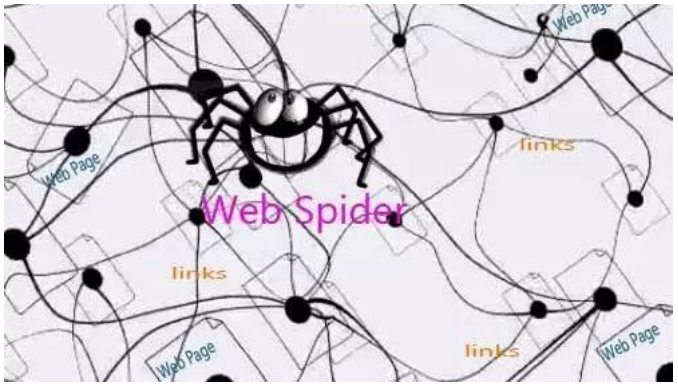
个人分享 | 我的常规爬虫流程

原创： hoxis



阅读文本大概需要 3.6 分钟。

其实，我鼓捣的有些也算不上是爬虫。

首先，爬虫不是我的本职工作，我爬虫一般是为了一些有意思的东西，获取一些信息，或者是实现一些可以自动化完成的任务，比如签到。

一般我的爬虫流程是这样的：

1、浏览器访问待爬网页，并提前打开开发者工具（F12），选中 Nework 选项卡，这样就可以看到网络交互信息；

或者，右键查看网页源代码，查找目标信息。

2、在网络交互信息流中筛选出自己需要的，然后在 postman 中模拟请求，看是否仍然可以获取到想要的信息；

postman 除了可以进行请求测试外，还有一个优势就是，代码可以直接生成，这样就可以方便得进行最终的整合了。

3、数据解析，从请求的响应中解析出我们的目标数据，至于得到数据后如何处理，那就是你的事情了。

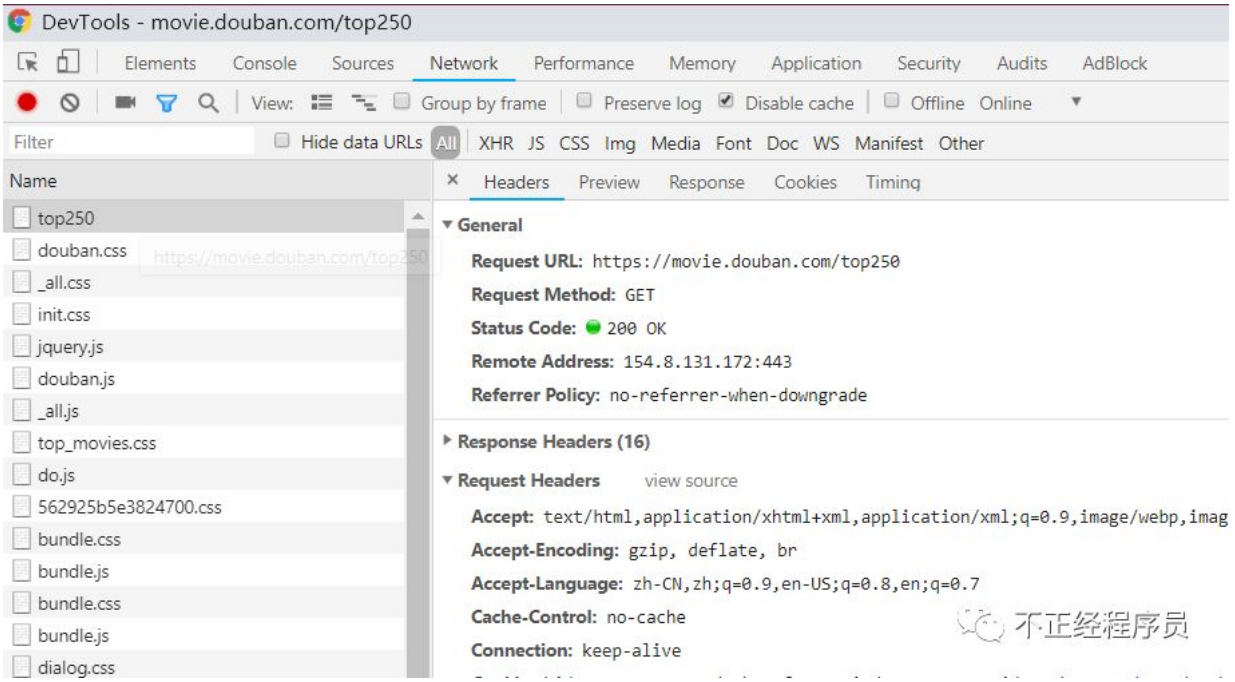
下面就以大家耳熟能详（landajie）的豆瓣电影 TOP250 为例。

实例分析

请求梳理

首先，我们要访问待爬取的网页：https://movie.douban.com/top250。

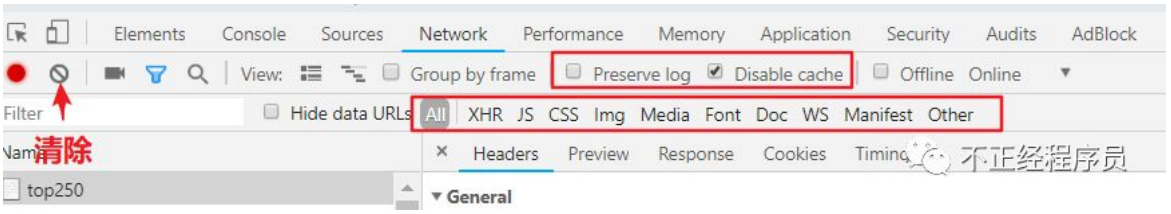
一般情况下，我都是直接按下 F12 调出 DevTools，点击 Network 选项卡：



豆瓣TOP250

有时请求已经加载完成了，可以把数据全部 clear 掉，然后重新刷新网页，这时候请求流会重新加载。

这里有几个点需要注意，主要是下图圈红的几个：



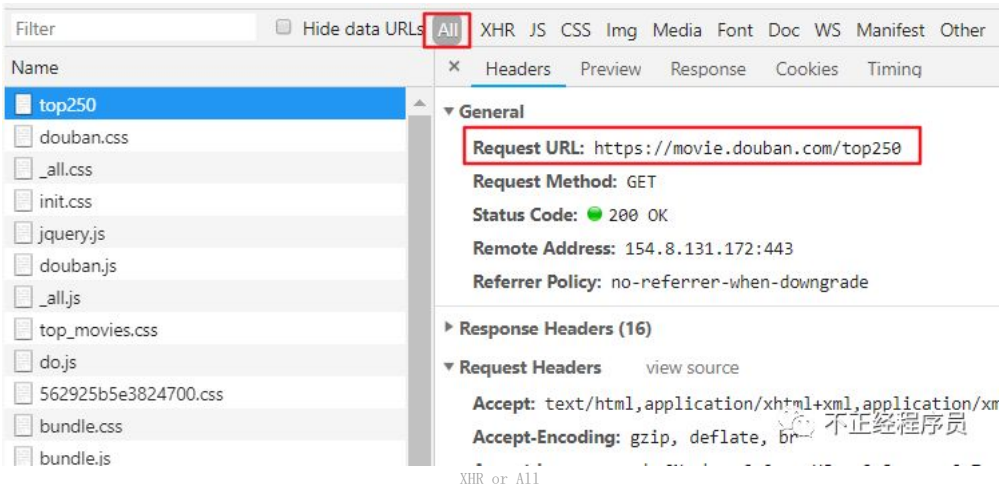
DevTools选项

1、有些网页请求会有自动跳转，这是请求流会重新加载，这是勾选了 Preserve log 的话，数据就会持续打印，不会被冲掉；

2、勾选 Disable cache 可以禁用缓存；

3、请求流的筛选：XHR 是 XMLHttpRequest 的意思，大多数情况下只要点击 XHR 就行了，但是若此时发现没有想要的请求数据，那么就要点击 All 展示所有请求流。

比如豆瓣的这个，XHR 中是没有我们的目标请求的。

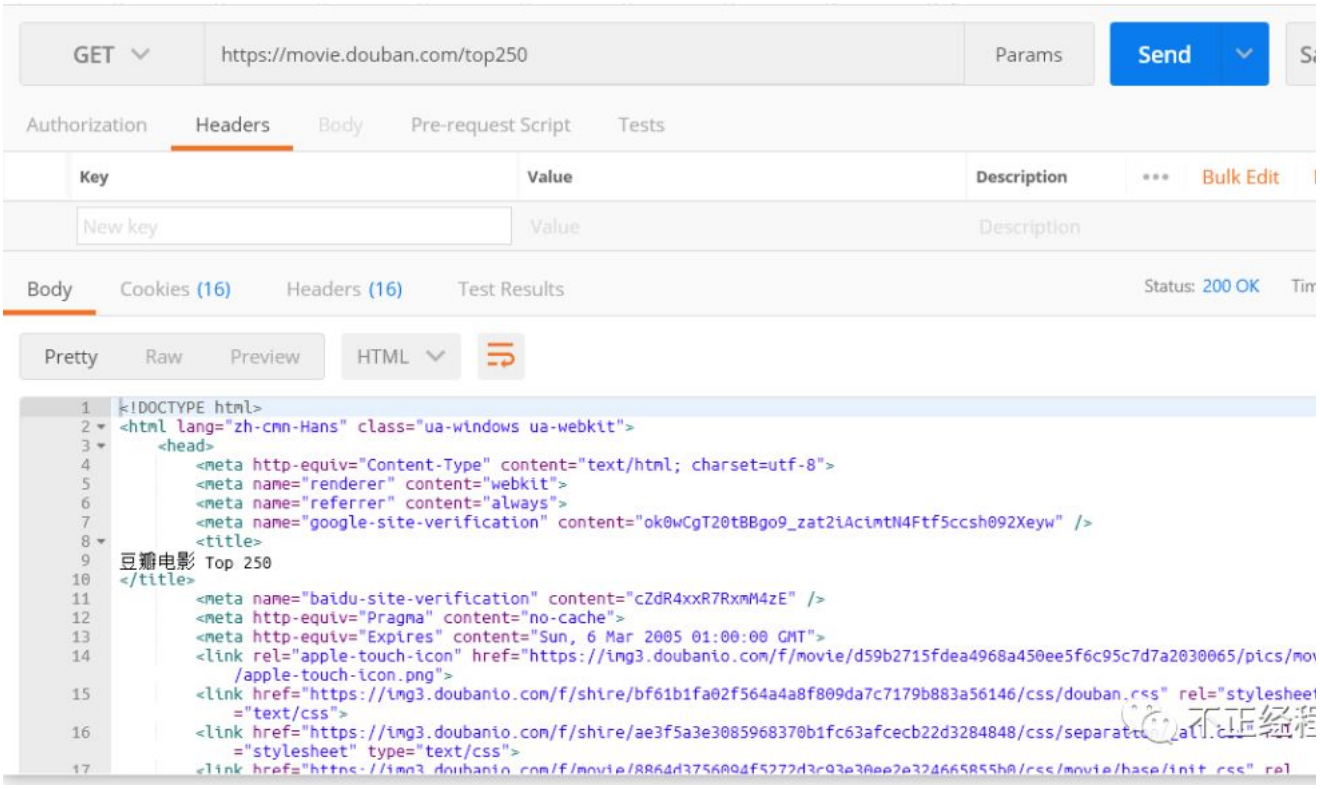


XHR or All

请求模拟

通过上面的步骤，我们能够确定通过哪些请求能够得到我们的目标数据，然后把这些请求放到 postman 中进行模拟。

比如，我们在 postman 中访问豆瓣的网站：

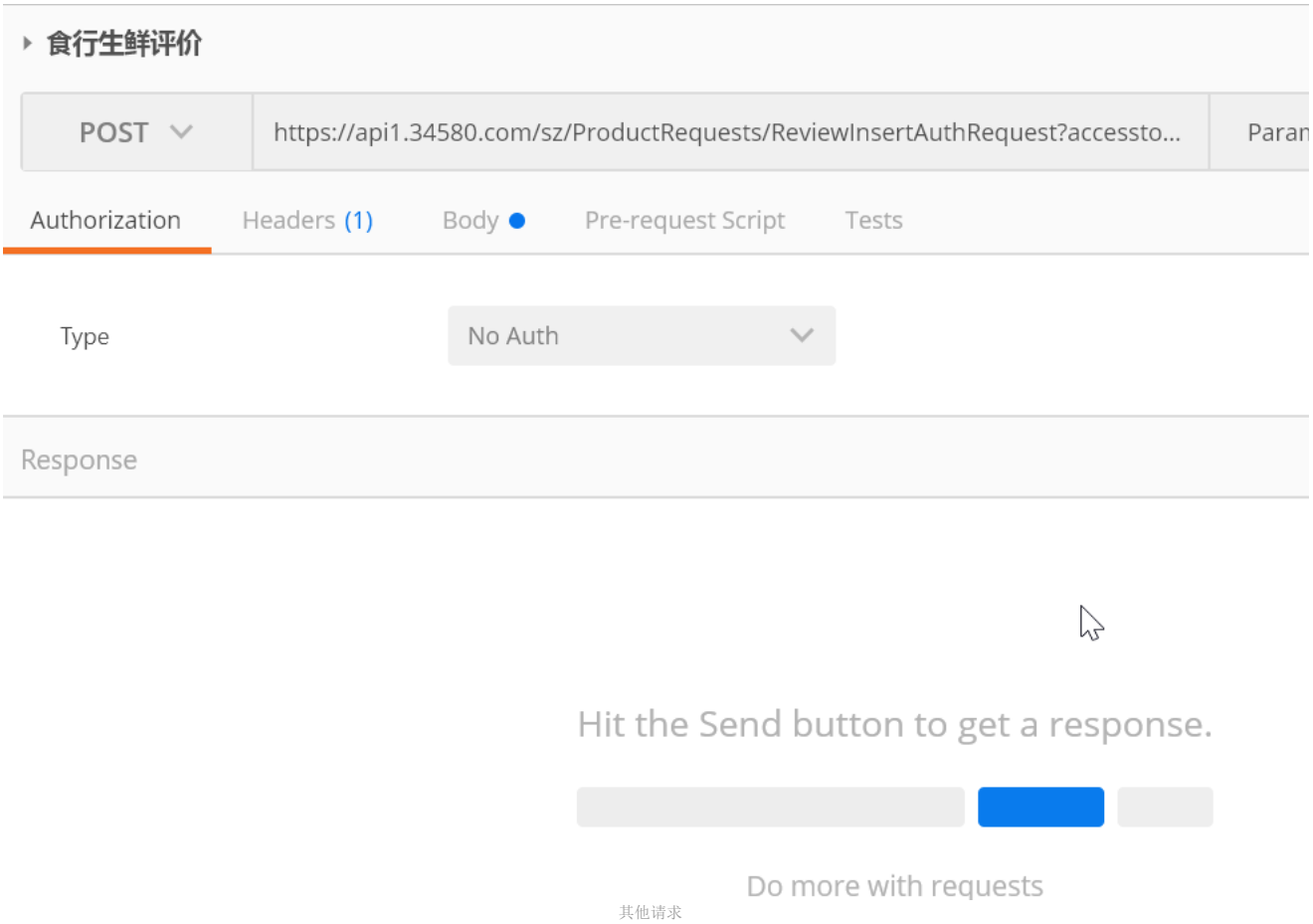


postman访问

这里的请求比较简单，直接 get url 就能获取到目标数据。

其实大部分情况下，都是需要添加一些访问参数的，这是我们可以在 Headers 里添加。

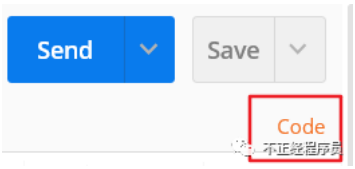
另外，postman 还支持其他请求，如 post、delete 等等：



其他请求

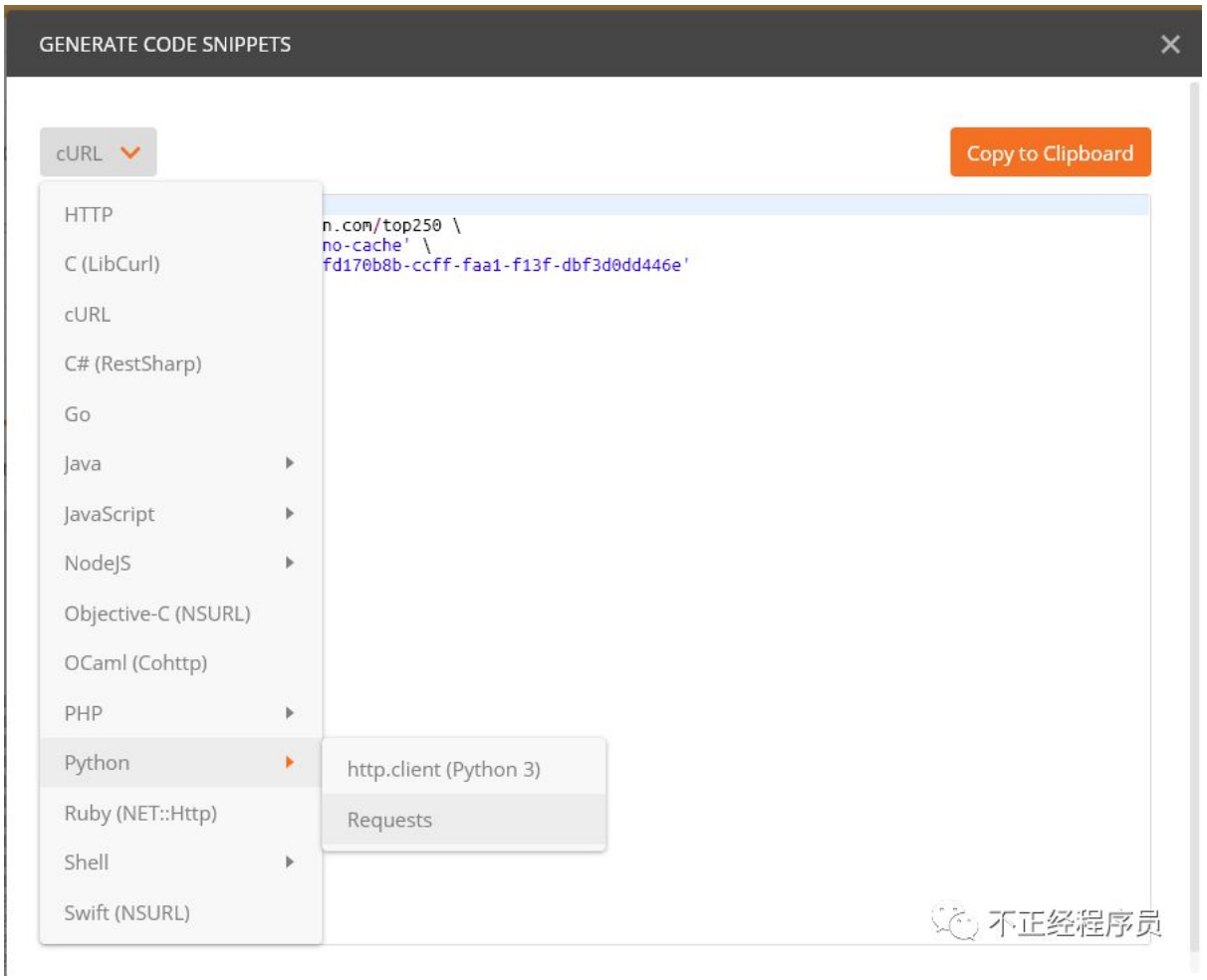
- 生成代码

点击右侧的 code 按钮，就可以获取到对应的代码：



code

支持生成多种语言的代码：



多种语言

比如，我们这里选择 Python Requests，就可以得到如下代码：

```
import requests

url = "https://movie.douban.com/top250"

headers = {

    'cache-control': "no-cache",

    'postman-token': "d2e1def2-7a3c-7bcc-50d0-eb6baf18560c"

    }

response = requests.request("GET", url, headers=headers)

print(response.text)
```

这样我们只要把这些代码合并到我们的业务逻辑里就行了，当然其中的 postman 相关的参数是不需要的。

数据解析

下面要做的就是从响应中解析目标数据。

有些响应是返回 HTML，有些是返回 json 数据，有的还是返回 XML，当然也有其他的，这就需要不同的解析逻辑。

具体如何解析，这里我们不再赘述，之前的爬虫文章中都有涉及，有兴趣的可以翻一翻。

总结

本来打算写 postman 的使用的，但是写来写去，成了我的一般爬虫流程梳理。

本文涉及的爬虫都是比较初级的，至于 ip 代理、验证码解析等高端功能，后面有时间再单独说。

不知道你的一般流程是什么样的，不妨留言分享下。

选 Python 还是Java ？

Python 抓取「知识星球」精华并生成电子书

谁说 HTTP GET 就不能通过 Body 来发送数据呢？

网址中最后的斜杠 / 是干嘛的?

如何将 Python 程序打包成 .exe 文件？

教你用 Python 来朗读网页

Python 玩转 Excel

你还在用 format 格式化字符串？

阅读原文