

HTTP 对接协议

V2.0

目录

目录

HTTP 对接协议	1
目录	2
更新说明	4
工作流程：	5
1 设备发送	6
1.1 taskRequest	6
1.2 gpsReport	6
1.3 alarmReport	7
2 服务器返回的任务	9
1. 添加更新用户：addUser / updateUser	9
2. 删除用户：deleteUser	11
3. 同步时间：setDevTime	12
4. 任务空闲：idleTime	12
5. 设置屏保图片：setScreenSave	13
6. 远程升级 upgrade	14
7. 获取日志 getLogFile	14
8. 获取设备状态 getDevStatus	15
9. 删除用户列表：delUserList	16
10. 设置断网续传 setResumeTransfer	17
11. 设置服务器地址 setPlatformServer	18
12. 抓拍命令 faceCmd	19
13. 人脸参数 SetFaceConfig	20
3 任务完成结果	24

4 抓拍记录上传.....	25
---------------	----

更新说明

版本	更新时间	内容
V0.9	2020.08.22	初始版本
v1.0	2020.11.30	1. adduser 新增 group, user_type, 更改 image_type 类型 2. 抓拍上传新增 门禁卡号 3. 没有新任务时, 服务器返回 idleTime
v1.1	2020.12.3	1. 新增屏保设置, 2. 新增 updateUser
v1.2	2020.12.28	1.新增删除用户列表
v1.2.2	2021.6	1.推送增加人脸角度及 ID; 增加断网续传开关
v1.2.3	2021.7	1. 增加 gps 信息上传 (gpsReport) 2. 增加修改服务器接口 (setPlatformServer)
v1.3	2021.8	完善部分接口信息
V1.4	2023.2	新增机动车报警信息上传
V1.6	2023.2	完善报警信息上传
V1.6	2023.2	报警信息更改设备 ip 属性上传
V1.7	2023.4	新增抓拍命令
V1.8	2023.4	补充结构化抓拍上传信息
V1.9	2023.4	补充人脸参数设置
V2.0	2023.4	补充性别年龄信息说明

工作流程：

设备启动后向服务器发送 taskRequest，服务器收到后返回指令到设备，如：添加，删除人脸，同步设备时间等，如果没有设备需要操作的指令，服务器返回空闲指令(idleTime)。设备完成后发送完成结果到服务器。设备如果收到“空闲指令”消息，则等 20 秒后再次发送 taskRequest，否则 5 秒发送下一个 taskRequest

1 设备发送

1.1 taskRequest

说明：设备发送到服务器的任务请求（POST）

方法	URL		
POST	http://server_url/taskRequest		
参数	定义	必须	备注
device_id	设备 ID	Y	String,
device_mac	设备有线 mac 地址	Y	string
device_name	设备名称	Y	string
time_stamp	UTC 时间戳	Y	number
software_version	主程序版本	Y	string
lib_platform_version	平台库版本	Y	string
lib_ai_version	ai 库版本	N	string

JSON 格式

```
{
  "device_id":"ID12345678901234",
  "time_stamp": 1577690264
  "software_version":"1.020.1.1_MAIN_V28.1(20210309)"
  "lib_platform_version":"v3.5"
}
```

1.2 gpsReport

说明：上传 gps 位置信息，时间间隔超过 5 秒，或者移动距离超过 20 米，上报一次

方法	URL		
POST	http://server_url/gpsReport		
参数	定义	必须	备注
device_id	设备 ID	Y	String,
device_mac	设备有线 mac 地址	Y	string
time_stamp	UTC 时间戳	Y	number, 卫星返回的时间戳
longitude	经度	Y	number
latitude	纬度	Y	number
distance	距离上一次上报位置的 距离	Y	number, 单位米
status	gps 状态	Y	number

JSON 格式

```
{
  "device_id": "ID12345678901234",
  "time_stamp": 1577690264,
  "longitude": 114.0859,
  "latitude": 22.5,
  "distance": 30
}
```

1.3 alarmReport

说明：上传报警信息

方法	URL		
POST	http://server_url/deviceAlarm		
参数	定义	必须	备注

device_ip	设备 IP	Y	
device_id	设备 ID	Y	
device_mac	设备有线 mac 地址	Y	
device_name	设备名称	Y	
lib_platform_version	平台库版本	Y	
software_version	软件版本	Y	
lib_ai_version	ai 库版本	Y	
time_stamp	时间戳	Y	时间戳单位:秒
alarm_status	状态	Y	number, 报警输入状态, 报警消失: 0, 触发报警: 1,
alarm_type	报警类型	Y	Number, 移动侦测: 0, 输入报警 1: 1, 输入报警 2: 9, 输入报警 3: 10, 输入报警 4: 11, 机动车报警: 12, 区域入侵报警: 8,
image	抓拍图片数据	N	Base64, 仅在机动车报警时上传

JSON 格式

```
{
  "device_ip": "192.168.1.10",
  "device_id": "ID12345678901234",
  "device_name": "test",
  "device_mac": "88-07-18-00-01-71",
  "lib_platform_version": "platform_l v4.5.4",
  "software_version": "10.001.11.1_MAIN_V3.4(220802)",
  "lib_ai_version": "ai v2.0.2_20220228",
  "alarm_status": 1,
```

```

    "alarm_type":9,
    "time_stamp":1577690264,
    "image":"...",
}

```

2 服务器返回的任务

1 . 添加更新用户：addUser / updateUser

说明：adduser 是添加用户，如果用户 id 已存在则添加失败，updateUser 如果 id 已存在则更新信息，如果不存在则添加用户

参数	定义	必须	备注
request_type	addUser/updateUser	Y	string, 服务器下发任务
request_id	命令 id	Y	number, 每条消息对应的唯一 id, 由服务器生成
user_list	用户	N	array, 用户列表
user_list	用户类型	N	number, 用户列表中单个用户的分组, 1: 黑名单, 2: 白名单, 默认白名单
user_id	人脸 id	Y	string
image_type	图片类型 ● image ● URL	N	string
image_content	人脸数据	N	当 image_type 为 "URL" 时, image_content 为 人脸图片地址, 当 image_type 为 "image" 时, image_content 为 人脸图片数据 (base64 编码)
user_info	● name: 用户名 ● phone_number: 电话号码 ● sex 性别 ● other: 其他信息	Y	用户资料, 长度 256B ● name: string ● phone_number: string ● sex: Number;0 表示女性, 1

			表示男性 ● other: string
access_card	门禁卡	N	number 门禁卡
group	用户组	N	number 最多支持 20 个用户组 (0-19)
user_type	用户类型	N	number, 0: 人脸用户, 1: 门禁卡用户 (不含人脸), 默认添加人脸用户

JSON 格式

```
{
  "request_id":100002000,
  "request_type":"addUser"
  "user_list":
    [
      {
        "user_id":"10086",
        "image_type  ": "URL"
        "image_content":"http://abc.com/face.jpg",
        "user_info": {
          "name":"ls",
          "phone_number":"12345"
          "sex":1,
          "other":"teacher"
        }
        "access_card":1200
        "group":0,
        "user_type":1      //门禁卡用户
        "user_list": 1    //黑名单用户
      }
    ]
}
```

```
        "user_id":"10010",
        "user_info": {
            "name":"zx",
            "phone_number":"12345"
            "sex":1,
            "other":"student"
        },
        "access_card":12001,
        "group":1,
    }
]
}
```

2. 删除用户: deleteUser

参数	定义	必须	备注
request_type	deleteUser	Y	string, 服务器下发任务
request_id	命令 id	Y	number, 每条消息对应的唯一 id, 由服务器生成
user_id	人脸 id	Y	string

JSON 格式

```
{
    "request_id":100002001
    "request_type":"deleteUser"
    "user_id":"10086"
}
```

3. 同步时间：setDevTime

参数	定义	必须	备注
request_id	命令 id	Y	number, 每条消息对应的唯一 id, 由服务器生成
request_type	setDevTime	Y	string, 服务器下发任务
time_stamp	UTC 时间戳	Y	Number

JSON 格式

```
{
  "request_id":10002200,
  "request_type":"setDevTime",
  "time_stamp":1577690264, //(UNIX 时间戳:秒)
}
```

4. 任务空闲：idleTime

说明：当服务器没有任务时，返回空闲命令

参数	定义	必须	备注
request_type	idleTime	Y	任务空闲
time_stamp	UTC 时间戳	Y	Number

JSON 格式

```
{
  "request_type":"idleTime",
  "time_stamp":1577690264, //(UNIX 时间戳:秒)
}
```

5 . 设置屏保图片：setScreenSave

说明：可以设置最多 5 张屏保图片，系统会用现有屏保图片循环播放

参数	定义	必须	备注
request_type	setScreenSave	Y	string, 服务器下发任务
request_id	命令 id	Y	number, 每条消息对应的唯一 id, 由服务器生成
enable	使能屏保功能	Y	number
display_time	每天屏保时间段	N	string 如: 07:00-12:00
interval_time	图片切换间隔时间	N	number, 单位秒
start_time	人离开后进入屏保时间	N	number, 单位秒, 范围 10-600
image_1	屏保图片 1	N	string, 屏保图片的 url 地址, png 格式图片, 目前仅支持 http 地址, 如果为空表示删除该张屏保
image_2	屏保图片 2	N	string
image_3	屏保图片 3	N	string
image_4	屏保图片 4	N	string
image_5	屏保图片 5	N	string

JSON 格式

```
{
  "request_type":"setScreenSave",
  "request_id": 123456,
  "enable": 1,
  "display_time": "08:00-23:59",
  "interval_time":10,
```

```

    "start_time": 8,
    "image_1": "http://abc.com/sc1.png"           //设置屏保 1
    "image_2": ""                                   //删除屏保 2
    "image_3": "http://abc.com/sc3.png"           //设置屏保 3
}

```

6 . 远程升级 upgrade

说明：远程升级

参数	定义	必须	备注
request_type	upgrade	Y	string, 服务器下发任务
request_id	命令 id	Y	number, 每条消息对应的唯一 id, 由服务器生成
URL	升级文件路径	Y	string, 服务器上的升级文件地址

JSON 格式

```

{
    "request_id": 10002200,
    "request_type": "upgrade",
    "URL": "http://abc.com/test.img"
}

```

7 . 获取日志 getLogFile

说明：远程升级

参数	定义	必须	备注
request_type	getLogFile	Y	string, 服务器下发任务
request_id	命令 id	Y	number, 每条消息对应的唯一 id, 由服务器生成

			— id, 由服务器生成
log_name	日志文件名	Y	string, 服务器上的日志文件名

JSON 格式

```
{
  "request_id":10002200,
  "request_type":"getLogFile",
  "log_name":"abc.log"
}
```

设备返回(http://server_url/taskResult):

```
{
  "device_id": "ID12345678901234",
  "request_id": 100002000,
  "code": 0,
  "log":"'getLogFile' success!",
  "log_info":"this is log ....."
}
```

8 . 获取设备状态 getDevStatus

说明：远程升级

参数	定义	必须	备注
request_type	getDevStatus	Y	
request_id	命令 id	Y	number, 每条消息对应的唯一 id, 由服务器生成

JSON 格式

```
{
```

```
"request_id":10002200,
"request_type":"upgrade",
}
返回
{
  "device_id": "ID12345678901234",
  "request_id": 100002000,
  "code": 0,
  "log":"'getDevStatus' success!",
  "total_face":2000,
  "main_ver":"1.02.v23",
  "platform_ver":"v24",
  "web_ver":"v24",
  "ai_version":"v1.5"
}
```

9. 删除用户列表：delUserList

参数	定义	必须	备注
request_id	命令 id	Y	number，每条消息对应的唯一 id，由服务器生成
request_type	delUserList	Y	
user_list	人脸 id	Y	Number, 1:黑名单, 2:白名单, 目前仅支持白名单

JSON 格式

```
{
  "request_id":100003001,
  "request_type":"delUserList",
  "user_list": 2
}
```

返回: (http://server_url/taskResult):

```

{
  "device_id": "ID12345678901234",
  "request_id": 100002000,
  "resp_list":
    [
      {
        "user_id": "10086",
        "code": 0,
      }
      {
        "user_id": "10011",
        "code": -200,
      }
    ],
  "code": 0
}

```

10 . 设置断网续传 setResumeTransfer

说明：远程升级

参数	定义	必须	备注
request_type	setResumeTransfer	Y	
request_id	命令 id	Y	number，每条消息对应的唯一 id，由服务器生成
enable	使能	Y	number,是否使能断网续传

JSON 格式

```

{
  "request_id": 10002200,
  "request_type": "setResumeTransfer",

```

```
    "enable": 0
  }
  返回 ( http://server_url/taskResult):
  {
    "device_id": "ID12345678901234",
    "request_id": 100002000,
    "code": 0,
    "log":"'setResumeTransfer' success!"
  }
```

11 . 设置服务器地址 **setPlatformServer**

说明：远程升级

参数	定义	必须	备注
request_type	setPlatformServer	Y	
request_id	命令 id	Y	number，每条消息对应的唯一 id，由服务器生成
serverAddr	http 服务器地址	N	string，http 服务器地址，格式：abc.com:90/tetest
wsServerAddr	websocket 服务器地址	N	string，ws 服务器地址，格式：ws.abc.com:90

JSON 格式

```
{
  "request_type":"setPlatformServer",
  "request_id":10002200,
  "serverAddr":"abc.com:90/test"
  "wsServerAddr":"abc.com:90"
}
```

设备返回:

```
{
  "resp_type": "setPlatformServer",
  "request_id": 10002200,
  "code": 0,
  "log": "'setPlatformServer' success!",
}
```

12 . 抓拍命令 faceCmd

说明: 抓拍命令

参数	定义	必须	备注
request_type	faceCmd	Y	
request_id	命令 id	Y	number, 每条消息对应的唯一 id, 由服务器生成

JSON 格式

```
{
  "request_id": 10002200,
  "request_type": "faceCmd"
}
```

返回 (http://server_url/taskResult):

```
{
  "device_id": "ID12345678901234",
  "request_id": 100002000,
  "code": 0,
  "log": "'faceCmdr' success!"
}
```

13 . 人脸参数 SetFaceConfig

说明：抓拍命令

参数	定义	必须	备注
request_type	SetFaceConfig	Y	可以通过 GetFaceConfig 获取当前值
request_id	命令 id	Y	number, 每条消息对应的唯一 id, 由服务器生成
FaceQuality	人脸灵敏度	N	number, max:10, min:1
FaceTrackTenable	人脸框开启	N	number, open:1, close:0
MaskDetectEnable	口罩检测开启	N	number, open:1, close:0
FaceMiniPixel	人脸最小像素	N	number, max:300, min:30
FaceMaxPixel	人脸最大像素	N	number, max:700, min:300
DaEnable	人脸检测区域启用	N	number, open:1, close:0
DetectAreaX	人脸检测区域坐标 x	N	number, max:1920, min:0
DetectAreaY	人脸检测区域坐标 y	N	number,

			max:1920, min:0
DetectAreaW	人脸检测区域坐标 w	N	number, max:1920, min:0
DetectAreaH	人脸检测区域坐标 h	N	number, max:1920, min:0
LivenessEnable	活体检测模式	N	number, 0:关闭, 1:彩色活体检测,
LivenessThreshold	活体阈值	N	number, max:0.99, min:0
SnapMode	抓拍模式	N	number, 0: '离开后抓拍', 1: '快速+离开后抓拍', 2: '间隔抓拍-秒单位', 3: '间隔抓拍-帧为单位', 5: '离开后抓拍-识别优先',
IntervalFrame	间隔帧数	N	number, max:1500, min:10
IntervalTime	间隔时间	N	number, max:30, min:1
SnapNum	抓拍数量	N	number, 0:无限制, max:16, min:1

UploadMode	本地图片保存	N	number, 0: '人脸图', 1: '背景图', 2: '人脸+背景图', 3: '人脸+人形图', 4: '人脸+人形+背景图', 5: '人形图',
ChooseMode	选优模式	N	number, 0: '质量选优', 1: '角度选优',
Yaw	人脸左右偏转角	N	number, max:100, min:0
Pitch	人脸上下俯仰角	N	number, max:100, min:0
Roll	人脸头歪翻滚角	N	number, max:100, min:0
FacePicQuality	人脸图质量	N	number, max:99, min:0
PicQuality	场景图质量	N	number, max:90, min:0
SnapFaceArea	人脸灵敏度	N	number, max:10, min:0
MultiFace	人脸属性开关	N	number, open:1, close:0

BodyQuality	人形编码质量	N	number, max:90, min:0
BodyAreaEx	人形扩展系数	N	number, max:10, min:0
ExposureMode	曝光模式	N	number, 0: '普通场景', 1: '逆光场景',
PicUploadMode	图片上传类型	N	Number,bit, 0001: '人脸图', 0010: '人形图', 0100: '背景图', Eg. 人脸图+背景图: 0101 传 5

JSON 格式

```
{
  "request_id":10002200,
  "request_type":"SetFaceConfig",
  "FaceQuality": 5,
  "FaceTrackTnable": 1,
  "MaskDetectEnable": 0,
  "FaceMiniPixel": 80,
  "FaceMaxPixel": 700,
  "DaEnable": 0,
  "DetectAreaX": 0,
  "DetectAreaY": 0,
  "DetectAreaW": 1920,
  "DetectAreaH": 1080,
  "LivenessEnable": 0,
  "LivenessThreshold": 0.98,
```

```

    "SnapMode": 0,
    "IntervalFrame": 25,
    "IntervalTime": 2,
    "SnapNum": 1,
    "UploadMode": 0,
    "ChooseMode": 0,
    "Yaw": 30,
    "Pitch": 30,
    "Roll": 30,
    "FacePicQuality": 99,
    "PicQuality": 40,
    "SnapFaceArea": 3,
    "MultiFace": 0,
    "BodyQuality": 80,
    "BodyAreaEx": 0,
    "ExposureMode": 0,
    "PicUploadMode": 1
}
返回 ( http://server_url/taskResult):
{
    "device_id": "ID12345678901234",
    "request_id": 100002000,
    "code": 0,
    "log": "'SetFaceConfig' success!"
}

```

3 任务完成结果

说明：设备完成任务返回结果到服务器

方法	URL		
updateResult	http://server_url/taskResult		
名称	定义	必须	备注

device_id	设备 ID	Y	String
request_id	命令 id	Y	number, 服务器传回的消息 ID, 表明该结果对应的服务器任务
code	失败代码	Y	number, 0 表示成功, 否则表示失败

4 抓拍记录上传

说明:

设备识别人脸后, 向服务器 POST 一条识别记录。

方法	URL		
faceRecognition	http://server_url/faceRecognition		
名称	定义	必须	备注
device_id	设备 ID	Y	
IP	设备 IP	Y	一般为设备局域网 ip
frpic_name	抓拍图片名称	Y	string, 服务器收到后回传给设备, 服务器可保存此信息用于后期查询相应抓拍图片
snap_time_ms	UTC 时间戳	Y	Number, 抓拍时间 (毫秒)
user_id	用户 id	N	string, 比对不通过时为空
user_name	用户名	N	string
group	用户组	N	number, 用户所在的用户组
comp_score	相似度	N	number
user_list	用户列表	N	number, 1 黑名单, 2 白名单
user_temperature	人体温度	N	number
image	抓拍人脸	Y	Base64 编码的人脸图片

user_name2	用户别名	N	string
issuer	证件签发地址	N	string,
idcard_number	证件号码	N	string
idcard_image	证件上的预留图片	N	string, Base64 编码的人脸图片
fullview_image	全景图片	N	string, Base64 编码的抓拍图片, 需要配置为背景图上传模式
body_image	人形图片	N	string, Base64 编码的人形图片
face_yaw	人脸左右侧脸角度	N	number, -100, 100,数值越大测斜角度越大
face_pitch	人脸上下俯仰角度	N	number, -100, 100, 同上
face_roll	人脸歪斜角度	N	number, -100, 100, 同上
face_id	人脸 id	N	number, 一直保持在画面中的同一张人脸id不变,人脸出画面,或者人脸改变后 id 改变
qr_string	二维码字符串	N	string, 扫描二维码得到的字符串
eva_age	算法估算年纪	N	number
sex	性别	N	number, 0 男, 1 女
eva_sex	算法估算性别	N	string
device_ip	设备 IP	Y	
timestamp	UTC 时间戳	Y	Number,命令时间戳
snap_time	UTC 时间戳	Y	Number, 抓拍时间 (秒)
mask	口罩	N	number, 0 无口罩, 6 有口罩
access_card	卡号	N	number
face_quality	人脸质量分数	N	Number, 0~9999
car_label	结构化抓拍类型	Y	Number, 1: '人脸',

			2: '行人', 3: '骑行者(人+非机动车)', 4: '机动车', 5: '非机动车', 6: '头肩', 7: '垃圾'
car_type	车类型	Y	Number, 1: '摩托车', 2: '电动车', 3: '自行车', 4: '三轮车', 5: '小汽车', 6: '面包车', 7: '小卡车', 8: '大卡车', 9: 'SUV', 10: '大巴', 32: '其它',
car_color	车颜色	Y	1: '灰色', 2: '白色', 3: '红色', 4: '黑色', 5: '蓝色', 6: '绿色', 7: '棕色', 8: '紫色', 9: '粉色', 10: '黄色', 32: '其它',
car_number	车牌	N	String,识别车牌成功时携带发送
car_brand	车品牌	N	String,识别车品牌成功时携带发送

JSON 格式:

抓拍上传:

```
{
  "IP": "192.168.1.92",
  "device_ip": "192.168.1.92",
  "device_id": "T01000114010100010126",
  "timestamp": 1678847868,
  "snap_time": 1678847868,
  "snap_time_ms": 1678847868887,
  "frpic_name": "FACE_0_20230315103748887_21.jpg",
  "user_list": 2,
  "mask": 0,
  "user_name": "1676511938219",
  "user_temperature": 36.6,
  "user_id": "131",
  "access_card": 1,
  "group": 0,
  "comp_score": 98.192848,
  "sex": 0,
  "qr_string": "qr info",
  "image": "原图图片数据 base64",
  "idcard_number": "440523199001011234",
  "face_yaw": -1,
  "face_pitch": -7,
  "face_roll": -9,
  "face_id": 57,
  "face_quality": 7458,
  "eva_age": 36,
  "eva_sex": "male",
  "car_label": 0,
  "car_type": 0,
  "car_color": 0
}
```

```
}
```

服务器返回：

接收成功

```
{  
    "code":0,  
    "snap_time":1577690264, //设备推送消息时的时间戳  
    "log": "'faceRecognition' success"  
}
```

失败

```
{  
    "code":1,  
    "snap_time":1577690264, //设备推送消息时的时间戳  
    "log": "'faceRecognition' failed" //错误日志  
}
```