## 5、前后端交互

## 1.管理员系统和后端交互

（1）AdminController

登录管理员用户，验证登录，根据用户输入得到的用户名和密码查询数据库，验证该用户是否存在与数据库中。



在前端的Login.vue实现

**submitForm**: 提交表单的方法，包括发送登录请求、处理返回结果和跳转路由等操作。

（2）SingerController

删除歌手信息

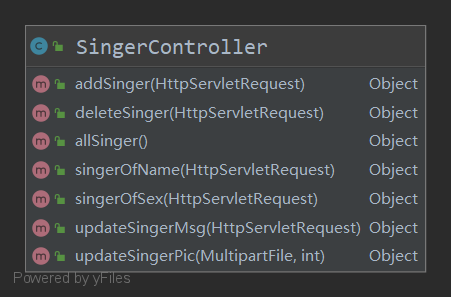
查找歌手信息

添加歌手信息

更新歌手头像

更新歌手信息

在service/impl/SingerService里面定义一个名为 SingerService 的公共接口，查找歌手信息、通过boolean addSinger()方法添加歌手信息，更新歌手信息，更新头像，最后根据歌手ID删除歌手信息，根据获取的歌手列表查询数据库，验证歌手信息及更新是否在数据库中。



（3）歌曲管理SongController

删除歌曲信息

查找歌曲信息

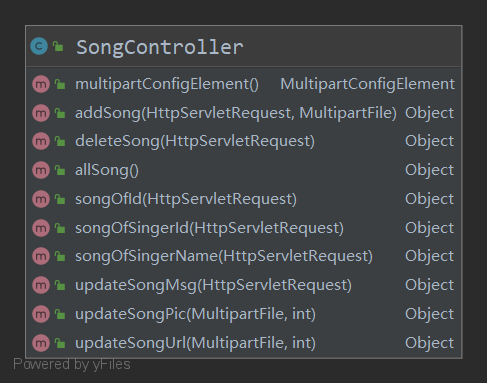
添加歌曲信息

更新歌曲信息

更新歌曲图片

更新歌曲文件

在controller/SongController里面查找出歌手信息，通过AddSong()方法添加歌曲，更新歌曲信息，更新图片，更新歌曲文件，删除歌曲信息在更新一次歌曲信息，根据获取的歌曲信息查询数据库进行交互，查看歌曲信息修改是否成功。



（4）ConsumerController

删除用户信息

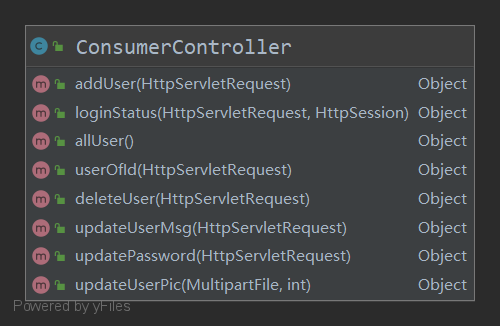
查找用户信息

添加用户信息

更新用户头像

更新用户信息

通过/user/updata更新用户信息，查找用户信息，添加用户信息，更新用户头像，更新用户信息，删除用户信息后再次对用户信息进行更新，依据得到的用户信息与数据库进行交互，以此来查看修改用户信息是否成功。



（5）SongListController

删除歌单信息

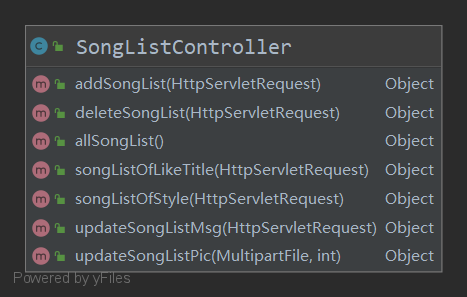
查找歌单信息

添加歌单信息

更新歌单头像

更新歌单信息

在controller/SongListController中查找歌单信息，依据查找到的歌单信息添加歌单信息，更新歌单头像，更新歌单信息，删除歌单信息后再次更新歌单信息，依据最后得到的歌单信息查询数据库，验证歌单信息是否在数据库中。



## 2.音乐系统页面和后端交互

## 普通用户

RankListController

查看歌单信息

查看歌曲信息

歌单评分

在controller/RankListController中查看歌单信息，查看歌曲信息，歌单评分，在数据库中查询，查看是否交互成功。

播放歌曲

循环播放、上一首、暂停、下一首、收藏、下载

CollectControlle

收藏歌曲

取消收藏歌曲

在controller/CollectController中收藏歌曲，取消收藏，是否收藏，更新收藏的歌曲，在数据库中查看是否取消收藏成功。

CommentController

提交评论：

a. 添加评论，用户查看歌单之后可以发表自己的观点，对歌单进行评论。

b. 显示所有的评论信息，当用户查看歌单的具体内容时，将所有的评论显示至该歌单的下方。

删除评论：

 a.删除评论，当用户思考之后觉得该评论有不符合的地方需要进行删除时，则根据评论编号将该评论的信息从数据库中删除。

 b. 根据歌单编号查看所有与该歌单有关的评论，用于删除歌单的时候先将与该歌单有关的评论先删除。

系统配置

系统开发 JDK + Windows + Maven

开发语言 JavaEE + vue

后端框架Springboot

前端Vue

数据库和工具Mysql + Navicat

开发工具 Idea VSCode

游览器Microsoft Edge

系统搭建步骤

1创建后台springboot并启动测试

2导入并编译前端代码vue：网站和后台管理

编译：npm install

试运行：npm run dev

3设计数据库

4正式开发