山东大学 软件 学院

Web数据管理 课程实验报告

|  |  |  |  |
| --- | --- | --- | --- |
| 学号：202000300125 | 姓名：贾星宇 | | 班级：2020级5班 |
| 实验题目：克服反爬虫障碍 | | | |
| 实验学时：6 | | 实验日期：2023年4月16日 | |
| 实验目的：  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：  Intel(R) Core(TM) i5-10210U CPU @ 1.60GHz 2.11 GHz | | | |
| 软件环境：  Intellj IDEA 2022.2 | | | |
| 实验步骤与内容：   1. 目标网站分析：   此次实验爬取的页面为大疆无人机的官方论坛。  url：<https://bbs.dji.com/forum-84-1.html>  通过实验一的基础，我们首先尝试最基本的方式来爬取此页面，发现爬到的信息存在错误：    经过反爬虫策略的分析，意识到此网页存在反爬虫策略，需要伪装请求头等方式来克服。  因此，进入网页查看源代码，进入网络部分，找到希望得到的内容所对应的请求头如下：    通过分析，此网页通过user-agent等方式提供握手协议，验证网页请求的合法性。因此进行伪装如下：  url = 'https://bbs.dji.com/forum.php?mod=forumdisplay&fid=84&page='+str(page) headers = {  "user-agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/112.0.0.0 Safari/537.36 Edg/112.0.1722.64" } response = requests.get(url, headers=headers)  同时，此网页还存在需要点击翻页的部分：     1. 所使用的开发工具：   本次爬虫使用python语言，主要使用urllib库进行请求的发起，随后使用beautifulsoup对网页进行分析清理。   1. 操作过程： 2. 对于网页的反爬虫，我们伪装请求头来实现反扒。同时，对网页的编码格式存在要求。通过查找内容，发现网页编码格式为utf8：     针对请求头的伪装，我们在找到请求头内容之后进行设计：  url = 'https://bbs.dji.com/forum.php?mod=forumdisplay&fid=84&page='+str(page) headers = {  "user-agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/112.0.0.0 Safari/537.36 Edg/112.0.1722.64" } response = requests.get(url, headers=headers)   1. 针对网页的分页特性，我们在网页上进行点击发现每一个页面的url是不同的，且存在规律：   第一页：<https://bbs.dji.com/forum-84-1.html>  第二页：  <https://bbs.dji.com/forum.php?mod=forumdisplay&fid=84&page=2>  第三页：  <https://bbs.dji.com/forum.php?mod=forumdisplay&fid=84&page=3>  ……  第2151页：  https://bbs.dji.com/forum.php?mod=forumdisplay&fid=84&page=2151  可以发现，除了第一页，其他页面通过page=X的方式实现url的跳转。因此我们在代码中加入循环变量来指示页码变化，通过跳转url的方式来实现对页面按钮点击的模拟：  from bs4 import BeautifulSoup import requests *#129013条+-1条数据（论坛，航拍无人机板块）* file = open("bbs.txt", "a",encoding="utf-8") for page in range(2143):  *# 爬取基本网页* url = 'https://bbs.dji.com/forum.php?mod=forumdisplay&fid=84&page='+str(page)  headers = {  "user-agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/112.0.0.0 Safari/537.36 Edg/112.0.1722.64"  }  response = requests.get(url, headers=headers)  soup = BeautifulSoup(response.text, "html.parser")   *# 寻找有用价值* soup = soup.find(id='threadlisttableid').find\_all('tbody')  for each\_body in soup:  get\_a = each\_body.find\_all('a')  for each\_a in get\_a:  if each\_a.get\_text().strip() != '':  print(each\_a.get\_text() + ' ' + 'https://bbs.dji.com/' + each\_a.attrs['href'])  file.write(each\_a.get\_text() + ' ' + 'https://bbs.dji.com/' + each\_a.attrs['href']+'\n')  print('\n')  file.write('\n')  file.close()  在代码中，首先打开一个文件用来保存爬取到的有用信息，随后使用for循环依次遍历每一个页，在进入到相应页面后，使用header来实现伪装，利用beautifulsoup实现网页有用信息的查找。分析页面发现，有用信息存在于id='threadlisttableid'中，通过列表的方式列出了一页中的每一条bbs。因此，继续循环，进入每一条bbs，发现有用信息存在于a标签中，随后通过去除空格换行等数据清洗操作，找到相关信息以及相关连接，将其以行的方式存储到txt文件中。   1. 爬取数据：   经过统计，总共爬取到的有意义bbs共**129013**条，存储到了txt文件中，共59337KB，数据存储方式如下： | | | |
| 结论分析与体会：  不知道你们在用爬虫爬数据的时候是否有发现，越来越多的网站都有自己的反爬机制，抓取数据已经不像以前那么容易，目前常见的反爬机制主要有以下几种：  数据是通过动态加载的，比如微博，今日头条，b站  需要登录，需要验证码，比如铁路12306，淘宝，京东  请求次数频繁，IP地址在同一时间访问次数过多，导致IP被封  数据屏蔽方式，比如访问的数据不在源码中，数据隐藏在js中，比如今日分享，b站  爬虫与反爬虫一直以来就像是左右手互博，你有反爬机制，我就有反反爬技术，即见即可爬，道高一尺魔高一丈。 | | | |