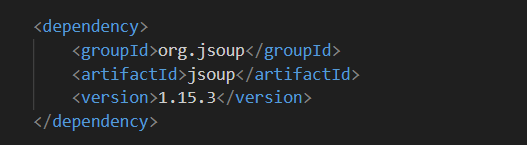
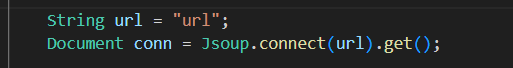
本篇文章用于记录自己的爬虫使用经验：

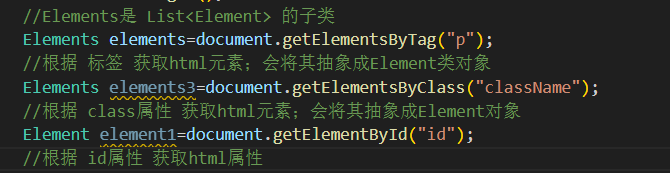


通过在maven中导入jsoup依赖，可以进行简单的爬虫操作

**根据url发送http请求获取对应的html页面，封装成Document类对象**：



**常用的提取元素的方式**：



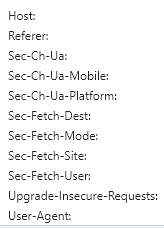
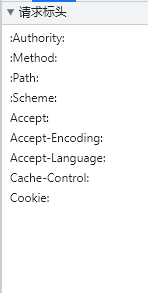
**提取元素中的内容**：一般用于过滤信息

**但是实际情况是，许多网站都会有点“反爬虫”措施，或者需要你登录才能看到想爬取的信息，这种情况该怎么办呢**？

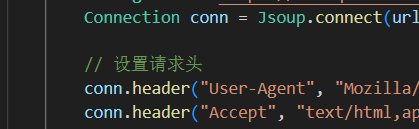
为了解决这个问题，我们需要知道上面的爬虫发出的http请求和我们平常访问网站时有什么不同：

首先，你得能正常访问需要爬取的网站，之后利用f12查看正常访问时的http请求

之后找到左边这样的http请求头

但是呢，我们上面发送的爬虫请求，是没有带请求头的http请求

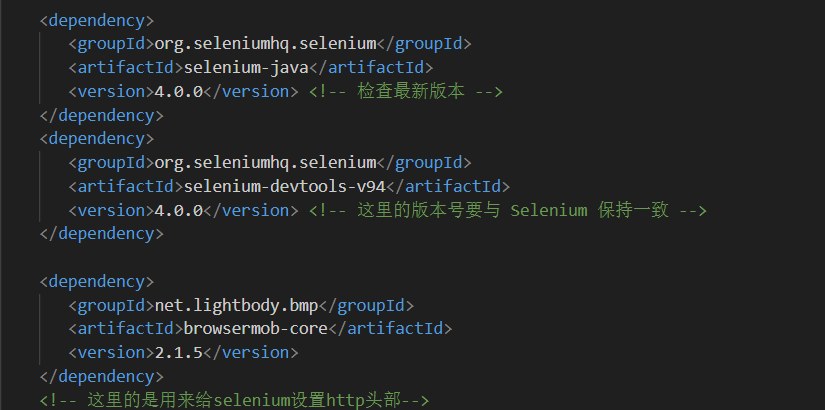
**因此我们只需要在爬虫请求中添加这些请求头，就能骗过目标服务器，具体操作**为：

在conn对象调用get方法前，设置请求头即可

此外，通过请求头也可以设置cookie，JWT等信息，因此也能通过这种方式完成**登录**。

此外，如果**遇上前后端分离的项目，即数据是通过ajax请求获取的情况，以上方法就不适用了，怎么解决呢**？🡪需要我们用上selenium（甚至可以解决人机验证问题）

**事前准备**：



首先，需要通过maven导入相关依赖

其次，需要有一个游览器，以及该游览器版本对应的WebDriver

**简单操作过程**：



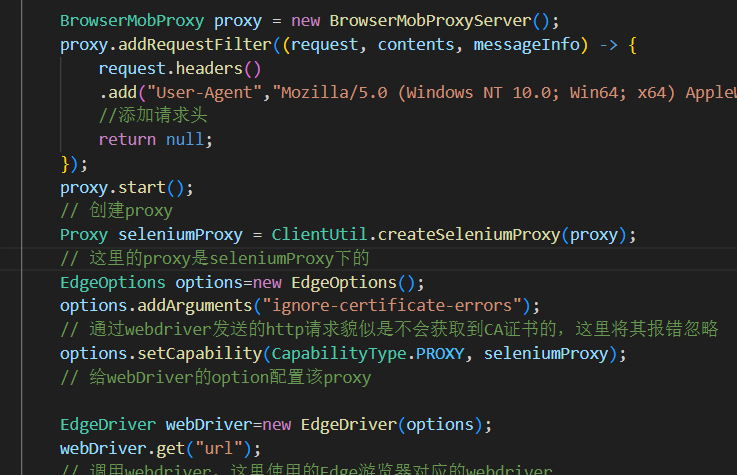
具体来说，上述代码会操控WebDriver，WebDriver再操控游览器，最终达到“**模拟一个用户正常访问页面**”的效果

此外，**WebDriver可以通过JS代码，达成“点击某个按钮进行人机验证”，“提交表单”等**Jsoup做不到的操作，但暂时没有爬取需要人机验证的网站的需求，了解不深，这里不多说明

如果网站仍然需要检验http请求头的情况，怎么办呢？**selenium添加请求头的过程**：

<https://stackoverflow.com/questions/60836031/how-to-set-headers-to-selenium-webdriver-each-requests-can-be-use-different-hea>

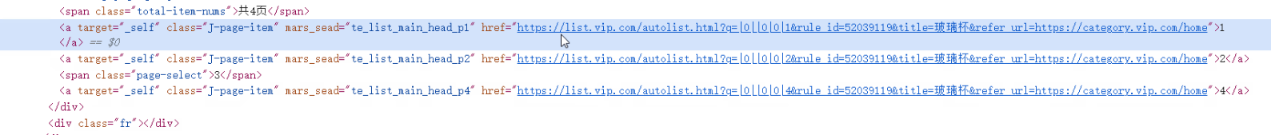
由于网上给selenium请求添加请求头的教程很少，找了半天都只能找到英文教程，凑合着使用



这是添加http请求头的过程



值得注意的是，这部分的构造函数需要将options作为参数



**能通过json请求（一般是ajax请求）直接获取到数据🡪就直接访问这些url获取**【而不是访问整个页面】

**通过f12中的网络监控，找所有请求🡪可以通过“真正数据”+ （ctrl+f） 进行筛选**🡪最终找到需要的ajax请求；【能否通过这种方式，避免一些反爬虫措施？】

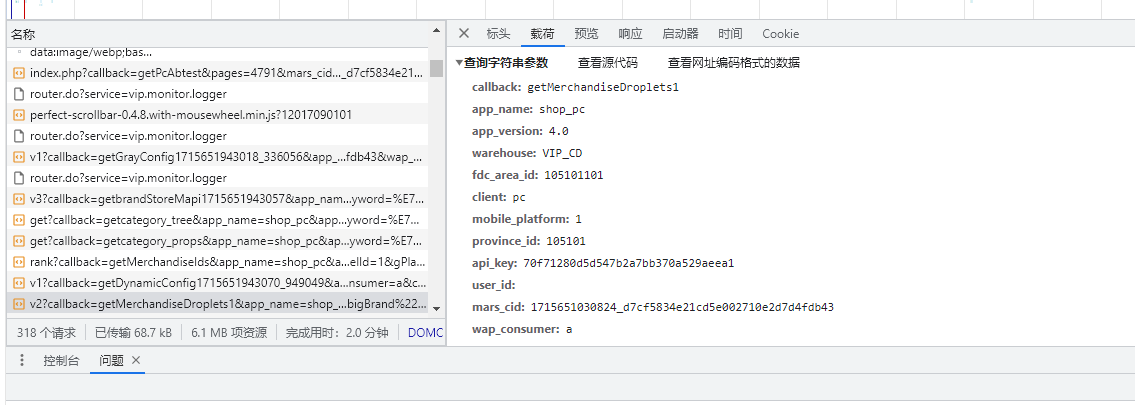
之前访问今日头条时，用静态jsoup爬取会导致一堆乱码🡪还没试过用Jsoup直接访问json请求【会有种情况，就是返回的并不全是“json类型数据”🡪需要将其变成String类型数据，截取一部分后，再将其变为json数据🡪之后再解析（至少json字符串是完全可以用来序列化成Java对象的）】

**但是，如果页面是通过ajax请求获取数据的，直接访问ajax请求可以提高效率**：

下面以唯品会搜索“玻璃杯”获取的数据为爬虫例子

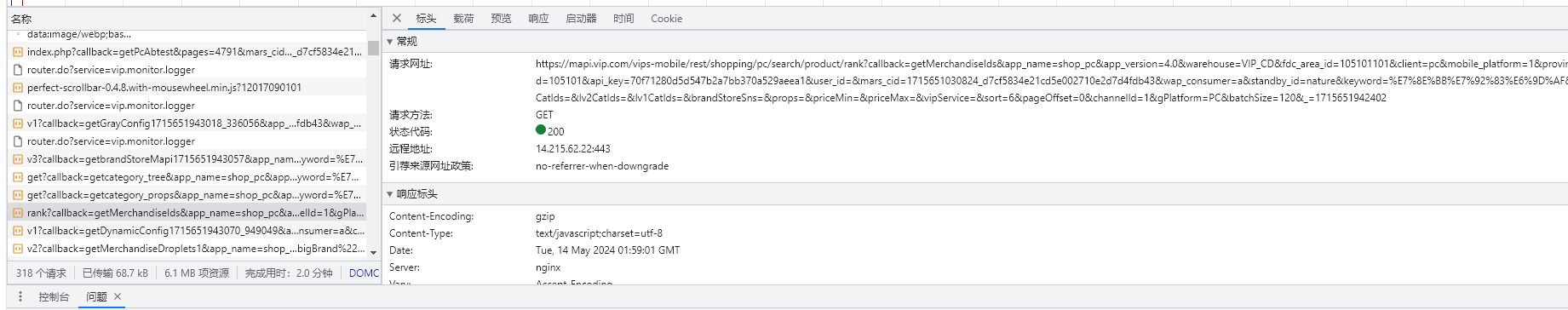


这里能看见响应体



这里能看见payload（载荷）🡪get请求的url上的参数【如果是用消息体携带数据，则没有载荷吧】

可以从图中看见，有productIds产品id作为参数（这部分数据又要靠其他ajax请求获取）



对应的是这个请求🡪获取pid的请求