人脸识别app对接后台接口

V1.1

# 1. 接口规范

## 1.1 HTTP接口

* 接口地址：http://服务端ip:端口/cpfr/app
* 测试地址：http://132.232.108.138/cpfr/app
* 接口描述：通过GET/POST发送请求，连接到后台
* 接口要求：全部通过POST来发送请求，请求时需要将access\_app\_token=xx放入请求头中
* 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 |
| code | 状态码 | int | 是 | 0成功 -1失败 其他见状态码表 |
| message | 状态描述 | String | 是 | 详细描述 |
| data | 返回数据 | String(json) | 是 | 返回请求的数据 |

## 1.2 WebSocket接口

* 接口地址：ws:// 服务端ip:端口/cpfr/ws
* 测试地址：ws://132.232.108.138/cpfr/ws
* 接口描述：通过WebSocket与后台建立长连接，用于后台主动推送
* 接口要求：建立连接时需要将device\_sn=xxx放入请求头中。
* 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 |
| code | 状态码 | int | 是 | 1000成功 999失败 其他见状态码表 |
| message | 状态描述 | String | 是 | 详细描述 |
| data | 返回数据 | String(json) | 是 | 返回请求的数据 |

# 2. 接口说明

## 2.1 HTTP

### 2.1.1注册设备(请求头不用放入access\_app\_token)

* URI：/device\_register
* method：POST
* 请求头：
* 参数列表：
  + device\_sn=xx
* 返回示例：

## {"code":0,"message":"设备已注册","data":null}

### 2.1.2 获取设备信息

* URI：/device\_info
* method：POST
* 请求头：access\_app\_token=xx
* 参数列表：
  + lmt=xx （本地最新时间）
* 返回示例：

{"code":0,"message":"操作成功","data":{"device\_name":"abcdefg123456","lmt":1542952862000,"device\_id":5,"device\_sn":"abcdefg123456","mac\_grant\_key":"aaaaaaaaa"}}

### 2.1.3获取人员列表

* URI：/person\_list
* method：POST
* 请求头：access\_app\_token=xx
* 参数列表：
  + lmt=xx （本地最新时间）
* 返回示例：

{"code":0,"message":"操作成功","data":[{"lmt":1543815472000,"base\_image":"","emp\_number":"","person\_name":"疯","person\_id":1},{"lmt":1543808226000,"base\_image ":"","emp\_number":"","person\_name":"ccc","person\_id":21}]}

### 2.1.4 获取人员对应权限列表

* URI：/grant\_list
* method：POST
* 请求头：access\_app\_token=xx
* 参数列表：
  + lmt=xx （本地最新时间）
* 返回示例：

{"code":0,"message":"操作成功","data":[{"lmt":1542857308000,"pass\_start\_time":-1,"pass\_end\_time":-1,"pass\_number":-1,"type":0,"person\_id":1},{"lmt":1542331821000,"pass\_start\_time":-1,"pass\_end\_time":-1,"pass\_number":-1,"type":1,"person\_id":21}]}

### 2.1.5 上传识别记录

* URI：/upload\_record
* method：POST
* 请求头：access\_app\_token=xx
* 参数列表：
  + person\_id=xx
  + recog\_type=xx
  + record\_time=xx
  + pass\_number=xx
  + file=xx
* 返回示例：
* {"code":0,"message":"操作成功","data":null}

### 2.1.6 app端注册人员

* URI：/add\_person
* method：POST
* 请求头：access\_app\_token=xx
* 参数列表：
  + person\_name=xx
  + emp\_number=xx
  + group\_name=xx
  + file=xx
* 返回示例：
* {"code":0,"message":"操作成功","data":null}

### 2.1.7 获取服务器时间

* URI：/current\_date
* method：POST
* 返回示例：
* {"code":0,"message":"操作成功",{"current\_date":""}}

### 2.1.8 比较服务器的人员数量，返回未下载的人员列表

* URI：/compare\_downl\_num
* method：POST
* 请求头：access\_app\_token=xx
* 参数列表：
  + person\_ids=‘1,2,3,4,5’（逗号分隔）
* 返回示例：
* {"code":0,"message":"操作成功"," data":null }

### 2.1.9 获取新版本信息

* URI：/last\_version\_info
* method：POST
* 请求头：access\_app\_token=xx
* 返回示例：

{"code":0,"message":"操作成功","data":{"apk\_offline\_version":"101","apk\_online\_version":"201"}}

### 2.2.0 下载离线版本apk

* URI：/downl\_offline\_apk
* method：POST
* 请求头：access\_app\_token=xx

### 2.2.1 下载在线版本apk

* URI：/downl\_online\_apk
* method：POST
* 请求头：access\_app\_token=xx

## 3.2 WebSocket

### 3.2.1显示设备在线

* 建立websocket后，设备的状态即时在线，断开websocket后设备下线。
* 请求头：
  + device\_sn=xx

### 3.2.2 接收码

* 返回示例：

## {"code":1000,"message":"操作成功","data":null}

|  |  |  |
| --- | --- | --- |
| code | message | data |
| 999 | 操作失败 | null |
| 1000 | 操作成功 | null |
| 1001 | 激活成功 | {"access\_app\_token":"","arcface\_appid":"","arcface\_sdkkey":"","baidu\_tts\_appid":"","baidu\_tts\_appkey":"","baidu\_tts\_secretkey":""} |
| 1002 | 修改设备 | null |
| 1003 | 修改人员 | null |
| 1004 | 修改通行 | null |
| 1005 | 设备删除 | null |
| 1006 | 连接成功 | {"access\_app\_token":"","arcface\_appid":"","arcface\_sdkkey":"","baidu\_tts\_appid":"","baidu\_tts\_appkey":"","baidu\_tts\_secretkey":""} |
| 1007 | 未激活 | null |