, opt, bool, 是否支持预约来访时间*/
    "isSupportAppointmentLeaveTime": true,
    /*ro, opt, bool, 是否支持预约离开时间*/
    "receptionistName": {
      /*ro, opt, object, 接待人员姓名长度*/
      "@min": 1,

```

```

/*ro, opt, int, 最小值, range:[1,64]*/
"@max": 64
/*ro, opt, int, 最大值, range:[1,64]*/
},
"receptionistDepartment": {
/*ro, opt, object, 接待人员部门长度*/
"@min": 1,
/*ro, opt, int, 最小值, range:[1,256]*/
"@max": 256
/*ro, opt, int, 最大值, range:[1,256]*/
},
"plateNo": {
/*ro, opt, object, 车牌号长度*/
"@min": 1,
/*ro, opt, int, 最小值, range:[1,12]*/
"@max": 12
/*ro, opt, int, 最大值, range:[1,12]*/
},
"company": {
/*ro, opt, object, 访客单位长度*/
"@min": 1,
/*ro, opt, int, 最小值, range:[1,120]*/
"@max": 120
/*ro, opt, int, 最大值, range:[1,120]*/
},
"belongings": {
/*ro, opt, object, 携带物品长度*/
"@min": 1,
/*ro, opt, int, 最小值, range:[1,120]*/
"@max": 120
/*ro, opt, int, 最大值, range:[1,120]*/
},
"address": {
/*ro, opt, object, 访客地址*/
"@min": 1,
/*ro, opt, int, 最小值, range:[1,240]*/
"@max": 240
/*ro, opt, int, 最大值, range:[1,240]*/
},
"remark": {
/*ro, opt, object, 备注长度*/
"@min": 1,
/*ro, opt, int, 最小值, range:[1,32]*/
"@max": 32
/*ro, opt, int, 最大值, range:[1,32]*/
},
"VisitorRightList": {
/*ro, opt, object, 访客配置的权限列表*/
"uuid": {
/*ro, opt, object, UUID, desc:32字节（由平台生成，平台唯一）*/
"@min": 1,
/*ro, opt, int, 最小值, range:[1,32]*/
"@max": 32
/*ro, opt, int, 最大值, range:[1,32]*/
},
"remark": {
/*ro, opt, object, 名称长度*/
"@min": 1,
/*ro, opt, int, 最小值, range:[1,32]*/
"@max": 32
/*ro, opt, int, 最大值, range:[1,32]*/
}
},
"CustomInfoList": {
/*ro, opt, object, 自定义信息列表*/
"@size": 3,
/*ro, opt, int, 最大个数, range:[1,3]*/
"id": {
/*ro, req, object, 自定义项编号*/
"@min": 1,
/*ro, opt, int, 最小值, range:[1,3]*/
"@max": 3
/*ro, opt, int, 最大值, range:[1,3]*/
},
"customName": {
/*ro, opt, object, 自定义项名称*/
"@min": 1,
/*ro, opt, int, 最小值, range:[1,64]*/
"@max": 64
/*ro, opt, int, 最大值, range:[1,64]*/
},
"customContent": {
/*ro, opt, object, 自定义项内容*/
"@min": 1,
/*ro, opt, int, 最小值, range:[1,64]*/
"@max": 64
/*ro, opt, int, 最大值, range:[1,64]*/
}
},
"thermometryUnit": {
/*ro, opt, object, 测温单位*/
"@opt": ["celsius", "fahrenheit", "kelvin"]
/*ro, opt, array, 可选项, subType:string, desc:celsius-摄氏度（默认）,fahrenheit-华氏度,kelvin-开尔文*/

```

```

    },
    "currTemperature": {
      /*ro, opt, object, 人脸温度*/
      "@min": 1.0,
      /*ro, opt, float, 最小值*/
      "@max": 45.0
      /*ro, opt, float, 最大值*/
    },
    "isAbnormalTemperature": {
      /*ro, opt, object, 人脸测温是否温度异常*/
      "@opt": [true, false]
      /*ro, opt, array, 可选项, subType:bool, desc:true-是, false-否*/
    },
  },
}
}
}

```

69.58.10 获取访客高级参数配置能力

Request URL

GET /ISAPI/AccessControl/visitorAdvanceParamCfg/capabilities?format=json

查询参数

无

请求报文

无

响应报文

```

{
  "VisitorAdvanceParamCfgCap": {
    /*ro, req, object, 访客高级参数配置能力*/
    "autoSyncInfoEnabled": {
      /*ro, opt, object, 是否自动同步访客信息, desc:true-同步（访客机在到达规定的时间间隔（autoSyncInfoTimeInterval），向平台/客户端请求访客登记/签离信息，用于多台访客机间信息同步），false-不同步（默认）*/
      "@opt": [true, false]
      /*ro, opt, array, 可选项, subType:bool*/
    },
    "autoSyncInfoTimeInterval": {
      /*ro, opt, object, 自动同步访客信息时间间隔范围, desc:5分钟-60分钟，默认10分钟*/
      "@min": 5,
      /*ro, opt, int, 最小值*/
      "@max": 60
      /*ro, opt, int, 最大值*/
    },
    "visitorAppointEnabled": {
      /*ro, opt, object, 是否开启访客预约功能, desc:true-开启（访客在平台/客户端完成预约登记后，在访客机上输入预约凭证，访客机向平台/客户端请求预约信息），false-不开启（默认）*/
      "@opt": [true, false]
      /*ro, opt, array, 可选项, subType:bool*/
    },
    "currentEventConfirmEnabled": {
      /*ro, opt, object, 是否开启访客实时登记和签离事件确认功能, desc:true-开启（访客在访客机登记/签离时，是否需要平台/客户端进行确认），false-不开启（默认）*/
      "@opt": [true, false]
      /*ro, opt, array, 可选项, subType:bool*/
    }
  }
}

```

69.58.11 配置访客高级参数

Request URL

PUT /ISAPI/AccessControl/visitorAdvanceParamCfg?format=json

查询参数

无

请求报文

```
{
  "VisitorAdvanceParamCfg": {
    /*req, object, 访客高级参数配置*/
    "autoSyncInfoEnabled": true,
    /*opt, bool, 是否自动同步访客信息*/
    "autoSyncInfoTimeInterval": 5,
    /*opt, int, 自动同步访客信息时间间隔, range:[5,60]*/
    "visitorAppointEnabled": true,
    /*opt, bool, 是否开启访客预约功能*/
    "currentEventConfirmEnabled": true
    /*opt, bool, 是否开启访客实时登记和签离事件确认功能*/
  }
}
```

响应报文

```
{
  "statusCode": 1,
  /*ro, opt, int, 状态码, desc:无法用1表示时（1表示成功且无特殊状态）必须返回*/
  "statusString": "OK",
  /*ro, opt, string, 状态描述, range:[1,64], desc:无法用OK表示时（OK表示成功且无特殊状态）必须返回*/
  "subStatusCode": "ok",
  /*ro, opt, string, 子状态码, range:[1,64], desc:无法用ok表示（ok表示成功且无特殊状态）必须返回*/
  "errorCode": 1,
  /*ro, opt, int, 错误码, desc:当statusCode不为1时，与subStatusCode对应*/
  "errorMsg": "ok"
  /*ro, opt, string, 错误信息, desc:当statusCode不为1时，必须返回,解释信息在协议约束中，允许设备在后续的版本迭代中，进行优化丰富提升（不限制死）*/
}
```

69.58.12 获取访客高级参数

Request URL

GET /ISAPI/AccessControl/visitorAdvanceParamCfg?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "VisitorAdvanceParamCfg": {
    /*ro, req, object, 访客高级参数配置*/
    "autoSyncInfoEnabled": true,
    /*ro, opt, bool, 是否自动同步访客信息*/
    "autoSyncInfoTimeInterval": 5,
    /*ro, opt, int, 自动同步访客信息时间间隔, range:[5,60]*/
    "visitorAppointEnabled": true,
    /*ro, opt, bool, 是否开启访客预约功能*/
    "currentEventConfirmEnabled": true
    /*ro, opt, bool, 是否开启访客实时登记和签离事件确认功能*/
  }
}
```

69.58.13 配置访客权限列表

Request URL

PUT /ISAPI/AccessControl/visitorRightListCfg?format=json

查询参数

无

请求报文

```
{
  "VisitorRightListCfg": {
    /*req, object, 访客权限列表配置*/
    "VisitorRightList": [
      /*opt, array, 访客权限列表, subType:object, range:[1,512], desc:Windows访客机数目固定, 只支持6个权限; 安卓访客机项目权限扩充为512个*/
      {
        "uuid": "test",
        /*opt, string, UUID, range:[1,32]*/
        "remark": "test"
        /*opt, string, 名称, range:[1,256]*/
      }
    ]
  }
}
```

响应报文

```
{
  "statusCode": 1,
  /*ro, opt, int, 状态码, desc:无法用1表示时（1表示成功且无特殊状态）必须返回*/
  "statusString": "OK",
  /*ro, opt, string, 状态描述, range:[1,64], desc:无法用OK表示时（OK表示成功且无特殊状态）必须返回*/
  "subStatusCode": "ok",
  /*ro, opt, string, 子状态码, range:[1,64], desc:无法用ok表示（ok表示成功且无特殊状态）必须返回*/
  "errorCode": 1,
  /*ro, opt, int, 错误码, desc:当statusCode不为1时, 与subStatusCode对应*/
  "errorMsg": "ok"
  /*ro, opt, string, 错误信息, desc:当statusCode不为1时, 必须返回,解释信息在协议约束中, 允许设备在后续的版本迭代中, 进行优化丰富提升（不限制死）*/
}
```

69.58.14 获取访客权限列表配置

Request URL

GET /ISAPI/AccessControl/visitorRightListCfg?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "VisitorRightListCfg": {
    /*ro, req, object, 访客权限列表配置*/
    "VisitorRightList": [
      /*ro, opt, array, 访客权限列表, subType:object, range:[1,512], desc:Windows访客机数目固定, 只支持6个权限; 安卓访客机项目扩充为512个权限*/
      {
        "uuid": "test",
        /*ro, opt, string, UUID, range:[1,32]*/
        "remark": "test"
        /*ro, opt, string, 名称, range:[1,256]*/
      }
    ]
  }
}
```

69.58.15 获取访客权限列表配置能力

Request URL

GET /ISAPI/AccessControl/visitorRightListCfg/capabilities?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "VisitorRightListCfgCap": {
    /*ro, req, object, 访客权限列表配置能力*/
    "VisitorRightList": {
      /*ro, opt, object, 访客权限列表*/
      "@size": 512,
      /*ro, opt, int, 权限最大个数, desc:如果返回了@size, 则通过size来判断权限个数, 如果没有返回, 则默认为6个权限 (与Windows访客机保持一致) */
      "uuid": {
        /*ro, opt, object, UUID*/
        "@min": 1,
        /*ro, opt, int, 最小值*/
        "@max": 32
        /*ro, opt, int, 最大值*/
      },
      "remark": {
        /*ro, opt, object, 名称长度*/
        "@min": 1,
        /*ro, opt, int, 最小值*/
        "@max": 256
        /*ro, opt, int, 最大值*/
      }
    }
  }
}
```

69.58.16 获取访客事件确认能力

Request URL

GET /ISAPI/AccessControl/visitorEventConfirm/capabilities?format=json

查询参数

无

请求报文

无

响应报文

湖南云舟电力科技有限公司
有效期：2024-08-15

```

{
  "VisitorEventConfirmCap": {
    /*ro, req, object, 访客事件确认能力*/
    "uuid": {
      /*ro, opt, object, UUID, desc:32字节（由设备生成，全局唯一标识符，访客一次登记/签离/预约唯一）*/
      "@min": 1,
      /*ro, opt, int, 最小值*/
      "@max": 32
      /*ro, opt, int, 最大值*/
    },
    "result": {
      /*ro, opt, object, 结果, desc:succcess-成功, appointmentNotExist-预约访客不存在, appointmentRegistered-预约访客已登记*/
      "@opt": ["success", "appointmentNotExist", "appointmentRegistered", "visitPurposeNotExist", "visiteeNotExist", "incorrectIDType",
        "numberOfVisitorsExceedLimit", "addingVisitorPictureFailed", "visitorCardNoAlreadyExist", "blocklistVisitorRegNotAllowed",
        "visitorAccessPermissionNotExist", "visitorIDNoAlreadyExist", "editCheckedOutVisitorNotAllowed"]
      /*ro, opt, array, 可选项, subType:string*/
    },
    "QRCodeInfo": {
      /*ro, opt, object, 二维码信息*/
      "cardNo": {
        /*ro, opt, object, 卡号长度*/
        "@min": 1,
        /*ro, opt, int, 最小值*/
        "@max": 32
        /*ro, opt, int, 最大值*/
      },
      "valid": {
        /*ro, opt, object, 二维码有效期范围, desc:单位: 分钟*/
        "@min": 1,
        /*ro, opt, int, 最小值*/
        "@max": 32
        /*ro, opt, int, 最大值*/
      },
      "times": {
        /*ro, opt, object, 二维码最大开锁次数范围*/
        "@min": 1,
        /*ro, opt, int, 最小值*/
        "@max": 4
        /*ro, opt, int, 最大值*/
      },
      "version": {
        /*ro, opt, object, 设备支持的二维码版本, desc:如不返回该字段，则默认支持v1.0版本，参考《二维码概要设计.docx》*/
        "@opt": ["v1.0"]
        /*ro, opt, array, 可选项, subType:string*/
      }
    }
  }
}

```

69.58.17 配置人证比对参数

Request URL

PUT /ISAPI/AccessControl/certificateCompareCfg?format=json

查询参数

无

请求报文

```

{
  "CertificateCompareCfg": {
    /*req, object, 认证比对配置*/
    "threshold": 90,
    /*opt, int, 人脸比对阈值, range:[0,100]*/
    "verifyMode": "face",
    /*opt, enum, 验证方式, subType:string, [face#证件照片和抓拍人脸比对, fp#证件指纹和采集指纹比对, faceOrFp#证件照片和抓拍人脸比对或证件指纹和采集指纹比对, faceAndFp#证件照片和抓拍人脸比对+证件指纹和采集指纹比对, no#无需验证]*/
    "bottomCameraMode": "certificateCapature"
    /*opt, enum, 底部摄像头工作模式, subType:string, [certificateCapature#证件抓拍, QRCodeScan#二维码扫描], desc:桌面人证设备包含有三个摄像头，上面两个用于抓拍人脸，通道1为可见光，通道2为热成像，底部摄像头用于抓拍证件或者二维码扫描*/
  }
}

```

响应报文

```
{
  "statusCode": 1,
  /*ro, opt, int, 状态码, desc:无法用1表示时（1表示成功且无特殊状态）必须返回*/
  "statusString": "OK",
  /*ro, opt, string, 状态描述, range:[1,64], desc:无法用OK表示时（OK表示成功且无特殊状态）必须返回*/
  "subStatusCode": "ok",
  /*ro, opt, string, 子状态码, range:[1,64], desc:无法用ok表示（ok表示成功且无特殊状态）必须返回*/
  "errorCode": 1,
  /*ro, opt, int, 错误码, desc:当statusCode不为1时，与subStatusCode对应*/
  "errorMsg": "ok"
  /*ro, opt, string, 错误信息, desc:当statusCode不为1时，必须返回,解释信息在协议约束中，允许设备在后续的版本迭代中，进行优化丰富提升（不限制死）*/
}
```

69.58.18 获取人证比对参数

Request URL

GET /ISAPI/AccessControl/certificateCompareCfg?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "CertificateCompareCfg": {
    /*ro, req, object, 认证比对配置*/
    "threshold": 90,
    /*ro, opt, int, 人脸比对阈值, range:[0,100]*/
    "verifyMode": "face",
    /*ro, opt, enum, 验证方式, subType:string, [face#证件照片和抓拍人脸比对,fp#证件指纹和采集指纹比对,faceOrFp#证件照片和抓拍人脸比对或证件指纹和采集指纹比对,faceAndFp#证件照片和抓拍人脸比对+证件指纹和采集指纹比对,no#无需验证]*/
    "bottomCameraMode": "certificateCapature"
    /*ro, opt, enum, 底部摄像头工作模式, subType:string, [certificateCapature#证件抓拍,QRCodeScan#二维码扫描], desc:桌面人证设备包含有三个摄像头，上面两个用于抓拍人脸，通道1为可见光，通道2为热成像，底部摄像头用于抓拍证件或者二维码扫描*/
  }
}
```

69.58.19 获取人证比对参数配置能力

Request URL

GET /ISAPI/AccessControl/certificateCompareCfg/capabilities?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "CertificateCompareCfgCap": {
    /*ro, req, object, 人证比对配置能力*/
    "threshold": {
      /*ro, opt, object, 人脸比对阈值范围, desc:范围0-100, 范围不可修改*/
      "@min": 0,
      /*ro, opt, int, 最小值*/
      "@max": 100
      /*ro, opt, int, 最大值*/
    },
    "verifyMode": {
      /*ro, opt, object, 验证方式, desc:face-证件照片和抓拍人脸比对（默认），fp-证件指纹和采集指纹比对，faceOrFp-（证件照片和抓拍人脸比对）或（证件指纹和采集指纹比对），faceAndFp-（证件照片和抓拍人脸比对）+（证件指纹和采集指纹比对），no-无需验证*/
      "@opt": ["face", "fp", "faceOrFp", "faceAndFp", "no"]
      /*ro, opt, array, 可选项, subType:string*/
    },
    "bottomCameraMode": {
      /*ro, opt, object, 底部摄像头工作模式, desc:桌面人证设备包含有三个摄像头，上面两个用于抓拍人脸，通道1为可见光，通道2为热成像，底部摄像头用于抓拍证件或者二维码扫描*/
      "@opt": ["certificateCapature", "QRCodeScan"]
      /*ro, req, array, 可选项, subType:string, desc:certificateCapature-证件抓拍，QRCodeScan-二维码扫描*/
    }
  }
}
```

69.58.20 无证比对事件

事件标识：UncertificateCompareEvent

```
{
  "ipAddress": "172.6.64.7",
  /*ro, req, string, 报警设备IPv4地址*/
  "ipv6Address": "1080:0:0:8:800:200C:417A",
  /*ro, opt, string, 报警设备IPv6地址*/
  "portNo": 80,
  /*ro, opt, int, 报警设备端口号*/
  "protocol": "HTTP",
  /*ro, opt, enum, 传输通信协议类型, subType:string, [HTTP#HTTP,HTTPS#HTTPS,EHome#EHome], desc:传输通信协议HTTP/HTTPS/EHome（萤石透传ISAPI 的时候赋值HTTP；ISUP透传ISAPI 的时候赋值EHome）protocolType作为历史遗留由平台兼容*/
  "macAddress": "01:17:24:45:D9:FA",
  /*ro, opt, string, MAC地址*/
  "channelID": 1,
  /*ro, opt, int, 触发报警的设备通道号, desc:触发的视频通道号（1、在SDK透传ISAPI协议的时候，上传的是私有协议对应的视频通道号；2、在萤石透传ISAPI协议的时候，上传的是萤石协议对应的视频通道号；3、在ISUP透传ISAPI协议的时候，上传的是ISUP协议对应的视频通道号。*/
  "dateTime": "2004-05-03T17:30:08+08:00",
  /*ro, req, datetime, 报警触发时间*/
  "activePostCount": 1,
  /*ro, opt, int, 同一个报警已经上传的次数, desc:事件触发频次脉冲事件 定义：事件持续触发（按照设备的检测频率），例如：移动侦测。瞬时事件 定义：区分目标，一个目标触发一次，例如：人脸识别activePostCount 在脉冲事件 类型触发的时候，用于区分是否是同一触发源触发的事件。例如：移动侦测，按照设备检测频率会一直上传；如果触发源发生了变化，这个时候计数就可以重新开始了。这个可以作为事件触发频次的方式来集成；*/
  "eventType": "UncertificateCompareEvent",
  /*ro, req, string, 事件类型, range:[1,128], desc:UncertificateCompareEvent-无证比对事件*/
  "eventState": "active",
  /*ro, req, enum, 事件状态, subType:string, [active#有效事件,inactive#无效事件], desc:针对持续性事件active - 表示有效事件（开始 或者 无过程状态也使用该字）；inactive - 表示无效事件（结束）；remark:在心跳类型下，该字段赋值（表示心跳数据,10s上传一次）；*/
  "eventDescription": "Uncertificate Compare Event",
  /*ro, req, string, 事件描述, desc:Uncertificate Compare Event-无证比对事件*/
  "deviceId": "test0123",
  /*ro, opt, string, 即PUID, desc:在ISUP协议接入透传ISAPI事件信息中必须返回*/
  "UncertificateCompareEvent": {
    /*ro, opt, object, 无证比对事件*/
    "IDCardInfo": {
      /*ro, opt, object, 身份证信息*/
      "name": "test",
      /*ro, opt, string, 姓名, range:[1,128]*/
      "IDCardNo": "test"
      /*ro, opt, string, 身份证号码, range:[1,30]*/
    },
    "serialNo": 1,
    /*ro, opt, int, 事件流水号, range:[1,3000000000], desc:事件流水号上限一般为30亿*/
    "frontSerialNo": 1,
    /*ro, opt, int, 上一条事件流水号, range:[1,3000000000], desc:若设备没返回该字段，平台根据serialNo判断是否丢失事件；若设备返回该字段，平台根据该字段和serialNo字段共同判断是否丢失事件,主要用于解决报警订阅后导致serialNo不连续的情况*/
    "thermometryUnit": "celsius",
    /*ro, opt, enum, 测温单位, subType:string, [celsius#摄氏度（默认）,fahrenheit#华氏度,kelvin#开尔文]*/
    "currTemperature": 36.1,
    /*ro, opt, float, 手腕温度, desc:精确到小数点后一位，热电堆设备测的是手腕温度*/
    "isAbnormalTemperature": true,
    /*ro, opt, bool, 手腕测温是否温度异常, desc:true-是，false-否*/
    "HealthInfo": {
      /*ro, opt, object, 健康信息*/
      "healthCode": 1,
      /*ro, opt, enum, 健康码状态, subType:int, [0#未请求,1#未申领,2#绿码,3#黄码,4#红码,5#无此人员,6#其他错误信息（如接口异常导致查询失败）,7#查询健康码超时]*/
      "NADCode": 1,
      /*ro, opt, enum, 核酸检测结果, subType:int, [0#未查询到核酸检测结果,1#核酸检测阴性（代表正常）,2#核酸检测阳性（代表确诊）,3#核酸检测有效期已过,4#查询核酸检测失败]*/
      "travelCode": 1,

```

```
/*ro, opt, enum, 行程编码, subType:int, [0#14天内一直在当地,1#14天内离开过当地,2#14天内到过疫区,3#其他,4#查询行程信息失败]*/
"travelInfo": "test",
/*ro, opt, string, 行程信息, desc:空字符串表示查询到行程信息失败*/
"vaccineStatus": 1,
/*ro, opt, enum, 打疫苗状态, subType:int, [0#未打疫苗,1#部分注射疫苗,2#已完成疫苗,3#查询疫苗信息失败,4#已完成加强针疫苗]*/
"vaccineNum": 1
/*ro, opt, int, 打疫苗针数, step:1*/
},
"mask": "unknown",
/*ro, opt, enum, 是否戴口罩, subType:string, [unknown#未知,yes#戴口罩,no#不戴口罩]*/
"similarity": 90,
/*ro, opt, int, 相似度, range:[0,100], desc:公安平台存有每个人的身份证照片, 该相似度指的是设备上传的人脸抓拍照片与公安系统存有的身份证照片的相似度。*/
"result": "success",
/*ro, opt, enum, 比对结果, subType:string, [success#成功,failed#失败], desc:黑名单列表是在设备本地维护, 当萤石返回给设备是比对成功, 设备需要做黑名单校验(前提是黑名单校验功能已开启), 如果是黑名单里面的人员, 会上报无证比对成功事件, 同时上报身份证黑名单事件。*/
"reason": "incorrectIDCardInfo",
/*ro, opt, enum, 比对失败原因, subType:string, [incorrectIDCardInfo#输入的身份证信息有误,IDCardInfoAndFacePicMismatch#输入的身份证信息与抓拍人像不匹配,IDCardNotExist#公安系统无此身份证信息,numberOfRequestsExceedLimit#超过最大请求次数,requestTimedOut#请求超时,unknownError#未知错误,IDCardAndFacePicLowSimilarity#身份证照片和抓拍人脸相似度不够],
desc:1、当result为failed时有效。
2、IDCardInfoAndFacePicMismatch, 是身份证和错误人脸比对; IDCardAndFacePicLowSimilarity, 是身份证和本人人脸比对, 但是相似度太低(比如: 人脸图像位置不对或模糊的情况)。*/
"showFaceCapPic": true,
/*ro, opt, bool, 是否显示人脸抓拍图片, desc:true-显示, false-不显示, 默认为true, 基于保护用户隐私的需求, Hicore上传给UI的人脸抓拍图片可以配置为不显示。*/
"facePicURL": "test12345",
/*ro, opt, string, 人脸抓拍图片URL, range:[1,256]*/
"faceContentID": "FacePic",
/*ro, opt, string, 图片contentID, range:[1,64], desc:直传二进制图片数据时使用,与报文Content-ID对应,人脸抓拍图片使用该字段*/
"picturesNumber": 1
/*ro, opt, int, 图片数量,
desc:后面所带的图片数目, 没图片时, 该字段为0或不返回。
无论比对成功或失败, 均只包含人脸抓拍照片一张图片。*/
}
}
```

参数名称	参数值	参数类型(Content-Type)	Content-ID	文件名称(filename)	描述
UncertificateCompareEvent	[报文内容]	application/json	--	--	--
设备未实现	[图片二进制数据]	image/jpeg	FacePic	设备未实现	人脸抓拍图片

说明：此协议通过表单元格式传输，表单框架详情说明见章节4.5.1.4，如下实例：

```
--<frontier>
Content-Disposition: form-data; name=参数名称; filename=文件名称
Content-Type: 参数类型
Content-Length: ****
Content-ID: 内容ID
参数值
```

- 参数名称：表单元Headers中Content-Disposition的name属性，表示此表单元的名称
- 参数类型（Content-Type）：表单元Headers中Content-Type属性
- 文件名称（filename）：表单元Headers中Content-Disposition的filename属性，仅在本表单元传输数据为文件类型存在，表示表单元Body的文件名
- 参数值：表单元的body内容

69.58.21 证件抓拍比对事件

事件标识：CertificateCaptureEvent

```

{
  "ipAddress": "172.6.64.7",
  /*ro, req, string, 报警设备IPv4地址*/
  "ipv6Address": "1080:0:0:8:800:200C:417A",
  /*ro, opt, string, 报警设备IPv6地址*/
  "portNo": 80,
  /*ro, opt, int, 报警设备端口号*/
  "protocol": "HTTP",
  /*ro, opt, enum, 传输通信协议类型, subType:string, [HTTP#HTTP,HTTPS#HTTPS,EHome#EHome], desc:传输通信协议HTTP/HTTPS/EHome（萤石透传ISAPI 的时候赋值
  HTTP; ISUP透传ISAPI 的时候赋值 EHome）protocolType作为历史遗留由平台兼容*/
  "macAddress": "01:17:24:45:D9:F4",
  /*ro, opt, string, MAC地址*/
  "channelID": 1,
  /*ro, opt, int, 触发报警的设备通道号, desc:触发的视频通道号（1、 在SDK透传ISAPI协议的时候，上传的是 私有协议对应的视频通道号；（2、 在萤石透传
  ISAPI协议的时候，上传的是 萤石协议对应的视频通道号；（3、 在ISUP透传ISAPI协议的时候，上传的是 ISUP协议对应的视频通道号；*/
  "dateTime": "2004-05-03T17:30:08+08:00",
  /*ro, req, datetime, 报警触发时间*/
  "activePostCount": 1,
  /*ro, opt, int, 同一个报警已经上传的次数, desc:事件触发频次脉冲事件 定义：事件持续触发（按照设备的检测频率），例如：移动侦测。瞬时事件 定义：区分目标，
  一个目标触发一次，例如：人脸识别activePostCount 在脉冲事件 类型触发的时候，用于区分是否是同一触发源触发的事件。例如：移动侦测，按照设备检测频率会一直上传；
  如果触发源发生了变化，这个时候计数就可以重新开始了。这个可以作为事件触发频次的方式来集成；*/
  "eventType": "CertificateCaptureEvent",
  /*ro, req, string, 事件类型, range:[1,128]*/
  "eventState": "active",
  /*ro, req, enum, 事件状态, subType:string, [active#有效事件,inactive#无效事件], desc:针对持续性事件active - 表示有效事件（开始 或者 无过程状态也使用该字
  段）；inactive - 表示无效事件（结束）；remark:在心跳类型下，该字段赋值（表示心跳数据,10s上传一次）；*/
  "eventDescription": "Certificate Capture Comparison Event",
  /*ro, req, string, 事件描述*/
  "deviceId": "test0123",
  /*ro, opt, string, 即PUID, desc:在ISUP协议接入透传ISAPI事件信息中必须返回*/
  "CertificateCaptureEvent": {
    /*ro, opt, object, 证件抓拍比对事件*/
    "deviceName": "device1",
    /*ro, opt, string, 设备名称, range:[1,32]*/
    "majorEventType": 1,
    /*ro, req, int, 报警主类型, desc:参考宏定义（此处传递的是10进制数据，不能传递16进制，如传递1代表0x1（报警主类型）），0x5-事件*/
    "subEventType": 2,
    /*ro, req, int, 报警次类型, desc:参考宏定义（此处传递的是10进制数据，不能传递16进制，如传递1024代表0x400（防区短路报警次类型）），（复用人证比对宏定
    义，0x69-人证比对通过，0x70-人证比对失败）*/
    "cardReaderNo": 1,
    /*ro, opt, int, 读卡器编号*/
    "doorNo": 1,
    /*ro, opt, int, 门编号（楼层编号）*/
    "deviceNo": 1,
    /*ro, opt, int, 设备编号*/
    "serialNo": 1,
    /*ro, opt, int, 事件流水号*/
    "currentEvent": true,
    /*ro, opt, bool, 是否为实时事件, desc:true-是（实时事件），false-否（离线事件）*/
    "frontSerialNo": 1,
    /*ro, opt, int, 上一条事件流水号, desc:若设备没返回该字段，平台根据serialNo判断是否丢失事件；若设备返回该字段，平台根据该字段和SerialNo字段共同判断
    是否丢失事件，主要用于解决报警订阅后导致serialNo不连续的情况*/
    "thermometryUnit": "celsius",
    /*ro, opt, enum, 测温单位, subType:string, [celsius#摄氏度（默认）,fahrenheit#华氏度,kelvin#开尔文]*/
    "currTemperature": 36.1,
    /*ro, opt, float, 手腕温度, desc:精确到小数点后一位，热电堆设备测的是手腕温度*/
    "isAbnormalTemperature": true,
    /*ro, opt, bool, 手腕测温是否温度异常, desc:true-是，false-否*/
    "mask": "unknown",
    /*ro, opt, enum, 是否戴口罩, subType:string, [unknown#未知,yes#戴口罩,no#不戴口罩]*/
    "showFaceCapPic": true,
    /*ro, opt, bool, 是否显示人脸抓拍图片, desc:true-显示，false-不显示，默认为true，基于保护用户隐私的需求，Hicore上传给UI的人脸抓拍图片可以配置为不显
    示。*/
    "pictureURL": "test12345",
    /*ro, opt, string, 人脸抓拍图片URL, range:[1,256]*/
    "certificatePicURL": "test12345",
    /*ro, opt, string, 证件抓拍图片URL, range:[1,256]*/
    "contentId": "FacePic",
    /*ro, opt, string, 人脸抓拍图片contentID, range:[1,64], desc:直传二进制图片数据时使用,与报文Content-ID对应,人脸抓拍图片使用该字段*/
    "certificateContentId": "CertificatePic",
    /*ro, opt, string, 证件抓拍图片contentid, range:[1,64], desc:直传二进制图片数据时使用,与报文Content-ID对应,证件抓拍图片使用该字段*/
    "picturesNumber": 2,
    /*ro, opt, int, 图片数量,
    desc:后面所带的图片数目，没图片时，该字段为0或不返回
    （1）比对成功包含2张图片，抓拍证件图片和人脸图片
    （2）比对失败：包含2张图片，抓拍证件图片和人脸图片，或者只有1张抓拍证件图片*/
    "isWearingMaskRequired": true,
    /*ro, opt, bool, 是否需要戴口罩,
    desc:true-需要，false-不需要
    当设备开启口罩检测功能后，在证件抓拍环节，若检测到访客（人员）未佩戴口罩，Hicore需向UI返回该字段，UI进行提示访客（人员）佩戴口罩。*/
  }
}

```

参数名称	参数值	参数类型(Content-Type)	Content-ID	文件名称(filename)	描述
CertificateCaptureEvent	[报文内容]	application/json	--	--	--
设备未实现	[图片二进制数据]	image/jpeg	FacePic	设备未实现	
设备未实现	[图片二进制数据]	image/jpeg	CertificatePic	设备未实现	

说明：此协议通过表单格式传输，表单框架详情说明见章节4.5.1.4，如下实例：

```
--<frontier>
Content-Disposition: form-data; name=参数名称;filename=文件名称
Content-Type: 参数类型
Content-Length: ****
Content-ID: 内容ID
参数值
```

- 参数名称：表单单元Headers中Content-Disposition的name属性，表示此表单单元的名称
- 参数类型（Content-Type）：表单单元Headers中Content-Type属性
- 文件名称（filename）：表单单元Headers中Content-Disposition的filename属性，仅在本表单单元传输数据为文件类型存在，表示表单单元Body的文件名
- 参数值：表单单元的body内容

69.59 访客展示信息管理

69.59.1 配置访客登记信息

Request URL

PUT /ISAPI/AccessControl/visitorRegisterInfoCfg?format=json

查询参数

无

请求报文

```

{
  "VisitorRegisterInfoCfg": {
    /*req, object, 访客登记信息配置*/
    "IDNumDisplay": "displayAndReq",
    /*opt, enum, 身份证号（证件号码）显示方式, subType:string, [displayAndReq#显示必须（默认）,displayAndOpt#显示可选,hide#隐藏]*/
    "desensitiseIDNum": true,
    /*opt, bool, 显示时是否进行身份证号（证件号码）脱敏, dep:or,{$.VisitorRegisterInfoCfg.IDNumDisplay,eq,displayAndReq或者displayAndOpt}*/
    "phoneNoDisplay": "displayAndReq",
    /*opt, enum, 手机号是否显示, subType:string, [displayAndReq#显示必须（默认）,displayAndOpt#显示可选,hide#隐藏]*/
    "desensitisePhoneNo": true,
    /*opt, bool, 显示时是否进行手机号脱敏, dep:or,{$.VisitorRegisterInfoCfg.phoneNoDisplay,eq,displayAndReq或者displayAndOpt}*/
    "plateNoDisplay": "displayAndReq",
    /*opt, enum, 车牌号是否显示, subType:string, [displayAndReq#显示必须,displayAndOpt#显示可选（默认）,hide#隐藏]*/
    "visitTimeDisplay": "displayAndReq",
    /*opt, enum, 来访时间是否显示, subType:string, [displayAndReq#显示必须,displayAndOpt#显示可选（默认）,hide#隐藏]*/
    "receptionistDisplay": "displayAndReq",
    /*opt, enum, 接待人员是否显示, subType:string, [displayAndReq#显示必须,displayAndOpt#显示可选（默认）,hide#隐藏]*/
    "desensitiseReceptionist": true,
    /*opt, bool, 显示时是否进行接待人员脱敏, dep:or,{$.VisitorRegisterInfoCfg.receptionistDisplay,eq,displayAndReq或者displayAndOpt}*/
    "belongingsDisplay": "displayAndReq",
    /*opt, enum, 携带物品是否显示, subType:string, [displayAndReq#显示必须,displayAndOpt#显示可选（默认）,hide#隐藏]*/
    "visitPurposesDisplay": "displayAndReq",
    /*opt, enum, 来访目的是否显示, subType:string, [displayAndReq#显示必须,displayAndOpt#显示可选（默认）,hide#隐藏]*/
    "remarkDisplay": "displayAndReq",
    /*opt, enum, 备注是否显示, subType:string, [displayAndReq#显示必须,displayAndOpt#显示可选（默认）,hide#隐藏]*/
    "visitorNameDisplay": "displayAndReq",
    /*req, enum, 访客姓名是否显示, subType:string, [displayAndReq#显示必须（默认）,displayAndOpt#显示可选,hide#隐藏]*/
    "desensitiseVisitorName": true,
    /*opt, bool, 显示时是否进行访客姓名脱敏, dep:or,{$.VisitorRegisterInfoCfg.visitorNameDisplay,eq,displayAndReq或者displayAndOpt}*/
    "visitorSexDisplay": "displayAndReq",
    /*opt, enum, 访客性别是否显示, subType:string, [displayAndReq#显示必须,displayAndOpt#显示可选（默认）,hide#隐藏]*/
    "visitorCertificateTypeDisplay": "displayAndReq",
    /*opt, enum, 访客证件类型是否显示, subType:string, [displayAndReq#显示必须,displayAndOpt#显示可选（默认）,hide#隐藏]*/
    "visitorCompanyDisplay": "displayAndReq",
    /*opt, enum, 访客单位是否显示, subType:string, [displayAndReq#显示必须,displayAndOpt#显示可选（默认）,hide#隐藏]*/
    "visitAreaDisplay": "displayAndReq",
    /*opt, enum, 来访区域是否显示, subType:string, [displayAndReq#显示必须,displayAndOpt#显示可选（默认）,hide#隐藏]*/
    "CustomList": [
    /*opt, array, 自定义项列表, subType:object, range:[1,3]*/
      {
        "id": 1,
        /*opt, int, 自定义项编号, range:[1,3]*/
        "customInfo": "displayAndReq",
        /*opt, enum, 自定义信息是否显示, subType:string, [displayAndReq#显示必须,displayAndOpt#显示可选（默认）,hide#隐藏]*/
      }
    ],
    "visitorAddressDisplay": "displayAndReq",
    /*opt, enum, 访客地址是否显示, subType:string, [displayAndReq#显示必须（默认）,displayAndOpt#显示可选,hide#隐藏]*/
    "desensitiseVisitorAddress": true,
    /*opt, bool, 显示时是否进行访客地址脱敏, dep:or,{$.VisitorRegisterInfoCfg.visitorAddressDisplay,eq,displayAndReq或者displayAndOpt}*/
    "visitorTemperatureDisplay": "displayAndReq",
    /*opt, enum, 访客体温是否显示, subType:string, [displayAndReq#显示必须,displayAndOpt#显示可选（默认）,hide#隐藏]*/
    "appointmentLeaveTimeDisplay": "displayAndReq",
    /*opt, enum, 访客预约离开时间是否显示, subType:string, [displayAndReq#显示必须,displayAndOpt#显示可选（默认）,hide#隐藏]*/
    "receptionistDepartmentDisplay": "displayAndReq",
    /*opt, enum, 接待人员部门是否显示, subType:string, [displayAndReq#显示必须,displayAndOpt#显示可选（默认）,hide#隐藏]*/
    "visitorEmailDisplay": "displayAndReq",
    /*opt, enum, 访客邮箱是否显示, subType:string, [displayAndReq#显示必须,displayAndOpt#显示可选（默认）,hide#隐藏]*/
    "informedConsentDisplay": "displayAndReq",
    /*opt, enum, 信息收集知情确认是否显示, subType:string, [displayAndReq#显示必须,displayAndOpt#显示可选（默认）,hide#隐藏]*/
    "healthCodeDisplay": "displayAndReq",
    /*opt, enum, 健康码是否显示, subType:string, [displayAndReq#显示必须,displayAndOpt#显示可选（默认）,hide#隐藏]*/
  }
}

```

响应报文

```

{
  "statusCode": 1,
  /*ro, opt, int, 状态码, desc:无法用1表示时（1表示成功且无特殊状态）必须返回*/
  "statusString": "OK",
  /*ro, opt, string, 状态描述, range:[1,64], desc:无法用OK表示时（OK表示成功且无特殊状态）必须返回*/
  "subStatusCode": "ok",
  /*ro, opt, string, 子状态码, range:[1,64], desc:无法用ok表示（ok表示成功且无特殊状态）必须返回*/
  "errorCode": 1,
  /*ro, opt, int, 错误码, desc:当statusCode不为1时，与subStatusCode对应*/
  "errorMsg": "ok",
  /*ro, opt, string, 错误信息, desc:当statusCode不为1时，必须返回,解释信息在协议约束中，允许设备在后续的版本迭代中，进行优化丰富提升（不限制死）*/
}

```

69.59.2 获取访客登记信息配置

Request URL

GET /ISAPI/AccessControl/visitorRegisterInfoCfg?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "VisitorRegisterInfoCfg": {
    /*ro, req, object, 访客登记信息配置*/
    "IDNumDisplay": "displayAndReq",
    /*ro, opt, enum, 身份证号（证件号码）显示方式, subType:string, [displayAndReq#显示必须（默认）,displayAndOpt#显示可选,hide#隐藏]*/
    "desensitiseIDNum": true,
    /*ro, opt, bool, 显示时是否进行身份证号（证件号码）脱敏, dep:or,{$.VisitorRegisterInfoCfg.IDNumDisplay,eq,displayAndReq或者displayAndOpt}*/
    "phoneNoDisplay": "displayAndReq",
    /*ro, opt, enum, 手机号是否显示, subType:string, [displayAndReq#显示必须（默认）,displayAndOpt#显示可选,hide#隐藏]*/
    "desensitisePhoneNo": true,
    /*ro, opt, bool, 显示时是否进行手机号脱敏, dep:or,{$.VisitorRegisterInfoCfg.phoneNoDisplay,eq,displayAndReq或者displayAndOpt}*/
    "plateNoDisplay": "displayAndReq",
    /*ro, opt, enum, 车牌号是否显示, subType:string, [displayAndReq#显示必须,displayAndOpt#显示可选（默认）,hide#隐藏]*/
    "visitTimeDisplay": "displayAndReq",
    /*ro, opt, enum, 来访时间是否显示, subType:string, [displayAndReq#显示必须,displayAndOpt#显示可选（默认）,hide#隐藏]*/
    "receptionistDisplay": "displayAndReq",
    /*ro, opt, enum, 接待人员是否显示, subType:string, [displayAndReq#显示必须,displayAndOpt#显示可选（默认）,hide#隐藏]*/
    "desensitiseReceptionist": true,
    /*ro, opt, bool, 显示时是否进行接待人员脱敏, dep:or,{$.VisitorRegisterInfoCfg.receptionistDisplay,eq,displayAndReq或者displayAndOpt}*/
    "belongingsDisplay": "displayAndReq",
    /*ro, opt, enum, 携带物品是否显示, subType:string, [displayAndReq#显示必须,displayAndOpt#显示可选（默认）,hide#隐藏]*/
    "visitPurposesDisplay": "displayAndReq",
    /*ro, opt, enum, 来访目的是否显示, subType:string, [displayAndReq#显示必须,displayAndOpt#显示可选（默认）,hide#隐藏]*/
    "remarkDisplay": "displayAndReq",
    /*ro, opt, enum, 备注是否显示, subType:string, [displayAndReq#显示必须,displayAndOpt#显示可选（默认）,hide#隐藏]*/
    "visitorNameDisplay": "displayAndReq",
    /*ro, req, enum, 访客姓名是否显示, subType:string, [displayAndReq#显示必须（默认）,displayAndOpt#显示可选,hide#隐藏]*/
    "desensitiseVisitorName": true,
    /*ro, opt, bool, 显示时是否进行访客姓名脱敏, dep:or,{$.VisitorRegisterInfoCfg.visitorNameDisplay,eq,displayAndReq或者displayAndOpt}*/
    "visitorSexDisplay": "displayAndReq",
    /*ro, opt, enum, 访客性别是否显示, subType:string, [displayAndReq#显示必须,displayAndOpt#显示可选（默认）,hide#隐藏]*/
    "visitorCertificateTypeDisplay": "displayAndReq",
    /*ro, opt, enum, 访客证件类型是否显示, subType:string, [displayAndReq#显示必须,displayAndOpt#显示可选（默认）,hide#隐藏]*/
    "visitorCompanyDisplay": "displayAndReq",
    /*ro, opt, enum, 访客单位是否显示, subType:string, [displayAndReq#显示必须,displayAndOpt#显示可选（默认）,hide#隐藏]*/
    "visitAreaDisplay": "displayAndReq",
    /*ro, opt, enum, 来访区域是否显示, subType:string, [displayAndReq#显示必须,displayAndOpt#显示可选（默认）,hide#隐藏]*/
    "CustomList": [
      /*ro, opt, array, 自定义项列表, subType:object, range:[1,3]*/
      {
        "id": 1,
        /*ro, opt, int, 自定义项编号, range:[1,3]*/
        "customInfo": "displayAndReq",
        /*ro, opt, enum, 自定义信息是否显示, subType:string, [displayAndReq#显示必须,displayAndOpt#显示可选（默认）,hide#隐藏]*/
      }
    ],
    "visitorAddressDisplay": "displayAndReq",
    /*ro, opt, enum, 访客地址是否显示, subType:string, [displayAndReq#显示必须（默认）,displayAndOpt#显示可选,hide#隐藏]*/
    "desensitiseVisitorAddress": true,
    /*ro, opt, bool, 显示时是否进行访客地址脱敏, dep:or,{$.VisitorRegisterInfoCfg.visitorAddressDisplay,eq,displayAndReq或者displayAndOpt}*/
    "visitorTemperatureDisplay": "displayAndReq",
    /*ro, opt, enum, 访客体温是否显示, subType:string, [displayAndReq#显示必须,displayAndOpt#显示可选（默认）,hide#隐藏]*/
    "appointmentLeaveTimeDisplay": "displayAndReq",
    /*ro, opt, enum, 访客预约离开时间是否显示, subType:string, [displayAndReq#显示必须,displayAndOpt#显示可选（默认）,hide#隐藏]*/
    "receptionistDepartmentDisplay": "displayAndReq",
    /*ro, opt, enum, 接待人员部门是否显示, subType:string, [displayAndReq#显示必须,displayAndOpt#显示可选（默认）,hide#隐藏]*/
    "visitorEmailDisplay": "displayAndReq",
    /*ro, opt, enum, 访客邮箱是否显示, subType:string, [displayAndReq#显示必须,displayAndOpt#显示可选（默认）,hide#隐藏]*/
    "informedConsentDisplay": "displayAndReq",
    /*ro, opt, enum, 信息收集知情确认是否显示, subType:string, [displayAndReq#显示必须,displayAndOpt#显示可选（默认）,hide#隐藏]*/
    "healthCodeDisplay": "displayAndReq",
    /*ro, opt, enum, 健康码是否显示, subType:string, [displayAndReq#显示必须,displayAndOpt#显示可选（默认）,hide#隐藏]*/
  }
}
```

69.59.3 获取访客登记信息配置能力

Request URL

GET /ISAPI/AccessControl/visitorRegisterInfoCfg/capabilities?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "VisitorRegisterInfoCfgCap": {
    /*ro, req, object, 访客登记信息配置能力*/
    "IDNumDisplay": {
      /*ro, opt, object, 身份证号显示方式, desc:displayAndReq-显示必须（默认）， displayAndOpt-显示可选, hide-隐藏*/
      "@opt": ["displayAndReq", "displayAndOpt", "hide"]
      /*ro, req, array, 可选项, subType:string*/
    },
    "desensitiseIDNum": {
      /*ro, opt, object, 显示时是否进行身份证号（证件号码）脱敏*/
      "@opt": [true, false]
      /*ro, req, array, 可选项, subType:bool*/
    },
    "phoneNoDisplay": {
      /*ro, opt, object, 手机号是否显示, desc:displayAndReq-显示必须（默认）， displayAndOpt-显示可选, hide-隐藏*/
      "@opt": ["displayAndReq", "displayAndOpt", "hide"]
      /*ro, req, array, 可选项, subType:string*/
    },
    "desensitisePhoneNo": {
      /*ro, opt, object, 显示时是否进行手机号脱敏*/
      "@opt": [true, false]
      /*ro, req, array, 可选项, subType:bool*/
    },
    "plateNoDisplay": {
      /*ro, opt, object, 车牌号是否显示, desc:displayAndReq-显示必须, displayAndOpt-显示可选（默认）， hide-隐藏*/
      "@opt": ["displayAndReq", "displayAndOpt", "hide"]
      /*ro, req, array, 可选项, subType:string*/
    },
    "visitTimeDisplay": {
      /*ro, opt, object, 来访时间是否显示, desc:displayAndReq-显示必须（默认）， displayAndOpt-显示可选, hide-隐藏*/
      "@opt": ["displayAndReq", "displayAndOpt", "hide"]
      /*ro, req, array, 可选项, subType:string*/
    },
    "receptionistDisplay": {
      /*ro, opt, object, 接待人员是否显示,
      desc:displayAndReq-显示必须, displayAndOpt-显示可选（默认）， hide-隐藏
      注意：5032系列安卓访客机V1.1版本升级项目中，该字段默认为显示必须。*/
      "@opt": ["displayAndReq", "displayAndOpt", "hide"]
      /*ro, req, array, 可选项, subType:string*/
    },
    "desensitiseReceptionist": {
      /*ro, opt, object, 显示时是否进行接待人员脱敏*/
      "@opt": [true, false]
      /*ro, req, array, 可选项, subType:bool*/
    },
    "belongingsDisplay": {
      /*ro, opt, object, 携带物品是否显示, desc:displayAndReq-显示必须, displayAndOpt-显示可选（默认）， hide-隐藏*/
      "@opt": ["displayAndReq", "displayAndOpt", "hide"]
      /*ro, req, array, 可选项, subType:string*/
    },
    "visitPurposesDisplay": {
      /*ro, opt, object, 来访目的是否显示, desc:displayAndReq-显示必须, displayAndOpt-显示可选（默认）， hide-隐藏*/
      "@opt": ["displayAndReq", "displayAndOpt", "hide"]
      /*ro, req, array, 可选项, subType:string*/
    },
    "remarkDisplay": {
      /*ro, opt, object, 备注是否显示, desc:displayAndReq-显示必须, displayAndOpt-显示可选（默认）， hide-隐藏*/
      "@opt": ["displayAndReq", "displayAndOpt", "hide"]
      /*ro, req, array, 可选项, subType:string*/
    },
    "visitorNameDisplay": {
      /*ro, req, object, 访客姓名是否显示, desc:displayAndReq-显示必须（默认）， displayAndOpt-显示可选, hide-隐藏*/
      "@opt": ["displayAndReq", "displayAndOpt", "hide"]
      /*ro, req, array, 可选项, subType:string*/
    },
    "desensitiseVisitorName": {
      /*ro, opt, object, 显示时是否进行访客姓名脱敏*/
      "@opt": [true, false]
      /*ro, req, array, 可选项, subType:bool*/
    },
    "visitorSexDisplay": {
      /*ro, opt, object, 访客性别是否显示, desc:displayAndReq-显示必须, displayAndOpt-显示可选（默认）， hide-隐藏*/
      "@opt": ["displayAndReq", "displayAndOpt", "hide"]
      /*ro, req, array, 可选项, subType:string*/
    },
    "visitorCertificateTypeDisplay": {
      /*ro, opt, object, 访客证件类型是否显示, desc:displayAndReq-显示必须, displayAndOpt-显示可选（默认）， hide-隐藏*/
      "@opt": ["displayAndReq", "displayAndOpt", "hide"]
      /*ro, req, array, 可选项, subType:string*/
    },
    "visitorCompanyDisplay": {
      /*ro, opt, object, 访客单位是否显示, desc:displayAndReq-显示必须, displayAndOpt-显示可选（默认）， hide-隐藏*/
      "@opt": ["displayAndReq", "displayAndOpt", "hide"]
      /*ro, req, array, 可选项, subType:string*/
    },
    "visitAreaDisplay": {
      /*ro, opt, object, 来访区域是否显示, desc:displayAndReq-显示必须, displayAndOpt-显示可选（默认）， hide-隐藏*/
      "@opt": ["displayAndReq", "displayAndOpt", "hide"]
    }
  }
}
```

```

    /*ro, req, array, 可选项, subType:string*/
  },
  "CustomList": {
    /*ro, opt, object, 自定义项列表*/
    "@size": 3,
    /*ro, req, int, 最大个数, range:[1,3], step:1*/
    "id": {
      /*ro, req, object, 自定义项编号*/
      "@min": 1,
      /*ro, req, int, 最小值, range:[1,3], step:1*/
      "@max": 3
      /*ro, req, int, 最大值, range:[1,3], step:1*/
    },
    "customInfo": {
      /*ro, opt, object, 自定义信息是否显示, desc:displayAndReq-显示必须, displayAndOpt-显示可选（默认）, hide-隐藏*/
      "@opt": ["displayAndReq", "displayAndOpt", "hide"]
      /*ro, req, array, 可选项, subType:string*/
    }
  },
  "visitorAddressDisplay": {
    /*ro, opt, object, 访客地址是否显示, desc:displayAndReq-显示必须, displayAndOpt-显示可选（默认）, hide-隐藏*/
    "@opt": ["displayAndReq", "displayAndOpt", "hide"]
    /*ro, req, array, 可选项, subType:string*/
  },
  "desensitiseVisitorAddress": {
    /*ro, opt, object, 显示时是否进行访客地址脱敏*/
    "@opt": [true, false]
    /*ro, req, array, 可选项, subType:bool*/
  },
  "visitorTemperatureDisplay": {
    /*ro, opt, object, 访客体温是否显示, desc:displayAndReq-显示必须（默认）, displayAndOpt-显示可选, hide-隐藏*/
    "@opt": ["displayAndReq", "displayAndOpt", "hide"]
    /*ro, req, array, 可选项, subType:string*/
  },
  "appointmentLeaveTimeDisplay": {
    /*ro, opt, object, 访客预约离开时间是否显示, desc:displayAndReq-显示必须, displayAndOpt-显示可选（默认）, hide-隐藏*/
    "@opt": ["displayAndReq", "displayAndOpt", "hide"]
    /*ro, req, array, 可选项, subType:string*/
  },
  "receptionistDepartmentDisplay": {
    /*ro, opt, object, 接待人员部门是否显示, desc:displayAndReq-显示必须, displayAndOpt-显示可选（默认）, hide-隐藏*/
    "@opt": ["displayAndReq", "displayAndOpt", "hide"]
    /*ro, req, array, 可选项, subType:string*/
  },
  "visitorEmailDisplay": {
    /*ro, opt, object, 访客邮箱是否显示, desc:displayAndReq-显示必须, displayAndOpt-显示可选（默认）, hide-隐藏*/
    "@opt": ["displayAndReq", "displayAndOpt", "hide"]
    /*ro, req, array, 可选项, subType:string*/
  },
  "informedConsentDisplay": {
    /*ro, opt, object, 信息收集知情确认是否显示, desc:displayAndReq-显示必须, displayAndOpt-显示可选（默认）, hide-隐藏*/
    "@opt": ["displayAndReq", "displayAndOpt", "hide"]
    /*ro, req, array, 可选项, subType:string*/
  },
  "healthCodeDisplay": {
    /*ro, opt, object, 健康码是否显示, desc:displayAndReq-显示必须, displayAndOpt-显示可选（默认）, hide-隐藏*/
    "@opt": ["displayAndReq", "displayAndOpt", "hide"]
    /*ro, req, array, 可选项, subType:string*/
  }
}
}
}

```

69.59.4 配置自定义信息

Request URL

PUT /ISAPI/AccessControl/customInfoCfg?format=json

查询参数

无

请求报文

```

{
  "CustomInfoCfgList": [
    /*req, array, 自定义信息配置列表, subType:object, range:[1,3]*/
    {
      "id": 1,
      /*req, int, 自定义项编号, range:[1,3]*/
      "customName": "test"
      /*opt, string, 自定义项名称, range:[1,64]*/
    }
  ]
}

```

响应报文

```
{
  "statusCode": 1,
  /*ro, opt, int, 状态码, desc:无法用1表示时（1表示成功且无特殊状态）必须返回*/
  "statusString": "OK",
  /*ro, opt, string, 状态描述, range:[1,64], desc:无法用OK表示时（OK表示成功且无特殊状态）必须返回*/
  "subStatusCode": "ok",
  /*ro, opt, string, 子状态码, range:[1,64], desc:无法用ok表示（ok表示成功且无特殊状态）必须返回*/
  "errorCode": 1,
  /*ro, opt, int, 错误码, desc:当statusCode不为1时，与subStatusCode对应*/
  "errorMsg": "ok"
  /*ro, opt, string, 错误信息, desc:当statusCode不为1时，必须返回,解释信息在协议约束中，允许设备在后续的版本迭代中，进行优化丰富提升（不限制死）*/
}
```

69.59.5 获取自定义信息

Request URL

GET /ISAPI/AccessControl/customInfoCfg?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "CustomInfoCfgList": [
    /*ro, req, array, 自定义信息配置列表, subType:object, range:[1,3]*/
    {
      "id": 1,
      /*ro, req, int, 自定义项编号, range:[1,3]*/
      "customName": "test"
      /*ro, opt, string, 自定义项名称, range:[1,64]*/
    }
  ]
}
```

69.59.6 获取自定义信息参数配置能力

Request URL

GET /ISAPI/AccessControl/customInfoCfg/capabilities?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "CustomInfoCfgList": {
    /*ro, req, object, 自定义信息列表*/
    "@size": 3,
    /*ro, opt, int, 最大个数*/
    "id": {
      /*ro, req, object, 自定义项编号*/
      "@min": 1,
      /*ro, opt, int, 最小值*/
      "@max": 3
      /*ro, opt, int, 最大值*/
    },
    "customName": {
      /*ro, opt, object, 自定义项名称*/
      "@min": 1,
      /*ro, opt, int, 最小值*/
      "@max": 64
      /*ro, opt, int, 最大值*/
    }
  }
}
```

69.60 梯控参数管理

69.60.1 获取梯控参数能力

Request URL

GET /ISAPI/VideoIntercom/Elevators/<elevatorID>/ControlCfg/capabilities?format=json

查询参数

参数名称	参数类型	描述
elevatorID	string	

请求报文

无

响应报文

```
{
  "ElevatorControlCfg": {
    /*ro, opt, object, 梯控参数能力*/
    "elevatorNo": {
      /*ro, opt, object, 梯控编号*/
      "@min": 1,
      /*ro, opt, int, 最小值*/
      "@max": 1
      /*ro, opt, int, 最大值*/
    },
    "enable": {
      /*ro, opt, object, 是否启用梯控*/
      "@opt": ["true", "false"]
      /*ro, opt, string, 枚举*/
    },
    "numOfNegFloors": {
      /*ro, opt, object, 负楼层数量*/
      "@min": 0,
      /*ro, opt, int, 最小值*/
      "@max": 10
      /*ro, opt, int, 最大值*/
    },
    "interfaceType": {
      /*ro, opt, object, 接口类型*/
      "@opt": [1, 2]
      /*ro, opt, array, 枚举, subType:int*/
    },
    "deviceType": {
      /*ro, opt, object, 梯控器类型*/
      "@opt": ["DS-K2201", "DS-K2210", "custom"]
      /*ro, opt, array, 枚举, subType:string*/
    },
    "ServerAddress": {
      /*ro, opt, object, 梯控服务器地址*/
      "addressingFormatType": {
        /*ro, opt, object, 地址类型*/
        "@opt": ["ipaddress", "hostname"]
        /*ro, opt, array, 枚举, subType:string*/
      },
      "hostName": {
        /*ro, opt, object, 域名*/
        "@min": 1,
        /*ro, opt, int, 最小长度*/
        "@max": 1
        /*ro, opt, int, 最大长度*/
      },
      "ipAddress": {
        /*ro, opt, object, ipv4地址*/
        "@min": 1,
        /*ro, opt, int, 最小长度*/
        "@max": 1
        /*ro, opt, int, 最大长度*/
      },
      "ipv6Address": {
        /*ro, opt, object, ipv6地址*/
        "@min": 1,
        /*ro, opt, int, 最小长度*/
        "@max": 1
        /*ro, opt, int, 最大长度*/
      }
    },
    "serverPort": {
      /*ro, opt, object, 服务器端口*/
      "@min": 1,
      /*ro, opt, int, 最小值*/
      "@max": 1
      /*ro, opt, int, 最大值*/
    }
  }
}
```

```
    },
    "userName": {
      /*ro, opt, object, 用户名*/
      "@min": 0,
      /*ro, opt, int, 最小长度*/
      "@max": 128
      /*ro, opt, int, 最大长度*/
    },
    "password": {
      /*ro, opt, object, 密码*/
      "@min": 1,
      /*ro, opt, int, 最小长度*/
      "@max": 1
      /*ro, opt, int, 最大长度*/
    },
    "isSupportRS485": {
      /*ro, opt, object, 是否支持485接入*/
      "@opt": ["DS-K2201", "DS-K2210"]
      /*ro, opt, array, 枚举, subType:string*/
    },
    "isSupportNetwork": {
      /*ro, opt, object, 是否支持网络接入*/
      "@opt": ["DS-K2201", "DS-K2210"]
      /*ro, opt, array, 枚举, subType:string*/
    },
    "isSupportNegativeFloor": {
      /*ro, opt, object, 是否支持负号楼层层数配置*/
      "@opt": ["DS-K2201", "DS-K2210"]
      /*ro, opt, array, 枚举, subType:string*/
    }
  }
}
```

69.60.2 获取梯控参数

Request URL

GET /ISAPI/VideoIntercom/Elevators/<elevatorID>/ControlCfg?format=json&security=<security>&iv=<iv>

查询参数

参数名称	参数类型	描述
elevatorID	string	从1开始
security	string	没有security表示数据不加密 security=1,表示报文中的userName, password字段AES128加密 security=2,表示报文中的userName, password字段AES256加密
iv	string	iv表示初始化向量，在security=1或者2时为必须参数，长度限制为32

请求报文

无

响应报文

```
{
  "ElevatorControlCfg": {
    /*ro, opt, object, 梯控参数*/
    "enable": "true",
    /*ro, opt, string, 是否启用梯控*/
    "numOfNegFloors": 1,
    /*ro, opt, int, 负楼层数量*/
    "interfaceType": 1,
    /*ro, opt, enum, 接口类型, subType:int, [1#RS485,2#网口]*/
    "deviceType": "DS-K2201",
    /*ro, opt, enum, 梯控器类型, subType:string, [DS-K2201#DS-K2201,DS-K2210#DS-K2210,custom#custom]*/
    "ServerAddress": {
      /*ro, opt, object, IP地址, desc:当interfaceType为2时有效*/
      "addressingFormatType": "ipaddress",
      /*ro, req, enum, 地址格式类型, subType:string, [ipaddress#ip地址,hostname#域名]*/
      "hostName": "test",
      /*ro, opt, string, 域名*/
      "ipAddress": "test",
      /*ro, opt, string, ipv4地址*/
      "ipv6Address": "test"
      /*ro, opt, string, ipv6地址*/
    },
    "serverPort": 0,
    /*ro, opt, int, 梯控服务器端口号, desc:当interfaceType为2时有效*/
    "userName": "test",
    /*ro, opt, string, 用户名, desc:当interfaceType为2时有效*/
    "password": "test"
    /*ro, opt, string, 密码, desc:当interfaceType为2时有效*/
  }
}
```

69.60.3 配置梯控参数

Request URL

PUT /ISAPI/VideoIntercom/Elevators/<elevatorID>/ControlCfg?format=json&security=<security>&iv=<iv>

查询参数

参数名称	参数类型	描述
elevatorID	string	--
security	string	--
iv	string	--

请求报文

```
{
  "ElevatorControlCfg": {
    /*opt, object, 梯控参数*/
    "enable": "true",
    /*opt, string, 是否启用梯控*/
    "numOfNegFloors": 1,
    /*opt, int, 负楼层数量*/
    "interfaceType": 1,
    /*opt, enum, 接口类型, subType:int, [1#RS485,2#网口]*/
    "deviceType": "DS-K2201",
    /*opt, enum, 梯控器类型, subType:string, [DS-K2201#DS-K2201,DS-K2210#DS-K2210,custom#custom]*/
    "ServerAddress": {
      /*opt, object, IP地址, desc:当interfaceType为2时有效*/
      "addressingFormatType": "ipaddress",
      /*req, enum, 地址格式类型, subType:string, [ipaddress#ip地址,hostname#域名]*/
      "hostName": "test",
      /*opt, string, 域名*/
      "ipAddress": "test",
      /*opt, string, ipv4地址*/
      "ipv6Address": "test"
      /*opt, string, ipv6地址*/
    },
    "serverPort": 0,
    /*opt, int, 梯控服务器端口号, desc:当interfaceType为2时有效*/
    "userName": "test",
    /*opt, string, 用户名, desc:当interfaceType为2时有效*/
    "password": "test"
    /*opt, string, 密码, desc:当interfaceType为2时有效*/
  }
}
```

响应报文

```
{
  "statusCode": 1,
  /*ro, opt, int, 状态码, desc:无法用1表示时（1表示成功且无特殊状态）必须返回*/
  "statusString": "OK",
  /*ro, opt, string, 状态描述, range:[1,64], desc:无法用OK表示时（OK表示成功且无特殊状态）必须返回*/
  "subStatusCode": "ok",
  /*ro, opt, string, 子状态码, range:[1,64], desc:无法用ok表示（ok表示成功且无特殊状态）必须返回*/
  "errorCode": 1,
  /*ro, opt, int, 错误码, desc:当statusCode不为1时，与subStatusCode对应*/
  "errorMsg": "ok"
  /*ro, opt, string, 错误信息, desc:当statusCode不为1时，必须返回,解释信息在协议约束中，允许设备在后续的版本迭代中，进行优化丰富提升（不限制死）*/
}
```

69.61 通话交互管理

69.61.1 获取通话信令交互能力

Request URL

GET /ISAPI/VideoIntercom/callSignal/capabilities?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "CallSignal": {
    /*ro, req, object, 通话信令交互参数*/
    "cmdType": {
      /*ro, req, object, 信令类型*/
      "@opt": ["request", "cancel", "answer", "reject", "bellTimeout", "hangUp", "deviceOnCall", "callFailed", "clientOnCall"]
      /*ro, req, array, 取值范围, subType:string, range:[0,7], desc:request -请求呼叫, cancel -取消本次呼叫, answer -接听本次呼叫 reject -拒绝来电呼叫
      bellTimeout -被叫响铃超时 hangUp -结束本次通话, deviceOnCall -设备正在通话中, callFailed-呼叫失败, clientOnCall-客户端处于通话中*/
    },
    "sessionId": {
      /*ro, opt, object, 操作标识, desc:用于标记操作者、会话标识*/
      "@min": 0,
      /*ro, req, int, 最小值, range:[0,64]*/
      "@max": 64
      /*ro, req, int, 最大值, range:[0,64]*/
    },
    "periodNumber": {
      /*ro, opt, object, 期号/关押室/院区*/
      "@min": 0,
      /*ro, req, int, 最小值, range:[0,65535]*/
      "@max": 65535
      /*ro, req, int, 最大值, range:[0,65535]*/
    },
    "buildingNumber": {
      /*ro, opt, object, 楼号/幢号*/
      "@min": 0,
      /*ro, req, int, 最小值, range:[0,65535]*/
      "@max": 65535
      /*ro, req, int, 最大值, range:[0,65535]*/
    },
    "unitNumber": {
      /*ro, opt, object, 单元号/监区/科室*/
      "@min": 0,
      /*ro, req, int, 最小值, range:[0,65535]*/
      "@max": 65535
      /*ro, req, int, 最大值, range:[0,65535]*/
    },
    "floorNumber": {
      /*ro, opt, object, 层号*/
      "@min": 0,
      /*ro, req, int, 最小值, range:[0,65535]*/
      "@max": 65535
      /*ro, req, int, 最大值, range:[0,65535]*/
    },
    "roomNumber": {
      /*ro, opt, object, 房间号/监室/病房*/
      "@min": 0,
      /*ro, req, int, 最小值, range:[0,65535]*/
      "@max": 65535
      /*ro, req, int, 最大值, range:[0,65535]*/
    },
    "devIndex": {
      /*ro, opt, object, 设备编号/床号*/
      "@min": 0,
      /*ro, req, int, 最小值, range:[0,65535]*/
      "@max": 65535
      /*ro, req, int, 最大值, range:[0,65535]*/
    }
  }
}
```

```

/*ro, req, int, 取小值, range:[0,65535]*/
"@max": 65535
/*ro, req, int, 最大值, range:[0,65535]*/
},
"communityNumber": {
/*ro, opt, object, 小区编号*/
"@min": 0,
/*ro, req, int, 最小值, range:[0,65535]*/
"@max": 65535
/*ro, req, int, 最大值, range:[0,65535]*/
},
"callNumber": {
/*ro, opt, object, 呼叫号码*/
"@min": 1,
/*ro, req, int, 最小值, range:[1,64]*/
"@max": 64
/*ro, req, int, 最大值, range:[1,64]*/
},
"unitType": {
/*ro, opt, object, 设备类型*/
"@opt": ["indoor", "villa", "confirm", "outdoor", "fence", "doorbell", "manage", "acs", "interactive", "center", "ezviz",
"windowCounterIntercom"]
/*ro, req, array, 取值范围, subType:string, range:[0,11], desc:indoor-室内机 villa-别墅门口机 confirm-二次确认机 outdoor-门口机 fence-围墙机
doorbell-门铃机 manage-管理机 acs-门禁设备 interactive-终端 center-中心 ezviz-萤石云 windowCounterIntercom#柜台窗口IP对讲设备*/
},
"personUUID": {
/*ro, opt, object, 人员UUID*/
"@min": 1,
/*ro, req, int, 最小值, range:[1,64]*/
"@max": 64
/*ro, req, int, 最大值, range:[1,64]*/
},
"personType": {
/*ro, opt, object, 人员类型*/
"@opt": ["student", "parent"]
/*ro, req, array, 取值范围, subType:string, range:[0,2], desc:student-学生,parent-家长,默认家长*/
},
"industryType": {
/*ro, opt, object, 行业(场景)类型, desc:buildings#楼宇,prison#监所,medicalTreatment#医疗,broadcasting#广播*/
"@opt": ["buildings", "prison", "medicalTreatment", "broadcasting"]
/*ro, req, array, 取值范围, subType:string, range:[0,4]*/
},
"callType": {
/*ro, opt, object, 通话类型*/
"@opt": ["voice", "video"]
/*ro, req, array, 取值范围, subType:string, range:[0,2], desc:voice-语音,video-视频,默认视频*/
},
"model": {
/*ro, opt, object, 设备型号字符串长度取值范围, desc:range:[1,64]*/
"@min": 1,
/*ro, req, int, 最小值, range:[1,64]*/
"@max": 64
/*ro, req, int, 最大值, range:[1,64]*/
},
"callWaitingDelayTime": {
/*ro, opt, object, 呼叫等待延时时间*/
"@min": 10,
/*ro, req, int, 最小值, range:[10,60], unit:s, unitType:时间*/
"@max": 60
/*ro, req, int, 最大值, range:[10,60], unit:s, unitType:时间*/
},
"CallInfo": {
/*ro, opt, object, 通话信息*/
"STSPort": {
/*ro, opt, object, 转发服务端口*/
"@min": 0,
/*ro, opt, int, 最小值*/
"@max": 0
/*ro, opt, int, 最大值*/
},
"STSIpV4Address": {
/*ro, opt, object, 转发服务地址*/
"@min": 0,
/*ro, opt, int, 最小值*/
"@max": 0
/*ro, opt, int, 最大值*/
},
"clientID": {
/*ro, opt, object, 客户端ID*/
"@min": 0,
/*ro, opt, int, 最小值*/
"@max": 0
/*ro, opt, int, 最大值*/
},
"VCPort": {
/*ro, opt, object, 会控服务端口*/
"@min": 0,
/*ro, opt, int, 最小值*/
"@max": 0
/*ro, opt, int, 最大值*/
},
"VCIPV4Address": {
/*ro, opt, object, 会控服务地址*/
"@min": 0,

```

```

        /*ro, opt, int, 最小值*/
        "@max": 0
        /*ro, opt, int, 最大值*/
    },
    "roomID": {
        /*ro, opt, object, 房间号*/
        "@min": 0,
        /*ro, opt, int, 最小值*/
        "@max": 0
        /*ro, opt, int, 最大值*/
    },
    "password": {
        /*ro, opt, object, 密码*/
        "@min": 0,
        /*ro, opt, int, 最小值*/
        "@max": 0
        /*ro, opt, int, 最大值*/
    },
    "conferenceMaxCount": {
        /*ro, opt, object, 最大会议方数*/
        "@min": 0,
        /*ro, opt, int, 最小值*/
        "@max": 0
        /*ro, opt, int, 最大值*/
    }
},
"messageID": {
    /*ro, opt, object, 消息id*/
    "@min": 0,
    /*ro, opt, int, 最小值*/
    "@max": 0
    /*ro, opt, int, 最大值, desc:用于标识和对讲交互事件voiceTalkEvent的对应关系*/
},
"exceptionInfo": {
    /*ro, opt, object, 异常信息*/
    "@min": 0,
    /*ro, opt, int, 最小值*/
    "@max": 0
    /*ro, opt, int, 最大值*/
},
"coderType": {
    /*ro, opt, object, 编解码类型, desc:本节点当前仅HEOP协议中会使用。APK根据编解码类型决定语音数据使用萤石SDK编解码还是使用DSP进行编解码。萤石SDK编解码的情况来自交互方为公网上的云眸平台。DSP编解码的情况来自传统的楼宇对讲*/
    "@opt": ["ezviz", "dsp"]
    /*ro, req, array, 取值范围, subType:string, range:[0,2], desc:[ezviz#萤石SDK编解码,dsp#DSP编解码]*/
},
"assemblyAgreementNum": {
    /*ro, opt, object, 号码组装协议取值范围, desc:对讲设备呼叫时, 不同协议类型接入的被叫设备号码组装规则不同。无此节点时默认SIP接入*/
    "@opt": ["sip", "ezviz", "ISUP"]
    /*ro, opt, array, 号码组装协议取值枚举, subType:string*/
}
}
}
}

```

69.61.2 获取通话状态能力

Request URL

GET /ISAPI/VideoIntercom/callStatus/capabilities?format=json

查询参数

无

请求报文

无

响应报文

```

{
  "CallStatus": {
    /*ro, opt, object, 通话状态*/
    "status": {
        /*ro, req, object, 状态*/
        "@opt": ["idle", "ring", "onCall"]
        /*ro, req, array, 取值范围, subType:string*/
    }
  }
}

```

69.61.3 获取通话状态

Request URL

GET /ISAPI/VideoIntercom/callStatus?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "CallStatus": {
    /*ro, opt, object, 通话状态*/
    "status": "idle"
    /*ro, req, enum, 状态, subType:string, [idle#空闲,ring#响铃,onCall#通话中]*/
  }
}
```

69.61.4 获取主叫信息

Request URL

GET /ISAPI/VideoIntercom/callerInfo?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "CallerInfo": {
    /*ro, req, object, 主叫信息*/
    "buildingNo": 1,
    /*ro, opt, int, 楼号, range:[0,16]*/
    "floorNo": 1,
    /*ro, opt, int, 层号, range:[0,16]*/
    "zoneNo": 1,
    /*ro, opt, int, 期号, range:[0,16]*/
    "unitNo": 1,
    /*ro, opt, int, 单元号, range:[0,16]*/
    "devNo": 1,
    /*ro, opt, int, 设备序号, range:[0,16]*/
    "devType": 1,
    /*ro, req, int, 设备类型, desc:1-门口机, 2-管理机, 3-室内机, 4-围墙机, 5-别墅门口机, 6-二次确认机, 7-8700客户端, 8-4200客户端, 9-APP, 10- 门铃
    机, 11-VOIP客户端, 12-监控点IPC设备, 13-门禁设备, 14-报警设备*/
    "lockNum": 1,
    /*ro, opt, int, 锁数量, range:[0,16]*/
    "voipNo": "test",
    /*ro, opt, string, voip客户端号码, desc:devType为11时有效*/
    "status": "idle"
    /*ro, req, string, 状态, desc:idle-空闲, ring-响铃, onCall-通话中*/
  }
}
```

69.61.5 获取主叫信息能力

Request URL

GET /ISAPI/VideoIntercom/callerInfo/capabilities?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "CallerInfo": {
    /*ro, req, object, 主叫信息*/
    "buildingNo": {
      /*ro, opt, object, 楼号*/
      "@min": 0,
      /*ro, opt, int, 最小值*/
      "@max": 16
      /*ro, opt, int, 最大值*/
    },
    "floorNo": {
      /*ro, opt, object, 层号*/
      "@min": 0,
      /*ro, opt, int, 最小值*/
      "@max": 16
      /*ro, opt, int, 最大值*/
    },
    "zoneNo": {
      /*ro, opt, object, 期号*/
      "@min": 0,
      /*ro, opt, int, 最小值*/
      "@max": 16
      /*ro, opt, int, 最大值*/
    },
    "unitNo": {
      /*ro, opt, object, 单元号*/
      "@min": 0,
      /*ro, opt, int, 最小值*/
      "@max": 16
      /*ro, opt, int, 最大值*/
    },
    "devNo": {
      /*ro, opt, object, 设备编号*/
      "@min": 0,
      /*ro, opt, int, 最小值*/
      "@max": 16
      /*ro, opt, int, 最大值*/
    },
    "devType": {
      /*ro, req, object, 设备类型, desc:1-门口机, 2-管理机, 3-室内机, 4-围墙机, 5-别墅门口机, 6-二次确认机, 7-8700客户端, 8-4200客户端, 9-APP, 10- 门铃
      机, 11-VOIP客户端, 12-监控点IPC设备, 13-门禁设备, 14-报警设备*/
      "@opt": "1,2,3,4,5,6,7,8,9,10,11,12,13,14"
      /*ro, opt, string, 可选值*/
    },
    "lockNum": {
      /*ro, opt, object, 锁数量*/
      "@min": 0,
      /*ro, opt, int, 最小值*/
      "@max": 16
      /*ro, opt, int, 最大值*/
    },
    "voipNo": {
      /*ro, opt, object, voip客户端号码, desc:devType为11时有效*/
      "@min": 0,
      /*ro, opt, int, 最小值*/
      "@max": 16
      /*ro, opt, int, 最大值*/
    },
    "status": {
      /*ro, req, object, 状态, desc:idle-空闲, ring-响铃, onCall-通话中*/
      "@opt": "idle,ring,onCall"
      /*ro, opt, string, 可选值*/
    }
  }
}
```

69.62 对讲基础信息管理

69.62.1 配置设备编号参数

Request URL

PUT /ISAPI/VideoIntercom/deviceId?readerID=<readerID>

查询参数

参数名称	参数类型	描述
readerID	string	智能主机存在多路读卡器，需要支持按ID配置，需要注意readerID不存在时，表示对主机的配置。主机支持SIP服务和SIP客户端，SIP客户端用于注册到管理机

请求报文

```
<?xml version="1.0" encoding="UTF-8"?>

<DeviceId xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, 设备编号, attr:version{req, string, 协议版本}-->
  <unitType>
    <!--req, enum, 设备类型, subType:string, [villa#villa,confirm#confirm,outdoor#门口机,fence#围墙机,doorbell#doorbell,manage#管理机,indoor#室内机,analogIndoor#模拟室内机,decoder#解码器,netAudio#网络音箱,amplifier#功放设备,pagingMicrophone#寻呼话筒,interactive#终端设备]-->villa
  </unitType>
  <periodNumber>
    <!--opt, int, 期号-->0
  </periodNumber>
  <buildingNumber>
    <!--opt, int, 楼号-->0
  </buildingNumber>
  <unitNumber>
    <!--opt, int, 单元号-->0
  </unitNumber>
  <floorNumber>
    <!--opt, int, 层号, desc:地下一层用-1表示, 地上一层用1表示-->0
  </floorNumber>
  <roomNumber>
    <!--opt, int, 房间号-->0
  </roomNumber>
  <deviceIndex>
    <!--opt, int, 设备序号-->0
  </deviceIndex>
  <communityNumber>
    <!--opt, string, 小区编号-->test
  </communityNumber>
  <customDeviceId>
    <!--opt, string, 自定义编号-->test
  </customDeviceId>
  <periodNumberName>
    <!--opt, string, periodNumber自定义名称, range:[0,32]-->test
  </periodNumberName>
  <buildingNumberName>
    <!--opt, string, buildingNumber自定义名称, range:[0,32]-->test
  </buildingNumberName>
  <unitNumberName>
    <!--opt, string, unitNumber自定义名称, range:[0,32]-->test
  </unitNumberName>
  <floorNumberName>
    <!--opt, string, floorNumber自定义名称, range:[0,32]-->test
  </floorNumberName>
  <roomNumberName>
    <!--opt, string, roomNumber自定义名称, range:[0,32]-->test
  </roomNumberName>
  <deviceIndexName>
    <!--opt, string, deviceIndex自定义名称, range:[0,32]-->test
  </deviceIndexName>
  <communityNumberName>
    <!--opt, string, communityNumber自定义名称, range:[0,128]-->test
  </communityNumberName>
  <enabled>
    <!--opt, bool, 设备编号(老长号)使能-->true
  </enabled>
  <industryType>
    <!--opt, enum, 行业(场景)类型, subType:string, [buildings#楼宇,prison#监所,medicalTreatment#医疗,broadcasting#广播]-->buildings
  </industryType>
</DeviceId>
```

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 响应消息, attr:version{ro, req, string, 版本}-->
  <requestURL>
    <!--ro, req, string, 请求的URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, 状态码, subType:int, [0#OK,1#OK,2#Device Busy,3#Device Error,4#Invalid Operation,5#Invalid XML Format,6#Invalid XML Content,7#Reboot Required]-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, 状态信息, subType:string, [OK#成功,Device Busy#设备忙,Device Error#设备异常,Invalid Operation#无效的操作,Invalid XML Format#无效的XML格式,Invalid XML Content#无效的XML内容,Reboot#设备重启]-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, 详细错误码英文描述, desc:详细错误码的英文描述-->OK
  </subStatusCode>
</ResponseStatus>
```

69.62.2 获取设备编号参数

Request URL

GET /ISAPI/VideoIntercom/deviceId?readerID= <readerID>

查询参数

参数名称	参数类型	描述
readerID	string	智能主机存在多路读卡器，需要支持按ID配置，需要注意readerID不存在时，表示对主机的配置。主机支持SIP服务和SIP客户端，SIP客户端用于注册到管理机

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<DeviceId xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 设备编号, attr:version{req, string, 协议版本}-->
  <unitType>
    <!--ro, req, enum, 设备类型, subType:string, [villa#villa,confirm#confirm,outdoor#门口机,fence#围墙机,doorbell#doorbell,manage#管理机,indoor#室内机,analogIndoor#模拟室内机,decoder#解码器,netAudio#网络音箱,ampLifier#功放设备,pagingMicrophone#寻呼话筒,interactive#终端设备]-->villa
  </unitType>
  <periodNumber>
    <!--ro, opt, int, 期号-->0
  </periodNumber>
  <buildingNumber>
    <!--ro, opt, int, 楼号-->0
  </buildingNumber>
  <unitNumber>
    <!--ro, opt, int, 单元号-->0
  </unitNumber>
  <floorNumber>
    <!--ro, opt, int, 层号, desc:地下一层用-1表示, 地上一层用1表示-->0
  </floorNumber>
  <roomNumber>
    <!--ro, opt, int, 房间号-->0
  </roomNumber>
  <deviceIndex>
    <!--ro, opt, int, 设备序号-->0
  </deviceIndex>
  <communityNumber>
    <!--ro, opt, string, 小区编号-->test
  </communityNumber>
  <customDeviceId>
    <!--ro, opt, string, 自定义编号-->test
  </customDeviceId>
  <periodNumberName>
    <!--ro, opt, string, periodNumber自定义名称, range:[0,32]-->test
  </periodNumberName>
  <buildingNumberName>
    <!--ro, opt, string, buildingNumber自定义名称, range:[0,32]-->test
  </buildingNumberName>
  <unitNumberName>
    <!--ro, opt, string, unitNumber自定义名称, range:[0,32]-->test
  </unitNumberName>
  <floorNumberName>
    <!--ro, opt, string, floorNumber自定义名称, range:[0,32]-->test
  </floorNumberName>
  <roomNumberName>
    <!--ro, opt, string, roomNumber自定义名称, range:[0,32]-->test
  </roomNumberName>
  <deviceIndexName>
    <!--ro, opt, string, deviceIndex自定义名称, range:[0,32]-->test
  </deviceIndexName>
  <communityNumberName>
    <!--ro, opt, string, communityNumber自定义名称, range:[0,128]-->test
  </communityNumberName>
  <enabled>
    <!--ro, opt, bool, 设备编号（老长号）使能-->true
  </enabled>
  <industryType>
    <!--ro, opt, enum, 行业(场景)类型, subType:string, [buiLidings#楼宇,prison#监所,medicalTreatment#医疗,broadcasting#广播]-->buildings
  </industryType>
</DeviceId>
```

69.62.3 获取语音参数

Request URL

GET /ISAPI/VideoIntercom/Configurations/voice?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "Voice": {
    /*ro, opt, object, 语音参数*/
    "enableSpeechRecognition": true,
    /*ro, opt, bool, 语音识别使能*/
    "mode": "wakeup",
    /*ro, opt, enum, 语音识别模式, subType:string, [wakeup#唤醒,keywords#关键词]*/
    "timeout": 1,
    /*ro, opt, int, 语音识别超时时间, range:[0,60], step:1, unit:s, unitType:时间*/
    "enableVoiceBroadcast": true,
    /*ro, opt, bool, 使能语音播报*/
    "continuousSpeechEnabled": true
    /*ro, opt, bool, 连续语音识别使能, desc:动词和名词组合成*/
  }
}
```

69.62.4 获取语音参数配置能力

Request URL

GET /ISAPI/VideoIntercom/Configurations/voice/capabilities?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "VoiceCap": {
    /*ro, opt, object, 语音参数配置能力集*/
    "enableSpeechRecognition": {
      /*ro, opt, object, 语音识别使能*/
      "@opt": [true, false]
      /*ro, req, array, 语音识别使能取值范围, subType:bool*/
    },
    "mode": {
      /*ro, opt, object, 语音识别模式取值范围*/
      "@opt": ["wakeup", "keywords"]
      /*ro, req, array, 语音识别模式枚举值, subType:string*/
    },
    "timeout": {
      /*ro, opt, object, 语音识别超时时间取值范围*/
      "@min": 0,
      /*ro, req, int, 语音识别超时时间取值最小值, unit:s, unitType:时间*/
      "@max": 60
      /*ro, req, int, 语音识别超时时间取值最大值, unit:s, unitType:时间*/
    },
    "enableVoiceBroadcast": {
      /*ro, opt, object, 使能语音播报*/
      "@opt": [true, false]
      /*ro, req, array, 语音识别使能取值范围, subType:bool*/
    },
    "continuousSpeechEnabled": {
      /*ro, opt, object, 连续语音识别使能*/
      "@opt": [true, false]
      /*ro, req, array, 可选项, subType:bool*/
    }
  }
}
```

69.62.5 配置语音参数

Request URL

PUT /ISAPI/VideoIntercom/Configurations/voice?format=json

查询参数

无

请求报文

```
{
  "Voice": {
    /*wo, opt, object, 语音参数*/
    "enableSpeechRecognition": true,
    /*opt, bool, 语音识别使能*/
    "mode": "wakeup",
    /*opt, enum, 语音识别模式, subType:string, [wakeup#唤醒,keywords#关键词]*/
    "timeout": 1,
    /*opt, int, 语音识别超时时间, range:[0,60], step:1, unit:s, unitType:时间*/
    "enableVoiceBroadcast": true,
    /*opt, bool, 使能语音播报*/
    "continuousSpeechEnabled": true
    /*opt, bool, 连续语音识别使能, desc:动词和名词组合成*/
  }
}
```

响应报文

```
{
  "statusCode": 1,
  /*ro, opt, int, 状态码, desc:无法用1表示时（1表示成功且无特殊状态）必须返回*/
  "statusString": "OK",
  /*ro, opt, string, 状态描述, range:[1,64], desc:无法用OK表示时（OK表示成功且无特殊状态）必须返回*/
  "subStatusCode": "ok",
  /*ro, opt, string, 子状态码, range:[1,64], desc:无法用ok表示（ok表示成功且无特殊状态）必须返回*/
  "errorCode": 1,
  /*ro, opt, int, 错误码, desc:当statusCode不为1时，与subStatusCode对应*/
  "errorMsg": "ok"
  /*ro, opt, string, 错误信息, desc:当statusCode不为1时，必须返回,解释信息在协议约束中，允许设备在后续的版本迭代中，进行优化丰富提升（不限制死）*/
}
```

69.62.6 获取设备编号参数能力

Request URL

GET /ISAPI/VideoIntercom/deviceId/capabilities?unitType=<unitType>

查询参数

参数名称	参数类型	描述
unitType	enum	

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<DeviceId xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, 设备编号, attr:version{req, string, 协议版本}-->
  <unitType opt="villa,confirm,outdoor,fence,doorbell,manage,indoor,analogIndoor,decoder,netAudio,amplifier,pagingMicrophone,interactive">
    <!--ro, req, string, 设备类型, attr:opt{req, string, 取值范围}-->test
  </unitType>
  <periodNumber min="0" max="10">
    <!--ro, opt, int, 期号, attr:min{req, int, 最小值},max{req, int, 最大值}-->0
  </periodNumber>
  <buildingNumber min="0" max="10">
    <!--ro, opt, int, 楼号, attr:min{req, int, 最小值},max{req, int, 最大值}-->0
  </buildingNumber>
  <unitNumber min="0" max="10">
    <!--ro, opt, int, 单元号, attr:min{req, int, 最小值},max{req, int, 最大值}-->0
  </unitNumber>
  <floorNumber min="0" max="10">
    <!--ro, opt, int, 层号, attr:min{req, int, 最小值},max{req, int, 最大值}-->0
  </floorNumber>
  <roomNumber min="0" max="10">
    <!--ro, opt, int, 房间号, attr:min{req, int, 最小值},max{req, int, 最大值}-->0
  </roomNumber>
  <deviceIndex min="0" max="10">
    <!--ro, opt, int, 设备序号, attr:min{req, int, 最小值},max{req, int, 最大值}-->0
  </deviceIndex>
  <readerDeviceIndex min="0" max="10">
    <!--ro, opt, int, 读卡器设备序号, step:1, attr:min{req, int, 最小值},max{req, int, 最大值}, desc:智能主机外接网络读卡器支持SIP呼叫, 且读卡器的设备序号与智能主机的设备序号范围不一致, 主机仅支持0, 读卡器支持0-99-->0
  </readerDeviceIndex>
  <communityNumber min="0" max="10">
    <!--ro, opt, string, 小区编号, attr:min{req, int, 最小值},max{req, int, 最大值}-->test
  </communityNumber>
  <customDeviceId min="0" max="10">
    <!--ro, opt, string, 自定义编号, attr:min{req, int, 最小值},max{req, int, 最大值}-->test
  </customDeviceId>
  <periodNumberName min="0" max="32">
    <!--ro, opt, string, periodNumber自定义名称, range:[0,32], attr:min{req, int, 最小值},max{req, int, 最大值}-->test
  </periodNumberName>
  <buildingNumberName min="0" max="32">
    <!--ro, opt, string, buildingNumber自定义名称, range:[0,32], attr:min{req, int, 最小值},max{req, int, 最大值}-->test
  </buildingNumberName>
  <unitNumberName min="0" max="32">
    <!--ro, opt, string, unitNumber自定义名称, range:[0,32], attr:min{req, int, 最小值},max{req, int, 最大值}-->test
  </unitNumberName>
  <floorNumberName min="0" max="32">
    <!--ro, opt, string, floorNumber自定义名称, range:[0,32], attr:min{req, int, 最小值},max{req, int, 最大值}-->test
  </floorNumberName>
  <roomNumberName min="0" max="32">
    <!--ro, opt, string, roomNumber自定义名称, range:[0,32], attr:min{req, int, 最小值},max{req, int, 最大值}-->test
  </roomNumberName>
  <deviceIndexName min="0" max="32">
    <!--ro, opt, string, deviceIndex自定义名称, range:[0,32], attr:min{req, int, 最小值},max{req, int, 最大值}-->test
  </deviceIndexName>
  <communityNumberName min="0" max="32">
    <!--ro, opt, string, communityNumber自定义名称, range:[0,32], attr:min{req, int, 最小值},max{req, int, 最大值}-->test
  </communityNumberName>
  <enabled opt="true,false">
    <!--ro, opt, bool, 设备编号(老长号)使能, attr:opt{req, string, 取值范围}, desc:启用后表示设备可以配置老长号,关闭后表示不可配置,默认开启-->true
  </enabled>
  <industryType opt="buildings,prison,medicalTreatment,broadcasting">
    <!--ro, opt, string, 行业(场景)类型, attr:opt{req, string, 取值范围}, desc:[buildings#楼宇,prison#监所,medicalTreatment#医疗,broadcasting#广播]-->test
  </industryType>
  <readerID min="1" max="8">
    <!--ro, opt, int, 识别单元编号, range:[1,8], attr:min{req, int, 最小值},range:[1,8],max{req, int, 最大值},range:[1,8], desc:表示获取及配置协议的URL中是否支持readerID参数-->1
  </readerID>
  <requireReaderID>
    <!--ro, opt, bool, 请求URL中是否必须包含readerID,
    desc:字段不存在默认为false.
    对于智能主机仅支持URL按readerID配置-->true
  </requireReaderID>
  <isSupportZeroFloorNumber>
    <!--ro, opt, bool, 是否支持0号楼层, desc:和floorNumber字段一起使用,desc: 默认支持-->true
  </isSupportZeroFloorNumber>
</DeviceId>
```

69.62.7 获取关联的设备地址信息(关联网路)

Request URL

GET /ISAPI/VideoIntercom/relatedDeviceAddress

查询参数

无

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<RelatedDeviceAddress xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, 关联的设备地址信息, attr:version{req, string, 协议版本}-->
  <unitType>
    <!--ro, opt, enum, 设备类型, subType:string, [outdoor#门口机,fence#室内机]-->outdoor
  </unitType>
  <MainOutdoorAddress>
    <!--ro, opt, object, 主门口机地址-->
    <addressingFormatType>
      <!--ro, req, enum, 地址类型, subType:string, [ipaddress#IP地址,hostname#主机域名]-->ipaddress
    </addressingFormatType>
    <hostName>
      <!--ro, opt, string, 主机域名, range:[0,128]-->test
    </hostName>
    <ipAddress>
      <!--ro, opt, string, IPV4地址, range:[0,128]-->255.255.255.255
    </ipAddress>
    <ipv6Address>
      <!--ro, opt, string, IPV6地址, range:[0,128]-->ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff
    </ipv6Address>
  </MainOutdoorAddress>
  <SIPServerAddress>
    <!--ro, opt, object, SIP服务器地址-->
    <addressingFormatType>
      <!--ro, req, enum, 地址类型, subType:string, [ipaddress#IP地址,hostname#主机域名]-->ipaddress
    </addressingFormatType>
    <hostName>
      <!--ro, opt, string, 主机域名, range:[0,128]-->test
    </hostName>
    <ipAddress>
      <!--ro, opt, string, IPV4地址, range:[0,128]-->255.255.255.255
    </ipAddress>
    <ipv6Address>
      <!--ro, opt, string, IPV6地址, range:[0,128]-->ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff
    </ipv6Address>
  </SIPServerAddress>
  <centerPort>
    <!--ro, opt, int, 中心平台端口号, range:[0,65535]-->1
  </centerPort>
  <CenterAddress>
    <!--ro, opt, object, 中心平台地址-->
    <addressingFormatType>
      <!--ro, req, enum, 地址类型, subType:string, [ipaddress#IP地址,hostname#主机域名]-->ipaddress
    </addressingFormatType>
    <hostName>
      <!--ro, opt, string, 主机域名, range:[0,128]-->test
    </hostName>
    <ipAddress>
      <!--ro, opt, string, IPV4地址, range:[0,128]-->255.255.255.255
    </ipAddress>
    <ipv6Address>
      <!--ro, opt, string, IPV6地址, range:[0,128]-->ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff
    </ipv6Address>
  </CenterAddress>
  <MainRoomAddress>
    <!--ro, opt, object, 主室内机地址-->
    <addressingFormatType>
      <!--ro, req, enum, 地址类型, subType:string, [ipaddress#IP地址,hostname#主机域名]-->ipaddress
    </addressingFormatType>
    <hostName>
      <!--ro, opt, string, 主机域名, range:[0,128]-->test
    </hostName>
    <ipAddress>
      <!--ro, opt, string, IPV4地址, range:[0,128]-->255.255.255.255
    </ipAddress>
    <ipv6Address>
      <!--ro, opt, string, IPV6地址, range:[0,128]-->ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff
    </ipv6Address>
  </MainRoomAddress>
  <ManageAddress>
    <!--ro, opt, object, 管理机IP-->
    <addressingFormatType>
      <!--ro, req, enum, 地址类型, subType:string, [ipaddress#IP地址,hostname#主机域名]-->ipaddress
    </addressingFormatType>
    <hostName>
      <!--ro, opt, string, 主机域名, range:[0,128]-->test
    </hostName>
    <ipAddress>
      <!--ro, opt, string, IPV4地址, range:[0,128]-->255.255.255.255
    </ipAddress>
    <ipv6Address>
      <!--ro, opt, string, IPV6地址, range:[0,128]-->ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff
    </ipv6Address>
  </ManageAddress>
</RelatedDeviceAddress>
```

69.62.8 配置关联的设备地址信息(关联网络)

Request URL

PUT /ISAPI/VideoIntercom/relatedDeviceAddress

查询参数

无

请求报文

湖南云舟电力科技有限公司
有效期：2024-08-15

```
<?xml version="1.0" encoding="UTF-8"?>

<RelatedDeviceAddress xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--opt, object, 关联的设备地址信息, attr:version{req, string, 协议版本}-->
  <unitType>
    <!--opt, enum, 设备类型, subType:string, [outdoor#门口机,fence#室内机]-->outdoor
  </unitType>
  <MainOutdoorAddress>
    <!--opt, object, 主门口机地址-->
    <addressingFormatType>
      <!--req, enum, 地址类型, subType:string, [ipaddress#IP地址,hostname#主机域名]-->ipaddress
    </addressingFormatType>
    <hostName>
      <!--opt, string, 主机域名, range:[0,128]-->test
    </hostName>
    <ipAddress>
      <!--opt, string, IPV4地址, range:[0,128]-->255.255.255.255
    </ipAddress>
    <ipv6Address>
      <!--opt, string, IPV6地址, range:[0,128]-->ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff
    </ipv6Address>
  </MainOutdoorAddress>
  <SIPServerAddress>
    <!--opt, object, SIP服务器地址-->
    <addressingFormatType>
      <!--req, enum, 地址类型, subType:string, [ipaddress#IP地址,hostname#主机域名]-->ipaddress
    </addressingFormatType>
    <hostName>
      <!--opt, string, 主机域名, range:[0,128]-->test
    </hostName>
    <ipAddress>
      <!--opt, string, IPV4地址, range:[0,128]-->255.255.255.255
    </ipAddress>
    <ipv6Address>
      <!--opt, string, IPV6地址, range:[0,128]-->ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff
    </ipv6Address>
  </SIPServerAddress>
  <centerPort>
    <!--opt, int, 中心平台端口号, range:[0,65535]-->1
  </centerPort>
  <CenterAddress>
    <!--opt, object, 中心平台地址-->
    <addressingFormatType>
      <!--req, enum, 地址类型, subType:string, [ipaddress#IP地址,hostname#主机域名]-->ipaddress
    </addressingFormatType>
    <hostName>
      <!--opt, string, 主机域名, range:[0,128]-->test
    </hostName>
    <ipAddress>
      <!--opt, string, IPV4地址, range:[0,128]-->255.255.255.255
    </ipAddress>
    <ipv6Address>
      <!--opt, string, IPV6地址, range:[0,128]-->ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff
    </ipv6Address>
  </CenterAddress>
  <MainRoomAddress>
    <!--opt, object, 主室内机地址-->
    <addressingFormatType>
      <!--req, enum, 地址类型, subType:string, [ipaddress#IP地址,hostname#主机域名]-->ipaddress
    </addressingFormatType>
    <hostName>
      <!--opt, string, 主机域名, range:[0,128]-->test
    </hostName>
    <ipAddress>
      <!--opt, string, IPV4地址, range:[0,128]-->255.255.255.255
    </ipAddress>
    <ipv6Address>
      <!--opt, string, IPV6地址, range:[0,128]-->ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff
    </ipv6Address>
  </MainRoomAddress>
  <ManageAddress>
    <!--opt, object, 管理机IP-->
    <addressingFormatType>
      <!--req, enum, 地址类型, subType:string, [ipaddress#IP地址,hostname#主机域名]-->ipaddress
    </addressingFormatType>
    <hostName>
      <!--opt, string, 主机域名, range:[0,128]-->test
    </hostName>
    <ipAddress>
      <!--opt, string, IPV4地址, range:[0,128]-->255.255.255.255
    </ipAddress>
    <ipv6Address>
      <!--opt, string, IPV6地址, range:[0,128]-->ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff
    </ipv6Address>
  </ManageAddress>
</RelatedDeviceAddress>
```

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 响应消息, attr:version{ro, req, string, 版本}-->
  <requestURL>
    <!--ro, req, string, 请求的URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, 状态码, subType:int, [0#OK,1#OK,2#Device Busy,3#Device Error,4#Invalid Operation,5#Invalid XML Format,6#Invalid XML Content,7#Reboot Required]-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, 状态信息, subType:string, [OK#成功,Device Busy#设备忙,Device Error#设备异常,Invalid Operation#无效的操作,Invalid XML Format#无效的XML格式,Invalid XML Content#无效的XML内容,Reboot#设备重启]-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, 详细错误码英文描述, desc:详细错误码的英文描述-->OK
  </subStatusCode>
</ResponseStatus>
```

69.62.9 获取关联的设备地址(关联网络)能力

Request URL

GET /ISAPI/VideoIntercom/relatedDeviceAddress/capabilities

查询参数

无

请求报文

无

响应报文

湖南云舟电力科技有限公司
有效期：2024-08-15

```

<?xml version="1.0" encoding="UTF-8"?>

<RelatedDeviceAddress xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, 关联的设备地址(关联网络)能力集, attr:version{req, string, 协议版本}-->
  <unitType opt="outdoor,fence" ">
    <!--ro, opt, enum, 设备类型, subType:string, [ipaddress#门口机,fence#室内机], attr:opt{req, string, 取值范围}-->outdoor
  </unitType>
  <MainOutdoorAddress>
    <!--ro, opt, object, 主门口机地址-->
    <addressingFormatType opt="ipaddress,hostname">
      <!--ro, req, enum, 地址类型, subType:string, [ipaddress#IP地址,hostname#主机域名], attr:opt{req, string, 取值范围}-->ipaddress
    </addressingFormatType>
    <hostName>
      <!--ro, opt, string, 主机域名, range:[0,128]-->test
    </hostName>
    <ipAddress>
      <!--ro, opt, string, IPV4地址, range:[0,128]-->255.255.255.255
    </ipAddress>
    <ipv6Address>
      <!--ro, opt, string, IPV6地址, range:[0,128]-->ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff
    </ipv6Address>
  </MainOutdoorAddress>
  <SIPServerAddress>
    <!--ro, opt, object, SIP服务器地址-->
    <addressingFormatType opt="ipaddress,hostname">
      <!--ro, req, enum, 地址类型, subType:string, [ipaddress#IP地址,hostname#主机域名], attr:opt{req, string, 取值范围}-->ipaddress
    </addressingFormatType>
    <hostName>
      <!--ro, opt, string, 主机域名, range:[0,128]-->test
    </hostName>
    <ipAddress>
      <!--ro, opt, string, IPV4地址, range:[0,128]-->255.255.255.255
    </ipAddress>
    <ipv6Address>
      <!--ro, opt, string, IPV6地址, range:[0,128]-->ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff
    </ipv6Address>
  </SIPServerAddress>
  <centerPort>
    <!--ro, opt, int, 中心平台端口号, range:[0,65535]-->0
  </centerPort>
  <CenterAddress>
    <!--ro, opt, object, 中心平台地址-->
    <addressingFormatType opt="ipaddress,hostname">
      <!--ro, req, enum, 地址类型, subType:string, [ipaddress#IP地址,hostname#主机域名], attr:opt{req, string, 取值范围}-->ipaddress
    </addressingFormatType>
    <hostName>
      <!--ro, opt, string, 主机域名, range:[0,128]-->test
    </hostName>
    <ipAddress>
      <!--ro, opt, string, IPV4地址, range:[0,128]-->255.255.255.255
    </ipAddress>
    <ipv6Address>
      <!--ro, opt, string, IPV6地址, range:[0,128]-->ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff
    </ipv6Address>
  </CenterAddress>
  <MainRoomAddress>
    <!--ro, opt, object, 主室内机地址-->
    <addressingFormatType opt="ipaddress,hostname">
      <!--ro, req, enum, 地址类型, subType:string, [ipaddress#IP地址,hostname#主机域名], attr:opt{req, string, 取值范围}-->ipaddress
    </addressingFormatType>
    <hostName>
      <!--ro, opt, string, 主机域名, range:[0,128]-->test
    </hostName>
    <ipAddress>
      <!--ro, opt, string, IPV4地址, range:[0,128]-->255.255.255.255
    </ipAddress>
    <ipv6Address>
      <!--ro, opt, string, IPV6地址, range:[0,128]-->ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff
    </ipv6Address>
  </MainRoomAddress>
  <ManageAddress>
    <!--ro, opt, object, 管理机地址-->
    <addressingFormatType opt="ipaddress,hostname">
      <!--ro, req, enum, 地址类型, subType:string, [ipaddress#IP地址,hostname#主机域名], attr:opt{req, string, 取值范围}-->ipaddress
    </addressingFormatType>
    <hostName>
      <!--ro, opt, string, 主机域名, range:[0,128]-->test
    </hostName>
    <ipAddress>
      <!--ro, opt, string, IPV4地址, range:[0,128]-->255.255.255.255
    </ipAddress>
    <ipv6Address>
      <!--ro, opt, string, IPV6地址, range:[0,128]-->ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff
    </ipv6Address>
  </ManageAddress>
</RelatedDeviceAddress>

```

69.63.1 创建单个页面参数

Request URL

POST /ISAPI/Publish/ProgramMgr/program/<programID>/page

查询参数

参数名称	参数类型	描述
programID	string	节目ID

请求报文

```
<?xml version="1.0" encoding="UTF-8"?>

<Page xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, 页面, attr:version{req, string, 协议版本}-->
  <id>
    <!--req, int, 索引-->1
  </id>
  <PageBasicInfo>
    <!--req, object, 页面基本信息-->
    <pageName>
      <!--opt, string, 页面名称-->test
    </pageName>
    <BackgroundColor>
      <!--opt, object, 背景颜色-->
      <RGB>
        <!--opt, int, RGB值, desc: 十进制数据表示三原色, 如16777215表示0xFFFFFF-->1
      </RGB>
    </BackgroundColor>
    <playDurationMode>
      <!--opt, enum, 页面播放时间模式, subType:string, [selfDefine#自定义, auto#自动, loop#循环], desc: 自定义的playDuration节点有效; 自动时根据播放内容的时间
      *playCount计算: 循环时表示一直循环-->selfDefine
    </playDurationMode>
    <playDuration>
      <!--opt, int, 页面播放时长, unit:s, unitType:时间-->1
    </playDuration>
    <playCount>
      <!--opt, int, 页面播放次数-->1
    </playCount>
    <switchDuration>
      <!--opt, int, 切换间隔, unit:s, unitType:时间-->1
    </switchDuration>
    <switchEffect>
      <!--opt, enum, 切换效果, subType:string, [none#无, random#随机, boxShrink#盒状收缩, boxSpread#盒状展开, cycleShrink#圆形收缩, cycleSpread#圆形展开, eraseUp#向上擦除, eraseDown#向下擦除, eraseLeft#向左擦除, eraseRight#向右擦除, verticalShelter#垂直屏蔽, horizontalShelter#水平屏蔽, verticalChessboard#纵向棋盘式, horizontalChessboard#横向棋盘式, dissolve#随机溶解, LeftRightToCenter#左右向中间缩进, centerToLeftRight#中央向左右扩展, upDownToCenter#上下向中央缩进, centerToUpDown#中央向上下扩展, drawOutLeftDown#从左上抽出, drawOutLeftUp#从左上抽出, drawOutRightDown#从右下抽出, drawOutRightUp#从右上抽出, verticalLine#随机垂直线条, horizontalLine#随机垂直线条]-->none
    </switchEffect>
    <backgroundPic>
      <!--opt, int, 背景图片素材ID-->1
    </backgroundPic>
    <SignInCfg>
      <!--opt, object, 签到信息, desc: 当页面用于签到或考勤界面时有效-->
      <VoiceBroadcast>
        <!--opt, object, 语音播报-->
        <enabled>
          <!--opt, bool, 是否启用语音播报-->true
        </enabled>
        <broadcastSpeed>
          <!--opt, enum, 播报语速, subType:string, [slower#慢, slow#较慢, medium#中速, fast#较快, faster#快]-->slower
        </broadcastSpeed>
        <inLibVoice>
          <!--opt, string, 在库人员正常体温语音播报内容-->test
        </inLibVoice>
        <outLibVoice>
          <!--opt, string, 未知人员正常体温语音播报内容-->test
        </outLibVoice>
        <inLibAbnormalTemperatureVoice>
          <!--opt, string, 在库人员异常温度语音播报内容-->test
        </inLibAbnormalTemperatureVoice>
        <outLibAbnormalTemperatureVoice>
          <!--opt, string, 未知人员异常温度语音播报内容-->test
        </outLibAbnormalTemperatureVoice>
      </VoiceBroadcast>
      <WelcomeWord>
        <!--opt, object, 欢迎词-->
        <enabled>
          <!--opt, bool, 是否启用欢迎词-->true
        </enabled>
        <inLibWord>
          <!--opt, string, 在库人员欢迎词内容-->test
        </inLibWord>
        <outLibWord>
          <!--opt, string, 未知人员欢迎词内容-->test
        </outLibWord>
      </WelcomeWord>
    </SignInCfg>
  </PageBasicInfo>

```

```
</WelcomeWord>
<SignCartoon>
  <!--opt, object, 签到动画-->
  <signCartoonSpeed>
    <!--opt, enum, 动画速度, subType:string, [fast#快,medium#终端,slow#慢]-->fast
  </signCartoonSpeed>
</SignCartoon>
<backgroundSignEnabled>
  <!--opt, bool, 是否启用后台签到-->true
</backgroundSignEnabled>
<TemperatureDetection>
  <!--opt, object, 温度监测参数-->
  <enabled>
    <!--req, bool, 是否启用-->true
  </enabled>
  <threshold>
    <!--req, float, 正常温度值-->0.000
  </threshold>
</TemperatureDetection>
</SignInCfg>
</PageBasicInfo>
<characterMode>
  <!--opt, enum, 页面中的欢迎词模式, subType:string, [mode1#模式1,mode2#模式2,mode3#模式3], desc:门禁增加, 门禁欢迎词节目位置固定, 且存在三种模式-->mode1
</characterMode>
<WindowsList>
  <!--opt, array, 窗口信息列表, subType:object-->
  <Windows>
    <!--opt, object, 窗口信息-->
    <id>
      <!--req, int, 内容编号-->1
    </id>
    <Position>
      <!--opt, object, 内容位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
      <positionX>
        <!--req, int, x坐标-->1
      </positionX>
      <positionY>
        <!--req, int, y坐标-->1
      </positionY>
      <height>
        <!--req, int, 宽-->1
      </height>
      <width>
        <!--req, int, 高-->1
      </width>
    </Position>
    <CornerRadius>
      <!--opt, object, 圆角配置, desc:目前仅对picture图片窗口和character文字窗口有效-->
      <TopLeft>
        <!--opt, int, 左上角半径-->1
      </TopLeft>
      <TopRight>
        <!--opt, int, 右上角半径-->1
      </TopRight>
      <BottomLeft>
        <!--opt, int, 左下角半径-->1
      </BottomLeft>
      <BottomRight>
        <!--opt, int, 右下角半径-->1
      </BottomRight>
    </CornerRadius>
    <layerNo>
      <!--req, int, 图层编号-->1
    </layerNo>
    <WinMaterialInfo>
      <!--opt, object, 窗口素材信息-->
      <materialType>
        <!--req, enum, 素材类型, subType:string, [static#静态,dynamic#动态,other#其他]-->static
      </materialType>
      <staticMaterialType>
        <!--opt, enum, 静态素材类型, subType:string, [picture#图片,flash#flash,audio#音频,video#视频,document#文本,ppt#ppt,doc#文档,excel#表格,pdf#pdf,web#网页,app#app,signIn#签到,conference#会议]-->picture
      </staticMaterialType>
      <dynamicType>
        <!--opt, enum, 动态素材类型, subType:string,
[web#web,socket#socket,rss#rss,realStream#realStream,call#call,dynamicPic#dynamicPic,capturePic#capturePic,character#character,chart#图表数据]-->web
      </dynamicType>
      <otherType>
        <!--opt, enum, 其他素材类型, subType:string, [clock#时钟,weather#天气,countdown#倒计时,localInput#本地输入,hyperLinkBtn#超链接按钮,event#事件,callBtn#呼叫按钮,openDoorBtn#开门按钮,QRCODEBtn#二维码按钮,authenticationSuccess#认证成功,authenticationFail#认证失败,statusBar#状态栏,ExamTemplate#考试模板]-->clock
      </otherType>
    </WinMaterialInfo>
    <TouchProperty>
      <!--opt, object, 触摸属性-->
      <windType>
        <!--opt, enum, 窗口类型, subType:string, [popup#弹窗,page#页面]-->popup
      </windType>
      <hyperLinkType>
        <!--opt, enum, 超链接类型, subType:string, [window#页面超链接,page#窗口超链接], desc:窗口类型为弹窗时有效-->window
      </hyperLinkType>
      <windowId>
        <!--opt, int, 窗口编号, desc:当前页面的窗口,超链接类型为窗口时有效-->1
      </windowId>
    </TouchProperty>
  </Windows>
</WindowsList>
```

```
</windowId>
<pageId>
  <!--opt, int, 页面编号, desc:超链接类型为页面时有效-->1
</pageId>
</TouchProperty>
<PlayItemList>
  <!--opt, array, 窗口播放列表, subType:object-->
  <PlayItem>
    <!--req, object, 窗口播放信息-->
    <id>
      <!--req, int, 播放序号-->1
    </id>
    <materialNo>
      <!--opt, int, 素材编号-->1
    </materialNo>
    <inputChannel>
      <!--opt, int, 绑定的IPC通道号-->1
    </inputChannel>
    <playEffect>
      <!--req, enum, 播放效果, subType:string, [none#无,marquee#跑马灯]-->none
    </playEffect>
    <MarqueeInfo>
      <!--opt, object, 跑马灯信息-->
      <scrollType>
        <!--req, enum, 跑马灯滚动类型, subType:string, [none#不滚动,Loops#循环滚动,once#滚动一次,backAndForth#来回滚动]-->none
      </scrollType>
      <scrollDeriction>
        <!--req, enum, 跑马灯滚动方向, subType:string, [none#无,up#向上,down#向下,left#向左,right#向右]-->none
      </scrollDeriction>
      <scrollSpeed>
        <!--req, int, 跑马灯滚动速度-->1
      </scrollSpeed>
    </MarqueeInfo>
    <PlayDuration>
      <!--opt, object, 窗口内素材播放时间, desc:静态素材可以配置该时间,视频直播和IPC通道都可配置该节点-->
      <durationType>
        <!--req, enum, 持续方式, subType:string, [materialTime#按素材时间,selfDefine#自定义,unlimited#无限时长]-->selfDefine
      </durationType>
      <duration>
        <!--req, int, 持续时间, unit:s, unitType:时间-->1
      </duration>
    </PlayDuration>
    <CharactersEffect>
      <!--opt, object, 文字显示效果, desc:当素材类型为文本txt和时有效-->
      <fontSize>
        <!--req, int, 字体大小-->1
      </fontSize>
      <fontColor>
        <!--req, object, 字体颜色-->
        <RGB>
          <!--opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
        </RGB>
      </FontColor>
      <backgroundType>
        <!--opt, enum, 背景类型, subType:string, [color#颜色(默认),picture#图片], desc:字段不存在时,默认为颜色-->color
      </backgroundType>
      <backPicId>
        <!--opt, int, 背景图片素材ID-->1
      </backPicId>
      <BackColor>
        <!--req, object, 背景颜色-->
        <RGB>
          <!--opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
        </RGB>
      </BackColor>
      <backTransparent>
        <!--req, int, 背景透明度-->1
      </backTransparent>
      <subtitlesEnabled>
        <!--req, bool, 子标题使能-->true
      </subtitlesEnabled>
      <scrollDirection>
        <!--req, enum, 文字滚动方向, subType:string, [left#向左,right#向右,up#向上,down#向下]-->left
      </scrollDirection>
      <scrollSpeed>
        <!--req, int, 文字滚动速度-->1
      </scrollSpeed>
    </CharactersEffect>
    <switchEffect>
      <!--opt, object, 窗口素材切换效果, subType:string, [none#无,leftInRightOut#左进右出,rightInLeftOut#右进左出,bottomInTopOut#下进上出,topInBottomOut#上进下出,fadeInFadeOut#淡入淡出,middleExit#中间退出,topPop#顶部弹出,rightBottomIn#右下角进入,leftTopIn#左上角进入,horizontalOpen#水平打开,verticalOpen#垂直打开,random#随机效果], desc:图片素材时有效-->random
    </switchEffect>
    <pageTime>
      <!--opt, int, 翻页时间, unit:s, unitType:时间, desc:当素材为word,ppt,pdf,excel时有效-->1
    </pageTime>
    <scrollSpeed>
      <!--req, int, 滚动速度, desc:当素材为静态web时有效-->1
    </scrollSpeed>
    <scaleType>
      <!--opt, enum, 素材缩放参数配置, subType:string, [fitXY#拉伸填充,fitCenter#自适应,centerCrop#截取填充], desc:目前仅当图片窗口有效-->centerCrop
    </scaleType>
    <playType>
      <!--opt, enum, 播放方式, subType:string, [normal#正常,faceAttendance#考勤,PeopleCounting#客流,temperature#测温], desc:目前仅当图片窗口有效-->normal
    </playType>
```

>normal

```
</playType>
<CharactersAttribute>
  <!--opt, object, 文字属性, desc:当dynamicType为character时有效-->
  <fontSize>
    <!--req, int, 字体大小-->1
  </fontSize>
  <FontColor>
    <!--req, object, 字体颜色-->
    <RGB>
      <!--opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
    </RGB>
  </FontColor>
  <backgroundType>
    <!--req, enum, 背景类型, subType:string, [color#颜色,picture#图片]-->color
  </backgroundType>
  <BackColor>
    <!--req, object, 背景色-->
    <RGB>
      <!--opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
    </RGB>
  </BackColor>
  <backTransparent>
    <!--opt, int, 背景透明度-->1
  </backTransparent>
  <alignType>
    <!--req, enum, 对齐方式, subType:string, [left#左,right#右,middle#中]-->middle
  </alignType>
  <verticalAlignType>
    <!--req, enum, 垂直对齐方式, subType:string, [top#顶,bottom#底,verticalCenter#中]-->top
  </verticalAlignType>
  <characterContent>
    <!--opt, string, 文字内容, range:[0,512]-->test
  </characterContent>
  <backPicId>
    <!--opt, int, 背景图片素材ID-->1
  </backPicId>
  <characterStyle>
    <!--req, enum, 字体效果, subType:string, [normal#常规,bold#粗体]-->normal
  </characterStyle>
  <characterType>
    <!--req, enum, 字体类型, subType:string, [mainTitle#主标题,subTitle#副标题]-->mainTitle
  </characterType>
</CharactersAttribute>
<WebProperty>
  <!--opt, object, 网页属性-->
  <webType>
    <!--opt, enum, 网页类型, subType:string, [desktop#桌面版,mobile#移动版], desc:默认按照设备的系统类型（Windows或Android）来显示网页-->desktop
  </webType>
</WebProperty>
</PlayItem>
</PlayItemList>
<enabledAudio>
  <!--opt, bool, 音频使能-->true
</enabledAudio>
<enableHide>
  <!--opt, bool, 隐藏使能-->true
</enableHide>
<enableLock>
  <!--opt, bool, 锁定使能-->true
</enableLock>
<AppWindow>
  <!--opt, object, App窗口信息-->
  <WindowInfoList>
    <!--req, array, 窗口信息列表, subType:object-->
    <WindowInfo>
      <!--req, object, 窗口信息-->
      <id>
        <!--req, int, 索引-->1
      </id>
      <materialNo>
        <!--req, int, 素材编号, desc:采用本字段下发时, 用户需要先将App安装包当做素材存储到设备中, 得到素材编号再进行下发-->1
      </materialNo>
      <path>
        <!--opt, string, App路径, range:[0,1024],
desc:path和materialNo字段同时存在时, 设备优先解析path字段;
采用本字段下发时, 设备需要确保App已经安装在设备内, 直接传入App的路径即可;
对于Android设备, path传的是App的包名; 对于Windows设备, path传的是App的应用程序绝对路径; -->test
      </path>
    </WindowInfo>
  </WindowInfoList>
</AppWindow>
<DataSource>
  <!--opt, object, 数据来源, desc:当窗口为叫号和动态弹图时有效-->
  <materialNo>
    <!--req, int, 素材编号-->1
  </materialNo>
  <dataSourceUUID>
    <!--opt, string, 数据源UUID-->1
  </dataSourceUUID>
</DataSource>
<Call>
  <!--opt, object, 叫号数据, desc:对应表格数据-->
```

```

<materialNo>
  <!--req, int, 素材编号-->1
</materialNo>
<tableRow>
  <!--req, int, 表格行-->1
</tableRow>
<tableColumn>
  <!--req, int, 表格列-->1
</tableColumn>
<tableDirection>
  <!--req, enum, 表格方向, subType:string, [vertical#纵向,horizontal#纵向]-->vertical
</tableDirection>
<tableType>
  <!--req, enum, 表格类型, subType:string, [template1#模板1,template2#模板2,template3#模板3,template4#模板4,template5#模板5,template6#模板6]-->template1
</tableType>
<backPicId>
  <!--opt, int, 背景图片素材编号-->1
</backPicId>
<alignType>
  <!--req, enum, 对齐方式, subType:string, [left#左,right#右,middle#中]-->left
</alignType>
<refreshDirection>
  <!--req, enum, 刷新方向, subType:string, [upTdown#从上到下,downToup#从下到上,LeftToright#从左到右,rightToLeft#从右到左]-->upTdown
</refreshDirection>
<HeadDataList>
  <!--opt, array, 表头数据列表, subType:object-->
  <HeadData>
    <!--opt, object, 表头数据-->
    <id>
      <!--req, int, 索引-->1
    </id>
    <data>
      <!--req, string, 数据-->test
    </data>
  </HeadData>
</HeadDataList>
<ItemStyleList>
  <!--opt, array, 表格列或行样式列表, subType:object-->
  <ItemStyle>
    <!--opt, object, 表格列或行样式-->
    <id>
      <!--req, int, 索引-->1
    </id>
    <width>
      <!--req, int, 每列宽所占百分比-->1
    </width>
    <fontSize>
      <!--req, int, 字体大小-->1
    </fontSize>
    <FontColor>
      <!--req, object, 字体颜色-->
      <RGB>
        <!--opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
      </RGB>
    </FontColor>
    <BackColor>
      <!--req, object, 背景色-->
      <RGB>
        <!--opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
      </RGB>
    </BackColor>
  </ItemStyle>
</ItemStyleList>
<playType>
  <!--opt, enum, 数据播放方式, subType:string, [roll#滚动播放,flipOver#翻页播放]-->roll
</playType>
</Call>
<DynamicPic>
  <!--opt, object, 动态弹图窗口配置-->
  <materialNo>
    <!--req, int, 素材编号-->1
  </materialNo>
  <backPicId>
    <!--opt, int, 背景图片素材ID-->1
  </backPicId>
</DynamicPic>
<CapturePic>
  <!--opt, object, 抓拍图片参数-->
  <backPicId>
    <!--opt, int, 背景图片素材ID-->1
  </backPicId>
  <ipcMaterialNo>
    <!--req, int, IPC素材编号-->1
  </ipcMaterialNo>
  <cancelType>
    <!--req, enum, 取消方式, subType:string, [auto#自动>manual#手动]-->>manual
  </cancelType>
  <duration>
    <!--opt, int, 持续时间, unit:s, unitType:时间-->1
  </duration>
</CapturePic>
<ClockParam>
  <!--opt, object, 时钟参数-->

```

```
<!--opt, object, 时钟参数-->
<backPicId>
  <!--opt, int, 背景图片素材ID-->1
</backPicId>
<ClockIcon>
  <!--req, object, 时钟图标参数-->
  <enabled>
    <!--req, bool, 使能-->true
  </enabled>
  <type>
    <!--opt, enum, 时钟类型, subType:string, [cLock1#cLock1,cLock2#cLock2]-->cLock1
  </type>
  <Position>
    <!--opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--req, int, x坐标-->1
    </positionX>
    <positionY>
      <!--req, int, y坐标-->1
    </positionY>
    <height>
      <!--req, int, 宽-->1
    </height>
    <width>
      <!--req, int, 高-->1
    </width>
  </Position>
</ClockIcon>
<YmdParam>
  <!--req, object, 时钟年月日参数-->
  <enabled>
    <!--req, bool, 使能-->true
  </enabled>
  <fontSize>
    <!--req, int, 字体大小-->1
  </fontSize>
  <FontColor>
    <!--req, object, 字体颜色-->
    <RGB>
      <!--opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
    </RGB>
  </FontColor>
  <BackColor>
    <!--req, object, 背景色-->
    <RGB>
      <!--opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
    </RGB>
  </BackColor>
  <Position>
    <!--opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--req, int, x坐标-->1
    </positionX>
    <positionY>
      <!--req, int, y坐标-->1
    </positionY>
    <height>
      <!--req, int, 宽-->1
    </height>
    <width>
      <!--req, int, 高-->1
    </width>
  </Position>
</YmdParam>
<HmsParam>
  <!--req, object, 时钟时分秒参数-->
  <enabled>
    <!--req, bool, 使能-->true
  </enabled>
  <fontSize>
    <!--req, int, 字体大小-->1
  </fontSize>
  <FontColor>
    <!--req, object, 字体颜色-->
    <RGB>
      <!--opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
    </RGB>
  </FontColor>
  <BackColor>
    <!--req, object, 背景色-->
    <RGB>
      <!--opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
    </RGB>
  </BackColor>
  <Position>
    <!--opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--req, int, x坐标-->1
    </positionX>
    <positionY>
      <!--req, int, y坐标-->1
    </positionY>
    <height>
      <!--req, int, 宽-->1
    </height>
```

```
</height>
<width>
  <!--req, int, 高-->1
</width>
</Position>
</HmsParam>
<WeekParam>
  <!--req, object, 时钟星期参数-->
  <enabled>
    <!--req, bool, 使能-->true
  </enabled>
  <fontSize>
    <!--req, int, 字体大小-->1
  </fontSize>
  <FontColor>
    <!--req, object, 字体颜色-->
    <RGB>
      <!--opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
    </RGB>
  </FontColor>
  <BackColor>
    <!--req, object, 背景色-->
    <RGB>
      <!--opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
    </RGB>
  </BackColor>
  <Position>
    <!--opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--req, int, x坐标-->1
    </positionX>
    <positionY>
      <!--req, int, y坐标-->1
    </positionY>
    <height>
      <!--req, int, 宽-->1
    </height>
    <width>
      <!--req, int, 高-->1
    </width>
  </Position>
</WeekParam>
</ClockParam>
<WeatherParam>
  <!--opt, object, 天气参数-->
  <backPicId>
    <!--opt, int, 天气背景图片索引id-->1
  </backPicId>
  <WeatherIcon>
    <!--opt, object, 天气图标参数-->
    <enabled>
      <!--req, bool, 使能-->true
    </enabled>
    <Position>
      <!--opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
      <positionX>
        <!--req, int, x坐标-->1
      </positionX>
      <positionY>
        <!--req, int, y坐标-->1
      </positionY>
      <height>
        <!--req, int, 宽-->1
      </height>
      <width>
        <!--req, int, 高-->1
      </width>
    </Position>
  </WeatherIcon>
  <Date>
    <!--opt, object, 天气日期参数-->
    <enabled>
      <!--req, bool, 使能-->true
    </enabled>
    <fontSize>
      <!--req, int, 字体大小-->1
    </fontSize>
    <FontColor>
      <!--req, object, 字体颜色-->
      <RGB>
        <!--opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
      </RGB>
    </FontColor>
    <BackColor>
      <!--req, object, 背景色-->
      <RGB>
        <!--opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
      </RGB>
    </BackColor>
    <Position>
      <!--opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
      <positionX>
        <!--req, int, x坐标-->1
      </positionX>
```

```
</positionX>
<positionY>
  <!--req, int, y坐标-->1
</positionY>
<height>
  <!--req, int, 宽-->1
</height>
<width>
  <!--req, int, 高-->1
</width>
</Position>
</Date>
<Temperature>
  <!--opt, object, 天气温度参数-->
  <enabled>
    <!--req, bool, 使能-->true
  </enabled>
  <fontSize>
    <!--req, int, 字体大小-->1
  </fontSize>
  <FontColor>
    <!--req, object, 字体颜色-->
    <RGB>
      <!--opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
    </RGB>
  </FontColor>
  <BackColor>
    <!--req, object, 背景色-->
    <RGB>
      <!--opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
    </RGB>
  </BackColor>
  <Position>
    <!--opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--req, int, x坐标-->1
    </positionX>
    <positionY>
      <!--req, int, y坐标-->1
    </positionY>
    <height>
      <!--req, int, 宽-->1
    </height>
    <width>
      <!--req, int, 高-->1
    </width>
  </Position>
</Temperature>
<WeatherContent>
  <!--opt, object, 天气情况参数-->
  <enabled>
    <!--req, bool, 使能-->true
  </enabled>
  <fontSize>
    <!--req, int, 字体大小-->1
  </fontSize>
  <FontColor>
    <!--req, object, 字体颜色-->
    <RGB>
      <!--opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
    </RGB>
  </FontColor>
  <BackColor>
    <!--req, object, 背景色-->
    <RGB>
      <!--opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
    </RGB>
  </BackColor>
  <Position>
    <!--opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--req, int, x坐标-->1
    </positionX>
    <positionY>
      <!--req, int, y坐标-->1
    </positionY>
    <height>
      <!--req, int, 宽-->1
    </height>
    <width>
      <!--req, int, 高-->1
    </width>
  </Position>
</WeatherContent>
<City>
  <!--opt, object, 天气城市参数-->
  <enabled>
    <!--req, bool, 使能-->true
  </enabled>
  <cityId>
    <!--req, string, 城市编号-->test
  </cityId>
  <cityName>
    <!--req, string, 城市名称-->test
```

```
</cityName>
<fontSize>
  <!--req, int, 字体大小-->1
</fontSize>
<FontColor>
  <!--req, object, 字体颜色-->
  <RGB>
    <!--opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
  </RGB>
</FontColor>
<BackColor>
  <!--req, object, 背景色-->
  <RGB>
    <!--opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
  </RGB>
</BackColor>
<Position>
  <!--opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
  <positionX>
    <!--req, int, x坐标-->1
  </positionX>
  <positionY>
    <!--req, int, y坐标-->1
  </positionY>
  <height>
    <!--req, int, 宽-->1
  </height>
  <width>
    <!--req, int, 高-->1
  </width>
</Position>
</City>
<Humidity>
  <!--opt, object, 天气湿度参数-->
  <enabled>
    <!--req, bool, 使能-->true
  </enabled>
  <fontSize>
    <!--req, int, 字体大小-->1
  </fontSize>
  <FontColor>
    <!--req, object, 字体颜色-->
    <RGB>
      <!--opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
    </RGB>
  </FontColor>
  <BackColor>
    <!--req, object, 背景色-->
    <RGB>
      <!--opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
    </RGB>
  </BackColor>
  <Position>
    <!--opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--req, int, x坐标-->1
    </positionX>
    <positionY>
      <!--req, int, y坐标-->1
    </positionY>
    <height>
      <!--req, int, 宽-->1
    </height>
    <width>
      <!--req, int, 高-->1
    </width>
  </Position>
</Humidity>
<AirQuality>
  <!--opt, object, 天气空气质量参数-->
  <enabled>
    <!--req, bool, 使能-->true
  </enabled>
  <fontSize>
    <!--req, int, 字体大小-->1
  </fontSize>
  <FontColor>
    <!--req, object, 字体颜色-->
    <RGB>
      <!--opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
    </RGB>
  </FontColor>
  <BackColor>
    <!--req, object, 背景色-->
    <RGB>
      <!--opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
    </RGB>
  </BackColor>
  <Position>
    <!--opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--req, int, x坐标-->1
    </positionX>
```

```
<positionY>
  <!--req, int, y坐标-->1
</positionY>
<height>
  <!--req, int, 宽-->1
</height>
<width>
  <!--req, int, 高-->1
</width>
</Position>
</AirQuality>
<UpdateTime>
  <!--opt, object, 天气更新时间参数-->
  <enabled>
    <!--req, bool, 使能-->true
  </enabled>
  <refreshTime>
    <!--req, string, 刷新时间, desc:设备早期返回时间格式有误, 错为: 20200826T051500+08, 为保证XSD校验通过, 改为string类型-->20200826T051500+08
  </refreshTime>
  <updateInterval>
    <!--req, int, 刷新间隔, unit:minute-->1
  </updateInterval>
  <fontSize>
    <!--req, int, 字体大小-->1
  </fontSize>
  <FontColor>
    <!--req, object, 字体颜色-->
    <RGB>
      <!--opt, int, RGB值, desc:十进制数据表示三原色, 如16777215表示0xFFFFFF-->1
    </RGB>
  </FontColor>
  <BackColor>
    <!--req, object, 背景色-->
    <RGB>
      <!--opt, int, RGB值, desc:十进制数据表示三原色, 如16777215表示0xFFFFFF-->1
    </RGB>
  </BackColor>
  <Position>
    <!--opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--req, int, x坐标-->1
    </positionX>
    <positionY>
      <!--req, int, y坐标-->1
    </positionY>
    <height>
      <!--req, int, 宽-->1
    </height>
    <width>
      <!--req, int, 高-->1
    </width>
  </Position>
</UpdateTime>
<Wind>
  <!--opt, object, 天气风力参数-->
  <enabled>
    <!--req, bool, 使能-->true
  </enabled>
  <fontSize>
    <!--req, int, 字体大小-->1
  </fontSize>
  <FontColor>
    <!--req, object, 字体颜色-->
    <RGB>
      <!--opt, int, RGB值, desc:十进制数据表示三原色, 如16777215表示0xFFFFFF-->1
    </RGB>
  </FontColor>
  <BackColor>
    <!--req, object, 背景色-->
    <RGB>
      <!--opt, int, RGB值, desc:十进制数据表示三原色, 如16777215表示0xFFFFFF-->1
    </RGB>
  </BackColor>
  <Position>
    <!--opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--req, int, x坐标-->1
    </positionX>
    <positionY>
      <!--req, int, y坐标-->1
    </positionY>
    <height>
      <!--req, int, 宽-->1
    </height>
    <width>
      <!--req, int, 高-->1
    </width>
  </Position>
</Wind>
</WeatherParam>
<Countdown>
  <!--opt, object, 倒计时素材-->
  <endTime>
    <!--req, string, 倒计时时间点, desc:设备早期返回时间格式有误, 错为: 20200826T051500+08, 为保证XSD校验通过, 改为string类型-->20200826T051500+08
```

```
</endTime>
<template>
  <!--req, enum, 模板类型, subType:string, [template1#模板1,template2#模板2,openingActivity1#活动开幕1,openingActivity2#活动开幕
2,commercialActivity1#商业活动1,commercialActivity2#商业活动2,collegeEntranceExam#高考,overseasTimeFormat#海外时间格式]-->template1
</template>
<timeUnit>
  <!--req, enum, 时间单位, subType:string, [year#年,month#月,day#日,week#周,hour#时,minute#分,second#秒]-->year
</timeUnit>
<backPicId>
  <!--opt, int, 背景图片素材ID-->1
</backPicId>
<TimeFontCfg>
  <!--opt, object, 时间字体参数-->
  <fontSize>
    <!--req, int, 字体大小-->1
  </fontSize>
  <fontColor>
    <!--req, object, 字体颜色-->
    <RGB>
      <!--opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
    </RGB>
  </fontColor>
  <Position>
    <!--opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--req, int, x坐标-->1
    </positionX>
    <positionY>
      <!--req, int, y坐标-->1
    </positionY>
    <height>
      <!--req, int, 宽-->1
    </height>
    <width>
      <!--req, int, 高-->1
    </width>
  </Position>
</TimeFontCfg>
<timeUnitEnabled>
  <!--opt, bool, 时间单位是否有效-->true
</timeUnitEnabled>
<maxTimeUnit>
  <!--req, enum, 最大时间单位, subType:string, [year#年,month#月,day#日,week#周,hour#时,minute#分,second#秒,week#周], desc:表示当前倒计时时的最大显示
单元,例如:maxTimeUnit为月,则1年3个月,对应月份值为15-->year
</maxTimeUnit>
<ModuleInfoList>
  <!--opt, array, 界面模块信息列表, subType:object-->
  <ModuleInfo>
    <!--opt, object, 界面模块信息-->
    <moduleType>
      <!--req, enum, 模块类型, subType:string, [year#年,month#月,day#日,hour#时,minute#分,second#秒,week#周,text#文字], desc:text为时间旁边的文字,例
如年下面的文字"年"-->second
    </moduleType>
    <enabled>
      <!--req, bool, 使能-->true
    </enabled>
    <fontSize>
      <!--req, int, 字体大小-->1
    </fontSize>
    <fontColor>
      <!--req, object, 字体颜色-->
      <RGB>
        <!--opt, int, 十进制数据表示三原色-->1
      </RGB>
    </fontColor>
    <text>
      <!--opt, string, 文字, desc:当moduleType为text时有效-->test
    </text>
    <Position>
      <!--opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
      <positionX>
        <!--req, int, X坐标-->1
      </positionX>
      <positionY>
        <!--req, int, Y坐标-->1
      </positionY>
      <height>
        <!--req, int, 宽-->1
      </height>
      <width>
        <!--req, int, 高-->1
      </width>
    </Position>
  </ModuleInfo>
</ModuleInfoList>
</Countdown>
<localInputNo>
  <!--req, enum, 本地输入类型, subType:string, [VGA#VGA,HDMI#HDMI]-->VGA
</localInputNo>
<HyperlinkBtn>
  <!--opt, object, 超链接按钮-->
  <backPicId>
    <!--opt, int, 背景图片素材ID-->1
```

```
</backPicId>
</HyperlinkBtn>
<CharactersAttribute>
  <!--opt, object, 文字属性, desc: 当dynamicType为character时有效-->
  <fontSize>
    <!--req, int, 字体大小-->1
  </fontSize>
  <fontColor>
    <!--req, object, 字体颜色-->
    <RGB>
      <!--opt, int, RGB值, desc: 十进制数据表示三原色, 如16777215表示0xFFFFFF-->1
    </RGB>
  </fontColor>
  <fontType>
    <!--opt, enum, 字体类型, subType:string, [normal#常规,bold#粗体]-->normal
  </fontType>
  <backgroundType>
    <!--req, enum, 背景类型, subType:string, [color#颜色,picture#图片]-->color
  </backgroundType>
  <BackColor>
    <!--req, object, 背景色-->
    <RGB>
      <!--opt, int, RGB值, desc: 十进制数据表示三原色, 如16777215表示0xFFFFFF-->1
    </RGB>
  </BackColor>
  <fillEnabled>
    <!--opt, bool, 背景色填充使能, desc: 默认为true, 表示背景色填充-->true
  </fillEnabled>
  </BackColor>
  <backTransparent>
    <!--opt, int, 背景透明度-->1
  </backTransparent>
  <alignType>
    <!--req, enum, 对齐方式, subType:string, [Left#左,right#右,middle#中]-->middle
  </alignType>
  <verticalAlignType>
    <!--req, enum, 垂直对齐方式, subType:string, [top#顶,bottom#底,verticalCenter#中]-->top
  </verticalAlignType>
  <characterContent>
    <!--opt, string, 文字内容, range:[0,512]-->test
  </characterContent>
  <backPicId>
    <!--opt, int, 背景图片素材ID-->1
  </backPicId>
  <characterStyle>
    <!--req, enum, 字体效果, subType:string, [normal#常规,bold#粗体]-->normal
  </characterStyle>
  <characterType>
    <!--req, enum, 字体类型, subType:string, [mainTitle#主标题,subTitle#副标题]-->mainTitle
  </characterType>
  <scrollDirection>
    <!--opt, enum, 文字滚动方向, subType:string, [Left#向左,right#向右,up#向上,down#向下,turnPage#翻页]-->left
  </scrollDirection>
  <scrollSpeed>
    <!--opt, int, 文字滚动速度, range:[1,10]-->1
  </scrollSpeed>
</CharactersAttribute>
<SignIn>
  <!--opt, object, 人员签到展示窗口, desc: 当staticMaterialType为signIn时有效-->
  <backPicId>
    <!--opt, int, 背景图片素材ID-->1
  </backPicId>
  <ShowInfo>
    <!--opt, object, 展示信息-->
    <row>
      <!--opt, int, 窗口内人员签到展示行数-->1
    </row>
    <column>
      <!--opt, int, 窗口内人员签到展示列数-->1
    </column>
    <Position>
      <!--opt, object, 位置, desc: 左上角为坐标原点, 取值范围和设备分辨率保持一致-->
      <positionX>
        <!--req, int, x坐标-->1
      </positionX>
      <positionY>
        <!--req, int, y坐标-->1
      </positionY>
      <height>
        <!--req, int, 宽-->1
      </height>
      <width>
        <!--req, int, 高-->1
      </width>
    </Position>
  </ShowInfo>
  <inLib>
    <!--opt, object, 在库信息-->
    <nameEnabled>
      <!--opt, bool, 在库人员是否展示姓名-->true
    </nameEnabled>
    <genderEnabled>
      <!--opt, bool, 在库人员是否展示性别-->true
    </genderEnabled>
    <selfDefine1Enabled>
      <!--opt, bool, 在库人员是否展示自定义字段1-->true
    </selfDefine1Enabled>
  </inLib>
</SignIn>
```

```
<!--opt, bool, 在库人员是否展示自定义字段1-->true
</selfDefine1Enabled>
<selfDefine2Enabled>
  <!--opt, bool, 在库人员是否展示自定义字段2-->true
</selfDefine2Enabled>
<signInTimeEnabled>
  <!--opt, bool, 在库人员是否展示签到时间-->true
</signInTimeEnabled>
<temperatureEnabled>
  <!--opt, bool, 在库人员是否展示温度-->true
</temperatureEnabled>
</inLib>
<outLib>
  <!--opt, object, 不在库信息-->
  <nameEnabled>
    <!--opt, bool, 未知人员是否展示姓名-->true
  </nameEnabled>
  <name>
    <!--opt, string, 未知人员姓名-->test
  </name>
  <genderEnabled>
    <!--opt, bool, 未知人员是否展示性别-->true
  </genderEnabled>
  <signInTimeEnabled>
    <!--opt, bool, 未知人员是否展示签到时间-->true
  </signInTimeEnabled>
  <temperatureEnabled>
    <!--opt, bool, 未知人员是否展示温度-->true
  </temperatureEnabled>
</outLib>
<showDuration>
  <!--req, enum, 签到记录显示时间, subType:string, [always#不消失,10s#10s,20s#20s,30s#30s,no#不显示]-->always
</showDuration>
</ShowInfo>
<MainTitle>
  <!--opt, object, 主标题-->
  <enabled>
    <!--req, bool, 使能-->true
  </enabled>
  <fontSize>
    <!--req, int, 字体大小-->1
  </fontSize>
  <FontColor>
    <!--req, object, 字体颜色-->
    <RGB>
      <!--opt, int, 十进制数据表示三原色-->1
    </RGB>
  </FontColor>
  <Position>
    <!--opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--req, int, x坐标-->1
    </positionX>
    <positionY>
      <!--req, int, y坐标-->1
    </positionY>
    <height>
      <!--req, int, 宽-->1
    </height>
    <width>
      <!--req, int, 高-->1
    </width>
  </Position>
  <text>
    <!--opt, string, 标题文字-->test
  </text>
</MainTitle>
<SubTitle>
  <!--opt, object, 副标题-->
  <enabled>
    <!--req, bool, 使能-->true
  </enabled>
  <fontSize>
    <!--req, int, 字体大小-->1
  </fontSize>
  <FontColor>
    <!--req, object, 字体颜色-->
    <RGB>
      <!--opt, int, 十进制数据表示三原色-->1
    </RGB>
  </FontColor>
  <Position>
    <!--opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--req, int, x坐标-->1
    </positionX>
    <positionY>
      <!--req, int, y坐标-->1
    </positionY>
    <height>
      <!--req, int, 宽-->1
    </height>
    <width>
      <!--req, int, 高-->1
    </width>
```

```
</width>
</Position>
<text>
  <!--opt, string, 标题文字-->test
</text>
</SubTitle>
<NormalTemperature>
  <!--opt, object, 正常体温信息-->
  <PeopleNumber>
    <!--opt, object, 正常体温人数信息-->
    <enabled>
      <!--req, bool, 使能-->true
    </enabled>
    <fontSize>
      <!--req, int, 字体大小-->1
    </fontSize>
    <FontColor>
      <!--req, object, 字体颜色-->
      <RGB>
        <!--opt, int, 十进制数据表示三原色-->1
      </RGB>
    </FontColor>
    <Position>
      <!--opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
      <positionX>
        <!--req, int, X坐标-->1
      </positionX>
      <positionY>
        <!--req, int, Y坐标-->1
      </positionY>
      <height>
        <!--req, int, 宽-->1
      </height>
      <width>
        <!--req, int, 高-->1
      </width>
    </Position>
    <text>
      <!--opt, string, 标题文字-->test
    </text>
  </PeopleNumber>
  <Icon>
    <!--opt, object, 正常体温图标-->
    <enabled>
      <!--req, bool, 使能-->true
    </enabled>
    <Position>
      <!--opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
      <positionX>
        <!--req, int, X坐标-->1
      </positionX>
      <positionY>
        <!--req, int, Y坐标-->1
      </positionY>
      <height>
        <!--req, int, 宽-->1
      </height>
      <width>
        <!--req, int, 高-->1
      </width>
    </Position>
    <materialNo>
      <!--opt, int, 素材ID-->1
    </materialNo>
  </Icon>
</NormalTemperature>
<AbnormalTemperature>
  <!--opt, object, 异常体温信息-->
  <PeopleNumber>
    <!--opt, object, 异常体温人数信息-->
    <enabled>
      <!--req, bool, 使能-->true
    </enabled>
    <fontSize>
      <!--req, int, 字体大小-->1
    </fontSize>
    <FontColor>
      <!--req, object, 字体颜色-->
      <RGB>
        <!--opt, int, 十进制数据表示三原色-->1
      </RGB>
    </FontColor>
    <Position>
      <!--opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
      <positionX>
        <!--req, int, X坐标-->1
      </positionX>
      <positionY>
        <!--req, int, Y坐标-->1
      </positionY>
      <height>
        <!--req, int, 宽-->1
      </height>
```

```
<width>
  <!--req, int, 高-->1
</width>
</Position>
<text>
  <!--opt, string, 标题文字-->test
</text>
</PeopleNumber>
<Icon>
  <!--opt, object, 异常体温图标-->
  <enabled>
    <!--req, bool, 使能-->true
  </enabled>
  <Position>
    <!--opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--req, int, X坐标-->1
    </positionX>
    <positionY>
      <!--req, int, Y坐标-->1
    </positionY>
    <height>
      <!--req, int, 宽-->1
    </height>
    <width>
      <!--req, int, 高-->1
    </width>
  </Position>
  <materialNo>
    <!--opt, int, 素材ID-->1
  </materialNo>
</Icon>
</AbnormalTemperature>
</SignIn>
<EventParam>
  <!--opt, object, 事件参数-->
  <EventResourceList>
    <!--opt, array, 事件源信息列表, subType:object-->
    <EventResource>
      <!--opt, object, 事件源信息-->
      <ipVersion>
        <!--req, enum, ip地址版本, subType:string, [v4#v4,v6#v6,dual#双]-->v4
      </ipVersion>
      <ipAddress>
        <!--opt, string, ipv4地址-->test
      </ipAddress>
      <ipv6Address>
        <!--opt, string, ipv6地址-->test
      </ipv6Address>
      <portNo>
        <!--req, int, 端口号-->1
      </portNo>
      <userName>
        <!--req, string, 用户名-->test
      </userName>
      <passWord>
        <!--req, string, 密码-->test
      </passWord>
      <channels>
        <!--opt, string, 通道号, desc:多选以逗号隔开-->1,2
      </channels>
    </EventResource>
  </EventResourceList>
  <backPicId>
    <!--opt, int, 背景图片素材ID-->1
  </backPicId>
  <Title>
    <!--opt, object, 标题-->
    <enabled>
      <!--req, bool, 使能-->true
    </enabled>
    <fontSize>
      <!--req, int, 字体大小-->1
    </fontSize>
    <FontColor>
      <!--req, object, 字体颜色-->
      <RGB>
        <!--opt, int, 十进制数据表示三原色-->1
      </RGB>
    </FontColor>
    <Position>
      <!--opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
      <positionX>
        <!--req, int, X坐标-->1
      </positionX>
      <positionY>
        <!--req, int, Y坐标-->1
      </positionY>
      <height>
        <!--req, int, 宽-->1
      </height>
      <width>
        <!--req, int, 高-->1
      </width>
```

```

</Position>
<text>
  <!--opt, string, 标题文字-->test
</text>
</Title>
<PeopleCounting>
  <!--opt, object, 人数统计提醒-->
  <maxPeopleNumber>
    <!--opt, int, 最大可进入人数-->1
  </maxPeopleNumber>
  <unavailableBackPicId>
    <!--opt, int, 人员不能进入时控件背景图片id-->1
  </unavailableBackPicId>
  <supportChangeByPeople>
    <!--req, enum, 支持根据人数修改样式的字段, subType:string, [prompt#prompt,peopleIcon#peopleIcon,availablePeopleNumber#availablePeopleNumber]-->
    <supportChangeByPeople>
  </supportChangeByPeople>
  <Prompt>
    <!--opt, object, 提示信息-->
    <enabled>
      <!--req, bool, 使能-->true
    </enabled>
    <fontSize>
      <!--req, int, 字体大小-->1
    </fontSize>
    <Position>
      <!--opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
      <positionX>
        <!--req, int, X坐标-->1
      </positionX>
      <positionY>
        <!--req, int, Y坐标-->1
      </positionY>
      <height>
        <!--req, int, 宽-->1
      </height>
      <width>
        <!--req, int, 高-->1
      </width>
    </Position>
    <text>
      <!--opt, string, 标题文字-->test
    </text>
    <FontColor>
      <!--opt, object, 人员可以进入时的字体颜色-->
      <RGB>
        <!--opt, int, 十进制数据表示三原色-->1
      </RGB>
    </FontColor>
    <unavailableText>
      <!--opt, string, 人员不能进入时的标题文字-->test
    </unavailableText>
    <unavailableFontColor>
      <!--opt, object, 人员不能进入时的字体颜色-->
      <RGB>
        <!--opt, int, 十进制数据表示三原色-->1
      </RGB>
    </unavailableFontColor>
  </Prompt>
  <PeopleIcon>
    <!--opt, object, 行人通行图标-->
    <enabled>
      <!--req, bool, 使能-->true
    </enabled>
    <Position>
      <!--opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
      <positionX>
        <!--req, int, X坐标-->1
      </positionX>
      <positionY>
        <!--req, int, Y坐标-->1
      </positionY>
      <height>
        <!--req, int, 宽-->1
      </height>
      <width>
        <!--req, int, 高-->1
      </width>
    </Position>
    <materialNo>
      <!--opt, int, 素材ID-->1
    </materialNo>
  </PeopleIcon>
  <InsidePeopleTitle>
    <!--opt, object, 已经进入人数标题-->
    <enabled>
      <!--req, bool, 使能-->true
    </enabled>
    <fontSize>
      <!--req, int, 字体大小-->1
    </fontSize>
    <FontColor>
      <!--req, object, 字体颜色-->

```

```
<RGB>
  <!--opt, int, 十进制数据表示三原色-->1
</RGB>
</FontColor>
<Position>
  <!--opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
  <positionX>
    <!--req, int, X坐标-->1
  </positionX>
  <positionY>
    <!--req, int, Y坐标-->1
  </positionY>
  <height>
    <!--req, int, 宽-->1
  </height>
  <width>
    <!--req, int, 高-->1
  </width>
</Position>
<text>
  <!--opt, string, 标题文字-->test
</text>
</InsidePeopleTitle>
<InsidePeopleNumber>
  <!--opt, object, 已经进入人数-->
  <enabled>
    <!--req, bool, 使能-->true
  </enabled>
  <fontSize>
    <!--req, int, 字体大小-->1
  </fontSize>
  <FontColor>
    <!--req, object, 字体颜色-->
    <RGB>
      <!--opt, int, 十进制数据表示三原色-->1
    </RGB>
  </FontColor>
  <Position>
    <!--opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--req, int, X坐标-->1
    </positionX>
    <positionY>
      <!--req, int, Y坐标-->1
    </positionY>
    <height>
      <!--req, int, 宽-->1
    </height>
    <width>
      <!--req, int, 高-->1
    </width>
  </Position>
</InsidePeopleNumber>
<AvailablePeopleTitle>
  <!--opt, object, 还可以进入人数标题-->
  <enabled>
    <!--req, bool, 使能-->true
  </enabled>
  <fontSize>
    <!--req, int, 字体大小-->1
  </fontSize>
  <FontColor>
    <!--req, object, 字体颜色-->
    <RGB>
      <!--opt, int, 十进制数据表示三原色-->1
    </RGB>
  </FontColor>
  <Position>
    <!--opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--req, int, X坐标-->1
    </positionX>
    <positionY>
      <!--req, int, Y坐标-->1
    </positionY>
    <height>
      <!--req, int, 宽-->1
    </height>
    <width>
      <!--req, int, 高-->1
    </width>
  </Position>
  <text>
    <!--opt, string, 标题文字-->test
  </text>
</AvailablePeopleTitle>
<AvailablePeopleNumber>
  <!--opt, object, 还可以进入人数-->
  <enabled>
    <!--req, bool, 使能-->true
  </enabled>
  <fontSize>
    <!--req, int, 字体大小-->1
  </fontSize>
```

```
</FontColor>
<Position>
  <!--opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
  <positionX>
    <!--req, int, X坐标-->1
  </positionX>
  <positionY>
    <!--req, int, Y坐标-->1
  </positionY>
  <height>
    <!--req, int, 宽-->1
  </height>
  <width>
    <!--req, int, 高-->1
  </width>
</Position>
<FontColor>
  <!--opt, object, 人员可以进入时的字体颜色-->
  <RGB>
    <!--opt, int, 十进制数据表示三原色-->1
  </RGB>
</FontColor>
<UnavailableFontColor>
  <!--opt, object, 人员不能进入时的字体颜色-->
  <RGB>
    <!--opt, int, 十进制数据表示三原色-->1
  </RGB>
</UnavailableFontColor>
</AvailablePeopleNumber>
<UnavailablePeopleIcon>
  <!--opt, object, 人员不能进入时的行人图标-->
  <enabled>
    <!--req, bool, 使能-->true
  </enabled>
  <Position>
    <!--opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--req, int, X坐标-->1
    </positionX>
    <positionY>
      <!--req, int, Y坐标-->1
    </positionY>
    <height>
      <!--req, int, 宽-->1
    </height>
    <width>
      <!--req, int, 高-->1
    </width>
  </Position>
  <materialNo>
    <!--opt, int, 素材ID-->1
  </materialNo>
</UnavailablePeopleIcon>
<AudioAlarm>
  <!--opt, object, 音频报警信息-->
  <enabled>
    <!--req, bool, 使能-->true
  </enabled>
  <materialNo>
    <!--opt, int, 音频素材ID-->1
  </materialNo>
</AudioAlarm>
</PeopleCounting>
<Mask>
  <!--opt, object, 未戴口罩提醒-->
  <enabled>
    <!--req, bool, 使能-->true
  </enabled>
  <fontSize>
    <!--req, int, 字体大小-->1
  </fontSize>
  <FontColor>
    <!--req, object, 字体颜色-->
    <RGB>
      <!--opt, int, 十进制数据表示三原色-->1
    </RGB>
  </FontColor>
  <BackColor>
    <!--req, object, 背景颜色-->
    <RGB>
      <!--opt, int, 十进制数据表示三原色-->1
    </RGB>
  </BackColor>
  <Position>
    <!--opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--req, int, X坐标-->1
    </positionX>
    <positionY>
      <!--req, int, Y坐标-->1
    </positionY>
    <height>
      <!--req, int, 宽-->1
    </height>
```

```
<width>
  <!--req, int, 高-->1
</width>
</Position>
<text>
  <!--opt, string, 标题文字-->test
</text>
<MaskIcon>
  <!--opt, object, 未戴口罩图标-->
  <enabled>
    <!--req, bool, 使能-->true
  </enabled>
  <Position>
    <!--opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--req, int, X坐标-->1
    </positionX>
    <positionY>
      <!--req, int, Y坐标-->1
    </positionY>
    <height>
      <!--req, int, 宽-->1
    </height>
    <width>
      <!--req, int, 高-->1
    </width>
  </Position>
  <materialNo>
    <!--opt, int, 素材ID-->1
  </materialNo>
</MaskIcon>
<AudioAlarm>
  <!--opt, object, 音频报警信息-->
  <enabled>
    <!--req, bool, 使能-->true
  </enabled>
  <materialNo>
    <!--opt, int, 音频素材ID-->1
  </materialNo>
</AudioAlarm>
</Mask>
<Temperature>
  <!--opt, object, 温度异常提醒-->
  <Prompt>
    <!--opt, object, 提示信息-->
    <enabled>
      <!--req, bool, 使能-->true
    </enabled>
    <fontSize>
      <!--req, int, 字体大小-->1
    </fontSize>
    <fontColor>
      <!--req, object, 字体颜色-->
      <RGB>
        <!--opt, int, 十进制数据表示三原色-->1
      </RGB>
    </fontColor>
    <BackColor>
      <!--req, object, 背景颜色-->
      <RGB>
        <!--opt, int, 十进制数据表示三原色-->1
      </RGB>
    </BackColor>
    <Position>
      <!--opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
      <positionX>
        <!--req, int, X坐标-->1
      </positionX>
      <positionY>
        <!--req, int, Y坐标-->1
      </positionY>
      <height>
        <!--req, int, 宽-->1
      </height>
      <width>
        <!--req, int, 高-->1
      </width>
    </Position>
    <text>
      <!--opt, string, 标题文字-->test
    </text>
  </Prompt>
  <TemperatureIcon>
    <!--opt, object, 温度异常图标-->
    <enabled>
      <!--req, bool, 使能-->true
    </enabled>
    <Position>
      <!--opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
      <positionX>
        <!--req, int, X坐标-->1
      </positionX>
      <positionY>
        <!--req, int, Y坐标-->1
      </positionY>
```

```
<!--req, int, 标题-->1
</positionY>
<height>
  <!--req, int, 宽-->1
</height>
<width>
  <!--req, int, 高-->1
</width>
</Position>
<materialNo>
  <!--opt, int, 素材ID-->1
</materialNo>
</TemperatureIcon>
<AudioAlarm>
  <!--opt, object, 音频报警信息-->
  <enabled>
    <!--req, bool, 使能-->true
  </enabled>
  <materialNo>
    <!--opt, int, 音频素材ID-->1
  </materialNo>
</AudioAlarm>
<abnormalHighTemperature>
  <!--opt, float, 异常高温值, unit: ℃, unitType: 温度, desc: 保留一位小数-->37.5
</abnormalHighTemperature>
</Temperature>
<WearMask>
  <!--opt, object, 已戴口罩提醒-->
  <enabled>
    <!--req, bool, 使能-->true
  </enabled>
  <fontSize>
    <!--req, int, 字体大小-->1
  </fontSize>
  <fontColor>
    <!--req, object, 字体颜色-->
    <RGB>
      <!--opt, int, 十进制数据表示三原色-->1
    </RGB>
  </FontColor>
  <BackColor>
    <!--req, object, 背景颜色-->
    <RGB>
      <!--opt, int, 十进制数据表示三原色-->1
    </RGB>
  </BackColor>
  <Position>
    <!--opt, object, 位置, desc: 左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--req, int, X坐标-->1
    </positionX>
    <positionY>
      <!--req, int, Y坐标-->1
    </positionY>
    <height>
      <!--req, int, 宽-->1
    </height>
    <width>
      <!--req, int, 高-->1
    </width>
  </Position>
  <text>
    <!--opt, string, 标题文字-->test
  </text>
  <MaskIcon>
    <!--opt, object, 已戴口罩图标-->
    <enabled>
      <!--req, bool, 使能-->true
    </enabled>
    <Position>
      <!--opt, object, 位置, desc: 左上角为坐标原点, 取值范围和设备分辨率保持一致-->
      <positionX>
        <!--req, int, X坐标-->1
      </positionX>
      <positionY>
        <!--req, int, Y坐标-->1
      </positionY>
      <height>
        <!--req, int, 宽-->1
      </height>
      <width>
        <!--req, int, 高-->1
      </width>
    </Position>
    <materialNo>
      <!--opt, int, 素材ID-->1
    </materialNo>
  </MaskIcon>
</WearMask>
<NormalTemperature>
  <!--opt, object, 温度正常提醒-->
  <Prompt>
    <!--opt, object, 提示信息-->
    <enabled>
```

```
<!--req, bool, 使能-->true
</enabled>
<fontSize>
<!--req, int, 字体大小-->1
</fontSize>
<FontColor>
<!--req, object, 字体颜色-->
<RGB>
<!--opt, int, 十进制数据表示三原色-->1
</RGB>
</FontColor>
<BackColor>
<!--req, object, 背景颜色-->
<RGB>
<!--opt, int, 十进制数据表示三原色-->1
</RGB>
</BackColor>
<Position>
<!--opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
<positionX>
<!--req, int, X坐标-->1
</positionX>
<positionY>
<!--req, int, Y坐标-->1
</positionY>
<height>
<!--req, int, 宽-->1
</height>
<width>
<!--req, int, 高-->1
</width>
</Position>
<text>
<!--opt, string, 标题文字-->test
</text>
</Prompt>
<TemperatureIcon>
<!--opt, object, 温度正常图标-->
<enabled>
<!--req, bool, 使能-->true
</enabled>
<Position>
<!--opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
<positionX>
<!--req, int, X坐标-->1
</positionX>
<positionY>
<!--req, int, Y坐标-->1
</positionY>
<height>
<!--req, int, 宽-->1
</height>
<width>
<!--req, int, 高-->1
</width>
</Position>
<materialNo>
<!--opt, int, 素材ID-->1
</materialNo>
</TemperatureIcon>
</NormalTemperature>
<NormalStatusAudioAlarm>
<!--opt, object, 正常状态下音频报警信息, desc:人员在检测时未发现上面的人数超限、未戴口罩、体温异常等事件时设备进行播报-->
<enabled>
<!--req, bool, 使能-->true
</enabled>
<materialNo>
<!--opt, int, 音频素材ID-->1
</materialNo>
</NormalStatusAudioAlarm>
</EventParam>
<AlarmDataSourceProperty>
<!--opt, object, 数据源信息, desc:该字段能力集缺少, 仅信发组件和设备使用-->
<PictureProperty>
<!--opt, object, 图片属性-->
<PictureSwitchEffect>
<!--opt, enum, 图片切换效果, subType:string, [LeftInRightOut#左进右出,rightInLeftOut#右进左出]-->leftInRightOut
</PictureSwitchEffect>
<PictureScaleType>
<!--opt, enum, 图片缩放类型, subType:string, [fitCenter#拉伸填充,centerCrop#截取填充,fixXY#适应宽高]-->fitCenter
</PictureScaleType>
</PictureProperty>
<AlarmDataSource>
<!--opt, object, 数据源-->
<MaterialNo>
<!--opt, int, 素材编号-->1
</MaterialNo>
<Name>
<!--opt, string, 素材名称-->test
</Name>
<Address>
<!--opt, string, 素材URL地址-->test
</Address>
```

```
<webProperty>
  <!--opt, object, 网页属性-->
  <webType>
    <!--opt, enum, 网页类型, subType:string, [desktop#桌面版,mobile#移动版], desc:默认按照设备的系统类型（Windows或Android）来显示网页-->desktop
  </webType>
</webProperty>
</AlarmDataSource>
</AlarmDataSourceProperty>
<ConferenceParam>
  <!--opt, object, 会议信息, desc:当页面用于会议界面时有效-->
  <backPicId>
    <!--opt, int, 背景图片素材ID-->1
  </backPicId>
  <Title>
    <!--opt, object, 会议标题（主题）, desc:1、会议标题（主题）的文字内容通过场地预约配置协议下发,对应/ISAPI/Publish/Signage/placeReservation?format=json中的programID字段。2、可通过会议计划（/ISAPI/InteractivePanel/Meeting/plan?format=json）中的meetingTopic字段下发-->
  <enabled>
    <!--req, bool, 使能-->true
  </enabled>
  <fontSize>
    <!--req, int, 字体大小-->1
  </fontSize>
  <fontColor>
    <!--req, object, 字体颜色-->
    <RGB>
      <!--opt, int, 十进制数据表示三原色-->1
    </RGB>
  </fontColor>
  <BackColor>
    <!--req, object, 背景颜色-->
    <RGB>
      <!--opt, int, 十进制数据表示三原色-->1
    </RGB>
  </BackColor>
  <Position>
    <!--opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--req, int, X坐标-->1
    </positionX>
    <positionY>
      <!--req, int, Y坐标-->1
    </positionY>
    <height>
      <!--req, int, 宽-->1
    </height>
    <width>
      <!--req, int, 高-->1
    </width>
    </Position>
  <alignType>
    <!--req, enum, 对齐方式, subType:string, [Left#左,right#右,middle#中]-->middle
  </alignType>
</Title>
<VoiceBroadcast>
  <!--opt, object, 语音播报-->
  <enabled>
    <!--opt, bool, 是否启用语音播报-->true
  </enabled>
  <broadcastSpeed>
    <!--opt, enum, 播报语速, subType:string, [slower#慢,slow#较慢,medium#中速,fast#较快,faster#快]-->slower
  </broadcastSpeed>
  <inLibVoice>
    <!--opt, string, 在库人员语音播报内容-->test
  </inLibVoice>
  <outLibVoice>
    <!--opt, string, 未知人员语音播报内容-->test
  </outLibVoice>
</VoiceBroadcast>
<WelcomeWord>
  <!--opt, object, 欢迎词-->
  <enabled>
    <!--opt, bool, 是否启用欢迎词-->true
  </enabled>
  <inLibWord>
    <!--opt, string, 在库人员欢迎词内容, range:[1,128]-->test
  </inLibWord>
  <outLibWord>
    <!--opt, string, 未知人员欢迎词内容, range:[1,128]-->test
  </outLibWord>
</WelcomeWord>
<ShowInfo>
  <!--opt, object, 展示信息, desc:下发该字段需要人脸签到, 不下发该字段, 不需要人脸签到-->
  <row>
    <!--opt, int, 窗口内人员签到展示行数-->1
  </row>
  <column>
    <!--opt, int, 窗口内人员签到展示列数-->1
  </column>
  <Position>
    <!--opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--req, int, x坐标-->1
    </positionX>
    <positionY>
```

```

        <!--req, int, y坐标-->1
    </positionY>
    <height>
        <!--req, int, 宽-->1
    </height>
    <width>
        <!--req, int, 高-->1
    </width>
</Position>
<inLib>
    <!--opt, object, 在库信息-->
    <nameEnabled>
        <!--opt, bool, 在库人员是否展示姓名-->true
    </nameEnabled>
</inLib>
<defaultShowInfoEnabled>
    <!--opt, bool, 是否采用人员默认展示信息, desc: 下发该字段, 且为true, 表示人员展示信息采用默认值。坐标信息等不用下发-->true
</defaultShowInfoEnabled>
</ShowInfo>
<ModuleInfoList>
    <!--opt, array, 界面模块信息列表, subType:object-->
    <ModuleInfo>
        <!--opt, object, 界面模块信息-->
        <moduleType>
            <!--req, enum, 模块类型, subType:string, [conferenceTime#会议时间(对应timeRange), conferenceSite#会议地点(对应会议室名称meetingName), conferenceStatus#会议状态(对应meetingStatus), signInPersonNum#已签到人数, totalPersonNum#应到人数, unSignInPersonNum#未签到人数, signInAndTotalPersonNum#已签到/应到人数, QRCode#会议室二维码, meetingHostName#主持人, employeeNo#参会人, customKey#自定义字段, nextMeeting#当天下一场会议信息]-->conferenceTime
        </moduleType>
        <enabled>
            <!--req, bool, 使能-->true
        </enabled>
        <fontSize>
            <!--req, int, 字体大小-->1
        </fontSize>
        <fontColor>
            <!--req, object, 字体颜色-->
            <RGB>
                <!--opt, int, 十进制数据表示三原色-->1
            </RGB>
        </fontColor>
        <BackColor>
            <!--opt, object, 背景颜色-->
            <RGB>
                <!--opt, int, 十进制数据表示三原色-->1
            </RGB>
        </BackColor>
        <backPicId>
            <!--opt, int, 背景图片素材ID, desc: 背景图和背景颜色同时仅支持一种, 两者均下发, 设备优先解析背景图字段-->1
        </backPicId>
        <Position>
            <!--opt, object, 位置, desc: 左上角为坐标原点, 取值范围和设备分辨率保持一致-->
            <positionX>
                <!--req, int, X坐标-->1
            </positionX>
            <positionY>
                <!--req, int, Y坐标-->1
            </positionY>
            <height>
                <!--req, int, 宽-->1
            </height>
            <width>
                <!--req, int, 高-->1
            </width>
        </Position>
        <alignType>
            <!--req, enum, 对齐方式, subType:string, [Left#左, right#右, middle#中]-->middle
        </alignType>
        <CustomKey>
            <!--opt, object, 自定义字段-->
            <key>
                <!--opt, string, 自定义字段-->test
            </key>
            <associateProtocolURL>
                <!--opt, string, 关联协议URL-->test
            </associateProtocolURL>
        </CustomKey>
    </ModuleInfo>
</ModuleInfoList>
<conferenceStatus>
    <!--opt, enum, 会议状态, subType:string, [idle#空闲, signing#签到中, meeting#开会中], desc: 根据不同的会议状态, 配置不同的会议界面-->idle
</conferenceStatus>
<BackColor>
    <!--opt, object, 背景颜色-->
    <RGB>
        <!--opt, int, 十进制数据表示三原色-->1
    </RGB>
</BackColor>
</ConferenceParam>
<CallBtn>
    <!--opt, object, 呼叫按钮窗口-->
    <backPicId>
        <!--opt, int, 背景图片素材ID-->1
    </backPicId>

```

```

</backPicId>
</CallBtn>
<OpenDoorBtn>
  <!--opt, object, 开门按钮窗口-->
  <backPicId>
    <!--opt, int, 背景图片素材ID-->1
  </backPicId>
</OpenDoorBtn>
<QRCodeBtn>
  <!--opt, object, 二维码按钮窗口-->
  <backPicId>
    <!--opt, int, 背景图片素材ID-->1
  </backPicId>
</QRCodeBtn>
<AuthenticationSuccess>
  <!--opt, object, 认证成功窗口-->
  <BackColor>
    <!--opt, object, 背景颜色-->
    <RGB>
      <!--opt, int, 十进制数据表示三原色-->1
    </RGB>
  </BackColor>
  <backPicId>
    <!--opt, int, 背景图片素材ID, desc: 背景图和背景颜色同时仅支持一种, 两者均下发, 设备优先解析背景图字段-->1
  </backPicId>
  <SubWindowList>
    <!--opt, object, 子窗口列表-->
    <SubWindow>
      <!--opt, object, 子窗口-->
      <subWindowType>
        <!--req, enum, 子窗口类型, subType:string, [temperature#温度, facePic#人脸图片, name#姓名, employeeNo#工号, group#组织/部门, tips#提示信息]-->
        </subWindowType>
      <enabled>
        <!--req, bool, 使能-->true
      </enabled>
      <fontSize>
        <!--opt, int, 字体大小, unit:px, unitType:图像显示-->1
      </fontSize>
      <fontColor>
        <!--opt, int, 字体颜色-->1
      </fontColor>
      <fontStyle>
        <!--opt, string, 字体风格, desc:bold-加粗, italic-倾斜, underline-下划线, strikethrough-删除线-->bold,italic,underline,strikeout
      </fontStyle>
      <Position>
        <!--opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
        <positionX>
          <!--req, int, x坐标-->1
        </positionX>
        <positionY>
          <!--req, int, y坐标-->1
        </positionY>
      <height>
        <!--req, int, 宽-->1
      </height>
      <width>
        <!--req, int, 高-->1
      </width>
    </Position>
  </SubWindow>
</SubWindowList>
</AuthenticationSuccess>
<AuthenticationFail>
  <!--opt, object, 认证失败窗口-->
  <BackColor>
    <!--opt, object, 背景颜色-->
    <RGB>
      <!--opt, int, 十进制数据表示三原色-->1
    </RGB>
  </BackColor>
  <backPicId>
    <!--opt, int, 背景图片素材ID, desc: 背景图和背景颜色同时仅支持一种, 两者均下发, 设备优先解析背景图字段-->1
  </backPicId>
  <SubWindowList>
    <!--opt, object, 子窗口列表-->
    <SubWindow>
      <!--opt, object, 子窗口-->
      <subWindowType>
        <!--req, enum, 子窗口类型, subType:string, [temperature#温度, facePic#人脸图片, name#姓名, employeeNo#工号, group#组织/部门, tips#提示信息]-->
        </subWindowType>
      <enabled>
        <!--req, bool, 使能-->true
      </enabled>
      <fontSize>
        <!--opt, int, 字体大小, unit:px, unitType:图像显示-->1
      </fontSize>
      <fontColor>
        <!--opt, int, 字体颜色-->1
      </fontColor>
      <fontStyle>
        <!--opt, string, 字体风格, desc:bold-加粗, italic-倾斜, underline-下划线, strikethrough-删除线-->bold,italic,underline,strikeout
      </fontStyle>
    </SubWindow>
  </SubWindowList>
</AuthenticationFail>

```

```

</Position>
<!--opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
<positionX>
  <!--req, int, x坐标-->1
</positionX>
<positionY>
  <!--req, int, y坐标-->1
</positionY>
<height>
  <!--req, int, 宽-->1
</height>
<width>
  <!--req, int, 高-->1
</width>
</Position>
<facePicID>
  <!--ro, opt, int, 默认人脸图片ID, step:1, dep:and,
{$.PageList.Page.WindowsList[*].Windows.AuthenticationFail.SubWindowList.SubWindow.subWindowType,eq,facePic}-->1
</facePicID>
</SubWindow>
</SubWindowList>
</AuthenticationFail>
<StatusBar>
  <!--opt, object, 状态栏, desc:当素材类型为状态栏时有效, 且必填-->
  <backPicId>
    <!--opt, int, 背景图片素材ID, desc:背景图和背景颜色同时仅支持一种, 两者均下发, 设备优先解析背景图字段-->1
  </backPicId>
  <statusBarType>
    <!--req, enum, 状态栏类型, subType:string, [fixed#固定的], desc:当为固定类型时, 窗口位置不生效, 设备将状态栏固定在顶部显示-->fixed
  </statusBarType>
  <SubStatusList>
    <!--opt, object, 子状态列表-->
    <SubStatus>
      <!--opt, object, 子状态-->
      <statusType>
        <!--req, enum, 子状态, subType:string, [dial#拨号状态(3G4G等),wifi#wifi状态,EZVIZ#萤石状态,wiredNetwork#有线网络状态,deploy#布防状态,temperature#测温]-->dial
      </statusType>
      <enabled>
        <!--req, bool, 是否显示-->true
      </enabled>
    </SubStatus>
  </SubStatusList>
</StatusBar>
<ExamTemplate>
  <!--opt, object, 考试模板, desc:当页面用于会议界面时有效-->
  <examTemplateID>
    <!--opt, string, 模板ID-->test
  </examTemplateID>
  <backPicURL>
    <!--opt, string, 背景图片素材URL, range:[1,1024]-->test
  </backPicURL>
  <backgroundDirection>
    <!--opt, enum, 背景图方向, subType:string, [horizontal#水平,vertical#垂直]-->horizontal
  </backgroundDirection>
  <ModuleInfoList>
    <!--opt, array, 界面模块信息列表, subType:object-->
    <ModuleInfo>
      <!--opt, object, 界面模块信息-->
      <moduleType>
        <!--opt, enum, 模块类型, subType:string, [schoolName#学校名称,currentTime#当前时间,examinationPlanLabel#考试计划标签,examinationPlanName#考试计划名称,examRoomNumberLabel#考场号标签,examRoomNumber#考场号,examRoomAddress#考试地点,examRoomAddressLabel,examDescriptionLabel#考试说明标签,examDescription#考试说明,examSubjectLabel#考试科目标签,examSubject#考试科目,examTimeLabel#考试时间标签,examTime#考试时间,teacherLabel#监考老师标签,teacher#监考老师,teacherLabel#监考老师标签,teacher#监考老师,examNumberRangeLabel#考号范围,examNumberRange#考号范围]-->schoolName
      </moduleType>
      <enabled>
        <!--opt, bool, 使能-->true
      </enabled>
      <Position>
        <!--opt, object, 位置, desc:左上角为坐标原点-->
        <positionX>
          <!--req, int, X坐标-->1
        </positionX>
        <positionY>
          <!--req, int, Y坐标-->1
        </positionY>
        <height>
          <!--req, int, 高-->1
        </height>
        <width>
          <!--req, int, 宽-->1
        </width>
      </Position>
      <fontSize>
        <!--opt, int, 字体大小-->1
      </fontSize>
      <fontStyleList>
        <!--opt, array, 字体风格列表, subType:object-->
        <fontStyle>
          <!--opt, enum, 字体风格, subType:string, [bold#加粗,italic#倾斜,regular#常规,striethrough#中划线,underline#下划线]-->bold
        </fontStyle>
      </fontStyleList>
    </ModuleInfo>
  </ModuleInfoList>
  <fontType>

```

```

        <!--opt, enum, 字体类型, subType:string, [custom#自定义,default#默认,kaiTi#楷体,simHei#黑体,simSun#宋体]-->custom
    </FontType>
    <FontColor>
        <!--opt, object, 字体颜色-->
        <RGB>
            <!--opt, int, 十进制数据表示三原色-->1
        </RGB>
    </FontColor>
    <fontTransparent>
        <!--opt, int, 字体透明度-->1
    </fontTransparent>
    <alignType>
        <!--opt, enum, 字体对齐方式, subType:string, [left#左,right#右,middle#中]-->middle
    </alignType>
    <characterSpacing>
        <!--opt, int, 字符间距-->1
    </characterSpacing>
    <lineSpacing>
        <!--opt, int, 行间距-->1
    </lineSpacing>
    <layerNo>
        <!--opt, int, 图层编号, desc:数字越大, 表示在越上层-->1
    </layerNo>
    </ModuleInfo>
    </ModuleInfoList>
    <ExamTemplate>
    <ChartData>
        <!--opt, object, 图表数据-->
        <materialNo>
            <!--req, int, 素材编号-->1
        </materialNo>
        <chartType>
            <!--opt, enum, 图表类型, subType:string, [histogram#柱状图,pieChart#饼图]-->histogram
        </chartType>
        <backPicId>
            <!--opt, int, 背景图片素材编号-->1
        </backPicId>
        <HeadDataList>
            <!--opt, array, 表头数据列表, subType:object-->
            <HeadData>
                <!--opt, object, 表头数据-->
                <id>
                    <!--req, int, 索引-->1
                </id>
                <data>
                    <!--req, string, 数据-->test
                </data>
                <headType>
                    <!--opt, enum, 表头类型, subType:string, [axis#轴,value#值]-->axis
                </headType>
            </HeadData>
        </HeadDataList>
    </ChartData>
    </Windows>
    </WindowsList>
</Page>

```

响应报文

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
    <!--ro, req, object, 响应消息, attr:version{ro, req, string, 版本}-->
    <requestURL>
        <!--ro, req, string, 请求的URL-->null
    </requestURL>
    <statusCode>
        <!--ro, req, enum, 状态码, subType:int, [0#OK,1#OK,2#Device Busy,3#Device Error,4#Invalid Operation,5#Invalid XML Format,6#Invalid XML Content,7#Reboot Required]-->0
    </statusCode>
    <statusString>
        <!--ro, req, enum, 状态信息, subType:string, [OK#成功,Device Busy#设备忙,Device Error#设备异常,Invalid Operation#无效的操作,Invalid XML Format#无效的XML格式,Invalid XML Content#无效的XML内容,Reboot#设备重启]-->OK
    </statusString>
    <subStatusCode>
        <!--ro, req, string, 详细错误码英文描述, desc:详细错误码的英文描述-->OK
    </subStatusCode>
</ResponseStatus>

```

69.63.2 获取单个页面参数

Request URL

GET /ISAPI/Publish/ProgramMgr/program/<programID>/page/<pageID>

查询参数

参数名称	参数类型	描述
programID	string	节目ID
pageID	string	页面ID

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<Page xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 页面, attr:version{req, string, 协议版本}-->
  <id>
    <!--ro, req, int, 索引-->1
  </id>
  <PageBasicInfo>
    <!--ro, req, object, 页面基本信息-->
    <pageName>
      <!--ro, opt, string, 页面名称-->test
    </pageName>
    <BackgroundColor>
      <!--ro, opt, object, 背景颜色-->
      <RGB>
        <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
      </RGB>
    </BackgroundColor>
    <playDurationMode>
      <!--ro, opt, enum, 页面播放时间模式, subType:string, [selfDefine#自定义,auto#自动,loop#循环], desc:自定义时playDuration节点有效;自动时根据播放内容的
      时间*PlayCount计算;循环时表示一直循环-->selfDefine
    </playDurationMode>
    <playDuration>
      <!--ro, opt, int, 页面播放时长, unit:s, unitType:时间-->1
    </playDuration>
    <playCount>
      <!--ro, opt, int, 页面播放次数-->1
    </playCount>
    <switchDuration>
      <!--ro, opt, int, 切换间隔, unit:s, unitType:时间-->1
    </switchDuration>
    <switchEffect>
      <!--ro, opt, enum, 切换效果, subType:string, [none#无,random#随机,boxShrink#盒状收缩,boxSpread#盒状展开,cycleShrink#圆形收缩,cycSpread#圆形展
      开,eraseUp#向上擦除,eraseDown#向下擦除,eraseLeft#向左擦除,eraseRight#向右擦除,verticalShelter#垂直屏蔽,horizontalShelter#水平屏蔽,verticalChessboard#纵向棋
      盘式,horizontalChessboard#横向棋盘式,dissolve#随机溶解,LeftRightToCenter#左右向中间缩进,ceterToLeftRight#中央向左右扩展,upDownToCenter#上下向中央缩
      进,centerToUpDown#中央向上下扩展,drawOutLeftDown#从左下抽出,drawOutLeftUp#从左上抽出,drawOutRightDown#从右下抽出,drawOutRightUp#从右上抽出,verticalLine#随机
      垂直线条,horizontalLine#随机垂直线条]-->none
    </switchEffect>
    <backgroundPic>
      <!--ro, opt, int, 背景图片素材ID-->1
    </backgroundPic>
    <SignInCfg>
      <!--ro, opt, object, 签到信息, desc:当页面用于签到或考勤界面时有效-->
      <VoiceBroadcast>
        <!--ro, opt, object, 语音播报-->
        <enabled>
          <!--ro, opt, bool, 是否启用语音播报-->true
        </enabled>
        <broadcastSpeed>
          <!--ro, opt, enum, 播报语速, subType:string, [sLower#慢,sLow#较慢,medium#中速,fast#较快,faster#快]-->slower
        </broadcastSpeed>
        <inLibVoice>
          <!--ro, opt, string, 在库人员正常体温语音播报内容-->test
        </inLibVoice>
        <outLibVoice>
          <!--ro, opt, string, 未知人员正常体温语音播报内容-->test
        </outLibVoice>
        <inLibAbnormalTemperatureVoice>
          <!--ro, opt, string, 在库人员异常温度语音播报内容-->test
        </inLibAbnormalTemperatureVoice>
        <outLibAbnormalTemperatureVoice>
          <!--ro, opt, string, 未知人员异常温度语音播报内容-->test
        </outLibAbnormalTemperatureVoice>
      </VoiceBroadcast>
      <WelcomeWord>
        <!--ro, opt, object, 欢迎词-->
        <enabled>
          <!--ro, opt, bool, 是否启用欢迎词-->true
        </enabled>
        <inLibWord>
          <!--ro, opt, string, 在库人员欢迎词内容-->test
        </inLibWord>
        <outLibWord>
          <!--ro, opt, string, 未知人员欢迎词内容-->test
        </outLibWord>
      </WelcomeWord>
    </SignCartoon>
  </PageBasicInfo>
  <SignCartoon>
  </SignCartoon>
</Page>
```

```

<!--ro, opt, object, 签到动画-->
<signCartoonSpeed>
  <!--ro, opt, enum, 动画速度, subType:string, [fast#快,medium#终端,slow#慢]-->fast
</signCartoonSpeed>
</SignCartoon>
<backgroundSignEnabled>
  <!--ro, opt, bool, 是否启用后台签到-->true
</backgroundSignEnabled>
<TemperatureDetection>
  <!--ro, opt, object, 温度监测参数-->
  <enabled>
    <!--ro, req, bool, 是否启用-->true
  </enabled>
  <threshold>
    <!--ro, req, float, 正常温度值-->0.000
  </threshold>
</TemperatureDetection>
</SignInCfg>
</PageBasicInfo>
<characterMode>
  <!--ro, opt, enum, 页面中的欢迎词模式, subType:string, [mode1#模式1,mode2#模式2,mode3#模式3], desc:门禁增加, 门禁欢迎词节目位置固定, 且存在三种模式-->mode1
</characterMode>
<windowsList>
  <!--ro, opt, array, 窗口信息列表, subType:object-->
  <windows>
    <!--ro, opt, object, 窗口信息-->
    <id>
      <!--ro, req, int, 内容编号-->1
    </id>
    <position>
      <!--ro, opt, object, 内容位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
      <positionX>
        <!--ro, req, int, x坐标-->1
      </positionX>
      <positionY>
        <!--ro, req, int, y坐标-->1
      </positionY>
      <height>
        <!--ro, req, int, 宽-->1
      </height>
      <width>
        <!--ro, req, int, 高-->1
      </width>
    </position>
    <CornerRadius>
      <!--ro, opt, object, 圆角配置, desc:目前仅对picture图片窗口和character文字窗口有效-->
      <TopLeft>
        <!--ro, opt, int, 左上角半径-->1
      </TopLeft>
      <TopRight>
        <!--ro, opt, int, 右上角半径-->1
      </TopRight>
      <BottomLeft>
        <!--ro, opt, int, 左下角半径-->1
      </BottomLeft>
      <BottomRight>
        <!--ro, opt, int, 右下角半径-->1
      </BottomRight>
    </CornerRadius>
    <layerNo>
      <!--ro, req, int, 图层编号-->1
    </layerNo>
    <winMaterialInfo>
      <!--ro, opt, object, 窗口素材信息-->
      <materialType>
        <!--ro, req, enum, 素材类型, subType:string, [static#静态,dynamic#动态,other#其他]-->static
      </materialType>
      <staticMaterialType>
        <!--ro, opt, enum, 静态素材类型, subType:string, [picture#图片,flash#flash,audio#音频,video#视频,document#文本,ppt#ppt,doc#文档,excel#表格,pdf#pdf,web#网页,app#app,signIn#签到,conference#会议]-->picture
      </staticMaterialType>
      <dynamicType>
        <!--ro, opt, enum, 动态素材类型, subType:string, [web#web,socket#socket,rss#rss,realStream#realStream,call#call,dynamicPic#dynamicPic,capturePic#capturePic,character#character,chart#图表数据]-->web
      </dynamicType>
      <otherType>
        <!--ro, opt, enum, 其他素材类型, subType:string, [clock#时钟,weather#天气,countdown#倒计时,localInput#本地输入,hyperLinkBtn#超链接按钮,event#事件,callBtn#呼叫按钮,openDoorBtn#开门按钮,QRCODEBtn#二维码按钮,authenticationSuccess#认证成功,authenticationFail#认证失败,statusBar#状态栏,ExamTemplate#考试模板]-->clock
      </otherType>
    </winMaterialInfo>
    <TouchProperty>
      <!--ro, opt, object, 触摸属性-->
      <windType>
        <!--ro, opt, enum, 窗口类型, subType:string, [popup#弹窗,page#页面]-->popup
      </windType>
      <hyperlinkType>
        <!--ro, opt, enum, 超链接类型, subType:string, [window#页面超链接,page#窗口超链接], desc:窗口类型为弹窗时有效-->window
      </hyperlinkType>
      <windowId>
        <!--ro, opt, int, 窗口编号, desc:当前页面的窗口,超链接类型为窗口时有效-->1
      </windowId>
    </TouchProperty>
  </windows>
</windowsList>

```

```
<pageId>
  <!--ro, opt, int, 页面编号, desc:超链接类型为页面时有效-->1
</pageId>
</TouchProperty>
<PlayItemList>
  <!--ro, opt, array, 窗口播放列表, subType:object-->
  <PlayItem>
    <!--ro, req, object, 窗口播放信息-->
    <id>
      <!--ro, req, int, 播放序号-->1
    </id>
    <materialNo>
      <!--ro, opt, int, 素材编号-->1
    </materialNo>
    <inputChannel>
      <!--ro, opt, int, 绑定的IPC通道号-->1
    </inputChannel>
    <playEffect>
      <!--ro, req, enum, 播放效果, subType:string, [none#无,marquee#跑马灯]-->none
    </playEffect>
    <MarqueeInfo>
      <!--ro, opt, object, 跑马灯信息-->
      <scrollType>
        <!--ro, req, enum, 跑马灯滚动类型, subType:string, [none#不滚动,Loops#循环滚动,once#滚动一次,backAndForth#来回滚动]-->none
      </scrollType>
      <scrollDeriction>
        <!--ro, req, enum, 跑马灯滚动方向, subType:string, [none#无,up#向上,down#向下,left#向左,right#向右]-->none
      </scrollDeriction>
      <scrollSpeed>
        <!--ro, req, int, 跑马灯滚动速度-->1
      </scrollSpeed>
    </MarqueeInfo>
    <PlayDuration>
      <!--ro, opt, object, 窗口内素材播放时间, desc:静态素材可以配置该时间,视频直播和IPC通道都可配置该节点-->
      <durationType>
        <!--ro, req, enum, 持续方式, subType:string, [materialTime#按素材时间,selfDefine#自定义,unlimited#无限时长]-->selfDefine
      </durationType>
      <duration>
        <!--ro, req, int, 持续时间, unit:s, unitType:时间-->1
      </duration>
    </PlayDuration>
    <CharactersEffect>
      <!--ro, opt, object, 文字显示效果, desc:当素材类型为文本txt和时有效-->
      <fontSize>
        <!--ro, req, int, 字体大小-->1
      </fontSize>
      <fontColor>
        <!--ro, req, object, 字体颜色-->
        <RGB>
          <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
        </RGB>
      </fontColor>
      <backgroundType>
        <!--ro, opt, enum, 背景类型, subType:string, [color#颜色（默认）,picture#图片], desc:字段不存在时,默认为颜色-->color
      </backgroundType>
      <backPicId>
        <!--ro, opt, int, 背景图片素材ID-->1
      </backPicId>
      <BackColor>
        <!--ro, req, object, 背景颜色-->
        <RGB>
          <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
        </RGB>
      </BackColor>
      <backTransparent>
        <!--ro, req, int, 背景透明度-->1
      </backTransparent>
      <subtitlesEnabled>
        <!--ro, req, bool, 字幕使能-->true
      </subtitlesEnabled>
      <scrollDirection>
        <!--ro, req, enum, 文字滚动方向, subType:string, [Left#向左,right#向右,up#向上,down#向下]-->left
      </scrollDirection>
      <scrollSpeed>
        <!--ro, req, int, 文字滚动速度-->1
      </scrollSpeed>
    </CharactersEffect>
    <switchEffect>
      <!--ro, opt, enum, 窗口素材切换效果, subType:string, [none#无,LeftInRightOut#左进右出,rightInLeftOut#右进左出,bottomInTopOut#下进上出,topInBottomOut#上进下出,fadeInFadeOut#淡入淡出,middleExit#中间退出,topPop#顶部弹出,rightBottomIn#右下角进入,leftTopIn#左上角进入,horizontalOpen#水平打开,verticalOpen#垂直打开,random#随机效果], desc:图片素材时有效-->random
    </switchEffect>
    <pageTime>
      <!--ro, opt, int, 翻页时间, unit:s, unitType:时间, desc:当素材为word,ppt,pdf,excel时有效-->1
    </pageTime>
    <scrollSpeed>
      <!--ro, req, int, 滚动速度, desc:当素材为静态web时有效-->1
    </scrollSpeed>
    <scaleType>
      <!--ro, opt, enum, 素材缩放参数配置, subType:string, [fitXY#拉伸填充,fitCenter#自适应,centerCrop#截取填充], desc:目前仅当图片窗口有效-->centerCrop
    </scaleType>
    <playType>
      <!--ro, opt, enum, 播放方式, subType:string, [normal#正常,faceAttendance#考勤,PeopleCounting#客流,temperature#测温], desc:目前仅当图片窗口有效-->normal
    </playType>
```

>normal

```
</playType>
<CharactersAttribute>
  <!--ro, opt, object, 文字属性, desc:当dynamicType为character时有效-->
  <fontSize>
    <!--ro, req, int, 字体大小-->1
  </fontSize>
  <FontColor>
    <!--ro, req, object, 字体颜色-->
    <RGB>
      <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
    </RGB>
  </FontColor>
  <backgroundType>
    <!--ro, req, enum, 背景类型, subType:string, [color#颜色,picture#图片]-->color
  </backgroundType>
  <BackColor>
    <!--ro, req, object, 背景色-->
    <RGB>
      <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
    </RGB>
  </BackColor>
  <backTransparent>
    <!--ro, opt, int, 背景透明度-->1
  </backTransparent>
  <alignType>
    <!--ro, req, enum, 对齐方式, subType:string, [Left#左,right#右,middle#中]-->middle
  </alignType>
  <verticalAlignType>
    <!--ro, req, enum, 垂直对齐方式, subType:string, [top#顶,bottom#底,verticalCenter#中]-->top
  </verticalAlignType>
  <characterContent>
    <!--ro, opt, string, 文字内容, range:[0,512]-->test
  </characterContent>
  <backPicId>
    <!--ro, opt, int, 背景图片素材ID-->1
  </backPicId>
  <characterStyle>
    <!--ro, req, enum, 字体效果, subType:string, [normal#常规,bold#粗体]-->normal
  </characterStyle>
  <characterType>
    <!--ro, req, enum, 字体类型, subType:string, [mainTitle#主标题,subTitle#副标题]-->mainTitle
  </characterType>
</CharactersAttribute>
<WebProperty>
  <!--ro, opt, object, 网页属性-->
  <webType>
    <!--ro, opt, enum, 网页类型, subType:string, [desktop#桌面版,mobile#移动版], desc:默认按照设备的系统类型（windows或Android）来显示网页-->1
  </webType>
</WebProperty>
</PlayItem>
</PlayItemList>
<enabledAudio>
  <!--ro, opt, bool, 音频使能-->true
</enabledAudio>
<enableHide>
  <!--ro, opt, bool, 隐藏使能-->true
</enableHide>
<enableLock>
  <!--ro, opt, bool, 锁定使能-->true
</enableLock>
<AppWindow>
  <!--ro, opt, object, App窗口信息-->
  <WindowInfoList>
    <!--ro, req, array, 窗口信息列表, subType:object-->
    <WindowInfo>
      <!--ro, req, object, 窗口信息-->
      <id>
        <!--ro, req, int, 索引-->1
      </id>
      <materialNo>
        <!--ro, req, int, 素材编号, desc:采用本字段下发时, 用户需要先将App安装包当做素材存储到设备中, 得到素材编号再进行下发-->1
      </materialNo>
      <path>
        <!--ro, opt, string, App路径, range:[0,1024],
desc:path和materialNo字段同时存在时, 设备优先解析path字段;
采用本字段下发时, 设备需要确保App已经安装在设备内, 直接传入App的路径即可;
对于Android设备, path传的是App的包名; 对于Windows设备, path传的是App的应用程序绝对路径; -->test
      </path>
    </WindowInfo>
  </WindowInfoList>
</AppWindow>
<DataSource>
  <!--ro, opt, object, 数据来源, desc:当窗口为叫号和动态弹图时有效-->
  <materialNo>
    <!--ro, req, int, 素材编号-->1
  </materialNo>
  <dataSourceUUID>
    <!--ro, opt, string, 数据来源UUID-->1
  </dataSourceUUID>
</DataSource>
<Call>
```

>desktop

```
</webType>
</WebProperty>
</PlayItem>
</PlayItemList>
<enabledAudio>
  <!--ro, opt, bool, 音频使能-->true
</enabledAudio>
<enableHide>
  <!--ro, opt, bool, 隐藏使能-->true
</enableHide>
<enableLock>
  <!--ro, opt, bool, 锁定使能-->true
</enableLock>
<AppWindow>
  <!--ro, opt, object, App窗口信息-->
  <WindowInfoList>
    <!--ro, req, array, 窗口信息列表, subType:object-->
    <WindowInfo>
      <!--ro, req, object, 窗口信息-->
      <id>
        <!--ro, req, int, 索引-->1
      </id>
      <materialNo>
        <!--ro, req, int, 素材编号, desc:采用本字段下发时, 用户需要先将App安装包当做素材存储到设备中, 得到素材编号再进行下发-->1
      </materialNo>
      <path>
        <!--ro, opt, string, App路径, range:[0,1024],
desc:path和materialNo字段同时存在时, 设备优先解析path字段;
采用本字段下发时, 设备需要确保App已经安装在设备内, 直接传入App的路径即可;
对于Android设备, path传的是App的包名; 对于Windows设备, path传的是App的应用程序绝对路径; -->test
      </path>
    </WindowInfo>
  </WindowInfoList>
</AppWindow>
<DataSource>
  <!--ro, opt, object, 数据来源, desc:当窗口为叫号和动态弹图时有效-->
  <materialNo>
    <!--ro, req, int, 素材编号-->1
  </materialNo>
  <dataSourceUUID>
    <!--ro, opt, string, 数据来源UUID-->1
  </dataSourceUUID>
</DataSource>
<Call>
```

```

<!--ro, opt, object, 叫号数据, desc:对应表格数据-->
<materialNo>
  <!--ro, req, int, 素材编号-->1
</materialNo>
<tableRow>
  <!--ro, req, int, 表格行-->1
</tableRow>
<tableColumn>
  <!--ro, req, int, 表格列-->1
</tableColumn>
<tableDirection>
  <!--ro, req, enum, 表格方向, subType:string, [vertical#纵向,horizontal#纵向]-->vertical
</tableDirection>
<tableType>
  <!--ro, req, enum, 表格类型, subType:string, [template1#模板1,template2#模板2,template3#模板3,template4#模板4,template5#模板5,template6#模板6]-->template1
</tableType>
<backPicId>
  <!--ro, opt, int, 背景图片素材编号-->1
</backPicId>
<alignType>
  <!--ro, req, enum, 对齐方式, subType:string, [left#左,right#右,middle#中]-->left
</alignType>
<refreshDirection>
  <!--ro, req, enum, 刷新方向, subType:string, [upTdown#从上到下,downToup#从下到上,LeftToright#从左到右,rightToLeft#从右到左]-->upTdown
</refreshDirection>
<HeadDataList>
  <!--ro, opt, array, 表头数据列表, subType:object-->
  <HeadData>
    <!--ro, opt, object, 表头数据-->
    <id>
      <!--ro, req, int, 索引-->1
    </id>
    <data>
      <!--ro, req, string, 数据-->test
    </data>
  </HeadData>
</HeadDataList>
<ItemStyleList>
  <!--ro, opt, array, 表格列或行样式列表, subType:object-->
  <ItemStyle>
    <!--ro, opt, object, 表格列或行样式-->
    <id>
      <!--ro, req, int, 索引-->1
    </id>
    <width>
      <!--ro, req, int, 每列宽度所占百分比-->1
    </width>
    <fontSize>
      <!--ro, req, int, 字体大小-->1
    </fontSize>
    <fontColor>
      <!--ro, req, object, 字体颜色-->
      <RGB>
        <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
      </RGB>
    </fontColor>
    <BackColor>
      <!--ro, req, object, 背景色-->
      <RGB>
        <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
      </RGB>
    </BackColor>
  </ItemStyle>
</ItemStyleList>
<playType>
  <!--ro, opt, enum, 数据播放方式, subType:string, [roll#滚动播放,flipOver#翻页播放]-->roll
</playType>
</Call>
<DynamicPic>
  <!--ro, opt, object, 动态弹图窗口配置-->
  <materialNo>
    <!--ro, req, int, 素材编号-->1
  </materialNo>
  <backPicId>
    <!--ro, opt, int, 背景图片素材ID-->1
  </backPicId>
</DynamicPic>
<CapturePic>
  <!--ro, opt, object, 抓拍图片参数-->
  <backPicId>
    <!--ro, opt, int, 背景图片素材ID-->1
  </backPicId>
  <ipcMaterialNo>
    <!--ro, req, int, IPC素材编号-->1
  </ipcMaterialNo>
  <cancelType>
    <!--ro, req, enum, 取消方式, subType:string, [auto#自动>manual#手动]-->>manual
  </cancelType>
  <duration>
    <!--ro, opt, int, 持续时间, unit:s, unitType:时间-->1
  </duration>
</CapturePic>
</ClockParam>

```

```
<!--ro, opt, object, 时钟参数-->
<backPicId>
  <!--ro, opt, int, 背景图片素材ID-->1
</backPicId>
<ClockIcon>
  <!--ro, req, object, 时钟图标参数-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <type>
    <!--ro, opt, enum, 时钟类型, subType:string, [cLock1#cLock1,cLock2#cLock2]-->cLock1
  </type>
  <Position>
    <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--ro, req, int, x坐标-->1
    </positionX>
    <positionY>
      <!--ro, req, int, y坐标-->1
    </positionY>
    <height>
      <!--ro, req, int, 宽-->1
    </height>
    <width>
      <!--ro, req, int, 高-->1
    </width>
  </Position>
</ClockIcon>
<YmdParam>
  <!--ro, req, object, 时钟年月日参数-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <fontSize>
    <!--ro, req, int, 字体大小-->1
  </fontSize>
  <FontColor>
    <!--ro, req, object, 字体颜色-->
    <RGB>
      <!--ro, opt, int, RGB值, desc:十进制数据表示三原色, 如16777215表示0xFFFFFF-->1
    </RGB>
  </FontColor>
  <BackColor>
    <!--ro, req, object, 背景色-->
    <RGB>
      <!--ro, opt, int, RGB值, desc:十进制数据表示三原色, 如16777215表示0xFFFFFF-->1
    </RGB>
  </BackColor>
  <Position>
    <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--ro, req, int, x坐标-->1
    </positionX>
    <positionY>
      <!--ro, req, int, y坐标-->1
    </positionY>
    <height>
      <!--ro, req, int, 宽-->1
    </height>
    <width>
      <!--ro, req, int, 高-->1
    </width>
  </Position>
</YmdParam>
<HmsParam>
  <!--ro, req, object, 时钟时分秒参数-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <fontSize>
    <!--ro, req, int, 字体大小-->1
  </fontSize>
  <FontColor>
    <!--ro, req, object, 字体颜色-->
    <RGB>
      <!--ro, opt, int, RGB值, desc:十进制数据表示三原色, 如16777215表示0xFFFFFF-->1
    </RGB>
  </FontColor>
  <BackColor>
    <!--ro, req, object, 背景色-->
    <RGB>
      <!--ro, opt, int, RGB值, desc:十进制数据表示三原色, 如16777215表示0xFFFFFF-->1
    </RGB>
  </BackColor>
  <Position>
    <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--ro, req, int, x坐标-->1
    </positionX>
    <positionY>
      <!--ro, req, int, y坐标-->1
    </positionY>
    <height>
```

```
<!--ro, req, int, 宽-->1
</height>
<width>
<!--ro, req, int, 高-->1
</width>
</Position>
</HmsParam>
<WeekParam>
<!--ro, req, object, 时钟星期参数-->
<enabled>
<!--ro, req, bool, 使能-->true
</enabled>
<fontSize>
<!--ro, req, int, 字体大小-->1
</fontSize>
<FontColor>
<!--ro, req, object, 字体颜色-->
<RGB>
<!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
</RGB>
</FontColor>
<BackColor>
<!--ro, req, object, 背景色-->
<RGB>
<!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
</RGB>
</BackColor>
<Position>
<!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
<positionX>
<!--ro, req, int, x坐标-->1
</positionX>
<positionY>
<!--ro, req, int, y坐标-->1
</positionY>
<height>
<!--ro, req, int, 宽-->1
</height>
<width>
<!--ro, req, int, 高-->1
</width>
</Position>
</WeekParam>
</ClockParam>
<WeatherParam>
<!--ro, opt, object, 天气参数-->
<backPicId>
<!--ro, opt, int, 天气背景图片素材id-->1
</backPicId>
<WeatherIcon>
<!--ro, opt, object, 天气图标参数-->
<enabled>
<!--ro, req, bool, 使能-->true
</enabled>
<Position>
<!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
<positionX>
<!--ro, req, int, x坐标-->1
</positionX>
<positionY>
<!--ro, req, int, y坐标-->1
</positionY>
<height>
<!--ro, req, int, 宽-->1
</height>
<width>
<!--ro, req, int, 高-->1
</width>
</Position>
</WeatherIcon>
<Date>
<!--ro, opt, object, 天气日期参数-->
<enabled>
<!--ro, req, bool, 使能-->true
</enabled>
<fontSize>
<!--ro, req, int, 字体大小-->1
</fontSize>
<FontColor>
<!--ro, req, object, 字体颜色-->
<RGB>
<!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
</RGB>
</FontColor>
<BackColor>
<!--ro, req, object, 背景色-->
<RGB>
<!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
</RGB>
</BackColor>
<Position>
<!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
<positionX>
```

```
<!--ro, req, int, x坐标-->1
</positionX>
<positionY>
<!--ro, req, int, y坐标-->1
</positionY>
<height>
<!--ro, req, int, 宽-->1
</height>
<width>
<!--ro, req, int, 高-->1
</width>
</Position>
</Date>
<Temperature>
<!--ro, opt, object, 天气温度参数-->
<enabled>
<!--ro, req, bool, 使能-->true
</enabled>
<fontSize>
<!--ro, req, int, 字体大小-->1
</fontSize>
<FontColor>
<!--ro, req, object, 字体颜色-->
<RGB>
<!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
</RGB>
</FontColor>
<BackColor>
<!--ro, req, object, 背景色-->
<RGB>
<!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
</RGB>
</BackColor>
<Position>
<!--ro, opt, object, 位置, desc:左上角为坐标原点,取值范围和设备分辨率保持一致-->
<positionX>
<!--ro, req, int, x坐标-->1
</positionX>
<positionY>
<!--ro, req, int, y坐标-->1
</positionY>
<height>
<!--ro, req, int, 宽-->1
</height>
<width>
<!--ro, req, int, 高-->1
</width>
</Position>
</Temperature>
<WeatherContent>
<!--ro, opt, object, 天气情况参数-->
<enabled>
<!--ro, req, bool, 使能-->true
</enabled>
<fontSize>
<!--ro, req, int, 字体大小-->1
</fontSize>
<FontColor>
<!--ro, req, object, 字体颜色-->
<RGB>
<!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
</RGB>
</FontColor>
<BackColor>
<!--ro, req, object, 背景色-->
<RGB>
<!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
</RGB>
</BackColor>
<Position>
<!--ro, opt, object, 位置, desc:左上角为坐标原点,取值范围和设备分辨率保持一致-->
<positionX>
<!--ro, req, int, x坐标-->1
</positionX>
<positionY>
<!--ro, req, int, y坐标-->1
</positionY>
<height>
<!--ro, req, int, 宽-->1
</height>
<width>
<!--ro, req, int, 高-->1
</width>
</Position>
</WeatherContent>
<City>
<!--ro, opt, object, 天气城市参数-->
<enabled>
<!--ro, req, bool, 使能-->true
</enabled>
<cityId>
<!--ro, req, string, 城市编号-->test
</cityId>
<cityName>
```

```
<!--ro, req, string, 城市名称-->test
</cityName>
<fontSize>
<!--ro, req, int, 字体大小-->1
</fontSize>
<FontColor>
<!--ro, req, object, 字体颜色-->
<RGB>
<!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
</RGB>
</FontColor>
<BackColor>
<!--ro, req, object, 背景色-->
<RGB>
<!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
</RGB>
</BackColor>
<Position>
<!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
<positionX>
<!--ro, req, int, x坐标-->1
</positionX>
<positionY>
<!--ro, req, int, y坐标-->1
</positionY>
<height>
<!--ro, req, int, 宽-->1
</height>
<width>
<!--ro, req, int, 高-->1
</width>
</Position>
</City>
<Humidity>
<!--ro, opt, object, 天气湿度参数-->
<enabled>
<!--ro, req, bool, 使能-->true
</enabled>
<fontSize>
<!--ro, req, int, 字体大小-->1
</fontSize>
<FontColor>
<!--ro, req, object, 字体颜色-->
<RGB>
<!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
</RGB>
</FontColor>
<BackColor>
<!--ro, req, object, 背景色-->
<RGB>
<!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
</RGB>
</BackColor>
<Position>
<!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
<positionX>
<!--ro, req, int, x坐标-->1
</positionX>
<positionY>
<!--ro, req, int, y坐标-->1
</positionY>
<height>
<!--ro, req, int, 宽-->1
</height>
<width>
<!--ro, req, int, 高-->1
</width>
</Position>
</Humidity>
<AirQuality>
<!--ro, opt, object, 天气空气质量参数-->
<enabled>
<!--ro, req, bool, 使能-->true
</enabled>
<fontSize>
<!--ro, req, int, 字体大小-->1
</fontSize>
<FontColor>
<!--ro, req, object, 字体颜色-->
<RGB>
<!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
</RGB>
</FontColor>
<BackColor>
<!--ro, req, object, 背景色-->
<RGB>
<!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
</RGB>
</BackColor>
<Position>
<!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
<positionX>
<!--ro, req, int, x坐标-->1
```

```
</positionX>
<positionY>
  <!--ro, req, int, y坐标-->1
</positionY>
<height>
  <!--ro, req, int, 宽-->1
</height>
<width>
  <!--ro, req, int, 高-->1
</width>
</Position>
</AirQuality>
<UpdateTime>
  <!--ro, opt, object, 天气更新时间参数-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <refreshTime>
    <!--ro, req, string, 刷新时间, desc:设备早期返回时间格式有误, 错为: 20200826T051500+08, 为保证XSD校验通过, 改为string类型-->20200826T051500+08
  </refreshTime>
  <updateInterval>
    <!--ro, req, int, 刷新间隔, unit:minute-->1
  </updateInterval>
  <fontSize>
    <!--ro, req, int, 字体大小-->1
  </fontSize>
  <FontColor>
    <!--ro, req, object, 字体颜色-->
    <RGB>
      <!--ro, opt, int, RGB值, desc:十进制数据表示三原色, 如16777215表示0xFFFFFF-->1
    </RGB>
  </FontColor>
  <BackColor>
    <!--ro, req, object, 背景色-->
    <RGB>
      <!--ro, opt, int, RGB值, desc:十进制数据表示三原色, 如16777215表示0xFFFFFF-->1
    </RGB>
  </BackColor>
  <Position>
    <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--ro, req, int, x坐标-->1
    </positionX>
    <positionY>
      <!--ro, req, int, y坐标-->1
    </positionY>
    <height>
      <!--ro, req, int, 宽-->1
    </height>
    <width>
      <!--ro, req, int, 高-->1
    </width>
  </Position>
</UpdateTime>
<Wind>
  <!--ro, opt, object, 天气风力参数-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <fontSize>
    <!--ro, req, int, 字体大小-->1
  </fontSize>
  <FontColor>
    <!--ro, req, object, 字体颜色-->
    <RGB>
      <!--ro, opt, int, RGB值, desc:十进制数据表示三原色, 如16777215表示0xFFFFFF-->1
    </RGB>
  </FontColor>
  <BackColor>
    <!--ro, req, object, 背景色-->
    <RGB>
      <!--ro, opt, int, RGB值, desc:十进制数据表示三原色, 如16777215表示0xFFFFFF-->1
    </RGB>
  </BackColor>
  <Position>
    <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--ro, req, int, x坐标-->1
    </positionX>
    <positionY>
      <!--ro, req, int, y坐标-->1
    </positionY>
    <height>
      <!--ro, req, int, 宽-->1
    </height>
    <width>
      <!--ro, req, int, 高-->1
    </width>
  </Position>
</Wind>
</WeatherParam>
<Countdown>
  <!--ro, opt, object, 倒计时素材-->
<endTime>
```

```
<!--ro, req, string, 倒计时时间点, desc:设备早期返回时间格式有误, 错为: 20200826T051500+08, 为保证XSD校验通过, 改为string类型-->20200826T051500+08
</endTime>
<template>
  <!--ro, req, enum, 模板类型, subType:string, [template1#模板1,template2#模板2,openingActivity1#活动开幕1,openingActivity2#活动开幕
2,commercialActivity1#商业活动1,commercialActivity2#商业活动2,colLegeEntranceExam#高考,overseasTimeFormat#海外时间格式]-->template1
</template>
<timeUnit>
  <!--ro, req, enum, 时间单位, subType:string, [year#年,month#月,day#日,week#周,hour#时,minute#分,second#秒]-->year
</timeUnit>
<backPicId>
  <!--ro, opt, int, 背景图片素材ID-->1
</backPicId>
<TimeFontCfg>
  <!--ro, opt, object, 时间字体参数-->
  <fontSize>
    <!--ro, req, int, 字体大小-->1
  </fontSize>
  <FontColor>
    <!--ro, req, object, 字体颜色-->
    <RGB>
      <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
    </RGB>
    </FontColor>
  <Position>
    <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--ro, req, int, x坐标-->1
    </positionX>
    <positionY>
      <!--ro, req, int, y坐标-->1
    </positionY>
    <height>
      <!--ro, req, int, 宽-->1
    </height>
    <width>
      <!--ro, req, int, 高-->1
    </width>
  </Position>
</TimeFontCfg>
<timeUnitEnabled>
  <!--ro, opt, bool, 时间单位是否有效-->true
</timeUnitEnabled>
<maxTimeUnit>
  <!--ro, req, enum, 最大时间单位, subType:string, [year#年,month#月,day#日,week#周,hour#时,minute#分,second#秒,week#周], desc:表示当前倒计时的最大
显示单元,例如:maxTimeUnit为月,则1年3个月,对应月份值为15-->year
</maxTimeUnit>
<ModuleInfoList>
  <!--ro, opt, array, 界面模块信息列表, subType:object-->
  <ModuleInfo>
    <!--ro, opt, object, 界面模块信息-->
    <moduleType>
      <!--ro, req, enum, 模块类型, subType:string, [year#年,month#月,day#日,hour#时,minute#分,second#秒,week#周,text#文字], desc:text为时间旁边的文
字,例如年下面的文字"年"-->second
    </moduleType>
    <enabled>
      <!--ro, req, bool, 使能-->true
    </enabled>
    <fontSize>
      <!--ro, req, int, 字体大小-->1
    </fontSize>
    <FontColor>
      <!--ro, req, object, 字体颜色-->
      <RGB>
        <!--ro, opt, int, 十进制数据表示三原色-->1
      </RGB>
      </FontColor>
    <text>
      <!--ro, opt, string, 文字, desc:当moduleType为text时有效-->test
    </text>
    <Position>
      <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
      <positionX>
        <!--ro, req, int, X坐标-->1
      </positionX>
      <positionY>
        <!--ro, req, int, Y坐标-->1
      </positionY>
      <height>
        <!--ro, req, int, 宽-->1
      </height>
      <width>
        <!--ro, req, int, 高-->1
      </width>
    </Position>
  </ModuleInfo>
</ModuleInfoList>
</Countdown>
<localInputNo>
  <!--ro, req, enum, 本地输入类型, subType:string, [VGA#VGA,HDMI#HDMI]-->VGA
</localInputNo>
<HyperlinkBtn>
  <!--ro, opt, object, 超链接按钮-->
  <backPicId>
```

```
<!--ro, opt, int, 背景图片素材ID-->1
</backPicId>
</HyperlinkBtn>
<CharactersAttribute>
<!--ro, opt, object, 文字属性, desc: 当dynamicType为character时有效-->
<fontSize>
<!--ro, req, int, 字体大小-->1
</fontSize>
<fontColor>
<!--ro, req, object, 字体颜色-->
<RGB>
<!--ro, opt, int, RGB值, desc: 十进制数据表示三原色, 如16777215表示0xFFFFFF-->1
</RGB>
</FontColor>
<fontType>
<!--ro, opt, enum, 字体类型, subType:string, [normal#常规,bold#粗体]-->normal
</fontType>
<backgroundType>
<!--ro, req, enum, 背景类型, subType:string, [color#颜色,picture#图片]-->color
</backgroundType>
<BackColor>
<!--ro, req, object, 背景色-->
<RGB>
<!--ro, opt, int, RGB值, desc: 十进制数据表示三原色, 如16777215表示0xFFFFFF-->1
</RGB>
<fillEnabled>
<!--ro, opt, bool, 背景色填充使能, desc: 默认为true, 表示背景色填充-->true
</fillEnabled>
</BackColor>
<backTransparent>
<!--ro, opt, int, 背景透明度-->1
</backTransparent>
<alignType>
<!--ro, req, enum, 对齐方式, subType:string, [left#左,right#右,middle#中]-->middle
</alignType>
<verticalAlignType>
<!--ro, req, enum, 垂直对齐方式, subType:string, [top#顶,bottom#底,verticalCenter#中]-->top
</verticalAlignType>
<characterContent>
<!--ro, opt, string, 文字内容, range:[0,512]-->test
</characterContent>
<backPicId>
<!--ro, opt, int, 背景图片素材ID-->1
</backPicId>
<characterStyle>
<!--ro, req, enum, 字体效果, subType:string, [normal#常规,bold#粗体]-->normal
</characterStyle>
<characterType>
<!--ro, req, enum, 字体类型, subType:string, [mainTitle#主标题,subTitle#副标题]-->mainTitle
</characterType>
<scrollDirection>
<!--ro, opt, enum, 文字滚动方向, subType:string, [left#向左,right#向右,up#向上,down#向下,turnPage#翻页]-->left
</scrollDirection>
<scrollSpeed>
<!--ro, opt, int, 文字滚动速度, range:[1,10]-->1
</scrollSpeed>
</CharactersAttribute>
<SignIn>
<!--ro, opt, object, 人员签到展示窗口, desc: 当staticMaterialType为signIn时有效-->
<backPicId>
<!--ro, opt, int, 背景图片素材ID-->1
</backPicId>
<ShowInfo>
<!--ro, opt, object, 展示信息-->
<row>
<!--ro, opt, int, 窗口内人员签到展示行数-->1
</row>
<column>
<!--ro, opt, int, 窗口内人员签到展示列数-->1
</column>
<Position>
<!--ro, opt, object, 位置, desc: 左上角为坐标原点, 取值范围和设备分辨率保持一致-->
<positionX>
<!--ro, req, int, x坐标-->1
</positionX>
<positionY>
<!--ro, req, int, y坐标-->1
</positionY>
<height>
<!--ro, req, int, 宽-->1
</height>
<width>
<!--ro, req, int, 高-->1
</width>
</Position>
<inLib>
<!--ro, opt, object, 在库信息-->
<nameEnabled>
<!--ro, opt, bool, 在库人员是否展示姓名-->true
</nameEnabled>
<genderEnabled>
<!--ro, opt, bool, 在库人员是否展示性别-->true
</genderEnabled>
<selfDefineEnabled>
```

```
<!--ro, opt, bool, 在库人员是否展示自定义字段1-->true
</selfDefine1Enabled>
<selfDefine2Enabled>
  <!--ro, opt, bool, 在库人员是否展示自定义字段2-->true
</selfDefine2Enabled>
<signInTimeEnabled>
  <!--ro, opt, bool, 在库人员是否展示签到时间-->true
</signInTimeEnabled>
<temperatureEnabled>
  <!--ro, opt, bool, 在库人员是否展示温度-->true
</temperatureEnabled>
</inLib>
<outLib>
  <!--ro, opt, object, 不在库信息-->
  <nameEnabled>
    <!--ro, opt, bool, 未知人员是否展示姓名-->true
  </nameEnabled>
  <name>
    <!--ro, opt, string, 未知人员姓名-->test
  </name>
  <genderEnabled>
    <!--ro, opt, bool, 未知人员是否展示性别-->true
  </genderEnabled>
  <signInTimeEnabled>
    <!--ro, opt, bool, 未知人员是否展示签到时间-->true
  </signInTimeEnabled>
  <temperatureEnabled>
    <!--ro, opt, bool, 未知人员是否展示温度-->true
  </temperatureEnabled>
</outLib>
<showDuration>
  <!--ro, req, enum, 签到记录显示时间, subType:string, [always#不消失,10s#10s,20s#20s,30s#30s,no#不显示]-->always
</showDuration>
</ShowInfo>
<MainTitle>
  <!--ro, opt, object, 主标题-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <fontSize>
    <!--ro, req, int, 字体大小-->1
  </fontSize>
  <FontColor>
    <!--ro, req, object, 字体颜色-->
    <RGB>
      <!--ro, opt, int, 十进制数据表示三原色-->1
    </RGB>
  </FontColor>
  <Position>
    <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--ro, req, int, x坐标-->1
    </positionX>
    <positionY>
      <!--ro, req, int, y坐标-->1
    </positionY>
    <height>
      <!--ro, req, int, 宽-->1
    </height>
    <width>
      <!--ro, req, int, 高-->1
    </width>
  </Position>
  <text>
    <!--ro, opt, string, 标题文字-->test
  </text>
</MainTitle>
<SubTitle>
  <!--ro, opt, object, 副标题-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <fontSize>
    <!--ro, req, int, 字体大小-->1
  </fontSize>
  <FontColor>
    <!--ro, req, object, 字体颜色-->
    <RGB>
      <!--ro, opt, int, 十进制数据表示三原色-->1
    </RGB>
  </FontColor>
  <Position>
    <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--ro, req, int, x坐标-->1
    </positionX>
    <positionY>
      <!--ro, req, int, y坐标-->1
    </positionY>
    <height>
      <!--ro, req, int, 宽-->1
    </height>
    <width>
```

```
<!--ro, req, int, 高-->1
</width>
</Position>
<text>
<!--ro, opt, string, 标题文字-->test
</text>
</SubTitle>
<NormalTemperature>
<!--ro, opt, object, 正常体温信息-->
<PeopleNumber>
<!--ro, opt, object, 正常体温人数信息-->
<enabled>
<!--ro, req, bool, 使能-->true
</enabled>
<fontSize>
<!--ro, req, int, 字体大小-->1
</fontSize>
<fontColor>
<!--ro, req, object, 字体颜色-->
<RGB>
<!--ro, opt, int, 十进制数据表示三原色-->1
</RGB>
</FontColor>
<Position>
<!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
<positionX>
<!--ro, req, int, X坐标-->1
</positionX>
<positionY>
<!--ro, req, int, Y坐标-->1
</positionY>
<height>
<!--ro, req, int, 宽-->1
</height>
<width>
<!--ro, req, int, 高-->1
</width>
</Position>
<text>
<!--ro, opt, string, 标题文字-->test
</text>
</PeopleNumber>
<Icon>
<!--ro, opt, object, 正常体温图标-->
<enabled>
<!--ro, req, bool, 使能-->true
</enabled>
<Position>
<!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
<positionX>
<!--ro, req, int, X坐标-->1
</positionX>
<positionY>
<!--ro, req, int, Y坐标-->1
</positionY>
<height>
<!--ro, req, int, 宽-->1
</height>
<width>
<!--ro, req, int, 高-->1
</width>
</Position>
<materialNo>
<!--ro, opt, int, 素材ID-->1
</materialNo>
</Icon>
</NormalTemperature>
<AbnormalTemperature>
<!--ro, opt, object, 异常体温信息-->
<PeopleNumber>
<!--ro, opt, object, 异常体温人数信息-->
<enabled>
<!--ro, req, bool, 使能-->true
</enabled>
<fontSize>
<!--ro, req, int, 字体大小-->1
</fontSize>
<fontColor>
<!--ro, req, object, 字体颜色-->
<RGB>
<!--ro, opt, int, 十进制数据表示三原色-->1
</RGB>
</FontColor>
<Position>
<!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
<positionX>
<!--ro, req, int, X坐标-->1
</positionX>
<positionY>
<!--ro, req, int, Y坐标-->1
</positionY>
<height>
<!--ro, req, int, 宽-->1
</height>
<width>
<!--ro, req, int, 高-->1
</width>
```

```
</neight>
<width>
  <!--ro, req, int, 高-->1
</width>
</Position>
<text>
  <!--ro, opt, string, 标题文字-->test
</text>
</PeopleNumber>
<Icon>
  <!--ro, opt, object, 异常体温图标-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <Position>
    <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--ro, req, int, X坐标-->1
    </positionX>
    <positionY>
      <!--ro, req, int, Y坐标-->1
    </positionY>
    <height>
      <!--ro, req, int, 宽-->1
    </height>
    <width>
      <!--ro, req, int, 高-->1
    </width>
  </Position>
  <materialNo>
    <!--ro, opt, int, 素材ID-->1
  </materialNo>
</Icon>
</AbnormalTemperature>
</SignIn>
<EventParam>
  <!--ro, opt, object, 事件参数-->
  <EventResourceList>
    <!--ro, opt, array, 事件源信息列表, subType:object-->
    <EventResource>
      <!--ro, opt, object, 事件源信息-->
      <ipVersion>
        <!--ro, req, enum, ip地址版本, subType:string, [v4#v4,v6#v6,dual#双]-->v4
      </ipVersion>
      <ipAddress>
        <!--ro, opt, string, ipv4地址-->test
      </ipAddress>
      <ipv6Address>
        <!--ro, opt, string, ipv6地址-->test
      </ipv6Address>
      <portNo>
        <!--ro, req, int, 端口号-->1
      </portNo>
      <userName>
        <!--ro, req, string, 用户名-->test
      </userName>
      <password>
        <!--ro, req, string, 密码-->test
      </password>
      <channels>
        <!--ro, opt, string, 通道号, desc:多选以逗号隔开-->1,2
      </channels>
    </EventResource>
  </EventResourceList>
  <backPicId>
    <!--ro, opt, int, 背景图片素材ID-->1
  </backPicId>
  <Title>
    <!--ro, opt, object, 标题-->
    <enabled>
      <!--ro, req, bool, 使能-->true
    </enabled>
    <fontSize>
      <!--ro, req, int, 字体大小-->1
    </fontSize>
    <FontColor>
      <!--ro, req, object, 字体颜色-->
      <RGB>
        <!--ro, opt, int, 十进制数据表示三原色-->1
      </RGB>
    </FontColor>
    <Position>
      <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
      <positionX>
        <!--ro, req, int, X坐标-->1
      </positionX>
      <positionY>
        <!--ro, req, int, Y坐标-->1
      </positionY>
      <height>
        <!--ro, req, int, 宽-->1
      </height>
      <width>
        <!--ro, req, int, 高-->1
      </width>
```

```
</width>
</Position>
<text>
  <!--ro, opt, string, 标题文字-->test
</text>
</Title>
<PeopleCounting>
  <!--ro, opt, object, 人数统计提醒-->
  <maxPeopleNumber>
    <!--ro, opt, int, 最大可进入人数-->1
  </maxPeopleNumber>
  <unavailableBackPicId>
    <!--ro, opt, int, 人员不能进入时控件背景图片id-->1
  </unavailableBackPicId>
  <supportChangeByPeople>
    <!--ro, req, enum, 支持根据人数修改样式的字段, subType:string,
[prompt#prompt,peopleIcon#peopleIcon,availablePeopleNumber#availablePeopleNumber]-->prompt
  </supportChangeByPeople>
  <Prompt>
    <!--ro, opt, object, 提示信息-->
    <enabled>
      <!--ro, req, bool, 使能-->true
    </enabled>
    <fontSize>
      <!--ro, req, int, 字体大小-->1
    </fontSize>
    <Position>
      <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
      <positionX>
        <!--ro, req, int, X坐标-->1
      </positionX>
      <positionY>
        <!--ro, req, int, Y坐标-->1
      </positionY>
      <height>
        <!--ro, req, int, 宽-->1
      </height>
      <width>
        <!--ro, req, int, 高-->1
      </width>
    </Position>
    <text>
      <!--ro, opt, string, 标题文字-->test
    </text>
    <FontColor>
      <!--ro, opt, object, 人员可以进入时的字体颜色-->
      <RGB>
        <!--ro, opt, int, 十进制数据表示三原色-->1
      </RGB>
    </FontColor>
    <unavailableText>
      <!--ro, opt, string, 人员不能进入时的标题文字-->test
    </unavailableText>
    <UnavailableFontColor>
      <!--ro, opt, object, 人员不能进入时的字体颜色-->
      <RGB>
        <!--ro, opt, int, 十进制数据表示三原色-->1
      </RGB>
    </UnavailableFontColor>
  </Prompt>
  <PeopleIcon>
    <!--ro, opt, object, 行人通行图标-->
    <enabled>
      <!--ro, req, bool, 使能-->true
    </enabled>
    <Position>
      <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
      <positionX>
        <!--ro, req, int, X坐标-->1
      </positionX>
      <positionY>
        <!--ro, req, int, Y坐标-->1
      </positionY>
      <height>
        <!--ro, req, int, 宽-->1
      </height>
      <width>
        <!--ro, req, int, 高-->1
      </width>
    </Position>
    <materialNo>
      <!--ro, opt, int, 素材ID-->1
    </materialNo>
  </PeopleIcon>
  <InsidePeopleTitle>
    <!--ro, opt, object, 已经进入人数标题-->
    <enabled>
      <!--ro, req, bool, 使能-->true
    </enabled>
    <fontSize>
      <!--ro, req, int, 字体大小-->1
    </fontSize>
    <FontColor>
```

```
<!--ro, req, object, 字体颜色-->
<RGB>
<!--ro, opt, int, 十进制数据表示三原色-->1
</RGB>
</FontColor>
<Position>
<!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
<positionX>
<!--ro, req, int, X坐标-->1
</positionX>
<positionY>
<!--ro, req, int, Y坐标-->1
</positionY>
<height>
<!--ro, req, int, 宽-->1
</height>
<width>
<!--ro, req, int, 高-->1
</width>
</Position>
<text>
<!--ro, opt, string, 标题文字-->test
</text>
</InsidePeopleTitle>
<InsidePeopleNumber>
<!--ro, opt, object, 已经进入人数-->
<enabled>
<!--ro, req, bool, 使能-->true
</enabled>
<fontSize>
<!--ro, req, int, 字体大小-->1
</fontSize>
<FontColor>
<!--ro, req, object, 字体颜色-->
<RGB>
<!--ro, opt, int, 十进制数据表示三原色-->1
</RGB>
</FontColor>
<Position>
<!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
<positionX>
<!--ro, req, int, X坐标-->1
</positionX>
<positionY>
<!--ro, req, int, Y坐标-->1
</positionY>
<height>
<!--ro, req, int, 宽-->1
</height>
<width>
<!--ro, req, int, 高-->1
</width>
</Position>
</InsidePeopleNumber>
<AvailablePeopleTitle>
<!--ro, opt, object, 还可以进入人数标题-->
<enabled>
<!--ro, req, bool, 使能-->true
</enabled>
<fontSize>
<!--ro, req, int, 字体大小-->1
</fontSize>
<FontColor>
<!--ro, req, object, 字体颜色-->
<RGB>
<!--ro, opt, int, 十进制数据表示三原色-->1
</RGB>
</FontColor>
<Position>
<!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
<positionX>
<!--ro, req, int, X坐标-->1
</positionX>
<positionY>
<!--ro, req, int, Y坐标-->1
</positionY>
<height>
<!--ro, req, int, 宽-->1
</height>
<width>
<!--ro, req, int, 高-->1
</width>
</Position>
<text>
<!--ro, opt, string, 标题文字-->test
</text>
</AvailablePeopleTitle>
<AvailablePeopleNumber>
<!--ro, opt, object, 还可以进入人数-->
<enabled>
<!--ro, req, bool, 使能-->true
</enabled>
<fontSize>
<!--ro, req, int, 字体大小-->1
</fontSize>
```

```
</FontSize>
<Position>
  <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
  <positionX>
    <!--ro, req, int, X坐标-->1
  </positionX>
  <positionY>
    <!--ro, req, int, Y坐标-->1
  </positionY>
  <height>
    <!--ro, req, int, 宽-->1
  </height>
  <width>
    <!--ro, req, int, 高-->1
  </width>
</Position>
<FontColor>
  <!--ro, opt, object, 人员可以进入时的字体颜色-->
  <RGB>
    <!--ro, opt, int, 十进制数据表示三原色-->1
  </RGB>
</FontColor>
<UnavailableFontColor>
  <!--ro, opt, object, 人员不能进入时的字体颜色-->
  <RGB>
    <!--ro, opt, int, 十进制数据表示三原色-->1
  </RGB>
</UnavailableFontColor>
</AvailablePeopleNumber>
<UnavailablePeopleIcon>
  <!--ro, opt, object, 人员不能进入时的行人图标-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <Position>
    <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--ro, req, int, X坐标-->1
    </positionX>
    <positionY>
      <!--ro, req, int, Y坐标-->1
    </positionY>
    <height>
      <!--ro, req, int, 宽-->1
    </height>
    <width>
      <!--ro, req, int, 高-->1
    </width>
  </Position>
  <materialNo>
    <!--ro, opt, int, 素材ID-->1
  </materialNo>
</UnavailablePeopleIcon>
<AudioAlarm>
  <!--ro, opt, object, 音频报警信息-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <materialNo>
    <!--ro, opt, int, 音频素材ID-->1
  </materialNo>
</AudioAlarm>
</PeopleCounting>
<Mask>
  <!--ro, opt, object, 未戴口罩提醒-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <fontSize>
    <!--ro, req, int, 字体大小-->1
  </fontSize>
  <FontColor>
    <!--ro, req, object, 字体颜色-->
    <RGB>
      <!--ro, opt, int, 十进制数据表示三原色-->1
    </RGB>
  </FontColor>
  <BackColor>
    <!--ro, req, object, 背景颜色-->
    <RGB>
      <!--ro, opt, int, 十进制数据表示三原色-->1
    </RGB>
  </BackColor>
  <Position>
    <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--ro, req, int, X坐标-->1
    </positionX>
    <positionY>
      <!--ro, req, int, Y坐标-->1
    </positionY>
    <height>
      <!--ro, req, int, 宽-->1
```

```
</height>
<width>
  <!--ro, req, int, 高-->1
</width>
</Position>
<text>
  <!--ro, opt, string, 标题文字-->test
</text>
<MaskIcon>
  <!--ro, opt, object, 未戴口罩图标-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <Position>
    <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--ro, req, int, X坐标-->1
    </positionX>
    <positionY>
      <!--ro, req, int, Y坐标-->1
    </positionY>
    <height>
      <!--ro, req, int, 宽-->1
    </height>
    <width>
      <!--ro, req, int, 高-->1
    </width>
  </Position>
  <materialNo>
    <!--ro, opt, int, 素材ID-->1
  </materialNo>
</MaskIcon>
<AudioAlarm>
  <!--ro, opt, object, 音频报警信息-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <materialNo>
    <!--ro, opt, int, 音频素材ID-->1
  </materialNo>
</AudioAlarm>
</Mask>
<Temperature>
  <!--ro, opt, object, 温度异常提醒-->
  <Prompt>
    <!--ro, opt, object, 提示信息-->
    <enabled>
      <!--ro, req, bool, 使能-->true
    </enabled>
    <fontSize>
      <!--ro, req, int, 字体大小-->1
    </fontSize>
    <fontColor>
      <!--ro, req, object, 字体颜色-->
      <RGB>
        <!--ro, opt, int, 十进制数据表示三原色-->1
      </RGB>
    </fontColor>
    <BackColor>
      <!--ro, req, object, 背景颜色-->
      <RGB>
        <!--ro, opt, int, 十进制数据表示三原色-->1
      </RGB>
    </BackColor>
  </Prompt>
  <Position>
    <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--ro, req, int, X坐标-->1
    </positionX>
    <positionY>
      <!--ro, req, int, Y坐标-->1
    </positionY>
    <height>
      <!--ro, req, int, 宽-->1
    </height>
    <width>
      <!--ro, req, int, 高-->1
    </width>
  </Position>
  <text>
    <!--ro, opt, string, 标题文字-->test
  </text>
</Prompt>
<TemperatureIcon>
  <!--ro, opt, object, 温度异常图标-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <Position>
    <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--ro, req, int, X坐标-->1
    </positionX>
    <positionY>
```

```
</Position>
<!--ro, req, int, Y坐标-->1
</positionY>
<height>
<!--ro, req, int, 宽-->1
</height>
<width>
<!--ro, req, int, 高-->1
</width>
</Position>
<materialNo>
<!--ro, opt, int, 素材ID-->1
</materialNo>
<TemperatureIcon>
<AudioAlarm>
<!--ro, opt, object, 音频报警信息-->
<enabled>
<!--ro, req, bool, 使能-->true
</enabled>
<materialNo>
<!--ro, opt, int, 音频素材ID-->1
</materialNo>
</AudioAlarm>
<abnormalHighTemperature>
<!--ro, opt, float, 异常高温值, unit: °C, unitType: 温度, desc: 保留一位小数-->37.5
</abnormalHighTemperature>
</Temperature>
<WearMask>
<!--ro, opt, object, 已戴口罩提醒-->
<enabled>
<!--ro, req, bool, 使能-->true
</enabled>
<fontSize>
<!--ro, req, int, 字体大小-->1
</fontSize>
<fontColor>
<!--ro, req, object, 字体颜色-->
<RGB>
<!--ro, opt, int, 十进制数据表示三原色-->1
</RGB>
</FontColor>
<BackColor>
<!--ro, req, object, 背景颜色-->
<RGB>
<!--ro, opt, int, 十进制数据表示三原色-->1
</RGB>
</BackColor>
<Position>
<!--ro, opt, object, 位置, desc: 左上角为坐标原点, 取值范围和设备分辨率保持一致-->
<positionX>
<!--ro, req, int, X坐标-->1
</positionX>
<positionY>
<!--ro, req, int, Y坐标-->1
</positionY>
<height>
<!--ro, req, int, 宽-->1
</height>
<width>
<!--ro, req, int, 高-->1
</width>
</Position>
<text>
<!--ro, opt, string, 标题文字-->test
</text>
<MaskIcon>
<!--ro, opt, object, 已戴口罩图标-->
<enabled>
<!--ro, req, bool, 使能-->true
</enabled>
<Position>
<!--ro, opt, object, 位置, desc: 左上角为坐标原点, 取值范围和设备分辨率保持一致-->
<positionX>
<!--ro, req, int, X坐标-->1
</positionX>
<positionY>
<!--ro, req, int, Y坐标-->1
</positionY>
<height>
<!--ro, req, int, 宽-->1
</height>
<width>
<!--ro, req, int, 高-->1
</width>
</Position>
<materialNo>
<!--ro, opt, int, 素材ID-->1
</materialNo>
</MaskIcon>
</WearMask>
<NormalTemperature>
<!--ro, opt, object, 温度正常提醒-->
<Prompt>
<!--ro, opt, object, 提示信息-->
```

```
<enabled>
  <!--ro, req, bool, 使能-->true
</enabled>
<fontSize>
  <!--ro, req, int, 字体大小-->1
</fontSize>
<fontColor>
  <!--ro, req, object, 字体颜色-->
  <RGB>
    <!--ro, opt, int, 十进制数据表示三原色-->1
  </RGB>
</FontColor>
<BackColor>
  <!--ro, req, object, 背景颜色-->
  <RGB>
    <!--ro, opt, int, 十进制数据表示三原色-->1
  </RGB>
</BackColor>
<Position>
  <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
  <positionX>
    <!--ro, req, int, X坐标-->1
  </positionX>
  <positionY>
    <!--ro, req, int, Y坐标-->1
  </positionY>
  <height>
    <!--ro, req, int, 宽-->1
  </height>
  <width>
    <!--ro, req, int, 高-->1
  </width>
</Position>
<text>
  <!--ro, opt, string, 标题文字-->test
</text>
</Prompt>
<TemperatureIcon>
  <!--ro, opt, object, 温度正常图标-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <Position>
    <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--ro, req, int, X坐标-->1
    </positionX>
    <positionY>
      <!--ro, req, int, Y坐标-->1
    </positionY>
    <height>
      <!--ro, req, int, 宽-->1
    </height>
    <width>
      <!--ro, req, int, 高-->1
    </width>
  </Position>
  <materialNo>
    <!--ro, opt, int, 素材ID-->1
  </materialNo>
</TemperatureIcon>
</NormalTemperature>
<NormalStatusAudioAlarm>
  <!--ro, opt, object, 正常状态下音频报警信息, desc:人员在检测时未发现上面的人数超限、未戴口罩、体温异常等事件时设备进行播报-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <materialNo>
    <!--ro, opt, int, 音频素材ID-->1
  </materialNo>
</NormalStatusAudioAlarm>
</EventParam>
<AlarmDataSourceProperty>
  <!--ro, opt, object, 数据源信息, desc:该字段能力集缺少, 仅信发组件和设备使用-->
  <PictureProperty>
    <!--ro, opt, object, 图片属性-->
    <PictureSwitchEffect>
      <!--ro, opt, enum, 图片切换效果, subType:string, [LeftInRightOut#左进右出,rightInLeftOut#右进左出]-->leftInRightOut
    </PictureSwitchEffect>
    <PictureScaleType>
      <!--ro, opt, enum, 图片缩放类型, subType:string, [fitCenter#拉伸填充,centerCrop#截取填充,fixXY#适应宽高]-->fitCenter
    </PictureScaleType>
  </PictureProperty>
  <AlarmDataSource>
    <!--ro, opt, object, 数据源-->
    <MaterialNo>
      <!--ro, opt, int, 素材编号-->1
    </MaterialNo>
    <Name>
      <!--ro, opt, string, 素材名称-->test
    </Name>
    <Address>
      <!--ro, opt, string, 素材URL地址-->test
    </Address>
```

```

</Address>
<WebProperty>
  <!--ro, opt, object, 网页属性-->
  <webType>
    <!--ro, opt, enum, 网页类型, subType:string, [desktop#桌面版,mobile#移动版], desc:默认按照设备的系统类型（Windows或Android）来显示网页-->
    >desktop
  </webType>
</WebProperty>
</AlarmDataSource>
</AlarmDataSourceProperty>
<ConferenceParam>
  <!--ro, opt, object, 会议信息, desc:当页面用于会议界面时有效-->
  <backPicId>
    <!--ro, opt, int, 背景图片素材ID-->1
  </backPicId>
  <Title>
    <!--ro, opt, object, 会议标题（主题）, desc:1、会议标题（主题）的文字内容通过场地预约配置协议下发,对应/ISAPI/Publish/Signage/placeReservation?format=json中的programID字段。2、可通过会议计划（/ISAPI/InteractivePanel/Meeting/plan?format=json）中的meetingTopic字段下发-->
    <enabled>
      <!--ro, req, bool, 使能-->true
    </enabled>
    <fontSize>
      <!--ro, req, int, 字体大小-->1
    </fontSize>
    <FontColor>
      <!--ro, req, object, 字体颜色-->
      <RGB>
        <!--ro, opt, int, 十进制数据表示三原色-->1
      </RGB>
    </FontColor>
    <BackColor>
      <!--ro, req, object, 背景颜色-->
      <RGB>
        <!--ro, opt, int, 十进制数据表示三原色-->1
      </RGB>
    </BackColor>
    <Position>
      <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
      <positionX>
        <!--ro, req, int, X坐标-->1
      </positionX>
      <positionY>
        <!--ro, req, int, Y坐标-->1
      </positionY>
      <height>
        <!--ro, req, int, 高-->1
      </height>
      <width>
        <!--ro, req, int, 宽-->1
      </width>
      <alignType>
        <!--ro, req, enum, 对齐方式, subType:string, [Left#左,right#右,middle#中]-->middle
      </alignType>
    </Position>
    <Title>
  </Title>
  <VoiceBroadcast>
    <!--ro, opt, object, 语音播报-->
    <enabled>
      <!--ro, opt, bool, 是否启用语音播报-->true
    </enabled>
    <broadcastSpeed>
      <!--ro, opt, enum, 播报语速, subType:string, [slower#慢,sLow#较慢,medium#中速,fast#较快,faster#快]-->slower
    </broadcastSpeed>
    <inLibVoice>
      <!--ro, opt, string, 在库人员语音播报内容-->test
    </inLibVoice>
    <outLibVoice>
      <!--ro, opt, string, 未知人员语音播报内容-->test
    </outLibVoice>
  </VoiceBroadcast>
  <WelcomeWord>
    <!--ro, opt, object, 欢迎词-->
    <enabled>
      <!--ro, opt, bool, 是否启用欢迎词-->true
    </enabled>
    <inLibWord>
      <!--ro, opt, string, 在库人员欢迎词内容, range:[1,128]-->test
    </inLibWord>
    <outLibWord>
      <!--ro, opt, string, 未知人员欢迎词内容, range:[1,128]-->test
    </outLibWord>
  </WelcomeWord>
  <ShowInfo>
    <!--ro, opt, object, 展示信息, desc:下发该字段需要人脸签到, 不下发该字段, 不需要人脸签到-->
    <row>
      <!--ro, opt, int, 窗口内人员签到展示行数-->1
    </row>
    <column>
      <!--ro, opt, int, 窗口内人员签到展示列数-->1
    </column>
    <Position>
      <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
      <positionX>
        <!--ro, req, int, x坐标-->1
      </positionX>
    </Position>
  </ShowInfo>

```

```

</positionX>
<positionY>
  <!--ro, req, int, y坐标-->1
</positionY>
<height>
  <!--ro, req, int, 宽-->1
</height>
<width>
  <!--ro, req, int, 高-->1
</width>
</Position>
<inLib>
  <!--ro, opt, object, 在库信息-->
  <nameEnabled>
    <!--ro, opt, bool, 在库人员是否展示姓名-->true
  </nameEnabled>
</inLib>
<defaultShowInfoEnabled>
  <!--ro, opt, bool, 是否采用人员默认展示信息, desc: 下发该字段, 且为true, 表示人员展示信息采用默认值。坐标信息等不用下发-->true
</defaultShowInfoEnabled>
</ShowInfo>
<ModuleInfoList>
  <!--ro, opt, array, 界面模块信息列表, subType:object-->
  <ModuleInfo>
    <!--ro, opt, object, 界面模块信息-->
    <moduleType>
      <!--ro, req, enum, 模块类型, subType:string, [conferenceTime#会议时间(对应timeRange), conferenceSite#会议地点(对应会议室名称
meetingName), conferenceStatus#会议状态(对应meetingStatus), signInPersonNum#已签到人数, totalPersonNum#应到人数, unSignInPersonNum#未签到人
数, signInAndTotalPersonNum#已签到/应到人数, QRCode#会议室二维码, meetingHostName#主持人, employeeNo#参会人, customKey#自定义字段, nextMeeting#当天下一场会议信
息]-->conferenceTime
    </moduleType>
    <enabled>
      <!--ro, req, bool, 使能-->true
    </enabled>
    <fontSize>
      <!--ro, req, int, 字体大小-->1
    </fontSize>
    <fontColor>
      <!--ro, req, object, 字体颜色-->
      <RGB>
        <!--ro, opt, int, 十进制数据表示三原色-->1
      </RGB>
    </fontColor>
    <BackColor>
      <!--ro, opt, object, 背景颜色-->
      <RGB>
        <!--ro, opt, int, 十进制数据表示三原色-->1
      </RGB>
    </BackColor>
    <backPicId>
      <!--ro, opt, int, 背景图片素材ID, desc: 背景图和背景颜色同时仅支持一种, 两者均下发, 设备优先解析背景图字段-->1
    </backPicId>
    <Position>
      <!--ro, opt, object, 位置, desc: 左上角为坐标原点, 取值范围和设备分辨率保持一致-->
      <positionX>
        <!--ro, req, int, X坐标-->1
      </positionX>
      <positionY>
        <!--ro, req, int, Y坐标-->1
      </positionY>
      <height>
        <!--ro, req, int, 宽-->1
      </height>
      <width>
        <!--ro, req, int, 高-->1
      </width>
    </Position>
    <alignType>
      <!--ro, req, enum, 对齐方式, subType:string, [left#左, right#右, middle#中]-->middle
    </alignType>
    <CustomKey>
      <!--ro, opt, object, 自定义字段-->
      <key>
        <!--ro, opt, string, 自定义字段-->test
      </key>
      <associateProtocolURL>
        <!--ro, opt, string, 关联协议URL-->test
      </associateProtocolURL>
    </CustomKey>
  </ModuleInfo>
</ModuleInfoList>
<conferenceStatus>
  <!--ro, opt, enum, 会议状态, subType:string, [idle#空闲, signing#签到中, meeting#开会中], desc: 根据不同的会议状态, 配置不同的会议界面-->idle
</conferenceStatus>
<BackColor>
  <!--ro, opt, object, 背景颜色-->
  <RGB>
    <!--ro, opt, int, 十进制数据表示三原色-->1
  </RGB>
</BackColor>
</ConferenceParam>
<CallBtn>
  <!--ro, opt, object, 呼叫按键窗口-->

```

```

<backPicId>
  <!--ro, opt, int, 背景图片素材ID-->1
</backPicId>
</CallBtn>
<OpenDoorBtn>
  <!--ro, opt, object, 开门按键窗口-->
  <backPicId>
    <!--ro, opt, int, 背景图片素材ID-->1
  </backPicId>
</OpenDoorBtn>
<QRCodeBtn>
  <!--ro, opt, object, 二维码按键窗口-->
  <backPicId>
    <!--ro, opt, int, 背景图片素材ID-->1
  </backPicId>
</QRCodeBtn>
<AuthenticationSuccess>
  <!--ro, opt, object, 认证成功窗口-->
  <BackColor>
    <!--ro, opt, object, 背景颜色-->
    <RGB>
      <!--ro, opt, int, 十进制数据表示三原色-->1
    </RGB>
  </BackColor>
  <backPicId>
    <!--ro, opt, int, 背景图片素材ID, desc: 背景图和背景颜色同时仅支持一种, 两者均不发, 设备优先解析背景图字段-->1
  </backPicId>
  <SubWindowList>
    <!--ro, opt, object, 子窗口列表-->
    <SubWindow>
      <!--ro, opt, object, 子窗口-->
      <subWindowType>
        <!--ro, req, enum, 子窗口类型, subType:string, [temperature#温度, facePic#人脸图片, name#姓名, employeeNo#工号, group#组织/部门, tips#提示信息]--
>temperature
      </subWindowType>
      <enabled>
        <!--ro, req, bool, 使能-->true
      </enabled>
      <fontSize>
        <!--ro, opt, int, 字体大小, unit:px, unitType:图像显示-->1
      </fontSize>
      <fontColor>
        <!--ro, opt, int, 字体颜色-->1
      </fontColor>
      <fontStyle>
        <!--ro, opt, string, 字体风格, desc:bold-加粗, italic-倾斜, underline-下划线, strikethrough-删除线-->bold, italic, underline, strikethrough
      </fontStyle>
      <Position>
        <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
        <positionX>
          <!--ro, req, int, x坐标-->1
        </positionX>
        <positionY>
          <!--ro, req, int, y坐标-->1
        </positionY>
        <height>
          <!--ro, req, int, 宽-->1
        </height>
        <width>
          <!--ro, req, int, 高-->1
        </width>
      </Position>
    </SubWindow>
  </SubWindowList>
</AuthenticationSuccess>
<AuthenticationFail>
  <!--ro, opt, object, 认证失败窗口-->
  <BackColor>
    <!--ro, opt, object, 背景颜色-->
    <RGB>
      <!--ro, opt, int, 十进制数据表示三原色-->1
    </RGB>
  </BackColor>
  <backPicId>
    <!--ro, opt, int, 背景图片素材ID, desc: 背景图和背景颜色同时仅支持一种, 两者均不发, 设备优先解析背景图字段-->1
  </backPicId>
  <SubWindowList>
    <!--ro, opt, object, 子窗口列表-->
    <SubWindow>
      <!--ro, opt, object, 子窗口-->
      <subWindowType>
        <!--ro, req, enum, 子窗口类型, subType:string, [temperature#温度, facePic#人脸图片, name#姓名, employeeNo#工号, group#组织/部门, tips#提示信息]--
>temperature
      </subWindowType>
      <enabled>
        <!--ro, req, bool, 使能-->true
      </enabled>
      <fontSize>
        <!--ro, opt, int, 字体大小, unit:px, unitType:图像显示-->1
      </fontSize>
      <fontColor>
        <!--ro, opt, int, 字体颜色-->1
      </fontColor>
      <fontStyle>

```

```

<!--ro, opt, string, 字体风格, desc:bold-加粗, italic-倾斜, underline-下划线, strikethrough-删除线-->bold,italic,underline,strikethrough
</fontStyle>
<Position>
  <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
  <positionX>
    <!--ro, req, int, x坐标-->1
  </positionX>
  <positionY>
    <!--ro, req, int, y坐标-->1
  </positionY>
  <height>
    <!--ro, req, int, 宽-->1
  </height>
  <width>
    <!--ro, req, int, 高-->1
  </width>
</Position>
<facePicID>
  <!--ro, opt, int, 默认人脸图片ID, step:1, dep:and,
{$.PageList.Page.WindowsList[*].Windows.AuthenticationFail.SubWindowList.SubWindow.subWindowType,eq,facePic)-->1
</facePicID>
</SubWindow>
</SubWindowList>
</AuthenticationFail>
<StatusBar>
  <!--ro, opt, object, 状态栏, desc:当素材类型为状态栏时有效, 且必填-->
  <backPicID>
    <!--ro, opt, int, 背景图片素材ID, desc:背景图和背景颜色同时仅支持一种,两者均下发,设备优先解析背景图字段-->1
  </backPicID>
  <statusBarType>
    <!--ro, req, enum, 状态栏类型, subType:string, [fixed#固定的], desc:当为固定类型时, 窗口位置不生效, 设备将状态栏固定在顶部显示-->fixed
  </statusBarType>
  <SubStatusList>
    <!--ro, opt, object, 子状态列表-->
    <SubStatus>
      <!--ro, opt, object, 子状态-->
      <statusType>
        <!--ro, req, enum, 子状态, subType:string, [dial#拨号状态(3G4G等),wifi#wifi状态,EZVIZ#萤石状态,wiredNetwork#有线网络状态,deploy#布防状态,temperature#测温]-->dial
      </statusType>
      <enabled>
        <!--ro, req, bool, 是否显示-->true
      </enabled>
    </SubStatus>
  </SubStatusList>
</StatusBar>
<ExamTemplate>
  <!--ro, opt, object, 考试模板, desc:当页面用于会议界面时有效-->
  <examTemplateID>
    <!--ro, opt, string, 模板ID-->test
  </examTemplateID>
  <backPicURL>
    <!--ro, opt, string, 背景图片素材URL, range:[1,1024]-->test
  </backPicURL>
  <backgroundDirection>
    <!--ro, opt, enum, 背景图方向, subType:string, [horizontal#水平,vertical#垂直]-->horizontal
  </backgroundDirection>
  <ModuleInfoList>
    <!--ro, opt, array, 界面模块信息列表, subType:object-->
    <ModuleInfo>
      <!--ro, opt, object, 界面模块信息-->
      <moduleType>
        <!--ro, opt, enum, 模块类型, subType:string, [SchoolName#学校名称,currentTime#当前时间,examinationPlanLable#考试计划标签,examinationPlanName#
考试计划名称,examRoomNumberLable#考场号标签,examRoomNumber#考场号,examRoomAddress#考试地点,examRoomAddressLable,examDescriptionLable#考试说明标
签,examDescription#考试说明,examSubjectLable#考试科目标签,examSubject#考试科目,examTimeLable#考试时间标签,examTime#考试时间,teacherLable#监考老师标
签,teacher#监考老师,teacherLable#监考老师标签,teacher#监考老师,examNumberRangeLable#考号范围,examNumberRange#考号范围]-->SchoolName
      </moduleType>
      <enabled>
        <!--ro, opt, bool, 使能-->true
      </enabled>
      <Position>
        <!--ro, opt, object, 位置, desc:左上角为坐标原点-->
        <positionX>
          <!--ro, req, int, X坐标-->1
        </positionX>
        <positionY>
          <!--ro, req, int, Y坐标-->1
        </positionY>
        <height>
          <!--ro, req, int, 高-->1
        </height>
        <width>
          <!--ro, req, int, 宽-->1
        </width>
      </Position>
      <fontSize>
        <!--ro, opt, int, 字体大小-->1
      </fontSize>
      <fontStyleList>
        <!--ro, opt, array, 字体风格列表, subType:object-->
        <fontStyle>
          <!--ro, opt, enum, 字体风格, subType:string, [bold#加粗,italic#倾斜,regular#常规,strikethrough#中划线,underline#下划线]-->bold
        </fontStyle>
      </fontStyleList>
    </ModuleInfo>
  </ModuleInfoList>

```

```
</fontStyleList>
<fontType>
  <!--ro, opt, enum, 字体类型, subType:string, [custom#自定义,default#默认,kaiTi#楷体,simHei#黑体,simSun#宋体]-->custom
</fontType>
<fontColor>
  <!--ro, opt, object, 字体颜色-->
  <RGB>
    <!--ro, opt, int, 十进制数据表示三原色-->1
  </RGB>
</fontColor>
<fontTransparent>
  <!--ro, opt, int, 字体透明度-->1
</fontTransparent>
<alignType>
  <!--ro, opt, enum, 字体对齐方式, subType:string, [left#左,right#右,middle#中]-->middle
</alignType>
<characterSpacing>
  <!--ro, opt, int, 字符间距-->1
</characterSpacing>
<lineSpacing>
  <!--ro, opt, int, 行间距-->1
</lineSpacing>
<layerNo>
  <!--ro, opt, int, 图层编号, desc: 数字越大, 表示在越上层-->1
</layerNo>
</ModuleInfo>
</ModuleInfoList>
</ExamTemplate>
<ChartData>
  <!--ro, opt, object, 图表数据-->
  <materialNo>
    <!--ro, req, int, 素材编号-->1
  </materialNo>
  <chartType>
    <!--ro, opt, enum, 图表类型, subType:string, [histogram#柱状图,pieChart#饼图]-->histogram
  </chartType>
  <backPicId>
    <!--ro, opt, int, 背景图片素材编号-->1
  </backPicId>
  <HeadDataList>
    <!--ro, opt, array, 表头数据列表, subType:object-->
    <HeadData>
      <!--ro, opt, object, 表头数据-->
      <id>
        <!--ro, req, int, 索引-->1
      </id>
      <data>
        <!--ro, req, string, 数据-->test
      </data>
      <headType>
        <!--ro, opt, enum, 表头类型, subType:string, [axis#轴,value#值]-->axis
      </headType>
    </HeadData>
  </HeadDataList>
</ChartData>
</Windows>
</WindowsList>
</Page>
```

69.63.3 获取全部页面参数

Request URL

GET /ISAPI/Publish/ProgramMgr/program/<programID>/page

查询参数

参数名称	参数类型	描述
programID	string	节目ID

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>
<PageList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, 页面列表, attr:version{req, string, 协议版本}-->
  <Page>
    <!--ro, req, object, 页面-->
    <id>
      <!--ro, req, int, 索引-->1
    </id>
    <PageBasicInfo>
```

```
<!--no, req, object, 页面基本信息-->
<pageName>
  <!--no, opt, string, 页面名称-->test
</pageName>
<BackgroundColor>
  <!--no, opt, object, 背景颜色-->
  <RGB>
    <!--no, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
  </RGB>
</BackgroundColor>
<playDurationMode>
  <!--no, opt, enum, 页面播放时间模式, subType:string, [selfDefine#自定义,auto#自动,loop#循环], desc:自定义时playDuration节点有效;自动时根据播放内容的
  时间*playCount计算:循环时表示一直循环-->selfDefine
</playDurationMode>
<playDuration>
  <!--no, opt, int, 页面播放时长, unit:s, unitType:时间-->1
</playDuration>
<playCount>
  <!--no, opt, int, 页面播放次数-->1
</playCount>
<switchDuration>
  <!--no, opt, int, 切换间隔, unit:s, unitType:时间-->1
</switchDuration>
<switchEffect>
  <!--no, opt, enum, 切换效果, subType:string, [none#无,random#随机,boxShrink#盒状收缩,boxSpread#盒状展开,cycleShrink#圆形收缩,cycSpread#圆形展
  开,eraseUp#向上擦除,eraseDown#向下擦除,eraseLeft#向左擦除,eraseRight#向右擦除,verticalShelter#垂直屏蔽,horizontalShelter#水平屏蔽,verticalChessboard#纵向棋
  盘式,horizontalChessboard#横向棋盘式,dissolve#随机溶解,LeftRightToCenter#左右向中间缩进,ceterToLeftRight#中央向左右扩展,upDownToCenter#上下向中央缩
  进,centerToUpDown#中央向上下扩展,drawOutLeftDown#从左下抽出,drawOutLeftUp#从左上抽出,drawOutRightDown#从右下抽出,drawOutRightUp#从右上抽出,verticalLine#随机
  垂直线条,horizontalLine#随机垂直线条]-->none
</switchEffect>
<backgroundPic>
  <!--no, opt, int, 背景图片素材ID-->1
</backgroundPic>
<SignInCfg>
  <!--no, opt, object, 签到信息, desc:当页面用于签到或考勤界面时有效-->
  <VoiceBroadcast>
    <!--no, opt, object, 语音播报-->
    <enabled>
      <!--no, opt, bool, 是否启用语音播报-->true
    </enabled>
    <broadcastSpeed>
      <!--no, opt, enum, 播报语速, subType:string, [slower#慢,slow#较慢,medium#中速,fast#较快,faster#快]-->slower
    </broadcastSpeed>
    <inLibVoice>
      <!--no, opt, string, 在库人员正常体温语音播报内容-->test
    </inLibVoice>
    <outLibVoice>
      <!--no, opt, string, 未知人员正常体温语音播报内容-->test
    </outLibVoice>
    <inLibAbnormalTemperatureVoice>
      <!--no, opt, string, 在库人员异常温度语音播报内容-->test
    </inLibAbnormalTemperatureVoice>
    <outLibAbnormalTemperatureVoice>
      <!--no, opt, string, 未知人员异常温度语音播报内容-->test
    </outLibAbnormalTemperatureVoice>
  </VoiceBroadcast>
  <WelcomeWord>
    <!--no, opt, object, 欢迎词-->
    <enabled>
      <!--no, opt, bool, 是否启用欢迎词-->true
    </enabled>
    <inLibWord>
      <!--no, opt, string, 在库人员欢迎词内容-->test
    </inLibWord>
    <outLibWord>
      <!--no, opt, string, 未知人员欢迎词内容-->test
    </outLibWord>
  </WelcomeWord>
  <SignCartoon>
    <!--no, opt, object, 签到动画-->
    <signCartoonSpeed>
      <!--no, opt, enum, 动画速度, subType:string, [fast#快,medium#终端,slow#慢]-->fast
    </signCartoonSpeed>
  </SignCartoon>
  <backgroundSignEnabled>
    <!--no, opt, bool, 是否启用后台签到-->true
  </backgroundSignEnabled>
  <TemperatureDetection>
    <!--no, opt, object, 温度监测参数-->
    <enabled>
      <!--no, req, bool, 是否启用-->true
    </enabled>
    <threshold>
      <!--no, req, float, 正常温度值-->0.000
    </threshold>
  </TemperatureDetection>
</SignInCfg>
</PageBasicInfo>
<characterMode>
  <!--no, opt, enum, 页面中的欢迎词模式, subType:string, [mode1#模式1,mode2#模式2,mode3#模式3], desc:门禁增加,门禁欢迎词节目位置固定,且存在三种模式--
  >mode1
</characterMode>
<WindowsList>
```

```
<!--ro, opt, array, 窗口信息列表, subType:object-->
<windows>
  <!--ro, opt, object, 窗口信息-->
  <id>
    <!--ro, req, int, 内容编号-->1
  </id>
  <Position>
    <!--ro, opt, object, 内容位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--ro, req, int, x坐标-->1
    </positionX>
    <positionY>
      <!--ro, req, int, y坐标-->1
    </positionY>
    <height>
      <!--ro, req, int, 宽-->1
    </height>
    <width>
      <!--ro, req, int, 高-->1
    </width>
  </Position>
  <CornerRadius>
    <!--ro, opt, object, 圆角配置, desc:目前仅对picture图片窗口和character文字窗口有效-->
    <TopLeft>
      <!--ro, opt, int, 左上角半径-->1
    </TopLeft>
    <TopRight>
      <!--ro, opt, int, 右上角半径-->1
    </TopRight>
    <BottomLeft>
      <!--ro, opt, int, 左下角半径-->1
    </BottomLeft>
    <BottomRight>
      <!--ro, opt, int, 右下角半径-->1
    </BottomRight>
  </CornerRadius>
  <layerNo>
    <!--ro, req, int, 图层编号-->1
  </layerNo>
  <WinMaterialInfo>
    <!--ro, opt, object, 窗口素材信息-->
    <materialType>
      <!--ro, req, enum, 素材类型, subType:string, [static#静态,dynamic#动态,other#其他]-->static
    </materialType>
    <staticMaterialType>
      <!--ro, opt, enum, 静态素材类型, subType:string, [picture#图片,flash#flash,audio#音频,video#视频,document#文本,ppt#ppt,doc#文档,excel#表格,pdf#pdf,web#网页,app#app,signIn#签到,conference#会议]-->picture
    </staticMaterialType>
    <dynamicType>
      <!--ro, opt, enum, 动态素材类型, subType:string,
[web#web,socket#socket,rss#rss,realStream#realStream,call#call,dynamicPic#dynamicPic,capturePic#capturePic,character#character,chart#图表数据]-->web
    </dynamicType>
    <otherType>
      <!--ro, opt, enum, 其他素材类型, subType:string, [clock#时钟,weather#天气,countdown#倒计时,localInput#本地输入,hyperLinkBtn#超链接按钮,event#事件,callBtn#呼叫按钮,openDoorBtn#开门按钮,QRCodeBtn#二维码按钮,authenticationSuccess#认证成功,authenticationFail#认证失败,statusBar#状态栏,ExamTemplate#考试模板]-->clock
    </otherType>
  </WinMaterialInfo>
  <TouchProperty>
    <!--ro, opt, object, 触摸属性-->
    <windType>
      <!--ro, opt, enum, 窗口类型, subType:string, [popup#弹窗,page#页面]-->popup
    </windType>
    <hyperLinkType>
      <!--ro, opt, enum, 超链接类型, subType:string, [window#页面超链接,page#窗口超链接], desc:窗口类型为弹窗时有效-->window
    </hyperLinkType>
    <windowId>
      <!--ro, opt, int, 窗口编号, desc:当前页面的窗口,超链接类型为窗口时有效-->1
    </windowId>
    <pageId>
      <!--ro, opt, int, 页面编号, desc:超链接类型为页面时有效-->1
    </pageId>
  </TouchProperty>
  <PlayItemList>
    <!--ro, opt, array, 窗口播放列表, subType:object-->
    <PlayItem>
      <!--ro, req, object, 窗口播放信息-->
      <id>
        <!--ro, req, int, 播放序号-->1
      </id>
      <materialNo>
        <!--ro, opt, int, 素材编号-->1
      </materialNo>
      <inputChannel>
        <!--ro, opt, int, 绑定的IPC通道号-->1
      </inputChannel>
      <playEffect>
        <!--ro, req, enum, 播放效果, subType:string, [none#无,marquee#跑马灯]-->none
      </playEffect>
      <MarqueeInfo>
        <!--ro, opt, object, 跑马灯信息-->
        <scrollType>
          <!--ro, req, enum, 跑马灯滚动类型, subType:string, [none#不滚动,Loops#循环滚动,once#滚动一次,backAndForth#来回滚动]-->none
        </scrollType>
      </MarqueeInfo>
    </PlayItem>
  </PlayItemList>
</windows>
```

```
</scrollType>
<scrollDeriction>
  <!--ro, req, enum, 跑马灯滚动方向, subType:string, [none#无,up#向上,down#向下,left#向左,right#向右]-->none
</scrollDeriction>
<scrollSpeed>
  <!--ro, req, int, 跑马灯滚动速度-->1
</scrollSpeed>
</MarqueeInfo>
<PlayDuration>
  <!--ro, opt, object, 窗口内素材播放时间, desc:静态素材可以配置该时间,视频直播和IPC通道都可配置该节点-->
  <durationType>
    <!--ro, req, enum, 持续方式, subType:string, [materialTime#按素材时间,selfDefine#自定义,unlimited#无限时长]-->selfDefine
  </durationType>
  <duration>
    <!--ro, req, int, 持续时间, unit:s, unitType:时间-->1
  </duration>
</PlayDuration>
<CharactersEffect>
  <!--ro, opt, object, 文字显示效果, desc:当素材类型为文本txt和时有效-->
  <fontSize>
    <!--ro, req, int, 字体大小-->1
  </fontSize>
  <fontColor>
    <!--ro, req, object, 字体颜色-->
    <RGB>
      <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
    </RGB>
  </fontColor>
  <backgroundType>
    <!--ro, opt, enum, 背景类型, subType:string, [color#颜色(默认),picture#图片], desc:字段不存在时,默认为颜色-->color
  </backgroundType>
  <backPicId>
    <!--ro, opt, int, 背景图片素材ID-->1
  </backPicId>
  <BackColor>
    <!--ro, req, object, 背景颜色-->
    <RGB>
      <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
    </RGB>
  </BackColor>
  <backTransparent>
    <!--ro, req, int, 背景透明度-->1
  </backTransparent>
  <subtitlesEnabled>
    <!--ro, req, bool, 子标题使能-->true
  </subtitlesEnabled>
  <scrollDirection>
    <!--ro, req, enum, 文字滚动方向, subType:string, [left#向左,right#向右,up#向上,down#向下]-->left
  </scrollDirection>
  <scrollSpeed>
    <!--ro, req, int, 文字滚动速度-->1
  </scrollSpeed>
</CharactersEffect>
<switchEffect>
  <!--ro, opt, enum, 窗口素材切换效果, subType:string, [none#无,leftInRightOut#左进右出,rightInLeftOut#右进左出,bottomInTopOut#下进上出,topInBottomOut#上进下出,fadeInFadeOut#淡入淡出,middleExit#中间退出,topPop#顶部弹出,rightBottomIn#右下角进入,leftTopIn#左上角进入,horizontalOpen#水平打开,verticalOpen#垂直打开,random#随机效果], desc:图片素材时有效-->random
</switchEffect>
  <pageTime>
    <!--ro, opt, int, 翻页时间, unit:s, unitType:时间, desc:当素材为word,ppt,pdf,excel时有效-->1
  </pageTime>
  <scrollSpeed>
    <!--ro, req, int, 滚动速度, desc:当素材为静态web时有效-->1
  </scrollSpeed>
  <scaleType>
    <!--ro, opt, enum, 素材缩放参数配置, subType:string, [fitXY#拉伸填充,fitCenter#自适应,centerCrop#截取填充], desc:目前仅当图片窗口有效-->centerCrop
  </scaleType>
  <playType>
    <!--ro, opt, enum, 播放方式, subType:string, [normal#正常,faceAttendance#考勤,PeopleCounting#客流,temperature#测温], desc:目前仅当图片窗口有效-->normal
  </playType>
  <CharactersAttribute>
    <!--ro, opt, object, 文字属性, desc:当dynamicType为character时有效-->
    <fontSize>
      <!--ro, req, int, 字体大小-->1
    </fontSize>
    <fontColor>
      <!--ro, req, object, 字体颜色-->
      <RGB>
        <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
      </RGB>
    </fontColor>
    <backgroundType>
      <!--ro, req, enum, 背景类型, subType:string, [color#颜色,picture#图片]-->color
    </backgroundType>
    <BackColor>
      <!--ro, req, object, 背景色-->
      <RGB>
        <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
      </RGB>
    </BackColor>
    <backTransparent>
      <!--ro, opt, int, 背景透明度-->1
```

```

</backTransparent>
<alignType>
  <!--ro, req, enum, 对齐方式, subType:string, [left#左,right#右,middle#中]-->middle
</alignType>
<verticalAlignType>
  <!--ro, req, enum, 垂直对齐方式, subType:string, [top#顶,bottom#底,verticalCenter#中]-->top
</verticalAlignType>
<characterContent>
  <!--ro, opt, string, 文字内容, range:[0,512]-->test
</characterContent>
<backPicId>
  <!--ro, opt, int, 背景图片素材ID-->1
</backPicId>
<characterStyle>
  <!--ro, req, enum, 字体效果, subType:string, [normal#常规,bold#粗体]-->normal
</characterStyle>
<characterType>
  <!--ro, req, enum, 字体类型, subType:string, [mainTitle#主标题,subTitle#副标题]-->mainTitle
</characterType>
</CharactersAttribute>
<WebProperty>
  <!--ro, opt, object, 网页属性-->
  <webType>
    <!--ro, opt, enum, 网页类型, subType:string, [desktop#桌面版,mobile#移动版], desc:默认按照设备的系统类型（Windows或Android）来显示网页--
>desktop
  </webType>
</WebProperty>
</PlayItem>
</PlayItemList>
<enabledAudio>
  <!--ro, opt, bool, 音频使能-->true
</enabledAudio>
<enableHide>
  <!--ro, opt, bool, 隐藏使能-->true
</enableHide>
<enableLock>
  <!--ro, opt, bool, 锁定使能-->true
</enableLock>
<AppWindow>
  <!--ro, opt, object, App窗口信息-->
  <WindowInfoList>
    <!--ro, req, array, 窗口信息列表, subType:object-->
    <WindowInfo>
      <!--ro, req, object, 窗口信息-->
      <id>
        <!--ro, req, int, 索引-->1
      </id>
      <materialNo>
        <!--ro, req, int, 素材编号, desc:采用本字段下发时, 用户需要先将App安装包当做素材存储到设备中, 得到素材编号再进行下发-->1
      </materialNo>
      <path>
        <!--ro, opt, string, App路径, range:[0,1024],
desc:path和materialNo字段同时存在时, 设备优先解析path字段;
采用本字段下发时, 设备需要确保App已经安装在设备内, 直接传入App的路径即可;
对于Android设备, path传的是App的包名; 对于Windows设备, path传的是App的应用程序绝对路径; -->test
      </path>
    </WindowInfo>
  </WindowInfoList>
</AppWindow>
<DataSource>
  <!--ro, opt, object, 数据来源, desc:当窗口为叫号和动态弹图时有效-->
  <materialNo>
    <!--ro, req, int, 素材编号-->1
  </materialNo>
  <dataSourceUUID>
    <!--ro, opt, string, 数据源UUID-->1
  </dataSourceUUID>
</DataSource>
<Call>
  <!--ro, opt, object, 叫号数据, desc:对应表格数据-->
  <materialNo>
    <!--ro, req, int, 素材编号-->1
  </materialNo>
  <tableRow>
    <!--ro, req, int, 表格行-->1
  </tableRow>
  <tableColumn>
    <!--ro, req, int, 表格列-->1
  </tableColumn>
  <tableDirection>
    <!--ro, req, enum, 表格方向, subType:string, [vertical#横向,horizontal#纵向]-->vertical
  </tableDirection>
  <tableType>
    <!--ro, req, enum, 表格类型, subType:string, [template1#模板1,template2#模板2,template3#模板3,template4#模板4,template5#模板5,template6#模板6]--
>template1
  </tableType>
  <backPicId>
    <!--ro, opt, int, 背景图片素材编号-->1
  </backPicId>
  <alignType>
    <!--ro, req, enum, 对齐方式, subType:string, [left#左,right#右,middle#中]-->left
  </alignType>
  <refreshDirection>
    <!--ro, req, enum, 刷新方向, subType:string, [vertical#纵向,horizontal#横向]-->vertical
  </refreshDirection>

```

```
<!--ro, req, enum, 刷新方向, subType:string, [up1oown#从上到下,down1oup#从下到上,left1oright#从左到右,right1oleft#从右到左]-->up1oown
</refreshDirection>
<HeadDataList>
<!--ro, opt, array, 表头数据列表, subType:object-->
<HeadData>
<!--ro, opt, object, 表头数据-->
<id>
<!--ro, req, int, 索引-->1
</id>
<data>
<!--ro, req, string, 数据-->test
</data>
</HeadData>
</HeadDataList>
<ItemStyleList>
<!--ro, opt, array, 表格列或行样式列表, subType:object-->
<ItemStyle>
<!--ro, opt, object, 表格列或行样式-->
<id>
<!--ro, req, int, 索引-->1
</id>
<width>
<!--ro, req, int, 每列宽度所占百分比-->1
</width>
<fontSize>
<!--ro, req, int, 字体大小-->1
</fontSize>
<FontColor>
<!--ro, req, object, 字体颜色-->
<RGB>
<!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
</RGB>
</FontColor>
<BackColor>
<!--ro, req, object, 背景色-->
<RGB>
<!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
</RGB>
</BackColor>
</ItemStyle>
</ItemStyleList>
<playType>
<!--ro, opt, enum, 数据播放方式, subType:string, [roll#滚动播放,flipOver#翻页播放]-->roll
</playType>
</Call>
<DynamicPic>
<!--ro, opt, object, 动态弹图窗口配置-->
<materialNo>
<!--ro, req, int, 素材编号-->1
</materialNo>
<backPicId>
<!--ro, opt, int, 背景图片素材ID-->1
</backPicId>
</DynamicPic>
<CapturePic>
<!--ro, opt, object, 抓拍图片参数-->
<backPicId>
<!--ro, opt, int, 背景图片素材ID-->1
</backPicId>
<ipcMaterialNo>
<!--ro, req, int, IPC素材编号-->1
</ipcMaterialNo>
<cancelType>
<!--ro, req, enum, 取消方式, subType:string, [auto#自动>manual#手动]-->>manual
</cancelType>
<duration>
<!--ro, opt, int, 持续时间, unit:s, unitType:时间-->1
</duration>
</CapturePic>
<ClockParam>
<!--ro, opt, object, 时钟参数-->
<backPicId>
<!--ro, opt, int, 背景图片素材ID-->1
</backPicId>
<ClockIcon>
<!--ro, req, object, 时钟图标参数-->
<enabled>
<!--ro, req, bool, 使能-->true
</enabled>
<type>
<!--ro, opt, enum, 时钟类型, subType:string, [clock1#clock1,clock2#clock2]-->clock1
</type>
<Position>
<!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
<positionX>
<!--ro, req, int, x坐标-->1
</positionX>
<positionY>
<!--ro, req, int, y坐标-->1
</positionY>
<height>
<!--ro, req, int, 宽-->1
</height>
<width>
```

```
<!--ro, req, int, 高-->1
</width>
</Position>
</ClockIcon>
<YmdParam>
<!--ro, req, object, 时钟年月日参数-->
<enabled>
<!--ro, req, bool, 使能-->true
</enabled>
<fontSize>
<!--ro, req, int, 字体大小-->1
</fontSize>
<FontColor>
<!--ro, req, object, 字体颜色-->
<RGB>
<!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
</RGB>
</FontColor>
<BackColor>
<!--ro, req, object, 背景色-->
<RGB>
<!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
</RGB>
</BackColor>
<Position>
<!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
<positionX>
<!--ro, req, int, x坐标-->1
</positionX>
<positionY>
<!--ro, req, int, y坐标-->1
</positionY>
<height>
<!--ro, req, int, 宽-->1
</height>
<width>
<!--ro, req, int, 高-->1
</width>
</Position>
</YmdParam>
<HmsParam>
<!--ro, req, object, 时钟时分秒参数-->
<enabled>
<!--ro, req, bool, 使能-->true
</enabled>
<fontSize>
<!--ro, req, int, 字体大小-->1
</fontSize>
<FontColor>
<!--ro, req, object, 字体颜色-->
<RGB>
<!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
</RGB>
</FontColor>
<BackColor>
<!--ro, req, object, 背景色-->
<RGB>
<!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
</RGB>
</BackColor>
<Position>
<!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
<positionX>
<!--ro, req, int, x坐标-->1
</positionX>
<positionY>
<!--ro, req, int, y坐标-->1
</positionY>
<height>
<!--ro, req, int, 宽-->1
</height>
<width>
<!--ro, req, int, 高-->1
</width>
</Position>
</HmsParam>
<WeekParam>
<!--ro, req, object, 时钟星期参数-->
<enabled>
<!--ro, req, bool, 使能-->true
</enabled>
<fontSize>
<!--ro, req, int, 字体大小-->1
</fontSize>
<FontColor>
<!--ro, req, object, 字体颜色-->
<RGB>
<!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
</RGB>
</FontColor>
<BackColor>
<!--ro, req, object, 背景色-->
<RGB>
```

```
<!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
</RGB>
</BackColor>
<Position>
<!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
<positionX>
<!--ro, req, int, x坐标-->1
</positionX>
<positionY>
<!--ro, req, int, y坐标-->1
</positionY>
<height>
<!--ro, req, int, 宽-->1
</height>
<width>
<!--ro, req, int, 高-->1
</width>
</Position>
</WeekParam>
</ClockParam>
<WeatherParam>
<!--ro, opt, object, 天气参数-->
<backPicId>
<!--ro, opt, int, 天气背景图片素材id-->1
</backPicId>
<WeatherIcon>
<!--ro, opt, object, 天气图标参数-->
<enabled>
<!--ro, req, bool, 使能-->true
</enabled>
<Position>
<!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
<positionX>
<!--ro, req, int, x坐标-->1
</positionX>
<positionY>
<!--ro, req, int, y坐标-->1
</positionY>
<height>
<!--ro, req, int, 宽-->1
</height>
<width>
<!--ro, req, int, 高-->1
</width>
</Position>
</WeatherIcon>
<Date>
<!--ro, opt, object, 天气日期参数-->
<enabled>
<!--ro, req, bool, 使能-->true
</enabled>
<fontSize>
<!--ro, req, int, 字体大小-->1
</fontSize>
<FontColor>
<!--ro, req, object, 字体颜色-->
<RGB>
<!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
</RGB>
</FontColor>
<BackColor>
<!--ro, req, object, 背景色-->
<RGB>
<!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
</RGB>
</BackColor>
<Position>
<!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
<positionX>
<!--ro, req, int, x坐标-->1
</positionX>
<positionY>
<!--ro, req, int, y坐标-->1
</positionY>
<height>
<!--ro, req, int, 宽-->1
</height>
<width>
<!--ro, req, int, 高-->1
</width>
</Position>
</Date>
<Temperature>
<!--ro, opt, object, 天气温度参数-->
<enabled>
<!--ro, req, bool, 使能-->true
</enabled>
<fontSize>
<!--ro, req, int, 字体大小-->1
</fontSize>
<FontColor>
<!--ro, req, object, 字体颜色-->
<RGB>
<!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
```

```
</RGB>
</FontColor>
<BackColor>
  <!--ro, req, object, 背景色-->
  <RGB>
    <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFFFF-->1
  </RGB>
</BackColor>
<Position>
  <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
  <positionX>
    <!--ro, req, int, x坐标-->1
  </positionX>
  <positionY>
    <!--ro, req, int, y坐标-->1
  </positionY>
  <height>
    <!--ro, req, int, 宽-->1
  </height>
  <width>
    <!--ro, req, int, 高-->1
  </width>
</Position>
</Temperature>
<WeatherContent>
  <!--ro, opt, object, 天气情况参数-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <fontSize>
    <!--ro, req, int, 字体大小-->1
  </fontSize>
  <FontColor>
    <!--ro, req, object, 字体颜色-->
    <RGB>
      <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFFFF-->1
    </RGB>
  </FontColor>
  <BackColor>
    <!--ro, req, object, 背景色-->
    <RGB>
      <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFFFF-->1
    </RGB>
  </BackColor>
  <Position>
    <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--ro, req, int, x坐标-->1
    </positionX>
    <positionY>
      <!--ro, req, int, y坐标-->1
    </positionY>
    <height>
      <!--ro, req, int, 宽-->1
    </height>
    <width>
      <!--ro, req, int, 高-->1
    </width>
  </Position>
</WeatherContent>
<City>
  <!--ro, opt, object, 天气城市参数-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <cityId>
    <!--ro, req, string, 城市编号-->test
  </cityId>
  <cityName>
    <!--ro, req, string, 城市名称-->test
  </cityName>
  <fontSize>
    <!--ro, req, int, 字体大小-->1
  </fontSize>
  <FontColor>
    <!--ro, req, object, 字体颜色-->
    <RGB>
      <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFFFF-->1
    </RGB>
  </FontColor>
  <BackColor>
    <!--ro, req, object, 背景色-->
    <RGB>
      <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFFFF-->1
    </RGB>
  </BackColor>
  <Position>
    <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--ro, req, int, x坐标-->1
    </positionX>
    <positionY>
      <!--ro, req, int, y坐标-->1
    </positionY>
```

```
</positionY>
<height>
  <!--ro, req, int, 宽-->1
</height>
<width>
  <!--ro, req, int, 高-->1
</width>
</Position>
</City>
<Humidity>
  <!--ro, opt, object, 天气湿度参数-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <fontSize>
    <!--ro, req, int, 字体大小-->1
  </fontSize>
  <FontColor>
    <!--ro, req, object, 字体颜色-->
    <RGB>
      <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
    </RGB>
  </FontColor>
  <BackColor>
    <!--ro, req, object, 背景色-->
    <RGB>
      <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
    </RGB>
  </BackColor>
  <Position>
    <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--ro, req, int, x坐标-->1
    </positionX>
    <positionY>
      <!--ro, req, int, y坐标-->1
    </positionY>
    <height>
      <!--ro, req, int, 宽-->1
    </height>
    <width>
      <!--ro, req, int, 高-->1
    </width>
  </Position>
</Humidity>
<AirQuality>
  <!--ro, opt, object, 天气空气质量参数-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <fontSize>
    <!--ro, req, int, 字体大小-->1
  </fontSize>
  <FontColor>
    <!--ro, req, object, 字体颜色-->
    <RGB>
      <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
    </RGB>
  </FontColor>
  <BackColor>
    <!--ro, req, object, 背景色-->
    <RGB>
      <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
    </RGB>
  </BackColor>
  <Position>
    <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--ro, req, int, x坐标-->1
    </positionX>
    <positionY>
      <!--ro, req, int, y坐标-->1
    </positionY>
    <height>
      <!--ro, req, int, 宽-->1
    </height>
    <width>
      <!--ro, req, int, 高-->1
    </width>
  </Position>
</AirQuality>
<UpdateTime>
  <!--ro, opt, object, 天气更新时间参数-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <refreshTime>
    <!--ro, req, string, 刷新时间, desc:设备早期返回时间格式有误, 错为: 20200826T051500+08, 为保证XSD校验通过, 改为string类型-->20200826T051500+08
  </refreshTime>
  <updateInterval>
    <!--ro, req, int, 刷新闻隔, unit:minute-->1
  </updateInterval>
  <fontSize>
    <!--ro, req, int, 字体大小-->1
```

```
<!--ro, req, int, 字体大小-->1
</fontSize>
<FontColor>
<!--ro, req, object, 字体颜色-->
<RGB>
<!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
</RGB>
</FontColor>
<BackColor>
<!--ro, req, object, 背景色-->
<RGB>
<!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
</RGB>
</BackColor>
<Position>
<!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
<positionX>
<!--ro, req, int, x坐标-->1
</positionX>
<positionY>
<!--ro, req, int, y坐标-->1
</positionY>
<height>
<!--ro, req, int, 宽-->1
</height>
<width>
<!--ro, req, int, 高-->1
</width>
</Position>
</UpdateTime>
<Wind>
<!--ro, opt, object, 天气风力参数-->
<enabled>
<!--ro, req, bool, 使能-->true
</enabled>
<fontSize>
<!--ro, req, int, 字体大小-->1
</fontSize>
<FontColor>
<!--ro, req, object, 字体颜色-->
<RGB>
<!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
</RGB>
</FontColor>
<BackColor>
<!--ro, req, object, 背景色-->
<RGB>
<!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
</RGB>
</BackColor>
<Position>
<!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
<positionX>
<!--ro, req, int, x坐标-->1
</positionX>
<positionY>
<!--ro, req, int, y坐标-->1
</positionY>
<height>
<!--ro, req, int, 宽-->1
</height>
<width>
<!--ro, req, int, 高-->1
</width>
</Position>
</Wind>
</WeatherParam>
<Countdown>
<!--ro, opt, object, 倒计时素材-->
<endTime>
<!--ro, req, string, 倒计时时间点, desc:设备早期返回时间格式有误, 错为: 20200826T051500+08, 为保证XSD校验通过, 改为string类型-->20200826T051500+08
</endTime>
<template>
<!--ro, req, enum, 模板类型, subType:string, [template1#模板1,template2#模板2,openingActivity1#活动开幕1,openingActivity2#活动开幕2,commercialActivity1#商业活动1,commercialActivity2#商业活动2,colLegeEntranceExam#高考,overseasTimeFormat#海外时间格式]-->template1
</template>
<timeUnit>
<!--ro, req, enum, 时间单位, subType:string, [year#年,month#月,day#日,week#周,hour#时,minute#分,second#秒]-->year
</timeUnit>
<backPicId>
<!--ro, opt, int, 背景图片素材ID-->1
</backPicId>
<TimeFontCfg>
<!--ro, opt, object, 时间字体参数-->
<fontSize>
<!--ro, req, int, 字体大小-->1
</fontSize>
<FontColor>
<!--ro, req, object, 字体颜色-->
<RGB>
<!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
</RGB>
</FontColor>
```

```
<Position>
  <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
  <positionX>
    <!--ro, req, int, x坐标-->1
  </positionX>
  <positionY>
    <!--ro, req, int, y坐标-->1
  </positionY>
  <height>
    <!--ro, req, int, 宽-->1
  </height>
  <width>
    <!--ro, req, int, 高-->1
  </width>
</Position>
</TimeFontCfg>
<timeUnitEnabled>
  <!--ro, opt, bool, 时间单位是否有效-->true
</timeUnitEnabled>
<maxTimeUnit>
  <!--ro, req, enum, 最大时间单位, subType:string, [year#年,month#月,day#日,week#周,hour#时,minute#分,second#秒,week#周], desc:表示当前倒计时的最大显示单元,例如:maxTimeUnit为月,则1年3个月,对应月份值为15-->year
</maxTimeUnit>
<ModuleInfoList>
  <!--ro, opt, array, 界面模块信息列表, subType:object-->
  <ModuleInfo>
    <!--ro, opt, object, 界面模块信息-->
    <moduleType>
      <!--ro, req, enum, 模块类型, subType:string, [year#年,month#月,day#日,hour#时,minute#分,second#秒,week#周,text#文字], desc:text为时间旁边的文字,例如年下面的文字"年"-->second
    </moduleType>
    <enabled>
      <!--ro, req, bool, 使能-->true
    </enabled>
    <fontSize>
      <!--ro, req, int, 字体大小-->1
    </fontSize>
    <fontColor>
      <!--ro, req, object, 字体颜色-->
      <RGB>
        <!--ro, opt, int, 十进制数据表示三原色-->1
      </RGB>
    </FontColor>
    <text>
      <!--ro, opt, string, 文字, desc:当moduleType为text时有效-->test
    </text>
    <Position>
      <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
      <positionX>
        <!--ro, req, int, X坐标-->1
      </positionX>
      <positionY>
        <!--ro, req, int, Y坐标-->1
      </positionY>
      <height>
        <!--ro, req, int, 宽-->1
      </height>
      <width>
        <!--ro, req, int, 高-->1
      </width>
    </Position>
  </ModuleInfo>
</ModuleInfoList>
</Countdown>
<localInputNo>
  <!--ro, req, enum, 本地输入口类型, subType:string, [VGA#VGA,HDMI#HDMI]-->VGA
</localInputNo>
<HyperlinkBtn>
  <!--ro, opt, object, 超链接按钮-->
  <backPicId>
    <!--ro, opt, int, 背景图片素材ID-->1
  </backPicId>
</HyperlinkBtn>
<CharactersAttribute>
  <!--ro, opt, object, 文字属性, desc:当dynamicType为character时有效-->
  <fontSize>
    <!--ro, req, int, 字体大小-->1
  </fontSize>
  <fontColor>
    <!--ro, req, object, 字体颜色-->
    <RGB>
      <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
    </RGB>
  </FontColor>
  <fontType>
    <!--ro, opt, enum, 字体类型, subType:string, [normal#常规,bold#粗体]-->normal
  </fontType>
  <backgroundType>
    <!--ro, req, enum, 背景类型, subType:string, [color#颜色,picture#图片]-->color
  </backgroundType>
  <BackColor>
    <!--ro, req, object, 背景色-->
    <RGB>
```

```
<!--ro, opt, int, 码值, desc:十进制数据表示二原巴,如10111111表示0x7FFFFFFF-->1
</RGB>
<fillEnabled>
  <!--ro, opt, bool, 背景色填充使能, desc:默认为true, 表示背景色填充-->true
</fillEnabled>
</BackColor>
<backTransparent>
  <!--ro, opt, int, 背景透明度-->1
</backTransparent>
<alignType>
  <!--ro, req, enum, 对齐方式, subType:string, [Left#左,right#右,middle#中]-->middle
</alignType>
<verticalAlignType>
  <!--ro, req, enum, 垂直对齐方式, subType:string, [top#顶,bottom#底,verticalCenter#中]-->top
</verticalAlignType>
<characterContent>
  <!--ro, opt, string, 文字内容, range:[0,512]-->test
</characterContent>
<backPicId>
  <!--ro, opt, int, 背景图片素材ID-->1
</backPicId>
<characterStyle>
  <!--ro, req, enum, 字体效果, subType:string, [normal#常规,bold#粗体]-->normal
</characterStyle>
<characterType>
  <!--ro, req, enum, 字体类型, subType:string, [mainTitle#主标题,subTitle#副标题]-->mainTitle
</characterType>
<scrollDirection>
  <!--ro, opt, enum, 文字滚动方向, subType:string, [Left#向左,right#向右,up#向上,down#向下,turnPage#翻页]-->left
</scrollDirection>
<scrollSpeed>
  <!--ro, opt, int, 文字滚动速度, range:[1,10]-->1
</scrollSpeed>
</CharactersAttribute>
<SignIn>
  <!--ro, opt, object, 人员签到展示窗口, desc:当staticMaterialType为signIn时有效-->
  <backPicId>
    <!--ro, opt, int, 背景图片素材ID-->1
  </backPicId>
  <ShowInfo>
    <!--ro, opt, object, 展示信息-->
    <row>
      <!--ro, opt, int, 窗口内人员签到展示行数-->1
    </row>
    <column>
      <!--ro, opt, int, 窗口内人员签到展示列数-->1
    </column>
    <Position>
      <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
      <positionX>
        <!--ro, req, int, x坐标-->1
      </positionX>
      <positionY>
        <!--ro, req, int, y坐标-->1
      </positionY>
      <height>
        <!--ro, req, int, 宽-->1
      </height>
      <width>
        <!--ro, req, int, 高-->1
      </width>
    </Position>
    <inLib>
      <!--ro, opt, object, 在库信息-->
      <nameEnabled>
        <!--ro, opt, bool, 在库人员是否展示姓名-->true
      </nameEnabled>
      <genderEnabled>
        <!--ro, opt, bool, 在库人员是否展示性别-->true
      </genderEnabled>
      <selfDefine1Enabled>
        <!--ro, opt, bool, 在库人员是否展示自定义字段1-->true
      </selfDefine1Enabled>
      <selfDefine2Enabled>
        <!--ro, opt, bool, 在库人员是否展示自定义字段2-->true
      </selfDefine2Enabled>
      <signInTimeEnabled>
        <!--ro, opt, bool, 在库人员是否展示签到时间-->true
      </signInTimeEnabled>
      <temperatureEnabled>
        <!--ro, opt, bool, 在库人员是否展示温度-->true
      </temperatureEnabled>
    </inLib>
    <outLib>
      <!--ro, opt, object, 不在库信息-->
      <nameEnabled>
        <!--ro, opt, bool, 未知人员是否展示姓名-->true
      </nameEnabled>
      <name>
        <!--ro, opt, string, 未知人员姓名-->test
      </name>
      <genderEnabled>
        <!--ro, opt, bool, 未知人员是否展示性别-->true
      </genderEnabled>
```

```
-
<signInTimeEnabled>
  <!--ro, opt, bool, 未知人员是否展示签到时间-->true
</signInTimeEnabled>
<temperatureEnabled>
  <!--ro, opt, bool, 未知人员是否展示温度-->true
</temperatureEnabled>
</outLib>
<showDuration>
  <!--ro, req, enum, 签到记录显示时间, subType:string, [always#不消失,10s#10s,20s#20s,30s#30s,no#不显示]-->always
</showDuration>
</ShowInfo>
<MainTitle>
  <!--ro, opt, object, 主标题-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <fontSize>
    <!--ro, req, int, 字体大小-->1
  </fontSize>
  <fontColor>
    <!--ro, req, object, 字体颜色-->
    <RGB>
      <!--ro, opt, int, 十进制数据表示三原色-->1
    </RGB>
  </FontColor>
  <Position>
    <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--ro, req, int, x坐标-->1
    </positionX>
    <positionY>
      <!--ro, req, int, y坐标-->1
    </positionY>
    <height>
      <!--ro, req, int, 宽-->1
    </height>
    <width>
      <!--ro, req, int, 高-->1
    </width>
  </Position>
  <text>
    <!--ro, opt, string, 标题文字-->test
  </text>
</MainTitle>
<SubTitle>
  <!--ro, opt, object, 副标题-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <fontSize>
    <!--ro, req, int, 字体大小-->1
  </fontSize>
  <fontColor>
    <!--ro, req, object, 字体颜色-->
    <RGB>
      <!--ro, opt, int, 十进制数据表示三原色-->1
    </RGB>
  </FontColor>
  <Position>
    <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--ro, req, int, x坐标-->1
    </positionX>
    <positionY>
      <!--ro, req, int, y坐标-->1
    </positionY>
    <height>
      <!--ro, req, int, 宽-->1
    </height>
    <width>
      <!--ro, req, int, 高-->1
    </width>
  </Position>
  <text>
    <!--ro, opt, string, 标题文字-->test
  </text>
</SubTitle>
<NormalTemperature>
  <!--ro, opt, object, 正常体温信息-->
  <PeopleNumber>
    <!--ro, opt, object, 正常体温人数信息-->
    <enabled>
      <!--ro, req, bool, 使能-->true
    </enabled>
    <fontSize>
      <!--ro, req, int, 字体大小-->1
    </fontSize>
    <fontColor>
      <!--ro, req, object, 字体颜色-->
      <RGB>
        <!--ro, opt, int, 十进制数据表示三原色-->1
      </RGB>
    </FontColor>
```

```
<Position>
<!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
<positionX>
<!--ro, req, int, X坐标-->1
</positionX>
<positionY>
<!--ro, req, int, Y坐标-->1
</positionY>
<height>
<!--ro, req, int, 宽-->1
</height>
<width>
<!--ro, req, int, 高-->1
</width>
</Position>
<text>
<!--ro, opt, string, 标题文字-->test
</text>
</PeopleNumber>
<Icon>
<!--ro, opt, object, 正常体温图标-->
<enabled>
<!--ro, req, bool, 使能-->true
</enabled>
<Position>
<!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
<positionX>
<!--ro, req, int, X坐标-->1
</positionX>
<positionY>
<!--ro, req, int, Y坐标-->1
</positionY>
<height>
<!--ro, req, int, 宽-->1
</height>
<width>
<!--ro, req, int, 高-->1
</width>
</Position>
<materialNo>
<!--ro, opt, int, 素材ID-->1
</materialNo>
</Icon>
</NormalTemperature>
<AbnormalTemperature>
<!--ro, opt, object, 异常体温信息-->
<PeopleNumber>
<!--ro, opt, object, 异常体温人数信息-->
<enabled>
<!--ro, req, bool, 使能-->true
</enabled>
<fontSize>
<!--ro, req, int, 字体大小-->1
</fontSize>
<fontColor>
<!--ro, req, object, 字体颜色-->
<RGB>
<!--ro, opt, int, 十进制数据表示三原色-->1
</RGB>
</FontColor>
<Position>
<!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
<positionX>
<!--ro, req, int, X坐标-->1
</positionX>
<positionY>
<!--ro, req, int, Y坐标-->1
</positionY>
<height>
<!--ro, req, int, 宽-->1
</height>
<width>
<!--ro, req, int, 高-->1
</width>
</Position>
<text>
<!--ro, opt, string, 标题文字-->test
</text>
</PeopleNumber>
<Icon>
<!--ro, opt, object, 异常体温图标-->
<enabled>
<!--ro, req, bool, 使能-->true
</enabled>
<Position>
<!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
<positionX>
<!--ro, req, int, X坐标-->1
</positionX>
<positionY>
<!--ro, req, int, Y坐标-->1
</positionY>
<height>
<!--ro, req, int, 宽-->1
</height>
```

```

</height>
<width>
  <!--ro, req, int, 高-->1
</width>
</Position>
<materialNo>
  <!--ro, opt, int, 素材ID-->1
</materialNo>
</Icon>
</AbnormalTemperature>
</SignIn>
<EventParam>
  <!--ro, opt, object, 事件参数-->
  <EventResourceList>
    <!--ro, opt, array, 事件源信息列表, subType:object-->
    <EventResource>
      <!--ro, opt, object, 事件源信息-->
      <ipVersion>
        <!--ro, req, enum, ip地址版本, subType:string, [v4#v4,v6#v6,dual#双]-->v4
      </ipVersion>
      <ipAddress>
        <!--ro, opt, string, ipv4地址-->test
      </ipAddress>
      <ipv6Address>
        <!--ro, opt, string, ipv6地址-->test
      </ipv6Address>
      <portNo>
        <!--ro, req, int, 端口号-->1
      </portNo>
      <userName>
        <!--ro, req, string, 用户名-->test
      </userName>
      <password>
        <!--ro, req, string, 密码-->test
      </password>
      <channels>
        <!--ro, opt, string, 通道号, desc:多选以逗号隔开-->1,2
      </channels>
    </EventResource>
  </EventResourceList>
  <backPicId>
    <!--ro, opt, int, 背景图片素材ID-->1
  </backPicId>
  <Title>
    <!--ro, opt, object, 标题-->
    <enabled>
      <!--ro, req, bool, 使能-->true
    </enabled>
    <fontSize>
      <!--ro, req, int, 字体大小-->1
    </fontSize>
    <fontColor>
      <!--ro, req, object, 字体颜色-->
      <RGB>
        <!--ro, opt, int, 十进制数据表示三原色-->1
      </RGB>
    </fontColor>
    <Position>
      <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
      <positionX>
        <!--ro, req, int, X坐标-->1
      </positionX>
      <positionY>
        <!--ro, req, int, Y坐标-->1
      </positionY>
      <height>
        <!--ro, req, int, 宽-->1
      </height>
      <width>
        <!--ro, req, int, 高-->1
      </width>
    </Position>
    <text>
      <!--ro, opt, string, 标题文字-->test
    </text>
  </Title>
  <PeopleCounting>
    <!--ro, opt, object, 人数统计提醒-->
    <maxPeopleNumber>
      <!--ro, opt, int, 最大可进入人数-->1
    </maxPeopleNumber>
    <unavailableBackPicId>
      <!--ro, opt, int, 人员不能进入时控件背景图片id-->1
    </unavailableBackPicId>
    <supportChangeByPeople>
      <!--ro, req, enum, 支持根据人数修改样式的字段, subType:string,
[prompt#prompt,peopleIcon#peopleIcon,availablePeopleNumber#availablePeopleNumber]-->prompt
    </supportChangeByPeople>
    <Prompt>
      <!--ro, opt, object, 提示信息-->
      <enabled>
        <!--ro, req, bool, 使能-->true
      </enabled>

```

```
<fontSize>
  <!--ro, req, int, 字体大小-->1
</fontSize>
<Position>
  <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
  <positionX>
    <!--ro, req, int, X坐标-->1
  </positionX>
  <positionY>
    <!--ro, req, int, Y坐标-->1
  </positionY>
  <height>
    <!--ro, req, int, 宽-->1
  </height>
  <width>
    <!--ro, req, int, 高-->1
  </width>
</Position>
<text>
  <!--ro, opt, string, 标题文字-->test
</text>
<FontColor>
  <!--ro, opt, object, 人员可以进入时的字体颜色-->
  <RGB>
    <!--ro, opt, int, 十进制数据表示三原色-->1
  </RGB>
</FontColor>
<unavailableText>
  <!--ro, opt, string, 人员不能进入时的标题文字-->test
</unavailableText>
<UnavailableFontColor>
  <!--ro, opt, object, 人员不能进入时的字体颜色-->
  <RGB>
    <!--ro, opt, int, 十进制数据表示三原色-->1
  </RGB>
</UnavailableFontColor>
</Prompt>
<PeopleIcon>
  <!--ro, opt, object, 行人通行图标-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <Position>
    <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--ro, req, int, X坐标-->1
    </positionX>
    <positionY>
      <!--ro, req, int, Y坐标-->1
    </positionY>
    <height>
      <!--ro, req, int, 宽-->1
    </height>
    <width>
      <!--ro, req, int, 高-->1
    </width>
  </Position>
  <materialNo>
    <!--ro, opt, int, 素材ID-->1
  </materialNo>
</PeopleIcon>
<InsidePeopleTitle>
  <!--ro, opt, object, 已经进入人数标题-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <fontSize>
    <!--ro, req, int, 字体大小-->1
  </fontSize>
  <FontColor>
    <!--ro, req, object, 字体颜色-->
    <RGB>
      <!--ro, opt, int, 十进制数据表示三原色-->1
    </RGB>
  </FontColor>
  <Position>
    <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--ro, req, int, X坐标-->1
    </positionX>
    <positionY>
      <!--ro, req, int, Y坐标-->1
    </positionY>
    <height>
      <!--ro, req, int, 宽-->1
    </height>
    <width>
      <!--ro, req, int, 高-->1
    </width>
  </Position>
  <text>
    <!--ro, opt, string, 标题文字-->test
  </text>
</InsidePeopleTitle>
```

```
</UnavailableFontColor>
<InsidePeopleNumber>
  <!--ro, opt, object, 已经进入人数-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <fontSize>
    <!--ro, req, int, 字体大小-->1
  </fontSize>
  <FontColor>
    <!--ro, req, object, 字体颜色-->
    <RGB>
      <!--ro, opt, int, 十进制数据表示三原色-->1
    </RGB>
  </FontColor>
  <Position>
    <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--ro, req, int, X坐标-->1
    </positionX>
    <positionY>
      <!--ro, req, int, Y坐标-->1
    </positionY>
    <height>
      <!--ro, req, int, 宽-->1
    </height>
    <width>
      <!--ro, req, int, 高-->1
    </width>
  </Position>
</InsidePeopleNumber>
<AvailablePeopleTitle>
  <!--ro, opt, object, 还可以进入人数标题-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <fontSize>
    <!--ro, req, int, 字体大小-->1
  </fontSize>
  <FontColor>
    <!--ro, req, object, 字体颜色-->
    <RGB>
      <!--ro, opt, int, 十进制数据表示三原色-->1
    </RGB>
  </FontColor>
  <Position>
    <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--ro, req, int, X坐标-->1
    </positionX>
    <positionY>
      <!--ro, req, int, Y坐标-->1
    </positionY>
    <height>
      <!--ro, req, int, 宽-->1
    </height>
    <width>
      <!--ro, req, int, 高-->1
    </width>
  </Position>
  <text>
    <!--ro, opt, string, 标题文字-->test
  </text>
</AvailablePeopleTitle>
<AvailablePeopleNumber>
  <!--ro, opt, object, 还可以进入人数-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <fontSize>
    <!--ro, req, int, 字体大小-->1
  </fontSize>
  <Position>
    <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--ro, req, int, X坐标-->1
    </positionX>
    <positionY>
      <!--ro, req, int, Y坐标-->1
    </positionY>
    <height>
      <!--ro, req, int, 宽-->1
    </height>
    <width>
      <!--ro, req, int, 高-->1
    </width>
  </Position>
  <FontColor>
    <!--ro, opt, object, 人员可以进入时的字体颜色-->
    <RGB>
      <!--ro, opt, int, 十进制数据表示三原色-->1
    </RGB>
  </FontColor>
</UnavailableFontColor>
```

```
<!--ro, opt, object, 人员不能进入时的字体颜色-->
<RGB>
  <!--ro, opt, int, 十进制数据表示三原色-->1
</RGB>
</UnavailableFontColor>
</AvailablePeopleNumber>
<UnavailablePeopleIcon>
  <!--ro, opt, object, 人员不能进入时的行人图标-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <Position>
    <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--ro, req, int, X坐标-->1
    </positionX>
    <positionY>
      <!--ro, req, int, Y坐标-->1
    </positionY>
    <height>
      <!--ro, req, int, 宽-->1
    </height>
    <width>
      <!--ro, req, int, 高-->1
    </width>
  </Position>
  <materialNo>
    <!--ro, opt, int, 素材ID-->1
  </materialNo>
</UnavailablePeopleIcon>
<AudioAlarm>
  <!--ro, opt, object, 音频报警信息-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <materialNo>
    <!--ro, opt, int, 音频素材ID-->1
  </materialNo>
</AudioAlarm>
<PeopleCounting>
<Mask>
  <!--ro, opt, object, 未戴口罩提醒-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <fontSize>
    <!--ro, req, int, 字体大小-->1
  </fontSize>
  <fontColor>
    <!--ro, req, object, 字体颜色-->
    <RGB>
      <!--ro, opt, int, 十进制数据表示三原色-->1
    </RGB>
  </FontColor>
  <BackColor>
    <!--ro, req, object, 背景颜色-->
    <RGB>
      <!--ro, opt, int, 十进制数据表示三原色-->1
    </RGB>
  </BackColor>
  <Position>
    <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--ro, req, int, X坐标-->1
    </positionX>
    <positionY>
      <!--ro, req, int, Y坐标-->1
    </positionY>
    <height>
      <!--ro, req, int, 宽-->1
    </height>
    <width>
      <!--ro, req, int, 高-->1
    </width>
  </Position>
  <text>
    <!--ro, opt, string, 标题文字-->test
  </text>
  <MaskIcon>
    <!--ro, opt, object, 未戴口罩图标-->
    <enabled>
      <!--ro, req, bool, 使能-->true
    </enabled>
    <Position>
      <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
      <positionX>
        <!--ro, req, int, X坐标-->1
      </positionX>
      <positionY>
        <!--ro, req, int, Y坐标-->1
      </positionY>
      <height>
        <!--ro, req, int, 宽-->1
      </height>
```

```
</neight>
<width>
  <!--ro, req, int, 高-->1
</width>
</Position>
<materialNo>
  <!--ro, opt, int, 素材ID-->1
</materialNo>
</MaskIcon>
<AudioAlarm>
  <!--ro, opt, object, 音频报警信息-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <materialNo>
    <!--ro, opt, int, 音频素材ID-->1
  </materialNo>
</AudioAlarm>
</Mask>
<Temperature>
  <!--ro, opt, object, 温度异常提醒-->
  <Prompt>
    <!--ro, opt, object, 提示信息-->
    <enabled>
      <!--ro, req, bool, 使能-->true
    </enabled>
    <fontSize>
      <!--ro, req, int, 字体大小-->1
    </fontSize>
    <FontColor>
      <!--ro, req, object, 字体颜色-->
      <RGB>
        <!--ro, opt, int, 十进制数据表示三原色-->1
      </RGB>
    </FontColor>
    <BackColor>
      <!--ro, req, object, 背景颜色-->
      <RGB>
        <!--ro, opt, int, 十进制数据表示三原色-->1
      </RGB>
    </BackColor>
    <Position>
      <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
      <positionX>
        <!--ro, req, int, X坐标-->1
      </positionX>
      <positionY>
        <!--ro, req, int, Y坐标-->1
      </positionY>
      <height>
        <!--ro, req, int, 宽-->1
      </height>
      <width>
        <!--ro, req, int, 高-->1
      </width>
    </Position>
    <text>
      <!--ro, opt, string, 标题文字-->test
    </text>
  </Prompt>
  <TemperatureIcon>
    <!--ro, opt, object, 温度异常图标-->
    <enabled>
      <!--ro, req, bool, 使能-->true
    </enabled>
    <Position>
      <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
      <positionX>
        <!--ro, req, int, X坐标-->1
      </positionX>
      <positionY>
        <!--ro, req, int, Y坐标-->1
      </positionY>
      <height>
        <!--ro, req, int, 宽-->1
      </height>
      <width>
        <!--ro, req, int, 高-->1
      </width>
    </Position>
    <materialNo>
      <!--ro, opt, int, 素材ID-->1
    </materialNo>
  </TemperatureIcon>
  <AudioAlarm>
    <!--ro, opt, object, 音频报警信息-->
    <enabled>
      <!--ro, req, bool, 使能-->true
    </enabled>
    <materialNo>
      <!--ro, opt, int, 音频素材ID-->1
    </materialNo>
  </AudioAlarm>
</abnormalHighTemperature>
```

```
<!--ro, opt, float, 异常高温值, unit:℃, unitType:温度, desc:保留一位小数-->37.5
</abnormalHighTemperature>
</Temperature>
<wearMask>
  <!--ro, opt, object, 已戴口罩提醒-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <fontSize>
    <!--ro, req, int, 字体大小-->1
  </fontSize>
  <fontColor>
    <!--ro, req, object, 字体颜色-->
    <RGB>
      <!--ro, opt, int, 十进制数据表示三原色-->1
    </RGB>
  </FontColor>
  <BackColor>
    <!--ro, req, object, 背景颜色-->
    <RGB>
      <!--ro, opt, int, 十进制数据表示三原色-->1
    </RGB>
  </BackColor>
  <Position>
    <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--ro, req, int, X坐标-->1
    </positionX>
    <positionY>
      <!--ro, req, int, Y坐标-->1
    </positionY>
    <height>
      <!--ro, req, int, 宽-->1
    </height>
    <width>
      <!--ro, req, int, 高-->1
    </width>
  </Position>
  <text>
    <!--ro, opt, string, 标题文字-->test
  </text>
  <MaskIcon>
    <!--ro, opt, object, 已戴口罩图标-->
    <enabled>
      <!--ro, req, bool, 使能-->true
    </enabled>
    <Position>
      <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
      <positionX>
        <!--ro, req, int, X坐标-->1
      </positionX>
      <positionY>
        <!--ro, req, int, Y坐标-->1
      </positionY>
      <height>
        <!--ro, req, int, 宽-->1
      </height>
      <width>
        <!--ro, req, int, 高-->1
      </width>
    </Position>
    <materialNo>
      <!--ro, opt, int, 素材ID-->1
    </materialNo>
  </MaskIcon>
</wearMask>
<NormalTemperature>
  <!--ro, opt, object, 温度正常提醒-->
  <Prompt>
    <!--ro, opt, object, 提示信息-->
    <enabled>
      <!--ro, req, bool, 使能-->true
    </enabled>
    <fontSize>
      <!--ro, req, int, 字体大小-->1
    </fontSize>
    <fontColor>
      <!--ro, req, object, 字体颜色-->
      <RGB>
        <!--ro, opt, int, 十进制数据表示三原色-->1
      </RGB>
    </FontColor>
    <BackColor>
      <!--ro, req, object, 背景颜色-->
      <RGB>
        <!--ro, opt, int, 十进制数据表示三原色-->1
      </RGB>
    </BackColor>
    <Position>
      <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
      <positionX>
        <!--ro, req, int, X坐标-->1
      </positionX>
```

```

<positionY>
  <!--ro, req, int, Y坐标-->1
</positionY>
<height>
  <!--ro, req, int, 宽-->1
</height>
<width>
  <!--ro, req, int, 高-->1
</width>
</Position>
<text>
  <!--ro, opt, string, 标题文字-->test
</text>
</Prompt>
<TemperatureIcon>
  <!--ro, opt, object, 温度正常图标-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <Position>
    <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--ro, req, int, X坐标-->1
    </positionX>
    <positionY>
      <!--ro, req, int, Y坐标-->1
    </positionY>
    <height>
      <!--ro, req, int, 宽-->1
    </height>
    <width>
      <!--ro, req, int, 高-->1
    </width>
  </Position>
  <materialNo>
    <!--ro, opt, int, 素材ID-->1
  </materialNo>
</TemperatureIcon>
</NormalTemperature>
<NormalStatusAudioAlarm>
  <!--ro, opt, object, 正常状态下音频报警信息, desc:人员在检测时未发现上面的人数超限、未戴口罩、体温异常等事件时设备进行播报-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <materialNo>
    <!--ro, opt, int, 音频素材ID-->1
  </materialNo>
</NormalStatusAudioAlarm>
</EventParam>
<AlarmDataSourceProperty>
  <!--ro, opt, object, 数据源信息, desc:该字段能力集缺少, 仅信发组件和设备使用-->
  <PictureProperty>
    <!--ro, opt, object, 图片属性-->
    <PictureSwitchEffect>
      <!--ro, opt, enum, 图片切换效果, subType:string, [LeftInRightOut#左进右出,rightInLeftOut#右进左出]-->leftInRightOut
    </PictureSwitchEffect>
    <PictureScaleType>
      <!--ro, opt, enum, 图片缩放类型, subType:string, [fitCenter#拉伸填充,centerCrop#截取填充,fixXY#适应宽高]-->fitCenter
    </PictureScaleType>
  </PictureProperty>
  <AlarmDataSource>
    <!--ro, opt, object, 数据源-->
    <MaterialNo>
      <!--ro, opt, int, 素材编号-->1
    </MaterialNo>
    <Name>
      <!--ro, opt, string, 素材名称-->test
    </Name>
    <Address>
      <!--ro, opt, string, 素材URL地址-->test
    </Address>
    <WebProperty>
      <!--ro, opt, object, 网页属性-->
      <webType>
        <!--ro, opt, enum, 网页类型, subType:string, [desktop#桌面版,mobile#移动版], desc:默认按照设备的系统类型（windows或Android）来显示网页-->desktop
      </webType>
    </WebProperty>
  </AlarmDataSource>
</AlarmDataSourceProperty>
<ConferenceParam>
  <!--ro, opt, object, 会议信息, desc:当页面用于会议界面时有效-->
  <backPicId>
    <!--ro, opt, int, 背景图片素材ID-->1
  </backPicId>
  <Title>
    <!--ro, opt, object, 会议标题（主题）, desc:1、会议标题（主题）的文字内容通过场地预约配置协议下发,对应/ISAPI/Publish/Signage/placeReservation?format=json中的programID字段。2、可通过会议计划（/ISAPI/InteractivePanel/Meeting/plan?format=json）中的meetingTopic字段下发-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <fontSize>
    <!--ro, req, int, 字体大小-->1
  </fontSize>

```

```
<!--- FontColor -->
<FontColor>
  <!--ro, req, object, 字体颜色-->
  <RGB>
    <!--ro, opt, int, 十进制数据表示三原色-->1
  </RGB>
</FontColor>
<!--- BackColor -->
<BackColor>
  <!--ro, req, object, 背景颜色-->
  <RGB>
    <!--ro, opt, int, 十进制数据表示三原色-->1
  </RGB>
</BackColor>
<!--- Position -->
<Position>
  <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
  <positionX>
    <!--ro, req, int, X坐标-->1
  </positionX>
  <positionY>
    <!--ro, req, int, Y坐标-->1
  </positionY>
  <height>
    <!--ro, req, int, 宽-->1
  </height>
  <width>
    <!--ro, req, int, 高-->1
  </width>
</Position>
<!--- alignType -->
<alignType>
  <!--ro, req, enum, 对齐方式, subType:string, [Left#左,right#右,middle#中]-->middle
</alignType>
<!--- Title -->
<VoiceBroadcast>
  <!--ro, opt, object, 语音播报-->
  <enabled>
    <!--ro, opt, bool, 是否启用语音播报-->true
  </enabled>
  <broadcastSpeed>
    <!--ro, opt, enum, 播报语速, subType:string, [slower#慢,slow#较慢,medium#中速,fast#较快,faster#快]-->slower
  </broadcastSpeed>
  <inLibVoice>
    <!--ro, opt, string, 在库人员语音播报内容-->test
  </inLibVoice>
  <outLibVoice>
    <!--ro, opt, string, 未知人员语音播报内容-->test
  </outLibVoice>
</VoiceBroadcast>
<!--- WelcomeWord -->
<WelcomeWord>
  <!--ro, opt, object, 欢迎词-->
  <enabled>
    <!--ro, opt, bool, 是否启用欢迎词-->true
  </enabled>
  <inLibWord>
    <!--ro, opt, string, 在库人员欢迎词内容, range:[1,128]-->test
  </inLibWord>
  <outLibWord>
    <!--ro, opt, string, 未知人员欢迎词内容, range:[1,128]-->test
  </outLibWord>
</WelcomeWord>
<!--- ShowInfo -->
<ShowInfo>
  <!--ro, opt, object, 展示信息, desc:下发该字段需要人脸签到, 不下发该字段, 不需要人脸签到-->
  <row>
    <!--ro, opt, int, 窗口内人员签到展示行数-->1
  </row>
  <column>
    <!--ro, opt, int, 窗口内人员签到展示列数-->1
  </column>
  <Position>
    <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--ro, req, int, x坐标-->1
    </positionX>
    <positionY>
      <!--ro, req, int, y坐标-->1
    </positionY>
    <height>
      <!--ro, req, int, 宽-->1
    </height>
    <width>
      <!--ro, req, int, 高-->1
    </width>
  </Position>
  <inLib>
    <!--ro, opt, object, 在库信息-->
    <nameEnabled>
      <!--ro, opt, bool, 在库人员是否展示姓名-->true
    </nameEnabled>
  </inLib>
  <defaultShowInfoEnabled>
    <!--ro, opt, bool, 是否采用人员默认展示信息, desc:下发该字段, 且为true, 表示人员展示信息采用默认值. 坐标信息等不用下发-->true
  </defaultShowInfoEnabled>
</ShowInfo>
<!--- ModuleInfoList -->
<ModuleInfoList>
  <!--ro, opt, array, 界面模块信息列表, subType:object-->
```

```

<ModuleInfo>
  <!--ro, opt, object, 界面模块信息-->
  <moduleType>
    <!--ro, req, enum, 模块类型, subType:string, [conferenceTime#会议时间(对应timeRange), conferenceSite#会议地点(对应会议室名称
meetingName), conferenceStatus#会议状态(对应meetingStatus), signInPersonNum#已签到人数, totalPersonNum#应到人数, unSignInPersonNum#未签到人
数, signInAndTotalPersonNum#已签到/应到人数, QRCode#会议室二维码, meetingHostName#主持人, empLoyeeNo#参会人, customKey#自定义字段, nextMeeting#当下一场会议信
息]-->conferenceTime
  </moduleType>
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <fontSize>
    <!--ro, req, int, 字体大小-->1
  </fontSize>
  <FontColor>
    <!--ro, req, object, 字体颜色-->
    <RGB>
      <!--ro, opt, int, 十进制数据表示三原色-->1
    </RGB>
  </FontColor>
  <BackColor>
    <!--ro, opt, object, 背景颜色-->
    <RGB>
      <!--ro, opt, int, 十进制数据表示三原色-->1
    </RGB>
  </BackColor>
  <backPicId>
    <!--ro, opt, int, 背景图片素材ID, desc: 背景图和背景颜色同时仅支持一种, 两者均不发, 设备优先解析背景图字段-->1
  </backPicId>
  <Position>
    <!--ro, opt, object, 位置, desc: 左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--ro, req, int, X坐标-->1
    </positionX>
    <positionY>
      <!--ro, req, int, Y坐标-->1
    </positionY>
    <height>
      <!--ro, req, int, 宽-->1
    </height>
    <width>
      <!--ro, req, int, 高-->1
    </width>
  </Position>
  <alignType>
    <!--ro, req, enum, 对齐方式, subType:string, [left#左, right#右, middle#中]-->middle
  </alignType>
  <CustomKey>
    <!--ro, opt, object, 自定义字段-->
    <key>
      <!--ro, opt, string, 自定义字段-->test
    </key>
    <associateProtocolURL>
      <!--ro, opt, string, 关联协议URL-->test
    </associateProtocolURL>
  </CustomKey>
</ModuleInfo>
</ModuleInfoList>
<conferenceStatus>
  <!--ro, opt, enum, 会议状态, subType:string, [idle#空闲, signing#签到中, meeting#开会中], desc: 根据不同的会议状态, 配置不同的会议界面-->idle
</conferenceStatus>
<BackColor>
  <!--ro, opt, object, 背景颜色-->
  <RGB>
    <!--ro, opt, int, 十进制数据表示三原色-->1
  </RGB>
</BackColor>
</ConferenceParam>
<CallBtn>
  <!--ro, opt, object, 呼叫按键窗口-->
  <backPicId>
    <!--ro, opt, int, 背景图片素材ID-->1
  </backPicId>
</CallBtn>
<OpenDoorBtn>
  <!--ro, opt, object, 开门按键窗口-->
  <backPicId>
    <!--ro, opt, int, 背景图片素材ID-->1
  </backPicId>
</OpenDoorBtn>
<QRCodeBtn>
  <!--ro, opt, object, 二维码按键窗口-->
  <backPicId>
    <!--ro, opt, int, 背景图片素材ID-->1
  </backPicId>
</QRCodeBtn>
<AuthenticationSuccess>
  <!--ro, opt, object, 认证成功窗口-->
  <BackColor>
    <!--ro, opt, object, 背景颜色-->
    <RGB>
      <!--ro, opt, int, 十进制数据表示三原色-->1
    </RGB>
  </BackColor>

```

```

</backPicId>
<backPicId>
  <!--ro, opt, int, 背景图片素材ID, desc:背景图和背景颜色同时仅支持一种,两者均下发,设备优先解析背景图字段-->1
</backPicId>
<SubWindowList>
  <!--ro, opt, object, 子窗口列表-->
  <SubWindow>
    <!--ro, opt, object, 子窗口-->
    <subWindowType>
      <!--ro, req, enum, 子窗口类型, subType:string, [temperature#温度,facePic#人脸图片,name#姓名,employeeNo#工号,group#组织/部门,tips#提示信息]--
>temperature
      </subWindowType>
      <enabled>
        <!--ro, req, bool, 使能-->true
      </enabled>
      <fontSize>
        <!--ro, opt, int, 字体大小, unit:px, unitType:图像显示-->1
      </fontSize>
      <fontColor>
        <!--ro, opt, int, 字体颜色-->1
      </fontColor>
      <fontStyle>
        <!--ro, opt, string, 字体风格, desc:bold-加粗, italic-倾斜, underline-下划线, strikethrough-删除线-->bold,italic,underline,strikeout
      </fontStyle>
      <Position>
        <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
        <positionX>
          <!--ro, req, int, x坐标-->1
        </positionX>
        <positionY>
          <!--ro, req, int, y坐标-->1
        </positionY>
        <height>
          <!--ro, req, int, 宽-->1
        </height>
        <width>
          <!--ro, req, int, 高-->1
        </width>
      </Position>
    </SubWindow>
  </SubWindowList>
</AuthenticationSuccess>
<AuthenticationFail>
  <!--ro, opt, object, 认证失败窗口-->
  <BackColor>
    <!--ro, opt, object, 背景颜色-->
    <RGB>
      <!--ro, opt, int, 十进制数据表示三原色-->1
    </RGB>
  </BackColor>
  <backPicId>
    <!--ro, opt, int, 背景图片素材ID, desc:背景图和背景颜色同时仅支持一种,两者均下发,设备优先解析背景图字段-->1
  </backPicId>
  <SubWindowList>
    <!--ro, opt, object, 子窗口列表-->
    <SubWindow>
      <!--ro, opt, object, 子窗口-->
      <subWindowType>
        <!--ro, req, enum, 子窗口类型, subType:string, [temperature#温度,facePic#人脸图片,name#姓名,employeeNo#工号,group#组织/部门,tips#提示信息]--
>temperature
        </subWindowType>
        <enabled>
          <!--ro, req, bool, 使能-->true
        </enabled>
        <fontSize>
          <!--ro, opt, int, 字体大小, unit:px, unitType:图像显示-->1
        </fontSize>
        <fontColor>
          <!--ro, opt, int, 字体颜色-->1
        </fontColor>
        <fontStyle>
          <!--ro, opt, string, 字体风格, desc:bold-加粗, italic-倾斜, underline-下划线, strikethrough-删除线-->bold,italic,underline,strikeout
        </fontStyle>
        <Position>
          <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
          <positionX>
            <!--ro, req, int, x坐标-->1
          </positionX>
          <positionY>
            <!--ro, req, int, y坐标-->1
          </positionY>
          <height>
            <!--ro, req, int, 宽-->1
          </height>
          <width>
            <!--ro, req, int, 高-->1
          </width>
        </Position>
        <facePicID>
          <!--ro, opt, int, 默认人脸图片ID, step:1, dep:and,
{$.PageList.Page.WindowsList[*].Windows.AuthenticationFail.SubWindowList.SubWindow.subWindowType,eq,facePic}-->1
        </facePicID>
      </SubWindow>
    </SubWindowList>

```

```
</AuthenticationFail>
<StatusBar>
  <!--ro, opt, object, 状态栏, desc:当素材类型为状态栏时有效, 且必填-->
  <backPicId>
    <!--ro, opt, int, 背景图片素材ID, desc:背景图和背景颜色同时仅支持一种,两者均下发,设备优先解析背景图字段-->1
  </backPicId>
  <statusBarType>
    <!--ro, req, enum, 状态栏类型, subType:string, [fixed#固定的], desc:当为固定类型时, 窗口位置不生效, 设备将状态栏固定在顶部显示-->fixed
  </statusBarType>
  <SubStatusList>
    <!--ro, opt, object, 子状态列表-->
    <SubStatus>
      <!--ro, opt, object, 子状态-->
      <statusType>
        <!--ro, req, enum, 子状态, subType:string, [dial#拨号状态(3G4G等),wifi#wifi状态,EZVIZ#萤石状态,wiredNetwork#有线网络状态,deploy#布防状态,temperature#测温]-->dial
      </statusType>
      <enabled>
        <!--ro, req, bool, 是否显示-->true
      </enabled>
    </SubStatus>
  </SubStatusList>
</StatusBar>
<ExamTemplate>
  <!--ro, opt, object, 考试模板, desc:当页面用于会议界面时有效-->
  <examTemplateID>
    <!--ro, opt, string, 模板ID-->test
  </examTemplateID>
  <backPicURL>
    <!--ro, opt, string, 背景图片素材URL, range:[1,1024]-->test
  </backPicURL>
  <backgroundDirection>
    <!--ro, opt, enum, 背景图方向, subType:string, [horizontal#水平,vertical#垂直]-->horizontal
  </backgroundDirection>
  <ModuleInfoList>
    <!--ro, opt, array, 界面模块信息列表, subType:object-->
    <ModuleInfo>
      <!--ro, opt, object, 界面模块信息-->
      <moduleType>
        <!--ro, opt, enum, 模块类型, subType:string, [schoolName#学校名称,currentTime#当前时间,examinationPlanLable#考试计划标签,examinationPlanName#考试计划名称,examRoomNumberLable#考场号标签,examRoomNumber#考场号,examRoomAddress#考试地点,examRoomAddressLable,examDescriptionLable#考试说明标签,examDescription#考试说明,examSubjectLable#考试科目标签,examSubject#考试科目,examTimeLable#考试时间标签,examTime#考试时间,teacherLable#监考老师标签,teacher#监考老师,teacherLable#监考老师标签,teacher#监考老师,examNumberRangeLable#考号范围,examNumberRange#考号范围]-->schoolName
      </moduleType>
      <enabled>
        <!--ro, opt, bool, 使能-->true
      </enabled>
      <Position>
        <!--ro, opt, object, 位置, desc:左上角为坐标原点-->
        <positionX>
          <!--ro, req, int, X坐标-->1
        </positionX>
        <positionY>
          <!--ro, req, int, Y坐标-->1
        </positionY>
        <height>
          <!--ro, req, int, 高-->1
        </height>
        <width>
          <!--ro, req, int, 宽-->1
        </width>
      </Position>
      <fontSize>
        <!--ro, opt, int, 字体大小-->1
      </fontSize>
      <fontStyleList>
        <!--ro, opt, array, 字体风格列表, subType:object-->
        <fontStyle>
          <!--ro, opt, enum, 字体风格, subType:string, [bold#加粗,italic#倾斜,regular#常规,striketrough#中划线,underLine#下划线]-->bold
        </fontStyle>
      </fontStyleList>
      <fontType>
        <!--ro, opt, enum, 字体类型, subType:string, [custom#自定义,default#默认,kaiTi#楷体,simHei#黑体,simSun#宋体]-->custom
      </fontType>
      <fontColor>
        <!--ro, opt, object, 字体颜色-->
        <RGB>
          <!--ro, opt, int, 十进制数据表示三原色-->1
        </RGB>
      </fontColor>
      <fontTransparent>
        <!--ro, opt, int, 字体透明度-->1
      </fontTransparent>
      <alignType>
        <!--ro, opt, enum, 字体对齐方式, subType:string, [left#左,right#右,middle#中]-->middle
      </alignType>
      <characterSpacing>
        <!--ro, opt, int, 字符间距-->1
      </characterSpacing>
      <lineSpacing>
        <!--ro, opt, int, 行间距-->1
      </lineSpacing>
      <layerNo>
```

```
<!--ro, opt, int, 图层编号, desc:数字越大, 表示往上层-->1
</layerNo>
</ModuleInfo>
</ModuleInfoList>
</ExamTemplate>
<ChartData>
  <!--ro, opt, object, 图表数据-->
  <materialNo>
    <!--ro, req, int, 素材编号-->1
  </materialNo>
  <chartType>
    <!--ro, opt, enum, 图表类型, subType:string, [histogram#柱状图,pieChart#饼图]-->histogram
  </chartType>
  <backPicId>
    <!--ro, opt, int, 背景图片素材编号-->1
  </backPicId>
  <HeadDataList>
    <!--ro, opt, array, 表头数据列表, subType:object-->
    <HeadData>
      <!--ro, opt, object, 表头数据-->
      <id>
        <!--ro, req, int, 索引-->1
      </id>
      <data>
        <!--ro, req, string, 数据-->test
      </data>
      <headType>
        <!--ro, opt, enum, 表头类型, subType:string, [axis#轴,value#值]-->axis
      </headType>
    </HeadData>
  </HeadDataList>
</ChartData>
</Windows>
</WindowsList>
</Page>
</PageList>
```

69.63.4 获取页面参数配置能力

Request URL

GET /ISAPI/Publish/ProgramMgr/program/<programID>/page/<pageID>/capabilities

查询参数

参数名称	参数类型	描述
programID	string	节目ID
pageID	string	页面ID

请求报文

无

响应报文

```

<?xml version="1.0" encoding="UTF-8"?>
<Page xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 页面, attr:version{req, string, 协议版本}-->
  <id>
    <!--ro, req, int, 索引-->1
  </id>
  <PageBasicInfo>
    <!--ro, req, object, 页面基本信息-->
    <pageName max="10">
      <!--ro, opt, string, 页面名称, attr:max{req, int, 最大值}-->test
    </pageName>
    <switchDuration min="1" max="10">
      <!--ro, opt, int, 切换间隔, unit:s, unitType:时间, attr:min{req, int, 最小值},max{req, int, 最大值}-->1
    </switchDuration>
    <switchEffect
      opt="none,random,boxShrink,boxSpread,cycleShrink,cycSpread,eraseUp,eraseDown,eraseLeft,eraseRight,verticalShelter,horizontalShelter,verticalChessboard,horizontalChessboard,dissolve,leftRightToCenter,ceterToLeftRight,upDownToCenter,centerToUpDown,drawOutLeftDown,drawOutLeftUp,drawOutRightDown,drawOutRightUp,verticalLine,horizontalLine">
      <!--ro, opt, enum, 切换效果, subType:string, [none#无,random#随机,boxShrink#盒状收缩,boxSpread#盒状展开,cycleShrink#圆形收缩,cycSpread#圆形展开,eraseUp#向上擦除,eraseDown#向下擦除,eraseLeft#向左擦除,eraseRight#向右擦除,verticalShelter#垂直屏蔽,horizontalShelter#水平屏蔽,verticalChessboard#纵向棋盘式,horizontalChessboard#横向棋盘式,dissolve#随机溶解,leftRightToCenter#左右向中间缩进,ceterToLeftRight#中央向左右扩展,upDownToCenter#上下向中央缩进,centerToUpDown#中央向上下扩展,drawOutLeftDown#从左下抽出,drawOutLeftUp#从左上抽出,drawOutRightDown#从右下抽出,drawOutRightUp#从右上抽出,verticalLine#随机垂直线条,horizontalLine#随机垂直线条], attr:opt{req, string, 取值范围}-->none
    </switchEffect>
  </PageBasicInfo>
  <WindowsList size="10">
    <!--ro, opt, array, 窗口信息列表, subType:object, attr:size{req, int, 数组元素总数}-->
    <Windows>
      <!--ro, opt, object, 窗口信息-->
      <id>
        <!--ro, req, int, 内容编号-->1
      </id>
      <Position>
        <!--ro, opt, object, 内容位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
        <positionX min="1" max="10">
          <!--ro, req, int, x坐标, attr:min{req, int, 最小值},max{req, int, 最大值}-->1
        </positionX>
        <positionY min="1" max="10">
          <!--ro, req, int, y坐标, attr:min{req, int, 最小值},max{req, int, 最大值}-->1
        </positionY>
        <height min="1" max="10">
          <!--ro, req, int, 高, attr:min{req, int, 最小值},max{req, int, 最大值}-->1
        </height>
        <width min="1" max="10">
          <!--ro, req, int, 宽, attr:min{req, int, 最小值},max{req, int, 最大值}-->1
        </width>
      </Position>
      <layerNo min="1" max="10">
        <!--ro, req, int, 图层编号, attr:min{req, int, 最小值},max{req, int, 最大值}-->1
      </layerNo>
      <WinMaterialInfo>
        <!--ro, opt, object, 窗口素材信息-->
        <materialType opt="static,dynamic,other">
          <!--ro, req, enum, 素材类型, subType:string, [static#静态,dynamic#动态,other#其他], attr:opt{req, string, 取值范围}-->static
        </materialType>
        <staticMaterialType opt="picture,flash,audio,video,document,ppt,doc,excel,pdf,web,app,signIn,conference">
          <!--ro, opt, enum, 静态素材类型, subType:string, [picture#图片,flash#flash,audio#音频,video#视频,document#文本,ppt#ppt,doc#文档,excel#表格,pdf#pdf,web#网页,app#app,signIn#签到,conference#会议], attr:opt{req, string, 取值范围}-->picture
        </staticMaterialType>
      </WinMaterialInfo>
      <PlayItemList size="10">
        <!--ro, opt, array, 窗口播放列表, subType:object, attr:size{req, int, 数组元素总数}-->
        <PlayItem>
          <!--ro, req, object, 窗口播放信息-->
          <id>
            <!--ro, req, int, 播放序号-->1
          </id>
          <materialNo>
            <!--ro, opt, int, 素材编号-->1
          </materialNo>
          <playEffect opt="none,marquee">
            <!--ro, req, enum, 播放效果, subType:string, [none#无,marquee#跑马灯], attr:opt{req, string, 取值范围}-->none
          </playEffect>
          <PlayDuration>
            <!--ro, opt, object, 窗口内素材播放时间, desc:静态素材可以配置该时间,视频直播和IPC通道都可配置该节点-->
            <durationType opt="materialTime,selfDefine,unlimited">
              <!--ro, req, enum, 持续方式, subType:string, [materialTime#按素材时间,selfDefine#自定义,unlimited#无限时长], attr:opt{req, string, 取值范围}-->selfDefine
            </durationType>
            <duration min="1" max="10" default="null">
              <!--ro, req, int, 持续时间, unit:s, unitType:时间, attr:min{req, int, 最小值},max{req, int, 最大值},default{opt, string, 默认值}-->1
            </duration>
          </PlayDuration>
        </PlayItem>
      </PlayItemList>
    </Windows>
  </WindowsList>
</Page>

```

69.63.5 配置单个页面参数

Request URL

PUT /ISAPI/Publish/ProgramMgr/program/<programID>/page/<pageID>

查询参数

参数名称	参数类型	描述
programID	string	节目ID
pageID	string	页面ID

请求报文

```
<?xml version="1.0" encoding="UTF-8"?>

<Page xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, 页面, attr:version{req, string, 协议版本}-->
  <id>
    <!--req, int, 索引-->1
  </id>
  <PageBasicInfo>
    <!--req, object, 页面基本信息-->
    <pageName>
      <!--opt, string, 页面名称-->test
    </pageName>
    <BackgroundColor>
      <!--opt, object, 背景颜色-->
      <RGB>
        <!--opt, int, RGB值, desc: 十进制数据表示三原色, 如16777215表示0xFFFFFF-->1
      </RGB>
    </BackgroundColor>
    <playDurationMode>
      <!--opt, enum, 页面播放时间模式, subType:string, [selfDefine#自定义, auto#自动, loop#循环], desc: 自定义时playDuration节点有效; 自动时根据播放内容的时间
      *playCount计算: 循环时表示一直循环-->selfDefine
    </playDurationMode>
    <playDuration>
      <!--opt, int, 页面播放时长, unit:s, unitType:时间-->1
    </playDuration>
    <playCount>
      <!--opt, int, 页面播放次数-->1
    </playCount>
    <switchDuration>
      <!--opt, int, 切换间隔, unit:s, unitType:时间-->1
    </switchDuration>
    <switchEffect>
      <!--opt, enum, 切换效果, subType:string, [none#无, random#随机, boxShrink#盒状收缩, boxSpread#盒状展开, cycleShrink#圆形收缩, cycSpread#圆形展开, eraseUp#向上擦除, eraseDown#向下擦除, eraseLeft#向左擦除, eraseRight#向右擦除, verticalShelter#垂直屏蔽, horizontalShelter#水平屏蔽, verticalChessboard#纵向棋盘式, horizontalChessboard#横向棋盘式, dissolve#随机溶解, LeftRightToCenter#左右向中间缩进, ceterToLeftRight#中央向左右扩展, upDownToCenter#上下向中央缩进, centerToUpDown#中央向上下扩展, drawOutLeftDown#从左下抽出, drawOutLeftUp#从左上抽出, drawOutRightDown#从右下抽出, drawOutRightUp#从右上抽出, verticalLline#随机垂直线条, horizontalLine#随机垂直线条]-->none
    </switchEffect>
    <backgroundPic>
      <!--opt, int, 背景图片素材ID-->1
    </backgroundPic>
    <SignInCfg>
      <!--opt, object, 签到信息, desc: 当页面用于签到或考勤界面时有效-->
      <VoiceBroadcast>
        <!--opt, object, 语音播报-->
        <enabled>
          <!--opt, bool, 是否启用语音播报-->true
        </enabled>
        <broadcastSpeed>
          <!--opt, enum, 播报语速, subType:string, [sLower#慢, sLow#较慢, medium#中速, fast#较快, faster#快]-->sLower
        </broadcastSpeed>
        <inLibVoice>
          <!--opt, string, 在库人员正常体温语音播报内容-->test
        </inLibVoice>
        <outLibVoice>
          <!--opt, string, 未知人员正常体温语音播报内容-->test
        </outLibVoice>
        <inLibAbnormalTemperatureVoice>
          <!--opt, string, 在库人员异常温度语音播报内容-->test
        </inLibAbnormalTemperatureVoice>
        <outLibAbnormalTemperatureVoice>
          <!--opt, string, 未知人员异常温度语音播报内容-->test
        </outLibAbnormalTemperatureVoice>
      </VoiceBroadcast>
      <WelcomeWord>
        <!--opt, object, 欢迎词-->
        <enabled>
          <!--opt, bool, 是否启用欢迎词-->true
        </enabled>
        <inLibWord>
          <!--opt, string, 在库人员欢迎词内容-->test
        </inLibWord>
```

```
<outLibWord>
  <!--opt, string, 未知人员欢迎词内容-->test
</outLibWord>
</WelcomeWord>
<SignCartoon>
  <!--opt, object, 签到动画-->
  <signCartoonSpeed>
    <!--opt, enum, 动画速度, subType:string, [fast#快,medium#终端,slow#慢]-->fast
  </signCartoonSpeed>
</SignCartoon>
<backgroundSignEnabled>
  <!--opt, bool, 是否启用后台签到-->true
</backgroundSignEnabled>
<TemperatureDetection>
  <!--opt, object, 温度监测参数-->
  <enabled>
    <!--req, bool, 是否启用-->true
  </enabled>
  <threshold>
    <!--req, float, 正常温度值-->0.000
  </threshold>
</TemperatureDetection>
</SignInCfg>
</PageBasicInfo>
<characterMode>
  <!--opt, enum, 页面中的欢迎词模式, subType:string, [mode1#模式1,mode2#模式2,mode3#模式3], desc:门禁增加, 门禁欢迎词节目位置固定, 且存在三种模式-->mode1
</characterMode>
<WindowsList>
  <!--opt, array, 窗口信息列表, subType:object-->
  <Windows>
    <!--opt, object, 窗口信息-->
    <id>
      <!--req, int, 内容编号-->1
    </id>
    <Position>
      <!--opt, object, 内容位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
      <positionX>
        <!--req, int, x坐标-->1
      </positionX>
      <positionY>
        <!--req, int, y坐标-->1
      </positionY>
      <height>
        <!--req, int, 宽-->1
      </height>
      <width>
        <!--req, int, 高-->1
      </width>
    </Position>
    <CornerRadius>
      <!--opt, object, 圆角配置, desc:目前仅对picture图片窗口和character文字窗口有效-->
      <TopLeft>
        <!--opt, int, 左上角半径-->1
      </TopLeft>
      <TopRight>
        <!--opt, int, 右上角半径-->1
      </TopRight>
      <BottomLeft>
        <!--opt, int, 左下角半径-->1
      </BottomLeft>
      <BottomRight>
        <!--opt, int, 右下角半径-->1
      </BottomRight>
    </CornerRadius>
    <layerNo>
      <!--req, int, 图层编号-->1
    </layerNo>
    <WinMaterialInfo>
      <!--opt, object, 窗口素材信息-->
      <materialType>
        <!--req, enum, 素材类型, subType:string, [static#静态,dynamic#动态,other#其他]-->static
      </materialType>
      <staticMaterialType>
        <!--opt, enum, 静态素材类型, subType:string, [picture#图片,flash#flash,audio#音频,video#视频,document#文本,ppt#ppt,doc#文档,excel#表格,pdf#pdf,web#网页,app#app,signIn#签到,conference#会议]-->picture
      </staticMaterialType>
      <dynamicType>
        <!--opt, enum, 动态素材类型, subType:string,
[web#web,socket#socket,rss#rss,realStream#realStream,call#call,dynamicPic#dynamicPic,capturePic#capturePic,character#character,chart#图表数据]-->web
      </dynamicType>
      <otherType>
        <!--opt, enum, 其他素材类型, subType:string, [cLock#时钟,weather#天气,countdown#倒计时,LocalInput#本地输入,hyperLinkBtn#超链接按钮,event#事件,callBtn#呼叫按钮,openDoorBtn#开门按钮,QRCodeBtn#二维码按钮,authenticationSuccess#认证成功,authenticationFail#认证失败,statusBar#状态栏,ExamTemplate#考试模板]-->cLock
      </otherType>
    </WinMaterialInfo>
    <TouchProperty>
      <!--opt, object, 触摸属性-->
      <windType>
        <!--opt, enum, 窗口类型, subType:string, [popup#弹窗,page#页面]-->popup
      </windType>
      <hyperLinkType>
        <!--opt, enum, 超链接类型, subType:string, [window#页面超链接,page#窗口超链接], desc:窗口类型为弹窗时有效-->window
      </hyperLinkType>
```

```
</type></type>
<windowId>
  <!--opt, int, 窗口编号, desc:当前页面的窗口,超链接类型为窗口时有效-->1
</windowId>
<pageId>
  <!--opt, int, 页面编号, desc:超链接类型为页面时有效-->1
</pageId>
</TouchProperty>
<PlayItemList>
  <!--opt, array, 窗口播放列表, subType:object-->
  <PlayItem>
    <!--req, object, 窗口播放信息-->
    <id>
      <!--req, int, 播放序号-->1
    </id>
    <materialNo>
      <!--opt, int, 素材编号-->1
    </materialNo>
    <inputChannel>
      <!--opt, int, 绑定的IPC通道号-->1
    </inputChannel>
    <playEffect>
      <!--req, enum, 播放效果, subType:string, [none#无,marquee#跑马灯]-->none
    </playEffect>
    <MarqueeInfo>
      <!--opt, object, 跑马灯信息-->
      <scrollType>
        <!--req, enum, 跑马灯滚动类型, subType:string, [none#不滚动,Loops#循环滚动,once#滚动一次,backAndForth#来回滚动]-->none
      </scrollType>
      <scrollDeriction>
        <!--req, enum, 跑马灯滚动方向, subType:string, [none#无,up#向上,down#向下,left#向左,right#向右]-->none
      </scrollDeriction>
      <scrollSpeed>
        <!--req, int, 跑马灯滚动速度-->1
      </scrollSpeed>
    </MarqueeInfo>
    <PlayDuration>
      <!--opt, object, 窗口内素材播放时间, desc:静态素材可以配置该时间,视频直播和IPC通道都可配置该时间-->
      <durationType>
        <!--req, enum, 持续方式, subType:string, [materialTime#按素材时间,selfDefine#自定义,unlimited#无限时长]-->selfDefine
      </durationType>
      <duration>
        <!--req, int, 持续时间, unit:s, unitType:时间-->1
      </duration>
    </PlayDuration>
    <CharactersEffect>
      <!--opt, object, 文字显示效果, desc:当素材类型为文本txt和时有效-->
      <fontSize>
        <!--req, int, 字体大小-->1
      </fontSize>
      <FontColor>
        <!--req, object, 字体颜色-->
        <RGB>
          <!--opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
        </RGB>
        </FontColor>
      <backgroundType>
        <!--opt, enum, 背景类型, subType:string, [color#颜色(默认),picture#图片], desc:字段不存在时,默认为颜色-->color
      </backgroundType>
      <backPicId>
        <!--opt, int, 背景图片素材ID-->1
      </backPicId>
      <BackColor>
        <!--req, object, 背景颜色-->
        <RGB>
          <!--opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
        </RGB>
        </BackColor>
      <backTransparent>
        <!--req, int, 背景透明度-->1
      </backTransparent>
      <subtitlesEnabled>
        <!--req, bool, 子标题使能-->true
      </subtitlesEnabled>
      <scrollDirection>
        <!--req, enum, 文字滚动方向, subType:string, [left#向左,right#向右,up#向上,down#向下]-->left
      </scrollDirection>
      <scrollSpeed>
        <!--req, int, 文字滚动速度-->1
      </scrollSpeed>
    </CharactersEffect>
    <switchEffect>
      <!--opt, enum, 窗口素材切换效果, subType:string, [none#无,leftInRightOut#左进右出,rightInLeftOut#右进左出,bottomInTopOut#下进上出,topInBottomOut#上进下出,fadeInFadeOut#淡入淡出,middleExit#中间退出,topPop#顶部弹下,rightBottomIn#右下角进入,leftTopIn#左上角进入,horizontalOpen#水平打开,verticalOpen#垂直打开,random#随机效果], desc:图片素材时有效-->random
    </switchEffect>
    <pageTime>
      <!--opt, int, 翻页时间, unit:s, unitType:时间, desc:当素材为word,ppt,pdf,excel时有效-->1
    </pageTime>
    <scrollSpeed>
      <!--req, int, 滚动速度, desc:当素材为静态web时有效-->1
    </scrollSpeed>
    <scaleType>
      <!--opt, enum, 素材缩放参数配置, subType:string, [fitXY#拉伸填充,fitCenter#自适应,centerCrop#截取填充], desc:目前仅当图片窗口有效-->centerCrop
```

```

</scaleType>
<playType>
  <!--opt, enum, 播放方式, subType:string, [normal#正常,faceAttendance#考勤,PeopleCounting#客流,temperature#测温], desc:目前仅当图片窗口有效-->
  <normal>
</playType>
<CharactersAttribute>
  <!--opt, object, 文字属性, desc:当dynamicType为character时有效-->
  <fontSize>
    <!--req, int, 字体大小-->1
  </fontSize>
  <fontColor>
    <!--req, object, 字体颜色-->
    <RGB>
      <!--opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
    </RGB>
  </fontColor>
  <backgroundType>
    <!--req, enum, 背景类型, subType:string, [color#颜色,picture#图片]-->color
  </backgroundType>
  <BackColor>
    <!--req, object, 背景色-->
    <RGB>
      <!--opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
    </RGB>
  </BackColor>
  <backTransparent>
    <!--opt, int, 背景透明度-->1
  </backTransparent>
  <alignType>
    <!--req, enum, 对齐方式, subType:string, [Left#左,right#右,middle#中]-->middle
  </alignType>
  <verticalAlignType>
    <!--req, enum, 垂直对齐方式, subType:string, [top#顶,bottom#底,verticalCenter#中]-->top
  </verticalAlignType>
  <characterContent>
    <!--opt, string, 文字内容, range:[0,512]-->test
  </characterContent>
  <backPicId>
    <!--opt, int, 背景图片素材ID-->1
  </backPicId>
  <characterStyle>
    <!--req, enum, 字体效果, subType:string, [normal#常规,bold#粗体]-->normal
  </characterStyle>
  <characterType>
    <!--req, enum, 字体类型, subType:string, [mainTitle#主标题,subTitle#副标题]-->mainTitle
  </characterType>
</CharactersAttribute>
<WebProperty>
  <!--opt, object, 网页属性-->
  <webType>
    <!--opt, enum, 网页类型, subType:string, [desktop#桌面版,mobile#移动版], desc:默认按照设备的系统类型（Windows或Android）来显示网页-->desktop
  </webType>
</WebProperty>
</PlayItem>
</PlayItemList>
<enabledAudio>
  <!--opt, bool, 音频使能-->true
</enabledAudio>
<enableHide>
  <!--opt, bool, 隐藏使能-->true
</enableHide>
<enableLock>
  <!--opt, bool, 锁定使能-->true
</enableLock>
<AppWindow>
  <!--opt, object, App窗口信息-->
  <WindowInfoList>
    <!--req, array, 窗口信息列表, subType:object-->
    <WindowInfo>
      <!--req, object, 窗口信息-->
      <id>
        <!--req, int, 索引-->1
      </id>
      <materialNo>
        <!--req, int, 素材编号, desc:采用本字段下发时, 用户需要先将App安装包当做素材存储到设备中, 得到素材编号再进行下发-->1
      </materialNo>
      <path>
        <!--opt, string, App路径, range:[0,1024],
desc:path和materialNo字段同时存在时, 设备优先解析path字段;
采用本字段下发时, 设备需要确保App已经安装在设备内, 直接传入App的路径即可;
对于Android设备, path传的是App的包名; 对于Windows设备, path传的是App的应用程序绝对路径: -->test
      </path>
    </WindowInfo>
  </WindowInfoList>
</AppWindow>
<DataSource>
  <!--opt, object, 数据来源, desc:当窗口为叫号和动态弹图时有效-->
  <materialNo>
    <!--req, int, 素材编号-->1
  </materialNo>
  <dataSourceUUID>
    <!--opt, string, 数据源UUID-->1
  </dataSourceUUID>
</DataSource>

```

```

</dataSource>
<Call>
  <!--opt, object, 叫号数据, desc:对应表格数据-->
  <materialNo>
    <!--req, int, 素材编号-->1
  </materialNo>
  <tableRow>
    <!--req, int, 表格行-->1
  </tableRow>
  <tableColumn>
    <!--req, int, 表格列-->1
  </tableColumn>
  <tableDirection>
    <!--req, enum, 表格方向, subType:string, [vertical#横向,horizontal#纵向]-->vertical
  </tableDirection>
  <tableType>
    <!--req, enum, 表格类型, subType:string, [template1#模板1,template2#模板2,template3#模板3,template4#模板4,template5#模板5,template6#模板6]-->template1
  </tableType>
  <backPicId>
    <!--opt, int, 背景图片素材编号-->1
  </backPicId>
  <alignType>
    <!--req, enum, 对齐方式, subType:string, [Left#左,right#右,middle#中]-->left
  </alignType>
  <verticalAlignType>
    <!--req, enum, 垂直对齐方式, subType:string, [top#顶,bottom#底,verticalCenter#中]-->top
  </verticalAlignType>
  <refreshDirection>
    <!--req, enum, 刷新方向, subType:string, [upToDown#从上到下,downToUp#从下到上,LeftToRight#从左到右,rightToLeft#从右到左]-->upToDown
  </refreshDirection>
  <HeadDataList>
    <!--opt, array, 表头数据列表, subType:object-->
    <HeadData>
      <!--opt, object, 表头数据-->
      <id>
        <!--req, int, 索引-->1
      </id>
      <data>
        <!--req, string, 数据-->test
      </data>
    </HeadData>
  </HeadDataList>
  <ItemStyleList>
    <!--opt, array, 表格列或行样式列表, subType:object-->
    <ItemStyle>
      <!--opt, object, 表格列或行样式-->
      <id>
        <!--req, int, 索引-->1
      </id>
      <width>
        <!--req, int, 每列宽度所占百分比-->1
      </width>
      <fontSize>
        <!--req, int, 字体大小-->1
      </fontSize>
      <fontColor>
        <!--req, object, 字体颜色-->
        <RGB>
          <!--opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
        </RGB>
      </FontColor>
      <BackColor>
        <!--req, object, 背景色-->
        <RGB>
          <!--opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
        </RGB>
      </BackColor>
    </ItemStyle>
  </ItemStyleList>
  <playType>
    <!--opt, enum, 数据播放方式, subType:string, [roll#滚动播放,flipOver#翻页播放]-->roll
  </playType>
</Call>
<DynamicPic>
  <!--opt, object, 动态弹图窗口配置-->
  <materialNo>
    <!--req, int, 素材编号-->1
  </materialNo>
  <backPicId>
    <!--opt, int, 背景图片素材ID-->1
  </backPicId>
</DynamicPic>
<CapturePic>
  <!--opt, object, 抓拍图片参数-->
  <backPicId>
    <!--opt, int, 背景图片素材ID-->1
  </backPicId>
  <ipcMaterialNo>
    <!--req, int, IPC素材编号-->1
  </ipcMaterialNo>
  <cancelType>
    <!--req, enum, 取消方式, subType:string, [auto#自动>manual#手动]-->>manual
  </cancelType>

```

```
<duration>
  <!--opt, int, 持续时间 , unit:s, unitType:时间-->1
</duration>
</CapturePic>
<ClockParam>
  <!--opt, object, 时钟参数-->
  <backPicId>
    <!--opt, int, 背景图片素材ID-->1
  </backPicId>
  <ClockIcon>
    <!--req, object, 时钟图标参数-->
    <enabled>
      <!--req, bool, 使能-->true
    </enabled>
    <type>
      <!--opt, enum, 时钟类型, subType:string, [cLock1#cLock1,cLock2#cLock2]-->cLock1
    </type>
  </ClockIcon>
  <Position>
    <!--opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--req, int, x坐标-->1
    </positionX>
    <positionY>
      <!--req, int, y坐标-->1
    </positionY>
    <height>
      <!--req, int, 宽-->1
    </height>
    <width>
      <!--req, int, 高-->1
    </width>
  </Position>
</ClockIcon>
<YmdParam>
  <!--req, object, 时钟年月日参数-->
  <enabled>
    <!--req, bool, 使能-->true
  </enabled>
  <fontSize>
    <!--req, int, 字体大小-->1
  </fontSize>
  <FontColor>
    <!--req, object, 字体颜色-->
    <RGB>
      <!--opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
    </RGB>
  </FontColor>
  <BackColor>
    <!--req, object, 背景色-->
    <RGB>
      <!--opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
    </RGB>
  </BackColor>
  <Position>
    <!--opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--req, int, x坐标-->1
    </positionX>
    <positionY>
      <!--req, int, y坐标-->1
    </positionY>
    <height>
      <!--req, int, 宽-->1
    </height>
    <width>
      <!--req, int, 高-->1
    </width>
  </Position>
</YmdParam>
<HmsParam>
  <!--req, object, 时钟时分秒参数-->
  <enabled>
    <!--req, bool, 使能-->true
  </enabled>
  <fontSize>
    <!--req, int, 字体大小-->1
  </fontSize>
  <FontColor>
    <!--req, object, 字体颜色-->
    <RGB>
      <!--opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
    </RGB>
  </FontColor>
  <BackColor>
    <!--req, object, 背景色-->
    <RGB>
      <!--opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
    </RGB>
  </BackColor>
  <Position>
    <!--opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--req, int, x坐标-->1
```

```
</positionX>
<positionY>
  <!--req, int, y坐标-->1
</positionY>
<height>
  <!--req, int, 宽-->1
</height>
<width>
  <!--req, int, 高-->1
</width>
</Position>
</HmsParam>
<WeekParam>
  <!--req, object, 时钟星期参数-->
  <enabled>
    <!--req, bool, 使能-->true
  </enabled>
  <fontSize>
    <!--req, int, 字体大小-->1
  </fontSize>
  <FontColor>
    <!--req, object, 字体颜色-->
    <RGB>
      <!--opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
    </RGB>
    </FontColor>
  <BackColor>
    <!--req, object, 背景色-->
    <RGB>
      <!--opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
    </RGB>
  </BackColor>
  <Position>
    <!--opt, object, 位置, desc:左上角为坐标原点,取值范围和设备分辨率保持一致-->
    <positionX>
      <!--req, int, x坐标-->1
    </positionX>
    <positionY>
      <!--req, int, y坐标-->1
    </positionY>
    <height>
      <!--req, int, 宽-->1
    </height>
    <width>
      <!--req, int, 高-->1
    </width>
  </Position>
</WeekParam>
</ClockParam>
<WeatherParam>
  <!--opt, object, 天气参数-->
  <backPicId>
    <!--opt, int, 天气背景图片素材id-->1
  </backPicId>
  <WeatherIcon>
    <!--opt, object, 天气图标参数-->
    <enabled>
      <!--req, bool, 使能-->true
    </enabled>
    <Position>
      <!--opt, object, 位置, desc:左上角为坐标原点,取值范围和设备分辨率保持一致-->
      <positionX>
        <!--req, int, x坐标-->1
      </positionX>
      <positionY>
        <!--req, int, y坐标-->1
      </positionY>
      <height>
        <!--req, int, 宽-->1
      </height>
      <width>
        <!--req, int, 高-->1
      </width>
    </Position>
  </WeatherIcon>
  <Date>
    <!--opt, object, 天气日期参数-->
    <enabled>
      <!--req, bool, 使能-->true
    </enabled>
    <fontSize>
      <!--req, int, 字体大小-->1
    </fontSize>
    <FontColor>
      <!--req, object, 字体颜色-->
      <RGB>
        <!--opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
      </RGB>
      </FontColor>
    <BackColor>
      <!--req, object, 背景色-->
      <RGB>
        <!--opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
      </RGB>
```

```
</BackColor>
<Position>
  <!--opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
  <positionX>
    <!--req, int, x坐标-->1
  </positionX>
  <positionY>
    <!--req, int, y坐标-->1
  </positionY>
  <height>
    <!--req, int, 宽-->1
  </height>
  <width>
    <!--req, int, 高-->1
  </width>
</Position>
</Date>
<Temperature>
  <!--opt, object, 天气温度参数-->
  <enabled>
    <!--req, bool, 使能-->true
  </enabled>
  <fontSize>
    <!--req, int, 字体大小-->1
  </fontSize>
  <FontColor>
    <!--req, object, 字体颜色-->
    <RGB>
      <!--opt, int, RGB值, desc:十进制数据表示三原色, 如16777215表示0xFFFFFF-->1
    </RGB>
  </FontColor>
  <BackColor>
    <!--req, object, 背景色-->
    <RGB>
      <!--opt, int, RGB值, desc:十进制数据表示三原色, 如16777215表示0xFFFFFF-->1
    </RGB>
  </BackColor>
  <Position>
    <!--opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--req, int, x坐标-->1
    </positionX>
    <positionY>
      <!--req, int, y坐标-->1
    </positionY>
    <height>
      <!--req, int, 宽-->1
    </height>
    <width>
      <!--req, int, 高-->1
    </width>
  </Position>
</Temperature>
<WeatherContent>
  <!--opt, object, 天气情况参数-->
  <enabled>
    <!--req, bool, 使能-->true
  </enabled>
  <fontSize>
    <!--req, int, 字体大小-->1
  </fontSize>
  <FontColor>
    <!--req, object, 字体颜色-->
    <RGB>
      <!--opt, int, RGB值, desc:十进制数据表示三原色, 如16777215表示0xFFFFFF-->1
    </RGB>
  </FontColor>
  <BackColor>
    <!--req, object, 背景色-->
    <RGB>
      <!--opt, int, RGB值, desc:十进制数据表示三原色, 如16777215表示0xFFFFFF-->1
    </RGB>
  </BackColor>
  <Position>
    <!--opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--req, int, x坐标-->1
    </positionX>
    <positionY>
      <!--req, int, y坐标-->1
    </positionY>
    <height>
      <!--req, int, 宽-->1
    </height>
    <width>
      <!--req, int, 高-->1
    </width>
  </Position>
</WeatherContent>
<City>
  <!--opt, object, 天气城市参数-->
  <enabled>
    <!--req, bool, 使能-->true
```

```
</enabled>
<cityId>
  <!--req, string, 城市编号-->test
</cityId>
<cityName>
  <!--req, string, 城市名称-->test
</cityName>
<fontSize>
  <!--req, int, 字体大小-->1
</fontSize>
<FontColor>
  <!--req, object, 字体颜色-->
  <RGB>
    <!--opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
  </RGB>
</FontColor>
<BackColor>
  <!--req, object, 背景色-->
  <RGB>
    <!--opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
  </RGB>
</BackColor>
<Position>
  <!--opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
  <positionX>
    <!--req, int, x坐标-->1
  </positionX>
  <positionY>
    <!--req, int, y坐标-->1
  </positionY>
  <height>
    <!--req, int, 宽-->1
  </height>
  <width>
    <!--req, int, 高-->1
  </width>
</Position>
</City>
<Humidity>
  <!--opt, object, 天气湿度参数-->
  <enabled>
    <!--req, bool, 使能-->true
  </enabled>
  <fontSize>
    <!--req, int, 字体大小-->1
  </fontSize>
  <FontColor>
    <!--req, object, 字体颜色-->
    <RGB>
      <!--opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
    </RGB>
  </FontColor>
  <BackColor>
    <!--req, object, 背景色-->
    <RGB>
      <!--opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
    </RGB>
  </BackColor>
  <Position>
    <!--opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--req, int, x坐标-->1
    </positionX>
    <positionY>
      <!--req, int, y坐标-->1
    </positionY>
    <height>
      <!--req, int, 宽-->1
    </height>
    <width>
      <!--req, int, 高-->1
    </width>
  </Position>
</Humidity>
<AirQuality>
  <!--opt, object, 天气空气质量参数-->
  <enabled>
    <!--req, bool, 使能-->true
  </enabled>
  <fontSize>
    <!--req, int, 字体大小-->1
  </fontSize>
  <FontColor>
    <!--req, object, 字体颜色-->
    <RGB>
      <!--opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
    </RGB>
  </FontColor>
  <BackColor>
    <!--req, object, 背景色-->
    <RGB>
      <!--opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
    </RGB>
  </BackColor>
```

```
</Position>
<!--opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
<positionX>
  <!--req, int, x坐标-->1
</positionX>
<positionY>
  <!--req, int, y坐标-->1
</positionY>
<height>
  <!--req, int, 宽-->1
</height>
<width>
  <!--req, int, 高-->1
</width>
</Position>
</AirQuality>
<UpdateTime>
  <!--opt, object, 天气更新时间参数-->
  <enabled>
    <!--req, bool, 使能-->true
  </enabled>
  <refreshTime>
    <!--req, string, 刷新时间, desc:设备早期返回时间格式有误, 错为: 20200826T051500+08, 为保证XSD校验通过, 改为string类型-->20200826T051500+08
  </refreshTime>
  <updateInterval>
    <!--req, int, 刷新间隔, unit:minute-->1
  </updateInterval>
  <fontSize>
    <!--req, int, 字体大小-->1
  </fontSize>
  <FontColor>
    <!--req, object, 字体颜色-->
    <RGB>
      <!--opt, int, RGB值, desc:十进制数据表示三原色, 如16777215表示0xFFFFFF-->1
    </RGB>
  </FontColor>
  <BackColor>
    <!--req, object, 背景色-->
    <RGB>
      <!--opt, int, RGB值, desc:十进制数据表示三原色, 如16777215表示0xFFFFFF-->1
    </RGB>
  </BackColor>
  <Position>
    <!--opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--req, int, x坐标-->1
    </positionX>
    <positionY>
      <!--req, int, y坐标-->1
    </positionY>
    <height>
      <!--req, int, 宽-->1
    </height>
    <width>
      <!--req, int, 高-->1
    </width>
  </Position>
</UpdateTime>
<Wind>
  <!--opt, object, 天气风力参数-->
  <enabled>
    <!--req, bool, 使能-->true
  </enabled>
  <fontSize>
    <!--req, int, 字体大小-->1
  </fontSize>
  <FontColor>
    <!--req, object, 字体颜色-->
    <RGB>
      <!--opt, int, RGB值, desc:十进制数据表示三原色, 如16777215表示0xFFFFFF-->1
    </RGB>
  </FontColor>
  <BackColor>
    <!--req, object, 背景色-->
    <RGB>
      <!--opt, int, RGB值, desc:十进制数据表示三原色, 如16777215表示0xFFFFFF-->1
    </RGB>
  </BackColor>
  <Position>
    <!--opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--req, int, x坐标-->1
    </positionX>
    <positionY>
      <!--req, int, y坐标-->1
    </positionY>
    <height>
      <!--req, int, 宽-->1
    </height>
    <width>
      <!--req, int, 高-->1
    </width>
  </Position>
```

```
</Wind>
</WeatherParam>
<Countdown>
  <!--opt, object, 倒计时素材-->
  <endTime>
    <!--req, string, 倒计时时间点, desc:设备早期返回时间格式有误, 错为: 20200826T051500+08, 为保证XSD校验通过, 改为string类型-->20200826T051500+08
  </endTime>
  <template>
    <!--req, enum, 模板类型, subType:string, [template1#模板1,template2#模板2,openingActivity1#活动开幕1,openingActivity2#活动开幕2,commercialActivity1#商业活动1,commercialActivity2#商业活动2,colLegeEntranceExam#高考,overseasTimeFormat#海外时间格式]-->template1
  </template>
  <timeUnit>
    <!--req, enum, 时间单位, subType:string, [year#年,month#月,day#日,week#周,hour#时,minute#分,second#秒]-->year
  </timeUnit>
  <backPicId>
    <!--opt, int, 背景图片素材ID-->1
  </backPicId>
  <timeFontCfg>
    <!--opt, object, 时间字体参数-->
    <fontSize>
      <!--req, int, 字体大小-->1
    </fontSize>
    <fontColor>
      <!--req, object, 字体颜色-->
      <RGB>
        <!--opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
      </RGB>
    </fontColor>
    <position>
      <!--opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
      <positionX>
        <!--req, int, x坐标-->1
      </positionX>
      <positionY>
        <!--req, int, y坐标-->1
      </positionY>
      <height>
        <!--req, int, 宽-->1
      </height>
      <width>
        <!--req, int, 高-->1
      </width>
    </position>
  </timeFontCfg>
  <timeUnitEnabled>
    <!--opt, bool, 时间单位是否有效-->true
  </timeUnitEnabled>
  <maxTimeUnit>
    <!--req, enum, 最大时间单位, subType:string, [year#年,month#月,day#日,week#周,hour#时,minute#分,second#秒,week#周], desc:表示当前倒计时的最大显示单元,例如:maxTimeUnit为月,则1年3个月,对应月份值为15-->year
  </maxTimeUnit>
  <moduleInfoList>
    <!--opt, array, 界面模块信息列表, subType:object-->
    <moduleInfo>
      <!--opt, object, 界面模块信息-->
      <moduleType>
        <!--req, enum, 模块类型, subType:string, [year#年,month#月,day#日,hour#时,minute#分,second#秒,week#周,text#文字], desc:text为时间旁边的文字,例如年下面的文字"年"-->second
      </moduleType>
      <enabled>
        <!--req, bool, 使能-->true
      </enabled>
      <fontSize>
        <!--req, int, 字体大小-->1
      </fontSize>
      <fontColor>
        <!--req, object, 字体颜色-->
        <RGB>
          <!--opt, int, 十进制数据表示三原色-->1
        </RGB>
      </fontColor>
      <text>
        <!--opt, string, 文字, desc:当moduleType为text时有效-->test
      </text>
      <position>
        <!--opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
        <positionX>
          <!--req, int, X坐标-->1
        </positionX>
        <positionY>
          <!--req, int, Y坐标-->1
        </positionY>
        <height>
          <!--req, int, 宽-->1
        </height>
        <width>
          <!--req, int, 高-->1
        </width>
      </position>
    </moduleInfo>
  </moduleInfoList>
</Countdown>
<localInputNo>
```

```
<!--req, enum, 本地输入口类型, subType:string, [VGA#VGA,HDMI#HDMI#HDMI]-->VGA
</LocalInputNo>
<HyperlinkBtn>
  <!--opt, object, 超链接按钮-->
  <backPicId>
    <!--opt, int, 背景图片素材ID-->1
  </backPicId>
</HyperlinkBtn>
<CharactersAttribute>
  <!--opt, object, 文字属性, desc:当dynamicType为character时有效-->
  <fontSize>
    <!--req, int, 字体大小-->1
  </fontSize>
  <fontColor>
    <!--req, object, 字体颜色-->
    <RGB>
      <!--opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
    </RGB>
  </fontColor>
  <fontType>
    <!--opt, enum, 字体类型, subType:string, [normal#常规,bold#粗体]-->normal
  </fontType>
  <backgroundType>
    <!--req, enum, 背景类型, subType:string, [color#颜色,picture#图片]-->color
  </backgroundType>
  <BackColor>
    <!--req, object, 背景色-->
    <RGB>
      <!--opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
    </RGB>
  <fillEnabled>
    <!--opt, bool, 背景色填充使能, desc:默认为true, 表示背景色填充-->true
  </fillEnabled>
</BackColor>
<backTransparent>
  <!--opt, int, 背景透明度-->1
</backTransparent>
<alignType>
  <!--req, enum, 对齐方式, subType:string, [Left#左,right#右,middle#中]-->middle
</alignType>
<verticalAlignType>
  <!--req, enum, 垂直对齐方式, subType:string, [top#顶,bottom#底,verticalCenter#中]-->top
</verticalAlignType>
<characterContent>
  <!--opt, string, 文字内容, range:[0,512]-->test
</characterContent>
<backPicId>
  <!--opt, int, 背景图片素材ID-->1
</backPicId>
<characterStyle>
  <!--req, enum, 字体效果, subType:string, [normal#常规,bold#粗体]-->normal
</characterStyle>
<characterType>
  <!--req, enum, 字体类型, subType:string, [mainTitle#主标题,subTitle#副标题]-->mainTitle
</characterType>
<scrollDirection>
  <!--opt, enum, 文字滚动方向, subType:string, [Left#向左,right#向右,up#向上,down#向下,turnPage#翻页]-->left
</scrollDirection>
<scrollSpeed>
  <!--opt, int, 文字滚动速度, range:[1,10]-->1
</scrollSpeed>
</CharactersAttribute>
<SignIn>
  <!--opt, object, 人员签到展示窗口, desc:当staticMaterialType为signIn时有效-->
  <backPicId>
    <!--opt, int, 背景图片素材ID-->1
  </backPicId>
  <ShowInfo>
    <!--opt, object, 展示信息-->
    <row>
      <!--opt, int, 窗口内人员签到展示行数-->1
    </row>
    <column>
      <!--opt, int, 窗口内人员签到展示列数-->1
    </column>
    <Position>
      <!--opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
      <positionX>
        <!--req, int, x坐标-->1
      </positionX>
      <positionY>
        <!--req, int, y坐标-->1
      </positionY>
      <height>
        <!--req, int, 宽-->1
      </height>
      <width>
        <!--req, int, 高-->1
      </width>
    </Position>
  </ShowInfo>
  <inLib>
    <!--opt, object, 在库信息-->
    <nameEnabled>
      <!--opt, bool, 在库人员是否展示姓名-->true
    </nameEnabled>
  </inLib>
</SignIn>
```

```
</nameEnabled>
<genderEnabled>
  <!--opt, bool, 在库人员是否展示性别-->true
</genderEnabled>
<selfDefine1Enabled>
  <!--opt, bool, 在库人员是否展示自定义字段1-->true
</selfDefine1Enabled>
<selfDefine2Enabled>
  <!--opt, bool, 在库人员是否展示自定义字段2-->true
</selfDefine2Enabled>
<signInTimeEnabled>
  <!--opt, bool, 在库人员是否展示签到时间-->true
</signInTimeEnabled>
<temperatureEnabled>
  <!--opt, bool, 在库人员是否展示温度-->true
</temperatureEnabled>
</inLib>
<outLib>
  <!--opt, object, 不在库信息-->
  <nameEnabled>
    <!--opt, bool, 未知人员是否展示姓名-->true
  </nameEnabled>
  <name>
    <!--opt, string, 未知人员姓名-->test
  </name>
  <genderEnabled>
    <!--opt, bool, 未知人员是否展示性别-->true
  </genderEnabled>
  <signInTimeEnabled>
    <!--opt, bool, 未知人员是否展示签到时间-->true
  </signInTimeEnabled>
  <temperatureEnabled>
    <!--opt, bool, 未知人员是否展示温度-->true
  </temperatureEnabled>
</outLib>
<showDuration>
  <!--req, enum, 签到记录显示时间, subType:string, [always#不消失,10s#10s,20s#20s,30s#30s,no#不显示]-->always
</showDuration>
</ShowInfo>
<MainTitle>
  <!--opt, object, 主标题-->
  <enabled>
    <!--req, bool, 使能-->true
  </enabled>
  <fontSize>
    <!--req, int, 字体大小-->1
  </fontSize>
  <FontColor>
    <!--req, object, 字体颜色-->
    <RGB>
      <!--opt, int, 十进制数据表示三原色-->1
    </RGB>
  </FontColor>
  <Position>
    <!--opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--req, int, x坐标-->1
    </positionX>
    <positionY>
      <!--req, int, y坐标-->1
    </positionY>
    <height>
      <!--req, int, 宽-->1
    </height>
    <width>
      <!--req, int, 高-->1
    </width>
  </Position>
  <text>
    <!--opt, string, 标题文字-->test
  </text>
</MainTitle>
<SubTitle>
  <!--opt, object, 副标题-->
  <enabled>
    <!--req, bool, 使能-->true
  </enabled>
  <fontSize>
    <!--req, int, 字体大小-->1
  </fontSize>
  <FontColor>
    <!--req, object, 字体颜色-->
    <RGB>
      <!--opt, int, 十进制数据表示三原色-->1
    </RGB>
  </FontColor>
  <Position>
    <!--opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--req, int, x坐标-->1
    </positionX>
    <positionY>
      <!--req, int, y坐标-->1
    </positionY>
```

```
</positionY>
<height>
  <!--req, int, 宽-->1
</height>
<width>
  <!--req, int, 高-->1
</width>
</Position>
<text>
  <!--opt, string, 标题文字-->test
</text>
</SubTitle>
<NormalTemperature>
  <!--opt, object, 正常体温信息-->
  <PeopleNumber>
    <!--opt, object, 正常体温人数信息-->
    <enabled>
      <!--req, bool, 使能-->true
    </enabled>
    <fontSize>
      <!--req, int, 字体大小-->1
    </fontSize>
    <FontColor>
      <!--req, object, 字体颜色-->
      <RGB>
        <!--opt, int, 十进制数据表示三原色-->1
      </RGB>
    </FontColor>
    <Position>
      <!--opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
      <positionX>
        <!--req, int, X坐标-->1
      </positionX>
      <positionY>
        <!--req, int, Y坐标-->1
      </positionY>
      <height>
        <!--req, int, 宽-->1
      </height>
      <width>
        <!--req, int, 高-->1
      </width>
    </Position>
    <text>
      <!--opt, string, 标题文字-->test
    </text>
  </PeopleNumber>
  <Icon>
    <!--opt, object, 正常体温图标-->
    <enabled>
      <!--req, bool, 使能-->true
    </enabled>
    <Position>
      <!--opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
      <positionX>
        <!--req, int, X坐标-->1
      </positionX>
      <positionY>
        <!--req, int, Y坐标-->1
      </positionY>
      <height>
        <!--req, int, 宽-->1
      </height>
      <width>
        <!--req, int, 高-->1
      </width>
    </Position>
    <materialNo>
      <!--opt, int, 素材ID-->1
    </materialNo>
  </Icon>
</NormalTemperature>
<AbnormalTemperature>
  <!--opt, object, 异常体温信息-->
  <PeopleNumber>
    <!--opt, object, 异常体温人数信息-->
    <enabled>
      <!--req, bool, 使能-->true
    </enabled>
    <fontSize>
      <!--req, int, 字体大小-->1
    </fontSize>
    <FontColor>
      <!--req, object, 字体颜色-->
      <RGB>
        <!--opt, int, 十进制数据表示三原色-->1
      </RGB>
    </FontColor>
    <Position>
      <!--opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
      <positionX>
        <!--req, int, X坐标-->1
      </positionX>
      <positionY>
```

```
<!--req, int, Y坐标-->1
</positionY>
<height>
<!--req, int, 宽-->1
</height>
<width>
<!--req, int, 高-->1
</width>
</Position>
<text>
<!--opt, string, 标题文字-->test
</text>
</PeopleNumber>
<Icon>
<!--opt, object, 异常体温图标-->
<enabled>
<!--req, bool, 使能-->true
</enabled>
<Position>
<!--opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
<positionX>
<!--req, int, X坐标-->1
</positionX>
<positionY>
<!--req, int, Y坐标-->1
</positionY>
<height>
<!--req, int, 宽-->1
</height>
<width>
<!--req, int, 高-->1
</width>
</Position>
<materialNo>
<!--opt, int, 素材ID-->1
</materialNo>
</Icon>
</AbnormalTemperature>
</SignIn>
<EventParam>
<!--opt, object, 事件参数-->
<EventResourceList>
<!--opt, array, 事件源信息列表, subType:object-->
<EventResource>
<!--opt, object, 事件源信息-->
<ipVersion>
<!--req, enum, ip地址版本, subType:string, [v4#v4,v6#v6,dual#双]-->v4
</ipVersion>
<ipAddress>
<!--opt, string, ipv4地址-->test
</ipAddress>
<ipv6Address>
<!--opt, string, ipv6地址-->test
</ipv6Address>
<portNo>
<!--req, int, 端口号-->1
</portNo>
<userName>
<!--req, string, 用户名-->test
</userName>
<password>
<!--req, string, 密码-->test
</password>
<channels>
<!--opt, string, 通道号, desc:多选以逗号隔开-->1,2
</channels>
</EventResource>
</EventResourceList>
<backPicId>
<!--opt, int, 背景图片素材ID-->1
</backPicId>
<Title>
<!--opt, object, 标题-->
<enabled>
<!--req, bool, 使能-->true
</enabled>
<fontSize>
<!--req, int, 字体大小-->1
</fontSize>
<fontColor>
<!--req, object, 字体颜色-->
<RGB>
<!--opt, int, 十进制数据表示三原色-->1
</RGB>
</FontColor>
<Position>
<!--opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
<positionX>
<!--req, int, X坐标-->1
</positionX>
<positionY>
<!--req, int, Y坐标-->1
</positionY>
```

```
<height>
  <!--req, int, 宽-->1
</height>
<width>
  <!--req, int, 高-->1
</width>
</Position>
<text>
  <!--opt, string, 标题文字-->test
</text>
</Title>
<PeopleCounting>
  <!--opt, object, 人数统计提醒-->
  <maxPeopleNumber>
    <!--opt, int, 最大可进入人数-->1
  </maxPeopleNumber>
  <unavailableBackPicId>
    <!--opt, int, 人员不能进入时控件背景图片id-->1
  </unavailableBackPicId>
  <supportChangeByPeople>
    <!--req, enum, 支持根据人数修改样式的字段, subType:string, [prompt#prompt,peopleIcon#peopleIcon,availablePeopleNumber#availablePeopleNumber]--
    </supportChangeByPeople>
  </supportChangeByPeople>
<Prompt>
  <!--opt, object, 提示信息-->
  <enabled>
    <!--req, bool, 使能-->true
  </enabled>
  <fontSize>
    <!--req, int, 字体大小-->1
  </fontSize>
  <Position>
    <!--opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--req, int, X坐标-->1
    </positionX>
    <positionY>
      <!--req, int, Y坐标-->1
    </positionY>
    <height>
      <!--req, int, 宽-->1
    </height>
    <width>
      <!--req, int, 高-->1
    </width>
  </Position>
  <text>
    <!--opt, string, 标题文字-->test
  </text>
  <FontColor>
    <!--opt, object, 人员可以进入时的字体颜色-->
    <RGB>
      <!--opt, int, 十进制数据表示三原色-->1
    </RGB>
  </FontColor>
  <unavailableText>
    <!--opt, string, 人员不能进入时的标题文字-->test
  </unavailableText>
  <unavailableFontColor>
    <!--opt, object, 人员不能进入时的字体颜色-->
    <RGB>
      <!--opt, int, 十进制数据表示三原色-->1
    </RGB>
  </unavailableFontColor>
</Prompt>
<PeopleIcon>
  <!--opt, object, 行人通行图标-->
  <enabled>
    <!--req, bool, 使能-->true
  </enabled>
  <Position>
    <!--opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--req, int, X坐标-->1
    </positionX>
    <positionY>
      <!--req, int, Y坐标-->1
    </positionY>
    <height>
      <!--req, int, 宽-->1
    </height>
    <width>
      <!--req, int, 高-->1
    </width>
  </Position>
  <materialNo>
    <!--opt, int, 素材ID-->1
  </materialNo>
</PeopleIcon>
<InsidePeopleTitle>
  <!--opt, object, 已经进入人数标题-->
  <enabled>
    <!--req, bool, 使能-->true
  </enabled>
```

```
</checkbox>
<fontSize>
  <!--req, int, 字体大小-->1
</fontSize>
<fontColor>
  <!--req, object, 字体颜色-->
  <RGB>
    <!--opt, int, 十进制数据表示三原色-->1
  </RGB>
</FontColor>
<Position>
  <!--opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
  <positionX>
    <!--req, int, X坐标-->1
  </positionX>
  <positionY>
    <!--req, int, Y坐标-->1
  </positionY>
  <height>
    <!--req, int, 宽-->1
  </height>
  <width>
    <!--req, int, 高-->1
  </width>
</Position>
<text>
  <!--opt, string, 标题文字-->test
</text>
</InsidePeopleTitle>
<InsidePeopleNumber>
  <!--opt, object, 已经进入人数-->
  <enabled>
    <!--req, bool, 使能-->true
  </enabled>
  <fontSize>
    <!--req, int, 字体大小-->1
  </fontSize>
  <fontColor>
    <!--req, object, 字体颜色-->
    <RGB>
      <!--opt, int, 十进制数据表示三原色-->1
    </RGB>
  </FontColor>
  <Position>
    <!--opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--req, int, X坐标-->1
    </positionX>
    <positionY>
      <!--req, int, Y坐标-->1
    </positionY>
    <height>
      <!--req, int, 宽-->1
    </height>
    <width>
      <!--req, int, 高-->1
    </width>
  </Position>
</InsidePeopleNumber>
<AvailablePeopleTitle>
  <!--opt, object, 还可以进入人数标题-->
  <enabled>
    <!--req, bool, 使能-->true
  </enabled>
  <fontSize>
    <!--req, int, 字体大小-->1
  </fontSize>
  <fontColor>
    <!--req, object, 字体颜色-->
    <RGB>
      <!--opt, int, 十进制数据表示三原色-->1
    </RGB>
  </FontColor>
  <Position>
    <!--opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--req, int, X坐标-->1
    </positionX>
    <positionY>
      <!--req, int, Y坐标-->1
    </positionY>
    <height>
      <!--req, int, 宽-->1
    </height>
    <width>
      <!--req, int, 高-->1
    </width>
  </Position>
  <text>
    <!--opt, string, 标题文字-->test
  </text>
</AvailablePeopleTitle>
<AvailablePeopleNumber>
  <!--opt, object, 还可以进入人数-->
```

```
<enabled>
  <!--req, bool, 使能-->true
</enabled>
<fontSize>
  <!--req, int, 字体大小-->1
</fontSize>
<Position>
  <!--opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
  <positionX>
    <!--req, int, X坐标-->1
  </positionX>
  <positionY>
    <!--req, int, Y坐标-->1
  </positionY>
  <height>
    <!--req, int, 宽-->1
  </height>
  <width>
    <!--req, int, 高-->1
  </width>
</Position>
<FontColor>
  <!--opt, object, 人员可以进入时的字体颜色-->
  <RGB>
    <!--opt, int, 十进制数据表示三原色-->1
  </RGB>
</FontColor>
<UnavailableFontColor>
  <!--opt, object, 人员不能进入时的字体颜色-->
  <RGB>
    <!--opt, int, 十进制数据表示三原色-->1
  </RGB>
</UnavailableFontColor>
</AvailablePeopleNumber>
<UnavailablePeopleIcon>
  <!--opt, object, 人员不能进入时的行人图标-->
  <enabled>
    <!--req, bool, 使能-->true
  </enabled>
  <Position>
    <!--opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--req, int, X坐标-->1
    </positionX>
    <positionY>
      <!--req, int, Y坐标-->1
    </positionY>
    <height>
      <!--req, int, 宽-->1
    </height>
    <width>
      <!--req, int, 高-->1
    </width>
  </Position>
  <materialNo>
    <!--opt, int, 素材ID-->1
  </materialNo>
</UnavailablePeopleIcon>
<AudioAlarm>
  <!--opt, object, 音频报警信息-->
  <enabled>
    <!--req, bool, 使能-->true
  </enabled>
  <materialNo>
    <!--opt, int, 音频素材ID-->1
  </materialNo>
</AudioAlarm>
</PeopleCounting>
<Mask>
  <!--opt, object, 未戴口罩提醒-->
  <enabled>
    <!--req, bool, 使能-->true
  </enabled>
  <fontSize>
    <!--req, int, 字体大小-->1
  </fontSize>
  <FontColor>
    <!--req, object, 字体颜色-->
    <RGB>
      <!--opt, int, 十进制数据表示三原色-->1
    </RGB>
  </FontColor>
  <BackColor>
    <!--req, object, 背景颜色-->
    <RGB>
      <!--opt, int, 十进制数据表示三原色-->1
    </RGB>
  </BackColor>
  <Position>
    <!--opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--req, int, X坐标-->1
    </positionX>
```

```
<positionY>
  <!--req, int, Y坐标-->1
</positionY>
<height>
  <!--req, int, 宽-->1
</height>
<width>
  <!--req, int, 高-->1
</width>
</Position>
<text>
  <!--opt, string, 标题文字-->test
</text>
<MaskIcon>
  <!--opt, object, 未戴口罩图标-->
  <enabled>
    <!--req, bool, 使能-->true
  </enabled>
  <Position>
    <!--opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--req, int, X坐标-->1
    </positionX>
    <positionY>
      <!--req, int, Y坐标-->1
    </positionY>
    <height>
      <!--req, int, 宽-->1
    </height>
    <width>
      <!--req, int, 高-->1
    </width>
  </Position>
  <materialNo>
    <!--opt, int, 素材ID-->1
  </materialNo>
</MaskIcon>
<AudioAlarm>
  <!--opt, object, 音频报警信息-->
  <enabled>
    <!--req, bool, 使能-->true
  </enabled>
  <materialNo>
    <!--opt, int, 音频素材ID-->1
  </materialNo>
</AudioAlarm>
</Mask>
<Temperature>
  <!--opt, object, 温度异常提醒-->
  <Prompt>
    <!--opt, object, 提示信息-->
    <enabled>
      <!--req, bool, 使能-->true
    </enabled>
    <fontSize>
      <!--req, int, 字体大小-->1
    </fontSize>
    <FontColor>
      <!--req, object, 字体颜色-->
      <RGB>
        <!--opt, int, 十进制数据表示三原色-->1
      </RGB>
    </FontColor>
    <BackColor>
      <!--req, object, 背景颜色-->
      <RGB>
        <!--opt, int, 十进制数据表示三原色-->1
      </RGB>
    </BackColor>
  </Prompt>
  <Position>
    <!--opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--req, int, X坐标-->1
    </positionX>
    <positionY>
      <!--req, int, Y坐标-->1
    </positionY>
    <height>
      <!--req, int, 宽-->1
    </height>
    <width>
      <!--req, int, 高-->1
    </width>
  </Position>
  <text>
    <!--opt, string, 标题文字-->test
  </text>
</Prompt>
<TemperatureIcon>
  <!--opt, object, 温度异常图标-->
  <enabled>
    <!--req, bool, 使能-->true
  </enabled>
  <Position>
```

```
<!--opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
<positionX>
  <!--req, int, X坐标-->1
</positionX>
<positionY>
  <!--req, int, Y坐标-->1
</positionY>
<height>
  <!--req, int, 宽-->1
</height>
<width>
  <!--req, int, 高-->1
</width>
</Position>
<materialNo>
  <!--opt, int, 素材ID-->1
</materialNo>
</TemperatureIcon>
<AudioAlarm>
  <!--opt, object, 音频报警信息-->
  <enabled>
    <!--req, bool, 使能-->true
  </enabled>
  <materialNo>
    <!--opt, int, 音频素材ID-->1
  </materialNo>
</AudioAlarm>
<abnormalHighTemperature>
  <!--opt, float, 异常高温值, unit:℃, unitType:温度, desc:保留一位小数-->37.5
</abnormalHighTemperature>
</Temperature>
<WearMask>
  <!--opt, object, 已戴口罩提醒-->
  <enabled>
    <!--req, bool, 使能-->true
  </enabled>
  <fontSize>
    <!--req, int, 字体大小-->1
  </fontSize>
  <fontColor>
    <!--req, object, 字体颜色-->
    <RGB>
      <!--opt, int, 十进制数据表示三原色-->1
    </RGB>
  </FontColor>
  <BackColor>
    <!--req, object, 背景颜色-->
    <RGB>
      <!--opt, int, 十进制数据表示三原色-->1
    </RGB>
  </BackColor>
</Position>
  <!--opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
  <positionX>
    <!--req, int, X坐标-->1
  </positionX>
  <positionY>
    <!--req, int, Y坐标-->1
  </positionY>
  <height>
    <!--req, int, 宽-->1
  </height>
  <width>
    <!--req, int, 高-->1
  </width>
</Position>
<text>
  <!--opt, string, 标题文字-->test
</text>
<MaskIcon>
  <!--opt, object, 已戴口罩图标-->
  <enabled>
    <!--req, bool, 使能-->true
  </enabled>
  <Position>
    <!--opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--req, int, X坐标-->1
    </positionX>
    <positionY>
      <!--req, int, Y坐标-->1
    </positionY>
    <height>
      <!--req, int, 宽-->1
    </height>
    <width>
      <!--req, int, 高-->1
    </width>
  </Position>
  <materialNo>
    <!--opt, int, 素材ID-->1
  </materialNo>
</MaskIcon>
```

```
</WearMask>
<NormalTemperature>
  <!--opt, object, 温度正常提醒-->
  <Prompt>
    <!--opt, object, 提示信息-->
    <enabled>
      <!--req, bool, 使能-->true
    </enabled>
    <fontSize>
      <!--req, int, 字体大小-->1
    </fontSize>
    <FontColor>
      <!--req, object, 字体颜色-->
      <RGB>
        <!--opt, int, 十进制数据表示三原色-->1
      </RGB>
    </FontColor>
    <BackColor>
      <!--req, object, 背景颜色-->
      <RGB>
        <!--opt, int, 十进制数据表示三原色-->1
      </RGB>
    </BackColor>
    <Position>
      <!--opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
      <positionX>
        <!--req, int, X坐标-->1
      </positionX>
      <positionY>
        <!--req, int, Y坐标-->1
      </positionY>
      <height>
        <!--req, int, 宽-->1
      </height>
      <width>
        <!--req, int, 高-->1
      </width>
    </Position>
    <text>
      <!--opt, string, 标题文字-->test
    </text>
  </Prompt>
  <TemperatureIcon>
    <!--opt, object, 温度正常图标-->
    <enabled>
      <!--req, bool, 使能-->true
    </enabled>
    <Position>
      <!--opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
      <positionX>
        <!--req, int, X坐标-->1
      </positionX>
      <positionY>
        <!--req, int, Y坐标-->1
      </positionY>
      <height>
        <!--req, int, 宽-->1
      </height>
      <width>
        <!--req, int, 高-->1
      </width>
    </Position>
    <materialNo>
      <!--opt, int, 素材ID-->1
    </materialNo>
  </TemperatureIcon>
</NormalTemperature>
<NormalStatusAudioAlarm>
  <!--opt, object, 正常状态下音频报警信息, desc:人员在检测时未发现上面的人数超限、未戴口罩、体温异常等事件时设备进行播报-->
  <enabled>
    <!--req, bool, 使能-->true
  </enabled>
  <materialNo>
    <!--opt, int, 音频素材ID-->1
  </materialNo>
</NormalStatusAudioAlarm>
</EventParam>
<AlarmDataSourceProperty>
  <!--opt, object, 数据源信息, desc:该字段能力集缺少, 仅信发组件和设备使用-->
  <PictureProperty>
    <!--opt, object, 图片属性-->
    <PictureSwitchEffect>
      <!--opt, enum, 图片切换效果, subType:string, [LeftInRightOut#左进右出,rightInLeftOut#右进左出]-->LeftInRightOut
    </PictureSwitchEffect>
    <PictureScaleType>
      <!--opt, enum, 图片缩放类型, subType:string, [fitCenter#拉伸填充,centerCrop#截取填充,fixXY#适应宽高]-->fitCenter
    </PictureScaleType>
  </PictureProperty>
  <AlarmDataSource>
    <!--opt, object, 数据源-->
    <MaterialNo>
      <!--opt, int, 素材编号-->1
    </MaterialNo>
    <Name>
```

```
<!--opt, string, 素材名称-->test
</Name>
<Address>
  <!--opt, string, 素材URL地址-->test
</Address>
<WebProperty>
  <!--opt, object, 网页属性-->
  <webType>
    <!--opt, enum, 网页类型, subType:string, [desktop#桌面版,mobile#移动版], desc:默认按照设备的系统类型（Windows或Android）来显示网页-->desktop
  </webType>
</WebProperty>
</AlarmDataSource>
</AlarmDataSourceProperty>
<ConferenceParam>
  <!--opt, object, 会议信息, desc:当页面用于会议界面时有效-->
  <backPicId>
    <!--opt, int, 背景图片素材ID-->1
  </backPicId>
  <Title>
    <!--opt, object, 会议标题（主题）, desc:1、会议标题（主题）的文字内容通过场地预约配置协议下发,对应/ISAPI/Publish/Signage/placeReservation?
format=json中的programID字段。2、可通过会议计划（/ISAPI/InteractivePanel/Meeting/plan?format=json）中的meetingTopic字段下发-->
    <enabled>
      <!--req, bool, 使能-->true
    </enabled>
    <fontSize>
      <!--req, int, 字体大小-->1
    </fontSize>
    <fontColor>
      <!--req, object, 字体颜色-->
      <RGB>
        <!--opt, int, 十进制数据表示三原色-->1
      </RGB>
    </fontColor>
    <BackColor>
      <!--req, object, 背景颜色-->
      <RGB>
        <!--opt, int, 十进制数据表示三原色-->1
      </RGB>
    </BackColor>
    <Position>
      <!--opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
      <positionX>
        <!--req, int, X坐标-->1
      </positionX>
      <positionY>
        <!--req, int, Y坐标-->1
      </positionY>
      <height>
        <!--req, int, 宽-->1
      </height>
      <width>
        <!--req, int, 高-->1
      </width>
    </Position>
    <alignType>
      <!--req, enum, 对齐方式, subType:string, [Left#左,right#右,middle#中]-->middle
    </alignType>
  </Title>
</VoiceBroadcast>
<!--opt, object, 语音播报-->
<enabled>
  <!--opt, bool, 是否启用语音播报-->true
</enabled>
<broadcastSpeed>
  <!--opt, enum, 播报语速, subType:string, [slower#慢,slow#较慢,medium#中速,fast#较快,faster#快]-->slower
</broadcastSpeed>
<inLibVoice>
  <!--opt, string, 在库人员语音播报内容-->test
</inLibVoice>
<outLibVoice>
  <!--opt, string, 未知人员语音播报内容-->test
</outLibVoice>
</VoiceBroadcast>
<WelcomeWord>
  <!--opt, object, 欢迎词-->
  <enabled>
    <!--opt, bool, 是否启用欢迎词-->true
  </enabled>
  <inLibWord>
    <!--opt, string, 在库人员欢迎词内容, range:[1,128]-->test
  </inLibWord>
  <outLibWord>
    <!--opt, string, 未知人员欢迎词内容, range:[1,128]-->test
  </outLibWord>
</WelcomeWord>
<ShowInfo>
  <!--opt, object, 展示信息, desc:下发该字段需要人脸签到, 不下发该字段, 不需要人脸签到-->
  <row>
    <!--opt, int, 窗口内人员签到展示行数-->1
  </row>
  <column>
    <!--opt, int, 窗口内人员签到展示列数-->1
  </column>
```

```

<Position>
  <!--opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
  <positionX>
    <!--req, int, x坐标-->1
  </positionX>
  <positionY>
    <!--req, int, y坐标-->1
  </positionY>
  <height>
    <!--req, int, 宽-->1
  </height>
  <width>
    <!--req, int, 高-->1
  </width>
</Position>
<inLib>
  <!--opt, object, 在库信息-->
  <nameEnabled>
    <!--opt, bool, 在库人员是否展示姓名-->true
  </nameEnabled>
</inLib>
<defaultShowInfoEnabled>
  <!--opt, bool, 是否采用人员默认展示信息, desc:下发该字段, 且为true, 表示人员展示信息采用默认值。坐标信息等不用下发-->true
</defaultShowInfoEnabled>
<ShowInfo>
<ModuleInfoList>
  <!--opt, array, 界面模块信息列表, subType:object-->
  <ModuleInfo>
    <!--opt, object, 界面模块信息-->
    <moduleType>
      <!--req, enum, 模块类型, subType:string, [conferenceTime#会议时间(对应timeRange), conferenceSite#会议地点(对应会议室名称meetingName), conferenceStatus#会议状态(对应meetingStatus), signInPersonNum#已签到人数, totalPersonNum#应到人数, unSignInPersonNum#未签到人数, signInAndTotalPersonNum#已签到/应到人数, QRCode#会议室二维码, meetingHostName#主持人, employeeNo#参会人, customKey#自定义字段, nextMeeting#当天下一场会议信息]-->conferenceTime
    </moduleType>
    <enabled>
      <!--req, bool, 使能-->true
    </enabled>
    <fontSize>
      <!--req, int, 字体大小-->1
    </fontSize>
    <FontColor>
      <!--req, object, 字体颜色-->
      <RGB>
        <!--opt, int, 十进制数据表示三原色-->1
      </RGB>
    </FontColor>
    <BackColor>
      <!--opt, object, 背景颜色-->
      <RGB>
        <!--opt, int, 十进制数据表示三原色-->1
      </RGB>
    </BackColor>
    <backPicId>
      <!--opt, int, 背景图片素材ID, desc:背景图和背景颜色同时仅支持一种, 两者均下发, 设备优先解析背景图字段-->1
    </backPicId>
    <Position>
      <!--opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
      <positionX>
        <!--req, int, X坐标-->1
      </positionX>
      <positionY>
        <!--req, int, Y坐标-->1
      </positionY>
      <height>
        <!--req, int, 宽-->1
      </height>
      <width>
        <!--req, int, 高-->1
      </width>
    </Position>
    <alignType>
      <!--req, enum, 对齐方式, subType:string, [Left#左, right#右, middle#中]-->middle
    </alignType>
    <CustomKey>
      <!--opt, object, 自定义字段-->
      <key>
        <!--opt, string, 自定义字段-->test
      </key>
      <associateProtocolURL>
        <!--opt, string, 关联协议URL-->test
      </associateProtocolURL>
    </CustomKey>
  </ModuleInfo>
</ModuleInfoList>
<conferenceStatus>
  <!--opt, enum, 会议状态, subType:string, [idle#空闲, signing#签到中, meeting#开会中], desc:根据不同的会议状态, 配置不同的会议界面-->idle
</conferenceStatus>
<BackColor>
  <!--opt, object, 背景颜色-->
  <RGB>
    <!--opt, int, 十进制数据表示三原色-->1
  </RGB>
</BackColor>

```

```

</backPicId>
</ConferenceParam>
<CallBtn>
  <!--opt, object, 呼叫按键窗口-->
  <backPicId>
    <!--opt, int, 背景图片素材ID-->1
  </backPicId>
</CallBtn>
<OpenDoorBtn>
  <!--opt, object, 开门按键窗口-->
  <backPicId>
    <!--opt, int, 背景图片素材ID-->1
  </backPicId>
</OpenDoorBtn>
<QRCodeBtn>
  <!--opt, object, 二维码按键窗口-->
  <backPicId>
    <!--opt, int, 背景图片素材ID-->1
  </backPicId>
</QRCodeBtn>
<AuthenticationSuccess>
  <!--opt, object, 认证成功窗口-->
  <BackColor>
    <!--opt, object, 背景颜色-->
    <RGB>
      <!--opt, int, 十进制数据表示三原色-->1
    </RGB>
  </BackColor>
  <backPicId>
    <!--opt, int, 背景图片素材ID, desc: 背景图和背景颜色同时仅支持一种, 两者均下发, 设备优先解析背景图字段-->1
  </backPicId>
  <SubWindowList>
    <!--opt, object, 子窗口列表-->
    <SubWindow>
      <!--opt, object, 子窗口-->
      <subWindowType>
        <!--req, enum, 子窗口类型, subType:string, [temperature#温度, facePic#人脸图片, name#姓名, employeeNo#工号, group#组织/部门, tips#提示信息]-->temperature
      </subWindowType>
      <enabled>
        <!--req, bool, 使能-->true
      </enabled>
      <fontSize>
        <!--opt, int, 字体大小, unit:px, unitType:图像显示-->1
      </fontSize>
      <fontColor>
        <!--opt, int, 字体颜色-->1
      </fontColor>
      <fontStyle>
        <!--opt, string, 字体风格, desc:bold-加粗, italic-倾斜, underline-下划线, strikethrough-删除线-->bold,italic,underline,strikeout
      </fontStyle>
      <Position>
        <!--opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
        <positionX>
          <!--req, int, x坐标-->1
        </positionX>
        <positionY>
          <!--req, int, y坐标-->1
        </positionY>
        <height>
          <!--req, int, 宽-->1
        </height>
        <width>
          <!--req, int, 高-->1
        </width>
      </Position>
    </SubWindow>
  </SubWindowList>
</AuthenticationSuccess>
<AuthenticationFail>
  <!--opt, object, 认证失败窗口-->
  <BackColor>
    <!--opt, object, 背景颜色-->
    <RGB>
      <!--opt, int, 十进制数据表示三原色-->1
    </RGB>
  </BackColor>
  <backPicId>
    <!--opt, int, 背景图片素材ID, desc: 背景图和背景颜色同时仅支持一种, 两者均下发, 设备优先解析背景图字段-->1
  </backPicId>
  <SubWindowList>
    <!--opt, object, 子窗口列表-->
    <SubWindow>
      <!--opt, object, 子窗口-->
      <subWindowType>
        <!--req, enum, 子窗口类型, subType:string, [temperature#温度, facePic#人脸图片, name#姓名, employeeNo#工号, group#组织/部门, tips#提示信息]-->temperature
      </subWindowType>
      <enabled>
        <!--req, bool, 使能-->true
      </enabled>
      <fontSize>
        <!--opt, int, 字体大小, unit:px, unitType:图像显示-->1
      </fontSize>

```

```

<fontColor>
  <!--opt, int, 字体颜色-->1
</fontColor>
<fontStyle>
  <!--opt, string, 字体风格, desc:bold-加粗, italic-倾斜, underline-下划线, strikeout-删除线-->bold,italic,underline,strikeout
</fontStyle>
<Position>
  <!--opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
  <positionX>
    <!--req, int, x坐标-->1
  </positionX>
  <positionY>
    <!--req, int, y坐标-->1
  </positionY>
  <height>
    <!--req, int, 宽-->1
  </height>
  <width>
    <!--req, int, 高-->1
  </width>
</Position>
<facePicID>
  <!--ro, opt, int, 默认人脸图片ID, step:1, dep:and,
{$.PageList.Page.WindowsList[*].Windows.AuthenticationFail.SubWindowList.SubWindow.subWindowType,eq,facePic}-->1
  </facePicID>
</SubWindow>
</SubWindowList>
</AuthenticationFail>
<StatusBar>
  <!--opt, object, 状态栏, desc:当素材类型为状态栏时有效, 且必填-->
  <backPicId>
    <!--opt, int, 背景图片素材ID, desc:背景图和背景颜色同时仅支持一种, 两者均下发, 设备优先解析背景图字段-->1
  </backPicId>
  <statusBarType>
    <!--req, enum, 状态栏类型, subType:string, [fixed#固定的], desc:当为固定类型时, 窗口位置不生效, 设备将状态栏固定在顶部显示-->fixed
  </statusBarType>
  <SubStatusList>
    <!--opt, object, 子状态列表-->
    <SubStatus>
      <!--opt, object, 子状态-->
      <statusType>
        <!--req, enum, 子状态, subType:string, [dial#拨号状态(3G4G等),wifi#wifi状态,EZVIZ#萤石状态,wiredNetwork#有线网络状态,deploy#布防状态,temperature#测温]-->dial
      </statusType>
      <enabled>
        <!--req, bool, 是否显示-->true
      </enabled>
    </SubStatus>
  </SubStatusList>
</StatusBar>
<ExamTemplate>
  <!--opt, object, 考试模板, desc:当页面用于会议界面时有效-->
  <examTemplateID>
    <!--opt, string, 模板ID-->test
  </examTemplateID>
  <backPicURL>
    <!--opt, string, 背景图片素材URL, range:[1,1024]-->test
  </backPicURL>
  <backgroundDirection>
    <!--opt, enum, 背景图方向, subType:string, [horizontal#水平,vertical#垂直]-->horizontal
  </backgroundDirection>
  <ModuleInfoList>
    <!--opt, array, 界面模块信息列表, subType:object-->
    <ModuleInfo>
      <!--opt, object, 界面模块信息-->
      <moduleType>
        <!--opt, enum, 模块类型, subType:string, [schoolName#学校名称,currentTime#当前时间,examinationPlanLable#考试计划标签,examinationPlanName#考试计划名称,examRoomNumberLable#考场号标签,examRoomNumber#考场号,examRoomAddress#考试地点examRoomAddressLable,examDescriptionLable#考试说明标签,examDescription#考试说明,examSubjectLable#考试科目标签,examSubject#考试科目,examTimeLable#考试时间标签,examTime#考试时间,teacherLable#监考老师标签,teacher#监考老师,teacherLable#监考老师标签,teacher#监考老师,examNumberRangeLable#考号范围,examNumberRange#考号范围]-->schoolName
      </moduleType>
      <enabled>
        <!--opt, bool, 使能-->true
      </enabled>
    </ModuleInfo>
  </ModuleInfoList>
  <Position>
    <!--opt, object, 位置, desc:左上角为坐标原点-->
    <positionX>
      <!--req, int, X坐标-->1
    </positionX>
    <positionY>
      <!--req, int, Y坐标-->1
    </positionY>
    <height>
      <!--req, int, 高-->1
    </height>
    <width>
      <!--req, int, 宽-->1
    </width>
  </Position>
  <fontSize>
    <!--opt, int, 字体大小-->1
  </fontSize>
  <fontStyleList>

```

```

<!--opt, array, 子节点风格列表, subtype:object-->
<fontStyle>
  <!--opt, enum, 字体风格, subtype:string, [bold#加粗,italic#倾斜,regular#常规,strikeThrough#中划线,underline#下划线]-->bold
</fontStyle>
</fontStyleList>
<fontType>
  <!--opt, enum, 字体类型, subtype:string, [custom#自定义,default#默认,kaiTi#楷体,simHei#黑体,simSun#宋体]-->custom
</fontType>
<fontColor>
  <!--opt, object, 字体颜色-->
  <RGB>
    <!--opt, int, 十进制数据表示三原色-->1
  </RGB>
</fontColor>
<fontTransparent>
  <!--opt, int, 字体透明度-->1
</fontTransparent>
<alignType>
  <!--opt, enum, 字体对齐方式, subtype:string, [left#左,right#右,middle#中]-->middle
</alignType>
<characterSpacing>
  <!--opt, int, 字符间距-->1
</characterSpacing>
<lineSpacing>
  <!--opt, int, 行间距-->1
</lineSpacing>
<layerNo>
  <!--opt, int, 图层编号, desc:数字越大, 表示在越上层-->1
</layerNo>
</ModuleInfo>
</ModuleInfoList>
</ExamTemplate>
<ChartData>
  <!--opt, object, 图表数据-->
  <materialNo>
    <!--req, int, 素材编号-->1
  </materialNo>
  <chartType>
    <!--opt, enum, 图表类型, subtype:string, [histogram#柱状图,pieChart#饼图]-->histogram
  </chartType>
  <backPicId>
    <!--opt, int, 背景图片素材编号-->1
  </backPicId>
  <HeadDataList>
    <!--opt, array, 表头数据列表, subtype:object-->
    <HeadData>
      <!--opt, object, 表头数据-->
      <id>
        <!--req, int, 索引-->1
      </id>
      <data>
        <!--req, string, 数据-->test
      </data>
      <headType>
        <!--opt, enum, 表头类型, subtype:string, [axis#轴,value#值]-->axis
      </headType>
    </HeadData>
  </HeadDataList>
</ChartData>
</Windows>
</WindowsList>
</Page>

```

响应报文

```

<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 响应消息, attr:version{ro, req, string, 版本}-->
  <requestURL>
    <!--ro, req, string, 请求的URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, 状态码, subtype:int, [0#OK,1#OK,2#Device Busy,3#Device Error,4#Invalid Operation,5#Invalid XML Format,6#Invalid XML Content,7#Reboot Required]-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, 状态信息, subtype:string, [OK#成功,Device Busy#设备忙,Device Error#设备异常,Invalid Operation#无效的操作,Invalid XML Format#无效的XML格式,Invalid XML Content#无效的XML内容,Reboot#设备重启]-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, 详细错误码英文描述, desc:详细错误码的英文描述-->OK
  </subStatusCode>
</ResponseStatus>

```

69.63.6 删除单个页面参数

Request URL

DELETE /ISAPI/Publish/ProgramMgr/program/<programID>/page/<pageID>

查询参数

参数名称	参数类型	描述
programID	string	节目ID
pageID	string	页面ID

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 响应消息, attr:version{ro, req, string, 版本}-->
  <requestURL>
    <!--ro, req, string, 请求的URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, 状态码, subType:int, [0#OK,1#OK,2#Device Busy,3#Device Error,4#Invalid Operation,5#Invalid XML Format,6#Invalid XML Content,7#Reboot Required]-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, 状态信息, subType:string, [OK#成功,Device Busy#设备忙,Device Error#设备异常,Invalid Operation#无效的操作,Invalid XML Format#无效的XML格式,Invalid XML Content#无效的XML内容,Reboot#设备重启]-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, 详细错误码英文描述, desc:详细错误码的英文描述-->OK
  </subStatusCode>
</ResponseStatus>
```

69.64 日程管理

69.64.1 创建单个日程

Request URL

POST /ISAPI/Publish/ScheduleMgr/playSchedule

查询参数

无

请求报文

```
<?xml version="1.0" encoding="UTF-8"?>

<PlaySchedule xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--opt, object, 播放日程, attr:version{req, string, 协议版本}-->
  <id>
    <!--req, int, 日程索引-->1
  </id>
  <scheduleName>
    <!--req, string, 日程名称-->test
  </scheduleName>
  <scheduleRemarks>
    <!--opt, string, 日程描述-->test
  </scheduleRemarks>
  <approveState>
    <!--opt, enum, 审核状态标志, subType:string, [approved#审核通过,notPass#未通过,notApprove#未审核]-->approved
  </approveState>
  <approveRemarks>
    <!--opt, string, 审核意见-->test
  </approveRemarks>
  <scheduleMode>
    <!--opt, enum, 日程模式, subType:string, [normal#标准,decode#解码,touch#触摸,decodeTouch#解码触摸]-->normal
  </scheduleMode>
  <orgNo>
    <!--opt, int, 所属组织编号-->1
  </orgNo>
  <scheduleType>
    <!--opt, enum, 日程类型, subType:string, [selfDefine#自定义日程,daily#日日程,weekly#周日程,loop#循环日程,defaultSchedule#垫片日程,emergency#紧急日程], desc:当日程为紧急日程时, 下发后并不立即播放,在出现紧急情况时(如火灾、地震等), 平台选择该日程下对应的紧急节目进行播放, 对应接口: /ISAPI/Publish/ProgramMgr/programPlay?format=json. 播放优先级高于其他日程和插播. 紧急情况消除后, 平台停止紧急日程的播放. 设备恢复其他日程或插播的播放. ]-->selfDefine
  </scheduleType>
</PlaySchedule>
```

```
</scheduleType>
<shareProperty>
  <!--opt, enum, 共享属性, subType:string, [public#公有,private#私有]-->public
</shareProperty>
<DailySchedule>
  <!--opt, object, 日日程-->
  <PlaySpanList>
    <!--req, array, 播放计划列表, subType:object-->
    <PlaySpan>
      <!--opt, object, 播放计划-->
      <id>
        <!--req, int, 时间段序号-->1
      </id>
      <programNo>
        <!--req, int, 某个时间段播放的节目索引号-->1
      </programNo>
      <programUUID>
        <!--no, opt, string, 节目UUID, range:[1,32], desc:平台下发-->test
      </programUUID>
      <TimeRange>
        <!--req, object, 时间段起始点和结束点-->
        <beginTime>
          <!--req, string, 开始时间, desc:设备最早实现错误（格式为20200826T051500+08）,暂不修改-->test
        </beginTime>
        <endTime>
          <!--req, string, 结束时间, desc:设备最早实现错误（格式为20200826T051500+08）,暂不修改-->test
        </endTime>
      </TimeRange>
    </PlaySpan>
  </PlaySpanList>
  <PlaySpanSecondList>
    <!--opt, array, 双屏节目中辅屏节目的播放计划列表, subType:object, desc:新增该字段是为了兼容双屏节目下发到单屏终端的情况,保证单屏终端可以播放主屏节目;同时也是考虑到主辅屏的播放计划不同的情况,下面带second的字段同理-->
    <PlaySpanSecond>
      <!--opt, object, 辅屏播放计划-->
      <id>
        <!--req, int, 时间段序号-->1
      </id>
      <programNo>
        <!--req, int, 某个时间段播放的节目索引号-->1
      </programNo>
      <programUUID>
        <!--no, opt, string, 节目UUID, range:[1,32], desc:平台下发-->test
      </programUUID>
      <TimeRange>
        <!--req, object, 时间段起始点和结束点-->
        <beginTime>
          <!--req, string, 开始时间, desc:设备最早实现错误（格式为20200826T051500+08）,暂不修改-->test
        </beginTime>
        <endTime>
          <!--req, string, 结束时间, desc:设备最早实现错误（格式为20200826T051500+08）,暂不修改-->test
        </endTime>
      </TimeRange>
    </PlaySpanSecond>
  </PlaySpanSecondList>
</DailySchedule>
<WeeklySchedule>
  <!--opt, array, 周播放计划, subType:object-->
  <DayList>
    <!--req, object, 日列表-->
    <Day>
      <!--opt, object, 日信息-->
      <id>
        <!--req, int, 天序号-->1
      </id>
      <dayOfWeek>
        <!--req, enum, 星期, subType:string, [monday#星期一,tuesday#星期二,wednesday#星期三,thursday#星期四,friday#星期五,saturday#星期六,sunday#星期日]-->
monday
      </dayOfWeek>
    </Day>
  </DayList>
  <PlaySpanList>
    <!--req, array, 播放计划列表, subType:object-->
    <PlaySpan>
      <!--opt, object, 播放计划-->
      <id>
        <!--req, int, 时间段序号-->1
      </id>
      <programNo>
        <!--req, int, 某个时间段播放的节目索引号-->1
      </programNo>
      <programUUID>
        <!--no, opt, string, 节目UUID, range:[1,32], desc:平台下发-->test
      </programUUID>
      <TimeRange>
        <!--req, object, 时间段起始点和结束点-->
        <beginTime>
          <!--req, string, 开始时间, desc:设备最早实现错误（格式为20200826T051500+08）,暂不修改-->test
        </beginTime>
        <endTime>
          <!--req, string, 结束时间, desc:设备最早实现错误（格式为20200826T051500+08）,暂不修改-->test
        </endTime>
      </TimeRange>
    </PlaySpan>
  </PlaySpanList>
  <PlaySpanSecondList>
```

<!--opt, array, 双屏节目中辅屏节目的播放计划列表, subType:object, desc:新增该字段是为了兼容双屏节目下发到单屏终端的情况, 保证单屏终端可以播放主屏节目; 同时也是考虑到主辅屏的播放计划不同的情况, 下面带second的字段同理-->

<PlaySpanSecond>

<!--opt, object, 辅屏播放计划-->

<id>

<!--req, int, 时间段序号-->1

</id>

<programNo>

<!--req, int, 某个时间段播放的节目索引号-->1

</programNo>

<programUUID>

<!--ro, opt, string, 节目UUID, range:[1,32], desc:平台下发-->test

</programUUID>

<TimeRange>

<!--req, object, 时间段起始点和结束点-->

<beginTime>

<!--req, string, 开始时间, desc:设备最早实现错误(格式为20200826T051500+08), 暂不修改-->test

</beginTime>

<endTime>

<!--req, string, 结束时间, desc:设备最早实现错误(格式为20200826T051500+08), 暂不修改-->test

</endTime>

</TimeRange>

</PlaySpanSecond>

</PlaySpanSecondList>

</Day>

</DayList>

</WeeklySchedule>

<LoopSchedule>

<!--opt, object, 循环计划-->

<ProgramList>

<!--opt, object, 主屏节目-->

<program>

<!--opt, object, 播放计划-->

<id>

<!--req, int, 时间段序号-->08

</id>

<programNo>

<!--req, int, 某个时间段播放的节目索引号-->1

</programNo>

<programUUID>

<!--ro, opt, string, 节目UUID, range:[1,32], desc:平台下发-->test

</programUUID>

</program>

</ProgramList>

<ProgramNoList>

<!--req, array, 循环播放的节目列表, subType:object, desc:普通模式(适合单屏节目或双屏节目中主屏节目)-->

<programNo>

<!--opt, int, 节目编号-->1

</programNo>

</ProgramNoList>

<ProgramNoSecondList>

<!--req, array, 循环播放的辅屏节目列表, subType:object, desc:普通模式(双屏节目中辅屏节目的列表)-->

<programNoSecond>

<!--opt, int, 节目编号-->1

</programNoSecond>

</ProgramNoSecondList>

<ProgramSecondList>

<!--opt, object, 副屏节目-->

<program>

<!--opt, object, 播放计划-->

<id>

<!--req, int, 时间段序号-->08

</id>

<programNo>

<!--req, int, 某个时间段播放的节目索引号-->1

</programNo>

<programUUID>

<!--ro, opt, string, 节目UUID, range:[1,32], desc:平台下发-->test

</programUUID>

</program>

</ProgramSecondList>

<LoopTimeSpanList>

<!--opt, array, 时间段轮播列表, subType:object, desc:轮播中配置了普通模式, 时间段模式最多只能配置7个时间段(适合单屏节目或双屏节目中主屏节目的轮播计划)-->

<LoopTimeSpan>

<!--opt, object, 时间段轮播-->

<TimeRange>

<!--req, object, 时间段起始点和结束点-->

<beginTime>

<!--req, string, 开始时间, desc:设备最早实现错误(格式为20200826T051500+08), 暂不修改-->test

</beginTime>

<endTime>

<!--req, string, 结束时间, desc:设备最早实现错误(格式为20200826T051500+08), 暂不修改-->test

</endTime>

</TimeRange>

<ProgramNoList>

<!--req, array, 节目编号列表, subType:object-->

<programNo>

<!--opt, int, 节目编号-->1

</programNo>

</ProgramNoList>

<ProgramList>

<!--opt, object, 主屏节目-->

```
<program>
  <!--opt, object, 播放计划-->
  <id>
    <!--req, int, 时间段序号-->08
  </id>
  <programNo>
    <!--req, int, 某个时间段播放的节目索引号-->1
  </programNo>
  <programUUID>
    <!--ro, opt, string, 节目UUID, range:[1,32], desc:平台下发-->test
  </programUUID>
</program>
</ProgramList>
</LoopTimeSpan>
</LoopTimeSpanList>
<LoopTimeSpanSecondList>
  <!--opt, array, 时间段辅屏轮播列表, subType:object-->
  <LoopTimeSpanSecond>
    <!--opt, object, 时间段辅屏轮播-->
    <TimeRange>
      <!--req, object, 时间段起始点和结束点-->
      <beginTime>
        <!--req, string, 开始时间, desc:设备最早实现错误（格式为20200826T051500+08）, 暂不修改-->test
      </beginTime>
      <endTime>
        <!--req, string, 结束时间, desc:设备最早实现错误（格式为20200826T051500+08）, 暂不修改-->test
      </endTime>
    </TimeRange>
    <ProgramNoSecondList>
      <!--req, array, 节目编号列表, subType:object-->
      <programNoSecond>
        <!--opt, int, 节目编号-->1
      </programNoSecond>
    </ProgramNoSecondList>
    <ProgramSecondList>
      <!--opt, object, 副屏节目-->
      <program>
        <!--opt, object, 播放计划-->
        <id>
          <!--req, int, 时间段序号-->08
        </id>
        <programNo>
          <!--req, int, 某个时间段播放的节目索引号-->1
        </programNo>
        <programUUID>
          <!--ro, opt, string, 节目UUID, range:[1,32], desc:平台下发-->test
        </programUUID>
      </program>
    </ProgramSecondList>
  </LoopTimeSpanSecond>
</LoopTimeSpanSecondList>
</LoopSchedule>
<SelfDefineSchedule>
  <!--opt, object, 自定义日程-->
  <SelfDefineList>
    <!--req, array, 自定义计划列表, subType:object-->
    <SelfDefine>
      <!--opt, object, 自定义计划-->
      <id>
        <!--req, int, 自定义时间段序号-->1
      </id>
      <programNo>
        <!--req, int, 节目编号-->1
      </programNo>
      <programUUID>
        <!--ro, opt, string, 节目UUID, range:[1,32], desc:平台下发-->test
      </programUUID>
      <TimeRange>
        <!--req, object, 时间段起始点和结束点-->
        <beginTime>
          <!--req, string, 开始时间, desc:设备最早实现错误（格式为20200826T051500+08）, 暂不修改-->test
        </beginTime>
        <endTime>
          <!--req, string, 结束时间, desc:设备最早实现错误（格式为20200826T051500+08）, 暂不修改-->test
        </endTime>
      </TimeRange>
    </SelfDefine>
  </SelfDefineList>
  <SelfDefineSecondList>
    <!--opt, array, 辅屏自定义计划列表, subType:object-->
    <SelfDefineSecond>
      <!--opt, object, 辅屏自定义计划-->
      <id>
        <!--req, int, 自定义时间段序号-->1
      </id>
      <programNoSecond>
        <!--req, int, 节目编号-->1
      </programNoSecond>
      <programUUIDSecond>
        <!--ro, opt, string, 节目UUID, range:[1,32], desc:平台下发-->test
      </programUUIDSecond>
      <TimeRange>
        <!--req, object, 时间段起始点和结束点-->
        <beginTime>
```

```
<!--req, string, 开始时间, desc:设备最早实现错误 (格式为20200826T051500+08), 暂不修改-->test
</beginTime>
<endTime>
<!--req, string, 结束时间, desc:设备最早实现错误 (格式为20200826T051500+08), 暂不修改-->test
</endTime>
</TimeRange>
</SelfDefineSecond>
</SelfDefineSecondList>
</SelfDefineSchedule>
<DefaultSchedule>
<!--opt, object, 垫片日程-->
<programNo>
<!--req, int, 垫片节目-->1
</programNo>
<programUUID>
<!--ro, opt, string, 节目UUID, range:[1,32], desc:平台下发-->test
</programUUID>
<programNoSecond>
<!--opt, int, 辅屏垫片节目-->1
</programNoSecond>
<programUUIDSecond>
<!--ro, opt, string, 节目UUID, range:[1,32], desc:平台下发-->test
</programUUIDSecond>
</DefaultSchedule>
<HolidaySchedule>
<!--opt, object, 假日计划信息, desc:假日计划为可选节点, 可以附加在日计划和周计划日程里 (假日计划新的终端均已不在支持, 因此该部分协议不进行双屏的扩展) -->
<!--req, array, 播放计划列表, subType:object-->
<PlaySpan>
<!--opt, object, 播放计划-->
<id>
<!--req, int, 时间段编号-->1
</id>
<programNo>
<!--req, int, 节目索引号-->1
</programNo>
<programUUID>
<!--ro, opt, string, 节目UUID, range:[1,32], desc:平台下发-->test
</programUUID>
<TimeRange>
<!--req, object, 时间段起始点和结束点-->
<beginTime>
<!--req, string, 开始时间, desc:设备最早实现错误 (格式为20200826T051500+08), 暂不修改-->test
</beginTime>
<endTime>
<!--req, string, 结束时间, desc:设备最早实现错误 (格式为20200826T051500+08), 暂不修改-->test
</endTime>
</TimeRange>
</PlaySpan>
</PlaySpanList>
</HolidaySchedule>
<scheduleUUID>
<!--ro, opt, string, 日程UUID, range:[1,32], desc:平台返回-->test
</scheduleUUID>
<EmergencySchedule>
<!--opt, object, 紧急日程-->
<ProgramNoList>
<!--req, array, 紧急日程播放的节目列表, subType:object, desc:普通模式 (适合单屏节目或双屏节目中主屏节目) -->
<programNo>
<!--opt, int, 节目编号-->1
</programNo>
<programUUID>
<!--ro, opt, string, 节目UUID, range:[1,32], desc:平台下发-->test
</programUUID>
</ProgramNoList>
<ProgramNoSecondList>
<!--req, array, 紧急日程播放的辅屏节目列表, subType:object, desc:普通模式 (双屏节目中辅屏节目的列表) -->
<programNoSecond>
<!--opt, int, 节目编号-->1
</programNoSecond>
<programUUID>
<!--ro, opt, string, 节目UUID, range:[1,32], desc:平台下发-->test
</programUUID>
</ProgramNoSecondList>
</EmergencySchedule>
</PlaySchedule>
```

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 响应消息, attr:version{ro, req, string, 版本}-->
  <requestURL>
    <!--ro, req, string, 请求的URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, 状态码, subType:int, [0#OK,1#OK,2#Device Busy,3#Device Error,4#Invalid Operation,5#Invalid XML Format,6#Invalid XML Content,7#Reboot Required]-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, 状态信息, subType:string, [OK#成功,Device Busy#设备忙,Device Error#设备异常,Invalid Operation#无效的操作,Invalid XML Format#无效的XML格式,Invalid XML Content#无效的XML内容,Reboot#设备重启]-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, 详细错误码英文描述, desc: 详细错误码的英文描述-->OK
  </subStatusCode>
  <ID>
    <!--ro, opt, string, 索引ID-->test
  </ID>
</ResponseStatus>
```

69.64.2 获取单个日程参数

Request URL

GET /ISAPI/Publish/ScheduleMgr/playSchedule/<scheduleID>

查询参数

参数名称	参数类型	描述
scheduleID	string	日程ID

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<PlaySchedule xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, 播放日程, attr:version{req, string, 协议版本}-->
  <id>
    <!--ro, req, int, 日程索引-->1
  </id>
  <scheduleName>
    <!--ro, req, string, 日程名称-->test
  </scheduleName>
  <scheduleRemarks>
    <!--ro, opt, string, 日程描述-->test
  </scheduleRemarks>
  <approveState>
    <!--ro, opt, enum, 审核状态标志, subType:string, [approved#审核通过,notPass#未通过,notApprove#未审核]-->approved
  </approveState>
  <approveRemarks>
    <!--ro, opt, string, 审核意见-->test
  </approveRemarks>
  <scheduleMode>
    <!--ro, opt, enum, 日程模式, subType:string, [normal#标准,decode#解码,touch#触摸,decodeTouch#解码触摸]-->normal
  </scheduleMode>
  <orgNo>
    <!--ro, opt, int, 所属组织编号-->1
  </orgNo>
  <scheduleType>
    <!--ro, opt, enum, 日程类型, subType:string, [selfDefine#自定义日程,daily#日日程,weekly#周日程,loop#循环日程,defaultSchedule#垫片日程,emergency#紧急日程],
    desc: 当日程为紧急日程时, 下发后并不立即播放, 在出现紧急情况时 (如火灾、地震等),
    平台选择该日程下对应的紧急节目进行播放, 对应接口: /ISAPI/Publish/ProgramMgr/programPlay?format=json. 播放优先级高于其他日程和插播. 紧急情况消除后, 平台
    停止紧急日程的播放. 设备恢复其他日程或插播的播放. ]-->selfDefine
  </scheduleType>
  <shareProperty>
    <!--ro, opt, enum, 共享属性, subType:string, [public#公有,private#私有]-->public
  </shareProperty>
  <DailySchedule>
    <!--ro, opt, object, 日日程-->
    <PlaySpanList>
      <!--ro, req, array, 播放计划列表, subType:object-->
      <PlaySpan>
        <!--ro, opt, object, 播放计划-->
        <id>
          <!--ro, req, int, 时间段序号-->1
        </id>
```

```
<programNo>
  <!--ro, req, int, 某个时间段播放的节目索引号-->1
</programNo>
<programUUID>
  <!--ro, opt, string, 节目UUID, range:[1,32], desc:平台下发-->test
</programUUID>
<TimeRange>
  <!--ro, req, object, 时间段起始点和结束点-->
  <beginTime>
    <!--ro, req, string, 开始时间, desc:设备最早实现错误（格式为20200826T051500+08）,暂不修改-->test
  </beginTime>
  <endTime>
    <!--ro, req, string, 结束时间, desc:设备最早实现错误（格式为20200826T051500+08）,暂不修改-->test
  </endTime>
</TimeRange>
</PlaySpan>
</PlaySpanList>
<PlaySpanSecondList>
  <!--ro, opt, array, 双屏节目中辅屏节目的播放计划列表, subType:object, desc:新增该字段是为了兼容双屏节目下发到单屏终端的情况，保证单屏终端可以播放主屏节目；同时也是考虑到主辅屏的播放计划不同的情况，下面带second的字段同理-->
  <PlaySpanSecond>
    <!--ro, opt, object, 辅屏播放计划-->
    <id>
      <!--ro, req, int, 时间段序号-->1
    </id>
    <programNo>
      <!--ro, req, int, 某个时间段播放的节目索引号-->1
    </programNo>
    <programUUID>
      <!--ro, opt, string, 节目UUID, range:[1,32], desc:平台下发-->test
    </programUUID>
    <TimeRange>
      <!--ro, req, object, 时间段起始点和结束点-->
      <beginTime>
        <!--ro, req, string, 开始时间, desc:设备最早实现错误（格式为20200826T051500+08）,暂不修改-->test
      </beginTime>
      <endTime>
        <!--ro, req, string, 结束时间, desc:设备最早实现错误（格式为20200826T051500+08）,暂不修改-->test
      </endTime>
    </TimeRange>
  </PlaySpanSecond>
</PlaySpanSecondList>
</DailySchedule>
<WeeklySchedule>
  <!--ro, opt, array, 周播放计划, subType:object-->
  <DayList>
    <!--ro, req, object, 日列表-->
    <Day>
      <!--ro, opt, object, 日信息-->
      <id>
        <!--ro, req, int, 天序号-->1
      </id>
      <dayOfWeek>
        <!--ro, req, enum, 星期, subType:string, [monday#星期一,tuesday#星期二,wednesday#星期三,thursday#星期四,friday#星期五,saturday#星期六,sunday#星期日]-->monday
      </dayOfWeek>
    </Day>
  </DayList>
  <PlaySpanList>
    <!--ro, req, array, 播放计划列表, subType:object-->
    <PlaySpan>
      <!--ro, opt, object, 播放计划-->
      <id>
        <!--ro, req, int, 时间段序号-->1
      </id>
      <programNo>
        <!--ro, req, int, 某个时间段播放的节目索引号-->1
      </programNo>
      <programUUID>
        <!--ro, opt, string, 节目UUID, range:[1,32], desc:平台下发-->test
      </programUUID>
      <TimeRange>
        <!--ro, req, object, 时间段起始点和结束点-->
        <beginTime>
          <!--ro, req, string, 开始时间, desc:设备最早实现错误（格式为20200826T051500+08）,暂不修改-->test
        </beginTime>
        <endTime>
          <!--ro, req, string, 结束时间, desc:设备最早实现错误（格式为20200826T051500+08）,暂不修改-->test
        </endTime>
      </TimeRange>
    </PlaySpan>
  </PlaySpanList>
  <PlaySpanSecondList>
    <!--ro, opt, array, 双屏节目中辅屏节目的播放计划列表, subType:object, desc:新增该字段是为了兼容双屏节目下发到单屏终端的情况，保证单屏终端可以播放主屏节目；同时也是考虑到主辅屏的播放计划不同的情况，下面带second的字段同理-->
    <PlaySpanSecond>
      <!--ro, opt, object, 辅屏播放计划-->
      <id>
        <!--ro, req, int, 时间段序号-->1
      </id>
      <programNo>
        <!--ro, req, int, 某个时间段播放的节目索引号-->1
      </programNo>
      <programUUID>
        <!--ro, opt, string, 节目UUID, range:[1,32], desc:平台下发-->test
      </programUUID>
```

```
</!-ro, req, array, 时间段轮播列表, subType:object, desc:轮播中配置了普通模式, 时间段模式最多只能配置7个时间段 (适合单屏节目或双屏节目中主屏节目的轮播计划) -->
<LoopTimeSpan>
  <!--ro, opt, object, 时间段轮播-->
  <TimeRange>
    <!--ro, req, object, 时间段起始点和结束点-->
    <beginTime>
      <!--ro, req, string, 开始时间, desc:设备最早实现错误 (格式为20200826T051500+08), 暂不修改-->test
    </beginTime>
    <endTime>
      <!--ro, req, string, 结束时间, desc:设备最早实现错误 (格式为20200826T051500+08), 暂不修改-->test
    </endTime>
  </TimeRange>
  <ProgramNoList>
    <!--ro, req, array, 节目编号列表, subType:object-->
    <programNo>
      <!--ro, opt, int, 节目编号-->1
    </programNo>
    <programUUID>
      <!--ro, opt, string, 节目UUID, range:[1,32], desc:平台下发-->test
    </programUUID>
  </ProgramNoList>
  <ProgramNoSecondList>
    <!--ro, req, array, 节目编号列表, subType:object-->
    <programNoSecond>
      <!--ro, opt, int, 节目编号-->1
    </programNoSecond>
  </ProgramNoSecondList>
  <LoopTimeSpanSecondList>
    <!--ro, opt, array, 时间段辅屏轮播列表, subType:object-->
    <LoopTimeSpanSecond>
      <!--ro, opt, object, 时间段辅屏轮播-->
      <TimeRange>
        <!--ro, req, object, 时间段起始点和结束点-->
        <beginTime>
          <!--ro, req, string, 开始时间, desc:设备最早实现错误 (格式为20200826T051500+08), 暂不修改-->test
        </beginTime>
        <endTime>
          <!--ro, req, string, 结束时间, desc:设备最早实现错误 (格式为20200826T051500+08), 暂不修改-->test
        </endTime>
      </TimeRange>
      <ProgramNoSecondList>
        <!--ro, req, array, 节目编号列表, subType:object-->
        <programNoSecond>
          <!--ro, opt, int, 节目编号-->1
        </programNoSecond>
      </ProgramNoSecondList>
    </LoopTimeSpanSecond>
  </LoopTimeSpanSecondList>
</LoopTimeSpan>
</LoopTimeSpanList>
<SelfDefineSchedule>
  <!--ro, opt, object, 自定义日程-->
  <SelfDefineList>
    <!--ro, req, array, 自定义计划列表, subType:object-->
    <SelfDefine>
      <!--ro, opt, object, 自定义计划-->
      <id>
        <!--ro, req, int, 自定义时间段序号-->1
      </id>
      <programNo>
        <!--ro, req, int, 节目编号-->1
      </programNo>
      <programUUID>
        <!--ro, opt, string, 节目UUID, range:[1,32], desc:平台下发-->test
      </programUUID>
```

```
<TimeRange>
  <!--ro, req, object, 时间段起始点和结束点-->
  <beginTime>
    <!--ro, req, string, 开始时间, desc:设备最早实现错误（格式为20200826T051500+08）, 暂不修改-->test
  </beginTime>
  <endTime>
    <!--ro, req, string, 结束时间, desc:设备最早实现错误（格式为20200826T051500+08）, 暂不修改-->test
  </endTime>
</TimeRange>
</SelfDefine>
</SelfDefineList>
<SelfDefineSecondList>
  <!--ro, opt, array, 辅屏自定义计划列表, subType:object-->
  <SelfDefineSecond>
    <!--ro, opt, object, 辅屏自定义计划-->
    <id>
      <!--ro, req, int, 自定义时间段序号-->1
    </id>
    <programNoSecond>
      <!--ro, req, int, 节目编号-->1
    </programNoSecond>
    <TimeRange>
      <!--ro, req, object, 时间段起始点和结束点-->
      <beginTime>
        <!--ro, req, string, 开始时间, desc:设备最早实现错误（格式为20200826T051500+08）, 暂不修改-->test
      </beginTime>
      <endTime>
        <!--ro, req, string, 结束时间, desc:设备最早实现错误（格式为20200826T051500+08）, 暂不修改-->test
      </endTime>
    </TimeRange>
  </SelfDefineSecond>
</SelfDefineSecondList>
</SelfDefineSchedule>
<DefaultSchedule>
  <!--ro, opt, object, 垫片日程-->
  <programNo>
    <!--ro, req, int, 垫片节目-->1
  </programNo>
  <programUUID>
    <!--ro, opt, string, 节目UUID, range:[1,32], desc:平台下发-->test
  </programUUID>
  <programNoSecond>
    <!--ro, opt, int, 辅屏垫片节目-->1
  </programNoSecond>
</DefaultSchedule>
<HolidaySchedule>
  <!--ro, opt, object, 假日计划信息, desc:假日计划为可选节点, 可以附加在日计划和周计划日程里（假日计划新的终端均已不在支持, 因此该部分协议不进行双屏的扩展）-->
  <PlaySpanList>
    <!--ro, req, array, 播放计划列表, subType:object-->
    <PlaySpan>
      <!--ro, opt, object, 播放计划-->
      <id>
        <!--ro, req, int, 时间段编号-->1
      </id>
      <programNo>
        <!--ro, req, int, 节目索引号-->1
      </programNo>
      <programUUID>
        <!--ro, opt, string, 节目UUID, range:[1,32], desc:平台下发-->test
      </programUUID>
      <TimeRange>
        <!--ro, req, object, 时间段起始点和结束点-->
        <beginTime>
          <!--ro, req, string, 开始时间, desc:设备最早实现错误（格式为20200826T051500+08）, 暂不修改-->test
        </beginTime>
        <endTime>
          <!--ro, req, string, 结束时间, desc:设备最早实现错误（格式为20200826T051500+08）, 暂不修改-->test
        </endTime>
      </TimeRange>
    </PlaySpan>
  </PlaySpanList>
</HolidaySchedule>
<scheduleUUID>
  <!--ro, opt, string, 日程UUID, range:[1,32], desc:平台返回-->test
</scheduleUUID>
<EmergencySchedule>
  <!--ro, opt, object, 紧急日程-->
  <ProgramNoList>
    <!--ro, req, array, 紧急日程播放的节目列表, subType:object, desc:普通模式（适合单屏节目或双屏节目中主屏节目）-->
    <programNo>
      <!--ro, opt, int, 节目编号-->1
    </programNo>
    <programUUID>
      <!--ro, opt, string, 节目UUID, range:[1,32], desc:平台下发-->test
    </programUUID>
  </ProgramNoList>
  <ProgramNoSecondList>
    <!--ro, req, array, 紧急日程播放的辅屏节目列表, subType:object, desc:普通模式（双屏节目中辅屏节目的列表）-->
    <programNoSecond>
      <!--ro, opt, int, 节目编号-->1
    </programNoSecond>
    <programUUID>
```

```
<!--ro, opt, string, 节目ID, range:[1,32], desc:节目ID-->test
</programUUID>
</ProgramNoSecondList>
</EmergencySchedule>
</PlaySchedule>
```

69.64.3 获取全部日程参数

Request URL

GET /ISAPI/Publish/ScheduleMgr/playSchedule

查询参数

无

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<PlayScheduleList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, array, 播放日程列表, subType:object, attr:version{req, string, 协议版本}-->
  <PlaySchedule>
    <!--ro, opt, object, 播放日程-->
    <id>
      <!--ro, req, int, 日程索引-->1
    </id>
    <scheduleName>
      <!--ro, req, string, 日程名称-->test
    </scheduleName>
    <scheduleRemarks>
      <!--ro, opt, string, 日程描述-->test
    </scheduleRemarks>
    <approveState>
      <!--ro, opt, enum, 审核状态标志, subType:string, [approved#审核通过,notPass#未通过,notApprove#未审核]-->approved
    </approveState>
    <approveRemarks>
      <!--ro, opt, string, 审核意见-->test
    </approveRemarks>
    <scheduleMode>
      <!--ro, opt, enum, 日程模式, subType:string, [normal#标准,decode#解码,touch#触摸,decodeTouch#解码触摸]-->normal
    </scheduleMode>
    <orgNo>
      <!--ro, opt, int, 所属组织编号-->1
    </orgNo>
    <scheduleType>
      <!--ro, opt, enum, 日程类型, subType:string, [selfDefine#自定义日程,daily#日日程,weekly#周日程,loop#循环日程,defaultSchedule#整片日程,emergency#紧急日程],
      desc: 当日程为紧急日程时, 下发后并不立即播放, 在出现紧急情况时(如火灾、地震等),
      平台选择该日程下对应的紧急节目进行播放, 对应接口: /ISAPI/Publish/ProgramMgr/programPlay?format=json. 播放优先级高于其他日程和插播。紧急情况消除后, 平台
      停止紧急日程的播放。设备恢复其他日程或插播的播放。]-->selfDefine
    </scheduleType>
    <shareProperty>
      <!--ro, opt, enum, 共享属性, subType:string, [public#公有,private#私有]-->public
    </shareProperty>
    <DailySchedule>
      <!--ro, opt, object, 日日程-->
      <PlaySpanList>
        <!--ro, req, array, 播放计划列表, subType:object-->
        <PlaySpan>
          <!--ro, opt, object, 播放计划-->
          <id>
            <!--ro, req, int, 时间段序号-->1
          </id>
          <programNo>
            <!--ro, req, int, 某个时间段播放的节目索引号-->1
          </programNo>
          <TimeRange>
            <!--ro, req, object, 时间段起始点和结束点-->
            <beginTime>
              <!--ro, req, string, 开始时间, desc:设备最早实现错误(格式为20200826T051500+08), 暂不修改-->test
            </beginTime>
            <endTime>
              <!--ro, req, string, 结束时间, desc:设备最早实现错误(格式为20200826T051500+08), 暂不修改-->test
            </endTime>
          </TimeRange>
        </PlaySpan>
      </PlaySpanList>
      <PlaySpanSecondList>
        <!--ro, opt, array, 双屏节目中辅屏节目的播放计划列表, subType:object, desc:新增该字段是为了兼容双屏节目下发到单屏终端的情况, 保证单屏终端可以播放主
        屏节目; 同时也是考虑到主辅屏的播放计划不同的情况, 下面带second的字段同理-->
        <PlaySpanSecond>
          <!--ro, opt, object, 辅屏播放计划-->
          <id>
            <!--ro, req, int, 时间段序号-->1
          </id>
```

```

</id>
<programNo>
  <!--ro, req, int, 某个时间段播放的节目索引号-->1
</programNo>
<TimeRange>
  <!--ro, req, object, 时间段起始点和结束点-->
  <beginTime>
    <!--ro, req, string, 开始时间, desc:设备最早实现错误（格式为20200826T051500+08）, 暂不修改-->test
  </beginTime>
  <endTime>
    <!--ro, req, string, 结束时间, desc:设备最早实现错误（格式为20200826T051500+08）, 暂不修改-->test
  </endTime>
</TimeRange>
</PlaySpanSecond>
</PlaySpanSecondList>
</DailySchedule>
<WeeklySchedule>
  <!--ro, opt, array, 周播放计划, subType:object-->
  <DayList>
    <!--ro, req, object, 日列表-->
    <Day>
      <!--ro, opt, object, 日信息-->
      <id>
        <!--ro, req, int, 天序号-->1
      </id>
      <dayOfWeek>
        <!--ro, req, enum, 星期, subType:string, [monday#星期一,tuesday#星期二,wednesday#星期三,thursday#星期四,friday#星期五,saturday#星期六,sunday#星期日]-->monday
      </dayOfWeek>
      <PlaySpanList>
        <!--ro, req, array, 播放计划列表, subType:object-->
        <PlaySpan>
          <!--ro, opt, object, 播放计划-->
          <id>
            <!--ro, req, int, 时间段序号-->1
          </id>
          <programNo>
            <!--ro, req, int, 某个时间段播放的节目索引号-->1
          </programNo>
          <TimeRange>
            <!--ro, req, object, 时间段起始点和结束点-->
            <beginTime>
              <!--ro, req, string, 开始时间, desc:设备最早实现错误（格式为20200826T051500+08）, 暂不修改-->test
            </beginTime>
            <endTime>
              <!--ro, req, string, 结束时间, desc:设备最早实现错误（格式为20200826T051500+08）, 暂不修改-->test
            </endTime>
          </TimeRange>
          </PlaySpan>
        </PlaySpanList>
        <PlaySpanSecondList>
          <!--ro, opt, array, 双屏节目中辅屏节目的播放计划列表, subType:object, desc:新增该字段是为了兼容双屏节目下发到单屏终端的情况, 保证单屏终端可以播放主屏节目; 同时也是考虑到主辅屏的播放计划不同的情况, 下面带second的字段同理-->
          <PlaySpanSecond>
            <!--ro, opt, object, 辅屏播放计划-->
            <id>
              <!--ro, req, int, 时间段序号-->1
            </id>
            <programNo>
              <!--ro, req, int, 某个时间段播放的节目索引号-->1
            </programNo>
            <TimeRange>
              <!--ro, req, object, 时间段起始点和结束点-->
              <beginTime>
                <!--ro, req, string, 开始时间, desc:设备最早实现错误（格式为20200826T051500+08）, 暂不修改-->test
              </beginTime>
              <endTime>
                <!--ro, req, string, 结束时间, desc:设备最早实现错误（格式为20200826T051500+08）, 暂不修改-->test
              </endTime>
            </TimeRange>
            </PlaySpanSecond>
          </PlaySpanSecondList>
        </Day>
      </DayList>
    </WeeklySchedule>
  </LoopSchedule>
  <!--ro, opt, object, 循环计划-->
  <ProgramNoList>
    <!--ro, req, array, 循环播放的节目列表, subType:object, desc:普通模式（适合单屏节目或双屏节目中主屏节目）-->
    <programNo>
      <!--ro, opt, int, 节目编号-->1
    </programNo>
  </ProgramNoList>
  <ProgramNoSecondList>
    <!--ro, req, array, 循环播放的辅屏节目列表, subType:object, desc:普通模式（双屏节目中辅屏节目的列表）-->
    <programNoSecond>
      <!--ro, opt, int, 节目编号-->1
    </programNoSecond>
  </ProgramNoSecondList>
  <LoopTimeSpanList>
    <!--ro, opt, array, 时间段轮播列表, subType:object, desc:轮播中配置了普通模式, 时间段模式最多只能配置7个时间段（适合单屏节目或双屏节目中主屏节目的轮播计划）-->
    <LoopTimeSpan>

```

```
<!--ro, opt, object, 时间段轮播-->
<TimeRange>
  <!--ro, req, object, 时间段起始点和结束点-->
  <beginTime>
    <!--ro, req, string, 开始时间, desc:设备最早实现错误（格式为20200826T051500+08）, 暂不修改-->test
  </beginTime>
  <endTime>
    <!--ro, req, string, 结束时间, desc:设备最早实现错误（格式为20200826T051500+08）, 暂不修改-->test
  </endTime>
</TimeRange>
<ProgramNoList>
  <!--ro, req, array, 节目编号列表, subType:object-->
  <programNo>
    <!--ro, opt, int, 节目编号-->1
  </programNo>
</ProgramNoList>
</LoopTimeSpan>
</LoopTimeSpanList>
<LoopTimeSpanSecondList>
  <!--ro, opt, array, 时间段辅屏轮播列表, subType:object-->
  <LoopTimeSpanSecond>
    <!--ro, opt, object, 时间段辅屏轮播-->
    <TimeRange>
      <!--ro, req, object, 时间段起始点和结束点-->
      <beginTime>
        <!--ro, req, string, 开始时间, desc:设备最早实现错误（格式为20200826T051500+08）, 暂不修改-->test
      </beginTime>
      <endTime>
        <!--ro, req, string, 结束时间, desc:设备最早实现错误（格式为20200826T051500+08）, 暂不修改-->test
      </endTime>
    </TimeRange>
    <ProgramNoSecondList>
      <!--ro, req, array, 节目编号列表, subType:object-->
      <programNoSecond>
        <!--ro, opt, int, 节目编号-->1
      </programNoSecond>
    </ProgramNoSecondList>
  </LoopTimeSpanSecond>
</LoopTimeSpanSecondList>
</LoopSchedule>
<SelfDefineSchedule>
  <!--ro, opt, object, 自定义日程-->
  <SelfDefineList>
    <!--ro, req, array, 自定义计划列表, subType:object-->
    <SelfDefine>
      <!--ro, opt, object, 自定义计划-->
      <id>
        <!--ro, req, int, 自定义时间段序号-->1
      </id>
      <programNo>
        <!--ro, req, int, 节目编号-->1
      </programNo>
      <TimeRange>
        <!--ro, req, object, 时间段起始点和结束点-->
        <beginTime>
          <!--ro, req, string, 开始时间, desc:设备最早实现错误（格式为20200826T051500+08）, 暂不修改-->test
        </beginTime>
        <endTime>
          <!--ro, req, string, 结束时间, desc:设备最早实现错误（格式为20200826T051500+08）, 暂不修改-->test
        </endTime>
      </TimeRange>
    </SelfDefine>
  </SelfDefineList>
  <SelfDefineSecondList>
    <!--ro, opt, array, 辅屏自定义计划列表, subType:object-->
    <SelfDefineSecond>
      <!--ro, opt, object, 辅屏自定义计划-->
      <id>
        <!--ro, req, int, 自定义时间段序号-->1
      </id>
      <programNoSecond>
        <!--ro, req, int, 节目编号-->1
      </programNoSecond>
      <TimeRange>
        <!--ro, req, object, 时间段起始点和结束点-->
        <beginTime>
          <!--ro, req, string, 开始时间, desc:设备最早实现错误（格式为20200826T051500+08）, 暂不修改-->test
        </beginTime>
        <endTime>
          <!--ro, req, string, 结束时间, desc:设备最早实现错误（格式为20200826T051500+08）, 暂不修改-->test
        </endTime>
      </TimeRange>
    </SelfDefineSecond>
  </SelfDefineSecondList>
</SelfDefineSchedule>
<DefaultSchedule>
  <!--ro, opt, object, 垫片日程-->
  <programNo>
    <!--ro, req, int, 垫片节目-->1
  </programNo>
  <programNoSecond>
    <!--ro, opt, int, 辅屏垫片节目-->1
  </programNoSecond>
</DefaultSchedule>
```

```
<HolidaySchedule>
  <!--ro, opt, object, 假日计划信息, desc:假日计划为可选节点, 可以附加在日计划和周计划日程里 (假日计划新的终端均已不在支持, 因此该部分协议不进行双屏的扩展)-->
</HolidaySchedule>
<PlaySpanList>
  <!--ro, req, array, 播放计划列表, subType:object-->
  <PlaySpan>
    <!--ro, opt, object, 播放计划-->
    <id>
      <!--ro, req, int, 时间段编号-->1
    </id>
    <programNo>
      <!--ro, req, int, 节目索引号-->1
    </programNo>
    <TimeRange>
      <!--ro, req, object, 时间段起始点和结束点-->
      <beginTime>
        <!--ro, req, string, 开始时间, desc:设备最早实现错误 (格式为20200826T051500+08), 暂不修改-->test
      </beginTime>
      <endTime>
        <!--ro, req, string, 结束时间, desc:设备最早实现错误 (格式为20200826T051500+08), 暂不修改-->test
      </endTime>
    </TimeRange>
  </PlaySpan>
</PlaySpanList>
</HolidaySchedule>
<scheduleUUID>
  <!--ro, opt, string, 日程UUID, range:[1,32], desc:平台返回-->test
</scheduleUUID>
<EmergencySchedule>
  <!--ro, opt, object, 紧急日程-->
  <ProgramNoList>
    <!--ro, req, array, 紧急日程播放的节目列表, subType:object, desc:普通模式 (适合单屏节目或双屏节目中主屏节目) -->
    <programNo>
      <!--ro, opt, int, 节目编号-->1
    </programNo>
    <programUUID>
      <!--ro, opt, string, 节目UUID, range:[1,32], desc:平台下发-->test
    </programUUID>
  </ProgramNoList>
  <ProgramNoSecondList>
    <!--ro, req, array, 紧急日程播放的辅屏节目列表, subType:object, desc:普通模式 (双屏节目中辅屏节目的列表) -->
    <programNoSecond>
      <!--ro, opt, int, 节目编号-->1
    </programNoSecond>
    <programUUID>
      <!--ro, opt, string, 节目UUID, range:[1,32], desc:平台下发-->test
    </programUUID>
  </ProgramNoSecondList>
</EmergencySchedule>
</PlaySchedule>
</PlayScheduleList>
```

69.64.4 配置单个日程参数

Request URL

PUT /ISAPI/Publish/ScheduleMgr/playSchedule/<scheduleID>

查询参数

参数名称	参数类型	描述
scheduleID	string	日程ID

请求报文

```
<?xml version="1.0" encoding="UTF-8"?>

<PlaySchedule xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--opt, object, 播放日程, attr:version{req, string, 协议版本}-->
  <id>
    <!--req, int, 日程索引-->1
  </id>
  <scheduleName>
    <!--req, string, 日程名称-->test
  </scheduleName>
  <scheduleRemarks>
    <!--opt, string, 日程描述-->test
  </scheduleRemarks>
  <approveState>
    <!--opt, enum, 审核状态标志, subType:string, [approved#审核通过,notPass#未通过,notApprove#未审核]-->approved
  </approveState>
  <approveRemarks>
    <!--opt, string, 审核意见-->test
  </approveRemarks>
  <scheduleMode>
    <!--opt, enum, 日程模式, subType:string, [normal#标准.decode#解码.touch#触控.decodeTouch#解码触控]-->normal
  </scheduleMode>
```

```
</scheduleMode>
<orgNo>
  <!--opt, int, 所属组织编号-->1
</orgNo>
<scheduleType>
  <!--opt, enum, 日程类型, subType:string, [selfDefine#自定义日程,daily#日日程,weekly#周日程,loop#循环日程,defaultSchedule#垫片日程,emergency#紧急日程],
desc:当日程为紧急日程时, 下发后并不立即播放, 在出现紧急情况时(如火灾、地震等),
  平台选择该日程下对应的紧急节目进行播放, 对应接口: /ISAPI/Publish/ProgramMgr/programPlay?format=json. 播放优先级高于其他日程和插播. 紧急情况消除后, 平台
  停止紧急日程的播放. 设备恢复其他日程或插播的播放. ]-->selfDefine
</scheduleType>
<shareProperty>
  <!--opt, enum, 共享属性, subType:string, [public#公有,private#私有]-->public
</shareProperty>
<DailySchedule>
  <!--opt, object, 日日程-->
  <PlaySpanList>
    <!--req, array, 播放计划列表, subType:object-->
    <PlaySpan>
      <!--opt, object, 播放计划-->
      <id>
        <!--req, int, 时间段序号-->1
      </id>
      <programNo>
        <!--req, int, 某个时间段播放的节目索引号-->1
      </programNo>
      <TimeRange>
        <!--req, object, 时间段起始点和结束点-->
        <beginTime>
          <!--req, string, 开始时间, desc:设备最早实现错误(格式为20200826T051500+08), 暂不修改-->test
        </beginTime>
        <endTime>
          <!--req, string, 结束时间, desc:设备最早实现错误(格式为20200826T051500+08), 暂不修改-->test
        </endTime>
      </TimeRange>
    </PlaySpan>
  </PlaySpanList>
  <PlaySpanSecondList>
    <!--opt, array, 双屏节目中辅屏节目的播放计划列表, subType:object, desc:新增该字段是为了兼容双屏节目下落到单屏终端的情况, 保证单屏终端可以播放主屏节
    目; 同时也是考虑到主辅屏的播放计划不同的情况, 下面带second的字段同理-->
    <PlaySpanSecond>
      <!--opt, object, 辅屏播放计划-->
      <id>
        <!--req, int, 时间段序号-->1
      </id>
      <programNo>
        <!--req, int, 某个时间段播放的节目索引号-->1
      </programNo>
      <TimeRange>
        <!--req, object, 时间段起始点和结束点-->
        <beginTime>
          <!--req, string, 开始时间, desc:设备最早实现错误(格式为20200826T051500+08), 暂不修改-->test
        </beginTime>
        <endTime>
          <!--req, string, 结束时间, desc:设备最早实现错误(格式为20200826T051500+08), 暂不修改-->test
        </endTime>
      </TimeRange>
    </PlaySpanSecond>
  </PlaySpanSecondList>
</DailySchedule>
<WeeklySchedule>
  <!--opt, array, 周播放计划, subType:object-->
  <DayList>
    <!--req, object, 日列表-->
    <Day>
      <!--opt, object, 日信息-->
      <id>
        <!--req, int, 天序号-->1
      </id>
      <dayOfWeek>
        <!--req, enum, 星期, subType:string, [monday#星期一,tuesday#星期二,wednesday#星期三,thursday#星期四,friday#星期五,saturday#星期六,sunday#星期日]--
>monday
      </dayOfWeek>
    </Day>
  </DayList>
  <PlaySpanList>
    <!--req, array, 播放计划列表, subType:object-->
    <PlaySpan>
      <!--opt, object, 播放计划-->
      <id>
        <!--req, int, 时间段序号-->1
      </id>
      <programNo>
        <!--req, int, 某个时间段播放的节目索引号-->1
      </programNo>
      <TimeRange>
        <!--req, object, 时间段起始点和结束点-->
        <beginTime>
          <!--req, string, 开始时间, desc:设备最早实现错误(格式为20200826T051500+08), 暂不修改-->test
        </beginTime>
        <endTime>
          <!--req, string, 结束时间, desc:设备最早实现错误(格式为20200826T051500+08), 暂不修改-->test
        </endTime>
      </TimeRange>
    </PlaySpan>
  </PlaySpanList>
```

```
<PlaySpanSecondList>
  <!--opt, array, 双屏节目中辅屏节目的播放计划列表, subType:object, desc:新增该字段是为了兼容双屏节目下发到单屏终端的情况, 保证单屏终端可以播放主屏
  节目; 同时也是考虑到主辅屏的播放计划不同的情况, 下面带second的字段同理-->
  <PlaySpanSecond>
    <!--opt, object, 辅屏播放计划-->
    <id>
      <!--req, int, 时间段序号-->1
    </id>
    <programNo>
      <!--req, int, 某个时间段播放的节目索引号-->1
    </programNo>
    <TimeRange>
      <!--req, object, 时间段起始点和结束点-->
      <beginTime>
        <!--req, string, 开始时间, desc:设备最早实现错误(格式为20200826T051500+08), 暂不修改-->test
      </beginTime>
      <endTime>
        <!--req, string, 结束时间, desc:设备最早实现错误(格式为20200826T051500+08), 暂不修改-->test
      </endTime>
    </TimeRange>
  </PlaySpanSecond>
</PlaySpanSecondList>
</Day>
</DayList>
</WeeklySchedule>
<LoopSchedule>
  <!--opt, object, 循环计划-->
  <ProgramNoList>
    <!--req, array, 循环播放的节目列表, subType:object, desc:普通模式(适合单屏节目或双屏节目中主屏节目)-->
    <programNo>
      <!--opt, int, 节目编号-->1
    </programNo>
  </ProgramNoList>
  <ProgramNoSecondList>
    <!--req, array, 循环播放的辅屏节目列表, subType:object, desc:普通模式(双屏节目中辅屏节目的列表)-->
    <programNoSecond>
      <!--opt, int, 节目编号-->1
    </programNoSecond>
  </ProgramNoSecondList>
  <LoopTimeSpanList>
    <!--opt, array, 时间段轮播列表, subType:object, desc:轮播中配置了普通模式, 时间段模式最多只能配置7个时间段(适合单屏节目或双屏节目中主屏节目的轮播计
    划)-->
    <LoopTimeSpan>
      <!--opt, object, 时间段轮播-->
      <TimeRange>
        <!--req, object, 时间段起始点和结束点-->
        <beginTime>
          <!--req, string, 开始时间, desc:设备最早实现错误(格式为20200826T051500+08), 暂不修改-->test
        </beginTime>
        <endTime>
          <!--req, string, 结束时间, desc:设备最早实现错误(格式为20200826T051500+08), 暂不修改-->test
        </endTime>
      </TimeRange>
      <ProgramNoList>
        <!--req, array, 节目编号列表, subType:object-->
        <programNo>
          <!--opt, int, 节目编号-->1
        </programNo>
      </ProgramNoList>
    </LoopTimeSpan>
  </LoopTimeSpanList>
  <LoopTimeSpanSecondList>
    <!--opt, array, 时间段辅屏轮播列表, subType:object-->
    <LoopTimeSpanSecond>
      <!--opt, object, 时间段辅屏轮播-->
      <TimeRange>
        <!--req, object, 时间段起始点和结束点-->
        <beginTime>
          <!--req, string, 开始时间, desc:设备最早实现错误(格式为20200826T051500+08), 暂不修改-->test
        </beginTime>
        <endTime>
          <!--req, string, 结束时间, desc:设备最早实现错误(格式为20200826T051500+08), 暂不修改-->test
        </endTime>
      </TimeRange>
      <ProgramNoSecondList>
        <!--req, array, 节目编号列表, subType:object-->
        <programNoSecond>
          <!--opt, int, 节目编号-->1
        </programNoSecond>
      </ProgramNoSecondList>
    </LoopTimeSpanSecond>
  </LoopTimeSpanSecondList>
</LoopSchedule>
<SelfDefineSchedule>
  <!--opt, object, 自定义日程-->
  <SelfDefineList>
    <!--req, array, 自定义计划列表, subType:object-->
    <SelfDefine>
      <!--opt, object, 自定义计划-->
      <id>
        <!--req, int, 自定义时间段序号-->1
      </id>
      <programNo>
        <!--req, int, 节目编号-->1
      </programNo>
    </SelfDefine>
  </SelfDefineList>
</SelfDefineSchedule>
```

```
<!--req, int, 节目编号-->1
</programNo>
<TimeRange>
  <!--req, object, 时间段起始点和结束点-->
  <beginTime>
    <!--req, string, 开始时间, desc:设备最早实现错误（格式为20200826T051500+08）, 暂不修改-->test
  </beginTime>
  <endTime>
    <!--req, string, 结束时间, desc:设备最早实现错误（格式为20200826T051500+08）, 暂不修改-->test
  </endTime>
</TimeRange>
</SelfDefine>
</SelfDefineList>
<SelfDefineSecondList>
  <!--opt, array, 辅屏自定义计划列表, subType:object-->
  <SelfDefineSecond>
    <!--opt, object, 辅屏自定义计划-->
    <id>
      <!--req, int, 自定义时间段序号-->1
    </id>
    <programNoSecond>
      <!--req, int, 节目编号-->1
    </programNoSecond>
    <TimeRange>
      <!--req, object, 时间段起始点和结束点-->
      <beginTime>
        <!--req, string, 开始时间, desc:设备最早实现错误（格式为20200826T051500+08）, 暂不修改-->test
      </beginTime>
      <endTime>
        <!--req, string, 结束时间, desc:设备最早实现错误（格式为20200826T051500+08）, 暂不修改-->test
      </endTime>
    </TimeRange>
    </SelfDefineSecond>
  </SelfDefineSecondList>
</SelfDefineSchedule>
<DefaultSchedule>
  <!--opt, object, 垫片日程-->
  <programNo>
    <!--req, int, 垫片节目-->1
  </programNo>
  <programNoSecond>
    <!--opt, int, 辅屏垫片节目-->1
  </programNoSecond>
</DefaultSchedule>
<HolidaySchedule>
  <!--opt, object, 假日计划信息, desc:假日计划为可选节点，可以附加在日计划和周计划日程里（假日计划新的终端均已不在支持，因此该部分协议不进行双屏的扩展）-->
</HolidaySchedule>
<PlaySpanList>
  <!--req, array, 播放计划列表, subType:object-->
  <PlaySpan>
    <!--opt, object, 播放计划-->
    <id>
      <!--req, int, 时间段编号-->1
    </id>
    <programNo>
      <!--req, int, 节目索引号-->1
    </programNo>
    <TimeRange>
      <!--req, object, 时间段起始点和结束点-->
      <beginTime>
        <!--req, string, 开始时间, desc:设备最早实现错误（格式为20200826T051500+08）, 暂不修改-->test
      </beginTime>
      <endTime>
        <!--req, string, 结束时间, desc:设备最早实现错误（格式为20200826T051500+08）, 暂不修改-->test
      </endTime>
    </TimeRange>
  </PlaySpan>
</PlaySpanList>
<HolidaySchedule>
<scheduleUUID>
  <!--opt, string, 日程UUID, range:[1,32], desc:平台返回-->test
</scheduleUUID>
<EmergencySchedule>
  <!--opt, object, 紧急日程-->
  <ProgramNoList>
    <!--req, array, 紧急日程播放的节目列表, subType:object, desc:普通模式（适合单屏节目或双屏节目中主屏节目）-->
    <programNo>
      <!--opt, int, 节目编号-->1
    </programNo>
    <programUUID>
      <!--ro, opt, string, 节目UUID, range:[1,32], desc:平台下发-->test
    </programUUID>
  </ProgramNoList>
  <ProgramNoSecondList>
    <!--req, array, 紧急日程播放的辅屏节目列表, subType:object, desc:普通模式（双屏节目中辅屏节目的列表）-->
    <programNoSecond>
      <!--opt, int, 节目编号-->1
    </programNoSecond>
    <programUUID>
      <!--ro, opt, string, 节目UUID, range:[1,32], desc:平台下发-->test
    </programUUID>
  </ProgramNoSecondList>
</EmergencySchedule>
</PlaySchedule>
```

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 响应消息, attr:version{ro, req, string, 版本}-->
  <requestURL>
    <!--ro, req, string, 请求的URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, 状态码, subType:int, [0#OK,1#OK,2#Device Busy,3#Device Error,4#Invalid Operation,5#Invalid XML Format,6#Invalid XML Content,7#Reboot Required]-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, 状态信息, subType:string, [OK#成功,Device Busy#设备忙,Device Error#设备异常,Invalid Operation#无效的操作,Invalid XML Format#无效的XML格式,Invalid XML Content#无效的XML内容,Reboot#设备重启]-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, 详细错误码英文描述, desc: 详细错误码的英文描述-->OK
  </subStatusCode>
</ResponseStatus>
```

69.64.5 获取日程参数配置能力

Request URL

GET /ISAPI/Publish/ScheduleMgr/playSchedule/<scheduleID>/capabilities

查询参数

参数名称	参数类型	描述
scheduleID	string	日程ID

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>
<PlaySchedule xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, 日程参数, attr:version{req, string, 协议版本}-->
  <id min="1" max="8">
    <!--ro, req, int, 日程索引, attr:min{opt, int, 最小值},max{opt, int, 最大值}-->1
  </id>
  <scheduleName max="10">
    <!--ro, req, string, 日程名称, attr:max{req, int, 最大值}-->test
  </scheduleName>
  <scheduleMode opt="normal,decode,touch,decodeTouch">
    <!--ro, opt, enum, 日程模式, subType:string, [normal#标准,decode#解码,touch#触摸,decodeTouch#解码触摸], attr:opt{req, string, 取值范围}-->normal
  </scheduleMode>
  <scheduleType opt="selfDefine,daily,weekly,loop,defaultSchedule">
    <!--ro, opt, enum, 日程类型, subType:string, [selfDefine#自定义日程,daily#日日程,weekly#周日程,loop#循环日程,defaultSchedule#垫片日程], attr:opt{req, string, 取值范围}-->selfDefine
  </scheduleType>
  <DailySchedule>
    <!--ro, opt, object, 日日程-->
    <PlaySpanList size="10">
      <!--ro, req, array, 播放计划列表, subType:object, attr:size{req, int, 数组元素总数}-->
      <PlaySpan>
        <!--ro, opt, object, 播放计划-->
        <id>
          <!--ro, req, int, 时间段序号-->1
        </id>
        <programNo>
          <!--ro, req, int, 某个时间段播放的节目索引号-->1
        </programNo>
        <TimeRange>
          <!--ro, req, object, 时间段起始点和结束点-->
          <beginTime>
            <!--ro, req, string, 开始时间, desc:设备最早实现错误（格式为20200826T051500+08）, 暂不修改-->test
          </beginTime>
          <endTime>
            <!--ro, req, string, 结束时间, desc:设备最早实现错误（格式为20200826T051500+08）, 暂不修改-->test
          </endTime>
        </TimeRange>
      </PlaySpan>
    </PlaySpanList>
  </DailySchedule>
</PlaySchedule>
```

69.64.6 删除单个日程

Request URL

DELETE /ISAPI/Publish/ScheduleMgr/playSchedule/<scheduleID>

查询参数

参数名称	参数类型	描述
scheduleID	string	日程ID

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 响应消息, attr:version{ro, req, string, 版本}-->
  <requestURL>
    <!--ro, req, string, 请求的URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, 状态码, subType:int, [0#OK,1#OK,2#Device Busy,3#Device Error,4#Invalid Operation,5#Invalid XML Format,6#Invalid XML Content,7#Reboot Required]-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, 状态信息, subType:string, [OK#成功,Device Busy#设备忙,Device Error#设备异常,Invalid Operation#无效的操作,Invalid XML Format#无效的XML格式,Invalid XML Content#无效的XML内容,Reboot#设备重启]-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, 详细错误码英文描述, desc:详细错误码的英文描述-->OK
  </subStatusCode>
</ResponseStatus>
```

69.64.7 获取日程管理能力集

Request URL

GET /ISAPI/Publish/ScheduleMgr/capabilities?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "ScheduleMgrCap": {
    /*ro, opt, object, 未定义*/
    "isSupportPostSchedule": true
    /*ro, opt, bool, 是否支持日程发布（该日程发布仅支持终端基线平台和8520平台）*/
  }
}
```

69.64.8 获取日程查找参数能力

Request URL

GET /ISAPI/Publish/ScheduleMgr/scheduleSearch/profile

查询参数

无

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<ScheduleSearchProfile xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, 未定义, attr:version{req, string, 协议版本}-->
  <maxTimeSpansNum>
    <!--ro, req, int, 支持的最大时间段数-->1
  </maxTimeSpansNum>
  <maxMatchResults>
    <!--ro, req, int, 一次搜索返回的最大匹配结果个数-->1
  </maxMatchResults>
  <maxSearchTimeout>
    <!--ro, opt, int, 搜索超时时间-->1
  </maxSearchTimeout>
  <maxConcurrentSearches>
    <!--ro, opt, int, 支持的最大并发搜索个数-->1
  </maxConcurrentSearches>
  <approveState opt="approved,notPass,notApprove,all">
    <!--ro, opt, enum, 审核状态类型, subType:string, [approved#已通过,notPass#未通过,notApprove#未审核,all#全部], attr:opt{req, string, 取值范围}-->all
  </approveState>
  <shareProperty opt="public,private,all">
    <!--ro, opt, enum, 共享属性类型, subType:string, [public#公有,private#私有,all#所有], attr:opt{req, string, 取值范围}-->all
  </shareProperty>
  <isSupportUploader>
    <!--ro, opt, bool, 是否支持按上传者条件搜索-->true
  </isSupportUploader>
  <scheduleType opt="selfDefine,daily,weekly,loop,defaultSchedule,all">
    <!--ro, opt, enum, 支持搜索的日程类型, subType:string, [selfDefine#自定义,daily#日程,weekly#周日程,loop#循环日程,defaultSchedule#默认日程,all#所有日程], attr:opt{req, string, 取值范围}-->all
  </scheduleType>
  <scheduleMode opt="normal,decode,touch,decodeTouch">
    <!--ro, opt, enum, 日程模式, subType:string, [normal#普通,decode#解码,touch#触摸,decodeTouch#解码触摸], attr:opt{req, string, 取值范围}-->normal
  </scheduleMode>
  <isSupportScheduleName>
    <!--ro, opt, bool, 是否支持日程名称搜索-->true
  </isSupportScheduleName>
  <isSupportScheduleNameLike>
    <!--ro, opt, bool, 是否支持日程名称模糊搜索-->true
  </isSupportScheduleNameLike>
  <isSupportScheduleRemarksLike>
    <!--ro, opt, bool, 是否支持日程描述模糊搜索-->true
  </isSupportScheduleRemarksLike>
  <isSupportModer>
    <!--ro, opt, bool, 是否支持按修改者搜索-->true
  </isSupportModer>
  <isSupportOrgName>
    <!--ro, opt, bool, 是否支持按组织搜索-->true
  </isSupportOrgName>
  <isSupportSubOrg>
    <!--ro, opt, bool, 是否支持子组织-->true
  </isSupportSubOrg>
  <isSupportSort>
    <!--ro, opt, bool, 是否支持排序-->true
  </isSupportSort>
  <isSupportKeyword>
    <!--ro, opt, bool, 是否支持关键字搜索-->true
  </isSupportKeyword>
</ScheduleSearchProfile>
```

69.65 素材管理

69.65.1 创建单个素材

Request URL

POST /ISAPI/Publish/MaterialMgr/material?security=<security>&iv=<iv>

查询参数

参数名称	参数类型	描述
security	string	没有security表示数据不加密；security=1,表示报文中的accessKey和secretKey字段AES128加密；security=2,表示报文中的accessKey和secretKey字段AES256加密；
iv	string	随机字符串，长度限制为32；iv表示初始化向量，在security=1或者2时为必须参数；

请求报文

```
<?xml version="1.0" encoding="UTF-8"?>

<Material xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, 素材信息, attr:version{req, string, 协议版本}-->
```

```
<id>
  <!--req, int, 素材索引-->1
</id>
<materialName>
  <!--req, string, 素材名称-->test
</materialName>
<materialRemarks>
  <!--req, string, 素材描述-->test
</materialRemarks>
<materialType>
  <!--req, enum, 素材类型, subType:string, [static#静态,dynamic#动态]-->static
</materialType>
<approveState>
  <!--opt, enum, 审核状态标志, subType:string, [approved#审核通过,notPass#未通过,notApprove#未审核]-->approved
</approveState>
<approveRemarks>
  <!--opt, string, 审核意见-->test
</approveRemarks>
<shareProperty>
  <!--opt, enum, 共享属性, subType:string, [public#共有,private#私有]-->public
</shareProperty>
<uploadUser>
  <!--req, string, 上传用户-->test
</uploadUser>
<uploadTime>
  <!--req, datetime, 上传时间-->1970-01-01T00:00:00+08:00
</uploadTime>
<materialEncrypt>
  <!--opt, string, 素材校验串, desc:用来校验终端收到的素材正确性,该字段仅在服务器到终端链路生效,具体加密方式参见《信息发布素材加解密方案》,该方案文档可以从素材领域中获取到-->JEmNTA3NDg5NCY0JjI3OTMSMjAmYwEmMzVyOTMSOCZhMCY0MjAzMDQwJmI1JjQzMzc3ODgmdG=
</materialEncrypt>
<orgNo>
  <!--opt, int, 所属组织编号-->1
</orgNo>
<replaceTerminal>
  <!--opt, bool, 素材是否更新到终端, desc:素材替换时该字段有效-->true
</replaceTerminal>
<StaticMaterial>
  <!--opt, object, 静态素材, dep:and,{$.Material.materialType,eq,static}, desc:当materialType 类型为static 时有效-->
    <staticMaterialType>
      <!--opt, enum, 静态素材类型, subType:string, [picture#图片,flash#flash,audio#音频,video#视频,document#文本,ppt#ppt,doc#文档,excel#表格,pdf#pdf,web#网页,app#app], dep:and,{$.Material.StaticMaterial,eq,true}, desc:当StaticMaterial节点存在时有效-->picture
    </staticMaterialType>
    <picFormat>
      <!--opt, enum, 图片格式, subType:string, [gif#gif,bmp#bmp,jpg#jpg,png#png], dep:and,{$.Material.StaticMaterial.staticMaterialType,eq,picture}, desc:当staticMaterialType为picture时有效-->png
    </picFormat>
    <flashFormat>
      <!--opt, enum, flash格式, subType:string, [swf#swf], desc:当staticMaterialType为flash时有效-->swf
    </flashFormat>
    <audioFormat>
      <!--opt, enum, 音频格式, subType:string, [mp3#mp3,wav#wav,wma#wma], desc:当staticMaterialType为audio时有效-->wma
    </audioFormat>
    <videoFormat>
      <!--opt, enum, 视频格式, subType:string, [rm#rm,rmvb#rmvb,asf#asf,avi#avi,mpg#mpg,3gp#3gp,mov#mov,mkv#mkv,wmv#wmv,flv#flv,mp4#mp4,hkps#hkps], desc:当staticMaterialType为video时有效-->mp4
    </videoFormat>
    <documentFormat>
      <!--opt, enum, 文本格式, subType:string, [txt#txt], desc:当staticMaterialType为document时有效-->txt
    </documentFormat>
    <pptFormat>
      <!--opt, enum, PPT格式, subType:string, [ppt#ppt,ptx#ptx], desc:当staticMaterialType为ppt时有效-->ppt
    </pptFormat>
    <docFormat>
      <!--opt, enum, 文档格式, subType:string, [doc#doc,docx#docx], desc:当staticMaterialType为doc时有效-->doc
    </docFormat>
    <excelFormat>
      <!--opt, enum, 表格格式, subType:string, [xls#xls,xlsx#xlsx], desc:当staticMaterialType为excel时有效-->xlsx
    </excelFormat>
    <pdfFormat>
      <!--opt, enum, pdf格式, subType:string, [pdf#pdf], desc:当staticMaterialType为pdf时有效-->pdf
    </pdfFormat>
    <webFormat>
      <!--opt, enum, 网页格式, subType:string, [html#html,htm#htm], desc:当staticMaterialType为web时有效-->htm
    </webFormat>
    <appFormat>
      <!--opt, enum, app格式, subType:string, [apk#apk], desc:当staticMaterialType为app时有效-->apk
    </appFormat>
    <fileSize>
      <!--req, int, 文件大小, unit:Byte, unitType:信息量-->1
    </fileSize>
    <duration>
      <!--opt, int, 素材本身播放时长, unit:s, unitType:时间, dep:and,{$.Material.StaticMaterial.staticMaterialType,eq,flash}, {$.Material.StaticMaterial.staticMaterialType,eq,video}, desc:当素材为视频和动画时有效-->1
    </duration>
    <staticMaterialurl>
      <!--opt, string, 存储服务器的url, desc:当Material->StorageInfo存在时有效, 素材在http存储服务器的url-->test
    </staticMaterialurl>
    <materialURL>
      <!--opt, string, 素材url, desc:门禁产品增加, 与staticMaterialUrl重复, 后续将迁移到staticMaterialUrl字段, 其他产品严禁使用该字段-->test
    </materialURL>
    <StorageInfo>
      <!--opt, object, 存储信息-->
```

```

<addressingFormatType>
  <!--req, enum, 地址类型, subType:string, [ipaddress#IP地址,hostname#主机名]-->ipaddress
</addressingFormatType>
<hostName>
  <!--opt, string, 主机名, dep:and,{$.Material.StaticMaterial.StorageInfo.addressingFormatType,eq,hostname}-->test
</hostName>
<ipAddress>
  <!--opt, string, ipv4地址, dep:and,{$.Material.StaticMaterial.StorageInfo.addressingFormatType,eq,ipaddress}-->test
</ipAddress>
<ipv6Address>
  <!--opt, string, ipv6地址, dep:and,{$.Material.StaticMaterial.StorageInfo.addressingFormatType,eq,ipaddress}-->test
</ipv6Address>
<port>
  <!--req, int, 端口号-->1
</port>
<accessKey>
  <!--opt, string, 密码, desc:AccessKey是云存储访问Key;Key都由云存储管理系统生成,资源池用户对应密钥;武汉云2.0引入",传入AK、SK,则文件采用普通下载方式,
否则仍然采用Token下载方式-->test
</accessKey>
<secretKey>
  <!--opt, string, 加密密码, desc:SecretKey是云存储协议加密Key,Key都由云存储管理系统生成,资源池用户对应密钥;武汉云2.0引入",传入AK、SK,则文件采用普通下
载方式,否则仍然采用Token下载方式-->test
</secretKey>
<bucket>
  <!--opt, string, 存储容器, desc:若上层没有配置,设备默认为bucket1-->test
</bucket>
</StorageInfo>
</StaticMaterial>
<DynamicMaterial>
  <!--opt, object, 动态素材信息, desc:当materialType 类型为dynamic时有效-->
  <dynamicMaterialType>
    <!--opt, enum, 动态素材类型, subType:string,
[web#web,socket#socket,rss#rss,realStream#realStream,generalData#generalData,picUrl#picUrl,dataSource#dataSource], desc:当DynamicMaterial节点存在时有效--
>dataSource
  </dynamicMaterialType>
  <webUrl>
    <!--opt, string, 网页内容素材的地址, desc:当dynamicMaterialType为web时有效-->test
  </webUrl>
  <rssUrl>
    <!--opt, string, rss新闻素材地址, desc:当dynamicMaterialType为rss时有效-->test
  </rssUrl>
  <picUrl>
    <!--opt, string, 图片url, desc:当dynamicMaterialType为picUrl时有效-->test
  </picUrl>
  <RealStream>
    <!--opt, object, dep 实时流素材, desc:当dynamicMaterialType为streamMedia时有效-->
    <destinationType>
      <!--req, enum, 取流终端类型, subType:string, [streamMedia#流媒体服务器,normalIPC#普通的IPC], desc:当RealStream节点存在时有效-->streamMedia
    </destinationType>
    <streamMediaUrl>
      <!--opt, string, 流媒体服务器URL, desc:当destinationType为streamMedia时有效-->test
    </streamMediaUrl>
    <NormalIPC>
      <!--opt, object, 普通IPC信息, desc:当destinationType为normalIPC时有效-->
      <ipAddress>
        <!--opt, object, IP地址, desc:当NormalIPC节点存在时有效-->
        <ipVersion>
          <!--req, enum, IP地址版本, subType:string, [v4#v4地址,v6#v6地址,dual#双]-->v4
        </ipVersion>
        <ipAddress>
          <!--opt, string, IPv4地址, desc:当ipVersion为v4时有效-->test
        </ipAddress>
        <ipv6Address>
          <!--opt, string, IPv6地址, desc:当ipVersion为v6时有效-->test
        </ipv6Address>
        </ipAddress>
        <portNo>
          <!--req, int, 端口号-->1
        </portNo>
        <channelNo>
          <!--req, int, 通道号-->1
        </channelNo>
        <userName>
          <!--opt, string, 登陆设备用户名, desc:传输时需要加密, 具体加密方案参见《ISAPI明文传输处理方案》-->test
        </userName>
        <password>
          <!--req, string, 登陆设备密码, desc:传输时需要加密, 具体加密方案参见《ISAPI明文传输处理方案》-->test
        </password>
        <transmitProtocol>
          <!--opt, enum, 传输码流协议, subType:string, [tcp#tcp,udp#udp,mcast#多播]-->tcp
        </transmitProtocol>
        <streamType>
          <!--opt, enum, 码流类型, subType:string, [main#主码流,sub#子码流,third#三码流]-->main
        </streamType>
        <iv>
          <!--opt, string, 向量, desc:服务器将素材xml文件下发给终端时或从服务器导出时需要-->test
        </iv>
        <secretKey>
          <!--opt, string, 密钥, desc:服务器将素材xml文件下发给终端时或从服务器导出时需要-->test
        </secretKey>
      </NormalIPC>
    <dataType>
      <!--opt, enum, 数据类型, subType:string, [capture#抓拍,LiveVideo#视频直播]-->capture
    </dataType>
  </RealStream>

```

```
</RealStream>
<GeneralData>
  <!--opt, object, 第三方通用数据, desc: 当dynamicMaterialType为generalData时有效-->
  <SrcAddress>
    <!--opt, object, 源IP, desc: 当GeneralData节点存在时有效-->
    <ipVersion>
      <!--req, enum, IP地址版本, subType:string, [v4#v4地址,v6#v6地址,dual#双]-->v4
    </ipVersion>
    <ipAddress>
      <!--opt, string, IPv4地址, desc: 当ipVersion为v4时有效-->test
    </ipAddress>
    <ipv6Address>
      <!--opt, string, IPv6地址, desc: 当ipVersion为v6时有效-->test
    </ipv6Address>
  </SrcAddress>
  <dataType>
    <!--opt, enum, 数据类型, subType:string, [popPic#弹图,calL#叫号]-->popPic
  </dataType>
</GeneralData>
</DynamicMaterial>
</Material>
```

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 响应消息, attr:version{ro, req, string, 版本}-->
  <requestURL>
    <!--ro, req, string, 请求的URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, 状态码, subType:int, [0#OK,1#OK,2#Device Busy,3#Device Error,4#Invalid Operation,5#Invalid XML Format,6#Invalid XML Content,7#Reboot Required]-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, 状态信息, subType:string, [OK#成功,Device Busy#设备忙,Device Error#设备异常,Invalid Operation#无效的操作,Invalid XML Format#无效的XML格式,Invalid XML Content#无效的XML内容,Reboot#设备重启]-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, 详细错误码英文描述, desc: 详细错误码的英文描述-->OK
  </subStatusCode>
  <ID>
    <!--ro, opt, string, 索引-->test
  </ID>
</ResponseStatus>
```

69.65.2 获取单个素材参数

Request URL

GET /ISAPI/Publish/MaterialMgr/material/<materialID>?security=<security>&iv=<iv>

查询参数

参数名称	参数类型	描述
materialID	string	
security	string	没有security表示数据不加密；security=1,表示报文中的accessKey和secretKey字段AES128加密；security=2,表示报文中的accessKey和secretKey字段AES256加密；
iv	string	随机字符串，长度限制为32；iv表示初始化向量，在security=1或者2时为必须参数；

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<Material xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 素材信息, attr:version{req, string, 协议版本}-->
  <id>
    <!--ro, req, int, 素材索引-->1
  </id>
  <seq>
    <!--ro, opt, int, 素材序列号, desc: 该字段仅在服务器到终端链路生效,修改一次素材, seq会变化一次-->0
  </seq>
</Material>
```

```
<!--ro, req, string, 素材名称-->test
</materialName>
<materialRemarks>
  <!--ro, req, string, 素材描述-->test
</materialRemarks>
<materialType>
  <!--ro, req, enum, 素材类型, subType:string, [static#静态,dynamic#动态]-->static
</materialType>
<approveState>
  <!--ro, opt, enum, 审核状态标志, subType:string, [approved#审核通过,notPass#未通过,notApprove#未审核]-->approved
</approveState>
<approveRemarks>
  <!--ro, opt, string, 审核意见-->test
</approveRemarks>
<shareProperty>
  <!--ro, opt, enum, 共享属性, subType:string, [public#共有,private#私有]-->public
</shareProperty>
<uploadUser>
  <!--ro, req, string, 上传用户-->test
</uploadUser>
<uploadTime>
  <!--ro, req, datetime, 上传时间-->1970-01-01T00:00:00+08:00
</uploadTime>
<materialEncrypt>
  <!--ro, opt, int, 素材校验串, desc:用来校验终端收到的素材正确性,该字段仅在服务器到终端链路生效,具体加密方式参见《信息发布素材加解密方案》,该方案文档可
  以从素材领域中获取到-->1
</materialEncrypt>
<orgNo>
  <!--ro, opt, int, 所属组织编号-->1
</orgNo>
<orgName>
  <!--ro, opt, string, 所属组织名称, desc:该字段只读-->test
</orgName>
<replaceTerminal>
  <!--ro, opt, bool, 素材是否更新到终端, desc:素材替换时该字段有效-->true
</replaceTerminal>
<StaticMaterial>
  <!--ro, opt, object, 静态素材-->
  <staticMaterialType>
    <!--ro, opt, enum, 静态素材类型, subType:string, [picture#图片,flash#flash,audio#音频,video#视频,document#文本,ppt#ppt,doc#文档,excel#表
    格,pdf#pdf,web#网页,app#app]-->picture
  </staticMaterialType>
  <picFormat>
    <!--ro, opt, enum, 图片格式, subType:string, [gif#gif,bmp#bmp,jpg#jpg,png#png]-->png
  </picFormat>
  <flashFormat>
    <!--ro, opt, enum, flash格式, subType:string, [swf#swf]-->swf
  </flashFormat>
  <audioFormat>
    <!--ro, opt, enum, 音频格式, subType:string, [mp3#mp3,wav#wav,wma#wma]-->wma
  </audioFormat>
  <videoFormat>
    <!--ro, opt, enum, 视频格式, subType:string, [rm#rm,rmvb#rmvb,asf#asf,avi#avi,mpg#mpg,3gp#3gp,mov#mov,mkv#mkv,wmv#wmv,flv#flv,mp4#mp4,hkps#hkps]-->mp4
  </videoFormat>
  <documentFormat>
    <!--ro, opt, enum, 文本格式, subType:string, [txt#txt]-->txt
  </documentFormat>
  <pptFormat>
    <!--ro, opt, enum, PPT格式, subType:string, [ppt#ppt,pptx#pptx]-->ppt
  </pptFormat>
  <docFormat>
    <!--ro, opt, enum, 文档格式, subType:string, [doc#doc,docx#docx]-->doc
  </docFormat>
  <excelFormat>
    <!--ro, opt, enum, 表格格式, subType:string, [xls#xls,xlsx#xlsx]-->xlsx
  </excelFormat>
  <pdfFormat>
    <!--ro, opt, enum, pdf格式, subType:string, [pdf#pdf]-->pdf
  </pdfFormat>
  <webFormat>
    <!--ro, opt, enum, 网页格式, subType:string, [html#html,htm#htm]-->htm
  </webFormat>
  <appFormat>
    <!--ro, opt, enum, app格式, subType:string, [apk#apk]-->apk
  </appFormat>
  <fileSize>
    <!--ro, req, int, 文件大小, unit:Byte, unitType:信息量-->1
  </fileSize>
  <duration>
    <!--ro, opt, int, 素材本身播放时长, unit:s, unitType:时间, dep:and,{$.Material.StaticMaterial.staticMaterialType,eq,flash},
    {$.Material.StaticMaterial.staticMaterialType,eq,video}-->1
  </duration>
  <uuid>
    <!--ro, opt, string, 服务器下发的UUID, desc:标识素材, 目前仅8520平台会将素材存储在存储服务器-->test
  </uuid>
  <staticMaterialurl>
    <!--ro, opt, string, 存储服务器的url, desc:当Material->StorageInfo存在时有效, 素材在http存储服务器的url-->test
  </staticMaterialurl>
  <materialURL>
    <!--ro, opt, string, 素材url, desc:门禁产品增加, 与staticMaterialUrl重复, 后续将迁移到staticMaterialURL字段, 其他产品严禁使用该字段-->test
  </materialURL>
  <StorageInfo>
```

```
<!--ro, opt, object, 存储信息-->
<addressingFormatType>
  <!--ro, req, enum, 地址类型, subType:string, [ipaddress#IP地址,hostname#主机名]-->ipaddress
</addressingFormatType>
<hostName>
  <!--ro, opt, string, 主机名, dep:and,{$.Material.StaticMaterial.StorageInfo.addressingFormatType,eq,hostname}-->test
</hostName>
<ipAddress>
  <!--ro, opt, string, ipv4地址, dep:and,{$.Material.StaticMaterial.StorageInfo.addressingFormatType,eq,ipaddress}-->test
</ipAddress>
<ipv6Address>
  <!--ro, opt, string, ipv6地址, dep:and,{$.Material.StaticMaterial.StorageInfo.addressingFormatType,eq,ipaddress}-->test
</ipv6Address>
<port>
  <!--ro, req, int, 端口号-->1
</port>
<accessKey>
  <!--ro, opt, string, 密码, desc:AccessKey是云存储访问Key;Key都由云存储管理系统生成,资源池用户对应密钥;武汉云2.0引入",传入AK、SK,则文件采用普通下载方式,否则仍然采用Token下载方式-->test
</accessKey>
<secretKey>
  <!--ro, opt, string, 加密密码, desc:SecretKey是云存储协议加密Key,Key都由云存储管理系统生成,资源池用户对应密钥;武汉云2.0引入",传入AK、SK,则文件采用普通下载方式,否则仍然采用Token下载方式-->test
</secretKey>
<bucket>
  <!--ro, opt, string, 存储容器, desc:若上层没有配置,设备默认为bucket1-->test
</bucket>
</StorageInfo>
</StaticMaterial>
<DynamicMaterial>
  <!--ro, opt, object, 动态素材信息-->
  <dynamicMaterialType>
    <!--ro, opt, enum, 动态素材类型, subType:string,
[web#web,socket#socket,rss#rss,realStream#realStream,generalData#generalData,picUrl#picUrl,dataSource#dataSource]-->dataSource
  </dynamicMaterialType>
  <webUrl>
    <!--ro, opt, string, 网页内容素材的地址-->test
  </webUrl>
  <rssUrl>
    <!--ro, opt, string, rss新闻素材地址-->test
  </rssUrl>
  <picUrl>
    <!--ro, opt, string, 图片url-->test
  </picUrl>
  <RealStream>
    <!--ro, opt, object, dep 实时流素材-->
    <destinationType>
      <!--ro, req, enum, 取流终端类型, subType:string, [streamMedia#流媒体服务器,normalIPC#普通的IPC,privateStreamMedia#私有流媒体]-->streamMedia
    </destinationType>
    <streamMediaUrl>
      <!--ro, opt, string, 流媒体服务器URL-->test
    </streamMediaUrl>
    <NormalIPC>
      <!--ro, opt, object, 普通IPC信息-->
      <ipAddress>
        <!--ro, opt, object, IP地址-->
        <ipVersion>
          <!--ro, req, enum, IP地址版本, subType:string, [v4#v4地址,v6#v6地址,dual#双]-->v4
        </ipVersion>
        <ipAddress>
          <!--ro, opt, string, IPv4地址-->test
        </ipAddress>
        <ipv6Address>
          <!--ro, opt, string, IPv6地址-->test
        </ipv6Address>
      </ipAddress>
      <portNo>
        <!--ro, req, int, 端口号-->1
      </portNo>
      <channelNo>
        <!--ro, req, int, 通道号-->1
      </channelNo>
      <userName>
        <!--ro, req, string, 登陆设备用户名, desc:传输时需要加密, 具体加密方案参见《ISAPI明文传输处理方案》-->test
      </userName>
      <password>
        <!--ro, req, string, 登陆设备密码, desc:传输时需要加密, 具体加密方案参见《ISAPI明文传输处理方案》-->test
      </password>
      <transmitProtocol>
        <!--ro, opt, enum, 传输码流协议, subType:string, [tcp#tcp,udp#udp,mcast#多播]-->tcp
      </transmitProtocol>
      <streamType>
        <!--ro, opt, enum, 码流类型, subType:string, [main#主码流,sub#子码流,third#三码流]-->main
      </streamType>
      <iv>
        <!--ro, opt, string, 向量, desc:服务器将素材xml文件下发给终端时或从服务器导出时需要-->test
      </iv>
      <secretKey>
        <!--ro, opt, string, 密钥, desc:服务器将素材xml文件下发给终端时或从服务器导出时需要-->test
      </secretKey>
    </NormalIPC>
    <privateStreamMediaUrl>
      <!--ro, opt, string, 海康私有的流媒体数据-->test
    </privateStreamMediaUrl>
```

```
</privateStreamMaterial>
<dataType>
  <!--ro, opt, enum, 数据类型, subType:string, [capture#抓拍,LiveVideo#视频直播]-->capture
</dataType>
</RealStream>
<GeneralData>
  <!--ro, opt, object, 第三方通用数据-->
  <SrcAddress>
    <!--ro, opt, object, 源IP-->
    <ipVersion>
      <!--ro, req, enum, IP地址版本, subType:string, [v4#v4地址,v6#v6地址,dual#双]-->v4
    </ipVersion>
    <ipAddress>
      <!--ro, opt, string, IPv4地址-->test
    </ipAddress>
    <ipv6Address>
      <!--ro, opt, string, IPv6地址-->test
    </ipv6Address>
  </SrcAddress>
  <dataType>
    <!--ro, opt, enum, 数据类型, subType:string, [popPic#弹图,call#叫号]-->popPic
  </dataType>
</GeneralData>
<dataSourceUrl>
  <!--ro, opt, string, 数据源在存储服务器url, desc:当动态素材类型为dataSource时有效, 目前仅8520平台存在数据源素材类型-->test
</dataSourceUrl>
</DynamicMaterial>
</Material>
```

69.65.3 配置单个素材参数

Request URL

PUT /ISAPI/Publish/MaterialMgr/material/<materialID>?security=<security>&iv=<iv>

查询参数

参数名称	参数类型	描述
materialID	string	
security	string	没有security表示数据不加密；security=1,表示报文中的accessKey和secretKey字段AES128加密；security=2,表示报文中的accessKey和secretKey字段AES256加密；
iv	string	随机字符串，长度限制为32；iv表示初始化向量，在security=1或者2时为必须参数；

请求报文

```
<?xml version="1.0" encoding="UTF-8"?>

<Material xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, 素材信息, attr:version{req, string, 协议版本}-->
  <id>
    <!--req, int, 素材索引-->1
  </id>
  <seq>
    <!--opt, int, 素材序列号, desc:该字段仅在服务器到终端链路生效,修改一次素材, seq会变化一次-->0
  </seq>
  <materialName>
    <!--req, string, 素材名称-->test
  </materialName>
  <materialRemarks>
    <!--req, string, 素材描述-->test
  </materialRemarks>
  <materialType>
    <!--req, enum, 素材类型, subType:string, [static#静态,dynamic#动态]-->static
  </materialType>
  <approveState>
    <!--opt, enum, 审核状态标志, subType:string, [approved#审核通过,notPass#未通过,notApprove#未审核]-->approved
  </approveState>
  <approveRemarks>
    <!--opt, string, 审核意见-->test
  </approveRemarks>
  <shareProperty>
    <!--opt, enum, 共享属性, subType:string, [public#共有,private#私有]-->public
  </shareProperty>
  <uploadUser>
    <!--req, string, 上传用户-->test
  </uploadUser>
  <uploadTime>
    <!--req, datetime, 上传时间-->1970-01-01T00:00:00+08:00
  </uploadTime>
  <materialEncrypt>
    <!--opt, int, 素材校验由, desc:用来校验终端收到的素材正确性.该字段仅在服务器到终端链路生效.且体加密方式参见《信息发布素材加密方案》.该方案文档可以从
```

素材领域中获取到-->1

```
</materialEncrypt>
<orgNo>
  <!--opt, int, 所属组织编号-->1
</orgNo>
<orgName>
  <!--opt, string, 所属组织名称, desc:该字段只读-->test
</orgName>
<replaceTerminal>
  <!--opt, bool, 素材是否更新到终端, desc:素材替换时该字段有效-->true
</replaceTerminal>
<staticMaterial>
  <!--opt, object, 静态素材-->
  <staticMaterialType>
    <!--opt, enum, 静态素材类型, subType:string, [picture#图片,flash#flash,audio#音频,video#视频,document#文本,ppt#ppt,doc#文档,excel#表格,pdf#pdf,web#网页,app#app]-->picture
  </staticMaterialType>
  <picFormat>
    <!--opt, enum, 图片格式, subType:string, [gif#gif,bmp#bmp,jpg#jpg,png#png]-->png
  </picFormat>
  <flashFormat>
    <!--opt, enum, flash格式, subType:string, [swf#swf]-->swf
  </flashFormat>
  <audioFormat>
    <!--opt, enum, 音频格式, subType:string, [mp3#mp3,wav#wav,wma#wma]-->wma
  </audioFormat>
  <videoFormat>
    <!--opt, enum, 视频格式, subType:string, [rm#rm,rmvb#rmvb,asf#asf,avi#avi,mpg#mpg,3gp#3gp,mov#mov,mkv#mkv,wmv#wmv,flv#flv,mp4#mp4,hkps#hkps]-->mp4
  </videoFormat>
  <documentFormat>
    <!--opt, enum, 文本格式, subType:string, [txt#txt]-->txt
  </documentFormat>
  <pptFormat>
    <!--opt, enum, PPT格式, subType:string, [ppt#ppt,pptx#pptx]-->ppt
  </pptFormat>
  <docFormat>
    <!--opt, enum, 文档格式, subType:string, [doc#doc,docx#docx]-->doc
  </docFormat>
  <excelFormat>
    <!--opt, enum, 表格格式, subType:string, [xls#xls,xlsx#xlsx]-->xlsx
  </excelFormat>
  <pdfFormat>
    <!--opt, enum, pdf格式, subType:string, [pdf#pdf]-->pdf
  </pdfFormat>
  <webFormat>
    <!--opt, enum, 网页格式, subType:string, [html#html,htm#htm]-->htm
  </webFormat>
  <appFormat>
    <!--opt, enum, app格式, subType:string, [apk#apk]-->apk
  </appFormat>
  <fileSize>
    <!--req, int, 文件大小, unit:Byte, unitType:信息量-->1
  </fileSize>
  <duration>
    <!--opt, int, 素材本身播放时长, unit:s, unitType:时间, dep:and,{$.Material.StaticMaterial.staticMaterialType,eq,flash},
    {$.Material.StaticMaterial.staticMaterialType,eq,video}-->1
  </duration>
  <uuid>
    <!--opt, string, 服务器下发的UUID, desc:标识素材, 目前仅8520平台会将素材存储在存储服务器-->test
  </uuid>
  <staticMaterialUrl>
    <!--opt, string, 存储服务器的url, desc:当Material->StorageInfo存在时有效, 素材在http存储服务器的url-->test
  </staticMaterialUrl>
  <materialURL>
    <!--opt, string, 素材url, desc:门禁产品增加, 与staticMaterialUrl重复, 后续将迁移到staticMaterialUrl字段, 其他产品严禁使用该字段-->test
  </materialURL>
  <StorageInfo>
    <!--opt, object, 存储信息-->
    <addressingFormatType>
      <!--req, enum, 地址类型, subType:string, [ipaddress#IP地址,hostname#主机名]-->ipaddress
    </addressingFormatType>
    <hostName>
      <!--opt, string, 主机名, dep:and,{$.Material.StaticMaterial.StorageInfo.addressingFormatType,eq,hostname}-->test
    </hostName>
    <ipAddress>
      <!--opt, string, ipv4地址, dep:and,{$.Material.StaticMaterial.StorageInfo.addressingFormatType,eq,ipaddress}-->test
    </ipAddress>
    <ipv6Address>
      <!--opt, string, ipv6地址, dep:and,{$.Material.StaticMaterial.StorageInfo.addressingFormatType,eq,ipaddress}-->test
    </ipv6Address>
    <port>
      <!--req, int, 端口号-->1
    </port>
    <accessKey>
      <!--opt, string, 密码, desc:AccessKey是云存储访问Key;Key都由云存储管理系统生成,资源池用户对应密钥;武汉云2.0引入", 传入AK、SK, 则文件采用普通下载方式,
      否则仍然采用Token下载方式-->test
    </accessKey>
    <secretKey>
      <!--opt, string, 加密密码, desc:SecretKey是云存储协议加密Key,Key都由云存储管理系统生成,资源池用户对应密钥;武汉云2.0引入", 传入AK、SK, 则文件采用普通
      下载方式, 否则仍然采用Token下载方式-->test
    </secretKey>
    <bucket>
      <!--opt, string, 存储容器, desc:若上层没有配置,设备默认为bucket1-->test
    </bucket>
  </StorageInfo>
</staticMaterial>
</material>
```

```

</bucket>
</StorageInfo>
</StaticMaterial>
<DynamicMaterial>
  <!--opt, object, 动态素材信息-->
  <dynamicMaterialType>
    <!--opt, enum, 动态素材类型, subType:string,
[web#web,socket#socket,rss#rss,realStream#realStream,generalData#generalData,picUrl#picUrl,dataSource#dataSource]-->dataSource
  </dynamicMaterialType>
  <webUrl>
    <!--opt, string, 网页内容素材的地址-->test
  </webUrl>
  <rssUrl>
    <!--opt, string, rss新闻素材地址-->test
  </rssUrl>
  <picUrl>
    <!--opt, string, 图片url-->test
  </picUrl>
  <RealStream>
    <!--opt, object, dep 实时流素材-->
    <destionType>
      <!--req, enum, 取流终端类型, subType:string, [streamMedia#流媒体服务器,normlIPC#普通的IPC,privateStreamMedia#私有流媒体]-->streamMedia
    </destionType>
    <streamMediaUrl>
      <!--opt, string, 流媒体服务器URL-->test
    </streamMediaUrl>
    <NormalIPC>
      <!--opt, object, 普通IPC信息-->
      <IpAddress>
        <!--opt, object, IP地址-->
        <ipVersion>
          <!--req, enum, IP地址版本, subType:string, [v4#v4地址,v6#v6地址,dual#双]-->v4
        </ipVersion>
        <ipAddress>
          <!--opt, string, IPv4地址-->test
        </ipAddress>
        <ipv6Address>
          <!--opt, string, IPv6地址-->test
        </ipv6Address>
      </IpAddress>
      <portNo>
        <!--req, int, 端口号-->1
      </portNo>
      <channelNo>
        <!--req, int, 通道号-->1
      </channelNo>
      <userName>
        <!--req, string, 登陆设备用户名, desc:传输时需要加密, 具体加密方案参见《ISAPI明文传输处理方案》-->test
      </userName>
      <password>
        <!--req, string, 登陆设备密码, desc:传输时需要加密, 具体加密方案参见《ISAPI明文传输处理方案》-->test
      </password>
      <transmitProtocol>
        <!--opt, enum, 传输码流协议, subType:string, [tcp#tcp,udp#udp,mcast#多播]-->tcp
      </transmitProtocol>
      <streamType>
        <!--opt, enum, 码流类型, subType:string, [main#主码流,sub#子码流,third#三码流]-->main
      </streamType>
      <iv>
        <!--opt, string, 向量, desc:服务器将素材xml文件下发给终端时或从服务器导出时需要-->test
      </iv>
      <secretKey>
        <!--opt, string, 密钥, desc:服务器将素材xml文件下发给终端时或从服务器导出时需要-->test
      </secretKey>
    </NormalIPC>
    <privateStreamMediaUrl>
      <!--opt, string, 海康私有的流媒体数据-->test
    </privateStreamMediaUrl>
    <dataType>
      <!--opt, enum, 数据类型, subType:string, [capture#抓拍,LiveVideo#视频直播]-->capture
    </dataType>
  </RealStream>
  <GeneralData>
    <!--opt, object, 第三方通用数据-->
    <SrcAddress>
      <!--opt, object, 源IP-->
      <ipVersion>
        <!--req, enum, IP地址版本, subType:string, [v4#v4地址,v6#v6地址,dual#双]-->v4
      </ipVersion>
      <ipAddress>
        <!--opt, string, IPv4地址-->test
      </ipAddress>
      <ipv6Address>
        <!--opt, string, IPv6地址-->test
      </ipv6Address>
    </SrcAddress>
    <dataType>
      <!--opt, enum, 数据类型, subType:string, [popPic#弹图,call#叫号]-->popPic
    </dataType>
  </GeneralData>
  <dataSourceUrl>
    <!--opt, string, 数据源在存储服务器url, desc:当动态素材类型为dataSource时有效, 目前仅8520平台存在数据源素材类型-->test
  </dataSourceUrl>
</DynamicMaterial>

```

```
</DynamicParameterId>  
</Material>
```

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>  
  
<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">  
  <!--ro, req, object, 响应消息, attr:version{ro, req, string, 版本}-->  
  <requestURL>  
    <!--ro, req, string, 请求的URL-->null  
  </requestURL>  
  <statusCode>  
    <!--ro, req, enum, 状态码, subType:int, [0#OK,1#OK,2#Device Busy,3#Device Error,4#Invalid Operation,5#Invalid XML Format,6#Invalid XML Content,7#Reboot Required]-->0  
  </statusCode>  
  <statusString>  
    <!--ro, req, enum, 状态信息, subType:string, [OK#成功,Device Busy#设备忙,Device Error#设备异常,Invalid Operation#无效的操作,Invalid XML Format#无效的XML格式,Invalid XML Content#无效的XML内容,Reboot#设备重启]-->OK  
  </statusString>  
  <subStatusCode>  
    <!--ro, req, string, 详细错误码英文描述, desc: 详细错误码的英文描述-->OK  
  </subStatusCode>  
</ResponseStatus>
```

69.65.4 获取全部素材参数

Request URL

GET /ISAPI/Publish/MaterialMgr/material?security=<security>&iv=<iv>

查询参数

参数名称	参数类型	描述
security	string	没有security表示数据不加密；security=1,表示报文中的accessKey和secretKey字段AES128加密；security=2,表示报文中的accessKey和secretKey字段AES256加密；
iv	string	随机字符串，长度限制为32；iv表示初始化向量，在security=1或者2时为必须参数；

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>  
  
<MaterialList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">  
  <!--ro, req, array, 素材信息列表, subType:object, attr:version{req, string, 协议版本}-->  
  <Material>  
    <!--ro, opt, object, 素材信息-->  
    <id>  
      <!--ro, req, int, 素材索引-->1  
    </id>  
    <materialName>  
      <!--ro, req, string, 素材名称-->test  
    </materialName>  
    <materialRemarks>  
      <!--ro, req, string, 素材描述-->test  
    </materialRemarks>  
    <materialType>  
      <!--ro, req, enum, 素材类型, subType:string, [static#静态,dynamic#动态]-->static  
    </materialType>  
    <approveState>  
      <!--ro, opt, enum, 审核状态标志, subType:string, [approved#审核通过,notPass#未通过,notApprove#未审核]-->approved  
    </approveState>  
    <approveRemarks>  
      <!--ro, opt, string, 审核意见-->test  
    </approveRemarks>  
    <shareProperty>  
      <!--ro, opt, enum, 共享属性, subType:string, [public#共有,private#私有]-->public  
    </shareProperty>  
    <uploadUser>  
      <!--ro, req, string, 上传用户-->test  
    </uploadUser>  
    <uploadTime>  
      <!--ro, req, datetime, 上传时间-->1970-01-01T00:00:00+08:00  
    </uploadTime>  
    <materialEncrypt>  
      <!--ro, opt, string, 素材校验串, desc:用来校验终端收到的素材正确性,该字段仅在服务器到终端链路生效,具体加密方式参见《信息发布素材加解密方案》, 该方案文
```

档可以从素材领域获取到-->jJEmNTA3NDg5NCY8jI3OTM5MjAmYWEhMzYyOTM5OCZhMCY8MjAzMDQwJmI1JjQzMzc3ODgmNDg=

```
</materialEncrypt>
<orgNo>
  <!--ro, opt, int, 所属组织编号-->1
</orgNo>
<replaceTerminal>
  <!--ro, opt, bool, 素材是否更新到终端, desc: 素材替换时该字段有效-->true
</replaceTerminal>
<StaticMaterial>
  <!--ro, opt, object, 静态素材-->
  <staticMaterialType>
    <!--ro, opt, enum, 静态素材类型, subType:string, [picture#图片, flash#flash, audio#音频, video#视频, document#文本, ppt#ppt, doc#文档, excel#表格, pdf#pdf, web#网页, app#app]-->picture
  </staticMaterialType>
  <picFormat>
    <!--ro, opt, enum, 图片格式, subType:string, [gif#gif, bmp#bmp, jpg#jpg, png#png]-->png
  </picFormat>
  <flashFormat>
    <!--ro, opt, enum, flash格式, subType:string, [swf#swf]-->swf
  </flashFormat>
  <audioFormat>
    <!--ro, opt, enum, 音频格式, subType:string, [mp3#mp3, wav#wav, wma#wma]-->wma
  </audioFormat>
  <videoFormat>
    <!--ro, opt, enum, 视频格式, subType:string, [rm#rm, rmvb#rmvb, asf#asf, avi#avi, mpg#mpg, 3gp#3gp, mov#mov, mkv#mkv, wmv#wmv, flv#flv, mp4#mp4, h264#h264, h265#h265]-->mp4
  </videoFormat>
  <documentFormat>
    <!--ro, opt, enum, 文本格式, subType:string, [txt#txt]-->txt
  </documentFormat>
  <pptFormat>
    <!--ro, opt, enum, PPT格式, subType:string, [ppt#ppt, pptx#pptx]-->ppt
  </pptFormat>
  <docFormat>
    <!--ro, opt, enum, 文档格式, subType:string, [doc#doc, docx#docx]-->doc
  </docFormat>
  <excelFormat>
    <!--ro, opt, enum, 表格格式, subType:string, [xls#xls, xlsx#xlsx]-->xlsx
  </excelFormat>
  <pdfFormat>
    <!--ro, opt, enum, pdf格式, subType:string, [pdf#pdf]-->pdf
  </pdfFormat>
  <webFormat>
    <!--ro, opt, enum, 网页格式, subType:string, [html#html, htm#htm]-->html
  </webFormat>
  <appFormat>
    <!--ro, opt, enum, app格式, subType:string, [apk#apk]-->apk
  </appFormat>
  <fileSize>
    <!--ro, req, int, 文件大小, unit:Byte, unitType:信息量-->1
  </fileSize>
  <duration>
    <!--ro, opt, int, 素材本身播放时长, unit:s, unitType:时间, dep:and, {$.Material.StaticMaterial.staticMaterialType, eq, flash}, {$.Material.StaticMaterial.staticMaterialType, eq, video}-->1
  </duration>
  <staticMaterialUrl>
    <!--ro, opt, string, 存储服务器的url, desc: 当Material->StorageInfo存在时有效, 素材在http存储服务器的url-->test
  </staticMaterialUrl>
  <StorageInfo>
    <!--ro, opt, object, 存储信息-->
    <addressingFormatType>
      <!--ro, req, enum, 地址类型, subType:string, [ipaddress#IP地址, hostname#主机名]-->ipaddress
    </addressingFormatType>
    <hostName>
      <!--ro, opt, string, 主机名, dep:and, {$.Material.StaticMaterial.StorageInfo.addressingFormatType, eq, hostname}-->test
    </hostName>
    <ipAddress>
      <!--ro, opt, string, ipv4地址, dep:and, {$.Material.StaticMaterial.StorageInfo.addressingFormatType, eq, ipaddress}-->test
    </ipAddress>
    <ipv6Address>
      <!--ro, opt, string, ipv6地址, dep:and, {$.Material.StaticMaterial.StorageInfo.addressingFormatType, eq, ipaddress}-->test
    </ipv6Address>
    <port>
      <!--ro, req, int, 端口号-->1
    </port>
    <accessKey>
      <!--ro, opt, string, 密码, desc: AccessKey是云存储访问Key; Key都由云存储管理系统生成, 资源池用户对应密钥; 武汉云2.0引入", 传入AK、SK, 则文件采用普通下载方式, 否则仍然采用Token下载方式-->test
    </accessKey>
    <secretKey>
      <!--ro, opt, string, 加密密码, desc: SecretKey是云存储协议加密Key; Key都由云存储管理系统生成, 资源池用户对应密钥; 武汉云2.0引入", 传入AK、SK, 则文件采用普通下载方式, 否则仍然采用Token下载方式-->test
    </secretKey>
    <bucket>
      <!--ro, opt, string, 存储容器, desc: 若上层没有配置, 设备默认为bucket1-->test
    </bucket>
  </StorageInfo>
  <materialURL>
    <!--ro, opt, string, 素材url, desc: 门禁产品增加, 与staticMaterialUrl重复, 后续将迁移到staticMaterialUrl字段, 其他产品严禁使用该字段-->test
  </materialURL>
</StaticMaterial>
<DynamicMaterial>
  <!--ro, opt, object, 动态素材信息-->
  <dynamicMaterialType>
```

```

<dynamicMaterialType>
  <!--ro, opt, enum, 动态素材类型, subType:string,
[web#web,socket#socket,rss#rss,realStream#realStream,generalData#generalData,picUrl#picUrl,dataSource#dataSource]-->dataSource
</dynamicMaterialType>
<webUrl>
  <!--ro, opt, string, 网页内容素材的地址-->test
</webUrl>
<rssUrl>
  <!--ro, opt, string, rss新闻素材地址-->test
</rssUrl>
<picUrl>
  <!--ro, opt, string, 图片url-->test
</picUrl>
<RealStream>
  <!--ro, opt, object, dep 实时流素材-->
  <destionType>
    <!--ro, req, enum, 取流终端类型, subType:string, [streamMedia#流媒体服务器,normaLIPC#普通的IPC]-->streamMedia
  </destionType>
  <streamMediaUrl>
    <!--ro, opt, string, 流媒体服务器URL-->test
  </streamMediaUrl>
  <NormalIPC>
    <!--ro, opt, object, 普通IPC信息-->
    <IpAddress>
      <!--ro, opt, object, IP地址-->
      <ipVersion>
        <!--ro, req, enum, IP地址版本, subType:string, [v4#v4地址,v6#v6地址,dual#双]-->v4
      </ipVersion>
      <ipAddress>
        <!--ro, opt, string, IPv4地址-->test
      </ipAddress>
      <ipv6Address>
        <!--ro, opt, string, IPv6地址-->test
      </ipv6Address>
    </IpAddress>
    <portNo>
      <!--ro, req, int, 端口号-->1
    </portNo>
    <channelNo>
      <!--ro, req, int, 通道号-->1
    </channelNo>
    <userName>
      <!--ro, req, string, 登陆设备用户名, desc:传输时需要加密, 具体加密方案参见《ISAPI明文传输处理方案》-->test
    </userName>
    <password>
      <!--ro, req, string, 登陆设备密码, desc:传输时需要加密, 具体加密方案参见《ISAPI明文传输处理方案》-->test
    </password>
    <transmitProtocol>
      <!--ro, opt, enum, 传输码流协议, subType:string, [tcp#tcp,udp#udp,mcast#多播]-->tcp
    </transmitProtocol>
    <streamType>
      <!--ro, opt, enum, 码流类型, subType:string, [main#主码流,sub#子码流,third#三码流]-->main
    </streamType>
    <iv>
      <!--ro, opt, string, 向量, desc:服务器将素材xml文件下发给终端时或从服务器导出时需要-->test
    </iv>
    <secretKey>
      <!--ro, opt, string, 密钥, desc:服务器将素材xml文件下发给终端时或从服务器导出时需要-->test
    </secretKey>
  </NormalIPC>
  <dataType>
    <!--ro, opt, enum, 数据类型, subType:string, [capture#抓拍,LiveVideo#视频直播]-->capture
  </dataType>
</RealStream>
<GeneralData>
  <!--ro, opt, object, 第三方通用数据-->
  <SrcAddress>
    <!--ro, opt, object, 源IP-->
    <ipVersion>
      <!--ro, req, enum, IP地址版本, subType:string, [v4#v4地址,v6#v6地址,dual#双]-->v4
    </ipVersion>
    <ipAddress>
      <!--ro, opt, string, IPv4地址-->test
    </ipAddress>
    <ipv6Address>
      <!--ro, opt, string, IPv6地址-->test
    </ipv6Address>
  </SrcAddress>
  <dataType>
    <!--ro, opt, enum, 数据类型, subType:string, [popPic#弹图,call#叫号]-->popPic
  </dataType>
</GeneralData>
</DynamicMaterial>
</Material>
</MaterialList>

```

69.65.5 上传静态素材

Request URL

POST /ISAPI/Publish/MaterialMgr/material/<materialID>/upload

查询参数

参数名称	参数类型	描述
materialID	string	素材ID

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 响应消息, attr:version{ro, req, string, 版本}-->
  <requestURL>
    <!--ro, req, string, 请求的URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, 状态码, subType:int, [0#OK,1#OK,2#Device Busy,3#Device Error,4#Invalid Operation,5#Invalid XML Format,6#Invalid XML Content,7#Reboot Required]-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, 状态信息, subType:string, [OK#成功,Device Busy#设备忙,Device Error#设备异常,Invalid Operation#无效的操作,Invalid XML Format#无效的XML格式,Invalid XML Content#无效的XML内容,Reboot#设备重启]-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, 详细错误码英文描述, desc:详细错误码的英文描述-->OK
  </subStatusCode>
</ResponseStatus>
```

69.65.6 获取素材管理参数能力

Request URL

GET /ISAPI/Publish/material/<materialID>/capabilities

查询参数

参数名称	参数类型	描述
materialID	string	

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>
<Material xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, 素材管理参数, attr:version{req, string, 协议版本}-->
  <id>
    <!--ro, req, int, 素材索引-->1
  </id>
  <materialName max="10">
    <!--ro, req, string, 素材名称, attr:max{req, int, 最大值}-->test
  </materialName>
  <materialRemarks max="10">
    <!--ro, req, string, 素材描述, attr:max{req, int, 最大值}-->test
  </materialRemarks>
  <materialType opt="static,dynamic">
    <!--ro, req, enum, 素材类型, subType:string, [static#静态,dynamic#动态], attr:opt{req, string, 取值范围}-->static
  </materialType>
  <StaticMaterial>
    <!--ro, opt, object, 静态素材-->
    <staticMaterialType opt="picture,flash,audio,video,document,ppt,doc,excel,pdf,web">
      <!--ro, opt, enum, 静态素材类型, subType:string, [picture#图片,flash#flash,audio#音频,video#视频,document#文本,ppt#ppt,doc#文档,excel#表格,pdf#pdf,web#网页], attr:opt{req, string, 取值范围}-->picture
    </staticMaterialType>
    <picFormat opt="gif,bmp,jpg,png">
      <!--ro, opt, enum, 图片格式, subType:string, [gif#gif,bmp#bmp,jpg#jpg,png#png], attr:opt{req, string, 取值范围}-->png
    </picFormat>
    <picResolution>
      <!--ro, opt, object, 图片分辨率限制-->
      <height min="0" max="10">
        <!--ro, opt, int, 高度, attr:min{req, int, 最小值},max{req, int, 最大值}-->0
      </height>
      <width min="0" max="10">
        <!--ro, opt, int, 宽度, attr:min{req, int, 最小值},max{req, int, 最大值}-->0
      </width>
    </picResolution>
    <fileSize max="10">
      <!--ro, req, int, 文件大小, unit:Byte, unitType:信息量, attr:max{req, int, 最大值}-->1
    </fileSize>
    <picFileSize max="10">
      <!--ro, opt, int, 图片文件大小, unit:Byte, unitType:信息量, attr:max{req, int, 最大值}-->0
    </picFileSize>
    <materialURL max="10">
      <!--ro, opt, string, 素材URL, attr:max{req, int, 最大值}, desc:门禁产品增加, 与staticMaterialUrl重复, 后续将迁移到staticMaterialUrl字段, 其他产品严禁使用该字段-->test
    </materialURL>
  </StaticMaterial>
</Material>
```

69.65.7 删除单个素材

Request URL

DELETE /ISAPI/Publish/MaterialMgr/material/<materialID>

查询参数

参数名称	参数类型	描述
materialID	string	素材ID

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 响应消息, attr:version{ro, req, string, 版本}-->
  <requestURL>
    <!--ro, req, string, 请求的URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, 状态码, subType:int, [0#OK,1#OK,2#Device Busy,3#Device Error,4#Invalid Operation,5#Invalid XML Format,6#Invalid XML Content,7#Reboot Required]-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, 状态信息, subType:string, [OK#成功,Device Busy#设备忙,Device Error#设备异常,Invalid Operation#无效的操作,Invalid XML Format#无效的XML格式,Invalid XML Content#无效的XML内容,Reboot#设备重启]-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, 详细错误码英文描述, desc:详细错误码的英文描述-->OK
  </subStatusCode>
</ResponseStatus>
```

69.65.8 批量删除素材

Request URL

PUT /ISAPI/Publish/MaterialMgr/material/batchDelete

查询参数

无

请求报文

```
<?xml version="1.0" encoding="UTF-8"?>

<MaterialIdList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--opt, array, 素材ID列表, subType:object, attr:version{req, string, 协议版本}-->
  <materialId>
    <!--req, int, 素材ID-->1
  </materialId>
</MaterialIdList>
```

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 响应消息, attr:version{ro, req, string, 版本}-->
  <requestURL>
    <!--ro, req, string, 请求的URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, 状态码, subType:int, [0#OK,1#OK,2#Device Busy,3#Device Error,4#Invalid Operation,5#Invalid XML Format,6#Invalid XML Content,7#Reboot Required]-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, 状态信息, subType:string, [OK#成功,Device Busy#设备忙,Device Error#设备异常,Invalid Operation#无效的操作,Invalid XML Format#无效的XML格式,Invalid XML Content#无效的XML内容,Reboot#设备重启]-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, 详细错误码英文描述, desc:详细错误码的英文描述-->OK
  </subStatusCode>
</ResponseStatus>
```

69.65.9 获取素材查找参数能力

Request URL

GET /ISAPI/Publish/MaterialMgr/materialSearch/profile

查询参数

无

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<MaterialSearchProfile xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, 未定义, attr:version{req, string, 协议版本}-->
  <maxTimeSpansNum>
    <!--ro, req, int, 支持的最大时间段数-->1
  </maxTimeSpansNum>
  <maxMatchResults>
    <!--ro, req, int, 一次搜索返回的最大匹配结果个数-->1
  </maxMatchResults>
  <maxSearchTimeout>
    <!--ro, opt, int, 搜索超时时间-->1
  </maxSearchTimeout>
  <maxConcurrentSearches>
    <!--ro, opt, int, 支持的最大并发搜索个数-->1
  </maxConcurrentSearches>
  <approveState opt="approved,notPass,notApprove,all">
    <!--ro, opt, enum, 审核状态类型, subType:string, [approved#已审核通过,notPass#未通过,notApprove#未审核,all#全部], attr:opt{req, string, 取值范围}-->all
  </approveState>
  <materialType opt="static,dynamic,all">
    <!--ro, opt, enum, 素材类型, subType:string, [static#静态,dynamic#动态,all#全部], attr:opt{req, string, 取值范围}-->all
  </materialType>
  <staticMaterialType opt=" picture,flash,audio,video,document,ppt,doc,excel,pdf,web,all">
    <!--ro, opt, enum, 静态素材类型, subType:string,
[picture#picture,flash#flash,audio#audio,video#video,document#document,ppt#ppt,doc#doc,excel#excel,pdf#pdf,web#web,all#全部], attr:opt{req, string, 取值范围}-->all
  </staticMaterialType>
  <dynamicMaterialType opt=" web,socket,rss,realStream,generalData,all">
    <!--ro, opt, enum, 动态素材类型, subType:string, [web#web,socket#socket,rss#rss,realStream#realStream,generalData#generalData,all#全部], attr:opt{req, string, 取值范围}-->all
  </dynamicMaterialType>
  <realStreamType opt="streamMedia,normlIPC">
    <!--ro, opt, enum, 实时流类型, subType:string, [streamMedia#流媒体服务器,normlIPC#普通的IPC], attr:opt{req, string, 取值范围}-->streamMedia
  </realStreamType>
  <shareProperty opt="public,private,all">
    <!--ro, opt, enum, 共享属性类型, subType:string, [public#公有,private#私有,all#所有], attr:opt{req, string, 取值范围}-->all
  </shareProperty>
  <isSupportUploader>
    <!--ro, opt, bool, 是否支持按上传者条件搜索-->true
  </isSupportUploader>
  <isSupportMaterialName>
    <!--ro, opt, bool, 是否支持按素材名搜索-->true
  </isSupportMaterialName>
  <isSupportMaterialNameLike>
    <!--ro, opt, bool, 是否支持按素材名模糊搜索-->true
  </isSupportMaterialNameLike>
  <isSupportMaterialRemarksLike>
    <!--ro, opt, bool, 是否支持按素材描述模糊搜索-->true
  </isSupportMaterialRemarksLike>
  <isSupportOrgName>
    <!--ro, opt, bool, 是否支持按组织搜索-->true
  </isSupportOrgName>
  <isSupportSubOrg>
    <!--ro, opt, bool, 是否支持子组织-->true
  </isSupportSubOrg>
  <isSupportSort>
    <!--ro, opt, bool, 是否支持排序-->true
  </isSupportSort>
  <isSupportKeyword>
    <!--ro, opt, bool, 是否支持关键字搜索-->true
  </isSupportKeyword>
  <generalDataType opt="popPic,call,all">
    <!--ro, opt, enum, 三方数据类型, subType:string, [popPic#弹图,call#叫号,all#所有], attr:opt{req, string, 取值范围}-->all
  </generalDataType>
  <streamDataType opt="capture,liveVideo,all">
    <!--ro, opt, enum, 流数据类型, subType:string, [capture#抓拍,liveVideo#视频直播,all#所有], attr:opt{req, string, 取值范围}-->all
  </streamDataType>
</MaterialSearchProfile>
```

69.66 节目管理

69.66.1 查询节目

Request URL

POST /ISAPI/Publish/ProgramMgr/programSearch

查询参数

无

请求报文

```
<?xml version="1.0" encoding="UTF-8"?>

<ProgramSearchDescription xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--opt, object, 查询条件, attr:version{req, string, 协议版本}-->
  <searchID>
    <!--req, string, 搜索记录唯一标识, desc:用来标识是否是同一个搜索, 设备端先匹配此搜索ID, 未匹配则认为是新搜索, searchID存活周期为5min-->test
  </searchID>
  <approveState>
    <!--opt, enum, 审核状态标志, subType:string, [approved#审核通过,notPass#未通过,notApprove#未审核,all#所有]-->approved
  </approveState>
  <shareProperty>
    <!--req, enum, 共享属性, subType:string, [public#共有,private#私有,all#所有]-->public
  </shareProperty>
  <uploader>
    <!--opt, string, 上传用户名, desc:无此节点表示不限上传者-->test
  </uploader>
  <Resolution>
    <!--req, object, 节目的分辨率-->
    <resolutionName>
      <!--opt, enum, 分辨率名称, subType:string, [1920*1080#1920*1080,1080*1920#1080*1920,selfDefine#自定义]-->1920*1080
    </resolutionName>
    <imageWidth>
      <!--req, int, 分辨率宽-->1
    </imageWidth>
    <imageHeight>
      <!--req, int, 分辨率高-->1
    </imageHeight>
  </Resolution>
  <TimeSpanList>
    <!--opt, array, 节目修改的时间范围列表, subType:object, desc:不填此节点表示不限修改时间-->
    <TimeSpan>
      <!--opt, object, 时间段-->
      <startTime>
        <!--req, datetime, 开始时间-->1970-01-01T00:00:00+08:00
      </startTime>
      <endTime>
        <!--req, datetime, 结束时间-->1970-01-01T00:00:00+08:00
      </endTime>
    </TimeSpan>
  </TimeSpanList>
  <maxResults>
    <!--opt, int, 本次搜索允许返回的最大结果个数-->1
  </maxResults>
  <searchResultsPosition>
    <!--opt, int, 本次搜索的起始结果索引值, desc:当搜索的结果总数大于一次返回的结果数时, 客户端可以用相同的searchID继续搜索, 指定起始搜索结果的索引值。(跳过前多少个)-->1
  </searchResultsPosition>
</ProgramSearchDescription>
```

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<ProgramSearchResult xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, 查询结果, attr:version{req, string, 协议版本}-->
  <searchID>
    <!--ro, req, string, 搜索记录唯一标识, desc:用来确认上层客户端是否为同一个(倘若为同一个,则设备记录内存,下次搜索加快速度)-->test
  </searchID>
  <responseStatus>
    <!--ro, req, bool, 查询结果-->true
  </responseStatus>
  <responseStatusString>
    <!--ro, req, enum, 查询状态字符串描述, subType:string, [OK#查询结束,MORE#还有数据等待查询,NO_MATCH#没有匹配数据,PARAM_ERROR#参数错误,TIMEOUT#超时,FAILED#失败]-->OK
  </responseStatusString>
  <totalMatches>
    <!--ro, req, int, 符合条件的记录总条数-->1
  </totalMatches>
  <numOfMatches>
    <!--ro, req, int, 本次返回的记录条数-->1
  </numOfMatches>
  <MatchElementList>
    <!--ro, opt, array, 匹配的节目列表, subType:object-->
    <MatchElement>
      <!--ro, req, object, 匹配的节目信息-->
      <id>
        <!--ro, req, int, 节目索引号-->1
      </id>
      <programName>
        <!--ro, req, string, 节目名称-->test
      </programName>
      <programRemarks>
        <!--ro, req, string, 节目注释-->test
      </programRemarks>
      <shareProperty>
        <!--ro, req, enum, 共享属性, subType:string, [public#共有,private#私有]-->public
      </shareProperty>
      <approveState>
        <!--ro, opt, enum, 审核状态标志, subType:string, [approved#审核通过,notPass#未通过,notApprove#未审核]-->approved
      </approveState>
      <approveRemarks>
        <!--ro, opt, string, 审核说明-->test
      </approveRemarks>
      <Resolution>
        <!--ro, req, object, 节目的分辨率-->
        <resolutionName>
          <!--ro, opt, enum, 分辨率名称, subType:string, [1920*1080#1920*1080,1080*1920#1080*1920,selfDefine#自定义]-->test
        </resolutionName>
        <imageWidth>
          <!--ro, req, int, 分辨率宽-->1
        </imageWidth>
        <imageHeight>
          <!--ro, req, int, 分辨率高-->1
        </imageHeight>
      </Resolution>
      <programSize>
        <!--ro, req, int, 节目总大小, unit:Byte, unitType:信息量-->1
      </programSize>
      <programLength>
        <!--ro, req, int, 节目总时长, unit:seconds-->1
      </programLength>
      <moder>
        <!--ro, opt, string, 复用者-->test
      </moder>
      <modeTime>
        <!--ro, opt, datetime, 修改时间-->1970-01-01T00:00:00+08:00
      </modeTime>
      <orgName>
        <!--ro, opt, string, 组织名称-->test
      </orgName>
      <programType>
        <!--ro, opt, enum, 节目类型, subType:string, [normal#普通,decode#解码,touch#触摸,decodeTouch#解码触摸]-->normal
      </programType>
    </MatchElement>
  </MatchElementList>
</ProgramSearchResult>
```

69.66.2 获取节目查找参数能力

Request URL

GET /ISAPI/Publish/ProgramMgr/programSearch/profile

查询参数

无

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<ProgramSearchProfile xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, opt, object, 节目查找配置参数, attr:version{req, string, 协议版本}-->
  <maxTimeSpansNum>
    <!--ro, req, int, 支持的最大时间段数-->1
  </maxTimeSpansNum>
  <maxMatchResults>
    <!--ro, req, int, 一次搜索返回的最大匹配结果个数-->1
  </maxMatchResults>
  <maxSearchTimeout>
    <!--ro, opt, int, 搜索超时时间-->1
  </maxSearchTimeout>
  <maxConcurrentSearches>
    <!--ro, opt, int, 支持的最大并发搜索个数-->1
  </maxConcurrentSearches>
  <approveState opt="approved,notPass,notApprove,all">
    <!--ro, opt, enum, 审核状态类型, subType:string, [approved#已审核通过,notPass#未通过,notApprove#未审核,all#全部], attr:opt{req, string, 取值范围}-->all
  </approveState>
  <shareProperty opt="public,private,all">
    <!--ro, opt, enum, 共享属性类型, subType:string, [public#公有,private#私有,all#所有], attr:opt{req, string, 取值范围}-->all
  </shareProperty>
  <isSupportUploader>
    <!--ro, opt, bool, 是否支持按上传者条件搜索-->true
  </isSupportUploader>
  <isSupportResolution>
    <!--ro, opt, bool, 是否支持按分辨率条件搜索-->true
  </isSupportResolution>
  <isSupportProgramName>
    <!--ro, opt, bool, 是否支持按节目名搜索-->true
  </isSupportProgramName>
  <isSupportProgramNameLike>
    <!--ro, opt, bool, 是否支持按节目名模糊搜索-->true
  </isSupportProgramNameLike>
  <isSupportProgramRemarksLike>
    <!--ro, opt, bool, 是否支持按节目描述模糊搜索-->true
  </isSupportProgramRemarksLike>
  <isSupportModer>
    <!--ro, opt, bool, 是否支持按修改者搜索-->true
  </isSupportModer>
  <isSupportOrgName>
    <!--ro, opt, bool, 是否支持按组织名称搜索-->true
  </isSupportOrgName>
  <programType opt="normal,decode,touch,decodeTouch">
    <!--ro, opt, enum, 节目类型, subType:string, [normal#普通,decode#解码,touch#触摸,decodeTouch#解码触摸], attr:opt{req, string, 取值范围}-->normal
  </programType>
  <maxResolutionNum>
    <!--ro, opt, int, 支持最大分辨率数-->1
  </maxResolutionNum>
  <isSupportSubOrg>
    <!--ro, opt, bool, 是否支持子组织-->true
  </isSupportSubOrg>
  <isSupportSort>
    <!--ro, opt, bool, 是否支持排序-->true
  </isSupportSort>
  <isSupportKeyword>
    <!--ro, opt, bool, 是否支持关键字搜索-->true
  </isSupportKeyword>
  <isSptSearchByID>
    <!--ro, opt, bool, 是否支持通过ID搜索-->true
  </isSptSearchByID>
</ProgramSearchProfile>
```

69.66.3 获取节目参数配置能力

Request URL

GET /ISAPI/Publish/ProgramMgr/program/<programID>/capabilities

查询参数

参数名称	参数类型	描述
programID	string	节目ID

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>
<Program xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 节目参数, attr:version{req, string, 协议版本}-->
  <id>
    <!--ro, req, int, 节目索引-->1
  </id>
  <programName max="10">
    <!--ro, req, string, 节目名称, attr:max{req, int, 最大值}-->test
  </programName>
  <programRemarks max="10">
    <!--ro, req, string, 节目描述, attr:max{req, int, 最大值}-->test
  </programRemarks>
  <programType opt="normal,decode,touch,decodeTouch,doubleSide,conference,screensaver,character,mainInterface,idleScreen,txt">
    <!--ro, opt, enum, 节目类型, subType:string, [normal#普通,decode#解码,touch#触摸,decodeTouch#解码触摸,doubleSide#双面屏,conference#会议,screensaver#图片(门禁图片广告),character#欢迎词(门禁欢迎词广告),mainInterface#主界面(门禁),idleScreen#待机画面(门禁),txt#文本节目(门禁)], attr:opt{req, string, 取值范围}-->normal
  </programType>
  <Resolution>
    <!--ro, req, object, 分辨率-->
    <resolutionName opt="1920*1080,1080*1920,3840*2160,2160*3840,selfDefine">
      <!--ro, opt, string, 分辨率名称, attr:opt{req, string, 取值范围}-->test
    </resolutionName>
    <imageWidth min="1" max="10">
      <!--ro, req, int, 分辨率宽, attr:min{req, int, 最小值},max{req, int, 最大值}-->1
    </imageWidth>
    <imageHeight min="1" max="10">
      <!--ro, req, int, 分辨率高, attr:min{req, int, 最小值},max{req, int, 最大值}-->1
    </imageHeight>
  </Resolution>
  <PageList size="10">
    <!--ro, opt, array, 页面列表, subType:object, attr:size{req, int, 数组元素总数}-->
    <Page>
      <!--ro, opt, object, 页面-->
      <id>
        <!--ro, req, int, 索引-->1
      </id>
      <PageBasicInfo>
        <!--ro, req, object, 页面基本信息-->
        <pageName max="10">
          <!--ro, opt, string, 页面名称, attr:max{req, int, 最大值}-->test
        </pageName>
        <switchDuration min="1" max="10">
          <!--ro, req, int, 切换间隔, unit:s, unitType:时间, attr:min{req, int, 最小值},max{req, int, 最大值}-->1
        </switchDuration>
        <switchEffect
          opt="none,random,boxShrink,boxSpread,cycleShrink,cycSpread,eraseUp,eraseDown,eraseLeft,eraseRight,verticalShelter,horizontalShelter,verticalChessboard,horizontalChessboard,dissolve,leftRightToCenter,ceterToLeftRight,upDownToCenter,centerToUpDown,drawOutLeftDown,drawOutLeftUp,drawOutRightDown,drawOutRightUp,verticalLine,horizontalLine">
            <!--ro, req, enum, 切换效果, subType:string, [none#无,random#随机,boxShrink#盒状收缩,boxSpread#盒状展开,cycleShrink#圆形收缩,cycSpread#圆形展开,eraseUp#向上擦除,eraseDown#向下擦除,eraseLeft#向左擦除,eraseRight#向右擦除,verticalShelter#垂直屏蔽,horizontalShelter#水平屏蔽,verticalChessboard#纵向棋盘式,horizontalChessboard#横向棋盘式,dissolve#随机溶解,leftRightToCenter#左右向中间缩进,ceterToLeftRight#中央向左右扩展,upDownToCenter#上下向中央缩进,centerToUpDown#中央向上下扩展,drawOutLeftDown#从左上抽出,drawOutLeftUp#从左上抽出,drawOutRightDown#从右下抽出,drawOutRightUp#从右上抽出,verticalLine#随机垂直线条,horizontalLine#随机垂直线条], attr:opt{req, string, 取值范围}-->none
          </switchEffect>
        </PageBasicInfo>
        <WindowsList size="10">
          <!--ro, opt, array, 窗口信息列表, subType:object, attr:size{req, int, 数组元素总数}-->
          <Windows>
            <!--ro, opt, object, 窗口信息-->
            <id>
              <!--ro, req, int, 内容编号-->1
            </id>
            <Position>
              <!--ro, opt, object, 内容位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
              <positionX min="1" max="10">
                <!--ro, req, int, x坐标, attr:min{req, int, 最小值},max{req, int, 最大值}-->1
              </positionX>
              <positionY min="1" max="10">
                <!--ro, req, int, y坐标, attr:min{req, int, 最小值},max{req, int, 最大值}-->1
              </positionY>
              <height min="1" max="10">
                <!--ro, req, int, 宽, attr:min{req, int, 最小值},max{req, int, 最大值}-->1
              </height>
              <width min="1" max="10">
                <!--ro, req, int, 高, attr:min{req, int, 最小值},max{req, int, 最大值}-->1
              </width>
            </Position>
            <layerNo min="1" max="10">
              <!--ro, req, int, 图层编号, attr:min{req, int, 最小值},max{req, int, 最大值}-->1
            </layerNo>
            <WinMaterialInfo>
              <!--ro, opt, object, 窗口素材信息-->
              <materialType opt="static,dynamic,other">
                <!--ro, req, enum, 素材类型, subType:string, [static#静态,dynamic#动态,other#其他], attr:opt{req, string, 取值范围}-->static
              </materialType>
              <staticMaterialType opt="picture,flash,audio,video,document,ppt,doc,excel,pdf,web,app,signIn,conference">
                <!--ro, opt, enum, 静态素材类型, subType:string, [picture#图片,flash#flash,audio#音频,video#视频,document#文本,ppt#ppt,doc#文档,excel#表格,pdf#pdf,web#网页,app#app,signIn#签到,conference#会议], attr:opt{req, string, 取值范围}-->picture
              </staticMaterialType>
            </WinMaterialInfo>
            <PlayItemList size="10">
              <!--ro, opt, array, 窗口播放列表, subType:object, attr:size{req, int, 数组元素总数}-->
              <PlayItem
```

```

<!--ro, req, object, 窗口播放信息-->
<id>
<!--ro, req, int, 播放序号-->1
</id>
<materialNo>
<!--ro, opt, int, 素材编号-->1
</materialNo>
<playEffect opt="none,marquee">
<!--ro, req, enum, 播放效果, subType:string, [none#无,marquee#跑马灯], attr:opt{req, string, 取值范围}-->none
</playEffect>
<PlayDuration>
<!--ro, opt, object, 素材播放时间, desc:静态素材可以配置该时间,视频直播和IPC通道都可配置该节点-->
<durationType opt="materialTime,selfDefine">
<!--ro, req, enum, 持续方式, subType:string, [materialTime#按素材时间,selfDefine#自定义], attr:opt{req, string, 取值范围}-->selfDefine
</durationType>
<duration min="1" max="10" default="null">
<!--ro, req, int, 持续时间, unit:s, unitType:时间, attr:min{req, int, 最小值},max{req, int, 最大值},default{opt, string, 默认值}-->1
</duration>
</PlayDuration>
</PlayItem>
</PlayItemList>
</Windows>
</WindowsList>
</Page>
</PageList>
<coordinateType opt="uniformCoordinate,resolutionCoordinate">
<!--ro, opt, enum, 节目中坐标基准值类型, subType:string, [uniformCoordinate#统一坐标(实际的分辨率坐标按基准坐标等比例进行转换),resoLutionCoordinate#实际分辨率坐标], attr:opt{req, string, 取值范围}-->uniformCoordinate
</coordinateType>
<PageIdList size="10">
<!--ro, opt, array, 页面id列表, subType:object, attr:size{req, int, 数组元素总数}-->
<PageId>
<!--ro, opt, object, 页面id-->
<id min="1" max="10">
<!--ro, req, int, 页面ID, attr:min{req, int, 最小值},max{req, int, 最大值}-->1
</id>
</PageId>
</PageIdList>
</Program>

```

69.66.4 获取节目动态能力

Request URL

GET /ISAPI/Publish/ProgramMgr/program/dynamicCap

查询参数

无

请求报文

无

响应报文

```

<?xml version="1.0" encoding="UTF-8"?>

<ProgramDynamicCap xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
<!--ro, opt, object, 能力, attr:version{req, string, 协议版本}-->
<ProgramCapList>
<!--ro, req, array, 节目能力列表, subType:object-->
<ProgramCap>
<!--ro, req, object, 节目能力-->
<id>
<!--ro, req, int, 索引-->1
</id>
<programType>
<!--ro, req, enum, 节目类型, subType:string, [normal#标准,decode#解码,touch#触摸,decodeTouch#解码触摸,doubleSide#双面屏,conference#会议,screensaver#图片(门禁广告),character#欢迎词(门禁广告),mainInterface#主界面(门禁),idleScreen#待机画面(门禁),txt#文本节目(门禁)]-->normal
</programType>
<maxPageNum>
<!--ro, req, int, 最大页面个数-->1
</maxPageNum>
<maxWinNum>
<!--ro, req, int, 每个页面最大窗口个数-->1
</maxWinNum>
<maxMaterialNum>
<!--ro, req, int, 每个窗口最大素材个数-->1
</maxMaterialNum>
<isSupportTouch>
<!--ro, req, bool, 是否支持触摸, desc:如果支持触摸,节目里面可以配置触摸属性-->true
</isSupportTouch>
<WinCapList>
<!--ro, opt, array, 窗口能力列表, subType:object-->
<WinCap>
<!--ro, opt, object, 窗口能力-->
<id>
<!--ro, req, int, 序号-->1

```

```

</id>
<BaseRes>
  <!--ro, req, object, 基础分辨率-->
  <resWidth>
    <!--ro, req, int, 分辨率宽-->1
  </resWidth>
  <resHeight>
    <!--ro, req, int, 分辨率高-->1
  </resHeight>
</BaseRes>
<WinSizeList>
  <!--ro, req, array, 窗口大小列表, subType:object-->
  <winSize>
    <!--ro, opt, object, 窗口高范围-->
    <id>
      <!--ro, req, int, 索引-->1
    </id>
    <winMaterialInfo>
      <!--ro, req, object, 窗口素材信息-->
      <materialType opt="static,dynamic">
        <!--ro, req, enum, 素材类型, subType:string, [static#静态,dynamic#动态,other#其他], attr:opt(req, string, 取值范围)-->static
      </materialType>
      <staticMaterialType opt="picture,flash,audio,video,document,ppt,doc,excel,pdf,web">
        <!--ro, opt, enum, 静态素材类型, subType:string, [picture#图片,flash#flash,audio#音频,video#视频,document#文本,ppt#ppt,doc#文档,excel#表格,pdf#pdf,web#网页], attr:opt(req, string, 取值范围)-->picture
      </staticMaterialType>
      <dynamicType opt="web,socket,rss,call,dynamicPic,realStream,capturePic,character">
        <!--ro, opt, enum, 动态素材类型, subType:string,
[web#web,socket#socket,rss#rss,realStream#realStream,call#call,dynamicPic#dynamicPic,capturePic#capturePic,character#文本], attr:opt(req, string, 取值范围)-->web
      </dynamicType>
      <otherType
opt="clock,weather,countdown,localInput,hyperLinkBtn,callBtn,openDoorBtn,QRCodeBtn,authenticationSuccess,authenticationFail,statusBar">
        <!--ro, opt, enum, 其他素材类型, subType:string, [clock#时钟,weather#天气,countdown#倒计时,localInput#本地输入,hyperLinkBtn#超链接按钮,callBtn#呼叫按键,openDoorBtn#开门按键,QRCodeBtn#二维码按键,authenticationSuccess#认证成功,authenticationFail#认证失败,statusBar#状态栏], attr:opt(req, string, 取值范围)-->clock
      </otherType>
    </winMaterialInfo>
    <maxWinNum>
      <!--ro, opt, int, 在某个分辨率下此种窗口的最大个数-->1
    </maxWinNum>
    <width min="1" max="10">
      <!--ro, opt, int, 窗口宽范围, attr:min(req, int, 最小值),max(req, int, 最大值),
desc:由于门禁产品窗口固定, 此时min max应该为相同值;
此处返回的width按统一坐标类型返回, 统一坐标类型是指实际的分辨率坐标按基准坐标1920做等比例转换, 即如果屏幕宽的实际分辨率为1080, 窗口宽占据540个像素点, 则width应该填写为540/1080 * 1920-->1
    </width>
    <height min="1" max="10">
      <!--ro, opt, int, 窗口高范围, attr:min(req, int, 最小值),max(req, int, 最大值),
desc:由于门禁产品窗口固定, 此时min max应该为相同值;
此处返回的height按统一坐标类型返回, 统一坐标类型是指实际的分辨率坐标按基准坐标1920做等比例转换, 即如果屏幕高的实际分辨率为1080, 窗口高占据540个像素点, 则height应该填写为540/1080 * 1920-->1
    </height>
    <DefaultSize>
      <!--ro, opt, object, 窗口默认大小-->
      <defaultWidth>
        <!--ro, req, int, 窗口默认大小宽-->1
      </defaultWidth>
      <defaultHeight>
        <!--ro, req, int, 窗口默认大小高-->1
      </defaultHeight>
    </DefaultSize>
    <x min="0" max="10">
      <!--ro, opt, int, 窗口左上角x坐标, attr:min(req, int, 最小值),max(req, int, 最大值),
desc:由于门禁产品窗口固定, 此时min max应该为相同值;
门禁产品增加, 此处返回的x按统一坐标类型返回, 字段不返回表示不做min max限制-->1
    </x>
    <y min="0" max="10">
      <!--ro, opt, int, 窗口左上角y坐标, attr:min(req, int, 最小值),max(req, int, 最大值),
desc:由于门禁产品窗口固定, 此时min max应该为相同值;
门禁产品增加, 此处返回的y按统一坐标类型返回, 字段不返回表示不做min max限制-->1
    </y>
    <characterType>
      <!--ro, opt, enum, 文字类型, subType:string, [mainTitle#主标题,subTitle#副标题,subTitle2#副标题2], desc:当素材类型为character时可选有效, 门禁产品增加, 用于返回不同标题在设备上的固定位置信息【由于设备当前仅支持固定窗口位置, 因此增加该字段, 待设备支持自定义窗口位置后, 该字段后续废弃】-->mainTitle
    </characterType>
    <characterMode>
      <!--ro, opt, enum, 文字模式, subType:string, [mode1#模式1,mode2#模式2,mode3#模式3], desc:当素材类型为character时可选有效, 用于返回不同文字模式下, 主标题、副标题、副标题2窗口的位置-->mode1
    </characterMode>
    <fontSize min="1" max="10">
      <!--ro, opt, int, 字体大小, unit:px, unitType:图像显示, attr:min(req, int, 最小值),max(req, int, 最大值), desc:当素材类型为character时可选有效, 门禁增加, 用于返回不同文字模式, 不同文字类型下, 字体的大小范围-->1
    </fontSize>
    <characterContent min="1" max="512">
      <!--ro, opt, string, 文字内容, range:[0,512], attr:min(req, int, 最小值),max(req, int, 最大值), desc:当素材类型为character时可选有效, 门禁增加, 用于返回不同文字模式, 不同文字类型下, 字体的内容长度范围-->test
    </characterContent>
  </winSize>
</WinSizeList>
<maxCharacterWinNum>
  <!--ro, opt, int, 最大文字窗口个数-->1
</maxCharacterWinNum>

```

```
</WinCap>
</WinCapList>
</ProgramCap>
</ProgramCapList>
</ProgramDynamicCap>
```

69.66.5 创建单个节目

Request URL

POST /ISAPI/Publish/ProgramMgr/program

查询参数

无

请求报文

```
<?xml version="1.0" encoding="UTF-8"?>

<Program xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--req, object, 节目信息, attr:version{req, string, 协议版本}-->
  <id>
    <!--req, int, 节目索引-->1
  </id>
  <programName>
    <!--req, string, 节目名称-->test
  </programName>
  <programRemarks>
    <!--req, string, 节目描述-->test
  </programRemarks>
  <shareProperty>
    <!--opt, enum, 共享属性, subType:string, [public#公有,private#私有]-->public
  </shareProperty>
  <approveState>
    <!--opt, enum, 审核状态标志, subType:string, [approved#审核通过,notPass#未通过,notApprove#未审核]-->approved
  </approveState>
  <approveRemarks>
    <!--opt, string, 审核意见-->test
  </approveRemarks>
  <programType>
    <!--opt, enum, 节目类型, subType:string, [normal#普通,decode#解码,touch#触摸,decodeTouch#解码触摸,doubleSide#双面屏,conference#会议,screensaver#图片/屏保(门禁),character#欢迎词(门禁),mainInterface#主界面(门禁),txt#文本节目(门禁),advertise#广告], desc:当节目类型为conference#会议,screensaver#图片/屏保(门禁),character#欢迎词(门禁),mainInterface#主界面(门禁),txt#文本节目(门禁),advertise#广告时,该节目不依赖与日程,直接单独下发该节目后,在设备上展示展示该节目下的页面信息.-->normal
  </programType>
  <orgNo>
    <!--opt, int, 所属组织编号-->1
  </orgNo>
  <Resolution>
    <!--req, object, 分辨率-->
    <resolutionName>
      <!--opt, string, 分辨率名称-->test
    </resolutionName>
    <imageWidth>
      <!--req, int, 分辨率宽-->1
    </imageWidth>
    <imageHeight>
      <!--req, int, 分辨率高-->1
    </imageHeight>
  </Resolution>
  <PageList>
    <!--ro, opt, object, 页面列表-->
    <Page>
      <!--ro, req, object, 页面-->
      <id>
        <!--ro, req, int, 索引-->1
      </id>
      <PageBasicInfo>
        <!--ro, req, object, 页面基本信息-->
        <pageName>
          <!--ro, opt, string, 页面名称-->test
        </pageName>
        <BackgroundColor>
          <!--ro, opt, object, 背景颜色-->
          <RGB>
            <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
          </RGB>
        </BackgroundColor>
        <playDurationMode>
          <!--ro, opt, enum, 页面播放时间模式, subType:string, [selfDefine#自定义,auto#自动,loop#循环], desc:自定义时playDuration节点有效;自动时根据播放内容的时长*playCount计算;循环时表示一直循环-->selfDefine
        </playDurationMode>
        <playDuration>
          <!--ro, opt, int, 页面播放时长, unit:s, unitType:时间-->1
        </playDuration>
        <playCount>
          <!--ro, opt, int, 页面播放次数-->1
        </playCount>
      </Page>
    </PageList>
```

```
<switchDuration>
  <!--ro, opt, int, 切换间隔, unit:s, unitType:时间-->1
</switchDuration>
<switchEffect>
  <!--ro, opt, enum, 切换效果, subType:string, [none#无,random#随机,boxShrink#盒状收缩,boxSpread#盒状展开,cycleShrink#圆形收缩,cycSpread#圆形展开,eraseUp#向上擦除,eraseDown#向下擦除,eraseLeft#向左擦除,eraseRight#向右擦除,verticalShelter#垂直屏蔽,horizontalShelter#水平屏蔽,verticalChessboard#纵向棋盘式,horizontalChessboard#横向棋盘式,dissolve#随机溶解,LeftRightToCenter#左右向中间缩进,ceterToLeftRight#中央向左右扩展,upDownToCenter#上下向中央缩进,centerToUpDown#中央向上下扩展,drawOutLeftDown#从左下抽出,drawOutLeftUp#从左上抽出,drawOutRightDown#从右下抽出,drawOutRightUp#从右上抽出,verticalLine#随机垂直线条,horizontalLine#随机垂直线条]-->none
</switchEffect>
<backgroundPic>
  <!--ro, opt, int, 背景图片素材ID-->1
</backgroundPic>
<SignInCfg>
  <!--ro, opt, object, 签到信息, desc:当页面用于签到或考勤界面时有效-->
<VoiceBroadcast>
  <!--ro, opt, object, 语音播报-->
  <enabled>
    <!--ro, opt, bool, 是否启用语音播报-->true
  </enabled>
  <broadcastSpeed>
    <!--ro, opt, enum, 播报语速, subType:string, [slower#慢,slow#较慢,medium#中速,fast#较快,faster#快]-->slower
  </broadcastSpeed>
  <inLibVoice>
    <!--ro, opt, string, 在库人员正常体温语音播报内容-->test
  </inLibVoice>
  <outLibVoice>
    <!--ro, opt, string, 未知人员正常体温语音播报内容-->test
  </outLibVoice>
  <inLibAbnormalTemperatureVoice>
    <!--ro, opt, string, 在库人员异常温度语音播报内容-->test
  </inLibAbnormalTemperatureVoice>
  <outLibAbnormalTemperatureVoice>
    <!--ro, opt, string, 未知人员异常温度语音播报内容-->test
  </outLibAbnormalTemperatureVoice>
</VoiceBroadcast>
<WelcomeWord>
  <!--ro, opt, object, 欢迎词-->
  <enabled>
    <!--ro, opt, bool, 是否启用欢迎词-->true
  </enabled>
  <inLibWord>
    <!--ro, opt, string, 在库人员欢迎词内容-->test
  </inLibWord>
  <outLibWord>
    <!--ro, opt, string, 未知人员欢迎词内容-->test
  </outLibWord>
</WelcomeWord>
<SignCartoon>
  <!--ro, opt, object, 签到动画-->
  <signCartoonSpeed>
    <!--ro, opt, enum, 动画速度, subType:string, [fast#快,medium#终端,slow#慢]-->fast
  </signCartoonSpeed>
</SignCartoon>
<backgroundSignEnabled>
  <!--ro, opt, bool, 是否启用后台签到-->true
</backgroundSignEnabled>
<TemperatureDetection>
  <!--ro, opt, object, 温度监测参数-->
  <enabled>
    <!--ro, req, bool, 是否启用-->true
  </enabled>
  <threshold>
    <!--ro, req, float, 正常温度值-->0.000
  </threshold>
</TemperatureDetection>
</SignInCfg>
</PageBasicInfo>
<characterMode>
  <!--ro, opt, enum, 页面中的欢迎词模式, subType:string, [mode1#模式1,mode2#模式2,mode3#模式3], desc:门禁增加, 门禁欢迎词节目位置固定, 且存在三种模式-->mode1
</characterMode>
<windowsList>
  <!--ro, opt, array, 窗口信息列表, subType:object-->
  <windows>
    <!--ro, opt, object, 窗口信息-->
    <id>
      <!--ro, req, int, 内容编号-->1
    </id>
    <position>
      <!--ro, opt, object, 内容位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
      <positionX>
        <!--ro, req, int, x坐标-->1
      </positionX>
      <positionY>
        <!--ro, req, int, y坐标-->1
      </positionY>
      <height>
        <!--ro, req, int, 宽-->1
      </height>
      <width>
        <!--ro, req, int, 高-->1
      </width>
```

```
</Position>
<CornerRadius>
  <!--ro, opt, object, 圆角配置, desc:目前仅对picture图片窗口和character文字窗口有效-->
  <TopLeft>
    <!--ro, opt, int, 左上角半径-->1
  </TopLeft>
  <TopRight>
    <!--ro, opt, int, 右上角半径-->1
  </TopRight>
  <BottomLeft>
    <!--ro, opt, int, 左下角半径-->1
  </BottomLeft>
  <BottomRight>
    <!--ro, opt, int, 右下角半径-->1
  </BottomRight>
</CornerRadius>
<layerNo>
  <!--ro, req, int, 图层编号-->1
</layerNo>
<winMaterialInfo>
  <!--ro, opt, object, 窗口素材信息-->
  <materialType>
    <!--ro, req, enum, 素材类型, subType:string, [static#静态,dynamic#动态,other#其他]-->static
  </materialType>
  <staticMaterialType>
    <!--ro, opt, enum, 静态素材类型, subType:string, [picture#图片,flash#flash,audio#音频,video#视频,document#文本,ppt#ppt,doc#文档,excel#表格,pdf#pdf,web#网页,app#app,signIn#签到,conference#会议]-->picture
  </staticMaterialType>
  <dynamicType>
    <!--ro, opt, enum, 动态素材类型, subType:string,
[web#web,socket#socket,rss#rss,realStream#realStream,call#call,dynamicPic#dynamicPic,capturePic#capturePic,character#character]-->web
  </dynamicType>
  <otherType>
    <!--ro, opt, enum, 其他素材类型, subType:string, [cLock#时钟,weather#天气,countdown#倒计时,localInput#本地输入,hyperLinkBtn#超链接按钮,event#事件,callBtn#呼叫按键,openDoorBtn#开门按键,QRCODEBtn#二维码按键,authenticationSuccess#认证成功,authenticationFail#认证失败,statusBar#状态栏]-->cLock
  </otherType>
</winMaterialInfo>
<TouchProperty>
  <!--ro, opt, object, 触摸属性-->
  <windType>
    <!--ro, opt, enum, 窗口类型, subType:string, [popup#弹窗,page#页面]-->popup
  </windType>
  <hyperLinkType>
    <!--ro, opt, enum, 超链接类型, subType:string, [window#页面超链接,page#窗口超链接], desc:窗口类型为弹窗时有效-->window
  </hyperLinkType>
  <windowId>
    <!--ro, opt, int, 窗口编号, desc:当前页面的窗口,超链接类型为窗口时有效-->1
  </windowId>
  <pageId>
    <!--ro, opt, int, 页面编号, desc:超链接类型为页面时有效-->1
  </pageId>
</TouchProperty>
<PlayItemList>
  <!--ro, opt, array, 窗口播放列表, subType:object-->
  <PlayItem>
    <!--ro, req, object, 窗口播放信息-->
    <id>
      <!--ro, req, int, 播放序号-->1
    </id>
    <materialNo>
      <!--ro, opt, int, 素材编号-->1
    </materialNo>
    <inputChannel>
      <!--ro, opt, int, 绑定的IPC通道号-->1
    </inputChannel>
    <playEffect>
      <!--ro, req, enum, 播放效果, subType:string, [none#无,marquee#跑马灯]-->none
    </playEffect>
    <MarqueeInfo>
      <!--ro, opt, object, 跑马灯信息-->
      <scrollType>
        <!--ro, req, enum, 跑马灯滚动类型, subType:string, [none#不滚动,Loops#循环滚动,once#滚动一次,backAndForth#来回滚动]-->none
      </scrollType>
      <scrollDeriction>
        <!--ro, req, enum, 跑马灯滚动方向, subType:string, [none#无,up#向上,down#向下,Left#向左,right#向右]-->none
      </scrollDeriction>
      <scrollSpeed>
        <!--ro, req, int, 跑马灯滚动速度-->1
      </scrollSpeed>
    </MarqueeInfo>
  </PlayItem>
</PlayItemList>
<PlayDuration>
  <!--ro, opt, object, 窗口内素材播放时间, desc:静态素材可以配置该时间,视频直播和IPC通道都可配置该节点-->
  <durationType>
    <!--ro, req, enum, 持续方式, subType:string, [materialTime#按素材时间,selfDefine#自定义]-->selfDefine
  </durationType>
  <duration>
    <!--ro, req, int, 持续时间, unit:s, unitType:时间-->1
  </duration>
</PlayDuration>
<CharactersEffect>
  <!--ro, opt, object, 文字显示效果, desc:当素材类型为文本txt和时有效-->
  <fontSize>
    <!--ro, req, int, 字体大小-->1
  </fontSize>
```

```
<FontColor>
  <!--ro, req, object, 字体颜色-->
  <RGB>
    <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
  </RGB>
</FontColor>
<backgroundType>
  <!--ro, opt, enum, 背景类型, subType:string, [color#颜色(默认),picture#图片], desc:字段不存在时,默认为颜色-->color
</backgroundType>
<backPicId>
  <!--ro, opt, int, 背景图片素材ID-->1
</backPicId>
<BackColor>
  <!--ro, req, object, 背景颜色-->
  <RGB>
    <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
  </RGB>
</BackColor>
<backTransparent>
  <!--ro, req, int, 背景透明度-->1
</backTransparent>
<subtitlesEnabled>
  <!--ro, req, bool, 子标题使能-->true
</subtitlesEnabled>
<scrollDirection>
  <!--ro, req, enum, 文字滚动方向, subType:string, [Left#向左,right#向右,up#向上,down#向下]-->left
</scrollDirection>
<scrollSpeed>
  <!--ro, req, int, 文字滚动速度-->1
</scrollSpeed>
<CharactersEffect>
<switchEffect>
  <!--ro, opt, enum, 窗口素材切换效果, subType:string, [none#无,LeftInRightOut#左进右出,rightInLeftOut#右进左出,bottomInTopOut#下进上出,topInBottomOut#上进下出,fadeInFadeOut#淡入淡出,middleExit#中间退出,topPop#顶部弹出,rightBottomIn#右下角进入,LeftTopIn#左上角进入,horizontalOpen#水平打开,verticalOpen#垂直打开,random#随机效果], desc:图片素材时有效-->random
</switchEffect>
<pageTime>
  <!--ro, opt, int, 翻页时间, unit:s, unitType:时间, desc:当素材为word,ppt,pdf,excel时有效-->1
</pageTime>
<scrollSpeed>
  <!--ro, req, int, 滚动速度, desc:当素材为静态web时有效-->1
</scrollSpeed>
<scaleType>
  <!--ro, opt, enum, 素材缩放参数配置, subType:string, [fitXY#拉伸填充,fitCenter#自适应,centerCrop#截取填充], desc:目前仅当图片窗口有效-->centerCrop
</scaleType>
<playType>
  <!--ro, opt, enum, 播放方式, subType:string, [normal#正常,faceAttendance#考勤,PeopleCounting#客流,temperature#测温], desc:目前仅当图片窗口有效-->normal
</playType>
<CharactersAttribute>
  <!--ro, opt, object, 文字属性, desc:当dynamicType为character时有效-->
  <fontSize>
    <!--ro, req, int, 字体大小-->1
  </fontSize>
  <FontColor>
    <!--ro, req, object, 字体颜色-->
    <RGB>
      <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
    </RGB>
  </FontColor>
  <backgroundType>
    <!--ro, req, enum, 背景类型, subType:string, [color#颜色,picture#图片]-->color
  </backgroundType>
  <BackColor>
    <!--ro, req, object, 背景色-->
    <RGB>
      <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
    </RGB>
  </BackColor>
  <backTransparent>
    <!--ro, opt, int, 背景透明度-->1
  </backTransparent>
  <alignType>
    <!--ro, req, enum, 对齐方式, subType:string, [Left#左,right#右,middle#中]-->middle
  </alignType>
  <verticalAlignType>
    <!--ro, req, enum, 垂直对齐方式, subType:string, [top#顶,bottom#底,verticalCenter#中]-->top
  </verticalAlignType>
  <characterContent>
    <!--ro, opt, string, 文字内容, range:[0,512]-->test
  </characterContent>
  <backPicId>
    <!--ro, opt, int, 背景图片素材ID-->1
  </backPicId>
  <characterStyle>
    <!--ro, req, enum, 字体效果, subType:string, [normal#常规,bold#粗体]-->normal
  </characterStyle>
  <characterType>
    <!--ro, req, enum, 字体类型, subType:string, [mainTitle#主标题,subTitle#副标题]-->mainTitle
  </characterType>
</CharactersAttribute>
<WebProperty>
```

```

        <!--ro, opt, object, 网页属性-->
        <webType>
        <!--ro, opt, enum, 网页类型, subType:string, [desktop#桌面版,mobile#移动版], desc:默认按照设备的系统类型（Windows或Android）来显示网页--
>desktop
        </webType>
    </WebProperty>
    </PlayItem>
</PlayItemList>
<enabledAudio>
    <!--ro, opt, bool, 音频使能-->true
</enabledAudio>
<enableHide>
    <!--ro, opt, bool, 隐藏使能-->true
</enableHide>
<enableLock>
    <!--ro, opt, bool, 锁定使能-->true
</enableLock>
<AppWindow>
    <!--ro, opt, object, App窗口信息-->
    <WindowInfoList>
        <!--ro, req, array, 窗口信息列表, subType:object-->
        <WindowInfo>
            <!--ro, req, object, 窗口信息-->
            <id>
                <!--ro, req, int, 索引-->1
            </id>
            <materialNo>
                <!--ro, req, int, 素材编号, desc:采用本字段下发时, 用户需要先将App安装包当做素材存储到设备中, 得到素材编号再进行下发-->1
            </materialNo>
            <path>
                <!--ro, opt, string, App路径, range:[0,1024],
desc:path和materialNo字段同时存在时, 设备优先解析path字段;
采用本字段下发时, 设备需要确保App已经安装在设备内, 直接传入App的路径即可;
对于Android设备, path传的是App的包名; 对于Windows设备, path传的是App的应用程序绝对路径; -->test
            </path>
        </WindowInfo>
    </WindowInfoList>
</AppWindow>
<DataSource>
    <!--ro, opt, object, 数据来源, desc:当窗口为叫号和动态弹图时有效-->
    <materialNo>
        <!--ro, req, int, 素材编号-->1
    </materialNo>
</DataSource>
<Call>
    <!--ro, opt, object, 叫号数据-->
    <materialNo>
        <!--ro, req, int, 素材编号-->1
    </materialNo>
    <tableRow>
        <!--ro, req, int, 表格行-->1
    </tableRow>
    <tableColumn>
        <!--ro, req, int, 表格列-->1
    </tableColumn>
    <tableDirection>
        <!--ro, req, enum, 表格方向, subType:string, [vertical#纵向,horizontal#纵向]-->vertical
    </tableDirection>
    <tableType>
        <!--ro, req, enum, 表格类型, subType:string, [template1#模板1,template2#模板2,template3#模板3,template4#模板4,template5#模板5,template6#模板
6]-->template1
    </tableType>
    <backPicId>
        <!--ro, opt, int, 背景图片素材编号-->1
    </backPicId>
    <alignType>
        <!--ro, req, enum, 对齐方式, subType:string, [left#左,right#右,middle#中]-->left
    </alignType>
    <refreshDirection>
        <!--ro, req, enum, 刷新方向, subType:string, [upTdown#从上到下,downToup#从下到上,LeftToright#从左到右,rightToLeft#从右到左]-->upTdown
    </refreshDirection>
    <HeadDataList>
        <!--ro, opt, array, 表头数据列表, subType:object-->
        <HeadData>
            <!--ro, opt, object, 表头数据-->
            <id>
                <!--ro, req, int, 索引-->1
            </id>
            <data>
                <!--ro, req, string, 数据-->test
            </data>
        </HeadData>
    </HeadDataList>
    <ItemStyleList>
        <!--ro, opt, array, 表格列或行样式列表, subType:object-->
        <ItemStyle>
            <!--ro, opt, object, 表格列或行样式-->
            <id>
                <!--ro, req, int, 索引-->1
            </id>
            <width>
                <!--ro, req, int, 每列宽度所占百分比-->1
            </width>
            <fontSize>

```

```

</fontSize>
<FontColor>
  <!--ro, req, object, 字体颜色-->
  <RGB>
    <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
  </RGB>
</FontColor>
<BackColor>
  <!--ro, req, object, 背景色-->
  <RGB>
    <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
  </RGB>
</BackColor>
</ItemStyle>
</ItemStyleList>
</Call>
<DynamicPic>
  <!--ro, opt, object, 动态弹图窗口配置-->
  <materialNo>
    <!--ro, req, int, 素材编号-->1
  </materialNo>
  <backPicId>
    <!--ro, opt, int, 背景图片素材ID-->1
  </backPicId>
</DynamicPic>
<CapturePic>
  <!--ro, opt, object, 抓拍图片参数-->
  <backPicId>
    <!--ro, opt, int, 背景图片素材ID-->1
  </backPicId>
  <ipcMaterialNo>
    <!--ro, req, int, IPC素材编号-->1
  </ipcMaterialNo>
  <cancelType>
    <!--ro, req, enum, 取消方式, subType:string, [auto#自动,manual#手动]-->manual
  </cancelType>
  <duration>
    <!--ro, opt, int, 持续时间, unit:s, unitType:时间-->1
  </duration>
</CapturePic>
<ClockParam>
  <!--ro, opt, object, 时钟参数-->
  <backPicId>
    <!--ro, opt, int, 背景图片素材ID-->1
  </backPicId>
  <ClockIcon>
    <!--ro, req, object, 时钟图标参数-->
    <enabled>
      <!--ro, req, bool, 使能-->true
    </enabled>
    <type>
      <!--ro, opt, enum, 时钟类型, subType:string, [cLock1#cLock1,cLock2#cLock2]-->cLock1
    </type>
  </ClockIcon>
  <Position>
    <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--ro, req, int, x坐标-->1
    </positionX>
    <positionY>
      <!--ro, req, int, y坐标-->1
    </positionY>
    <height>
      <!--ro, req, int, 宽-->1
    </height>
    <width>
      <!--ro, req, int, 高-->1
    </width>
  </Position>
</ClockIcon>
<YmdParam>
  <!--ro, req, object, 时钟年月日参数-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <fontSize>
    <!--ro, req, int, 字体大小-->1
  </fontSize>
  <FontColor>
    <!--ro, req, object, 字体颜色-->
    <RGB>
      <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
    </RGB>
  </FontColor>
  <BackColor>
    <!--ro, req, object, 背景色-->
    <RGB>
      <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
    </RGB>
  </BackColor>
  <Position>
    <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>

```

```
<!--ro, req, int, x坐标-->1
</positionX>
<positionY>
<!--ro, req, int, y坐标-->1
</positionY>
<height>
<!--ro, req, int, 宽-->1
</height>
<width>
<!--ro, req, int, 高-->1
</width>
</Position>
</YmdParam>
<HmsParam>
<!--ro, req, object, 时钟时分秒参数-->
<enabled>
<!--ro, req, bool, 使能-->true
</enabled>
<fontSize>
<!--ro, req, int, 字体大小-->1
</fontSize>
<FontColor>
<!--ro, req, object, 字体颜色-->
<RGB>
<!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
</RGB>
</FontColor>
<BackColor>
<!--ro, req, object, 背景色-->
<RGB>
<!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
</RGB>
</BackColor>
<Position>
<!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
<positionX>
<!--ro, req, int, x坐标-->1
</positionX>
<positionY>
<!--ro, req, int, y坐标-->1
</positionY>
<height>
<!--ro, req, int, 宽-->1
</height>
<width>
<!--ro, req, int, 高-->1
</width>
</Position>
</HmsParam>
<WeekParam>
<!--ro, req, object, 时钟星期参数-->
<enabled>
<!--ro, req, bool, 使能-->true
</enabled>
<fontSize>
<!--ro, req, int, 字体大小-->1
</fontSize>
<FontColor>
<!--ro, req, object, 字体颜色-->
<RGB>
<!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
</RGB>
</FontColor>
<BackColor>
<!--ro, req, object, 背景色-->
<RGB>
<!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
</RGB>
</BackColor>
<Position>
<!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
<positionX>
<!--ro, req, int, x坐标-->1
</positionX>
<positionY>
<!--ro, req, int, y坐标-->1
</positionY>
<height>
<!--ro, req, int, 宽-->1
</height>
<width>
<!--ro, req, int, 高-->1
</width>
</Position>
</WeekParam>
</ClockParam>
<WeatherParam>
<!--ro, opt, object, 天气参数-->
<backPicId>
<!--ro, opt, int, 天气背景图片素材id-->1
</backPicId>
<WeatherIcon>
<!--ro, opt, object, 天气图标参数-->
<enabled>
```

```
</enable>
<!--ro, req, bool, 使能-->true
</enable>
<Position>
<!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
<positionX>
<!--ro, req, int, x坐标-->1
</positionX>
<positionY>
<!--ro, req, int, y坐标-->1
</positionY>
<height>
<!--ro, req, int, 宽-->1
</height>
<width>
<!--ro, req, int, 高-->1
</width>
</Position>
</WeatherIcon>
<Date>
<!--ro, opt, object, 天气日期参数-->
<enable>
<!--ro, req, bool, 使能-->true
</enable>
<fontSize>
<!--ro, req, int, 字体大小-->1
</fontSize>
<FontColor>
<!--ro, req, object, 字体颜色-->
<RGB>
<!--ro, opt, int, RGB值, desc:十进制数据表示三原色, 如16777215表示0xFFFFFF-->1
</RGB>
</FontColor>
<BackColor>
<!--ro, req, object, 背景色-->
<RGB>
<!--ro, opt, int, RGB值, desc:十进制数据表示三原色, 如16777215表示0xFFFFFF-->1
</RGB>
</BackColor>
<Position>
<!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
<positionX>
<!--ro, req, int, x坐标-->1
</positionX>
<positionY>
<!--ro, req, int, y坐标-->1
</positionY>
<height>
<!--ro, req, int, 宽-->1
</height>
<width>
<!--ro, req, int, 高-->1
</width>
</Position>
</Date>
<Temperature>
<!--ro, opt, object, 天气温度参数-->
<enable>
<!--ro, req, bool, 使能-->true
</enable>
<fontSize>
<!--ro, req, int, 字体大小-->1
</fontSize>
<FontColor>
<!--ro, req, object, 字体颜色-->
<RGB>
<!--ro, opt, int, RGB值, desc:十进制数据表示三原色, 如16777215表示0xFFFFFF-->1
</RGB>
</FontColor>
<BackColor>
<!--ro, req, object, 背景色-->
<RGB>
<!--ro, opt, int, RGB值, desc:十进制数据表示三原色, 如16777215表示0xFFFFFF-->1
</RGB>
</BackColor>
<Position>
<!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
<positionX>
<!--ro, req, int, x坐标-->1
</positionX>
<positionY>
<!--ro, req, int, y坐标-->1
</positionY>
<height>
<!--ro, req, int, 宽-->1
</height>
<width>
<!--ro, req, int, 高-->1
</width>
</Position>
</Temperature>
<WeatherContent>
<!--ro, opt, object, 天气情况参数-->
<enable>
```

```
<!--ro, req, bool, 使能-->true
</enabled>
<fontSize>
  <!--ro, req, int, 字体大小-->1
</fontSize>
<FontColor>
  <!--ro, req, object, 字体颜色-->
  <RGB>
    <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
  </RGB>
</FontColor>
<BackColor>
  <!--ro, req, object, 背景色-->
  <RGB>
    <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
  </RGB>
</BackColor>
<Position>
  <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
  <positionX>
    <!--ro, req, int, x坐标-->1
  </positionX>
  <positionY>
    <!--ro, req, int, y坐标-->1
  </positionY>
  <height>
    <!--ro, req, int, 宽-->1
  </height>
  <width>
    <!--ro, req, int, 高-->1
  </width>
</Position>
<WeatherContent>
<City>
  <!--ro, opt, object, 天气城市参数-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <cityId>
    <!--ro, req, string, 城市编号-->test
  </cityId>
  <cityName>
    <!--ro, req, string, 城市名称-->test
  </cityName>
  <fontSize>
    <!--ro, req, int, 字体大小-->1
  </fontSize>
  <FontColor>
    <!--ro, req, object, 字体颜色-->
    <RGB>
      <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
    </RGB>
  </FontColor>
  <BackColor>
    <!--ro, req, object, 背景色-->
    <RGB>
      <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
    </RGB>
  </BackColor>
  <Position>
    <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--ro, req, int, x坐标-->1
    </positionX>
    <positionY>
      <!--ro, req, int, y坐标-->1
    </positionY>
    <height>
      <!--ro, req, int, 宽-->1
    </height>
    <width>
      <!--ro, req, int, 高-->1
    </width>
  </Position>
</City>
<Humidity>
  <!--ro, opt, object, 天气湿度参数-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <fontSize>
    <!--ro, req, int, 字体大小-->1
  </fontSize>
  <FontColor>
    <!--ro, req, object, 字体颜色-->
    <RGB>
      <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
    </RGB>
  </FontColor>
  <BackColor>
    <!--ro, req, object, 背景色-->
    <RGB>
      <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
    </RGB>
  </BackColor>
```

```
</Kuts>
</BackColor>
<Position>
  <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
  <positionX>
    <!--ro, req, int, x坐标-->1
  </positionX>
  <positionY>
    <!--ro, req, int, y坐标-->1
  </positionY>
  <height>
    <!--ro, req, int, 宽-->1
  </height>
  <width>
    <!--ro, req, int, 高-->1
  </width>
</Position>
</Humidity>
<AirQuality>
  <!--ro, opt, object, 天气空气质量参数-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <fontSize>
    <!--ro, req, int, 字体大小-->1
  </fontSize>
  <fontColor>
    <!--ro, req, object, 字体颜色-->
    <RGB>
      <!--ro, opt, int, RGB值, desc:十进制数据表示三原色, 如16777215表示0xFFFFFF-->1
    </RGB>
  </FontColor>
  <BackColor>
    <!--ro, req, object, 背景色-->
    <RGB>
      <!--ro, opt, int, RGB值, desc:十进制数据表示三原色, 如16777215表示0xFFFFFF-->1
    </RGB>
  </BackColor>
  <Position>
    <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--ro, req, int, x坐标-->1
    </positionX>
    <positionY>
      <!--ro, req, int, y坐标-->1
    </positionY>
    <height>
      <!--ro, req, int, 宽-->1
    </height>
    <width>
      <!--ro, req, int, 高-->1
    </width>
  </Position>
</AirQuality>
<UpdateTime>
  <!--ro, opt, object, 天气更新时间参数-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <refreshTime>
    <!--ro, req, string, 刷新时间, desc:设备早期返回时间格式有误, 错为: 20200826T051500+08, 为保证XSD校验通过, 改为string类型-->20200826T051500+08
  </refreshTime>
  <updateInterval>
    <!--ro, req, int, 刷新间隔, unit:minute-->1
  </updateInterval>
  <fontSize>
    <!--ro, req, int, 字体大小-->1
  </fontSize>
  <fontColor>
    <!--ro, req, object, 字体颜色-->
    <RGB>
      <!--ro, opt, int, RGB值, desc:十进制数据表示三原色, 如16777215表示0xFFFFFF-->1
    </RGB>
  </FontColor>
  <BackColor>
    <!--ro, req, object, 背景色-->
    <RGB>
      <!--ro, opt, int, RGB值, desc:十进制数据表示三原色, 如16777215表示0xFFFFFF-->1
    </RGB>
  </BackColor>
  <Position>
    <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--ro, req, int, x坐标-->1
    </positionX>
    <positionY>
      <!--ro, req, int, y坐标-->1
    </positionY>
    <height>
      <!--ro, req, int, 宽-->1
    </height>
    <width>
      <!--ro, req, int, 高-->1
    </width>
```

```
</width>
</Position>
</UpdateTime>
<Wind>
  <!--ro, opt, object, 天气风力参数-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <fontSize>
    <!--ro, req, int, 字体大小-->1
  </fontSize>
  <FontColor>
    <!--ro, req, object, 字体颜色-->
    <RGB>
      <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
    </RGB>
  </FontColor>
  <BackColor>
    <!--ro, req, object, 背景色-->
    <RGB>
      <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
    </RGB>
  </BackColor>
  <Position>
    <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--ro, req, int, x坐标-->1
    </positionX>
    <positionY>
      <!--ro, req, int, y坐标-->1
    </positionY>
    <height>
      <!--ro, req, int, 宽-->1
    </height>
    <width>
      <!--ro, req, int, 高-->1
    </width>
  </Position>
</Wind>
</WeatherParam>
<Countdown>
  <!--ro, opt, object, 倒计时素材-->
  <endTime>
    <!--ro, req, string, 倒计时时间点, desc:设备早期返回时间格式有误, 错为: 20200826T051500+08, 为保证XSD校验通过, 改为string类型-->20200826T051500+08
  </endTime>
  <template>
    <!--ro, req, enum, 模板类型, subType:string, [template1#模板1,template2#模板2,openingActivity1#活动开幕1,openingActivity2#活动开幕2,commercialActivity1#商业活动1,commercialActivity2#商业活动2,colLegeEntranceExam#高考,overseasTimeFormat#海外时间格式]-->template1
  </template>
  <timeUnit>
    <!--ro, req, enum, 时间单位, subType:string, [year#年,month#月,day#日,week#周,hour#时,minute#分,second#秒]-->year
  </timeUnit>
  <backPicId>
    <!--ro, opt, int, 背景图片素材ID-->1
  </backPicId>
  <TimeFontCfg>
    <!--ro, opt, object, 时间字体参数-->
    <fontSize>
      <!--ro, req, int, 字体大小-->1
    </fontSize>
    <FontColor>
      <!--ro, req, object, 字体颜色-->
      <RGB>
        <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
      </RGB>
    </FontColor>
  </TimeFontCfg>
  <Position>
    <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--ro, req, int, x坐标-->1
    </positionX>
    <positionY>
      <!--ro, req, int, y坐标-->1
    </positionY>
    <height>
      <!--ro, req, int, 宽-->1
    </height>
    <width>
      <!--ro, req, int, 高-->1
    </width>
  </Position>
  <TimeFontCfg>
    <timeUnitEnabled>
      <!--ro, opt, bool, 时间单位是否有效-->true
    </timeUnitEnabled>
    <maxTimeUnit>
      <!--ro, req, enum, 最大时间单位, subType:string, [year#年,month#月,day#日,week#周,hour#时,minute#分,second#秒,week#周], desc:表示当前倒计时时的最大显示单元,例如:maxTimeUnit为月,则1年3个月,对应月份值为15-->year
    </maxTimeUnit>
  </TimeFontCfg>
  <ModuleInfoList>
    <!--ro, opt, array, 界面模块信息列表, subType:object-->
    <ModuleInfo>
```

```
<!--ro, opt, object, 界面模块信息-->
<moduleType>
  <!--ro, req, enum, 模块类型, subType:string, [year#年,month#月,day#日,hour#时,minute#分,second#秒,week#周,text#文字], desc:text为时间旁边
  的文字,例如年下面的文字"年"-->second
  </moduleType>
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <fontSize>
    <!--ro, req, int, 字体大小-->1
  </fontSize>
  <fontColor>
    <!--ro, req, object, 字体颜色-->
    <RGB>
      <!--ro, opt, int, 十进制数据表示三原色-->1
    </RGB>
  </fontColor>
  <text>
    <!--ro, opt, string, 文字, desc:当moduleType为text时有效-->test
  </text>
  <Position>
    <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--ro, req, int, X坐标-->1
    </positionX>
    <positionY>
      <!--ro, req, int, Y坐标-->1
    </positionY>
    <height>
      <!--ro, req, int, 宽-->1
    </height>
    <width>
      <!--ro, req, int, 高-->1
    </width>
  </Position>
</ModuleInfo>
</ModuleInfoList>
<Countdown>
<localInputNo>
  <!--ro, req, enum, 本地输入类型, subType:string, [VGA#VGA,HDMI#HDMI]-->VGA
</localInputNo>
<HyperlinkBtn>
  <!--ro, opt, object, 超链接按钮-->
  <backPicId>
    <!--ro, opt, int, 背景图片素材ID-->1
  </backPicId>
</HyperlinkBtn>
<CharactersAttribute>
  <!--ro, opt, object, 文字属性, desc:当dynamicType为character时有效-->
  <fontSize>
    <!--ro, req, int, 字体大小-->1
  </fontSize>
  <fontColor>
    <!--ro, req, object, 字体颜色-->
    <RGB>
      <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
    </RGB>
  </fontColor>
  <fontType>
    <!--ro, opt, enum, 字体类型, subType:string, [normal#常规,bold#粗体]-->normal
  </fontType>
  <backgroundType>
    <!--ro, req, enum, 背景类型, subType:string, [color#颜色,picture#图片]-->color
  </backgroundType>
  <BackColor>
    <!--ro, req, object, 背景色-->
    <RGB>
      <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
    </RGB>
  <fillEnabled>
    <!--ro, opt, bool, 背景色填充使能, desc:默认为true, 表示背景色填充-->true
  </fillEnabled>
</BackColor>
<backTransparent>
  <!--ro, opt, int, 背景透明度-->1
</backTransparent>
<alignType>
  <!--ro, req, enum, 对齐方式, subType:string, [Left#左,right#右,middle#中]-->middle
</alignType>
<verticalAlignType>
  <!--ro, req, enum, 垂直对齐方式, subType:string, [top#顶,bottom#底,verticalCenter#中]-->top
</verticalAlignType>
<characterContent>
  <!--ro, opt, string, 文字内容, range:[0,512]-->test
</characterContent>
<backPicId>
  <!--ro, opt, int, 背景图片素材ID-->1
</backPicId>
<characterStyle>
  <!--ro, req, enum, 字体效果, subType:string, [normal#常规,bold#粗体]-->normal
</characterStyle>
<characterType>
  <!--ro, req, enum, 字体类型, subType:string, [mainTitle#主标题,subTitle#副标题]-->mainTitle
</characterType>
```

```
</CharactersAttribute>
<SignIn>
  <!--ro, opt, object, 人员签到展示窗口, desc:当staticMaterialType为signIn时有效-->
  <backPicId>
    <!--ro, opt, int, 背景图片素材ID-->1
  </backPicId>
  <ShowInfo>
    <!--ro, opt, object, 展示信息-->
    <row>
      <!--ro, opt, int, 窗口内人员签到展示行数-->1
    </row>
    <column>
      <!--ro, opt, int, 窗口内人员签到展示列数-->1
    </column>
    <Position>
      <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
      <positionX>
        <!--ro, req, int, x坐标-->1
      </positionX>
      <positionY>
        <!--ro, req, int, y坐标-->1
      </positionY>
      <height>
        <!--ro, req, int, 宽-->1
      </height>
      <width>
        <!--ro, req, int, 高-->1
      </width>
    </Position>
    <inLib>
      <!--ro, opt, object, 在库信息-->
      <nameEnabled>
        <!--ro, opt, bool, 在库人员是否展示姓名-->true
      </nameEnabled>
      <genderEnabled>
        <!--ro, opt, bool, 在库人员是否展示性别-->true
      </genderEnabled>
      <selfDefine1Enabled>
        <!--ro, opt, bool, 在库人员是否展示自定义字段1-->true
      </selfDefine1Enabled>
      <selfDefine2Enabled>
        <!--ro, opt, bool, 在库人员是否展示自定义字段2-->true
      </selfDefine2Enabled>
      <signInTimeEnabled>
        <!--ro, opt, bool, 在库人员是否展示签到时间-->true
      </signInTimeEnabled>
      <temperatureEnabled>
        <!--ro, opt, bool, 在库人员是否展示温度-->true
      </temperatureEnabled>
    </inLib>
    <outLib>
      <!--ro, opt, object, 不在库信息-->
      <nameEnabled>
        <!--ro, opt, bool, 未知人员是否展示姓名-->true
      </nameEnabled>
      <name>
        <!--ro, opt, string, 未知人员姓名-->test
      </name>
      <genderEnabled>
        <!--ro, opt, bool, 未知人员是否展示性别-->true
      </genderEnabled>
      <signInTimeEnabled>
        <!--ro, opt, bool, 未知人员是否展示签到时间-->true
      </signInTimeEnabled>
      <temperatureEnabled>
        <!--ro, opt, bool, 未知人员是否展示温度-->true
      </temperatureEnabled>
    </outLib>
    <showDuration>
      <!--ro, req, enum, 签到记录显示时间, subType:string, [always#不消失,10s#10s,20s#20s,30s#30s,no#不显示]-->always
    </showDuration>
  </ShowInfo>
</MainTitle>
  <!--ro, opt, object, 主标题-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <fontSize>
    <!--ro, req, int, 字体大小-->1
  </fontSize>
  <fontColor>
    <!--ro, req, object, 字体颜色-->
    <RGB>
      <!--ro, opt, int, 十进制数据表示三原色-->1
    </RGB>
  </FontColor>
</Position>
```

```
<!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
<positionX>
  <!--ro, req, int, x坐标-->1
</positionX>
<positionY>
  <!--ro, req, int, y坐标-->1
</positionY>
<height>
  <!--ro, req, int, 宽-->1
</height>
<width>
  <!--ro, req, int, 高-->1
</width>
</Position>
<text>
  <!--ro, opt, string, 标题文字-->test
</text>
</MainTitle>
<SubTitle>
  <!--ro, opt, object, 副标题-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <fontSize>
    <!--ro, req, int, 字体大小-->1
  </fontSize>
  <FontColor>
    <!--ro, req, object, 字体颜色-->
    <RGB>
      <!--ro, opt, int, 十进制数据表示三原色-->1
    </RGB>
    </FontColor>
  <Position>
    <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--ro, req, int, x坐标-->1
    </positionX>
    <positionY>
      <!--ro, req, int, y坐标-->1
    </positionY>
    <height>
      <!--ro, req, int, 宽-->1
    </height>
    <width>
      <!--ro, req, int, 高-->1
    </width>
  </Position>
  <text>
    <!--ro, opt, string, 标题文字-->test
  </text>
</SubTitle>
<NormalTemperature>
  <!--ro, opt, object, 正常体温信息-->
  <PeopleNumber>
    <!--ro, opt, object, 正常体温人数信息-->
    <enabled>
      <!--ro, req, bool, 使能-->true
    </enabled>
    <fontSize>
      <!--ro, req, int, 字体大小-->1
    </fontSize>
    <FontColor>
      <!--ro, req, object, 字体颜色-->
      <RGB>
        <!--ro, opt, int, 十进制数据表示三原色-->1
      </RGB>
      </FontColor>
    <Position>
      <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
      <positionX>
        <!--ro, req, int, X坐标-->1
      </positionX>
      <positionY>
        <!--ro, req, int, Y坐标-->1
      </positionY>
      <height>
        <!--ro, req, int, 宽-->1
      </height>
      <width>
        <!--ro, req, int, 高-->1
      </width>
    </Position>
    <text>
      <!--ro, opt, string, 标题文字-->test
    </text>
  </PeopleNumber>
  <Icon>
    <!--ro, opt, object, 正常体温图标-->
    <enabled>
      <!--ro, req, bool, 使能-->true
    </enabled>
    <Position>
      <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
      <positionX>
```

```
<!--ro, req, int, X坐标-->1
</positionX>
<positionY>
<!--ro, req, int, Y坐标-->1
</positionY>
<height>
<!--ro, req, int, 宽-->1
</height>
<width>
<!--ro, req, int, 高-->1
</width>
</Position>
<materialNo>
<!--ro, opt, int, 素材ID-->1
</materialNo>
</Icon>
</NormalTemperature>
<AbnormalTemperature>
<!--ro, opt, object, 异常体温信息-->
<PeopleNumber>
<!--ro, opt, object, 异常体温人数信息-->
<enabled>
<!--ro, req, bool, 使能-->true
</enabled>
<fontSize>
<!--ro, req, int, 字体大小-->1
</fontSize>
<fontColor>
<!--ro, req, object, 字体颜色-->
<RGB>
<!--ro, opt, int, 十进制数据表示三原色-->1
</RGB>
</FontColor>
<Position>
<!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
<positionX>
<!--ro, req, int, X坐标-->1
</positionX>
<positionY>
<!--ro, req, int, Y坐标-->1
</positionY>
<height>
<!--ro, req, int, 宽-->1
</height>
<width>
<!--ro, req, int, 高-->1
</width>
</Position>
<text>
<!--ro, opt, string, 标题文字-->test
</text>
</PeopleNumber>
<Icon>
<!--ro, opt, object, 异常体温图标-->
<enabled>
<!--ro, req, bool, 使能-->true
</enabled>
<Position>
<!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
<positionX>
<!--ro, req, int, X坐标-->1
</positionX>
<positionY>
<!--ro, req, int, Y坐标-->1
</positionY>
<height>
<!--ro, req, int, 宽-->1
</height>
<width>
<!--ro, req, int, 高-->1
</width>
</Position>
<materialNo>
<!--ro, opt, int, 素材ID-->1
</materialNo>
</Icon>
</AbnormalTemperature>
</SignIn>
<EventParam>
<!--ro, opt, object, 事件参数-->
<EventResourceList>
<!--ro, opt, array, 事件源信息列表, subType:object-->
<EventResource>
<!--ro, opt, object, 事件源信息-->
<ipVersion>
<!--ro, req, enum, ip地址版本, subType:string, [v4#v4, v6#v6, dual#双]-->v4
</ipVersion>
<ipAddress>
<!--ro, opt, string, ipv4地址-->test
</ipAddress>
<ipv6Address>
<!--ro, opt, string, ipv6地址-->test
</ipv6Address>
```

```

<portNo>
  <!--ro, req, int, 端口号-->1
</portNo>
<userName>
  <!--ro, req, string, 用户名-->test
</userName>
<password>
  <!--ro, req, string, 密码-->test
</password>
<channels>
  <!--ro, opt, string, 通道号, desc:多选以逗号隔开-->1,2
</channels>
</EventResource>
</EventResourceList>
<backPicId>
  <!--ro, opt, int, 背景图片素材ID-->1
</backPicId>
<Title>
  <!--ro, opt, object, 标题-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <fontSize>
    <!--ro, req, int, 字体大小-->1
  </fontSize>
  <fontColor>
    <!--ro, req, object, 字体颜色-->
    <RGB>
      <!--ro, opt, int, 十进制数据表示三原色-->1
    </RGB>
    </FontColor>
  <Position>
    <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--ro, req, int, X坐标-->1
    </positionX>
    <positionY>
      <!--ro, req, int, Y坐标-->1
    </positionY>
    <height>
      <!--ro, req, int, 宽-->1
    </height>
    <width>
      <!--ro, req, int, 高-->1
    </width>
  </Position>
  <text>
    <!--ro, opt, string, 标题文字-->test
  </text>
</Title>
<PeopleCounting>
  <!--ro, opt, object, 人数统计提醒-->
  <maxPeopleNumber>
    <!--ro, opt, int, 最大可进入人数-->1
  </maxPeopleNumber>
  <unavailableBackPicId>
    <!--ro, opt, int, 人员不能进入时控件背景图片id-->1
  </unavailableBackPicId>
  <supportChangeByPeople>
    <!--ro, req, enum, 支持根据人数修改样式的字段, subType:string,
[prompt#prompt,peopleIcon#peopleIcon,availablePeopleNumber#availablePeopleNumber]-->prompt
  </supportChangeByPeople>
  <Prompt>
    <!--ro, opt, object, 提示信息-->
    <enabled>
      <!--ro, req, bool, 使能-->true
    </enabled>
    <fontSize>
      <!--ro, req, int, 字体大小-->1
    </fontSize>
    <Position>
      <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
      <positionX>
        <!--ro, req, int, X坐标-->1
      </positionX>
      <positionY>
        <!--ro, req, int, Y坐标-->1
      </positionY>
      <height>
        <!--ro, req, int, 宽-->1
      </height>
      <width>
        <!--ro, req, int, 高-->1
      </width>
    </Position>
    <text>
      <!--ro, opt, string, 标题文字-->test
    </text>
    <fontColor>
      <!--ro, opt, object, 人员可以进入时的字体颜色-->
      <RGB>
        <!--ro, opt, int, 十进制数据表示三原色-->1
      </RGB>
    </FontColor>
  </Prompt>

```

```
</FontColor>
<unavailableText>
  <!--ro, opt, string, 人员不能进入时的标题文字-->test
</unavailableText>
<UnavailableFontColor>
  <!--ro, opt, object, 人员不能进入时的字体颜色-->
  <RGB>
    <!--ro, opt, int, 十进制数据表示三原色-->1
  </RGB>
</UnavailableFontColor>
</Prompt>
<PeopleIcon>
  <!--ro, opt, object, 行人通行图标-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <Position>
    <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--ro, req, int, X坐标-->1
    </positionX>
    <positionY>
      <!--ro, req, int, Y坐标-->1
    </positionY>
    <height>
      <!--ro, req, int, 宽-->1
    </height>
    <width>
      <!--ro, req, int, 高-->1
    </width>
  </Position>
  <materialNo>
    <!--ro, opt, int, 素材ID-->1
  </materialNo>
</PeopleIcon>
<InsidePeopleTitle>
  <!--ro, opt, object, 已经进入人数标题-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <fontSize>
    <!--ro, req, int, 字体大小-->1
  </fontSize>
  <FontColor>
    <!--ro, req, object, 字体颜色-->
    <RGB>
      <!--ro, opt, int, 十进制数据表示三原色-->1
    </RGB>
  </FontColor>
  <Position>
    <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--ro, req, int, X坐标-->1
    </positionX>
    <positionY>
      <!--ro, req, int, Y坐标-->1
    </positionY>
    <height>
      <!--ro, req, int, 宽-->1
    </height>
    <width>
      <!--ro, req, int, 高-->1
    </width>
  </Position>
  <text>
    <!--ro, opt, string, 标题文字-->test
  </text>
</InsidePeopleTitle>
<InsidePeopleNumber>
  <!--ro, opt, object, 已经进入人数-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <fontSize>
    <!--ro, req, int, 字体大小-->1
  </fontSize>
  <FontColor>
    <!--ro, req, object, 字体颜色-->
    <RGB>
      <!--ro, opt, int, 十进制数据表示三原色-->1
    </RGB>
  </FontColor>
  <Position>
    <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--ro, req, int, X坐标-->1
    </positionX>
    <positionY>
      <!--ro, req, int, Y坐标-->1
    </positionY>
    <height>
      <!--ro, req, int, 宽-->1
    </height>
    <width>
```

```
<!--ro, req, int, 高-->1
</width>
</Position>
</InsidePeopleNumber>
<AvailablePeopleTitle>
<!--ro, opt, object, 还可以进入人数标题-->
<enabled>
<!--ro, req, bool, 使能-->true
</enabled>
<fontSize>
<!--ro, req, int, 字体大小-->1
</fontSize>
<FontColor>
<!--ro, req, object, 字体颜色-->
<RGB>
<!--ro, opt, int, 十进制数据表示三原色-->1
</RGB>
</FontColor>
<Position>
<!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
<positionX>
<!--ro, req, int, X坐标-->1
</positionX>
<positionY>
<!--ro, req, int, Y坐标-->1
</positionY>
<height>
<!--ro, req, int, 宽-->1
</height>
<width>
<!--ro, req, int, 高-->1
</width>
</Position>
<text>
<!--ro, opt, string, 标题文字-->test
</text>
</AvailablePeopleTitle>
<AvailablePeopleNumber>
<!--ro, opt, object, 还可以进入人数-->
<enabled>
<!--ro, req, bool, 使能-->true
</enabled>
<fontSize>
<!--ro, req, int, 字体大小-->1
</fontSize>
<Position>
<!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
<positionX>
<!--ro, req, int, X坐标-->1
</positionX>
<positionY>
<!--ro, req, int, Y坐标-->1
</positionY>
<height>
<!--ro, req, int, 宽-->1
</height>
<width>
<!--ro, req, int, 高-->1
</width>
</Position>
<FontColor>
<!--ro, opt, object, 人员可以进入时的字体颜色-->
<RGB>
<!--ro, opt, int, 十进制数据表示三原色-->1
</RGB>
</FontColor>
<UnavailableFontColor>
<!--ro, opt, object, 人员不能进入时的字体颜色-->
<RGB>
<!--ro, opt, int, 十进制数据表示三原色-->1
</RGB>
</UnavailableFontColor>
</AvailablePeopleNumber>
<UnavailablePeopleIcon>
<!--ro, opt, object, 人员不能进入时的行人图标-->
<enabled>
<!--ro, req, bool, 使能-->true
</enabled>
<Position>
<!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
<positionX>
<!--ro, req, int, X坐标-->1
</positionX>
<positionY>
<!--ro, req, int, Y坐标-->1
</positionY>
<height>
<!--ro, req, int, 宽-->1
</height>
<width>
<!--ro, req, int, 高-->1
</width>
</Position>
```

```
<materialNo>
  <!--ro, opt, int, 素材ID-->1
</materialNo>
</UnavailablePeopleIcon>
<AudioAlarm>
  <!--ro, opt, object, 音频报警信息-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <materialNo>
    <!--ro, opt, int, 音频素材ID-->1
  </materialNo>
</AudioAlarm>
</PeopleCounting>
<Mask>
  <!--ro, opt, object, 未戴口罩提醒-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <fontSize>
    <!--ro, req, int, 字体大小-->1
  </fontSize>
  <fontColor>
    <!--ro, req, object, 字体颜色-->
    <RGB>
      <!--ro, opt, int, 十进制数据表示三原色-->1
    </RGB>
  </fontColor>
  <BackColor>
    <!--ro, req, object, 背景颜色-->
    <RGB>
      <!--ro, opt, int, 十进制数据表示三原色-->1
    </RGB>
  </BackColor>
  <Position>
    <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--ro, req, int, X坐标-->1
    </positionX>
    <positionY>
      <!--ro, req, int, Y坐标-->1
    </positionY>
    <height>
      <!--ro, req, int, 宽-->1
    </height>
    <width>
      <!--ro, req, int, 高-->1
    </width>
  </Position>
  <text>
    <!--ro, opt, string, 标题文字-->test
  </text>
  <MaskIcon>
    <!--ro, opt, object, 未戴口罩图标-->
    <enabled>
      <!--ro, req, bool, 使能-->true
    </enabled>
    <Position>
      <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
      <positionX>
        <!--ro, req, int, X坐标-->1
      </positionX>
      <positionY>
        <!--ro, req, int, Y坐标-->1
      </positionY>
      <height>
        <!--ro, req, int, 宽-->1
      </height>
      <width>
        <!--ro, req, int, 高-->1
      </width>
    </Position>
    <materialNo>
      <!--ro, opt, int, 素材ID-->1
    </materialNo>
  </MaskIcon>
  <AudioAlarm>
    <!--ro, opt, object, 音频报警信息-->
    <enabled>
      <!--ro, req, bool, 使能-->true
    </enabled>
    <materialNo>
      <!--ro, opt, int, 音频素材ID-->1
    </materialNo>
  </AudioAlarm>
</Mask>
<Temperature>
  <!--ro, opt, object, 温度异常提醒-->
  <Prompt>
    <!--ro, opt, object, 提示信息-->
    <enabled>
      <!--ro, req, bool, 使能-->true
    </enabled>
    <fontSize>
```

```
<!--ro, req, int, 字体大小-->1
</fontSize>
<FontColor>
<!--ro, req, object, 字体颜色-->
<RGB>
<!--ro, opt, int, 十进制数据表示三原色-->1
</RGB>
</FontColor>
<BackColor>
<!--ro, req, object, 背景颜色-->
<RGB>
<!--ro, opt, int, 十进制数据表示三原色-->1
</RGB>
</BackColor>
<Position>
<!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
<positionX>
<!--ro, req, int, X坐标-->1
</positionX>
<positionY>
<!--ro, req, int, Y坐标-->1
</positionY>
<height>
<!--ro, req, int, 宽-->1
</height>
<width>
<!--ro, req, int, 高-->1
</width>
</Position>
<text>
<!--ro, opt, string, 标题文字-->test
</text>
</Prompt>
<TemperatureIcon>
<!--ro, opt, object, 温度异常图标-->
<enabled>
<!--ro, req, bool, 使能-->true
</enabled>
<Position>
<!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
<positionX>
<!--ro, req, int, X坐标-->1
</positionX>
<positionY>
<!--ro, req, int, Y坐标-->1
</positionY>
<height>
<!--ro, req, int, 宽-->1
</height>
<width>
<!--ro, req, int, 高-->1
</width>
</Position>
<materialNo>
<!--ro, opt, int, 素材ID-->1
</materialNo>
</TemperatureIcon>
<AudioAlarm>
<!--ro, opt, object, 音频报警信息-->
<enabled>
<!--ro, req, bool, 使能-->true
</enabled>
<materialNo>
<!--ro, opt, int, 音频素材ID-->1
</materialNo>
</AudioAlarm>
<abnormalHighTemperature>
<!--ro, opt, float, 异常高温值, unit:℃, unitType:温度, desc:保留一位小数-->37.5
</abnormalHighTemperature>
</Temperature>
<WearMask>
<!--ro, opt, object, 已戴口罩提醒-->
<enabled>
<!--ro, req, bool, 使能-->true
</enabled>
<fontSize>
<!--ro, req, int, 字体大小-->1
</fontSize>
<FontColor>
<!--ro, req, object, 字体颜色-->
<RGB>
<!--ro, opt, int, 十进制数据表示三原色-->1
</RGB>
</FontColor>
<BackColor>
<!--ro, req, object, 背景颜色-->
<RGB>
<!--ro, opt, int, 十进制数据表示三原色-->1
</RGB>
</BackColor>
<Position>
<!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
<positionX>
```

```
<!--ro, req, int, X坐标-->1
</positionX>
<positionY>
<!--ro, req, int, Y坐标-->1
</positionY>
<height>
<!--ro, req, int, 宽-->1
</height>
<width>
<!--ro, req, int, 高-->1
</width>
</Position>
<text>
<!--ro, opt, string, 标题文字-->test
</text>
<MaskIcon>
<!--ro, opt, object, 已戴口罩图标-->
<enabled>
<!--ro, req, bool, 使能-->true
</enabled>
<Position>
<!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
<positionX>
<!--ro, req, int, X坐标-->1
</positionX>
<positionY>
<!--ro, req, int, Y坐标-->1
</positionY>
<height>
<!--ro, req, int, 宽-->1
</height>
<width>
<!--ro, req, int, 高-->1
</width>
</Position>
<materialNo>
<!--ro, opt, int, 素材ID-->1
</materialNo>
</MaskIcon>
</WearMask>
<NormalTemperature>
<!--ro, opt, object, 温度正常提醒-->
<Prompt>
<!--ro, opt, object, 提示信息-->
<enabled>
<!--ro, req, bool, 使能-->true
</enabled>
<fontSize>
<!--ro, req, int, 字体大小-->1
</fontSize>
<fontColor>
<!--ro, req, object, 字体颜色-->
<RGB>
<!--ro, opt, int, 十进制数据表示三原色-->1
</RGB>
</FontColor>
<BackColor>
<!--ro, req, object, 背景颜色-->
<RGB>
<!--ro, opt, int, 十进制数据表示三原色-->1
</RGB>
</BackColor>
<Position>
<!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
<positionX>
<!--ro, req, int, X坐标-->1
</positionX>
<positionY>
<!--ro, req, int, Y坐标-->1
</positionY>
<height>
<!--ro, req, int, 宽-->1
</height>
<width>
<!--ro, req, int, 高-->1
</width>
</Position>
<text>
<!--ro, opt, string, 标题文字-->test
</text>
</Prompt>
<TemperatureIcon>
<!--ro, opt, object, 温度正常图标-->
<enabled>
<!--ro, req, bool, 使能-->true
</enabled>
<Position>
<!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
<positionX>
<!--ro, req, int, X坐标-->1
</positionX>
<positionY>
<!--ro, req, int, Y坐标-->1
</positionY>
```

```

</Position>
<!--ro, req, int, 宽-->1
</height>
<!--ro, req, int, 高-->1
</width>
</Position>
<materialNo>
<!--ro, opt, int, 素材ID-->1
</materialNo>
</TemperatureIcon>
</NormalTemperature>
<NormalStatusAudioAlarm>
<!--ro, opt, object, 正常状态下音频报警信息, desc: 人员在检测时未发现上面的人数超限、未戴口罩、体温异常等事件时设备进行播报-->
<enabled>
<!--ro, req, bool, 使能-->true
</enabled>
<materialNo>
<!--ro, opt, int, 音频素材ID-->1
</materialNo>
</NormalStatusAudioAlarm>
</EventParam>
<AlarmDataSourceProperty>
<!--ro, opt, object, 数据源信息, desc: 该字段能力集缺少, 仅信发组件和设备使用-->
<PictureProperty>
<!--ro, opt, object, 图片属性-->
<PictureSwitchEffect>
<!--ro, opt, enum, 图片切换效果, subType:string, [LeftInRightOut#左进右出, rightInLeftOut#右进左出]-->LeftInRightOut
</PictureSwitchEffect>
<PictureScaleType>
<!--ro, opt, enum, 图片缩放类型, subType:string, [fitCenter#拉伸填充, centerCrop#截取填充, fixXY#适应宽高]-->fitCenter
</PictureScaleType>
</PictureProperty>
<AlarmDataSource>
<!--ro, opt, object, 数据源-->
<MaterialNo>
<!--ro, opt, int, 素材编号-->1
</MaterialNo>
<Name>
<!--ro, opt, string, 素材名称-->test
</Name>
<Address>
<!--ro, opt, string, 素材URL地址-->test
</Address>
<WebProperty>
<!--ro, opt, object, 网页属性-->
<webType>
<!--ro, opt, enum, 网页类型, subType:string, [desktop#桌面版, mobile#移动版], desc: 默认按照设备的系统类型 (Windows或Android) 来显示网页--
>desktop
</webType>
</WebProperty>
</AlarmDataSource>
</AlarmDataSourceProperty>
<ConferenceParam>
<!--ro, opt, object, 会议信息, desc: 当页面用于会议界面时有效-->
<backPicId>
<!--ro, opt, int, 背景图片素材ID-->1
</backPicId>
<Title>
<!--ro, opt, object, 会议标题 (主题), desc: 1、会议标题 (主题) 的文字内容通过场地预约配置协议下发, 对应/ISAPI/Publish/Signage/placeReservation?
format=json中的programID字段。2、可通过会议计划 (/ISAPI/InteractivePanel/Meeting/plan?format=json) 中的meetingTopic字段下发-->
<enabled>
<!--ro, req, bool, 使能-->true
</enabled>
<fontSize>
<!--ro, req, int, 字体大小-->1
</fontSize>
<fontColor>
<!--ro, req, object, 字体颜色-->
<RGB>
<!--ro, opt, int, 十进制数据表示三原色-->1
</RGB>
</FontColor>
<BackColor>
<!--ro, req, object, 背景颜色-->
<RGB>
<!--ro, opt, int, 十进制数据表示三原色-->1
</RGB>
</BackColor>
<Position>
<!--ro, opt, object, 位置, desc: 左上角为坐标原点, 取值范围和设备分辨率保持一致-->
<positionX>
<!--ro, req, int, X坐标-->1
</positionX>
<positionY>
<!--ro, req, int, Y坐标-->1
</positionY>
<height>
<!--ro, req, int, 宽-->1
</height>
<width>
<!--ro, req, int, 高-->1
</width>

```

```

</Position>
</Title>
<VoiceBroadcast>
  <!--ro, opt, object, 语音播报-->
  <enabled>
    <!--ro, opt, bool, 是否启用语音播报-->true
  </enabled>
  <broadcastSpeed>
    <!--ro, opt, enum, 播报语速, subType:string, [sLower#慢,sLow#较慢,medium#中速,fast#较快,faster#快]-->sLower
  </broadcastSpeed>
  <inLibVoice>
    <!--ro, opt, string, 在库人员语音播报内容-->test
  </inLibVoice>
  <outLibVoice>
    <!--ro, opt, string, 未知人员语音播报内容-->test
  </outLibVoice>
</VoiceBroadcast>
<WelcomeWord>
  <!--ro, opt, object, 欢迎词-->
  <enabled>
    <!--ro, opt, bool, 是否启用欢迎词-->true
  </enabled>
  <inLibWord>
    <!--ro, opt, string, 在库人员欢迎词内容, range:[1,128]-->test
  </inLibWord>
  <outLibWord>
    <!--ro, opt, string, 未知人员欢迎词内容, range:[1,128]-->test
  </outLibWord>
</WelcomeWord>
<ShowInfo>
  <!--ro, opt, object, 展示信息-->
  <row>
    <!--ro, opt, int, 窗口内人员签到展示行数-->1
  </row>
  <column>
    <!--ro, opt, int, 窗口内人员签到展示列数-->1
  </column>
  <Position>
    <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--ro, req, int, x坐标-->1
    </positionX>
    <positionY>
      <!--ro, req, int, y坐标-->1
    </positionY>
    <height>
      <!--ro, req, int, 宽-->1
    </height>
    <width>
      <!--ro, req, int, 高-->1
    </width>
  </Position>
  <inLib>
    <!--ro, opt, object, 在库信息-->
    <nameEnabled>
      <!--ro, opt, bool, 在库人员是否展示姓名-->true
    </nameEnabled>
  </inLib>
  <defaultShowInfoEnabled>
    <!--ro, opt, bool, 是否采用人员默认展示信息, desc:下发该字段, 且为true, 表示人员展示信息采用默认值。坐标信息等不用下发-->true
  </defaultShowInfoEnabled>
</ShowInfo>
<ModuleInfoList>
  <!--ro, opt, array, 界面模块信息列表, subType:object-->
  <ModuleInfo>
    <!--ro, opt, object, 界面模块信息-->
    <moduleType>
      <!--ro, req, enum, 模块类型, subType:string, [conferenceTime#会议时间(对应timeRange), conferenceSite#会议地点(对应会议室名称meetingName), conferenceStatus#会议状态(对应meetingStatus), signInPersonNum#已签到人数, totalPersonNum#应到人数, unSignInPersonNum#未签到人数, signInAndTotalPersonNum#已签到/应到人数, QRCode#会议室二维码, meetingHostName#主持人, employeeNo#参会人, customKey#自定义字段, nextMeeting#当天下一会议信息]-->conferenceTime
    </moduleType>
    <enabled>
      <!--ro, req, bool, 使能-->true
    </enabled>
    <fontSize>
      <!--ro, req, int, 字体大小-->1
    </fontSize>
    <fontColor>
      <!--ro, req, object, 字体颜色-->
      <RGB>
        <!--ro, opt, int, 十进制数据表示三原色-->1
      </RGB>
    </fontColor>
    <BackColor>
      <!--ro, opt, object, 背景颜色-->
      <RGB>
        <!--ro, opt, int, 十进制数据表示三原色-->1
      </RGB>
    </BackColor>
    <backPicId>
      <!--ro, opt, int, 背景图片素材ID, desc:背景图和背景颜色同时仅支持一种, 两者均下发, 设备优先解析背景图字段-->1
    </backPicId>
  </ModuleInfo>

```

```

<Position>
  <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
  <positionX>
    <!--ro, req, int, X坐标-->1
  </positionX>
  <positionY>
    <!--ro, req, int, Y坐标-->1
  </positionY>
  <height>
    <!--ro, req, int, 宽-->1
  </height>
  <width>
    <!--ro, req, int, 高-->1
  </width>
</Position>
<CustomKey>
  <!--ro, opt, object, 自定义字段-->
  <key>
    <!--ro, opt, string, 自定义字段-->test
  </key>
  <associateProtocolURL>
    <!--ro, opt, string, 关联协议URL-->test
  </associateProtocolURL>
</CustomKey>
</ModuleInfo>
</ModuleInfoList>
<conferenceStatus>
  <!--ro, opt, enum, 会议状态, subType:string, [idle#空闲,signing#签到中,meeting#开会中], desc:根据不同的会议状态, 配置不同的会议界面-->idle
</conferenceStatus>
</ConferenceParam>
<CallBtn>
  <!--ro, opt, object, 呼叫按键窗口-->
  <backPicId>
    <!--ro, opt, int, 背景图片素材ID-->1
  </backPicId>
</CallBtn>
<OpenDoorBtn>
  <!--ro, opt, object, 开门按键窗口-->
  <backPicId>
    <!--ro, opt, int, 背景图片素材ID-->1
  </backPicId>
</OpenDoorBtn>
<QRCodeBtn>
  <!--ro, opt, object, 二维码按键窗口-->
  <backPicId>
    <!--ro, opt, int, 背景图片素材ID-->1
  </backPicId>
</QRCodeBtn>
<AuthenticationSuccess>
  <!--ro, opt, object, 认证成功窗口-->
  <BackColor>
    <!--ro, opt, object, 背景颜色-->
    <RGB>
      <!--ro, opt, int, 十进制数据表示三原色-->1
    </RGB>
  </BackColor>
  <backPicId>
    <!--ro, opt, int, 背景图片素材ID, desc:背景图和背景颜色同时仅支持一种,两者均下发,设备优先解析背景图字段-->1
  </backPicId>
  <SubWindowList>
    <!--ro, opt, object, 子窗口列表-->
    <SubWindow>
      <!--ro, opt, object, 子窗口-->
      <subWindowType>
        <!--ro, req, enum, 子窗口类型, subType:string, [temperature#温度,facePic#人脸图片,name#姓名,employeeNo#工号,group#组织/部门,tips#提示信息]-->temperature
      </subWindowType>
      <enabled>
        <!--ro, req, bool, 使能-->true
      </enabled>
      <fontSize>
        <!--ro, opt, int, 字体大小, unit:px, unitType:图像显示-->1
      </fontSize>
      <fontColor>
        <!--ro, opt, int, 字体颜色-->1
      </fontColor>
      <fontStyle>
        <!--ro, opt, string, 字体风格, desc:bold-加粗, italic-倾斜, underline-下划线, strikethrough-删除线-->bold,italic,underline,strikethrough
      </fontStyle>
      <Position>
        <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
        <positionX>
          <!--ro, req, int, x坐标-->1
        </positionX>
        <positionY>
          <!--ro, req, int, y坐标-->1
        </positionY>
        <height>
          <!--ro, req, int, 宽-->1
        </height>
        <width>
          <!--ro, req, int, 高-->1
        </width>
      </Position>
    </SubWindow>
  </SubWindowList>

```

```

        </SubWindow>
    </SubWindowList>
</AuthenticationSuccess>
<AuthenticationFail>
    <!--ro, opt, object, 认证失败窗口-->
    <BackColor>
        <!--ro, opt, object, 背景颜色-->
        <RGB>
            <!--ro, opt, int, 十进制数据表示三原色-->1
        </RGB>
    </BackColor>
    <backPicId>
        <!--ro, opt, int, 背景图片素材ID, desc: 背景图和背景颜色同时仅支持一种, 两者均下发, 设备优先解析背景图字段-->1
    </backPicId>
    <SubWindowList>
        <!--ro, opt, object, 子窗口列表-->
        <SubWindow>
            <!--ro, opt, object, 子窗口-->
            <subWindowType>
                <!--ro, req, enum, 子窗口类型, subType:string, [temperature#温度, facePic#人脸图片, name#姓名, employeeNo#工号, group#组织/部门, tips#提示信息]-->temperature
            </subWindowType>
            <enabled>
                <!--ro, req, bool, 使能-->true
            </enabled>
            <fontSize>
                <!--ro, opt, int, 字体大小, unit:px, unitType:图像显示-->1
            </fontSize>
            <fontColor>
                <!--ro, opt, int, 字体颜色-->1
            </fontColor>
            <fontStyle>
                <!--ro, opt, string, 字体风格, desc:bold-加粗, italic-倾斜, underline-下划线, strikethrough-删除线-->b,italic,underline,strikeout
            </fontStyle>
            <Position>
                <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
                <positionX>
                    <!--ro, req, int, x坐标-->1
                </positionX>
                <positionY>
                    <!--ro, req, int, y坐标-->1
                </positionY>
                <height>
                    <!--ro, req, int, 宽-->1
                </height>
                <width>
                    <!--ro, req, int, 高-->1
                </width>
            </Position>
            <facePicID>
                <!--ro, opt, int, 默认人脸图片ID, step:1, dep:and,
                </facePicID>
            </SubWindow>
        </SubWindowList>
    </AuthenticationFail>
    <StatusBar>
        <!--ro, opt, object, 状态栏, desc:当素材类型为状态栏时有效, 且必填-->
        <backPicId>
            <!--ro, opt, int, 背景图片素材ID, desc:背景图和背景颜色同时仅支持一种, 两者均下发, 设备优先解析背景图字段-->1
        </backPicId>
        <statusBarType>
            <!--ro, req, enum, 状态栏类型, subType:string, [fixed#固定的], desc:当为固定类型时, 窗口位置不生效, 设备将状态栏固定在顶部显示-->fixed
        </statusBarType>
        <SubStatusList>
            <!--ro, opt, object, 子状态列表-->
            <SubStatus>
                <!--ro, opt, object, 子状态-->
                <statusType>
                    <!--ro, req, enum, 子状态, subType:string, [dial#拨号状态 (3G4G等), wifi#wifi状态, EZVIZ#萤石状态, wiredNetwork#有线网络状态, deploy#布防状态, temperature#测温]-->dial
                </statusType>
                <enabled>
                    <!--ro, req, bool, 是否显示-->true
                </enabled>
            </SubStatus>
        </SubStatusList>
    </StatusBar>
</Windows>
</WindowsList>
</Page>
</PageList>
<programSize>
    <!--opt, int, 节目大小, unit:Byte, unitType:信息量-->1
</programSize>
<programLength>
    <!--opt, int, 节目时长, unit:s, unitType:时间-->1
</programLength>
<coordinateType>
    <!--opt, enum, 节目中坐标基准值类型, subType:string, [uniformCoordinate#统一坐标 (实际的分辨率坐标按基准坐标等比例进行转换), resolutionCoordinate#实际分辨率坐标]-->uniformCoordinate
</coordinateType>
<screenType>

```

```
<!--opt, enum, 屏幕类型, subType:string, [first#主屏,second#辅屏], desc:当programType为doubleSide表示创建双目屏节目时生效-->first
</screenType>
<audioEnabled>
  <!--opt, bool, 节目音频使能, desc:当programType为doubleSide表示创建双目屏节目时生效,用来防止主屏声音和辅屏声音混淆,主辅屏节目的音频只开一个-->true
</audioEnabled>
<PageIdList>
  <!--opt, array, 页面id列表, subType:object-->
  <PageId>
    <!--opt, object, 页面id-->
    <id>
      <!--req, int, 页面ID-->1
    </id>
  </PageId>
</PageIdList>
<programUUID>
  <!--ro, opt, string, 节目UUID, range:[1,32], desc:平台下发-->test
</programUUID>
</Program>
```

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 响应消息, attr:version{ro, req, string, 版本}-->
  <requestURL>
    <!--ro, req, string, 请求的URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, 状态码, subType:int, [0#OK,1#OK,2#Device Busy,3#Device Error,4#Invalid Operation,5#Invalid XML Format,6#Invalid XML Content,7#Reboot Required]-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, 状态信息, subType:string, [OK#成功,Device Busy#设备忙,Device Error#设备异常,Invalid Operation#无效的操作,Invalid XML Format#无效的XML格式,Invalid XML Content#无效的XML内容,Reboot#设备重启]-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, 详细错误码英文描述, desc:详细错误码的英文描述-->OK
  </subStatusCode>
  <ID>
    <!--ro, opt, string, 索引-->test
  </ID>
  <AttachInfo>
    <!--ro, opt, object, 附加信息-->
    <StatusList>
      <!--ro, req, array, 状态列表, subType:object-->
      <Status>
        <!--ro, req, object, 状态-->
        <statusCode>
          <!--ro, req, enum, 状态码, subType:int, [0#OK,1#OK,2#Device Busy,3#Device Error,4#Invalid Operation,5#Invalid XML Format,6#Invalid XML Content,7#Reboot Required]-->0
        </statusCode>
        <statusString>
          <!--ro, req, enum, 状态信息, subType:string, [OK#成功,Device Busy#设备忙,Device Error#设备异常,Invalid Operation#无效的操作,Invalid XML Format#无效的XML格式,Invalid XML Content#无效的XML内容,Reboot#设备重启]-->OK
        </statusString>
        <subStatusCode>
          <!--ro, req, string, 详细错误码英文描述, desc:详细错误码的英文描述-->OK
        </subStatusCode>
        <ID>
          <!--ro, opt, string, 索引-->test
        </ID>
      </Status>
    </StatusList>
  </AttachInfo>
</ResponseStatus>
```

69.66.6 配置单个节目参数

Request URL

PUT /ISAPI/Publish/ProgramMgr/program/<programID>

查询参数

参数名称	参数类型	描述
programID	string	节目ID

请求报文

```
<?xml version="1.0" encoding="UTF-8"?>

<Program xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
```

```
<!--req, object, 节目信息, attr:version{req, string, 协议版本}-->
<id>
<!--req, int, 节目索引-->1
</id>
<programName>
<!--req, string, 节目名称-->test
</programName>
<programRemarks>
<!--req, string, 节目描述-->test
</programRemarks>
<shareProperty>
<!--opt, enum, 共享属性, subType:string, [public#公有,private#私有]-->public
</shareProperty>
<approveState>
<!--opt, enum, 审核状态标志, subType:string, [approved#审核通过,notPass#未通过,notApprove#未审核]-->approved
</approveState>
<approveRemarks>
<!--opt, string, 审核意见-->test
</approveRemarks>
<programType>
<!--opt, enum, 节目类型, subType:string, [normal#普通,decode#解码,touch#触摸,decodeTouch#解码触摸,doubleSide#双面屏,conference#会议,screensaver#图片/屏保(门禁),character#欢迎词(门禁),mainInterface#主界面(门禁),txt#文本节目(门禁),advertise#广告], desc:当节目类型为conference#会议,screensaver#图片/屏保(门禁),character#欢迎词(门禁),mainInterface#主界面(门禁),txt#文本节目(门禁),advertise#广告时,该节目不依赖与日程,直接单独下发该节目后,在设备上展示该节目下的页面信息。-->normal
</programType>
<orgNo>
<!--opt, int, 所属组织编号-->1
</orgNo>
<Resolution>
<!--req, object, 分辨率-->
<resolutionName>
<!--opt, string, 分辨率名称-->test
</resolutionName>
<imageWidth>
<!--req, int, 分辨率宽-->1
</imageWidth>
<imageHeight>
<!--req, int, 分辨率高-->1
</imageHeight>
</Resolution>
<PageList>
<!--ro, opt, object, 页面列表-->
<Page>
<!--ro, req, object, 页面-->
<id>
<!--ro, req, int, 索引-->1
</id>
<PageBasicInfo>
<!--ro, req, object, 页面基本信息-->
<pageName>
<!--ro, opt, string, 页面名称-->test
</pageName>
<BackgroundColor>
<!--ro, opt, object, 背景颜色-->
<RGB>
<!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
</RGB>
</BackgroundColor>
<playDurationMode>
<!--ro, opt, enum, 页面播放时间模式, subType:string, [selfDefine#自定义,auto#自动,loop#循环], desc:自定义时playDuration节点有效;自动时根据播放内容的时长*playCount计算;循环时表示一直循环-->selfDefine
</playDurationMode>
<playDuration>
<!--ro, opt, int, 页面播放时长, unit:s, unitType:时间-->1
</playDuration>
<playCount>
<!--ro, opt, int, 页面播放次数-->1
</playCount>
<switchDuration>
<!--ro, opt, int, 切换间隔, unit:s, unitType:时间-->1
</switchDuration>
<switchEffect>
<!--ro, opt, enum, 切换效果, subType:string, [none#无,random#随机,boxShrink#盒状收缩,boxSpread#盒状展开,cycleShrink#圆形收缩,cycleSpread#圆形展开,eraseUp#向上擦除,eraseDown#向下擦除,eraseLeft#向左擦除,eraseRight#向右擦除,verticalLSheLter#垂直屏蔽,horizontalLSheLter#水平屏蔽,verticalChessboard#纵向棋盘式,horizontalChessboard#横向棋盘式,dissolve#随机溶解,LeftRightToCenter#左右向中间缩进,ceterToLeftRight#中央向左右扩展,upDownToCenter#上下向中央缩进,centerToUpDown#中央向上下扩展,drawOutLeftDown#从左上抽出,drawOutLeftUp#从左上抽出,drawOutRightDown#从右下抽出,drawOutRightUp#从右上抽出,verticalLLine#随机垂直线条,horizontalLLine#随机垂直线条]-->none
</switchEffect>
<backgroundPic>
<!--ro, opt, int, 背景图片素材ID-->1
</backgroundPic>
<SignInCfg>
<!--ro, opt, object, 签到信息, desc:当页面用于签到或考勤界面时有效-->
<VoiceBroadcast>
<!--ro, opt, object, 语音播报-->
<enabled>
<!--ro, opt, bool, 是否启用语音播报-->true
</enabled>
<broadcastSpeed>
<!--ro, opt, enum, 播报语速, subType:string, [sLower#慢,sLow#较慢,medium#中速,fast#较快,faster#快]-->sLower
</broadcastSpeed>
<inLibVoice>
<!--ro, opt, string, 在库人员正常体温语音播报内容-->test
</inLibVoice>
```

```
<outLibVoice>
  <!--ro, opt, string, 未知人员正常体温语音播报内容-->test
</outLibVoice>
<inLibAbnormalTemperatureVoice>
  <!--ro, opt, string, 在库人员异常温度语音播报内容-->test
</inLibAbnormalTemperatureVoice>
<outLibAbnormalTemperatureVoice>
  <!--ro, opt, string, 未知人员异常温度语音播报内容-->test
</outLibAbnormalTemperatureVoice>
</VoiceBroadcast>
<WelcomeWord>
  <!--ro, opt, object, 欢迎词-->
  <enabled>
    <!--ro, opt, bool, 是否启用欢迎词-->true
  </enabled>
  <inLibWord>
    <!--ro, opt, string, 在库人员欢迎词内容-->test
  </inLibWord>
  <outLibWord>
    <!--ro, opt, string, 未知人员欢迎词内容-->test
  </outLibWord>
</WelcomeWord>
<SignCartoon>
  <!--ro, opt, object, 签到动画-->
  <signCartoonSpeed>
    <!--ro, opt, enum, 动画速度, subType:string, [fast#快,medium#终端,slow#慢]-->fast
  </signCartoonSpeed>
</SignCartoon>
<backgroundSignEnabled>
  <!--ro, opt, bool, 是否启用后台签到-->true
</backgroundSignEnabled>
<TemperatureDetection>
  <!--ro, opt, object, 温度监测参数-->
  <enabled>
    <!--ro, req, bool, 是否启用-->true
  </enabled>
  <threshold>
    <!--ro, req, float, 正常温度值-->0.000
  </threshold>
</TemperatureDetection>
</SignInCfg>
</PageBasicInfo>
<characterMode>
  <!--ro, opt, enum, 页面中的欢迎词模式, subType:string, [mode1#模式1,mode2#模式2,mode3#模式3], desc:门禁增加, 门禁欢迎词节目位置固定, 且存在三种模式-
->mode1
  </characterMode>
<windowsList>
  <!--ro, opt, array, 窗口信息列表, subType:object-->
  <windows>
    <!--ro, opt, object, 窗口信息-->
    <id>
      <!--ro, req, int, 内容编号-->1
    </id>
    <position>
      <!--ro, opt, object, 内容位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
      <positionX>
        <!--ro, req, int, x坐标-->1
      </positionX>
      <positionY>
        <!--ro, req, int, y坐标-->1
      </positionY>
      <height>
        <!--ro, req, int, 宽-->1
      </height>
      <width>
        <!--ro, req, int, 高-->1
      </width>
    </position>
    <CornerRadius>
      <!--ro, opt, object, 圆角配置, desc:目前仅对picture图片窗口和character文字窗口有效-->
      <TopLeft>
        <!--ro, opt, int, 左上角半径-->1
      </TopLeft>
      <TopRight>
        <!--ro, opt, int, 右上角半径-->1
      </TopRight>
      <BottomLeft>
        <!--ro, opt, int, 左下角半径-->1
      </BottomLeft>
      <BottomRight>
        <!--ro, opt, int, 右下角半径-->1
      </BottomRight>
    </CornerRadius>
    <layerNo>
      <!--ro, req, int, 图层编号-->1
    </layerNo>
    <winMaterialInfo>
      <!--ro, opt, object, 窗口素材信息-->
      <materialType>
        <!--ro, req, enum, 素材类型, subType:string, [static#静态,dynamic#动态,other#其他]-->static
      </materialType>
      <staticMaterialType>
        <!--ro, opt, enum, 静态素材类型, subType:string, [picture#图片,flash#flash,audio#音频,video#视频,document#文本,ppt#ppt,doc#文档,excel#表
```

格,pdf#pdf,web#网页,app#app,signIn#签到,conference#会议]-->picture

```
</staticMaterialType>
<dynamicType>
  <!--ro, opt, enum, 动态素材类型, subType:string,
[web#web,socket#socket,rss#rss,realStream#realStream,call#call,dynamicPic#dynamicPic,capturePic#capturePic,character#character]-->web
</dynamicType>
<otherType>
  <!--ro, opt, enum, 其他素材类型, subType:string, [clock#时钟,weather#天气,countdown#倒计时,localInput#本地输入,hyperLinkBtn#超链接按钮,event#
事件,callBtn#呼叫按钮,openDoorBtn#开门按钮,QRCODEBtn#二维码按钮,authenticationSuccess#认证成功,authenticationFail#认证失败,statusBar#状态栏]-->clock
</otherType>
</WinMaterialInfo>
<TouchProperty>
  <!--ro, opt, object, 触摸属性-->
<windType>
  <!--ro, opt, enum, 窗口类型, subType:string, [popup#弹窗,page#页面]-->popup
</windType>
<hyperLinkType>
  <!--ro, opt, enum, 超链接类型, subType:string, [window#页面超链接,page#窗口超链接], desc:窗口类型为弹窗时有效-->window
</hyperLinkType>
<windowId>
  <!--ro, opt, int, 窗口编号, desc:当前页面的窗口,超链接类型为窗口时有效-->1
</windowId>
<pageId>
  <!--ro, opt, int, 页面编号, desc:超链接类型为页面时有效-->1
</pageId>
</TouchProperty>
<PlayItemList>
  <!--ro, opt, array, 窗口播放列表, subType:object-->
  <PlayItem>
    <!--ro, req, object, 窗口播放信息-->
    <id>
      <!--ro, req, int, 播放序号-->1
    </id>
    <materialNo>
      <!--ro, opt, int, 素材编号-->1
    </materialNo>
    <inputChannel>
      <!--ro, opt, int, 绑定的IPC通道号-->1
    </inputChannel>
    <playEffect>
      <!--ro, req, enum, 播放效果, subType:string, [none#无,marquee#跑马灯]-->none
    </playEffect>
    <MarqueeInfo>
      <!--ro, opt, object, 跑马灯信息-->
      <scrollType>
        <!--ro, req, enum, 跑马灯滚动类型, subType:string, [none#不滚动,loops#循环滚动,once#滚动一次,backAndForth#来回滚动]-->none
      </scrollType>
      <scrollDeriction>
        <!--ro, req, enum, 跑马灯滚动方向, subType:string, [none#无,up#向上,down#向下,left#向左,right#向右]-->none
      </scrollDeriction>
      <scrollSpeed>
        <!--ro, req, int, 跑马灯滚动速度-->1
      </scrollSpeed>
    </MarqueeInfo>
    <PlayDuration>
      <!--ro, opt, object, 窗口内素材播放时间, desc:静态素材可以配置该时间,视频直播和IPC通道都可配置该节点-->
      <durationType>
        <!--ro, req, enum, 持续方式, subType:string, [materialTime#按素材时间,selfDefine#自定义]-->selfDefine
      </durationType>
      <duration>
        <!--ro, req, int, 持续时间, unit:s, unitType:时间-->1
      </duration>
    </PlayDuration>
    <CharactersEffect>
      <!--ro, opt, object, 文字显示效果, desc:当素材类型为文本txt和时有效-->
      <fontSize>
        <!--ro, req, int, 字体大小-->1
      </fontSize>
      <fontColor>
        <!--ro, req, object, 字体颜色-->
        <RGB>
          <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
        </RGB>
      </FontColor>
      <backgroundType>
        <!--ro, opt, enum, 背景类型, subType:string, [color#颜色(默认),picture#图片], desc:字段不存在时,默认为颜色-->color
      </backgroundType>
      <backPicId>
        <!--ro, opt, int, 背景图片素材ID-->1
      </backPicId>
      <BackColor>
        <!--ro, req, object, 背景颜色-->
        <RGB>
          <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
        </RGB>
      </BackColor>
      <backTransparent>
        <!--ro, req, int, 背景透明度-->1
      </backTransparent>
      <subtitlesEnabled>
        <!--ro, req, bool, 子标题使能-->true
      </subtitlesEnabled>
      <scrollDirection>
        <!--ro, req, enum, 字幕滚动方向, subType:string, [left#向左,right#向右,up#向上,down#向下]-->left
      </scrollDirection>
```

```
<!--ro, req, enum, 关于滚动方向, subType:string, [left#向左, right#向右, up#向上, down#向下]-->1
</scrollDirection>
<scrollSpeed>
  <!--ro, req, int, 文字滚动速度-->1
</scrollSpeed>
</CharactersEffect>
<switchEffect>
  <!--ro, opt, enum, 窗口素材切换效果, subType:string, [none#无, leftInRightOut#左进右出, rightInLeftOut#右进左出, bottomInTopOut#下进上
出, topInBottomOut#上进下出, fadeInFadeOut#淡入淡出, middleExit#中间退出, topPop#顶部弹出, rightBottomIn#右下角进入, leftTopIn#左上角进入, horizontalOpen#水平打
开, verticalOpen#垂直打开, random#随机效果], desc:图片素材时有效-->random
</switchEffect>
<pageTime>
  <!--ro, opt, int, 翻页时间, unit:s, unitType:时间, desc:当素材为word,ppt,pdf,excel时有效-->1
</pageTime>
<scrollSpeed>
  <!--ro, req, int, 滚动速度, desc:当素材为静态web时有效-->1
</scrollSpeed>
<scaleType>
  <!--ro, opt, enum, 素材缩放参数配置, subType:string, [fitXY#拉伸填充, fitCenter#自适应, centerCrop#截取填充], desc:目前仅当图片窗口有效--
>centerCrop
</scaleType>
<playType>
  <!--ro, opt, enum, 播放方式, subType:string, [normal#正常, faceAttendance#考勤, PeopleCounting#客流, temperature#测温], desc:目前仅当图片窗口有
效-->normal
</playType>
<CharactersAttribute>
  <!--ro, opt, object, 文字属性, desc:当dynamicType为character时有效-->
  <fontSize>
    <!--ro, req, int, 字体大小-->1
  </fontSize>
  <fontColor>
    <!--ro, req, object, 字体颜色-->
    <RGB>
      <!--ro, opt, int, RGB值, desc:十进制数据表示三原色, 如16777215表示0xFFFFFF-->1
    </RGB>
  </FontColor>
  <backgroundType>
    <!--ro, req, enum, 背景类型, subType:string, [color#颜色, picture#图片]-->color
  </backgroundType>
  <BackColor>
    <!--ro, req, object, 背景色-->
    <RGB>
      <!--ro, opt, int, RGB值, desc:十进制数据表示三原色, 如16777215表示0xFFFFFF-->1
    </RGB>
  </BackColor>
  <backTransparent>
    <!--ro, opt, int, 背景透明度-->1
  </backTransparent>
  <alignType>
    <!--ro, req, enum, 对齐方式, subType:string, [left#左, right#右, middle#中]-->middle
  </alignType>
  <verticalAlignType>
    <!--ro, req, enum, 垂直对齐方式, subType:string, [top#顶, bottom#底, verticalCenter#中]-->top
  </verticalAlignType>
  <characterContent>
    <!--ro, opt, string, 文字内容, range:[0,512]-->test
  </characterContent>
  <backPicId>
    <!--ro, opt, int, 背景图片素材ID-->1
  </backPicId>
  <characterStyle>
    <!--ro, req, enum, 字体效果, subType:string, [normal#常规, bold#粗体]-->normal
  </characterStyle>
  <characterType>
    <!--ro, req, enum, 字体类型, subType:string, [mainTitle#主标题, subTitle#副标题]-->mainTitle
  </characterType>
</CharactersAttribute>
<WebProperty>
  <!--ro, opt, object, 网页属性-->
  <webType>
    <!--ro, opt, enum, 网页类型, subType:string, [desktop#桌面版, mobile#移动版], desc:默认按照设备的系统类型（Windows或Android）来显示网页--
>desktop
  </webType>
</WebProperty>
</PlayItem>
</PlayItemList>
<enabledAudio>
  <!--ro, opt, bool, 音频使能-->true
</enabledAudio>
<enableHide>
  <!--ro, opt, bool, 隐藏使能-->true
</enableHide>
<enableLock>
  <!--ro, opt, bool, 锁定使能-->true
</enableLock>
<AppWindow>
  <!--ro, opt, object, App窗口信息-->
  <WindowInfoList>
    <!--ro, req, array, 窗口信息列表, subType:object-->
    <WindowInfo>
      <!--ro, req, object, 窗口信息-->
      <id>
        <!--ro, req, int, 索引-->1
      </id>
```

```

        <materialNo>
            <!--ro, req, int, 素材编号, desc:采用本字段下发时, 用户需要先将App安装包当做素材存储到设备中, 得到素材编号再进行下发-->1
        </materialNo>
        <path>
            <!--ro, opt, string, App路径, range:[0,1024],
desc:path和materialNo字段同时存在时, 设备优先解析path字段;
采用本字段下发时, 设备需要确保App已经安装在设备内, 直接传入App的路径即可;
对于Android设备, path传的是App的包名; 对于Windows设备, path传的是App的应用程序绝对路径; -->test
        </path>
        </WindowInfo>
    </WindowInfoList>
</AppWindow>
<DataSource>
    <!--ro, opt, object, 数据来源, desc:当窗口为叫号和动态弹图时有效-->
    <materialNo>
        <!--ro, req, int, 素材编号-->1
    </materialNo>
</DataSource>
<Call>
    <!--ro, opt, object, 叫号数据-->
    <materialNo>
        <!--ro, req, int, 素材编号-->1
    </materialNo>
    <tableRow>
        <!--ro, req, int, 表格行-->1
    </tableRow>
    <tableColumn>
        <!--ro, req, int, 表格列-->1
    </tableColumn>
    <tableDirection>
        <!--ro, req, enum, 表格方向, subType:string, [vertical#纵向,horizontal#纵向]-->vertical
    </tableDirection>
    <tableType>
        <!--ro, req, enum, 表格类型, subType:string, [template1#模板1,template2#模板2,template3#模板3,template4#模板4,template5#模板5,template6#模板
6]-->template1
    </tableType>
    <backPicId>
        <!--ro, opt, int, 背景图片素材编号-->1
    </backPicId>
    <alignType>
        <!--ro, req, enum, 对齐方式, subType:string, [Left#左,right#右,middle#中]-->left
    </alignType>
    <refreshDirection>
        <!--ro, req, enum, 刷新方向, subType:string, [upTdown#从上到下,downToup#从下到上,LeftToRight#从左到右,rightToLeft#从右到左]-->upTdown
    </refreshDirection>
    <HeadDataList>
        <!--ro, opt, array, 表头数据列表, subType:object-->
        <HeadData>
            <!--ro, opt, object, 表头数据-->
            <id>
                <!--ro, req, int, 索引-->1
            </id>
            <data>
                <!--ro, req, string, 数据-->test
            </data>
        </HeadData>
    </HeadDataList>
    <ItemStyleList>
        <!--ro, opt, array, 表格列或行样式列表, subType:object-->
        <ItemStyle>
            <!--ro, opt, object, 表格列或行样式-->
            <id>
                <!--ro, req, int, 索引-->1
            </id>
            <width>
                <!--ro, req, int, 每列宽度所占百分比-->1
            </width>
            <fontSize>
                <!--ro, req, int, 字体大小-->1
            </fontSize>
            <FontColor>
                <!--ro, req, object, 字体颜色-->
                <RGB>
                    <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
                </RGB>
            </FontColor>
            <BackColor>
                <!--ro, req, object, 背景色-->
                <RGB>
                    <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
                </RGB>
            </BackColor>
        </ItemStyle>
    </ItemStyleList>
</Call>
<DynamicPic>
    <!--ro, opt, object, 动态弹图窗口配置-->
    <materialNo>
        <!--ro, req, int, 素材编号-->1
    </materialNo>
    <backPicId>
        <!--ro, opt, int, 背景图片素材ID-->1
    </backPicId>

```

```

</dynamic>
<CapturePic>
  <!--ro, opt, object, 抓拍图片参数-->
  <backPicId>
    <!--ro, opt, int, 背景图片素材ID-->1
  </backPicId>
  <ipcMaterialNo>
    <!--ro, req, int, IPC素材编号-->1
  </ipcMaterialNo>
  <cancelType>
    <!--ro, req, enum, 取消方式, subType:string, [auto#自动,manual#手动]-->manual
  </cancelType>
  <duration>
    <!--ro, opt, int, 持续时间, unit:s, unitType:时间-->1
  </duration>
</CapturePic>
<ClockParam>
  <!--ro, opt, object, 时钟参数-->
  <backPicId>
    <!--ro, opt, int, 背景图片素材ID-->1
  </backPicId>
  <ClockIcon>
    <!--ro, req, object, 时钟图标参数-->
    <enabled>
      <!--ro, req, bool, 使能-->true
    </enabled>
    <type>
      <!--ro, opt, enum, 时钟类型, subType:string, [cLock1#cLock1,cLock2#cLock2]-->cLock1
    </type>
    <Position>
      <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
      <positionX>
        <!--ro, req, int, x坐标-->1
      </positionX>
      <positionY>
        <!--ro, req, int, y坐标-->1
      </positionY>
      <height>
        <!--ro, req, int, 宽-->1
      </height>
      <width>
        <!--ro, req, int, 高-->1
      </width>
    </Position>
  </ClockIcon>
  <YmdParam>
    <!--ro, req, object, 时钟年月日参数-->
    <enabled>
      <!--ro, req, bool, 使能-->true
    </enabled>
    <fontSize>
      <!--ro, req, int, 字体大小-->1
    </fontSize>
    <fontColor>
      <!--ro, req, object, 字体颜色-->
      <RGB>
        <!--ro, opt, int, RGB值, desc:十进制数据表示三原色, 如16777215表示0xFFFFFF-->1
      </RGB>
    </fontColor>
    <BackColor>
      <!--ro, req, object, 背景色-->
      <RGB>
        <!--ro, opt, int, RGB值, desc:十进制数据表示三原色, 如16777215表示0xFFFFFF-->1
      </RGB>
    </BackColor>
    <Position>
      <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
      <positionX>
        <!--ro, req, int, x坐标-->1
      </positionX>
      <positionY>
        <!--ro, req, int, y坐标-->1
      </positionY>
      <height>
        <!--ro, req, int, 宽-->1
      </height>
      <width>
        <!--ro, req, int, 高-->1
      </width>
    </Position>
  </YmdParam>
  <HmsParam>
    <!--ro, req, object, 时钟时分秒参数-->
    <enabled>
      <!--ro, req, bool, 使能-->true
    </enabled>
    <fontSize>
      <!--ro, req, int, 字体大小-->1
    </fontSize>
    <fontColor>
      <!--ro, req, object, 字体颜色-->
      <RGB>
        <!--ro, opt, int, RGB值, desc:十进制数据表示三原色, 如16777215表示0xFFFFFF-->1
      </RGB>

```

```
</FontColor>
<BackColor>
  <!--ro, req, object, 背景色-->
  <RGB>
    <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
  </RGB>
</BackColor>
<Position>
  <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
  <positionX>
    <!--ro, req, int, x坐标-->1
  </positionX>
  <positionY>
    <!--ro, req, int, y坐标-->1
  </positionY>
  <height>
    <!--ro, req, int, 宽-->1
  </height>
  <width>
    <!--ro, req, int, 高-->1
  </width>
</Position>
</HmsParam>
<WeekParam>
  <!--ro, req, object, 时钟星期参数-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <fontSize>
    <!--ro, req, int, 字体大小-->1
  </fontSize>
  <FontColor>
    <!--ro, req, object, 字体颜色-->
    <RGB>
      <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
    </RGB>
  </FontColor>
<BackColor>
  <!--ro, req, object, 背景色-->
  <RGB>
    <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
  </RGB>
</BackColor>
<Position>
  <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
  <positionX>
    <!--ro, req, int, x坐标-->1
  </positionX>
  <positionY>
    <!--ro, req, int, y坐标-->1
  </positionY>
  <height>
    <!--ro, req, int, 宽-->1
  </height>
  <width>
    <!--ro, req, int, 高-->1
  </width>
</Position>
</WeekParam>
</ClockParam>
<WeatherParam>
  <!--ro, opt, object, 天气参数-->
  <backPicId>
    <!--ro, opt, int, 天气背景图片素材id-->1
  </backPicId>
  <WeatherIcon>
    <!--ro, opt, object, 天气图标参数-->
    <enabled>
      <!--ro, req, bool, 使能-->true
    </enabled>
    <Position>
      <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
      <positionX>
        <!--ro, req, int, x坐标-->1
      </positionX>
      <positionY>
        <!--ro, req, int, y坐标-->1
      </positionY>
      <height>
        <!--ro, req, int, 宽-->1
      </height>
      <width>
        <!--ro, req, int, 高-->1
      </width>
    </Position>
  </WeatherIcon>
  <Date>
    <!--ro, opt, object, 天气日期参数-->
    <enabled>
      <!--ro, req, bool, 使能-->true
    </enabled>
    <fontSize>
      <!--ro, req, int, 字体大小-->1
    </fontSize>
```

```
</fontSize>
<FontColor>
  <!--ro, req, object, 字体颜色-->
  <RGB>
    <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
  </RGB>
</FontColor>
<BackColor>
  <!--ro, req, object, 背景色-->
  <RGB>
    <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
  </RGB>
</BackColor>
<Position>
  <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
  <positionX>
    <!--ro, req, int, x坐标-->1
  </positionX>
  <positionY>
    <!--ro, req, int, y坐标-->1
  </positionY>
  <height>
    <!--ro, req, int, 宽-->1
  </height>
  <width>
    <!--ro, req, int, 高-->1
  </width>
</Position>
</Date>
<Temperature>
  <!--ro, opt, object, 天气温度参数-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <fontSize>
    <!--ro, req, int, 字体大小-->1
  </fontSize>
  <FontColor>
    <!--ro, req, object, 字体颜色-->
    <RGB>
      <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
    </RGB>
  </FontColor>
  <BackColor>
    <!--ro, req, object, 背景色-->
    <RGB>
      <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
    </RGB>
  </BackColor>
  <Position>
    <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--ro, req, int, x坐标-->1
    </positionX>
    <positionY>
      <!--ro, req, int, y坐标-->1
    </positionY>
    <height>
      <!--ro, req, int, 宽-->1
    </height>
    <width>
      <!--ro, req, int, 高-->1
    </width>
  </Position>
</Temperature>
<WeatherContent>
  <!--ro, opt, object, 天气情况参数-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <fontSize>
    <!--ro, req, int, 字体大小-->1
  </fontSize>
  <FontColor>
    <!--ro, req, object, 字体颜色-->
    <RGB>
      <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
    </RGB>
  </FontColor>
  <BackColor>
    <!--ro, req, object, 背景色-->
    <RGB>
      <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
    </RGB>
  </BackColor>
  <Position>
    <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--ro, req, int, x坐标-->1
    </positionX>
    <positionY>
      <!--ro, req, int, y坐标-->1
    </positionY>
    <height>
```

```
</height>
<!--ro, req, int, 宽-->1
</height>
<width>
<!--ro, req, int, 高-->1
</width>
</Position>
</WeatherContent>
<City>
<!--ro, opt, object, 天气城市参数-->
<enabled>
<!--ro, req, bool, 使能-->true
</enabled>
<cityId>
<!--ro, req, string, 城市编号-->test
</cityId>
<cityName>
<!--ro, req, string, 城市名称-->test
</cityName>
<fontSize>
<!--ro, req, int, 字体大小-->1
</fontSize>
<FontColor>
<!--ro, req, object, 字体颜色-->
<RGB>
<!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
</RGB>
</FontColor>
<BackColor>
<!--ro, req, object, 背景色-->
<RGB>
<!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
</RGB>
</BackColor>
<Position>
<!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
<positionX>
<!--ro, req, int, x坐标-->1
</positionX>
<positionY>
<!--ro, req, int, y坐标-->1
</positionY>
<height>
<!--ro, req, int, 宽-->1
</height>
<width>
<!--ro, req, int, 高-->1
</width>
</Position>
</City>
</Humidity>
<!--ro, opt, object, 天气湿度参数-->
<enabled>
<!--ro, req, bool, 使能-->true
</enabled>
<fontSize>
<!--ro, req, int, 字体大小-->1
</fontSize>
<FontColor>
<!--ro, req, object, 字体颜色-->
<RGB>
<!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
</RGB>
</FontColor>
<BackColor>
<!--ro, req, object, 背景色-->
<RGB>
<!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
</RGB>
</BackColor>
<Position>
<!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
<positionX>
<!--ro, req, int, x坐标-->1
</positionX>
<positionY>
<!--ro, req, int, y坐标-->1
</positionY>
<height>
<!--ro, req, int, 宽-->1
</height>
<width>
<!--ro, req, int, 高-->1
</width>
</Position>
</Humidity>
<AirQuality>
<!--ro, opt, object, 天气空气质量参数-->
<enabled>
<!--ro, req, bool, 使能-->true
</enabled>
<fontSize>
<!--ro, req, int, 字体大小-->1
</fontSize>
```

```
<FontColor>
  <!--ro, req, object, 字体颜色-->
  <RGB>
    <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
  </RGB>
</FontColor>
<BackColor>
  <!--ro, req, object, 背景色-->
  <RGB>
    <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
  </RGB>
</BackColor>
<Position>
  <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
  <positionX>
    <!--ro, req, int, x坐标-->1
  </positionX>
  <positionY>
    <!--ro, req, int, y坐标-->1
  </positionY>
  <height>
    <!--ro, req, int, 宽-->1
  </height>
  <width>
    <!--ro, req, int, 高-->1
  </width>
</Position>
</AirQuality>
<UpdateTime>
  <!--ro, opt, object, 天气更新时间参数-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <refreshTime>
    <!--ro, req, string, 刷新时间, desc:设备早期返回时间格式有误, 错为: 20200826T051500+08, 为保证XSD校验通过, 改为string类型-->20200826T051500+08
  </refreshTime>
  <updateInterval>
    <!--ro, req, int, 刷新间隔, unit:minute-->1
  </updateInterval>
  <fontSize>
    <!--ro, req, int, 字体大小-->1
  </fontSize>
  <FontColor>
    <!--ro, req, object, 字体颜色-->
    <RGB>
      <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
    </RGB>
  </FontColor>
  <BackColor>
    <!--ro, req, object, 背景色-->
    <RGB>
      <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
    </RGB>
  </BackColor>
  <Position>
    <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--ro, req, int, x坐标-->1
    </positionX>
    <positionY>
      <!--ro, req, int, y坐标-->1
    </positionY>
    <height>
      <!--ro, req, int, 宽-->1
    </height>
    <width>
      <!--ro, req, int, 高-->1
    </width>
  </Position>
</UpdateTime>
<Wind>
  <!--ro, opt, object, 天气风力参数-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <fontSize>
    <!--ro, req, int, 字体大小-->1
  </fontSize>
  <FontColor>
    <!--ro, req, object, 字体颜色-->
    <RGB>
      <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
    </RGB>
  </FontColor>
  <BackColor>
    <!--ro, req, object, 背景色-->
    <RGB>
      <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
    </RGB>
  </BackColor>
  <Position>
    <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
```

```
<!--ro, req, int, x坐标-->1
</positionX>
<positionY>
<!--ro, req, int, y坐标-->1
</positionY>
<height>
<!--ro, req, int, 宽-->1
</height>
<width>
<!--ro, req, int, 高-->1
</width>
</Position>
</Wind>
</WeatherParam>
<Countdown>
<!--ro, opt, object, 倒计时素材-->
<endTime>
<!--ro, req, string, 倒计时时间点, desc:设备早期返回时间格式有误, 错为: 20200826T051500+08, 为保证XSD校验通过, 改为string类型-->20200826T051500+08
</endTime>
<template>
<!--ro, req, enum, 模板类型, subType:string, [template1#模板1,template2#模板2,openingActivity1#活动开幕1,openingActivity2#活动开幕2,commercialActivity1#商业活动1,commercialActivity2#商业活动2,collegeEntranceExam#高考,overseasTimeFormat#海外时间格式]-->template1
</template>
<timeUnit>
<!--ro, req, enum, 时间单位, subType:string, [year#年,month#月,day#日,week#周,hour#时,minute#分,second#秒]-->year
</timeUnit>
<backPicId>
<!--ro, opt, int, 背景图片素材ID-->1
</backPicId>
<TimeFontCfg>
<!--ro, opt, object, 时间字体参数-->
<fontSize>
<!--ro, req, int, 字体大小-->1
</fontSize>
<fontColor>
<!--ro, req, object, 字体颜色-->
<RGB>
<!--ro, opt, int, RGB值, desc:十进制数据表示三原色, 如16777215表示0xFFFFFF-->1
</RGB>
</FontColor>
<Position>
<!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
<positionX>
<!--ro, req, int, x坐标-->1
</positionX>
<positionY>
<!--ro, req, int, y坐标-->1
</positionY>
<height>
<!--ro, req, int, 宽-->1
</height>
<width>
<!--ro, req, int, 高-->1
</width>
</Position>
</TimeFontCfg>
<timeUnitEnabled>
<!--ro, opt, bool, 时间单位是否有效-->true
</timeUnitEnabled>
<maxTimeUnit>
<!--ro, req, enum, 最大时间单位, subType:string, [year#年,month#月,day#日,week#周,hour#时,minute#分,second#秒,week#周], desc:表示当前倒计时时的最大显示单元,例如:maxTimeUnit为月,则1年3个月,对应月份值为15-->year
</maxTimeUnit>
<ModuleInfoList>
<!--ro, opt, array, 界面模块信息列表, subType:object-->
<ModuleInfo>
<!--ro, opt, object, 界面模块信息-->
<moduleType>
<!--ro, req, enum, 模块类型, subType:string, [year#年,month#月,day#日,hour#时,minute#分,second#秒,week#周,text#文字], desc:text为时间旁边的文字,例如年下面的文字"年"-->second
</moduleType>
<enabled>
<!--ro, req, bool, 使能-->true
</enabled>
<fontSize>
<!--ro, req, int, 字体大小-->1
</fontSize>
<fontColor>
<!--ro, req, object, 字体颜色-->
<RGB>
<!--ro, opt, int, 十进制数据表示三原色-->1
</RGB>
</FontColor>
<text>
<!--ro, opt, string, 文字, desc:当moduleType为text时有效-->test
</text>
<Position>
<!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
<positionX>
<!--ro, req, int, X坐标-->1
</positionX>
<positionY>
```

```

        <!--ro, req, int, Y坐标-->1
    </positionY>
    <height>
        <!--ro, req, int, 宽-->1
    </height>
    <width>
        <!--ro, req, int, 高-->1
    </width>
    </Position>
    </ModuleInfo>
    </ModuleInfoList>
</Countdown>
<LocalInputNo>
    <!--ro, req, enum, 本地输入类型, subType:string, [VGA#VGA,HDMI#HDMI]-->VGA
</LocalInputNo>
<HyperLinkBtn>
    <!--ro, opt, object, 超链接按钮-->
    <backPicId>
        <!--ro, opt, int, 背景图片素材ID-->1
    </backPicId>
</HyperLinkBtn>
<CharactersAttribute>
    <!--ro, opt, object, 文字属性, desc:当dynamicType为character时有效-->
    <fontSize>
        <!--ro, req, int, 字体大小-->1
    </fontSize>
    <fontColor>
        <!--ro, req, object, 字体颜色-->
    <RGB>
        <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
    </RGB>
    </FontColor>
    <fontType>
        <!--ro, opt, enum, 字体类型, subType:string, [normal#常规,bold#粗体]-->normal
    </fontType>
    <backgroundType>
        <!--ro, req, enum, 背景类型, subType:string, [color#颜色,picture#图片]-->color
    </backgroundType>
    <BackColor>
        <!--ro, req, object, 背景色-->
    <RGB>
        <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
    </RGB>
    <fillEnabled>
        <!--ro, opt, bool, 背景色填充使能, desc:默认为true, 表示背景色填充-->true
    </fillEnabled>
</BackColor>
<backTransparent>
    <!--ro, opt, int, 背景透明度-->1
</backTransparent>
<alignType>
    <!--ro, req, enum, 对齐方式, subType:string, [Left#左,right#右,middle#中]-->middle
</alignType>
<verticalAlignType>
    <!--ro, req, enum, 垂直对齐方式, subType:string, [top#顶,bottom#底,verticalCenter#中]-->top
</verticalAlignType>
<characterContent>
    <!--ro, opt, string, 文字内容, range:[0,512]-->test
</characterContent>
<backPicId>
    <!--ro, opt, int, 背景图片素材ID-->1
</backPicId>
<characterStyle>
    <!--ro, req, enum, 字体效果, subType:string, [normal#常规,bold#粗体]-->normal
</characterStyle>
<characterType>
    <!--ro, req, enum, 字体类型, subType:string, [mainTitle#主标题,subTitle#副标题]-->mainTitle
</characterType>
<scrollDirection>
    <!--ro, opt, enum, 文字滚动方向, subType:string, [Left#向左,right#向右,up#向上,down#向下,turnPage#翻页]-->left
</scrollDirection>
<scrollSpeed>
    <!--ro, opt, int, 文字滚动速度, range:[1,10]-->1
</scrollSpeed>
</CharactersAttribute>
<SignIn>
    <!--ro, opt, object, 人员签到展示窗口, desc:当staticMaterialType为signIn时有效-->
    <backPicId>
        <!--ro, opt, int, 背景图片素材ID-->1
    </backPicId>
    <ShowInfo>
        <!--ro, opt, object, 展示信息-->
    <row>
        <!--ro, opt, int, 窗口内人员签到展示行数-->1
    </row>
    <column>
        <!--ro, opt, int, 窗口内人员签到展示列数-->1
    </column>
    <Position>
        <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
        <!--ro, req, int, x坐标-->1
    </positionX>
    <positionY>
        <!--ro, req, int, y坐标-->1
    </positionY>
    </Position>
    </ShowInfo>
</SignIn>

```

```
<positionY>
  <!--ro, req, int, y坐标-->1
</positionY>
<height>
  <!--ro, req, int, 宽-->1
</height>
<width>
  <!--ro, req, int, 高-->1
</width>
</Position>
<inLib>
  <!--ro, opt, object, 在库信息-->
  <nameEnabled>
    <!--ro, opt, bool, 在库人员是否展示姓名-->true
  </nameEnabled>
  <genderEnabled>
    <!--ro, opt, bool, 在库人员是否展示性别-->true
  </genderEnabled>
  <selfDefine1Enabled>
    <!--ro, opt, bool, 在库人员是否展示自定义字段1-->true
  </selfDefine1Enabled>
  <selfDefine2Enabled>
    <!--ro, opt, bool, 在库人员是否展示自定义字段2-->true
  </selfDefine2Enabled>
  <signInTimeEnabled>
    <!--ro, opt, bool, 在库人员是否展示签到时间-->true
  </signInTimeEnabled>
  <temperatureEnabled>
    <!--ro, opt, bool, 在库人员是否展示温度-->true
  </temperatureEnabled>
</inLib>
<outLib>
  <!--ro, opt, object, 不在库信息-->
  <nameEnabled>
    <!--ro, opt, bool, 未知人员是否展示姓名-->true
  </nameEnabled>
  <name>
    <!--ro, opt, string, 未知人员姓名-->test
  </name>
  <genderEnabled>
    <!--ro, opt, bool, 未知人员是否展示性别-->true
  </genderEnabled>
  <signInTimeEnabled>
    <!--ro, opt, bool, 未知人员是否展示签到时间-->true
  </signInTimeEnabled>
  <temperatureEnabled>
    <!--ro, opt, bool, 未知人员是否展示温度-->true
  </temperatureEnabled>
</outLib>
<showDuration>
  <!--ro, req, enum, 签到记录显示时间, subType:string, [always#不消失,10s#10s,20s#20s,30s#30s,no#不显示]-->always
</showDuration>
</ShowInfo>
<MainTitle>
  <!--ro, opt, object, 主标题-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <fontSize>
    <!--ro, req, int, 字体大小-->1
  </fontSize>
  <fontColor>
    <!--ro, req, object, 字体颜色-->
    <RGB>
      <!--ro, opt, int, 十进制数据表示三原色-->1
    </RGB>
  </FontColor>
  <Position>
    <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--ro, req, int, x坐标-->1
    </positionX>
    <positionY>
      <!--ro, req, int, y坐标-->1
    </positionY>
    <height>
      <!--ro, req, int, 宽-->1
    </height>
    <width>
      <!--ro, req, int, 高-->1
    </width>
  </Position>
  <text>
    <!--ro, opt, string, 标题文字-->test
  </text>
</MainTitle>
<SubTitle>
  <!--ro, opt, object, 副标题-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <fontSize>
    <!--ro, req, int, 字体大小-->1
  </fontSize>
```

```
<FontColor>
  <!--ro, req, object, 字体颜色-->
  <RGB>
    <!--ro, opt, int, 十进制数据表示三原色-->1
  </RGB>
</FontColor>
<Position>
  <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
  <positionX>
    <!--ro, req, int, x坐标-->1
  </positionX>
  <positionY>
    <!--ro, req, int, y坐标-->1
  </positionY>
  <height>
    <!--ro, req, int, 宽-->1
  </height>
  <width>
    <!--ro, req, int, 高-->1
  </width>
</Position>
<text>
  <!--ro, opt, string, 标题文字-->test
</text>
</SubTitle>
<NormalTemperature>
  <!--ro, opt, object, 正常体温信息-->
  <PeopleNumber>
    <!--ro, opt, object, 正常体温人数信息-->
    <enabled>
      <!--ro, req, bool, 使能-->true
    </enabled>
    <fontSize>
      <!--ro, req, int, 字体大小-->1
    </fontSize>
    <FontColor>
      <!--ro, req, object, 字体颜色-->
      <RGB>
        <!--ro, opt, int, 十进制数据表示三原色-->1
      </RGB>
    </FontColor>
    <Position>
      <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
      <positionX>
        <!--ro, req, int, X坐标-->1
      </positionX>
      <positionY>
        <!--ro, req, int, Y坐标-->1
      </positionY>
      <height>
        <!--ro, req, int, 宽-->1
      </height>
      <width>
        <!--ro, req, int, 高-->1
      </width>
    </Position>
    <text>
      <!--ro, opt, string, 标题文字-->test
    </text>
  </PeopleNumber>
  <Icon>
    <!--ro, opt, object, 正常体温图标-->
    <enabled>
      <!--ro, req, bool, 使能-->true
    </enabled>
    <Position>
      <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
      <positionX>
        <!--ro, req, int, X坐标-->1
      </positionX>
      <positionY>
        <!--ro, req, int, Y坐标-->1
      </positionY>
      <height>
        <!--ro, req, int, 宽-->1
      </height>
      <width>
        <!--ro, req, int, 高-->1
      </width>
    </Position>
    <materialNo>
      <!--ro, opt, int, 素材ID-->1
    </materialNo>
  </Icon>
</NormalTemperature>
<AbnormalTemperature>
  <!--ro, opt, object, 异常体温信息-->
  <PeopleNumber>
    <!--ro, opt, object, 异常体温人数信息-->
    <enabled>
      <!--ro, req, bool, 使能-->true
    </enabled>
    <fontSize>
```

```
<!--ro, req, int, 字体大小-->1
</FontSize>
<FontColor>
<!--ro, req, object, 字体颜色-->
<RGB>
<!--ro, opt, int, 十进制数据表示三原色-->1
</RGB>
</FontColor>
<Position>
<!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
<positionX>
<!--ro, req, int, X坐标-->1
</positionX>
<positionY>
<!--ro, req, int, Y坐标-->1
</positionY>
<height>
<!--ro, req, int, 宽-->1
</height>
<width>
<!--ro, req, int, 高-->1
</width>
</Position>
<text>
<!--ro, opt, string, 标题文字-->test
</text>
</PeopleNumber>
<Icon>
<!--ro, opt, object, 异常体温图标-->
<enabled>
<!--ro, req, bool, 使能-->true
</enabled>
<Position>
<!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
<positionX>
<!--ro, req, int, X坐标-->1
</positionX>
<positionY>
<!--ro, req, int, Y坐标-->1
</positionY>
<height>
<!--ro, req, int, 宽-->1
</height>
<width>
<!--ro, req, int, 高-->1
</width>
</Position>
<materialNo>
<!--ro, opt, int, 素材ID-->1
</materialNo>
</Icon>
</AbnormalTemperature>
</SignIn>
<EventParam>
<!--ro, opt, object, 事件参数-->
<EventResourceList>
<!--ro, opt, array, 事件源信息列表, subtype:object-->
<EventResource>
<!--ro, opt, object, 事件源信息-->
<ipVersion>
<!--ro, req, enum, ip地址版本, subtype:string, [v4#v4, v6#v6, dual#双]-->v4
</ipVersion>
<ipAddress>
<!--ro, opt, string, ipv4地址-->test
</ipAddress>
<ipv6Address>
<!--ro, opt, string, ipv6地址-->test
</ipv6Address>
<portNo>
<!--ro, req, int, 端口号-->1
</portNo>
<userName>
<!--ro, req, string, 用户名-->test
</userName>
<password>
<!--ro, req, string, 密码-->test
</password>
<channels>
<!--ro, opt, string, 通道号, desc:多选以逗号隔开-->1,2
</channels>
</EventResource>
</EventResourceList>
<backPicId>
<!--ro, opt, int, 背景图片素材ID-->1
</backPicId>
<Title>
<!--ro, opt, object, 标题-->
<enabled>
<!--ro, req, bool, 使能-->true
</enabled>
<fontSize>
<!--ro, req, int, 字体大小-->1
</fontSize>
<FontColor>
```

```
<!--ro, req, object, 字体颜色-->
<RGB>
  <!--ro, opt, int, 十进制数据表示三原色-->1
</RGB>
</FontColor>
<Position>
  <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
  <positionX>
    <!--ro, req, int, X坐标-->1
  </positionX>
  <positionY>
    <!--ro, req, int, Y坐标-->1
  </positionY>
  <height>
    <!--ro, req, int, 宽-->1
  </height>
  <width>
    <!--ro, req, int, 高-->1
  </width>
</Position>
<text>
  <!--ro, opt, string, 标题文字-->test
</text>
</Title>
<PeopleCounting>
  <!--ro, opt, object, 人数统计提醒-->
  <maxPeopleNumber>
    <!--ro, opt, int, 最大可进入人数-->1
  </maxPeopleNumber>
  <unavailableBackPicId>
    <!--ro, opt, int, 人员不能进入时控件背景图片id-->1
  </unavailableBackPicId>
  <supportChangeByPeople>
    <!--ro, req, enum, 支持根据人数修改样式的字段, subType:string,
[prompt#prompt,peopleIcon#peopleIcon,availablePeopleNumber#availablePeopleNumber]-->prompt
  </supportChangeByPeople>
  <Prompt>
    <!--ro, opt, object, 提示信息-->
    <enabled>
      <!--ro, req, bool, 使能-->true
    </enabled>
    <fontSize>
      <!--ro, req, int, 字体大小-->1
    </fontSize>
    <Position>
      <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
      <positionX>
        <!--ro, req, int, X坐标-->1
      </positionX>
      <positionY>
        <!--ro, req, int, Y坐标-->1
      </positionY>
      <height>
        <!--ro, req, int, 宽-->1
      </height>
      <width>
        <!--ro, req, int, 高-->1
      </width>
    </Position>
    <text>
      <!--ro, opt, string, 标题文字-->test
    </text>
    <FontColor>
      <!--ro, opt, object, 人员可以进入时的字体颜色-->
      <RGB>
        <!--ro, opt, int, 十进制数据表示三原色-->1
      </RGB>
    </FontColor>
    <unavailableText>
      <!--ro, opt, string, 人员不能进入时的标题文字-->test
    </unavailableText>
    <UnavailableFontColor>
      <!--ro, opt, object, 人员不能进入时的字体颜色-->
      <RGB>
        <!--ro, opt, int, 十进制数据表示三原色-->1
      </RGB>
    </UnavailableFontColor>
  </Prompt>
  <PeopleIcon>
    <!--ro, opt, object, 行人通行图标-->
    <enabled>
      <!--ro, req, bool, 使能-->true
    </enabled>
    <Position>
      <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
      <positionX>
        <!--ro, req, int, X坐标-->1
      </positionX>
      <positionY>
        <!--ro, req, int, Y坐标-->1
      </positionY>
      <height>
        <!--ro, req, int, 宽-->1
      </height>
```

```
</height>
<width>
  <!--ro, req, int, 高-->1
</width>
</Position>
<materialNo>
  <!--ro, opt, int, 素材ID-->1
</materialNo>
<PeopleIcon>
<InsidePeopleTitle>
  <!--ro, opt, object, 已经进入人数标题-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <fontSize>
    <!--ro, req, int, 字体大小-->1
  </fontSize>
  <fontColor>
    <!--ro, req, object, 字体颜色-->
    <RGB>
      <!--ro, opt, int, 十进制数据表示三原色-->1
    </RGB>
  </FontColor>
  <Position>
    <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--ro, req, int, X坐标-->1
    </positionX>
    <positionY>
      <!--ro, req, int, Y坐标-->1
    </positionY>
    <height>
      <!--ro, req, int, 宽-->1
    </height>
    <width>
      <!--ro, req, int, 高-->1
    </width>
  </Position>
  <text>
    <!--ro, opt, string, 标题文字-->test
  </text>
</InsidePeopleTitle>
<InsidePeopleNumbers>
  <!--ro, opt, object, 已经进入人数-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <fontSize>
    <!--ro, req, int, 字体大小-->1
  </fontSize>
  <fontColor>
    <!--ro, req, object, 字体颜色-->
    <RGB>
      <!--ro, opt, int, 十进制数据表示三原色-->1
    </RGB>
  </FontColor>
  <Position>
    <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--ro, req, int, X坐标-->1
    </positionX>
    <positionY>
      <!--ro, req, int, Y坐标-->1
    </positionY>
    <height>
      <!--ro, req, int, 宽-->1
    </height>
    <width>
      <!--ro, req, int, 高-->1
    </width>
  </Position>
</InsidePeopleNumber>
<AvailablePeopleTitle>
  <!--ro, opt, object, 还可以进入人数标题-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <fontSize>
    <!--ro, req, int, 字体大小-->1
  </fontSize>
  <fontColor>
    <!--ro, req, object, 字体颜色-->
    <RGB>
      <!--ro, opt, int, 十进制数据表示三原色-->1
    </RGB>
  </FontColor>
  <Position>
    <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--ro, req, int, X坐标-->1
    </positionX>
    <positionY>
      <!--ro, req, int, Y坐标-->1
    </positionY>
```

```
</Position>
<height>
  <!--ro, req, int, 宽-->1
</height>
<width>
  <!--ro, req, int, 高-->1
</width>
</Position>
<text>
  <!--ro, opt, string, 标题文字-->test
</text>
</AvailablePeopleTitle>
<AvailablePeopleNumber>
  <!--ro, opt, object, 还可以进入人数-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <fontSize>
    <!--ro, req, int, 字体大小-->1
  </fontSize>
  <Position>
    <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--ro, req, int, X坐标-->1
    </positionX>
    <positionY>
      <!--ro, req, int, Y坐标-->1
    </positionY>
    <height>
      <!--ro, req, int, 宽-->1
    </height>
    <width>
      <!--ro, req, int, 高-->1
    </width>
  </Position>
  <FontColor>
    <!--ro, opt, object, 人员可以进入时的字体颜色-->
    <RGB>
      <!--ro, opt, int, 十进制数据表示三原色-->1
    </RGB>
  </FontColor>
  <UnavailableFontColor>
    <!--ro, opt, object, 人员不能进入时的字体颜色-->
    <RGB>
      <!--ro, opt, int, 十进制数据表示三原色-->1
    </RGB>
  </UnavailableFontColor>
</AvailablePeopleNumber>
<UnavailablePeopleIcon>
  <!--ro, opt, object, 人员不能进入时的行人图标-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <Position>
    <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--ro, req, int, X坐标-->1
    </positionX>
    <positionY>
      <!--ro, req, int, Y坐标-->1
    </positionY>
    <height>
      <!--ro, req, int, 宽-->1
    </height>
    <width>
      <!--ro, req, int, 高-->1
    </width>
  </Position>
  <materialNo>
    <!--ro, opt, int, 素材ID-->1
  </materialNo>
</UnavailablePeopleIcon>
<AudioAlarm>
  <!--ro, opt, object, 音频报警信息-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <materialNo>
    <!--ro, opt, int, 音频素材ID-->1
  </materialNo>
</AudioAlarm>
</PeopleCounting>
<Mask>
  <!--ro, opt, object, 未戴口罩提醒-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <fontSize>
    <!--ro, req, int, 字体大小-->1
  </fontSize>
  <FontColor>
    <!--ro, req, object, 字体颜色-->
    <RGB>
      <!--ro, opt, int, 十进制数据表示三原色-->1
```

```
</RGB>
</FontColor>
<BackColor>
  <!--ro, req, object, 背景颜色-->
  <RGB>
    <!--ro, opt, int, 十进制数据表示三原色-->1
  </RGB>
</BackColor>
<Position>
  <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
  <positionX>
    <!--ro, req, int, X坐标-->1
  </positionX>
  <positionY>
    <!--ro, req, int, Y坐标-->1
  </positionY>
  <height>
    <!--ro, req, int, 宽-->1
  </height>
  <width>
    <!--ro, req, int, 高-->1
  </width>
</Position>
<text>
  <!--ro, opt, string, 标题文字-->test
</text>
<MaskIcon>
  <!--ro, opt, object, 未戴口罩图标-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <Position>
    <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--ro, req, int, X坐标-->1
    </positionX>
    <positionY>
      <!--ro, req, int, Y坐标-->1
    </positionY>
    <height>
      <!--ro, req, int, 宽-->1
    </height>
    <width>
      <!--ro, req, int, 高-->1
    </width>
  </Position>
  <materialNo>
    <!--ro, opt, int, 素材ID-->1
  </materialNo>
</MaskIcon>
<AudioAlarm>
  <!--ro, opt, object, 音频报警信息-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <materialNo>
    <!--ro, opt, int, 音频素材ID-->1
  </materialNo>
</AudioAlarm>
</Mask>
<Temperature>
  <!--ro, opt, object, 温度异常提醒-->
  <Prompt>
    <!--ro, opt, object, 提示信息-->
    <enabled>
      <!--ro, req, bool, 使能-->true
    </enabled>
  </Prompt>
  <fontSize>
    <!--ro, req, int, 字体大小-->1
  </fontSize>
  <FontColor>
    <!--ro, req, object, 字体颜色-->
    <RGB>
      <!--ro, opt, int, 十进制数据表示三原色-->1
    </RGB>
  </FontColor>
  <BackColor>
    <!--ro, req, object, 背景颜色-->
    <RGB>
      <!--ro, opt, int, 十进制数据表示三原色-->1
    </RGB>
  </BackColor>
  <Position>
    <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--ro, req, int, X坐标-->1
    </positionX>
    <positionY>
      <!--ro, req, int, Y坐标-->1
    </positionY>
    <height>
      <!--ro, req, int, 宽-->1
    </height>
  </Position>
```

```
<width>
  <!--ro, req, int, 高-->1
</width>
</Position>
<text>
  <!--ro, opt, string, 标题文字-->test
</text>
</Prompt>
<TemperatureIcon>
  <!--ro, opt, object, 温度异常图标-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <Position>
    <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--ro, req, int, X坐标-->1
    </positionX>
    <positionY>
      <!--ro, req, int, Y坐标-->1
    </positionY>
    <height>
      <!--ro, req, int, 宽-->1
    </height>
    <width>
      <!--ro, req, int, 高-->1
    </width>
  </Position>
  <materialNo>
    <!--ro, opt, int, 素材ID-->1
  </materialNo>
</TemperatureIcon>
<AudioAlarm>
  <!--ro, opt, object, 音频报警信息-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <materialNo>
    <!--ro, opt, int, 音频素材ID-->1
  </materialNo>
</AudioAlarm>
<abnormalHighTemperature>
  <!--ro, opt, float, 异常高温值, unit:℃, unitType:温度, desc:保留一位小数-->37.5
</abnormalHighTemperature>
</Temperature>
<WearMask>
  <!--ro, opt, object, 已戴口罩提醒-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <fontSize>
    <!--ro, req, int, 字体大小-->1
  </fontSize>
  <FontColor>
    <!--ro, req, object, 字体颜色-->
    <RGB>
      <!--ro, opt, int, 十进制数据表示三原色-->1
    </RGB>
  </FontColor>
  <BackColor>
    <!--ro, req, object, 背景颜色-->
    <RGB>
      <!--ro, opt, int, 十进制数据表示三原色-->1
    </RGB>
  </BackColor>
  <Position>
    <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--ro, req, int, X坐标-->1
    </positionX>
    <positionY>
      <!--ro, req, int, Y坐标-->1
    </positionY>
    <height>
      <!--ro, req, int, 宽-->1
    </height>
    <width>
      <!--ro, req, int, 高-->1
    </width>
  </Position>
  <text>
    <!--ro, opt, string, 标题文字-->test
  </text>
  <MaskIcon>
    <!--ro, opt, object, 已戴口罩图标-->
    <enabled>
      <!--ro, req, bool, 使能-->true
    </enabled>
    <Position>
      <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
      <positionX>
        <!--ro, req, int, X坐标-->1
      </positionX>
      <positionY>
```

```
<!--ro, req, int, Y坐标-->1
</positionY>
<!--ro, req, int, 宽-->1
</height>
<!--ro, req, int, 高-->1
</width>
</Position>
<materialNo>
<!--ro, opt, int, 素材ID-->1
</materialNo>
</MaskIcon>
</WearMask>
<NormalTemperature>
<!--ro, opt, object, 温度正常提醒-->
<Prompt>
<!--ro, opt, object, 提示信息-->
<enabled>
<!--ro, req, bool, 使能-->true
</enabled>
<fontSize>
<!--ro, req, int, 字体大小-->1
</fontSize>
<fontColor>
<!--ro, req, object, 字体颜色-->
<RGB>
<!--ro, opt, int, 十进制数据表示三原色-->1
</RGB>
</FontColor>
<BackColor>
<!--ro, req, object, 背景颜色-->
<RGB>
<!--ro, opt, int, 十进制数据表示三原色-->1
</RGB>
</BackColor>
<Position>
<!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
<positionX>
<!--ro, req, int, X坐标-->1
</positionX>
<positionY>
<!--ro, req, int, Y坐标-->1
</positionY>
<height>
<!--ro, req, int, 宽-->1
</height>
<width>
<!--ro, req, int, 高-->1
</width>
</Position>
<text>
<!--ro, opt, string, 标题文字-->test
</text>
</Prompt>
<TemperatureIcon>
<!--ro, opt, object, 温度正常图标-->
<enabled>
<!--ro, req, bool, 使能-->true
</enabled>
<Position>
<!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
<positionX>
<!--ro, req, int, X坐标-->1
</positionX>
<positionY>
<!--ro, req, int, Y坐标-->1
</positionY>
<height>
<!--ro, req, int, 宽-->1
</height>
<width>
<!--ro, req, int, 高-->1
</width>
</Position>
<materialNo>
<!--ro, opt, int, 素材ID-->1
</materialNo>
</TemperatureIcon>
</NormalTemperature>
<NormalStatusAudioAlarm>
<!--ro, opt, object, 正常状态下音频报警信息, desc:人员在检测时未发现上面的人数超限、未戴口罩、体温异常等事件时设备进行播报-->
<enabled>
<!--ro, req, bool, 使能-->true
</enabled>
<materialNo>
<!--ro, opt, int, 音频素材ID-->1
</materialNo>
</NormalStatusAudioAlarm>
</EventParam>
<AlarmDataSourceProperty>
<!--ro, opt, object, 数据源信息, desc:该字段能力集缺少, 仅信发组件和设备使用-->
<PictureProperty>
```

```
<!--ro, opt, object, 图片属性-->
<PictureSwitchEffect>
  <!--ro, opt, enum, 图片切换效果, subType:string, [LeftInRightOut#左进右出,rightInLeftOut#右进左出]-->LeftInRightOut
</PictureSwitchEffect>
<PictureScaleType>
  <!--ro, opt, enum, 图片缩放类型, subType:string, [fitCenter#拉伸填充,centerCrop#截取填充,fixXY#适应宽高]-->fitCenter
</PictureScaleType>
</PictureProperty>
<AlarmDataSource>
  <!--ro, opt, object, 数据源-->
  <MaterialNo>
    <!--ro, opt, int, 素材编号-->1
  </MaterialNo>
  <Name>
    <!--ro, opt, string, 素材名称-->test
  </Name>
  <Address>
    <!--ro, opt, string, 素材URL地址-->test
  </Address>
  <WebProperty>
    <!--ro, opt, object, 网页属性-->
    <webType>
      <!--ro, opt, enum, 网页类型, subType:string, [desktop#桌面版,mobile#移动版], desc:默认按照设备的系统类型（Windows或Android）来显示网页--
>desktop
    </webType>
  </WebProperty>
</AlarmDataSource>
</AlarmDataSourceProperty>
<ConferenceParam>
  <!--ro, opt, object, 会议信息, desc:当页面用于会议界面时有效-->
  <backPicId>
    <!--ro, opt, int, 背景图片素材ID-->1
  </backPicId>
  <Title>
    <!--ro, opt, object, 会议标题（主题）， desc:1、会议标题（主题）的文字内容通过场地预约配置协议下发,对应/ISAPI/Publish/Signage/placeReservation?
format=json中的programID字段。2、可通过会议计划（/ISAPI/InteractivePanel/Meeting/plan?format=json）中的meetingTopic字段下发-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <fontSize>
    <!--ro, req, int, 字体大小-->1
  </fontSize>
  <fontColor>
    <!--ro, req, object, 字体颜色-->
    <RGB>
      <!--ro, opt, int, 十进制数据表示三原色-->1
    </RGB>
  </fontColor>
  <BackColor>
    <!--ro, req, object, 背景颜色-->
    <RGB>
      <!--ro, opt, int, 十进制数据表示三原色-->1
    </RGB>
  </BackColor>
  <Position>
    <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--ro, req, int, X坐标-->1
    </positionX>
    <positionY>
      <!--ro, req, int, Y坐标-->1
    </positionY>
    <height>
      <!--ro, req, int, 宽-->1
    </height>
    <width>
      <!--ro, req, int, 高-->1
    </width>
  </Position>
</Title>
<VoiceBroadcast>
  <!--ro, opt, object, 语音播报-->
  <enabled>
    <!--ro, opt, bool, 是否启用语音播报-->true
  </enabled>
  <broadcastSpeed>
    <!--ro, opt, enum, 播报语速, subType:string, [sLower#慢,sLow#较慢,medium#中速,fast#较快,faster#快]-->sLower
  </broadcastSpeed>
  <inLibVoice>
    <!--ro, opt, string, 在库人员语音播报内容-->test
  </inLibVoice>
  <outLibVoice>
    <!--ro, opt, string, 未知人员语音播报内容-->test
  </outLibVoice>
</VoiceBroadcast>
<WelcomeWord>
  <!--ro, opt, object, 欢迎词-->
  <enabled>
    <!--ro, opt, bool, 是否启用欢迎词-->true
  </enabled>
  <inLibWord>
    <!--ro, opt, string, 在库人员欢迎词内容, range:[1,128]-->test
  </inLibWord>
  <outLibWord>
```

```
<!--ro, opt, string, 未知人员欢迎词内容, range:[1,128]-->test
</outLibWord>
</WelcomeWord>
<ShowInfo>
  <!--ro, opt, object, 展示信息-->
  <row>
    <!--ro, opt, int, 窗口内人员签到展示行数-->1
  </row>
  <column>
    <!--ro, opt, int, 窗口内人员签到展示列数-->1
  </column>
  <Position>
    <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--ro, req, int, x坐标-->1
    </positionX>
    <positionY>
      <!--ro, req, int, y坐标-->1
    </positionY>
    <height>
      <!--ro, req, int, 宽-->1
    </height>
    <width>
      <!--ro, req, int, 高-->1
    </width>
  </Position>
  <inLib>
    <!--ro, opt, object, 在库信息-->
    <nameEnabled>
      <!--ro, opt, bool, 在库人员是否展示姓名-->true
    </nameEnabled>
  </inLib>
  <defaultShowInfoEnabled>
    <!--ro, opt, bool, 是否采用人员默认展示信息, desc:下发该字段, 且为true, 表示人员展示信息采用默认值。坐标信息等不用下发-->true
  </defaultShowInfoEnabled>
</ShowInfo>
<ModuleInfoList>
  <!--ro, opt, array, 界面模块信息列表, subType:object-->
  <ModuleInfo>
    <!--ro, opt, object, 界面模块信息-->
    <moduleType>
      <!--ro, req, enum, 模块类型, subType:string, [conferenceTime#会议时间(对应timeRange), conferenceSite#会议地点(对应会议室名称
meetingName), conferenceStatus#会议状态(对应meetingStatus), signInPersonNum#已签到人数, totalPersonNum#应到人数, unSignInPersonNum#未签到人
数, signInAndTotalPersonNum#已签到/应到人数, QRCode#会议室二维码, meetingHostName#主持人, employeeNo#参会人, customKey#自定义字段, nextMeeting#当天下一场会议信
息]-->conferenceTime
    </moduleType>
    <enabled>
      <!--ro, req, bool, 使能-->true
    </enabled>
    <fontSize>
      <!--ro, req, int, 字体大小-->1
    </fontSize>
    <fontColor>
      <!--ro, req, object, 字体颜色-->
      <RGB>
        <!--ro, opt, int, 十进制数据表示三原色-->1
      </RGB>
    </fontColor>
    <BackColor>
      <!--ro, opt, object, 背景颜色-->
      <RGB>
        <!--ro, opt, int, 十进制数据表示三原色-->1
      </RGB>
    </BackColor>
    <backPicId>
      <!--ro, opt, int, 背景图片素材ID, desc:背景图和背景颜色同时仅支持一种, 两者均下发, 设备优先解析背景图字段-->1
    </backPicId>
    <Position>
      <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
      <positionX>
        <!--ro, req, int, X坐标-->1
      </positionX>
      <positionY>
        <!--ro, req, int, Y坐标-->1
      </positionY>
      <height>
        <!--ro, req, int, 宽-->1
      </height>
      <width>
        <!--ro, req, int, 高-->1
      </width>
    </Position>
    <CustomKey>
      <!--ro, opt, object, 自定义字段-->
      <key>
        <!--ro, opt, string, 自定义字段-->test
      </key>
      <associateProtocolURL>
        <!--ro, opt, string, 关联协议URL-->test
      </associateProtocolURL>
    </CustomKey>
  </ModuleInfo>
</ModuleInfoList>
```

```

<conferenceStatus>
  <!--ro, opt, enum, 会议状态, subType:string, [idle#空闲,signing#签到中,meeting#开会中], desc:根据不同的会议状态, 配置不同的会议界面-->idle
</conferenceStatus>
</ConferenceParam>
<CallBtn>
  <!--ro, opt, object, 呼叫按钮窗口-->
  <backPicId>
    <!--ro, opt, int, 背景图片素材ID-->1
  </backPicId>
</CallBtn>
<OpenDoorBtn>
  <!--ro, opt, object, 开门按钮窗口-->
  <backPicId>
    <!--ro, opt, int, 背景图片素材ID-->1
  </backPicId>
</OpenDoorBtn>
<QRCodeBtn>
  <!--ro, opt, object, 二维码按钮窗口-->
  <backPicId>
    <!--ro, opt, int, 背景图片素材ID-->1
  </backPicId>
</QRCodeBtn>
<AuthenticationSuccess>
  <!--ro, opt, object, 认证成功窗口-->
  <BackColor>
    <!--ro, opt, object, 背景颜色-->
    <RGB>
      <!--ro, opt, int, 十进制数据表示三原色-->1
    </RGB>
  </BackColor>
  <backPicId>
    <!--ro, opt, int, 背景图片素材ID, desc:背景图和背景颜色同时仅支持一种,两者均下发,设备优先解析背景图字段-->1
  </backPicId>
  <SubWindowList>
    <!--ro, opt, object, 子窗口列表-->
    <SubWindow>
      <!--ro, opt, object, 子窗口-->
      <subWindowType>
        <!--ro, req, enum, 子窗口类型, subType:string, [temperature#温度,facePic#人脸图片,name#姓名,employeeNo#工号,group#组织/部门,tips#提示信息]-->temperature
      </subWindowType>
      <enabled>
        <!--ro, req, bool, 使能-->true
      </enabled>
      <fontSize>
        <!--ro, opt, int, 字体大小, unit:px, unitType:图像显示-->1
      </fontSize>
      <fontColor>
        <!--ro, opt, int, 字体颜色-->1
      </fontColor>
      <fontStyle>
        <!--ro, opt, string, 字体风格, desc:bold-加粗, italic-倾斜, underline-下划线, strikethrough-删除线-->bold,italic,underline,strikeout
      </fontStyle>
      <Position>
        <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
        <positionX>
          <!--ro, req, int, x坐标-->1
        </positionX>
        <positionY>
          <!--ro, req, int, y坐标-->1
        </positionY>
        <height>
          <!--ro, req, int, 宽-->1
        </height>
        <width>
          <!--ro, req, int, 高-->1
        </width>
      </Position>
    </SubWindow>
  </SubWindowList>
</AuthenticationSuccess>
<AuthenticationFail>
  <!--ro, opt, object, 认证失败窗口-->
  <BackColor>
    <!--ro, opt, object, 背景颜色-->
    <RGB>
      <!--ro, opt, int, 十进制数据表示三原色-->1
    </RGB>
  </BackColor>
  <backPicId>
    <!--ro, opt, int, 背景图片素材ID, desc:背景图和背景颜色同时仅支持一种,两者均下发,设备优先解析背景图字段-->1
  </backPicId>
  <SubWindowList>
    <!--ro, opt, object, 子窗口列表-->
    <SubWindow>
      <!--ro, opt, object, 子窗口-->
      <subWindowType>
        <!--ro, req, enum, 子窗口类型, subType:string, [temperature#温度,facePic#人脸图片,name#姓名,employeeNo#工号,group#组织/部门,tips#提示信息]-->temperature
      </subWindowType>
      <enabled>
        <!--ro, req, bool, 使能-->true
      </enabled>
      <fontSize>

```

```

</fontSize>
<!--ro, opt, int, 字体大小, unit:px, unitType:图像显示-->1
</fontSize>
<fontColor>
<!--ro, opt, int, 字体颜色-->1
</fontColor>
<fontStyle>
<!--ro, opt, string, 字体风格, desc:bold-加粗, italic-倾斜, underline-下划线, strikethrough-删除线-->bold,italic,underline,strikethrough
</fontStyle>
<Position>
<!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
<positionX>
<!--ro, req, int, x坐标-->1
</positionX>
<positionY>
<!--ro, req, int, y坐标-->1
</positionY>
<height>
<!--ro, req, int, 宽-->1
</height>
<width>
<!--ro, req, int, 高-->1
</width>
</Position>
<facePicID>
<!--ro, opt, int, 默认人脸图片ID, step:1, dep:and,
{$.PageList.Page.WindowsList[*].Windows.AuthenticationFail.SubWindowList.SubWindow.subWindowType,eq,facePic}-->1
</facePicID>
</SubWindow>
</SubWindowList>
</AuthenticationFail>
<StatusBar>
<!--ro, opt, object, 状态栏, desc:当素材类型为状态栏时有效, 且必填-->
<backPicId>
<!--ro, opt, int, 背景图片素材ID, desc:背景图和背景颜色同时仅支持一种, 两者均下发, 设备优先解析背景图字段-->1
</backPicId>
<statusBarType>
<!--ro, req, enum, 状态栏类型, subType:string, [fixed#固定的], desc:当为固定类型时, 窗口位置不生效, 设备将状态栏固定在顶部显示-->fixed
</statusBarType>
<SubStatusBarType>
<!--ro, opt, object, 子状态列表-->
<SubStatus>
<!--ro, opt, object, 子状态-->
<statusType>
<!--ro, req, enum, 子状态, subType:string, [dial#拨号状态(3G4G等),wifi#wifi状态,EZVIZ#萤石状态,wiredNetwork#有线网络状态,deploy#布防状态,temperature#测温]-->dial
</statusType>
<enabled>
<!--ro, req, bool, 是否显示-->true
</enabled>
</SubStatus>
</SubStatusBarType>
</StatusBar>
</Windows>
</WindowsList>
</Page>
</PageList>
<programSize>
<!--opt, int, 节目大小, unit:Byte, unitType:信息量-->1
</programSize>
<programLength>
<!--opt, int, 节目时长, unit:s, unitType:时间-->1
</programLength>
<coordinateType>
<!--opt, enum, 节目中坐标基准值类型, subType:string, [uniformCoordinate#统一坐标(实际的分辨率坐标按基准坐标等比例进行转换),resolutionCoordinate#实际分辨率坐标]-->uniformCoordinate
</coordinateType>
<screenType>
<!--opt, enum, 屏幕类型, subType:string, [first#主屏,second#辅屏], desc:当programType为doubleSide表示创建双目屏节目时生效-->first
</screenType>
<audioEnabled>
<!--opt, bool, 节目音频使能, desc:当programType为doubleSide表示创建双目屏节目时生效,用来防止主屏声音和辅屏声音混淆,主辅屏节目的音频只开一个-->true
</audioEnabled>
<PageIdList>
<!--opt, array, 页面id列表, subType:object-->
<PageId>
<!--opt, object, 页面id-->
<id>
<!--req, int, 页面ID-->1
</id>
</PageId>
</PageIdList>
<programUUID>
<!--ro, opt, string, 节目UUID, range:[1,32], desc:平台下发-->test
</programUUID>
</Program>

```

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 响应消息, attr:version{ro, req, string, 版本}-->
  <requestURL>
    <!--ro, req, string, 请求的URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, 状态码, subType:int, [0#OK,1#OK,2#Device Busy,3#Device Error,4#Invalid Operation,5#Invalid XML Format,6#Invalid XML Content,7#Reboot Required]-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, 状态信息, subType:string, [OK#成功,Device Busy#设备忙,Device Error#设备异常,Invalid Operation#无效的操作,Invalid XML Format#无效的XML格式,Invalid XML Content#无效的XML内容,Reboot#设备重启]-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, 详细错误码英文描述, desc:详细错误码的英文描述-->OK
  </subStatusCode>
</ResponseStatus>
```

69.66.7 获取单个节目参数

Request URL

GET /ISAPI/Publish/ProgramMgr/program/<programID>

查询参数

参数名称	参数类型	描述
programID	string	节目ID

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<Program xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 节目信息, attr:version{req, string, 协议版本}-->
  <id>
    <!--ro, req, int, 节目索引-->1
  </id>
  <programName>
    <!--ro, req, string, 节目名称-->test
  </programName>
  <programRemarks>
    <!--ro, req, string, 节目描述-->test
  </programRemarks>
  <shareProperty>
    <!--ro, opt, enum, 共享属性, subType:string, [public#公有,private#私有]-->public
  </shareProperty>
  <approveState>
    <!--ro, opt, enum, 审核状态标志, subType:string, [approved#审核通过,notPass#未通过,notApprove#未审核]-->approved
  </approveState>
  <approveRemarks>
    <!--ro, opt, string, 审核意见-->test
  </approveRemarks>
  <programType>
    <!--ro, opt, enum, 节目类型, subType:string, [normal#普通,decode#解码,touch#触摸,decodeTouch#解码触摸,doubleSide#双面屏,conference#会议,screensaver#图片/屏保(门禁),character#欢迎词(门禁),mainInterface#主界面(门禁),txt#文本节目(门禁),advertise#广告], desc:当节目类型为conference#会议,screensaver#图片/屏保(门禁),character#欢迎词(门禁),mainInterface#主界面(门禁),txt#文本节目(门禁),advertise#广告时,该节目不依赖与日程,直接单独下发该节目后,在设备上展示该节目下的页面信息.-->normal
  </programType>
  <orgNo>
    <!--ro, opt, int, 所属组织编号-->1
  </orgNo>
  <Resolution>
    <!--ro, req, object, 分辨率-->
    <resolutionName>
      <!--ro, opt, string, 分辨率名称-->test
    </resolutionName>
    <imageWidth>
      <!--ro, req, int, 分辨率宽-->1
    </imageWidth>
    <imageHeight>
      <!--ro, req, int, 分辨率高-->1
    </imageHeight>
  </Resolution>
  <PageList>
    <!--ro, opt, object, 页面列表-->
    <Page>
      <!--ro, req, object, 页面-->
```

```
<id>
  <!--ro, req, int, 索引-->1
</id>
<PageBasicInfo>
  <!--ro, req, object, 页面基本信息-->
  <pageName>
    <!--ro, opt, string, 页面名称-->test
  </pageName>
  <BackgroundColor>
    <!--ro, opt, object, 背景颜色-->
    <RGB>
      <!--ro, opt, int, RGB值, desc: 十进制数据表示三原色, 如16777215表示0xFFFFFF-->1
    </RGB>
  </BackgroundColor>
  <playDurationMode>
    <!--ro, opt, enum, 页面播放时间模式, subType:string, [selfDefine#自定义,auto#自动,loop#循环], desc: 自定义时playDuration节点有效; 自动时根据播放内容的时
    间*playCount计算; 循环时表示一直循环-->selfDefine
  </playDurationMode>
  <playDuration>
    <!--ro, opt, int, 页面播放时长, unit:s, unitType:时间-->1
  </playDuration>
  <playCount>
    <!--ro, opt, int, 页面播放次数-->1
  </playCount>
  <switchDuration>
    <!--ro, opt, int, 切换间隔, unit:s, unitType:时间-->1
  </switchDuration>
  <switchEffect>
    <!--ro, opt, enum, 切换效果, subType:string, [none#无,random#随机,boxShrink#盒状收缩,boxSpread#盒状展开,cycleShrink#圆形收缩,cycleSpread#圆形展
    开,eraseUp#向上擦除,eraseDown#向下擦除,eraseLeft#向左擦除,eraseRight#向右擦除,verticalShelter#垂直屏蔽,horizontalShelter#水平屏蔽,verticalChessboard#纵向棋
    盘式,horizontalChessboard#横向棋盘式,dissolve#随机溶解,leftRightToCenter#左右向中间缩进,ceterToLeftRight#中央向左右扩展,upDownToCenter#上下向中央缩
    进,centerToUpDown#中央向上下扩展,drawOutLeftDown#从左下抽出,drawOutLeftUp#从左上抽出,drawOutRightDown#从右下抽出,drawOutRightUp#从右上抽出,verticalLine#随机
    垂直线条,horizontalLine#随机垂直线条]-->none
  </switchEffect>
  <backgroundPic>
    <!--ro, opt, int, 背景图片素材ID-->1
  </backgroundPic>
  <SignInCfg>
    <!--ro, opt, object, 签到信息, desc: 当页面用于签到或考勤界面时有效-->
    <VoiceBroadcast>
      <!--ro, opt, object, 语音播报-->
      <enabled>
        <!--ro, opt, bool, 是否启用语音播报-->true
      </enabled>
      <broadcastSpeed>
        <!--ro, opt, enum, 播报语速, subType:string, [slower#慢,slow#较慢,medium#中速,fast#较快,faster#快]-->slower
      </broadcastSpeed>
      <inLibVoice>
        <!--ro, opt, string, 在库人员正常体温语音播报内容-->test
      </inLibVoice>
      <outLibVoice>
        <!--ro, opt, string, 未知人员正常体温语音播报内容-->test
      </outLibVoice>
      <inLibAbnormalTemperatureVoice>
        <!--ro, opt, string, 在库人员异常温度语音播报内容-->test
      </inLibAbnormalTemperatureVoice>
      <outLibAbnormalTemperatureVoice>
        <!--ro, opt, string, 未知人员异常温度语音播报内容-->test
      </outLibAbnormalTemperatureVoice>
    </VoiceBroadcast>
    <WelcomeWord>
      <!--ro, opt, object, 欢迎词-->
      <enabled>
        <!--ro, opt, bool, 是否启用欢迎词-->true
      </enabled>
      <inLibWord>
        <!--ro, opt, string, 在库人员欢迎词内容-->test
      </inLibWord>
      <outLibWord>
        <!--ro, opt, string, 未知人员欢迎词内容-->test
      </outLibWord>
    </WelcomeWord>
    <SignCartoon>
      <!--ro, opt, object, 签到动画-->
      <signCartoonSpeed>
        <!--ro, opt, enum, 动画速度, subType:string, [fast#快,medium#终端,slow#慢]-->fast
      </signCartoonSpeed>
    </SignCartoon>
    <backgroundSignEnabled>
      <!--ro, opt, bool, 是否启用后台签到-->true
    </backgroundSignEnabled>
    <TemperatureDetection>
      <!--ro, opt, object, 温度监测参数-->
      <enabled>
        <!--ro, req, bool, 是否启用-->true
      </enabled>
      <threshold>
        <!--ro, req, float, 正常温度值-->0.000
      </threshold>
    </TemperatureDetection>
  </SignInCfg>
</PageBasicInfo>
<characterMode>
  <!--ro, opt, enum, 页面中的欢迎词模式, subType:string, [mode1#模式1,mode2#模式2,mode3#模式3], desc: 门禁增加, 门禁欢迎词节日位置固定, 只在左二轴模式
  下有效-->mode1
</characterMode>
```

```
->model1
</characterMode>
<windowsList>
  <!--ro, opt, array, 窗口信息列表, subType:object-->
  <windows>
    <!--ro, opt, object, 窗口信息-->
    <id>
      <!--ro, req, int, 内容编号-->1
    </id>
    <Position>
      <!--ro, opt, object, 内容位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
      <positionX>
        <!--ro, req, int, x坐标-->1
      </positionX>
      <positionY>
        <!--ro, req, int, y坐标-->1
      </positionY>
      <height>
        <!--ro, req, int, 宽-->1
      </height>
      <width>
        <!--ro, req, int, 高-->1
      </width>
    </Position>
    <CornerRadius>
      <!--ro, opt, object, 圆角配置, desc:目前仅对picture图片窗口和character文字窗口有效-->
      <TopLeft>
        <!--ro, opt, int, 左上角半径-->1
      </TopLeft>
      <TopRight>
        <!--ro, opt, int, 右上角半径-->1
      </TopRight>
      <BottomLeft>
        <!--ro, opt, int, 左下角半径-->1
      </BottomLeft>
      <BottomRight>
        <!--ro, opt, int, 右下角半径-->1
      </BottomRight>
    </CornerRadius>
    <layerNo>
      <!--ro, req, int, 图层编号-->1
    </layerNo>
    <WinMaterialInfo>
      <!--ro, opt, object, 窗口素材信息-->
      <materialType>
        <!--ro, req, enum, 素材类型, subType:string, [static#静态,dynamic#动态,other#其他]-->static
      </materialType>
      <staticMaterialType>
        <!--ro, opt, enum, 静态素材类型, subType:string, [picture#图片,flash#flash,audio#音频,video#视频,document#文本,ppt#ppt,doc#文档,excel#表格,pdf#pdf,web#网页,app#app,signIn#签到,conference#会议]-->picture
      </staticMaterialType>
      <dynamicType>
        <!--ro, opt, enum, 动态素材类型, subType:string,
[web#web,socket#socket,rss#rss,realStream#realStream,call#call,dynamicPic#dynamicPic,capturePic#capturePic,character#character]-->web
      </dynamicType>
      <otherType>
        <!--ro, opt, enum, 其他素材类型, subType:string, [clock#时钟,weather#天气,countdown#倒计时,localInput#本地输入,hyperLinkBtn#超链接按钮,event#事件,callBtn#呼叫按钮,openDoorBtn#开门按钮,QRCodeBtn#二维码按钮,authenticationSuccess#认证成功,authenticationFail#认证失败,statusBar#状态栏]-->clock
      </otherType>
    </WinMaterialInfo>
    <TouchProperty>
      <!--ro, opt, object, 触摸属性-->
      <windType>
        <!--ro, opt, enum, 窗口类型, subType:string, [popup#弹窗,page#页面]-->popup
      </windType>
      <hyperLinkType>
        <!--ro, opt, enum, 超链接类型, subType:string, [window#页面超链接,page#窗口超链接], desc:窗口类型为弹窗时有效-->window
      </hyperLinkType>
      <windowId>
        <!--ro, opt, int, 窗口编号, desc:当前页面的窗口,超链接类型为窗口时有效-->1
      </windowId>
      <pageId>
        <!--ro, opt, int, 页面编号, desc:超链接类型为页面时有效-->1
      </pageId>
    </TouchProperty>
    <PlayItemList>
      <!--ro, opt, array, 窗口播放列表, subType:object-->
      <PlayItem>
        <!--ro, req, object, 窗口播放信息-->
        <id>
          <!--ro, req, int, 播放序号-->1
        </id>
        <materialNo>
          <!--ro, opt, int, 素材编号-->1
        </materialNo>
        <inputChannel>
          <!--ro, opt, int, 绑定的IPC通道号-->1
        </inputChannel>
        <playEffect>
          <!--ro, req, enum, 播放效果, subType:string, [none#无,marquee#跑马灯]-->none
        </playEffect>
        <MarqueeInfo>
          <!--ro, opt, object, 跑马灯信息-->

```

```
<scrollType>
  <!--ro, req, enum, 跑马灯滚动类型, subType:string, [none#不滚动,Loops#循环滚动,once#滚动一次,backAndForth#来回滚动]-->none
</scrollType>
<scrollDeriction>
  <!--ro, req, enum, 跑马灯滚动方向, subType:string, [none#无,up#向上,down#向下,Left#向左,right#向右]-->none
</scrollDeriction>
<scrollSpeed>
  <!--ro, req, int, 跑马灯滚动速度-->1
</scrollSpeed>
</MarqueeInfo>
<PlayDuration>
  <!--ro, opt, object, 窗口内素材播放时间, desc:静态素材可以配置该时间,视频直播和IPC通道都可配置该节点-->
  <durationType>
    <!--ro, req, enum, 持续方式, subType:string, [materialTime#按素材时间,selfDefine#自定义]-->selfDefine
  </durationType>
  <duration>
    <!--ro, req, int, 持续时间, unit:s, unitType:时间-->1
  </duration>
</PlayDuration>
<CharactersEffect>
  <!--ro, opt, object, 文字显示效果, desc:当素材类型为文本txt和时有效-->
  <fontSize>
    <!--ro, req, int, 字体大小-->1
  </fontSize>
  <fontColor>
    <!--ro, req, object, 字体颜色-->
    <RGB>
      <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
    </RGB>
  </fontColor>
  <backgroundType>
    <!--ro, opt, enum, 背景类型, subType:string, [color#颜色(默认),picture#图片], desc:字段不存在时,默认为颜色-->color
  </backgroundType>
  <backPicId>
    <!--ro, opt, int, 背景图片素材ID-->1
  </backPicId>
  <BackColor>
    <!--ro, req, object, 背景颜色-->
    <RGB>
      <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
    </RGB>
  </BackColor>
  <backTransparent>
    <!--ro, req, int, 背景透明度-->1
  </backTransparent>
  <subtitlesEnabled>
    <!--ro, req, bool, 字幕使能-->true
  </subtitlesEnabled>
  <scrollDirection>
    <!--ro, req, enum, 文字滚动方向, subType:string, [Left#向左,right#向右,up#向上,down#向下]-->left
  </scrollDirection>
  <scrollSpeed>
    <!--ro, req, int, 文字滚动速度-->1
  </scrollSpeed>
</CharactersEffect>
<switchEffect>
  <!--ro, opt, enum, 窗口素材切换效果, subType:string, [none#无,LeftInRightOut#左进右出,rightInLeftOut#右进左出,bottomInTopOut#下进上出,topInBottomOut#上进下出,fadeInFadeOut#淡入淡出,middleExit#中间退出,topPop#顶部弹出,rightBottomIn#右下角进入,LeftTopIn#左上角进入,horizontalOpen#水平打开,verticalOpen#垂直打开,random#随机效果], desc:图片素材时有效-->random
  </switchEffect>
  <pageTime>
    <!--ro, opt, int, 翻页时间, unit:s, unitType:时间, desc:当素材为word,ppt,pdf,excel时有效-->1
  </pageTime>
  <scrollSpeed>
    <!--ro, req, int, 滚动速度, desc:当素材为静态web时有效-->1
  </scrollSpeed>
  <scaleType>
    <!--ro, opt, enum, 素材缩放参数配置, subType:string, [fitXY#拉伸填充,fitCenter#自适应,centerCrop#截取填充], desc:目前仅当图片窗口有效-->centerCrop
  </scaleType>
  <playType>
    <!--ro, opt, enum, 播放方式, subType:string, [normal#正常,faceAttendance#考勤,PeopleCounting#客流,temperature#测温], desc:目前仅当图片窗口有效-->normal
  </playType>
  <CharactersAttribute>
    <!--ro, opt, object, 文字属性, desc:当dynamicType为character时有效-->
    <fontSize>
      <!--ro, req, int, 字体大小-->1
    </fontSize>
    <fontColor>
      <!--ro, req, object, 字体颜色-->
      <RGB>
        <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
      </RGB>
    </fontColor>
    <backgroundType>
      <!--ro, req, enum, 背景类型, subType:string, [color#颜色,picture#图片]-->color
    </backgroundType>
    <BackColor>
      <!--ro, req, object, 背景色-->
      <RGB>
        <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
      </RGB>
    </BackColor>
```

```

</backgroundColor>
<backTransparent>
  <!--ro, opt, int, 背景透明度-->1
</backTransparent>
<alignType>
  <!--ro, req, enum, 对齐方式, subType:string, [left#左,right#右,middle#中]-->middle
</alignType>
<verticalAlignType>
  <!--ro, req, enum, 垂直对齐方式, subType:string, [top#顶,bottom#底,verticalCenter#中]-->top
</verticalAlignType>
<characterContent>
  <!--ro, opt, string, 文字内容, range:[0,512]-->test
</characterContent>
<backPicId>
  <!--ro, opt, int, 背景图片素材ID-->1
</backPicId>
<characterStyle>
  <!--ro, req, enum, 字体效果, subType:string, [normal#常规,bold#粗体]-->normal
</characterStyle>
<characterType>
  <!--ro, req, enum, 字体类型, subType:string, [mainTitle#主标题,subTitle#副标题]-->mainTitle
</characterType>
</CharactersAttribute>
<WebProperty>
  <!--ro, opt, object, 网页属性-->
<webType>
  <!--ro, opt, enum, 网页类型, subType:string, [desktop#桌面版,mobile#移动版], desc:默认按照设备的系统类型（Windows或Android）来显示网页--
>desktop
  </webType>
</WebProperty>
</PlayItem>
</PlayItemList>
<enabledAudio>
  <!--ro, opt, bool, 音频使能-->true
</enabledAudio>
<enableHide>
  <!--ro, opt, bool, 隐藏使能-->true
</enableHide>
<enableLock>
  <!--ro, opt, bool, 锁定使能-->true
</enableLock>
<AppWindow>
  <!--ro, opt, object, App窗口信息-->
  <windowInfoList>
    <!--ro, req, array, 窗口信息列表, subType:object-->
    <windowInfo>
      <!--ro, req, object, 窗口信息-->
      <id>
        <!--ro, req, int, 索引-->1
      </id>
      <materialNo>
        <!--ro, req, int, 素材编号, desc:采用本字段下发时, 用户需要先将App安装包当做素材存储到设备中, 得到素材编号再进行下发-->1
      </materialNo>
      <path>
        <!--ro, opt, string, App路径, range:[0,1024],
desc:path和materialNo字段同时存在时, 设备优先解析path字段;
采用本字段下发时, 设备需要确保App已经安装在设备内, 直接传入App的路径即可;
对于Android设备, path传的是App的包名; 对于Windows设备, path传的是App的应用程序绝对路径; -->test
      </path>
    </windowInfo>
  </windowInfoList>
</AppWindow>
<DataSource>
  <!--ro, opt, object, 数据来源, desc:当窗口为叫号和动态弹图时有效-->
  <materialNo>
    <!--ro, req, int, 素材编号-->1
  </materialNo>
</DataSource>
<Call>
  <!--ro, opt, object, 叫号数据-->
  <materialNo>
    <!--ro, req, int, 素材编号-->1
  </materialNo>
  <tableRow>
    <!--ro, req, int, 表格行-->1
  </tableRow>
  <tableColumn>
    <!--ro, req, int, 表格列-->1
  </tableColumn>
  <tableDirection>
    <!--ro, req, enum, 表格方向, subType:string, [vertical#纵向,horizontal#横向]-->vertical
  </tableDirection>
  <tableType>
    <!--ro, req, enum, 表格类型, subType:string, [template1#模板1,template2#模板2,template3#模板3,template4#模板4,template5#模板5,template6#模板
6]-->template1
  </tableType>
  <backPicId>
    <!--ro, opt, int, 背景图片素材编号-->1
  </backPicId>
  <alignType>
    <!--ro, req, enum, 对齐方式, subType:string, [left#左,right#右,middle#中]-->left
  </alignType>
  <refreshDirection>
    <!--ro, req, enum, 刷新方向, subType:string, [upToDown#从上到下,downToUp#从下到上,leftToRight#从左到右,rightToLeft#从右到左]-->upToDown
  </refreshDirection>

```

```
</refreshDirection>
<HeadDataList>
  <!--ro, opt, array, 表头数据列表, subType:object-->
  <HeadData>
    <!--ro, opt, object, 表头数据-->
    <id>
      <!--ro, req, int, 索引-->1
    </id>
    <data>
      <!--ro, req, string, 数据-->test
    </data>
  </HeadData>
</HeadDataList>
<ItemStyleList>
  <!--ro, opt, array, 表格列或行样式列表, subType:object-->
  <ItemStyle>
    <!--ro, opt, object, 表格列或行样式-->
    <id>
      <!--ro, req, int, 索引-->1
    </id>
    <width>
      <!--ro, req, int, 每列宽度所占百分比-->1
    </width>
    <fontSize>
      <!--ro, req, int, 字体大小-->1
    </fontSize>
    <fontColor>
      <!--ro, req, object, 字体颜色-->
      <RGB>
        <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
      </RGB>
    </fontColor>
    <BackColor>
      <!--ro, req, object, 背景色-->
      <RGB>
        <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
      </RGB>
    </BackColor>
  </ItemStyle>
</ItemStyleList>
</Call>
<DynamicPic>
  <!--ro, opt, object, 动态弹图窗口配置-->
  <materialNo>
    <!--ro, req, int, 素材编号-->1
  </materialNo>
  <backPicId>
    <!--ro, opt, int, 背景图片素材ID-->1
  </backPicId>
</DynamicPic>
<CapturePic>
  <!--ro, opt, object, 抓拍图片参数-->
  <backPicId>
    <!--ro, opt, int, 背景图片素材ID-->1
  </backPicId>
  <ipcMaterialNo>
    <!--ro, req, int, IPC素材编号-->1
  </ipcMaterialNo>
  <cancelType>
    <!--ro, req, enum, 取消方式, subType:string, [auto#自动,manual#手动]-->manual
  </cancelType>
  <duration>
    <!--ro, opt, int, 持续时间, unit:s, unitType:时间-->1
  </duration>
</CapturePic>
<ClockParam>
  <!--ro, opt, object, 时钟参数-->
  <backPicId>
    <!--ro, opt, int, 背景图片素材ID-->1
  </backPicId>
  <ClockIcon>
    <!--ro, req, object, 时钟图标参数-->
    <enabled>
      <!--ro, req, bool, 使能-->true
    </enabled>
    <type>
      <!--ro, opt, enum, 时钟类型, subType:string, [cLock1#cLock1,cLock2#cLock2]-->cLock1
    </type>
  </ClockIcon>
  <Position>
    <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--ro, req, int, x坐标-->1
    </positionX>
    <positionY>
      <!--ro, req, int, y坐标-->1
    </positionY>
    <height>
      <!--ro, req, int, 宽-->1
    </height>
    <width>
      <!--ro, req, int, 高-->1
    </width>
  </Position>
```

```

</ClockIcon>
<YmdParam>
  <!--ro, req, object, 时钟年月日参数-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <fontSize>
    <!--ro, req, int, 字体大小-->1
  </fontSize>
  <FontColor>
    <!--ro, req, object, 字体颜色-->
    <RGB>
      <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
    </RGB>
  </FontColor>
  <BackColor>
    <!--ro, req, object, 背景色-->
    <RGB>
      <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
    </RGB>
  </BackColor>
  <Position>
    <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--ro, req, int, x坐标-->1
    </positionX>
    <positionY>
      <!--ro, req, int, y坐标-->1
    </positionY>
    <height>
      <!--ro, req, int, 宽-->1
    </height>
    <width>
      <!--ro, req, int, 高-->1
    </width>
  </Position>
</YmdParam>
<HmsParam>
  <!--ro, req, object, 时钟时分秒参数-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <fontSize>
    <!--ro, req, int, 字体大小-->1
  </fontSize>
  <FontColor>
    <!--ro, req, object, 字体颜色-->
    <RGB>
      <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
    </RGB>
  </FontColor>
  <BackColor>
    <!--ro, req, object, 背景色-->
    <RGB>
      <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
    </RGB>
  </BackColor>
  <Position>
    <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--ro, req, int, x坐标-->1
    </positionX>
    <positionY>
      <!--ro, req, int, y坐标-->1
    </positionY>
    <height>
      <!--ro, req, int, 宽-->1
    </height>
    <width>
      <!--ro, req, int, 高-->1
    </width>
  </Position>
</HmsParam>
<WeekParam>
  <!--ro, req, object, 时钟星期参数-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <fontSize>
    <!--ro, req, int, 字体大小-->1
  </fontSize>
  <FontColor>
    <!--ro, req, object, 字体颜色-->
    <RGB>
      <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
    </RGB>
  </FontColor>
  <BackColor>
    <!--ro, req, object, 背景色-->
    <RGB>
      <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
    </RGB>
  </BackColor>
  <Position>

```

```
<!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
<positionX>
  <!--ro, req, int, x坐标-->1
</positionX>
<positionY>
  <!--ro, req, int, y坐标-->1
</positionY>
<height>
  <!--ro, req, int, 宽-->1
</height>
<width>
  <!--ro, req, int, 高-->1
</width>
</Position>
</WeekParam>
</ClockParam>
<WeatherParam>
  <!--ro, opt, object, 天气参数-->
  <backPicId>
    <!--ro, opt, int, 天气背景图片素材id-->1
  </backPicId>
  <WeatherIcon>
    <!--ro, opt, object, 天气图标参数-->
    <enabled>
      <!--ro, req, bool, 使能-->true
    </enabled>
    <Position>
      <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
      <positionX>
        <!--ro, req, int, x坐标-->1
      </positionX>
      <positionY>
        <!--ro, req, int, y坐标-->1
      </positionY>
      <height>
        <!--ro, req, int, 宽-->1
      </height>
      <width>
        <!--ro, req, int, 高-->1
      </width>
    </Position>
  </WeatherIcon>
  <Date>
    <!--ro, opt, object, 天气日期参数-->
    <enabled>
      <!--ro, req, bool, 使能-->true
    </enabled>
    <fontSize>
      <!--ro, req, int, 字体大小-->1
    </fontSize>
    <FontColor>
      <!--ro, req, object, 字体颜色-->
      <RGB>
        <!--ro, opt, int, RGB值, desc:十进制数据表示三原色, 如16777215表示0xFFFFFF-->1
      </RGB>
    </FontColor>
    <BackColor>
      <!--ro, req, object, 背景色-->
      <RGB>
        <!--ro, opt, int, RGB值, desc:十进制数据表示三原色, 如16777215表示0xFFFFFF-->1
      </RGB>
    </BackColor>
    <Position>
      <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
      <positionX>
        <!--ro, req, int, x坐标-->1
      </positionX>
      <positionY>
        <!--ro, req, int, y坐标-->1
      </positionY>
      <height>
        <!--ro, req, int, 宽-->1
      </height>
      <width>
        <!--ro, req, int, 高-->1
      </width>
    </Position>
  </Date>
  <Temperature>
    <!--ro, opt, object, 天气温度参数-->
    <enabled>
      <!--ro, req, bool, 使能-->true
    </enabled>
    <fontSize>
      <!--ro, req, int, 字体大小-->1
    </fontSize>
    <FontColor>
      <!--ro, req, object, 字体颜色-->
      <RGB>
        <!--ro, opt, int, RGB值, desc:十进制数据表示三原色, 如16777215表示0xFFFFFF-->1
      </RGB>
    </FontColor>
    <BackColor>
```

```
<!--ro, req, object, 背景色-->
<RGB>
<!--ro, opt, int, RGB值, desc:十进制数据表示三原色, 如16777215表示0xFFFFFF-->1
</RGB>
</BackColor>
<Position>
<!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
<positionX>
<!--ro, req, int, x坐标-->1
</positionX>
<positionY>
<!--ro, req, int, y坐标-->1
</positionY>
<height>
<!--ro, req, int, 宽-->1
</height>
<width>
<!--ro, req, int, 高-->1
</width>
</Position>
</Temperature>
<WeatherContent>
<!--ro, opt, object, 天气情况参数-->
<enabled>
<!--ro, req, bool, 使能-->true
</enabled>
<fontSize>
<!--ro, req, int, 字体大小-->1
</fontSize>
<FontColor>
<!--ro, req, object, 字体颜色-->
<RGB>
<!--ro, opt, int, RGB值, desc:十进制数据表示三原色, 如16777215表示0xFFFFFF-->1
</RGB>
</FontColor>
<BackColor>
<!--ro, req, object, 背景色-->
<RGB>
<!--ro, opt, int, RGB值, desc:十进制数据表示三原色, 如16777215表示0xFFFFFF-->1
</RGB>
</BackColor>
<Position>
<!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
<positionX>
<!--ro, req, int, x坐标-->1
</positionX>
<positionY>
<!--ro, req, int, y坐标-->1
</positionY>
<height>
<!--ro, req, int, 宽-->1
</height>
<width>
<!--ro, req, int, 高-->1
</width>
</Position>
</WeatherContent>
<City>
<!--ro, opt, object, 天气城市参数-->
<enabled>
<!--ro, req, bool, 使能-->true
</enabled>
<cityId>
<!--ro, req, string, 城市编号-->test
</cityId>
<cityName>
<!--ro, req, string, 城市名称-->test
</cityName>
<fontSize>
<!--ro, req, int, 字体大小-->1
</fontSize>
<FontColor>
<!--ro, req, object, 字体颜色-->
<RGB>
<!--ro, opt, int, RGB值, desc:十进制数据表示三原色, 如16777215表示0xFFFFFF-->1
</RGB>
</FontColor>
<BackColor>
<!--ro, req, object, 背景色-->
<RGB>
<!--ro, opt, int, RGB值, desc:十进制数据表示三原色, 如16777215表示0xFFFFFF-->1
</RGB>
</BackColor>
<Position>
<!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
<positionX>
<!--ro, req, int, x坐标-->1
</positionX>
<positionY>
<!--ro, req, int, y坐标-->1
</positionY>
<height>
<!--ro, req, int, 宽-->1
</height>
```

```

</height>
<width>
  <!--ro, req, int, 高-->1
</width>
</Position>
</City>
<Humidity>
  <!--ro, opt, object, 天气湿度参数-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <fontSize>
    <!--ro, req, int, 字体大小-->1
  </fontSize>
  <FontColor>
    <!--ro, req, object, 字体颜色-->
    <RGB>
      <!--ro, opt, int, RGB值, desc:十进制数据表示三原色, 如16777215表示0xFFFFFF-->1
    </RGB>
    </FontColor>
  <BackColor>
    <!--ro, req, object, 背景色-->
    <RGB>
      <!--ro, opt, int, RGB值, desc:十进制数据表示三原色, 如16777215表示0xFFFFFF-->1
    </RGB>
    </BackColor>
  <Position>
    <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--ro, req, int, x坐标-->1
    </positionX>
    <positionY>
      <!--ro, req, int, y坐标-->1
    </positionY>
    <height>
      <!--ro, req, int, 宽-->1
    </height>
    <width>
      <!--ro, req, int, 高-->1
    </width>
    </Position>
  </Humidity>
  <AirQuality>
    <!--ro, opt, object, 天气空气质量参数-->
    <enabled>
      <!--ro, req, bool, 使能-->true
    </enabled>
    <fontSize>
      <!--ro, req, int, 字体大小-->1
    </fontSize>
    <FontColor>
      <!--ro, req, object, 字体颜色-->
      <RGB>
        <!--ro, opt, int, RGB值, desc:十进制数据表示三原色, 如16777215表示0xFFFFFF-->1
      </RGB>
      </FontColor>
    <BackColor>
      <!--ro, req, object, 背景色-->
      <RGB>
        <!--ro, opt, int, RGB值, desc:十进制数据表示三原色, 如16777215表示0xFFFFFF-->1
      </RGB>
      </BackColor>
    <Position>
      <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
      <positionX>
        <!--ro, req, int, x坐标-->1
      </positionX>
      <positionY>
        <!--ro, req, int, y坐标-->1
      </positionY>
      <height>
        <!--ro, req, int, 宽-->1
      </height>
      <width>
        <!--ro, req, int, 高-->1
      </width>
      </Position>
    </AirQuality>
    <UpdateTime>
      <!--ro, opt, object, 天气更新时间参数-->
      <enabled>
        <!--ro, req, bool, 使能-->true
      </enabled>
      <refreshTime>
        <!--ro, req, string, 刷新时间, desc:设备早期返回时间格式有误, 错为: 20200826T051500+08, 为保证XSD校验通过, 改为string类型-->20200826T051500+08
      </refreshTime>
      <updateInterval>
        <!--ro, req, int, 刷新间隔, unit:minute-->1
      </updateInterval>
      <fontSize>
        <!--ro, req, int, 字体大小-->1
      </fontSize>
      <FontColor>

```

```
<!--ro, req, object, 字体颜色-->
<RGB>
  <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
</RGB>
</FontColor>
<BackColor>
  <!--ro, req, object, 背景色-->
  <RGB>
    <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
  </RGB>
</BackColor>
<Position>
  <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
  <positionX>
    <!--ro, req, int, x坐标-->1
  </positionX>
  <positionY>
    <!--ro, req, int, y坐标-->1
  </positionY>
  <height>
    <!--ro, req, int, 宽-->1
  </height>
  <width>
    <!--ro, req, int, 高-->1
  </width>
</Position>
</UpdateTime>
<Wind>
  <!--ro, opt, object, 天气风力参数-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <fontSize>
    <!--ro, req, int, 字体大小-->1
  </fontSize>
  <FontColor>
    <!--ro, req, object, 字体颜色-->
    <RGB>
      <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
    </RGB>
  </FontColor>
  <BackColor>
    <!--ro, req, object, 背景色-->
    <RGB>
      <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
    </RGB>
  </BackColor>
  <Position>
    <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--ro, req, int, x坐标-->1
    </positionX>
    <positionY>
      <!--ro, req, int, y坐标-->1
    </positionY>
    <height>
      <!--ro, req, int, 宽-->1
    </height>
    <width>
      <!--ro, req, int, 高-->1
    </width>
  </Position>
</Wind>
</WeatherParam>
<Countdown>
  <!--ro, opt, object, 倒计时素材-->
  <endTime>
    <!--ro, req, string, 倒计时时间点, desc:设备早期返回时间格式有误, 错为: 20200826T051500+08, 为保证XSD校验通过, 改为string类型-->20200826T051500+08
  </endTime>
  <template>
    <!--ro, req, enum, 模板类型, subType:string, [template1#模板1,template2#模板2,openingActivity1#活动开幕1,openingActivity2#活动开幕2,commercialActivity1#商业活动1,commercialActivity2#商业活动2,colLegeEntranceExam#高考,overseasTimeFormat#海外时间格式]-->template1
  </template>
  <timeUnit>
    <!--ro, req, enum, 时间单位, subType:string, [year#年,month#月,day#日,week#周,hour#时,minute#分,second#秒]-->year
  </timeUnit>
  <backPicId>
    <!--ro, opt, int, 背景图片素材ID-->1
  </backPicId>
  <TimeFontCfg>
    <!--ro, opt, object, 时间字体参数-->
    <fontSize>
      <!--ro, req, int, 字体大小-->1
    </fontSize>
    <FontColor>
      <!--ro, req, object, 字体颜色-->
      <RGB>
        <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
      </RGB>
    </FontColor>
    <Position>
      <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
```

```
<!--ro, req, int, x坐标-->1
</positionX>
<positionY>
<!--ro, req, int, y坐标-->1
</positionY>
<height>
<!--ro, req, int, 宽-->1
</height>
<width>
<!--ro, req, int, 高-->1
</width>
</Position>
</TimeFontCfg>
<timeUnitEnabled>
<!--ro, opt, bool, 时间单位是否有效-->true
</timeUnitEnabled>
<maxTimeUnit>
<!--ro, req, enum, 最大时间单位, subType:string, [year#年,month#月,day#日,week#周,hour#时,minute#分,second#秒,week#周], desc:表示当前倒计时时的
最大显示单元,例如:maxTimeUnit为月,则1年3个月,对应月份值为15-->year
</maxTimeUnit>
<ModuleInfoList>
<!--ro, opt, array, 界面模块信息列表, subType:object-->
<ModuleInfo>
<!--ro, opt, object, 界面模块信息-->
<moduleType>
<!--ro, req, enum, 模块类型, subType:string, [year#年,month#月,day#日,hour#时,minute#分,second#秒,week#周,text#文字], desc:text为时间旁边
的文字,例如年下面的文字"年"-->second
</moduleType>
<enabled>
<!--ro, req, bool, 使能-->true
</enabled>
<fontSize>
<!--ro, req, int, 字体大小-->1
</fontSize>
<fontColor>
<!--ro, req, object, 字体颜色-->
<RGB>
<!--ro, opt, int, 十进制数据表示三原色-->1
</RGB>
</FontColor>
<text>
<!--ro, opt, string, 文字, desc:当moduleType为text时有效-->test
</text>
<Position>
<!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
<positionX>
<!--ro, req, int, X坐标-->1
</positionX>
<positionY>
<!--ro, req, int, Y坐标-->1
</positionY>
<height>
<!--ro, req, int, 宽-->1
</height>
<width>
<!--ro, req, int, 高-->1
</width>
</Position>
</ModuleInfo>
</ModuleInfoList>
</Countdown>
<localInputNo>
<!--ro, req, enum, 本地输入口类型, subType:string, [VGA#VGA,HDMI#HDMI#HDMI]-->VGA
</localInputNo>
<HyperlinkBtn>
<!--ro, opt, object, 超链接按钮-->
<backPicId>
<!--ro, opt, int, 背景图片素材ID-->1
</backPicId>
</HyperlinkBtn>
<CharactersAttribute>
<!--ro, opt, object, 文字属性, desc:当dynamicType为character时有效-->
<fontSize>
<!--ro, req, int, 字体大小-->1
</fontSize>
<fontColor>
<!--ro, req, object, 字体颜色-->
<RGB>
<!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
</RGB>
</FontColor>
<fontType>
<!--ro, opt, enum, 字体类型, subType:string, [normal#常规,bold#粗体]-->normal
</fontType>
<backgroundType>
<!--ro, req, enum, 背景类型, subType:string, [color#颜色,picture#图片]-->color
</backgroundType>
<BackColor>
<!--ro, req, object, 背景色-->
<RGB>
<!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
</RGB>
</fillEnabled>
```

```
<!--ro, opt, bool, 背景色填充使能, desc:默认为true, 表示背景色填充-->true
</fillEnabled>
</BackColor>
<backTransparent>
  <!--ro, opt, int, 背景透明度-->1
</backTransparent>
<alignType>
  <!--ro, req, enum, 对齐方式, subType:string, [left#左,right#右,middle#中]-->middle
</alignType>
<verticalAlignType>
  <!--ro, req, enum, 垂直对齐方式, subType:string, [top#顶,bottom#底,verticalCenter#中]-->top
</verticalAlignType>
<characterContent>
  <!--ro, opt, string, 文字内容, range:[0,512]-->test
</characterContent>
<backPicId>
  <!--ro, opt, int, 背景图片素材ID-->1
</backPicId>
<characterStyle>
  <!--ro, req, enum, 字体效果, subType:string, [normal#常规,bold#粗体]-->normal
</characterStyle>
<characterType>
  <!--ro, req, enum, 字体类型, subType:string, [mainTitle#主标题,subTitle#副标题]-->mainTitle
</characterType>
<scrollDirection>
  <!--ro, opt, enum, 文字滚动方向, subType:string, [left#向左,right#向右,up#向上,down#向下,turnPage#翻页]-->left
</scrollDirection>
<scrollSpeed>
  <!--ro, opt, int, 文字滚动速度, range:[1,10]-->1
</scrollSpeed>
</CharactersAttribute>
<SignIn>
  <!--ro, opt, object, 人员签到展示窗口, desc:当staticMaterialType为signIn时有效-->
  <backPicId>
    <!--ro, opt, int, 背景图片素材ID-->1
  </backPicId>
  <ShowInfo>
    <!--ro, opt, object, 展示信息-->
    <row>
      <!--ro, opt, int, 窗口内人员签到展示行数-->1
    </row>
    <column>
      <!--ro, opt, int, 窗口内人员签到展示列数-->1
    </column>
    <Position>
      <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
      <positionX>
        <!--ro, req, int, x坐标-->1
      </positionX>
      <positionY>
        <!--ro, req, int, y坐标-->1
      </positionY>
      <height>
        <!--ro, req, int, 宽-->1
      </height>
      <width>
        <!--ro, req, int, 高-->1
      </width>
    </Position>
    <inLib>
      <!--ro, opt, object, 在库信息-->
      <nameEnabled>
        <!--ro, opt, bool, 在库人员是否展示姓名-->true
      </nameEnabled>
      <genderEnabled>
        <!--ro, opt, bool, 在库人员是否展示性别-->true
      </genderEnabled>
      <selfDefine1Enabled>
        <!--ro, opt, bool, 在库人员是否展示自定义字段1-->true
      </selfDefine1Enabled>
      <selfDefine2Enabled>
        <!--ro, opt, bool, 在库人员是否展示自定义字段2-->true
      </selfDefine2Enabled>
      <signInTimeEnabled>
        <!--ro, opt, bool, 在库人员是否展示签到时间-->true
      </signInTimeEnabled>
      <temperatureEnabled>
        <!--ro, opt, bool, 在库人员是否展示温度-->true
      </temperatureEnabled>
    </inLib>
    <outLib>
      <!--ro, opt, object, 不在库信息-->
      <nameEnabled>
        <!--ro, opt, bool, 未知人员是否展示姓名-->true
      </nameEnabled>
      <name>
        <!--ro, opt, string, 未知人员姓名-->test
      </name>
      <genderEnabled>
        <!--ro, opt, bool, 未知人员是否展示性别-->true
      </genderEnabled>
      <signInTimeEnabled>
        <!--ro, opt, bool, 未知人员是否展示签到时间-->true
      </signInTimeEnabled>
```

```
</signInTimeEnabled>
<temperatureEnabled>
  <!--ro, opt, bool, 未知人员是否展示温度-->true
</temperatureEnabled>
</outLib>
<showDuration>
  <!--ro, req, enum, 签到记录显示时间, subType:string, [always#不消失,10s#10s,20s#20s,30s#30s,no#不显示]-->always
</showDuration>
</ShowInfo>
<MainTitle>
  <!--ro, opt, object, 主标题-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <fontSize>
    <!--ro, req, int, 字体大小-->1
  </fontSize>
  <FontColor>
    <!--ro, req, object, 字体颜色-->
    <RGB>
      <!--ro, opt, int, 十进制数据表示三原色-->1
    </RGB>
  </FontColor>
  <Position>
    <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--ro, req, int, x坐标-->1
    </positionX>
    <positionY>
      <!--ro, req, int, y坐标-->1
    </positionY>
    <height>
      <!--ro, req, int, 宽-->1
    </height>
    <width>
      <!--ro, req, int, 高-->1
    </width>
  </Position>
  <text>
    <!--ro, opt, string, 标题文字-->test
  </text>
</MainTitle>
<SubTitle>
  <!--ro, opt, object, 副标题-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <fontSize>
    <!--ro, req, int, 字体大小-->1
  </fontSize>
  <FontColor>
    <!--ro, req, object, 字体颜色-->
    <RGB>
      <!--ro, opt, int, 十进制数据表示三原色-->1
    </RGB>
  </FontColor>
  <Position>
    <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--ro, req, int, x坐标-->1
    </positionX>
    <positionY>
      <!--ro, req, int, y坐标-->1
    </positionY>
    <height>
      <!--ro, req, int, 宽-->1
    </height>
    <width>
      <!--ro, req, int, 高-->1
    </width>
  </Position>
  <text>
    <!--ro, opt, string, 标题文字-->test
  </text>
</SubTitle>
<NormalTemperature>
  <!--ro, opt, object, 正常体温信息-->
  <PeopleNumber>
    <!--ro, opt, object, 正常体温人数信息-->
    <enabled>
      <!--ro, req, bool, 使能-->true
    </enabled>
    <fontSize>
      <!--ro, req, int, 字体大小-->1
    </fontSize>
    <FontColor>
      <!--ro, req, object, 字体颜色-->
      <RGB>
        <!--ro, opt, int, 十进制数据表示三原色-->1
      </RGB>
    </FontColor>
    <Position>
      <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
      <positionX>
```

```

<!--ro, req, int, X坐标-->1
</positionX>
<positionY>
<!--ro, req, int, Y坐标-->1
</positionY>
<height>
<!--ro, req, int, 宽-->1
</height>
<width>
<!--ro, req, int, 高-->1
</width>
</Position>
<text>
<!--ro, opt, string, 标题文字-->test
</text>
</PeopleNumber>
<Icon>
<!--ro, opt, object, 正常体温图标-->
<enabled>
<!--ro, req, bool, 使能-->true
</enabled>
<Position>
<!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
<positionX>
<!--ro, req, int, X坐标-->1
</positionX>
<positionY>
<!--ro, req, int, Y坐标-->1
</positionY>
<height>
<!--ro, req, int, 宽-->1
</height>
<width>
<!--ro, req, int, 高-->1
</width>
</Position>
<materialNo>
<!--ro, opt, int, 素材ID-->1
</materialNo>
</Icon>
</NormalTemperature>
<AbnormalTemperature>
<!--ro, opt, object, 异常体温信息-->
<PeopleNumber>
<!--ro, opt, object, 异常体温人数信息-->
<enabled>
<!--ro, req, bool, 使能-->true
</enabled>
<fontSize>
<!--ro, req, int, 字体大小-->1
</fontSize>
<fontColor>
<!--ro, req, object, 字体颜色-->
<RGB>
<!--ro, opt, int, 十进制数据表示三原色-->1
</RGB>
</FontColor>
<Position>
<!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
<positionX>
<!--ro, req, int, X坐标-->1
</positionX>
<positionY>
<!--ro, req, int, Y坐标-->1
</positionY>
<height>
<!--ro, req, int, 宽-->1
</height>
<width>
<!--ro, req, int, 高-->1
</width>
</Position>
<text>
<!--ro, opt, string, 标题文字-->test
</text>
</PeopleNumber>
<Icon>
<!--ro, opt, object, 异常体温图标-->
<enabled>
<!--ro, req, bool, 使能-->true
</enabled>
<Position>
<!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
<positionX>
<!--ro, req, int, X坐标-->1
</positionX>
<positionY>
<!--ro, req, int, Y坐标-->1
</positionY>
<height>
<!--ro, req, int, 宽-->1
</height>
<width>

```

```
<!--ro, req, int, 高-->1
</width>
</Position>
<materialNo>
<!--ro, opt, int, 素材ID-->1
</materialNo>
</Icon>
</AbnormalTemperature>
</SignIn>
<EventParam>
<!--ro, opt, object, 事件参数-->
<EventResourceList>
<!--ro, opt, array, 事件源信息列表, subType:object-->
<EventResource>
<!--ro, opt, object, 事件源信息-->
<ipVersion>
<!--ro, req, enum, ip地址版本, subType:string, [v4#v4,v6#v6,dual#双]-->v4
</ipVersion>
<ipAddress>
<!--ro, opt, string, ipv4地址-->test
</ipAddress>
<ipv6Address>
<!--ro, opt, string, ipv6地址-->test
</ipv6Address>
<portNo>
<!--ro, req, int, 端口号-->1
</portNo>
<userName>
<!--ro, req, string, 用户名-->test
</userName>
<password>
<!--ro, req, string, 密码-->test
</password>
<channels>
<!--ro, opt, string, 通道号, desc:多选以逗号隔开-->1,2
</channels>
</EventResource>
</EventResourceList>
<backPicId>
<!--ro, opt, int, 背景图片素材ID-->1
</backPicId>
<Title>
<!--ro, opt, object, 标题-->
<enabled>
<!--ro, req, bool, 使能-->true
</enabled>
<fontSize>
<!--ro, req, int, 字体大小-->1
</fontSize>
<fontColor>
<!--ro, req, object, 字体颜色-->
<RGB>
<!--ro, opt, int, 十进制数据表示三原色-->1
</RGB>
</FontColor>
<Position>
<!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
<positionX>
<!--ro, req, int, X坐标-->1
</positionX>
<positionY>
<!--ro, req, int, Y坐标-->1
</positionY>
<height>
<!--ro, req, int, 宽-->1
</height>
<width>
<!--ro, req, int, 高-->1
</width>
</Position>
<text>
<!--ro, opt, string, 标题文字-->test
</text>
</Title>
<PeopleCounting>
<!--ro, opt, object, 人数统计提醒-->
<maxPeopleNumber>
<!--ro, opt, int, 最大可进入人数-->1
</maxPeopleNumber>
<unavailableBackPicId>
<!--ro, opt, int, 人员不能进入时控件背景图片id-->1
</unavailableBackPicId>
<supportChangeByPeople>
<!--ro, req, enum, 支持根据人数修改样式的字段, subType:string,
[prompt#prompt,peopleIcon#peopleIcon,availablePeopleNumber#availablePeopleNumber]-->prompt
</supportChangeByPeople>
<Prompt>
<!--ro, opt, object, 提示信息-->
<enabled>
<!--ro, req, bool, 使能-->true
</enabled>
<fontSize>
<!--ro, req, int, 字体大小-->1
</fontSize>
```

```
</fontSize>
<Position>
  <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
  <positionX>
    <!--ro, req, int, X坐标-->1
  </positionX>
  <positionY>
    <!--ro, req, int, Y坐标-->1
  </positionY>
  <height>
    <!--ro, req, int, 宽-->1
  </height>
  <width>
    <!--ro, req, int, 高-->1
  </width>
</Position>
<text>
  <!--ro, opt, string, 标题文字-->test
</text>
<FontColor>
  <!--ro, opt, object, 人员可以进入时的字体颜色-->
  <RGB>
    <!--ro, opt, int, 十进制数据表示三原色-->1
  </RGB>
</FontColor>
<unavailableText>
  <!--ro, opt, string, 人员不能进入时的标题文字-->test
</unavailableText>
<UnavailableFontColor>
  <!--ro, opt, object, 人员不能进入时的字体颜色-->
  <RGB>
    <!--ro, opt, int, 十进制数据表示三原色-->1
  </RGB>
</UnavailableFontColor>
</Prompt>
<PeopleIcon>
  <!--ro, opt, object, 行人通行图标-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <Position>
    <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--ro, req, int, X坐标-->1
    </positionX>
    <positionY>
      <!--ro, req, int, Y坐标-->1
    </positionY>
    <height>
      <!--ro, req, int, 宽-->1
    </height>
    <width>
      <!--ro, req, int, 高-->1
    </width>
  </Position>
  <materialNo>
    <!--ro, opt, int, 素材ID-->1
  </materialNo>
</PeopleIcon>
<InsidePeopleTitle>
  <!--ro, opt, object, 已经进入人数标题-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <fontSize>
    <!--ro, req, int, 字体大小-->1
  </fontSize>
  <FontColor>
    <!--ro, req, object, 字体颜色-->
    <RGB>
      <!--ro, opt, int, 十进制数据表示三原色-->1
    </RGB>
  </FontColor>
  <Position>
    <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--ro, req, int, X坐标-->1
    </positionX>
    <positionY>
      <!--ro, req, int, Y坐标-->1
    </positionY>
    <height>
      <!--ro, req, int, 宽-->1
    </height>
    <width>
      <!--ro, req, int, 高-->1
    </width>
  </Position>
  <text>
    <!--ro, opt, string, 标题文字-->test
  </text>
</InsidePeopleTitle>
<InsidePeopleNumber>
  <!--ro, opt, object, 已经进入人数-->
```

```
<enabled>
  <!--ro, req, bool, 使能-->true
</enabled>
<fontSize>
  <!--ro, req, int, 字体大小-->1
</fontSize>
<fontColor>
  <!--ro, req, object, 字体颜色-->
  <RGB>
    <!--ro, opt, int, 十进制数据表示三原色-->1
  </RGB>
</FontColor>
<Position>
  <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
  <positionX>
    <!--ro, req, int, X坐标-->1
  </positionX>
  <positionY>
    <!--ro, req, int, Y坐标-->1
  </positionY>
  <height>
    <!--ro, req, int, 宽-->1
  </height>
  <width>
    <!--ro, req, int, 高-->1
  </width>
</Position>
</InsidePeopleNumber>
<AvailablePeopleTitle>
  <!--ro, opt, object, 还可以进入人数标题-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <fontSize>
    <!--ro, req, int, 字体大小-->1
  </fontSize>
  <fontColor>
    <!--ro, req, object, 字体颜色-->
    <RGB>
      <!--ro, opt, int, 十进制数据表示三原色-->1
    </RGB>
  </FontColor>
  <Position>
    <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--ro, req, int, X坐标-->1
    </positionX>
    <positionY>
      <!--ro, req, int, Y坐标-->1
    </positionY>
    <height>
      <!--ro, req, int, 宽-->1
    </height>
    <width>
      <!--ro, req, int, 高-->1
    </width>
  </Position>
  <text>
    <!--ro, opt, string, 标题文字-->test
  </text>
</AvailablePeopleTitle>
<AvailablePeopleNumber>
  <!--ro, opt, object, 还可以进入人数-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <fontSize>
    <!--ro, req, int, 字体大小-->1
  </fontSize>
  <Position>
    <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--ro, req, int, X坐标-->1
    </positionX>
    <positionY>
      <!--ro, req, int, Y坐标-->1
    </positionY>
    <height>
      <!--ro, req, int, 宽-->1
    </height>
    <width>
      <!--ro, req, int, 高-->1
    </width>
  </Position>
  <fontColor>
    <!--ro, opt, object, 人员可以进入时的字体颜色-->
    <RGB>
      <!--ro, opt, int, 十进制数据表示三原色-->1
    </RGB>
  </FontColor>
  <UnavailableFontColor>
    <!--ro, opt, object, 人员不能进入时的字体颜色-->
    <RGB>
```

```
<!--ro, opt, int, 十进制数据表示三原色-->1
</RGB>
</UnavailableFontColor>
</AvailablePeopleNumber>
<UnavailablePeopleIcon>
<!--ro, opt, object, 人员不能进入时的行人图标-->
<enabled>
<!--ro, req, bool, 使能-->true
</enabled>
<Position>
<!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
<positionX>
<!--ro, req, int, X坐标-->1
</positionX>
<positionY>
<!--ro, req, int, Y坐标-->1
</positionY>
<height>
<!--ro, req, int, 宽-->1
</height>
<width>
<!--ro, req, int, 高-->1
</width>
</Position>
<materialNo>
<!--ro, opt, int, 素材ID-->1
</materialNo>
</UnavailablePeopleIcon>
<AudioAlarm>
<!--ro, opt, object, 音频报警信息-->
<enabled>
<!--ro, req, bool, 使能-->true
</enabled>
<materialNo>
<!--ro, opt, int, 音频素材ID-->1
</materialNo>
</AudioAlarm>
</PeopleCounting>
<Mask>
<!--ro, opt, object, 未戴口罩提醒-->
<enabled>
<!--ro, req, bool, 使能-->true
</enabled>
<fontSize>
<!--ro, req, int, 字体大小-->1
</fontSize>
<fontColor>
<!--ro, req, object, 字体颜色-->
<RGB>
<!--ro, opt, int, 十进制数据表示三原色-->1
</RGB>
</FontColor>
<BackColor>
<!--ro, req, object, 背景颜色-->
<RGB>
<!--ro, opt, int, 十进制数据表示三原色-->1
</RGB>
</BackColor>
<Position>
<!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
<positionX>
<!--ro, req, int, X坐标-->1
</positionX>
<positionY>
<!--ro, req, int, Y坐标-->1
</positionY>
<height>
<!--ro, req, int, 宽-->1
</height>
<width>
<!--ro, req, int, 高-->1
</width>
</Position>
<text>
<!--ro, opt, string, 标题文字-->test
</text>
<MaskIcon>
<!--ro, opt, object, 未戴口罩图标-->
<enabled>
<!--ro, req, bool, 使能-->true
</enabled>
<Position>
<!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
<positionX>
<!--ro, req, int, X坐标-->1
</positionX>
<positionY>
<!--ro, req, int, Y坐标-->1
</positionY>
<height>
<!--ro, req, int, 宽-->1
</height>
<width>
<!--ro, req, int, 高-->1
```

```
</width>
</Position>
<materialNo>
  <!--ro, opt, int, 素材ID-->1
</materialNo>
</MaskIcon>
<AudioAlarm>
  <!--ro, opt, object, 音频报警信息-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <materialNo>
    <!--ro, opt, int, 音频素材ID-->1
  </materialNo>
</AudioAlarm>
</Mask>
<Temperature>
  <!--ro, opt, object, 温度异常提醒-->
  <Prompt>
    <!--ro, opt, object, 提示信息-->
    <enabled>
      <!--ro, req, bool, 使能-->true
    </enabled>
    <fontSize>
      <!--ro, req, int, 字体大小-->1
    </fontSize>
    <fontColor>
      <!--ro, req, object, 字体颜色-->
      <RGB>
        <!--ro, opt, int, 十进制数据表示三原色-->1
      </RGB>
    </fontColor>
    <BackColor>
      <!--ro, req, object, 背景颜色-->
      <RGB>
        <!--ro, opt, int, 十进制数据表示三原色-->1
      </RGB>
    </BackColor>
    <Position>
      <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
      <positionX>
        <!--ro, req, int, X坐标-->1
      </positionX>
      <positionY>
        <!--ro, req, int, Y坐标-->1
      </positionY>
      <height>
        <!--ro, req, int, 宽-->1
      </height>
      <width>
        <!--ro, req, int, 高-->1
      </width>
    </Position>
    <text>
      <!--ro, opt, string, 标题文字-->test
    </text>
  </Prompt>
  <TemperatureIcon>
    <!--ro, opt, object, 温度异常图标-->
    <enabled>
      <!--ro, req, bool, 使能-->true
    </enabled>
    <Position>
      <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
      <positionX>
        <!--ro, req, int, X坐标-->1
      </positionX>
      <positionY>
        <!--ro, req, int, Y坐标-->1
      </positionY>
      <height>
        <!--ro, req, int, 宽-->1
      </height>
      <width>
        <!--ro, req, int, 高-->1
      </width>
    </Position>
    <materialNo>
      <!--ro, opt, int, 素材ID-->1
    </materialNo>
  </TemperatureIcon>
  <AudioAlarm>
    <!--ro, opt, object, 音频报警信息-->
    <enabled>
      <!--ro, req, bool, 使能-->true
    </enabled>
    <materialNo>
      <!--ro, opt, int, 音频素材ID-->1
    </materialNo>
  </AudioAlarm>
  <abnormalHighTemperature>
    <!--ro, opt, float, 异常高温值, unit:℃, unitType:温度, desc:保留一位小数-->37.5
  </abnormalHighTemperature>
```

```
</Temperature>
<wearMask>
  <!--ro, opt, object, 已戴口罩提醒-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <fontSize>
    <!--ro, req, int, 字体大小-->1
  </fontSize>
  <FontColor>
    <!--ro, req, object, 字体颜色-->
    <RGB>
      <!--ro, opt, int, 十进制数据表示三原色-->1
    </RGB>
  </FontColor>
  <BackColor>
    <!--ro, req, object, 背景颜色-->
    <RGB>
      <!--ro, opt, int, 十进制数据表示三原色-->1
    </RGB>
  </BackColor>
  <Position>
    <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--ro, req, int, X坐标-->1
    </positionX>
    <positionY>
      <!--ro, req, int, Y坐标-->1
    </positionY>
    <height>
      <!--ro, req, int, 宽-->1
    </height>
    <width>
      <!--ro, req, int, 高-->1
    </width>
  </Position>
  <text>
    <!--ro, opt, string, 标题文字-->test
  </text>
  <MaskIcon>
    <!--ro, opt, object, 已戴口罩图标-->
    <enabled>
      <!--ro, req, bool, 使能-->true
    </enabled>
    <Position>
      <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
      <positionX>
        <!--ro, req, int, X坐标-->1
      </positionX>
      <positionY>
        <!--ro, req, int, Y坐标-->1
      </positionY>
      <height>
        <!--ro, req, int, 宽-->1
      </height>
      <width>
        <!--ro, req, int, 高-->1
      </width>
    </Position>
    <materialNo>
      <!--ro, opt, int, 素材ID-->1
    </materialNo>
  </MaskIcon>
</wearMask>
<NormalTemperature>
  <!--ro, opt, object, 温度正常提醒-->
  <Prompt>
    <!--ro, opt, object, 提示信息-->
    <enabled>
      <!--ro, req, bool, 使能-->true
    </enabled>
    <fontSize>
      <!--ro, req, int, 字体大小-->1
    </fontSize>
    <FontColor>
      <!--ro, req, object, 字体颜色-->
      <RGB>
        <!--ro, opt, int, 十进制数据表示三原色-->1
      </RGB>
    </FontColor>
    <BackColor>
      <!--ro, req, object, 背景颜色-->
      <RGB>
        <!--ro, opt, int, 十进制数据表示三原色-->1
      </RGB>
    </BackColor>
    <Position>
      <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
      <positionX>
        <!--ro, req, int, X坐标-->1
      </positionX>
      <positionY>
        <!--ro, req, int, Y坐标-->1
      </positionY>
    </Position>
```

```

</height>
<!--ro, req, int, 宽-->1
</height>
<width>
<!--ro, req, int, 高-->1
</width>
</Position>
<text>
<!--ro, opt, string, 标题文字-->test
</text>
</Prompt>
<TemperatureIcon>
<!--ro, opt, object, 温度正常图标-->
<enabled>
<!--ro, req, bool, 使能-->true
</enabled>
<Position>
<!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
<positionX>
<!--ro, req, int, X坐标-->1
</positionX>
<positionY>
<!--ro, req, int, Y坐标-->1
</positionY>
<height>
<!--ro, req, int, 宽-->1
</height>
<width>
<!--ro, req, int, 高-->1
</width>
</Position>
<materialNo>
<!--ro, opt, int, 素材ID-->1
</materialNo>
</TemperatureIcon>
</NormalTemperature>
<NormalStatusAudioAlarm>
<!--ro, opt, object, 正常状态下音频报警信息, desc:人员在检测时未发现上面的人数超限、未戴口罩、体温异常等事件时设备进行播报-->
<enabled>
<!--ro, req, bool, 使能-->true
</enabled>
<materialNo>
<!--ro, opt, int, 音频素材ID-->1
</materialNo>
</NormalStatusAudioAlarm>
</EventParam>
<AlarmDataSourceProperty>
<!--ro, opt, object, 数据源信息, desc:该字段能力集缺少, 仅信发组件和设备使用-->
<PictureProperty>
<!--ro, opt, object, 图片属性-->
<PictureSwitchEffect>
<!--ro, opt, enum, 图片切换效果, subType:string, [LeftInRightOut#左进右出, rightInLeftOut#右进左出]-->leftInRightOut
</PictureSwitchEffect>
<PictureScaleType>
<!--ro, opt, enum, 图片缩放类型, subType:string, [fitCenter#拉伸填充, centerCrop#截取填充, fixXY#适应宽高]-->fitCenter
</PictureScaleType>
</PictureProperty>
<AlarmDataSource>
<!--ro, opt, object, 数据源-->
<MaterialNo>
<!--ro, opt, int, 素材编号-->1
</MaterialNo>
<Name>
<!--ro, opt, string, 素材名称-->test
</Name>
<Address>
<!--ro, opt, string, 素材URL地址-->test
</Address>
<WebProperty>
<!--ro, opt, object, 网页属性-->
<webType>
<!--ro, opt, enum, 网页类型, subType:string, [desktop#桌面版, mobiLe#移动版], desc:默认按照设备的系统类型 (Windows或Android) 来显示网页-->
</webType>
</WebProperty>
</AlarmDataSource>
</AlarmDataSourceProperty>
<ConferenceParam>
<!--ro, opt, object, 会议信息, desc:当页面用于会议界面时有效-->
<backPicId>
<!--ro, opt, int, 背景图片素材ID-->1
</backPicId>
<Title>
<!--ro, opt, object, 会议标题 (主题), desc:1、会议标题 (主题) 的文字内容通过场地预约配置协议下发, 对应/ISAPI/Publish/Signage/placeReservation?format=json中的programID字段。2、可通过会议计划 (/ISAPI/InteractivePanel/Meeting/plan?format=json) 中的meetingTopic字段下发-->
<enabled>
<!--ro, req, bool, 使能-->true
</enabled>
<fontSize>
<!--ro, req, int, 字体大小-->1
</fontSize>
<fontColor>
<!--ro, req, object, 字体颜色-->

```

```

<RGB>
  <!--ro, opt, int, 十进制数据表示三原色-->1
</RGB>
</FontColor>
<BackColor>
  <!--ro, req, object, 背景颜色-->
  <RGB>
    <!--ro, opt, int, 十进制数据表示三原色-->1
  </RGB>
</BackColor>
<Position>
  <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
  <positionX>
    <!--ro, req, int, X坐标-->1
  </positionX>
  <positionY>
    <!--ro, req, int, Y坐标-->1
  </positionY>
  <height>
    <!--ro, req, int, 宽-->1
  </height>
  <width>
    <!--ro, req, int, 高-->1
  </width>
</Position>
</Title>
<VoiceBroadcast>
  <!--ro, opt, object, 语音播报-->
  <enabled>
    <!--ro, opt, bool, 是否启用语音播报-->true
  </enabled>
  <broadcastSpeed>
    <!--ro, opt, enum, 播报语速, subType:string, [sLower#慢, sLow#较慢, medium#中速, fast#较快, faster#快]-->sLower
  </broadcastSpeed>
  <inLibVoice>
    <!--ro, opt, string, 在库人员语音播报内容-->test
  </inLibVoice>
  <outLibVoice>
    <!--ro, opt, string, 未知人员语音播报内容-->test
  </outLibVoice>
</VoiceBroadcast>
<WelcomeWord>
  <!--ro, opt, object, 欢迎词-->
  <enabled>
    <!--ro, opt, bool, 是否启用欢迎词-->true
  </enabled>
  <inLibWord>
    <!--ro, opt, string, 在库人员欢迎词内容, range:[1,128]-->test
  </inLibWord>
  <outLibWord>
    <!--ro, opt, string, 未知人员欢迎词内容, range:[1,128]-->test
  </outLibWord>
</WelcomeWord>
<ShowInfo>
  <!--ro, opt, object, 展示信息-->
  <row>
    <!--ro, opt, int, 窗口内人员签到展示行数-->1
  </row>
  <column>
    <!--ro, opt, int, 窗口内人员签到展示列数-->1
  </column>
  <Position>
    <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--ro, req, int, x坐标-->1
    </positionX>
    <positionY>
      <!--ro, req, int, y坐标-->1
    </positionY>
    <height>
      <!--ro, req, int, 宽-->1
    </height>
    <width>
      <!--ro, req, int, 高-->1
    </width>
  </Position>
  <inLib>
    <!--ro, opt, object, 在库信息-->
    <nameEnabled>
      <!--ro, opt, bool, 在库人员是否展示姓名-->true
    </nameEnabled>
  </inLib>
  <defaultShowInfoEnabled>
    <!--ro, opt, bool, 是否采用人员默认展示信息, desc:下发该字段, 且为true, 表示人员展示信息采用默认值。坐标信息等不用下发-->true
  </defaultShowInfoEnabled>
</ShowInfo>
<ModuleInfoList>
  <!--ro, opt, array, 界面模块信息列表, subType:object-->
  <ModuleInfo>
    <!--ro, opt, object, 界面模块信息-->
    <moduleType>
      <!--ro, req, enum, 模块类型, subType:string, [conferenceTime#会议时间(对应timeRange), conferenceSite#会议地点(对应会议室名称
meetingName), conferenceStatus#会议状态(对应meetingStatus), signInPersonNum#已签到人数, totalPersonNum#应到人数, unSignInPersonNum#未签到人
数, signTotalPersonNum#已签到/应到人数, nfcCode#会议室二维码, meetingHostName#主持人, emailNum#参会人, customCode#自定义空码, nextMeeting#当于下一场会议信

```

meetingName), conferenceStatus#会议状态(对应meetingStatus), signInPersonNum#已签到人数, totalPersonNum#应到人数, unSignInPersonNum#未签到人数, signTotalPersonNum#已签到/应到人数, nfcCode#会议室二维码, meetingHostName#主持人, emailNum#参会人, customCode#自定义空码, nextMeeting#当于下一场会议信

[illegible]

```

</moduleType>
<enabled>
  <!--ro, req, bool, 使能-->true
</enabled>
<fontSize>
  <!--ro, req, int, 字体大小-->1
</fontSize>
<fontColor>
  <!--ro, req, object, 字体颜色-->
  <RGB>
    <!--ro, opt, int, 十进制数据表示三原色-->1
  </RGB>
</FontColor>
<BackColor>
  <!--ro, opt, object, 背景颜色-->
  <RGB>
    <!--ro, opt, int, 十进制数据表示三原色-->1
  </RGB>
</BackColor>
<backPicId>
  <!--ro, opt, int, 背景图片素材ID, desc: 背景图和背景颜色同时仅支持一种, 两者均下发, 设备优先解析背景图字段-->1
</backPicId>
<Position>
  <!--ro, opt, object, 位置, desc: 左上角为坐标原点, 取值范围和设备分辨率保持一致-->
  <positionX>
    <!--ro, req, int, X坐标-->1
  </positionX>
  <positionY>
    <!--ro, req, int, Y坐标-->1
  </positionY>
  <height>
    <!--ro, req, int, 宽-->1
  </height>
  <width>
    <!--ro, req, int, 高-->1
  </width>
</Position>
<CustomKey>
  <!--ro, opt, object, 自定义字段-->
  <key>
    <!--ro, opt, string, 自定义字段-->test
  </key>
  <associateProtocolURL>
    <!--ro, opt, string, 关联协议URL-->test
  </associateProtocolURL>
</CustomKey>
</ModuleInfo>
</ModuleInfoList>
<conferenceStatus>
  <!--ro, opt, enum, 会议状态, subType:string, [idle#空闲, signing#签到中, meeting#开会中], desc: 根据不同的会议状态, 配置不同的会议界面-->idle
</conferenceStatus>
</ConferenceParam>
<CallBtn>
  <!--ro, opt, object, 呼叫按键窗口-->
  <backPicId>
    <!--ro, opt, int, 背景图片素材ID-->1
  </backPicId>
</CallBtn>
<OpenDoorBtn>
  <!--ro, opt, object, 开门按键窗口-->
  <backPicId>
    <!--ro, opt, int, 背景图片素材ID-->1
  </backPicId>
</OpenDoorBtn>
<QRCodeBtn>
  <!--ro, opt, object, 二维码按键窗口-->
  <backPicId>
    <!--ro, opt, int, 背景图片素材ID-->1
  </backPicId>
</QRCodeBtn>
<AuthenticationSuccess>
  <!--ro, opt, object, 认证成功窗口-->
  <BackColor>
    <!--ro, opt, object, 背景颜色-->
    <RGB>
      <!--ro, opt, int, 十进制数据表示三原色-->1
    </RGB>
  </BackColor>
  <backPicId>
    <!--ro, opt, int, 背景图片素材ID, desc: 背景图和背景颜色同时仅支持一种, 两者均下发, 设备优先解析背景图字段-->1
  </backPicId>
  <SubWindowList>
    <!--ro, opt, object, 子窗口列表-->
    <SubWindow>
      <!--ro, opt, object, 子窗口-->
      <subWindowType>
        <!--ro, req, enum, 子窗口类型, subType:string, [temperature#温度, facePic#人脸图片, name#姓名, employeeNo#工号, group#组织/部门, tips#提示信息]-->temperature
      </subWindowType>
    </SubWindow>
  </SubWindowList>
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>

```

```
<fontSize>
  <!--ro, opt, int, 字体大小, unit:px, unitType:图像显示-->1
</fontSize>
<fontColor>
  <!--ro, opt, int, 字体颜色-->1
</fontColor>
<fontStyle>
  <!--ro, opt, string, 字体风格, desc:bold-加粗, italic-倾斜, underline-下划线, strikethrough-删除线-->bold,italic,underline,strikethrough
</fontStyle>
<Position>
  <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
  <positionX>
    <!--ro, req, int, x坐标-->1
  </positionX>
  <positionY>
    <!--ro, req, int, y坐标-->1
  </positionY>
  <height>
    <!--ro, req, int, 宽-->1
  </height>
  <width>
    <!--ro, req, int, 高-->1
  </width>
</Position>
</SubWindow>
</SubWindowList>
</AuthenticationSuccess>
<AuthenticationFail>
  <!--ro, opt, object, 认证失败窗口-->
  <BackColor>
    <!--ro, opt, object, 背景颜色-->
    <RGB>
      <!--ro, opt, int, 十进制数据表示三原色-->1
    </RGB>
  </BackColor>
  <backPicId>
    <!--ro, opt, int, 背景图片素材ID, desc:背景图和背景颜色同时仅支持一种,两者均下发,设备优先解析背景图字段-->1
  </backPicId>
  <SubWindowList>
    <!--ro, opt, object, 子窗口列表-->
    <SubWindow>
      <!--ro, opt, object, 子窗口-->
      <subWindowType>
        <!--ro, req, enum, 子窗口类型, subType:string, [temperature#温度,facePic#人脸图片,name#姓名,employeeNo#工号,group#组织/部门,tips#提示信息]-->temperature
      </subWindowType>
      <enabled>
        <!--ro, req, bool, 使能-->true
      </enabled>
      <fontSize>
        <!--ro, opt, int, 字体大小, unit:px, unitType:图像显示-->1
      </fontSize>
      <fontColor>
        <!--ro, opt, int, 字体颜色-->1
      </fontColor>
      <fontStyle>
        <!--ro, opt, string, 字体风格, desc:bold-加粗, italic-倾斜, underline-下划线, strikethrough-删除线-->bold,italic,underline,strikethrough
      </fontStyle>
      <Position>
        <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
        <positionX>
          <!--ro, req, int, x坐标-->1
        </positionX>
        <positionY>
          <!--ro, req, int, y坐标-->1
        </positionY>
        <height>
          <!--ro, req, int, 宽-->1
        </height>
        <width>
          <!--ro, req, int, 高-->1
        </width>
      </Position>
      <facePicID>
        <!--ro, opt, int, 默认人脸图片ID, step:1, dep:and,
        {$.PageList.Page.WindowsList[*].Windows.AuthenticationFail.SubWindowList.SubWindow.subWindowType,eq,facePic}-->1
      </facePicID>
    </SubWindow>
  </SubWindowList>
</AuthenticationFail>
<StatusBar>
  <!--ro, opt, object, 状态栏, desc:当素材类型为状态栏时有效, 且必填-->
  <backPicId>
    <!--ro, opt, int, 背景图片素材ID, desc:背景图和背景颜色同时仅支持一种,两者均下发,设备优先解析背景图字段-->1
  </backPicId>
  <statusBarType>
    <!--ro, req, enum, 状态栏类型, subType:string, [fixed#固定的], desc:当为固定类型时, 窗口位置不生效, 设备将状态栏固定在顶部显示-->fixed
  </statusBarType>
  <SubStatusList>
    <!--ro, opt, object, 子状态列表-->
    <SubStatus>
      <!--ro, opt, object, 子状态-->
    </SubStatus>
  </SubStatusList>
```

```
<!--ro, req, enum, 子状态, subType:string, [alarm#报警状态(3040寺),wifi#wifi状态,tv1#客厅状态,wirenetwork#有线网络状态,deploy#布防状态,temperature#测温]-->dial
</statusType>
<enabled>
<!--ro, req, bool, 是否显示-->>true
</enabled>
</SubStatus>
</SubStatusList>
</StatusBar>
</Windows>
</WindowsList>
</Page>
</PageList>
<programSize>
<!--ro, opt, int, 节目大小, unit:Byte, unitType:信息量-->1
</programSize>
<programLength>
<!--ro, opt, int, 节目时长, unit:s, unitType:时间-->1
</programLength>
<coordinateType>
<!--ro, opt, enum, 节目中坐标基准值类型, subType:string, [uniformCoordinate#统一坐标(实际的分辨率坐标按基准坐标等比例进行转换),resolutionCoordinate#实际分辨率坐标]-->uniformCoordinate
</coordinateType>
<screenType>
<!--ro, opt, enum, 屏幕类型, subType:string, [first#主屏,second#辅屏], desc:当programType为doubleSide表示创建双屏节目时生效-->first
</screenType>
<audioEnabled>
<!--ro, opt, bool, 节目音频使能, desc:当programType为doubleSide表示创建双屏节目时生效,用来防止主屏声音和辅屏声音混淆,主辅屏节目的音频只开一个-->true
</audioEnabled>
<PageIdList>
<!--ro, opt, array, 页面id列表, subType:object-->
<PageId>
<!--ro, opt, object, 页面id-->
<id>
<!--ro, req, int, 页面ID-->1
</id>
</PageId>
</PageIdList>
<programUUID>
<!--ro, opt, string, 节目UUID, range:[1,32], desc:平台下发-->test
</programUUID>
</Program>
```

69.66.8 获取全部节目参数

Request URL

GET /ISAPI/Publish/ProgramMgr/program

查询参数

无

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>
<ProgramList xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
<!--ro, req, array, 节目列表, subType:object, attr:version(req, string, 协议版本)-->
<Program>
<!--ro, req, object, 节目信息-->
<id>
<!--ro, req, int, 节目索引-->1
</id>
<programName>
<!--ro, req, string, 节目名称-->test
</programName>
<programRemarks>
<!--ro, req, string, 节目描述-->test
</programRemarks>
<shareProperty>
<!--ro, opt, enum, 共享属性, subType:string, [public#公有,private#私有]-->public
</shareProperty>
<approveState>
<!--ro, opt, enum, 审核状态标志, subType:string, [approved#审核通过,notPass#未通过,notApprove#未审核]-->approved
</approveState>
<approveRemarks>
<!--ro, opt, string, 审核意见-->test
</approveRemarks>
<programType>
<!--ro, opt, enum, 节目类型, subType:string, [normal#普通,decode#解码,touch#触摸,decodeTouch#解码触摸,doubleSide#双面屏,conference#会议,screensaver#图片/屏保(门禁),character#欢迎词(门禁),mainInterface#主界面(门禁),txt#文本节目(门禁),advertise#广告], desc:当节目类型为conference#会议,screensaver#图片/屏保(门禁),character#欢迎词(门禁),mainInterface#主界面(门禁),txt#文本节目(门禁),advertise#广告时,该节目不依赖与日程,直接单独下发该节目后,在设备上展示该节目下的页面信息.-->normal
</programType>
```

```
<orgNo>
  <!--ro, opt, int, 所属组织编号-->1
</orgNo>
<Resolution>
  <!--ro, req, object, 分辨率-->
  <resolutionName>
    <!--ro, opt, string, 分辨率名称-->test
  </resolutionName>
  <imageWidth>
    <!--ro, req, int, 分辨率宽-->1
  </imageWidth>
  <imageHeight>
    <!--ro, req, int, 分辨率高-->1
  </imageHeight>
</Resolution>
<PageList>
  <!--ro, opt, object, 页面列表-->
  <Page>
    <!--ro, req, object, 页面-->
    <id>
      <!--ro, req, int, 索引-->1
    </id>
    <PageBasicInfo>
      <!--ro, req, object, 页面基本信息-->
      <pageName>
        <!--ro, opt, string, 页面名称-->test
      </pageName>
      <BackgroundColor>
        <!--ro, opt, object, 背景颜色-->
        <RGB>
          <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
        </RGB>
        </BackgroundColor>
        <playDurationMode>
          <!--ro, opt, enum, 页面播放时间模式, subType:string, [selfDefine#自定义,auto#自动,loop#循环], desc:自定义时playDuration节点有效;自动时根据播放
          内容的时间*playCount计算;循环时表示一直循环-->selfDefine
        </playDurationMode>
        <playDuration>
          <!--ro, opt, int, 页面播放时长, unit:s, unitType:时间-->1
        </playDuration>
        <playCount>
          <!--ro, opt, int, 页面播放次数-->1
        </playCount>
        <switchDuration>
          <!--ro, opt, int, 切换间隔, unit:s, unitType:时间-->1
        </switchDuration>
        <switchEffect>
          <!--ro, opt, enum, 切换效果, subType:string, [none#无,random#随机,boxShrink#盒状收缩,boxSpread#盒状展开,cycleShrink#圆形收缩,cycSpread#圆形展
          开,eraseUp#向上擦除,eraseDown#向下擦除,eraseLeft#向左擦除,eraseRight#向右擦除,verticalShelter#垂直屏蔽,horizontalShelter#水平屏蔽,verticalChessboard#纵向棋
          盘式,horizontalChessboard#横向棋盘式,dissolve#随机溶解,LeftRightToCenter#左右向中间缩进,ceterToLeftRight#中央向左右扩展,upDownToCenter#上下向中央缩
          进,centerToUpDown#中央向上下扩展,drawOutLeftDown#从左下抽出,drawOutLeftUp#从左上抽出,drawOutRightDown#从右下抽出,drawOutRightUp#从右上抽出,verticalLine#随机
          垂直线条,horizontalLine#随机垂直线条]-->none
        </switchEffect>
        <backgroundPic>
          <!--ro, opt, int, 背景图片素材ID-->1
        </backgroundPic>
        <SignInCfg>
          <!--ro, opt, object, 签到信息, desc:当前用于签到或考勤界面时有效-->
          <VoiceBroadcast>
            <!--ro, opt, object, 语音播报-->
            <enabled>
              <!--ro, opt, bool, 是否启用语音播报-->true
            </enabled>
            <broadcastSpeed>
              <!--ro, opt, enum, 播报语速, subType:string, [slower#慢,slow#较慢,medium#中速,fast#较快,faster#快]-->slower
            </broadcastSpeed>
            <inLibVoice>
              <!--ro, opt, string, 在库人员正常体温语音播报内容-->test
            </inLibVoice>
            <outLibVoice>
              <!--ro, opt, string, 未知人员正常体温语音播报内容-->test
            </outLibVoice>
            <inLibAbnormalTemperatureVoice>
              <!--ro, opt, string, 在库人员异常温度语音播报内容-->test
            </inLibAbnormalTemperatureVoice>
            <outLibAbnormalTemperatureVoice>
              <!--ro, opt, string, 未知人员异常温度语音播报内容-->test
            </outLibAbnormalTemperatureVoice>
          </VoiceBroadcast>
          <WelcomeWord>
            <!--ro, opt, object, 欢迎词-->
            <enabled>
              <!--ro, opt, bool, 是否启用欢迎词-->true
            </enabled>
            <inLibWord>
              <!--ro, opt, string, 在库人员欢迎词内容-->test
            </inLibWord>
            <outLibWord>
              <!--ro, opt, string, 未知人员欢迎词内容-->test
            </outLibWord>
          </WelcomeWord>
          <SignCartoon>
            <!--ro, opt, object, 签到动画-->
```

```

<signCartoonSpeed>
  <!--ro, opt, enum, 动画速度, subType:string, [fast#快,medium#终端,slow#慢]-->fast
</signCartoonSpeed>
</SignCartoon>
<backgroundSignEnabled>
  <!--ro, opt, bool, 是否启用后台签到-->true
</backgroundSignEnabled>
<TemperatureDetection>
  <!--ro, opt, object, 温度监测参数-->
  <enabled>
    <!--ro, req, bool, 是否启用-->true
  </enabled>
  <threshold>
    <!--ro, req, float, 正常温度值-->0.000
  </threshold>
</TemperatureDetection>
</SignInCfg>
</PageBasicInfo>
<characterMode>
  <!--ro, opt, enum, 页面中的欢迎词模式, subType:string, [mode1#模式1,mode2#模式2,mode3#模式3], desc:门禁增加, 门禁欢迎词节目位置固定, 且存在三种模式-->mode1
  </characterMode>
<WindowsList>
  <!--ro, opt, array, 窗口信息列表, subType:object-->
  <Windows>
    <!--ro, opt, object, 窗口信息-->
    <id>
      <!--ro, req, int, 内容编号-->1
    </id>
    <Position>
      <!--ro, opt, object, 内容位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
      <positionX>
        <!--ro, req, int, x坐标-->1
      </positionX>
      <positionY>
        <!--ro, req, int, y坐标-->1
      </positionY>
      <height>
        <!--ro, req, int, 宽-->1
      </height>
      <width>
        <!--ro, req, int, 高-->1
      </width>
    </Position>
    <CornerRadius>
      <!--ro, opt, object, 圆角配置, desc:目前仅对picture图片窗口和character文字窗口有效-->
      <TopLeft>
        <!--ro, opt, int, 左上角半径-->1
      </TopLeft>
      <TopRight>
        <!--ro, opt, int, 右上角半径-->1
      </TopRight>
      <BottomLeft>
        <!--ro, opt, int, 左下角半径-->1
      </BottomLeft>
      <BottomRight>
        <!--ro, opt, int, 右下角半径-->1
      </BottomRight>
    </CornerRadius>
    <layerNo>
      <!--ro, req, int, 图层编号-->1
    </layerNo>
    <WinMaterialInfo>
      <!--ro, opt, object, 窗口素材信息-->
      <materialType>
        <!--ro, req, enum, 素材类型, subType:string, [static#静态,dynamic#动态,other#其他]-->static
      </materialType>
      <staticMaterialType>
        <!--ro, opt, enum, 静态素材类型, subType:string, [picture#图片,flash#flash,audio#音频,video#视频,document#文本,ppt#ppt,doc#文档,excel#表格,pdf#pdf,web#网页,app#app,signIn#签到,conference#会议]-->picture
      </staticMaterialType>
      <dynamicType>
        <!--ro, opt, enum, 动态素材类型, subType:string,
[web#web,socket#socket,rss#rss,realStream#realStream,call#call,dynamicPic#dynamicPic,capturePic#capturePic,character#character]-->web
      </dynamicType>
      <otherType>
        <!--ro, opt, enum, 其他素材类型, subType:string, [cLock#时钟,weather#天气,countdown#倒计时,localInput#本地输入,hyperLinkBtn#超链接按钮,event#事件,callBtn#呼叫按键,openDoorBtn#开门按键,QRCODEBtn#二维码按键,authenticationSuccess#认证成功,authenticationFail#认证失败,statusBar#状态栏]-->cLock
      </otherType>
    </WinMaterialInfo>
    <TouchProperty>
      <!--ro, opt, object, 触摸属性-->
      <windType>
        <!--ro, opt, enum, 窗口类型, subType:string, [popup#弹窗,page#页面]-->popup
      </windType>
      <hyperLinkType>
        <!--ro, opt, enum, 超链接类型, subType:string, [window#页面超链接,page#窗口超链接], desc:窗口类型为弹窗时有效-->window
      </hyperLinkType>
      <windowId>
        <!--ro, opt, int, 窗口编号, desc:当前页面的窗口,超链接类型为窗口时有效-->1
      </windowId>
      <pageId>
        <!--ro, opt, int, 页面编号, desc:超链接类型为页面时有效-->1
      </pageId>
    </TouchProperty>
  </Windows>
</WindowsList>

```

```
</pageId>
</TouchProperty>
<PlayItemList>
  <!--ro, opt, array, 窗口播放列表, subType:object-->
  <PlayItem>
    <!--ro, req, object, 窗口播放信息-->
    <id>
      <!--ro, req, int, 播放序号-->1
    </id>
    <materialNo>
      <!--ro, opt, int, 素材编号-->1
    </materialNo>
    <inputChannel>
      <!--ro, opt, int, 绑定的IPC通道号-->1
    </inputChannel>
    <playEffect>
      <!--ro, req, enum, 播放效果, subType:string, [none#无,marquee#跑马灯]-->none
    </playEffect>
    <MarqueeInfo>
      <!--ro, opt, object, 跑马灯信息-->
      <scrollType>
        <!--ro, req, enum, 跑马灯滚动类型, subType:string, [none#不滚动,Loops#循环滚动,once#滚动一次,backAndForth#来回滚动]-->none
      </scrollType>
      <scrollDeriction>
        <!--ro, req, enum, 跑马灯滚动方向, subType:string, [none#无,up#向上,down#向下,Left#向左,right#向右]-->none
      </scrollDeriction>
      <scrollSpeed>
        <!--ro, req, int, 跑马灯滚动速度-->1
      </scrollSpeed>
    </MarqueeInfo>
    <PlayDuration>
      <!--ro, opt, object, 窗口内素材播放时间, desc:静态素材可以配置该时间,视频直播和IPC通道都可配置该节点-->
      <durationType>
        <!--ro, req, enum, 持续方式, subType:string, [materialTime#按素材时间,selfDefine#自定义]-->selfDefine
      </durationType>
      <duration>
        <!--ro, req, int, 持续时间, unit:s, unitType:时间-->1
      </duration>
    </PlayDuration>
    <CharactersEffect>
      <!--ro, opt, object, 文字显示效果, desc:当素材类型为文本txt和时有效-->
      <fontSize>
        <!--ro, req, int, 字体大小-->1
      </fontSize>
      <FontColor>
        <!--ro, req, object, 字体颜色-->
        <RGB>
          <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
        </RGB>
        </FontColor>
      <backgroundType>
        <!--ro, opt, enum, 背景类型, subType:string, [color#颜色（默认）,picture#图片], desc:字段不存在时,默认为颜色-->color
      </backgroundType>
      <backPicId>
        <!--ro, opt, int, 背景图片素材ID-->1
      </backPicId>
      <BackColor>
        <!--ro, req, object, 背景颜色-->
        <RGB>
          <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
        </RGB>
        </BackColor>
      <backTransparent>
        <!--ro, req, int, 背景透明度-->1
      </backTransparent>
      <subtitlesEnabled>
        <!--ro, req, bool, 子标题使能-->true
      </subtitlesEnabled>
      <scrollDirection>
        <!--ro, req, enum, 文字滚动方向, subType:string, [Left#向左,right#向右,up#向上,down#向下]-->left
      </scrollDirection>
      <scrollSpeed>
        <!--ro, req, int, 文字滚动速度-->1
      </scrollSpeed>
    </CharactersEffect>
    <switchEffect>
      <!--ro, opt, enum, 窗口素材切换效果, subType:string, [none#无,LeftInRightOut#左进右出,rightInLeftOut#右进左出,bottomInTopOut#下进上出,topInBottomOut#上进下出,fadeInFadeOut#淡入淡出,middleExit#中间退出,topPop#顶部弹出,rightBottomIn#右下角进入,LeftTopIn#左上角进入,horizontalOpen#水平打开,verticalOpen#垂直打开,random#随机效果], desc:图片素材时有效-->random
    </switchEffect>
    <pageTime>
      <!--ro, opt, int, 翻页时间, unit:s, unitType:时间, desc:当素材为word,ppt,pdf,excel时有效-->1
    </pageTime>
    <scrollSpeed>
      <!--ro, req, int, 滚动速度, desc:当素材为静态web时有效-->1
    </scrollSpeed>
    <scaleType>
      <!--ro, opt, enum, 素材缩放参数配置, subType:string, [fitXY#拉伸填充,fitCenter#自适应,centerCrop#截取填充], desc:目前仅当图片窗口有效-->centerCrop
    </scaleType>
    <playType>
      <!--ro, opt, enum, 播放方式, subType:string, [normal#正常,faceAttendance#考勤,PeopleCounting#客流,temperature#测温], desc:目前仅当图片窗口有效-->normal
    </playType>
  </PlayItem>
</PlayItemList>
```

```

</playType>
<CharactersAttribute>
  <!--ro, opt, object, 文字属性, desc:当dynamicType为character时有效-->
  <fontSize>
    <!--ro, req, int, 字体大小-->1
  </fontSize>
  <fontColor>
    <!--ro, req, object, 字体颜色-->
    <RGB>
      <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
    </RGB>
  </fontColor>
  <backgroundType>
    <!--ro, req, enum, 背景类型, subType:string, [color#颜色,picture#图片]-->color
  </backgroundType>
  <BackColor>
    <!--ro, req, object, 背景色-->
    <RGB>
      <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
    </RGB>
  </BackColor>
  <backTransparent>
    <!--ro, opt, int, 背景透明度-->1
  </backTransparent>
  <alignType>
    <!--ro, req, enum, 对齐方式, subType:string, [Left#左,right#右,middle#中]-->middle
  </alignType>
  <verticalAlignType>
    <!--ro, req, enum, 垂直对齐方式, subType:string, [top#顶,bottom#底,verticalCenter#中]-->top
  </verticalAlignType>
  <characterContent>
    <!--ro, opt, string, 文字内容, range:[0,512]-->test
  </characterContent>
  <backPicId>
    <!--ro, opt, int, 背景图片素材ID-->1
  </backPicId>
  <characterStyle>
    <!--ro, req, enum, 字体效果, subType:string, [normal#常规,bold#粗体]-->normal
  </characterStyle>
  <characterType>
    <!--ro, req, enum, 字体类型, subType:string, [mainTitle#主标题,subTitle#副标题]-->mainTitle
  </characterType>
</CharactersAttribute>
<WebProperty>
  <!--ro, opt, object, 网页属性-->
  <webType>
    <!--ro, opt, enum, 网页类型, subType:string, [desktop#桌面版,mobile#移动版], desc:默认按照设备的系统类型(windows或Android)来显示网页-->desktop
  </webType>
</WebProperty>
</PlayItem>
</PlayItemList>
<enabledAudio>
  <!--ro, opt, bool, 音频使能-->true
</enabledAudio>
<enableHide>
  <!--ro, opt, bool, 隐藏使能-->true
</enableHide>
<enableLock>
  <!--ro, opt, bool, 锁定使能-->true
</enableLock>
<AppWindow>
  <!--ro, opt, object, App窗口信息-->
  <WindowInfoList>
    <!--ro, req, array, 窗口信息列表, subType:object-->
    <WindowInfo>
      <!--ro, req, object, 窗口信息-->
      <id>
        <!--ro, req, int, 索引-->1
      </id>
      <materialNo>
        <!--ro, req, int, 素材编号, desc:采用本字段下发时,用户需要先将App安装包当做素材存储到设备中,得到素材编号再进行下发-->1
      </materialNo>
      <path>
        <!--ro, opt, string, App路径, range:[0,1024],
desc:path和materialNo字段同时存在时,设备优先解析path字段;
采用本字段下发时,设备需要确保App已经安装在设备内,直接传入App的路径即可;
对于Android设备,path传的是App的包名;对于Windows设备,path传的是App的应用程序绝对路径; -->test
      </path>
    </WindowInfo>
  </WindowInfoList>
</AppWindow>
<DataSource>
  <!--ro, opt, object, 数据来源, desc:当窗口为叫号和动态弹图时有效-->
  <materialNo>
    <!--ro, req, int, 素材编号-->1
  </materialNo>
</DataSource>
<Call>
  <!--ro, opt, object, 叫号数据-->
  <materialNo>
    <!--ro, req, int, 素材编号-->1
  </materialNo>
</tableRow>

```

```

<!--ro, req, int, 表格行-->1
</tableRow>
<tableColumn>
  <!--ro, req, int, 表格列-->1
</tableColumn>
<tableDirection>
  <!--ro, req, enum, 表格方向, subType:string, [vertical#纵向,horizontal#纵向]-->vertical
</tableDirection>
<tableType>
  <!--ro, req, enum, 表格类型, subType:string, [template1#模板1,template2#模板2,template3#模板3,template4#模板4,template5#模板5,template6#模板
6]-->template1
</tableType>
<backPicId>
  <!--ro, opt, int, 背景图片素材编号-->1
</backPicId>
<alignType>
  <!--ro, req, enum, 对齐方式, subType:string, [left#左,right#右,middle#中]-->left
</alignType>
<refreshDirection>
  <!--ro, req, enum, 刷新方向, subType:string, [upTdown#从上到下,downToup#从下到上,LeftToright#从左到右,rightToLeft#从右到左]-->upTdown
</refreshDirection>
<HeadDataList>
  <!--ro, opt, array, 表头数据列表, subType:object-->
  <HeadData>
    <!--ro, opt, object, 表头数据-->
    <id>
      <!--ro, req, int, 索引-->1
    </id>
    <data>
      <!--ro, req, string, 数据-->test
    </data>
  </HeadData>
</HeadDataList>
<ItemStyleList>
  <!--ro, opt, array, 表格列或行样式列表, subType:object-->
  <ItemStyle>
    <!--ro, opt, object, 表格列或行样式-->
    <id>
      <!--ro, req, int, 索引-->1
    </id>
    <width>
      <!--ro, req, int, 每列宽度所占百分比-->1
    </width>
    <fontSize>
      <!--ro, req, int, 字体大小-->1
    </fontSize>
    <FontColor>
      <!--ro, req, object, 字体颜色-->
      <RGB>
        <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
      </RGB>
    </FontColor>
    <BackColor>
      <!--ro, req, object, 背景色-->
      <RGB>
        <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
      </RGB>
    </BackColor>
  </ItemStyle>
</ItemStyleList>
</Call>
<DynamicPic>
  <!--ro, opt, object, 动态弹图窗口配置-->
  <materialNo>
    <!--ro, req, int, 素材编号-->1
  </materialNo>
  <backPicId>
    <!--ro, opt, int, 背景图片素材ID-->1
  </backPicId>
</DynamicPic>
<CapturePic>
  <!--ro, opt, object, 抓拍图片参数-->
  <backPicId>
    <!--ro, opt, int, 背景图片素材ID-->1
  </backPicId>
  <ipcMaterialNo>
    <!--ro, req, int, IPC素材编号-->1
  </ipcMaterialNo>
  <cancelType>
    <!--ro, req, enum, 取消方式, subType:string, [auto#自动>manual#手动]-->>manual
  </cancelType>
  <duration>
    <!--ro, opt, int, 持续时间, unit:s, unitType:时间-->1
  </duration>
</CapturePic>
<ClockParam>
  <!--ro, opt, object, 时钟参数-->
  <backPicId>
    <!--ro, opt, int, 背景图片素材ID-->1
  </backPicId>
  <ClockIcon>
    <!--ro, req, object, 时钟图标参数-->
    <enabled>

```

```
<!--ro, req, bool, 使能-->true
</enabled>
<type>
<!--ro, opt, enum, 时钟类型, subType:string, [clock1#clock1,clock2#clock2]-->clock1
</type>
<Position>
<!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
<positionX>
<!--ro, req, int, x坐标-->1
</positionX>
<positionY>
<!--ro, req, int, y坐标-->1
</positionY>
<height>
<!--ro, req, int, 宽-->1
</height>
<width>
<!--ro, req, int, 高-->1
</width>
</Position>
</ClockIcon>
<YmdParam>
<!--ro, req, object, 时钟年月日参数-->
<enabled>
<!--ro, req, bool, 使能-->true
</enabled>
<fontSize>
<!--ro, req, int, 字体大小-->1
</fontSize>
<fontColor>
<!--ro, req, object, 字体颜色-->
<RGB>
<!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
</RGB>
</FontColor>
<BackColor>
<!--ro, req, object, 背景色-->
<RGB>
<!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
</RGB>
</BackColor>
<Position>
<!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
<positionX>
<!--ro, req, int, x坐标-->1
</positionX>
<positionY>
<!--ro, req, int, y坐标-->1
</positionY>
<height>
<!--ro, req, int, 宽-->1
</height>
<width>
<!--ro, req, int, 高-->1
</width>
</Position>
</YmdParam>
<HmsParam>
<!--ro, req, object, 时钟时分秒参数-->
<enabled>
<!--ro, req, bool, 使能-->true
</enabled>
<fontSize>
<!--ro, req, int, 字体大小-->1
</fontSize>
<fontColor>
<!--ro, req, object, 字体颜色-->
<RGB>
<!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
</RGB>
</FontColor>
<BackColor>
<!--ro, req, object, 背景色-->
<RGB>
<!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
</RGB>
</BackColor>
<Position>
<!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
<positionX>
<!--ro, req, int, x坐标-->1
</positionX>
<positionY>
<!--ro, req, int, y坐标-->1
</positionY>
<height>
<!--ro, req, int, 宽-->1
</height>
<width>
<!--ro, req, int, 高-->1
</width>
</Position>
</HmsParam>
<ClockParam>
```

```
<!-->
<!--ro, req, object, 时钟星期参数-->
<enabled>
  <!--ro, req, bool, 使能-->true
</enabled>
<fontSize>
  <!--ro, req, int, 字体大小-->1
</fontSize>
<fontColor>
  <!--ro, req, object, 字体颜色-->
  <RGB>
    <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
  </RGB>
</fontColor>
<BackColor>
  <!--ro, req, object, 背景色-->
  <RGB>
    <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
  </RGB>
</BackColor>
<Position>
  <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
  <positionX>
    <!--ro, req, int, x坐标-->1
  </positionX>
  <positionY>
    <!--ro, req, int, y坐标-->1
  </positionY>
  <height>
    <!--ro, req, int, 宽-->1
  </height>
  <width>
    <!--ro, req, int, 高-->1
  </width>
</Position>
</WeekParam>
</ClockParam>
<WeatherParam>
  <!--ro, opt, object, 天气参数-->
  <backPicId>
    <!--ro, opt, int, 天气背景图片素材id-->1
  </backPicId>
  <WeatherIcon>
    <!--ro, opt, object, 天气图标参数-->
    <enabled>
      <!--ro, req, bool, 使能-->true
    </enabled>
    <Position>
      <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
      <positionX>
        <!--ro, req, int, x坐标-->1
      </positionX>
      <positionY>
        <!--ro, req, int, y坐标-->1
      </positionY>
      <height>
        <!--ro, req, int, 宽-->1
      </height>
      <width>
        <!--ro, req, int, 高-->1
      </width>
    </Position>
  </WeatherIcon>
  <Date>
    <!--ro, opt, object, 天气日期参数-->
    <enabled>
      <!--ro, req, bool, 使能-->true
    </enabled>
    <fontSize>
      <!--ro, req, int, 字体大小-->1
    </fontSize>
    <fontColor>
      <!--ro, req, object, 字体颜色-->
      <RGB>
        <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
      </RGB>
    </fontColor>
    <BackColor>
      <!--ro, req, object, 背景色-->
      <RGB>
        <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
      </RGB>
    </BackColor>
    <Position>
      <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
      <positionX>
        <!--ro, req, int, x坐标-->1
      </positionX>
      <positionY>
        <!--ro, req, int, y坐标-->1
      </positionY>
      <height>
        <!--ro, req, int, 宽-->1
      </height>
```

```
<width>
  <!--ro, req, int, 高-->1
</width>
</Position>
</Date>
<Temperature>
  <!--ro, opt, object, 天气温度参数-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <fontSize>
    <!--ro, req, int, 字体大小-->1
  </fontSize>
  <FontColor>
    <!--ro, req, object, 字体颜色-->
    <RGB>
      <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
    </RGB>
  </FontColor>
  <BackColor>
    <!--ro, req, object, 背景色-->
    <RGB>
      <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
    </RGB>
  </BackColor>
  <Position>
    <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--ro, req, int, x坐标-->1
    </positionX>
    <positionY>
      <!--ro, req, int, y坐标-->1
    </positionY>
    <height>
      <!--ro, req, int, 宽-->1
    </height>
    <width>
      <!--ro, req, int, 高-->1
    </width>
  </Position>
</Temperature>
<WeatherContent>
  <!--ro, opt, object, 天气情况参数-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <fontSize>
    <!--ro, req, int, 字体大小-->1
  </fontSize>
  <FontColor>
    <!--ro, req, object, 字体颜色-->
    <RGB>
      <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
    </RGB>
  </FontColor>
  <BackColor>
    <!--ro, req, object, 背景色-->
    <RGB>
      <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
    </RGB>
  </BackColor>
  <Position>
    <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--ro, req, int, x坐标-->1
    </positionX>
    <positionY>
      <!--ro, req, int, y坐标-->1
    </positionY>
    <height>
      <!--ro, req, int, 宽-->1
    </height>
    <width>
      <!--ro, req, int, 高-->1
    </width>
  </Position>
</WeatherContent>
<City>
  <!--ro, opt, object, 天气城市参数-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <cityId>
    <!--ro, req, string, 城市编号-->test
  </cityId>
  <cityName>
    <!--ro, req, string, 城市名称-->test
  </cityName>
  <fontSize>
    <!--ro, req, int, 字体大小-->1
  </fontSize>
  <FontColor>
    <!--ro, req, object, 字体颜色-->
```

```
<RGB>
  <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
</RGB>
</FontColor>
<BackColor>
  <!--ro, req, object, 背景色-->
  <RGB>
    <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
  </RGB>
</BackColor>
<Position>
  <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
  <positionX>
    <!--ro, req, int, x坐标-->1
  </positionX>
  <positionY>
    <!--ro, req, int, y坐标-->1
  </positionY>
  <height>
    <!--ro, req, int, 宽-->1
  </height>
  <width>
    <!--ro, req, int, 高-->1
  </width>
</Position>
</City>
<Humidity>
  <!--ro, opt, object, 天气湿度参数-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <fontSize>
    <!--ro, req, int, 字体大小-->1
  </fontSize>
  <FontColor>
    <!--ro, req, object, 字体颜色-->
    <RGB>
      <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
    </RGB>
  </FontColor>
  <BackColor>
    <!--ro, req, object, 背景色-->
    <RGB>
      <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
    </RGB>
  </BackColor>
  <Position>
    <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--ro, req, int, x坐标-->1
    </positionX>
    <positionY>
      <!--ro, req, int, y坐标-->1
    </positionY>
    <height>
      <!--ro, req, int, 宽-->1
    </height>
    <width>
      <!--ro, req, int, 高-->1
    </width>
  </Position>
</Humidity>
<AirQuality>
  <!--ro, opt, object, 天气空气质量参数-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <fontSize>
    <!--ro, req, int, 字体大小-->1
  </fontSize>
  <FontColor>
    <!--ro, req, object, 字体颜色-->
    <RGB>
      <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
    </RGB>
  </FontColor>
  <BackColor>
    <!--ro, req, object, 背景色-->
    <RGB>
      <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
    </RGB>
  </BackColor>
  <Position>
    <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--ro, req, int, x坐标-->1
    </positionX>
    <positionY>
      <!--ro, req, int, y坐标-->1
    </positionY>
    <height>
      <!--ro, req, int, 宽-->1
    </height>
    <width>
```

```

        <!--ro, req, int, 高-->1
    </width>
</Position>
</AirQuality>
<UpdateTime>
    <!--ro, opt, object, 天气更新时间参数-->
    <enabled>
        <!--ro, req, bool, 使能-->true
    </enabled>
    <refreshTime>
        <!--ro, req, string, 刷新时间, desc:设备早期返回时间格式有误, 错为: 20200826T051500+08, 为保证XSD校验通过, 改为string类型-->20200826T051500+08
    </refreshTime>
    <updateInterval>
        <!--ro, req, int, 刷新间隔, unit:minute-->1
    </updateInterval>
    <fontSize>
        <!--ro, req, int, 字体大小-->1
    </fontSize>
    <fontColor>
        <!--ro, req, object, 字体颜色-->
        <RGB>
            <!--ro, opt, int, RGB值, desc:十进制数据表示三原色, 如16777215表示0xFFFFFF-->1
        </RGB>
    </FontColor>
    <BackColor>
        <!--ro, req, object, 背景色-->
        <RGB>
            <!--ro, opt, int, RGB值, desc:十进制数据表示三原色, 如16777215表示0xFFFFFF-->1
        </RGB>
    </BackColor>
    <Position>
        <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
        <positionX>
            <!--ro, req, int, x坐标-->1
        </positionX>
        <positionY>
            <!--ro, req, int, y坐标-->1
        </positionY>
        <height>
            <!--ro, req, int, 宽-->1
        </height>
        <width>
            <!--ro, req, int, 高-->1
        </width>
    </Position>
</UpdateTime>
<Wind>
    <!--ro, opt, object, 天气风力参数-->
    <enabled>
        <!--ro, req, bool, 使能-->true
    </enabled>
    <fontSize>
        <!--ro, req, int, 字体大小-->1
    </fontSize>
    <fontColor>
        <!--ro, req, object, 字体颜色-->
        <RGB>
            <!--ro, opt, int, RGB值, desc:十进制数据表示三原色, 如16777215表示0xFFFFFF-->1
        </RGB>
    </FontColor>
    <BackColor>
        <!--ro, req, object, 背景色-->
        <RGB>
            <!--ro, opt, int, RGB值, desc:十进制数据表示三原色, 如16777215表示0xFFFFFF-->1
        </RGB>
    </BackColor>
    <Position>
        <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
        <positionX>
            <!--ro, req, int, x坐标-->1
        </positionX>
        <positionY>
            <!--ro, req, int, y坐标-->1
        </positionY>
        <height>
            <!--ro, req, int, 宽-->1
        </height>
        <width>
            <!--ro, req, int, 高-->1
        </width>
    </Position>
</Wind>
</WeatherParam>
<Countdown>
    <!--ro, opt, object, 倒计时素材-->
    <endTime>
        <!--ro, req, string, 倒计时时间点, desc:设备早期返回时间格式有误, 错为: 20200826T051500+08, 为保证XSD校验通过, 改为string类型-->20200826T051500+08
    </endTime>
    <template>
        <!--ro, req, enum, 模板类型, subType:string, [template1#模板1, template2#模板2, openingActivity1#活动开幕1, openingActivity2#活动开幕2, commercialActivity1#商业活动1, commercialActivity2#商业活动2, collegeEntranceExam#高考, overseasTimeFormat#海外时间格式]-->template1
    </template>

```

```
</template>
<timeUnit>
  <!--ro, req, enum, 时间单位, subType:string, [year#年,month#月,day#日,week#周,hour#时,minute#分,second#秒]-->year
</timeUnit>
<backPicId>
  <!--ro, opt, int, 背景图片素材ID-->1
</backPicId>
<TimeFontCfg>
  <!--ro, opt, object, 时间字体参数-->
  <fontSize>
    <!--ro, req, int, 字体大小-->1
  </fontSize>
  <fontColor>
    <!--ro, req, object, 字体颜色-->
    <RGB>
      <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
    </RGB>
  </FontColor>
  <Position>
    <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--ro, req, int, x坐标-->1
    </positionX>
    <positionY>
      <!--ro, req, int, y坐标-->1
    </positionY>
    <height>
      <!--ro, req, int, 宽-->1
    </height>
    <width>
      <!--ro, req, int, 高-->1
    </width>
  </Position>
</TimeFontCfg>
<timeUnitEnabled>
  <!--ro, opt, bool, 时间单位是否有效-->true
</timeUnitEnabled>
<maxTimeUnit>
  <!--ro, req, enum, 最大时间单位, subType:string, [year#年,month#月,day#日,week#周,hour#时,minute#分,second#秒,week#周], desc:表示当前倒计时
  的最大显示单元,例如:maxTimeUnit为月,则1年3个月,对应月份值为15-->year
</maxTimeUnit>
<ModuleInfoList>
  <!--ro, opt, array, 界面模块信息列表, subType:object-->
  <ModuleInfo>
    <!--ro, opt, object, 界面模块信息-->
    <moduleType>
      <!--ro, req, enum, 模块类型, subType:string, [year#年,month#月,day#日,hour#时,minute#分,second#秒,week#周,text#文字], desc:text为时间旁
      边的文字,例如年下面的文字"年"-->second
    </moduleType>
    <enabled>
      <!--ro, req, bool, 使能-->true
    </enabled>
    <fontSize>
      <!--ro, req, int, 字体大小-->1
    </fontSize>
    <fontColor>
      <!--ro, req, object, 字体颜色-->
      <RGB>
        <!--ro, opt, int, 十进制数据表示三原色-->1
      </RGB>
    </FontColor>
    <text>
      <!--ro, opt, string, 文字, desc:当moduleType为text时有效-->test
    </text>
    <Position>
      <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
      <positionX>
        <!--ro, req, int, X坐标-->1
      </positionX>
      <positionY>
        <!--ro, req, int, Y坐标-->1
      </positionY>
      <height>
        <!--ro, req, int, 宽-->1
      </height>
      <width>
        <!--ro, req, int, 高-->1
      </width>
    </Position>
  </ModuleInfo>
</ModuleInfoList>
</Countdown>
<localInputNo>
  <!--ro, req, enum, 本地输入类型, subType:string, [VGA#VGA,HDMI#HDMI]-->VGA
</localInputNo>
<HyperlinkBtn>
  <!--ro, opt, object, 超链接按钮-->
  <backPicId>
    <!--ro, opt, int, 背景图片素材ID-->1
  </backPicId>
</HyperlinkBtn>
<CharactersAttribute>
  <!--ro, opt, object, 文字属性, desc:当dynamicType为character时有效-->
  <fontSize>
```

```
</fontSize>
<!--ro, req, int, 字体大小-->1
</fontSize>
<FontColor>
  <!--ro, req, object, 字体颜色-->
  <RGB>
    <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
  </RGB>
</FontColor>
<fontType>
  <!--ro, opt, enum, 字体类型, subType:string, [normal#常规,bold#粗体]-->normal
</fontType>
<backgroundType>
  <!--ro, req, enum, 背景类型, subType:string, [color#颜色,picture#图片]-->color
</backgroundType>
<BackColor>
  <!--ro, req, object, 背景色-->
  <RGB>
    <!--ro, opt, int, RGB值, desc:十进制数据表示三原色,如16777215表示0xFFFFFF-->1
  </RGB>
  <fillEnabled>
    <!--ro, opt, bool, 背景色填充使能, desc:默认为true, 表示背景色填充-->true
  </fillEnabled>
</BackColor>
<backTransparent>
  <!--ro, opt, int, 背景透明度-->1
</backTransparent>
<alignType>
  <!--ro, req, enum, 对齐方式, subType:string, [left#左,right#右,middle#中]-->middle
</alignType>
<verticalAlignType>
  <!--ro, req, enum, 垂直对齐方式, subType:string, [top#顶,bottom#底,verticalCenter#中]-->top
</verticalAlignType>
<characterContent>
  <!--ro, opt, string, 文字内容, range:[0,512]-->test
</characterContent>
<backPicId>
  <!--ro, opt, int, 背景图片素材ID-->1
</backPicId>
<characterStyle>
  <!--ro, req, enum, 字体效果, subType:string, [normal#常规,bold#粗体]-->normal
</characterStyle>
<characterTypes>
  <!--ro, req, enum, 字体类型, subType:string, [mainTitle#主标题,subTitle#副标题]-->mainTitle
</characterTypes>
<scrollDirection>
  <!--ro, opt, enum, 文字滚动方向, subType:string, [left#向左,right#向右,up#向上,down#向下,turnPage#翻页]-->left
</scrollDirection>
<scrollSpeed>
  <!--ro, opt, int, 文字滚动速度, range:[1,10]-->1
</scrollSpeed>
</CharactersAttribute>
<SignIn>
  <!--ro, opt, object, 人员签到展示窗口, desc:当staticMaterialType为signIn时有效-->
  <backPicId>
    <!--ro, opt, int, 背景图片素材ID-->1
  </backPicId>
  <ShowInfo>
    <!--ro, opt, object, 展示信息-->
    <row>
      <!--ro, opt, int, 窗口内人员签到展示行数-->1
    </row>
    <column>
      <!--ro, opt, int, 窗口内人员签到展示列数-->1
    </column>
    <Position>
      <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
      <positionX>
        <!--ro, req, int, x坐标-->1
      </positionX>
      <positionY>
        <!--ro, req, int, y坐标-->1
      </positionY>
      <height>
        <!--ro, req, int, 宽-->1
      </height>
      <width>
        <!--ro, req, int, 高-->1
      </width>
    </Position>
  </inLib>
  <!--ro, opt, object, 在库信息-->
  <nameEnabled>
    <!--ro, opt, bool, 在库人员是否展示姓名-->true
  </nameEnabled>
  <genderEnabled>
    <!--ro, opt, bool, 在库人员是否展示性别-->true
  </genderEnabled>
  <selfDefine1Enabled>
    <!--ro, opt, bool, 在库人员是否展示自定义字段1-->true
  </selfDefine1Enabled>
  <selfDefine2Enabled>
    <!--ro, opt, bool, 在库人员是否展示自定义字段2-->true
  </selfDefine2Enabled>
```

```
<signInTimeEnabled>
  <!--ro, opt, bool, 在库人员是否展示签到时间-->true
</signInTimeEnabled>
<temperatureEnabled>
  <!--ro, opt, bool, 在库人员是否展示温度-->true
</temperatureEnabled>
</inLib>
<outLib>
  <!--ro, opt, object, 不在库信息-->
  <nameEnabled>
    <!--ro, opt, bool, 未知人员是否展示姓名-->true
  </nameEnabled>
  <name>
    <!--ro, opt, string, 未知人员姓名-->test
  </name>
  <genderEnabled>
    <!--ro, opt, bool, 未知人员是否展示性别-->true
  </genderEnabled>
  <signInTimeEnabled>
    <!--ro, opt, bool, 未知人员是否展示签到时间-->true
  </signInTimeEnabled>
  <temperatureEnabled>
    <!--ro, opt, bool, 未知人员是否展示温度-->true
  </temperatureEnabled>
</outLib>
<showDuration>
  <!--ro, req, enum, 签到记录显示时间, subType:string, [always#不消失,10s#10s,20s#20s,30s#30s,no#不显示]-->always
</showDuration>
</ShowInfo>
<MainTitle>
  <!--ro, opt, object, 主标题-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <fontSize>
    <!--ro, req, int, 字体大小-->1
  </fontSize>
  <fontColor>
    <!--ro, req, object, 字体颜色-->
    <RGB>
      <!--ro, opt, int, 十进制数据表示三原色-->1
    </RGB>
  </FontColor>
  <Position>
    <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--ro, req, int, x坐标-->1
    </positionX>
    <positionY>
      <!--ro, req, int, y坐标-->1
    </positionY>
    <height>
      <!--ro, req, int, 宽-->1
    </height>
    <width>
      <!--ro, req, int, 高-->1
    </width>
  </Position>
  <text>
    <!--ro, opt, string, 标题文字-->test
  </text>
</MainTitle>
<SubTitle>
  <!--ro, opt, object, 副标题-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <fontSize>
    <!--ro, req, int, 字体大小-->1
  </fontSize>
  <fontColor>
    <!--ro, req, object, 字体颜色-->
    <RGB>
      <!--ro, opt, int, 十进制数据表示三原色-->1
    </RGB>
  </FontColor>
  <Position>
    <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--ro, req, int, x坐标-->1
    </positionX>
    <positionY>
      <!--ro, req, int, y坐标-->1
    </positionY>
    <height>
      <!--ro, req, int, 宽-->1
    </height>
    <width>
      <!--ro, req, int, 高-->1
    </width>
  </Position>
  <text>
    <!--ro, opt, string, 标题文字-->test
  </text>
```

```
</text>
</SubTitle>
<NormalTemperature>
  <!--ro, opt, object, 正常体温信息-->
  <PeopleNumber>
    <!--ro, opt, object, 正常体温人数信息-->
    <enabled>
      <!--ro, req, bool, 使能-->true
    </enabled>
    <fontSize>
      <!--ro, req, int, 字体大小-->1
    </fontSize>
    <FontColor>
      <!--ro, req, object, 字体颜色-->
      <RGB>
        <!--ro, opt, int, 十进制数据表示三原色-->1
      </RGB>
    </FontColor>
    <Position>
      <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
      <positionX>
        <!--ro, req, int, X坐标-->1
      </positionX>
      <positionY>
        <!--ro, req, int, Y坐标-->1
      </positionY>
      <height>
        <!--ro, req, int, 宽-->1
      </height>
      <width>
        <!--ro, req, int, 高-->1
      </width>
    </Position>
    <text>
      <!--ro, opt, string, 标题文字-->test
    </text>
  </PeopleNumber>
  <Icon>
    <!--ro, opt, object, 正常体温图标-->
    <enabled>
      <!--ro, req, bool, 使能-->true
    </enabled>
    <Position>
      <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
      <positionX>
        <!--ro, req, int, X坐标-->1
      </positionX>
      <positionY>
        <!--ro, req, int, Y坐标-->1
      </positionY>
      <height>
        <!--ro, req, int, 宽-->1
      </height>
      <width>
        <!--ro, req, int, 高-->1
      </width>
    </Position>
    <materialNo>
      <!--ro, opt, int, 素材ID-->1
    </materialNo>
  </Icon>
</NormalTemperature>
<AbnormalTemperature>
  <!--ro, opt, object, 异常体温信息-->
  <PeopleNumber>
    <!--ro, opt, object, 异常体温人数信息-->
    <enabled>
      <!--ro, req, bool, 使能-->true
    </enabled>
    <fontSize>
      <!--ro, req, int, 字体大小-->1
    </fontSize>
    <FontColor>
      <!--ro, req, object, 字体颜色-->
      <RGB>
        <!--ro, opt, int, 十进制数据表示三原色-->1
      </RGB>
    </FontColor>
    <Position>
      <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
      <positionX>
        <!--ro, req, int, X坐标-->1
      </positionX>
      <positionY>
        <!--ro, req, int, Y坐标-->1
      </positionY>
      <height>
        <!--ro, req, int, 宽-->1
      </height>
      <width>
        <!--ro, req, int, 高-->1
      </width>
    </Position>
    <text>
```

```
<!--ro, opt, string, 标题文字-->test
</text>
</PeopleNumber>
<Icon>
  <!--ro, opt, object, 异常体温图标-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <Position>
    <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--ro, req, int, X坐标-->1
    </positionX>
    <positionY>
      <!--ro, req, int, Y坐标-->1
    </positionY>
    <height>
      <!--ro, req, int, 宽-->1
    </height>
    <width>
      <!--ro, req, int, 高-->1
    </width>
  </Position>
  <materialNo>
    <!--ro, opt, int, 素材ID-->1
  </materialNo>
</Icon>
</AbnormalTemperature>
</SignIn>
<EventParam>
  <!--ro, opt, object, 事件参数-->
  <EventResourceList>
    <!--ro, opt, array, 事件源信息列表, subType:object-->
    <EventResource>
      <!--ro, opt, object, 事件源信息-->
      <ipVersion>
        <!--ro, req, enum, ip地址版本, subType:string, [v4#v4, v6#v6, dual#双]-->v4
      </ipVersion>
      <ipAddress>
        <!--ro, opt, string, ipv4地址-->test
      </ipAddress>
      <ipv6Address>
        <!--ro, opt, string, ipv6地址-->test
      </ipv6Address>
      <portNo>
        <!--ro, req, int, 端口号-->1
      </portNo>
      <userName>
        <!--ro, req, string, 用户名-->test
      </userName>
      <password>
        <!--ro, req, string, 密码-->test
      </password>
      <channels>
        <!--ro, opt, string, 通道号, desc:多选以逗号隔开-->1,2
      </channels>
    </EventResource>
  </EventResourceList>
  <backPicId>
    <!--ro, opt, int, 背景图片素材ID-->1
  </backPicId>
  <Title>
    <!--ro, opt, object, 标题-->
    <enabled>
      <!--ro, req, bool, 使能-->true
    </enabled>
    <fontSize>
      <!--ro, req, int, 字体大小-->1
    </fontSize>
    <fontColor>
      <!--ro, req, object, 字体颜色-->
      <RGB>
        <!--ro, opt, int, 十进制数据表示三原色-->1
      </RGB>
    </fontColor>
  </Title>
  <Position>
    <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--ro, req, int, X坐标-->1
    </positionX>
    <positionY>
      <!--ro, req, int, Y坐标-->1
    </positionY>
    <height>
      <!--ro, req, int, 宽-->1
    </height>
    <width>
      <!--ro, req, int, 高-->1
    </width>
  </Position>
  <text>
    <!--ro, opt, string, 标题文字-->test
  </text>
```

```
</Title>
<PeopleCounting>
  <!--ro, opt, object, 人数统计提醒-->
  <maxPeopleNumber>
    <!--ro, opt, int, 最大可进入人数-->1
  </maxPeopleNumber>
  <unavailableBackPicId>
    <!--ro, opt, int, 人员不能进入时控件背景图片id-->1
  </unavailableBackPicId>
  <supportChangeByPeople>
    <!--ro, req, enum, 支持根据人数修改样式的字段, subType:string,
[prompt#prompt,peopleIcon#peopleIcon,availablePeopleNumber#availablePeopleNumber]-->prompt
  </supportChangeByPeople>
  <Prompt>
    <!--ro, opt, object, 提示信息-->
    <enabled>
      <!--ro, req, bool, 使能-->true
    </enabled>
    <fontSize>
      <!--ro, req, int, 字体大小-->1
    </fontSize>
    <Position>
      <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
      <positionX>
        <!--ro, req, int, X坐标-->1
      </positionX>
      <positionY>
        <!--ro, req, int, Y坐标-->1
      </positionY>
      <height>
        <!--ro, req, int, 宽-->1
      </height>
      <width>
        <!--ro, req, int, 高-->1
      </width>
    </Position>
    <text>
      <!--ro, opt, string, 标题文字-->test
    </text>
    <FontColor>
      <!--ro, opt, object, 人员可以进入时的字体颜色-->
      <RGB>
        <!--ro, opt, int, 十进制数据表示三原色-->1
      </RGB>
    </FontColor>
    <unavailableText>
      <!--ro, opt, string, 人员不能进入时的标题文字-->test
    </unavailableText>
    <unavailableFontColor>
      <!--ro, opt, object, 人员不能进入时的字体颜色-->
      <RGB>
        <!--ro, opt, int, 十进制数据表示三原色-->1
      </RGB>
    </UnavailableFontColor>
  </Prompt>
  <PeopleIcon>
    <!--ro, opt, object, 行人通行图标-->
    <enabled>
      <!--ro, req, bool, 使能-->true
    </enabled>
    <Position>
      <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
      <positionX>
        <!--ro, req, int, X坐标-->1
      </positionX>
      <positionY>
        <!--ro, req, int, Y坐标-->1
      </positionY>
      <height>
        <!--ro, req, int, 宽-->1
      </height>
      <width>
        <!--ro, req, int, 高-->1
      </width>
    </Position>
    <materialNo>
      <!--ro, opt, int, 素材ID-->1
    </materialNo>
  </PeopleIcon>
  <InsidePeopleTitle>
    <!--ro, opt, object, 已经进入人数标题-->
    <enabled>
      <!--ro, req, bool, 使能-->true
    </enabled>
    <fontSize>
      <!--ro, req, int, 字体大小-->1
    </fontSize>
    <FontColor>
      <!--ro, req, object, 字体颜色-->
      <RGB>
        <!--ro, opt, int, 十进制数据表示三原色-->1
      </RGB>
    </FontColor>
  <Position>
```

```
<!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
<positionX>
  <!--ro, req, int, X坐标-->1
</positionX>
<positionY>
  <!--ro, req, int, Y坐标-->1
</positionY>
<height>
  <!--ro, req, int, 宽-->1
</height>
<width>
  <!--ro, req, int, 高-->1
</width>
</Position>
<text>
  <!--ro, opt, string, 标题文字-->test
</text>
</InsidePeopleTitle>
<InsidePeopleNumber>
  <!--ro, opt, object, 已经进入人数-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <fontSize>
    <!--ro, req, int, 字体大小-->1
  </fontSize>
  <FontColor>
    <!--ro, req, object, 字体颜色-->
    <RGB>
      <!--ro, opt, int, 十进制数据表示三原色-->1
    </RGB>
  </FontColor>
</Position>
<!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
<positionX>
  <!--ro, req, int, X坐标-->1
</positionX>
<positionY>
  <!--ro, req, int, Y坐标-->1
</positionY>
<height>
  <!--ro, req, int, 宽-->1
</height>
<width>
  <!--ro, req, int, 高-->1
</width>
</Position>
</InsidePeopleNumber>
<AvailablePeopleTitle>
  <!--ro, opt, object, 还可以进入人数标题-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <fontSize>
    <!--ro, req, int, 字体大小-->1
  </fontSize>
  <FontColor>
    <!--ro, req, object, 字体颜色-->
    <RGB>
      <!--ro, opt, int, 十进制数据表示三原色-->1
    </RGB>
  </FontColor>
</Position>
<!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
<positionX>
  <!--ro, req, int, X坐标-->1
</positionX>
<positionY>
  <!--ro, req, int, Y坐标-->1
</positionY>
<height>
  <!--ro, req, int, 宽-->1
</height>
<width>
  <!--ro, req, int, 高-->1
</width>
</Position>
<text>
  <!--ro, opt, string, 标题文字-->test
</text>
</AvailablePeopleTitle>
<AvailablePeopleNumber>
  <!--ro, opt, object, 还可以进入人数-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <fontSize>
    <!--ro, req, int, 字体大小-->1
  </fontSize>
  <Position>
    <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--ro, req, int, X坐标-->1
```

```
</positionX>
<positionY>
  <!--ro, req, int, Y坐标-->1
</positionY>
<height>
  <!--ro, req, int, 宽-->1
</height>
<width>
  <!--ro, req, int, 高-->1
</width>
</Position>
<FontColor>
  <!--ro, opt, object, 人员可以进入时的字体颜色-->
  <RGB>
    <!--ro, opt, int, 十进制数据表示三原色-->1
  </RGB>
</FontColor>
<UnavailableFontColor>
  <!--ro, opt, object, 人员不能进入时的字体颜色-->
  <RGB>
    <!--ro, opt, int, 十进制数据表示三原色-->1
  </RGB>
</UnavailableFontColor>
</AvailablePeopleNumber>
<UnavailablePeopleIcon>
  <!--ro, opt, object, 人员不能进入时的行人图标-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <Position>
    <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--ro, req, int, X坐标-->1
    </positionX>
    <positionY>
      <!--ro, req, int, Y坐标-->1
    </positionY>
    <height>
      <!--ro, req, int, 宽-->1
    </height>
    <width>
      <!--ro, req, int, 高-->1
    </width>
  </Position>
  <materialNo>
    <!--ro, opt, int, 素材ID-->1
  </materialNo>
</UnavailablePeopleIcon>
<AudioAlarm>
  <!--ro, opt, object, 音频报警信息-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <materialNo>
    <!--ro, opt, int, 音频素材ID-->1
  </materialNo>
</AudioAlarm>
</PeopleCounting>
<Mask>
  <!--ro, opt, object, 未戴口罩提醒-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <fontSize>
    <!--ro, req, int, 字体大小-->1
  </fontSize>
  <FontColor>
    <!--ro, req, object, 字体颜色-->
    <RGB>
      <!--ro, opt, int, 十进制数据表示三原色-->1
    </RGB>
  </FontColor>
  <BackColor>
    <!--ro, req, object, 背景颜色-->
    <RGB>
      <!--ro, opt, int, 十进制数据表示三原色-->1
    </RGB>
  </BackColor>
</Position>
  <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
  <positionX>
    <!--ro, req, int, X坐标-->1
  </positionX>
  <positionY>
    <!--ro, req, int, Y坐标-->1
  </positionY>
  <height>
    <!--ro, req, int, 宽-->1
  </height>
  <width>
    <!--ro, req, int, 高-->1
  </width>
</Position>
<text>
```

```

<!--ro, opt, string, 标题文字-->test
</text>
<MaskIcon>
  <!--ro, opt, object, 未戴口罩图标-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <Position>
    <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--ro, req, int, X坐标-->1
    </positionX>
    <positionY>
      <!--ro, req, int, Y坐标-->1
    </positionY>
    <height>
      <!--ro, req, int, 宽-->1
    </height>
    <width>
      <!--ro, req, int, 高-->1
    </width>
  </Position>
  <materialNo>
    <!--ro, opt, int, 素材ID-->1
  </materialNo>
</MaskIcon>
<AudioAlarm>
  <!--ro, opt, object, 音频报警信息-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <materialNo>
    <!--ro, opt, int, 音频素材ID-->1
  </materialNo>
</AudioAlarm>
</Mask>
<Temperature>
  <!--ro, opt, object, 温度异常提醒-->
  <Prompt>
    <!--ro, opt, object, 提示信息-->
    <enabled>
      <!--ro, req, bool, 使能-->true
    </enabled>
    <fontSize>
      <!--ro, req, int, 字体大小-->1
    </fontSize>
    <FontColor>
      <!--ro, req, object, 字体颜色-->
      <RGB>
        <!--ro, opt, int, 十进制数据表示三原色-->1
      </RGB>
    </FontColor>
    <BackColor>
      <!--ro, req, object, 背景颜色-->
      <RGB>
        <!--ro, opt, int, 十进制数据表示三原色-->1
      </RGB>
    </BackColor>
    <Position>
      <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
      <positionX>
        <!--ro, req, int, X坐标-->1
      </positionX>
      <positionY>
        <!--ro, req, int, Y坐标-->1
      </positionY>
      <height>
        <!--ro, req, int, 宽-->1
      </height>
      <width>
        <!--ro, req, int, 高-->1
      </width>
    </Position>
    <text>
      <!--ro, opt, string, 标题文字-->test
    </text>
  </Prompt>
<TemperatureIcon>
  <!--ro, opt, object, 温度异常图标-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <Position>
    <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--ro, req, int, X坐标-->1
    </positionX>
    <positionY>
      <!--ro, req, int, Y坐标-->1
    </positionY>
    <height>
      <!--ro, req, int, 宽-->1
    </height>

```

```
<width>
  <!--ro, req, int, 高-->1
</width>
</Position>
<materialNo>
  <!--ro, opt, int, 素材ID-->1
</materialNo>
</TemperatureIcon>
<AudioAlarm>
  <!--ro, opt, object, 音频报警信息-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <materialNo>
    <!--ro, opt, int, 音频素材ID-->1
  </materialNo>
</AudioAlarm>
<abnormalHighTemperature>
  <!--ro, opt, float, 异常高温值, unit:℃, unitType:温度, desc:保留一位小数-->37.5
</abnormalHighTemperature>
</Temperature>
<WearMask>
  <!--ro, opt, object, 已戴口罩提醒-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <fontSize>
    <!--ro, req, int, 字体大小-->1
  </fontSize>
  <fontColor>
    <!--ro, req, object, 字体颜色-->
    <RGB>
      <!--ro, opt, int, 十进制数据表示三原色-->1
    </RGB>
  </FontColor>
  <BackColor>
    <!--ro, req, object, 背景颜色-->
    <RGB>
      <!--ro, opt, int, 十进制数据表示三原色-->1
    </RGB>
  </BackColor>
  <Position>
    <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--ro, req, int, X坐标-->1
    </positionX>
    <positionY>
      <!--ro, req, int, Y坐标-->1
    </positionY>
    <height>
      <!--ro, req, int, 宽-->1
    </height>
    <width>
      <!--ro, req, int, 高-->1
    </width>
  </Position>
  <text>
    <!--ro, opt, string, 标题文字-->test
  </text>
  <MaskIcon>
    <!--ro, opt, object, 已戴口罩图标-->
    <enabled>
      <!--ro, req, bool, 使能-->true
    </enabled>
    <Position>
      <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
      <positionX>
        <!--ro, req, int, X坐标-->1
      </positionX>
      <positionY>
        <!--ro, req, int, Y坐标-->1
      </positionY>
      <height>
        <!--ro, req, int, 宽-->1
      </height>
      <width>
        <!--ro, req, int, 高-->1
      </width>
    </Position>
    <materialNo>
      <!--ro, opt, int, 素材ID-->1
    </materialNo>
  </MaskIcon>
</WearMask>
<NormalTemperature>
  <!--ro, opt, object, 温度正常提醒-->
  <Prompt>
    <!--ro, opt, object, 提示信息-->
    <enabled>
      <!--ro, req, bool, 使能-->true
    </enabled>
    <fontSize>
      <!--ro, req, int, 字体大小-->1
    </fontSize>
```

```

</fontSize>
<FontColor>
  <!--ro, req, object, 字体颜色-->
  <RGB>
    <!--ro, opt, int, 十进制数据表示三原色-->1
  </RGB>
</FontColor>
<BackColor>
  <!--ro, req, object, 背景颜色-->
  <RGB>
    <!--ro, opt, int, 十进制数据表示三原色-->1
  </RGB>
</BackColor>
<Position>
  <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
  <positionX>
    <!--ro, req, int, X坐标-->1
  </positionX>
  <positionY>
    <!--ro, req, int, Y坐标-->1
  </positionY>
  <height>
    <!--ro, req, int, 宽-->1
  </height>
  <width>
    <!--ro, req, int, 高-->1
  </width>
</Position>
<text>
  <!--ro, opt, string, 标题文字-->test
</text>
</Prompt>
<TemperatureIcon>
  <!--ro, opt, object, 温度正常图标-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <Position>
    <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
    <positionX>
      <!--ro, req, int, X坐标-->1
    </positionX>
    <positionY>
      <!--ro, req, int, Y坐标-->1
    </positionY>
    <height>
      <!--ro, req, int, 宽-->1
    </height>
    <width>
      <!--ro, req, int, 高-->1
    </width>
  </Position>
  <materialNo>
    <!--ro, opt, int, 素材ID-->1
  </materialNo>
</TemperatureIcon>
</NormalTemperature>
<NormalStatusAudioAlarm>
  <!--ro, opt, object, 正常状态下音频报警信息, desc:人员在检测时未发现上面的人数超限、未戴口罩、体温异常等事件时设备进行播报-->
  <enabled>
    <!--ro, req, bool, 使能-->true
  </enabled>
  <materialNo>
    <!--ro, opt, int, 音频素材ID-->1
  </materialNo>
</NormalStatusAudioAlarm>
</EventParam>
<AlarmDataSourceProperty>
  <!--ro, opt, object, 数据源信息, desc:该字段能力集缺少, 仅信发组件和设备使用-->
  <PictureProperty>
    <!--ro, opt, object, 图片属性-->
    <PictureSwitchEffect>
      <!--ro, opt, enum, 图片切换效果, subType:string, [leftInRightOut#左进右出,rightInLeftOut#右进左出]-->leftInRightOut
    </PictureSwitchEffect>
    <PictureScaleType>
      <!--ro, opt, enum, 图片缩放类型, subType:string, [fitCenter#拉伸填充,centerCrop#截取填充,fixXY#适应宽高]-->fitCenter
    </PictureScaleType>
  </PictureProperty>
  <AlarmDataSource>
    <!--ro, opt, object, 数据源-->
    <MaterialNo>
      <!--ro, opt, int, 素材编号-->1
    </MaterialNo>
    <Name>
      <!--ro, opt, string, 素材名称-->test
    </Name>
    <Address>
      <!--ro, opt, string, 素材URL地址-->test
    </Address>
    <WebProperty>
      <!--ro, opt, object, 网页属性-->
      <webType>
        <!--ro, opt, enum, 网页类型, subType:string, [desktop#桌面版,mobile#移动版], desc:默认按照设备的系统类型（Windows或Android）来显示网页--

```

```
</webType>
</WebProperty>
</AlarmDataSource>
</AlarmDataSourceProperty>
<ConferenceParam>
  <!--ro, opt, object, 会议信息, desc:当页面用于会议界面时有效-->
  <backPicId>
    <!--ro, opt, int, 背景图片素材ID-->1
  </backPicId>
  <Title>
    <!--ro, opt, object, 会议标题（主题）, desc:1、会议标题（主题）的文字内容通过场地预约配置协议下发,对
应/ISAPI/Publish/Signage/PlaceReservation?format=json中的programID字段。2、可通过会议计划（/ISAPI/InteractivePanel/Meeting/plan?format=json）中的
meetingTopic字段下发-->
    <enabled>
      <!--ro, req, bool, 使能-->true
    </enabled>
    <fontSize>
      <!--ro, req, int, 字体大小-->1
    </fontSize>
    <FontColor>
      <!--ro, req, object, 字体颜色-->
      <RGB>
        <!--ro, opt, int, 十进制数据表示三原色-->1
      </RGB>
    </FontColor>
    <BackColor>
      <!--ro, req, object, 背景颜色-->
      <RGB>
        <!--ro, opt, int, 十进制数据表示三原色-->1
      </RGB>
    </BackColor>
    <Position>
      <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
      <positionX>
        <!--ro, req, int, X坐标-->1
      </positionX>
      <positionY>
        <!--ro, req, int, Y坐标-->1
      </positionY>
      <height>
        <!--ro, req, int, 宽-->1
      </height>
      <width>
        <!--ro, req, int, 高-->1
      </width>
    </Position>
  </Title>
  <VoiceBroadcast>
    <!--ro, opt, object, 语音播报-->
    <enabled>
      <!--ro, opt, bool, 是否启用语音播报-->true
    </enabled>
    <broadcastSpeed>
      <!--ro, opt, enum, 播报语速, subType:string, [sLower#慢,sLow#较慢,medium#中速,fast#较快,faster#快]-->sLower
    </broadcastSpeed>
    <inLibVoice>
      <!--ro, opt, string, 在库人员语音播报内容-->test
    </inLibVoice>
    <outLibVoice>
      <!--ro, opt, string, 未知人员语音播报内容-->test
    </outLibVoice>
  </VoiceBroadcast>
  <WelcomeWord>
    <!--ro, opt, object, 欢迎词-->
    <enabled>
      <!--ro, opt, bool, 是否启用欢迎词-->true
    </enabled>
    <inLibWord>
      <!--ro, opt, string, 在库人员欢迎词内容, range:[1,128]-->test
    </inLibWord>
    <outLibWord>
      <!--ro, opt, string, 未知人员欢迎词内容, range:[1,128]-->test
    </outLibWord>
  </WelcomeWord>
  <ShowInfo>
    <!--ro, opt, object, 展示信息-->
    <row>
      <!--ro, opt, int, 窗口内人员签到展示行数-->1
    </row>
    <column>
      <!--ro, opt, int, 窗口内人员签到展示列数-->1
    </column>
    <Position>
      <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
      <positionX>
        <!--ro, req, int, x坐标-->1
      </positionX>
      <positionY>
        <!--ro, req, int, y坐标-->1
      </positionY>
      <height>
        <!--ro, req, int, 宽-->1
      </height>
```

```

<!--ro, req, int, 高-->1
</width>
</Position>
<inLib>
<!--ro, opt, object, 在库信息-->
<nameEnabled>
<!--ro, opt, bool, 在库人员是否展示姓名-->true
</nameEnabled>
</inLib>
<defaultShowInfoEnabled>
<!--ro, opt, bool, 是否采用人员默认展示信息, desc: 下发该字段, 且为true, 表示人员展示信息采用默认值。坐标信息等不用下发-->true
</defaultShowInfoEnabled>
</ShowInfo>
<ModuleInfoList>
<!--ro, opt, array, 界面模块信息列表, subType:object-->
<ModuleInfo>
<!--ro, opt, object, 界面模块信息-->
<moduleType>
<!--ro, req, enum, 模块类型, subType:string, [conferenceTime#会议时间(对应timeRange), conferenceSite#会议地点(对应会议室名称
meetingName), conferenceStatus#会议状态(对应meetingStatus), signInPersonNum#已签到人数, totalPersonNum#应到人数, unSignInPersonNum#未签到人
数, signInAndTotalPersonNum#已签到/应到人数, QRCode#会议室二维码, meetingHostName#主持人, empLoyeeNo#参会人, customKey#自定义字段, nextMeeting#当天下一场会议信
息]-->conferenceTime
</moduleType>
<enabled>
<!--ro, req, bool, 使能-->true
</enabled>
<fontSize>
<!--ro, req, int, 字体大小-->1
</fontSize>
<fontColor>
<!--ro, req, object, 字体颜色-->
<RGB>
<!--ro, opt, int, 十进制数据表示三原色-->1
</RGB>
</FontColor>
<BackColor>
<!--ro, opt, object, 背景颜色-->
<RGB>
<!--ro, opt, int, 十进制数据表示三原色-->1
</RGB>
</BackColor>
<backPicId>
<!--ro, opt, int, 背景图片素材ID, desc: 背景图和背景颜色同时仅支持一种, 两者均下发, 设备优先解析背景图字段-->1
</backPicId>
<Position>
<!--ro, opt, object, 位置, desc: 左上角为坐标原点, 取值范围和设备分辨率保持一致-->
<positionX>
<!--ro, req, int, X坐标-->1
</positionX>
<positionY>
<!--ro, req, int, Y坐标-->1
</positionY>
<height>
<!--ro, req, int, 宽-->1
</height>
<width>
<!--ro, req, int, 高-->1
</width>
</Position>
<CustomKey>
<!--ro, opt, object, 自定义字段-->
<key>
<!--ro, opt, string, 自定义字段-->test
</key>
<associateProtocolURL>
<!--ro, opt, string, 关联协议URL-->test
</associateProtocolURL>
</CustomKey>
</ModuleInfo>
</ModuleInfoList>
<conferenceStatus>
<!--ro, opt, enum, 会议状态, subType:string, [idle#空闲, signing#签到中, meeting#开会中], desc: 根据不同的会议状态, 配置不同的会议界面-->idle
</conferenceStatus>
</ConferenceParam>
<CallBtn>
<!--ro, opt, object, 呼叫按钮窗口-->
<backPicId>
<!--ro, opt, int, 背景图片素材ID-->1
</backPicId>
</CallBtn>
<OpenDoorBtn>
<!--ro, opt, object, 开门按钮窗口-->
<backPicId>
<!--ro, opt, int, 背景图片素材ID-->1
</backPicId>
</OpenDoorBtn>
<QRCodeBtn>
<!--ro, opt, object, 二维码按钮窗口-->
<backPicId>
<!--ro, opt, int, 背景图片素材ID-->1
</backPicId>
</QRCodeBtn>
<AuthenticationSuccess>

```

```

<!--ro, opt, object, 认证成功窗口-->
<BackColor>
  <!--ro, opt, object, 背景颜色-->
  <RGB>
    <!--ro, opt, int, 十进制数据表示三原色-->1
  </RGB>
</BackColor>
<backPicId>
  <!--ro, opt, int, 背景图片素材ID, desc: 背景图和背景颜色同时仅支持一种, 两者均下发, 设备优先解析背景图字段-->1
</backPicId>
<SubWindowList>
  <!--ro, opt, object, 子窗口列表-->
  <SubWindow>
    <!--ro, opt, object, 子窗口-->
    <subWindowType>
      <!--ro, req, enum, 子窗口类型, subType:string, [temperature#温度, facePic#人脸图片, name#姓名, employeeNo#工号, group#组织/部门, tips#提示信
息]-->temperature
    </subWindowType>
    <enabled>
      <!--ro, req, bool, 使能-->true
    </enabled>
    <fontSize>
      <!--ro, opt, int, 字体大小, unit:px, unitType:图像显示-->1
    </fontSize>
    <fontColor>
      <!--ro, opt, int, 字体颜色-->1
    </fontColor>
    <fontStyle>
      <!--ro, opt, string, 字体风格, desc:bold-加粗, italic-倾斜, underline-下划线, strikethrough-删除线-->bold,italic,underline,strikeout
    </fontStyle>
    <Position>
      <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
      <positionX>
        <!--ro, req, int, x坐标-->1
      </positionX>
      <positionY>
        <!--ro, req, int, y坐标-->1
      </positionY>
      <height>
        <!--ro, req, int, 宽-->1
      </height>
      <width>
        <!--ro, req, int, 高-->1
      </width>
    </Position>
  </SubWindow>
</SubWindowList>
</AuthenticationSuccess>
<AuthenticationFail>
  <!--ro, opt, object, 认证失败窗口-->
  <BackColor>
    <!--ro, opt, object, 背景颜色-->
    <RGB>
      <!--ro, opt, int, 十进制数据表示三原色-->1
    </RGB>
  </BackColor>
  <backPicId>
    <!--ro, opt, int, 背景图片素材ID, desc: 背景图和背景颜色同时仅支持一种, 两者均下发, 设备优先解析背景图字段-->1
  </backPicId>
  <SubWindowList>
    <!--ro, opt, object, 子窗口列表-->
    <SubWindow>
      <!--ro, opt, object, 子窗口-->
      <subWindowType>
        <!--ro, req, enum, 子窗口类型, subType:string, [temperature#温度, facePic#人脸图片, name#姓名, employeeNo#工号, group#组织/部门, tips#提示信
息]-->temperature
      </subWindowType>
      <enabled>
        <!--ro, req, bool, 使能-->true
      </enabled>
      <fontSize>
        <!--ro, opt, int, 字体大小, unit:px, unitType:图像显示-->1
      </fontSize>
      <fontColor>
        <!--ro, opt, int, 字体颜色-->1
      </fontColor>
      <fontStyle>
        <!--ro, opt, string, 字体风格, desc:bold-加粗, italic-倾斜, underline-下划线, strikethrough-删除线-->bold,italic,underline,strikeout
      </fontStyle>
      <Position>
        <!--ro, opt, object, 位置, desc:左上角为坐标原点, 取值范围和设备分辨率保持一致-->
        <positionX>
          <!--ro, req, int, x坐标-->1
        </positionX>
        <positionY>
          <!--ro, req, int, y坐标-->1
        </positionY>
        <height>
          <!--ro, req, int, 宽-->1
        </height>
        <width>
          <!--ro, req, int, 高-->1
        </width>
      </Position>
    </SubWindow>
  </SubWindowList>

```

```

        </Position>
        <facePicID>
            <!--ro, opt, int, 默认人脸图片ID, step:1, dep:and,
            {$.PageList.Page.WindowsList[*].Windows.AuthenticationFail.SubWindowList.SubWindow.subWindowType,eq,facePic}-->1
        </facePicID>
        </SubWindow>
    </SubWindowList>
</AuthenticationFail>
<StatusBar>
    <!--ro, opt, object, 状态栏, desc:当素材类型为状态栏时有效, 且必填-->
    <backPicId>
        <!--ro, opt, int, 背景图片素材ID, desc:背景图和背景颜色同时仅支持一种, 两者均下发, 设备优先解析背景图字段-->1
    </backPicId>
    <statusBarType>
        <!--ro, req, enum, 状态栏类型, subType:string, [fixed#固定的], desc:当为固定类型时, 窗口位置不生效, 设备将状态栏固定在顶部显示-->fixed
    </statusBarType>
    <SubStatusList>
        <!--ro, opt, object, 子状态列表-->
        <SubStatus>
            <!--ro, opt, object, 子状态-->
            <statusType>
                <!--ro, req, enum, 子状态, subType:string, [dial#拨号状态(3G4G等),wifi#wifi状态,EZVIZ#萤石状态,wiredNetwork#有线网络状态,deploy#布防状态,temperature#测温]-->dial
            </statusType>
            <enabled>
                <!--ro, req, bool, 是否显示-->true
            </enabled>
        </SubStatus>
    </SubStatusList>
</StatusBar>
</Windows>
</WindowsList>
</Page>
</PageList>
<programSize>
    <!--ro, opt, int, 节目大小, unit:Byte, unitType:信息量-->1
</programSize>
<programLength>
    <!--ro, opt, int, 节目时长, unit:s, unitType:时间-->1
</programLength>
<coordinateType>
    <!--ro, opt, enum, 节目中坐标基准值类型, subType:string, [uniformCoordinate#统一坐标(实际的分辨率坐标按基准坐标等比例进行转换),resolutionCoordinate#实际分辨率坐标]-->uniformCoordinate
</coordinateType>
<screenType>
    <!--ro, opt, enum, 屏幕类型, subType:string, [first#主屏,second#辅屏], desc:当programType为doubleSide表示创建双屏节目时生效-->first
</screenType>
<audioEnabled>
    <!--ro, opt, bool, 节目音频使能, desc:当programType为doubleSide表示创建双屏节目时生效,用来防止主屏声音和辅屏声音混淆,主辅屏节目的音频只开一个-->true
</audioEnabled>
<PageIdList>
    <!--ro, opt, array, 页面id列表, subType:object-->
    <PageId>
        <!--ro, opt, object, 页面id-->
        <id>
            <!--ro, req, int, 页面ID-->1
        </id>
    </PageId>
</PageIdList>
<programUUID>
    <!--ro, opt, string, 节目UUID, range:[1,32], desc:平台下发-->test
</programUUID>
</Program>
</ProgramList>

```

69.66.9 删除单个节目

Request URL

DELETE /ISAPI/Publish/ProgramMgr/program/<programID>

查询参数

参数名称	参数类型	描述
programID	string	节目ID

请求报文

无

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>

<ResponseStatus xmlns="http://www.isapi.org/ver20/XMLSchema" version="2.0">
  <!--ro, req, object, 响应消息, attr:version{ro, req, string, 版本}-->
  <requestURL>
    <!--ro, req, string, 请求的URL-->null
  </requestURL>
  <statusCode>
    <!--ro, req, enum, 状态码, subType:int, [0#OK,1#OK,2#Device Busy,3#Device Error,4#Invalid Operation,5#Invalid XML Format,6#Invalid XML Content,7#Reboot Required]-->0
  </statusCode>
  <statusString>
    <!--ro, req, enum, 状态信息, subType:string, [OK#成功,Device Busy#设备忙,Device Error#设备异常,Invalid Operation#无效的操作,Invalid XML Format#无效的XML格式,Invalid XML Content#无效的XML内容,Reboot#设备重启]-->OK
  </statusString>
  <subStatusCode>
    <!--ro, req, string, 详细错误码英文描述, desc:详细错误码的英文描述-->OK
  </subStatusCode>
</ResponseStatus>
```

69.67 事件消息推送管理

69.67.1 获取报告上传方式配置能力

Request URL

GET /ISAPI/SecurityCP/ReportCenterCfg/capabilities?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "ReportCenterCfg": {
    /*ro, req, object, 能力节点*/
    "CenterID": {
      /*ro, opt, object, 中心组序号*/
      "@min": 1,
      /*ro, opt, int, 最小值*/
      "@max": 1
      /*ro, opt, int, 最大值*/
    },
    "enable": "true,false",
    /*ro, opt, string, 使能*/
    "ChanAlarmMode": {
      /*ro, opt, object, 中心组报警通道*/
      "maxSize": 1,
      /*ro, opt, int, 最大通道数*/
      "id": {
        /*ro, opt, object, 通道ID, desc:1-主通道, 2-备份通道1, 3-备份通道2, 4-备份通道3*/
        "@min": 1,
        /*ro, opt, int, 最小值*/
        "@max": 2
        /*ro, opt, int, 最大值*/
      },
      "chanAlarmMode": {
        /*ro, opt, object, 报警通道模式, desc:T1-T1通道, T2-T2通道, N1-N1通道, N2-N2通道, G1-G1通道, G2-G2通道, N3-N3通道, N4-N4通道*/
        "@opt": "T1,T2,N1,N2,G1,G2,N3,N4"
        /*ro, opt, string, 可选项*/
      }
    }
  }
}
```

69.67.2 配置报告上传方式

Request URL

PUT /ISAPI/SecurityCP/ReportCenterCfg/<centerID>?format=json

查询参数

参数名称	参数类型	描述
centerID	string	

请求报文

```
{
  "ReportCenterCfg": {
    /*req, object, 上报中心配置节点*/
    "enable": true,
    /*opt, bool, 使能*/
    "ChanAlarmMode": [
      /*opt, array, 中心组报警通道, subType:object*/
      {
        "id": 1,
        /*opt, enum, 通道ID, subType:int, [1#主通道,2#备份通道1,3#备份通道2,4#备份通道3], desc:1-主通道, 2-备份通道1, 3-备份通道2, 4-备份通道3*/
        "chanAlarmMode": "T1"
        /*opt, enum, 报警通道模式, subType:string, [T1#T1通道,T2#T2通道,N1#N1通道,N2#N2通道,G1#G1通道,G2#G2通道,N3#N3通道,N4#N4通道],
        desc:T-无线, N-有线, G-移动网络
        支持配置为空, 表示未选择模式*/
      }
    ]
  }
}
```

响应报文

```
{
  "statusCode": 1,
  /*ro, opt, int, 状态码, desc:无法用1表示时（1表示成功且无特殊状态）必须返回*/
  "statusString": "OK",
  /*ro, opt, string, 状态描述, range:[1,64], desc:无法用OK表示时（OK表示成功且无特殊状态）必须返回*/
  "subStatusCode": "ok",
  /*ro, opt, string, 子状态码, range:[1,64], desc:无法用ok表示（ok表示成功且无特殊状态）必须返回*/
  "errorCode": 1,
  /*ro, opt, int, 错误码, desc:当statusCode不为1时, 与subStatusCode对应*/
  "errorMsg": "ok"
  /*ro, opt, string, 错误信息, desc:当statusCode不为1时, 必须返回,解释信息在协议约束中, 允许设备在后续的版本迭代中, 进行优化丰富提升（不限制死）*/
}
```

69.67.3 获取报告上传方式

Request URL

GET /ISAPI/SecurityCP/ReportCenterCfg/<centerID>?format=json

查询参数

参数名称	参数类型	描述
centerID	string	

请求报文

无

响应报文

```
{
  "ReportCenterCfg": {
    /*ro, req, object, 上报中心配置节点*/
    "enable": true,
    /*ro, opt, bool, 使能*/
    "ChanAlarmMode": [
      /*ro, opt, array, 中心组报警通道, subType:object*/
      {
        "id": 1,
        /*ro, opt, enum, 通道ID, subType:int, [1#主通道,2#备份通道1,3#备份通道2,4#备份通道3], desc:1-主通道, 2-备份通道1, 3-备份通道2, 4-备份通道3*/
        "chanAlarmMode": "T1"
        /*ro, opt, enum, 报警通道模式, subType:string, [T1#T1通道,T2#T2通道,N1#N1通道,N2#N2通道,G1#G1通道,G2#G2通道,N3#N3通道,N4#N4通道],
        desc:T-无线, N-有线, G-移动网络
        支持配置为空, 表示未选择模式*/
      }
    ]
  }
}
```

69.68 报警输入/输出(S)

69.68.1 获取全部报警输出口参数

Request URL

GET /ISAPI/SecurityCP/AlarmOutCfg?format=json

查询参数

无

请求报文

无

响应报文

```
{
  "AlarmOutCfgList": [
    /*ro, req, array, 报警输出配置列表, subType:object*/
    {
      "AlarmOutCfg": {
        /*ro, opt, object, 报警输出*/
        "alarmID": 1,
        /*ro, opt, int, 报警输出编号, desc:后补入的字段, 部分产品可能不支持, 没有id时默认按列表顺序从1开始*/
        "name": "test",
        /*ro, opt, string, 名称*/
        "delay": 1,
        /*ro, opt, int, 输出延迟(持续)时间, desc:表示输出信号的延迟结束时间,即持续时间.单位:秒, 范围: 0-3599s, 0表示一直有输出*/
        "triggerIndex": 1,
        /*ro, opt, int, 触发器号*/
        "associateAlarmIn": [1, 3, 5],
        /*ro, opt, array, 表示警号跟随的报警输入通道, subType:int, desc:多个报警输入同时触发一个警号输出, [1,3,5]代表: 警号跟随报警输入1、报警输入
3、报警输入5*/
        "moduleType": 1,
        /*ro, opt, enum, 外接触发器类型, subType:int, [1#本地触发器,2#4路触发器,3#8路触发器,4#单防区触发器,5#32路触发器,6#1门就地控制器,7#2门就地控
制器,8#4门就地控制器,9#2路触发器,10#键盘,11#扩展触发器,12#网络凭证识别终端], desc:1-本地触发器, 2-4路触发器, 3-8路触发器, 4-单防区触发器, 5-32路触发器, 6-1
门就地控制器, 7-2门就地控制器, 8-4门就地控制器, 9-2路触发器*/
        "moduleStatus": "online",
        /*ro, opt, enum, 外接触发器状态, subType:string, [online#在线,offline#离线], desc:online-在线, offline-离线*/
        "moduleAddress": 1,
        /*ro, opt, int, 外接触发器地址, desc:扩展模块从0-255, 65535表示无效*/
        "moduleChan": 1,
        /*ro, opt, int, 外接触发器通道号, desc:从1开始, 最大值根据模块类型来决定, 255表示无效*/
        "workMode": "linkage",
        /*ro, opt, enum, 工作模式, subType:string, [linkage#联动, followUp#随动], desc:linkage-联动, followUp-随动*/
        "alarmOutMode": "nonPluse",
        /*ro, opt, enum, 输出模式, subType:string, [nonPluse#非脉冲模式, pluse#脉冲模式], desc:nonPluse-非脉冲模式, pluse-脉冲模式*/
        "timeOn": 1,
        /*ro, opt, int, 开时间, desc:1-60s*/
        "timeOff": 1,
        /*ro, opt, int, 关时间, desc:1-60s*/
        "triggerMode": "alwaysOpen",
        /*ro, opt, enum, 触发模式, subType:string, [alwaysOpen#常开触发, alwaysClose#常闭触发]*/
      }
    }
  ]
}
```

69.68.2 获取设置报警输出口输出状态参数能力

Request URL

GET /ISAPI/SecurityCP/SetAlarmHostOut/capabilities?format=json

查询参数

无

请求报文

无

响应报文

```

{
  "SetAlarmHostOut": {
    /*ro, req, object, 能力节点*/
    "alarmOutPort": {
      /*ro, req, object, 报警输出索引, desc:从0开始, 65535表示全部报警输出索引*/
      "@min": 1,
      /*ro, opt, int, 最小值*/
      "@max": 1
      /*ro, req, int, 最大值*/
    },
    "alarmOutStatus": {
      /*ro, req, object, 报警输出状态, desc:stop-停止输出, start-输出*/
      "@max": "stop,start"
      /*ro, req, string, 最大值*/
    }
  }
}

```

69.68.3 配置报警输出输出状态

Request URL

PUT /ISAPI/SecurityCP/SetAlarmHostOut?format=json

查询参数

无

请求报文

```

{
  "SetAlarmHostOut": {
    /*req, object, 设置报警输出*/
    "alarmOutPort": 1,
    /*req, int, 报警输出索引, desc:从0开始, 65535表示全部报警输出索引*/
    "alarmOutStatus": "stop"
    /*req, enum, 报警输出状态, subType:string, [stop#停止输出,start#输出], desc:stop-停止输出, start-输出*/
  }
}

```

响应报文

```

{
  "statusCode": 1,
  /*ro, opt, int, 状态码, desc:无法用1表示时（1表示成功且无特殊状态）必须返回*/
  "statusString": "OK",
  /*ro, opt, string, 状态描述, range:[1,64], desc:无法用OK表示时（OK表示成功且无特殊状态）必须返回*/
  "subStatusCode": "ok",
  /*ro, opt, string, 子状态码, range:[1,64], desc:无法用ok表示（ok表示成功且无特殊状态）必须返回*/
  "errorCode": 1,
  /*ro, opt, int, 错误码, desc:当statusCode不为1时, 与subStatusCode对应*/
  "errorMsg": "ok"
  /*ro, opt, string, 错误信息, desc:当statusCode不为1时, 必须返回,解释信息在协议约束中, 允许设备在后续的版本迭代中, 进行优化丰富提升（不限制死）*/
}

```

69.68.4 获取报警输出参数配置能力

Request URL

GET /ISAPI/SecurityCP/AlarmOutCfg/capabilities?format=json

查询参数

无

请求报文

无

响应报文

```

{
  "AlarmOutCfg": {
    /*ro, req, object, 能力节点*/
    "maxSize": 10,
    /*ro, opt, int, 批量配置list数组最大数量*/
    "alarmOutNo": {
      /*ro, opt, object, 报警输出编号*/
      "@min": 1,
      /*ro, opt, int, 最小值*/
    }
  }
}

```

```

        /*ro, opt, int, 最大值*/
        "@max": 1
    /*ro, opt, int, 最大值*/
    },
    "name": {
    /*ro, opt, object, 名称*/
        "@min": 1,
        /*ro, opt, int, 最小值*/
        "@max": 32
        /*ro, opt, int, 最大值*/
    },
    "delay": {
    /*ro, opt, object, 输出延迟(持续)时间, desc:表示输出信号的延迟结束时间,即持续时间.单位: 秒, 范围: 0-3599s, 0表示一直有输出*/
        "@min": 0,
        /*ro, opt, int, 最小值*/
        "@max": 3599
        /*ro, opt, int, 最大值*/
    },
    "triggerIndex": {
    /*ro, opt, object, 触发器号*/
        "@min": 1,
        /*ro, opt, int, 最小值*/
        "@max": 1
        /*ro, opt, int, 最大值*/
    },
    "associateAlarmIn": {
    /*ro, opt, object, 表示警号跟随的报警输入通道, desc:多个报警输入同时触发一个警号输出*/
        "@min": 1,
        /*ro, opt, int, 最小值*/
        "@max": 1
        /*ro, opt, int, 最大值*/
    },
    "moduleType": {
    /*ro, opt, object, 外接触发器类型, desc:1-本地触发器, 2-4路触发器, 3-8路触发器, 4-单防区触发器, 5-32路触发器, 6-1门就地控制器, 7-2门就地控制器, 8-4
    门就地控制器, 9-2路触发器,10-键盘, 11-扩展触发器,12-网络凭证识别终端*/
        "@opt": "1,2,3,4,5,6,7,8,9,10,11,12"
        /*ro, opt, string, 可选项*/
    },
    "moduleStatus": {
    /*ro, opt, object, 外接触发器状态, desc:onLine-在线, offLine-离线*/
        "@opt": "onLine,offLine"
        /*ro, opt, string, 可选项*/
    },
    "moduleAddress": {
    /*ro, opt, object, 外接触发器地址, desc:扩展模块从0-255, 65535表示无效*/
        "@min": 0,
        /*ro, opt, int, 最小值*/
        "@max": 255
        /*ro, opt, int, 最大值*/
    },
    "moduleChan": {
    /*ro, opt, object, 外接触发器通道号, desc:从1开始, 最大值根据模块类型来决定, 255表示无效*/
        "@min": 1,
        /*ro, opt, int, 最小值*/
        "@max": 1
        /*ro, opt, int, 最大值*/
    },
    "workMode": {
    /*ro, opt, object, 工作模式, desc:linkage-联动, followUp-随动*/
        "@opt": "linkage,followUp"
        /*ro, opt, string, 可选项*/
    },
    "alarmOutMode": {
    /*ro, opt, object, 输出模式, desc:nonPluse-非脉冲模式, pluse-脉冲模式*/
        "@opt": "nonPluse,pluse"
        /*ro, opt, string, 可选项*/
    },
    "timeOn": {
    /*ro, opt, object, 开时间, desc:1-60s*/
        "@min": 1,
        /*ro, opt, int, 最小值*/
        "@max": 60
        /*ro, opt, int, 最大值*/
    },
    "timeOff": {
    /*ro, opt, object, 关时间, desc:1-60s*/
        "@min": 1,
        /*ro, opt, int, 最小值*/
        "@max": 60
        /*ro, opt, int, 最大值*/
    },
    "triggerMode": {
    /*ro, opt, object, 触发模式取值范围*/
        "@opt": ["alwaysOpen", "alwaysClose"]
        /*ro, opt, array, 可选项, subType:string, desc:alwaysOpen-常开触发,alwaysClose-常闭触发*/
    }
}
}
}

```