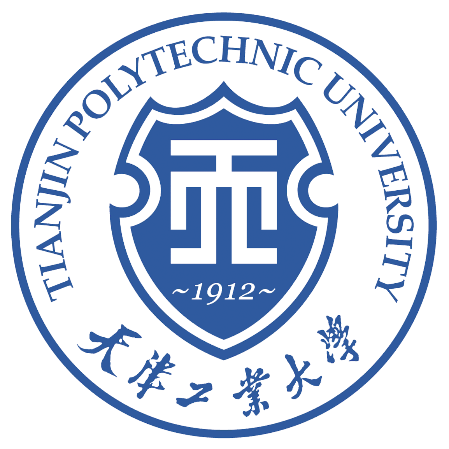
天津工业大学计算机科学与技术学院

实训总结报告



　项目名称: 豆瓣数据分析平台

　合作培养企业：天津市融创软通科技股份有限公司

　专业方向：大数据方向

　项目负责人：何晶

　专业班级: 软件z1801

姓名：顼古月

　实训起止日期：2020.6.15-2020.7.15

目录

[1.引言 3](#_Toc45639727)

[1.1项目概述 3](#_Toc45639728)

[1.2 需求分析 4](#_Toc45639729)

[1.3 运行环境 5](#_Toc45639730)

[2 项目设计 6](#_Toc45639731)

[2.1 设计思路 6](#_Toc45639732)

[2.2 模块功能介绍 6](#_Toc45639733)

[2. 2. 1角色划分： 6](#_Toc45639734)

[2.2. 2模块划分 7](#_Toc45639735)

[2.3 模块结构图 8](#_Toc45639736)

[2.4 程序流程图 8](#_Toc45639737)

[2.5 功能设计分工 11](#_Toc45639738)

[3 详细设计 12](#_Toc45639739)

[3.1 数据库设计 12](#_Toc45639740)

[3.1.1音乐区相关内容数据库设计 12](#_Toc45639741)

[3.1.2 用户数据库设计 12](#_Toc45639742)

[3.2 Spring技术 13](#_Toc45639743)

[3.3 Scrapy技术 13](#_Toc45639744)

[4 实现方案 14](#_Toc45639745)

[4.1 豆瓣音乐区数据的爬取 14](#_Toc45639746)

[4.1.1Python爬虫架构： 14](#_Toc45639747)

[4.1.2 Scrapy抓取步骤: 14](#_Toc45639748)

[4.1.3具体实现 15](#_Toc45639749)

[4.2平台前端代码展示 26](#_Toc45639750)

[4.3 平台功能实现思路及js和后端源代码 43](#_Toc45639751)

[4.3.1登陆功能 43](#_Toc45639752)

[4.3.2注册功能 46](#_Toc45639753)

[4.3.3显示数据功能 48](#_Toc45639754)

[4.3.4搜索功能 50](#_Toc45639755)

[4.3.1 数据图形化展示功能 54](#_Toc45639756)

[4.4 运行展示 55](#_Toc45639757)

[5 系统测试 59](#_Toc45639758)

[5.1测试方法及过程 59](#_Toc45639759)

[5.2出现的问题及解决方案 60](#_Toc45639760)

# 1.引言

**1 引言**

## 1.1项目概述

1.1项目概述

**项目目的：**

数据挖掘及分析已成为我们洞察艺术作品创作、传播内在规律的有效途径。此次项目以豆瓣网数据为源头，抓取了近年来各类艺术作品的导演、演员、类型、评价数量以及评分等数据，运用社会网络分析、数据统计等方法，对各类艺术作品的相关数据间做了可视化展示，也进行了研究，可以让用户更为直观的了解到这些年来各类作品的发展受众情况。

**项目背景：**

当今互联网技术高速发展的时代，用户的集体智慧在互联网上形成了社区式的带动效应，用户购买或其他相关体验某一产品后对此做出的判断，通常用于商业反馈，形成口碑效应，用户评分及评论正是其典型应用。在这类系统中，用户在分享资源内容的同时会传播该资源内容的口碑，可以简单地表明自己的喜好， 系统会通过评分来帮助用户进行信息自动分类与查找，这类系统在国外主要有 YouTube，国内比较成熟的是豆瓣系列。

本项目选取豆瓣网作为分析对象主要基于以下两点原因：首先，豆瓣网是中国最大的艺术品分享与评论社区，截止到2014年12月底，豆瓣网的注册用户已达1亿，月度独立用户数达到2亿。豆瓣电影或其他数据的80%的用户是1、2线城市中18-35岁间的大学生及白领，而这些人也构成了当下中国电影消费的主体，因此豆瓣网的评论具备代表性，可以比较准确地反映当下中国的艺术作品舆情。其次，豆瓣网也是重要的数据开放平台。研究者可以通过豆瓣提供的开放接口或爬虫程序自动、快捷、大规模地获取相关数据。为了使分析结果更具针对性，本项目编写爬虫程序抓取了近年来在全世界各类艺术品的演员、影片类型、评价数量以及评分等数据 ，并运用社会网络分析、统计学等方法对这三方面数据进行多维度的分析。

**项目意义：**

在当今电子商务的快速发展中，用户评价已经成为许多商场非常重要的展示内容，其内容甚至超过了产品本身的描述。随着出版规模的不断扩大，如何提高出版质量，缩小与世界出版权的差距，值得深思。特别是，目前的网络技术已经发展到 Web2.0 的时代。如何构建科学，合理，客观，全面的数据影响评价体系，对促进中国大数据行业的健康发展具有深远的意义。

**参考资料：**

豆瓣作为重要的信息集散地，每天都会发布大量的评论，尤其是针对电影、电视剧、书籍、音乐、小组，均有非常有价值的讨论以及专业的评论，相关信息量较大，尤其是针对电影类内容，更是拥有大量的相关信息，本系统通过爬虫机制针对豆瓣上的电影频道（书籍、电视剧、音乐、小组）进行数据收集并分析，同时，对于电影库要求对比猫眼中的评分和评论，生成对应的分析图表，对比两个数据来源中的相同数据（通过电影名称、年份）获取交叉对比数据，生成的图表能够非常直观的展示某一电影在双方影迷心中的地位。对于豆瓣上的小组类别，进行热评搜索和分析，生成热评类别的不同和可视化分析。同时，该系统还需要一个必要的管理系统，用来控制爬虫指向，保存个性数据，展示分析结果图表等内容。

## 1.2 需求分析

1.2 需求分析

豆瓣网已成为国内信息全、用户数量大且非常活跃的专业读书网站，专注于为用户提供全面且精细化服务。通过获取、分析豆瓣各类艺术作品评论数据，可以对用户的选择提供借鉴意义。采用 Python 可以快速获取、分析大量的豆瓣书评数据，得出可靠、准确的评论结果。

大众中有很多电影图书音乐等的爱好者，在网站的产品设计上，是以满足用户对各类作品的交流分享需求出发的。考虑到需要用户产生内容的网站开始最困难，需要想办法找些基础的内容让用户来参与，然后产生滚雪球效应。为此我们想到的办法是提供评分系统。这个系统实际上是一个搜索引擎，用户可以利用这个搜索引擎，找到一个作品对应的评分或其他数据。而豆瓣的数据服务、社区交流等，是一个不是以买卖关系为目的，而是以分享作品体验和结识朋友为目的的社区，这为我们此次项目提供良好的资源来源。

本文对豆瓣网相关数据，通过应用 python 中的 Scrapy 框架提取， 然后存储在 MYSQL 数据中，Python 与 MYSQL 通过 Navicat forMYSQL 进行搭建， 然后对抓取的数据进行清洗，最后利用 IntelliJ IDEA 进行数据可视化。然后进一步分析粉丝所喜欢的作品的国家分布、时长、年份，和粉丝对高分作品的评分区间。通过多角度对豆瓣的数据分布情况。给出粉丝选择的建议。进一步我们可以分析 豆瓣作品的类型，和对应的评分数量，分析豆瓣排名、评分和评分数目的相互作用关系。为粉丝提供更多的选择方案。同样，可以为作为制作行业的相关参考，也可以作为作品推送的指导。

## 1.3 运行环境

1.3 运行环境

**软件工具:**

数据来源：豆瓣网（电影、电视剧、音乐、图书）

爬虫技术：Python Scrapy框架

服务器端开发框架：SSM（Spring + SpringMVC + MyBatis）

数据库技术：MySQL

前端开发技术：jQuery + LayUI + eCharts

平台开发模式：基于JSON数据的前后端分离式开发

数据库管理工具：Navicat

开发工具：Idea Pycharm

系统：Windows 10

**硬件工具：**

处理器：Intel(R) Core(TM) i5-8250U cpu@ 1.6.0GHz 1.80GHz

内存:8.00GB

系统类型：64位操作系统，基于x64的处理器

# 2 项目设计

**2 项目设计**

## 2.1 设计思路

2.1 设计思路

数据获取:使用scrapy爬虫框架从豆瓣的音乐区将相关数据爬取下来，写入到Navicat for Mysql数据库中

平台设计: 使用jQuery + LayUI框架并利用相关html css JavaScript技术搭建平台前端页面，运用springboot 框架实现登陆注册功能

数据分析: 在搭建好的平台进行数据分析功能的完善，将数据库表中的数据显示在表格中，实现搜索功能(详见下面的详细设计)，运用echarts实现数据的图形化显示。

## 2.2 模块功能介绍

2.2 模块功能介绍

**2. 2. 1角色划分：**

### 2. 2. 1角色划分：

* **平台管理者：**

负责对平台参数进行设置，查看统计信息，管理后台数据字典，管理平台用户等功能；

* **外部用户：**

**共通部分：**自行注册，管理自身信息；

**电影库：**提交影片检索任务，提交评论分析任务，管理个人影库，对个人影库进行分析并图例化

**个人书库：**提交图书检索任务，提交评论分析任务，管理个人书库，对个人图书库进行分析并图例化

**个人电视剧库：**提交电视剧检索任务，提交评论分析任务，管理个人电视剧库，对个人电视剧库进行分析并图例化

**个人音乐库：**提交音乐歌曲检索任务，提交评论分析任务，管理个人音乐库，对个人音乐库进行分析并图例化

**个人小组库：**提交小组内容检索任务，提交评论分析任务，管理个人小组库，对个人小组库进行分析并图例化

### 2.2. 2模块划分

**2.2. 2模块划分**

**模块一：系统基础管理模块**

1) 系统用户管理（包括平台管理人员的建立维护以及平台外部注册用户的基础信息管理等功能）。

2) 系统数据字典管理及维护（管理系统部分下拉列表内容项目）。

3) 平台主界面设计及开发。

4) 管理首页Dashboard设计与开发，图形化当前系统情况，包括已注册用户数量、已贮存影片（书籍、电视剧、音乐）数量、已分析的评论数、各分类的检索前十名汇总等信息；

5) 权限管理功能，不同类别用户进入系统后，拥有不同的权限以及功能树

**模块二：平台外部用户个人信息管理**

1) 平台允许通过相关页面进行个人用户的注册。

2) 能够对个人信息进行更新。

3) 允许重置个人密码。

4）个人登陆后的信息汇总Dashboard，包括个人已储存的影片（书籍、电视剧、音乐）数量、已分析的评论数等信息

**模块三：指定分类数据单页面抓取**

1) 允许平台用户选取不同的类别后填写需要抓取的关键字，系统会根据关键字进行平台（豆瓣）检索，并将检索后的结果反馈给当前用户。

2) 将相关详情进行入库操作并建立与当前用户的关联，形成该用户的在该分类下的检索历史。

3) 检索历史管理功能，可分类查看检索历史，并删除相关检索历史条目，同时删除该条目对应的一切信息（评论、分析汇总等）

4) 允许平台用户在本平台为抓取的项目评分，编写评论内容，用于生成该平台影片库的评论内容。

**模块四：多平台内容抓取及对比功能（仅针对影片、图书模块）**

1）针对影片库，需要同时抓取 豆瓣 以及 猫眼（或其他第三方平台，例如IMDB.cn） 上的相关信息，生成交叉对比数据，例如，豆瓣评分 ： 猫眼评分，豆瓣评分人数 ： 猫眼评分人数，；

2）针对图书库，需要同时抓取 豆瓣 以及 京东 上的相关信息，生成交叉对比数据；

**模块五：评论收集分析功能**

1）针对指定电影、图书、电视剧、音乐进行 评论抓取；

2）将抓取内容进行分析，例如，针对用户性别对该影片的评分进行分类汇总，查看是男性给予更高的分数，还是女性给予更高的分数，或者根据日期，汇总评论数量，检查评论在一定时间段内的分布趋势，或者汇总不同分数的评论数，形成饼状数据图进行展示；

**个人主要负责部分：**

**Welcome页面**作为主页面，能够直接进入登陆或注册页面

**登陆功能**能够通过读取数据库中的用户来进行页面跳转，并进行报错提示，通过点击链接能跳转至注册界面。

**注册界面**能够在数据库中添加新用户，通过点击链接跳转至登陆页面，登陆界面下面的微信qq微博登陆功能未实现，点击会跳转至别的页面

**音乐区数据分析模块**具有分页显示全部数据，按照艺术家搜索，支持模糊查询，选择流派，选择发行时间这些功能。

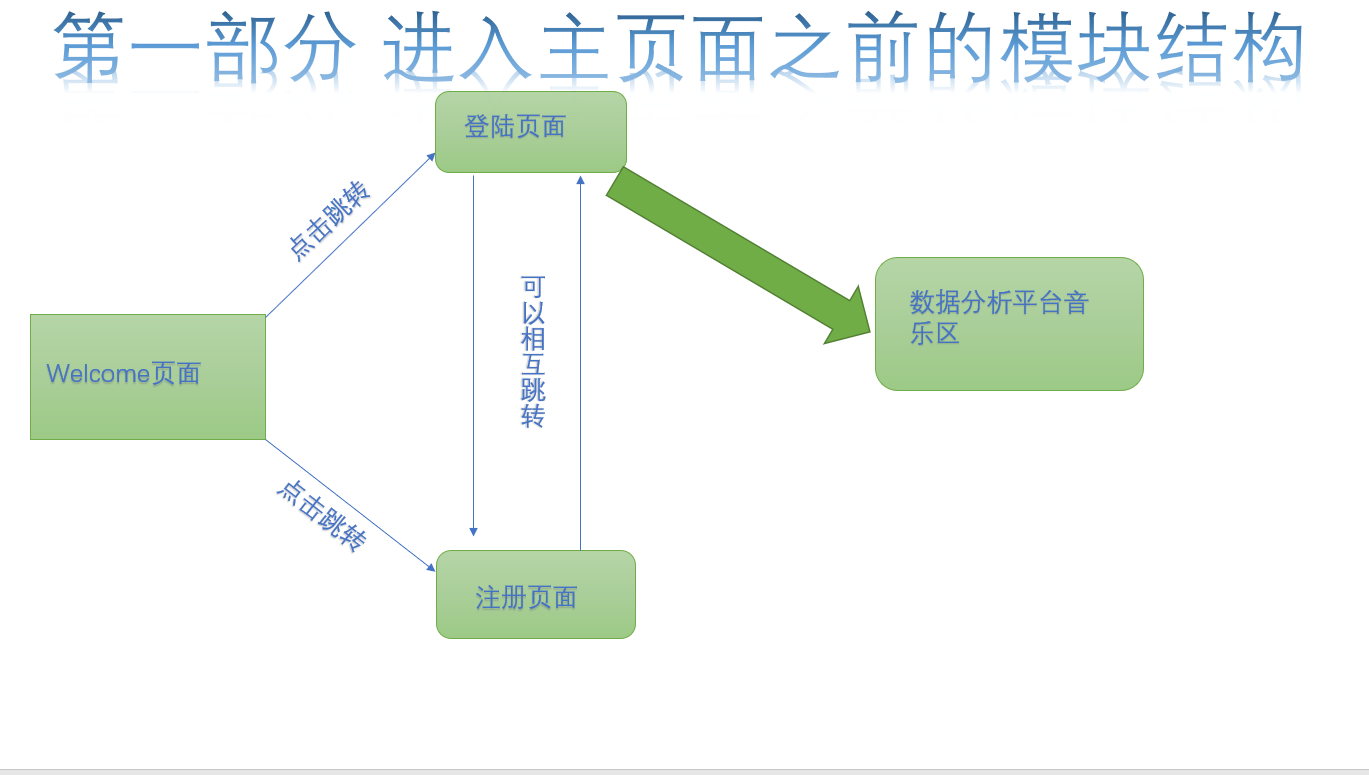
**视图显示区域有三个模块：**

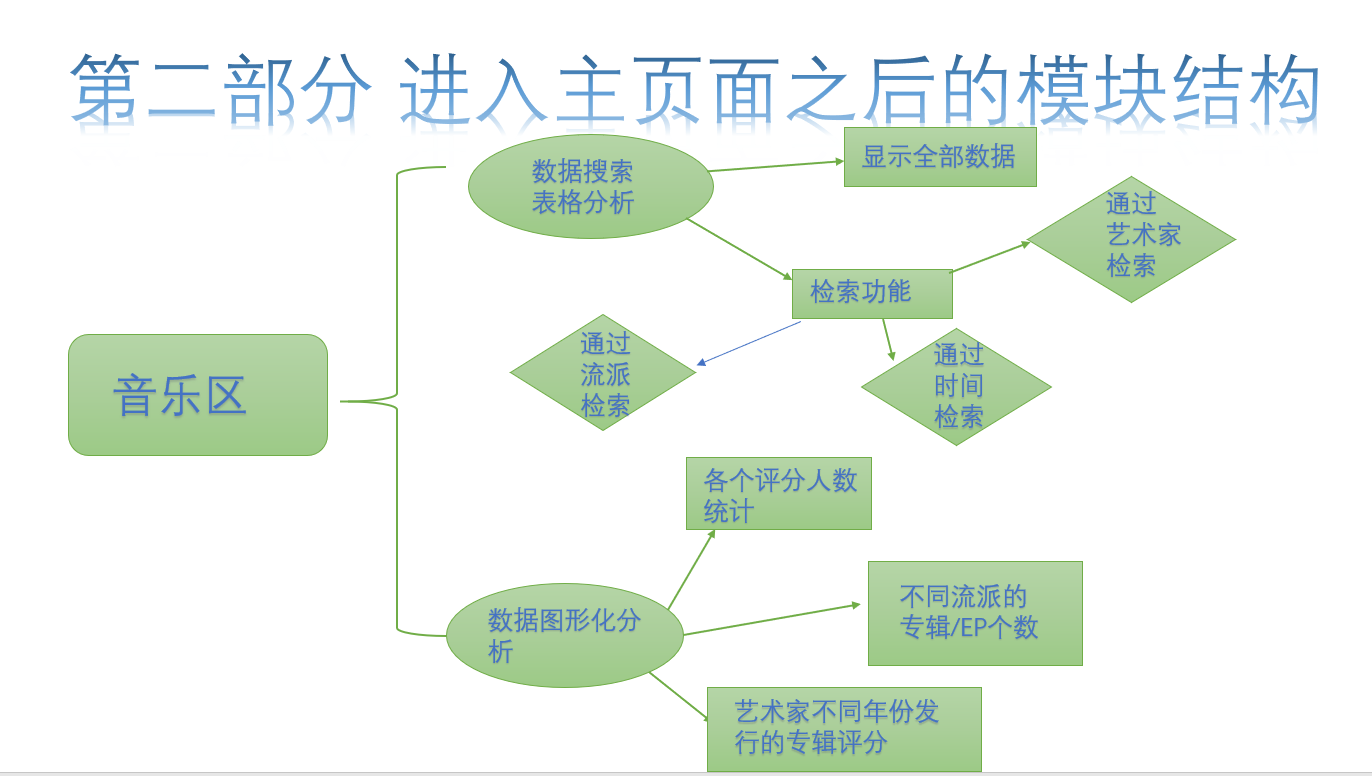
1. 各个评分的数量，通过条形图饼图漏斗图三种图展现出来
2. 不同流派的专辑/EP的数量，通过条形图饼图漏斗图三中图展现出来
3. 通过搜索实现对特定歌手发行专辑的年份及评分的折线图数据分析

**2.3 模块结构图**

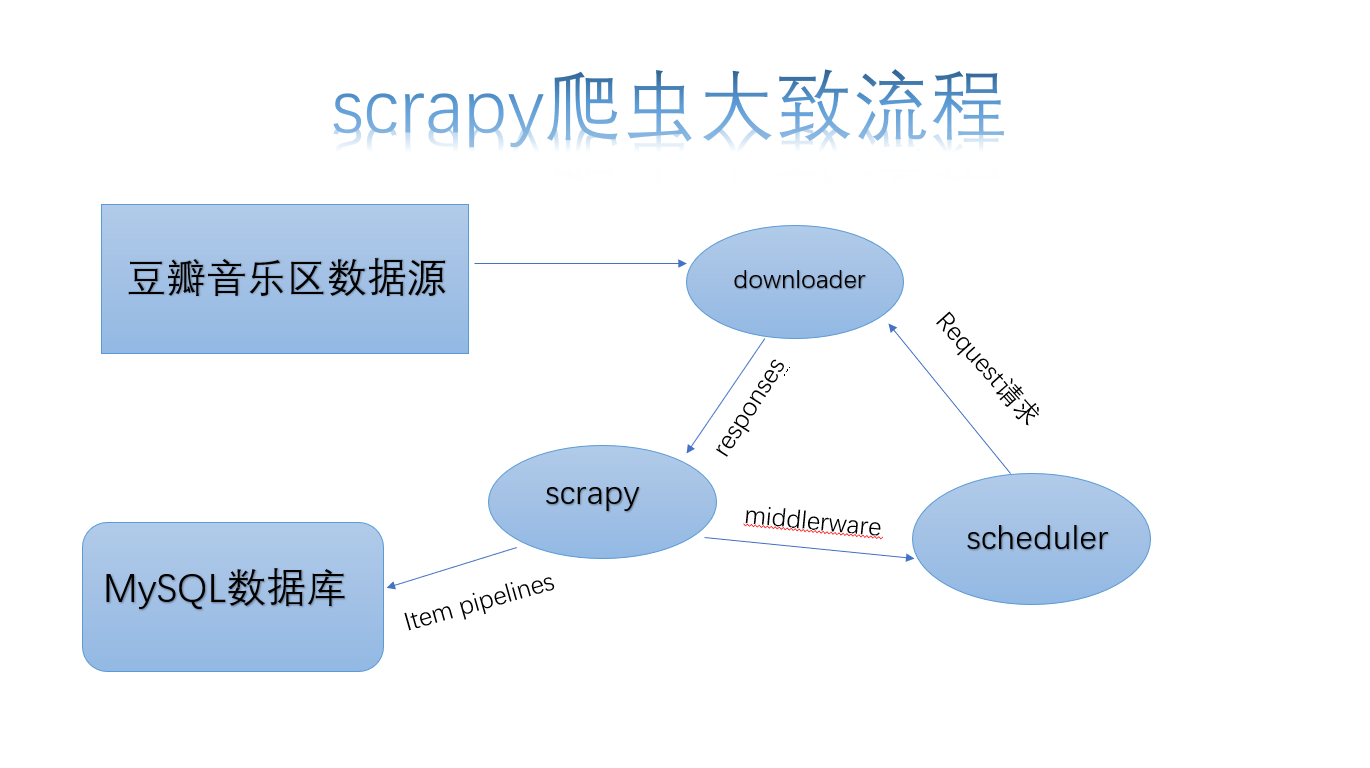
## 2.3 模块结构图

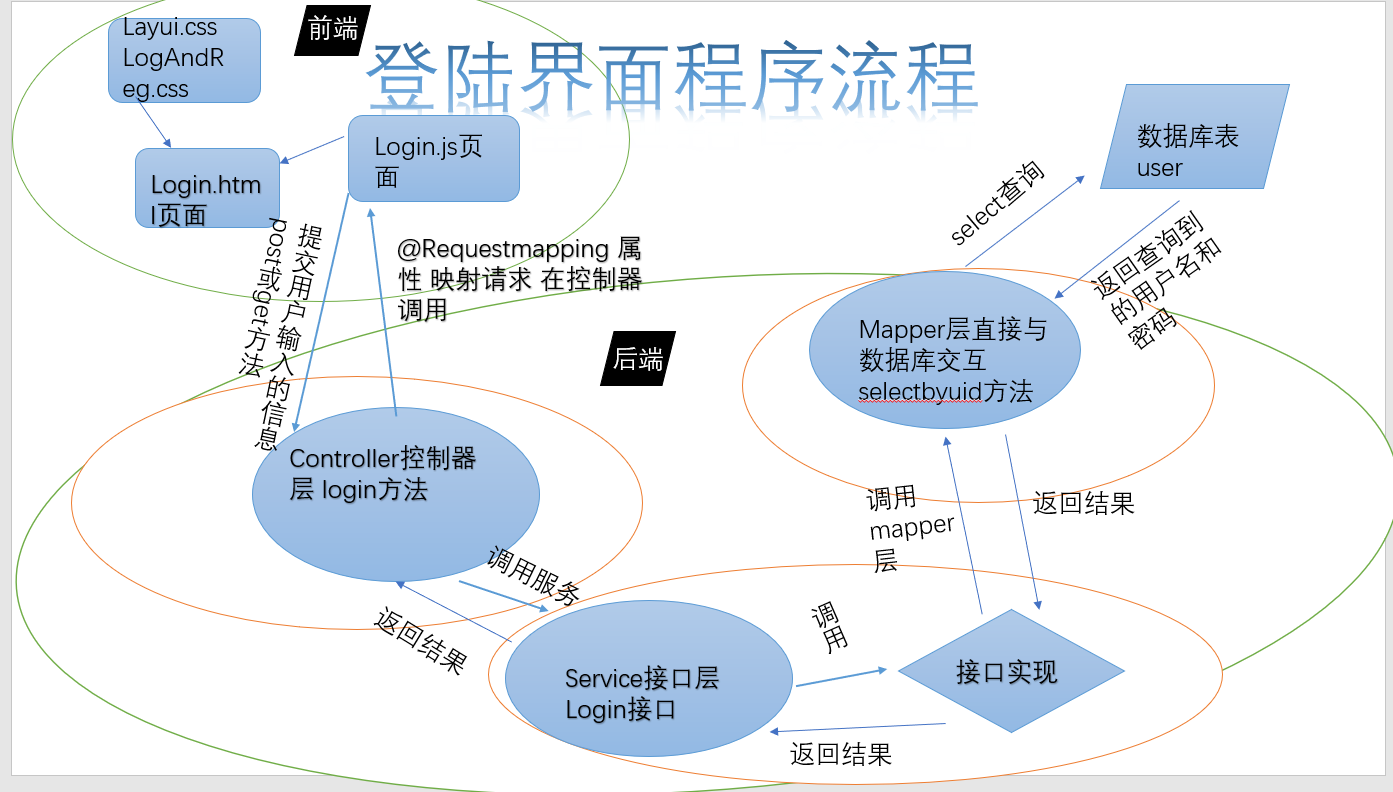
## 2.4 程序流程图

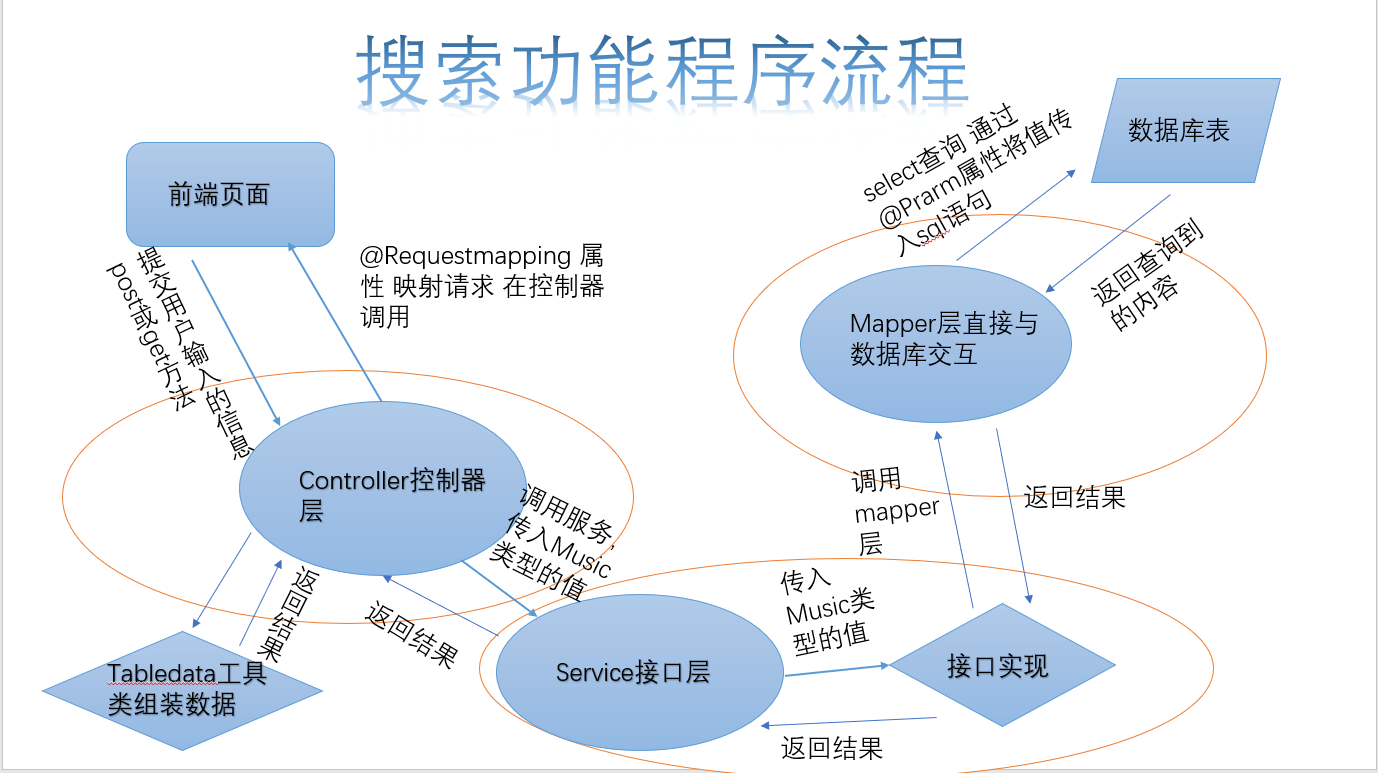


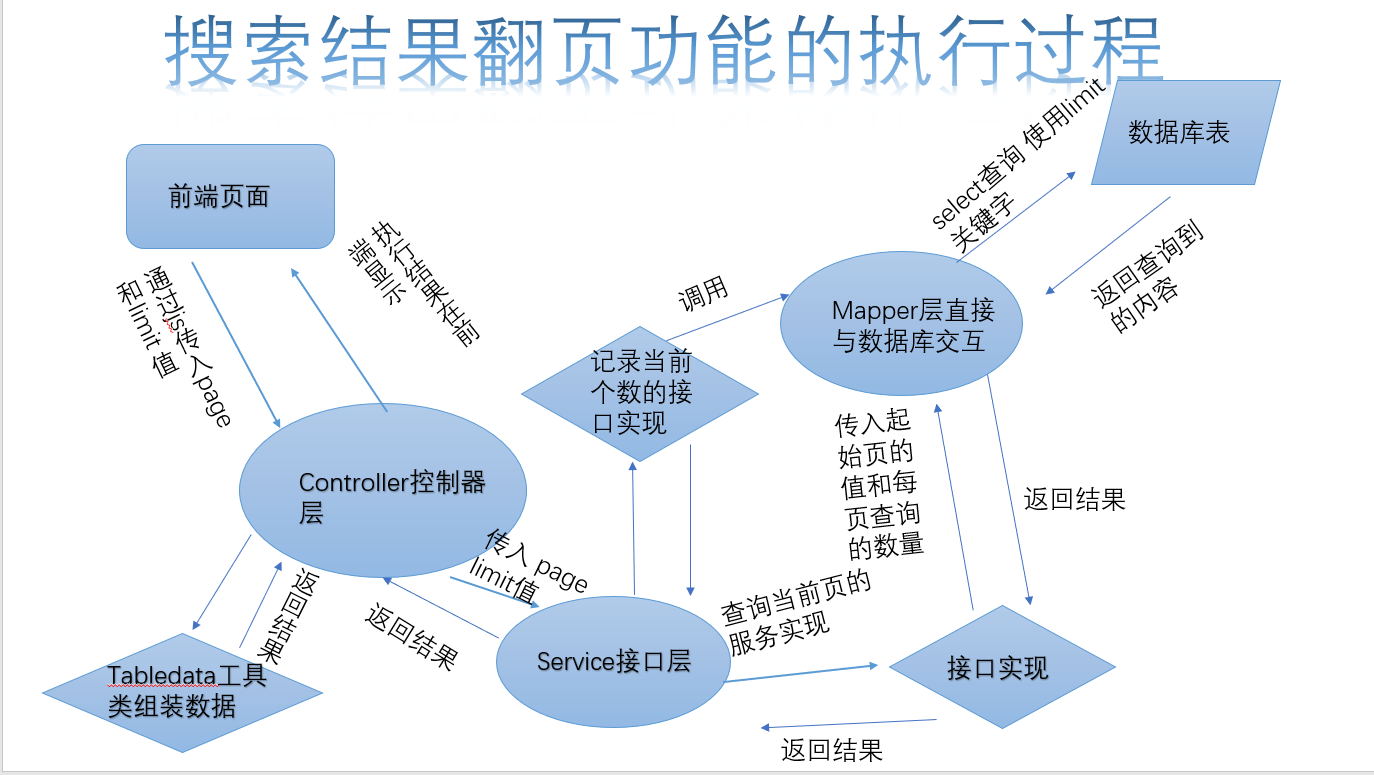


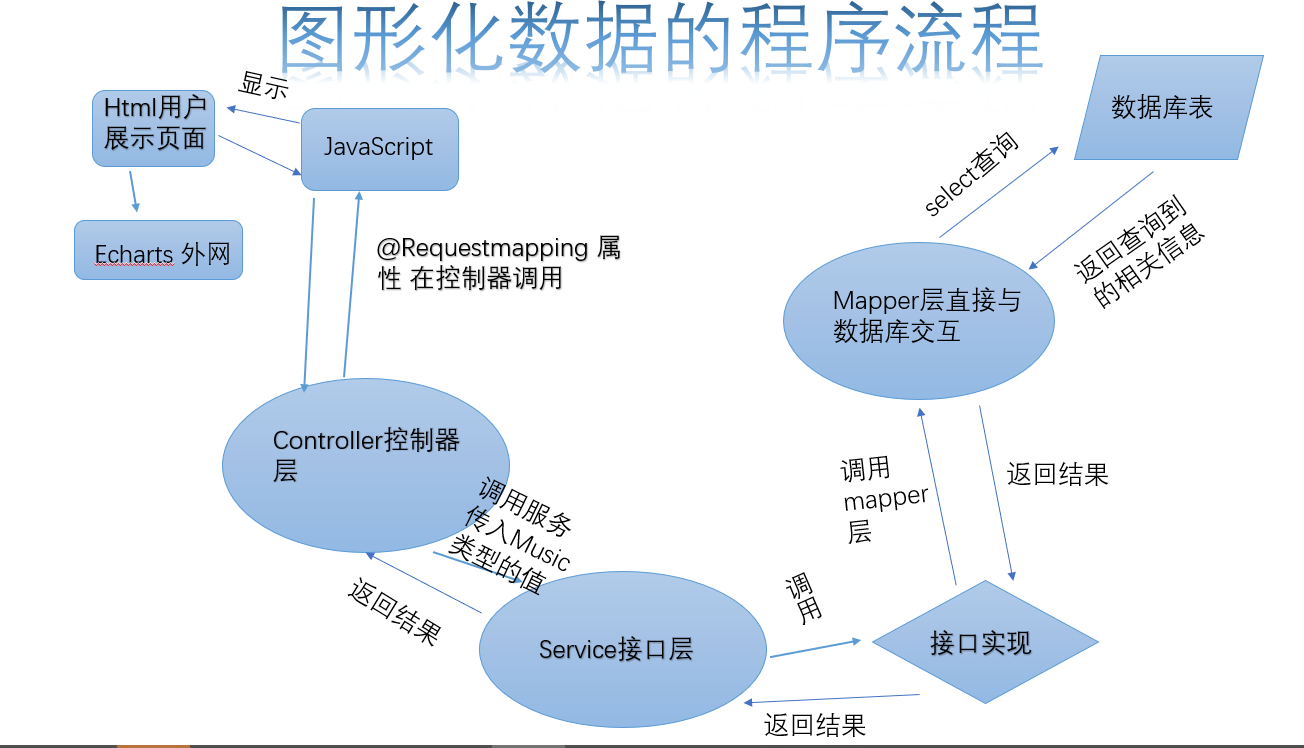
2.4 程序流程图











**2.5 功能设计分工**

## 2.5 功能设计分工

**小组分工：**

顼古月：音乐管理界面及相关数据可视化

杨昊霖：电影管理界面及相关数据可视化

刘欣妍：电视剧管理界面及相关数据可视化

杨晓庆：图书管理界面及相关数据可视化

许光皓：小组管理界面及相关数据可视化

**个人主要负责部分：**

1、豆瓣音乐区数据的爬取及写入数据库

2、登陆注册部分功能的实现及页面优化 welcome主页面的设计实现

3、音乐库中数据的显示，搜索功能的实现，关键字搜索功能

4、图形化显示中布局设计，通过三个不同角度进行图表化分析的具体实现，通过搜索关键字将结果显示到图表

# 3 详细设计

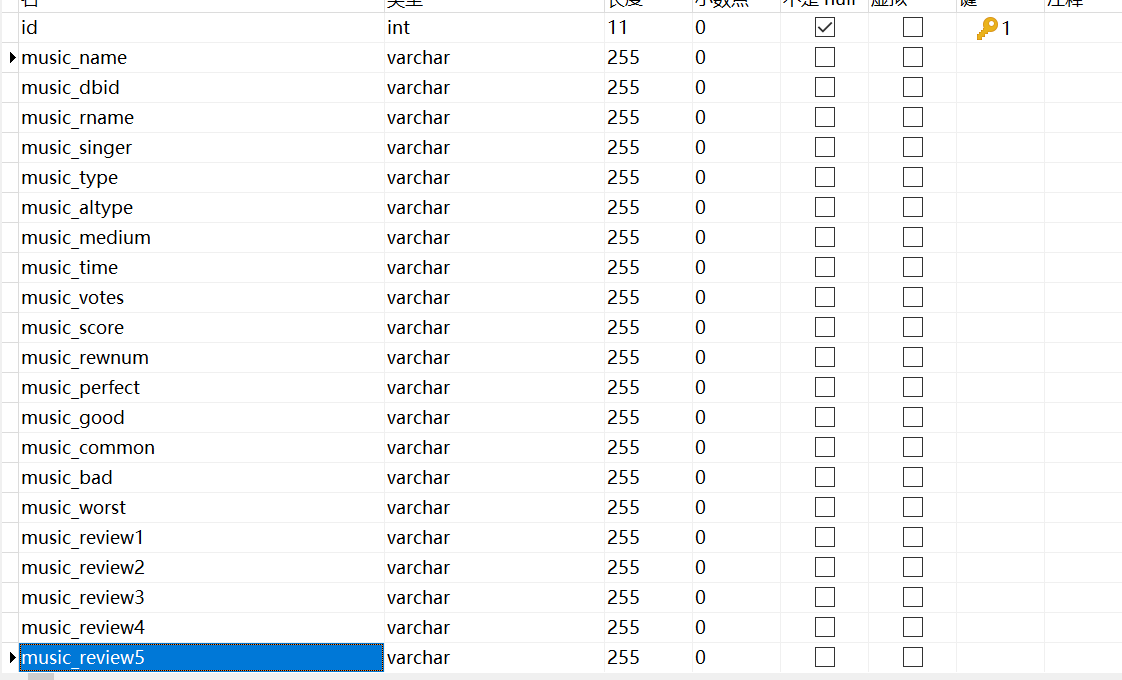
**3 详细设计**

## 3.1 数据库设计

**3.1 数据库设计**

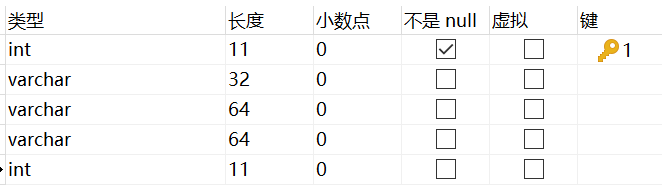
### 3.1.1音乐区相关内容数据库设计

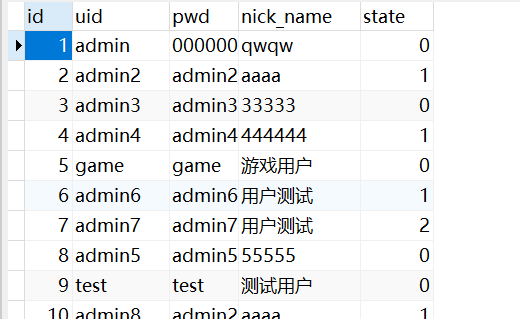
**3.1.1音乐区相关内容数据库设计**



### 3.1.2 用户数据库设计

**3.1.2 用户数据库设计**





## 3.2 Spring技术

**3.2 Spring技术**

前端界面提交用户输入的信息post或get方法到Controller控制器层，Controller控制器调用服务，传入Movie类型的值到Service接口层，Service接口层传入Movie类型的值进行接口实现，调用mapper层，Mapper层直接与数据库交互，select查询 通过@Prarm属性将值传入sql语句到数据库表，数据库将查询到的内容返回到Mapper层，一直返回到Controller控制器在前端@Requestmapping属性映射请求在前端调用

## 3.3 Scrapy技术

**3.3 Scrapy技术**

1.引擎打开一个网站(open a domain)，找到处理该网站的Spider并向该spider请求第一个要爬取的URL(s)。

2.引擎从Spider中获取到第一个要爬取的URL并在调度器(Scheduler)以Request调度。

3.引擎向调度器请求下一个要爬取的URL。

4.调度器返回下一个要爬取的URL给引擎，引擎将URL通过下载中间件(请求(request)方向)转发给下载器(Downloader)。

5.一旦页面下载完毕，下载器生成一个该页面的Response，并将其通过下载中间件(返回(response)方向)发送给引擎。

6.引擎从下载器中接收到Response并通过Spider中间件(输入方向)发送给Spider处理。

7.Spider处理Response并返回爬取到的Item及(跟进的)新的Request给引擎。

8.引擎将(Spider返回的)爬取到的Item给Item Pipeline，将(Spider返回的)Request给调度器。

9.(从第二步)重复直到调度器中没有更多地request，引擎关闭该网站。

# 4 实现方案

**4 实现方案**

## 4.1 豆瓣音乐区数据的爬取

**4.1 豆瓣音乐区数据的爬取**

运用了Pycharm开发工具，采用Python中的Scrapy框架实现了对豆瓣电视剧详情页及对应电视剧其评论相关数据的爬虫。

### 4.1.1Python爬虫架构：

**4.1.1Python爬虫架构：**

调度端：用来启动、停止、和监视爬虫

URL管理：对等待爬取和已经爬取的URL进行管理，简单来说就是为后续模块提供可供爬取的URL

网页下载器：将供爬取的URL的网页下载下来，组成供解析的字符串

网页解析器：将字符串解析

### 4.1.2 Scrapy抓取步骤:

**4.1.2 Scrapy抓取步骤:**

新建项目 (Project) ：新建一个新的爬虫项目

明确目标（Items）：明确你想要抓取的目标

制作爬虫（Spider）：制作爬虫开始爬取网页

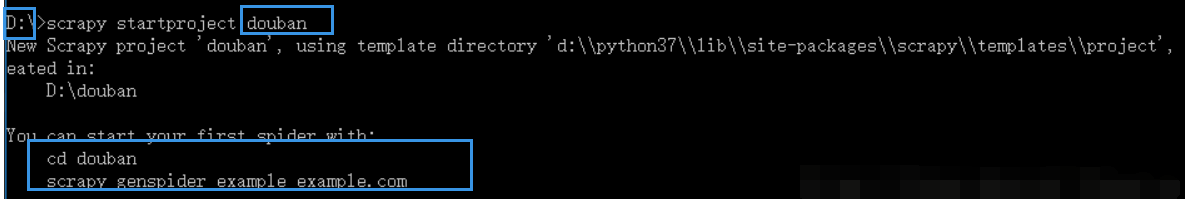
存储内容（Pipeline）：设计管道存储爬取内容

### 4.1.3具体实现

**4.1.3具体实现**

**①新建项目**

打开DOM命令窗口，进入桌面上的tjpu01文件，输入scrapy startproject guyuemusic，新建名称为 “guyuemusic” 的爬虫项目

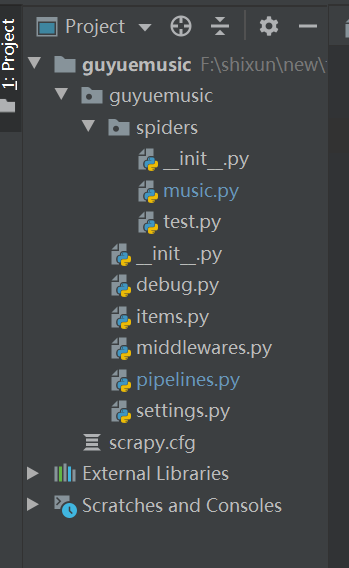


上图因为文件很早前建立只能借用一下其他文件信息，将前两个框内信息替换，再按照第三个框内提示内进行下面的操作进入新建的文件夹并连接到我们要爬取地址就可以了，我主要负责的网页地址是music.douban.com，具体如下：

输入命令：cd guyuemusic

输入命令：scrapy genspider music music.douban.com，新建名称为“guyuemusic\_spider”的文件，其中“music.douban.com”为域名（douban\_spider文件名称可以另取,该文件用于编写XPath和正则表达式）

**②开启PyCharm查看新建的爬虫项目**



**③明确目标（明确需要抓取的网页信息）**

打开items.py文件,在类 DoubanItem下定义抓取目标,代码如下：

class DetailItem(scrapy.Item):

music\_url = scrapy.Field() #音乐网址  
 music\_name = scrapy.Field() #名字  
 music\_dbid = scrapy.Field()  
 music\_rname = scrapy.Field() #又名  
 music\_singer = scrapy.Field()  
 music\_type = scrapy.Field() #流派  
 music\_altype = scrapy.Field() #专辑类型  
 music\_medium = scrapy.Field() #介质  
 music\_time = scrapy.Field() #发行时间  
 music\_votes = scrapy.Field() #出版者  
 music\_score = scrapy.Field() #评分  
 music\_rewnum = scrapy.Field() #评价人数  
 music\_perfect = scrapy.Field() #五星占比  
 music\_good = scrapy.Field() #四星占比  
 music\_common = scrapy.Field() #三星占比  
 music\_bad = scrapy.Field() #二星占比  
 music\_worst = scrapy.Field() #一星占比  
 music\_review1 = scrapy.Field() #五个短评论  
 music\_review2 = scrapy.Field()  
 music\_review3 = scrapy.Field()  
 music\_review4 = scrapy.Field()  
 music\_review5 = scrapy.Field()

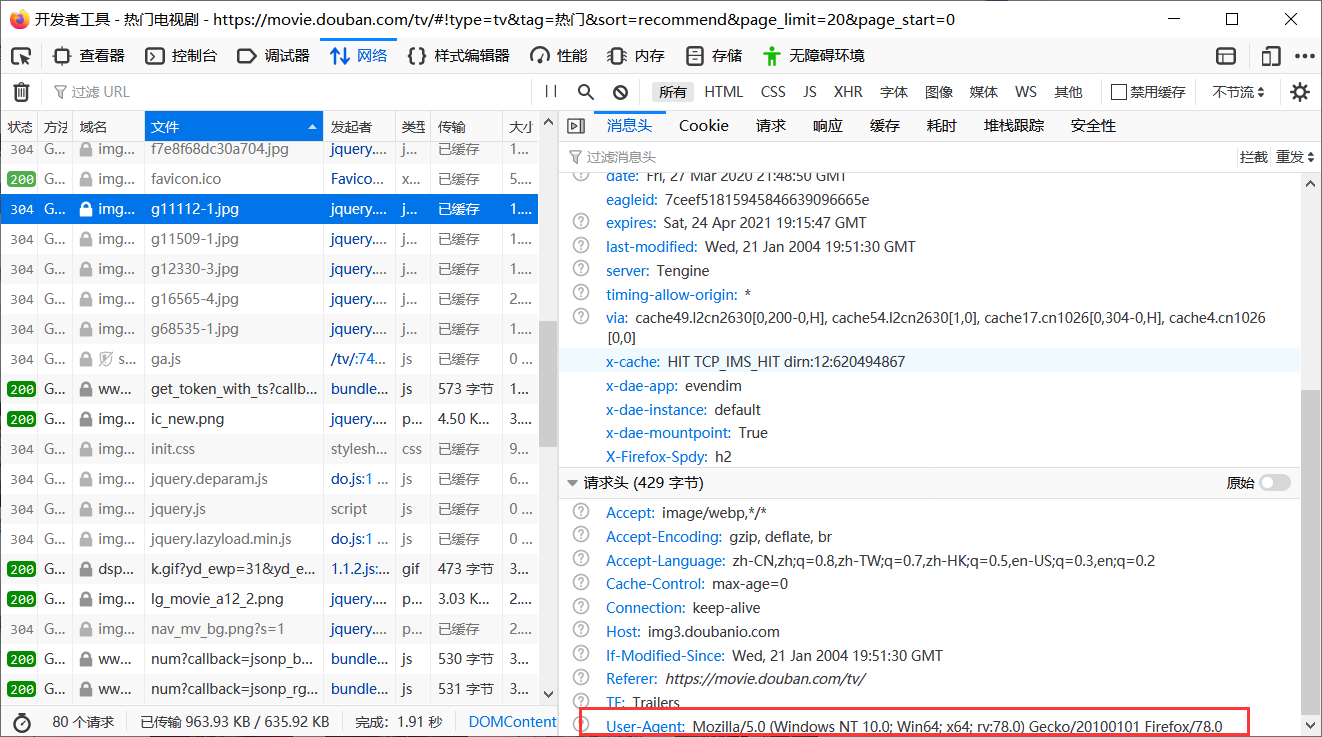
**④修改setting的USER\_AGENT值**

1.进入豆瓣网页

（https://music/movie.com）

2.按F12，点击 Network ,再按F5刷新网页

3.点击网络里面任意一项的消息头找到 User-Agent 项，复制其内容



**⑤打开 settings 文件**

加上数据库配置：

BOT\_NAME = 'guyuemusic'

SPIDER\_MODULES = ['guyuemusic.spiders']  
NEWSPIDER\_MODULE = 'guyuemusic.spiders'  
MYSQL\_HOST = 'www.91iedu.com'  
MYSQL\_PORT = 3391  
MYSQL\_DBNAME = 'team01'  
MYSQL\_USER = 'team01'  
MYSQL\_PASSWD = 'team01'

去掉USER\_AGENT这行的注释，清空原内容，粘贴上述从浏览器获取的内容，即加上我们的USER\_AGENT信息

USER\_AGENT = 'Mozilla/5.0(Windows nt 10.0;Win64; x64; rv:67.0)

Gecko/20100101 Firefox/67.0'

将一些配置加上，避免豆瓣网拒绝我们的爬虫

ROBOTSTXT\_OBEY = False

CONCURRENT\_REQUESTS = 2

DEPTH\_LIMIT = 3

DOWNLOAD\_DELAY = 4

DEFAULT\_REQUEST\_HEADERS = {  
 'Accept': 'text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8',  
 'Accept-Language': 'en',  
}

DOWNLOADER\_MIDDLEWARES = {  
 'guyuemusic.middlewares.GuyuemusicDownloaderMiddleware': 543,  
 'guyuemusic.middlewares.RotateUserAgentMiddleware':542,  
 'scrapy.downloadermiddlewares.useragent.UserAgentMiddleware':None  
}

设置并打开pipelines，为接下来的将数据插进数据库做准备

ITEM\_PIPELINES = {

'guyuemusic.pipelines.GuyuemusicPipeline': 300,  
}

**⑥制作爬虫（music\_spider文件编写），代码如下：**

class MusicSpider(CrawlSpider):

name = 'music'  
 allowed\_domains = ['music.douban.com']  
 start\_urls = ['https://music.douban.com/tag/华语']  
  
 rules = (Rule(LinkExtractor(allow=r"\?start=\d+\&type=T$"), follow=True, callback='parse\_list'),  
 Rule(LinkExtractor(allow=r"/subject/\d+/$"), callback='parse\_detail', follow=False),)  
 def parse\_detail(self, response):  
  
 item = DetailItem()  
 #专辑名字和url  
 item['music\_name'] = response.xpath('//\*[@id="wrapper"]/h1/span/text()').extract()[0]  
 item['music\_url'] = response.url  
 item['music\_dbid'] = item['music\_url'].split('/')[-2]  
 content = "".join(response.xpath('//\*[@id="info"]').extract())  
 info = response.xpath('//\*[@id="info"]/span').extract()  
 # print(info)  
 rating\_wrap = response.xpath('//\*[@class="rating\_wrap clearbox"]/span').extract()  
 # print(rating\_wrap)  
  
 #音乐基本信息的爬取  
 item['music\_rname'] = ""  
 item['music\_singer'] = ""  
 item['music\_type'] = ""  
 item['music\_altype'] = ""  
 item['music\_medium'] = ""  
 item['music\_time'] = ""  
 item['music\_votes'] = ""  
 for i in range(0, len(info)):  
 if "又名:" in info[i]:  
 music\_rname = re.findall(r"<span class=\"pl\">又名:</span>(.\*?)<br>", content, re.S)  
 item['music\_rname'] = "".join(music\_rname).replace("\xa0", "").replace("\n", "").replace(" ", "")  
  
  
 if "表演者" in info[i]:  
 if i == 0:  
 item['music\_singer'] = "|".join(response.xpath('//\*[@id="info"]/span[1]/span/a/text()').extract())  
  
 elif i == 1:  
 item['music\_singer'] = "|".join(response.xpath('//\*[@id="info"]/span[2]/span/a/text()').extract())  
  
 else:  
 item['music\_singer'] = ""  
  
 if "流派:" in info[i]:  
 music\_type = re.findall(r"<span class=\"pl\">流派:</span>(.\*?)<br>", content, re.S)  
 item['music\_type'] = "".join(music\_type).replace("\xa0", "").replace("\n", "").replace(" ", "")  
  
 if "专辑类型" in info[i]:  
 music\_altype = re.findall(r"<span class=\"pl\">专辑类型:</span>(.\*?)<br>", content, re.S)  
 item['music\_altype'] = "".join(music\_altype).replace("\xa0", "").replace("\n", "").replace(" ", "")  
  
 if "介质" in info[i]:  
 music\_medium = re.findall(r"<span class=\"pl\">介质:</span>(.\*?)<br>", content, re.S)  
 item['music\_medium'] = "".join(music\_medium).replace("\xa0", "").replace("\n", "").replace(" ", "")  
  
 if "发行时间" in info[i]:  
 music\_time = re.findall(r"<span class=\"pl\">发行时间:</span>(.\*?)<br>", content, re.S)  
 item['music\_time'] = "".join(music\_time).replace("\xa0", "").replace("\n", "").replace(" ", "")  
  
 if "出版者" in info[i]:  
 music\_votes = re.findall(r"<span class=\"pl\">出版者:</span>(.\*?)<br>", content, re.S)  
 item['music\_votes'] = "".join(music\_votes).replace("\xa0", "").replace("\n", "").replace(" ", "")  
  
 #评分及人数 占比的爬取  
 item['music\_score'] = response.xpath('//\*[@class="rating\_self clearfix"]/strong/text()').extract()[0]  
 item['music\_rewnum'] = response.xpath(  
 '//\*[@class="rating\_self clearfix"]/div/div[@class="rating\_sum"]/a/span/text()').extract()[0]  
  
 for i in range(0, len(rating\_wrap)):  
 if i == 1:  
 item['music\_perfect'] = response.xpath('//\*[@class="rating\_wrap clearbox"]/span[2]/text()').extract()[0]  
  
 if i == 3:  
 item['music\_good'] = response.xpath('//\*[@class="rating\_wrap clearbox"]/span[4]/text()').extract()[0]  
  
 if i == 5:  
 item['music\_common'] = response.xpath('//\*[@class="rating\_wrap clearbox"]/span[6]/text()').extract()[0]  
  
 if i == 7:  
 item['music\_bad'] = response.xpath('//\*[@class="rating\_wrap clearbox"]/span[8]/text()').extract()[0]  
  
 if i == 9:  
 item['music\_worst'] = response.xpath('//\*[@class="rating\_wrap clearbox"]/span[10]/text()').extract()[0]  
  
  
 #热门短评五条  
 item['music\_review1'] = \  
 response.xpath('//\*[@class="comment-list hot show"]/ul/li[1]/div/p/span[1]/text()').extract()[0]  
 item['music\_review2'] = \  
 response.xpath('//\*[@class="comment-list hot show"]/ul/li[2]/div/p/span[1]/text()').extract()[0]  
 item['music\_review3'] = \  
 response.xpath('//\*[@class="comment-list hot show"]/ul/li[3]/div/p/span[1]/text()').extract()[0]  
 item['music\_review4'] = \  
 response.xpath('//\*[@class="comment-list hot show"]/ul/li[4]/div/p/span[1]/text()').extract()[0]  
 item['music\_review5'] = \  
 response.xpath('//\*[@class="comment-list hot show"]/ul/li[5]/div/p/span[1]/text()').extract()[0]  
  
 yield item

**⑦通过pipelines插入到数据库当中：**

import pymysql  
from guyuemusic import settings

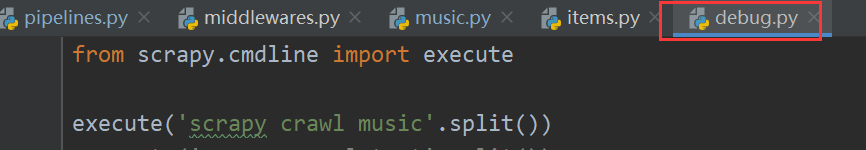
from guyuemusic.items import DetailIte  
  
class GuyuemusicPipeline(object):  
  
 def \_\_init\_\_(self):  
 self.connect = pymysql.connect(  
 host=settings.MYSQL\_HOST,  
 port=settings.MYSQL\_PORT,  
 db=settings.MYSQL\_DBNAME,  
 user=settings.MYSQL\_USER,  
 passwd=settings.MYSQL\_PASSWD,  
 charset='utf8mb4',  
 use\_unicode=True,  
 cursorclass=pymysql.cursors.DictCursor  
 )  
 self.cursor = self.connect.cursor()  
  
 def process\_item(self, item, spider):  
 # if isinstance(item, DetailItem):  
 # self.process\_music\_pop(item)  
 self.cursor.execute('''select \* from guyuemusictable where music\_dbid = %s''',  
 pymysql.escape\_string(item['music\_dbid'], ))  
 music = self.cursor.fetchone()  
 if music == None:  
 print("new")  
 self.cursor.execute(  
 '''insert into guyuemusictable(music\_name, music\_dbid, music\_rname, music\_singer, music\_type,   
 music\_altype, music\_medium, music\_time, music\_votes, music\_score, music\_rewnum,   
 music\_perfect, music\_good, music\_common, music\_bad, music\_worst, music\_review1,  
 music\_review2, music\_review3, music\_review4, music\_review5)  
 value (%s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s)''',  
 (pymysql.escape\_string(item["music\_name"]),  
 pymysql.escape\_string(item["music\_dbid"]),  
 pymysql.escape\_string(item["music\_rname"]),  
 pymysql.escape\_string(item["music\_singer"]),  
 pymysql.escape\_string(item["music\_type"]),  
 pymysql.escape\_string(item["music\_altype"]),  
 pymysql.escape\_string(item["music\_medium"]),  
 pymysql.escape\_string(item["music\_time"]),  
 pymysql.escape\_string(item["music\_votes"]),  
 pymysql.escape\_string(item["music\_score"]),  
 pymysql.escape\_string(item["music\_rewnum"]),  
 pymysql.escape\_string(item["music\_perfect"]),  
 pymysql.escape\_string(item["music\_good"]),  
 pymysql.escape\_string(item["music\_common"]),  
 pymysql.escape\_string(item["music\_bad"]),  
 pymysql.escape\_string(item["music\_worst"]),  
 pymysql.escape\_string(item["music\_review1"]),  
 pymysql.escape\_string(item["music\_review2"]),  
 pymysql.escape\_string(item["music\_review3"]),  
 pymysql.escape\_string(item["music\_review4"]),  
 pymysql.escape\_string(item["music\_review5"])))  
 else:  
 print("update")  
 self.cursor.execute('''  
 update guyuemusictable  
 set music\_name = %s,  
 music\_dbid = %s,  
 music\_rname = %s,  
 music\_singer = %s,  
 music\_type = %s,  
 music\_altype = %s,  
 music\_medium = %s,  
 music\_time = %s,  
 music\_votes = %s,  
 music\_score = %s,  
 music\_rewnum = %s,  
 music\_perfect = %s,  
 music\_good = %s,  
 music\_common = %s,  
 music\_bad = %s,  
 music\_worst = %s,  
 music\_review1 = %s,  
 music\_review2 = %s,  
 music\_review3 = %s,  
 music\_review4 = %s,  
 music\_review5 = %s  
 where id = %s''',  
  
 (pymysql.escape\_string(item['music\_name']),  
 pymysql.escape\_string(item['music\_dbid']),  
 pymysql.escape\_string(item['music\_rname']),  
 pymysql.escape\_string(item['music\_singer']),  
 pymysql.escape\_string(item['music\_type']),  
 pymysql.escape\_string(item['music\_altype']),  
 pymysql.escape\_string(item['music\_medium']),  
 pymysql.escape\_string(item['music\_time']),  
 pymysql.escape\_string(item['music\_votes']),  
 pymysql.escape\_string(item['music\_score']),  
 pymysql.escape\_string(item['music\_rewnum']),  
 pymysql.escape\_string(item['music\_perfect']),  
 pymysql.escape\_string(item['music\_good']),  
 pymysql.escape\_string(item['music\_common']),  
 pymysql.escape\_string(item['music\_bad']),  
 pymysql.escape\_string(item['music\_worst']),  
 pymysql.escape\_string(item['music\_review1']),  
 pymysql.escape\_string(item['music\_review2']),  
 pymysql.escape\_string(item['music\_review3']),  
 pymysql.escape\_string(item['music\_review4']),  
 pymysql.escape\_string(item['music\_review5']),  
 music['id']  
 ))  
 self.connect.commit()  
  
 return item  
  
 # def process\_music\_pop(self, item):

**⑧在middlewares.py中添加如下信息：**

class RotateUserAgentMiddleware(UserAgentMiddleware):

def \_\_init\_\_(self,user\_agent=''): self.user\_agent = user\_agent  
  
 def process\_request(self, request, spider):  
 ua = random.choice(self.user\_agent\_list)  
 if ua:  
 request.headers.setdefault('User-Agent', ua)  
 user\_agent\_list = [ "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.1 (KHTML, like Gecko) Chrome/22.0.1207.1 Safari/537.1",  
 "Mozilla/5.0 (X11; CrOS i686 2268.111.0) AppleWebKit/536.11 (KHTML, like Gecko) Chrome/20.0.1132.57 Safari/536.11",  
 "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/536.6 (KHTML, like Gecko) Chrome/20.0.1092.0 Safari/536.6",  
 "Mozilla/5.0 (Windows NT 6.2) AppleWebKit/536.6 (KHTML, like Gecko) Chrome/20.0.1090.0 Safari/536.6",  
 "Mozilla/5.0 (Windows NT 6.2; WOW64) AppleWebKit/537.1 (KHTML, like Gecko) Chrome/19.77.34.5 Safari/537.1",  
 "Mozilla/5.0 (X11; Linux x86\_64) AppleWebKit/536.5 (KHTML, like Gecko) Chrome/19.0.1084.9 Safari/536.5",  
 "Mozilla/5.0 (Windows NT 6.0) AppleWebKit/536.5 (KHTML, like Gecko) Chrome/19.0.1084.36 Safari/536.5",  
 "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/536.3 (KHTML, like Gecko) Chrome/19.0.1063.0 Safari/536.3",  
 "Mozilla/5.0 (Windows NT 5.1) AppleWebKit/536.3 (KHTML, like Gecko) Chrome/19.0.1063.0 Safari/536.3",  
 "Mozilla/5.0 (Macintosh; Intel Mac OS X 10\_8\_0) AppleWebKit/536.3 (KHTML, like Gecko) Chrome/19.0.1063.0 Safari/536.3",  
 "Mozilla/5.0 (Windows NT 6.2) AppleWebKit/536.3 (KHTML, like Gecko) Chrome/19.0.1062.0 Safari/536.3",  
 "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/536.3 (KHTML, like Gecko) Chrome/19.0.1062.0 Safari/536.3",  
 "Mozilla/5.0 (Windows NT 6.2) AppleWebKit/536.3 (KHTML, like Gecko) Chrome/19.0.1061.1 Safari/536.3",  
 "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/536.3 (KHTML, like Gecko) Chrome/19.0.1061.1 Safari/536.3",  
 "Mozilla/5.0 (Windows NT 6.1) AppleWebKit/536.3 (KHTML, like Gecko) Chrome/19.0.1061.1 Safari/536.3",  
 "Mozilla/5.0 (Windows NT 6.2) AppleWebKit/536.3 (KHTML, like Gecko) Chrome/19.0.1061.0 Safari/536.3",  
 "Mozilla/5.0 (X11; Linux x86\_64) AppleWebKit/535.24 (KHTML, like Gecko) Chrome/19.0.1055.1 Safari/535.24",  
 "Mozilla/5.0 (Windows NT 6.2; WOW64) AppleWebKit/535.24 (KHTML, like Gecko) Chrome/19.0.1055.1 Safari/535.24"]

**⑨通过启动文件运行**



## 4.2平台前端代码展示

**4.2平台前端代码展示**

MusicManage.heml页面

<!DOCTYPE html>

<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <title>豆瓣音乐管理</title>  
 <link rel="stylesheet" href="/js/layui/css/layui.css">  
</head>  
<body>  
<div class="layui-row" id="divSearch" >  
 <div class="layui-form" >  
 <div class="layui-form-item layui-inline" style="padding-left: 10%;">  
  
 <div class="layui-inline">  
 <input type="text" name="musicName" id="musicName" placeholder="需要检索的专辑名" class="layui-input" />  
 </div>  
 </div>  
  
 <div class="layui-form-item layui-inline" >  
  
 <div class="layui-inline">  
 <input type="text" name="musicName" id="musicVotes" placeholder="需要检索的唱片公司" class="layui-input" />  
 </div>  
 </div>  
  
 <div class="layui-form-item layui-inline" >  
<!-- <label class="layui-form-label">-->  
<!-- 艺术家-->  
<!-- </label>-->  
 <div class="layui-inline">  
 <input type="text" name="musicSinger" id="musicSinger" placeholder="需要检索的艺术家" class="layui-input" />  
 </div>  
 </div>  
  
 <div class="layui-form-item layui-inline" style="margin-left: 11%">  
<!-- <label class="layui-form-label">-->  
 流派  
<!-- </label>-->  
 <div class="layui-inline">  
 <select name="Type" id="musicType" class="layui-input">  
 <option value="" selected>全部</option>  
 <option value="流行" selected>流行</option>  
 <option value="民谣" selected>民谣</option>  
 <option value="说唱" selected>说唱</option>  
 <option value="摇滚" selected>摇滚</option>  
 <option value="原声" selected>原声</option>  
 <option value="古典" selected>古典</option>  
 <option value="电子" selected>电子</option>  
 <option value="爵士" selected>爵士</option>  
 <option value="" selected></option>  
 </select>  
 </div>  
 </div>  
  
 <div class="layui-form-item layui-inline" style="margin-left: 50px">  
 <!-- <label class="layui-form-label">-->  
 时间  
 <!-- </label>-->  
 <div class="layui-inline">  
 <select name="musicTime" id="musicTime" class="layui-input">  
 <option value="" selected>全部</option>  
 <option value="196" selected>1960-1969</option>  
 <option value="197" selected>1970-1979</option>  
 <option value="198" selected>1980-1989</option>  
 <option value="199" selected>1990-1999</option>  
 <option value="200" selected>2000-2009</option>  
 <option value="201" selected>2010-2019</option>  
 <option value="202" selected>2020-</option>  
 <option value="" selected></option>  
 </select>  
 </div>  
 </div>  
  
 <div class="layui-form-item layui-inline"style="margin-left: 40px">  
 <div class="layui-inline">  
 <button id="btnSearch" class="layui-btn layui-btn-radius">  
 <i class="layui-icon layui-icon-search"></i>检索  
 </button>  
 </div>  
 <div class="layui-inline">  
 <button id="btnSearchAll" class="layui-btn layui-btn-radius layui-btn-normal">  
 显示全部  
 </button>  
 </div>  
  
 </div>  
 </div>  
  
</div>  
  
<div class="layui-row" id="result">  
 <table id="tblResult" lay-filter="tblResult">  
 </table>  
</div>  
</body>  
<script type="text/javascript" src="/js/layui/layui.all.js"></script>  
<script type="text/javascript" src="/js/logic/musicManage.js"></script>  
</html>

Misucecharts.html源代码

<!DOCTYPE html>

<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <title>Title</title>  
 <link rel="stylesheet" href="/js/layui/css/layui.css">  
</head>  
<body>  
<div class="layui-container">  
 <div class="layui-row" >  
 <center><h1> 豆瓣音乐库数据图形化显示</h1></center>  
 </div>  
  
  
 <div class="layui-collapse" lay-accordion>  
  
 <div class="layui-colla-item">  
 <h2 class="layui-colla-title">专辑/EP分数统计</h2>  
 <div class="layui-colla-content layui-show">  
<!-- <div class="layui-row layui-col-space10">-->  
<!-- 嵌套的折叠面板-->  
 <div class="layui-collapse" lay-accordion>  
 <div class="layui-colla-item">  
 <h2 class="layui-colla-title">条形图展示</h2>  
 <div class="layui-colla-content layui-show">  
 <div class="layui-row layui-col-space10">  
 <div id="musicCount" style="width: 850px; height: 400px;">1</div>  
 </div>  
 </div>  
 </div>  
 <div class="layui-colla-item">  
 <h2 class="layui-colla-title">饼图展示</h2>  
 <div class="layui-colla-content layui-show">  
 <div class="layui-row layui-col-space10">  
 <div id="musicCount2" style="width: 850px; height: 400px;">1</div>  
 </div>  
 </div>  
 </div>  
 <div class="layui-colla-item">  
 <h2 class="layui-colla-title">漏斗图展示</h2>  
 <div class="layui-colla-content layui-show">  
 <div class="layui-row layui-col-space10">  
 <div id="musicCount3" style="width: 850px; height: 400px;">1</div>  
 </div>  
 </div>  
 </div>  
 </div>  
 </div>  
  
 </div>  
  
  
 <div class="layui-colla-item">  
 <h2 class="layui-colla-title">音乐类型图形化显示</h2>  
 <div class="layui-colla-content layui-show">  
  
<!-- 嵌套的折叠面板-->  
 <div class="layui-collapse" lay-accordion>  
 <div class="layui-colla-item">  
 <h2 class="layui-colla-title">饼图展示</h2>  
 <div class="layui-colla-content layui-show">  
 <div class="layui-row layui-col-space10">  
 <div id="type" style="width: 800px; height: 400px;"></div>  
<!-- <div class="layui-col-md6"></div>-->  
 </div>  
 </div>  
 </div>  
 <div class="layui-colla-item">  
 <h2 class="layui-colla-title">条形图展示</h2>  
 <div class="layui-colla-content layui-show">  
 <div class="layui-row layui-col-space10">  
 <div id="type2" style="width: 800px; height: 400px;"></div>  
<!-- <div class="layui-col-md6"></div>-->  
 </div>  
 </div>  
 </div>  
 <div class="layui-colla-item">  
 <h2 class="layui-colla-title">漏斗图展示</h2>  
 <div class="layui-colla-content layui-show">  
 <div class="layui-row layui-col-space10">  
 <div id="type3" style="width: 850px; height: 500px;"></div>  
<!-- <div class="layui-col-md6"></div>-->  
 </div>  
 </div>  
 </div>  
 </div>  
 </div>  
  
 </div>  
  
 <div class="layui-colla-item">  
 <h2 class="layui-colla-title">专辑时间及评分</h2>  
 <div class="layui-colla-content layui-show">  
 <div class="layui-row layui-col-space10">  
 <div class="layui-col-lg6" style=" ">  
 <!-- <h2>艺术家的作品评分</h2>-->  
 <div style=" margin-top: 5px">  
 <label class="layui-form-label">  
 艺术家  
 </label>  
 <div class="layui-inline">  
 <input type="text" name="musicSinger" id="musicSinger" placeholder="需要检索的艺术家" class="layui-input" />  
 </div>  
 <div class="layui-inline">  
 <button id="btnEcharts" class="layui-btn layui-btn-radius">  
 <i class="layui-icon layui-icon-search"></i>检索  
 </button>  
 </div>  
 </div>  
 <table id="artist" lay-filter="tblResult" style="width: 800px; height: 400px;"></table>  
 </div>  
 </div>  
 </div>  
 </div>  
  
 </div>

</div>  
  
</body>  
</html>  
<script src="https://cdn.bootcss.com/echarts/4.8.0/echarts.min.js"></script>  
<script src="https://echarts-www.cdn.bcebos.com/zh/asset/theme/shine.js"></script>  
<script type="text/javascript" src="/js/layui/layui.all.js"></script>  
<script type="text/javascript" src="/js/logic/musicEcharts.js"></script>  
  
<script src="/js/logic/dark.js"></script>

Login.html代码

<!DOCTYPE html>

<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <title>登陆-team01</title>  
 <link rel="stylesheet" href="/css/LogAndReg.css">  
 <link rel="stylesheet" href="/js/layui/css/layui.css">  
</head>  
<body>  
<div id="all">  
 <div id="contain">  
 <div id="left">  
 <img src="/image/left.png" />  
 </div>  
 <div id="right">  
 <div id="login" class="layui-form layui-form-pane">  
 <ul>  
 <li style="height: 30px">  
 <div id="login\_head" >  
 <div id="login\_head\_font" style="">  
 <h2>登陆</h2>  
 </div>  
 <div id="login\_head\_a" style="float: left;">  
 <p style="margin-left: 40%; width: 560px;">没有账号?&nbsp;&nbsp;&nbsp;<a href="adduser.html">点击注册</a></p>  
 </div>  
 </div>  
 </li>  
<!-- <form action ="/logic/user/login" method="post">-->  
 <li class="layui-form-item" >  
 <label class="layui-form-label">用户名:</label>  
 <div class="layui-input-block">  
 <input class="layui-input" type="text" id="uid" required lay-verify="required" name="uid">  
 </div>  
 </li>  
 <li class="layui-form-item">  
 <label class="layui-form-label">密码</label>  
 <div class="layui-input-block">  
 <input class="layui-input" type="password" id="pwd" required lay-verify="required" name="pwd">  
 </div>  
 </li>  
 <li class="layui-form-item">  
 <div id="login\_main\_but">  
 <!-- <input class="layui-input" type="button" value="登陆" />-->  
 <button class="layui-btn layui-btn-fluid" id="btnLogin" lay-submit lay-filter="Login">登陆</button>  
 </div>  
  
  
 </li>  
<!-- </form>-->  
  
 <li>  
 <div id="login\_main\_pwdlink">  
 <a href="">已有帐号，忘记密码</a>  
 </div>  
 </li>  
 <li>  
 <div id="login\_main\_middle">  
  
 </div>  
 </li>  
 <li>  
 <div id="login\_main\_hr">  
 <div id="login\_main\_hrleft">  
 <hr class="layui-bg-black">  
 </div>  
 <div id="login\_main\_hrcen">  
 <p>其他方式登陆</p>  
 </div>  
 <div id="login\_main\_hrright">  
 <hr class="layui-bg-black">  
 </div>  
  
 </div>  
 </li>  
  
 <li>  
 <div id="login\_main\_bottom">  
 <div id="login\_main\_botweichat">  
 <a href="https://wx.qq.com/">  
 <i class="layui-icon layui-icon-login-wechat" style="font-size: 30px; color: green;"></i>  
 </a>  
 </div>  
 <div id="login\_main\_botqq">  
 <a href="https://im.qq.com/">  
 <i class="layui-icon layui-icon-login-qq" style="font-size: 30px; color: #1E9FFF;"></i>  
 </a>  
 </div>  
 <div id="login\_main\_botweibo">  
 <a href="https://weibo.com/">  
 <i class="layui-icon layui-icon-login-weibo" style="font-size: 30px; color: #FFB800;"></i>  
 </a>  
  
 </div>  
 <div id="login\_main\_botwindows">  
 <a href="https://www.microsoftstore.com.cn/">  
 <i class="layui-icon layui-icon-windows" style="font-size: 30px; color: #01AAED;"></i>  
 </a>  
 </div>  
 <div id="login\_main\_botothers">  
 <div id="">  
 <i class="layui-icon layui-icon-more-vertical" style="font-size: 30px; color: #8D8D8D;"></i>  
 </div>  
 </div>  
  
  
 </div>  
 </li>  
  
 </ul>  
  
 </div>  
  
 </div>  
 </div>  
</div>  
</body>  
  
<script type="text/javascript" src="/js/layui/layui.all.js"></script>  
<script type="text/javascript" src="/js/logic/Login.js"></script>  
</html>

adduser.html代码

<!DOCTYPE html>

<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <title>注册</title>  
 <link rel="stylesheet" href="/js/layui/css/layui.css">  
 <link rel="stylesheet" href="/css/LogAndReg.css">  
</head>  
<body>  
<div id="all">  
 <div id="contain">  
 <div id="left">  
 <img src="/image/left.png" />  
 </div>  
 <div id="right">  
 <div id="regis" class="layui-form layui-form-pane">  
 <ul>  
 <li style="height: 30px">  
 <div id="regis\_head">  
 <div id="regis\_\_head\_font" >  
 <h2>注册</h2>  
 </div>  
 <div id="regis\_head\_a" >  
 <p style="margin-left: 40%; width: 560px;">已有账号?&nbsp;&nbsp;&nbsp;<a href="Login.html">点击登陆</a></p>  
 </div>  
 </div>  
  
  
 </li>  
 <li class="layui-form-item">  
 <label class="layui-form-label">用户名:</label>  
 <div class="layui-input-block">  
 <input class="layui-input" type="text" id="uid" required lay-verify="required" name="uid" />  
 </div>  
 </li>  
 <li class="layui-form-item">  
 <label class="layui-form-label">密码:</label>  
 <div class="layui-input-block">  
 <input class="layui-input" type="password" id="pwd" required lay-verify="required" name="pwd" />  
 </div>  
 </li>  
 <li class="layui-form-item">  
 <label class="layui-form-label">确认密码:</label>  
 <div class="layui-input-block">  
 <input class="layui-input" type="password" id="rePwd" required lay-verify="required" name="rePwd" />  
 </div>  
  
 </li>  
 <li class="layui-form-item">  
 <label class="layui-form-label">昵称:</label>  
 <div class="layui-input-block">  
 <input class="layui-input" type="text" id="nickName" name="nickName" />  
 </div>  
  
 </li>  
  
  
 <li class="layui-form-item" >  
  
 <button class="layui-btn layui-btn-fluid" id="btnSave" lay-submit lay-filter="saveAndCommit">注册新用户</button>  
  
 </li>  
  
 <li>  
 <div id="login\_main\_hr">  
 <div id="login\_main\_hrleft">  
 <hr class="layui-bg-black">  
 </div>  
 <div id="login\_main\_hrcen">  
 <p>其他方式登陆</p>  
 </div>  
 <div id="login\_main\_hrright">  
 <hr class="layui-bg-black">  
 </div>  
  
 </div>  
 </li>  
  
 <li>  
 <div id="login\_main\_bottom">  
 <div id="login\_main\_botweichat">  
 <a href="https://wx.qq.com/">  
 <i class="layui-icon layui-icon-login-wechat" style="font-size: 30px; color: green;"></i>  
 </a>  
 </div>  
 <div id="login\_main\_botqq">  
 <a href="https://im.qq.com/">  
 <i class="layui-icon layui-icon-login-qq" style="font-size: 30px; color: #1E9FFF;"></i>  
 </a>  
 </div>  
 <div id="login\_main\_botweibo">  
 <a href="https://weibo.com/">  
 <i class="layui-icon layui-icon-login-weibo" style="font-size: 30px; color: #FFB800;"></i>  
 </a>  
 </div>  
 <div id="login\_main\_botwindows">  
 <a href="https://www.microsoftstore.com.cn/">  
 <i class="layui-icon layui-icon-windows" style="font-size: 30px; color: #01AAED;"></i>  
 </a>  
 </div>  
 <div id="login\_main\_botothers">  
 <div id="">  
 <i class="layui-icon layui-icon-more-vertical" style="font-size: 30px; color: #8D8D8D;"></i>  
 </div>  
 </div>  
 </div>  
 </li>  
 </ul>  
 </div>  
 </div>  
 </div>  
</div>  
</body>  
<script type="text/javascript" src="/js/layui/layui.all.js"></script>  
<script type="text/javascript" src="/js/logic/addUser.js"></script>  
</html>

LogAndReg.css代码

body{

width:1400px;  
 height: 700px;  
 background-image: url("/image/back.jpg");  
 background-repeat: no-repeat;  
 background-size:contain;  
}  
#all{  
 width: 1400px;  
 height: 650px;  
}  
#contain{  
 width: 1400px;  
 height: 560px;  
 margin-left: 200px;  
 margin-top: 60px;  
}  
#left{  
 width: 450px;  
 height: 555px;  
 float: left;  
}  
#right{  
 margin-left: 0;  
 width: 454px;  
 height: 553px;  
 border-style:solid;  
 border-width:1px;  
 border-color:grey;  
 float: left;  
}  
#bottom{  
 width: 1400px;  
 height: 150px;  
}  
#bottom\_left{  
 float: left;  
 width: 200px;  
 height: 150px;  
}  
#bottom\_middle{  
 float: left;  
 width: 870px;  
 height: 150px;  
}  
#bottom-right{  
 float: left;  
 width: 200px;  
 height: 150px;  
}  
#login{  
 margin-top: 50px;  
 margin-left: 30px;  
 margin-right: 30px;  
}  
#login\_head\_font{  
 float: left;  
}  
#login\_head\_a{  
 float: left;  
}  
#login\_main\_but{  
 margin-top: 40px;  
}  
#login\_main\_pwdlink{  
 margin-top: 30px;  
 margin-left: 30%;  
}  
#login\_main\_middle{  
 height: 33px;  
 width: 400px;  
}  
#login\_main\_hr{  
 margin-top: 100px;  
}  
#login\_main\_hrleft{  
 width: 130px;  
 height: 30px;  
 float: left;  
}  
#login\_main\_hrcen{  
 width: 130px;  
 height: 30px;  
 text-align: center;  
 float: left;  
}  
#login\_main\_hrright{  
 width: 130px;  
 height: 30px;  
 float: left;  
}  
  
#login\_main\_botweichat{  
 margin-left: 30px;  
 float:left;  
}  
#login\_main\_botqq{  
 margin-left: 50px;  
 float: left;  
}  
#login\_main\_botweibo{  
 margin-left: 50px;  
 float: left;  
}  
#login\_main\_botwindows{  
 margin-left: 50px;  
 float: left;  
}  
#login\_main\_botothers{  
 margin-left: 50px;  
 float: left;  
}  
#regis\_\_head\_font{  
 float: left;  
}  
#regis\_head\_a{  
 float: left;  
}  
#regis{  
 margin-top: 50px;  
 margin-left: 30px;  
 margin-right: 30px;  
}

## 4.3 平台功能实现思路及js和后端源代码

**4.3 平台功能实现思路及js和后端源代码**

### 4.3.1登陆功能

**4.3.1登陆功能**

控制器处理前端web的login请求，调用service层的login接口，接口实现层调用mapper层的selectbyuid方法查询数据库中的值，将查询到的结果返回到接口实现层，此时有“登陆成功”“用户名不存在”“密码输入错误”“用户未激活”“用户被屏蔽”5种情况，接口返回不同的5个值到控制器，控制器将值传给js，js来控制界面跳转

核心代码部分

Js部分

layui.use(['layer','form','jquery'], function () {

var layer = layui.layer;  
 var form = layui.form;  
 var $ = layui.jquery;

$('#btnLogin').click(function (event) {  
 //alert('successful');  
 $.post(  
 "/logic/user/login",  
 {  
 'uid': $('#uid').val(),  
 'pwd': $('#pwd').val()  
 },  
 function (result) {  
 if (result == 1){  
 ***window***.location.href="/html/mainPage.html";  
 }  
 if (result == 2){  
 alert('密码输入错误');  
 }  
 if (result == 3){  
 alert('用户名不存在');  
 }  
 if(result == 4){  
 alert('该用户未激活');  
 }  
 if(result == 5){  
 alert('该用户被锁定');  
 }  
 }  
 )  
})

});

Controller层

@RequestMapping(value = "/login")

@ResponseBody  
public int login(String uid, String pwd){  
 int flag = service.login(uid,pwd);  
 return flag;  
}

接口实现层

@Override

public int login(String uid, String pwd) {  
 User user = mapper.login(uid, pwd);  
 User user2=mapper.LoginSelectUid(uid);  
 if (user != null ){  
 if( user.getState() == 1){  
 return 1;//可以成功登陆，返回1  
 }  
 else if (user.getState()==0){  
 return 4;//用户未激活  
 }  
 else if(user.getState()==2){  
 return 5;//用户被锁定  
 }  
 }  
 if (user == null && user2!= null && user2.getUid() != null && user2.getPwd() != pwd) {  
 return 2;//密码输入错误  
 }  
 if(user == null && user2 ==null){  
 return 3;//用户名不存在  
 }  
 return 0;  
}

Mapper层

@Select("select \* from user where uid = #{uid} and pwd = #{pwd}")

public User login(@Param("uid") String uid, @Param("pwd") String pwd);  
@Select("select \* from user where uid = #{uid}")  
public User LoginSelectUid(@Param("uid") String uid);

### 4.3.2注册功能

**4.3.2注册功能**

点击按钮后，js发送post add请求，由控制器处理前端web的请求，调用service层的adduser接口，接口实现层调用mapper层的adduser方法向数据库插入新的数据，返回的都是user类型的数据，js中接受到的值部位null则显示弹窗，“建立用户成功”

相关代码

Js相关部分

layui.use(['layer','form','jquery'], function () {

var layer = layui.layer;  
 var form = layui.form;  
 var $ = layui.jquery;  
  
 form.on('submit(saveAndCommit)', function (data) {  
 $.post(  
 '/logic/user/add',  
 {  
 'uid' : data.field.uid,  
 'pwd' : data.field.pwd,  
 'nickName' : data.field.nickName  
 },  
 function (returnData) {  
 if (returnData != null){  
 layer.alert("用户建立成功");  
 }  
 }  
 )  
 });  
});

Controller层

@RequestMapping(value = "/add")

@ResponseBody  
public User addUser(User user){  
 User newUser = service.addUser(user);  
 return newUser;  
}

接口实现层

@Override

public User addUser(User user) {  
  
 if (mapper.selectByUid(user.getUid()) == null) {  
 user.setState(0);  
 mapper.addUser(user);  
 return user;  
 }  
 return null;  
}

Mapper层

@Insert("insert into user(uid,pwd,nick\_name,state) " +

"values(#{user.uid},#{user.pwd}, #{user.nickName}, #{user.state})")  
@Options(useGeneratedKeys = true, keyProperty = "id")  
public int addUser(@Param("user") User user);

### 4.3.3显示数据功能

**4.3.3显示数据功能**

点击按钮后，web层发送searchall请求，controller 层的musiccontraller控制器通过searchall方法进行处理，调用service层的searchall接口，接口实现层，将Music 类型的数据传入mapper层，调用mapper层中的selectall方法，最终返回数据一直到controller层，controller层用tabledata类组装数据response到js中，在web中显示数据

相关代码

Js部分

$('#btnSearchAll').click(function (event) {

// alert('successful');  
 table.reload('tblResult',{  
 url:'/logic/music/search all',  
 method:'post',  
 page:{  
 curr: 1  
 }  
 })  
  
})

Controller层

@RequestMapping(value = "/search all")

@ResponseBody  
public TableData searchall(Music music, int page, int limit){  
  
 TableData data =new TableData();  
 List<Music> result = service.searchall(music,page,limit);  
 data.setCode(0);  
 data.setMsg("");  
 data.setCount(service.searchallcount());  
 data.setData(result);  
  
 return data;  
}

接口实现层

@Override

public List<Music> searchall(Music music,int page, int limit) {  
  
 if (page > 0 && limit > 0){  
 return mapper.selectAll((page-1) \* limit, limit);  
 }  
 return mapper.selectAll(null,null);  
}

Mapper层

@Select("select music\_name, music\_singer, music\_time, music\_type,

music\_score," +  
 "music\_votes,music\_perfect,music\_good,music\_common,music\_bad,music\_worst" +  
 " from guyuemusictable "+  
 "limit #{start}, #{limit}")  
public List<Music> selectAll(@Param("start") Integer start, @Param("limit") Integer limit);

### 4.3.4搜索功能

**4.3.4搜索功能**

点击查询按钮，web层发送search请求，controller 层的musiccontraller控制器通过search方法进行处理，调用service层的search接口，传入Music类型的数据，search接口的实现首先将music类型的数据执行模糊查询的属性加上“%”，之后调用mapper层的selectbywhere方法，传入Music类型的数据，mapper层的select查询语句使<where><if></if></where>相关语句。最终返回Music类型的数据到controller层，controller层用tabledata类组装数据response到js中，在web中显示数据。

相关代码截图

Js部分

layui.use(['jquery', 'form', 'table', 'layer'], function () {

var $ = layui.jquery,  
 form = layui.form,  
 table = layui.table  
 layer = layui.layer;  
  
 table.render ({  
 elem: '#tblResult',  
 page: true,  
 limit: 10,  
 cols: [[  
 {field:'musicName', title:'名称', align:'center'},  
 {field:'musicSinger', title: '作者', align: 'center'},  
 {field:'musicTime', title: '发行年份', align: 'center'},  
 {field: 'musicType', title:'流派', align: "center"},  
 {field: 'musicScore', title:'评分', align: "center"},  
 {field: 'musicVotes', title:'唱片公司', align: "center"},  
 {field: 'musicPerfect', title:'五星占比', align: "center"},  
 {field: 'musicGood', title:'四星占比', align: "center"},  
 {field: 'musicCommon', title:'三星占比', align: "center"},  
 {field: 'musicBad', title:'两星占比', align: "center"},  
 {field: 'musicWorst', title:'一星占比', align: "center"},

]],  
 data:[]  
});

$('#btnSearch').click(function(event) {  
 table.reload('tblResult', {  
 url:'/logic/music/search',  
 method:'post',  
 where: {  
 // 'musicName' : $('#musicName').val(),  
 'musicName':$('#musicName').val(),  
 'musicSinger' : $('#musicSinger').val(),  
 'musicVotes':$('#musicVotes').val(),  
 'musicType':$('#musicType').val(),  
 'musicTime' : $('#musicTime').val()  
 },  
 page:{  
 curr: 1  
 }  
 })  
  
});

});

Controller层

@RequestMapping(value = "/search")

@ResponseBody  
public TableData search(Music music, int page, int limit){  
  
 TableData data =new TableData();  
 List<Music> result = service.search(music,page,limit);  
 data.setCode(0);  
 data.setMsg("");  
 data.setCount(service.searchcount(music));  
 data.setData(result);  
  
  
 return data;  
}

接口实现层

@Override

public List<Music> search(Music music, int page, int limit) {  
  
 if (music !=null && !"".equals(music.getMusicTime().trim())){  
 music.setMusicTime(music.getMusicTime() + "%");  
 }  
  
 if (music != null && !"".equals(music.getMusicSinger().trim())){  
 music.setMusicSinger("%" + music.getMusicSinger() + "%");  
 }  
 if (music != null && !"".equals(music.getMusicName().trim())){  
 music.setMusicName("%" + music.getMusicName() + "%");  
 }  
 if (music != null && !"".equals(music.getMusicVotes().trim())){  
 music.setMusicVotes("%" + music.getMusicVotes() + "%");  
 }  
 if (page > 0 && limit > 0){  
 return mapper.selectByWhere(music,(page-1) \* limit,limit);  
 }  
 return mapper.selectByWhere(music,null,null);  
  
 }

Mapper层

@Select("<script>"+

"select music\_name, music\_singer, music\_time , music\_type, music\_score," +  
 "music\_votes,music\_perfect,music\_good,music\_common,music\_bad,music\_worst"+  
 " from guyuemusictable"+  
 "<where>"+  
 " <if test='music.musicName != null and music.musicName.length > 0'>" +  
 " and music\_name like #{music.musicName}"+  
 " </if>"+  
 " <if test='music.musicVotes != null and music.musicVotes.length > 0'>" +  
 " and music\_votes like #{music.musicVotes}"+  
 " </if>"+  
 " <if test='music.musicTime != null and music.musicTime.length > 0'>" +  
 " and music\_time like #{music.musicTime}" +  
 " </if>" +  
 " <if test='music.musicSinger != null and music.musicSinger.length > 0'>" +  
 " and music\_singer like #{music.musicSinger}"+  
 " </if>"+  
 " <if test='music.musicType != null and music.musicType.length > 0'>"+  
 " and music\_type like #{music.musicType}"+  
 " </if>" +  
 "</where>"+  
 "<if test='start != null and limit != null'>"+  
 " limit #{start}, #{limit}"+  
 "</if>"+  
 "</script>")  
public List<Music> selectByWhere(@Param("music") Music music,  
 @Param("start") Integer start,  
 @Param("limit") Integer limit);

### 4.3.1 数据图形化展示功能

**4.3.1 数据图形化展示功能(以统计分数的数量的图标为例)**

Web层发送scorecount请求，传给controller层，musiccontroller控制器使用musicscorecount方法进行处理，调用service层中的searchcount接口，searchcount接口调用mapper层中的countbyscore方法进行相关查询。最终返回music类型的值到controller，controller层将值传入musicecharts.js中，musicecharts.js文件进行数据组装并展示到html中

相关代码截图

Js部分

$.post(

"/logic/music/scorecount",  
 {},  
 function (result) {  
 ***console***.log(result);  
  
 let score = [];  
 let count = [];  
 let pieCount = [];  
 //为柱状图组装数据  
 for (i = 0;i<result.length;i++){  
 score.push(result[i].musicScore);  
 count.push(result[i].count);  
  
 if(i <= 10)  
 pieCount.push({'value':result[i].count, 'name':result[i].musicScore});  
 }  
 let option = {

Controller层

@RequestMapping(value = "/scorecount")

@ResponseBody  
public List<Music> musicscorecount(){  
 return service.scorecount();  
}

Service接口实现层

@Override

public List<Music> scorecount() {  
 return mapper.countByScore();  
}

Mapper层

@Select("select music\_score,count(1)as count" +

" from guyuemusictable " +  
 " group by music\_score" +  
 " order by music\_score +0 desc")  
public List<Music> countByScore();

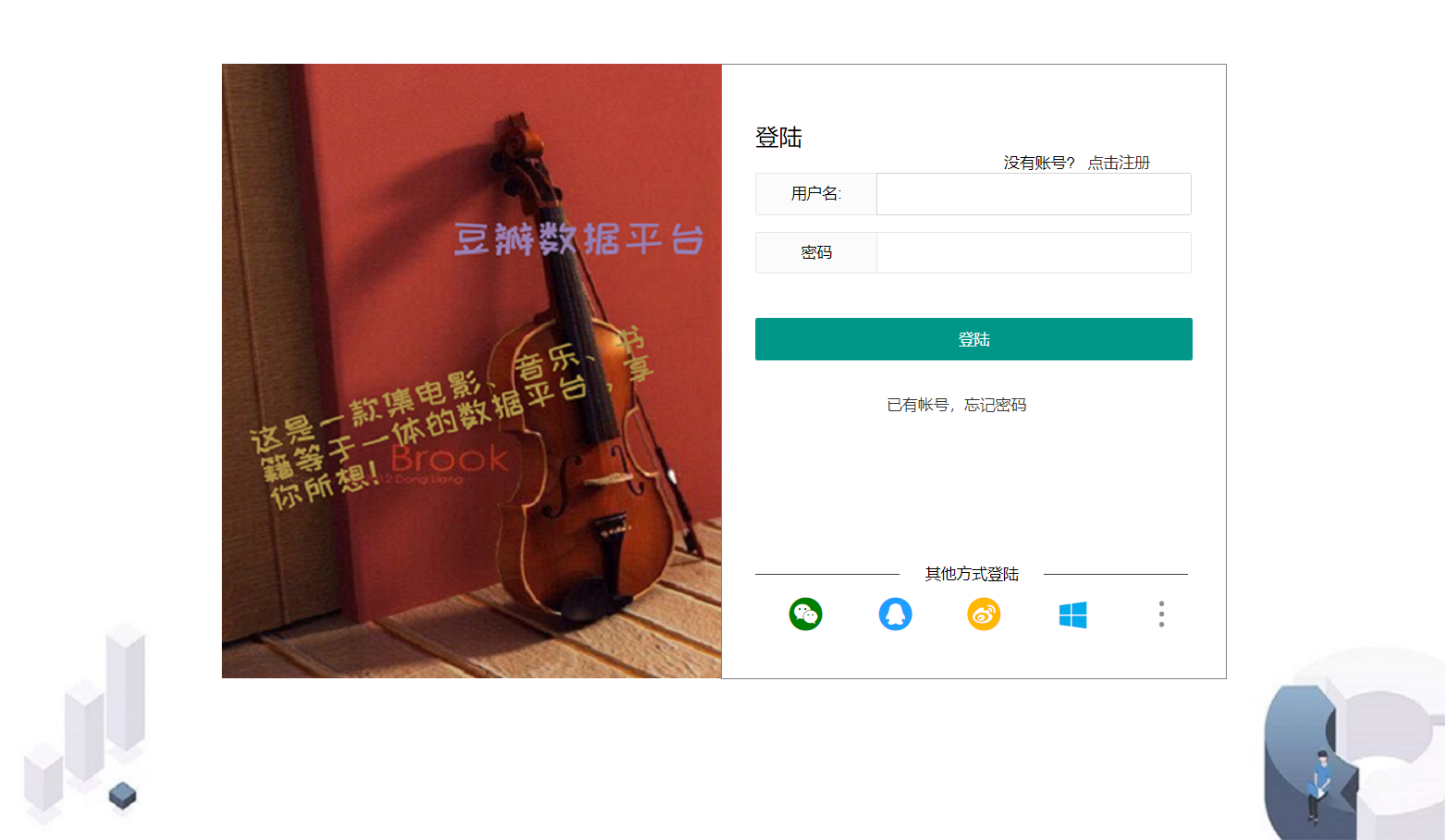
## 4.4 运行展示

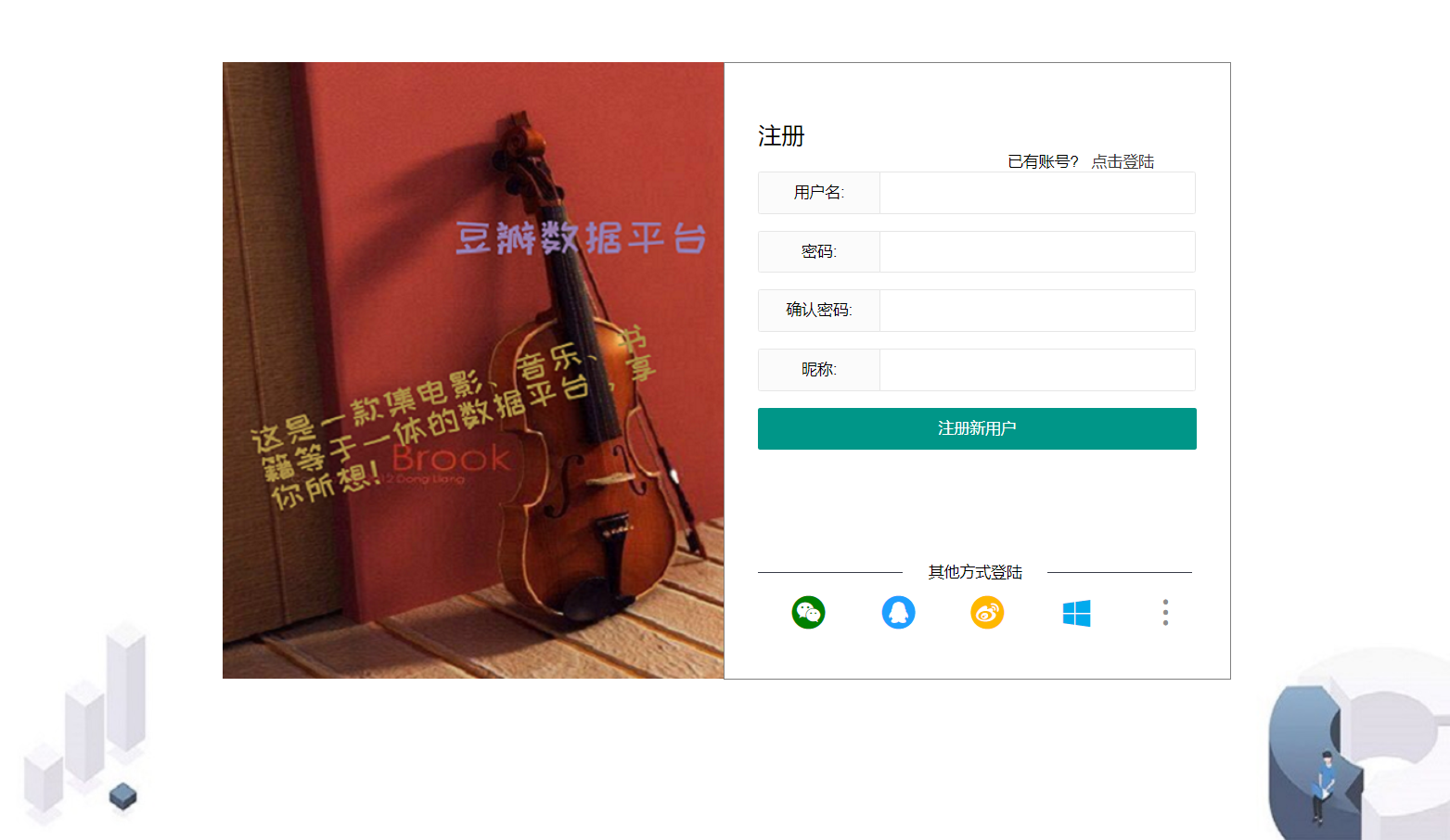
**4.4 运行展示**

Welcome页面 可跳转至登陆注册页面



登陆注册页面





进入主页面的音乐区，点击显示全部就会展示数据库中爬取的全部数据的相关信息



在搜索框内搜索关键字点击检索，可以查询到相关内容



在唱片公司的搜索框中输入关键字 点击检索，便可检索到相关内容



在歌手的搜索框里输入关键字，点击检索可以查询到相关内容

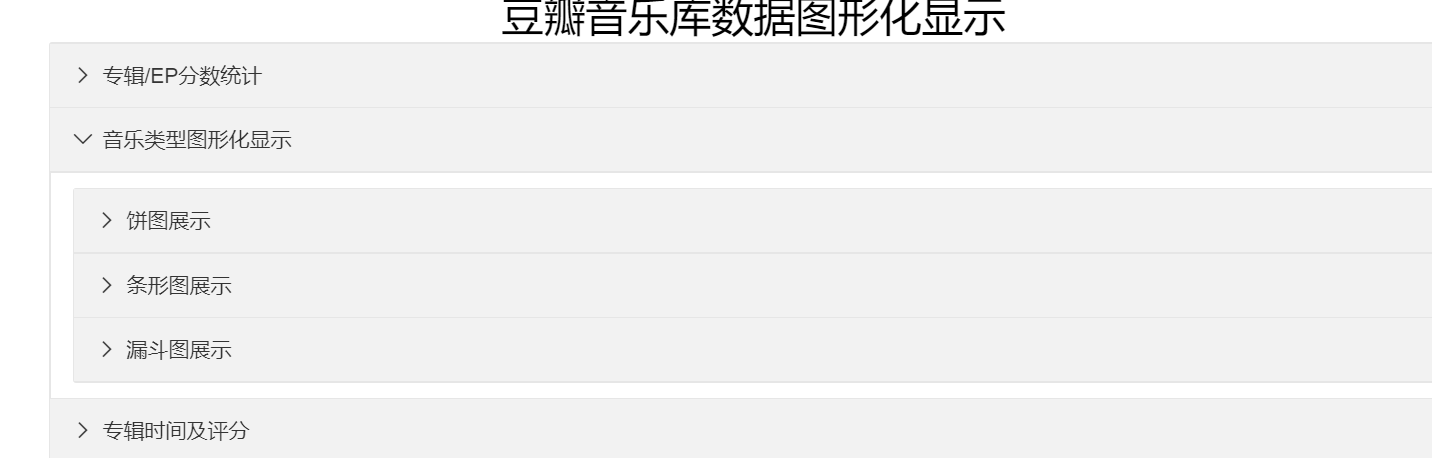


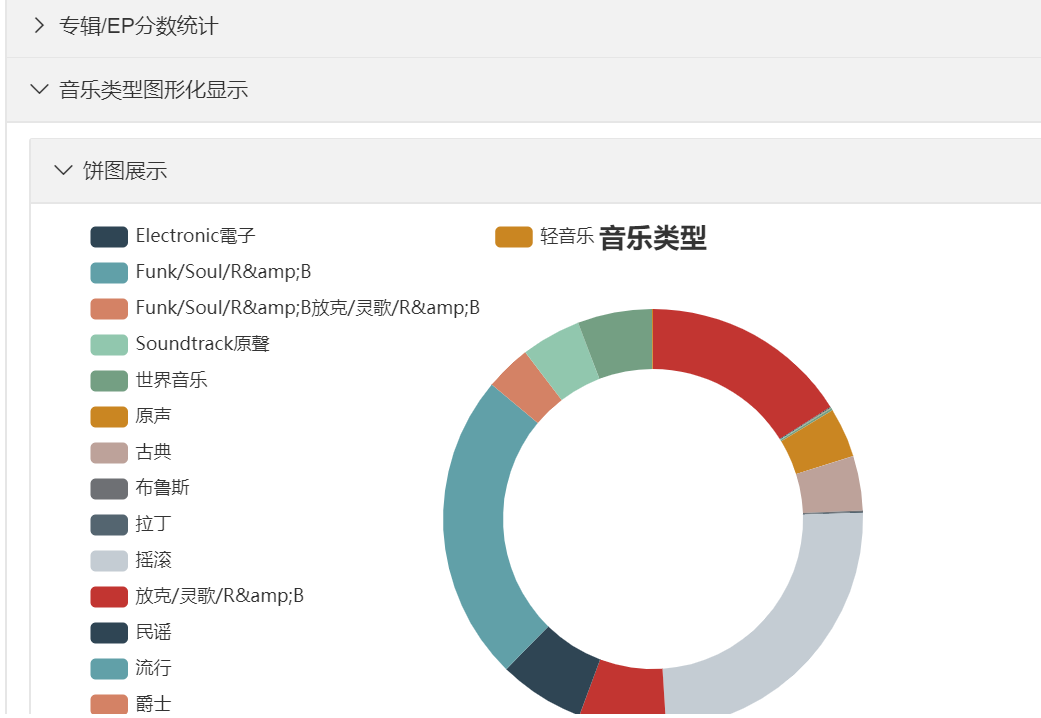
查询得到的结果通过分页展示给用户，同时支持翻页功能



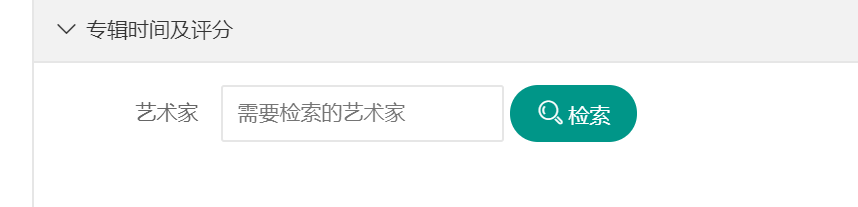


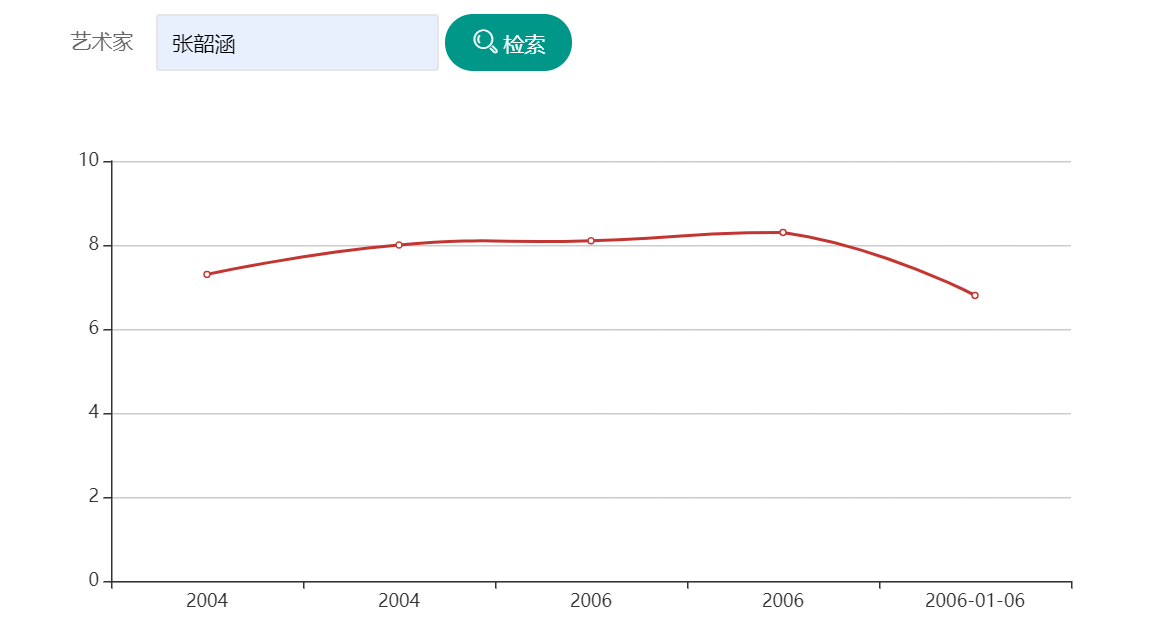
图形化显示采用折叠面板，点击可以显示出相关图形





其中，第三个图形支持搜索功能，通过搜索来给用户呈现出更有用的信息





# 5 系统测试

**5 系统测试**

## 5.1测试方法及过程

**5.1测试方法及过程**

本次项目开发作为全栈工程师的角色，对应模块的音乐区部分从数据获取到平台搭建再到功能完善，数据分析，等任务都是由一个人负责，清楚完整代码功能以及执行过程，因此本次使用的测试方法主要为白盒测试，对每一个模块分支进行调试。

拿数据图形化展示模块作为实例：

1. 搭建前端页面布局，为图表创建一个显示的位置，用颜色填充该模块在游览器上展示
2. 从echarts官网上将想要使用的图标复制到js代码部分，在游览器上看是否可以展示
3. 从数据库中将所需要统计的数据查询，调试查询成功后，在mapper层中写入相关代码
4. 在springboot的其他层写入代码将mapper中的值返回到控制器层，采用idea的debug方法查看是否返回需要的结果，或者在游览器的调试器上查看返回结果
5. 在js中进行数据组装，在平台上查看最终结果

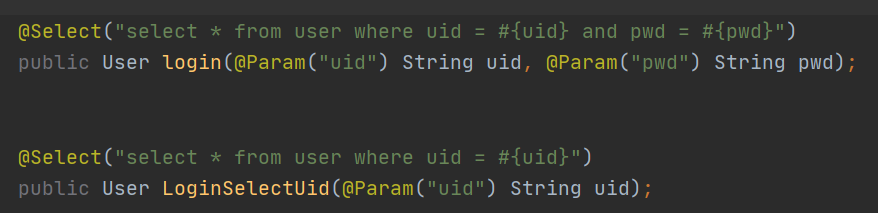
## 5.2出现的问题及解决方案

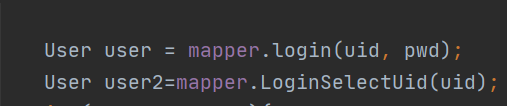
**5.2出现的问题及解决方案**

在平台开发过程中遇到了许多问题，其中登陆功能的实现及输入不正确的报错处理测试bug及调试的过程最为典型，拿来作为例子

最初只是实现了一个登陆的功能，controller返回一个字符串，直接跳转至登陆页面，但是如果输入错了，没有报错提示，对于用户的使用不够友好。因此需要对登陆功能进行优化。

首先controller层返回的值应该是整型来表示不同情况，加上@Responsbody注解，js接受返回值来进行不同操作。因为要实现对于密码输入错误和用户名输入错误以及用户状态等信息的提示信息，在maper层中还应加上只根据用户名来查找相关信息的方法。在接口实现层直接实例化两个对象，将mapper中的两个方法赋给实例化的对象，然后根据两个对象的属性值进行逻辑判断。

通过用户名密码 和 只用过用户名查找的结果 两个方法

 service层实例化的两个对象

在运行过程中，输入错误的用户名密码信息，控制台上报返回空指针的错误，百度查看可能是有对象未初始化，通过debug功能进行调试。如图为判断输入信息与数据库中不符的情况，此时user值为null，if语句的判断条件没有判断user为null的情况。user2.getPwd()写成了user.getPwd()，当用户名或密码输入不正确时， user为null，因此这样写找不到对应的state属性值，修改后，可以正常运行了。

