**网络音乐播放小程序作业总结及说明**

**作业说明：**

本系统使用Taro框架进行开发， Taro 是一个开放式跨端跨框架解决方案，支持使用 React/Vue/Nerv 等框架来开发 微信 / 京东 / 百度 / 支付宝 / 字节跳动 / QQ 小程序 / H5 / RN 等应用。现如今市面上端的形态多种多样，Web、React Native、微信小程序等各种端大行其道。当业务要求同时在不同的端都要求有所表现的时候，只编写一套代码就能够适配到多端的能力就显得极为需要。

Taro编译框架也是得遵循小程序的规则的，基本在运用api，路由以及组件上与小程序无异，**只是在代码风格不太一样，小程序的wxml更类似原生的html，taro的就相当于在html中嵌套js代码**

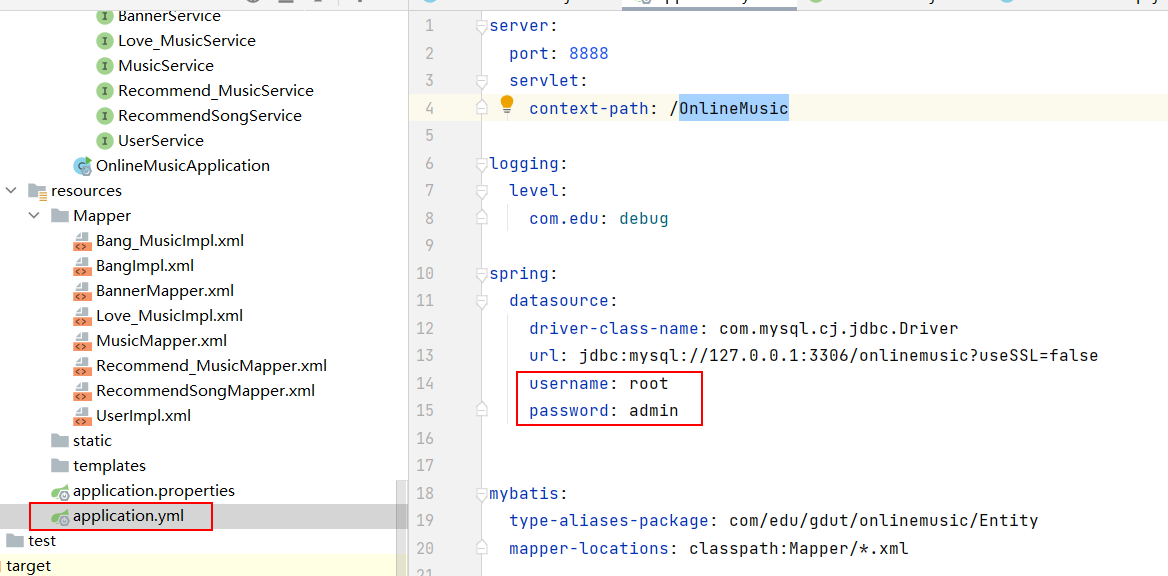
如果老师想测试一下taro(必须有taro环境)，可以打开“taro源文件”,首先输入命令yarn install下载依赖包，下载完成之后就用yarn dev:weapp编译出小程序代码，在开发者工具中导入生成的dist目录即可

**项目说明：**

本项目完成了**用户登录注册，可搜索展示播放音乐，用户资料的显示修改，最近播放和收藏歌曲，音乐的播放支持播放、暂停、播放进度条（支持拖动）、快进等**，能够实现音乐程序的大部分功能。本项目采用了前后端分离的技术，前端使用Taro框架，后端使用springboot，交互使用axios，UI组件库使用TaroUI，在音乐数据的查找方面使用了python爬虫程序，爬取酷我音乐的音乐数据，将数据下载到本地以及插入数据库

**使用说明：**

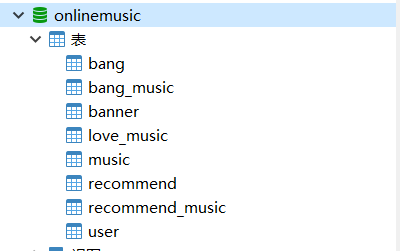
**1、首先需要修改后端的数据库配置**



**2、修改本地图片以及视频和用户头像存放的地址**



**3、打开navicat，运行onlinemusic.sql文件，生成如图所示表**

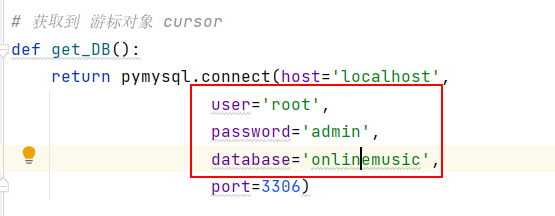


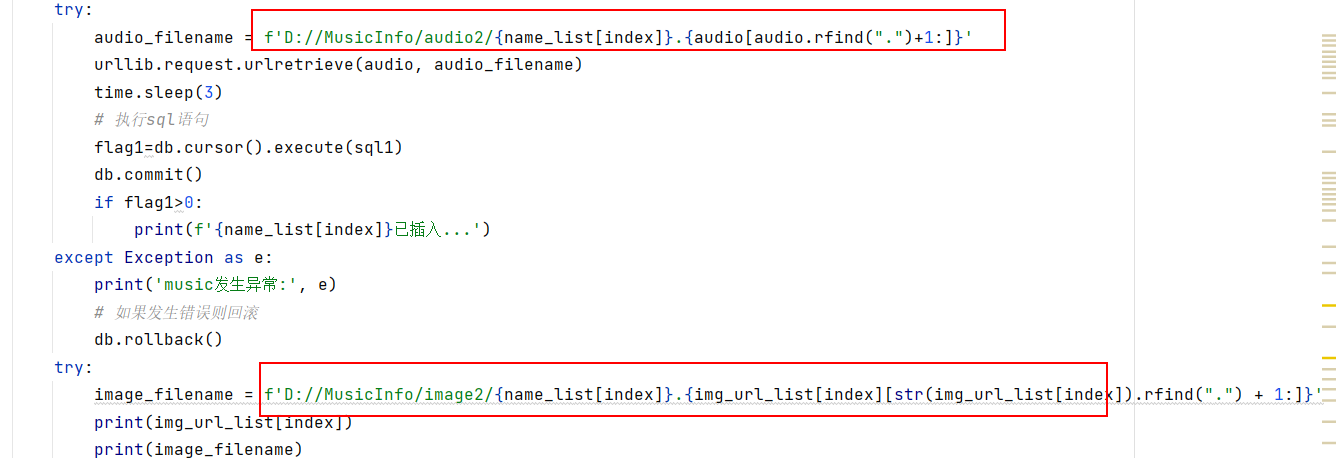
**4、开发者工具导入frontend中的文件，idea导入backend中的文件，运行后端项目**



**5、使用pycharm打开爬虫文件**

首先要修改DBUtil中数据库的配置，修改存放下载的资源的路径







**注意：由于访问酷我服务器太频繁，可能会导致远程服务器强制关闭当前的连接，所以最好图片和音频分开下载，并且酷我的服务器有些许不稳定，所以只将链接插入到数据库可能会导致访问有些问题**

**6、下载redis，以及python包**

python下载：

1、pip install json，

2、pip install requests，

3、pip install redis，

4、pip install jsonpath

去redis官网下载redis，测试项目的时候，**必须开启redis server服务器**

**7、打开backend后端项目，修改配置**

需要修改redis的本地配置，以及kg\_search.py所在的位置



**8、本地测试需要修改本地的音乐资源存放位置以及访问路径！**



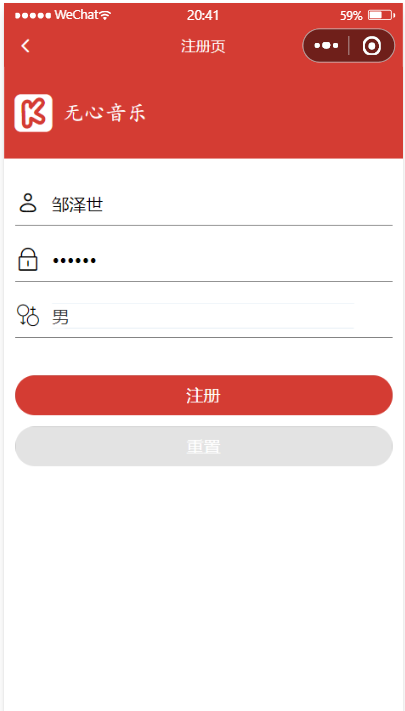
**流程演示：**

**1、用户界面**

首先，到我的页面进行注册登录

点击马上注册，跳转到注册页面，填写好基本信息注册成功后会弹出模态框回显账号信息，点击确定，跳转到登录页，自动填写账号，填写登录信息，登陆完成

在用户界面，还可以完善并修改个人的信息

**在播放音乐以及收藏音乐后，分别会在“最近播放”模块和“收藏模块”中展示歌曲信息，点击更多后，展示更多详细信息**



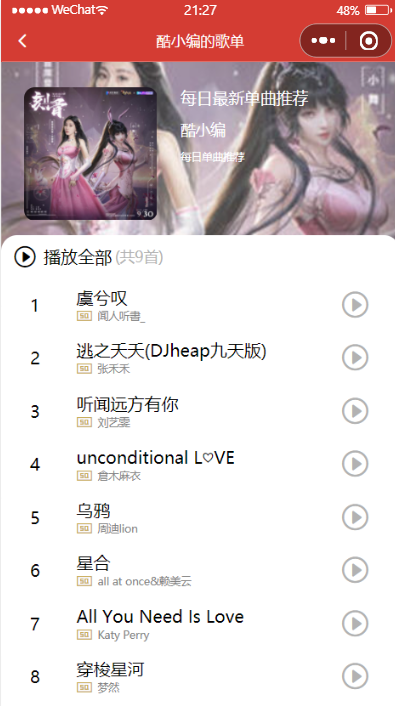
**2、音乐“发现”界面**

完成了**音乐搜索，轮播图展示热门信息，ScrollView展示推荐歌单，滑动视图模块展示各种排行榜**的信息

其中，搜索功能对接了酷狗音乐的真实搜索接口，每次搜索，将搜索数据传递到后端，由后端将搜索关键字写入到redis中，然后调用命令行执行爬虫程序，爬虫程序直接从redis中获取搜索关键字，模拟浏览器向酷我的搜索接口发送请求。这样，无论搜索什么歌曲，只要酷狗有的，我这里都可以搜索出来，而且还可以直接把数据插入到数据库中，这样，只要使用的人数够多，我的歌曲数据库就会越加充实。

缺点就是，有一些vip歌曲被做了限制，只能播放到一半。在代码进程上由于需要等待搜索进程执行完后才会继续往下执行，如果python出错，有进程阻塞的危险



**3、播放音乐界面**

**进度条支持拉动，能快进或快退，转盘能转动，能收藏歌曲，自由暂停和开始**



**遇到的问题：**

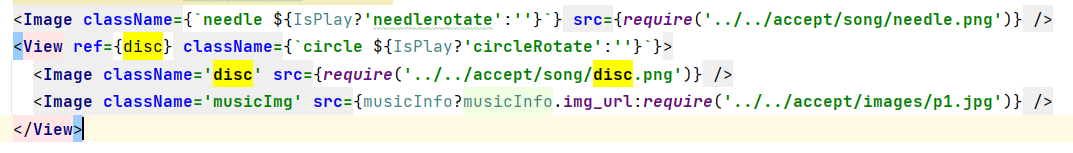
**1、自定义导航栏**

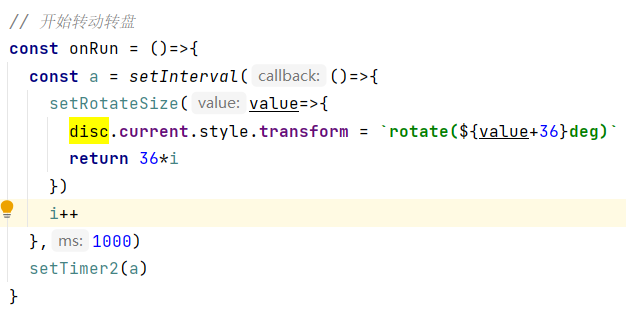
如以上的演示过程，我使用的都是自定义的导航栏，支持展示搜索框搜索，颜色，返回或者关闭等图标等功能



**2、音乐播放的圆盘转动以及控制问题**

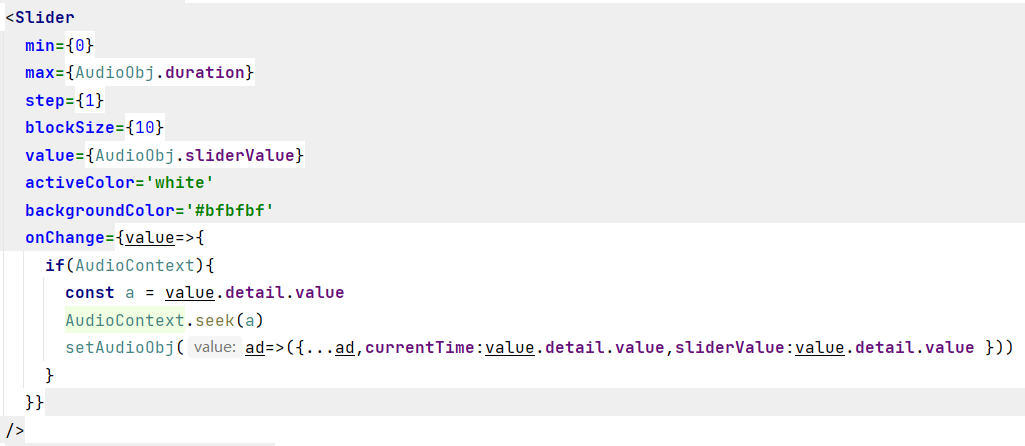
因为涉及到转动的角度以及每次暂停时保留之前转动的角度的问题，所以需要一个外部变量来存放转动的圈数，每次转动36度，通过ref来操作元素，给transform属性加一个rotate旋转角度，并给它加一个transition动画，速度为liner的平滑过渡效果，再开一个定时器，每秒自增一次，这样就解决了转动的以及控制问题





**3、音乐时长以及进度条的兼容问题**

因为音乐时长是xx:xx没60s进1的形式，但是进度条却是以每10进1的形式，为了避免复杂的换算，所以直接把音乐时长换算成秒，把总秒数作为进度条的最大值，再将步长设置为1即可



**4、上传头像，将本地资源上传到服务器**

上传文件，需要用到wx. chooseMessageFile方法，因为使用chooseImage会直接把文件保存到一个临时路径，这样后端就获取不到真实的文件名了，所以使用

chooseMessageFile方法，把文件以及用户id和资源名称传到后端



**5、音乐数据获取问题**

由于网易云的开放接口不好找，所以我打算自己写一个爬虫程序，先仔细对比接口的参数，再将各个接口进行整合，爬取酷我音乐的数据。然后由于酷我的服务器不太稳定，导致访问它的图片资源总会失败，所以我把他下载下来，保存到本地，在springboot中添加访问本地资源的方法，然后将访问的服务器路径直接在爬取到数据的时候存到数据库。

**注意：本地测试需要修改本地的音乐资源存放位置！**



**6、解决触摸穿透的问题**

做了这样一个功能，点击/长按/滑动menu的字母模块，能有提示展示的效果。



但是有一个问题，就是我滑动它下拉的时候，外层的ScrollView也在跟着滑动，这也就是触摸穿透的问题，经过查找发现，加入catchMove这个属性可以有效地阻止事件的冒泡，官方的解释也是，只要 catchMove 属性值为 true，就会使用 catchtouchmove 代替 bindtouchmove 进行事件绑定，从而获得阻止滑动穿透的能力。

**7、访问DOM节点的方法**

我们都知道，在react中，可以使用ref这个属性来访问到dom节点，但是在taro中这个有时不是特别的奏效，**因为ref访问到的是taro的虚拟dom节点，是不能获取到小程序真实的宽高之类的属性的**，所以我们得使用原生的**createSelectorQuery方法**，他能帮助我们访问小程序渲染层的 DOM 节点

经过测试，在useEffect中使用的话，可能会获取不到节点的信息返回个null，我猜测可能是useEffect执行的时间比节点渲染更早，但是有时确能获取到，一般加个setTimeout定时器过个200ms也能获取到。但是一般推荐在useReady中使用**createSelectorQuery方法**，毕竟是小程序适配的。





一般就是这样操作的，在回调的res参数中能获取到该节点的信息，可以做到打开页面直接滑动到该节点的功能

**个人总结**

这次的网络音乐播放小程序的开发学习到了很多，尤其是在页面的布局，组件的封装，以及前后端，java与python等的交互流程。在组件的封装过程中，能很好地保留组件的特点，也留给了一定的自定义空间给使用者发挥，我觉得这个是我做的比较好的地方，还有就是爬虫的使用，爬虫的编代码过程还是比较复杂的，不但需要去克服目的网址的反爬手段，还要对获取来的数据做一定的数据筛选操作。虽然我使用的是Taro框架，但是通过这次的开发，我也对小程序的原生生命周期，各种原生的api有了更深入的了解，而且我觉得我这次的作业整体做的还是比较好的，用到了很多如Taro，springboot，redis，python爬虫，TaroUI这些技术，也通过这次的实操，将这些技术很好地融合了在一起