数据同步应用开发文档

# 引言

此应用采用Python开发，需要开发人员具备以下的基础知识：

1. Python编程基础
2. Django Web开发基础
3. 一定的html/css/javascript基础
4. http协议基本概念

# **需求分析**

数据同步应用的主要作用是作为整个应用的后端部分，存储和分析相应的数据，并为Android终端提供通用的网络接口。

|  |  |  |  |
| --- | --- | --- | --- |
| 功能名称 | | | 备注 |
| 人脸数据录入 | 单个录入 | |  |
| 批量录入 | | 后期功能 |
| 查询 | |  |
| 修改 | |  |
| 删除 | |  |
| 去重复 | | 后期功能 |
| Android录入接口，包含增、删、改、查 | | Android终端主动调用 |
| 终端管理 | 心跳保持 | | Android终端主动调用 |
| 设备管理 | |  |
| 权限验证 | 用户权限验证 | |  |
| 数据备份 | 定期备份 | |  |
| 数据同步 | 人脸数据同步 | 版本差异计算 |  |
| 增量式数据更新 | Android终端主动调用 |
| 识别记录同步 | 记录上传 | Android终端主动调用 |
| 记录计算 | 后期功能 |

注：去重复功能的作用是保证数据库中无重复的人员数据，保证人员的唯一性；记录计算功能的主要作用为从收集到的数据中，生成需要的员工出勤记录等报表内容。

# 设计方案

## 2.1 整体思路

采用Python Django框架进行后端服务器的开发，使用WebSocket完成网页与后台的实时交互（主要是终端监控功能）。Android终端主动调用服务器的网络接口，无需服务器主动向Android终端推送数据。

## 详细设计

### 人脸数据录入

录入功能包括员工数据的增、删、改、查，这些功能在任何一个内容管理系统中都属于基础功能，只需要定义好员工的数据库字段即可。

|  |  |  |
| --- | --- | --- |
| 序号 | 字段名 | 备注 |
| 1 | id | 自动生成的整型主键 |
| 2 | uuid | 自动生成的uuid，全局唯一 |
| 3 | name | 姓名 |
| 4 | position | 职位 |
| 5 | image | Base64编码图片 |

### 数据接口

由于接口包含增、删、改、查，因此REST接口是最符合需求的，直接使用第三方库djangorestframework生成网站的REST接口，并添加自定义的内容。

### 心跳保持

为了监控当前连接至服务器的Android终端，使用第三方库channels为服务器增加实时数据推送的能力。Android终端定期调用心跳保持接口，服务器将Android终端的信息实时发布到终端监测功能页面上。

### 终端管理功能

使用与人脸数据相同的内容管理方式管理Android终端，提供相应的管理页面。终端管理的数据库字段如下：

|  |  |  |
| --- | --- | --- |
| 序号 | 字段名 | 备注 |
| 1 | id | 自动生成的整型主键 |
| 2 | uuid | 自动生成的uuid，全局唯一 |
| 3 | name | 终端名 |
| 4 | device\_type | 设备类型，目前可选项为“入”和“出” |
| 5 | description | 终端描述信息 |

### 用户权限验证

用户权限验证系统基于Django自带的权限验证系统，并基于djangorestframework拓展基于token的权限验证。

### 人脸数据同步

数据同步的目的是为了让Android终端和服务器数据库的人脸库保持一致，若无时无刻都返回全部数据，则会降低工作效率，并且大量的返回数据可能造成服务器堵塞。因此，令某一时刻，服务器所处的当前状态记为服务器的数据版本，每一个时刻只能处于某一个版本，版本号从0开始依次递增。服务器记录每一个版本与前一个版本的数据差异。Android终端进行同步时，可以选择同步全部数据或者是增量同步数据。例如，令当前版本号为0，新增一个员工1，版本号变成1；再增加一个员工2，版本号变成2；修改员工2的个人信息，版本号变成3。若同步全部数据，则服务器会返回员工1和员工2的全部信息；若Android终端当前处于版本1，增量同步时会返回修改后的员工2的信息。

### 识别记录同步

识别记录同步的数据流向为单向，即只从Android终端流向服务器，因此在服务器设置一个网络接口接受Android终端POST过来的数据即可。

# 代码说明

## 3.1 目录结构



pengfeng：项目的配置信息

rest：基本的model定义和REST API

sync：数据同步接口的定义和实现，终端监测接口的定义和实现

incremental\_update：一些接口交互数据的例子

functional\_tests：集成测试代码，目前基本没有作用

enhanced\_ui：自定义的网站页面，主要包含人脸数据的增删改查

## 部署说明

此应用可以直接使用Docker构建镜像和部署镜像，具体指令在根目录下的README.md中可以找到。注意，命令中的某些参数可能会需要变动。