|  |
| --- |
| 版权所有内部私密文件 |
| 摄像头协议集成开发指南 |
| 20210702 |
| **v0.3c** |
| **2021/7/2** |

本

**协议集成开发指南**

|  |  |  |  |
| --- | --- | --- | --- |
| 日期 | 版本 | 修订者 | 修订记录 |
| 2021.06.02 | V0.3c | yangxingyi | 新增3.6.24电梯状态检测api  扩展3.6.2获取智能能力集支持电梯状态检测字段  扩展3.6.3智能报警事件联动支持电梯状态检测联动 |
| 2021.05.21 | V0.3b | yuliang | 高抛检测接口[frmFallBasicParam](#_高抛检测基本配置)(高抛检测基本配置)增加Linkage(事件联动配置). |
| 2021.05.14 | V0.3a | yuliang | 修改含有密码参数的部分接口,比如[3.1.19用户信息](#_用户信息)(frmUserManage),增加"PasswordB64",从而扩展密码字符集以支持特殊字符.  增加高抛检测功能接口([3.6.23高抛检测](#_高抛检测)),frmFallVer(获取版本信息),frmFallBasicParam(高抛基础参数如使能,检测区域/屏蔽区域,楼层标定),frmFallAdvParam(高抛算法参数). |
| 2020.03.09 | V0.2v | yangxingyi | 3.2.22修改标题为4G/5G配置，  3.2.22.1获取4G/5G配置新增CCID,DNS1V4,DNS2V4,DNS1V6,DNS2V6字段  新增3.2.22.2获取4G/5G配置简易接口frmGetLteCardInfo  新增3.2.28.3阿里IoT服务重启接口frmGetLteCardInfo  新增3.2.28.4阿里IoT解绑接口frmGetLteCardInfo  新增3.2.29获取默认路由接口frmGetDefaultRoute |
| 2021.03.03 | V0.2u | humengmeng | 完善(新增)3.3.10.3 接口frmVideoParaEx中视频参数中关于对焦区域的几个字段，之前功能已添加，文档未同步，现已同步; 新增3.8.6 人脸操作状态码对应请求结果 |
| 2020.12.18 | V0.2t | humengmeng | 修改了三处分别是:  1)3.3.10.3视频参数配置接口frmVideoParaEx中：LightType: "Ir\_Warm"/"Ir\_White"-双光源字段中删掉.  2)视频参数部分[灯板光源可以有哪些种类?]说明文字中[4.变倍组合光源]的SupportLights的值LowBeam等三个字段中的小写b改为大写B.  3)3.1.11自动维护接口frmAutoReboot中，Month.time的说明文字做了微调. |
| 2020.12.10 | V0.2s | Yangxingyi | 3.1.5获取设备信息接口frmDevicePara扩展"DeviceTypeString"字段描述,新增警戒机细分类型:电梯警戒机,人形警戒机,车形警戒机  3.3.3音频参数接口删除多余字段定义  3.5.3遮挡报警报警接口订正灵敏度字段为"Sensitivity"  3.7.9格式化磁盘接口订正为frmHDFormat |
| 2020.12.02 | V0.2r | Yangxingyi | 3.1.5获取设备信息接口frmDevicePara新增"FunctionInfo.EnableAlarming"字段,标识设备是否支持布防撤防功能 |
| 2020.11.19 | V0.2q | humengmeng | 1.新增接口有:  (1)新增3.4.4.14智能跟踪配置接口frmSmartDomeTrack;  (2) 新增3.4.4.15手动追踪配置接口frmPTZDomeManualTrack;  (3) 新增3.4.4.16球机定位配置接口frmPTZDomeLocation;  2.订正视频参数配置中的几个错误：  （1）wdrType:宽动态模式，,0-关闭,2-数字宽动态,3-真宽动态.  （2）"LightCorrectMode": Int型,背光模式,0-关闭, 1-强光抑制,2-背光补偿.  （3）"LightCorrectLevel": Int型,背光模式校正强度,范围1~10. |
| 2020.11.07 | V0.2p | Yangxingyi | 新增3.2.28.1阿里云IoT配置接口[frmAliIoTCfg](#_阿里IoT配置)  新增3.2.28.2阿里云IoT服务状态获取接口[frmAliIoTState](#_获取阿里IoT服务状态) |
| 2020.10.16 | V0.2o | Yangxingyi | 1. 更正智能能力集字段说明 2. 电动车检测相似度改为灵敏度 3. 音频报警输出api加上AudioDataType字段说明 |
| 2020.09.15 | V0.2n | maou | (1)修改NTP时间接口[frmNetNtpPara](#_NTP) 添加CurrentTimeOffset字段用来计算与UTC(GMT0)时间的偏移(单位:秒,包含夏令时).  (2)新增接口[frmOnvifPara](#_ONVIF(全网通和免验证)_1),表示onvif的全网通和免验证配置.  (3)在接口[frmVideoParaEx](#_视频参数配置) 新增Type类型用来区分全天/白天/黑夜,新增接口[frmVideoParaExMode](#_视频参数配置切换策略_1)，用来切换配置策略,以便支持白天夜晚的ISP参数分别设置的功能. |
| 20200908 | V0.2m | Yangxingyi | 3.5.6加上获取布防状态的说明 |
| 20200907 | V0.2l | wanghaifeng | frmDeviceParam扩展语义DeviceTypeString和FunctionInfo,表示设备类型和集成功能的能力集  frmFaceCompare修正StatusCode = 1的释义为查询未完成 |
| 20200902 | V0.2k | Yangxingyi | 新增3.6.22电动车检测API,扩展智能能力集,扩展智能报警事件联动 |
| 20200706 | V0.2j | humengmeng | 修改如下:1.已有API新增字段,有:(1)3.2.6RTSP流地址获取frmGetRtspUrl,当Type为3和Type为13时,增加TimeMode字段,用来解析传进来的StartTime/StopTime;(2)3.2.18 GB28181新增streamType字段,支持五种码流类型;(3)3.8.3.1获取查询句柄,新增Type=9,按照温度和口罩状态查询.  2.新增API:新增3.11.3测温设置的API为frmVideoTemperatureCfg |
| 20200615 | V0.2i | humengmeng | 新增3.5.6布防撤防的API,frmGetAlarmInfo,说明了Type设置的类型 |
| 20200604 | V0.2h | humengmeng | 订正了三处误写:分别是:(1)修改了3.6.4声光报警联动项布防时间的获取和设置的API,现订正为frmActionAlarmTimePara,(2)修改了3.4.3.8两点扫描的API,现订正为frmPTZExtend\_SetScan,(3)3.1.13.3获取升级进度的API,现订正为GetProgress |
| 2020.05.29 | V0.2g | humengmeng | 新增3.11热成像相关API,增加了3.11.1口罩检测和3.11.2人体测温配置,共两个接口.  修改了3.4.2报警输出，增加Type类型,为2时表获取报警输出开关状态,为3时表开启报警输出,为4时表关闭报警输出. |
| 2020.05.09 | V0.2f | humengmeng | 修改3.2.3.1HTTP推送,增加用户名,密码,验证方式三个配置项的定义. |
| 2020.04.29 | V0.2e | humengmeng | 修改[3.3.10.3视频参数配置](#_视频参数配置),增加"ExposureLight"."Mode"(即红外补光)的定义. |
| 2020.04.28 | V0.2d | Yuliang | 修改[3.3.10.1视频参数能力集](#_视频参数能力集),增加灯板光源的能力定义.修改[3.3.10.3视频参数配置](#_视频参数配置)中"LightType"的定义. |
| 2020.04.15 | V0.2c | Yuliang | 修改[3.10.3视频参数配置](#_视频参数配置),接口frmVideoParaEx中参数项"LightType"的可选值,增加3-双光源.  修改报警联动中的词语,将"声光报警"改为"灯光告警",涉及[3.4.1报警输入](#_报警输入),[3.5.2移动侦测](#_移动侦测),[3.5.3遮挡报警](#_遮挡报警),[3.5.5异常报警](#_异常报警),[3.6.3智能报警事件联动](#_智能报警事件联动). |
| 2020.04.08 | V0.2b | Yangxingyi | Rtsp流地址获取标准地址和私有地址单独列出,录像查询api更新 |
| 2019.12.23 | V0.2a | Wangzhiyong | 添加http应用协议配置说明 |
| 2019.12.18 | V0.2 | Yangxingyi,Humengmeng | 补充协议 |
| 2019.11.15 | V0.1a | Yuliang | 整理协议 |
| 2019.11.13 | V0.1 | Yuliang | 拟定初稿 |

**目录**

[1 文档概要 12](#_Toc66195142)

[1.1 文档范围 12](#_Toc66195143)

[1.2 参考文档 12](#_Toc66195144)

[1.3 基本概念 12](#_Toc66195145)

[2 设计说明 14](#_Toc66195146)

[2.1 交互过程 14](#_Toc66195147)

[2.2 HTTP命令的基本结构 15](#_Toc66195148)

[2.3 HTTP命令的请求行和响应行 16](#_Toc66195149)

[2.4 HTTP命令的正文格式 16](#_Toc66195150)

[2.5 URL 19](#_Toc66195151)

[3 接口说明 26](#_Toc66195152)

[3.1 系统 26](#_Toc66195153)

[3.1.1 登录 26](#_Toc66195154)

[3.1.2 保持心跳 27](#_Toc66195155)

[3.1.3 登出 28](#_Toc66195156)

[3.1.4 系统信息 28](#_Toc66195157)

[3.1.5 设备信息 30](#_Toc66195158)

[3.1.6 设备API版本获取 32](#_Toc66195159)

[3.1.7 时间 33](#_Toc66195160)

[3.1.8 夏令时 34](#_Toc66195161)

[3.1.9 恢复默认 36](#_Toc66195162)

[3.1.10 重启 37](#_Toc66195163)

[3.1.11 自动维护 38](#_Toc66195164)

[3.1.12 黑白名单 42](#_Toc66195165)

[3.1.13 升级 47](#_Toc66195166)

[3.1.13.1 获取升级信息 47](#_Toc66195167)

[3.1.13.2 上传文件并升级 47](#_Toc66195168)

[3.1.13.3 获取升级进度 48](#_Toc66195169)

[3.1.14 日志 48](#_Toc66195170)

[3.1.15 配置文件导入 50](#_Toc66195171)

[3.1.16 配置文件导出 51](#_Toc66195172)

[3.1.17 二维码 52](#_Toc66195173)

[3.1.18 本地设置 55](#_Toc66195174)

[3.1.19 用户信息 57](#_Toc66195175)

[3.1.20 用户权限 59](#_Toc66195176)

[3.1.21 在线用户 64](#_Toc66195177)

[3.1.22 在线用户数 65](#_Toc66195178)

[3.1.23 密码重置 66](#_Toc66195179)

[3.2 网络 68](#_Toc66195180)

[3.2.1 网络接口配置 68](#_Toc66195181)

[3.2.2 HTTP/HTTPS 71](#_Toc66195182)

[3.2.3 HTTP推送 72](#_Toc66195183)

[3.2.3.1 HTTP推送配置 72](#_Toc66195184)

[3.2.3.2 HTTP推送服务器地址测试 74](#_Toc66195185)

[3.2.4 HTTP事件 75](#_Toc66195186)

[3.2.5 RTSP 77](#_Toc66195187)

[3.2.6 RTSP流地址获取 78](#_Toc66195188)

[3.2.7 RTMP 82](#_Toc66195189)

[3.2.8 多播 84](#_Toc66195190)

[3.2.9 DDNS 86](#_Toc66195191)

[3.2.9.1 获取DDNS服务器列表 86](#_Toc66195192)

[3.2.9.2 DDNS配置 87](#_Toc66195193)

[3.2.10 UPNP 89](#_Toc66195194)

[3.2.11 FTP 91](#_Toc66195195)

[3.2.12 EMAIL 93](#_Toc66195196)

[3.2.13 SNMP 96](#_Toc66195197)

[3.2.14 NAS 97](#_Toc66195198)

[3.2.15 平台管理 99](#_Toc66195199)

[3.2.16 P2P 102](#_Toc66195200)

[3.2.16.1 P2P配置 102](#_Toc66195201)

[3.2.16.2 获取P2P服务在线状态 103](#_Toc66195202)

[3.2.17 TURN服务 104](#_Toc66195203)

[3.2.17.1 TURN服务配置 104](#_Toc66195204)

[3.2.17.2 获取TURN服务状态 106](#_Toc66195205)

[3.2.18 GB28181 106](#_Toc66195206)

[3.2.19 GB35114 110](#_Toc66195207)

[3.2.20 韦根 114](#_Toc66195208)

[3.2.21 Smart1400 115](#_Toc66195209)

[3.2.21.1 Smart1400配置 115](#_Toc66195210)

[3.2.21.2 Smart1400服务器地址测试 117](#_Toc66195211)

[3.2.22 4G/5G 118](#_Toc66195212)

[3.2.22.1 4G/5G配置 118](#_Toc66195213)

[3.2.22.2 获取4G/5G配置简易接口 120](#_Toc66195214)

[3.2.23 WIFI 120](#_Toc66195215)

[3.2.24 SIP 126](#_Toc66195216)

[3.2.24.1 SIP配置 126](#_Toc66195217)

[3.2.24.2 手动触发Sip报警输出 130](#_Toc66195218)

[3.2.25 NTP 130](#_Toc66195219)

[3.2.26 TELNET 132](#_Toc66195220)

[3.2.27 ONVIF(全网通和免验证) 133](#_Toc66195221)

[3.2.28 阿里IoT 135](#_Toc66195222)

[3.2.28.1 阿里IoT配置 135](#_Toc66195223)

[3.2.28.2 获取阿里IoT服务状态 136](#_Toc66195224)

[3.2.28.3 阿里IoT服务重启 137](#_Toc66195225)

[3.2.28.4 阿里IoT解绑 137](#_Toc66195226)

[3.2.29 获取默认路由 138](#_Toc66195227)

[3.3 媒体 139](#_Toc66195228)

[3.3.1 视频制式 139](#_Toc66195229)

[3.3.1.1 获取视频制式能力集 139](#_Toc66195230)

[3.3.1.2 视频制式 140](#_Toc66195231)

[3.3.1.3 视频制式\_V2 141](#_Toc66195232)

[3.3.2 音频报警输出 142](#_Toc66195233)

[3.3.2.1 音频报警输出配置 142](#_Toc66195234)

[3.3.2.2音频报警试听 144](#_Toc66195235)

[3.3.3 音频参数 145](#_Toc66195236)

[3.3.3.1 获取音频参数能力集 145](#_Toc66195237)

[3.3.3.2 音频参数配置 146](#_Toc66195238)

[3.3.4 图像模式 148](#_Toc66195239)

[3.3.4.1 获取图像模式能力集 148](#_Toc66195240)

[3.3.4.2 图像模式配置 148](#_Toc66195241)

[3.3.5 视频编码参数 150](#_Toc66195242)

[3.3.5.1 获取视频编码能力集 150](#_Toc66195243)

[3.3.5.2 视频编码参数配置 152](#_Toc66195244)

[3.3.6 单行OSD 154](#_Toc66195245)

[3.3.6.1 单行OSD(带点阵) 154](#_Toc66195246)

[3.3.6.2 单行OSD(不带点阵) 157](#_Toc66195247)

[3.3.7 多行OSD 160](#_Toc66195248)

[3.3.7.1 多行OSD(带点阵) 160](#_Toc66195249)

[3.3.7.2 多行OSD(不带点阵) 163](#_Toc66195250)

[3.3.8 图像效果 165](#_Toc66195251)

[3.3.9 强制I帧 166](#_Toc66195252)

[3.3.10 视频参数 167](#_Toc66195253)

[3.3.10.1 视频参数能力集 167](#_Toc66195254)

[3.3.10.2 场景抓拍 168](#_Toc66195255)

[3.3.10.3 视频参数配置 170](#_Toc66195256)

[3.3.10.4 视频参数配置切换策略 176](#_Toc66195257)

[3.3.11 ROI配置 178](#_Toc66195258)

[3.3.12 抓图 180](#_Toc66195259)

[3.4 物理接口 180](#_Toc66195260)

[3.4.1 报警输入 180](#_Toc66195261)

[3.4.2 报警输出 183](#_Toc66195262)

[3.4.3 球机 185](#_Toc66195263)

[3.4.3.1 获取PTZ协议能力集 185](#_Toc66195264)

[3.4.3.2 编码参数 186](#_Toc66195265)

[3.4.3.3 云台控制 188](#_Toc66195266)

[3.4.3.4 预置点 189](#_Toc66195267)

[3.4.3.5 巡航线配置 189](#_Toc66195268)

[3.4.3.6 巡航线控制 192](#_Toc66195269)

[3.4.3.7 轨迹控制 192](#_Toc66195270)

[3.4.3.8 两点扫描 194](#_Toc66195271)

[3.4.3.9 空闲操作 194](#_Toc66195272)

[3.4.3.10 红外补光 197](#_Toc66195273)

[3.4.3.11 隐私遮蔽 198](#_Toc66195274)

[3.4.3.12 3D定位 200](#_Toc66195275)

[3.4.3.13 雨刷 201](#_Toc66195276)

[3.4.3.14 喷淋位置 202](#_Toc66195277)

[3.4.3.15 喷淋模式 202](#_Toc66195278)

[3.4.4 网络球机 203](#_Toc66195279)

[3.4.4.1 获取网络球机能力集 203](#_Toc66195280)

[3.4.4.2 获取协议列表 204](#_Toc66195281)

[3.4.4.3 UART配置 205](#_Toc66195282)

[3.4.4.4 云台控制 207](#_Toc66195283)

[3.4.4.5 预置点控制 207](#_Toc66195284)

[3.4.4.6 巡航线配置 208](#_Toc66195285)

[3.4.4.7 巡航线控制 210](#_Toc66195286)

[3.4.4.8 轨迹控制 211](#_Toc66195287)

[3.4.4.9 看守配置 212](#_Toc66195288)

[3.4.4.10 看守控制 213](#_Toc66195289)

[3.4.4.11 断电记忆 214](#_Toc66195290)

[3.4.4.12 计划任务 215](#_Toc66195291)

[3.4.4.13 OSD位置 218](#_Toc66195292)

[3.4.4.14 智能跟踪 219](#_Toc66195293)

[3.4.4.15 手动追踪 221](#_Toc66195294)

[3.4.4.16 球机定位 222](#_Toc66195295)

[3.4.5 鱼眼相机 223](#_Toc66195296)

[3.4.5.1 获取鱼眼相机能力集 223](#_Toc66195297)

[3.4.5.2 鱼眼模式配置 224](#_Toc66195298)

[3.4.5.3 获取鱼眼通道位置 226](#_Toc66195299)

[3.4.5.4 云台控制 227](#_Toc66195300)

[3.4.5.5 预置点 228](#_Toc66195301)

[3.4.5.6 巡航线配置 229](#_Toc66195302)

[3.4.5.7 巡航线控制 232](#_Toc66195303)

[3.5 报警事件 234](#_Toc66195304)

[3.5.1 报警事件联动 234](#_Toc66195305)

[3.5.2 移动侦测 235](#_Toc66195306)

[3.5.3 遮挡报警 238](#_Toc66195307)

[3.5.4 隐私遮蔽 241](#_Toc66195308)

[3.5.5 异常报警 243](#_Toc66195309)

[3.5.6 布防撤防 244](#_Toc66195310)

[3.6 智能报警 246](#_Toc66195311)

[3.6.1 报警信息查询 246](#_Toc66195312)

[3.6.2 获取智能能力集 247](#_Toc66195313)

[3.6.3 智能报警事件联动 248](#_Toc66195314)

[3.6.4 声光报警联动项布防时间 252](#_Toc66195315)

[3.6.5 目标计数 254](#_Toc66195316)

[3.6.6 物品检测 258](#_Toc66195317)

[3.6.7 区域检测(电子围栏) 261](#_Toc66195318)

[3.6.8 声音告警 263](#_Toc66195319)

[3.6.9 虚拟警戒线 265](#_Toc66195320)

[3.6.10 智能移动侦测 268](#_Toc66195321)

[3.6.11 视频诊断 270](#_Toc66195322)

[3.6.12 火灾检测 272](#_Toc66195323)

[3.6.13 车牌识别 274](#_Toc66195324)

[3.6.14 人脸检测 275](#_Toc66195325)

[3.6.15 逆行检测 281](#_Toc66195326)

[3.6.16 离岗检测 284](#_Toc66195327)

[3.6.17 密度检测 287](#_Toc66195328)

[3.6.18 限高检测 289](#_Toc66195329)

[3.6.19 场景变换 292](#_Toc66195330)

[3.6.20 人形检测 293](#_Toc66195331)

[3.6.21 安全帽检测 299](#_Toc66195332)

[3.6.22 电动车检测 302](#_Toc66195333)

[3.7 录像 305](#_Toc66195334)

[3.7.1 录像计划配置 305](#_Toc66195335)

[3.7.2 视频断开上报 307](#_Toc66195336)

[3.7.3 水印配置 308](#_Toc66195337)

[3.7.4 录像查询 310](#_Toc66195338)

[3.7.5 按月查询录像 311](#_Toc66195339)

[3.7.6 SD卡联动抓图查询 312](#_Toc66195340)

[3.7.7 磁盘信息 312](#_Toc66195341)

[3.7.8 磁盘强制格式化检查 313](#_Toc66195342)

[3.7.9 格式化磁盘 314](#_Toc66195343)

[3.7.10 获取磁盘格式化进度 315](#_Toc66195344)

[3.8 智能人脸 316](#_Toc66195345)

[3.8.1 人脸数据库操作 316](#_Toc66195346)

[3.8.1.1 添加人脸数据库 316](#_Toc66195347)

[3.8.1.2 删除人脸数据库 316](#_Toc66195348)

[3.8.1.3 修改人脸数据库 317](#_Toc66195349)

[3.8.1.4 清空人脸数据库 318](#_Toc66195350)

[3.8.1.5 获取人脸数据库数目 318](#_Toc66195351)

[3.8.1.6 获取人脸数据库 319](#_Toc66195352)

[3.8.1.7 获取人脸库绑定信息 320](#_Toc66195353)

[3.8.1.8 设置人脸库绑定信息 321](#_Toc66195354)

[3.8.1.9 设置人脸库版本号 322](#_Toc66195355)

[3.8.2 人脸模板操作 322](#_Toc66195356)

[3.8.2.1 人脸模板添加 322](#_Toc66195357)

[3.8.2.2 人脸模板删除 324](#_Toc66195358)

[3.8.2.3 人脸模板修改 324](#_Toc66195359)

[3.8.2.4 人脸模板查询 325](#_Toc66195360)

[3.8.3 人脸比对查询 329](#_Toc66195361)

[3.8.3.1 获取查询句柄 329](#_Toc66195362)

[3.8.3.2 获取查询进度 332](#_Toc66195363)

[3.8.3.3 获取查询结果 332](#_Toc66195364)

[3.8.3.4 查询句柄释放 334](#_Toc66195365)

[3.8.4 人脸库算法版本获取 334](#_Toc66195366)

[3.8.5 人脸高级配置 335](#_Toc66195367)

[3.8.6 人脸操作状态码 338](#_Toc66195368)

[3.9 智能结果统计 339](#_Toc66195369)

[3.9.1 获取查询句柄 339](#_Toc66195370)

[3.9.2 获取查询结果 340](#_Toc66195371)

[3.9.3 释放查询句柄 342](#_Toc66195372)

[3.10 光学 343](#_Toc66195373)

[3.10.1 自动镜头校正 343](#_Toc66195374)

[3.10.2 自定义镜头参数 344](#_Toc66195375)

[3.11 热成像 346](#_Toc66195376)

[3.11.1 口罩检测 346](#_Toc66195377)

[3.11.2 人体测温配置 347](#_Toc66195378)

[3.11.3 测温设置 349](#_Toc66195379)

# 文档概要

## 文档范围

I8H协议是一种定义于HTTP之上的API接口. 它为工作于IP网络中的视频安防设备和管理系统,提供了一种交互机制.

本协议文档专注于I8H协议之IPC设备的通信命令的描述,而媒体数据流的交互则采用RTSP协议.

## 参考文档

HTTP参考.

RTSP参考.

JSON参考.

UTF-8参考.

## 基本概念

HTTP

API

RTSP

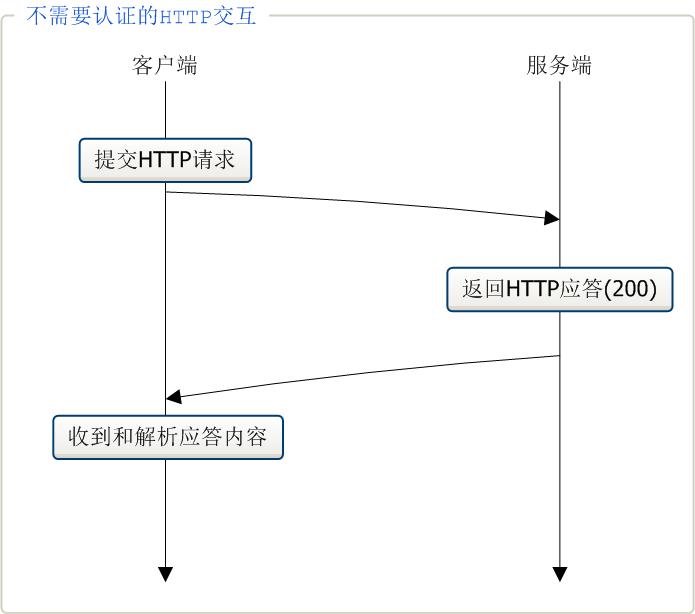
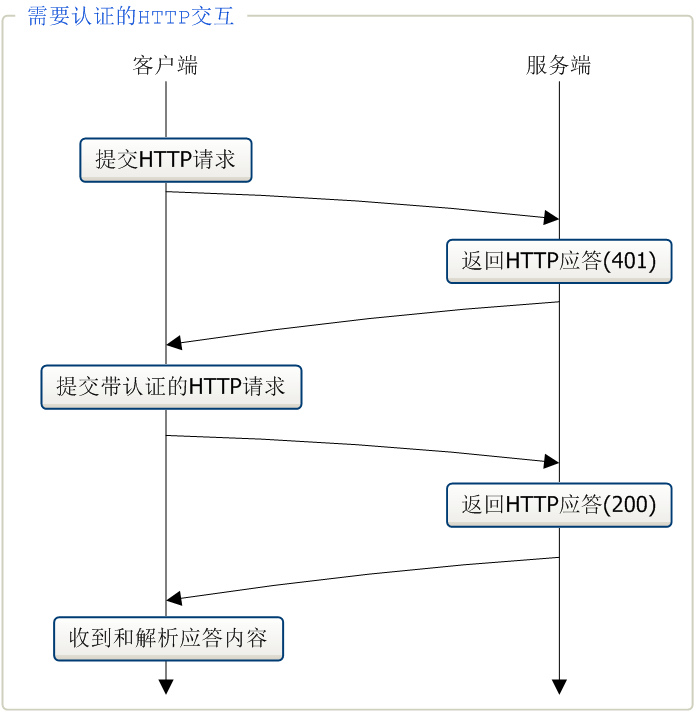
JSON

URL

# 设计说明

## 交互过程

I8H命令的一次HTTP交互过程,可能会有两种情况,(1)需要认证的HTTP交互,(2)不需要认证的HTTP交互.下面的图中描述了这两种情况的交互过程.



对于需要认证的HTTP交互,客户端在提交HTTP请求后,服务端会返回401 Unauthorized的应答,并且头部包含WWW-Authenticate项,表示客户端需要提供认证信息. HTTP头部的WWW-Authenticate项示例如下.

|  |
| --- |
| WWW-Authenticate: Digest realm="webserver", domain="::", qop="auth", nonce="ZmFkODI1Y2ZmZTQwYTM3MDJhZTRmMWI1ZWE5NTRiZWY6d2Vic2VydmVyOjVkZDQwODg3OjExNw==", opaque="5ccc069c403ebaf9f0171e9517f40e41", algorithm="MD5", stale="FALSE" |

客户端根据返回的WWW-Authenticate和用户名密码计算Authorization.在客户端发送新的请求中,头部应包含Authorization项,如果认证正确则服务端会返回200的应答. HTTP头部的Authorization的示例如下.

|  |
| --- |
| Authorization: Digest username="admin", realm="webserver", nonce="ZmFkODI1Y2ZmZTQwYTM3MDJhZTRmMWI1ZWE5NTRiZWY6d2Vic2VydmVyOjVkZDQwODg3OjExOQ==", uri="digest/frmDstPara", response="f84a841fef42be86c925b346324eba77", opaque="5ccc069c403ebaf9f0171e9517f40e41", qop=auth, nc=00000002, cnonce="dff275c9dd027985" |

## HTTP命令的基本结构

本协议定义的HTTP报文的基本结构分两种,HTTP请求和HTTP响应.

HTTP请求包含请求行,请求头部和请求正文.HTTP响应包含响应行,响应头部和响应正文.

请求行和请求头部都遵循**HTTP规范**,请求正文则是采用**JSON编码格式**的文本.

下文中请求正文和响应正文的内容以表格形式列出,但注意在HTTP正文均为JSON格式的文本.

示例如下表.

|  |  |
| --- | --- |
| 请求行 | POST/digest/frmDevicePara HTTP/1.1\r\n |
| 请求头部 | Accept: \*/\*  Content-Type: application/json  CV-SESSION-ID: 0  Referer: http://10.5.1.19/  Accept-Language: zh-CN  Accept-Encoding: gzip, deflate  User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko  Host: 10.5.1.19  Content-Length: 35  Connection: Keep-Alive  Cache-Control: no-cache  Cookie: plugin\_name=null; last\_login\_username=admin |
| 请求正文 | {"Type":0,"Dev":1,"Ch":1,"Data":{}} |

|  |  |
| --- | --- |
| 响应行 | HTTP/1.1 200 OK\r\n |
| 响应头部 | Server: nginx  Date: Tue, 19 Nov 2019 15:21:35 GMT  Content-Type: application/json  Transfer-Encoding: chunked  Connection: keep-alive  Cache-Control: no-cache  Pragma: no-cache  Expires: -1  X-Frame-Options: SAMEORIGIN  X-Content-Type-Options: nosniff  X-XSS-Protection: 1; mode=block |
| 响应正文 | {"Result":0,"Data":{"WebApiVersion":{"Standard":"V1.5.5","Build":"2019-11-19 10:14:04"},"FaceDetectNum":0,"SoftwareVersion":"V4.3.20191029","HardwareVersion":"V4.0","SerialNumber":"0bdc00020002a3fbc387","DVRType":3,"CustomerSN":"","DVRName":"","AlarmInPortNum":1,"AudioSource":0,"AudioInVol":50,"AudioOutVol":50}} |

## HTTP命令的请求行和响应行

在本协议中,采用的HTTP方法为**POST**,以及**HTTP/1.1**.

请求行的示例如下,其中"/digest/frmDevicePara"为协议的**URL**(Uniform Resource Locator).

|  |  |
| --- | --- |
| POST /digest/frmDevicePara HTTP/1.1\r\n |  |

响应行的示例如下,分别为成功和失败的情况.

|  |  |
| --- | --- |
| HTTP/1.1 200 OK\r\n |  |
| HTTP/1.1 401 OK\r\n |  |

## HTTP命令的正文格式

正文格式采用JSON语法和UTF-8字符集.

HTTP请求的正文格式,如下:

|  |  |
| --- | --- |
| **HTTP请求的正文** | 说明 |
| **{**  **"Dev": 0,**  **"Ch": 0,**  **"Type": 0,**  **"Data": {**  **}**  **}** | "Dev":设备号. 从1开始.对于IPC来说,可用于区分多个VI设备(可以理解为多个sensor).通常的IPC为单个VI,不必包含此字段.  "Ch":通道号.从1开始. 对于IPC/NVR来说,可用于区分视频通道或报警通道.  "Type":操作类型. 0表示查询参数,1表示设置参数.其它数字在特定API时才会有意义.  "Data":操作数据. 比如需要设置的参数及其值. |

若成功,HTTP响应的正文格式,如下:

|  |  |
| --- | --- |
| **HTTP响应成功的正文** | 说明 |
| **{**  **"Result": 0,**  **"Data": {**  **}**  **}** | "Result":返回的结果值. 0表示正确.其余负数表示错误.  "Data":返回的结果数据. 比如查询到的参数及其值. |

若失败,HTTP响应成功的正文格式,如下:

|  |  |
| --- | --- |
| **HTTP响应失败的正文** | 说明 |
| **{**  **"Result": -1,**  **"ErrorString": "Operation Failed!"**  **}** | "Result":返回的结果值. 0表示正确.其余负数表示错误.  "ErrorString":操作失败时的提示信息. |

**\* 接入http应用协议时**

HTTP请求的正文格式,如下:

|  |  |
| --- | --- |
| **HTTP请求的正文** | 说明 |
| **{**  *"Command" : "xxxxxx",*  *"CommandSeq" : n,*  **"Dev": 0,**  **"Ch": 0,**  **"Type": 0,**  **"Data": {**  **}**  **}** | *“Command”: 该次请求接口名称见 2.6*  *“CommandSeq”:该次请求seq 标识前请求唯一*  "Dev":设备号. 从1开始.对于IPC来说,可用于区分多个VI设备(可以理解为多个sensor).通常的IPC为单个VI,不必包含此字段.  "Ch":通道号.从1开始. 对于IPC/NVR来说,可用于区分视频通道或报警通道.  "Type":操作类型. 0表示查询参数,1表示设置参数.其它数字在特定API时才会有意义.  "Data":操作数据. 比如需要设置的参数及其值. |

若成功,HTTP响应的正文格式,如下:

|  |  |
| --- | --- |
| **HTTP响应成功的正文** | 说明 |
| **{**  *"DevType" : "IPC",*  *"DevName" : "dev1",*  *"SerialNum" : "06dd012382c2d8c8201c",*  *"LocalTime" : "19961220T003957Z",*  *"Command" : "xxxxxx",*  *"CommandSeq" : n,*  **"Result": 0,**  **"Data": {**  **}**  **}** | *“DevType”:标识设备类型 IPC、NVR*  *“DevName”:设备名称*  *“SerialNum”:设备SN*  *“LocalTime”:当前时间*  *“Command”: 同请求*  *“CommandSeq”: 同请求*  "Result":返回的结果值. 0表示正确.其余负数表示错误.  "Data":返回的结果数据. 比如查询到的参数及其值. |

若失败,HTTP响应成功的正文格式,如下:

|  |  |
| --- | --- |
| **HTTP响应失败的正文** | 说明 |
| **{**  *"DevType" : "IPC",*  *"DevName" : "dev1",*  *"SerialNum" : "06dd012382c2d8c8201c",*  *"LocalTime" : "19961220T003957Z",*  *"Command" : "xxxxxx",*  *"CommandSeq" : n,*  **"Result": -1,**  **"ErrorString": "Operation Failed!"**  **}** | *“DevType”:标识设备类型 IPC、NVR*  *“DevName”:设备名称*  *“SerialNum”:设备SN*  *“LocalTime”:当前时间*  *“Command”: 同请求*  *“CommandSeq”: 同请求*  "Result":返回的结果值. 0表示正确.其余负数表示错误.  "ErrorString":操作失败时的提示信息. |

错误码列表.

|  |  |
| --- | --- |
| **错误码** | 说明 |
| 0 | 操作成功 |
| -655361 | 通常性错误 |
| -655362 | 错误的参数 |
| -655363 | 内存不足 |
| -655364 | 没有取得认证 |
| -655365 | 权限不足 |
| -655366 | 操作被阻塞 |
| -655367 | json内容错误 |
| -655368 | 内部错误 |
| -655369 | 不支持的功能 |
| -655370 | 无法创建线程 |
| -655371 | 同样任务已经存在 |
| -655372 | 文件无法访问 |
| -655373 | 正忙(通常是无法取得锁) |
| -655374 | 单用户登录 |
| -655375 | 需要先登录 |
| -655376 | ip子网掩码网关不匹配 |
| -655377 | 端口被占用 |
| -655378 | 在线用户已达最大数量 |

## URL

所有URL均以/digest开头,形如/digest/xxxx,其中xxxx代表接口名称.

协议中支持的接口名称,列表如下.

|  |  |  |
| --- | --- | --- |
| 接口名称 | 类别 | 功能描述 |
| frmAlarmException | 报警事件 | 异常报警配置的获取和设置 |
| frmMotionDetPara | 报警事件 | 移动侦测配置的获取和设置 |
| frmVideoHidePara | 报警事件 | 视频遮挡配置的获取和设置 |
| frmPTZLinkCFG | 报警事件 | 报警事件联动配置的获取和设置 |
| frmVideoShelterPara | 报警事件 | 隐私遮蔽配置的获取和设置 |
| frmAutoLensCorrection | 光学 | 自动镜头校正 |
| frmLensCustomCfg | 光学 | 自定义镜头参数的获取和设置 |
| frmGetHDFormatProgress | 录像 | 获取磁盘格式化进度 |
| frmGetHDFSInfo | 录像 | 获取是否需要强制格式化磁盘 |
| frmGetHDInfo | 录像 | 获取磁盘信息 |
| frmHDFormat | 录像 | 格式化磁盘 |
| frmMediaDisconnect | 录像 | 视频断开上报配置的获取和设置 |
| frmSearchSDCardPics | 录像 | 查询SD卡联动抓图 |
| frmVideoPlanPara | 录像 | 录像计划配置的获取和设置 |
| frmVideoQueryByMonth | 录像 | 按月查询录像 |
| frmVideoRecordsQuery | 录像 | 查询录像 |
| frmWaterMark | 录像 | 水印配置的获取和设置 |
| CaptureV2 | 媒体 | 预览抓图 |
| frmAudioAlarmCfg | 媒体 | 音频报警输出配置的获取和设置 |
| frmAudioPara | 媒体 | 音频参数配置的获取和设置 |
| frmAudioParaAbility | 媒体 | 获取音频参数能力集 |
| frmGetImageModeAbility | 媒体 | 获取图像模式能力集 |
| frmGetPNFormatAbility | 媒体 | 获取视频制式能力集 |
| frmImageCapability | 媒体 | 获取设备TWDR支持信息 |
| frmImageType | 媒体 | 场景抓拍配置的获取和设置 |
| frmMultiLineOSD | 媒体 | 多行OSD配置的获取和设置 |
| frmReqIFrame | 媒体 | 强制I帧 |
| frmSingleLineOSD | 媒体 | 单行OSD配置的获取和设置 |
| frmSpeechAlarmCfg | 媒体 | 音频报警试听 |
| frmVideoCompressAbility | 媒体 | 获取图像编码能力集 |
| frmVideoEffect | 媒体 | 图像效果配置的获取和设置 |
| frmVideoFormatPara | 媒体 | 视频制式配置的获取和设置 |
| frmVideoFormatPara\_v2 | 媒体 | 视频制式v2接口 |
| frmVideoImageModePara | 媒体 | 图像模式配置的获取和设置 |
| frmVideoIPCSetPara | 媒体 | 视频编码参数配置的获取和设置 |
| frmVideoParaEx | 媒体 | 视频参数配置的获取和设置 |
| frmVideoRegionOfInterest | 媒体 | ROI配置的获取和设置 |
| frmDTMFTrigger | 网络 | 手动触发Sip报警输出 |
| frmEmailSetting | 网络 | Email配置的获取和设置 |
| frmFTPSetting | 网络 | FTP配置的获取和设置 |
| frmGetDDNSServiceAbility | 网络 | 获取DDNS服务器列表 |
| frmGetManagerHostsPara | 网络 | 平台管理配置的获取和设置 |
| frmGetRtspUrl | 网络 | 获取RTSP流地址 |
| frmHttpAddrTest | 网络 | HTTP推送服务器地址测试 |
| frmHttpEventSetting | 网络 | HTTP事件配置的获取和设置 |
| frmHttpHttpsConfig | 网络 | HTTP端口配置的获取和设置 |
| frmHttpPushCfg | 网络 | HTTP推送配置的获取和设置 |
| frmMulticast | 网络 | 多播配置的获取和设置 |
| frmNasSetting | 网络 | Nas配置的获取和设置 |
| frmNetDDNSPara | 网络 | DDNS配置的获取和设置 |
| frmNetLtepara | 网络 | 4G配置的获取和设置 |
| frmNetNtpPara | 网络 | Ntp配置的获取和设置 |
| frmNetSipPara | 网络 | Sip配置的获取和设置 |
| frmNetSnmp | 网络 | SNMP配置的获取和设置 |
| frmNetTelnetPara | 网络 | Telnet配置的获取和设置 |
| frmNetUPNPPara | 网络 | UPNP配置的获取和设置 |
| frmNetworkSettings | 网络 | 网络接口配置的获取和设置 |
| frmP2PCfg | 网络 | P2P配置的获取和设置 |
| frmP2pState | 网络 | 获取P2P服务在线状态 |
| frmTurnServerCfg | 网络 | Turn服务配置的获取和设置 |
| frmTurnServerState | 网络 | 获取Turn服务在线状态 |
| frmParaPlatform28181 | 网络 | GB28181配置的获取和设置 |
| frmParaPlatform35114 | 网络 | GB35114配置的获取和设置 |
| frmRtmpPushCfg | 网络 | RTMP配置的获取和设置 |
| frmRtspCfg | 网络 | RTSP配置的获取和设置 |
| frmSmart1400Cfg | 网络 | Smart1400协议配置的获取和设置 |
| frmSmart1400AddrTest | 网络 | Smart1400服务器地址测试 |
| frmWiegandCfg | 网络 | 韦根配置的获取和设置 |
| frmWifiPara | 网络 | WIFI配置的获取和设置 |
| frmAlarmInPara | 物理接口 | 报警输入配置的获取和设置 |
| frmAlarmOut | 物理接口 | 报警输出配置的获取和设置 |
| frmDecoderPara | 物理接口 | PTZ编码配置的获取和设置 |
| frmFisheyeAbility | 物理接口 | 获取鱼眼相机能力集 |
| frmFisheyeChanCruise | 物理接口 | 鱼眼相机巡航线操作相关 |
| frmFisheyeChanMove | 物理接口 | 鱼眼相机鱼台控制相关 |
| frmFisheyeChanPos | 物理接口 | 获取鱼眼通道位置 |
| frmFisheyeChaPreset | 物理接口 | 鱼眼相机预置点操作相关 |
| frmFisheyMode | 物理接口 | 鱼眼相机模式配置的获取和设置 |
| frmGetPTZProtocal | 物理接口 | 获取PTZ协议列表 |
| frmPTZControl | 物理接口 | 球机云台控制相关 |
| frmPTZCruise | 物理接口 | 球机巡航线操作相关 |
| frmPTZExtend\_3DPosition | 物理接口 | 球机3D定位相关 |
| frmPTZExtend\_IdleOperation | 物理接口 | 球机空闲操作相关 |
| frmPTZExtend\_IrlightCtrl | 物理接口 | 球机红外补光配置的获取和设置 |
| frmPTZExtend\_SetCover | 物理接口 | 球机隐私遮蔽操作相关 |
| frmPTZExtend\_SetScan | 物理接口 | 球机两点扫描相关 |
| frmPTZExtend\_SprayMode | 物理接口 | 球机喷淋模式设置 |
| frmPTZExtend\_SprayPos | 物理接口 | 球机喷淋位置设置 |
| frmPTZExtend\_Wiper | 物理接口 | 球机雨刷操作相关 |
| frmPTZPreset | 物理接口 | 球机预置点操作相关 |
| frmPTZTrack | 物理接口 | 球机轨迹操作相关 |
| frmPTZDomeGetProtocol | 物理接口 | 获取网络球机协议能力集 |
| frmPTZDomeGetAbility | 物理接口 | 获取网络球机能力集 |
| frmPTZDomeUart | 物理接口 | 网络球机Uart配置的获取和设置 |
| frmPTZDomePark | 物理接口 | 网络球机看守配置的获取和设置 |
| frmPTZDomePatrolPath | 物理接口 | 网络球机巡航线配置的获取和设置 |
| frmPTZDomePowerOffMemory | 物理接口 | 网络球机断电记忆配置的获取和设置 |
| frmPTZDomeScheduleTask | 物理接口 | 网络球机计划任务配置的获取和设置 |
| frmPTZDomePresetControl | 物理接口 | 网络球机预置点操作相关 |
| frmPTZDomePatternControl | 物理接口 | 网络球机轨迹控制相关 |
| frmPTZDomePtzControl | 物理接口 | 网络球机云台控制相关 |
| frmPTZDomeParkCallOnce | 物理接口 | 立即执行一次网络球机看守 |
| frmPTZDomePatrolPathControl | 物理接口 | 网络球机巡航线控制相关 |
| frmPTZDomeOSD | 物理接口 | 网络球机PTZ OSD位置的获取和设置 |
| frmAutoReboot | 系统 | 自动维护配置的获取和设置 |
| frmBlackWhiteList | 系统 | 黑白名单配置的获取和设置 |
| frmDevicePara | 系统 | 设备信息配置的获取和设置 |
| frmDeviceReboot | 系统 | 设备重启 |
| frmDeviceTimeCtrl | 系统 | 设备时间配置的获取和设置 |
| frmDstPara | 系统 | 夏令时配置的获取和设置 |
| frmExportConfig | 系统 | 导出配置文件(文件下载) |
| frmGetConfigFileV2 | 系统 | 导出配置文件(base64) |
| frmGetFactoryInfo | 系统 | 获取系统信息 |
| frmGetQRCode | 系统 | 获取二维码信息 |
| frmGetQRCodePicture | 系统 | 获取二维码图片(bmp图) |
| frmGetQRCodePictureV2 | 系统 | 获取二维码图片(base64) |
| frmLocalSettings | 系统 | 本地配置的获取和设置 |
| frmLogCtrl | 系统 | 查询日志 |
| frmParaSysRestore | 系统 | 恢复默认 |
| frmPasswordLost | 系统 | 忘记密码重置 |
| frmSetConfigFileV2 | 系统 | 导入配置文件(base64) |
| frmSysUpdate | 系统 | 获取升级信息 |
| frmUserLogin | 系统 | 用户登录 |
| frmUserLogout | 系统 | 用户登出 |
| frmUserManage | 系统 | 用户信息操作相关 |
| frmUserOnline | 系统 | 获取在线用户信息 |
| frmUserOnlineNum | 系统 | 在线用户数配置的获取和设置 |
| frmUserRights\_V2 | 系统 | 用户权限配置的获取和设置 |
| frmWebApiVersion | 系统 | 获取设备Web api版本 |
| GetProgress | 系统 | 获取升级进度 |
| Upload | 系统 | 文件升级/配置包导入 |
| frmActionAlarmTimePara | 智能报警 | 声光报警联动项布防时间的获取和设置 |
| frmDensityPara | 智能报警 | 密度检测配置的获取和设置 |
| frmDetectLinkPara | 智能报警 | 智能报警事件联动配置的获取和设置 |
| frmGetSmartAbility | 智能报警 | 获取智能报警事件能力集 |
| frmHelmetDectPara | 智能报警 | 安全帽检测配置的获取和设置 |
| frmMaxHeightPara | 智能报警 | 限高检测配置的获取和设置 |
| frmObjectDectPara | 智能报警 | 物品检测配置的获取和设置 |
| frmQueryAlarmInfo | 智能报警 | 报警信息查询 |
| frmRegionDectPara | 智能报警 | 区域检测(电子围栏)配置的获取和设置 |
| frmRetrogradePara | 智能报警 | 逆行检测配置的获取和设置 |
| frmSmartMotion | 智能报警 | 智能移动侦测配置的获取和设置 |
| frmSoundAlarm | 智能报警 | 声音告警配置的获取和设置 |
| frmTargetCountPara | 智能报警 | 目标计数配置的获取和设置 |
| frmVideoDiagnose | 智能报警 | 视频诊断配置的获取和设置 |
| frmVideoFaceDetectV2 | 智能报警 | 人脸检测配置的获取和设置 |
| frmVideoFireDetect | 智能报警 | 火灾检测配置的获取和设置 |
| frmVideoPersonPara | 智能报警 | 人形检测配置的获取和设置 |
| frmVideoPlateDetect | 智能报警 | 车牌检测配置的获取和设置 |
| frmVideoSceneChange | 智能报警 | 场景变换配置的获取和设置 |
| frmVirtualLinePara | 智能报警 | 虚拟警戒线配置的获取和设置 |
| frmAbsentDectPara | 智能报警 | 离岗检测配置的获取和设置 |
| frmPersonSmartRecords | 智能结果统计 | 智能结果查询相关 |
| frmVideoFaceRecognitionPara | 智能人脸 | 人脸高级配置的获取和设置 |
| frmFaceDatabase | 智能人脸 | 人脸数据库操作相关 |
| frmFacePicture | 智能人脸 | 人脸模板操作相关 |
| frmFaceCompare | 智能人脸 | 人脸对比结果查询 |
| frmFaceGetVersion | 智能人脸 | 获取人脸算法库版本 |
| frmFaceRecognizeCfg | 智能人脸 | 人脸高级配置的获取和设置 |

# 接口说明

按照I8H协议在业务中的功能将接口进行分类,大致按照如下类别对各接口进行说明:

(1)系统:时间,用户管理,配置(恢复默认,导入导出),重启(立即,定时重启),黑白名单,升级,日志,等.

(2)网络:以太网和无线网等网络接口配置,网络应用协议配置,如PPPOE/DDNS/UPNP/FTP/SMTP/SNMP/RTSP/RTMP等.

(3)媒体:图像参数,显示参数,视频编码参数,视频输出参数,音频参数,抓图,预览等

(4)物理接口:报警输入输出,RS485/PTZ,报警灯,等.

(5)报警事件:移动侦测,遮挡报警,等.

(6)录像:录像配置,录像查询,录像回放,等.

(7)智能报警:行为分析(4类),人形检测(4类),人脸检测抓拍,等.

(8)智能人脸:人脸比对配置,人脸库录入/查询,人脸结果查询,等.

(9)智能结果统计:报表查询.

(10)光学参数:光圈,镜头,等.

## 系统

### 登录

|  |  |
| --- | --- |
| 用户登录设备 | |
| POST /digest/frmUserLogin HTTP/1.1 | |
| 请求的正文内容  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容  {  "Result": 0,  "Data": {  "ChanNum": 1,  "AlarmOutPortNum": 1,  "AlarmInPortNum": 2,  "RTSPPort": 554,  "RTMPPort": 1935,  "DVRType": 3,  "HttpPort": 80,  "DVRPort": 0,  "SerialNumber": "01560123bc487b8d7c54",  "DefaultPwd": 1,  "SessionId": 18874368,  "SessionTimeOut": 60  }  } | "ChanNum": Int型,设备通道数  "AlarmOutPortNum": Int型,报警输出个数  "AlarmInPortNum": Int型,报警输入个数  "RTSPPort": Int型,RTSP端口  "RTMPPort": Int型,RTMP端口  "DVRType": Int型,设备类型,3-IPC  "HttpPort": Int型,HTTP端口  "DVRPort": Int型,设备端口,保留字段  "SerialNumber": String型,设备序列号  "DefaultPwd": Int型,是否是默认密码,1-是,0-否  "SessionId":Int型,SessionID,设备开启了Session时才会返回此字段.  若设备开启了Session,后续请求头中需带CV-SESSION-ID:SessionId字段即  "SessionTimeOut":Int型,Session超时时间,单位:秒,设备开启了Session时才会返回此字段 |

### 保持心跳

|  |  |
| --- | --- |
| 保持心跳API,设备开启了Session时有效 | |
| POST /digest/frmKeepSessionAlive HTTP/1.1 | |
| 请求的正文内容  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

### 登出

|  |  |
| --- | --- |
| 用户登出 | |
| POST /digest/frmUserLogout HTTP/1.1 | |
| 请求的正文内容  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

### 系统信息

|  |  |
| --- | --- |
| 获取系统信息 | |
| POST /digest/frmGetFactoryInfo HTTP/1.1 | |
| 请求的正文内容  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容  {  "Result": 0,  "Data": {  "DeviceName": "",  "ProductName": "",  "DeviceType": "IPC",  "DeviceModel": "",  "Web": "",  "Tel": "",  "Copyright": "",  "Manufacturer": "",  "Brand": "",  "Customer": "DEFAULT",  "SensorModel": "IMX274",  "HwVersion": "V4.0-13.1.0.1",  "SwVersion": "V4.0",  "BuildDate": "20191202",  "ProductDate": "20191202",  "IsofDome": "n",  "Status": "Activated",  "CfgVersion": "1.0.0",  "LensSupport": "y",  "LensDrvType": "MS41908",  "LensType": "None",  "IrisSupport": "y",  "IrisType": "DC-IRIS",  "OptionalDevChanStreams": [  [  3  ]  ],  "IsOfFishEye": "n"  }  } | "DeviceName": String型,设备名称  "ProductName": String型,产品名称  "DeviceType": String型,设备类型  "DeviceModel": String型,设备模型  "Web": String型,产品网站  "Tel": String型,联系电话  "Copyright": String型,版权所有者  "Manufacturer": String型,制造商  "Brand": String型,品牌  "Customer": String型,客户  "SensorModel": String型,传感器模型  "HwVersion": String型,硬件版本  "SwVersion": String型,软件版本  "BuildDate": String型,软件打包日期  "ProductDate": String型,产品日期  "IsofDome": String型,是否是球机,y-是,y\_ex-网络球机,n-否  "Status": String型,授权状态  "CfgVersion": String型,配置版本  "LensSupport": String型,是否支持电动镜头,y-是,n-否  "LensDrvType": String型,镜头芯片类型  "LensType": String型,镜头类型  "IrisSupport": String型,是否支持电动光圈,y-是,n-否  "IrisType": String型,光圈类型  "OptionalDevChanStreams": Int型二维数组,各设备的各通道的实时流数目,第一维表示设备sensor个数,第二维表示各sensor支持的通道的实时流个数  "IsOfFishEye": String型,是否是鱼眼相机, y-是,n-否 |

### 设备信息

|  |  |
| --- | --- |
| 获取设备信息 | |
| POST /digest/frmDevicePara HTTP/1.1 | |
| 请求的正文内容  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容  {  "Result": 0,  "Data": {  "WebApiVersion": {  "Standard": "V1.6.4",  "Build": "2019-12-9 11:11:54"  },  "FaceDetectNum": 0,  "SoftwareVersion": "V4.0.20191202",  "HardwareVersion": "V4.0-13.1.0.1",  "SerialNumber": "02d00123a8ec7c024a4d",  "DVRType": 3,  "CustomerSN": "",  "CfgVersion": "1.0.0",  "CfgDate": "",  "DVRName": "",  "AlarmInPortNum": 2,  "DeviceTypeString":"ipc\_normal",  "FunctionInfo":{  "SmartInfo":{  "List": [{  "CounterWire": 1  }, {  "ElectronicFence": 2  }, {  "Retrograde": 1  }, {  "DetectAbsent": 1  }, {  "DetectHelmet": 0  }]  },  "PersonRecord":1,  "Face":{  "EnableTest":1,  "EnableSnap":1,  "EnableComp":1  },  "EnableAlarming":1  }  }  } | "WebApiVersion"."Standard": String型,api版本号  "WebApiVersion"."Build": String型,编译日期  "FaceDetectNum": Int型,是否是人脸识别机1-是,0-否  "SoftwareVersion": String型,软件版本  "HardwareVersion": String型,硬件版本  "SerialNumber": String型,设备序列号  "DVRType": Int型,设备类型,3-IPC  "CustomerSN": String型,客户序列号  "CfgVersion": String型,配置版本  "CfgDate": String型,配置日期  "DVRName": String型,设备名称  "AlarmInPortNum": Int型,报警输入个数  "DeviceTypeString": "ipc\_normal",//普通ipc  "ipc\_dome",//普通球机  "ipc\_netdome",//网络球机  "ipc\_temperature",//红外测温ipc  "ipc\_fisheye",//鱼眼ipc  "ipc\_4in1",//4目拼接  "ipc\_guard"//警戒ipc  "ipc\_guard\_ebike"//电梯警戒机ipc  "ipc\_guard\_car"//车型警戒机ipc  "ipc\_guard\_person"//人形警戒机ipc  "FunctionInfo.SmartInfo":ipc支持的智能能力集,内容与frmGetSmartAbility一致  "FunctionInfo.PersonRecord": Int型,是否支持计数区域逆行离岗的统计信息数据库,1-支持,0-不支持  "FunctionInfo.Face.EnableTest":Int型,是否支持人脸检测,1-支持,0-不支持  "FunctionInfo.Face.EnableSnap":Int型,是否支持人脸抓拍,1-支持,0-不支持  "FunctionInfo.Face.EnableComp":Int型,是否支持人脸识别,1-支持,0-不支持  "FunctionInfo.EnableAlarming":Int型,是否支持布防撤防,1-支持,0-不支持 |

|  |  |
| --- | --- |
| 设置设备信息 | |
| POST /digest/frmDevicePara HTTP/1.1 | |
| 请求的正文内容  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "DVRName": ""  }  } |  |
| 响应的正文内容  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

### 设备API版本获取

|  |  |
| --- | --- |
| 获取设备API版本 | |
| POST /digest/frmWebApiVersion HTTP/1.1 | |
| 请求的正文内容  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容  {  "Result": 0,  "Data": {  "Standard": "V1.6.4",  "Build": "2019-12-9 11:11:54"  }  } | "Standard": String型,api版本号  "Build": String型,编译日期 |

### 时间

|  |  |
| --- | --- |
| 查询设备本地时间 | |
| POST /digest/frmDeviceTimeCtrl HTTP/1.1 | |
| 请求的正文内容  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容  {  "Result": 0,  "Data": {  "Time": [2019,11,20,17,3,7]  }  } | "Time":Int型数组,[年,月,日,时,分,秒] |

|  |  |
| --- | --- |
| 设置设备本地时间 | |
| POST /digest/frmDeviceTimeCtrl HTTP/1.1 | |
| 请求的正文内容  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "Time": [2019,11,20,17,3,7]  }  } |  |
| 响应的正文内容  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

### 夏令时

|  |  |
| --- | --- |
| 获取夏令时配置 | |
| POST /digest/frmDstPara HTTP/1.1 | |
| 请求的正文内容  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容  {  "Result": 0,  "Data": {  "Enable": 0,  "Offset": 0,  "Type": 1,  "StartDay": {  "MonthWeekDay": "000000",  "Time": "000000"  },  "StopDay": {  "MonthWeekDay": "000000",  "Time": "000000"  }  }  } | "Enable": Int型,是否启用夏令时,1-启用,0-不启用  "Offset": Int型,偏移秒数  "Type": Int型,1-月周模式（目前仅支持1）  2-普通日模式(不考虑闰日),YearDay范围1-365  3-含闰日模式,YearDay范围0-365  "StartDay"."MonthWeekDay": String型,前两位-月,中两位-周,后两位-天  "StartDay"."Time": String型,前两位-时,中两位-分,后两位-秒  "StopDay"."MonthWeekDay": String型,前两位-月,中两位-周,后两位-天  "StopDay"."Time": String型,前两位-时,中两位-分,后两位-秒 |

|  |  |
| --- | --- |
| 设置夏令时配置 | |
| POST /digest/frmDstPara HTTP/1.1 | |
| 请求的正文内容  {  "Type": 1,  "Ch": 1,  "Data": {  "Enable": 0,  "Offset": 0,  "Type": 1,  "StartDay": {  "MonthWeekDay": "000000",  "Time": "000000"  },  "StopDay": {  "MonthWeekDay": "000000",  "Time": "000000"  }  }  } |  |
| 响应的正文内容  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

### 恢复默认

|  |  |
| --- | --- |
| 恢复默认 | |
| POST /digest/frmParaSysRestore HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "Network": 0,  "Alarm": 0,  "Account": 0,  "OtherCfg": 0  }  } | "Network",Int型, 网络配置是否恢复默认,0-否,1-是.  "Alarm",Int型, 报警配置是否恢复默认,0-否,1-是.  "Account",Int型, 用户配置是否恢复默认,0-否,1-是.  "OtherCfg",Int型, 其它配置是否恢复默认,0-否,1-是. |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

### 重启

|  |  |
| --- | --- |
| 重启 | |
| POST /digest/frmDeviceReboot HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

### 自动维护

|  |  |
| --- | --- |
| 获取自动维护配置 | |
| POST /digest/frmAutoReboot HTTP/1.1 | |
| 请求的正文内容  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容  {  "Result": 0,  "Data": {  "AutoRebootMode": 0,  "EveryDayTime": 0,  "DateTime": [  1970,  1,  1,  0,  0,  0  ],  "Week": [  {  "Enable": 0,  "Time": 0  },  ……  ],  "Month": {  "Days": [  0,  ……  ],  "Time": 0  }  }  } | "AutoRebootMode": Int型,维护模式,0-不维护,1-每天定时维护,2-每周定时维护,3-单次维护,4-每月选定的天维护  "EveryDayTime": Int型,每天的维护时间, 0-7位是分钟,8-15位是小时  "Week": Object型数组,长度为7,每周的维护时间,从星期天开始  "Week"[]."Enable":Int型,当天是否启用维护,1-启用,0-不启用  "Week"[]."Time": Int型,当天的维护时间, 0-7位是分钟,8-15位是小时  "Month"."Days": Int型数组,长度为31,每月启用的天数,1-启用,0-不启用  "Month"."Time": Int型,每月的维护时间, 0-7位是分钟,8-15位是小时 |

|  |  |
| --- | --- |
| 设置自动维护配置 | |
| POST /digest/frmAutoReboot HTTP/1.1 | |
| 请求的正文内容 |  |
| 1. 禁用:   {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "AutoRebootMode": 0  }  } |
| 2.每天:  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "AutoRebootMode": 1,  "EveryDayTime": 0  }  } |
| 3.每周:  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "AutoRebootMode": 2,  "Week": [  {  "Enable": 0,  "Time": 0  },  ……  ]  }  } |
| 4.单次  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "AutoRebootMode": 3,  "DateTime": [  1970,  1,  1,  0,  0,  0  ]  }  } |
| 5.每月  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "AutoRebootMode": 4,  "Month": {  "Days": [  0,  ……  ],  "Time": 0  }  }  } |
| 响应的正文内容  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

### 黑白名单

|  |  |
| --- | --- |
| 获取黑白名单配置 | |
| POST /digest/frmBlackWhiteList HTTP/1.1 | |
| 请求的正文内容  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容 | "Mode": Int型,0-均不启用,1-启用白名单,2-启用黑名单  "Index": Int型,序号  "Ipv4": String型,起始ipv4地址  "EndIpv4": String型, 结束ipv4地址  "Ipv6": String型, 起始ipv6地址  "EndIpv6": String型, 结束ipv6地址  "Mac": String型,MAC地址  "Direction": Int型,方向,0-all,1-in,2-out  "Protocol": Int型,协议,0-all,1-tcp,2-udp |
| 1.均不启用时:  {  "Result": 0,  "Data": {  "Mode": 0  }  } |
| 2.启用白名单时:  {  "Result": 0,  "Data": {  "Mode": 1,  "WhiteList": [  {  "Index": 0,  "Ipv4": "10.3.0.1",  "EndIpv4": "10.3.0.251",  "Ipv6": "",  "EndIpv6": "",  "Mac": "",  "Direction": 0,  "Protocol": 0  }  ]  }  } |
| 3.启用黑名单时:  {  "Result": 0,  "Data": {  "Mode": 2,  "BlackList": [  {  "Index": 0,  "Ipv4": "10.0.0.1",  "EndIpv4": "10.0.0.2",  "Ipv6": "",  "EndIpv6": "",  "Mac": "",  "Direction": 0,  "Protocol": 0  }  ]  }  } |

|  |  |
| --- | --- |
| 设置黑白名单配置 | |
| POST /digest/frmBlackWhiteList HTTP/1.1 | |
| 请求的正文内容 | "Action":String型,Set-切换模式,Add-添加数据,Delete-删除数据 |
| 1.切换模式:  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "Action": "Set",  "Mode": 2  }  } |
| 2.保存白名单配置:  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "Action": "Add",  "Mode": 1,  "IpList": [  {  "Ipv4": "0.0.0.0",  "EndIpv4": "255.255.255.255",  "Ipv6": "",  "EndIpv6": "",  "Direction": 0,  "Protocol": 0,  "Mac": ""  }  ]  }  } |
| 3.保存黑名单配置:  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "Action": "Add",  "Mode": 2,  "IpList": [  {  "Ipv4": "10.0.0.1",  "EndIpv4": "10.0.0.2",  "Ipv6": "",  "EndIpv6": "",  "Direction": 0,  "Protocol": 0,  "Mac": ""  }  ]  }  } |
| 响应的正文内容  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

### 升级

#### 获取升级信息

|  |  |
| --- | --- |
| 获取升级信息 | |
| POST /digest/frmSysUpdate HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "HDCount": 0,  "RemoteRight": 1,  "Path": "/dev/ipc.update",  "FreeSize": 15728640  }  } | "HDCount": Int型,磁盘个数  "RemoteRight": Int型,权限判断  "Path": String型,文件上传后的路径  "FreeSize": Int型,剩余空间 |

#### 上传文件并升级

|  |  |
| --- | --- |
| 上传文件 | |
| 使用form表单提交action到/digest/upload | |
| 因为上传需要Username Token 验证  使用input [type=hidden]隐藏域来传递 Username PasswordDigest Nonce Created 和uploadType  验证通过才会开始升级，否则返回401 | "Username":String型,登录设备的用户名  "PasswordDigest":String型,密码字符串,加密规则base64(sha1(Nonce + Created + Password)  "Created":String型,客户端生成token的时间(DateTime ISO格式)  "Nonce":String型,生成规则base64(sha1(Created + secret key))  "uploadType":String型,Update-升级包, Config-导入的配置包 |

#### 获取升级进度

|  |  |
| --- | --- |
| 获取升级信息 | |
| POST /digest/GetProgress HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "Progress": 0  }  } | "Progress":Int型,升级进度,范围0-100,进度为100时请调用重启API重启设备 |

### 日志

|  |  |
| --- | --- |
| 查询日志 | |
| POST /digest/frmLogCtrl HTTP/1.1 | |
| 请求的正文内容  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {  "StartTime": 1575820800,  "EndTime": 1575907199,  "MajorType": 0,  "PageNum": 20,  "CurrentPage": 1  }  } | "StartTime": Int型,开始时间,UTC时间  "EndTime": Int型, 结束时间,UTC时间  "MajorType": Int型,查询日志的主类型, 0-全部,1-报警,2-异常,3-操作  "PageNum": Int型,分页参数,每页条数  "CurrentPage": Int型,分页参数,当前是第几页 |
| 响应的正文内容  {  "Result": 0,  "Data": {  "LogResults": {  "LogItemCount": 82,  "TotalPages": 5,  "Items": [  {  "No": 1,  "LogDateTime": "2019-12-09 18:53:26",  "MajorType": 3,  "MinorType": 112,  "ChannelOrPort": 0,  "IP": "10.3.0.215",  "Status": 0  },  ……  ]  }  }  } | "No": Int型,序号  "LogDateTime": String型,日志记录时间  "MajorType": Int型, 日志的主类型, 0-全部,1-报警,2-异常,3-操作  "MinorType": Int型, 日志子类型  "ChannelOrPort": Int型,低16位表示通道号；高16位表示区域号  "IP": String型,操作关联的ip地址,  "Status": Int型,报警状态,1-开始,0-结束 |

### 配置文件导入

|  |  |
| --- | --- |
| 配置文件导入(base64) | |
| POST /digest/frmSetConfigFileV2 HTTP/1.1 | |
| 请求的正文内容  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "Base64Text ": ""  }  } | "Base64Text ":String型,配置文件base64编码的字符串 |
| 响应的正文内容  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

|  |  |
| --- | --- |
| 配置文件导入(文件上传方式) | |
| 使用form表单提交action到/digest/upload | |
| 因为上传需要Username Token 验证  使用input [type=hidden]隐藏域来传递 Username PasswordDigest Nonce Created 和uploadType  验证通过才会开始升级，否则返回401 | "Username":String型,登录设备的用户名  "PasswordDigest":String型,密码字符串,加密规则base64(sha1(Nonce + Created + Password)  "Created":String型,客户端生成token的时间(DateTime ISO格式)  "Nonce":String型,生成规则base64(sha1(Created + secret key))  "uploadType":String型,Update-升级包, Config-导入的配置包 |

### 配置文件导出

|  |  |
| --- | --- |
| 配置文件导出(base64) | |
| POST /digest/frmGetConfigFileV2 HTTP/1.1 | |
| 请求的正文内容  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容  {  "Result": 0,  "Data": {  "Base64Text": "XXX"  }  } | "Base64Text ":String型,配置文件base64编码的字符串 |

|  |  |
| --- | --- |
| 配置文件导出(文件下载) | |
| POST /digest/frmExportConfig HTTP/1.1 | |
| 请求的正文内容  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容  {  "Result": 0,  "Data": {  "FileName": "exportcfg.tgz",  "FileSize": 12817,  "Md5sum": "139daa83e08b68ae238edcb65a6c8c49"  }  } | "FileName": String型,导出生成的配置文件的名称,直接<http://ip/文件名>可下载导出的配置文件  "FileSize": Int型,导出生成的配置文件大小  "Md5sum": String型,导出生成的配置文件的md5值 |

### 二维码

|  |  |
| --- | --- |
| 获取二维码信息 | |
| POST /digest/frmGetQRCodeHTTP/1.1 | |
| 请求的正文内容  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容  {  "Result": 0,  "Data": {  "QRSelects": [  [  "Fseye",  [  "IOS App",  "http://itunes.apple.com/cn/app/fseye/id1069292577?mt=8"  ],  [  "Android App",  "http://play.google.com/store/apps/details?id=com.FSeye"  ],  [  "Device ID",  "umks02p7uaxa"  ]  ]  ]  }  } | "QRSelects":二维数组,第一维表示个数,第二维表示具体信息  "QRSelects"[][0]:String型,分组名称  "QRSelects"[][1][0]:String型,二维码信息的描述字符串  "QRSelects"[][1][1]:String型,二维码信息的字符串 |

|  |  |
| --- | --- |
| 获取二维码图片(bmp图) | |
| GET /digest/frmGetQRCodePicture?Pid=xxx&Username=admin&PasswordDigest=xx&Created=xx&Nonce=xxx HTTP/1.1 | |
|  | "Pid":String型,待转换成二维码的字符串  "Username":String型,登录设备的用户名  "PasswordDigest":String型,密码字符串,加密规则base64(sha1(Nonce + Created + Password)  "Created":String型,客户端生成token的时间(DateTime ISO格式)  "Nonce":String型,生成规则base64(sha1(Created + secret key)) |
| 响应的正文内容  C:\Users\Administrator\AppData\Roaming\Tencent\Users\2851660741\QQEIM\WinTemp\RichOle\V195~]9@1%)Y(@8F9}{J0ZI.jpg | Bmp格式图片 |

|  |  |
| --- | --- |
| 获取二维码图片(base64) | |
| POST /digest/frmGetQRCodePictureV2 HTTP/1.1 | |
| 请求的正文内容  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {  "QrCodeList": [  "umks02p7uaxa"  ]  }  } | "QrCodeList":String型数组,待转换成二维码的字符串 |
| 响应的正文内容  {  "Result": 0,  "Data": {  "QrCodeList": [  "xxxxxx"  ]  }  } | "QrCodeList":String型数组,转换成的二维码的base64字符串 |

### 本地设置

|  |  |
| --- | --- |
| 获取本地配置 | |
| POST /digest/frmLocalSettings HTTP/1.1 | |
| 请求的正文内容  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容  {  "Result": 0,  "Data": {  "WndMode": 0,  "PrevCapture": "E:\\PREVIEW",  "PbCapture": "F:\\PLAYBACK",  "FileCapture": "F:\\FILECAPTURE",  "BackupPath": "F:\\BACKUP",  "RecPath": "F:\\REC",  "PlatePics": "F:\\PLATE",  "FacePics": "F:\\FacePics",  "RecFormat": 1,  "PicQuality": 9,  "PasswordTips": 0,  "FacePicFormat": "name\_describe\_wiegandId\_faceId",  "EnableGB35114": 0  }  } | "WndMode": Int型,窗口模式,0-充满, 1-4:3, 2-16:9, 3-原画  "PrevCapture": String型,预览抓图路径  "PbCapture": String型,回放抓图路径  "FileCapture": String型,文件管理抓图路径  "BackupPath": String型,备份路径  "RecPath": String型, 预览录像路径  "PlatePics": String型, 车牌识别抓图路径  "FacePics": String型,人脸检测抓图路径  "RecFormat": Int型,预览录像文件类型  "PicQuality": Int型,预览质量(1-30)  "PasswordTips": Int型,1-提示修改密码,0-不提示  "FacePicFormat": String型,人脸批量录入文件名格式  "EnableGB35114": Int型,GB35114是否启用,1-启用,0-未启用 |

|  |  |
| --- | --- |
| 设置本地配置 | |
| POST /digest/frmLocalSettings HTTP/1.1 | |
| 请求的正文内容  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "WndMode": 0,  "PrevCapture": "E:\\PREVIEW",  "PbCapture": "F:\\PLAYBACK",  "FileCapture": "F:\\FILECAPTURE",  "BackupPath": "F:\\BACKUP",  "RecPath": "F:\\REC",  "PlatePics": "F:\\PLATE",  "FacePics": "F:\\FacePics",  "RecFormat": 1,  "PicQuality": 9,  "PasswordTips": 0,  "FacePicFormat": "name\_describe\_wiegandId\_faceId"  }  } |  |
| 响应的正文内容  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

### 用户信息

|  |  |
| --- | --- |
| 获取用户信息 | |
| POST /digest/frmUserManage HTTP/1.1 | |
| 请求的正文内容  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容  {  "Result": 0,  "Data": {  "UserInfoList": [  {  "UserName": "admin",  "Password": "",  "Priority": 2,  "BindIPv4": "",  "BindIPv6": "",  "BindMAC": ""  },  {  "UserName": "guest",  "Password": "",  "Priority": 0,  "BindIPv4": "",  "BindIPv6": "",  "BindMAC": ""  }  ]  }  } | "UserName": String型,用户名,  "PasswordB64": String型,密码.将密码经过Base64编码后的字符串. 有此PasswordB64字段则可忽略Password字段.通过接口获取时,此参数无意义.  "Password": String型,密码,明文.通过接口获取时,此参数无意义.  "Priority": Int型,用户类型,0-普通用户, 1-操作员, 2-管理员  "BindIPv4": String型,绑定ipv4  "BindIPv6": String型,绑定ipv6  "BindMAC": String型,绑定mac |

|  |  |
| --- | --- |
| 新增/修改用户 | |
| POST /digest/frmUserManage HTTP/1.1 | |
| 请求的正文内容  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "UserName": "x1",  "UserIndex": -1,  "Priority": 0,  "BindIPv4": "",  "BindIPv6": "",  "BindMAC": "",  "PasswordB64": "MTIzNDU2Nzg=",  "Password": "12345678"  }  } | "UserIndex": Int型,用户操作编号,-1表示添加新用户,其它表示各用户在用户管理列表上的编号，从0开始 |
| 响应的正文内容  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

|  |  |
| --- | --- |
| 删除用户 | |
| POST /digest/frmUserManage HTTP/1.1 | |
| 请求的正文内容  {  "Type": 2,  "Dev": 1,  "Ch": 1,  "Data": {  "UserName": "x1"  }  } | "UserName": String型,要删除的用户名 |
| 响应的正文内容  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

### 用户权限

|  |  |
| --- | --- |
| 获取用户权限 | |
| POST /digest/frmUserRights\_V2 HTTP/1.1 | |
| 请求的正文内容  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {  "UserName": "xxxx"  }  } | "UserName": String型,用户名,获取此用户的权限 |
| 响应的正文内容  {  "Result": 0,  "Data": {  "LocalRight": [  1,  1,  1,  1,  1,  1,  1,  1,  1,  1  ],  "RemoteRight": [  0,  0,  1,  0,  1,  0,  0,  1,  0,  0,  0,  0,  0,  1  ],  "LocalPreviewRight": [  1  ],  "NetPreviewRight": [  1  ],  "LocalPlaybackRight": [  1  ],  "NetPlaybackRight": [  2  ],  "LocalBackupRight": [  1  ],  "LocalPTZRight": [  1  ],  "NetPTZRight": [  0  ],  "LocalRecordRight": [  1  ],  "NetRecordRight": [  0  ]  }  } | "LocalRight": Int型数组,长度为10,本地权限列表  "LocalRight"[0]: Int型,本地控制云台,0-无权限,1-有权限  "LocalRight"[1]: Int型,本地手动录像,0-无权限,1-有权限  "LocalRight"[2]: Int型,本地回放,0-无权限,1-有权限  "LocalRight"[3]: Int型,本地设置参数,0-无权限,1-有权限  "LocalRight"[4]: Int型,本地查看状态,日志,0-无权限,1-有权限  "LocalRight"[5]: Int型,本地高级操作(升级,格式化,重启),0-无权限,1-有权限  "LocalRight"[6]: Int型,本地查看参数,0-无权限,1-有权限  "LocalRight"[7]: Int型,本地管理模拟和IPC,0-无权限,1-有权限  "LocalRight"[8]: Int型,本地备份,0-无权限,1-有权限  "LocalRight"[9]: Int型,本地关机,重启,0-无权限,1-有权限  "RemoteRight": Int型数组,长度为10,远程权限列表  "RemoteRight"[0]: Int型,远程控制云台,0-无权限,1-有权限  "RemoteRight"[1]: Int型,远程手动录像,0-无权限,1-有权限  "RemoteRight"[2]: Int型,远程回放,0-无权限,1-有权限  "RemoteRight"[3]: Int型,远程设置参数,0-无权限,1-有权限  "RemoteRight"[4]: Int型,远程查看状态,日志,0-无权限,1-有权限  "RemoteRight"[5]: Int型,远程高级操作(升级,格式化,重启),0-无权限,1-有权限  "RemoteRight"[6]: Int型,远程发起语音对讲,0-无权限,1-有权限  "RemoteRight"[7]: Int型,远程预览,0-无权限,1-有权限  "RemoteRight"[8]: Int型,远程请求报警上传,报警输出,0-无权限,1-有权限  "RemoteRight"[9]: Int型,远程控制,本地输出,0-无权限,1-有权限  "RemoteRight"[10]: Int型,远程控制串口,0-无权限,1-有权限  "RemoteRight"[11]: Int型,远程查看参数,0-无权限,1-有权限  "RemoteRight"[12]: Int型,远程管理模拟和IPC,0-无权限,1-有权限  "RemoteRight"[13]: Int型,远程关机,重启,0-无权限,1-有权限  "LocalPreviewRight"[13]: Int型,本地预览权限,0-无权限,1-有权限  "NetPreviewRight"[0]: Int型, 远程预览权限,0-无权限,1-有权限  "LocalPlaybackRight"[0]: Int型, 本地回放权限,0-无权限,1-有权限  "NetPlaybackRight"[0]: Int型, 远程回放权限,0-无权限,1-有权限  "LocalBackupRight"[0]: Int型, 本地备份权限,0-无权限,1-有权限  "LocalPTZRight"[0]: Int型, 本地控制PTZ权限,0-无权限,1-有权限  "NetPTZRight"[0]: Int型, 远程控制PTZ权限,0-无权限,1-有权限  "LocalRecordRight"[0]: Int型, 本地录像权限,0-无权限,1-有权限  "NetRecordRight"[0]: Int型, 远程录像权限,0-无权限,1-有权限 |

|  |  |
| --- | --- |
| 设置用户权限 | |
| POST /digest/frmUserRights\_V2 HTTP/1.1 | |
| 请求的正文内容  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "UserName": "xxxx",  "LocalRight": [  1,  ……  ],  "RemoteRight": [  0,  ……  ],  "LocalPreviewRight": [  1  ],  "NetPreviewRight": [  1  ],  "LocalPlaybackRight": [  1  ],  "NetPlaybackRight": [  1  ],  "LocalBackupRight": [  1  ],  "LocalPTZRight": [  1  ],  "NetPTZRight": [  1  ],  "LocalRecordRight": [  1  ],  "NetRecordRight": [  0  ]  }  } |  |
| 响应的正文内容  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

### 在线用户

|  |  |
| --- | --- |
| 获取在线用户 | |
| POST /digest/frmUserOnline HTTP/1.1 | |
| 请求的正文内容  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容  {  "Result": 0,  "Data": {  "UserInfoList": [  {  "UserName": "",  "ConnectInfo": [  {  "IPV4": "10.3.0.215",  "CreateTime": 1575957696,  "SessionId": 185606146  },  {  "IPV4": "10.3.0.215",  "CreateTime": 1575957697,  "SessionId": 202387458  }  ],  "StreamCount": 2  }  ]  }  } | "UserName": String型,用户名  "IPV4": String型,连接IP地址  "CreateTime": Int型,连接创建时间,UTC时间  "SessionId": Int型,连接会话id,  "StreamCount":Int型,连接打开的流数量 |

### 在线用户数

|  |  |
| --- | --- |
| 获取在线用户数配置 | |
| POST /digest/frmMaxUserOnlineNum HTTP/1.1 | |
| 请求的正文内容  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容  {  "Result": 0,  "Data": {  "SessionCount": 5  }  } | "SessionCount": Int型,最大在线用户数,设备开启了Session时有效 |

|  |  |
| --- | --- |
| 设置在线用户数配置 | |
| POST /digest/frmMaxUserOnlineNum HTTP/1.1 | |
| 请求的正文内容  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "SessionCount": 5  }  } |  |
| 响应的正文内容  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

### 密码重置

|  |  |
| --- | --- |
| 获取密码重置信息 | |
| POST /digest/frmPasswordLost HTTP/1.1 | |
| 请求的正文内容  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {  "UserName": "admin"  }  } | "UserName":String型,需要重置密码的用户名 |
| 响应的正文内容  {  "Result": 0,  "Data": {  "SerialNumber": "02d00123a8ec7c024a4d",  "Token": "D2VDl10ip3KRC59g"  }  } | "SerialNumber": String型,设备序列号  "Token": String型,唯一标识符 |

|  |  |
| --- | --- |
| 重置用户密码 | |
| POST /digest/frmPasswordLost HTTP/1.1 | |
| 请求的正文内容  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "UserName": "admin",  "SerialNumber": "02d00123a8ec7c024a4d",  "Token": "U98Uc8nzg0BIT2ex",  "RestoreInfo": "xxxc"  }  } | "RestoreInfo":String型,密码重置信息,请联系厂家获取 |
| 响应的正文内容  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

## 网络

### 网络接口配置

|  |  |
| --- | --- |
| 获取网络接口配置 | |
| POST /digest/frmNetworkSettings HTTP/1.1 | |
| 请求的正文内容  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容  {  "Result": 0,  "Data": {  "NetInterface": 5,  "UseDhcp": 0,  "DVRIP": "10.3.203.206",  "DvrPort": 8000,  "DVRIPMask": "255.255.0.0",  "GatewayIpAddr": "10.3.0.1",  "MACAddr": "00:EC:7C:02:4A:4E",  "EnableBindGateway": 0,  "GatewayMacAddr": "",  "RtspPort": 554,  "RtspAuthMode": 0,  "RTMPPort": 1935,  "HttpPortNo": 80,  "HttpsPort": 443,  "MulticastIpAddr": "238.255.255.255",  "MulticastPort": 28080,  "DefaultRoute": 0,  "NetworkCardNum": 1,  "AlarmHostIpAddr": "",  "AlarmHostIpPort": 0,  "DnsServer1IpAddr": "8.8.8.8",  "DnsServer2IpAddr": "008.008.004.004",  "EnableSnmp": 0,  "SnmpHostIp": "",  "SnmpCount": 3,  "SnmpInterval": 60,  "EnablePPPOE": 1,  "PPPoEUser": "12",  "PPPoEIP": "",  "PPPoEPassword": "12"  }  } | "NetInterface": Int型,网络接口,1-10MBase-T, 2-10MBase-T全双工, 3-100MBase-TX,4-100M全双工,5-10M/100M自适应  "UseDhcp": Int型,是否启用DHCP,0-不启用,1-启用  "DVRIP": String型,设备IP地址  "DvrPort": Int型,通讯端口,默认8000  "DVRIPMask": String型,子网掩码  "GatewayIpAddr": String型,网关地址  "MACAddr": String型,物理地址  "EnableBindGateway": Int型,是否绑定网关地址,1-是,0-否  "GatewayMacAddr": String型,网关物理地址  "RtspPort": Int型,RTSP端口  "RtspAuthMode": Int型,  "RTMPPort": Int型,RTMP端口  "HttpPortNo": Int型,HTTP端口  "HttpsPort": Int型,HTTPS端口  "MulticastIpAddr": String型,多播地址(目前无效,保留字段)  "MulticastPort": Int型,多播端口(目前无效,保留字段)  "DefaultRoute": Int型,默认路由  "NetworkCardNum": Int型,设备实际可配置的网卡数目  "AlarmHostIpAddr": String型,报警主机IP地址  "AlarmHostIpPort": Int型,报警主机端口  "DnsServer1IpAddr": String型,首选DNS地址  "DnsServer2IpAddr": String型,备用DNS地址  "EnableSnmp": Int型,是否启用SNMP,1-启用,0-不启用  "SnmpHostIp": String型,SNMP地址  "SnmpCount": Int型,SNMP发送次数  "SnmpInterval": Int型,SNMP发送间隔  "EnablePPPOE": Int型,是否启用PPPOE拨号,1-启用,0-不启用  "PPPoEUser": String型,PPPOE用户名  "PPPoEIP": String型,PPPOE网络地址  "PPPoEPassword": String型,PPPOE密码 |
|  |  |

|  |  |
| --- | --- |
| 设置网络接口配置 | |
| POST /digest/frmNetworkSettings HTTP/1.1 | |
| 请求的正文内容  {  "Type": 1,  "Ch": 1,  "Data": {  "NetInterface": 5,  "UseDhcp": 0,  "DVRIP": "10.3.203.206",  "DvrPort": 8000,  "DVRIPMask": "255.255.0.0",  "GatewayIpAddr": "10.3.0.1",  "EnableBindGateway": 0,  "GatewayMacAddr": "",  "RtspPort": 554,  "RtspAuthMode": 0,  "RTMPPort": 1935,  "HttpPortNo": 80,  "HttpsPort": 443,  "MulticastIpAddr": "238.255.255.255",  "MulticastPort": 28080,  "DefaultRoute": 0,  "NetworkCardNum": 1,  "AlarmHostIpAddr": "",  "AlarmHostIpPort": 0,  "DnsServer1IpAddr": "8.8.8.8",  "DnsServer2IpAddr": "8.8.4.4",  "EnableSnmp": 0,  "SnmpHostIp": "",  "SnmpCount": 3,  "SnmpInterval": 60,  "EnablePPPOE": 1,  "PPPoEUser": "12",  "PPPoEIP": "",  "PPPoEPassword": "12"  }  } |  |
| 响应的正文内容  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

### HTTP/HTTPS

|  |  |
| --- | --- |
| 获取HTTP/HTTPS端口配置 | |
| POST /digest/frmHttpHttpsConfig HTTP/1.1 | |
| 请求的正文内容  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容  {  "Result": 0,  "Data": {  "HttpPortNo": 80,  "HttpsPort": 443,  "EnableHttp": 1,  "EnableHttps": 0,  "RedirectHttpToHttps": 0  }  } | "HttpPortNo": Int型,HTTP端口  "HttpsPort": Int型,HTTPS端口  "EnableHttp": Int型,是否启用HTTP,1-启用,0-不启用  "EnableHttps": Int型,是否启用HTTPS,1-启用,0-不启用  "RedirectHttpToHttps": Int型,是否强制HTTPS,1-是,0-否 |
|  |  |

特别说明：不支持http和https 同时关闭，若设置配置时发生此种情况Http会默认开启

|  |  |
| --- | --- |
| 设置HTTP/HTTPS端口配置 | |
| POST /digest/frmHttpHttpsConfig HTTP/1.1 | |
| 请求的正文内容  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "HttpPortNo": 80,  "HttpsPort": 443,  "EnableHttp": 1,  "EnableHttps": 1,  "RedirectHttpToHttps": 0  }  } |  |
| 响应的正文内容  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

### HTTP推送

#### HTTP推送配置

|  |  |
| --- | --- |
| 获取HTTP推送配置 | |
| POST /digest/frmHttpPushCfg HTTP/1.1 | |
| 请求的正文内容  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容  {  "Result": 0,  "Data": {  "Enable": 1,  "ServerAddr": "10.3.0.103:8086",  "ServerPort": 0,  "HeartBeatInterval": 3,  "EventListMaxLen": 8,  "Username": "admin",  "PasswordB64": "MTIzNDU2Nzg=",  "Password": "h123456",  "Auth": 0  }  } | "Enable": Int型,是否启用HTTP推送,1-启用,0-不启用  "ServerAddr": String型,服务器地址  "ServerPort": Int型,服务器端口  "HeartBeatInterval": Int型,心跳间隔  "EventListMaxLen": Int型,事件列表最大长度,  "Username": String型,用户名,  "PasswordB64": String型,密码,经过Base64编码后的字符串. 有此字段则忽略"Password": String型,用户名,  "Auth": Int型,验证方式 |
|  |  |

|  |  |
| --- | --- |
| 设置HTTP推送配置 | |
| POST /digest/frmHttpPushCfg HTTP/1.1 | |
| 请求的正文内容  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "Enable": 1,  "ServerAddr": "10.3.0.103:8086",  "ServerPort": 0,  "HeartBeatInterval": 3,  "EventListMaxLen": 8,  "Username": "admin",  "Password": "h123456",  "Auth": 0  }  } |  |
| 响应的正文内容  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

#### HTTP推送服务器地址测试

|  |  |
| --- | --- |
| HTTP推送服务器地址测试 | |
| POST /digest/frmHttpAddrTest HTTP/1.1 | |
| 请求的正文内容  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {  "ServerAddr": "10.3.0.103:8086"  }  } |  |
| 响应的正文内容  {  "Result": 0,  "Data": {  "Code": -1,  "ErrorInfo": "Couldn't connect to server"  }  } | "Code":Int型,测试结果,0-地址可用,其他-地址不可用  "ErrorInfo":String型,错误详细信息,Code不为0时才会返回此字段 |

### HTTP事件

|  |  |
| --- | --- |
| 获取HTTP事件配置 | |
| POST /digest/frmHttpEventSetting HTTP/1.1 | |
| 请求的正文内容  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容  {  "Result": 0,  "Data": {  "HPort": 0,  "HType": 0,  "HAuthMode": 0,  "HSSL": 0,  "HServer": "",  "HUserName": "",  "HPassword": "",  "HParam": ""  }  } | "HPort": Int型,端口  "HType": Int型,请求类型,0-GET,1-PUT,2-POST  "HAuthMode": Int型,是否启用认证,1-启用,0-不启用  "HSSL": Int型,是否启用SSL,1-启用,0-不启用  "HServer": String型,服务器地址  "HUserName": String型,用户名  "HPassword": String型,密码  "HParam": String型,请求参数 |

|  |  |
| --- | --- |
| 设置HTTP事件配置 | |
| POST /digest/frmHttpEventSetting HTTP/1.1 | |
| 请求的正文内容  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "HPort": 0,  "HType": 0,  "HAuthMode": 0,  "HSSL": 0,  "HServer": "",  "HUserName": "",  "HPassword": "",  "HParam": ""  }  } |  |
| 响应的正文内容  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

### RTSP

|  |  |
| --- | --- |
| 获取RTSP配置 | |
| POST /digest/frmRtspCfg HTTP/1.1 | |
| 请求的正文内容  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容  {  "Result": 0,  "Data": {  "Enable": 1,  "RTSPPort": 554,  "RtspHttpPort": 8002,  "Auth": 0  }  } | "Enable": Int型,是否启用RTSP, 1-启用,0-不启用  "RTSPPort": Int型,RTSP端口  "RtspHttpPort": Int型,RTSP HTTP端口  "Auth": Int型,启用认证,按位操作,第一位表示是否启用base64认证,第二位表示是否启用digest认证 |

|  |  |
| --- | --- |
| 设置RTSP配置 | |
| POST /digest/frmRtspCfg HTTP/1.1 | |
| 请求的正文内容  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "Enable": 1,  "RTSPPort": 554,  "RtspHttpPort": 8002,  "Auth": 0  }  } |  |
| 响应的正文内容  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

### RTSP流地址获取

|  |  |
| --- | --- |
| 获取RTSP标准实时流地址 | |
| POST /digest/frmGetRtspUrl HTTP/1.1 | |
| 请求的正文内容  {  "Type": 10,  "Dev": 1,  "Ch": 1,  "Data": {  "Channel": 1,  "StreamType": 0  }  } | "Type":Int型,10-获取标准实时流地址  "Channel": Int型,通道号  "StreamType": Int型,码流类型,0-主码流,1-子码流,2-第三码流 |
| 响应的正文内容  {  "Result": 0,  "Data": {  "RtspUrl": "rtsp://10.3.203.206:554/ch01.264"  }  } | "RtspUrl": String型,返回的标准实时流地址 |

|  |  |
| --- | --- |
| 获取RTSP私有实时流地址 | |
| POST /digest/frmGetRtspUrl HTTP/1.1 | |
| 请求的正文内容  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {  "Channel": 1,  "StreamType": 0  }  } | "Type":Int型, 0-获取私有实时流地址  "Channel": Int型,通道号  "StreamType": Int型,码流类型,0-主码流,1-子码流,2-第三码流 |
| 响应的正文内容  {  "Result": 0,  "Data": {  "RtspUrl": "rtsp://10.3.203.206:554/living\_comb01.264"  }  } | "RtspUrl": String型,返回的私有实时流地址 |

|  |  |
| --- | --- |
| 获取RTSP对讲流地址 | |
| POST /digest/frmGetRtspUrl HTTP/1.1 | |
| 请求的正文内容  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "Channel": 1,  "AudioType":1  }  } | "Type":Int型,1-获取对讲流地址  "Channel": Int型,通道号  "AudioType": 请求的音频类型,1-对讲 |
| 响应的正文内容  {  "Result": 0,  "Data": {  "RtspUrl": "rtsp://10.3.3.6:554/audioback/ch\_01/type\_g711a"  }  } | "RtspUrl": String型,返回的对讲流地址 |

|  |  |
| --- | --- |
| 获取RTSP标准回放流地址 | |
| POST /digest/frmGetRtspUrl HTTP/1.1 | |
| 请求的正文内容  {  "Type": 13,  "Ch": 1,  "Data": {  "Channel": 1,  "StreamType": 0,  "StartTime": "20191130000000",  "StopTime": "20191130235959",  "TimeMode":"Local"  }  } | "Type":Int型,13-获取标准回放流地址  "Channel": Int型,通道号  "StreamType": Int型,码流类型,0-主码流,1-子码流,2-第三码流  "StartTime": String型,开始时间,格式:年月日时分秒,YYYYMMDDhhmmss  "StopTime": String型,结束时间,格式:年月日时分秒,YYYYMMDDhhmmss  "TimeMode":String型,时间类型(用来解析传进来的StartTime/StopTime是何种时间), "local"表Ipc本地时间,"GMT"表零时区时间,非必填参数,若不填则默认"GMT",目的是向前兼容. |
| 响应的正文内容  {  "Result": 0,  "Data": {  "RtspUrl": "rtsp://10.3.203.206:554/recording?ch=1&stream=0  &start=20191130000000&stop=20191130235959&rec\_inquiry\_type=0"  }  } | "RtspUrl": String型,返回的标准回放流地址 |

|  |  |
| --- | --- |
| 获取RTSP私有回放流地址 | |
| POST /digest/frmGetRtspUrl HTTP/1.1 | |
| 请求的正文内容  {  "Type": 3,  "Ch": 1,  "Data": {  "Channel": 1,  "StreamType": 0,  "StartTime": "20191130000000",  "StopTime": "20191130235959",  "TimeMode":"Local"  }  } | "Type":Int型, 3-获取私有回放流地址  "Channel": Int型,通道号  "StreamType": Int型,码流类型,0-主码流,1-子码流,2-第三码流  "StartTime": String型,开始时间,格式:年月日时分秒,YYYYMMDDhhmmss  "StopTime": String型,结束时间,格式:年月日时分秒,YYYYMMDDhhmmss  "TimeMode":String型,时间类型(用来解析传进来的StartTime/StopTime是何种时间), "Local"表Ipc本地时间,"GMT"表零时区时间,非必填参数,若不填则默认"GMT",目的是向前兼容. |
| 响应的正文内容  {  "Result": 0,  "Data": {  "RtspUrl": "rtsp://10.3.203.206:554/recording\_comb?ch=1&strea  m=0&start=20191130000000&stop=20191130235959&rec\_inquiry\_type=0"  }  } | "RtspUrl": String型,返回的私有回放流地址 |

### RTMP

|  |  |
| --- | --- |
| 获取RTMP配置 | |
| POST /digest/frmRtmpPushCfg HTTP/1.1 | |
| 请求的正文内容  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容  {  "Result": 0,  "Data": {  "Enable": 1,  "RTMPPort": 1935,  "PushCfg": {  "Client0": {  "Stream0": {  "Enable": 0,  "EnableAudio": 0,  "Url": ""  }  }  }  }  } | "Enable": Int型,是否启用RTMP,1-启用,0-不启用  "RTMPPort": Int型,RTMP端口  "Client0": 客户端1  "Stream0": 主码流  "Stream0"."Enable":Int型,主码流是否启用RTMP,1-启用,0-不启用  "Stream0"."EnableAudio":Int型,主码流RTMP是否启用音频,1-启用,0-不启用  "Stream0"."Url":String型,IP地址 |

|  |  |
| --- | --- |
| 设置RTMP配置 | |
| POST /digest/frmRtmpPushCfg HTTP/1.1 | |
| 请求的正文内容  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "Enable": 1,  "RTMPPort": 1935,  "PushCfg": {  "Client0": {  "Stream0": {  "Enable": 0,  "EnableAudio": 0,  "Url": ""  }  }  }  }  } |  |
| 响应的正文内容  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

### 多播

|  |  |
| --- | --- |
| 获取多播配置 | |
| POST /digest/frmMulticast HTTP/1.1 | |
| 请求的正文内容  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容  {  "Result": 0,  "Data": {  "IP": "238.255.0.8",  "Port": 28090,  "TTL": 252,  "AudioIP": "238.255.0.9",  "AudioPort": 28070,  "AudioTTL": 241,  "AuxIP": "238.255.0.35",  "AuxPort": 28023,  "AuxTTL": 211,  "AuxAudioIP": "238.255.0.1",  "AuxAudioPort": 28067,  "AuxAudioTTL": 219,  "ThirdIP": "238.255.0.44",  "ThirdPort": 28099,  "ThirdTTL": 223,  "ThirdAudioIP": "238.255.0.0",  "ThirdAudioPort": 28055,  "ThirdAudioTTL": 233,  "Enable": 1  }  } | "Enable": Int型,是否启用多播,1-启用,0-不启用  "IP": String型,主码流视频IP地址  "Port": Int型, 主码流视频端口  "TTL": Int型, 主码流视频TTL  "AudioIP": String型,主码流音频IP地址  "AudioPort": Int型, 主码流音频端口  "AudioTTL": Int型, 主码流音频TTL  "AuxIP": String型,子码流视频IP地址  "AuxPort": Int型, 子码流视频端口  "AuxTTL": Int型, 子码流视频TTL  "AuxAudioIP": String型, 子码流音频IP地址  "AuxAudioPort": Int型, 子码流音频端口  "AuxAudioTTL": Int型, 子码流音频TTL  "ThirdIP": String型,第三码流视频IP地址  "ThirdPort": Int型, 第三码流视频端口  "ThirdTTL": Int型, 第三码流视频TTL  "ThirdAudioIP": String型, 第三码流音频IP地址  "ThirdAudioPort": Int型, 第三码流音频端口  "ThirdAudioTTL": Int型, 第三码流音频TTL |

|  |  |
| --- | --- |
| 设置多播配置 | |
| POST /digest/frmMulticast HTTP/1.1 | |
| 请求的正文内容  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "IP": "238.255.0.8",  "Port": 28090,  "TTL": 252,  "AudioIP": "238.255.0.9",  "AudioPort": 28070,  "AudioTTL": 241,  "AuxIP": "238.255.0.35",  "AuxPort": 28023,  "AuxTTL": 211,  "AuxAudioIP": "238.255.0.1",  "AuxAudioPort": 28067,  "AuxAudioTTL": 219,  "ThirdIP": "238.255.0.44",  "ThirdPort": 28099,  "ThirdTTL": 223,  "ThirdAudioIP": "238.255.0.0",  "ThirdAudioPort": 28055,  "ThirdAudioTTL": 233,  "Enable": 1  }  } |  |
| 响应的正文内容  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

### DDNS

#### 获取DDNS服务器列表

|  |  |
| --- | --- |
| 获取DDNS服务器列表 | |
| POST /digest/frmGetDDNSServiceAbility HTTP/1.1 | |
| 请求的正文内容  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容  {  "Result": 0,  "Data": {  "ServiceList": [  {  "Describe": "PeanutHull",  "ServerName": "ddns.oray.org",  "Index": 0,  "ServerPort": 80  },  {  "Describe": "DnsDynamic",  "ServerName": "www.dnsdynamic.org",  "Index": 4,  "ServerPort": 80  }  ]  }  } | "Describe":String型,服务器描述  "ServerName":String型,服务器域名  "Index":Int型,内部序号:0-私有DDNS,1-Dyndns;2－PeanutHull(花生壳), 3-NO-IP,4-qdns,7-DnsDynamic,8-ndns.ir  "ServerPort":Int型,服务器端口 |

#### DDNS配置

|  |  |
| --- | --- |
| 获取DDNS配置 | |
| POST /digest/frmNetDDNSPara HTTP/1.1 | |
| 请求的正文内容  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容  {  "Result": 0,  "Data": {  "EnableDDNS": 0,  "DdnsList": [  {  "Enable": 0,  "UserName": "",  "Password": "",  "DomainName": ""  },  {  "Enable": 0,  "UserName": "",  "Password": "",  "DomainName": ""  }  ]  }  } | "EnableDDNS": Int型,是否启用DDNS, 1-启用,0-不启用  "Enable": Int型,是否启用, 1-启用,0-不启用  "UserName": String型,用户名  "Password": String型,密码  "DomainName": String型,设备域名 |

|  |  |
| --- | --- |
| 设置DDNS配置 | |
| POST /digest/frmNetDDNSPara HTTP/1.1 | |
| 请求的正文内容  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "EnableDDNS": 0,  "DdnsList": [  {  "Enable": 0,  "UserName": "",  "Password": "",  "DomainName": ""  },  {  "Enable": 0,  "UserName": "",  "Password": "",  "DomainName": ""  }  ]  }  } |  |
| 响应的正文内容  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

### UPNP

|  |  |
| --- | --- |
| 获取UPNP配置 | |
| POST /digest/frmNetUPNPPara HTTP/1.1 | |
| 请求的正文内容  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容  {  "Result": 0,  "Data": {  "Enable": 1,  "UpnpList": [  {  "InPort": 1,  "OutPort": 1,  "OutIP": "172.31.70.28",  "TcpOrUdp": 0,  "Status": 1  }  ]  }  } | "Enable": Int型,是否启用UPNP, 1-启用,0-不启用  "InPort": Int型,内部端口  "OutPort": Int型,外部端口  "OutIP": String型,外部IP  "TcpOrUdp": Int型,协议,0-TCP,1-UDP  "Status": Int型,状态,0-失败,1-成功 |

|  |  |
| --- | --- |
| 设置UPNP配置 | |
| POST /digest/frmNetUPNPPara HTTP/1.1 | |
| 请求的正文内容  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "Enable": 1,  "UpnpList": [  {  "InPort": 1,  "OutPort": 1,  "OutIP": "172.31.70.28",  "TcpOrUdp": 0,  "Status": 1  }  ]  }  } |  |
| 响应的正文内容  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

### FTP

|  |  |
| --- | --- |
| 获取FTP配置 | |
| POST /digest/frmFTPSetting HTTP/1.1 | |
| 请求的正文内容  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容  {  "Result": 0,  "Data": {  "FLinkMode": 0,  "FPort": 21,  "FServer": "",  "FUserName": "kk",  "FPassword": "123456",  "FServerDir": ""  }  } | "Enable": Int型,是否启用UPNP, 1-启用,0-不启用  "FLinkMode": Int型,联动模式,0-被动,1-主动  "FPort": Int型,FTP端口  "FServer": String型,服务器地址  "FUserName": String型,用户名  "FPasswordB64": String型,密码,经过Base64编码后的字符串.有此字段则可忽略FPassword字段.  "FPassword": String型,密码  "FServerDir": String型,路径 |

|  |  |
| --- | --- |
| 设置FTP配置 | |
| POST /digest/frmFTPSetting HTTP/1.1 | |
| 请求的正文内容  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "FLinkMode": 0,  "FPort": 21,  "FServer": "",  "FUserName": "kk",  "FPasswordB64": "MTIzNDU2",  "FPassword": "123456",  "FServerDir": ""  }  } |  |
| 响应的正文内容  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

### EMAIL

|  |  |
| --- | --- |
| 获取EMAIL配置 | |
| POST /digest/frmEmailSetting HTTP/1.1 | |
| 请求的正文内容  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容  {  "Result": 0,  "Data": {  "SenderAddress": "",  "PasswordB64": "",  "Password": "",  "EnableSSL": 0,  "Attachment": 0,  "EnableVerify": 0,  "RecvArray": [  {  "RecvName": "",  "RecvAddress": ""  },  {  "RecvName": "",  "RecvAddress": ""  },  {  "RecvName": "",  "RecvAddress": ""  }  ],  "MailInterval": 0,  "SmtpServer": "",  "Pop3Server": "",  "SmtpPort": 0  }  } | "Enable": Int型,是否启用UPNP, 1-启用,0-不启用  "FLinkMode": Int型,联动模式,0-被动,1-主动  "FPort": Int型,FTP端口  "FServer": String型,服务器地址  "FUserName": String型,用户名  "FPassword": String型,密码  "FServerDir": String型,路径  "SenderAddress": String型,发件人地址  "PasswordB64": String型,密码,经过Base64编码后的字符串.有此字段则可忽略"Password"字段.  "Password": String型,密码  "EnableSSL": Int型,是否启用SSL,1-启用,0-不启用  "Attachment": Int型,是否带有附件,0-不带,1-带  "EnableVerify": Int型, 是否启用服务器身份验证,1-启用,0-不启用  "RecvArray": Object数组,收件人列表,长度为3  "RecvName": String型,收件人名称  "RecvAddress": String型,收件人地址  "MailInterval": Int型,邮件发送间隔,单位:秒  "SmtpServer": String型,SMTP服务器地址  "Pop3Server": String型,POP3服务器地址  "SmtpPort": Int型,SMTP端口 |

|  |  |
| --- | --- |
| 设置EMAIL配置 | |
| POST /digest/frmEmailSetting HTTP/1.1 | |
| 请求的正文内容  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "SenderAddress": "",  "PasswordB64": "",  "Password": "",  "EnableSSL": 0,  "Attachment": 0,  "EnableVerify": 0,  "RecvArray": [  {  "RecvName": "",  "RecvAddress": ""  },  {  "RecvName": "",  "RecvAddress": ""  },  {  "RecvName": "",  "RecvAddress": ""  }  ],  "MailInterval": 0,  "SmtpServer": "",  "Pop3Server": "",  "SmtpPort": 0  }  } |  |
| 响应的正文内容  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

### SNMP

|  |  |
| --- | --- |
| 获取SNMP配置 | |
| POST /digest/frmNetSnmp HTTP/1.1 | |
| 请求的正文内容  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容  {  "Result": 0,  "Data": {  "SNMPEnable": 0,  "ServerPort": 161,  "ReadCommunity": "",  "WriteCommunity": "",  "TrapHostIP": "",  "TrapHostPort": 162,  "SendCount": 3,  "SendInterval": 60  }  } | "SNMPEnable": Int型,是否启用SNMP, 1-启用,0-不启用  "ServerPort": Int型,SNMP端口  "ReadCommunity": String型,只读团体名  "WriteCommunity": String型,读写团体名  "TrapHostIP": String型,Trap主机IP  "TrapHostPort": Int型,Trap主机端口  "SendCount": Int型,发送次数  "SendInterval": Int型,发送间隔,单位:秒 |

|  |  |
| --- | --- |
| 设置SNMP配置 | |
| POST /digest/frmNetSnmp HTTP/1.1 | |
| 请求的正文内容  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "SNMPEnable": 0,  "ServerPort": 161,  "ReadCommunity": "",  "WriteCommunity": "",  "TrapHostIP": "",  "TrapHostPort": 162,  "SendCount": 3,  "SendInterval": 60  }  } |  |
| 响应的正文内容  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

### NAS

|  |  |
| --- | --- |
| 获取NAS配置 | |
| POST /digest/frmNasSetting HTTP/1.1 | |
| 请求的正文内容  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容  {  "Result": 0,  "Data": {  "Enable": 0,  "Type": 0,  "ServerAddr": "",  "FilePath": "",  "UserName": "",  "Password": ""  }  } | "Enable": Int型,是否启用NAS, 1-启用,0-不启用  "Type": Int型,挂载类型,0-NFS,1-SMB/CIFS  "ServerAddr": String型,服务器地址  "FilePath": String型,路径  "UserName": String型,用户名  "Password": String型,密码 |

|  |  |
| --- | --- |
| 设置NAS配置 | |
| POST /digest/frmNasSetting HTTP/1.1 | |
| 请求的正文内容  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "Enable": 0,  "Type": 0,  "ServerAddr": "",  "FilePath": "",  "UserName": "",  "Password": ""  }  } |  |
| 响应的正文内容  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

### 平台管理

|  |  |
| --- | --- |
| 获取平台管理配置 | |
| POST /digest/frmGetManagerHostsPara HTTP/1.1 | |
| 请求的正文内容  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容  {  "Result": 0,  "Data": {  "Protocols": [  {  "Type": 8,  "ProtocolName": "I8S",  "ItemCount": 0,  "ItemLables": []  },  {  "Type": 11,  "ProtocolName": "Fseye",  "ItemCount": 2,  "ItemLables": [  "Port",  "IfEnableSrvCheck"  ]  }  ],  "Platform": [  {  "Type": 8,  "EnableManagerHost": 0,  "Fixed": 0,  "OtherItemValues": []  },  {  "Type": 11,  "EnableManagerHost": 1,  "Fixed": 0,  "OtherItemValues": [  "",  ""  ]  }  ]  }  } | "Protocols": Object数组,协议列表  "Protocols"[]."Type": Int型,协议编号  "Protocols"[]."ProtocolName": String型,协议名  "Protocols"[]."ItemCount": Int型,参数个数  "Protocols"[]."ItemLables": String型数组,参数列表  "Platform": Object数组,平台信息  "Platform"[]."Type": Int型,协议类型,与Protocols列表里的Type对应  "Platform"[]."EnableManagerHost" Int型,是否启用,1-启用,0-不启用  "Platform"[]."Fixed": Int型,0-可修改,1-不可修改  "Platform"[]."OtherItemValues":String数组,其他项的值,与Protocols列表里的ItemLables对应 |

|  |  |
| --- | --- |
| 设置平台管理配置 | |
| POST /digest/frmGetManagerHostsPara HTTP/1.1 | |
| 请求的正文内容  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "Platform": [  {  "Type": 11,  "EnableManagerHost": 1,  "Fixed": 0,  "OtherItemValues": [  "",  ""  ]  },  {  "Type": 8,  "EnableManagerHost": 0,  "Fixed": 0,  "OtherItemValues": []  }  ]  }  } |  |
| 响应的正文内容  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

### P2P

#### P2P配置

|  |  |
| --- | --- |
| 获取P2P配置 | |
| POST /digest/frmP2PCfg HTTP/1.1 | |
| 请求的正文内容  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容  {  "Result": 0,  "Data": {  "EnableP2P": 0,  "EnableAlarm": 0,  "SerialNumber": "02d00123a8ec7c024a4d",  "AndroidAPKURI": "https://play.google.com/store/apps/details  ?id=com.ml.cloudeye&rdid=com.ml.cloudeye",  "IOSAPKURI": "https://play.google.com/store/apps/details  ?id=com.ml.cloudeye&rdid=com.ml.cloudeye"  }  } | "EnableP2P": Int型,是否启用P2P, 1-启用,0-不启用  "EnableAlarm": Int型, 是否启用报警推送, 1-启用,0-不启用  "SerialNumber": String型,设备序列号  "AndroidAPKURI": String型,Android APP下载地址  "IOSAPKURI": String型,IOS APP下载地址 |

|  |  |
| --- | --- |
| 设置P2P配置 | |
| POST /digest/frmP2PCfg HTTP/1.1 | |
| 请求的正文内容  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "EnableP2P": 0,  "EnableAlarm": 0  }  } |  |
| 响应的正文内容  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

#### 获取P2P服务在线状态

|  |  |
| --- | --- |
| 获取P2P服务在线状态 | |
| POST /digest/frmP2PState HTTP/1.1 | |
| 请求的正文内容  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容  {  "Result": 0,  "Data": {  "p2p\_state": 0  }  } | "p2p\_state": Int型,P2P服务在线状态, 1-离线,2-在线 |

### TURN服务

#### TURN服务配置

|  |  |
| --- | --- |
| 获取TURN服务配置 | |
| POST /digest/frmTurnServerCfg HTTP/1.1 | |
| 请求的正文内容  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容  {  "Result": 0,  "Data": {  "Enable": 0,  "ServerHost": "10.0.0.5",  "ServerPort": 6000,  "SN": "02d00123a8ec7c024a4d"  }  } | "Enable": Int型,是否启用TURN服务, 1-启用,0-不启用  "ServerPort": Int型, TURN服务器端口  "ServerHost": String型,TURN服务器地址  "SN": String型,设备序列号 |

|  |  |
| --- | --- |
| 设置TURN服务配置 | |
| POST /digest/frmTurnServerCfg HTTP/1.1 | |
| 请求的正文内容  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "Enable": 0,  "ServerHost": "10.0.0.5",  "ServerPort": 6000  }  } |  |
| 响应的正文内容  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

#### 获取TURN服务状态

|  |  |
| --- | --- |
| 获取TURN服务状态 | |
| POST /digest/frmTurnServerState HTTP/1.1 | |
| 请求的正文内容  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容  {  "Result": 0,  "Data": {  "Status": 2,  "SuperServer": "10.0.0.5",  "TurnServer": "10.0.0.25"  }  } | "Status": Int型, TURN服务在线状态, 1-离线,2-在线  "SuperServer": String型,SUPER服务器地址,Status为1时不返回此字段  "TurnServer": String型,TURN服务器地址,Status为1时不返回此字段 |

### GB28181

|  |  |
| --- | --- |
| 获取GB28181配置 | |
| POST /digest/frmParaPlatform28181 HTTP/1.1 | |
| 请求的正文内容  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容  {  "Result": 0,  "Data": {  "GB28181CFG": {  "SIP\_Code": "",  "SIP\_Zone": "",  "SIP\_DeviceID": "",  "SIP\_Password": "",  "SIP\_IP": "",  "SIP\_Port": 0,  "SIP\_Code2": "",  "SIP\_Zone2": "",  "SIP\_Password2": "",  "SIP\_IP2": "",  "SIP\_Port2": 0,  "Local\_Port": 0,  "Expires": 0,  "Keepalive": 0,  "KeepaliveCnt": 0,  "MulticastEnable": 0,  "MulticastAddr": "",  "MulticastPort": 0  },  "ChanNum": 1,  "AlarmNum": 2,  "ChanInfo": [  {  "Index": 0,  "ChanID": "",  "Level": 0  }  ],  "AlarmInfo": [  {  "Index": 0,  "ChanID": "",  "Level": 0  },  {  "Index": 1,  "ChanID": "",  "Level": 0  }  ],  "streamType":0  }  } | "SIP\_Code": String型,SIP服务器编号  "SIP\_Zone": String型,SIP服务器域  "SIP\_DeviceID": String型,设备编号  "SIP\_Password": String型,注册密码  "SIP\_IP": String型,SIP服务器IP  "SIP\_Port": Int型,SIP服务器端口  "SIP\_Code2": String型,SIP服务器编号2  "SIP\_Zone2": String型,SIP服务器域2  "SIP\_Password2": String型,注册密码2  "SIP\_IP2": String型,SIP服务器IP2  "SIP\_Port2": Int型,SIP服务器端口2  "Local\_Port": Int型,本地端口  "Expires": Int型,注册有效期(秒)  "Keepalive": Int型,心跳周期(秒)  "KeepaliveCnt": Int型,最大心跳超时次数  "MulticastEnable": Int型,是否启用多播,1-启用,0-不启用  "MulticastAddr": String型,多播地址  "MulticastPort": Int型,多播端口  "ChanNum": Int型,通道数量  "AlarmNum": Int型,报警数量  "ChanInfo": Object数组,通道信息  "ChanInfo"[]."Index": Int型,通道索引  "ChanInfo"[]."ChanID": String型,通道编号  "ChanInfo"[]."Level": Int型,报警级别  "AlarmInfo": Object数组,报警信息  "AlarmInfo"[]."Index": Int型,报警索引  "AlarmInfo"[]."ChanID": String型,报警编号  "AlarmInfo"[]."Level": Int型,报警级别  "streamType": Int型,编码类型,0-主码流,1-子码流,3-第三码流, 4-第四码流, 5-第五码流. |

|  |  |
| --- | --- |
| 设置GB28181配置 | |
| POST /digest/frmParaPlatform28181 HTTP/1.1 | |
| 请求的正文内容  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "GB28181CFG": {  "SIP\_Code": "",  "SIP\_Zone": "",  "SIP\_DeviceID": "",  "SIP\_Password": "",  "SIP\_IP": "",  "SIP\_Port": 0,  "SIP\_Code2": "",  "SIP\_Zone2": "",  "SIP\_Password2": "",  "SIP\_IP2": "",  "SIP\_Port2": 0,  "Local\_Port": 0,  "Expires": 0,  "Keepalive": 0,  "KeepaliveCnt": 0,  "MulticastEnable": 0,  "MulticastAddr": "",  "MulticastPort": 0  },  "ChanNum": 1,  "AlarmNum": 2,  "ChanInfo": [  {  "Index": 0,  "ChanID": "",  "Level": 0  }  ],  "AlarmInfo": [  {  "Index": 0,  "ChanID": "",  "Level": 0  },  {  "Index": 1,  "ChanID": "",  "Level": 0  }  ],  "streamType":0  }  } |  |
| 响应的正文内容  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

### GB35114

|  |  |
| --- | --- |
| 获取GB35114配置 | |
| POST /digest/frmParaPlatform35114 HTTP/1.1 | |
| 请求的正文内容  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容  {  "Result": 0,  "Data": {  "Enable": 0,  "LocalPort": 0,  "Expires": 0,  "Keepalive": 0,  "KeepAliveMaxCount": 0,  "SipIP": "",  "SipDeviceID": "",  "SipZone": "",  "SipPort": 0,  "Device\_ID": "",  "Domain\_Name": "",  "Password": "",  "Svr\_Ip": "",  "Svr\_Port": 0,  "ChanNum": 0,  "AlarmNum": 0,  "ChanInfo": [  {  "ChanID": ""  }  ],  "AlarmInfo": [  {  "AlarmID": "",  "AlarmLevel": 0  },  {  "AlarmID": "",  "AlarmLevel": 0  }  ]  }  } | "Enable": Int型,是否启用GB35114,1-启用,0-不启用  "LocalPort": Int型,本地端口  "Expires": Int型,注册有效期(秒)  "Keepalive": Int型,心跳周期(秒)  "KeepaliveMaxCount": Int型,最大心跳超时次数  "SipIP": String型,SIP服务器IP  "SipDeviceID": String型,SIP设备编号  "SipZone": String型,SIP服务器域  "SipPort": Int型,SIP服务器端口  "Device\_ID": String型,设备编号  "Domain\_Name": String型,域名  "Password": String型,注册密码  "Svr\_Ip": String型,服务器IP  "Svr\_Port": Int型,服务器端口  "ChanNum": Int型,通道数量  "AlarmNum": Int型,报警数量  "ChanInfo": Object数组,通道信息  "ChanInfo"[]."ChanID": String型,通道编号  "AlarmInfo": Object数组,报警信息  "AlarmInfo"[]."ChanID": String型,报警编号  "AlarmInfo"[]."Level": Int型,报警级别 |

|  |  |
| --- | --- |
| 设置GB35114配置 | |
| POST /digest/frmParaPlatform35114 HTTP/1.1 | |
| 请求的正文内容  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "Enable": 0,  "LocalPort": 0,  "Expires": 0,  "Keepalive": 0,  "KeepAliveMaxCount": 0,  "SipIP": "",  "SipDeviceID": "",  "SipZone": "",  "SipPort": 0,  "Device\_ID": "",  "Domain\_Name": "",  "Password": "",  "Svr\_Ip": "",  "Svr\_Port": 0,  "ChanNum": 0,  "AlarmNum": 0,  "ChanInfo": [  {  "ChanID": ""  }  ],  "AlarmInfo": [  {  "AlarmID": "",  "AlarmLevel": 0  },  {  "AlarmID": "",  "AlarmLevel": 0  }  ]  }  } |  |
| 响应的正文内容  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

### 韦根

|  |  |
| --- | --- |
| 获取韦根配置 | |
| POST /digest/frmWiegandCfg HTTP/1.1 | |
| 请求的正文内容  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容  {  "Result": 0,  "Data": {  "MaxChanNum": 1,  "WiegandInfoList": [  {  "Device": 0,  "DataLen": 26,  "Mode": 0,  "CommonId": 11111  }  ]  }  } | "MaxChanNum": Int型,最大韦根通道数  "WiegandInfoList": Object型数组, 韦根信息列表  "Device": Int型,通道  "DataLen": Int型,数据长度,26,34  "Mode": Int型,模式,1-扩展,解CommonId的低24位,0-不扩展,解CommonId的低16位  "CommonId": Int型,公共韦根号 |

|  |  |
| --- | --- |
| 设置韦根配置 | |
| POST /digest/frmWiegandCfg HTTP/1.1 | |
| 请求的正文内容  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "WiegandInfoList": [  {  "Device": 0,  "DataLen": 26,  "Mode": 0,  "CommonId": 11111  }  ]  }  } |  |
| 响应的正文内容  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

### Smart1400

#### Smart1400配置

|  |  |
| --- | --- |
| 获取Smart1400协议配置 | |
| POST /digest/frmSmart1400Cfg HTTP/1.1 | |
| 请求的正文内容  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容  {  "Result": 0,  "Data": {  "Enable": 0,  "ServerAddr": "http://0.0.0.0",  "DeviceId": "100004",  "EventListMaxLen": 8,  "Username": "admin",  "Password": "123456",  "Auth": 2,  "HeartInterval": 60,  "TimeInterval": 7200  }  } | "Enable": Int型,是否启用Smart1400协议, 1-启用,0-不启用,  "ServerAddr": String型,服务器地址  "DeviceId": String型,设备ID  "EventListMaxLen": Int型,  "Username": String型,用户名  "Password": String型,密码  "Auth": Int型,认证方式,0-不认证,1-base64,2-digest,3-base64+digest  "HeartInterval": Int型,心跳间隔,单位:秒  "TimeInterval": Int型,校时间隔,单位:秒 |

|  |  |
| --- | --- |
| 设置Smart1400协议配置 | |
| POST /digest/frmSmart1400Cfg HTTP/1.1 | |
| 请求的正文内容  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "Enable": 0,  "ServerAddr": "http://0.0.0.0",  "DeviceId": "100004",  "EventListMaxLen": 8,  "Username": "admin",  "Password": "123456",  "Auth": 2,  "HeartInterval": 60,  "TimeInterval": 7200  }  } |  |
| 响应的正文内容  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

#### Smart1400服务器地址测试

|  |  |
| --- | --- |
| Smart1400服务器地址测试 | |
| POST /digest/frmSmart1400AddrTest HTTP/1.1 | |
| 请求的正文内容  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {  "ServerAddr": "10.3.0.103:8086"  }  } |  |
| 响应的正文内容  {  "Result": 0,  "Data": {  "Code": -1,  "ErrorInfo": ""  }  } | "Code":Int型,测试结果,0-地址可用,其他-地址不可用  "ErrorInfo":String型,错误详细信息,Code不为0时才会返回此字段 |

### 4G/5G

#### 4G/5G配置

|  |  |
| --- | --- |
| 获取4G/5G配置 | |
| POST /digest/frmNetLtepara HTTP/1.1 | |
| 请求的正文内容  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容  {  "Result": 0,  "Data": {  "IsSupported": 0,  "Enable": 0,  "Info": {  "IMEI": "NA",  "InsertedStatus": 2,  "NetInfo": "NA",  "CCID": "",  "ServiceInfo": "NA",  "SignalStrength": 0,  "LinkStatus": 0,  "IpAddrV4": "NA",  "DNS1V4": "",  "DNS2V4": "",  "IpAddrV6": "NA"  "DNS1V6": "",  "DNS2V6": ""  }  }  } | "Result": Int型,0-正常,-1-不支持4G/5G  "IsSupported": Int型,是否支持4G,1-支持,0-不支持  "Enable": Int型,是否启用4G/5G, 1-启用,0-不启用,  "IMEI": String型,IMEI码  "InsertedStatus": Int型,SIM卡状态,0-未插卡,1-已插卡,2-未知3-N/A  "CCID": String型,CCID  "NetInfo": String型,网络信息  "ServiceInfo": String型,运营商信息  "SignalStrength": Int型,信号强度  "LinkStatus": Int型,连接状态,0-网络异常,1-网络正常,2-无服务  "IpAddrV4": String型,ipv4地址  "DNS1V4": String型, ipv4 ddns1  "DNS2V4": String型, ipv4 ddns2  "IpAddrV6": String型,ipv6地址  "DNS1V6": String型, ipv6 ddns1  "DNS2V6": String型, ipv6 ddns2 |

|  |  |
| --- | --- |
| 设置4G/5G配置 | |
| POST /digest/frmNetLtepara HTTP/1.1 | |
| 请求的正文内容  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "Enable": 0  }  } |  |
| 响应的正文内容  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

#### 获取4G/5G配置简易接口

|  |  |
| --- | --- |
| 获取4G/5G配置简易接口 | |
| POST /digest/frmGetLteCardInfo HTTP/1.1 | |
| 请求的正文内容  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容  {  "Result": 0,  "Data": {  "IMEI": "",  "CCID": "",  "LinkStatus": 0,  "SignalStrength": 0,  "ErrorCode": 0  }  } | "IMEI": String型,IMEI码  "CCID": String型,CCID  "LinkStatus": Int型,连接状态,0-网络异常,1-网络正常,2-无服务  "SignalStrength": Int型,信号强度,范围:0-100  "ErrorCode": Int型,错误码, 0: 正常, -1: 模块异常, -2: SIM卡未就绪, -3: 网络服务未就绪 |

### WIFI

|  |  |
| --- | --- |
| 获取WIFI配置 | |
| POST /digest/frmWifiPara HTTP/1.1 | |
| 请求的正文内容  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容 | |
| 1. 关闭:   {  "Result": 0,  "Data": {  "WorkStatus": {  "WorkMode": -1,  "WorkInfo": {}  }  }  } | "Result": Int型,0-正常,-1-不支持WIFI  "WorkMode": Int型,工作模式,-1-关闭,0-热点模式,1-无线模式 |
| 1. AP模式:   {  "Result": 0,  "Data": {  "WorkStatus": {  "WorkMode": 0,  "WorkInfo": {  "Ssid": "IPC-AP-0000-50f8",  "Psk": "12345678",  "IP": "192.168.10.1",  "Mac": "b4:04:18:31:50:f8",  "AuthType": 6  }  },  "WifiStaList": [  {  "IP": "192.168.10.103",  "Mac": "fc:d7:33:d0:22:bc"  }  ]  }  } | "WorkInfo": Object型,WIFI信息  "WorkInfo"."Ssid": String型,WIFI名称  "WorkInfo"."Psk": String型,WIFI密码  "WorkInfo"."IP": String型,IP地址  "WorkInfo"."Mac": String型,物理地址  "WorkInfo"."AuthType": Int型,认证类型  0-none  1-wep  2-wpa  4-wpa2  6-wpa/wpa2  "WifiStaList": Object型数组,已连接设备列表  "WifiStaList"[]."IP": String型,已连接设备IP地址  "WifiStaList"[]."Mac": String型,已连接设备物理地址 |
| 1. STA模式   {  "Result": 0,  "Data": {  "WorkStatus": {  "WorkMode": 1,  "WorkInfo": {  "Ssid": "",  "StaStatus": 3,  "IpAddr": "192.168.1.107",  "Mac": "192.168.1.107",  "Signal": 0  }  },  "WifiScanList": [  {  "Ssid": "TP-LINK\_22BC",  "Mac": "fc:d7:33:d0:22:bc",  "AuthType": 14,  "Signal": 98  },  {  "Ssid": "ZKTECO",  "Mac": "78:44:fd:b0:89:e9",  "AuthType": 14,  "Signal": 92  }  ],  "WifiApList": [  {  "AuthCfg": {  "Ssid": "TP-LINK\_22BC",  "Psk": "12345678",  "Mac": "",  "AuthType": 6  },  "IpAddrCfg": {  "Dhcp": 0,  "IP": "192.168.1.107",  "NetMask": "255.255.255.0",  "GateWay": "192.168.1.1"  }  }  ]  }  } | "WorkInfo": Object型,已连接WIFI信息  "WorkInfo"."Ssid": String型, 已连接WIFI名称  "WorkInfo"."StaStatus": Int型, 连接状态   1. 未连接 2. 接口禁用 3. 无效 4. 扫描中 5. 认证中 6. 配对中 7. 配对成功 8. 四次握手 9. 组播握手 10. 已连接   "WorkInfo"."IpAddr": String型,IP地址  "WorkInfo"."Mac": String型,物理地址  "WorkInfo"."Signal": Int型,信号强度,单位:asu  "WifiScanList": Object型数组,扫描到的WIFI列表  "WifiScanList"[]."Ssid": String型,WIFI名称  "WifiScanList"[]."Mac": String型,WIFI物理地址  "WifiScanList"[]."AuthType": Int型,WIFI认证类型,取AuthType%8后的值:  0-none  1-wep  2-wpa  4-wpa2  6-wpa/wpa2  "WifiScanList"[]."Signal": Int型,WIFI信号强度,单位:asu  "WifiApList": Object型,已保存WIFI列表  "AuthCfg"."Ssid": String型, 已保存WIFI名称  "AuthCfg"."Mac": String型,已保存WIFI物理地址  "AuthCfg"."AuthType": Int型,已保存WIFI认证类型,取AuthType%8后的值:  0-none  1-wep  2-wpa  4-wpa2  6-wpa/wpa2  "AuthCfg"."Signal": Int型,已保存WIFI信号强度,单位:asu  "IpAddrCfg"."Dhcp": Int型,是否自动获取IP地址,1-是,0-否  "IpAddrCfg"."IP": String型,IP地址  "IpAddrCfg"."NetMask": String型,子网掩码  "IpAddrCfg"."GateWay": String型,网关 |

|  |  |
| --- | --- |
| 设置WIFI配置 | |
| POST /digest/frmWifiPara HTTP/1.1 | |
| 请求的正文内容 | |
| 1. 设置关闭WIFI:   {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "WorkStatus": {  "WorkMode": -1  }  }  } |  |
| 1. AP模式设置热点配置   {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "WorkStatus": {  "WorkMode": 1,  "WorkInfo": {  "Ssid": "IPC-AP-0000-50f8",  "Psk": "12345678",  "AuthType": 6  }  }  }  } |  |
| 1. STA模式连接WIFI:   {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "ModifyAp": {  "AuthCfg": {  "Ssid": "TP-LINK\_22BC",  "Psk": "12345678",  "Mac": "",  "AuthType": 6  },  "IpAddrCfg": {  "Dhcp": 0,  "IP": "192.168.1.107",  "NetMask": "255.255.255.0",  "GateWay": "192.168.1.1"  }  }  }  } |  |
| 1. STA模式删除WIFI   {  "Type": 2,  "Dev": 1,  "Ch": 1,  "Data": {  "ModifyAp": {  "AuthCfg": {  "Ssid": "TP-LINK\_22BC",  "Psk": "12345678",  "Mac": "",  "AuthType": 6  },  "IpAddrCfg": {  "Dhcp": 0,  "IP": "192.168.1.107",  "NetMask": "255.255.255.0",  "GateWay": "192.168.1.1"  }  }  }  } |  |
| 响应的正文内容  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

### SIP

#### SIP配置

|  |  |
| --- | --- |
| 获取SIP配置 | |
| POST /digest/frmNetSipPara HTTP/1.1 | |
| 请求的正文内容  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容  {  "Result": 0,  "Data": {  "Enable": 0,  "SIPConfig": [  {  "SIPName": "SIP#0",  "Enable": 0,  "StreamType": 0,  "CallNumber": "",  "CallNumber1": "",  "CallDelayMSec": 0,  "RegServer": {  "Name": "",  "Number": "",  "UserName": "",  "PassWord": "",  "IP": "",  "Port": 0  },  "SIPServer": {  "IP": "",  "Port": 0  },  "OnHoldMusicCfg": {  "SoundName": "",  "Enable": 0,  "Playtimes": 0  },  "DTMF": {  "Key\_1": "",  "Enable": 0,  "DelayTime": 0  }  }  ]  }  } | "Enable": Int型,是否启用SIP, 1-启用,0-不启用  "SIPConfig": Object型数组,长度为几就是支持多少个账号  "SIPConfig"[]."SIPName": String型,SIP名  "SIPConfig"[]."Enable": Int型,是否启用当前账号, 1-启用,0-不启用,  "SIPConfig"[]."StreamType": Int型,码流类型  "SIPConfig"[]."CallNumber": String型,被叫号码1  "SIPConfig"[]."CallNumber1": String型,被叫号码2,CallNumber返回几个就是支持几个  "SIPConfig"[]."CallDelayMSec": Int型,呼叫延时,单位:毫秒  "RegServer": Object型,注册服务器参数  "RegServer"."Name": String型,显示名  "RegServer"."Number": String型,号码  "RegServer"."UserName": String型,用户名  "RegServer"."PassWord": String型,密码  "RegServer"."IP": String型,注册服务器IP地址  "RegServer"."Port": Int型,注册服务器端口  "SIPServer"."IP": String型,SIP服务器地址  "SIPServer"."Port": Int型,SIP服务器端口  "OnHoldMusicCfg"."SoundName": String型,音频名称  "OnHoldMusicCfg"."Enable": Int型,是否自定义铃声,1-是,0-否  "OnHoldMusicCfg"."Playtimes": Int型,铃声播放次数  "DTMF"."Key\_1": String型,DTMF按键,长度为1,支持数字\*#  "DTMF"."Enable": Int型,是否启用DTMF,1-启用,0-不启用  "DTMF"."DelayTime": Int型,DTMF延时时间 |

|  |  |
| --- | --- |
| 设置SIP配置 | |
| POST /digest/frmNetSipPara HTTP/1.1 | |
| 请求的正文内容  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "Enable": 0,  "SIPConfig": [  {  "SIPName": "SIP#0",  "Enable": 0,  "StreamType": 0,  "CallNumber": "",  "CallNumber1": "",  "CallDelayMSec": 0,  "RegServer": {  "Name": "",  "Number": "",  "UserName": "",  "PassWord": "",  "IP": "",  "Port": 0  },  "SIPServer": {  "IP": "",  "Port": 0  },  "OnHoldMusicCfg": {  "SoundName": "1.wav",  "AudioData": "xxx",  "Enable": 0,  "Playtimes": 0  },  "DTMF": {  "Key\_1": "",  "Enable": 0,  "DelayTime": 0  }  }  ]  }  } | "OnHoldMusicCfg"."AudioData": String型,自定义铃声base64数据,自支持WAV格式音频 |
| 响应的正文内容  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

#### 手动触发Sip报警输出

|  |  |
| --- | --- |
| 手动触发Sip报警输出 | |
| GET /digest/frmDTMFTrigger?Username=admin&PasswordDigest=xx&Created=xx&Nonce=xxx HTTP/1.1 | |
|  | "Username":String型,登录设备的用户名  "PasswordDigest":String型,密码字符串,加密规则base64(sha1(Nonce + Created + Password)  "Created":String型,客户端生成token的时间(DateTime ISO格式)  "Nonce":String型,生成规则base64(sha1(Created + secret key)) |
| 响应的正文内容(text格式文本,非json格式)  {Result:0,StatusCode:"Success"} |  |

### NTP

|  |  |
| --- | --- |
| 获取NTP配置 | |
| POST /digest/frmNetNtpPara HTTP/1.1 | |
| 请求的正文内容  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容  {  "Result": 0,  "Data": {  "EnableNTP": 0,  "NTPServer": "",  "TimeInterval": 0,  "NTPPort": 123,  "TimeOffsetHour": 8,  "TimeOffsetMinute": 0,  "CurrentTimeOffset": 86400  }  } | "EnableNTP": Int型,是否启用NTP, 1-启用,0-不启用,  "NTPServer": String型,NTP服务器地址  "TimeInterval": Int型,校时间隔,单位:秒  "NTPPort": Int型,NTP端口  "TimeOffsetHour": Int型,与UTC(GMT0)的小时偏移  "TimeOffsetMinute": Int型,与UTC(GMT0)的分钟偏移  "CurrentTimeOffset": Int型,目前设备生效时区与UTC(GMT0)时间的偏移(单位:秒,包含夏令时). GMT+N时区时该值为正数,GMT-N时区时该值为负数. |

|  |  |
| --- | --- |
| 设置NTP配置 | |
| POST /digest/frmNetNtpPara HTTP/1.1 | |
| 请求的正文内容  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "EnableNTP": 0,  "NTPServer": "",  "TimeInterval": 0,  "NTPPort": 123,  "TimeOffsetHour": 8,  "TimeOffsetMinute": 0  }  } |  |
| 响应的正文内容  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

### TELNET

|  |  |
| --- | --- |
| 获取TELNET配置 | |
| POST /digest/frmNetTelnetPara HTTP/1.1 | |
| 请求的正文内容  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容  {  "Result": 0,  "Data": {  "Enable": 1,  "Port": 23,  "Password": "\*\*\*\*\*\*"  }  } | "Enable": Int型,是否启用TELNET, 1-启用,0-不启用,  "Port": Int型,TELNET端口  "Password": String型,TELNET密码,不会返回真实密码 |

|  |  |
| --- | --- |
| 设置TELNET配置 | |
| POST /digest/frmNetTelnetPara HTTP/1.1 | |
| 请求的正文内容  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "Enable": 1,  "Port": 23,  "Password": "\*\*\*\*\*\*"  }  } |  |
| 响应的正文内容  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

### ONVIF(全网通和免验证)

|  |  |
| --- | --- |
| 获取ONVIF配置 | |
| POST /digest/frmOnvifPara HTTP/1.1 | |
| 请求的正文内容  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容  {  "Result": 0,  "Data": {  "adaptiveIp": 0,  "authEnable": 1  }  } | "adaptiveIp": Int型,是否启用全网通, 1-启用,0-不启用,  "authEnable": Int型,是否免验证, 1-验证,0-免验证 |

|  |  |
| --- | --- |
| 设置ONVIF配置 | |
| POST /digest/frmOnvifPara HTTP/1.1 | |
| 请求的正文内容  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "adaptiveIp": 0,  "authEnable": 1  }  } |  |
| 响应的正文内容  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

### 阿里IoT

#### 阿里IoT配置

|  |  |
| --- | --- |
| 获取阿里IoT配置 | |
| POST /digest/frmAliIoTCfg HTTP/1.1 | |
| 请求的正文内容  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容  {  "Result": 0,  "Data": {  "EnableIoT": 1,  "EnableAlarm": 0,  "Version": "V01.02.02",  "BuildDate": "2020-10-26-18",  "QRCode": "http://cveye-service.p2p-platform.com:10000/stati  c/html/cloudAI.html?serialNo=03760002000ccd766f9f&active=1"  }  } | "EnableIoT": Int型,是否启用IoT, 1-启用,0-不启用,  "EnableAlarm": Int型, 是否启用报警推送, 1-启用,0-不启用  "Version": String型,只读字段,IoT版本信息  "BuildDate": String型,只读字段,IoT编译日期  "QRCode": String型,只读字段,手机APP下载地址和设备添加地址 |

|  |  |
| --- | --- |
| 设置阿里IoT配置 | |
| POST /digest/frmAliIoTCfg HTTP/1.1 | |
| 请求的正文内容  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "EnableIoT": 1,  "EnableAlarm": 0  }  } |  |
| 响应的正文内容  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

#### 获取阿里IoT服务状态

|  |  |
| --- | --- |
| 获取阿里IoT服务状态 | |
| POST /digest/frmAliIoTState HTTP/1.1 | |
| 请求的正文内容  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容  {  "Result": 0,  "Data": {  "BurnRunState": 0,  "BurnState": 0,  "IoTRunState": 0  }  } | "BurnRunState": Int型,只读字段,烧录服务器连接状态,1-已连接,0-未连接  "BurnState": Int型,只读字段,IoT激活状态, 1-已激活,0-未激活  "IoTRunState": Int型,只读字段,设备在线状态, 1-在线,0-离线 |

注:设备未激活时,烧录服务器已连接上才能进行设备激活操作; 设备激活后,烧录服务器会主动断开连接.

#### 阿里IoT服务重启

|  |  |
| --- | --- |
| 重启阿里IoT服务 | |
| POST /digest/frmAliIoTReboot HTTP/1.1 | |
| 请求的正文内容  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

注:重启IoT服务后，IoT服务重新上线预计需要等待15-30s，接口调用者应给用户友好提示。

#### 阿里IoT解绑

|  |  |
| --- | --- |
| 阿里IoT解绑 | |
| POST /digest/frmAliIoTUnbinding HTTP/1.1 | |
| 请求的正文内容  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

注:解绑后默认会重启IOT服务，等待15-30s上线后才可重新绑定，接口调用者应给用户友好提示。

### 获取默认路由

|  |  |
| --- | --- |
| 获取默认路由 | |
| POST /digest/frmGetDefaultRoute HTTP/1.1 | |
| 请求的正文内容  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容  {  "Result": 0,  "Data": {  "NetType": 1,  "DefaultRoute": "eth0"  }  } | "NetType": Int型,默认路由网络类型, 0-未知, 1-有线, 2-wifi, 3-4g/5g  "DefaultRoute": String型,默认路由名称 |

## 媒体

### 视频制式

#### 获取视频制式能力集

|  |  |
| --- | --- |
| 获取视频制式能力集,该接口与视频制式v2接口一起使用 | |
| POST /digest/frmGetPNFormatAbility HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "FormatList": [  {  "W": 3840,  "H": 2160,  "Fps": 30,  "Format": 0  },  ……  ]  }  } | "FormatList": Object型数组,支持的视频制式列表.  "W": Int型,视频宽度.  "H": Int型,视频高度.  "Fps": Int型,视频帧数.Fps%30为0为NTSC,不为0为PAL  "Format": Int型,视频制式序号. |

#### 视频制式

|  |  |
| --- | --- |
| 获取视频制式配置 | |
| POST /digest/frmVideoFormatPara HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "VideoFormat": 1  }  } | "VideoFormat": Int型,视频制式,1-NTSC,2-PAL. |

|  |  |
| --- | --- |
| 设置视频制式配置 | |
| POST /digest/frmVideoFormatPara HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "VideoFormat": 1  }  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "StatusCode":"Operation Ok"  }  } |  |

#### 视频制式\_V2

|  |  |
| --- | --- |
| 获取视频制式配置 | |
| POST /digest/frmVideoFormatPara\_v2 HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "VideoFormat": 1  }  } | "VideoFormat": Int型,视频制式,取值参考视频制式能力集 |

|  |  |
| --- | --- |
| 设置视频制式配置 | |
| POST /digest/frmVideoFormatPara\_v2 HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "VideoFormat": 0,  "Fps": 30,  "W": 1920,  "H": 1080  }  } | "VideoFormat": Int型,视频制式,取值参考视频制式能力集  "Fps": Int型,视频帧数,取值参考视频制式能力集.  "W": Int型,视频宽度,取值参考视频制式能力集.  "H": Int型,视频高度,取值参考视频制式能力集. |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "StatusCode":"Operation Ok"  }  } |  |

### 音频报警输出

#### 音频报警输出配置

|  |  |
| --- | --- |
| 获取音频报警输出配置 | |
| POST /digest/frmAudioAlarmCfg HTTP/1.1 | |
| 请求的正文内容:  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容:  {  "Result": 0,  "Data": {  "ResList": [  {  "Id": 1,  "Name": "",  "Times": 0  },  ……  ]  }  } | "ResList": Object型数组,音频配置列表,长度为8  "Id":Int型,音频序号.  "Name": String型,音频名称.  "Times": Int型,播放次数,100为循环播放. |

|  |  |
| --- | --- |
| 设置声音报警输出配置 | |
| POST /digest/frmAudioAlarmCfg HTTP/1.1 | |
| 请求的正文内容:  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "ResList": [  {  "Id": 1,  "Name": "水深危险",  "Times": 0,  "AudioDataType": 0,  "AudioData": "xxx"  },  ……  ]  }  } | "AudioDataType":Int型,音频数据格式,0-g711u,1-wav  "AudioData":16bit 8K 单声道Base64数据. |
| 响应的正文内容:  {  "Result": 0,  "Data": {  "StatusCode":"Operation Ok"  }  } |  |

#### 音频报警试听

|  |  |
| --- | --- |
| 音频报警试听 | |
| POST digest/frmSpeechAlarmCfg HTTP/1.1 | |
| 请求的正文内容:  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "SoundId": 1  }  } | "SoundId":Int型,音频序号 |
| 响应的正文内容  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

### 音频参数

#### 获取音频参数能力集

|  |  |
| --- | --- |
| 获取音频参数能力集 | |
| POST /digest/frmAudioParaAbility HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "AencChanNum": 1,  "AdecChanNum": 1,  "AencFormatList": [  {  "Name": "G711A",  "Value": 1,  "FrameLenList": [  80,  160,  240,  320,  480  ]  },  ……  ]  }  } | "AencChanNum": Int型,Aenc通道个数  "AdecChanNum": Int型,Adec通道个数  "AencFormatList":Object型数组,Aenc支持格式列表  "Name": String型,格式名  "Value":Int型,格式序号  "FrameLenList": Int型数组,音频帧长度列表 |

#### 音频参数配置

|  |  |
| --- | --- |
| 获取音频参数配置 | |
| POST /digest/frmAudioPara HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "AudioSource": 0,  "AudioInVol": 50,  "AudioOutVol": 50,  "AudioEncFormat": 2,  "AudioEnable": 1,  "FrameLen": 320  }  } | "AudioSource": Int型,音频源,0-线入,1-麦克风.  "AudioInVol": Int型,音频输入音量,范围0-100.  "AudioOutVol": Int型,音频输出音量,范围0-100.  "AudioEncFormat": Int型,音频编码格式,参考音频能力集  "AudioEnable": Int型,是否启用音频编码,1-启用,0-不启用  "FrameLen": Int型,音频帧长度,参考音频能力集 |

|  |  |
| --- | --- |
| 设置音频参数配置 | |
| POST /digest/frmAudioPara HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "AudioSource": 0,  "AudioInVol": 50,  "AudioOutVol": 50,  "AudioEncFormat": 2,  "AudioEnable": 1,  "FrameLen": 320  }  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

### 图像模式

#### 获取图像模式能力集

|  |  |
| --- | --- |
| 获取图像模式能力集 | |
| POST /digest/frmGetImageModeAbility HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "ImageList": [  {  "DesString": "5MP@15FPS",  "Value": 0  },  ……  ]  }  } | "ImageList": Object型数组,图像模式能力集.  "DesString": String型,描述.  "Value": Int型,值. |

#### 图像模式配置

|  |  |
| --- | --- |
| 获取图像模式配置 | |
| POST /digest/frmVideoImageModePara HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "DesString": "5MP@15FPS",  "Value": 0  }  } | "DesString": String型,描述,参考图像模式能力集  "Value": Int型,值,参考图像模式能力集 |

|  |  |
| --- | --- |
| 设置图像模式配置 | |
| POST /digest/frmVideoImageModePara HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "DesString": "5MP@15FPS",  "Value": 0  }  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

### 视频编码参数

#### 获取视频编码能力集

|  |  |
| --- | --- |
| 获取视频编码能力集 | |
| POST /digest/frmVideoCompressAbility HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {  "AbilityType": 0  }  } | "AbilityType": Int型,编码类型,0-主码流,1-子码流,3-第三码流. |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "BaudRate": [  [  "Main Stream",  0  ],  ……  ],  "Resolution": [  [  "D1(720\*576)",  6,  30  ],  ……  ],  "FrameRate": [  [  "1",  1  ],  ……  ],  "BitrateType": [  [  "VBR",  0  ],  ……  ],  "Bitrate": [  [  "16Kbps",  1  ],  ……  ],  "VideoEncType": [  [  "H264",  0  ],  ……  ],  "H264PROFILE": [  [  "Baseline",  0  ],  ……  ]  }  } | "BaudRate":数组,支持的码流类型列表  "BaudRate"[][0]:String型,码流类型描述  "BaudRate"[][1]:Int型,码流类型对应的值  "Resolution":数组,码流支持的分辨率列表  "Resolution"[][0]:String型,分辨率描述  "Resolution"[][1]:分辨率对应的值  "Resolution"[][2]:该分辨率支持的最大Fps  "FrameRate":数组,视频帧率列表  "FrameRate"[][0]:String型,帧率描述  "FrameRate"[][1]:Int型,帧率对应的值  "BitrateType":数组,位率类型列表  "BitrateType"[][0]: String型,位率类型描述,"VBR"-变码率,"CBR"-定码率  "BitrateType"[][1]:Innt型,位率类型对应的值  "Bitrate":数组,位率列表  "Bitrate"[][0]:String型,位率描述  "Bitrate"[][1]:Int型,位率对应的值  "VideoEncType":数组,编码类型列表; ["H264",1]表示:"H264编码描述;1-对应的值.  编码类型有:0-私有h.264 ;1-标准h.264; 2-标准mpeg4; 3-MJPEG;4-H265;5-SVAC.  "VideoEncType"[][0]:String型,编码类型描述  "VideoEncType"[][1]:Int型,编码类型对应的值  "H264PROFILE": 数组,编码级别  "H264PROFILE"[][0]:String型,编码级别描述  "H264PROFILE"[][1]:Int型,编码级别对应的值 |

#### 视频编码参数配置

|  |  |
| --- | --- |
| 获取视频编码参数配置 | |
| POST /digest/frmVideoIPCSetPara HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {  "AbilityType": 0  }  } | "AbilityType": Int型,编码类型,0-主码流,1-子码流,3-第三码流. |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "AbilityType": 0,  "PicQuality": 0,  "VideoFrameRate": 25,  "IFrameInterval": 25,  "BitrateType": 1,  "VideoEncType": 0,  "VideoH264Profile": 2,  "VideoBitrate": 23,  "Resolution": 0,  "StreamType": 1  }  } | "PicQuality": Int型,画像质量,当位率类型为定码率时,图像质量取决于码率;0-最好;1-次好;2-较好;3-一般;4-较差;5-差.  "VideoFrameRate":Int型,视频帧率,取值范围参考视频编码能力集  "IFrameInterval": Int型,I帧间隔.  "BitrateType": Int型,位率类型,取值范围参考视频编码能力集  "VideoEncType": Int型,编码类型,取值范围参考视频编码能力集  "VideoH264Profile": Int型,编码级别,取值范围参考视频编码能力集  "VideoBitrate": Int型,位率,取值范围参考视频编码能力集  "Resolution": Int型,分辨率; 取值范围参考视频编码能力集  "StreamType": Int型,码流类型:0-视频流;1-视频和音频流(复合流) |

|  |  |
| --- | --- |
| 设置视频编码参数配置 | |
| POST /digest/frmVideoIPCSetPara HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "AbilityType": 0,  "PicQuality": 0,  "VideoFrameRate": 25,  "IFrameInterval": 25,  "BitrateType": 1,  "VideoEncType": 0,  "VideoH264Profile": 2,  "VideoBitrate": 23,  "Resolution": 0,  "StreamType": 1  }  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

### 单行OSD

#### 单行OSD(带点阵)

|  |  |
| --- | --- |
| 获取单行OSD(带点阵)配置 | |
| POST /digest/frmSingleLineOSD HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 10,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "OSDLattice": {  "Enable": 1,  "PosX": 0,  "PosY": 0,  "ChanNameLocation": 0,  "FontName": "YouYuan",  "Text": "IpCamera0",  "MainStreamFontSize": 60,  "BitColor": "f8f8f8",  "SubStreamFontSize": 30,  "ThirdStreamFontSize": 20,  "EnableDateOSD": 1,  "DateOSDPosX": 973,  "DateOSDPosY": 0,  "TimeLocation": 0,  "DateOSDType": 0,  "DateOSDHourType": 0  }  }  } | "Enable": Int型,是否显示单行OSD,1-显示,0-不显示  "PosX": Int型,单行OSD左上角点的横坐标(0-1000).  "PosY": Int型,单行OSD左上角点的纵坐标(0-1000).  "ChanNameLocation": Int型,单行OSD位置  0-自定义,位置坐标为(PosX,PosY)  1-左上角  2-右上角  3-左下角  4-右下角.  "FontName": String型,单行OSD字体,取值范围:"SongTi","YaHei","HeiTi",  "YouYuan","Arial"  "Text": String型,单行OSD文本  "MainStreamFontSize": Int型,主码流OSD大小,取值范围:10,15,20,30,40,50,  60,70,80  "BitColor": String型,单行OSD颜色  "SubStreamFontSize": Int型,子码流OSD大小,取值范围:10,15,20,30,40  "ThirdStreamFontSize": Int型,第三码流OSD大小,取值范围:10,15,20,30,40,  50,60,70,80  "EnableDateOSD": Int型,是否显示日期OSD,1-显示,0-不显示  "DateOSDPosX" Int型,:日期OSD左上角点的横坐标(0-1000).  "DateOSDPosY": Int型,日期OSD左上角点的纵坐标(0-1000).  "TimeLocation": Int型,日期OSD位置  0--自定义,位置坐标为(DateOSDPosX,DateOSDPosY)  1--左上角  2--右上角  3--左下角  4--右下角.  "DateOSDType": Int型,日期格式,0-年月日,1-月日年,2-日月年  "DateOSDHourType": Int型,时间格式,0-24小时制,1-12小时制. |

|  |  |
| --- | --- |
| 设置单行OSD(带点阵)配置 | |
| POST /digest/frmSingleLineOSD HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 11,  "Dev": 1,  "Ch": 1,  "Data": {  "OSDLattice": {  "Enable": 1,  "PosX": 0,  "PosY": 0,  "ChanNameLocation": 0,  "FontName": "YouYuan",  "Text": "IpCamera0",  "MainStreamFontSize": 60,  "BitColor": "f8f8f8",  "SubStreamFontSize": 30,  "ThirdStreamFontSize": 20,  "EnableDateOSD": 1,  "DateOSDPosX": 973,  "DateOSDPosY": 0,  "TimeLocation": 0,  "DateOSDType": 0,  "DateOSDHourType": 0,  "TextLength": 9,  "Main": {  "BytesArray": "xxx",  "BytesLength": 5238,  "Width": 432,  "Height": 97  },  "Sub": {  "BytesArray": "xxx",  "BytesLength": 1296,  "Width": 216,  "Height": 48  },  "Third": {  "BytesArray": "xxx",  "BytesLength": 594,  "Width": 144,  "Height": 33  }  }  }  } | "TextLength": Int型,单行OSD长度  "BytesArray": String型,点阵字节数据,用base64编码.  "BytesLength": Int型,点阵字节数组长度.  "Width": Int型,像素宽度.  "Height": Int型,像素高度. |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

#### 单行OSD(不带点阵)

|  |  |
| --- | --- |
| 获取单行OSD(不带点阵)配置 | |
| POST /digest/frmSingleLineOSD HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "OSD": {  "IsShowChanName": 1,  "ChanNameTopLeftX": 0,  "ChanNameTopLeftY": 0,  "ChanNameLocation": 0,  "FontName": "YouYuan",  "FirstStreamOsdSize": 60,  "ChanName": "IpCamera0",  "BitColor": "f8f8f8",  "SecondStreamOsdSize": 30,  "ThirdStreamOsdSize": 20,  "IsShowOSD": 1,  "OSDTopLeftX": 342,  "OSDTopLeftY": 0,  "TimeLocation": 0,  "OSDType": 0,  "OSDHourType": 0  }  }  } | "IsShowChanName": Int型,是否显示单行OSD,1-显示,0-不显示  "ChanNameTopLeftX": Int型,单行OSD左上角点的横坐标(0-352).  "ChanNameTopLeftY": Int型,单行OSD左上角点的纵坐标(0-288).  "ChanNameLocation": Int型,单行OSD位置  0-自定义,位置坐标为(ChanNameTopLeftX,ChanNameTopLeftY)  1-左上角  2-右上角  3-左下角  4-右下角.  "FontName": String型,单行OSD字体,取值范围:"SongTi","YaHei","HeiTi",  "YouYuan","Arial"  "ChanName": String型,单行OSD文本  "FirstStreamOsdSize": Int型,主码流OSD大小,取值范围:10,15,20,30,40,50,  60,70,80  "BitColor": String型,单行OSD颜色  "SecondStreamOsdSize": Int型,子码流OSD大小,取值范围:10,15,20,30,40  "ThirdStreamFontSize": Int型,第三码流OSD大小,取值范围:10,15,20,30,40,  50,60,70,80  "IsShowOSD": Int型,是否显示日期OSD,1-显示,0-不显示  "OSDTopLeftX" Int型,:日期OSD左上角点的横坐标(0-352).  "OSDTopLeftY": Int型,日期OSD左上角点的纵坐标(0-288).  "TimeLocation": Int型,日期OSD位置  0--自定义,位置坐标为(DateOSDPosX,DateOSDPosY)  1--左上角  2--右上角  3--左下角  4--右下角.  "OSDType": Int型,日期格式,0-年月日,1-月日年,2-日月年  "OSDHourType": Int型,时间格式,0-24小时制,1-12小时制. |

|  |  |
| --- | --- |
| 设置单行OSD(不带点阵)配置 | |
| POST /digest/frmSingleLineOSD HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "OSD": {  "IsShowChanName": 1,  "ChanNameTopLeftX": 0,  "ChanNameTopLeftY": 0,  "ChanNameLocation": 0,  "FontName": "YouYuan",  "FirstStreamOsdSize": 60,  "ChanName": "IpCamera0",  "BitColor": "f8f8f8",  "SecondStreamOsdSize": 30,  "ThirdStreamOsdSize": 20,  "IsShowOSD": 1,  "OSDTopLeftX": 342,  "OSDTopLeftY": 0,  "TimeLocation": 0,  "OSDType": 0,  "OSDHourType": 0  }  }  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

### 多行OSD

#### 多行OSD(带点阵)

|  |  |
| --- | --- |
| 获取多行OSD(带点阵)配置 | |
| POST /digest/frmMultiLineOSD HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 10,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "MultiOSDLattice": {  "PosX": 988,  "PosY": 985,  "Location": 0,  "FontName": "SongTi",  "Enable": 0,  "Text": "123\n123\n123",  "MainStreamFontSize": 60,  "BitColor": "f8f8f8",  "SubStreamFontSize": 30,  "ThirdStreamFontSize": 20  }  }  } | "Enable": Int型,是否显示多行OSD,1-显示,0-不显示  "PosX": Int型,多行OSD左上角点的横坐标(0-1000).  "PosY": Int型,多行OSD左上角点的纵坐标(0-1000).  "Location": Int型,多行OSD位置  0-自定义,位置坐标为(PosX,PosY)  1-左上角  2-右上角  3-左下角  4-右下角.  "FontName": String型,多行OSD字体,取值范围:"SongTi","YaHei","HeiTi",  "YouYuan","Arial"  "Text": String型,多行OSD文本  "MainStreamFontSize": Int型,主码流OSD大小,取值范围:10,15,20,30,40,50,  60,70,80  "BitColor": String型,单行OSD颜色  "SubStreamFontSize": Int型,子码流OSD大小,取值范围:10,15,20,30,40  "ThirdStreamFontSize": Int型,第三码流OSD大小,取值范围:10,15,20,30,40,  50,60,70,80 |

|  |  |
| --- | --- |
| 设置单行OSD(带点阵)配置 | |
| POST /digest/frmMultiLineOSD HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 11,  "Dev": 1,  "Ch": 1,  "Data": {  "MultiOSDLattice": {  "Enable": 1,  "PosX": 0,  "PosY": 0,  "Location": 0,  "FontName": "YouYuan",  "Text": "IpCamera0",  "MainStreamFontSize": 60,  "BitColor": "f8f8f8",  "SubStreamFontSize": 30,  "ThirdStreamFontSize": 20,  "TextLength": 9,  "Main": {  "BytesArray": "xxx",  "BytesLength": 5238,  "Width": 432,  "Height": 97  },  "Sub": {  "BytesArray": "xxx",  "BytesLength": 1296,  "Width": 216,  "Height": 48  },  "Third": {  "BytesArray": "xxx",  "BytesLength": 594,  "Width": 144,  "Height": 33  }  }  }  } | "TextLength": Int型,多行OSD长度  "BytesArray": String型,点阵字节数据,用base64编码.  "BytesLength": Int型,点阵字节数组长度.  "Width": Int型,像素宽度.  "Height": Int型,像素高度. |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

#### 多行OSD(不带点阵)

|  |  |
| --- | --- |
| 获取多行OSD(不带点阵)配置 | |
| POST /digest/frmMultiLineOSD HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "MultiOSD": {  "TopLeftX": 347,  "TopLeftY": 283,  "Location": 0,  "FontName": "SongTi",  "IsShowMultiOSD": 0,  "Text": "123\n123\n123",  "TextSize": 12,  "BitColor": "f8f8f8",  "FirstStreamOsdSize": 60,  "SecondStreamOsdSize": 30,  "ThirdStreamOsdSize": 20  }  }  } | "IsShowMultiOSD": Int型,是否显示多行OSD,1-显示,0-不显示  "TopLeftX": Int型,多行OSD左上角点的横坐标(0-352).  "TopLeftY": Int型,多行OSD左上角点的纵坐标(0-288).  "Location": Int型,多行OSD位置  0-自定义,位置坐标为(TopLeftX,TopLeftY)  1-左上角  2-右上角  3-左下角  4-右下角.  "FontName": String型,多行OSD字体,取值范围:"SongTi","YaHei","HeiTi",  "YouYuan","Arial"  "Text": String型,多行OSD文本  "TextSize": Int型,多行OSD文本长度  "FirstStreamOsdSize": Int型,主码流OSD大小,取值范围:10,15,20,30,40,50,  60,70,80  "BitColor": String型,单行OSD颜色  "SecondStreamOsdSize": Int型,子码流OSD大小,取值范围:10,15,20,30,40  "ThirdStreamFontSize": Int型,第三码流OSD大小,取值范围:10,15,20,30,40,  50,60,70,80 |

|  |  |
| --- | --- |
| 设置多行OSD(不带点阵)配置 | |
| POST /digest/frmMultiLineOSD HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "MultiOSD": {  "TopLeftX": 347,  "TopLeftY": 283,  "Location": 0,  "FontName": "SongTi",  "IsShowMultiOSD": 0,  "Text": "123\n123\n123",  "TextSize": 12,  "BitColor": "f8f8f8",  "FirstStreamOsdSize": 60,  "SecondStreamOsdSize": 30,  "ThirdStreamOsdSize": 20  }  }  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

### 图像效果

|  |  |
| --- | --- |
| 获取图像效果配置 | |
| POST /digest/frmVideoEffect HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "VideoEffect": {  "Brightness": 128,  "Contrast": 128,  "Saturation": 128,  "Hue": 128  }  }  } | "Brightness": Int型,亮度,范围0-255  "Contrast": Int型,对比度,范围0-255  "Saturation": Int型,饱和度,范围0-255  "Hue": Int型,色度,范围0-255 |

|  |  |
| --- | --- |
| 设置图像效果配置 | |
| POST /digest/frmVideoEffect HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "VideoEffect": {  "Brightness": 128,  "Contrast": 128,  "Saturation": 128,  "Hue": 128  }  }  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

### 强制I帧

|  |  |
| --- | --- |
| 强制I帧 | |
| POST /digest/frmReqIFrame HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "StreamType":0  }  } | "StreamType":Int型,码流类型,0-主码流,1-子码流,3-第三码流 |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

### 视频参数

#### 视频参数能力集

|  |  |
| --- | --- |
| 获取TWDR支持信息 | |
| POST /digest/frmImageCapability HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "SupportTWDR": 1,  "SupportLights":["Ir\_Warm","Ir","Warm"]  }  } | "SupportTWDR":Int型,是否支持TWDR,1-支持,0-不支持.  "SupportLights":支持的灯板工作模式.字符串组成的数组,每个字符串代表一种支持的灯板工作模式.具体说明请参考下文的内容. |

**灯板光源可以有哪些种类?**

1,单光源(补光灯\*1)

补光灯,可以是:(1)红外灯,Ir.(2)暖光灯,Warm.(3)白光灯,White.

此时,补光灯光源是单一的,因此只有一种可选工作模式.

根据补光灯的类型,可选项为,"SupportLights":["Ir"],或者 "SupportLights":["Warm"], 或者"SupportLights":["White"].

2,警戒机光源(补光灯+报警灯)

补光灯,定义同上.

报警灯,用于发生报警事件时联动灯光闪烁的效果,因此报警灯不用于补光.

此时,补光灯光源是单一的,因此只有一种可选工作模式.

根据补光灯的类型,可选项为,"SupportLights":["Ir"],或者"SupportLights":["Warm"], 或者"SupportLights":["White"].

3,双光源(补光灯1+补光灯2)

补光灯,定义同上.

此时,两组补光灯为不同类光源,通常补光灯1为红外灯,补光灯2为暖光灯或白光灯.

补光灯光源是两种不同的,因此总共有三种可选工作模式,红外,暖光/白光,双光源.

根据补光灯的类型,可选项为,"SupportLights":["Ir\_Warm","Ir","Warm"],或者 "SupportLights":["Ir\_White","Ir","White"].

4,变倍组合光源(补光灯[近光]+补光灯[远光],补光灯[近光]+补光灯[中光]+补光灯[远光])

补光灯,定义同上.

此时,在不同倍率情况下,开启不同补光灯.在倍率较低时,开启近光灯,在倍率较高时,开启远光灯.

补光灯是两组同类光源,可以选择自动/近光/远光/全开. 当选择"自动",表示有程序根据倍率自动控制两组灯光.

可选项为,"SupportLights":["Auto","Close","LowBeam","HighBeam","AllBeam"].

#### 场景抓拍

|  |  |
| --- | --- |
| 获取场景抓拍配置 | |
| POST /digest/frmImageType HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "ImageType": 1,  "ImageTypeName": []  }  } | "ImageType":Int型,场景抓拍模式,0-缺省,1-大厅模式,2-车库模式,3-入口模式-速通,4-入口模式-慢通,5-出口模式,6-室外模式.  "ImageTypeName":保留字段 |

|  |  |
| --- | --- |
| 设置场景抓拍配置 | |
| POST /digest/frmImageType HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "ImageType": 1  }  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

#### 视频参数配置

|  |  |
| --- | --- |
| 获取视频参数配置 | |
| POST /digest/frmVideoParaEx HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } | "Type": 0-获取全天配置 2-获取白天配置 3-获取黑夜配置 |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "WBMode": -1,  "DarkCompensation": 0,  "ShowZoomRate": 0,  "ShowCoordinate": 0,  "ShowStatus": 0,  "FocusSpeed": 1,  "ZoomSpeed": 0,  "ElecAntiQuake": 0,  "CameraType": 0,  "EnableAutoConfig": -1,  "VideoType": 1,  "DayNightMode": {  "DayNightMode": 0,  "Delay": 0,  "NightToDayThreshold": 0,  "DayToNightThreshold": 0,  "DayStart": 0,  "DayEnd": 0  },  "ExtInTrig": 0,  "IcrOutTrig": 0,  "ShutterMode": 1,  "GainMode": 0,  "AeRouteMode": 0,  "WhiteBalance": {  "Mode": "Auto",  "Gain": {  "R": 66,  "G": 89,  "B": 71  }  },  "SharpnessLevel": 0,  "NRTfode": 0,  "WDMode": 1,  "WdrType": 0,  "DefogLevel": 1,  "Defog": 0,  "GammaMode": 1,  "PicQualityMode": 1,  "MinorMode": 1,  "FreqMode": 1,  "IcrLightMode": 0,  "IcrLightAue": 1,  "AutoLensMode": 1,  "IrisMode": 1,  "AutoSlowShutter": 1,  "SnMode": 1,  "DynamicBpCali": 1,  "LightCorrectMode": 1,  "LightCorrectLevel": 5,  "AECompensation": {  "Level": 50  },  "NR2D": {  "Mode": 50  },  "Dis": {  "Mode": 0  },  "LightType": "Ir",  "ExposureLight": {  "Mode": 0  },  "AfArea": 1,  "AfSearch": 1,  "AfSensitivity": 32,  "EnableDigitZoom": 0,  }  } | "WBMode": Int型,白平衡,-1-不支持,0-自动白平衡,1-室内模式,2-室外模式,3-ATW  "DarkCompensation": Int型,暗区补偿 0-关闭 1-低 2-中 3-高  "ShowZoomRate": Int型,显示变倍,0-隐藏,1-显示  "ShowCoordinate": Int型,显示坐标,0-隐藏,1-显示  "ShowStatus": Int型,显示状态,0-隐藏,1-显示  "FocusSpeed": Int型,变焦速度,0-低,1-中,2-高  "ZoomSpeed": Int型,变倍速度,0-低,1-中,2-高  "ElecAntiQuake": Int型,电子防抖,0-关闭 1-开启  "CameraType": Int型,MDI摄像机类型,仅获取  "EnableAutoConfig": Int型,是否开启摄像机自动配置,-1不支持  "VideoType"：Int型,设备类型,0-非球机,1-球机  "DayNightMode.DayNightMode": Int型,日夜切换模式,1-不支持,0-外部红外控制,1-自动模式,2-强制白天,3-强制黑夜  "DayNightMode.Delay": Int型,自动转换延迟,自动模式有效。0-30  "DayNightMode.NighttoDayThreshold": Int型,自动转换黑夜到白天的阈值,0-255,默认0xEE  "DayNightMode.DaytoNightThreshold": Int型,自动转换白天到黑夜的阈值,0-255,默认0x57  "DayNightMode.DayStart": Int型,自定义开始时间  "DayNightMode.DayEnd": Int型, 自定义开始时间  "ExtInTrig": Int型,日夜模式为外部触发时信号的输入电平为高或低时触发切换,0-默认,1-低电平,2-高电平  "IcrOutTrig": Int型,红外滤光片的触发方式,0-正向,1-反向  "ShutterMode": Int型,快门模式,-1-不支持,0-自动快门,手动增益,0x1~0x7f:快门优先,手动快门,自动增益  0x01:1/30(1/25),0x02:1/60(1/50),0x03:Flicker,0x04:1/250,0x05:1/500,0x06:1/1000,0x07:1/2000,0x08:1/5000,0x09:1/10000,0x0A:1/50000,0x0B:x2,0x0C:x4,0x0D:x6,0x0E:x8,0x0F:x10,0x10:x15,0x11:x20,0x12:x25,0x13:x30,0x100:全自动,快门自动,增益自动(最大),0x101~:全手动,快门手动,增益手动(GainMode为l),0x200:光圈优先,0x300~0x3ff:不解析快门时间,透传到机芯的快门值  "GainMode": Int型,增益,1-不支持,0-低;1-较低,2-中,3-较高,4-高  "AeRouteMode": Int型,曝光分配策略,0-快门优先,1-增益优先  "WhiteBalance"."Mode": String型,白平衡,取值:  Auto-自动,  Custom-自定义,  LockedWb-锁定白平衡,  IncandescentLamp-白炽灯,  WarmLight-暖光灯  NaturalLight-自然光  DaylightLamp-荧光灯"WhiteBalance"."Gain": Object型,白平衡自定义的值  "WhiteBalance"."Gain"."R": Int型,三色值(红),范围0-255  "WhiteBalance"."Gain"."G": Int型,三色值(绿),范围0-255  "WhiteBalance"."Gain"."B": Int型,三色值(蓝),范围0-255  "SharpnessLevel": Int型,锐利度,范围0-255  "NRTfode": Int型,3D降噪时域,-1-不支持,0-关闭,1-低,2-中,3-较高,4-高"WDMode": Int型,宽动态等级,1-低,2-中,3-高  "WdrType": Int型,宽动态模式,0-关闭,2-数字宽动态,3-真宽动态  "DefogLevel": Int型,去雾等级  "Defog": Int型,去雾功能,0-关闭,1-开启  "GammaMode": Int型,Gamma模式,-1-不支持,0-Curve\_1\_6,1-Curve\_1\_8  "PicQualityMode": Int型,图像效果,-1-不支持,0-正常,1-艳丽,2-自然  "MinorMode": Int型,镜像,-1-不支持,0-正常,1-水平翻转,2-垂直翻转,3-180度翻转,4-90度旋转,5-270度旋转"FreqMode": Int型,抗闪模式,-1-不支持,0-自动,1-50HZ,2-60HZ  "IcrLightMode": Int型,红外灯板状态,0-关闭,1-手动2-自动  "IcrLightAue": Int型,红外灯板亮度,范围0-100  "AutoLensMode": Int型,1-可以自动聚焦  "IrisMode": Int型,镜头光圈模式,-1不支持,0自动光圈,1手动或固定光圈"AutoSlowShutter": Int型,自动慢快门,0-关闭,N-N秒  "SnMode": Int型,提升信噪比,0-不启用,1-启用  "LightCorrectMode": Int型,背光模式,0-关闭,1-强光抑制,2-背光补偿"LightCorrectLevel": Int型,背光模式校正强度,范围1~10  "NR2D"."Mode": Int型,2D降噪,取值1-100  "Dis"."Mode": Int型,电子防抖,0-关闭,1-低,2-中,3-高  "LightType":String型,灯板工作模式,"Ir"-红外,"Warm"-暖光,"White"-白光  "ExposureLight"."Mode": Int型,红外收光,0-不收光,1-收光  "AfArea": Int型,对焦区域  "AfSearch": Int型,对焦区域搜索方式  "AfSensitivity": Int型,对焦灵敏度  "EnableDigitZoom": 0, Int型,数字ZOOM允许, 0-不启用,1-启用  "AECompensation. Level": Int型,目标亮度等级 |

|  |  |
| --- | --- |
| 设置视频参数配置 | |
| POST /digest/frmVideoParaEx HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "WBMode": -1,  "DarkCompensation": 0,  "ShowZoomRate": 0,  "ShowCoordinate": 0,  "ShowStatus": 0,  "FocusSpeed": 1,  "ZoomSpeed": 0,  "ElecAntiQuake": 0,  "CameraType": 0,  "EnableAutoConfig": -1,  "VideoType": 1,  "DayNightMode": {  "DayNightMode": 0,  "Delay": 0,  "NightToDayThreshold": 0,  "DayToNightThreshold": 0,  "DayStart": 0,  "DayEnd": 0  },  "ExtInTrig": 0,  "IcrOutTrig": 0,  "ShutterMode": 1,  "GainMode": 0,  "AeRouteMode": 0,  "WhiteBalance": {  "Mode": "Auto",  "Gain": {  "R": 66,  "G": 89,  "B": 71  }  },  "SharpnessLevel": 0,  "NRTfode": 0,  "WDMode": 1,  "WdrType": 0,  "DefogLevel": 1,  "Defog": 0,  "GammaMode": 1,  "PicQualityMode": 1,  "MinorMode": 1,  "FreqMode": 1,  "IcrLightMode": 0,  "IcrLightAue": 1,  "AutoLensMode": 1,  "IrisMode": 1,  "AutoSlowShutter": 1,  "SnMode": 1,  "DynamicBpCali": 1,  "LightCorrectMode": 1,  "LightCorrectLevel": 5,  "AECompensation": {  "Level": 50  },  "NR2D": {  "Mode": 50  },  "Dis": {  "Mode": 0  },  "LightType": 0,  "ExposureLight": {  "Mode": 0  },  "AfArea": 1,  "AfSearch": 1,  "AfSensitivity": 32,  "EnableDigitZoom": 0,  }  } | "Type": 1-设置全天配置 4-设置白天配置 5-设置黑夜配置 |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

#### 视频参数配置切换策略

|  |  |
| --- | --- |
| 获取视频参数配置切换策略 | |
| POST /digest/frmVideoParaExMode HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "SwitchProfile":[{  "Mode":0,  "DayStart":21600,  "DayEnd":64800  }]  }  } | "Mode":Int型,配置切换策略,0-通用,1-白天,2-黑夜,3-按时切换,4-按光敏切换  "DayStart":Int型,开始时间  "DayEnd":Int型,结束时间 |

|  |  |
| --- | --- |
| 设置视频参数配置切换策略 | |
| POST /digest/frmVideoParaExMode HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "SwitchProfile":[{  "Mode":0,  "DayStart":21600,  "DayEnd":64800  }]  }  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

### ROI配置

|  |  |
| --- | --- |
| 获取ROI配置 | |
| POST /digest/frmVideoRegionOfInterest HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "ROI": [  {  "Enable": 0,  "League": 1,  "X": 702,  "Y": 0,  "Width": 0,  "Height": 0  },  ……  ]  }  } | "ROI": Object数组,返回的长度是支持的ROI个数  "Enable": Int型,是否启用当前块,1-启用,0-不启用  "League": Int型,等级,范围1-6  "X": Int型,左上角点,横坐标,  "Y": Int型,左上角点,纵坐标,  "Width": Int型,区域宽度  "Height": Int型,区域高度 |

|  |  |
| --- | --- |
| 设置ROI配置 | |
| POST /digest/frmVideoRegionOfInterest HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "ROI": [  {  "Enable": 0,  "League": 1,  "X": 702,  "Y": 0,  "Width": 0,  "Height": 0  },  ……  ]  }  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

### 抓图

|  |  |
| --- | --- |
| 预览抓图 | |
| GET /digest/CaptureV2?Username=admin&PasswordDigest=xx&Created=xx&Nonce=xxx HTTP/1.1 | |
|  | "Username":String型,登录设备的用户名  "PasswordDigest":String型,密码字符串,加密规则base64(sha1(Nonce + Created + Password)  "Created":String型,客户端生成token的时间(DateTime ISO格式)  "Nonce":String型,生成规则base64(sha1(Created + secret key)) |
| 响应的正文内容 | jpeg格式图片 |

## 物理接口

### 报警输入

|  |  |
| --- | --- |
| 获取报警输入配置 | |
| POST /digest/frmAlarmInPara HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {  "AlarmCh": 1  }  } | "AalrmCh": Int星,报警输入通道号,从1开始 |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "AINum": 2,  "AIName": "1234",  "AIType": 1,  "AIHandle": 1,  "AlarmTime": [  [  [  0,  0,  23,  59  ],  ……  ],  ……  ],  "EnableHandle": 0,  "HandleType": 0,  "SnapCount": 1,  "SnapInterval": 1,  "EnableAudio": 0,  "AudioNo": 1,  "AlarmOut": 0,  "EnablePreset": 0,  "PresetNo": 0  }  } | "AINum": Int型,报警输入个数  "AIName": String型,报警输入名称  "AIType": Int型,报警器状态,0-常开,1-常闭  "AIHandle": Int型,是否启用报警输入,0-不启用,1-启用  "AlarmTime":布防时间结构,7天,每天8个时间段,"[0,0,23,59]"表示：开始时间0时0分,结束时间23时59分  "EnableHandle":Int型,联动总开关,0-不启用,1-启用  "HandleType": Int型,报警类型,按位操作:  "0x00"–不触发报警  "0x01"-监视器上报警  "0x02"-声音告警  "0x04"-上传中心  "0x08"-触发报警输出  "0x10"-邮件联动  "0x20"-触发报警录像  "0x40"-屏幕截图  "0x80"-联动Ftp  "0x100"-联动Http  "0x200"-联动灯光报警  "SnapCount": Int型,抓拍数量  "SnapInterval": Int型,抓拍间隔  "EnablePreset": Int型,是否调用预置点,1-是,0-否  "PresetNo": Int型,预置点号,范围0-255  "EnableAudio": Int型,是否联动音频 0-否,1-是  "AudioNo": Int型,联动的音频文件序号(1-8)  "AlarmOut": Int型,联动报警输出按位操作,bit0-报警输出1,bit1-报警输出2 |

|  |  |
| --- | --- |
| 设置报警输入配置 | |
| POST /digest/frmAlarmInPara HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "AlarmCh": 1,  "AINum": 2,  "AIName": "1234",  "AIType": 1,  "AIHandle": 1,  "AlarmTime": [  [  [  0,  0,  23,  59  ],  ……  ],  ……  ],  "EnableHandle": 0,  "HandleType": 0,  "SnapCount": 1,  "SnapInterval": 1,  "EnableAudio": 0,  "AudioNo": 1,  "AlarmOut": 0,  "EnablePreset": 0,  "PresetNo": 0  }  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

### 报警输出

|  |  |
| --- | --- |
| 获取报警输出配置 | |
| POST /digest/frmAlarmOut HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {  "AlarmCh": 1  }  } | "Type": Int型,0-获取参数,1-设置参数,2-获取报警输出的开关状态,3-开启报警输出,4-关闭报警输出,  "AalrmCh": Int型,报警输出通道号,从1开始 |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "AlarmOutCount": 1,  "AlarmType": 0,  "SelfDefinedDelay": 5,  "AlarmOutDelay": 0  }  } | "AlarmOutCount": Int型,报警输出个数  "AIName": String型,报警输入名称  "AlarmType": Int型,报警输出继电器状态,0-常开,1-常闭  "SelfDefinedDelay": Int型,自定义延时,单位:秒  "AlarmOutDelay": Int型,报警输出延时,   1. 不延时   5—5S  10—10S  30—30S  60—1min  120—2min  300—5min  600—10min; |

|  |  |
| --- | --- |
| 设置报警输出配置 | |
| POST /digest/frmAlarmOut HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "AlarmCh": 1,  "AlarmOutCount": 1,  "AlarmType": 0,  "SelfDefinedDelay": 5,  "AlarmOutDelay": 0  }  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

### 球机

#### 获取PTZ协议能力集

|  |  |
| --- | --- |
| 获取PTZ协议能力集 | |
| POST /digest/frmGetPTZProtocal HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "PTZProtocal": [  {  "Describe": "pelcoP",  "Type": 1  },  {  "Describe": "pelcoD",  "Type": 2  }  ]  }  } | "PTZProtocal": Object型数组,设备支持的PTZ协议列表  "Describe": String型,PTZ协议描述  "Type": Int型,PTZ协议编号 |

#### 编码参数

|  |  |
| --- | --- |
| 获取编码参数 | |
| POST /digest/frmDecoderPara HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "DecoderAddress": 1,  "DecoderType": 2,  "BaudRate": 9  }  } | "DecoderAddress": Int型,解码器地址,范围1-255  "DecoderType": Int型,解码器类型,参考PTZ协议能力集  "BaudRate": Int型,波特率(bps)  0-50  1-75  2-110  3-150  4-300  5-600  6-1200  7-2400  8-4800  9-9600  10-19200  11-38400  12-57600  14-115200 |

|  |  |
| --- | --- |
| 设置编码参数 | |
| POST /digest/frmDecoderPara HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "DecoderAddress": 1,  "DecoderType": 2,  "BaudRate": 9  }  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

#### 云台控制

|  |  |
| --- | --- |
| 云台控制 | |
| POST /digest/frmPTZControl HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "Cmd": 22,  "IsStop": 0,  "Speed": 5  }  } | "Cmd": Int型,控制命令字  11:缩放+  12:缩放-  13:聚焦+  14:聚焦-  15:光圈+  16:光圈-  21:上  22:下  23:左  24:右  25:左上  26:右上  27:左下  28:右下  29:自动  "IsStop": Int型,云台控制状态,0-开始,1-停止  "Speed": Int型,云台速度,范围1-10 |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

#### 预置点

|  |  |
| --- | --- |
| 预置点操作 | |
| POST /digest/frmPTZPreset HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "Cmd": 8,  "Index": 0  }  } | "Cmd": Int型,控制命令字  8:设置预置点  39:调用预置点  "Index": Int型,预置点编号,从0开始 |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

#### 巡航线配置

|  |  |
| --- | --- |
| 获取巡航线配置 | |
| POST /digest/frmPTZCruise HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {  "CruisePath": 1  }  } | "CruisePath": Int型,巡航线编号,从1开始 |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "Cruise": [  {  "PresetNo": 0,  "Dwell": 1,  "Speed": 5  },  {  "PresetNo": 2,  "Dwell": 1,  "Speed": 5  }  ]  }  } | "Cruise": Object型数组,巡航线的预置点列表  "Cruise"[]."PresetNo": Int型,预置点号,从0开始  "Cruise"[]."Dwell": Int型,停留时间,单位:秒  "Cruise"[]."Speed": Int型,速度,范围1-10 |

|  |  |
| --- | --- |
| 巡航线添加预置点 | |
| POST /digest/frmPTZCruise HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "CruisePath": 1,  "PresetNo": 0,  "Dwell": 1,  "Speed": 5  }  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

|  |  |
| --- | --- |
| 巡航线删除预置点 | |
| POST /digest/frmPTZCruise HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 2,  "Dev": 1,  "Ch": 1,  "Data": {  "CruisePath": 1,  "CruiseItemNo": 3  }  } | "CruiseItemNo": Int型,巡航子项序号,只支持末位删除,暂时只能是末位项序号,从1开始 |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

#### 巡航线控制

|  |  |
| --- | --- |
| 巡航线控制 | |
| POST /digest/frmPTZCruise HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 3,  "Dev": 1,  "Ch": 1,  "Data": {  "CruisePath": 1  }  } | "Type": Int型,操作类型,3-删除巡航线,4-调用巡航线  "CruisePath": Int型,巡航线编号,从1开始 |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

#### 轨迹控制

|  |  |
| --- | --- |
| 记录轨迹 | |
| POST /digest/frmPTZTrack HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {  "TrackIndex": 1,  "Run": 1  }  } | "TrackIndex": Int型,轨迹编号,从1开始  "Run":Int型,是否记录,1-开始记录,0-停止记录 |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

|  |  |
| --- | --- |
| 调用轨迹 | |
| POST /digest/frmPTZTrack HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "TrackIndex": 1  }  } | "TrackIndex": Int型,轨迹编号,从1开始 |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

#### 两点扫描

|  |  |
| --- | --- |
| 两点扫描控制 | |
| POST /digest/frmPTZExtend\_SetScan | |
| 请求的正文内容：  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {  "Index": 1  }  } | "Type": Int型,操作类型,0-设置开始,1-设置停止,2-调用  "Index": Int型,两点扫描编号,从1开始 |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

#### 空闲操作

|  |  |
| --- | --- |
| 获取空闲操作配置 | |
| POST /digest/frmPTZExtend\_IdleOperation HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "Type": 0,  "Index": 0,  "IdleTime": 60  }  } | "Type": Int型,操作类型,0-不动作,1-预置点,2-巡航,3-轨迹,4-两点扫描  "Index": Int型,设置为第几组(第几组预置点或巡航等),从0开始计数  "IdleTime": Int型,空闲时间,单位:秒 |

|  |  |
| --- | --- |
| 设置空闲操作配置 | |
| POST /digest/frmPTZExtend\_IdleOperation HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {  "Type": 0,  "Index": 0,  "IdleTime": 60  }  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

|  |  |
| --- | --- |
| 调用空闲操作 | |
| POST /digest/frmPTZExtend\_IdleOperation HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 2,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

#### 红外补光

|  |  |
| --- | --- |
| 获取红外补光配置 | |
| POST /digest/frmPTZExtend\_IrlightCtrl HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "Mode": 1,  "Lighting": 0  }  } | "Mode": Int型,红外补光模式,0-自动,1-手动  "Lighting": Int型,按位表示每个灯,  bit0-近灯  bit1-中灯  bit2-远灯 |

|  |  |
| --- | --- |
| 设置红外补光配置 | |
| POST /digest/frmPTZExtend\_IrlightCtrl HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "Mode": 1,  "Lighting": 0  }  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

#### 隐私遮蔽

|  |  |
| --- | --- |
| 获取隐私遮蔽配置 | |
| POST /digest/frmPTZExtend\_SetCover HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "CoverEnable": [  0,  0,  0,  0,  0,  0,  0,  0  ]  }  } | "CoverEnable": Int型数组,所有隐私遮蔽区域的状态列表,0-关闭,1-开启 |

|  |  |
| --- | --- |
| 设置隐私遮蔽配置 | |
| POST /digest/frmPTZExtend\_SetCover HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "Enable": 0,  "Index": 1  }  } | "Enable": Int型,当前隐私遮蔽区域状态,0-关闭,1-开启  "Index": Int型,当前隐私遮蔽区域编号,从1开始 |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

|  |  |
| --- | --- |
| 隐私遮蔽区域控制 | |
| POST /digest/frmPTZExtend\_SetCover HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 2,  "Dev": 1,  "Ch": 1,  "Data": {  "Index": 2  }  } | "Type": Int型,操作类型,2-开始设置当前隐私遮蔽区域,3-停止设置当前隐私遮蔽区域  "Index": Int型,当前隐私遮蔽区域编号,从1开始 |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

#### 3D定位

|  |  |
| --- | --- |
| 3D定位控制 | |
| POST /digest/frmPTZExtend\_3DPosition HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "StartX": 23216,  "StartY": 15184,  "StopX": 35510,  "StopY": 25308  }  } | "StartX": Int型,起始点横坐标.范围0~65535.  "StartY": Int型,起始点纵坐标.范围0~65535.  "StopX": Int型,终止点横坐标.范围0~65535.  "StopY": Int型,终止点纵坐标.范围0~65535.  放大:StartY应该小于StopY  缩小:StartY应该大于StopY  转向:StartX应该等于StopX,StartY应该等于StopY. |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

#### 雨刷

|  |  |
| --- | --- |
| 雨刷控制 | |
| POST /digest/frmPTZExtend\_Wiper HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "Enable": 0  }  } | "Enable": Int型,雨刷状态,1-开启,0-关闭 |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

#### 喷淋位置

|  |  |
| --- | --- |
| 喷淋位置控制 | |
| POST /digest/frmPTZExtend\_SprayPos HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "Code": 0  }  } | "Code": Int型,雨刷状态,0-设置位置,1-删除位置 |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

#### 喷淋模式

|  |  |
| --- | --- |
| 喷淋模式设置 | |
| POST /digest/frmPTZExtend\_SprayPos HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "Code": 0  }  } | "Code": Int型,喷淋模式,0-自动,1-手动 |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

### 网络球机

#### 获取网络球机能力集

|  |  |
| --- | --- |
| 获取网络球机能力集 | |
| POST /digest/frmPTZDomeGetAbility HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "MaxPersetNum": 256,  "MaxPatrolPathNum": 8,  "MaxPersetNumPrePatrolPath": 8,  "MaxPatternNum": 4,  "MaxScheduleNodePreDay": 10,  "ParkAndScheduleTimeMin": 5,  "ParkAndScheduleTimeMax": 720  }  } | "MaxPersetNum": Int型,预置点个数  "MaxPatrolPathNum": Int型,巡航线个数  "MaxPersetNumPrePatrolPath": Int型,每个巡航线包含的预置点个数  "MaxPatternNum": Int型,轨迹个数  "MaxScheduleNodePreDay": Int型,计划任务,每天最大任务数  "ParkAndScheduleTimeMin": Int型,看守位和计划任务恢复时间选择范围最小值  "ParkAndScheduleTimeMax": Int型,看守位和计划任务恢复时间选择范围最大值 |

#### 获取协议列表

|  |  |
| --- | --- |
| 获取网络球机协议能力集 | |
| POST /digest/frmPTZDomeGetProtocol HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "ProtocolList": [  {  "Name": "pelcoD",  "Idx": 1  }  ]  }  } | "PTZProtocal": Object型数组,设备支持的PTZ协议列表  "Name": String型,PTZ协议描述  "Idx": Int型,PTZ协议编号 |

#### UART配置

|  |  |
| --- | --- |
| 获取UART配置 | |
| POST /digest/frmPTZDomeUart HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "ProtocolIdx": 1,  "Address": 255,  "BaudRate": 115200,  "DataLen": 8,  "StopLen": 1,  "Parity": 0,  "FlowCtl": 0  }  } | "ProtocolIdx": Int型,解码器类型,参考PTZ协议能力集  "Address": Int型,解码器地址,范围1-255  "BaudRate": Int型,波特率(bps)  50-50  75-75  110-110  150-150  300-300  600-600  1200-1200  2400-2400  4800-4800  9600-9600  19200-19200  38400-38400  57600-57600  115200-115200  "DataLen": Int型,数据长度  "StopLen": Int型,停止位  "Parity": Int型,奇偶性  "FlowCtl": Int型,流量控制 |

|  |  |
| --- | --- |
| 设置UART配置 | |
| POST /digest/frmPTZDomeUart HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "ProtocolIdx": 1,  "Address": 255,  "BaudRate": 115200,  "DataLen": 8,  "StopLen": 1,  "Parity": 0,  "FlowCtl": 0  }  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

#### 云台控制

|  |  |
| --- | --- |
| 云台控制 | |
| POST /digest/frmPTZDomePtzControl HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "Cmd": 0,  "Speed": 5  }  } | "Cmd": Int型,控制命令字  0:自动  1:上  2:下  3:左  4:右  5:左上  6:右上  7:左下  8:右下  "Speed": Int型,云台速度,0-停止,速度1-10 |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

#### 预置点控制

|  |  |
| --- | --- |
| 预置点操作 | |
| POST /digest/frmPTZDomePresetControl HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "Cmd": 1,  "Idx": 0  }  } | "Cmd": Int型,控制命令字  1:设置预置点  2:调用预置点  3:清除预置点  "Idx": Int型,预置点编号,从0开始 |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

#### 巡航线配置

|  |  |
| --- | --- |
| 获取巡航线配置 | |
| POST /digest/frmPTZDomePatrolPath HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {  "PatrolPathIdx": 0  }  } | "PatrolPathIdx": Int型,巡航线号,0:所有巡航线配置,>0:指定巡航线配置 |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "PatrolPath1": {  "Name": "",  "Presets": [  {  "PresetIdx": 0,  "Delay": 1,  "Speed": 5  },  {  "PresetIdx": 1,  "Delay": 1,  "Speed": 5  }  ]  },  ……  }  } | "PatrolPath[1-8]": Object型,巡航线配置  "Name": String型,巡航线名  "Presets": Object型数组,巡航线的预置点列表  "Presets"[]."PresetIdx": Int型,预置点号,从0开始  "Presets"[]."Delay": Int型,停留时间,单位:秒  "Presets"[]."Speed": Int型,速度,范围1-10 |

|  |  |
| --- | --- |
| 设置巡航线配置.添加/删除巡航线的预置点,就是增/删Presets数组里值的个数,删除巡航线,就是清空Presets数组. | |
| POST /digest/frmPTZDomePatrolPath HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "PatrolPath1": {  "Name": "",  "Presets": [  {  "PresetIdx": 0,  "Delay": 1,  "Speed": 5  }  ]  }  }  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

#### 巡航线控制

|  |  |
| --- | --- |
| 巡航线控制 | |
| POST /digest/frmPTZDomePatrolPathControl HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "Cmd": 1,  "Idx": 1  }  } | "Cmd": Int型,控制命令字  1:停止巡航线  2:调用巡航线  "Idx": Int型,巡航线号,从1开始 |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

#### 轨迹控制

|  |  |
| --- | --- |
| 轨迹控制 | |
| POST /digest/frmPTZDomePatternControl HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "Cmd": 0,  "Idx": 1  }  } | "Cmd": Int型,控制命令字  0:开始记录  1:停止记录  2:调用  3:停止调用  4:清除轨迹  "Idx": Int型,轨迹编号,从1开始 |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

#### 看守配置

|  |  |
| --- | --- |
| 获取看守配置 | |
| POST /digest/frmPTZDomePark HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "Enable": 0,  "ParkTimeOut": 5,  "ParkType": 1,  "ParkIdx": 1,  "ParkSpeed": 1  }  } | "Enable": Int型,是否启用看守功能,1-启用,0-不启用  "ParkTimeOut": Int型,等待时间,单位:秒  "ParkType": Int型,看守模式,0-预置点,1-巡航,2-轨迹,3-水平扫描  "ParkIdx": Int型,看守编号  "ParkSpeed": Int型,速度 |

|  |  |
| --- | --- |
| 设置看守配置 | |
| POST /digest/frmPTZDomePark HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "Enable": 0,  "ParkTimeOut": 5,  "ParkType": 1,  "ParkIdx": 1,  "ParkSpeed": 1  }  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

#### 看守控制

|  |  |
| --- | --- |
| 立即执行一次看守 | |
| POST /digest/frmPTZDomeParkCallOnce HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

#### 断电记忆

|  |  |
| --- | --- |
| 获取断电记忆配置 | |
| POST /digest/frmPTZDomePowerOffMemoryHTTP/1.1 | |
| 请求的正文内容：  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "Enable": 1,  "Times": 10  }  } | "Enable": Int型,是否启用断电记忆,1-启用,0-不启用  "Times": Int型,超时时间,单位:秒 |

|  |  |
| --- | --- |
| 设置断电记忆配置 | |
| POST /digest/frmPTZDomePowerOffMemory HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "Enable": 1,  "Times": 10  }  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

#### 计划任务

|  |  |
| --- | --- |
| 获取计划任务配置 | |
| POST /digest/frmPTZDomeScheduleTaskHTTP/1.1 | |
| 请求的正文内容：  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "Enable": 0,  "Timeout": 5,  "Schedule": [  [  {  "Enable": 0,  "Type": 0,  "No": 0,  "Speed": 0,  "Time": [  0,  0,  0,  0  ]  },  ……  ],  ……  ]  }  } | "Enable": Int型,是否启用计划任务,1-启用,0-不启用  "Timeout": Int型,超时时间,单位:秒  "Schedule": 7x10数组,7天,每天十个计划时间段  "Enable": Int型,当前时间段是否启用,1-启用,0-不启用  "Type": Int型,任务类型,0-关闭,1-自动,2-预置点,3-巡航,4-轨迹  "No": Int型,计划任务编号  "Speed": Int型,速度(Type为自动时有效)  "Time": Int型数组,计划时间段  "Time"[0]: Int型,计划开始时间(时)  "Time"[1]: Int型, 计划开始时间(分)  "Time"[2]: Int型, 计划结束时间(时)  "Time"[3]: Int型, 计划结束时间(分) |

|  |  |
| --- | --- |
| 设置计划任务配置 | |
| POST /digest/frmPTZDomeScheduleTask HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "Enable": 0,  "Timeout": 5,  "Schedule": [  [  {  "Enable": 0,  "Type": 0,  "No": 0,  "Speed": 0,  "Time": [  0,  0,  0,  0  ]  },  ……  ],  ……  ]  }  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

#### OSD位置

|  |  |
| --- | --- |
| 获取OSD位置配置 | |
| POST /digest/frmPTZDomeOSDHTTP/1.1 | |
| 请求的正文内容：  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "ShowPtzPosition": 5,  "ShowPresetName": 0  }  } | "ShowPtzPosition": Int型,是否显示PTZ位置,1-显示,0-不显示  "ShowPresetName": Int型,是否显示预置点名,1-显示,0-不显示 |

|  |  |
| --- | --- |
| 设置OSD位置配置 | |
| POST /digest/frmPTZDomeOSD HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "ShowPtzPosition": 5,  "ShowPresetName": 0  }  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

#### 智能跟踪

|  |  |
| --- | --- |
| 获取智能跟踪配置 | |
| POST /digest/frmSmartDomeTrack | |
| 请求的正文内容：  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "Enable": 1,  "DetectType": 0,  "ZoomRatio": 5,  "EnableAutoTrack": 1,  "AutoTrackMode": 0,  "MinArea": 1,  "MinTotalArea": 10000,  "MoveSensitivity": 5,  "ColorMatchSensitivity": 6,  "ObjectType": 0  }  } | "Enable": Int型,是否启用智能跟踪,1-启用,0-不启用  "DetectType": 暂未使用，预留.  "ZoomRatio": Int型,目标变倍,范围1~10,  "EnableAutoTrack": Int型,是否启用自动跟踪模式,1-启用,0-不启用,  "AutoTrackMode": Int型,自动跟踪模式,0-默认,1-运动模式,2-体积模式,  "MinArea": Int型,目标占比,范围0~10000,  "MinTotalArea": Int型, 总占比,值为10000,  "MoveSensitivity": Int型,运动灵敏度,范围1~10,值越大,越容易误跟踪,  "ColorMatchSensitivity": Int型,颜色匹配灵敏度,范围1~10, 值越大,越容易误跟踪,  "ObjectType": Int型,目标类型,高3字节为子类型,低1字节为主类型,主类型默认为1,主类型定义如下:  0:人  1:车  2:全部 |

|  |  |
| --- | --- |
| 设置智能跟踪配置 | |
| POST /digest/frmSmartDomeTrack | |
| 请求的正文内容：  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "AutoTrackMode": 0,  "ColorMatchSensitivity": 6,  "DetectType": 0,  "Enable": 1,  "EnableAutoTrack": 1,  "MinArea": 1,  "MinTotalArea": 10000,  "MoveSensitivity": 5,  "ObjectType": 0  "ZoomRatio": 5,  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

#### 手动追踪

|  |  |
| --- | --- |
| 手动追踪控制 | |
| POST /digest/frmPTZDomeManualTrack | |
| 请求的正文内容：  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "StartX": 23216,  "StartY": 15184,  "StopX": 35510,  "StopY": 25308  }  } | "StartX": Int型,起始点横坐标.范围0~65535.  "StartY": Int型,起始点纵坐标.范围0~65535.  "StopX": Int型,终止点横坐标.范围0~65535.  "StopY": Int型,终止点纵坐标.范围0~65535.  放大:StartY应该小于StopY  缩小:StartY应该大于StopY  转向:StartX应该等于StopX,StartY应该等于StopY. |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

#### 球机定位

|  |  |
| --- | --- |
| 获取球机定位配置 | |
| POST /digest/frmPTZDomeLocation | |
| 请求的正文内容：  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "P": 34245,  "T": 1541,  "Z": 2117  }  }} | "P": Int型, 水平位置,范围0 ~ 36000 (即0 ~ 360°).  "T": Int型,垂直位置, 范围-9000 ~ 9000(即-90°~90°),  "Z": Int型,视角, 每个镜头不太一样, 一般范围是0~10000 . |

|  |  |
| --- | --- |
| 设置球机定位配置 | |
| POST /digest/frmPTZDomeLocation | |
| 请求的正文内容：  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "P": 34245,  "T": 1541,  "Z": 2117  }  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

### 鱼眼相机

#### 获取鱼眼相机能力集

|  |  |
| --- | --- |
| 获取鱼眼相机能力集 | |
| POST /digest/frmFisheyeAbility HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "ViewMode": [  {  "FisheyeView": 1,  "FullView": 0,  "PtzView": 0  },  ……  ],  "PtzNum": {  "PresetNum": 256,  "CruiseNum": 32,  "CruisePresetNum": 16,  "SpeedStart": 1,  "SpeedEnd": 10,  "DelayStart": 1,  "DelayEnd": 120  },  "OptionalDevChanStreams": [  [  2,  2,  2,  2,  2  ]  ]  }  } | "ViewMode": Object型数组,设备支持的鱼眼模式  "ViewMode"[]."FisheyeView": Int型,语言通道个数  ViewMode"[]."FullView": Int型,全景通道个数  ViewMode"[]."PtzView": Int型,PTZ通道个数  "PtzNum": Object型,通道能力集  "PtzNum"."PresetNum": Int型,预置点个数  "PtzNum"."CruiseNum": Int型,巡航线个数  "PtzNum"."CruisePresetNum": Int型,巡航线的预置点个数  "PtzNum"."SpeedStart": Int型,最小速度  "PtzNum"."SpeedEnd": Int型,最大速度  "PtzNum"."DelayStart": Int型,最小停留时间  "PtzNum"."DelayEnd": Int型,最大停留时间  "OptionalDevChanStreams": Int型二维数组,各设备的各通道的实时流数目,第一维表示设备sensor个数,第二维表示各sensor支持的通道的实时流个数 |

#### 鱼眼模式配置

|  |  |
| --- | --- |
| 获取鱼眼模式配置 | |
| POST /digest/frmFisheyModeHTTP/1.1 | |
| 请求的正文内容：  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "InstallMode": 0,  "ViewMode": {  "FisheyeView": 1,  "FullView": 1,  "PtzView": 3  },  "FullViewMode": 0  }  } | "InstallMode": Int型,安装模式,0-顶装,1-壁装,2-地装  "ViewMode": Object型,鱼眼相机当前模式  "FisheyeView": Int型,鱼眼通道个数  "FullView": Int型,全景通道个数  "PtzView": Int型,全景通道个数  "FullViewMode": Int型,全景模式,0-180度,1-360度 |

|  |  |
| --- | --- |
| 设置鱼眼模式配置,切换鱼眼模式需要重启生效 | |
| POST /digest/frmFisheyMode HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "InstallMode": 0,  "ViewMode": {  "FisheyeView": 1,  "FullView": 1,  "PtzView": 3  },  "FullViewMode": 0  }  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

#### 获取鱼眼通道位置

|  |  |
| --- | --- |
| 获取鱼眼通道位置 | |
| POST /digest/frmFisheyeChanPosHTTP/1.1 | |
| 请求的正文内容：  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {  "ChanNum": 2  }  } | "ChanNum": Int型,鱼眼相机的通道号 |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "PanAngle": 54,  "TiltAngle": 90,  "ZoomRate": 9999  }  } | "PanAngle": Int型,水平角度:-180~+180(顶装和地装方式),-90~+90(壁装)  "tiltAngle": Int型,垂直角度:0-90(顶装和地装方式),-90~+90(壁装)  "ZoomRate": Int型 ,倍率:1~9999 |

#### 云台控制

|  |  |
| --- | --- |
| 云台控制 | |
| POST /digest/frmFisheyeChanMove HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "ChanNum": 2,  "Operation": "ContinousMove",  "PanAngle": -1,  "PanSpeed": 0.5,  "tiltAngle": 1,  "TileSpeed": 0.5,  "ZoomRate": 1,  "ZoomSpeed": 0.5  }  } | "ChanNum": Int型,鱼眼相机的云台通道号  "Operation": String型,云台操作,  "RelativeMove":相对移动  "AbsoluteMove":绝对移动  "ContinousMove":连续移动  "PanAngle": Int型,参数可选,水平角度:-180~+180(顶装和地装方式),-90~+90(壁装),连续移动时,-1表示左,0表示停止,1表示右.  "PanSpeed": Float型,参数可选,水平移动速度：0~1.0  "tiltAngle": Int型,参数可选,垂直角度:0-90(顶装和地装方式),-90~+90(壁装),连续移动时,-1表示下,0表示停止,1表示上.  "TileSpeed": Float型,参数可选,垂直移动速度：0~1.00~1.0  "ZoomRate": Int型,参数可选,倍率:1~9999,连续移动时,-1表示缩小,0表示停止,1表示放大.  "ZoomSpeed": Float型,参数可选,倍率速度:0~1.0 |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

#### 预置点

|  |  |
| --- | --- |
| 获取预置点位置 | |
| POST /digest/frmFisheyeChanPresetHTTP/1.1 | |
| 请求的正文内容：  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {  "ChanNum": 2,  "Preset": 0  }  } | "ChanNum": Int型,鱼眼相机的云台通道号  "Preset": Int型,预置点序号 |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "ChanNum": 2,  "Preset": 0,  "PanAngle": 54,  "TiltAngle": 90,  "ZoomRate": 9999  }  } | "PanAngle": Int型,水平角度:-180~+180(顶装和地装方式),-90~+90(壁装)  "tiltAngle": Int型,垂直角度:0-90(顶装和地装方式),-90~+90(壁装)  "ZoomRate": Int型 ,倍率:1~9999 |

|  |  |
| --- | --- |
| 设置/调用/删除预置点 | |
| POST /digest/frmFisheyeChanPreset HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "ChanNum": 2,  "Preset": 0  }  } | "Type": Int型,操作类型,1表示设置,2表示调用,3表示删除 |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

#### 巡航线配置

|  |  |
| --- | --- |
| 获取巡航线配置 | |
| POST /digest/frmFisheyeChanCruise HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {  "Route": 0,  "ChanNum": 2  }  } | "Type": Int型,请求类型,0-获取,1-设置,2-调用,3-删除  "Route": Int型,巡航线号,从0开始  "ChanNum": Int型,鱼眼相机的云台通道号 |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "ChanNum": 2,  "Route": 0,  "Name": "",  "RouteList": [  {  "Point": 0,  "Delay": 1,  "Speed": 5  }  ]  }  } | "Name": String型,巡航线名  "RouteList": Object型数组,巡航线的预置点列表  "RouteList" []."Point": Int型,预置点号,从0开始  "RouteList" []."Delay": Int型,停留时间,单位:秒  "RouteList" []."Speed": Int型,速度,范围1-10 |

|  |  |
| --- | --- |
| 巡航线添加预置点 | |
| POST /digest/frmFisheyeChanCruise HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "ChanNum": 2,  "OperationType": 0,  "Route": 2,  "Point": 0,  "Delay": 1,  "Speed": 5  }  } | "ChanNum": Int型,鱼眼相机的云台通道号  "OperationType": Int型,0-添加预置点,1-修改预置点属性,2-批量添加预置点,3-修改巡航线  "Route": Int型,巡航线号,从0开始  "Point": Int型,预置点号,从0开始  "Delay": Int型,停留时间,单位:秒  "Speed": Int型,速度,范围1-10 |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "ChanNum": 2,  "Route": 2,  "StatusCode": "Operation Ok"  }  } |  |

|  |  |
| --- | --- |
| 巡航线删除预置点 | |
| POST /digest/frmFisheyeChanCruise HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 3,  "Dev": 1,  "Ch": 1,  "Data": {  "ChanNum": 2,  "OperationType": 1,  "Route": 2,  "Index": 0  }  } | "Type": Int型,请求类型,3-删除  "ChanNum": Int型,鱼眼相机的云台通道号  "OperationType": Int型,操作类型,0-删除巡航线,1-删除预置点  "Route": Int型,巡航线号,从0开始  "Index": Int型,"RouteList"下标,删除第几个预置点,从0开始 |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "ChanNum": 2,  "Route": 2,  "StatusCode": "Operation Ok"  }  } |  |

#### 巡航线控制

|  |  |
| --- | --- |
| 删除巡航线 | |
| POST /digest/frmFisheyeChanCruise HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 3,  "Dev": 1,  "Ch": 1,  "Data": {  "ChanNum": 2,  "OperationType": 1,  "Route": 2  }  } | "Type": Int型,请求类型,3-删除  "ChanNum": Int型,鱼眼相机的云台通道号  "OperationType": Int型,操作类型,0-删除巡航线,1-删除预置点  "Route": Int型,巡航线号,从0开始  "Point": Int型,预置点号,从0开始  "Delay": Int型,停留时间,单位:秒  "Speed": Int型,速度,范围1-10 |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "ChanNum": 2,  "Route": 2,  "StatusCode": "Operation Ok"  }  } |  |

|  |  |
| --- | --- |
| 调用巡航线 | |
| POST /digest/frmFisheyeChanCruise HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 2,  "Dev": 1,  "Ch": 1,  "Data": {  "ChanNum": 2,  "OperationType": 1,  "Route": 2  }  } | "Type": Int型,请求类型,2-调用  "ChanNum": Int型,鱼眼相机的云台通道号  "OperationType": Int型,0-停止,1-开始  "Route": Int型,巡航线号,从0开始  "Point": Int型,预置点号,从0开始  "Delay": Int型,停留时间,单位:秒  "Speed": Int型,速度,范围1-10 |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "ChanNum": 2,  "Route": 2,  "StatusCode": "Operation Ok"  }  } |  |

## 报警事件

### 报警事件联动

|  |  |
| --- | --- |
| 获取报警事件联动配置 | |
| POST /digest/frmPTZLinkCFG HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {  "PTZLinkType": 1  }  } | "PTZLinkType": Int型,报警联动类型,1-移动侦测报警联动,2-遮挡报警联动 |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "EnablePreset": 0,  "PresetNo": 0  }  } | "EnablePreset": Int型,是否调用预置点,1-是,0-否  "PresetNo": Int型,预置点号,范围0-255 |

|  |  |
| --- | --- |
| 设置报警事件联动配置 | |
| POST /digest/frmPTZLinkCFG HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "PTZLinkType": 1,  "EnablePreset": 0,  "PresetNo": 0  }  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

### 移动侦测

|  |  |
| --- | --- |
| 获取移动侦测配置 | |
| POST digest/frmMotionDetPara HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "Enable": 0,  "Sensitive": 5,  "MotionScope": [  [  0,  ……  ],  ……  ],  "AlarmTime": [  [  [  0,  0,  23,  59  ],  ……  ]  ……  ],  "EnableHandle": 0,  "HandleType": 0,  "SnapCount": 1,  "SnapInterval": 1,  "EnableAudio": 0,  "AudioNo": 1,  "AlarmOut": 0  }  } | "Enable": Int型,是否启用移动侦测:0-不启用,1-启用  "Sensitive": Int型,灵敏度：范围为0-5, 灵敏度越高越灵敏  "MotionScope":Int型二位数组,移动侦测范围,18x22数组,18行,每行22块,总共22\*18个块区,"1"-侦测范围, "0"-不在侦测范围内  "AlarmTime":布防时间结构,7天,1天8个时间段,"[0,0,23,59]"表示：开始时间0时0分,结束时间23时59分  "EnableHandle":Int型,联动总开关,0-不启用,1-启用  "HandleType": Int型,报警类型,按位操作:  "0x00"–不触发报警  "0x01"-监视器上报警  "0x02"-声音告警  "0x04"-上传中心  "0x08"-触发报警输出  "0x10"-邮件联动  "0x20"-触发报警录像  "0x40"-屏幕截图  "0x80"-联动Ftp  "0x100"-联动Http  "0x200"-联动灯光报警  "SnapCount": Int型,抓拍数量  "SnapInterval": Int型,抓拍间隔  "EnableAudio": Int型,是否联动音频 0-否,1-是  "AudioNo": Int型,联动的音频文件序号(1-8)  "AlarmOut": Int型,联动报警输出按位操作,bit0-报警输出1,bit1-报警输出2 |

|  |  |
| --- | --- |
| 设置移动侦测配置 | |
| POST digest/frmMotionDetPara HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "Enable": 0,  "Sensitive": 5,  "MotionScope": [  [  0,  ……  ],  ……  ],  "AlarmTime": [  [  [  0,  0,  23,  59  ],  ……  ]  ……  ],  "EnableHandle": 0,  "HandleType": 0,  "SnapCount": 1,  "SnapInterval": 1,  "EnableAudio": 0,  "AudioNo": 1,  "AlarmOut": 0  }  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "StatusCode":"Operation Ok"  }  } |  |

### 遮挡报警

|  |  |
| --- | --- |
| 获取遮挡报警配置 | |
| POST digest/frmVideoHidePara HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {  "AlarmTimeType": 1,  "NewSemantics": 1  }  } | "AlarmTimeType":Int型,时间段类型,0-每天一个时间段,1-每天八个时间段  "NewSemantics":Int型,是否是新结构,强制为1 |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "Enable": 0,  "Sensitivity": 2,  "Shelters": [  702,  0,  0,  0  ],  "AlarmTime": [  [  [  0,  0,  23,  59  ],  ……  ]  ……  ],  "EnableHandle": 0,  "HandleType": 0,  "SnapCount": 1,  "SnapInterval": 1,  "EnableAudio": 0,  "AudioNo": 1,  "AlarmOut": 0  }  } | "Enable": Int型,是否启用遮挡报警:0-不启用,1-启用  "Sensitivity": Int型,灵敏度：范围为0-2, 灵敏度越高越灵敏  "Shelters": Int型数组,长度为4,遮挡区域  "Shelters"[0]: Int型,遮挡区域的x坐标,范围0-704  "Shelters"[1]: Int型,遮挡区域的y坐标,范围0-576  "Shelters"[2]: Int型,遮挡区域的宽度,范围0-704  "Shelters"[3]: Int型,遮挡区域的高度,范围0-576  "AlarmTime":布防时间结构,7天,1天8个时间段,"[0,0,23,59]"表示：开始时间0时0分,结束时间23时59分  "EnableHandle":Int型,联动总开关,0-不启用,1-启用  "HandleType": Int型,报警类型,按位操作:  "0x00"–不触发报警  "0x01"-监视器上报警  "0x02"-声音告警  "0x04"-上传中心  "0x08"-触发报警输出  "0x10"-邮件联动  "0x20"-触发报警录像  "0x40"-屏幕截图  "0x80"-联动Ftp  "0x100"-联动Http  "0x200"-联动灯光报警  "SnapCount": Int型,抓拍数量  "SnapInterval": Int型,抓拍间隔  "EnableAudio": Int型,是否联动音频 0-否,1-是  "AudioNo": Int型,联动的音频文件序号(1-8)  "AlarmOut": Int型,联动报警输出按位操作,bit0-报警输出1,bit1-报警输出2 |

|  |  |
| --- | --- |
| 设置遮挡报警配置 | |
| POST digest/frmVideoHidePara HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "AlarmTimeType": 1,  "NewSemantics": 1,  "Enable": 0,  "Sensitive": 2,  "Shelters": [  702,  0,  0,  0  ],  "AlarmTime": [  [  [  0,  0,  23,  59  ],  ……  ]  ……  ],  "EnableHandle": 0,  "HandleType": 0,  "SnapCount": 1,  "SnapInterval": 1,  "EnableAudio": 0,  "AudioNo": 1,  "AlarmOut": 0  }  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "StatusCode":"Operation Ok"  }  } |  |

### 隐私遮蔽

|  |  |
| --- | --- |
| 获取隐私遮蔽配置 | |
| POST /digest/frmVideoShelterPara HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "Shelters": [  {  "HideAreaTopLeftX": 0,  "HideAreaTopLeftY": 0,  "HideAreaWidth": 704,  "HideAreaHeight": 576  },  ……  ],  "EnableHide": 0  }  } | "EnableHide": Int型,是否启用遮挡报警:0-不启用,1-启用  "Shelters": Object型数组,隐私遮蔽区域,长度为4或5,当前设备支持的区域个数看返回的数组长度  "HideAreaTopLeftX":Int型, 隐私遮蔽区域的x坐标,范围0-704  "HideAreaTopLeftY":Int型, 隐私遮蔽区域的y坐标,范围0-576  "HideAreaWidth":Int型, 隐私遮蔽区域的宽度,范围0-704  "HideAreaHeight": Int型, 隐私遮蔽区域的高度,范围0-576 |

|  |  |
| --- | --- |
| 设置隐私遮蔽配置 | |
| POST /digest/frmVideoShelterPara HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "Shelters": [  {  "HideAreaTopLeftX": 0,  "HideAreaTopLeftY": 0,  "HideAreaWidth": 704,  "HideAreaHeight": 576  },  ……  ],  "EnableHide": 0  }  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

### 异常报警

|  |  |
| --- | --- |
| 获取异常报警配置 | |
| POST /digest/frmAlarmException HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {  "Type": 2  }  } | "Type": Int型,异常类型,2-网线断,3-IP冲突,4-非法访问 |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "HandleType": 40,  "AlarmOut": 1  }  } | "HandleType": Int型,报警类型,按位操作:  "0x00"–不触发报警  "0x01"-监视器上报警  "0x02"-声音告警  "0x04"-上传中心  "0x08"-触发报警输出  "0x10"-邮件联动  "0x20"-触发报警录像  "0x40"-屏幕截图  "0x80"-联动Ftp  "0x100"-联动Http  "0x200"-联动灯光报警  "AlarmOut": Int型,联动报警输出按位操作,bit0-报警输出1,bit1-报警输出2 |

|  |  |
| --- | --- |
| 设置异常报警配置 | |
| POST /digest/frmAlarmException HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "Type": 2,  "HandleType": 40,  "AlarmOut": 1  }  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

### 布防撤防

|  |  |
| --- | --- |
| 获取布防状态 | |
| POST /digest/frmGetAlarmInfo | |
| 请求的正文内容：  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } | "arming\_status": Int型,布防状态,1-布防,0-撤防 |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "arming\_status": 1  }  } |  |

|  |  |
| --- | --- |
| 设置布防撤防信息 | |
| POST /digest/frmGetAlarmInfo | |
| 请求的正文内容：  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {}  } | "Type": Int型,设置类型,1-设置布防,2-设置撤防 |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

## 智能报警

### 报警信息查询

|  |  |
| --- | --- |
| 获取报警信息 | |
| POST /digest/frmQueryAlarmInfo HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "AlarmInfo": [  {  "Source": "1",  "MajorType": 0,  "MinorType": 0,  "State": 1,  "StartTime": 1576425667,  "StopTime": 1576490873  },  {  "Source": "1",  "MajorType": 0,  "MinorType": 6,  "State": 0,  "StartTime": 1576490870,  "StopTime": 1576490871  }  ]  }  } | "AlarmInfo": Object型数组,报警信息列表  "Source": String型,报警的通道和区域信息  "MajorType": Int型,未使用  "MinorType": 报警事件类型,0-报警输入, 1-硬盘满,2-信号丢失, 3－移动侦测, 4－硬盘未格式化, 5-读写硬盘出错, 6-遮挡报警, 7-制式不匹配, 8-非法访问, 9-视频信号异常, 10-录像异常, 20-网线断, 21-IP冲突, 22-目标计数, 23-虚拟警戒线, 24-区域检测, 25-物品检测, 26-声音异常, 27-车牌检测, 28-人脸检测, 29-火灾检测, 30-图像偏色, 31-亮度过亮, 32-图像模糊, 33-智能移动侦测报警, 34-视频亮度过暗  "State": Int型,报警状态,1-正在触发,0-停止  "StartTime": Int型,报警开始时间,UTC时间  "StopTime": Int型,报警结束时间,UTC时间 |

### 获取智能能力集

|  |  |
| --- | --- |
| 获取智能能力集 | |
| POST /digest/frmGetSmartAbility HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "List": [  {  "CounterWire": 0  },  {  "DetectWire": 0  },  {  "DetectRegion": 0  },  {  "ObjectRegion": 0  },  {  "RecognitionFace": 0  },  {  "Retrograde": 0  },  {  "HighDensity": 0  },  {  "MaxHeight": 0  },  {  "ScenceChange": 0  }  ]  }  } | "List": Object型数组,设备支持的智能能力集以及对应的版本列表  "[Key]:[value]":支持的智能事件:智能事件的版本  "CounterWire"-目标计数  "DetectWire"-虚拟警戒线  "DetectRegion"-区域检测  "ElectronFence"-电子围栏  "ObjectRegion"-物品检测  "RecognitionFace"-人脸识别  "Retrograde"-逆行检测  "HighDensity"-密度检测  "MaxHeight"-限高检测  "ScenceChange"-场景变换  "SmartMotion"-智能移动侦测  "DetectPlate"-车牌检测  "DetectFire"-火灾检测  "VideoDiagnose"-视频诊断  "SoundDetect"-声音告警  "DetectFace"-人脸检测  "DetectPerson"-人形检测  "DetectAbsent"-离岗检测  "EBike"-电动车检测  "Elevator"-电梯状态检测 |

### 智能报警事件联动

|  |  |
| --- | --- |
| 获取智能报警事件联动配置 | |
| POST /digest/frmDetectLinkPara HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {  "DetectLinkType": 1,  "AlarmTimeType": 1  }  } | "DetectLinkType": Int型,智能报警联动类型,   1. 目标计数报警联动, 2. 虚拟警戒线报警联动 3. 区域检测(电子围栏)报警联动 4. 物品检测报警联动 5. 声音检测报警联动 6. 智能移动侦测报警联动 7. 视频诊断报警联动 8. 火灾检测报警联动 9. 车牌识别报警联动 10. 人脸检测报警联动 11. 逆行检测报警联动 12. 密度检测报警联动 13. 限高检测报警联动 14. 已废弃 15. 场景变换报警联动 16. 人形检测报警联动 17. 离岗检测报警联动 18. 安全帽检测报警联动 19. 人体测温报警联动 20. 口罩检测报警联动 21. 电动车检测报警联动 22. 电梯状态检测联动   "AlarmTimeType":Int型,时间段类型,0-每天一个时间段,1-每天八个时间段 |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "AlarmTime": [  [  [  0,  0,  23,  59  ],  ……  ],  ……  ],  "EnableHandle": 1,  "HandleType": 0,  "SnapCount": 1,  "SnapInterval": 1,  "EnablePreset": 0,  "PresetNo": 0,  "EnableAudio": 0,  "AudioNo": 1,  "AlarmOut": 0,  "Wiegand": [  {  "Enable": 0,  "Id": 1  }  ]  }  } | "AlarmTime":布防时间结构,7天,根据AlarmTimeType确定每天返回多少个时间段,"[0,0,23,59]"表示：开始时间0时0分,结束时间23时59分  "EnableHandle":Int型,联动总开关,0-不启用,1-启用  "HandleType": Int型,报警类型,按位操作:  "0x00"–不触发报警  "0x01"-监视器上报警  "0x02"-声音告警  "0x04"-上传中心  "0x08"-触发报警输出  "0x10"-邮件联动  "0x20"-触发报警录像  "0x40"-屏幕截图  "0x80"-联动Ftp  "0x100"-联动Http  "0x200"-联动灯光报警  "SnapCount": Int型,抓拍数量  "SnapInterval": Int型,抓拍间隔  "EnablePreset": Int型,是否调用预置点,1-是,0-否  "PresetNo": Int型,预置点号,范围0-255  "EnableAudio": Int型,是否联动音频 0-否,1-是  "AudioNo": Int型,联动的音频文件序号(1-8)  "AlarmOut": Int型,联动报警输出按位操作,bit0-报警输出1,bit1-报警输出2  "Wiegand": object型数组,联动韦根,数组长度就是当前设备支持联动的韦根数  "Wiegand"[]."Enable": Int型,是否联动当前韦根,1-是,0-否  "Wiegand"[]."Id": Int型,联动的韦根ID |

|  |  |
| --- | --- |
| 设置智能报警事件联动配置 | |
| POST /digest/frmDetectLinkPara HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "DetectLinkType": 1,  "AlarmTimeType": 1,  "AlarmTime": [  [  [  0,  0,  23,  59  ],  ……  ],  ……  ],  "EnableHandle": 1,  "HandleType": 0,  "SnapCount": 1,  "SnapInterval": 1,  "EnablePreset": 0,  "PresetNo": 0,  "EnableAudio": 0,  "AudioNo": 1,  "AlarmOut": 0,  "Wiegand": [  {  "Enable": 0,  "Id": 1  }  ]  }  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

### 声光报警联动项布防时间

|  |  |
| --- | --- |
| 获取声光报警联动项布防时间配置 | |
| POST /digest/frmActionAlarmTimePara | |
| 请求的正文内容：  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {  "ActionType": 1  }  } | "ActionType": Int型,智能报警联动类型,   1. 声音报警联动, 2. 光报警联动 |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "AlarmTime": [  [  [  0,  0,  23,  59  ],  ……  ],  ……  ]  }  } | "AlarmTime":布防时间结构,7天,每天8个时间段,"[0,0,23,59]"表示：开始时间0时0分,结束时间23时59分 |

|  |  |
| --- | --- |
| 设置声光报警联动项布防时间配置 | |
| POST /digest/frmActionAlarmTimePara | |
| 请求的正文内容：  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "ActionType": 1,  "AlarmTime": [  [  [  0,  0,  23,  59  ],  ……  ],  ……  ]  }  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

### 目标计数

|  |  |
| --- | --- |
| 获取目标计数配置 | |
| POST /digest/frmTargetCountPara HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "SupportLineNum": 4,  "SupportPointNum": 20,  "EnableTargetCount": 0,  "DetectIntervalMSec": 40,  "CounterPara": [  {  "DetType": 0,  "ObjectType": 256,  "CounterPoint": [  [  0,  20  ],  [  0,  0  ],  [  0,  0  ],  [  0,  0  ]  ],  "Scale": 0,  "EnableFlowCount": 0,  "CountIntervalTime": 0,  "CountAlarmThreshold": 0,  "EnableTimeFlowCount": 0,  "CountTime": [  0,  0,  0,  0  ],  "TotalAlarmThreshold": 0  },  ……  ]  }  } | "SupportLineNum": Int型,支持的最大检测线条数  "SupportPointNum": Int型,检测线最多支持的点的个数  "EnableTargetCount": Int型,是否启用目标计数,1-启用,0-不启用  "DetectIntervalMSec": Int型,检测间隔时间,单位:毫秒  "CounterPara": Object型数组,每条检测线的配置列表,长度与SupportLineNum保持一致  "DetType": Int型,离开统计方式,0-忽略,1-减少计数,2-增加计数  "ObjectType": Int型,目标类型,高3字节为子类型,低1字节为主类型,主类型默认为1,子类型定义如下:  0x001:人  0x002:自行车  0x004:摩托车  0x008:小汽车  0x010:巴士  0x020:卡车  0x040:飞机  0x080:火车  0x0100:船  "CounterPoint"[0]: Int型数组,当前检测线点的个数,以及支持的最大的点的个数  "CounterPoint"[1]: Int型数组,A点的坐标(x,y)  "CounterPoint"[2]: Int型数组,B点的坐标(x,y)  "CounterPoint"[3]: Int型数组,Result点的坐标(x,y)  "CounterPoint"[4-23]: Int型数组,所画区域点的坐标  "Scale": Int型,目标占比,范围0-10000  "EnableFlowCount": Int型,是否启用流量统计,1-是,0-否  "CountIntervalTime": Int型,流量统计间隔  "CountAlarmThreshold": Int型,流量统计阈值  "EnableTimeFlowCount": Int型,是否启用总量统计,1-是,0-否  "CountTime": Int型数组,总量统计时间段  "CountTime"[0]: Int型,总量统计开始时间(时)  "CountTime"[1]: Int型,总量统计开始时间(分)  "CountTime"[2]: Int型,总量统计结束时间(时)  "CountTime"[3]: Int型,总量统计结束时间(分)  "TotalAlarmThreshold": Int型数组,总量统计报警阈值 |

|  |  |
| --- | --- |
| 设置目标计数配置 | |
| POST /digest/frmTargetCountPara HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "SupportLineNum": 4,  "SupportPointNum": 20,  "EnableTargetCount": 0,  "DetectIntervalMSec": 40,  "CounterPara": [  {  "DetType": 0,  "ObjectType": 256,  "CounterPoint": [  [  0,  20  ],  [  0,  0  ],  [  0,  0  ],  [  0,  0  ]  ],  "Scale": 0,  "EnableFlowCount": 0,  "CountIntervalTime": 0,  "CountAlarmThreshold": 0,  "EnableTimeFlowCount": 0,  "CountTime": [  0,  0,  0,  0  ],  "TotalAlarmThreshold": 0  },  ……  ]  }  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

### 物品检测

|  |  |
| --- | --- |
| 获取物品检测配置 | |
| POST /digest/frmObjectDectPara HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "SupportRegionNum": 4,  "SupportPointNum": 20,  "EnableObjDect": 0,  "DetectIntervalMSec": 40,  "GoodsDetParam": [  {  "DetType": 0,  "Scale": 0,  "IntervalTime": 0,  "CounterPoint": [  [  0,  20  ]  ]  },  ……  ]  }  } | "SupportRegionNum": Int型,支持的最大检测区域  "SupportPointNum": Int型,检测区域最多支持的点的个数  "EnableObjDect": Int型,是否启用物品检测,1-启用,0-不启用  "DetectIntervalMSec": Int型,检测间隔时间,单位:毫秒  "GoodsDetParam": Object型数组,检测区域的配置列表,长度与SupportRegionNum保持一致  "DetType": Int型,检测类型,0-物品丢失,1-物品遗留,2-物品丢失或遗留  "Scale": Int型,目标占比,范围0-10000  "IntervalTime": Int型,检测时间  "CounterPoint"[0]: Int型数组,当前检测线点的个数,以及支持的最大的点的个数  "CounterPoint"[1-20]: Int型数组,所画区域点的坐标 |

|  |  |
| --- | --- |
| 设置物品检测配置 | |
| POST /digest/frmObjectDectPara HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "SupportRegionNum": 4,  "SupportPointNum": 20,  "EnableObjDect": 0,  "DetectIntervalMSec": 40,  "GoodsDetParam": [  {  "DetType": 0,  "Scale": 0,  "IntervalTime": 0,  "CounterPoint": [  [  0,  20  ]  ]  },  ……  ]  }  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

### 区域检测(电子围栏)

|  |  |
| --- | --- |
| 获取区域检测(电子围栏)配置 | |
| POST /digest/frmRegionDectPara HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "SupportRegionNum": 4,  "SupportPointNum": 20,  "EnableRegiDect": 0,  "DetectIntervalMSec": 40,  "AreaDetParam": [  {  "DetType": 0,  "ObjectType": 256,  "Scale": 0,  "IntervalTime": 0,  "CounterPoint": [  [  0,  20  ]  ]  },  ……  ]  }  } | "SupportRegionNum": Int型,支持的最大检测区域  "SupportPointNum": Int型,检测区域最多支持的点的个数  "EnableRegiDect": Int型,是否启用区域检测(电子围栏),1-启用,0-不启用  "DetectIntervalMSec": Int型,检测间隔时间,单位:毫秒  "AreaDetParam": Object型数组,检测区域的配置列表,长度与SupportRegionNum保持一致  "DetType": Int型,检测类型,0-目标进入,1-目标离开,2-目标进入或离开3-目标徘徊  4-入侵检测  "Scale": Int型,目标占比,范围0-10000  "ObjectType": Int型,目标类型,高3字节为子类型,低1字节为主类型,主类型默认为1,子类型定义如下:  0x001:人  0x002:自行车  0x004:摩托车  0x008:小汽车  0x010:巴士  0x020:卡车  0x040:飞机  0x080:火车  0x0100:船  "IntervalTime": Int型,检测时间  "CounterPoint"[0]: Int型数组,当前检测线点的个数,以及支持的最大的点的个数  "CounterPoint"[1-20]: Int型数组,所画区域点的坐标 |

|  |  |
| --- | --- |
| 设置区域检测(电子围栏)配置 | |
| POST /digest/frmRegionDectPara HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "SupportRegionNum": 4,  "SupportPointNum": 20,  "EnableRegiDect": 0,  "DetectIntervalMSec": 40,  "AreaDetParam": [  {  "DetType": 0,  "ObjectType": 256,  "Scale": 0,  "IntervalTime": 0,  "CounterPoint": [  [  0,  20  ]  ]  },  ……  ]  }  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

### 声音告警

|  |  |
| --- | --- |
| 获取声音告警配置 | |
| POST /digest/frmSoundAlarm HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "Enable\_infant": 0,  "DetectIntervalMSec": 40,  "SensitiveGrade\_infant": 0,  "Enable\_screaming": 0,  "SensitiveGrade\_screaming": 0,  "Enable\_gunshot": 0,  "SensitiveGrade\_gunshot": 0,  "Enable\_explosion": 0,  "SensitiveGrade\_explosion": 0  }  } | "Enable\_infant": Int型,是否检测婴儿哭声,1-是,0-否  "DetectIntervalMSec": Int型,算法检测间隔(毫秒),  "SensitiveGrade\_infant": Int型, 婴儿哭声检测灵敏度,1-低,2-中,3-高  "Enable\_screaming": Int型, 是否检测尖叫声,1-是,0-否  "SensitiveGrade\_screaming": Int型,检测灵敏度,1-低,2-中,3-高  "Enable\_gunshot": Int型, 是否检测枪声,1-是,0-否  "SensitiveGrade\_gunshot": Int型,检测灵敏度,1-低,2-中,3-高  "Enable\_explosion": Int型, 是否检测爆炸声,1-是,0-否  "SensitiveGrade\_explosion": Int型,检测灵敏度,1-低,2-中,3-高 |

|  |  |
| --- | --- |
| 设置声音告警配置 | |
| POST /digest/frmSoundAlarm HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "Enable\_infant": 0,  "DetectIntervalMSec": 40,  "SensitiveGrade\_infant": 0,  "Enable\_screaming": 0,  "SensitiveGrade\_screaming": 0,  "Enable\_gunshot": 0,  "SensitiveGrade\_gunshot": 0,  "Enable\_explosion": 0,  "SensitiveGrade\_explosion": 0  }  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

### 虚拟警戒线

|  |  |
| --- | --- |
| 获取虚拟警戒线配置 | |
| POST /digest/frmVirtualLinePara HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "SupportLineNum": 4,  "SupportPointNum": 20,  "EnableVirtuLine": 0,  "DetectIntervalMSec": 40,  "VirtualLineParam": [  {  "DetType": 0,  "Scale": 0,  "CounterPoint": [  [  0,  20  ],  [  0,  0  ],  [  0,  0  ]  ]  },  ……  ]  }  } | "SupportLineNum": Int型,支持的最大检测线条数  "SupportPointNum": Int型,检测线最多支持的点的个数  "EnableVirtuLine": Int型,是否启用虚拟警戒线,1-启用,0-不启用  "DetectIntervalMSec": Int型,检测间隔时间,单位:毫秒  "VirtualLineParam": Object型数组,检测线的配置列表,长度与SupportRegionNum保持一致  "DetType": Int型,检测类型,0-A->B报警,1- A<->B报警  "Scale": Int型,目标占比,范围0-10000  "CounterPoint"[0]: Int型数组,当前检测线点的个数,以及支持的最大的点的个数  "CounterPoint"[1]: Int型数组,A点的坐标(x,y)  "CounterPoint"[2]: Int型数组,B点的坐标(x,y)  "CounterPoint"[3-22]: Int型数组,所画区域点的坐标 |

|  |  |
| --- | --- |
| 设置虚拟警戒线配置 | |
| POST /digest/frmVirtualLinePara HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "SupportLineNum": 4,  "SupportPointNum": 20,  "EnableVirtuLine": 0,  "DetectIntervalMSec": 40,  "VirtualLineParam": [  {  "DetType": 0,  "Scale": 0,  "CounterPoint": [  [  0,  20  ],  [  0,  0  ],  [  0,  0  ]  ]  },  ……  ]  }  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

### 智能移动侦测

|  |  |
| --- | --- |
| 获取智能移动侦测配置 | |
| POST /digest/frmSmartMotion HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "Enable": 0,  "DetectIntervalMSec": 40,  "Sensitive": 0,  "MinSize": 0,  "MotionScope": [  [  0,  ……  ],  ……  ]  }  } | "Enable": Int型,是否启用智能移动侦测,1-是,0-否  "DetectIntervalMSec": Int型,检测间隔时间,单位:毫秒  "Sensitive": Int型,灵敏度：范围为0-5, 灵敏度越高越灵敏  "MinSize": Int型,目标占比,范围0-10000  "MotionScope":Int型二位数组,移动侦测范围,18x22数组,18行,每行22块,总共22\*18个块区,"1"-侦测范围, "0"-不在侦测范围内 |

|  |  |
| --- | --- |
| 设置智能移动侦测配置 | |
| POST /digest/frmSmartMotion HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "Enable": 0,  "DetectIntervalMSec": 40,  "Sensitive": 0,  "MinSize": 0,  "MotionScope": [  [  0,  ……  ],  ……  ]  }  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

### 视频诊断

|  |  |
| --- | --- |
| 获取视频诊断配置 | |
| POST /digest/frmVideoDiagnose HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "DiagnoseParam": {  "DiagnoseType": 0,  "DetectIntervalMSec": 40,  "ColorSensitivity": 1,  "DarkSensitivity": 1,  "BrightSensitivity": 1,  "StripeSensitivity": 1,  "SnowSensitivity": 1,  "ShieldSensitivity": 1,  "FreezeSensitivity": 1,  "LostSensitivity": 1,  "BlurSensitivity": 1,  "JitterSensitivity": 1,  "PtzSensitivity": 1  }  }  } | "DiagnoseType": Int型,诊断类型,按位操作  0x00-停止诊断  0x01-图像偏色  0x02-亮度过暗  0x04-亮度过亮  0x08-条纹干扰  0x10-雪花干扰  0x20-视频遮挡  0x40-画面冻结  0x80-视频丢失  0x100-图像模糊  0x200-画面抖动  0x400-PTZ异常  "DetectIntervalMSec": Int型,算法检测间隔(毫秒)  "ColorSensitivity": Int型,图像偏色灵敏度,范围1-100  "DarkSensitivity": Int型,亮度过暗灵敏度,范围1-100  "BrightSensitivity": Int型,亮度过亮灵敏度,范围1-100  "StripeSensitivity": Int型,条纹干扰灵敏度,范围1-100  "SnowSensitivity": Int型,雪花干扰灵敏度,范围1-100  "ShieldSensitivity": Int型,视频遮挡灵敏度,范围1-100  "FreezeSensitivity": Int型,画面冻结灵敏度,范围1-100  "LostSensitivity": Int型,视频丢失灵敏度,范围1-100(暂未实现)  "BlurSensitivity": Int型,图像模糊灵敏度,范围1-100  "JitterSensitivity": Int型,画面抖动灵敏度,范围1-100(暂未实现)  "PtzSensitivity": Int型,PTZ异常灵敏度,范围1-100(暂未实现) |

|  |  |
| --- | --- |
| 设置视频诊断配置 | |
| POST /digest/frmVideoDiagnose HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "DiagnoseParam": {  "DiagnoseType": 0,  "DetectIntervalMSec": 40,  "ColorSensitivity": 1,  "DarkSensitivity": 1,  "BrightSensitivity": 1,  "StripeSensitivity": 1,  "SnowSensitivity": 1,  "ShieldSensitivity": 1,  "FreezeSensitivity": 1,  "LostSensitivity": 1,  "BlurSensitivity": 1,  "JitterSensitivity": 1,  "PtzSensitivity": 1  }  }  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

### 火灾检测

|  |  |
| --- | --- |
| 获取火灾检测配置 | |
| POST /digest/frmVideoFireDetect HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "Enable": 0,  "Sensitive": 100,  "DetectIntervalMSec": 40,  "LeftTopX": 101,  "LeftTopY": 104,  "RectWidth": 464,  "RectHeight": 377  }  } | "Enable": Int型,是否启用火灾检测,1-是,0-否  "DetectIntervalMSec": Int型,检测间隔时间,单位:毫秒  "Sensitive": Int型,灵敏度：范围为0-100, 灵敏度越高越灵敏  "LeftTopX": Int型,检测区域的x坐标,范围:0-704  "LeftTopY": Int型,检测区域的y坐标,范围:0-576  "RectWidth": Int型,检测区域的宽,范围:0-704  "RectHeight": Int型,检测区域的高,范围:0-576 |

|  |  |
| --- | --- |
| 设置火灾检测配置 | |
| POST /digest/frmVideoFireDetect HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "Enable": 0,  "Sensitive": 100,  "DetectIntervalMSec": 40,  "LeftTopX": 101,  "LeftTopY": 104,  "RectWidth": 464,  "RectHeight": 377  }  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

### 车牌识别

|  |  |
| --- | --- |
| 获取车牌识别配置 | |
| POST /digest/frmVideoPlateDetect HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "Enable": 1,  "Sensitive": 53,  "PlateProvince": 2,  "DetectIntervalMSec": 40,  "LeftTopX": 37,  "LeftTopY": 13,  "RectWidth": 629,  "RectHeight": 526  }  } | "Enable": Int型,是否启用车牌识别,1-是,0-否  "DetectIntervalMSec": Int型,检测间隔时间,单位:毫秒  "Sensitive": Int型,灵敏度：范围为0-100, 灵敏度越高越灵敏  "PlateProvince": Int型,车牌检测时默认的省份  0-四川,1-重庆,2-湖北,3-江西,4-广西,5-黑龙江,6-上海,7-河北,8-天津,9-北京  10-吉林,11-辽宁,12-山东,13-福建,14-宁夏,15-青海,16-海南,17-陕西,18-江苏  19-山西,20-安徽,21-湖南,22-河南,23-广东,24-云南,25-西藏,26-浙江,27-内蒙古  28-甘肃,29-新疆,30-贵州,31-未设置  "LeftTopX": Int型,检测区域的x坐标,范围:0-704  "LeftTopY": Int型,检测区域的y坐标,范围:0-576  "RectWidth": Int型,检测区域的宽,范围:0-704  "RectHeight": Int型,检测区域的高,范围:0-576 |

|  |  |
| --- | --- |
| 设置车牌识别配置 | |
| POST /digest/frmVideoPlateDetect HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "Enable": 1,  "Sensitive": 53,  "PlateProvince": 2,  "DetectIntervalMSec": 40,  "LeftTopX": 37,  "LeftTopY": 13,  "RectWidth": 629,  "RectHeight": 526  }  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

### 人脸检测

|  |  |
| --- | --- |
| 获取人脸检测配置 | |
| POST /digest/frmVideoFaceDetectV2 HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "Enable": 0,  "DetectIntervalMSec": 0,  "LandMark": 1,  "MinSize": 80,  "Sensitivity": 0,  "FaceExtRatioW": 0,  "FaceExtRatioH": 0,  "CaptureQuality": 99,  "OnlyMaxFaceEnable": 1,  "DetectRegion": {  "X": 0,  "Y": 0,  "W": 1,  "H": 1  },  "MaskRegion0": {  "X": 0,  "Y": 0,  "W": 0,  "H": 0  },  "MaskRegion1": {  "X": 0,  "Y": 0,  "W": 0,  "H": 0  },  "MaskRegion2": {  "X": 0,  "Y": 0,  "W": 0,  "H": 0  },  "MaskRegion3": {  "X": 0,  "Y": 0,  "W": 0,  "H": 0  },  "EnableFaceExposure": 0,  "FaceExposureDelaySec": 60,  "FaceExposureBrightness": 55,  "SnapFaceEnable": 1,  "EnableMaxImageWidth": 1,  "CaptureMaxWidth": 1920,  "EnableMaxFaceWidth": 1,  "CaptureMaxFaceWidth": 1920,  "EnableKeyScore": 0,  "CaptureType": 0,  "CaptureIntervalSec": 0,  "CaptureIntervalMSec": 2,  "CaptureInSec": 0,  "CaptureInMSec": 1,  "CaptureInNum": 1,  "CaptureOutNum": 1,  "FtpEnable": 1,  "FtpFullJpeg": 0,  "CaptureFullQuality": 99,  "EnableDrawFace": 0,  "RecEnable": 1,  "RecMode": 1,  "RecSimilarity": 60,  "FaceRecIntervalMSec": 0,  "DetectFaceLiveness": 0,  "SensitivityType": 0,  "HttpEnable": 0,  "ImageType": 0,  "MaxSize": 1920,  "DrawFaceRGB": "000000"  }  } | "Enable": Int型,是否启用人脸检测,1-是,0-否  "DetectIntervalMSec": Int型,检测间隔时间,单位:毫秒  "LandMark": Int型,标定点(暂时不能修改)  "MinSize": Int型,目标占比,范围0-10000  "Sensitivity": Int型,灵敏度：范围为1-100, 灵敏度越高越灵敏  "FaceExtRatioW": Int型,人脸外扩系数(宽),范围0-100  "FaceExtRatioH": Int型,人脸外扩系数(高),范围0-100  "CaptureQuality": Int型,抓拍图像质量,范围1-99  "OnlyMaxFaceEnable": Int型,只检测最大人脸,1-是,0-否  "DetectRegion":Object型,人脸检测区域(在该区域就检测) X,Y,W,H,的取值范围为0-1的小数  "DetectRegion"."X": Float型,检测区域的x坐标,范围0-1  "DetectRegion"."Y": Float型,检测区域的y坐标,范围0-1  "DetectRegion"."W": Float型,检测区域的宽,范围0-1  "DetectRegion"."H": Float型,检测区域的高,范围0-1  "MaskRegion[0-3]":Object型,人脸屏蔽区域(在该区域人脸不检测)  "MaskRegion[0-3]"."X": Float型,屏蔽区域的x坐标,范围0-1  "MaskRegion[0-3]"."Y": Float型,屏蔽区域的y坐标,范围0-1  "MaskRegion[0-3]"."W": Float型,屏蔽区域的宽,范围0-1  "MaskRegion[0-3]"."H": Float型,屏蔽区域的高,范围0-1  "EnableFaceExposure": Int型,是否启用人脸曝光策略,1-启用,0-不启用  "FaceExposureDelaySec": Int型,人脸曝光延时时间,单位:毫秒  "FaceExposureBrightness": Int型,参考亮度,范围0-100  "SnapFaceEnable": Int型,是否启用人脸抓拍,1-是,0-否  "EnableMaxImageWidth": Int型,是否限制抠图图片尺寸,1-是,0-否  "CaptureMaxWidth": Int型,抠图图片最大宽度,范围32-1920  "EnableMaxFaceWidth": Int型,是否限制抠图人脸尺寸,1-是,0-否"CaptureMaxFaceWidth": Int型,抠图人脸最大宽度,范围32-1920  "EnableKeyScore": Int型,是否启用关键分,1-是,0-否  "CaptureType": Int型,抠图模式,0-间隔模式,1-进入模式,2-离开模式,3-进入或离开模式  "CaptureIntervalSec": Int型,间隔模式抓图间隔时间(秒),范围1-10  "CaptureIntervalMSec": Int型,未使用,保留字段  "CaptureInSec": Int型,进入模式抓图间隔时间(秒),范围1-5  "CaptureInMSec": Int型,未使用,保留字段  "CaptureInNum": Int型,进入模式抓图数量,范围1-3  "CaptureOutNum": Int型,离开模式抓图数量,范围1-3  "FtpEnable": Int型,是否启用FTP,1-是,0-否  "FtpFullJpeg": Int型,是否启用场景图上传,1-是,0-否  "CaptureFullQuality": Int型,场景图图像质量(V1.4.1新增)  "EnableDrawFace": Int型,是否启用场景人脸标识,1-是,0-否  "RecEnable": Int型,是否启用人脸识别,1-是,0-否  "RecMode": Int型,未使用,保留字段  "RecSimilarity": Int型,未使用,保留字段  "FaceRecIntervalMSec": Int型,人脸识别间隔时间,单位:秒  "DetectFaceLiveness": Int型,未使用,保留字段  "SensitivityType": Int型,未使用,保留字段  "HttpEnable": Int型,未使用,保留字段  "ImageType": Int型,未使用,保留字段  "MaxSize": Int型,人脸最大占比,范围0-10000  "DrawFaceRGB": String型,标识颜色 |

|  |  |
| --- | --- |
| 设置人脸检测配置 | |
| POST /digest/frmVideoFaceDetectV2 HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "Enable": 0,  "DetectIntervalMSec": 0,  "LandMark": 1,  "MinSize": 80,  "Sensitivity": 0,  "FaceExtRatioW": 0,  "FaceExtRatioH": 0,  "CaptureQuality": 99,  "OnlyMaxFaceEnable": 1,  "DetectRegion": {  "X": 0,  "Y": 0,  "W": 1,  "H": 1  },  "MaskRegion0": {  "X": 0,  "Y": 0,  "W": 0,  "H": 0  },  "MaskRegion1": {  "X": 0,  "Y": 0,  "W": 0,  "H": 0  },  "MaskRegion2": {  "X": 0,  "Y": 0,  "W": 0,  "H": 0  },  "MaskRegion3": {  "X": 0,  "Y": 0,  "W": 0,  "H": 0  },  "EnableFaceExposure": 0,  "FaceExposureDelaySec": 60,  "FaceExposureBrightness": 55,  "SnapFaceEnable": 1,  "EnableMaxImageWidth": 1,  "CaptureMaxWidth": 1920,  "EnableMaxFaceWidth": 1,  "CaptureMaxFaceWidth": 1920,  "EnableKeyScore": 0,  "CaptureType": 0,  "CaptureIntervalSec": 0,  "CaptureIntervalMSec": 2,  "CaptureInSec": 0,  "CaptureInMSec": 1,  "CaptureInNum": 1,  "CaptureOutNum": 1,  "FtpEnable": 1,  "FtpFullJpeg": 0,  "CaptureFullQuality": 99,  "EnableDrawFace": 0,  "RecEnable": 1,  "RecMode": 1,  "RecSimilarity": 60,  "FaceRecIntervalMSec": 0,  "DetectFaceLiveness": 0,  "SensitivityType": 0,  "HttpEnable": 0,  "ImageType": 0,  "MaxSize": 1920,  "DrawFaceRGB": "000000"  }  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

### 逆行检测

|  |  |
| --- | --- |
| 获取逆行检测配置 | |
| POST /digest/frmRetrogradePara HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "SupportLineNum": 4,  "SupportPointNum": 20,  "Enable": 0,  "DetectIntervalMSec": 40,  "RetrogradeParam": [  {  "DetType": 0,  "Scale": 0,  "ObjectType": 256,  "CounterPoint": [  [  0,  20  ],  [  0,  0  ],  [  0,  0  ]  ]  },  ……  ]  }  } | "SupportLineNum": Int型,支持的最大检测线条数  "SupportPointNum": Int型,检测线最多支持的点的个数  "Enable": Int型,是否启用逆行检测,1-启用,0-不启用  "DetectIntervalMSec": Int型,检测间隔时间,单位:毫秒  "RetrogradeParam": Object型数组,检测线的配置列表,长度与SupportRegionNum保持一致  "DetType": Int型,检测类型,未使用  "Scale": Int型,目标占比,范围0-10000  "ObjectType": Int型,目标类型,高3字节为子类型,低1字节为主类型,主类型默认为1,子类型定义如下:  0x001:人  0x002:自行车  0x004:摩托车  0x008:小汽车  0x010:巴士  0x020:卡车  0x040:飞机  0x080:火车  0x0100:船  "CounterPoint"[0]: Int型数组,当前检测线点的个数,以及支持的最大的点的个数  "CounterPoint"[1]: Int型数组,A点的坐标(x,y)  "CounterPoint"[2]: Int型数组,B点的坐标(x,y)  "CounterPoint"[3-22]: Int型数组,所画区域点的坐标 |

|  |  |
| --- | --- |
| 设置逆行检测配置 | |
| POST /digest/frmRetrogradePara HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "SupportLineNum": 4,  "SupportPointNum": 20,  "Enable": 0,  "DetectIntervalMSec": 40,  "RetrogradeParam": [  {  "DetType": 0,  "Scale": 0,  "ObjectType": 256,  "CounterPoint": [  [  0,  20  ],  [  0,  0  ],  [  0,  0  ]  ]  },  ……  ]  }  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

### 离岗检测

|  |  |
| --- | --- |
| 获取离岗检测配置 | |
| POST /digest/frmAbsentDectPara HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "SupportAbsentNum": 4,  "SupportPointNum": 20,  "EnableAbsentDect": 0,  "DetectIntervalMSec": 40,  "AbsentDectParam": [  {  "DetType": 0,  "Scale": 0,  "IntervalTime": 5,  "DetectMotion": 0,  "CounterPoint": [  [  0,  20  ]  ]  },  ……  ]  }  } | "SupportAbsentNum": Int型,支持的最大检测区域数  "SupportPointNum": Int型,检测区域最多支持的点的个数  "EnableAbsentDect": Int型,是否启用离岗检测,1-启用,0-不启用  "DetectIntervalMSec": Int型,检测间隔时间,单位:毫秒  "RetrogradeParam": Object型数组,检测线的配置列表,长度与SupportRegionNum保持一致  "DetType": Int型,检测类型,未使用  "Scale": Int型,目标占比,范围0-10000  "IntervalTime": Int型,离岗时间(秒),范围0-3600  "DetectMotion": Int型,是否检测运动物体,1-是,0-否  "CounterPoint"[0]: Int型数组,当前检测区域点的个数,以及支持的最大的点的个数  "CounterPoint"[1]: Int型数组,A点的坐标(x,y)  "CounterPoint"[2]: Int型数组,B点的坐标(x,y)  "CounterPoint"[3-22]: Int型数组,所画区域点的坐标 |

|  |  |
| --- | --- |
| 设置离岗检测配置 | |
| POST /digest/frmAbsentDectPara HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "SupportAbsentNum": 4,  "SupportPointNum": 20,  "EnableAbsentDect": 0,  "DetectIntervalMSec": 40,  "AbsentDectParam": [  {  "DetType": 0,  "Scale": 0,  "IntervalTime": 5,  "DetectMotion": 0,  "CounterPoint": [  [  0,  20  ]  ]  },  ……  ]  }  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

### 密度检测

|  |  |
| --- | --- |
| 获取密度检测配置 | |
| POST /digest/frmDensityPara HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "SupportRegionNum": 4,  "SupportPointNum": 20,  "Enable": 0,  "DetectIntervalMSec": 40,  "DensityParam": [  {  "Sensitivity": 1,  "Scale": 0,  "CounterPoint": [  [  0,  20  ]  ]  },  ……  ]  }  } | "SupportRegionNum": Int型,支持的最大检测区域数  "SupportPointNum": Int型,检测区域最多支持的点的个数  "Enable": Int型,是否启用密度检测,1-启用,0-不启用  "DetectIntervalMSec": Int型,检测间隔时间,单位:毫秒  "DensityParam": Object型数组,检测区域的配置列表,长度与SupportRegionNum保持一致  "Sensitivity": Int型,灵敏度,范围1-100  "Scale": Int型,目标占比,范围0-10000  "CounterPoint"[0]: Int型数组,当前检测区域点的个数,以及支持的最大的点的个数  "CounterPoint"[1-20]: Int型数组,所画区域点的坐标 |

|  |  |
| --- | --- |
| 设置密度检测配置 | |
| POST /digest/frmDensityPara HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "SupportRegionNum": 4,  "SupportPointNum": 20,  "Enable": 0,  "DetectIntervalMSec": 40,  "DensityParam": [  {  "Sensitivity": 1,  "Scale": 0,  "CounterPoint": [  [  0,  20  ]  ]  },  ……  ]  }  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

### 限高检测

|  |  |
| --- | --- |
| 获取限高检测配置 | |
| POST /digest/frmMaxHeightPara HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "SupportLineNum": 4,  "SupportPointNum": 20,  "Enable": 0,  "DetectIntervalMSec": 40,  "VirtualLineParam": [  {  "DetType": 0,  "Scale": 0,  "CounterPoint": [  [  0,  20  ],  [  0,  0  ],  [  0,  0  ]  ]  },  ……  ]  }  } | "SupportLineNum": Int型,支持的最大检测线条数  "SupportPointNum": Int型,检测线最多支持的点的个数  "Enable": Int型,是否启用限高检测,1-启用,0-不启用  "DetectIntervalMSec": Int型,检测间隔时间,单位:毫秒  "VirtualLineParam": Object型数组,检测线的配置列表,长度与SupportRegionNum保持一致  "DetType": Int型,检测类型  "Scale": Int型,目标占比,范围0-10000  "CounterPoint"[0]: Int型数组,当前检测线点的个数,以及支持的最大的点的个数  "CounterPoint"[1]: Int型数组,A点的坐标(x,y)  "CounterPoint"[2]: Int型数组,B点的坐标(x,y)  "CounterPoint"[3-22]: Int型数组,所画区域点的坐标 |

|  |  |
| --- | --- |
| 设置限高检测配置 | |
| POST /digest/frmMaxHeightPara HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "SupportLineNum": 4,  "SupportPointNum": 20,  "Enable": 0,  "DetectIntervalMSec": 40,  "VirtualLineParam": [  {  "DetType": 0,  "Scale": 0,  "CounterPoint": [  [  0,  20  ],  [  0,  0  ],  [  0,  0  ]  ]  },  ……  ]  }  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

### 场景变换

|  |  |
| --- | --- |
| 获取场景变换配置 | |
| POST /digest/frmVideoSceneChange HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "Enable": 0,  "DetectIntervalMSec": 40  }  } | "Enable": Int型,是否启用场景变换,1-是,0-否  "DetectIntervalMSec": Int型,检测间隔时间,单位:毫秒 |

|  |  |
| --- | --- |
| 设置场景变换配置 | |
| POST /digest/frmVideoSceneChange HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "Enable": 0,  "DetectIntervalMSec": 40  }  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

### 人形检测

|  |  |
| --- | --- |
| 获取人形检测配置 | |
| POST /digest/frmVideoPersonPara HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "Enable": 0,  "DetectIntervalMSec": 40,  "LandMark": 1,  "MinSize": 80,  "Sensitivity": 3,  "ExtRatioW": 40,  "ExtRatioH": 85,  "CaptureQuality": 99,  "OnlyMaxEnable": 0,  "DetectRegion": {  "X": 0,  "Y": 0,  "W": 1,  "H": 1  },  "MaskRegion0": {  "X": 0,  "Y": 0,  "W": 0,  "H": 0  },  "MaskRegion1": {  "X": 0,  "Y": 0,  "W": 0,  "H": 0  },  "MaskRegion2": {  "X": 0,  "Y": 0,  "W": 0,  "H": 0  },  "MaskRegion3": {  "X": 0,  "Y": 0,  "W": 0,  "H": 0  },  "EnableExposure": 0,  "ExposureDelaySec": 60,  "ExposureBrightness": 50,  "SnapEnable": 1,  "EnableMaxImageWidth": 0,  "CaptureMaxWidth": 320,  "EnableMaxWidth": 0,  "CaptureMaxPersonWidth": 1920,  "EnableKeyScore": 0,  "CaptureType": 0,  "CaptureIntervalSec": 1,  "CaptureIntervalMSec": 0,  "CaptureInSec": 1,  "CaptureInMSec": 0,  "CaptureInNum": 1,  "CaptureOutNum": 1,  "FtpEnable": 0,  "SnapFullJpeg": 0,  "CaptureFullQuality": 50,  "EnableDraw": 0,  "RecEnable": 1,  "RecMode": 1,  "RecSimilarity": 60,  "RecIntervalMSec": 300,  "HttpEnable": 0,  "FtpFullJpeg": 0,  "DrawRGB": "000000"  }  } | "Enable": Int型,是否启用人形检测,1-是,0-否  "DetectIntervalMSec": Int型,检测间隔时间,单位:毫秒  "LandMark": Int型,标定点(暂时不能修改)  "MinSize": Int型,目标占比,范围0-10000  "Sensitivity": Int型,灵敏度：范围为1-100, 灵敏度越高越灵敏  "ExtRatioW": Int型,人形外扩系数(宽),范围0-100  "ExtRatioH": Int型,人形外扩系数(高),范围0-100  "CaptureQuality": Int型,抓拍图像质量,范围1-99  "OnlyMaxEnable": Int型,只检测最大人形,1-是,0-否  "DetectRegion":Object型,人形检测区域(在该区域就检测)  "DetectRegion"."X": Float型,检测区域的x坐标,范围0-1  "DetectRegion"."Y": Float型,检测区域的y坐标,范围0-1  "DetectRegion"."W": Float型,检测区域的宽,范围0-1  "DetectRegion"."H": Float型,检测区域的高,范围0-1  "MaskRegion[0-3]":Object型,人形屏蔽区域(在该区域人脸不检测)  "MaskRegion[0-3]"."X": Float型,屏蔽区域的x坐标,范围0-1  "MaskRegion[0-3]"."Y": Float型,屏蔽区域的y坐标,范围0-1  "MaskRegion[0-3]"."W": Float型,屏蔽区域的宽,范围0-1  "MaskRegion[0-3]"."H": Float型,屏蔽区域的高,范围0-1  "EnableExposure": Int型,是否启用人形曝光策略,1-启用,0-不启用  "ExposureDelaySec": Int型,人形曝光延时时间,单位:毫秒  "ExposureBrightness": Int型,参考亮度,范围0-100  "SnapEnable": Int型,是否启用人形抓拍,1-是,0-否  "EnableMaxImageWidth": Int型,是否限制抠图图片尺寸,1-是,0-否  "CaptureMaxWidth": Int型,抠图图片最大宽度,范围32-1920  "EnableMaxWidth": Int型,是否限制抠图人形尺寸,1-是,0-否"CaptureMaxPersonWidth": Int型,抠图人形最大宽度,范围32-1920  "EnableKeyScore": Int型,是否启用关键分,1-是,0-否  "CaptureType": Int型,抠图模式,0-间隔模式,1-进入模式,2-离开模式,3-进入或离开模式  "CaptureIntervalSec": Int型,间隔模式抓图间隔时间(秒),范围1-10  "CaptureIntervalMSec": Int型,未使用,保留字段  "CaptureInSec": Int型,进入模式抓图间隔时间(秒),范围1-5  "CaptureInMSec": Int型,未使用,保留字段  "CaptureInNum": Int型,进入模式抓图数量,范围1-3  "CaptureOutNum": Int型,离开模式抓图数量,范围1-3  "FtpEnable": Int型,是否启用FTP,1-是,0-否  "SnapFullJpeg": Int型, 是否启用场景图抓拍,1-是,0-否  "CaptureFullQuality": Int型,场景图图像质量(V1.4.1新增)  "EnableDraw": Int型,是否启用场景人形标识,1-是,0-否  "RecEnable": Int型,是否启用人形识别,1-是,0-否  "RecMode": Int型,未使用,保留字段  "RecSimilarity": Int型,未使用,保留字段  "RecIntervalMSec": Int型,人形识别间隔时间,单位:秒  "HttpEnable": Int型,未使用,保留字段  "FtpFullJpeg": Int型,是否启用场景图上传,1-是,0-否  "DrawFaceRGB": String型,标识颜色 |

|  |  |
| --- | --- |
| 设置人形检测配置 | |
| POST /digest/frmVideoPersonPara HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "Enable": 0,  "DetectIntervalMSec": 40,  "LandMark": 1,  "MinSize": 80,  "Sensitivity": 3,  "ExtRatioW": 40,  "ExtRatioH": 85,  "CaptureQuality": 99,  "OnlyMaxEnable": 0,  "DetectRegion": {  "X": 0,  "Y": 0,  "W": 1,  "H": 1  },  "MaskRegion0": {  "X": 0,  "Y": 0,  "W": 0,  "H": 0  },  "MaskRegion1": {  "X": 0,  "Y": 0,  "W": 0,  "H": 0  },  "MaskRegion2": {  "X": 0,  "Y": 0,  "W": 0,  "H": 0  },  "MaskRegion3": {  "X": 0,  "Y": 0,  "W": 0,  "H": 0  },  "EnableExposure": 0,  "ExposureDelaySec": 60,  "ExposureBrightness": 50,  "SnapEnable": 1,  "EnableMaxImageWidth": 0,  "CaptureMaxWidth": 320,  "EnableMaxWidth": 0,  "CaptureMaxPersonWidth": 1920,  "EnableKeyScore": 0,  "CaptureType": 0,  "CaptureIntervalSec": 1,  "CaptureIntervalMSec": 0,  "CaptureInSec": 1,  "CaptureInMSec": 0,  "CaptureInNum": 1,  "CaptureOutNum": 1,  "FtpEnable": 0,  "SnapFullJpeg": 0,  "CaptureFullQuality": 50,  "EnableDraw": 0,  "RecEnable": 1,  "RecMode": 1,  "RecSimilarity": 60,  "RecIntervalMSec": 300,  "HttpEnable": 0,  "FtpFullJpeg": 0,  "DrawRGB": "000000"  }  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

### 安全帽检测

|  |  |
| --- | --- |
| 获取安全帽检测配置 | |
| POST /digest/frmHelmetDectPara HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "Enable": 0,  "MinSize": 20,  "MaskRegion0": {  "X": 0,  "Y": 0.01,  "W": 0.1999,  "H": 0.1999  },  "MaskRegion1": {  "X": 0.0208,  "Y": 0.0208,  "W": 0.1999,  "H": 0.1999  },  "MaskRegion2": {  "X": 0.103,  "Y": 0.103,  "W": 0.3,  "H": 0.3  },  "MaskRegion3": {  "X": 0.1399,  "Y": 0.1399,  "W": 0.3999,  "H": 0.3999  },  "DetectRegion": {  "X": 0,  "Y": 0.5,  "W": 1,  "H": 0.1999  }  }  } | "Enable": Int型,是否启用安全帽检测,1-是,0-否  "MinSize": Int型,目标占比,范围0-10000  "DetectRegion":Object型,检测区域(在该区域就检测)  "DetectRegion"."X": Float型,检测区域的x坐标,范围0-1  "DetectRegion"."Y": Float型,检测区域的y坐标,范围0-1  "DetectRegion"."W": Float型,检测区域的宽,范围0-1  "DetectRegion"."H": Float型,检测区域的高,范围0-1  "MaskRegion[0-3]":Object型,检测屏蔽区域(在该区域人脸不检测)  "MaskRegion[0-3]"."X": Float型,屏蔽区域的x坐标,范围0-1  "MaskRegion[0-3]"."Y": Float型,屏蔽区域的y坐标,范围0-1  "MaskRegion[0-3]"."W": Float型,屏蔽区域的宽,范围0-1  "MaskRegion[0-3]"."H": Float型,屏蔽区域的高,范围0-1 |

|  |  |
| --- | --- |
| 设置安全帽检测配置 | |
| POST /digest/frmHelmetDectPara HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "Enable": 0,  "MinSize": 20,  "MaskRegion0": {  "X": 0,  "Y": 0.01,  "W": 0.1999,  "H": 0.1999  },  "MaskRegion1": {  "X": 0.0208,  "Y": 0.0208,  "W": 0.1999,  "H": 0.1999  },  "MaskRegion2": {  "X": 0.103,  "Y": 0.103,  "W": 0.3,  "H": 0.3  },  "MaskRegion3": {  "X": 0.1399,  "Y": 0.1399,  "W": 0.3999,  "H": 0.3999  },  "DetectRegion": {  "X": 0,  "Y": 0.5,  "W": 1,  "H": 0.1999  }  }  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

### 电动车检测

|  |  |
| --- | --- |
| 获取电动车检测配置 | |
| POST /digest/frmEBikeCfg HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "SupportRegionNum": 4,  "SupportPointNum": 20,  "Enable": 0,  "DetectIntervalMSec": 40,  "Similarity": -1,  "DetectParam": [  {  "ObjectType": 1,  "Scale": 0,  "IntervalTime": 0,  "CounterPoint": [  [  0,  20  ]  ]  },  ……  ]  }  } | "SupportRegionNum": Int型,支持的最大检测区域  "SupportPointNum": Int型,检测区域最多支持的点的个数  "Enable": Int型,是否启用电动车检测,1-启用,0-不启用  "DetectIntervalMSec": Int型,检测间隔时间,单位:毫秒  "Similarity": Int型,灵敏度,-1:不支持,0-10:支持  "DetectParam": Object型数组,检测区域的配置列表,长度与SupportRegionNum保持一致  "Scale": Int型,目标占比,范围0-10000  "ObjectType": Int型,目标类型,高3字节为子类型,低1字节为主类型,主类型默认为1,主类型定义如下:  0:人  1:车  2:全部  子类型定义如下:  0x002:自行车  0x004:电动车  "IntervalTime": Int型,检测时间  "CounterPoint"[0]: Int型数组,当前检测线点的个数,以及支持的最大的点的个数  "CounterPoint"[1-20]: Int型数组,所画区域点的坐标 |
|  |  |

|  |  |
| --- | --- |
| 设置电动车检测配置 | |
| POST /digest/frmEBikeCfg HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "SupportRegionNum": 4,  "SupportPointNum": 20,  "Enable": 0,  "Similarity": -1,  "DetectIntervalMSec": 40,  "DetectParam": [  {  "ObjectType": 1,  "Scale": 0,  "IntervalTime": 0,  "CounterPoint": [  [  0,  20  ]  ]  },  ……  ]  }  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

### 高抛检测

高抛检测功能,是指通过视频分析检测出画面中存在的抛物现象,高抛检测摄像机通常应用于小区物业管理的安全系统中. 高抛检测功能的接口包括:

(1)获取高抛算法版本信息.

(2)获取及修改高抛检测基本配置.

(3)获取及修改高抛算法配置.

(4)获取及修改高抛事件联动配置.

#### 高抛算法版本信息

|  |  |
| --- | --- |
| 获取高抛检测的算法版本信息(此接口不支持设置). | |
| POST /digest/frmFallVer HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "Scheme":"FD-A1",  "Version":"V0.1.20210517"  }  } | "Scheme": String型,算法方案. 不同算法那方案的逻辑不同,参数配置存在差异. 如"FD-A1"支持算法运行参数设置,"FD-B1"则不支持算法运行参数配置.  "Version": String型,算法版本. |

#### 高抛检测基本配置

|  |  |
| --- | --- |
| 获取高抛检测的基本配置(使能,检测区域,楼层标定,事件联动) | |
| POST /digest/frmFallBasicParam HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "Enable": 1,  "FloorNum": 0,  "Floors": [{  "Name": "5F",  "Line": [100, 7000, 1500, 7000]  }, {  "Name": "10F",  "Line": [100, 3500, 1500, 3500]  }  ],  "IncludeRegionNum": 2,  "IncludeRegions": [{  "VertexCnt": 4,  "VertexPts": [0, 0, 10000, 0, 10000, 10000, 0, 10000]  }, {  "VertexCnt": 4,  "VertexPts": [100, 100, 10000, 100, 10000, 10000, 100, 10000]  }  ],  "ExcludeRegionNum": 1,  "ExcludeRegions": [{  "VertexCnt": 4,  "VertexPts": [200, 500, 900, 500, 900, 10000, 0, 10000]  }  ],  Linkage:{  "AlarmTime": [  [  [  0, 0, 23, 59  ]  ],  ……  ],  "HandleType": 0,  "SnapCount": 1,  "SnapInterval": 1,  "EnableAudio": 0,  "AudioNo": 1,  "AlarmOut": 0,  "EnablePreset": 0,  "PresetNo": 0  }  }  } | "Enable": Int型,是否使能高抛检测功能.  "FloorNum": Int型,楼层标定中,总楼层数量.  "Floors": Array\_of\_Struct型,楼层的名称和位置. 其中"Line"数组由四个Int组成,分别表示连线的起点(x,y)和终点(x,y). 注意数字为相对坐标,表示视频宽高的万分比.  检测区域的形状不固定,从三角形/四边形到N边形.  "IncludeRegionNum": Int型,检测区域的数量.  "IncludeRegions":检测区域构成的数组. 其中,"VertexCnt"表示该区域的顶点数量. "VertexPts"数组依次表示每个顶点的相对坐标(画面尺寸万分比)的x和y.所以"VertexPts"数组的长度应该是VertexCnt的两倍.  "ExcludeRegionNum": Int型,检测区域的数量.  "ExcludeRegions":检测区域构成的数组. 其中,"VertexCnt"表示该区域的顶点数量. "VertexPts"数组依次表示每个顶点的相对坐标(画面尺寸万分比)的x和y.所以"VertexPts"数组的长度应该是VertexCnt的两倍.  "Linkage":高抛事件联动配置.详情如下.  "AlarmTime":布防时间结构,7天,每天8个时间段,"[0,0,23,59]"表示：开始时间0时0分,结束时间23时59分  "HandleType": Int型,报警类型,按位操作:  "0x00"–不触发报警  "0x01"-监视器上报警  "0x02"-声音告警  "0x04"-上传中心  "0x08"-触发报警输出  "0x10"-邮件联动  "0x20"-触发报警录像  "0x40"-屏幕截图  "0x80"-联动Ftp  "0x100"-联动Http  "0x200"-联动灯光报警  "SnapCount": Int型,抓拍数量  "SnapInterval": Int型,抓拍间隔  "EnablePreset": Int型,是否调用预置点,1-是,0-否(对于高抛相机来说预置点通常无用,建议界面隐藏)  "PresetNo": Int型,预置点号,范围0-255(对于高抛相机来说预置点通常无用,建议界面隐藏)  "EnableAudio": Int型,是否联动音频 0-否,1-是  "AudioNo": Int型,联动的音频文件序号(1-8)  "AlarmOut": Int型,联动报警输出按位操作,bit0-报警输出1,bit1-报警输出2 |
|  |  |

|  |  |
| --- | --- |
| 设置高抛检测的基本配置(使能,检测区域,楼层标定) | |
| POST /digest/frmFallBasicParam HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "Enable": 1,  "FloorNum": 0,  "Floors": [{  "Name": "5F",  "Line": [100, 7000, 1500, 7000]  }, {  "Name": "10F",  "Line": [100, 3500, 1500, 3500]  }  ],  "IncludeRegionNum": 2,  "IncludeRegions": [{  "VertexCnt": 4,  "VertexPts": [0, 0, 10000, 0, 10000, 10000, 0, 10000]  }, {  "VertexCnt": 4,  "VertexPts": [100, 100, 10000, 100, 10000, 10000, 100, 10000]  }  ],  "ExcludeRegionNum": 1,  "ExcludeRegions": [{  "VertexCnt": 4,  "VertexPts": [200, 500, 900, 500, 900, 10000, 0, 10000]  }  ]  }  } | "Enable": Int型,是否使能高抛检测功能.  "FloorNum": Int型,楼层标定中,总楼层数量.  "Floors": Array\_of\_Struct型,楼层的名称和位置. 其中"Line"数组由四个Int组成,分别表示连线的起点(x,y)和终点(x,y). 注意数字为相对坐标,表示视频宽高的万分比.  检测区域的形状不固定,从三角形/四边形到20边形.  "IncludeRegionNum": Int型,检测区域的数量.  "IncludeRegions":检测区域构成的数组. 其中,"VertexCnt"表示该区域的顶点数量. "VertexPts"数组依次表示每个顶点的相对坐标(画面尺寸万分比)的x和y. 所以"VertexPts"数组的长度应该是VertexCnt的两倍.  "ExcludeRegionNum": Int型,检测区域的数量.  "ExcludeRegions":检测区域构成的数组. 其中,"VertexCnt"表示该区域的顶点数量. "VertexPts"数组依次表示每个顶点的相对坐标(画面尺寸万分比)的x和y. 所以"VertexPts"数组的长度应该是VertexCnt的两倍. |
|  |  |

#### 高抛算法配置

|  |  |
| --- | --- |
| 获取高抛检测的算法配置 | |
| POST /digest/frmFallAdvParam HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "ObjDetect": {  "SmoothKernelSizeOpt": [0,5],  "SmoothKernelSize": 3,  "BgPixelDiffOpt": [1,100],  "BgPixelDiff": 16,  "ObjSizeOpt": [4, 1000],  "ObjSizeMin": 4,  "ObjSizeMax": 200,  "ObjCntMaxOpt": [1, 1000],  "ObjCntMax": 500  },  "ObjTack": {  "FrmCntToLostTrackOpt": [0, 10],  "FrmCntToLostTrack": 3,  "TrackExpRatioOpt": [0.0, 1.0],  "TrackExpRatio": 0.05,  "TrackIouOpt": [0.0, 0.5],  "TrackIou": 0.1,  "TrackDistOpt": [16, 64],  "TrackDist": 64,  "TrackDistToSpeedOpt": [1, 16],  "TrackDistToSpeed": 4,  "TrackSizeChangeOpt": [1, 64],  "TrackSizeChange": 16,  },  "LimitControl": {  "UseFgControl": 1,  "ObjCntToPauseOpt": [10, 10000],  "ObjCntToPause": 200,  "ObjAreaToPauseOpt": [0.001, 1.0],  "ObjAreaToPause": 0.1,  "FrmsAfterPauseOpt": [1, 100],  "FrmsAfterPause": 10,  },  "AlarmRules": {  "UseCalibration": 0,  "FrmInTrackOpt": [5, 200],  "FrmMaxInTrack": 50,  "FrmMinInTrack": 10,  "ObjMaxInTrackOpt": [1, 10],  "ObjMaxInTrack": 4,  "HeightMinInMeterOpt": [1, 100],  "HeightMinInMeter": 2,  "HeightMinInRatioOpt": [0.01, 0.99],  "HeightMinInRatio": 0.150000,  "FallAngleMinOpt": [0, 90],  "FallAngleMin": 45,  "MatchMinOfObjOpt": [0.0, 1.0],  "MatchMinOfObj": 0.5,  "FiltByArea": 1,  "AreaVarMaxOpt": [0.1, 1.0],  "AreaVarMax": 0.8,  "FiltByDirect": 1,  "FallAngleVarMaxOpt": [1, 180],  "FallAngleVarMax": 15,  "FallSpeedMinOpt": [0, 1000],  "FallSpeedMin": 4,  "FiltByPos": 1,  "PosDeviaMaxOpt": [0.1, 10],  "PosDeviaMax": 0.8,  "SpeedMinCalcPosOpt": [0, 1000],  "SpeedMinCalcPos": 4,  "ExcludeExtreme": 1,  "FallAccuMinOpt": [1, 10],  "FallAccuMin": 2  }  }  } | "SmoothKernelSizeOpt":Array型,表示"SmoothKernelSize"的取值范围(只读).  "SmoothKernelSize": Int型,图像预处理的平滑窗口大小.  "BgPixelDiffOpt":Array型,表示"BgPixelDiff"的取值范围(只读).  "BgPixelDiff": Int型,与背景差的阈值,单位为灰度级,默认值16. 物体和背景图像灰度的差异,越小越容易检测出物体,但是噪声多.  "ObjSizeOpt":Array型,表示"ObjSizeMin"和"ObjSizeMax"的取值范围(只读).  "ObjSizeMin": Int型,目标的最小尺寸.单位为像素.小于"ObjSizeMax".  "ObjSizeMax": Int型,目标的最大尺寸.单位为像素.大于"ObjSizeMin".  "ObjCntMaxOpt": Array型,表示"ObjCntMax"的取值范围(只读).  "ObjCntMax": Int型,目标的最大数量.  "FrmCntToLostTrackOpt":Array型,表示"FrmCntToLostTrack"的范围(只读).  "FrmCntToLostTrack": Int型,目标轨迹丢失阈值. 目标连续没有检测到次数超过该值,则视为丢失.  "TrackExpRatioOpt": Array型,表示"TrackExpRatio"的范围(只读).  "TrackExpRatio": Float型,目标跟踪匹配的范围(与图像高度的比例).高速下落的目标搜索范围更大.  "TrackIouOpt": Array型,表示"TrackIou"的范围(只读).  "TrackIou": Float型,目标跟踪匹配度阈值. 目标框匹配度超过该值，则认为匹配成功. 完全重叠的两个框,匹配度0.5. 没有重叠区域的两个框,匹配度0.  "TrackDistOpt": Array型,表示"TrackDist"的范围(只读).  "TrackDist": Float型,跟踪框匹配最大距离. 如果前后两帧目标框中心点之间的距离超过此值,则不匹配.  "TrackDistToSpeedOpt": Array型,表示"TrackDistToSpeed"的范围(只读).  "TrackDistToSpeed": Float型,跟踪框匹配最大距离与速度比. 如果前后两帧目标框中心点之间的距离超过目标速度乘以此倍数,则不匹配.  "TrackSizeChangeOpt": Array型,表示"TrackSizeChange"的范围(只读).  "TrackSizeChange":Float型,跟踪框匹配最大面积变化比例. 如果前后两帧目标框的面积比(大比小)超过此值,则不匹配.  "UseFgControl":Int型,是否使能前景超限控制.是否在检测到大量前景目标时暂停分析一段时间. 在发生摄像机晃动/剧烈光照变化等情况时会出现大量前景目标,有可能引起误报.  "ObjCntToPauseOpt":Array型,表示"ObjCntToPause"的范围(只读).  "ObjCntToPause": Int型,暂停分析前景目标数阈值.  "ObjAreaToPauseOpt":Array型,表示"ObjAreaToPause"的范围(只读).  "ObjAreaToPause":Float型,暂停分析前景目标面积占比阈值.  "FrmsAfterPauseOpt":Array型,表示"FrmsAfterPause"的范围(只读).  "FrmsAfterPause":Int型,暂停分析帧数.  "UseFloorCali":Int型,是否使用楼层标定数据.  "FrmInTrackOpt":Array型,表示"FrmMaxInTrack"和"FrmMinInTrack"的范围(只读).  "FrmMaxInTrack":Int型,目标跟踪的最大帧数目.  "FrmMinInTrack":Int型,目标跟踪的最小帧数目.判定高空抛物时至少要分析的帧数.  "ObjMaxInTrackOpt":Array型,表示"ObjMaxInTrack"的范围(只读).  "ObjMaxInTrack":Int型,落物匹配目标数阈值.如果当前帧中有超出此数目的目标匹配到同一条轨迹上,则认为该轨迹不是落物.  "HeightMinInMeterOpt":Array型,表示"HeightMinInMeter"的范围(只读).  "HeightMinInMeter":Float型,目标位移最小值.判定高空抛物时,在场景标定为true的时候,目标下落距离超过该值报警.  "HeightMinInRatioOpt":Array型,表示"HeightMinInRatio"的范围(只读).  "HeightMinInRatio":Float型,目标相对位移最小值.判定高空抛物时,目标下落距离与图像高度的最小比例.  "FallAngleMinOpt":Array型,表示"FallAngleMin"的范围(只读).  "FallAngleMin":Float型,判定高空抛物时,目标下落轨迹与地面夹角的最小值.  "MatchMinOfObjOpt":Array型,表示"MatchMinOfObj"的范围(只读).  "MatchMinOfObj":Float型,目标与轨迹匹配度阈值. 判定高空抛物时,目标与下落轨迹匹配度的最小值.  "FiltByArea":Int型,是否使用面积波动条件.  "AreaVarMaxOpt":Array型,表示"AreaVarMax"的范围(只读).  "AreaVarMax":Float型,目标面积变化最大值. 目标面积变化太大,则视为假目标.面积波动率定义为轨迹上目标框面积的标准差除以面积的平均值.  "FiltByDirect":Int型,是否使用方向波动条件.  "FallAngleVarMaxOpt":Array型,表示"FallAngleVarMax"的范围(只读).  "FallAngleVarMax":Float型,方向波动最大值. 方向波动率定义为前后两帧运动方向夹角的平均值. 当方向波动率超过此值时,则视为假目标.  "FallSpeedMinOpt":Array型,表示"FallSpeedMin"的范围(只读).  "FallSpeedMin": Float型,速度最小值. 如果前后两帧计算的速度小于此值,则不计算方向变化.  "FiltByPos":Int型,是否使用位置波动条件.  "PosDeviaMaxOpt":Array型,表示"PosDeviaMax"的范围(只读).  "PosDeviaMax":Float型,位置波动最大值. 位置波动率定义为位置偏移率的平均值. 位置偏移率定义为当前位置偏离预测位置的距离除以前一帧的速度大小. 当位置波动率超过此值时,认为是假目标.  "SpeedMinCalcPosOpt":Array型,表示"SpeedMinCalcPos"的范围(只读).  "SpeedMinCalcPos":Float型,计算位置偏移率时速度最小值. 如果前一帧速度小于此值,则使用此值作为速度计算位置偏移率.  "ExcludeExtreme":Int型,是否排除极值点.在计算告警过滤统计值时,是否排除轨迹上的极值点:面积最大和最小的目标框, 速度方向变化最大的目标框, 位置偏移率最大的目标框.  "FallAccuMinOpt":Array型,表示"FallAccuMin"的范围(只读).  "FallAccuMin":Int型,满足下落条件累计帧数最小值. 目标轨迹上须至少有此帧数符合下落条件并符合前述过滤条件才会触发告警. |
|  |  |

|  |  |
| --- | --- |
| 设置高抛检测的算法参数 | |
| POST /digest/frmFallAdvParam HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "ObjDetect": {  "SmoothKernelSize": 3,  "BgPixelDiff": 16,  "ObjSizeMin": 4,  "ObjSizeMax": 200,  "ObjCntMax": 500  },  "ObjTack": {  "FrmCntToLostTrack": 3,  "TrackExpRatio": 0.05,  "TrackIou": 0.1,  "TrackDist": 64,  "TrackDistToSpeed": 4,  "TrackSizeChange": 16,  },  "LimitControl": {  "UseFgControl": 1,  "ObjCntToPause": 200,  "ObjAreaToPause": 0.1,  "FrmsAfterPause": 10,  },  "AlarmRules": {  "UseCalibration": 0,  "FrmMaxInTrack": 50,  "FrmMinInTrack": 10,  "ObjMaxInTrack": 4,  "HeightMinInMeter": 2,  "HeightMinInRatio": 0.150000,  "FallAngleMin": 45,  "MatchMinOfObj": 0.5,  "FiltByArea": 1,  "AreaVarMax": 0.8,  "FiltByDirect": 1,  "FallAngleVarMax": 15,  "FallSpeedMin": 4,  "FiltByPos": 1,  "PosDeviaMax": 0.8,  "SpeedMinCalcPos": 4,  "ExcludeExtreme": 1,  "FallAccuMin": 2  }  }  } | "SmoothKernelSize": Int型,图像预处理的平滑窗口大小.  "BgPixelDiff": Int型, 与背景差的阈值,单位为灰度级,默认值16. 物体和背景图像灰度的差异,越小越容易检测出物体,但是噪声多.  "ObjSizeMin": Int型,目标的最小尺寸. 单位为像素. 小于"ObjSizeMax".  "ObjSizeMax": Int型,目标的最大尺寸. 单位为像素. 大于"ObjSizeMin".  "ObjCntMax": Int型,目标的最大数量.  "FrmCntToLostTrack": Int型,目标轨迹丢失阈值. 目标连续没有检测到次数超过该值,则视为丢失.  "TrackExpRatio": Float型,目标跟踪匹配的范围(与图像高度的比例).高速下落的目标搜索范围更大.  "TrackIou": Float型,目标跟踪匹配度阈值. 目标框匹配度超过该值，则认为匹配成功. 完全重叠的两个框,匹配度0.5. 没有重叠区域的两个框,匹配度0.  "TrackDist": Float型, 跟踪框匹配最大距离. 如果前后两帧目标框中心点之间的距离超过此值,则不匹配.  "TrackDistToSpeed": Float型,跟踪框匹配最大距离与速度比. 如果前后两帧目标框中心点之间的距离超过目标速度乘以此倍数,则不匹配.  "TrackSizeChange": Float型,跟踪框匹配最大面积变化比例. 如果前后两帧目标框的面积比(大比小)超过此值,则不匹配.  "UseFgControl":Int型, 是否使能前景超限控制. 是否在检测到大量前景目标时暂停分析一段时间. 在发生摄像机晃动/剧烈光照变化等情况时会出现大量前景目标,有可能引起误报.  "ObjCntToPause": Int型, 暂停分析前景目标数阈值.  "ObjAreaToPause":Float型,暂停分析前景目标面积占比阈值.  "FrmsAfterPause": Int型,暂停分析帧数.  "UseFloorCali":Int型,是否使用楼层标定数据.  "FrmMaxInTrack":Int型,目标跟踪的最大帧数目.  "FrmMinInTrack": Int型,目标跟踪的最小帧数目. 判定高空抛物时至少要分析的帧数.  "ObjMaxInTrack":Int型,落物匹配目标数阈值. 如果当前帧中有超出此数目的目标匹配到同一条轨迹上,则认为该轨迹不是落物.  "HeightMinInMeter":Float型,目标位移最小值. 判定高空抛物时,在场景标定为true的时候,目标下落距离超过该值报警.  "HeightMinInRatio":Float型,目标相对位移最小值.判定高空抛物时,目标下落距离与图像高度的最小比例.  "FallAngleMin":Float型,判定高空抛物时,目标下落轨迹与地面夹角的最小值.  "MatchMinOfObj":Float型,目标与轨迹匹配度阈值. 判定高空抛物时,目标与下落轨迹匹配度的最小值.  "FiltByArea":Int型,是否使用面积波动条件.  "AreaVarMax":Float型,目标面积变化最大值. 目标面积变化太大,则视为假目标. 面积波动率定义为轨迹上目标框面积的标准差除以面积的平均值.  "FiltByDirect":Int型,是否使用方向波动条件.  "FallAngleVarMax":Float型,方向波动最大值. 方向波动率定义为前后两帧运动方向夹角的平均值. 当方向波动率超过此值时,则视为假目标.  "FallSpeedMin": Float型,速度最小值. 如果前后两帧计算的速度小于此值,则不计算方向变化.  "FiltByPos":Int型, 是否使用位置波动条件.  "PosDeviaMax":Float型,位置波动最大值. 位置波动率定义为位置偏移率的平均值. 位置偏移率定义为当前位置偏离预测位置的距离除以前一帧的速度大小. 当位置波动率超过此值时,认为是假目标.  "SpeedMinCalcPos":Float型,计算位置偏移率时速度最小值. 如果前一帧速度小于此值,则使用此值作为速度计算位置偏移率.  "ExcludeExtreme":Int型,是否排除极值点. 在计算告警过滤统计值时,是否排除轨迹上的极值点: 面积最大和最小的目标框, 速度方向变化最大的目标框, 位置偏移率最大的目标框.  "FallAccuMin":Int型,满足下落条件累计帧数最小值. 目标轨迹上须至少有此帧数符合下落条件并符合前述过滤条件才会触发告警. |
|  |  |

### 电梯状态检测

|  |  |
| --- | --- |
| 获取电梯状态检测配置 | |
| POST /digest/frmElevatorStatusDetect HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "EnableTrappedDetect": 0,  "TrappedDetectTime": 90,  "EnableNoLevelParking": 0  }  } | "EnableTrappedDetect": Int型,是否启用困人检测,1-启用,0-不启用  "TrappedDetectTime": Int型,困人检测时间,单位:秒  "EnableNoLevelParking": Int型,是否启用非平层停靠检测,1-启用,0-不启用 |
|  |  |

|  |  |
| --- | --- |
| 设置电梯状态检测配置 | |
| POST /digest/frmElevatorStatusDetect HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "EnableTrappedDetect": 0,  "TrappedDetectTime": 90,  "EnableNoLevelParking": 0  }  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

## 录像

### 录像计划配置

|  |  |
| --- | --- |
| 获取录像计划配置 | |
| POST /digest/frmVideoPlanPara HTTP/1.1 | |
| 请求的正文内容  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容  {  "Result": 0,  "Data": {  "PreRecordTime": 1,  "RecordTime": 2,  "RecordSched": [  [  [  0,  0,  23,  59  ],  ……  ],  ……  ],  "RecordMode": 1  }  } | "PreRecordTime": Int型,预录时间  0:不预录  1:5秒  2:10秒  "RecordTime": Int型,录像延时  0:5秒  1:20秒  2:30秒  3:1分  4:2分  5:5分  6:10分  "RecordSched": 录像时间段,7x8格式  "RecordMode": Int型,录像模式,0-不录像 1-定时录像 2-报警录像 3-定时+报警 |

|  |  |
| --- | --- |
| 设置录像计划配置 | |
| POST /digest/frmVideoPlanPara HTTP/1.1 | |
| 请求的正文内容  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "PreRecordTime": 1,  "RecordTime": 2,  "RecordSched": [  [  [  0,  0,  23,  59  ],  ……  ],  ……  ],  "RecordMode": 1  }  } |  |
| 响应的正文内容  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

### 视频断开上报

|  |  |
| --- | --- |
| 获取视频断开上报配置 | |
| POST /digest/frmMediaDisconnect HTTP/1.1 | |
| 请求的正文内容  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容  {  "Result": 0,  "Data": {  "DisconnectIP": "0.0.0.0",  "DeviceId": 0,  "ChannelId": 0,  "StreamId": 0  }  } | "DisconnectIP": String型,视频Edge IP地址  "DeviceId": Int型,设备号,从0开始,0-设备1  "ChannelId": Int型,通道号,从0开始,0-通道1  "StreamId": Int型,录像模式,0-主码流 1-子码流3-第三码流 |

|  |  |
| --- | --- |
| 设置视频断开上报配置 | |
| POST /digest/frmMediaDisconnect HTTP/1.1 | |
| 请求的正文内容  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "DisconnectIP": "0.0.0.0",  "DeviceId": 0,  "ChannelId": 0,  "StreamId": 0  }  } |  |
| 响应的正文内容  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

### 水印配置

|  |  |
| --- | --- |
| 获取水印配置 | |
| POST /digest/frmWaterMark HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "Enable": 0,  "MarkText": ""  }  } | "Enable": Int型,是否启用水印功能,1-启用,0-不启用  "MarkText": String型,水印字符串 |

|  |  |
| --- | --- |
| 设置水印配置 | |
| POST /digest/frmWaterMark HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "Enable": 0,  "MarkText": ""  }  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

### 录像查询

|  |  |
| --- | --- |
| 录像查询 | |
| POST /digest/frmVideoRecordsQuery HTTP/1.1 | |
| 请求的正文内容  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {  "Channels": [  1  ],  "DeviceLocalDateTimeStart": "20200408000000",  "DeviceLocalDateTimeStop": "20200408235959",  "QueryType": 255  }  } | "Channels": Int型数组,通道列表  "DeviceLocalDateTimeStart": String型,开始时间,本地时间,格式:年月日时分秒  "DeviceLocalDateTimeStop": String型,结束时间,本地时间, 格式:年月日时分秒  "QueryType": Int型,录像类型,255-全部 |
| 响应的正文内容  {  "Result": 0,  "Data": {  "SearchResults": [  {  "Dev": 1,  "ChannelNo": 1,  "Records": [  [  1,  1586326769,  1586337316,  0,  -480  ]  ]  }  ]  }  } | "Dev": Int型,设备号  "ChannelNo": Int型,通道号  "Records"[0]: Int型,录像类型,1:定时录像,其他:报警录像  "Records"[1]: Int型,录像开始时间,UTC时间  "Records"[2]: Int型,录像结束时间,UTC时间  "Records"[3]: Int型,未使用  "Records"[4]: Int型,录像夏令时和时区偏移的总时间,单位:分 |

### 按月查询录像

|  |  |
| --- | --- |
| 按月查询录像 | |
| POST /digest/frmVideoQueryByMonthHTTP/1.1 | |
| 请求的正文内容  {  "Type": 0,  "Ch": 1,  "Data": {  "StartYear": 2019,  "StartMonth": 12  }  } | "StartYear": Int型,要查询的年份  "StartMonth": Int型,要查询的月份 |
| 响应的正文内容  {  "Result": 0,  "Data": {  "QueryResult": 3072  }  } | "QueryResult": Int型,查询结果，查询结果为32bit，第0位代表1号，第1位代表2号，依此类推，第30位表示31号，若当天有录像，对应的位置为1，若没有，为0 |

### SD卡联动抓图查询

|  |  |
| --- | --- |
| SD卡联动抓图查询 | |
| POST /digest/frmSearchSDCardPicsHTTP/1.1 | |
| 请求的正文内容  {  "Type": 0,  "Ch": 1,  "Data": {  "StartTime": 1575216000,  "EndTime": 1575302399  }  } | "StartTime": Int型,开始时间,UTC时间  "EndTime": Int型,结束时间,UTC时间 |
| 响应的正文内容  {  "Result": 0,  "Data": {  "SearchResults": [  "20191202/20191202-182032\_32\_0\_0.jpeg",  "20191202/20191202-182035\_32\_0\_0.jpeg",  "20191202/20191202-235927\_32\_0\_0.jpeg"  ]  }  } | "SearchResults": String型数组,查询到的抓图文件名集合,  图片完整地址:<http://ip/snapshot/文件名> |

### 磁盘信息

|  |  |
| --- | --- |
| 磁盘信息获取 | |
| POST /digest/frmGetHDInfo HTTP/1.1 | |
| 请求的正文内容  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容  {  "Result": 0,  "Data": {  "HDInfoList": [  {  "HDNo": 1,  "HdStatus": 0,  "Capacity": 55568,  "FreeSpace": 406,  "HDType": 0  }  ]  }  } | "HDNo": Int型,磁盘编号  "HdStatus": Int型,磁盘状态,bit0:0-正常,1-异常,bit1:1-当前写的盘  "Capacity": Int型,磁盘容量  "FreeSpace": Int型,磁盘剩余空间  "HDType": Int型,磁盘类型,0-本地磁盘 |

### 磁盘强制格式化检查

|  |  |
| --- | --- |
| 磁盘强制格式化检查 | |
| POST /digest/frmGetHDFSInfo HTTP/1.1 | |
| 请求的正文内容  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容  {  "Result": 0,  "Data": {  "FsSupport": "RAW",  "DiskFs": "MKV",  "NeedFormat": 1  }  } | "FsSupport": String型,支持格式  "DiskFs": String型,磁盘格式  "NeedFormat": Int型,是否需要强制格式化,1-需要,2-不需要 |

### 格式化磁盘

|  |  |
| --- | --- |
| 格式化磁盘 | |
| POST /digest/frmHDFormat HTTP/1.1 | |
| 请求的正文内容  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {  "FormatList": [  1  ]  }  } | "FormatList":Int型数组,要格式化的磁盘编号列表 |
| 响应的正文内容  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

### 获取磁盘格式化进度

|  |  |
| --- | --- |
| 格式化磁盘 | |
| POST /digest/frmGetHDFormatProgress HTTP/1.1 | |
| 请求的正文内容  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容  {  "Result": 0,  "Data": {  "ProgressList": [  {  "DiskNo": 1,  "DiskProgress": 50  }  ]  }  } | "DiskNo":Int型,正在格式化的磁盘序号  "DiskProgress": Int型,格式化进度 |

## 智能人脸

### 人脸数据库操作

#### 添加人脸数据库

|  |  |
| --- | --- |
| 添加人脸数据库 | |
| POST /digest/frmFaceDatabase HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {  "Enable": 1,  "TargetType": 1,  "Similarity": 75,  "TargetName": "zc"  }  } | "Enable": Int型,是否启用参与比对,1-启用,0-不启用  "TargetType": Int型,数据库类型,1表示白名单,2表示黑名单,  "Similarity": Int型,相似度  "TargetName": String型,数据库名称 |
| 响应的正文内容：  {  "Result" : 0,  "Data" : {  "StatusCode" : "Operation Ok"  }  } |  |

#### 删除人脸数据库

|  |  |
| --- | --- |
| 删除人脸数据库 | |
| POST /digest/frmFaceDatabase HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "TargetName": "zc"  }  } | "TargetName": String型,要删除的数据库的名称. |
| 响应的正文内容：  {  "Result" : 0,  "Data" : {  "StatusCode" : "Operation Ok"  }  } |  |

#### 修改人脸数据库

|  |  |
| --- | --- |
| 主动配置**（**在主动型服务器下，由服务器不断询问设备配置结果，反复发送以下请求，直到配置完成。**）** | |
| POST /digest/frmFaceDatabase HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 2,  "Dev": 1,  "Ch": 1,  "Data": {  "Enable": 1,  "TargetType": 1,  "Similarity": 75,  "TargetName": "zc",  "NewTargetName": "zcx"  }  } | "Enable": Int型,是否启用参与比对,1-启用,0-不启用  "TargetType": Int型,数据库类型,1表示白名单,2表示黑名单,  "Similarity": Int型,相似度,范围0-100  "TargetName": String型,当前的数据库的名称  "NewTargetName": String型,修改后的数据库的名称 |
| 响应的正文内容：  {  "Result" : 0,  "Data" : {  "StatusCode" : "Operation Ok"  }  } |  |

#### 清空人脸数据库

|  |  |
| --- | --- |
| 清空人脸数据库 | |
| POST /digest/frmFaceDatabase HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 3,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容：  {  "Result" : 0,  "Data" : {  "StatusCode" : "Operation Ok"  }  } |  |

#### 获取人脸数据库数目

|  |  |
| --- | --- |
| 获取人脸数据库数目 | |
| POST /digest/frmFaceDatabase HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 4,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "TargetItemTotalNum": 2  }  } | "TargetItemTotalNum":Int型,人脸数据库个数 |

#### 获取人脸数据库

|  |  |
| --- | --- |
| 获取人脸数据库 | |
| POST /digest/frmFaceDatabase HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 5,  "Dev": 1,  "Ch": 1,  "Data": {  "Offset": 0,  "Num": 10  }  } | "Offset": Int型,查询结果的偏移量  "Num": Int型,查询数量 |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "TargetResults": {  "TargetItems": [  {  "TargetName": "zc",  "GroupId": 16,  "Enable": 1,  "TargetType": 1,  "Similarity": 75,  "Count": 1,  "TargetVersion": -2  }  ]  }  }  } | "TargetItems":Object型数组,查询到的数据库列表  "TargetName": String型,数据库名称  "GroupId": Int型,人脸库ID  "Enable": Int型,是否启用,0-不启用,1-启用  "TargetType": Int型,数据库类型,1-白名单,2-黑名单  "Similarity": Int型,相似度  "Count": Int型,当前数据库里的人脸模板数量  "TargetVersion": Int型,人脸库版本号 |

#### 获取人脸库绑定信息

|  |  |
| --- | --- |
| 获取人脸数据库绑定信息 | |
| POST /digest/frmFaceDatabase HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 7,  "Dev": 1,  "Ch": 1,  "Data": {  "TargetName": "zc"  }  } | "TargetName": String型,数据库名称 |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "TargetName": "zc",  "LinkChList": [  {  "Ch": 1,  "Similarity": 75  }  ]  }  } | "TargetName": String型,数据库名称  "Ch": Int型,通道号  "Similarity": Int型,相似度 |

#### 设置人脸库绑定信息

|  |  |
| --- | --- |
| 获取人脸数据库 | |
| POST /digest/frmFaceDatabase HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 6,  "Dev": 1,  "Ch": 1,  "Data": {  "TargetName": "zc",  "LinkChList": [  {  "Ch": 1,  "Similarity": 76  }  ]  }  } | "TargetName": String型,数据库名称  "Ch": Int型,通道号  "Similarity": Int型,相似度 |
| 响应的正文内容：  {  "Result" : 0,  "Data" : {  "StatusCode" : "Operation Ok"  }  } |  |

#### 设置人脸库版本号

|  |  |
| --- | --- |
| 获取人脸数据库 | |
| POST /digest/frmFaceDatabase HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 8,  "Dev": 1,  "Ch": 1,  "Data": {  "TargetName": "zc",  "TargetVersion": 11  }  } | "TargetName": String型,数据库名称  "TargetVersion": Int型,人脸库版本号 |
| 响应的正文内容：  {  "Result" : 0,  "Data" : {  "StatusCode" : "Operation Ok"  }  } |  |

### 人脸模板操作

#### 人脸模板添加

|  |  |
| --- | --- |
| 添加人脸模板 | |
| POST /digest/frmFacePicture HTTP/1.1 | |
| 请求的正文内容  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {  "TargetName": "zc",  "List": [{  "Name": "419",  "Describe": "XWJ",  "FaceData": "xxx",  "WiegandId": 1  }]  }  } | "TargetName", String型, 操作的人脸库名称  "List":Object型数组,要录入的人脸信息  "Name", String型, 人脸名称  "Describe", String型, 人脸描述  "FaceId", String型, 人脸ID,为空或不带此字段时则自动生成  "FaceData", String型, 人脸图片数据base64编码，目前仅支持jpeg  "WiegandId", Int型, 韦根号 |
| 响应的正文内容  {  "Result": 0,  "Data": {  "ResultList": [  {  "Result": 0,  "StatusCode": "Operation Ok",  "FaceId": "5c4242c9-1512-41fa-9509-057ff692b1a3"  }  ]  }  } | "FaceId":String型,录入的人脸图的人脸ID |

#### 人脸模板删除

|  |  |
| --- | --- |
| 删除人脸模板 | |
| POST /digest/frmFacePicture HTTP/1.1 | |
| 请求的正文内容  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "FaceId": "bf6183ff-8988-4d02-92f3-5b3fd0cad5db"  }  } | "FaceId",String型, 要删除的人脸模板的ID |
| 响应的正文内容  {  "Result" : 0,  "Data" : {  "StatusCode" : "Operation Ok"  }  } |  |

#### 人脸模板修改

|  |  |
| --- | --- |
| 修改人脸模板信息 | |
| POST /digest/frmFacePictureHTTP/1.1 | |
| 请求的正文内容  {  "Type": 4,  "Dev": 1,  "Ch": 1,  "Data": {  "TargetName": "zc",  "List": [  {  "Name": "4191",  "Describe": "XWJ",  "WiegandId": 1,  "SrcFaceId": "bf6183ff-8988-4d02-92f3-5b3fd0cad5db",  "NeedModifyFacePic": 1,  "FaceData": "xxx"  }  ]  }  } | "TargetName", String型, 操作的人脸分组名字（每次操作都只能对一个组操作）  "SrcFaceId", String型, 要修改的人脸ID  "NeedModifyFacePic": Int型,是否修改人脸图,0-不修改,1-修改  "FaceData": String型,人脸图的base64数据  "Describe": String型, 要修改的人脸描述  "WiegandId": Int型,要修改的韦根号  "Name": String型, 要修改的人脸名 |
| 响应的正文内容(示例)  {  "Result": 0,  "Data": {  "ResultList": [  {  "Result": 0,  "StatusCode": "Operation Ok",  "FaceId": "bf6183ff-8988-4d02-92f3-5b3fd0cad5db"  }  ]  }  } |  |

#### 人脸模板查询

|  |  |
| --- | --- |
| 查询符合条件的人脸数目 | |
| POST /digest/frmFacePicture HTTP/1.1 | |
| 请求的正文内容  {  "Type": 2,  "Dev": 1,  "Ch": 1,  "Data": {  "MatchType": 0,  "ParamMask": 1,  "TargetName": "zc",  "Name": "",  "Describe": "",  "FaceId": "",  "FaceData": "",  "Similarity": 0  }  } | "MatchType", Int型, 0:全词匹配，1模糊匹配  "TargetName", String型, 操作的人脸库名字  "ParamMask", Int型,参与查询的有效项,按位操作:  bit 0: TargetName有效；  bit 1: Name有效；  bit 2: Describe有效；  bit 3: FaceId有效；  bit 4: FaceData有效；  bit 5: Similarity有效；  "Name", String型,人脸名  "Describe", String型,人脸描述  "FaceId", String型,人脸ID  "FaceData": String型,以图搜图的图片base64数据  "Similarity": Int型,相似度 |
| 响应的正文内容  {  "Result": 0,  "Data": {  "FaceResults": 2  }  } | "FaceResults":Int型,查询到的人脸模板数 |

|  |  |
| --- | --- |
| 查询符合条件的人脸模板信息 | |
| POST /digest/frmFacePicture HTTP/1.1 | |
| 请求的正文内容  {  "Type": 3,  "Dev": 1,  "Ch": 1,  "Data": {  "MatchType": 0,  "Offset": 0,  "Num": 10,  "ParamMask": 1,  "TargetName": "zc",  "Name": "",  "Describe": "",  "FaceId": "",  "FaceData": "",  "Similarity": 0  }  } | "MatchType", Int型, 0:全词匹配，1模糊匹配  "Offset": Int型,查询结果的偏移量  "Num": Int型,查询数量  "TargetName", String型, 操作的人脸库名字  "ParamMask", Int型,参与查询的有效项,按位操作:  bit 0: TargetName有效；  bit 1: Name有效；  bit 2: Describe有效；  bit 3: FaceId有效；  bit 4: FaceData有效；  bit 5: Similarity有效；  "Name", String型,人脸名  "Describe", String型,人脸描述  "FaceId", String型,人脸ID  "FaceData": String型,以图搜图的图片base64数据  "Similarity": Int型,相似度 |
| 响应的正文内容  {  "Result": 0,  "Data": {  "FaceResults": [  {  "FaceId": "5c4242c9-1512-41fa-9509-057ff692b1a3",  "Name": "419",  "Describe": "XWJ",  "TargetName": "zc",  "ModelVersion": "",  "GroupId": 16,  "GroupType": 0,  "WiegandId": 1,  "Version": 2147483647,  "CreateTime": "1970-01-01 08:00:00",  "UpdateTime": "1970-01-01 08:00:00",  "FaceData": "xxx",  "DataType": 0  }  ],  "TotalCount": 2  }  } | "Name", String型,人脸名  "Describe", String型,人脸描述  "FaceId", String型,人脸ID  "FaceData": String型,人脸模板图片base64数据  "TargetName", String型, 操作的人脸库名字  "ModelVersion":String型,保留字段  "GroupId": Int型,人脸库ID  "GroupType": Int型,数据库类型,1-白名单,2-黑名单  "WiegandId": Int型,韦根号  "Version": Int型,人脸库版本  "CreateTime": String型,此条数据创建时间  "UpdateTime": String型,此条数据最后更新时间  "DataType": Int型,数据库类型,1-白名单,2-黑名单  "TotalCount":Int型,查询结果总数 |

|  |  |
| --- | --- |
| 查询符合条件的人脸模板信息(简化接口) | |
| POST /digest/frmFacePicture HTTP/1.1 | |
| 请求的正文内容  {  "Type": 5,  "Dev": 1,  "Ch": 1,  "Data": {  "MatchType": 0,  "Offset": 0,  "Num": 10,  "ParamMask": 1,  "TargetName": "zc",  "Name": "",  "Describe": "",  "FaceId": "",  "FaceData": "",  "Similarity": 0  }  } | "MatchType", Int型, 0:全词匹配，1模糊匹配  "Offset": Int型,查询结果的偏移量  "Num": Int型,查询数量  "TargetName", String型, 操作的人脸库名字  "ParamMask", Int型,参与查询的有效项,按位操作:  bit 0: TargetName有效；  bit 1: Name有效；  bit 2: Describe有效；  bit 3: FaceId有效；  bit 4: FaceData有效；  bit 5: Similarity有效；  "Name", String型,人脸名  "Describe", String型,人脸描述  "FaceId", String型,人脸ID  "FaceData": String型,以图搜图的图片base64数据  "Similarity": Int型,相似度 |
| 响应的正文内容  {  "Result": 0,  "Data": {  "FaceResults": [  {  "FaceId": "852cf26a-350b-40b4-b3f4-2737b2c0f2e9",  "Version": 2147483647  },  {  "FaceId": "5c4242c9-1512-41fa-9509-057ff692b1a3",  "Version": 2147483647  }  ],  "TotalCount": 2  }  } | "Name", String型,人脸名  "FaceId", String型,人脸ID  "Version": Int型,人脸库版本 |

### 人脸比对查询

#### 获取查询句柄

|  |  |
| --- | --- |
| 获取查询句柄 | |
| POST /digest/frmFaceCompare HTTP/1.1 | |
| 请求的正文内容 | "BeginDateTime": String型,开始时间,格式:"Y-M-D h:m:s"  "EndDateTime": String型,结束时间,格式:"Y-M-D h:m:s"  "QueryMode": Int型,查询模式,0-全部,1-识别成功,2-识别失败  "FaceID": String型,人脸ID  "FaceData": String型,人脸图base64数据  "Similarity": Int型,相似度阈值  "SearchType": Int型,数据库类型,按位操作:bit1-白名单,bit2-黑名单  "TemperatureMin": Int型,温度值,最小温度  "TemperatureMax": Int型, 温度值,最大温度  "MaskType": Int型,口罩检测状态, 0-不检测,1-未佩戴,2-已佩戴 |
| 1.识别结果查询  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {  "BeginDateTime": "2019-12-2 00:00:00",  "EndDateTime": "2019-12-2 16:37:24",  "QueryMode": 0  }  } |
| 1. 人脸ID查询   {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "BeginDateTime": "2019-12-2 00:00:00",  "EndDateTime": "2019-12-2 16:37:24",  "FaceID": "asd",  "Similarity": 75  }  } |
| 1. 以图搜图   {  "Type": 2,  "Dev": 1,  "Ch": 1,  "Data": {  "BeginDateTime": "2019-12-2 00:00:00",  "EndDateTime": "2019-12-2 16:37:24",  "FaceData": "xxx",  "Similarity": 75  }  } |
| 4.数据库类型查询  {  "Type": 3,  "Dev": 1,  "Ch": 1,  "Data": {  "BeginDateTime": "2019-12-2 00:00:00",  "EndDateTime": "2019-12-2 16:37:24",  "SearchType": 2  }  } |
| 5.按温度、口罩状态查询  {  "Type":9,  "Dev":1,  "Ch":1,  "Data":{  "BeginDateTime":"2020-7-6 00:00:00",  "EndDateTime":"2020-7-6 23:59:59",  "QueryMode":0,  "SearchType":0,  "TemperatureMin":30,  "TemperatureMax":40,  "MaskType":0  }  } |
| 响应的正文内容  {  "Result": 0,  "Data": {  "ResultHandle": 983224  }  } | "ResultHandle":Int型,查询句柄 |

#### 获取查询进度

|  |  |
| --- | --- |
| 获取查询进度 | |
| POST /digest/frmFaceCompare HTTP/1.1 | |
| 请求的正文内容  {  "Type": 4,  "Dev": 1,  "Ch": 1,  "Data": {  "ResultHandle": 983224  }  } |  |
| 响应的正文内容  {  "Result": 0,  "Data": {  "TotalSearchNum": 0,  "CurrentSearchNum": 0,  "MatchNum": 0,  "StatusCode": 0  }  } | "TotalSearchNum": Int型,搜索总数  "CurrentSearchNum": Int型,当前已搜索个数  "MatchNum": Int型,匹配上的个数  "StatusCode": Int型,搜索状态,0-正常,1查询中 |

#### 获取查询结果

|  |  |
| --- | --- |
| 获取查询结果 | |
| POST /digest/frmFaceCompare HTTP/1.1 | |
| 请求的正文内容  {  "Type": 5,  "Dev": 1,  "Ch": 1,  "Data": {  "ResultHandle": 983224,  "Offset": 0,  "Num": 10  }  } | "Offset": Int型,查询结果的偏移量  "Num": Int型,查询数量  "ResultHandle":Int型,查询句柄 |
| 响应的正文内容  {  "Result": 0,  "Data": {  "FaceResults": [  {  "Similarity": 0,  "Time": "2019-12-10 17:50:12",  "DataType": 0,  "FaceData": "xxx",  "Target": 0,  "TargetName": "",  "Name": "",  "Describe": "",  "FaceId": "",  "TargetSimilarity": 0,  "TargetDataType": 0,  "TargetType": 0  "TargetFaceData": "xxx"  }  ]  }  } | "Similarity": Int型,相似度  "Time": String型,比对时间  "DataType": Int型,抓拍图片数据类型,0-base64数据,1-二进制数据  "FaceData": String型,抓拍图  "Target": Int型,是否存在目标模板,0-无模板信息，1-有模板信息  "TargetName", String型,目标模板对应的数据库名  "Name", String型,模板名  "Describe", String型,模板描述  "FaceId", String型,模板人脸ID  "TargetSimilarity": Int型,与模板的相似度,  "TargetDataType": Int型,模板图片数据类型,0-base64数据,1-二进制数据,  "TargetType": Int型,数据库类型,1表示白名单,2表示黑名单  "TargetFaceData":String型,模板图 |

#### 查询句柄释放

|  |  |
| --- | --- |
| 获取查询结果 | |
| POST /digest/frmFaceCompare HTTP/1.1 | |
| 请求的正文内容  {  "Type": 8,  "Dev": 1,  "Ch": 1,  "Data": {  "ResultHandle": 983224  }  } | "ResultHandle":Int型,查询句柄 |
| 响应的正文内容  {  "Result" : 0,  "Data" : {  "StatusCode" : "Operation Ok"  }  } |  |

### 人脸库算法版本获取

|  |  |
| --- | --- |
| 获取人脸库算法版本 | |
| POST /digest/frmFaceGetVersion HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "FaceDetect": "v4.0.0",  "FaceRec": "v2.0\_18\_11\_05"  }  } | "FaceDetect":String型,人脸检测库版本  "FaceRec"::String型,人脸识别库版本 |

### 人脸高级配置

|  |  |
| --- | --- |
| 获取人脸高级配置 | |
| POST /digest/frmFaceRecognizeCfg HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {  "GroupType": 0,  "AlarmTimeType": 1  }  } | "GroupType":Int型,0-白名单联动,1-黑名单联动,2-识别失败联动  "AlarmTimeType":Int型,时间段类型,0-每天一个时间段,1-每天八个时间段 |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "AlarmTime": [  [  [  0,  0,  23,  59  ],  ……  ],  ……  ],  "EnableHandle": 0,  "LinkCfg": {  "HandleType": 0,  "SnapCount": 1,  "SnapInterval": 1,  "EnablePreset": 0,  "PresetNo": 0,  "EnableAudio": 0,  "AudioNo": 1,  "AlarmOut": 0,  "Wiegand": []  }  }  } | "AlarmTime":Int型数组,报警时间段  "EnableHandle":Int型,联动总开关,0-不启用,1-启用  "HandleType":Int型,  "SnapCount":Int型,联动抓图  "SnapInterval":Int型,抓图间隔时间  "EnablePreset":Int型,是否联动预置点,0-不联动,1-联动  "PresetNo":Int型,联动的预置点号  "EnableAudio":Int型,是否联动音频,0-不联动,1-联动  "AudioNo":Int型,联动的音频编号  "AlarmOut":Int型,联动报警输出  "Wiegand": 联动韦根 |

|  |  |
| --- | --- |
| 设置人脸高级配置 | |
| POST /digest/frmFaceRecognizeCfg HTTP/1.1 | |
| 请求的正文内容：  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "GroupType": 0,  "AlarmTimeType": 1  "AlarmTime": [  [  [  0,  0,  23,  59  ],  ……  ],  ……  ],  "EnableHandle": 0,  "LinkCfg": {  "HandleType": 0,  "SnapCount": 1,  "SnapInterval": 1,  "EnablePreset": 0,  "PresetNo": 0,  "EnableAudio": 0,  "AudioNo": 1,  "AlarmOut": 0,  "Wiegand": []  }  }  } |  |
| 响应的正文内容：  {  "Result" : 0,  "Data" : {  "StatusCode" : "Operation Ok"  }  } |  |

### 人脸操作状态码

|  |  |
| --- | --- |
| 状态码 | 请求结果 |
| 0  -1  -2  -3  -4  -5  -6  -7  -8  ~~-9~~  -10  -11  -12  -13  -14  -15  -16  -17  -18  -19  -20  -21  -22  -23  -24 | 0 成功  -1 失败  -2 未初始化  -3 查询还未开始  -4 查询在忙状态  -5 参数无效  -6 句柄值无效  -7 没有内存  -8 不支持  ~~-9 …(无效状态码)~~  -10 最大预设人脸模板数  -11 提取特征值失败  -12 创建人脸基库失败  -13 创建人脸组库失败  -14 创建人脸比对历史库失败  -15 获取人脸模板总数失败  -16 获取人脸比对历史总数失败  -17 人脸组已存在  -18 人脸组不存在  -19 缓冲目录未准备好  -20 缓冲文件保存失败  -21 修改人脸模板时组名不存在  -22 添加人脸模板磁盘空间不够  -23 不支持人脸比对记录  -24 导入模板时检测到多个人脸 |

## 智能结果统计

### 获取查询句柄

|  |  |
| --- | --- |
| 获取查询句柄 | |
| POST /digest/frmPersonSmartRecords HTTP/1.1 | |
| 请求的正文内容  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {  "TableName": "Counter",  "QueryType": 1,  "TimeType": 0,  "RegionIdx": 0,  "StartTime": 1575216000,  "StopTime": 1575216000  }  } | "TableName": String型,报表名,Counter:计数报表,Region:电子围栏报表,Absent:离岗报表,Retrograde:逆行报表  "QueryType": Int型,查询方式,  0:默认  1:按月（底层会忽略StopTime，查询StartTime当月的天数个结果）  2:按日（底层会忽略StopTime，查询StartTime当日的24个结果）  "TimeType": Int型,时间格式  0：int型utc时间  1：字符串本地时间，格式（年月日空格时分秒，例：20190805 143030）  "RegionIdx": Int型,区域号,从0开始  "StartTime":开始时间,根据TimeType,是数字或字符串  "StopTime":结束时间,根据TimeType,是数字或字符串 |
| 响应的正文内容  {  "Result": 0,  "Data": {  "SessionHandle": 2,  "ResultCount": 24,  "SumEnterCnt": 342,  "SumLeaveCnt": 329  }  } | "SessionHandle": Int型,查询句柄  "ResultCount": Int型,结果个数  "SumEnterCnt": Int型,进入总数,计数报表有效,其他报表时与ResultCount一致  "SumLeaveCnt": Int型,离开总数,计数报表有效,其他报表时与ResultCount一致 |

### 获取查询结果

|  |  |
| --- | --- |
| 获取查询结果 | |
| POST /digest/frmPersonSmartRecords HTTP/1.1 | |
| 请求的正文内容  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "SessionHandle": 1,  "StartCount": 1,  "StopCount": 20,  "TimeType": 0  }  } | "SessionHandle": Int型,查询句柄  "TimeType": Int型,时间格式  0：int型utc时间  1：字符串本地时间，格式（年月日空格时分秒，例：20190805 143030）  "StartCount": Int型,查询起始位置  "StopCount": Int型,查询结束位置 |
| 响应的正文内容 | "TableName": String型,表名  "TotalCnt": Int型,结果总个数  "RegionIdx": Int型,区域号,从0开始  "LeaveCnt": Int型,离开个数  "EnterCnt": Int型,进入个数  "StartTime": 开始时间,根据TimeType,是数字或字符串  "StopTime": 结束时间,根据TimeType,是数字或字符串  "Time": 报警时间,根据TimeType,是数字或字符串  "SubType": Int型,报警子类型,保留字段  "IsHappen": Int型,报警是否发生,目前默认为1 |
| 1.计数报表:  {  "Result": 0,  "Data": {  "ResultList": [  {  "RegionIdx": 0,  "LeaveCnt": 605,  "EnterCnt": 378,  "StartTime": 1575993600,  "StopTime": 1575997199  },  ……  ],  "TableName": "Counter",  "TotalCnt": 24  }  } |
| 2.其他报表:  {  "Result": 0,  "Data": {  "ResultList": [  {  "RegionIdx": 0,  "Time": 1576033714,  "SubType": 0,  "IsHappen": 1  },  ……  ],  "TableName": "Retrograde",  "TotalCnt": 1240  }  } |

### 释放查询句柄

|  |  |
| --- | --- |
| 释放查询句柄 | |
| POST /digest/frmPersonSmartRecords HTTP/1.1 | |
| 请求的正文内容  {  "Type": 2,  "Dev": 1,  "Ch": 1,  "Data": {  "SessionHandle": 1  }  } | "SessionHandle": Int型,查询句柄 |
| 响应的正文内容  {  "Result" : 0,  "Data" : {  "StatusCode" : "Operation Ok"  }  } |  |

## 光学

### 自动镜头校正

|  |  |
| --- | --- |
| 自动镜头校正 | |
| POST /digest/frmAutoLensCorrection HTTP/1.1 | |
| 请求的正文内容  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容  {  "Result" : 0,  "Data" : {  "StatusCode" : "Operation Ok"  }  } |  |

### 自定义镜头参数

|  |  |
| --- | --- |
| 获取自定义镜头参数 | |
| POST /digest/frmLensCustomCfg HTTP/1.1 | |
| 请求的正文内容  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容  {  "Result": 0,  "Data": {  "Enable": 1,  "CurrentSelect": 2,  "List": [  {  "LensType": "YS\_15\_3\_XM02",  "BoardLine": "101",  "LdcParamPath": ""  },  {  "BoardLine": "101",  "LensType": "YT\_6022\_3MP",  "LdcParamPath": ""  },  {  "BoardLine": "101",  "LensType": "YT\_550\_2MP",  "LdcParamPath": ""  }  ]  }  } | "Enable": Int型,是否启用自定义镜头参数, 1-启用,0-不启用,  "CurrentSelect": Int型,已选择镜头参数项  "List": Object型数组,自定义镜头参数列表  "LensType": String型,镜头类型  "BoardLine": String型,Board Line  "LdcParamPath": String型,LDC参数路径 |

|  |  |
| --- | --- |
| 设置自定义镜头参数 | |
| POST /digest/frmLensCustomCfg HTTP/1.1 | |
| 请求的正文内容  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "Enable": 1,  "CurrentSelect": 1,  "List": [  {  "LensType": "YS\_15\_3\_XM02",  "BoardLine": "101",  "LdcParamPath": ""  },  {  "BoardLine": "101",  "LensType": "YT\_6022\_3MP",  "LdcParamPath": ""  },  {  "BoardLine": "101",  "LensType": "YT\_550\_2MP",  "LdcParamPath": ""  }  ]  }  } |  |
| 响应的正文内容  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

## 热成像

### 口罩检测

|  |  |
| --- | --- |
| 获取口罩检测参数 | |
| POST /digest/frmFaceMaskPara | |
| 请求的正文内容：  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  Enable:1  }  } | "Enable": Int型,是否启用口罩检测, 1-启用,0-不启用, |
| 设置口罩检测参数 | |
| POST /digest/frmFaceMaskPara | |
| 请求的正文内容：  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "Enable": 1  }  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

### 人体测温配置

|  |  |
| --- | --- |
| 获取人体测温配置参数 | |
| POST /digest/frmHumanTemperaturePara | |
| 请求的正文内容：  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "TemperatureUnit": 0,  "Enable": 1,  "MaxTemperature": 37.300000  }  } | "TemperatureUnit": Int型,温度单位,0-摄氏温度,1-华氏温度  "Enable": Int型,是否启用人体测温,1-启用,0-不启用  "MaxTemperature": Float型,报警温度阈值 |
| 设置人体测温配置参数 | |
| POST /digest/frmHumanTemperaturePara | |
| 请求的正文内容：  {  "Type": 1,  "Dev": 1,  "Ch": 1,  "Data": {  "TemperatureUnit": 0,  "Enable": 1,  "MaxTemperature": 37.3  }  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |

### 测温设置

|  |  |
| --- | --- |
| 获取测温设置参数 | |
| POST /digest/frmVideoTemperatureCfg | |
| 请求的正文内容：  {  "Type": 0,  "Dev": 1,  "Ch": 1,  "Data": {}  } |  |
| 响应的正文内容：  {  "Result":0,  "Data":{  "Enable":1,  "InstallDistance":3,  "CompensateMode":0,  "ImageType":2,  "TemperatureUnit":0,  "TempCompensate":1,  "EmissionRate":0.980000,  "TargetDistance":3,  "TargetTemp":39,  "TargetX":321,  "TargetY":768  }  } | "Enable":1, Int型,是否启用人体测温,1-启用,0-不启用  "InstallDistance":3,Float型,目标距离,范围:[0~10.0]  "CompensateMode":0,Int型,温度补偿模式 0-自动,1-手动  "ImageType":2,Int型,图像模式,0-白光,1-黑光,2-微彩  "TemperatureUnit":0,Int型,温度单位,0-摄氏温度,1-华氏温度  "TempCompensate":1,Float型,温度补偿值,摄氏温度范围:[-10.0~10.0],华氏温度范围:[-18.0~18.0]  "EmissionRate":0.980000,Float型,发射率  "TargetDistance":3,Float型,黑体距离,范围:[0~10.0]  "TargetTemp":39,Float型,黑体温度  "TargetX":321,Int型,黑体位置,x轴坐标,范围:[0~1000]  "TargetY":768,Int型,黑体位置,Y轴坐标,范围:[0~1000] |
| 设置测温设置参数 | |
| POST /digest/frmVideoTemperatureCfg | |
| 请求的正文内容：  {  "Type":1,  "Dev":1,  "Ch":1,  "Data":{  "Enable":1,  "InstallDistance":3,  "CompensateMode":0,  "ImageType":2,  "TemperatureUnit":0,  "TempCompensate":1,  "EmissionRate":0.98,  "TargetDistance":3,  "TargetTemp":39,  "TargetX":321,  "TargetY":768  }  } |  |
| 响应的正文内容：  {  "Result": 0,  "Data": {  "StatusCode": "Operation Ok"  }  } |  |