

# Python数据抓取与爬虫基础 I



## 爬虫知识储备与房租信息抓取



九章算法-Justin

United States

林平之 老师



Scan the QR code to add me on WeChat

**对课程有疑问？购买时遇到问题？获取更多优惠信息？  
扫一扫，在线咨询**

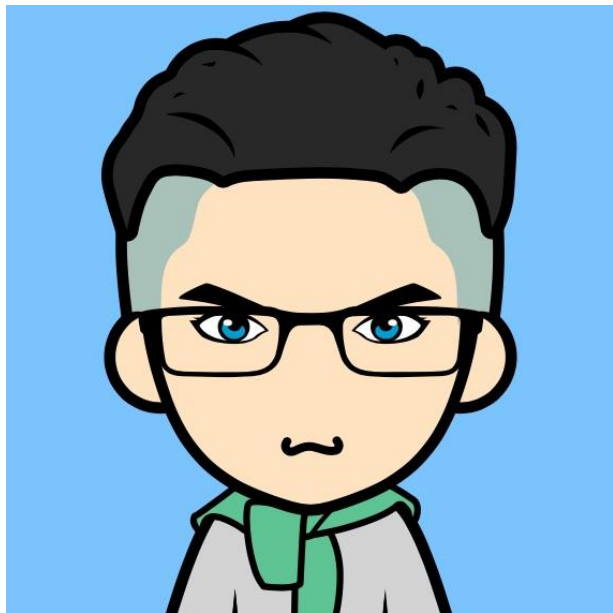
微信: [ninechapter](#)

知乎专栏: <http://zhuanlan.zhihu.com/jiuzhang>

微博: <http://www.weibo.com/ninechapter>

官网: [www.jiuzhang.com](http://www.jiuzhang.com)

九章课程不提供视频，也严禁录制视频的侵权行为  
否则将追求法律责任和经济赔偿  
请不要缺课



林老师

全国算法竞赛一等奖  
国内TOP2名校毕业  
参加国家信息学竞赛NOI  
前FLAG工程师  
拥有丰富的面试经验

# 什么是爬虫？

- Spider 形象的说法，沿着网络抓取周围的事物
- Google, Baidu 通过爬取的数据，存储数据，最后你通过搜索，得到展示的数据
- 学习机器学习的我们，爬虫就是获取数据的一种途径，网上的信息成百上千，只要我们懂爬虫，都能轻松获取数据



- 使用Python的requests相关包发送请求
- Python之Scrapy
- 实现第一个Spider抓取链家租房信息

# 如何使用Python请求这个URL的内容？

HTTP的请求包含哪些？

Python发送网络请求的常见模块：

- requests 包
- import requests



```
import requests

response = requests.get('https://www.jiuzhang.com')
print(type(response))
print(response.status_code)
print(response.encoding)
print(response.text)
```

我们看到的结果是这样的：

```
<class 'requests.models.Response'>
200
utf-8
<!DOCTYPE html>
<html>
<head>
.....
```

如何理解200, 如何理解404, 403, 500等等？

<https://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html>

如何理解这个utf-8 ？



在<http://www.turingapi.com/> 图灵机器人上注册一个机器人

TURING ROBOT

帮助中心 意见反馈

 机器人管理

 图灵机器人 今日聊天数量: 5 apikey: ad84...  机器人版本: 免费版

创建机器人

获取apiKey, 如:ad844aee\*\*\*\*\*

api接入

apikey: ad844aee

密钥: ☐

[api使用文档](#)

查看了解机器人接入文档：

<https://www.kancloud.cn/turing/www-tuling123-com/718226>

接口地址

<http://openapi.tuling123.com/openapi/api/v2>

apiKey 是重要参数

```
{
  "reqType": 0,
  "perception": {
    "inputText": {
      "text": "附近的酒店"
    },
    "inputImage": {
      "url": "imageUrl"
    },
    "selfInfo": {
      "location": {
        "city": "北京",
        "province": "北京",
        "street": "信息路"
      }
    }
  },
  "userInfo": {
    "apiKey": "",
    "userId": ""
  }
}
```

## 使用requests中的post向api发送数据

```
def ask(question):
    url = 'http://openapi.tuling123.com/openapi/api/v2'
    data = {
        'perception': {
            'inputText': {
                'text': question,
            }
        },
        'userInfo': {
            'apiKey': 'ad844aee57c74406a68a891d3c19e7fb',
            'userId': '408963'
        }
    }
    data = json.dumps(data)
    r = requests.post(url, data=data)
    data = json.loads(r.text)
    return data['results'][0]['values']['text']
```

把Ask函数封装进入Robot类中，使得我们可以按如下要求使用：

```
robot = Robot()

while True:

    print('提问：')
    text = input()
    print('机器人回复：')
    print(robot.ask(text))
```

提问:如何让两个机器人互相吵架?

# GET VS POST

	Get	POST
后退按钮/刷新	无副作用	会被重新提交
缓存	可以被缓存	不能缓存
浏览器历史	参数会被保存在浏览器历史中	参数不会被保存
安全性	参数暴露在URL中, 提交密码等重要信息一定不能用GET	POST相比GET更加的安全
对数据长度的限制	有限制	无限制



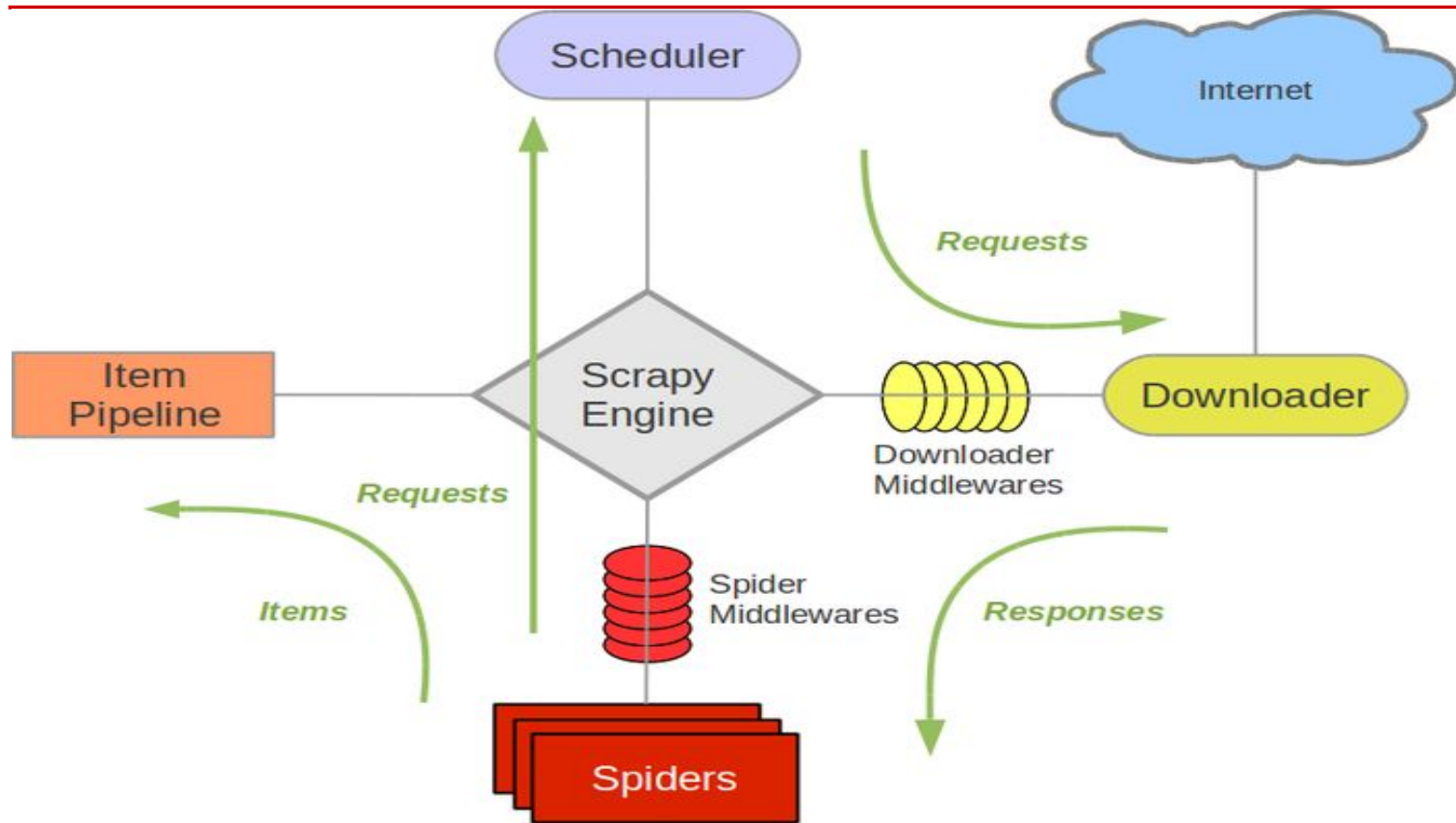
- 之前的请求中，每次请求其实都相当于发起了一个新的请求，他们都是独立的
- 在一些情况下，我们需要保持一个持久的会话，比如登入以后，我们要在登入的状态下，继续和网站交互
- 比如淘宝的时候，我们在不同的选项卡内挑选物品，我们必须保持同一个会话



# Spider介绍



九章算法



Spider爬取的循环类如下：

1 以初始的URL初始化Request, 并设置回调函数。当该request下载完毕并返回时, 将生成response, 并作为参数传给该回调函数

2 spider中初始的request是通过调用 start\_requests() 来获取的。  
start\_requests() 读取 start\_urls 中的URL, 并以 parse 为回调函数生成Request, 因此继承过程中, 我们也可以重写start\_requests()

Spider爬取的循环类如下：

3 在回调函数内分析返回的(网页)内容，可以通过pipeline处理item，也可以加入新的request，下载相应的内容，并调用设置的callback函数

4 在回调函数内，我们可以使用如BeautifulSoup, lxml 或者任何其他解析器)来分析网页内容，并根据分析的数据生成item

5 由spider返回的item将被存到数据库(由某些 Item Pipeline 处理)或使用Feed exports 存入到文件中

<https://bj.lianjia.com/zufang/>



当代名筑, 通透开间, 家具家电齐全

当代名筑 1室0厅 10.56平米 南北

梨园租房 / 低楼层(共12层) / 2006年建板塔结合



近地铁



集中供暖

3600元/月

2018.11.30 更新

0人

看过此房



花园闸小区 2室1厅 5200元

花园闸小区 2室1厅 64.54平米 西

定福庄租房 / 高楼层(共18层) / 2000年建塔楼



近地铁



集中供暖

5200元/月

2018.12.05 更新

2人

看过此房

- scrapy startproject lianjiaspider （开始一个Spider project）
- cd lianjiaspider/
- scrapy genspider example example （开始生成一个Spider）

目录结构中：

这些文件分别是：

- scrapy.cfg: 项目的配置文件
- lianjiaspider/: 该项目的python模块，之后我们将在此加入代码。
- lianjiaspider/items.py: 项目中的item文件。
- lianjiaspider/pipelines.py: 项目中的pipelines文件。
- lianjiaspider/settings.py: 项目的设置文件。
- lianjiaspider/spiders/: 放置spider代码的目录。

```
class LianjiaSpider(scrapy.Spider):  
    name = 'lianjia'  
    allowed_domains = ['lianjia.com']  
    start_urls = ['https://bj.lianjia.com/zufang/']
```

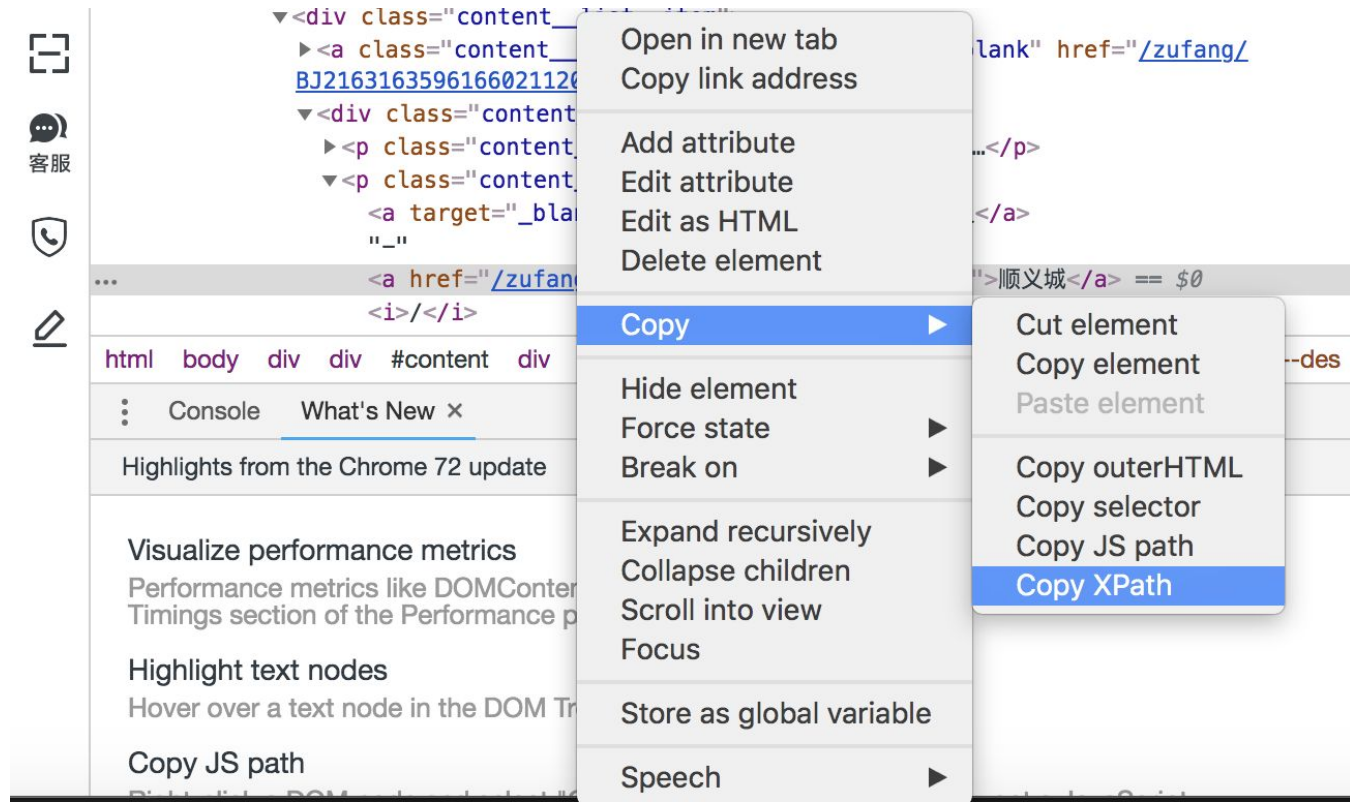
- name 创建spider的唯一名字, 用于命令行启动crawler
- allowed\_domains 允许爬去的domains, 如果这个列表为空, 结果如何?
- start\_urls 初始启动的urls



通过XPath来获取页面中我们需要的信息：

XPath 是一门在 XML 文档中查找信息的语言。XPath 可用来在 XML 文档中对元素和属性进行遍历。

## 使用Chrome浏览器，右键审查元素(inspect)



定义抓取的字段, 在lianjiaspider下的items.py文件下进行定义

```
import scrapy

class HouseItem(scrapy.Item):

    # 标题
    title = scrapy.Field()
    # 小区名
    name = scrapy.Field()
    # 户型
    house_type = scrapy.Field()
    # 面积
    area = scrapy.Field()
    # 所在位置
    position = scrapy.Field()
    # 房子楼层
    floor = scrapy.Field()
    # 房子建筑时间
    build_time = scrapy.Field()
    # 月租金
    monthly_rental = scrapy.Field()
```

对之前的spider文件进行修改，主要修改的是parse函数

- 页面信息提取
- 页面信息解析

尝试运行crawl

- scrapy crawl lianjia

```
anjia.com/zufang/>
{'area': '82.83平米\xa0\xa0',
 'build_time': '2005年建塔楼',
 'floor': '中楼层(共26层)',
 'house_type': '1室1厅\xa0\xa0',
 'monthly_rental': '9500',
 'name': '博雅国际\xa0\xa0',
 'position': '望京租房',
 'title': '博雅国际 1室1厅 9500元'}
```

能否把抓取的结果写入csv中？

使用Pipeline中处理item的方式，每抓到一个item写入到csv的文件中

```
import csv
```

```
创建csv writer
```

## 将item写入csv最中

```
import csv

class PipelineCsv(object):

    def open_spider(self, spider):
        self.f = open("list.csv", "w", newline="")
        self.fw = csv.writer(self.f)
        self.fw.writerow(['title', 'name', 'house_type', 'area', 'floor', \
                           'build_time', 'monthly_rental'])

    def close_spider(self, spider):
        self.f.close()

    def process_item(self, item, spider):
        self.fw.writerow([item['title'], item['name'], item['house_type'], \
                           item['area'], item['floor'], item['build_time'], \
                           item['monthly_rental']])
```

在settings.py文件中添加PipelineCsv

```
ITEM_PIPELINES = {  
    'lianjiaspider.Pipelines.PipelineCsv': 300,  
}
```



思考：

当前只抓取了第一页的租房信息，如果我们想抓取第二页，第三页..... 所有的租房信息我们应该怎么做呢？

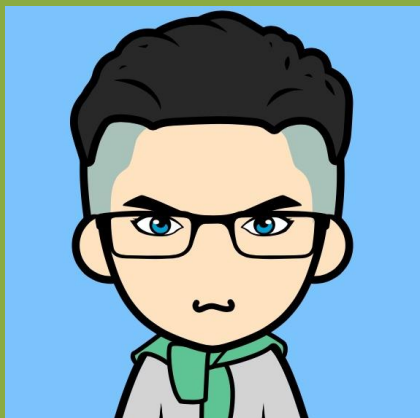


扫描二维码关注微信/微博  
获取最新面试题及权威解答

微信: [ninechapter](#)

微博: <http://www.weibo.com/ninechapter>

官网: [www.jiuzhang.com](http://www.jiuzhang.com)



谢谢大家