Universe Scientific Publishing
Computer System Network and Telecommunications,计算机系统网络和电信(1) 2019,3
ISSN: 2661-3719(Print) ISBN:2661-3727(Online)

基于 Python 的数据爬取系统的设计——以房屋信息爬取为例

毛红霞
四川大学锦城学院 计算机与软件学院 四川 成都 611731

DOI: 10.18686/jsjxt.v1i3.1260

【摘要】互联网技术的广泛应用使得网络资源爆炸式增长,在海量数据中查找所需数据是十分耗时耗力的事情。房屋信息是国民关注的热点话题之一,运用网络爬虫技术,可以快速准确地获取各大平台的房屋信息。本文采用 Python 语言结合爬虫技术设计了房屋信息数据爬取系统,设计了 URL 管理器、网页下载、网页分析、数据采集、数据保存等模块。通过系统的运行,成功地将目标网站上的房屋信息及图片保存下来。

【关键词】Python;数据爬取;反爬策略;

0 引言

随着互联网技术的快速发展,信息技术得到了飞速发展,特别是互联网上的数据资源以巨大的速度迅速增长和积累,网络资源爆炸式增长,在海量数据中快速准确地查找有价值的信息,会越来越困难。因此,网络爬虫应运而生,能够根据自己的需要从目标网站上准确高效地爬取所需数据。数据爬取会给网站带来一些负担,因此不同网站也采取了相应的反爬虫策略,数据爬取系统要不定时分析目标网站进行反爬虫机制的研究,以保证数据爬取系统能够正常运行并爬取到所需数据。

现阶段房屋信息是人民关注的一个重要主题,人们对新房、二手房、租房价格都有极高的关注热情,但各大平台都有数据壁垒不能涵盖所有的住房信息,因此构建一个能够爬取网络上房屋信息的系统显得尤为重要。本文采用 Python 作为数据采集系统的编程语言,爬取网络中的房屋信息。

1 爬虫原理

网络爬虫是一种程序,它的主要目的是将互联网上的网页下载到本地并提取出相关数据。网络爬虫可以自动化地浏览网络中的信息,然后根据指定要求制定的规则下载、提取信所需数据。基础的网络爬虫的架构如图 1 所示。

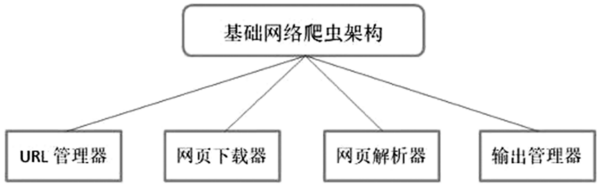

图 1 基础网络爬虫的架构

URL 管理器:管理将要爬取的 URL,防止重复抓取和循环抓取。

网页下载器:这是下载网页的组件,用来将互联网上 URL 对应的网页下载到本地,是爬虫的核心部分之一。

网页解析器:这是解析网页的组件,用来从网页中提取有价值的数据,是爬虫的另一个核心部分。

输出管理器:这是保存信息的组件,用来把解析出来的内容输出到文件或、数据库中。

2 房屋数据爬取系统设计

本文针对 Q 房网进行房屋数据爬取。数据爬取系统将对 Q 房网上各个城市的网站链接、二手房、租房、新房的基本信息进行爬取,并将爬取下来的房屋信息进行保存。

2.1 网页 URL 管理分析

使用谷歌浏览器打开 Q 房网,进入二手房链接,点击位置获取到相应城市的 URL 之后,尝试更换城市,观察发现相应二手房网站 URL 是有规律的。如

ISSN:2661-3719(Print) ISBN:2661-3727(Online)

图 2 及图 3 所示:

图 2 北京地区网址 URL

图 3 佛山地区网址 URL

更换城市,发现该网站网址 URL 设置的规律为:https://城市拼音.qfang.com/sale。

Q 房网站每个城市对应的网页上最多显示 30 个房屋信息。点击下一页,网页 URL 会产生相应的变化,变化的规律为:每个城市进入后的 url 为:http://foshan.qfang.com/sale,

点击第二页 url 为:http://foshan.qfang.com/sale/f2。如图 4 所示。

图 4 佛山地区房源第 2 页

因此第 n 页的 url 的构造规律为:

http://foshan.qfang.com/sale/fn。

由此,Q 房网房屋信息翻页 URL 的代码为:

```
pre_url = 'http://foshan.qfang.com/sale/f'
for x in range(1,11):
    url = pre_url + str(x)
```

2.2 网页下载

Requests 是一个在编写爬虫代码时会用到的一个库。Requests 继承了 urllib2 的所有特性。Requests 支持 HTTP 连接保持和连接池,支持使用 cookie 保持会话,支持文件上传,支持自动确定响应内容的编码,支持国际化的 URL 和 POST 数据自动编码。使用 Requests 下载网页的代码如下:

```
pre_url = 'http://foshan.qfang.com/sale/f'
for x in range(1,11):
    url = pre_url + str(x)
    html = requests.get(url, headers=headers)
```

2.3 网页解析

使用谷歌浏览器的开发者模式,可以定位要爬取的数据在网页源代码中的位置,提取 XPath 路径,如图 5 所示。

图 5 网站房屋信息前端代码

调用 spider 函数获取相应页面的房屋信息,代码如图 6 所示:

```
def spider(url):
    '''爬虫函数'''
    selector = download(url)

    house_list = selector.xpath("//*[@id='cycleListings']/ul/li")
    for house in house_list:
        xiaoqu = house.xpath('div[1]/p[1]/a/text()')[0]
        huxiing = house.xpath('div[1]/p[2]/span[2]/text()')[0]
        mianji = house.xpath('div[1]/p[2]/span[4]/text()')[0]
        weizhi = house.xpath('div[1]/p[3]/span[2]/a[1]/text()')[0]
        zongjia = house.xpath('div[2]/span[1]/text()')[0]
        #构造详情页URL
        house_url = ('http://beijing.qfang.com'
                     + house.xpath('div[1]/p[1]/a/@href')[0])
        sel = download(house_url)
        house_year = sel.xpath("//div[@class='housing-info']/ul/li[2]/div/ul/li[3]/div/text()")[0]
        mortgage_info =  sel.xpath("//div[@class='housing-info']/ul/li[2]/div/ul/li[5]/div/text()")[0]
        #构造要写入文件的数据项
        item = [xiaoqu,huxiing,mianji,weizhi,zongjia,house_year,mortgage_info]
        #写入文件
        data_writer(item)
        print('正在抓取',xiaoqu )
```

图 6 房屋信息爬取代码

2.4 输出保存

将爬取下来的房源信息保存至文件名为 afang_foshan 的 csv 文件中,代码如下图 7 所示。

```
def data_writer(item):
    with open('qfang_foshan.csv', 'a',encoding='utf-8',newline='') as csvfile:
        writer = csv.writer(csvfile)
        writer.writerow(item)
```

图 7 房源信息保存代码

将房屋的图片信息以二进制格式进行保存,代码如下图 8 所示。

```
def image_saver(url,xiaoqu):
    '''
    图片保存函数
    param url: 图片网页URL
    param xiaoqu: 图片小区名称
    :return: 无
    '''
    img = requests.get(url,headers = headers)
    with open('./Qfang_image/{}.jpg'.format(xiaoqu),'wb') as f:
        f.write(img.content)
```

图 8 房源图片保存代码

3 应对反爬策略

网站通常会对网络爬虫做一些反爬的机制,最常见的反爬技术就是通过 Headers 反爬虫和基于用户行为反爬虫。

从用户请求的 Headers 反爬虫是最常见的反爬虫策略。很多网站都会对 Headers 的 user-agent

ISSN: 2661-3719(Print) ISBN:2661-3727(Online)

(用户代理)进行检测。针对这种反爬虫机制,本文在设计爬虫代码时,在爬虫代码中添加 Headers,将浏览器的 user-agent 赋值到爬虫的 Headers 中,使网站服务器能够识别客户使用的操作系统及版本、CPU 类型、浏览器及版本、浏览器渲染引擎、浏览器语言、浏览器插件等,从而来应对反爬虫策略。

本文爬取的网站通过检测用户行为,如同一账号短时间进行多次相同操作来判断是否为爬虫。针对这种情况,在爬虫代码中控制每次请求后随机间隔几秒再进行下一次请求。

4 结束语

本文研究的是使用 Python 语言进行数据爬取系统的设计,通过爬虫的相关技术和代码的实现,对 Q 房网站的房屋数据进行爬取。从网页 URL 管理,到网页下载、网页分析、数据提取、数据保存、图片保存、应对网站反爬机制的处理,实现了数据爬取系统的设计。

【参考文献】

[1]刘宇,郑成焕. 基于 Scrapy 的深层网络爬虫研究[J]. 软件,2017,38(07):111-114

[2]严斐,肖璞. Python 框架下基于主题的数据爬取技术研究与实现[J]. 计算机时代,2018(11):10-13

[3]刘贵平,刘娜,段红义. 基于聚焦网络爬虫技术的人才招聘数据采集[J]. 电脑编程技巧与维护,2018(05):69-71

[4]刘顺程,岳思颖. 大数据时代下基于 Python 的网络信息爬取技术[J]. 电子技术局与软件工程,2017(21):160-160

[5]李琳. 基于 Python 的网络爬虫系统的设计与实现[J]. 信息通信,2017(9):26-27