



宇泛智能

局域网平台服务

局域网（离线版）系统接口手册

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

使用须知

本节内容的目的是确保用户通过本手册能够正确使用产品，以避免操作中的危险或财产损失。在使用此产品之前，请认真阅读产品手册并妥善保存以备日后参考。

符号约定

对于文档中出现的符号，说明如下所示。

符号	说明
 注意	<ul style="list-style-type: none"> 解释说明：方便理解、特殊情况说明 注意事项：表示会导致问题，若不避免，可能造成设备损坏、操作中断
参见	超链接，点击跳转文档内部内容或外部网站

适用型号

本手册适用于以下型号的人脸识别终端。

产品名称	产品型号
人脸识别设备	Uface-M5201 系列：Uface-M5201、Uface-M5201-IR、
	Uface-M7202 系列：Uface-M7202、Uface-M7202(SR)、Uface-M7202-IC、Uface-M7202(SR)-IC、
人脸识别闸机立柱	Uface-M5201-GH、Uface-M7212
支架立柱	Uface-A103(ZJ)

型号后缀中，IR（Infrared Ray，红外线）表示产品摄像头为红外类型；SR（Silver）表示产品颜色为银色；IC 表示设备自带 IC 卡读卡功能；ZJ（支架拼音缩写）表示产品为支架配件。

版权所有©杭州宇泛智能科技有限公司 2018。保留一切权利。

本手册的任何部分，包括文字、图片、图形等均归属于杭州宇泛智能科技有限公司（以下简称“本公司”或“宇泛智能”）。未经书面许可，任何单位和个人不得以任何方式摘录、复制、翻译、修改本手册的全部或部分。除非另有约定，本公司不对本手册提供任何明示或默示的声明或保证。

关于本手册

本手册作为指导使用。手册中所提供照片、图形、图表和插图等，仅用于解释和说明目的，与具体产品可能存在差异，请以实物为准。

因产品版本升级或其他需要，本公司可能对本手册进行更新，如您需要最新版手册，请您登录公司官网查阅(www.uni-ubi.com)。

为方便阅读，建议使用阅读器时打开左侧目录/导航/文档结构图。

宇泛智能建议您在专业人员的指导下使用本手册。

商标声明



宇泛智能

为宇泛智能的注册商标。本手册涉及的其他商标由其所有人各自拥有。

责任声明

- 1、在法律允许的最大范围内，本手册所描述的产品(含其硬件、软件、固件等)均“按照现状”提供，可能存在瑕疵、错误或故障，本公司不提供任何形式的明示或默示保证，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；亦不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。
- 2、若您将产品接入互联网需自担风险，包括但不限于产品可能遭受网络攻击、黑客攻击、病毒感染等，本公司不对因此造成的产品工作异常、信息泄露等问题承担责任，但本公司将及时为您提供产品相关技术支持。
- 3、使用本产品时，请您严格遵循适用的法律。若本产品被用于侵犯第三方权利或其他不当用途，本公司概不承担任何责任。
- 4、如本手册内容与适用的法律相冲突，则以法律规定为准。

目录

使用须知	i
目录	i
更新说明	IV
第一章 接口操作说明	1
1.1 接口说明	1
1.1.1 接口规范.....	1
1.1.2 接口返回.....	1
1.1.3 接口调用流程.....	3
1.1.4 注意事项.....	4
第二章 接口操作	5
2.2 设备管理类接口	5
2.2.1 设置设备密码.....	5
2.2.2 设备序列号获取.....	7
2.2.3 设备配置.....	8
2.2.4 识别模式配置.....	18
2.2.5 陌生人串口 & 韦根输出.....	24
2.2.6 修改 Logo.....	26
2.2.7 修改屏幕方向.....	27
2.2.8 设置设备时间.....	28
2.2.9 有线网络配置.....	30
2.2.10 Wi-Fi 配置.....	32
2.2.11 设备重启.....	34
2.2.12 设备重置.....	35
2.2.13 设备开门.....	37
2.2.14 大屏模式接口.....	40
2.2.15 大屏背景图片配置.....	42
2.2.16 隐藏 IP、设备序列号、人数和照片数.....	44
2.2.17 识别回调.....	45
2.2.18 注册照片回调.....	49
2.2.19 设备心跳回调.....	51
2.2.20 数据写入接口.....	53
2.2.21 数据读出接口.....	54

2.3 人员管理类接口	55
2.3.1 人员注册	55
2.3.2 人员卡号注册	58
2.3.3 人员更新	60
2.3.4 人员删除（批量）	62
2.3.5 人员分页查询	64
2.3.6 人员信息查询	69
2.3.7 时间段权限删除	73
2.3.8 人员有效期限设置	74
2.3.9 有效期限设置删除	76
2.4 照片管理类接口	77
2.4.1 照片注册（base64）	77
2.4.2 照片注册（url）	79
2.4.3 照片注册（File）	81
2.4.4 拍照注册	81
2.4.4 照片查询	84
2.4.5 特征注册	86
2.4.8 清空人员注册照片	92
2.4.6 照片更新	94
2.4.7 照片删除	96
2.4.9 人脸照片相似性比对	97
2.5 识别记录管理接口	99
2.5.1 识别记录查询（推荐使用）	99
2.5.2 识别记录删除（推荐使用）	102
2.5.3 识别记录删除（通过 Unix 毫秒级时间戳删除）	104
2.5.4 刷脸记录查询	106
2.5.5 刷脸记录删除	111
2.5.6 刷脸记录删除（通过 Unix 毫秒级时间戳删除）	112
2.5.7 刷卡记录查询	113
2.5.8 刷卡记录删除	115
2.5.9 刷卡记录删除（通过 Unix 毫秒级时间戳删除）	116
第三章 注意事项	117
附录 1 注册照片规范说明	119
附录 2 安装环境要求	123
2.1 面部光照强度要求	123
2.2 设备安装位置建议	126
附录 3 安装/摆放高度	128
附录 4 网络连接说明	131
4.1 网络要求	131

4.2 路由器检查	131
4.2.1 检查路由器网络状态.....	132
4.2.2 路由器黑名单/白名单设置.....	134
4.2.3 路由器是否为 DHCP（自动获取 IP）模式.....	137
4.2.4 路由器可连接的设备数量限制&设备 IP 冲突检查.....	140
4.3 有线连接及检查	142
4.4 无线连接及检查	143
4.5 网络常见问题	151
附录 5 设备本地管理页面	156
附录 6 回调示例详情	159
附录 7 msg 信息及异常码	162

更新说明

设备版本号	发布日期	说明
V 3.2	2017.12.20	1.竖屏模式支持开启设备拍照功能 2.修复删除注册照片后，FTP 文件还能访问的问题 3.新增清空人员注册照片接口，可清空某人员 id 的所有注册照片
V 3.3	2017.12.29	1.新增时间段权限设置接口，人员到期设备会定时删除该人员 2.新增时间段权限删除接口，清除人员的权限设置，人员不会被定时删除 3.设备配置接口新增字段 multiplayerDetection : 多个人脸检测设置，默认 1: 检测多个人脸并进行识别，2: 只检测多个人脸中最大的人脸，并进行识别 4.新增设备开门接口，密码验证正确，设备就会控制开启门禁（设备硬件上需接门禁小板）
V 3.4	2018.01.16	设备日志优化
V 3.5	2018.01.23	1.新增大屏显示模式接口，可选择传统模式或自定义背景模式 2.新增大屏背景图片配置接口，可上传自定义图片作为大屏背景（需将大屏显示模式接口设置为 2: 自定义背景模式，图片设置才会生效） 3.优化读卡方式 4.识别回调新增参数 path ，为现场照保存在设备内的路径，访问此 url 需设备在线，且发送请求的客户端与设备处于同一网段 5.修复了 passTime 不能设置成 8 点前的问题 6.接口文档新增附录 msg 信息及异常码
V 3.6	2018.03.15	<p>注：若从低版本升级到 V3.6 版本，设备会恢复出厂设置，升级完成后需重新设置设备密码 (pass)、设备配置、设备屏幕方向及回调配置；人员、照片、识别记录升级后仍然保留在设备内</p> 1.设备配置接口新增字段 recRank : 识别等级，1: 识别速度最快，精确率最低；默认 2: 识别速度较快，精确率较高；3: 识别速度较慢，精确度最高。一代设备暂不支持活体检测，二代设备升到此版本后支持活体检测 2.设备配置接口新增字段 wg ，二代设备可通过串口配置韦根信号输出（设备需外接信号转换小板），也可通过韦根口配置韦根信号输出，传参格式为 #WG{idcardNum}# 或 #WG{id}# 3.新增 Wi-Fi 配置接口，配有 Wi-Fi 模块的设备可通过该接

		<p>口连接 Wi-Fi</p> <p>4.新增指定日期前刷卡记录删除接口</p> <p>5.新增刷卡记录删除接口（通过毫秒级时间戳进行删除）</p> <p>6.回调、识别记录新增陌生人类型及其现场照，陌生人 personId 使用固定字段 STRANGERBABY，识别出的人员类型有 0：时间段内，1：时间段外，2：陌生人/识别失败</p> <p>7.识别回调数据从 POST 请求的 url 和 body 都可接收到</p> <p>8.一代设备、二代设备（仅支持竖屏）界面显示优化</p> <p>9.设备序列号获取接口(3.3.3)不需要传参数 pass，请求方式由 GET 改为 POST</p> <p>10.设备心跳回调数据新增字段 version，表示设备版本号</p> <p>11.修复了二代设备现场照倒置的问题</p> <p>12.修改了特征注册接口(3.2.7)参数 featureKey 的加密方式，从低版本设备上获取的 feature 和 featureKey 在 V3.6 版本会注册失败</p> <p>13.附录新增回调示例详情</p>
V 3.6025-V3.6 027	2018.07.31	<p>1.新增人员分页查询接口。</p> <p>2.Uface-M72XX 系列设备新增外接鼠标配置 Wi-Fi 功能，可进行：有线固定 IP、wifi 连接（自动获取 IP）、wifi 连接（固定 IP）。</p> <p>3.注册照回调新增字段 newImgPath，表示照片存在设备内的完整路径。</p> <p>4.设备配置接口新增 delayTimeForCloseTime 字段，可配置继电器控制开门到关门的时间。</p> <p>5.新增 web，用于固件升级功能。在浏览器中输入 http://设备 IP:8090，并访问。访问成功后，可选择电脑本地固件包给设备进行升级。</p> <p>6.新增韦根 34 功能，详情见设备配置接口。</p> <p>7.设备升级界面优化，显示下载速度、剩余时间。</p> <p>8.设备心跳回调 body 中增加心跳数据，格式为 Json。</p> <p>9.修复现场照断裂、识别回调慢、循环加人崩溃、屏保异常问题。</p> <p>10.优化人脸框显示、权限不足提示。</p>
V 3.6206-V3.6 208	2018.10.10	<p>注：由于设备算法更新，若从低版本升到 V3.6206 及以上版本，需清空设备数据库中的人脸数据才可继续使用（建议重置设备，清空数据库）</p> <p>1.照片注册接口、照片更新接口新增参数 isEasyWay, true: 宽松注册，兼容以前的注册照片；默认 false: 严格注册，对注册照片使用新算法的规范要求</p> <p>2.设备配置接口修改识别距离说明。Uface-72XX 系列设备设置为等级 3 时，识别距离最远为 1.5 米，可设置的有效识别距离为 1-3 档，1：0.5 米以内，默认 2：1 米以内，3：</p>

		<p>1.5 米以内。</p> <p>3.注册照片的特征码 feature 使用新算法，不兼容低版本的 feature</p> <p>4.新算法下的注册照片特征码 feature 较长，若使用注册照片回调接口接收 feature，需将 HTTP 头部长度限制改为 102400</p> <p>5.使用新算法，注册照片及其特征占用空间较大，设备允许存储 8000 张 130KB 的注册照片（或 1 万张 100KB 的注册照片）</p>
V3.6211	2018.11.10	<p>1.修复 3.6210 读出的身份证物理卡号与之前版本不一致问题</p> <p>2.修复删除所有人员时，删除 Logo 问题</p> <p>3.Uface-M5201 系列设备新增功能：支持通过三合一小板读卡，支持新旧刷卡模块；支持通过三合一小板串口输出</p> <p>4. Uface-M5201 系列设备支持连接 OTG 线连接摄像头。</p> <p>5.优化算法</p>
V3.6303	2018.11.16	<p>1.新增识别模式配置接口。设备默认模式为刷脸/刷卡。支持刷脸模式开/关、刷卡模式开/关、卡&人脸双重认证模式开/关、人证比对模式开/关</p> <p>2.新增照片注册接口（url）。接口调用成功后，设备会访问 url 并下载图片，若图片检测合格，该图即成为人员的一张注册照片</p> <p>3.新增识别记录查询接口。可以查询所有模式的记录，也可以查询单个模式的识别记录</p> <p>4.新增识别记录删除接口、识别记录删除（通过 Unix 毫秒级时间戳删除）接口。可以按时间段删除所有识别记录，也可以删除单个模式的识别记录</p>
V3.6521	20190506	<p>新增接口：</p> <ol style="list-style-type: none"> 1. 数据写入接口 2. 数据读出接口 3. 陌生人串口&韦根输出接口 4. 隐藏 IP、设备序列号、人数和照片数接口 <p>更新接口：</p> <ol style="list-style-type: none"> 1. 设备开门接口：增加串口、韦根输出功能及自定义显示和自定义播报功能 2. 人员注册接口：增加权限配置 3. 人员更新接口：增加权限配置 4. 设备配置接口：增加白名单 whitelist 字段，继电器开关 isOpenRelay 字段。

文档更新日志：

文档版本	修订内容	修改人
V3.6	1, 人员信息查询, 返回字段缺失增补 “IDNumber” “facePermisson”“idCardPermisson”“faceAndCardPermisson” “idPermisson” 2, 人员注册、人员更新接口字段示例字段类型改为 int, 实际支持 int 和 string, 同时增加各字段描述 3, 设备配置接口增加 “delayTimeForCloseDoor” 继电器延时时间, 范围 100~25500ms 4, 韦根 26 格式统一为#26WG linux 集成时需兼容#WG 格式, 老用户在用, 传#WG 是默认转换为韦根 26 5, 增加识别成功屏幕显示文字 2 的参数 "recSucDisplayText2Type","recSucDisplayText2Type"	胡雪攀 201908 12

第一章 接口操作说明

1.1 接口说明

1.1.1 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) 作为接口安全校验密钥。

1.1.2 接口返回

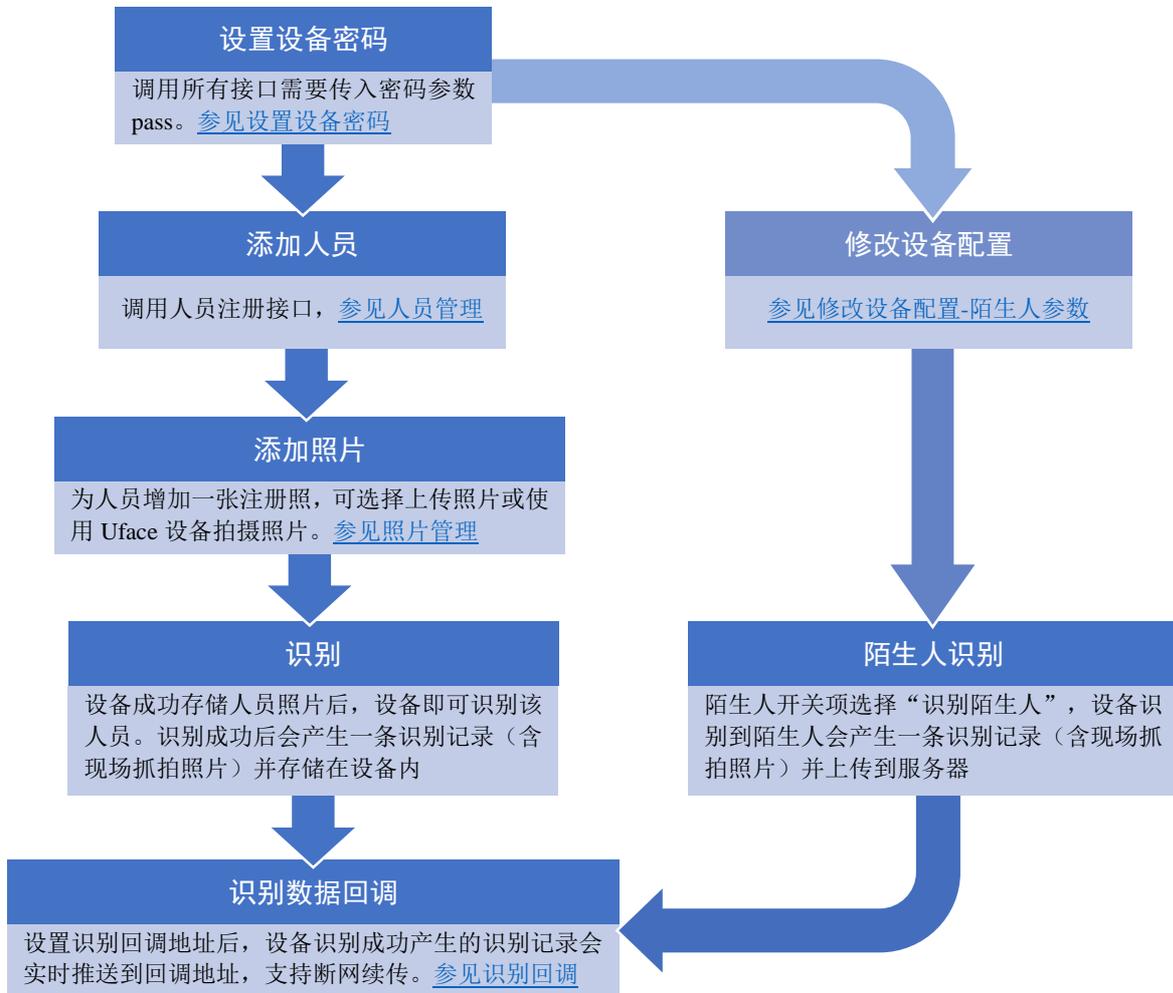
接口通用返回说明：

```
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。

1.1.3 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。

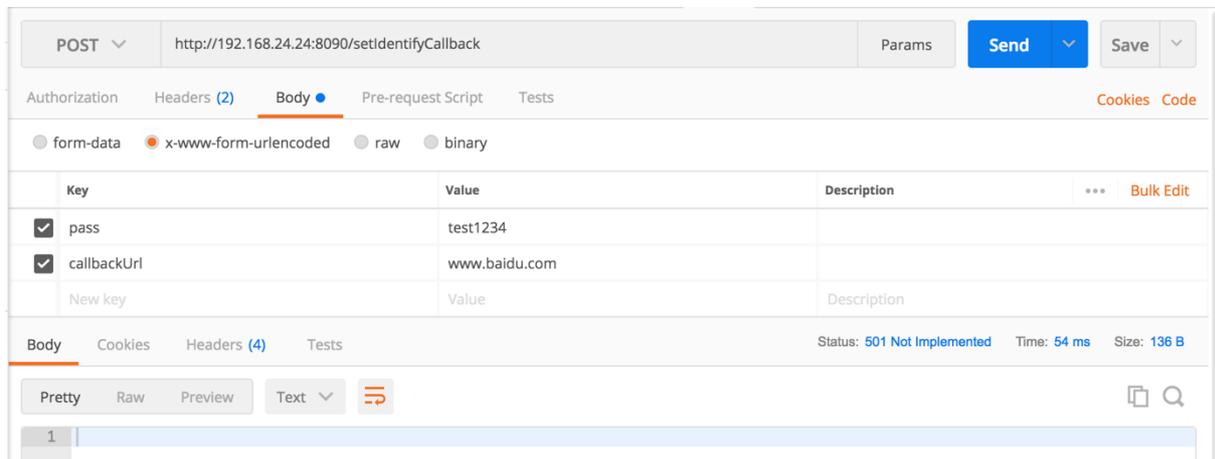


- 设备初始无密码，调用所有接口都需要传入密码参数；在设置设备密码接口，newPass 和 oldPass 传入一样的值，即成为设备密码
- 照片注册成功后，该注册照片会保存在设备内；可通过照片查询接口获得人员注册照片 url，访问 url 的照片时，设备需连网在线，且客户端与设备处于同一网段
- 人脸识别成功后，会显示注册人员的姓名；若填写了卡号，可直接刷对应卡号的卡进行识别，识别成功后会显示与该卡号对应的注册人员姓名

- 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

1.1.4 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-urlencoded`）等。

第二章 接口操作

2.2 设备管理类接口

2.2.1 设置设备密码

请求数据

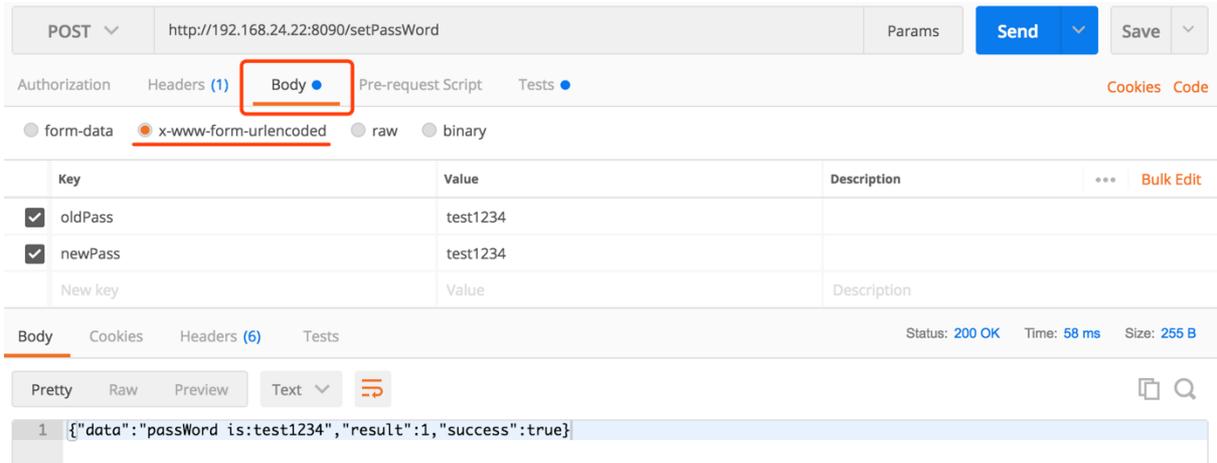
Method	URL			
POST	http://设备 IP:8090/setPassWord			
参数名	描述	类型	必传	附加说明
oldPass	旧密码	String	Y	<ul style="list-style-type: none"> • 新设备或重置后（恢复初始化）的设备，调用其他接口前，需要先进行初始密码设置，oldPass 和 newPass 传入一样的值即可 • 修改密码时，分别传入新、旧密码 • 此接口调用不需要传入 pass 密码不可为空或空格
newPass	新密码	String	Y	

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



POST http://192.168.24.22:8090/setPassWord

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

form-data x-www-form-urlencoded raw binary

Key	Value	Description
<input checked="" type="checkbox"/> oldPass	test1234	
<input checked="" type="checkbox"/> newPass	test1234	
New key	Value	Description

Body Cookies Headers (6) Tests Status: 200 OK Time: 58 ms Size: 255 B

Pretty Raw Preview Text

```
1 [{"data": "passWord is:test1234", "result": 1, "success": true}]
```

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  
}
```

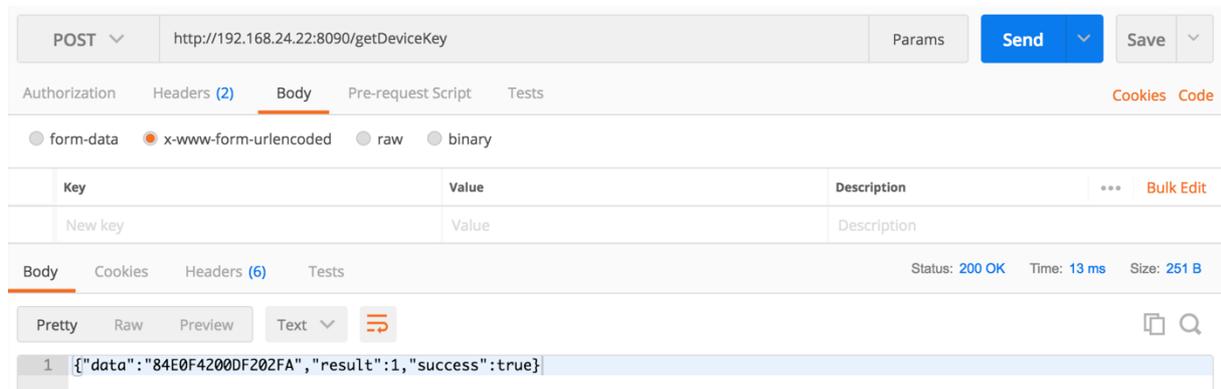
2.2.2 设备序列号获取

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



The screenshot shows a Postman interface for a POST request. The URL is `http://192.168.24.22:8090/getDeviceKey`. The request body is set to 'x-www-form-urlencoded' and contains the JSON payload: `{"data": "84E0F4200DF202FA", "result": 1, "success": true}`. The response status is 200 OK, with a time of 13 ms and a size of 251 B. The response body is displayed in 'Pretty' format.

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
}
```

2.2.3 设备配置

请求数据

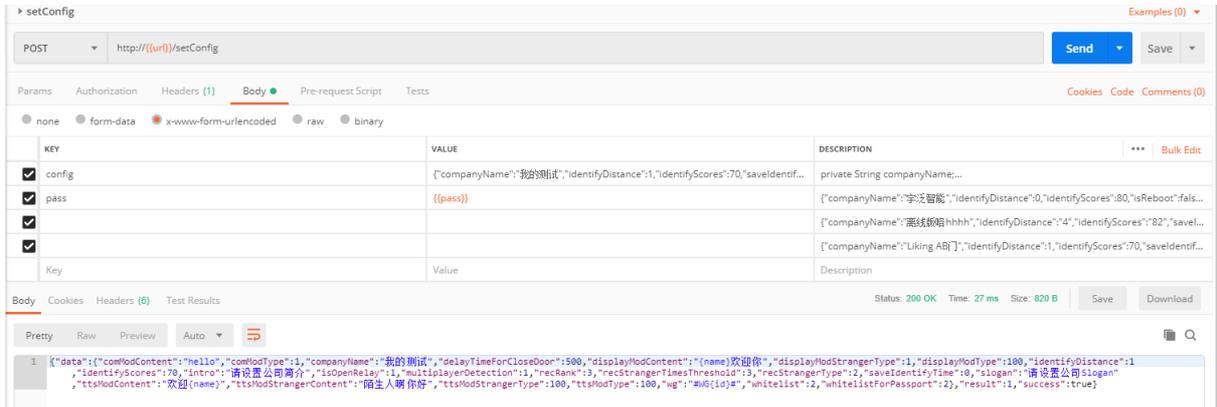
Method	URL			
POST	http://设备 IP:8090/setConfig			
参数名	描述	类型	必传	附加说明
pass	设备密码	String	Y	
config	设备配置集合	Json	Y	<p>Json 示例:</p> <pre>{ "companyName": "我的测试", "identifyDistance": 1, "identifyScores": 80, "saveIdentifyTime": 0, "ttsModType": 100, "ttsModContent": "{name}", "comModType": 100, "comModContent": "hello", "displayModType": 100, "displayModContent": "{name}欢迎你", "recSucDisplayText2Type": 100, "recSucDisplayText2Type": "识别成功", "slogan": "科技有限公司", "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": "#26WG{id}#", "recRank": 3, "whitelist": 1 }</pre> <p>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 slogan 和 intro 仅在大屏模式下展示 陌生人显示模式参数 displayModStrangerType 传入 100，则自定义参数 displayModStrangerContent 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） whitelist :添加人证比对白名单开关；类型: int； 1: 关； 2: 开；默认关闭 isOpenRelay: 2 代继电器控制开关。1: 输出继电器开关量； 2: 不输出继电器开关量；默认输出。 delayTimeForCloseDoor: 继电器延时时间；类型: int； 范围: 100~25500ms；默认 500ms</p>

postman 示例

i 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": {  
  "companyName"://设备名称，默认显示为“请设置公司名称”，可传入自定义内容  
  "identifyDistance"://识别距离，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。一代设备设置 0-6 档均有效；二  
    代设备若 recRank 设置为 1，则识别距离可设置 0-6 档，若 recRank 设置为 2 或 3，则  
    识别距离仅可设置 0-3 档  
  "identifyScores"://识别分数，识别命中分数，默认为 80，传入值务必在 50-100 之间  
  "saveIdentifyTime"://N 秒钟只保存一条识别记录配置，用于配置 N 秒内多次识别只记  
    录一次，默认为 0：每次识别记录都会计入数据库  
  "ttsModType"://语音模式类型，默认 1：不需要语音播报，2：播报名字，... 100：自定  
    义  
  "ttsModContent"://语音播报模式自定义内容，模板中只允许{name}字段，字段格式固定；  
    模板中只允许数字、英文、中文和“{”、“}”；内容长度限制 255 个字，请自行调整。  
    例：{name}欢迎光临  
  "displayModType"://屏幕显示模式类型，默认 1：显示名字，2：关闭识别弹窗；100：  
    自定义  
  "displayModContent"://显示模式自定义内容，模板中只允许{name}字段，字段格式固  
    定；模板中只允许数字、中英文、中英文符号和“{”、“}”；内容长度限制 255 个字，  
    请自行调整。例：{name}，签到成功！  
  "recSucDisplayText2Type"://识别成功参数.屏幕显示文字 2 类型 1:识别成功(默认)  
    100:自定义  
  "recSucDisplayText2Content"://识别成功参数.屏幕显示文字 2 自定义内容  
  "comModType"://串口模式类型，默认 1：开门，2：不输出，3：输出人员 id，4：输出  
    身份证/IC 卡号 idcardNum，... 100：自定义  
  "comModContent"://串口模式自定义输出内容，模板中只允许{id}、{idcardNum}字段，  
    字段格式固定；模板中只允许英文和英文符号；内容长度限制 255 个字符，请自行调  
    整。例（韦根输出）：#26WG{idcardNum}#或#26WG{id}#（其中 idcardNum、id 可配  
    置的范围必须为 1-65535，否则信号输出失效；传空，则韦根不输出；若使用串口输  
    出韦根信号，设备串口需外接定制的信号转换小板
```

```
"wg";//韦根口输出韦根信号（非串口），目前只支持输出 idcardNum 或 id，传参格式为#26WG{idcardNum}#或#26WG{id}#，传空，则韦根不输出。一代设备只能通过自定义串口输出配置韦根输出，二代设备既可通过串口配置韦根输出，也可使用此参数配置韦根口输出韦根信号（目前只支持韦根 26，idcardNum、id 可配置的范围必须为 1-65535，否则信号输出失效）
"slogan";//标语，大屏展示
"intro";//公司简介，大屏展示
"recStrangerType";//陌生人开关（是否进行陌生人识），默认 1：不识别陌生人，2：识别陌生人
"recStrangerTimesThreshold";//设备判定某人为陌生人所需时间等级（陌生人开关打开情况下设置有效），默认 3；1 表示快速判定但精确率最低，随着数值增加，判定时间增加，精确度提高生人识），1：不识别陌生人，2：识别陌生人
"ttsModStrangerType";//陌生人语音模式类型（陌生人开关打开情况下设置有效），默认 1：不需要语音播报；2：陌生人警报；.....100：自定义
"ttsModStrangerContent";//陌生人语音播报自定义内容，默认 1：模板中只允许数字、英文和中文；2：内容长度限制 255 个字。例：注意陌生人
"displayModStrangerType";//开启识别陌生人后，参数为 100 时，识别陌生人可以开启自定义弹窗显示内容
"displayModStrangerContent";//当 displayModStrangerType 参数为 100 时，陌生人自定义显示内容，不能为空。
"multiplayerDetection";//多个人脸检测设置，默认 1：检测多个人脸并进行识别，2：只检测多个人脸中最大的人脸，并进行识别
"recRank";//识别等级，1：识别速度最快，精确率最低；2：识别速度较快，精确率较高；3：识别速度较慢，精确度最高。此参数仅对二代设备有效，二代设备默认等级 2；一代设备此参数无效，默认等级 1
"whitelist";//添加人证比对白名单开关；类型：int； 1：关；2：开；白名单开关，若打开，读取身份证信息后将身份证号与数据库内的所有人员的身份证号比对，若存在则进行人证比对流程；若不存在，则提示权限不足。
    若白名单开关关闭，读取的身份证信息后直接进行人证比对流程。
"isOpenRelay";//2 代继电器控制开关。类型：int； 1：输出；2：不输出；默认为 1}
"delayTimeForCloseDoor";//继电器延时时间。类型：int；范围：100~25500ms；默认 500ms
"result": 1,//接口调通
"success": true//设备配置设置成功
```

4.2.3.1 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。

i 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，2：关闭识别弹窗，100：自定义。

i 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。

i 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": "#26WG{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": "#26WG{id}#" 或 "wg": "#26WG{idcardNum}#"

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

i 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。

4.2.3.2 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。

i 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。

i 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。

i 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。

4.2.3.3 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）都会进行识别。
- **【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。

i 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。

i 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。

i 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。

i 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。

i 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

4.2.3.4 其他参数

(1) 标语

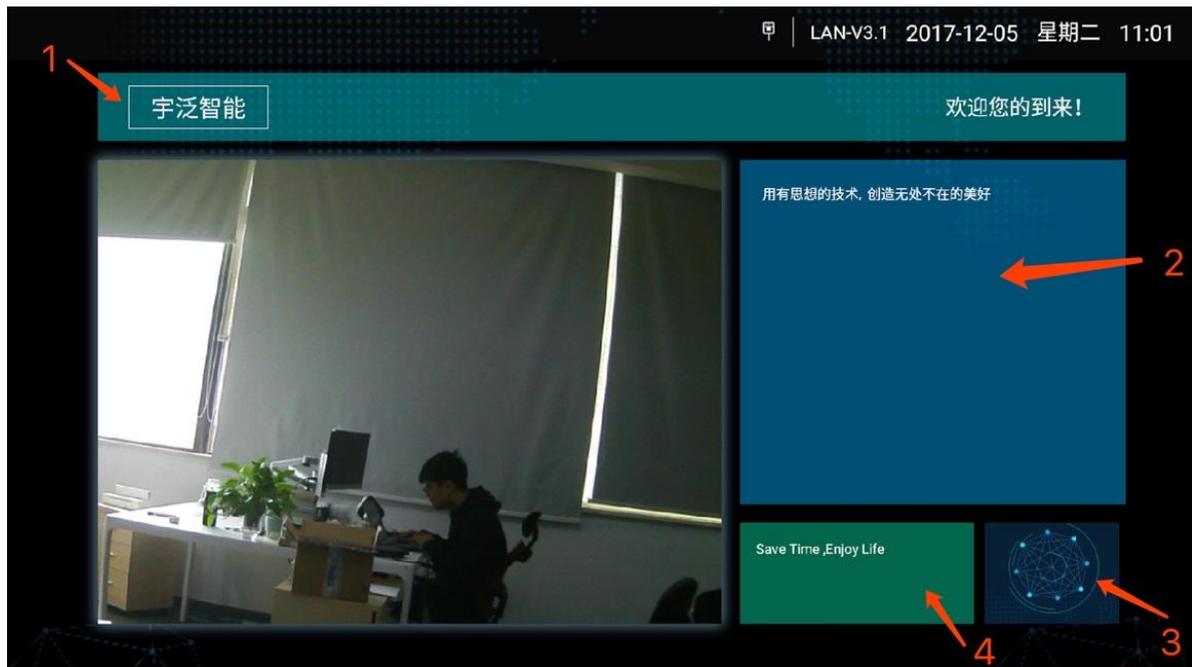
设备通过 HDMI 接口外接大屏时显示的文字内容，图见（2）所示。

(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。



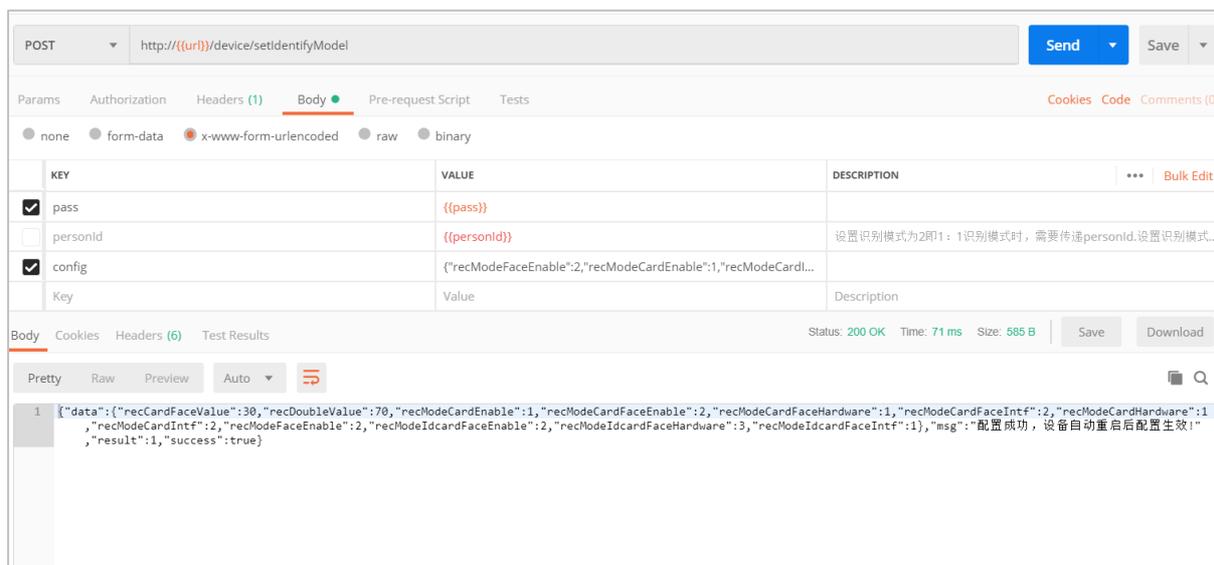
- 1.设备名称:** 可自行设备配置接口对设备名称(companyName)进行自定义
- 2.公司简介:** 可自行通过设备配置接口对公司简介(intro)进行编辑
- 3. Logo 展示区:** 默认 Logo 为动画, 可通过 Logo 修改接口对该区域展示内容进行修改, 目前仅支持静态自定义图片
- 4.slogan (标语):** 可自行通过设备配置接口对标语(slogan)进行编辑
- 5.视频预览区:** 显示摄像头实时画面

2.2.4 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"> Json 示例: <pre>{ "recModeFaceEnable":2,"recModeCardEnable":2,"recModeCardIntf":2,"recModeCardHardware":1,"recModeCardFaceEnable":1,"recModeCardFaceIntf":2,"recModeCardFaceHardware":1,"recModeIdcardFaceEnable":1,"recModeIdcardFaceIntf":1,"recModeIdcardFaceHardware":3,"recCardFaceValue":30,"recDoubleValue": 70 }</pre> 所有参数均为必传 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

postman 示例

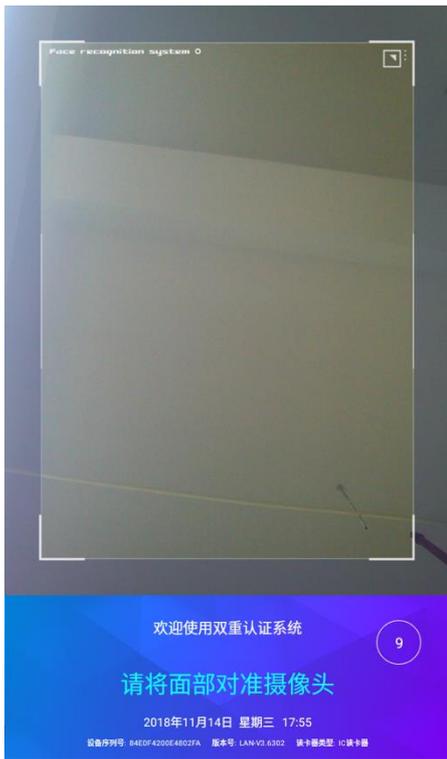


The screenshot shows a Postman request configuration for a POST method to the URL `http://{{url}}/device/setIdentifyModel`. The request body is set to `x-www-form-urlencoded` and contains three parameters: `pass` (value `{{pass}}`), `personid` (value `{{personid}}`), and `config` (value `{"recModeFaceEnable":2,"recModeCardEnable":1,"recModeCardIntf":2,"recModeCardHardware":1,"recModeIdcardFaceEnable":1,"recModeIdcardFaceIntf":1,"recModeIdcardFaceHardware":3,"recCardFaceValue":30,"recDoubleValue":70}`). The response status is `200 OK` with a time of `71 ms` and size of `585 B`. The response body is displayed in JSON format:

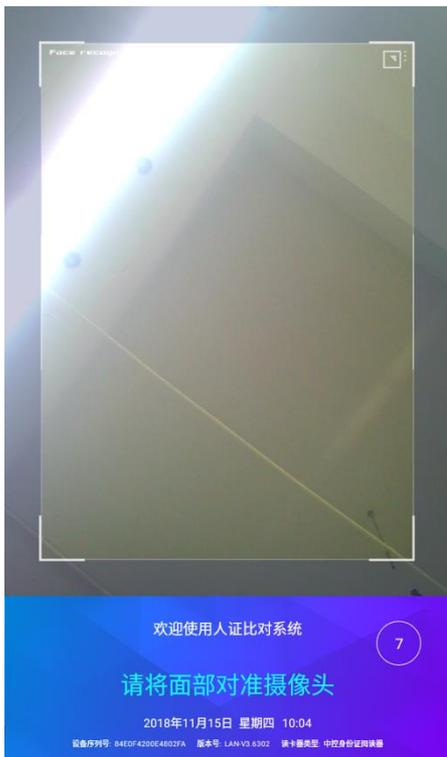
```
1 { "data": { "recCardFaceValue":30,"recDoubleValue":70,"recModeCardEnable":1,"recModeCardFaceEnable":2,"recModeCardFaceHardware":1,"recModeCardFaceIntf":2,"recModeCardHardware":1,"recModeCardIntf":2,"recModeFaceEnable":2,"recModeIdcardFaceEnable":2,"recModeIdcardFaceHardware":3,"recModeIdcardFaceIntf":1}, "msg": "配置成功, 设备自动重启后配置生效!", "result":1, "success":true }
```

界面说明

卡&人脸双重认证



人证比对



返回示例

```
{
  "data": {
    "recCardFaceValue":30,//人证比对, 识别阈值分数, 默认 30 分
    "recDoubleValue":70,//卡&人脸双重认证,识别阈值分数, 默认 70 分
    "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 (2 代默认) * 2.232 串口 (1 代默认) • 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)

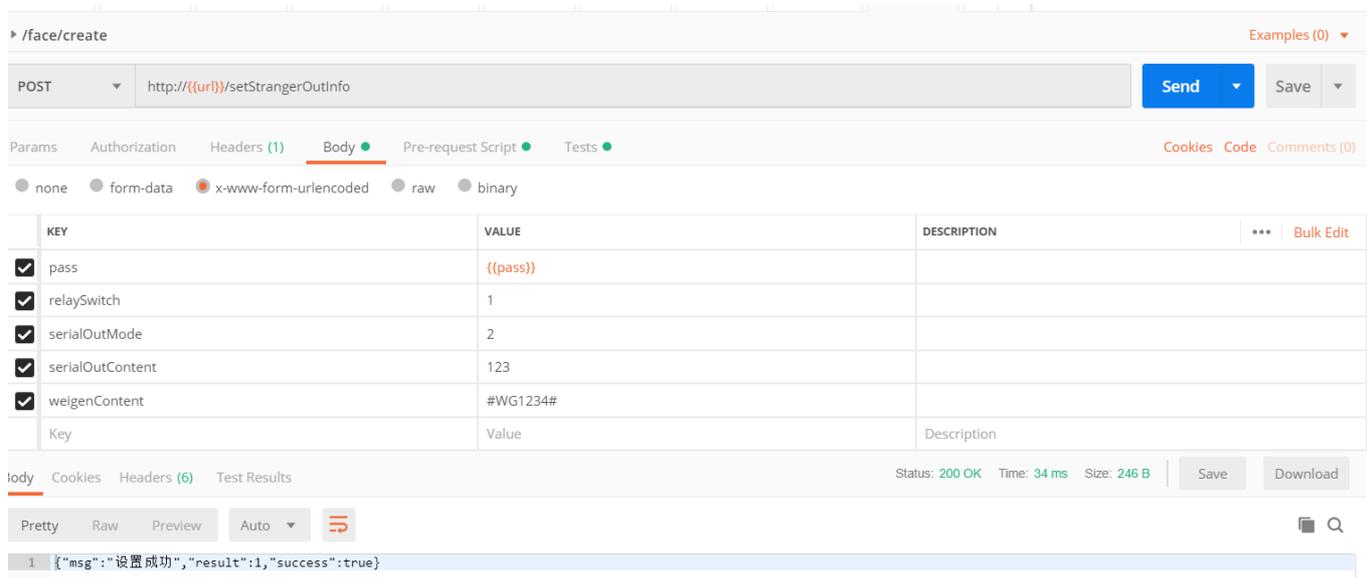
	<ul style="list-style-type: none"> • Uface-M72XX 设备自带的刷卡模块属于 IC 读卡器
卡&人脸双重认证	
recModeCardFaceEnable	卡&人脸双重认证开关 * 1.关闭（默认） * 2.打开
recModeCardFaceIntf	卡号传输接口 * 1.TTL 串口（Uface-M72XX 默认） * 2.232 串口（Uface-M5201 系列默认） <ul style="list-style-type: none"> • Uface-M72XX 设备自带的刷卡模块使用的是 TTL 串口 • TTL 或 232 串口接口只能被一种识别模式使用，若>1 种模式使用 TTL 或 232 串口则报错
recModeCardFaceHardware	外接硬件类型 * 1.IC 读卡器（默认） <ul style="list-style-type: none"> • Uface-M72XX 设备自带的刷卡模块属于 IC 读卡器
卡&人脸双重认证	
recModeIdCardFaceEnable	人证比对开关 * 1.关闭（默认） * 2.打开
recModeIdCardFaceIntf	卡号传输接口 * 1.TTL 串口 * 2.232 串口（默认） * 3.USB <ul style="list-style-type: none"> • 默认使用 232 串口外接新中新身份证阅读器 • USB（或 TTL 或 232）接口只能被一种识别模式使用，若>1 种模式使用 USB（或 TTL 或 232）则报错
recModeIdCardFaceHardware	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"> • 默认使用 232 串口外接新中新身份证阅读器
卡&人脸双重认证、人证比对参数	
recFaceCardValue	人证比对，默认识别分数为 30 分。 <ul style="list-style-type: none"> • 请输入 0-100 之间的整数。分数越高，识别准确率越高，但识别速度会变慢。
recDoubleValue	<ul style="list-style-type: none"> • 卡&人脸双重认证,识别阈值分数，默认 70 分 • 请输入 0-100 之间的整数。分数越高，识别准确率越高，但识别速度会变慢。

2.2.5 陌生人串口 & 韦根输出

请求数据

Method	URL			
POST	http://设备 IP:8090/setStrangerOutInfo			
参数名	描述	类型	必传	附加说明
pass	设备密码	String	Y	
relaySwitch	继电器开关量	Int	Y	0: 关, 1: 开。 (二代设备继电器开门)
serialOutMode	串口输出类型	int	Y	0: 关闭, 1: 开门 (一代设备串口开门), 2: 自定义。
serialOutContent	串口自定义内容	String	Y	串口自定义内容。只允许数字、英文和英文字符, 长度限制 255 个字符。
weigenContent	韦根自定义内容	String	Y	韦根自定义内容。自定义内容传入格式: 韦根 26: #26WG 任意数字#, 韦根 34: #34WG 任意数字#。

postman 示例



The screenshot shows a Postman interface for a POST request to `http://{{url}}/setStrangerOutInfo`. The request body is set to `x-www-form-urlencoded` and contains the following parameters:

KEY	VALUE	DESCRIPTION
pass	{{pass}}	
relaySwitch	1	
serialOutMode	2	
serialOutContent	123	
weigenContent	#WG1234#	

The response status is 200 OK, and the response body is:

```
1 {"msg": "设置成功", "result": 1, "success": true}
```

返回示例

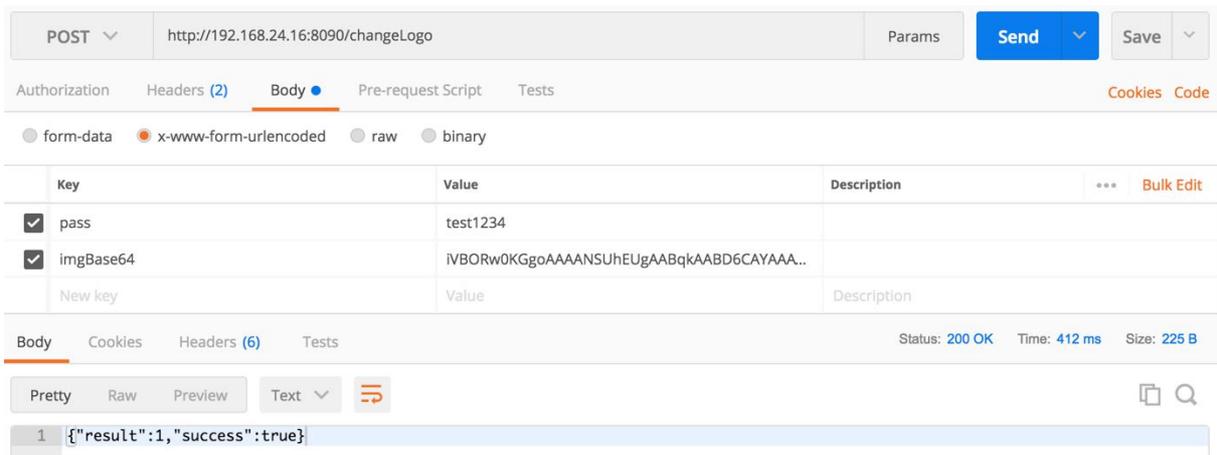
```
{  
  "msg": "设置成功。",  
  "result": 1,//接口调通  
  "success": true//继电器、串口、串口自定义内容、韦根自定义内容设置成功  
}
```

2.2.6 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"> 不加头部，如：data:image/jpg;base64, 若要恢复设备默认 logo，请传入-1 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 尺寸参见设备界面 Logo 图片格式仅支持 JPG 和 PNG

postman 示例



POST http://192.168.24.16:8090/changeLogo

Authorization Headers (2) Body Pre-request Script Tests Cookies Code

form-data x-www-form-urlencoded raw binary

Key	Value	Description
pass	test1234	
imgBase64	iVBORw0KGgoAAAANSUHEUgAABqkAABD6CAYAAA...	
New key	Value	Description

Body Cookies Headers (6) Tests Status: 200 OK Time: 412 ms Size: 225 B

Pretty Raw Preview Text

```
1 {"result":1,"success":true}
```

返回示例

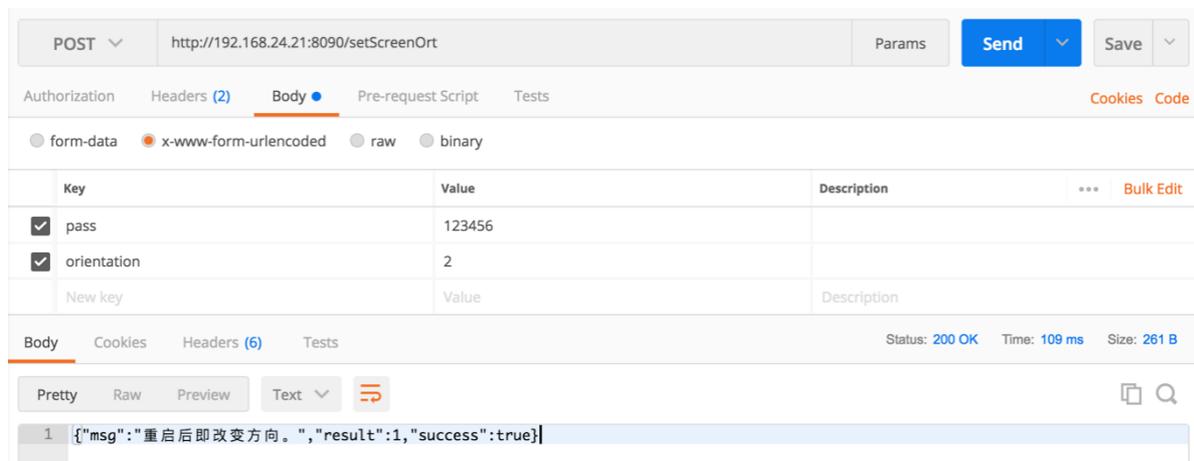
```
{
  "result": 1, //接口调通
  "success": true //Logo 修改成功
}
```

2.2.7 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"> 默认横屏，1：横屏，2：竖屏 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 Uface-M72XX 设备目前只支持竖屏，修改屏幕方向对 Uface-M72XX 设备无效

postman 示例



The screenshot shows a Postman request configuration for a POST method to the URL `http://192.168.24.21:8090/setScreenOrt`. The request body is set to `x-www-form-urlencoded`. Two parameters are defined: `pass` with value `123456` and `orientation` with value `2`. The response status is `200 OK`, with a time of `109 ms` and size of `261 B`. The response body is shown in the `Text` view as `{ "msg": "重启后即改变方向。", "result": 1, "success": true }`.

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
}
```

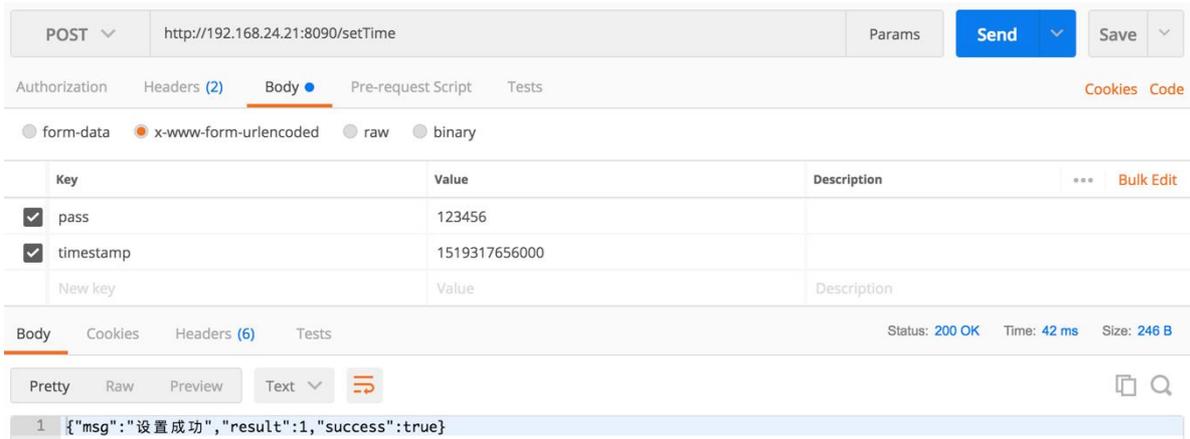
2.2.8 设置设备时间

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"> 配置成功后，设备时间即被改为当前所设置的时间 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 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

 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



The screenshot shows a Postman request configuration for a POST method to the URL `http://192.168.24.21:8090/setTime`. The request body is set to `x-www-form-urlencoded` and contains two parameters: `pass` with value `123456` and `timestamp` with value `1519317656000`. The response status is `200 OK` with a time of `42 ms` and size of `246 B`. The response body is displayed in JSON format: `{\"msg\": \"设置成功\", \"result\": 1, \"success\": true}`.

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  
}
```

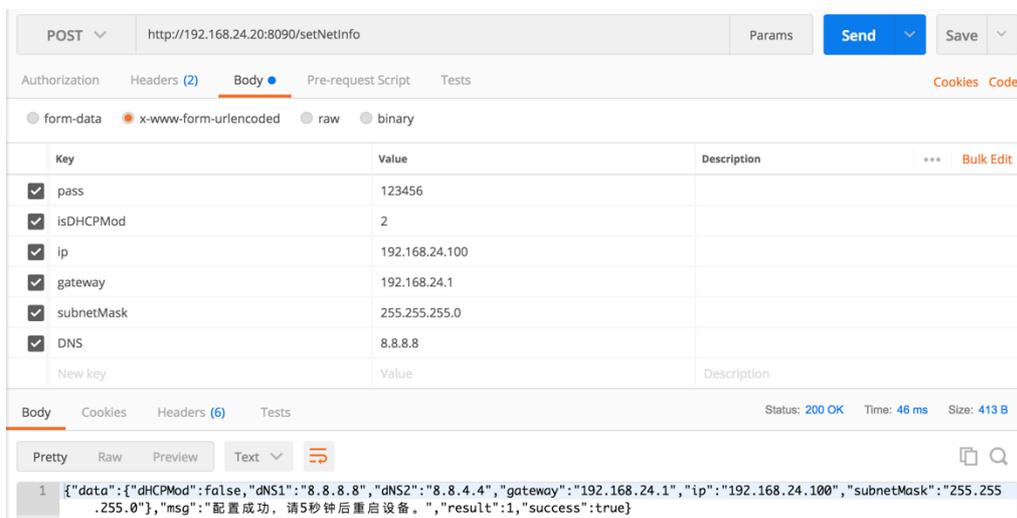
2.2.9 有线网络配置

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 <ul style="list-style-type: none"> 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



The screenshot shows a Postman request configuration for a POST method to the URL `http://192.168.24.20:8090/setNetInfo`. The request body is set to `x-www-form-urlencoded` and contains the following parameters:

Key	Value	Description
pass	123456	
isDHCPMod	2	
ip	192.168.24.100	
gateway	192.168.24.1	
subnetMask	255.255.255.0	
DNS	8.8.8.8	

The response status is `200 OK` with a time of `46 ms` and a size of `413 B`. The response body is shown in JSON format:

```

1 [{"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,"success":true}]
    
```

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
}
```

2.2.10 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: • 自动获取 IP {"ssId":"TP-LINK_E2.4G","pwd":"test-1234","isDHCPMod":true} • 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"}

i 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
}
```

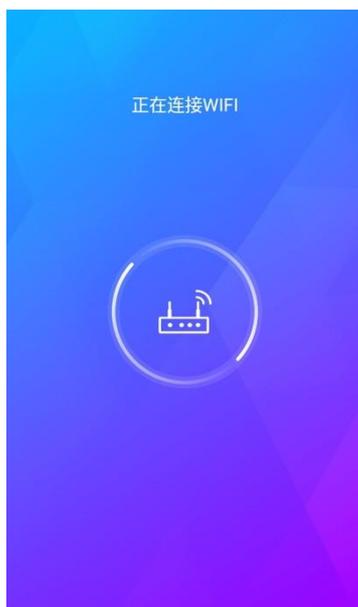
postman 示例



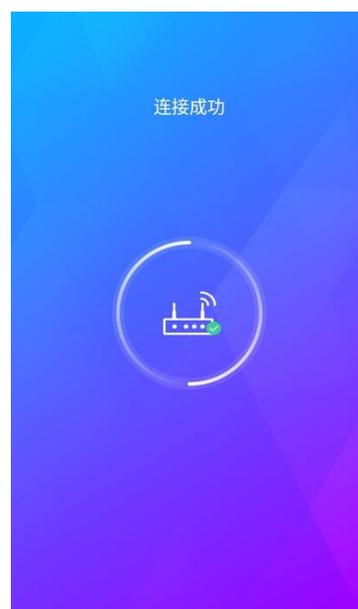
返回示例

```
{
  "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



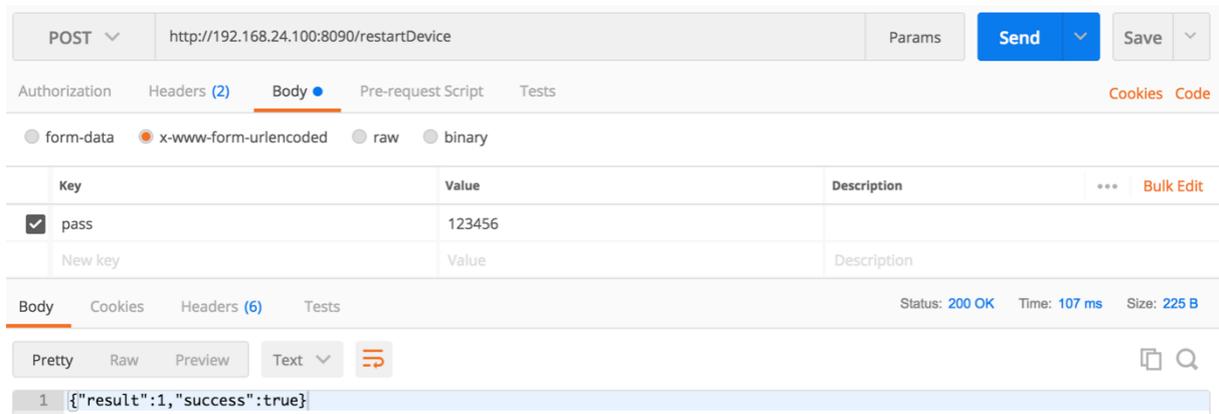
Wi-Fi 连接成功

2.2.11 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



The screenshot shows a Postman interface for a POST request. The URL is `http://192.168.24.100:8090/restartDevice`. The request body is set to `x-www-form-urlencoded` and contains a single parameter: `pass` with the value `123456`. The response status is `200 OK` with a time of `107 ms` and a size of `225 B`. The response body is displayed in JSON format: `{\"result\":1,\"success\":true}`.

返回示例

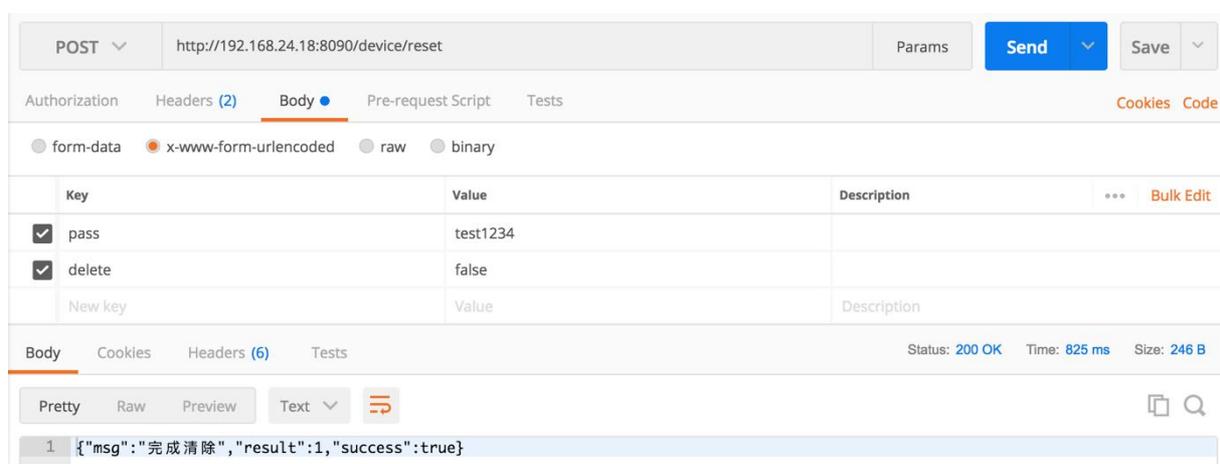
```
{
  "result":1,//接口调通
  "success":true//设备开始执行重启指令
}
```

2.2.12 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"> 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 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 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

postman 示例



The screenshot shows a Postman request configuration for a POST method to the URL `http://192.168.24.18:8090/device/reset`. The body is set to `x-www-form-urlencoded` with two parameters: `pass` with value `test1234` and `delete` with value `false`. The response status is `200 OK` with a time of `825 ms` and size of `246 B`. The response body is displayed in JSON format: `{\"msg\": \"完成清除\", \"result\": 1, \"success\": true}`.

i 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  
}
```

2.2.13 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	
type	设备交互类型	int	N	非必传，1：开门；2：串口输出；3：韦根输出；4：表示自定义文字弹框、自定义语音播报；默认是 1
content	输出内容	String	N	<p>Type=1: 开门</p> <p>Type=2:</p> <ul style="list-style-type: none"> • 串口，只允许数字、英文和英文字符，长度限制 255 个字符。 • 串口支持输出韦根信号，一代设备需要外接串口→韦根信号转换小板，小板由本公司定制。自定义内容传入格式：韦根 26：#26WG 任意数字#，韦根 34：#34WG 任意数字# <p>Type=3: 支持二代设备韦根输出 韦根，只允许数字。 韦根 26：#26WG 任意数字#，韦根 34：#34WG 任意数字#</p> <p>Type=4:</p> <ul style="list-style-type: none"> • content={"ttsModContent":"自定义语音","displayModContent":"自定义文字"}；其他见“沃土&离线需求_20181204_V6.docx”中“远程交互新增串口输出和韦根输出”。

postman 示例

POST http://192.168.11.92:8090/device/openDoorControl Send Save

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

none form-data x-www-form-urlencoded raw binary

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> pass	{{pass}}	
<input checked="" type="checkbox"/> type	1	
<input type="checkbox"/> content	*9j/4AAQSkZJRgABAQAAQABAAD/zwBDAAEBAQEBAQEBAQEBAQEB...	
Key	Value	Description

Body Cookies Headers (5) Test Results Status: 200 OK Time: 98 ms Size: 225 B Download

Pretty Raw Preview XML UI Q

```
1 [{"msg": "开门成功", "result": 1, "success": true}]
```

POST http://192.168.11.92:8090/device/openDoorControl Send Save

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

none form-data x-www-form-urlencoded raw binary

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> pass	{{pass}}	
<input checked="" type="checkbox"/> type	2	
<input checked="" type="checkbox"/> content	123456abc	
Key	Value	Description

Body Cookies Headers (5) Test Results Status: 200 OK Time: 94 ms Size: 231 B Download

Pretty Raw Preview XML UI Q

```
1 [{"msg": "输出串口成功", "result": 1, "success": true}]
```

POST http://192.168.11.92:8090/device/openDoorControl Send Save

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

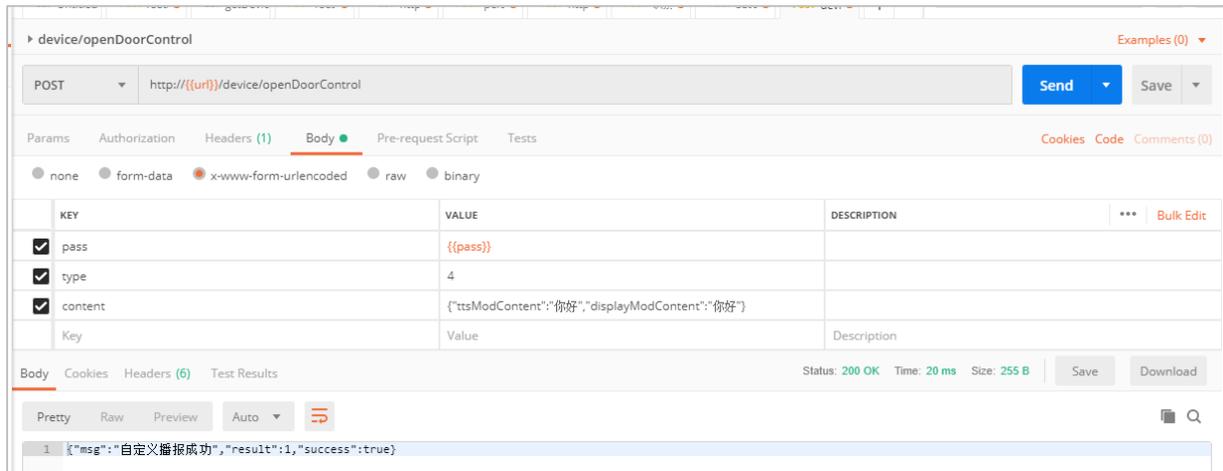
none form-data x-www-form-urlencoded raw binary

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> pass	{{pass}}	
<input checked="" type="checkbox"/> type	3	X
<input checked="" type="checkbox"/> content	#WG123456#	
Key	Value	Description

Body Cookies Headers (5) Test Results Status: 200 OK Time: 68 ms Size: 231 B Download

Pretty Raw Preview XML UI Q

```
1 [{"msg": "输出串口成功", "result": 1, "success": true}]
```



device/openDoorControl

POST http://{{url}}/device/openDoorControl

Params Authorization Headers (1) Body Pre-request Script Tests

none form-data x-www-form-urlencoded raw binary

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> pass	{{pass}}	
<input checked="" type="checkbox"/> type	4	
<input checked="" type="checkbox"/> content	{\"ttsModContent\": \"你好\", \"displayModContent\": \"你好\"}	

Body Cookies Headers (6) Test Results

Status: 200 OK Time: 20 ms Size: 255 B

1 {\"msg\": \"自定义播报成功\", \"result\": 1, \"success\": true}

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

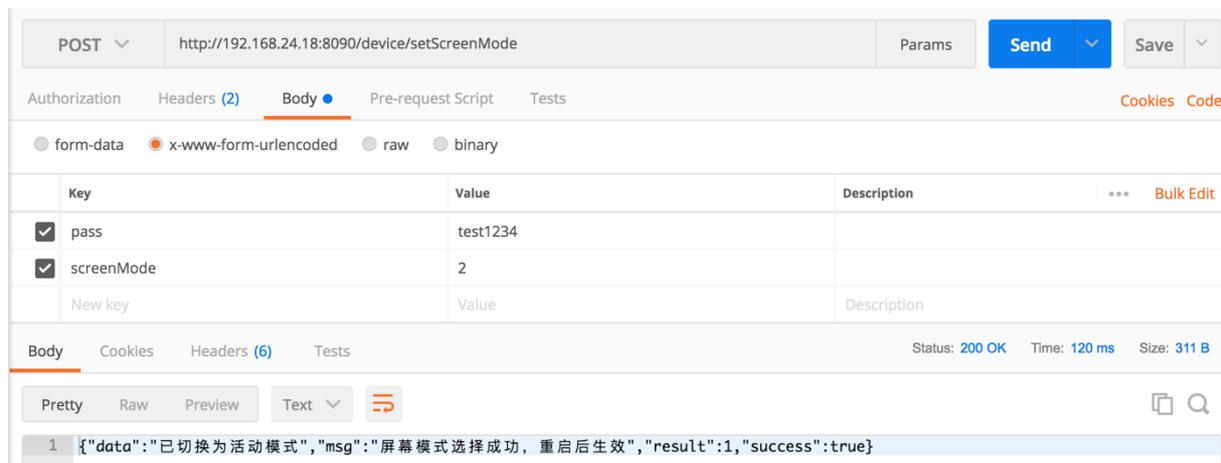
```
{
  "msg": "开门成功",
  "result": 1, //接口调通
  "success": true //设备开始执行开门指令
}
```

2.2.14 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 参见设备配置 , Logo 编辑 参见修改 Logo 2: 自定义大屏背景模式, 更换背景图片 参见大屏背景图片配置 • 大屏界面 参见设备配置

postman 示例



POST http://192.168.24.18:8090/device/setScreenMode

Authorization Headers (2) Body Pre-request Script Tests Cookies Code

form-data x-www-form-urlencoded raw binary

Key	Value	Description
pass	test1234	
screenMode	2	
New key	Value	Description

Body Cookies Headers (6) Tests Status: 200 OK Time: 120 ms Size: 311 B

Pretty Raw Preview Text

```
1 {"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, "success": true}
```

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

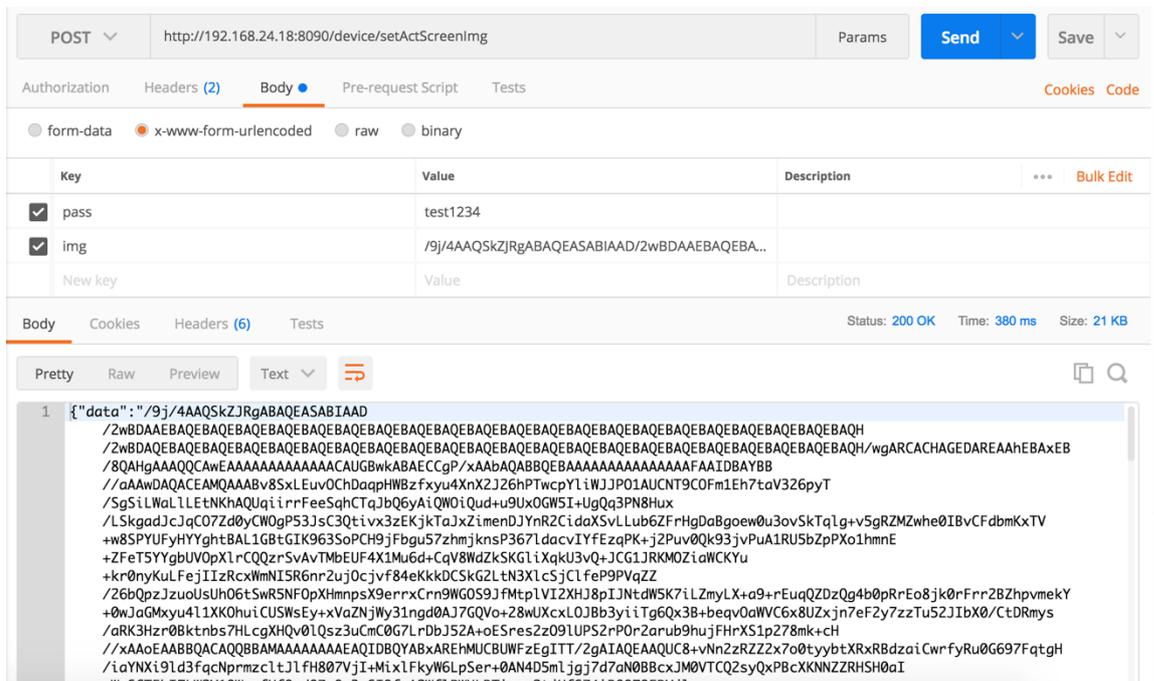
```
{  
  "data": "已切换为活动模式",  
  "msg": "屏幕模式选择成功，重启后生效",  
  "result": 1,//接口调通  
  "success": true//大屏模式设置成功  
}
```

2.2.15 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  
}
```

2.2.16 隐藏 IP、设备序列号、人数和照片数

请求数据

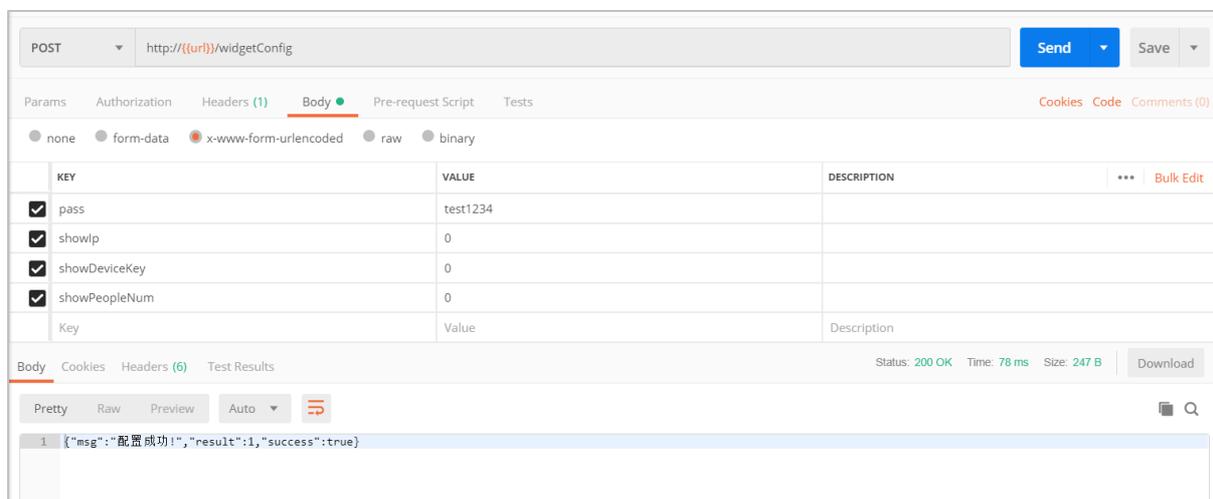
Method	URL			
POST	http://设备 IP:8090/widgetConfig			
参数名	描述	类型	必传	附加说明
pass	设备密码	String	Y	
showIp	设备 IP	int	Y	0: 显示, 1: 隐藏; 默认为 0
showDeviceKey	设备序列号	int	Y	0: 显示, 1: 隐藏; 默认为 0
showPeopleNum	人员/照片	int	Y	0: 显示, 1: 隐藏; 默认为 0

i 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

Postman 示例:



The screenshot shows a Postman interface for a POST request to `http://{{url}}/widgetConfig`. The request body is configured as `x-www-form-urlencoded` with the following parameters:

KEY	VALUE	DESCRIPTION
pass	test1234	
showIp	0	
showDeviceKey	0	
showPeopleNum	0	

The response is shown in the bottom panel, indicating a successful status (200 OK) with the following JSON body:

```
1 {"msg": "配置成功!", "result": 1, "success": true}
```

返回示例:

```
{
  "msg": "配置成功",
  "result": 1, //接口调通
  "success": true //回调地址设置成功
}
```

2.2.17 识别回调

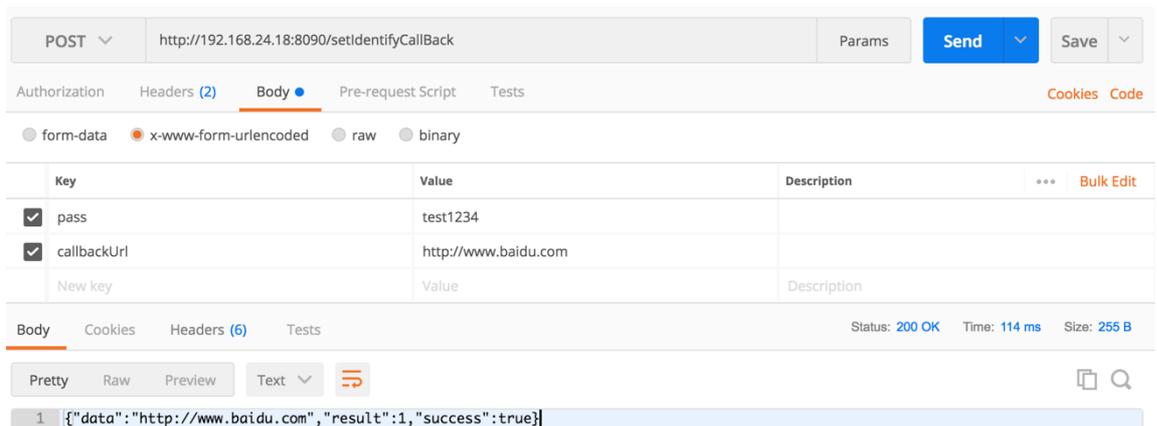
请求数据

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"> 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, 参见回调字段说明 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 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\\&%_\\./-~]*)*?"

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);
}
```

i 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，从数据库中读取出回调失败 (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"> 识别方式（face/faceAndcard/idcard/card）、识别出的人员类 face_0（刷脸识别，且该人员在 passtime 权限时间内） face_1（刷脸识别，且该人员在 passtime 权限时间外） face_2（刷脸识别，且识别失败或识别到的是陌生人） card_0（刷卡识别，且该人员在 passtime 权限时间内） card_1（刷卡识别，且该人员在 passtime 权限时间外） faceAndcard_0（双重认证，且该人员在 passtime 权限时间内） faceAndcard_1（双重认证，且该人员在 passtime 权限时间外） faceAndcard_2（双重认证，且识别失败或识别到的是陌生人） idcard_0

		(人证比对, 且该人员在 passtime 权限时间内) idcard_2 (人证比对, 且识别失败或识别到的是陌生人)
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

2.2.18 注册照片回调

请求数据

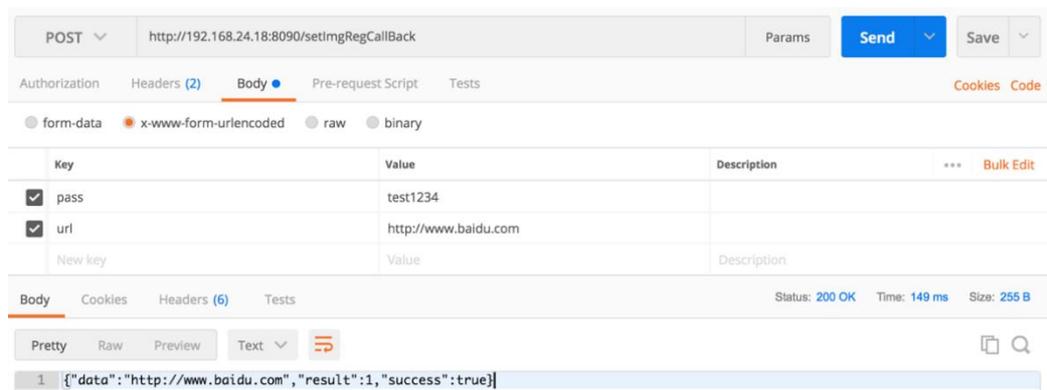
Method	URL			
POST	http://设备 IP:8090/setImgRegCallBack			
参数名	描述	类型	必传	附加说明
pass	设备密码	String	Y	
url	回调地址	String	Y	<ul style="list-style-type: none"> 照片注册成功（包括设备拍照注册）后，设备会向预先设置的回调接口 POST 字段 deviceKey、personId、time、imgPath、faceId、ip、feature 和 featureKey，参见回调字段说明 传入内容为空可以清空回调地址，清空后识别成功将不再进行回调 回调 URL 需要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

i 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	特征密钥, 通过特征码注册时需要该字段进行特征有效性校验

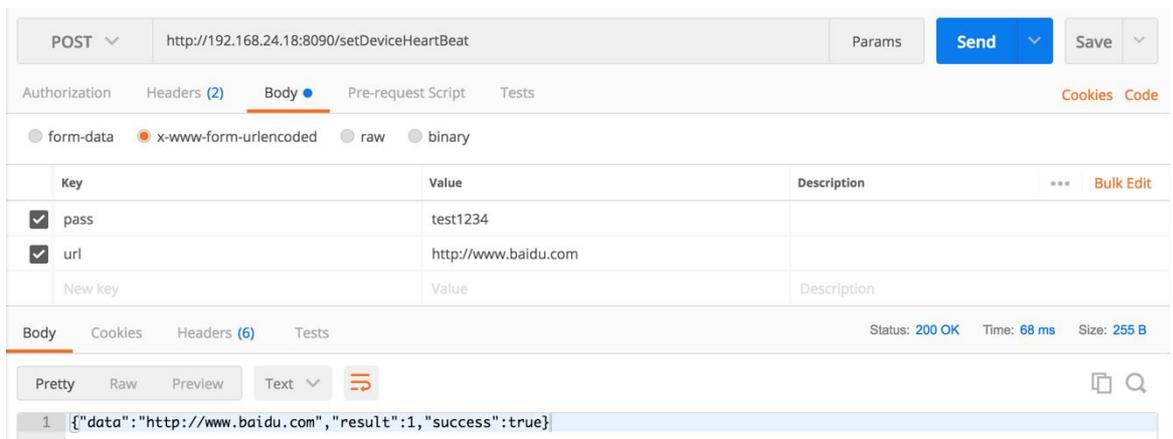
2.2.19 设备心跳回调

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"> 设备每隔一分钟会向该接口 POST 字段 deviceKey、time、ip、personCount、faceCount 和 version, 参见回调字段说明 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 回调 URL 需要符合正则表达式: String check = <code>"((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\\&%_\\./~~]*)?"</code>

i 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



The screenshot shows a Postman interface for a POST request to `http://192.168.24.18:8090/setDeviceHeartBeat`. The request body is `x-www-form-urlencoded` with two parameters: `pass` (value: `test1234`) and `url` (value: `http://www.baidu.com`). The response status is `200 OK` with a time of `68 ms` and size of `255 B`. The response body is `{\"data\": \"http://www.baidu.com\", \"result\": 1, \"success\": true}`.

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

i 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

2.2.20 数据写入接口

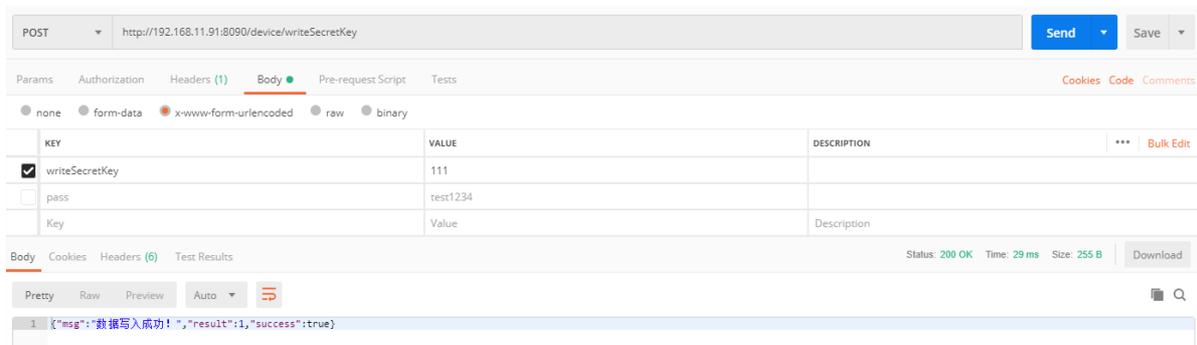
URL: /device/writeSecretKey

method: POST

参数列表:

参数名	描述	类型	必传	附加说明
writeSecretKey	写入的数据	String	Y	<ul style="list-style-type: none"> 写数据进口，允许对接方存放数据到设备固定区域，该区域数据，无论设备是否重启，都不会被清空。

postman 示例



返回示例

```
{
  "msg": "数据写入成功!",
  "result": 1,
  "success": true
}
```

2.2.21 数据读出接口

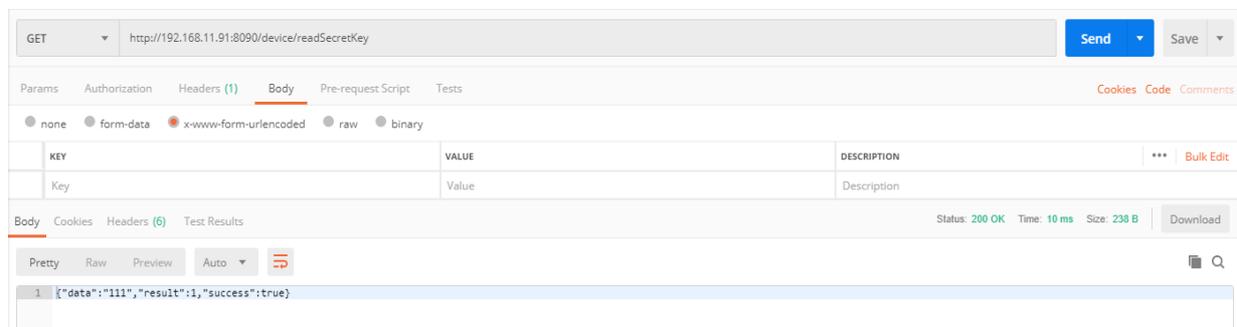
URL: /device/readSecretKey

method: GET

参数列表:

参数名	描述	类型	必传	附加说明
无参				

postman 示例



返回示例

```
{
  "data": "111",
  "result": 1,
  "success": true
}
```

2.3 人员管理类接口

2.3.1 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"> • Json 示例: {"id":"6689","idcardNum":"3147B78440029087","name":"TONY","IDNumber":"330821199602132033","facePermission":2,"idCardPermission":2,"faceAndCardPermission":2,"IDPermission":2} • id、idcardNum、IDNumber、facePermission、idCardPermission、faceAndCardPermission、IDPermission 内容可传空。 • 若传入 id 值，id 只允许数字和英文字母，区分大小写，位数限制为 0-255 位。

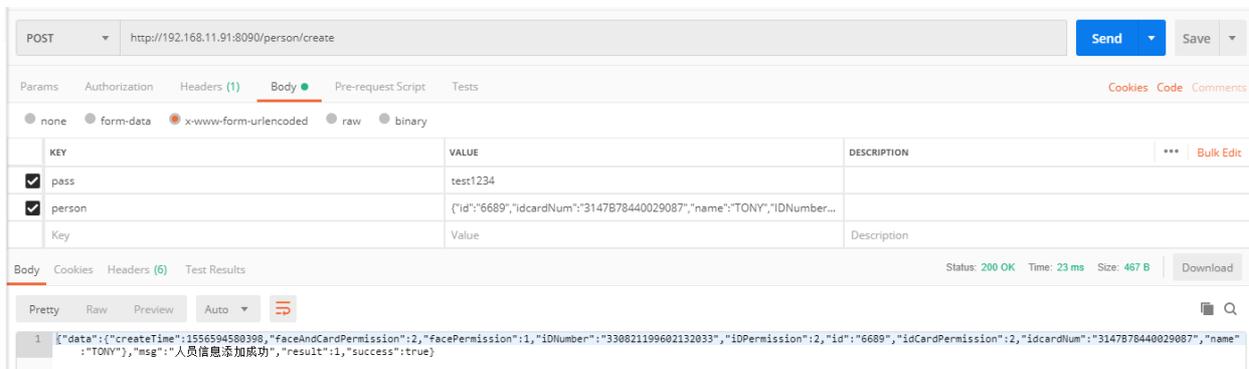
Json 字段说明:

参数名	描述	类型	必传	附加说明
id	人员 id	String	N	<ul style="list-style-type: none"> • id 为人员标识，只允许数字和英文字母，限制 255 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
name	姓名	String	Y	
idcardNum	卡号	String	N	<ul style="list-style-type: none"> • 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
IDNumber	身份证号	String	N	注册时可以不填，长度、内容无限制。
facePermission	刷脸权限	int	N	1: 关 2: 开 (默认)
idCardPermission	刷卡权限	int	N	1: 关 2: 开 (默认)

faceAndCard Permission	人卡合一权限	int	N	1: 关 2: 开 (默认)
IDPermission	人证比对权限	int	N	1: 关 2: 开 (默认)

postman 示例

- i 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 255 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
    "IDNumber": "33082119920811782", //String, 身份证编号。
    "facePermission": 2, //刷脸权限，默认为 2 开。
    "idCardPermission": 2, //刷卡权限，默认为 2 开。
    "faceAndCardPermission": 2, //人卡权限，默认为 2 开。
    "IDPermission": 2, //人证比对权限 默认为 2 开。
  },
  "msg": "人员信息添加成功",
  "result": 1, //接口调通
  "success": true //人员注册成功
}
```

i 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。

2.3.2 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"> 给指定人员 id 注册卡号 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 注册成功后，可调用人员信息查询接口查询该人员的卡号

i 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



POST http://192.168.24.100:8090/face/icCardRegist

form-data x-www-form-urlencoded raw binary

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

Body Cookies Headers (6) Tests Status: 200 OK Time: 56 ms Size: 354 B

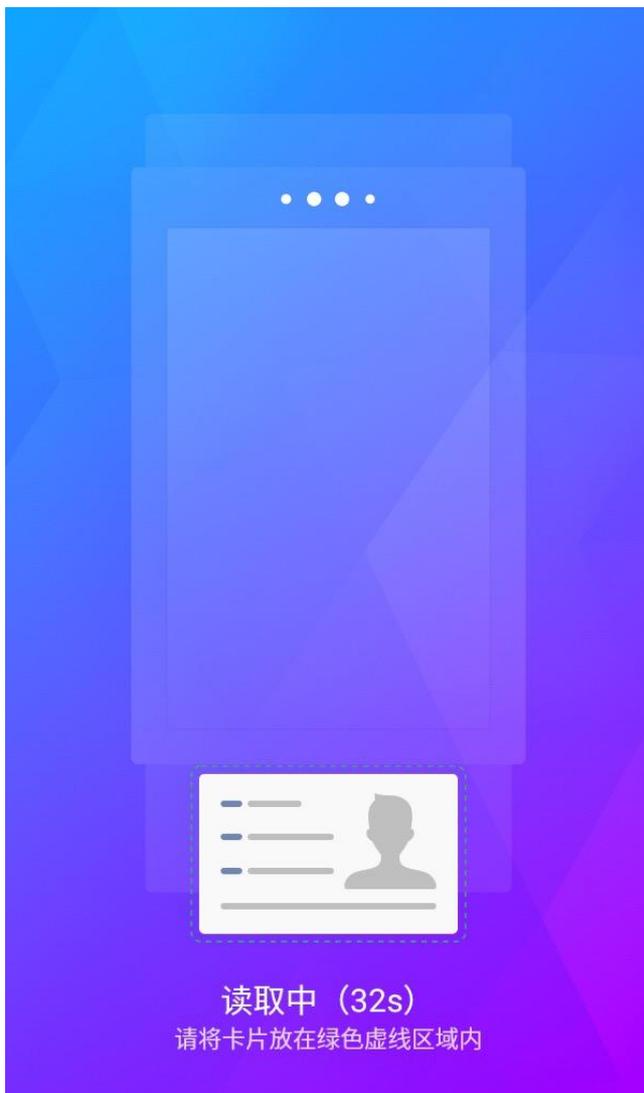
1 [{"msg": "正在开启IC卡注册模式，注册成功后可根据personId查询该人的IC卡ID。请根据引导完成注册。", "result": 1, "success": true}]

i 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

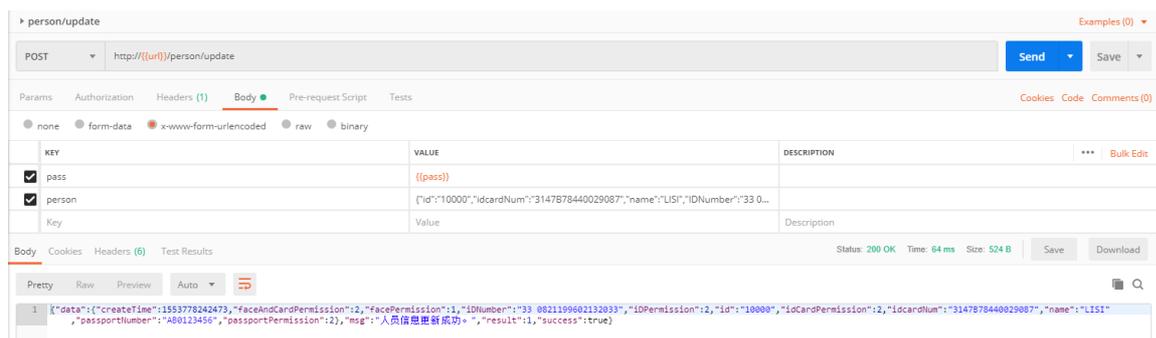


2.3.3 人员更新

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"> Json 示例: <pre>{ "id": "6689", "idcardNum": "3147B78440029087", "name": "TONY", "ID Number": "330821199602132033", "facePermission": 2, "idCardPermission": 2, "faceAndCardPermission": 2, "IDPermission": 2 }</pre> id、name 不可为空 idcardNum 传空则会清空原信息 iDNumber、facePermission、idCardPermission、faceAndCardPermission、iDPermission 传空则保留上一次的值

postman 示例



The screenshot shows a Postman request configuration for the endpoint `http://(url)/person/update`. The request method is `POST`. The body is set to `form-urlencoded` and contains two parameters: `pass` with value `{{pass}}` and `person` with value `{"id":"10000","idcardNum":"3147B78440029087","name":"LISI","IDNumber":"33 0..."}`. The response status is `200 OK` with a time of `64 ms` and size of `524 B`. The response body is shown in the `Pretty` view:

```
[{"data":{"createTime":1553778242473,"faceAndCardPermission":2,"facePermission":1,"IDNumber":"33 0821199602132033","IDPermission":2,"id":"10000","idCardPermission":2,"idcardNum":"3147B78440029087","name":"LISI","passportNumber":"AB0123456","passportPermission":2},"msg":"人员信息更新成功","result":1,"success":true}
```

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 255 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
    "IDNumber": "33082119920811782", //String, 身份证编号。
    "facePermission": 2, //刷脸权限，默认为 2 开。
    "idCardPermission": 2, //刷卡权限，默认为 2 开。
    "faceAndCardPermission": 2, //人卡权限，默认为 2 开。
    "IDPermission": 2, //人证比对权限 默认为 2 开。
  },
  "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
}
```

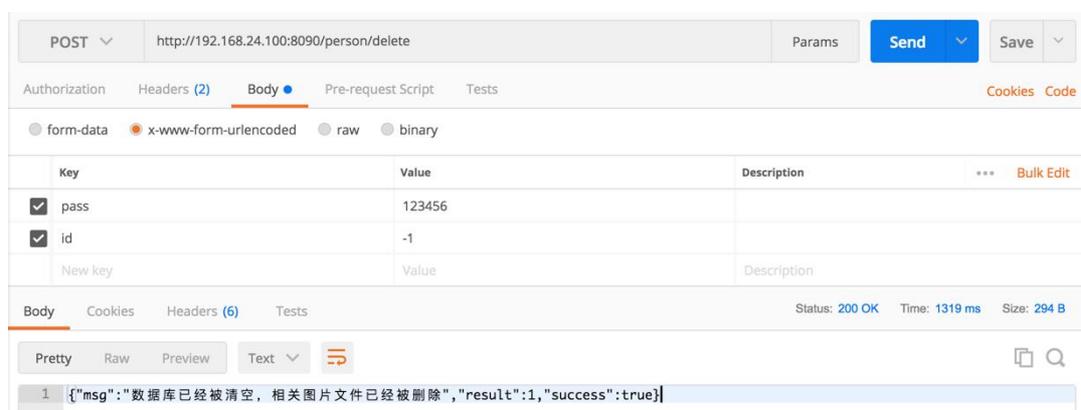
2.3.4 人员删除（批量）

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"> 删除多个人员，personId 用英文逗号拼接 传入-1 则删除所有人员 以下数据会被删除，本地将不再做存储： <ul style="list-style-type: none"> 该人员 id、人员信息 该人员对应的识别记录、现场抓拍照片 该人员对应的注册照片

postman 示例

(1) id 传入-1，删除设备上的所有人员，数据不可恢复



POST http://192.168.24.100:8090/person/delete

Params Send Save

Authorization Headers (2) Body Pre-request Script Tests Cookies Code

form-data x-www-form-urlencoded raw binary

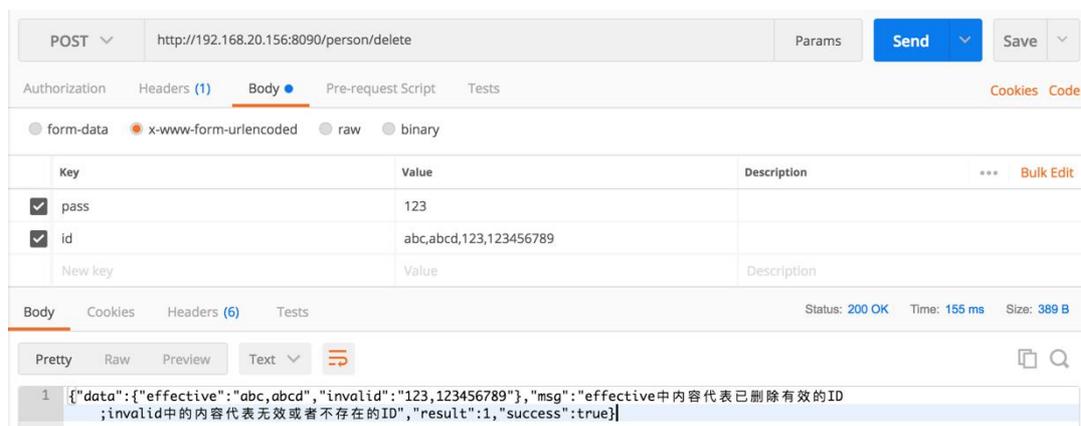
Key	Value	Description
pass	123456	
id	-1	

Body Cookies Headers (6) Tests Status: 200 OK Time: 1319 ms Size: 294 B

Pretty Raw Preview Text

```
1 [{"msg": "数据库已经被清空，相关图片文件已经被删除", "result": 1, "success": true}]
```

(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。



POST http://192.168.20.156:8090/person/delete

Params Send Save

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

form-data x-www-form-urlencoded raw binary

Key	Value	Description
pass	123	
id	abc,abcd,123,123456789	

Body Cookies Headers (6) Tests Status: 200 OK Time: 165 ms Size: 389 B

Pretty Raw Preview Text

```
1 [{"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, "success": true}]
```

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
}
```

2.3.5 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"> 查询指定 id 的人员信息 id 传入-1 表示不局限于 personId 查询人员
length	每页最大数量	Int	N	<ul style="list-style-type: none"> length 传入值要求为(0,1000]之间的正整数 若 length 不传，默认值为 1000，返回最近添加的 1000 人
index	页码	Int	N	页码，从 0 开始

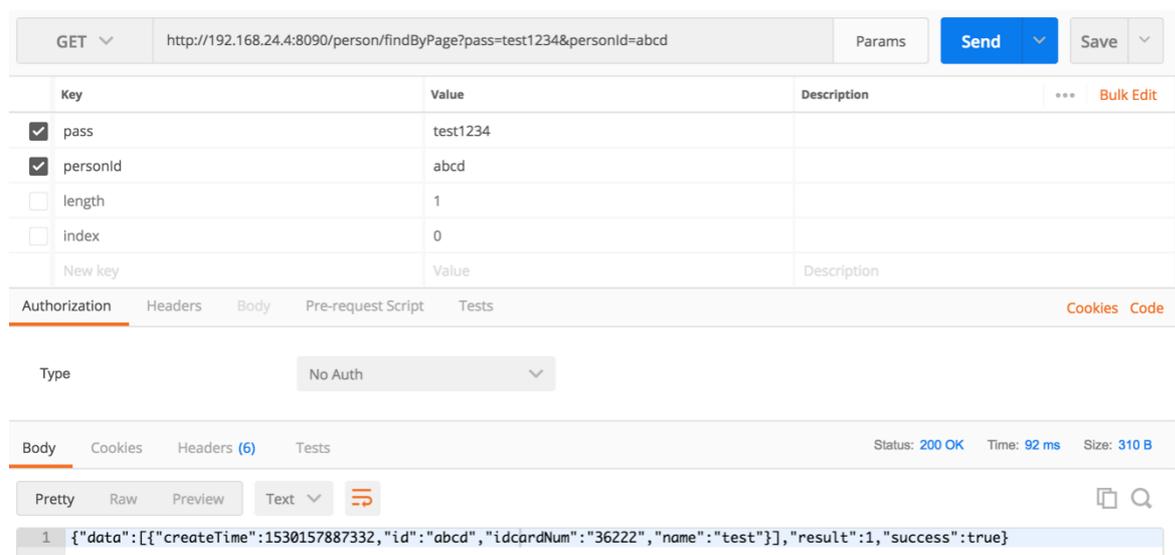
不同情况说明

i 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



The screenshot shows a Postman interface for a GET request. The URL is `http://192.168.24.4:8090/person/findByPage?pass=test1234&personId=abcd`. The Params tab is active, displaying a table of parameters:

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

The Body tab shows the response in JSON format:

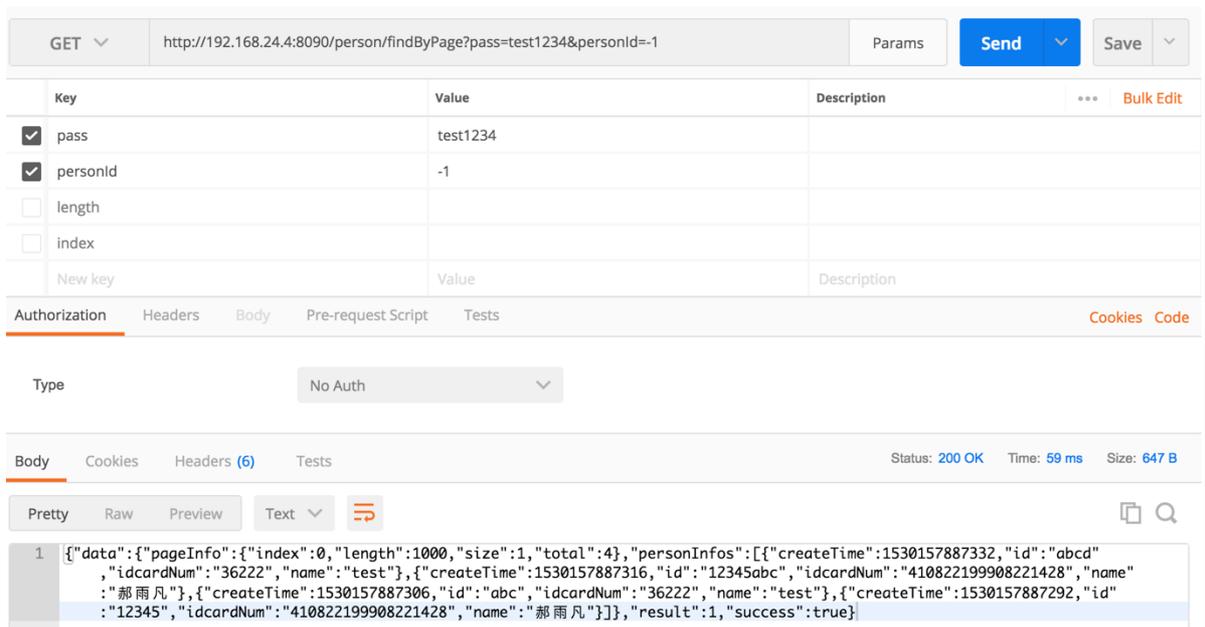
```
1 [{"data":[{"createTime":1530157887332,"id":"abcd","idcardNum":"36222","name":"test"}],"result":1,"success":true}
```

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
    "IDNumber": "33082119920811782" //String，身份证编号。
    "facePermission": 2, //刷脸权限，默认为 2 开。
    "idCardPermission": 2, //刷卡权限，默认为 2 开。
    "faceAndCardPermission": 2, //人卡权限，默认为 2 开。
    "IDPermission": 2, //人证比对权限 默认为 2 开
  }],
  "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: Status: 200 OK Time: 59 ms Size: 647 B

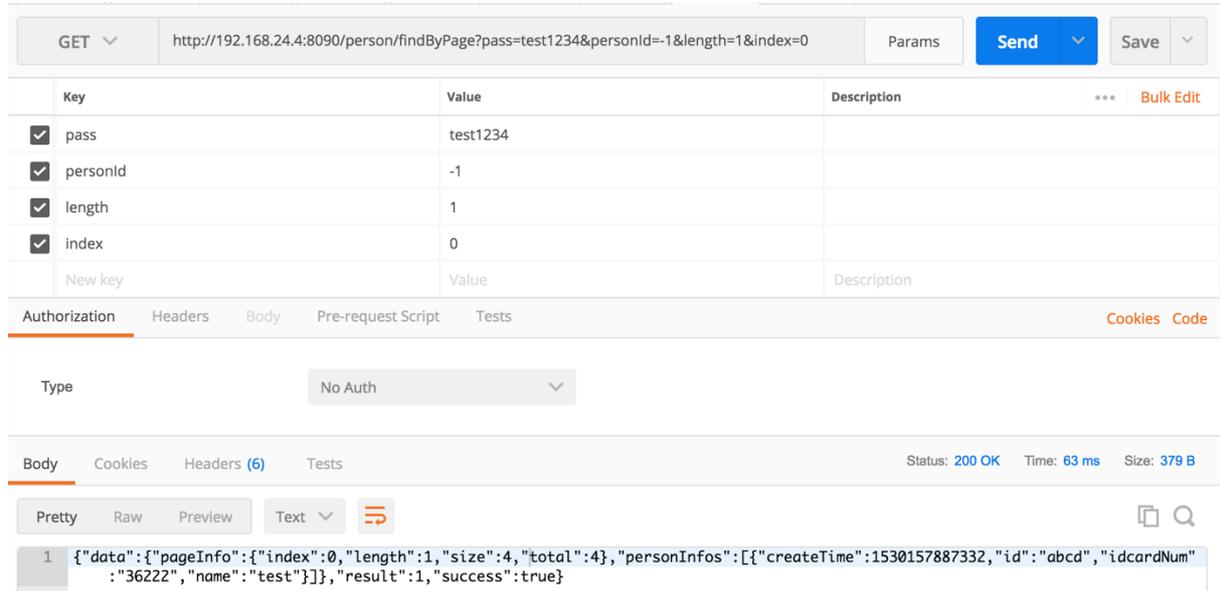
```
1 [{"data":{"pageInfo":{"index":0,"length":1000,"size":1,"total":4},"personInfos":[{"createTime":1530157887332,"id":"abcd","idcardNum":"36222","name":"test"}, {"createTime":1530157887316,"id":"12345abc","idcardNum":"410822199908221428","name":"郝雨凡"}, {"createTime":1530157887306,"id":"abc","idcardNum":"36222","name":"test"},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



GET `http://192.168.24.4:8090/person/findByPage?pass=test1234&personId=-1&length=1&index=0` Params Send Save

Key	Value	Description
pass	test1234	
personId	-1	
length	1	
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: 63 ms Size: 379 B

Pretty Raw Preview Text

```
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

```
{
  "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
}
```

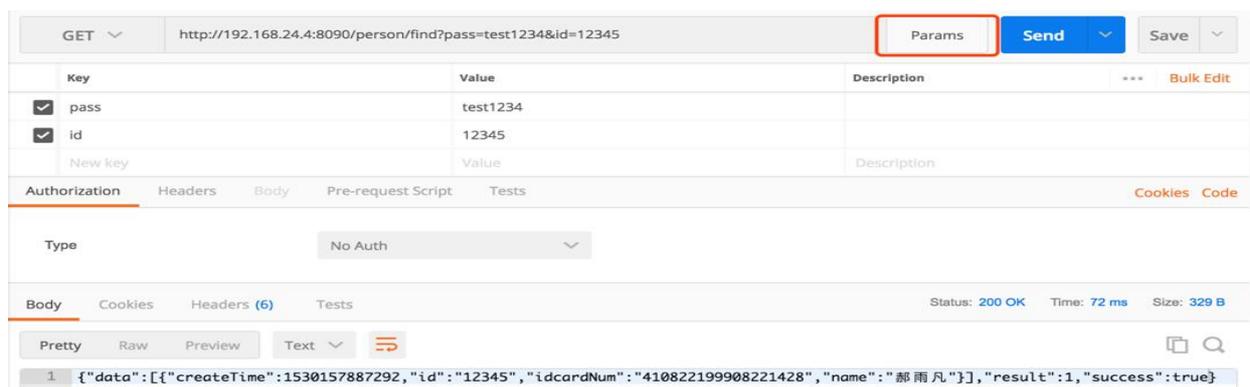
2.3.6 人员信息查询

请求数据

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"> 查询指定 id 的人员信息 id 传入-1 为查询设备上所有人员信息

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



备注注册人数较多时查询导致设备崩溃。

4.3.7 时间段权限设置

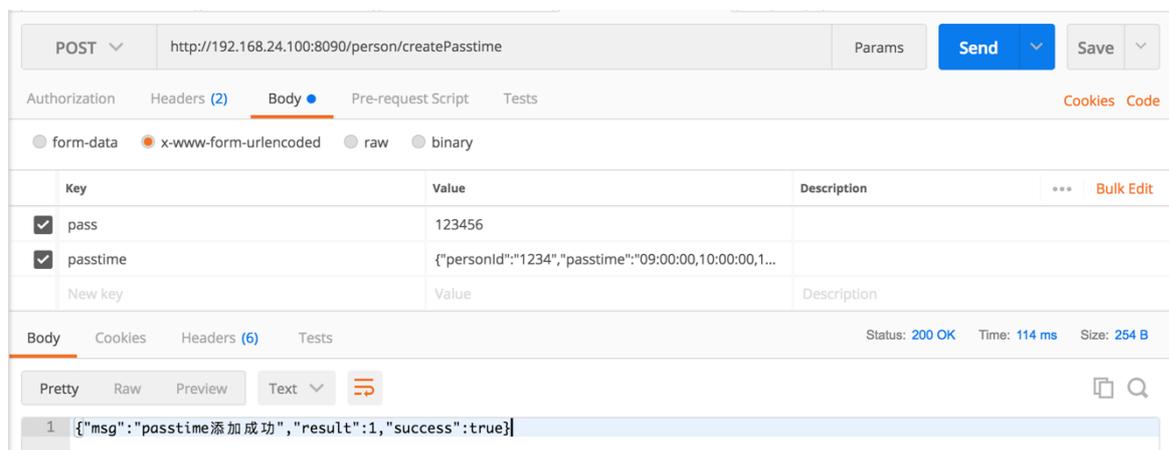
```
{
  "data": [{
    "createTime": 1530157887292, //人员注册成功、写入数据库的时间，Unix 毫秒时间戳；
    //较低版本人员注册成功后没有时间戳一项，因此更新到新版本后，会自动对数据库内所有
    //createTime 为空的人员进行自动填充，以设备重启时的时间为基础，按照人员写入数据库的顺序
    //依次增加 createTime（人员时间间隔无规律）。
    "id": "12345", //人员 id，限制 64 位
    "idcardNum": "410822199908221428", //卡号，长度无限制
    "name": "郝雨凡" //姓名，不可为空
    "IDNumber": "33082119920811782" //String, 身份证编号。
    "facePermission": 2, //刷脸权限，默认为 2 开。
    "idCardPermission": 2, //刷卡权限，默认为 2 开。
    "faceAndCardPermission": 2, //人卡权限，默认为 2 开。
    "IDPermission": 2, //人证比对权限 默认为 2 开

  }],
  "result": 1, //接口调通
  "success": true //成功查询到人员信息并返回
}
```

请求数据

postman 示例

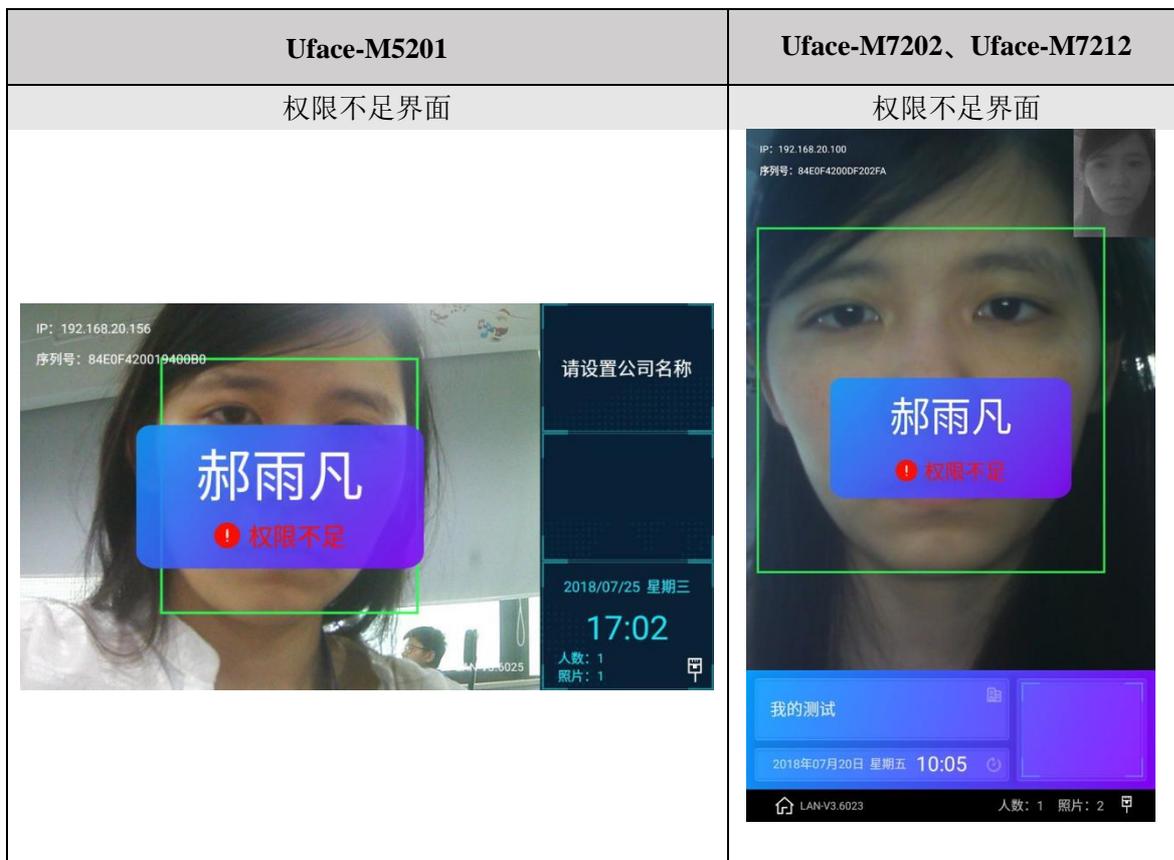
Method	URL			
POST	http://设备 IP:8090/person/createPasstime			
参数名	描述	类型	必传	附加说明
pass	设备密码	String	Y	
passtime	某人员每日允许进入的时间段	Json	Y	<ul style="list-style-type: none"> • Json 示例: {"personId":"9eccc839cd7941c5a4d3165202dd3c32","passtime":"09:00:00,10:00:00,17:00:00,17:30:00,18:30:00,20:25:00"} • 范围为[00:00:00,23:59:59], 以设备上的时间为标准 • 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 • 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



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

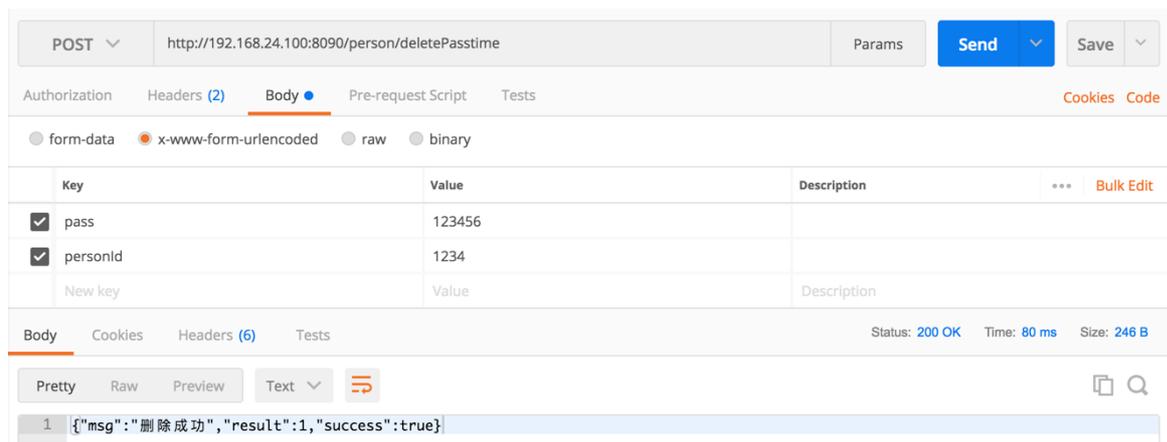
```
{
  "msg": "passtime 添加成功",
  "result": 1,//接口调通
  "success": true//人员 passtime 设置成功
}
```

2.3.7 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"> 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 传入-1，可清除所有人员的 passtime

postman 示例



POST http://192.168.24.100:8090/person/deletePasstime

Authorization Headers (2) Body Pre-request Script Tests Cookies Code

form-data x-www-form-urlencoded raw binary

Key	Value	Description
pass	123456	
personId	1234	
New key	Value	Description

Body Cookies Headers (6) Tests Status: 200 OK Time: 80 ms Size: 246 B

Pretty Raw Preview Text

```
1 {"msg": "删除成功", "result": 1, "success": true}
```

返回示例

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

2.3.8 人员有效期限设置

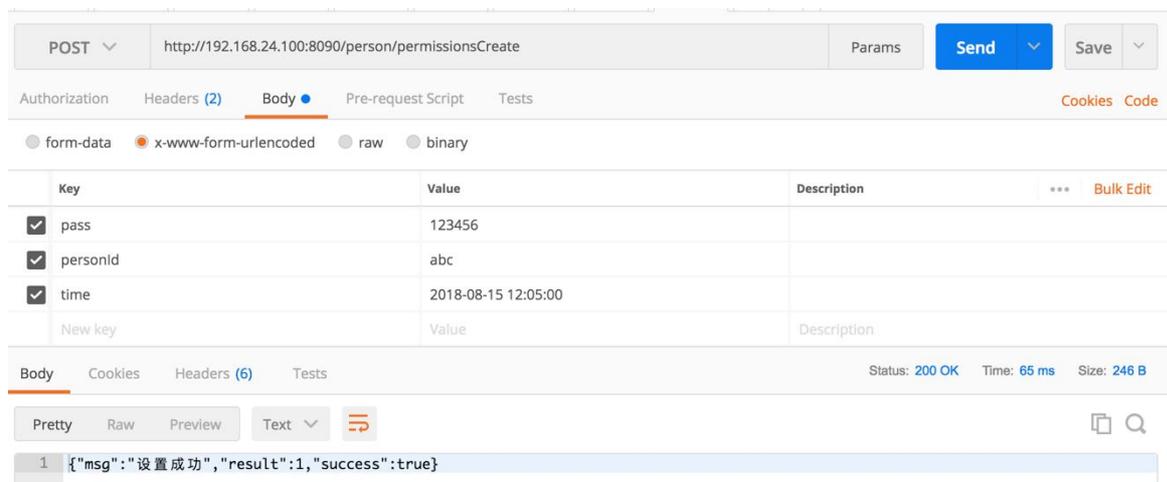
请求数据

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"> • 该人员 id、人员信息 • 该人员对应的识别记录、现场抓拍照拍 • 该人员对应的注册照片
time	有效期限时间，定时删除该人员	String	Y	<ul style="list-style-type: none"> • 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 • 传入时间格式为（年-月-日 时:分:秒）： 2017-07-15 12:05:00

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](#)）。



POST http://192.168.24.100:8090/person/permissionsCreate

Authorization Headers (2) Body Pre-request Script Tests Cookies Code

form-data x-www-form-urlencoded raw binary

Key	Value	Description
pass	123456	
personid	abc	
time	2018-08-15 12:05:00	
New key	Value	Description

Body Cookies Headers (6) Tests Status: 200 OK Time: 65 ms Size: 246 B

Pretty Raw Preview Text

```
1 {"msg": "设置成功", "result": 1, "success": true}
```

返回示例

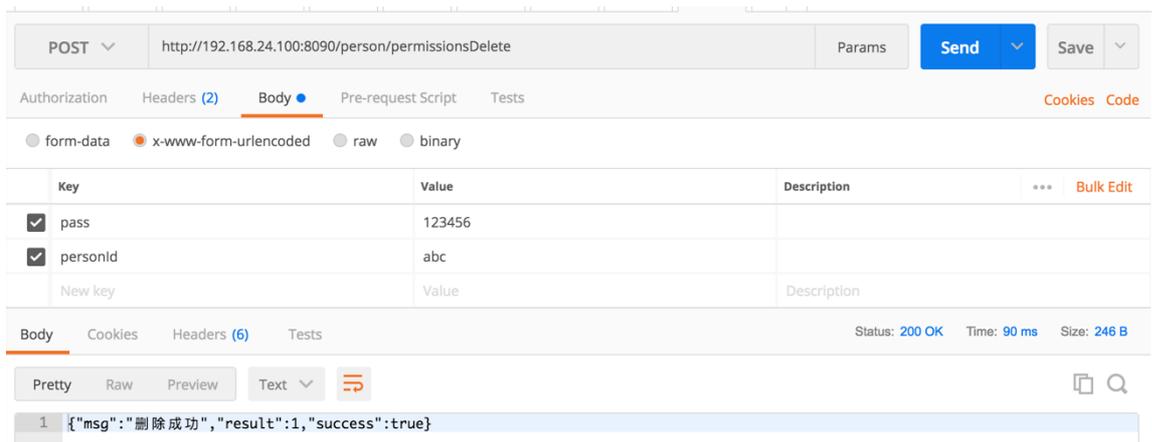
```
{  
  "msg": "设置成功",  
  "result": 1,//接口调通  
  "success": true//人员有效期限设置成功  
}
```

2.3.9 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"> 删除该人员的有效期限设置，该人员不会被定时删除 若该人员未设置过有效期限，则调用此接口会报错

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
}
```

2.4 照片管理类接口

2.4.1 照片注册 (base64)

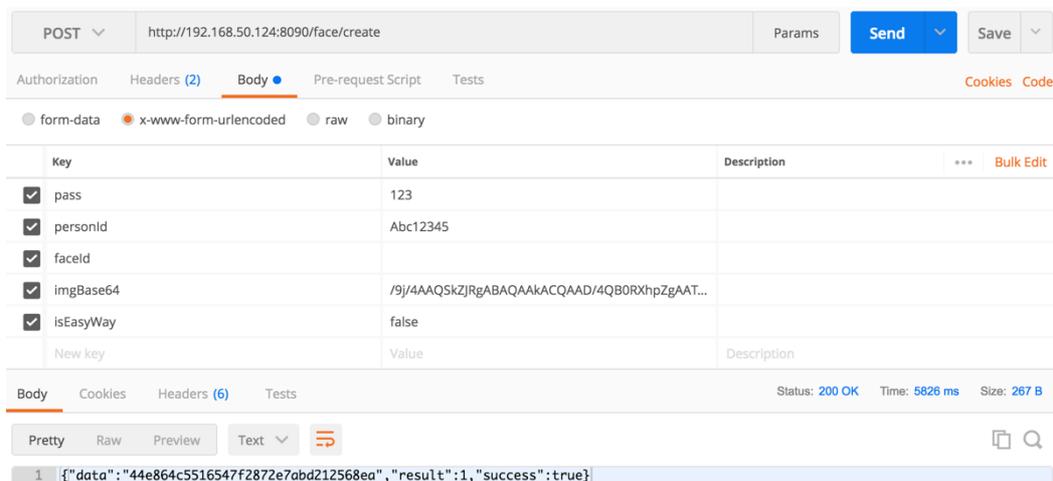
请求数据

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"> • 用于标识该照片属于某个人 id • 必须先注册人员，才能添加照片
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"> • 不加头部，如：data:image/jpg;base64, • 图片格式支持 png、jpg、jpeg
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



The screenshot shows a Postman request configuration for a POST method to the URL `http://192.168.50.124:8090/face/create`. The body is set to `x-www-form-urlencoded` with the following parameters:

Key	Value	Description
pass	123	
personId	Abc12345	
faceId		
imgBase64	/9j/4AAQSkZJRgABAQAAkACQAAD/4QB0RXhpZgAAT...	
isEasyWay	false	

The response status is 200 OK, with a time of 5826 ms and a size of 267 B. The response body is:

```
1 [{"data": "44e864c5516547f2872e7abd212568ea", "result": 1, "success": true}]
```

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  
}
```

2.4.2 照片注册 (url)

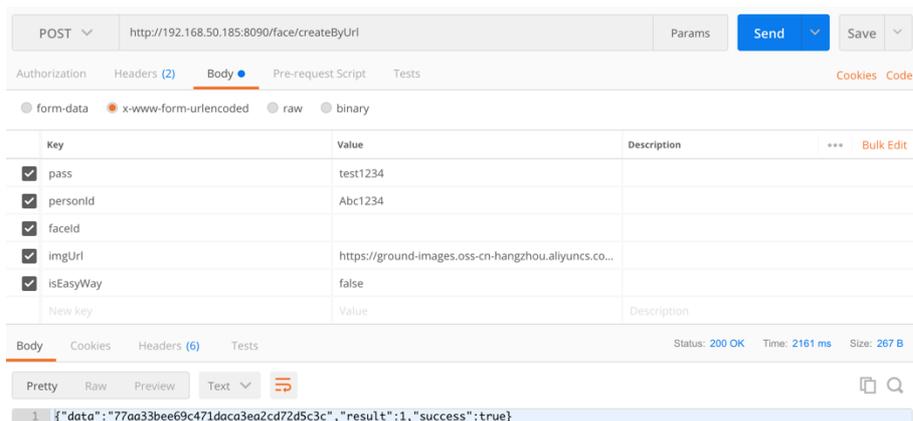
请求数据

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"> • 用于标识该照片属于某个人 id • 必须先注册人员，才能添加照片
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"> • 先通过 url 下载图片到本地，然后从图片中提取特征 • 只支持 http 协议存储位置的图片，不支持本地路径 • 图片格式支持 png、jpg、jpeg
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": "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  
}
```

2.4.3 照片注册 (File)

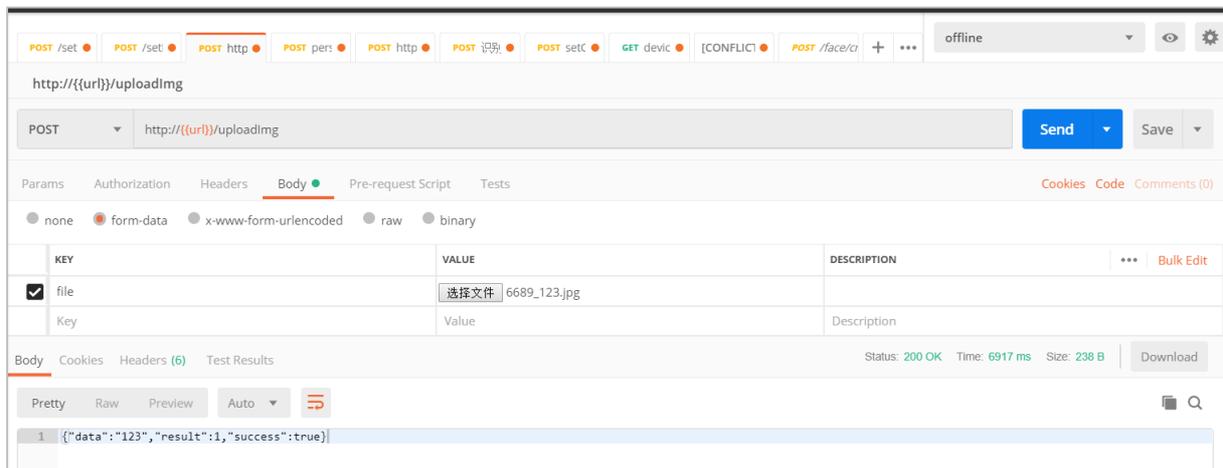
请求数据

Method	URL			
POST	http://设备 IP:8090/uploadImg			
参数名	描述	类型	必传	附加说明
file	照片文件	File	Y	照片文件命名规则： personId_faceId.jpg(如： a666_57a8.jpg)personId、faceId 限制是 0 至 32 位以内的数字字母组合。

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": "123", //照片 id，每张注册照都有一个 id，限制 32 位。
  "result": 1, //接口调通
  "success": true //照片注册成功
}
```

2.4.4 拍照注册

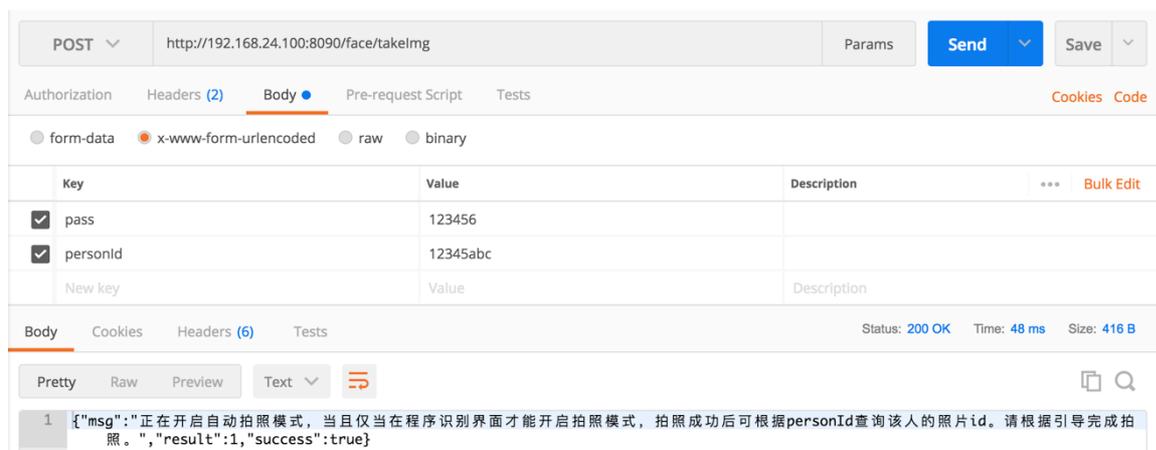
请求数据

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。

Method	URL			
POST	http://设备 IP:8090/face/takeImg			
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"> 给指定人员 id 拍摄照片 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

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



The screenshot shows a Postman request configuration for a POST method to the URL `http://192.168.24.100:8090/face/takeimg`. The request body is set to `x-www-form-urlencoded` and contains two parameters: `pass` with value `123456` and `personId` with value `12345abc`. The response status is `200 OK` with a time of `48 ms` and size of `416 B`. The response body is displayed in JSON format:

```

1 [{"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, "success": true}

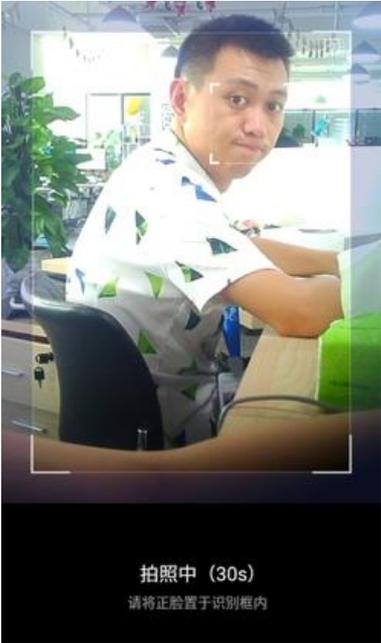
```

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

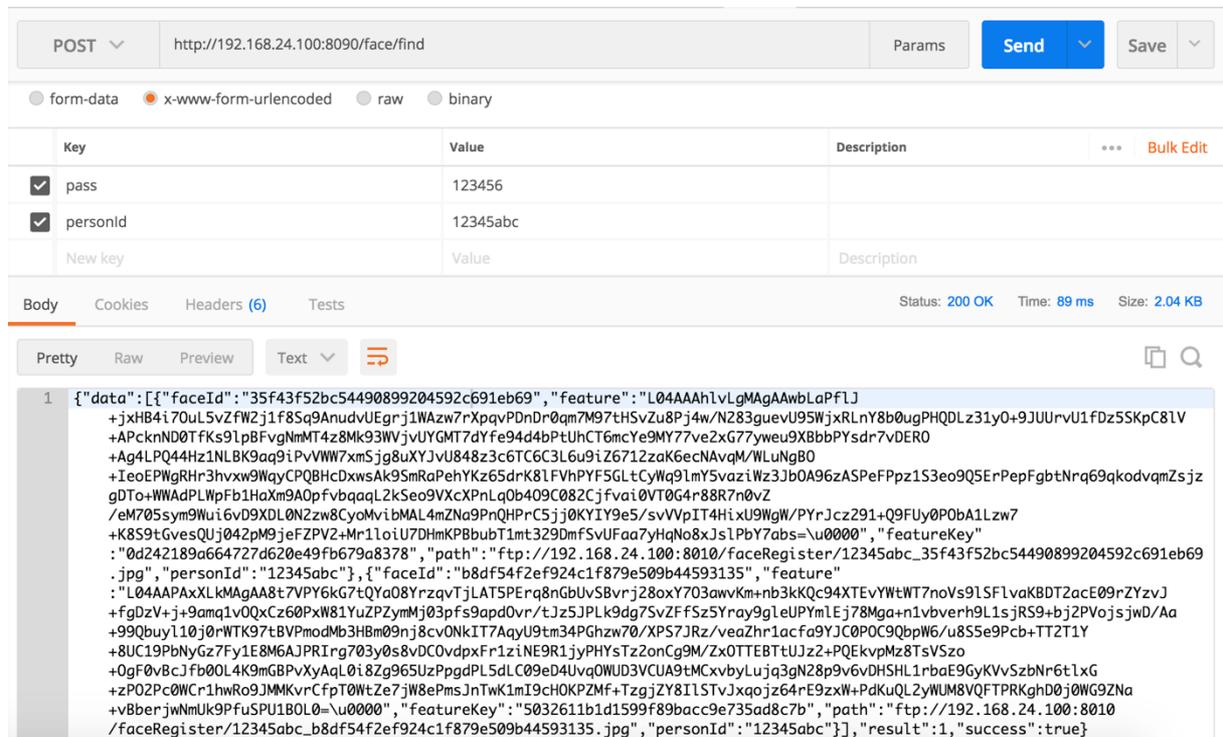
Uface-M5201	Uface-M7202、Uface-M7212
<p style="text-align: center;">设备拍照模式界面</p> 	<p style="text-align: center;">设备拍照模式界面</p> 

2.4.4 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



The screenshot shows a Postman request configuration for a POST method to the URL `http://192.168.24.100:8090/face/find`. The request body is x-www-form-urlencoded with the following parameters:

- `pass`: 123456
- `personId`: 12345abc

The response is a JSON object:

```

1 {"data": [{"faceId": "35f43f52bc54490899204592c691eb69", "feature": "L04AAAh1vLgMAgAAwbLaPflj
+ jxHB4i70uL5vZFW2j1f8S9AnudvUEgrj1WAzwrXpavPdNr0qm7M97tHsvZu8Pj4w/N283guevU95WjxRLnY8b0ugPHQDLz31y0+9JUUrVU1fDz5SKpC8lV
+ APcknND0TfKs9lPBFvgNmMT4z8Mk93WVjvUYGMT7dYfe94d4bPtUhcT6mcYe9MY77ve2xG77yweu9XBbbPYsdr7vDER0
+ Ag4LPQ44Hz1NLBK9aa9iPvVWW7xm5jg8uXYJvU848z3c6TC6C3L6u9iZ6712zaK6ecNAvaqM/WLuNgB0
+ IeoEPWgRHR3hvxw9WqyCPQBhcDxwsAk9SmRaPehYkZ65drK8lFVhPYF5GLtCyWq9lMYSvaziWz3Jb0A96zASPeFPpz1S3eo95ErPepFgbtNra69qkdvqmZsJz
gDT0+WWAdPLWpFb1HaXm9A0pFvbqaaL2kSeo9VXcXpNLaQb409C082Cjfvai0VT0G4r88R7n0vZ
/eM705sym9Wui6vD9XDL0N2zw8CyoMvibMAL4mZna9PnQHPrc5jj0KYIY9e5/svVpIT4HiXU9WgW/PYrJcz291+Q9Fuy0P0bA1Lzw7
+ K859tGvesQUj042pM9jefZPV2+Mr1loiU7DHmKPbubT1mt329DmfSvUfAa7yHqNo8xJs1PbY7abs=\u0000", "featureKey"
: "0d242189a664727d620e49fb679a8378", "path": "ftp://192.168.24.100:8010/faceRegister/12345abc_35f43f52bc54490899204592c691eb69
.jpg", "personId": "12345abc"}, {"faceId": "b8df54f2ef924c1f879e509b44593135", "feature"
: "L04AAPAxLkMAgAA8t7VPY6G7tQYao8YrzqvTjLAT5PEraq8nGbuV5Bvrj28oxY703awwKm+nb3kKQC94XTEvYwTWT7noVs9l5FlvaKBdT2acE09rZyVj
+ fgDzV+j+9amq1v0XcZ60PxW81YuZPZymMj03pfs9apd0vr/tJz5JPLk9dg7SvZFfS5Zyray9glEUPYmLEj78Mga+n1vbverh9L1sJRS9+bj2PV0jsjwD/Aa
+ 99Qbuyl10j0rWTK97tBVPmodMb3HBm09nj8cv0nkIT7AayU9tm34PGhz70/XPS7JRz/veaZhr1acfa9YJC0P0C9QbpW6/u8S5e9Pcb+TT2T1Y
+ 8UC19PbNyGz7Fy1E8M6AJPRirg703y0s8vDC0vdpxFr1ziNE9R1jyPHYsTzZonCg9M/Zx0TTEBTtUJz2+PQEkvpMz8TsvSzo
+ 0gF0vbcjfb0L4K9mGBpvXyAqL0i8Zg965UzPpgdPL5dLC09eD4UvaQWUD3VUA9tMCxvbyLuja3gN28p9v6vDHS11rbaE9GyKVvSzbNr6tlxG
+ zP02Pc0WCr1hwRo9JMMKvrCfpT0WtZe7jW8ePmsJnTwk1mI9cHOKPZMF+TzgjZY8l1StvJxqojz64rE9zxW+PdKuQL2yWUM8VQFTPRKghD0j0Wg9ZNa
+ vBberjwNmUk9PfuSPU1B0L0=\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": [{
    "facelId": "b8df54f2ef924c1f879e509b44593135", //照片 id
    "feature": "L04AAPAxXLkMAgAA8t7VPY6kG7tQYaO8YrzqvTjLAT5PErq8nGbUvSBvrj28oxY7O3awvKm+nb3kKQc94XTEvYWtWT7noVs9ISFlvaKBDT2acE09rZYzvJ+fgDzV+j+9amq1vOQxCz60PxW81YuZPZymMj03pfs9apdOvr/tJz5JPLk9dg7SvZFfSsz5Yray9gleUPYmIEj78Mga+n1vbverh9L1sjRS9+bj2PVojsjwD/Aa+99Qbuy10j0rWTK97tBVPmodMb3HBm09nj8cvONkIT7AqyU9tm34PGhzw70/XPS7JRz/veaZhr1acfa9YJC0POC9QbpW6/u8S5e9Pcb+TT2T1Y+8UC19PbNyGz7Fy1E8M6AJPRlrg703y0s8vDCOvdpxFr1ziNE9R1jyPHYsTz2onCg9M/ZxOTTEBTtUJz2+PQEkvpMz8TsVSzo+OgF0vBcJfb0OL4K9mGBPvXyAqL0i8Zg965UzPpgdPL5dLC09eD4UvqOWUD3VCUA9tMCxvbyLujq3gN28p9v6vDHSHL1rbaE9GyKVvSzbNr6tlxG+zPO2Pc0WCr1hwRo9JMMKvrCfpT0WtZe7jW8ePmsJnTwK1ml9cHOKPZMf+TzgjZY8IISTvJxqojz64rE9zxW+PdKuQL2yWUM8VQFTPRKghD0j0WG9ZNa+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
}
```

2.4.5 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"> 给指定人员 id 注册一个特征 人员 id 必须为已存在；若不存在该人员 id，则无法添加
faceId	照片 id	String	Y	<ul style="list-style-type: none"> 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 照片更新时，以 faceId 为唯一标识进行更新
feature	特征码	String	Y	<ul style="list-style-type: none"> 可由注册照片回调接收，也可由照片查询接口获得 注册一个特征相当注册于一张照片 每个人员 id 下，包括特征和照片，总数最多为 3 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
featureKey	特征密钥	String	Y	<ul style="list-style-type: none"> 特征注册成功后，由回调接收 特征有效性校验 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

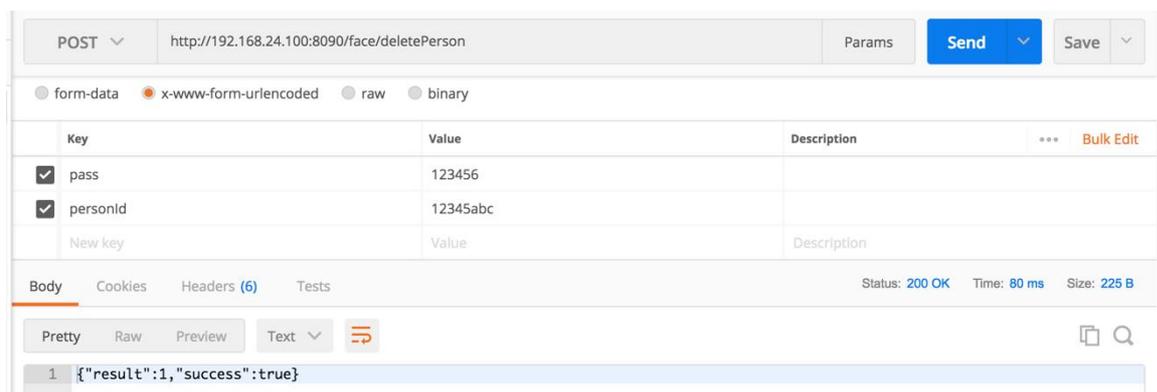
- i** 注意：a) 特征码注册等同于照片 base64 码注册，一个特征相当于一张注册照片。
- b) 特征注册比 base64 注册速度更快，适用于快速注册。
- c) 特征注册并非原图注册，因此无法查询到特征注册照片。若使用照片查询接口，特征 path 为空。
- d) 特征的所有处理方式（删、改、查等）与注册照片一致。
- e) 若设置了注册照片回调地址，特征注册成功后设备会将特征信息通过 POST 请求方式回调给预先设置的回调地址，[参见注册照片回调](#)。

2.4.8 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  
}
```

2.4.6 照片更新

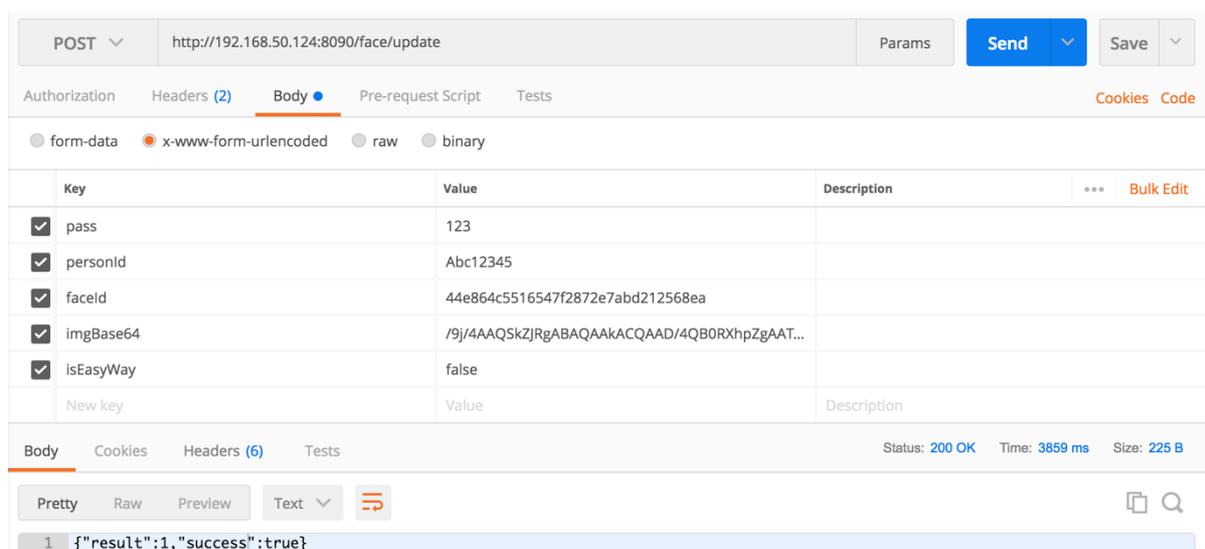
请求数据

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

i 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



POST http://192.168.50.124:8090/face/update

Authorization Headers (2) Body Pre-request Script Tests Cookies Code

form-data x-www-form-urlencoded raw binary

Key	Value	Description
pass	123	
personId	Abc12345	
faceId	44e864c5516547f2872e7abd212568ea	
imgBase64	/9j/4AAQSkZJRgABAQAAkACQAAD/4QB0RXhpZgAAT...	
isEasyWay	false	
New key	Value	Description

Body Cookies Headers (6) Tests Status: 200 OK Time: 3859 ms Size: 225 B

Pretty Raw Preview Text

```
1 {"result":1,"success":true}
```

返回示例

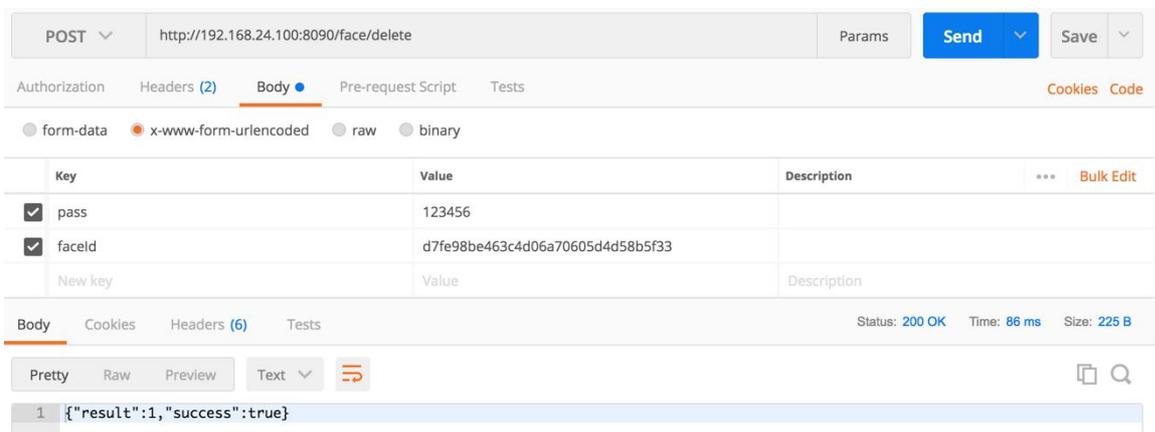
```
{  
  "result": 1, //接口调通  
  "success": true //照片更新成功  
}
```

2.4.7 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



The screenshot shows a Postman interface for a POST request. The URL is `http://192.168.24.100:8090/face/delete`. The request body is set to `x-www-form-urlencoded` and contains two parameters: `pass` with value `123456` and `faceId` with value `d7fe98be463c4d06a70605d4d58b5f33`. The response status is `200 OK` with a time of `86 ms` and size of `225 B`. The response body is `{\"result\":1,\"success\":true}`.

返回示例

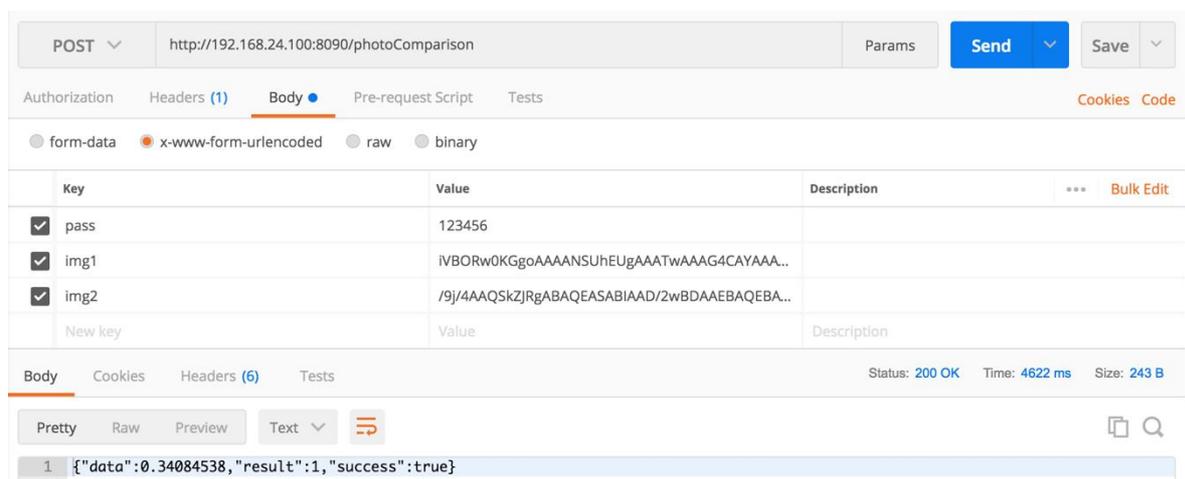
```
{
  "result": 1, //接口调通
  "success": true //照片删除成功
}
```

2.4.9 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	<ul style="list-style-type: none"> 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  
}
```

2.5 识别记录管理接口

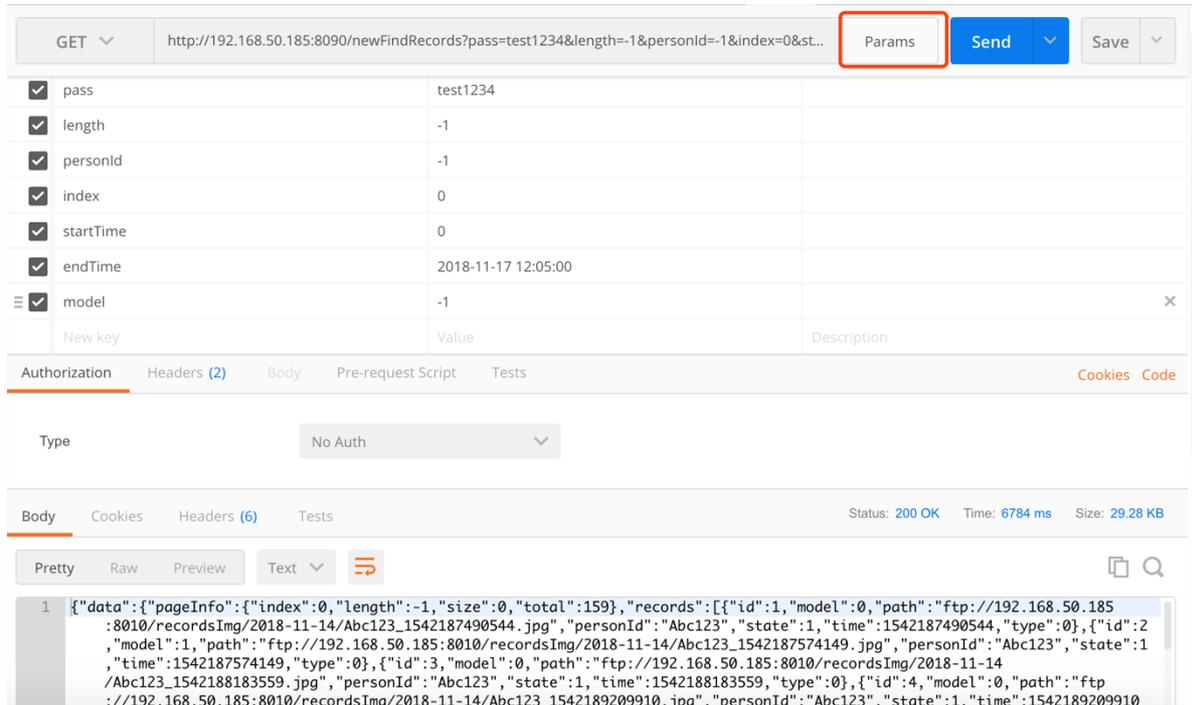
2.5.1 识别记录查询（推荐使用）

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"> • 查询指定 id 的人员识别记录 • 传入-1 可查询所有人员的识别记录，包括陌生人 • 传入 STRANGERBABY，可查询所有陌生人/识别失败记录 • 传入 IDCARD，可查询所有人证比对记录
length	每页最大数量	Int	Y	<ul style="list-style-type: none"> • 传入-1 为不分页 • 若不传-1，请务必大于 0
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"> • 若不按时间查询，请分别传入 0 • 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
endTime	记录结束时间	Date	Y	
model	识别记录类型	Int	Y	<ul style="list-style-type: none"> -1: 所有类型的识别记录 0: 刷脸识别记录 1: 卡&人脸双重认证识别记录 2: 人证比对识别记录 3: 刷卡识别记录
order	排序方式	Int	N	<ul style="list-style-type: none"> 1: 按时间升序排列； 默认：按时间降序排列

postman 示例

 注意：GET 请求参数放在 url 里，点击“Params”添加参数。



The screenshot shows a Postman request configuration for a GET request to `http://192.168.50.185:8090/newFindRecords?pass=test1234&length=-1&personId=-1&index=0&st...`. The parameters are:

Key	Value
pass	test1234
length	-1
personId	-1
index	0
startTime	0
endTime	2018-11-17 12:05:00
model	-1

The response body is a JSON object:

```

1 {"data":{"pageInfo":{"index":0,"length":-1,"size":0,"total":159},"records":[{"id":1,"model":0,"path":"ftp://192.168.50.185:8010/recordsImg/2018-11-14/Abc123_1542187490544.jpg","personId":"Abc123","state":1,"time":1542187490544,"type":0},{id":2,"model":1,"path":"ftp://192.168.50.185:8010/recordsImg/2018-11-14/Abc123_1542187574149.jpg","personId":"Abc123","state":1,"time":1542187574149,"type":0},{id":3,"model":0,"path":"ftp://192.168.50.185:8010/recordsImg/2018-11-14/Abc123_1542188183559.jpg","personId":"Abc123","state":1,"time":1542188183559,"type":0},{id":4,"model":0,"path":"ftp://192.168.50.185:8010/recordsIma/2018-11-14/Abc123_1542189209910.ipa","personId":"Abc123","state":1,"time":1542189209910}]}

```

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": 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：刷卡，1：卡&人脸双重认证，2：人证比对，3：刷卡
      "order": 1, //排序方式,1: 按时间降序排列；默认：按时间逆序排列
      "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/recordslmg/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\":"84E0F4200E4802FA\", \"id\":0,\"idNum\":\"330326199301010818\", \"issuingOrgan\":\"西湖区公安局\", \"name\":\"郝雨凡\", \"nation\":\"汉
      \", \"photoPath\":\"ftp://192.168.50.32:8010/recordslmg/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/recordslmg/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/recordslmg/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

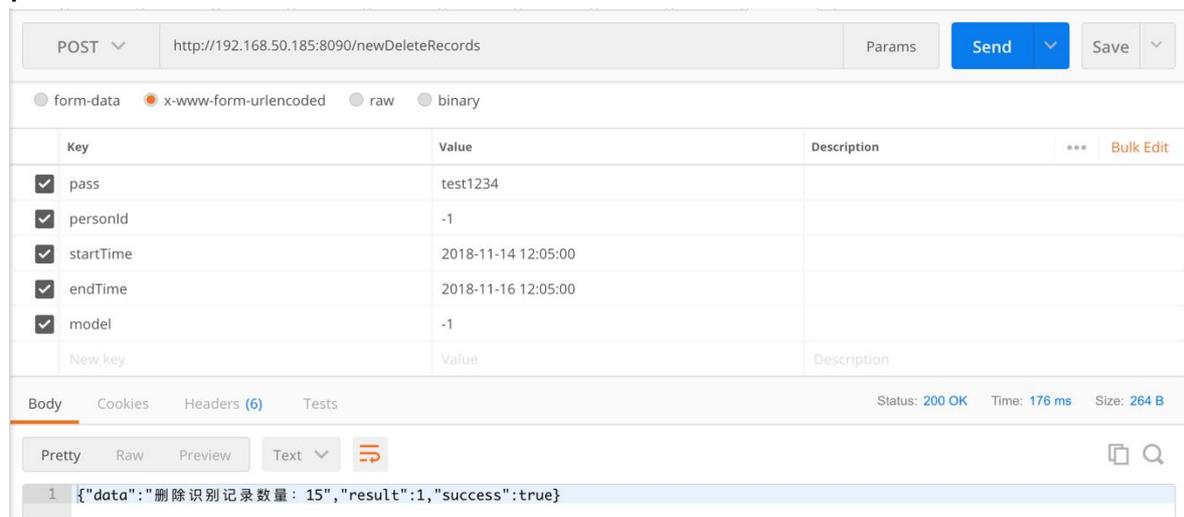
```

2.5.2 识别记录删除（推荐使用）

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"> 删除指定 id 的人员识别记录 传入-1 可删除所有人员的识别记录，包括陌生人 传入 STRANGERBABY，可删除所有陌生人/识别失败记录 传入 IDCARD，可删除所有人证比对的识别记录
startTime	记录起始时间	Date	Y	<ul style="list-style-type: none"> 删除时间段内所有人脸识别记录及现场照 传入时间格式为（年-月-日 时:分:秒）： 2017-07-15 12:05:00
endTime	记录结束时间	Date	Y	精确删除请用 unixtime
model	记录类型	Int	Y	-1: 所有类型的识别记录 0: 刷脸识别记录 1: 卡&人脸双重认证识别记录 2: 人证比对识别记录 3: 刷卡识别记录其他为非法值

postman 示例



POST http://192.168.50.185:8090/newDeleteRecords

Parameters:

Key	Value	Description
pass	test1234	
personId	-1	
startTime	2018-11-14 12:05:00	
endTime	2018-11-16 12:05:00	
model	-1	

Response:

```
1 [{"data": "删除识别记录数量: 15", "result": 1, "success": true}]
```

返回示例

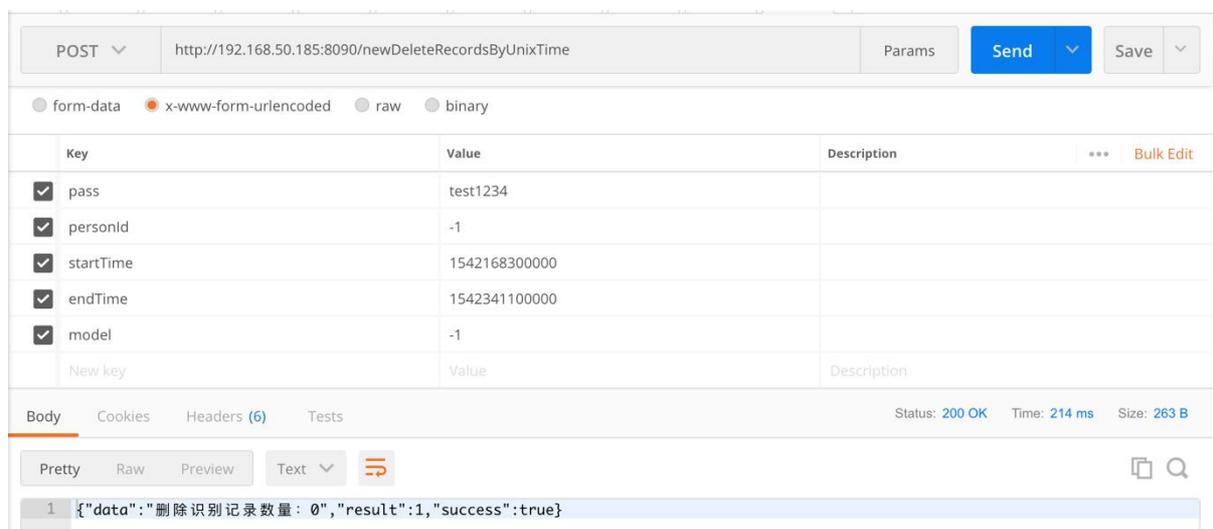
```
{  
  "data": "删除识别记录数量: 1",//删除的识别记录数量  
  "result": 1,//接口调通  
  "success": true//成功删除识别记录  
}
```

2.5.3 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"> 删除指定 id 的人员识别记录 传入-1 可删除所有人员的识别记录，包括陌生人 传入 STRANGERBABY，可删除所有陌生人/识别失败记录 传入 IDCARD，可删除所有人证比对的识别记录
startTime	记录起始时间	Date	Y	<ul style="list-style-type: none"> 删除时间段内所有人脸识别记录及现场照 传入时间格式为：1507773661000
endTime	记录结束时间	Date	Y	
model	记录类型	Int	Y	-1: 所有类型的识别记录 0: 刷脸识别记录 1: 卡&人脸双重认证识别记录 2: 人证比对识别记录 3: 刷卡识别记录其他为非法值

postman 示例



POST http://192.168.50.185:8090/newDeleteRecordsByUnixTime

Params Send Save

form-data x-www-form-urlencoded raw binary

Key	Value	Description
pass	test1234	
personId	-1	
startTime	1542168300000	
endTime	1542341100000	
model	-1	
New key	Value	Description

Body Cookies Headers (6) Tests Status: 200 OK Time: 214 ms Size: 263 B

Pretty Raw Preview Text

```
1 [{"data": "删除识别记录数量: 0", "result": 1, "success": true}]
```

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  
}
```

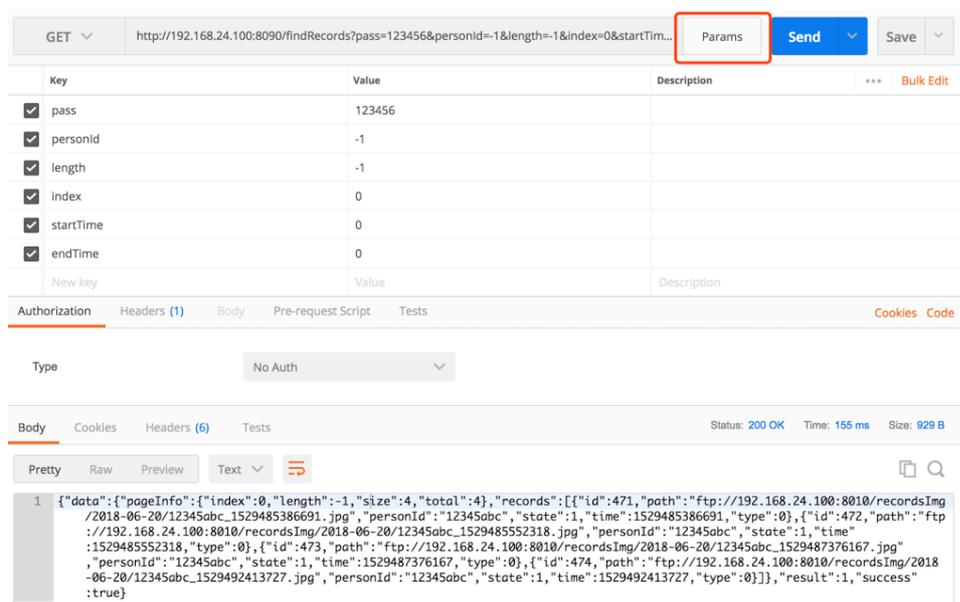
2.5.4 刷脸记录查询

请求数据

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"> 查询指定 id 的人员识别记录 传入-1 可查询所有人员的识别记录，包括陌生人 传入 STRANGERBABY，可查询所有陌生人/识别失败记录
length	每页最大数量	Int	Y	<ul style="list-style-type: none"> 传入-1 为不分页 若不传-1，请务必大于 0
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"> 若不按时间查询，请分别传入 0 若需要按时间查询，请按照如下格式（年-月-日 时:分:秒）：
endTime	记录结束时间	Date	Y	<ul style="list-style-type: none"> 2017-07-15 12:05:00

postman 示例

 注意：GET 请求参数放在 url 里，点击“Params”添加参数。



The screenshot shows a Postman interface for a GET request. The URL is `http://192.168.24.100:8090/findRecords?pass=123456&personid=-1&length=-1&index=0&startTime=0&endTime=0`. The 'Params' tab is selected, displaying a table of parameters:

Key	Value	Description
pass	123456	
personid	-1	
length	-1	
index	0	
startTime	0	
endTime	0	

The 'Body' tab shows a JSON response:

```

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_152948552318.jpg", "personId": "12345abc", "state": 1, "time": 152948552318, "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/recordslmg/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/recordslmg/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/recordslmg/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
}

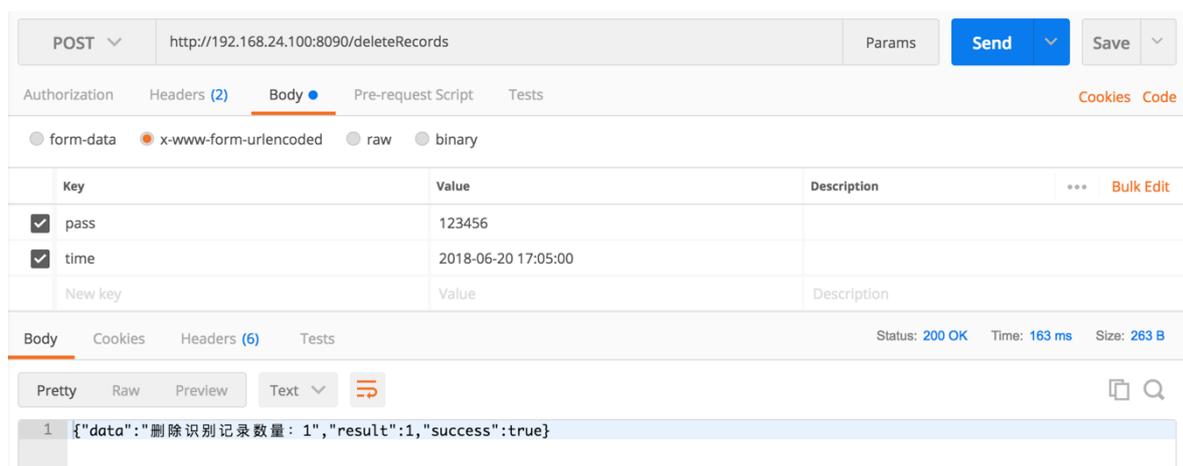
```


2.5.5 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"> 删除所传日期时间点前的所有人脸识别记录及现场照 传入时间格式为（年-月-日 时:分:秒）： 2017-07-15 12:05:00

请求数据

postman 示例



The screenshot shows a Postman interface for a POST request to `http://192.168.24.100:8090/deleteRecords`. The request body is set to `x-www-form-urlencoded`. The parameters are:

Key	Value	Description
pass	123456	
time	2018-06-20 17:05:00	

The response is shown in the bottom panel, displaying a JSON object:

```
1 {"data": "删除识别记录数量: 1", "result": 1, "success": true}
```

返回示例

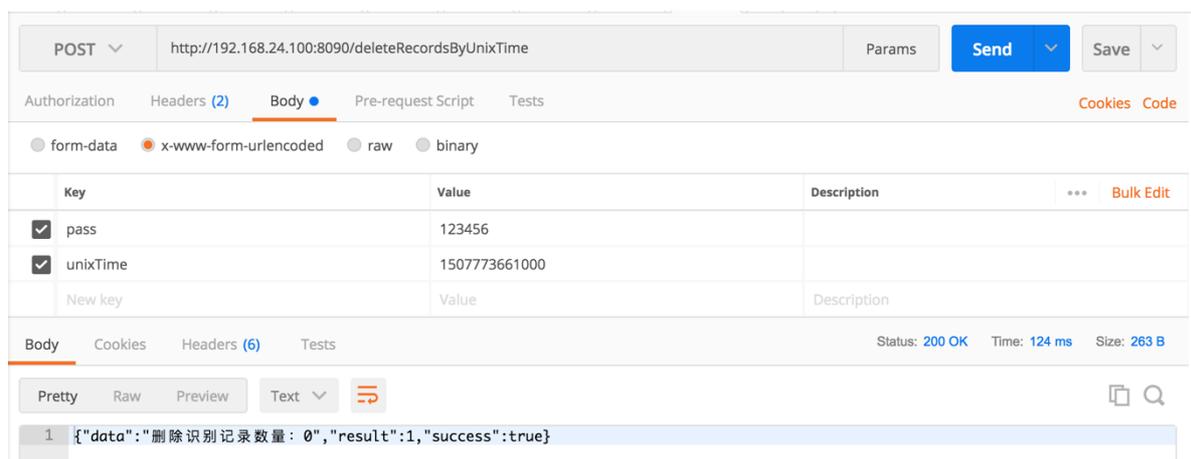
```
{
  "data": "删除识别记录数量: 1", //删除的识别记录数量
  "result": 1, //接口调通
  "success": true //成功删除刷脸识别记录
}
```

2.5.6 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"> 删除所传参数时间戳之前的所有人脸识别记录及现场照 传入时间格式为：1507773661000

postman 示例



POST http://192.168.24.100:8090/deleteRecordsByUnixTime

Authorization Headers (2) Body Pre-request Script Tests Cookies Code

form-data x-www-form-urlencoded raw binary

Key	Value	Description
pass	123456	
unixTime	1507773661000	
New key	Value	Description

Body Cookies Headers (6) Tests Status: 200 OK Time: 124 ms Size: 263 B

Pretty Raw Preview Text

```
1 {"data": "删除识别记录数量：0", "result": 1, "success": true}
```

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
}
```

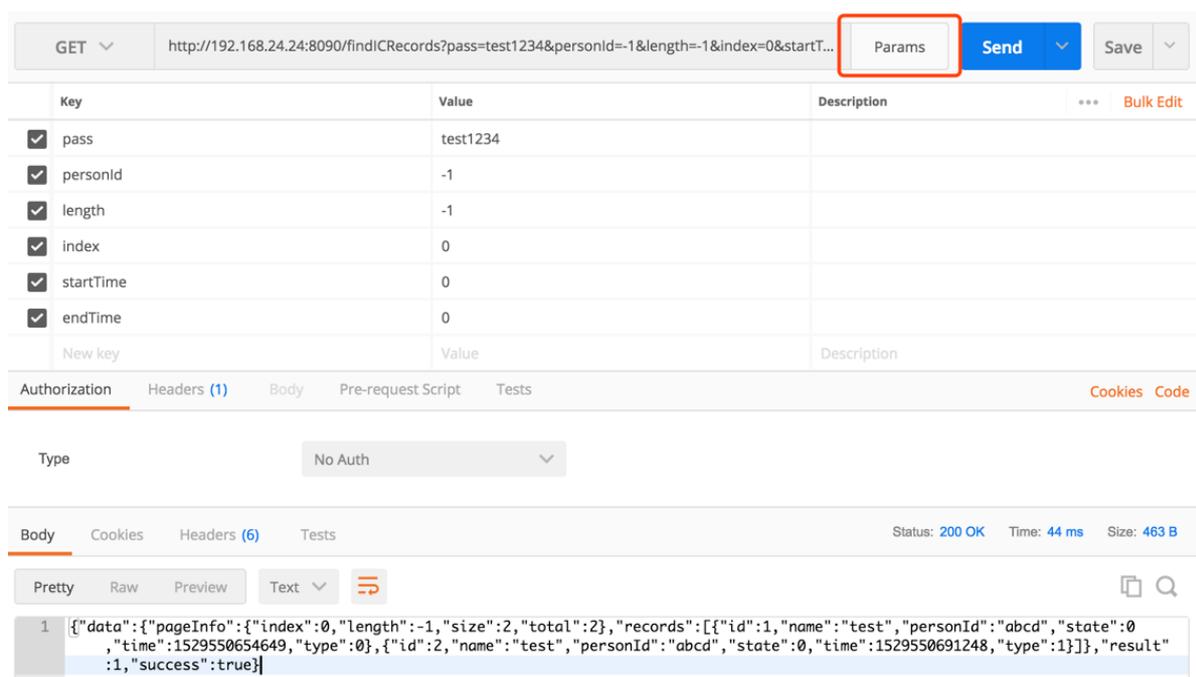
2.5.7 刷卡记录查询

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"> • 查询指定 id 的人员刷卡记录 • 传入-1 可查询所有人员的刷卡记录 • 识别失败的卡片不会生成刷卡记录
length	每页最大数量	Int	Y	<ul style="list-style-type: none"> • 传入-1 为不分页 • 若不传-1, 请务必大于 0
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"> • 若不按时间查询, 请分别传入 0 • 若需要按时间查询, 请按照如下格式 (年-月-日 时:分:秒):
endTime	记录结束时间	Date	Y	2017-07-15 12:05:00

postman 示例

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



The screenshot shows a Postman interface for a GET request. The URL is `http://192.168.24.24:8090/findICRecords?pass=test1234&personId=-1&length=-1&index=0&startT...`. The 'Params' tab is active, showing a list of parameters: pass (test1234), personId (-1), length (-1), index (0), startTime (0), and endTime (0). The 'Body' tab is also visible, showing the response in JSON format:

```
{ "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

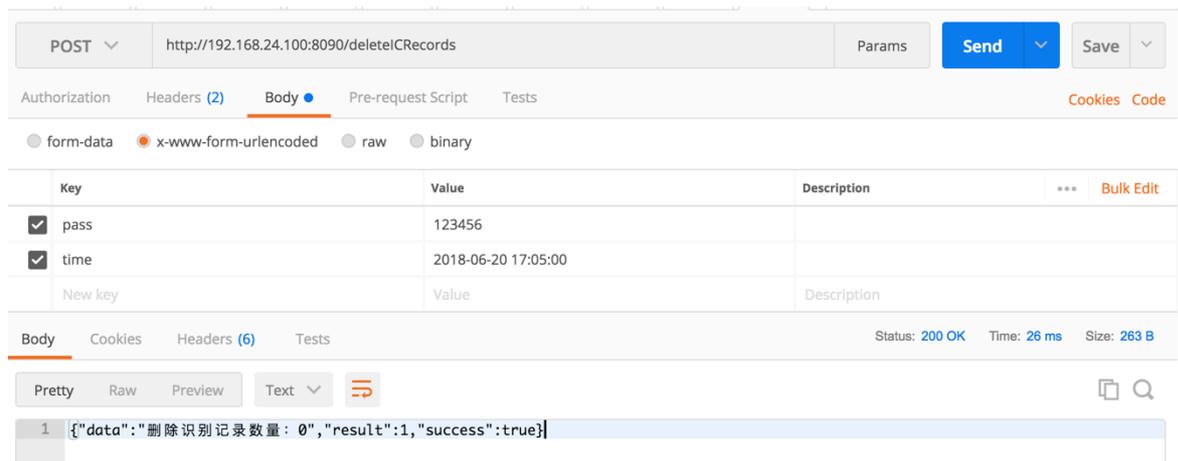
```
{
  "data": {
    "pageInfo": {
      "index": 0, //页码。当前数据所在页码
      "length": -1, //每页最大数量。-1 表示不分页，所有数据均显示在第一页(index=0)
      "size": 0, //总页数。若 length=-1，size 默认为 0；若 length>0，size≥1
      "total": 2 //总数据条数
    },
    "records": [{
      "id": 1, //识别记录 id，由设备自动生成并管理。用户无须使用
      "name": "test", //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
}
```

2.5.8 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"> 删除所传日期时间点前的所有刷卡记录 传入时间格式为（年-月-日 时:分:秒）： 2017-07-15 12:05:00

postman 示例



The screenshot shows a Postman interface for a POST request to `http://192.168.24.100:8090/deleteICRecords`. The request body is configured as `x-www-form-urlencoded` with the following parameters:

Key	Value	Description
pass	123456	
time	2018-06-20 17:05:00	

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

```
1 [{"data": "删除识别记录数量: 0", "result": 1, "success": true}]
```

返回示例

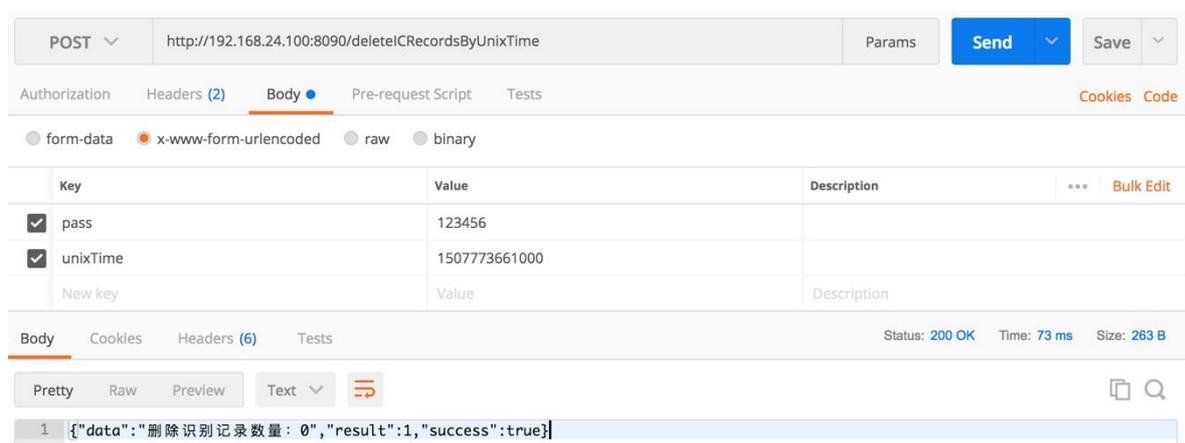
```
{
  "data": "删除识别记录数量: 0", //删除的识别记录数量
  "result": 1, //接口调通
  "success": true //成功删除刷卡识别记录
}
```

2.5.9 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"> 删除所传参数时间戳之前的所有人脸识别记录及现场照 传入时间格式为：1507773661000

postman 示例



The screenshot shows a Postman interface for a POST request to `http://192.168.24.100:8090/deleteICRecordsByUnixTime`. The request body is configured as `x-www-form-urlencoded` with two parameters: `pass` (value: 123456) and `unixTime` (value: 1507773661000). The response status is `200 OK` with a time of `73 ms` and size of `263 B`. The response body is displayed in JSON format:

```
1 [{"data": "删除识别记录数量: 0", "result": 1, "success": true}]
```

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
}
```

第三章 注意事项

尝试重启、重置

设备长时间停顿在某个界面时,用户可以通过插拔电源开关,将其上电复位并重新启动。若无效,则可以长按 RESET 小孔 8-12 秒重置设备。另外,可及时联系本公司人员反馈问题。

防尘、防水

本设备防护等级为 IP42,能防止直径或厚度大于 1.0mm 的工具、电线及类似的小型外物侵入而接触到电器内部的零件,滴水不会影响设备使用。

防摔

请避免跌落、砸摔、弯曲、重压等操作。

清洁

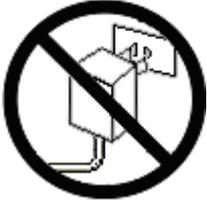
请以软布等材料轻轻擦拭显示屏及面板,避免使用水洗和清洁剂擦洗。

低温环境

本设备正常工作环境温度范围为-10~60°C(Uface-M5201 系列建议 55°C),若温度低于此范围,设备可能无法开机。



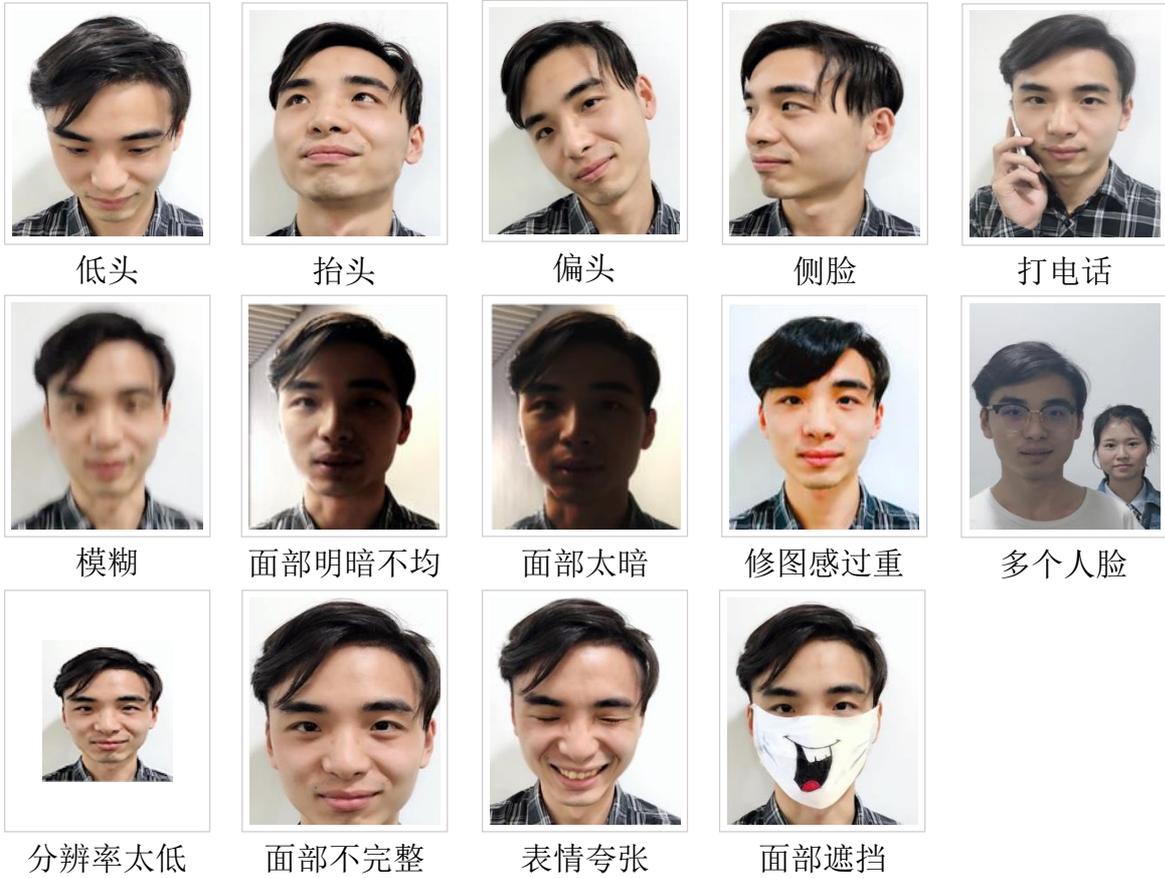
切勿用油或尖锐的物品污损显示屏幕



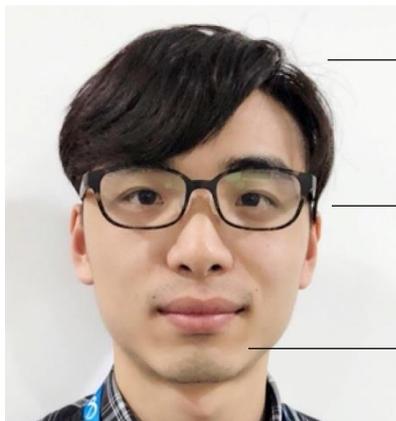
请勿使用不明电源适配器，以免烧坏设备

附录 1 注册照片规范说明

可能导致无法识别的照片



注册照片示例（男生版）



—— 简单背景

- 正脸
- 表情正常
- 镜片无反光
- 面部光照强度适中
- 面部明暗均匀

照片要求：

小于 400K

面部区域像素不低于 128×128

人脸大小占整张照片 1/3 以上

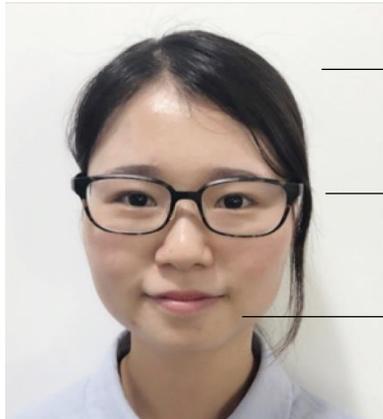
Save time, Enjoy life

增加注册照



根据设备所处场景中人脸的常见形态，增加注册照，可以提高识别体验。注册照最多 3 张。

注册照片示例（女生版）



—— 简单背景

- 正脸
- 表情正常
- 镜片无反光
- 面部光照强度适中
- 面部明暗均匀

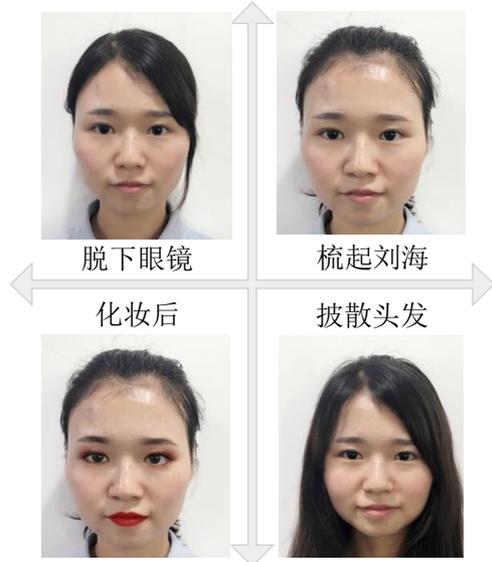
照片要求：

小于 400K

面部区域像素不低于 128×128

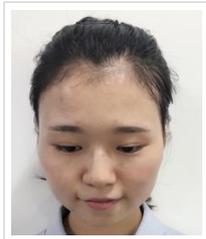
人脸大小占整张照片 1/3 以上

增加注册照



根据设备所处场景中人脸的常见形态，增加注册照，可以提高识别体验。注册照最多 3 张。

可能导致无法识别的照片



低头



抬头



偏头



侧脸



打电话



模糊



面部明暗不均



面部太暗



修图感过重



多个人脸



分辨率太低



面部不完整



表情夸张



面部遮挡



低头、抬头、人脸角度、表情夸张、打电话，会影响脸型轮廓和五官形态。如：照片中正脸和抬头，人眼会误判为不是同一个人，机器也会这样判断。



修图会影响脸型轮廓和五官形态，滤镜会消除脸部纹理，PS 后的图片，人眼会认为是不同的人，机器也会这样认为。



模糊、分辨率太低，会影响脸部特征的提取，可以从人眼的角度理解为“看不清长相”，描述不准确。



面部明暗不均、太暗，一方面面部太暗会影响脸部特征提取，另一方面从人眼的角度看，这个人看起来就是半边白半边黑，那个人肤色就很均匀，机器也会认为这两个是不同的人。



多人脸、面部不完整、面部有遮挡，则无法成功提取整张脸的特征，注册照片相当于无效。图中面部不完整的人缺少下巴，无法提取到下巴部分的特征点。

根据人员常用的识别状态，增加注册照可以提高识别率，提升识别体验。

- 佩戴眼镜的人，鉴于不方便佩戴眼镜的情况，可以增加一张不戴眼镜的注册照。
- 发型有刘海的人，鉴于发型会变动，可以增加一张无刘海的注册照。
- 平时化妆的人，鉴于妆容变动，可以增加一张未化妆的注册照。注意：化妆前后变化较大，有时人眼都无法分辨，机器也会无法分辨。
- 健身房、体育课等出汗较多的特殊场景，注册照可以增加一张运动完的状态。注意：运动完后妆容会有变化，热胀冷缩对面部及五官也有一定影响。

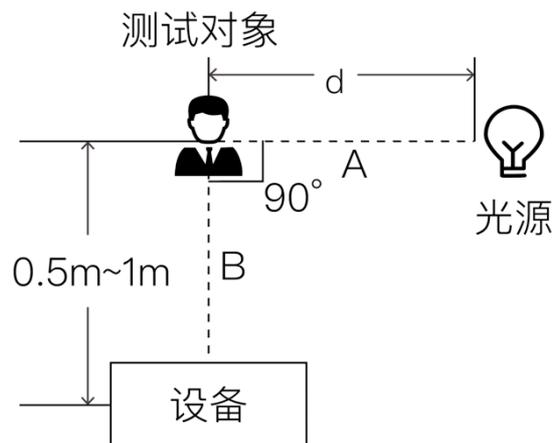
附录 2 安装环境要求

2.1 面部光照强度要求

设备识别到的人脸，面部光照强度（简称照度）应不低于 30 Lux。设备自带补光灯，当设备处于全黑环境时，距离设备 0.5 米处的人脸光照强度约为 100 Lux（Uface-M5201 系列）、300 Lux（Uface-M72XX 系列），距离设备越近，面部光照强度越高。若面部曝光度过高，则会导致特征点无法提取，识别较难。

光照强度测试

面部亮度测试标准如图所示：虚线 A 与虚线 B 的夹角为 90° ，采用全光谱普通照度计测量，光源采用 U30 标准光源，调整光源与测试对象的距离，以此时测试对象脸部光照强度为准。



面部光照强度参考



30Lux, 偏暗

50Lux, 正常

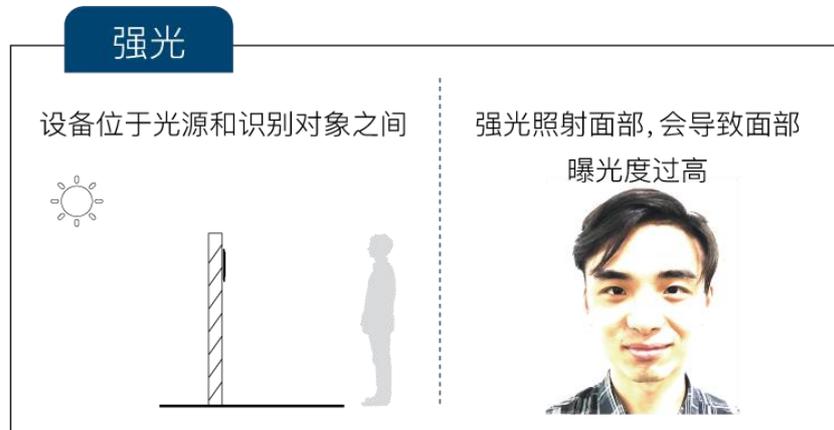
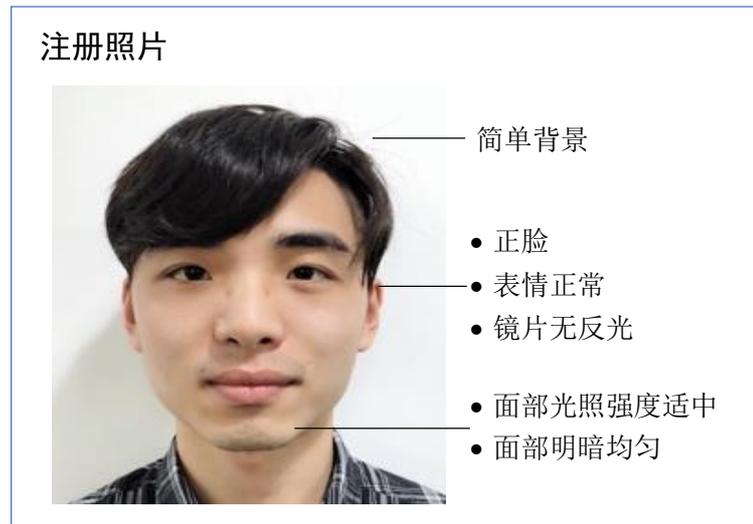
100Lux, 明亮

300Lux, 微曝光

500Lux, 高曝光

识别现场光线对面部的影响

根据 Uface 算法原理，识别现场抓拍照片与注册照片进行比对。若注册照十分规范，但识别现场光线影响较大，会导致现场照片与注册照对比差异过大。逆光（人脸面部太暗）、侧光（面部光线不均匀）、强光（脸部曝光度过高）都会影响识别体验。





从人眼的角度理解，面部太黑看不清，或过了一个夏天晒黑了，对人脸身份就会出现误判；曝光区域无法看清，无论曝光区域大小，都会影响身份判断。

改善方法：

- 1、将设备安装在光源影响较小的地方
- 2、根据现场情况增加一张注册照片，建议使用 Uface 设备录入照片

i 注意：一般情况下，虽然设备识别效果不好，但人员还是能识别出来的。实际操作的时候，也可通过找到设备上那个人识别出来的时候的现场抓拍照片，挑选质量较好的新增为注册照。

- 3、根据使用情况，适当降低识别阈值。

2.2 设备安装位置建议

室内安装，参考以下 4 张图示。

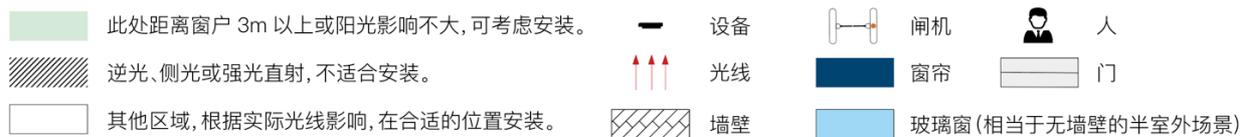
半室外安装，面向室外的区域参考图示玻璃墙、玻璃窗。

室外安装无任何遮光物时，建议在设备上方安装遮光罩（也可遮雨）。

i 注意：若在室内安装，设备距离日光灯等光源至少 2 米，距离透光窗户至少 3 米。

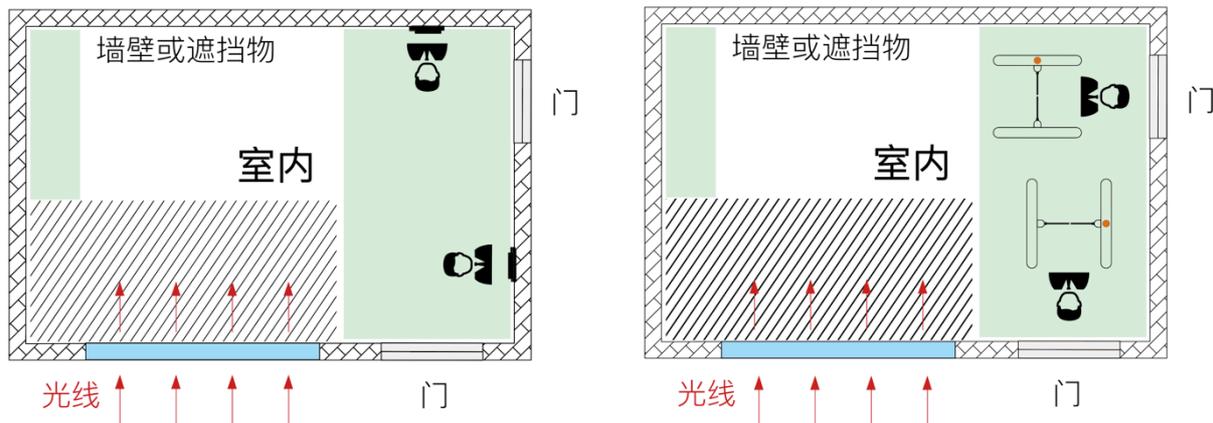
阳光/强光环境安装要求

Uface 人脸识别设备可安装于室内、室外环境，但阳光直射和强光会对现场抓拍的人脸图像造成较大影响。进行识别时，设备和人脸都不宜被强光照照射影响；侧光会导致人脸明暗不均，同样会影响识别。以下例图为室内/半室内安装地点参考，室外安装应给设备安装遮光板。若在室内安装，设备距离光源至少 2 米，距离窗户至少 3 米。



例图 1：阳光照进室内

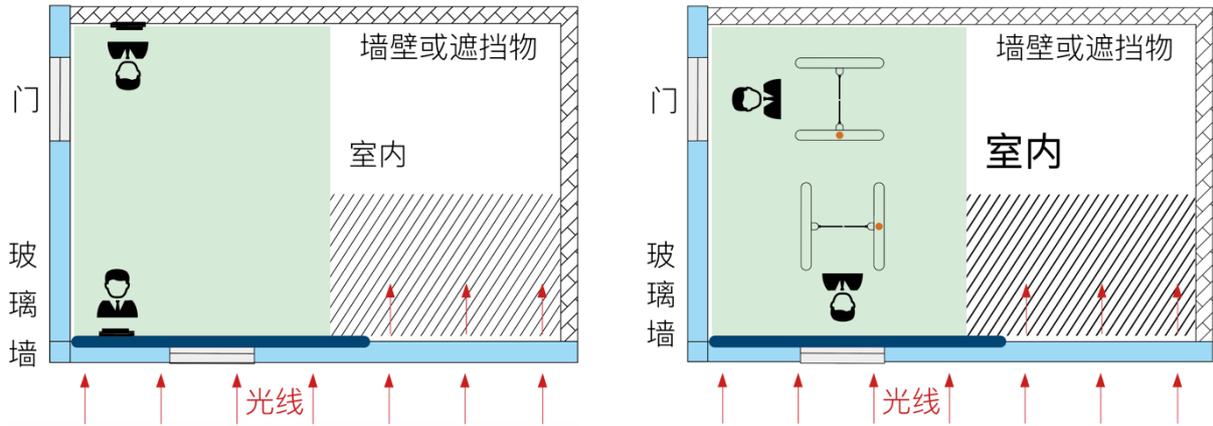
当有太阳光线从窗口照射进室内，应避免将人脸设备终端安装在如图黑色阴影位置，尽量安装在太阳光线影响较小、没有侧光和背光的位置。玻璃窗户相当于是无墙壁。



例图 2：房间墙壁是玻璃墙

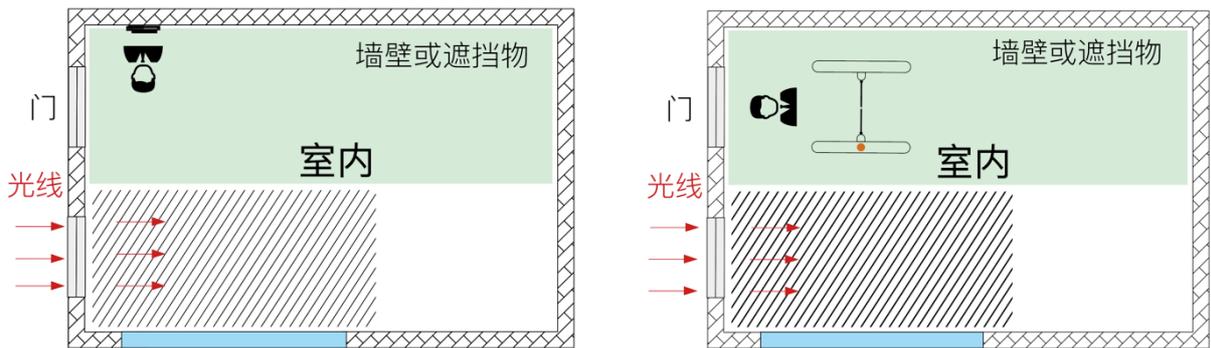
如室内装修为玻璃墙，太阳光线无遮挡时，建议给玻璃窗挂上窗帘，在窗帘遮挡处

安装人脸设备终端，以免太阳光线对识别造成影响。此场景中玻璃墙相当于无墙壁。



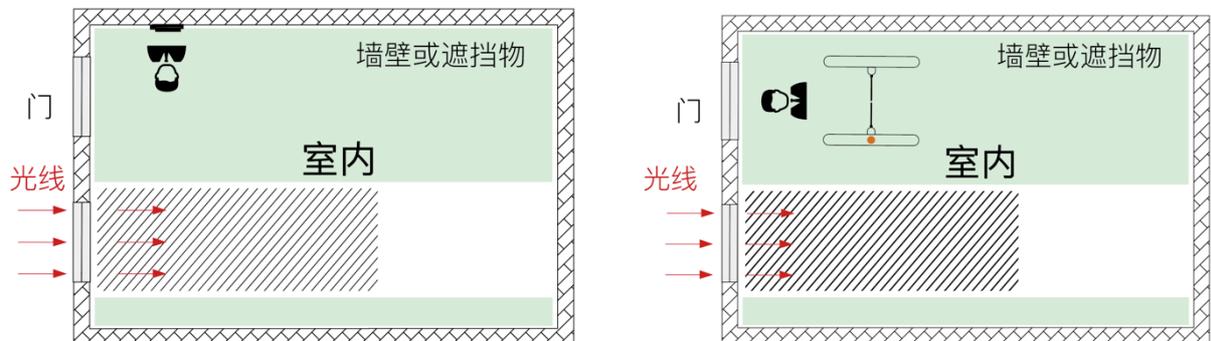
例图 3：阳光从房门照进且一侧有玻璃窗

阳光从房门照进，设备应安装于有遮挡物、光线影响较小的位置。玻璃窗户相当于无遮挡，设备应安装于没有玻璃窗的一侧或挂上窗帘。



例图 4：阳光从房门照进

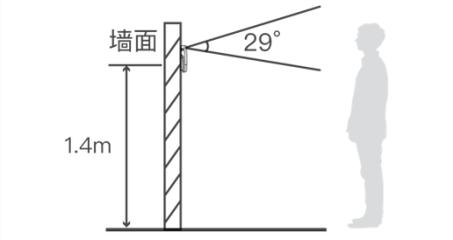
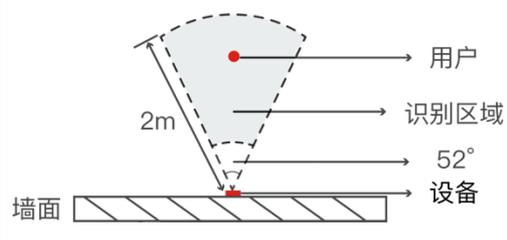
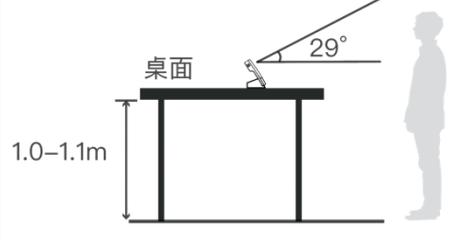
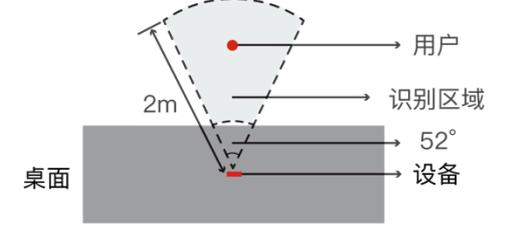
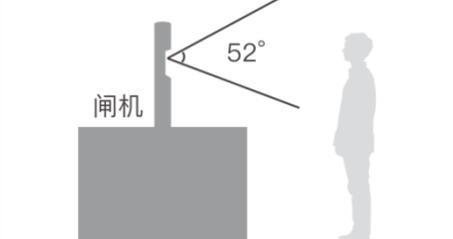
阳光从房门照进，设备应安装在有遮挡物、光线影响较小的位置。



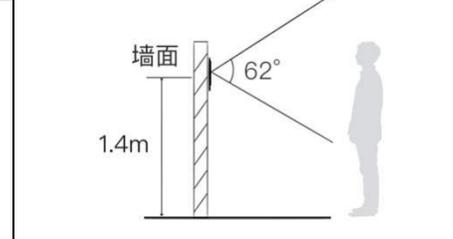
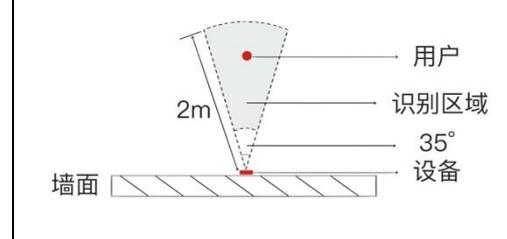
附录 3 安装/摆放高度

当设备放置在一定高度后,可识别的人员身高范围受限制于摄像头广角。Uface-5201 系列、Uface-M72XX 设备采用的摄像头型号不同,因此广角不同;Uface-M5201 系列设备可横竖屏切换,摄像头垂直广角、水平广角互换。

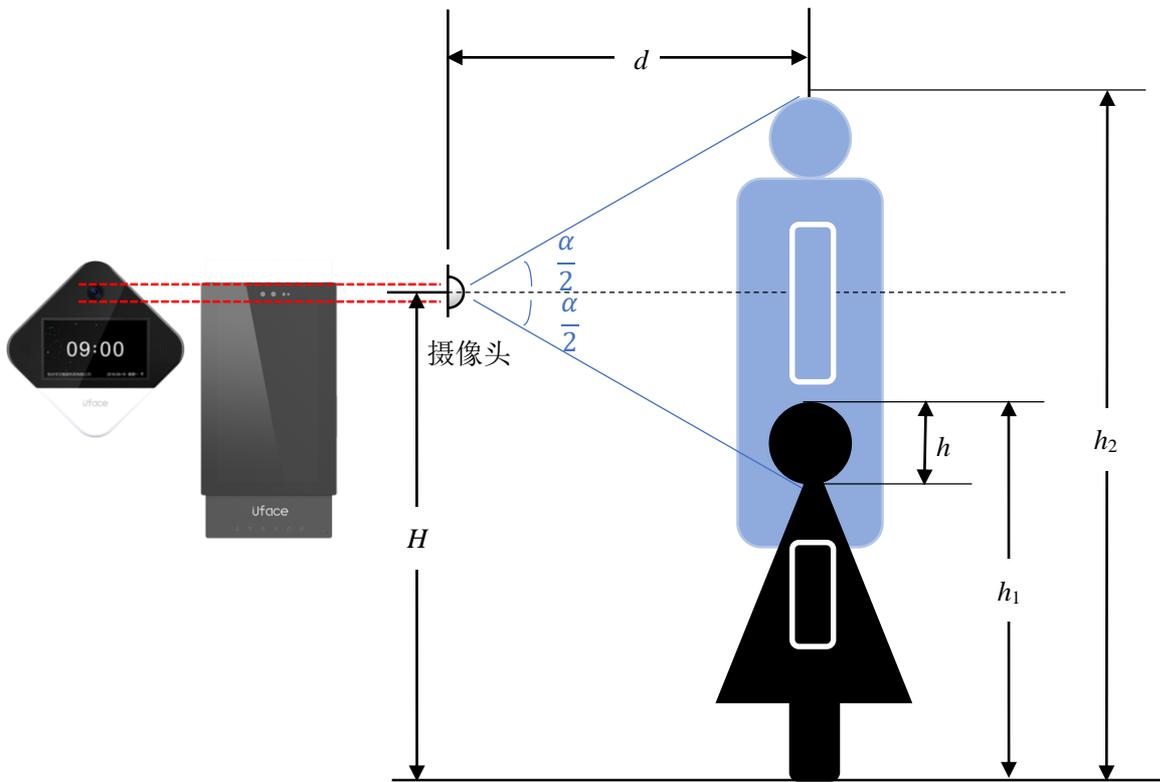
Uface-M5201 系列设备

M5201	摄像头垂直广角	摄像头水平广角
壁挂式 (横屏) 		
桌面式 (横屏) 		
闸机头 (竖屏) 		

Uface-M72XX 设备

M7202	摄像头垂直广角	摄像头水平广角
壁挂式 (竖屏) 		

以垂直摆放为例，计算识别身高范围



H ——摄像头距离地面的高度

d ——人脸到摄像头的水平距离

h_1 ——摄像头捕捉完整人脸的最低身高

h_2 ——摄像头捕捉完整人脸的最高身高

h ——人脸长度，一般取 0.2m

α ——摄像头垂直方向广角，Uace-M7202 设备 $\alpha=62^\circ$

计算示例：

设备距离地面 1.4m 时，计算人员距离设备 0.5m 时，可捕捉完整人脸的身高范围。

设备距离地面高度 $H=1.4\text{m}$

人员距离设备距离 $d=0.5\text{m}$

人脸长度 $h=0.2\text{m}$

$$h_2 = H + d \tan \frac{\alpha}{2} = H + d \tan \frac{62}{2} = 1.4 + 0.5 \times 0.6 = 1.70\text{m}$$

$$h_1 = H - d \tan \frac{\alpha}{2} + h = H - d \tan \frac{62}{2} + h = 1.4 - 0.5 \times 0.6 + 0.2 = 1.30\text{m}$$

角度计算参考

设备型号	安装方式	垂直广角	角度计算	[最低身高, 最高身高]
M5201	壁挂式（横屏）	$\alpha=29^\circ$	$\tan \frac{29}{2}=0.26$	$[H-0.26+h, H+0.26]$
	桌面式（横屏）	$\alpha=29^\circ$	$\tan 29=0.55$	$[H+h, H+0.55]$
	闸机头（竖屏）	$\alpha=52^\circ$	$\tan \frac{52}{2}=0.49$	$[H-0.26+h, H+0.49]$
M7202	壁挂式（竖屏）	$\alpha=67^\circ$ (已停产)	$\tan \frac{67}{2}=0.66$	$[H-0.66+h, H+0.66]$
	壁挂式（竖屏）	$\alpha=62^\circ$	$\tan \frac{62}{2}=0.6$	$[H-0.6+h, H+0.6]$

附录 4 网络连接说明

4.1 网络要求

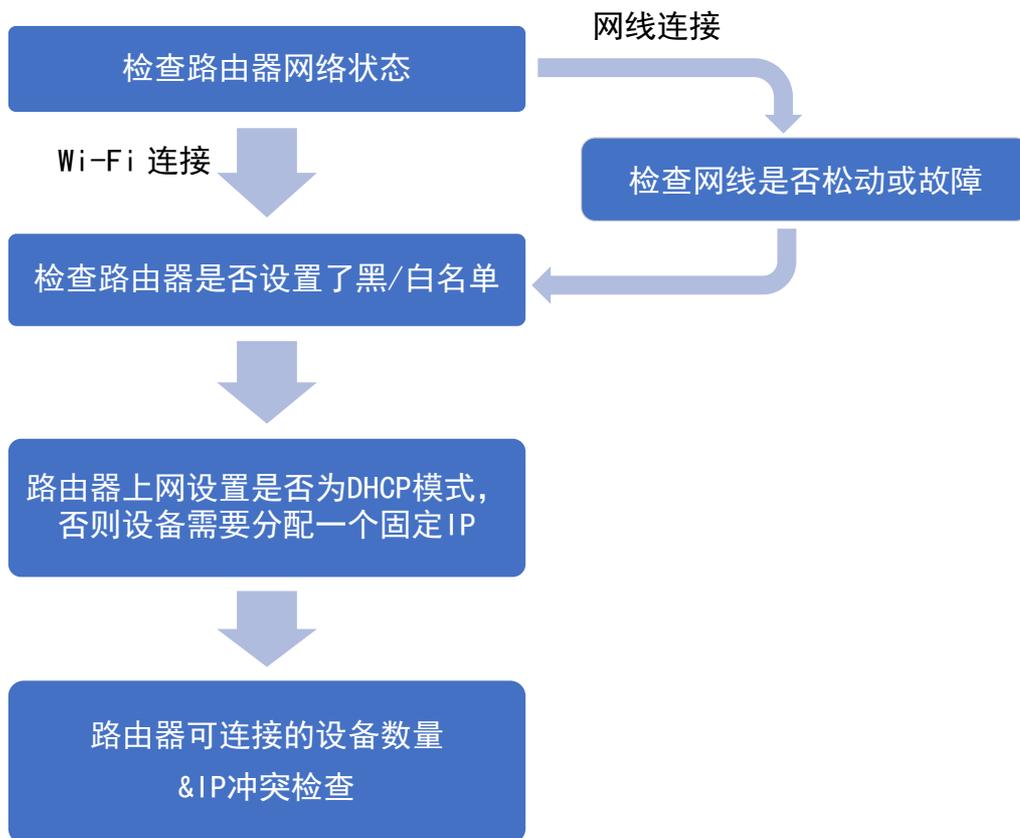
局域网网络，网络通畅。

i 注意：设备也可连接公网使用，此时若要与设备通信，请使用与设备同网段的客户端。相当于还是局域网内通信。

必备条件：设备屏幕上网络连接标志无异常（异常标志：）。

4.2 路由器检查

若路由器网络故障、IP 分配额不足、连接列表有黑名单、白名单设置等，会导致设备无法接入网络。请按以下步骤检查网络：



4.2.1 检查路由器网络状态

- 1、登录路由器设置页面（管理界面）。不同的无线路由器，设置页面（管理页面）的地址不一样，有些路由器是 IP 地址，有些是网址。可以在路由器底部标签中，查看路由器的设置网址信息，如下图所示。



- 2、若设置了管理员密码，需输入密码登录，如图所示（下文若无说明均以 TP-LINK 路由器为例）。**管理员密码一般为设备连接路由器的 Wi-Fi 密码。**



- 3、登录后进入路由器主页面，可查询路由器网络状态是否正常，如图所示。



4.2.2 路由器黑名单/白名单设置

若路由器设置了允许访问的设备（白名单）列表，则其他设备即时输入正确的账号密码也无法访问网络。需将设备的 MAC 地址添加到允许访问的设备列表中。

1、登录路由器设置页面（管理页面），点击“应用管理”模块，选择“无线设备接入控制”，如图所示。



2、有两种模式可选择。

不开启：未被禁用的设备都可以正常连接路由器网络。



开启：只允许列表内的设备连接到路由器网络（列表中的设备需勾选）。



选择设备添加

若设备已连接上路由器网络，则可点击“选择设备添加”。

以下设备已连接到路由器，请选择添加允许接入的设备：

<input type="checkbox"/>	设备	MAC地址	IP地址
<input checked="" type="checkbox"/>	Guests-MBP	98-01-A7-8A-0E-C3 (本机)	192.168.24.11
<input type="checkbox"/>	Smartisan-U3-Pr...	B4-0B-44-EC-7D-82	192.168.24.9

注意：如果要添加的设备不在列表中，请将该设备连接主人网络后刷新。

刷新

确定

取消

输入 MAC 地址添加

若设备未连接上路由器网络，可点击“输入 MAC 地址添加”。

无线设备接入控制 ?

接入控制功能 开启

只允许列表内的设备连接到主人网络

保存

允许接入设备列表

选择设备添加
输入MAC地址添加
删除全部
删除所选

<input type="checkbox"/>	主机	MAC地址	编辑
	<input style="width: 90%;" type="text"/>	<input style="width: 90%;" type="text"/>	保存 取消
<input type="checkbox"/>	OnePlus_5	94-65-2D-6A-69-B1	<input type="text" value="✎"/>
<input type="checkbox"/>	BOOMdeMBP	8C-85-90-02-78-9E	<input type="text" value="✎"/>

4.2.3 路由器是否为 DHCP（自动获取 IP）模式

路由器上网设置是否为 DHCP（自动获取 IP）模式，若 DHCP 模式关闭，则设备需要分配一个固定 IP。

路由器 DHCP 服务器

登录路由器设置页面（管理页面），点击“路由设置”模块。选择“DHCP”服务器，可以看到路由器的 DHCP 模式是否开启。选择开启，则路由器会为每个接入的设备自动分配一个 IP；选择关闭，则没有设置固定 IP 的设备，即时接入路由器也无法访问网络。



分配固定 IP

- 1、 登录路由器设置页面（管理页面），点击“应用管理”模块，选择“IP 与 MAC 绑定”，如图所示。



2、IP 与 MAC 映射表，状态“绑定”表示该 MAC 地址的设备已经由路由器分配了固定 IP。

IP与MAC映射表

刷新

主机	MAC地址	IP地址	状态	添加到绑定设置
匿名主机	74-1F-4A-0F-A4-75	192.168.20.1	未绑定	+
D42	84-E0-F4-20-0D-42	192.168.24.198	绑定	+
JX9NBF0B3...	C8-FF-28-89-BC-3E	192.168.24.19	未绑定	+
android-3291dc4...	EC-3D-FD-28-25-4F	192.168.24.13	未绑定	+
Guests-MBP	98-01-A7-8A-0E-C3	192.168.24.11	未绑定	+
android-2a0f2bf...	84-E0-F4-20-01-94	192.168.24.10	未绑定	+
Smartisan-U3-...	B4-0B-44-EC-7D-82	192.168.24.9	未绑定	+
iPhone-2	F4-0F-24-B3-A6-1C	192.168.24.8	未绑定	+

⏪ ⏩ 1 2 ⏪ ⏩

3、IP 与 MAC 绑定设置，点击“添加”，输入设备名称、MAC 地址、IP 地址。设备 MAC 地址为设备序列号前 12 位，如图划红线部分所示。

IP与MAC绑定设置

添加 删除全部 删除所选

<input type="checkbox"/>	主机	MAC地址	IP地址	编辑
<input type="checkbox"/>				保存 取消
<input type="checkbox"/>	D42	<u>84-E0-F4-20-0D-42</u>	192.168.24.198	✎

4.2.4 路由器可连接的设备数量限制&设备 IP 冲突检查

若路由器连了多个交换机，连接的设备较多，IP 就会分配完。

路由器可连接的设备数量

路由器连接的设备数量是有限制的，这与路由器性能与宽带大小有关。

手机、电脑等设备要连接路由器上网，必须获得一个路由器的本地 IP 地址。路由器 IP 地址通常为 192.168.1.1-192.168.1.255，其中 192.168.1.1 一般为路由器后台管理地址，所以理论上路由器分配给设备的 IP 地址只有 $255-1=254$ 个，即最多支持 254 个设备同时连接上网。但实际上一般的路由器在性能上无法同时满足这么多设备的上网请求，另外宽带大小也很难满足。

实际上，无线路由器最多能连接多少用户主要取决路由器的性能与宽带大小，一般的家用路由器能够同时连接 10-40 个左右的无线设备。常见的无线路由器最多只能连接 16 个无线设备，多于 16 个设备连接就会出现某些设备连接不上的情况。

如：办公室内的 Wi-Fi 一些设备可以连上，另外一些设备连不上；如果有设备把连接的 Wi-Fi 断开，原来连接不上的设备就可以连上，而且数量是相等的。到路由管理页面查看，正好是 16 台连上 Wi-Fi，再多就连不上。

因此，一般家用路由器连接设备是有限制的，也达不到理论上 254 台。若想要路由器尽可能连接更多设备，则需要增加宽带、购买性能强的路由器，比如双频多天线无线路由或购买企业级路由器，能够支持的设备会更多。

检查 IP 是否冲突

点击任务栏的“开始-运行”，在运行对话框上输入 cmd 命令，执行该命令即可打开命令提示符，在界面中键入“ping [IP 地址]”，若正常返回字节和响应时间，则该 IP 地址已被占用。



```
管理员: C:\Windows\system32\CMD.exe
Microsoft Windows [版本 6.1.7601]
版权所有 (c) 2009 Microsoft Corporation。保留所有权利。

C:\Users\Administrator>ping 192.168.1.110

正在 Ping 192.168.1.110 具有 32 字节的数据:
来自 192.168.1.110 的回复: 字节=32 时间<1ms TTL=63

192.168.1.110 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
    往返行程的估计时间<以毫秒为单位>:
        最短 = 0ms, 最长 = 0ms, 平均 = 0ms

C:\Users\Administrator>
```

可尝试的解决方法有：关闭路由器，待设备断开网络连接后，再重新打开路由器，等待路由器重新分配 IP 地址；手动设置固定 IP。

4.3 有线连接及检查

设备网口连接网线，若出现“网络异常”或网络标志异常，则检查步骤为：

- 1、检查网线是否松动，尝试插拔网线。
- 2、将同一根网线连接其他设备，如电脑，检查是否能使用网络。
- 3、换一根网线，该网线连接别的电脑能上网，检查连接设备是否能上网。

若第 2 步结果为其他设备、设备都不能上网，第 3 步结果为设备能正常上网，则考虑第一根网线故障；

若 2、3 步结果为网线其他设备能上网，连接设备不能上网，则考虑路由器问题，参考第二章，路由器黑白名单、连接设备数量限制、固定 IP、IP 冲突

若确定以上步骤项均正常，则考虑是设备本身网络模块问题。若有另外一台人脸设备，重复步骤 1、2、3，若该设备能正常上网，则第一台设备网络连接模块故障。

4.4 无线连接及检查

设备支持连接 2.4G 频段 Wi-Fi，不支持连接以下 Wi-Fi:

- 1、5G 频段的 Wi-Fi
- 2、需要 Web 页面认证 Wi-Fi
- 3、需要证书认证的 Wi-Fi

设备连接 Wi-Fi 步骤

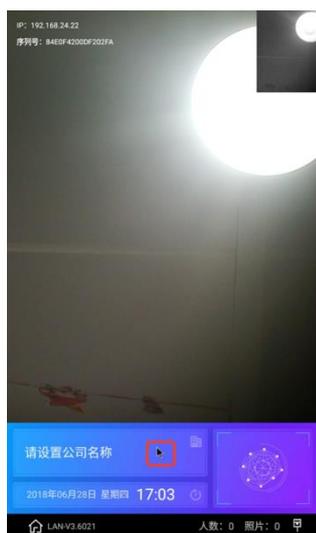
设备连接 Wi-Fi 共有两种方式，可选择:

方式一: Wi-Fi 配置接口向设备发送 Wi-Fi 账号密码，参见 Wi-Fi 配置接口。

方式二: 设备 USB 口外接鼠标打开设备菜单页面，见下文。

设备 USB 口外接鼠标（Uface-M5201 系列设备不支持此方式），点击打开设备菜单，设置网络配置。操作步骤及说明界面如下:

(1) 设备 USB 口插上鼠标后，设备屏幕上会出现鼠标，如图所示。点击屏幕左下角图标，即可打开设备菜单。



出现鼠标



点击菜单图标

(2) 首次使用设备，需设置设备密码，如图所示。输入密码后，点击提交。每次打开设备菜单，都需要登录。每台设备都有唯一专属密码，请牢记此密码，若遗忘密码，则需重置设备。重置及恢复数据操作[参见重置设备](#)。



设置设备密码



提交密码

(3) 设置设备密码成功后，再次输入密码进行登录，如下图（左）所示。若设备已有密码，打开菜单后直接输入密码进行登录即可。菜单主界面如下图（右）所示，可进行网络配置和修改密码操作。



登录



菜单界面

(4) 点击修改密码，在修改密码界面输入新旧并提交，即可修改成功，如图所示。



点击修改密码



输入新旧密码并提交

(5) 点击网络配置，左侧可选择有线、无线功能模块如图所示。



选择有线或无线



选择网络配置

(5) 在有线模块，根据现场网络环境要求可选择设备 IP 模式为 DHCP（自动获取 IP）或固定 IP，点击保存即可。



选择 DHCP
(自动获取 IP)



选择固定 IP
并填写网络信息

(5) 若选择无线模块，则需断开网线连接。无线模块界面如图所示，点击可刷新 Wi-Fi 列表。



选择 DHCP
(自动获取 IP)



选择固定 IP
并填写网络信息

(5) 选择需要连接的 Wi-Fi，输入密码，若现场网络允许设备自动获取 IP，点击连接即可，如图所示。



选择 Wi-Fi



输入密码



连接中

(6) 若现场网络要求设备固定 IP，点击高级，可对设备设置固定 IP。连接成功后，Wi-Fi 列表中该 Wi-Fi 会有绿色勾选提示，如图所示。



选择高级
并填写网络信息



设备当前连接的 Wi-Fi

连上 Wi-Fi 但无网络

输入 Wi-Fi 账号密码，成功连上 Wi-Fi 后，设备还是显示网络异常界面（[参见设备界面说明](#)）。则检查步骤为：

- 1、 检查连接此 Wi-Fi 的其他设备是否能正常上网。
- 2、 检查设备是否通过路由器或接口设置过固定 IP，若设置过固定 IP，则连接的网络需要跟设备固定 IP 是同一网段，设备才可访问网络。

i 注意：若设备之前通过路由器设置页面绑定过 MAC 和 IP，该固定 IP 只对连接此路由器时有效；即换一台路由器不影响设备使用，可由别的路由器自动分配 IP 或固定 IP。

若其他设备能上网，设备不能上网，考虑路由器问题，参考第二章，路由器黑白名单、连接设备数量限制、固定 IP、IP 冲突

若确定以上各项均正常，则考虑是设备本身网络模块问题。若有另外一台人脸设备，连接同样的 Wi-Fi 并进行正确的路由配置，若该设备能正常上网，则第一台设备网络连接模块故障。

4.5 网络常见问题

Q1：网络环境丢包率

丢包率高会导致设备与服务器通信失败，服务器推送的同步、加人、删人等消息设备无法接收到，设备向服务器上传的识别记录、现场照等会丢失。

丢包率高的原因有：

1、物理线路故障

网络线路时通时断，有可能是网络线路出现故障，也可能是用户方面的原因。

由物理线路引起的丢包现象还有很多，如光纤连接问题，跳线没有对准设备接口，双绞线及 RJ-45 接头有问题等。另外，通信线路受到随机噪声或者突发噪声造成的数据

报错误，射频信号的干扰和信号的衰减等都可能造成数据包的丢失。可以借助网络测试仪来检查线路的质量。

2、连网设备故障

连网设备故障主要是硬件方面的故障。如网卡故障，交换机的某个端口出现物理故障，光纤收发器的电端口与网络设备接口，或两端设备接口的双工模式不匹配。

如：交换机端口的光纤模块故障，交换机接收到的数据包在转发过程中不会被发送出去，也不会被丢弃。它们将会堆积在动态缓存中，等到缓存中堆积满了，就会造成交换机死机的现象。解决方法是更换一条新的光纤模块。

3、网络被堵塞、拥堵

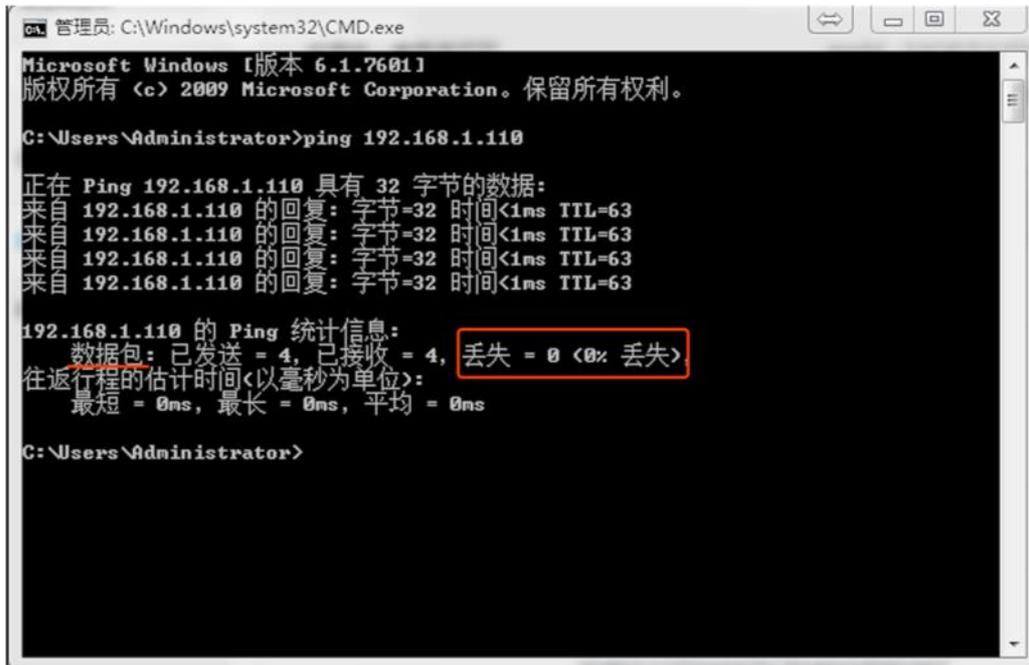
网络拥塞造成丢包率上升的原因很多，主要是路由器资源被大量占用造成的。

其他造成网络拥塞的情况，如大量的 UDP 流量、大量的组播流、广播包穿越路由器，路由器配置了 IP NAT 并且有很多 DNS 包穿越路由器等。上述情况造成网络拥塞后，通信双方会采取流量控制，丢弃不能传输的包。

4、路由错误

网络路径错误也会导致数据包不能到达目标，如路由配置错误，设备发出的访问云端服务器的数据包会被网关丢弃。但此类丢包属于正常情况下的丢包，是意料之中的，不会对网络造成影响。

查询网络丢包率



点击任务栏的“开始-运行”，在运行对话框上输入 cmd 命令，执行该命令即可打开命令提示符，在界面中键入“ping[网址]”，显示最后一行(x% loss)就是对目标地址 ping 包的丢包率，如图所示。为了更好确定丢包原因，建议尝试以下操作：

- 1、 同一路由器，电脑（有线连接）分别 ping 设备 IP、www.baidu.com
- 2、 同一路由器，电脑（Wi-Fi 连接）ping 设备 IP、www.baidu.com

Q2: 路由器限速或弱网环境

路由器限速会导致人脸设备与云端服务器通信不顺畅，影响识别记录、现场照上传到服务器，也会影响服务器向设备下达的指令及传输的数据（丢包率会提高）。具体表现有：识别记录丢失、现场照无法上传或上传较慢，设备拍照注册、卡号注册上传失败，同步数据较慢或丢失。

网络环境通信速度表

网络状况	上传/下载速度	说明
极差	0~4KB/S	设备与服务器无法通信
弱网	10 /30KB/S	<ul style="list-style-type: none"> • 识别抓拍照片上传失败、识别记录上传时断时续 • 服务器下达指令丢失，人员、照片同步到设备较慢且容

		易丢失 <ul style="list-style-type: none"> • 设备升级容易失败
2G	15 /50KB/S	<ul style="list-style-type: none"> • 识别抓拍照片上传较慢 • 服务器下达指令有概率丢失，人员、照片同步到设备过程中会出现丢失 • 设备升级较慢，有概率失败
3G	384 /2800KB/S	设备正常使用，数据丢失概率极低。一般网络波动会导致数据丢失， 参见网络丢包率 。

Q3: 固定 IP

是否需要固定 IP 是由现场路由器的设置要求决定的。若路由器为 DHCP 模式，且连接的设备较多，超出路由器允许连接的设备数量，会导致设备无法上网，[参见附录 6.2.3 路由器是否为 DHCP（自动获取 IP）模式](#)；此时可将设备 MAC 地址与 IP 绑定，[参见分配固定 IP](#)。

Q4: IP 冲突

设置固定 IP 时，若重复给两台设备设置相同 IP，会导致 IP 冲突。路由器自动分配的 IP 也会出现 IP 冲突。可通过 ping IP 地址来确认该 IP 是否已被使用，[参见检查 IP 是否冲突](#)。

Q5: 连上 Wi-Fi 连接，没有网络异常标志，但无法上网

一般出现这种情况导致的原因有：网络、路由器、设备。主要考虑路由器 IP 冲突、黑/白名单，[参见附录 6.2 路由器检查](#)，检查步骤[参见附录 6.4 无线连接检查](#)。同时检查路由器网络是否有限速，路由器网络是否被大量占用使用。

Q6: 连上网线连接，没有网络异常标志，但无法上网

一般出现这种情况导致的原因有：网络、路由器、设备。主要考虑路由器 IP 冲突、黑/白名单，[参见附录 6.2 路由器检查](#)，检查步骤[参见附录 6.3 有线连接检查](#)。同时检查路由器网络是否有限速，路由器网络是否被大量占用使用。

Q7: 数据丢失、上传较慢、指服务器下达指令丢失或较慢等

首先检查网络环境限速和丢包率，[参见 Q1、Q2](#)。

若有另外一台设备，可使用同一根网线连接或连接同一个 Wi-Fi，将两台设备进行比较，若第二台设备正常，则考虑第一台设备网络模块故障。

若设备同时支持 Wi-Fi、网线连接，可更换连接方式，若更换连接方式后设备通信正常，则考虑是前一种连接方式模块故障。

Q8: 同时连接了网线和 Wi-Fi

若设备同时连接 Wi-Fi、网线且网络正常，设备优先使用有线网络配置。

若设备 Wi-Fi 配置固定过 IP，有线网络也固定过 IP，此时以有线网络固定 IP 为准。

若设备 Wi-Fi 配置固定过 IP，有线网络有 DHCP 模式，此时以有线网络自动分配 IP 模式为准。

Q9: 取消 Wi-Fi 配置、配置新的 Wi-Fi

若要取消 Wi-Fi，则需重置设备配置。设备背面的 RESET 小孔，用针长按 8-12 秒，设备会恢复出厂设置，此时设备上所有数据都会清空。

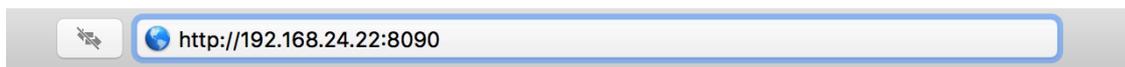
若要配置其他 Wi-Fi，需要让设备重新进入开启热点的网络配置界面（[参见设备界面说明](#)）。以下方式可使设备出现该界面：

- 1、 将设备放置在没有配置过 Wi-Fi 的环境中。
- 2、 将配置过的 Wi-Fi 断开，可更改路由器密码或关闭路由器。
- 3、 重置设备，备背面的 RESET 小孔，用针长按 8-12 秒，设备会恢复出厂设置，此时设备上所有数据都会清空。

附录 5 设备本地管理页面

使用与设备处于局域网内同一网段的电脑，可以通过设备 IP 访问设备本地管理页面，目前该页面支持本地上传固件包升级。操作步骤如下：

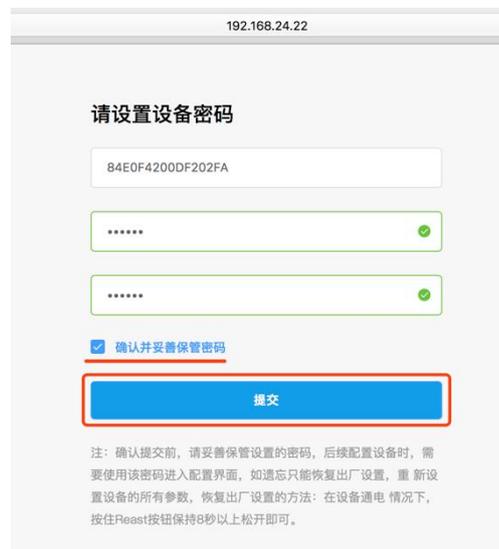
- (1) 在浏览器中输入 `http://设备 IP:8090`，并访问。如图所示。



- (2) 若首次访问设备，需要先设置设备密码，如图所示。



设置密码



确认并提交

i 注意：a) 该密码与使用鼠标打开设备菜单的密码、通过接口设置的设备密码一致，都是设备密码，设备有且只有唯一的设备密码。使用鼠标打开设备菜单设置密码[参见附录 6 设备 USB 口外接鼠标](#)。

b) 通过此方法访问设备，设备联网方式允许公网、局域网或与电脑直连，但电脑必须和设备处于局域网同一网段。

(3) 登录成功后进入本地管理主界面，目前功能仅支持本地上传固件包升级。本地升级界面如图所示。



(4) 选取本地的升级包文件并上传，升级包为 zip 文件，由我司提供，遵从命名规则，如图所示。



(5) 向设备传送固件包界面如图所示。



i 注意：若跟设备网络通信有问题，会出现如下界面。



(6) 升级包传送过程中，设备会出现下载界面，如图所示。

i 注意：固件上传过程中请勿刷新页面。刷新页面会导致升级中断，需重新升级。



i 注意：若升级过程中断电或断网，需从设备本地管理页面重新点击升级，设备会重新开始下载。

```
public class BaseResult {
    public static final byte CALL_RESULT_FAILED = 0;
    public static final byte CALL_RESULT_SUCCEED = 1;
    protected byte result;
    protected String code;
    private String msg;

    public BaseResult() {
    }

    public BaseResult(String code) {
        this.result = 1;
        this.code = code;
        this.msg = "success";
    }

    public BaseResult(String code, String msg) {
        this.result = 1;
        this.code = code;
        this.msg = msg;
    }

    public BaseResult(byte result, String code) {
        this.result = result;
        this.code = code;
        this.msg = "success";
    }

    public BaseResult(byte result, String code, String msg) {
        this.result = result;
        this.code = code;
        this.msg = msg;
    }

    public byte getResult() {
        return this.result;
    }
}
```

附录 6 回调示例详情

```
public void setResult(byte result) {
    this.result = result;
}

public String getCode() {
    return this.code;
}

public void setCode(String code) {
    this.code = code;
}

public String getMsg() {
    return this.msg;
}

public void setMsg(String msg) {
    this.msg = msg;
}
```

```
public class ContentResult<T> extends BaseResult {
    private T data;

    public ContentResult() {
    }

    public ContentResult(String code) {
        super(code);
    }

    public ContentResult(String code, T t) {
        super(code);
        this.data = t;
    }

    public ContentResult(byte result, String code) {
        super(result, code);
    }
}
```

```
public ContentResult(byte result, String code, T t) {
    super(result, code);
    this.data = t;
}

public T getData() {
    return this.data;
}

public void setData(T data) {
    this.data = data;
}
}
```

```
//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);
}
```

附录 7 msg 信息及异常码

msg 信息

"参数异常"
"请使用 POST 请求"
"接口服务未设置密码，请先设置密码"
"密码中不能包含中文！"
"密码错误，请检查密码正确性"
"请检查密码是否为空"
"初次设置密码，请确保 oldPass，newPass 相同"
"旧密码错误，要修改密码，请确保旧密码和原来设置的密码相同"
"ttsModType 参数异常"
"ttsModStrangerType 参数异常"
"displayModType 参数异常"
"comModType 参数异常"
"ttsModContent 参数异常"
"comModContent 参数异常"
"重启后即改变方向。"
"已设置为 DHCP 模式,请 5 秒钟后重启设备。"
"配置成功，请 5 秒钟后重启设备。"
"ip 参数异常"
"isDHCPMod 参数异常"
"dns 参数异常"
"gateway 参数异常"
"ssId 参数异常"
"完成清除"
"设置成功"
"重启后即改变方向。"
"屏幕模式选择成功，重启后生效"
"修改背景图片成功"
"图片过大"
"请输入正确格式的 url 地址"
"name 参数不能为空"
"创建人员失败，请检查数据格式"
"不存在该人"
"参数异常，页码及长度信息请务必为数字"
"personId 不能为空！"
"一页显示数量要求为(0,1000]的正整数"
"index 页码不能小于 0"

"personId=-1,length=-1 时总页码 size=0,index 不能大于 0! "
"已经存在期限设置, 请勿重复设置。"
"passtime 时间格式错误"
"参数异常, 请检查日期格式是否正确"
"参数异常, 开始时间须早于结束时间"
"base64 编码格式不对! "
"base64 图片解析异常"
"图片解析异常"
"已达到最大照片数量限定"
"该人照片数量超过 3 张, 请先进行照片删除。"
"设备正忙, 请稍后再试,设备不支持并发"
"更新照片异常"
"照片 ID 不存在, 请先调用照片创建接口。"
"未查询到该人,请先添加该人员。"
"特征无效, 请检查特征值及 featureKey 是否合法。"
"featureKey 不合法。"
"参数异常, 页码及长度信息请务必为数字"
"数据库中没有该 personId! "
"数据库中没有该 personId 的识别记录! "
"数据库中没有该 personId 的刷卡记录! "

异常码

-8001	FaceSDK no Prepare
-8006	未检测到人脸
-8007	检测到多个人脸
-8010	人脸过小
-8011	FaceSDK 无法从照片中提到特征
-8012	FaceSDK 提取特征异常
-8013	人像超出或过于靠近屏幕边缘
-8014	人脸角度过大
-8015	光照检测未通过
-8016	模糊度检测不通过
-8017	阴阳脸检测不通过

设备异常

101: "摄像头没打开, 请尝试重启。如重启后未能解决, 请联系客服。"

102: "摄像头没打开, 设备将重启。"