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: 高数-您的高级数据信息服务平台

　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Z1802班

姓名：廖嘉伟

　实训起止日期：2020.6.15-2020.7.15

## 1 引言

#### 1.1项目概述

###### 1.1.1项目目的

大数据时代，信息海量且源源不绝，面对庞大的数据，如果运用合适的技术手段，就能提取出有价值的信息，这也就是本项目的目的。

本项目正是基于此目的，先使用基于Python的**Scrapy网络爬虫框架**爬取特定网站的信息数据，包括**豆瓣电影、音乐、图书以及当当图书、B站动漫评论**，并将数据保存到远端的**MariaDB数据库**中，然后使用前后端分离的开发模式，后端使用**SpringBoot**开发，前端则采用基于NodeJs的**Vue框架**，并通过**Mybatis**实现数据查询及数据分析功能，并通过**eCharts**将数据的分析结果展示到前端，最终完成数据、前端与后端相连接，形成完整的项目，使信息可以更方便轻松地被用户提取运用。

整个过程中，小组团队运用了代码托管平台Gitee以及代码版本控制器git。，、Python、Java、Ajax、JSON等在现代项目开发条环境下的主要技术。

###### 1.1.2项目背景

当下正处于信息爆炸的时代，互联网几乎是人们工作、生活、学习都无法脱离的平台，每天都会产生海量数据，音乐、电影、书籍、评论等种种数据种类繁杂且数量庞大因此现在也被称为大数据时代。几乎可以说，谁掌握了数据，谁就拥有了先机。

但我们也知道，数据越庞大，我们处理起来也会越棘手，只有通过合适的技术手段才能更有效地提取出更有价值的信息。于是我们采用了一系列的技术手段，加以正确的组合连接，设计了本项目。

###### 1.1.3项目意义

拿项目中豆瓣图书一块来举例，在数据查询与信息展示功能中，用户可以自定义地输入自己想要查询的条件，比如书名、作者名的关键词查询，也可以加入多个书籍类型以筛选自己想要的书籍类型，此外还可以通过筛选豆瓣的评分人数以及豆瓣的评分区间来找到当下的热书、好书，更方便用户找到想要的书。

而在数据分析功能中，用户则可以看到在庞大的数据下，书籍信息有着什么样的规律，比如什么类型的书更多，说明了该类型的书的需求更大；亦或者什么类型的书平均分高，而什么样的书平均评分低，可以得到什么类型的书更受欢迎的信息；通过分析书籍的出版时间，则可以了解人们从什么时候开始有读书热潮，书籍出版数量井喷式增长。这些分析信息都可以为用户的各类活动带来参考价值。

###### 1.1.4参考资料

企业授课内容：

Python3，idea2019，git等工具包；

python基础及相关ppt参考资料；

scrapy框架及讲解录屏；

spring boot框架及讲解录屏；

mybatis框架（数据接口）及讲解录屏；

layui基础及讲解录屏；

echarts讲解录屏

提供demo：

豆瓣电影top250scrapy框架python码参考；

数据接口java码参考；

Layui和echarts码参考；

#### 1.2 需求分析

###### 1.2.1功能需求

随着信息技术的发展，各类娱乐、学习等网站层出不穷，豆瓣、哔哩哔哩、当当等各类网站呈现给人们纷繁的数据信息。为了给用户呈现直观的网站数据，本项目旨在通过建立用户管理系统，控制相关网站爬取数据的走向（数据量>10000），并将分析图表展示给用户，同时建立搜索查找功能，便于用户搜索、查找想要的数据信息。

###### 1.2.2性能需求

**数据分析精确度：**考虑到本项目，因此对于数据可视化的部分精确度无较高要求，只需能体现相对占比等数据特征即可。而对于回头客的判断则需要较高的预测准确率，在这里可以拿2019年的数据进行检验来判断预测。

**时间特性：**

响应时间：图片显示时间在0~500ms以内。

更新处理时间：本项目的数据在短期内数据分析可靠度较高，无需实时获取，但需定期更新。同时考虑到用户体验可考虑增加实时更新的功能。

数据转换与传输时间：数据转换主要是涉及到数据库的问题。为了降低一次显示过多的记录会消耗太多的时间这个问题，我们在设计时将让数据库尽量分页显示。传输时间的问题比较复杂，这和服务器的处理能力，电信部门给我们提供的带宽，银行的电子交易网的响应速度有关。

运行时间：本网站如非特殊原因应保持24小时开通。

**适应性：**这里主要是解决用户使用何种设备/系统进行浏览的问题，该项目对于任何设备均适应性良好。

###### 1.2.3非功能性需求

* 扩展性和可维护性

1） 可扩展性方面，有较强的拓展空间。

① 可考虑将使用平台延伸到终端和手机端。终端安装可为受众群体在公众场合使用提供免费平台；网页手机可以直接打开，适合各场景下、不同人群的观看习惯。

2） 可维护性：

① 维护人员理解网页的结构、接口、功能和内部过程的难度不大，即可理解性强；

② 测试和诊断软件错误的难度不大，即可测试性强

③ 软件需要调整时，修改网页的难度中等，即有较好的可修改性；

④ 网页功能的模块化、信息隐蔽性、高内聚、低耦合等性质，都对于提高网页的可理解性、 可维护性和可修改性有很重要的作用。

* 易用性

直接使用网页链接打开，不需要下载任何插件，轻松简单。

（1）所有内容都以所有人能理解的格式来呈现。

（2）网页里的功能按钮接口组件对任何人来说都可操控。

（3）系统可以满足有经验和无经验的用户，允许用户进行频繁或缓慢的操作。

###### 1.2.4接口需求

硬件接口：可拓展，暂无。

软件接口：错误信息处理接口、登陆注册信息传输接口、数据查询分析传输接口、数据库信息传输接口

###### 1.2.5出错处理需求

错误/异常信息对我们程序的调试 、完善有着重要的指导意义，但是我们不希望用户看到的错误/异常信息，所以我们认为，异常应该在控制/返回消息转发之前尽量处理完毕，同时发生异常的模块记录 log 日志，以便于我们开发者的进一步调试和改进。

**数据层**：数据层部分的异常信息主要是数据类型、数据范围以及约束条件等。该部分的异常信息在一定程度上需要返回给接口调用者，以便于指导用户的数据修改和完善，我们在捕获异常信息之后会对异常类型进行标注处理，做为指导性更正建议。

###### 1.2.6其他需求

**可使用性：**

用户操作简单，体验好，数据展现直观易懂。

**可移植性：**

无需下载通过浏览器访问网页即可使用，可移植性高。

① **PC 端浏览器**：应用可直接在 PC 端浏览器打开。支持浏览器：IE9、Chrome7、360 浏览器、火狐浏览器、UC 浏览器等主流浏览器。

②**移动端浏览器**：应用可直接在移动端端浏览器打开。支持浏览器：safari 浏览器、无线 Mozilla、Chrome、百度浏览器、小米浏览器、UC 浏览器等主流浏览器。

#### 1.3 运行环境

###### 1.3.1硬件环境

1.笔记本1台

2.内存：4GB以上。

4.局域网畅通

###### 1.3.2软件环境

1. 操作系统：Windows 10。

2. 文档编写工具：Microsoft office 2016（word excel powerpoint visio）。

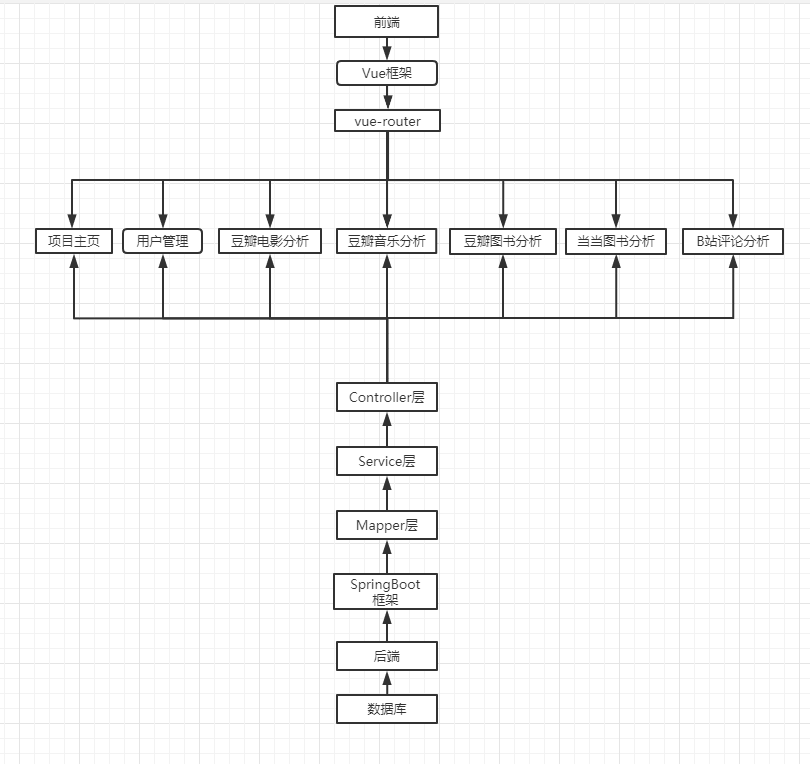
3. 开发工具：IDEA2019，PyCharm2019，MySQL5.7.x，WebStorm2019、Navicat、ECharts。

4. 版本控制工具：Git/码云。

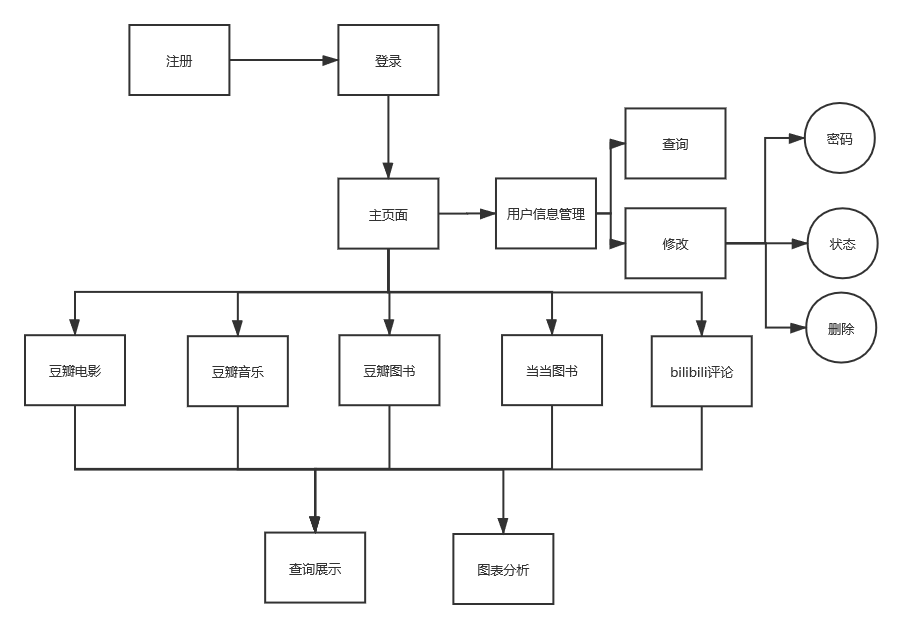
## 2 项目设计

#### 2.1 设计思路

###### 2.1.1总体设计思路



###### 2.1.2总体程序流程图



#### 2.2 模块功能介绍

###### 2.2.1豆瓣电影数据爬取

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。

###### 2.2.2豆瓣电影数据查询

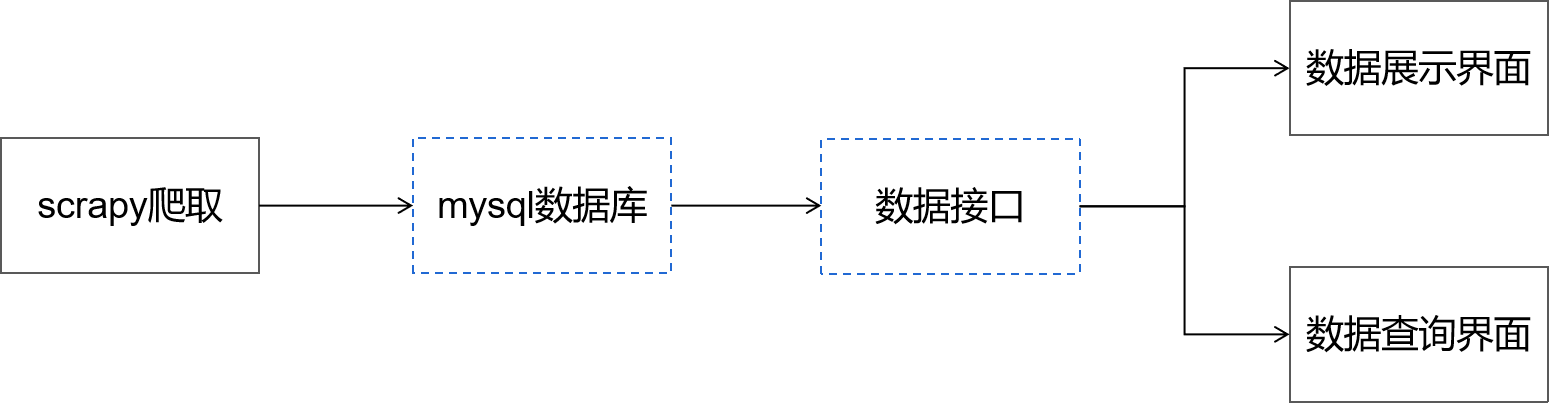
1. 针对豆瓣影片库，爬取电影相关数据字段，同时选取电影名、评分、主演、导演、类型等关键字段展示给用户
2. 允许平台用户在豆瓣影片查询界面通过查询关键字段：电影名、导演、主演根据需求检索相关电影数据。

###### 2.2.3豆瓣电影数据展示

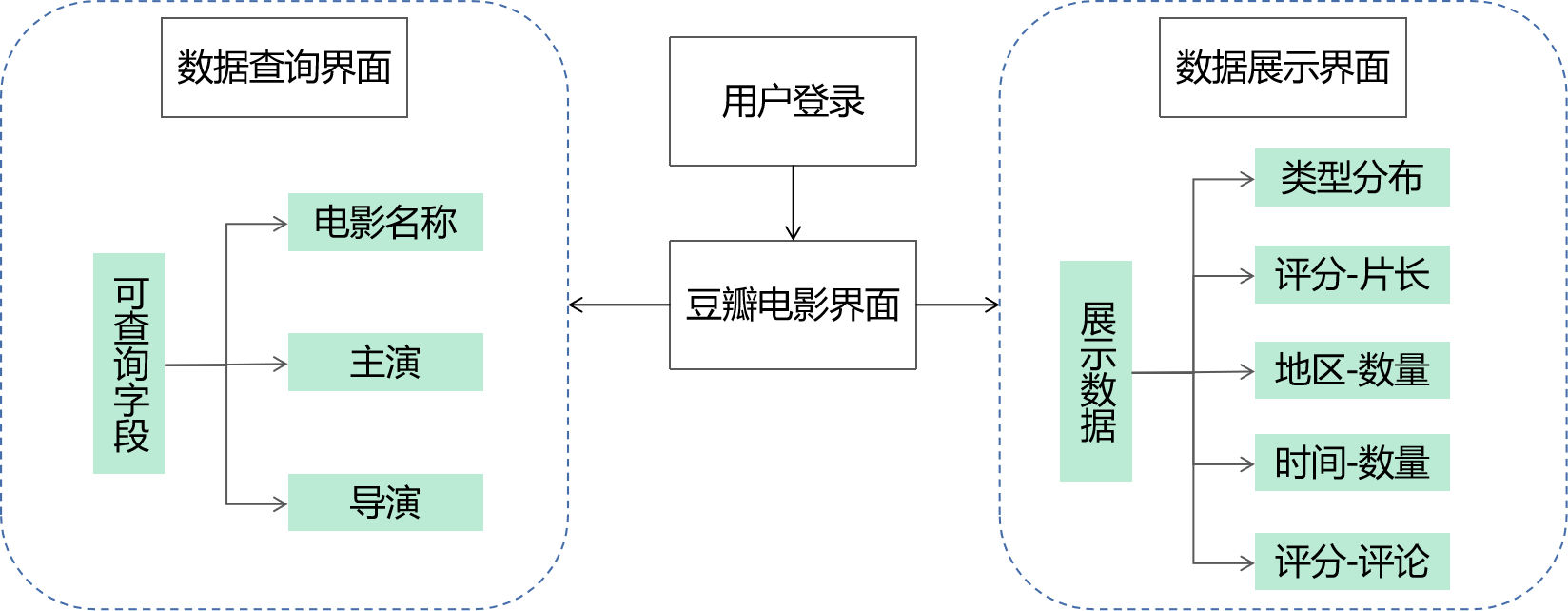
根据爬取来的豆瓣影片数据，主要做如下数据展示：

1. 数据信息点：电影总数、总评论数、平均片长、最高评分、最低评分
2. 各电影类型占比情况-饼图
3. 不同地区产电影数量的比较-柱状图
4. 随时间产生电影数量的变化曲线-折线图
5. 平常与评分的关系-散点图
6. 评论人数与评分的关系-折线图

#### 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。

**3 详细设计**

#### 3.1系统设计

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；

#### 3.2数据库设计

###### 3.2.1逻辑模型

影片数据表（bdcy\_douban\_mivie）：电影名、评分、评论人数、1-5星占比、短评数量、影评数量、电影类型、导演、编剧、主演、国家/地区、语言、上映时间、片长

###### 3.2.2物理模型

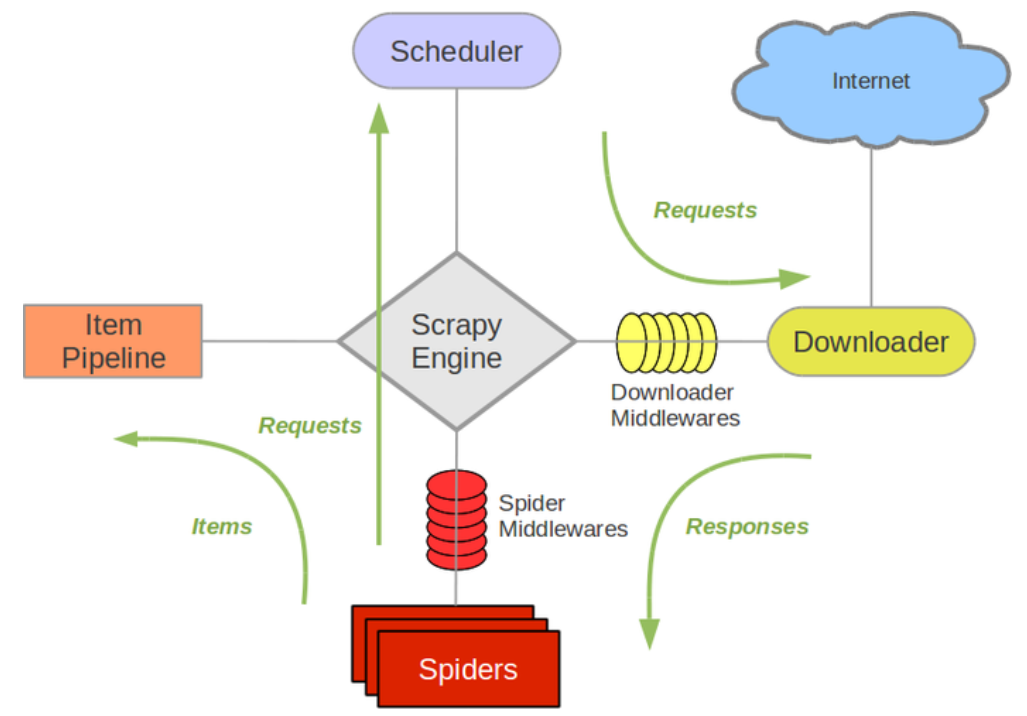
豆瓣电影数据库：

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 字段含义 | 类型 | 长度 | 允许为空 | 主键 | 说明 |
| 1 | id | 电影序列 | int | 11 | 否 | 是 |  |
| 2 | movie\_name | 电影名 | varchar | 255 | 否 | 否 |  |
| 3 | score | 电影评分 | float | 255 | 是 | 否 |  |
| 4 | people\_count | 评论人数 | int | 11 | 是 | 否 |  |
| 5 | five\_star | 五星占比 | float | 255 | 是 | 否 |  |
| 6 | four\_star | 四星占比 | float | 255 | 是 | 否 |  |
| 7 | three\_star | 三星占比 | float | 255 | 是 | 否 |  |
| 8 | two\_star | 两星占比 | float | 255 | 是 | 否 |  |
| 9 | one\_star | 一星占比 | float | 255 | 是 | 否 |  |
| 10 | short\_review\_count | 短评数量 | int | 11 | 是 | 否 |  |
| 11 | review\_count | 评论数量 | int | 11 | 是 | 否 |  |
| 12 | movie\_type | 电影类型 | varchar | 255 | 是 | 否 |  |
| 13 | move\_director | 导演 | varchar | 255 | 是 | 否 |  |
| 14 | movie\_script | 编剧 | varchar | 255 | 是 | 否 |  |
| 15 | movie\_role | 主演 | varchar | 255 | 是 | 否 |  |
| 16 | movie\_country | 地区 | varchar | 255 | 是 | 否 |  |
| 17 | movie\_language | 语言 | varchar | 255 | 是 | 否 |  |
| 18 | movie\_start\_time | 上映时间 | varchar | 255 | 是 | 否 |  |
| 19 | movie\_length | 片长 | int | 11 | 是 | 否 |  |
| 20 | movie\_url | 链接 | varchar | 255 | 是 | 否 |  |

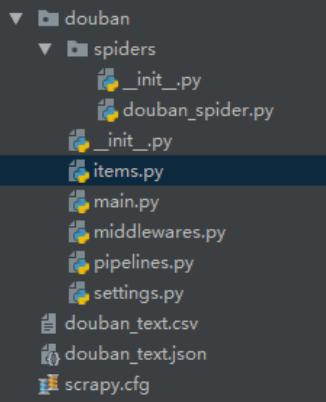
**4 实现方案**

#### 4.1爬取豆瓣电影数据功能实现

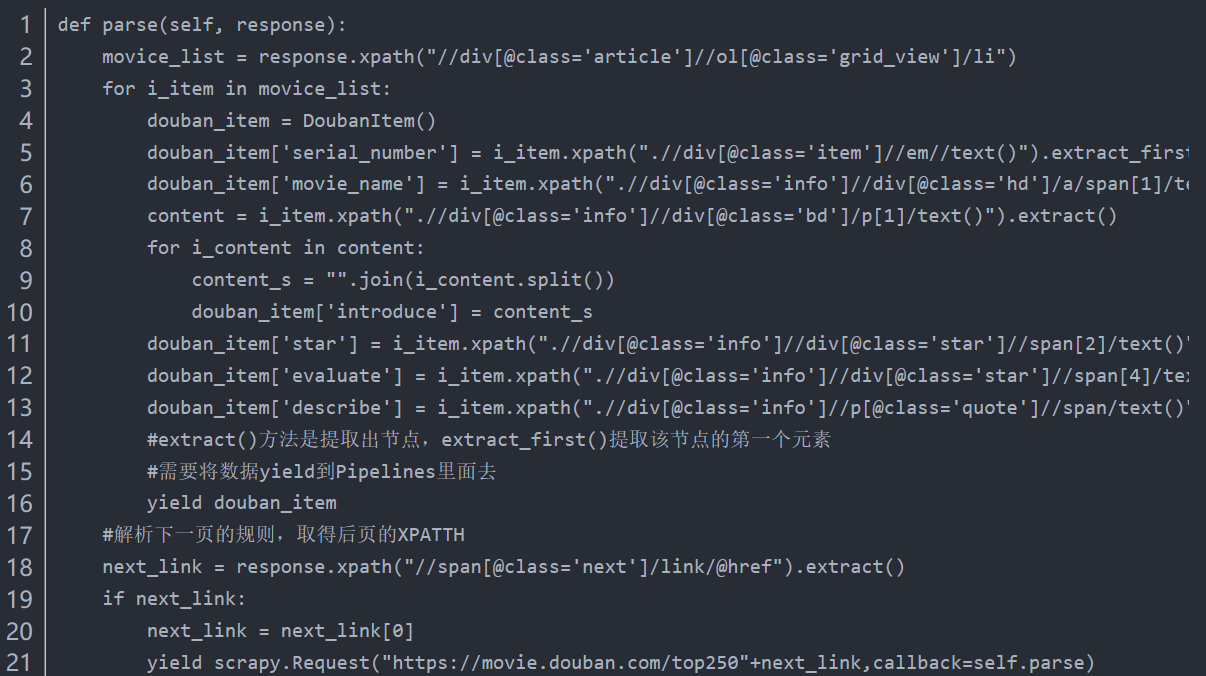
###### 4.1.1 scrapy爬取框架



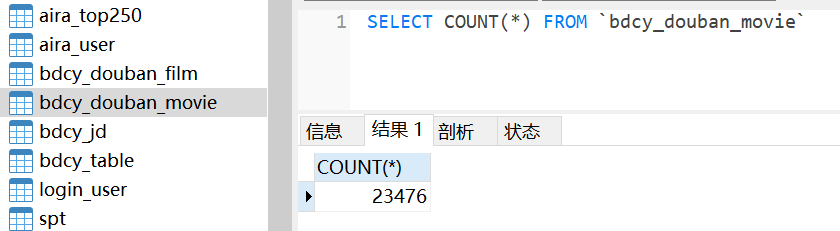
###### 4.1.2 项目部署目录结构



###### 4.1.3 核心代码展示



###### 4.1.4 成果展示



#### 4.2后端数据接口功能实现

###### 4.2.1 项目部署目录结构

###### 4.2.2 核心代码

###### 4.2.3 成果展示

#### 4.3前端界面功能实现

###### 4.3.1 项目部署目录结构

###### 4.3.2 数据展示界面-核心代码

###### 4.3.3 数据查询界面-核心代码

详细写明项目功能实现描述、核心代码及实现结果等；

并对调试好的程序，从执行程序弹出界面开始，每一步操作截一个图，并附加说明（图片格式为JPG格式）。

**5 系统测试**

详细写明测试方法、测试描述（bug与修正方法等）、预期结果等；