山东大学 软件 学院

WEB数据管理 课程实验报告

|  |  |  |  |
| --- | --- | --- | --- |
| 学号：201700301076 | 姓名：成曦泽 | | 班级：2017级7班 |
| 实验题目：克服反爬虫障碍 | | | |
| 实验学时：4 | | 实验日期：2020.03.11 | |
| 实验目的：   * 【实验要求】   + 选择具有反爬虫障碍的网站为目标     - 不要选择封锁IP的网站：百度，豆瓣       * 提前百度一下     - 不要使用爬虫app, 必须自己编写爬虫代码   + 分析网站的反爬虫障碍   + 编写爬虫软件，克服障碍获取相应的数据 | | | |
| 硬件环境：  笔记本 | | | |
| 软件环境：  Windows  Pycharm | | | |
| 实验步骤与内容：   1. 目标网站分析 2. 网站名称URL   京东：  京东一级网页：<https://list.jd.com/list.html?cat=1713,9291>爬取每一种类别的图书目录页  京东二级网页：<https://item.jd.com/12508277.html>等每一个图书的单品页  当当：  当当一级网页：<http://book.dangdang.com/>  当当二级网页：<http://category.dangdang.com/cp01.49.05.01.00.00.html>  当当三级网页：http://product.dangdang.com/27926484.html   1. 该网站采用反爬虫技术   京东：   1. 在京东网页中，所有的页面的属性信息都是使用Ajax进行异步加载，需要响应之后才能读取信息。 2. 京东的页面有分页显示，所以需要通过切换页面来获取所有页面的信息。   当当：   1. 当当的二级页面中是使用分页进行存储的，所以需要通过切换页面来获取所有的信息。 2. 当当的二级页面中，每一个类别只能显示最多100页面 3. 所使用开发工具 4. 开发语言   Java:项目中有一个功能需使用java爬虫  Python:网页数据的定时爬取   1. 使用的模块、工具   Java:jsoup、fastjson  Python: BeautifulSoup、json、pandas、requests、pymysql   1. 操作过程 2. 京东： 3. 爬取思路   首先爬取所有图书种类的目录    然后在每一个目录中，爬取当前目录下的所有图书信息。由于每一个目录是使用分页进行的，我们还需要分页爬取。     1. 工作思路   首先、从某一图书种类的目录中，获取所有图书种类的链接；  随后我们对每一个目录中的书目进行爬取，为了解决分页的问题，我们观察url“<https://list.jd.com/list.html?cat=1713,3259&page=2>”，该url中的参数page，表示的就是页数，所以我们可以通过修改url中的参数，更改爬取的页面，获取所有数据信息。   1. 操作步骤   观察页面元素的特征，通过BeautifulSoup获得对应的href和title，并存入数据库    从数据库中读取所有的图书种类链接，并在每一个种类中，获取总页面数，通过更改页面参数，爬取每一个页面的所有图书链接。    最后进入每一个页面的售卖链接，爬取页面相关的图书信息，检测数据库中是否已经存在对应信息，并将新的数据塞入数据库中。     1. 源代码分析   爬取所有的种类链接    设置爬虫参数：  设置请求头的User-Agent，指定要爬取的url，最后根据网页中约定的编码，设置编码格式。      根据DOM结构，爬取对应的数据，并将其存入到数据库中    爬取总页面数：    爬取所有商品页的链接：根据网页的总页面数，爬取所有的页面，并再深层次，获取对应图书信息。    爬取商品信息：在每个页面中，由于图书名称和价钱是通过ajax异步加载的，所以将在后面，方法中介绍，ISBN等信息可以直接在商品详情中看到，我们直接进行爬取即可。    随后将爬取的信息，存入到数据库中：    获取异步信息：分析网页刷新时的network，寻找是否有对应的js执行，找到了获得对应信息的js代码，通过request获取对应的结果。       1. 当当： 2. 爬取思路   首先爬取所有的目录种类    随后在每个种类中，爬取通过更改页面参数，切换页面，并爬取所有的图书链接和信息。    最后，在图书商品的页面，爬取所需要的所有图书信息。     1. 工作思路   首先、从某一图书种类的目录中，获取所有图书种类的链接；  随后我们对每一个目录中的书目进行爬取，为了解决分页的问题，我们观察url“http://category.dangdang.com/pg2-cp01.45.57.01.00.00.html”，该url中的参数pg，表示的就是页数，所以我们可以通过修改url中的参数，更改爬取的页面，获取所有数据信息。  3、操作步骤  首先，我们分析图书目录中的DOM结构，发现所有的类别都在同一个，类名为‘con flq\_head’的div下，获取所有的链接之后保存入数据库中。    在每一个目录的页面内，先获取该目录的总页面数，然后通过改变页数的参数，更换页面，爬取所有的链接及书本信息。    最后在商品页面内，爬取其他所需的商品信息。    4、源代码分析  首先爬取所有的目录链接，即con flq\_body的div下的所有a标签。  由于网页中的编码是GB2312，所以我们要提前将我们的字符格式设置为GB2312。所有的链接爬取之后，存入数据库中。    爬虫获取每一个目录的最多页数：该内容和京东部分类似不再赘述。    爬取目录页内的所有页面的每一个商品链接：    爬虫爬取商品信息页面，并存留所有信息，由于这个页面有可能是电子书，页面DOM架构不相同，且我们不准备爬取，所以可以使用捕获异常，随后抛出。  由于我们要将数据存入数据库，所以，我们要额外注意将含有英文单引号的字符串进行\转义。    为保证多线程爬取不会读写锁冲突，我选择了先进行本地存储，所有进程爬取完毕后，再一并传入数据库中。   1. 爬取数据 2. 保存格式   BOOK表：    目录Link表：     1. 数据量大小   京东：约36万条数据/单月，设置了定时脚本，每月爬取进行比价。    当当：约86万条数据/单月，设置了定时脚本，每月爬取进行比价。 | | | |
| 结论分析与体会：  在爬取数据的过程中遇到了多种困难，异步加载，分页存储等。这是一个爬虫与反爬虫竞争的过程，技术在不断进步，我们也需要不断学习。 | | | |