使用requests获取 “数据分析师”职位信息

：

工具准备：

1.Anaconda3，[下载链接](https://www.anaconda.com/download/)，[清华的镜像](https://mirrors.tuna.tsinghua.edu.cn/anaconda/archive/)，选择适合自己平台的版本，我使用的是macos，python3。使用Anaconda的优势在于它集成了很多科学计算包，包括pandas，numpy，matplotlib等，包括我们本次使用的requests包和jupyter notebook。直接安装anaconda能够避免很多单独安装这些包时发生的莫名其妙的问题。

2.chrome或者firefox。我们使用浏览器的开发者模式分析网络请求。对于任何一种爬虫，对于请求的分析必不可少。我最常用的是Chrome浏览器。

本次爬虫的流程：

1.分析url。分析目标页面，找到数据的url接口。经过考察，智联招聘和拉勾网的职位信息都是以json格式返回的。Json是一种使用键值对保存数据的，类似于python中的dict格式，处理起来很方便。这样的数据格式对于爬虫也是十分“友好”的，只要找到了数据的url，通过模拟网络请求很容易一次性拿到所有数据。

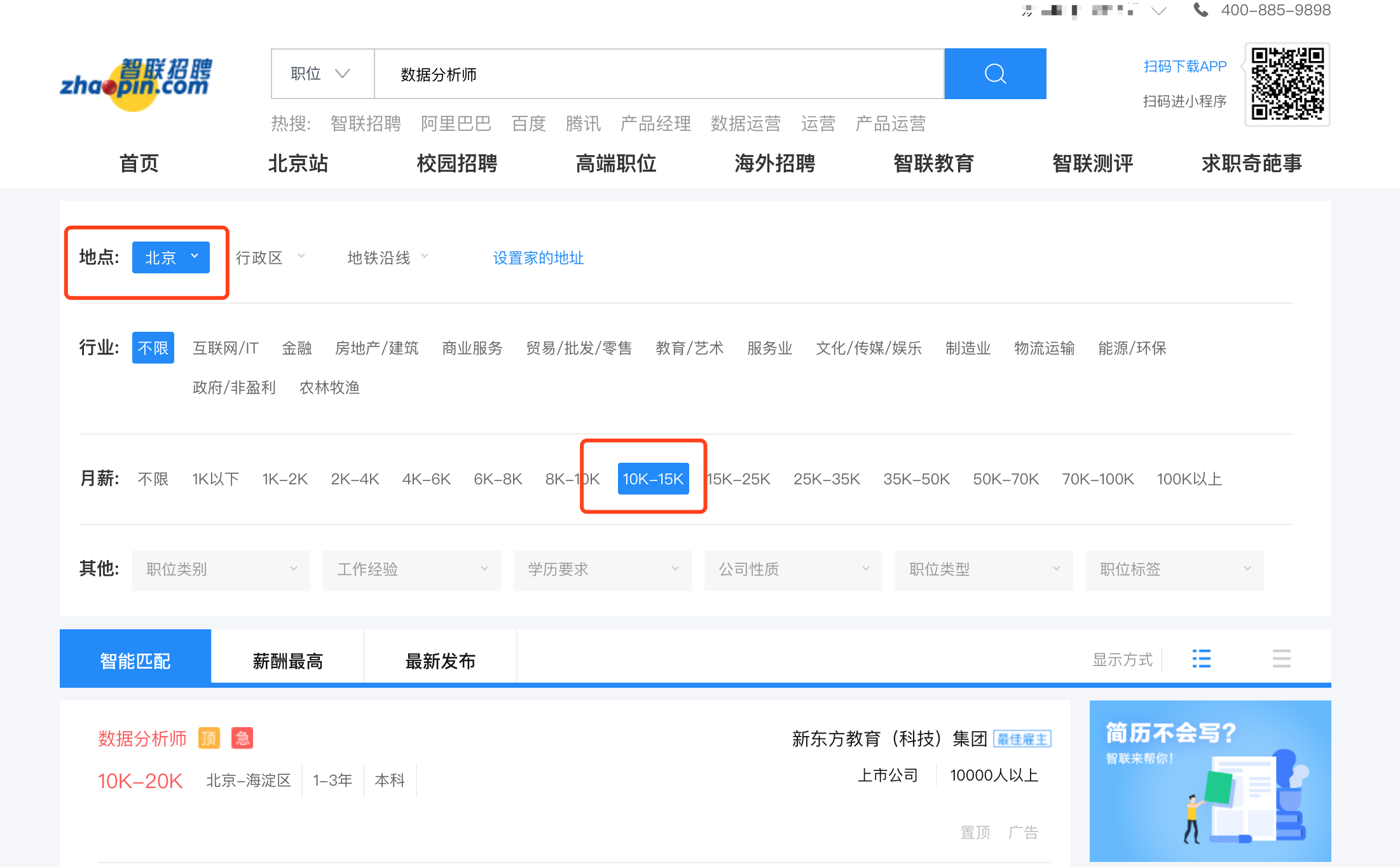
**智联招聘数据分析师职位**

**请求分析：**



1）登录智联招聘，默认地区是“北京市”，在搜索框中输入“数据分析师”，电子搜索

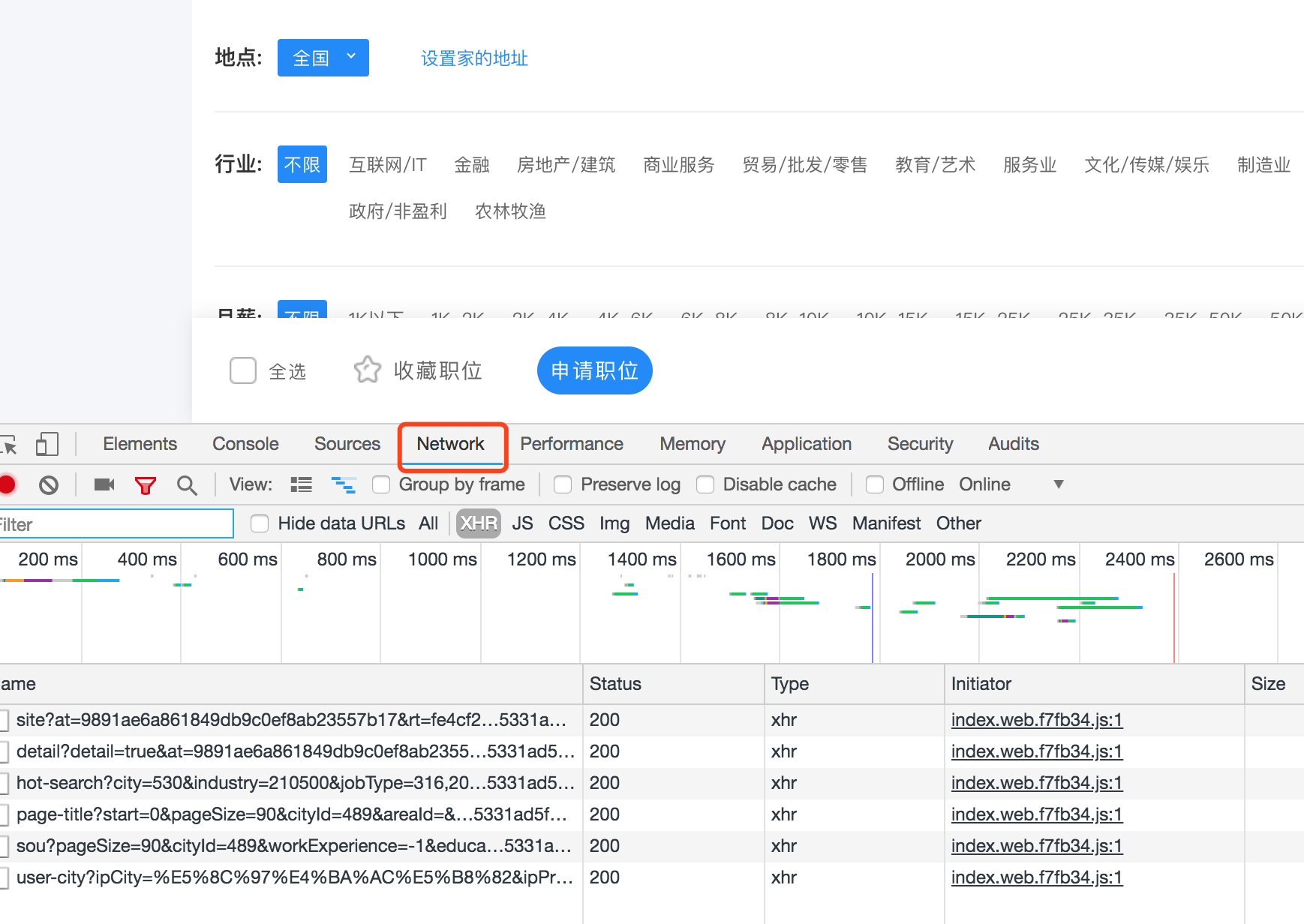
2）在搜索结果中发现，默认展示的是北京地区，10k-15k的数据分析师职位，排在第一位的是一条新东方教育的招聘信息(但其实是广告位)。我们将地点改为“全国”，月薪改为“不限”。



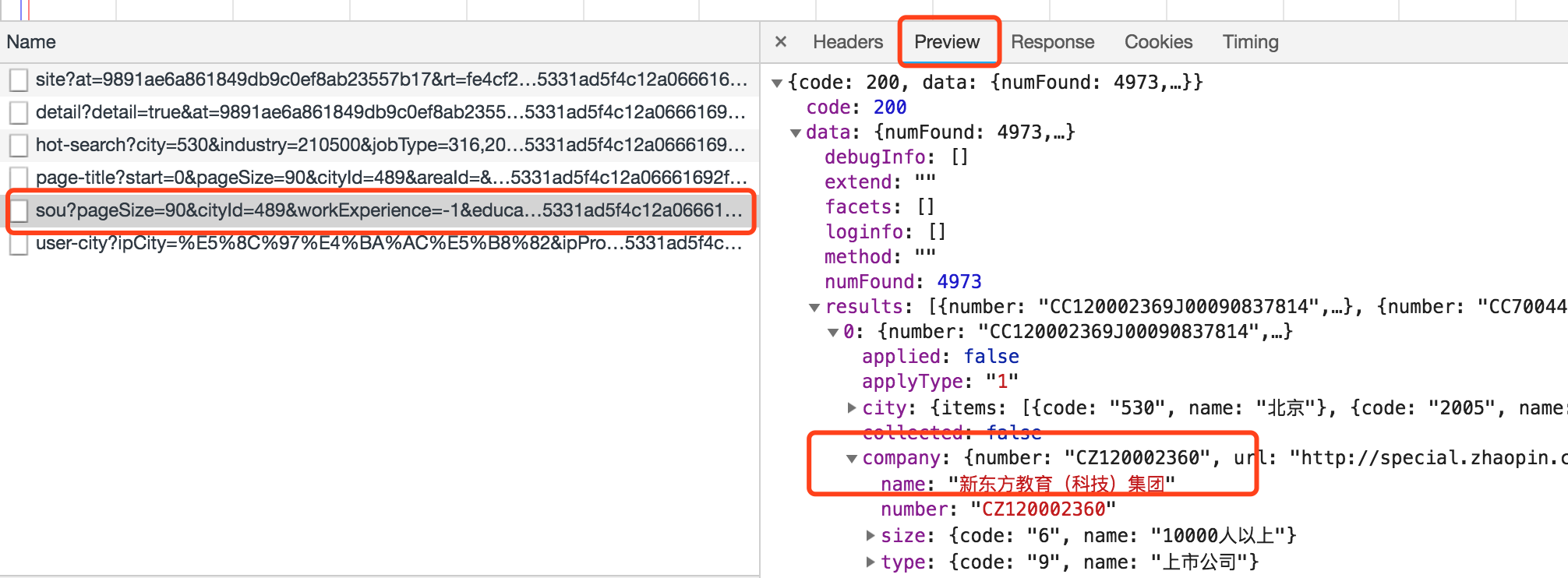




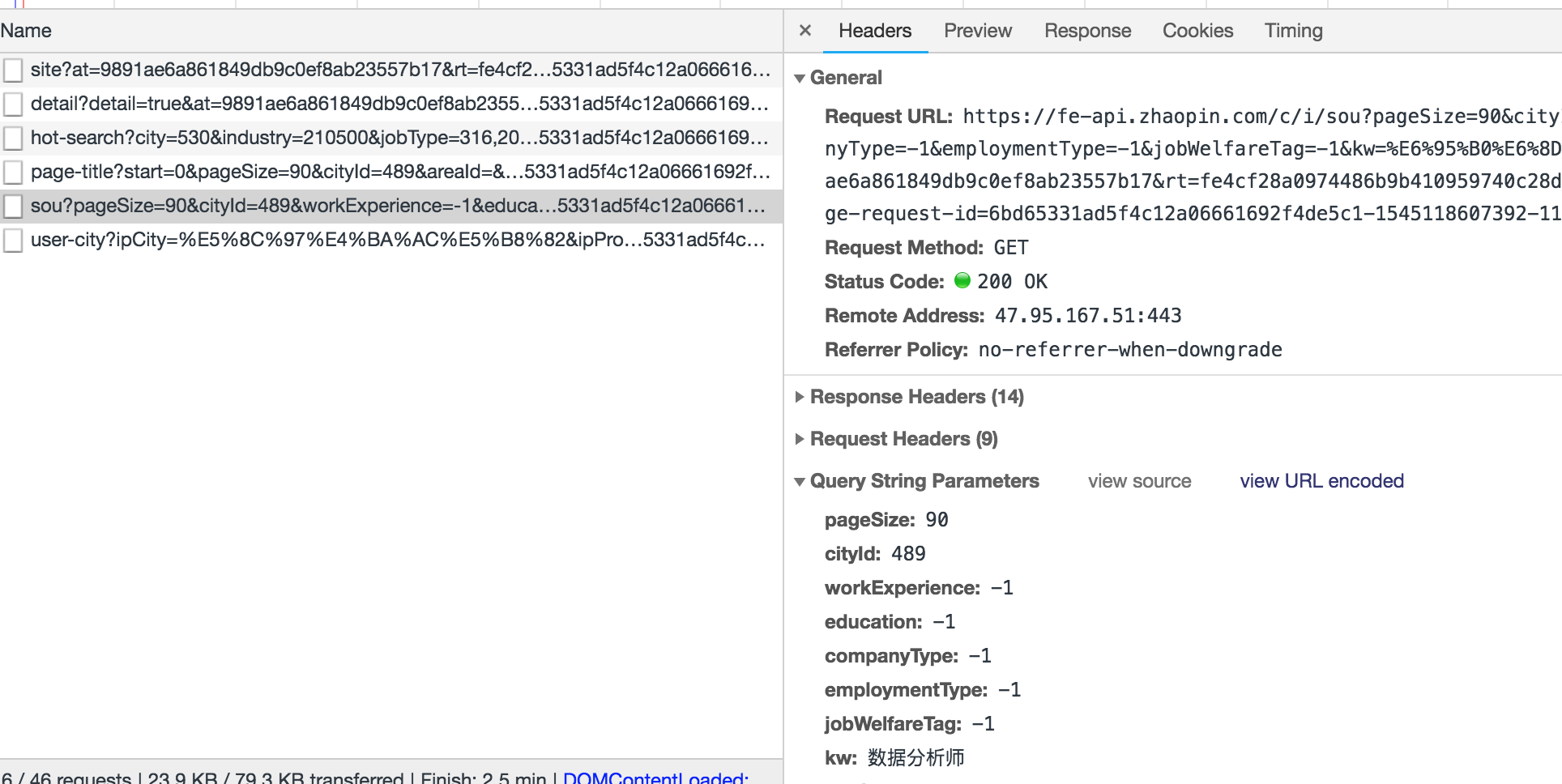
3）右键选择检查元素，打开开发者模式，切换到network选项卡，并重新加载网络请求（刷新页面即可）。在最下方的区域看到了很多请求，这些都是和当前页面相关的请求。倒数第二个，”sou?”开头的就“藏着”我们需要的职位信息，下面我们详细分析。

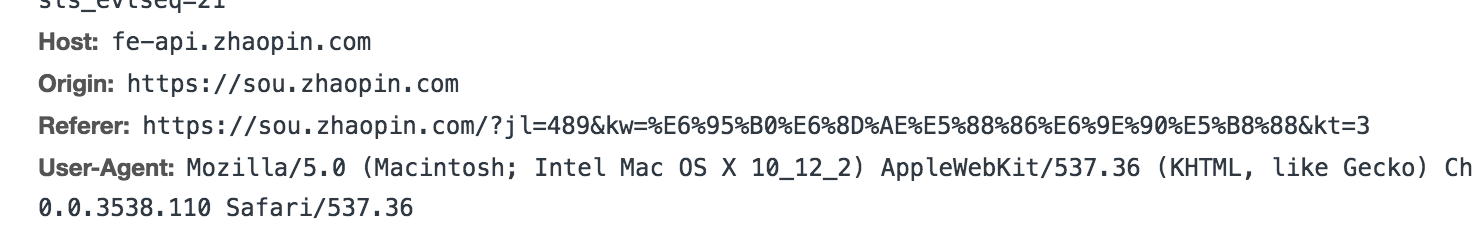


4）点击倒数第二个请求（sou开头的那个url），查看preview选项卡和header选项卡。Preview中是当前这个请求response的“预览”模式，这里的response是json格式的，我们依次展开“data”,”results”,”0”,就可以看到与页面对应的第一条的信息：新东方教育集团的招聘信息。职位名称，薪资，经验要求，公司资质等信息也可以找到。因此，我们确认这一请求就是我们需要的网络请求。



5）切换到headers选项卡，我们可以看到该请求的相关细节。





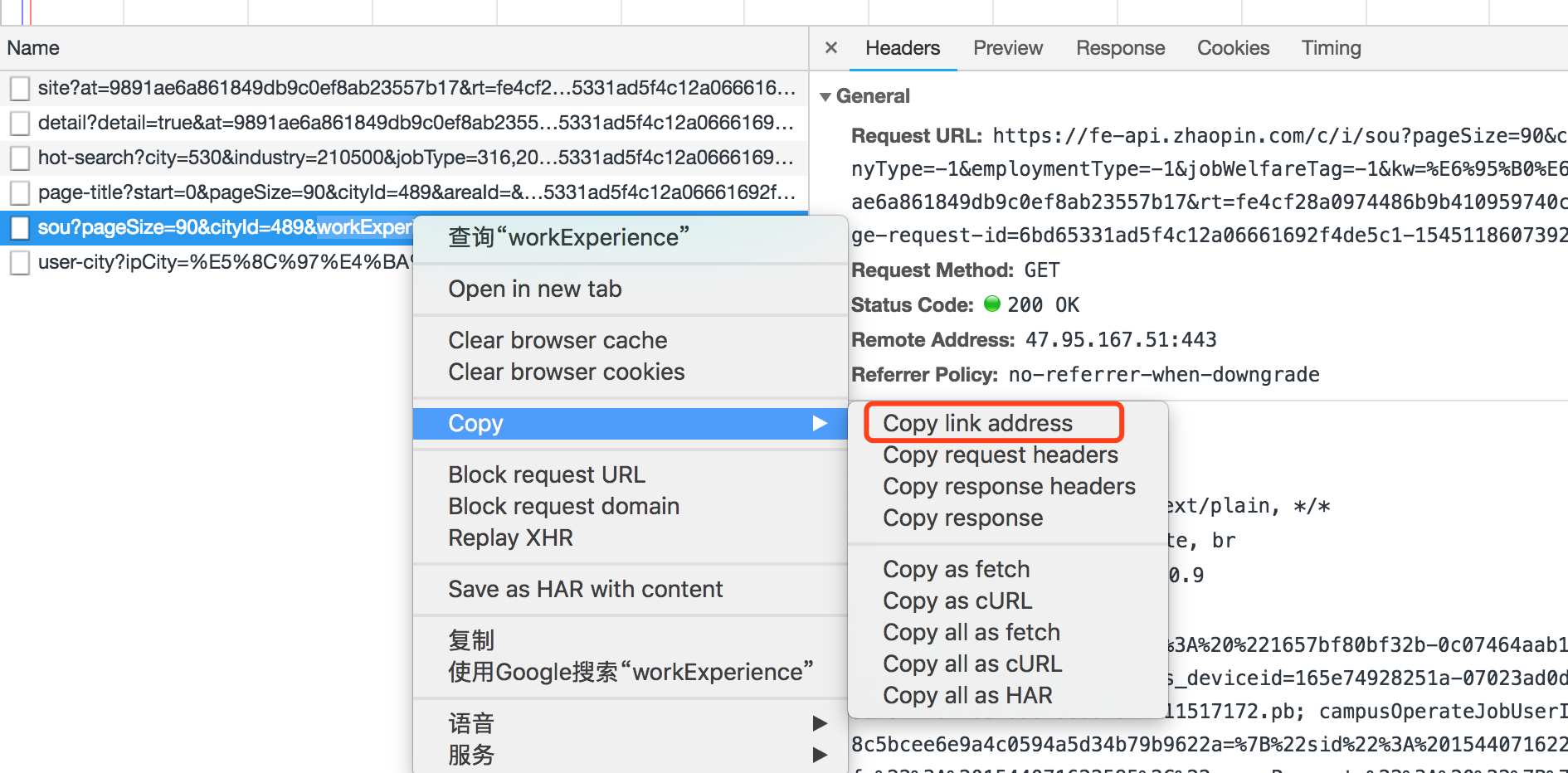
其中常用的比较重要的信息有以下几项，在我们的代码里都要用到

General 下的Request Method: GET

Request Headers下的 Referer，user-agent和Cookie，由于cookie太长我没有截图，cookie通常用来验证身份，加上它通常可以避免一些不必要的麻烦。但可能失效，如果写死了，过一段时间发现数据获取不到，可能是需要换新的cookie。

**代码实战：**

1）在请求分析找到目标url之后，可以右键选择“copy link address”得到目标url地址。注意这里只是第一页的职位信息，如果要找到后面页的职位信息，需要观察各页的url之间有何关联。

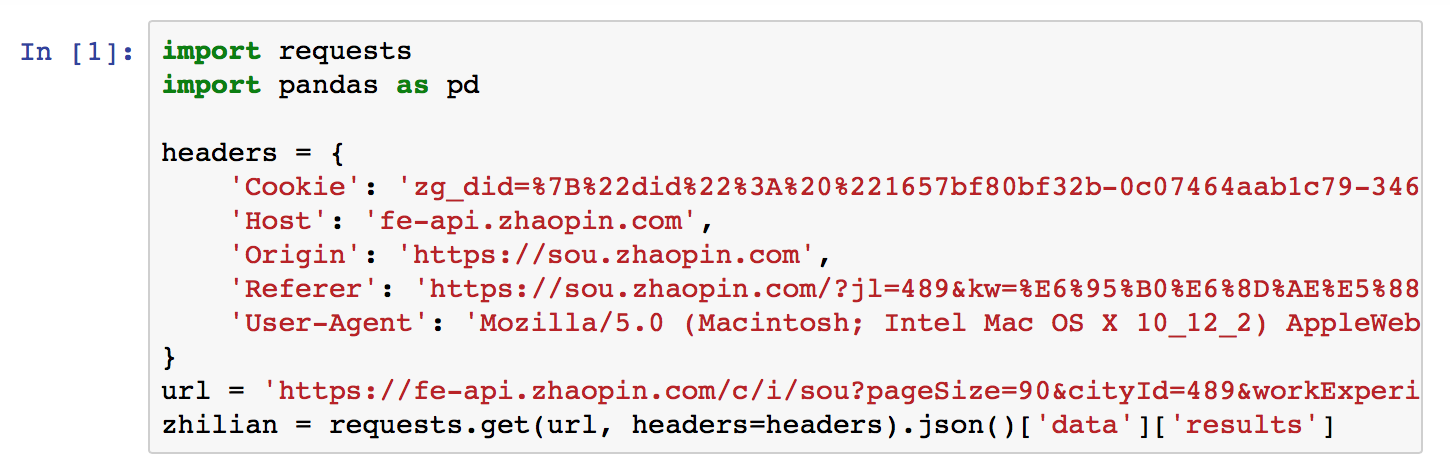


url = ‘

https://fe-api.zhaopin.com/c/i/sou?pageSize=90&cityId=489&workExperience=-1&education=-1&companyType=-1&employmentType=-1&jobWelfareTag=-1&kw=%E6%95%B0%E6%8D%AE%E5%88%86%E6%9E%90%E5%B8%88&kt=3&at=9891ae6a861849db9c0ef8ab23557b17&rt=fe4cf28a0974486b9b410959740c28d3&\_v=0.55651002&userCode=1006555709&x-zp-page-request-id=6bd65331ad5f4c12a06661692f4de5c1-1545118607392-116050’

2）使用代码模拟url请求

导入requests包，调用requests.get方法模拟get请求，并将刚刚分析的headers参数同时传入。这样返回的结果就是刚刚preview的结果，但注意需要加一个json()方法，才能让结果变得更加“可读”。我们只关心data下的results，所以在代码里有所体现，具体请看：



上面代码中，zhilian，就是包含职位信息的一个“字典”，但这个字典里有一些我们不太需要甚至不知道含义的信息，因此需要进行一定的“筛选”。由于我打算使用pandas来存储数据，所以用job\_detail字典构造了一个DataFrame对象job\_df，用打印了前5行结果和job\_df的长度。调用job\_df的to\_csv方法就可以将数据存储到csv文件中。



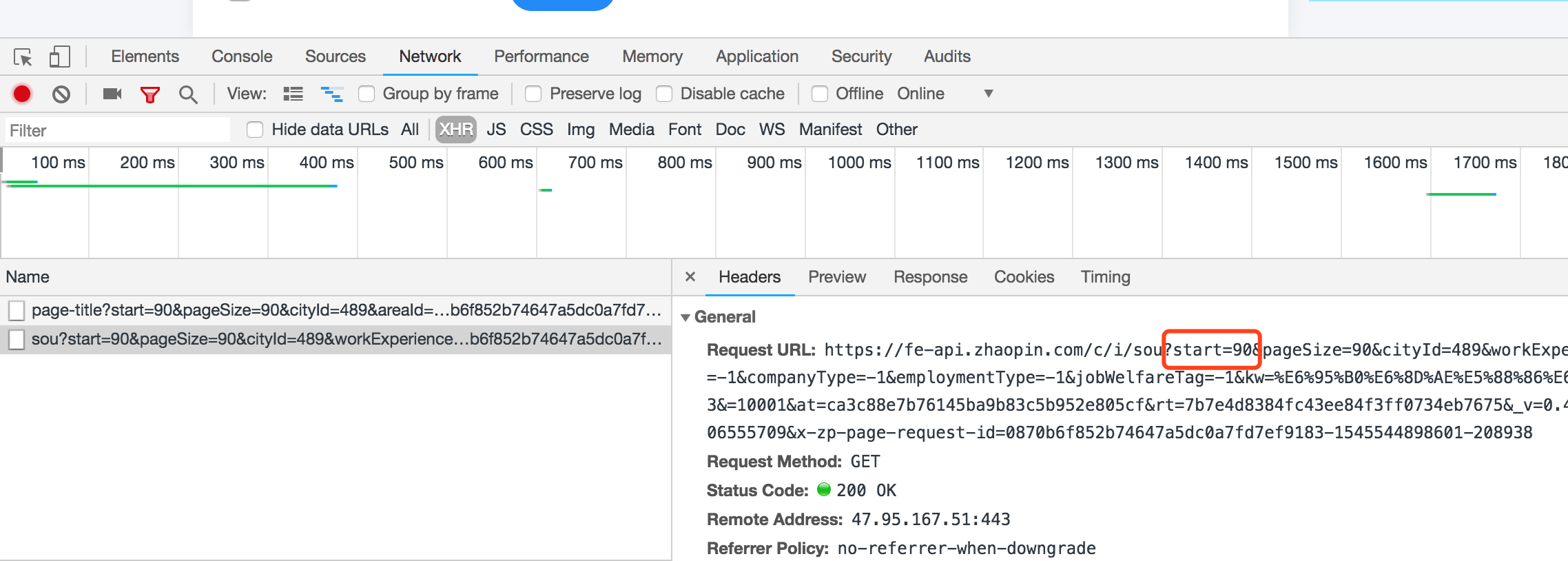




这样我们就将第一页的职位信息共90条完整地获取到并且存到了文件中。

完整代码和分析：[查看](https://github.com/chenghengchao/useful_code/tree/master/zhilian)

**翻页分析**



再点击下一页之前，将当前的请求清空，以免造成干扰。然后点击第二页，容易看到，同样是以sou?开头的请求包含了目标数据。此时的url比之前第一页的多了一个参数：start=90，而一页的信息刚好是90条，因此我们可以猜测后面页只要修改这个参数即可。可以继续分析第三页第四页验证确实是这样。并且第一页可以设置start=0。如此我们就可以进行翻页的批量抓取了。那么如何计算总页数呢？注意到有一个数据是numFound：6686。这其实是职位的总条数，根据这个我们可以计算出总页数是75页。



**TODO：**

**拉勾网数据分析师职位请求分析**