分析 AJAX 传递的 JSON 获取数据

获取动态数据两种的思路

1. 分析页面请求，模拟 ajax 请求

2. 利用 selenium 模拟浏览器行为或其他抓包工具直接获取

普通 ajax 请求

案例：获取豆瓣电影的信息

需求

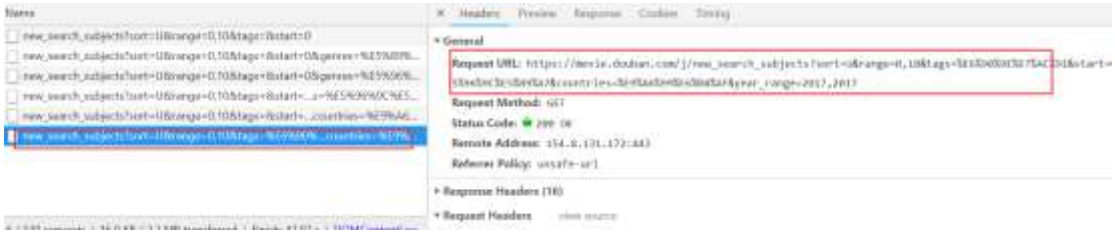
https://movie.douban.com/explore#!type=movie&tag=%E7%83%AD%E9%97%A8&sort=recommend&page_limit=20&page_start=0

爬取电影的名称，url，评分，封面图片

审查元素

打 开
https://movie.douban.com/explore#!type=movie&tag=%E7%83%AD%E9%97%A8&sort=recommend&page_limit=20&page_start=0

选择电影标签



Request URL:
https://movie.douban.com/j/new_search_subjects?sort=U&range=0,10&tags=%E6%90%9E%E7%AC%91&start=0&genres=%E5%96%9C%E5%89%A7&countries=%E9%A6%99%E6%B8%AF&year_range=2017,2017

查看 Query string

sort:T　排序

range:0,10　评分范围

tags:电影,剧情　标签（电影类型）

#genres:类型

#countries：国家和地区

start:0 数据的起始位置 从 0 开始 每页 20 条数据，取值 0，20，40。。。

year_range:年份

代码实现

```
from urllib import request
from urllib import parse
import os
import json

def load_page(url,page):
    """发送请求加载页面"""
    headers = {
        'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/68.0.3440.75 Safari/537.36'}
    # 构造请求
    req = request.Request(url, headers=headers)
    try:
        # 发送请求
        response = request.urlopen(req)
        if response.getcode() == 200:
            json_data = response.read().decode()
            data = json.loads(json_data)['data']
            print(type(data))
            print(len(data))
            for item in data:
                print('directors:',''.join(item['directors']))
                print('rate:',item['rate'])
                print('star:',item['star'])
                print('title:', item['title'])
                print('url:', item['url'])
                print('casts:',''.join(item['casts']))
                print('cover:', item['cover'])
                print('='*60)

            if not os.path.exists('./data'):
                os.mkdir('./data')
```

```
                filename = 'douban_'+str(page)+'.json'
                file_path = os.path.join('./data', filename)
                with open(file_path, 'w',encoding='utf-8') as fp:
                    fp.write(json_data)
            else:
                print('请求出错')
        except Exception as err:
            print(err)


def spider(page):
    """爬虫程序"""
    base_url = 'https://movie.douban.com/j/new_search_subjects?'
    # 构造关键字  请求需要带上的参数
    key_words = {
        "sort": "T",  # 排序的方式
        "range": "0,10",  # 电影评分的范围
        "tags": "电影,剧情,美国",  # 检索的标签
        #"playable": "1",  # 是否可以播放
        "start": (page-1)*20,  # 检索的开始位置  (这里可以去改变的 从 0 开始 一个电影代表的是一条数据)
        "genres": "喜剧",#类型
        "countries": "中国大陆", # 国家地区
    }
    # 编码
    key_words = parse.urlencode(key_words)
    url = base_url + key_words
    # 调用函数
    load_page(url,page)


if __name__ == "__main__":
    for i in range(3):
        print('page:',i+1)
        spider(i+1)
```

http 请求头中的 Referer

Referer 请求头的一部分,表示一个来源。

作用:

1.防盗链。

比如我只允许我自己的网站访问自己的图片服务器，假设域名是 www.haibao.com，那么图片服务器每次取到 Referer 来判断一下域名是不是 www.haibao.com，如果是就继续访问，不是就拦截。

2.防止恶意请求。

比如静态请求是*.html 结尾的，动态请求是*.shtml，那么由此可以这么用，所有的*.shtml 请求，必须 Referer 为我们我自己的网站。

一个 HTTP 请求中可能不包含 Referer 或者 Referer 内容为空，意味着允许直接在浏览器的地址栏中输入一个资源的 URL 地址直接访问

案例：获取 A 股上市公司信息

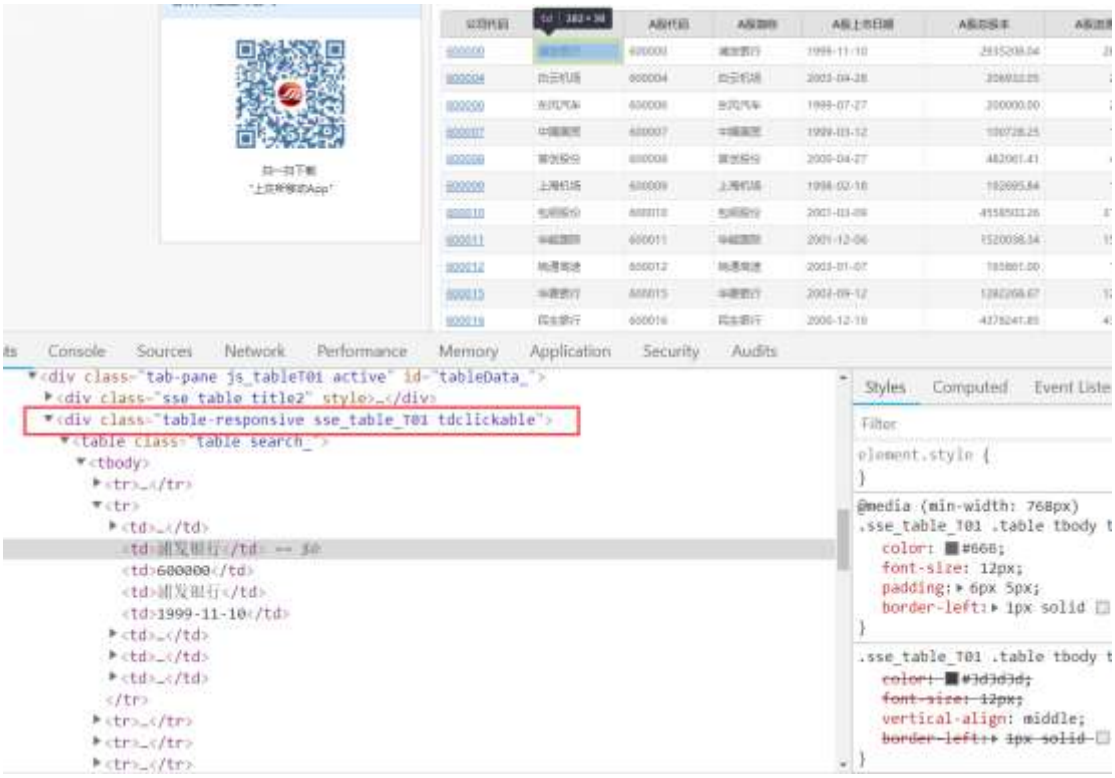
需求

爬取上海证券交易所网站，获取 A 股上市公司信息，包括公司代码，公司简称，A 股代码，A 股简称以及 A 股总资本和 A 股流通资本这几项

审查元素

打开 http://www.sse.com.cn/assortment/stock/list/share/

查看 A 股上市公司信息



发现上市公司条目在 class 为 table-responsive sse_table_T01 tdclickable 的 div 下的 table 中

查看网页源码，发现该 div 为空，我们用普通的爬取方法是获取不到数据的

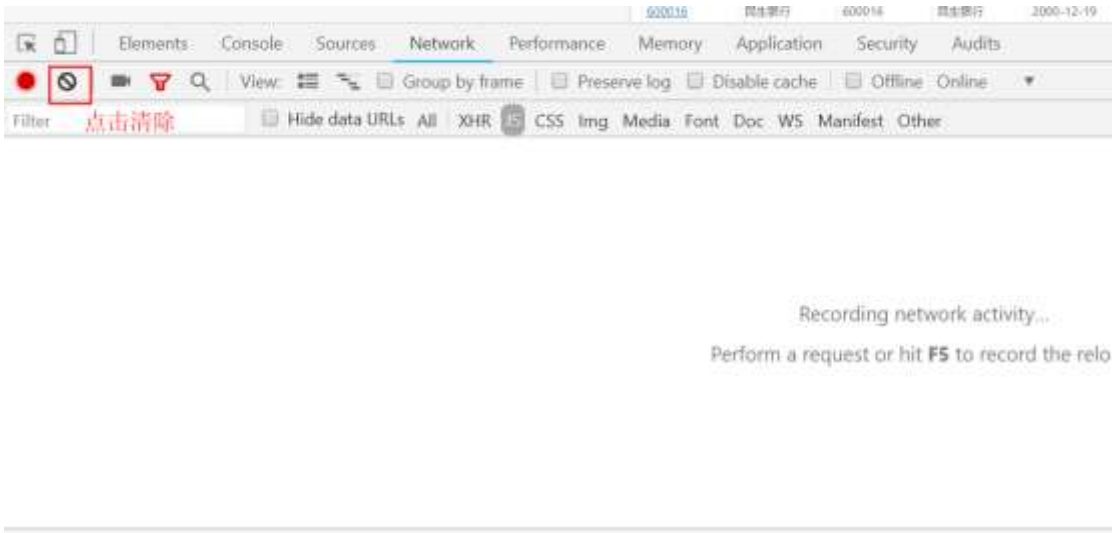
```
<div class="table-responsive sse_table_T01 tdclickable"  >
<table class="table search_">
<script type="text/javascript">

</script>
            </table></div>
```

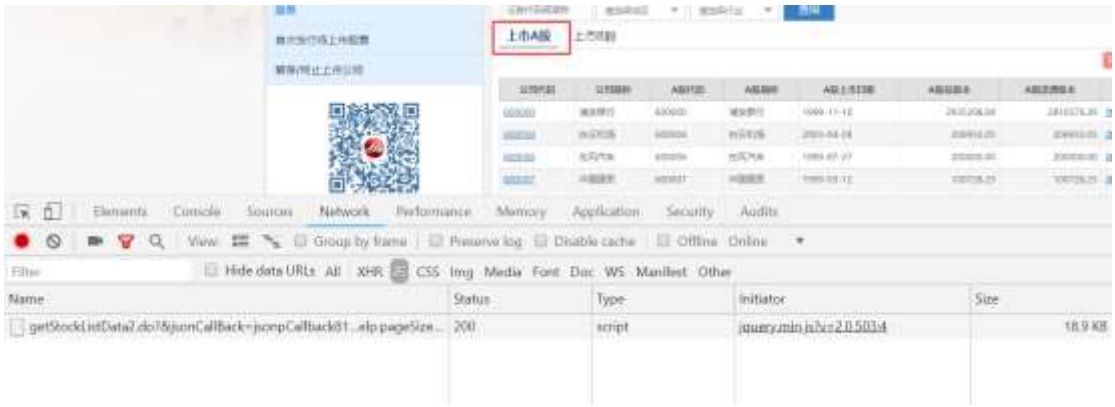
这是因为这个网页是使用了 AJAX 技术的**动态网页。**它可以通过与服务器进行少量数据交换而在不重新加载整个网页的情况下对部分网页进行异步更新。

分析请求数据的 URL

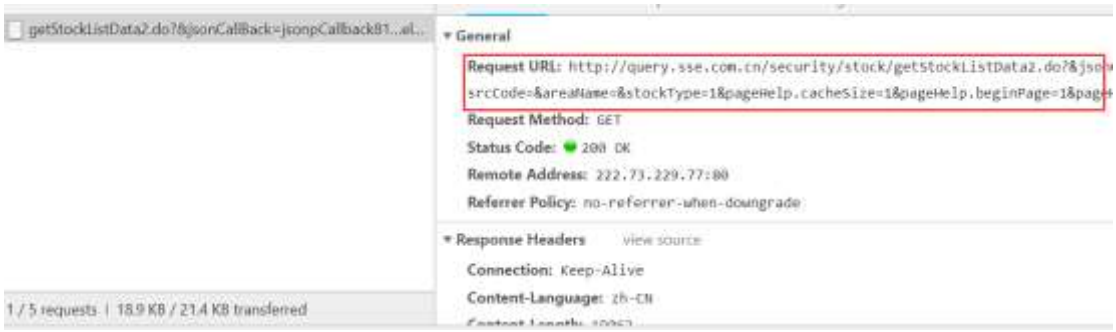
打开开发者工具，选 Network 进行 JS 分析，点击清除，更方便定位



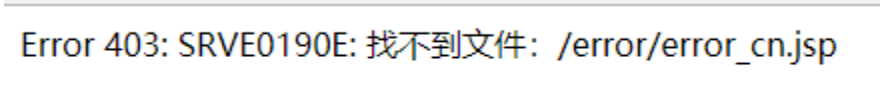
点击网页中上市 A 股，进行加载刷新



出现了一个条目，就是我们要找的数据的 url，（有的网站会出现一堆的 js 条目，就需要去分析一下了）



浏览器里打开目标验证一下有没有我们需要的数据



403 码表示我们没有权限浏览目标地址。这是网站的自我保护行为。

JSON 数据的获取

思路：修改 Request-Headers 中 Cookie，User-Agent，Referer 等信息，让爬虫模拟人的操作

需要修改的内容可以在 Headers 中查看：

代码实现

```
import requests

url='http://query.sse.com.cn/security/stock/getStockListData2.do?&jsonCallBack=jsonpCallback81793&isPagination=true&stockCode=&csrcCode=&areaName=&stockType=1&pageHelp.cacheSize=1&pageHelp.beginPage=1&pageHelp.pageSize=25&pageHelp.pageNo=1&_=1531271694633'
headers={'Cookie':'yfx_c_g_u_id_10000042=_ck18071109145317901555880472178;
yfx_f_l_v_t_10000042=f_t_1531271693783__r_t_1531271693783__v_t_1531272783524__r_c_0; VISITED_MENU=%5B%228528%22%2C%228464%22%5D',
'User-Agent':'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/67.0.3396.99 Safari/537.36',
'Referer':'http://www.sse.com.cn/assortment/stock/list/share/'
}
response=requests.get(url,headers=headers)
```

```
print(response.text)
a='{"content":'+response.text[19:-1]+'}'#去除开都的
jsonpCallback81793(和结尾的）
b=json.loads(a)
print(b)
```

http 请求头中添加 Cookies