Day05回顾

动态加载网站数据抓取

```
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)
```

有道翻译流程梳理

```
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，分析并用python实现
7、断点调试
8、完善程序
```

cookie模拟登陆

```
1、适用网站类型：爬取网站页面时需要登录后才能访问，否则获取不到页面的实际响应数据
2、方法1（利用cookie）
   1、先登录成功1次,获取到携带登陆信息的Cookie（处理headers）
   2、利用处理的headers向URL地址发请求
3、方法2（利用session会话保持）
   1、登陆，找到POST地址：form -> action对应地址
   2、定义字典，创建session实例发送请求
      # 字典key ：<input>标签中name的值(email,password)
      # post_data = {'email':'','password':''}
```

Day06笔记

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
```

```

```

- **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
```

```

```

百度翻译破解案例

目标

```
破解百度翻译接口，抓取翻译结果数据
```

实现步骤

- **1、F12抓包,找到json的地址,观察查询参数**

```
1、POST地址：https://fanyi.baidu.com/v2transapi
2、Form表单数据（多次抓取在变的字段）
   from: zh
   to: en
   sign: 54706.276099  #这个是如何生成的?
   token: a927248ae7146c842bb4a94457ca35ee # 基本固定,但也想办法获取
```

- **2、抓取相关JS文件**

```
右上角 - 搜索 - sign: - 找到具体JS文件(index_c8a141d.js) - 格式化输出
```

3、在JS中寻找sign的生成代码

```
1、在格式化输出的JS代码中搜索：sign: 找到如下JS代码：sign: m(a),
2、通过设置断点，找到m(a)函数的位置，即生成sign的具体函数
   # 1. a 为要翻译的单词
   # 2. 鼠标移动到 m(a) 位置处，点击可进入具体m(a)函数代码块
```

4、生成sign的m(a)函数具体代码如下(在一个大的define中)

```
function a(r) {
        if (Array.isArray(r)) {
            for (var o = 0, t = Array(r.length); o < r.length; o++)
                t[o] = r[o];
            return t
        }
        return Array.from(r)
    }
function n(r, o) {
    for (var t = 0; t < o.length - 2; t += 3) {
        var a = o.charAt(t + 2);
        a = a >= "a" ? a.charCodeAt(0) - 87 : Number(a),
            a = "+" === o.charAt(t + 1) ? r >>> a : r << a,
```

```
            r = "+" === o.charAt(t) ? r + a & 4294967295 : r ^ a
    }
    return r
}
function e(r) {
    var o = r.match(/[\uD800-\uDBFF][\uDC00-\uDFFF]/g);
    if (null === o) {
        var t = r.length;
        t > 30 && (r = "" + r.substr(0, 10) + r.substr(Math.floor(t / 2) - 5, 10) +
r.substr(-10, 10))
    } else {
        for (var e = r.split(/[\uD800-\uDBFF][\uDC00-\uDFFF]/), C = 0, h = e.length, f = []; h
> C; C++)
            "" !== e[C] && f.push.apply(f, a(e[C].split(""))),
                C !== h - 1 && f.push(o[C]);
        var g = f.length;
        g > 30 && (r = f.slice(0, 10).join("") + f.slice(Math.floor(g / 2) - 5, Math.floor(g /
2) + 5).join("") + f.slice(-10).join(""))
    }
//    var u = void 0
//    , l = "" + String.fromCharCode(103) + String.fromCharCode(116) +
String.fromCharCode(107);
//    u = null !== i ? i : (i = window[l] || "") || "";
//  断点调试,然后从网页源码中找到 window.gtk的值
    var u = '320305.131321201'

    for (var d = u.split("."), m = Number(d[0]) || 0, s = Number(d[1]) || 0, S = [], c = 0, v
= 0; v < r.length; v++) {
        var A = r.charCodeAt(v);
        128 > A ? S[c++] = A : (2048 > A ? S[c++] = A >> 6 | 192 : (55296 === (64512 & A) && v
+ 1 < r.length && 56320 === (64512 & r.charCodeAt(v + 1)) ? (A = 65536 + ((1023 & A) << 10) +
(1023 & r.charCodeAt(++v)),
            S[c++] = A >> 18 | 240,
            S[c++] = A >> 12 & 63 | 128) : S[c++] = A >> 12 | 224,
                                                                    S[c++] = A >> 6 & 63 |
128),
                                S[c++] = 63 & A | 128)
    }
    for (var p = m, F = "" + String.fromCharCode(43) + String.fromCharCode(45) +
String.fromCharCode(97) + ("" + String.fromCharCode(94) + String.fromCharCode(43) +
String.fromCharCode(54)), D = "" + String.fromCharCode(43) + String.fromCharCode(45) +
String.fromCharCode(51) + ("" + String.fromCharCode(94) + String.fromCharCode(43) +
String.fromCharCode(98)) + ("" + String.fromCharCode(43) + String.fromCharCode(45) +
String.fromCharCode(102)), b = 0; b < S.length; b++)
        p += S[b],
            p = n(p, F);
    return p = n(p, D),
        p ^= s,
        0 > p && (p = (2147483647 & p) + 2147483648),
        p %= 1e6,
        p.toString() + "." + (p ^ m)
}
var i = null;
//此行报错，直接注释掉即可
//t.exports = e
```

- **5、直接将代码写入本地js文件,利用pyexecjs模块执行js代码进行调试**

```
1
```

- **获取token**

```
# 在js中
token: window.common.token
# 在响应中想办法获取此值
token_url = 'https://fanyi.baidu.com/?aldtype=16047'
regex: "token: '(.*?)'"
```

- **具体代码实现**

```
1
```

民政部网站数据抓取

目标

```
1、URL: http://www.mca.gov.cn/ - 民政数据 - 行政区划代码
   即: http://www.mca.gov.cn/article/sj/xzqh/2019/
2、目标: 抓取最新中华人民共和国县以上行政区划代码
```

实现步骤

- **1、从民政数据网站中提取最新行政区划代码**

```
# 特点
1、最新的在上面
2、命名格式: 2019年X月中华人民共和国县以上行政区划代码
```

- **2、从二级页面链接中提取真实链接（反爬-响应内容中嵌入JS，指向新的链接）**

```
1、向二级页面链接发请求得到响应内容，并查看嵌入的JS代码
2、正则提取真实的二级页面链接
```

- **3、在数据库表中查询此条链接是否已经爬取，建立增量爬虫**

```
1、数据库中建立version表，存储爬取的链接
2、每次执行程序和version表中记录核对，查看是否已经爬取过
```

- **4、代码实现**

```
1
```

多线程爬虫

应用场景

```
1、多进程 ： CPU密集程序
2、多线程 ： 爬虫(网络I/O)、本地磁盘I/O
```

知识点回顾

- 队列

```
# 导入模块
from queue import Queue
# 使用
q = Queue()
q.put(url)
q.get() # 当队列为空时，阻塞
q.empty() # 判断队列是否为空，True/False
```

- 线程模块

```
# 导入模块
from threading import Thread

# 使用流程
t = Thread(target=函数名) # 创建线程对象
t.start() # 创建并启动线程
t.join()  # 阻塞等待回收线程

# 如何创建多线程，如下方法你觉得怎么样？？？？？
for i in range(5):
    t = Thread(target=函数名)
    t.start()
    t.join()
```

小米应用商店抓取(多线程)

- 目标

```
1、网址 ： 百度搜 - 小米应用商店，进入官网
2、目标 ： 应用分类 - 聊天社交
   应用名称
   应用链接
```

- 实现步骤

1. 确认是否为动态加载

```
1、页面局部刷新
2、右键查看网页源代码，搜索关键字未搜到
# 此网站为动态加载网站，需要抓取网络数据包分析
```

2. F12抓取网络数据包

```
1、抓取返回json数据的URL地址（Headers中的Request URL）
   http://app.mi.com/categotyAllListApi?page={}&categoryId=2&pageSize=30

2、查看并分析查询参数（headers中的Query String Parameters）
   page: 1
   categoryId: 2
   pageSize: 30
   # 只有page再变，0 1 2 3 ... ... ，这样我们就可以通过控制page的直拼接多个返回json数据的URL地址
```

- 代码实现

```

```

json解析模块

json.loads(json格式字符串)

- 作用

```
把json格式的字符串转为Python数据类型
```

- 示例

```
html_json = json.loads(res.text)
```

json.dump(python,f,ensure_ascii=False)

- 作用

```
把python数据类型 转为 json格式的字符串
# 一般让你把抓取的数据保存为json文件时使用
```

- 参数说明

```
第1个参数：python类型的数据(字典，列表等)
第2个参数：文件对象
第3个参数：ensure_ascii=False # 序列化时编码
```

- 示例

```python
import json

app_dict = {
    '应用名称' : 'QQ',
    '应用链接' : 'http://qq.com'
}
with open('小米.json','a') as f:
    json.dump(app_dict,f,ensure_ascii=False)
```

今日作业

```
1、有道翻译案例复写一遍
2、百度翻译案例复写一遍
3、民政部数据抓取案例完善
   # 1、将抓取的数据存入数据库，最好分表按照层级关系去存
   # 2、增量爬取时表中数据也要更新
4、把链家二手房案例改写为多线程
```