**实验报告**

专业：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

姓名：商凯

学号：

日期：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

地点：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

课程名称：java应用技术基础 指导老师：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_成绩：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

实验名称：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 实验类型：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_同组学生姓名：\_\_\_\_\_\_\_\_\_\_

**Java网络爬虫————简单爬取一篇读过的小说**

1. 实验原理及设计思路

结合课上老师所讲内容以及网上搜索到的资料，决定利用HttpClients + Jsoup工具来完成本次网络小说爬取工作。

自己初步是想直接利用java本身自带的net来实现的，但发现后面数据解析比较麻烦，直接使用现成的工具包更为方便。

实验步骤：

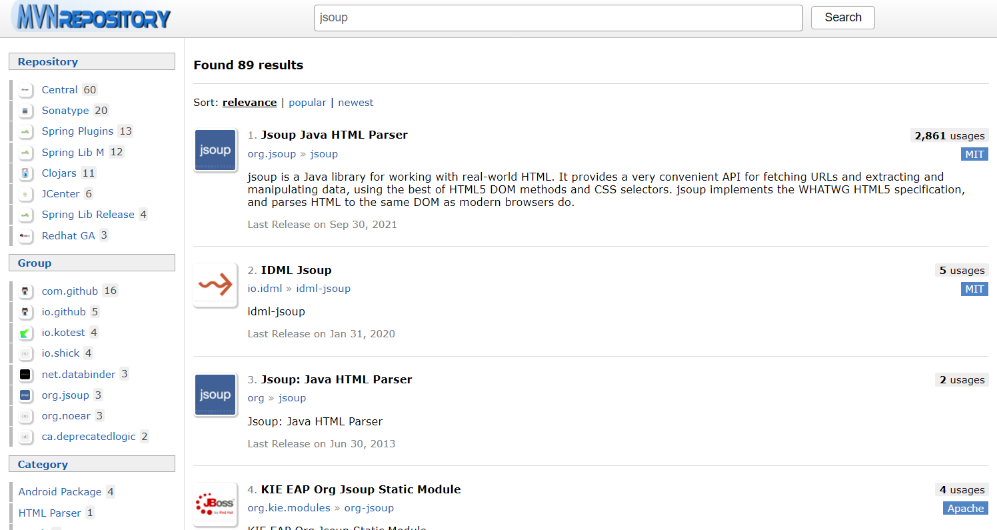
1 在IDEA上建立一个Maven工程，可以直接在pom.xml配置文件中导入HttpClients 和 Jsoup的jar包

2利用HttpClients工具，发送请求，类似模仿人的点击游览，可以获取页面数据

3 利用Jsoup工具从宽泛的页面数据中来解析我们所需要的部分

4 将解析到的数据下载到我们的txt文件中

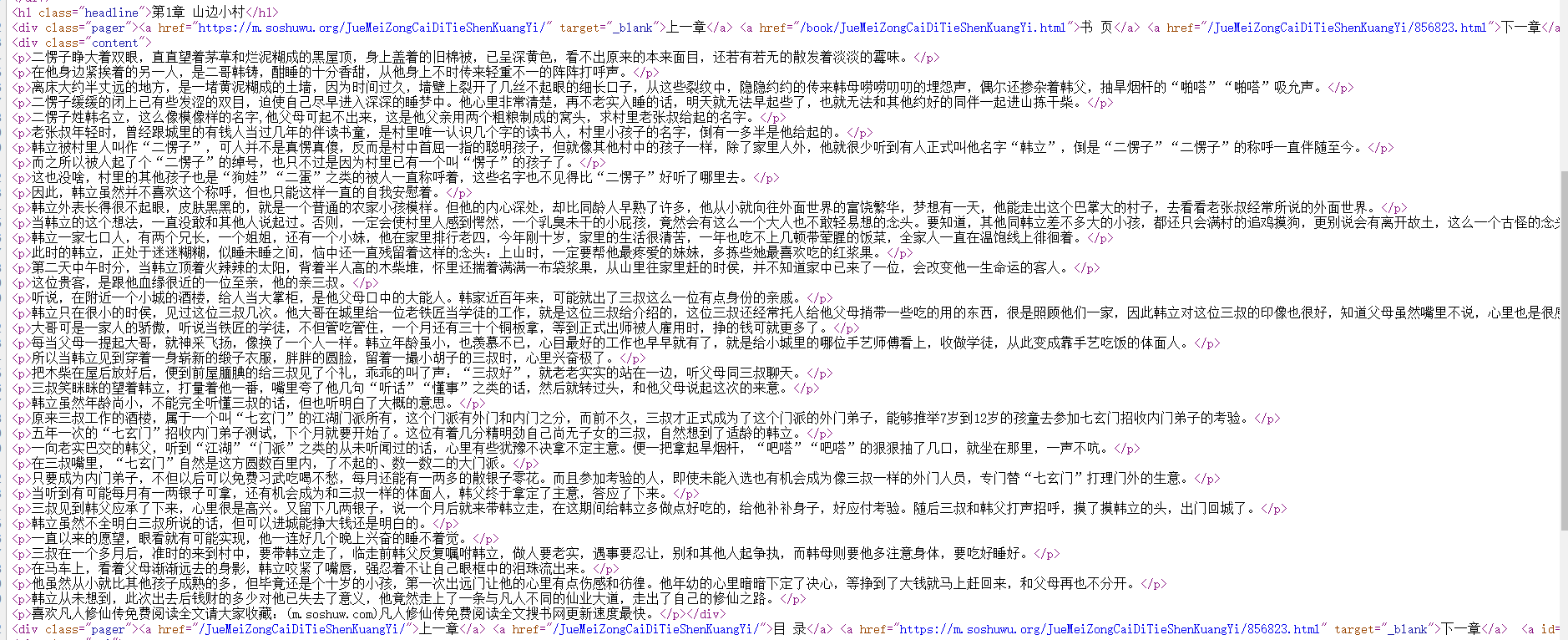
1. 使用方法及运行实例
2. 添加相应的jar包



2使用HttpClients工具jar包模仿人访问页面

3.使用Jsoup解析页面

Jsoup在学习前端的稍微一起学习了一点可以很方便来解析页面的数据，可以轻松将我们所需要的小说和标题资源存储到String中



4．将读取到的数据写入novel.txt文件中

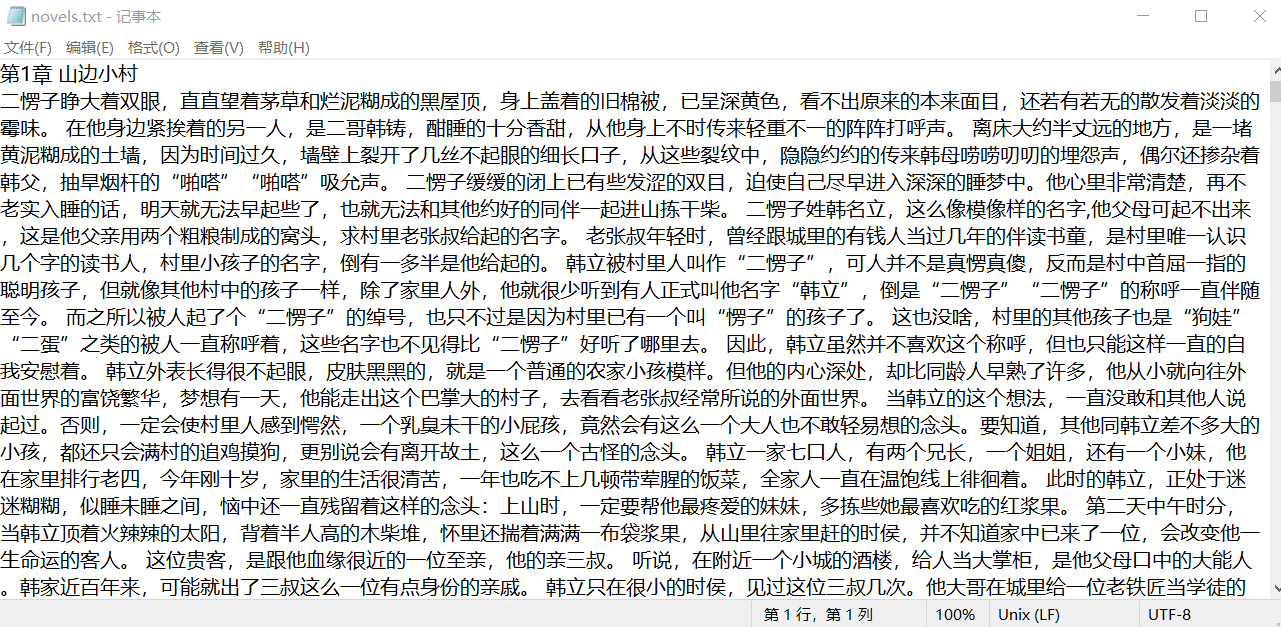


完成

为正确判别我们的小说是否下载成功，我们要在最后idea中打出每一章的标题以及每一章是否成功存入txt文件中



这里鉴于该本小说章节过多，我们就只存取前1000章，可以发现每一章都成功答应出来



Txt中内容也没有问题，能够正常阅读。

1. 改进与不足之处

为了便于阅读小说的话，应该每50章创建一个txt文件，将所有章节小说存取在一个txt文件中，不便于定位上次阅读地点。

我们应该也可以直接在html页面中解析出下一章内容的uri地址



而不是通过找规律直接输入下一章的uri地址。

本次爬取小说步骤比较简单，如果是复杂功能则需要分模块编写。