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: 基于豆瓣资源的个性化信息整合库平台

　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: 软件工程Z1801

姓名：史芯源

　实训起止日期：2020.6.15-2020.7.15

**1 引言** （一级标题，小三黑体，居中）

1.1项目概述（二级标题，四号黑体，左对齐）

随着科技时代的发展，我们的社会也在飞速的发展。作为一个信息化的社会，我们的衣食住行都离不开各种各样的信息。例如我们习惯网购各式各样的衣服，在网上搜索各种各样的美食，然后网站收集你习惯搜索的信息，再为你进行个性化推荐。毫不夸张的说，我们的生活已经离不开各种各样的信息了，我们每一个人每时每刻都在产生大量的数据。而就是为了处理如此海量的数据，大数据这门研究就应运而生了。

我们这个项目叫做基于豆瓣资源的个性化信息整合库平台，其目的就是做一个可以收集大量数据将其插入数据库，并在后台提取出来对数据进行分析整合的平台。这么做的意义在于我们可以收集大量我们想要的数据，然后针对我们想要的方面进行统计分析。例如拿我所爬取的电视剧数据来说，我们可以统计哪一年评分最高的前X名的电影，亦或是可以统计5星级评分的整体分布，甚至可以精确到某一条评论来估计是正面情绪还是负面情绪。我们将结果以图表的形式展现，这样可以更直观的得到我们想要的结果。通过一系列功能的实现，用户可以更好地检索自己想要的东西，我们也可以通过海量数据更好地了解用户，更好地为用户服务。

参考资料：通过老师对各个模块的讲解，以及老师写的简单例子，加上网上资料的自学，逐步搭建起整个框架。

1.2 需求分析

我们身处于信息化的时代，不仅我们想要的东西可以快速的检索出来，我们自己无时无刻也在产生大量数据，这就是大数据给我们生活所带来的实际影响。但是杂乱无章的数据是没有任何意义可言的，我们也无法快速找到我们想要的数据，更别提统计了。这就需要我们利用技术对数据加以整合分析，留下我们想要的数据，剔除我们不要的数据，然后将其规范化，最后可以利用得到的数据进行我们想要的分析，将结果进行可视化，这是我们做此项目的初衷。做出这个项目我们可以从任何一个网站使用爬虫爬出我们想要的数据，将其用一张规范的二维表整合进数据库，然后使用BS项目的开发结合前端的开发，将得到的数据提取出来，用前端技术做成一个平台，我们可以在这个平台上检索我们想要的数据，亦或是可以选择一些项目进行分析，用eCharts技术进行可视化，使展示更加直观、美观。这就是此项目的意义。

同时我们在此平台加入用户管理功能，使用此平台前需要登录，如果没有账户则需要注册。此外用户还分为普通用户和管理员用户，二者的权限有所差别，同时管理员用户可以管理普通用户。并且用户拥有标记自己喜欢的或者已看过的图书、电影的功能。

1.3 运行环境

软件环境：Gitee负责版本管理，提交项目；STS负责编写爬虫代码，利用爬虫爬取数据；IDEA负责服务器端以及前端代码编写——服务器端开发框架SSM（Spring+SpringMVC+MyBatis），前端开发技术（jQuery+LayUI+eCharts）；连接远程数据库MySQL(Mariadb)；平台开发模式：基于JSON数据的前后端分离式开发。最后使用浏览器打开最终成型的平台。

硬件环境：运行内存8G的PC即可。

**2 项目设计**（小三黑体，居中）（以个人主要负责内容为主）

2.1 设计思路

首先作为一个数据分析后台管理，第一步我们要获取数据，这边我们使用了scrpay框架进行数据爬取，我个人没有使用scrpay框架直接写了爬虫代码，设置了动态IP和用户协议来防止IP被封，提高爬取速度。然后就是前端界面，这块我们使用了layui框架，我构建了welcome界面，以及电视剧这部分。后端用java实现，创建了一套从数据库到前端界面的代码包括：controller（js文件调用，可以将所需数据push到前端）,mapper（从数据库提取数据的主要模块，撰写sql语言的地方）,entity（所需数据）,impl,service（接口）,

这一套流程可以将所需要的数据封装push到前端，供给之后之用的echarts图表，echarts作为数据展示的主要工具，进行数据可视化。

前端只写了两块包括数据查询/修改（因为爬取的数据没有修改点，这边电视剧模块没有写修改的功能，但是在用户那块进行了编写），以及数据可视化。

2.2 模块功能介绍

主要分为两部分：

第一部分：

数据查询功能，可以将爬虫爬取到的豆瓣电视剧信息在前端界面展示，可以针对导演名称，以及电视剧名称进行查询（这里可以添加更多的查询条件）。

第二部分：

数据分析/可视化：

第一个图表：

这里对爬取的电视剧评分进行的统计，由高到低的进行了数据展示，但是9.0之后的数据进行隐藏，全部展示略显拥挤。

第二个图表：

这个图表是针对电视剧发布时间进行的统计，统计了近年来电视剧的发布情况，得出近年来电视剧生产数量数年增加，2020年仅半年就有追平2019年的趋势，说明整个中国电视剧产业走向良好

第三个图表：

对爬取的所有电视剧星级信息进行了整理，对不同星级的占比进行了曲线展示，数据可得高星级还是少数，并且呈递减趋势。

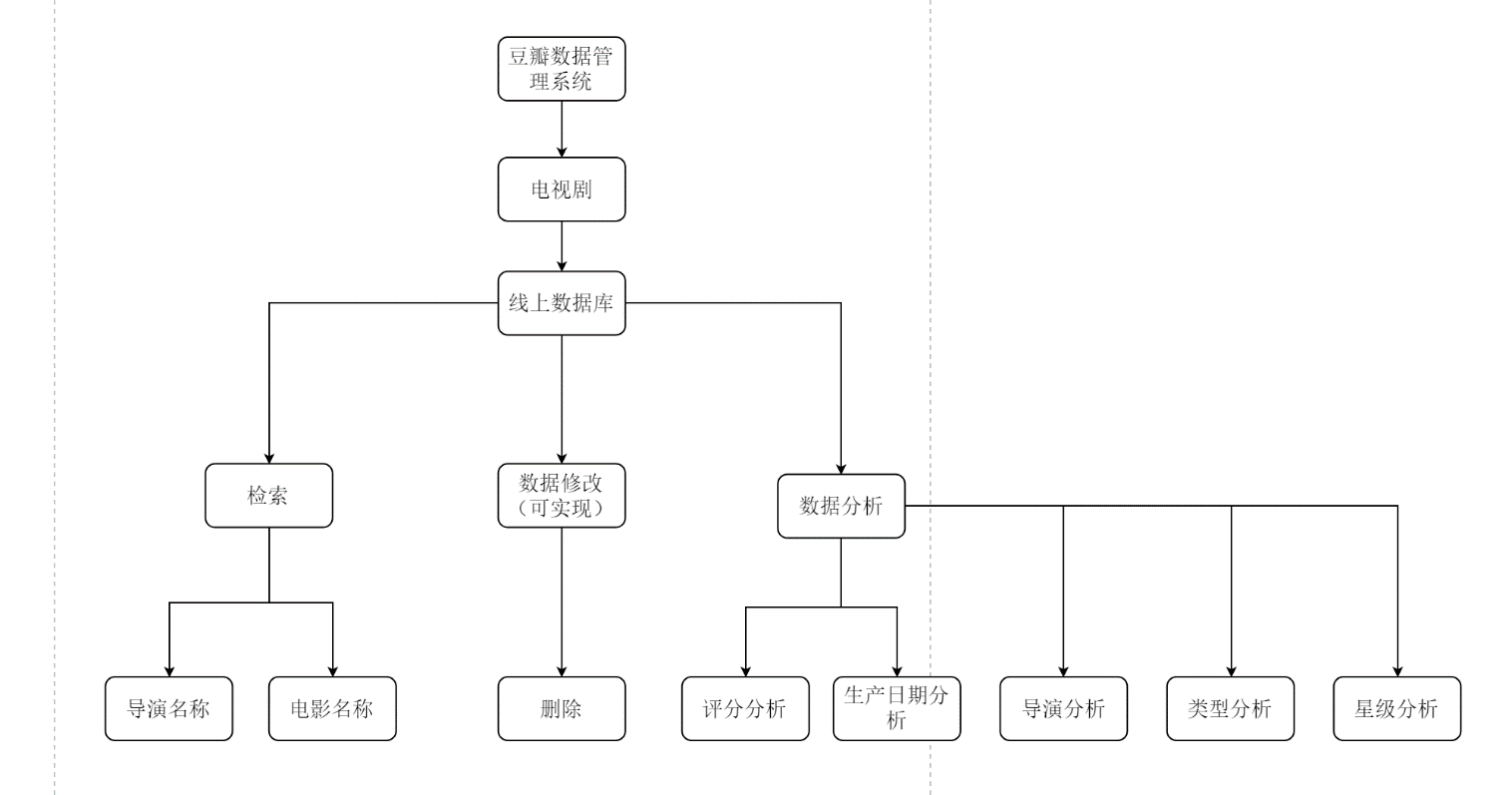
第四个图表：

对我所爬取的数据进行了简单的分类统计，统计了不同类型的电影各有多少。

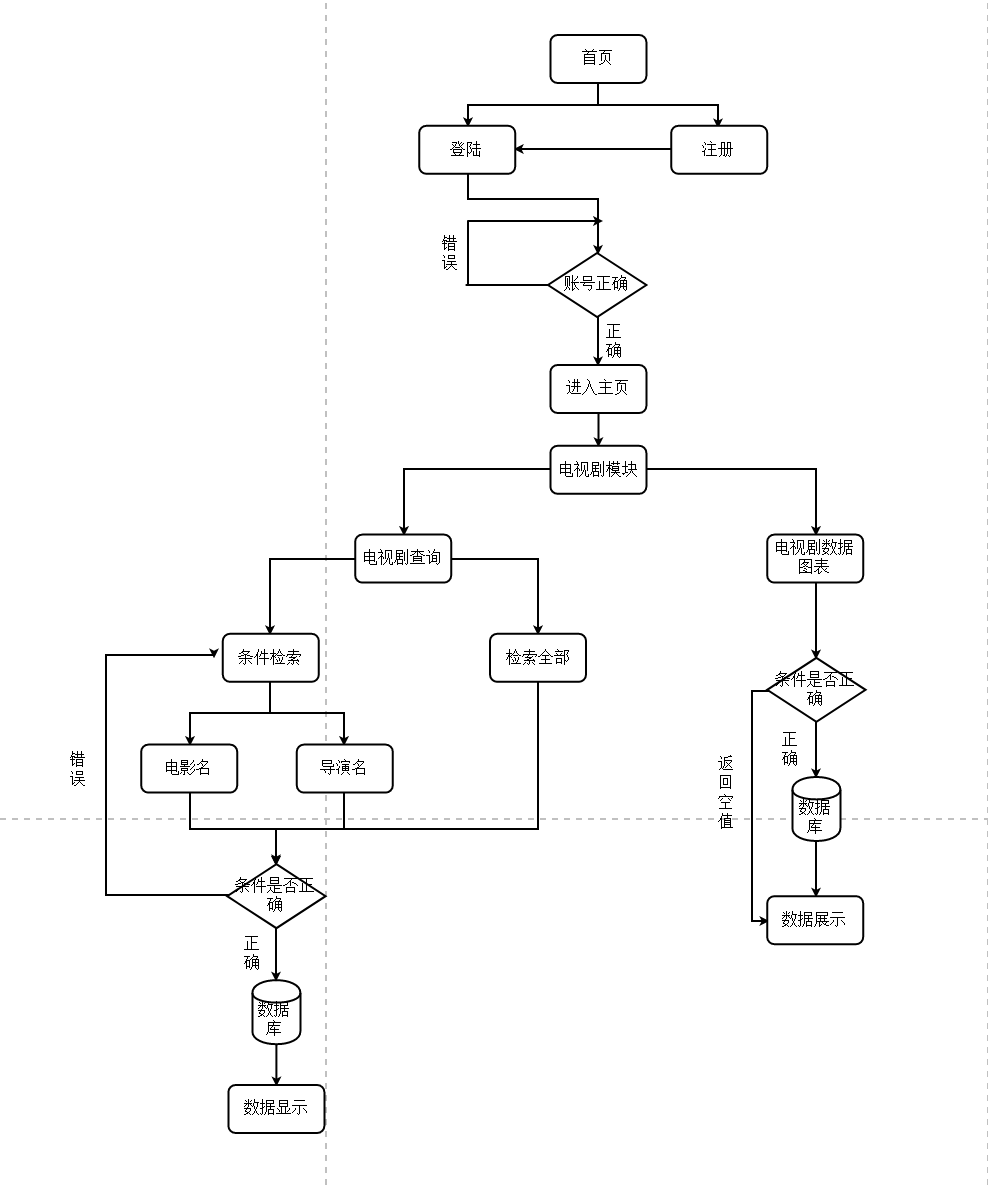
第五个图表：

对爬取的导演进行了数据汇总，汇总爬取到的数据有多少是他们创作的。分析导演的产出率。

2.3 模块结构图



2.4 程序流程图



2.5 功能设计分工

需详细标注每部分的内容，具体任务及任务安排等。

此次小组共四个人，在scrapy爬虫阶段每个人负责自己模块的编写，通过对不同页面的解析，爬取自己相对应的数据。各自在远程数据中建立自己的二维表，然后将爬来的数据插入二维表中，为后续奠定基础。

进入下一步编写后，仍然存在个人项目，但此时个人项目只是意义上的个人项目，大家通过版本控制实则在编写一个共通项目。需要由组长上传必要组件以及配置文件，组员通过gitee下载，然后系统自动导入所需运行文件即可。大家的文件夹目录均是一样的，在目录下创建自己的模块，自己模块包括数据的检索以及数据的展示。

再然后就是用户模块，用户模块分为前端代码以及后端代码，前端代码作为html代码以简洁、美观作为重要因素，后端代码需要实现一系列功能，如对接数据库，用户登录注册等内容。

分工完毕之后，先共同编写主界面等内容，然后再各自细化、为自己的模块添砖加瓦。总之，各个页面模块存在跳转、调用等关系，所以在编写时要注意它们之间的联系，它们既是独立存在也是紧密相联的，所以既要有独立的设计，也要存在互相联系。

**3 详细设计**（小三黑体，居中）

详细写明项目的设计（系统设计、数据库设计）等，并写明对所涉技术、核心功能等描述；

系统设计：

PYTHON部分：

我没有使用scrap框架直接编写的爬虫代码。我首先提取了豆瓣网主页的标签url对相应的url直接进行爬取，使用了beautifulsoup函数目的在于针对特定页面进行特定搜索，以此减少不必要的搜索，提高爬取效率。在爬取过程中使用了网络上的代理IP提高爬取速度。

进入相应的url就会自动搜索每个电影的url从而进入下一层，因为使用了beautifulsoup函数查找特定数据也比较方便。然后存入数据库中。并写了自动翻页的代码自动进入下一页，对于不能存储或失败的数据直接跳过。

JAVA部分：

使用mybatis框架，将页面数据通过js->controller->service->mapper到层，通过mapper层的底层实现达成前端页面与数据库的交互操作

前端页面包括用户模块(登录、注册、检索)，主体模块(书籍、音乐、电影、电视剧)，主题模块内每一分类又分为多个模块，包括查询模块以及多个不同的数据分析展示模块。

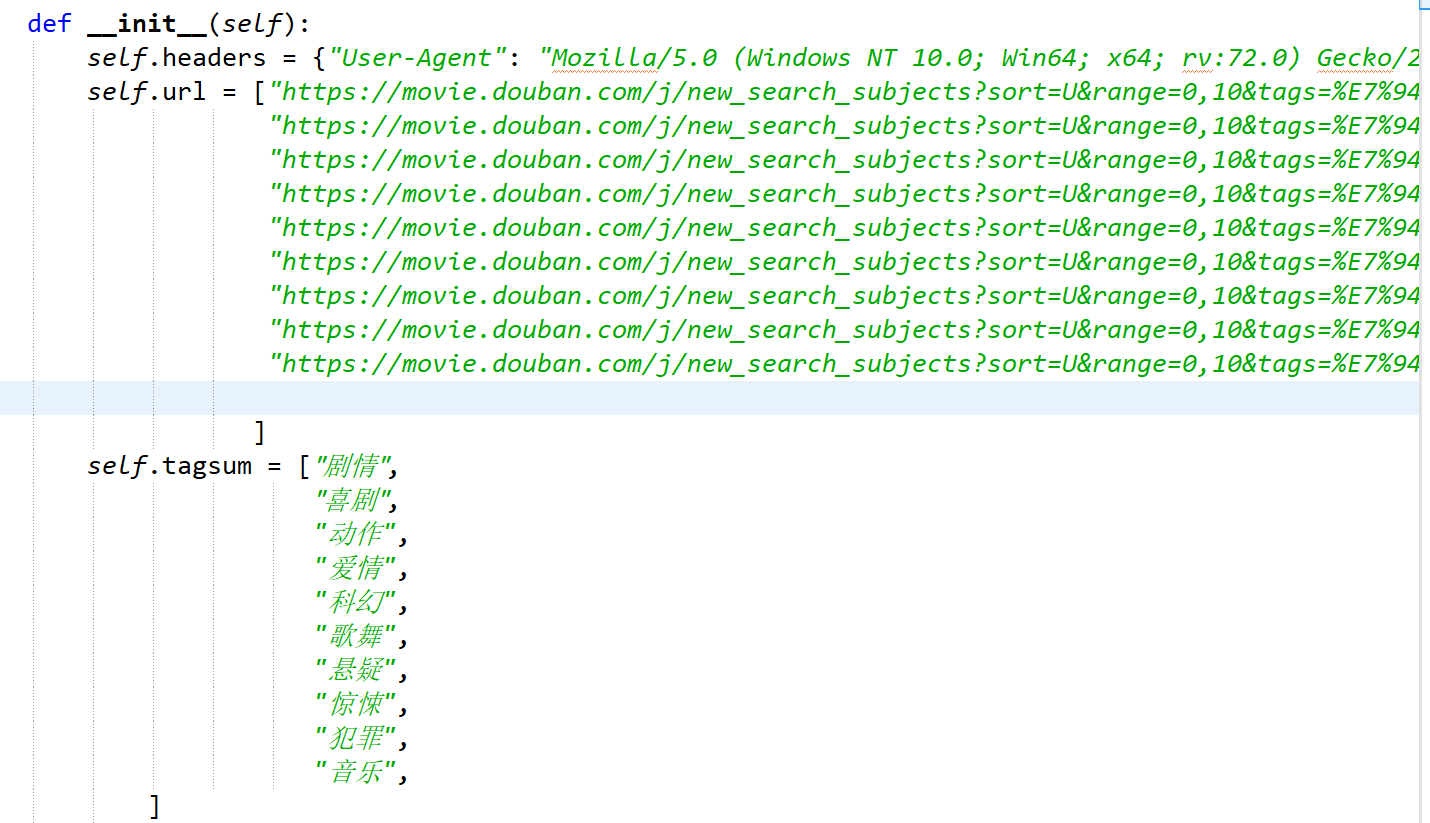
数据库设计：

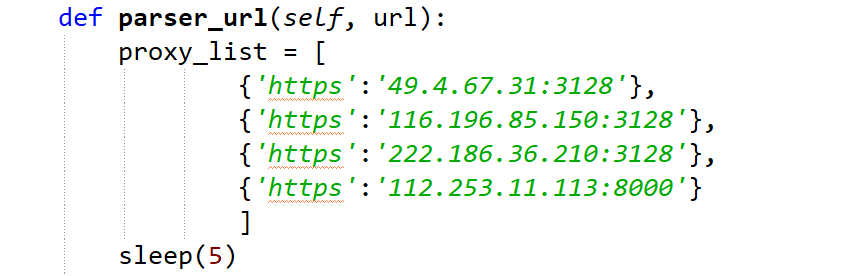
数据库我只建立了二张表，一个是电视剧信息表用来存储电视剧的相关数据。信息包括电视剧类型，评分，导演，等等。一个是用户管理表用户管理表是包含用户名、密码、昵称、用户状态、用户头像存储路径。

**4 实现方案**（小三黑体，居中）

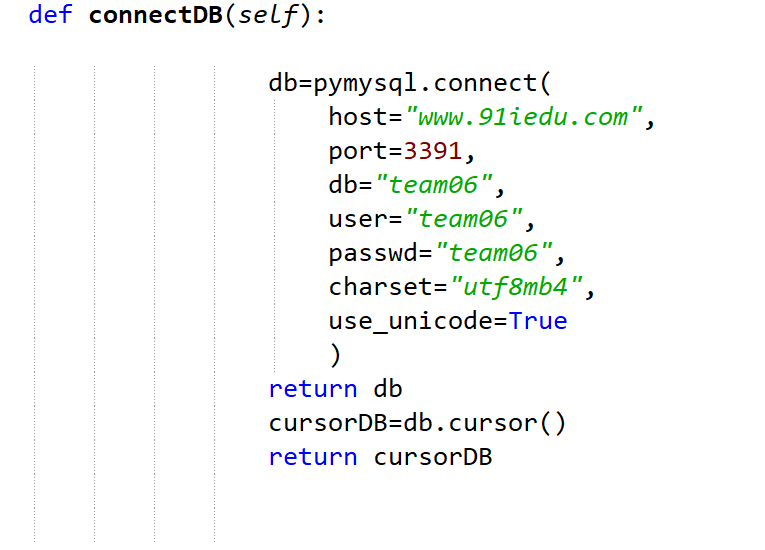
爬虫实现：

我个人没有使用scrapy框架。这里分步介绍一下我的代码结构：





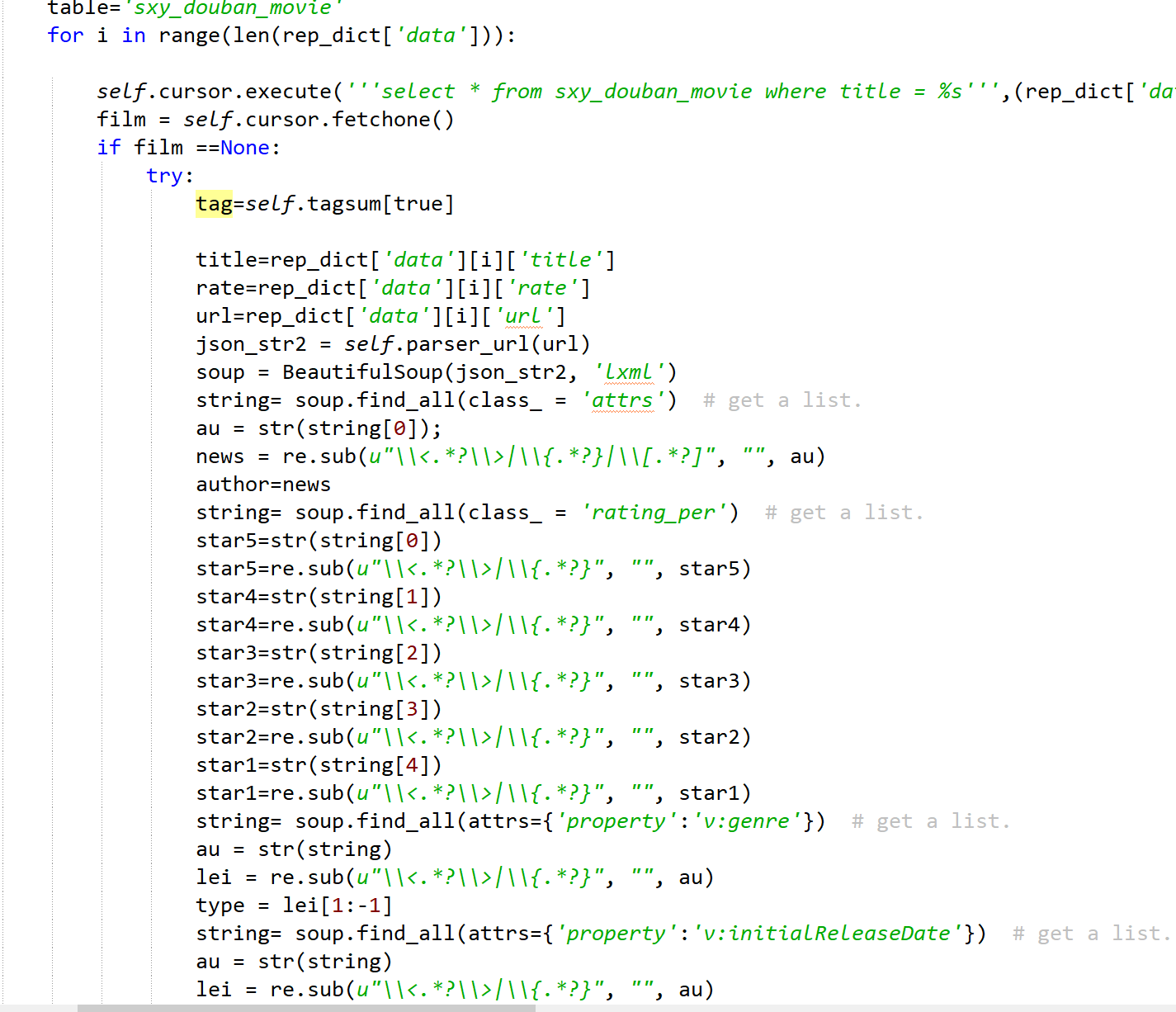
这部分为爬取目标页和对应的标签，以及IP代理我这边使用了<Ipidea球>的代理IP，因为我自身的ip在最初测试的时候因爬取速度过快被封掉了.



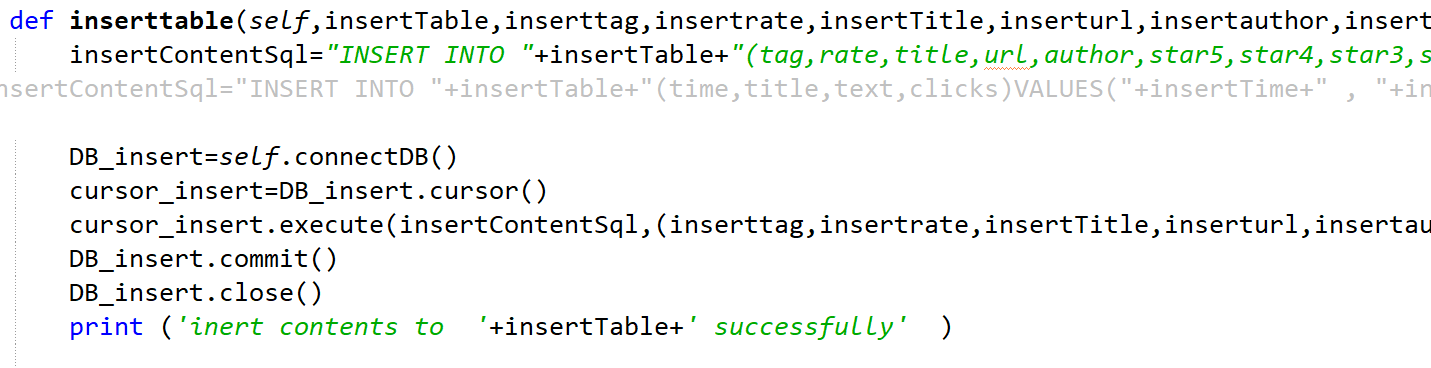
这部分是对数据库的链接



主题函数前半部分，这里主要控制着翻页以及数据的读取，和目前的读取量的显示.

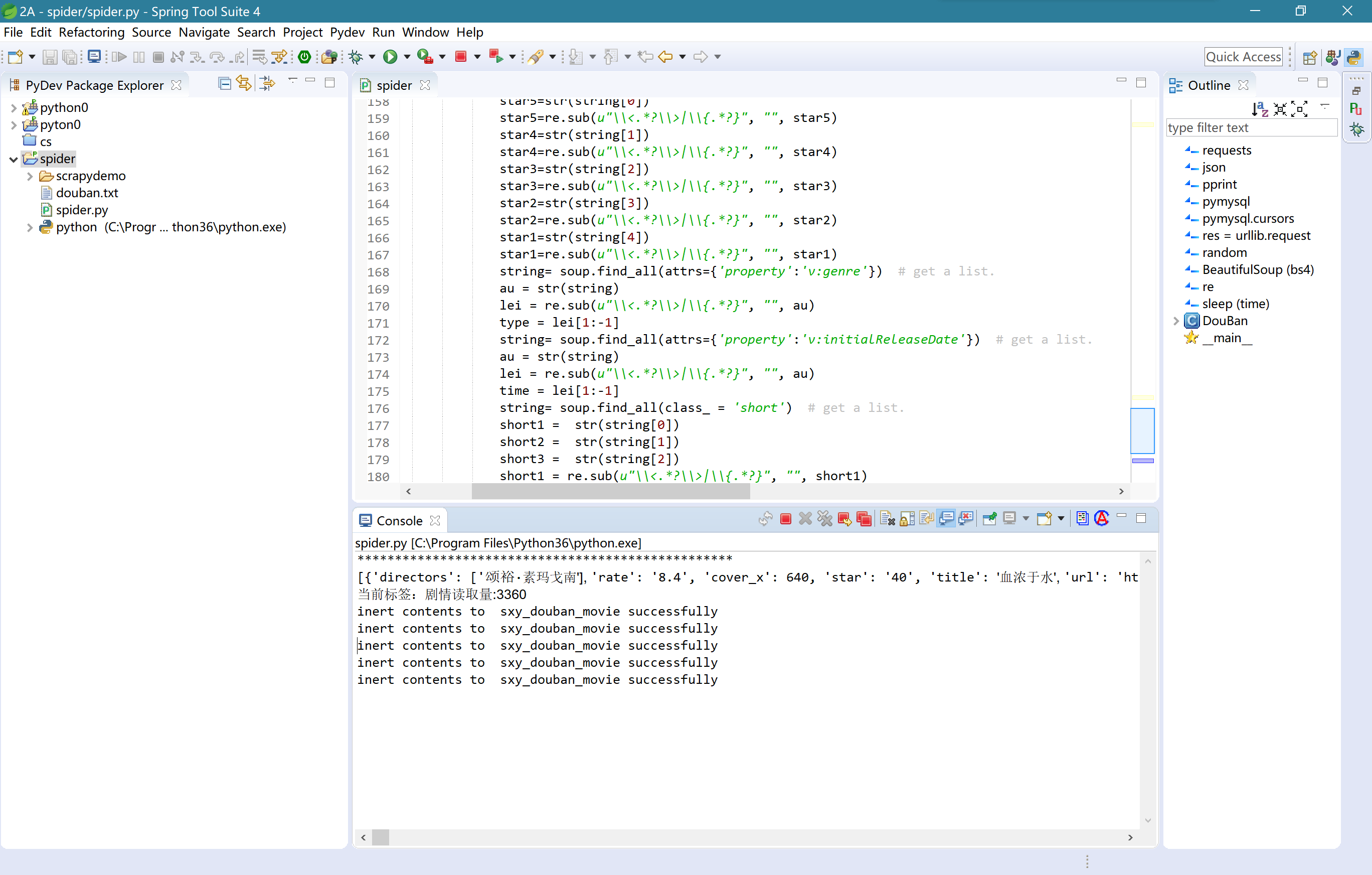


数据检索，插入主要在这里控制，这里使用了BeautifulSoup函数可以直接针对class定义id搜寻需要的目标比较方便.插入函数在下面:



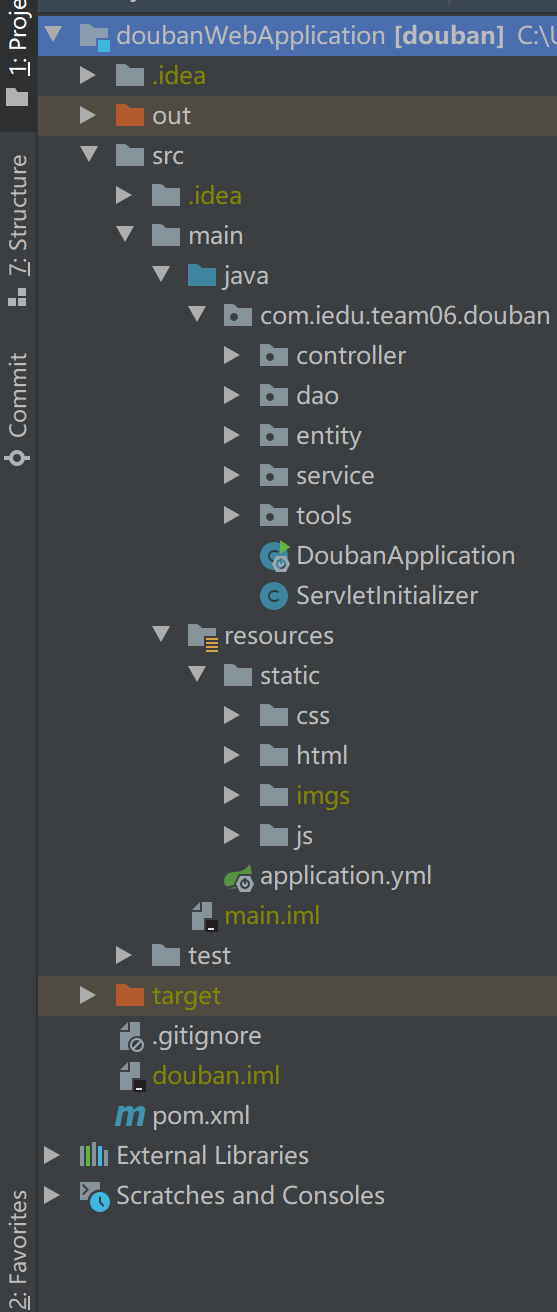
这就是我的爬虫基本结构。

运行结果:



IEDA部分：

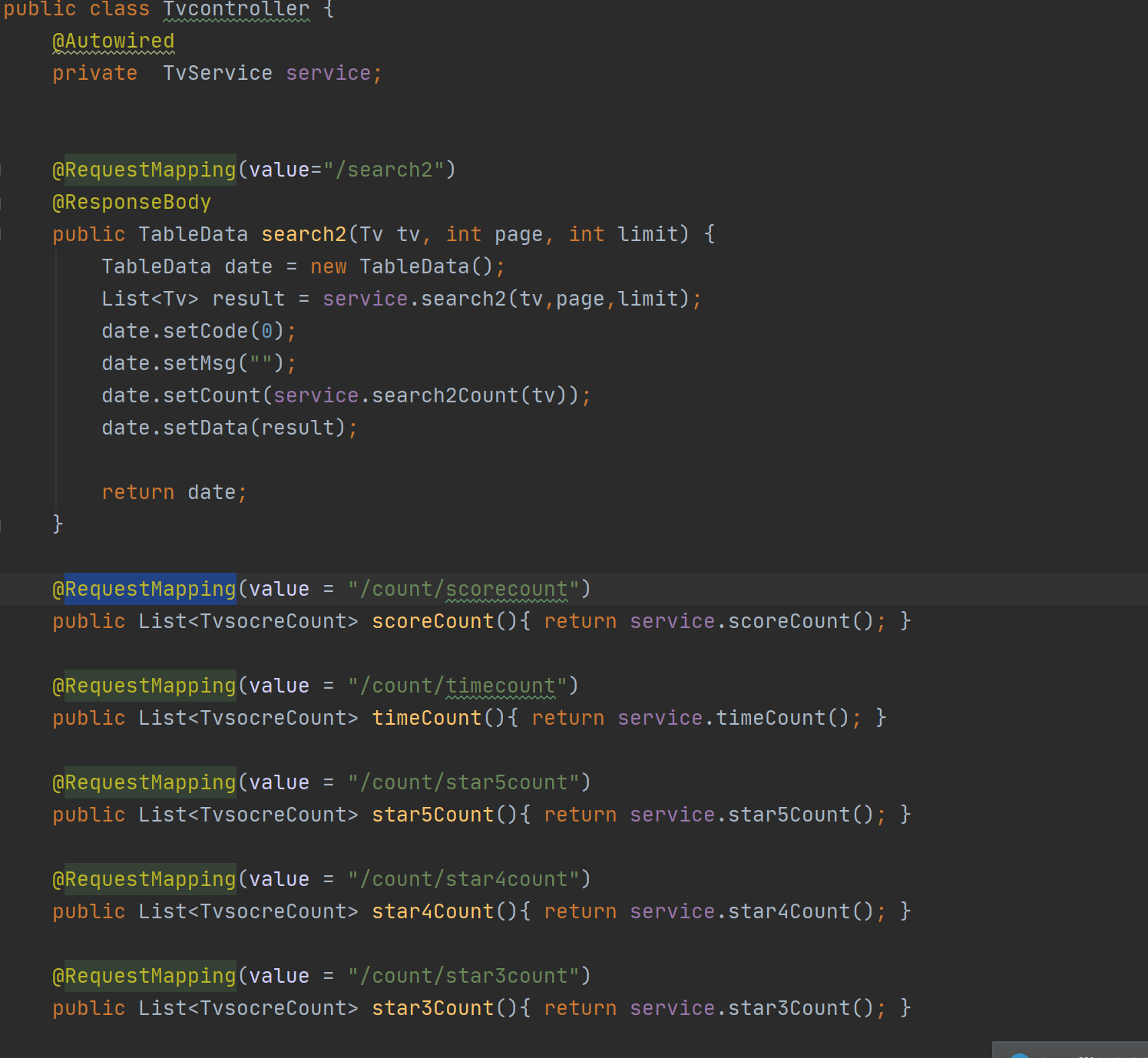
项目总述：



首先java部分：

包括controller，dao，entity，service，tool

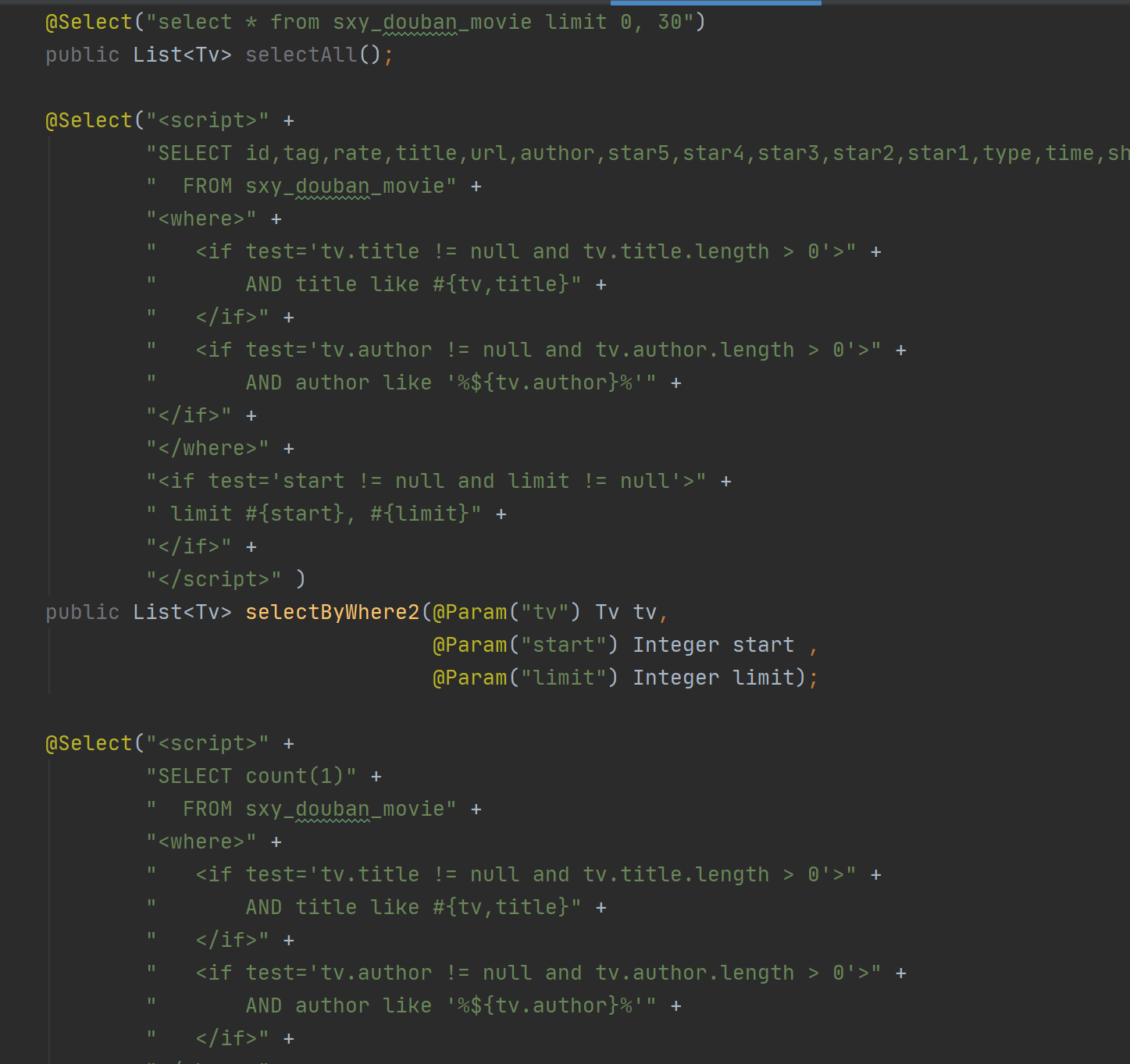
Controller层主要负责映射请求，在我们写好html以及对应的js文件，我们通过js文件调用controller中的请求使用我们的数据库信息。其中requestmapping就是映射请求。一会供给js文件使用。使用entity中的实体类，通过端口service，返回dao中查询到的数据。

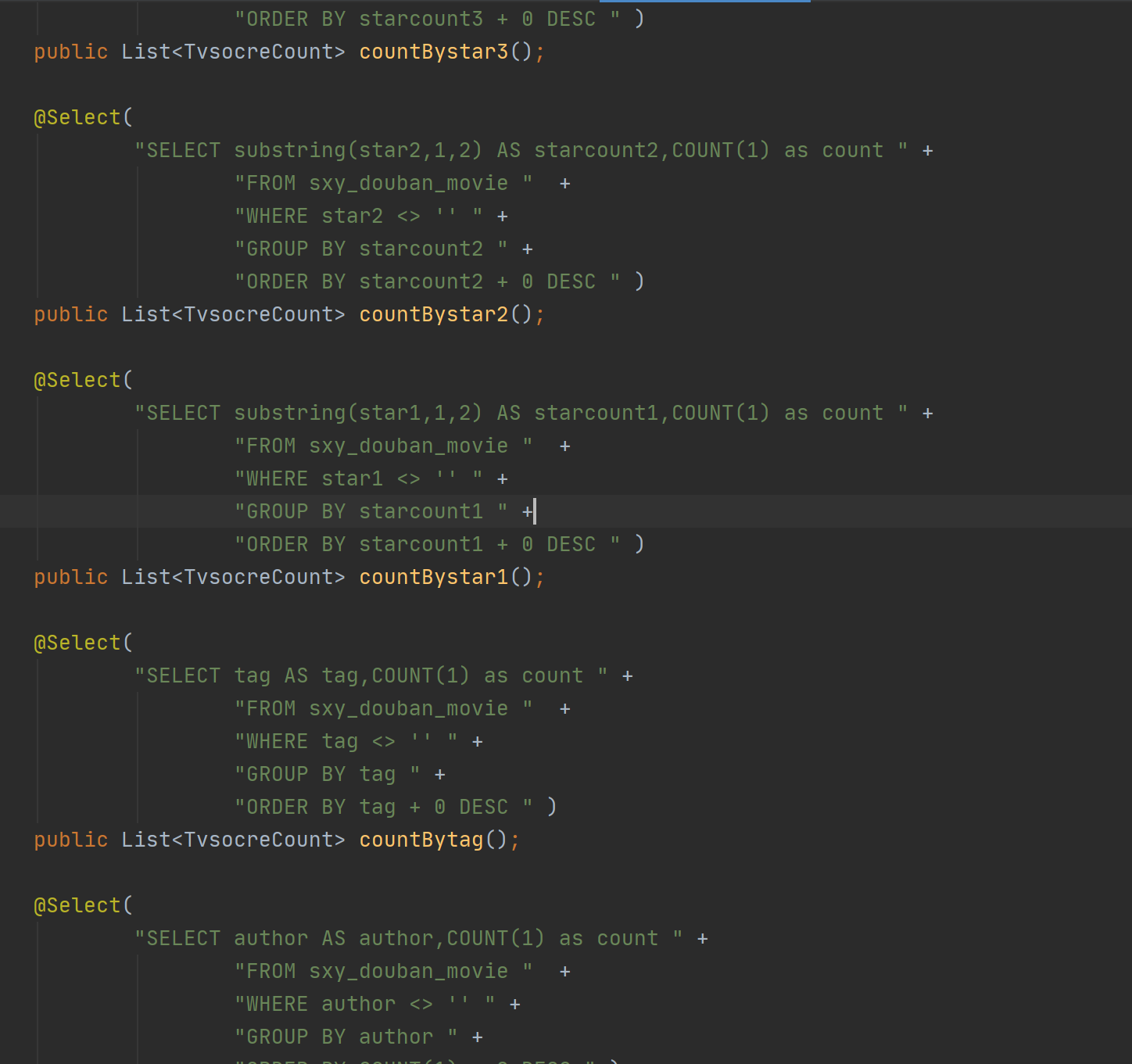


Dao层：

主要负责了sql语言的编写，从数据库提取信息并发送给端口。这里第一张截图主要负责了翻页功能，通过limit和page两个参数进行每页的数量和当前的页数。

第二张图主要是之后html中echarts图标中的数据查询。通过这一套代码push到前端界面。





Entity层：

实体类用于写所需数据的类型，也就是用来存储dao层查询到的数据的。

Service层：

就是端口用来连接这一整套的流程



第二部分resources

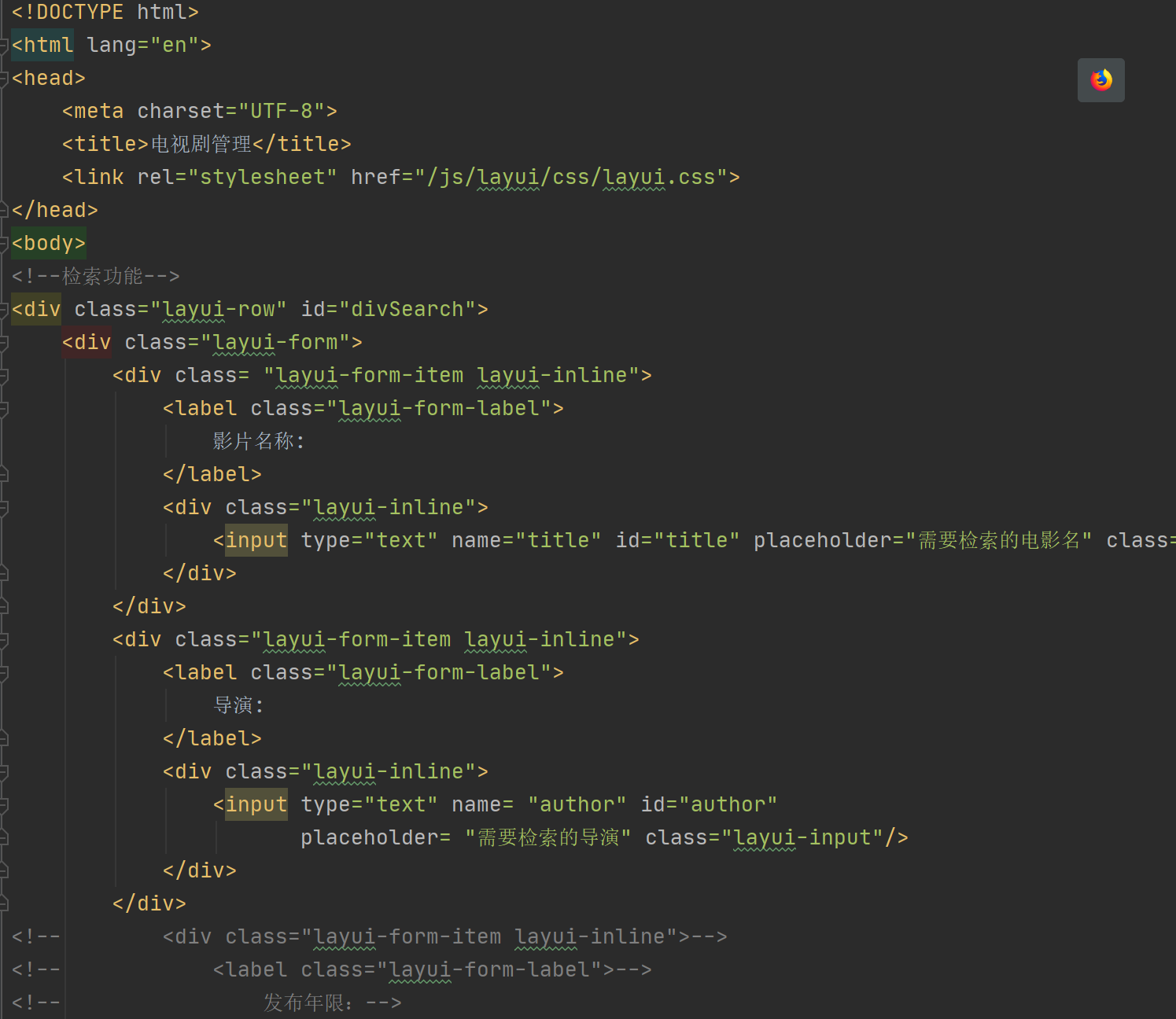
主要存储的前端界面以及相关配置。

主要是由html和js组成

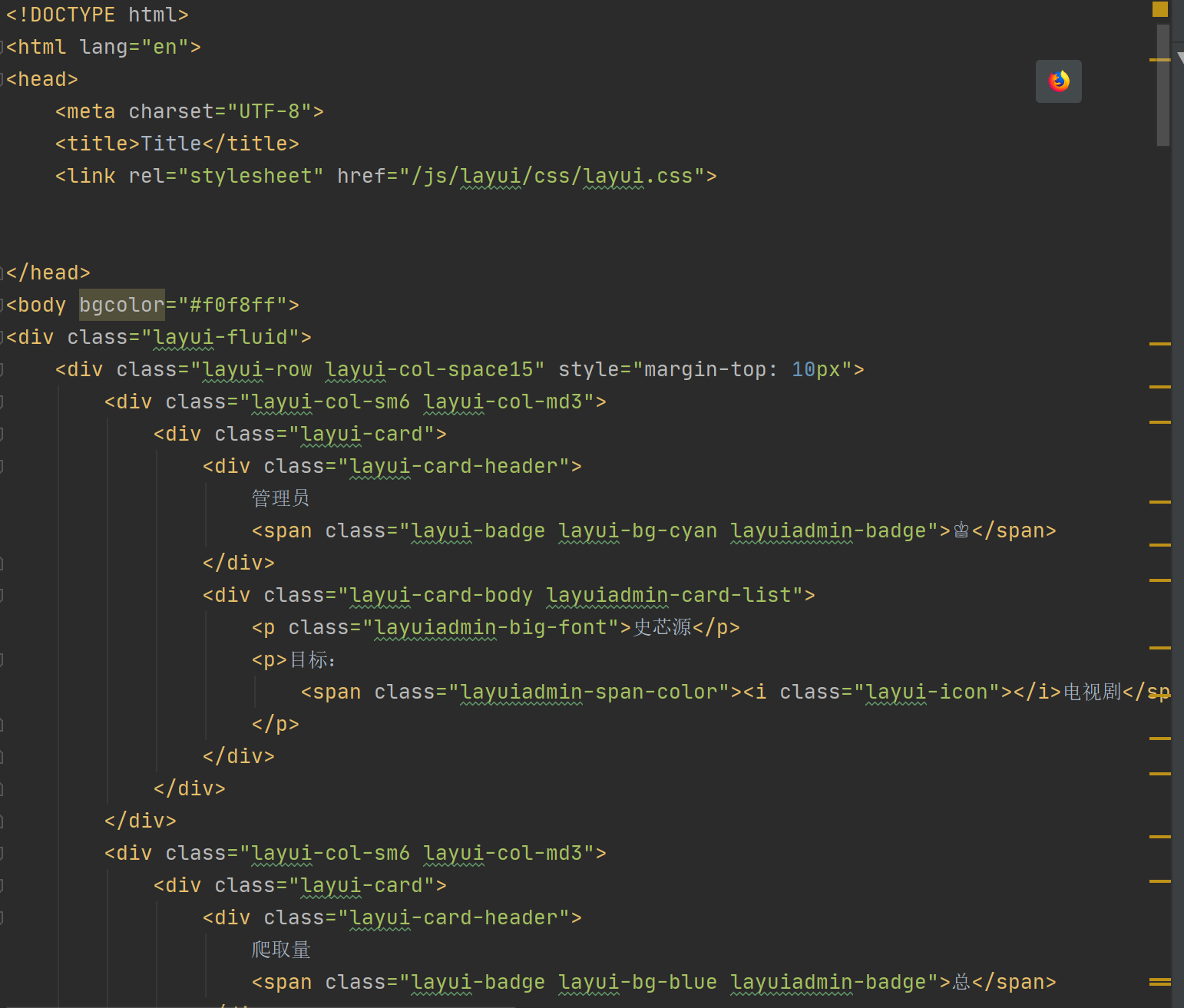
html层：

使用了layui框架整个前端由它支撑。包括一些动态效果。

我负责的部分是电视剧部分，主要有数据查询和数据分析两部分组成也有两部分代码：



这块就是两个检索条件的框子其他内容省略没截取

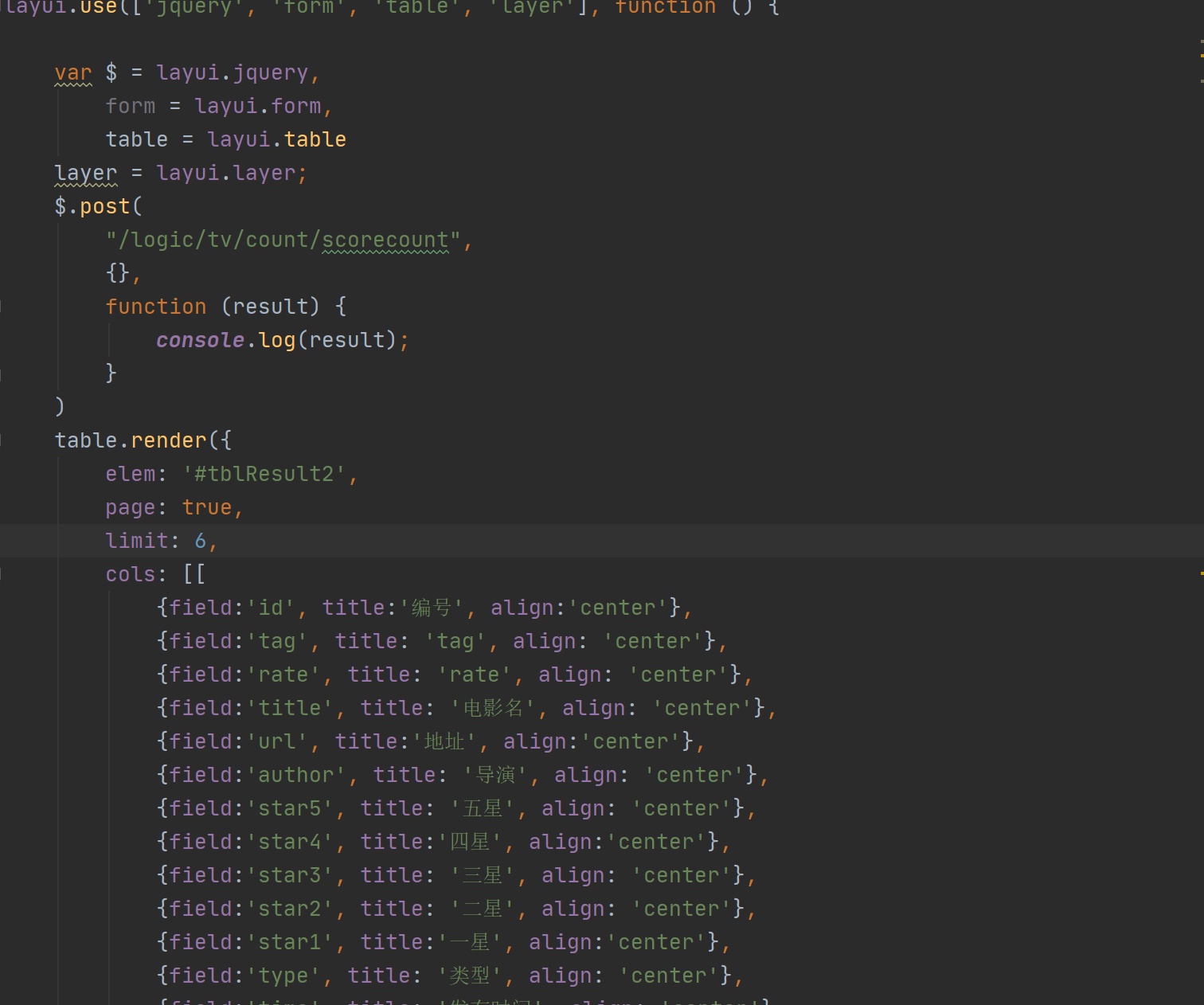


这一块就是数据分析的头部，用来描述爬取两，负责人。

Js层：

也分为两部分对应前边的两部分html

首先就是数据查询js：



这块内容就是对table，也就是查询表的一个表头处理。对应的数据以及上面的显示内容

第二部分分析js：

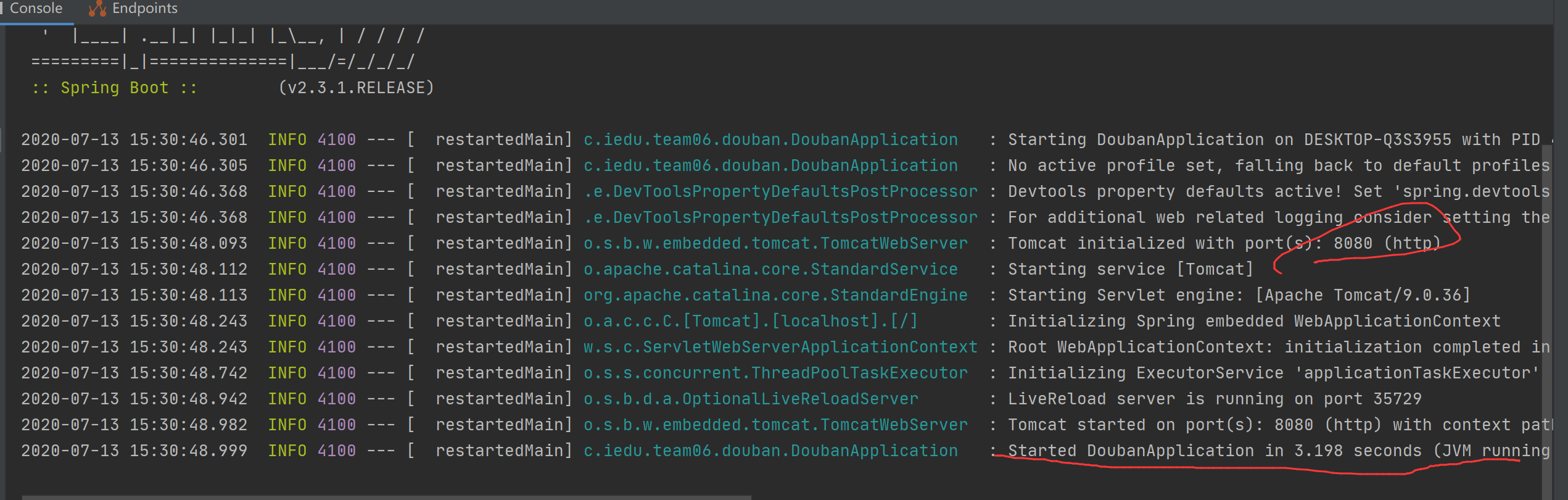


这块内容比较多，因为echarts的option内容比较多，就不一一截图，首先就是对echarts的定义，并且说明连接的div是哪个需要对div进行id定义。

然后就是post函数通过前面的controller的映射亲求，将dao层搜找到的数据发送到前端。共给echarts图标使用。

执行结果：

成功截图:

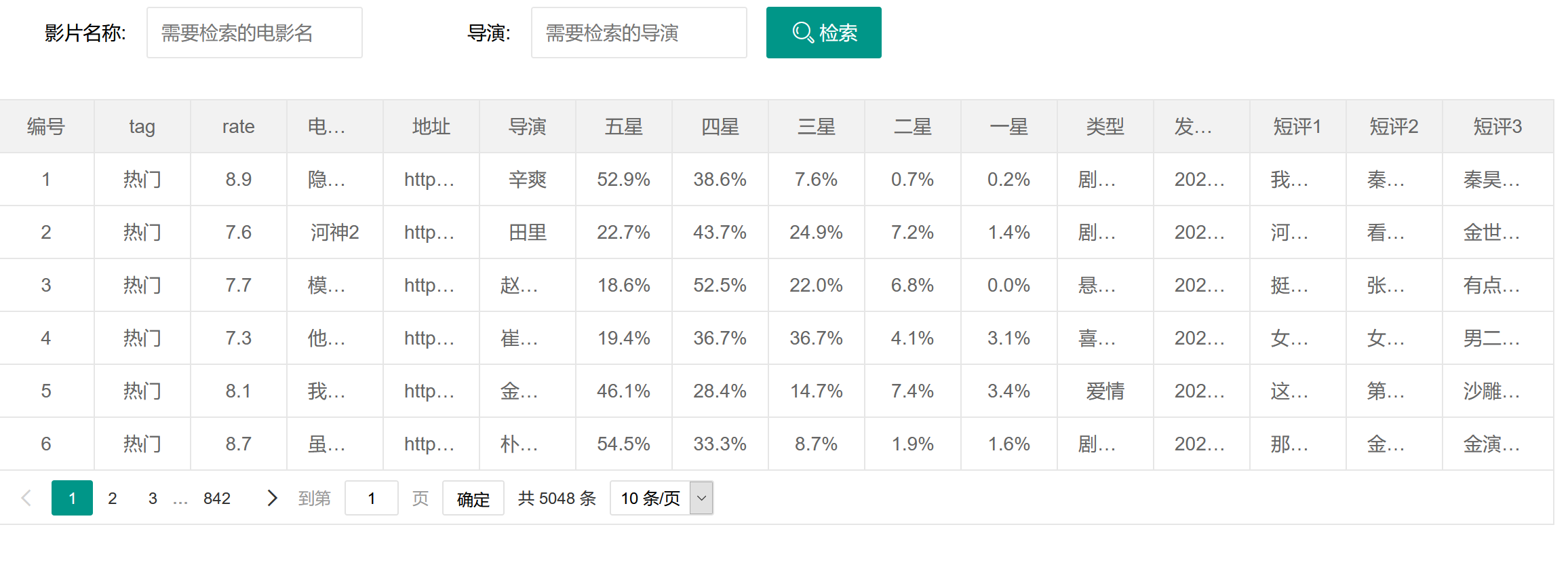


前端数据调用成功截图：

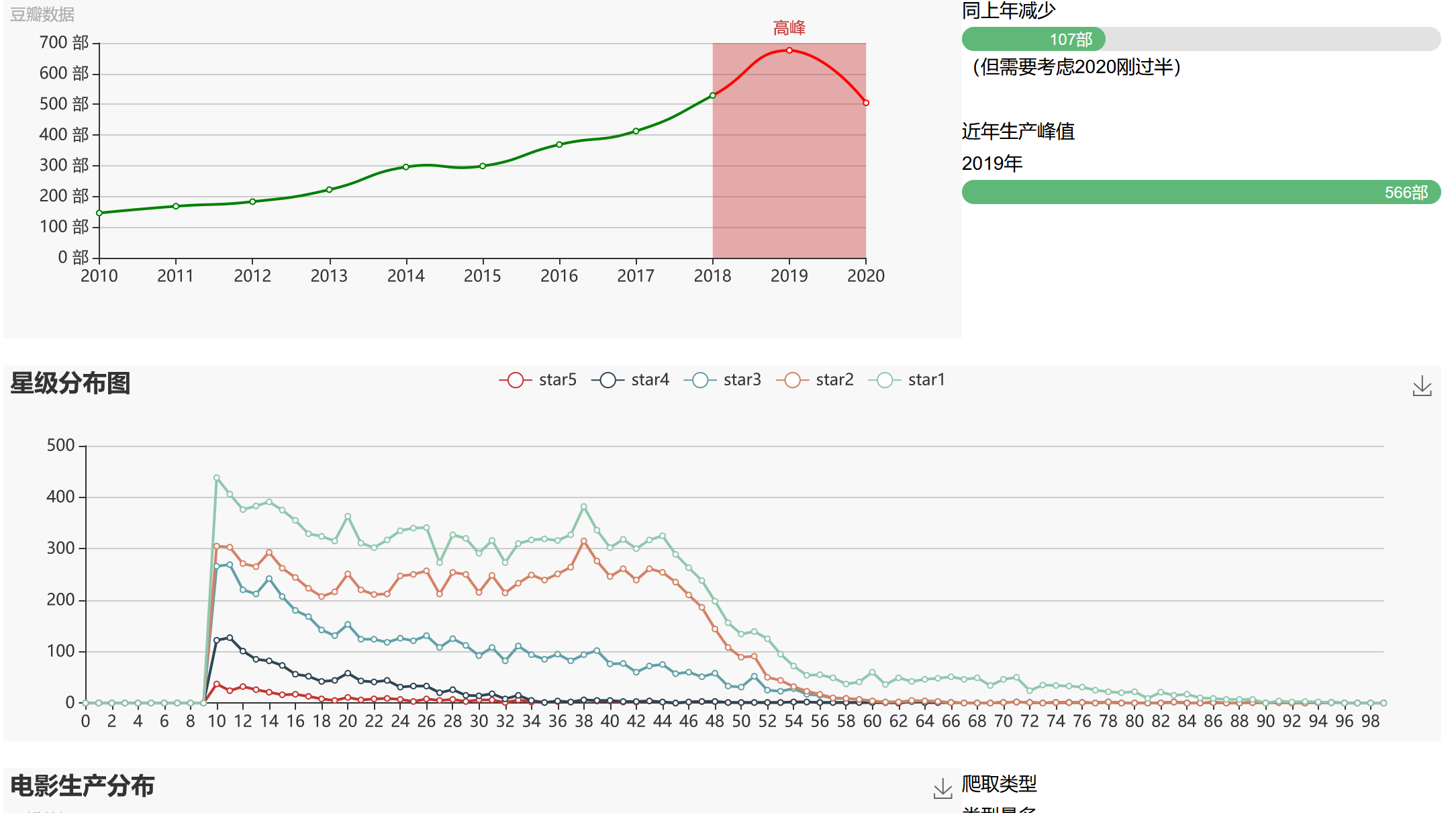


这里的0，0数组适用于后面图表显示的，有部分数值不需要归0了.

搜寻界面：



数据分析（这里只进行的部分截图）



**5 系统测试**（小三黑体，居中）

详细写明测试方法、测试描述（bug与修正方法等）、预期结果等；

在通过js文件调用数据库数据时总是出错，首先就是通过网页的F12查看报错信息，然后根据报错信息，寻找到错误的代码行，同时idea下面也会给出相应信息，结合处理。如果还没有解决就会打断点，通过bug模式寻找错误的地方。