链滴

使用 java httpClient 做爬虫, 一些总结

作者：sulinfyx
原文链接：https://ld246.com/article/1512384813303
来源网站：链滴

总结

过去一直使用 mechanize 包来做爬虫, 功能非常强大, 几乎无所不能。最近正在将技术栈转移到Java, 以开发抓取功能时也想找到和 mechanize 一样强大的包。

工具采用 HttpClient

http请求头信息

文档: http://hc.apache.org/httpclient-3.x/logging.html

通过配置，能得到像 mechanize 里面的请求头信息

```
DEBUG [org.apache.http.headers] http-outgoing-0 >> GET / HTTP/1.1
DEBUG [org.apache.http.headers] http-outgoing-0 >> Host: www.baidu.com
DEBUG [org.apache.http.headers] http-outgoing-0 >> Connection: Keep-Alive
DEBUG [org.apache.http.headers] http-outgoing-0 >> User-Agent: Apache-HttpClient/4.5.3 (
ava/1.8.0_144)
DEBUG [org.apache.http.headers] http-outgoing-0 >> Accept-Encoding: gzip,deflate
DEBUG [org.apache.http.headers] http-outgoing-0 << HTTP/1.1 200 OK
DEBUG [org.apache.http.headers] http-outgoing-0 << Server: bfe/1.0.8.18
DEBUG [org.apache.http.headers] http-outgoing-0 << Date: Mon, 04 Dec 2017 07:29:18 GMT
DEBUG [org.apache.http.headers] http-outgoing-0 << Content-Type: text/html
DEBUG [org.apache.http.headers] http-outgoing-0 << Last-Modified: Mon, 23 Jan 2017 13:28
24 GMT
DEBUG [org.apache.http.headers] http-outgoing-0 << Transfer-Encoding: chunked
DEBUG [org.apache.http.headers] http-outgoing-0 << Connection: Keep-Alive
DEBUG [org.apache.http.headers] http-outgoing-0 << Cache-Control: private, no-cache, no-s
ore, proxy-revalidate, no-transform
DEBUG [org.apache.http.headers] http-outgoing-0 << Pragma: no-cache
DEBUG [org.apache.http.headers] http-outgoing-0 << Set-Cookie: BDORZ=27315; max-age=
6400; domain=.baidu.com; path=/
DEBUG [org.apache.http.headers] http-outgoing-0 << Content-Encoding: gzip
```

修改user-agent

通过请求头信息看到, User-Agent: Apache-HttpClient/4.5.3 (Java/1.8.0_144) 这肯定是不行的，高爬虫肯定是需要做到100%模拟用户行为，所以User-Agent肯定需要进行修改，mechanize 里面很易修改，httpclient也能做到非常简单的修改。

```
CloseableHttpClient httpclient = HttpClients.createDefault();
HttpGet httpGet = new HttpGet("http://www.baidu.com");
httpGet.setHeader("User-Agent", "Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:50.0) Gecko/2
100101 Firefox/50.0");
```

这样设置是不方便的，每次请求都需要设置, 如果能在初始化 httpclient 的时候，就设置好，后面每请求就不用单独设置了。代码如下:

```
CloseableHttpClient httpclient = HttpClients.custom()
  .setUserAgent("Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:50.0) Gecko/20100101 Firefox/5
.0")
```

```
  .build();
HttpGet httpGet = new HttpGet("http://www.baidu.com");
CloseableHttpResponse response1 = httpclient.execute(httpGet);
```

修改reffer

reffer 在http请求中是非常重要的, 直接标明来源, 防爬虫策略中校验是否存在reffer或者判断reffer是正确, 显示非常重要。mechanize 在请求中修改reffer是非常容易的, 我们来看看 httpclient 中如何来改reffer。

```
CloseableHttpClient httpclient = HttpClients.custom()
  .setUserAgent("Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:50.0) Gecko/20100101 Firefox/5
.0")
  .build();
HttpGet httpGet = new HttpGet("http://www.baidu.com");
httpGet.addHeader("Referer", "http://www.google.com");
CloseableHttpResponse response1 = httpclient.execute(httpGet);
```

请求上下文

爬虫往往不仅仅只访问一个页面,而是一连串几个页面, 比如获取用户的余额, 步骤为:

1、访问登录界面，进行用户名和密码登录

2、进入个人中心页面

3、访问余额页面, 解析出余额

以上3步,是连续的,需要每一次访问种的Cookie,需要一个关联的上下文信息。在 mechanize 里，是非容易做到这一点的, 声明一个httpclient对象, 在一个进程里，多次请求自然就关联上了。httpclient里又是如何操作呢? HttpClient 4.x 的版本已经自动实现了

参考文档: http://blog.csdn.net/column/details/httpclient.html

代理

使用代理去爬取相关网页,在爬虫技术中是必须要掌握的, httpclient非常简单, 如下:

```
HttpGet httpGet = new HttpGet("http://www.baidu.com");
HttpHost proxy = new HttpHost("111.178.233.26", 8081);
RequestConfig requestConfig = RequestConfig.custom().setProxy(proxy).build();
httpGet.setConfig(requestConfig);
CloseableHttpResponse response1 = httpclient.execute(httpGet);
```

解析网页

jsoup 是一款 Java 的HTML 解析器，可直接解析某个URL地址、HTML文本内容。它提供了一套非省力的API，可通过DOM，CSS以及类似于jQuery的操作方法来取出和操作数据。