Day04回顾

requests.get()参数

```
1、url
2、params -> {} : 查询参数 Query String
3、proxies -> {}
   proxies = {
      'http':'http://1.1.1.1:8888',
      'https':'https://1.1.1.1:8888'
   }
4、auth -> ('tarenacode','code_2013')
5、verify -> True/False
6、timeout
```

requests.post()

```
data -> {} Form表单数据 : Form Data
```

控制台抓包

- 打开方式及常用选项

```
1、打开浏览器，F12打开控制台，找到Network选项卡
2、控制台常用选项
   1、Network：抓取网络数据包
        1、ALL：抓取所有的网络数据包
        2、XHR：抓取异步加载的网络数据包
        3、JS ：抓取所有的JS文件
   2、Sources：格式化输出并打断点调试JavaScript代码，助于分析爬虫中一些参数
   3、Console：交互模式，可对JavaScript中的代码进行测试
3、抓取具体网络数据包后
   1、单击左侧网络数据包地址，进入数据包详情，查看右侧
   2、右侧：
      1、Headers：整个请求信息
           General、Response Headers、Request Headers、Query String、Form Data
      2、Preview：对响应内容进行预览
      3、Response：响应内容
      4、Cookies：请求及响应中Cookies信息
```

- 有道翻译过程梳理

1.
```
1. 打开首页
2. 准备抓包：F12开启控制台
3. 寻找地址
   页面中输入翻译单词，控制台中抓取到网络数据包，查找并分析返回翻译数据的地址
4. 发现规律
   找到返回具体数据的地址，在页面中多输入几个单词，找到对应URL地址，分析对比 Network - All(或者XHR) - Form Data，发现对应的规律
5. 寻找JS文件
   右上角 ... -> Search -> 搜索关键字 -> 单击 -> 跳转到Sources，左下角格式化符号{}
6、查看JS代码
   搜索关键字，找到相关加密方法
7、断点调试
8、完善程序
```

常见的反爬机制及处理方式

```
1、Headers反爬虫 ： Cookie、Referer、User-Agent
   解决方案：通过F12获取headers,传给requests.get()方法

2、IP限制 ： 网站根据IP地址访问频率进行反爬,短时间内进制IP访问
   解决方案：
        1、构造自己IP代理池,每次访问随机选择代理,经常更新代理池
        2、购买开放代理或私密代理IP
        3、降低爬取的速度

3、User-Agent限制 ： 类似于IP限制
   解决方案：构造自己的User-Agent池,每次访问随机选择

4、Ajax动态加载 ： 从url加载网页的源代码后,会在浏览器执行JavaScript程序,这些程序会加载更多内容
   解决方案：F12或抓包工具抓包处理

5、对查询参数加密
   解决发难：找到JS文件,分析加密算法,用Python实现加密执行JS文件中的代码,返回加密数据

6、对响应内容做处理
   解决方案：打印并查看响应内容,用xpath或正则做处理
```

python中正则处理headers和formdata

```
1、pycharm进入方法 ： Ctrl + r
2、处理headers和formdata
  (.*?): (.*)
  "$1": "$2",
3、点击 Replace All
```

Day05笔记

动态加载数据抓取-Ajax

- 特点

```
1、右键 -> 查看网页源码中没有具体数据
2、滚动鼠标滑轮或其他动作时加载
```

- 抓取

```
1、F12打开控制台，页面动作抓取网络数据包
2、抓取json文件URL地址
# 控制台中 XHR ： 异步加载的数据包
# XHR -> Query String(查询参数)
```

豆瓣电影数据抓取案例

- 目标

```
1、地址：豆瓣电影 - 排行榜 - 剧情
2、目标：电影名称、电影评分
```

- F12抓包（XHR）

```
1、Request URL(基准URL地址) ： https://movie.douban.com/j/chart/top_list?
2、Query String(查询参数)
# 抓取的查询参数如下：
type: 13
interval_id: 100:90
action: ''
start: 0
limit: 用户输入的电影数量
```

- json模块的使用

```
1、json.loads(json格式的字符串)：把json格式的字符串转为python数据类型
# 示例
html = json.loads(res.text)
print(type(html))
```

- 代码实现

```

```

思考: 实现用户在终端输入电影类型和电影数量，将对应电影信息抓取到数据库

```
1
```

练习: 腾讯招聘案例抓包看看?

- URL地址及目标

1. 确定URL地址及目标

```
1、URL：百度搜索腾讯招聘 - 查看工作岗位
2、目标：职位名称、工作职责、岗位要求
```

2. F12抓包
3. 一级页面json地址(index变,timestamp未检查)

```
https://careers.tencent.com/tencentcareer/api/post/Query?timestamp=1559294378106&countryId=&cityId=&bgIds=&productId=&categoryId=&parentCategoryId=&attrId=&keyword=&pageIndex={}&pageSize=10&language=zh-cn&area=cn
```

4. 二级页面地址(postId在变,在一级页面中可拿到)

```
https://careers.tencent.com/tencentcareer/api/post/ByPostId?timestamp=1559&postId={}&language=zh-cn
```

- 具体代码实现

```
1
```

cookie模拟登录

- 适用网站及场景

```
抓取需要登录才能访问的页面
```

- **方法一**

```
1、先登录成功1次,获取到携带登陆信息的Cookie
   F12打开控制台,在页面输入用户名、密码,登录成功,找到/home(一般在抓到地址的上面)
2、携带着cookie发请求
   ** Cookie
   ** Referer(源,代表你从哪里转过来的)
   ** User-Agent
```

```
1
```

- **方法二**

1. 知识点

```
利用requests模块中的session会话保持功能
```

2. session会话使用流程

```
1、实例化session对象
   session = requests.session()
2、让session对象发送get或者post请求
   res = session.get(url,headers=headers)
```

3. 具体步骤

```
1、寻找登录时POST的地址
   查看网页源码,查看form,找action对应的地址: http://www.renren.com/PLogin.do

2、发送用户名和密码信息到POST的地址
   * 用户名和密码信息以什么方式发送?  -- 字典
     键 : <input>标签中name的值(email,password)
     值 : 真实的用户名和密码
     post_data = {'email':'','password':''}
```

4. 程序实现

```
整体思路
1、先POST: 把用户名和密码信息POST到某个地址中
2、再GET:  正常请求去获取页面信息
```

```

```

selenium+phantomjs/Chrome/Firefox

selenium

- 定义

```
1、Web自动化测试工具，可运行在浏览器,根据指令操作浏览器
2、只是工具，必须与第三方浏览器结合使用
```

- 安装

```
Linux: sudo pip3 install selenium
Windows: python -m pip install selenium
```

phantomjs浏览器

- 定义

```
无界面浏览器(又称无头浏览器)，在内存中进行页面加载,高效
```

- 安装(phantomjs、chromedriver、geckodriver)

Windows

```
1、下载对应版本的phantomjs、chromedriver、geckodriver
2、把chromedriver.exe拷贝到python安装目录的Scripts目录下(添加到系统环境变量)
   查看python安装路径：where python
3、验证
   cmd命令行：chromedriver

# 下载地址
chromedriver ：下载对应版本
http://chromedriver.storage.googleapis.com/index.html

geckodriver
https://github.com/mozilla/geckodriver/releases
```

Linux

```
1、下载后解压
   tar -zxvf geckodriver.tar.gz
2、拷贝解压后文件到 /usr/bin/  (添加环境变量)
   sudo cp geckodriver /usr/bin/
```

- 使用

示例代码一：使用 selenium+浏览器 打开百度

```

```

示例代码二：打开百度，搜索赵丽颖，查看

```

```

- 浏览器对象(browser)方法

```
1、browser = webdriver.Firefox(executable_path='path')
2、browser.get(url)
3、browser.page_source # 查看响应内容
4、browser.page_source.find('字符串')
   # 从html源码中搜索指定字符串,没有找到返回：-1
5、browser.quit() # 关闭浏览器
```

- 定位节点

单元素查找(1个节点对象)

```
1、browser.find_element_by_id('')
2、browser.find_element_by_name('')
3、browser.find_element_by_class_name('')
4、browser.find_element_by_xpath('')
... ...
```

多元素查找([节点对象列表])

```
1、browser.find_elements_by_id('')
2、browser.find_elements_by_name('')
3、browser.find_elements_by_class_name('')
4、browser.find_elements_by_xpath('')
... ...
```

- 节点对象操作

```
1、ele.send_keys('') # 搜索框发送内容
2、ele.click()
3、ele.text          # 获取文本内容
4、ele.get_attribute('src') # 获取属性值
```

今日作业

作业1: 哔哩哔哩小视频下载

```
哔哩哔哩小视频下载
# 1、url ： http://vc.bilibili.com/p/eden/rank#/?tab=全部
# 2、抓取目标 ： 所有异步加载的小视频
```

作业2：京东商品数据抓取

```
1、网址 ： https://www.jd.com/
2、目标：名称、价格、评价、商家
3、思路
    跳到商品页面后,匹配所有商品节点对象列表
    把节点对象的文本内容拿出来,想办法处理
4、下一页(browser.page_source...)
    **********time.sleep()**********
```