WOS 爬虫总结

目的

- 快速获得文献背景：出版年份，被引频次，作者，DOI，文献类型，引用的参考文献
- 获得参考文献的下载链接，实现文献的批量化下载

条件

- 所处机构或者学校**购买WOS的数据库**，并且将爬虫**置于校园网环境中。**
- 如果要实现后期的文献下载需要**购买**所需文献的**数据库。**
- 后期有时间会完善用账号密码校外访问数据库。

使用方法

1. 所需要安装的python3+包

```
pip install requests
pip install lxml
pip install bs4
```

2. 测试例子(没有将程序打包，所以需要将程序下载使用)

- 导出所有的检索结果

```
test = 'TS=LN AND PY=(2018-2020)' # 检索式一定要有两个条件以上
test_start = 1  # 导出起始页码
test_end = 501  # 导出终止页码
file_name='LNOI' # 保存文件的名称，默认为 .txt 文件，如果想要保存其他格式，那是不可能的!
file_type = 'fieldtagged'
demo = export_paper(search_expression=test, export_start=test_start,
                        export_end=
test_end,file_name='LNOI',file_type=file_type)
demo.save()
```

- 导出所有的参考文献

```
# 运行结束会生成两个txt文件，一个是'file_name.txt'为所选需要的文献，一个是'no_doi.txt' 用于存储没有DOI的文献信息
aim = 'TS=LNOI AND PY=2020 AND DO=10.1515/nanoph-2020-0013' # 建议用DOI搜索，这样保证搜索结果的唯一性
file_name = 'LNOI'
aim_paper = get_references(search_expression=aim, file_name=file_name)
aim_paper.get_main() # 接口和上一个有点不一样，两个爬取逻辑有点小差异
```

- WOS 检索式参考

```
# ''' 高级检索参考 :
# 布尔运算符: AND、OR、NOT、SAME、NEAR
# 字段标识:
#          TS= 主题
```

```
#          TI= 标题
#          AU= 作者 [索引]
#          AI= 作者识别号
#          GP= 团体作者 [索引]
#          ED= 编者
#          AB= 摘要
#          AK= 作者关键词
#          KP= Keyword Plus ®
#          SO= 出版物名称 [索引]
#          DO= DOI
#          PY= 出版年
#          AD= 地址
#          SU= 研究方向
#          IS= ISSN/ISBN
# '''
```

爬虫细节分享

参考资料

- 主要参考博主的思路,博主的代码仓库地址

需要的关于爬虫的基础知识

- 爬虫原理与cookies，session
- python Requests 基础使用
- 关于异步加载和异步传输的概念
- 爬虫问题的重定向302错误
- BeautifulSoup 模块使用指南
- python 正则 re 表达式

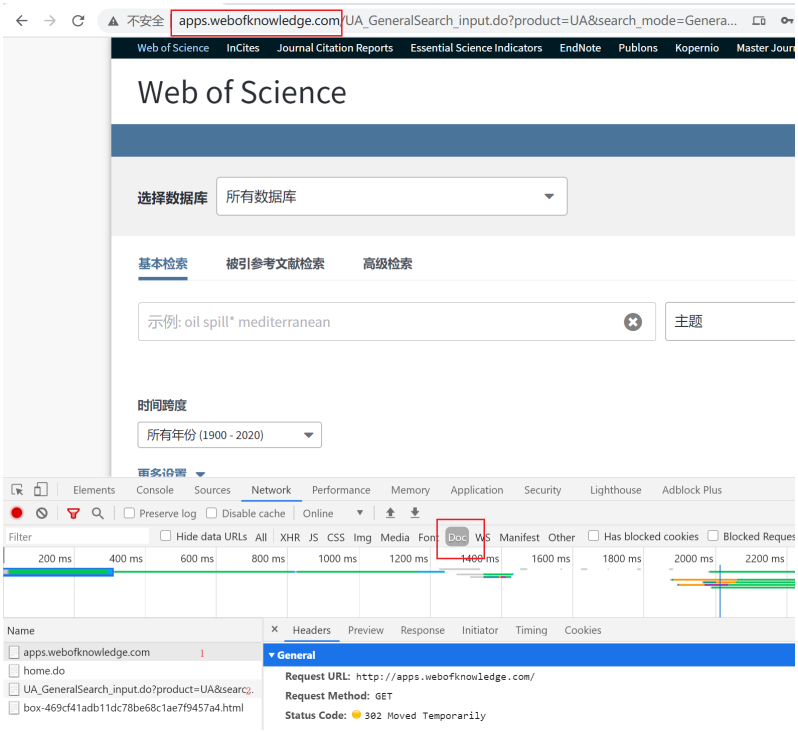
操作爬虫时的好用的工具

- 在线解析工具
- 新建txt文档

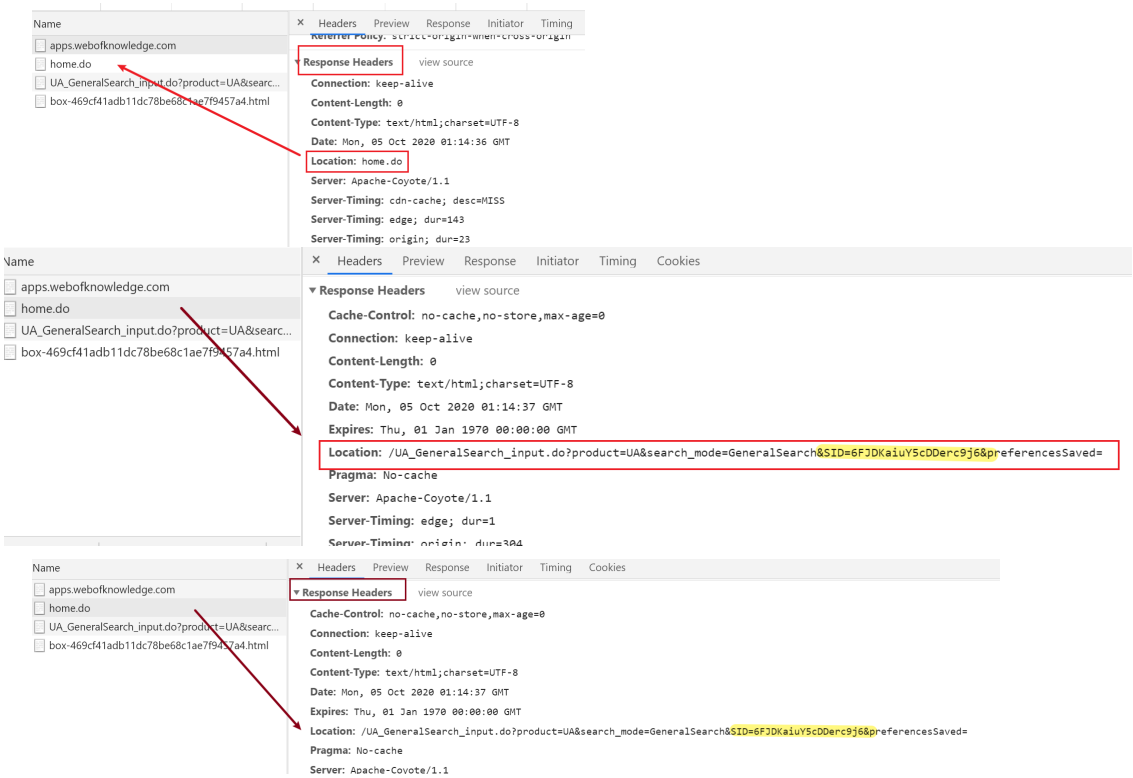
爬取逻辑

1. 获取SID

-

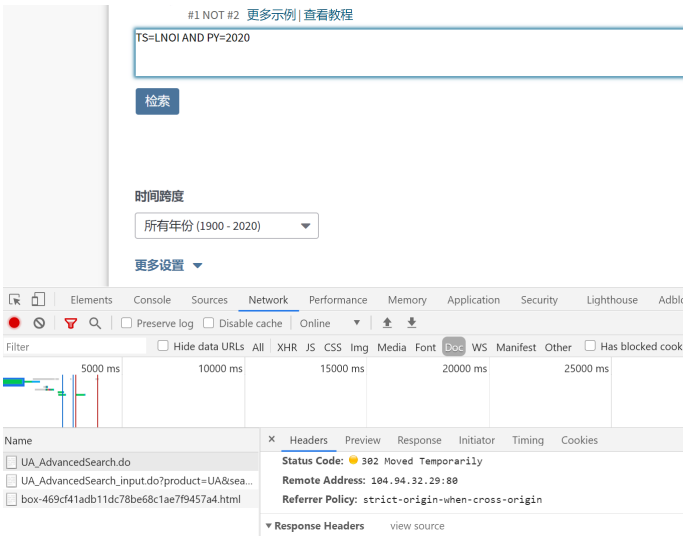


如图所示，当我们输入`http://apps.webofknowledge.com/`时，浏览器就加载两个重要页面，一个是我们输入的url（图中1），另一个就是基本检索url（图中2），这时会发现`Status Code`为302，简单理解就是他跳转其他页面了，跳转的页面可以在`Response Headers`下的`Location`找到，然后看看看基本检索的url（图中的2），其构造多一个SID参数，于是我们得顺腾摸瓜寻找SID参数。摸瓜方式就是寻找他的响应值，参考下图可以很容易找到，然后将其提取出来。

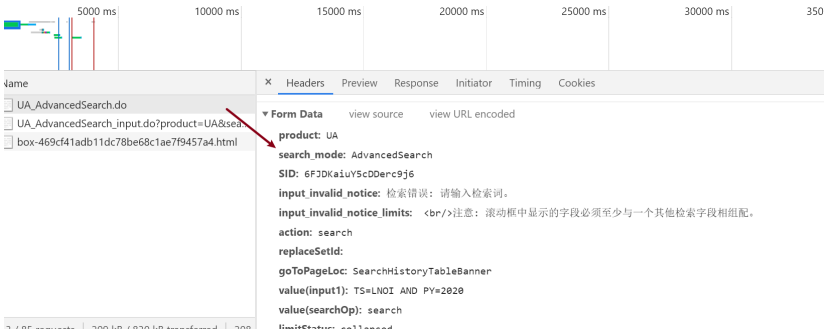


2. 进行高级检索

- 检索所需要的语法介绍
- 发起POST请求

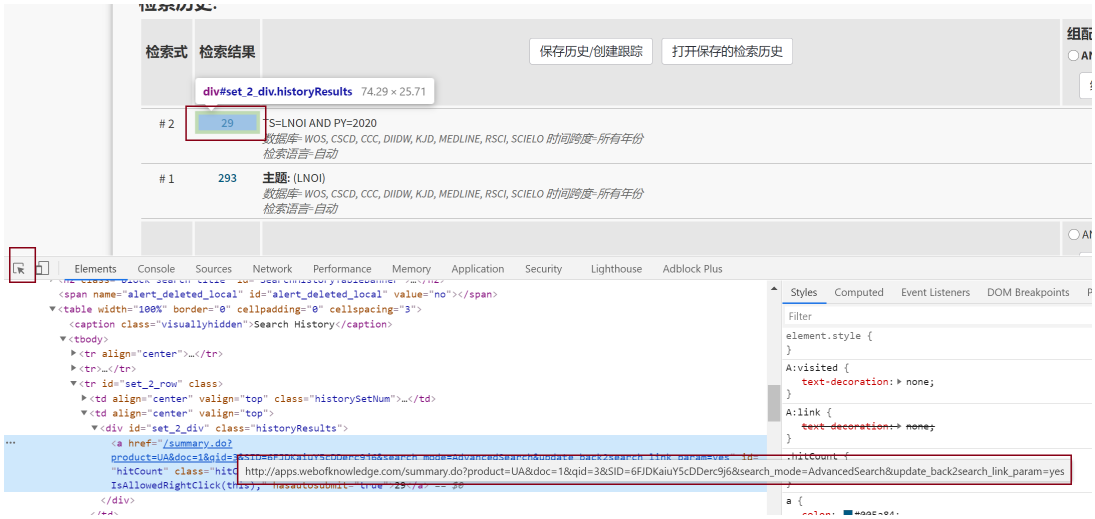
1. 

如图所示，当发起请求之后，出现了老朋友302，不过这里是POST请求，所以需要`Form Data`进行提交。按照浏览器找到的元素保存为dict就行。具体参数分析见参考资料中的大佬的分享。

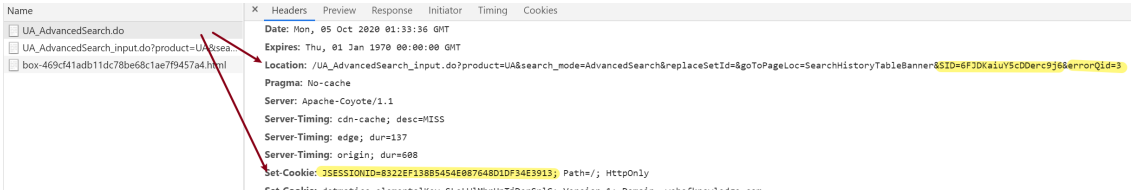


3. 高级检索结果分析

1. 高级检索后不会直接跳转页面，而是在网页下边检索历史中，我们所需要的结果在检索结果中，点击数字即可跳转。



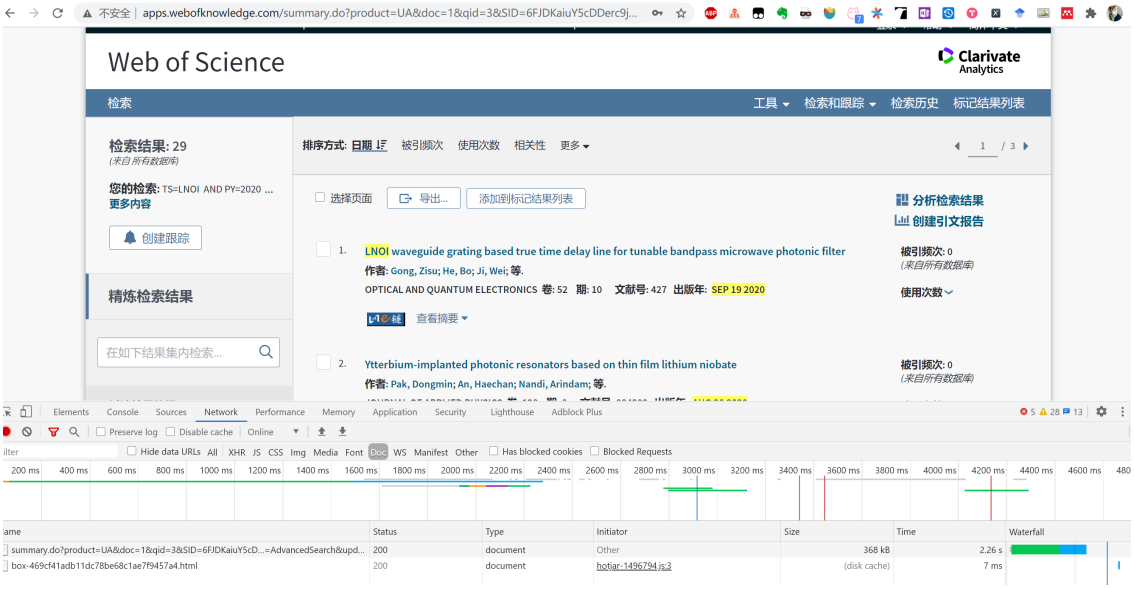
2. 分析上图框框里的url，可以看到里面又多了个参数`qid`,所以，我们需要找到这个参数在哪里？其实很简单，和上面的方法是一样的



可以看到，我上面标了3个参数`qid`,`SID`,`JSESSIONID`.这个是我在爬取的时候，发现`JSESSIONID`参数加上之后可以让爬虫更稳定，不然有时候就获取不了网页结果。更新之后的url为

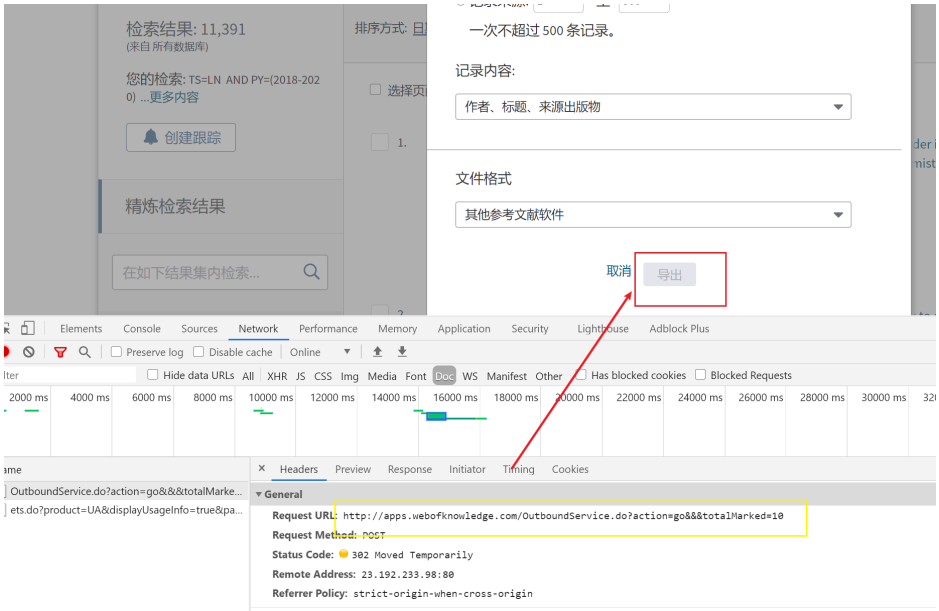
```
entry_url = 'http://apps.webofknowledge.com/summary.do;jsessionid={jse}?
product=UA&doc=1&qid={qid}&SID=
{sid}&search_mode=AdvancedSearch&update_back2search_link_param=yes'
```

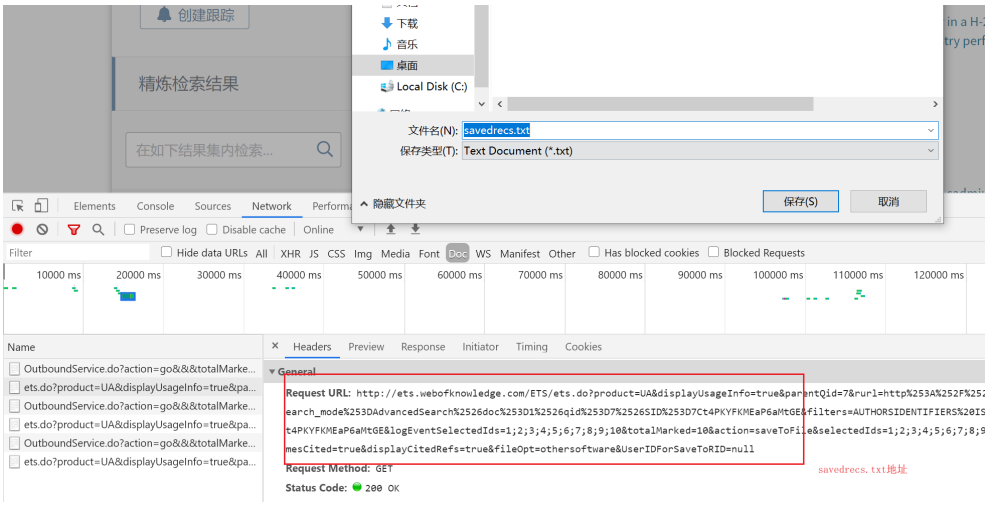
3. GET 提取的 URL 就进入了我们所需要的界面，



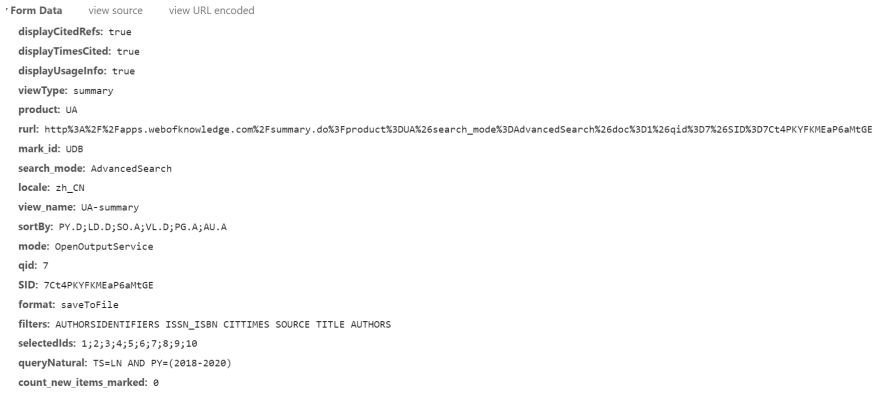
4. 导出页面分析

1. 导出页面由两部分url组成，一部分为导出所选标记，另一部分为下载所选文件，其所对应的页面逻辑分别为





2. 可以看到这也就是一个重定向的POST+GET方式，和上面相同，所以直接POST好对应的 `form_data` 就行。



3. 大概流程就是这样，目前实现了两个功能。
    - 根据检索方式导出页面论文，支持导出500+，WOS限制为一次只能导出500，但是不能超过上限20000
    - 根据 DOI 检索获取该论文的所有参考文献。同时将WOS中无法导出的论文信息保存在 `no_doi.txt` 文件中。

遇到的奇葩问题总结

1. 使用 `print(response.text())` 发现控制台(我的IDE为VSCODE)的显示内容与在网页端的查看源代码显示不同，可能是VSCODE后台省略了，可以保存为 txt 文件或者用bs4解码，看看是否已经获取到了想要的界面。
2. 获取到的源码，没有中文，中文显示都是 `...` 这个问题我百度了好久，都没有解决，后来脑子一抽，修改了一下 `headers` 的 `'Accept-Language': 'zh-CN,zh;q=0.9'`，居然就成功了， 我很开心，哈哈哈！
3. 在获取所有的参考文献的时候出了点一点意外，无法成功导出文献，原因是提交的 `form_data` 出问题了，需要仔细检查！