

2019

Homework





CONTENTS

This Is A List

01

豆瓣top250爬虫

02

Django网站介绍

03

利用selenium动态爬取



Part One



YIDENG

0 | 九月

```
def get_html(url):  
    try:  
        response=requests.get(url,allow_redirects=False)  
        if response.status_code==200:  
            html = response.text.encode(response.encoding).decode('utf-8', 'ignore')  
            return html  
        if response.status_code==302:  
            print('302')  
    except ConnectionError:  
        return get_html(url)
```

这里首先要导入requests库来获取到网页的源代码，因为考虑到有些网页会有重定向，所以这里有用到allow_redirects，为了判断响应是否成功，所以这里用if来判断网页的状态码，获取到的html还是不能获取，发现是编码的问题，于是对获取到的网页进行编码



0 | 十月

```
def get_info():  
    html = get_html("https://movie.douban.com/top250")  
    soup=BeautifulSoup(html,"lxml")  
    info1=soup.find_all('div', {'class': "hd"})  
    info2=soup.find_all('p', {'class': ""})  
    print(str(info2[0]).replace("\n",""))  
    info4=soup.find_all('img', {'class': ""})  
    info3 = soup.find_all('div', {'class': "star"})
```

前面我们已经获得到了html源代码，这里要对源代码进行解析，于是我们用到Beautifulsoup库，导入方法的是from bs4 import BeautifulSoup，然后我们用find_all方法找到我们需要数据的div块等等，然后用正则进行匹配出我们所需要的数据.正则的代码看下图



01

```
for i in range(0,25):
    dict["id"]=i+1
    dict["图片链接"]=re.findall(r'alt=".*?" class="" src="(.*?)" width=".*?"',str(info4[i]))[0]
    dict["电影名"]=re.findall(r'<span class="title">(.*?)</span>', str(info1[i]))[0]
    dict["类型"]=re.findall(r'^(p class="">.*?<br/>.*?/.*?/(.*?)</p>$',str(info2[i]).replace("\n",""))
    dict["影片详细链接"]=re.findall(r'<a class="" href="(.*?)">',str(info1[i]))
    dict["评分"]=re.findall(r'<span class="rating_num" property="v:average">(.*?)</span>',
                            str(info3[i]))
    dict["主演导演"]=re.findall(r'^(p class="">\s+(.*?)<br/>',str(info2[i]).replace("\n",""))
    bb=str(dict["主演导演"]).replace(' xa0','').replace("\\","").replace(" ","")
    dict["主演导演"]=bb
    aa=str(dict["类型"]).replace(' xa0','').replace("\\","").replace(" ","")
    dict["类型"] =aa
```

我们这里有用到**replace**方法是因为我们获取到的数据不是在同一行，然后为了美观进行的简单的数据处理最后存到字典中。



01

```
def save_mysqlinfo():  
    conn = pymysql.connect(  
        host='localhost',  
        port=3306,  
        user='root',  
        passwd='admin',  
        db='douban',  
    )  
    cur = conn.cursor()
```

```
def save_info():  
    conn = pymongo.MongoClient('localhost', 27017)  
    db = conn.moviedb  
    my_set = db.test_set  
    my_set.insert({'电影名': dict['电影名'], '主演导演': dict['主演导演'],  
                  '评分': dict['评分']})
```

```
sql = "INSERT INTO `douban_movie` (id, moviename, actoranddirector, grade, movieimg, movieurl, type) values (%s, %s, %s, %s, %s, %s, %s)"  
cur.execute(sql, [dict["id"], dict['电影名'], str(dict["主演导演"]).replace(' ', ' ').replace(']', ' '),  
                  str(dict["评分"]).replace(' ', ' ').replace(']', ' '),  
                  str(dict["图片链接"]),  
                  str(dict["影片详细链接"]).replace(' ', ' ').replace(']', ' ').replace('"', ''),  
                  str(dict["类型"]).replace(' ', ' ').replace(']', ' ').replace('"', '')])  
conn.commit()  
cur.close()  
conn.close()
```

这里是两个不同的数据库存储方法，一个是mysql的一个是mongodb的，个人还是喜欢mongodb的比较方便也不繁琐，由于django的要求于是使用了mysql的，最后使用get_info调用后爬取



Part Two

Add your text here



YIDENG

02 Django配置

众所周知，我们在进行django写网页是需要一大堆的setting配置等等这里贴出我们所需要的配置。

```
INSTALLED_APPS = [  
    'django.contrib.admin',  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.messages',  
    'django.contrib.staticfiles',  
    'douban.apps.DoubanConfig',  
]
```

```
LANGUAGE_CODE = 'zh-hans'
```

```
DATABASES = {  
    'default': {  
        'ENGINE': 'django.db.backends.mysql',  
        'NAME': 'douban',           # 你要存储数  
        'USER': 'root',           # 数据库用户名  
        'PASSWORD': 'admin',      # 密码  
        'HOST': 'localhost',      # 主机  
        'PORT': '3306',           # 数据库使用的  
    }  
}
```

```
import pymysql  
pymysql.install_as_MySQLdb()
```



02 页面功能详情

登录

注册

查找

详情

02

这边登录首先要有账户，账户我们存在数据库中，如果没有，我们将不能登录

我们数据库中是没有shiqikai这个用户名的，当我们点击登录时会跳转到错误界面，这里我们错误界面只显示bbb代表跳转到错误界面后续可以进行美化等工作

登录成功后我们会跳转到搜索电影页面



用户名

密 码

用户名

密 码

bbb

02

当我们没有用户的时候我们就需要进行注册用户，这里我们点击注册会跳转到注册页面

我们填写所需要的用户名和密码，成功后会增加到数据库中，之后便可以在登录的时候成功登陆这里我们注册了admin用户，密码为123



注册新用户

* 用户名:

* 登录密码:

* 确认密码:

* 验证码:

☐ 我已阅读并同意《用户注册协议》

同意以上协议并注册

Part Three

Add your text here

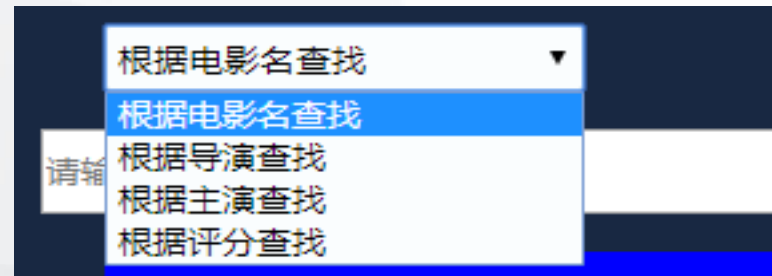


YIDENG

03

A dark blue rectangular search interface. At the top, there is a white dropdown menu with the text '根据电影名查找' and a downward arrow. Below this, on the left, is the text '请输入关键字' in a light blue font. To its right is a white input field with the placeholder text '请输入内容'. At the bottom center, there is a large blue button with the white text '查找'.

登录成功后我们会跳转到电影搜索页面，输入我们想要的电影名，评分，导演都可以搜索到我们想要的电影

A close-up of the search dropdown menu. The menu is open, showing a list of search criteria. The first item, '根据电影名查找', is highlighted with a blue background. Below it are three other items: '根据导演查找', '根据主演查找', and '根据评分查找'. To the left of the dropdown, the text '请输入' is visible in a light blue font.

Part Four

Add your text here



YIDENG

04

请输入关键字 肖申克的救赎



1 点击获得详细内容

我们通过搜索肖申克的救赎得到了肖申克救赎电影的信息



YIDENG

04

我们通过触碰点击获得详细内容，我们就会得到电影的详细内容，这里用到了onmouseover的js方法



1
点击获得详细内容

肖申克的救赎

评分: '9.6'

'导演:弗兰克·德拉邦特
FrankDarabont主演:蒂姆·罗宾斯
TimRobbins/...'
犯罪剧情



1
点击获得详细内容



2
点击获得详细内容

霸王别姬

评分: '9.6'

'导演:陈凯歌KaigeChen
主演:张国荣
LeslieCheung/张丰毅
FengyiZha...'
剧情爱情同性

这里是根据评分查找的电影我们查找的是评分9.6的

04

```
import requests
from selenium import webdriver
from selenium.webdriver.support import expected_conditions as EC
from selenium.webdriver.support.wait import WebDriverWait
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.support.select import Select
import re
from bs4 import BeautifulSoup
browser=webdriver.Chrome(service_args=['--ignore-ssl-errors=true', '--ssl-protocol=TLSv1'])
wait=WebDriverWait(browser, 10)
selector="#sjina_C"
selectorl="_02 > a"
url4=[]
price=[]
```



04

```
def get_url(url):  
    # d 2 blank lines, found 0  
    nge(20, 40):  
        browser.get("http://hz.newhouse.fang.com/house/s/?ctm=1.hz.xf_search.list_type.1")  
        wait.until(EC.element_to_be_clickable((By.CSS_SELECTOR, selector+str(i)+selector1))).click()  
        normal_window = browser.current_window_handle  
        all_Handles = browser.window_handles  
        for pay_window in all_Handles:  
            if pay_window != normal_window:  
                browser.switch_to_window(pay_window)  
        soup = BeautifulSoup(browser.page_source, "lxml")  
        url1=soup.find("div", id='originalNaviBox')  
        url2=re.findall(r'<a href="(.*?)" id="xf.*?_B03_08" target="_self">楼盘详情</a>', str(url1))  
        url3="".join(url2)  
        url4.append(url3)  
    return url4
```



04

```
def get_html(url):  
    try:  
        response=requests.get(url)  
        if response.status_code==200:  
            html = response.text.encode(response.encoding).decode('gb2312','ignore')  
            return html  
        if response.status_code==302:  
            print('302')  
    except ConnectionError:  
        return get_html(url)
```



04

```
def main():  
    new_url= get_url("http://hz.newhouse.fang.com/house/s/?ctm=1.hz.xf_search.list_type.1")  
    for i in range(len(new_url)):  
        try:  
            html=get_html(new_url[i])  
            soup=BeautifulSoup(html,"lxml")  
            lab=soup.find_all('div',attrs=('class','main-info-price'))  
            price.append(re.findall(r'<em>\s+(.*?)\s+</em>',str(lab)))  
        except:  
            continue  
    print(price)  
  
main()
```



04

以上是利用selenium爬取房天下楼盘数据的，因为这里我们要点击才能获得他的数据所以我们要用到selenium动态爬取，其中因为要转化窗口，所以会用到下面的代码

```
normal_window = browser.current_window_handle
all_Handles = browser.window_handles
for pay_window in all_Handles:
    if pay_window != normal_window:
        browser.switch_to_window(pay_window)
```





Thanks

For Your Watching



YIDENG