



局域网版说明文档

局域网是针对在局域网内部署的设备，设备随版本迭代完善接口功能，开发人员可通过同局域网内的客户端发送 HTTP 请求与设备直接进行通信。本文档对局域网版设备（又称离线版）进行简单介绍，并对设备提供的接口加以解释说明，以便开发人员能够更好得使用和理解各个接口。

设备版本说明

设备版本号	发布日期	说明
V1.0	2017.07.14	初版
V1.1	2017.07.18	1.新增设备序列号获取接口(3.3.3) 2.优化识别记录查询接口，支持按时间查询识别记录 3.新增应用程序生命周期描述
V1.2	2017.07.26	1.新增识别回调设置接口(3.3.5) 2.新增接口密码配置接口(3.3.4) 3.所有接口增加密码验证
V1.3	2017.08.02	新增屏幕方向配置接口(3.3.6)
V1.4	2017.08.10	新增设备网络信息配置接口(3.3.7)，可手动配置 IP，设备默认为 DHCP 自动获取 IP 模式
V1.5	2017.08.24	1.设备新增大屏模式，HDMI 输出 1080P 高清画面，仅限横屏模式，仅主界面与小屏幕有区分 2.新增大屏信息展示区内容配置接口（3.3.8）
V1.6	2017.08.30	1.新增自动拍照接口(3.2.4)，对已注册人员通过设备自主拍照进行注册 2.新增指定日期前识别记录删除接口(3.4.2)
V1.7	2017.09.15	1.新增身份证/IC 卡录入注册接口(3.2.6) 2.新增 IC 卡识别记录查询接口(3.4.3) 3.查看注册人员识别记录时新增人脸识别记录现场照片路径，可直接访问(3.4.1) 4.新增注册照查询接口，返回人员注册照信息及注册照路径，可直接访问(3.2.5) 5.新增设备重启接口(3.3.9) 6.识别回调中，新增打卡类型区分字段，type: face/card 7.修复识别距离无法配置的 BUG
V1.8	2017.09.20	1.新增自动拍照及照片注册结果回调设置接口(3.3.10) 2.所有接口说明进行了完整性补充
V1.9	2017.09.21	1.新增特征注册人脸接口(3.2.6) 2.新增设备心跳配置接口(3.3.11)
V2.0	2017.09.26	1.修复升级应用 V2.5 后人脸信息没存进数据库的问题 2.照片查询接口(3.2.4)返回的数据不包含现场照特征信息 3.时间窗由分钟改为秒(只有秒)，修复识别回调没用上时间窗的 bug 4.修复竖屏屏保显示不正确的问题 5.修复识别分数高于 92 分时，现场照特征替换不了的问题 6.修复竖屏模式下，现场照方向不正确的问题 7.修复相机没数据后，没有重新开启相机的问题。
V2.3	2017.09.27	新增照片人脸相似性比对接口(3.2.7)
V2.6	2017.09.29	1.属性配置接口新增串口输出模式 3,为韦根信号输出，详见属性配置接口(3.3.1) 2.新增设备时间设置接口(3.3.12) 3.修改照片添加逻辑为必须现有人员才能添加

V2.7	2017.10.09	1.修改设备名称默认为“人脸识别系统” 2.默认 logo 完全中性化
V2.8	2017.10.12	1.接口及文档版本与应用版本统一 2.修改打卡回调的时间戳，使得打卡回调的上报的时间戳和通过接口查询打卡记录的时间戳保持一致 3.修改识别记录删除接口(3.4.2)，参数由原来的精确到“年月日”改为精确到“年月日时分秒” 4.新增一个通过“毫秒级时间戳”删除识别记录接口(3.4.4) 5.属性配置接口(3.3.1)新增设置是否自动重启开关，如果设置了自动重启，每天凌晨 3: 00 设备会自动重启，以保证设备后续能够稳定运行 6.修改大屏默认文字信息提示区域内容为“人脸识别系统” 7.查询识别记录接口(3.4.1)参数的查询时间精确到秒
V 2.9	2017.11.06	1.新增人员创建(3.1.1)、人员更新接口(3.1.2)，替换人员创建或更新接口 2.新增照片创建(3.2.1)、照片更新接口(3.2.2)，替换照片创建或更新接口 3.新增设备重置接口(3.3.13)，可选择清空设备内所有文件或清空文件并恢复出厂设置 4.属性配置接口(3.3.1)setConfig 新增字段 ttsModType（语音模式类型）、ttsModContent（语音播报模式自定义内容）、displayModType（屏幕显示模式类型）、displayModContent（屏幕显示模式自定义内容）、comModType（串口模式类型）、comModContent（串口模式自定义内容）、slogan（标语，大屏模式）、intro（公司简介，大屏模式）；原语音模式、显示模式、串口模式字段已删除 5.人员删除接口(3.1.3)修改为人员批量删除接口，personId 用英文逗号间隔，可同时删除多个人员；传入“-1”则删除所有人员，删除人员的时候同时删除该人的注册照、特征、现场照、识别记录 6.版本号增加前缀，格式为：LAN-V2.9 7.大屏模式更新，增加 slogan（标语）、intro（公司简介）可编辑区域，去掉左上角 Offline 文字 8.记录查询接口 3.4.1、3.4.2、3.4.3 时间参数格式修改为 yy-mm-dd hh:mm:ss 9.修改设备名称默认为“请设置公司名称”
V 3.0	2017.11.13	内部优化，设备数据库维护
V 3.1	2017.11.29	1.设备重置后，设备恢复出厂设置，恢复成 DHCP 自动获取 IP 模式，恢复成横屏模式 2.识别回调 url 传入空可清空回调地址设置，清空后识别成功将不再进行回调 3.修复竖屏模式下，人员名字显示的问题 4.属性配置接口(3.3.1)增加陌生人配置，新增字段 recStrangerType（陌生人开关）、recStrangerTimesThreshold（陌生人判定）、ttsModStrangerType（陌生人语音模式类型）、ttsModStrangerContent（陌生人语音播报自定义内容）

		5.新增时间段权限设置接口(3.1.5)，已注册的人员始终可识别，但若人员在非允许时间段内识别，设备会提示“姓名+权限不足”；若设备开启语音播报，则会播报“权限不足”；若设备串口输出连接门禁，则不会开门 6.新增时间段权限删除接口(3.1.6)，删除人员的时间段权限设置，人员不再有时段权限限制 7.识别记录查询接口(3.4.1)、IC 卡识别记录查询接口(3.4.3)返回数据增加字段 state（识别记录回调结果，成功/失败）、type（识别出的人员类型，时间段内/外） 8.若设备设置了识别回调地址，则设备每间隔 10 分钟，从数据库读取回调失败(state:0)的识别记录，再次向回调地址发送请求，请求成功后更新数据库中对应的回调状态(state:1) 9.回调数据字段 type 格式为：face/card_0/1，face、card 表示识别方式，0、1 表示识别出的人员类型，0：时间段内，1：时间段外 10.属性配置接口(3.3.1)配置成功，返回结果 result=0 改为 result=1；识别记录查询接口(3.4.1)、IC 卡识别记录查询接口(3.4.3)，返回字段名 recordses 改为 records
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.新增人员照片清空接口(3.2.9)，可清空某人员 id 的所有注册照片
V 3.3	2017.12.29	1.新增人员权限设置接口(3.1.7)，人员到期设备会定时删除该人员 2.新增人员权限设置删除接口(3.1.8)，清除人员的权限设置，人员不会被定时删除 3.属性配置接口(3.3.1)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(3.3.14)，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(3.3.15)，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(3.3.16)，可上传自定义图片作为大屏背景（需将 3.3.15 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	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 1.属性配置接口(3.3.1)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.属性配置接口(3.3.1)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(3.3.17)，配有 Wi-Fi 模块的设备可通过该接口连接 Wi-Fi 4.新增指定日期前刷卡记录删除接口(3.4.5) 5.新增刷卡记录删除接口（通过毫秒级时间戳进行删除）(3.4.6) 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 7.识别回调数据从 POST 请求的 url 和 body 都可接收到 8.一代设备、二代设备（仅支持竖屏）界面显示优化 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 10.设备心跳回调数据新增字段 version，表示设备版本号 11.修复了二代设备现场照倒置的问题 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 13.附录新增回调示例详情
--	--	--

目录

1.概述.....	- 8 -
1.1 简述.....	- 8 -
1.2 接口规范.....	- 8 -
1.3 接口返回.....	- 8 -
1.4 设备识别人员接口调用流程.....	- 9 -
2.设备应用程序的生命周期.....	- 10 -
2.1 启动界面.....	- 10 -
2.2 程序主界面.....	- 10 -
2.3 屏保界面.....	- 11 -
2.4 设备异常提醒界面.....	- 11 -
2.5 大屏模式（仅横屏模式下可用）.....	- 12 -
3 接口说明.....	- 13 -
3.1 人员管理接口.....	- 13 -
3.1.1 人员创建接口.....	- 13 -
3.1.2 人员更新接口.....	- 13 -
3.1.3 人员批量删除接口.....	- 14 -
3.1.4 人员信息查询接口.....	- 14 -
3.1.5 时间段权限设置接口.....	- 15 -
3.1.6 时间段权限删除接口.....	- 16 -
3.1.7 人员权限设置接口.....	- 16 -
3.1.8 人员权限设置删除接口.....	- 17 -
3.2 照片管理接口.....	- 18 -
3.2.1 照片创建接口.....	- 18 -
3.2.2 照片更新接口.....	- 18 -
3.2.3 照片删除接口.....	- 19 -
3.2.4 自动拍照接口.....	- 19 -
3.2.5 照片查询接口.....	- 20 -
3.2.6 身份证/IC 卡注册接口.....	- 20 -
3.2.7 特征注册接口.....	- 21 -
3.2.8 照片人脸相似性比对接口.....	- 22 -
3.2.9 人员照片清空接口.....	- 23 -
3.3 设备管理接口.....	- 24 -
3.3.1 属性配置接口.....	- 24 -
3.3.2 Logo 修改接口.....	- 26 -
3.3.3 设备序列号获取接口.....	- 26 -
3.3.4 接口密码设置及修改接口.....	- 26 -
3.3.5 设置识别回调接口.....	- 27 -
3.3.6 修改屏幕方向接口.....	- 29 -
3.3.7 网络信息配置接口.....	- 29 -

3.3.8 大屏文字展示信息配置接口（已弃用，请使用 3.3.1 属性配置接口）	- 30 -
3.3.9 设备重启接口	- 31 -
3.3.10 自动拍照及照片注册结果回调设置接口	- 31 -
3.3.11 设备心跳配置接口	- 32 -
3.3.12 设备时间设置接口	- 33 -
3.3.13 设备重置接口	- 33 -
3.3.14 设备开门接口	- 34 -
3.3.15 大屏显示模式接口	- 34 -
3.3.16 大屏背景图片配置接口	- 35 -
3.3.17 Wi-Fi 配置接口	- 35 -
3.4 识别记录管理接口	- 37 -
3.4.1 识别记录查询接口	- 37 -
3.4.2 识别记录删除接口	- 38 -
3.4.3 IC 卡识别记录查询接口	- 38 -
3.4.4 识别记录删除接口（通过毫秒级时间戳进行删除）	- 39 -
3.4.5 刷卡记录删除接口	- 40 -
3.4.6 刷卡记录删除接口（通过毫秒级时间戳进行删除）	- 40 -
附录 1(msg 信息及异常码)	- 41 -
附录 2(回调示例详情)	- 43 -

1.概述

1.1 简述

局域网版本，即无需公网，局域网内就可通过设备对外提供的接口即可对设备进行操作。接口围绕人员管理、照片管理、设备信息配置等诸应用相关的核心业务，提供统一的对外服务接口，供客户进行调用。

1.2 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对接口(3.4)进行初始化密码设置，后续调用**任何接口**都需要传入 **pass** 作为接口安全校验密钥。
- **接口返回：**所有接口返回的数据，都包含三个字段，即 result、success、msg，称为基本字段；个别接口返回含 data 字段，携带接口响应数据处理结果，称为业务字段。

基本字段	描述	类型	附加说明
result	接口响应	Int	1 成功，0 失败
success	操作状态	Boolean	true 成功，false 失败
msg	返回信息	String	接口返回的信息，通常是错误类型码的原因信息

业务字段	描述	类型	附加说明
data	返回数据	Int/String/Object/List 等	返回的业务数据，类型可为数值、字符串或集合等

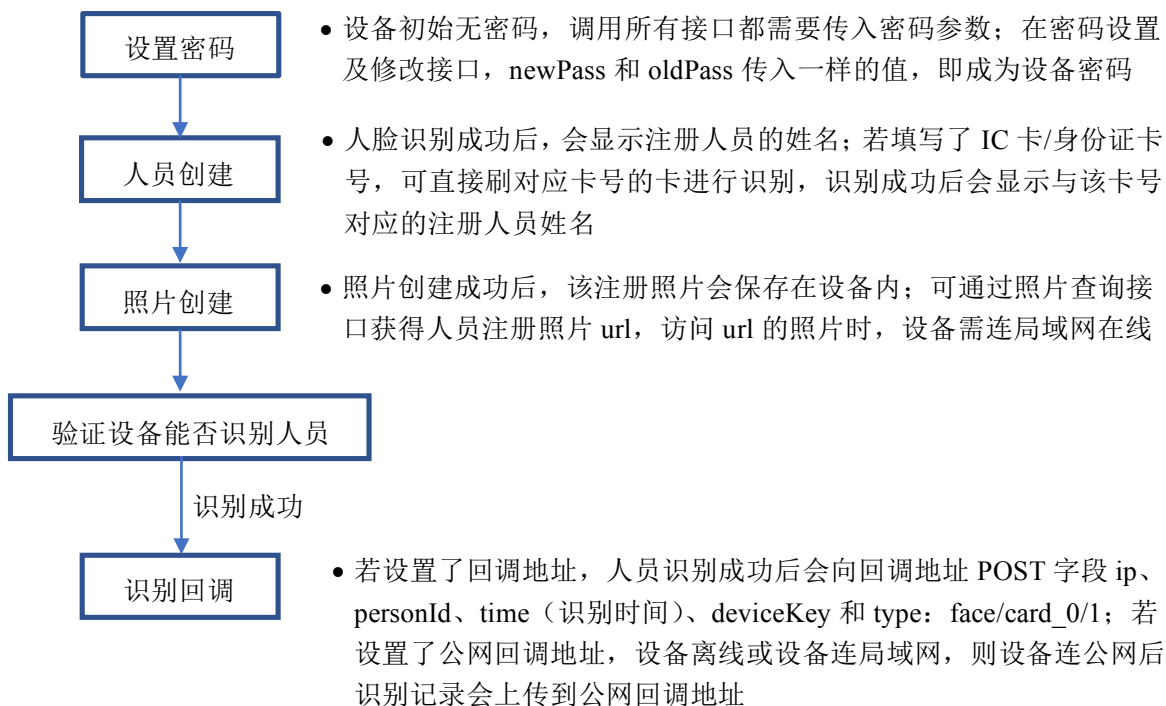
1.3 接口返回

接口通用返回说明：

```
public class ResultInfo<T> {
    private Int result;//仅表示接口调用状态，1 成功，0 失败，通常只要服务器能响应，该值均为 1
    private Boolean success;//操作状态，成功为 true，以该字段为准标识操作状态
    private T data;//接口返回数据封装类或集合
    private String msg;//异常信息提示
}
```

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。

1.4 设备识别人员接口调用流程



2.设备应用程序的生命周期

新设备初次启动，首先会自动进行应用程序的安装，安装完成后设备会自动重新启动。
应用程序启动后会有如下几个界面，左边为一代设备界面，右边为二代设备界面。

2.1 启动界面



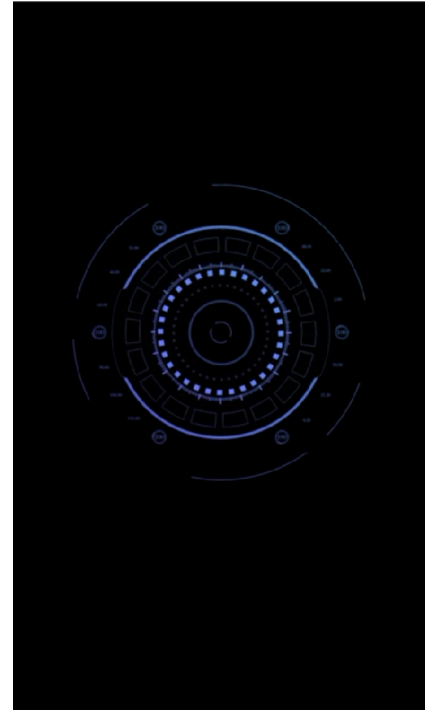
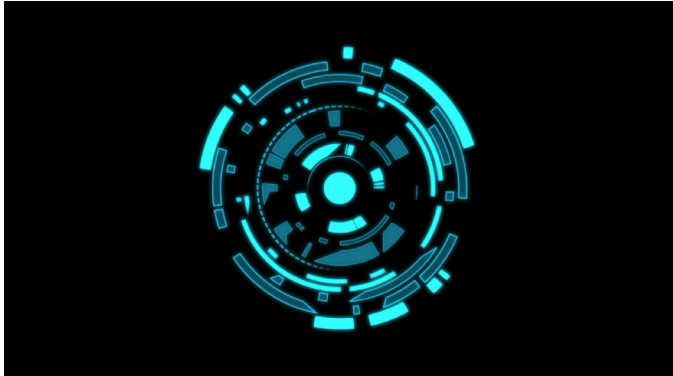
开机动画执行结束后会启动应用程序，显示如图。左上角会显示当前设备在局域网内的IP地址，用于访问设备进行接口调用。如果获取不到IP地址，会显示“未获取到IP地址”或“null”字样。设备每分钟会刷新一次IP。

2.2 程序主界面



- 1.设备名称：可自行通过属性配置接口(3.3.1)对设备名称(companyName)进行自定义
2. Logo 展示区：默认 Logo 为动画，可通过属性配置接口(3.3.1)对如图红框区域展示内容进行修改，目前仅支持静态自定义图片
- 3.信息：设备版本号，时间，注册人数展示区域

2.3 屏保界面



在主界面也就是识别界面 3 分钟没有人脸出现将会显示如图屏保界面，有人脸出现时会马上恢复到识别界面。

2.4 设备异常提醒界面



使用中如果设备出现异常，设备会进行自主提示，显示异常码和异常提醒等内容，并有提醒解决方案。

2.5 大屏模式（仅横屏模式下可用）

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。目前二代设备不支持大屏。



- 1.设备名称：**可自行通过属性配置接口(3.3.1)对设备名称(companyName)进行自定义
- 2.公司简介：**可自行通过属性配置接口(3.3.1)对公司简介(intro)进行编辑
- 3. Logo 展示区：**默认 Logo 为动画，可通过属性配置接口(3.3.1)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）：**可自行通过属性配置接口(3.3.1)对标语(slogan)进行编辑
- 5.视频预览区：**显示摄像头实时画面

3 接口说明

3.1 人员管理接口

3.1.1 人员创建接口

URL : /person/create

method : POST

参数列表 :

参数名	描述	类型	必传	附加说明
pass	接口安全校验密钥	String	Y	
person	person 对象的 Json 格式	Json	Y	• {"id":"9999911","idcardNum":"410822199908221428","name":"郝帅"} • id、idcardNum、name 内容可传空

返回示例 :

```
{"data":{"id":"9999911","idcardNum":"410822199908221428","name":" 郝 帅"}, "result":1, "success":true}
```

返回值 :

Person 类集合

说明:

```
public class Person {
    private String id;//人员 id, id 为人员标识, 创建人员时, 若 id 不填, 则系统会自动生成 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
    private String name;//姓名, 识别通过后会在屏幕上显示该名字
    private String idcardNum;//IC 卡卡号, 注册时可以不填; 若填写了 IC 卡号, 可直接刷对应卡号的卡进行识别, 也显示与该卡号对应的人员的名字
}
```

3.1.2 人员更新接口

URL : /person/update

method : POST

参数列表 :

参数名	描述	类型	必传	附加说明
pass	接口安全校验密钥	String	Y	
person	person 对象的 Json 格式	Json	Y	• {"id":"9999911","idcardNum":"410822199908221428","name":"郝帅"}

				• id 不可为空 • idcardNum、name 传入为空则会清空原信息
--	--	--	--	---

返回示例：

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

返回值：

通用操作状态

3.1.3 人员批量删除接口

URL : /person/delete

method : POST

参数列表：

参数名	描述	类型	必传	附加说明
pass	接口安全校验密钥	String	Y	
id	人员 id	String	Y	• 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： • 该人员 id、人员信息 • 该人员保存在设备库中的照片 • 该人员对应的识别记录 • 该人员对应的注册照片

返回示例：

• id 传入-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}
```

• id 传入有效及无效的人员 id

```
{"data":{"effective":"95194ab34e3549a8aa0a13d7a1478de3,8113a6ac883a408bb8adcba01c252c56","invalid":"ebd711a7d7594562b652bfcac5b5bfd42,4627d668fc864ba1ae2a15e490f46689"},"msg":"effective: 删除有效的 ID\ninvalid: 无效的 ID, 也就是不存在的","result":1,"success":true}
```

返回值：

会返回有效的 id（英文逗号拼接）及无效的 id（英文逗号拼接）

3.1.4 人员信息查询接口

URL : /person/find

method : GET

参数列表：

参数名	描述	类型	必传	附加说明
pass	接口安全校验密钥	String	Y	
id	人员 id	String	Y	- 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

返回示例：

```
{ "data": { "id": "9999911", "idcardNum": "410822199908221428", "name": " 郝 帅" }, "result": 1, "success": true }
```

返回值：

Person 类集合

```
public class Person {
    private String id;//人员 id
    private String name;//人员姓名
    private String idcardNum;//IC 卡卡号
}
```

3.1.5 时间段权限设置接口

URL : /person/createPasstime

method : POST

参数列表：

参数名	描述	类型	必传	附加说明
pass	接口安全校验密钥	String	Y	
passtime	某人员每日允许进入的时间段	Json	Y	{"personId": "9eccc839cd7941c5a4d3165202d3c32", "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 - 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 - 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 - 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

返回示例：

```
{ "msg": "passtime 添加成功", "result": 1, "success": true }
```

返回值：

passtime 设置结果

其他说明：

- 若传入 passtime，已注册的人员始终可识别，但若人员在非允许时间段内识别，设备会提示“姓名+权限不足”；若设备开启语音播报，则会播报“权限不足”；若设备串口输出连接门禁，则不会开门
- 回调字段 type，face/card 表示识别方式，0：时间段内，1：时间段外；face_0（表示是人脸识别，且该人员在 passtime 权限时间内），card_1（表示是刷卡识别，且该人员在 passtime 权限时间外）

3.1.6 时间段权限删除接口

URL： /person/deletePasstime

method： POST

参数列表：

参数名	描述	类型	必传	附加说明
pass	接口安全校验秘钥	String	Y	
personId	人员 id	String	Y	• 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，则可清除所有人员的 passtime

返回示例：

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

返回值：

passtime 删除结果

3.1.7 人员权限设置接口

URL： /person/permissionsCreate

method： POST

参数列表：

参数名	描述	类型	必传	附加说明
pass	接口安全校验秘钥	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： • 该人员 id、人员信息 • 该人员保存在设备库中的照片 • 该人员对应的识别记录 • 该人员对应的注册照片

time	权限到期时间，定时删除人员	String	Y	- 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 - 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 - 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 - 传入时间格式为（年-月-日 时:分:秒）： 2017-07-15 12:05:00
------	---------------	--------	---	--

返回示例：

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

返回值：

通用操作状态

3.1.8 人员权限设置删除接口

URL : /person/permissionsDelete

method : POST

参数列表：

参数名	描述	类型	必传	附加说明
pass	接口安全校验密钥	String	Y	
personId	人员 id	String	Y	- 删除该人员的权限设置，该人员不会被定时删除 - 若该人员未设置过权限时间，则调用此接口会报错

返回示例：

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

返回值：

通用操作状态

3.2 照片管理接口

3.2.1 照片创建接口

URL : /face/create

method : POST

参数列表 :

参数名	描述	类型	必传	附加说明
pass	接口安全校验密钥	String	Y	
personId	人员 id	String	Y	• 用于标识该照片属于某个人 id • 必须先创建人, 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 faceId 并在照片创建成功后返回
imgBase64	照片的 base64 编码	String	Y	不加头部, 如: data:image/jpg;base64,

返回示例 :

```
{"data":"061ce9d000c44d98818bf69021b916ec","result":1,"success":true}
```

返回值 :

faceId

其他说明 :

- 照片创建成功后, 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
- 若设置了照片注册或自动拍照注册回调接口, 注册成功后设备会将注册信息通过 POST 请求方式回调给预先设置的接口, 其中参数包含:

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	照片本地路径
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

3.2.2 照片更新接口

URL : /face/update

method : POST

参数列表 :

参数名	描述	类型	必传	附加说明
pass	接口安全校验密钥	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 对应的照片
imgBase64	照片的 base64 编码	String	Y	不加头部，如：data:image/jpg;base64,

返回示例：

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

返回值：

通用操作状态

3.2.3 照片删除接口

URL : /face/delete

method : POST

参数列表：

参数名	描述	类型	必传	附加说明
pass	接口安全校验密钥	String	Y	
faceId	照片 id	String	Y	调用该接口，会同步删除存储在本地（设备内）的注册照文件

返回示例：

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

返回值：

通用操作状态

3.2.4 自动拍照接口

URL : /face/takemg

method : POST

参数列表：

参数名	描述	类型	必传	附加说明
pass	接口安全校验密钥	String	Y	
personId	人员 id	String	Y	- 给指定人员 id 拍摄照片 - 人员 id 必须为已创建；若不存在该人员 id，则无法进入自动拍照模式

返回示例：

```
{"msg":"正在开启自动拍照模式，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}
```

返回值：

通用操作状态

3.2.5 照片查询接口

URL : /face/find

method : POST

参数列表：

参数名	描述	类型	必传	附加说明
pass	接口安全校验密钥	String	Y	
personId	人员 id	String	Y	• 查询指定人员 id 的注册照片

返回示例：

```
{ "data": [{"faceId": "fb196994efec4b6a800b621cc193787b", "feature": "L04AADj+jbkMAgAAgUaAPKxeq71cLeK9TJNsPDihwbtvt8Q9azqkvdZ3bwGQKk9gyTKPXj6w70FV969tpdRPaqolL1k6gK+zNHAvXrkDT2ZcAq+RRo8veEUijzggky+tgS4PYyJNr63m0g+Lir4PNxBR77dgGg801d3PW5Qiz2KJeU8YY2uvTTvj7zcgLs9eYWivZl9xTlMY8A8spHnOwMgGj1Y8gM90zE801RqDLzgvpI80b6zvW9+Cz01q9A9zv09vaXSmb3bfy295meauGSnD77Kaa88yktyPZu5gz3blb29M8mkvHTixj3025+8XRh3vWwUj1NIBI9D0D/PWaUy72HNzO7ez6TPY6Zdz2LPQs+epvUvBJ36byB+DQ+iHgRvhWEA76jZWK91YAtPsFIDT72XqI94H0wPYRcHTwkBIK9Ag1VPQwHDb2P9Uk9BcS9PISHsjtEErY3CqlLvsNXyb2gjLg5opC0vQGP2jw9Lre8+qBJvVF8wD0hqIc9Lpr+vSLTlrykFHG9jibNPeLF3TwJp/i8Dxw+uVW3BL2oZ6C9q9QBvlBoSz1qCdS9GsOBvRuEqLxc6zo+lz/avIr3GD0dwDc9BCCIPeUIXrzXYai909yQPQpWrb0ieyE+O55wPv1BEb7pBwe+u1P7vdMH37wxaBm+geyUPWkwz7zMT7W9F1PJPQ1JkTw=\u0000", "featureKey": "66c45100bd0ccac7361450684d9c02fb", "path": "ftp://192.168.0.105:8010/faceRegister/9999935_fb196994efec4b6a800b621cc193787b.jpg", "personId": "9999935"}], "result": 1, "success": true}
```

返回值：

包含 faceId, feature, featureKey, path, personId 对象的数据集合

3.2.6 身份证/IC 卡注册接口

URL : /face/icCardRegist

method : POST

参数列表：

参数名	描述	类型	必传	附加说明
pass	接口安全校验密钥	String	Y	
personId	人员 id	String	Y	• 给指定人员 id 注册卡号 • 人员 id 必须为已创建；若不存在该人员 id，则无法进入 IC 卡注册模式 • 注册成功后，可根据 personId 调用人员信

				息查询接口查询该人员的卡号
--	--	--	--	---------------

返回示例：

```
{ "data": { "id": "9999911", "idcardNum": "410822199908221428", "name": " 郝 帅" }, "result": 1, "success": true }
```

返回值：

Person 类集合：

```
public class Person {
    private String id;//人员 id
    private String name;//人员姓名
    private String idcardNum;//IC 卡卡号
}
```

3.2.7 特征注册接口

URL : /face/featureReg

method : POST

参数列表：

参数名	描述	类型	必传	附加说明
pass	接口安全校验密钥	String	Y	
personId	人员 id	String	Y	- 给指定人员 id 创建特征照片 - 人员 id 必须为已创建；若不存在该人员 id，则无法添加
faceId	照片 id	String	Y	- 若 faceId 传入内容为空，则系统会自动生成 faceId 并在照片创建成功后返回 - 照片更新时，以 faceId 为唯一标识进行更新
feature	特征码	String 赋值形式传入	Y	- 可由自动拍照及照片注册结果回调设置接口(3.3.10)接收，也可由照片查询接口(3.2.5)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
featureKey	特征密钥	String 赋值形式传入	Y	- 照片创建成功后，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
--	--	--	--	------

返回示例：

```
{ "data": { "faceId": "f13b417db27240408d2497fc088c033b", "feature": "L04AADB407oMAgAAVms8vNhgGD5l7dm8bkjFPDIWFz7Kuk0+BAgQPZw7KTxi10a+zbWRPLSahz2MJVM84di1PPYCIT509Tm9cf8WPepYT3fubo9MCQZvc06V73iWZm72Z0rvQfRlD3AXOe9X24XP a4rHT4CVdQ9/C6xva+Gdr3lYgM+AGU8PffLHr5pOFq8KjCdPdFxdTwFIk8FdlEvWHGojxJFay9Bo5oPSwylDxYsmC8ttrsPW9ooDlFXF69I7+lvVC+IL7qg5M8BNuuPVi/27tDlhq9/9sBPqH3uTwz82g95mwrPPetQbzW7CS+pDdTPYZKQbtkcFw9PCHHvQ6wnDlvkgi+2r4kvjx8oLwFPz8+/HzjvUey3z0esIU9NPUEvmhZIL5vBrI8nJ8oPV9ZF76IFmo9IgAAvelloLzVPkG+yv0vPd4Yqr2fsa+9asXwPHk0a76gh4i8HpEAPaYaQj0sZPu9EiSYPGUScjwd3Ow9rI+dvYP4jj1K6Ym8kNK1vPN3KL7lZpC9V5GrvW+Kuj0mtKM9v/zxPT9HGD4axVm8HycJvnIKVLxsDLc9TZEYPD5ZHr2hlVI8608cvRWlwd0YakY9txpcPb7h077C8Es96RIAvD4XIL2/niI+fFgaPY4bp7lmzYe9xGY6veWN6bxc5d481Jy3Or2jWb1oslC9zle1PQEU8L0=\u0000", "featureKey": "95ae63cb3280bbf7e2638cfa7c1c6b7a", "path": "", "personId": "14796648ea2f4883ad55210d88093ce7"}, "result": 1, "success": true }
```

返回值：

包含 faceId, feature, featureKey, path, personId 对象的数据集合，其中 path 为空；特征注册并非照片，无路径查询图片

3.2.8 照片人脸相似性比对接口

URL： /photoComparison

method： POST

参数列表：

参数名	描述	类型	必传	附加说明
pass	接口安全校验密钥	String	Y	
img1	照片 1 的 base64 编码	String	Y	不加头部，如：data:image/jpg;base64,
img2	照片 2 的 base64 编码	String	Y	设备会对传入的两张照片进行人脸检测和相似度比对

返回示例：

- 异常：{"msg": "FaceCoreException{expCode=7,expDesc='照片中未检测到人脸', attachedInfo=EXP-FACE-3}", "result": 1, "success": false}
- 正常：{"data": 0.9999999, "result": 1, "success": true}

返回值：

相似度比对分数

3.2.9 人员照片清空接口

URL : /face/deletePerson

method : POST

参数列表 :

参数名	描述	类型	必传	附加说明
pass	接口安全校验密钥	String	Y	
personId	人员片 id	String	Y	调用该接口，该人员 id 的所有注册照片 id 会注销，并同步删除存储在本地（设备内）的注册照文件

返回示例 :

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

返回值 :

通用操作状态

其他说明 :

- {"msg":"**没有这个人的照片**","result":1,"success":false}

若该人员 id 没有注册照片，则会提示“没有这个人的照片”；result:1 表示接口调通，success:false 表示删除操作失败

- {"msg":"**人员不存在**","result":1,"success":false}

若该人员 id 不存在，则会提示“人员不存在”；result:1 表示接口调通，success:false 表示删除操作失败

3.3 设备管理接口

3.3.1 属性配置接口

URL : /setConfig

method : POST

参数列表 :

参数名	描述	类型	必传	附加说明
pass	接口安全校验密钥	String	Y	
config	Outconfig 对象的 Json 格式	Json	Y	• {"companyName":"我的测试", "identifyDistance":1, "identifyScores":80, "saveIdentifyTime":0, "ttsModType":100, "ttsModContent": "欢迎 {name}", "comModType":100, "comModContent": "hello", "displayModType":100, "displayModContent": "{name} 欢迎你", "slogan": "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":"#WG{id}#", "recRank":2} • 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 • 若模式参数 ModType 传入 100，则自定义参数 ModContent 必传；其他参数均为非必传，若不传则为默认值（不会保留上次传入的值）

返回示例 :

```
{ "data": { "comModContent": "hello", "comModType": 100, "companyName": "我的测试", "displayModContent": "{name} 欢迎你", "displayModType": 100, "identifyDistance": 1, "identifyScores": 80, "intro": "智能智能智能智能 \n 真的智能", "multiplayerDetection": 1, "recRank": 2, "recStrangerTimesThreshold": 3, "recStrangerType": 2, "saveIdentifyTime": 0, "slogan": "科技 \n 有限公司", "ttsModContent": "欢迎 {name}", "ttsModType": 100, "ttsModStrangerContent": "陌生人啊你好", "ttsModStrangerType": 100, "ttsModType": 100, "wg": "#WG{id}#", "result": 1, "success": true }
```

返回值 :

```

private String 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
private Int 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
private Int identifyScores;//识别分数，识别命中分数，默认为 80，传入值务必在 60-100 之间
private Int 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
private Int 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
private String 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 32 个字，请自行调整。例：{name}欢迎光临
private Int displayModType;//屏幕显示模式类型，默认 1：显示名字，... 100：自定义
private String 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 32 个字，请自行调整。例：{name}，签到成功！
private Int 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
private String 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 64 个字符，请自行调整。例（韦根输出）：#WG{idcardNum}#或#WG{id}#（其中 idcardNum、id 可配置的范围必须为 1-65535，否则信号输出失效；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）
private String 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 #WG{idcardNum}#或#WG{id}#。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）
private String slogan;//标语，大屏展示
private String intro;//公司简介，大屏展示
private Int 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
private String 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
private String 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
private String 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 32 个字。例：注意陌生人
private Int 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
private Int 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

```

3.3.2 Logo 修改接口

URL : /changeLogo

method : POST

参数列表 :

参数名	描述	类型	必传	附加说明
pass	接口安全校验密钥	String	Y	
imgBase64	logo 图片的 Base64 编码字符串	String	Y	• 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 214×160px, 或等比例适当缩放。尺寸建议不要过大

返回示例 :

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

返回值 :

通用操作状态

3.3.3 设备序列号获取接口

URL : /getDeviceKey

method : POST

参数列表 :

调用该接口不需要传入参数, 不需要传入 pass

返回示例 :

```
{"data":"84E0F420020B00B0","result":1,"success":true}
```

返回值 :

- 设备唯一标识码。每台设备对应唯一不可变的设备序列号
- 设备序列号的前 12 位为设备以太网卡的 MAC 地址, 如 84-E0-F4-20-02-0B

3.3.4 接口密码设置及修改接口

URL : /setPassWord

method : POST

参数列表 :

参数名	描述	类型	必传	附加说明
oldPass	旧密码	String	Y	• 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,

newPass	新密码	String	Y	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
---------	-----	--------	---	---

设置初始密码示例：

POST http://192.168.20.167:8090/setPassWord... Params Send

Authorization Headers (1) Body Pre-request Script Tests

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

Key	Value	Description
oldPass	12345	
newPass	12345	
New key	Value	Description

Body Cookies Headers (6) Tests Status: 200 OK Time: 79 ms

Pretty Raw Preview Text

```
1 {"data":"passWord is:12345","result":1,"success":true}
```

返回示例：

```
{"data":"passWord is:12345","result":1,"success":true}
```

返回值：

密码设置或修改状态，成功后将会返回成功设置的密码

3.3.5 设置识别回调接口

URL : /setIdentifyCallback

method : POST

参数列表：

参数名	描述	类型	必传	附加说明
pass	接口安全校验秘钥	String	Y	
callbackUrl	外部服务接口	String	Y	- 给设备设置一个外部回调地址 - 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 - 设置 saveIdentifyTime（识别记录时间窗）后，在该时间间隔内，只会向回调地址 POST 一条识别记录 - 外部接口的 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\\&%_\\./~]*)?"; • 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
--	--	--	--	---

返回示例：

```
{ "data": "http://www.baidu.com", "result": 1, "success": true }
```

返回值：

设置成功后会将接口地址返回

回调示例：

ContentResult 里封装了 result 和 success 字段，回调成功则返回 {"result":1,"success":ture}，详情见文末附录。

```
//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);
}
```

回调数据示例：**• 注册人员**

```
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

```
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"}
```

其他说明：

- 回调地址接口接收到识别数据后，需返回{"result": 1,"success":ture}给设备，设备解析到{"result": 1,"success":ture}，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
- 设备每间隔 10 分钟，从数据库中读取回调失败（state:0）的识别记录，再次向回调地址发送请求，请求成功则更新数据库中对应的回调状态（state:1）
- 若设备未设置回调，则默认标记识别记录中的字段 state:1
- 回调参数字段说明：

参数名	类型	描述
ip	String	设备 IP 地址
deviceKey	String	设备唯一标识码
personId	String	人员 id，陌生人为 STRANGERBABY
time	String	时间戳
type	String	识别方式（face/card）、识别出的人员类型；face_0（表示是人脸识别，且该人员在 passtime 权限时间内），card_1(表示是刷卡识别，且该人员在 passtime 权限时间外)，face_2（表示是人脸识别，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，且请求访问的客户端与设备处于局域网同一网段

3.3.6 修改屏幕方向接口

URL : /setScreenOrt

method : POST

参数列表：

参数名	描述	类型	必传	附加说明
pass	接口安全校验密钥	String	Y	
orientation	屏幕方向	Int	Y	- 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 - 二代设备目前只支持竖屏，此配置对二代设备无效

返回示例：

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

返回值：

接口调用成功后会将设置的 Int 值返回

3.3.7 网络信息配置接口

URL : /setNetInfo

method : POST

参数列表：

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

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,"success":true}
```

• 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":"192.168.1.1","ip":"192.168.1.22","subnetMask":"255.255.255.0"},"msg":"配置成功，请 5 秒钟后重启设备。","result":1,"success":true}
```

返回值：

设置成功后会将设置值返回

3.3.8 大屏文字展示信息配置接口（已弃用，请使用 3.3.1 属性配置接口）

URL : /setBigScreenInfo

method : POST

参数列表：

参数名	描述	类型	必传	附加说明
pass	接口安全校验密钥	String	Y	
info	编辑文本内容	String	Y	- 仅大屏模式下展示 - 大屏可编辑文字区域在第 9 页图中可见 - 传入为空，则显示默认文字“人脸识别系统”

返回示例：

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

返回值：

通用操作状态

3.3.9 设备重启接口

URL : /restartDevice

method : POST

参数列表：

参数名	描述	类型	必传	附加说明
pass	接口安全校验密钥	String	Y	

返回示例：

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

返回值：

通用操作状态

3.3.10 自动拍照及照片注册结果回调设置接口

URL : /setImgRegCallBack

method : POST

参数列表：

参数名	描述	类型	必传	附加说明
pass	接口安全校验密钥	String	Y	
url	接口全路径地址	String	Y	- 给设备设置一个外部回调地址 - 自动拍照或照片注册成功后，设备会回调该接口，将自动拍照照片的 URL 回调给该接口，外部可以通过该 URL 访问到当前设备中的注册照片 - 照片注册成功后，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 - 外部接口的 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\\&%_\\./~]*)?"; - 传入内容为空可以清空回调地址，清空后将不再进行回调

返回示例：

```
{"data":"http://www.baidu.com","result":1,"success":true}
```

返回值：

设置成功后会将接口地址返回

其他说明：

POST 字段内容包含：

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	照片本地路径
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

3.3.11 设备心跳配置接口

URL： /setDeviceHeartBeat

method： POST

参数列表：

参数名	描述	类型	必传	附加说明
pass	接口安全校验密钥	String	Y	
url	外部接收设备心跳监听的服务器接口地址	String	Y	- 给设备设置一个外部回调地址 - 设备每隔一分钟会向该接口 POST 字段 deviceKey、time、ip、personCount、faceCount 和 version - 外部接口的 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_\\-\\/\\~*])?"; - 传入内容为空可以清空回调地址，清空后将不再进行回调

返回示例：

```
{"data":"http://www.baidu.com","result":1,"success":true}
```

返回值：

设置成功后会将接口地址返回

其他说明：

POST 字段内容包含：

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

3.3.12 设备时间设置接口

URL : /setTime

method : POST

参数列表 :

参数名	描述	类型	必传	附加说明
pass	接口安全校验密钥	String	Y	
timestamp	Unix 毫秒级时间戳	String	Y	- 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 - 若设备连入公网，设备本身有网络时间校对机制，每隔 1 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

返回示例 :

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

返回值 :

通用操作状态

3.3.13 设备重置接口

URL : /device/reset

method : POST

参数列表 :

参数名	描述	类型	必传	附加说明
pass	接口安全校验密钥	String	Y	
delete	选择性删除	Boolean	Y	- 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，并删除通过属性配置接口设置的属性

				传入 false，删除以上所有信息，但不删除通过属性配置接口已经设置的属性
--	--	--	--	---

返回示例：

```
{"msg":"完成清除","result":1,"success":true}
```

返回值：

通用操作状态

其他说明：

- delete 字段传入 true，设备清空数据库，恢复 DHCP 自动获取 IP 模式，恢复横屏模式；设备自动重启
- 物理重置：设备背面的 RESET 小孔长按 8 秒；设备清空数据库，恢复出厂设置，功能同 delete 字段传入 true

3.3.14 设备开门接口**URL : /device/openDoorControl****method : POST****参数列表：**

参数名	描述	类型	必传	附加说明
pass	接口安全校验密钥	String	Y	

返回示例：

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

返回值：

通用操作状态

其他说明：

- 密码验证正确，设备就会控制开启门禁
- 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

3.3.15 大屏显示模式接口**URL : /device/setScreenMode****method : POST****参数列表：**

参数名	描述	类型	必传	附加说明
pass	接口安全校验密钥	String	Y	
screenMode	选择大屏显示模式	Int	Y	1：传统模式，通过属性配置接口(3.3.1)配置

				2: 自定义大屏背景模式，通过大屏背景图片配置接口(3.3.16)配置
--	--	--	--	-------------------------------------

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, "success": true}
```

返回值：

设置成功的屏幕模式

3.3.16 大屏背景图片配置接口

URL : /device/setActScreenImg

method : POST

参数列表：

参数名	描述	类型	必传	附加说明
pass	接口安全校验密钥	String	Y	
img	图片的 base64 编码字符串	String	Y	已限制必须分辨率小于 1920×1080

返回示例：

```
{"data": "base64 编码", "msg": "修改背景图片成功", "result": 1, "success": true}
```

返回值：

设置成功的图片 base64 编码

其他说明：

- 若使用了大屏背景图片配置接口配置图片成功，则属性配置接口(3.3.1)中的参数 slogan(标语)、intro(简介)、companyName(设备名称)、logo 不会在大屏上显示
- 需先将大屏显示模式接口(3.3.15)设置成 2：自定义大屏背景模式，设置成功的图片才会生效

3.3.17 Wi-Fi 配置接口

URL : /device/setWifi

method : POST

参数列表：

参数名	描述	类型	必传	附加说明
pass	接口安全校验密钥	String	Y	
wifiMsg	无线配置信息	Json	Y	{"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"}

返回示例：

```
{"msg":"配置 Wifi 开始","result":1,"success":true}
```

返回值：

通用操作状态

其他说明：

wifiMsg 类集合

说明：

```
public class wifiMsg {  
    private String ssid;//Wi-Fi 名称，必传。不允许特殊字符  
    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  
}
```

其他说明：

设备硬件需配有 Wi-Fi 模块才能支持 Wi-Fi 连接

3.4 识别记录管理接口

3.4.1 识别记录查询接口

URL : /findRecords

method : GET

参数列表 :

参数名	描述	类型	必传	附加说明
pass	接口安全校验密钥	String	Y	
personId	人员 id	String	Y	- 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	每页显示数据数量	Int	Y	- 传入-1 为不分页 - 若不传-1, 请务必大于 0
index	查询页码	Int	Y	页码, 从 0 开始
startTime	记录开始时间	Date	Y	- 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

返回示例:

```
{ "data": { "pageInfo": { "index": 0, "length": 20, "size": 0, "total": 3 }, "records": [ { "id": 1, "path": "ftp://192.168.20.222:8010/recordsImg/2017-11-29/cc3254898b5e4d3e969d9767284d7b93_1511927588123.jpg", "personId": "cc3254898b5e4d3e969d9767284d7b93", "state": 1, "time": 1511927588123, "type": 1 }, { "id": 2, "path": "ftp://192.168.20.222:8010/recordsImg/2017-11-29/cc3254898b5e4d3e969d9767284d7b93_1511928417440.jpg", "personId": "cc3254898b5e4d3e969d9767284d7b93", "state": 1, "time": 1511928417440, "type": 0 }, { "id": 3, "path": "ftp://192.168.20.222:8010/recordsImg/2017-11-29/STRANGERBABY_1511928451042.jpg", "personId": "STRANGERBABY", "state": 1, "time": 1511928451042, "type": 2 } ] }, "result": 1, "success": true }
```

返回值 :

IdentifyRecords 类集合：

```
public class IdentifyRecords{
    private String id;//识别记录 id，由设备自动生成并分配
    private String personId;//人员 id，陌生人/识别失败显示 STRANGERBABY
    private String path;//现场照保存路径，保存在设备中，当设备内存存储现场照满 3G 时，会自动删除较早的 1G 现场照，设备一共 16G；若需要请及时存储
    private Int state;//回调结果，0：回调失败，1：回调成功或未设置回调地址
    private long time;//识别出人员的时间，Unix 时间戳
    private Int type;//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
}
```

3.4.2 识别记录删除接口

URL : /deleteRecords

method : POST

参数列表：

参数名	描述	类型	必传	附加说明
pass	接口安全校验密钥	String	Y	
time	时间节点	String	Y	- 删除所传日期时间点前的所有人脸识别记录 - 传入时间格式为（年-月-日 时:分:秒）： 2017-07-15 12:05:00

返回示例：

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

返回值：

删除识别记录的数量

3.4.3 IC 卡识别记录查询接口

URL : /findICRecords

method : GET

参数列表：

参数名	描述	类型	必传	附加说明
pass	接口安全校验密钥	String	Y	
personId	人员 id	String	Y	- 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
length	每页显示数据数量	Int	Y	- 传入-1 为不分页 - 若不传-1，请务必大于 0

index	查询页码	Int	Y	页码，从 0 开始
startTime	记录开始时间	Date	Y	- 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	

返回示例:

```
{
  "data": {
    "pageInfo": {
      "index": 0,
      "length": 20,
      "size": 0,
      "total": 10
    },
    "records": [
      {
        "id": 1,
        "name": "test",
        "personId": "1218bc9c9d734ab4a36efa9343cf792e",
        "state": 1,
        "time": 1511945288801,
        "type": 0
      },
      {
        "id": 2,
        "name": "test",
        "personId": "1218bc9c9d734ab4a36efa9343cf792e",
        "state": 1,
        "time": 1511945532971,
        "type": 0
      },
      {
        "id": 3,
        "name": "test",
        "personId": "1218bc9c9d734ab4a36efa9343cf792e",
        "state": 1,
        "time": 1511945683074,
        "type": 0
      }
    ]
  },
  "result": 1,
  "success": true
}
```

返回值：

ICCardRecords 类集合：

```
public class ICCardRecords {
    private String id; // 识别记录 id，由设备自动生成并分配
    private String name; // 人员姓名
    private String personId; // 人员 id
    private Int state; // 回调结果，0：回调失败，1：回调成功或未设置回调地址
    private long time; // 识别出人员的时间，Unix 时间戳
    private Int type; // 识别出的人员类型，0：时间段内，1：时间段外
}
```

3.4.4 识别记录删除接口（通过毫秒级时间戳进行删除）**URL : /deleteRecordsByUnixTime****method : POST****参数列表：**

参数名	描述	类型	必传	附加说明
pass	接口安全校验密钥	String	Y	
unixTime	Unix 毫秒级时间戳	String	Y	- 删除所传参数时间戳之前的所有人脸识别记录 - 传入时间格式为：1507773661000

返回示例：

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

返回值：

删除识别记录的数量

3.4.5 刷卡记录删除接口

URL : /deleteICRecords

method : POST

参数列表 :

参数名	描述	类型	必传	附加说明
pass	接口安全校验秘钥	String	Y	
time	时间节点	String	Y	- 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

返回示例 :

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

返回值 :

删除识别记录的数量

3.4.6 刷卡记录删除接口（通过毫秒级时间戳进行删除）

URL : /deleteICRecordsByUnixTime

method : POST

参数列表 :

参数名	描述	类型	必传	附加说明
pass	接口安全校验秘钥	String	Y	
unixTime	Unix 毫秒级时间戳	String	Y	- 删除所传参数时间戳之前的所有刷卡识别记录 - 传入时间格式为：1507773661000

返回示例 :

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

返回值 :

删除识别记录的数量

附录 1(msg 信息及异常码)

msg 信息

"请输入正确格式的 url 地址"
"参数异常"
"请使用 POST 请求"
"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。"
"请确认输入的网络配置格式是否正确。"
"照片 ID 不存在, 请先调用照片创建接口。"
"图片解析异常"
"未查询到该人,请先添加该人员。"
"重启后即改变方向。"
"base64 图片解析异常"
"完成清除"
"设置成功"
"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"
"接口服务未设置密码, 请先设置密码"
"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"
"参数异常, 页码及长度信息请务必为数字"
"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"
"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。"
"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"

异常码

EXP_DESC_7	照片中未检测到人脸
EXP_DESC_8	照片中包含多个人脸
EXP_DESC_9	无法从照片中提到特征

EXP_DESC_10	提取特征异常
EXP_DESC_16	人脸框侧脸角度大于 15 度
EXP_DESC_17	人脸框面积(w*h)大小占图片面积(w*h)大小的 8%以下
EXP_DESC_18	人脸倾斜度大于 15 度

设备异常

-990: "未发现相机"

-991: "相机不可用"

附录 2(回调示例详情)

```
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;
    }

    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 boolean isSuccess() {  
    return this.result == 1;  
}
```

```
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);
}
```