

边缘产品对外标准接口

资料版本：V2.00

声明


版权声明

©2019 浙江宇视科技有限公司。保留一切权利。

未经浙江宇视科技有限公司（下称“本公司”）的书面许可，任何单位和个人不得擅自摘抄、复制本手册内容的部分或全部，并不得以任何形式传播。

本手册描述的产品中，可能包含本公司及其可能存在的许可人享有版权的软件。未经相关权利人的许可，任何人不能以任何形式对前述软件进行复制、分发、修改、摘录、反编译、反汇编、解密、反向工程、出租、转让、分许可等侵犯软件版权的行为。

商标声明

 是浙江宇视科技有限公司的商标或注册商标。

在本手册以及本手册描述的产品中，出现的其他商标、产品名称、服务名称以及公司名称，由其各自的所有人拥有。

出口管制合规声明

本公司遵守包括中国、美国等全球范围内适用的出口管制法律法规，并且贯彻执行与硬件、软件、技术的出口、再出口及转让相关的要求。就本手册所描述的产品，请您全面理解并严格遵守国内外适用的出口管制法律法规。

责任声明

- 本公司对使用本手册或使用本公司产品导致的任何特殊、附带、偶然或间接的损害不承担责任、不进行赔偿，包括但不限于商业利润损失、数据或文档丢失产生的损失。
- 本文档中描述的产品均“按照现状”提供，除非适用法律要求，本手册仅作为使用指导，所有陈述、信息和建议不构成任何明示或暗示的担保，包括但不限于适销性、质量满意度、适合特定目的、不侵犯第三方权利等保证。
- 若您将产品接入互联网需自担风险，包括但不限于可能遭受网络攻击、黑客攻击、病毒感染等，请您加强网络、设备数据和个人信息等的保护，采取保障设备网络安全的必要措施。本公司对因此造成的产品工作异常、信息泄露等问题不承担任何责任，但本公司会及时提供产品相关安全维护支持。
- 在适用法律未明令禁止的情况下，对于因使用或无法使用本产品或服务而引起的任何直接或间接损失，包括但不限于利润或销售损失、数据丢失或采购替代商品或服务的成本、业务中断等，本公司及其员工、许可方或附属公司都不承担赔偿责任，即使其已被告知存在此种损害的可能性也是如此。某些司法管辖区不允许对人身伤害、附带或从属损害等进行责任限制，则此限制可能不适用于您。
- 本公司对您的所有损害承担的总责任限额不超过您购买本公司产品所支付的价款。

隐私保护提醒

本公司遵循适用的隐私保护法律法规。您可以访问宇视科技官网查询我们的隐私政策。

本手册描述的产品，可能会采集人脸、指纹、车牌、邮箱、电话、GPS 等个人信息，在使用产品过程中，请遵守所在地区或国家的隐私保护法律法规要求，保障您和他人的合法权益。

关于本手册

- 本手册供多个型号产品使用，产品外观和功能请以实物为准。
- 本手册配套多个软件版本，产品界面和功能请以实际软件为准。
- 本手册内容可能包含技术性误差或印刷性错误，以本公司最终解释为准。
- 因未按本手册中的指导进行操作而造成的任何损失，由使用方自己承担责任。
- 本公司保留在没有通知或提示的情况下修改本手册中任何信息的权利。由于产品版本升级或相关地区的法律法规要求等原因，本手册内容会不定期进行更新，更新的内容将体现在新版本中。

目录

声明	2
1 鉴权	11
1.1 鉴权流程	11
1.2 鉴权信息说明	11
1.3 摘要计算过程	12
1.4 样例	12
2 人员库管理	14
2.1 新增人员库	14
基本信息	14
请求参数	14
返回数据	14
样例	15
2.2 删除人员库	15
基本信息	15
请求参数	15
返回数据	16
样例	16
2.3 修改人员库信息	16
基本信息	16
请求参数	17
返回数据	17
样例	17
2.4 获取人员库列表	18
基本信息	18
请求参数	18
返回数据	18
样例	19
3 人员管理	21
3.1 新增人员	21
基本信息	21
请求参数	21
返回数据	22
样例	22
3.2 批量新增人员	23
基本信息	23
请求参数	23
返回数据	24
样例	24
3.3 批量删除人员	25
基本信息	25

请求参数	25
返回数据	26
样例	26
3.4 修改人员信息	27
基本信息	27
请求参数	27
返回数据	28
样例	28
3.5 获取人员列表	28
基本信息	28
请求参数	29
返回数据	29
样例	30
4 点位管理	33
4.1 获取 rsa 公钥	33
基本信息	33
请求参数	33
返回数据	33
样例	33
4.2 通过 onvif 获取设备码流信息	34
基本信息	34
请求参数	34
返回数据	34
样例	35
4.3 新增点位	37
基本信息	37
请求参数	37
返回数据	38
样例	38
4.4 删除点位	39
基本信息	39
请求参数	39
返回数据	40
样例	40
4.5 修改点位信息	41
基本信息	41
请求参数	41
返回数据	42
样例	42
4.6 获取点位列表	43
基本信息	43
请求参数	43
返回数据	44
样例	45

4.7 批量新增点位	46
基本信息	46
请求参数	46
返回数据	47
样例	47
5 点位布防时间	49
5.1 添加/修改布防时间	49
基本信息	49
请求参数	49
返回数据	50
样例	50
5.2 获取布防时间	53
基本信息	53
请求参数	53
返回数据	53
样例	54
6 智能配置	58
6.1 新增（修改）智能分析任务	58
基本信息	58
请求参数	58
返回数据	61
样例	61
6.2 获取智能分析任务列表	62
基本信息	62
请求参数	62
返回数据	63
样例	65
6.3 获取算法类型列表	66
基本信息	66
请求参数	67
返回数据	67
样例	67
6.4 获取扩展配置	68
基本信息	68
请求参数	68
返回数据	69
样例	69
6.5 修改扩展配置	73
基本信息	73
请求参数	73
返回数据	74
样例	74
6.6 获取点位参数配置	77
基本信息	77

请求参数	77
返回数据	78
样例	79
6.7 修改点位参数配置	79
基本信息	79
请求参数	80
返回数据	80
样例	81
6.8 任务启停接口	81
基本信息	81
请求参数	82
返回数据	82
样例	82
6.9 实时轮巡设置	83
基本信息	83
请求参数	83
返回数据	83
样例	84
7 工服库管理	85
7.1 新增工服库	85
基本信息	85
请求参数	85
返回数据	85
样例	86
7.2 修改工服库	86
基本信息	86
请求参数	86
返回数据	87
样例	87
7.3 删除工服库	87
基本信息	87
请求参数	88
返回数据	88
样例	88
7.4 查询工服库信息	89
基本信息	89
请求参数	89
返回数据	89
样例	90
8 工服管理	92
8.1 新增工服-单个	92
基本信息	92
请求参数	92
返回数据	92

样例	93
8.2 新增工服-批量	93
基本信息	93
请求参数	93
返回数据	94
样例	94
8.3 删除工服	95
基本信息	95
请求参数	95
返回数据	96
样例	96
8.4 查找工服列表	96
基本信息	96
请求参数	97
返回数据	97
样例	98
9 设备信息	99
9.1 修改设备信息	99
基本信息	99
请求参数	99
返回数据	99
样例	100
9.2 查询设备信息	100
基本信息	100
请求参数	100
返回数据	101
样例	101
10 升级	103
10.1 上传升级包	103
基本信息	103
请求参数	103
返回数据	103
样例	103
10.2 获取升级状态	104
基本信息	104
请求参数	104
返回数据	104
样例	105
11 数据检索	106
11.1 行为分析检索	106
基本信息	106
请求参数	106
返回数据	107
样例	108

11.2 人脸抓拍检索	110
基本信息	110
请求参数	110
返回数据	111
样例	113
11.3 行为比对检索	115
基本信息	115
请求参数	115
返回数据	116
样例	117
11.4 人脸比对检索	119
基本信息	119
请求参数	119
返回数据	121
样例	122
12 平台对接	125
12.1 新增/修改平台对接信息	125
基本信息	125
请求参数	125
返回数据	126
样例	126
12.2 查询平台对接信息	128
基本信息	128
请求参数	129
返回数据	129
样例	130
12.3 查询平台对接能力	133
基本信息	133
请求参数	133
返回数据	133
样例	133
13 数据上报	135
13.1 告警数据上报	135
基本信息	135
请求参数	135
返回数据	138
样例	138
13.2 告警视频上报	139
基本信息	139
请求参数	140
返回数据	140
样例	140
14 附录	142
14.1 错误码	142

14.2 适配产品.....	143
14.3 告警类型.....	143

1 鉴权

为保证调用双方身份认证及数据安全，第三方调用本接口时，必需进行摘要鉴权。

1.1 鉴权流程

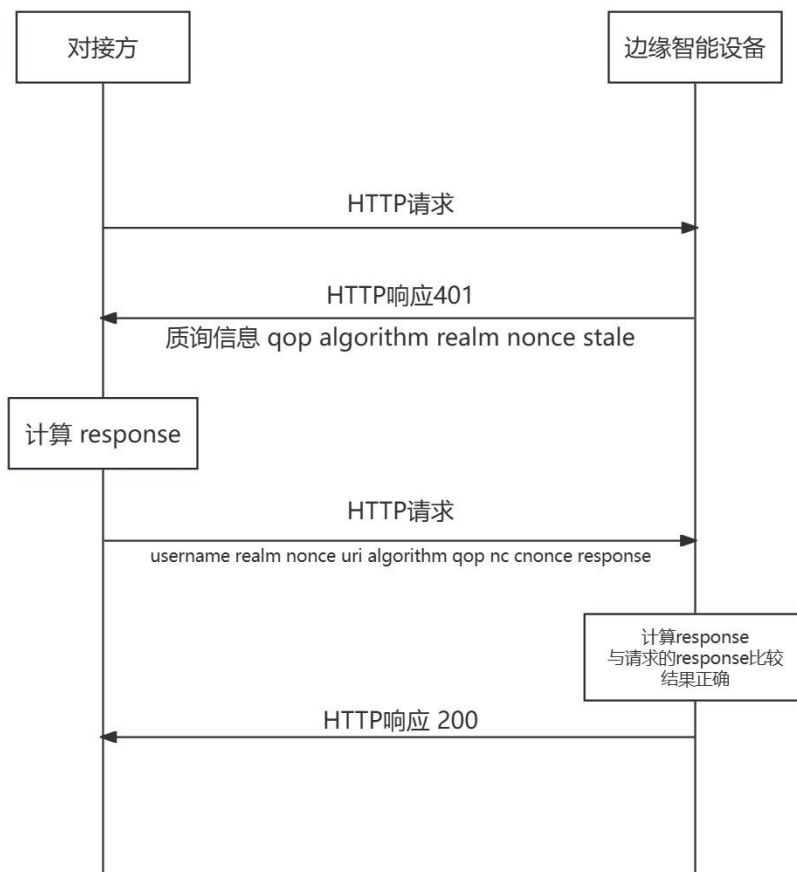


图 1.1 摘要鉴权流程

1. 调用任意接口，请求头中不携带 **Authorization**，设备返回 401。响应的头信息字段 **WWW-Authenticate** 中携带鉴权所需要的信息：**qop**、**algorithm**、**realm**、**nonce**、**stale**。
2. 根据返回的数据字段生成摘要信息，并放在请求头的 **Authorization** 字段中，再次请求设备接口，请求成功。

1.2 鉴权信息说明

字段	说明	示例
username	鉴权用户名，由设备分配	admin
password	鉴权用户密码，由设备分配	123456
realm	摘要加密字段，首次请求时由设备返回	NVRDVR
nonce	随机字符串，每当返回 401 时就会生成一个新的值，原值失效，客户端需使用此值生成新的摘要信息	1983339259
uri	当前请求的 URI	/api/v2/algo/list
method	当前请求的方法	POST
algorithm	摘要加密算法，首次请求时由设备返回	MD5
qop	保护质量，首次请求时由设备返回	auth
nc	nonce 计数器，防止重放攻击，由请求方生成	00000002
cnonce	唯一字符串，用于提供额外防护，由请求方生成	1a42d15d190f2f7
response	摘要信息，用于验证身份，由请求方生成	8413c8efa3a4
stale	指示上个请求中的 nonce 是否过期，由设备返回	FALSE

1.3 摘要计算过程

1. 将 **username**、**realm**、**password** 依次使用:拼接，并用 md5 加密后进行十六进制编码，将结果记作 **r1**。
2. 将 **method**、**uri** 依次使用:拼接，并用 md5 加密后进行十六进制编码，结果记作 **r2**。
3. 将 **r1**、**nonce**、**nc**、**cnonce**、**qop**、**r2** 依次使用:拼接，并用 md5 加密后进行十六进制编码，结果即为 **response**。

1.4 样例

首次请求：

设备提供的用户名：admin，密码：admin123.，请求 URI：/api/v2/algo/list，请求方法：POST

响应头部信息：

```
WWW-Authenticate: Digest  
qop="auth", algorithm="MD5", realm="NVRDVR", nonce="1923473498", stale="FALSE"
```

响应体数据:

```
{  
  "code": 3,  
  "msg": "Not Authorized",  
  "status_code": 401  
}
```

再次请求:

请求头信息:

```
Authorization: Digest username="admin", realm="NVRDVR", nonce="1923473498",  
uri="/api/v2/algo/list", algorithm="MD5", qop=auth, nc=00000002,  
cnonce="1a42d15d190f2f7c93cfe0f749aa4674", response="ebaadf53a48e3156cd5a1485b63658b0"
```

计算过程:

```
r1      = hex(md5(username:realm:password))  
        = hex(md5(admin:NVRDVR:admin123.))  
        = 98ca67a03cba1d7167bf0096cb957f7d  
  
r2      = hex(md5(method:uri))  
        = hex(md5(POST:/api/v2/algo/list))  
        = 305c43426cbcddb71ce670bae226bbc1  
  
response = hex(md5(r1:nonce:nc:cnonce:qop:r2))  
        = hex(md5(98ca67a03cba1d7167bf0096cb957f7d:1923473498:00000002:1  
a42d15d190f2f7c93cfe0f749aa4674:auth:305c43426cbcddb71ce670bae226bb  
c1))  
        = ebaadf53a48e3156cd5a1485b63658b0
```

响应体数据:

```
{  
  "code": 0,  
  "msg": "Succeed",  
  "status_code": 0,  
  "data": {}  
}
```

2 人员库管理

2.1 新增人员库

基本信息

Path: /api/v2/personlib/add

Method: POST

接口描述:

请求参数

Headers

参数名称	参数值	必填	示例	备注
Content-Type	application/json	是		

Body

名称	类型	必填	默认值	备注	其他信息
lib_name	string	是		人员库名称	最大长度：64

返回数据

名称	类型	必填	默认值	备注
code	integer	是		0: success
data	object	否		
└ lib_id	integer	否		人员库 id
msg	string	是		

样例

请求：

```
{  
  "lib_name": "test2"  
}
```

响应：

```
{  
  "code": 0,  
  "msg": "Succeed",  
  "status_code": 0,  
  "data": {  
    "lib_id": 5  
  }  
}
```

2.2 删除人员库

基本信息

Path: /api/v2/personlib/delete

Method: POST

接口描述：

请求参数

Headers

参数名称	参数值	必填	示例	备注
Content-Type	application/json	是		

Body

名称	类型	必填	默认值	备注
lib_id	integer []	是		人员库 id

返回数据

名称	类型	必填	默认值	备注
code	integer	是		0: success
msg	string	是		

样例

请求:

```
{
  "lib_id": [
    4
  ]
}
```

响应:

```
{
  "code": 0,
  "msg": "Succeed",
  "status_code": 0
}
```

2.3 修改人员库信息

基本信息

Path: /api/v2/personlib/update

Method: POST

接口描述:

请求参数

Headers

参数名称	参数值	必填	示例	备注
Content-Type	application/json	是		

Body

名称	类型	必填	默认值	备注	其他信息
lib_name	string	是		人员库名称	最大长度：64
lib_id	integer	是		人员库 id	

返回数据

名称	类型	必填	默认值	备注
code	integer	是		0: success
msg	string	是		

样例

请求:

```
{  
  "lib_name": "test4",  
  "lib_id": 5  
}
```

响应：

```
{
  "code": 0,
  "msg": "Succeed",
  "status_code": 0
}
```

2.4 获取人员库列表

基本信息

Path: /api/v2/personlib/list

Method: POST

接口描述：

请求参数

Headers

参数名称	参数值	必填	示例	备注
Content-Type	application/json	是		

Body

名称	类型	必填	默认值	备注
lib_name	string	否		人员库名称（模糊查询）
page	integer	是		当前页
pagesize	integer	是		每页条数

返回数据

名称	类型	必填	默认值	备注	其他信息
code	integer	是		0: success	
data	object	是			
└ list	object []	是			
└─ lib_id	integer	否		人员库 id	
└─ lib_name	string	否		人员库名称	最大长度：64
└ total	integer	是		总条数	
msg	string	是			

样例

请求:

```
{
  "lib_name": "",
  "pagesize": 16,
  "page": 1
}
```

响应:

```
{
  "code": 0,
  "msg": "Succeed",
  "status_code": 0,
  "data": {
    "total": 4,
    "list": [
      {
        "lib_id": 1,
        "lib_name": "placeat"
      },
      {
        "lib_id": 2,
        "lib_name": "test"
      }
    ]
  }
}
```

```
    },  
    {  
      "lib_id": 3,  
      "lib_name": "test-2"  
    },  
    {  
      "lib_id": 5,  
      "lib_name": "test4"  
    }  
  ]  
}  
}
```

3 人员管理

3.1 新增人员

基本信息

Path: /api/v2/person/add

Method: POST

接口描述:

请求参数

Headers

参数名称	参数值	必填	示例	备注
Content-Type	application/json	是		

Body

名称	类型	必填	默认值	备注	其他信息
person_name	string	是		姓名	最大长度：64
image_base64	string	是		人脸照片 base64	
sex	integer	否		性别 1：男 2：女 99：未说明	
email	string	否		邮箱	最大长度：64
tel	string	否		联系方式	最大长度：32
certificate_type	integer	否		证件类型 1：身份证 2：护照 3：行驶证 99：其它	
certificate_no	string	否		证件号码	最大长度：64

名称	类型	必填	默认值	备注	其他信息
birth_date	string	否		出生日期	
lib_id	integer	是		人员库 id	

返回数据

名称	类型	必填	默认值	备注
code	integer	是		0: success
data	object	否		
└ person_id	integer	否		人员 id
msg	string	是		

样例

请求:

```
{
  "person_name": "bar",
  "image_base64": "{image_base64}",
  "sex": 1,
  "email": "bar@gmail.com",
  "tel": "",
  "certificate_type": 1,
  "certificate_no": "370109200001010055",
  "birth_date": "2000-01-01",
  "lib_id": 1
}
```

响应:

```
{
  "code": 0,
  "msg": "Succeed",
}
```

```
"status_code": 0,  
"data": {  
  "person_id": 2007  
}  
}
```

3.2 批量新增人员

基本信息

Path: /api/v2/person/import

Method: POST

接口描述:

请求参数

Headers

参数名称	参数值	必填	示例	备注
Content-Type	application/json	是		

Body

名称	类型	必填	默认值	备注	其他信息
person_list	object []	是			
└ lib_id	integer	是		人员库 id	
└ person_name	string	是		姓名	最大长度：64
└ image_base64	string	是		人脸照片 base64	
└ sex	integer	否		性别 1：男 2：女 99：未说明	
└ email	string	否		邮箱	最大长度：64

名称	类型	必填	默认值	备注	其他信息
└ tel	string	否		联系方式	最大长度：32
└ certificate_type	integer	否		证件类型 1：身份证 2：护照 3：行驶证 99：其它	
└ certificate_no	string	否		证件号码	最大长度：64
└ birth_date	string	否		出生日期	

返回数据

名称	类型	必填	默认值	备注
code	integer	是		0: success
data	object	否		
└ fail_list	string []	否		导入失败的人员（姓名）
msg	string	是		

样例

请求：

```
{
  "person_list": [
    {
      "person_name": "bar",
      "image_base64": "{image_base64}",
      "sex": 1,
      "email": "bar@gmail.com",
      "tel": "",
      "certificate_type": 1,
      "certificate_no": "370109200001010055",
      "birth_date": "2000-01-01",
      "lib_id": 1
    }
  ]
}
```



```
    },  
    {  
      "person_name": "buzz",  
      "image_base64": "{image_base64}",  
      "sex": 1,  
      "email": "buzz@gmail.com",  
      "tel": "",  
      "certificate_type": 1,  
      "certificate_no": "",  
      "birth_date": "2000-01-02",  
      "lib_id": 1  
    }  
  ]  
}
```

响应:

```
{  
  "code": 0,  
  "msg": "Succeed",  
  "status_code": 0,  
  "data": {  
    "fail_list": []  
  }  
}
```

3.3 批量删除人员

基本信息

Path: /api/v2/person/delete

Method: POST

接口描述:

请求参数

Headers

参数名称	参数值	必填	示例	备注
Content-Type	application/json	是		

Body

名称	类型	必填	默认值	备注
person_id_list	integer []	是		人脸 id

返回数据

名称	类型	必填	默认值	备注
code	integer	是		0: success
msg	string	是		

样例

请求:

```
{
  "person_id_list": [
    2008,
    2006
  ]
}
```

响应:

```
{
  "code": 0,
  "msg": "Succeed",
  "status_code": 0
}
```

3.4 修改人员信息

基本信息

Path: /api/v2/person/update

Method: POST

接口描述:

请求参数

Headers

参数名称	参数值	必填	示例	备注
Content-Type	application/json	是		

Body

名称	类型	必填	默认值	备注	其他信息
person_id	integer	是		人员 id	
person_name	string	否		名称	最大长度：64
sex	integer	否		性别 1：男 2：女 99：未说明	
email	string	否		邮箱	最大长度：64
tel	string	否		联系方式	最大长度：32
certificate_type	integer	否		证件类型 1：身份证 2：护照 3：行驶证 99：其它	
certificate_no	string	否		证件号码	最大长度：64
image_base64	string	否		人脸照片 base64	
birth_date	string	否		出生日期	

返回数据

名称	类型	必填	默认值	备注
code	integer	是		0: success
msg	string	是		

样例

请求:

```
{
  "person_id": 2007,
  "person_name": "bar",
  "image_base64": "{image_base64}",
  "sex": 2,
  "email": "sample@gmail.com",
  "tel": "",
  "certificate_type": 1,
  "certificate_no": "370109200001010055",
  "birth_date": "2000-01-01",
  "lib_id": 1
}
```

响应:

```
{
  "code": 0,
  "msg": "Succeed",
  "status_code": 0
}
```

3.5 获取人员列表

基本信息

Path: /api/v2/person/list

Method: POST

接口描述:

请求参数

Headers

参数名称	参数值	必填	示例	备注
Content-Type	application/json	是		

Body

名称	类型	必填	默认值	备注	其他信息
lib_id	integer	是		人员库 id	
page	integer	是		当前页码	
pagesize	integer	是		每页条数	
person_name	string	否		姓名（支持模糊查询）	最大长度：64
sex	integer	否		性别 999：全部 1：男 2：女 99：未说明	
certificate_no	string	否		证件号	最大长度：64
model_status	integer	否		建模状态 999：全部； 1：未建模 2：建模中； 3：建模成功；4：建模失败	

返回数据

名称	类型	必填	默认值	备注	其他信息
code	string	是		0: success	
data	object	否			
└ list	object []	否			
└─ person_id	integer	是		人员 id	
└─ person_name	string	是		姓名	最大长度：64
└─ sex	integer	否		性别 1: 男 2: 女 99: 未说明	
└─ email	string	否		邮箱	最大长度：64
└─ tel	string	否		联系方式	最大长度：32
└─ modeling_type	integer	是		建模状态 1: 未建模 2: 建模中; 3: 建模成功; 4: 建模失败	
└─ certificate_type	integer	否		证件类型 1: 身份证 2: 护照 3: 行驶证 99: 其它	
└─ certificate_no	string	否		证件号码	最大长度：64
└─ image_path	string	是		图片路径	最大长度：256
└─ lib_id	integer	是		人员库 id	
└─ birth_date	string	否		出生日期	
└─ create_time	integer	否		创建时间	
└ total	integer	是		总条数	
msg	string	是			

样例

请求:

```
{
```

```
"lib_id": 1,
"page": 1,
"pagesize": 12,
"sex": 999,
"person_name": "",
"certificate_no": "",
"model_status": 999
}
```

响应:

```
{
  "code": 0,
  "msg": "Succeed",
  "status_code": 0,
  "data": {
    "total": 2,
    "list": [
      {
        "person_id": 2001,
        "person_name": "1",
        "sex": 1,
        "email": "",
        "tel": "",
        "modeling_type": 3,
        "certificate_type": 1,
        "certificate_no": "",
        "image_path": "/api/v2/smart/picture?Type=3&Index=ubs_0_0_0_0_2_1716429305/",
        "lib_id": 1,
        "birth_date": "",
        "create_time": 2737353968
      },
      {
        "person_id": 2002,
        "person_name": "2",
        "sex": 1,
        "email": "",

```

```
        "tel": "",
        "modeling_type": 3,
        "certificate_type": 1,
        "certificate_no": "",
        "image_path": "/api/v2/smart/picture?Type=3&Index=ubs_0_0_0_1_2_1716429313/",
        "lib_id": 1,
        "birth_date": "",
        "create_time": 2737362084
    }
]
}
}
```


4 点位管理

4.1 获取 rsa 公钥

基本信息

Path: /api/v2/rsa/publickey

Method: POST

接口描述:

请求参数

返回数据

名称	类型	必填	默认值	备注	其他信息
code	integer	是			
msg	string	是			
data	object	是			
└ public_key	string	是		公钥	

样例

请求:

响应:

```
{  
  "code": 0,  
  "msg": "Succeed",  
  "data": {  
    "public_key": "123456789"  
  }  
}
```

```
}  
}
```

4.2 通过 onvif 获取设备码流信息

基本信息

Path: /api/v2/channel/device/info

Method: POST

接口描述:

请求参数

Headers

参数名称	参数值	必填	示例	备注
Content-Type	application/json	是		

Body

名称	类型	必填	默认值	备注	其他信息
ip	string	是		相机 ip	
port	integer	是		相机端口	
username	string	是		相机用户名	最大长度：64
pwd	string	是		相机密码 (RSA 加密)	最大长度：512

返回数据

名称	类型	必填	默认值	备注	其他信息
code	integer	是			
data	object	否			
└ manufacturer	string	否		厂商	最大长度: 32
└ model	string	否		型号	最大长度: 32
└ firmware_version	string	否		固件版本	最大长度: 128
└ serial_number	string	否		序列号	最大长度: 64
└ hardware_id	string	否		硬件 id	最大长度: 32
└ video	object []	是		码流列表	
└ ─ id	integer	是		码流 id	
└ ─ rtsp	string	是		rtsp 地址	最大长度: 256
└ ─ bitrate	string	否		码率	最大长度: 32
└ ─ encoding	string	否		编码格式	最大长度: 32
└ ─ fps	string	否		帧率	最大长度: 32
└ ─ gop	string	否		画面组	最大长度: 32
└ ─ quality	string	否		质量	最大长度: 32
└ ─ ratio	string	否		分辨率	最大长度: 32
msg	string	必须			

样例

请求:

```
{
  "ip": "217.1.0.143",
  "port": 80,
  "username": "admin",
  "pwd": "gMsL3NeLw==="}

```

响应:

```
{
  "code": 0,
  "msg": "Succeed",
  "status_code": 0,
  "data": {
    "manufacturer": "HD",
    "model": "A27",
    "firmware_version": "IPC_C6101-B0001P80L01D1809NB",
    "serial_number": "18980000000000V00001",
    "hardware_id": "A27",
    "video": [
      {
        "id": 1,
        "rtsp": "rtsp://217.1.0.143/media/video1",
        "bitrate": "1024",
        "fps": "25",
        "encoding": "H265",
        "gop": "25",
        "quality": "5",
        "ratio": "1920*1080"
      },
      {
        "id": 2,
        "rtsp": "rtsp://217.1.0.143/media/video2",
        "bitrate": "1024",
        "fps": "25",
        "encoding": "H264",
        "gop": "50",
        "quality": "5",
        "ratio": "720*576"
      },
      {
        "id": 3,
        "rtsp": "rtsp://217.1.0.143/media/video3",
        "bitrate": "128",
```

```
        "fps": "25",
        "encoding": "H264",
        "gop": "50",
        "quality": "5",
        "ratio": "352*288"
    }
}
]
```

4.3 新增点位

基本信息

Path: /api/v2/channel/add

Method: POST

接口描述:

请求参数

Headers

参数名称	参数值	必填	示例	备注
Content-Type	application/json	是		

Body

名称	类型	必填	默认值	备注	其他信息
type	integer	是		视频协议 1: onvif 2: rtsp	
transport_type	integer	否		传输协议(AIBOX 需要, 一体机不需要) 1: tcp 2: udp	
channel_name	string	是		点位名称	最大长度: 64

名称	类型	必填	默认值	备注	其他信息
					最小长度：1
rtsp	string	否		rtsp 地址（当视频协议为 rtsp 时，该字段必填）	最大长度：256
ip	string	否		相机 ip（onvif 添加流时必填）	
port	integer	否		相机端口（onvif 添加流时必填）	
username	string	否		相机用户名	最大长度：64
pwd	string	否		相机密码（RSA 加密）	最大长度：512
video_type	integer	否		onvif 设备码流添加方式：1：主码流 2：自定义码流	
video_id	integer	否		码流 id：当选择自定义码流时，该字段必填，值为获取设备码流信息接口返回的码流 id	
custom_code	string	否		自定义编码	最大长度：64

返回数据

名称	类型	必填	默认值	备注	其他信息
code	integer	是			
data	object	否			
└ channel_id	integer	是		点位 id	
msg	string	是			

样例

请求：

```
{
  "ip": "217.1.0.143",
  "type": 1,
  "port": 80,
  "username": "admin",
  "pwd": "pwdw===",
  "rtsp": "rtsp://217.1.0.143/media/video1",
  "transport_type": 1,
  "channel_name": "Camera",
  "serialnumber": "189802C04B7201V00081",
  "manufacturer": "HD",
  "model": "A27"
}
```

响应:

```
{
  "code": 0,
  "msg": "Succeed",
  "status_code": 0,
  "data": {
    "channel_id": 1
  }
}
```

4.4 删除点位

基本信息

Path: /api/v2/channel/delete

Method: POST

接口描述:

请求参数

Headers

参数名称	参数值	必填	示例	备注
Content-Type	application/json	是		

Body

名称	类型	必填	默认值	备注
channel_id_list	integer []	是		

返回数据

名称	类型	必填	默认值	备注
code	integer	必须		0: success
msg	string	必须		

样例

请求:

```
{
  "channel_id_list": [
    1
  ]
}
```

响应:

```
{
  "code": 0,
  "msg": "Succeed"
}
```


4.5 修改点位信息

基本信息

Path: /api/v2/channel/update

Method: POST

接口描述:

请求参数

Headers

参数名称	参数值	必填	示例	备注
Content-Type	application/json	是		

Body

名称	类型	必填	默认值	备注	其他信息
channel_id	integer	是		点位 id	
transport_type	integer	否		传输协议(AIBOX 需要, 一体机不需要) 1: tcp 2: udp	
channel_name	string	是		点位名称	最大长度: 64
rtsp	string	否		rtsp 地址(当视频协议为 rtsp 时, 该字段必填)	最大长度: 256
ip	string	否		相机 ip (onvif 添加流时必填)	
port	integer	否		相机端口 (onvif 添加流时必填)	
username	string	否		相机用户名	最大长度: 64
pwd	string	否		相机密码 (RSA 加密)	最大长度: 512

名称	类型	必填	默认值	备注	其他信息
video_type	integer	否		onvif 设备码流添加方式：1：主码流 2：自定义码流	
video_id	integer	否		码流 id：当选择自定义码流时，该字段必填，值为获取设备码流信息接口返回的码流 id	
custom_code	string	否		自定义编码，长度 64	最大长度：64

返回数据

名称	类型	必填	默认值	备注
code	integer	必须		0: success
msg	string	必须		

样例

请求：

```
{
  "channel_id": 1,
  "channel_name": "Camera1",
  "status": 1,
  "ip": "217.1.0.143",
  "port": 80,
  "type": 1,
  "username": "admin",
  "off_reason": 1,
  "manufacturer": "HD",
  "model": "A27",
  "serial_number": "189802C04B7201V00081",
  "algo_model": "",
  "Nums": 1,
```

```
"pwd": "pwdQ===",
"serialnumber": "189802C04B7201V00081",
"rtsp": "rtsp://217.1.0.143/media/video1"
}
```

响应:

```
{
  "code": 0,
  "msg": "Succeed",
  "status_code": 0
}
```

4.6 获取点位列表

基本信息

Path: /api/v2/channel/list

Method: POST

接口描述:

请求参数

Headers

参数名称	参数值	必填	示例	备注
Content-Type	application/json	是		

Body

名称	类型	必填	默认值	备注	其他信息
channel_name	string	否		点位名称 (支持模糊查询)	最大长度: 64
page	integer	是		当前页	

名称	类型	必填	默认值	备注	其他信息
pagesize	integer	是		每页条数	

返回数据

名称	类型	必填	默认值	备注	其他信息
code	string	是		0: success	
data	object	是			
└ total	integer	是		总条数	
└ channel_list	object []	否		点位列表	
└─ channel_id	integer	是		点位 id	
└─ custom_code	string	否		自定义编码	最大长度：64
└─ channel_name	string	是		点位名称	最大长度：64
└─ type	integer	是		视频流协议 1: onvif 2: rtsp	
└─ transport_type	integer	否		传输协议 (AIBOX 需要, 一体机不需要) 1: tcp 2: udp	
└─ ip	string	否		点位 ip	
└─ port	number	否		点位端口	
└─ serial_number	string	否		相机序列号	最大长度：64
└─ model	string	否		型号	最大长度：32
└─ manufacturer	string	否		厂商	最大长度：32
└─ status	integer	是		1: 在线; 2: 离线	
└─ rtsp	string	否		rtsp 地址	
└─ flv	string	否		flv 地址 (一体机返回)	

名称	类型	必填	默认值	备注	其他信息
msg	string	是			

样例

请求：

```
{  
  "page": 1,  
  "pagesize": 10,  
  "channel_name": ""  
}
```

响应：

```
{  
  "code": 0,  
  "msg": "Succeed",  
  "status_code": 0,  
  "data": {  
    "total": 10,  
    "channel_list": [  
      {  
        "channel_id": 1,  
        "channel_name": "Camera1",  
        "status": 1,  
        "ip": "217.1.0.143",  
        "port": 80,  
        "type": 1,  
        "username": "admin",  
        "off_reason": 1,  
        "manufacturer": "HD",  
        "model": "A27",  
        "serial_number": "189802C0000000V00001"  
      }  
    ]  
  }  
}
```

```
}  
}
```

4.7 批量新增点位

基本信息

Path: /api/v2/channel/import

Method: POST

接口描述:

请求参数

Headers

参数名称	参数值	必填	示例	备注
Content-Type	application/json	是		

Body

名称	类型	必填	默认值	备注	其他信息
channel_list	object []	是		点位列表	
└ type	integer	是		视频流协议 1: onvif 2: rtsp	
└ transport_type	integer	否		传输协议(AIBOX 需 要, 一体机不需要) 1: tcp 2: udp	
└ channel_name	string	是		点位名称	最大长度: 64
└ rtsp	string	否		视频流 rtsp 地址	最大长度: 256
└ ip	string	否		相机 ip (onvif 添加流 时必填)	

名称	类型	必填	默认值	备注	其他信息
└ port	string	否		相机端口（onvif 添加流时必填）	
└ username	string	否		相机用户名	最大长度：64
└ pwd	string	否		相机密码（RSA 加密）	最大长度：512
└ video_type	integer	否		onvif 设备码流添加方式：1：主码流 2：自定义码流	
└ video_id	integer	否		码流 id：当选择自定义码流时，该字段必填，值为获取设备码流信息接口返回的码流 id	
└ custom_code	string	是		自定义编码	最大长度：64

返回数据

名称	类型	必填	默认值	备注
code	integer	是		0: success
data	object	否		
└ fail_list	string []	否		导入失败的点位（名称）
msg	string	是		

样例

请求：

```
{
  "channel_list": [
    {
      "ip": "217.1.0.143",
```

```
        "type": 1,
        "port": 80,
        "username": "admin",
        "pwd": "pwdw===",
        "rtsp": "rtsp://217.1.0.143/media/video1",
        "transport_type": 1,
        "channel_name": "Camera",
        "serialnumber": "189802C04B7201V00081",
        "manufacturer": "HD",
        "model": "A27"
    },
    {
        "ip": "217.1.0.144",
        "type": 1,
        "port": 80,
        "username": "admin",
        "pwd": "pwdw===",
        "rtsp": "rtsp://217.1.0.144/media/video1",
        "transport_type": 1,
        "channel_name": "Camera",
        "serialnumber": "189802C04B7201V00082",
        "manufacturer": "HD",
        "model": "A27"
    }
]
}
```

响应:

```
{
    "code": 0,
    "msg": "Succeed",
    "data": {
        "fail_list": []
    }
}
```


5 点位布防时间

5.1 添加/修改布防时间

基本信息

Path: /api/v2/control/time/update

Method: POST

接口描述:

请求参数

Headers

参数名称	参数值	必填	示例	备注
Content-Type	application/json	是		

Body

名称	类型	必填	默认值	备注
channel_id	integer	是		点位 id
algo_model	string	是		算法类型
control_time	object	是		布防时间
├ monday	object []	是		
├─ start	string	否		例：00:00
└─ end	string	否		例：24:00
├ tuesday	object []	是		
├─ start	string	否		
└─ end	string	否		
├ wednesday	object []	是		

名称	类型	必填	默认值	备注
└─ start	string	否		
└─ end	string	否		
└─ thursday	object []	是		
└─ start	string	否		
└─ end	string	否		
└─ friday	object []	是		
└─ start	string	否		
└─ end	string	否		
└─ saturday	object []	是		
└─ start	string	否		
└─ end	string	否		
└─ sunday	object []	是		
└─ start	string	否		
└─ end	string	否		

返回数据

名称	类型	必填	默认值	备注
code	integer	是		
msg	string	是		

样例

请求：

```
{
  "channel_id": 1,
  "algo_model": "FallOverAlarm",
```

```
"control_time": {
  "monday": [
    {
      "start": "00:00",
      "end": "04:00"
    },
    {
      "start": "12:30",
      "end": "24:00"
    }
  ],
  "tuesday": [
    {
      "start": "00:00",
      "end": "05:00"
    },
    {
      "start": "15:00",
      "end": "24:00"
    }
  ],
  "wednesday": [
    {
      "start": "00:00",
      "end": "05:00"
    },
    {
      "start": "15:30",
      "end": "24:00"
    }
  ],
  "thursday": [
    {
      "start": "00:00",
      "end": "04:00"
    },
    {
      "start": "12:30",
```

```
        "end": "24:00"
      }
    ],
    "friday": [
      {
        "start": "00:00",
        "end": "04:00"
      },
      {
        "start": "12:30",
        "end": "24:00"
      }
    ],
    "saturday": [
      {
        "start": "00:00",
        "end": "10:00"
      },
      {
        "start": "14:30",
        "end": "24:00"
      }
    ],
    "sunday": [
      {
        "start": "00:00",
        "end": "10:00"
      },
      {
        "start": "14:30",
        "end": "24:00"
      }
    ]
  ]
}
```

响应:

```
{
  "code": 0,
  "msg": "Succeed",
  "status_code": 0
}
```

5.2 获取布防时间

基本信息

Path: /api/v2/control/time/get

Method: POST

接口描述:

请求参数

Headers

参数名称	参数值	必填	示例	备注
Content-Type	application/json	是		

Body

名称	类型	必填	默认值	备注
channel_id	integer	是		通道 id
algo_model	string	是		算法类型

返回数据

名称	类型	必填	默认值	备注
code	integer	是		

名称	类型	必填	默认值	备注
data	object	否		
└ monday	object []	是		
└─ start	string	否		
└─ end	string	否		
└ tuesday	object []	是		
└─ start	string	否		
└─ end	string	否		
└ wednesday	object []	是		
└─ start	string	否		
└─ end	string	否		
└ thursday	object []	是		
└─ start	string	否		
└─ end	string	否		
└ friday	object []	是		
└─ start	string	否		
└─ end	string	否		
└ saturday	object []	是		
└─ start	string	否		
└─ end	string	否		
└ sunday	object []	是		
└─ start	string	否		
└─ end	string	否		
msg	string	是		

样例

请求：

```
{
  "channel_id": 1,
  "algo_model": "FallOverAlarm"
}
```

响应:

```
{
  "code": 0,
  "msg": "Succeed",
  "status_code": 0,
  "data": {
    "monday": [
      {
        "start": "00:00",
        "end": "04:00"
      },
      {
        "start": "12:30",
        "end": "24:00"
      }
    ],
    "tuesday": [
      {
        "start": "00:00",
        "end": "05:00"
      },
      {
        "start": "15:00",
        "end": "24:00"
      }
    ],
    "wednesday": [
      {
        "start": "00:00",
        "end": "05:00"
      },
      {

```

```
        "start": "15:30",
        "end": "24:00"
    }
],
"thursday": [
    {
        "start": "00:00",
        "end": "04:00"
    },
    {
        "start": "12:30",
        "end": "24:00"
    }
],
"friday": [
    {
        "start": "00:00",
        "end": "04:00"
    },
    {
        "start": "12:30",
        "end": "24:00"
    }
],
"saturday": [
    {
        "start": "00:00",
        "end": "10:00"
    },
    {
        "start": "14:30",
        "end": "24:00"
    }
],
"sunday": [
    {
        "start": "00:00",
        "end": "10:00"
```



```
    },  
    {  
      "start": "14:30",  
      "end": "24:00"  
    }  
  ]  
}  
}
```

6 智能配置

6.1 新增（修改）智能分析任务

基本信息

Path: /api/v2/smart/update

Method: POST

接口描述:

请求参数

Headers







参数名称	参数值	必填	示例	备注
Content-Type	application/json	是		

Body

名称	类型	必填	默认值	备注
smart_list	object []	是		智能任务列表
└ algo_model	string	否		算法类型 最大长度：64
└ time_threshold	integer	否		触发时间（秒）
└ person_num_limit	integer	否		人数限制
└ person_float_limit	integer	否		浮动人数
└ back_time_threshold	integer	否		背景刷新时间（秒）
└ line_dpc_enable	integer	否		绊线人数定时清零开关 （一体机不支持此参数）1：开启 0：关闭

名称	类型	必填	默认值	备注
└ line_dpc_time	integer	否		绊线人数清零时间：整数 0-23（一体机不支持）
└ detour_area	integer	否		是否为绕行区域 0 否 1 是
└ object_type	integer []	否		区域入侵检测类型 1 行人 2 非机动车 3 机动车可多选
└ max_person_object_height	integer	否		行人最大目标高，最小可为 0，最大小于 1080
└ max_person_object_width	integer	否		行人最大目标宽，最小可为 0，最大小于 1920
└ min_person_object_height	integer	否		行人最小目标高，最小可为 0，最大小于 1080
└ min_person_object_width	integer	否		行人最小目标宽，最小可为 0，最大小于 1920
└ max_vehicle_object_height	integer	否		机动车最大目标高，最小可为 0，最大小于 1080
└ max_vehicle_object_width	integer	否		机动车最大目标宽，最小可为 0，最大小于 1920
└ min_vehicle_object_height	integer	否		机动车最小目标高，最小可为 0，最大小于 1080
└ min_vehicle_object_width	integer	否		机动车最小目标宽，最小可为 0，最大小于 1920
└ max_nonVehicle_object_height	integer	否		非机动车最大目标高，最小可为 0，最大小于 1080

名称	类型	必填	默认值	备注
└ max_nonVehicle_object_width	integer	否		非机动车最大目标宽，最小可为0，最大小于1920
└ min_nonVehicle_object_height	integer	否		非机动车最小目标高，最小可为0，最大小于1080
└ min_nonVehicle_object_width	integer	否		非机动车最小目标宽，最小可为0，最大小于1920
└ workclothes_lib_list	integer []	否		工服库 id
└ graphs	object []	否		区域信息（注：各坐标值范围均为0~10000）
└─ graph_type	string	否		区域类型 区域：polygon 绊线：line 混合：blend
└─ polygon	object	否		区域坐标 当区域类型为polygon 或 blend 时，该字段代表所绘制区域的坐标
└─ detour_area	integer	否		是否为绕行区域 0 否 1 是
└─ point_x	string	否		横坐标 区域类型为polygon 或 blend 时，需绘制3-6个点，格式为x1, x2, x3, x4, x5, x6
└─ point_y	string	否		纵坐标 区域类型为polygon 或 blend 时，需绘制3-6个点，格式为y1, y2, y3, y4, y5, y6
└─ line	object	否		绊线坐标 当区域类型为line 或 blend 时，该字段为绊线坐标

名称	类型	必填	默认值	备注
 point_x	string	否		横坐标 区域类型为 line 或 blend 时，需绘制 2 个点，格式为 x1, x2
 point_y	string	否		纵坐标 区域类型为 line 或 blend 时，需绘制 2 个点，格式为 y1, y2
 direction_line	object	否		方向线
 point_x	string	否	x1, x2	
 point_y	string	否	y1, y2	
person_control_info	object	否		人员布控信息
 alarm_threshold	integer	否		告警阈值 (0-100)
 person_lib_id	integer []	否		人员库 id
channel_id	integer	是		点位 id

返回数据

名称	类型	必填	默认值	备注
code	integer	是		0: success
msg	string	是		

样例

请求:

```
{
  "smart_list": [
    {
      "algo_model": "WorkClothesAlarm",
      "workclothes_lib_list": [
```

```
        1
      ],
      "graphs": [
        {
          "graph_type": "polygon",
          "polygon": {
            "point_x": "50,9950,9950,50",
            "point_y": "50,50,9950,9950"
          }
        }
      ],
      "title": "No Work Clothes Detection",
      "id": 5,
      "paintType": "polygon"
    }
  ],
  "channel_id": 1
}
```

响应:

```
{
  "code": 0,
  "msg": "Succeed",
  "status_code": 0
}
```

6.2 获取智能分析任务列表

基本信息

Path: /api/v2/smart/list

Method: POST

接口描述:

请求参数

Headers

参数名称	参数值	必填	示例	备注
Content-Type	application/json	是		

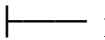
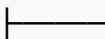
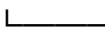
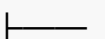
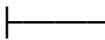
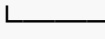
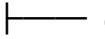
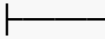
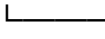
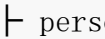
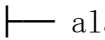
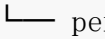
Body

名称	类型	必填	默认值	备注
channel_id	integer	必须		点位 id

返回数据

名称	类型	必填	默认值	备注
code	number	是		0: success
data	object	否		
├ smart_list	object []	否		智能任务列表
├─ algo_model	string	是		算法类型 最大长度：64
├─ time_threshold	integer	否		触发时间（秒）
├─ person_num_limit	integer	否		人数限制
├─ back_time_threshold	integer	否		背景刷新时间（秒）
├─ person_float_limit	integer	否		浮动人数
├─ line_dpc_enable	integer	否		绊线人数定时清零开关 （一体机不支持此参数） 1：开启 0：关闭
├─ line_dpc_time	integer	否		绊线人数清零时间：整数 0-23（一体机不支持）
├─ object_type	integer []	否		区域入侵检测类型 1 行人 2 非机动车 3 机动车 可多选

名称	类型	必填	默认值	备注
└─ max_person_object_height	integer	否		行人最大目标高，最小可为0，最大小于1080
└─ max_person_object_width	integer	否		行人最大目标宽，最小可为0，最大小于1920
└─ min_person_object_height	integer	否		行人最小目标高，最小可为0，最大小于1080
└─ min_person_object_width	integer	否		行人最小目标宽，最小可为0，最大小于1920
└─ max_vehicle_object_height	integer	否		机动车最大目标高，最小可为0，最大小于1080
└─ max_vehicle_object_width	integer	否		机动车最大目标宽，最小可为0，最大小于1920
└─ min_vehicle_object_height	integer	否		机动车最小目标高，最小可为0，最大小于1080
└─ min_vehicle_object_width	integer	否		机动车最小目标宽，最小可为0，最大小于1920
└─ max_nonVehicle_object_height	integer	否		非机动车最大目标高，最小可为0，最大小于1080
└─ max_nonVehicle_object_width	integer	否		非机动车最大目标宽，最小可为0，最大小于1920
└─ min_nonVehicle_object_height	integer	否		非机动车最小目标高，最小可为0，最大小于1080
└─ min_nonVehicle_object_width	integer	否		非机动车最小目标宽，最小可为0，最大小于1920
└─ workclothes_lib_list	integer []	否		工服库 id
└─ graphs	object []	是		区域信息（注：各坐标值范围均为0~10000）
└─ graph_type	string	是		区域类型 区域：polygon 绊线：line 混合：blend

名称	类型	必填	默认值	备注
 polygon	object	否		区域坐标 当区域类型为 polygon 或 blend 时，该字段代表所绘制区域的坐标
 point_x	string	是		横坐标 区域类型为 polygon 或 blend 时，需绘制 3-6 个点，格式为 x1, x2, x3, x4, x5, x6
 point_y	string	是		纵坐标 区域类型为 polygon 或 blend 时，需绘制 3-6 个点，格式为 y1, y2, y3, y4, y5, y6
 line	object	否		绊线坐标 当区域类型为 line 或 blend 时，该字段为绊线坐标
 point_x	string	是		横坐标 区域类型为 line 或 blend 时，需绘制 2 个点，格式为 x1, x2
 point_y	string	是		纵坐标 区域类型为 line 或 blend 时，需绘制 2 个点，格式为 y1, y2
 direction_line	object	否		方向线
 point_x	string	是		x1, x2
 point_y	string	是		y1, y2
 person_control_info	object	否		人员布控信息
 alarm_threshold	string	否		告警阈值 (0-100)
 person_lib_id	integer []	否		人员库 id
msg	string	是		

样例

请求：

```
{
  "channel_id": 1
}
```

响应:

```
{
  "code": 0,
  "msg": "Succeed",
  "status_code": 0,
  "data": {
    "smart_list": [
      {
        "algo_model": "FallOverAlarm",
        "time_threshold": 5,
        "graphs": [
          {
            "polygon": {
              "point_x": "50,9950,9950,50",
              "point_y": "50,50,9950,9950"
            },
            "graph_type": "polygon"
          }
        ]
      }
    ]
  }
}
```

6.3 获取算法类型列表

基本信息

Path: /api/v2/algo/list

Method: POST

接口描述:

请求参数

Headers

参数名称	参数值	必填	示例	备注
Content-Type	application/json	是		

Body

名称	类型	必填	默认值	备注	其他信息
----	----	----	-----	----	------

返回数据

名称	类型	必填	默认值	备注
code	number	否		0: success
data	object	否		
└ algo_model	string []	是		算法类型
msg	string	否		

样例

请求:

响应:

```
{
  "code": 0,
  "msg": "Succeed",
  "status_code": 0,
  "data": {
    "algo_model": [
      "ObjectIsRecognized",
      "FieldDetectorObjectsInside",
    ]
  }
}
```

```
        "LineRuleData",
        "AreaRuleData",
        "AccessElevatorAlarm",
        "ObjectRemoved",
        "SafetyHelmetAlarm",
        "WorkClothesAlarm",
        "TelephoningAlarm",
        "NoMaskAlarm",
        "FallOverAlarm",
        "OffDutyDetectionAlarm",
        "SleepingDetectionAlarm",
        "ReflectiveClothesDetectionAlarm",
        "SmokingAlarm",
        "PlayMobilePhoneDetection",
        "FireDetection",
        "FumesAlarmBegin",
        "CrowdDensityCriticalAlarm",
        "AbnormalParkingDetection",
        "ChannelBlockageDetection"
    ]
}
}
```

6.4 获取扩展配置

基本信息

Path: /api/v2/extend/get

Method: POST

接口描述:

请求参数

Headers

参数名称	参数值	必填	示例	备注
Content-Type	application/json	是		

Body

名称	类型	必填	默认值	备注
channel_id	integer	否		点位 id, 不传则返回全部点位的扩展配置信息

返回数据

名称	类型	必填	默认值	备注
code	integer	是		
data	object []	是		
└ channel_id	integer	是		点位 id
└ status	integer	是		使能开关 1：开启 0：关闭 当使能开关关闭时，点位使用默认的灵敏度、告警间隔进行下发
└ smart_list	object []	是		
└─ algo_model	string	是		算法类型 最大长度：64
└─ sensitive	integer	否		灵敏度（1-100）
└─ report_rate	integer	否		告警间隔（单位：秒；设置为0时，一次告警只上报一次）
msg	string	是		

样例

请求：

```
{  
  "channel_id": 1  
}
```

```
}
```

响应:

```
{  
  "code": 0,  
  "msg": "Succeed",  
  "status_code": 0,  
  "data": [  
    {  
      "status": 0,  
      "channel_id": 1,  
      "smart_list": [  
        {  
          "algo_model": "FieldDetectorObjectsInside",  
          "sensitive": 80  
        },  
        {  
          "algo_model": "LineDetectorCrossed",  
          "sensitive": 80  
        },  
        {  
          "algo_model": "EnterArea",  
          "sensitive": 80  
        },  
        {  
          "algo_model": "LeaveArea",  
          "sensitive": 80  
        },  
        {  
          "algo_model": "ObjectRemoved",  
          "sensitive": 80,  
          "report_rate": 0  
        },  
        {  
          "algo_model": "LineRuleData",  
          "sensitive": 80  
        },  
      ],  
    },  
  ],  
}
```

```
{
  "algo_model": "AreaRuleData",
  "sensitive": 80
},
{
  "algo_model": "AccessElevatorAlarm",
  "sensitive": 80,
  "report_rate": 0
},
{
  "algo_model": "SmokingAlarm",
  "sensitive": 80,
  "report_rate": 0
},
{
  "algo_model": "SafetyHelmetAlarm",
  "sensitive": 80,
  "report_rate": 0
},
{
  "algo_model": "WorkClothesAlarm",
  "sensitive": 80,
  "report_rate": 0
},
{
  "algo_model": "TelephoningAlarm",
  "sensitive": 80,
  "report_rate": 0
},
{
  "algo_model": "NoMaskAlarm",
  "sensitive": 80,
  "report_rate": 0
},
{
  "algo_model": "FallOverAlarm",
  "sensitive": 80,
  "report_rate": 0
}
```

```
    },
    {
      "algo_model": "LongStayDetection",
      "report_rate": 0
    },
    {
      "algo_model": "FightDetectionAlarm",
      "sensitive": 80,
      "report_rate": 0
    },
    {
      "algo_model": "OffDutyDetectionAlarm",
      "report_rate": 0
    },
    {
      "algo_model": "SleepingDetectionAlarm",
      "sensitive": 80,
      "report_rate": 0
    },
    {
      "algo_model": "ReflectiveClothesDetectionAla
rm",
      "sensitive": 80,
      "report_rate": 0
    },
    {
      "algo_model": "PlayMobilePhoneDetection",
      "sensitive": 80,
      "report_rate": 0
    },
    {
      "algo_model": "FireDetection",
      "sensitive": 80,
      "report_rate": 0
    },
    {
      "algo_model": "FumesAlarmBegin",
      "sensitive": 80,
```



```
        "report_rate": 0
    },
    {
        "algo_model": "CrowdDensityCriticalAlarm",
        "report_rate": 0
    },
    {
        "algo_model": "AbnormalParkingDetection",
        "report_rate": 0
    },
    {
        "algo_model": "ChannelBlockageDetection",
        "report_rate": 0
    }
]
}
]
```

6.5 修改扩展配置

基本信息

Path: /api/v2/extend/update

Method: POST

接口描述:

请求参数

Headers

参数名称	参数值	必填	示例	备注
Content-Type	application/json	是		

Body

名称	类型	必填	默认值	备注
list	object []	是		
└ channel_id	integer	是		点位 id
└ status	integer	是		使能开关 1：开启 0：关闭 当使能开关关闭时，点位使用默认的灵敏度、告警间隔进行下发
└ smart_list	object []	是		
└─ algo_model	string	是		算法类型 最大长度：64
└─ sensitive	integer	否		灵敏度（1-100）
└─ report_rate	integer	否		告警间隔（单位：秒；设置为 0 时，一次告警只上报一次）

返回数据

名称	类型	必填	默认值	备注
code	integer	是		
msg	string	是		

样例

请求：

```
{
  "list": [
    {
      "channel_id": 1,
      "smart_list": [
        {
          "algo_model": "FieldDetectorObjectsInside",
          "sensitive": 80
        }
      ],
    }
  ]
}
```

```
"algo_model": "ObjectRemoved",
"sensitive": 80,
"report_rate": 0
},
{
"algo_model": "LineRuleData",
"sensitive": 80
},
{
"algo_model": "AreaRuleData",
"sensitive": 80
},
{
"algo_model": "AccessElevatorAlarm",
"sensitive": 80,
"report_rate": 0
},
{
"algo_model": "SmokingAlarm",
"sensitive": 80,
"report_rate": 0
},
{
"algo_model": "SafetyHelmetAlarm",
"sensitive": 80,
"report_rate": 0
},
{
"algo_model": "WorkClothesAlarm",
"sensitive": 80,
"report_rate": 0
},
{
"algo_model": "TelephoningAlarm",
"sensitive": 80,
"report_rate": 0
},
{
```

```
    "algo_model": "NoMaskAlarm",
    "sensitive": 80,
    "report_rate": 0
  },
  {
    "algo_model": "FallOverAlarm",
    "sensitive": 80,
    "report_rate": 0
  },
  {
    "algo_model": "OffDutyDetectionAlarm",
    "report_rate": 0
  },
  {
    "algo_model": "SleepingDetectionAlarm",
    "sensitive": 80,
    "report_rate": 0
  },
  {
    "algo_model": "ReflectiveClothesDetectionAla
rm",
    "sensitive": 80,
    "report_rate": 0
  },
  {
    "algo_model": "PlayMobilePhoneDetection",
    "sensitive": 80,
    "report_rate": 0
  },
  {
    "algo_model": "FireDetection",
    "sensitive": 80,
    "report_rate": 0
  },
  {
    "algo_model": "FumesAlarmBegin",
    "sensitive": 80,
    "report_rate": 0
  }
```

```
    },
    {
      "algo_model": "CrowdDensityCriticalAlarm",
      "report_rate": 0
    },
    {
      "algo_model": "AbnormalParkingDetection",
      "report_rate": 0
    },
    {
      "algo_model": "ChannelBlockageDetection",
      "report_rate": 0
    }
  ],
  "status": 1
}
]
```

响应:

```
{
  "code": 0,
  "msg": "Succeed",
  "status_code": 0
}
```

6.6 获取点位参数配置

基本信息

Path: /api/v2/smart/params/get

Method: POST

接口描述:

请求参数

Headers

参数名称	参数值	必填	示例	备注
Content-Type	application/json	是		

Body

名称	类型	必填	默认值	备注
channel_id	integer	是		点位 id

返回数据

名称	类型	必填	默认值	备注
code	integer	是		
data	object	是		
└ alarm_video_param	object	是		告警视频参数
└─ status	integer	是		是否开启告警视频（默认关闭） 1：开启 0：关闭
└─ befor_alarmed_time	integer	是		告警前时间（单位：秒；范围 5-30）
└─ after_alarmed_time	integer	是		告警后时间（单位：秒；范围 5-30）
└ status_recovery_param	object	是		告警恢复参数
└─ status	integer	是		是否开启告警恢复（默认关闭） 1：开启 0：关闭
└ face_param	object	是		人脸参数
└─ min_width	string	是		人脸最小过滤尺寸宽度，建议默认 10
└─ min_height	string	是		人脸最小过滤尺寸高度，建议默认 10
msg	string	是		

样例

请求:

```
{  
  "channel_id": 1  
}
```

响应:

```
{  
  "code": 0,  
  "msg": "Succeed",  
  "status_code": 0,  
  "data": {  
    "alarm_video_param": {  
      "status": 0,  
      "befor_alarmed_time": 5,  
      "after_alarmed_time": 5  
    },  
    "face_param": {  
      "min_width": 30,  
      "min_height": 30  
    },  
    "status_recovery_param": {  
      "status": 0  
    }  
  }  
}
```

6.7 修改点位参数配置

基本信息

Path: /api/v2/smart/params/update

Method: POST

接口描述：

请求参数

Headers

参数名称	参数值	必填	示例	备注
Content-Type	application/json	是		

Body

名称	类型	必填	默认值	备注
channel_id	integer	是		点位 id
alarm_video_param	object	否		告警视频参数
└ status	integer	是		是否开启告警视频（默认关闭） 1：开启 0：关闭
└ befor_alarmed_time	integer	是		告警前时间（单位：秒；范围 5-30）
└ after_alarmed_time	integer	是		告警后时间（单位：秒；范围 5-30）
status_recovery_param	object	否		告警恢复参数
└ status	integer	是		是否开启告警恢复（默认关闭） 1：开启 0：关闭
face_param	object	否		人脸参数
└ min_width	integer	是		人脸最小过滤尺寸宽度，建议默认 10
└ min_height	integer	是		人脸最小过滤尺寸高度，建议默认 10

返回数据

名称	类型	必填	默认值	备注
code	integer	必须		
msg	string	必须		

样例

请求：

```
{
  "channel_id": 1,
  "alarm_video_param": {
    "status": 0,
    "befor_alarmed_time": 5,
    "after_alarmed_time": 5
  },
  "status_recovery_param": {
    "status": 0
  },
  "face_param": {
    "min_width": 30,
    "min_height": 30
  }
}
```

响应：

```
{
  "code": 0,
  "msg": "Succeed",
  "status_code": 0
}
```

6.8 任务启停接口

基本信息

Path: /api/v2/smart/task

Method: POST

接口描述:

请求参数

Headers

参数名称	参数值	必填	示例	备注
Content-Type	application/json	是		

Body

名称	类型	必填	默认值	备注
channel_id	integer	是		点位 id
status	integer	是		1: 开启 0: 关闭

返回数据

名称	类型	必填	默认值	备注
code	integer	是		
msg	string	是		

样例

请求:

```
{  
  "channel_id": 1,  
  "status": 1  
}
```

响应:

```
{
  "code": 0,
  "msg": "Succeed",
  "status_code": 0
}
```

6.9 实时轮巡设置

基本信息

Path: /api/v2/smart/cycle

Method: POST

接口描述:

请求参数

Headers

参数名称	参数值	必填	示例	备注
Content-Type	application/json	是		

Body

名称	类型	必填	默认值	备注
channel_id	integer	是		点位 id
status	integer	是		1: 实时 2: 轮巡

返回数据

名称	类型	必填	默认值	备注
code	integer	是		

名称	类型	必填	默认值	备注
msg	string	是		

样例

请求：

```
{  
  "channel_id": 1,  
  "status": 1  
}
```

响应：

```
{  
  "code": 0,  
  "msg": "Succeed",  
  "status_code": 0  
}
```

7 工服库管理

7.1 新增工服库

基本信息

Path: /api/v2/workclotheslib/add

Method: POST

接口描述:

请求参数

Headers

参数名称	参数值	必填	示例	备注
Content-Type	application/json	是		

Body

名称	类型	必填	默认值	备注	其他信息
lib_name	string	是		工服库名称	最大长度：64 最小长度：1

返回数据

名称	类型	必填	默认值	备注
code	integer	是		
data	object	是		
└ lib_id	integer	是		工服库 id
msg	string	是		

样例

请求：

```
{
  "lib_name": "lib1"
}
```

响应：

```
{
  "code": 0,
  "msg": "Succeed",
  "status_code": 0,
  "data": {
    "lib_id": 2
  }
}
```

7.2 修改工服库

基本信息

Path: /api/v2/workclotheslib/update

Method: POST

接口描述：

请求参数

Headers

参数名称	参数值	必填	示例	备注
Content-Type	application/json	是		

Body

名称	类型	必填	默认值	备注	其他信息
lib_id	integer	是		工服库 id	
lib_name	string	是		工服库名称	最大长度：64 最小长度：1

返回数据

名称	类型	必填	默认值	备注
code	integer	是		0: success
msg	string	是		

样例

请求:

```
{
  "lib_name": "lib2",
  "lib_id": 2
}
```

响应:

```
{
  "code": 0,
  "msg": "Succeed",
  "status_code": 0
}
```

7.3 删除工服库

基本信息

Path: /api/v2/workclotheslib/delete

Method: POST

接口描述:

请求参数

Headers

参数名称	参数值	必填	示例	备注
Content-Type	application/json	是		

Body

名称	类型	必填	默认值	备注
lib_id	integer	是		工服库 id

返回数据

名称	类型	必填	默认值	备注
code	integer	是		
msg	string	是		

样例

请求:

```
{  
  "lib_id": 2  
}
```

响应:

```
{  
  "code": 0,  
  "msg": "Succeed",  
}
```



```
"status_code": 0  
}
```

7.4 查询工服库信息

基本信息

Path: /api/v2/workclotheslib/list

Method: POST

接口描述:

请求参数

Headers

参数名称	参数值	必填	示例	备注
Content-Type	application/json	是		

Body

名称	类型	必填	默认值	备注
lib_name	string	否		工服库名称（支持模糊查询）
page	integer	是		当前页
pagesize	integer	是		每页条数

返回数据

名称	类型	必填	默认值	备注
code	integer	是		
data	object	否		

名称	类型	必填	默认值	备注
└─ workcloth_lib_list	object []	是		
└─ lib_id	integer	是		工服库 id
└─ lib_name	string	是		工服库名称 最大长度： 64
└─ create_time	integer	是		创建时间，10 位秒级时间戳
└─ total	integer	是		总条数
msg	string	是		

样例

请求：

```
{
  "lib_name": "",
  "page": 1,
  "pagesize": 12
}
```

响应：

```
{
  "code": 0,
  "msg": "Succeed",
  "status_code": 0,
  "data": {
    "total": 1,
    "workcloth_lib_list": [
      {
        "lib_id": 1,
        "lib_name": "默认工服库",
        "create_time": 1550108725
      }
    ]
  }
}
```

```
}  
}
```

8 工服管理

8.1 新增工服-单个

基本信息

Path: /api/v2/workclothes/add

Method: POST

接口描述:

请求参数

Headers

参数名称	参数值	必填	示例	备注
Content-Type	application/json	是		

Body

名称	类型	必填	默认值	备注
lib_id	number	是		工服库 id
image_base64	string	是		工服图片 base64

返回数据

名称	类型	必填	默认值	备注
code	integer	是		
data	object	否		
└ workclothes_id	integer	否		工服 id

名称	类型	必填	默认值	备注
msg	string	是		

样例

请求：

```
{  
  "lib_id": 1,  
  "image_base64": "/9j/4AAQSkZJRgABAQEAYABg"  
}
```

响应：

```
{  
  "code": 0,  
  "msg": "Succeed",  
  "status_code": 0,  
  "data": {  
    "workclothes_id": 2001  
  }  
}
```

8.2 新增工服-批量

基本信息

Path: /api/v2/workclothes/batchadd

Method: POST

接口描述：

一次最多添加 50 条

请求参数

Headers

参数名称	参数值	必填	示例	备注
Content-Type	application/json	是		

Body

名称	类型	必填	默认值	备注
lib_id	integer	是		工服库 id
image_base64	string []	是		工服图片 base64

返回数据

名称	类型	必填	默认值	备注
code	integer	是		
data	object	否		
└─ workclothes_id	integer []	否		成功工服 id
└─ fail_list	object []	否		失败图片列表
└─ index	integer	是		失败图片序号 从 0 开始
└─ reason	string	是		失败原因 最大长度：64
msg	string	是		

样例

请求：

```
{
  "lib_id": 1,
  "image_base64": [
    "/9j/4AAQSkZJRgABAQEAYABg"
  ]
}
```

响应：

```
{
  "code": 0,
  "msg": "Succeed",
  "status_code": 0,
  "data": {
    "workclothes_id": [
      2001
    ],
    "fail_list": []
  }
}
```

8.3 删除工服

基本信息

Path: /api/v2/workclothes/delete

Method: POST

接口描述：

请求参数

Headers

参数名称	参数值	必填	示例	备注
Content-Type	application/json	是		

Body

名称	类型	必填	默认值	备注
lib_id	integer	是		工服库 id

名称	类型	必填	默认值	备注
workclothes_id_list	integer []	是		工服 id

返回数据

名称	类型	必填	默认值	备注
code	integer	是		
msg	string	是		

样例

请求：

```
{
  "lib_id": 1,
  "workclothes_id_list": [
    2001
  ]
}
```

响应：

```
{
  "code": 0,
  "msg": "Succeed",
  "status_code": 0
}
```

8.4 查找工服列表

基本信息

Path: /api/v2/workclothes/list

Method: POST

接口描述：

请求参数

Headers

参数名称	参数值	必填	示例	备注
Content-Type	application/json	是		

Body

名称	类型	必填	默认值	备注
lib_id	integer	是		工服库 id
page	integer	是		当前页
pagesize	integer	是		每页条数
model_status	integer	否		建模状态 999：全部；1：未建模 2：建模中；3：建模成功；4：建模失败

返回数据

名称	类型	必填	默认值	备注
code	integer	是		
data	object	是		
└ total	integer	是		总条数
└ workclothes_list	object	是		
└─ lib_id	number	是		工服库 id
└─ workclothes_id	number	是		工服 id
└─ image_path	string	是		图片相对路径 最大长度：256
└─ create_time	integer	是		创建时间，10 位时间戳，秒级

名称	类型	必填	默认值	备注
└─ model_status	number	否		建模状态 1：未建模 2：建模中； 3：建模成功； 4：建模失败
msg	string	是		

样例

请求：

```
{  
  "lib_id": 1,  
  "model_status": 999,  
  "page": 1,  
  "pagesize": 12  
}
```

响应：

```
{  
  "code": 0,  
  "msg": "Succeed",  
  "status_code": 0,  
  "data": {  
    "workclothes_list": [],  
    "total": 0  
  }  
}
```

9 设备信息

9.1 修改设备信息

基本信息

Path: /api/v2/device/update

Method: POST

接口描述:

请求参数

Headers

参数名称	参数值	必填	示例	备注
Content-Type	application/json	是		

Body

名称	类型	必填	默认值	备注	其他信息
device_name	string	是		设备名称	最大长度：64
device_code	integer	是		设备编号, 1-255	最大值：255
custom_code	string	否		自定义编码	最大长度：64

返回数据

名称	类型	必填	默认值	备注
code	integer	是		0: success
msg	string	是		

样例

请求：

```
{
  "device_name": "ECS-B501@8HA-SF",
  "device_code": 1
}
```

响应：

```
{
  "code": 0,
  "msg": "Succeed",
  "status_code": 0
}
```

9.2 查询设备信息

基本信息

Path: /api/v2/device/get

Method: POST

接口描述：

请求参数

Headers

参数名称	参数值	必填	示例	备注
Content-Type	application/json	是		

Body

名称	类型	必填	默认值	备注	其他信息
----	----	----	-----	----	------

返回数据

名称	类型	必填	默认值	备注	其他信息
code	integer	是		0: success	
msg	string	是			
data	object	是			
└ device_name	string	是		设备名称	最大长度：64
└ device_model	string	是		设备型号	最大长度：32
└ serial_number	string	是		设备序列号	最大长度：64
└ firmware_version	string	是		软件版本号	最大长度：128
└ release_time	integer	是		版本发布时间，时间戳	
└ device_code	integer	是		设备编码， 1-255。 默认是1。	
└ custom_code	string	否		自定义编码	最大长度：64

样例

请求：

响应：

```
{  
  "code": 0,  
  "msg": "Succeed",  
  "status_code": 0,  
  "data": {  
    "device_name": "ECS-B501@8HA-SF",  
    "device_model": "ECS-B501@8HA-SF",  
    "serial_number": "210235C8TS5804696123",
```

```
"firmware_version": "ECS_B501_SF-B1105.2.05.240516",  
"release_time": 1715796720,  
"device_code": 1  
}  
}
```

10 升级

10.1 上传升级包

基本信息

Path: /api/v2/upgrade

Method: POST

接口描述:

请求参数

Headers

参数名称	参数值	必填	示例	备注
Content-Type	multipart/form-data	是		

Body

参数名称	参数类型	必填	示例	备注
file	file	是		升级包

返回数据

名称	类型	必填	默认值	备注
code	integer	是		
msg	string	是		

样例

请求:

响应:

```
{
  "code": 0,
  "msg": "Succeed",
  "status_code": 0
}
```

10.2 获取升级状态

基本信息

Path: /api/v2/upgrade/status

Method: POST

接口描述:

请求参数

Headers

参数名称	参数值	必填	示例	备注
Content-Type	application/x-www-form-urlencoded	是		

返回数据

名称	类型	必填	默认值	备注
code	integer	是		
data	object	是		
└ status	integer	是		1: 传输中 2: 升级中 3: 升级成功 4: 升级失败 注: 升级完成后设备会进行重启, 此时接口无法调通, 请耐心等待直到返回升级成功或升级失败
msg	string	是		

样例

请求：

响应：

```
{  
  "code": 0,  
  "msg": "Succeed",  
  "data": {  
    "status": 2  
  },  
  "status_code": 0  
}
```

11 数据检索

11.1 行为分析检索

基本信息

Path: /api/v2/search/behavior

Method: POST

接口描述:

请求参数

Headers

参数名称	参数值	必填	示例	备注
Content-Type	application/json	是		

Body

名称	类型	必填	默认值	备注
page	integer	是		页码数
pagesize	integer	是		每页数据量
channel_id	integer []	是		通道 id
algo_model	string []	否		算法类型，送空查全部
start_time	integer	是		开始时间，10 位时间戳（秒级）
end_time	integer	是		结束时间，10 位时间戳（秒级）
compare_flag	integer	否		是否有比对：1 全部，2 有比对 不传查全部

名称	类型	必填	默认值	备注
object_type	integer[]	否		目标类型 区域入侵告警查询条件 1 行人 2 机动车 3 非机动车
plate_no	string	否		车牌号

返回数据

名称	类型	必填	默认值	备注	其他信息
code	integer	是		0: success	
msg	string	是			
data	object	是			
├ total	integer	是		总条数	
├ list	object []	否			
├─ alarm_id	integer	是		告警 id	
├─ big_picture_url	string	是		告警大图地址	最大长度：256
├─ algo_model	string	是		算法类型	最大长度：128
├─ capture_time	integer	是		告警发生的时间戳，秒级	
├─ channel_id	integer	是		告警所在通道	
├─ channel_name	string	是		通道名称	最大长度：64
├─ video_url	string	否		告警视频地址	最大长度：256
├─ object_list	object []	是			
├─ plate_no	string	是		车牌号（仅车辆违停占用上报）	最大长度：16
├─ compare_list	object []	否		人脸比对结果	

名称	类型	必填	默认值	备注	其他信息
base_image_id	integer	否		人脸底图 id	
base_image_url	string	否		人脸底图 url	最大长度：256
name	string	否		姓名	最大长度：64
certificate_no	string	否		证件号	最大长度：64
similarity	integer	否		相似度 1-100	
match_type	number	是		1：比对成功，2 比对失败	

样例

请求：

```
{
  "page": 1,
  "pagesize": 10,
  "channel_id": [
    1,
    2,
    3,
    4,
    5,
    6,
    7,
    8
  ],
  "algo_model": [
    "FieldDetectorObjectsInside",
    "SafetyHelmetAlarm",
    "WorkClothesAlarm",
    "TelephoningAlarm",
    "NoMaskAlarm",
    "FallOverAlarm",
    "OffDutyDetectionAlarm",
```

```
    "SmokingAlarm",
    "SleepingDetectionAlarm",
    "ReflectiveClothesDetectionAlarm",
    "PlayMobilePhoneDetection",
    "FireDetection",
    "FumesAlarmBegin",
    "CrowdDensityCriticalAlarm",
    "AbnormalParkingDetection",
    "ChannelBlockageDetection",
    "AccessElevatorAlarm",
    "ObjectRemoved"
  ],
  "start_time": 1716480000,
  "end_time": 1716566399,
  "compare_flag": 1
}
```

响应:

```
{
  "code": 0,
  "msg": "Succeed",
  "status_code": 0,
  "data": {
    "total": 1210,
    "list": [
      {
        "alarm_id": 6318,
        "capture_time": 1716521787,
        "channel_id": 1,
        "channel_name": "Camera1",
        "algo_model": "SafetyHelmetAlarm",
        "big_picture_url": "/api/v2/smart/picture?Type=1
&Index=ubs_5_0_0_0_0_0_752_55_64647_1716521787&Size=222974",
        "object_list": [
          {
            "plate_no": ""
          }
        ]
      }
    ]
  }
}
```

```
    ],  
    "compare_list": [  
      {  
        "match_type": 1,  
        "base_image_id": 2005,  
        "name": "5",  
        "similarity": 95,  
        "certificate_no": "",  
        "base_image_url": "/api/v2/smart/picture  
?Type=3&Index=ubs_0_0_0_4_2_1716429337/Pic_1716429337015.jpg&Size=9999"  
      }  
    ]  
  }  
]
```

11.2 人脸抓拍检索

基本信息

Path: /api/v2/search/facecap

Method: POST

接口描述:

请求参数

Headers

参数名称	参数值	必填	示例	备注
Content-Type	application/json	是		

Body

名称	类型	必填	默认值	备注	其他信息
page	integer	是		页码数	
pagesize	integer	是		每页数据量	
channel_id	integer []	是		通道 id	
start_time	integer	是		开始时间，10 位时间戳（秒级）	
end_time	integer	是		结束时间，10 位时间戳（秒级）	
faceattr	object	否			
└ gender	integer	否		性别：999：全部 1：男 2：女 99：未知	
└ age	integer	否		年龄：999：全部 5：儿童 20：少年 40：青年 60：中年 99：老年	
└ glass	integer	否		眼镜：999：全部 1：未佩戴 眼睛 2：佩戴眼镜	
└ mask	integer	否		口罩：999：全部 1：无口罩 2：有口罩	
compare_flag	integer	是		是否有比对：1 全部，2 有比对 不传查全部	
name	string	否		人员姓名，模糊查询 最大长度：50 最小长度：10	最大长度：64
certificate_no	string	否		证件号，模糊查询	最大长度：64
similarity	object	否		相似度，0-100	
└ min	integer	是		最小值	
└ max	integer	是		最大值	

返回数据

名称	类型	必填	默认值	备注	其他信息
code	integer	是		0: success	
msg	string	是			
data	object	否			
└ total	integer	是		总条数	
└ list	object []	是			
├─ alarm_id	integer	是		告警 id	
├─ big_picture_url	string	是		告警大图地址	最大长度：256
├─ small_picture_url	string	是		告警小图地址	最大长度：256
├─ capture_time	integer	是		告警发生的时间戳，秒级	
├─ channel_id	integer	是		告警所在通道	
├─ channel_name	string	是		通道名称	最大长度：64
├─ gender	integer	是		性别： 1：男 2：女 99：未知	
├─ glass	integer	是		眼镜： 1：未佩戴眼睛 2：佩戴眼镜	
├─ age	integer	是		年龄 5：儿童 20：少年 40：青年 60：中年 99：老年	
├─ mask	integer	是		口罩： 1：无口罩 2： 有口罩	
├─ compare_list	object []	否			
├─ base_image_id	integer	否		人脸底图 id	
├─ base_image_url	string	否		人脸底图 url	最大长度：256
├─ name	string	否		姓名	最大长度：64
├─ similarity	integer	否		相似度 1-100	
├─ certificate_no	string	否		证件号	最大长度：64

名称	类型	必填	默认值	备注	其他信息
match_type	number	是		1: 比对成功, 2: 比对失败	

样例

请求:

```
{
  "page": 1,
  "pagesize": 10,
  "channel_id": [
    1,
    2,
    3,
    4,
    5,
    6,
    7,
    8
  ],
  "start_time": 1715702400,
  "end_time": 1716566399,
  "faceattr": {
    "gender": 999,
    "age": 999,
    "glass": 999,
    "mask": 999
  },
  "compare_flag": 1,
  "name": "",
  "match_type": 999,
  "certificate_no": "",
  "similarity": {
    "min": 1,
    "max": 100
  }
}
```

```
}
```

```
}
```

响应:

```
{
```

```
  "code": 0,
```

```
  "msg": "Succeed",
```

```
  "status_code": 0,
```

```
  "data": {
```

```
    "total": 2028,
```

```
    "list": [
```

```
      {
```

```
        "alarm_id": 5109,
```

```
        "capture_time": 1716464847,
```

```
        "channel_id": 1,
```

```
        "channel_name": "Camera1",
```

```
        "algo_model": "ObjectIsRecognized",
```

```
        "big_picture_url": "/api/v2/smart/picture?Type=1  
&Index=ubs_5_0_0_0_0_0_751_75_40042_1716464847&Size=304742",
```

```
        "small_picture_url": "/api/v2/smart/picture?Type  
=2&Index=ubs_5_0_0_0_0_0_751_1_40041_1716464847&Size=3841",
```

```
        "gender": 2,
```

```
        "age": 40,
```

```
        "glass": 1,
```

```
        "mask": 1,
```

```
        "compare_list": [
```

```
          {
```

```
            "match_type": 1,
```

```
            "base_image_id": 2005,
```

```
            "name": "5",
```

```
            "similarity": 95,
```

```
            "certificate_no": "",
```

```
            "base_image_url": "/api/v2/smart/picture  
?Type=3&Index=ubs_0_0_0_4_2_1716429337/Pic_1716429337015.jpg&Siz  
e=9999"
```

```
          }
```

```
        ]
```

```
    }  
  ]  
}  
}
```

11.3 行为比对检索

基本信息

Path: /api/v2/search/behaviormatch

Method: POST

接口描述:

请求参数

Headers

参数名称	参数值	必填	示例	备注
Content-Type	application/json	是		

Body

名称	类型	必填	默认值	备注	其他信息
page	integer	是		页码数	
pagesize	integer	是		每页数据量	
channel_id	integer []	是		通道 id	
start_time	integer	是		开始时间，10 位时间戳（秒级）	
end_time	integer	是		结束时间，10 位时间戳（秒级）	
name	string	否		人员姓名	最大长度：64

名称	类型	必填	默认值	备注	其他信息
certificate_no	string	否		证件号，模糊匹配	最大长度：64
algo_model	string []	否		算法类型，送空查全部	
similarity	object	否		相似度，0-100	
└ min	integer	是		最小值	
└ max	integer	是		最大值	

返回数据

名称	类型	必填	默认值	备注	其他信息
code	integer	是		0: success	
msg	string	是			
data	object	否			
└ total	integer	是		总条数	
└ list	object []	是			
└─ alarm_id	integer	是		告警 id	
└─ big_picture_url	string	是		告警大图地址	最大长度：256
└─ algo_model	string	是		告警算法类型	最大长度：128
└─ capture_time	integer	是		告警发生的时间戳，秒级	
└─ channel_id	integer	是		告警所在通道	
└─ channel_name	string	是		通道名称	最大长度：64
└─ object_list	object []	否			

名称	类型	必填	默认值	备注	其他信息
└── base_image_id	integer	否		人脸底图 id	
└── base_image_url	string	否		人脸底图 url	最大长度：256
└── name	string	否		姓名	最大长度：64
└── certificate_no	string	否		证件号	最大长度：64
└── similarity	integer	否		相似度 1-100	
└── match_type	number	是		1：比对成功，2 比对失败	

样例

请求：

```
{  
  "page": 1,  
  "pagesize": 10,  
  "channel_id": [  
    1,  
    2,  
    3,  
    4,  
    5,  
    6,  
    7,  
    8  
  ],  
  "start_time": 1714579199,  
  "end_time": 1716480000,  
  "name": "",  
  "match_type": 999,  
  "certificate_no": "",  
  "algo_model": [  

```

```
    "FieldDetectorObjectsInside",
    "SafetyHelmetAlarm",
    "WorkClothesAlarm",
    "TelephoningAlarm",
    "NoMaskAlarm",
    "FallOverAlarm",
    "OffDutyDetectionAlarm",
    "SmokingAlarm",
    "SleepingDetectionAlarm",
    "ReflectiveClothesDetectionAlarm",
    "PlayMobilePhoneDetection",
    "FireDetection",
    "FumesAlarmBegin",
    "CrowdDensityCriticalAlarm",
    "AbnormalParkingDetection",
    "ChannelBlockageDetection",
    "AccessElevatorAlarm",
    "ObjectRemoved"
  ],
  "similarity": {
    "min": 1,
    "max": 100
  }
}
```

响应:

```
{
  "code": 0,
  "msg": "Succeed",
  "status_code": 0,
  "data": {
    "total": 131,
    "list": [
      {
        "alarm_id": 3741,
        "capture_time": 1716429656,
        "channel_id": 1,
```

```
    "channel_name": "Camera1",
    "algo_model": " NoMaskAlarm",
    "big_picture_url": "/api/v2/smart/picture?Type=1
&Index=ubs_5_0_0_0_0_0_751_48_40622_1716429656&Size=192509",
    "gender": 2,
    "age": 40,
    "glass": 1,
    "mask": 1,
    "object_list": [
      {
        "match_type": 1,
        "base_image_id": 2005,
        "name": "5",
        "similarity": 95,
        "certificate_no": "",
        "base_image_url": "/api/v2/smart/picture
?Type=3&Index=ubs_0_0_0_4_2_1716429337/Pic_1716429337015.jpg&Siz
e=9999"
      }
    ]
  }
}
```

11.4 人脸比对检索

基本信息

Path: /api/v2/search/facematch

Method: POST

接口描述:

请求参数

Headers

参数名称	参数值	必填	示例	备注
Content-Type	application/json	是		

Body

名称	类型	必填	默认值	备注	其他信息
page	integer	是		页码数	
pagesize	integer	是		每页数据量	
channel_id	integer []	是		通道 id	
match_type	integer	否		比对结果 1 成功 2 失败 99 全部	
start_time	integer	是		开始时间，10 位时间戳（秒级）	
end_time	integer	是		结束时间，10 位时间戳（秒级）	
name	string	否		人员姓名，模糊查询 最大长度：50 最小长度：10	最大长度：64
certificate_no	string	否		证件号，模糊查询	最大长度：64
similarity	object	否		相似度，0-100	
└ min	integer	是		最小值	
└ max	integer	是		最大值	
faceattr	object	否			
└ gender	integer	否		性别：999：全部 1：男 2：女 3：未知	
└ age	integer	否		年龄：999：全部 5：儿童 20：少年 40：青年 60：中年 99：老年	
└ glass	string	否		眼睛：999：全部 1：未佩戴眼睛 2：佩戴眼镜	

名称	类型	必填	默认值	备注	其他信息
└ mask	string	否		口罩：999：全部 1：无口罩 2：有口罩	

返回数据

名称	类型	必填	默认值	备注	其他信息
code	integer	是		0: success	
msg	string	是			
data	object	否			
└ total	integer	是		总条数	
└ list	object []	是			
├─ alarm_id	integer	是		告警 id	
├─ big_picture_url	string	是		告警大图地址	最大长度：256
├─ algo_model	string	是		报警算法类型	最大长度：128
├─ capture_time	integer	是		告警发生的时间戳，秒级	
├─ channel_id	integer	是		告警所在通道	
├─ channel_name	string	是		通道名称	最大长度：64
├─ object_list	object []	否			
├─ base_image_id	integer	否		人脸底图 id	
├─ base_image_url	string	否		人脸底图 url	最大长度：256
├─ name	string	否		姓名	最大长度：64
├─ similarity	integer	否		相似度 1-100	
├─ certificate_no	string	否		证件号	最大长度：64

名称	类型	必填	默认值	备注	其他信息
small_picture_url	string	是		告警小图地址	最大长度：256
match_type	number	是		1：比对成功，2：比对失败	

样例

请求：

```
{
  "page": 1,
  "pagesize": 10,
  "channel_id": [
    1,
    2,
    3,
    4,
    5,
    6,
    7,
    8
  ],
  "start_time": 1714579199,
  "end_time": 1716480000,
  "name": "",
  "match_type": 999,
  "certificate_no": "",
  "similarity": {
    "min": 1,
    "max": 100
  }
}
```

响应：

```
{
```

```
"code": 0,
"msg": "Succeed",
"status_code": 0,
"data": {
  "total": 131,
  "list": [
    {
      "alarm_id": 3741,
      "capture_time": 1716429656,
      "channel_id": 1,
      "channel_name": "Camera1",
      "algo_model": "ObjectIsRecognized",
      "big_picture_url": "/api/v2/smart/picture?Type=1
&Index=ubs_5_0_0_0_0_0_751_48_40622_1716429656&Size=192509",
      "gender": 2,
      "age": 40,
      "glass": 1,
      "mask": 1,
      "object_list": [
        {
          "small_picture_url": "/api/v2/smart/pict
ure?Type=2&Index=ubs_5_0_0_0_0_0_751_2_40620_1716429656&Size=478
3",
          "match_type": 1,
          "base_image_id": 2005,
          "name": "5",
          "similarity": 95,
          "certificate_no": "",
          "base_image_url": "/api/v2/smart/picture
?Type=3&Index=ubs_0_0_0_4_2_1716429337/Pic_1716429337015.jpg&Siz
e=9999"
        }
      ]
    }
  ]
}
```


12 平台对接

12.1 新增/修改平台对接信息

基本信息

Path: /api/v2/docking/config/update

Method: POST

接口描述:

请求参数

Headers

参数名称	参数值	必填	示例	备注
Content-Type	application/json	是		

Body

名称	类型	必填	默认值	备注	其他信息
platform	object []	是		平台配置	
└ id	integer	是		对接平台 id。最多支持 2 个平台，0	
└ version_code	string	是		接口版本，“V2.0”。	最大长度：128
└ enabled	integer	是		接口版本。1：开启 0：关闭	
└ url	string	是		告警上报地址	最大长度：256
└ picture_enable	integer	是		上传图片使能开关， 1：开启 0：关闭	

名称	类型	必填	默认值	备注	其他信息
└ video_enable	integer	是		上传短视频使能开关， 1：开启 0：关闭	
└ http_alive	integer	是		http 保活使能开关， 1：开启 0：关闭	
└ alive_interval	integer	是		保活上报间隔，单位是 秒，范围：30-600	
└ channel_enable	integer	是		通道状态上报使能开 关，1：开启 0：关闭	
time_conf	object []	是		上传时间间隔配置	
└ channel_id	integer	是		通道 id	
└ interval	integer	是		时间间隔，单位是分 钟，范围：0-10080	
└ enable	integer	是		间隔使能开关，1：开 启 2：关闭	

返回数据

名称	类型	必填	默认值	备注
code	integer	必须		错误码
msg	string	必须		错误信息

样例

请求：

```
{
  "platform": [
    {
      "id": 1,
      "url": "http://217.1.1.188:8000",

```

```
        "enabled": 1,
        "picture_enable": 0,
        "video_enable": 0,
        "http_alive": 0,
        "alive_interval": 30,
        "channel_enable": 0,
        "version_code": "V2.0"
    },
    {
        "id": 2,
        "version_code": "20240423AIBOX",
        "enabled": 1,
        "url": "http://217.1.1.188:8002",
        "picture_enable": 0,
        "video_enable": 0,
        "http_alive": 0,
        "alive_interval": 30,
        "channel_enable": 0
    }
],
"time_conf": [
    {
        "channel_id": 1,
        "enable": 0,
        "interval": 100
    },
    {
        "channel_id": 2,
        "enable": 0,
        "interval": 200
    },
    {
        "channel_id": 3,
        "enable": 0,
        "interval": 300
    },
    {
        "channel_id": 4,
```

```
        "enable": 0,  
        "interval": 400  
    },  
    {  
        "channel_id": 5,  
        "enable": 1,  
        "interval": 500  
    },  
    {  
        "channel_id": 6,  
        "enable": 0,  
        "interval": 600  
    },  
    {  
        "channel_id": 7,  
        "enable": 1,  
        "interval": 700  
    },  
    {  
        "channel_id": 8,  
        "enable": 1,  
        "interval": 800  
    }  
]  
}
```

响应:

```
{  
    "code": 0,  
    "msg": "Succeed",  
    "status_code": 0  
}
```

12.2 查询平台对接信息

基本信息

Path: /api/v2/docking/config/get

Method: POST

接口描述:

请求参数

Headers

参数名称	参数值	必填	示例	备注
Content-Type	application/json	是		

Body

名称	类型	必填	默认值	备注
----	----	----	-----	----

返回数据

名称	类型	必填	默认值	备注	其他信息
code	integer	是			
data	object	是			
├ time_conf	object []	是		上传时间配置	
├─ channel_id	integer	是		通道 id	
├─ interval	integer	是		时间间隔，单位秒	
├─ enable	integer	是		间隔使能开关，1：开启 0：关闭	
└─ channel_name	string	是		通道名称	最大长度：64
├ platform	object []	是			
├─ id	integer	是		平台 id	

名称	类型	必填	默认值	备注	其他信息
└─ version_code	string	是		接口版本。例如 V1.0:ApiV1.0 V2.0:V2.0	最大长度：128
└─ enabled	integer	是		使能开关，1：开启 0：关闭	
└─ url	string	否		告警上报地址	最大长度：256
└─ picture_enable	integer	是		上传图片使能开关，1：开启 0：关闭	
└─ video_enable	integer	是		上传短视频使能开关，1：开启 0：关闭	
└─ http_alive	integer	是		http 保活使能开关，1：开启 0：关闭	
└─ alive_interval	integer	否		保活上报间隔，单位是分钟	
└─ channel_enable	integer	是		通道状态上报使能开关，1：开启 0：关闭	
msg	string	是			

样例

请求：

响应：

```
{
  "code": 0,
  "msg": "Succeed",
  "status_code": 0,
  "data": {
    "time_conf": [
      {
        "channel_id": 1,
```

```
        "enable": 0,  
        "channel_name": "",  
        "interval": 100  
    },  
    {  
        "channel_id": 2,  
        "enable": 0,  
        "channel_name": "",  
        "interval": 200  
    },  
    {  
        "channel_id": 3,  
        "enable": 0,  
        "channel_name": "",  
        "interval": 300  
    },  
    {  
        "channel_id": 4,  
        "enable": 0,  
        "channel_name": "",  
        "interval": 400  
    },  
    {  
        "channel_id": 5,  
        "enable": 1,  
        "channel_name": "",  
        "interval": 500  
    },  
    {  
        "channel_id": 6,  
        "enable": 0,  
        "channel_name": "",  
        "interval": 600  
    },  
    {  
        "channel_id": 7,  
        "enable": 1,  
        "channel_name": "",
```

```
        "interval": 700
    },
    {
        "channel_id": 8,
        "enable": 1,
        "channel_name": "",
        "interval": 800
    }
],
"platform": [
    {
        "id": 1,
        "url": "http://217.1.1.188:8000",
        "enabled": 1,
        "picture_enable": 0,
        "video_enable": 0,
        "http_alive": 0,
        "alive_interval": 30,
        "channel_enable": 0,
        "version_code": "V2.0"
    }
]
}
}
```

12.3 查询平台对接能力

基本信息

Path: /api/v2/docking/ability/get

Method: POST

接口描述：

请求参数

Headers

参数名称	参数值	必填	示例	备注
Content-Type	application/json	是		

Body

名称	类型	必填	默认值	备注
----	----	----	-----	----

返回数据

名称	类型	必填	默认值	备注	其他信息
code	integer	是			
data	object	是			
├ list	object []	是			
├─ version_code	string	是		版本编码	
├─ version_name	string	是		版本名称	

样例

请求：

响应：

```
{
  "code": 0,
  "msg": "Succeed",
  "status_code": 0,
  "data": {
    "list": [
      {
        "version_code": "V1.0",
        "version_name": "1.0 接口"
      },
      {
        "version_code": "V2.0",
        "version_name": "2.0 接口"
      }
    ]
  }
}
```

13 数据上报

13.1 告警数据上报

基本信息

Path: /custom/router1

Method: POST

接口描述:

请求参数

Headers

参数名称	参数值	必填	示例	备注
Content-Type	application/json	是		

Body

名称	类型	必填	默认值	备注	其他信息
type	integer	是		上报内容 1: 行为分析告警数据 2: 告警恢复数据 3: 人脸抓拍数据 4: 人脸比对数据 5: 行为比对数据 6: 保活数据 7: 点位变更数据	
event_id	string	是		告警 id	
has_result	integer	是		有无比对结果 1 有 0 无	
device_info	object	是		设备信息	

名称	类型	必填	默认值	备注	其他信息
└ device_code	integer	是		设备编码（1-255）	
└ device_name	string	是		设备名称	最大长度：64
└ device_sn	string	是		设备序列号	最大长度：64
└ custom_code	string	否		自定义编码	最大长度：64
channel_info	object	否		点位信息	
└ mode	integer	是		上报模式： 0：点位信息 1：添加点位 2：修改点位 3：删除点位 4：点位状态变化	
└ ipc_sn	string	否		相机序列号	最大长度：64
└ ipc_addr	string	否		相机 ip 地址	
└ channel_id	integer	是		点位 id	
└ channel_name	string	是		点位名称	最大长度：64
└ status	integer	否		点位状态 1：在线 2：离线	
└ custom_code	string	否		自定义编码	最大长度：64
behaviour	object	否		行为分析(告警恢复)数据	
└ algo_model	string	是		算法类型	最大长度：128
└ image_base64	string	否		图片 base64	
└ image_path	string	否		图片相对路径	最大长度：256
└ capture_time	integer	是		抓拍时间	
└ in_num	integer	否		绊线进入人数（瞬时）	
└ out_num	integer	否		绊线离开人数（瞬时）	
└ area_num	integer	否		区域统计人数	

名称	类型	必填	默认值	备注	其他信息
└ video_uuid	string	否		告警视频 uuid	最大长度：256
└ video_url	string	否		告警视频相对路径	最大长度：256
└ capture_info	object []	否		目标信息	
└─ target_id	integer	否		目标 id	
└─ point_x	string	否		目标横坐标信息-- "x0, x1"	
└─ point_y	string	否		目标纵坐标信息-- "y0, y1"	
└─ object_type	integer	否		区域入侵目标类型 1 人 2 机动车 3 非机动车	
└─ plate_no	string	否		车牌号	
face	object	否		人脸抓拍数据	
└ capture_time	integer	是		抓拍时间	
└ point_x	string	否		目标横坐标信息-- "x0, x1"	
└ point_y	string	否		目标纵坐标信息-- "y0, y1"	
└ orig_image_path	string	是		抓拍人脸大图相对路径	最大长度：256
└ crop_image_path	string	是		抓拍人脸小图相对路径	最大长度：256
└ orig_image_base64	string	否		抓拍人脸大图 base64	
└ crop_image_base64	string	否		抓拍人脸小图 base64	
compare_results	object []	否		人脸比对结果（为空代表比对失败）	
└ target_id	integer	是		目标 id	
└ lib_id	integer	是		人员库 id	
└ lib_name	string	是		人员库名称	最大长度：64
└ person_id	integer	是		人员 id	

名称	类型	必填	默认值	备注	其他信息
└ person_name	string	是		人员姓名	最大长度：64
└ sex	integer	否		性别 1：男 2：女 99：未说明	
└ email	string	否		邮箱	最大长度：64
└ tel	string	否		手机号	最大长度：32
└ certificate_type	integer	否		证件类型 1：身份证 2：护照 3：行驶证 99：其它	
└ certificate_no	string	否		证件号码	最大长度：64
└ image_path	string	是		库内人脸图相对路径	最大长度：256
└ image_base64	string	否		库内人脸图 base64	
└ similarity	number	是		相似度	

返回数据

名称	类型	必填	默认值	备注
code	integer	是		
msg	string	是		

样例

请求：

```
{
  "type": 1,
  "event_id": "123456789",
  "has_result": 1,
  "device_info": {
    "device_code": 1,

```

```
    "device_name": "ECS-B501@8HA-SF",
    "device_sn": "210235C8TS5804696123"
  },
  "channel_info": {
    "mode": 0,
    "ipc_sn": "189802C04B7201V00081",
    "ipc_addr": "217.1.0.143",
    "channel_id": 1,
    "channel_name": "Camera1"
  },
  "behaviour": {
    "algo_model": "SafetyHelmetAlarm",
    "image_base64": "/9j/D/2Q==",
    "image_path": "/api/v2/smart/picture?Type=1&Index=ubs_5_0_0_0_0_752_54_63652_1716533613&Size=219148",
    "capture_time": 1716533613,
    "capture_info": [
      {
        "point_x": "4296,4796",
        "point_y": "398,1287",
        "PlateNo": ""
      }
    ]
  }
}
```

响应:

```
{
  "code": 0,
  "msg": "Succeed",
  "status_code": 0
}
```

13.2 告警视频上报

基本信息

Path: /custom/router2

Method: POST

接口描述:

请求参数

Headers

参数名称	参数值	必填	示例	备注
Content-Type	multipart/form-data	是		

Body

参数名称	参数类型	必填	示例	备注
video_uuid	text	是		唯一标识，用于与告警上报的数据关联
file	file	是		告警视频流

返回数据

名称	类型	必填	默认值	备注
code	integer	是		错误码
msg	string	是		错误信息

样例

请求:

响应:

```
{
  "code": 0,
  "msg": "Succeed",
  "status_code": 0
}
```


14 附录

14.1 错误码

● IA57XX@XX

0:	成功	Succeed
1:	通用错误	Common Error
2:	参数非法	Invalid Arguments
3:	用户无权限	Not Authorized
4:	设备不支持	Not Supported
5:	用户状态异常	Abnormal User Status
6:	系统繁忙	System Busy
1000	鉴权模块	
1000:	用户名或密码错误	Incorrect username or passwd
1001:	用户无权限	User does not have permission
1002:	用户登录数已达上限	The maximum number of user logins has been reaches
1003:	用户安全登录 IP 已达上限	The user login ip has reaches the upper limit.
1004:	用户登录 IP 已锁定	User login ip locked.
1100	人员库管理	
1100	人员库数量已达上限	Face lib num has reached the upper limit
1101	存在同名人员库	The face lib name already exists
1200	人员管理	
1200:	人员数量已达上限	The number of faces has reached the max limit
1201:	人员导入库成员证件号类型非法	Wrong ID number type
1202:	人员导入库成员图片格式错误	Image format error
1203:	人员导入库成员图片过大	Image size is too large
1300	点位管理	
1400	点位布防时间	
1500	智能配置	
1500:	算法任务不支持同时配置	Algo type does not support simultaneous config
1600	工服库管理	
1600:	存在同名工作服样式库	The workclothes lib name already exists
1601:	工作服样式库数量已达到上限	Workclothes lib num has reached the upper limit

1700 工服管理

1700: 工作服样式库成员数量已达到上限 The number of workclothes has reached the max limit

1701: 工作服导入库成员图片过大 Image size is too large

1702: 图片导入失败 Importing picture failed

1800 设备管理

1800: 获取设备信息出错 Error in obtaining device information

1801: 设置设备信息出错 Error setting device information

1900 升级

1900: 升级失败 Upgrade failure

1901: 文件校验出错 File verification failure

1902: 设备升级, 版本号不匹配 The version number does not match

1903: 设备升级, 设备类型不匹配 The device type does not match

1904: 设备升级, 可能申请内存或者其他原因失败 Failed to apply for memory or other reasons

1905: 设备升级, 数字签名校验失败 Digital signature verification fails

1906: 升级进度获取失败 Failed to obtain upgrade progress

2000 数据检索

2100 平台对接

2200 数据上报

14.2 适配产品

- IA57XX@XX
- ECS-X5X1
- ECS-50XX

14.3 告警类型

参数说明

时间单位 s

参数	解释	范围
sensitive	灵敏度	1-100
report_rate	告警间隔	≥ 0
direction_line	方向线	/
time_threshold	触发时间	≥ 0
person_num_limit	人数限制	≥ 0
person_float_limit	浮动人数	≥ 0
back_time_threshold	背景刷新时间	≥ 0

ECS-X5X1-SF, ECS-50XX-SF:

算法类型	算法名称	区域类型	灵敏度	告警间隔	方向线	触发时间	人数限制	浮动人数	背景刷新时间	目标类型
SafetyHelmetAlarm	未戴安全帽检测	polygon	√	√						
WorkClothesAlarm	未穿工服检测	polygon	√	√						
TelephoningAlarm	打电话检测	polygon	√	√						
SmokingAlarm	吸烟检测	polygon	√	√						
SleepingDetectionAlarm	睡岗检测	polygon	√	√		√				
OffDutyDetectionAlarm	离岗检测	polygon		√		√	√			
ChannelBlockageDetection	逃生通道堵塞	polygon		√		√				
ObjectRemoved	物品遗失	polygon	√	√		√			√	
FieldDetectorObjectsInside	区域入侵	polygon	√							√
AccessElevatorAlarm	电动车禁入	polygon	√	√						
NoMaskAlarm	未戴口罩检测	polygon	√	√						
FallOverAlarm	跌倒检测	polygon	√	√		√				
CrowdDensityCriticalAlarm	人员数量超限	polygon		√		√	√			
ReflectiveClothesDetectionAlarm	未穿反光服检测	polygon	√	√						
AbnormalParkingDetection	车辆违停占用	polygon		√		√				
FumesAlarmBegin	烟雾检测	polygon	√	√						
PlayMobilePhoneDetection	玩手机检测	polygon	√	√		√				
FireDetection	明火检测	polygon	√	√						
LongStayDetection	逗留检测	polygon		√		√				
FightDetectionAlarm	打架检测	polygon	√	√						

算法类型	算法名称	区域类型	灵敏度	告警间隔	方向线	触发时间	人数限制	浮动人数	背景刷新时间	目标类型
LineDetectorCrossed	越界检测	polygon	√							
EnterArea	区域进入	polygon	√							
LeaveArea	区域离开	polygon	√							
AreaPeopleCounting	区域人数统计	polygon	√							
LinePeopleCounting	绊线人数统计	line	√		√					
ObjectIsRecognized	人脸检测	polygon								
NonMotorAbnormalParkingDetection	非机动车违停	polygon		√		√				

ECS-X5X1-SY:

算法类型	算法名称	区域类型	灵敏度	告警间隔	方向线	触发时间	人数限制	浮动人数	背景刷新时间	目标类型
KnifeStickDetection	手持刀棍检测	polygon	√	√						
SmokingAlarm	吸烟检测	polygon	√	√						
SleepingDetectionAlarm	睡岗检测	polygon	√	√		√				
OffDutyDetectionAlarm	离岗检测	polygon		√		√	√			
ClimbingDetectionAlarm	攀高检测	blend								
FieldDetectorObjectsInside	区域入侵	polygon	√							√
PeopleGathering	人员聚集	polygon				√	√	√		
FastMoving	快速移动	polygon	√	√						
FallOverAlarm	跌倒检测	polygon	√	√		√				
FumesAlarmBegin	烟雾检测	polygon	√	√						
FireDetection	明火检测	polygon	√	√						
LongStayDetection	逗留检测	polygon		√		√				
FightDetectionAlarm	打架检测	polygon	√	√						
LinePeopleCounting	绊线人数统计	line	√		√					
ObjectIsRecognized	人脸检测	polygon								

ECS-X5X1-VK:

算法类型	算法名称	区域类型	灵敏度	告警间隔	方向线	触发时间	人数限制	浮动人数	背景刷新时间	目标类型
ChefHatAlarm	未戴厨师帽检测	polygon	√	√						

算法类型	算法名称	区域类型	灵敏度	告警间隔	方向线	触发时间	人数限制	浮动人数	背景刷新时间	目标类型
SmokingAlarm	吸烟检测	polygon	√	√						
TelephoningAlarm	打电话检测	polygon	√	√						
NoMaskAlarm	未戴口罩检测	polygon	√	√						
ChefClothesDetection	未穿厨师服检测	polygon	√	√						
UncoveredTrashCanDetection	未盖垃圾桶检测	polygon	√	√		√				
ShirtlessDetection	赤膊检测	polygon	√	√						
MouseDetect	鼠患检测	polygon	√	√						
AreaPeopleCounting	区域人数统计	polygon	√							
LinePeopleCounting	绊线人数统计	line	√		√					
ObjectIsRecognized	人脸检测	polygon								

ECS-X5X1-CS:

算法类型	算法名称	区域类型	灵敏度	告警间隔	方向线	触发时间	人数限制	浮动人数	绕行区域	背景刷新时间	目标类型
SafetyHelmetAlarm	未戴安全帽检测	polygon	√	√							
WorkClothesAlarm	未穿工服检测	polygon	√	√							
TelephoningAlarm	打电话检测	polygon	√	√							
SmokingAlarm	吸烟检测	polygon	√	√							
OffDutyDetectionAlarm	离岗检测	polygon		√		√	√				
FieldDetectorObjectsInside	区域入侵	polygon	√								√
FallOverAlarm	跌倒检测	polygon	√	√		√					
CrowdDensityCriticalAlarm	人员数量超限	polygon		√		√	√				
ReflectiveClothesDetectionAlarm	未穿反光服检测	polygon	√	√							
AbnormalParkingDetection	车辆违停占用	polygon		√		√					
ObjectIsRecognized	人脸检测	polygon									
FumesAlarmBegin	烟雾检测	polygon	√	√							
FireDetection	明火检测	polygon	√	√							
NoMaskAlarm	未戴口罩检测	polygon	√	√							
DisorderStackingDetection	物料乱堆放	polygon	√	√		√					
BareSoilCoverDetection	裸土未覆盖	polygon	√	√							
UnwashedVehicleDetection	车辆未清洗	polygon				√					
SafetyHarnessDetection	未穿安全背带	polygon	√	√							

IA57XX@PI:

算法类型	算法名称	区域类型	灵敏度	告警间隔	方向线	触发时间	人数限制	浮动人数	背景刷新时间	告警恢复
LinePeopleCounting	绊线人数统计	line	√		√					
AbnormalParkingDetection	车辆违停占用	polygon		√		√				√
TelephoningAlarm	打电话检测	polygon	√	√						√
FightDetectionAlarm	打架检测	polygon	√	√						√
AccessElevatorAlarm	电瓶车禁入	polygon	√	√						√
FallOverAlarm	跌倒检测	polygon	√	√		√				√
LongStayDetection	逗留检测	polygon		√		√				√
StayAloneDetection	多人作业	polygon		√		√	√			√
FastMoving	快速移动	polygon	√	√						√
OffDutyDetectionAlarm	离岗检测	polygon		√		√	√			√
FireDetection	明火告警	polygon	√	√						√
ClimbingDetectionAlarm	攀高检测	blend								√
AreaPeopleCounting	区域人数统计	polygon	√							
FieldDetectorObjectsInside	区域入侵	polygon	√							√
PeopleGathering	人员聚集	polygon					√			
CrowdDensityCriticalAlarm	人员数量超限	polygon		√		√	√			√
SleepingDetectionAlarm	睡岗检测	polygon	√	√		√				√
ChannelBlockageDetection	逃生通道堵塞	polygon		√		√				√
PlayMobilePhoneDetection	玩手机检测	polygon	√	√		√				√
WorkClothesAlarm	未穿工服检测	polygon	√	√						√
SafetyHelmetAlarm	未戴安全帽检测	polygon	√	√						√
NoMaskAlarm	未戴口罩检测	polygon	√	√						√
SmokingAlarm	吸烟检测	polygon	√	√						√
FumesAlarmBegin	烟雾告警	polygon	√	√						√
LineDetectorCrossed	越界检测	line	√		√					

IA57XX@UM:

算法类型	算法名称	区域类型	灵敏度	告警间隔	方向线	触发时间	人数限制	浮动人数	背景刷新时间	告警恢复
ExposedGarbageDetection	暴露垃圾	polygon	√	√		√				
OutsideTheStoreBusinessDetection	出店经营	polygon	√	√		√				

算法类型	算法名称	区域类型	灵敏度	告警间隔	方向线	触发时间	人数限制	浮动人数	背景刷新时间	告警恢复
PackedGarbageDetection	打包垃圾	polygon	√	√		√				
NonMotorAbnormalParkingDetection	非机动车违停	polygon	√	√		√				√
SharedBicyclesRandomParkingDetection	共享单车无序停放	polygon	√	√		√				
FloatingObjOnRiverDetection	河道漂浮物	polygon	√	√						
AbnormalParkingDetection	机动车违停	polygon	√	√		√				√
WasteTruckDetection	禁止渣土车	polygon	√	√						√
TrashOverflowingDetection	垃圾桶满溢	polygon	√	√		√				
UncoveredTrashCanDetection	垃圾桶未盖	polygon	√	√		√				
SetFileOnTheRoadDetection	路口焚烧	polygon	√	√						√
DirtyRoadDetection	路面不洁	polygon	√	√		√				
SurfaceWaterDetection	路面积水	polygon	√	√		√				
DogDetection	犬类识别	polygon	√	√						
PeopleGathering	人群聚集	polygon					√			
IllegalUseOfAnUmbrellaDetection	违规撑伞	polygon	√	√		√				
IllegalBillboardsDetection	违规广告牌	polygon	√	√						
DisorderStackingDetection	物料堆积	polygon	√	√		√				
DryingOnTheStreetDetection	沿街晾晒	polygon	√	√						
StreetVendorDetection	游商	polygon	√	√		√				
RoadsideStallBusinessDetection	占道经营	polygon	√	√		√				