

# 局域网平台服务

局域网（离线版）系统接口说明

邮编：310012

电话：4000686857

地址：杭州市余杭区文一西路 1217 号 IT 公园 A 幢 1 号楼 10-13 层

用有思想的技术，创造无处不在的美好

# 目录

1 接口说明	4
1.1 接口规范	4
1.2 接口返回	4
1.3 接口调用流程	5
1.4 注意事项	6
2 设备管理类接口	7
2.1 设置设备密码	7
2.2 设备序列号获取	9
2.3 设备配置	10
2.3.1 识别成功参数	12
2.3.2 陌生人参数	13
2.3.3 识别参数	14
2.3.4 其他参数	16
2.4 识别模式配置	18
2.5 修改 Logo	22
2.6 修改屏幕方向	23
2.7 设置设备时间	24
2.8 有线网络配置	26
2.9 Wi-Fi 配置	28
2.10 设备重启	31
2.11 设备重置	32
2.12 设备开门	34
2.13 大屏模式接口	35
2.14 大屏背景图片配置	37
2.15 识别回调	39
2.16 注册照片回调	43
2.17 设备心跳回调	46
3 人员管理类接口	48
3.1 人员注册	48
3.2 人员卡号注册	50
3.3 人员更新	52
3.4 人员删除（批量）	54
3.5 人员分页查询	56
3.6 人员信息查询	61
3.7 时间段权限设置	62
3.8 时间段权限删除	66
3.9 人员有效期限设置	67
3.10 有效期限设置删除	69
4 照片管理类接口	70
4.1 照片注册（base64）	70
4.2 照片注册（url）	72
4.3 拍照注册	74

4.4 照片查询	76
4.5 特征注册	78
4.6 照片更新	86
4.7 照片删除	88
4.8 清空人员注册照片	89
4.9 人脸照片相似性比对	91
5 识别记录管理接口	93
5.1 识别记录查询	93
5.2 识别记录删除	96
5.3 识别记录删除（通过 Unix 毫秒级时间戳删除）	98
5.4 刷脸记录查询	100
5.5 刷脸记录删除	104
5.6 刷脸记录删除（通过 Unix 毫秒级时间戳删除）	105
5.7 刷卡记录查询	106
5.8 刷卡记录删除	108
5.9 刷卡记录删除（通过 Unix 毫秒级时间戳删除）	109

## 1 接口说明

### 1.1 接口规范

**接口根地址：**http://设备 ip 地址:8090/

**接口形式：**通过 HTTP 请求的方式对外提供服务。

**接口安全：**初次调用接口需要先**设置设备密码**（pass），后续调用任何接口都需要传入设备密码（pass）作为接口安全校验秘钥。

### 1.2 接口返回

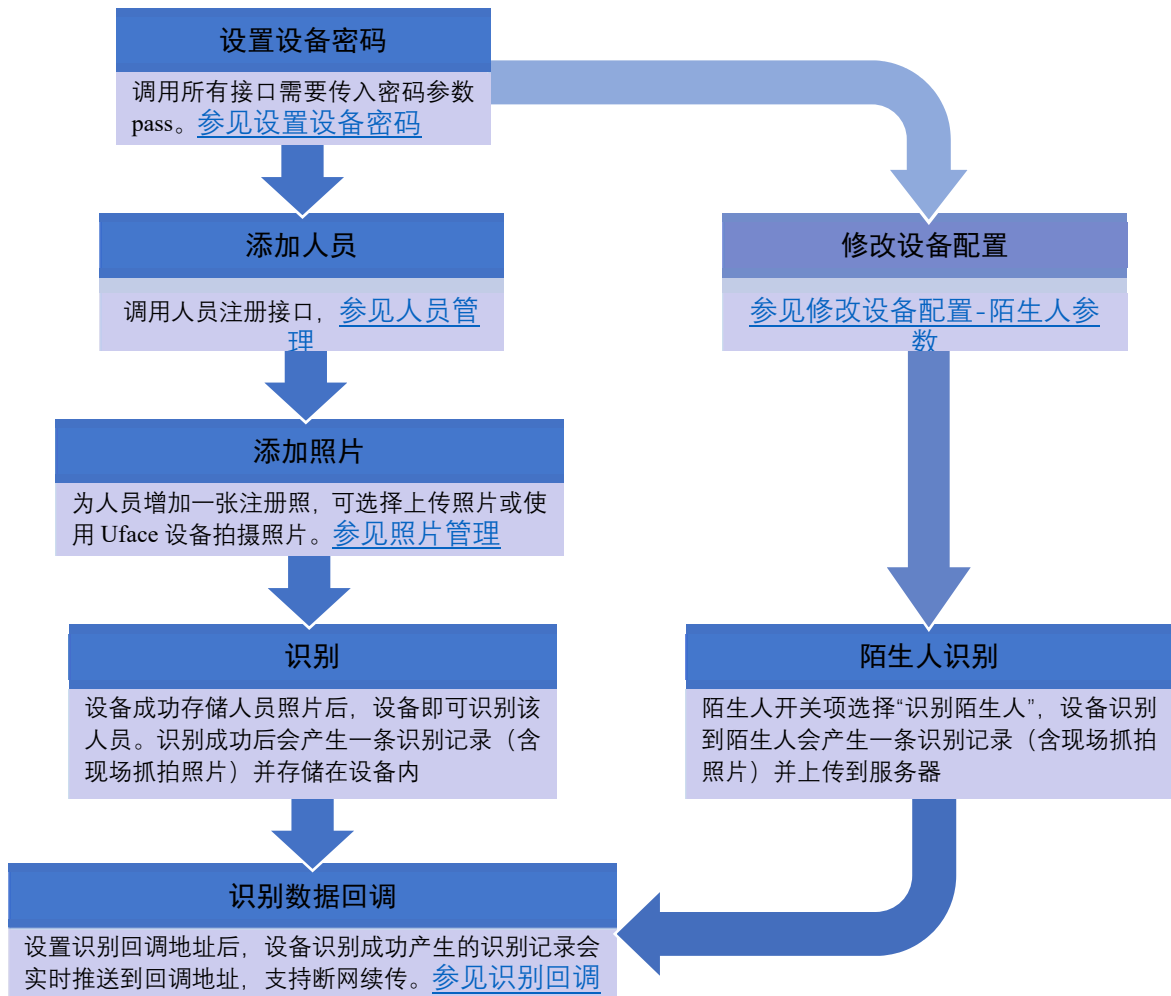
**接口通用返回说明：**

```
public class ResultInfo<T> {  
    private Int result;//表示接口是否调通，1 成功，0 失败，通常只要设备服务器能响应，该值均为 1  
    private Boolean success;//此次操作是否成功，成功为 true，失败为 false  
    private T data;//接口返回的业务数据，类型可为数值、字符串或集合等  
    private String msg;//接口返回的信息，通常是错误类型码的原因信息  
}
```

文档中所涉及到的接口返回示例，个别接口的返回数据会有略微调整，须以真实的返回结果为准。

### 1.3 接口调用流程

**首次使用设备：**设备开机后进入识别主界面。设备未调通过接口，设备无初始密码。操作流程见流程图。

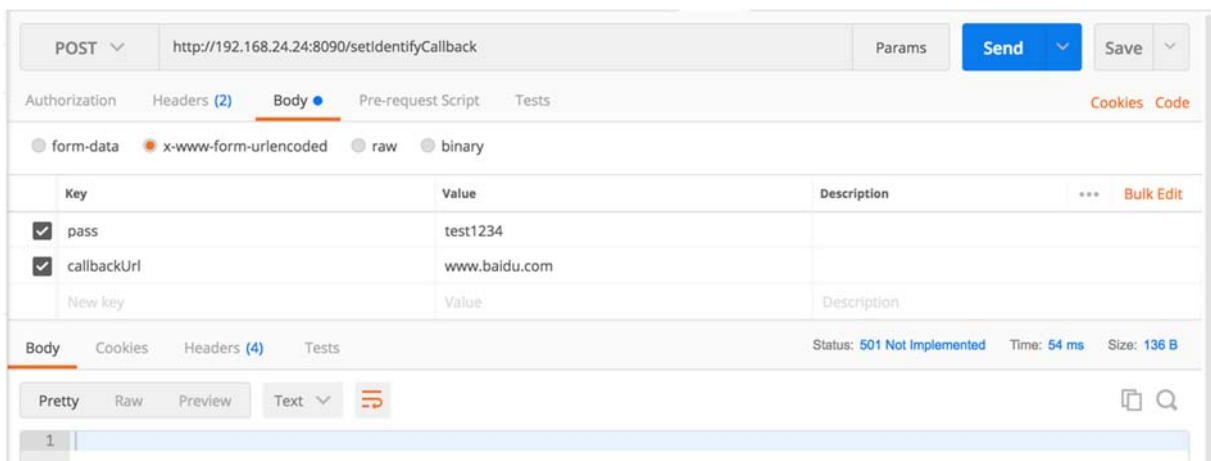


- 设备初始无密码，调用所有接口都需要传入密码参数；在设置设备密码接口，newPass 和 oldPass 传入一样的值，即成为设备密码
- 照片注册成功后，该注册照片会保存在设备内；可通过照片查询接口获得人员注册照片 url，访问 url 的照片时，设备需连网在线，且客户端与设备处于同一网段
- 人脸识别成功后，会显示注册人员的姓名；若填写了卡号，可直接刷对应卡号的卡进行识别，识别成功后会显示与该卡号对应的注册人员姓名

- 若设置了回调地址，人员识别成功后会向回调地址 POST 字段 ip、personId、time（识别时间）、deviceKey 和 type: face/card\_0/1；若设置了公网回调地址，识别时设备离线或设备连局域网，则设备连公网后识别记录会上传到公网回调地址

## 1.4 注意事项

- 1、调用设备接口时，请勿同时在其他客户端服务器上调用同一台设备的接口。
- 2、若调用接口返回提示“参数异常”，请按以下步骤检查：
  - a) 检查参数名称是否拼错、参数名称中是否带有空格或回车
  - b) 检查参数值是否符合规范，如 Int 值未定义、id 含有数字字母以外的非法字符等
  - c) Json 中是否含有空格（直接使用接口返回的 Json 字符串需要去空格），是否多了或少了"，请仔细核对每个接口的 postman 示例
- 3、若调用接口返回为空，表示 url 错误（IP 有误、拼写错误、字段遗漏等），如图所示。下图中 url 大小写有误。



- 4、若调用接口返回 Could not get any response，可能原因有：IP 地址错误、端口号错误或遗漏、传参形式错误（POST 请求参数放在 body 里，格式为 x-www-urlencodcd）等。


## 2 设备管理类接口

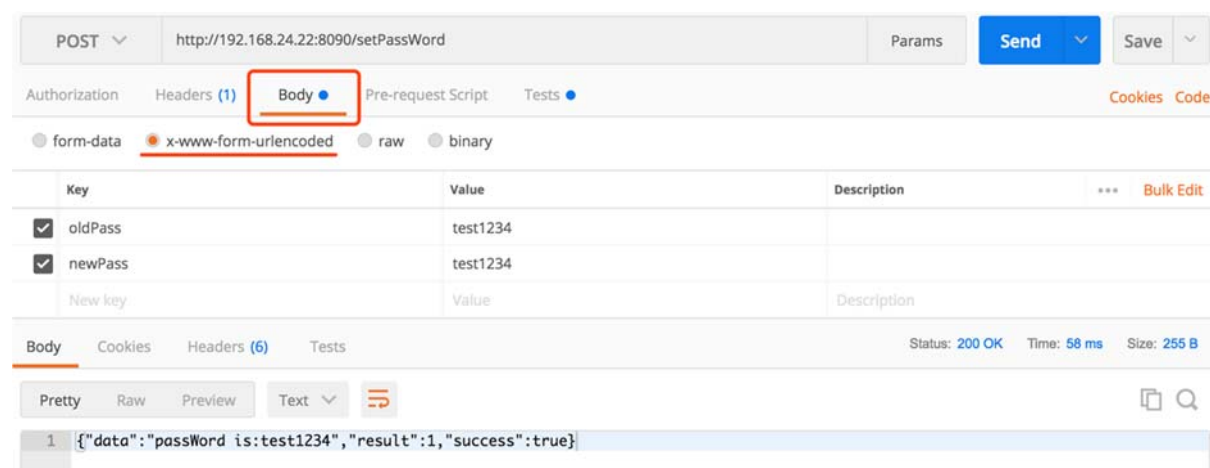
### 2.1 设置设备密码

#### 请求数据

Method	URL			
POST	http://设备 IP:8090/setPassWord			
参数名	描述	类型	必传	附加说明
oldPass	旧密码	String	Y	<ul style="list-style-type: none"><li>• 新设备或重置后（恢复初始化）的设备，调用其他接口前，需要先进行初始密码设置，oldPass 和 newPass 传入一样的值即可</li><li>• 修改密码时，分别传入新、旧密码</li><li>• 此接口调用不需要传入 pass</li><li>• 密码不可为空或空格</li></ul>
newPass	新密码	String	Y	

#### 设置初始密码示例

-  注意：a) POST 请求参数放在 body 里
- b) ContentType: application/x-www-form-urlencoded
- c) 本章所有示例均使用 postman



## 返回示例


```
{  
  "data": "passWord is:12345", //设备密码，也称接口调用密码，请妥善保管。若忘记密码，则需  
  重置设备，设备会清空所有数据  
  "result": 1, //接口调通  
  "success": true //设备密码设置成功  
}
```



## 2.2 设备序列号获取

### 请求数据

Method	URL
POST	http://设备 IP:8090/getDeviceKey

 注意：调用该接口不需要传入参数，不需要传入 pass

### postman 示例



### 返回示例


```
{
  "data": "84E0F4200DF202FA", //设备序列号，设备出厂唯一标识码
  "result": 1, //接口调通
  "success": true //成功获取设备序列号
}
```

## 2.3 设备配置

### 请求数据

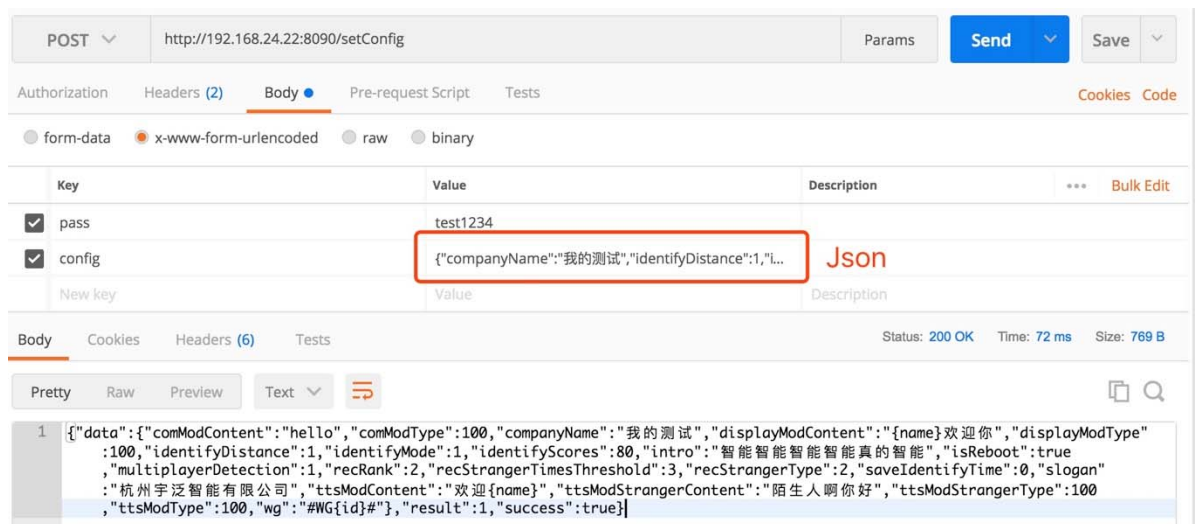
Method	URL			
POST	http://设备 IP:8090/setConfig			
参数名	描述	类型	必传	附加说明
pass	设备密码	String	Y	
config	设备配置集合	Json	Y	<ul style="list-style-type: none"><li>• Json 示例： {"companyName":"我的测试", "identifyDistance":1, "identifyScores":80, "saveIdentifyTime":0, "ttsModType":100, "ttsModContent": "欢迎 {name}", "comModType":100, "comModContent": "hello", "displayModType":100, "displayModContent": "{name}欢迎你", "slogan": "杭州宇泛智能有限公司", "intro": "智能智能智能智能真的智能", "recStrangerTimesThreshold":3, "recStrangerType":2, "ttsModStrangerType":100, "ttsModStrangerContent": "陌生人啊你好", "multiplayerDetection":1, "wg":"#WG{idcardNum}##", "recRank":2, "delayTimeForCloseDoor":500}</li><li>• Uface-M72XX 设备默认输出韦根 26 信号，可修改参数 wg 传入值输出韦根 34 信号</li><li>• slogan 和 intro 中传入空格、回车键均能生效；返回数据中会对应显示空格位和\n</li><li>• slogan 和 intro 仅在大屏模式下展示</li><li>• 若模式参数 ModType 传入 100，则自定义参数 ModContent 必传；其他参数均为非必传，若不传则为默认值（不会保留上次传入的值）</li></ul>

### postman 示例

 注意：a) POST 请求参数放在 body 里

b) ContentType: application/x-www-form-urlencoded

c) 本章所有示例均使用 postman




## 返回示例

```
{
  "data": {
    "comModContent": "hello", //String, 串口输出模式自定义内容
    "comModType": 100, //Int, 串口输出模式
    "companyName": "我的测试", //String, 设备名称, 显示位置参见设备屏幕
    "delayTimeForCloseDoor": 500, //Int, 继电器控制开门到关门的时间间隔, 单位 ms
    "displayModContent": "{name}欢迎你", //String, 屏幕显示模式自定义内容
    "displayModType": 100, //Int, 屏幕显示模式
    "identifyDistance": 1, //Int, 识别距离
    "identifyScores": 80, //Int, 识别分数
    "intro": "智能智能智能智能真的智能", //String, 公司简介 (大屏展示)
    "multiplayerDetection": 1, //String, 多个人脸检测设置
    "recRank": 2, //String, 识别等级
    "recStrangerTimesThreshold": 3, //String, 陌生人判定
    "recStrangerType": 2, //String, 陌生人开关 (是否进行陌生人识别)
    "saveIdentifyTime": 0, //String, 识别记录时间窗
    "slogan": "杭州宇泛智能有限公司", //String, 标语 (大屏展示)
    "ttsModContent": "欢迎{name}", //String, 语音播报模式自定义内容
    "ttsModStrangerContent": "陌生人啊你好", //String, 陌生人语音播报模式自定义内容
    "ttsModStrangerType": 100, //Int, 陌生人语音播报模式
    "ttsModType": 100, //Int, 语音播报模式
    "wg": "#WG{id}#", //String, 韦根类型及输出
  },
  "result": 1, //接口调通
  "success": true //设备配置设置成功
}
```

### 2.3.1 识别成功参数

#### (1) 语音播报模式、语音播报模式自定义内容

设备成功识别人员后，默认 1：不播报语音。可选择 1：不播报语音，2：播报名字，100：自定义。


 注意：a) 播报自定义内容，只允许{name}字段，{name}字段格式固定，其他内容只允许数字、英文和汉字，不允许符号，长度限制 255 个字符。如：{name}欢迎光临。

b) 生僻字、大写汉字、除英文外的其他语言文字无法播报，可播报简单的英文单词。

c) 若人员设置了时间段权限 `passTime` ([参见时间段权限设置](#))，人员在非允许的时间段内识别，设备会播报“姓名权限不足”。

#### (2) 屏幕显示模式、屏幕显示模式自定义内容


设备成功识别人员后，默认 1：显示人员姓名。可选择 1：显示名字，100：自定义。

 注意：a) 显示自定义内容，只允许{name}字段，{name}字段格式固定，其他内容只允许数字、中英文和中英文符号，长度限制 255 个字符。如：{name}，签到成功！

c) 若人员设置了时间段权限 `passTime`，人员在非允许的时间段内识别，设备识别人员后会显示“姓名+权限不足” ([参见时间段权限设置](#))。

#### (3) 串口输出模式、串口输出模式自定义内容

设备成功人员后，串口输出默认 1：开门信号，若设备连接了门禁，人员识别成功后就会触发开门。串口输出可选择 1：开门，2：不输出，3：输出人员 id，4：输出卡号 idcardNum，100 自定义。

 注意：a) 只允许{id}、{idcardNum}字段，字段格式固定，其他内容只允许英文和英文字符，长度限制 255 个字符。

b) 串口支持输出 韦根信号，设备需要外接串口→韦根信号转换小板，小板由本公司定制。串口输出模式选择自定义，写入格式如图所示。{id}、{idcardNum} 字段+数字组合后，韦根 26 范围为 1-65535，有效范围为 5 位；韦根 34 范围为 1-4294967295，有效范围 10 位。若超出范围，则输出的信号会进行转换，输出无效信号。


- 韦根 26 输出，"comModContent": "#WG{id}#"
- 韦根 34 输出，"comModContent": "#34WG{id}#"

#### (4) 韦根类型及输出

设置韦根口的韦根信号输出，此参数只对 Uface-M72XX 设备有效。

韦根 26 传参格式为："wg": "#WG{id}#" 或 "wg": "#WG{idcardNum}#"

韦根 34 传参格式为："wg": "#34WG{id}#" 或 "wg": "#34WG{idcardNum}#"


 注意：a) Uface-M5201 系列设备可以通过串口转换输出韦根信号，[参见识别成功参数 \(3\) 串口输出模式](#)。Uface-M7202 系列、Uface-M7212 设备硬件接口有串口和韦根口，既可通过串口转换输出韦根信号，也可通过韦根口直接输出韦根信号。

b) 韦根 26 范围为 1-65535，有效范围为 5 位；韦根 34 范围为 1-4294967295，有效范围为 10 位。若超出范围，则输出的信号会进行转换，输出无效信号。

### 2.3.2 陌生人参数


#### (1) 陌生人开关

设备默认 1：不识别陌生人，即只识别注册人员，对检测到的陌生人（非注册人员）不会记录识别结果。可选择 2：识别陌生人，识别回调（[参见识别回调](#)）和识别记录都会有陌生人识别结果。

 注意：选择“识别陌生人”选项后，陌生人语音播报模式、陌生人判定配置项才会生效，参见（2）。

### （2）陌生人语音播报模式、陌生人语音播报自定义内容


设备识别到陌生人后，设备不会弹窗文字提示，默认 1：不播报语音。可选择 1：不播报语音，2：语音播报 “陌生人警报”，100：自定义。

 注意：a) 播报自定义内容，只允许数字、英文和汉字，不允许符号，长度限制 255 个字符。如：注意陌生人。

b) 生僻字、大写汉字、除英文外的其他语言文字无法播报，可播报简单的英文单词。

### （3）陌生人判定

设备判定人脸为陌生人所需时间等级，默认 3；传入值请选择 1-20 之间的整数，1 表示快速判定但精确率最低，随着数值增加，判定时间增长，精确度提高。


 注意：设备检测到人脸并进行识别，若多次识别均为非注册人员，则判定为陌生人。此处“时间等级”可理解为对同一人脸的识别次数。

## 2.3.3 识别参数

### （1）识别距离

- 设备对识别距离范围内的人脸进行检测识别，超出识别距离的人脸不会进行检测。默认 0，无距离限制，只要设备检测到人脸（即出现人脸框）都会进行检测。
- 【Uface-M5201 系列设备】默认 0：无限制，1：0.5 米以内，2：1 米以内，3：1.5 米以内，4：2 米以内，5：3 米以内，6：4 米以内。
- 【Uface-M72XX 系列设备】若识别等级设置为 1 或 2，则识别距离可设置 0-6 档，默认 0：无限制，1：0.5 米以内，2：1 米以内，3：1.5 米以内，4：2 米以内，5：3 米以内，6：4 米以内。


若识别等级设置为 3，则识别距离仅 1-3 档实际生效，若设置为 0 或 4、5、6，默认等于 1.5 米以内，即 0：1.5 米以内，1：0.5 米以内，默认 2：1 米以内，3：1.5 米以内，4：1.5 米以内，5：1.5 米以内，6：1.5 米以内。参见（4）识别等级。

 a) 注意：识别距离不是通过距离感应，而是检测到的人脸框的大小通过函数计算得来的，因此识别距离不是精确的。

b) 识别距离 0：无限制。这里无限制表示只要人脸大小达到检测要求，就对人脸进行识别。识别等级 1、等级 2 最远识别距离可达 4-5 米，识别等级 3 最远识别距离为 1.5 米。

## （2）识别（分数）阈值


设备识别人脸结果的过程，实际上是抓拍到的人脸与库内人员的注册照片进行比对，比对分数达到分数阈值，则判定人脸身份。识别分数阈值默认 80，要求传入值为 50-100 的整数，分数越高，识别准确率越高，但识别速度会变慢。

 注意：a) 设备对同一人脸进行多次比对，若前几次达不到分数阈值，则设备不会给出识别结果，因此会感觉识别时间较长、设备反应慢。

b) 若设置分数阈值达到 85 分以上，抓拍人脸与注册照比对有很大概率达不到分数阈值，设备无法给出识别结果，即“不识别”。

## （3）识别记录时间窗


时间窗内（N 秒），同一人脸多次识别，只保留第一条识别记录。若设置了识别回调（[参见识别回调](#)），只回调第一条记录。默认值为 0，即回调无时间间隔限制，每次识别都会进行回调。

 注意：a) 时间窗对陌生人无效。

b) 设备对人脸的每次识别结果，在网络通畅的情况下，都会实时推送到回调地址，时间差为毫秒级。

## （4）识别等级

是否需要活体识别，此参数只对配备活体检测功能的 Uface-72XX 设备有效。等级 1：快速识别，不能拒绝照片；等级 2：能拒绝部分照片；等级 3：能拒绝照片和视频。


 注意：Uface-M72XX 设备，默认等级 3，可更改识别等级。使用等级 3 时，识别距离最大建议 1.5m。参见（1）识别距离。

### （5）多个人脸检测设置

设备默认 1：检测多个人脸并进行识别，即只要设备检测到人脸都会进行识别，每个人脸都会有识别结果（成功或失败）。可选择 2：只检测多个人脸中最大的人脸并进行识别，即多个人脸只有最大人脸会有一个识别结果（成功或失败），适用于闸机等一次一人的场景。

### （6）继电器控制时间

识别成功后，继电器输出开关量信号的持续的时间，默认 500ms。连接门禁时表现为：识别成功后，开门到关门的时间间隔。传入值要求为 100-25500，单位为 ms。

 注意：a) 根据使用的场景，选择开门到关门之间的时间间隔。

b) 输入数值小于 100 会报错；数值向下取整百，如：输入 101-199 之间的数值，实际生效时间均为 100ms。

c) Uface-M5201 设备只有串口，因此外接信号转换后只能韦根、开关量（继电器）二选一。同理，Uface-M5201-GH 内置三合一小板，接线有串口、韦根、继电器，但原始信号（串口）只有一个，只能选择一种信号输出

## 2.3.4 其他参数

### （1）标语

设备通过 HDMI 接口外接大屏时显示的文字内容，图见（2）所示。

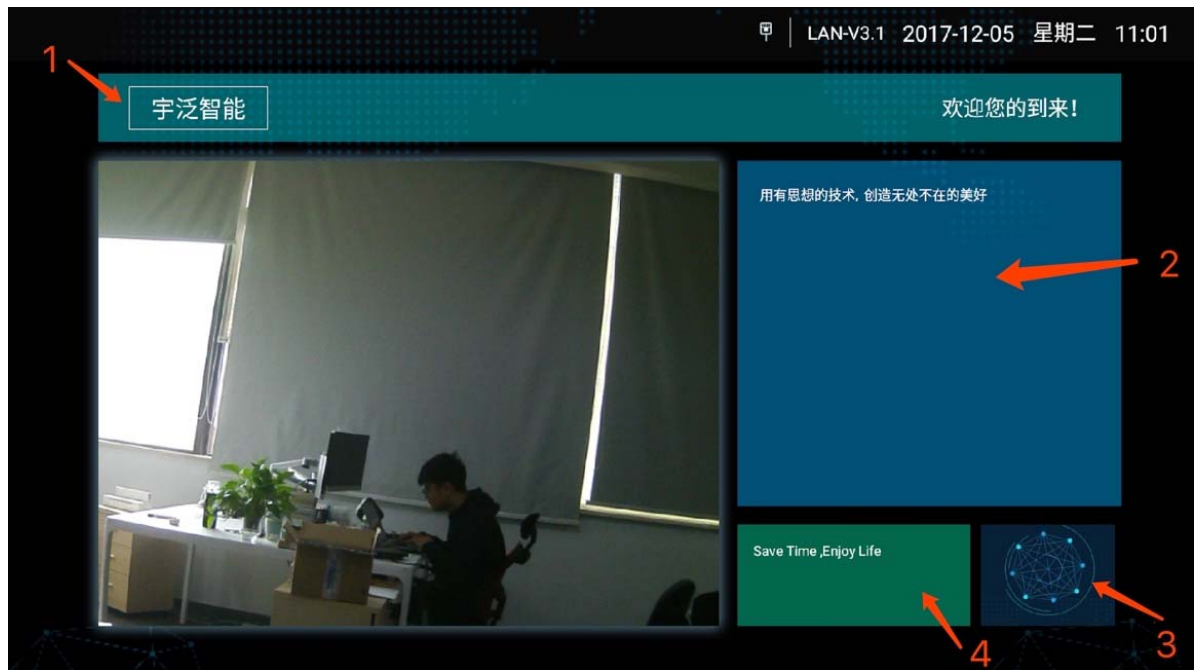
### （2）简介



设备通过 HDMI 接口外接大屏时显示的文字内容，如图所示。

**i** 注意：a) 设备后端有 HDMI 输出接口，输出分辨率为 1920×1080，HDMI 接口不支持热插拔，连接 HDMI 请先将设备关机，插入 HDMI 后再接入电源。

b) 目前 Uface-M72XX 设备不支持大屏。



**1.设备名称：**可自行设备配置接口对设备名称(companyName)进行自定义

**2.公司简介：**可自行通过设备配置接口对公司简介(intro)进行编辑

**3. Logo 展示区：**默认 Logo 为动画，可通过 Logo 修改接口对该区域展示内容进行修改，目前仅支持静态自定义图片

**4.slogan（标语）：**可自行通过设备配置接口对标语(slogan)进行编辑

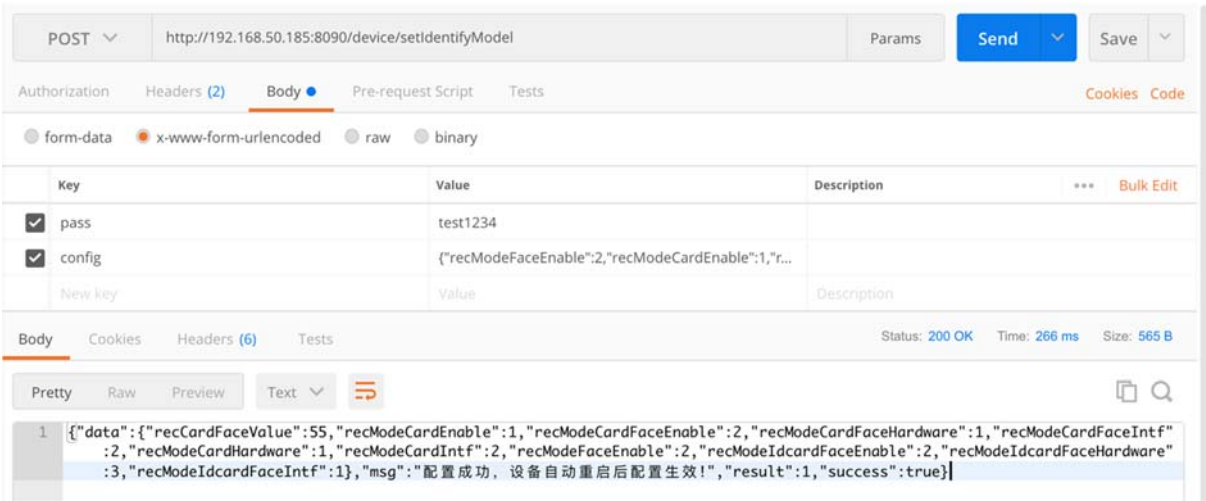
**5.视频预览区：**显示摄像头实时画面

## 2.4 识别模式配置





### 请求数据

Method	URL			
POST	http://设备 IP:8090/device/setIdentifyModel			
参数名	描述	类型	必传	附加说明
pass	设备密码	String	Y	
config	识别模式配置集合	Json	Y	<ul style="list-style-type: none"><li>• Json 示例: { "recModeFaceEnable":2,"recModeCardEnable":1,"recModeCardIntf":2,"recModeCardHardware":1,"recModeCardFaceEnable":2,"recModeCardFaceIntf":2,"recModeCardFaceHardware":1,"recModeIdcardFaceEnable":2,"recModeIdcardFaceIntf":1,"recModeIdcardFaceHardware":3,"recCardFaceValue":55}</li><li>• 所有参数均为必传</li><li>• 刷卡、卡&amp;脸双重认证模式不能同时使用</li><li>• 识别模式接口调通后，设备会自动重启，重启后识别模式配置才会生效</li></ul>

### postman 示例



界面说明

卡&人脸双重认证	
	
人证比对	
	

返回示例

```
{
  "data": {
    "recCardFaceValue": 60,
    "recModeCardEnable": 1,//Int，刷卡模式开关
    "recModeCardFaceEnable": 2,// Int，卡&人脸双重模式开关
    "recModeCardFaceHardware": 1,// Int，卡&人脸双重认证模式外接硬件类型
    "recModeCardFaceIntf": 2,// Int，卡&人脸双重认证模式卡号传输接口
    "recModeCardHardware": 1,// Int，刷卡模式外接硬件类型
    "recModeCardIntf": 2,// Int，刷卡模式卡号传输接口
    "recModeFaceEnable": 2,// Int，刷脸模式开关
    "recModeldcardFaceEnable": 2,// Int，人证比对模式开关
    "recModeldcardFaceHardware": 3,// Int，人证比对模式外接硬件类型
    "recModeldcardFaceIntf": 1// Int，人证比对模式卡号传输接口
  },
  "msg": "配置成功，设备自动重启后配置生效！"，
  "result": 1,//接口调通
  "success": true//识别模式配置成功
}
```

参数说明

字段名称	含义及取值
刷脸识别参数	
recModeFaceEnable	刷脸模式开关 * 1.关闭 * 2.打开（默认）
刷卡识别参数	
recModeCardEnable	* 1.关闭 * 2.打开（默认）
recModeCardIntf	卡号传输接口 * 1.TTL 串口（Uface-M72XX 默认） * 2.232 串口（Uface-M5201 系列默认） • Uface-M72XX 设备自带的刷卡模块使用的是 TTL 串口 • TTL 或 232 串口接口只能被一种识别模式使用，若>1 种模式使用 TTL 或 232 串口则报错
recModeCardHardware	外接硬件类型 * 1.IC 读卡器（默认）

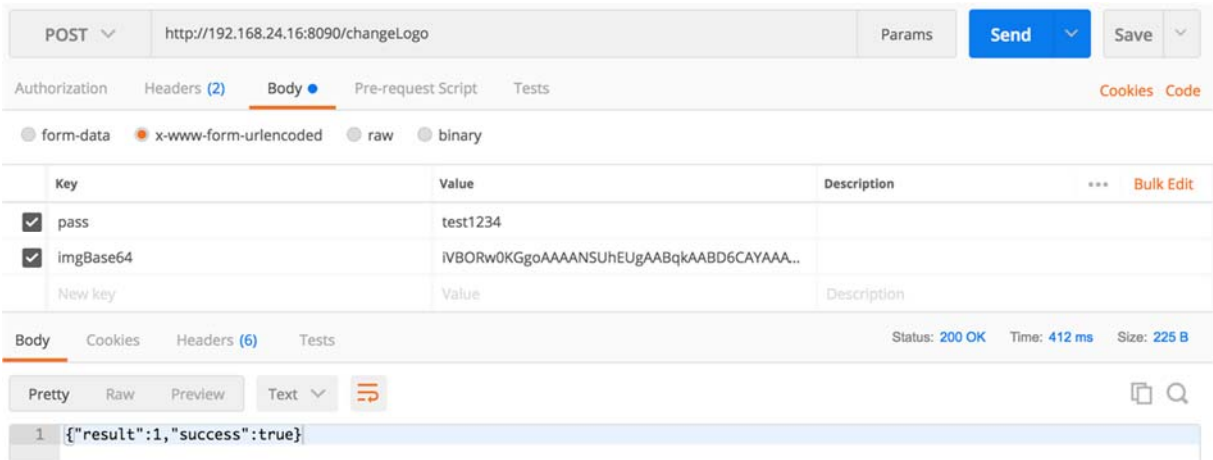
	<ul style="list-style-type: none"> <li>Uface-M72XX 设备自带的刷卡模块属于 IC 读卡器</li> </ul>
卡&人脸双重认证	
recModeCardFaceEnable	卡&人脸双重认证开关 * 1.关闭（默认） * 2.打开
recModeCardFaceIntf	卡号传输接口 * 1.TTL 串口（Uface-M72XX 默认） * 2.232 串口（Uface-M5201 系列默认） <ul style="list-style-type: none"> <li>Uface-M72XX 设备自带的刷卡模块使用的是 TTL 串口</li> <li>TTL 或 232 串口接口只能被一种识别模式使用，若&gt;1 种模式使用 TTL 或 232 串口则报错</li> </ul>
recModeCardFaceHardware	外接硬件类型 * 1.IC 读卡器（默认） <ul style="list-style-type: none"> <li>Uface-M72XX 设备自带的刷卡模块属于 IC 读卡器</li> </ul>
人证对比参数	
recModeldCardFaceEnable	人证比对开关 * 1.关闭（默认） * 2.打开
recModeldCardFaceIntf	卡号传输接口 * 1.TTL 串口 * 2.232 串口（默认） * 3.USB <ul style="list-style-type: none"> <li>默认使用 232 串口外接新中新身份证阅读器</li> <li>USB（或 TTL 或 232）接口只能被一种识别模式使用，若&gt;1 种模式使用 USB（或 TTL 或 232）则报错</li> </ul>
recModeldCardFaceHardware	外接硬件类型 * 1.新中新身份证阅读器(默认) * 2.ZTK 中控身份证阅读器 * 3.精伦身份证阅读器 * 4.华视身份证阅读器 <ul style="list-style-type: none"> <li>默认使用 232 串口外接新中新身份证阅读器</li> </ul>
卡&人脸双重认证、人证比对共用参数	
recFaceCardValue	人脸识别阈值 * 55（默认） <ul style="list-style-type: none"> <li>请输入 50-100 之间的整数。分数越高，识别准确率越高，但识别速度会变慢。</li> </ul>

## 2.5 修改 Logo

### 请求数据

Method	URL			
POST	http://设备 IP:8090/changeLogo			
参数名	描述	类型	必传	附加说明
pass	设备密码	String	Y	
imgBase64	Logo 图片的 Base64 编码字符串	String	Y	<ul style="list-style-type: none"><li>不加头部，如：data:image/jpg;base64,</li><li>若要恢复设备默认 logo，请传入 -1</li><li>图片尺寸暂无强行限定条件，建议按照 Logo 展示区尺寸等比例适当缩放。尺寸建议不要过大。Logo 尺寸<a href="#">参见设备界面</a></li><li>Logo 图片格式仅支持 JPG 和 PNG</li></ul>

### postman 示例



### 返回示例

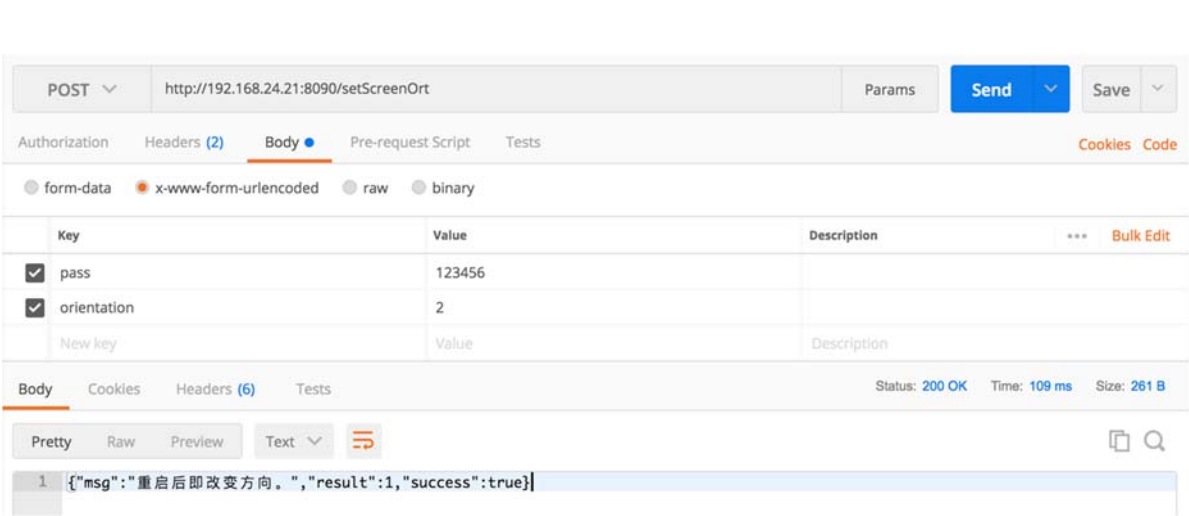
```
{
  "result": 1, //接口调通
  "success": true //Logo 修改成功
}
```

## 2.6 修改屏幕方向

### 请求数据

Method	URL			
POST	http://设备 IP:8090/setScreenOrt			
参数名	描述	类型	必传	附加说明
pass	设备密码	String	Y	
orientation	屏幕方向	String	Y	<ul style="list-style-type: none"><li>默认横屏，1：横屏，2：竖屏</li><li>该接口成功调用后，约 3 秒钟之后手动断电重启设备或调用设备重启接口，屏幕方向设置在设备重启后生效</li><li>Uface-M72XX 设备目前只支持竖屏，修改屏幕方向对 Uface-M72XX 设备无效</li></ul>

### postman 示例




### 返回示例

```
{
  "msg": "重启后即改变方向。",
  "result": 1,//接口调通
  "success": true//设备屏幕方向设置成功
}
```

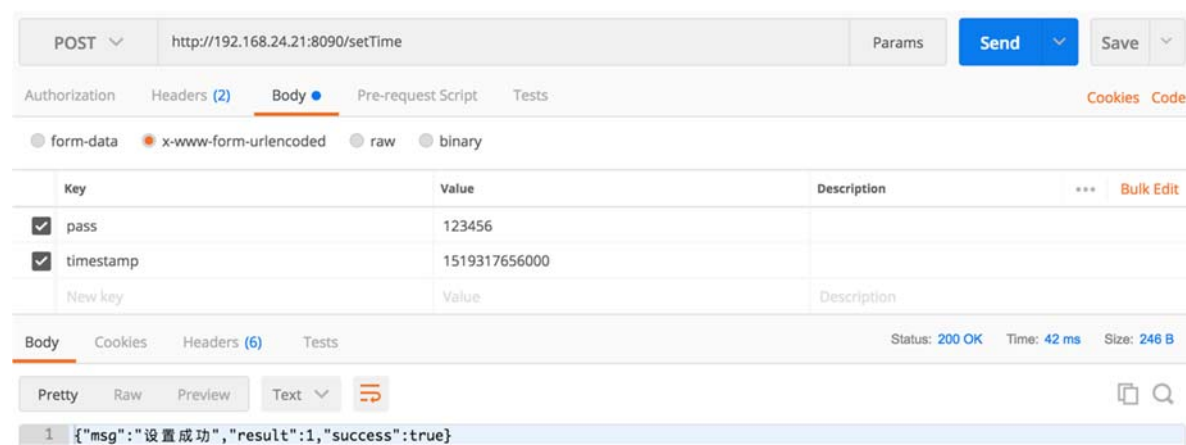
## 2.7 设置设备时间

### 请求数据

Method	URL			
POST	http://设备 IP:8090/setTime			
参数名	描述	类型	必传	附加说明
pass	设备密码	String	Y	
timestamp	Unix 毫秒级时间戳	String	Y	<ul style="list-style-type: none"><li>配置成功后，设备时间即被改为当前所设置的时间</li><li>若设备未连入公网，时间设置成功后，会按照设置的时间增长；设备每隔 5 分钟刷新时间</li><li>若设备连入公网，设备本身有网络时间校对机制，每隔 5 分钟会联网校对一次时间，将设备时间调整与公网时间一致</li></ul>

 注意：若要设备显示手动设置的时间，设备必须处于局域网内。若连上公网，设备刷新自身时间时默认使用公网时间。

### postman 示例






## 返回示例

```
{  
  "msg": "设置成功",  
  "result": 1,//接口调通  
  "success": true//设备时间设置成功  
}
```

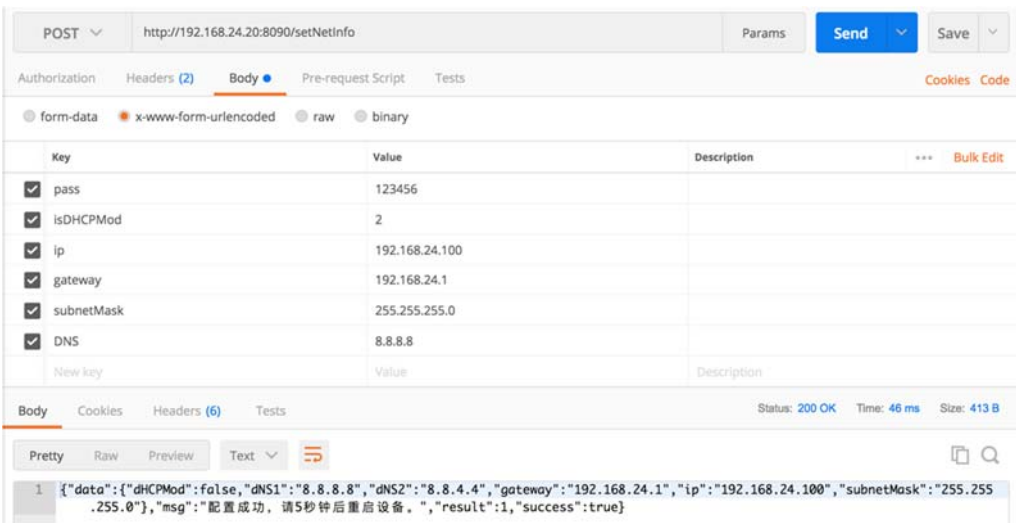
## 2.8 有线网络配置

### 请求数据

Method	URL			
POST	http://设备 IP:8090/setNetInfo			
参数名	描述	类型	必传	附加说明
pass	设备密码	String	Y	
isDHCPMod	DHCP 模式选择	Int	Y	设备默认为 DHCP 模式，即自动获取 IP 地址 • 传入 1：设备配置为 DHCP 模式，自动获取动态 IP，无须传入其余参数 • 传入 2：自定义网络配置，即手动设置静态 IP，其余参数必须传入且不可为空 设备网络通畅，则每分钟都会自动刷新 IP 地址
ip	ip 地址	String		此处 ip 字段名必须传入小写，ip 不可大于 255
gateway	网关	String		
subnetMask	子网掩码	String		
DNS	DNS 服务器	String		

 注意：此接口设置的固定 IP 适用于网线连接，Wi-Fi 连接设置固定 IP 请[参见 Wi-Fi 配置接口](#)。

### postman 示例



## 返回示例

- 配置为 DHCP 模式（自动获取 IP）

```
{
  "msg": "已设置为 DHCP 模式",
  "result": 1,//接口调通
  "success": true//DHCP 模式设置成功
}
```


- 配置为自定义模式（手动配置静态 IP）

```
{
  "data": {
    "dHCPMod": false,
    "dNS1": "8.8.8.8",
    "dNS2": "8.8.4.4",
    "gateway": "192.168.24.1",
    "ip": "192.168.24.100",
    "subnetMask": "255.255.255.0"
  },
  "msg": "配置成功，请 5 秒钟后重启设备。",
  "result": 1,//接口调通
  "success": true//固定 IP 设置成功
}
```

## 2.9 Wi-Fi 配置

### 请求数据

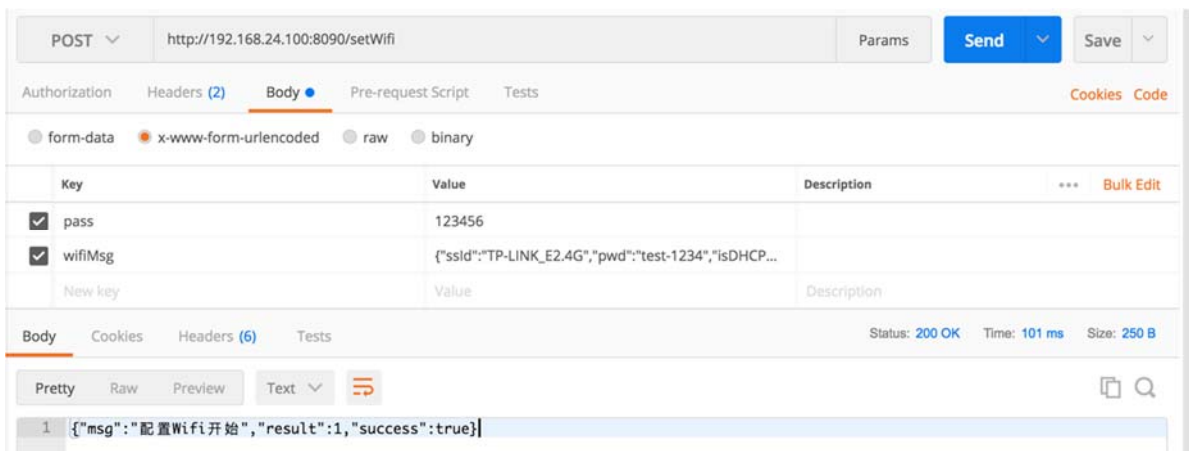
Method	URL			
POST	http://设备 IP:8090/setWifi			
参数名	描述	类型	必传	附加说明
pass	设备密码	String	Y	
wifiMsg	无线配置信息集合	Json	Y	Json 示例: <ul style="list-style-type: none"><li>• 自动获取 IP { "ssid": "TP-LINK_E2.4G", "pwd": "test-1234", "isDHCPMod": true }</li><li>• 固定 IP { "ssid": "TP-LINK_E2.4G", "pwd": "test-1234", "isDHCPMod": false, "ip": "192.168.20.66", "gateway": "192.168.20.1", "dns": "8.8.8.8" }</li></ul>

 注意：设备仅支持连接 2.4G 频段的 Wi-Fi，不支持 5G 频段的 Wi-Fi，不支持需要 Web 页面认证 Wi-Fi，不支持需要证书认证的 Wi-Fi。

### wifiMsg 类集合说明

```
public class wifiMsg {  
    private String ssid;//Wi-Fi 名称，必传  
    private String pwd;//Wi-Fi 密码，只允许数字、英文和英文字符；若 Wi-Fi 无密码，则传入空或者任意字符都可  
    private Boolean isDHCPMod;//是否设置为动态 IP，必传。若传入 false，则 ip，gateway，dns 必传且不可为空；若传入 true，则 ip，gateway，dns 无需传入，传入也不会生效  
    private String ip;//IP 地址  
    private String gateway;//网关  
    private String dns;//DNS 服务器  
}
```

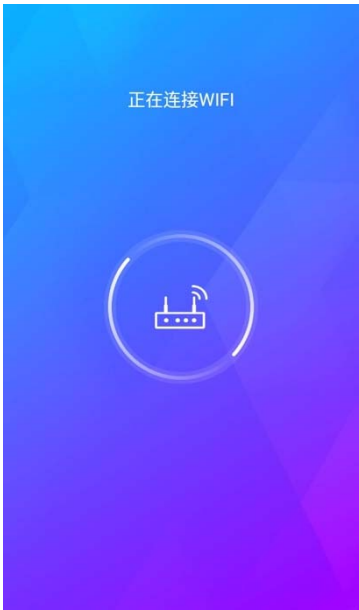
## postman 示例



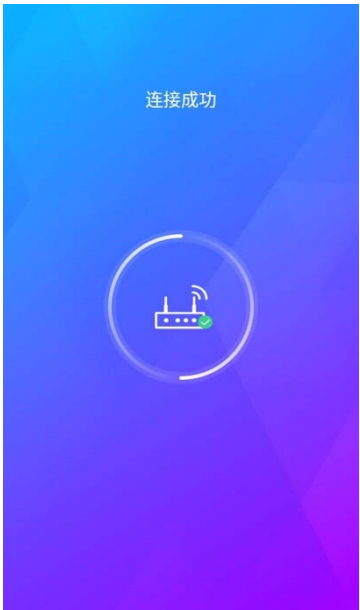
## 返回示例

```
{
  "msg": "配置 Wifi 开始",
  "result": 1,//接口调通
  "success": true//Wi-Fi 账号密码已发送到设备
}
```

## 设备界面说明



Wi-Fi 连接中



Wi-Fi 连接成功

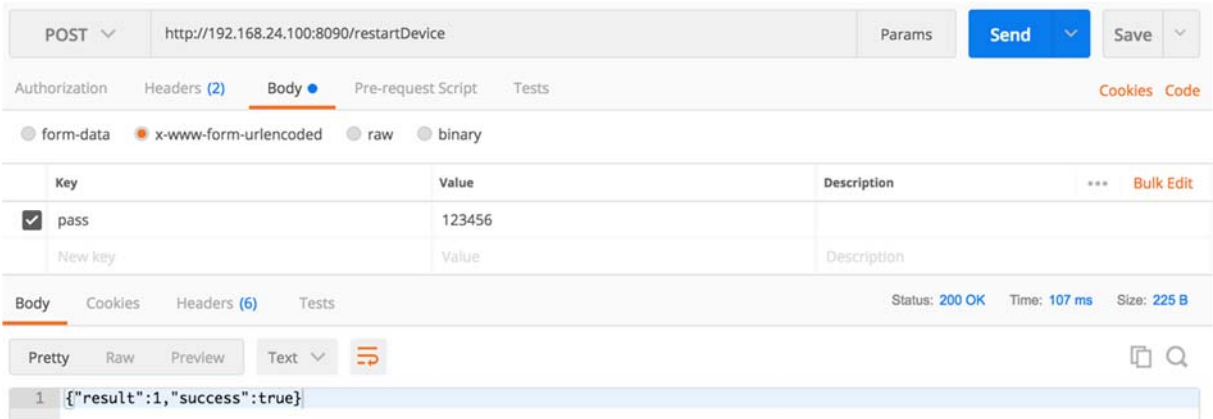


## 2.10 设备重启

### 请求数据

Method	URL			
POST	http://设备 IP:8090/restartDevice			
参数名	描述	类型	必传	附加说明
pass	设备密码	String	Y	

### postman 示例



### 返回示例

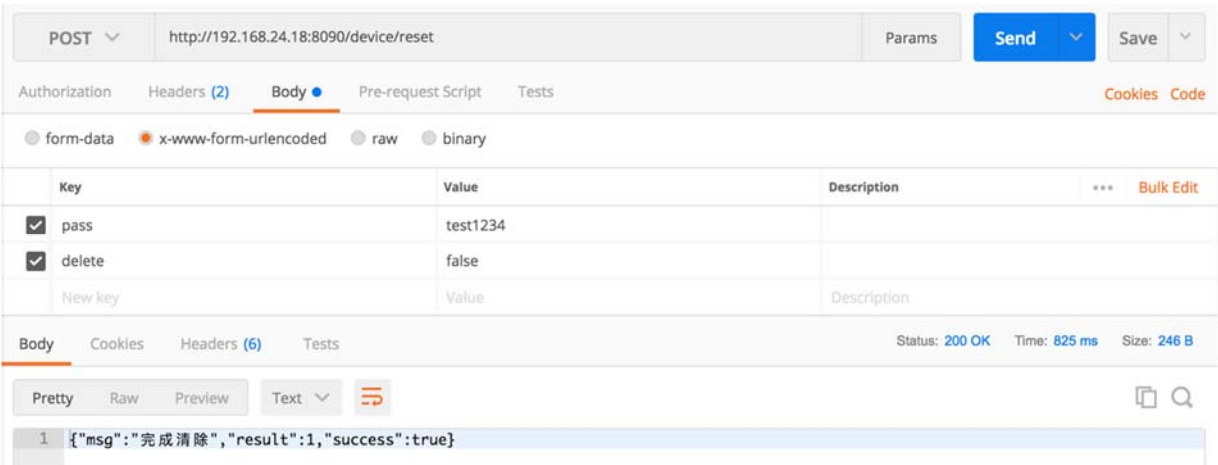
```
{
  "result":1,//接口调通
  "success":true//设备开始执行重启指令
}
```

## 2.11 设备重置

### 请求数据

Method	URL			
POST	http://设备 IP:8090/device/reset			
参数名	描述	类型	必传	附加说明
pass	设备密码	String	Y	
delete	选择性删除	Boolean	Y	<ul style="list-style-type: none"><li>删除设备上所有的识别记录、注册照、现场照，人员、特征等所有的数据，清空所有的数据库</li><li>传入 true，删除以上所有信息，并删除通过设备配置接口设置的属性</li><li>传入 false，删除以上所有信息，但不删除通过设备配置接口已经设置的属性</li></ul>

### postman 示例



**注意：**a) delete 字段传入 true，设备清空自身数据库，网络配置恢复 DHCP 自动获取 IP 模式，设备配置恢复出厂设置但设备屏幕方向保持不变；设备自动重启。

b) delete 字段传入 false，设备配置、Logo、屏幕方向保持不变；设备清空存储内容（包括现场抓拍照，注册照，人员数据库，有线/无线配置，大屏背景图片，设备日志）。



c)物理重置:设备背面的 RESET 小孔长按 8-12 秒;功能同 delete 字段传入 true。

#### 返回示例

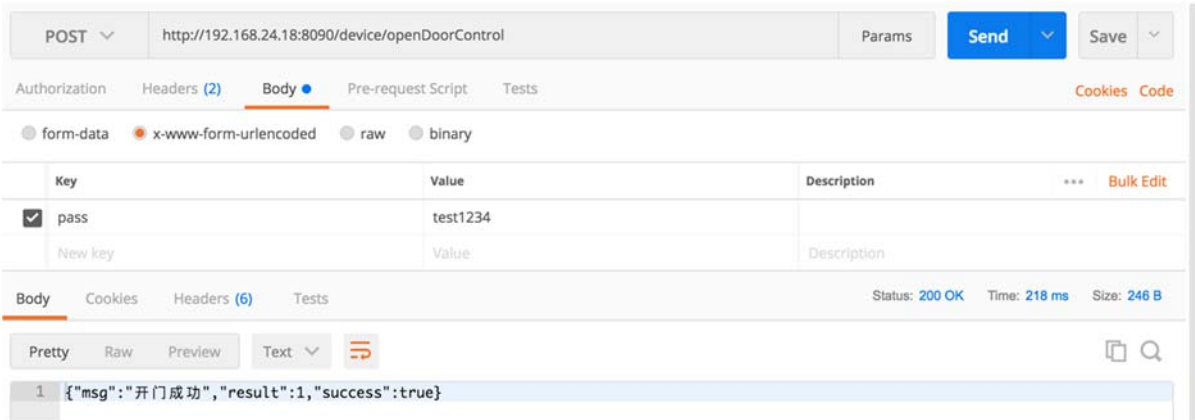
```
{  
  "msg": "完成清除",  
  "result":1,//接口调通  
  "success":true//设备开始执行重置指令  
}
```

## 2.12 设备开门

### 请求数据

Method	URL			
POST	http://设备 IP:8090/device/openDoorControl			
参数名	描述	类型	必传	附加说明
pass	设备密码	String	Y	

### postman 示例



**i** 注意：a) 密码验证正确，设备就会控制开启门禁

b)返回 result=1,表示设备开门接口调用成功;设备硬件上需接有门禁控制小板，才会控制开启门禁

### 返回示例

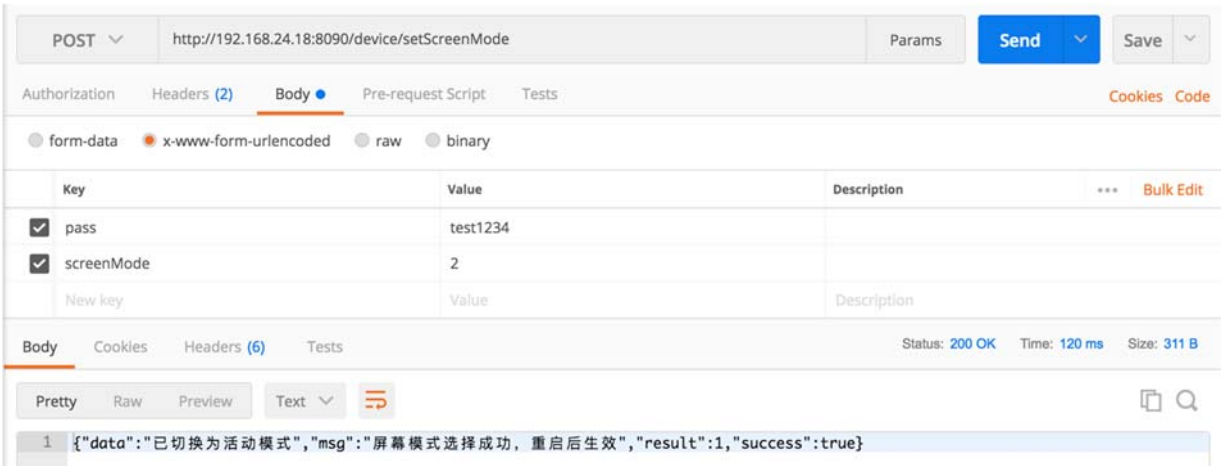
```
{
  "msg": "开门成功",
  "result": 1, //接口调通
  "success": true //设备开始执行开门指令
}
```

## 2.13 大屏模式接口

### 请求数据

Method	URL			
POST	http://设备 IP:8090/device/setScreenMode			
参数名	描述	类型	必传	附加说明
pass	设备密码	String	Y	
screenMode	选择大屏显示模式	Int	Y	1: 传统模式, 文字内容编辑 <a href="#">参见设备配置</a> , Logo 编辑 <a href="#">参见修改 Logo</a> 2: 自定义大屏背景模式, 更换背景图片 <a href="#">参见大屏背景图片配置</a> • 大屏界面 <a href="#">参见设备配置</a>

### postman 示例



**i** 注意: a) 设备后端有 HDMI 输出接口, 输出分辨率为 1920×1080, HDMI 接口不支持热插拔, 连接 HDMI 请先将设备关机, 插入 HDMI 后再接入电源。

b) 目前 Uface-M7202 系列、Uface-M7212 设备不支持大屏。

## 返回示例

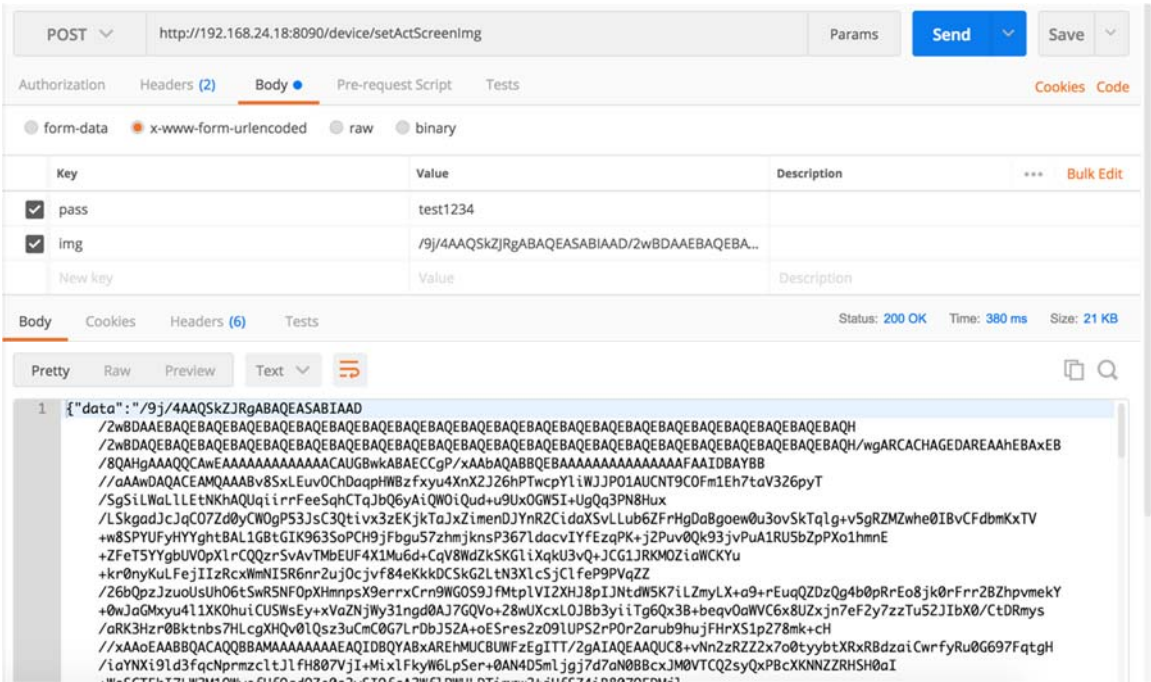
```
{  
  "data": "已切换为活动模式",  
  "msg": "屏幕模式选择成功，重启后生效",  
  "result": 1,//接口调通  
  "success": true//大屏模式设置成功  
}
```

2.14 大屏背景图片配置

请求数据

Method	URL			
POST	http://设备 IP:8090/device/setActScreenImg			
参数名	描述	类型	必传	附加说明
pass	设备密码	String	Y	
img	图片的 base64 编码字符串	String	Y	图片分辨率必须小于 1920×1080

postman 示例



**i** 注意：a) 需先将大屏模式接口设置成 2：自定义大屏背景模式，设置成功的图片才会生效。

b) 若使用本接口配置图片成功，则设备配置中的参数 slogan（标语）、intro（简介）、companyName（设备名称）以及 Logo 不会在大屏上显示。

## 返回示例

```
{  
  "data": "base64 编码字符串",  
  "msg": "修改背景图片成功",  
  "result": 1,//接口调通  
  "success": true//大屏背景图片设置成功  
}
```

## 2.15 识别回调

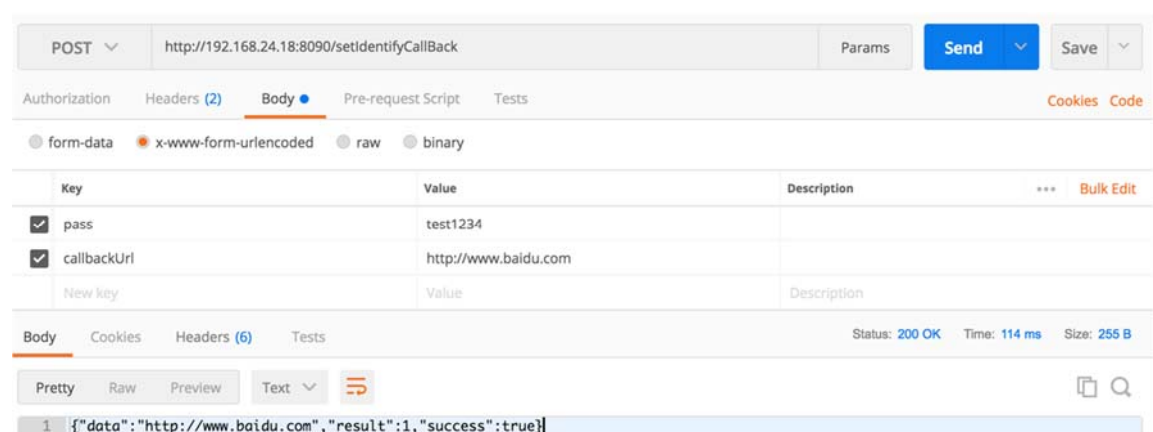
### 请求数据

Method	URL			
POST	http://设备 IP:8090/setIdentifyCallBack			
参数名	描述	类型	必传	附加说明
pass	设备密码	String	Y	
callbackUrl	回调地址	String	Y	<ul style="list-style-type: none"><li>设备成功识别人员后，会向该地址 POST 字段 ip、personId、time（识别时间）、deviceKey、type: face/card_0/1/2 和 path, <a href="#">参见回调字段说明</a></li><li>传入内容为空可以清空回调地址，清空后识别成功将不再进行回调</li><li>回调 URL 需要符合正则表达式: String check = "((http ftp https):/)(([a-zA-Z0-9\\_\\-]+\\.[a-zA-Z]{2,6}) ([0-9]{1,3}\\.[0-9]{1,3}\\.[0-9]{1,3}\\.[0-9]{1,3}))(:[0-9]{1,4})*/([a-zA-Z0-9\\_\\-\\/\\.~]*)*?"</li></ul>

**i** 注意：a) 设备配置接口设置 saveIdentifyTime（识别记录时间窗）后，在该时间间隔内，同一个人多次识别只会向回调地址 POST 一条识别记录

b) URL 端口说明：若使用页面作为回调地址，端口号须为 8090；若使用 API 作为回调地址，端口号最大为 9999，且路由器对该端口号开放。

### postman 示例




## 返回示例

```
{
  "data": "http://www.baidu.com,
  "result": 1,//接口调通
  "success": true//回调地址设置成功
}
```

## 回调示例

- ContentResult 里封装了 result 和 success 字段，回调成功则返回 {"result":1, "success":true}，完整的回调示例[参见附录 8 回调示例详情](#)。

```
//setIdentifyCallBack
@RequestMapping(value = "/xcy/test")
public ContentResult<Map<String, Object>> test(String personId, String deviceKey, String type,
String ip, long time,String path) {
    Map<String, Object> map = new HashMap<>();
    map.put("personId", personId);
    map.put("deviceKey", deviceKey);
    map.put("type", type);
    map.put("ip", ip);
    map.put("time", time);
    map.put("path", path);
    SimpleDateFormat sdf = new
SimpleDateFormat(ModelConstant.DATE_FORMAT_COMPLICATED);
    System.out.println(sdf.format(new Date()));
    return new ContentResult<Map<String,Object>>("SUS_001", map);
}
```

 注意：a) 回调地址接收到识别数据后，需返回 {"result":1,"success":true} 给设备，设备解析到 {"result":1,"success":true}，会标记回调成功，标记识别记录中的字段 state:1；若设备对回调返回的结果解析失败，则将该次识别记录存到本地数据库，标记识别记录中的字段 state:0。

b) 设备每间隔 10 分钟，从数据库中读取回调失败（state:0）的识别记录，再次向回调地址发送请求，请求成功则更新数据库中对应的回调状态（state:1）。

c) 若设备未设置回调，则默认标记识别记录中的字段 state:1。



## 回调数据示例

### • 注册人员刷脸

```
2018-02-28 11:00:22 {"path":"ftp://192.168.5.101:8010/recordsImg/2018-02-28/00100_1519786791566.jpg","ip":"192.168.5.101","deviceKey":"84E0F4200E5102FA","personId":"00100","time":1519786791566,"type":"face_1"}
```

### • 陌生人刷脸

```
2018-02-28 11:00:29 {"path":"ftp://192.168.5.101:8010/recordsImg/2018-02-28/STRANGERBABY_1519786799282.jpg","ip":"192.168.5.101","deviceKey":"84E0F4200E5102FA","personId":"STRANGERBABY","time":1519786799282,"type":"face_2"}
```

### • 刷卡

```
{"ip":"192.168.11.128","deviceKey":"84E0F42050D902F8","personId":"6689","time":1542175567398,"type":"card_0"}
```


### • 人卡合一

```
{"path":"ftp://192.168.11.128:8010/recordsImg/2018-11-14/6689_1542175790575.jpg","ip":"192.168.11.128","deviceKey":"84E0F42050D902F8","personId":"6689","time":1542175790575,"type":"faceAndcard_0"}
```

### • 人证比对

```
{"path":"ftp://192.168.11.128:8010/recordsImg/2018-11-14/IDCARD_1542175785475.jpg","ip":"192.168.11.128","deviceKey":"84E0F42050D902F8","personId":"IDCARD","time":1542175785475,"type":"idcard_0","data":"身份证 json 字符串"}
```

## 识别回调参数字段说明

 注意：回调数据在 url 和 body 中各有一份，url 中数据是 Json 格式，body 中数据也是 Json 格式。


参数名	类型	描述
ip	String	设备 IP 地址
deviceKey	String	设备出厂唯一标识码
personId	String	人员 id，陌生人为 STRANGERBABY
time	String	识别记录时间戳（以设备时间为准）
type	String	<ul style="list-style-type: none"><li>• 识别方式（face/faceAndcard/idcard/card）、识别出的人员类</li><li>• face_0（刷脸识别，且该人员在 passtime 权限时间内）</li><li>• face_1（刷脸识别，且该人员在 passtime 权限时间外）</li><li>• face_2（刷脸识别，且识别失败或识别到的是陌生人）</li><li>• card_0（刷卡识别，且该人员在 passtime 权限时间内）</li><li>• card_1（刷卡识别，且该人员在 passtime 权限时间外）</li><li>• faceAndcard_0（双重认证，且该人员在 passtime 权限时间内）</li><li>• faceAndcard_1（双重认证，且该人员在 passtime 权限时间外）</li><li>• faceAndcard_2（双重认证，且识别失败或识别到的是陌生人）</li><li>• idcard_0（人证比对，且该人员在 passtime 权限时间内）</li><li>• idcard_2（人证比对，且识别失败或识别到的是陌生人）</li></ul>

path	String	现场照在设备内的保存路径，访问此 url 需设备局域网在线，且发送请求的客户端与设备处于局域网同一网段
data	String	人证比对时回调才有该字段，身份证相关信息

## 2.16 注册照片回调

### 请求数据

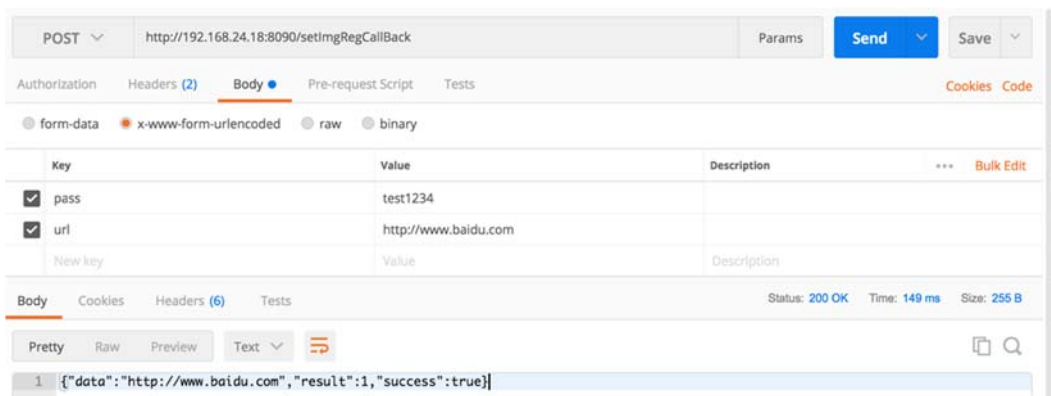
Method	URL			
POST	http://设备 IP:8090/setImgRegCallBack			
参数名	描述	类型	必传	附加说明
pass	设备密码	String	Y	
url	回调地址	String	Y	<ul style="list-style-type: none"><li>照片注册成功（包括设备拍照注册）后，设备会向预先设置的回调接口 POST 字段 deviceKey、personId、time、imgPath、faceId、ip、feature 和 featureKey，<a href="#">参见回调字段说明</a></li><li>传入内容为空可以清空回调地址，清空后识别成功将不再进行回调</li><li>回调 URL 需要符合正则表达式：String check = "((http ftp https):/)(([a-zA-Z0-9\\._-]+\\.[a-zA-Z]{2,6}) ([0-9]{1,3}\\.[0-9]{1,3}\\.[0-9]{1,3}\\.[0-9]{1,3}))(:[0-9]{1,4})*/([a-zA-Z0-9\\&amp;%_\\./-~*])?"</li></ul>

 注意：a) 设备拍照注册实际上是设备先给人员拍摄照片，再调用照片注册接口，将拍摄的照片注册为人员的注册照。因此拍照注册完成后，也会将拍照所得的照片进行注册照片回调。[参见拍照注册](#)。

b) 照片更新，实际上是先删除原照片，再调用照片注册接口注册照片。因此照片更新成功后，会进行注册照片回调。[参见照片更新](#)。

c) URL 端口说明：若使用页面作为回调地址，端口号须为 8090；若使用 API 作为回调地址，端口号最大为 9999，且路由器对该端口号开放。

## postman 示例



## 返回示例

```
{
  "data": "http://www.baidu.com",
  "result": 1, //接口调通
  "success": true //回调地址设置成功
}
```

## 注册照片回调参数字段说明

**i** 注意：a) 回调数据在 url 和 body 中各有一份，url 中数据是 Json 格式，body 中数据也是 Json 格式。

b) 照片注册、照片更新、特征注册都会触发注册照片回调。其中特征注册不是原图注册，因此无法查看图片，imgPath、newImgPath 参数内容为空。


参数名	类型	描述
deviceKey	String	设备唯一标识码
personId	String	人员 id
time	String	时间戳
imgPath	String	照片路径（ftp 路径不完整，旧版参数），如： "imgPath": "/faceRegister/002_ad8c1119bd3c473d838de5d80ce389c9.jpg"
newImgPath	String	照片路径（ftp 路径完整，新版参数），如： "newImgPath": "ftp://192.168.1.224:8010/faceRegister/002_ad8c1119bd3c473d838de5d80ce389c9.jpg"
faceId	String	照片 id
ip	String	设备 IP 地址
feature	String	特征码

featureKey	String	特征秘钥，通过特征码注册时需要该字段进行特征有效性校验
------------	--------	-----------------------------

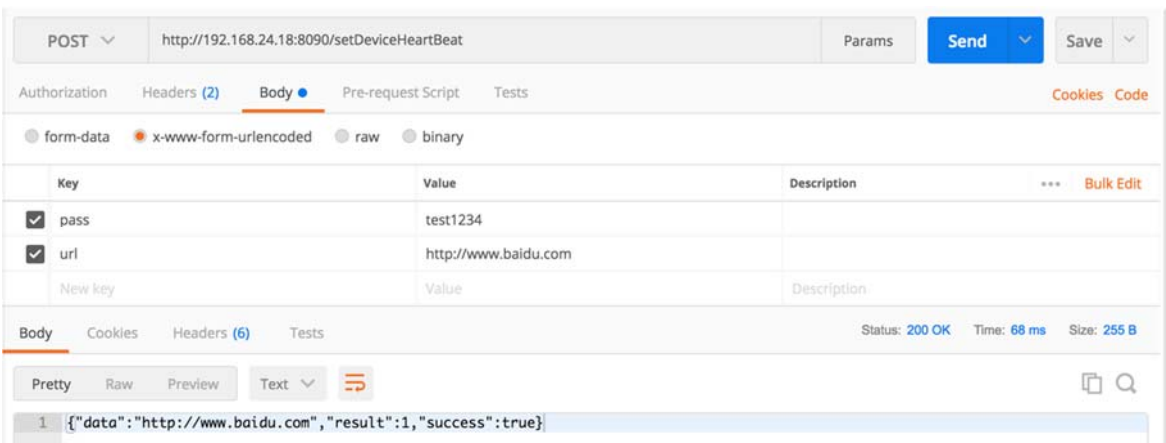
## 2.17 设备心跳回调

### 请求数据

Method	URL			
POST	http://设备 IP:8090/setDeviceHeartBeat			
参数名	描述	类型	必传	附加说明
pass	设备密码	String	Y	
url	回调地址	String	Y	<ul style="list-style-type: none"><li>设备每隔一分钟会向该接口 POST 字段 deviceKey、time、ip、personCount、faceCount 和 version, <a href="#">参见回调字段说明</a></li><li>传入内容为空可以清空回调地址, 清空后识别成功将不再进行回调</li><li>回调 URL 需要符合正则表达式: String check = "((http ftp https):/)(([a-zA-Z0-9\\._-]+\\.[a-zA-Z]{2,6}) ([0-9]{1,3}\\.[0-9]{1,3}\\.[0-9]{1,3}\\.[0-9]{1,3}))(:[0-9]{1,4})*/([a-zA-Z0-9\\&amp;%_\\./-~])*?"</li></ul>

 URL 端口说明: 若使用页面作为回调地址, 端口号须为 8090; 若使用 API 作为回调地址, 端口号最大为 9999, 且路由器对该端口号开放。


### postman 示例



返回示例

```
{
  "data": "http://www.baidu.com",
  "result": 1,//接口调通
  "success": true//回调地址设置成功
}
```

设备心跳回调参数字段说明

 注意：回调数据在 url 和 body 中各有一份，url 中数据是键值对形式，body 中数据是 Json 格式。

参数名	类型	描述
deviceKey	String	设备唯一标识码
time	String	设备当前时间戳
ip	String	设备当前 IP 地址
personCount	String	设备当前注册人员数量
faceCount	String	设备当前注册的照片数量
version	String	设备版本号

## 3 人员管理类接口

### 3.1 人员注册

#### 请求数据

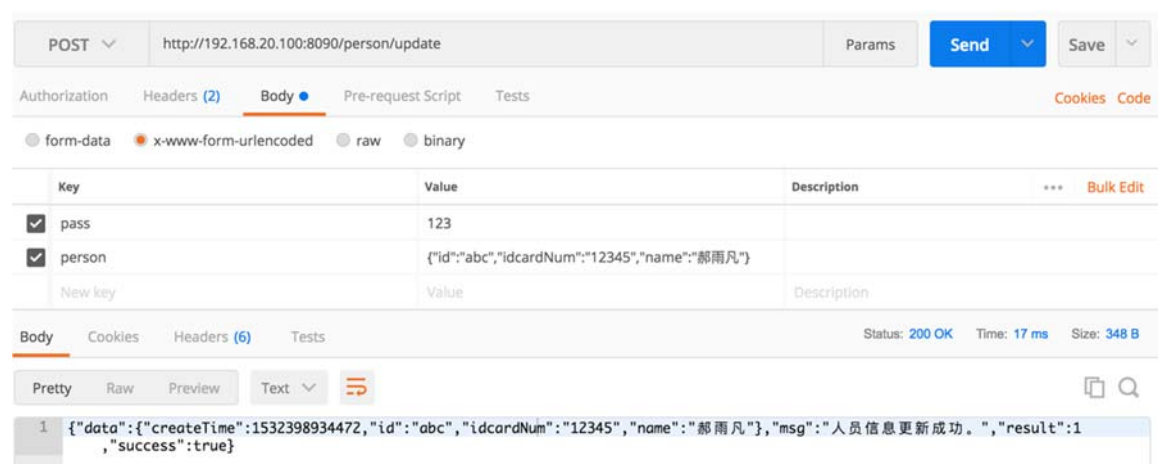
Method	URL			
POST	http://设备 IP:8090/person/create			
参数名	描述	类型	必传	附加说明
pass	设备密码	String	Y	
person	人员信息集合	Json	Y	<ul style="list-style-type: none"><li>Json 示例： {"id":"Abc12345","idcardNum":"410822199908221428","name":"郝雨凡"}</li><li>id、idcardNum 内容可以传空，如： {"id":"","idcardNum":"","name":"郝雨凡"}</li><li>若传入 id 值，id 只允许数字和英文字母，区分大小写</li></ul>

#### postman 示例

 注意：a) POST 请求参数放在 body 里

b) ContentType: application/x-www-form-urlencoded


c) 本章所有示例均使用 postman





## 返回示例


```
{
  "data": {
    "createTime": 1532398934472, //String, 人员创建时的时间戳（毫秒级），以设备时间为准
    "id": "abc", //String, 人员 id，id 为人员标识，限制 64 位。注册人员时，若 id 不填，则系统会自动生成 32 位 id 并返回；若填了 id，系统会以此 id 为主键存入本地数据库中；若 id 重复，会报错
    "idcardNum": "410822199908221428", //String, 卡号，注册时可以不填，长度无限制。若注册人员时填写了卡号，可直接刷对应卡号的卡进行识别，屏幕也会显示与该卡号对应的人员的名字
    "name": "郝雨凡", //String, 姓名，不可为空。识别成功后会在屏幕上显示该名字
  },
  "msg": "人员信息添加成功",
  "result": 1, //接口调通
  "success": true //人员注册成功
}
```

 注意：姓名、卡号、id 注册成功，实际上是人员信息写入设备数据库。若使用设备读取卡号进行注册，实际上是先读取卡号，再将卡号写入数据库，因此与注册人员时直接填写卡号效果一样。

### 3.2 人员卡号注册


#### 请求数据

Method	URL			
POST	http://设备 IP:8090/face/icCardRegist			
参数名	描述	类型	必传	附加说明
pass	设备密码	String	Y	
personId	人员 id	String	Y	<ul style="list-style-type: none"><li>给指定人员 id 注册卡号</li><li>人员 id 必须为已存在；若不存在该人员 id，则无法进入卡号注册模式</li><li>注册成功后，可调用人员信息查询接口查询该人员的卡号</li></ul>

 注意：卡号注册的过程实际上是设备先读取卡号，再将卡号写入人员信息数据库（两个过程合称录入卡号），因此与人员注册时直接填写卡号效果一致。[参见人员注册](#)。

#### postman 示例

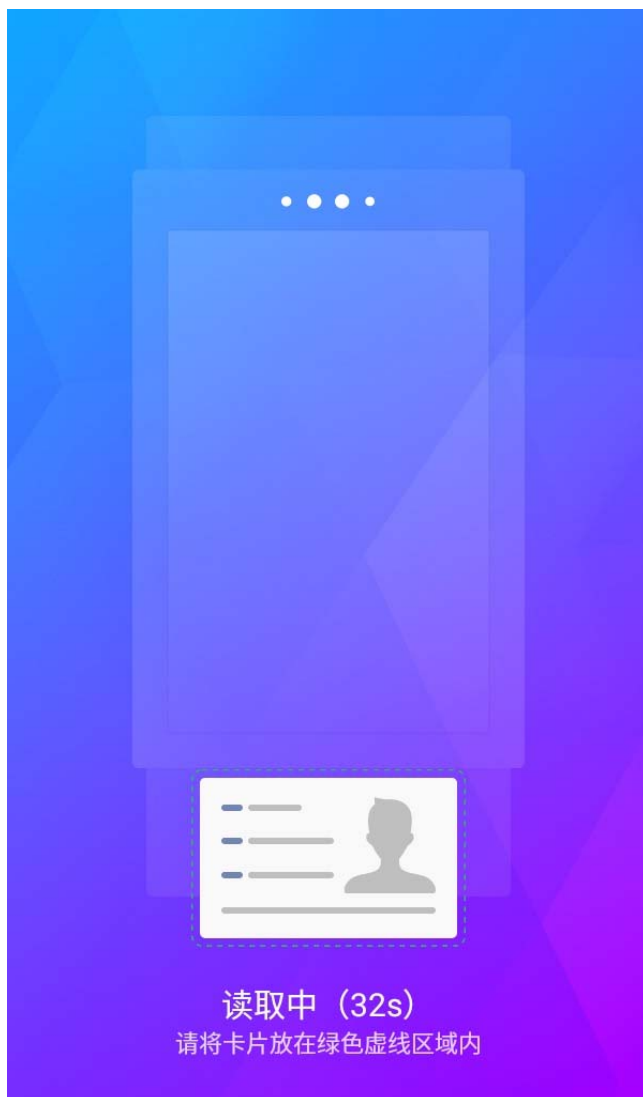


 注意：接口调用成功后，设备开始执行录入卡号的指令。录入卡号过程是否成功需查看设备界面，或通过卡号是否写入人员信息库判断卡号注册是否成功。

## 返回示例

```
{  
  "msg": "正在开启 IC 卡注册模式，注册成功后可根据 personId 查询该人的 IC 卡 ID。请根据  
引导完成注册。",  
  "result": 1,//接口调通  
  "success": true//设备开始执行录入卡号的指令  
}
```

## 设备界面说明



### 3.3 人员更新

#### 请求数据

Method	URL			
POST	http://设备 IP:8090/person/update			
参数名	描述	类型	必传	附加说明
pass	设备密码	String	Y	
person	人员信息集合	Json	Y	<ul style="list-style-type: none"><li>Json 示例: {"id":"Abc12345","idcardNum":"410822199908221428","name":"郝雨凡"}</li><li>id、name 不可传空</li><li>idcardNum 传空则会清空原信息</li></ul>

#### postman 示例



## 返回示例

```
{
  "data": {
    "createTime": 1532398934472, //String, 人员创建时的时间戳（毫秒级），以设备时间为准。注意不是人员更新时的时间
    "id": "abc", //人员 id， 人员唯一标识。更新时根据人员 id 修改设备内数据库信息
    "idcardNum": "12345", //卡号， 传空则清空
    "name": "新名字" //姓名， 不可传空
  },
  "msg": "人员信息更新成功。",
  "result": 1, //接口调通
  "success": true //人员信息更新成功
}
```

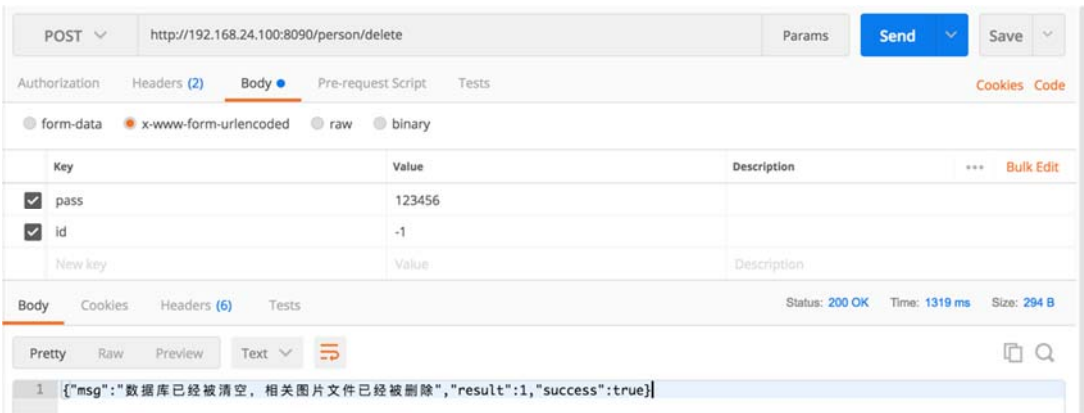
### 3.4 人员删除（批量）

#### 请求数据

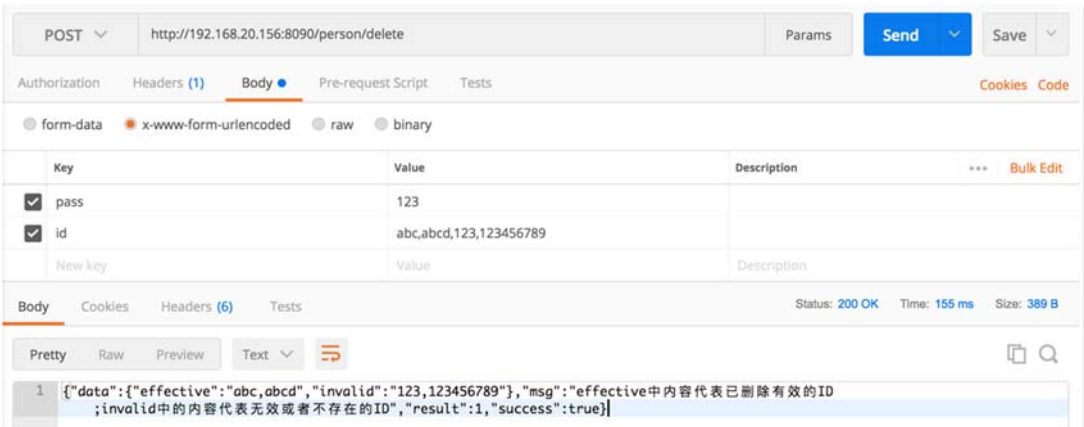
Method	URL			
POST	http://设备 IP:8090/person/delete			
参数名	描述	类型	必传	附加说明
pass	设备密码	String	Y	
id	人员 id	String	Y	<ul style="list-style-type: none"><li>删除多个人员，personId 用英文逗号拼接</li><li>传入-1 则删除所有人员</li></ul> 以下数据会被删除，本地将不再做存储： <ul style="list-style-type: none"><li>该人员 id、人员信息</li><li>该人员对应的识别记录、现场抓拍照片</li><li>该人员对应的注册照片</li></ul>

#### postman 示例

(1) id 传入-1，删除设备上的所有人员，数据不可恢复



(2) id 传入多个人员，用英文逗号拼接；返回结果区分已删除的 id 和无效 id。



## 返回示例


```
{
  "data": {
    "effective": "abc,abcd",
    "invalid": "123,123456789"
  },
  "msg": "effective 中内容代表已删除有效的 ID;invalid 中的内容代表无效或者不存在的 ID",
  "result": 1,//接口调通
  "success": true//已删除有效的人员 id
}
```

### 3.5 人员分页查询

#### 请求数据

Method	URL			
GET	http://设备 IP:8090/person/findByPage			
参数名	描述	类型	必传	附加说明
pass	设备密码	String	Y	
personId	人员 id	String	Y	<ul style="list-style-type: none"><li>• 查询指定 id 的人员信息</li><li>• id 传入-1 表示不局限于 personId 查询人员</li></ul>
length	每页最大数量	Int	N	<ul style="list-style-type: none"><li>• length 传入值要求为(0,1000]之间的正整数</li><li>• 若 length 不传，默认值为 1000，返回最近添加的 1000 人</li></ul>
index	页码	Int	N	页码，从 0 开始

#### 不同情况说明

 注意：a) 使用 postman 时，GET 请求参数放在 url 里，点击“Params”添加参数。

b) 此接口与人员信息查询接口功能类似，建议使用此接口进行分页，避免设备注册人数较多时查询导致设备崩溃。

- **情况一：**peronId 传入指定人员 id，length、index 无意义，可不传，即时传入值也不会生效。



postman 示例

GET

http://192.168.24.4:8090/person/findByPage?pass=test1234&personId=abcd

Params

Send

Save

Key	Value	Description
<input checked="" type="checkbox"/> pass	test1234	
<input checked="" type="checkbox"/> personId	abcd	
<input type="checkbox"/> length	1	
<input type="checkbox"/> index	0	
New key	Value	Description

Authorization

Headers

Body

Pre-request Script

Tests

Cookies

Code

Type

No Auth

Body

Cookies

Headers (6)

Tests

Status: 200 OK

Time: 92 ms

Size: 310 B

Pretty

Raw

Preview

Text

1

```
{
  "data": [
    {
      "createTime": 1530157887332,
      "id": "abcd",
      "idcardNum": "36222",
      "name": "test"
    }
  ],
  "result": 1,
  "success": true
}
```

## 返回示例

```
{
  "data": [{
    "createTime": 1530157887332, //人员注册成功、写入数据库的时间，Unix 毫秒时间戳；
    //较低版本人员注册成功后没有时间戳一项，因此更新到新版本后，会自动对数据库内所有
    //createTime 为空的人员进行自动填充，以设备重启时的时间为基础，按照人员写入数据库的顺
    //序依次增加 createTime（人员时间间隔无规律）。
    "id": "abcd", //人员 id，限制 64 位
    "idcardNum": "36222", //卡号，长度无限制
    "name": "test" //人员姓名，不可为空
  }],
  "result": 1, //接口调通
  "success": true //查询人员信息成功返回
}
```

- 情况二：personId 传入-1，length 和 index 可不传，length 不传默认为 1000，index 不传默认为 0。

## postman 示例

GET http://192.168.24.4:8090/person/findByPage?pass=test1234&personId=-1

Key	Value	Description
<input checked="" type="checkbox"/> pass	test1234	
<input checked="" type="checkbox"/> personId	-1	
<input type="checkbox"/> length		
<input type="checkbox"/> index		
New key	Value	Description

Authorization: No Auth

Body: Cookies Headers (6) Tests

Status: 200 OK Time: 59 ms Size: 647 B

Pretty Raw Preview Text

```
1 {
  "data": {
    "pageInfo": {
      "index": 0,
      "length": 1000,
      "size": 1,
      "total": 4
    },
    "personInfos": [
      {
        "createTime": 1530157887332,
        "id": "abcd",
        "idcardNum": "36222",
        "name": "test"
      },
      {
        "createTime": 1530157887316,
        "id": "12345abc",
        "idcardNum": "410822199908221428",
        "name": "郝雨凡"
      },
      {
        "createTime": 1530157887306,
        "id": "abc",
        "idcardNum": "36222",
        "name": "test"
      },
      {
        "createTime": 1530157887292,
        "id": "12345",
        "idcardNum": "410822199908221428",
        "name": "郝雨凡"
      }
    ]
  },
  "result": 1,
  "success": true
}
```

## 返回示例

```
{
  "data": {
    "pageInfo": {
      "index": 0, //页码。当前数据所在页码，从 0 开始计数。当 personId 传入-1 时，若
      index 不传默认为 0
      "length": 1000, //每页最大数量，允许范围为(0,1000]。当 personId 传入-1 时，若
      length 不传默认为 1000
      "size": 1, //总页数。因 index 从 0 开始计数，因此 index < size
      "total": 4 //总数据条数，设备上的总注册人数（不受 length 限制）
    },
    "personInfos": [{
      "createTime": 1530157887332, //人员注册成功、写入数据库的时间，Unix 毫秒时
      间戳；其他说明同上
      "id": "abcd", //人员 id，限制 64 位
      "idcardNum": "36222", //卡号，长度无限制
      "name": "test" //姓名，不可为空
    }, {
      "createTime": 1530157887316,
      "id": "12345abc",
      "idcardNum": "410822199908221428",
      "name": "郝雨凡"
    }, {
      "createTime": 1530157887306,
      "id": "abc",
      "idcardNum": "36222",
      "name": "test"
    }, {
      "createTime": 1530157887292,
      "id": "12345",
      "idcardNum": "410822199908221428",
      "name": "郝雨凡"
    }
  ]
},
  "result": 1, //接口调通
  "success": true //查询人员信息返回成功
}
```

- 情况三： personId 传入-1，若参数传入 length，传入值务必为（0，1000]之间的正整数，index 若不传默认从 0 开始。

## postman 示例

The screenshot shows a Postman interface for a GET request. The URL is `http://192.168.24.4:8090/person/findByPage?pass=test1234&personId=-1&length=1&index=0`. The parameters are defined in a table below:

Key	Value	Description
<input checked="" type="checkbox"/> pass	test1234	
<input checked="" type="checkbox"/> personId	-1	
<input checked="" type="checkbox"/> length	1	
<input checked="" type="checkbox"/> index	0	
New key	Value	Description

The request is sent with the following status and details:

- Status: 200 OK
- Time: 63 ms
- Size: 379 B

The response body is shown in the 'Body' tab, displaying a JSON object in 'Pretty' format:

```
1 {"data":{"pageInfo":{"index":0,"length":1,"size":4,"total":4},"personInfos":[{"createTime":1530157887332,"id":"abcd","idcardNum":"36222","name":"test"}]},{"result":1,"success":true}}
```

返回示例


Method	URL			
GET	http://设备 IP:8090/person/find			
参数名	描述	类型	必传	附加说明
pass	设备密码	String	Y	
id	人员 id	String	Y	<ul style="list-style-type: none"><li>• 查询指定 id 的人员信息</li><li>• id 传入-1 为查询设备上所有人员信息</li></ul>

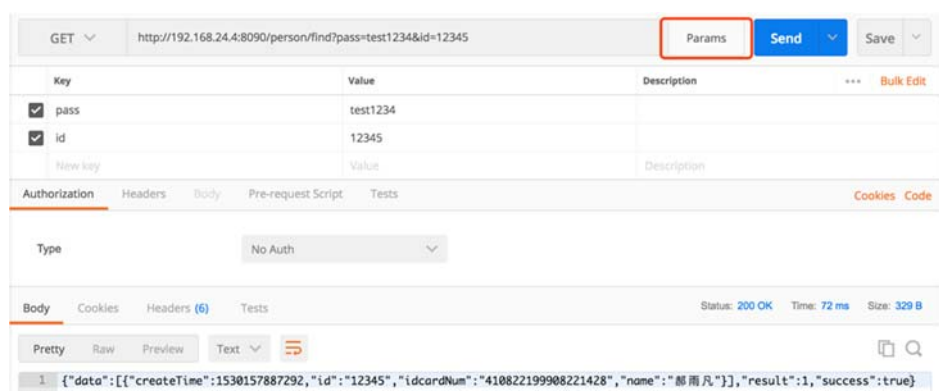
```
{
  "data": {
    "pageInfo": {
      "index": 0, //页码。当前数据所在页码，从 0 开始计数。当 personId 传入-1 时，若 index 不传默认为 0
      "length": 1, //每页最大数量，允许范围为(0,1000]。当 personId 传入-1 时，若 length 不传默认为 1000
      "size": 4, //总页数。因 index 从 0 开始计数，因此 index < size
      "total": 4 //总数据条数，设备上的总注册人数（不受 length 限制）
    },
    "personInfos": [{
      "createTime": 1530157887332, //人员注册成功、写入数据库的时间，Unix 毫秒时间戳；其他说明同上
      "id": "abcd", //人员 id，限制 64 位
      "idcardNum": "36222", //卡号，长度无限制
      "name": "test" //姓名，不可为空
    }]
  },
  "result": 1, //接口调通
  "success": true //成功查询到人员信息并返回
}
```

3.6 人员信息查询

请求数据

postman 示例

 注意：此接口与人员分页查询接口功能相似，建议使用[人员分页查询接口](#)，避免设备注册人数较多时查询导致设备崩溃。



## 返回示例

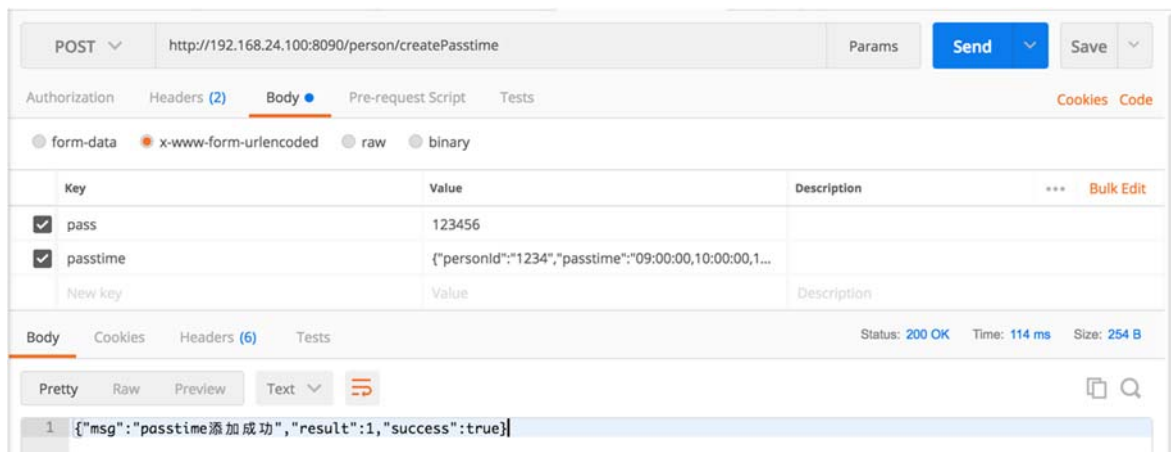
```
{
  "data": [{
    "createTime": 1530157887292, //人员注册成功、写入数据库的时间，Unix 毫秒时间戳；
    //较低版本人员注册成功后没有时间戳一项，因此更新到新版本后，会自动对数据库内所有
    //createTime 为空的人员进行自动填充，以设备重启时的时间为基础，按照人员写入数据库的顺序
    //依次增加 createTime（人员时间间隔无规律）。
    "id": "12345", //人员 id，限制 64 位
    "idcardNum": "410822199908221428", //卡号，长度无限制
    "name": "郝雨凡" //姓名，不可为空
  }],
  "result": 1, //接口调通
  "success": true //成功查询到人员信息并返回
}
```

## 3.7 时间段权限设置

请求数据

Method	URL			
POST	http://设备 IP:8090/person/createPasstime			
参数名	描述	类型	必传	附加说明
pass	设备密码	String	Y	
passtime	某人员每日允许进入的时间段	Json	Y	<ul style="list-style-type: none"><li>• Json 示例： <pre>{"personId":"9eecc839cd7941c5a4d3165202dd3c32","passtime":"09:00:00,10:00:00,17:00:00,17:30:00,18:30:00,20:25:00"}</pre></li><li>• 范围为[00:00:00,23:59:59]，以设备上的时间为标准</li><li>• 时间段格式（startTime,endTime 英文逗号隔开）：  09:00:00,11:00:00,13:00:00,15:00:00,17:00:00,19:00:00</li><li>• passtime 最多可设置 3 段，若只设置 1 段，则后两段不传即可，如：  09:00:00,11:00:00</li></ul>

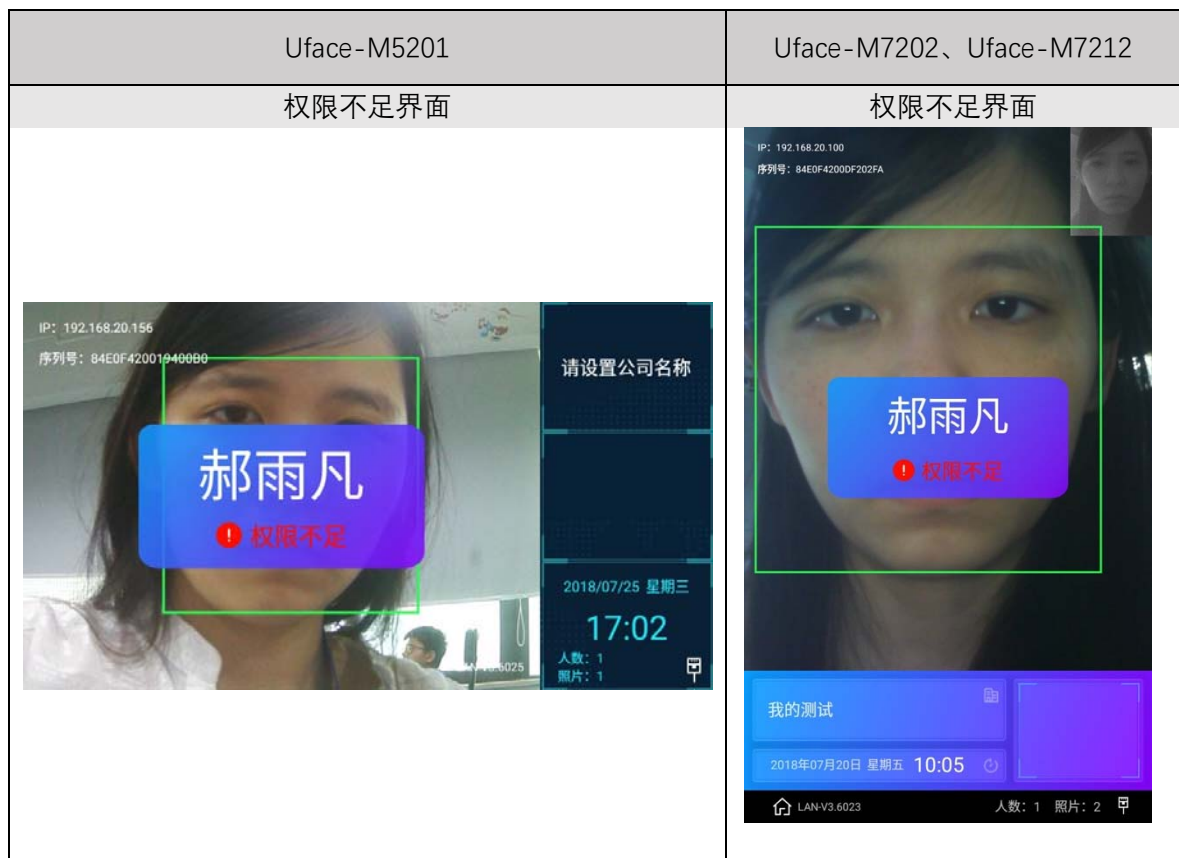
postman 示例



**i** 注意：a) 若要更新人员的 passtime，可再次调用时间段权限设置接口，重新传入 passtime。

b) passTime 报错类型：时间段参数数量不正确或超出 3 段限制、时间段参数后时间段早于前时间段、时间段参数超出限制、时间段参数格式错误。

c) 人员在非允许的时间段内识别，设备会播报“姓名权限不足”，设备屏幕会显示“姓名+权限不足”，如下图所示。





## 返回示例

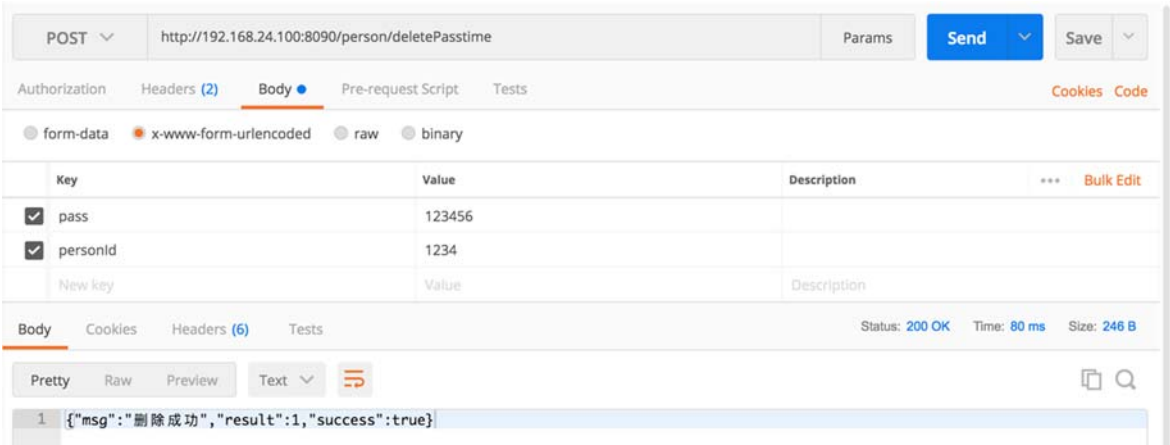
```
{  
  "msg": "passtime 添加成功",  
  "result": 1,//接口调通  
  "success": true//人员 passtime 设置成功  
}
```

### 3.8 时间段权限删除

#### 请求数据

Method	URL			
POST	http://设备 IP:8090/person/deletePasstime			
参数名	描述	类型	必传	附加说明
pass	设备密码	String	Y	
personId	人员 id	String	Y	<ul style="list-style-type: none"><li>删除该人员的时间段权限设置，该人员不再有时间段权限限制</li><li>传入-1，可清除所有人员的 passtime</li></ul>

#### postman 示例



#### 返回示例


```
{
  "msg": "删除成功",
  "result": 1,//接口调通
  "success": true//人员 passtime 删除成功
}
```

## 3.9 人员有效期限设置

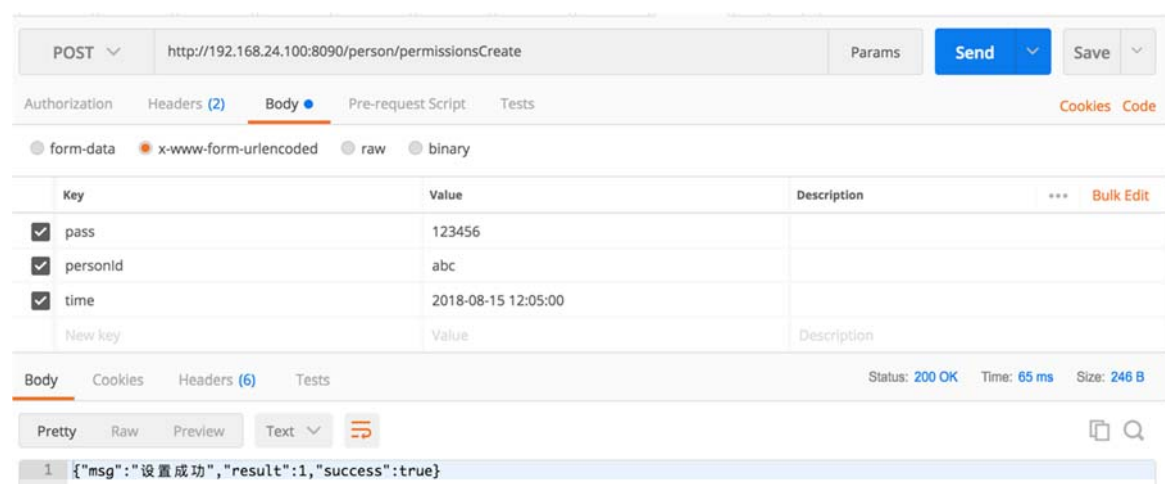
### 请求数据

Method	URL			
POST	http://设备 IP:8090/person/permissionsCreate			
参数名	描述	类型	必传	附加说明
pass	设备密码	String	Y	
personId	人员 id	String	Y	定时删除人员后，以下数据会被删除，本地将不再做存储： <ul style="list-style-type: none"><li>• 该人员 id、人员信息</li><li>• 该人员对应的识别记录、现场抓拍照拍</li><li>• 该人员对应的注册照片</li></ul>
time	有效期限时间，定时删除该人员	String	Y	<ul style="list-style-type: none"><li>• 以设备系统时间为准，当人员权限时间到期时，人员将在 5 秒内进行删除</li><li>• 传入时间格式为（年-月-日 时:分:秒）： <b>2017-07-15 12:05:00</b></li></ul>

### postman 示例

 注意：a) 若传入的时间早于设备系统当前时间，则人员会在 5 秒内进行删除。

b) 若人员已设置了权限时间，则再次设置会报错；需先删除权限，再重新设置权限时间（[参见有效期限设置删除](#)）。



## 返回示例

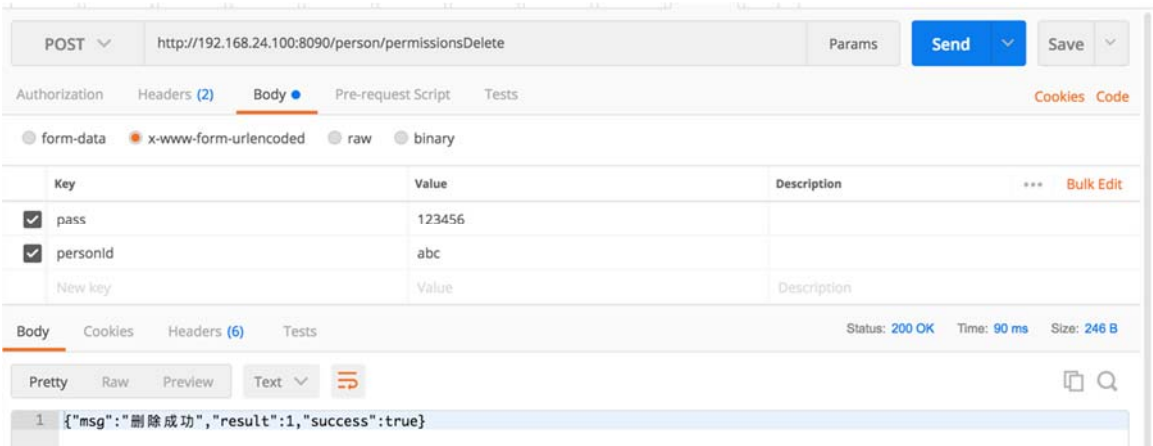
```
{  
  "msg": "设置成功",  
  "result": 1,//接口调通  
  "success": true//人员有效期限设置成功  
}
```

### 3.10 有效期限设置删除

#### 请求数据

Method	URL			
POST	http://设备 IP:8090/person/permissionsDelete			
参数名	描述	类型	必传	附加说明
pass	设备密码	String	Y	
personId	人员 id	String	Y	<ul style="list-style-type: none"><li>删除该人员的有效期限设置，该人员不会被定时删除</li><li>若该人员未设置过有效期限，则调用此接口会报错</li></ul>

#### postman 示例



#### 返回示例

```
{
  "msg": "删除成功",
  "result": 1, //接口调通
  "success": true //人员有效期限设置删除成功
}
```

## 4 照片管理类接口

### 4.1 照片注册（base64）

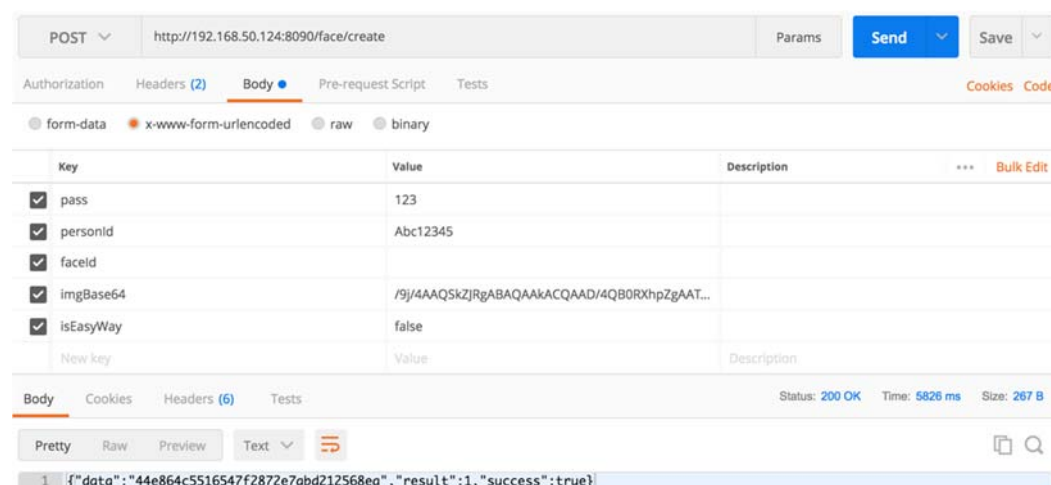
#### 请求数据

Method	URL			
POST	http://设备 IP:8090/face/create			
参数名	描述	类型	必传	附加说明
pass	设备密码	String	Y	
personId	人员 id	String	Y	<ul style="list-style-type: none"><li>• 用于标识该照片属于某个人 id</li><li>• 必须先注册人员，才能添加照片</li></ul>
faceId	照片 id	String	Y	若 faceId 传入内容为空，则系统会自动生成一个 32 位的 faceId 并在照片注册成功后返回
imgBase64	照片的 base64 编码字符串	String	Y	<ul style="list-style-type: none"><li>• 不加头部，如：data:image/jpg;base64,</li><li>• 图片格式支持 png、jpg、jpeg</li></ul>
isEasyWay	选择宽松 or 严格照片注册方式	Boolean	N	非必传，默认 false：严格检测照片质量；true：宽松检测照片质量

**i** 注意：a) 照片注册成功后，设备会将该照片保存到设备中，方便外部后续对注册照进行查找和其他用途。

b) 若设置了注册照片回调地址，注册成功后设备会将注册照信息通过 POST 请求方式回调给预先设置的回调地址，[参见注册照片回调](#)。

#### postman 示例




## 返回示例

```
{  
  "data": "d7fe98be463c4d06a70605d4d58b5f33", //照片 id, 每张注册照都有一个 id, 限制 32  
  位。若 faceId 不传, 由系统分配的 faceId 为 32 位  
  "result": 1, //接口调通  
  "success": true //照片注册成功  
}
```

## 4.2 照片注册（url）

### 请求数据

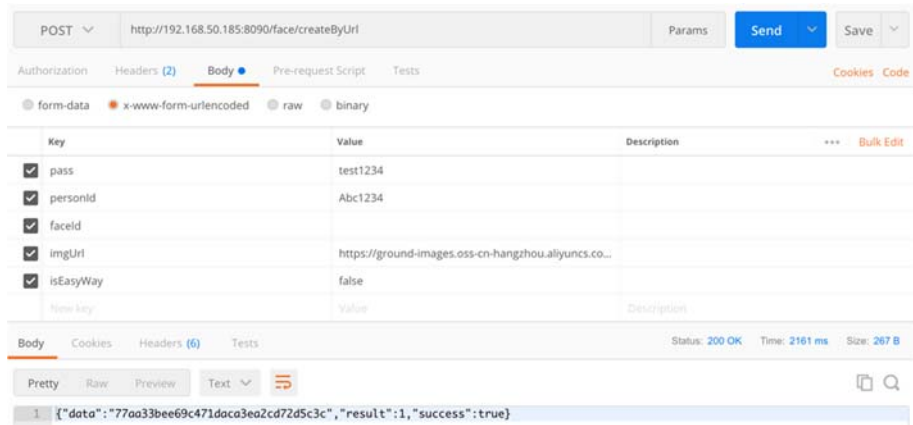
Method	URL			
POST	http://设备 IP:8090/face/createByUrl			
参数名	描述	类型	必传	附加说明
pass	设备密码	String	Y	
personId	人员 id	String	Y	<ul style="list-style-type: none"><li>• 用于标识该照片属于某个人 id</li><li>• 必须先注册人员，才能添加照片</li></ul>
faceId	照片 id	String	Y	若 faceId 传入内容为空，则系统会自动生成一个 32 位的 faceId 并在照片注册成功后返回
imgUrl	照片的 base64 编码字符串	String	Y	<ul style="list-style-type: none"><li>• 先通过 url 下载图片到本地，然后从图片中提取特征</li><li>• 只支持 http 协议存储位置的图片，不支持本地路径</li><li>• 图片格式支持 png、jpg、jpeg</li></ul>
isEasyWay	选择宽松 or 严格照片注册方式	Boolean	N	非必传，默认 false：严格检测照片质量；true：宽松检测照片质量

 注意：a) 照片注册成功后，设备会将该照片保存到设备中，方便外部后续对注册照进行查找和其他用途。

b) 若设置了注册照片回调地址，注册成功后设备会将注册照信息通过 POST 请求方式回调给预先设置的回调地址，[参见注册照片回调](#)。

### postman 示例





## 返回示例

```
{
  "data": "061ce9d000c44d98818bf69021b916ec", //照片 id, 每张注册照都有一个 id, 限制 32
  位。若 faceId 不传, 由系统分配的 faceId 为 32 位
  "result": 1, //接口调通
  "success": true //照片注册成功
}
```

## 4.3 拍照注册

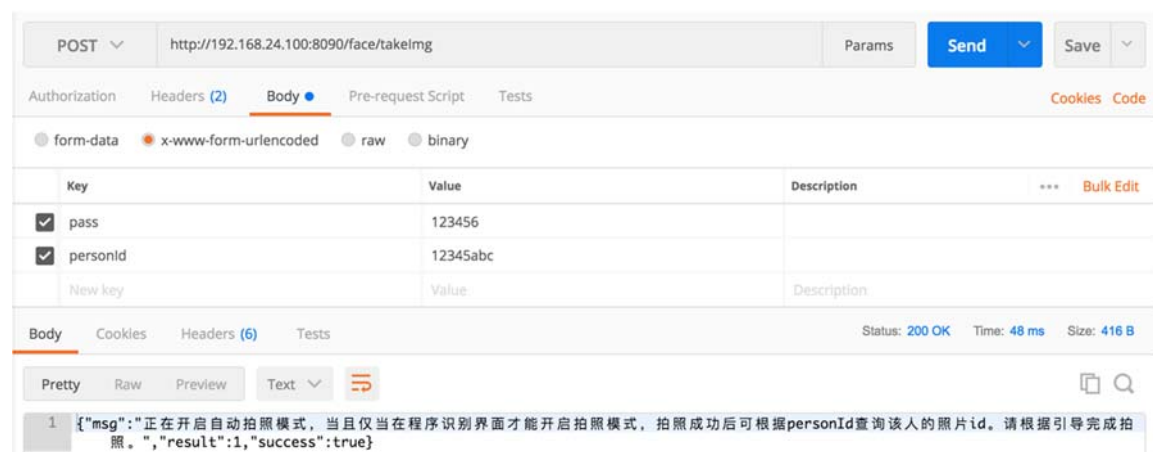
### 请求数据

Method	URL			
POST	http://设备 IP:8090/face/takelmng			
参数名	描述	类型	必传	附加说明
pass	设备密码	String	Y	
personId	人员 id	String	Y	<ul style="list-style-type: none"><li>给指定人员 id 拍摄照片</li><li>人员 id 必须为已存在；若不存在该人员 id，则无法进入自动拍照模式</li></ul>

**i** 注意：a) 拍照注册的过程实际上是设备先拍摄照片，再调用照片注册接口注册照片，faceId 由设备随机生成。

b) 若设置了注册照片回调地址，拍照注册成功后设备会将注册照信息通过 POST 请求方式回调给预先设置的回调地址，[参见注册照片回调](#)。

### postman 示例



**i** 注意：接口调用成功后，设备开始执行拍照指令。拍摄过程是否成功需查看设备界面，或通过照片是否生成判断拍照是否成功。

返回示例

```
{
  "msg": "正在开启自动拍照模式，当且仅当在程序识别界面才能开启拍照模式，拍照成功后  
可根据 personId 查询该人的照片 id。请根据引导完成拍照。",
  "result": 1,//接口调通
  "success": true//设备开始执行拍照指令
}
```

设备界面说明

Uface-M5201	Uface-M7202、Uface-M7212
设备拍照模式界面	设备拍照模式界面
	

## 4.4 照片查询

### 请求数据

Method	URL			
POST	http://设备 IP:8090/face/find			
参数名	描述	类型	必传	附加说明
pass	设备密码	String	Y	
personId	人员 id	String	Y	查询该人员的所有注册照片

### postman 示例

POSThttp://192.168.24.100:8090/face/findParamsSendSave

form-data x-www-form-urlencodedrawbinary

Key	Value	Description
pass	123456	
personId	12345abc	
New key	Value	Description

BodyCookiesHeaders (6)Tests

Status: 200 OKTime: 89 msSize: 2.04 KB

PrettyRawPreviewText

```
1 [{"data": [{"faceId": "35f43f52bc54490899204592c691eb69", "feature": "L04AAAhLvLgMAgAAwbLaPflJ
+ jxH84i70uL5vZfwZj1f8Sq9AnudvUEgrj1WAzw7rXpqvPdnDr0qm7M97tHSvZu8Pj4w/NZ83guevU95WjxRLnY8b0ugPHQDLz31yO+9JUUrVU1fDz5SKpC81V
+ APcknND0TFks91p8FvgNmMT4z8Mk93WVjvUYGMT7dYfe94d4bPtUhcT6mcYe9MY77ve2xG77yweu9X8bbPYsdr7vDERO
+ Ag4LPQ44Hz1NLBK9a9iPvVWW7xmSjg8uXYJvU848z3c6TC6C3L6u9iZ671ZzaK6ecNAvaqM/WLuNg80
+ IeoEPWgRhr3hvxw9WqyCPQBHCdxsAk9SmRaPehYKz65drK8lFVhPYF5GLtCyWq9lmYsvaziWz3Jb0A96zASPeFPpz1S3eo9Q5ErPepFgbtNr4q69qkdvqmZsJz
gDTo+WWAdPLWpFb1HaXm9A0pfvbaqaL2kSeo9VXcXpnlq0b409C082Cjfvai0VT0G4r88R7n0vZ
/eM705sym9Wui6vD9XDL0N2zw8CyoMvibMAL4mZNa9PnQHPrC5jj0KYIY9e5/svVpIT4HixU9WgW/PYrJcz291+Q9Fuy0P0bA1Lzw7
+ K8S9tGvesQUj042pM9jeFZPVZ+Mr1loiU7DhmKPBubT1mt329DmfSvUfaa7yHqNo8xJs1PbY7abs=\u0000", "featureKey"
: "0d242189a664727d620e49fb679a8378", "path": "ftp://192.168.24.100:8010/faceRegister/12345abc_35f43f52bc54490899204592c691eb69
.jpg", "personId": "12345abc"}, {"faceId": "b8df54f2ef924c1f879e509b44593135", "feature"
: "L04AAPAxXlKMAgAA8t7VPY6kG7tQYa08YrzqvTjLAT5PERq8nGbUvSBvrj28oxY703awvKm+nb3kKQc94XTEvYWT7noVs9lSflvaKBDT2acE09rZYzvj
+ fgDzV+j+9amq1v0QxCz60PxW81YuZPZymMj03pfs9apd0vr/tJz5JPLk9dg7SvZfS5Yray9glEj78Mga+n1vbverh9L1sjRS9+bj2PVojswD/Aa
+ 99Qbuyl10j0rWTK97tBVPmodMb3HBm09nj8cv0NkIT7AqyU9tm34PGhzW70/XPS7JRz/veaZhr1acfa9YJC0P0C9QbpW6/u8S5e9Pcb+TT2T1Y
+ 8UC19PbNyGz7fy1E8M6AJPRIrg703y0s8vDC0vdpXFr1ziNE9R1jpyPhYsTz2onCg9M/Zx0TTEBTtUJz2+PQEkvpMz8TsVszo
+ 0gF0vbCjfb00L4K9mG8PvXyAqL0i8Zg965UzPpgdPL5dLC09eD4UvqOWUD3VCUA9tMCxvbyLujq3gN28p9v6vDHS1rbaE9GyKvVszbNr6tLxG
+ zP02Pc0WCr1hwRo9JMMKvrcFpT0WtZe7jW8ePmsJnTwK1mI9cHOKPZMF+TzgJZY8I1StvJxqojz64rE9zxW+PdKuQLZyWUM8VQFTPRKghD0j0WG9ZNa
+ v8berjwNmUk9PFuSPU1B0L0=\u0000", "featureKey": "5032611b1d1599f89bacc9e735ad8c7b", "path": "ftp://192.168.24.100:8010
/faceRegister/12345abc_b8df54f2ef924c1f879e509b44593135.jpg", "personId": "12345abc"}], "result": 1, "success": true}]
```


## 返回示例

```
{
  "data": [{
    "faceId": "b8df54f2ef924c1f879e509b44593135",//照片 id
    "feature": "L04AAPAxXLkMAgAA8t7VPY6kG7tQYaO8YrzqvTjLAT5PErq8nGbUvSBvrj28oxY7O3awvKm+nb3kKQc94XTEvYWtWT7noVs9ISFlvaKBDT2acE09rZYzvJ+fgDzV+j+9amq1vOQxCz60PxW81YuZPZymMj03pfs9apdOvr/tJz5JPLk9dg7SvZFfSz5Yray9gleUPYmIEj78Mga+n1vbverh9L1sjRS9+bj2PVojsjwD/Aa+99Qbuy110j0rWTK97tBVPmodMb3HBm09nj8cvONkIT7AqyU9tm34PGhzw70/XPS7JRz/veaZhr1acfa9YJC0P OC9QbpW6/u8S5e9Pcb+TT2T1Y+8UC19PbNyGz7Fy1E8M6AJPRlrg703y0s8vDCO vdpXFr1ziNE9R1jyPHYsTz2onCg9M/ZxOTTEBTtUJz2+PQEkvpMz8TsVSzo+OgF0v BcJfb0OL4K9mGBPvXyAqL0i8Zg965UzPpgdPL5dLC09eD4UvqOWUD3VCUA9tMC xvbyLujq3gN28p9v6vDHS HL1rbaE9GyKVvSz bNr6tlxG+zPO2Pc0WCr1hwRo9JMMK vrCfpT0WtZe7jW8ePmsJnTwK1mI9cHOKPZMf+TzgjZY8IISTvJxqojz64rE9zxW+PdK uQL2yWUM8VQFTPRKghD0j0WG9ZNa+vBberjwNmUk9PfuSPU1BOL0=",//特征码, 不可直接复制, 字符会产生转译。请使用变量赋值的方式使用
    "featureKey": "5032611b1d1599f89bacc9e735ad8c7b",//特征密钥 (加密)
    "path": "ftp://192.168.24.100:8010/faceRegister/12345abc_b8df54f2ef924c1f879e509b44593135.jpg",//注册照片存储在设备内的路径
    "personId": "12345abc"//人员 id
  }],
  "result": 1,
  "success": true
}
```

## 4.5 特征注册

### 请求数据

Method	URL			
POST	http://设备 IP:8090/face/featureReg			
参数名	描述	类型	必传	附加说明
pass	设备密码	String	Y	
personId	人员 id	String	Y	<ul style="list-style-type: none"><li>给指定人员 id 注册一个特征</li><li>人员 id 必须为已存在；若不存在该人员 id，则无法添加</li></ul>
faceId	照片 id	String	Y	<ul style="list-style-type: none"><li>若 faceId 传入内容为空，则系统会自动生成 faceId 并在特征注册成功后返回</li><li>照片更新时，以 faceId 为唯一标识进行更新</li></ul>
feature	特征码	String	Y	<ul style="list-style-type: none"><li>可由注册照片回调接收，也可由照片查询接口获得</li><li>注册一个特征相当注册于一张照片</li><li>每个人员 id 下，包括特征和照片，总数最多为 3</li><li>V3.6206 版本使用新的特征码，需清空设备内低版本遗留的人脸数据库，才能使用设备</li></ul>
featureKey	特征密钥	String	Y	<ul style="list-style-type: none"><li>特征注册成功后，由回调接收</li><li>特征有效性校验</li><li>V3.6 版本开始 featureKey 加密方式改变，从低版本设备上获取的 feature 和 featureKey 在 V3.6 版本会注册失败</li></ul>

 注意：a) 特征码注册等同于照片 base64 码注册，一个特征相当于一张注册照片。

b) 特征注册比 base64 注册速度更快，适用于快速注册。

c) 特征注册并非原图注册，因此无法查询到特征注册照片。若使用照片查询接口，特征 path 为空。

d) 特征的所有处理方式（删、改、查等）与注册照片一致。

e) 若设置了注册照片回调地址，特征注册成功后设备会将特征信息通过 POST 请求方式回调给预先设置的回调地址，[参见注册照片回调](#)。



## postman 示例

[illegible]

## 返回示例

[illegible]



[illegible]

[illegible]

[illegible]






## 4.6 照片更新

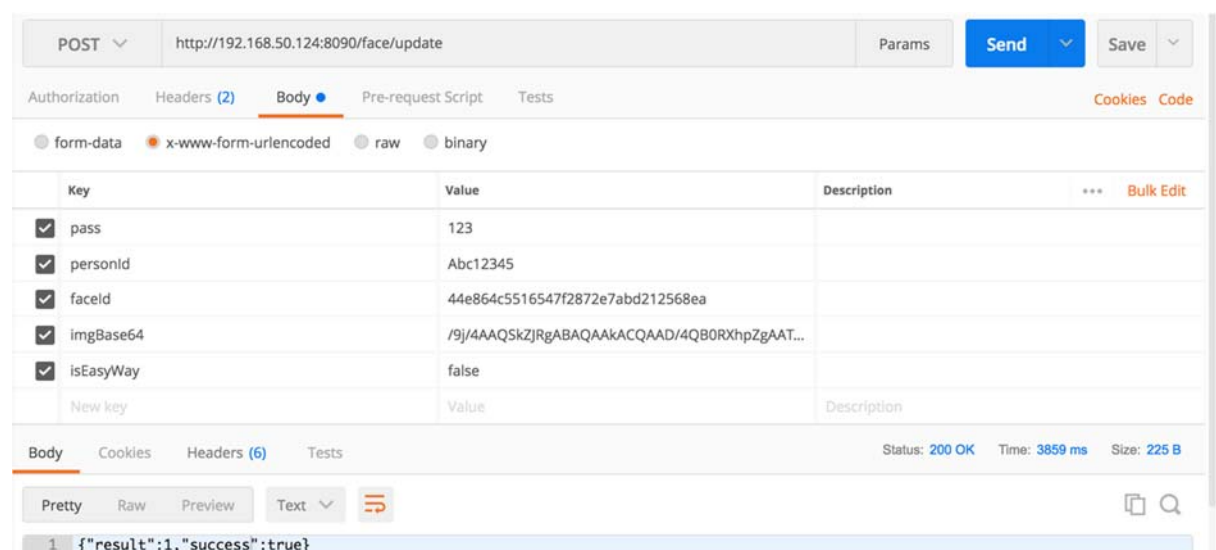
### 请求数据

Method	URL			
POST	http://设备 IP:8090/face/update			
参数名	描述	类型	必传	附加说明
pass	设备密码	String	Y	
personId	人员 id	String	Y	用于标识该照片属于某个人 id
faceId	照片 id	String	Y	更新该 faceId 对应的注册照片
imgBase64	照片的 base64 编码字符串	String	Y	不加头部，如：data:image/jpg;base64,
isEasyWay	选择宽松 or 严格照片注册方式	Boolean	N	非必传，默认 false：严格照片注册方式；true：宽松照片注册

 注意： a) 照片更新接口调用成功后，实际上是先删除原照片，再调用照片注册接口，使用该 faceId 注册照片。

b) 若设置了注册照片回调地址，照片更新成功后，也会触发注册照片回调。[参见注册照片回调](#)。

### postman 示例



## 返回示例

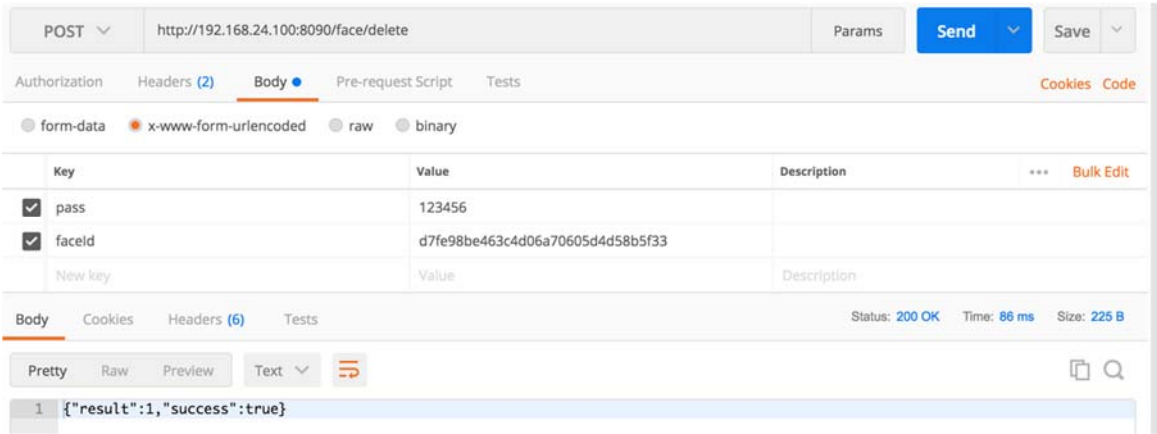
```
{  
  "result": 1,//接口调通  
  "success": true//照片更新成功  
}
```

## 4.7 照片删除

### 请求数据

Method	URL			
POST	http://设备 IP:8090/face/delete			
参数名	描述	类型	必传	附加说明
pass	设备密码	String	Y	
faceId	照片 id	String	Y	删除该 faceId 对应的注册照片，不可恢复

### postman 示例



### 返回示例

```
{
  "result": 1,//接口调通
  "success": true//照片删除成功
}
```

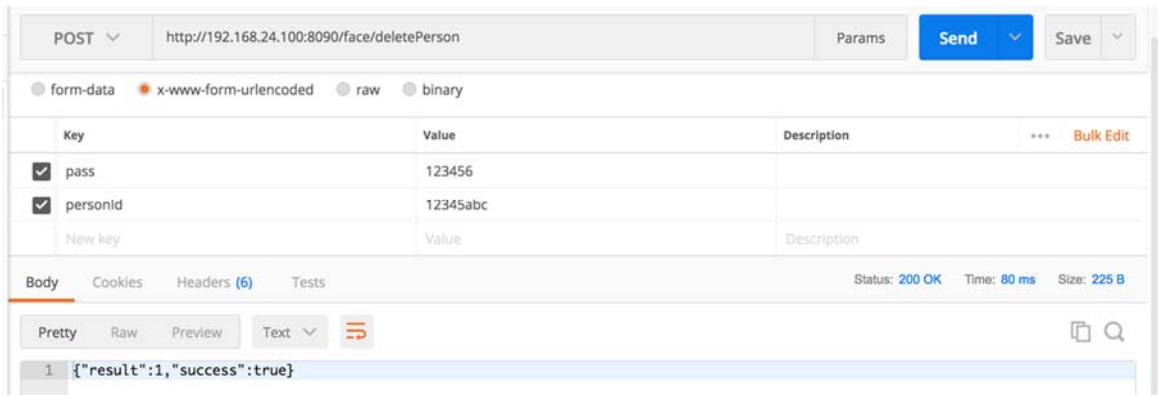


## 4.8 清空人员注册照片

### 请求数据

Method	URL			
POST	http://设备 IP:8090/face/deletePerson			
参数名	描述	类型	必传	附加说明
pass	设备密码	String	Y	
personId	人员 id	String	Y	调用该接口，该人员的所有注册照片 id 会注销，并同步删除存储在设备内的该人员所有注册照片

### postman 示例



### 返回示例

- 人员注册照片清空成功

```
{
  "result": 1,//接口调通
  "success": true//人员注册照片清空成功
}
```

- 人员没有注册照片

```
{  
  "msg": "没有这个人的照片",  
  "result": 1,//接口调通  
  "success": false//删除操作失败  
}
```

- 人员不存在

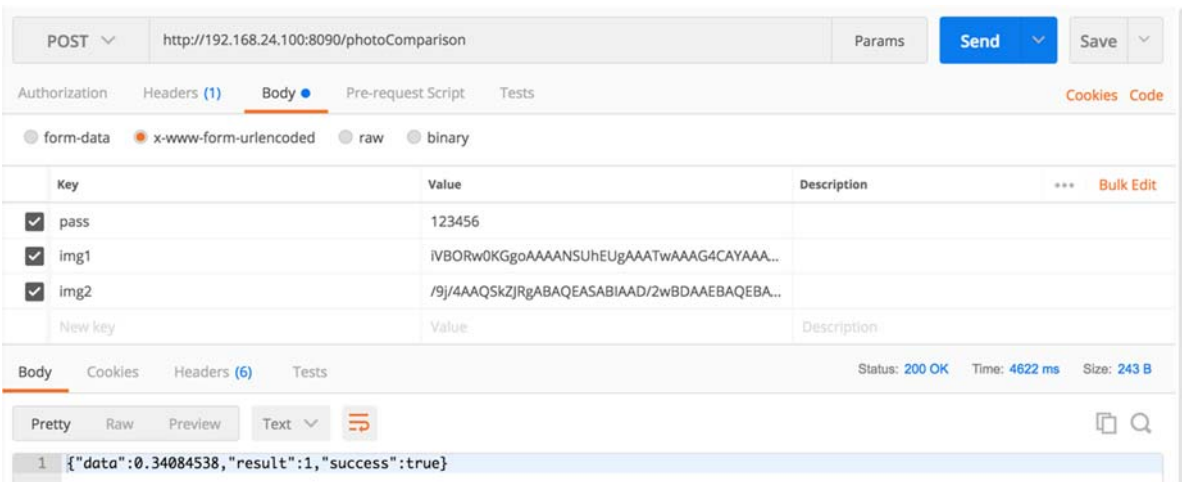
```
{  
  "msg": "人员不存在",  
  "result": 1,//接口调通  
  "success": false//删除操作失败  
}
```

## 4.9 人脸照片相似性比对

### 请求数据

Method	URL			
POST	http://设备 IP:8090/photoComparison			
参数名	描述	类型	必传	附加说明
pass	设备密码	String	Y	
img1	照片 1 的 base64 编码	String	Y	• 不加头部，如：data:image/jpg;base64, • 设备会对传入的两张照片进行人脸检测和相似度比对
img2	照片 2 的 base64 编码	String	Y	

### postman 示例



### 返回示例

#### • 两张照片中人脸规范

```
{
  "data": 0.34084538, // 两张照片中人脸的相似度比对结果
  "result": 1, // 接口调通
  "success": true // 人脸照片比对成功
}
```

- 照片中人脸不规范

```
{  
  "msg": "FaceCoreException{expCode=7,expDesc=' 照 片 中 未 检 测 到 人 脸 ',  
attachedInfo=EXP-FACE-3}",  
  "result": 1,//接口调通  
  "success": true//人脸照片比对成功  
}
```

## 5 识别记录管理接口

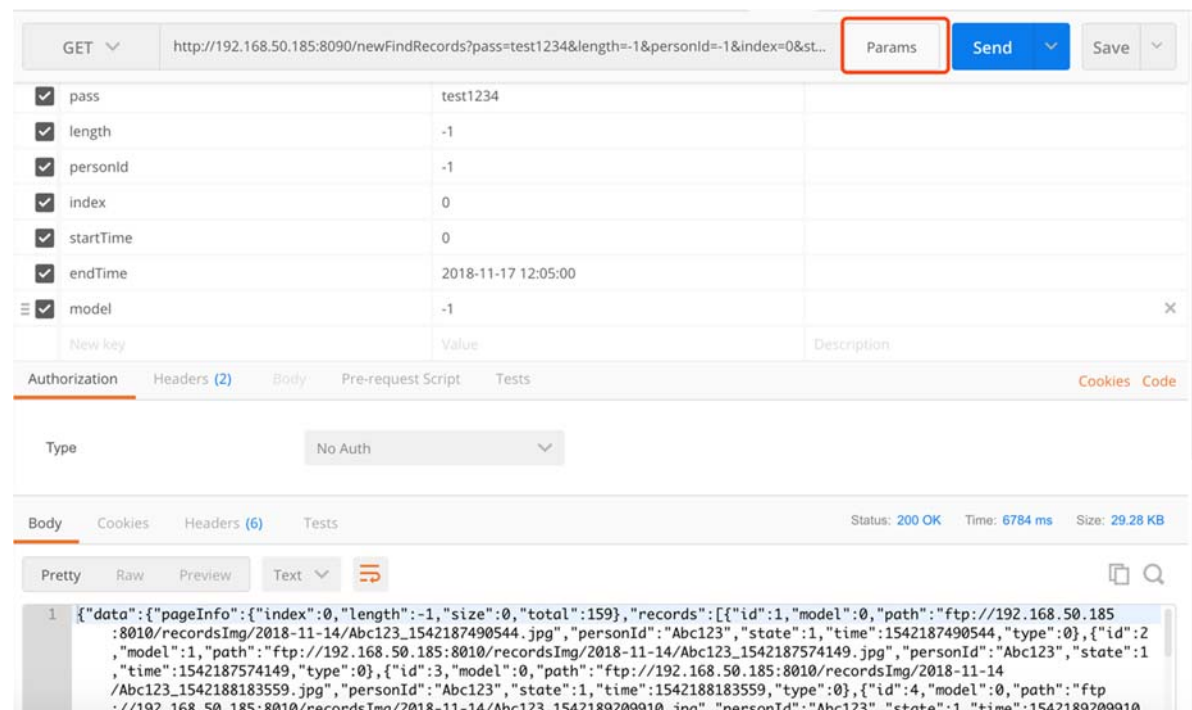
### 5.1 识别记录查询

#### 请求数据

Method	URL			
GET	http://设备 IP:8090/newFindRecords			
参数名	描述	类型	必传	附加说明
pass	设备密码	String	Y	
personId	人员 id	String	Y	<ul style="list-style-type: none"><li>• 查询指定 id 的人员识别记录</li><li>• 传入-1 可查询所有人员的识别记录，包括陌生人</li><li>• 传入 STRANGERBABY，可查询所有陌生人/识别失败记录</li><li>• 传入 IDCARD，可查询所有人证比对记录</li></ul>
length	每页最大数量	Int	Y	<ul style="list-style-type: none"><li>• 传入-1 为不分页</li><li>• 若不传-1，请务必大于 0</li></ul>
index	页码	Int	Y	页码，从 0 开始
startTime	记录开始时间	Date	Y	<ul style="list-style-type: none"><li>• 若不按时间查询，请分别传入 0</li><li>• 若需要按时间查询，请按照如下格式（年-月-日 时:分:秒）： 2017-07-15 12:05:00</li></ul>
endTime	记录结束时间	Date	Y	
model	识别记录类型	Int	Y	<ul style="list-style-type: none"><li>-1: 所有类型的识别记录</li><li>0: 刷脸识别记录</li><li>1: 卡&amp;人脸双重认证识别记录</li><li>2: 人证比对识别记录</li><li>3: 刷卡识别记录</li></ul>

## postman 示例

**i** 注意：GET 请求参数放在 url 里，点击“Params”添加参数。



## 返回示例

```
{
  "data": {
    "pageInfo": {
      "index": 0, //页码。当前数据所在页码，从 0 开始计数
      "length": -1, //每页最大数量。-1 表示不分页，所有数据均显示在首页(index=0)
      "size": 0, //总页数。若 length=-1，size 默认为 0；若 length > 0，size ≥ 1
      "total": 3 //总数据条数
    },
    "records": [
      {
        "id": 1, //识别记录 id，由设备自动生成并管理。用户无须使用
        "model": 0, //识别模式，0：刷脸，1：卡&人脸双重认证，2：人证比对，3：刷卡
        "path": "ftp://192.168.50.32:8010/recordsImg/2018-11-14/Abc123_1542187490544.jpg", //现场抓拍照片存储在设备内的路径。当设备内现场照存储满 3G 时，会自动删除较早的 1G 现场照；若有需要请及时存储
        "personId": "Abc123", //人员 id。陌生人/识别失败显示 id 为 STRANGERBABY
        "state": 1, //回调结果。0：回调失败，1：回调成功或未设置回调地址
        "time": 1542187490544, //识别成功时的设备时间，Unix 毫秒时间戳
        "type": 0 //识别出的人员类型，0：时间段内，1：时间段外，2：陌生人/识别失败
      }
    ]
  }
}
```

```

    {
      "id": 2,
      "model": 0,//识别模式, 0: 刷脸, 1: 卡&人脸双重认证, 2: 人证比对, 3: 刷卡
      "path": "ftp://192.168.50.32:8010/recordsImg/2018-11-14/Abc123_1542189365804.jpg",
      "personId": "Abc123",
      "state": 1,
      "time": 1542189365804,
      "type": 0//识别出的人员类型, 0: passTime 时间段内, 1: passTime 时间段外, 2: 陌生人/识别失败
    }, {
      "id": 3,
      "idcard": "{\"address\":\"浙江省杭州市西湖区文一路\",\"birthday\":\"1993-01-01\",\"compareResult\":true,\"createTime\":1542247383226,\"deviceKey\":\"84E0F4200E4802FAI\",\"id\":\"0\",\"idNum\":\"330326199301010818\",\"issuingOrgan\":\"西湖区公安局\",\"name\":\"郝雨凡\",\"nation\":\"汉\",\"photoPath\":\"ftp://192.168.50.32:8010/recordsImg/2018-11-15/IDCARD_1542247383218_idcard.jpg\",\"sex\":\"女\",\"usefulLife\":\"2011-2021\"}",//人证比对详情
      "model": 2,//识别模式, 0: 刷脸, 1: 卡&人脸双重认证, 2: 人证比对, 3: 刷卡
      "path": "ftp://192.168.50.32:8010/recordsImg/2018-11-15/IDCARD_1542247383218.jpg",
      "personId": "IDCARD",//人证比对结果 id 为 IDCARD
      "state": 1,
      "time": 1542247383218,
      "type": 0//识别出的人员类型, 0: passTime 时间段内, 1: passTime 时间段外, 2: 陌生人/识别失败
    }, {
      "id": 4,
      "path": "ftp://192.168.24.100:8010/recordsImg/2018-11-14/STRANGERBABY_1542247436772.jpg",//陌生人/识别失败现场抓拍照片也会存储在设备内
      "personId": "STRANGERBABY",//陌生人/识别失败显示 id 为 STRANGERBABY
      "state": 1,
      "time": 1542247436772,
      "type": 2//识别出的人员类型, 0: passTime 时间段内, 1: passTime 时间段外, 2: 陌生人/识别失败
    }
  ],
  "result": 1,//接口调通
  "success": true//识别记录查询成功并返回
}

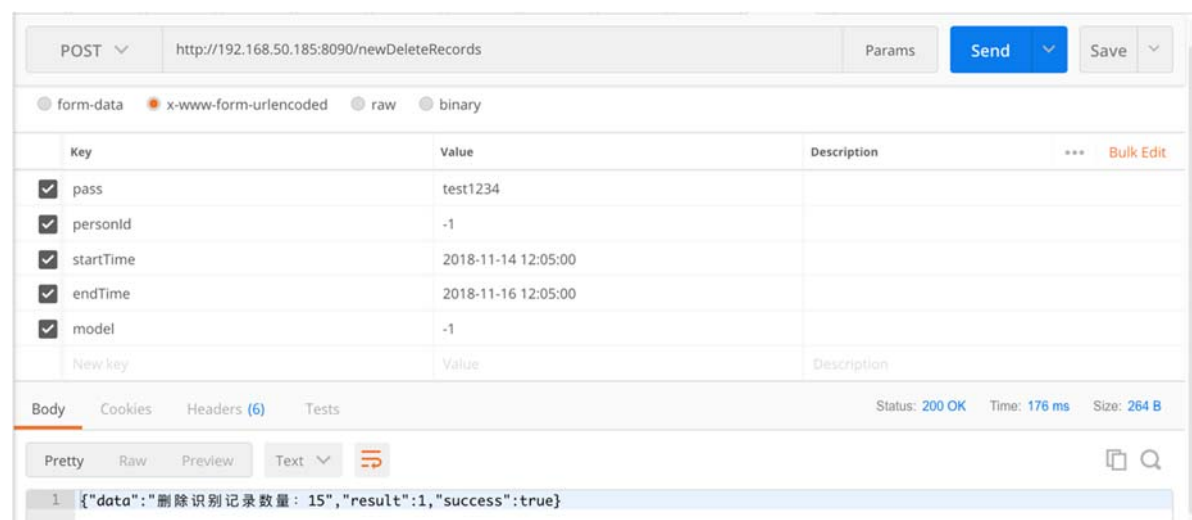
```

## 5.2 识别记录删除

### 请求数据

Method	URL			
POST	http://设备 IP:8090/newDeleteRecords			
参数名	描述	类型	必传	附加说明
pass	设备密码	String	Y	
personId	人员 id	String	Y	<ul style="list-style-type: none"><li>删除指定 id 的人员识别记录</li><li>传入 -1 可删除所有人员的识别记录，包括陌生人</li><li>传入 STRANGERBABY，可删除所有陌生人 / 识别失败记录</li><li>传入 IDCARD，可删除所有人证比对的识别记录</li></ul>
startTime	记录起始时间	Date	Y	<ul style="list-style-type: none"><li>删除时间段内所有人脸识别记录及现场照</li><li>startTime 传入 0，默认起始时间为最早的一条记录</li><li>传入时间格式为（年-月-日 时:分:秒）： 2017-07-15 12:05:00</li></ul> 精确删除请用 unixtime
endTime	记录结束时间	Date	Y	
model	记录类型	Int	Y	<ul style="list-style-type: none"><li>-1：所有类型的识别记录</li><li>0：刷脸识别记录</li><li>1：卡&amp;人脸双重认证识别记录</li><li>2：人证比对识别记录</li><li>3：刷卡识别记录其他为非法值</li></ul>

### postman 示例





## 返回示例

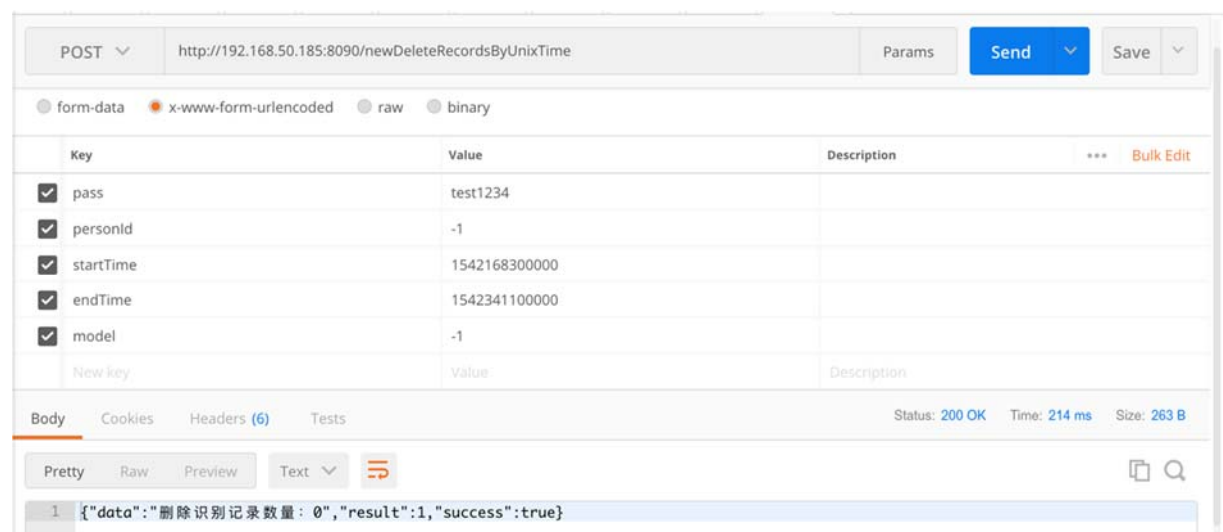
```
{  
  "data": "删除识别记录数量: 1",//删除的识别记录数量  
  "result": 1,//接口调通  
  "success": true//成功删除识别记录  
}
```

## 5.3 识别记录删除（通过 Unix 毫秒级时间戳删除）

### 请求数据

Method	URL			
POST	http://设备 IP:8090/newDeleteRecordsByUnixTime			
参数名	描述	类型	必传	附加说明
pass	设备密码	String	Y	
personId	人员 id	String	Y	<ul style="list-style-type: none"><li>删除指定 id 的人员识别记录</li><li>传入 -1 可删除所有人员的识别记录，包括陌生人</li><li>传入 STRANGERBABY，可删除所有陌生人/识别失败记录</li><li>传入 IDCARD，可删除所有人证比对的识别记录</li></ul>
startTime	记录起始时间	Date	Y	<ul style="list-style-type: none"><li>删除时间段内所有人脸识别记录及现场照</li><li>startTime 传入 0，默认起始时间为最早的一条记录</li><li>传入时间格式为：1507773661000</li></ul>
endTime	记录结束时间	Date	Y	
model	记录类型	Int	Y	<ul style="list-style-type: none"><li>-1：所有类型的识别记录</li><li>0：刷脸识别记录</li><li>1：卡&amp;人脸双重认证识别记录</li><li>2：人证比对识别记录</li><li>3：刷卡识别记录其他为非法值</li></ul>

### postman 示例



## 返回示例


```
{  
  "data": "删除识别记录数量: 0",//删除的识别记录数量  
  "result": 1,//接口调通  
  "success": true//成功删除识别记录  
}
```

## 5.4 刷脸记录查询

Method	URL			
GET	http://设备 IP:8090/findRecords			
参数名	描述	类型	必传	附加说明
pass	设备密码	String	Y	
personId	人员 id	String	Y	<ul style="list-style-type: none"><li>• 查询指定 id 的人员识别记录</li><li>• 传入-1 可查询所有人员的识别记录，包括陌生人</li><li>• 传入 STRANGERBABY，可查询所有陌生人/识别失败记录</li></ul>
length	每页最大数量	Int	Y	<ul style="list-style-type: none"><li>• 传入-1 为不分页</li><li>• 若不传-1，请务必大于 0</li></ul>
index	页码	Int	Y	页码，从 0 开始
startTime	记录开始时间	Date	Y	<ul style="list-style-type: none"><li>• 若不按时间查询，请分别传入 0</li><li>• 若需要按时间查询，请按照如下格式（年-月-日 时:分:秒）： 2017-07-15 12:05:00</li></ul>
endTime	记录结束时间	Date	Y	

### 请求数据

### postman 示例

 注意：GET 请求参数放在 url 里，点击“Params”添加参数。

GET

http://192.168.24.100:8090/findRecords?pass=123456&personId=1&length=-1&index=0&startTim...

Params

Send

Save

Key	Value	Description	*** Bulk Edit
<input checked="" type="checkbox"/> pass	123456		
<input checked="" type="checkbox"/> personId	-1		
<input checked="" type="checkbox"/> length	-1		
<input checked="" type="checkbox"/> index	0		
<input checked="" type="checkbox"/> startTime	0		
<input checked="" type="checkbox"/> endTime	0		
New key	Value	Description	

Authorization

Headers (1)

Body

Pre-request Script

Tests

Cookies

Code

Type

No Auth

Body

Cookies

Headers (6)

Tests

Status: 200 OK

Time: 155 ms

Size: 929 B

Pretty

Raw

Preview

Text

1

```
{
  "data": {
    "pageInfo": {
      "index": 0,
      "length": -1,
      "size": 4,
      "total": 4
    },
    "records": [
      {
        "id": 471,
        "path": "ftp://192.168.24.100:8010/recordsImg/2018-06-20/12345abc_1529485386691.jpg",
        "personId": "12345abc",
        "state": 1,
        "time": 1529485386691,
        "type": 0
      },
      {
        "id": 472,
        "path": "ftp://192.168.24.100:8010/recordsImg/2018-06-20/12345abc_1529485552318.jpg",
        "personId": "12345abc",
        "state": 1,
        "time": 1529485552318,
        "type": 0
      },
      {
        "id": 473,
        "path": "ftp://192.168.24.100:8010/recordsImg/2018-06-20/12345abc_1529487376167.jpg",
        "personId": "12345abc",
        "state": 1,
        "time": 1529487376167,
        "type": 0
      },
      {
        "id": 474,
        "path": "ftp://192.168.24.100:8010/recordsImg/2018-06-20/12345abc_1529492413727.jpg",
        "personId": "12345abc",
        "state": 1,
        "time": 1529492413727,
        "type": 0
      }
    ]
  },
  "result": 1,
  "success": true
}
```

## 返回示例

```
{
  "data": {
    "pageInfo": {
      "index": 0, //页码。当前数据所在页码，从 0 开始计数
      "length": -1, //每页最大数量。-1 表示不分页，所有数据均显示在第一页(index=0)
      "size": 0, //总页数。若 length=-1，size 默认为 0；若 length > 0，size ≥ 1
      "total": 3 //总数据条数
    },
    "records": [{
      "id": 471, //识别记录 id，由设备自动生成并管理。用户无须使用
      "path": "ftp://192.168.24.100:8010/recordsImg/2018-06-20/12345abc_1529485386691.jpg", //现场抓拍照片存储在设备内的路径。当设备内现场照存储满 3G 时，会自动删除较早的 1G 现场照；若有需要请及时存储
      "personId": "12345abc", //人员 id。陌生人/识别失败显示 id 为 STRANGERBABY
      "state": 1, //回调结果。0：回调失败，1：回调成功或未设置回调地址
      "time": 1529485386691, //识别成功时的设备时间，Unix 毫秒时间戳
      "type": 0 //识别出的人员类型，0：时间段内，1：时间段外，2：陌生人/识别失败
    }, {
      "id": 472,
      "path": "ftp://192.168.24.100:8010/recordsImg/2018-06-20/12345abc_1529485552318.jpg",
      "personId": "12345abc",
      "state": 1,
      "time": 1529485552318,
      "type": 1 //识别出的人员类型，0：passTime 时间段内，1：passTime 时间段外，2：陌生人/识别失败
    }, {
      "id": 473,
      "path": "ftp://192.168.24.100:8010/recordsImg/2018-06-20/STRANGERBABY_1529492413727.jpg", //陌生人/识别失败现场抓拍照片也会存储在设备内
      "personId": "STRANGERBABY", //陌生人/识别失败显示 id 为 STRANGERBABY
      "state": 1,
      "time": 1529492413727,
      "type": 2 //识别出的人员类型，0：passTime 时间段内，1：passTime 时间段外，2：陌生人/识别失败
    }
  ],
  "result": 1, //接口调通
  "success": true //识别记录查询成功并返回
}
```

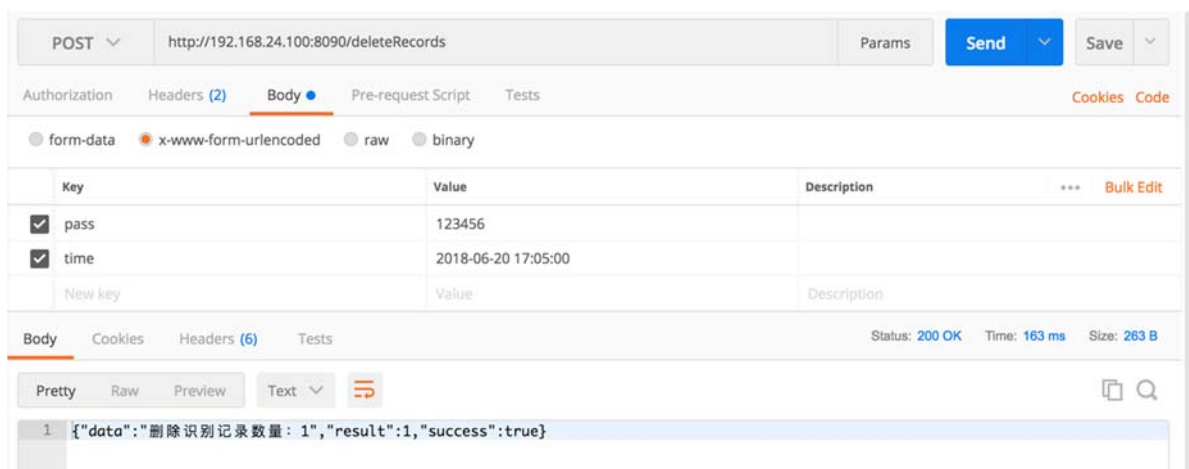


## 5.5 刷脸记录删除

Method	URL			
POST	http://设备 IP:8090/deleteRecords			
参数名	描述	类型	必传	附加说明
pass	设备密码	String	Y	
time	时间节点	String	Y	<ul style="list-style-type: none"><li>删除所传日期时间点前的所有人脸识别记录及现场照</li><li>传入时间格式为（年-月-日 时:分:秒）： 2017-07-15 12:05:00</li></ul>

### 请求数据

### postman 示例



### 返回示例

```
{
  "data": "删除识别记录数量: 1", //删除的识别记录数量
  "result": 1, //接口调通
  "success": true //成功删除刷脸识别记录
}
```

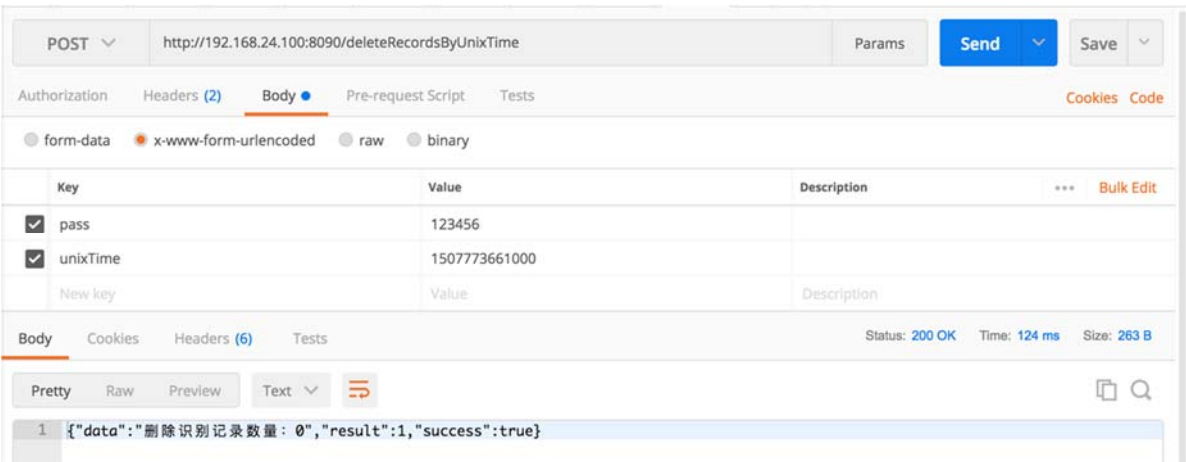


## 5.6 刷脸记录删除（通过 Unix 毫秒级时间戳删除）

### 请求数据

Method	URL			
POST	http://设备 IP:8090/deleteRecordsByUnixTime			
参数名	描述	类型	必传	附加说明
pass	设备密码	String	Y	
unixTime	Unix 毫秒级时间戳	String	Y	<ul style="list-style-type: none"><li>删除所传参数时间戳之前的所有人脸识别记录及现场照</li><li>传入时间格式为：1507773661000</li></ul>

### postman 示例



### 返回示例


```
{
  "data": "删除识别记录数量: 0", // 删除的识别记录数量
  "result": 1, // 接口调通
  "success": true // 成功删除刷脸识别记录
}
```

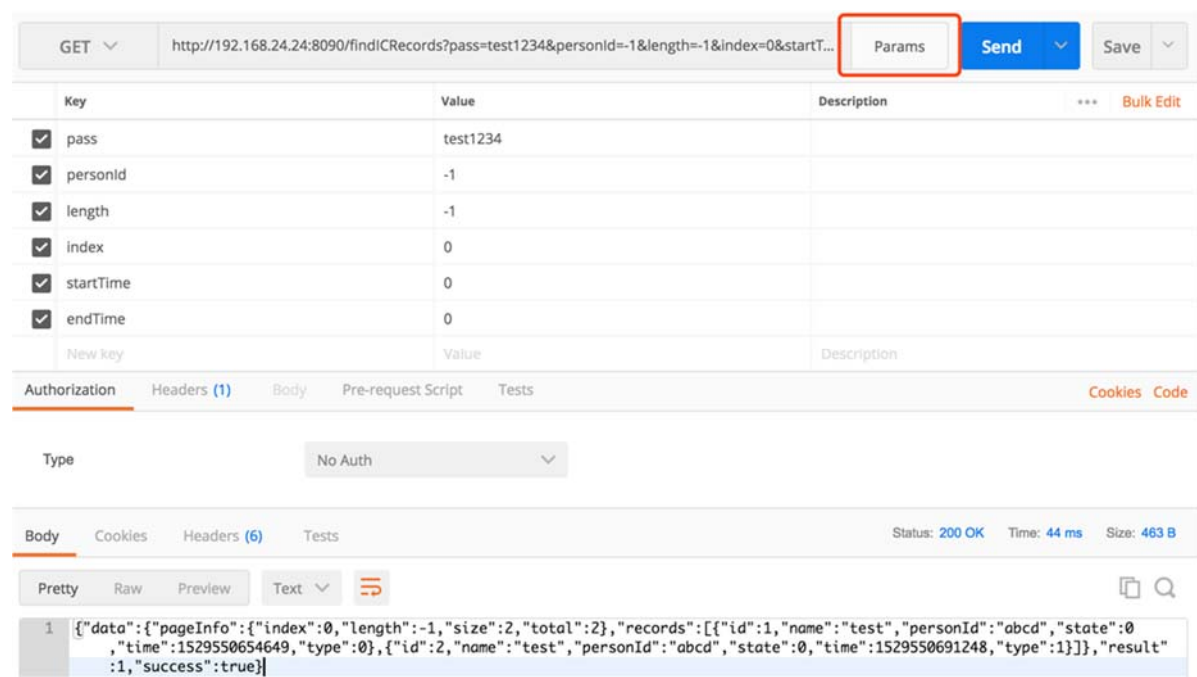
## 5.7 刷卡记录查询

### 请求数据

Method	URL			
GET	http://设备 IP:8090/findICRecords			
参数名	描述	类型	必传	附加说明
pass	设备密码	String	Y	
personId	人员 id	String	Y	<ul style="list-style-type: none"><li>查询指定 id 的人员刷卡记录</li><li>传入-1 可查询所有人员的刷卡记录</li><li>识别失败的卡片不会生成刷卡记录</li></ul>
length	每页最大数量	Int	Y	<ul style="list-style-type: none"><li>传入-1 为不分页</li><li>若不传-1, 请务必大于 0</li></ul>
index	页码	String	Y	页码, 从 0 开始
startTime	记录开始时间	Date	Y	<ul style="list-style-type: none"><li>若不按时间查询, 请分别传入 0</li><li>若需要按时间查询, 请按照如下格式 (年-月-日 时:分:秒): 2017-07-15 12:05:00</li></ul>
endTime	记录结束时间	Date	Y	

### postman 示例

 注意: GET 请求参数放在 url 里, 点击 “Params” 添加参数。



GET `http://192.168.24.24:8090/findICRecords?pass=test1234&personId=-1&length=-1&index=0&startTime=0&endTime=0` Params Send Save

Key	Value	Description
pass	test1234	
personId	-1	
length	-1	
index	0	
startTime	0	
endTime	0	

Authorization Headers (1) Body Pre-request Script Tests Cookies Code

Type No Auth

Body Cookies Headers (6) Tests Status: 200 OK Time: 44 ms Size: 463 B

Pretty Raw Preview Text

```
1 [{"data":{"pageInfo":{"index":0,"length":-1,"size":2,"total":2},"records":[{"id":1,"name":"test","personId":"abcd","state":0,"time":1529550654649,"type":0}, {"id":2,"name":"test","personId":"abcd","state":0,"time":1529550691248,"type":1}], "result":1,"success":true}]
```

## 返回示例

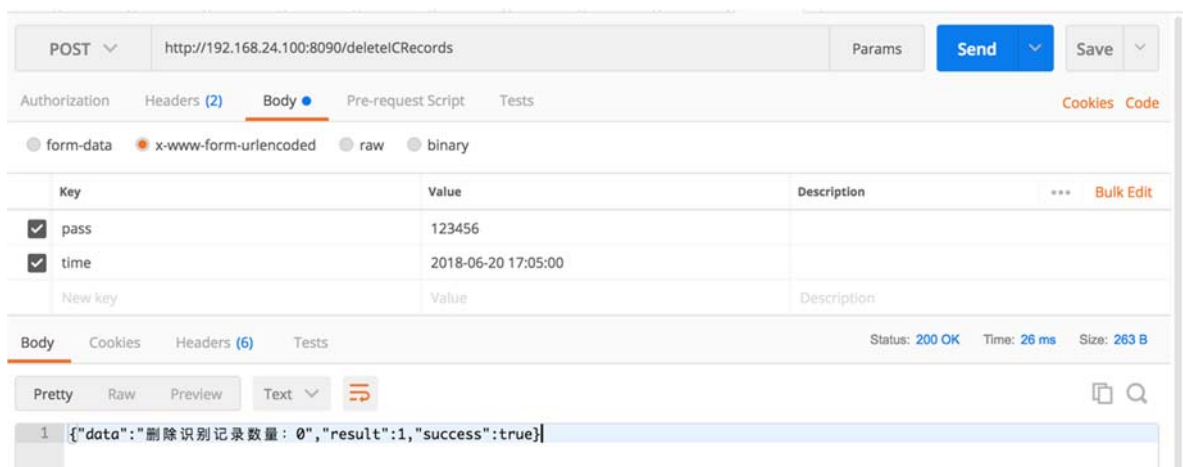
```
{
  "data": {
    "pageInfo": {
      "index": 0, //页码。当前数据所在页码
      "length": -1, //每页最大数量。-1 表示不分页，所有数据均显示在第一页(index=0)
      "size": 0, //总页数。若 length=-1，size 默认为 0；若 length > 0，size ≥ 1
      "total": 2 //总数据条数
    },
    "records": [{
      "id": 1, //识别记录 id，由设备自动生成并管理。用户无须使用
      "name": "test", //该卡号对应的人员姓名
      "personId": "abcd", //该卡号对应的人员 id
      "state": 1, //回调结果。0：回调失败，1：回调成功或未设置回调地址
      "time": 1529550654649, //识别成功时的设备时间，Unix 毫秒时间戳
      "type": 0 //识别出的人员类型，0：passTime 时间段内，1：passTime 时间段外
    }, {
      "id": 2,
      "name": "test",
      "personId": "abcd",
      "state": 0,
      "time": 1529550691248,
      "type": 1 //识别出的人员类型，0：passTime 时间段内，1：passTime 时间段外
    }]
  },
  "result": 1, //接口调通
  "success": true //刷卡记录查询成功并返回
}
```

## 5.8 刷卡记录删除

### 请求数据

Method	URL			
POST	http://设备 IP:8090/deleteICRecords			
参数名	描述	类型	必传	附加说明
pass	设备密码	String	Y	
time	时间节点	String	Y	<ul style="list-style-type: none"><li>删除所传日期时间点前的所有刷卡记录</li><li>传入时间格式为（年-月-日 时:分:秒）： 2017-07-15 12:05:00</li></ul>

### postman 示例



### 返回示例

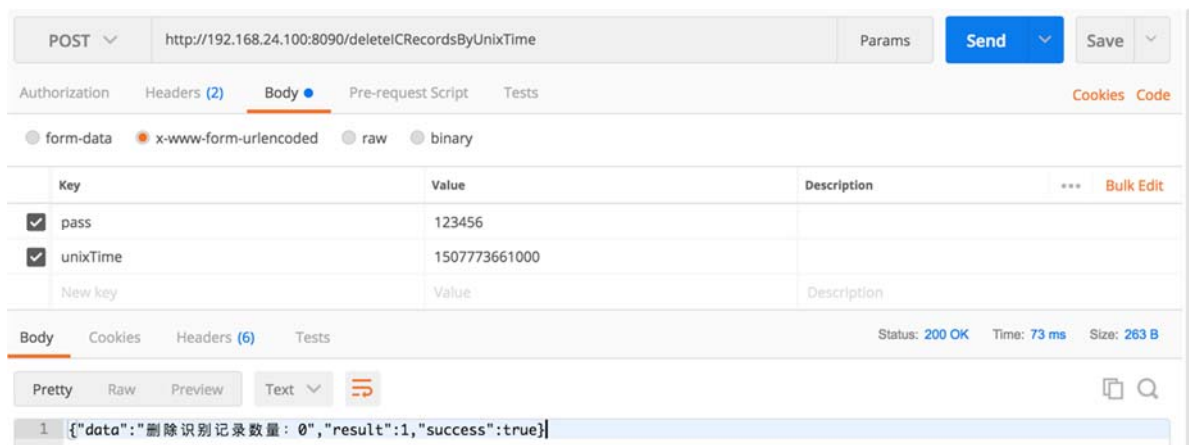
```
{
  "data": "删除识别记录数量: 0", //删除的识别记录数量
  "result": 1, //接口调通
  "success": true //成功删除刷卡识别记录
}
```

## 5.9 刷卡记录删除（通过 Unix 毫秒级时间戳删除）

### 请求数据

Method	URL			
POST	http://设备 IP:8090/deleteICRecordsByUnixTime			
参数名	描述	类型	必传	附加说明
pass	设备密码	String	Y	
unixTime	Unix 毫秒级时间戳	String	Y	<ul style="list-style-type: none"><li>删除所传参数时间戳之前的所有人脸识别记录及现场照</li><li>传入时间格式为：1507773661000</li></ul>

### postman 示例



### 返回示例

```
{
  "data": "删除识别记录数量: 0", //删除的识别记录数量
  "result": 1, //接口调通
  "success": true //成功删除刷卡识别记录
}
```