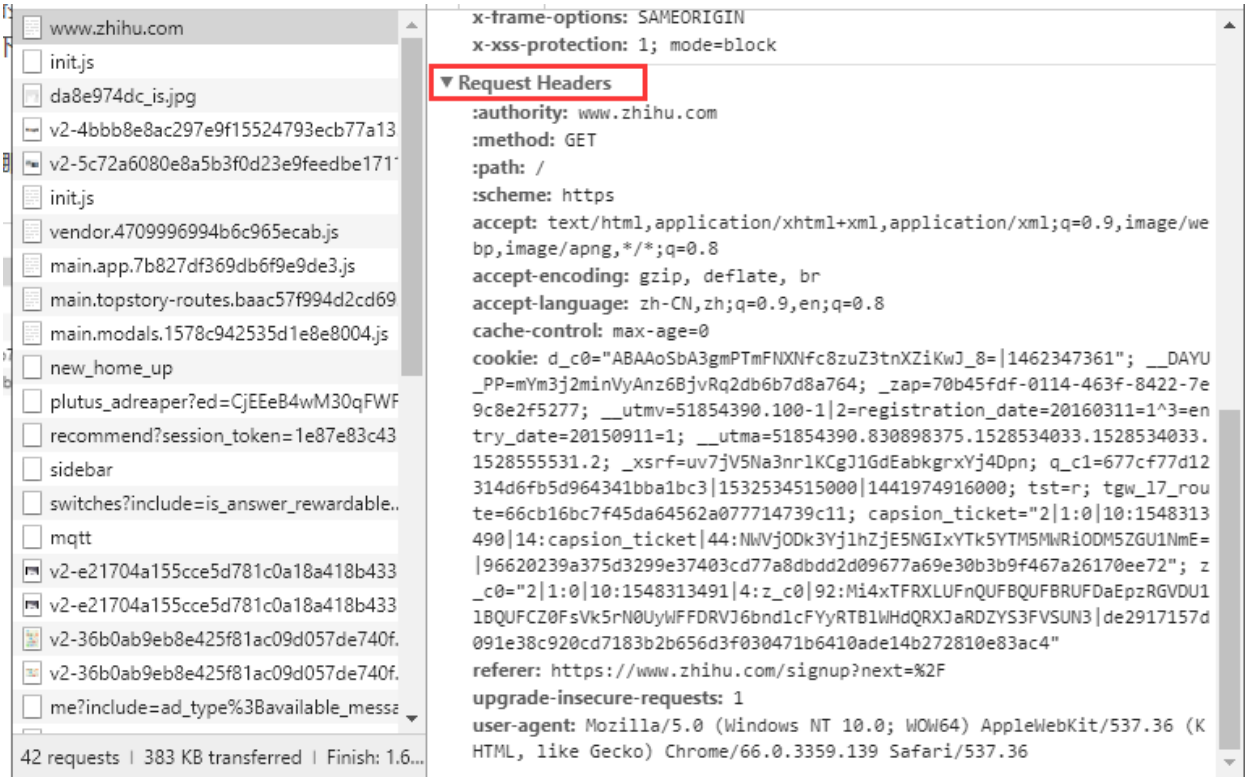
爬虫01-HTTP协议之请求

和大家简单分享一下HTTP协议中的请求，这些内容是与爬虫息息相关的。

我们向一个网址（也就是URL，之后采取的都会是URL的称呼）发起请求，这个URL所在的服务器会相应地返回我们一个结果，这个结果就是响应。

看起来似乎是一个很简单的流程，其实不然。无论是请求还是响应，都会隐形地携带很多和请求、响应有关的内容，接下来，我们就来看看都会携带哪些参数（参数有很多，我们挑下重要的讲讲）。

我们来看看请求报文中我们需要了解哪些与爬虫有关的信息。



1. method：这个字段是用来指明请求的方法是哪一种的，常用的请求方法有GET、POST，这两种请求有什么区别、以及分别适用什么场景，后面我们会详细讲解。如果method是GET的时候，在使用requests的时候，就只能用requests.get()，比如这样：

```
1 import requests
```

```
response = requests.get('https://www.zhihu.com')
```

如果method是POST的时候，在使用requests的时候，就只能用requests.post()，而post是请求是需要传递数据的，这个之后会详细介绍。比如这样：

```
import requests
data = {
 'username': 'shiyue',
 'password': 'shiyue'
}
response = requests.post('https://www.wzhihu.com', data=data)
```

1. Accept：这个字段是用来通知服务器，用户代理（浏览器等客户端）能够处理的媒体类型及媒体类型的相对优先级。可以使用type/subtype这种形式，一次可指定多种媒体类型。常用的媒体类型有以下几类：

文本文件：text/html，text/plain，text/css，application/xhtml+xml，application/xml...
图片文件：image/jpeg，image/gif，image/png...
视频文件：video/mpeg，vedio/quicktime...
应用程序使用的二进制文件：application/octer-stream，application/zip...

比如说，浏览器不支持图片PNG的显示，那么accept就不指定image/png，因为浏览器处理不了。（这个字段需要了解，尤其是如果后期想从事网站开发的童鞋）

1. Cookie：客户端发起请求时，服务器会返回一个键值对形式的数据给浏览器，下一次浏览器再访问这个域名下的网页时，就需要携带这些键值对数据在 Cookie中，用来记录用户在当前域名下的历史行为的。

提到Cookie就不得不提，HTTP本身是一种无状态的协议，它是不会保存每次请求和响应的相关信息的，就比如我们登录淘宝，如果没有Cookie技术，那么我们每进入一个淘宝的页面都需要重新登录一次，这样是不是会特别麻烦？

就是因为这个原因，才引入了Cookie技术，使得用户在一个域名下的历史行为能够得以保存，只要登录一次淘宝就可以了，不需要频繁地登录，而且能够看到历史记录。

这个字段很重要，在爬虫中经常会用到，因为有的数据只有携带了Cookie才能够爬取到，所以经常会根据前次访问得到 cookie数据，然后添加到下一次的访问请求头中。

就像这样：

```python
import requests
url = 'https://www.baidu.com'

headers = {
 'cookie': 'PSTM=1496322685; BIDUPSID=BC36002F7DA142E6674AE290CD5A38DB; __cfduid=ddf4836dd1f1ac99eeea8ef0f140493301522406372; BAIDUID=5FA7A2B4FDDA3CECC6BE9B74FDCD00B8:FG=1; sugstore=1; BDUSS=1hvT3VoQmc5TDl5bFE1c2NjcGpOenBycnJTOEstZ0ZKcGs5UnB1dzVyU1hpYXRiQVFBQUFBJCQAAAAAAAAAAAAEAAABCcYSYAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAJf8g1uX~INbR; BD_UPN=12314753; MCITY=-340%3A; delPer=0; BD_CK_SAM=1; PSINO=7; BDRCVFR[Dq4jqEr7erC]=mk3SLVN4HKm; H_PS_PSSID=; BDORZ=FFFB88E999055A3F8A630C64834BD6D0; H_PS_645EC=216feJgh%2BnAm%2BJD6G3sw10RBbYN1O%2FeCJqUhgtRyZ3OuJO0EOqbXUwL8Kgf8zhqXH7RWxBnn; BDSVRTM=0; ispeed_lsm=6'
}

response = requests.get(url, headers=headers)
```

1. Refer:这个字段用来记录浏览器上次访问的URL，有的网站会通过请求中有没有携带这个参数来判断是不是爬虫，从而确定是否限制访问。所以有时候也需要在headers中添加上这个参数。

1. User-Agent:是用来标识请求的浏览器身份的，大部分网站都会通过请求中有没有携带这个参数来判断是不是爬虫，从而确定是否限制访问。所以有时候也需要在headers中添加上这个参数。

像这样：

```python
import requests
url = 'https://www.baidu.com'
headers = {
 'user-agent': 'Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/66.0.3359.139 Safari/537.36'
}
response = requests.get(url, headers=headers)
```

但当我们要爬取的数据量比较大的时候，仅仅用一个user-agent是不够的，因为服务器又不傻，你一个浏览器不停地在访问我的URL，而且频率那么快，肯定不是人在后面操作，然后就会限制你的访问了，所以我们经常会用一个user-agent列表，来回地切换。这样服务器就会以为是多个浏览器（也就是多个用户）在访问URL，会判断这是正常的。

像这样：

```
import requests,  random
url = 'https://www.baidu.com'
agent_list = ['Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/66.0.3359.139 Safari/537.36',
'Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10_6_8; en-us) AppleWebKit/534.50 (KHTML, like Gecko) Version/5.1 Safari/534.50',
'Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Trident/5.0',
'Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)'
]
# 这是一个User-Agent列表，在使用的时候随机从中选取一个作为请求头的参数传递进去
headers = {
  'user-agent': random.choice(agent_list)
 }
response = requests.get(url, headers=headers)
```

这就是请求报文中比较常用且比较重要的5个字段，希望能够对大家去理解爬虫会有一点帮助。

HTTP请求报文

组成： 请求行 + 请求头 + 请求体

请求行：请求方法，请求地址（URL），HTTp版本

请求头（报文头）：以key-value方式存储，存储是客户端信息

请求体（报文体）：需要传递的一些参数信息

HTTP响应报文

HTTP响应报文

组成： 响应行 + 响应头 + 响应体

响应行：HTTp版本，状态码

响应头（报文头）：服务端信息

响应体（报文体）：需要传递的数据

状态码：

2开头

2xx:成功类

3xx：重定向

4xx：客户端错误

5xx：服务端错误