如何爬取看雪学院的课程

原创

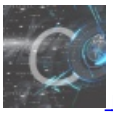
~巴哥~ 于 2021-03-16 21:21:18 发布 208 收藏

分类专栏： 爬虫 python 文章标签： python 爬虫


爬虫 同时被 2 个专栏收录

9 篇文章 1 订阅

订阅专栏


python

7 篇文章 0 订阅

订阅专栏

看雪学院是看雪论坛（https://www.kanxue.com/）下面的一个mooc平台，上面有一些性价比很高的安全课程。
下面介绍如何使用python来爬取看雪学院上已购买的课程。

环境

- window 7 x64
- python3.7 (Anaconda 3)
- vscode
  使用python包有
- requests 模拟http请求
- bs4 解析html文档
- re 正则表达式库

代码实现

这个爬虫很简单，直接上代码。

```
#-*- coding:utf-8 -*-
# kanxue_spider.py
#使用方法 python kanxue_spider.py [课程章节页url]
#如python kanxue_spider.py https://www.kanxue.com/book-section_list-40.htm
import requests
from requests.packages import urllib3
import os,re
import hashlib
from bs4 import BeautifulSoup

#全局变量
#requests库使用headers
headers = {
    'User-Agent':'Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:71.0) Gecko/20100101 Firefox/71.0',
    'Referer':'https://www.kanxue.com/'
```

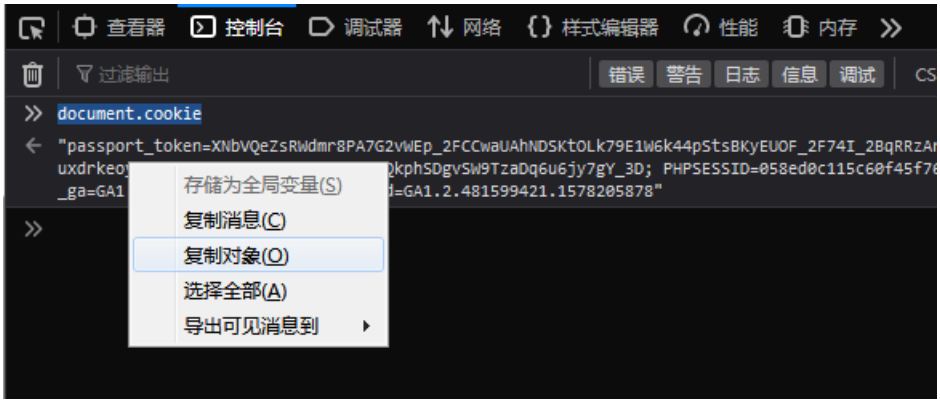
```
    'Referer' : 'https://www.kanxue.com/'
}

home = 'https://www.kanxue.com/'
#cookies,如何得到见下文
cookies_str = "xxxxx"
s = requests.Session()
cookies={}

#根据url下载视频，保存为filename
def download_video(video_url,filename):

    headers1 = {
        'Referer':'https://www.kanxue.com/',
        'Host':'video.kanxue.com',
        'Accept':'video/webm,video/ogg,video/*;q=0.9,application/ogg;q=0.7,audio/*;q=0.6,*/*;q=0.5',
        'User-Agent':'Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:72.0) Gecko/20100101 Firefox/72.0'
    }
    res1 = s.get(video_url, headers=headers1,cookies=cookies,stream=True)
    with open(filename, "wb") as f:
        for chunk in res1.iter_content(chunk_size=1024):
            if chunk:
                f.write(chunk)
    print(filename,'下载完毕')

#生成x的md5值
def md5(x):
    m = hashlib.md5()
    m.update(x.encode('utf-8'))
    return m.hexdigest()

#根据uri返回完成的url
def get_complete_url(x):
    url = ''
    if x.startswith('http'):
        url = x
    elif x.startswith('//'):
        url = 'https:'+x
    elif x.startswith('/'):
        url = home[:-1]+x
    else:
        url = home + x
    return url

#下载网课的web页面，因为其中含有一些文档
def download_html(html,filename):

    #解析html文档
    soup = BeautifulSoup(html,'lxml')
    data = html

    #静态文件的存储目录
    static_files_path = 'data/static'
    #该目录不存在的话，将其创建
    if not os.path.isdir(static_files_path):
        os.mkdir(static_files_path)

    #爬取页面中的css文件
    for tag in soup.find_all('link'):
        css_url = get_complete_url(tag['href'])
```

```
        #print(css_url)
        target_filename = '%s\\%s' % (static_files_path,md5(css_url))
        if not os.path.isfile(target_filename):
            with open(target_filename,'w',encoding='utf-8') as f:
                r = s.get(css_url,headers=headers,verify=False,cookies=cookies)
                f.write(r.text)
        #将原始页面中的路径替换成本地的相对路径
        data = data.replace(tag['href'],'../../static/%s' % md5(css_url))

    #爬取页面中的js文件
    for tag in soup.find_all('img'):
        img_url = get_complete_url(tag['src'])
        #print(img_url)
        target_filename = '%s\\%s' % (static_files_path,md5(img_url))
        if not os.path.isfile(target_filename):
            with open(target_filename,'wb') as f:
                r = s.get(img_url,headers=headers,verify=False,cookies=cookies)
                f.write(r.content)
        #将原始页面中的路径替换成本地的相对路径
        data = data.replace(tag['src'],'../../static/%s' % md5(img_url))

    #爬取页面中的js文件
    for tag in soup.find_all('script'):
        if not tag.get('src',''):
            continue
        js_url = get_complete_url(tag['src'])
        #print(js_url)
        target_filename = '%s\\%s' % (static_files_path,md5(js_url))
        if not os.path.isfile(target_filename):
            with open(target_filename,'w',encoding='utf-8') as f:
                r = s.get(js_url,headers=headers,verify=False,cookies=cookies)

                f.write(r.text)
        #将原始页面中的路径替换成本地的相对路径
        data = data.replace(tag['src'],'../../static/%s' % md5(js_url))

    with open('%s.html'%filename,'w',encoding='utf-8') as f:
        f.write(data)
        print('%s.html'%filename,'下载完毕')

#爬取课时内容,
#keshi_url为课时的url, filename为课时的名字
def spider_keshi(filename,keshi_url):
    res = s.get(keshi_url,headers = headers,cookies= cookies,verify=False)
    if res.status_code ==  200:
        #使用正则表达式找到视频的地址
        x = re.findall(r'value="(https://video.kanxue.com/.*?)"',res.text)
        if x:
            video_url = x[0]
            #print(video_url)
            video_filename = '%s.mp4'%filename
            #若视频文件不存在
            if not os.path.isfile(video_filename):
                #下载视频
                download_video(video_url,video_filename)
        #下载课时的页面
        download_html(res.text,filename)

#爬虫主函数
def main():
```

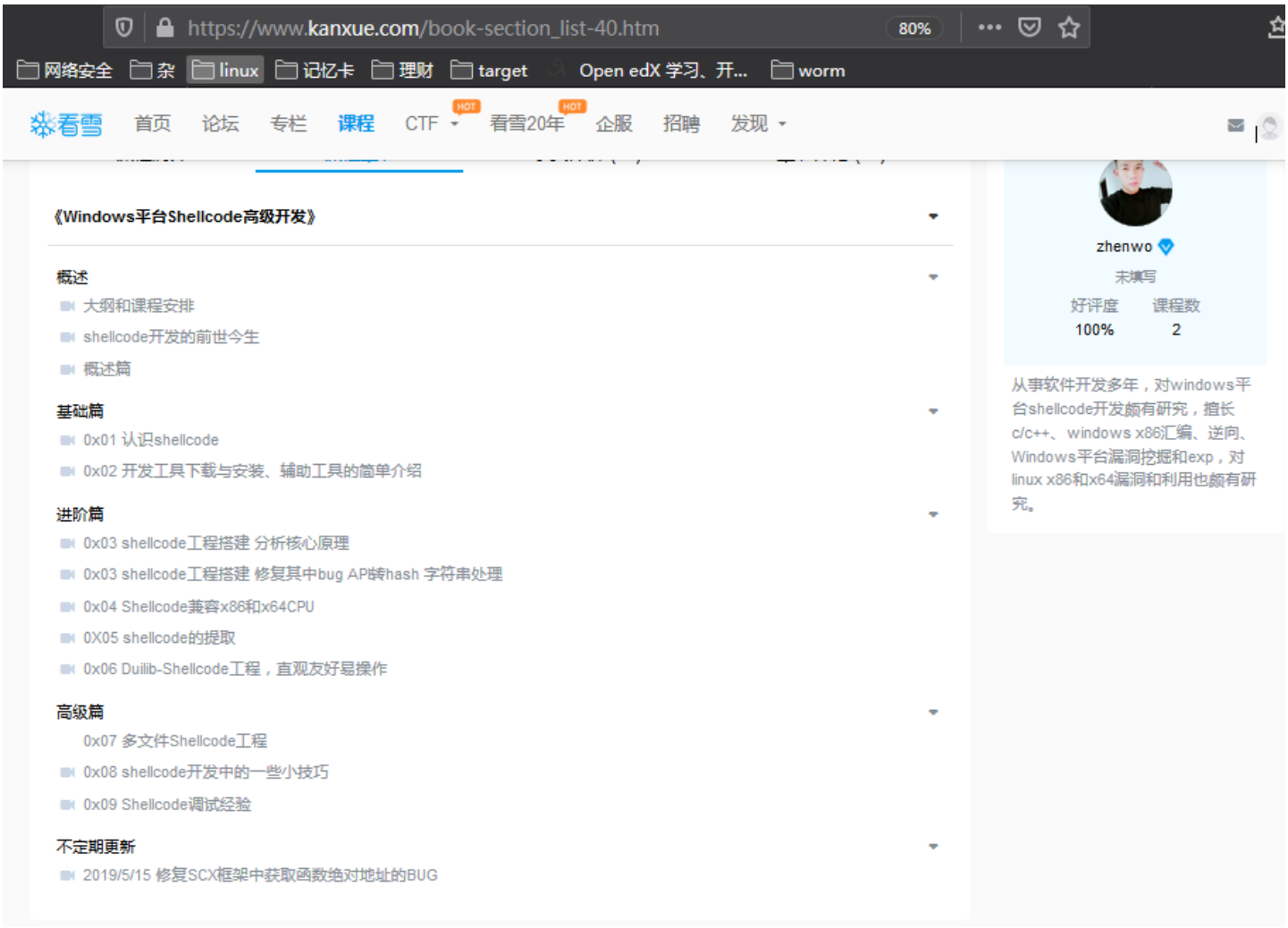
```
def main():
    #声明全局变量
    global headers,cookies

    #加载cookies
    for item in cookies_str.split(';'):
        k,v = item.strip().split('=')
        cookies[k] = v
    res = s.get(url=home,headers=headers,cookies=cookies)

    res1 = s.get(
        sys.argv[1],
        headers = headers,
        cookies = cookies,
        verify = False
    )
    if res1.status_code == 200:
        #解析html文档
        soup = BeautifulSoup(res1.text,'lxml').find('div',id='section_list')
        course_name = soup.b.text.strip()
        for li in soup.find_all('li',class_='mb-1'):
            #章的名称
            zhang_tilte = li.span.text.strip()
            #print(zhang_tilte)
            #构造存储路径
            path = 'data\\%s\\%s' % (course_name,zhang_tilte)
            #若该路径不存在，将其创建
            if not os.path.isdir(path):
                os.makedirs(path)
            #找到所有的课时
            for a in li.find_all('a'):
                #课时的title
                keshi_title = a['title'].replace('/','_')
                #课时完整的url
                keshi_url = 'https://www.kanxue.com/'+a['href']
                #爬取课时内容
                spider_keshi('%s\\%s' % (path,keshi_title),keshi_url)


main()
```

如何使用这个代码

首先需要使用Firefox或Chrome，用用户名密码登陆看雪学院，按F12打开开发者工具，在控制台中输入document.cookie，右键复制对象，将cookies复制赋值给代码中的全局变量cookies_str



找到需要下载的课程，在课程主页中打开课程章节，如下图



这个url为脚本的第1个参数

参考资料

- requests如何下载二进制流文件
  https://2.python-requests.org//zh_CN/latest/user/quickstart.html#id5