

MegBox-B4H 系列智能分析盒
Web API 接口说明
软件版本：V1.3.1

版权所有，保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

注意

您购买的产品、服务或特性等应受旷视科技公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，旷视科技公司对本文档内容不做任何明示或默示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

修订记录

版本	日期	修订内容
V1.0	2018/4/10	初稿
V1.1	2018/5/23	针对于 B3 多路接入进行修改，同时针对不合理的地方进行改进
V1.2	2018/6/8	对照片流 API 添加厂商设备信息、添加 Wifi、有线网口开关。
V2.0	2018/8/2	添加支持教育行业、洞鉴接口： 1. 支持 Feature 的更新下发； 2. 布控需要有一个布控计划表，支持 20 次以上的布控计划；
V2.1	2018/11/1	删除无用接口，重排优先级，拆分大 Videos 接口为三个接口：视频接入、布控、上报设置
V2.2	2019/6/24	增加人脸属性，特征等上报数据判断字段；增加用户接入接口；部分配置字段更改；
	2019/7/10	加入 MD5 转换示例（Java 版本），删除无用接口，统一错误码列表规范
	2019/7/17	增加回复人脸图片检测结果 <code>detectionResult{ }</code> ，加入 MD5 转换示例（C 和 Python 版本）
V2.3	2019/9/3	增加提示， <code>/faces</code> ； <code>/alerts</code> ； <code>/captures</code> 等接口需按固定顺序解析 URI，不支持随意顺序参数；
V2.4	2019/9/24	增加抓拍大图使能 <code>captureFullPicEnabled</code> ； <code>/devices/reports</code> 的 <code>post</code> 方法中 <code>alertResend</code> 范围限定为 1 和 2 合法；

版本	日期	修订内容
V2.5	2019/10/26	删除无用接口（特征入库）； 合入部分 B3R v2.0 新增接口 1. 规则权限 API 2. 继电器配置 API 3. HTTP 主动上报 API 4. 二次识别和陌生人 retry 5. 背景大图压缩等级配置 6. 视频布控获取背景图
	2019/10/29	心跳上报数据 deviceId 修改，固定 20 位，mac+后面补 0
	2019/11/18	二次识别和陌生人 retry 功能生效说明
	2019/12/27	删除人脸分组接口数据类型修改
	2020/1/7	1. trackId 最大支持 20 位整数； 2. 入库照片描述 100 字节限制扩大至 255 字节；
v2.6	2020/3/25	修改人脸属性相关协议
v2.6	2020/4/2	增加图片流资源 API;增加 ROI 区域资源 API
v2.6	2020/4/14	增加 gat1400server 协议

版本	日期	修订内容
v2.6	2020/4/15	为视频、图片布控增加全局陌生人配置参数
v2.6	2020/4/16	同步 GAT1400 客户端功能协议
v2.6	2020/5/13	增加 FTP 推图协议
v2.6	2020/5/15	修改创建视频布点协议字段 videoName 为必填项，原为非必填
v2.6	2020/6/15	补充入库失败列表 API
v2.6	2020/7/1	补充视频布控重置 TrackReset 接口
v2.6	2020/7/28	补充 5 个 GAT1400 非必选参数
v2.7	2020/8/6	增加图片流二次识别参数
v2.8	2020/8/20	增加通道结构化布控配置
v2.8	2020/9/7	增加日志模块 API
v2.8	2020/9/9	补充 onvif 搜索模块 API
v2.8	2020/10/8	增加繁忙状态 API
v2.8	2020/10/12	增加双网卡模式配置
v2.8	2020/10/20	http 上报增加 track_die 字段
v2.8	2020/10/27	视频结构化全量抓拍获取 视频结构化指定目标绑定关系获取 修改人体背包属性描述不正确的问题

版本	日期	修订内容
		人体属性细分类添加预留字段的描述
v2.8	2020/11/10	增加 14.5 编辑抓拍机通道
v2.9	2021/1/27	增加 digestMaxAuthTries 和 digestUnlockTime 字段
v2.9	2021/4/19	为 3 个布控指令增加字段 bestStrangerUpload，控制是否上报全局最优陌生人
V2.9	2021/5/13	configs/detect 增加警戒模式
V2.9	2021/5/17	增加警戒模式布控协议、查询抓拍增加字段
V2.9	2021/5/31	GAT1400 客户端协议增加是否上传背景图字段
V2.9	2021/6/9	增加警戒模式下的 HTTP 主动上报，对接盘古
V2.9	2021/7/5	警戒布控协议增加字段 alertRulesValidity
V2.9	2021/7/16	车辆警戒布控增加检测目标类型字段； 查询抓拍回复增加目标类型字段。
v2.9	2021/7/22	结构化布控推图模式增加无限推图
v2.9	2021/8/20	单路多算法功能协议相关。结构化布控协议扩展，新增警戒多子规则布控协议，新增多算法布控参数获取协议
V2.9	2021/08/24	新增 GUI/layout 协议控制本地 GUI 输出
V2.9	2021/08/26	新增单路多算法布控使能开关协议
V2.9	2021/09/01	新增警戒下的 websocket 主动上报，警戒查询新增 rule_id、area_id 字段
B4H-v1.1	2021/9/28	增加 debug 相关 API 大标题，增加 debug_ping 相关 API

版本	日期	修订内容
B4H-v1.1	2021/9/29	再增加语音报警相关 API
B4H-v1.2	2021/11/11	视图分析增加分数上报
B4H-v1.3	2021/12/13	算法仓相关接口更新
B4H-v1.3	2021/12/15	更新 26.4 算法能力集获取接口章节内容。 新增 26.5 事件规则卡片属性获取接口章节内容。 新增 26.6 算法仓数据字典章节内容。

1 概述

1.1 概述

本文主要介绍智能分析盒人脸识别终端 Web API 的功能与调用方法，MegBox 提供丰富的 API 接口，可在智能设备上通过网络接口实现对智能分析盒终端的控制与各种功能的调用。方便用户进行二次开发与业务服务。

1.2 调用方式

智能分析盒终端提供 RESTful API 模式与客户端进行通信，客户端的所有操作与数据也都是通过网口进行数据的交互。RESTful API 简单说明：

- 使用协议：HTTP/HTTPS
- 访问地址：设备 IP 地址与端口
- 版本：API 的版本信息
- 路径：表示 API 指向的具体资源
- URL 格式：https:// IP/版本号/接口类型/接口资源?参数

示例：http://192.168.1.100/v1/MEGBOX/faceGroups

对于资源 RESTful API 的具体操作类型：

- GET (SELECT)：从服务器取出资源（一项或多项）。
- POST (CREATE)：在服务器新建一个资源。
- PUT (UPDATE)：在服务器更新资源（客户端提供改变后的完整资源）。
- PATCH (UPDATE)：在服务器更新资源（客户端提供改变的属性）。
- DELETE (DELETE)：从服务器删除资源。

*当前 API 接口为单线程模式，不支持并发操作

1.3 安全机制

为了访问设备的安全性，如接入公网需要使用 HTTPS 协议。如在私有网络或局域网内，可使用 HTTP 协议，具体方式由用户选择。

1.4 多用户登录互斥机制

对于设备，有两类操作方式，一是通过 Web 页面设置，二是通过 RESTful API 接口操作。

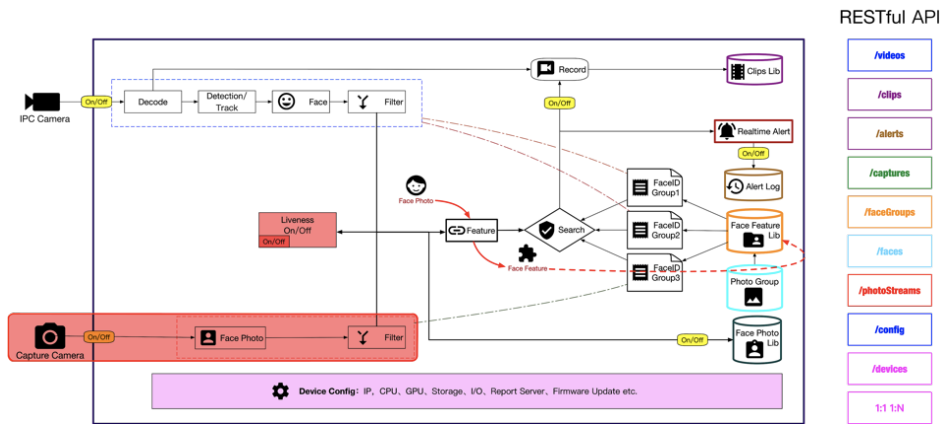
主动上报接口分三类，报警、抓拍、心跳。报警与抓拍上报按视频通道部署，而心跳接口是全局的，一个设备就一个，采用 socket 接口方向主动发起链接。

设备允许支持三个用户同时登录，一个是 web 页面，二个是 RESTful API 接口，RESTful 接口两个用户，是为适应“边”上系统与云端系统同时调用时的需求。

接口类型	优先级	互斥
设置接口 POST/PUT	高	在设置接口被调用，并且没有运行完成时，将不允许其它接口再被调用，如其它任何接口调用将返回“104004”无访问权限。
修改/删除接口 PATCH/DELETE	中	如调用修改删除接口，而设置接口没有被调用时，允许被调用，但如过在执行过程中调用了设置接口，将停止该接口的操作，返回“104004”无访问权限，并执行设置接口操作。

查询接口 GET	低	在其它接口没有被调用的时候，允许调用该接口，如在运行时，其它接口开始调用，将停止该接口操作，返回“104004”无访问权限，并执行其它接口的操作。
-------------	---	---

2 Web API 说明



2.1 API 资源列表

序号	资源名	说明
1	videos	视频流，接入 RTSP 流并进行布点工作
2	faces	人脸照片，指人脸图像信息。将人脸进行入库、查询与删除等操作。
3	faceGroups	人员分组，人脸照片的集合。
4	devices	指 MEGBOX 硬件设备，对设备进行配置
5	alerts	报警信息上报与查询删除

序号	资源名	说明
6	captures	抓拍人脸资源
7	configs	公共参数的设置
8	search/compare	人脸对比
9	keyword	登录验证密码设置
10	access	规则权限
11	relay	继电器相关配置
12	bindRelation	视频结构化绑定关系查

2.2 接口功能列表

2.2.1 configs 公共参数设置 API 列表

序号	接口	操作	功能
1	/configs	GET	获得加载模型参数与版本，功能模块开关状态，报警视频、抓拍照片、报警缓存、二次识别、陌生人 retry、背景大图压缩等级配置
2	/configs	POST	设置加载模型参数，功能模块开关状态，报警视频、抓拍照片、报警缓存、二次识别、陌生人 retry、背景大图压缩等级配置

2.2.2 devices 硬件设备资源操作 API 列表

	接口	操作	功能
1	/devices	GET	获取设备状态
2	/devices	POST	设置设备状态
3	/devices/reports	POST	WS 模式主动上报设置
4	/devices/reports	GET	WS 模式主动上报参数查询
5	/devices/firmware	POST	固件升级
6	/devices/reboot	GET	重启设备
7	/devices/restoreFactorySet	GET	恢复出厂设置
8	/devices/formateDisk	DELETE	格式化存储容间
9	*设备心跳上报接口	—	主动上报设备状态
10	/devices/httpReports	POST	http 模式主动上报设置
11	/devices/httpReports	GET	http 模式主动上报参数查询

2.2.3 faces 人脸入库操作 API 列表

序号	接口	操作	功能
1	/faces	POST	创建/注册人脸
2	/faces	GET	查询人脸库所有人脸列表
3	/faces/faceToken/{faceToken}	DELETE	删除某一人脸
4	/faces/image/{imageId}	DELETE	删除某一原始照片

序号	接口	操作	功能
5	/faces//faceToken{faceToken}	GET	获取某一人脸信息与图片
6	/faces/image/{imageId}	GET	下载某原始照片
7	/faces/faceLib	DELETE	初始化人脸库
8	/face/batch	POST	批量人脸入库

2.2.4 faceGroups 人脸分组操作 API 列表

序号	接口	操作	功能
1	/faceGroups	GET	获取人脸分组列表
2	/faceGroups	POST	创建人脸分组
3	/faceGroups/{groupName}	DELETE	删除某人脸分组
4	/faceGroups/{groupName}	GET	查询某分组绑定人脸的列表
5	/faceGroups/{groupName}/{faceToken}	POST	绑定人脸到某一分组
6	/faceGroups/{groupName}/{faceToken}	DELETE	解绑人脸与分组关系
7	/ faceGroups / binding / {faceToken}	POST	批量关联分组
8	faceGroups / unbind / {faceToken}	DELETE	解绑人脸与分组关系

2.2.5 videos 视频流布点资源操作 API 列表

	接口	操作	功能
1	/videos	POST	接入一个视频
2	/videos/{videoNo}/surveillances	PUT	创建某视频的布控
3	/videos	GET	获取所有视频布点信息
4	/videos/{videoName}	DELETE	删除视频布点
5	/ videos / {videoNo} / snapFullImage	GET	视频布控获取背景图

2.2.6 alerts 报警/通行资源操作 API 列表

序号	接口	操作	功能
1	/alerts	GET	获取报警/通行事件列表
2	/alerts/{alertsId}	DELETE	删除某个报警信息
3	*主动上报报警接口	—	实时主动上报报警/通行消息

2.2.7 captures 抓拍人脸资源操作 API 列表

序号	接口	操作	功能
1	/captures	GET	获取抓拍事件列表
2	/captures/{trackId}	GET	获取某个抓拍信息
3	/captures/{trackId}	DELETE	删除某个抓拍信息
4	*主动上报抓拍接口	—	实时主动上报抓拍数据

2.2.8 人脸比对与搜索 API 列表

序号	接口	操作	功能
1	/compare	POST	1: 1 人脸比较, 得到两张图的相似度
2	/search	POST	1: N 一张图与人脸分组搜索, 得到 TOP5 的结果

2.2.9 用户接入密码修改 API 列表

序号	接口	操作	功能
1	keyword	PUT	修改用户接入验证密码

2.2.10 假日规则 API 列表

序号	接口	操作	功能
1	/access/calendar	GET	查询假日规则
2	/access/calendar	POST	增加假日规则
3	/access/calendar/{calendarName}	DELETE	删除假日规则
4	/access/calendar/{calendarName}	PUT	修改假日规则

2.2.11 时段规则 API 列表

序号	接口	操作	功能
----	----	----	----

1	/access/schedule	GET	查询时段规则
2	/access/schedule	POST	增加时段规则
3	/access/schedule/{scheduleName}	DELETE	删除时段规则
4	/access/schedule/{scheduleName}	PUT	修改时段规则

2.2.12 门禁权限规则 API 列表

序号	接口	操作	功能
1	/access/rules	GET	查询门禁规则
2	/access/rules	POST	增加门禁规则
3	/access/rules/{ruleName}	DELETE	删除门禁规则
4	/access/rules/{ruleName}	PUT	修改门禁规则

2.2.13 继电器 API 列表

序号	接口	操作	功能
1	/relay	POST	绑定继电器
2	/relay/{videoNo}	DELETE	删除继电器
3	/relay	GET	查询继电器状态

2.3 通用返回参数

如 API 有返回值时，都会包含以下内容，而 API 返回参数包含在 data 对象中。

参数	类型	说明
code	integer	返回代码，成功为 0，失败会返回错误码，具体参看错误码列表
message	string	错误消息，具体参看错误码列表
timestamp	integer	Unix 时间戳，服务器收到请求时的时间戳，单位为毫秒
timeCost	integer	设备接收到请求到完成任务消耗的时间，单位为毫秒
data {}	obj	API 返回的具体参数，如没有返回参数则为空

3 Config 公共参数设置 API

3.1.1 设置公共参数 POST/configs

操作类型：

POST

功能：

设置设备中公共的参数，包括模型与功能开关等设置。

接口 URI：

/configs

参数：

参数	类型	说明
modelSw{}	obj	工作模式配置。
—videoMode	string	选择设备工作在视频流分析模式，视频流前端接摄像机，而照片流前端接抓拍机。 参数：“videoStreamMode”为视频流模式，接普通 IPC RTSP 流（默认）
—workMode	string	“WhiteNameListMode”通行模式，一般用于白名单通行，对人脸识别精度要求高，要求误识率低（不可放不识的人进入）、响应速度快，一般配合“最快推图模式”使用，相对可以支持的底库人数少。 “BlackNameListMode”抓捕模式，（默认）黑名单安防，对人脸识别精度要求相对白名单低，要求漏识率低（不可放任何一个可能的坏人跑掉），响应速度不要求那么快，一般配合”最优推图模式”使用，相对可以支持更大的底库 重要说明：模式的改变将引起设备的初始化，清除全部底库与分组关联信息、全部布控信息第与底库相关联的设置。
—pushMode	string	“FastPushMode”最快推图模式，当人脸满足抓拍过滤条件后，立即推向后面 Search 模块进行比对，直到捕获到比之前人脸更优的人脸，再推到后端进行比对，比对次数按 retryCount 中设定值。 “BestPushMode”最优推图模式，（默认）当人脸进入视频后，只要被跟踪上，将在断跟踪（离开画面）后，推出一张这个过程中质量（角度、清晰度、分辨率综合判断分数）最好的一张。但有些情况人员会长时间停滞在摄像头前，为了保证一定实时性，人员进入视频后，10 秒推一张最优人脸，后一分钟推一张，直至人员离开。时序逻辑如下图： 

参数	类型	说明
—httpMode	string	“https” 安全模式，连接外网时推荐 “http” 普通模式，内网连接时使用（默认）
—libMode	bool	底库模式控制： “true”：代表打开特征入库； “false”：（默认）关闭特征入库，通过照片入库； 重要说明：模式的改变将引起设备的初始化，清除全部底库与分组关联信息、全部布控信息第与底库相关联的设置。
—digestMode	bool	- digest 认证功能是否启用： “true”：启用 digest 认证；（开启后，要求两次交互小于 60s） “false”：（默认）不启用 digest 认证。
—digestMaxAuthTries	integer	digest 登录次数限制，范围 3-10 次，默认 6 次
—digestUnlockTime	integer	digest 锁定后，解锁时间，范围 5-30 分钟，默认 30 分钟
funcModule {}	obj	各功能模块的使能操作
—capturePicEnabled	bool	是否存储抓拍照片 true/开，false/关，“WhiteNameListMode” 模式时（白名单）默认为关，黑名单模式默认为开。
—clipEnabled	bool	报警录像开关（暂不开放，可固定为 false）
—captureFullPicEnabled	bool	是否存储抓拍大图 true/开，false/关，“WhiteNameListMode” 模式时（白名单）默认为关，黑名单模式默认为开。

参数	类型	说明
—clipTime	integer	报警录像时间（暂不开放，可固定为0）
—detectType	integer	检测类型，0- 全人脸 ，1-混合模式，3-警戒模式，默认为全人脸，默认混合模式，非必选
—alertReportBufferSize	integer	报警消息缓存数据条数，如超过该条数将会从时间最老的一条开始删除。设置范围（0~10000），具体参看 Alerts 接口说明
—captureReportBufferSize	integer	抓拍消息缓存数据条数，如超过该条数将会从时间最久的一条开始删除。设置范围（0~50000），具体参看 Captures 接口说明
—fullPicCompressLevel	integer	背景大图压缩等级，取值范围(10~100)，最小单位为1
—fullPicDrawRectEnabled	bool	背景大图画框使能
—minPushInterval	integer	最小推图间隔，取值范围（500~10000 毫秒）
—pictureRetryCount	integer	图片流二次识别次数，取值范围（0~5000）
—pictureRetryInterval	integer	抓拍机最小识别间隔，取值范围（0~60 秒）
—pictureStrangerRetryCount	integer	图片流陌生人判定次数，取值范围（0~5）
—retryCount	integer	二次识别次数，取值范围（0~5000）
—retryInterval	integer	二次识别的间隔时长，取值范围（0~60 秒）
—strangerRetryCount	integer	陌生人重复次数，取值范围（0~5）
—rtspServerEnabled	bool	rtsp server 转发使能

返回值：

状态码	说明	内容
200	成功	

3.1.2 获取当前公共参数 GET/configs

操作类型：

GET

功能：

从设备中获取当前公共的参数列表，包括模型与功能开关等设置。

接口 URI：

/configs

参数：

无

返回值：

参数	类型	说明
MegFaceSDKVer	string	底层算法版本
model {}	obj	算法模型功能加载开关，不同算法模型实现不同的功能，如不打开将不实现该模型功能。

参数	类型	说明
—detectModelVer	string	人脸检测模型版本
—trackModelVer	string	人脸检测与跟踪的模型版本
—poseModelVer	string	人脸角度质量判断的模型版本
—blurModelVer	string	人脸模糊度质量判断模型版本
—searchModelVer	string	人脸识别模型版本
—livenessModelVer	string	人脸活体检测模型版本
funcModule {}	obj	各功能模块的使能操作
—capturePicEnabled	bool	是否存储抓拍照片 true/开，false/关
—clipEnabled	bool	报警录像开关（暂不开放）
—captureFullPicEnabled	bool	是否存储抓拍大图 true/开，false/关，“WhiteNameListMode”模式时（白名单）默认为关，黑名单模式默认为开。
—clipTime	integer	报警录像时间（暂不开放）
—detectType	integer	检测类型，0-全人脸，1-混合模式，3-警戒模式，默认为全人脸，默认混合模式
—alertReportBufferSize	integer	报警消息缓存数据条数，如超过该条数将会从时间最老的一条开始删除。设置范围（0~10000），具体参看 Alerts 接口说明
—captureReportBufferSize	integer	抓拍消息缓存数据条数，如超过该条数将会从时间最久的一条开始删除。设置范围（0~50000），具体参看 Captures 接口说明
—fullPicCompressLevel	integer	背景大图压缩等级，取值范围(10~100)，最小单位为 1
—fullPicDrawRectEnabled	bool	背景大图画框使能

参数	类型	说明
—minPushInterval	integer	最小推图间隔，取值范围（500~10000 毫秒）
—pictureRetryCount	integer	图片流二次识别次数，取值范围（0~5000）
—pictureRetryInterval	integer	抓拍机最小识别间隔，取值范围（0~60 秒）
—pictureStrangerRetryCount	integer	图片流陌生人判定次数，取值范围（0~5）
—retryCount	integer	二次识别次数，取值范围（0~5000）
—retryInterval	integer	二次识别的间隔时长，取值范围（0~60 秒）
—strangerRetryCount	integer	陌生人重复次数，取值范围（0~5）
—rtspServerEnabled	bool	rtsp server 转发使能
modelSw{}	obj	工作模式配置。
—videoMode	string	选择设备工作在视频流分析模式，视频流前端接摄像机，而照片流前端接抓拍机。 参数：“videoStreamMode”为视频流模式，接普通 IPC RTSP 流（默认）
—workMode	string	“WhiteNameListMode”通行模式，一般用于白名单通行，对人脸识别精度要求高，要求误识率低（不可放不识的人进入）、响应速度快，一般配合“最快推图模式”使用，相对可以支持的底库人数少。 “BlackNameListMode”抓捕模式，（默认）黑名单安防，对人脸识别精度要求相对白名单低，要求漏识率低（不可放任何一个可能的坏人跑掉），响应速度不要求那么快，一般配合”最优推图模式”使用，相对可以支持更大的底库

参数	类型	说明
		重要说明：模式的改变将引起设备的初始化，清除全部底库与分组关联信息、全部布控信息第与底库相关联的设置。
—pushMode	string	“FastPushMode”最快推图模式，当人脸满足抓拍过滤条件后，立即推向后面 Search 模块进行比对，直到捕获到比之前人脸更优的人脸，再推到后端进行比对，比对次数按 retryCount 中设定值。 “BestPushMode”最优推图模式，当人脸进入视频后，只要被跟踪上，将在断跟踪（离开画面）后，推出一张这个过程中质量（角度、清晰度、分辨率综合判断分数）最好的一张
—httpMode	string	“https” 安全模式，连接外网时推荐 “http” 普通模式，内网连接时使用
—libMode	bool	底库模式控制： “TURE”：代表打开特征入库； “FALSE”：（默认）关闭特征入库，通过照片入库；
—digestMode	bool	digest 认证功能是否启用： “true”：digest 认证启用； “false”：digest 认证未启用；
—digestMaxAuthTries	integer	digest 登录次数限制
—digestUnlockTime	integer	digest 锁定后，解锁时间

返回值:

状态码	说明	内容
200	成功	

响应数据示例:

```
"MegFaceSDKVer": "v1.0.0",
  "model": {
    "detectModelVer": "",
    "trackModelVer": "",
    "poseModelVer": "pose_blur_v1.0.1",
    "blurModelVer": "pose_blur_v1.0.1",
    "searchModelVer": "",
    "livenessModelVer": "liveness_v1.0.0"
  },
  "funcModule": {
    "capturePicEnabled": true,
    "clipEnabled": false,
    "captureFullPicEnabled": true,
    "clipTime": 10,
    "retryCount": 5,
    "retryInterval": 5,
    "strangerRetryCount": 0,
    "minPushInterval": 500,
```

```
    "alertReportBufferSize":10000,  
    "captureReportBufferSize":50000,  
    "fullPicCompressLevel":80  
  },  
  "modelSw":{  
    "workMode":"WhiteNameListMode",  
    "httpMode":"http",  
    "digestMode":false  
  }  
}
```

3.1.3 获取当前工作模式 GET /configs/detect

操作类型:

GET

功能:

从设备中获取当前工作模式好支持的工作模式的参数列表。

接口 URI:

/configs/detect

参数:

无

返回值:

参数	类型	说明
type	int	当前工作模式
support	array[]	0: 全人脸模式 , 1: 混合模式 3: 警戒模式

返回类型:

状态码	说明	内容
200	成功	

响应数据示例:

```
{
    "type":0,
    "support":[0, 1]
}
```

3.1.4 设置当前工作模式 POST /configs/detect

操作类型:

POST

功能:

设置设备的工作模式。

接口 URI:

/configs/detect

参数:

参数	类型	说明
type	int	当前工作模式，0:全人脸模式，1: 混合模式 3: 警戒模式

返回值:

状态码	说明	内容
200	成功	

3.1.5 设置 MegConnect 工作参数

操作类型:

POST

功能:

设置 MegConnect 工作参数。

接口 URL:

/devices/megConnect

参数:

参数	类型	说明
enable	bool	是否使能主动模式。
url	string	主动模式地址, ip+port, tcp://127.0.0.1:34567

返回值:

状态码	说明	内容
200	成功	

3.1.6 获取 MegConnect 工作参数

操作类型:

GET

功能:

获取 MegConnect 工作参数。

接口 URL:

/devices/megConnect

参数:

参数	类型	说明
enable	bool	主动模式使能状态。
url	string	主动模式地址，ip+port，tcp://127.0.0.1:34567
port	int	被动模式端口号，不可修改。

返回值：

状态码	说明	内容
200	成功	

4 Device 硬件设备管理 API

4.1.1 获取硬件设备状态 GET/devices

操作类型：

GET

功能：

获取设备硬件基础状态，包括设备的 CPU、内存、存储的状态与使用率，设备时间，设备号，设备固件版本，设备心跳上报服务器地址等。

接口 URI：

/devices

参数:

无

返回参数:

参数	类型	说明
hardwareVer	string	硬件版本
firmwareVer	string	固件版本
deviceId	string	设备编号，唯一代码（核心板）
deviceModel	string	设备型号，（eg: MegCube-B4H16-311）
channelNum	int	通道数量（1~16）
licenseKey	string	设备授权码，最多 64 字节
serialNumber	string	设备 SN 号，最多 64 字节（整机）
megConnectVersion	string	MegConnect 协议 SDK 版本号（eg: V2.1.1.A35）（非必选）
deviceHeartbeat {}	obj	心跳数据上传设置，心跳数据包含：DeviceID、时间等信息，具体参照“主动上报数据说明”，如内容全空将不会主动上报
—serverAdd	string	上传服务器域名或 IP 地址，可指定端口，用“:”分隔：服务器地址示例： ws://domain(ip):port/
—htInterval	integer	访问心跳接上报的时间间隔，（1~3600）单位秒
hardwareStatus {}	obj	硬件当前状态
—cpuUseRate	integer	CPU 使用率，百分比

参数	类型	说明
—cpuTemp	integer	CPU 温度，摄氏度
—gpuTemp	integer	GPU 温度，摄氏度
—caseTemp	integer	电路板温度，摄氏度
—memSize	integer	内存总大小，单位 MB
—memUse	integer	内存已使用大小，单位 MB
—storageSize	integer	存储磁盘总大小，单位 MB
—storageUseSize	integer	存储磁盘已使用大小，单位 MB
time {}	obj	时间相关参数
—deviceTime	integer	设备当前时间，以 unix 时间戳表示，1970/1/1,0 点开始计的秒
—ntpEnable	bool	true/开，false/关
—ntpServerAdd	string	上传服务器域名或 IP 地址，可指定端口，用 “:” 分隔，SNTP 协议支持 RFC 2030 - Simple Network Time Protocol (SNTP) Version 4
—ntpInterval	integer	访问 NTP 服务器的时间间隔，单位分钟（1~65535）
—ntpFailCount	integer	NTP 校时错误次数，0 表示无错误（最大 65535）
—timeZone	integer	index 对应实际的时差，具体参考下表，默认为 27(时差+8)
eth {}	obj	以太网口 0
—ethNetMode	int	0-DHCP；1-静态 IP
—ethIpAddr	string	有线网 IP 地址

参数	类型	说明
—ethSubmask	string	有线子网掩码
—ethGateway	string	有线网关
—ethDns	string	有线 DNS 服务器地址
—ethMac	string	有线网卡 MAC 地址
eth1 {}	obj	以太网口 1，双网卡模式下显示
—ethNetMode	int	0-DHCP；1-静态 IP
—ethIpAddr	string	有线网 IP 地址
—ethSubmask	string	有线子网掩码
—ethGateway	string	有线网关
—ethDns	string	有线 DNS 服务器地址
—ethMac	string	有线网卡 MAC 地址
—ethBondMode	int	2 表示 bond 平衡模式；1 表示 bond 主备模式；7 表示双网卡独立使用

返回值：

状态码	说明	内容
200	成功	

响应数据示例：

```
"hardwareVer": "000_01_000",
  "firmwareVer": "v3.0.1",
  "deviceId": "78CA8340050600000000",
  "channelNum": 16,
  "licenseKey": "KOALA-DEVICE-B3R",
  "deviceHeartbeat": {
    "serverAdd": "",
    "htInterval": 0
  },
  "hardwareStatus": {
    "cpuUseRate": 34,
    "cpuTemp": 42,
    "gpuTemp": 42,
    "caseTemp": 42,
    "memSize": 4096,
    "memUse": 3033,
    "storageSize": 25442,
    "storageUseSize": 6316
  },
  "time": {
    "deviceTime": 1588816196,
    "ntpEnable": false,
    "ntpInterval": 1,
    "ntpFailCount": 0,
    "ntpServerAdd": "10.169.246.175",
    "timeZone": 27
  },
  "eth": {
    "ethNetMode": 1,
    "ethIpAddr": "192.168.172.212",
```

```
"ethSubmask":"255.255.255.0",
"ethGateway":"192.168.172.1",
"ethDns":"0.0.0.0",
"ethMac":"78:ca:83:40:05:06"
},

"eth1":{
  "ethNetMode":1,
  "ethIpAddr":"192.168.172.213",
  "ethSubmask":"255.255.255.0",
  "ethGateway":"192.168.172.1",
  "ethDns":"0.0.0.0",
  "ethMac":"78:ca:83:40:05:07",

  "ethBondMode":7
}
```

时区对照表:

index	时差（小时）	描述
0	-12	(UTC-12:00) 国际日期变更线西/International Date Line West
1	-11	(UTC-11:00) 中途岛，萨摩亚/Midway Island, Samoa
2	-10	(UTC-10:00) 夏威夷，阿留申群岛/Hawaii, Aleutian Islands
3	-9.5	(UTC-09:30) 马克萨斯群岛/Marquesas Islands

4	-9	(UTC-09:00) 阿拉斯加, 冈比亚/Alaska, Gambia
5	-8	(UTC-08:00) 太平洋标准时间 (美国和加拿大), 洛杉矶/Pacific Time (US & Canada), Los Angeles
6	-7	(UTC-07:00) 山地标准时间 (美国和加拿大), 亚利桑那/Mountain Time (US & Canada), Arizona
7	-6	(UTC-06:00) 中部标准时间 (美国和加拿大), 墨西哥城, 复活节岛/Central Time (US & Canada), Mexico City, Easter Island
8	-5	(UTC-05:00) 东部标准时间 (美国和加拿大), 海地/Eastern Time (US & Canada), Haiti
9	-4.5	(UTC-04:30) 委内瑞拉/Venezuela
10	-4	(UTC-04:00) 大西洋标准时间, 亚马逊/Atlantic Time , Amazon
11	-3.5	(UTC-03:30) 纽芬兰/Newfoundland
12	-3	(UTC-03:00) 阿根廷, 格陵兰, 萨尔瓦多, 布宜诺斯艾利斯/Argentina, Greenland, Salvador, Buenos Aires
13	-2	(UTC-02:00) 南乔治亚岛/South Georgia Islands
14	-1	(UTC-01:00) 佛得角群岛, 亚速尔群岛/Cape Verde Islands, Azores Islands
15	0	(UTC+00:00) 格林威治标准时间, 都柏林, 爱丁堡, 伦敦/Greenwich Mean Time, Dublin, Edinburgh, London
16	1	(UTC+01:00) 阿姆斯特丹, 柏林, 哥本哈根, 巴黎/Amsterdam, Berlin, Copenhagen, Paris
17	2	(UTC+02:00) 大马士革, 开罗, 雅典, 耶路撒冷/Damascus, Cairo, Athens, Jerusalem
18	3	(UTC+03:00) 巴格达, 莫斯科, 圣彼得堡, 伊斯坦布尔/Baghdad, Moscow, st. Petersburg, Istanbul

19	3.5	(UTC+03:30) 德黑兰/Tehran
20	4	(UTC+04:00) 阿布扎比, 伊热夫斯克, 萨马拉/Abu Dhabi, Izhevsk, Samara
21	4.5	(UTC+04:30) 喀布尔/Kabul
22	5	(UTC+05:00) 塔什干, 伊斯兰堡, 卡拉奇/Tashkent, Islamabad, Karachi
23	5.5	(UTC+05:30) 钦奈, 加尔各答, 孟买, 新德里/Chennai, Kolkata, Mumbai, New Delhi
24	6	(UTC+06:00) 孟加拉, 不丹/Bangladesh, Bhutan
25	6.5	(UTC+06:30) 仰光/Yangon
26	7	(UTC+07:00) 曼谷, 河内, 雅加达/Bangkok, Hanoi, Jakarta
27	8	(UTC+08:00) 北京, 重庆, 香港特别行政区, 乌鲁木齐, 台北/Beijing, Chongqing, Hong Kong, Urumqi, Taipei
28	9	(UTC+09:00) 东京, 平壤, 首尔/Tokyo, Pyongyang, Seoul
29	9.5	(UTC+09:30) 达尔文/Darwin
30	10	(UTC+10:00) 关岛, 堪培拉, 墨尔本, 悉尼/Guam, Canberra, Melbourne, Sydney
31	10.5	(UTC+10:30) 豪勋爵岛/Lord Howe Island
32	11	(UTC+11:00) 所罗门群岛/Solomon Islands
33	11.5	(UTC+11:30) 诺福克/Norfolk
34	12	(UTC+12:00) 马绍尔群岛时间, 奥克兰, 斐济/Marshall islands time, Auckland, Fiji
35	13	(UTC+13:00) 萨摩亚群岛, 菲尼克斯群岛/Samoa, Phoenix Islands
36	14	(UTC+14:00) 莱恩群岛/Line islands

4.1.2 设置硬件设备 POST/devices

操作类型:

POST

功能:

设置设备硬件基础，包括设备的设备时间，网络，设备心跳上报服务器地址等。时间服务器协议遵循 rfc2030(sntp v4)

接口 URI:

/devices

参数:

参数	类型	说明
deviceHeartbeat {}	obj	心跳数据上传设置，心跳数据包含：DeviceID、时间等信息，具体参照“主动上报数据说明”
—serverAdd	string	主动上报服务域名或 IP 地址，如公网建议使用域名方式，防止服务迁移或电信运营商调整导致的 IP 变更 服务器地址示例：ws://domain(ip):port/
—htInterval	integer	访问心跳接上报的时间间隔，(1~3600) 单位秒
time {}	obj	时间相关参数

参数	类型	说明
—deviceTime	integer	设备当前时间，以 unix 时间戳表示，1970/1/1,0 点开始计的秒
—ntpEnable	bool	true/开，false/关
—ntpServerAdd	string	上传服务器域名或 IP 地址，可指定端口，用 “:” 分隔，SNTP 协议支持 RFC 2030 - Simple Network Time Protocol (SNTP) Version 4
—ntpInterval	integer	访问 NTP 服务器的时间间隔，单位分钟（1~65535）
—timeZone	integer	index 对应实际的时差，具体参考上面的表格，默认为 27(时差+8)
eth{}	obj	有线网络接口 0 状态
—ethNetMode	integer	0-DHCP；1-静态 IP
—ethIpAddr	string	有线网 IP 地址
—ethSubmask	string	有线子网掩码
—ethGateway	string	有线网关
—ethDns	string	有线 DNS 服务器地址
eth1{}	obj	有线网络接口 1 状态，双网卡时必需，单网卡无用。
—ethNetMode	integer	0-DHCP；1-静态 IP
—ethIpAddr	string	有线网 IP 地址
—ethSubmask	string	有线子网掩码
—ethGateway	string	有线网关
—ethDns	string	有线 DNS 服务器地址

参数	类型	说明
—ethBondMode	int	2 表示 bond 平衡模式；1 表示 bond 主备模式；7 表示双网卡独立使用。 bond 模式时，bond 网卡使用 eth0 的配置，eth1 配置只用于记录，不生效。

返回值：

状态码	说明	内容
200	成功	

响应数据示例：

```
"hardwareStatus":{
  "cpuUseRate":1,
  "cpuTemp":33,
  "gpuTemp":33,
  "caseTemp":33,
  "memSize":4096,
  "memUse":2527,
  "storageSize":25442,
  "storageUseSize":1846
},
"time":{
  "deviceTime":1569813184,
  "ntpEnable":false,
  "ntpInterval":1,
  "ntpFailCount":0,
```



```
    "ntpServerAdd": "{",
    "timeZone": 27
  },
  "eth": {
    "ethNetMode": 1,
    "ethIpAddr": "192.168.1.100",
    "ethSubmask": "255.255.255.0",
    "ethGateway": "192.168.1.200",
    "ethDns": "10.7.7.7",
    "ethMac": "78:ca:83:40:04:d3"
  },
  "eth1": {
    "ethNetMode": 1,
    "ethIpAddr": "192.168.1.100",
    "ethSubmask": "255.255.255.0",
    "ethGateway": "192.168.1.200",
    "ethDns": "10.7.7.7",
    "ethMac": "78:ca:83:40:04:d3",

    "ethBondMode": 7
  }
}
```

4.1.3 ws 主动上报设置 POST / devices / reports

操作类型:

POST

功能:

设置报警与抓拍信息的主动上报服务地址与参数。当接收端收到消息后，需要回复消息 ID 与 ACK，具体参考下方主动上报消息章节。

接口 URI:

/devices/reports

参数:

参数	类型	说明
alertConfig{}	obj	报警/通行配置参数，如打开报警，以下必填
— alertReportMode	string	1 “activeMode”：主动模式 主动上报 alert 消息，上报成功将删除该消息，接收服务器必须应答 200 消息代表成功。主动模式 reportServerURL 或 reportServerIP 两者必填一项，如有 reportServerURL 将以此为准。主动模式下，被动模式依然有效，如不填写 IP 或域名，将无法主动上报 1 “passiveMode”：被动模式 轮询方式查询 alert 消息队列，如查询成功将删除该消息，被动模式中只有 reportBufferSize 参数有效
— reportServerAdd	string	主动上报服务域名或 IP 地址，如公网建议使用域名方式，防止服务迁移或电信运营商调整导致的 IP 变更 服务器地址示例（用’:’分隔端口）：ws://domain(ip):port/
—alertInterval	integer	主动上报不成功重发的时间间隔，单位秒，范围：0-10

—alertResend	integer	主动上报不成功重发的尝试次数，范围限定为 0，1 和 2 为合法
—pushImageSW	bool	主动上片报时传送图片开关，如打开，同时通过接口上传图片信息。如关闭，将不传图片数据，全为空。
—pushBigImageSW	bool	主动上片报时传送全景图片开关，可选，开启主动上片报时传送图片总开关然后再开启该开关才会上报全景图。
—pushSmallImageSW	bool	主动上片报时传送小图开关，可选，开启主动上片报时传送
capture {}	obj	抓拍参数设置，当 capturePicEnabled 有效后，下面参数才需要设置
—captureReportMode	string	1 “activeMode”：主动模式 主动上报抓拍消息，上报成功将删除该消息，接收服务器必须应答 200 消息代表成功。主动模式 reportServerURL 或 reportServerIP 两者必填一项，如有 reportServerURL 将以此为准。如不填写 IP 或域名，将无法主动上报 1 “passiveMode”：被动轮询模式 轮询方式查询抓拍消息队列，主动模式下，抓拍模式也有同时有效，如信息被取走，将会删除。
—reportServerAdd	string	主动上报服务域名或 IP 地址，如公网建议使用域名方式，防止服务迁移或电信运营商调整导致的 IP 变更 服务器地址示例（用’：’分隔端口）：ws://domain(ip):port/
—pushImageSW	bool	主动上片报时传送图片开关，如打开，同时通过接口上传图片信息。如关闭，将不传图片数据，全为空。

—pushBigImageSW	bool	主动上片报时传送全景图片开关，可选，开启主动上片报时传送图片总开关然后再开启该开关才会上报全景图。
—pushSmallImageSW	bool	主动上片报时传送小图开关，可选

返回值：

状态码	说明	内容
200	成功	

响应数据示例：

```
"alertConfig":{
    "alertReportMode":"activeMode",
    "reportServerAdd":"ws://192.168.1.190:80/",
    "alertInterval":0,
    "alertResend":2,
    "pushImageSW":false
},
"capture":{
    "captureReportMode":"passiveMode",
    "reportServerAdd": "",
    "pushImageSW":false
}
```

错误码：

状态码	错误码	消息	说明
400	104000	ILLEGAL_JSON_BODY	json 结构解析错误
400	104001	ILLEGAL_ARGUMENT:<args>	参数格式错误，<args>返回错误参数名，逗号分隔
400	104002	MISSING_ARGUMENT:<args>	缺少必要参数，<args>返回缺少参数名，逗号分隔
400	104006	REQUEST_TIMEOUT	请求超时
400	104009	NO_VIDEO	无指定视频通道，未创建或通道与名称不匹配

更多参见错误码列表

4.1.4 ws 主动上报参数查询 GET /devices/ reports

操作类型：

GET

功能：

查询报警与抓拍信息的主动上报服务地址与参数。

接口 URI：

/devices/reports

参数：

无

返回参数：

参数	类型	说明
alertConfig{}	obj	报警/通行配置参数，如打开报警，以下必填
—alertReportMode	string	1 “activeMode”：主动模式 主动上报 alert 消息，上报成功将删除该消息，接收服务器必须应答 200 消息代表成功。主动模式 reportServerURL 或 reportServerIP 两者必填一项，如有 reportServerURL 将以此为准。主动模式下，被动模式依然有效，如不填写 IP 或域名，将无法主动上报 1 “passiveMode”：被动模式 轮询方式查询 alert 消息队列，如查询成功将删除该消息，被动模式中只有 reportBufferSize 参数有效
—reportServerAdd	string	主动上报服务域名或 IP 地址，如公网建议使用域名方式，防止服务迁移或电信运营商调整导致的 IP 变更 服务器地址示例（用‘:’分隔端口）：ws://domain(ip):port/
—alertInterval	integer	主动上报不成功重发的时间间隔，单位秒
—alertResend	integer	主动上报不成功重发的尝试次数
—pushImageSW	bool	主动上片报时传送图片开关，如打开，同时通过接口上传图片信息（Base64 格式）。如关闭，将不传图片数据，全为空。
—pushBigImageSW	bool	主动上片报时传送全景图片开关，可选，开启主动上片报时传送图片总开关然后再开启该开关才会上报全景图。

—pushSmallImageSW	bool	主动上片报时传送小图开关，可选，开启主动上片报时传送图片总开关然后再开启该开关才会上报小图。
capture {}	obj	抓拍参数设置，当 capturePicEnabled 有效后，下面参数才需要设置
—captureReportMode	string	1 “activeMode”：主动模式 主动上报抓拍消息，上报成功将删除该消息，接收服务器必须应答 200 消息代表成功。主动模式 reportServerURL 或 reportServerIP 两者必填一项，如有 reportServerURL 将以此为准。如不填写 IP 或域名，将无法主动上报 1 “passiveMode”：被动轮询模式 轮询方式查询抓拍消息队列，主动模式下，抓拍模式也有同时有效，如信息被取走，将会删除。纯被动模式中 reportServerURL、reportServerIP、reportServerPort 三个参数无用
—reportServerAdd	string	主动上报服务域名或 IP 地址，如公网建议使用域名方式，防止服务迁移或电信运营商调整导致的 IP 变更 服务器地址示例（用‘:’分隔端口）：ws://domain(ip):port/
—pushImageSW	bool	主动上片报时传送图片开关，如打开，同时通过接口上传图片信息（Base64 格式）。如关闭，将不传图片数据，全为空。
fullImage —pushBigImageSW	bool	主动上片报时传送全景图片开关，可选，开启主动上片报时传送图片总开关然后再开启该开关才会上报全景图。
—pushSmallImageSW	bool	主动上片报时传送小图开关，可选，开启主动上片报时传送图片总开关然后再开启该开关才会上报小图。

返回值：

状态码	说明	内容
200	成功	

响应数据示例:

```
"alertConfig":{
    "alertReportMode":"activeMode",
    "reportServerAdd":"ws://192.168.1.190:80/",
    "alertInterval":0,
    "alertResend":2,
    "pushImageSW":false
},
"capture":{
    "captureReportMode":"passiveMode",
    "reportServerAdd":"",
    "pushImageSW":false
}
```

4.1.5 设置 HTTP 主动上报 POST /devices/httpReports

操作类型:

POST

功能: 设置 HTTP 模式主动上报参数

接口 URI: /devices/httpReports

请求参数:

参数	类型	说明
enable	bool	是否开启信息主动上报
main	obj	上报主设备相关设置
—url	string	上报设备 URL 地址
—captureUpload	bool	是否上传抓拍图
—smallCaptureUpload	bool	是否上传小图 可选
—bigCaptureUpload	bool	是否上传背景大图 可选
—eventType	integer	上报数据类型：1，抓拍；2，识别；3，抓拍+识别。默认识别
—resend	integer	上报失败后尝试次数（0~3），默认 1
—interval	integer	上报重传尝试间隔，单位秒，范围 1~5，默认 1 秒
— featureUpload	bool	是否上传 feature 数据
—persistReport	bool	是否开启信息持续上报
slave	obj	上报从设备相关设置
—url	integer	上报设备 URL 地址
—captureUpload	bool	是否上传抓拍图
—smallCaptureUpload	bool	是否上传小图 可选
—bigCaptureUpload	bool	是否上传背景大图 可选
—eventType	integer	上报数据类型，同上

参数	类型	说明
—resend	integer	上报失败后尝试次数，同上
—interval	integer	上报重传尝试间隔，同上
— featureUpload	bool	是否上传 feature 数据

返回参数：

无

4.1.6 查询 HTTP 主动上报 GET /devices/httpReports

操作类型：

GET

功能：查询 HTTP 主动上报

请求参数：

无

返回参数：

参数	类型	说明
enable	bool	是否开启信息主动上报
main	obj	上报主设备相关设置
—url	string	上报设备 URL 地址
—captureUpload	bool	是否上传抓拍图
—smallCaptureUpload	bool	是否上传小图
—bigCaptureUpload	bool	是否上传背景大图
—eventType	integer	上报数据类型：1，抓拍；2，识别；3，抓拍+识别。默认识别
—resend	integer	上报失败后尝试次数（0~3），默认 1
—interval	integer	上报重传尝试间隔，单位秒，范围 1~5，默认 1 秒
—featureUpload	bool	是否上传 feature 数据
—persistReport	bool	是否开启信息持续上报
slave	obj	上报从设备相关设置
—url	integer	上报设备 URL 地址
—captureUpload	bool	是否上传抓拍图
—smallCaptureUpload	bool	是否上传小图
—bigCaptureUpload	bool	是否上传背景大图
—eventType	integer	上报数据类型，同上
—resend	integer	上报失败后尝试次数，同上

参数	类型	说明
—interval	integer	上报重传尝试间隔，同上
—featureUpload	bool	是否上传 feature 数据

4.1.7 固件升级 POST/devices/firmware

操作类型：

POST

功能：

上传更新的固件文件，上传成功后，在固件升级时设备会自动重启。

接口 URI：

/devices/firmware

请求参数：

参数	类型	说明
fileSize	string	固件升级包文件大小
firmwareFile	binary (file upload)	固件升级包
fileMd5	string	固件升级包的 md5 值（可选）

返回值：

状态码	说明	内容
200	成功	

失败

状态码	错误码	消息	说明
400	104401	SERVER_BUSY	服务器忙
400	104403	FIRMWARE_TOO_BIG	升级包文件大小太大
400	104404	FIRMWARE_RECV_ERROR	接收升级文件错误
400	104405	FIRMWARE_CHECK_ERROR	升级包格式错误
400	104406	FIRMWARE_MD5_ERROR	升级包 MD5 检查失败

id 注：所有返回接口的错误码，参考错误码列表

4.1.8 设备重启 GET/devices/reboot

操作类型：

GET

功能：

设备重新启动。

接口 URI：

/devices/reboot

请求参数:

无

返回参数:

无

4.1.9 恢复出厂设置 GET/devices/restoreFactorySet

操作类型:

GET

功能:

恢复出厂设置，清理内部所有数据与配置，恢复出厂状态。但固件版本不会回退到出厂状态。如果携带参数 reserveNet=true，则保留 IP 等网络配置

接口 URI:

/devices/restoreFactorySet

请求参数:

- 1 无参数，恢复出厂状态，不保留 IP 等网络配置
- 2 参数 reserveNet=false，不保留 IP 等网络配置

3 参数 reserveNet=true, 保留 IP 等网络配置

返回参数:

无

4.1.10 FTP 升级设置 POST /ftpUpgrade

操作类型:

POST

功能:

按照设定的时间段, 自动从设定的 ftp 服务器获取升级包升级

接口 URI:

/ftpUpgrade

请求参数:

参数	类型	说明
enableAuto	bool	是否启用自动升级
dayOfWeek	array	升级日期 数组大小为 7 分别代表 7 天, 填 0 表示未选, 非 0 表示已选
timeStart	integer	升级时间段的开始时间

参数	类型	说明
timeEnd	integer	升级时间段的结束时间
serverIp	string	服务器地址 127 字节
serverPath	string	服务器路径 255 字节
username	string	用户名 31 字节
password	string	密码 63 字节

返回值：

状态码	说明	内容
200	成功	

失败

状态码	错误码	消息	说明
400	104401	SERVER_BUSY	服务器忙
400	104001	ILLEGAL_ARGUMENT:<args>	参数格式错误，<args>返回错误参数名，逗号分隔
400	104002	MISSING_ARGUMENT:<args>	缺少必要参数，<args>返回缺少参数名，逗号分隔

4.1.11 FTP 立即触发升级 PUT/ftpUpgrade

操作类型:

PUT

功能:

立即从设定的 ftp 服务器获取升级包升级

接口 URI:

/ftpUpgrade

请求参数:

参数	类型	说明
enableAuto	bool	是否启用自动升级, 参数无效
dayOfWeek	array	升级日期 数组大小为 7 分别代表 7 天, 填 0 表示未选, 非 0 表示已选, 参数无效
timeStart	integer	升级时间段的开始时间, 参数无效
timeEnd	integer	升级时间段的结束时间, 参数无效
serverIp	string	服务器地址 127 字节
serverPath	string	服务器路径 255 字节
username	string	用户名 31 字节

参数	类型	说明
password	string	密码 63 字节
filename	string	升级包文件名 63 字节

返回值：

若请求参数 filename 为空则查找 ftp 服务器上的升级文件，并返回：

参数	类型	说明
filename	string	升级包文件名

若请求参数 filename 非空，无返回参数，并立即触发升级

状态码	说明	内容
200	成功	

失败

状态码	错误码	消息	说明
400	104401	SERVER_BUSY	服务器忙
400	104001	ILLEGAL_ARGUMENT:<args>	参数格式错误，<args>返回错误参数名，逗号分隔
400	104002	MISSING_ARGUMENT:<args>	缺少必要参数，<args>返回缺少参数名，逗号分隔
400	104410	FTP_UPGRADE_USER_PASSWORD_ERR	用户名密码校验失败
400	104411	FTP_UPGRADE_ADDR_CONNECT_ERR	服务器连接失败
400	104412	FTP_UPGRADE_PATH_IS_NOT_EXIST	目录不存在

4. 1. 12 查询 FTP 升级 GET/ftpUpgrade

操作类型:

GET

功能:

查询 ftp 服务器升级的参数设置

接口 URI:

/ftpUpgrade

请求参数：

无

返回参数：

参数	类型	说明
enableAuto	bool	是否启用自动升级
dayOfWeek	array	升级日期 数组大小为 7 分别代表 7 天，填 0 表示未选， 非 0 表示已选
timeStart	integer	升级时间段的开始时间
timeEnd	integer	升级时间段的结束时间
serverIp	string	服务器地址 127 字节
serverPath	string	服务器路径 255 字节
username	string	用户名 31 字节

参数	类型	说明
password	string	密码 63 字节

4.1.13 查询设备繁忙操作状态 /devices/busyOperationStatus

操作类型:

GET

功能:

查询设备状态

接口 URI:

/devices/busyOperationStatus

参数:

无

返回值:

参数	类型	说明
operation	integer	正在进行的操作 0:完成 1:下载文件 2:校验文件 3:重提特征 4:批量入库
rate	integer	百分比

4.1.14 查询最后一次升级状态 /devices/upgradeEndStatus

操作类型:

GET

功能:

查询最后一次升级状态

接口 URI:

/devices/upgradeEndStatus

参数:

无

返回值:

参数	类型	说明
type	integer	升级类型: 0:web 升级 1:手动 Ftp 2:自动 Ftp
result	integer	0:成功 1:校验失败 2:网络问题 3:无可用升级文件
version	string	版本号
timestamp	integer	升级时时间 Unix 时间戳, 单位为毫秒

4.1.15 查询繁忙操作状态 /devices/busyOperationStatus

操作类型:

GET

功能:

查询繁忙操作状态

接口 URI:

/devices/busyOperationStatus

参数:

无

返回值:

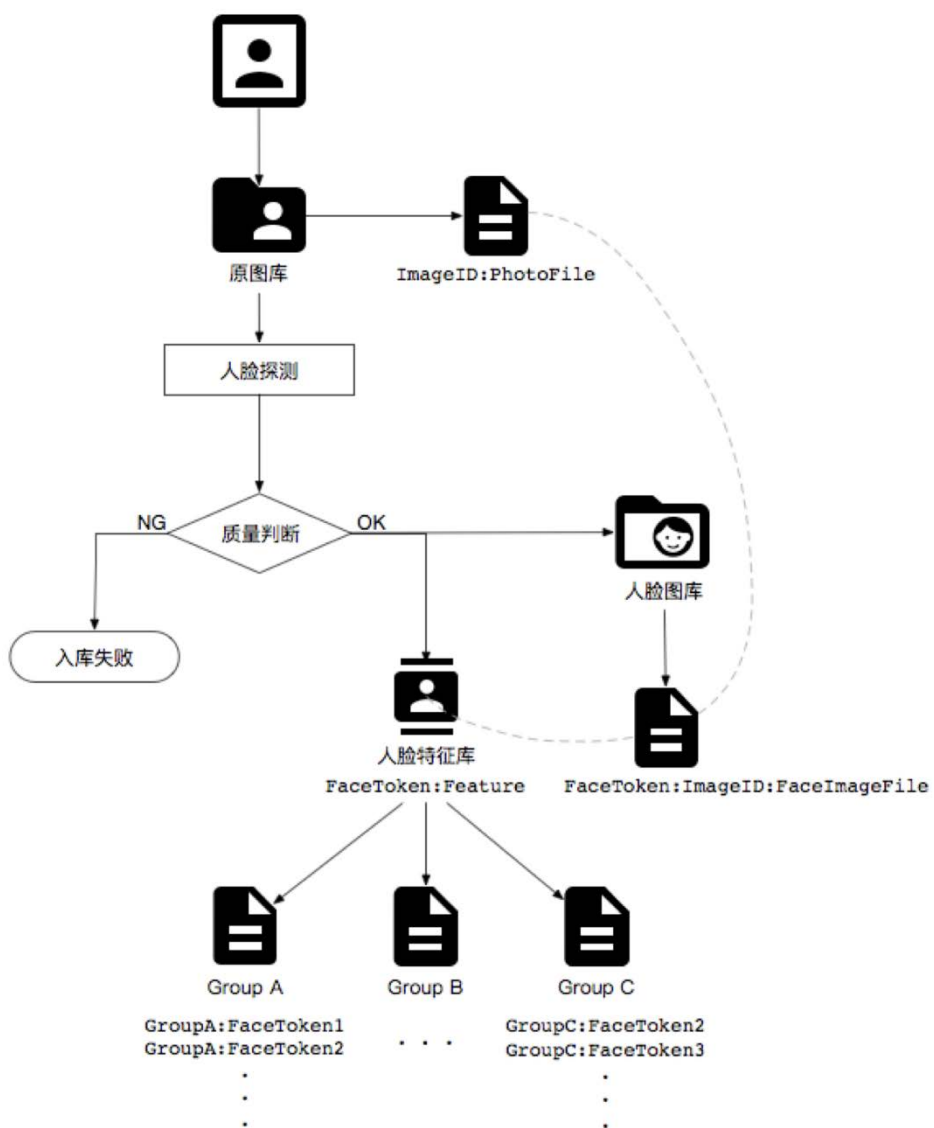
参数	类型	说明
operation	integer	正在进行的操作类型: 0:完成 1:下载文件 2:校验文件 3:重提特征 4:批量入库
rate	integer	上述操作进行的百分比(0-100), 如果上述状态为0(完成), 此项忽略

5 Faces 人脸入库 API

一张含有人脸照片在入库时，需要经历人脸探测、质量判断等多个流程，在不同流程中会都会有不同的记录产生，直至与某一人脸分组相关联，一个人脸可通过一个 FaceToken 与多个分组相关联。如下图，其中虚线代表为关联关系，如删除其中一个，将会删除相关的所有信息，也包括与人脸分组的绑定关系。

人脸入库与人脸删除接口支持多并发操作，可同时建立 10 条左右的链路进行入库与删除的批量操作，如超出系统可支持的链接数量，将返回链路数达到上限的错误。

图 5-1 人脸照片入库处理流程示意图



5.1.1 人脸注册/入库 POST/faces

操作类型:

POST

功能:

传入一张人脸照片到人脸库，照片限制:

1 图片类型: Jpg、Jpeg、PNG、BMP

1 图片大小 $\leq 4\text{MB}$ ，入库后直接抽取人脸图，如为白名单模式，将后压缩为 512KB 的图片进行存储备案，如黑名单模式，只存储抽取的人脸图;

1 图片最大尺寸 $\leq 4096*4096$

1 图片最小尺寸 $\geq 100*100$

1 如果照片中有多个个人脸，将会选择最大的一张人脸;

1 在入库时会做图片的质量判断，如质量差，将无法入库，主要包括人脸角度、模糊度、光线亮度与对比度;

上传图片成功后，会生成 `imageId`，来唯一指定该次上传的图片。后期用于上传历史查询。而上传的图片通过质量检测后，入库成功会返回一个 `faceToken` 为唯一标识，后面操作对人脸操作全依赖 `faceToken`。

接口 URI:

/faces

请求参数(form-data):

参数	类型	说明	必选
image	binary	上传的照片文件，图片限制参照接口功能描述	是
description	string	不超过 255 字节对照片文件的描述，UTF-8 编码	是
imageMd5	string	根据规则，基于图片的 md5 值生成的唯一标识	否

注：imageMd5 计算规则：

1. 先将文件生成 32 位的 md5 码，按每两位生成长度为 16 位的 16 进制数；
2. 将数组内每个 16 进制按 ASCII 码（可能有不可见字符），进行 Base64 转换；

最后将 base64 编码里的字符 ‘+’ 替换为 ‘-’ 、 ‘/’ 替换为 ‘_’ ，结果即为 imageId，与固件端生成结果一致。

JAVA 代码示例：



```

/*C代码示例*/
MD5_DIGEST_SIZE = 16 //MD5原始值长度

IVA_GROUP_IMAGE_TOKEN_LEN = 24 //imageMD5长度
/*****
ImageMD5的转换例程

Input:
    char* jpg_buf: 图片文件指针；
    int jpg_len: 图片长度；

Output:
    char* image_token: 用于存储imageMD5；
    int len : image_token的长度，正常使用IVA_GROUP_IMAGE_TOKEN_LEN；
*****/

```

```

int iva_group_stor_make_image_token(char* jpg_buf,int jpg_len,char* image_token,int len)
{
    Md5 md5_t = {0};
    unsigned char md5sum[MD5_DIGEST_SIZE] = {0};
    word32 out_len = len;
    int i = 0;

    InitMd5(&md5_t);
    Md5Update(&md5_t, (byte*)jpg_buf, jpg_len);
    Md5Final(&md5_t, md5sum); //图片转MD5
    if(Base64_Encode((unsigned char*)md5sum,MD5_DIGEST_SIZE,(unsigned char*)image_token,&out_len)!=0) //Base64转换
    {
        IVA_DBGPOS_ERR(IVA_FACE, "Base64_Encode err !!!\n");
        return RET_FAIL;
    }
    image_token[IVA_GROUP_IMAGE_TOKEN_LEN] = 0; //base64会在最后加上'\n'
    for(i = 0; i < IVA_GROUP_IMAGE_TOKEN_LEN; i++) //Base64中字符转换 '+'与 '/'转 '-'与 '_'
    {
        if(image_token[i] == '+')
        {
            image_token[i] = '-';
        }
        else if(image_token[i] == '/')
        {
            image_token[i] = '_';
        }
    }
    return RET_OK;
}

```

Python代码示例:

```
def get_Picture_ImageId(picPath, imageName):

Md5 = get_file_md5(picPath + "\\\" + imageName)
arr_md5 = []
for i in range(0, len(Md5), 2):
a = int(Md5[i:i + 2], 16)
arr_md5.append(a)
ImageId = (base64.b64encode(bytes(arr_md5)).decode()).replace("+", "-").replace('/', '_')
print(imageName + ' : ' + str(ImageId))
return ImageId

def get_file_md5(file_path):
if not os.path.isfile(file_path):
return
myhash = hashlib.md5()
f = open(file_path, 'rb')
while True:
b = f.read(8096)
if not b:
break
myhash.update(b)
f.close()
return myhash.hexdigest()
```

返回参数:

参数	类型	说明
imageId	string	上传的照片原图标识, 为 24 字节字符串, 用于调取原图, 仅在白名单有效
rect{}	obj	人脸在照片上的位置框, 用于人脸框显示与入库的质量判断, 检测到人脸即会返回改值。
—left	integer	脸部框左边界坐标
—top	integer	脸部框上边界坐标
—right	integer	脸部框右边界坐标

参数	类型	说明
—bottom	integer	脸部框下边界坐标
faceToken	string	通过质量监测，入库成功后，返回人脸的唯一标识，24 字节字符串，用于后面分组操作。入库不成功返回 0

返回值：

状态码	说明	内容
200	成功	

响应数据示例：

```
"imageId": "VsWi_xqPzx04ruLVdEoimQ==",
  "faceToken": "Eqv0cstHM_0DeNr-pAzA9Q==",
  "rect": {
    "left": 195,
    "top": 194,
    "right": 386,
    "bottom": 382
  }
```

错误码：

状态码	错误码	消息	说明
400	104200	FACE_NOT_FOUND	没有检测到人脸
400	104201	BAD_QUALITY:<reason>	人脸未通过质量判断，reason 为未通过的检测项目

状态码	错误码	消息	说明
400	104204	FACE_ALREADY_EXIST	人脸已经存在

更多错误类型参见错误码列表

5.1.2 查询人脸库所有人脸列表 GET/faces

操作类型:

GET

功能:

获取人脸注册库中人脸 imageID 与 faceToken 等信息列表。查询按时间排序输出，最新入库的在最后一个。

接口 URI:

/faces 在进行 URI 解析操作时，要按照 pageToken——>pageSize 的固定顺序，不支持随意顺序参数。

请求参数:

参数	类型	说明
pageOffset	integer	列表结果偏移量，从 0 开始，与 pageToken 不可一起使用
pageToken	integer	指示列表的起始位置，为空表未从 0 开始，与 pageOffset 不可一起使用，此值首次查询时不用填写，默认从 0 开始，后续填入返回的 nextPageToken 值

参数	类型	说明
pageSize	integer	返回结果一页的最大数量，与 pageToken 和 pageOffset 配合使用，如反回结果的总量超过 pageSize，在返回参数中会有 nextPageToken 字段用于指示下一页起始位置

返回参数：

参数	类型	说明
nextPageToken	integer	下一页的起始地址，放入下一次请求的 pageToken
totalFaces	integer	总人脸数
face[]	array	人脸列表数组，包含不同人脸的对象
{}	obj	数组中的一个人脸对象，包含 imageId, faceToken 两种信息，根据 pageSize 参数放入人脸个数
—imageId	string	上传成功的照片原图标识, 为 24 字节字符串，用于调取原图。
—faceToken	string	入库成功后，返回人脸的唯一标识，24 字节字符串，用于后面分组操作。如没有入库成功的照片该参数为空
—description	string	不超过 255 字节对照片文件的描述，UTF-8 编码

返回值：

状态码	说明	内容
200	成功	

响应数据示例：


```
"nextPageToken":1,
  "totalFaces":1995,
  "face":[
    {
      "imageId":"6NezwXmyWaB-6vvSxh7YhA==",
      "faceToken":"vGIGS4DEsCvFh_NaVwA8RQ==",
      "description":"龙石勇.jpg"
    }
  ]
}
```

错误码:

状态码	错误码	消息	说明
400	104001	ILLEGAL_ARGUMENT:<args>	参数格式错误，<args>返回错误参数名，逗号分隔
400	104002	MISSING_ARGUMENT:<args>	缺少必要参数，<args>返回缺少参数名，逗号分隔

5.1.3 删除某个人脸 DELETE/faces/faceToken/{faceToken}

操作类型:

DELETE

功能:

删除某一张人脸照片，将会删除 faceToken 对应的人脸图与分组相关绑定关系。URI 中 {faceToken} 填入具体人脸的 faceToken 24 字节字符串值。但对应的 ImageID、原图与描述仍然存在。

接口 URI:

/faces/{faceToken}

请求参数:

无

返回参数:

无

错误码:

状态码	错误码	消息	说明
400	104202	FACE_TOKEN_NOT_EXIST	人脸不存在

5.1.4 删除某个原始图片 DELETE/faces/image/{imageId}

操作类型:

DELETE

功能:

删除某一入库传入的原始照片，将会删除 imageId 对应的原始图片、人脸图与分组相关绑定关系。URI 中 {imageId} 填入具体人脸的 imageId 24 字节字符串值。所以需要特别注意，如删除原图，这个图相关的所有关系将也会被删除，包括：imageId 与 faceToken 将被删除，faceToken 与 groupName 之间的绑定关系也将被删除。

接口 URI:

/faces/image/{ imageId }

请求参数:

无

返回参数:

无

错误码:

状态码	错误码	消息	说明
400	104203	IMAGE_ID_NOT_EXIST	图片不存在

FacePass 接口: 无

操作:

- 1、 删除 ImageID 对应的原图与描述;
- 2、 删除 ImageID;
- 3、 删除 FaceToken 对应的人脸图; 如没有对应 FaceToken 到上一步结束。

4、 删除 FaceToken 与 FaceGroup 的所有关联关系；

5、 删除 FaceToken；

5.1.5 获取某人脸信息与扣图 GET/faces/faceToken/{faceToken}

操作类型：

GET

功能：

通过 faceToken 获取指定的人脸扣图 imageId 与人脸框位置信息。

接口 URI：

/faces/{faceToken}

请求参数：

无

返回参数：

参数	类型	说明
imageId	string	上传的照片原图标识, 为 24 字节字符串，上传操作即会返回该值。用于调取原图。
description	string	不超过 255 字节对照片文件的描述，UTF-8 编码
rect {}	obj	人脸在照片上的位置框，用于人脸框显示与入库的质量判断，检测到人脸即会返回改值。

参数	类型	说明
—left	integer	脸部框左边界坐标
—top	integer	脸部框上边界坐标
—right	integer	脸部框右边界坐标
—bottom	integer	脸部框下边界坐标
faceGroupList[]	array	人脸组列表数组，如没分组将为空。
—groupName	string	faecGroup 名称，数组中重复出现
image	string	BASE64 后的人脸图片

返回值：

状态码	说明	内容
200	成功	

响应数据示例：

```

"imageId": "M7LG-e8kOW8QKNXTvp4QDg==", =
  "description": "龙宇.jpg",
  "rect": {
    "left": 102,
    "top": 199,
    "right": 309,
    "bottom": 412
  }

```

```
},
"faceGroupList":[
  {
    "groupName":"PMPM"
  }
],
"image":"*****"
```

错误码:

状态码	错误码	消息	说明
400	104202	FACE_TOKEN_NOT_EXIST	人脸不存在

5.1.6 下载某原始照片 GET/faces/image/{imageId}

操作类型:

GET

功能:

通过 imageId 获取指定原始上传的照片文件流。

接口 URI:

/faces/image/{ imageId }

请求参数:

无

返回参数:

File Stream

错误码:

状态码	错误码	消息	说明
400	104203	IMAGE_ID_NOT_EXIST	图片不存在

5.1.7 初始化人脸库 DELETE /faces/ facesLib

操作类型:

DELETE

功能:

初始化人脸库，将删除所有入库的原始照片与人脸照片，并删除所有关联关系。需要谨慎使用。

删除的内容包括:

1. 原图库（白名单模式下）；
2. 人脸图库
3. 人脸与分组关联关系；

接口 URI:

/faces/facesLib

请求参数:

无

返回参数:

无

错误码:

无

5.1.8 批量人脸入库 POST / faces / batch

5.1.8.1 人脸批量入库 POST/faces/batch

操作类型:

POST

功能:

人脸批量入库操作，将一定数量的人脸照片一次性传入系统，再在系统内部进行后续的入库操作，如库完成后，可通过入库进度查询了解入库情况，流程属于异步操作。一次分割获取最多 50 个文件（亦可按照最大 size 来限制个数，注意：个数超过 50 个后端将报错）

接口 URI:

/faces / batch

请求参数(form-data)

参数	类型	说明	必选
total	integer	批量入库总个数	是
count	integer	单次传输图片个数，最大 50 个	是
data	string	存储图片信息的 json 数据，最大 2K	是
image	binary	图片二进制数据	是
data	string	存储图片信息的 json 数据，最大 2K	是
image	binary	图片二进制数据	是
data	string	存储图片信息的 json 数据，最大 2K	是
image	binary	图片二进制数据，其中 filename（默认）字段为图片名称	是
...			

注:data 中 json 串格式说明

{}			必选
—imageMd5	string	详见 wiki 说明，B3 MEGBOX 批量入库前端业务流程 此项与下面的 imageName 二选一	否
—imageName	string	人脸图片名称，此项与上面的 imageMd5 二选一	否

—batchNum	integer	批次号，同一 httppart 传输过来的数据相同，从 1 开始	是
—batchSeq	integer	批次内序号，同一 httppart 传输过来的图片递增，从 1 开始	否
—size	integer	图片数据原始长度，目前支持图片大小 4M 以下的图片	是
—description	string	不超过 255 字节对照片文件的描述，UTF-8 编码	否
— faceGroupList[]	array	人脸组列表数组，如没分组将为空	否
— —groupName	string	faecGroup 名称，数组中重复出现	否

例 1：下发 imageMd5：

```
{"imageMd5":"14hlGYOviONEvWwBRB9ijg==","batchNum":2,"batchSeq":13,"size":18427,"description":"","faceGroupList":["Group_test_ab_63","Group_test_ab_62"]}
```

例 2：下发 imageName：

```
{"imageName":"70SHG88uRyAccMXEBbdgJw=.jpg","batchNum":2,"batchSeq":13,"size":18427,"description":"","faceGroupList":["Group_test_ab_63","Group_test_ab_62"]}
```

返回值：

状态码	说明	内容
200	成功	

下发 imageMd5 的响应数据示例：

```
"data":{
  "totalCnt":4,
  "recvCnt":4,
```

```
    "successCnt":0,  
    "failCnt":0  
}
```

下发 imageName 的响应数据示例:

```
"data": {  
  "totalCnt": 2,  
  "recvCnt": 2,  
  "successCnt": 0,  
  "failCnt": 0,  
  "faceInfo": [  
    {  
      "imageName": "70SHG88uRyAccMXEBbdgJw==. jpg",  
      "imageMd5": "bgMca4THoQh1yKhv juiJGw=="  
    },  
    {  
      "imageName": "lADPDgQ9rUuB2lXNAbnNAWY_358_441. jpg",  
      "imageMd5": "s54iGAYVCe09-oS_yAC2Ig=="  
    }  
  ]  
}
```

5.1.8.2 人脸批量入库结束 POST/faces/faceBatchStop

请求参数

无

返回值：

状态码	说明	内容
200	成功	

5.1.8.3 获取人脸批量入库进度 GET/faces/faceBatchStatus

请求参数

无

返回值

参数	类型	说明	必选
totalCnt	integer	批量入库总个数	是
recvCnt	integer	实际接收图片个数	是
successCnt	integer	入库成功个数	是
failCnt	integer	入库失败个数	是

响应数据示例：

```
"data": {
  "totalCnt": 4,
  "recvCnt": 4,
  "successCnt": 0,
```

```
        "failCnt":0
    }
```

注：所有返回接口的错误码，详见错误码列表

注意事项：

1. POST /faces/batch 请求的时间间隔不能大于 30s

5.1.9 入库失败列表 API

5.1.9.1 获取入库失败列表 GET/faces/failList

操作类型：

GET

功能：

获取入库失败列表信息。

在进行 URI 解析操作时，要按照 pageToken——>pageSize 的固定顺序，不支持随意顺序参数。

接口 URI：

/faces/failList

请求参数：

参数	类型	说明
pageOffset	integer	列表结果偏移量，从 0 开始，与 pageToken 不可一起使用
pageToken	integer	指示列表的起始位置，为空表未从 0 开始，与 pageOffset 不可一起使用，此值首次查询时不用填写，默认从 0 开始，后续填入返回的 nextPageToken 值
pageSize	integer	返回结果一页的最大数量，与 pageToken 配合使用，如反回结果的总量超过 pageSize，在返回参数中会有 nextPageToken 字段用于指示下一页起始位置

返回参数：

参数	类型	说明
nextPageToken	integer	下一页的起始地址，放入下一次请求的 pageToken
totalFails	integer	现有总的失败数量
fails[]	array	入库失败列表
{}	obj	对像内放一条入库失败信息
—imageId	string	原图标识, 为 24 字节字符串
—imageName	string	人脸图片名称
—failReason	string	失败原因
—timeStamp	integer	Unix 时间戳，入库的时间，单位为秒

5.1.9.2 清空入库失败列表 DELETE/faces/failList

操作类型：

DELETE

功能:

清空失败列表

接口 URI:

/faces/failList

请求参数:

无

返回参数:

无

错误码:

无

6 FaceGroups 人脸分组操作 API

6.1.1 获取人脸组列表 GET/faceGroups

操作类型:

GET

功能：

获取当前设备中有的人脸分组列表。

接口 URI：

/faceGroups

请求参数：

无

返回参数：

参数	类型	说明
faceGroupList[]	array	人脸组列表数组，如没分组将为空。
—groupName	string	faecGroup 名称，数组中重复出现
—groupFaces	integer	分组中关联的人脸总数量

返回值：

状态码	说明	内容
200	成功	

响应数据示例：


```
"faceGroupList":[
  {
    "groupName":"PMPM2",
    "groupFaces":4
  },
  {
    "groupName":"PMPM",
    "groupFaces":1992
  }
]
```

6.1.2 创建人脸分组 POST/faceGroups

操作类型:

POST

功能:

创建人脸分组，用于 faceToken 的绑定分组。绑定与解绑只维护 faceToken 与 faceGroup 的映射关系。两个人脸组名称不可同名。最多一台设备上可创建 64 个分组。

接口 URI:

/faceGroups

请求参数:

参数	类型	说明
groupName	string	人脸分组名称，这将是分组的唯一标识，应满足以下规则： 1 支持 1-32 字节长度 1 支持英文大/小写字母、数字、半角下划线，支持中文及中文字符 1 建议正则/[^][a-zA-Z0-9_]{1,32}

返回参数：

无

返回值：

状态码	说明	内容
200	成功	

响应数据示例：

```

"faceGroupList":[
  {
    "groupName":"PMPM3",
    "groupFaces":0
  },
  {
    "groupName":"PMPM2",
    "groupFaces":4
  }
]

```

```
    },  
    {  
        "groupName": "PMPM",  
        "groupFaces": 1992  
    }  
]
```

错误码:

状态码	错误码	消息	说明
400	104300	GROUP_NAME_CONFLICT	分组名已存在
400	104302	GROUP_OVER_LIMIT	分组数超上限

6.1.3 删除某人脸分组 DELETE/faceGroups/{groupName}

操作类型:

DELETE

功能:

删除指定人脸分组，包括删除和清除两种操作，删除后与该分组绑定 faceToken。绑定与解绑只维护 faceToken 与 faceGroup 的映射关系。两个人脸组名称不可同名。
如删除分组，必须分组没有被使用，也就是没有布点在使用，如删除时该分组正在被使用，会删除失败，提示正在被使用。
如清除分组人员时，不影响布点。

接口 URI:

/faceGroups/{groupName}

请求参数：

参数	类型	说明	必填
deleteAllFace	integer	1：不删除组中的人脸； 2：删除该组中所有的人脸，同时解除组中人脸与其它组中的关联关系，默认值； 3：清除分组中全部人脸，如果只绑定当前分组，会将人共同删除，如果还绑定其他分组，将只解绑；	是

无返回参数：

错误码：

状态码	错误码	消息	说明
400	104300	GROUP_NAME_CONFLICT	分组名已存在或冲突
400	104304	GROUP_BEUSED	分组正被使用，不能删除

6.1.4 查询某分组绑定列表 GET/faceGroups/{groupName}

操作类型：

GET

功能:

通过 groupName 查询某人脸分组中绑定的人脸, 以 json 数组方式返回 faceToken 列表。

接口 URI:

/faceGroups/{groupName}

请求参数:

参数	类型	说明
groupName	string	faceGroup 名称, 非必需
pageOffset	integer	列表结果偏移量, 从 0 开始, 与 pageToken 不可一起使用
pageToken	integer	指示列表的起始位置, 为空表示从 0 开始, 与 pageOffset 不可一起使用
pageSize	integer	返回结果一页的最大数量, 与 pageToken, pageOffset 配合使用, 如返回结果的总量超过 pageSize, 在返回参数中会有 nextPageToken 字段用于指示下一页起始位置

返回参数:

参数	类型	说明
nextPageToken	integer	下一页的起始地址, 放入下一次请求的 pageToken
totalFaces	integer	此次返回的人脸数, 注: 不为该分组内的总人脸数。
faces[]	array	该分组绑定的人脸列表, 在此数组中列出, 如为空代表没有任何人脸被绑定

参数	类型	说明
—faceToken	string	人脸唯一 ID，24 字节字符串
—imageId	string	上传成功的照片原图标识，为 24 字节字符串，用于调取原图。
—description	string	不超过 255 字节对照片文件的描述，UTF-8 编码

响应数据示例：

```
"nextPageToken":1,
  "totalFaces":4,
  "face":[
    {
      "imageId":"VsWi_xqPzx04ruLVdEoimQ==",
      "faceToken":"K4ieJ-_sCnyVzRVCW_HxMQ==",
      "description":"小高. JPG"
    },
    {
      "imageId":"bZGm0IuWMyYD21B4jHRcrQ==",
      "faceToken":"tFzRWgcfAB7S152TjIZzdA==",
      "description":"小王. jpg"
    },
    {
      "imageId":"DY6ADfXU3IfIbr6JAVqP0g==",
      "faceToken":"L07IymTRv-bOMtBn7qsk0A==",
      "description":"小刘. JPG"
    }
  ]
```

错误码：

状态码	错误码	消息	说明
400	104301	GROUP_NAME_NOT_EXIST	分组不存在

6.1.5 绑定人脸到分组 POST / faceGroups / binding / {faceToken}

操作类型:

POST

功能:

将一个 faceToken 与一个 groupName 进行绑定。系统允许一个 faceToken 与多个 faceGroup 进行绑定。

接口 URI:

/faceGroups/binding/{faceToken}

请求参数:

参数	类型	说明	必选
faceGroupList[]	array	人脸组列表数组，如没分组将为空。	是

返回参数:

无

错误码:

状态码	错误码	消息	说明
400	104301	GROUP_NAME_NOT_EXIST	分组不存在
400	104202	FACE_TOKEN_NOT_EXIST	人脸不存在

6.1.6 解绑人脸与分组关系 DELETE / faceGroups / unbind / {faceToken}

操作类型:

DELETE

功能:

将一个 faceToken 与一个 groupName 进行解绑。但 faceToken 与其它 groupName 之间的绑定关系不影响。如过之前没有绑定关系，将会报错。

接口 URI:

/faceGroups/unbind/{faceToken}

请求参数:

参数	类型	说明	必填
faceGroupList[]	array	人脸组列表数组，如没分组将为空	是

返回参数:

无

错误码：

状态码	错误码	消息	说明
400	104303	GROUP_BINDING_NOT_EXIST	绑定关系不存在

7 Videos 视频流资源操作 API

■在设备中创建一个新的视频流布点，理论上建立好布点即开始正常工作状态。在新建一个布点之前，应完成如下工作：

- 1、 连接网络；
- 2、 配置设备时间、上传服务器等基础配置；
- 3、 配置报警/通行与抓拍接口；
- 4、 加载需要的模型与设置功能开关；
- 5、 添加人脸照片至人脸库并与相关分组绑定；
- 6、 新建布点。

■新建布点主要功能与流程：

- 1、 建立视频流连接；
- 2、 设置视频中人脸捕获时的过滤参数，如不满足质量要求的人脸将被过滤，不参与后面的识别等处理，由于质不高图片识别率也不高，以节省系统资源消耗；
- 3、 设置人脸活体阈值，当活体功能打开时有效，如在场景中不需要活体，可以关闭该功能，以节省系统资源消耗；

4、 绑定的人脸分组，当视频中的人脸被捕获到后，可以与分组中的人脸进行比对；

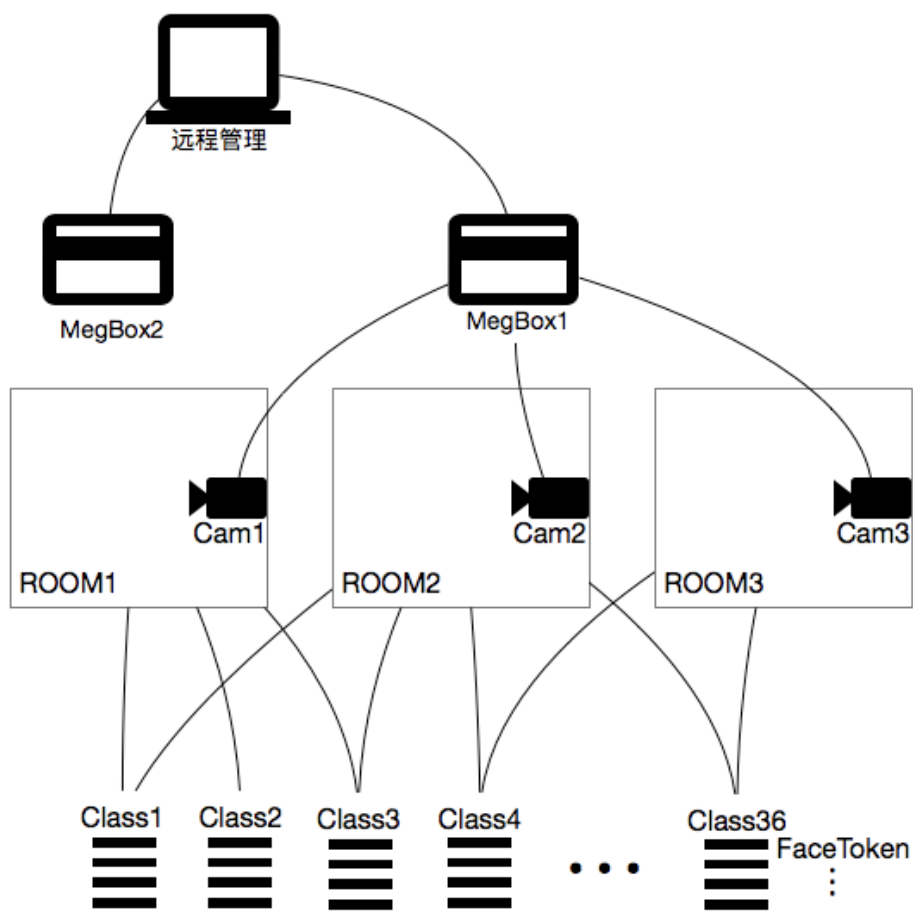
5、 设置人脸识别阈值，捕获的人脸通过与绑定分组中的有脸比对后，将得到一个相似度，如相似度大于识别阈值，将会触发报警/通行的后续动作；

相机视频流直连各方需求方，不由盒子转发。

■在教育点名场景应用：

物理场景与系统的对应关系：教室 (ROOM) → MegBox+摄像头，课表 → 时间 (分组有效期)&班级，班级 (Class) → groupName，班级中人员 → FaceToken；

图 7-1 班级人员与硬件部署关系图



首次更新方式：

1. 更新底库，可将全校学生进行同步，此操作会清除所有的分组与绑定关系，并撤销布控；
2. 建立班级分组，将所有在盒子覆盖的教室本年度会上课的班级分组进行同步，分组的 groupName 是唯一分组的 ID，代表着具体班级（如大二一班，GroupName= “23X299d1239E”，具体编码规则由业务系统指定，保证唯一即可）；

3. 建立分组与人员绑定关系，将分组里面绑定班级里的具体人员，需要业务系统控制，绑定的人员以 faceToken 与 GroupName 进行绑定；
4. 设定布控，根据课程表，查询当天在该 Room 上课的课程信息，其中包括班级与时间信息。在对应的时间里的布点的 Class→GroupName；

每日更新布控步骤：

1. 根据课程表，查询当天在该 Room 上课的课程信息，其中包括班级与时间信息。添加 Room→Cam+Box 的布点，摄像头布点对应的 Class→GroupName 组与 GroupName 的有效时间；
2. 如当天有调整，只需要更新该布控信息。

7.1.1 接入视频 POST/videos

操作类型：

POST

功能：

接入需要处理的视频流，以 RTSP 地址模式（不支持修改布控基本信息的修改，需要先调用 delete 接口，再使用 post 进行创建），目前支持 1080P h.264/h.265 (h.265 只识别，web 无法显示)

接口 URI：

/videos

请求参数：

参数	类型	说明
videoNo	integer	必填，视频通道编号：支持 16 路视频，值为 1~16，这将是此布点的唯一标识，后续操作都依赖这个名称
videoName	string	必填，摄像头布点名称，由用户起名，以下规则： • 支持 1-50 字节长度 • 支持英文大/小写字母、数字、半角下划线 • 建议正则 /^[a-zA-Z0-9_]{4,16}
url	string	必填，视频流地址, 通常为局域网内的 RTSP 地址，限制长度为 128 字节，超过则报错。
subUrl	string	选填，视频流子码流地址, 通常为局域网内的 RTSP 地址，限制长度为 128 字节，超过则报错。
username	string	IPC 接入的用户名，最长 31 字符
password	string	IPC 接入的密码，最长 63 字符
rotation	int	必填，图像旋转的角度，为配合不同的摄像头安装角度，目前只支持 4 个值：0，90.180，270。 分别代表顺时针旋转的角度 0°，90°，180°，270°，如错误将检测不到人脸。如输入不为以上 4 个值，将默认为 0° 来处理。

返回参数：

无

返回值：

状态码	说明	内容
200	成功	

响应数据示例:

```
"videoNo":6,
  "videoName":"pmpmppm",
    "url":"rtsp://192.168.1.24:554/1/1",
    "subUrl":"rtsp://192.168.1.24:554/1/1",
    "username":"admin",
    "password":"facepp123",
    "rotation":0,
```

错误码:

状态码	错误码	消息	说明
400	104000	ILLEGAL_JSON_BODY	json 结构解析错误
400	104001	ILLEGAL_ARGUMENT:<args>	参数格式错误，<args>返回错误参数名，逗号分隔
400	104002	MISSING_ARGUMENT:<args>	缺少必要参数，<args>返回缺少参数名，逗号分隔
400	104004	NO_ACCESS	无访问权限
400	104006	REQUEST_TIMEOUT	请求超时
400	104008	VIDEO_CH_OVER_LIMIT	超出限定视频路数
400	104010	VIDEO_CH_ACCESS_FAIL	视频流接入失败
400	104011	VIDEO_CH_EXIST	视频通道已存在

更多详细信息参见错误码列表

7.1.2 视频布控 PUT/ videos / {videoNo} / surveillances

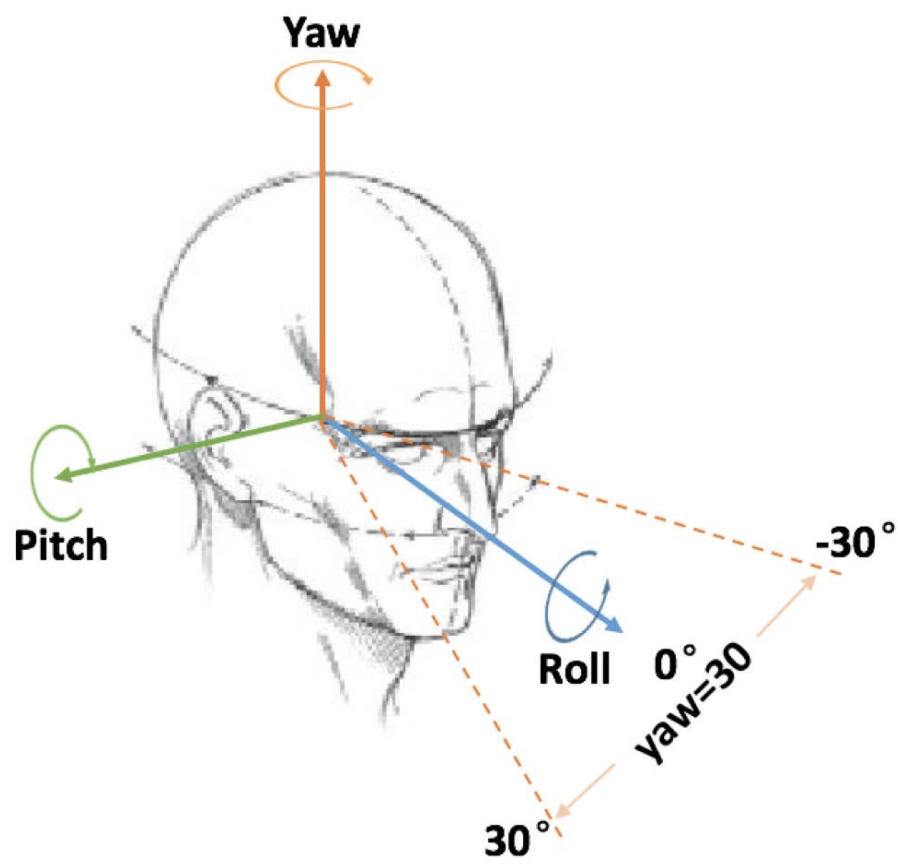
操作类型:

PUT

功能:

对应视频流编号, 设置关联报警底库、人脸抓拍的过滤条件、比对阈值, 并进行布控。

图 7-2 人脸角度示意图



接口 URI:

/serveillances

参数	类型	说明
surveillanceEnabled	bool	布控使能开关

filter{}	obj	过滤条件，条件越高，通过抓拍的人脸将越少，但人脸质量越高，识别的效果会越好。如条件越低，通过抓拍的人脸会越多，但人脸质量下线会降低，影响识别的效果。
—roll	integer	必填，人脸旋转角度限制（0~180），建议<=45°
—yaw	integer	必填，人脸水平角度限制（0~100），建议<=35°
—pitch	integer	必填，人脸垂直角度限制（0~100），建议<=45°
—blurness	integer	必填，图像清晰度限制（0~100），值越小越清晰，建议<=60
—highBrightness	integer	人脸最高亮度（50~100），建议 90
—lowBrightness	integer	人脸最低亮度（0~49），建议 20
—deviation	integer	人脸对比度（0~100），建议 50 值越高，代表允许人脸对比度越强，但影响识别效果。值越代，代表允许的对比越低，识别效果会提高，但抓拍率会降低。
—faceMin	integer	必填，人脸允许的长或宽的最小像素（20~380），值越小识别效果越差，建议>=80
group[]	array	如关联多个分组，在数组中添加多个分组对象，每个分组对象带一个分组唯一号与分组的有效时间。
{}	obj	设定关联的分组与有效时间，最多 20 组。
— groupName	string	必填，视频绑定的人脸分组名称，视频中抓拍的人脸将与这个 group 中的人脸进行识别。如高于比对阈值将引发报警或允许通行动作
— startTime	string	必填，该绑定分组有效的开始时间，格式：HHMMSS，例子：“080000”，为 24 小时制，默认为“000000”

— endTime	string	必填，该绑定分组失效的开始时间，格式：HHMMSS，例子：“184500”，为24小时制，默认为“000000”，如startTime与endTime两个值都为默认值“000000”，将24小时有效
— searchThresholdHigh	integer	必填，人脸识别的高限阈值，（0~100）之间的数，如最高相似度的超过阈值，可通过Alert接口得到超阈值的不同组里的比对相似度列表。建议70~80，具体值根据实际情况调整
— searchThresholdLow	integer	必填，人脸识别的低限阈值，（0~100）之间的数，如最高相似度低于低线阈值，代表这是一个陌生人。建议：20，可通过Alert接口得到低阈值的不同组里的比对相似度列表
globalStranger{}	obj	非必填，全局陌生人配置
—enable	bool	必填，配置开关 true 启用 false 禁用
— searchThresholdLow	integer	必填，人脸识别的低限阈值，（0~100）之间的数，如最高相似度低于低线阈值，代表这是一个陌生人。建议：20，可通过Alert接口得到低阈值的不同组里的比对相似度列表
—startTime	string	必填，该绑定分组有效的开始时间，格式：HHMMSS，例子：“080000”，为24小时制，默认为“000000”
—endTime	string	必填，该绑定分组失效的开始时间，格式：HHMMSS，例子：“184500”，为24小时制，默认为“000000”，如startTime与endTime两个值都为默认值“000000”，将24小时有效
livenessThreshold	integer	如打开活体判断功能必填，活体判断阈值，（0~100）之间的数，建议60，当活体开关关闭时，该值非必须，如设置为0也相当于关闭，但会消耗系统资源。如需要关闭该功能可到configs公共参数设置接口进行设置，但所有视频都将不做活体检测。
bestStrangerUpload	bool	非必填，是否上报全局最优陌生人，默认为false

返回参数:

无

错误码:

状态码	错误码	消息	说明
400	104000	ILLEGAL_JSON_BODY	json 结构解析错误
400	104001	ILLEGAL_ARGUMENT:<args>	参数格式错误, <args>返回错误参数名, 逗号分隔
400	104002	MISSING_ARGUMENT:<args>	缺少必要参数, <args>返回缺少参数名, 逗号分隔
400	104006	REQUEST_TIMEOUT	请求超时
400	104009	NO_VIDEO	无指定视频通道, 未创建或通道与名称不匹配

7.1.3 获取所有视频布点信息 GET/videos

操作类型:

GET

功能:

从设备中获取当前布点的视频流设置列表(只在纯人脸模式支持)

接口 URI:

/videos

请求参数:

无

返回参数:

参数	类型	说明
videos[]	array	放置视频布点信息的数组，内部可能会有多组视频信息。
{}	obj	一组布控信息在此对象中，如果有多组布控将有多个对象
—videoNo	integer	视频通道编号：支持 16 路视频，值为 1~16
—videoName	string	摄像头布点名称，由用户起名，这将是此布点的唯一标识，后续操作都依赖这个名称。以下规则： 1 支持 1-50 字节长度 1 支持英文大/小写字母、数字、半角下划线 建议正则 /^[a-zA-Z0-9_]{4,16}
—url	string	主码流，通常为局域网内的 RTSP 地址。
—subUrl	string	子码流，通常为局域网内的 RTSP 地址。
—username	string	IPC 接入的用户名，最长 31 字符
—password	string	IPC 接入的密码，最长 63 字符

参数	类型	说明
—rotation	int	必填，图像旋转的角度，为配合不同的摄像头安装角度，目前只支持 4 个值：0，90.180，270。分别代表顺时针旋转的角度 0°，90°，180°，270°，如错误将检测不到人脸。如输入不为以上 4 个值，将默认为 0° 来处理。
—livenessThreshold	float	活体判断阈值，（0~100）之间的浮点数，当活体开关关闭时，该值非必须，如设置为 0 也相当于关闭，但会消耗系统资源。如需要关闭该功能可到 configs 公共参数设置接口进行设置，但所有视频都将不做活体检测。
—surveillanceEnabled	bool	布控使能开关
—videoConStatus	int	视频流连接状态 0:连接成功 1:连接失败 2:认证失败 3:视频流格式错误 4:超出通道最大分辨率 5:解码失败 6:超出最大解码能力
—filter{}	obj	过滤条件，条件越高，通过抓拍的人脸将越少，但人脸质量越高，识别的效果会越高。如条件越低，通过抓拍的人脸会越多，但人脸质量下线会降低，影响识别的效果。
— —roll	integer	人脸旋转角度限制（0~180），建议<=45°
— —yaw	integer	人脸水平角度限制（0~100），建议<=35°
— —pitch	integer	人脸垂直角度限制（0~100），建议<=45°
— —blurness	float	图像清晰度限制（0~100），建议<=60
— —highBrightness	integer	人脸最高亮度（50~100），建议 90
— —lowBrightness	integer	人脸最低亮度（0~49），建议 20
— —deviation	integer	人脸对比度（0~100），建议 50

参数	类型	说明
		值越高，代表允许人脸对比度越强，但影响识别效果。值越低，代表允许的对比度越低，识别效果会提高，但抓拍率会降低。
— —faceMin	integer	人脸允许的长或宽的最小像素（20~380），值越小识别效果越差，建议>=80
— —group[]	array	如关联多个分组，在数组中添加多个项
{}	obj	设定关联的分组与有效时间，最多 20 组。
— —groupName	string	必填，视频绑定的人脸分组名称，视频中抓拍的人脸将与这个 group 中的人脸进行识别。如高于比对阈值将引发报警或允许通行动作
— — startTime	string	必填，该绑定分组有效的开始时间，格式：HHMMSS，例子：“080000”，为 24 小时制，默认为“000000”
— — endTime	string	必填，该绑定分组失效的开始时间，格式：HHMMSS，例子：“184500”，为 24 小时制，默认为“000000”，如 startTime 与 endTime 两个值都为默认值“000000”，将 24 小时有效，不支持跨天
— —searchThresholdHigh	float	必填，人脸识别的高限阈值，（0~100）之间的浮点数，如最高相似度的超过阈值，可通过 Alert 接口得到超阈值的不同组里的比对相似度列表。建议 70~80，具体值根据实际情况调整
— —searchThresholdLow	float	必填，人脸识别的低限阈值，（0~100）之间的浮点数，如最高相似度低于低线阈值，代表这是一个陌生人。可通过 Alert 接口得到低阈值的不同组里的比对相似度列表
— —globalStranger {}	obj	非必填，全局陌生人配置
— —enable	bool	必填，配置开关 true 启用 false 禁用

参数	类型	说明
— — searchThresholdLow	integer	必填，人脸识别的低限阈值，（0~100）之间的数，如最高相似度低于低线阈值，代表这是一个陌生人。建议：20，可通过 Alert 接口得到低阈值的不同组里的比对相似度列表
— —startTime	string	必填，该绑定分组有效的开始时间，格式：HHMMSS，例子：“080000”，为24小时制，默认为“000000”
— —endTime	string	必填，该绑定分组失效的开始时间，格式：HHMMSS，例子：“184500”，为24小时制，默认为“000000”，如 startTime 与 endTime 两个值都为默认值“000000”，将24小时有效

响应数据示例：

```
"videos":[
  {
    "videoNo":1,
    "videoName":"safsf",
    "url":"rtsp://192.168.1.22:554/1/1",
    "subUrl":"rtsp://",
    "username":"admin",
    "password":"admin123",
    "rotation":0,
    "livenessThreshold":50,
    "surveillanceEnabled":false,
    "filter":{
      "roll":45,
      "yaw":35,
      "pitch":45,
      "blurness":80,
```

```
        "highBrightness":90,
        "lowBrightness":20,
        "deviation":50,
        "faceMin":50
    },
    "group":[
        {
            "groupName":"sfdsaf",
            "startTime":"000000",
            "endTime":"000000",
            "searchThresholdHigh":60,
            "searchThresholdLow":0
        }
    ]
}
```

7.1.4 删除某视频布点 DELETE/videos/{videoNo}

操作类型:

DELETE

功能:

删除设备中某个视频流布点，同时删除所有配置与绑定关系。另与该布点相关的历史抓拍、报警信息也将全部删除。

接口 URI:

/videos/{videoNo}

请求参数:

参数	类型	说明
videoNo	integer	要删除的摄像头通道号

返回参数:

无

7.1.5 视频布控获取背景图 GET/ videos / {videoNo} / snapFullImage

操作类型:

GET

功能:

视频布控通道重置

接口 URI:

/snapFullImage

请求参数:

无

返回参数:

参数	类型	说明
image	string	BASE64 后的人脸图片

错误码：

状态码	错误码	消息	说明
400	104006	REQUEST_TIMEOUT	请求超时
400	104009	NO_VIDEO	无指定视频通道，未创建或通道与名称不匹配

7.1.6 视频布控重置 GET/ videos / {videoNo} / trackReset

操作类型：

GET

功能：

视频布控通道重置

接口 URI：

/trackReset

请求参数：

无

返回参数：

无

错误码：

状态码	错误码	消息	说明
400	104006	REQUEST_TIMEOUT	请求超时
400	104009	NO_VIDEO	无指定视频通道，未创建或通道与名称不匹配

7.1.7 编辑某视频布点 PUT/videos/{videoNo}

操作类型：

PUT

功能：

修改当前视频布点的相关信息。

接口 URI：

/videos

请求参数：

参数	类型	说明
videoName	string	必填，摄像头布点名称，由用户起名，以下规则： 支持 1-50 字节长度

参数	类型	说明
		- 支持英文大/小写字母、数字、半角下划线 - 建议正则/[^][a-zA-Z0-9_]{4,16}
url	string	必填，视频流地址, 通常为局域网内的 RTSP 地址，限制长度为 128 字节，超过则报错。
subUrl	string	选填，视频流子码流地址, 通常为局域网内的 RTSP 地址，限制长度为 128 字节，超过则报错。
username	string	IPC 接入的用户名，最长 31 字符
password	string	IPC 接入的密码，最长 63 字符
rotation	int	必填，图像旋转的角度，为配合不同的摄像头安装角度，目前只支持 4 个值：0，90.180，270。分别代表顺时针旋转的角度 0° ， 90° ， 180° ， 270° ， 如错误将检测不到人脸。如输入不为以上 4 个值，将默认为 0° 来处理。

返回参数：

无

返回值：

状态码	说明	内容
200	成功	

错误码：

状态码	错误码	消息	说明
400	104000	ILLEGAL_JSON_BODY	json 结构解析错误
400	104001	ILLEGAL_ARGUMENT:<args>	参数格式错误，<args>返回错误参数名，逗号分隔
400	104002	MISSING_ARGUMENT:<args>	缺少必要参数，<args>返回缺少参数名，逗号分隔
400	104004	NO_ACCESS	无访问权限
400	104006	REQUEST_TIMEOUT	请求超时
400	104008	VIDEO_CH_OVER_LIMIT	超出限定视频路数
400	104010	VIDEO_CH_ACCESS_FAIL	视频流接入失败
400	104017	VIDEO_CH_NOT_EXIST	视频通道不存在

更多详细信息参见错误码列表

8 Alerts 报警/通行资源操作 API

本接口提供报警/通行消息，只有在前端设备捕获的人脸与关联分组中的人脸相似度超过设定的阈值后才会产生该消息。本接口提供两种工作模式：

- 1 主动上报模式，通过用户设置的域名或 IP 地址与端口，使用 HTTP GET 进行主动上报，如上报成功，将会被删除。如上报失败，将会重复上报，直至消息队列为空，除非超出设置的 BufferSize 后会被自动删除。如此方式实时性高，但需要用户建立一个 Web Server 服务来接收消息，并返回确认信息。主动上报数据说明在其它《参数说明》章节中进行说明

1 被动轮询模式，设备将消息放在消息队列中，待外部调取，消息调取成功后任然保留，只有通过删除报警/通行记录来清空。

注意：最多保留 10000 条的报警信息，如超出将会自动删除。

人脸抓取与报警逻辑

1 黑名单模式	
停留时长	处理方法
≤10s	人脸进入视频内到离开，选择最优的一张，进行比对，并上报抓拍数据，包括进入、捕获、离开时间。
>10s	停留超过 10s 时，将 10s 内获取到的最优人脸进行上报，并比对，上报时离开时间为空。之后每隔一分钟上报一次，但不比对，直至最后离开。期间抓拍照片中的最优照片，每次更新都将更新抓拍的捕获时间，进入时间始终不变，如没有离开，离开为空，如离开就为离开时间。其中 trackID 保持不变。
1 白名单模式	
以最快速度进行人脸捕获，捕获到符合过滤条件的人脸即送去比对。同时仍然进行捕获，如捕获到质量比前一张好的，再送至比对，如此反复，直到超过尝试次数或已得到比对结果。	

图 9-1 Alerts 与 Clips 接口关系



8.1.1 获取报警/通行事件列表 GET/alerts

操作类型:

GET

功能:

从设备中获取 alert 全部消息队列，主要包括：视频流布点名称或照片流名称、抓拍人脸 ID、人脸分组名、入库人脸唯一 ID（faceToken）、时间等信息。

在进行 URI 解析操作时，要按照 pageToken——>pageSize 的固定顺序，不支持随意顺序参数。

接口 URI:

/alerts

请求参数:

参数	类型	说明
pageOffset	integer	列表结果偏移量，从 0 开始，与 pageToken 不可一起使用
pageToken	integer	指示列表的起始位置，为空表未从 0 开始，与 pageOffset 不可一起使用，此值首次查询时不用填写，默认从 0 开始，后续填入返回的 nextPageToken 值
pageSize	integer	返回结果一页的最大数量，与 pageToken 配合使用，如返回结果的总量超过 pageSize，在返回参数中会有 nextPageToken 字段用于指示下一页起始位置
QueryType	integer	查询类型。0：查询全部；1：查询底库人员；2：查询非底库人员（陌生人）。（非必选）
Dedup	integer	去重标志。0：不去重；1：去重。（非必选）
chnNo	integer	通道号 混合通道模式（1~8，17~48）全结构化模式（1~16）（非必选）
timeStart	integer	开始时间 Unix 时间戳 单位为秒（非必选）
timeEnd	integer	结束时间 Unix 时间戳 单位为秒（非必选）
faceToken	string	指定人脸进行查询。（非必选）

返回参数:

参数	类型	说明
nextPageToken	integer	下一页的起始地址，放入下一次请求的 pageToken
totalAlerts	integer	现有总的报警数量
alertsEvent[]	array	alert 消息列表，按时间排序
{}	obj	对像内放一条消息，如有多条消息将重复
—channelName	string	摄像头布点或抓拍机布点名称，就是 videoName 或 pictureName，根据工作模式 workMode 设置所决定
—trackId	int	视频中的一个人脸或一张抓拍机图片唯一编号，用于调用 captures 接口调取抓拍图片与 clips 接口调取剪辑录像使用，64 位 int，最大可达到 20 位整数。
—livenessScore	float	抓拍人脸活体判断分数
—captureFullImageURL	string	全景图 URL
—timestamp	integer	Unix 时间戳，事件发生的时间，单位为毫秒
—captureImageURL	string	视屏捕获到的人脸图片，对应一 trackId, 以 jpg 文件格式经过 BASE64 后的字节码
—videoNo	integer	通道号
—top1 {}	obj	比对结果 TOP1
— —searchScore	float	抓拍人脸与底库中人脸比对的相似度
— —alertGroup[]	array	报警人脸分组列表
— — —groupName	string	人脸分组名
— —group[]	array	人脸分组列表

参数	类型	说明
— —groupName	string	人脸分组名
— —faceToken	string	人脸的唯一 ID，用于获取分组中人脸照片
— —imageId	string	原始图片唯一 ID，用于获取底库中原始照片
— —description	string	人脸描述，UTF-8 编码
— —faceImage	string	底库中抽取的人脸抠图图片，对应一个 faceToken，以 jpg 文件格式经过 BASE64 后的字节码
—top2 {}	obj	比对结果 TOP2
—top3 {}	obj	比对结果 TOP3
—top4 {}	obj	比对结果 TOP4
—top5 {}	obj	比对结果 TOP5
—faceAttr {}	obj	人脸属性
— —age	integer	年龄
— —gender	integer	性别 0：男 1：女
— —hair	integer	0：光头 1：少量头发（包含秃顶）2：短发 3：长发 4：未知
— —beard	integer	0：无胡子或者胡子不明显 1：嘴唇上面的胡子 2：络腮胡 3：未知
— —hat	integer	0：无帽子 1：安全帽 2：厨师帽 3：学生帽 4：头盔 5：01 小白帽 6：头巾 7：其他帽子 8：未知
— —respirator	integer	0：无口罩 1：医用口罩 2：雾霾口罩 3：普通口罩 4：厨房用“透明口罩” 5：未知

参数	类型	说明
— —glasses	integer	0：无眼镜 1：深色框透明眼镜 2：普通透明眼镜 3：墨镜 4：未知
—stranger	integer	陌生人。0：非陌生人；1：陌生人（非必选）
—frequency	integer	目标出现频次（非必选）

响应数据示例：

```

    "nextPageToken":1,
    "totalAlerts":93,
    "alertsEvent":[
        {
            "channelName":"PMtest01",
            "trackId":623,
            "livenessScore":0,
            "captureImageURL":"./record_CHN0/2019Y10M10D15H/A05M02S826_623_1_c0.00_r-
8.82_p1.50_y32.33_b0.02_w0_80.64.jpg",
            "captureFullImageURL":"./record_CHN0/2019Y10M10D15H/A05M02S826_623_0_661_243_755_350_80.64.jpg",
            "timestamp":1570691102826,
            "top1":{
                "searchScore":80.64,
                "faceToken":"Dy9hOK9vKELki08CBeo4Uw==",
                "imageId":"a0nb1s5qP0AU6RRQDs-BoA==",
                "description":"刘晔.jpg",
                "faceImageURL":"./group/face_data/loe327/1te462/Dy9hOK9vKELki08CBeo4Uw==.jpg",
                "group":[
                    {
                        "groupName":"PMPM"
                    }
                ]
            }
        }
    ]

```

```
    }
  ],
  "alertGroup":[
    {
      "groupName":"PMPM"
    }
  ]
},
"top2":{

  *****
},
"top3":{
  *****
},
"top4":{
  *****
},
"top5":{
  *****
},
"faceAttr":{
  "age":31,
  "gender":0,
  "hair":0,
  "beard":1,
  "hat":1,
  "respirator":0,
  "glasses":2
```

}

}

错误码:

状态码	错误码	消息	说明
400	104007	RESOURCES_NOT_FOUND	访问的资源不存在

8.1.2 删除某报警/通行记录 DELETE/alerts/ {trackId} / {timestamp}

操作类型:

DELETE

功能:

删除某一路 alerts 报警记录，如报警记录超过布点时设定的条数，将会被自动删除。

接口 URI:

/alerts/ {trackId} / {timestamp}

请求参数:

无

返回参数:

无

9 Captures 抓拍人脸资源操作 API

抓拍接口是获取从视频流与照片流抓拍到的人脸照片的接口，前端设备所获取到的人脸照片会产生唯一的 trackId，通过 trackId 可以关联到具体的图片。如果在公共参数中关闭了抓拍照片功能（capturePicEnabled），此功能将失效，可不必设置，调用也会无效。主要包括两种模式：

1 主动上报模式，通过用户设置的域名或 IP 地址与端口，使用 HTTP GET 进行主动上报，如上报成功，将会被删除。如上报失败，将会重复上报，直至消息队列为空，除非超出设置的 BufferSize 后会被自动删除。此方式实时性高，但需要用户建立一个 Web Server 服务来接收消息，并返回确认信息。

1 被动轮询模式，设备将消息放在消息队列中，待外部调取，消息调取成功后任然保留，只有通过删除记录来清空。

通过查询接口，可以获取到抓拍的时间、布点名称、照片的唯一 ID:trackId，依据 trackId 可以获取到抓拍图片。

注意：只保留不超过 50000 条的信息，且使用存储容量不大于总容量 90%，超出超出 50000 条或使用存储容量大于总容量 90%情况下，会按时间倒序，自动删除旧的抓拍信息与图片，同时清除对应的列表信息。

图 10-1 人脸抓拍/捕获接口关系



Captures 人脸捕获

主动上报
上报消息最大100条

channelName 布点名称
trackId 人脸跟踪ID
enterTimestamp 进入时间
catchTimestamp 捕获时间
leaveTimestamp 离开时间

被动查询

channelName 布点名称
trackId 人脸跟踪ID
catchTimestamp 捕获时间

获取抓拍

channelName 布点名称
trackId 人脸跟踪ID
enterTimestamp 进入时间
catchTimestamp 捕获时间
leaveTimestamp 离开时间
facelImage 抓拍人脸图

删除报警 自动删除：if 存储照片>限定张
手动删除：Track ID



9.1.1 获取全量抓拍事件列表 GET/captures

操作类型:

GET

功能:

获取抓拍队列中的全部抓拍事件信息。

在进行 URI 解析操作时，要按照 pageToken——>pageSize 的固定顺序，不支持随意顺序参数。

接口 URI:

/captures

请求参数:

参数	类型	说明
pageOffset	integer	列表结果偏移量，从 0 开始，与 pageToken 不可一起使用
pageToken	integer	指示列表的起始位置，为空表未从 0 开始，与 pageOffset 不可一起使用，此值首次查询时不用填写，默认从 0 开始，后续填入返回的 nextPageToken 值
pageSize	integer	返回结果一页的最大数量，与 pageToken 配合使用，如反回结果的总量超过 pageSize，在返回参数中会有 nextPageToken 字段用于指示下一页起始位置
chnNo	integer	通道号 混合通道模式（1~48）全结构化模式（1~16）（非必选）

参数	类型	说明
timeStart	integer	开始时间 Unix 时间戳 单位为秒（非必选）
timeEnd	integer	结束时间 Unix 时间戳 单位为秒（非必选）

返回参数：

参数	类型	说明
nextPageToken	integer	下一页的起始地址，放入下一次请求的 pageToken
totalCaptures	integer	现有总的抓拍数量
captures[]	array	抓拍事件消息列表，按时间排序
{}	obj	对像内放一条消息，如有多条消息将重复
—channelName	string	摄像头布点或抓拍机布点名称，就是 videoName 或 pictureName，根据工作模式 workMode 设置所决定
—trackId	integer	视频中的一个人脸或一张抓拍机图片唯一编号，用于调用 captures 接口调取抓拍图片与 clips 接口调取剪辑录像使用
—faceImageURI	string	捕获到的人脸图片资源 URI，对应一 trackId, 从 URI 获取图片
—enterTimestamp	integer	Unix 时间戳，人脸第一次捕获到的时间，单位为毫秒
—catchTimestamp	integer	Unix 时间戳，人脸捕获最佳照片的时间，单位为毫秒
—leaveTimestamp	integer	Unix 时间戳，人脸离开，最后一次捕获到的时间，单位为毫秒，如后面还有获取照片，将为空

参数	类型	说明
—faceAttr{}	obj	人脸属性
— —age	integer	年龄
— —gender	integer	性别 0：男 1：女
— —hair	integer	0：光头 1：少量头发（包含秃顶） 2：短发 3：长发 4：未知
— —beard	integer	0：无胡子或者胡子不明显 1：嘴唇上面的胡子 2：络腮胡 3：未知
— —hat	integer	0：无帽子 1：安全帽 2：厨师帽 3：学生帽 4：头盔 5：01 白帽 6：头巾 7：其他帽子 8：未知
— —respirator	integer	0：无口罩 1：医用口罩 2：雾霾口罩 3：普通口罩 4：厨房用“透明口罩” 5：未知
— —glasses	integer	0：无眼镜 1：深色框透明眼镜 2：普通透明眼镜 3：墨镜 4：未知

返回参数

无

响应数据示例：

```

"nextPageToken":1,
  "totalCaptures":3833,
  "captures":[
    {
      "channelName":"PMtest01",
      "trackId":684,
      "faceImageURI":"./record_CHN0/2019Y10M10D15H/N13M12S918_684_1_c0.00_r-1.97_p-2.14_y-0.14_b0.21_w0_49.50.jpg",

```

```
    "fullImageURI": ". /record_CHN0/2019Y10M10D15H/N13M12S918_684_0_227_422_305_505_49.50.jpg",
    "enterTimestamp": 1570691591343,
    "catchTimestamp": 1570691592918,
    "leaveTimestamp": 1570691593620,
    "faceAttr": {
      "age": 50,
      "gender": 1,
      "hair": 1,
      "beard": 0,
      "hat": 1,
      "respirator": 1,
      "glasses": 3
    }
  },
```

错误码:

状态码	错误码	消息	说明
400	104400	CAPTURE_DISABLE	未使能功能

9.1.2 获取某抓拍信息与图片 GET/captures/{trackId}/{timestamp}

操作类型:

GET

功能:

通过 trackId 获取抓拍队列中的指定抓拍信息与图片。

接口 URI:

/captures/{trackId}/{catchTimestamp}

请求参数:

无

返回参数:

参数	类型	说明
channelName	string	摄像头布点名称，就是 videoName，根据工作模式 workMode 设置所决定
trackId	integer	视频中的一个人脸图片唯一编号，用于调用 captures 接口调取抓拍图片使用
{}	obj	获取到一次的图片与信息，如人员停留时间较长，会有多张照片产生。
—faceImageURI	string	捕获到的人脸图片资源 URI，对应一 trackId, 从 URI 获取图片
—enterTimestamp	integer	Unix 时间戳，人脸第一次捕获到的时间，单位为毫秒
—catchTimestamp	integer	Unix 时间戳，人脸捕获最佳照片的时间，单位为毫秒
—leaveTimestamp	integer	Unix 时间戳，人脸离开，最后一次捕获到的时间，单位为毫秒，如后面还有获取照片，将为空
—faceAttr {}	obj	人脸属性
— —age	integer	年龄
— —gender	integer	性别 0：男 1：女

参数	类型	说明
— —hair	integer	0: 光头 1: 少量头发（包含秃顶） 2: 短发 3: 长发 4: 未知
— —beard	integer	0: 无胡子或者胡子不明显 1: 嘴唇上面的胡子 2: 络腮胡 3: 未知
— —hat	integer	0: 无帽子 1: 安全帽 2: 厨师帽 3: 学生帽 4: 头盔 5: 01 小白帽 6: 头巾 7: 其他帽子 8: 未知
— —respirator	integer	0: 无口罩 1: 医用口罩 2: 雾霾口罩 3: 普通口罩 4: 厨房用“透明口罩” 5: 未知
— —glasses	integer	0: 无眼镜 1: 深色框透明眼镜 2: 普通透明眼镜 3: 墨镜 4: 未知

错误码:

状态码	错误码	消息	说明
400	104400	CAPTURE_DISABLE	未使能功能
400	104007	RESOURCES_NOT_FOUND	访问的资源不存在

9.1.3 删除某抓拍信息与图片 DELETE/captures/{trackId} / {timestamp}

操作类型:

DELETE

功能:

删除指定的抓拍记录与图片。

接口 URI:

/captures/{trackId}/ {catchTimestamp}

请求参数:

无

返回参数:

无

错误码:

状态码	错误码	消息	说明
400	104400	CAPTURE_DISABLE	未使能功能
400	104007	RESOURCES_NOT_FOUND	访问的资源不存在

10 人脸比对 API

10.1.1 1: 1 两张人脸相似度比对 POST/compare

操作类型:

POST

功能:

将两张上传的图片进行比对, 得到一个相似度值(0~100)之间, 越高表示为同一个人的可能性越高。

照片限制:

- 1 图片类型: JPG、JPEG、PNG、BMP
- 1 图片大小 $\leq 4\text{MB}$
- 1 图片最大尺寸 $\leq 4096*4096$
- 1 图片最小尺寸 $\geq 100*100$
- 1 每张图片中只能有一张人脸, 如有多张脸取最大

接口 URI:

/compare

请求参数

参数	类型	说明
image1	binary	人脸图片文件 1
image2	binary	人脸图片文件 2
livenessEnabled	bool	活体使能开关

返回参数

参数	类型	说明
compareScore	float	1:1 相似度（0~100）之间
detectionResult1{}	obj	人脸图片 1 检测结果
—left	integer	脸部框左边界坐标
—top	integer	脸部框上边界坐标
—right	integer	脸部框右边界坐标
—bottom	integer	脸部框下边界坐标
—roll	integer	人脸旋转角度
—yaw	integer	人脸水平角度
—pitch	integer	人脸垂直角度
—blurness	float	图像清晰度（0~100）值越小越清晰
—livenessScore	float	活体检测结果（0~100）
detectionResult2{}	obj	人脸图片 2 检测结果
—left	integer	脸部框左边界坐标
—top	integer	脸部框上边界坐标
—right	integer	脸部框右边界坐标
—bottom	integer	脸部框下边界坐标
—roll	integer	人脸旋转角度
—yaw	integer	人脸水平角度

参数	类型	说明
—pitch	integer	人脸垂直角度
—blurness	float	图像清晰度（0~100）值越小越清晰
—livenessScore	float	活体检测结果（0~100）

响应数据示例：

```

"compareScore":22.55,
  "detectionResult1":{
    "left":126,
    "top":151,
    "right":306,
    "bottom":337,
    "roll":-1,
    "yaw":0,
    "pitch":-4,
    "blurness":0,
    "livenessScore":0
  },
  "detectionResult2":{
    "left":195,
    "top":194,
    "right":386,
    "bottom":382,
    "roll":1,
    "yaw":3,
    "pitch":8,
    "blurness":1,

```

```
}    "livenessScore":0
```

错误码:

状态码	错误码	消息	说明
400	104202	FACE_TOKEN_NOT_EXIST	人脸不存在
400	104201	BAD_QUALITY:<reason>	人脸未通过质量判断，reason 为未通过的检测项目

10.1.2 1: N 一张图与人脸分组搜索 POST/search

操作类型:

POST

功能:

将上传的一张图片与人脸分组中的人脸进行比对，找出得分最高的 5 个人脸，并返回得分，越高表示为同一个人的可能性越高。

照片限制:

1 图片类型: JPG、JPEG、PNG、BMP

1 图片大小 <= 4MB

1 图片最大尺寸 <= 4096*4096

1 图片最小尺寸 >= 100*100

1 每张图片中只能有一张人脸，如有多张脸取最大

接口 URI:

/search

请求参数 (form) :

参数	类型	说明	必选
image	binary	人脸图片	是
group []	array	人脸分组数组列表	是
—groupName	string	人脸分组名	是
livenessEnabled	bool	活体使能开关，默认关	否
featureUpload	bool	返回 feature 数据开关，默认关	否

返回参数:

参数	类型	说明
detectionResult {}	float	人脸图片 1 检测结果
—left	integer	脸部框左边界坐标
—top	integer	脸部框上边界坐标
—right	integer	脸部框右边界坐标
—bottom	integer	脸部框下边界坐标

参数	类型	说明
—roll	integer	人脸旋转角度
—yaw	integer	人脸水平角度
—pitch	integer	人脸垂直角度
—blurness	float	图像清晰度（0~100）值越小越清晰
—livenessScore	float	活体检测结果（0~100）
—quality	float	图片质量 取值 0~1 值越大质量越大好
—age	integer	年龄
top1 {}	obj	比对结果 TOP1
—group []	array	人脸分组数组列表
— —groupName	string	人脸分组名
—searchScore	float	1:N 相似度（0~100）之间
—faceToken	string	分组中人脸 ID
—description	string	人脸描述，UTF-8 编码
—imageUrl	string	底库中人脸图片资源下下载地址
top2 {}	obj	比对结果 TOP2
—group []	array	人脸分组数组列表
— —groupName	string	人脸分组名
—searchScore	float	1:N 相似度（0~100）之间

参数	类型	说明
—faceToken	string	分组中人脸 ID
—description	string	人脸描述，UTF-8 编码
—imageUrl	string	底库中人脸图片资源下下载地址
top3 {}	obj	比对结果 TOP3
—group []	array	人脸分组数组列表
— —groupName	string	人脸分组名
—searchScore	float	1:N 相似度（0~100）之间
—faceToken	string	分组中人脸 ID
—description	string	人脸描述
—imageUrl	string	底库中人脸图片资源下下载地址
top4 {}	obj	比对结果 TOP4
—group []	array	人脸分组数组列表
— —groupName	string	人脸分组名
—searchScore	float	1:N 相似度（0~100）之间
—faceToken	string	分组中人脸 ID
—description	string	人脸描述
—imageUrl	string	底库中人脸图片资源下下载地址
top5 {}	obj	比对结果 TOP5

参数	类型	说明
—group []	array	人脸分组数组列表
— —groupName	string	人脸分组名
—searchScore	float	1:N 相似度（0~100）之间
—faceToken	string	分组中人脸 ID
—description	string	人脸描述
—imageUrl	string	底库中人脸图片资源下下载地址
featureInfo{}	obj	特征数据信息，输入参数 featureUpload 为 false 时，没有此字段
—feature	string	特征数据，base64 编码
—version	string	模型版本

响应数据示例：

```

"detectionResult":{
    "left":126,
    "top":151,
    "right":306,
    "bottom":337,
    "roll":-1,
    "yaw":0,
    "pitch":-4,
    "blurness":0,
    "livenessScore":0
},

```

```
"top1":{
  "searchScore":58.08,
  "faceToken":"p-Lw5yBtLzLv jFXbP jL7YQ==",
  "description":"袁德坤. jpg",
  "imageUrl":"./face_data/loe153/1te1008/p-Lw5yBtLzLv jFXbP jL7YQ==. jpg",
  "group":[
    {
      "groupName":"PMPM"
    }
  ]
},
"top2":{
  "searchScore":54.03,
  "faceToken":"UHx__H-2ytxY_DfiKo6tkQ==",
  "description":"宋金木. jpg",
  "imageUrl":"./face_data/loe225/1te183/UHx__H-2ytxY_DfiKo6tkQ==. jpg",
  "group":[
    {
      "groupName":"PMPM"
    }
  ]
},
"top3":{
  "searchScore":53.29,
  "faceToken":"mAOMujw55BGWbto86wTmZg==",
  "description":"娄超. jpg",
  "imageUrl":"./face_data/loe270/1te16/mAOMujw55BGWbto86wTmZg==. jpg",
  "group":[
    {
      "groupName":"PMPM"
    }
  ]
}
```

```
    }
  ]
},
"top4":{
  "searchScore":52.7,
  "faceToken":"M_Xj1SrW6P1K7eYX0h4xoA==",
  "description":"彭寿华.jpg",
  "imageUrl":"./face_data/loe443/1te649/M_Xj1SrW6P1K7eYX0h4xoA==.jpg",
  "group":[
    {
      "groupName":"PMPM"
    }
  ]
},
"top5":{
  "searchScore":52.6,
  "faceToken":"lgdQYA75mk_TZHSic9y3Aw==",
  "description":"袁野.jpg",
  "imageUrl":"./face_data/loe483/1te853/lgdQYA75mk_TZHSic9y3Aw==.jpg",
  "group":[
    {
      "groupName":"PMPM"
    }
  ]
}
```

错误码:

状态码	错误码	消息	说明
400	104202	FACE_TOKEN_NOT_EXIST	人脸不存在

11 用户接入验证

11.1 修改验证码 PUT/ keyword

操作类型：

PUT

功能：

修改验证码，当通过默认用户名与验证码登录后，允许修改验证码，修改后，旧验证码将失效。

系统默认用户名：admin（不可更改）。

默认密码：megbox

接口 URI：

/keyword

请求参数：

参数	类型	说明	必填
password	string	用户验证码，最大支持 64 字符，BASE64 转码传输，不可为空	是

返回参数:

无

12 规则权限 API

12.1 假日规则

12.1.1 查询假日规则 GET /access/calendar

操作类型:

GET

功能: 获取假日规则。

接口 URI:

/access/calendar

请求参数:

参数	类型	说明
pageToken	integer	指示列表的起始位置, 为空表未从 0 开始, 与 pageOffset 不可一起使用, 此值首次查询时不用填写, 默认从 0 开始, 后续填入返回的 nextPageToken 值
pageSize	integer	返回结果一页的最大数量, 与 pageToken 配合使用, 如反回结果的总量超过 pageSize, 在返回参数中会有 nextPageToken 字段用于指示下一页起始位置, pageSize 的设置范围为[1, 30]

参数	类型	说明
pageOffset	integer	列表结果偏移量，从 0 开始，与 pageToken 不可一起使用

返回参数：

参数	类型	说明
nextPageToken	integer	下一页的起始地址，放入下一次请求的 pageToken
totalCount	integer	总条数
calendar[]	array	假日列表
— calendarName	string	假日名称，长度不超过 40 字节
— dateRange[]	array	按日期存储的数组，日期字段分别为年，月，日。当年为 0 时，月日不为 0，表示按年重复；当年月为 0 是，日不为 0，表示按月重复；当年月日都不为 0 时，表示不重复， 例如： - 年重复 <pre>date_range: [[0, 8, 14], [0, 8, 28]]</pre> - 月重复 <pre>date_range: [[0, 0, 16], [0, 0, 31], [0, 0, 8]]</pre> - 不重复 <pre>date_range: [[2019, 8, 21], [2019, 8, 7]]</pre>
comment	string	备注，不超过 255 字节

12.1.2 增加假日规则 POST /access/calendar

操作类型:

POST

功能: 新建假日规则, 最多不超过 100 个假日规则。

接口 URI: /access/calendar

请求参数:

参数	类型	说明
calendarName	string	假日名称, 长度不超过 40 字节, 唯一, 不可重复
dateRange[]	array	按日期存储的数组, 不超过 365 个, 日期字段分别为年, 月, 日。当年为 0 时, 月日不为 0, 表示按年重复; 当年月为 0 是, 日不为 0, 表示按月重复; 当年月日都不为 0 时, 表示不重复, 例如: - 年重复 date_range: [[0, 8, 14], [0, 8, 28]] - 月重复 date_range: [[0, 0, 16], [0, 0, 31], [0, 0, 8]] - 不重复 date_range: [[2019, 8, 21], [2019, 8, 7]]
comment	string	备注, 不超过 255 字节

返回参数:

无

12.1.3 删除假日规则 DELETE /access/calendar/{calendarName}

操作类型:

DELETE

功能: 如果 calendar 绑定了 rules 则提示不允许删除, 否则可以删除。

接口 URI: /access/calendar/{calendarName}

请求参数:

{calendarName} 为假日名称

返回参数:

无

12.1.4 修改假日规则 PUT /access/calendar/{calendarName}

操作类型:

PUT

功能: 修改假日规则。

请求参数：

{calendarName} 为原假日名称

参数	类型	说明
calendarName	string	假日名称，长度不超过 40 字节
dateRange[]	array	按日期存储的数组，日期字段分别为年，月，日。当年为 0 时，月日不为 0，表示按年重复；当年月为 0 是，日不为 0，表示按月重复；当年月日都不为 0 时，表示不重复， 例如： - 年重复 date_range: [[0, 8, 14], [0, 8, 28]] - 月重复 date_range: [[0, 0, 16], [0, 0, 31], [0, 0, 8]] - 不重复 date_range: [[2019, 8, 21], [2019, 8, 7]]
comment	string	备注，不超过 255 字节

返回参数：

无

12.2 时段规则

12.2.1 查询时段规则 GET /access/schedule

操作类型:

GET

功能: 查询时段规则。

接口 URI: /access/schedule

请求参数:

参数	类型	说明
pageToken	integer	指示列表的起始位置，为空表未从 0 开始，与 pageOffset 不可一起使用，此值首次查询时不用填写，默认从 0 开始，后续填入返回的 nextPageToken 值
pageSize	integer	返回结果一页的最大数量，与 pageToken 配合使用，如反回结果的总量超过 pageSize，在返回参数中会有 nextPageToken 字段用于指示下一页起始位置，pageSize 的设置范围为[1, 30]
pageOffset	integer	列表结果偏移量，从 0 开始，与 pageToken 不可一起使用

返回参数:

参数	类型	说明
nextPageToken	integer	下一页的起始地址，放入下一次请求的 pageToken
totalCount	integer	总条数
schedule[]	array	时段数组
— scheduleName	string	时段名称，长度不超过 40 字节

参数	类型	说明
—startDate	string	时段生效开始日期，格式为年-月-日，例如 2019-09-12
—endDate	string	时段生效结束日期
— timeRange[]	array	一周七天的数组，每一天一个数组，每一天有三个时间段的数组，时间段里面是每一天的 一段的开始和结束的秒数。 例如： time_range: [[[3600, 86400], [0, 86400], [0, 86400]], [[3600, 86400], [0, 86400], [0, 86400]], [[3600, 86400], [0, 86400], [0, 86400]], [[3600, 86400], [0, 86400], [0, 86400]], [[3600, 86400], [0, 86400], [0, 86400]], [[3600, 86400], [0, 86400], [0, 86400]], [[3600, 86400], [0, 86400], [0, 86400]]]
—comment	string	备注，不超过 255 字节

12.2.2 增加时段规则 POST /access/schedule

操作类型：

POST

功能：新增时段规则，最多不超过 100 条时段规则。

接口 URI：/access/schedule

请求参数：

参数	类型	说明
scheduleName	string	时段名称，长度不超过 40 字节，唯一，不可重复
startDate	string	时段生效开始日期，格式为年-月-日，例如 2019-09-12
endDate	string	时段生效结束日期
timeRange[]	array	一周七天的数组，每一天一个数组，每一天最多有 6 个时间段的数组，时间段里面是每一天的开始的开始和结束的秒数。 例如： time_range: [[[3600, 86400], [0, 86400], [0, 86400]], [[3600, 86400], [0, 86400], [0, 86400]], [[3600, 86400], [0, 86400], [0, 86400]], [[3600, 86400], [0, 86400], [0, 86400]], [[3600, 86400], [0, 86400], [0, 86400]], [[3600, 86400], [0, 86400], [0, 86400]], [[3600, 86400], [0, 86400], [0, 86400]]]
comment	string	备注，不超过 255 字节

返回参数：

无

12.2.3 删除时段规则 DELETE /access/schedule/{scheduleName}

操作类型：

DELETE

功能：删除时段规则，如果 schedule 绑定了 rules 则提示不允许删除，否则可以删除。

接口 URI: /access/schedule/{scheduleName}

请求参数:

{scheduleName} 为时段名称

返回参数:

无

12.2.4 修改时段规则 PUT /access/schedule/{scheduleName}

操作类型:

PUT

功能: 修改时段规则。

接口 URI: /access/schedule/{scheduleName}

请求参数:

{scheduleName} 为原时段名称

参数	类型	说明
scheduleName	string	时段名称, 长度不超过 40 字节
startDate	string	时段生效开始日期, 格式为年-月-日, 例如 2019-09-12
endDate	string	时段生效结束日期

参数	类型	说明
timeRange[]	array	一周七天的数组，每一天一个数组，每一天有三个时间段的数组，时间段里面是每一天的一段的开始和结束的秒数。[0,0]表示无效时段 例如： <pre>time_range: [[[3600, 86400], [0, 86400], [0, 86400]], [[3600, 86400], [0, 86400], [0, 86400]], [[3600, 86400], [0, 86400], [0, 86400]], [[3600, 86400], [0, 86400], [0, 86400]], [[3600, 86400], [0, 86400], [0, 86400]], [[3600, 86400], [0, 86400], [0, 86400]], [[3600, 86400], [0, 86400], [0, 86400]]]</pre>
comment	string	备注，不超过 255 字节

返回参数：

无

12.3 门禁权限规则

12.3.1 查询门禁规则 GET /access/rules

操作类型：

GET

功能：获取门禁规权限则。

接口 URI: /access/rules

请求参数:

参数	类型	说明
pageToken	integer	指示列表的起始位置，为空表未从 0 开始，与 pageOffset 不可一起使用，此值首次查询时不用填写，默认从 0 开始，后续填入返回的 nextPageToken 值
pageSize	integer	返回结果一页的最大数量，与 pageToken 配合使用，如反回结果的总量超过 pageSize，在返回参数中会有 nextPageToken 字段用于指示下一页起始位置，pageSize 的设置范围为[1, 30]
pageOffset	integer	列表结果偏移量，从 0 开始，与 pageToken 不可一起使用

返回参数:

参数	类型	说明
nextPageToken	integer	下一页的起始地址，放入下一次请求的 pageToken
totalCount	integer	总条数
rules[]	array	
—ruleName	string	规则名称，长度不超过 40 字节
—videoNo[]	array	数组内容为 integer，视频通道编号，1~16
—groupNames[]	array	人员组数组
—scheduleNames[]	array	时段规则名称数组
—calendarNames[]	array	假日规则数组

参数	类型	说明
—enable	bool	是否启用，true：启用，false:禁用
—comment	string	备注，不超过 255 字节

12.3.2 增加门禁权限规则 POST /access/rules

操作类型：

POST

功能：新建门禁权限规则，最多可添加 100 条规则。

接口 URI： /access/rules

请求参数：

参数	类型	说明
ruleName	string	规则名称，长度不超过 40 字节，唯一，不可重复，非空
videoNo[]	array	数组内容为 integer，视频通道编号，1~16，非空
groupNames[]	array	人员组数组，非空
scheduleNames[]	array	时段规则名称数组，非空
calendarNames[]	array	假日规则数组
enable	bool	是否启用，true：启用，false:禁用
comment	string	备注，不超过 255 字节

12.3.3 删除门禁权限规则 DELETE /access/rules/{ruleName}

操作类型:

DELETE

功能: 删除门禁权限规则。

接口 URI: /access/rules/{ruleName}

请求参数:

{ruleName} 为门禁权限规则名称

返回参数:

无

12.3.4 修改门禁权限规则 PUT /access/rules/{ruleName}

操作类型:

PUT

功能: 修改门禁权限规则

接口 URI: /access/rules/{ruleName}

请求参数:

{ruleName} 为要修改的门禁权限规则名称

参数	类型	说明
ruleName	string	规则名称，长度不超过 40 字节
videoNo[]	array	数组内容为 integer，视频通道编号，1~16
groupNames[]	array	人员组数组
scheduleNames[]	array	时段规则名称数组
calendarNames[]	array	假日规则数组
enable	bool	是否启用，true: 启用，false:禁用
comment	string	备注，不超过 255 字节

返回参数：

无

13 继电器配置 API

13.1.1 继电器绑定 POST/relay

操作类型：

POST

功能：绑定继电器与视频通道，允许一个继电器绑定多个通道，但一个通道只有一个继电器，可接入最多 16 路继电器。

接口 URI:/relay

请求参数:

参数	类型	说明
videoNo	integer	视频通道编号, 1~16
relayIP	string	继电器 IP 地址
relayMAC	string	继电器 MAC 地址, 选填。
triggerChannel	integer	继电器通道: 0~1
enable	bool	是否启用
relayMode	int	工作模式, 0, 常开, 1, 常关
triggerTime	integer	触发时长, 单位 ms, 100~50000 (即 100ms~50s), 且为 100 的整数倍
delayTime	integer	触发延迟, 单位 s, 0~10s
triggerInterval	integer	连续触发间隔, 单位 ms, 0~50000ms
triggerCondition	integer	触发条件: 1, 抓拍结果触发、2, 识别结果触发、3, 识别&抓拍触发

返回参数:

无

13.1.2 继电器删除 DELETE/relay/{videoNo}

操作类型:

DELETE

功能：解除继电器与视频通道绑定关系

接口 URI： /relay/{videoNo}

请求参数：

{videoNo} 为视频通道编号，1~16

返回参数：

无

13.1.3 查询继电器状态 GET/relay

操作类型：

GET

功能：获取继电器工作状态。

接口 URI： /relay

请求参数：

无

返回参数：

参数	类型	说明
relay[]	array	
videoNo	integer	视频通道编号，1~16
relayIP	string	继电器 IP 地址
relayMAC	string	继电器 MAC 地址
triggerChannel	integer	继电器通道，0~1
enable	bool	是否启用
relayMode	int	工作模式，0，常开，1，常关
triggerTime	integer	触发时长，单位 ms，100~50000(即 100ms~50s)，且为 100 的整数倍
delayTime	integer	触发延迟，单位 s，0~10s
triggerInterval	integer	连续触发间隔，单位 ms，0~50000ms
triggerCondition	integer	触发条件：1，抓拍结果触发、2，识别结果触发、3, 识别&抓拍触发
status	bool	继电器当前输出状态，true:导通；false:断开
online	bool	继电器是否在线

13.1.4 触发继电器 POST /relayTrigger/{videoNo}

操作类型：

POST

功能：触发继电器。

接口 URI: /relayTrigger/{videoNo}

请求参数:

{videoNo} 为视频通道编号, 1~16

返回参数:

无

13.1.5 继电器工作参数设置 POST/relayWorkPara

操作类型:

POST

功能:

设置继电器工作参数。

接口 URI:

/relayPara

参数:

参数	类型	说明
netMode	integer	网络工作模式 0:tcp 1:udp
timeUnit	integer	时间单位 0:100 毫秒 1: 秒

参数	类型	说明
AESKey	string	通信密码

返回值：

无

13.1.6 继电器工作参数获取 GET/relayWorkPara

操作类型：

GET

功能：

获取继电器工作参数。

接口 URI：

/relayPara

参数：

无

返回值：

参数	类型	说明
netMode	integer	网络工作模式 0:tcp 1:udp
timeUnit	integer	时间单位 0:100 毫秒 1: 秒
AESKey	string	通信密码

14 pictures 图片资源操作 API

14.1 创建图片通道 POST /pictures

操作类型:

POST

功能:

接入需要处理的 1 图片流（不支持修改布控基本信息的修改，需要先调用 delete 接口，再使用 post 进行创建）

接口 URI:

/pictures

请求参数:

参数	类型	说明
pictureNo	integer	必填，图片通道编号：支持 32 路图片流接入，值为 17~48，这将是此布点的唯一标识，后续操作都依赖这个名称
pictureName	string	必填，摄像头布点名称，由用户起名，以下规则： • 支持 1-50 字节长度 • 支持英文大/小写字母、数字、半角下划线 • 建议正则/[^][a-zA-Z0-9_]{4,16}
deviceId	string	必填，图片流接入设备标识，限制长度为 20~128 字节，超过则报错，必须是由 0~9 数字组成。
subUrl	string	非必填，视频流子码流地址，通常为局域网内的 RTSP 地址，限制长度为 128 字节，超过则报错。
username	string	非必填，rtsp 接入的用户名，最长 31 字符
password	string	非必填，rtsp 接入的密码，最长 63 字符

返回值：

无

14.2 图片布控 PUT/pictures / {pictureNo} / surveillances

操作类型：

PUT

功能：

对应图片通道编号, 设置关联报警底库、人脸抓拍的过滤条件、比对阈值，并进行布控。

接口 URI：

/serveillances

请求参数：

参数	类型	说明
surveillanceEnabled	bool	布控使能开关
filter{}	obj	过滤条件，条件越高，通过抓拍的人脸将越少，但人脸质量越高，识别的效果会越好。如条件越低，通过抓拍的人脸会越多，但人脸质量下线会降低，影响识别的效果。
--roll	integer	必填，人脸旋转角度限制（0~180），建议<=45°
--yaw	integer	必填，人脸水平角度限制（0~100），建议<=35°
--pitch	integer	必填，人脸垂直角度限制（0~100），建议<=45°
--blurness	integer	必填，图像清晰度限制（0~100），值越小越清晰，建议<=60
--highBrightness	integer	人脸最高亮度（50~100），建议 90
--lowBrightness	integer	人脸最低亮度（0~49），建议 20
--deviation	integer	人脸对比度（0~100），建议 50

参数	类型	说明
		值越高，代表允许人脸对比度越强，但影响识别效果。值越低，代表允许的对比度越低，识别效果会提高，但抓拍率会降低。
—faceMin	integer	必填，人脸允许的长或宽的最小像素（20~380），值越小识别效果越差，建议>=80
group[]	array	如关联多个分组，在数组中添加多个分组对象，每个分组对象带一个分组唯一号与分组的有效时间。
{}	obj	设定关联的分组与有效时间，最多 20 组。
— groupName	string	必填，视频绑定的人脸分组名称，视频中抓拍的人脸将与这个 group 中的人脸进行识别。如高于比对阈值将引发报警或允许通行动作
— startTime	string	必填，该绑定分组有效的开始时间，格式：HHMMSS，例子：“080000”，为 24 小时制，默认为“000000”
— endTime	string	必填，该绑定分组失效的开始时间，格式：HHMMSS，例子：“184500”，为 24 小时制，默认为“000000”，如 startTime 与 endTime 两个值都为默认值“000000”，将 24 小时有效
— searchThresholdHigh	integer	必填，人脸识别的高限阈值，（0~100）之间的数，如最高相似度的超过阈值，可通过 Alert 接口得到超阈值的不同组里的比对相似度列表。建议 70~80，具体值根据实际情况调整
— searchThresholdLow	integer	必填，人脸识别的低限阈值，（0~100）之间的数，如最高相似度低于低线阈值，代表这是一个陌生人。建议：20，可通过 Alert 接口得到低阈值的不同组里的比对相似度列表
globalStranger {}	obj	非必填，全局陌生人配置
—enable	bool	必填，配置开关 true 启用 false 禁用

参数	类型	说明
—searchThresholdLow	integer	必填，人脸识别的低限阈值，（0~100）之间的数，如最高相似度低于低线阈值，代表这是一个陌生人。建议：20，可通过 Alert 接口得到低阈值的不同组里的比对相似度列表
—startTime	string	必填，该绑定分组有效的开始时间，格式：HHMMSS，例子：“080000”，为24小时制，默认为“000000”
—endTime	string	必填，该绑定分组失效的开始时间，格式：HHMMSS，例子：“184500”，为24小时制，默认为“000000”，如 startTime 与 endTime 两个值都为默认值“000000”，将24小时有效
livenessEnabled	bool	是否开启活体检测，默认开启
livenessThreshold	integer	如打开活体判断功能必填，活体判断阈值，（0~100）之间的数，建议60，当活体开关关闭时，该值非必须，如设置为0也相当于关闭，但会消耗系统资源。
bestStrangerUpload	bool	非必填，是否上报全局最优陌生人，默认为false

返回参数：

无

14.3 获取所有图片布点信息 GET/pictures

操作类型：

GET

功能：

从设备中获取当前布点的图片流设置列表

接口 URI:

/pictures

请求参数:

无

返回参数:

参数	类型	说明
pictures[]	array	放置图片布点信息的数组，内部可能会有多组图片布点信息。
{}	obj	一组布控信息在此对象中，如果有多组布控将有多组对象
—pictureNo	integer	必填，图片通道编号：支持 32 路图片，值为 17~48
—pictureName	string	必填，摄像头布点名称，由用户起名,这将是此布点的唯一标识，后续操作都依赖这个名称。以下规则： 1 支持 1-50 字节长度 1 支持英文大/小写字母、数字、半角下划线 建议正则 /^[a-zA-Z0-9_]{4,16}
—deviceId	string	必填，设备标识 20~128 字节。
—subUrl	string	非必填，子码流,通常为局域网内的 RTSP 地址。

参数	类型	说明
—username	string	非必填，rtsp 接入的用户名，最长 31 字符
—password	string	非必填，rtsp 接入的密码，最长 63 字符
—livenessEnabled	bool	是否开启活体检测
—livenessThreshold	float	活体判断阈值，（0~100）之间的浮点数，当活体开关关闭时，该值非必须，如设置为 0 也相当于关闭，但会消耗系统资源。如需要关闭该功能可到 configs 公共参数设置接口进行设置，但所有视频都将不做活体检测。
—surveillanceEnabled	bool	布控使能开关
—isOnline	bool	是否在线 true:在线 false:不在线
—filter{}	obj	过滤条件，条件越高，通过抓拍的人脸将越少，但人脸质量越高，识别的效果会越高。如条件越低，通过抓拍的人脸会越多，但人脸质量下线会降低，影响识别的效果。
— —roll	integer	人脸旋转角度限制（0~180），建议<=45°
— —yaw	integer	人脸水平角度限制（0~100），建议<=35°
— —pitch	integer	人脸垂直角度限制（0~100），建议<=45°
— —blurriness	float	图像清晰度限制（0~100），建议<=60
— —highBrightness	integer	人脸最高亮度（50~100），建议 90
— —lowBrightness	integer	人脸最低亮度（0~49），建议 20
— —deviation	integer	人脸对比度（0~100），建议 50

参数	类型	说明
		值越高，代表允许人脸对比度越强，但影响识别效果。值越低，代表允许的对比度越低，识别效果会提高，但抓拍率会降低。
— —faceMin	integer	人脸允许的长或宽的最小像素（20~380），值越小识别效果越差，建议>=80
— —group[]	array	如关联多个分组，在数组中添加多个项
{}	obj	设定关联的分组与有效时间，最多 20 组。
— —groupName	string	必填，视频绑定的人脸分组名称，视频中抓拍的人脸将与这个 group 中的人脸进行识别。如高于比对阈值将引发报警或允许通行动作
— — startTime	string	必填，该绑定分组有效的开始时间，格式：HHMMSS，例子：“080000”，为 24 小时制，默认为“000000”
— — endTime	string	必填，该绑定分组失效的开始时间，格式：HHMMSS，例子：“184500”，为 24 小时制，默认为“000000”，如 startTime 与 endTime 两个值都为默认值“000000”，将 24 小时有效，不支持跨天
— —searchThresholdHigh	float	必填，人脸识别的高限阈值，（0~100）之间的浮点数，如最高相似度的超过阈值，可通过 Alert 接口得到超阈值的不同组里的比对相似度列表。建议 70~80，具体值根据实际情况调整
— —searchThresholdLow	float	必填，人脸识别的低限阈值，（0~100）之间的浮点数，如最高相似度低于低线阈值，代表这是一个陌生人。可通过 Alert 接口得到低阈值的不同组里的比对相似度列表
— —globalStranger {}	obj	非必填，全局陌生人配置
— —enable	bool	必填，配置开关 true 启用 false 禁用

参数	类型	说明
— — searchThresholdLow	integer	必填，人脸识别的低限阈值，（0~100）之间的数，如最高相似度低于低线阈值，代表这是一个陌生人。建议：20，可通过 Alert 接口得到低阈值的不同组里的比对相似度列表
— —startTime	string	必填，该绑定分组有效的开始时间，格式：HHMMSS，例子：“080000”，为24小时制，默认为“000000”
— —endTime	string	必填，该绑定分组失效的开始时间，格式：HHMMSS，例子：“184500”，为24小时制，默认为“000000”，如 startTime 与 endTime 两个值都为默认值“000000”，将24小时有效

14.4 删除某图片布点 DELETE/pictures/{pictureNo}

操作类型：

DELETE

功能：

删除设备中某个图片流布点，同时删除所有配置与绑定关系。另与该布点相关的历史抓拍、报警信息也将全部删除。

接口 URI：

/pictures/{pictureNo}

请求参数：

无
无返回参数：

无

14.5 编辑图片通道 PUT/pictures/{pictureNo}

操作类型：

PUT

功能：

编辑图片通道信息

接口 URI：

/pictures

请求参数：

参数	类型	说明
pictureName	string	必填，摄像头布点名称，由用户起名，以下规则： - 支持 1-50 字节长度 - 支持英文大/小写字母、数字、半角下划线 - 建议正则/[^][a-zA-Z0-9_]{4,16}
deviceId	string	必填，图片流接入设备标识，限制长度为 20~128 字节，超过则报错，必须是由 0~9 数字组成。

参数	类型	说明
subUrl	string	非必填，视频流子码流地址, 通常为局域网内的 RTSP 地址，限制长度为 128 字节，超过则报错。
username	string	非必填，rtsp 接入的用户名，最长 31 字符
password	string	非必填，rtsp 接入的密码，最长 63 字符

返回值：

无

15 ROI 区域资源 API

15.1 获取 ROI 区域 GET /areaROI/{videoNO}

操作类型：

GET

功能：

获取 ROI 区域

接口 URI：

/areaROI/{videoNO}

参数：

无

返回值:

参数	类型	说明
enable	bool	是否启用
area[]	array	最多 5 个
— posX	integer	X 坐标
— posY	integer	Y 坐标

15.2 设置 ROI 区域 POST/areaROI

操作类型:

POST

功能:

设置 ROI 区域，每条通道视频可配置一个 ROI 区域，ROI 区域可设置为不规则 N 边形， $3 \leq N \leq 5$

接口 URI:

/areaROI

请求参数:

参数	类型	说明
videoNo	integer	视频通道编号，1~16
enable	bool	是否启用
area[]	array	最多 5 个
— posX	integer	X 坐标（0 ~ 1920）
— posY	integer	Y 坐标（0 ~ 1080）

返回参数：

无

15.3 删除 ROI 区域 DELETE /areaROI/{videoNO}

操作类型：

DELETE

功能：

删除 ROI 区域

接口 URI：

/areaROI/{videoNO}

请求参数：

无

返回参数:

无

16 GA /T 1400 客户端功能协议

16.1 设置 GA /T 1400 客户端工作参数 POST/gat1400

操作类型:

POST

功能:

设置 GAT1400 客户端工作参数

接口 URI:

/gat1400

请求参数:

参数	类型	说明 j gii	是否必选
devInfo[]	array	设备配置信息	是
—videoNo	integer	视频通道编号, 1~48	否
—deviceId	string	设备编号, 20 字节	否
enable	bool	是否启用	是
serverAddr	string	GA/T 1400 服务器 IP 地址	是
serverPort	integer	GA/T 1400 服务器端口号, 取值范围 1~65535	是
username	string	GA/T 1400 服务器用户名, 0~64 个字节	是
password	string	GA/T 1400 服务器密码, 0~64 个字节	是
keepInterval	integer	保活间隔, 单位 s, 30~300s	是
syncTimeEnable	bool	是否启用校时	否
syncTimeInterval	integer	同步时间间隔 1~65535	否
retransmissionTimes	integer	重传次数 0~3	否
retransmissionInterval	integer	重传间隔 1~5	否
platformNo	integer	视图库 ID 1~100	否
backgroundUpload	bool	是否上传背景图	否

返回参数:

无

16.2 获取 GA /T 1400 客户端工作参数 GET/gat1400

操作类型:

GET

功能:

获取 GAT1400 客户端工作参数

接口 URI:

/gat1400

请求参数:

无

返回参数：

参数	类型	说明	是否必选
devInfo[]	array	设备配置信息	是
—videoNo	integer	视频通道编号，1~48	否
—deviceId	string	设备编号，20 位	否
enable	bool	是否启用	是
serverAddr	string	GA/T 1400 服务器 IP 地址	是
serverPort	integer	GA/T 1400 服务器端口号	是
username	string	GA/T 1400 服务器用户名	是
password	string	GA/T 1400 服务器密码	是
keepInterval	integer	保活间隔，单位 s，30~300s	是
syncTimeEnable	bool	是否启用校时	否
syncTimeInterval	integer	同步时间间隔 1~65535	否
retransmissionTimes	integer	重传次数 0~3	否
retransmissionInterval	integer	重传间隔 1~5	否
platformNo	integer	视图库 ID 1~100	否

参数	类型	说明	是否必选
backgroundUpload	bool	是否上传背景图	否

16.3 获取客户端运行状态 GET/gat1400Status

操作类型：

GET

功能：

获取 GAT1400 客户端运行状态

接口 URI：

/gat1400Status

请求参数:

无

返回参数:

参数	类型	说明
status	integer	GA /T 1400 服务运行状态：0-未启用；1-未连接；2-已连接

17 GA /T 1400 服务器协议

17.1 设置 GA /T 1400 server POST/gat1400server

请求参数:

参数	类型	说明	是否必选
enable	bool	是否启用，默认 false	是
serverPort	integer	GA/T 1400 服务器端口号，取值范围 1024~65535，默认 8000，不可设置 6000/6666	是
authEnable	bool	是否开启认证，如果开启，则以下 authMode、username 和 password 为必填。默认 false	是
authMode	string	认证模式，支持： “basic”，basic 认证 “digest”，digest 认证（目前只支持 digest）	否
username	string	GA/T 1400 认证用户名，1~64 个字节	否
password	string	GA/T 1400 认证密码，1~64 个字节	否
keepInterval	integer	保活间隔，单位 s，30~300s，默认 90s	是

返回参数：

无

17.2 获取 GA /T 1400 server 配置 GET/gat1400server

请求参数：

无

返回参数：

参数	类型	说明
enable	bool	是否启用
serverPort	integer	GA/T 1400 服务器端口号，取值范围 1024~65535，默认 8000
authEnable	bool	是否开启认证
authMode	string	认证模式
username	string	GA/T 1400 认证用户名，1~64 个字节
password	string	GA/T 1400 认证密码，1~64 个字节
keepInterval	integer	保活间隔，单位 s，30~300s

18 云桥主动对接模式协议

18.1 设置云桥主动对接模式 POST/megActive

请求参数：

参数	类型	说明	是否必选
enable	bool	是否启用，默认 false	是
ip	string	云桥服务器的 ip 地址	是

参数	类型	说明	是否必选
port	integer	云桥服务器的端口号	是
name	string	本机在云桥服务器注册的设备名	是
id	string	当前设备被分配的 APPID，由云桥提供，默认可填 0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ	是
key	string	当前设备被分配的 APPKEY，由云桥提供，默认可填 0123456789abcdefghijklmnopqrstuvwxyz	是

返回参数：

无

18.2 获取云桥主动对接模式配置 GET/megActive

请求参数：

无

返回参数：

参数	类型	说明	是否必选
enable	bool	是否启用，默认 false	是
ip	string	云桥服务器的 ip 地址	是

参数	类型	说明	是否必选
port	integer	云桥服务器的端口号	是
name	string	本机在云桥服务器注册的设备名	是
id	string	当前设备被分配的 APPID，由云桥提供，默认可填 0123456789ABCDEFGHIJKLMNQPQRSTU	是
key	string	当前设备被分配的 APPKEY，由云桥提供，默认可填 0123456789abcdefghijklmnopqrstu	是

19 FTP 推图

19.1.1 FTP 推图参数设置 PUT /ftpUpload

操作类型：

PUT

功能：

设置 FTP 推图的相关参数

接口 URI：

/ftpUpload

请求参数：

参数	类型	说明
----	----	----

enable	bool	true: 使能 FTP 上传, false: 禁用 FTP 上传。默认禁用
serverAddr	string	服务器地址
serverPort	integer	端口号, 取值范围[1, 65535], 默认为 21, 非空(无论是否使能)
username	string	用户名, 最大支持 256 BYTE
password	string	密码, 最大支持 256 BYTE
remoteDir	string	远程目录, 最大支持 256 BYTE
enableBackground	bool	true: 使能背景大图上传, false: 禁用背景大图上传。默认禁用
pathMode[]	array	目录层级, 固定为包含 3 个元素的数组, 可选项: 0 : 表示无, 1: 表示按照通道号_相机名称创建目录(如果相机没有名称, 则只按通道号命名), 2: 表示按照设备编号创建目录, 3: 表示按照日期创建目录, 4: 表示按照自定义方式创建目录。 例如: "path_mode": [3, 1, 2] 表示按照: 日期/通道号_相机名称/设备编号/ 的形式创建分层目录
customPath	string	如果上面的设置方式包含了自定义模式, 则必须非空, 最大支持 256 BYTE

返回参数:

无

19.1.2 FTP 推图参数获取 GET/ftpUpload

操作类型:

GET

功能：

获取 FTP 推图的相关参数

接口 URI：

/ftpUpload

请求参数：

无

返回参数：

参数	类型	说明
enable	bool	true：使能 FTP 上传，false：禁用 FTP 上传
serverAddr	string	服务器地址
serverPort	integer	端口号，取值范围[1,65535]
username	string	用户名，最大支持 256 BYTE
password	string	密码，最大支持 256 BYTE
remoteDir	string	远程目录，最大支持 256 BYTE
enableBackground	bool	true：使能背景大图上传，false：禁用背景大图上传
pathMode[]	array	目录层级，固定为包含 3 个元素的数组，可选项：0 ：表示无，1：表示按照通道号_相机名称创建目录(如果相机没有名称，则只按通道号命名)，2：表示按照设备编号创建目录，3：表示按照日期创建目录，4：表示按照自定义方式创建目录。

		例如: "path_mode": [3, 1, 2] 表示按照: 日期/通道号_相机名称/设备编号/ 的形式创建分层目录
customPath	string	如果上面的设置方式包含了自定义模式, 则必须非空, 最大支持 256 BYTE

19.1.3 FTP 推图连接测试 POST/ftpUpload

操作类型:

POST

功能:

FTP 推图连接测试

接口 URI:

/ftpUpload

请求参数:

参数	类型	说明
serverAddr	string	服务器地址
serverPort	integer	端口号
username	string	用户名, 最大支持 256 BYTE
password	string	密码, 最大支持 256 BYTE

返回参数：

返回码	说明
0	FTP 连接测试成功
104618	FTP 连接测试失败

19.2 目录设置规则

各级目录可选的项与限制关系：

一级目录	二级目录	三级目录
无（存根目录）	✕	✕
相机名称	无	✕
	日期	✕
设备编号（默认）	无	✕
	通道号+相机名称	无
		日期
	日期（默认）	无
		相机名称（默认）
	无	✕

日期	设备编号	无
		相机名称
	通道号+相机名称	无
		设备编号
自定义	无	✕
	通道号+相机名称	无
		日期
	日期	无
		相机名称

19.3 图片文件命名规则

设备 ID_时间戳_通道_图片类型_目标类型_TrackID_{绑定的目标类型_绑定目标的 TrackID}. jpg

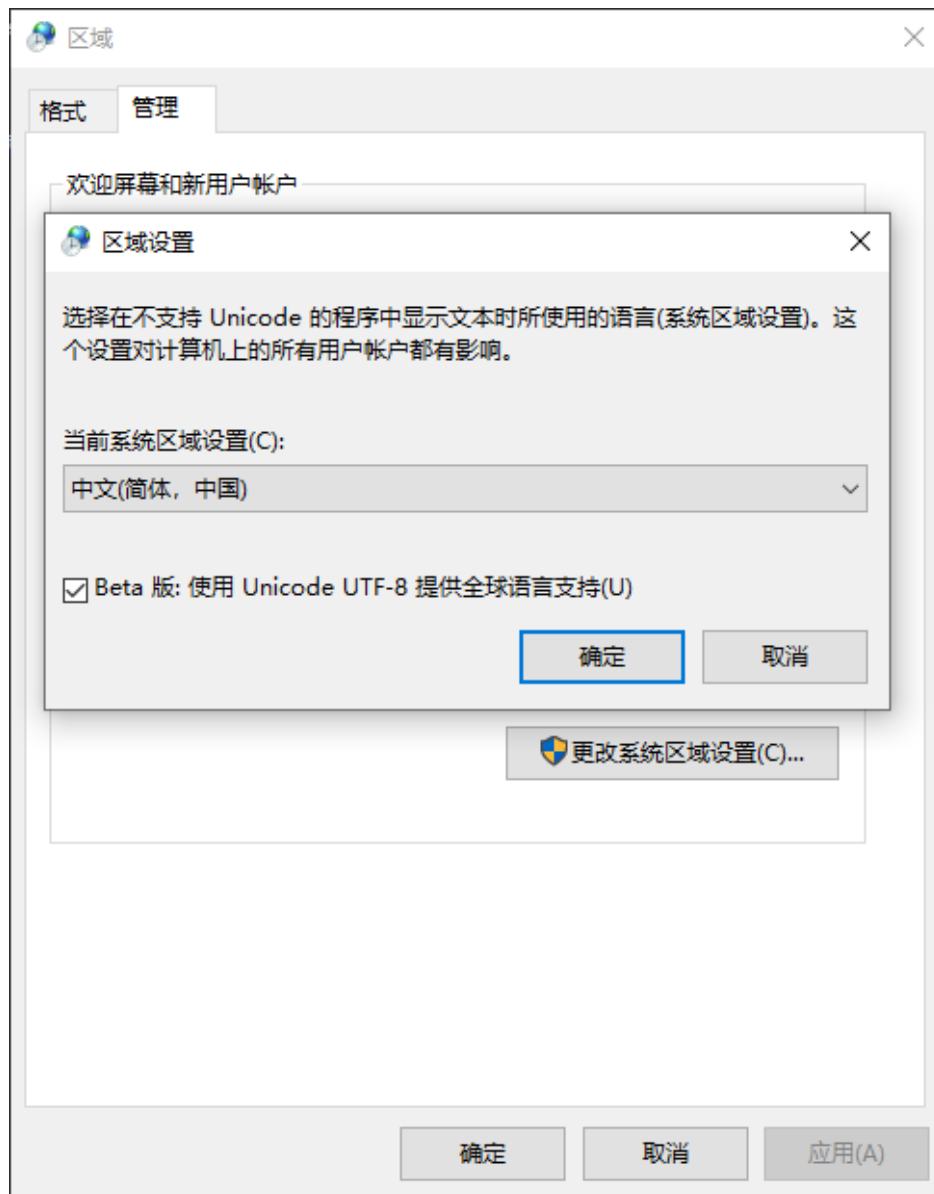
设备 ID	设备 MAC (无 ':' 分隔符)
时间戳	年月日时分秒毫秒
通道	通道号 01、02 ... 16

通道名称	CH+通道名称
图片类型	• 抓拍图-01 • 底库图-02 • 背景图-03
目标类型	• 人脸-01 • 人体-02 • 机动车-03 • 非机动车-04 • 车牌-05
目标 TrackID 后 4 位	
绑定的目标类型	• 人脸-01 • 人体-02 • 机动车-03 • 非机动车-04 • 车牌-05
绑定对象 TrackID 后 4 位	

- 绑定关系仅会出现在抓拍图中，其它类型图片中不会出现
- 如果不存在绑定关系，则省略“_{绑定的目标类型_绑定目标的 TrackID}”
- 如果存在多个绑定关系，则依次填写绑定目标类型和绑定目标的 TrackID，例如 “_{01_7657_04_7659}”

19.4 注意事项

Windows 上搭建 FTP 服务器时，若使用中文名称需要配置系统的字符编码格式，配置完成后应重启计算机以生效。



20 Structures 视频结构化资源操作 API

20.1.1 布控 POST/ structures / {videoNo} / surveillances

操作类型:

POST

功能:

对应视频流/图片流编号, 设置关联报警底库、结构化、人脸抓拍的过滤条件、比对阈值, 并进行布控。

接口 URI:

/surveillances

参数	类型	说明	
surveillanceEnabled	bool	布控使能开关	
attributeEnabled	bool	是否开启属性分析, 默认开启	
surveillanceType	integer	布控类型: 0-人脸(脸、脸人绑定), 1-结构化, 默认人脸。人脸时, 需填写后续的 faces 对象; 结构化时, 需填写后续的 structures 对象	
ruleIndex	integer	算法规则索引, 暂只支持同时布控 2 种规则, 取值 1 ~ 2	否
ruleEnable	bool	算法规则使能, 控制算法规则是否启用	否
structures {}	obj	结构化参数配置	
—detectTargetList []	array	检测目标数组, 至少包含如下五类目标中的一种	是

— —targetName	string	检测目标名称，可取值如下，默认全选： (1) 人脸："face" (2) 人形："human" (3) 机动车："motorVehicle"， (4) 非机动车："nonMotorVehicle" (5) 车牌："plate"	是
— —pushConfigs {}	obj	推图参数配置	
— —pushMode	integer	推图模式：0 - 间隔推图模式，1 - 最优推图模式，2 - 最快推图模式，3 - 无限推图	
— —pushInterval	integer	当推图模式设置为间隔推图时，推图间隔。（500-10000，单位是毫秒，默认 500ms）	
— —maxPushNumber	integer	最大张数推图张数。限定条件：1~10 张，默认 1 张	
— —captureConfigs {}	obj	抓拍模式参数配置	
— —captureMode	integer	抓拍模式，默认为高质量模式： 1 0：全抓模式 算法检测到目标（人、车、脸、车牌），不用质量判断模型过滤目标，抓到的目标全推出来 1 1：高质量模式	是

		只抓拍满足“质量阈值”的目标，其中质量阈值的定义针对不同目标包含不同维度 1 2：自定义模式 只抓拍满足“用户定义质量阈值”的目标	
— —quality{}	obj	视频结构化质量，非必选。上述抓拍模式配置为”自定义模式“时使用	否
— — —faceQuality	integer	人脸综合质量 取值范围【0,100】	否
— — —humanQuality	integer	人形综合质量 取值范围【0,100】	否
— — —motorVehicleQuality	integer	机动车综合质量 取值范围【0,100】	否
— — —nonMotorVehicleQuality	integer	非机动车综合质量 取值范围【0,100】	否
faces{}	obj	人脸（纯人脸、脸人绑定）参数配置	
—livenessEnabled	bool	必填，是否开启活体	
—livenessThreshold	integer	活体阈值。如上述活体开启，则此项必填。活体判断阈值，（0~100）之间的数，建议 60。如设置为 0 也相当于关闭，但会消耗系统资源。	
—pushConfigs{}	obj	推图参数配置	
— —pushMode	integer	推图模式：1 - 最优推图模式，2 - 最快推图模式	
— —retryCount	integer	二次识别次数。限定条件：0~5000 次范围，0 代表不进行二次识别；默认 5 次。仅在最快推图模式下生效	
— —retryInterval	integer	二次识别的间隔时长。限定条件：0~60 秒范围，0 代表关闭二次识别功能；默认为 5 秒。仅在最快推图模式下生效	

— —strangerRetryCount	integer	陌生人重新判定次数。对识别结果进行计数，如结果为未识别或陌生人，连续判断平均值低于陌生人阈值，将判定为陌生人。如为陌生人，将产生陌生人报警输出。范围：0~5 次，默认为 2 次。	
— —pushInterval	integer	最小推图间隔。（500-10000，单位是毫秒，默认 500ms）	
— —maxPushNumber	integer	最优推图的最大张数推图张数。限定条件：1~10 张，默认 1 张。当前不可配，固定为 1	
— —filter{}	obj	过滤条件，条件越高，通过抓拍的人脸将越少，但人脸质量越高，识别的效果会越好。如条件越低，通过抓拍的人脸会越多，但人脸质量下线会降低，影响识别的效果。	
— —roll	integer	必填，人脸旋转角度限制（0~180），建议<=45°	
— —yaw	integer	必填，人脸水平角度限制（0~100），建议<=35°	
— —pitch	integer	必填，人脸垂直角度限制（0~100），建议<=45°	
— —blurness	integer	必填，图像清晰度限制（0~100），值越小越清晰，建议<=60	
— —faceMin	integer	必填，人脸允许的长或宽的最小像素（20~380），值越小识别效果越差，建议>=80	
— —bodyMin	integer	必填，人体允许的长或宽的最小像素（20~380），值越小识别效果越差，建议>=80	
— —detectTargetList[]	array	检测目标数组，至少包含如下 2 类目标中的一种	是
— —targetName	string	检测目标名称，可取值如下，默认人脸： (1)人脸：“face” (2)人形：“human”	是

—captureConfigs{}	obj	抓拍模式参数配置	
— —captureMode	integer	抓拍模式，默认为全抓拍模式： 1 0：全抓模式 算法检测到目标（人、车、脸、车牌），不用质量判断模型过滤目标，抓到的目标全推出来 1 1：高质量模式，默认模式 只抓拍满足“质量阈值”的目标，其中质量阈值的定义针对不同目标包含不同维度 1 2：自定义模式 只抓拍满足“用户定义质量阈值”的目标	
— —quality{}	obj	视频结构化质量，非必选。上述抓拍模式配置为”自定义模式“时使用	
— — —faceQuality	integer	人脸综合质量 取值范围【0,100】	
— — —humanQuality	integer	人形综合质量 取值范围【0,100】	
—group[]	array	如关联多个分组，在数组中添加多个分组对象，每个分组对象带一个分组唯一号与分组的有效时间。	
{}	obj	设定关联的分组与有效时间，最多 20 组。	
— — groupName	string	必填，视频绑定的人脸分组名称，视频中抓拍的人脸将与这个 group 中的人脸进行识别。如高于比对阈值将引发报警或允许通行动作	

— — startTime	string	必填，该绑定分组有效的开始时间，格式：HHMMSS，例子：“080000”，为 24 小时制，默认为“000000”	
— — endTime	string	必填，该绑定分组失效的开始时间，格式：HHMMSS，例子：“184500”，为 24 小时制，默认为“000000”，如 startTime 与 endTime 两个值都为默认值“000000”，将 24 小时有效	
— — searchThresholdHigh	integer	必填，人脸识别的高限阈值，（0~100）之间的数，如最高相似度的超过阈值，可通过 Alert 接口得到超阈值的不同组里的比对相似度列表。建议 70~80，具体值根据实际情况调整	
— — searchThresholdLow	integer	必填，人脸识别的低限阈值，（0~100）之间的数，如最高相似度低于低线阈值，代表这是一个陌生人。建议：20，可通过 Alert 接口得到低阈值的不同组里的比对相似度列表	
—globalStranger{}	obj	非必填，全局陌生人配置	
— —enable	bool	必填，配置开关 true 启用 false 禁用	
— —searchThresholdLow	integer	必填，人脸识别的低限阈值，（0~100）之间的数，如最高相似度低于低线阈值，代表这是一个陌生人。建议：20，可通过 Alert 接口得到低阈值的不同组里的比对相似度列表	
— —startTime	string	必填，该绑定分组有效的开始时间，格式：HHMMSS，例子：“080000”，为 24 小时制，默认为“000000”	
— —endTime	string	必填，该绑定分组失效的开始时间，格式：HHMMSS，例子：“184500”，为 24 小时制，默认为“000000”，如 startTime 与 endTime 两个值都为默认值“000000”，将 24 小时有效	
bestStrangerUpload	bool	非必填，是否上报全局最优陌生人，默认为 false	

返回参数:

无

错误码:

状态码	错误码	消息	说明
400	104000	ILLEGAL_JSON_BODY	json 结构解析错误
400	104001	ILLEGAL_ARGUMENT:<args>	参数格式错误, <args>返回错误参数名, 逗号分隔
400	104002	MISSING_ARGUMENT:<args>	缺少必要参数, <args>返回缺少参数名, 逗号分隔
400	104006	REQUEST_TIMEOUT	请求超时
400	104009	NO_VIDEO	无指定视频通道, 未创建或通道与名称不匹配
400	104019	HASHRATE_OVER_LIMIT	算法能力超限

20.1.2 获取所有布控信息 GET/structures

操作类型:

GET

功能:

从设备中获取当前布控的结构化信息

接口 URI:

/structures

请求参数:

无

返回参数:

参数	类型	说明
videos[]	array	放置视频布点信息的数组，内部可能会有多组视频信息。
{}	obj	一组布控信息在此对象中，如果有多组布控将有多个对象
—videoNo	integer	视频通道编号：支持 16 路视频，值为 1~16
—videoName	string	摄像头布点名称，由用户起名,这将是此布点的唯一标识，后续操作都依赖这个名称。以下规则： 1 支持 1-50 字节长度 1 支持英文大/小写字母、数字、半角下划线 建议正则 /^[a-zA-Z0-9_]{4,16}
—url	string	主码流,通常为局域网内的 RTSP 地址。
—subUrl	string	子码流,通常为局域网内的 RTSP 地址。
—username	string	IPC 接入的用户名
—password	string	IPC 接入的密码
—ruleIndex	integer	算法规则索引，暂只支持同时布控 2 种规则，取值 1 ~ 2

参数	类型	说明
—ruleEnable	bool	算法规则使能，控制算法规则是否启用
—rotation	integer	图像旋转的角度，为配合不同的摄像头安装角度，目前只支持 4 个值：0，90.180，270。分别代表顺时针旋转的角度 0°，90°，180°，270°，如错误将检测不到人脸。如输入不为以上 4 个值，将默认为 0° 来处理。
—livenessEnabled	bool	是否开启活体
—livenessThreshold	float	活体判断阈值，（0~100）之间的浮点数，当活体开关关闭时，该值非必须，如设置为 0 也相当于关闭，但会消耗系统资源。如需要关闭该功能可到 configs 公共参数设置接口进行设置，但所有视频都将不做活体检测。
—surveillanceEnabled	bool	布控使能开关
—videoConStatus	integer	视频流连接状态 0:连接成功 1:连接失败 2:认证失败 3:视频流格式错误 4:超出通道最大分辨率 5:解码失败 6:超出最大解码能力
—attributeEnabled	bool	是否开启属性分
—surveillanceType	integer	布控类型：0-人脸（纯人脸、脸人绑定），1-结构化
—detectTargetList[]	array	检测目标数组，至少包含如下五类目标中的一种
— —targetName	string	检测目标名称，可取值如下，默认全选： (1) 人脸："face" (2) 人形："human" (3) 机动车："motorVehicle"，

参数	类型	说明
		(4) 非机动车: "nonMotorVehicle" (5) 车牌: "plate"
—pushConfigs {}	obj	推图参数配置
— —pushMode	string	0- 间隔推图模式, 1- 最优推图模式, 2 - 最快推图模式, 3 - 无限推图
— —retryCount	integer	二次识别次数。限定条件: 0~5000 次范围, 0 代表不进行二次识别; 默认 5 次。仅在最快推图模式下生效
— —retryInterval	integer	二次识别的间隔时长。限定条件: 0~60 秒范围, 0 代表关闭二次识别功能; 默认为 5 秒。仅在最快推图模式下生效
— —strangerRetryCount	integer	陌生人重新判定次数。对识别结果进行计数, 如结果为未识别或陌生人, 连续判断平均值低于陌生人阈值, 将判定为陌生人。如为陌生人, 将产生陌生人报警输出。范围: 0~5 次, 默认为 2 次。
— —pushInterval	integer	最小推图间隔。(500-10000, 单位是毫秒, 默认 500ms)
— —maxPushNumber	integer	最优推图的最大张数推图张数。限定条件: 1~10 张, 默认 1 张
—captureConfigs {}	obj	抓拍模式参数配置
— —captureMode	integer	抓拍模式, 默认为全抓拍模式: 1 0: 全抓模式 算法检测到目标(人、车、脸、车牌), 不用质量判断模型过滤目标, 抓到的目标全推出来

参数	类型	说明
		1 1: 高质量模式，默认模式 只抓拍满足“质量阈值”的目标，其中质量阈值的定义针对不同目标包含不同维度 1 2: 自定义模式 只抓拍满足“用户定义质量阈值”的目标
— —quality{}	obj	视频结构化质量，非必选。上述抓拍模式配置为”自定义模式“时使用
— — —faceQuality	integer	人脸综合质量 取值范围【0, 100】
— — —humanQuality	integer	人形综合质量 取值范围【0, 100】
— — — motorVehicleQuality	integer	机动车综合质量 取值范围【0, 100】
— — — nonMotorVehicleQuality	integer	非机动车综合质量 取值范围【0, 100】
— — —plateQuality	integer	车牌综合质量 取值范围【0, 100】
—filter{}	obj	过滤条件，条件越高，通过抓拍的人脸将越少，但人脸质量越高，识别的效果会越高。如条件越低，通过抓拍的人脸会越多，但人脸质量下线会降低，影响识别的效果。
— —roll	integer	人脸旋转角度限制（0~180），建议<=45°
— —yaw	integer	人脸水平角度限制（0~100），建议<=35°
— —pitch	integer	人脸垂直角度限制（0~100），建议<=45°

参数	类型	说明
— —blurness	float	图像清晰度限制（0~100），建议<=60
— —faceMin	integer	人脸允许的长或宽的最小像素（20~380），值越小识别效果越差，建议>=80
— —bodyMin	integer	人体允许的长或宽的最小像素（20~380），值越小识别效果越差，建议>=80
—group[]	array	如关联多个分组，在数组中添加多个项
{}	obj	设定关联的分组与有效时间，最多 20 组。
— —groupName	string	必填，视频绑定的人脸分组名称，视频中抓拍的人脸将与这个 group 中的人脸进行识别。如高于比对阈值将引发报警或允许通行动作
— — startTime	string	必填，该绑定分组有效的开始时间，格式：HHMMSS，例子：“080000”，为 24 小时制，默认为“000000”
— — endTime	string	必填，该绑定分组失效的开始时间，格式：HHMMSS，例子：“184500”，为 24 小时制，默认为“000000”，如 startTime 与 endTime 两个值都为默认值“000000”，将 24 小时有效，不支持跨天
— —searchThresholdHigh	float	必填，人脸识别的高限阈值，（0~100）之间的浮点数，如最高相似度的超过阈值，可通过 Alert 接口得到超阈值的不同组里的比对相似度列表。建议 70~80，具体值根据实际情况调整
— —searchThresholdLow	float	必填，人脸识别的低限阈值，（0~100）之间的浮点数，如最高相似度低于低线阈值，代表这是一个陌生人。可通过 Alert 接口得到低阈值的不同组里的比对相似度列表
—globalStranger{}	obj	非必填，全局陌生人配置
— —enable	bool	必填，配置开关 true 启用 false 禁用

参数	类型	说明
— —searchThresholdLow	integer	必填，人脸识别的低限阈值，（0~100）之间的数，如最高相似度低于低线阈值，代表这是一个陌生人。建议：20，可通过 Alert 接口得到低阈值的不同组里的比对相似度列表
— —startTime	string	必填，该绑定分组有效的开始时间，格式：HHMMSS，例子：“080000”，为24小时制，默认为“000000”
— —endTime	string	必填，该绑定分组失效的开始时间，格式：HHMMSS，例子：“184500”，为24小时制，默认为“000000”，如 startTime 与 endTime 两个值都为默认值“000000”，将24小时有效

响应数据示例：

```

"videos": [
  {
    "videoNo": 1,
    "videoName": "chn1",
    "url": "rtsp://10.170.160.3/111",
    "subUrl": "rtsp://10.170.160.4/122",
    "username": "",
    "password": "",
    "rotation": 0,
    "livenessEnabled": true,
    "livenessThreshold": 50,
    "surveillanceType": 0,
    "attributeEnabled": true,
    "pushConfigs": {

```



```
    "pushMode": 3,
    "retryCount": 2,
    "retryInterval": 10,
    "minPushNumber": 0,
    "pushInterval": 5000,
    "strangerRetryCount": 0
  },
  "detectTargetList": [
    "face"
  ],
  "captureConfigs": {
    "captureMode": 2,
    "quality": {
      "faceQuality": 0,
      "humanQuality": 0,
      "motorVehicleQuality": 0,
      "nonMotorVehicleQuality": 0,
      "plateQuality": 0
    }
  },
  "surveillanceEnabled": true,
  "videoConStatus": 1,
  "filter": {
    "roll": 180,
    "yaw": 100,
    "pitch": 100,
    "blurness": 80,
    "faceMin": 20,
    "bodyMin": 20
  },
}
```

```

    "group": [
      {
        "groupName": "cap",
        "startTime": "000000",
        "endTime": "000000",
        "searchThresholdHigh": 66,
        "searchThresholdLow": 0
      },
      {
        "groupName": "group1",
        "startTime": "000000",
        "endTime": "000000",
        "searchThresholdHigh": 66,
        "searchThresholdLow": 0
      }
    ],
    "globalStranger": {
      "enable": false,
      "searchThresholdLow": 0,
      "startTime": "000000",
      "endTime": "000000"
    }
  }
]

```

20.2 获取全量抓拍事件列表 GET/captures

操作类型:

GET

功能：

获取抓拍队列中的全部抓拍事件信息。

请求参数：

参数	类型	说明
pageOffset	integer	列表结果偏移量，从 0 开始，与 pageToken 不可一起使用
pageToken	integer	指示列表的起始位置，为空表未从 0 开始，与 pageOffset 不可一起使用，此值首次查询时不用填写，默认从 0 开始，后续填入返回的 nextPageToken 值
pageSize	integer	返回结果一页的最大数量，与 pageToken 配合使用，如反回结果的总量超过 pageSize，在返回参数中会有 nextPageToken 字段用于指示下一页起始位置
chnNo	integer	通道号 混合通道模式（1~16，17~48）全结构化模式（1~16）
timeStart	integer	开始时间 Unix 时间戳 单位为秒
timeEnd	integer	结束时间 Unix 时间戳 单位为秒

20.2.1 获取全量人脸抓拍事件列表 GET/captures/face

接口 URI：

/captures/face

请求参数：

见上文

返回参数：

参数	类型	说明
nextPageToken	integer	下一页的起始地址，放入下一次请求的 pageToken
totalCaptures	integer	现有总的抓拍数量
captures[]	array	抓拍事件消息列表，按时间排序
{}	obj	对象内放一条消息，如有多条消息将重复
—videoNo	integer	通道号 [1 16]
—channelName	string	摄像头布点名称
—trackId	integer	目标编号
—trackIdTimes	integer	目标编号出现的次数
—imageURI	string	捕获目标图片资源 URI
—fullImageURI	string	捕获背景图片资源 URI
—catchTimestamp	integer	Unix 时间戳，目标捕获照片的时间，单位为毫秒
—faceAttr {}	obj	人脸属性
— —age	integer	年龄
— —gender	integer	性别 0：未知 1：男 2：女
— —skinColor	integer	肤色分类

参数	类型	说明
		0: 无法判断 1: 黄皮肤 2: 白皮肤 3: 黑皮肤 4: 棕色皮肤
— —hair	integer	发型 0: 发型未知 1: 光头 2: 少量头发（包含秃顶）3: 短发 4: 长发 5: 其他发型
— —beard	integer	胡子类型 0: 胡子类型未知 1: 无胡子或者胡子不明显 2: 小胡子，嘴唇上面的胡子 3: 腮须 4: 其他类型
— —hat {}	obj	帽子属性
— — —hatWear	integer	0: 是否戴帽子未知 1: 戴帽子 2: 不戴帽子
— — —hatColor	integer	16: 帽子颜色未知 17: 白色 18: 其他
— — —hatType	integer	48: 安全帽 49: 厨师帽 50: 学生帽 51: 头盔 52: 安防全人种 01 小白帽 53: 头巾 54: 其他帽子 55: 未知类型
— —respirator {}	obj	口罩属性
— — — respiratorWear	integer	0: 是否戴口罩未知 1: 戴口罩 2: 不戴口罩
— — — respiratorType	integer	16: 医用口罩 17: 雾霾口罩 18: 普通口罩 19: 未知类型 20: 厨房透明口罩
— —glasses {}	obj	眼镜属性
— — — glassLeftWear	integer	0: 左眼是否戴眼镜未知 1: 左眼戴眼镜 2: 左眼不戴眼镜

参数	类型	说明
— — — glassLeftType	integer	32: 左眼眼镜类型未知 33: 左眼深色框透明眼镜 34: 左眼普通透明眼镜 35: 左眼太阳墨镜 36: 左眼其他
— — — glassRightWear	integer	16: 右眼是否戴眼镜未知 17: 右眼戴眼镜 18: 右眼不戴眼镜
— — — glassRightType	integer	48: 右眼眼镜类型未知 49: 右眼深色框透明眼镜 50: 右眼普通透明眼镜 51: 右眼太阳墨镜 52: 右眼其他

20.2.2 获取全量人体抓拍事件列表 GET/captures/human

接口 URI:

/captures/human

请求参数:

见上文

返回参数:

参数	类型	说明
nextPageToken	integer	下一页的起始地址，放入下一次请求的 pageToken
totalCaptures	integer	现有总的抓拍数量
captures[]	array	抓拍事件消息列表，按时间排序
{}	obj	对像内放一条消息，如有多条消息将重复

参数	类型	说明
—videoNo	integer	通道号 [1 16]
—channelName	string	摄像头布点名称
—trackId	integer	目标编号
—trackIdTimes	integer	目标编号出现的次数标记
—imageURI	string	捕获目标图片资源 URI
—fullImageURI	string	捕获背景图片资源 URI
—catchTimestamp	integer	Unix 时间戳，目标捕获照片的时间，单位为毫秒
—humanAttr {}	obj	人员属性
— —clothes {}	obj	上衣属性
— — —clothesColor	integer	0: 上衣颜色未知 1: 黑色上衣 2: 白色上衣 3: 灰色上衣 4: 红色上衣 5: 橙色上衣 6: 黄色上衣 7: 绿色上衣 8: 深蓝色上衣 9: 浅蓝色上衣 10: 紫色上衣 11: 粉红色上衣 12: 棕色上衣 13: 彩色上衣
— — —clothesSleeve	integer	32: 上衣长短袖未知 33: 上衣长袖 34: 上衣短袖
— —pants {}	obj	下衣属性
— — —pantsColor	integer	0: 颜色未知 1: 黑色裤子 2: 白色裤子 3: 灰色裤子 4: 红色裤子 5: 橙色裤子 6: 黄色裤子

参数	类型	说明
		7: 绿色 裤子 8: 深蓝色裤子 9: 浅蓝色裤子 10: 紫色裤子 11: 粉红色裤子 12: 棕色裤子 13: 彩色裤子
— — —pantsLen	integer	32: 长短裤未知 33: 长裤 34: 短裤
— — —pantsSkirt	integer	48: 穿裙子未知 49: 穿裙子 50: 不穿裙子
— —bags {}	obj	背包状态属性
— — —bagCarry	integer	0: 是否背包未知 1: 背包 2: 不背包
— — —bagType	integer	按位使能，可同时使能多位 0-不使能 1-使能 第 0 位: 包类型未知 第 1 位: 双肩背包 第 2 位: 单肩包、斜挎包 第 3 位: 手提包 第 4 位: 拉杆箱
— —rideBike (废弃)	integer	骑车状态 0: 是否骑车未知 1: 骑车 2: 不骑车
— —umbrella	integer	打伞状态 0: 是否打伞未知 1: 打伞 2: 不打伞

参数	类型	说明
— —gender	integer	性别 0: 性别未知 1: 男性 2: 女性
— —hair	integer	发型 0: 发型未知 1: 光头（预留字段） 2: 很少头发（包含秃顶）（预留字段） 3: 短头发 4: 长头发 5: 其他发型
— —hat {}	obj	帽子属性
— — —hatWear	integer	0: 是否带帽子未知 1: 带帽子 2: 不戴帽子
— — —hatColor（预留 字段）	integer	16: 帽子颜色未知 17: 白色 18: 其他
— — —hatType	integer	48: 安全帽（预留字段） 49: 厨师帽（预留字段） 50: 学生帽（预留字 段） 51: 头盔 52: 安防全人种 01 小白帽（预留字段） 53: 头巾（预留字 段） 54: 其他帽子 55: 未知类型
— —safetyHelmet {}	obj	安全帽
— — — safetyHelmetWear	integer	0: 是否带帽子未知 1: 带帽子 2: 不戴帽子
— — — safetyHelmetColor（预留 字段）	integer	16: 帽子颜色未知 17: 白色 18: 其他
— — — safetyHelmetType	integer	48: 安全帽 49: 厨师帽（预留字段） 50: 学生帽（预留字段） 51: 头盔 （预留字段） 52: 安防全人种 01 小白帽（预留字段） 53: 头巾（预留字 段） 54: 其他帽子（预留字段） 55: 未知类型（预留字段）

参数	类型	说明
— —action{}（预留字段）	obj	动作属性
— — —actionTumble	integer	0：是否跌倒未知 1：跌倒 2：没有跌倒
— — —actionSmoke	integer	16：是否吸烟未知 17：吸烟 18：没有吸烟
— — —actionRun	integer	32：是否跑步未知 33：跑步 34：没有跑步
— — —actionCalling	integer	48：是否打电话未知 49：打电话 50：没有打电话
— — — actionWatchPhone	integer	64：是否看手机未知 65：看手机 66：没有看手机

20.2.3 获取全量非机动车抓拍事件列表 GET/captures/cycle

接口 URI：

/captures/cycle

请求参数：

见上文

返回参数：

参数	类型	说明
nextPageToken	integer	下一页的起始地址，放入下一次请求的 pageToken
totalCaptures	integer	现有总的抓拍数量

参数	类型	说明
<code>captures[]</code>	array	抓拍事件消息列表，按时间排序
<code>{}</code>	obj	对像内放一条消息，如有多条消息将重复
<code> —videoNo</code>	integer	通道号 [1 16]
<code> —channelName</code>	string	摄像头布点名称
<code> —trackId</code>	integer	目标编号
<code> —trackIdTimes</code>	integer	目标编号出现的次数
<code> —imageURI</code>	string	捕获目标图片资源 URI
<code> —fullImageURI</code>	string	捕获背景图片资源 URI
<code> —catchTimestamp</code>	integer	Unix 时间戳，目标捕获照片的时间，单位为毫秒
<code> —cycleAttr{}</code>	obj	非机动车属性
<code> — —subClass</code>	integer	非机动车细分类 0：未知 1：自行车 2：电动车 3：摩托车 4：三轮车 5：其他

20.2.4 获取全量机动车抓拍事件列表 GET/captures/vehicle

接口 URI:

/captures/vehicle

返回参数:

参数	类型	说明
nextPageToken	integer	下一页的起始地址，放入下一次请求的 pageToken
totalCaptures	integer	现有总的抓拍数量
captures[]	array	抓拍事件消息列表，按时间排序
{}	obj	对像内放一条消息，如有多条消息将重复
—videoNo	integer	通道号 [1 16]
—channelName	string	摄像头布点名称
—trackId	integer	目标编号
—trackIdTimes	integer	目标编号出现的次数标记
—imageURI	string	捕获目标图片资源 URI
—fullImageURI	string	捕获背景图片资源 URI
—catchTimestamp	integer	Unix 时间戳，目标捕获照片的时间，单位为毫秒
—vehicleAttr{}	obj	机动车属性
— —subClass	integer	机动车细分类 0: 未知 1: 小轿车 2: SUV 3: 面包车 4: 中巴车和大巴车 5: 皮卡车 6: 卡车 7: 其他
— —brand	integer	[0~2461] 品牌索引编号 如 0: 含义是 “保时捷_Cayenne_06 年款”

参数	类型	说明
— —color	integer	8191: 未知 8192: 黑色 8193: 白色 8194: 深灰 8195: 棕色 8196: 红色 8197: 蓝色 8198: 黄色 8199: 绿色 8200: 花色 8201: 银灰 8202: 橙色 8203: 青色 8204: 粉色 8205: 紫色 8206: 香槟
— —orientation	integer	0x2020: 正向 0x2021: 背向 0x2022: 其他
— —brandname	string	机动车属性信息, 如: "保时捷_Cayenne_06 年款"

20.2.5 获取全量车牌抓拍事件列表 GET/captures/plate

接口 URI:

/captures/plate

请求参数:

见上文

返回参数:

参数	类型	说明
nextPageToken	integer	下一页的起始地址, 放入下一次请求的 pageToken
totalCaptures	integer	现有总的抓拍数量
captures[]	array	抓拍事件消息列表, 按时间排序
{}	obj	对像内放一条消息, 如有多条消息将重复

参数	类型	说明
—videoNo	integer	通道号 [1 16]
—channelName	string	摄像头布点名称
—trackId	integer	目标编号
—trackIdTimes	integer	目标编号出现的次数标记
—imageURI	string	捕获目标图片资源 URI
—fullImageURI	string	捕获背景图片资源 URI
—catchTimestamp	integer	Unix 时间戳，目标捕获照片的时间，单位为毫秒
—plateAttr {}	obj	车牌属性
— —type	integer	车牌类型 0：单行 1：双行 2：未知
— —color	integer	车牌颜色 0：蓝色 1：黄色 2：黑色 3：白色 4：绿色 5：小型新能源 6：大型新能源 7：缺省
— —num	string	车牌号码

20.3 获取指定目标的绑定关系 POST/bindRelation

操作类型：

POST

功能:

获取指定目标的绑定关系

请求参数:

参数	类型	说明
mainObject	integer	FACE = 1, HUMAN = 2, VEHICLE = 3, CYCLE = 4, PLATE = 5
videoNo	integer	通道号 [1 16]
trackId	integer	目标编号
trackIdTimes	integer	目标编号出现的次数标记, 非必选, 缺省相当于次数序号为 1

20.3.1 获取人脸的绑定关系 POST/bindRelation

接口 URI:

/bindRelation

请求参数:

见上文

返回参数:

参数	类型	说明
bindHuman {}	obj	人脸绑定的人体信息
—imageURI	string	捕获目标图片资源 URI
—humanAttr {}	obj	人员属性
— —clothes {}	obj	上衣属性
— — —clothesColor	integer	0: 上衣颜色未知 1: 黑色上衣 2: 白色上衣 3: 灰色上衣 4: 红色上衣 5: 橙色上衣 6: 黄色上衣 7: 绿色上衣 8: 深蓝色上衣 9: 浅蓝色上衣 10: 紫色上衣 11: 粉红色上衣 12: 棕色上衣 13: 彩色上衣
— — —clothesSleeve	integer	32: 上衣长短袖未知 33: 上衣长袖 34: 上衣短袖
— —pants {}	obj	下衣属性
— — —pantsColor	integer	0: 颜色未知 1: 黑色裤子 2: 白色裤子 3: 灰色裤子 4: 红色裤子 5: 橙色裤子 6: 黄色裤子 7: 绿色 裤子 8: 深蓝色裤子 9: 浅蓝色裤子 10: 紫色裤子 11: 粉红色裤子 12: 棕色裤子 13: 彩色裤子
— — —pantsLen	integer	32: 长短裤未知 33: 长裤 34: 短裤
— — —pantsSkirt	integer	48: 穿裙子未知 49: 穿裙子 50: 不穿裙子
— —bags {}	obj	背包状态属性

参数	类型	说明
— — —bagCarry	integer	0: 是否背包未知 1: 背包 2: 不背包
— — —bagType	integer	按位使能，可同时使能多位 0-不使能 1-使能 第 0 位：包类型未知 第 1 位：双肩背包 第 2 位：单肩包、斜挎包 第 3 位：手提包 第 4 位：拉杆箱
— —rideBike（废弃）	integer	骑车状态 0: 是否骑车未知 1: 骑车 2: 不骑车
— —umbrella	integer	打伞状态 0: 是否打伞未知 1: 打伞 2: 不打伞
— —gender	integer	性别 0: 性别未知 1: 男性 2: 女性
— —hair	integer	发型 0: 发型未知 1: 光头（预留字段） 2: 很少头发（包含秃顶）（预留字段） 3: 短头发 4: 长头发 5: 其他发型

参数	类型	说明
— —hat {}	obj	帽子属性
— — —hatWear	integer	0: 是否带帽子未知 1: 带帽子 2: 不戴帽子
— — —hatColor (预留字段)	integer	16: 帽子颜色未知 17: 白色 18: 其他
— — —hatType	integer	48: 安全帽 (预留字段) 49: 厨师帽 (预留字段) 50: 学生帽 (预留字段) 51: 头盔 52: 安防全人种 01 小白帽 (预留字段) 53: 头巾 (预留字段) 54: 其他帽子 55: 未知类型
— —safetyHelmet {}	obj	安全帽
— — —safetyHelmetWear	integer	0: 是否带帽子未知 1: 带帽子 2: 不戴帽子
— — —safetyHelmetColor (预留字段)	integer	16: 帽子颜色未知 17: 白色 18: 其他
— — —safetyHelmetType	integer	48: 安全帽 49: 厨师帽 (预留字段) 50: 学生帽 (预留字段) 51: 头盔 (预留字段) 52: 安防全人种 01 小白帽 (预留字段) 53: 头巾 (预留字段) 54: 其他帽子 (预留字段) 55: 未知类型 (预留字段)
— —action {} (预留字段)	obj	动作属性
— — —actionTumble	integer	0: 是否跌倒未知 1: 跌倒 2: 没有跌倒
— — —actionSmoke	integer	16: 是否吸烟未知 17: 吸烟 18: 没有吸烟
— — —actionRun	integer	32: 是否跑步未知 33: 跑步 34: 没有跑步

参数	类型	说明
— — —actionCalling	integer	48: 是否打电话未知 49: 打电话 50: 没有打电话
— — — actionWatchPhone	integer	64: 是否看手机未知
bindCycle {}	obj	人脸绑定的非机动车信息
—imageURI	string	捕获目标图片资源 URI
—cycleAttr {}	obj	非机动车属性
— —subClass	integer	非机动车细分类 0: 未知 1: 自行车 2: 电动车 3: 摩托车 4: 三轮车 5: 其他

20.3.2 获取人体的绑定关系 POST/bindRelation

接口 URI:

/bindRelation

请求参数:

见上文

返回参数:

参数	类型	说明
bindFace {}	obj	人体绑定的人脸信息

参数	类型	说明
—imageURI	string	捕获目标图片资源 URI
—faceAttr {}	obj	人脸属性
— —age	integer	年龄
— —gender	integer	性别 0: 未知 1: 男 2: 女
— —skinColor	integer	肤色分类 0: 无法判断 1: 黄皮肤 2: 白皮肤 3: 黑皮肤 4: 棕色皮肤
— —hair	integer	发型 0: 发型未知 1: 光头 2: 少量头发（包含秃顶）3: 短发 4: 长发 5: 其他发型
— —beard	integer	胡子类型 0: 胡子类型未知 1: 无胡子或者胡子不明显 2: 小胡子，嘴唇上面的胡子 3: 腮须 4: 其他类型
— —hat {}	obj	帽子属性
— — —hatWear	integer	0: 是否戴帽子未知 1: 戴帽子 2: 不戴帽子
— — —hatColor	integer	16: 帽子颜色未知 17: 白色 18: 其他
— — —hatType	integer	48: 安全帽 49: 厨师帽 50: 学生帽 51: 头盔 52: 安防全人种 01 小白帽 53: 头巾 54: 其他帽子 55: 未知类型
— —respirator {}	obj	口罩属性

参数	类型	说明
— — — respiratorWear	integer	0: 是否戴口罩未知 1: 戴口罩 2: 不戴口罩
— — — respiratorType	integer	16: 医用口罩 17: 雾霾口罩 18: 普通口罩 19: 未知类型 20: 厨房透明口罩
— —glasses{}	obj	眼镜属性
— — — glassLeftWear	integer	0: 左眼是否戴眼镜未知 1: 左眼戴眼镜 2: 左眼不戴眼镜
— — — glassLeftType	integer	32: 左眼眼镜类型未知 33: 左眼深色框透明眼镜 34: 左眼普通透明眼镜 35: 左眼太阳墨镜 36: 左眼其他
— — — glassRightWear	integer	16: 右眼是否戴眼镜未知 17: 右眼戴眼镜 18: 右眼不戴眼镜
— — — glassRightType	integer	48: 右眼眼镜类型未知 49: 右眼深色框透明眼镜 50: 右眼普通透明眼镜 51: 右眼太阳墨镜 52: 右眼其他
bindCycle{}	obj	人体绑定的非机动车信息
—imageURI	string	捕获目标图片资源 URI
—cycleAttr{}	obj	非机动车属性
— —subClass	integer	非机动车细分类 0: 未知 1: 自行车 2: 电动车 3: 摩托车 4: 三轮车 5: 其他

20.3.3 获取非机动车的绑定关系 POST/bindRelation

接口 URI:

/bindRelation

请求参数:

见上文

返回参数:

参数	类型	说明
bindArray[]	array	非机动车绑定的绑定关系
—bindHuman {}	obj	人脸绑定的人体信息
— —imageURI	string	捕获目标图片资源 URI
— —humanAttr {}	obj	人员属性
— — —clothes {}	obj	上衣属性
— — — —clothesColor	integer	0: 上衣颜色未知 1: 黑色上衣 2: 白色上衣 3: 灰色上衣 4: 红色上衣 5: 橙色上衣 6: 黄色上衣 7: 绿色上衣 8: 深蓝色上衣 9: 浅蓝色上衣 10: 紫色上衣 11: 粉红色上衣 12: 棕色上衣 13: 彩色上衣
— — — —clothesSleeve	integer	32: 上衣长短袖未知 33: 上衣长袖 34: 上衣短袖
— — —pants {}	obj	下衣属性

参数	类型	说明
— — — —pantsColor	integer	0: 颜色未知 1: 黑色裤子 2: 白色裤子 3: 灰色裤子 4: 红色裤子 5: 橙色裤子 6: 黄色裤子 7: 绿色 裤子 8: 深蓝色裤子 9: 浅蓝色裤子 10: 紫色裤子 11: 粉红色裤子 12: 棕色裤子 13: 彩色裤子
— — — —pantsLen	integer	32: 长短裤未知 33: 长裤 34: 短裤
— — — —pantsSkirt	integer	48: 穿裙子未知 49: 穿裙子 50: 不穿裙子
— — —bags{}	obj	背包状态属性
— — — —bagCarry	integer	0: 是否背包未知 1: 背包 2: 不背包
— — — —bagType	integer	按位使能，可同时使能多位 0-不使能 1-使能 第 0 位: 包类型未知 第 1 位: 双肩背包 第 2 位: 单肩包、斜挎包 第 3 位: 手提包 第 4 位: 拉杆箱
— — —rideBike (废弃)	integer	骑车状态 0: 是否骑车未知 1: 骑车 2: 不骑车

参数	类型	说明
— — —umbrella	integer	打伞状态 0: 是否打伞未知 1: 打伞 2: 不打伞
— — —gender	integer	性别 0: 性别未知 1: 男性 2: 女性
— — —hair	integer	发型 0: 发型未知 1: 光头（预留字段） 2: 很少头发（包含秃顶）（预留字段） 3: 短头发 4: 长头发 5: 其他发型
— — —hat {}	obj	帽子属性
— — — —hatWear	integer	0: 是否带帽子未知 1: 带帽子 2: 不戴帽子
— — — —hatColor（预留字段）	integer	16: 帽子颜色未知 17: 白色 18: 其他
— — — —hatType	integer	48: 安全帽（预留字段） 49: 厨师帽（预留字段） 50: 学生帽（预留字段） 51: 头盔 52: 安防全人种 01 小白帽（预留字段） 53: 头巾（预留字段） 54: 其他帽子 55: 未知类型
— — —safetyHelmet {}	obj	安全帽
— — — —safetyHelmetWear	integer	0: 是否带帽子未知 1: 带帽子 2: 不戴帽子
— — — —safetyHelmetColor（预留字段）	integer	16: 帽子颜色未知 17: 白色 18: 其他

参数	类型	说明
— — — — safetyHelmetType	integer	48: 安全帽 49: 厨师帽（预留字段） 50: 学生帽（预留字段） 51: 头盔（预留字段） 52: 安防全人种 01 小白帽（预留字段） 53: 头巾（预留字段） 54: 其他帽子（预留字段） 55: 未知类型（预留字段）
— — —action{}（预留字段）	obj	动作属性
— — — —actionTumble	integer	0: 是否跌倒未知 1: 跌倒 2: 没有跌倒
— — — —actionSmoke	integer	16: 是否吸烟未知 17: 吸烟 18: 没有吸烟
— — — —actionRun	integer	32: 是否跑步未知 33: 跑步 34: 没有跑步
— — — —actionCalling	integer	48: 是否打电话未知 49: 打电话 50: 没有打电话
— — — — actionWatchPhone	integer	64: 是否看手机未知
—bindFace{}	obj	人体绑定的人脸信息
— —imageURI	string	捕获目标图片资源 URI
— —faceAttr{}	obj	人脸属性
— — —age	integer	年龄
— — —gender	integer	性别 0: 未知 1: 男 2: 女
— — —skinColor	integer	肤色分类 0: 无法判断 1: 黄皮肤 2: 白皮肤 3: 黑皮肤 4: 棕色皮肤

参数	类型	说明
— — —hair	integer	发型 0: 发型未知 1: 光头 2: 少量头发（包含秃顶）3: 短发 4: 长发 5: 其他发型
— — —beard	integer	胡子类型 0: 胡子类型未知 1: 无胡子或者胡子不明显 2: 小胡子，嘴唇上面的胡子 3: 腮须 4: 其他类型
— — —hat {}	obj	帽子属性
— — — —hatWear	integer	0: 是否戴帽子未知 1: 戴帽子 2: 不戴帽子
— — — —hatColor	integer	16: 帽子颜色未知 17: 白色 18: 其他
— — — —hatType	integer	48: 安全帽 49: 厨师帽 50: 学生帽 51: 头盔 52: 安防全人种 01 小白帽 53: 头巾 54: 其他帽子 55: 未知类型
— — —respirator {}	obj	口罩属性
— — — —respiratorWear	integer	0: 是否戴口罩未知 1: 戴口罩 2: 不戴口罩
— — — —respiratorType	integer	16: 医用口罩 17: 雾霾口罩 18: 普通口罩 19: 未知类型 20: 厨房透明口罩
— — —glasses {}	obj	眼镜属性
— — — —glassLeftWear	integer	0: 左眼是否戴眼镜未知 1: 左眼戴眼镜 2: 左眼不戴眼镜
— — — —glassLeftType	integer	32: 左眼眼镜类型未知 33: 左眼深色框透明眼镜 34: 左眼普通透明眼镜 35: 左眼太阳墨镜 36: 左眼其他

参数	类型	说明
— — — — glassRightWear	integer	16: 右眼是否戴眼镜未知 17: 右眼戴眼镜 18: 右眼不戴眼镜
— — — — glassRightType	integer	48: 右眼眼镜类型未知 49: 右眼深色框透明眼镜 50: 右眼普通透明眼镜 51: 右眼太阳墨镜 52: 右眼其他

20.3.4 获取机动车的绑定关系 POST/bindRelation

接口 URI:

/bindRelation

请求参数:

见上文

返回参数:

参数	类型	说明
bindPlate{}	obj	机动车绑定的车牌信息
—imageURI	string	捕获目标图片资源 URI
plateAttr{}	obj	车牌属性
— —type	integer	车牌类型 0: 单行 1: 双行 2: 未知

参数	类型	说明
bindPlate{}	obj	机动车绑定的车牌信息
— —color	integer	车牌颜色 0: 蓝色 1: 黄色 2: 黑色 3: 白色 4: 绿色 5: 小型新能源 6: 大型新能源 7: 缺省
— —num	string	车牌号码

20.3.5 获取车牌的绑定关系 POST/bindRelation

接口 URI:

/bindRelation

请求参数:

见上文

返回参数:

参数	类型	说明
bindVehicle{}	obj	车牌绑定的机动车信息
—imageURI	string	捕获目标图片资源 URI
—vehicleAttr{}	obj	机动车属性

参数	类型	说明
bindVehicle{}	obj	车牌绑定的机动车信息
— —subClass	integer	机动车细分类 0: 未知 1: 小轿车 2: SUV 3: 面包车 4: 中巴车和大巴车 5: 皮卡车 6: 卡车 7: 其他
— —brand	integer	[0~2461] 品牌索引编号 如 0: 含义是 “保时捷_Cayenne_06 年款”
— —color	integer	0x1FFF: 未知 0x2000: 黑色 0x2001: 白色 0x2002: 深灰 0x2003: 棕色 0x2004: 红色 0x2005: 蓝色 0x2006: 黄色 0x2007: 绿色 0x2008: 花色 0x2009: 银灰 0x200A: 橙色 0x200B: 青色 0x200C: 粉色 0x200D: 紫色 0x200E: 香槟
— —orientation	integer	0x2020: 正向 0x2021: 背向 0x2022: 其他
— —brandname	string	机动车属性信息, 如: “保时捷_Cayenne_06 年款”

21 日志模块 API

21.1 查询日志

操作类型:

GET

功能:

查询系统保存的操作日志

接口 URI:

/log

请求参数:

参数	类型	说明
type	integer	日志类型 0:全部 1:用户操作 2:异常 3:系统 4:自定义
timeStart	integer	开始时间 Unix 时间戳 单位为秒
timeEnd	integer	结束时间 Unix 时间戳 单位为秒
pageToken	integer	指示列表的起始位置，默认从 0 开始，后续填入返回的 nextPageToken 值
pageSize	integer	返回结果一页的最大数量，与 pageToken 配合使用，如反回结果的总量超过 pageSize，在返回参数中会有 nextPageToken 字段用于指示下一页起始位置

返回参数:

参数	类型	说明
nextPageToken	integer	下一页的起始地址，放入下一次请求的 pageToken
totalLogs	integer	总日志条数
log[]	array	日志数组

参数	类型	说明
{}	obj	
—id	integer	在日志数据库中的序号
—type	integer	日志类型 1:操作 2:报警 3:系统 4:自定义
—userIP	string	用户 IP 地址
—timestamp	integer	Unix 时间戳, 生成日志时间, 单位为秒
—description	string	日志描述, UTF-8 编码

21.2 日志导出

操作类型:

GET

功能:

查询系统保存的操作日志

接口 URI:

/logFile

请求参数:

无

返回参数:

参数	类型	说明
logURI	string	日志文件下载地址

22 onvif 搜索

22.1 获取视频通道列表 GET /videos/onvif/list

操作类型:

GET

功能:

获取 onvif 搜索出来的视频通道列表

接口 URI:

/videos/onvif/list

请求参数:

无

返回参数:

参数	类型	说明
deviceInfo[]	array	搜索到的设备信息列表
—ipv4	string	设备 ip 地址。
—uri	string	设备 uri
—uuid	string	设备 uuid

22.2 获取视频通道码流 URL POST /videos/onvif/uri

操作类型：

POST

功能：

获取 onvif 搜索出来的其中一个通道的码流 URL

接口 URI：

/videos/onvif/uri

请求参数：

参数	类型	说明
uri	string	必填，搜索返回的设备 uri
username	string	IPC 接入的用户名

参数	类型	说明
password	string	IPC 接入的密码

返回参数：

参数	类型	说明
mainUrl	string	主码流地址
subUrl	string	子码流地址

23 其它参数说明

23.1 WS 主动上报数据

设备支持主动上报数据功能，需要在相关的接口中进行设置，选择上报模式，当选择主动上报模式后，将主动上报相关的数据，主要报括类型：设备心跳数据、Alerts 报警/通行消息与 capture 抓拍人脸相关数据。

上报数据采用 Socket 接口，TCP 协议。

23.1.1 心跳数据

心跳上报是为了使服务器可以找到设备，获得设备的公网 IP 与端口。心跳参数列表：

参数	类型	说明
deviceId	string	设备的唯一编号（固定 20 位），MAC+后面补 0

参数	类型	说明
timestamp	integer	Unix 时间戳，当前时间，单位为毫秒

服务器地址示例：ws://domain(ip):port/

23.1.2 WS 主动上报 Alerts 报警/通行消息

alert 的主动上报模式被打开时，当报警或通行比对超过阈值后，设备会按设置的服务器地址主动上报消息，得到消息后，可通过 Web API 进行抓拍人脸与底库中人脸的抽取，如某条 alerts 信息上报失败或超时，需要重尝试，总次数不超设置次数，发超过就丢弃该条数据。有新的数据产后会再次进行尝试上报。如上报后服务器接收到上报数据，需要回复 dataType+videoNo+timestamp+trackId 与 ACK 信息，中间用 “_” 分隔。例：Alert_01_1543893775_6018_ACK

发送协议将采用 HEADER（16 字节定长）+JSON+PIC（二进制）的组合方式进行发送。

服务器地址示例（用 ‘:’ 分隔端口）：ws://domain(ip):port/

pingpong 机制：

websocket 连接后，B3 web server 每隔 1S 向 client 发送 ping，client 需要立即回复 pong，如果 B3 web server 在 10s 后没有收到 pong，则断开此次 websocket 连接。

HEADER 参数列表：

参数	类型	说明
TYPE	char	固定为 “Alert”，二进制表示，8 个字节长度
HEADLEN	integer	后面 JSON 的长度，二进制表示，4 个字节长度

SEQ	integer	上传数据序列，二进制表示，4 个字节长度，每次发送加 1，主要用于确认是否有丢数据包
-----	---------	--

具体参数列表（JSON）：

参数	类型	说明
deviceId	string	设备的唯一编号
videoNo	integer	视频通道号
channelName	string	摄像头布点或抓拍机布点名称，就是 videoName 或 pictureName，根据工作模式 workMode 设置所决定
chnDetectType	integer	通道检测类型：0： 未知， 1： 纯人脸 ；2： 脸人；3： 结构化
alertGroup	Array	报警人脸分组列表，分组信息，可能会有多个 groupName
—groupName	string	当 searchScore 在底库人脸所在的分组的阈值范围内，添加此人脸分组名
group	Array	分组信息，可能会有多个 groupName
—groupName	string	布点绑定的人脸分组名称，如该布点绑定了多个分组，将有多组 groupName
faceToken	string	人脸的唯一 ID，用于获取分组中人脸照片
imageId	string	原始图片唯一 ID，用于获取底库中原始照片
images {}	obj	图片数据，受 pushImageSW 开关控制，如关闭数据为空
—captureFace	integer	抓拍人脸图数据长度
—fullImage	integer	全景图数据长度

参数	类型	说明
—Top1	integer	底库分最高人脸图数据长度
trackId	integer	视频中的一个人脸或一张抓拍机图片唯一编号，用于调用 captures 接口调取抓拍图片使用
trackIdTimes	integer	上述同一 trackId 出现的次数
bindingInfo[]	array	绑定的目标信息
—bindingType	integer	绑定目标的类型：1-人脸，2-人体，3-机动车，4-非机动车，5-车牌
—bindingTrackId	integer	绑定目标的 trackid
movingStatus	integer	行进状态：0-未知，1-静止，2-运动。注：人体、机动车、非机动车目标类型有此项，其他目标类型，此项为 0
movingDirection	integer	行进方向：0-未知，1-上，2-下，3-左，4-右，5-左上，6-左下，7-右上，8-右下。注：人体、机动车、非机动车目标类型有此项，其他目标类型，此项为 0
searchScore	float	抓拍与底库中人脸比对的相似度
livenessScore	float	抓拍图片的活体分数
description	string	描述，预留字段，UTF-8 编码
timestamp	integer	Unix 时间戳，事件发生的时间，单位为毫秒
openDoorStatus	integer	报警时，开门状态。 取值如下： DEV_OPEN_DOOR_ALLOW = 0, //@允许进入 DEV_OPEN_DOOR_VISITOR_NOT_ALLOW = 1, //@此门不允许访客进入

参数	类型	说明
		DEV_OPEN_DOOR_VISITOR_TIME_NOT_ALLOW = 2, //@您的来访时间未到或已过期 DEV_OPEN_DOOR_PERMISSION_NOT_ALLOW = 3, //@门禁不开放 DEV_OPEN_DOOR_DATE_NOT_ALLOW = 4, //@此刻门禁设置为关闭 DEV_OPEN_DOOR_TIME_NOT_ALLOW = 5, //@此刻门禁设置为关闭 DEV_OPEN_DOOR_SETTING_NOT_FOUND = 6, //@设置找不到 DEV_OPEN_DOOR_UNRECOGNIZED = 7, //@未识别 DEV_OPEN_DOOR_YELLOWLIST_NOT_ALLOW = 8, //@黄名单
objType	string	目标类型：人脸：“face”
faceAttr{}	obj	人脸属性
—age	integer	年龄
—gender	integer	性别 0：男 1：女
—hair	integer	0：光头 1：少量头发（包含秃顶）2：短发 3：长发 4：未知
—beard	integer	0：无胡子或者胡子不明显 1：嘴唇上面的胡子 2：络腮胡 3：未知
—hat	integer	0：无帽子 1：安全帽 2：厨师帽 3：学生帽 4：头盔 5：01 小白帽 6：头巾 7：其他帽子 8：未知
—respirator	integer	0：无口罩 1：医用口罩 2：雾霾口罩 3：普通口罩 4：厨房用”透明口罩“ 5：未知
—glasses	integer	0：无眼镜 1：深色框透明眼镜 2：普通透明眼镜 3：墨镜 4：未知
detectionResult{}	obj	小图基于背景图的坐标
—left	integer	左边界坐标
— top	integer	上边界坐标
— right	integer	右边界坐标

参数	类型	说明
— bottom	integer	下边界坐标
targetResult {}	obj	目标基于小图的坐标
— left	integer	左边界坐标
— top	integer	上边界坐标
— right	integer	右边界坐标
— bottom	integer	下边界坐标

PIC 列表:

参数	类型	说明
captureFace	char	抓拍人脸图，二进制表示
fullImage	char	全景图，二进制表示
Top1	char	底库分最高人脸图，二进制表示

23.1.3 WS 主动上报 Capture 抓拍消息

capture 的主动上报模式被打开时，当捕获到人脸时，设备会按设置的服务器地址主动上报消息，如得发送失败或超时，最多尝试 2 次，超过 2 次将丢弃，不再重新尝试。主服务器得到消息后，可通过 Web API 进行抓拍人脸与底库中人脸的抽取，如上报后服务器接收到上报数据，需要回复 dataType+videoNo+catchTimestamp+trackId 与 ACK 信息，中间用 “_” 分隔。
例：Capture_01_1543893775_6018_ACK

发送协议将采用 HEADER（16 字节定长）+JSON+PIC（二进制）的组合方式进行发送。

服务器地址示例（用’：’分隔端口）：ws://domain(ip):port/

HEADER 参数列表：

参数	类型	说明
TYPE	char	固定为“Capture”，二进制表示，8 个字节长度
HEADLEN	integer	后面 JSON 的长度，二进制表示，4 个字节长度
SEQ	integer	上传数据序列，二进制表示，4 个字节长度，每次发送加 1，主要用于确认是否有丢数据包

具体参数列表：

参数	类型	说明
deviceId	string	设备的唯一编号
videoNo	integer	视频通道号
channelName	string	摄像头布点或抓拍机布点名称，就是 videoName，根据工作模式 workMode 设置所决定
chnDetectType	integer	通道检测类型：0： 未知， 1： 纯人脸 ，2： 脸人；3： 结构化
trackId	integer	视频中的一个人脸或一张抓拍机图片唯一编号，用于调用 captures 接口调取抓拍图片与 clips 接口调取剪辑录像使用
trackIdTimes	integer	上述同一 trackId 出现的次数

参数	类型	说明
bindingInfo[]	array	绑定的目标信息
—bindingType	integer	绑定目标的类型：1-人脸，2-人体，3-机动车，4-非机动车，5-车牌
—bindingTrackId	integer	绑定目标的 trackid
movingStatus	integer	行进状态：0-未知，1-静止，2-运动。注：人体、机动车、非机动车目标类型有此项，其他目标类型，此项为 0
movingDirection	integer	行进方向：0-未知，1-上，2-下，3-左，4-右，5-左上，6-左下，7-右上，8-右下。注：人体、机动车、非机动车目标类型有此项，其他目标类型，此项为 0
bestStranger	bool	是否为未比中的质量最好的人脸，只在目标类型为“人脸”时有效，其他目标类型下，该字段不存在
captureFace	integer	抓拍人脸图数据长度，受 pushImageSW 总开关和 pushSmallImageSW 开关一同控制，如 pushImageSW 开启+pushSmallImageSW 关闭，则关闭数据为空。
fullImage	integer	抓拍全景图数据长度，受 pushImageSW 总开关和 pushBigImageSW 开关一同控制，如 pushImageSW 开启+pushBigImageSW 开启，则上报数据。
enterTimestamp	integer	Unix 时间戳，人脸第一次捕获到的时间，单位为毫秒
catchTimestamp	integer	Unix 时间戳，人脸捕获最佳照片的时间，单位为毫秒
leaveTimestamp	integer	Unix 时间戳，人脸离开，最后一次捕获到的时间，单位为毫秒
objType	string	目标类型，可取值如下，此值决定了接下来携带的目标属性字段 (1)人脸：“face” (2)人形：“human” (3)机动车：“vehicle”， (4)非机动车：“cycle” (5)车牌：“plate”

参数	类型	说明
faceAttr {}	obj	人脸属性
—age	integer	年龄
—gender	integer	性别 0：男 1：女
—hair	integer	0：光头 1：少量头发（包含秃顶）2：短发 3：长发 4：未知
—beard	integer	0：无胡子或者胡子不明显 1：嘴唇上面的胡子 2：络腮胡 3：未知
—hat	integer	0：无帽子 1：安全帽 2：厨师帽 3：学生帽 4：头盔 5：01 小白帽 6：头巾 7：其他帽子 8：未知
—respirator	integer	0：无口罩 1：医用口罩 2：雾霾口罩 3：普通口罩 4：厨房用”透明口罩 “ 5：未知
—glasses	integer	0：无眼镜 1：深色框透明眼镜 2：普通透明眼镜 3：墨镜 4：未知
humanAttr {}	obj	人员属性
—clothes {}	obj	上衣属性
— —clothesColor	integer	0：上衣颜色未知 1：黑色上衣 2：白色上衣 3：灰色上衣 4：红色上衣 5：橙色上衣 6：黄色上衣 7：绿色上衣 8：深蓝色上衣 9：浅蓝色上衣 10：紫色上衣 11： 粉红色上衣 12：棕色上衣 13：彩色上衣
— —clothesSleeve	integer	32：上衣长短袖未知 33：上衣长袖 34：上衣短袖
—pants {}	obj	下衣属性

参数	类型	说明
— —pantsColor	integer	0: 颜色未知 1: 黑色裤子 2: 白色裤子 3: 灰色裤子 4: 红色裤子 5: 橙色裤子 6: 黄色裤子 7: 绿色 裤子 8: 深蓝色裤子 9: 浅蓝色裤子 10: 紫色裤子 11: 粉红色裤子 12: 棕色裤子 13: 彩色裤子
— —pantsLen	integer	32: 长短裤未知 33: 长裤 34: 短裤
— —pantsSkirt	integer	48: 穿裙子未知 49: 穿裙子 50: 不穿裙子
—bags{}	obj	背包属性
— —bagCarry	integer	0: 是否背包未知 1: 背包 2: 不背包
— —bagType	integer	按位使能，可同时使能多位 0-不使能 1-使能 第 0 位: 包类型未知 第 1 位: 双肩背包 第 2 位: 单肩包、斜挎包 第 3 位: 手提包 第 4 位: 拉杆箱
—rideBike (废弃)	integer	骑车状态 0: 是否骑车未知 1: 骑车 2: 不骑车

参数	类型	说明
—umbrella	integer	打伞状态 0: 是否打伞未知 1: 打伞 2: 不打伞
—gender	integer	性别 0: 性别未知 1: 男性 2: 女性
—hair	integer	发型 0: 发型未知 1: 光头（预留字段） 2: 很少头发（包含秃顶）（预留字段） 3: 短头发 4: 长头发 5: 其他发型
—hat {}	obj	帽子属性
— —hatWear	integer	0: 是否带帽子未知 1: 带帽子 2: 不戴帽子
— —hatColor（预留字段）	integer	16: 帽子颜色未知 17: 白色 18: 其他
— —hatType	integer	48: 安全帽（预留字段） 49: 厨师帽（预留字段） 50: 学生帽（预留字段） 51: 头盔 52: 安防全人种 01 小白帽（预留字段） 53: 头巾（预留字段） 54: 其他帽子 55: 未知类型
—safetyHelmet {}	obj	安全帽属性
— —safetyHelmetWear	integer	0: 是否带帽子未知 1: 带帽子 2: 不戴帽子
— —safetyHelmetColor（预留字段）	integer	16: 帽子颜色未知 17: 白色 18: 其他

参数	类型	说明
— —safetyHelmetType	integer	48: 安全帽 49: 厨师帽（预留字段） 50: 学生帽（预留字段） 51: 头盔（预留字段） 52: 安防全人种 01 小白帽（预留字段） 53: 头巾（预留字段） 54: 其他帽子（预留字段） 55: 未知类型（预留字段）
—action{}（预留字段）	obj	动作属性
— —actionTumble	integer	0: 是否跌倒未知 1: 跌倒 2: 没有跌倒
— —actionSmoke	integer	16: 是否吸烟未知 17: 吸烟 18: 没有吸烟
— —actionRun	integer	32: 是否跑步未知 33: 跑步 34: 没有跑步
— —actionCalling	integer	48: 是否打电话未知 49: 打电话 50: 没有打电话
— —actionWatchPhone	integer	64: 是否看手机未知 65: 看手机 66: 没有看手机
vehicleAttr{}	obj	机动车属性
—subClass	integer	机动车细分类 0: 未知 1: 小轿车 2: SUV 3: 面包车 4: 中巴车和大巴车 5: 皮卡车 6: 卡车 7: 其他
—brand	integer	[0~2461] 品牌索引编号 如 0: 含义是“保时捷_Cayenne_06 年款”
—color	integer	0x1FFF: 未知 0x2000: 黑色 0x2001: 白色 0x2002: 深灰 0x2003: 棕色 0x2004: 红色 0x2005: 蓝色 0x2006: 黄色 0x2007: 绿色 0x2008: 花色 0x2009: 银灰 0x200A: 橙色 0x200B: 青色 0x200C: 粉色 0x200D: 紫色 0x200E: 香槟
—orientation	integer	0x2020: 正向 0x2021: 背向 0x2022: 其他

参数	类型	说明
—brandname	string	机动车属性信息，如：“保时捷_Cayenne_06 年款”
cycleAttr {}	obj	非机动车属性
—subClass	integer	非机动车细分类 0：未知 1：自行车 2：电动车 3：摩托车 4：三轮车 5：其他
—bindingCount	integer	非机动车绑定人形的个数
plateAttr {}	obj	车牌属性
—type	integer	车牌类型 0：单行 1：双行 2：未知
—color	integer	车牌颜色 0：蓝色 1：黄色 2：黑色 3：白色 4：绿色 5：小型新能源 6：大型新能源 7：缺省
—num	string	车牌号码
detectionResult {}	obj	小图基于背景图的坐标
—left	integer	左边界坐标
— top	integer	上边界坐标
— right	integer	右边界坐标
— bottom	integer	下边界坐标

参数	类型	说明
targetResult {}	obj	目标基于小图的坐标
—left	integer	左边界坐标
— top	integer	上边界坐标
— right	integer	右边界坐标
— bottom	integer	下边界坐标

PIC 列表：

参数	类型	说明
captureFace	char	抓拍人脸图，二进制表示
fullImage	char	全景图，二进制表示

23.2 HTTP 主动上报

- 开启上报失败数据重发功能，断网状态下，B3R 间隔 30 分钟尝试重新发送失败数据，直到上传成功。

数据类型为抓拍时的上报参数：

当目标类型为人脸时：

参数	类型	说明
screen_token	string	布控通道名
photo	string	抓拍图 base64 编码，选择不返回抓拍图时没有该字段
photo_md5	string	图片 md5 值，选择不返回抓拍图时没有该字段
fullImage	string	抓拍背景大图 base64 编码，选择不返回抓拍背景大图时没有该字段
fullImage_md5	string	背景大图 md5 值，选择不返回抓拍背景大图时没有该字段
age	float	年龄
gender	float	性别
quality	float	图片质量 取值 0~1 值越大质量越大好
roll	integer	人脸旋转角度
yaw	integer	人脸水平角度
pitch	integer	人脸垂直角度
blurness	float	图像清晰度（0~100）
confidence	float	识别比分 TOP1
event_type	int	识别记录类型 • 1：未识别 • 0：识别 • 2：密码开门
fmp	float	活体分数

参数	类型	说明
fmp_error	int	非活体判断结果， • - o 1: 非活体攻击 - o 0: 正常
track_id	string	视频流区分唯一 track 用，抓拍识别唯一标识
track_die	bool	track 消失标志 true:消失 false:未消失
timestamp	integer	时间戳
video_number	integer	视频通道号
chnDetectType	integer	通道检测类型：0： 未知，1： 纯人脸 ，2：脸人；3：结构化
device_id	string	设备 ID
bestStranger	bool	是否为未比中的质量最好的人脸，只在目标类型为“人脸”时有效
bindingInfo[]	array	绑定的目标信息
—bindingType	integer	绑定目标的类型：1-人脸，2-人体，3-机动车，4-非机动车，5-车牌
—bindingTrackId	integer	绑定目标的 trackid
objType	string	目标类型，固定为：“face”
faceAttr{}	obj	人脸属性
—age	integer	年龄

参数	类型	说明
—gender	integer	性别 0: 未知 1: 男 2: 女
—skinColor	integer	肤色分类 0: 无法判断 1: 黄皮肤 2: 白皮肤 3: 黑皮肤 4: 棕色皮肤（深色皮肤）
—hair	integer	发型 0: 发型未知 1: 光头 2: 少量头发（包含秃顶）3: 短发 4: 长发 5: 其他发型
—beard	integer	胡子类型 0: 胡子类型未知 1: 无胡子或者胡子不明显 2: 小胡子，嘴唇上面的胡子 3: 络腮胡 4: 其他类型
—hat {}	obj	帽子属性
— —hatWear	integer	0: 是否戴帽子未知 1: 戴帽子 2: 不戴帽子
— —hatColor	integer	16: 帽子颜色未知 17: 白色 18: 其他
— —hatType	integer	48: 安全帽 49: 厨师帽 50: 学生帽 51: 头盔 52: 安防全人种 01 小白帽 53: 头巾 54: 其他帽子 55: 未知类型
—respirator {}	obj	口罩属性
— — respiratorWear	integer	0: 是否戴口罩未知 1: 戴口罩 2: 不戴口罩
— — respiratorType	integer	16: 医用口罩 17: 雾霾口罩 18: 普通口罩 19: 未知类型 20: 厨房透明口罩

参数	类型	说明
—glasses{}	obj	眼镜属性
— — glassLeftWear	integer	0: 左眼是否戴眼镜未知 1: 左眼戴眼镜 2: 左眼不戴眼镜
— — glassLeftType	integer	32: 左眼眼镜类型未知 33: 左眼深色框透明眼镜 34: 左眼普通透明眼镜 35: 左眼太阳墨镜 36: 左眼其他
— — glassRightWear	integer	16: 右眼是否戴眼镜未知 17: 右眼戴眼镜 18: 右眼不戴眼镜
— — glassRightType	integer	48: 右眼眼镜类型未知 49: 右眼深色框透明眼镜 50: 右眼普通透明眼镜 51: 右眼太阳墨镜 52: 右眼其他
detectionResult{}	obj	小图基于背景图的坐标
—left	integer	左边界坐标
— top	integer	上边界坐标
— right	integer	右边界坐标
— bottom	integer	下边界坐标
targetResult{}	obj	目标基于小图的坐标
—left	integer	左边界坐标
— top	integer	上边界坐标
— right	integer	右边界坐标
— bottom	integer	下边界坐标
featureInfo{}	obj	特征数据信息，只有人脸并且是识别时才会携带此字段

参数	类型	说明
—feature	string	特征数据，base64 编码
—version	string	模型版本
algType	integer	算法类型，0：未知；1：脸人/结构化；2：警戒。（可选字段，多算法上报平台时用到此字段）

当目标类型为其他（人形、机动车、非机动车、车牌）时：

参数	类型	说明
screen_token	string	布控通道名
photo	string	抓拍图 base64 编码，选择不返回抓拍图时没有该字段
photo_md5	string	图片 md5 值，选择不返回抓拍图时没有该字段
fullImage	string	抓拍背景大图 base64 编码，选择不返回抓拍背景大图时没有该字段
fullImage_md5	string	背景大图 md5 值，选择不返回抓拍背景大图时没有该字段
quality	float	图片质量
track_id	string	视频流区分唯一 track 用，抓拍识别唯一标识
track_die	bool	track 消失标志 true:消失 false:未消失
timestamp	integer	时间戳
video_number	integer	视频通道号
chnDetectType	integer	通道检测类型：0：未知， 1：纯人脸 ；2：脸人；3：结构化
device_id	string	设备 ID

参数	类型	说明
bindingInfo[]	array	绑定的目标信息
—bindingType	integer	绑定目标的类型：1-人脸，2-人体，3-机动车，4-非机动车，5-车牌
—bindingTrackId	integer	绑定目标的 trackid
movingStatus	integer	行进状态：0-未知，1-静止，2-运动。注：人体、机动车、非机动车目标类型有此项，其他目标类型，此项为 0
movingDirection	integer	行进方向：0-未知，1-上，2-下，3-左，4-右，5-左上，6-左下，7-右上，8-右下。注：人体、机动车、非机动车目标类型有此项，其他目标类型，此项为 0
objType	string	目标类型，可取值如下，此值决定了接下来携带的目标属性字段 (2) 人形：“human” (3) 机动车：“vehicle”， (4) 非机动车：“cycle” (5) 车牌：“plate”
humanAttr {}	obj	人员属性
—clothes {}	obj	上衣属性
— —clothesColor	integer	0：未知 1：黑色上衣 2：白色上衣 3：灰色上衣 4：红色上衣 5：橙色上衣 6：黄色上衣 7：绿色上衣 8：深蓝色上衣 9：浅蓝色上衣 10：紫色上衣 11：粉红色上衣 12：棕色上衣 13：彩色上衣
— —clothesSleeve	integer	32：上衣长短袖未知 33：上衣长袖 34：上衣短袖
—pants {}	obj	下衣属性

参数	类型	说明
— —pantsColor	integer	0: 未知 1: 黑色裤子 2: 白色裤子 3: 灰色裤子 4: 红色裤子 5: 橙色裤子 6: 黄色裤子 7: 绿色 裤子 8: 深蓝色裤子 9: 浅蓝色裤子 10: 紫色裤子 11: 粉红色裤子 12: 棕色裤子 13: 彩色裤子
— —pantsLen	integer	32: 长短裤未知 33: 长裤 34: 短裤
— —pantsSkirt	integer	48: 穿裙子未知 49: 穿裙子 50: 不穿裙子
—bags{}	obj	背包属性
— —bagCarry	integer	0: 未知 1: 背包 2: 不背包
— —bagType	integer	按位使能，可同时使能多位 0-不使能 1-使能 第 0 位: 包类型未知 第 1 位: 双肩背包 第 2 位: 单肩包、斜挎包 第 3 位: 手提包 第 4 位: 拉杆箱
—rideBike (废弃)	integer	骑车状态 0: 未知 1: 骑车 2: 不骑车

参数	类型	说明
—umbrella	integer	打伞状态 0: 未知 1: 打伞 2: 不打伞
—gender	integer	性别 0: 未知 1: 男性 2: 女性
—hair	integer	发型 0: 发型未知 1: 光头（预留字段） 2: 很少头发（包含秃顶）（预留字段） 3: 短头发 4: 长头发 5: 其他发型
—hat {}	obj	帽子属性
— —hatWear	integer	0: 未知 1: 带帽子 2: 不戴帽子
— —hatColor（预留字段）	integer	16: 帽子颜色未知 17: 白色 18: 其他
— —hatType	integer	48: 安全帽（预留字段） 49: 厨师帽（预留字段） 50: 学生帽（预留字段） 51: 头盔 52: 安防全人种 01 小白帽（预留字段） 53: 头巾（预留字段） 54: 其他帽子 55: 未知类型
—safetyHelmet {}	obj	安全帽属性
— —safetyHelmetWear	integer	0: 未知 1: 带帽子 2: 不戴帽子
— —safetyHelmetColor（预留字段）	integer	16: 帽子颜色未知 17: 白色 18: 其他

参数	类型	说明
— —safetyHelmetType	integer	48: 安全帽 49: 厨师帽（预留字段） 50: 学生帽（预留字段） 51: 头盔（预留字段） 52: 安防全人种 01 小白帽（预留字段） 53: 头巾（预留字段） 54: 其他帽子（预留字段） 55: 未知类型（预留字段）
—action{}（预留字段）	obj	动作属性
— —actionTumble	integer	0: 未知 1: 跌倒 2: 没有跌倒
— —actionSmoke	integer	16: 是否吸烟未知 17: 吸烟 18: 没有吸烟
— —actionRun	integer	32: 是否跑步未知 33: 跑步 34: 没有跑步
— —actionCalling	integer	48: 是否打电话未知 49: 打电话 50: 没有打电话
— —actionWatchPhone	integer	64: 是否看手机未知 65: 看手机 66: 没有看手机
vehicleAttr{}	obj	机动车属性
—subClass	integer	机动车细分类 0: 未知 1: 小轿车 2: SUV 3: 面包车 4: 中巴车和大巴车 5: 皮卡车 6: 卡车 7: 其他
—brand	integer	[0~2461] 品牌索引编号 如 0: 含义是 “保时捷_Cayenne_06 年款”
—color	integer	0x1FFF: 未知 0x2000: 黑色 0x2001: 白色 0x2002: 深灰 0x2003: 棕色 0x2004: 红色 0x2005: 蓝色 0x2006: 黄色 0x2007: 绿色 0x2008: 花色 0x2009: 银灰 0x200A: 橙色 0x200B: 青色 0x200C: 粉色 0x200D: 紫色 0x200E: 香槟
—orientation	integer	0x2020: 正向 0x2021: 背向 0x2022: 其他

参数	类型	说明
—brandname	string	机动车属性信息，如：“保时捷_Cayenne_06 年款”
cycleAttr {}	obj	非机动车属性
—subClass	integer	非机动车细分类 0：未知 1：自行车 2：电动车 3：摩托车 4：三轮车 5：其他
—bindingCount	integer	非机动车绑定人形的个数
plateAttr {}	obj	车牌属性
—type	integer	车牌类型 0：单行 1：双行 2：未知
—color	integer	车牌颜色 0：蓝色 1：黄色 2：黑色 3：白色 4：绿色 5：小型新能源 6：大型新能源 7：缺省
—num	string	车牌号码
detectionResult {}	obj	小图基于背景图的坐标
—left	integer	左边界坐标
— top	integer	上边界坐标
— right	integer	右边界坐标
— bottom	integer	下边界坐标

参数	类型	说明
targetResult {}	obj	目标基于小图的坐标
—left	integer	左边界坐标
— top	integer	上边界坐标
— right	integer	右边界坐标
— bottom	integer	下边界坐标
algType	integer	算法类型，0：未知；1：脸人/结构化；2：警戒。（可选字段，多算法上报平台时用到此字段）

数据类型为识别告警时的上报参数：

参数	类型	说明
screen_token	string	布控通道名
photo	string	抓拍图 base64 编码，选择不返回抓拍图时没有该字段
photo_md5	string	图片 md5 值，选择不返回抓拍图时没有该字段
fullImage	string	抓拍背景大图 base64 编码，选择不返回抓拍背景大图时没有该字段
fullImage_md5	string	背景大图 md5 值，选择不返回抓拍背景大图时没有该字段
age	float	年龄
gender	float	性别
quality	float	图片质量
confidence	float	识别比分 TOP1

参数	类型	说明
event_type	int	识别记录类型 • 1: 未识别 • 0: 识别 • 2: 密码开门
fmp	float	活体分数
fmp_error	int	非活体判断结果， • - o 1: 非活体攻击 - o 0: 正常
track_id	string	视频流区分唯一 track 用，抓拍识别唯一标识
track_die	bool	track 消失标志 true:消失 false:未消失
timestamp	integer	时间戳
video_number	integer	视频通道号
chnDetectType	integer	通道检测类型：0: 未知， 1: 纯人脸 ，2: 脸人；3: 结构化
device_id	string	设备 ID
access_pass	integer	1: koala 权限允许通行，0: 不允许通行
stranger_flag	integer	1: 陌生人，0: 库中人
ImageID	string	原始图片唯一 ID，用于获取底库中原始照片

参数	类型	说明
Description	string	底库图片描述
alertGroup	Array	报警人脸分组列表，分组信息，可能会有多个 groupName
—groupName	string	当 searchScore 在底库人脸所在的分组的阈值范围内，添加此人脸分组名
bindingInfo[]	array	绑定的目标信息
—bindingType	integer	绑定目标的类型：1-人脸，2-人体，3-机动车，4-非机动车，5-车牌
—bindingTrackId	integer	绑定目标的 trackid
objType	string	目标类型，固定为：“face”
faceAttr{}	obj	人脸属性
—age	integer	年龄
—gender	integer	性别 0：未知 1：男 2：女
—skinColor	integer	肤色分类 0：无法判断 1：黄皮肤 2：白皮肤 3：黑皮肤 4：棕色皮肤
—hair	integer	发型 0：发型未知 1：光头 2：少量头发（包含秃顶）3：短发 4：长发 5：其他发型
—beard	integer	胡子类型 0：胡子类型未知 1：无胡子或者胡子不明显 2：小胡子，嘴唇上面的胡子 3：腮须 4：其他类型

参数	类型	说明
—hat {}	obj	帽子属性
— —hatWear	integer	0: 是否戴帽子未知 1: 戴帽子 2: 不戴帽子
— —hatColor	integer	16: 帽子颜色未知 17: 白色 18: 其他
— —hatType	integer	48: 安全帽 49: 厨师帽 50: 学生帽 51: 头盔 52: 安防全人种 01 小白帽 53: 头巾 54: 其他帽子 55: 未知类型
—respirator {}	obj	口罩属性
— — respiratorWear	integer	0: 是否戴口罩未知 1: 戴口罩 2: 不戴口罩
— — respiratorType	integer	16: 医用口罩 17: 雾霾口罩 18: 普通口罩 19: 未知类型 20: 厨房透明口罩
—glasses {}	obj	眼镜属性
— — glassLeftWear	integer	0: 左眼是否戴眼镜未知 1: 左眼戴眼镜 2: 左眼不戴眼镜
— — glassLeftType	integer	32: 左眼眼镜类型未知 33: 左眼深色框透明眼镜 34: 左眼普通透明眼镜 35: 左 眼太阳墨镜 36: 左眼其他
— — glassRightWear	integer	16: 右眼是否戴眼镜未知 17: 右眼戴眼镜 18: 右眼不戴眼镜
— — glassRightType	integer	48: 右眼眼镜类型未知 49: 右眼深色框透明眼镜 50: 右眼普通透明眼镜 51: 右 眼太阳墨镜 52: 右眼其他
detectionResult {}	obj	小图基于背景图的坐标
—left	integer	左边界坐标

参数	类型	说明
— top	integer	上边界坐标
— right	integer	右边界坐标
— bottom	integer	下边界坐标
targetResult {}	obj	目标基于小图的坐标
—left	integer	左边界坐标
— top	integer	上边界坐标
— right	integer	右边界坐标
— bottom	integer	下边界坐标
algType	integer	算法类型，0：未知；1：脸人/结构化；2：警戒。（可选字段，多算法上报平台时用到此字段）

23.3 websocket server 给 client 推送

23.3.1 websocket server 推送 Alerts 报警/通行消息

HEADER 参数列表：

参数	类型	说明
----	----	----

TYPE	char	固定为 “Alert”，二进制表示，8 个字节长度
HEADLEN	integer	后面 JSON 的长度，二进制表示，4 个字节长度
SEQ	integer	上传数据序列，二进制表示，4 个字节长度，每次发送加 1，主要用于确认是否有丢数据包

具体参数列表（JSON）：

参数	类型	说明
deviceId	string	设备的唯一编号
dataType	string	“Alert”：报警/识别
videoNo	integer	视频通道号
channelName	string	摄像头布点或抓拍机布点名称，就是 videoName 或 pictureName，根据工作模式 workMode 设置所决定
chnDetectType	integer	通道检测类型：0： 未知， 1：纯人脸 ；2：脸人；3：结构化
aboveHigh	integer	是否为库中人： 0:否，1:是
fmpErr	integer	是否为假人：0：真人，1：假人
alertGroup	Array	报警人脸分组列表，分组信息，可能会有多个 groupName
—groupName	string	当 searchScore 在底库人脸所在的分组的阈值范围内，添加此人脸分组名
group	Array	分组信息，可能会有多个 groupName
—groupName	string	布点绑定的人脸分组名称，如该布点绑定了多个分组，将有多组 groupName

参数	类型	说明
faceToken	string	人脸的唯一 ID，用于获取分组中人脸照片
imageId	string	原始图片唯一 ID，用于获取底库中原始照片
images {}	obj	图片数据，受 pushImageSW 开关控制，如关闭数据为空
—captureFace	string	base64 的抓拍人脸图
—fullImage	string	全景图 url
—Top1	integer	底库分最高人脸图数据长度
trackId	integer	视频中的一个人脸或一张抓拍机图片唯一编号，用于调用 captures 接口调取抓拍图片使用
trackIdTimes	integer	上述同一 trackId 出现的次数
bindingInfo[]	array	绑定的目标信息
—bindingType	integer	绑定目标的类型：1-人脸，2-人体，3-机动车，4-非机动车，5-车牌
—bindingTrackId	integer	绑定目标的 trackid
movingStatus	integer	行进状态：0-未知，1-静止，2-运动。注：人体、机动车、非机动车目标类型有此项，其他目标类型，此项为 0
movingDirection	integer	行进方向：0-未知，1-上，2-下，3-左，4-右，5-左上，6-左下，7-右上，8-右下。注：人体、机动车、非机动车目标类型有此项，其他目标类型，此项为 0
searchScore	float	抓拍与底库中人脸比对的相似度
livenessScore	float	抓拍图片的活体分数
description	string	描述，预留字段，UTF-8 编码

参数	类型	说明
timestamp	integer	Unix 时间戳，事件发生的时间，单位为毫秒
openDoorStatus	integer	报警时，开门状态。 取值如下： DEV_OPEN_DOOR_ALLOW = 0, //@允许进入 DEV_OPEN_DOOR_VISITOR_NOT_ALLOW = 1, //@此门不允许访客进入 DEV_OPEN_DOOR_VISITOR_TIME_NOT_ALLOW = 2, //@您的来访时间未到或已过期 DEV_OPEN_DOOR_PERMISSION_NOT_ALLOW = 3, //@门禁不开放 DEV_OPEN_DOOR_DATE_NOT_ALLOW = 4, //@此刻门禁设置为关闭 DEV_OPEN_DOOR_TIME_NOT_ALLOW = 5, //@此刻门禁设置为关闭 DEV_OPEN_DOOR_SETTING_NOT_FOUND = 6, //@设置找不到 DEV_OPEN_DOOR_UNRECOGNIZED = 7, //@未识别 DEV_OPEN_DOOR_YELLOWLIST_NOT_ALLOW = 8, //@黄名单
objType	string	目标类型：人脸：“face”
faceAttr{}	obj	人脸属性
—age	integer	年龄
—gender	integer	性别 0：男， 1：女，2：未知
—hair	integer	0：光头 1：少量头发（包含秃顶）2：短发 3：长发 4：未知
—beard	integer	0：无胡子或者胡子不明显 1：嘴唇上面的胡子 2：络腮胡 3：未知
—hat	integer	0：无帽子 1：安全帽 2：厨师帽 3：学生帽 4：头盔 5：01 小白帽 6：头巾 7：其他帽子 8：未知
—respirator	integer	0：无口罩 1：医用口罩 2：雾霾口罩 3：普通口罩 4：透明口罩 5：未知

参数	类型	说明
—glasses	integer	0：无眼镜 1：深色框透明眼镜 2：普通透明眼镜 3：墨镜 4：未知
detectionResult {}	obj	小图基于背景图的坐标
—left	integer	左边界坐标
— top	integer	上边界坐标
— right	integer	右边界坐标
— bottom	integer	下边界坐标
targetResult {}	obj	目标基于小图的坐标
—left	integer	左边界坐标
— top	integer	上边界坐标
— right	integer	右边界坐标
— bottom	integer	下边界坐标

23.3.2 websocket server 推送 Capture 抓拍消息

HEADER 参数列表：

参数	类型	说明
TYPE	char	固定为“Capture”，二进制表示，8个字节长度
HEADLEN	integer	后面JSON的长度，二进制表示，4个字节长度
SEQ	integer	上传数据序列，二进制表示，4个字节长度，每次发送加1，主要用于确认是否有丢数据包

具体参数列表：

参数	类型	说明
deviceId	string	设备的唯一编号
dataType	string	"Capture"：抓拍
videoNo	integer	视频通道号
channelName	string	摄像头布点或抓拍机布点名称，就是 videoName，根据工作模式 workMode 设置所决定
chnDetectType	integer	通道检测类型：0： 未知，1：纯人脸；2：脸人；3：结构化
trackId	integer	视频中的一个人脸或一张抓拍机图片唯一编号，用于调用 captures 接口调取抓拍图片与 clips 接口调取剪辑录像使用
trackIdTimes	integer	上述同一 trackId 出现的次数
bindingInfo[]	array	绑定的目标信息
—bindingType	integer	绑定目标的类型：1-人脸，2-人体，3-机动车，4-非机动车，5-车牌
—bindingTrackId	integer	绑定目标的 trackid
movingStatus	integer	行进状态：0-未知，1-静止，2-运动。注：人体、机动车、非机动车目标类型有此项，其他目标类型，此项为 0
movingDirection	integer	行进方向：0-未知，1-上，2-下，3-左，4-右，5-左上，6-左下，7-右上，8-右下。注：人体、机动车、非机动车目标类型有此项，其他目标类型，此项为 0
captureFace	string	base64 的抓拍人脸图数据

参数	类型	说明
fullImage	string	抓拍全景图 url
bestStranger	bool	是否为未比中的质量最好的人脸，只在目标类型为“人脸”时有效
enterTimestamp	integer	Unix 时间戳，人脸第一次捕获到的时间，单位为毫秒
catchTimestamp	integer	Unix 时间戳，人脸捕获最佳照片的时间，单位为毫秒
leaveTimestamp	integer	Unix 时间戳，人脸离开，最后一次捕获到的时间，单位为毫秒
objType	string	目标类型，可取值如下，此值决定了接下来携带的目标属性字段 (1)人脸：“face” (2)人形：“human” (3)机动车：“vehicle”， (4)非机动车：“cycle” (5)车牌：“plate”
faceAttr{}	obj	人脸属性
—age	integer	年龄
—gender	integer	性别 0：男， 1：女，2：未知
—hair	integer	0：光头 1：少量头发（包含秃顶）2：短发 3：长发 4：未知
—beard	integer	0：无胡子或者胡子不明显 1：嘴唇上面的胡子 2：络腮胡 3：未知
—hat	integer	0：无帽子 1：安全帽 2：厨师帽 3：学生帽 4：头盔 5：01 小白帽 6：头巾 7：其他帽子 8：未知
—respirator	integer	0：无口罩 1：医用口罩 2：雾霾口罩 3：普通口罩 4：透明口罩 5：未知
—glasses	integer	0：无眼镜 1：深色框透明眼镜 2：普通透明眼镜 3：墨镜 4：未知
humanAttr{}	obj	人员属性

参数	类型	说明
—clothes{}	obj	上衣属性
— —clothesColor	integer	0: 上衣颜色未知 1: 黑色上衣 2: 白色上衣 3: 灰色上衣 4: 红色上衣 5: 橙色上衣 6: 黄色上衣 7: 绿色上衣 8: 深蓝色上衣 9: 浅蓝色上衣 10: 紫色上衣 11: 粉红色上衣 12: 棕色上衣 13: 彩色上衣
— —clothesSleeve	integer	32: 上衣长短袖未知 33: 上衣长袖 34: 上衣短袖
—pants{}	obj	下衣属性
— —pantsColor	integer	0: 颜色未知 1: 黑色裤子 2: 白色裤子 3: 灰色裤子 4: 红色裤子 5: 橙色裤子 6: 黄色裤子 7: 绿色 裤子 8: 深蓝色裤子 9: 浅蓝色裤子 10: 紫色裤子 11: 粉红色裤子 12: 棕色裤子 13: 彩色裤子
— —pantsLen	integer	32: 长短裤未知 33: 长裤 34: 短裤
— —pantsSkirt	integer	48: 穿裙子未知 49: 穿裙子 50: 不穿裙子
—bags{}	obj	背包属性
— —bagCarry	integer	0: 是否背包未知 1: 背包 2: 不背包
— —bagType	integer	按位使能，可同时使能多位 0-不使能 1-使能

参数	类型	说明
		第 0 位：包类型未知 第 1 位：双肩背包 第 2 位：单肩包、斜挎包 第 3 位：手提包 第 4 位：拉杆箱
—rideBike（废弃）	integer	骑车状态 0：是否骑车未知 1：骑车 2：不骑车
—umbrella	integer	打伞状态 0：是否打伞未知 1：打伞 2：不打伞
—gender	integer	性别 0：性别未知 1：男性 2：女性
—hair	integer	发型 0：发型未知 1：光头（预留字段） 2：很少头发（包含秃顶）（预留字段） 3：短头发 4：长头发 5：其他发型
—hat {}	obj	帽子属性
— —hatWear	integer	0：是否带帽子未知 1：带帽子 2：不戴帽子

参数	类型	说明
— —hatColor（预留字段）	integer	16: 帽子颜色未知 17: 白色 18: 其他
— —hatType	integer	48: 安全帽（预留字段） 49: 厨师帽（预留字段） 50: 学生帽（预留字段） 51: 头盔 52: 安防全人种 01 小白帽（预留字段） 53: 头巾（预留字段） 54: 其他帽子 55: 未知类型
—safetyHelmet {}	obj	安全帽属性
— —safetyHelmetWear	integer	0: 是否带帽子未知 1: 带帽子 2: 不戴帽子
— —safetyHelmetColor（预留字段）	integer	16: 帽子颜色未知 17: 白色 18: 其他
— —safetyHelmetType	integer	48: 安全帽 49: 厨师帽（预留字段） 50: 学生帽（预留字段） 51: 头盔（预留字段） 52: 安防全人种 01 小白帽（预留字段） 53: 头巾（预留字段） 54: 其他帽子（预留字段） 55: 未知类型（预留字段）
—action {}（预留字段）	obj	动作属性
— —actionTumble	integer	0: 是否跌倒未知 1: 跌倒 2: 没有跌倒
— —actionSmoke	integer	16: 是否吸烟未知 17: 吸烟 18: 没有吸烟
— —actionRun	integer	32: 是否跑步未知 33: 跑步 34: 没有跑步
— —actionCalling	integer	48: 是否打电话未知 49: 打电话 50: 没有打电话
— —actionWatchPhone	integer	64: 是否看手机未知 65: 看手机 66: 没有看手机
vehicleAttr {}	obj	机动车属性

参数	类型	说明
—subClass	integer	机动车细分类 0: 未知 1: 小轿车 2: SUV 3: 面包车 4: 中巴车和大巴车 5: 皮卡车 6: 卡车 7: 其他
—brand	integer	[0~2461] 品牌索引编号 如 0: 含义是 “保时捷_Cayenne_06 年款”
—color	integer	0x1FFF: 未知 0x2000: 黑色 0x2001: 白色 0x2002: 深灰 0x2003: 棕色 0x2004: 红色 0x2005: 蓝色 0x2006: 黄色 0x2007: 绿色 0x2008: 花色 0x2009: 银灰 0x200A: 橙色 0x200B: 青色 0x200C: 粉色 0x200D: 紫色 0x200E: 香槟
—orientation	integer	0x2020: 正向 0x2021: 背向 0x2022: 其他
—brandname	string	机动车属性信息, 如: “保时捷_Cayenne_06 年款”
cycleAttr{}	obj	非机动车属性
—subClass	integer	非机动车细分类 0: 未知 1: 自行车 2: 电动车 3: 摩托车 4: 三轮车 5: 其他
—bindingCount	integer	非机动车绑定人形的个数
plateAttr{}	obj	车牌属性
—type	integer	车牌类型 0: 单行 1: 双行 2: 未知

参数	类型	说明
—color	integer	车牌颜色 0: 蓝色 1: 黄色 2: 黑色 3: 白色 4: 绿色 5: 小型新能源 6: 大型新能源 7: 缺省
—num	string	车牌号码
detectionResult {}	obj	小图基于背景图的坐标
—left	integer	左边界坐标
— top	integer	上边界坐标
— right	integer	右边界坐标
— bottom	integer	下边界坐标
targetResult {}	obj	目标基于小图的坐标
—left	integer	左边界坐标
— top	integer	上边界坐标
— right	integer	右边界坐标
— bottom	integer	下边界坐标

23.4 唯一 ID 说明

ID	说明
faceToken	人脸入库成功后会产生一个 faceToken，由系统自动生成 24 字节，不会重复

ID	说明
imageId	当人像照片上传成功后，不轮是否入库成功，即会产生一个 imageID，由系统自动生成 24 字节，不会重复
trackId	由设备根据抓拍到的人脸，累加产生的数据。由于数据范围很大，在较长的时间内是不会重复
deviceId	设备的唯一 ID，设备出厂时设定，不可更改

23.5 错误码列表

状态码	错误码	错误消息	说明
400	104000	ILLEGAL_JSON_BODY	json 结构解析错误
400	104001	ILLEGAL_ARGUMENT:<args>	参数格式错误，<args>返回错误参数名，逗号分隔
400	104002	MISSING_ARGUMENT:<args>	缺少必要参数，<args>返回缺少参数名，逗号分隔
400	104003	SYSTEM_REBOOT	系统重启
400	104004	NO_ACCESS	无访问权限
400	104005	PASSWORD_ERROR	密码错误
400	104006	REQUEST_TIMEOUT	请求超时
400	104007	RESOURCES_NOT_FOUND	访问的资源不存在
400	104008	VIDEO_CH_OVER_LIMIT	超出限定视频路数
400	104009	NO_VIDEO	无指定视频通道，未创建或通道与名称不匹配
400	104010	VIDEO_CH_ACCESS_FAIL	视频流接入失败

状态码	错误码	错误消息	说明
400	104011	VIDEO_CH_EXIST	视频通道已存在
400	104012	PROCESS_BUSY	处理能力达到上限
400	104013	CHANNEL_BIND_RULES	视频通道绑定了权限规则
400	104014	DEVICE_ID_IS_EXIST	设备标识已存在
400	104017	VIDEO_CH_NOT_EXIST	视频通道不存在
400	104100	NOT_ACTIVATE	未激活
400	104101	ACCOUNT_BLOCKED	登录冻结
400	104200	FACE_NOT_FOUND	没有检测到人脸
400	104201	BAD_QUALITY:<reason>	人脸未通过质量判断，reason 为未通过的检测项目，如下： INVALID_RESOLUTION：无效分辨率 INVALID_TYPE：无效类型 INVALID_SIZE：无效大小 BLURNESS：太模糊 POSE：角度太大 BLACK_WHITE：黑白比例过大 FACE_TOO_SMALL：人脸太小
400	104202	FACE_TOKEN_NOT_EXIST	人脸不存在
400	104203	IMAGE_ID_NOT_EXIST	图片不存在
400	104204	FACE_ALREADY_EXIST	人脸已经存在
400	104205	SPACE_IS_FULL	批量入库返回，硬盘满 or 人脸个数超限

状态码	错误码	错误消息	说明
400	104206	IMAGE_MD5_CHECK_FAIL	图片 md5 校验失败
400	104207	IMAGE_DECODE_FAIL	图片解码失败
400	104208	IMAGE_RESIZE_FAIL	图片 resize 失败
400	104209	READ_FILE_FAIL	读取文件失败
400	104210	INVALID_RESOLUTION	无效分辨率
400	104211	INVALID_TYPE	无效类型
400	104212	INVALID_SIZE	无效大小
400	104213	BLURNESS	太模糊
400	104214	POSE	角度太大
400	104215	BLACK_WHITE	黑白比例过大
400	104216	FACE_TOO_SMALL	人脸太小
400	104300	GROUP_NAME_CONFLICT	分组名已存在或者分组名冲突
400	104301	GROUP_NAME_NOT_EXIST	分组不存在
400	104302	GROUP_OVER_LIMIT	分组数超上限
400	104303	GROUP_BINDING_NOT_EXIST	绑定关系不存在
400	104304	GROUP_BEUSED	分组正被使用，不能删除
400	104305	FEATURE_VERSION_INCONSISTENT	特征版本与系统不一致
400	104306	GROUP_SYNC	人脸库同步中，请稍后

状态码	错误码	错误消息	说明
400	104307	GROUP_IS_BUSY	人脸库入库中，请稍后
400	104308	IMAGE_OVER_LIMIT	入库图片个数超过上限
400	104309	GROUP_BIND_RULES	分组绑定了权限
400	104400	CAPTURE_DISABLE	未使能功能
400	104401	SERVER_BUSY	服务器忙
400	104402	WORK_MODE_NOT_MATCH	工作模式不匹配
400	104403	FIRMWARE_TOO_BIG	升级包文件大小太大
400	104404	FIRMWARE_RECV_ERROR	接收升级文件错误
400	104405	FIRMWARE_CHECK_ERROR	升级包格式错误
400	104406	FIRMWARE_MD5_ERROR	升级包 MD5 检查失败
400	104504	TRIGGER_RELAY_FAIL	无效继电器无法触发
400	104503	RELAY_DOES_NOT_EXIST	无继电器无法解绑
400	104502	RELAY_VIDEO_UNUSED_OR_DISABLED	对应通道未布控视频流
400	104601	CALENDAR_BIND_RULES	calendar 已经绑定了 rules
400	104602	SCHEDULE_BIND_RULES	schedule 已经绑定了 rules
400	104603	CALENDAR_NOT_EXIST	calendar 不存在
400	104604	SCHEDULE_NOT_EXIST	schedule 不存在
400	104605	RULES_NOT_EXIST	rules 不存在

状态码	错误码	错误消息	说明
400	104606	CALENDAR_TABLE_FULL	calendar 表存满了
400	104607	SCHEDULE_TABLE_FULL	schedule 表存满了
400	104608	RULES_TABLE_FULL	rules 表存满了
400	104609	CALENDAR_ALREADY_EXIST	calendar 条目已经存在了
400	104610	SCHEDULE_ALREADY_EXIST	schedule 条目已经存在了
400	104611	RULES_ALREADY_EXIST	rules 条目已经存在了
400	104612	INVALID_TIME_RANGE	无效的 timerange 参数
400	104613	INVALID_DATE_RANGE	无效的 daterange 参数
400	104614	INVALID_DATE	无效的 date 参数
400	104615	CHANNEL_NOT_EXIST	rules 试图绑定的通道并不存在
400	104616	GROUP_NOT_EXIST	rules 试图绑定的 group 并不存在
400	104800	GUI_INIT_FAILED	本地 GUI 初始化失败
400	104801	GUI_DISPLAY_CH_OVER_LIMIT	GUI 播放视频路数超限
400	104802	GUI_CONNECT_DUPLICATE_CHN	GUI 在多个窗口上请求播放同一路视频
404			请求失败，接口不存在
500	105000	INTERNAL_ERROR	其它内部错误

参数格式：

参数	类型	说明
code	integer	返回代码，成功为 0，失败会返回错误码，具体参看错误码列表
message	string	错误消息，具体参看错误码列表
timestamp	integer	Unix 时间戳，服务器收到请求时的时间戳，单位为毫秒
timeCost	integer	设备接收到请求到完成任务消耗的时间，单位为毫秒
data {}	obj	API 返回的具体参数，如没有返回参数则为空 safetyHelmetWear

返回错误码举例：

```
{  
  
  "code":14001,  
  
  "message":"ILLEGAL_ARGUMENT: searchThreshold, rotation",  
  
  "timestamp" :1532489052000,  
  
  "timeCost" :23,  
  
  "data" : {}  
}
```

24 获取图片特征信息

24.1：获取图片特征信息 POST/feature

操作类型:

POST

功能:

传入一张图片，返回图片的特征数据、特征版本、图片质量信息

接口 URI:

/feature

请求参数 (form-data):

参数	类型	说明	必选
total	text	批量抽 feature 图片数，最大 25	是
image	file	人脸图片	是
...			
image	file	人脸图片	是

返回参数:

参数	类型	说明
result_array {}	array obj	feature 结果数组
—ret_code	integer	抽 feature 结果，0: 成功；其它: 失败
—featureInfo {}	obj	特征数据信息

参数	类型	说明
— —feature	string	特征数据，base64 编码
— —version	string	模型版本
—detectionResult {}	float	人脸图片检测结果
— —quality	float	图片质量 取值 0~1 值越大质量越大好
— —roll	integer	人脸旋转角度 -180~180
— —yaw	integer	人脸水平角度 -100~100
— —pitch	integer	人脸垂直角度 -100~100
— —blurness	float	图像清晰度（0~100）值越小越清晰
— —age	integer	年龄

25 警戒模式操作 API

25.1 警戒视频资源操作 API

25.1.1 布控 POST/ alertVideos / {videoNo} / surveillances

操作类型：

POST

功能:

对警戒视频通道进行布控，设置警戒规则等

接口 URI:

/surveillances

请求参数:

参数	类型	说明	是否必选
surveillanceEnabled	bool	布控使能开关	是
algType	integer	算法类型（插件预留）	否，默认10（警戒算法）
alertRules	obj	警戒规则	否
—enable	bool	警戒使能 true 表示使能，false 表示禁能	是
—alertType	integer	警戒类型, 取值[1-7]表示： 1:人员越界 2:人员入侵 3:车辆越界 4:车辆禁停 5:车辆离开 6:人员徘徊 7:人员翻墙 8:摔倒 9:抽烟 10:打电话 11:玩手机 12:奔跑 13:睡岗	是

参数	类型	说明	是否 必选
—alertSubType	integer	警戒子类型，,取值[1-6],表示：1:单绊线检测 2:进入区域 3:离开区域 4:进出区域 5:区域停留 6:翻墙检测 根据警戒类型的不同，可取不同值： alertType-人员越界： 1:单绊线检测 alertType-人员翻墙 6:翻墙检测 alertType-人员入侵： 2:进入区域 3:离开区域 4:进出区域 alertType-车辆越界： 1:单绊线检测 alertType-车辆离开： 3:离开区域 alertType-车辆禁停、人员徘徊、摔倒、抽烟、打电话、玩手机、奔跑、睡岗 5:区域停留	是
—areaType	integer	警戒区域类型,取值[1-5],表示：1:单向单绊线 2:双向单绊线 3:单向双绊线	是

参数	类型	说明	是否 必选
		4:双向双绊线 5:区域 根据警戒子类型不同，可取不同值： alertSubType-翻墙检测： 4:双向双绊线 alertSubType-单绊线检测： 1:单向单绊线 2:双向单绊线 alertSubType-进入区域： 5:区域 alertSubType-离开区域： 5:区域 alertSubType-进出区域： 5:区域 alertSubType-区域停留： 5:区域	

参数	类型	说明	是否 必选
— personPreOverWall	bool	是否开启人员翻墙预警 只有当 alertSubType 为 6:翻墙检测 时有效	否
— detectTargetList[]	array	检测目标数组，至少包含如下二类目标中的一种 在 alertType 为车辆越界 车辆禁停 车辆离开时有效	否
— —targetName	string	检测目标名称，可取值如下，默认全选： (1)机动车：“motorVehicle”， (2)非机动车：“nonMotorVehicle” 例：detectTargetList: ["motorVehicle", "nonMotorVehicle"]	是
—targetRange	array	目标大小，有效值为[1,100]，默认值[1,50]	是
—timeRange	array	布控时段, [0,86400]，（0 秒到 24*3600 秒）	是
—stayDuration	integer	滞留时长，0~120 秒，默认 5 秒 在警戒类型与警戒子类型取以下值时有效 人员入侵-进入区域：0~120 秒 车辆禁停、人员徘徊-区域停留：0~1800 秒 摔倒、抽烟、打电话、玩手机、奔跑、睡岗-区域停留：0~120 秒	否
—cooldownDuration	integer	报警间隔，1~60 秒 默认 1 秒	是
—alertArea	array	警戒区域 根据 areaType 的不同定义不同，绊线最多两条；	是

参数	类型	说明	是否 必选
		区域最多四个；每条绊线 2~5 个点，每个区域 3~5 个点； 带方向箭头的线，将坐标按画线顺序放在数组中 例 1：单向/双向 单绊线/一个区域 有 5 个点 "alertArea":[[{"x":0,"y":0}, {"x":1,"y":1}, {"x":2,"y":2}, {"x":3,"y":3}, {"x":4,"y":4}]] 例：2 条绊线，每条绊线各 5 个点 "alertArea":[[{"x":0,"y":0}, {"x":1,"y":1}, {"x":2,"y":2}, {"x":3,"y":3}, {"x":4,"y":4}], [{"x":0,"y":0}, {"x":1,"y":1}, {"x":2,"y":2}, {"x":3,"y":3}, {"x":4,"y":4}]]	
[]	array		否
— — —x	integer	X 坐标 (0 ~ 1920)	否
— — —y	integer	Y 坐标 (0 ~ 1080)	否
—maskArea	array	遮蔽区域 最多两个；每个区域 3~5 个点 例 1：1 组遮蔽区域 4 个点	

参数	类型	说明	是否 必选
		[[{"x":0,"y":0}, {"x":1,"y":1}, {"x":2,"y":2}, {"x":3,"y":3}]] 例 1： 2 组遮蔽区域 第 1 组 3 个点 第 2 组 5 个点 [[{"x":0,"y":0}, {"x":1,"y":1}, {"x":2,"y":2}], [{"x":0,"y":0}, {"x":1,"y":1}, {"x":2,"y":2}, {"x":3,"y":3}, {"x":4,"y":4}]]	
[]	array		
— — —x	integer	X 坐标 (0 ~ 1920)	
— — —y	integer	Y 坐标 (0 ~ 1080)	

返回参数：

无

25.1.2 获取所有警戒视频通道信息 GET /alertVideos

操作类型：

GET

功能：

获取警戒通道当前的信息

接口 URI:

/alertVideos

参数:

无

返回值:

参数	类型	说明	是否必选
videos[]	array	放置视频布点信息的数组，内部可能会有多组视频信息。	
{}	obj	一组布控信息在此对象中，如果有多组布控将有多组对象	
—videoNo	integer	视频通道编号：支持 16 路视频，值为 1~16	
—videoName	string	摄像头布点名称，由用户起名，这将是此布点的唯一标识，后续操作都依赖这个名称。以下规则： 1 支持 1-50 字节长度 1 支持英文大/小写字母、数字、半角下划线 建议正则 /^[a-zA-Z0-9_]{4,16}	

参数	类型	说明	是否必选
—url	string	主码流, 通常为局域网内的 RTSP 地址。	
—subUrl	string	子码流, 通常为局域网内的 RTSP 地址。	
—username	string	IPC 接入的用户名	
—password	string	IPC 接入的密码	
—surveillanceEnabled	bool	布控使能开关	
—videoConStatus	integer	视频流连接状态 0:连接成功 1:连接失败 2:认证失败 3:视频流格式错误 4:超出通道最大分辨率 5:解码失败 6:超出最大解码能力	
—algType	integer	算法类型（插件预留）	
—alertRules	obj	警戒规则	
— —enable	bool	警戒使能 true 表示使能，false 表示禁能	是

参数	类型	说明	是否必选
— — alertType	integer	警戒类型, 取值[1-7]表示: 1: 人员越界 2: 人员入侵 3: 车辆越界 4: 车辆禁停 5: 车辆离开 6: 人员徘徊 7: 人员翻墙 8: 摔倒 9: 抽烟 10: 打电话 11: 玩手机 12: 奔跑 13: 睡岗	是
— — alertSubType	integer	警戒子类型, , 取值[1-6], 表示: 1: 单绊线检测 2: 进入区域 3: 离开区域 4: 进出区域 5: 区域停留 6:: 翻墙检测 根据警戒类型的不同, 可取不同值: alertType-人员越界: 1: 单绊线检测 alertType-人员翻墙 6:: 翻墙检测 alertType-人员入侵: 2: 进入区域 3: 离开区域 4: 进出区域 alertType-车辆越界: 1: 单绊线检测 alertType-车辆离开: 3: 离开区域	是

参数	类型	说明	是否必选
		alertType-车辆禁停、人员徘徊、摔倒、抽烟、打电话、玩手机、奔跑、睡岗 5:区域停留	
 areaType	integer	警戒区域类型, 取值[1-5], 表示: 1:单向单绊线 2:双向单绊线 3:单向双绊线 4:双向双绊线 5:区域 根据警戒子类型不同, 可取不同值: alertSubType-翻墙检测: 4:双向双绊线 alertSubType-单绊线检测: 1:单向单绊线 2:双向单绊线 alertSubType-进入区域: 5:区域 alertSubType-离开区域: 5:区域 alertSubType-进出区域: 5:区域	是

参数	类型	说明	是否必选
		alertSubType-区域停留: 5: 区域	
— — personPreOverWall	boolean	是否开启人员翻墙预警 只有当 alertSubType 为 6:翻墙检测 时有效	否
— — detectTargetList[]	array	检测目标数组，至少包含如下二类目标中的一种 在 alertType 为车辆越界 车辆禁停 车辆离开时有效	否
— — — targetName	string	检测目标名称，可取值如下，默认全选： (1) 机动车: "motorVehicle", (2) 非机动车: "nonMotorVehicle" 例: detectTargetList: ["motorVehicle", "nonMotorVehicle"]	是
— — targetRange	array	目标大小，有效值为[1, 100]，默认值[1, 50]	是
— — timeRange	array	布控时段, [0, 86400], (0 秒到 24*3600 秒)	是
— — stayDuration	integer	滞留时长, 0~120 秒，默认 5 秒 在警戒类型与警戒子类型取以下值时有效	否

参数	类型	说明	是否必选
		人员入侵-进入区域：0~120 秒 车辆禁停、人员徘徊-区域停留：0~1800 秒 摔倒、抽烟、打电话、玩手机、奔跑、睡岗-区域停留：0~120 秒	
— — cooldownDuration	integer	报警间隔，1~60 秒 默认 1 秒	是
— — alertArea	array	警戒区域 根据 areaType 的不同定义不同，绊线最多两条；区域最多四个；每条绊线 2~5 个点，每个区域 3~5 个点； 带方向箭头的线，将坐标按画线顺序放在数组中 例 1：单向/双向 单绊线/一个区域 有 5 个点 "alertArea":[[{"x":0,"y":0}, {"x":1,"y":1}, {"x":2,"y":2}, {"x":3,"y":3}, {"x":4,"y":4}]] 例：2 条绊线，每条绊线各 5 个点 "alertArea":[[{"x":0,"y":0}, {"x":1,"y":1}, {"x":2,"y":2}, {"x":3,"y":3}, {"x":4,"y":4}], [{"x":0,"y":0}, {"x":1,"y":1}, {"x":2,"y":2}, {"x":3,"y":3}, {"x":4,"y":4}]]	是
[]	array		否

参数	类型	说明	是否必选
x	integer	X 坐标（0 ~ 1920）	否
y	integer	Y 坐标（0 ~ 1080）	否
maskArea	array	遮蔽区域 最多两个；每个区域 3~5 个点 例 1：1 组遮蔽区域 4 个点 [[{"x":0,"y":0}, {"x":1,"y":1}, {"x":2,"y":2}, {"x":3,"y":3}]] 例 1：2 组遮蔽区域 第 1 组 3 个点 第 2 组 5 个点 [[{"x":0,"y":0}, {"x":1,"y":1}, {"x":2,"y":2}], [{"x":0,"y":0}, {"x":1,"y":1}, {"x":2,"y":2}, {"x":3,"y":3}, {"x":4,"y":4}]]	
[]	array		
x	integer	X 坐标（0 ~ 1920）	
y	integer	Y 坐标（0 ~ 1080）	

25.2 警戒抓拍事件查询

25.2.1 获取全量抓拍事件列表 GET/captures

操作类型：

GET

功能：

获取抓拍队列中的全部抓拍事件信息。

请求参数：

参数	类型	说明
pageOffset	integer	列表结果偏移量，从 0 开始，与 pageToken 不可一起使用
pageToken	integer	指示列表的起始位置，为空表未从 0 开始，与 pageOffset 不可一起使用，此值首次查询时不用填写，默认从 0 开始，后续填入返回的 nextPageToken 值
pageSize	integer	返回结果一页的最大数量，与 pageToken 配合使用，如反回结果的总量超过 pageSize，在返回参数中会有 nextPageToken 字段用于指示下一页起始位置
chnNo	integer	通道号 1~16 非必选
timeStart	integer	开始时间 Unix 时间戳 单位为秒 非必选
timeEnd	integer	结束时间 Unix 时间戳 单位为秒 非必选

参数	类型	说明
alertType	integer	警戒模式使用 警戒类型,取值详见 算法仓数据字典 - IPU-Firmware&SDK Dev - Meg-Wiki (megvii-inc.com) 事件类型字典
alertObjType	string	目标类型 取值详见 算法仓数据字典 - IPU-Firmware&SDK Dev - Meg-Wiki (megvii-inc.com) 目标类型字典

返回参数:

参数	类型	说明
nextPageToken	integer	下一页的起始地址, 放入下一次请求的 pageToken
totalCaptures	integer	现有总的抓拍数量
captures[]	array	抓拍事件消息列表, 按时间排序
{}	obj	对象内放一条消息, 如有多条消息将重复
—videoNo	integer	通道号 [1 16]
—channelName	string	摄像头布点名称
—trackId	integer	目标编号
—trackIdTimes	integer	目标编号出现的次数标记
—rule_id	integer	警戒子规则 id
—area_id	integer	警戒区域 id
—imageURI	string	捕获目标图片资源 URI
—fullImageURI	string	捕获背景图片资源 URI

参数	类型	说明
—catchTimestamp	integer	Unix 时间戳，目标捕获照片的时间，单位为毫秒
—alertType	integer	警戒类型, 取值详见 算法仓数据字典 - IPU-Firmware&SDK Dev - Meg-Wiki (megvii-inc.com) 事件类型字典
—alertSubType	integer	警戒子类型，取值[1-6], 表示：1:单绊线检测 2:进入区域 3:离开区域 4:进出区域 5:区域停留 6:翻墙检测
—areaType	integer	警戒区域类型 取值详见 算法仓数据字典 - IPU-Firmware&SDK Dev - Meg-Wiki (megvii-inc.com) 区域类型字典
— personPreOverWall	bool	是否是人员翻墙预警 只有翻墙检测 时有效
—objType	string	目标类型 取值详见 算法仓数据字典 - IPU-Firmware&SDK Dev - Meg-Wiki (megvii-inc.com) 目标类型字典

25.3 警戒模式下的 HTTP 主动上报

对接盘古，警戒模式目前只有抓拍信息

参数	类型	说明
alert_events[]	array	警戒事件信息
—area_id	integer	警戒区域 id，多个区域之间 id 唯一不相同
—area_type	string	警戒区域类型，详见 算法仓数据字典 - IPU-Firmware&SDK Dev - Meg-Wiki (megvii-inc.com) 区域类型字典
—alertor_id	integer	警戒规则 id，多个规则之间 id 唯一不相同

—alertor_type	string	警戒事件类型，详见 算法仓数据字典 - IPU-Firmware&SDK Dev - Meg-Wiki (megvii-inc.com) 报警事件字典
—target {}	obj	目标信息
— —target_type	string	目标类型，详见 算法仓数据字典 - IPU-Firmware&SDK Dev - Meg-Wiki (megvii-inc.com) 目标类型字典
— —target_id	integer	报警对象的 id，这个对于单个视频是唯一的
— —points[]	array	报警对象的坐标点数组，含左上角坐标和右下角坐标
— — —x	integer	横坐标
— — —y	integer	纵坐标
photo	string	背景大图 base64 编码
device_id	string	设备编号
stream_id	integer	通道号
frame_id	integer	帧 id
pts	integer	解码器解码时间戳，Unix 格式 单位毫秒
pts	integer	解码器显示时间戳，Unix 格式 单位毫秒
recv_ts	integer	帧到达时间，Unix 格式 单位毫秒
algType	integer	算法类型，0：未知；1：脸人/结构化；2：警戒。（可选字段，多算法上报平台时用到此字段）

上报数据举例：

```
{
```

```
"alert_events": [{  
  "area_id": 0,  
  "area_type": "polygon",  
  "alertor_id": 0,  
  "alertor_type": "tripwire",  
  "target": {  
    "target_type": "person",  
    "target_id": 4593450,  
    "points": [{  
      "x": 454,  
      "y": 5  
    }, {  
      "x": 656,  
      "y": 315  
    }]  
  }  
}]
```

```
    }],  
  
    "photo":  "",  
  
    "deviceId": "M030201202105000021",  
  
    "stream_id": 0,  
  
    "frame_id": 3050,  
  
    "dts": 1596503018000,  
  
    "pts": 1596503018000,  
  
    "recv_ts": 1596503018000,  
  
    "algType": 2  
  
}
```

25.4 警戒模式下的 websocket 主动上报

警戒模式目前只有抓拍信息

参数	类型	说明
alert_events[]	array	警戒事件信息
—area_id	integer	警戒区域 id，多个区域之间 id 唯一不相同

—area_type	string	警戒区域类型，详见 算法仓数据字典 - IPU-Firmware&SDK Dev - Meg-Wiki (megvii-inc.com) 区域类型字典
—alertor_id	integer	警戒规则 id，多个规则之间 id 唯一不相同
—alertor_type	string	警戒事件类型，详见 算法仓数据字典 - IPU-Firmware&SDK Dev - Meg-Wiki (megvii-inc.com) 报警事件字典
—target {}	obj	目标信息
— —target_type	string	目标类型，详见 算法仓数据字典 - IPU-Firmware&SDK Dev - Meg-Wiki (megvii-inc.com) 目标类型字典
— —target_id	integer	报警对象的 id，这个对于单个视频是唯一的
— —points[]	array	报警对象的坐标点数组，含左上角坐标和右下角坐标
— — —x	integer	横坐标
— — —y	integer	纵坐标
captureFace	string	小图 base64 编码，总开关开启并且小图开关也开启==>才推小图
fullImage	string	背景大图 base64 编码，抓拍背景图开启+总开关开启+推大图开关开启==>才推大图
device_id	string	设备编号
stream_id	integer	通道号
frame_id	integer	帧 id
dts	integer	解码器解码时间戳，Unix 格式 单位毫秒
pts	integer	解码器显示时间戳，Unix 格式 单位毫秒
recv_ts	integer	帧到达时间，Unix 格式 单位毫秒

上报数据举例：

```
{  
  "alert_events": [{  
    "area_id": 0,  
    "area_type": "polygon",  
    "alertor_id": 0,  
    "alertor_type": "tripwire",  
    "target": {  
      "target_type": "person",  
      "target_id": 4593450,  
      "points": [{  
        "x": 454,  
        "y": 5  
      }, {  
        "x": 656,  
        "y": 315
```

```
        }  
    }  
},  
"fullImage": "",  
"deviceId": "M030201202105000021",  
"stream_id": 0,  
"frame_id": 3050,  
"dts": 1596503018000,  
"pts": 1596503018000,  
"recv_ts": 1596503018000  
}
```

26 单路多算法模式操作 API

26.1 警戒多子规则视频资源操作 API

26.1.1 布控 POST/multiAlgVideos/ {videoNo} /multiAlert

操作类型:

POST

功能:

对警戒视频通道进行布控，设置警戒规则等，支持多个子规则

接口 URI:

/multiAlgVideos/ {videoNo} /multiAlert

请求参数:

参数	类型	说明	是否必选
surveillanceEnabled	bool	布控使能开关	是
alertRulesValidity	bool	警戒规则有效性	否
algType	integer	算法类型，详见 算法仓数据字典 - IPU-Firmware&SDK Dev - Meg-Wiki (megvii-inc.com) 算法类型字典	是
ruleIndex	integer	算法规则索引，暂只支持同时布控 2 个规则，取值 1 ~ 2	是
ruleEnable	bool	算法规则使能，控制算法规则是否启用。true 表示使能，false 表示禁能	是
alertRules{ }	obj	警戒规则	是
—proc	string	固定值，表示 process = init_rules	是

—width	integer	视频分辨率-宽 (0 ~ 1920) 当使用归一化坐标时, 可省略	否
—height	integer	视频分辨率-高 (0 ~ 1080) 当使用归一化坐标时, 可省略	否
—fps	double	业务层送帧给 sdk 的帧率, 一般由算法仓开发人员推荐, 如警戒为 8fps, 城管为 0.5 目前统一由算法抽帧	否
— customizedOptions {}	obj		是
— —area_list[]	array	roi 区域 list	是
{}	obj		
— — —area_type	string	区域类型, 详见 算法仓数据字典 - IPU-Firmware&SDK Dev - Meg-Wiki (megvii-inc.com) 区域类型字典	是
— — —area_id	integer	roi id, 需要唯一	是
— — — alertor_ids[]	array	该 roi 区域关联的警报器, 一个区域可以设置多个警报器	是
— — —points[]	array	多边形的顶点坐标, 使用双拌线时, 由于包含两条拌线, points 项格式详见该表格下方	是
{}	obj		
— — — —x	double	多边形顶点 x 坐标 (0 ~ 1) 归一化坐标	是
— — — —y	double	多边形顶点 y 坐标 (0 ~ 1) 归一化坐标	是

— — alertor_configs[]	array	报警器 list	
{}	obj		
— — —alertor_id	integer	报警器 id, 需唯一	是
— — — alertor_type[]	array	报警事件, 详见 算法仓数据字典 - IPU-Firmware&SDK Dev - Meg-Wiki (megvii-inc.com) 事件类型字典	是
— — —target_min	double	目标大小过滤逻辑, 目标面积占图片面积最小比例 $0.00 \leq \text{target_min} < \text{target_max} \leq 1.00$ 默认 0.05	是
— — —target_max	double	目标大小过滤逻辑, 目标面积占图片面积最大比例 $0.00 \leq \text{target_min} < \text{target_max} \leq 1.00$ 默认 0.5	是
— — — target_type[]	array	目标类型, 详见 算法仓数据字典 - IPU-Firmware&SDK Dev - Meg-Wiki (megvii-inc.com) 目标类型字典	是
— — — allowed_colors[]	array	alertor_type 里面含安全工服时必须填, 未含时不填 red: 红色 blue: 蓝色 yellow: 黄色 gray: 灰色 安全工服支持红黄蓝灰自定义配置报警, 工服颜色不符合规定, 未穿戴工服均报警	alertor_type 里面含安全工服时必须填

— — — expected_num	integer	alertor_type 里面含消防设施时必须填，未含时不填 希望画面中出现的灭火器个数，默认为 1，画面中灭火器个数少于 'expected_num' 会触发报警	alertor_type 里面含消防设施时必须填，未含时不填
— — — target_min_num	integer	alertor_type 里面含人员离岗时必须填，未含时不填 目标个数最小值	alertor_type 里面含人员离岗时必须填，未含时不填
— — — target_max_num	integer	alertor_type 里面含人员离岗时必须填，未含时不填 目标个数最大值	alertor_type 里面含人员离岗时必须填，未含时不填
— — —obj_count		alertor_type 里面含人员聚集时必须填，未含时不填 区域内的人数>=obj_count 时才会报警	alertor_type 里面含人员聚集时必须填，未含时不填
— — —duty_ratio	float	值域为 0 - 1，只有 alertor_type 里面含杂物堆放(sundry_detect)有该参数，表示当杂物与布控区域比例大于该值时才报警	只有 alertor_type 里面含杂物堆放(sundry_detect)有该参数
— — — interrupt_endurance	integer	单位是秒，只有物品算法仓（即目前 alertor_type 里面含杂物堆放(sundry_detect)时）必填，其他不填 当 Track 连续 interrupt_endurance 帧不满足报警，则会重新记录这个 Track，之前累计的满足报警状态帧数会被清空	只有物品算法仓（即目前 alertor_type 里面含杂物堆放(sundry_detect)时）必填，其他不填
— — — personPreOverWall	int	alertor_type 里面含翻墙（overwall）时必须填，未含时不填	alertor_type 里面含 overwll 时必须填，未含时不填

		表示人员预翻墙是否报警	
— — — cooldown_duration	integer	报警间隔，单位：秒 [0~1,800]	是
— — —duration	integer	目标事件持续多久报警，单位：秒 [0~60]	是
— — —timeRange	array	布控时段[0, 86400]，(0 秒到 24*3600 秒)	是

使用双拌线时， 由于包含两条拌线， points 项格式为：

```

"points": [
    [
        // 第一条拌线的坐标
        {"x": 0.2, "y": 0.5}, {"x": 0.8, "y": 0.5}
    ],
    [
        // 第二条拌线的坐标
        {"x": 0.2, "y": 0.6}, {"x": 0.8, "y": 0.6}
    ],
]

```

例 1，配置车辆越界加屏蔽区域：

```

{
  "surveillanceEnabled": true,
  "ruleIndex": 1,
  "ruleEnable": true,
  "algType": 11,
  "alertRules": {
    "proc": "init_rules",
    "width": 1920,

```

```
"height":1080,
"fps":8,
"customizedOptions":{
  "area_list":[
    {
      "area_type":"single_line_both",
      "area_id":1,
      "alertor_ids":[
        1
      ],
      "points":[
        {
          "x":0.18,
          "y":0.77
        },
        {
          "x":0.42,
          "y":0.74
        }
      ]
    },
    {
      "area_type":"polygon",
      "area_id":2,
      "alertor_ids":[
        2
      ],
      "points":[
        {
          "x":0.01,
```

```

        "y":0.01
    },
    {
        "x":0.99,
        "y":0.01
    },
    {
        "x":0.99,
        "y":0.99
    },
    {
        "x":0.01,
        "y":0.99
    }
]
}
],
"alertor_configs":[
{
    "alertor_id":1,
    "alertor_type":[
        "tripwire"
    ],
    "target_min":0,
    "target_max":1,
    "target_type":[
        "car"
    ],
    "cooldown_duration":1,
    "duration":0,

```

```

"timeRange": [0, 86400]
    },
    {
      "alertor_id": 2,
      "alertor_type": [
        "mask_area"
      ],
      "target_type": [
        "car"
      ]

      "cooldown_duration": 0,
      "duration": 0,

"timeRange": [0, 86400]
    }
  ]
}
}
}
}

```

例 2，配置安全工服报警：

```

{
  "surveillanceEnabled": true,
  "ruleIndex": 1,
  "ruleEnable": true,
  "algType": 13,

```

```
"alertRules": {
  "proc": "init_rules",
  "width": 1920,
  "height": 1080,
  "fps": 25,
  "customizedOptions": {
    "area_list": [
      {
        "area_type": "polygon",
        "area_id": 1,
        "alertor_ids": [
          1
        ],
        "points": [
          {
            "x": 0.0,
            "y": 0.0
          },
          {
            "x": 1.0,
            "y": 0.0
          },
          {
            "x": 1.0,
            "y": 1.0
          },
          {
            "x": 0.0,
            "y": 1.0
          }
        ]
      }
    ]
  }
}
```



```
    ]
  }
],
"alertor_configs": [
  {
    "alertor_id": 1,
    "alertor_type": [
      "no_uniform"
    ],
    "target_min": 0.05,
    "target_max": 0.5,
    "target_type": [
      "uniform"
    ],
    "allowed_colors": [
      "red"
    ],
    "cooldown_duration": 1,

    "duration": 2,

    "timeRange": [0, 86400]
  }
]
}
}
```

返回参数:

无

26.2 获取所有多算法布控信息 GET/multiAlgVideos

操作类型:

GET

功能:

从设备中获取当前布控的参数信息

接口 URI:

/multiAlgVideos

请求参数:

无

返回参数:

参数	类型	说明	是否必选
----	----	----	------

videos[]	array	放置视频布点信息的数组，内部可能会有多组视频信息。	
{}	obj	一组布控信息在此对象中，如果有多组布控将有多个对象	
—videoNo	integer	视频通道编号：支持 16 路视频，值为 1~16	
—videoName	string	摄像头布点名称，由用户起名，这将是此布点的唯一标识，后续操作都依赖这个名称。以下规则： 1 支持 1-50 字节长度 1 支持英文大/小写字母、数字、半角下划线 建议正则 /^[a-zA-Z0-9_]{4,16}	
—url	string	主码流，通常为局域网内的 RTSP 地址。	
—subUrl	string	子码流，通常为局域网内的 RTSP 地址。	
—username	string	IPC 接入的用户名	
—password	string	IPC 接入的密码	
—rotation	integer	图像旋转的角度，为配合不同的摄像头安装角度，目前只支持 4 个值：0，90，180，270。分别代表顺时针旋转的角度 0°，90°，180°，270°，如错误将检测不到人脸。如输入不为以上 4 个值，将默认为 0°来处理。	
—surveillanceEnabled	bool	布控使能开关	

—videoConStatus	integer	视频流连接状态 0:连接成功 1:连接失败 2:认证失败 3:视频流格式错误 4:超出通道最大分辨率 5:解码失败 6:超出最大解码能力	
—algRules[]	array	算法规则数组	
{ }	obj		
— —ruleIndex	integer	算法规则索引，暂只支持同时布控 2 种规则，取值 1 ~ 2	
— —ruleEnable	bool	算法规则使能，控制算法规则是否启用。true 表示使能，false 表示禁能	
— —algType	integer	算法类型，根据不同的算法类型取不同的算法参数 0 - 脸 人： 取下方结构化参数 structuresRule 1 - 结构化： 取下方结构化参数 structuresRule 10-周界警戒：取下方 alertRules（以 1.2 格式解析） 11-行为警戒：取下方 alertRules（以 1.2 格式解析） >=12 其他算法仓：取下方 alertRules（以该表格格式解析）	
— —structuresRule	obj	结构化规则参数	
— — —livenessEnabled	bool	是否开启活体	

— — — livenessThreshold	float	活体判断阈值，（0~100）之间的浮点数，当活体开关关闭时，该值非必须，如设置为 0 也相当于关闭，但会消耗系统资源。如需要关闭该功能可到 configs 公共参数设置接口进行设置，但所有视频都将不做活体检测。	
— — — attributeEnabled	bool	是否开启属性分	
— — — detectTargetList[]	array	检测目标数组，至少包含如下五类目标中的一种	
— — — —targetName	string	检测目标名称，可取值如下，默认全选： (1) 人脸： "face" (2) 人形： "human" (3) 机动车： "motorVehicle", (4) 非机动车： "nonMotorVehicle" (5) 车牌： "plate"	
— — —pushConfigs {}	obj	推图参数配置	
— — — —pushMode	string	0- 间隔推图模式，1- 最优推图模式，2 - 最快推图模式，3 - 无限推图	
— — — —retryCount	integer	二次识别次数。限定条件： 0~5000 次范围，0 代表不进行二次识别；默认 5 次。仅在最快推图模式下生效	

— — — — retryInterval	integer	二次识别的间隔时长。限定条件：0~60 秒范围，0 代表关闭二次识别功能；默认为 5 秒。仅在最快推图模式下生效	
— — — — strangerRetryCount	integer	陌生人重新判定次数。对识别结果进行计数，如结果为未识别或陌生人，连续判断平均值低于陌生人阈值，将判定为陌生人。如为陌生人，将产生陌生人报警输出。范围：0~5 次，默认为 2 次。	
— — — — pushInterval	integer	最小推图间隔。（500-10000，单位是毫秒，默认 500ms）	
— — — — maxPushNumber	integer	最优推图的最大张数推图张数。限定条件：1~10 张，默认 1 张	
— — — captureConfigs {}	obj	抓拍模式参数配置	
— — — —captureMode	integer	抓拍模式，默认为全抓拍模式： 1 0：全抓模式 算法检测到目标（人、车、脸、车牌），不用质量判断模型过滤目标，抓到的目标全推出来 1 1：高质量模式，默认模式 只抓拍满足“质量阈值”的目标，其中质量阈值的定义针对不同目标包含不同维度 1 2：自定义模式	

		只抓拍满足“用户定义质量阈值”的目标	
— — — —quality{}	obj	视频结构化质量，非必选。上述抓拍模式配置为”自定义模式“时使用	
— — — — — faceQuality	integer	人脸综合质量 取值范围【0,100】	
— — — — — humanQuality	integer	人形综合质量 取值范围【0,100】	
— — — — — motorVehicleQuality	integer	机动车综合质量 取值范围【0,100】	
— — — — — nonMotorVehicleQuality	integer	非机动车综合质量 取值范围【0,100】	
— — — — — plateQuality	integer	车牌综合质量 取值范围【0,100】	
— — — —filter{}	obj	过滤条件，条件越高，通过抓拍的人脸将越少，但人脸质量越高，识别的效果会越高。如条件越低，通过抓拍的人脸会越多，但人脸质量下线会降低，影响识别的效果。	
— — — —roll	integer	人脸旋转角度限制（0~180），建议<=45°	
— — — —yaw	integer	人脸水平角度限制（0~100），建议<=35°	
— — — —pitch	integer	人脸垂直角度限制（0~100），建议<=45°	
— — — —blurness	float	图像清晰度限制（0~100），建议<=60	
— — — —faceMin	integer	人脸允许的长或宽的最小像素（20~380），值越小识别效果越差，建议>=80	

— — — —bodyMin	integer	人体允许的长或宽的最小像素（20~380），值越小识别效果越差，建议>=80	
— — —group[]	array	如关联多个分组，在数组中添加多个项	
{}	obj	设定关联的分组与有效时间，最多 20 组。	
— — — —groupName	string	必填，视频绑定的人脸分组名称，视频中抓拍的人脸将与这个 group 中的人脸进行识别。如高于比对阈值将引发报警或允许通行动作	
— — — — startTime	string	必填，该绑定分组有效的开始时间，格式：HHMMSS，例子：“080000”，为 24 小时制，默认为“000000”	
— — — — endTime	string	必填，该绑定分组失效的开始时间，格式：HHMMSS，例子：“184500”，为 24 小时制，默认为“000000”，如 startTime 与 endTime 两个值都为默认值“000000”，将 24 小时有效，不支持跨天	
— — — —searchThresholdHigh	float	必填，人脸识别的高限阈值，（0~100）之间的浮点数，如最高相似度的超过阈值，可通过 Alert 接口得到超阈值的不同组里的比对相似度列表。建议 70~80，具体值根据实际情况调整	
— — — —searchThresholdLow	float	必填，人脸识别的低限阈值，（0~100）之间的浮点数，如最高相似度低于低线阈值，代表这是一个陌生人。可通过 Alert 接口得到低阈值的不同组里的比对相似度列表	
— — —globalStranger {}	obj	非必填，全局陌生人配置	

— — — —enable	bool	必填，配置开关 true 启用 false 禁用	
— — — — searchThresholdLow	integer	必填，人脸识别的低限阈值，（0~100）之间的数，如最高相似度低于低线阈值，代表这是一个陌生人。建议：20，可通过 Alert 接口得到低阈值的不同组里的比对相似度列表	
— — — —startTime	string	必填，该绑定分组有效的开始时间，格式：HHMMSS，例子：“080000”，为 24 小时制，默认为“000000”	
— — — —endTime	string	必填，该绑定分组失效的开始时间，格式：HHMMSS，例子：“184500”，为 24 小时制，默认为“000000”，如 startTime 与 endTime 两个值都为默认值“000000”，将 24 小时有效	
— —alertRules{ }	obj	类警戒算法规则，一个算法支持两个子规则	
— — —proc	string	固定值，表示 process = init_rules	是
— — —width	integer	视频分辨率-宽（0 ~ 1920）当使用归一化坐标时，可省略	否
— — —height	integer	视频分辨率-高（0 ~ 1080）当使用归一化坐标时，可省略	否
— — —fps	double	业务层送帧给 sdk 的帧率，一般由算法仓开发人员推荐，如警戒为 8fps，城管为 0.5 目前统一由算法抽帧	否
— — — customizedOptions{ }	obj		是
— — — —area_list[]	array	roi 区域 list	是

{}	obj		
— — — — — area_type	string	区域类型，详见 算法仓数据字典 - IPU-Firmware&SDK Dev - Meg-Wiki (megvii-inc.com) 区域类型字典	是
— — — — —area_id	integer	roi id, 需要唯一	是
— — — — — alertor_ids[]	array	该 roi 区域关联的警报器，一个区域可以设置多个警报器	是
— — — — —points[]	array	多边形的顶点坐标，使用双拌线时， 由于包含两条拌线， points 项格式详见该表格下方	是
{}	obj		
— — — — — —x	double	多边形顶点 x 坐标 (0 ~ 1) 归一化坐标	是
— — — — — —y	double	多边形顶点 x 坐标 (0 ~ 1) 归一化坐标	是
— — — — alertor_configs[]	array	报警器 list	
{}	obj		
— — — — — alertor_id	integer	报警器 id, 需唯一	是
— — — — — alertor_type[]	array	报警事件, 详见 算法仓数据字典 - IPU-Firmware&SDK Dev - Meg-Wiki (megvii-inc.com) 事件类型字典	是
— — — — — target_min	double	目标大小过滤逻辑，目标面积占图片面积最小比例 0.00<=target_min<target_max<=1.00 默认 0.05	是
— — — — — target_max	double	目标大小过滤逻辑，目标面积占图片面积最大比例 0.00<=target_min<target_max<=1.00 默认 0.5	是

— — — — — target_type[]	array	目标类型，详见 算法仓数据字典 - IPU-Firmware&SDK Dev - Meg-Wiki (megvii-inc.com) 目标类型字典	是
— — — — — allowed_colors[]	array	alertor_type 里面含安全工服时必须填，未含时不填 red: 红色 blue: 蓝色 yellow: 黄色 gray: 灰色 安全工服支持红黄蓝灰自定义配置报警，工服颜色不符合规定，未穿戴工服均报警	alertor_type 里面含安全工服时必须填
— — — — — expected_num	int	alertor_type 里面含消防设施时必须填，未含时不填 希望画面中出现的灭火器个数，默认为1，画面中灭火器个数少于 'expected_num' 会触发报警	alertor_type 里面含消防设施时必须填，未含时不填
— — — — — target_min_num	integer	alertor_type 里面含人员离岗时必须填，未含时不填 目标个数最小值	alertor_type 里面含人员离岗时必须填，未含时不填
— — — — — target_max_num	integer	alertor_type 里面含人员离岗时必须填，未含时不填 目标个数最大值	alertor_type 里面含人员离岗时必须填，未含时不填
— — — — — obj_count		alertor_type 里面含人员聚集时必须填，未含时不填 区域内的人数>=obj_count 时才会报警	alertor_type 里面含人员聚集时必须填，未含时不填

— — — — — duty_ratio	float	值域为 0 - 1，只有 alertor_type 里面含杂物堆放 (sundry_detect) 有该参数，表示当杂物与布控区域比例大于该值时才报警	只有 alertor_type 里面含杂物堆放 (sundry_detect) 有该参数
— — — — — interrupt_endurance	integer	单位是秒，只有物品算法仓（即目前 alertor_type 里面含杂物堆放 (sundry_detect) 时）必填，其他不填 当 Track 连续 interrupt_endurance 帧不满足报警，则会重新记录这个 Track，之前累计的满足报警状态帧数会被清空	只有物品算法仓（即目前 alertor_type 里面含杂物堆放 (sundry_detect) 时）必填，其他不填
— — — — — personPreOverWall	int	alertor_type 里面含翻墙 (overwall) 时必填，未含时不填 表示人员预翻墙是否报警	alertor_type 里面含 overwall 时必填，未含时不填
— — — — — cooldown_duration	integer	报警间隔，单位：秒 [0~1,800]	是
— — — — —duration	integer	目标事件持续多久报警，单位：秒 [0~60]	是
— — — — — timeRange	array	布控时段[0, 86400]，(0 秒到 24*3600 秒)	是

使用双拌线时， 由于包含两条拌线，points 项格式为：

```
"points": [
    [                                // 第一条拌线的坐标
```

```
        {"x": 0.2, "y": 0.5}, {"x": 0.8, "y": 0.5}
    ],
    [
        // 第二条拌线的坐标
        {"x": 0.2, "y": 0.6}, {"x": 0.8, "y": 0.6}
    ],
]
```

26.3 多算法布控使能开关 POST/multiAlgVideos/ {videoNo} /surveillanceEnable

操作类型:

POST

功能:

对多算法频通道布控开关进行设置，开启或关闭布控

接口 URI:

/multiAlgVideos/ {videoNo} /surveillanceEnable

请求参数

参数	类型	说明	是否必选
surveillanceEnabled	bool	布控使能开关	是

返回参数:

无

26.4 算法能力集获取接口 GET/getAlgInfos

操作类型:

GET

功能:

从设备中获取当前算法布控的参数信息

接口 URI:

/getAlgInfos

请求参数:

无

返回参数:

参数	类型	说明
alg_infos[]	array	各个算法仓能力描述
{}	obj	单个算法仓能力描述

—alg_type	integer	算法类型数值, 数值含义具体可查询 算法仓数据字典 - IPU-Firmware&SDK Dev - Meg-Wiki (megvii-inc.com) 算法类型字典
—items_type	string	算法类型字符串, 含义具体可查询 算法仓数据字典 - IPU-Firmware&SDK Dev - Meg-Wiki (megvii-inc.com) 算法类型字典
—version	string	算法版本
—mask_area_num	integer	支持屏蔽区域的个数, 0 表示不支持
—area_max_num	integer	单规则支持的区域最大个数
— roi_area_points_range	array	roi 区域支持的点数范围 如: [3, 8]
— mask_area_points_range	array	屏蔽区域支持的点数范围 如: [3, 8]
—alertor_type[]	array	
{}	obj	
— —type	string	事件类型字符串 详见 算法仓数据字典 - IPU-Firmware&SDK Dev - Meg-Wiki (megvii-inc.com)
— —type_num	integer	事件类型数值 详见 算法仓数据字典 - IPU-Firmware&SDK Dev - Meg-Wiki (megvii-inc.com)

— —type_attr	integer	该事件的属性，由于之前警戒区分周界警戒和行为警戒，但现在警戒只有一个警戒算法仓，为了兼容，利用该字段进行形式上的区分 0：不区分； 1：属周界警戒； 2：属行为警戒； 该字段目前只有警戒仓有效
— —target_type[]	array	字符串数组，该事件支持的目标类型 详见 算法仓数据字典 - IPU-Firmware&SDK Dev - Meg-Wiki (megvii-inc.com)
— —area_type[]	array	字符串数组，该事件支持的区域类型 详见 算法仓数据字典 - IPU-Firmware&SDK Dev - Meg-Wiki (megvii-inc.com)

示例：

```
{
    "alg_infos":[
        {
            "alg_type": 14,
            "alg_frame_rate": 8.0,
```



```
    "items_type": "alert_alarm",

    "version": "V1",

    "mask_area_num": 2,

    "area_max_num" : 4,
    "roi_area_points_range": [3, 8],
    "mask_area_points_range": [3,8] ,

    "alertor_type":[

        {

            "type":" intrusion",

            "type_num":28,

            "target_type":["person"],

            "area_type":["polygon"]

        },

        {

            "type":" tripwire",

            "type_num":29,

            "target_type":["car", "cycle", "person"],
```

```

"area_type": ["single_line",
"single_line_both"]

}

]

},

{

    "alg_type": 13,

    "alg_frame_rate": 25.0,

    "items_type": "safety_alarm",

    "version": "V1",

    "mask_area_num": 0,

    "area_max_num" : 4,
    "roi_area_points_range": [3, 8],
    "mask_area_points_range": [3,8] ,

    "alertor_type":[

        {

            "type": " no_safetycap",

```

```

        "type_num":19,
        "target_type":["person"],
        "area_type":["polygon"]
    },
    {
        "type":" no_uniform",
        "type_num":20,
        "target_type":["person"],
        "area_type":["polygon"]
    }
]
}
]
}

```

26.5 事件规则卡片属性获取接口 GET/getRuleTypeAttr/{typeNo}

操作类型:

GET

功能：根据规则卡片类型编号获取该事件规则卡片属性信息。

接口 URI：/getRuleTypeAttr/{typeNo}

请求参数：

{typeNo}为规则卡片类型编号，参考[算法仓数据字典 - IPU-Firmware&SDK Dev - Meg-Wiki \(megvii-inc.com\)](http://megvii-inc.com/wiki/algorithm-data-dictionary-ipu-firmware&SDK-Dev-Meg-Wiki)

返回参数：

参数	类型	说明
type	string	事件规则卡片字符串名，参考 算法仓数据字典 - IPU-Firmware&SDK Dev - Meg-Wiki (megvii-inc.com)
value	integer	事件规则卡片编号，参考 算法仓数据字典 - IPU-Firmware&SDK Dev - Meg-Wiki (megvii-inc.com)
attr		该事件规则卡片的属性描述
{}	obj	
— target_type[]	array	字符串数组，该事件支持的目标类型 详见 算法仓数据字典 - IPU-Firmware&SDK Dev - Meg-Wiki (megvii-inc.com)
—area_type[]	array	字符串数组，该事件支持的区域类型 详见 算法仓数据字典 - IPU-Firmware&SDK Dev - Meg-Wiki (megvii-inc.com)
—params[]	array	该规则卡片支持的参数字段信息
{}	bool	是否启用该语音

— —param	string	参数名称，各个参数已在/multiAlgVideos/ {videoNo} / multiAlert 接口中列出
— —value_type	string	该参数的数值类型，int(整形)、double（浮点型）、string（字符串）、array_string（字符串数组）
— —default	类型依据 value_type 字段	该参数的默认值
— —range[]	array	该参数的范围，里面装载内容类型，依据 value_type 字段

示例：

```
{
  "type": "no_uniform",
  "value": 20,
  "attr": {
    "target_type": [
      "person"
    ],
    "area_type": [
      "polygon"
    ],
    "params": [
      {
        "param": "target_min",
        "value_type": "double",
        "default": 0.05,
        "range": [0, 1]
      }
    ]
  }
}
```

```

{
  "param": "target_max",
  "value_type": "double",
  "default": 0.5,
  "range": [0, 1]
},
{
  "param": "duration",
  "value_type": "int",
  "default": 30,
  "range": [0, 86399]
},
{
  "param": "cooldown_duration",
  "value_type": "int",
  "default": 3,
  "range": [0, 1799]
},
{
  "param": "allowed_colors",
  "value_type": "array_string",
  "default": [""],
  "range": ["red", "blue", "yellow", "gray"]
}
]
}
}

```

26.6 算法仓数据字典

算法类型

算法类型索引	算法类型字符串	代表的算法仓
10	perimeter_alarm	周界警戒算法（SDK）
11	behavior_alarm	行为警戒算法（SDK）
12	city_alarm	城管算法仓
13	safety_alarm	生产安监算法仓
14	alert_alarm	警戒算法仓（包含周界、行为）
15	mclz_alarm	明厨亮灶算法仓
16	goods_alarm	物品警戒算法仓
17	fire_alarm	消防算法仓

注：其中 10、11 为旧版本 SDK 形式的算法类型，对接算法仓后将不存在此两种类型。

事件类型

事件索引	事件类型	事件名称	所属算法仓	备注
1		人员越界	警戒算法仓	历史 SDK 版本
2		人员入侵		
3		车辆越界		
4	park	车辆禁停		

5	exit	车辆离开		
6	wander	人员徘徊		
7	overwall	人员越界-翻墙检测		
8	fall	摔倒检测		
9	smoke	抽烟检测		
10	call	打电话		
11	watch_phone	看手机		
12	run	人员奔跑		
13	sleep	睡岗检测		
14	待定	经营类	城管算法仓	
15	待定	杂物类		
16	待定	普通垃圾		
17	待定	广告		
18	待定	地面积水		
19	no_safetycap	未佩戴安全帽报警	安监算法仓	
20	no_uniform	未穿戴安全工服报警		
21	no_safetybelt	未佩戴安全带报警		
22	flame	火焰报警		
23	smog	烟雾报警		

24	oil_spill	液体泄露检测		
25	person_off_duty_querying	人员值岗/离岗-离岗	警戒算法仓新增项	
26	gathering	人员聚众		
27	fight	人员扭打		*不支持屏蔽区域
28	intrusion	入侵		
29	tripwire	越界		
30	no_vest	未佩戴反光衣报警	安监算法仓新增项	
31	lose_fire_equipment	消防设施检测(包含灭火器和灭火箱)		
32	mask_area	屏蔽区域	屏蔽区域	屏蔽区域独立为事件类型，仅警戒支持
33	trashbin	垃圾桶未盖报警	明厨亮灶算法仓	
34	mice	老鼠检测		
35	sundry_detect	杂物堆放	物品算法仓	

目标类型

目标索引	目标类型	目标名称
1	face	人脸
2	person	人
3	car	机动车

目标索引	目标类型	目标名称
4	cycle	非机动车
5	plate	车牌
6	head	人头
7	flame	火焰
8	crowd	聚众整体
9	fighter	打架者
10	smog	烟雾
11	oil_spill	油品
12	fire_equipment	消防设备
13	trashbin	垃圾桶
14	mice	老鼠
15	sundry	杂物

区域类型

区域索引	区域类型	区域名称
1	single_line	单向单绊线
2	single_line_both	双向单绊线
3	multi_line	单向双绊线

区域索引	区域类型	区域名称
4	multi_line_both	双向双绊线
5	polygon	区域

27 本地 GUI 配置 API

27.1 本地 GUI 布局 API

27.1.1 设置 GUI 布局 POST/ GUI/layout

操作类型:

POST

功能:

设置 GUI 布局，并连接相应的视频流

接口 URI:

GUI/layout

请求参数:

参数	类型	说明	是否必选
----	----	----	------

guiEnable	integer	是否使能本地 GUI 0-不使能 1-使能	是
windowNum	integer	窗口数，支持 1,4,9 分屏	guiEnable 为 1 时 必选
videos[]	array	窗口链接的视频通道编号。数组长度为 windowNum	guiEnable 为 1 时 必选
{}	obj		guiEnable 为 1 时 必选
— — videoChn	integer	当前数组下标对应窗口需要链接的视频通道号（1~16），-1 表示不需要连接视频	guiEnable 为 1 时 必选

返回参数：

无

27.1.2 获取 GUI 布局 GET/ GUI/layout

操作类型：

GET

功能：

从设备中获取本地 GUI 配置查询

接口 URI：

/GUI/layout

请求参数:

无

返回参数:

参数	类型	说明	是否必选
guiEnable	integer	是否使能本地 GUI 0-不使能 1-使能	是
windowNum	integer	窗口数, 支持 1, 4, 9 分屏	guiEnable 为 1 时必选
videos[]	array	窗口链接的视频通道编号。数组长度为 windowNum	guiEnable 为 1 时必选
{}	obj		guiEnable 为 1 时必选
— — videoChn	integer	当前数组下标对应窗口需要链接的视频通道号 (1~16), -1 表示不需要连接视频	guiEnable 为 1 时必选

28. 高级 Debug 相关 API

28.1 指定网卡测试 ip 网络通畅 GET/debug_ping

操作类型:

GET

功能:

指定网卡测试 ip 网络是否能 ping 通

接口 URI:

/debug_ping

请求参数:

参数	类型	说明	必选
eth	string	指定的设备网卡，wan 或 lan；若无指定，则使用默认网卡	否
ip	string	要测试的 ip 网络	是

接口示例:

/debug_ping?eth=wan&ip=192.168.1.100

返回参数:

参数	类型	说明
timeDelay	string	延时相关信息
packetReceive	string	发收包相关信息

响应数据示例：

```
{  
  
  "timeDelay": "64 bytes from 192.168.1.100: seq=0 ttl=254 time=5.192 ms",  
  
  "packetReceive": "1 packets transmitted, 1 packets received, 0% packet loss"  
  
}
```

错误码：

状态码	错误码	消息	说明
400	104900	ETH_IS_ERR	网卡错误不合法
400	104901	IP_ADDR_IS_ERR	IP 地址不合法
400	104902	IP_ADDR_TEST_FAIL	测试操作失败

28.2 设置 telnet GET/telnet_control

操作类型：

GET

功能：

配置 telnet

接口 URI: /telnet_control

请求参数:

参数	类型	说明
telnetEnabled	bool	开启和关闭 telnet

接口示例:

/telnet_control?telnetEnabled=true

返回参数:

1. 不带参数 telnetEnabled, 为获取 telnet 状态

```
{
  "code": 0,
  "message": "",
  "timestamp": 1639539718560,
  "time_cost": 0,
  "data":
  {
    "telnetEnabled": true
  }
}
```


2. 带参数 telnetEnabled, 设置 telnet

```
{
  "code": 0,
  "message": "",
  "timestamp": 1639539718560,
  "time_cost": 0,
}
```

错误码:

状态码	错误码	消息	说明
400	104001	ERR_ILLEGAL_ARGUMENT	参数格式错误
400	105000	ERR_INTERNAL_ERROR	内部错误, 设置失败

29 语音报警配置 API

29.1 语音文件操作

29.1.1 语音文件的导入 POST/setVoiceFile

操作类型:

POST

功能: 语音文件的导入。

语音文件限制:

1 语音文件类型支持：PCM、MP3

1 语音文件大小限制： PCM 文件<= 5MB MP3 文件<=1MB

1 语音文件名后缀限制： .pcm .mp3 .MP3

接口 URI:/ setVoiceFile

请求参数：

参数	类型	说明
voiceFileNo	integer	语音文件编号，1~10
voiceFileName	string	语音文件名称 小于 256 个字符
fileSize	integer	语音文件大小（可选）
voiceFile	binary	语音文件
fileMd5	string	语音文件的 md5 值（可选）

29.1.2 语音文件的查询 GET/voiceFiles

操作类型：

GET

功能：语音文件的查询。

接口 URI:/ voiceFiles

请求参数:

无

返回参数:

参数	类型	说明
voiceFiles[]	array	语音文件
{}	obj	单个语音文件的描述对象，若有多个文件，则有多个对象
—voiceFileNo	integer	语音文件编号，1~10
—voiceFileName	string	语音文件名称

29.1.3 语音文件删除 DELETE/ voiceFile /{ voiceFileNo }

操作类型:

DELETE

功能: 删除对应的语音文件

接口 URI: / voiceFile /{ voiceFileNo }

请求参数:

{ voiceFileNo }为语音文件编号，1~10

返回参数:

无

29.1.4 触发语音报警 POST / voiceTrigger /{ voiceFileNo }

操作类型:

POST

功能: 触发语音报警。

接口 URI: / voiceTrigger /{ voiceFileNo }

请求参数:

{ voiceFileNo }为语音文件编号, 1~10

返回参数:

无

29.2 语音报警绑定

29.2.1 语音报警绑定 POST/voice

操作类型:

POST

功能: 绑定语音报警于事件, 一个通道有多个事件, 一个通道中的一个事件绑定一个语音。

接口 URI:/ voice

请求参数:

参数	类型	说明
videoNo	integer	视频通道编号, 1~16
enable	bool	是否启用该通道语音报警
events[]	array	各事件详细语音报警信息
{}	obj	一个事件语音绑定信息在此对象中, 多个事件将有多个对象
—eventType	integer	事件类型 1:人员越界 2:人员入侵 3:车辆越界 4:车辆禁停 5:车辆离开 6:人员徘徊 7:人员翻墙 8:烟火检测
—ruleID	integer	规则 ID 从 1 开始, 支持多少个配置规则, 就有多少个 ID
—voiceFileNo	integer	语音文件编号, 即 voiceFile 接口中导入的语音文件编号, 1~10
—enable	bool	是否启用该语音
—triggerTimes	integer	报警次数, 即触发时, 语音循环报警次数, 默认 3 次, 1~1000
—triggerInterval	integer	连续触发间隔, 单位 ms, 500~50000ms, 默认 1000ms
—priority	integer	触发优先级, 1-高优先级; 2-中优先级; 3-低优先级 当正在播放较低优先级的语音, 此时较高优先级的语音触发, 会立即停止较低优先级的语音报警播放, 转而播放较高优先级的语音报警; 相同优先级之间的语音报警, 顺序播放

29.2.2 语音报警删除 DELETE/ voice /{videoNo}

操作类型:

DELETE

功能: 解除语音报警与视频通道绑定关系

接口 URI: / voice /{videoNo}

请求参数:

{videoNo} 为视频通道编号, 1~16

返回参数:

无

29.2.3 查询语音报警信息 GET/ voice/{ videoNo }

操作类型:

GET

功能: 根据视频通道号获取语音报警信息。

接口 URI: / voice/{ videoNo }

请求参数:

{ videoNo } 为视频通道编号, 1~16

返回参数：

参数	类型	说明
enable	bool	该视频通道语音报警是否使能
events[]	array	该通道下各事件详细语音报警信息
{}	obj	一个事件语音报警绑定信息在此对象中，如果有多个事件将有多个对象
—eventType	integer	事件类型 1:人员越界 2:人员入侵 3:车辆越界 4:车辆禁停 5:车辆离开 6:人员徘徊 7:人员翻墙 8:烟火检测
—ruleID	integer	规则 ID 从 1 开始，支持多少个配置规则，就有多少个 ID
—voiceFileNo	integer	语音报警文件编号，即 voiceFile 接口中导入的语音文件编号，1~10
—enable	bool	是否启用该语音
—triggerTimes	integer	报警次数，即触发时，语音循环报警次数，默认 3 次，1~1000
—triggerInterval	integer	连续触发间隔，单位 ms，500~50000ms，默认 1000ms
—priority	integer	触发优先级，1-高优先级；2-中优先级；3-低优先级

29.2.4 取消语音报警 POST / voiceCancel /{ videoNo }

操作类型：

POST

功能：取消&停止某个视频通道下的所有已经触发的语音报警。

接口 URI: / voiceCancel /{ videoNo }

请求参数:

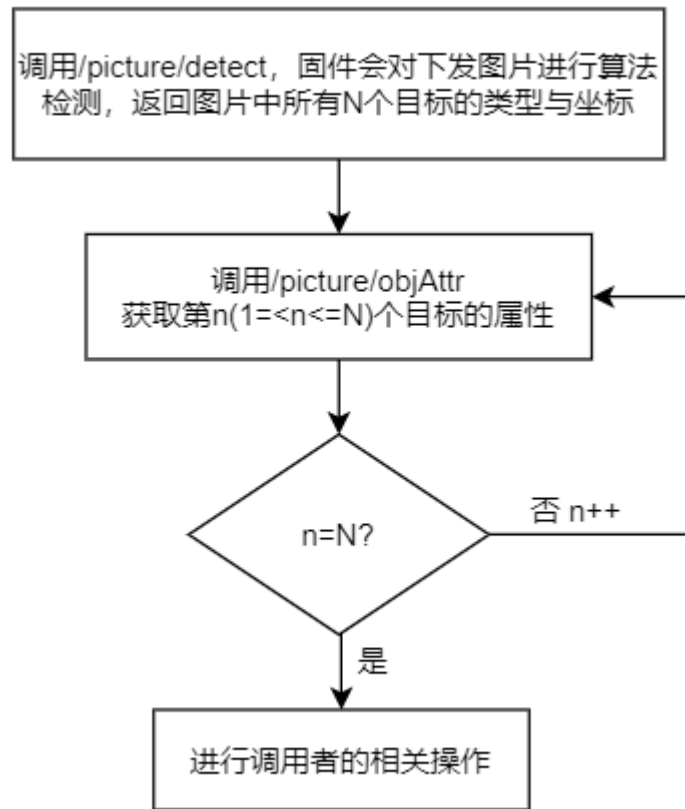
{ videoNo }为视频通道编号, 1~16

返回参数:

无

30 图片结构化检测

图片结构化检测, 要经过两个步骤, 如 30.1, 30.2, 具体流程如下图所示。



30.1 图片检测 POST /picture/detect

操作类型:

POST

功能:

对下发图片进行算法检测，返回图片中所有目标的坐标

照片限制：

1 图片类型：JPG、JPEG、PNG、BMP

1 图片大小 $\leq 4\text{MB}$

1 图片最大尺寸 $\leq 4096*4096$

1 图片最小尺寸 $\geq 100*100$

接口 URI：

/picture/detect

请求参数（form）：

参数	类型	说明	必选
image	binary	人脸图片	是

返回参数：

参数	类型	说明
detectionResult[]	obj	图片检测结果
-objType	string	目标类型

参数	类型	说明
		(1)人脸: "face" (2)人形: "human" (3)机动车: "vehicle", (4)非机动车: "cycle" (5)车牌: "plate"
—score	float	目标分值 (0~100)
—rect {}	obj	目标在图片上的位置框
— —left	integer	左边界坐标
— —top	integer	上边界坐标
— —right	integer	右边界坐标
— —bottom	integer	下边界坐标
—roll	integer	人脸水平角度 只有人脸携带
—yaw	integer	人脸垂直角度 只有人脸携带
—pitch	integer	有脸旋转角度 只有人脸携带
—blurness	float	图像清晰度 (0~100) 只有人脸携带

30.2 获取目标属性 POST /picture/objAttr

操作类型:

POST

功能:

获取某一目标的抠图数据、属性等信息

照片限制：

1 图片类型：JPG、JPEG、PNG、BMP

1 图片大小 $\leq 4\text{MB}$

1 图片最大尺寸 $\leq 4096*4096$

1 图片最小尺寸 $\geq 100*100$

接口 URI：

/picture/objAttr

请求参数（form）：

参数	类型	说明	必选
image	binary	图片	是
captureObjEnable	bool	是否回复目标抠图数据	是
objType	string	目标类型 (1)人脸：“face” (2)人形：“human” (3)机动车：“vehicle”， (4)非机动车：“cycle” (5)车牌：“plate”	
rect {}	obj	目标在图片上的位置框	是

参数	类型	说明	必选
—left	integer	左边界坐标	是
—top	integer	上边界坐标	是
—right	integer	右边界坐标	是
—bottom	integer	下边界坐标	是

返回参数：

参数	类型	说明
参数	类型	说明
captureObj	string	目标抠图 base64 数据，captureObjEnable 为 true 时携带
objType	string	目标类型，可取值如下，此值决定了接下来携带的目标属性字段 (1)人脸：“face” (2)人形：“human” (3)机动车：“vehicle”， (4)非机动车：“cycle” (5)车牌：“plate”
faceAttr{}	obj	人脸属性
—age	integer	年龄
—gender	integer	性别 0：未知 1：男 2：女
—skinColor	integer	肤色分类

参数	类型	说明
参数	类型	说明
		0: 无法判断 1: 黄皮肤 2: 白皮肤 3: 黑皮肤 4: 棕色皮肤（深色皮肤）
—hair	integer	发型 0: 发型未知 1: 光头 2: 少量头发（包含秃顶）3: 短发 4: 长发 5: 其他发型
—beard	integer	胡子类型 0: 胡子类型未知 1: 无胡子或者胡子不明显 2: 小胡子，嘴唇上面的胡子 3: 络腮胡 4: 其他类型
—hat {}	obj	帽子属性
— —hatWear	integer	0: 是否戴帽子未知 1: 戴帽子 2: 不戴帽子
— —hatColor	integer	16: 帽子颜色未知 17: 白色 18: 其他
— —hatType	integer	48: 安全帽 49: 厨师帽 50: 学生帽 51: 头盔 52: 安防全人种 01 小白帽 53: 头巾 54: 其他帽子 55: 未知类型
—respirator {}	obj	口罩属性
— —respiratorWear	integer	0: 是否戴口罩未知 1: 戴口罩 2: 不戴口罩
— —respiratorType	integer	16: 医用口罩 17: 雾霾口罩 18: 普通口罩 19: 未知类型 20: 厨房透明口罩
—glasses {}	obj	眼镜属性
— —glassLeftWear	integer	0: 左眼是否戴眼镜未知 1: 左眼戴眼镜 2: 左眼不戴眼镜

参数	类型	说明
参数	类型	说明
— —glassLeftType	integer	32: 左眼眼镜类型未知 33: 左眼深色框透明眼镜 34: 左眼普通透明眼镜 35: 左眼太阳墨镜 36: 左眼其他
— —glassRightWear	integer	16: 右眼是否戴眼镜未知 17: 右眼戴眼镜 18: 右眼不戴眼镜
— —glassRightType	integer	48: 右眼眼镜类型未知 49: 右眼深色框透明眼镜 50: 右眼普通透明眼镜 51: 右眼太阳墨镜 52: 右眼其他
humanAttr {}	obj	人员属性
—clothes {}	obj	上衣属性
— —clothesColor	integer	0: 上衣颜色未知 1: 黑色上衣 2: 白色上衣 3: 灰色上衣 4: 红色上衣 5: 橙色上衣 6: 黄色上衣 7: 绿色上衣 8: 深蓝色上衣 9: 浅蓝色上衣 10: 紫色上衣 11: 粉红色上衣 12: 棕色上衣 13: 彩色上衣
— —clothesSleeve	integer	32: 上衣长短袖未知 33: 上衣长袖 34: 上衣短袖
—pants {}	obj	下衣属性
— —pantsColor	integer	0: 颜色未知 1: 黑色裤子 2: 白色裤子 3: 灰色裤子 4: 红色裤子 5: 橙色裤子 6: 黄色裤子 7: 绿色裤子 8: 深蓝色裤子 9: 浅蓝色裤子 10: 紫色裤子 11: 粉红色裤子 12: 棕色裤子

参数	类型	说明
参数	类型	说明
		13: 彩色裤子
— —pantsLen	integer	32: 长短裤未知 33: 长裤 34: 短裤
— —pantsSkirt	integer	48: 穿裙子未知 49: 穿裙子 50: 不穿裙子
—bags {}	obj	背包属性
— —bagCarry	integer	0: 是否背包未知 1: 背包 2: 不背包
— —bagType	integer	按位使能, 可同时使能多位 0-不使能 1-使能 第 0 位: 包类型未知 第 1 位: 双肩背包 第 2 位: 单肩包、斜挎包 第 3 位: 手提包 第 4 位: 拉杆箱
—rideBike (废弃)	integer	骑车状态 0: 是否骑车未知 1: 骑车 2: 不骑车
—umbrella	integer	打伞状态 0: 是否打伞未知 1: 打伞 2: 不打伞

参数	类型	说明
参数	类型	说明
—gender	integer	性别 0: 性别未知 1: 男性 2: 女性
—hair	integer	发型 0: 发型未知 1: 光头（预留字段） 2: 很少头发（包含秃顶）（预留字段） 3: 短头发 4: 长头发 5: 其他发型
—hat {}	obj	帽子属性
— —hatWear	integer	0: 是否带帽子未知 1: 带帽子 2: 不戴帽子
— —hatColor（预留字段）	integer	16: 帽子颜色未知 17: 白色 18: 其他
— —hatType	integer	48: 安全帽（预留字段） 49: 厨师帽（预留字段） 50: 学生帽（预留字段） 51: 头盔 52: 安防全人种 01 小白帽（预留字段） 53: 头巾（预留字段） 54: 其他帽子 55: 未知类型
—safetyHelmet {}	obj	安全帽属性
— —safetyHelmetWear	integer	0: 是否带帽子未知 1: 带帽子 2: 不戴帽子
— —safetyHelmetColor（预留字段）	integer	16: 帽子颜色未知 17: 白色 18: 其他

参数	类型	说明
参数	类型	说明
— —safetyHelmetType	integer	48: 安全帽 49: 厨师帽（预留字段） 50: 学生帽（预留字段） 51: 头盔（预留字段） 52: 安防全人种 01 小白帽（预留字段） 53: 头巾（预留字段） 54: 其他帽子（预留字段） 55: 未知类型（预留字段）
—action{}（预留字段）	obj	动作属性
— —actionTumble	integer	0: 是否跌倒未知 1: 跌倒 2: 没有跌倒
— —actionSmoke	integer	16: 是否吸烟未知 17: 吸烟 18: 没有吸烟
— —actionRun	integer	32: 是否跑步未知 33: 跑步 34: 没有跑步
— —actionCalling	integer	48: 是否打电话未知 49: 打电话 50: 没有打电话
— —actionWatchPhone	integer	64: 是否看手机未知 65: 看手机 66: 没有看手机
vehicleAttr{}	obj	机动车属性
—subClass	integer	机动车细分类 0: 未知 1: 小轿车 2: SUV 3: 面包车 4: 中巴车和大巴车 5: 皮卡车 6: 卡车 7: 其他
—brand	integer	[0~2461] 品牌索引编号 如 0: 含义是“保时捷_Cayenne_06 年款”
—color	integer	0x1FFF: 未知 0x2000: 黑色 0x2001: 白色 0x2002: 深灰 0x2003: 棕色 0x2004: 红色 0x2005: 蓝色 0x2006: 黄色 0x2007: 绿色 0x2008: 花色 0x2009: 银灰 0x200A: 橙色 0x200B: 青色 0x200C: 粉色 0x200D: 紫色 0x200E: 香槟

参数	类型	说明
参数	类型	说明
—orientation	integer	0x2020：正向 0x2021：背向 0x2022：其他
—brandname	string	机动车属性信息，如：“保时捷_Cayenne_06 年款”
cycleAttr {}	obj	非机动车属性
—subClass	integer	非机动车细分类 0：未知 1：自行车 2：电动车 3：摩托车 4：三轮车 5：其他
—bindingCount	integer	非机动车绑定人形的个数
plateAttr {}	obj	车牌属性
—type	integer	车牌类型 0：单行 1：双行 2：未知
—color	integer	车牌颜色 0：蓝色 1：黄色 2：黑色 3：白色 4：绿色 5：小型新能源 6：大型新能源 7：缺省
—num	string	车牌号码