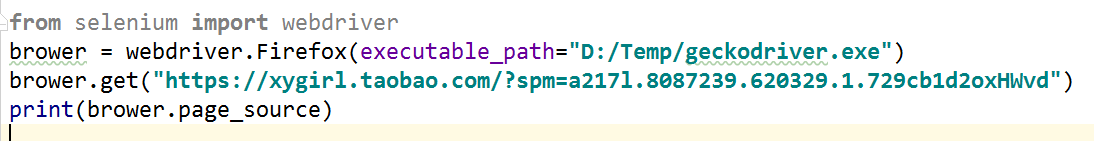
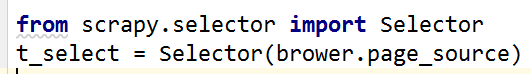
1. selenium是用程序模拟人来操作的一个‘工具’,一般在python中用到时如下：

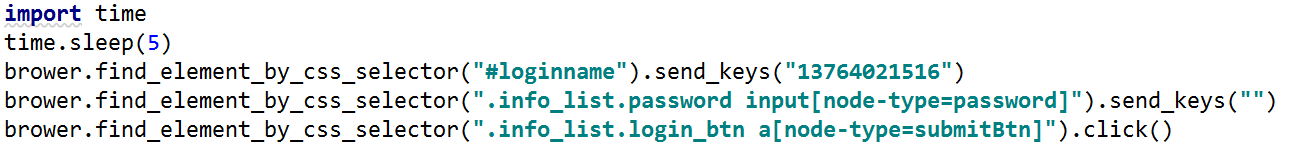
其中这里的Firefox是浏览器，其内是selenium官网(http://selenium-python.readthedocs.io/installation.html)上提供的相应的exe文件下载到本地的所在路径，而brower.page\_source相当于response.text

1. selenium是用纯python写的，而scrapy底层使用的C语言，所以在对页面元素进行提取并加载到item中用scrapy的Selector速度比较快，但当我们要利用selenium对提取出来的元素进行点击时，还是得用selenium.

下面是利用scrapy的selector的代码:

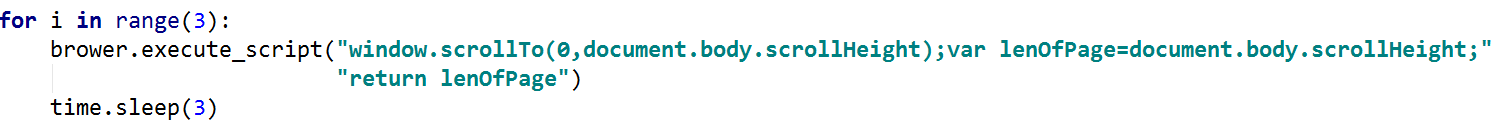


1. selenium通过自己的选择器选择出来元素后可以进行输入send\_keys()和点击click()操作,



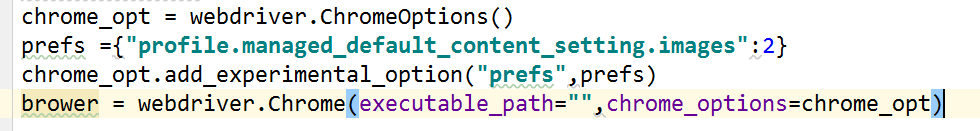
(通过selenium进行用户名，密码输入，并点击登录按钮)

还可以通过运行javascript代码来进行一些操作，如：滚动鼠标滚轮(拉动下拉条)

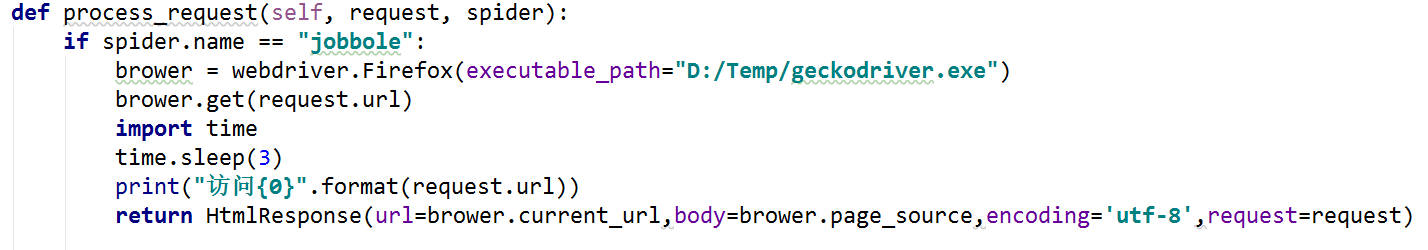


(通过运行javascript代码拉动下拉条3次)

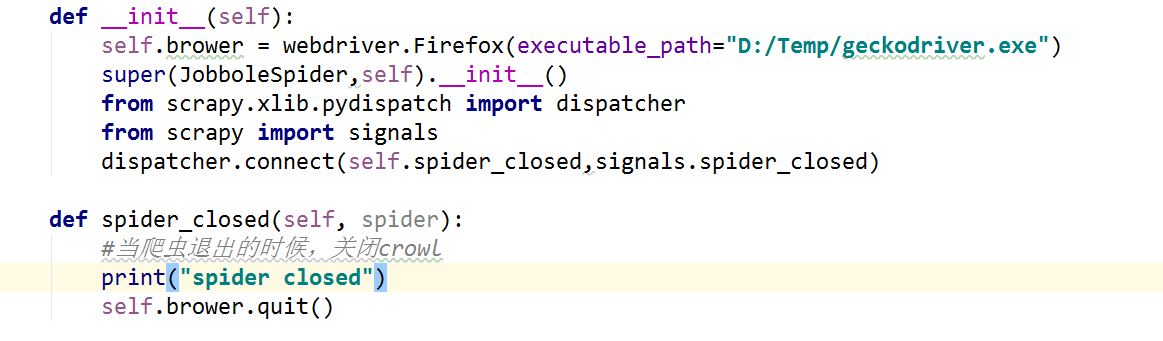
1. selenium操作Chrome设置不加载图片



1. 要将selenium和scrapy结合在一起，可以通过设置middlewares来操作，注意到middlewares一般是默认对response进行处理后再连接到download的，要让其直接返回，则可以在重写的process\_request中return一个Htmlresponse：



6.为了使selenium不会单独为每一个request请求都单独打开一个网页，我们可以把brower写成一个属于类的属性，又要能够自行关闭brower(执行brower.quit())，可以把brower放在spider类中(放在middlewares中不会执行)



这里为网页退出使用的信号和dispatcher类,当spider关闭时就会执行spider\_closed函数