目录

[第一章 引言 2](#_Toc519456887)

[基本需求 2](#_Toc519456888)

[目的和功能 3](#_Toc519456889)

[目的 3](#_Toc519456890)

[功能 3](#_Toc519456891)

[第二章 概要设计 4](#_Toc519456892)

[使用的平台和技术 4](#_Toc519456893)

[环境的搭建过程 4](#_Toc519456894)

[第三方的框架 7](#_Toc519456895)

[httpcomponent 7](#_Toc519456896)

[Jsoup 8](#_Toc519456897)

[数据库设计 8](#_Toc519456898)

[功能和设计思路 9](#_Toc519456899)

[设计思路 9](#_Toc519456900)

[未能实现的功能 9](#_Toc519456901)

[实现的功能 9](#_Toc519456902)

[第三章 详细设计 10](#_Toc519456903)

[数据库设计 10](#_Toc519456904)

[类的描述 13](#_Toc519456905)

[基于宽度优先搜索的队列 13](#_Toc519456906)

[向数据库中插入信息 14](#_Toc519456907)

[反爬虫设置user-agent 16](#_Toc519456908)

[将一个输入流转化为字符串 16](#_Toc519456909)

[利用Jsoup进行解析 17](#_Toc519456910)

[图片下载 19](#_Toc519456911)

[第四章 测试和演示 22](#_Toc519456912)

[第五章 总结 28](#_Toc519456913)

# 第一章 引言

## 基本需求

该项目是基于Java的httpcomponent和Jsoup技术实现的网络爬虫。利用网络爬虫来获取和下载信息，通过将信息存储到数据库中实现对信息的查询和访问。通过设计一个网络爬虫来下载网页文本信息，分解网页，遍历网络，存储主要信息。

利用httpcomponent通过HTTP协议来访问网络资源，下载网页的文本信息，将文本信息html的字符流转换为字节流存储到Java的String数据类型中。只获取文本信息，并不直接下载图片、视频、音频等。利用Jsoup来解析html的信息，通过分析html的标签、类名、CSS或jQuery样式来提取所需信息。设计表单结构，用Java链接数据库，在MySQL数据库中建立表单，将需要的信息插入到数据库的表单中，如将文本文档、图片的下载地址、下一页的URL链接等插入到表单中，以实现在数据库中查询访问信息。

在访问网络时采用带有偏好的宽度优先搜索，设置避免重复。提取MySQL中的URL链接，有选择性地将链接插入待访问队列中。提取URL网页链接，判断该链接是否已访问，若已访问则插入到哈希表单中，若未访问则将URL提取出队列，爬取该链接进行上述操作并获取所需信息插入到MySQL数据库中。

## 目的和功能

### 目的

该项目的主要目的是：通过实现网络爬虫来获取需要的信息，将抓取的信息存储到数据库中，在之后的项目实践中调用所需信息。例如，在深度学习的模型训练中需要大量的图片训练数据才能得到一个准确可靠的深度学习模型，通过提取并下载MySQL数据库中的图片链接就可以获得模型训练的训练数据集、校验数据集等内容。

### 功能

已知豆瓣用户每天都在对“看过”的电影进行“很差”到“力荐”的评价，豆瓣根据每部影片看过的人数以及该影片所得的评价等综合数据，通过算法分析产生豆瓣电影 Top 250。

而该项目的主要功能是：抓取豆瓣电影和豆瓣读书的评分榜单前250名的信息，如电影名称、电影评分、一句话精彩影评、电影的海报链接和电影详细页面信息等内容，并通过将信息存储到MySQL数据库中实现信息的查询和提取。例如通过查询数据库中电影海报或图书封面的链接下载图片，同时查询电影或图书的名称来赋予图片相匹配的名称，实现将图片存储到本地。

# 第二章 概要设计

# 使用的平台和技术

开发平台：Windows10

集成开发环境（IDE）：Jetbrain IDEA

编程语言：Java-jdk-9

数据库：mysql-5.6.40-winx64

开发工具：Notepad++，Typora，Mozilla Firefox

开源技术：mysql-connector-java-5.1.39，jsoup-1.11.3，

httpcomponents-client-4.5.5

## 环境的搭建过程

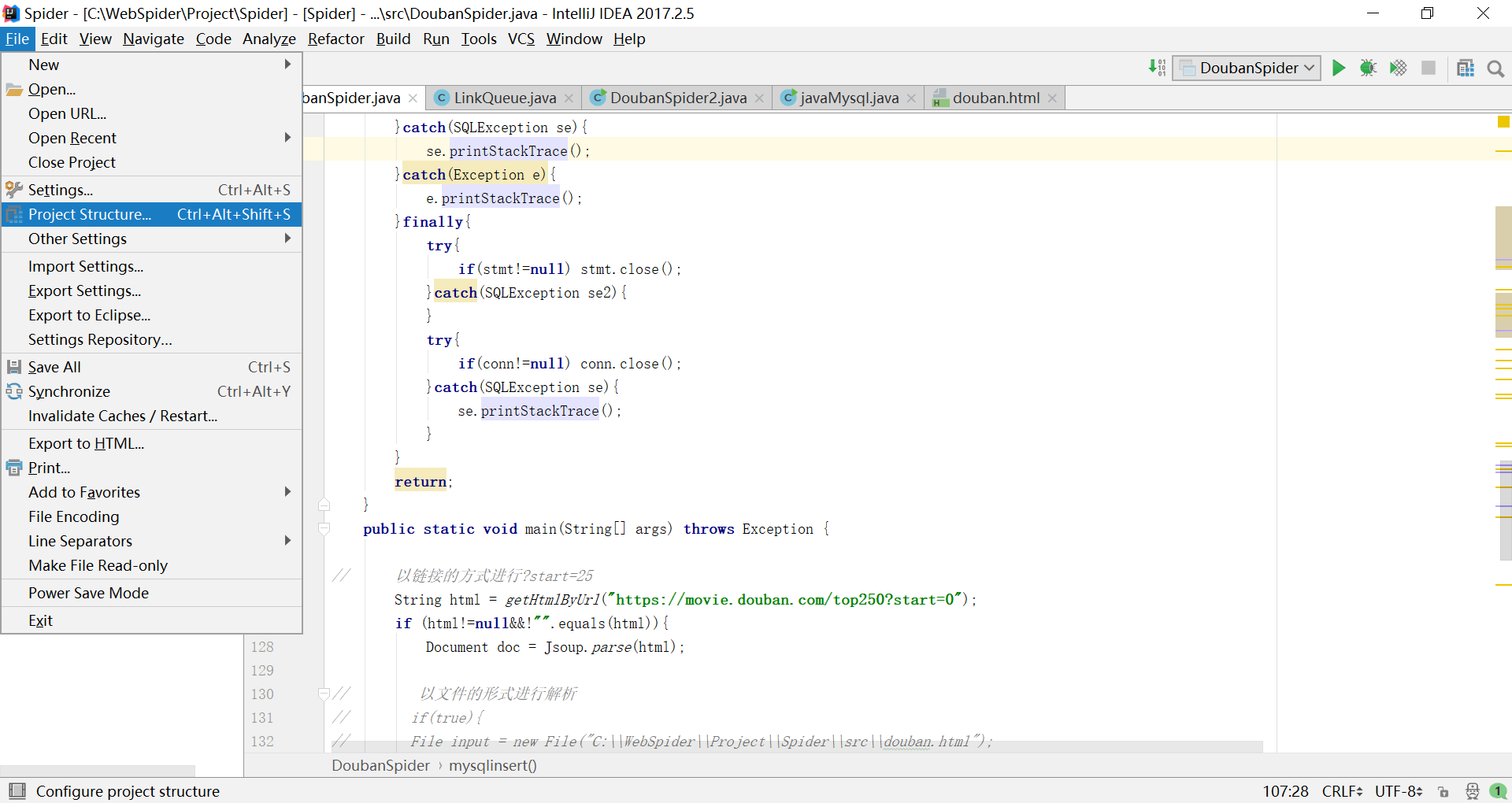
安装Java：首先我们需要下载java开发工具包JDK。安装完成后，右击"我的电脑"，点击"属性"，选择"高级系统设置"；在"系统变量"中设置3项属性，JAVA\_HOME,PATH,CLASSPATH,若已存在则点击"编辑"，不存在则点击"新建"。在 Windows10 中，因为系统的限制，path 变量只可以使用 JDK 的绝对路径。%JAVA\_HOME% 会无法识别，导致配置失败。如果使用1.5以上版本的JDK，不用设置CLASSPATH环境变量，也可以正常编译和运行Java程序。

在Jetbrain官网下载最新版本的Jetbrain IDEA，找到Java-9所在路径，配置Java解释器，运行一个输出程序查看能否正常运行。

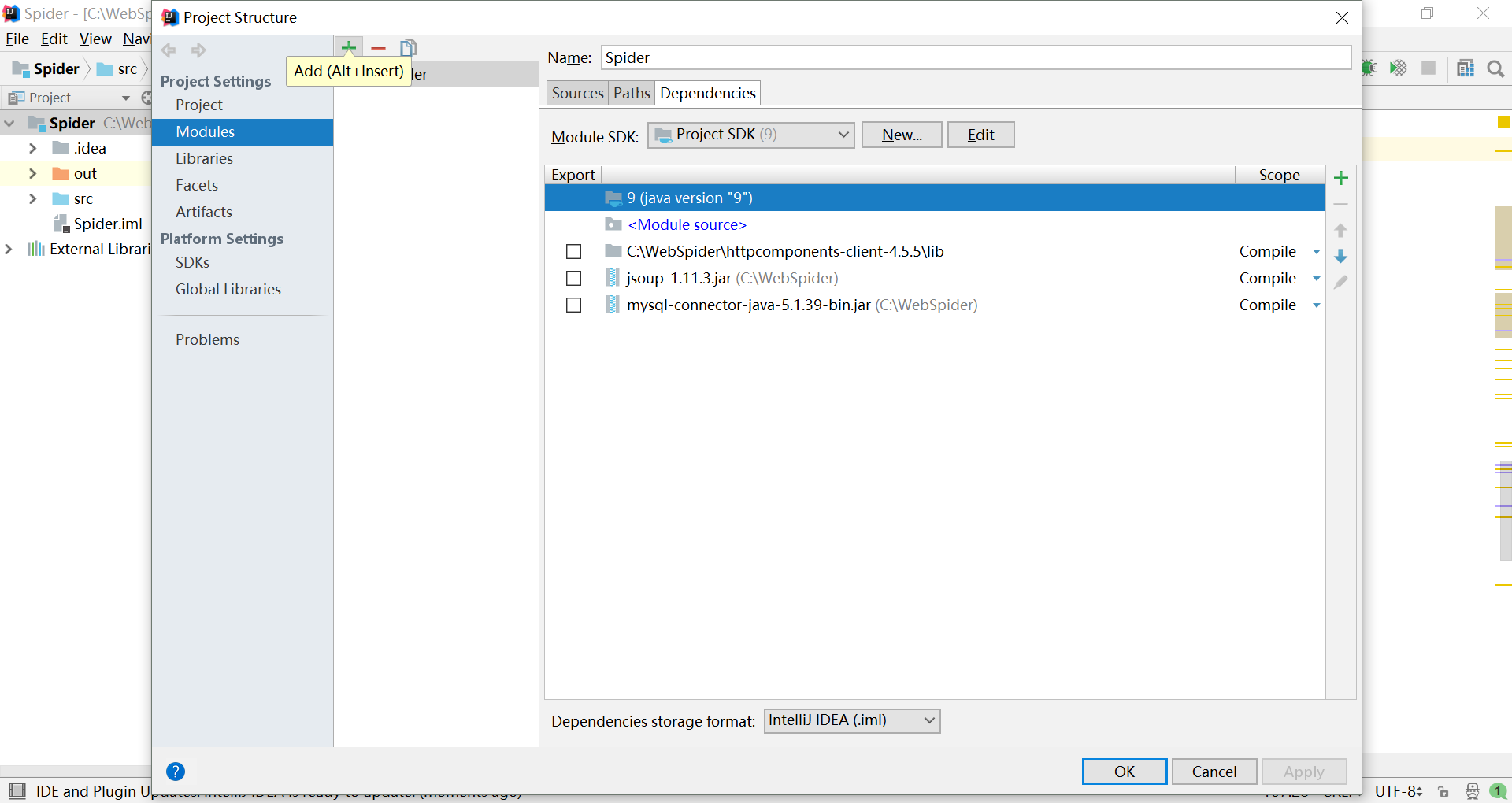
在开源库的官网下载最新版本的库，如下载mysql-connector-java-5.1.39，jsoup-1.11.3，httpcomponents-client-4.5.5等。

在集成开发环境IDEA中导入开源Jar包。

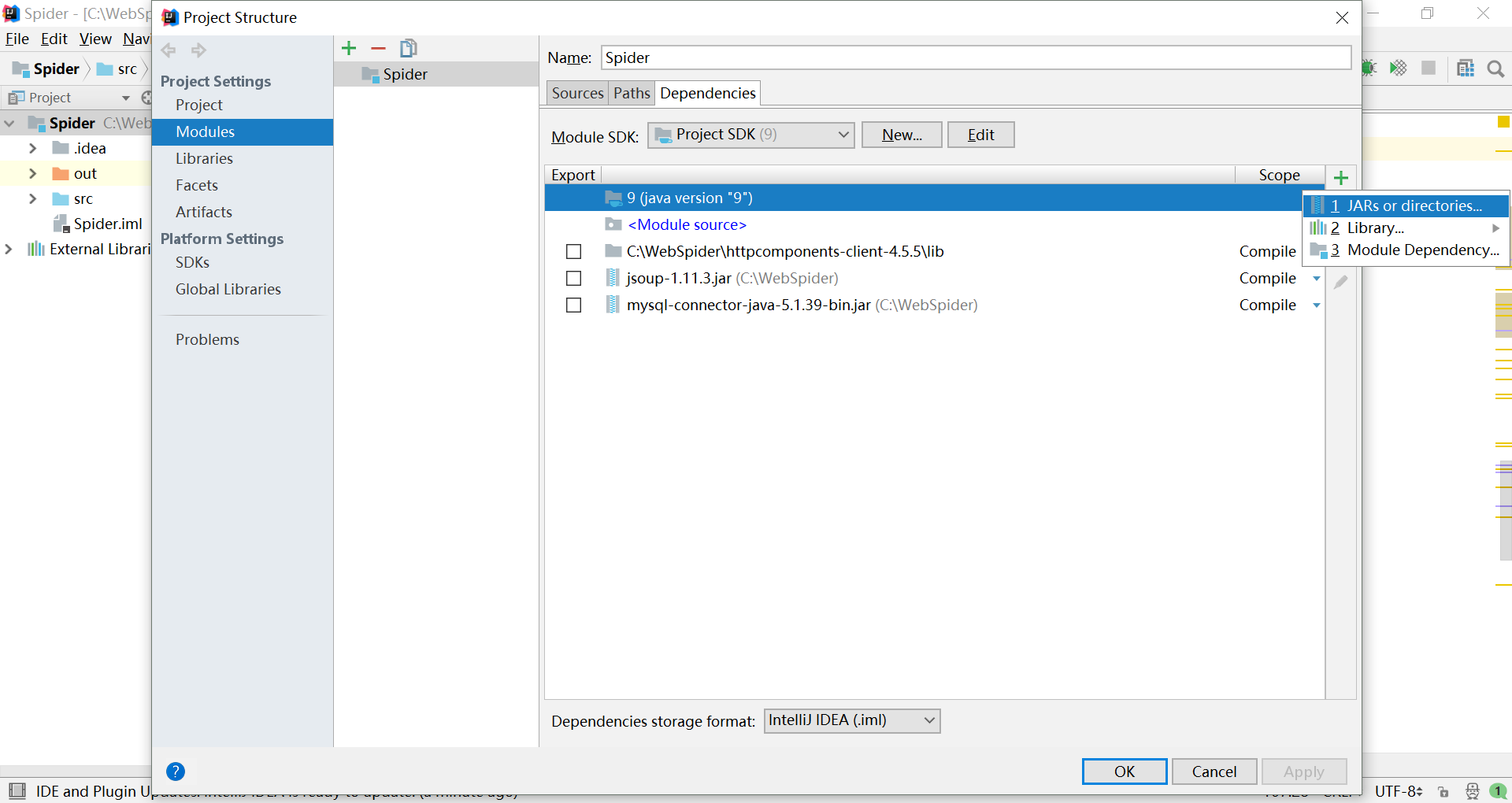
打开File中的Project Structure，查看文件结构



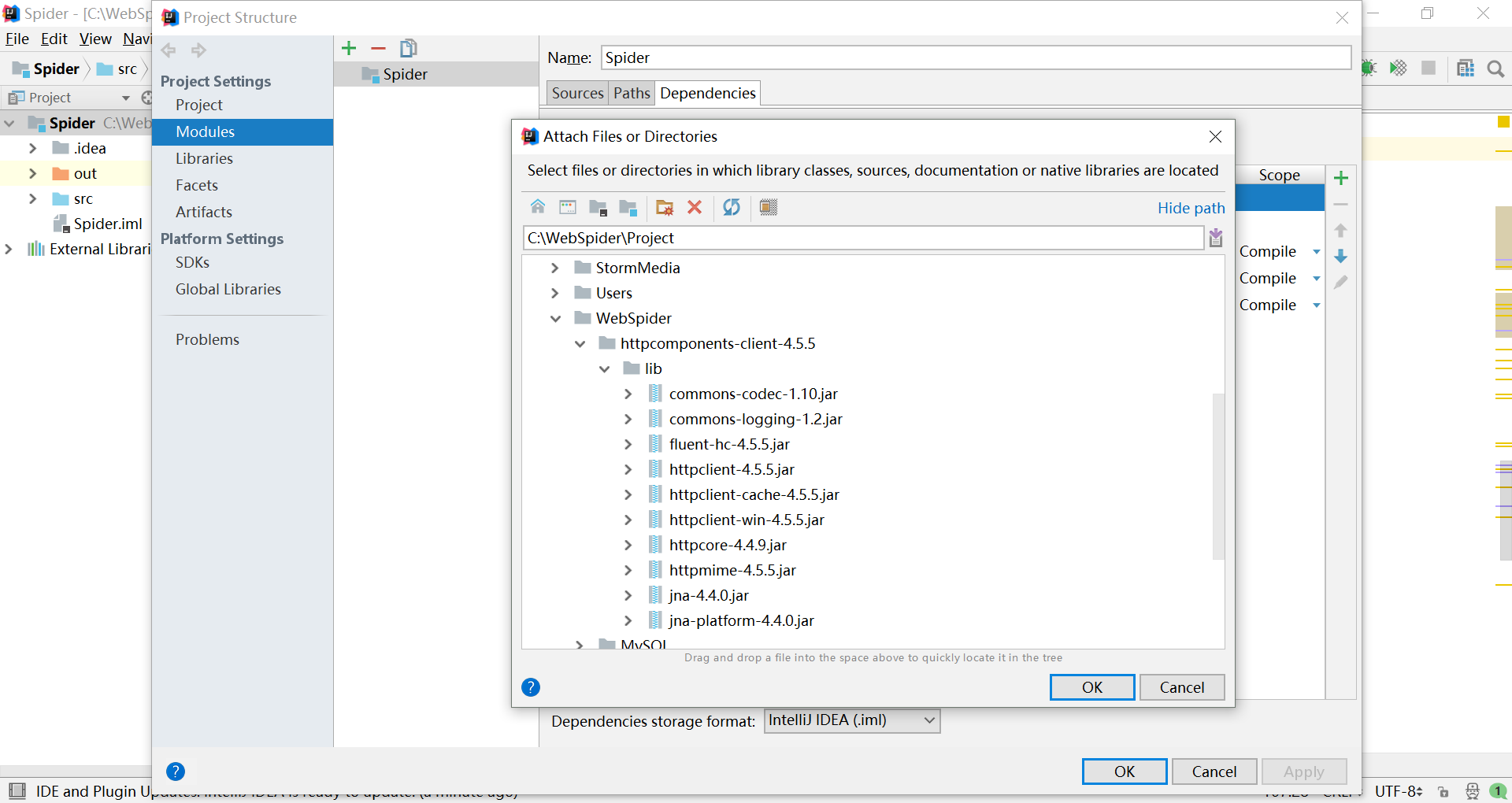
打开Moduels



点击加号添加Jar文件或文件夹



找到文件路径点击OK后完成



## 第三方的框架

第三方框架使用了开源的httpcompoent（即httpclient）和Jsop。

### httpcomponent

httpcompoent 是Apache Jakarta Common 下的子项目，可以用来提供高效的、最新的、功能丰富的支持 HTTP 协议的客户端编程工具包，并且它支持 HTTP 协议最新的版本和建议。使用 HttpClient 需要以下 6 个步骤：

1. 创建 HttpClient 的实例

2. 创建某种连接方法的实例，在这里是GetMethod。在 GetMethod 的构造函数中传入待连接的地址

3. 调用第一步中创建好的实例的 execute 方法来执行第二步中创建好的 method 实例

4. 读 response

5. 释放连接。无论执行方法是否成功，都必须释放连接

6. 对得到后的内容进行处理

### Jsoup

Jsoup 是一款Java 的HTML解析器，可直接解析某个URL地址、HTML文本内容。它提供了一套非常省力的API，可通过DOM，CSS以及类似于jQuery的操作方法来取出和操作数据。使用Jsoup需要以下3个步骤：

1. 从一个URL，文件或字符串中解析HTML

2. 使用DOM或CSS选择器来查找、取出数据

3. 可操作HTML元素、属性、文本

## 数据库设计

在MySQL数据库中创建了数据库doubanMovies和doubanBooks。

将在doubanMovies中建立一个存储详细电影信息的表单，而在doubanBooks中建立一个存储图书详细信息的表单。内容包括但不仅限于：排名、评分、一句话简介、海报（封面）链接、详细页面链接。

## 功能和设计思路

### 设计思路

在日常娱乐中，经常会有看电影这一活动，但看什么电影成为了我们的问题。基于此点，我设想做一个可以爬取电影信息的网络爬虫程序。除此之外，可将爬取的信息做算法推荐。基于用户的观看记录和个人喜好做电影推荐，与用户交互。更进一步可以构造一个集电影鉴赏、推送、评价和分享为一体的电影网站。

### 未能实现的功能

由于算法推荐的内容有较多复杂的算法因子，与机器学习内容相关，超出个人能力范畴。因此这部分内容未能实现，可在日后致力于实现此部分内容。

### 实现的功能

1宽度优先搜索的队列：该项目通过输入一个豆瓣电影评分榜单的URL链接，获取豆瓣电影评分前250部的详细信息。通过解析html判断榜单的下一页，将下一页的链接插入到队列中。通过提取队列中的URL链接获取全部影片的文本信息。当队列为空时，停止URL链接的出队、入队，此时程序终止。

2图片下载：将提取到的信息插入到数据库中，可以进行电影信息的查询与下载。

3与URL建立链接：此部分通过httpcomponent（httpclient）实现一个客户端，能够向URL发送GET、POST等请求，并获取回应Response。

4将字节流转换为字符流：BufferedWriter提供通用的缓冲方式文本读取，readLine读取一个文本行，从字符输入流中读取文本，缓冲各个字符，从而提供字符、数组和行的高效读取。BufferedWriter 由Writer 类扩展而来，提供通用的缓冲方式文本写入， newLine使用平台自己的行分隔符，将文本写入字符输出流，缓冲各个字符，从而提供单个字符、数组和字符串的高效写入。

5解析html文本信息：通过Jsoup解析字符流形式的html文本信息，通过html的标签、类名或者css样式、jQuery来解析文本，获取所需信息。

6链接数据库：通过Java链接数据库，直接向数据库中插入信息，并且可以查询数据库中的信息，并输出这些信息。

7反爬虫：通过设置user-agent的方式，在post或者get的时候添加请求头，可以使用**requests**库来添加，在post，get或者其他方法是加上**headers**参数就可以了。

# 第三章 详细设计

## 数据库设计

在MySQL数据库中创建了数据库doubanMovies和doubanBooks。

在doubanMovies中创建表单Movies，表单设计5列，分别为：id、name、rating、quote、imgurl。分别代表电影排名，电影名称，电影评分，一句话简介和海报链接。在MySQL中创建该表单

CREATE TABLE movies(

id INT NOT NULL AUTO\_INCREMENT,

name VARCHAR(20) NOT NULL,

rating float NOT NULL,

quote varchar(50) NOT NULL,

imgurl VARCHAR(100) NOT NULL,

PRIMARY KEY (id)

)ENGINE=InnoDB DEFAULT CHARSET=utf8;

在doubanBooks中创建表单Books，表单设计6列，分别为：id、name、 rating、quote、imgurl和infourl。分别代表图书排名、图书名称、图书评分、图书一句话简介、图书封面和图书详细信息页面。在MySQL中创建该表单：

CREATE TABLE books(

id INT NOT NULL AUTO\_INCREMENT,

name VARCHAR(20) NOT NULL,

rating float NOT NULL,

quote varchar(50) NOT NULL,

imgurl VARCHAR(100) NOT NULL,

infourl VARCHAR(100) NOT NULL,

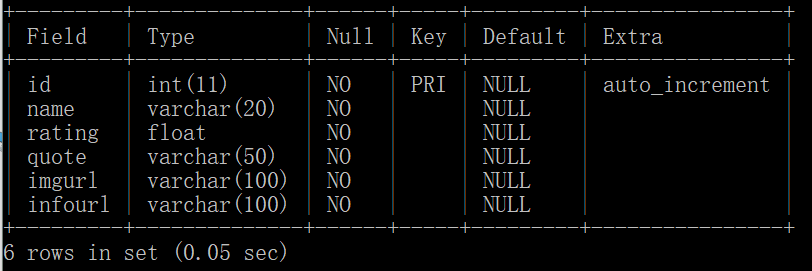
PRIMARY KEY (id)

)ENGINE=InnoDB DEFAULT CHARSET=utf8;

在MySQL中显示表单Movies的结构如下



在MySQL中显示表单Movies的结构如下



## 类的描述

### 基于宽度优先搜索的队列

通过解析html判断榜单的下一页，将下一页的链接插入到队列中。通过提取队列中的URL链接获取全部影片的文本信息。当队列为空时，停止URL链接的出队、入队，此时程序终止。

**public class** LinkQueue {  
 *//已访问的 url 集合* **private static** Set *visitedUrl* = **new** HashSet();  
 *//待访问的 url 集合* **private static** Queue *unVisitedUrl* = **new** Queue();  
 *//获得 URL 队列* **public static** Queue getUnVisitedUrl() {  
 **return** *unVisitedUrl*;  
 }  
 *//添加到访问过的 URL 队列中* **public static void** addVisitedUrl(String url) {  
 *visitedUrl*.add(url);  
 }  
 *//移除访问过的 URL* **public static void** removeVisitedUrl(String url) {  
 *visitedUrl*.remove(url);  
 }  
 *//未访问的 URL 出队列* **public static** Object unVisitedUrlDeQueue() {  
 **return** *unVisitedUrl*.deQueue();  
 }  
 *// 保证每个 URL 只被访问一次  
 //trim删除头尾的空格* **public static void** addUnvisitedUrl(String url) {  
 **if** (url != **null** && !url.trim().equals(**""**)  
 && !*visitedUrl*.contains(url)  
 && !*unVisitedUrl*.contians(url))  
 *unVisitedUrl*.enQueue(url);  
 }  
 *//获得已经访问的 URL 数目* **public static int** getVisitedUrlNum() {  
 **return** *visitedUrl*.size();  
 }  
 *//判断未访问的 URL 队列中是否为空* **public static boolean** unVisitedUrlsEmpty() {  
 **return** *unVisitedUrl*.empty();  
 }  
}

### 向数据库中插入信息

用Java链接数据库doubanmovie，向表movies中插入信息name，rating，quote，imgurl，infourl

**public class** insert{  
 **static final** String ***JDBC\_DRIVER*** = **"com.mysql.jdbc.Driver"**;  
 **static final** String ***DB\_URL*** = **"jdbc:mysql://localhost:3306/doubanmovie"**;  
 **static final** String ***USER*** = **"root"**;  
 **static final** String ***PASS*** = **"123456"**;  
   
 **public static void** main(String[] args){  
 Connection conn = **null**;  
 Statement stmt = **null**;  
 **try**{  
 Class.*forName*(**"com.mysql.jdbc.Driver"**);  
 conn = DriverManager.*getConnection*(***DB\_URL***,***USER***,***PASS***);  
 stmt = conn.createStatement();  
 String sql;  
 sql = **"\"INSERT INTO movies (name, rating, quote, imgurl, infourl) VALUES ('\"+name+\"','\"+rating+\"','\"+quote+\"','\"+imgurl+\"','\"+infourl+\"');\";"**;  
 **int** rowline = stmt.executeUpdate(sql);  
  
 stmt.close();  
 conn.close();  
 }**catch**(SQLException se){  
 se.printStackTrace();  
 }**catch**(Exception e){  
 e.printStackTrace();  
 }**finally**{  
 **try**{  
 **if**(stmt!=**null**) stmt.close();  
 }**catch**(SQLException se2){  
 }  
 **try**{  
 **if**(conn!=**null**) conn.close();  
 }**catch**(SQLException se){  
 se.printStackTrace();  
 }  
 }  
   
 **return**;  
 }  
}

### 反爬虫设置user-agent

通过设置user-agent的方式，在post或者get的时候添加请求头，可以使用requests库来添加，在post，get或者其他方法是加上headers参数就可以了。

CloseableHttpClient httpClient = HttpClients.*createDefault*();  
HttpGet httpget = **new** HttpGet(url);  
httpget.addHeader(**"Accept"**, **"text/html"**);  
httpget.addHeader(**"Accept-Charset"**, **"utf-8"**);  
httpget.addHeader(**"Accept-Encoding"**, **"gzip"**);  
httpget.addHeader(**"Accept-Language"**, **"zh-CN,zh"**);  
httpget.addHeader(**"User-Agent"**,**"Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/49.0.2623.110 Safari/537.36"**);  
RequestConfig config=RequestConfig.*custom*()  
 .setConnectTimeout(10000)  
 .setSocketTimeout(10000)  
 .build();  
httpget.setConfig(config);

### 将一个输入流转化为字符串

**public static** String getStreamString(InputStream inputstream){  
 **if** (inputstream != **null**){  
 **try**{  
 BufferedReader bufferedreader = **new** BufferedReader(**new** InputStreamReader(inputstream));  
 StringBuffer stringbuffer = **new** StringBuffer();  
 String sTempOneLine = **new** String(**""**);  
 **while** ((sTempOneLine = bufferedreader.readLine()) != **null**) {  
 stringbuffer.append(sTempOneLine + **"\n"**);  
 }  
 **return** stringbuffer.toString();  
 }**catch** (Exception ex){  
 ex.printStackTrace();  
 }  
 }  
 **return null**;  
}

### 利用Jsoup进行解析

通过Jsoup解析字符流形式的html文本信息，通过html的标签、类名或者css样式、jQuery来解析文本，获取所需信息

String[] names = **new** String[100];  
 String[] quotes = **new** String[100];  
 String[] imgurls = **new** String[100];  
 **float**[] ratings = **new float**[100];  
 **int** kase = 0;  
  
 kase = 0;  
 Elements titles = doc.getElementsByClass(**"title"**);  
 **for**(Element elem:titles){  
 String title = elem.text();  
 **if**(title.charAt(0) != **'/'**) {  
 names[kase++] = title;  
 }  
 }  
 kase = 0;  
 Elements rating\_num = doc.getElementsByClass(**"rating\_num"**);  
 **for**(Element rating:rating\_num){  
 **float** num = Float.*parseFloat*(rating.text());  
 ratings[kase++] = num;  
 }  
 kase = 0;  
 Elements quos = doc.select(**"p.quote"**);  
 **for**(Element quo:quos){  
 String text = quo.text();  
 quotes[kase++] = text;  
 }  
 kase = 0;  
 Elements links = doc.select(**"[width=100]"**);  
 **for**(Element link:links){  
 String linkHref = link.attr(**"src"**);  
 imgurls[kase++] = linkHref;  
 }  
  
 **for**(**int** i = 0; i < 25; ++ i){  
*// mysqlinsert(names[i],ratings[i],quotes[i],imgurls[i]);* System.***out***.println(i + 1);  
 System.***out***.println(**"片名 "** + names[i] + **" "**);  
 System.***out***.println(**"评分为 "** + ratings[i] + **" "**);  
 System.***out***.println(**"一句话影评 "** + quotes[i] + **" "**);  
 System.***out***.println(**"海报链接 "** + imgurls[i] + **" "**);  
 }

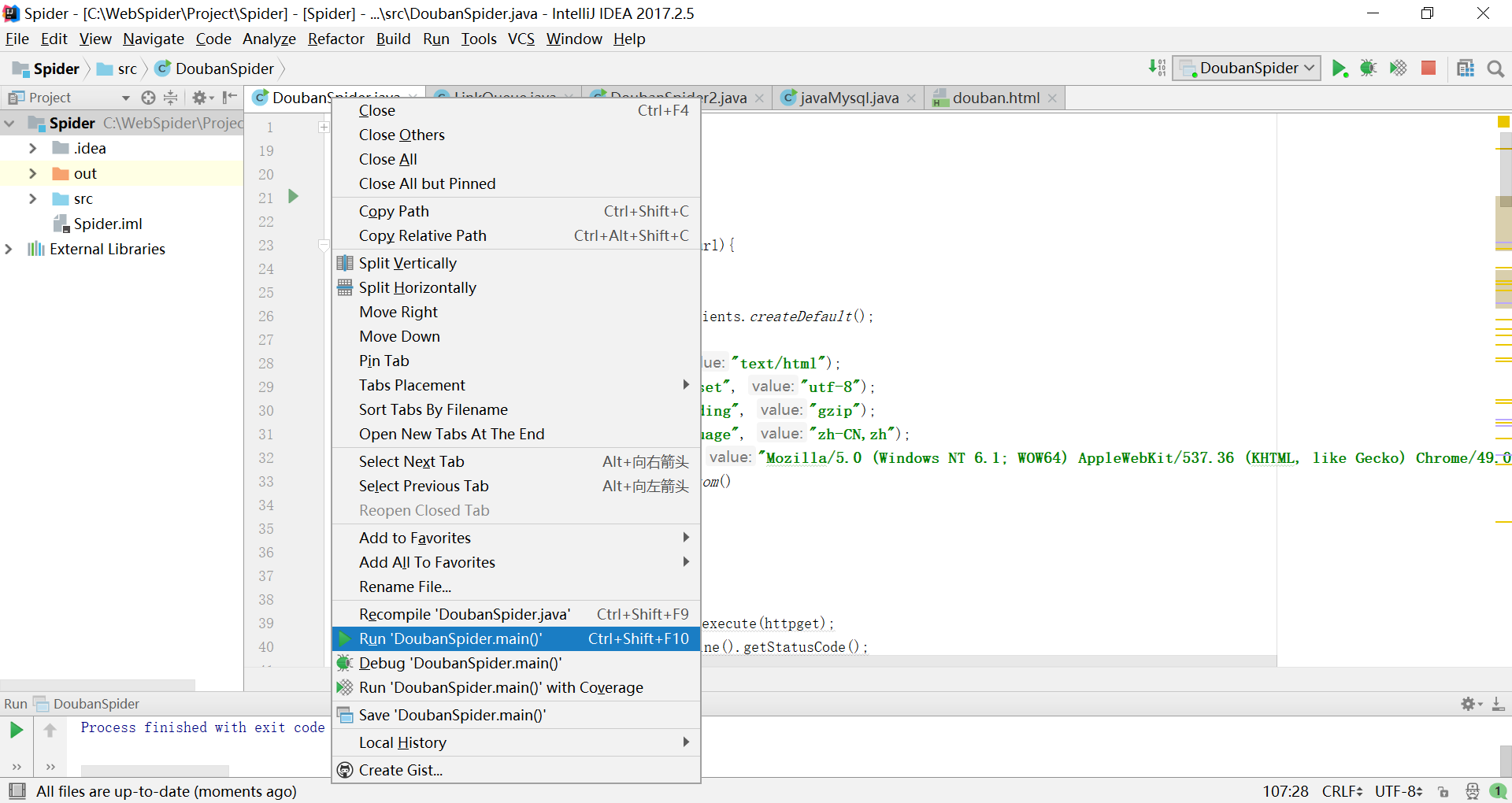
### 图片下载

查询数据库获取图片名称和图片链接，用java中的url下载。以字节流的形式存储到二维数组中。

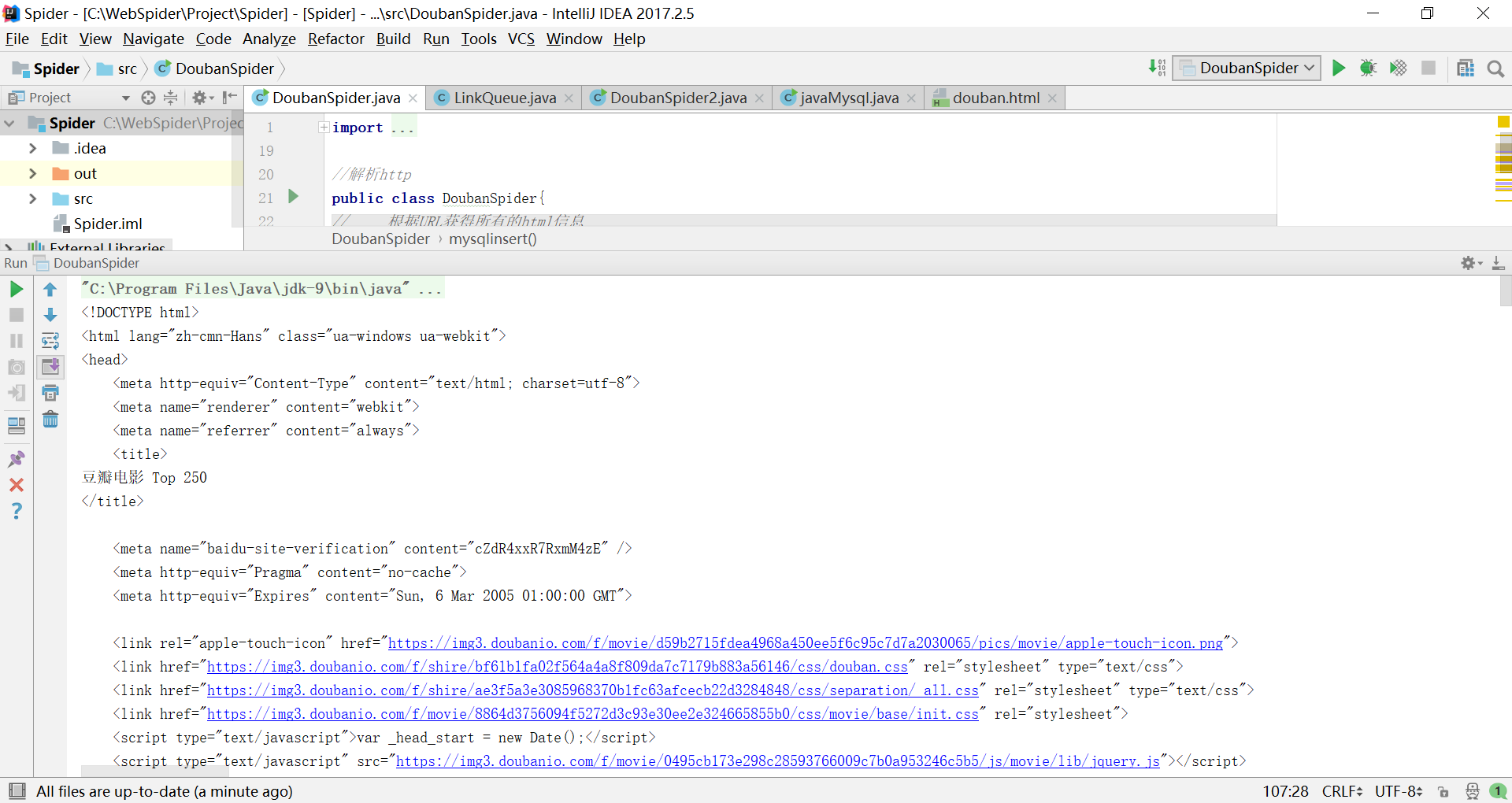
**public class** DownloadImg{  
 **private static void** Download(String urlList,String path) {  
 URL url = **null**;  
 **try** {  
 url = **new** URL(urlList);  
 DataInputStream dataInputStream = **new** DataInputStream(url.openStream());  
 FileOutputStream fileOutputStream = **new** FileOutputStream(**new** File(path));  
 ByteArrayOutputStream output = **new** ByteArrayOutputStream();  
 **byte**[] buffer = **new byte**[1024];  
 **int** length;  
 **while** ((length = dataInputStream.read(buffer)) > 0) {  
 output.write(buffer, 0, length);  
 }  
 fileOutputStream.write(output.toByteArray());  
 dataInputStream.close();  
 fileOutputStream.close();  
 } **catch** (MalformedURLException e) {  
 e.printStackTrace();  
 } **catch** (IOException e) {  
 e.printStackTrace();  
 }  
 }  
 *// JDBC 驱动名及数据库 URL* **static final** String ***JDBC\_DRIVER*** = **"com.mysql.jdbc.Driver"**;  
 **static final** String ***DB\_URL*** = **"jdbc:mysql://localhost:3306/doubanbooks"**;  
 *// 数据库的用户名与密码，需要根据自己的设置* **static final** String ***USER*** = **"root"**;  
 **static final** String ***PASS*** = **"123456"**;  
 **public static void** main(String[] args) {  
 Connection conn = **null**;  
 Statement stmt = **null**;  
 String[] names = **new** String[1000];  
 String[] imgurls = **new** String[1000];  
 **int** kase = 0;  
 **try**{  
 *// 注册 JDBC 驱动* Class.*forName*(**"com.mysql.jdbc.Driver"**);  
 *// 打开链接* System.***out***.println(**"连接数据库..."**);  
 conn = DriverManager.*getConnection*(***DB\_URL***,***USER***,***PASS***);  
 *// 执行查询* System.***out***.println(**"实例化Statement对象..."**);  
 stmt = conn.createStatement();  
 String sql;  
 sql = **"SELECT name, imgurl FROM books"**;  
 ResultSet rs = stmt.executeQuery(sql);  
 *// 展开结果集数据库* **while**(rs.next()){  
 *// 通过字段检索* String name = rs.getString(**"name"**);  
 String imgurl = rs.getString(**"imgurl"**);  
 names[kase] = name;  
 imgurls[kase] = imgurl;  
 kase++;  
 *// 输出数据  
// System.out.println("电影名称: " + name);  
// System.out.println("海报URL: " + imgurl);* String path = **"C:\\Users\\98766\\Desktop\\Books2\\"** + name + **".jpg"**;  
 *Download*(imgurl,path);  
 }  
 *// 完成后关闭* rs.close();  
 stmt.close();  
 conn.close();  
 }**catch**(SQLException se){  
 *// 处理 JDBC 错误* se.printStackTrace();  
 }**catch**(Exception e){  
 *// 处理 Class.forName 错误* e.printStackTrace();  
 }**finally**{  
 *// 关闭资源* **try**{  
 **if**(stmt!=**null**) stmt.close();  
 }**catch**(SQLException se2){  
 }*// 什么都不做* **try**{  
 **if**(conn!=**null**) conn.close();  
 }**catch**(SQLException se){  
 se.printStackTrace();  
 }  
 }  
 System.***out***.println(**"Goodbye!"**);  
 }  
}

# 第四章 测试和演示

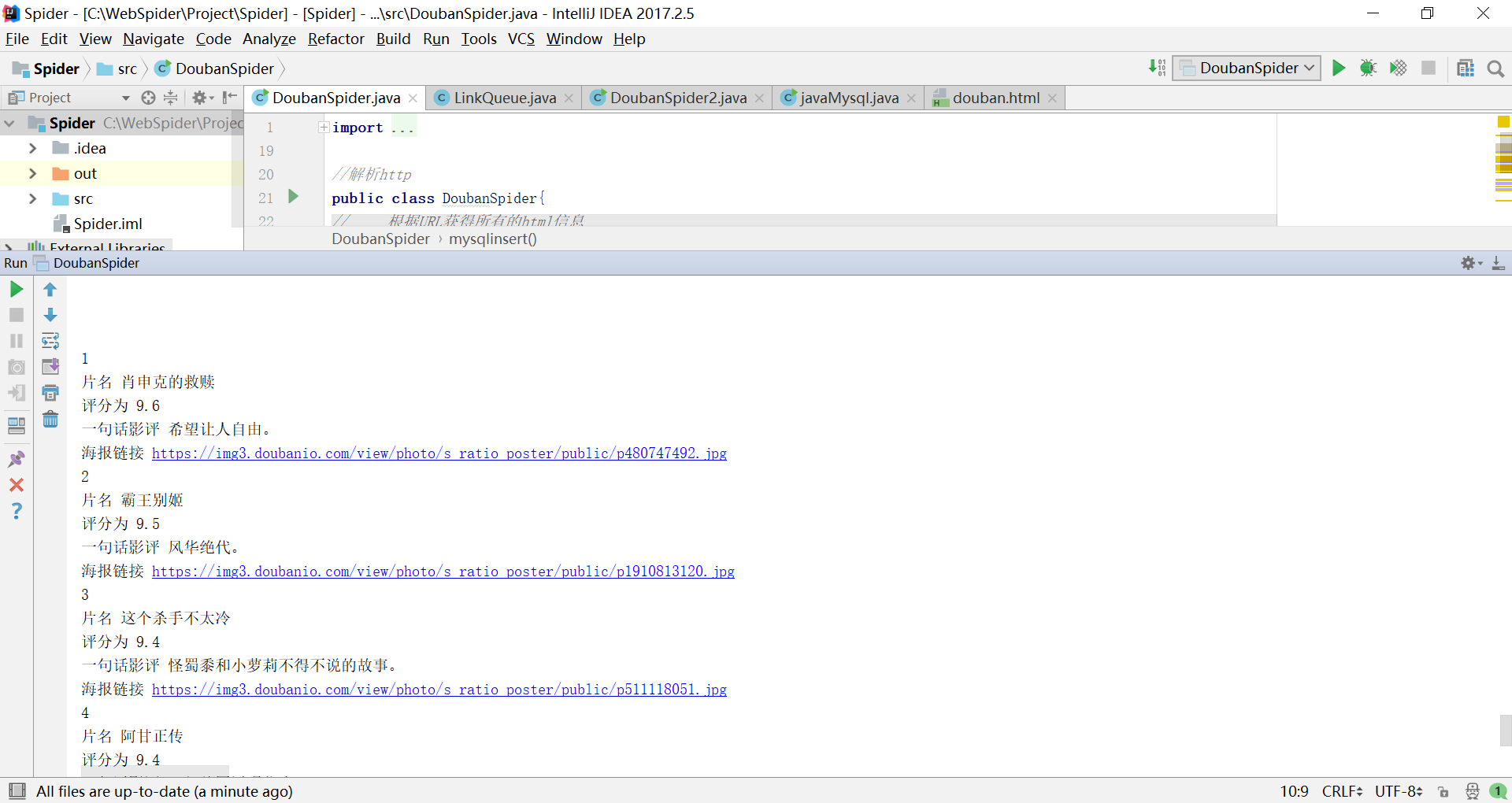
在IDEA上运行程序



以html的形式获取回应response



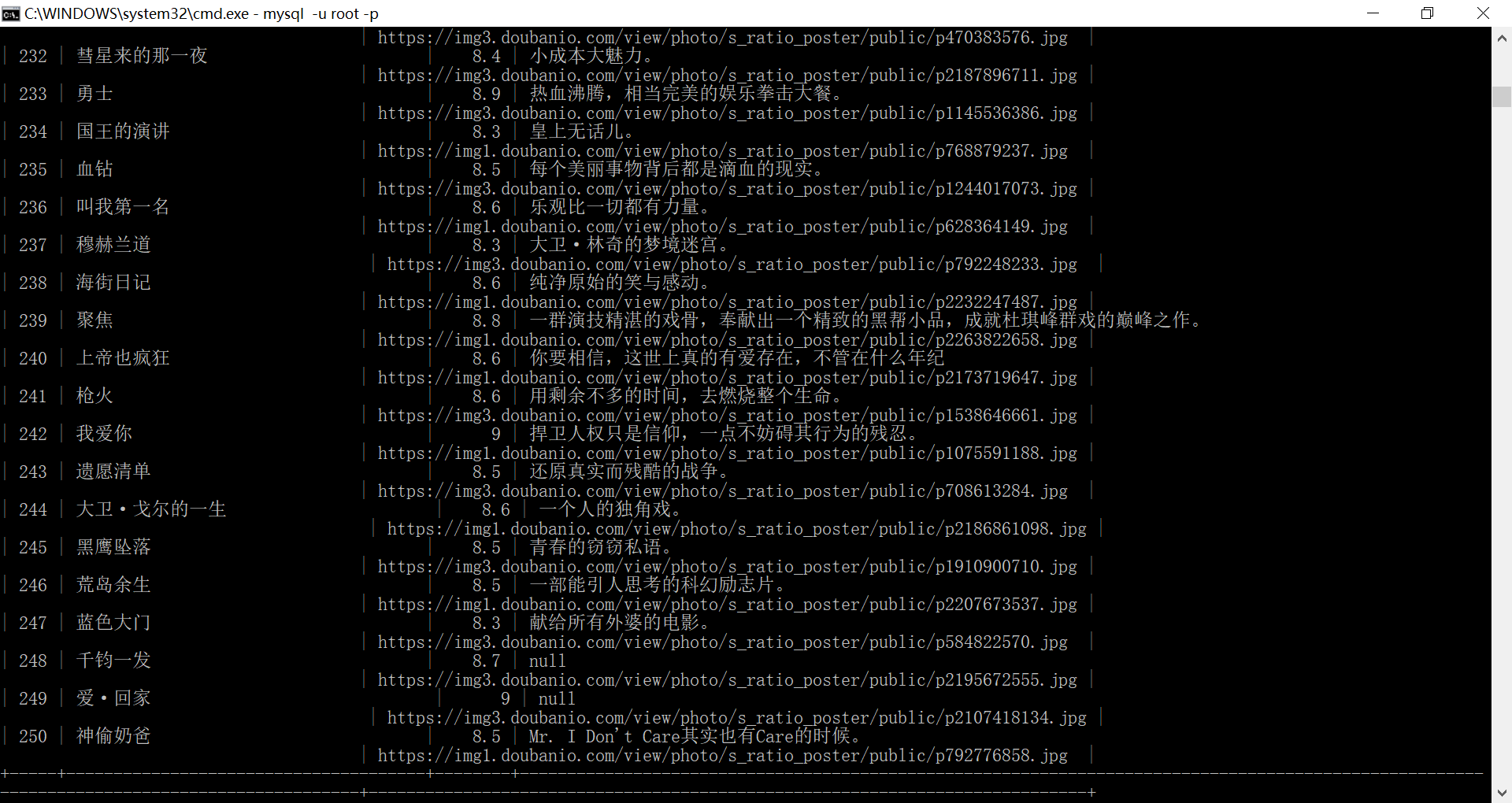
用Jsoup解析response的实体entity，获取电影信息。



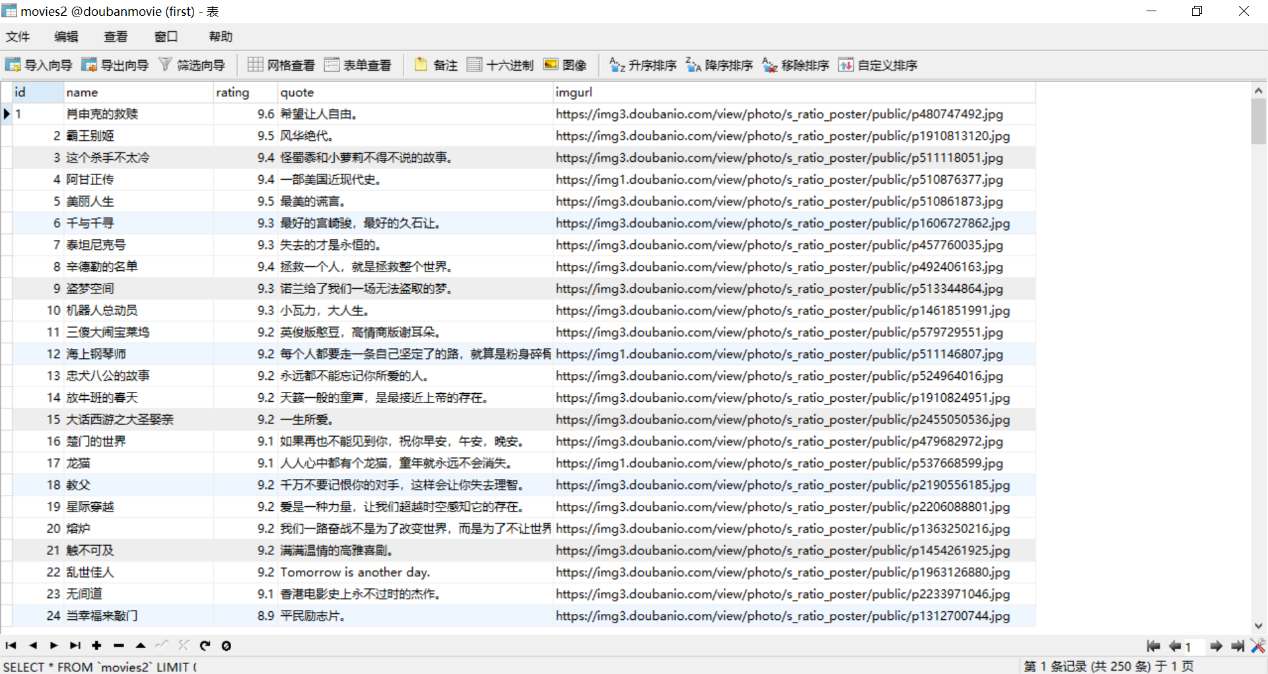
将电影信息插入到数据库中，查询信息如下。



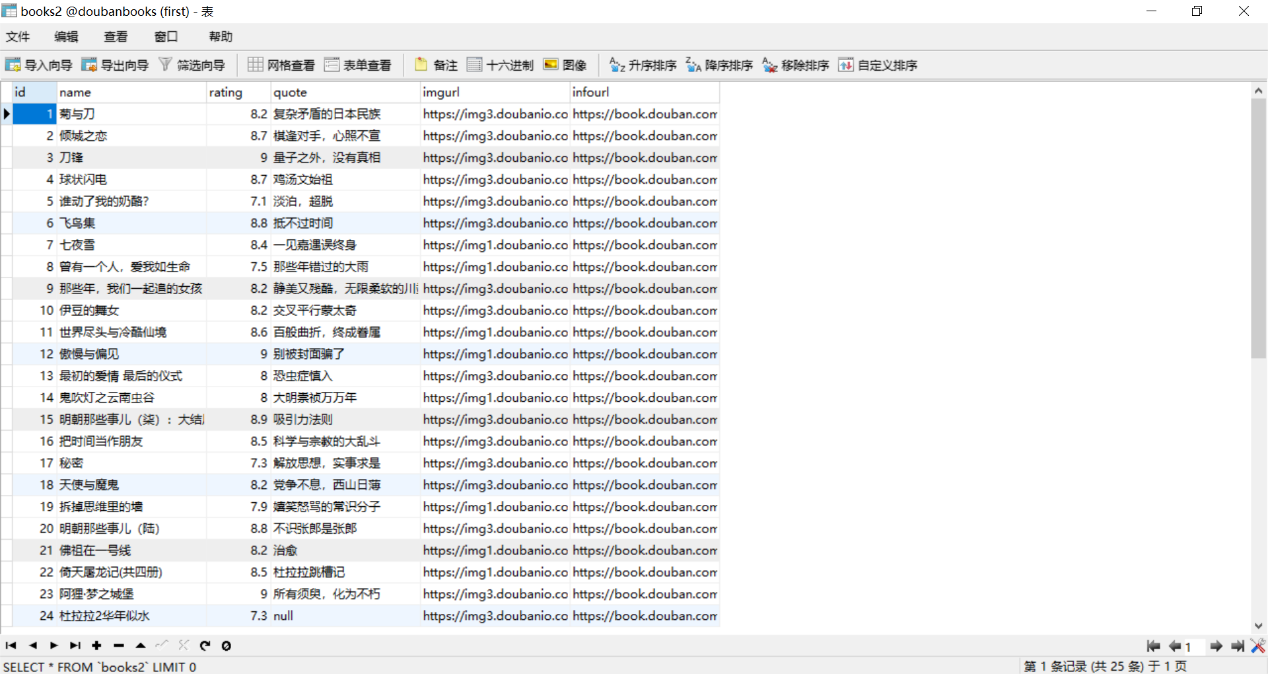
显示top250的所有电影



在数据库可视化窗口中更直观地查看数据库。

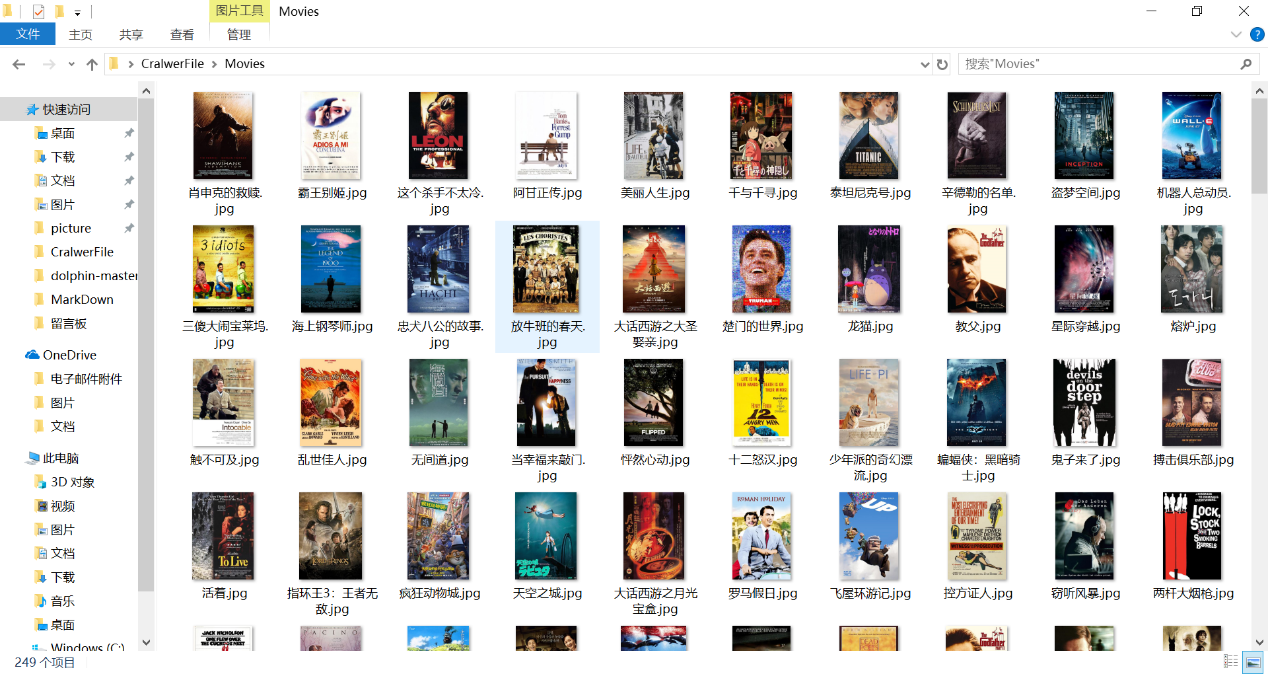


同理，也可以在数据化可视窗口中查看爬取的豆瓣图书top250的信息。不仅限于豆瓣网，其他网站也可以获取所需信息。

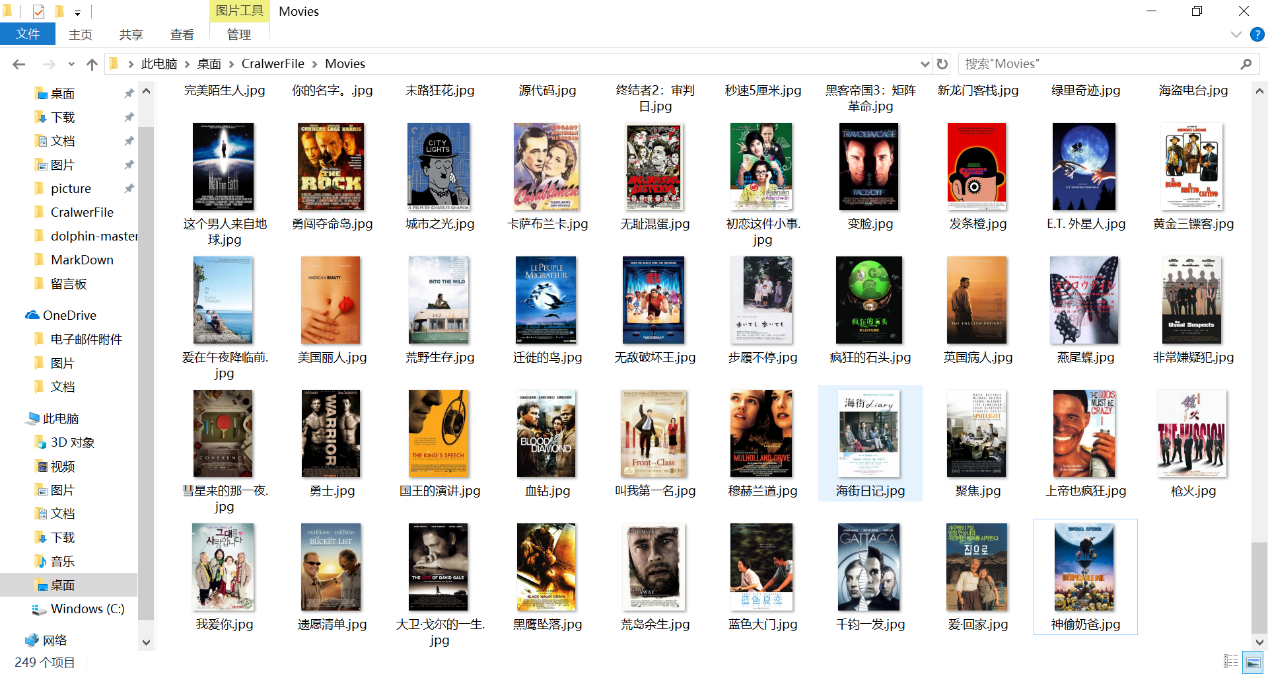


通过查询数据库中电影海报的URL，在Java中下载链接图片，将图片读成二进制流保存到二维数组中。

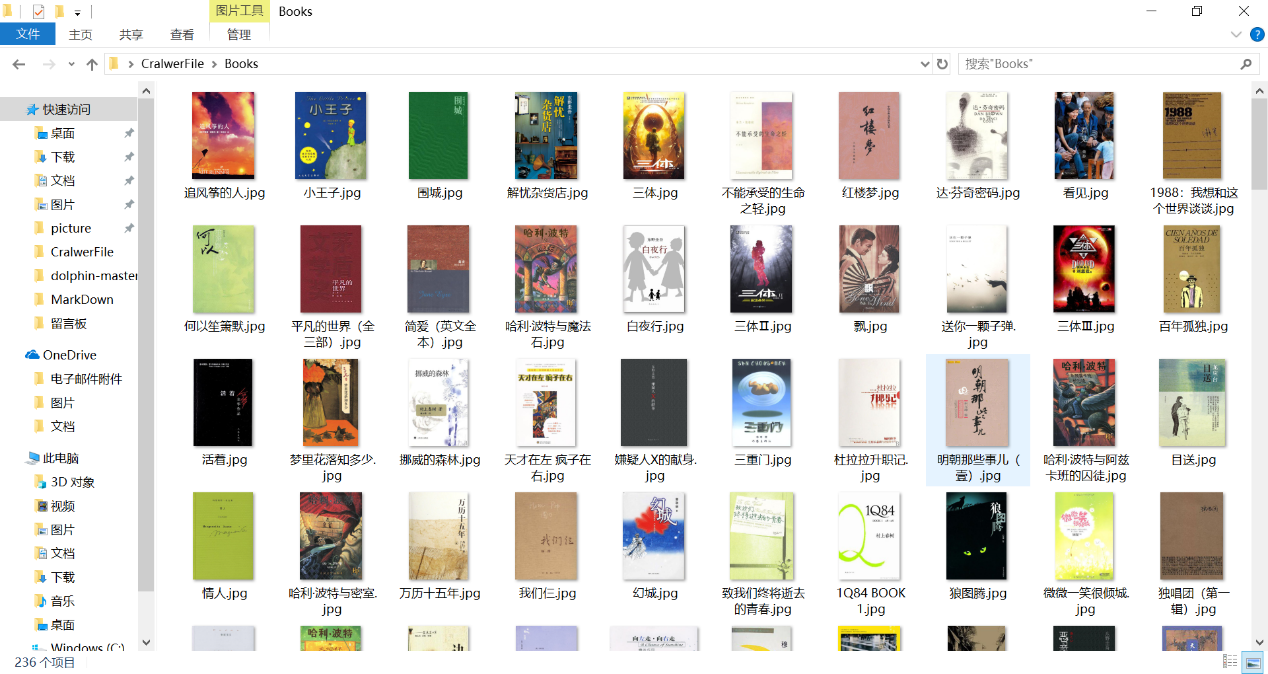
可以获得电影海报如下：



获取所有图片，存储到一个文件夹中。



同理也可以获得图书封面的URL。下载图书封面。



# 第五章 总结

通过此次实现了网络项目，掌握了Java的基本语法，MySQL环境的搭建配置，本地服务器的搭建运行，掌握了开源库httpclient、Jsoup的基本应用方法。熟悉了用Java连接数据库，并向数据库中插入信息，在发出请求时查询数据库中的信息。理解了MySQL的表单、列的构成，能够根据不同的需求建立相应的表单，并且修改表单信息。了解了爬虫的基本原理，能够独立完成一个爬虫项目，能够根据所需信息选择性地抓取文本信息。

由于算法推荐的内容有较多复杂的算法因子，与机器学习内容相关，超出个人能力范畴。因此这部分内容未能实现，可在日后致力于实现此部分内容。

在此次项目实现的过程中，能够快速地了解所需要的技术，并快速地学习技术，学习实例，并能通过使用来掌握。