目录

概述

两个任务

任务一，爬取内蒙古自治区环境保护厅环评审批“项目受理情况”，获取字段信息“项目名称”、“建设地点”、“建设单位”、“环境影响评价机构”和受理日期。

任务二， 网页遍历，爬取所有页面和记录页面间父子关系

爬取目标的一般思路

观察目标URL，分析目标结构，适配所有目标，导出数据

需求达成确认

编号	任务	提取的信息	准确度	完成打钩
1	环评信息爬取	项目名称	100%	✅
		建设地点	100%	✅
		建设单位	100%	✅
		环境影响评价机构	100%	✅
		受理日期	100%	✅
2	网页遍历	实现方式	phantomJS	
		整体完成度	100%	✅

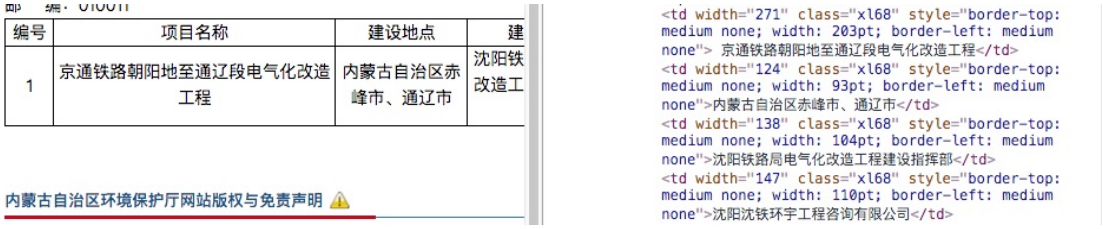
操作环境

```
Mac OS 10.12.6
PyCharm 2017.3
python 3.6.3
Scrapy 1.4.0 selenium 3.8.0 phantomJS 2.2.1 BeautifulSoup4 lxml 4.1.1
re requests json csv
```

实现思路

1. 环评信息爬取
    - 获取需要爬取的URL
    - 分析网页结构
        - 第一种结构为列表式的，解决方法正则匹配获取

        - 第二种结构为表格式的，解决方法BeautifulSoup + lxml解析



    - 数据整理并输出csv文件

1. 网页遍历
    - 观察页面结构，可分为两种类型
        - 第一种 a标签下的url
        - 第二种 JS事件
    - 初次想法
        - 应用Scrapy 框架循环遍历URL，对JS 事件应用selenium+Firefox模拟
    - 初次完成后，结果不尽人意，并没有完整的匹配和找出正确的页面关系，So，重构了算法
        - 新建一个用于获取页面完整路径的类`GetPath`
        - 创建一个二维数组Paths，用于记录页面完整路径
        - Paths下的每条路径的前一个都是后一个的父页面url，同样后一个都是前一个的子页面url

```
[[mian.html, p1l1.html, p2l1.html,...], [main.html, p2l1.html, p2l2.html....] ..... ]
```

        - 当前父页面的url与路径中最后一位的相比，若相等，直接在路径后面添加子页面的url

若和最后一位的前一位url相等，则新增一条路径，即用子页面url替换原路径的最后一位

- 每次执行时推入子页面的url 生成（更新）路径
- 对于JS事件的页面， 设置标志位

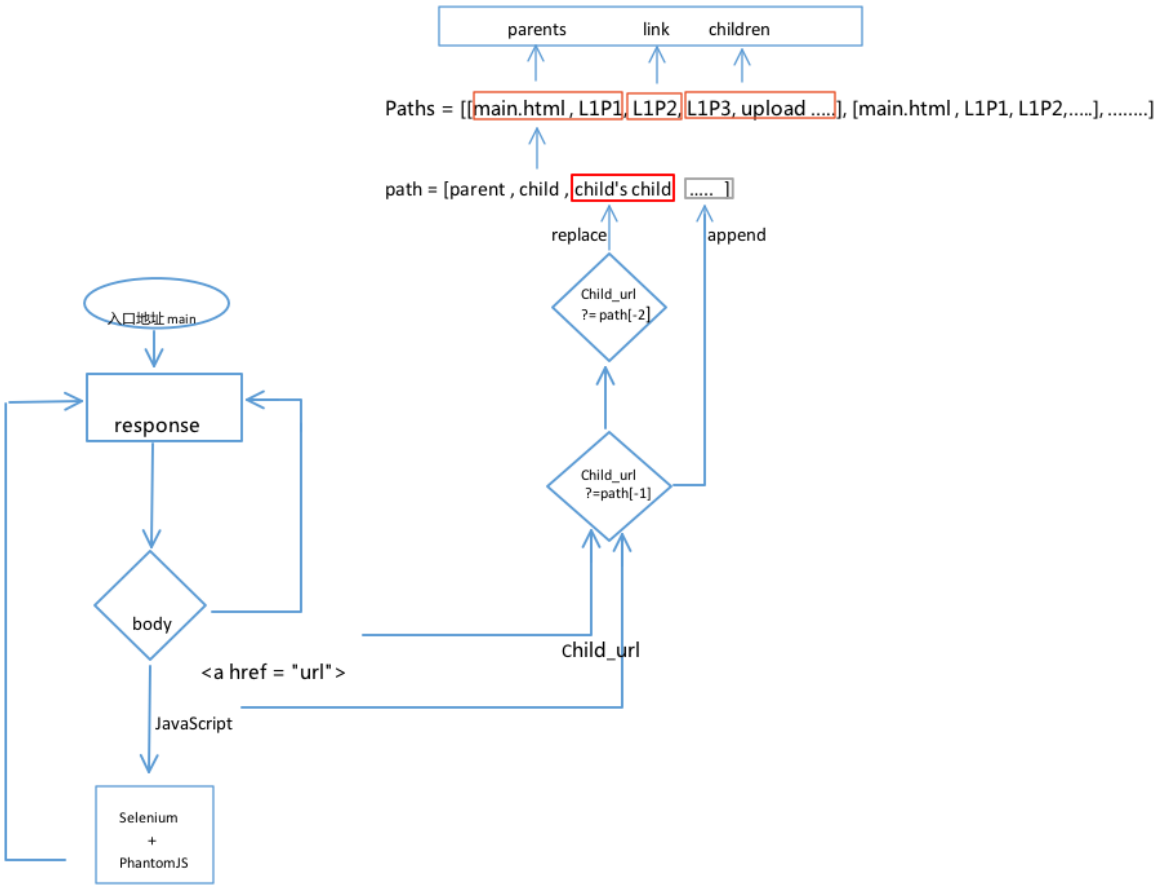
```
request.meta['req_url'] = request.url
request.meta['flag'] = flag
request.meta['PhantomJS'] = True
```

- 同时解析JS事件

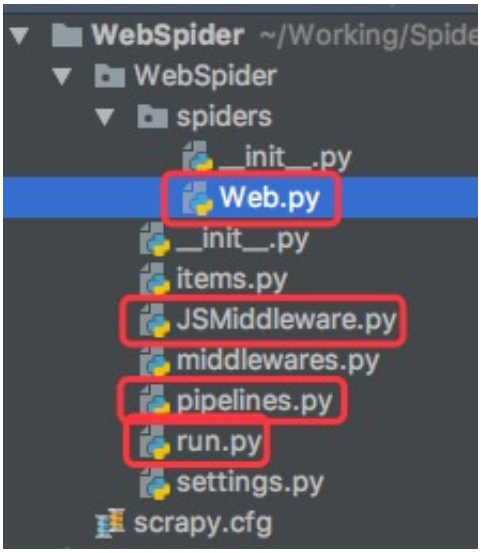
```
if 'PhantomJS' in request.meta.keys():
    try:
        driver = webdriver.PhantomJS(service_args=['--ignore-ssl-errors=true', '--ssl-protocol=TLSv1'])  # 容错
        driver.implicitly_wait(10)  # 隐式等待10s
        driver.get(request.url)

        flag = request.meta['flag']  # request 携带的JS代码
        driver.execute_script(flag)  # 执行JS
        body = driver.page_source
        # print(body)

        html = HtmlResponse(driver.current_url, encoding="utf-8", body=body, request=request)
        driver.quit()  # 断开
        return html

    except:
        return
```

- 生成需要的数据格式并导出

- 流程图



- 目录树



几个重要的文件，其他均是Scrapy 框架文件

Web.py 爬虫代码实现
JSMiddleware.py JS事件中转控制处理
pipelines.py 数据处理及导出文件
run.py 函数启动

运行(终端下)

1. 环评信息爬取

`python3 spiderInfo.py`

1. 网页遍历

> 若入口网址不同，请先修改入口网址，默认为http://127.0.0.1:8080/main.html
>
> 文件位置位于 WebSpider/spiders/Web.py

`scrapy crawl Web`

若命令不好使，请用PyChrm 打开，运行`run.py`

1. 成功运行将生成 `内蒙古自治区环评数据.csv` 和 `Webspider.json` 文件

小结

相对来说任务一较任务二简单一些，直接用requests 配合 正则表达式，但不能全部适应，之后加上 BeautifulSoup 完美适应。

而任务二，学习Scrapy，Selenium, 适配JS, 重构算法 ，期间还有期末考试，花费时间就多了一些。但收获也是颇多的，除了新框架知识，把以前遗忘的地方也重新熟悉了一遍。

碰到了一些有趣的问题， 比如爬取的时候遇到一个编码问题，python 的神奇编码遇见可是不少。最有趣的是对于算法的思考和优化。

总的来说，通过测试任务收获不少，整体难度不是很高，挺适合自己的。

附件

1. spiderInfo.py　测试任务一爬虫
2. WebSpider　　测试任务二爬虫