爬虫实验报告

一、实验要求

1.爬取学堂在线的计算机类课程页面内容

要求将课程名称、老师、所属学校和选课人数信息，保存到一个csv文件中。

2.爬取链家官网二手房的数据

要求爬取北京市东城、西城、海淀和朝阳四个城区的数据（每个区爬取5页），将楼盘名称、总价、平米数、单价保存到json文件中。

二、爬虫和scrapy介绍

1. 爬虫

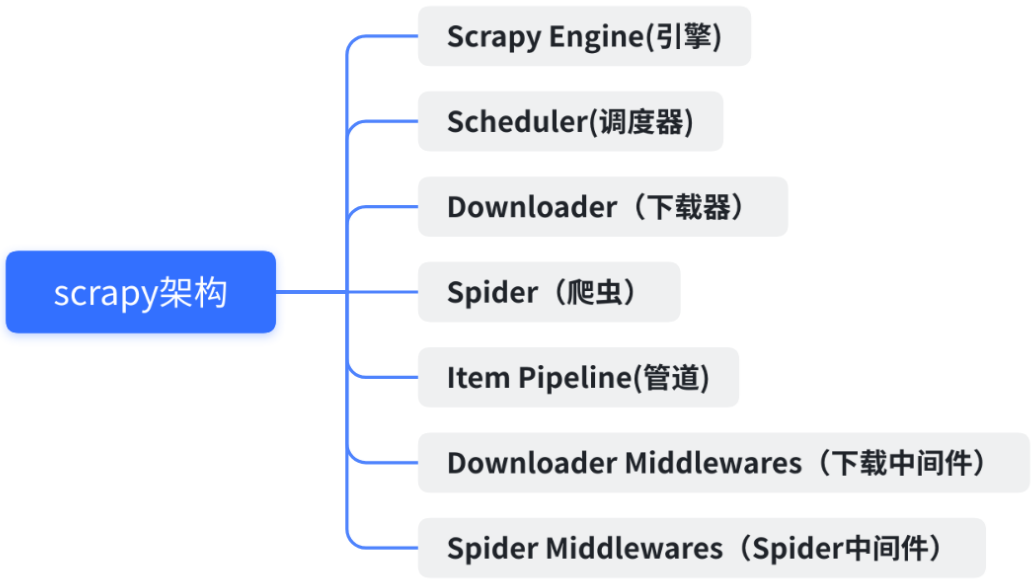
网络爬虫，是一种按照一定的规则，自动地抓取万维网信息的程序或者脚本。

网络爬虫是一个自动提取网页的程序，它为搜索引擎从万维网上下载网页，是搜索引擎的重要组成。传统爬虫从一个或若干初始网页的URL开始，获得初始网页上的URL，在抓取网页的过程中，不断从当前页面上抽取新的URL放入队列，直到满足系统的一定停止条件。聚焦爬虫的工作流程较为复杂，需要根据一定的网页分析算法过滤与主题无关的链接，保留有用的链接并将其放入等待抓取的URL队列。然后，它将根据一定的搜索策略从队列中选择下一步要抓取的网页URL，并重复上述过程，直到达到系统的某一条件时停止。另外，所有被爬虫抓取的网页将会被系统存贮，进行一定的分析、过滤，并建立索引，以便之后的查询和检索；对于聚焦爬虫来说，这一过程所得到的分析结果还可能对以后的抓取过程给出反馈和指导。

2. scrapy库

Scrapy是适用于Python的一个快速、高层次的屏幕抓取和web抓取框架，用于抓取web站点并从页面中提取结构化的数据。Scrapy用途广泛，可以用于数据挖掘、监测和自动化测试。

Scrapy吸引人的地方在于它是一个框架，任何人都可以根据需求方便的修改。它也提供了多种类型爬虫的基类，如BaseSpider、sitemap爬虫等，最新版本又提供了web2.0爬虫的支持。



Scrapy Engine(引擎)：负责Spider、ItemPipeline、Downloader、Scheduler中间的通讯，信号、数据传递等。

Scheduler(调度器)：它负责接受引擎发送过来的Request请求，并按照一定的方式进行整理排列，入队，当引擎需要时，交还给引擎。

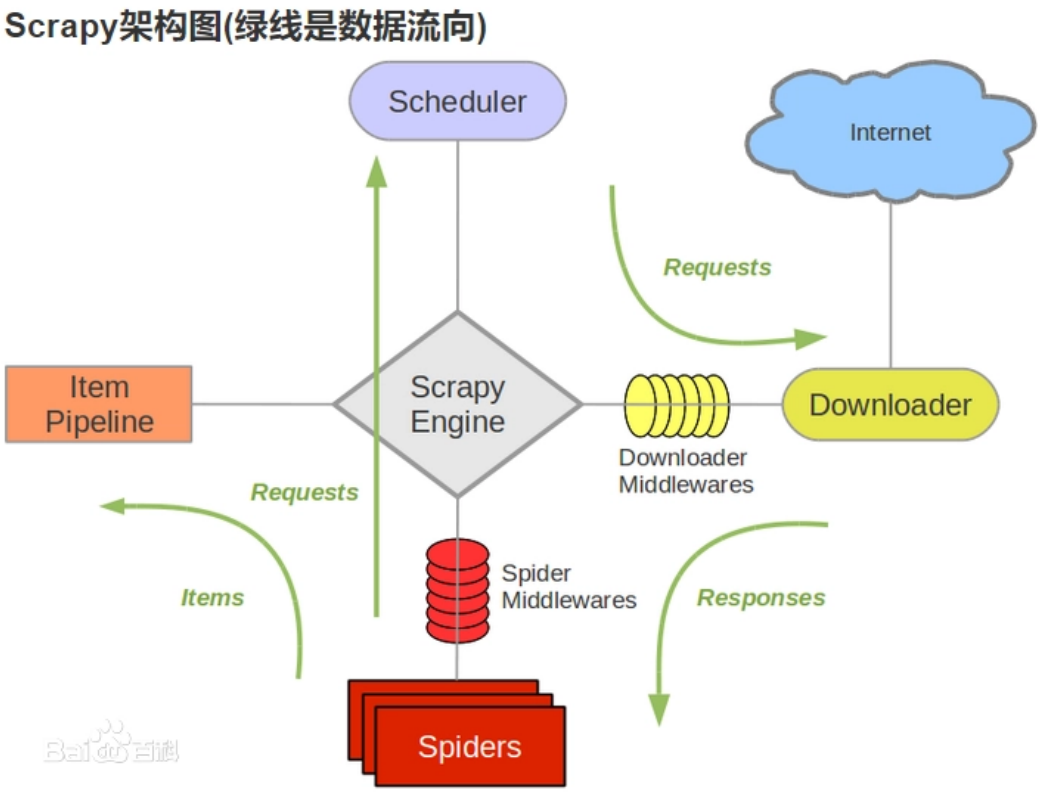
Downloader（下载器）：负责下载Scrapy Engine(引擎)发送的所有Requests请求，并将其获取到的Responses交还给Scrapy Engine(引擎)，由引擎交给Spider来处理。

Spider（爬虫）：它负责处理所有Responses,从中分析提取数据，获取Item字段需要的数据，并将需要跟进的URL提交给引擎，再次进入Scheduler(调度器)。

Item Pipeline(管道)：它负责处理Spider中获取到的Item，并进行进行后期处理（详细分析、过滤、存储等）的地方。

Downloader Middlewares（下载中间件）：一个可以自定义扩展下载功能的组件。

Spider Middlewares（Spider中间件）：一个可以自定扩展和操作引擎和Spider中间通信的功能组件。



三、使用的工具

实验使用scrapy库，结合selenium库实现**下载中间件**，从而实现进行动态爬取

selenium介绍

Selenium是一个用于Web应用程序测试的工具。Selenium测试直接运行在浏览器中，就像真正的用户在操作一样。支持的浏览器包括IE（7, 8, 9, 10, 11），Mozilla Firefox，Safari，Google Chrome，Opera，Edge等。这个工具的主要功能包括：测试与浏览器的兼容性——测试应用程序看是否能够很好得工作在不同浏览器和操作系统之上。测试系统功能——创建回归测试检验软件功能和用户需求。支持自动录制动作和自动生成.Net、Java、Perl等不同语言的测试脚本。

Selenium 有很多东西，但从本质上讲，它是一个 Web 浏览器自动化工具集，它使用可用的最佳技术远程控制浏览器实例并模拟用户与浏览器的交互。它允许用户模拟最终用户执行的常见活动；在字段中输入文本，选择下拉值和复选框，并单击文档中的链接。它还提供了许多其他控件，例如鼠标移动、任意 JavaScript 执行等等。

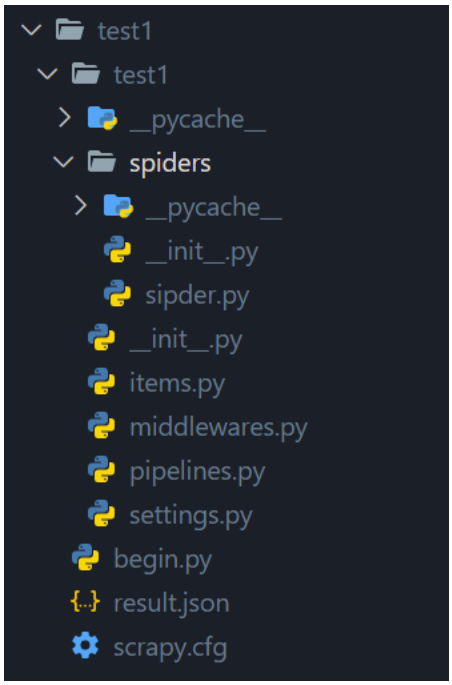
三、学堂在线爬取

1.创建scrapy项目

创建目录，执行命令

```
Apache
1   scrapy startproject test1
```

文件结构如下：



其中，spider.py是主要的爬虫文件，middleware是中间件的文件。

pipelines是管道文件，需要编写主要的流程代码。

settings.py是相关配置文件

更具体的：

Plain Text

```
project_folder -- 项目文件夹名称
|
|——project_name -- 该项目的python模块，一般和项目文件夹名称相同
|  |
|  |——spider -- 放置spider代码的包，以后所有的爬虫，都存放在这个里面
|  |
|  |——items.py -- 用来存放爬虫怕写来的数据的模型
|  |
|  |——middlewares.py -- 用来存放各种中间件的文件
|  |
|  |——pipelines.py -- 用来对items里面提取的数据做进一步处理，如保存到本地磁盘等
|  |
|  |——settings.py -- 本爬虫的一些配置信息(如请求头、多久发送一次请求、ip代理池等)
|
|——scrapy.cfg -- 项目的配置文件
```

2.begin.py

这里的作用是快速启动爬虫。

Python

```python
from scrapy import cmdline
cmdline.execute("scrapy crawl bupt".split())
```

3.pipeline.py

pipeline是主要的爬虫流程管道。

分为open_spider,process_item,colse_spider三个部分。

open_spider是爬虫初始化。

process_item是对于每个得到的item具体要怎么做。

这里写的是将item的内容加入大json_str里面。

close_spider是指将结果输出到文件

Python

```python
from itemadapter import ItemAdapter
import json

class Test1Pipeline:
    def open_spider(self,spider):
        try:
            self.file=open("result.json","w",encoding="utf-8")
        except Exception as err:
            print(err)
        self.json_str=[]
    def process_item(self, item, spider):
        dict_item = dict(item)
        self.json_str.append(dict_item)
        return item
    def close_spider(self,spider):
        self.file.write(json.dumps(self.json_str,ensure_ascii=False))
        self.file.close()
```

4.middlewares.py

因为需要处理动态网页，所以要重写process_request的内容，通过selenium里面的webdriver来访问网页。

option是对于webdriver的一些配置。

option里面的设置例如禁用gpu加速，不打开沙盒模式，防止被识别为爬虫等等。

最后将request通过htmlresponse来得到响应，将响应传给parse进行处理。

Python

```python
def process_request(self, request, spider):
        # Called for each request that goes through the downloader
        # middleware.

        # Must either:
        # - return None: continue processing this request
        # - or return a Response object
        # - or return a Request object
        # - or raise IgnoreRequest: process_exception() methods of
        #   installed downloader middleware will be called
        option=webdriver.ChromeOptions()
        option.add_argument("--headless")
        option.add_argument("--disable-gpu")
        option.add_argument("no-sandbox")
        option.add_argument("disable-blink-features=AutomationControlled")
        option.add_experimental_option('excludeSwitches', ['enable-automation'])
        driver = webdriver.Chrome(chrome_options=option)
        driver.execute_cdp_cmd("Page.addScriptToEvaluateOnNewDocument", {
            "source": """
                Object.defineProperty(navigator, 'webdriver', {
                  get: () => undefined
                })
              """
        })
        driver.get(request.url)
        driver.implicitly_wait(5)
        content = driver.page_source
        driver.quit()
        ret=HtmlResponse(request.url, encoding="utf-8", body=content, request=re
quest)
        return ret
```

5.items.py

定义需要的Field

Python

```python
class Test1Item(scrapy.Item):
    # define the fields for your item here like:
    # name = scrapy.Field()
    name=scrapy.Field()
    district=scrapy.Field()
    total_price=scrapy.Field()
    unit_price=scrapy.Field()
    square=scrapy.Field()
    pass
```

6.spider.py

这里重写start_requests，使得可以同时对四个连接进行爬取。

最后通过callback的方法进行翻页，模拟点击翻页的操作。

通过xpath路径找到对应的文字位置。

Python

```python
import scrapy
from scrapy.http import response
from test1.items import Test1Item
py2hz={"dongcheng":"东城","xicheng":"西城","haidian":"海淀","chaoyang":"朝阳"}
class mySpider(scrapy.spiders.Spider):
    name="bupt"
    allowed_domains=["bj.lianjia.com"]
    start_urls=["https://bj.lianjia.com/ershoufang/dongcheng/",
                "https://bj.lianjia.com/ershoufang/xicheng/",
                "https://bj.lianjia.com/ershoufang/chaoyang/",
                "https://bj.lianjia.com/ershoufang/haidian/"]
    def start_requests(self):
        for url in self.start_urls:
            yield scrapy.Request(url,callback=self.parse,cb_kwargs={"page_num":1})
    def parse(self,responce,**kwargs):
        self.logger.info(kwargs.get("page_num"))
        item=Test1Item()
        self.logger.info("========================================")
        for each in responce.xpath("/html/body/div[4]/div[1]/ul/*"):
            item['district']=py2hz[responce.url.split("/")[4]]
            item['name']=each.xpath("./div[1]/div[1]/a/text()").get()
            item['total_price']=each.xpath("./div[1]/div[6]/div[1]/span/text()").get()+"万"
            item['unit_price']=each.xpath("./div[1]/div[6]/div[2]/span/text()").get()
            item['square']=each.xpath("./div[1]/div[3]/div/text()").get().split("|")[1]
            if (item['name']):
                yield(item)

        np=responce.xpath("/html/body/div[4]/div[1]/div[7]/div[2]/div/a[last()]/@href").get();
        if (kwargs.get('page_num')!=5):
            yield responce.follow(np,callback=self.parse,cb_kwargs={"page_num":kwargs.get('page_num')+1})
```

7.结果

```
[
    {
        "district": "东城",
        "name": "天天家园 3室1厅 南",
        "total_price": "850万",
        "unit_price": "67,941元/平",
        "square": " 125.11平米 "
    },
    {
        "district": "东城",
        "name": "东城区 悠胜美苑 03年新楼 商品房 视野好",
        "total_price": "545万",
        "unit_price": "77,669元/平",
        "square": " 70.17平米 "
    },
    {
        "district": "东城",
        "name": "忠实里东区 中间层 带电梯 商品房",
        "total_price": "632万",
        "unit_price": "109,514元/平",
        "square": " 57.71平米 "
    },
    {
        "district": "东城",
        "name": "龙潭湖 龙潭北里小区 规整社区 配套齐全全南向三居室",
        "total_price": "630万",
        "unit_price": "96,375元/平",
        "square": " 65.37平米 "
    },
    {
        "district": "东城",
        "name": "北河沿大街 中间楼层东向两居室",
        "total_price": "666万",
        "unit_price": "125,590元/平",
        "square": " 53.03平米 "
    },
    {
        "district": "东城",
        "name": "菊儿胡同 4室1厅 南 北",
        "total_price": "1430万",
        "unit_price": "154,846元/平",
        "square": " 92.35平米 "
    },
    {
        "district": "东城",
        "name": "满五唯一 视野宽阔 南北通透 采光充足",
        "total_price": "1800万",
        "unit_price": "125,848元/平",
        "square": " 143.03平米 "
    },
    {
        "district": "东城",
        "name": "中海紫御公馆明厨明卫双南向大卧室两居",
        "total_price": "1028万",
        "unit_price": "114,121元/平",
        "square": " 90.08平米 "
    },
    {
        "district": "西城",
        "name": "汽北小区小区 南北通透两居室 诚心出售",
        "total_price": "650万",
        "unit_price": "123,107元/平",
        "square": " 52.8平米 "
    },
    {
        "district": "东城",
        "name": "广渠门内大街 不临街 交通方便",
        "total_price": "655万",
        "unit_price": "102,456元/平",
        "square": " 63.93平米 "
    },
    {
        "district": "东城",
        "name": "富贵园1居室中间楼层，小区人车分流距离地铁100米，",
        "total_price": "648万",
        "unit_price": "101,664元/平",
        "square": " 63.74平米 "
    },
    {
        "district": "东城",
        "name": "东城二环里新改造小区臺公园地铁",
        "total_price": "430万",
        "unit_price": "94,319元/平",
        "square": " 45.59平米 "
    },
    {
        "district": "东城",
        "name": "广渠门忠实里 三居室 东南向 有电梯采光好",
        "total_price": "720万",
        "unit_price": "87,592元/平",
        "square": " 82.2平米 "
    },
    {
        "district": "东城",
        "name": "国瑞城高层南北三居室 地铁崇文门站",
        "total_price": "1699万",
        "unit_price": "107,928元/平",
        "square": " 157.42平米 "
    },
```

```
{
    "district": "东城",
    "name": "崇文门国瑞城中区南北三居，前后观景，双明卫，有钥匙",
    "total_price": "2230万",
    "unit_price": "119,488元/平",
    "square": " 186.63平米 "
},
{
    "district": "朝阳",
    "name": "华腾园南向三居，正看花园，精装修，有固定车位",
    "total_price": "985万",
    "unit_price": "78,088元/平",
    "square": " 126.14平米 "
},
{
    "district": "西城",
    "name": "信和嘉园 南北通透 精装修 户型方正 看房方便 诚心卖",
    "total_price": "2688万",
    "unit_price": "87,132元/平",
    "square": " 308.5平米 "
},
{
    "district": "西城",
    "name": "红居南街 2室1厅 西北",
    "total_price": "469万",
    "unit_price": "102,469元/平",
    "square": " 45.77平米 "
},
{
    "district": "西城",
    "name": "侨办大院  三室一厅  有电梯  满五年唯一",
    "total_price": "1400万",
    "unit_price": "149,095元/平",
    "square": " 93.9平米 "
},
{
    "district": "西城",
    "name": "红莲晴园 三居双卫 阳台大 中间楼层 采光好 无遮挡",
    "total_price": "1200万",
    "unit_price": "101,981元/平",
    "square": " 117.67平米 "
},
{
    "district": "西城",
    "name": "三里河南二巷双南两居，采光好，无遮挡",
    "total_price": "789万",
    "unit_price": "143,716元/平",
    "square": " 54.9平米 "
},
{
    "district": "西城",
    "name": "小区车位充足，四居室，中组部分房，阜成门，月坛公园",
    "total_price": "1218万",
    "unit_price": "103,484元/平",
    "square": " 117.7平米 "
},
{
    "district": "东城",
    "name": "沙滩后街55号院 南北通透  精装修两居室 满五唯一",
    "total_price": "780万",
    "unit_price": "137,059元/平",
    "square": " 56.91平米 "
},
{
    "district": "东城",
    "name": "华城滨河世家+高层大一居+两居格局+满五唯一+钥匙房",
    "total_price": "830万",
    "unit_price": "125,568元/平",
    "square": " 66.1平米 "
},
{
    "district": "东城",
    "name": "本家润园C区  视野好  临地铁  两居室  配套成熟",
    "total_price": "720万",
    "unit_price": "102,535元/平",
    "square": " 70.22平米 "
},
{
    "district": "东城",
    "name": "海运仓规整社区 三室两卫复式带电梯 南北朝向采光好",
    "total_price": "1150万",
    "unit_price": "120,824元/平",
    "square": " 95.18平米 "
},
{
    "district": "东城",
    "name": "广渠门桥 金世纪嘉园西南两居室出售",
    "total_price": "850万",
    "unit_price": "78,624元/平",
    "square": " 108.11平米 "
},
{
    "district": "东城",
    "name": "满五唯一住房 明厨明卫 采光好 视野宽阔",
    "total_price": "1150万",
    "unit_price": "118,876元/平",
    "square": " 96.74平米 "
},
```

```
{
    "district": "朝阳",
    "name": "晨光B区  南北三居  满五唯一  诚意出售  随时签约",
    "total_price": "880万",
    "unit_price": "65,692元/平",
    "square": " 133.96平米 "
},
{
    "district": "朝阳",
    "name": "北工大膳房南里 带车位 满五年唯一",
    "total_price": "578万",
    "unit_price": "49,200元/平",
    "square": " 117.48平米 "
},
{
    "district": "朝阳",
    "name": "满五年唯一，楼层好，采光好，无遮挡，户型方正大厨房",
    "total_price": "429万",
    "unit_price": "64,376元/平",
    "square": " 66.64平米 "
},
{
    "district": "朝阳",
    "name": "户型方正，三个方向采光，高层无遮挡",
    "total_price": "440万",
    "unit_price": "40,084元/平",
    "square": " 109.77平米 "
},
{
    "district": "朝阳",
    "name": "农光里小区 精装修 户型方正 近地铁",
    "total_price": "430万",
    "unit_price": "64,391元/平",
    "square": " 66.78平米 "
},
{
    "district": "朝阳",
    "name": "6号线 十里堡 南北通透两居 交通便利 看房方便",
    "total_price": "308万",
    "unit_price": "52,867元/平",
    "square": " 58.26平米 "
},
{
    "district": "西城",
    "name": "1997北礼士路 高楼层 带电梯正规大客厅 房龄新 精装修",
    "total_price": "760万",
    "unit_price": "126,456元/平",
    "square": " 60.1平米 "
},
{
    "district": "西城",
    "name": "丰汇园小区 3室2厅 东 西",
    "total_price": "2450万",
    "unit_price": "164,210元/平",
    "square": " 149.2平米 "
},
{
    "district": "西城",
    "name": "西城区 南北三居 明厨明卫 低楼层 采光充足 满五年",
    "total_price": "780万",
    "unit_price": "100,283元/平",
    "square": " 77.78平米 "
},
{
    "district": "西城",
    "name": "新德街35号院2000年建成钢混结构南北向两居中间层",
    "total_price": "1340万",
    "unit_price": "149,721元/平",
    "square": " 89.5平米 "
},
{
    "district": "西城",
    "name": "德胜里一区 双南向2居室 业主诚售",
    "total_price": "810万",
    "unit_price": "161,677元/平",
    "square": " 50.1平米 "
},
{
    "district": "西城",
    "name": "常青藤嘉园 2室1厅 东",
    "total_price": "915万",
    "unit_price": "103,064元/平",
    "square": " 88.78平米 "
},
{
    "district": "东城",
    "name": "商本满五年唯一，精品装修。中高层把边户视野采光棒",
    "total_price": "1100万",
    "unit_price": "121,346元/平",
    "square": " 90.65平米 "
},
{
    "district": "东城",
    "name": "民安小区东直门内北小街 2室1厅 东 西",
    "total_price": "780万",
    "unit_price": "101,194元/平",
    "square": " 77.08平米 "
},
{
    "district": "东城",
    "name": "东城安定门 安德路47号院 南北两居室",
    "total_price": "666万",
    "unit_price": "114,081元/平",
    "square": " 58.38平米 "
```

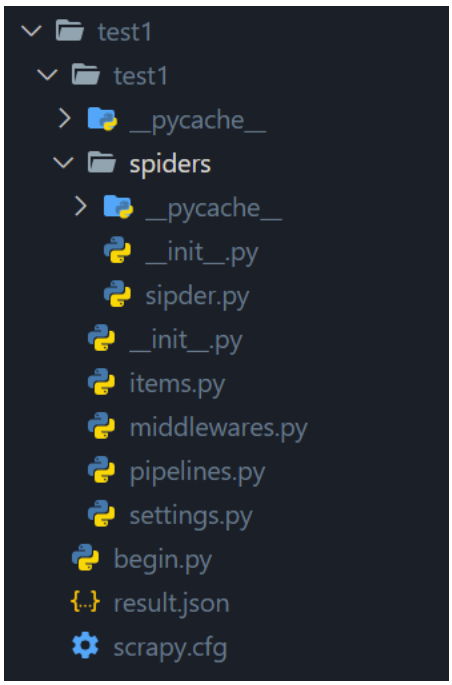
四、链家爬取

1.创建scrapy项目

创建目录，执行命令

```
Apache
1   scrapy startproject test1
```

文件结构如下：



其中，spider.py是主要的爬虫文件，middleware是中间件的文件。

pipelines是管道文件，需要编写主要的流程代码。

settings.py是相关配置文件

更具体的：

Plain Text

```
project_folder -- 项目文件夹名称
|
|——project_name -- 该项目的python模块，一般和项目文件夹名称相同
|  |
|  |——spider -- 放置spider代码的包，以后所有的爬虫，都存放在这个里面
|  |
|  |——items.py -- 用来存放爬虫怕写来的数据的模型
|  |
|  |——middlewares.py -- 用来存放各种中间件的文件
|  |
|  |——pipelines.py -- 用来对items里面提取的数据做进一步处理，如保存到本地磁盘等
|  |
|  |——settings.py -- 本爬虫的一些配置信息(如请求头、多久发送一次请求、ip代理池等)
|
|——scrapy.cfg -- 项目的配置文件
```

2.begin.py

这里的作用是快速启动爬虫。

Python

```python
from scrapy import cmdline
cmdline.execute("scrapy crawl bupt".split())
```

3.pipeline.py

pipeline是主要的爬虫流程管道。

分为open_spider,process_item,colse_spider三个部分。

open_spider是爬虫初始化。

process_item是对于每个得到的item具体要怎么做。

这里写的是将item的内容加入大json_str里面。

close_spider是指将结果输出到文件

Python

```python
from itemadapter import ItemAdapter
import json
import csv
from operator import itemgetter

from scrapy import item

class Test1Pipeline:
    def open_spider(self,spider):
        try:
            self.file=open("result.json","w",encoding="utf-8")
        except Exception as err:
            print(err)
        self.result=[]
    def process_item(self, item, spider):
        dict_item = dict(item)
        self.result.append(dict_item)
        return item
    def close_spider(self,spider):
        self.file.write(json.dumps(self.result,ensure_ascii=False))
        self.file.close()
        with open("result.csv","w",newline='',encoding="utf_8_sig") as csvfile:
            writer=csv.writer(csvfile)
            head=["name","teachers","school","stu_num"]
            writer.writerow(["课程名称","授课教师","学校","学生人数"])
            for course in self.result:
                writer.writerow(itemgetter(*head)(course))
```

4.middlewares.py

因为需要处理动态网页，所以要重写process_request的内容，通过selenium里面的webdriver来访问网页。

option是对于webdriver的一些配置。

option里面的设置例如禁用gpu加速，不打开沙盒模式，防止被识别为爬虫等等。

最后将request通过htmlresponse来得到响应，将响应传给parse进行处理。

Python

```python
def process_request(self, request, spider):
        # Called for each request that goes through the downloader
        # middleware.

        # Must either:
        # - return None: continue processing this request
        # - or return a Response object
        # - or return a Request object
        # - or raise IgnoreRequest: process_exception() methods of
        #   installed downloader middleware will be called
        option=webdriver.ChromeOptions()
        option.add_argument("--headless")
        option.add_argument("--disable-gpu")
        option.add_argument("no-sandbox")
        option.add_argument("disable-blink-features=AutomationControlled")
        option.add_experimental_option('excludeSwitches', ['enable-automation'])
        driver = webdriver.Chrome(chrome_options=option)
        driver.execute_cdp_cmd("Page.addScriptToEvaluateOnNewDocument", {
            "source": """
                Object.defineProperty(navigator, 'webdriver', {
                  get: () => undefined
                })
              """
        })
        driver.get(request.url)
        driver.implicitly_wait(5)
        content = driver.page_source
        driver.quit()
        ret=HtmlResponse(request.url, encoding="utf-8", body=content, request=re
quest)
        return ret
```

5.items.py

定义需要的Field

Python

```python
import scrapy
class Test1Item(scrapy.Item):
    # define the fields for your item here like:
    # name = scrapy.Field()
    name=scrapy.Field()
    school=scrapy.Field()
    teachers=scrapy.Field()
    stu_num=scrapy.Field()
    pass
```

6.spider.py

通过xpath路径找到对应的文字位置。

这里要特别注意处理一些不正常的数据。

有的课程只有参与人次，没有教师和学校。

处理方法是，用标签带有的class属性来进行获取

枚举每一个span，如果class的属性是teacher_con，就认为是描述教师。

如果是org_con就认为是描述学校，如果是空就认为是表示人数。

翻页的处理是通过每次修改url实现的。

AI TIME：ICML 2021-4

53 人次

AI TIME：ICML 2021-4

在实际运行过程中，会发现可能出现有的页面爬取不到的情况。

所以，每次爬取一个页面，统计里面的课程个数，如果是0，就将这个url重新放回到队列里面重新爬取。

特殊地，由于scrapy有去重机制，已经爬取的url不会重新进行爬取，所以要加入callback=*self.*parse禁用去重。

Python

```python
import scrapy
from scrapy.http import request, response
```

```
from test1.items import Test1Item
import time
class mySpider(scrapy.spiders.Spider):
    name="bupt"
    allowed_domains=["www.xuetangx.com/"]
    start_urls=["https://www.xuetangx.com/search?query=&org=&classify=1&type=&status=&page="]
    def start_requests(self):
        for url in self.start_urls:
            for page_num in range(1,53):
                yield scrapy.Request(url+str(page_num),callback=self.parse,dont_filter=True,cb_kwargs={"url":url+str(page_num)})
    def parse(self,responce,**kwargs):
        self.logger.info(kwargs.get("page_num"))
        item=Test1Item()
        cnt=0
        self.logger.info("=========================================")
        for each in responce.xpath("/html/body/div[1]/div/div[2]/div[1]/div[1]/div[2]/div[1]/*"):
            item.clear()
            item['name']=each.xpath("./div[2]/p[1]/span[1]/text()").get()
            for i in each.xpath("./div[2]/p[2]/*"):
                t=i.xpath("./@class").get()
                self.logger.info(t)
                if (t=="teacher_con"):
                    teacher=""
                    for a in i.xpath("./*"):
                        teacher+=a.xpath("./text()").get().replace(" ","")
                    item['teachers']=teacher
                elif (t=="org_con"):
                    item['school']=i.xpath("./span/text()").get()
                else:
                    item['stu_num']= str.strip(str(i.xpath("./text()").get()))
            if (item.get("teachers")==None):
                item["teachers"]="Unknown"
            if (item.get("school")==None):
                item["school"]="Unknown"
            if (item.get("stu_num")==None):
                item["stu_num"]="Unknown"
            if (item['name']):
                cnt+=1;
                yield(item)
        print("cnt=",cnt)
        print("url=",kwargs["url"])
        if (cnt==0):
            print("need restart")
            yield responce.follow(kwargs["url"],callback=self.parse,dont_filter=True,cb_kwargs=kwargs)
```

7.结果

	课程名称	授课教师	学校	学生人数
2	C++语言程序设计基础	郑莉 李超 等	清华大学	466920 人
3	数据结构(上)	邓俊辉	清华大学	456245 人
4	数据结构（下）	邓俊辉	清华大学	381158 人
5	操作系统	向勇 陈渝	清华大学	218913 人
6	Java程序设计	郑莉	清华大学	213235 人
7	网络技术与应用	沈鑫剡 俞海英 等	中国人民解放军陆军工程大学	199241 人
8	C++语言程序设计进阶	郑莉 李超 等	清华大学	135107 人
9	C程序设计案例教程（基础）	张莉	中国农业大学	117461 人
10	C程序设计案例教程（进阶）	张莉	中国农业大学	113000 人
11	大数据技术与应用	李军	清华大学	104733 人
12	软件工程	刘强 刘璘	清华大学	103441 人
13	计算机文化基础	李秀 姚瑞霞 等	清华大学	91339 人
14	程序设计基础	徐明星 王琦屏 等	清华大学	89956 人
15	人工智能原理	王文敏	北京大学	79370 人
16	组合数学	马昱春	清华大学	78962 人
17	大数据系统基础	王建民 徐葳 等	清华大学	78898 人
18	Office办公软件应用	史巧硕 朱怀忠 等	河北工业大学	75097 人
19	算法设计与分析	王振波	清华大学	73182 人
20	VC++面向对象与可视化程序设计（上）：Windows编程基础	黄维通	清华大学	71678 人
21	大数据平台核心技术	武永卫 姚文辉 等	清华大学	69720 人
22	VC++面向对象与可视化程序设计（下）：MFC编程基础	黄维通	清华大学	69490 人
23	Web前端攻城狮	刘强 吴亮 等	清华大学	64234 人
24	大学计算机教程	张莉 马钦	中国农业大学	63595 人
25	计算机应用基础	宋承继 王坤 等	陕西工业职业技术学院	62523 人
26	汇编语言程序设计	张悠慧 翟季冬	清华大学	60114 人
27	大学计算机基础	卫春芳	湖北大学	59131 人
28	学做小程序——基础篇	刘强 小程序慕课讲师	清华大学	57109 人
29	基于Linux的C++	乔林	清华大学	56122 人
30	微软亚洲研究院大数据系列讲座	洪小文 宋睿华 等	Microsoft	54241 人
31	计算机网络	袁华 杜广龙 等	华南理工大学	53107 人
32	大学计算机基础	徐红云 刘欣欣 等	华南理工大学	52762 人
33	JAVA程序设计进阶	许斌	清华大学	52681 人
34	大学计算机——计算思维的视角	郝兴伟	山东大学	52341 人
35	面向对象程序设计（C++）	黄震春 徐明星	清华大学	51817 人
36	计算几何	邓俊辉	清华大学	47962 人
37	R语言数据分析	艾新波	北京邮电大学	45128 人
38	Python程序设计基础	许志良	深圳信息职业技术学院	44729 人
39	大数据机器学习	袁春	清华大学	42247 人
40	数据库技术与程序设计	高裴裴 康介恢 等	南开大学	41801 人
41	程序设计基础（上）	赵宏 闫晓玉 等	南开大学	41222 人
42	单片机原理及应用	杨居义 王颖丽 等	绵阳职业技术学院	40849 人
43	Web开发技术	王成良 陈静 等	重庆大学	39558 人
44	ARM微控制器与嵌入式系统	曾鸣 薛涛 等	清华大学	39448 人