版本说明

日期	版本	描述	作者
2020-06-15	V1.4.8	更新	方

文档说明

在使用接口时，请先阅读以下内容，方便后续理解；若已阅读，可直接进入接口定义章节

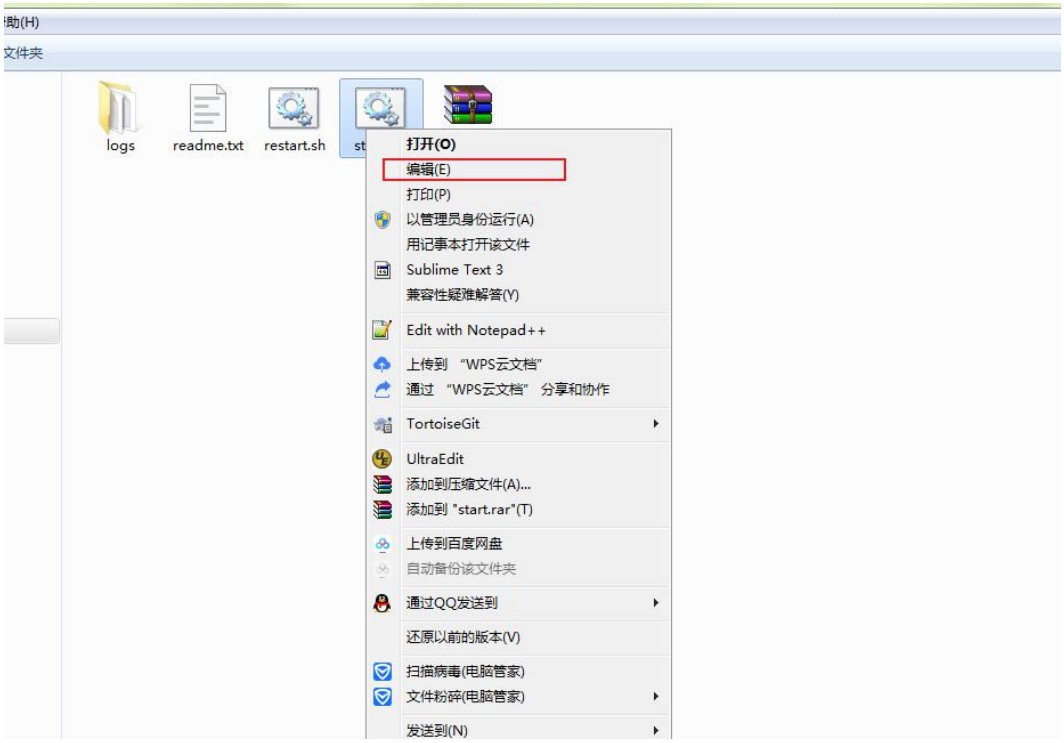
接口指南

服务器准备

1. 需开放 **10010** 、**10011** 、**8190** 端口
2. 支持 **windows** 、**Linux** 系统
3. 需安装 Java 运行环境 **JDK 1.8**

服务启动

1. windows 系统下需要修改 **start.bat** 文件的 jdk 路径，如：**set JAVA_EXE=D:"Program Files"\Java\jre8\bin\java**




```
@echo off
echo [Pre-Requirement] Makesure install JDK 8.0+ and set the JRE_HOME.

set JAVA_EXE=D:\Java\jdk1.8.0_102\jre\bin\java
set JAVA_OPTS=%JAVA_OPTS% -server -Xmx2g -XX:MetaspaceSize=128m -Dfile.encoding=UTF-8
set SERVER_NAME=tdx-face-sdk-server_release.jar

echo [Step] start application.
echo "%JAVA_EXE% %JAVA_OPTS% -jar %SERVER_NAME%"
start "TdxFaceSdkServer" %JAVA_EXE% %JAVA_OPTS% -Dlogging.file=logs\stdout.%date:~0,4%
%date:~5,2%%date:~8,2%.log -jar %SERVER_NAME%
if errorlevel 1 goto error

echo [INFO] Please wait a moment. When you see "[INFO] Started TdxFaceSdkServerApplication
in xxx seconds" in console, you can access below api:
echo [INFO] http://localhost:8190/api/xxx

goto end
:error
echo Error Happen!!
:end
pause
```

修改为本机JDK路径

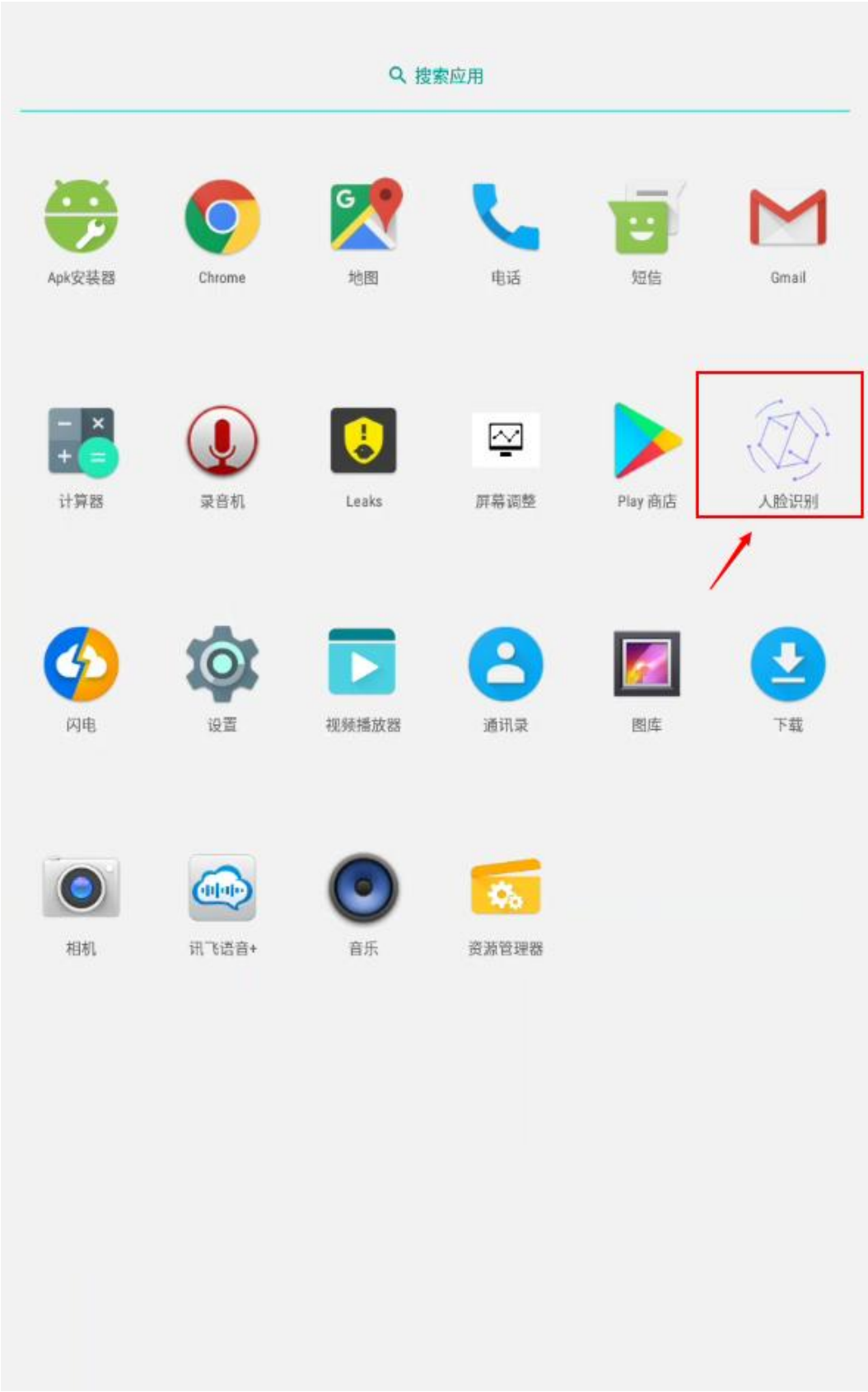
2. 运行 **start.bat**

[点击下载中间件]

设备云端配置

在与云端通信前，先要在人脸设备的 Web 界面中进行服务器配置。

注意： web 配置必须先运行 **人脸识别 App**。



1. 输入 Web 配置地址

打开电脑浏览器，输入地址：http://设备局域网 IP:8090。

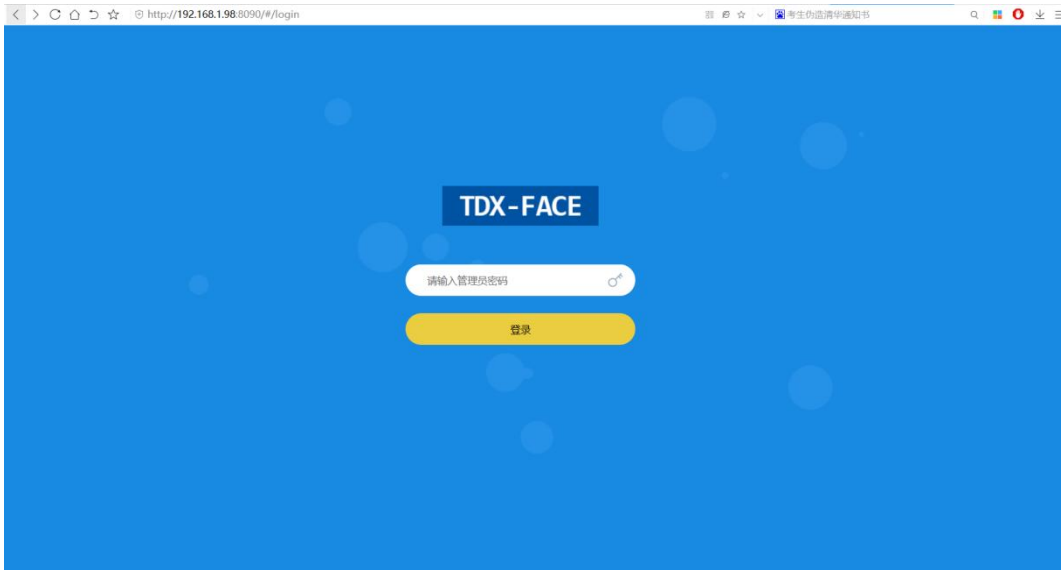
设备局域网 IP： 人脸识别 App 运行后，识别界面底部有 IP 信息。

打开正常如下图：

2. 登录

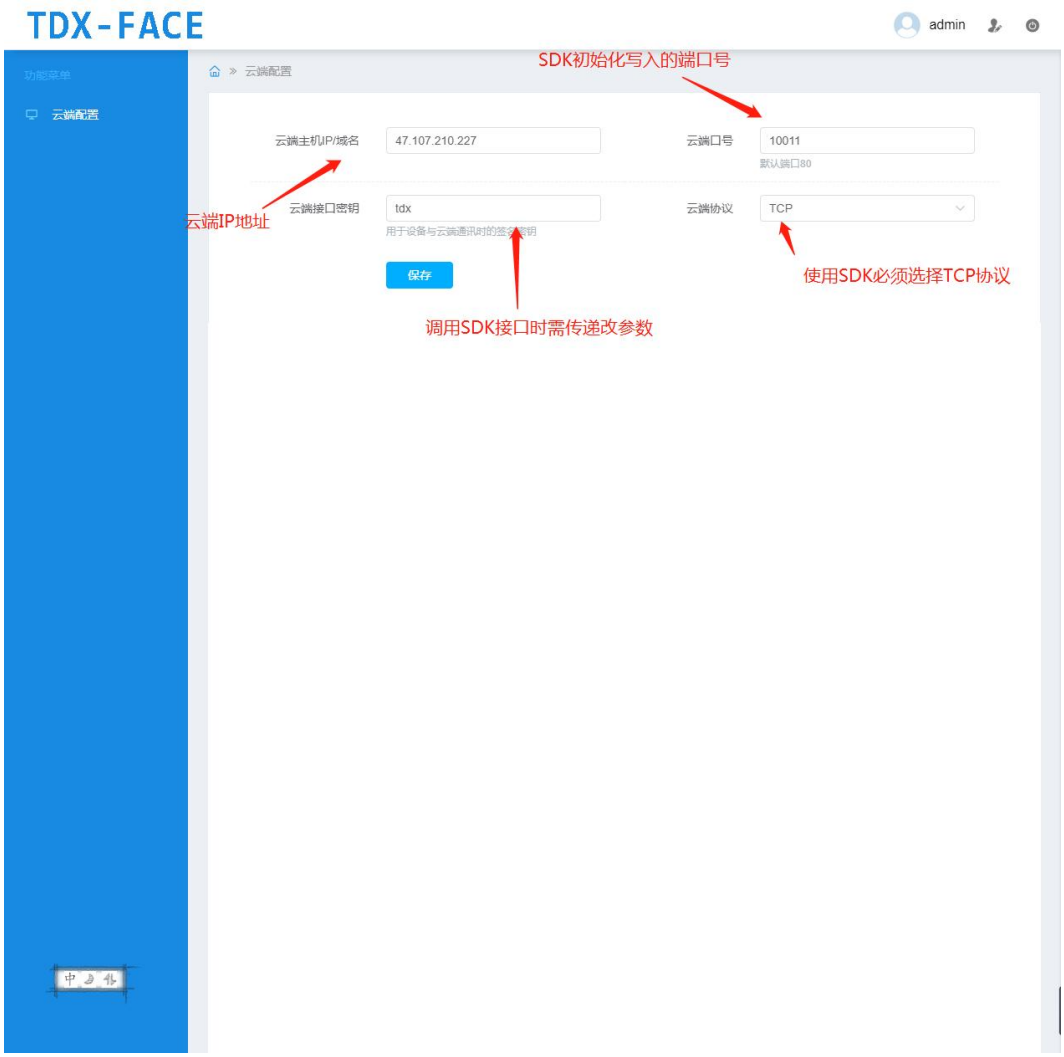
输入默认密码 **123456** 或自定义密码（如设备重置后自定义密码将被删除，请使用默认密码）。

点击登录



3. 配置云端参数

- 请先选择**云端协议**为：**TCP**
- 如需与云端通信请配置好 **服务器 IP**、**服务器端口号**、**服务器密钥** 等参数，如需取消与服务端通信则删除参数配置或重置设备。
- **说明：服务器密钥为与云端通信时进行接口签名验证的必要参数，需记录下来用于 SDK 接口调用时传递参数。**



4. 保存配置重启设备

接口规范

- **接口根地址：http://sdk 中间件服务的 IP:8190/api/**
- **接口形式：**通过转换 TCP 为 HTTP 方式对外提供服务
- **字符编码：**字符编码全部使用 **UTF-8**
- **请求头：ContentType：application/x-www-form-urlencoded**

接口返回

参数名	数据类型	说明
result	int	接口是否调通，1 成功，0 失败
success	boolean	操作是否成功，成功为 true，失败为 false
code	String(16)	返回码，对照表见附录说明
msg	String(128)	返回信息
data	Object	返回的数据对象

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。

接口列表

1. 设备配置

请求地址：**http://SDK 中间件服务器 IP:8190/api/device/config**

请求方法：**POST**

请求数据：

参数名	数据类型	必填	说明
deviceKey	String	是	设备序列号
secret	String	是	设备密码
comModContent	String	是	串口输出模式自定义内容
comModType	Int	是	串口输出模式
delayTimeForCloseDoor	Int	是	继电器控制开门到关门的时间间隔，单位 ms
displayModContent	String	是	识别文字显示模式自定义内容

displayModType	Int	是	识别文字显示模式
identifyDistance	Int	是	识别距离
saveIdentifyTime	Int	是	识别间隔
identifyScores	Int	是	识别分数
multiplayerDetection	Int	是	多个人脸检测设置
recRank	Int	是	识别等级
recStrangerTimesThreshold	Int	是	陌生人判定次数
recStrangerType	Int	是	陌生人开关（是否进行陌生人识别）
ttsModContent	String	是	语音播报模式自定义内容
ttsModType	Int	是	语音播报模式
ttsModStrangerType	Int	是	陌生人语音播报模式
ttsModStrangerContent	Sring	是	陌生人语音播报模式自定义内容
wg	String	是	韦根类型及输出
whitelist	Int	否	身份证比对白名单开关，1：关，2：开
saveIdentifyMode	Int	否	保存识别记录模式，0：不续传，1：续传

参数示例 (请根据实际情况调整)：

"comModContent": "hello", // String，串口输出模式自定义内容

"comModType": 100, // Int，串口输出模式

"companyName": "我的测试", // String，公司名称，显示位置参见设备屏幕

"delayTimeForCloseDoor": 500, // Int，继电器控制开门到关门的时间间隔，单位 ms

"displayModContent": "{name}欢迎你", // String，识别文字显示模式自定义内容

"displayModType": 100, // Int，识别文字显示模式

"identifyDistance": 1, // Int，识别距离，0：无限制，1：0.5 米以内，2：1 米以内，3：1.5 米以内，4：3 米以内

"saveIdentifyTime": 3, // Int，识别间隔

"identifyScores": 80, // Int，识别分数

"multiplayerDetection": 1, // Int，多个人脸检测设置

"recRank": 2, // Int，识别等级- 等级 1：不开启活体识别;等级 2：开启单目活体识别;等级 3：开启双目活体识别(红外)，识别距离最远为 1.5 米

"recStrangerTimesThreshold": 3, // Int，陌生人判定次数

"recStrangerType": 2, // Int，陌生人开关（是否进行陌生人识别）

"ttsModContent": "欢迎{name}", // String，语音播报模式自定义内容

"ttsModStrangerContent": "陌生人啊你好", // String，陌生人语音播报模式自定义内容

"ttsModStrangerType": 100, // Int，陌生人语音播报模式

"ttsModType": 100, // Int，语音播报模式

"wg": "#WG{id}#", // String，韦根类型及输出

"whitelist": 1, // Int, 身份证比对白名单开关，1：关，2：开

"saveIdentifyMode": 1, // 保存识别记录模式，0：不续传，1：续传

请求说明：

- ttsModType **语音播报模式**、 ttsModContent **语音播报模式自定义内容**：
  - 设备成功识别人员后，默认 1
  - 1：不播报语音
  - 2：播报名字

- 100：自定义
  - 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。
  - 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。
- 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，设备会播报 “姓名权限不足”。

- displayModType **识别文字显示模式**、displayModContent 识别文字**显示模式自定义内容**：
  - 设备成功识别人员后，默认 1
  - 1：显示名字
  - 100：自定义
    - 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！
  - 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” 。

- comModType **串口输出模式**、 comModContent **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。
  - 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
  - 2：不输出
  - 3：韦根信号输出人员 ID
  - 4：韦根信号输出身份证/IC 卡号

- recStrangerType 陌生人开关：
  - 设备默认 1
  - 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）不会识别
  - 2：识别陌生人
  - 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。
- ttsModStrangerType 陌生人语音播报模式、 ttsModStrangerContent 陌生人语音播报 自定义内容：
  - 设备识别到陌生人后， 默认 1
  - 1：不播报语音
  - 2：语音播报 “陌生人警报”
  - 100：自定义
    - 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。
    - 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。
- recStrangerTimesThreshold 陌生人判定：
  - 设备判定人脸为陌生人的识别失败次数，默认 3；
  - 传入值请选择 3-10 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。
- identifyDistance 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。
  - 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。

- 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。

- saveIdentifyTime **识别间隔（秒）**：
  - 设备对同一人脸的重复识别时间间隔。
  - 默认 3 秒，最大 60 秒。

- identifyScores **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 75，要求传入值为 60-100 的整数，分数越高，识别准确率越高，但识别速度会变慢。
  - 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。
  - 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”。

- recRank **识别等级**：
  - 默认等级 2
  - 等级 1：不开启活体识别
  - 等级 2：开启单目活体识别
  - 等级 3：开启双目活体识别，识别距离最远为 1.5 米

- multiplayerDetection 多个人脸检测设置：
  - 设备默认 1
  - 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）
  - 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

- delayTimeForCloseDoor **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 500-25500，单位为 ms。
- 根据使用的场景，选择开门到关门之间的时间间隔。

- whitelist **身份证白名单开关：**
  - 添加人证比对白名单开关，1：关，2：开, 默认 1。
  - 若打开，读取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；若不存在，则提示权限不足。
  - 若关闭，读取身份证后直接进行人证比对流程。
- saveIdentifyMode **识别记录回调模式**
  - 0: 不续传, 1:续传, 默认 1

响应数据：

```
{
  "data": {
    "comModContent": "hello", // String，串口输出模式自定义内容
    "comModType": 100, // Int，串口输出模式
    "companyName": "我的测试", // String，公司名称，显示位置参见设备屏幕
    "delayTimeForCloseDoor": 500, // Int，继电器控制开门到关门的时间间隔，单位 ms
    "displayModContent": "{name}欢迎你", // String，识别文字显示模式自定义内容
    "displayModType": 100, // Int，识别文字显示模式
    "identifyDistance": 1, // Int，识别距离，0：无限制，1：0.5 米以内，2：1 米以内，3：1.5 米以内，4：3 米以内
    "saveIdentifyTime": 3, // Int，识别间隔
```

"identifyScores": 80, // Int，识别分数

"multiplayerDetection": 1, // String，多个人脸检测设置

"recRank": 2, // String，识别等级- 等级 1：不开启活体识别;等级 2：开启单目活体识别;等级 3：开启双目活体识别(红外)，识别距离最远为 1.5 米

"recStrangerTimesThreshold": 3, // String，陌生人判定

"recStrangerType": 2, // String，陌生人开关（是否进行陌生人识别）

"ttsModContent": "欢迎{name}", // String，语音播报模式自定义内容

"ttsModStrangerContent": "陌生人啊你好", // String，陌生人语音播报模式自定义内容

"ttsModStrangerType": 100, // Int，陌生人语音播报模式

"ttsModType": 100, // Int，语音播报模式

"wg": "#WG{id}#", // String，韦根类型及输出

"whitelist": 1, // Int, 身份证比对白名单开关，1：关，2：开

"saveIdentifyMode": 1, // 保存识别记录模式，0：不续传，1：续传

},

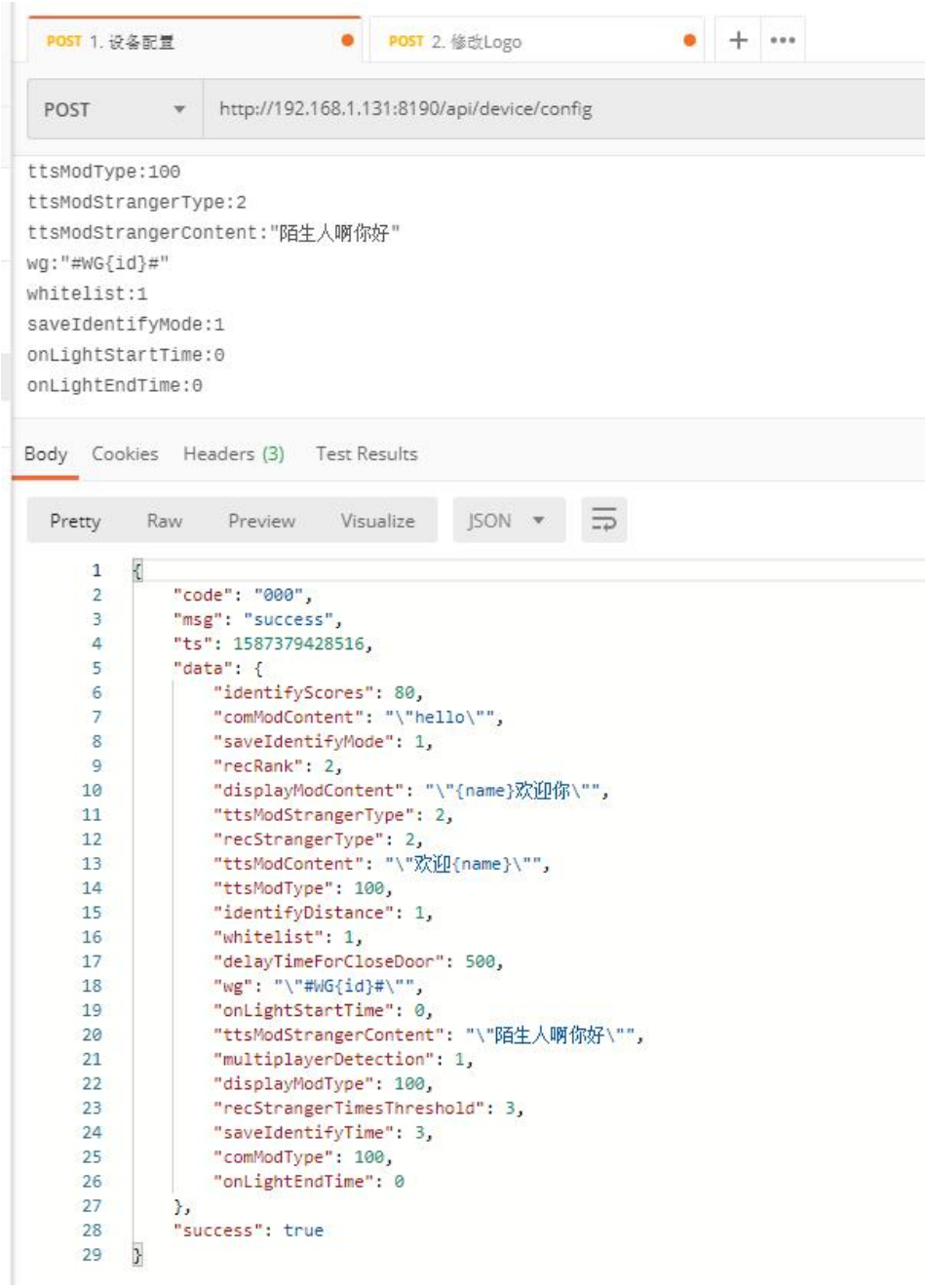
"code": "000",

"msg": "success",

"success": true //成功

}

PostMan 示例：



2. 修改 Logo

请求地址：**http://SDK 中间件服务器 IP:8190/api/device/setLogo**

请求方法：**POST**

请求数据：

参数名	数据类型	必填	说明

deviceKey	String	是	设备序列号
secret	String	是	设备密码
imgBase64	String	是	Logo 图片的 Base64 编码字符串

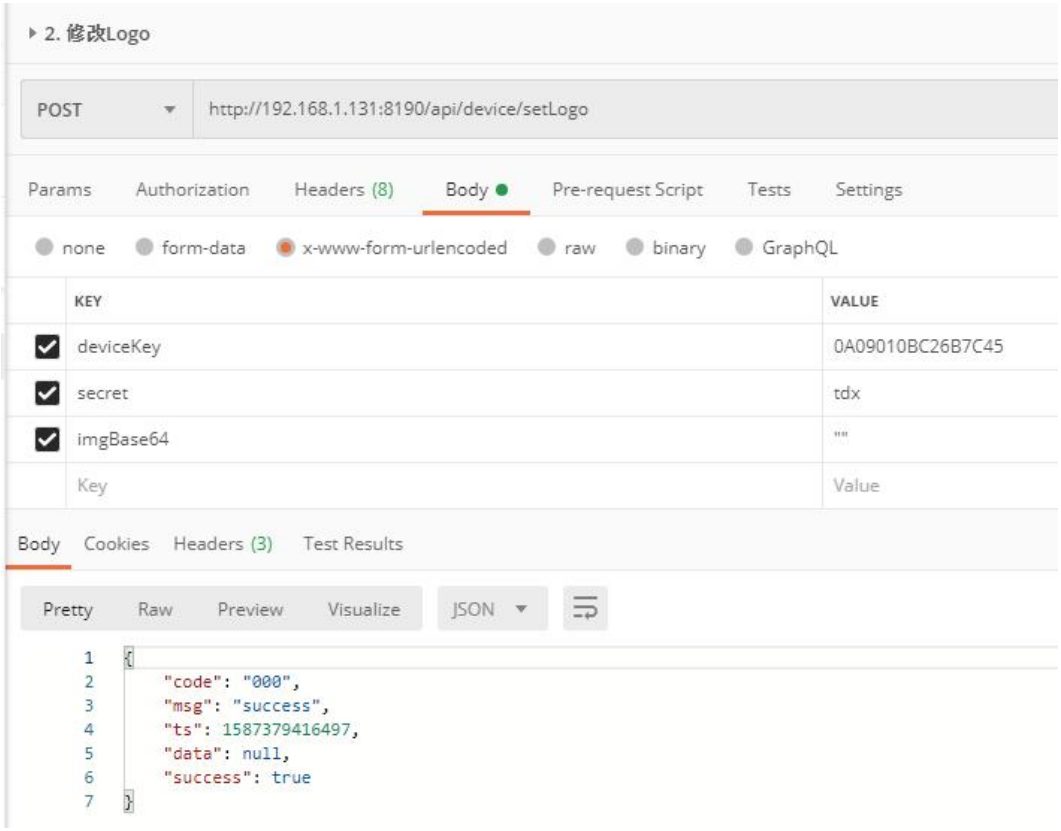
请求说明：

- 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 图片格式仅支持 JPG 和 PNG。
- Logo 大小尽量控制在 500K 以内。

响应数据：

```
{
  "code": "000",
  "msg": "success",
  "success": true //成功
}
```

PostMan 示例：



3. 设置设备时间

请求地址：**http://SDK 中间件服务器 IP:8190/api/device/setTime**

请求方法：**POST**

请求数据：

参数名	数据类型	必填	说明
deviceKey	String	是	设备序列号
secret	String	是	设备密码
timestamp	String	是	Unix 毫秒级时间戳（13 位）

请求说明：

- 配置成功后，设备时间即被改为当前所设置的时间
- 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

- 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。

响应数据：

{

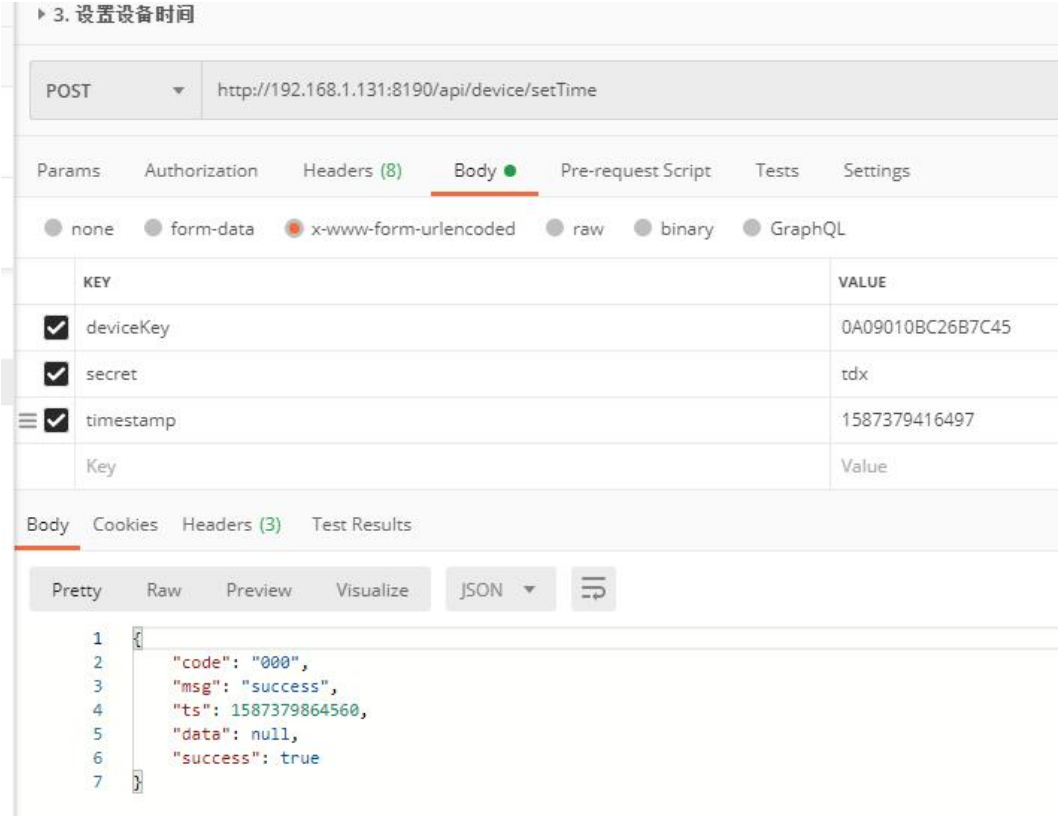
"code": "000",

"msg": "success",

"success": true //成功

}

PostMan 示例：



4. 设备重启

请求地址：**http://SDK 中间件服务器 IP:8190/api/device/reboot**

请求方法：**POST**

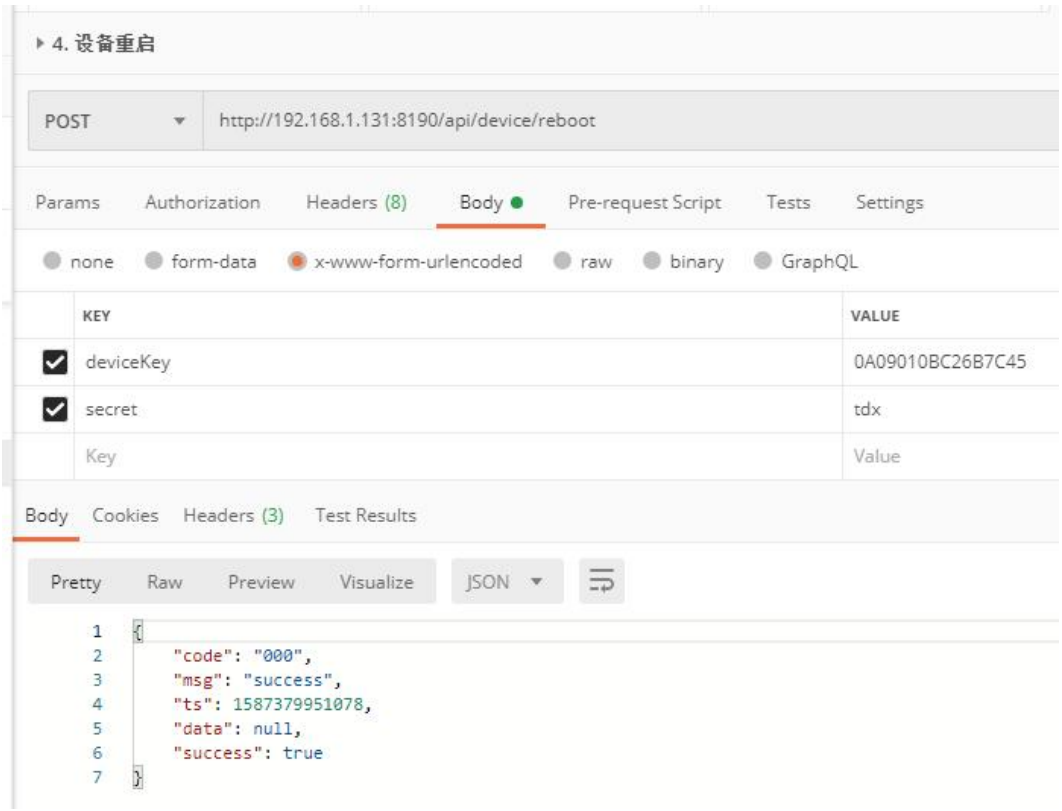
请求数据：

参数名	数据类型	必填	说明
deviceKey	String	是	设备序列号
secret	String	是	设备密码

响应数据：

```
{
  "code": "000",
  "msg": "success",
  "success": true //成功
}
```

PostMan 示例：



5. 设备重置

请求地址：**http://SDK 中间件服务器 IP:8190/api/device/reset**

请求方法：**POST**

请求数据：

参数名	数据类型	必填	说明
deviceKey	String	是	设备序列号
secret	String	是	设备密码

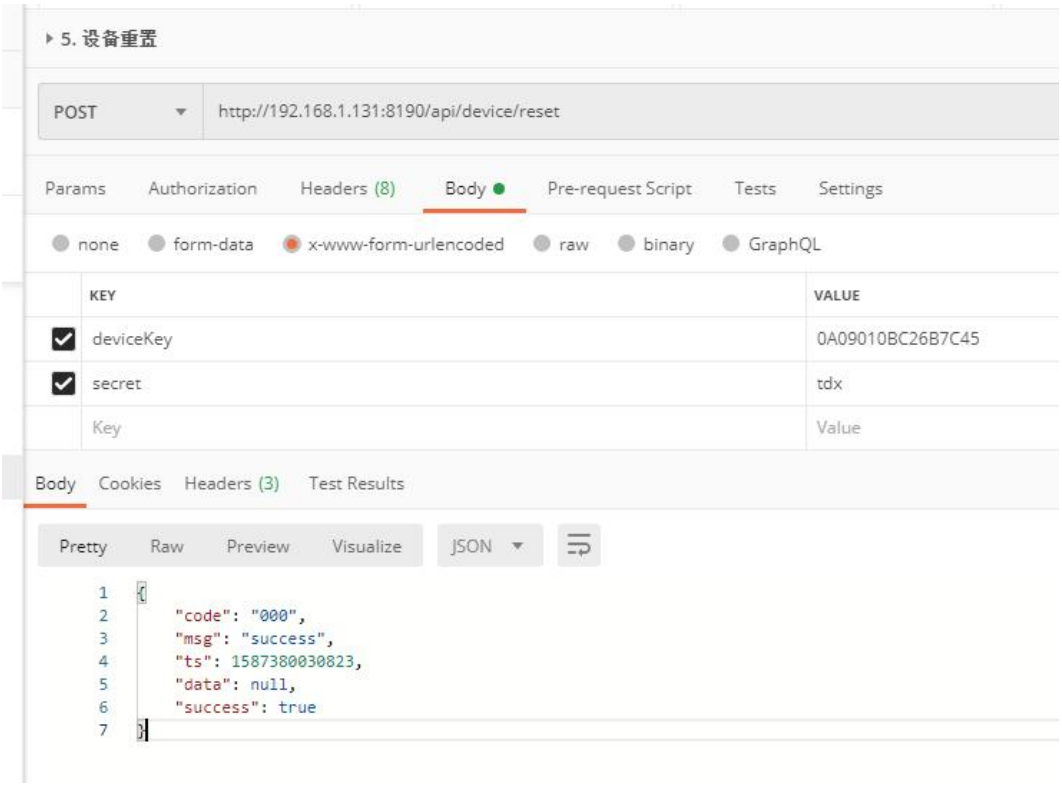
请求说明：

- 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
- 删除通过设备配置接口设置的属性

响应数据：

```
{

  "code": "000",

  "msg": "success",

  "success": true //成功

}
```

PostMan 示例：



6. 设备开门

请求地址：**http://SDK 中间件服务器 IP:8190/api/device/openDoor**

请求方法：**POST**

请求数据：

参数名	数据类型	必填	说明
deviceKey	String	是	设备序列号
secret	String	是	设备密码
type	Int	否	开门类型 **(v14163+ 支持)**
content	String	否	输出内容 **(v14163+ 支持)**

请求说明：

- 密码验证正确，设备就会控制开启门禁
- type

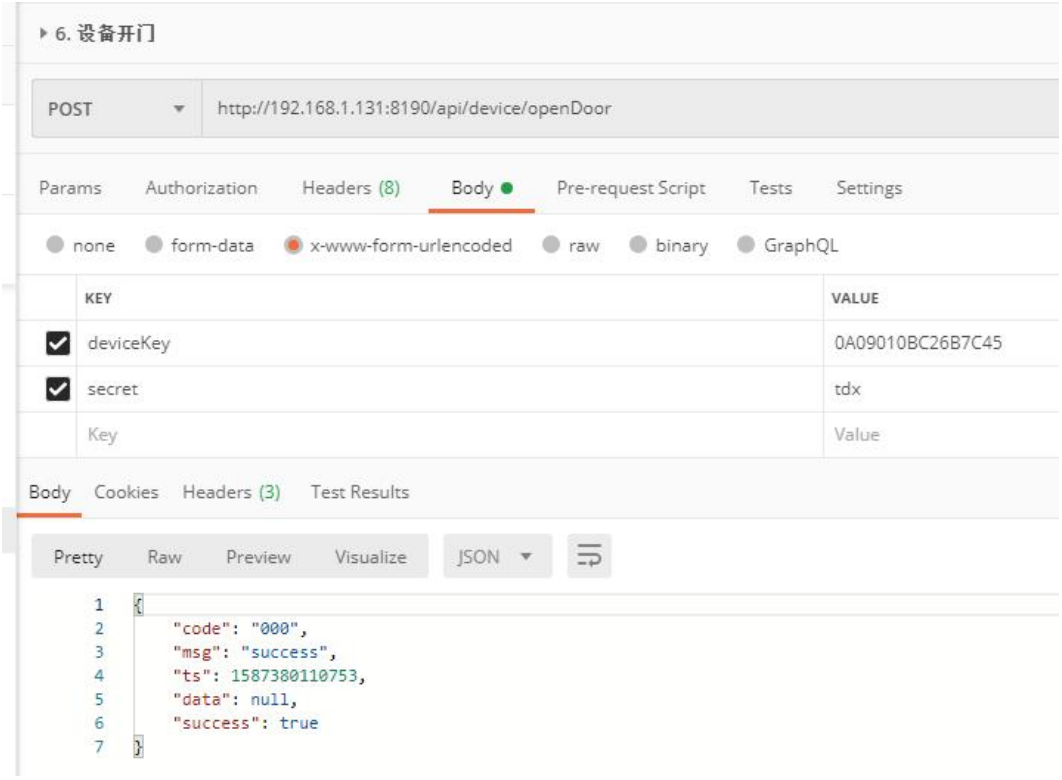
- 1：开门（默认）
- 2：串口输出
- 3：韦根输出

- content
  - type=1：开门
  - type=2：串口输出，允许 255 位任意字符
  - type=3：韦根输出，只允许数字

响应数据：


```
{
  "code": "000",
  "msg": "success",
  "success": true //成功
}
```


PostMan 示例：



7. 识别回调配置

请求地址：**http://SDK 中间件服务器 IP:8190/api/device/setIdentifyCallback**

请求方法：**POST**

请求数据：

参数名	数据类型	必填	说明
deviceKey	String	是	设备序列号
secret	String	是	设备密码
url	String	是	回调地址

请求说明：

- 回调 URL 需要符合正则表达式：**((http|https)://)((a-zA-Z0-9\._-{2,6})|(0-9{1,3}\.0-9{1,3}))(:0-9*)?**
- 设备只保存 10 天的的识别记录，10 天之前的识别记录设备会定时清理。

响应数据：

{

"data": "http://www.baidu.com",

"code": "000",

"msg": "success",

"success": true //成功

}

识别回调请求参数说明：

参数名	数据类型	说明
deviceKey	String	设备序列号
personId	String	人员 ID（陌生人 personId 是 STRANGERBABY）
time	String	识别记录时间戳（以设备时间为准）
type	String	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，且识别到是陌生人）
		card_0 刷卡记录
		faceAndcard_0 人&卡模式记录，且人员在 passtime 权限时间内
		faceAndcard_1 人&卡模式记录，且人员在 passtime 权限时间外
		faceAndcard_2 人&卡模式记录，且识别到是陌生人
		idcard_0 人&证模式记录
		qrcode 二维码模式记录
		palm_0 掌纹识别，且该人员在 passtime 权限时间内
		palm_1 掌纹识别，且该人员在 passtime 权限时间外

		face_palm_0 掌纹&人脸双重认证，且该人员在 passtime 权限时间内
		face_palm_1 掌纹&人脸双重认证，且该人员在 passtime 权限时间外
		face_palm_2 掌纹&人脸双重认证，且识别失败或识别到的是陌生人
path	String	访问此 url 需设备局域网在线，且客户端与设备处于局域网同一网段
imgBase64	String	抓拍照片 base64 串
data	String	人证模式的身份证 json 字符串 or 刷卡模式的 IC 卡号 or 二维码模式的二维码数据 or 问卷答案信息 **例:data=001_true||1_1**** 其中 **001** 为问卷 id, **true** 为答题结果,全部答对为 **true**,否则为 **false**, **1_1** 其中 **_** 前为题目 id, **_** 后为用户选择选项 id,多个选项以 ** 分隔.
ip	String	设备局域网 ip 地址
searchScore	String	识别比分，仅供参考
livenessScore	String	活体比分，仅供参考
temperature	String	人员温度测量值
standard	String	设置的体温异常值
temperatureState	String	体温状态:1 正常 2 异常
mask	String	-1 未开启 0 未佩戴口罩 1 佩戴口罩

根据外设的不同 **data** 字段上报数据有不同：

- 人证模式的身份证 json 字符串
- 刷卡模式的 IC 卡号
- 二维码模式的二维码数据
- 问卷答案信息

例: data=001 true||1_1**

其中 001 为问卷 id, true 为答题结果,全部答对为 true,否则为 false,

1_1 其中 _ 前为题目 id, _ 后为用户选择选项 id,

多个选项以 ** 分隔.

酒检json字符串

```
{
//浓度
"conc":0,
//单位
"unit":"mg/L"}
```

- identityData : 身份信息
  - name : 姓名
  - dsName : 脱敏姓名
  - idCard : 身份证号
  - dsIdCard : 脱敏身份证号
  - phone : 手机号
- 注意 : 使用身份证或身份证白名单时,信息存储在 身份证 json 字符串中的 extend 参数中

识别回调请求响应字符串：

注意: 如果未响应正确的字符串,记录会重复上传

{"result":1,"code":"000"}

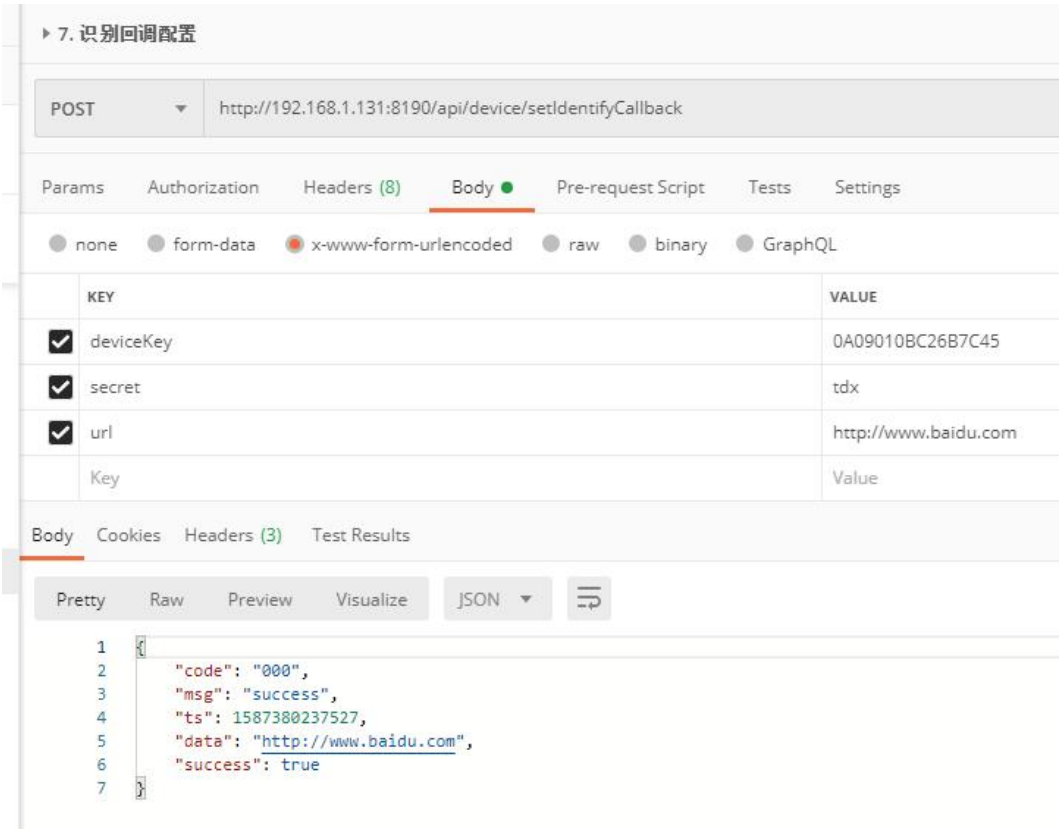
身份证 json 属性

```
public class CardInfo {
    /**
     * 姓名(the name of IDCard)*
     /
    private String name;/
    **
     * 性别(the sex of IDCard)*
     /
    private String sex;/
    **
     * 民族(the nation of IDCard)*
     /
    private String nation;/
    **
     * 家庭住址（family address）*
     /
```

```
private String address;

/**

 * 身份证号码（IDCard number）

 */

private String idNum;

/**

 * 签发机关（issuing authority ）

 */

private String issuingOrgan;



/**

 * 设备序列号

 */

private String deviceKey;

/**

 * 生日

 */

private String birthday;

/**

 * 身份证图片保存地址

 */

private String photoPath;
```

```
    /**
     * 身份证有效日期
     */
    private String usefulLife;
    /**
     * 创建时间
     */
    private Long createTime;
    private boolean compareResult = true;
}
```

PostMan 示例：



8. 注册照片回调配置

请求地址：**http://SDK 中间件服务器 IP:8190/api/device/setImgRegCallback**

请求方法：**POST**

请求数据：

参数名	数据类型	必填	说明
deviceKey	String	是	设备序列号
secret	String	是	设备密码
url	String	是	回调地址

请求说明：

- 回调 URL 需要符合正则表达式：**((http|https)://)((a-zA-Z0-9\._-{2,6})|(0-9{1,3}\.0-9{1,3}))(:0-9*)?**
- 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。

响应数据：

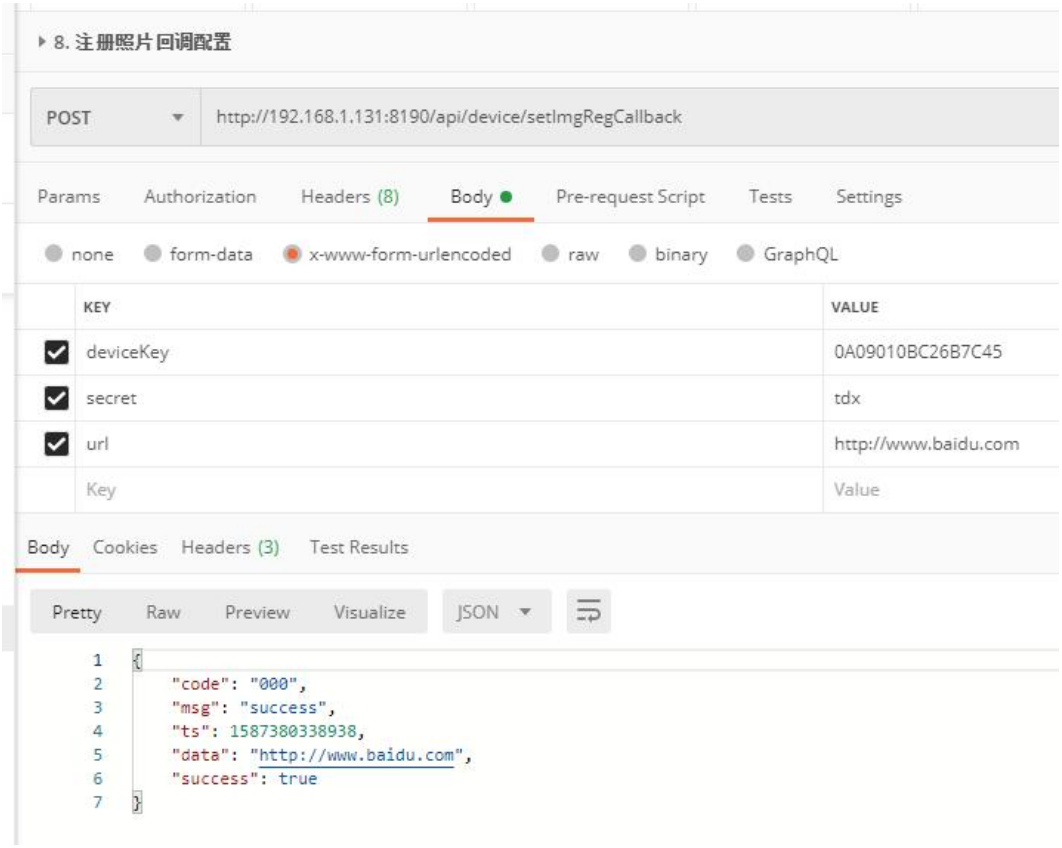
```
{
    "code": "000",
   "msg": "success",
   "success": true //成功
}
```

注册照片回调参数说明：

参数名	数据类型	说明
deviceKey	String	设备序列号
personId	String	人员 ID
time	String	拍照时间戳（以设备时间为准）

imgPath	String	照片路径
faceId	String	照片 ID
ip	String	设备 IP 地址
imgBase64	String	照片 base64 串

PostMan 示例：



9. 心跳地址配置

请求地址：**http://SDK 中间件服务器 IP:8190/api/device/setHeartbeatUrl**

请求方法：**POST**

请求数据：

参数名	数据类型	必填	说明

deviceKey	String	是	设备序列号
secret	String	是	设备密码
url	String	是	心跳地址

请求说明：

- 设备每隔一分钟会向该回调地址 **POST** 心跳信息
- 回调 URL 需要符合正则表达式：**((http|https)://)((a-zA-Z0-9\._-{2,6})|(0-9{1,3}\.0-9{1,3}))(:0-9*)?**

响应数据：

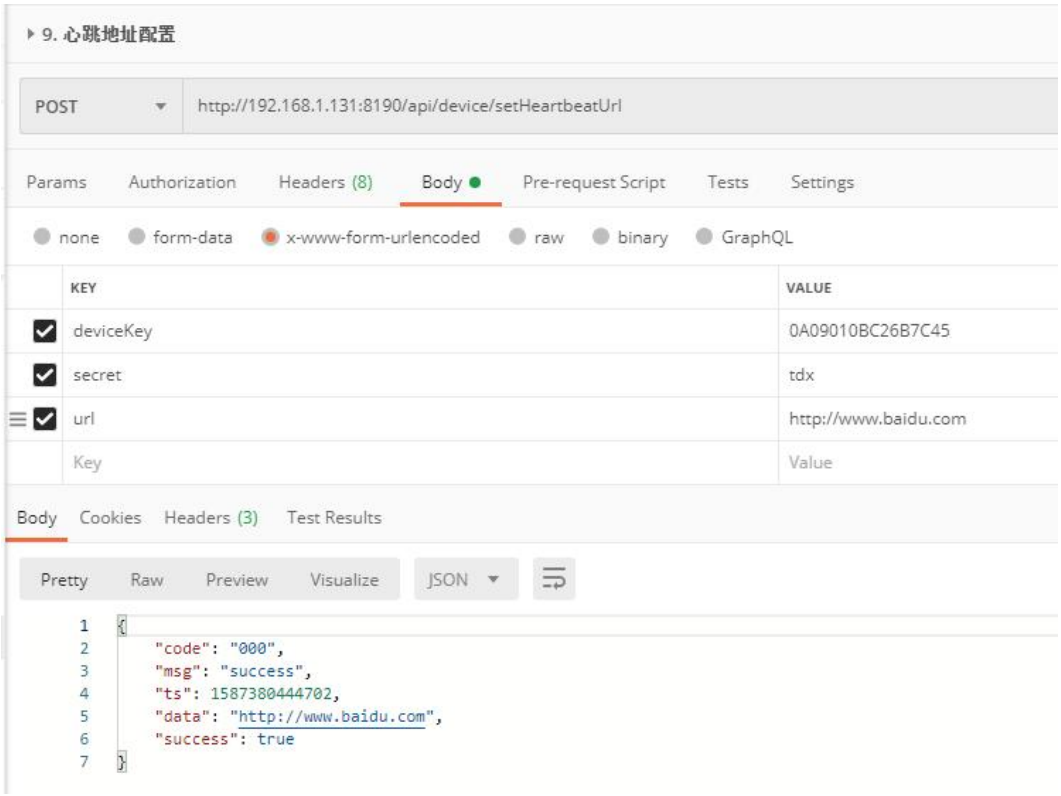
```
{
    "code": "000",
   "msg": "success",
   "success": true //成功
}
```

心跳参数说明：

参数名	数据类型	说明
deviceKey	String	设备序列号
time	String	识别记录时间戳（以设备时间为准）
personCount	String	设备注册人员数量
faceCount	String	设备注册照片数量
ip	String	设备 IP 地址
version	String	设备当前版本号

心跳请求返回参数说明：
无需返回信息，设备不做处理

PostMan 示例：



10. 拍照注册

请求地址：**http://SDK 中间件服务器 IP:8190/api/device/takeImg**

请求方法：**POST**

请求数据：

参数名	数据类型	必填	说明
deviceKey	String	是	设备序列号
secret	String	是	设备密码
personId	String	是	人员 id

请求说明：

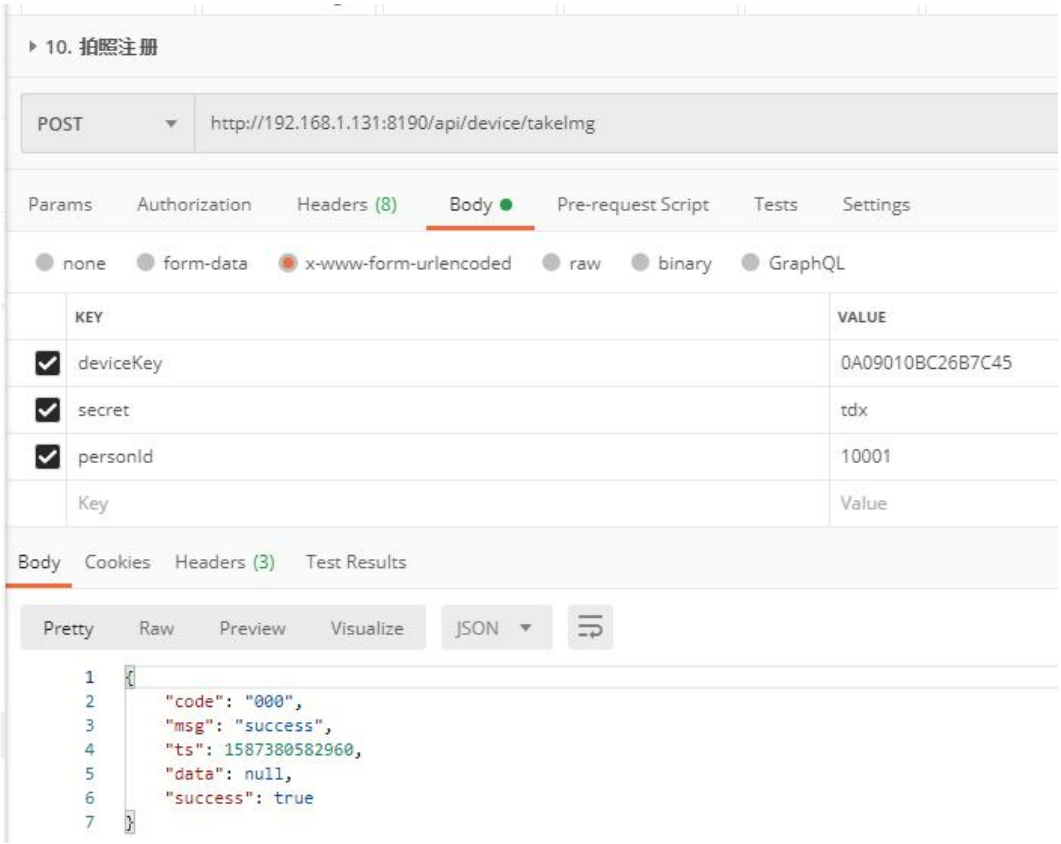
- 人员 id 必须为已存在；若不存在该人员 id，则无法进入自动拍照模式
- 拍照注册的过程实际上是设备先拍摄照片，再调用照片注册接口注册照片，faceId 由设备随机生成。

- 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。

响应数据：

```
{
    "code": "000",
   "msg": "success",
   "success": true //成功
}
```

PostMan 示例：



11. 识别模式配置

> 需设备支持，v1.2.X.X+以上版本支持，v1.3.1.9+以上版本支持二维码模式

> 指纹,掌纹设备,并且 v1.4.1.65 以上版本支持掌纹,指纹先关接口开关,

请求地址：**http://SDK 中间件服务器 IP:8190/api/device/configIdentifyModel**

请求方法：**POST**

请求数据：

参数名	数据类型	必填	说明
deviceKey	String	是	设备序列号
secret	String	是	设备密码
recCardFaceValue	Int	是	人证比对阀值
recModeFaceEnable	Int	是	刷脸模式开关
recModeCardEnable	Int	是	刷卡模式开关
recModeCardHardware	Int	是	刷卡模式外接硬件类型
recModeCardIntf	Int	是	刷卡模式卡号传输接口
recModeCardFaceEnable	Int	是	卡 &人脸双重模式开关
recModeCardFaceHardware	Int	是	卡 &人脸双重认证模式外接硬件类型
recModeCardFaceIntf	Int	是	卡 &人脸双重认证模式卡号传输接口
recModeIdcardFaceEnable	Int	是	人证比对模式开关
recModeIdcardFaceHardware	Int	是	人证比对模式外接硬件类型
recModeIdcardFaceIntf	Int	是	人证比对模式卡号传输接口
recModeQrcodeEnable	Int	是	二维码模式开关，1：关闭（默认），2：打开
recModeQrcodeIntf	Int	是	二维码传输接口 * 2.232 串口（默认）* 3.USB
recModeCardTemperatureEnable	Int	是	卡&测温开关 1.关闭（默认）2.打开
recModePalmEnable	Int	是	掌纹识别开关 1.关闭 2.打开（默认）
recModePalmFaceEnable	Int	是	掌纹&人脸开关 1.关闭（默

			认） 2.打开
recModeFingerEnable	Int	是	指纹识别开关 1.关闭 2.打开（默认）
recModeFingerFaceEnable	Int	是	指纹&人脸开关 1.关闭（默认） 2.打开

请求说明(示例，请根据实际情况调整)：

"recCardFaceValue": 75, // Int，人证比对阀值。默认：75。请输入 70-100 之间的整数。分数越高，识别准确率越高，但识别速度会变慢。

"recModeFaceEnable": 2, // Int，刷脸模式开关。1.关闭，2.打开（默认）

"recModeCardEnable": 1, //Int，刷卡模式开关。1.关闭（默认），2.打开

"recModeCardHardware": 1, // Int，刷卡模式外接硬件类型。1.IC 读卡器（默认）2.二维码阅读器

"recModeCardIntf": 2, // Int，刷卡模式卡号传输接口。1. 韦根（默认），2. 232 串口

"recModeCardFaceEnable": 1, // Int，卡 &人脸双重模式开关。1.关闭（默认），2.打开

"recModeCardFaceHardware": 1, // Int，卡 &人脸双重认证模式外接硬件类型。1.IC 读卡器（默认）2.二维码阅读器

"recModeCardFaceIntf": 2, // Int，卡 &人脸双重认证模式卡号传输接口。1.韦根（默认），2. 232 串口

"recModeIdcardFaceEnable": 1, // Int，人证比对模式开关。1.关闭（默认），2.打开

"recModeIdcardFaceHardware": 3, // Int，人证比对模式外接硬件类型。2. ZTK 中控身份证阅读器（默认）

"recModeIdcardFaceIntf": 1, // Int，人证比对模式卡号传输接口。3.USB（默认）

"recModeQrcodeEnable": 1, //Int, 刷二维码模式开关。1.关闭（默认），2.打开

"recModePalmEnable": 2, // Int, 掌纹识别开关,1.关闭 2.打开（默认）

"recModePalmFaceEnable": 1, // Int, 掌纹&人脸开关,1.关闭（默认） 2.打开

"recModeFingerEnable": 2, // Int, 指纹识别开关,1.关闭 2.打开（默认）

"recModeFingerFaceEnable": 1 // Int, 指纹&人脸开关,1.关闭（默认） 2.打开

响应数据：

{

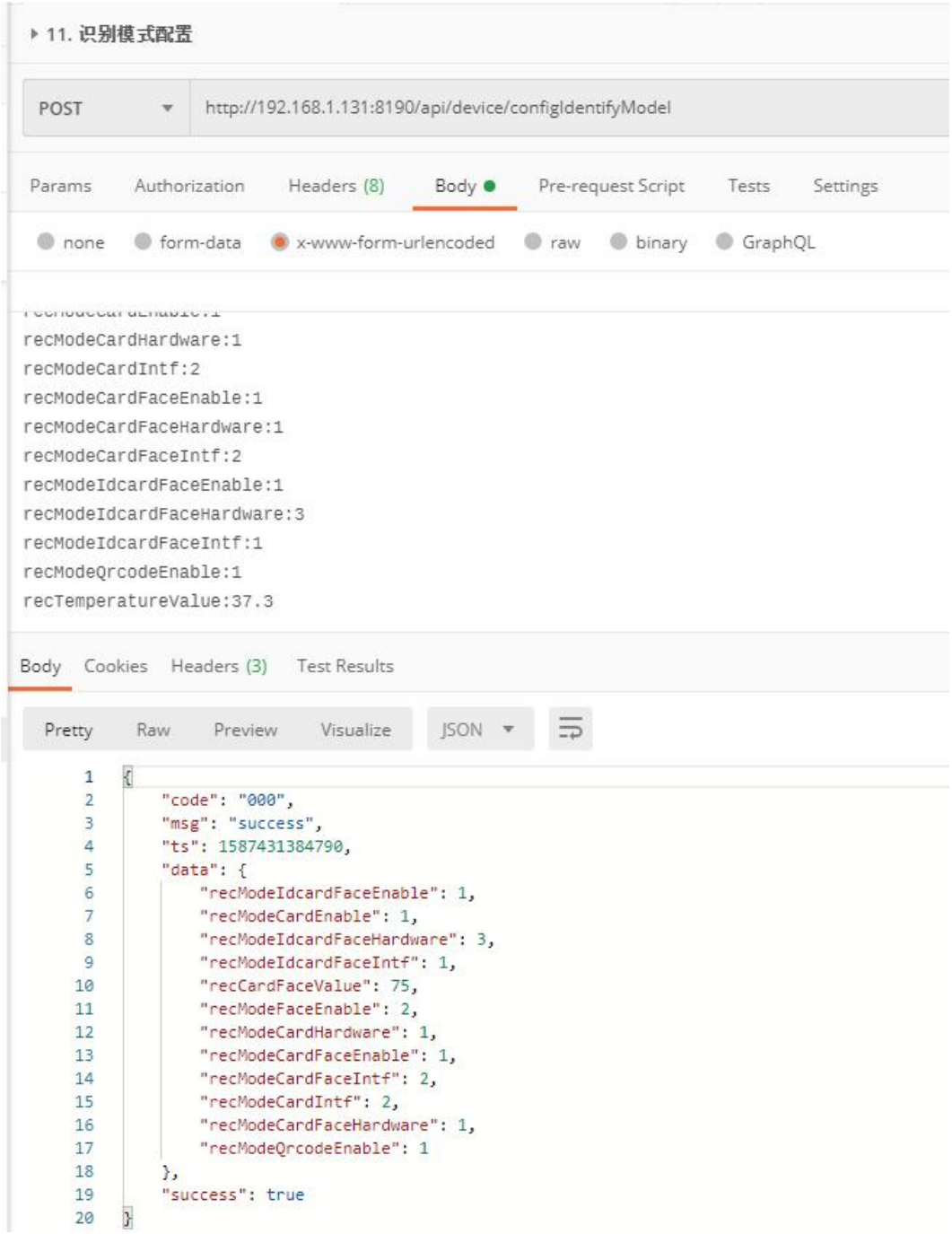
"code": "000",

"msg": "success",

"success": true //成功

}

PostMan 示例：



12. 人员创建

请求地址：**http://SDK 中间件服务器 IP:8190/api/person/add**

请求方法：**POST**

请求数据：

参数名	数据类型	必填	说明
deviceKey	String	是	设备序列号
secret	String	是	设备密码
id	String	否	人员 ID
name	String	是	人员名
idcardNum	String	否	卡号
expireTime	Long	否	过期时间戳，13 位
blacklist	Integer	否	黑名单权限
vaccination	Integer	否	是否接种疫苗
vaccinationTime	String	否	疫苗接种时间
remark	String	否	备注

请求说明：

- id、idcardNum 内容可以传空
- 若传入 id 值，id 只允许数字和英文字母，区分大小写
- 重复添加以最后一次为准

参数说明：

"id": "abc", //String，人员 id，id 为人员标识，限制 64 位。注册人员时，若 id 不填，则系统会自动生成 32 位 id 并返回；若填了 id，系统会以此 id 为主键存入本地数据库中；若 id 重复，会报错

"name": "小王", //String，姓名，不可为空。识别成功后会在屏幕上显示该名字

"idcardNum": "410822199908221428", //String，卡号，创建时可以不填，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

"blacklist": 1, //黑名单权限 0 和不填表示非黑名单用户 1 表示禁止通行 2 表示可以通行但后台报警

"vaccination": -1, //是否接种疫苗 -1 未知 0 未接种 1 已接种

"vaccinationTime": "2021-04-23 12:00:00", //疫苗接种时间

"remark": "remark" // 备注

响应数据：

{

"data": {

"createTime": 1532398934472, //String，人员创建时的时间戳（毫秒级），以设备时间为准

"personId": "abc", //String，人员 id，id 为人员标识，限制 64 位。注册人员时，若 id 不填，则系统会自动生成 32 位 id 并返回；若填了 id，系统会以此 id 为主键存入本地数据库中；若 id 重复，会报错

"idCard": "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": "小王", //String，姓名，不可为空。识别成功后会在屏幕上显示该名字

"expireTime": 1532398934472 // 过期时间戳,过期后人员将自动删除

},

"msg": "人员信息添加成功",

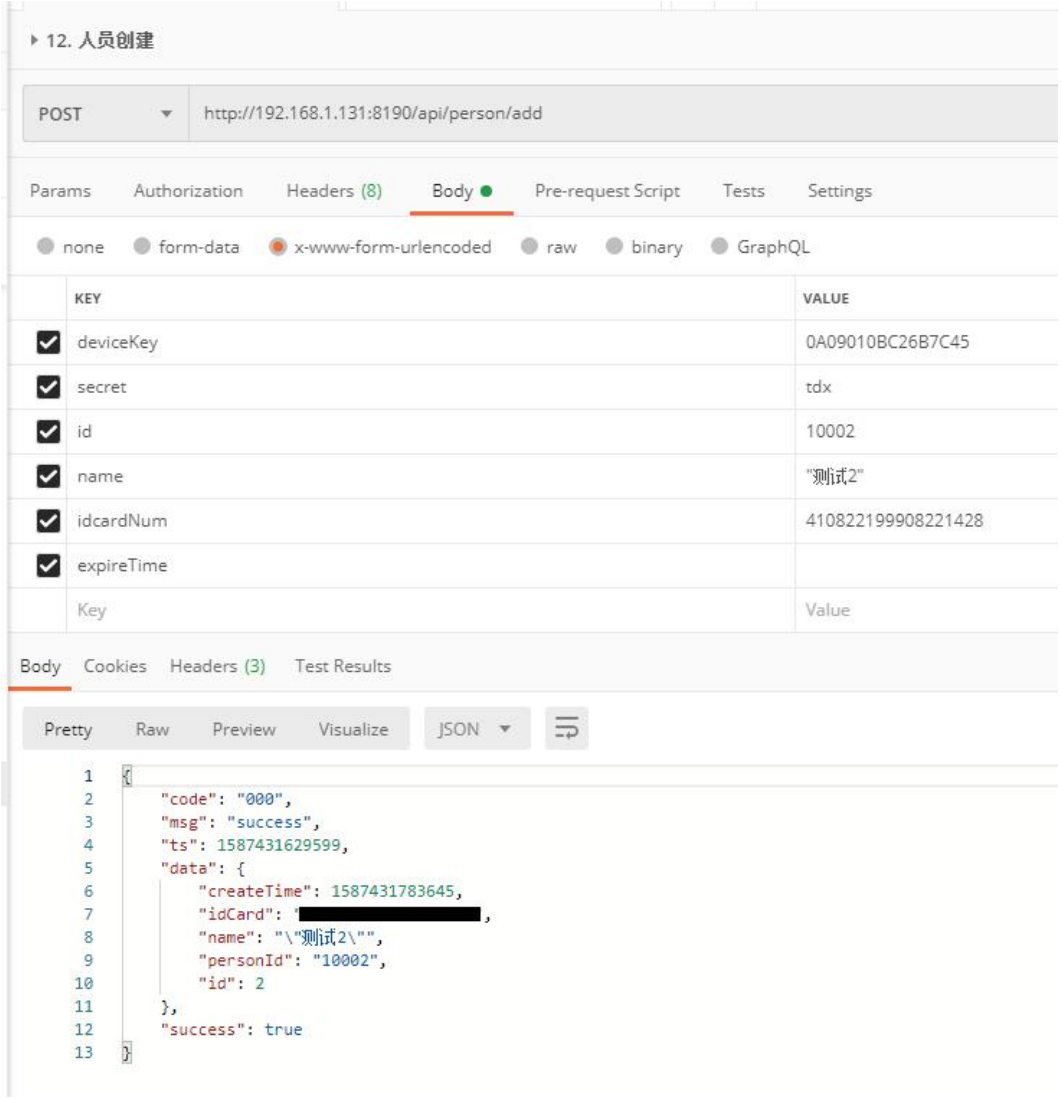
"code": "000",

"msg": "success",

"success": true //成功

}

PostMan 示例：



13. 人员删除（批量）

请求地址：**http://SDK 中间件服务器 IP:8190/api/person/del**

请求方法：**POST**

请求数据：

参数名	数据类型	必填	说明
deviceKey	String	是	设备序列号
secret	String	是	设备密码

personId	String	是	人员 ID

请求说明：

- 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，该人员对应的识别记录、现场抓拍照片，该人员对应的注册照片

响应数据：

{

"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",

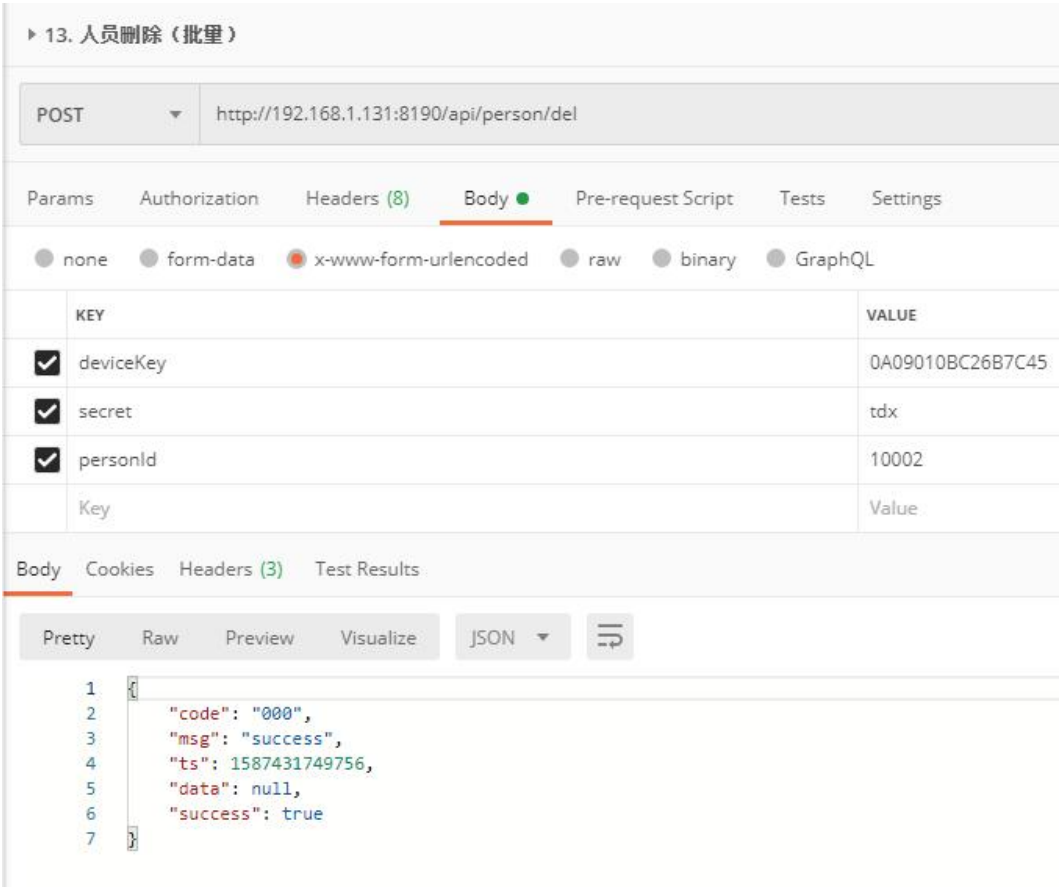
"code": "000",

"msg": "success",

"success": true //成功

}

PostMan 示例：



14. 人员信息查询

请求地址：**http://SDK 中间件服务器 IP:8190/api/person/find**

请求方法：**GET**

请求数据：

参数名	数据类型	必填	说明
deviceKey	String	是	设备序列号
secret	String	是	设备密码
personId	String	是	人员 id

响应数据：

{

"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）。

"personId": "12345", //人员 id，限制 64 位

"idCard": "xxx", //卡号，长度无限制

"name": "xxx", //姓名，不可为空

"expireTime": 1234567890123 // 过期时间，返回空则未设置

},

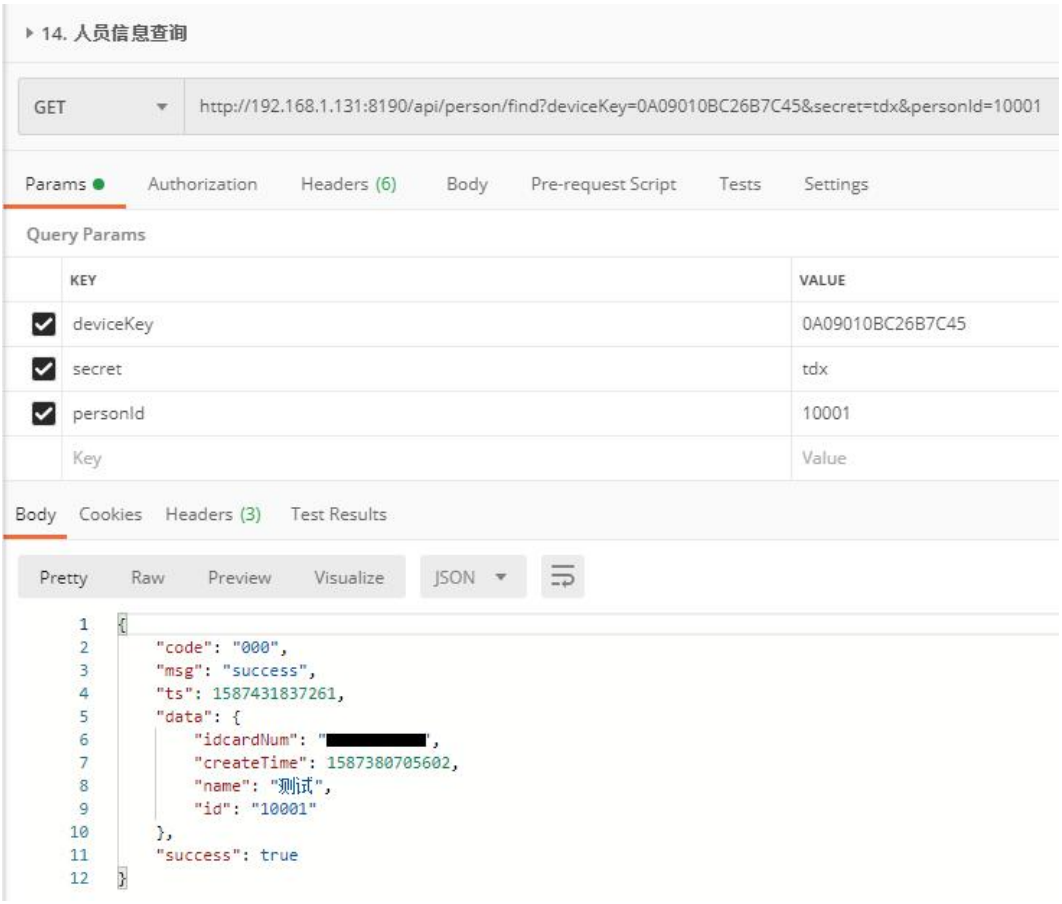
"code": "000",

"msg": "success",

"success": true //成功

}

PostMan 示例：



15. 时间段权限设置

请求地址：**http://SDK 中间件服务器 IP:8190/api/person/passtime/add**

请求方法：**POST**

请求数据：

参数名	数据类型	必填	说明
deviceKey	String	是	设备序列号
secret	String	是	设备密码
personId	String	是	人员 ID
passtime	String	是	某人员每日允许进入的时间段

请求说明：

- Json 示例：

  {"personId":"9eecc839cd7941c5a4d3165202dd3c32","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：

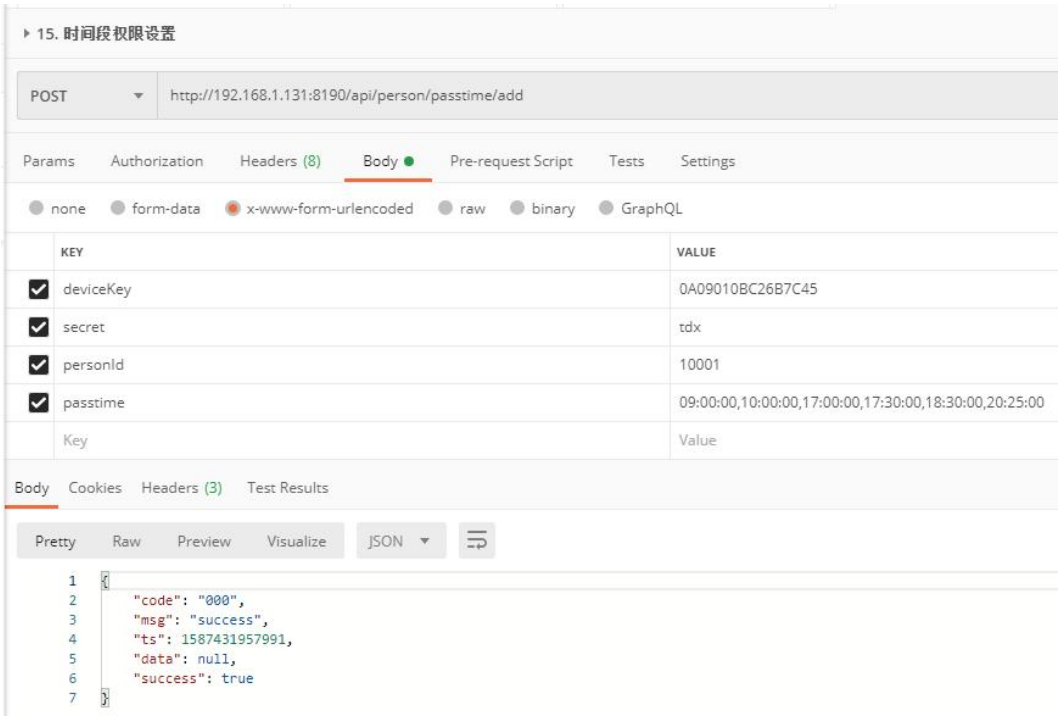
  09:00:00,11:00:00

- 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。
- 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”

响应数据：

```
{
      "code": "000",
   "msg": "success",
   "success": true //成功
}
```

PostMan 示例：



16. 时间段权限删除

请求地址：**http://SDK 中间件服务器 IP:8190/api/person/passtime/del**

请求方法：**POST**

请求数据：

参数名	数据类型	必填	说明
deviceKey	String	是	设备序列号
secret	String	是	设备密码
personId	String	是	人员 id

请求说明：

- 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

响应数据：

{

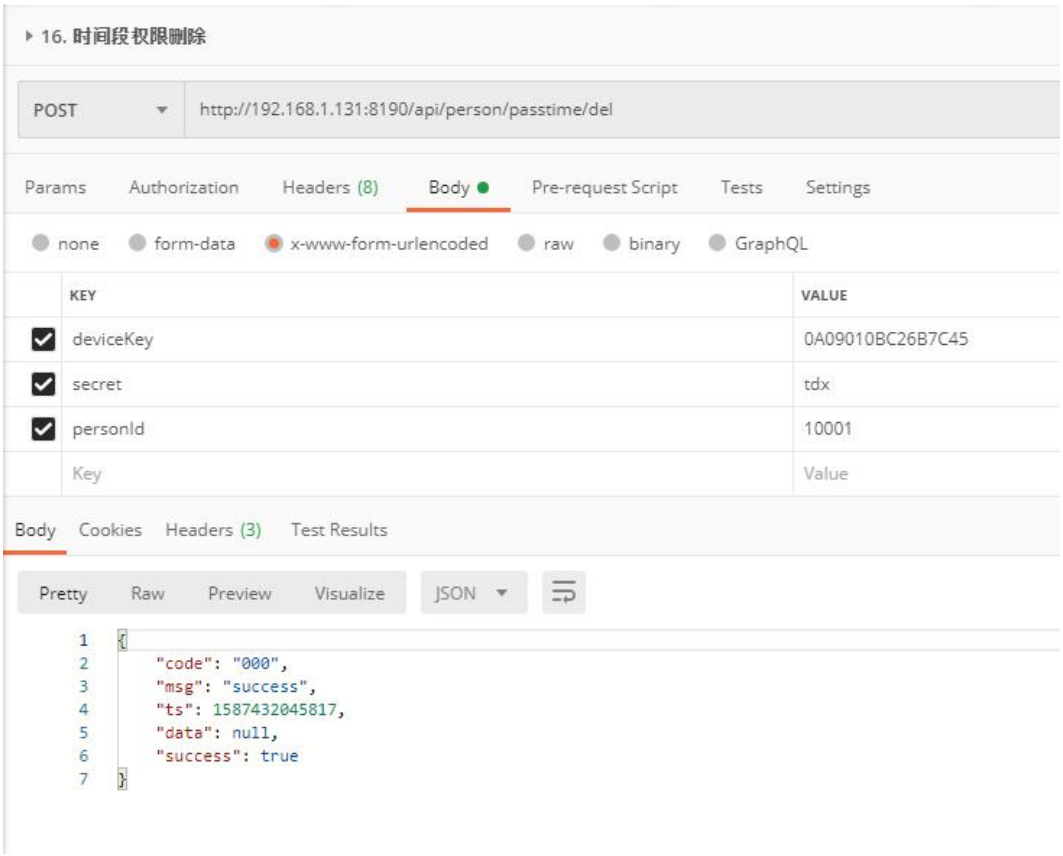
"code": "000",

"msg": "success",

"success": true //成功

}

PostMan 示例：



17. 照片注册（base64）

请求地址：**http://SDK 中间件服务器 IP:8190/api/face/add**

请求方法：**POST**

请求数据：

参数名	数据类型	必填	说明

deviceKey	String	是	设备序列号
secret	String	是	设备密码
personId	String	是	人员 id
faceId	String	否	照片 id
imgBase64	String	是	照片的 base64 编码字符串

请求说明：

- 必须先注册人员，才能添加照片
- 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
- 图片格式支持 png、jpg、jpeg；不加头部，如：data:image/jpg;base64

响应数据：

{

"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

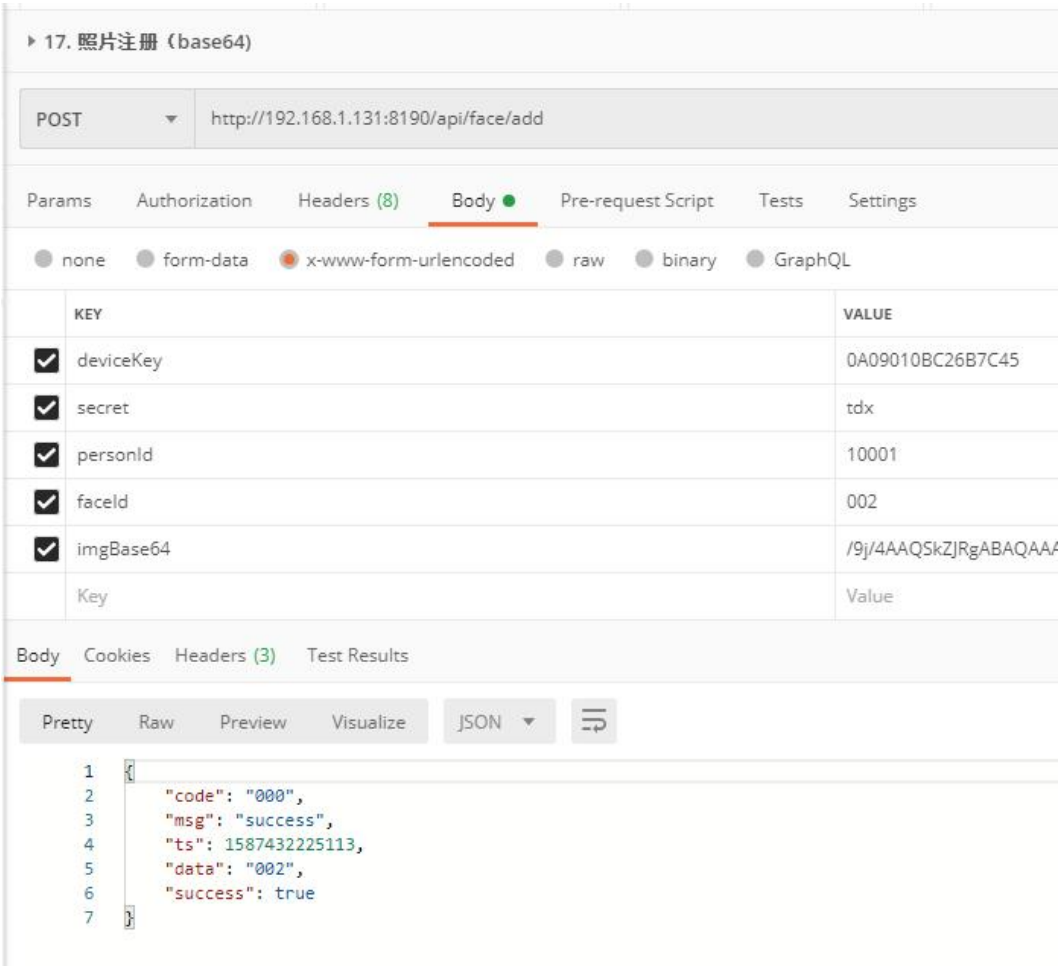
"code": "000",

"msg": "success",

"success": true //成功

}

PostMan 示例：



18. 照片查询

请求地址：**http://SDK 中间件服务器 IP:8190/api/face/find**

请求方法：**POST**

请求数据：

参数名	数据类型	必填	说明
deviceKey	String	是	设备序列号
secret	String	是	设备密码
personId	String	是	人员 id

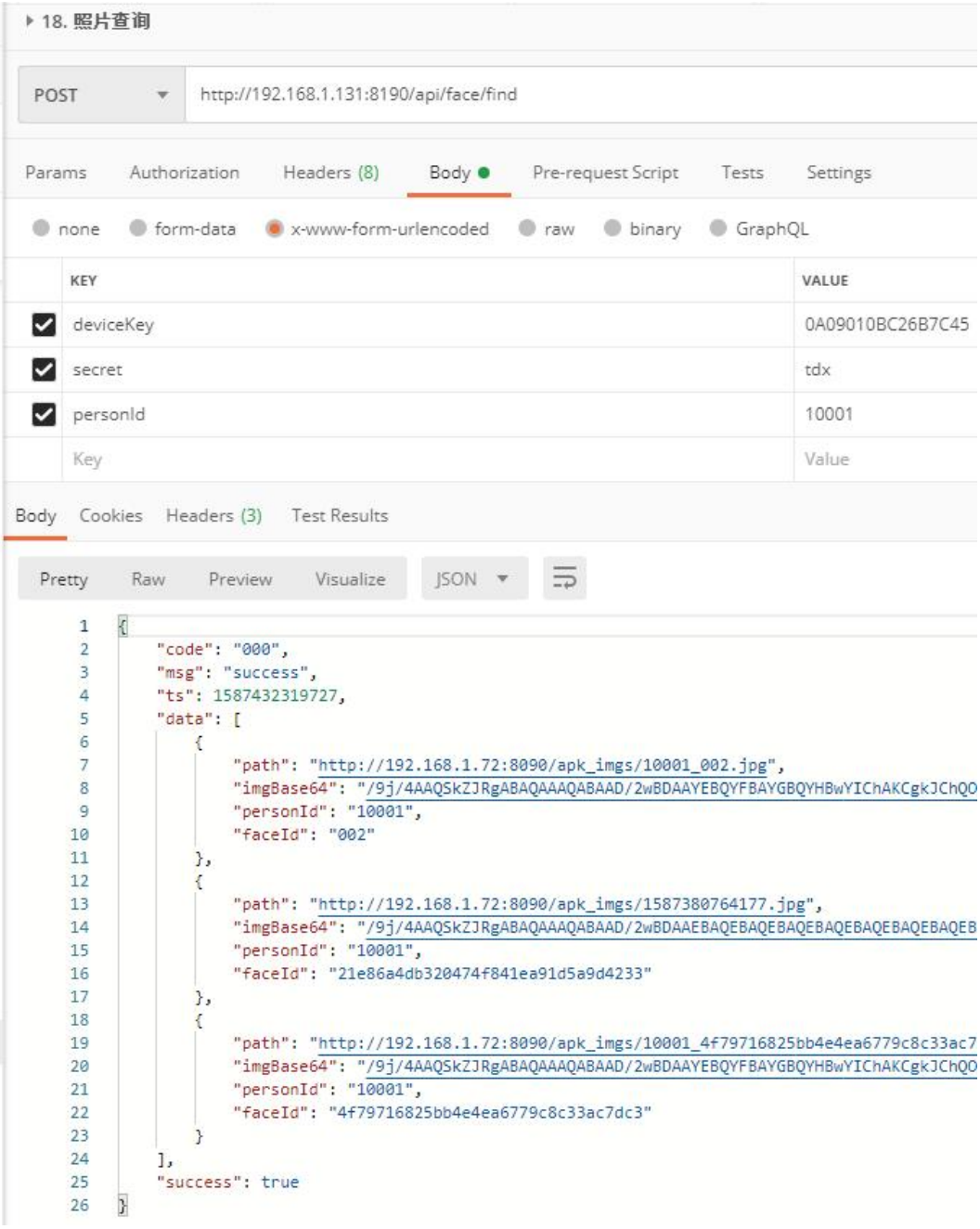
请求说明：

- 查询该人员的所有注册照片

响应数据：


```
{
  "data": [
    {
      "faceId": "b8df54f2ef924c1f879e509b44593135", //照片 id
      "path":
"http://192.168.24.100:8090/faceRegister/12345abc_b8df54f2ef924c1f879e509b445
93135.jpg", //注册照片存储在设备内的路径
      "personId": "12345abc" //人员 id
    }
  ],
  "code": "000",
  "msg": "success",
  "success": true //成功
}
```


PostMan 示例：



19. 照片删除

请求地址：**http://SDK 中间件服务器 IP:8190/api/face/del**

请求方法：**POST**

请求数据：

参数名	数据类型	必填	说明

deviceKey	String	是	设备序列号
secret	String	是	设备密码
faceId	String	是	照片 id

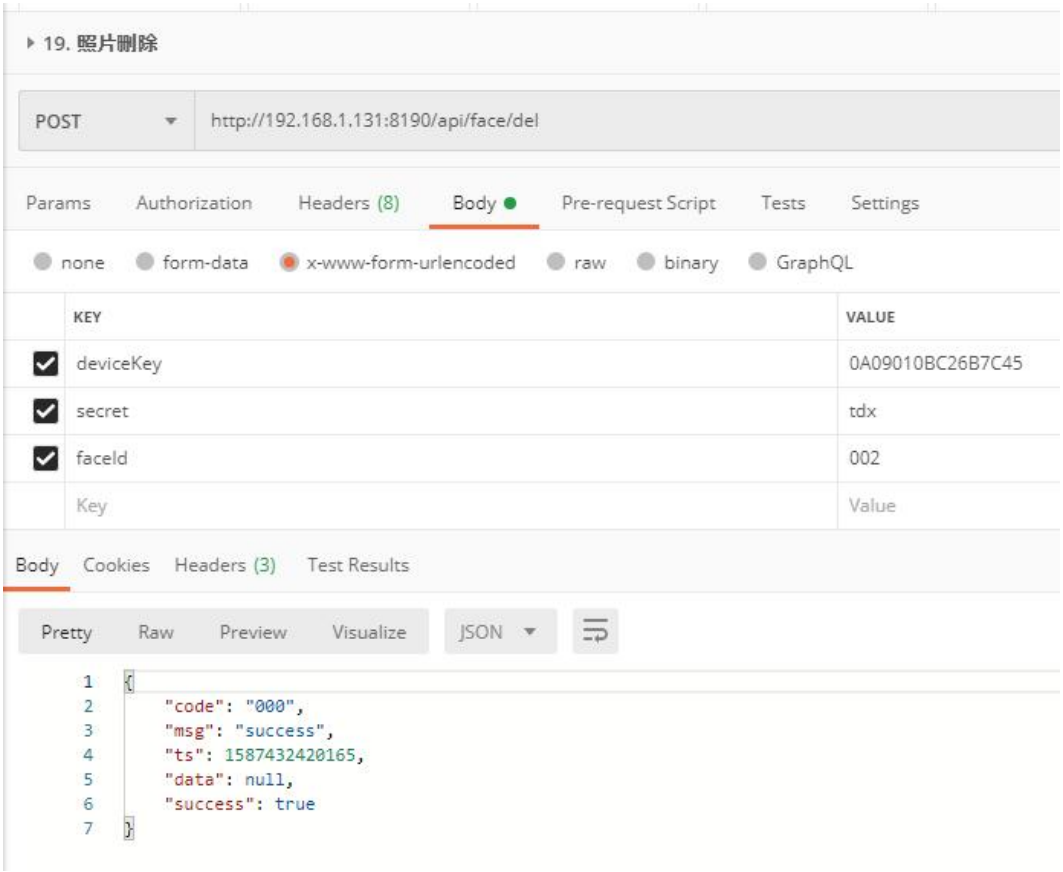
请求说明：

- 删除该 faceId 对应的注册照片，不可恢复

响应数据：

```
{
    "code": "000",
   "msg": "success",
   "success": true //成功
}
```

PostMan 示例：



20. 清空人员照片

请求地址：**http://SDK 中间件服务器 IP:8190/api/face/clear**

请求方法：**POST**

请求数据：

参数名	数据类型	必填	说明
deviceKey	String	是	设备序列号
secret	String	是	设备密码
personId	String	是	人员 id

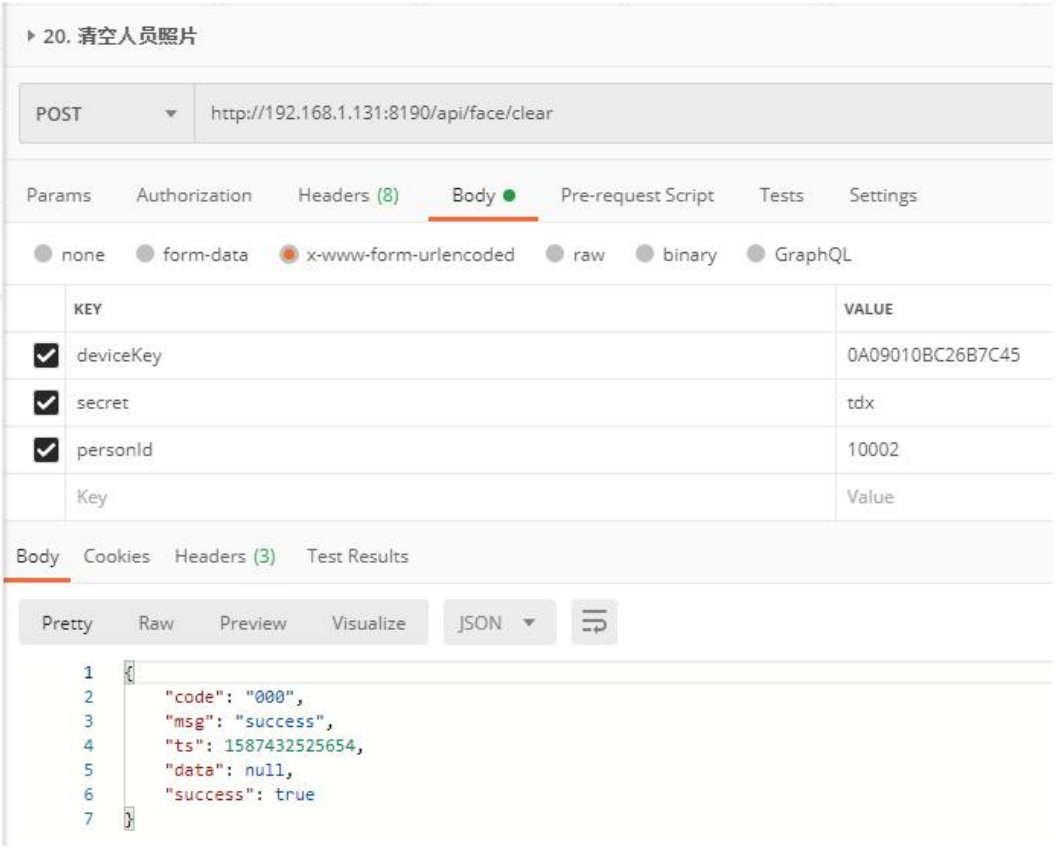
请求说明：

- 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

响应数据：

```
{

    "code": "000",

  "msg": "success",

  "success": true //成功

}
```

PostMan 示例：



21. 设备显示提示信息

> 参数 speak,showTime,rows,align,color 需 v1.41.6.1+版本支持

请求地址：**http://SDK 中间件服务器 IP:8190/api/device/showMessage**

请求方法：**POST**

请求数据：

参数名	数据类型	必填	说明
deviceKey	String	是	设备序列号
secret	String	是	设备密码
content	String	是	提示信息
speak	Boolean	否	是否语言播报

showTime	Int	否	信息显示时间，秒
rows	Int	否	信息显示行数
align	String	否	left 左，right 右，middle 居中
color	String	否	文字颜色如：#ffffff

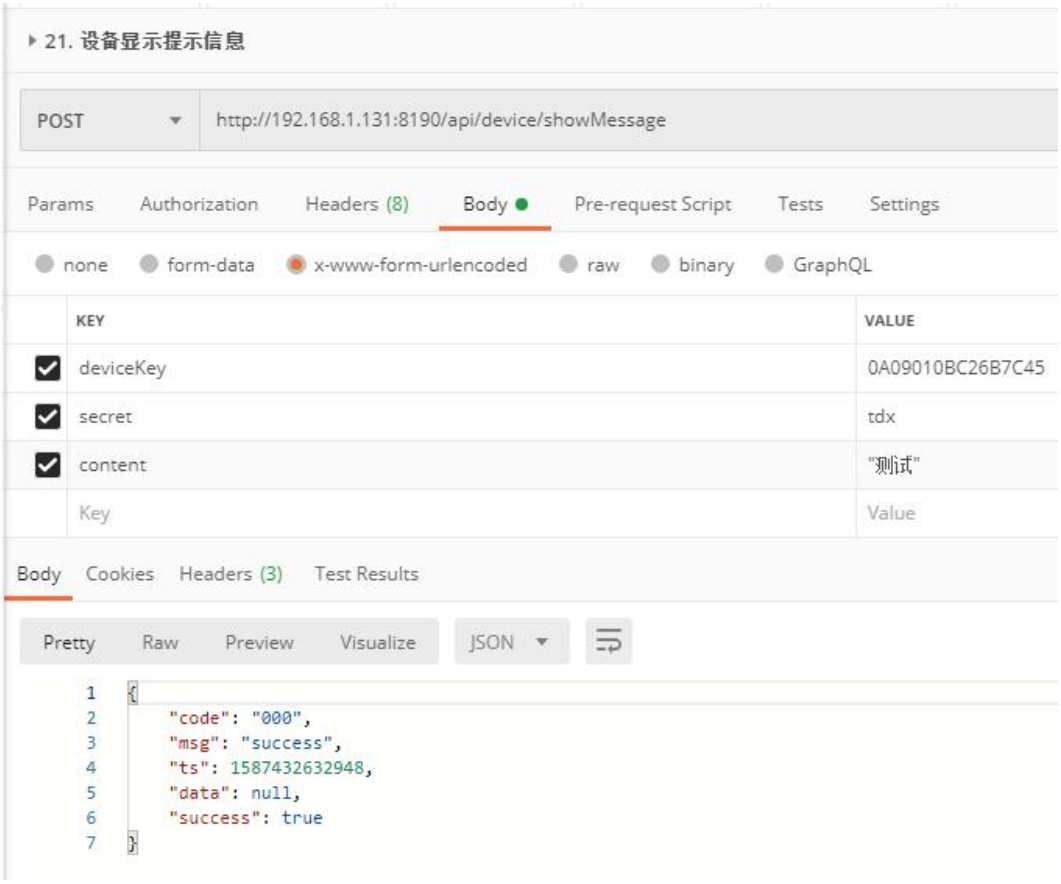
请求说明：

- 提示信息内容建议不超过 32 个字，避免影响设备正常使用

响应数据：

```
{
    "code": "000",
  "msg": "success",
  "success": true //成功
}
```

PostMan 示例：



~~22. 三维码模式回调配置(v1.41.1.2+已弃用)~~

请求地址：**http://SDK 中间件服务器 IP:8190/api/device/setQrcodeCallback**

请求方法：**POST**

请求数据：

参数名	数据类型	必填	说明
deviceKey	String	是	设备序列号
secret	String	是	设备密码
url	String	是	回调地址

请求说明：

- 传入内容为空可以清空回调地址，清空后将不再进行回调

- 回调 URL 需要符合正则表达式：**((http|https)://)((a-zA-Z0-9\._-{2,6})|(0-9{1,3}\.0-9{1,3}))(:0-9*)?**

响应数据：

```
{
    "code": "000",
   "msg": "success",
   "success": true //成功
}
```

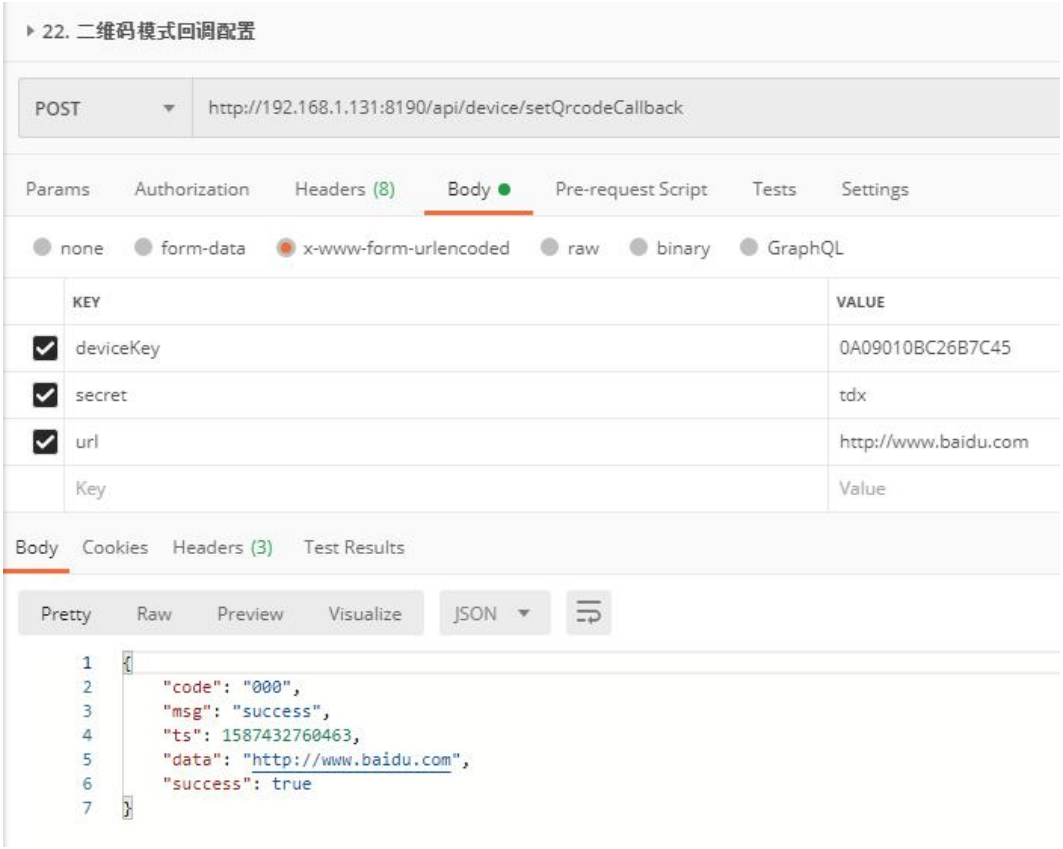
二维码回调参数说明：

参数名	数据类型	说明
deviceKey	String	设备序列号
time	String	识别记录时间戳（以设备时间为准）
qrcode	String	二维码
imgBase64	String	抓拍人脸数据
temperature	String	体温
temperatureState	String	体温状态 1 正常 2 高温

二维码回调返回参数说明：

无需返回信息，设备不做处理

PostMan 示例：



23. 照片注册（文件上传）

请求地址：**http://SDK 中间件服务器 IP:8190/api/face/upload**

请求方法：**POST**

请求数据：

参数名	数据类型	必填	说明
deviceKey	String	是	设备序列号
secret	String	是	设备密码
personId	String	是	人员 id
faceId	String	否	照片 id
imgFile	File	是	照片图片文件

请求说明：

- 必须先注册人员，才能添加照片
- 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
- 图片文件格式支持 png、jpg、jpeg

响应数据：

{

"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

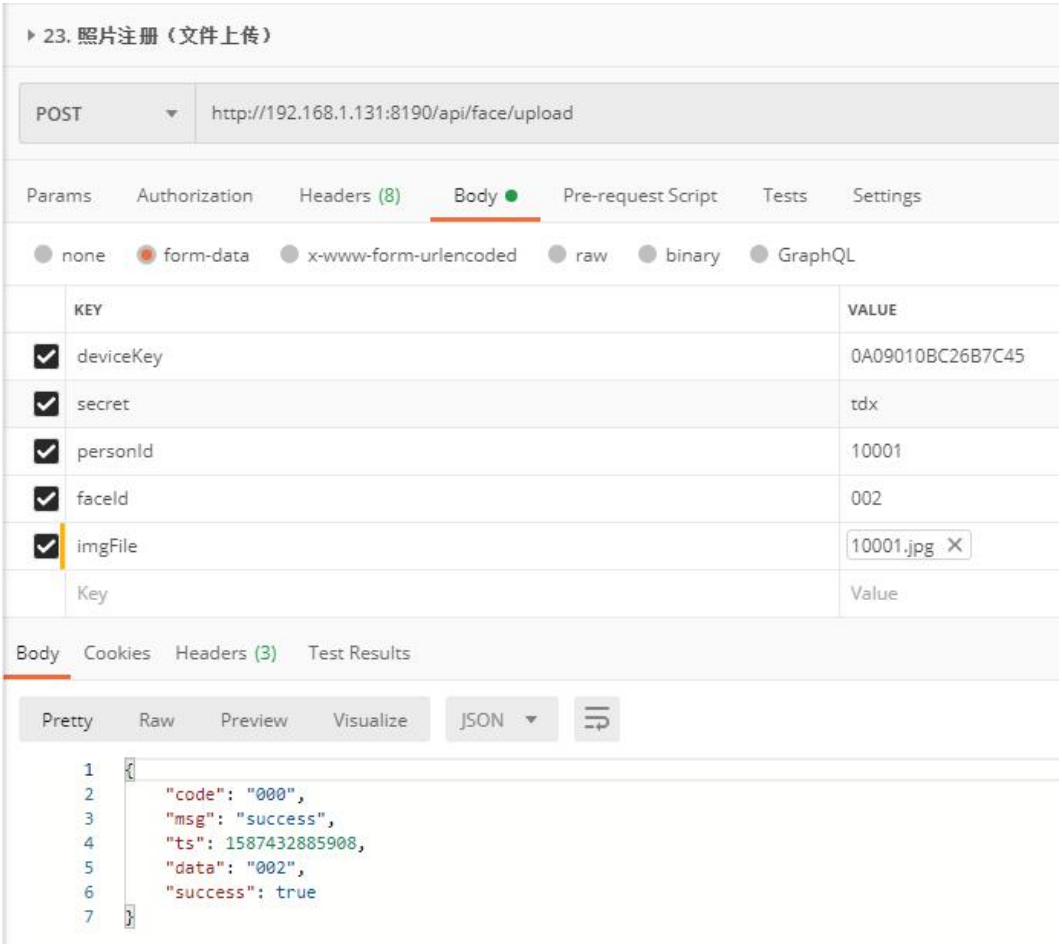
"code": "000",

"msg": "success",

"success": true //成功

}

PostMan 示例：



24. 测温参数配置

> v1.31.3.4+以上的测温版本支持

请求地址：**http://SDK 中间件服务器 IP:8190/api/device/configTemperature**

请求方法：**POST**

请求数据：

参数名	数据类型	必填	说明
deviceKey	String	是	设备序列号
secret	String	是	设备密码
errorTemperature	String	是	异常温度判断值，默认 37.3

isTemperatureOpen	Integer	否	是否打开测温模式 * 1：打开（默认） * 2：关闭

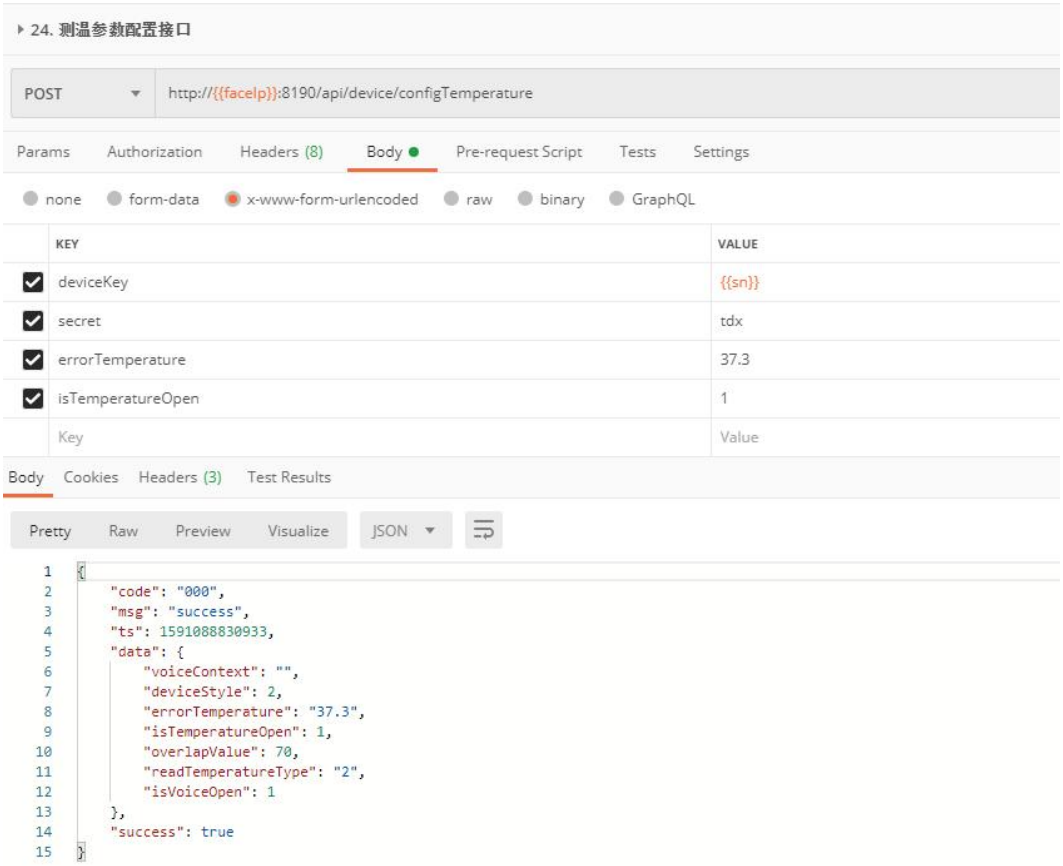
请求说明：

- 异常温度的判断值，设备默认值为 37.3，大于等于此值的均为异常温度。
- 只允许数字、精确到小数点后一位。

响应数据：

```
{

    "code": "000",

  "msg": "success",

  "success": true //成功

}
```

PostMan 示例：



25. 口罩检测配置

> v1.31.4.0+以上的测温版本支持

请求地址：**http://SDK 中间件服务器 IP:8190/api/device/configMask**

请求方法：**POST**

请求数据：

参数名	数据类型	必填	说明
deviceKey	String	是	设备序列号
secret	String	是	设备密码
isMaskOpen	String	是	是否打开口罩检测 1:打开(默认) 2:关闭
isVoiceOpen	String	是	口罩异常语音播报开启 1:打开(默认) 2:关闭
voiceContext	String	否	口罩异常语音播报自定义内容

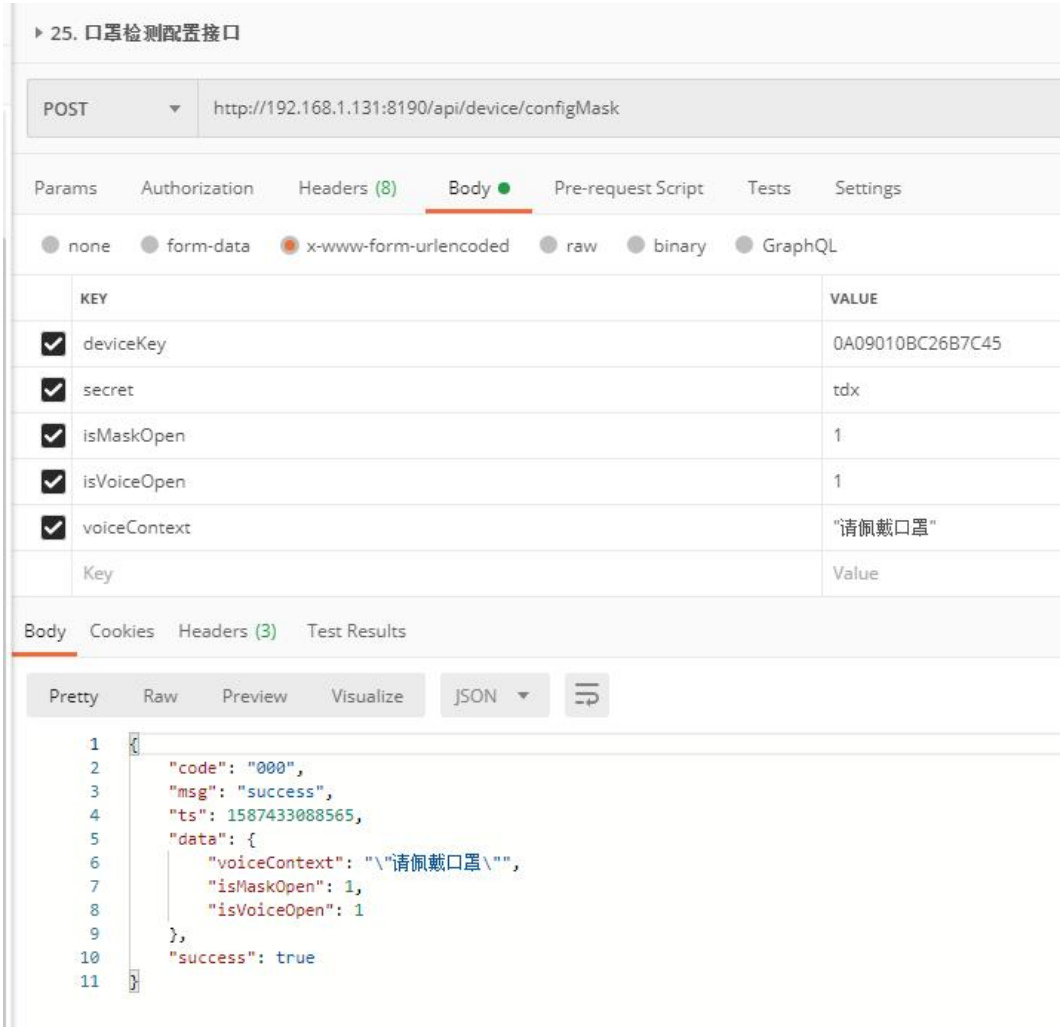
请求说明：

- voiceContext 设置为空，则默认播报“请佩戴口罩”。

响应数据：

```
{
    "code": "000",
  "msg": "success",
  "success": true //成功
}
```

PostMan 示例：



26. 清空回调地址

请求地址：**http://SDK 中间件服务器 IP:8190/api/device/clearCallbackUrl**

请求方法：**POST**

请求数据：

参数名	数据类型	必填	说明
deviceKey	String	是	设备序列号
secret	String	是	设备密码

响应数据：

{

"code": "000",

"msg": "success",

"success": true //成功

}

PostMan 示例：

27. 广告配置

请求地址：**http://SDK 中间件服务器 IP:8190/api/advertising/config**

请求方法：**POST**

请求数据：

参数名	数据类型	必填	说明
deviceKey	String	是	设备序列号
secret	String	是	设备密码
isOpen	Integer	是	是否打开广告 0：关闭 1：打开
splitScreen	Integer	是	分屏数量 1：全屏 2：二分屏
type	Integer	是	广告类型 1：图片 2：视频 3：混合
showTime	Integer	是	图片广告显示时间，默认 5 秒
showClock	Integer	是	广告播放时是否显示系统时钟 0 关闭 1 打开

响应数据：

{

"code": "000",

"msg": "success",

"success": true //成功

}

PostMan 示例：

28. 广告添加

请求地址：**http://SDK 中间件服务器 IP:8190/api/advertising/add**

请求方法：**POST**

请求数据：

参数名	数据类型	必填	说明
deviceKey	String	是	设备序列号
secret	String	是	设备密码
adId	String	是	广告 id，必须唯一
adType	Integer	是	广告类型 1：图片 2：视频 3：混合
adSort	Integer	是	广告显示顺序
adUrl	String	是	广告连接地址

响应数据：

{

"code": "000",

"msg": "success",

"success": true //成功

}

PostMan 示例：

29. 广告删除

请求地址：**http://SDK 中间件服务器 IP:8190/api/advertising/del**

请求方法：**POST**

请求数据：

参数名	数据类型	必填	说明
deviceKey	String	是	设备序列号
secret	String	是	设备密码
adId	String	是	广告 id，必须唯一

响应数据：

```
{
        "code": "000",
    "msg": "success",
    "success": true //成功
}
```

PostMan 示例：

30. 展示自定义图片

> v1.41.5.0+以上版本支持

请求地址：**http://SDK 中间件服务器 IP:8190/api/device/showImage**

请求方法：**POST**

请求数据：

参数名	数据类型	必填	说明
deviceKey	String	是	设备序列号
secret	String	是	设备密钥
imgBase64	String	是	图片的 base64 编码字符串
showTime	Int	否	显示时间，默认 5s

- 图片格式支持 png、jpg、jpeg；不加头部，如：data:image/jpg;base64

响应数据：


```
{
    "code": "000",
    "msg": "请求成功",
    "ts": 1679645672842,
    "data": {
        "sn": [
            "001",
            "002"
        ]
    },
    "success": true
}
```


31. 自定义设置配置

请求地址：**http://SDK 中间件服务器 IP:8190/api/device/configCustom**

请求方法：**POST**

请求数据：

参数名	数据类型	必填	说明
deviceKey	String	是	设备序列号
secret	String	是	设备密钥
data	String	是	自定义配置数据

data 请求参数(示例，请根据实际情况调整)：

```
{
  "high_temperature_alarm":true,
  "show_detect_tip":false
 }
```

响应数据：

```
{
    "code": "000",
    "data": {
        "high_temperature_alarm": true
    },
    "msg": "success",
    "result": 1,
    "success": true
}
```

自定义配置清单:

key	**value**	**说明**
high_temperature_alarm	true	高温报警配置
show_detect_tip	false	隐藏识别提示配置
show_record_info	true	显示通行记录
printer_open	true	打印开关
printer_connect_type	1:蓝牙（默认） 2.USB	打印机连接类型
printer_label_size	1: 54x35 （默认） 2: 40x30	打印机标签尺寸
printer_stranger_open	true	陌生人打印开关
printer_snapshot_open	true	抓拍图打印开关
printer_name_open	false	姓名打印开关
printer_device_key_open	false	设备名打印开关
printer_temperature_open	false	温度打印开关
printer_company_name_open	false	公司名打印开关
printer_pass_time_open	false	通行时间打印开关
show_cfg_temp_fix_manual	true	显示设置中心手动温度补偿
show_dialog_temperature	false	显示弹窗温度
show_temperature_tip	true | false	显示额头温度
fixed_temperature	false	是否开启温度修正配置,默认为开启
hide_low_temperature_info	true	隐藏低温信息
close_upgrade	true	关闭升级配置
hide_device_info	true	隐藏底部设备信息
un_mask_open	true	未戴口罩开门
high_temperature_open	true	高温开门，正常不开门
img_base64_upload	false	imgBase64 上传开关
stranger_open	false	陌生人不开门配置
stranger_photo_unsave	true	陌生人不保存相片配置

op_stranger_normal_up	false	关闭陌生人测温正常数据上报配置
close_stranger_upload	true	配置关闭陌生数据上报
stranger_led	red | green	陌生人测温成功亮灯颜色配置
temperature_unit	2:华氏	切换显示和播报的温度（摄氏和华氏切换）
out_232	ttyS1	232 串口输出识别信息
record_save_days	数字（小于等于 30）	识别记录图片保存天数配置
ui	1:第一套 2:第二套	设备 UI 切换
wg	26 or 34, 默认 34	wg26 配置
screen_saver_wait_time	5~86400,默认 60	屏保等待时间,单位秒
fixed_max_temperature	(35 - 44.9)，默认 35.7	温度修正上限值配置
fixed_min_temperature	(25 - 34.9)，默认 33	温度修正下限值配置
temperature_adjust	-15~15	设置最小修正温度值
temperature_frame_open	false	测温框显示开关
login_password		登录密码

32. 设备播报信息

> v1.41.6.1+版本支持

请求地址：**http://SDK 中间件服务器 IP:8190/api/device/speakMessage**

请求方法：**POST**

请求数据：

参数名	数据类型	必填	说明
deviceKey	String	是	设备序列号
secret	String	是	设备密码
content	String	是	提示信息

请求说明：

- 播报内容不建议太多，避免影响设备正常使用

响应数据：

```
{
    "code": "000",
  "msg": "success",
  "success": true //成功
}
```

PostMan 示例：

33. 门禁规则同步

> v1.41.6.2+以上版本支持

请求地址：**http://SDK 中间件服务器 IP:8190/api/access/rule/sync**

请求方法：**POST**

请求数据：

参数名	数据类型	必填	说明
deviceKey	String	是	设备序列号
secret	String	是	设备密码
content	String	是	门禁规则

content 请求参数(示例，请根据实际情况调整)：

```
[
  {
```

//门禁日期类型 ， 节假日类型在 ext 中填写节假日日期

monday,tuesday,wednesday,thursday,friday,staturday,sunday,holiday1,holiday2,holi

day3

"dayName": "sunday",

// 节假日日期 MM-dd 格式，多日期逗号分隔 04-02,04-01

"ext": "",

//通行时段 HH:mm - HH:mm 格式，00：00-00:00 为全天禁止通行，

最多支持三个时间段，对应名为 ruleA，ruleB,ruleC

"ruleA": "00:00-23:59"

},

{

"dayName": "saturday",

"ext": "",

"ruleA": "00:00-23:59",

"ruleB": "00:00-00:00",

"ruleC": "00:00-00:00"

},

{

"dayName": "tuesday",

"ext": "",

"ruleA": "00:00-23:59",

"ruleB": "00:00-00:00",

```
    "ruleC": "00:00-00:00"
},
{
    "dayName": "holiday3",
    "ext": "",
    "ruleA": "00:00-23:59",
    "ruleB": "00:00-00:00",
    "ruleC": "00:00-00:00"
},
{
    "dayName": "holiday2",
    "ext": "",
    "ruleA": "00:00-23:59",
    "ruleB": "00:00-00:00",
    "ruleC": "00:00-00:00"
},
{
    "dayName": "holiday1",
    "ext": "04-02,04-01",
    "ruleA": "00:00-23:59",
    "ruleB": "00:00-00:00",
    "ruleC": "00:00-00:00"
```

```
},
{
    "dayName": "friday",
    "ext": "",
    "ruleA": "00:00-23:59",
    "ruleB": "00:00-00:00",
    "ruleC": "00:00-00:00"
},
{
    "dayName": "thursday",
    "ext": "",
    "ruleA": "00:00-23:59",
    "ruleB": "00:00-00:00",
    "ruleC": "00:00-00:00"
},
{
    "dayName": "wednesday",
    "ext": "",
    "ruleA": "00:00-23:59",
    "ruleB": "00:00-00:00",
    "ruleC": "00:00-00:00"
},
```

```
  {
    "dayName": "monday",
    "ext": "",
    "ruleA": "18:15-19:15",
    "ruleB": "00:00-00:00",
    "ruleC": "00:00-00:00"
  }
]
```


请求说明：

- 设置指定星期或日期的通行时间，一天最多设置三个通行时间段;
- content 设置为 [],清除门禁设置

响应数据：


```
{
  "code": "000",
  "msg": "success",
  "success": true //成功
}
```


PostMan 示例：

34. 掌纹添加（掌纹版本 apk 支持）

请求地址：**http://SDK 中间件服务器 IP:8190/api/palm/add**

请求方法：**POST**

请求数据：

参数名	数据类型	必填	说明
deviceKey	String	是	设备序列号
secret	String	是	设备密码
personId	String	是	人员 id
palmId	String	否	掌纹 id
feature	String	是	掌纹特征

请求说明：

- 必须先注册人员，才能添加掌纹
- 若 palmId 传入内容为空，则系统会自动生成一个 24 位的 palmId 并在注册成功后返回

响应数据：

```
{
    "data": "d7fe98be463c4d06a70605d4d58b5f33",//掌纹 id，注册掌纹都有一个 id，限制 24 位。若 palmId 不传，由系统分配的 palmId 为 24 位
    "code": "000",
  "msg": "success",
  "success": true //成功
}
```

PostMan 示例：

35. 掌纹删除（掌纹版本 apk 支持）

请求地址：**http://SDK 中间件服务器 IP:8190/api/palm/del**

请求方法：**POST**

请求数据：

参数名	数据类型	必填	说明
deviceKey	String	是	设备序列号
secret	String	是	设备密码
palmId	String	是	掌纹 id

响应数据：

```
{
    "code": "000",
  "msg": "success",
  "success": true //成功
}
```

PostMan 示例：

36. 掌纹清空（掌纹版本 apk 支持）

请求地址：**http://SDK 中间件服务器 IP:8190/api/palm/clear**

请求方法：**POST**

请求数据：

参数名	数据类型	必填	说明
deviceKey	String	是	设备序列号
secret	String	是	设备密码
personId	String	是	人员 id

响应数据：

```
{
    "code": "000",
    "msg": "success",
    "success": true //成功
}
```

PostMan 示例：

37. 掌纹查找（掌纹版本 apk 支持）

请求地址：**http://SDK 中间件服务器 IP:8190/api/palm/find**

请求方法：**POST**

请求数据：

参数名	数据类型	必填	说明
deviceKey	String	是	设备序列号
secret	String	是	设备密码
personId	String	是	人员 id

响应数据：


```
{
    "data": [
     {
       "palmId": "b8df54f2ef924c1f879e509b44593135", //掌纹 id
       "feature": "xxx",
       "personId": "12345abc" //人员 id
     }
    ],
       "code": "000",
    "msg": "success",
    "success": true //成功
}
```


PostMan 示例：

38. 掌纹远程采集（掌纹版本 apk 支持）

请求地址：**http://SDK 中间件服务器 IP:8190/api/palm/take**

请求方法：**POST**

请求数据：

参数名	数据类型	必填	说明

deviceKey	String	是	设备序列号
secret	String	是	设备密码
personId	String	是	人员 id

请求说明：

- 人员 id 必须为已存在；若不存在该人员 id，则无法进入自动采集模式
- 采集注册的过程实际上是设备先采集掌纹，再调用掌纹注册接口注册掌纹，palmId 由设备随机生成。
- 若设置了注册掌纹回调地址，掌纹注册成功后设备会将掌纹信息通过 POST 请求方式回调给预先设置的回调地址。

响应数据：

```
{
    "code": "000",
  "msg": "success",
  "success": true //成功
}
```

PostMan 示例：

39. 掌纹注册回调配置（掌纹版本 apk 支持）

请求地址：**http://SDK 中间件服务器 IP:8190/api/device/setPalmRegCallback**

请求方法：**POST**

请求数据：

参数名	数据类型	必填	说明
deviceKey	String	是	设备序列号

secret	String	是	设备密码
url	String	是	回调地址

请求说明：

- 回调 URL 需要符合正则表达式：**((http|https)://)((a-zA-Z0-9\._-{2,6})|(0-9{1,3}\.0-9{1,3}))(:0-9*)?**
- 设备掌纹注册实际上是设备先给人员采集掌纹，再调用掌纹注册接口，将采集的掌纹注册为人员的掌纹特征值。因此掌纹采集完成后，也会将采集的掌纹特征值进行回调。

响应数据：

```
{
    "code": "000",
  "msg": "success",
  "success": true //成功
}
```

注册掌纹回调参数说明：

参数名	数据类型	说明
deviceKey	String	设备序列号
personId	String	人员 ID
time	String	时间戳（以设备时间为准）
feature	String	掌纹特征值
palmId	String	掌纹 ID
ip	String	设备 IP 地址

PostMan 示例：

40. 图片设置配置

请求地址：**http://SDK 中间件服务器 IP:8190/api/device/configImg**

请求方法：**POST**

请求数据：

参数名	数据类型	必填	说明
deviceKey	String	是	设备序列号
secret	String	是	设备密钥
data	String	是	自定义配置数据

data 请求参数(示例，请根据实际情况调整)：

```
{
  "img_identify_success":
"iVBORw0KGgoAAAANSUhEUgAAAX0AAAF9CAIAAABnGy8GAAAAA3NC..", //
imgbase64
  "img_identify_fail":
"iVBORw0KGgoAAAANSUhEUgAAAX0AAAF9CAIAAABnGy8GAAAAA3NC..", //
imgbase64
  "img_identify_switch": true, // 开关
  "img_identify_duration": 3 // 显示时间
 }
```

响应数据：

```
{
    "code": "000",
    "data": {
        "ui": 2
    },
    "msg": "success",
    "result": 1,
    "success": true
}
```


图片配置清单:

key	value	说明
img_identify_success	图片 base64 字符串	识别成功显示图片
img_identify_fail	图片 base64 字符串	识别失败显示图片
img_identify_switch	true or false	识别结果图片显示开关
img_identify_duration	3，单位秒	识别结果图片显示时长

41. 问卷同步（问卷版本 apk 支持）

> 该接口需客户端支持，问卷定制版才具备

请求地址：**http://SDK 中间件服务器 IP:8190/api/vote/merge**

请求方法：**POST**

请求数据：

参数名	数据类型	必填	说明
deviceKey	String	是	设备序列号
secret	String	是	设备密钥
data	String	是	问卷内容

data 请求参数(示例，请根据实际情况调整)：

```
{
    //开始时间 时间戳
    "beginTime":1601481600000,
    //结束时间 时间戳
    "endTime":1604073600000,
    //问卷状态 true:开启问卷  false:关闭问卷
    "status":true,
    //题目
    "subjects":[
        {
            //选项 问卷类型为填空时, "options" : []
            "options":[
                {
                    //是否默认
                    "def":true,
                    //选项名
                    "name":"123",
```

```
        //选项 Id
        "optionId":"1",
        //是否是正确答案
        "right":false
    },
    {
        //是否默认
        "def":false,
        //选项名
        "name":"123",
        //选项 Id
        "optionId":"2",
        //是否是正确答案
        "right":true
    }
],
//题目 Id
"subjectId":"1",
//标题
"title":"002",
//类型: 1 单选 2 多选 3 填空
"type":1
```

```
        }
    ],
    //问卷名
    "title":"002",
    //问卷编号
    "voteCode":"002"
}
```

响应数据：

```
{
    "code": "000",
    "msg": "success",
    "ts": 1623118716240,
    "data": null,
    "success": true
}
```

42. 指纹添加（指纹版本 apk 支持）

请求地址：**http://SDK 中间件服务器 IP:8190/api/finger/add**

请求方法：**POST**

请求数据：

参数名	数据类型	必填	说明
deviceKey	String	是	设备序列号
secret	String	是	设备密码
data	String	是	指纹数据

请求说明：

- 必须先注册人员，才能添加掌纹
- fingerId 必须唯一，若 fingerId 传入内容为空，则系统会自动生成一个 32 位的 fingerId 并在掌纹注册成功后返回

data 请求参数示例：

```
[
  {
    "fingerId":"", // 指纹 id
    "personId":"", // 人员 id
    "fingerNum":"", // 指纹序号,11 左拇指 12 左食指 13 左中指 14 左无名指 15 左小指 21 右拇指 22 右食指 23 右中指 24 右无名指 25 右小指
    "feature":"" // 特征值
  },
  {
    "fingerId":"", // 指纹 id
    "personId":"", // 人员 id
```

"fingerNum":"", // 指纹序号,11 左拇指 12 左食指 13 左中指 14 左无名指 15 左小指 21 右拇指 22 右食指 23 右中指 24 右无名指 25 右小指

"feature":"" // 特征值

}

]

响应数据：

```
{
    "data": null,n:
    "code": "000",
  "msg": "success",
  "success": true //成功
}
```

PostMan 示例：

43. 指纹删除（指纹版本 apk 支持）

请求地址：**http://SDK 中间件服务器 IP:8190/api/finger/del**

请求方法：**POST**

请求数据：

参数名	数据类型	必填	说明

deviceKey	String	是	设备序列号
secret	String	是	设备密码
fingerId	String	是	指纹 id

响应数据：

```
{
    "code": "000",
  "msg": "success",
  "success": true //成功
}
```

PostMan 示例：

44. 指纹清空（指纹版本 apk 支持）

请求地址：**http://SDK 中间件服务器 IP:8190/api/finger/clear**

请求方法：**POST**

请求数据：

参数名	数据类型	必填	说明
deviceKey	String	是	设备序列号
secret	String	是	设备密码
personId	String	是	人员 id

响应数据：

{

"code": "000",

"msg": "success",

"success": true //成功

}

PostMan 示例：

45. 指纹查找（指纹版本 apk 支持）

请求地址：**http://SDK 中间件服务器 IP:8190/api/finger/find**

请求方法：**POST**

请求数据：

参数名	数据类型	必填	说明
deviceKey	String	是	设备序列号
secret	String	是	设备密码
personId	String	是	人员 id

响应数据：

{

"data": [

{

"fingerId": "b8df54f2ef924c1f879e509b44593135", //指纹 id

"fingerNum": "11", // 指纹序号,11 左拇指 12 左食指 13 左中指 14 左无名指 15 左小指 21 右拇指 22 右食指 23 右中指 24 右无名指 25 右小指

"feature": "xxxxx", //指纹特征值

"personId": "12345abc" //人员 id

}

],

"code": "000",

"msg": "success",

"success": true //成功

}

PostMan 示例：

46. 指纹注册回调配置（指纹版本 apk 支持）

请求地址：**http://SDK 中间件服务器 IP:8190/api/device/setFingerRegCallback**

请求方法：**POST**

请求数据：

参数名	数据类型	必填	说明
deviceKey	String	是	设备序列号
secret	String	是	设备密码
url	String	是	回调地址

请求说明：

- 回调 URL 需要符合正则表达式：**((http|https)://)((a-zA-Z0-9\._-{2,6})|(0-9{1,3}\.0-9{1,3}))(:0-9*)?**
- 设备指纹注册实际上是设备先给人员采集指纹，再调用指纹注册接口，将采集的指纹注册为人员的指纹特征值。因此指纹采集完成后，也会将采集的指纹特征值进行回调。

响应数据：

```
{
    "code": "000",
   "msg": "success",
   "success": true //成功
}
```

注册掌纹回调参数说明：

参数名	数据类型	说明
deviceKey	String	设备序列号
personId	String	人员 ID
time	String	时间戳（以设备时间为准）
feature	String	指纹特征值
fingerId	String	指纹 ID
fingerNum	Int	指纹序号,11 左拇指 12 左食指 13 左中指 14 左无名指 15 左小指 21 右拇指 22 右食指 23 右中指 24 右无名指 25 右小指
ip	String	设备 IP 地址

PostMan 示例：

47. 人员有效期添加

请求地址：**http://SDK 中间件服务器 IP:8190/api/person/expire_time/add**

请求方法：**POST**

请求数据：

参数名	数据类型	必填	说明
deviceKey	String	是	设备序列号
secret	String	是	设备密码
personId	String	是	人员 id
expireTime	String	是	人员有效期, 时间格式为 “yyyy-MM-dd HH:mm:ss”,例： "2020-11-11 11:11:11"

响应数据：

```
{
    "code": "000",
  "msg": "success",
  "success": true //成功
}
```

PostMan 示例：

48. 人员有效期删除

请求地址：**http://SDK 中间件服务器 IP:8190/api/person/expire_time/del**

请求方法：**POST**

请求数据：

参数名	数据类型	必填	说明

deviceKey	String	是	设备序列号
secret	String	是	设备密码
personId	String	是	人员 id

响应数据：

{

"code": "000",

"msg": "success",

"success": true //成功

}

PostMan 示例：

49. 人员列表查询

请求地址：**http://SDK 中间件服务器 IP:8190/api/person/list/find**

请求方法：**GET**

请求数据：

参数名	数据类型	必填	说明
deviceKey	String	是	设备序列号
secret	String	是	设备密码
index	Int	是	开始位置
length	Int	是	长度

响应数据：

{

"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）。

"personId": "12345", //人员 id，限制 64 位

"idCard": "xxx", //卡号，长度无限制

"name": "xxx", //姓名，不可为空

"expireTime": 1234567890123 // 过期时间，返回空则未设置

}],

"code": "000",

"msg": "success",

"success": true //成功

}

PostMan 示例：

50. 掌纹注册回调配置（新）

请求地址：**http://SDK 中间件服务器 IP:8190/api/device/setPalmRegCallbackNew**

请求方法：**POST**

请求数据：

参数名	数据类型	必填	说明
deviceKey	String	是	设备序列号
secret	String	是	设备密码

url	String	是	回调地址

请求说明：

- 回调 URL 需要符合正则表达式：**((http|https)://)((a-zA-Z0-9\._-{2,6})|(0-9{1,3}\.0-9{1,3}))(:0-9*)?**
- 设备掌纹注册实际上是设备先给人员采集掌纹，再调用掌纹注册接口，将采集的掌纹注册为人员的掌纹特征值。因此掌纹采集完成后，也会将采集的掌纹特征值进行回调。

响应数据：

```
{
    "code": "000",
   "msg": "success",
   "success": true //成功
}
```

注册掌纹回调参数说明：

参数名	数据类型	说明
deviceKey	String	设备序列号
personId	String	人员 ID
time	String	时间戳（以设备时间为准）
feature	String	掌纹特征值
palmId	String	掌纹 ID
ip	String	设备 IP 地址

PostMan 示例：

51. 掌纹查找（新）

请求地址：**http://SDK 中间件服务器 IP:8190/api/palm/findNew**

请求方法：**POST**

请求数据：

参数名	数据类型	必填	说明
deviceKey	String	是	设备序列号
secret	String	是	设备密码
personId	String	是	人员 id

响应数据：

```
{
    "data": [
     {
       "palmId": "b8df54f2ef924c1f879e509b44593135", //掌纹 id
       "feature": "xxx",
       "personId": "12345abc" //人员 id
     }
    ],
       "code": "000",
    "msg": "success",
    "success": true //成功
}
```

PostMan 示例：

52. 掌纹添加（新）

请求地址：**http://SDK 中间件服务器 IP:8190/api/palm/addNew**

请求方法：**POST**

请求数据：

参数名	数据类型	必填	说明
deviceKey	String	是	设备序列号
secret	String	是	设备密码
personId	String	是	人员 id
palmId	String	否	掌纹 id
feature	String	是	掌纹特征

请求说明：

- 必须先注册人员，才能添加掌纹
- 若 palmId 传入内容为空，则系统会自动生成一个 24 位的 palmId 并在注册成功后返回

响应数据：

```
{
    "data": "d7fe98be463c4d06a70605d4d58b5f33",//掌纹 id，注册掌纹都有一个 id，限制 24 位。若 palmId 不传，由系统分配的 palmId 为 24 位
    "code": "000",
  "msg": "success",
  "success": true //成功
}
```

PostMan 示例：

53. 掌纹删除（新）

请求地址：**http://SDK 中间件服务器 IP:8190/api/palm/delNew**

请求方法：**POST**

请求数据：

参数名	数据类型	必填	说明
deviceKey	String	是	设备序列号
secret	String	是	设备密码
palmId	String	是	掌纹 id

响应数据：

```
{
        "code": "000",
    "msg": "success",
    "success": true //成功
}
```

PostMan 示例：

附录

接口返回码说明

返回码	说明
000	请求成功
400	请求参数验证失败
401	签名校验失败

403	权限设置原因拒绝请求
500	系统繁忙，请稍后重试(系统异常统一返回 500)
1010	设备重启失败, un root.
2000	人脸创建异常
2010	删除人脸失败