许 一清

QQ:2023503307

摘要

采用Python的Flask框架，并结合Sqlite数据库以提高性能

Opera AUDIO服务器端

基于Python、Flask、Sqlite

目录

[简介 1](#_Toc34293059)

[目录介绍 2](#_Toc34293060)

[如何部署 3](#_Toc34293061)

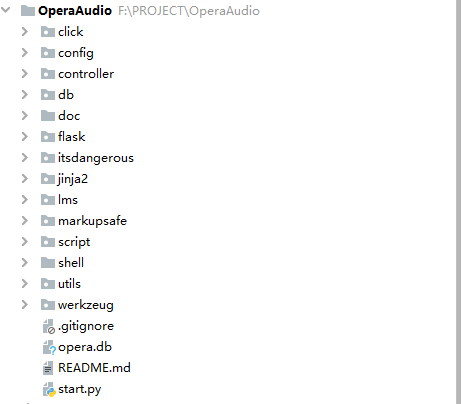
[重点难点解释 5](#_Toc34293062)

[常见问题解决 14](#_Toc34293063)

## 简介

该项目分为前端和服务器端（类似传统网站的前后端）。前端功能主要是APP的界面，与用户交互，传递数据到服务器端等，前期采用***Flutter***技术完成（黄亮负责），后期采用其他方式实现（张天怡负责）。服务器端使用***Python***语言的***Flask***框架，进行配置，调用LMS的API，将关键信息存入数据库等

## 目录介绍



打开OperaAudio目录，启动该项目是使用**start.py**，命令就是**python start.py**，其他文件夹大多都是一些依赖的库文件，gitignore文件和README.md文件是github文件，无需关心，opera.db是该项目的数据库文件（sqlite）；需要开发者关心的目录是：

（1）**config**：记录了配置信息，一些常量

（2）**controller**：类似于后端开发中的controller层，这里没有区分service层，全部写入了controller；这部分也是核心部分

（3）**doc**：存放一些文档，实际部署可以删除本目录

（4）**lms**：调用LMS的API需要用到的脚本，已经完全写好，可以直接调用，使用方法可以参照脚本的注释

（5）**script**：进行配置的一些脚本，比如如何挂载，如何配置WIFI，如何配置服务等，本质是调用Linux命令或者编辑配置文件

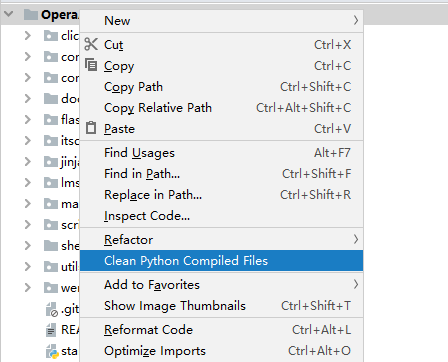
（6）**shell**：记录了如何将启动脚本配置到服务中，参照说明进行配置即可成功

（7）**utils**：一些简单的工具脚本

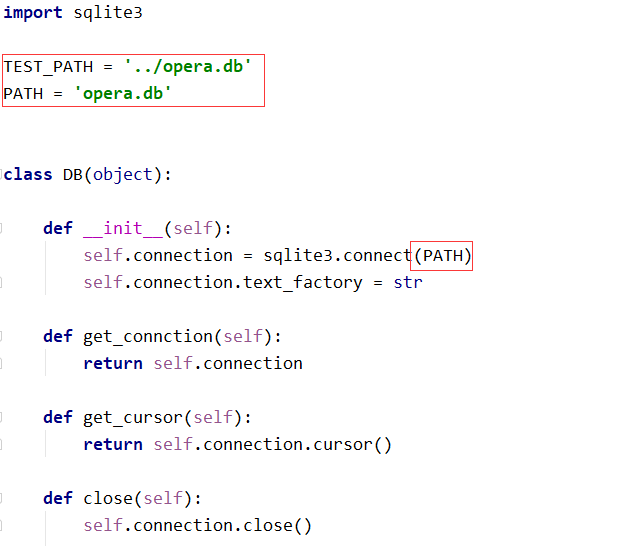
（8）**db**：数据库相关，为了一些功能的高效性，采用数据库保存关键信息，提高用户体验

## 如何部署

为了方便，直接将整个文件夹拷贝到目标设备的root目录下，拷贝前最好删除预编译文件pyc（以Pycharm为例），在目标机器上重新编译：



另外修改db下core.py文件，修改connect函数的参数为PATH：

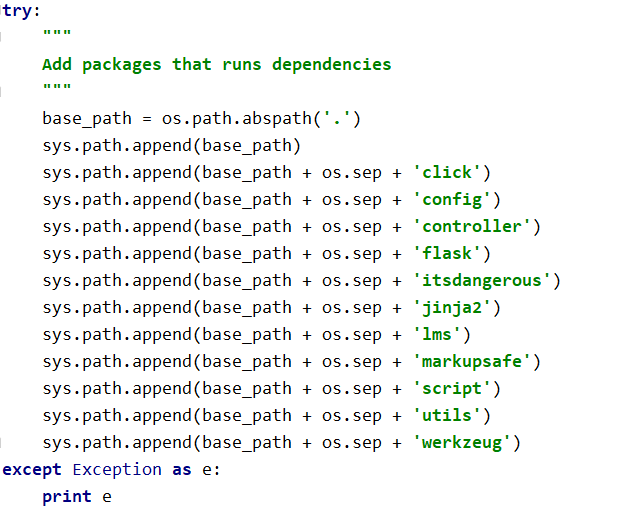


简单的部署（用于开发）：将整个文件夹复制过去，可以使用各种工具（例如Xshell的SFTP、图形化界面的WinSCP）。然后进入/root/OperaAudio，使用命令python start.py即可简单地启动项目，如果对Linux机器的连接断开，服务器端也将停止工作。

正式的部署（用于测试和实际使用）：将整个文件夹复制过去，进入OperaAudio目录，找到shell目录，里面详细写着如何将start.py设置为系统的服务，配置成功之后，即使断开连接，也不会停止工作。

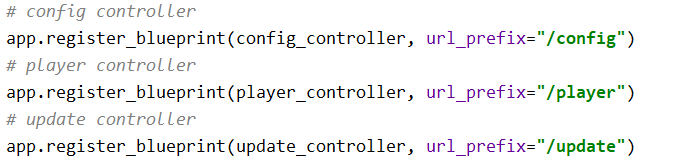
## 重点难点解释

（1）start.py中，一开始写了如下图这些：



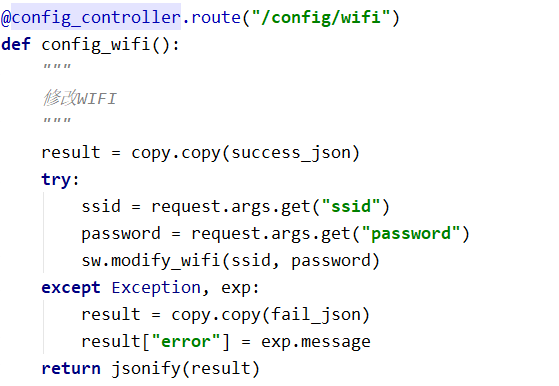
因为项目需要***Flask***框架，所以要各种依赖。我为了简化部署，将所有依赖写入代码之中，因此不需要使用**pip**命令下载依赖，然后配置环境等一系列复杂的操作。

（2）start.py中，有这样的一段代码：



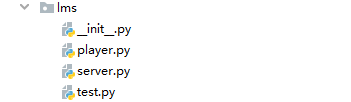
理解起来很简单，/config开头的请求一律交给controller/config\_controller来管理，以此类推，将三个controller分开来写，方便开发者。

（3）controller目录下的文件，其实都是一个规范：

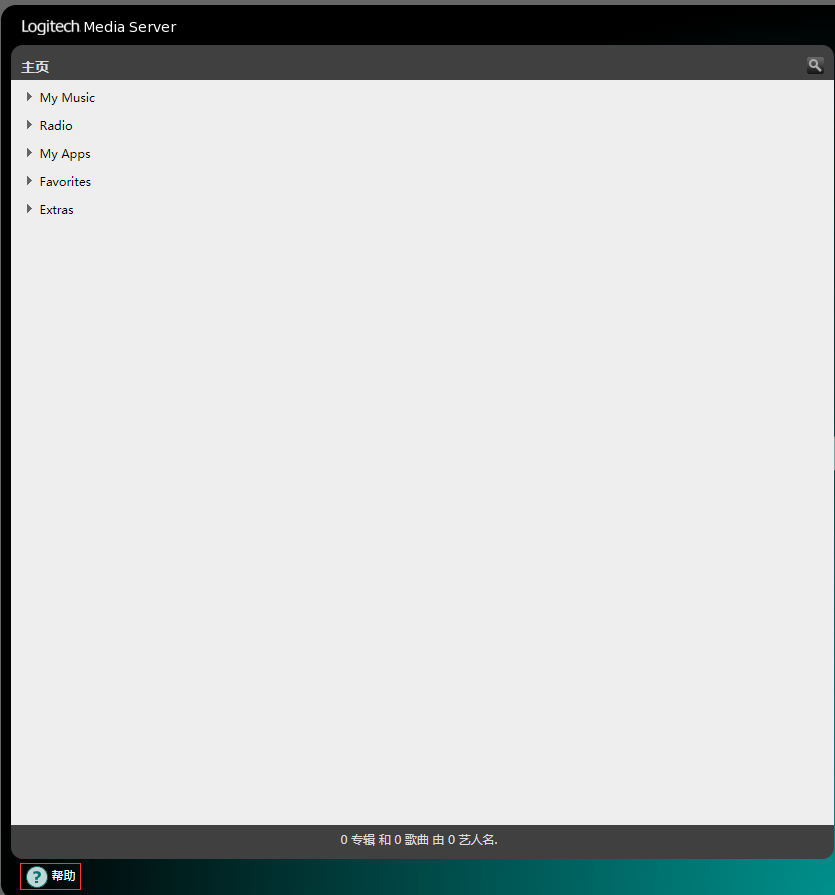


第一行类似java中的**@RequestMapping(“/config/wifi”)**；全局有一个success\_json的定义，但是我不能直接使用它，需要使用copy进行复制一个；request也是一个全局的变量，**request.args.get**就是获取请求的参数；这里没有返回数据，只返回了成功或失败的响应，如果要返回数据，直接编辑result即可（例如**result[“data”]=data**）；最后一行的**jsonify**函数是将Python的字典转为json格式。

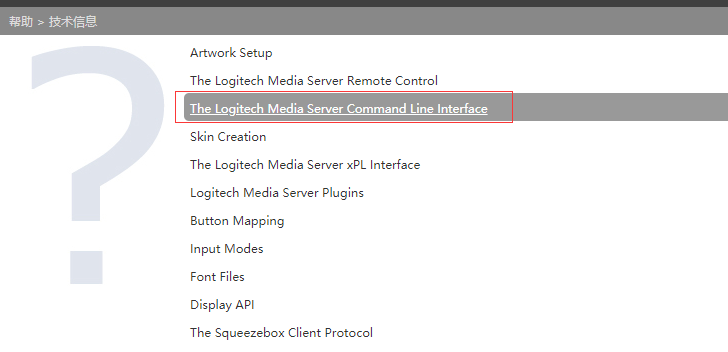
（4）LMS目录下的三个文件是该项目的核心：



脚本player.py是播放器的API：控制播放，暂停，播放列表等功能；脚本server.py是服务器的API：对歌曲管理，连接LMS等功能；脚本test.py主要用于测试API。具体的内容其实不复杂，就是使用telnet访问目标的9090端口，返回相应的字符串，再进行解析。这里是参考LMS官方的文档而做的：官方文档位于9000端口的页面左下角，有个帮助和支持的入口，进入之后，存在有很多的API文档。由于是全英文的，以下我尽量写一下我个人的理解，方便开发者快速上手。（下图为寻找官方文档的方式）

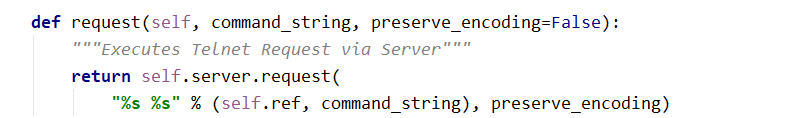




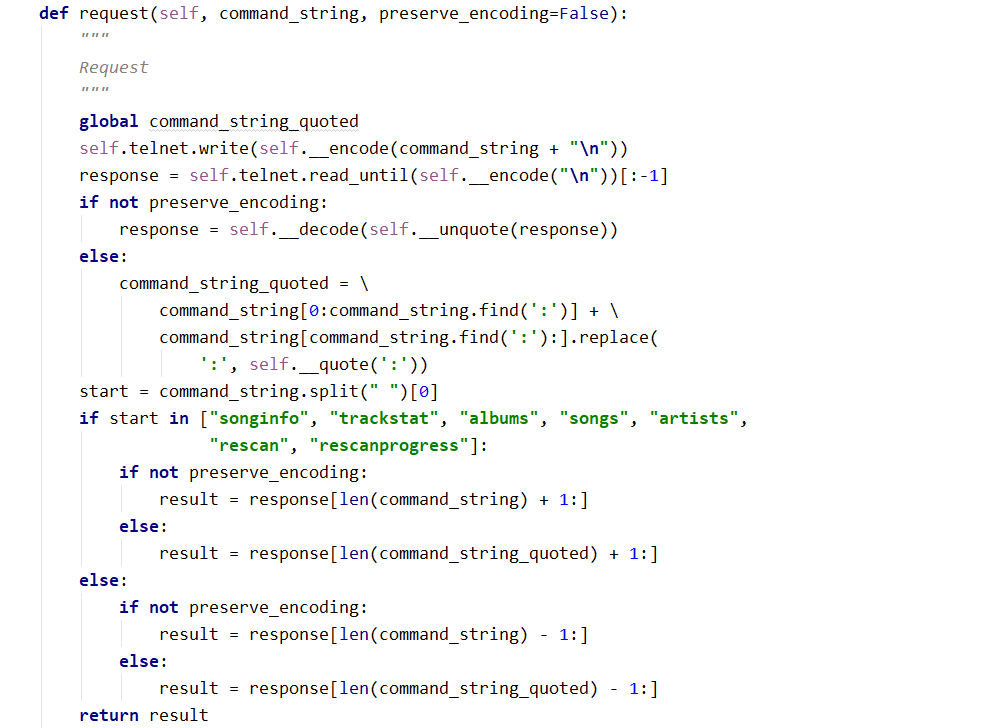


以下我将简单地分析player.py和server.py文件：

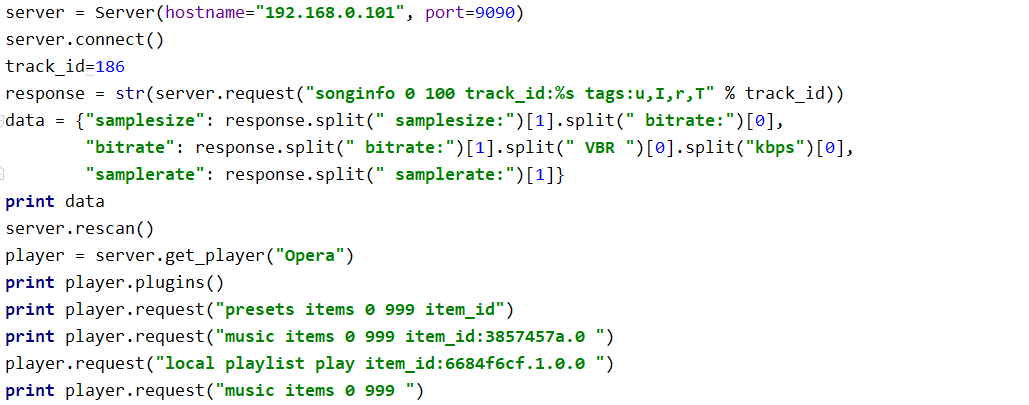
打开player.py文件，会发现整个文件就是一个类，我将一些方法和参数都封装到了一个类中，仔细观察后可以发现所有的函数几乎都是调用一个request函数，比如下图为player.py中的request函数，调用了server.py的request函数：



下图为server.py中的request函数，首先是使用telnet库发送请求，LMS的9090端口为telnet客户端，然后得到LMS的响应（这里需要进行转码的，telnet的编码和日常使用的字符串不太一样），处理后返回。另外有一个函数是：request\_with\_results，其实也是基于request函数的，至少做了进一步处理，使用的话，两种都可以，直接request然后手动处理响应字符串也是一种不错的方式，我一开始是手动处理的，没有使用封装的方法，个别地方使用了封装的方法，这导致可能存在部分垃圾代码，后来的开发者可以使用封装方法来优化一下老版本的代码，不过经过我的测试发现，封装好的request方法还是存在BUG的，很多问题都需要开发者自己去解决。



以下我将介绍如何使用LMS的API，查看test.py：



开头是一堆的依赖配置和导包，这里不截图了。首先是定义一个Server，需要目标的IP和端口，也就是开启LMS的telnet客户端。做任何操作都需要连接的，连接成功之后输入请求字符串，会得到响应，然后处理字符串即可；值得一提的是：player不是定义出来的，而是使用server.get\_player(“这里输入player的名称”)；需要测试新的命令的时候，直接在这里即可，参照我写的方式。

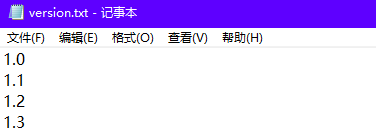
接下来我分析一下命令：以test.py中的命令为例，**songinfo 0 100 track\_id:%s tags:u,I,r,T**这一句话中，songinfo是查询歌曲信息，0 100是查询一百个（应该是从第一个到第一百个），如果数量不足100那么就返回实际个数；track\_id是歌曲id，作为参数传递进来的；tags是需要获取的内容，每一个字母代表信息，以下为我从LMS文档截取的内容：

**I samplesize Song sample size (in bits)**

也就是说字母I表示samplesize，歌曲采样率，其他三个字母在文档中都有写实际的作用，注意区分大小写，英文逗号分隔。

（5）配置脚本（script目录下的脚本的解释）：

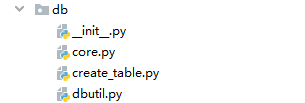
关机和重启直接调用命令即可；各种服务的开关使用service xxx start命令、查询使用service xxx status命令、自启相关使用systemctl enable xxx等命令；mout挂载是编辑rc.local启动文件（直接使用mount命令也是可以的，不过组长要求在开机启动文件中进行配置），使用mount命令；更新这里我在阿里云服务器上简单地跑着一个Tomcat（没有部署任何java项目，只是一个tomcat），放了一个txt写入版本信息，并模拟存储了多个版本，如下图：



更新流程是：读取到txt版本信息,我在tomcat的webapp下建了一个文件夹download，同时在里面保存了版本信息txt和对应的版本，获得版本信息：http://39.106.125.199:8080/download/version.txt，用户选择版本，然后下载对应版本的文件（是压缩成ZIP的文件），如果下载的版本是1.1版本，那么该更新调用的url是：<http://39.106.125.199:8080/download/OperaAudio-1.1.zip>，然后使用unzip命令进行解压，完成后重启设备（由于已经注册到服务并设置了开机自启，这里没有问题），这里我是覆盖了原文件，也可以删除掉所有文件再解压，Linux和Windows不同，正在使用中的文件也是可以删除的。

（6）软件的模式选择问题：进入APP后需要选择基本版和极致版，两种模式的功能不相同；简单理解就是：基本模式是有所有功能的模式，极致模式是把所有功能分到两个机器上进行配置，所以需要扫描到两台机器，分别进行管理和配置。这里具体如何进行拆分可以咨询梅华威老师。

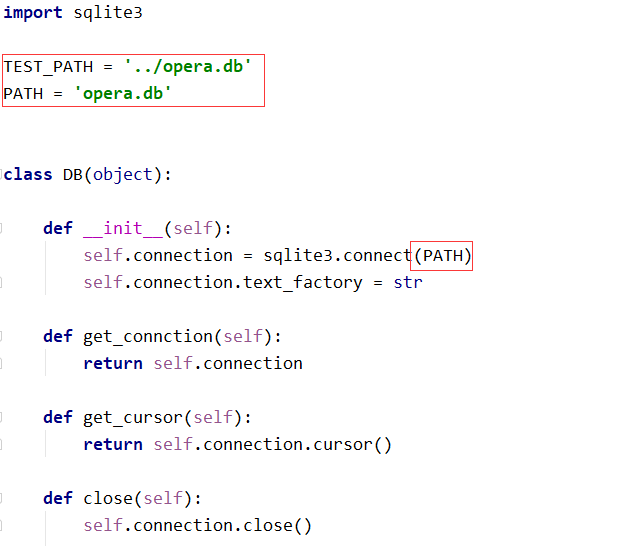
（7）数据库地方的问题，目录如下：



core文件是连接数据库的类，使用方式如下：



create\_table文件是数据库创建表的文件，一般情况下无需使用，只有根目录的opera.db被删除或者损坏后，手动执行一遍create\_table即可，注意，同时必须修改core文件connect函数参数，将PATH为TEST\_PATH：



dbutil文件主要是封装了数据库的增删改查函数，可以直接调用，注释写的比较全面

## 常见问题解决

1. 设备如果卡死，或者无法播放，LMS连接无响应，一般重启设备可以解决。
2. 极致版本中，先启动装有LMS客户端的机器，确定正常运行之后，再启动装有Squeezelite的机器。（具体情况和原理请咨询梅华威老师）