

---

# 人脸识别终端 HTTP 接口

## V 1.3

---

---

## 目录

<b>1. 概述</b> .....	<b>3</b>
1.1 简述.....	3
1.2 接口规范.....	3
<b>2. 版本介绍</b> .....	<b>4</b>
<b>3. 接口说明</b> .....	<b>4</b>
3.1 秘钥接口.....	4
3.2 设置设备基础信息.....	4
3.3 获取设备基础信息.....	5
3.4 设置设备时间.....	5
3.5 获取设备时间.....	6
3.6 设置过闸配置信息.....	6
3.7 获取过闸配置信息.....	7
3.8 获取访客过闸配置信息.....	7
3.9 设置访客过闸配置信息.....	8
3.10 创建过闸人员信息.....	8
3.11 根据 ID 获取过闸人员信息.....	9
3.12 获取所有过闸人员信息.....	9
3.13 根据 ID 删除过闸人员信息.....	10
3.14 获取过闸流水日志.....	11
3.15 删除过闸流水日志.....	11
3.16 设置过闸数据自动上传的 URL.....	12
3.17 获取过闸数据自动上传的 URL.....	12
3.18 重启终端设备.....	13
3.19 开闸.....	13
3.20 配置心跳 URL.....	13
3.21 获取心跳 URL.....	14
3.22 更新终端 APK.....	14
3.23 检查照片是否可用、重复.....	15
3.24 拍照.....	15
3.25 恢复出厂设置(未启用).....	16
<b>4. 状态码表</b> .....	<b>17</b>

---

---

# 1. 概述

## 1.1 简述

本接口协议是针对运行在局域网内的终端设备，开发人员可以通过HTTP 协议直接访问设备。

## 1.2 接口规范

- 接口地址：http://设备 ip:8088/
- 字符集：**UTF-8**
- 接口安全：第一次调用接口时使用秘钥接口更新终端设备的秘钥，以后每次每个接口都需要验证更新后的秘钥。默认秘钥为 abc。
- 返回数据格式：JSON 字符串
- 返回字段描述：

字段	描述	类型	必须	详细说明
status	状态码	int	是	0-成功，其他状态见状态码表
msg	状态描述	String	是	状态码的详细描述，比如：成功
data	返回数据	String(json)	否	返回请求的数据，比如：请求人员列表，则 data 即为人员列表的 json 字符串

## 2. 版本介绍

协议版本	发布日期	协议说明
V1.0	2018-10-16	初始版本
V1.1	2018-10-20	更改拍照接口返回字段名
V1.2	2018-10-29	<p>修复时间设置单位问题； 扩展设备设置的字段； 修复拍照功能输出图片</p> <p><b>1.setDeviceInfo 接口：</b> A. 修改字段。 faceFeaturePairSuccessOrFailWaitTime、openDoorContinueTime 的类型为 double； B. 添加字段appFailMsg(识别失败提示语)。</p> <p><b>2.getDeviceInfo 接口：</b> 添加返回字段 appFailMsg。</p> <p><b>3.setTime 接口：</b> 修改字段ts 类型为double, 单位为秒。</p> <p><b>4.getTime 接口：</b> 修改字段 ts 类型为 double, 单位为秒。</p> <p><b>5.setPerson 接口：</b> A. 修改字段startTs 和endTs 无限制时传-1 即可。 B. 修改字段 startTs 和 endTs 类型为 double, 单位为秒。</p> <p><b>6.getPerson 接口：</b> 修改返回字段startTs 和endTs 类型为 double, 单位为秒。</p> <p><b>7.listPerson 接口：</b> 修改返回字段startTs 和endTs 类型为 double, 单位为秒。</p> <p><b>8.listRecord 接口：</b> 修改字段 ts(包括返回字段)类型为 double, 单位为秒。</p> <p><b>9.removeRecord 接口：</b> 修改字段ts 类型为double, 单位为秒。</p> <p><b>10.setRecordCallback 接口：</b> 修改过闸数据上传字段 ts 为 double, 单位为秒。</p>
V1.3	2018-11-20	

---

---

		<p><b>11. setHeartBeat 接口:</b></p> <ul style="list-style-type: none"><li>A. 修改字段 period 类型为 double</li><li>B. 修改心跳数据上传字段 deviceElapsedRealtime 的类型为 double, 单位为秒。</li></ul> <p><b>12. 添加返回状态码 127、128</b></p> <p><b>13. setDeviceKey 接口:</b></p> <p>修改字段 oldKey 命名为 key。</p>
--	--	--

## 3. 接口说明

### 3.1 密钥接口

- **URI:setDeviceKey**
- **method:POST**
- **参数列表:**

字段	描述	类型	必传	详细说明
key	旧密钥	String	是	已经存在于终端设备的密钥，默认密钥为 abc
newKey	新密钥	String	是	即将配置到终端设备的密钥，最长不超过 20 个 byte.

- 返回示例:

```
{  
  "status": 0,  
  "msg": "ok"  
}
```

### 3.2 设置设备基础信息

- **URI:setDeviceInfo**
- **method:POST**
- **参数列表**

字段	描述	类型	必传	详细说明
key	密钥	String	是	终端设备的密钥
cameraDetectType	活体判断	Int		0:活体, 1 非活体(活体需要双目摄像头支持)
faceFeaturePairNumber	人脸比对阈值	float		人脸比对时候的阈值, 默认 0.6
faceFeaturePairSuccessOrFailWaitTime	人脸比对后等待时间	double		人脸识别成功或失败后等待的时间(单位:秒)
openDoorType	开门条件	Int		0:人脸, 1:身份证, 2:工号, 3:人脸+身份证, 4:人脸+工号(根据具体机器)
openDoorContinueTime	开门持续时间	double		识别成功后开门持续时间(单位:秒)
doorType	开门类型	Int		0:继电器 26:26 维根 34:34 维根 默认为 26 维根

idCardFaceFeaturePairNumber	身份证比对阈值	float	身份证比对阈值(机器需要对应模块)
appWelcomeMsg	识别成功提示	String	识别成功提示声音(需要设备联网认证成功后提示(不是连接后台,而是能上网)), 例:识别成功,@(人名替代符)
deviceSoundSize	设备声音	Int	设备声音大小(0-100)
deviceDefendTime	设备维护时间	String	指定时间设备自动重启 = 23:50
deviceName	设备昵称	String	设备昵称, 显示在设备首页
appFailMsg	提示语	String	终端识别失败后的提示语

- 返回示例:

```
{
  "status": 0,
  "msg": "ok"
}
```

### 3.3 获取设备基础信息

- URL:getDeviceInfo

- method:GET

- 参数列表

字段	描述	类型	必传	详细说明
key	秘钥	String	是	终端设备的秘钥

- 返回示例:

```
{
  "msg": "成功",
  "status": 0,
  "data": {
    "cameraDetectType": 100,
    "doorType": 100000,
    "APKVersion": "1.1.103",
    "sn": "f269bb519fe282c5",
    "idCardFaceFeaturePairNumber": 0.8,
    "deviceDefendTime": "21:50",
    "openDoorContinueTime": 5.0,
    "faceFeaturePairNumber": 0.2,
    "deviceSoundSize": 0,
    "faceFeaturePairSuccessOrFailWaitTime": 3.0,
    "version": "1",
    "name": "1号机",
  }
}
```

```

        "openDoorType": 0,
        "appWelcomeMsg": "请通行,@",
        "appFailMsg": "抱歉，识别失败"
    }
}

```

### 3.4 设置设备时间

- **URI:setTime**
- **method:POST**
- **参数列表**

字段	描述	类型	必传	详细说明
key	密钥	String	是	终端设备的密钥
ts	时间	double	是	设置设备的时间，单位：秒

- **返回示例：**

```

{
    "status": 0,
    "msg": "ok"
}

```

### 3.5 获取设备时间

- **URI:getTime**
- **method:GET**
- **参数列表**

字段	描述	类型	必传	详细说明
key	密钥	String	是	终端设备的密钥

- **返回示例：**

```

{
    "status": 0,
    "msg": "ok",
    "data": {
        "ts": 1539666601.235
    }
}

```

### 3.6 创建过闸人员信息

- **URI:setPerson**
- **method:POST**
- **参数列表**

字段	描述	类型	必传	详细说明
key	秘钥	String	是	终端设备的秘钥
id	人员 ID	String	是	ID, 验证人员唯一性的字段
name	名称	String		人员的名字
IC_NO	卡号	String		作为维根号发送到闸机
ID_NO	身份证号	String		员工身份证号
photo	照片	Base64		照片要求宽高 640x480 像素, 转为 Base64 字符串传输
passCount	最大过闸次数	long		一共可以过闸的最大次数 [0, 10000); 当此值设置等于 10000 时, 代表可以无限次过闸, 默认:10000
startTs	起始时间	double		此时间之后拥有过闸权限, 单位:秒 (传-1 表示无限制), 默认无限制
endTs	截止时间	double		此时间之后无过闸权限, 单位:秒 (传-1 表示无限制), 默认无限制

- 返回示例:

```
{
  "status": 0,
  "msg": "ok"
}
```

### 3.7 根据ID 获取过闸人员信息

- URI:getPerson
- method:GET
- 参数列表

字段	描述	类型	必传	详细说明
key	秘钥	String	是	终端设备的秘钥
id	人员 ID	String	是	ID, 验证人员唯一性的字段

- 返回示例:

```
{
  "msg": "成功",
  "status": 0,
  "data": {
    "startTs": 1542595445,
    "id": "1",
    "endTs": 1542595445.123,
    "name": "胖虎 1",
    "passCount": 10000,
  }
}
```

```

        "ID_NO": "990307189207026735",
        "photo": "( data : image base64)",
        "IC_NO": "7267e840"
    }
}

```

### 3.8 获取所有过闸人员信息

- URI:listPerson
- method:GET
- 参数列表

字段	描述	类型	必传	详细说明
key	密钥	String	是	终端设备的密钥

- 返回示例:

```

{
  "msg": "成功",
  "status": 0,
  "data": {
    "person": [{
      "IC_NO": "7267e840",
      "name": "胖虎 1",
      "passCount": 10000,
      "startTs": 0.0,
      "id": "1",
      "endTs": 0.0,
      "ID_NO": "990307189207026735"
    }, {
      "name": "foobar",
      "passCount": 10000,
      "startTs": 0.0,
      "id": "123457",
      "endTs": 0.0
    }
  ]
}

```

### 3.9 根据ID删除过闸人员信息

- URI:removePerson
- method:POST
- 参数列表

字段	描述	类型	必传	详细说明
key	密钥	String	是	终端设备的密钥

id	人员 ID	String/jsonArray	是	删除人员（可单人、批量、全部） [id1, id2, ..., idn], 如传入 [] 这删除全部
----	-------	------------------	---	---

- 返回示例:

```
{
  "msg": "成功",
  "status": 0
}
```

### 3.10 获取过闸流水日志

- URI: listRecord
- method: GET
- 参数列表

字段	描述	类型	必传	详细说明
key	密钥	String	是	终端设备的密钥
ts	时间戳	double	是	返回此时间戳之后的数据（不包括此时间, 单位: 秒）
count	记录次数	int	是	返回过闸记录数据条数

- 返回示例:

```
{
  "msg": "成功",
  "status": 0,
  "data": {
    "record": [
      {
        "personID": "10000",
        "ts": 1539666562.0,
        "passType": 0,
        "name": "zhang san"
      },
      {
        "personID": "10001",
        "ts": 1539666562.0,
        "passType": 0,
        "name": "li si"
      }
    ]
  }
}
```

## 3.11 删除过闸流水日志

- **URI:removeRecord**
- **method:POST**
- **参数列表**

字段	描述	类型	必传	详细说明
key	密钥	String	是	终端设备的密钥
ts	时间戳	double	是	删除ts之前的过闸记录（包括ts，单位：秒）

- **返回示例:**

```
{
  "msg": "成功",
  "status": 0
}
```

## 3.12 设置过闸数据自动上传的URL

- **URI:setRecordCallback**
- **method:POST**
- **参数列表**

字段	描述	类型	必传	详细说明
key	密钥	String	是	终端设备的密钥
url	上传数据的url	String	是	终端会自动根据url以POST方式上传过闸记录，上传字段与listRecord接口返回数据一致

返回参数说明:

1. deviceSn :设备 sn,唯一值
2. personID :人员 id
3. ts :人员通过时间
4. passType :人员通过类型
5. photo :人员通过照片

http 服务器返回数据到终端

{status:0,data:""} status = 0,终端上传成功后删除记录,否则保留记录

- **返回示例:**

```
{
  "msg": "成功",
  "status": 0
}
```

---

---

### 3.13 获取过闸数据自动上传的URL

- **URI: getRecordCallback**
- **method: GET**
- **参数列表**

字段	描述	类型	必传	详细说明
key	密钥	String	是	终端设备的密钥

- **返回示例:**

```
{
  "msg": "成功",
  "status": 0,
  "data": {
    "url": "http://ip:port/"
  }
}
```

### 3.14 重启终端设备

- **URI: reboot**
- **method: POST/GET**
- **参数列表**

字段	描述	类型	必传	详细说明
key	密钥	String	是	终端设备的密钥

注意:有 3000 毫秒延时

- **返回示例:**

```
{
  "msg": "成功",
  "status": 0
}
```

### 3.15 开闸

- **URI: open**
- **method: POST/GET**
- **参数列表**

字段	描述	类型	必传	详细说明
key	密钥	String	是	终端设备的密钥

- 返回示例:

```
{
  "msg": "成功",
  "status": 0
}
```

### 3.16 配置心跳 URL

- **URI:setHeartBeat**
- **method:POST**
- **参数列表**

字段	描述	类型	必传	详细说明
key	秘钥	String	是	终端设备的秘钥
url	心跳 url	String	是	终端根据 url 以 <b>POST</b> 方式访问服务器, 可能还会上传其他字段
period	心跳周期	double	是	每两次心跳间隔时间, 单位:秒, 大于 0

心跳请求参数说明:

1. deviceElapsedRealtime :设备开机到现在时间
2. deviceSystemVersion :设备系统版本
3. deviceSn :设备 SN 码
4. deviceAppVersion :APP 版本
5. deviceRomAvailableSize :SD 卡剩余可用空间
6. deviceRomSize :SD 卡总空间
7. deviceIpAddress :设备 IP 地址

- 返回示例:

```
{
  "msg": "成功",
  "status": 0
}
```

### 3.17 获取心跳 URL

- **URI:getHeartBeat**
- **method:GET**
- **参数列表**

字段	描述	类型	必传	详细说明
key	秘钥	String	是	终端设备的秘钥

- 返回示例:

```

{
  "msg": "成功",
  "status": 0,
  "data": {
    "period": 50,
    "url": "http://192.168.0.66:8888"
  }
}

```

### 3.18 更新终端 APK

- URI:updateAPK
- method:POST
- 参数列表

字段	描述	类型	必传	详细说明
key	密钥	String	是	终端设备的密钥
url	升级 apk 的 url	String	是	终端会以GET方式访问URL 下载 apk 更新

- 返回示例:

```

{
  "msg": "成功",
  "status": 0
}

```

### 3.19 检查照片是否可用、重复

- URI:check
- method:POST
- 参数列表

字段	描述	类型	必传	详细说明
key	密钥	String	是	终端设备的密钥
photo	照片	String/Base64	是	检测照片是否可用（能找到人脸、能提取特征值） 检测照片是否重复（根据阈值匹配特征值）

- 返回示例:

```

{
  "msg": "成功",
  "status": 0,
}

```

```

    "data": {
      "isValid": 0,
      "isExist": false
    }
  }
}

```

- **返回参数:**

isValid: int, 0-照片可用 1-尺寸错误 2-格式错误 3-找不到人脸 4-特征值提取失败

isExist: boolean, true-已经存在此人（不一定是同一张照片） false-不存在此人

## 3.20 拍照

- **URI:photo**

- **method:POST**

- **参数列表**

字段	描述	类型	必传	详细说明
key	密钥	String	是	终端设备的密钥
tipsBefore	拍照前提示语	String		
tipsAfter	拍照后提示语	String		
count	连拍次数	int		取值 1, 2, 3, 默认 1

注意:

1. 拍照每次间隔 1000 毫秒

2. 程序初始化/人员添加/人员删除等提示存在时不要拍照(APP 底部显示)

3. 拍照时并不会显示人脸框

- **返回示例:**

```

{
  "msg": "成功",
  "status": 0,
  "data": {
    "photos":["Base64", "Base64", ...]
  }
}

```

## 3.21 恢复出厂设置(未启用)

- **URI:reset**

- **method:POST**

- **参数列表**

字段	描述	类型	必传	详细说明
----	----	----	----	------

key	密钥	String	是	终端设备的密钥
-----	----	--------	---	---------

● 返回示例:

```
{
  "msg": "成功",
  "status": 0
}
```

## 4. 状态码表

状态码	描述
0	成功
1	未知错误
100	客户端错误
101	缺少参数
102	参数错误
103	密钥验证失败
104	图片保存失败
105	图片尺寸错误
106	图片找不到人脸
107	passCount 格式错误
108	passCount 取值范围错误
109	startTs 格式错误
110	startTs 取值范围错误
111	endTs 格式错误
112	endTs 取值范围错误
113	IDFaceRate 格式错误
114	IDFaceRate 取值范围错误
115	faceRate 格式错误
116	faceRate 取值范围错误
117	passField 格式错误

---

---

118	passField 取值范围错误
119	passType 格式错误
120	passType 取值范围错误
121	period 取值范围错误
122	url 格式错误
123	正在拍照
124	图片格式错误
125	图片提取特征值失败
126	人员不存在
127	图片解析失败
128	人员达到最大限度，无法继续添加
1000	服务端错误
10000	服务端未知错误