**爬取中国图书网的小说书籍信息**

1. **采集思路分析**

**1、数据采集所用到的技术**

Selenium插件的使用、Scrapy爬虫框架、Flume日志采集。

1. **数据采集实现思路**

**爬取中国图书网的小说书籍信息**

1. 判断网页是静态网页还是动态网页

打开网页查看网页源代码发现是静态网页



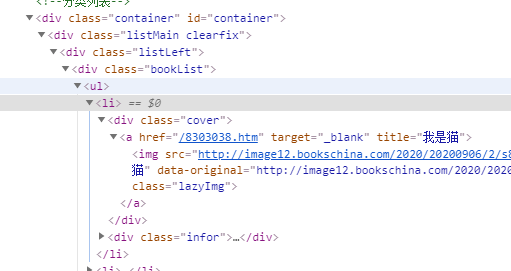
1. 选取需要爬取的数据

书名： names 、作者：author 、出版社： publisher 、出版时间： out\_data 、定价（折后）：price 、原价： price1、打折数： beat 、评论数：commnet、读者评分：collection、页数：paper 、包装：pack、Isbn： isbn；





需要爬取的数据均在详细页中，故此需要先爬取每本书详细页的链接在传入进去爬取详细页的内容。

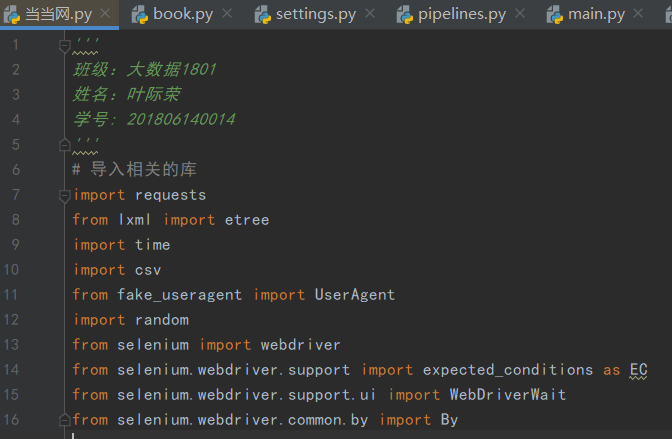


详细页的内容在li标签里的div里a标签里的href。发现里面的链接不是全部，需要对链接进行拼接。

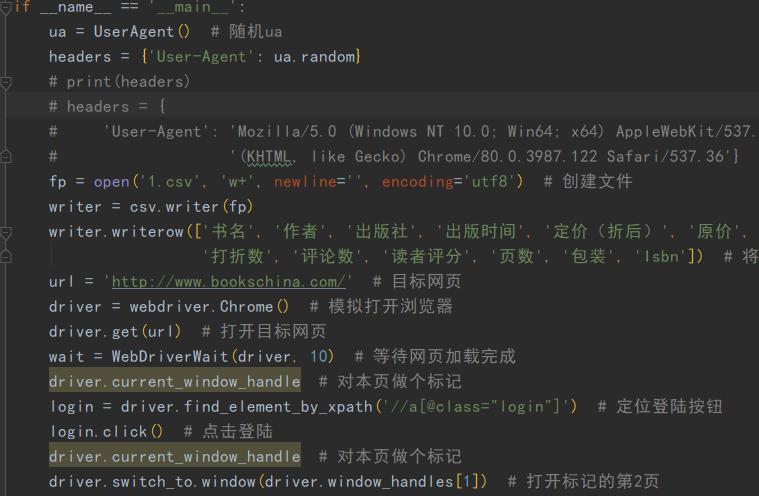
1. **数据采集实现步骤**
2. Selenium插件的使用
3. 利用Selenium插件模拟浏览器打开网页并登陆
4. 登陆后将解析后的网页内容传入爬取详细页的链接的函数中
5. 然后先爬取每本书的详细页的链接再进行循环爬取详细页的内容，爬取的每本书的详细信息保存在csv文件中再进去下一页的函数中
6. 利用Selenium插件模拟点击下一页然后再解析网页并传入到爬取详细页的链接的函数中。
7. 再利用try避免相关代码出错导致爬虫中断，以及判断是否爬取到最后一页。
8. Scrapy爬虫框架
9. 先创建一个Scrapy项目名为zhongtuwang，在进去zhongtuwang文件里创建一个book.py文件
10. 编写item容器，将需要爬取的字段编写完成。用于保存抓取的数据
11. 编写book.py，先在parse中编写爬取详细页的链接，爬取完拼接后传入parse\_dataile中进行爬取详细页内容，然后编写下一页的链接，拼接后传入parse爬取详细页链接。编写parse\_dataile爬取详细页的内容。
12. 编写pipelines.py。将爬取的内容进行保存在csv文件中
13. 编写settings.py，设置随机ua和USER\_AGENT。设置ROBOTSTXT\_OBEY为False防止打不开网页。设置休眠时间。设置ITEM\_PLPELINES让其数据可以传入pipelines进行数据保存。设置初始下载延迟。
14. 编写一个main.py文件。直接导入命令行函数。可以直接运行scrapy框架。
15. Flume日志采集。
16. 将采集的数据用Navicat Premium 15传入master节点的数据库中。
17. 启动所有的[Hadoop](http://lib.csdn.net/base/hadoop" \o "Hadoop知识库" \t "https://www.cnblogs.com/beeman/p/_blank)守护进程。打开数据库查看是否有导入的数据
18. 进入flume里的conf中创建编写flume配置文件book.conf。Agent，sources，channels，sink，bind。
19. 运行flume配置文件book.conf。查看是否出错，成功后则在浏览器中打开HDFS查看是否有文件输出
20. **数据采集代码实现**

**1、第一种采集技术Selenium插件的使用、代码及注释**

导入相关的库



在主函数上编写模拟浏览器操作打开网站并登陆，并且创建csv文件待会将爬取的数据保存在里面



传入相关的账号密码以及自行输入验证码：打开需要爬取详细页链接时解析网页并将解析的网页传入爬取详细链接的函数get\_info中。

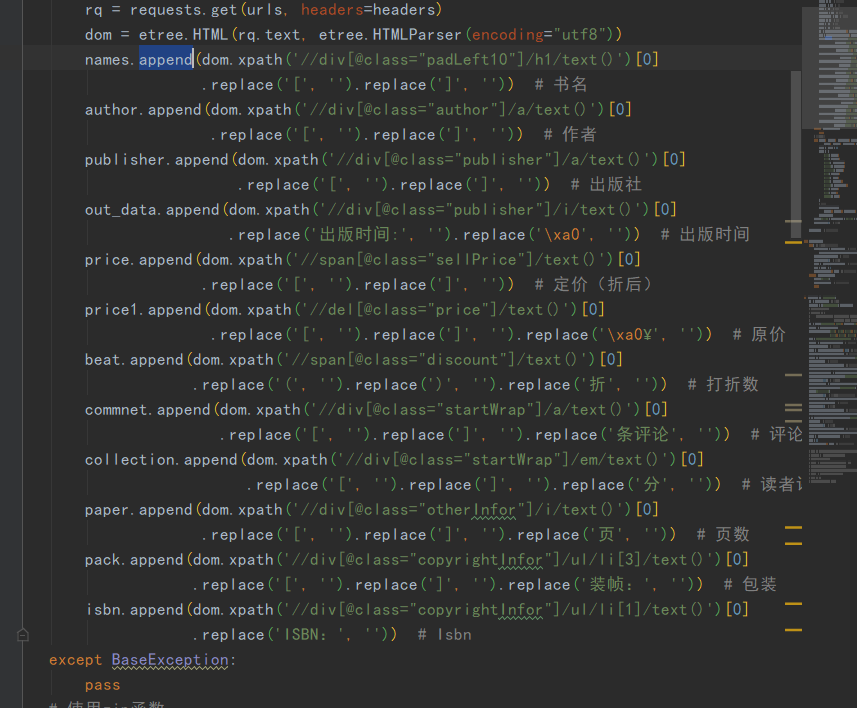




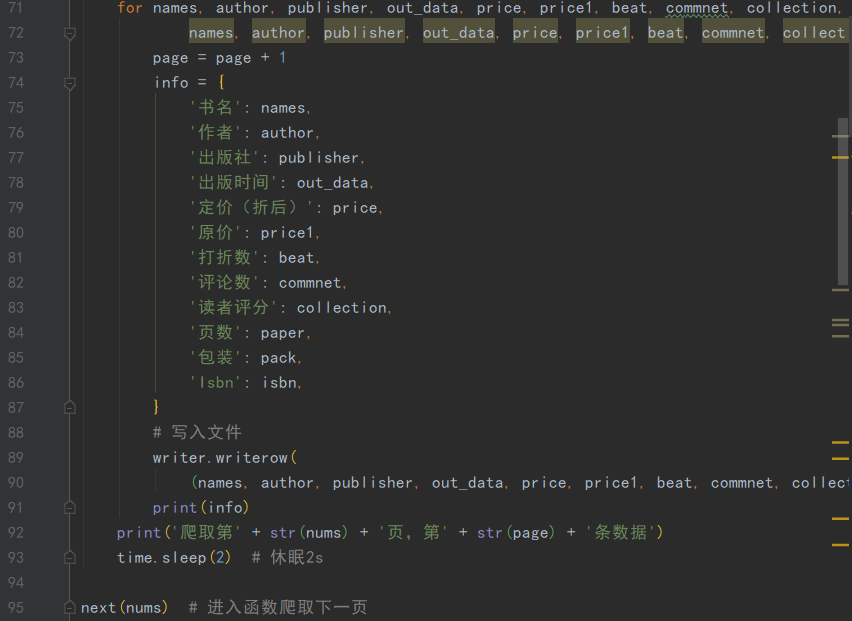
将爬取到的详细页的链接进行循环并拼接进行爬取详细页的内容，利用try避免出错导致爬虫中断。



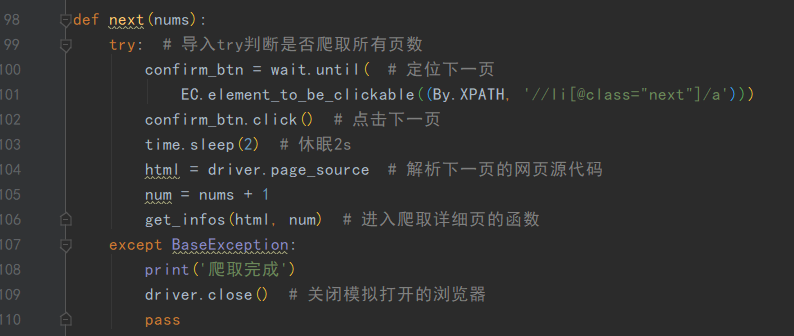
解析详细页网页进行爬取信息



使用zip函数将爬取的内容打包成元组然后返回由这些元组组成的列表，写入csv文件中。并打开next函数打开下一页



next函数中让浏览器打开下一页并解析网页传入爬取详细页链接的函数中。利用try判断是否爬取完成所有页数



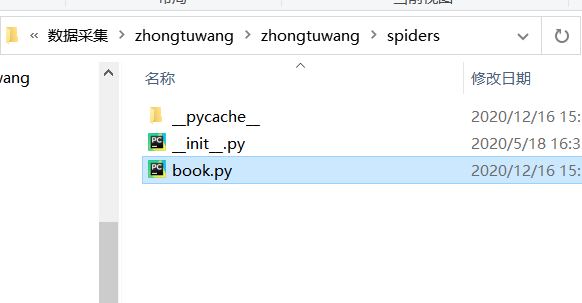
**2、第二种采集技术Scrapy爬虫框架代码及注释**

打开控制台创建名称为中图网的scrapy项目。

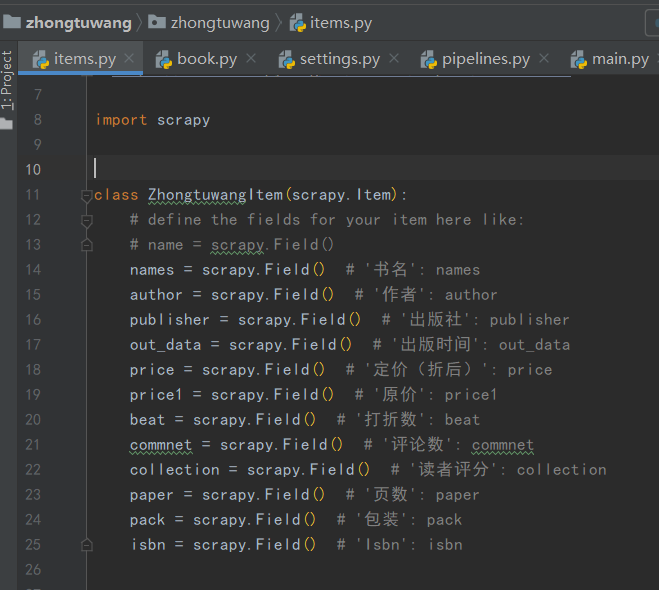
输入scrapy startproject zhongtuwang创建名称为zhongtuwang的scrapy项目,

进入zhongtuwang后输入scrapy genspider book bookschina.com

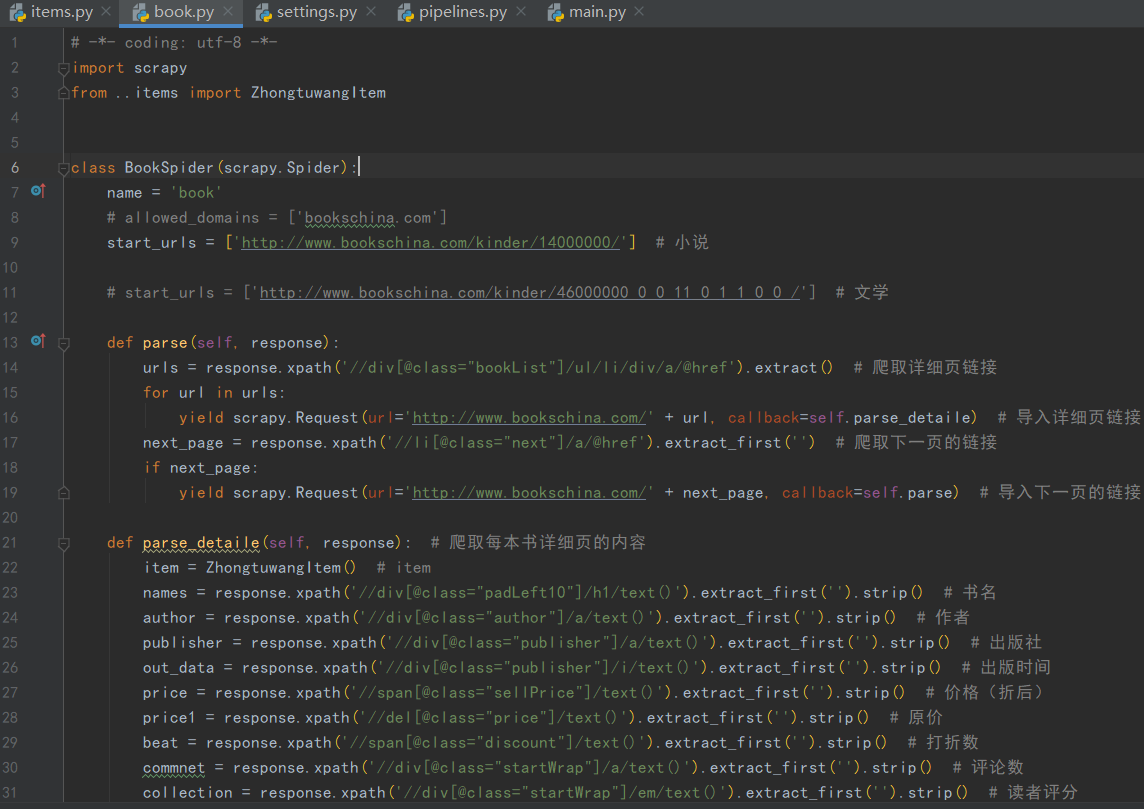
创建一个book的脚本文件。



用PyCharm,打开创建好的zhongtuwang项目，进入itmes.py，创建爬取内容字段

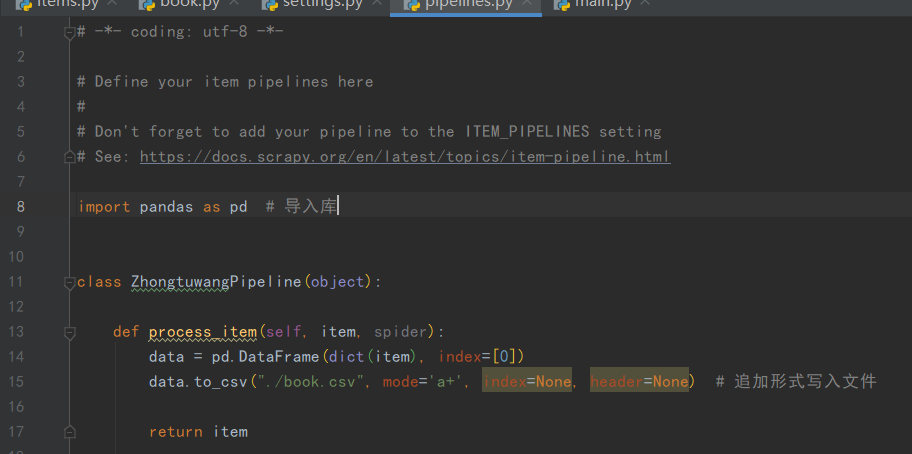


编写book.py，parse详细页的链接，下一页的链接以及详细页内容parse\_dataile。



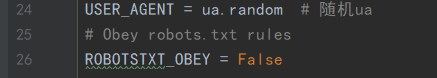


编写pipelines.py。将爬取的内容进行保存在csv文件中

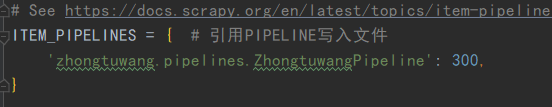


编写settings.py，配置相关参数



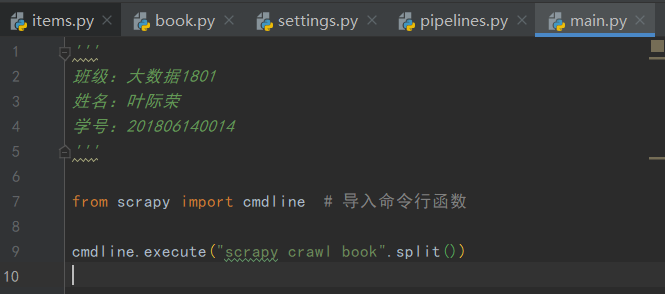






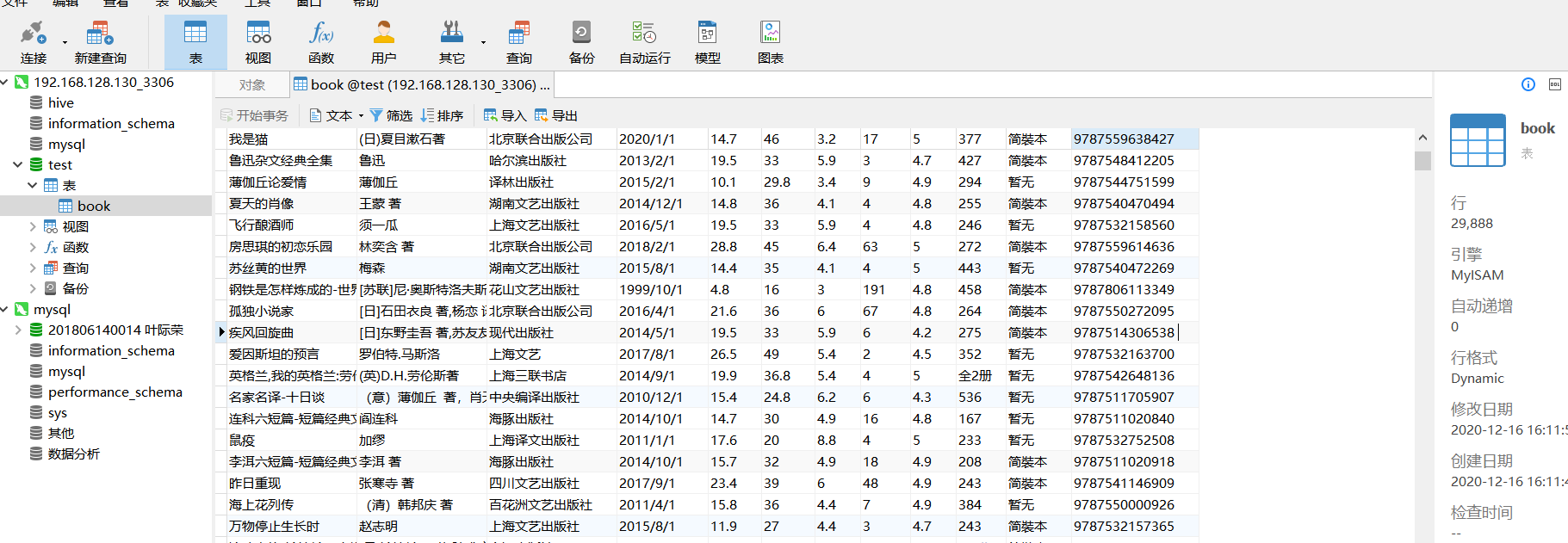


编写一个main.py文件。直接导入命令行函数

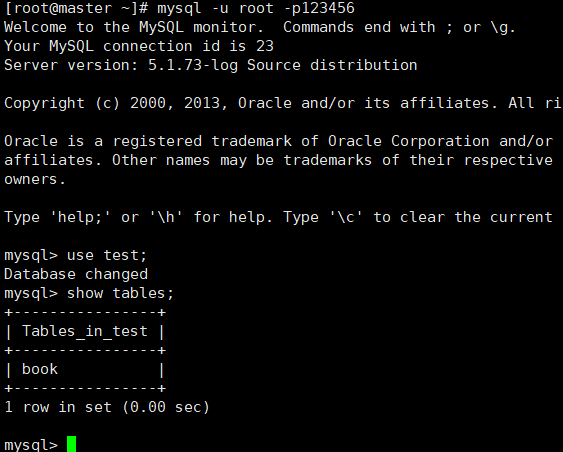
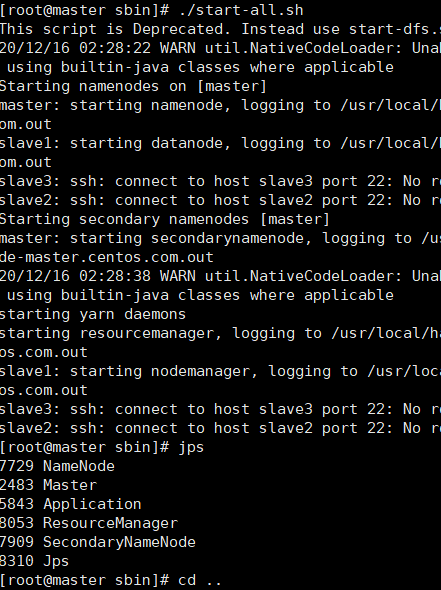


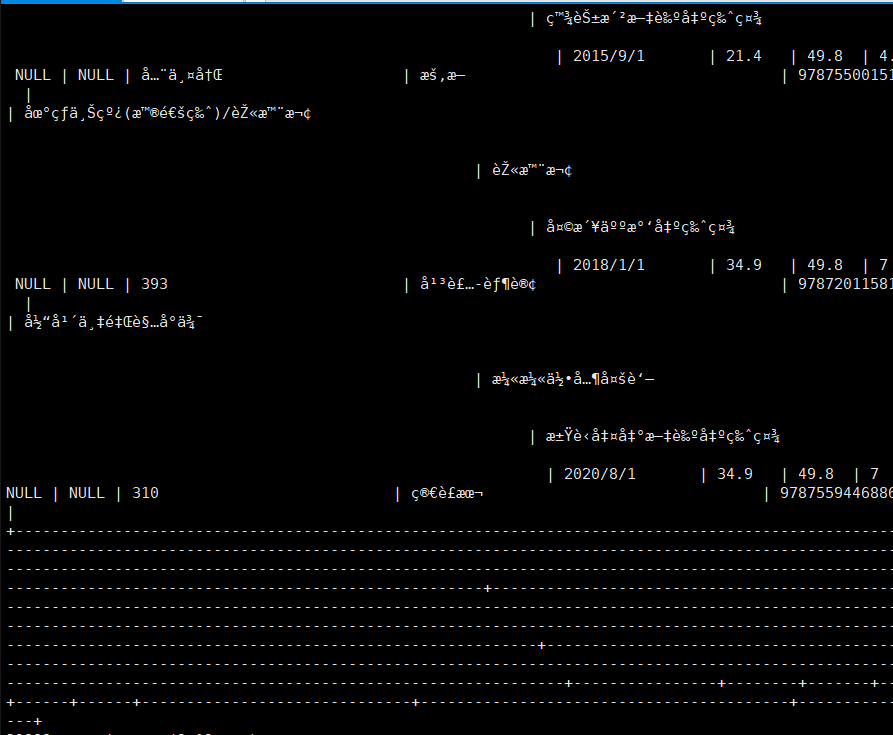
**3、第三种采集技术Flume日志采集代码及注释**

将采集的数据用Navicat Premium 15传入master节点的数据库中。

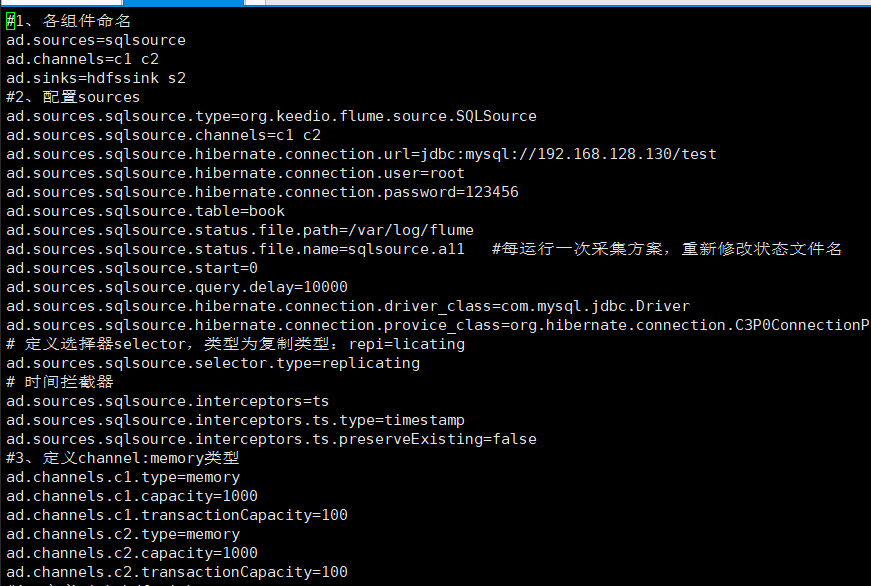


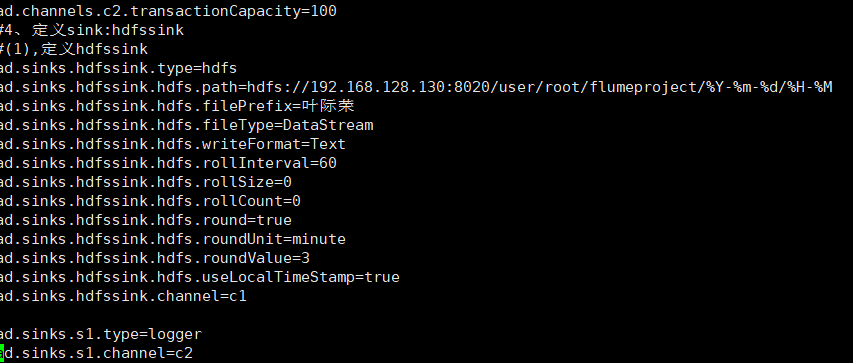
启动所有的[Hadoop](http://lib.csdn.net/base/hadoop" \o "Hadoop知识库" \t "https://www.cnblogs.com/beeman/p/_blank)守护进程。打开数据库查看是否有导入的数据



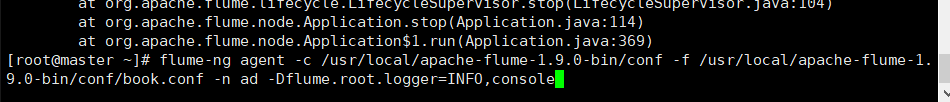


进入flume里的conf中创建编写flume配置文件book.conf。Agent，sources，channels，sink，bind。



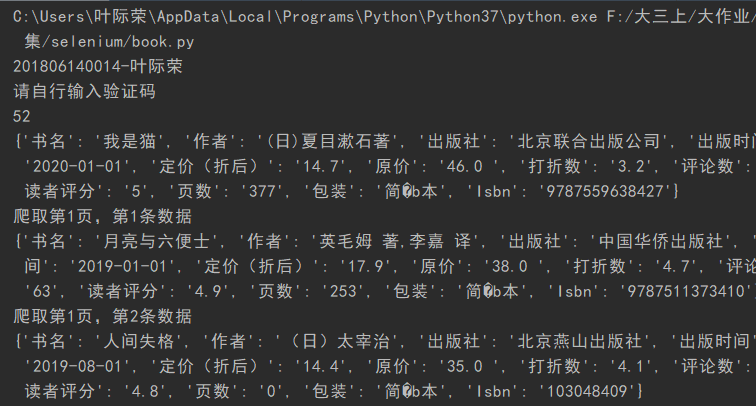


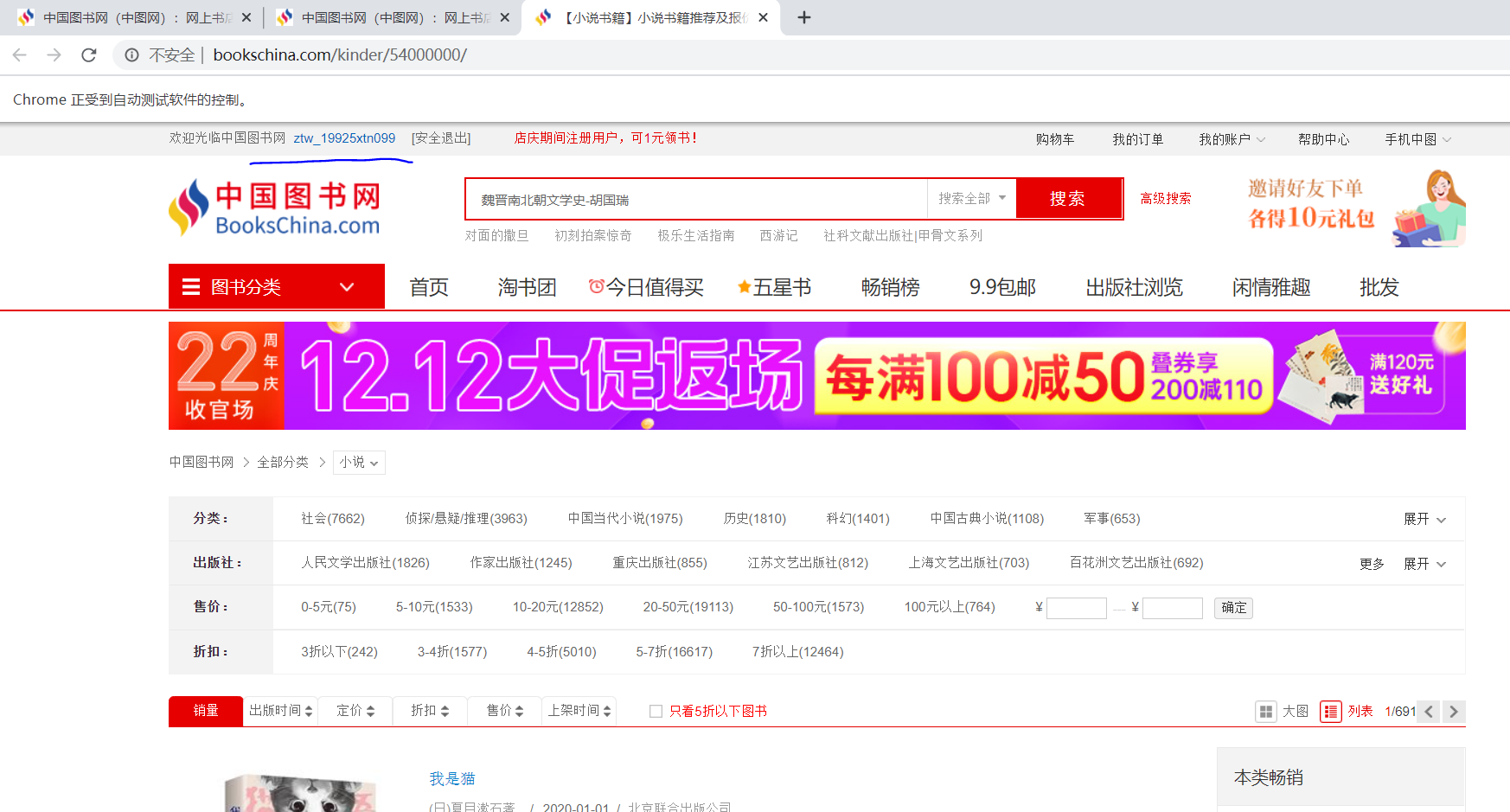
运行flume配置文件book.conf。

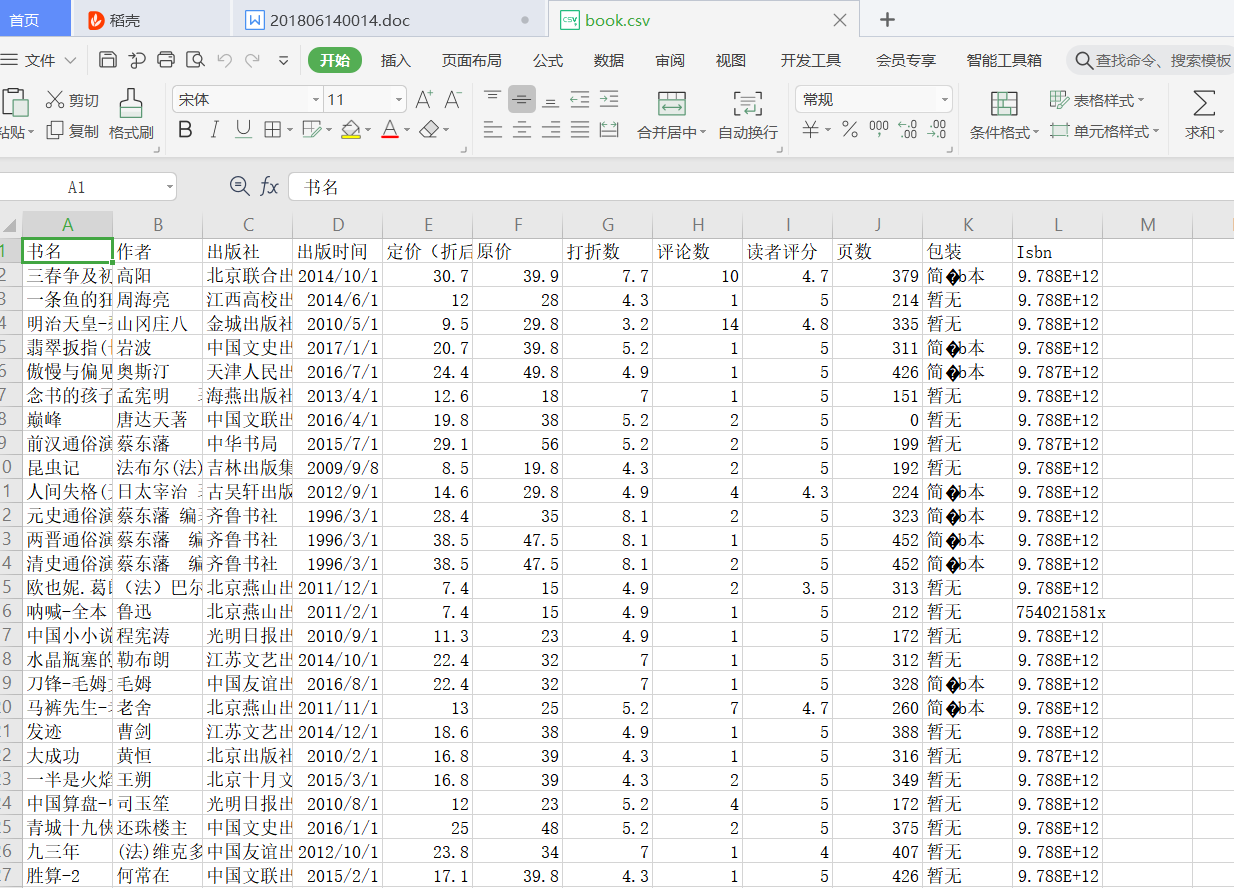


1. **数据采集运行结果及简要说明**

**1、第一种数据采集技术Selenium插件的使用运行结果及简要说明**

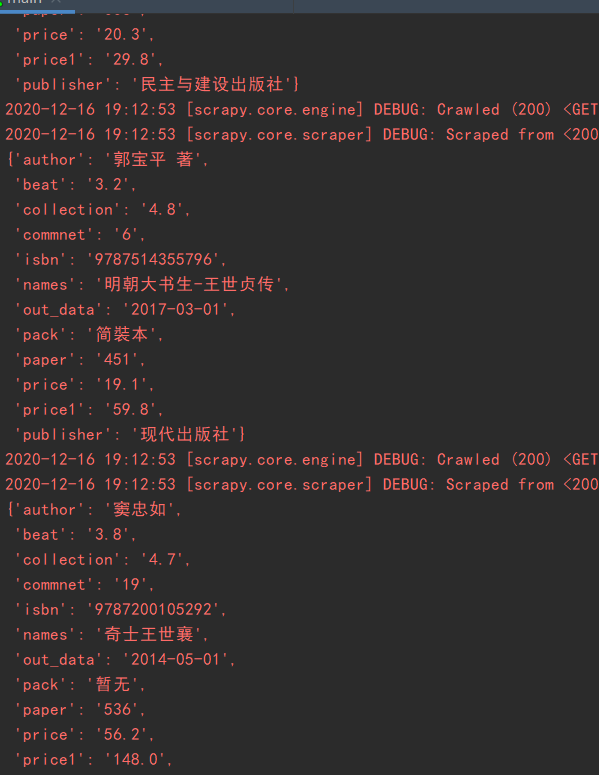


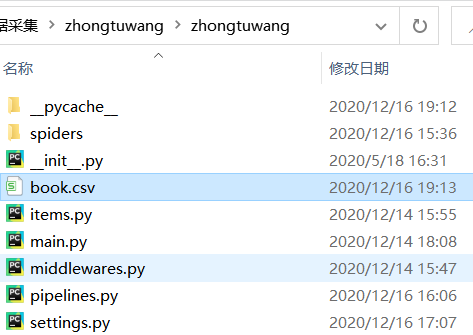


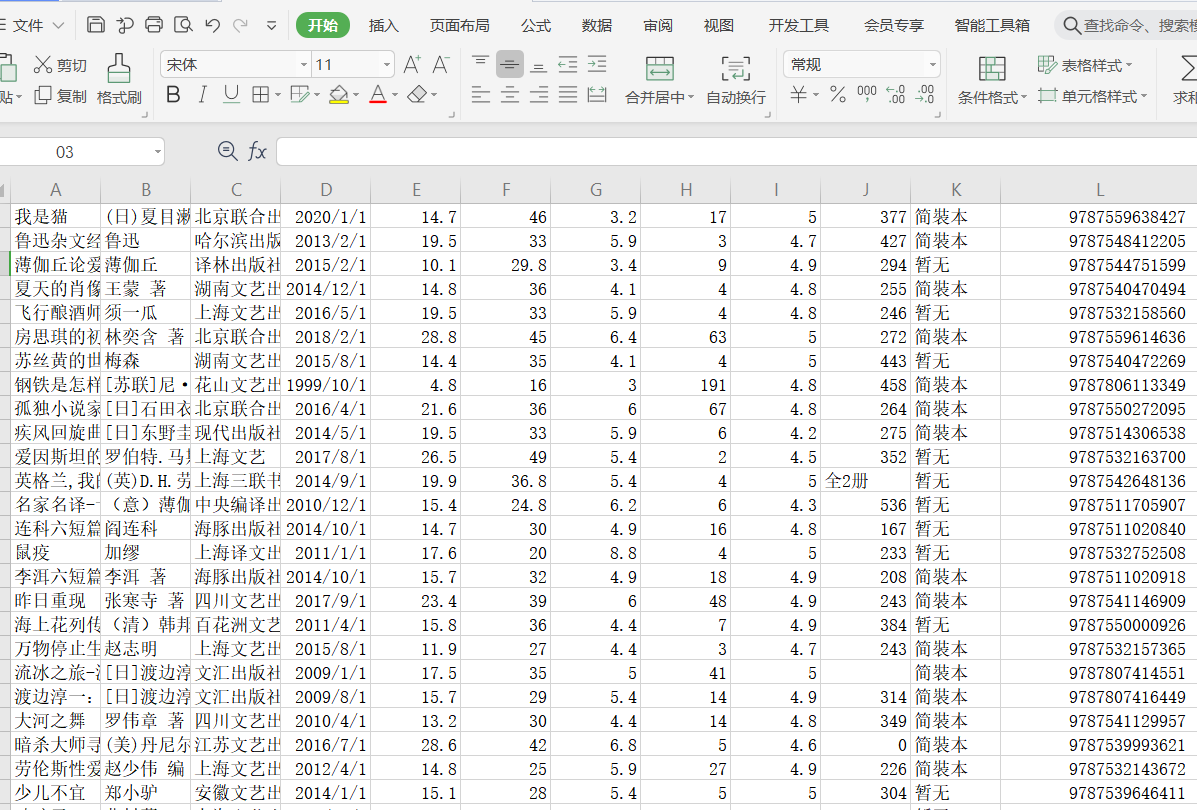


输出文件后查看文件。乱码的字体是繁体字，所以乱码

**2、第二种数据采集技术**Scrapy爬虫框架**运行结果及简要说明**

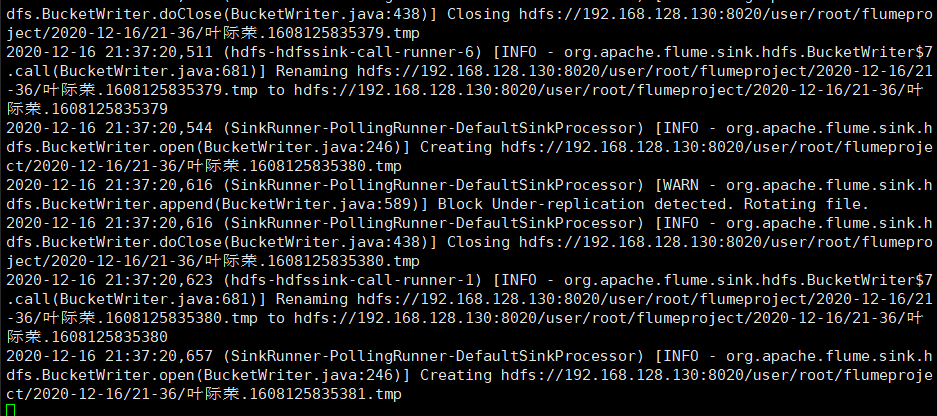




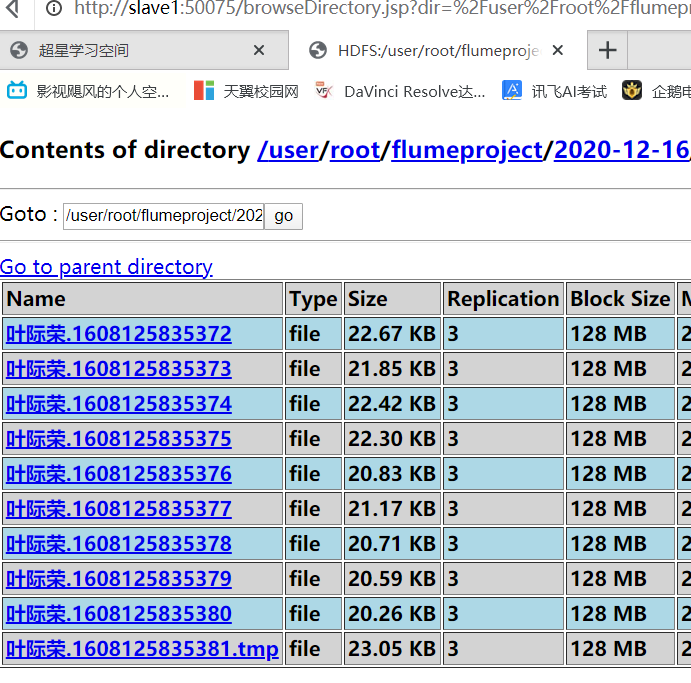


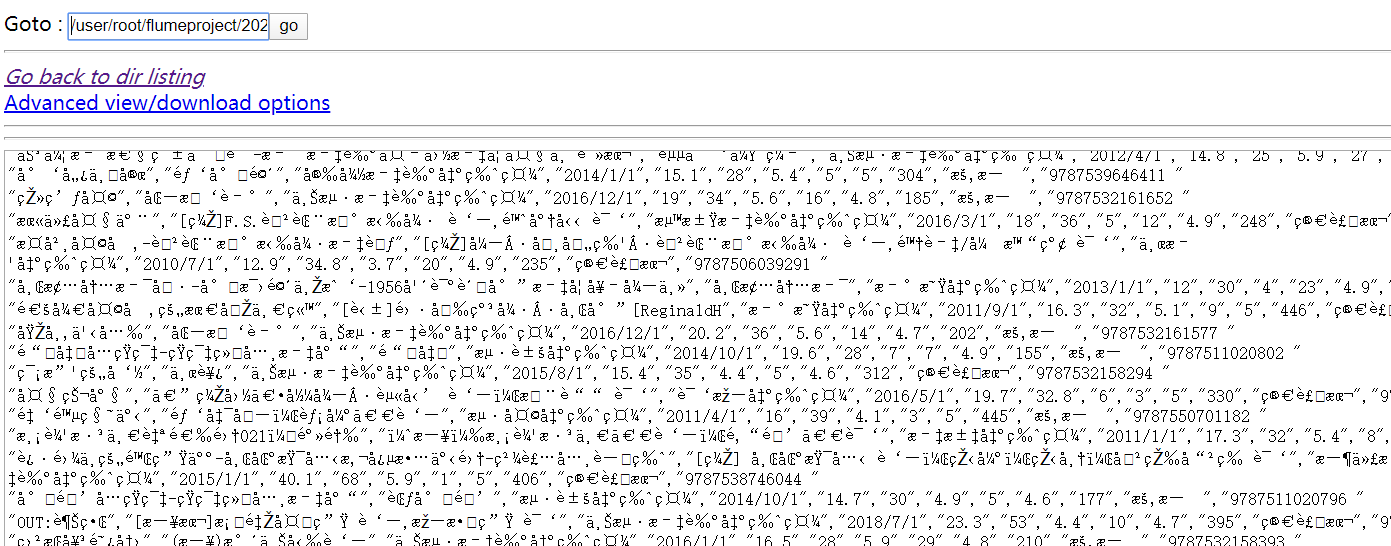
**3、第三种数据采集技术Flume日志采集运行结果及简要说明**

运行flume配置文件book.conf后：



打开hdfs,查看





查看数据后是乱码的，这个问题没解决

1. **课程思考总结**

本次使用的技术主要是Selenium插件的使用、Scrapy爬虫框架、Flume日志采集。主要是上个学期学习的内容。scrapy 是一个为了爬取网站数据，提取结构性数据而编写的应用框架，只需要实现少量代码，就能够快速的抓取到数据内容。对目标网站分析需要提取的数据，直接在item.py文件中添加字段即可。简单的scrapy爬虫需要5步。创建爬虫项目，创建应用。明确字段 （编写items.py），制作爬虫 （编写爬虫开始爬取网页，解析网页内容），存储内容 （设计管道存储爬取内容）。爬取数据166138条

爬取中国图书网由于是静态网页且反爬机制比较简单。开始先写过当当网的数据信息。实现代码爬取后发现爬取大量信息需要登录才可以访问详细页进行爬取。期间尝试过cookie进行爬取以及相关换ip代理等方法，但没有成功过。所以使用Selenium插件去模拟登陆和点击等进行爬取当当网详细书籍的信息。但发现爬取一本书就需要点击下网页这太过于麻烦需要循环点击网页太可以爬取详细信息。故此换了中国图书网进行爬取。对于Selenium插件的使用和Scrapy爬虫框架的使用。发现Scrapy爬虫框架比Selenium简单太多而且爬取的信息又快。

Flume日志采集本学期新学习的。但是相关的配置不熟练。正则过虑器。等相关不会。导致大作业里的正则过滤器写了也没有相关文件。所以删除了这部分。

我直接是在Navicat Premium 15连接master里的数据库，将文件导入数据库中。在配置flume将数据库里的文件提取出来到hdfs里并按照时间命名。配置方案第一步是定义agent(代理)及agent下的sources、channels、sinks即各组件的命名，第二步是定义sources(接收数据源)，第三步是设置channel(管道)的类型等，第四步是设置sink的类型和细节设置最后还有bind。

总的来说，对于反爬机制厉害的网页，自己心有余而力不足。对于flume相关的使用还不够熟练。需要自己多去实践以及理解相关的使用方法。