python 中的数据爬取

杨晨
学号 2021212171
北京邮电大学计算机学院

日期：2024 年 2 月 26 日

1 概述

1.1 实验内容

1. 爬取学堂在线的计算机类课程页面内容
   `https://www.xuetangx.com/search?query=&org=&classify=1&type=&status=&page=1`
   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
   `https://bj.lianjia.com/ershoufang/`
   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。

1.2 开发环境

- Windows10
- PyCharm 2023.2.4 (Professional Edition)

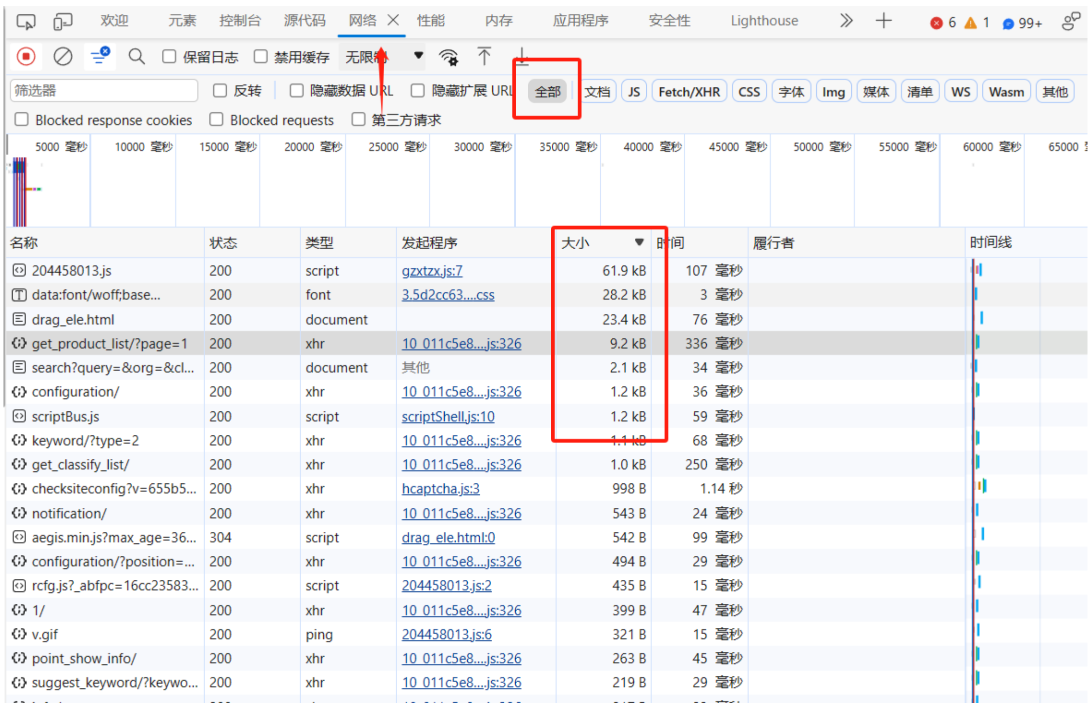
2 实验过程

2.1 学堂在线计算机类课程内容爬取

2.1.1 介绍

首先，按照 PPT 上的方法，用静态页面的方式进行爬取，但是发现爬虫关闭后，也没有爬取到任何数据

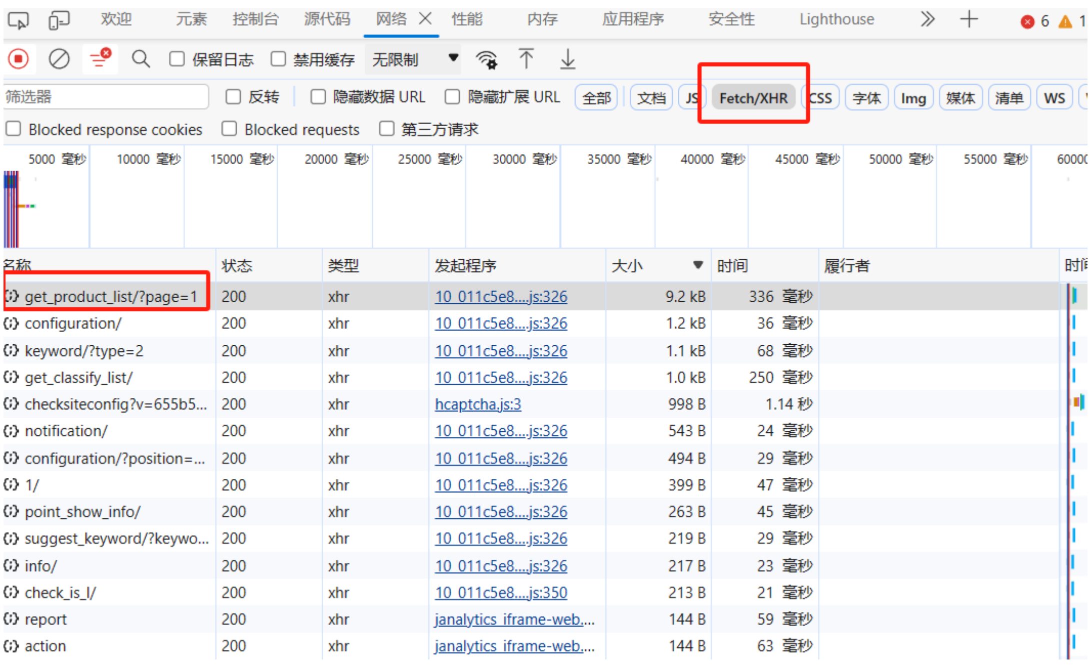
考虑网页是动态加载的，进入网页审查，发现以下结果


欢迎 元素 控制台 源代码 网络 性能 内存 应用程序 安全性 Lighthouse
保留日志 禁用缓存 无限制
筛选器 反转 隐藏数据 URL 隐藏扩展 URL 全部 文档 JS Fetch/XHR CSS 字体 Img 媒体 清单 WS Wasm 其他
Blocked response cookies Blocked requests 第三方请求

名称	状态	类型	发起程序	大小	时间
204458013.js	200	script	gzxtzx.js:7	61.9 kB	107 毫秒
data:font/woff;base...	200	font	3.5d2cc63....css	28.2 kB	3 毫秒
drag_ele.html	200	document		23.4 kB	76 毫秒
get_product_list/?page=1	200	xhr	10_011c5e8....js:326	9.2 kB	336 毫秒
search?query=&org=&cl...	200	document	其他	2.1 kB	34 毫秒
configuration/	200	xhr	10_011c5e8....js:326	1.2 kB	36 毫秒
scriptBus.js	200	script	scriptShell.js:10	1.2 kB	59 毫秒
keyword/?type=2	200	xhr	10_011c5e8....js:326	1.1 kB	68 毫秒
get_classify_list/	200	xhr	10_011c5e8....js:326	1.0 kB	250 毫秒
checksiteconfig?v=655b5...	200	xhr	hcaptcha.js:3	998 B	1.14 秒
notification/	200	xhr	10_011c5e8....js:326	543 B	24 毫秒
aegis.min.js?max_age=36...	304	script	drag_ele.html:0	542 B	99 毫秒
configuration/?position=...	200	xhr	10_011c5e8....js:326	494 B	29 毫秒
rcfg.js?_abfpc=16cc23583...	200	script	204458013.js:2	435 B	15 毫秒
1/	200	xhr	10_011c5e8....js:326	399 B	47 毫秒
v.gif	200	ping	204458013.js:6	321 B	15 毫秒
point_show_info/	200	xhr	10_011c5e8....js:326	263 B	45 毫秒
suggest_keyword/?keywo...	200	xhr	10_011c5e8....js:326	219 B	29 毫秒

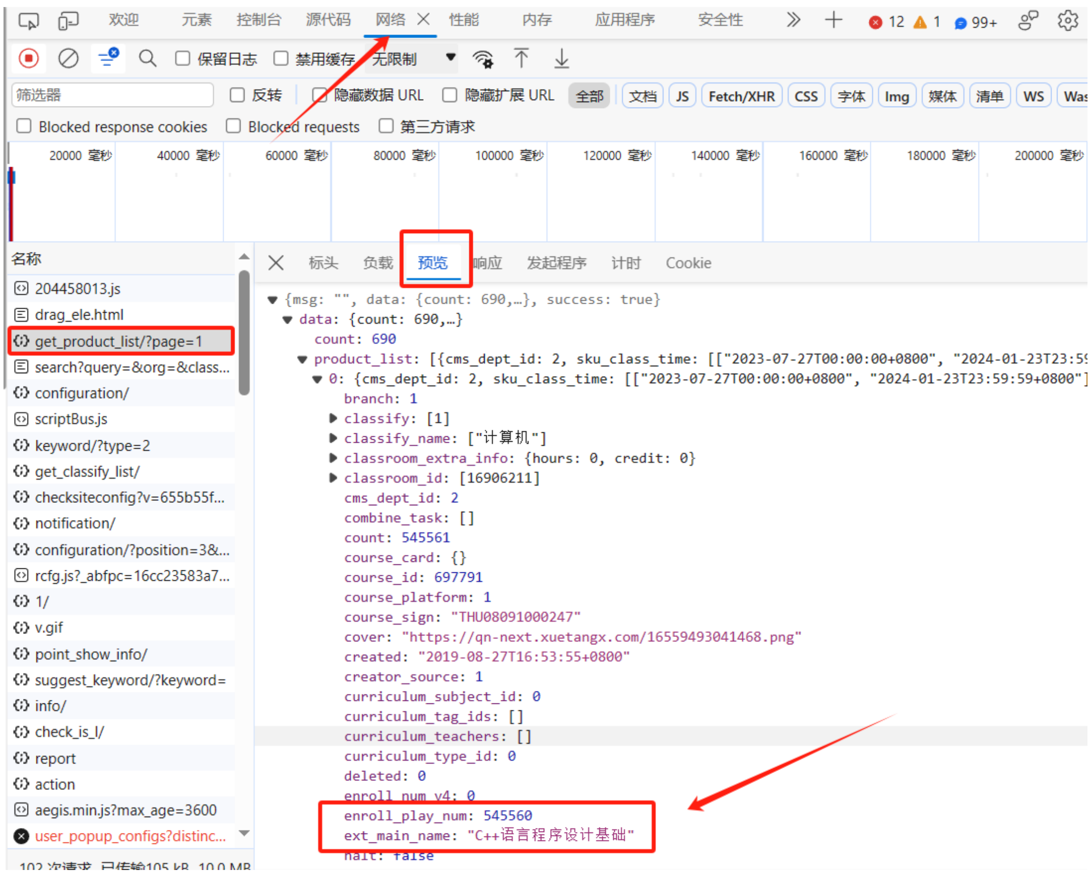
现在明确确实是动态页面。

然后进一步发现是 Ajax 类型，以下为截图


欢迎 元素 控制台 源代码 网络 性能 内存 应用程序 安全性 Lighthouse
保留日志 禁用缓存 无限制
筛选器 反转 隐藏数据 URL 隐藏扩展 URL 全部 文档 JS Fetch/XHR CSS 字体 Img 媒体 清单 WS
Blocked response cookies Blocked requests 第三方请求

名称	状态	类型	发起程序	大小	时间
get_product_list/?page=1	200	xhr	10_011c5e8....js:326	9.2 kB	336 毫秒
configuration/	200	xhr	10_011c5e8....js:326	1.2 kB	36 毫秒
keyword/?type=2	200	xhr	10_011c5e8....js:326	1.1 kB	68 毫秒
get_classify_list/	200	xhr	10_011c5e8....js:326	1.0 kB	250 毫秒
checksiteconfig?v=655b5...	200	xhr	hcaptcha.js:3	998 B	1.14 秒
notification/	200	xhr	10_011c5e8....js:326	543 B	24 毫秒
configuration/?position=...	200	xhr	10_011c5e8....js:326	494 B	29 毫秒
1/	200	xhr	10_011c5e8....js:326	399 B	47 毫秒
point_show_info/	200	xhr	10_011c5e8....js:326	263 B	45 毫秒
suggest_keyword/?keywo...	200	xhr	10_011c5e8....js:326	219 B	29 毫秒
info/	200	xhr	10_011c5e8....js:326	217 B	23 毫秒
check_is_l/	200	xhr	10_011c5e8....js:350	213 B	21 毫秒
report	200	xhr	janalytics_iframe-web....	144 B	59 毫秒
action	200	xhr	janalytics_iframe-web....	144 B	63 毫秒

点击打开 page 里的 product_lists 可以看见课程的相关信息，正是我们需要爬取的内容，如下所示



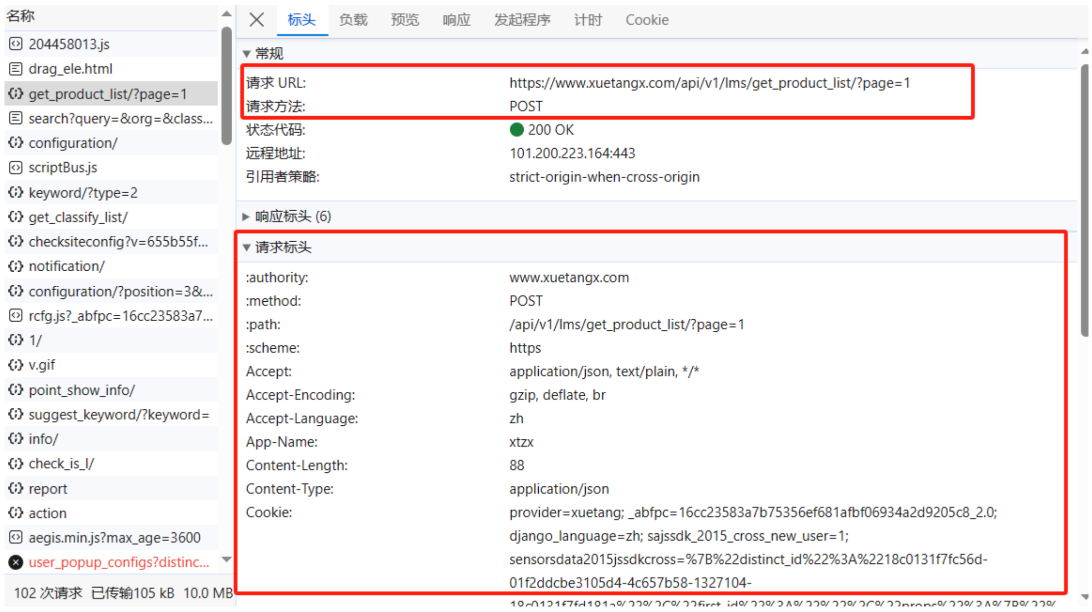
名称
204458013.js
drag_ele.html
get_product_list/?page=1
search?query=&org=&class...
configuration/
scriptBus.js
keyword/?type=2
get_classify_list/
checksiteconfig?v=655b55f...
notification/
configuration/?position=3&...
rcfg.js?_abfpc=16cc23583a7...
1/
v.gif
point_show_info/
suggest_keyword/?keyword=
info/
check_is_l/
report
action
aegis.min.js?max_age=3600
user_popup_configs?distinc...

```
▼ {msg: "", data: {count: 690,…}, success: true}
  ▼ data: {count: 690,…}
      count: 690
    ▼ product_list: [{cms_dept_id: 2, sku_class_time: [["2023-07-27T00:00:00+0800", "2024-01-23T23:5
      ▼ 0: {cms_dept_id: 2, sku_class_time: [["2023-07-27T00:00:00+0800", "2024-01-23T23:59:59+0800"]
          branch: 1
        ▶ classify: [1]
        ▶ classify_name: ["计算机"]
        ▶ classroom_extra_info: {hours: 0, credit: 0}
        ▶ classroom_id: [16906211]
          cms_dept_id: 2
          combine_task: []
          count: 545561
          course_card: {}
          course_id: 697791
          course_platform: 1
          course_sign: "THU08091000247"
          cover: "https://qn-next.xuetangx.com/16559493041468.png"
          created: "2019-08-27T16:53:55+0800"
          creator_source: 1
          curriculum_subject_id: 0
          curriculum_tag_ids: []
          curriculum_teachers: []
          curriculum_type_id: 0
          deleted: 0
          enroll_num_v4: 0
          enroll_play_num: 545560
          ext_main_name: "C++语言程序设计基础"
          halt: false
```

基于以上的发现明确进行动态页面的爬取

2.1.2 爬取方法

1. 找到爬取的链接以及请求的方式



名称
204458013.js
drag_ele.html
get_product_list/?page=1
search?query=&org=&class...
configuration/
scriptBus.js
keyword/?type=2
get_classify_list/
checksiteconfig?v=655b55f...
notification/
configuration/?position=3&...
rcfg.js?_abfpc=16cc23583a7...
1/
v.gif
point_show_info/
suggest_keyword/?keyword=
info/
check_is_l/
report
action
aegis.min.js?max_age=3600
user_popup_configs?distinc...
102 次请求 已传输105 kB 10.0 MB

▼ 常规

请求 URL:	https://www.xuetangx.com/api/v1/lms/get_product_list/?page=1
请求方法:	POST
状态代码:	● 200 OK
远程地址:	101.200.223.164:443
引用者策略:	strict-origin-when-cross-origin

▶ 响应标头 (6)

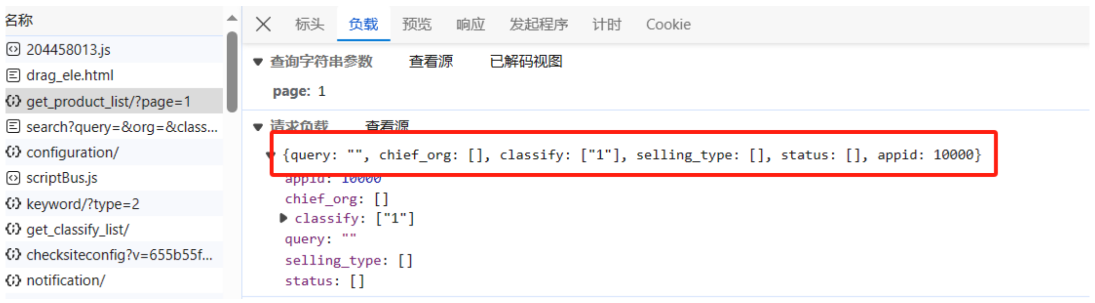
▼ 请求标头

:authority:	www.xuetangx.com
:method:	POST
:path:	/api/v1/lms/get_product_list/?page=1
:scheme:	https
Accept:	application/json, text/plain, */*
Accept-Encoding:	gzip, deflate, br
Accept-Language:	zh
App-Name:	xtzx
Content-Length:	88
Content-Type:	application/json
Cookie:	provider=xuetang; _abfpc=16cc23583a7b75356ef681afbf06934a2d9205c8_2.0; django_language=zh; sajssdk_2015_cross_new_user=1; sensorsdata2015jssdkcross=%7B%22distinct_id%22%3A%2218c0131f7fc56d-01f2ddcbe3105d4-4c657b58-1327104-

2. 构造 header，找到 Headers 选项，查看请求头
3. 将此代码赋值下来，填充 headers 的构造，代码如下所示

```
# 请求头
headers = {
    "authority": "www.xuetangx.com",
    "method": "POST",
    "scheme": "https",
    "accept": "application/json, text/plain, */*",
    "accept-encoding": "gzip, deflate, br",
    "accept-language": "zh",
    "content-type": "application/json",
    "cookie": "provider=xuetang; django_language=zh",
    "django-language": "zh",
    "origin": "https://www.xuetangx.com",
    "Referer": "https://www.xuetangx.com/search?query=&org=&classify=1&type=&
        status=&page=1",
    "sec-fetch-dest": "empty",
    "sec-fetch-mode": "cors",
    "sec-fetch-site": "same-origin",
    "user-agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit
        /537.36 "
    "(KHTML, like Gecko) Chrome/119.0.0.0 Safari/537.36",
    "X-Client": "web",
    "Xtbz": "xt",
}
```

4. 除了请求头此外，还需要 post 的提交表格数据



```
page: 1
{query: "", chief_org: [], classify: ["1"], selling_type: [], status: [], appid: 10000}
appid: 10000
chief_org: []
classify: ["1"]
query: ""
selling_type: []
status: []
```

代码如下

```
data = {
    "query": "",
    "chief_org": [],
    "classify": ["1"],
    "selling_type": [],
    "status": [],
    "appid": 10000,
}
```

5. 在前面第一步已经分析，采取的是 post 的页面的请求方法，那么借助`FormRequest`函数即可实现 post 请求。重写`tart_request(self)`函数，然后经过这个函数不断的循环发送请

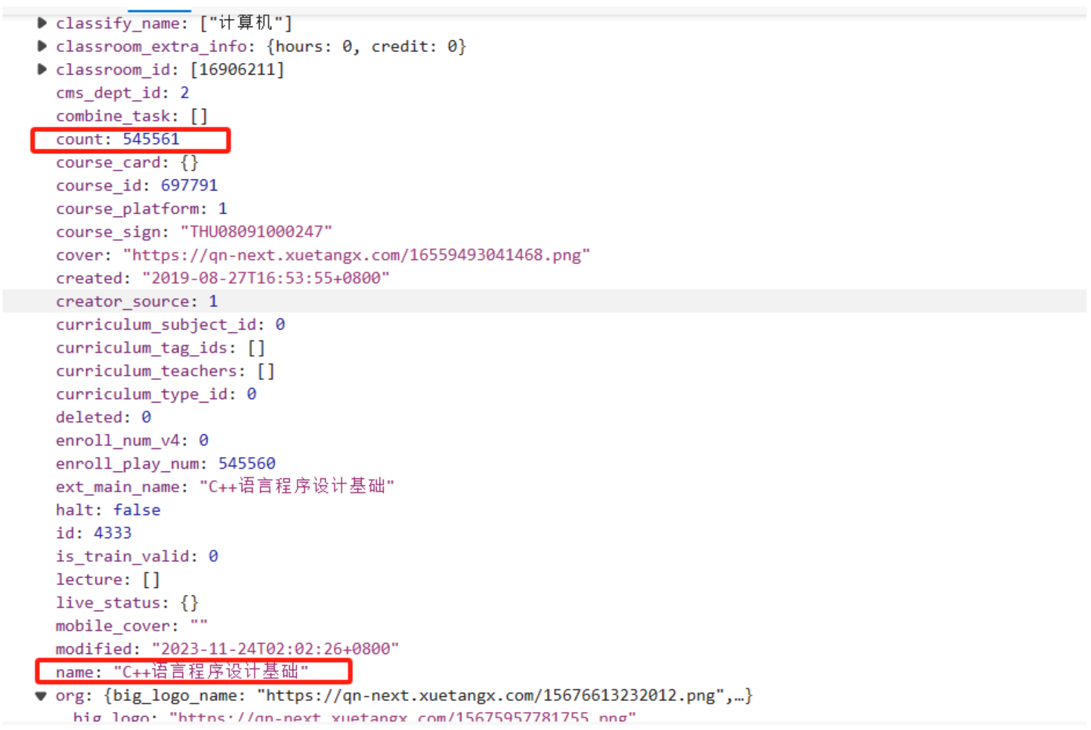
求，该函数代码实现如下

```python
def start_requests(self):
    for page_num in range(1, 70):
        yield scrapy.FormRequest(
            url="https://www.xuetangx.com/api/v1/lms/get_product_list/?page={}
                ".format(
                page_num
            ),
            headers=self.headers,
            method="POST",
            body=json.dumps(self.data),
            callback=self.parse,
        )
```

6. 接着就是解析内容的阶段根据我们的需求，需要爬取课程名称，老师，所属学校以及课程人数，那么将 items.py 文件如下实现：

```python
class XuetangxItem(scrapy.Item):
    # 课程名称、教师、学校、学生人数
    course_name = scrapy.Field()
    teacher = scrapy.Field()
    school = scrapy.Field()
    student_num = scrapy.Field()
```

7. 那么对于 response 返回的内容，使用 json.loads 处理获取到的 json 文件，json.loads() 函数是将 json 格式数据转换为字典。网页 product_list 部分显示如下：



```
▶ classify_name: ["计算机"]
▶ classroom_extra_info: {hours: 0, credit: 0}
▶ classroom_id: [16906211]
  cms_dept_id: 2
  combine_task: []
  count: 545561
  course_card: {}
  course_id: 697791
  course_platform: 1
  course_sign: "THU08091000247"
  cover: "https://qn-next.xuetangx.com/16559493041468.png"
  created: "2019-08-27T16:53:55+0800"
  creator_source: 1
  curriculum_subject_id: 0
  curriculum_tag_ids: []
  curriculum_teachers: []
  curriculum_type_id: 0
  deleted: 0
  enroll_num_v4: 0
  enroll_play_num: 545560
  ext_main_name: "C++语言程序设计基础"
  halt: false
  id: 4333
  is_train_valid: 0
  lecture: []
  live_status: {}
  mobile_cover: ""
  modified: "2023-11-24T02:02:26+0800"
  name: "C++语言程序设计基础"
▼ org: {big_logo_name: "https://qn-next.xuetangx.com/15676613232012.png",…}
    big_logo: "https://qn-next.xuetangx.com/15675957781755.png"
```

那么对进行 product_list 内容如下方式进行提取

```
def parse(self, response):
    for product in json.loads(response.body)["data"]["product_list"]:
        item = XuetangxItem()
        item["course_name"] = "课程名称: " + product["name"]
        item["teacher"] = "课程讲师: "
        for teacher in product["teacher"]:  # 一个课程可能有多个讲师
            item["teacher"] += teacher["name"] + " "
        item["school"] = "开课学校: " + product["org"]["name"]
        item["student_num"] = "学生人数: " + str(product["count"])
```

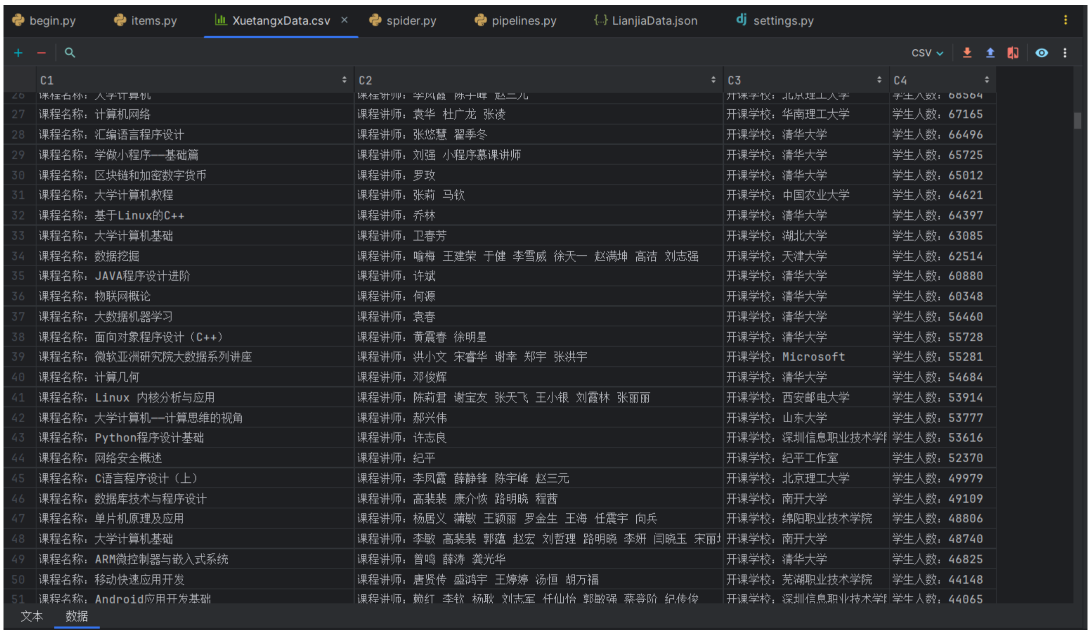
8. 然后 pipelines 进行将数据写入 csv 文件，那么 pipelines.py 文件如下所示

```
import csv

class XuetangxPipeline:
    def __init__(self):
        self.file = open('XuetangxData.csv', 'w+', encoding='utf-8', newline='
            ')
        self.writer = csv.writer(self.file)

    def process_item(self, item, spider):
        self.writer.writerow([item['course_name'], item['teacher'], item['
            school'], item['student_num']])
        return item

    def close_spider(self, spider):
        self.file.close()
```

9. 最后当然需要将管道打开，那么 settings.py 文件如下所示

```
ROBOTSTXT_OBEY = False
# ITEM_PIPELINES = {'myproject.pipelines.MyPipeline': 300}

ITEM_PIPELINES = {'myproject.pipelines.XuetangxPipeline': 300}
```

2.1.3 爬取结果



	C1	C2	C3	C4
26	课程名称：大学计算机	课程讲师：李凤霞 陈宇峰 赵三元	开课学校：北京理工大学	学生人数：68564
27	课程名称：计算机网络	课程讲师：袁华 杜广龙 张凌	开课学校：华南理工大学	学生人数：67165
28	课程名称：汇编语言程序设计	课程讲师：张悠慧 翟季冬	开课学校：清华大学	学生人数：66496
29	课程名称：学做小程序——基础篇	课程讲师：刘强 小程序慕课讲师	开课学校：清华大学	学生人数：65725
30	课程名称：区块链和加密数字货币	课程讲师：罗玫	开课学校：清华大学	学生人数：65012
31	课程名称：大学计算机教程	课程讲师：张莉 马钦	开课学校：中国农业大学	学生人数：64621
32	课程名称：基于Linux的C++	课程讲师：乔林	开课学校：清华大学	学生人数：64397
33	课程名称：大学计算机基础	课程讲师：卫春芳	开课学校：湖北大学	学生人数：63085
34	课程名称：数据挖掘	课程讲师：喻梅 王建荣 于健 李雪威 徐天一 赵满坤 高洁 刘志强	开课学校：天津大学	学生人数：62514
35	课程名称：JAVA程序设计进阶	课程讲师：许斌	开课学校：清华大学	学生人数：60880
36	课程名称：物联网概论	课程讲师：何源	开课学校：清华大学	学生人数：60348
37	课程名称：大数据机器学习	课程讲师：袁春	开课学校：清华大学	学生人数：56460
38	课程名称：面向对象程序设计（C++）	课程讲师：黄震春 徐明星	开课学校：清华大学	学生人数：55728
39	课程名称：微软亚洲研究院大数据系列讲座	课程讲师：洪小文 宋睿华 谢幸 郑宇 张洪宇	开课学校：Microsoft	学生人数：55281
40	课程名称：计算几何	课程讲师：邓俊辉	开课学校：清华大学	学生人数：54684
41	课程名称：Linux 内核分析与应用	课程讲师：陈莉君 谢宝友 张天飞 王小银 刘霞林 张丽丽	开课学校：西安邮电大学	学生人数：53914
42	课程名称：大学计算机——计算思维的视角	课程讲师：郝兴伟	开课学校：山东大学	学生人数：53777
43	课程名称：Python程序设计基础	课程讲师：许志良	开课学校：深圳信息职业技术学	学生人数：53616
44	课程名称：网络安全概述	课程讲师：纪平	开课学校：纪平工作室	学生人数：52370
45	课程名称：C语言程序设计（上）	课程讲师：李凤霞 薛静锋 陈宇峰 赵三元	开课学校：北京理工大学	学生人数：49979
46	课程名称：数据库技术与程序设计	课程讲师：高裴裴 康介恢 路明晓 程茜	开课学校：南开大学	学生人数：49109
47	课程名称：单片机原理及应用	课程讲师：杨居义 蒲敏 王颖丽 罗金生 王海 任震宇 向兵	开课学校：绵阳职业技术学院	学生人数：48806
48	课程名称：大学计算机基础	课程讲师：李敏 高裴裴 郭蕴 赵宏 刘哲理 路明晓 李妍 闫晓玉 宋丽	开课学校：南开大学	学生人数：48740
49	课程名称：ARM微控制器与嵌入式系统	课程讲师：曾鸣 薛涛 龚光华	开课学校：清华大学	学生人数：46825
50	课程名称：移动快速应用开发	课程讲师：唐贤传 盛鸿宇 王婷婷 汤恒 胡万福	开课学校：芜湖职业技术学院	学生人数：44148
51	课程名称：Android应用开发基础	课程讲师：赖红 李钦 杨耿 刘志军 任仙怡 郭敏强 蔡誉阶 纪传俊	开课学校：深圳信息职业技术学	学生人数：44065

图 1: 学堂在线爬取结果

2.2 链家官网北京二手房数据爬取

2.2.1 介绍

链家的官网是静态页面，相比之下要容易处理
审查页面，找到想提取的楼盘名称、总价、平米数、单价的 xpath，进行提取即可
此外，对于第 1 页，第 2 页，网页的 url 具有相似特征/pg1，/pg2；这使得爬取较为方便

2.2.2 爬取方法

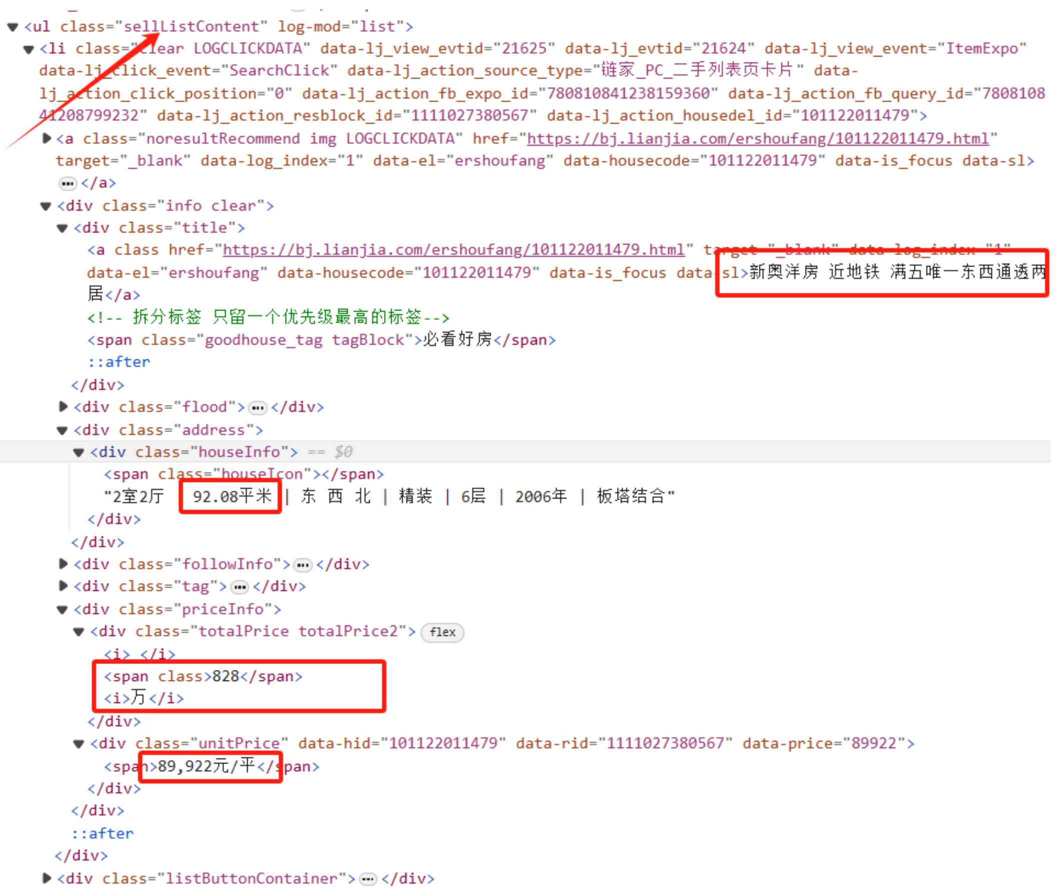
1. 构造 spider

```
class LianjiaSpider(scrapy.spiders.Spider):
    name = "lianjia"
    allowed_domains = ["lianjia.com"]
    start_urls = [
        "https://bj.lianjia.com/ershoufang/dongcheng/pg1/",
        "https://bj.lianjia.com/ershoufang/xicheng/pg1/",
        "https://bj.lianjia.com/ershoufang/chaoyang/pg1/",
        "https://bj.lianjia.com/ershoufang/haidian/pg1/",
    ]
```

2. 需要爬取的信息有：楼盘名称、总价、平米数、单价。那么 items.py 中的代码如下

```
class LianjiaItem(scrapy.Item):
    # 楼盘名称、总价、平米数、单价
    name = scrapy.Field()
    price = scrapy.Field()
    area = scrapy.Field()
    unit_price = scrapy.Field()
```

3. 审查页面，筛选想提取的信息，如下



```
▼<ul class="sellListContent" log-mod="list">
  ▼<li class="clear LOGCLICKDATA" data-lj_view_evtid="21625" data-lj_evtid="21624" data-lj_view_event="ItemExpo"
   data-lj_click_event="SearchClick" data-lj_action_source_type="链家_PC_二手列表页卡片" data-
   lj_action_click_position="0" data-lj_action_fb_expo_id="780810841238159360" data-lj_action_fb_query_id="7808108
   41208799232" data-lj_action_resblock_id="1111027380567" data-lj_action_housedel_id="101122011479">
    ▶<a class="noresultRecommend img LOGCLICKDATA" href="https://bj.lianjia.com/ershoufang/101122011479.html"
     target="_blank" data-log_index="1" data-el="ershoufang" data-housecode="101122011479" data-is_focus data-sl>
     ⋯</a>
    ▼<div class="info clear">
      ▼<div class="title">
         <a class href="https://bj.lianjia.com/ershoufang/101122011479.html" target="_blank" data-log_index="1"
         data-el="ershoufang" data-housecode="101122011479" data-is_focus data-sl>新奥洋房 近地铁 满五唯一东西通透两
         居</a>
         <!-- 拆分标签 只留一个优先级最高的标签-->
         <span class="goodhouse_tag tagBlock">必看好房</span>
         ::after
       </div>
      ▶<div class="flood">⋯</div>
      ▼<div class="address">
        ▼<div class="houseInfo"> == $0
           <span class="houseIcon"></span>
           "2室2厅  92.08平米 | 东 西 北 | 精装 | 6层 | 2006年 | 板塔结合"
         </div>
       </div>
      ▶<div class="followInfo">⋯</div>
      ▶<div class="tag">⋯</div>
      ▼<div class="priceInfo">
        ▼<div class="totalPrice totalPrice2"> flex
           <i> </i>
           <span class>828</span>
           <i>万</i>
         </div>
        ▼<div class="unitPrice" data-hid="101122011479" data-rid="1111027380567" data-price="89922">
           <span>89,922元/平</span>
         </div>
       </div>
       ::after
     </div>
    ▶<div class="listButtonContainer">⋯</div>
```

在每个 sellListContent 类下面，每个 li 代表一个房子的信息。进行提取的代码如下

```
def parse(self, response):
    item = LianjiaItem()
    distinct = response.url.split("/")[4]
    page = response.url.split("/")[5]
    for each in response.xpath('//ul[@class="sellListContent"]/li'):
        item["name"] = "楼盘名称: " + each.xpath("div/div/a/text()").get()
        price_value = each.xpath(
            "div/div[@class='priceInfo']/div[@class='totalPrice totalPrice2']/
                span/text()"
        ).get()
        price_unit = each.xpath(
            "div/div[@class='priceInfo']/div[@class='totalPrice totalPrice2']/
                i[last()]/text()"
        ).get()
        item["price"] = "总价: " + f"{price_value}{price_unit}"
        area_text = each.xpath(
            ".//div[@class='address']/div[@class='houseInfo']/text()"
        ).get()
        match = re.search(r"(\d+(\.\d+)?)平米", area_text)
        if match:
            item["area"] = "平米数: " + match.group(1) + "平米"
        else:
            item["area"] = "平米数: " + "unknown"
        item["unit_price"] = "单价: " + each.xpath(
            "div/div[@class='priceInfo']/div[@class='unitPrice']/span/text()"
        ).get()
        if item["name"] and item["price"] and item["area"] and item["
            unit_price"]:
            yield item
```

为了便于提取平米数，使用了正则表达式 re 库

4. 因为需要提取前 5 页的内容，所以需要构造下一页的 URL，并发送一个新的请求，回调函数为自身的parse方法，实现翻页功能

```
if page != "pg5":
    next_page = int(page[2]) + 1
    next_url = "https://bj.lianjia.com/ershoufang/{}/pg{}/".format(distinct,
        next_page)
    yield scrapy.Request(next_url, callback=self.parse)
```

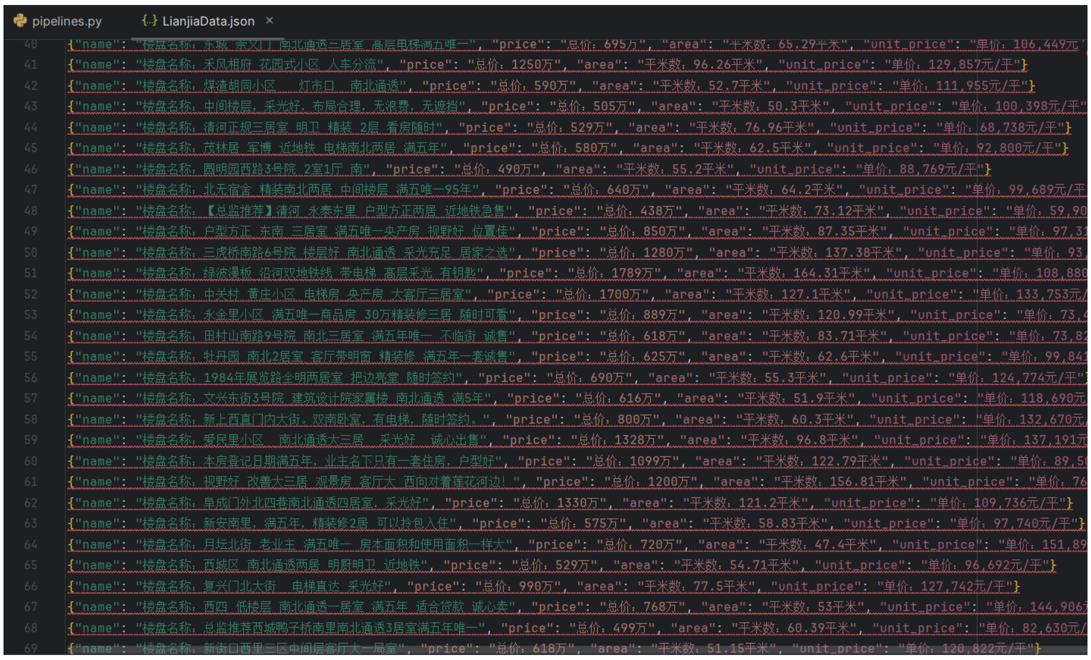
5. 然后 pipelines 进行将数据写入 json 文件，那么 pipelines.py 文件如下所示

```
class LianjiaPipeline:
    def __init__(self):
        self.file = open("LianjiaData.json", "w+", encoding="utf-8")
        self.writer = csv.writer(self.file)

    def process_item(self, item, spider):
        dict_item = dict(item)
        json_str = json.dumps(dict_item, ensure_ascii=False) + "\n"
        self.file.write(json_str)
        return item

    def close_spider(self, spider):
        self.file.close()
```

6. 最后需要将管道打开，那么 settings.py 文件如下所示

```
ROBOTSTXT_OBEY = False
# ITEM_PIPELINES = {'myproject.pipelines.MyPipeline': 300}

# ITEM_PIPELINES = {'myproject.pipelines.XuetangxPipeline': 300}

ITEM_PIPELINES = {'myproject.pipelines.LianjiaPipeline': 300}
```

2.2.3 爬取结果


pipelines.py　LianjiaData.json

```
{"name": "楼盘名称：东城 崇文门 南北通透三居室 高层电梯满五唯一", "price": "总价：695万", "area": "平米数：65.29平米", "unit_price": "单价：106,449元
{"name": "楼盘名称：禾风相府 花园式小区 人车分流", "price": "总价：1250万", "area": "平米数：96.26平米", "unit_price": "单价：129,857元/平"}
{"name": "楼盘名称：煤渣胡同小区    灯市口   南北通透", "price": "总价：590万", "area": "平米数：52.7平米", "unit_price": "单价：111,955元/平"}
{"name": "楼盘名称：中间楼层，采光好，布局合理，无浪费，无遮挡", "price": "总价：505万", "area": "平米数：50.3平米", "unit_price": "单价：100,398元/平"
{"name": "楼盘名称：清河正规三居室 明卫 精装 2层 看房随时", "price": "总价：529万", "area": "平米数：76.96平米", "unit_price": "单价：68,738元/平"}
{"name": "楼盘名称：茂林居 军博 近地铁 电梯南北两居 满五年", "price": "总价：580万", "area": "平米数：62.5平米", "unit_price": "单价：92,800元/平"}
{"name": "楼盘名称：圆明园西路3号院 2室1厅 南", "price": "总价：490万", "area": "平米数：55.2平米", "unit_price": "单价：88,769元/平"}
{"name": "楼盘名称：北无宿舍 精装南北两居 中间楼层 满五唯一95年", "price": "总价：640万", "area": "平米数：64.2平米", "unit_price": "单价：99,689元/平
{"name": "楼盘名称：【总监推荐】清河 永泰东里 户型方正两居 近地铁急售", "price": "总价：438万", "area": "平米数：73.12平米", "unit_price": "单价：59,90
{"name": "楼盘名称：户型方正 东南 三居室 满五唯一央产房 视野好 位置佳", "price": "总价：850万", "area": "平米数：87.35平米", "unit_price": "单价：97,31
{"name": "楼盘名称：三虎桥南路6号院 楼层好 南北通透 采光充足 居家之选", "price": "总价：1280万", "area": "平米数：137.38平米", "unit_price": "单价：93,
{"name": "楼盘名称：绿波漫板 沿河双地铁线 带电梯 高层采光 有钥匙", "price": "总价：1789万", "area": "平米数：164.31平米", "unit_price": "单价：108,880
{"name": "楼盘名称：中关村 黄庄小区 电梯房 央产房 大客厅三居室", "price": "总价：1700万", "area": "平米数：127.1平米", "unit_price": "单价：133,753元/
{"name": "楼盘名称：永金里小区 满五唯一商品房 30万精装修三居 随时可看", "price": "总价：889万", "area": "平米数：120.99平米", "unit_price": "单价：73,4
{"name": "楼盘名称：田村山南路9号院 南北三居室 满五年唯一 不临街 诚售", "price": "总价：618万", "area": "平米数：83.71平米", "unit_price": "单价：73,82
{"name": "楼盘名称：牡丹园 南北2居室 客厅带明窗 精装修 满五年一套诚售", "price": "总价：625万", "area": "平米数：62.6平米", "unit_price": "单价：99,841
{"name": "楼盘名称：1984年展览路全明两居室 把边亮堂 随时签约", "price": "总价：690万", "area": "平米数：55.3平米", "unit_price": "单价：124,774元/平"
{"name": "楼盘名称：文兴东街3号院 建筑设计院家属楼 南北通透 满5年", "price": "总价：616万", "area": "平米数：51.9平米", "unit_price": "单价：118,690元
{"name": "楼盘名称：新上西直门内大街。双南卧室，有电梯，随时签约。", "price": "总价：800万", "area": "平米数：60.3平米", "unit_price": "单价：132,670元
{"name": "楼盘名称：爱民里小区   南北通透大三居   采光好   诚心出售", "price": "总价：1328万", "area": "平米数：96.8平米", "unit_price": "单价：137,191元
{"name": "楼盘名称：本房登记日期满五年，业主名下只有一套住房，户型好", "price": "总价：1099万", "area": "平米数：122.79平米", "unit_price": "单价：89,50
{"name": "楼盘名称：视野好 改善大三居 观景房 客厅大 西向对着莲花河边！", "price": "总价：1200万", "area": "平米数：156.81平米", "unit_price": "单价：76
{"name": "楼盘名称：阜成门外北四巷南北通透四居室，采光好", "price": "总价：1330万", "area": "平米数：121.2平米", "unit_price": "单价：109,736元/平"}
{"name": "楼盘名称：新安南里，满五年，精装修2居 可以拎包入住", "price": "总价：575万", "area": "平米数：58.83平米", "unit_price": "单价：97,740元/平"}
{"name": "楼盘名称：月坛北街 老业主 满五唯一 房本面积和使用面积一样大", "price": "总价：720万", "area": "平米数：47.4平米", "unit_price": "单价：151,89
{"name": "楼盘名称：西城区 南北通透两居 明厨明卫 近地铁", "price": "总价：529万", "area": "平米数：54.71平米", "unit_price": "单价：96,692元/平"}
{"name": "楼盘名称：复兴门北大街   电梯直达 采光好", "price": "总价：990万", "area": "平米数：77.5平米", "unit_price": "单价：127,742元/平"}
{"name": "楼盘名称：西四 低楼层 南北通透一居室 满五年 适合贷款 诚心卖", "price": "总价：768万", "area": "平米数：53平米", "unit_price": "单价：144,906
{"name": "楼盘名称：总监推荐西城鸭子桥南里南北通透3居室满五年唯一", "price": "总价：499万", "area": "平米数：60.39平米", "unit_price": "单价：82,630元/
{"name": "楼盘名称：新街口西里三区中间层客厅大一局室", "price": "总价：618万", "area": "平米数：51.15平米", "unit_price": "单价：120,822元/平"}
```

图 2: 链家爬取结果

3 附录：spider.py 完整代码

3.1 爬取学堂在线

```
import scrapy
from ..items import XuetangxItem
import json


class XuetangxSpider(scrapy.spiders.Spider):
    name = "xuetangx"
    allowed_domains = ["xuetangx.com"]
    # 请求头
    headers = {
        "authority": "www.xuetangx.com",
        "method": "POST",
        "scheme": "https",
        "accept": "application/json, text/plain, */*",
        "accept-encoding": "gzip, deflate, br",
        "accept-language": "zh",
        "content-type": "application/json",
        "cookie": "provider=xuetang; django_language=zh",
        "django-language": "zh",
        "origin": "https://www.xuetangx.com",
```

```python
        "Referer": "https://www.xuetangx.com/search?query=&org=&classify=1&type=&
            status=&page=1",
        "sec-fetch-dest": "empty",
        "sec-fetch-mode": "cors",
        "sec-fetch-site": "same-origin",
        "user-agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
             "
        "(KHTML, like Gecko) Chrome/119.0.0.0 Safari/537.36",
        "X-Client": "web",
        "Xtbz": "xt",
    }
    data = {
        "query": "",
        "chief_org": [],
        "classify": ["1"],
        "selling_type": [],
        "status": [],
        "appid": 10000,
    }
    download_delay = 1

    def start_requests(self):
        for page_num in range(1, 70):
            yield scrapy.FormRequest(
                url="https://www.xuetangx.com/api/v1/lms/get_product_list/?page={}"
                    .format(
                    page_num
                ),
                headers=self.headers,
                method="POST",
                body=json.dumps(self.data),
                callback=self.parse,
            )

    def parse(self, response):
        for product in json.loads(response.body)["data"]["product_list"]:
            item = XuetangxItem()
            item["course_name"] = "课程名称: " + product["name"]
            item["teacher"] = "课程讲师: "
            for teacher in product["teacher"]:  # 一个课程可能有多个讲师
                item["teacher"] += teacher["name"] + " "
            item["school"] = "开课学校: " + product["org"]["name"]
            item["student_num"] = "学生人数: " + str(product["count"])

            if (
                item["course_name"]
                and item["teacher"]
                and item["school"]
                and item["student_num"]
            ):
                yield item
```

3.2 爬取链家

```
import scrapy
from ..items import LianjiaItem
import re


class LianjiaSpider(scrapy.spiders.Spider):
    name = "lianjia"
    allowed_domains = ["lianjia.com"]
    start_urls = [
        "https://bj.lianjia.com/ershoufang/dongcheng/pg1/",
        "https://bj.lianjia.com/ershoufang/xicheng/pg1/",
        "https://bj.lianjia.com/ershoufang/chaoyang/pg1/",
        "https://bj.lianjia.com/ershoufang/haidian/pg1/",
    ]

    def parse(self, response):
        item = LianjiaItem()
        distinct = response.url.split("/")[4]
        page = response.url.split("/")[5]
        for each in response.xpath('//ul[@class="sellListContent"]/li'):
            item["name"] = "楼盘名称: " + each.xpath("div/div/a/text()").get()
            price_value = each.xpath(
                "div/div[@class='priceInfo']/div[@class='totalPrice totalPrice2']/
                    span/text()"
            ).get()
            price_unit = each.xpath(
                "div/div[@class='priceInfo']/div[@class='totalPrice totalPrice2']/i
                    [last()]/text()"
            ).get()
            item["price"] = "总价: " + f"{price_value}{price_unit}"
            area_text = each.xpath(
                ".//div[@class='address']/div[@class='houseInfo']/text()"
            ).get()
            match = re.search(r"(\d+(\.\d+)?)平米", area_text)
            if match:
                item["area"] = "平米数: " + match.group(1) + "平米"
            else:
                item["area"] = "平米数: " + "unknown"
            item["unit_price"] = "单价: " + each.xpath(
                "div/div[@class='priceInfo']/div[@class='unitPrice']/span/text()"
            ).get()
            if item["name"] and item["price"] and item["area"] and item["unit_price
                "]:
                yield item

        if page != "pg5":
            next_page = int(page[2]) + 1
            next_url = "https://bj.lianjia.com/ershoufang/{}/pg{}/".format(distinct
                , next_page)
            yield scrapy.Request(next_url, callback=self.parse)
```