**爬取牛客网里的求职信息**

1. **爬虫思路分析**

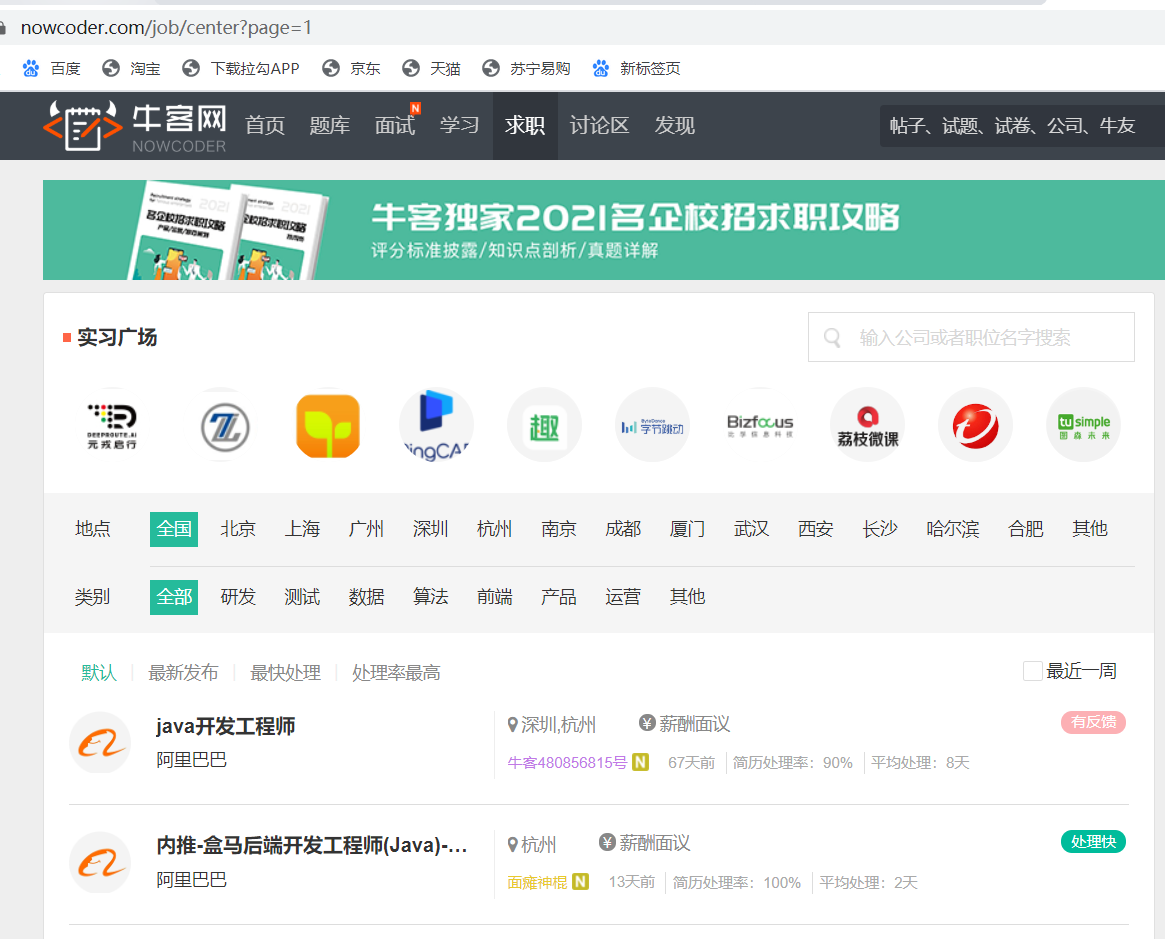
**1、网络数据采集所用到的技术**

所用到的技术有Scrapy爬虫框架,静态网页爬取技术以及反反爬技术

**2、爬虫实现思路**

1. 判断网页是静态网页还是动态网页

打开网页进入求职信息页面



判断网页是静态网页还是动态网页查看网页源代码



发现网页源代码有每个工作的相关信息，故为静态页面

1. 选取需要爬取的数据



确定爬取的数据有：工作名字 job\_name，工作地点job\_are，工作薪资job\_salary，公司名字company，发布时间release\_time，简历处理率rate，简历平均处理时间average，工作链接job\_link

查看网页结构:



发现每项工作的信息都在ul标签下的li标签里，所以可以先爬取li标签，然后从li标签爬取每项信息。先用xpath定位到li标签。

工作名字 job\_name：



因为是先定义li标签的，所以可以直接以li标签为根节点往下寻找。这里我用的是xpath方法。工作名字是在a标签的文本内容，所以xpath语法后面需要加上text()。

工作地点job\_are：



工作地点是在span标签里的文本内容。

工作薪资job\_salary：



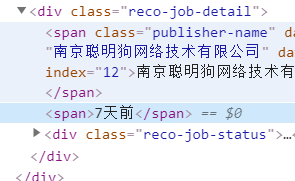
工作地点是在span标签里的文本内容。

公司名字company：



公司名字是在a标签的文本内容。

发布时间release\_time：



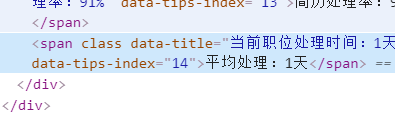
发布时间是在span标签里的文本内容。

简历处理率rate：



简历处理率是在span标签里的文本内容。

简历平均处理时间average：



简历平均处理时间是在span标签里的文本内容。

工作链接job\_link：



工作详细链接在a标签的href里，但发现该链接不是全部链接，单独拿出来打开不了。先手动打开几个详细链接看看

<https://www.nowcoder.com/job/5944?jobIds=23411&ncsr=>

<https://www.nowcoder.com/job/8752?jobIds=12781&ncsr=>

<https://www.nowcoder.com/job/8304?jobIds=21878&ncsr=>

发现工作详细链接和我们爬取的a标签里的href多了主页的链接https://www.nowcoder.com 所以需要将主页链接和爬取a标签里的href拼接起来。因此先爬取a标签里的href，再将其拼接起来。

1. 查看多页爬取需要的技术

查看前几页的网址，手动翻页进行查看：以下是前4页的网址

<https://www.nowcoder.com/job/center?page=1>

<https://www.nowcoder.com/job/center?page=2>

<https://www.nowcoder.com/job/center?page=3>

[https://www.nowcoder.com/job/center?page=4](https://www.nowcoder.com/job/center?page=3)

发下每页的网址只有page后面的数字发生改变，故爬取多页只需要改变数字即可。

爬取的数据过多可能会被网址禁止访问，所以需要反反爬技术即随机ip,随机userAgent。

随机ip在西刺代理网找到<https://www.xicidaili.com/>，还需要对ip进行验证可不可用。

**3、爬虫实现步骤**

1. 打开控制台创建名称为niuke的scrapy项目

输入scrapy startpoject niuke创建名称为niuke的scrapy项目,

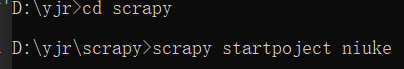
进入niuke后输入scrapy genspider job nowcoder.com 创建一个job的脚本文件。

1. 打开PyCharm,打开创建好的niuke项目。
2. 打开itmes.py，创建爬取内容字段
3. 打开job.py,创建爬取内容
4. 打开pipelines.py，写入代码对爬取的内容进行保存在job.csv中
5. 打开settings.py，创建项目所需要的的参数
6. 创建main.py,以代表控制台运行程序
7. 爬取几页试试代码可不可行并进行修改。
8. 进行反反爬，创建iptest.py,对在西刺网找的ip进行验证。多找几个。
9. 再次打开settings.py，用（8）里找到的ip构建一个ip池。
10. 打开middlewares.py。构造使用代理ip进行爬取网页。
11. 运行程序爬取所有信息。
12. **爬虫代码实现**

**1、第一种爬虫技术Scrapy爬虫框架、代码及注释**

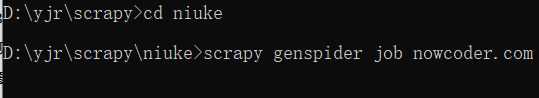
创建scrapy项目：

打开控制台输入scrapy startproject niuke创建名称为niuke的scrapy项目：



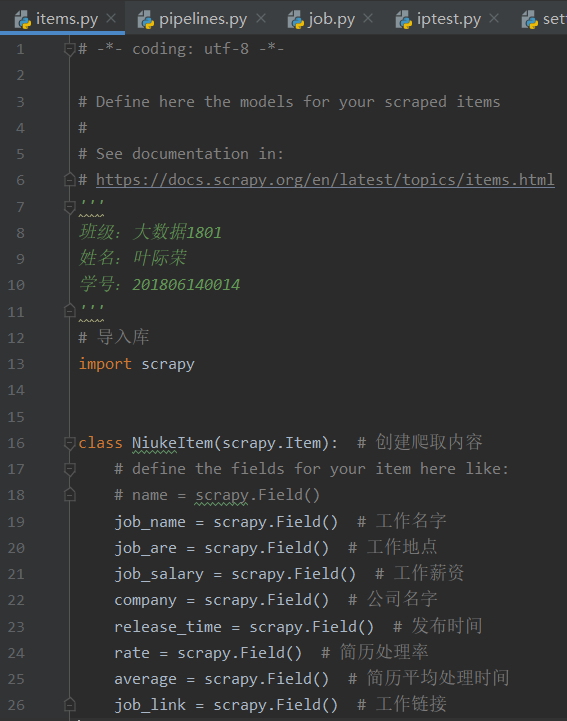


进入niuke后输入scrapy genspider job nowcoder.com 创建一个job的脚本文件。



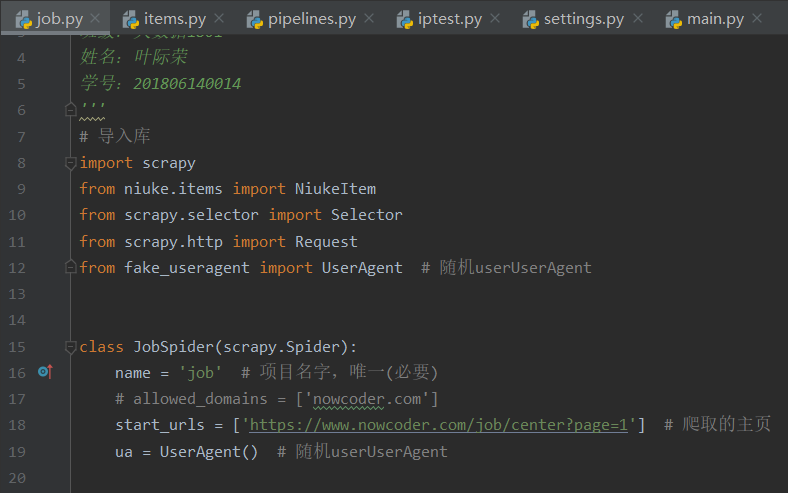


打开PyCharm,打开创建好的niuke项目，打开itmes.py，创建爬取内容字段：

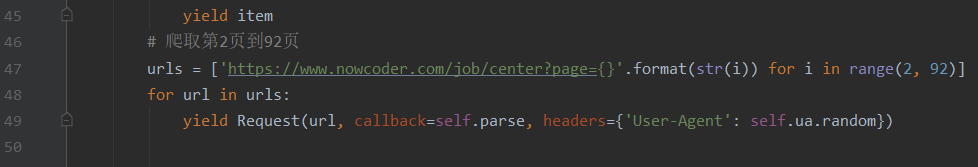


**2、第二种爬虫技术静态网页爬取技术、代码及注释**

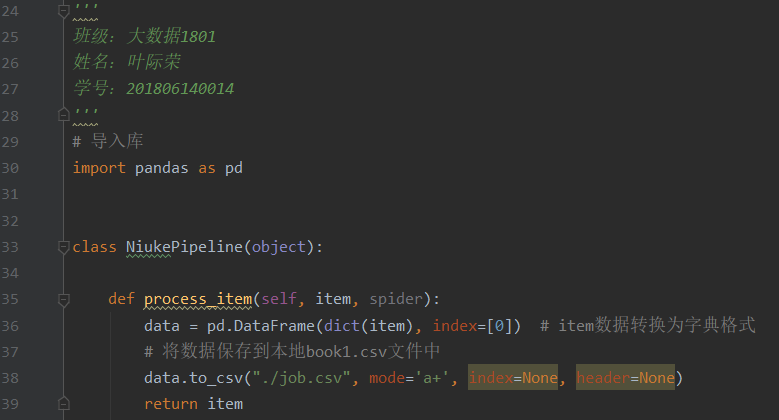
打开job.py，写入爬取的主页及用xpath爬取所有相关信息



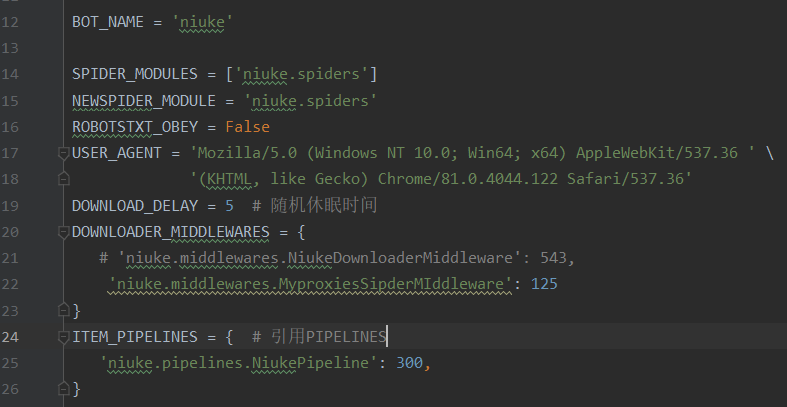




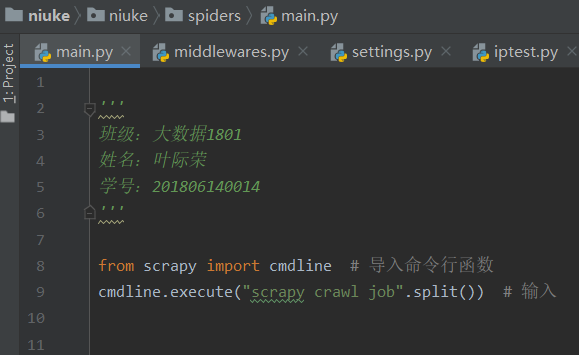
打开pipelines.py，写入代码对爬取的内容进行保存在job.csv中



打开settings.py，创建项目所需要的的参数

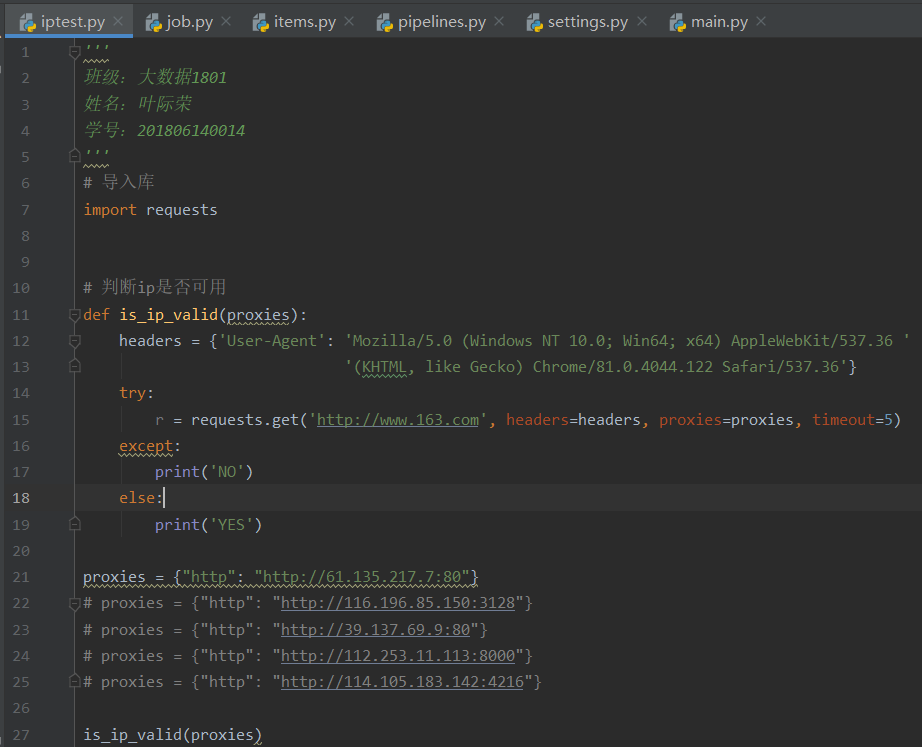


创建main.py,以代表控制台运行程序

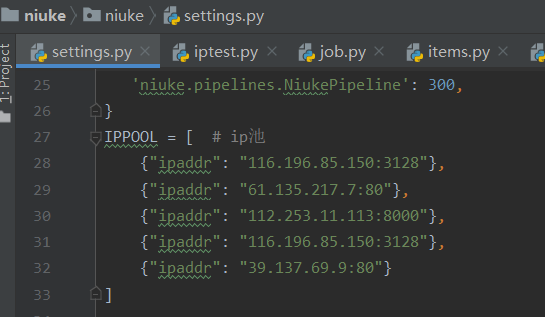


**3、第三种爬虫技术反反爬技术、代码及注释**

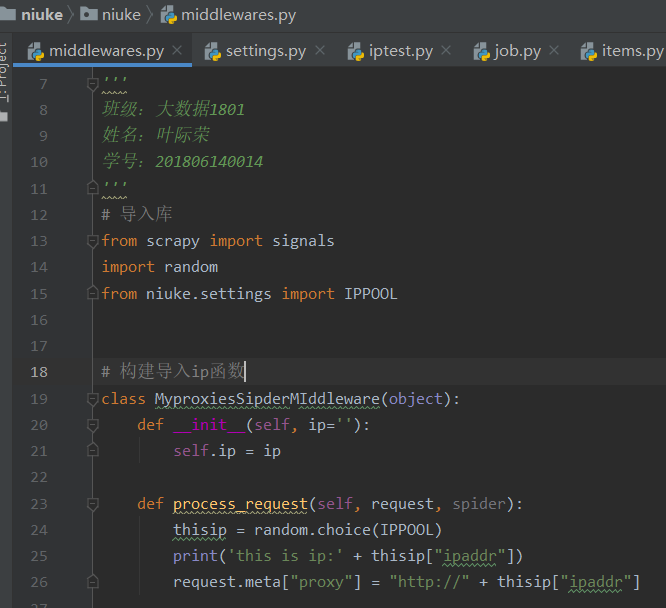
创建iptest.py,对在西刺网找的ip进行验证



再次打开settings.py，用找到的ip构建一个ip池



打开middlewares.py。构造程序运行时使用代理ip进行爬取网页

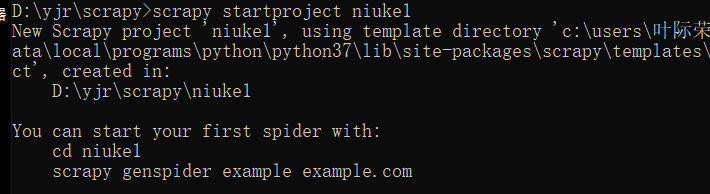


**三、爬虫运行结果及简要说明**

**1、第一种爬虫技术Scrapy爬虫框架运行结果及简要说明**

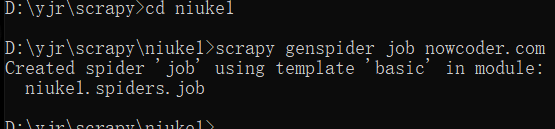
输入scrapy startpoject niuke创建名称为niuke的scrapy项目

因为我已经创建了niuke。这里用niuke1代替

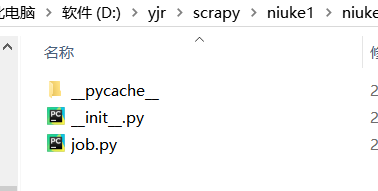


控制台出现这样就代表已经创建好了niuke1

进入niuke后输入scrapy genspider job nowcoder.com 创建一个job的脚本文件。



打开目录就会发现已经创建好了项目

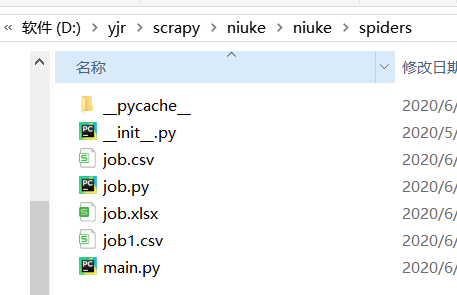


1. **第二种爬虫技术静态网页爬取技术运行结果及简要说明**

爬取的信息大概如下：



打开其项目目录下有保存的csv文件

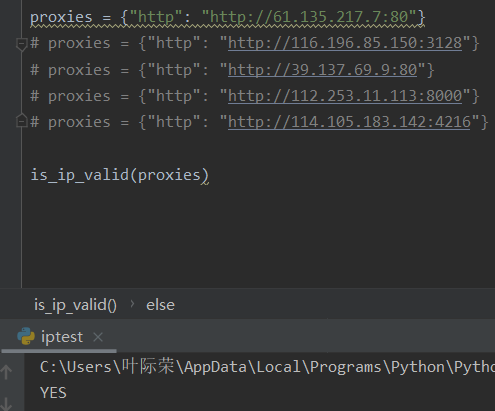


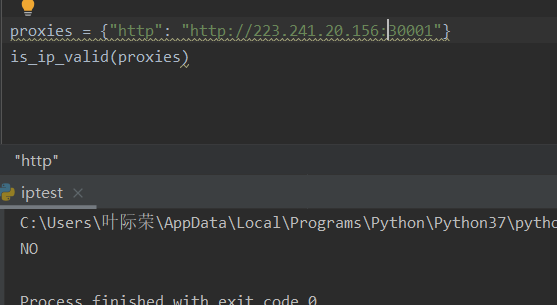
csv文件的部分内容



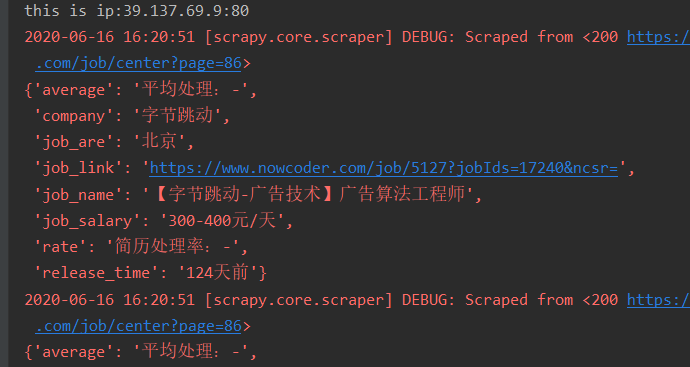
**3、第三种爬虫技术运行结果及简要说明**

对ip测试时运行为yes则词ip可用，no则为不可用。





爬取所有页数的信息是，每页都为不一样的ip



共爬取2031条信息，暂未对此数据进行处理