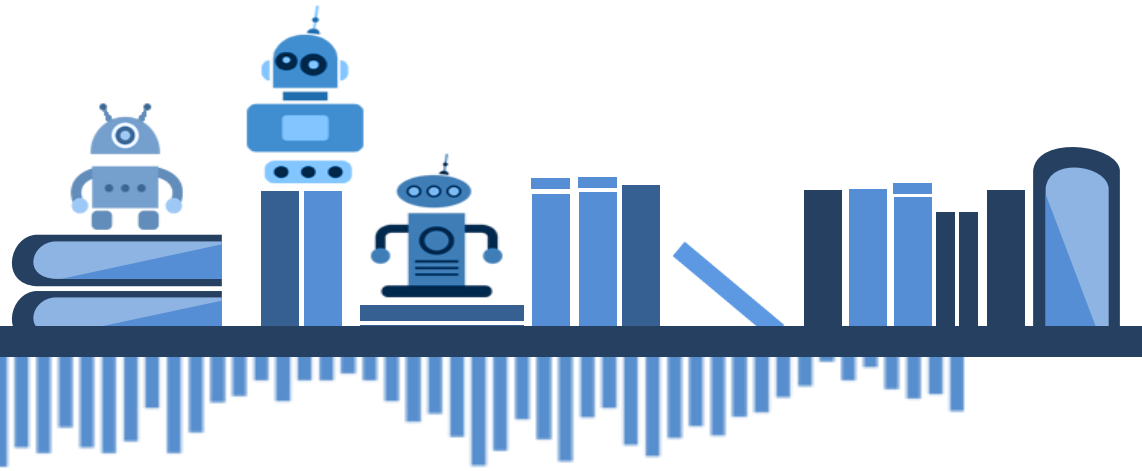
Ajax数据爬取

主讲老师：张涛

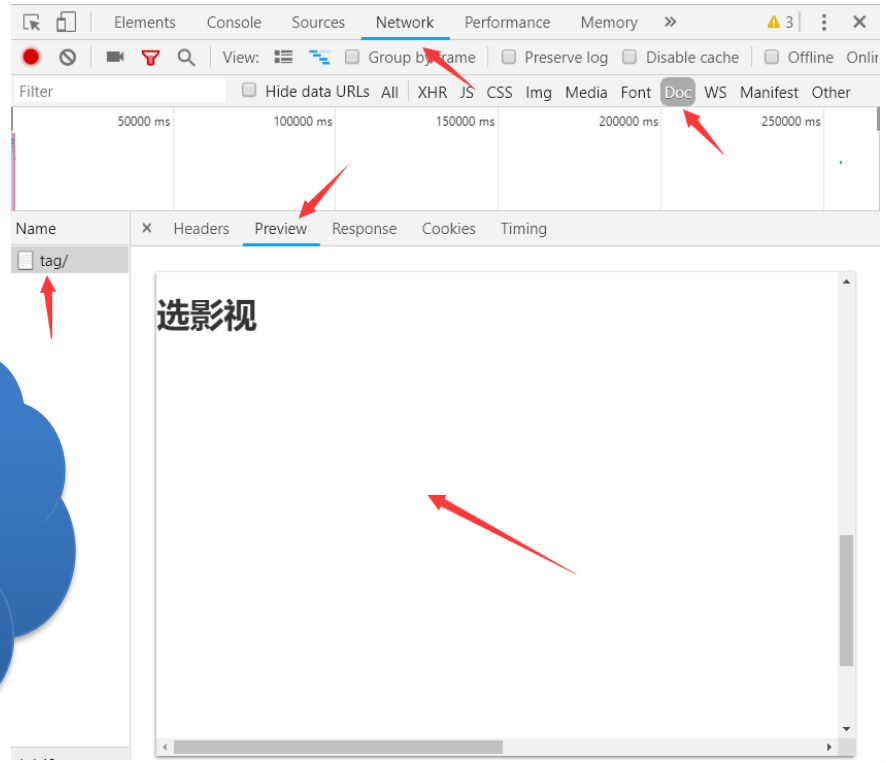


目录

问题

学到这里，很多人可能会跃跃欲试，到各大网站爬取自己想要的数据了。但是，现实是你可能会乘兴而来败兴而归。因为更大的难题还在等着我们，例如，看下面的例子。

下图是豆瓣电影的影视页，网址为：https://movie.douban.com/tag/#/。



什么数据都没爬到

现在大部分的网站，都使用一种叫做Ajax的技术来加载一些数据，而我们爬取的仅仅是不包含Ajax数据的原始HTML页面数据，所以，爬虫当然会失败了。

相对传统页面，使用Ajax的好处显而易见：

- 更好的用户体验

  无需频繁手动重新刷新整个页面，，减少了页面加载的等待时间，大大提升了用户体验，增加用户粘性。

- 节省流量

  由于只更新了局部页面的数据，相对加载整个页面来说，大大节省了流量，减轻了服务器的负担。

Ajax概念

Ajax，全称**A**synchronous **J**avaScript **a**nd **X**ML，即异步的JavaScript和XML。它不是新的编程语言，而是利用JavaScript在不重新加载整个页面的情况下，与服务器交换数据并更新部分网页内容的技术。

基本原理

来看一下Ajax实现的基本原理，发送Ajax请求到网页更新的过程简单分为三个步骤：

1、发送请求

当页面的某个地方需要更新数据时，会向服务器发送Ajax请求，请求的最底层是通过JavaScript实现的。

2、解析内容

服务器发送响应内容，可能是HTML代码或者JSON数据。使用JavaScript处理响应数据。

3、渲染网页

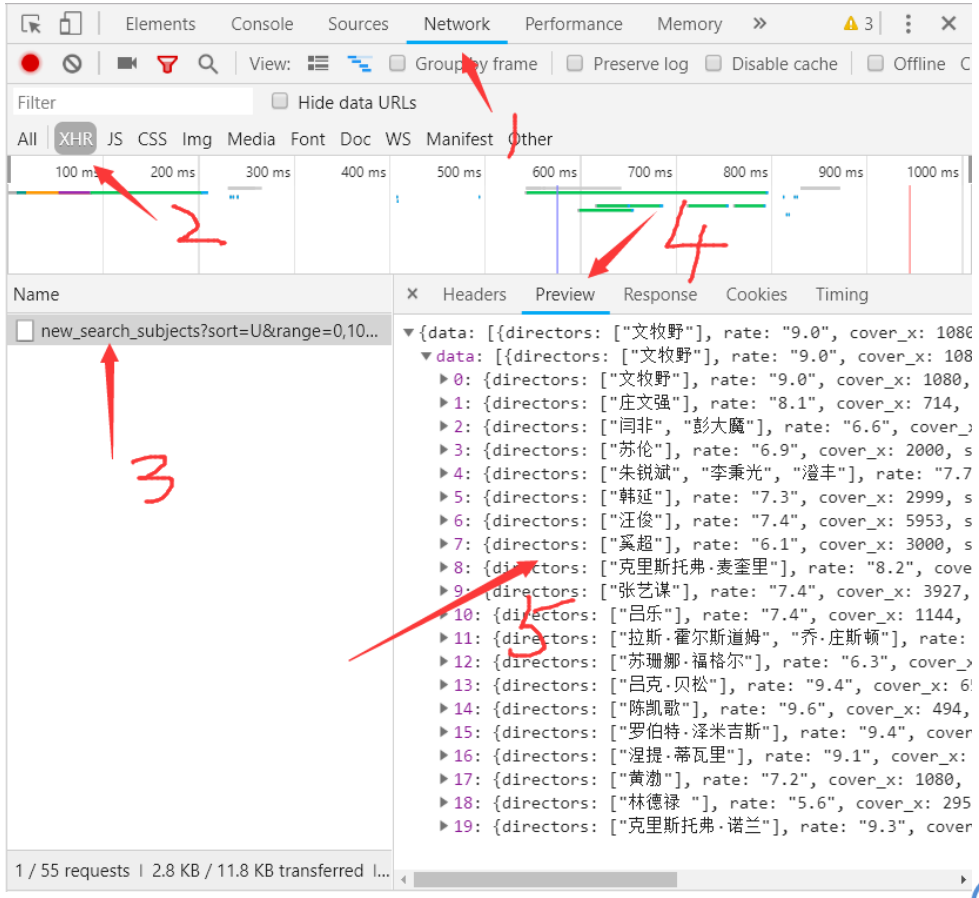
JavaScript有改变网页内容的能力，解析完响应内容之后，就可以调用JavaScript来针对解析完的内容对网页进行下一步处理了。

Ajax分析步骤

如何获取通过Ajax获取的数据呢?

答案是：使用Chrome浏览器的“开发者工具”。

以豆瓣电影的影视页面为例（https://movie.douban.com/tag/#/）。
打开页面后，使用“开发者工具”，按照下图的顺序点击。
我们会发现，影视信息的数据都显示在第5步中了。



获得了Ajax数据的URL，就可以像爬取普通URL一样，爬取Ajax的数据了，只是一般情况下，得到的是Json数据。

练习：爬取豆瓣电影的影视页面信息（https://movie.douban.com/tag/#/）

问题

当我们获取了Ajax数据的URL时，经常会发现一个令人头疼的问题，如下面为今日头条热点的的AjaxURL地址：

https://www.toutiao.com/api/pc/feed/?category=news_hot&utm_source=toutiao&widen=1&max_behot_time=0&max_behot_time_tmp=0&tadrequire=true&as=A195DBDC06ADC25&cp=5BC65DAC62854E1&_signature=ZtN.cQAAPRnM.D.xF5yhvGbTf2

没错，它：

- 既冗长又复杂
- 经过加密
- 具有时效性
- 毫无规律可言

如何解决呢?

一种方法是：使用模拟浏览器运行的方式。

它可以做到在浏览器中看到的是什么样，抓取的源码就是什么样，即可见即可爬。再也不用管网页内容是使用了JavaScrapt还是Ajax，也不用管接口有多复杂了（其实连接口是什么都不用管）。

Selenium

Selenium 是一个用于测试Web应用程序的工具，它直接运行在浏览器中，就像真正的用户在操作一样。通俗点讲，它能直接驱动浏览器，模拟人的各种操作行为，如下拉、鼠标点击、键盘输入、拖拽等操作。最后可以获取页面渲染后的HTML代码。

1. 安装Selenium库

pip install selenium

2.安装浏览器驱动程序

（1）查看Chrome浏览器的版本。

（2）下载Chromedriver。

官方下载地址：https://chromedriver.storage.googleapis.com/index.html。
其他下载地址：http://npm.taobao.org/mirrors/chromedriver/。

（3）配置环境变量。

在Windows下，将下载得到的chromedriver.exe文件拖到Anaconda3的Scripts目录下就可以了，如：C:\Anaconda3\Scripts下。

声明浏览器对象

```
from selenium import webdriver
driver = webdriver.Chrome()  #声明Chrome浏览器对象
driver = webdriver.ie()      #声明ie浏览器对象
driver = webdriver.firefox()  #声明firefox浏览器对象
driver = webdriver.phantomjs()#声明phantomjs浏览器对象
driver = webdriver.safari()   #声明safari浏览器对象
```

访问页面

```
driver.get("https://www.suning.com/")#请求页面
```

获取页面代码

```
#获取代码
HTML=driver.page_source
```

定位元素

以下为Selenium查找单个节点的方法。

- find_element_by_id：通过ID查找
- find_element_by_name：通过NAME查找
- find_element_by_xpath：通过xpath选择器查找
- find_element_by_link_text：通过链接的文本查找（完全匹配）
- find_element_by_partial_link_text：通过链接的文本查找（部分匹配）
- find_element_by_tag_name：通过标签名查找
- find_element_by_class_name：通过CLASS查找
- find_element_by_css_selector：通过css选择器查找


页面交互

Selenium可以模拟用户对页面执行一系列操作，如输入数据、清除数据、单击按钮等。以下代码实现了定位到搜索框后，清空搜索框中的文字，输入“iphone”，回车的功能。

```
input = driver.find_element_by_id("searchKeywords")#查找节点
input.clear()#清除输入框中默认文字
input.send_keys("iphone")#输入框中输入“iphone”
input.send_keys(Keys.RETURN)#回车功能
```

```
from selenium.webdriver.common.keys import Keys#导入Keys类
```

执行JavaScript

Selenium并未提供所有的页面交互操作方法，例如爬虫中用得最多的下拉页面（用于加载更多内容）。Selenium提供了execute_script()方法，用于执行JS，这样我们就可以通过JS代码实现这些操作了。以下代码实现了将页面下拉到底部的功能。

```
driver.execute_script('window.scrollTo(0, document.body.scrollHeight)')
```

需求：使用爬虫获取今日头条中热点新闻的获取，保存于CSV文件中。获取的字段有：新闻标题、来源和评论数。如下图所示。

结束 谢谢收看