

# 爬虫 day1 Spider

## 基本概念



爬虫就是获取网页并提取和保存信息的自动化程序

企业想用爬虫干什么？

你想用爬虫来干什么？

## HTTP基本原理

### URI和URL

URI (Uniform Resource Identifier) 统一资源标志符

URL(Universal Resource Locator) 统一资源定位符

URL是URI的子集，URI还包括一个子类URN (Universal Resource Name) 统一资源名称，URN只命名资源不指定如何定位资源。

### 超文本 (hypertext)

网页源代码，html代码

查看源代码工具和方法

# HTTP和HTTPS

HTTP (Hyper Text Transfer Protocol)

超文本传输协议

HTTPS (Hyper Text Transfer Protocol over Secure Socket Layer)

HTTP加入SSL层，传输内容通过SSL加密

- 安全通道保证数据传输安全
- 确认网站真实性

## HTTP请求过程

用浏览器开发者工具观察网络请求过程

### 请求

请求方法 (Request Method)

GET请求的参数直接在URL里，最多只有1024字节

POST请求数据一般通过表单提交，不会出现在URL里，大小没有限制

序号	方法	描述
1	GET	请求指定的页面信息，并返回实体主体。
2	HEAD	类似于get请求，只不过返回的响应中没有具体的内容，用于获取报头
3	POST	向指定资源提交数据进行处理请求（例如提交表单或者上传文件）。数据被包含在请求体中。POST请求可能会导致新的资源的建立和/或已有资源的修改。
4	PUT	从客户端向服务器传送的数据取代指定的文档的内容。
5	DELETE	请求服务器删除指定的页面。
6	CONNECT	HTTP/1.1协议中预留给能够将连接改为管道方式的代理服务器。
7	OPTIONS	允许客户端查看服务器的性能。
8	TRACE	回显服务器收到的请求，主要用于测试或诊断。

### 请求头

#### Cache-Control

指定了服务器和客户端在交互时遵循的缓存机制，即是否要留下缓存页面数据。

一般在使用浏览器访问时，都会在计算机本地留下缓存页面，相当于是浏览器中的页面保存和下载选项。但是爬虫就是为了从网络上爬取数据，所以几乎不会从缓存中读取数据。所以在设

置的时候要侧重从服务器请求数据而非加载缓存。

- no-cache: 客户端告诉服务器, 自己不要读取缓存, 要向服务器发起请求
- no-store: 同时也是响应头的参数, 请求和响应都禁止缓存, 即不存储
- max-age=0: 表示当访问过此网页后的多少秒内再次访问, 只加载缓存, 而不去服务器请求, 在爬虫时一般就写0秒

一般爬虫就使用以上几个参数, 其他的参数都是接受缓存的, 所以就不列出了。

- User-Agent

中文名用户代理, 服务器从此处知道客户端的 操作系统类型和版本, 电脑CPU类型, 浏览器种类版本, 浏览器渲染引擎, 等等。这是爬虫当中最最重要的一个请求头参数, 所以一定要伪造, 甚至多个。如果不进行伪造, 而直接使用各种爬虫框架中自定义的user-agent, 很容易被封禁。举例:

- User-Agent: Mozilla/5.0 (X11; Linux x86\_64; rv:52.0) Gecko/20100101 Firefox/52.0
- User-Agent: Mozilla/5.0 (Windows NT 6.3; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/52.0.2743.116 Safari/537.36

## Accept

指定客户端可以接受的内容类型, 比如文本, 图片, 应用等等, 内容的先后排序表示客户端接收的先后次序, 每种类型之间用逗号隔开。

其中, 对于每一种内容类型, 分号 ; 后面会加一个 q=0.6 这样的 q 值, 表示该种类型被客户端喜欢接受的程度, 如果没有表示 q=1, 数值越高, 客户端越喜欢这种类型。

爬虫的时候, 一般会伪造若干, 将想要找的文字, 图片放在前面, 其他的放在后面, 最后一定加上;q=0.8。

- 比如Accept: image/gif,image/x-bitmap,image/jpeg,application/x-shockwave-flash,application/vnd.ms-excel,application/vnd.ms-powerpoint,application/msword,txt/xml,txt/shtml: 文本类型, 斜杠后表示文档的类型, xml, 或者shtml
- application/xml,application/xhtml+xml: 应用类型, 后面表示文档类型,比如 flash动画, excel表格等等
- image/gif,image/x-bitmap: 图片类型, 表示接收何种类型的图片
- /: 表示接收任何类型, 但是这一条一般写在最后, 表示优先接收前面规定的类型, 然后再加载其他类型。

## Accept-Language

客户端可以接受的语言类型, 参数值规范和 accept的很像。一般就接收中文和英文, 有其他语言需求自行添加。比如:

- Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.6,en;q=0.4
- zh-CN: 中文简体大陆?
- zh: 其他中文

- en-US: 英语美语
- en: 其他英语

## Accept-Encoding

客户端接收编码类型，一些网络压缩格式：

- Accept-Encoding: gzip, deflate, sdch。相对来说，deflate是一种过时的压缩格式，现在常用的是gzip

## Accept-Charset

指的是规定好服务器处理表单数据所接受的字符集，也就是说，客户端浏览器告诉服务器自己的表单数据的字符集类型，用以正确接收。若没有定义，则默认值为“unknown”。如果服务器没有包含此种字符集，就无法正确接收。一般情况下，在爬虫时不定义该属性，如果定义，例子如下：

- Accept-Charset: gb2312,gbk;q=0.7,utf-8;q=0.7,\*;q=0.7

## Referer

浏览器上次访问的网页url，uri。由于http协议的无记忆性，服务器可从这里了解到客户端访问的前后路径，并做一些判断，如果后一次访问的 url 不能从前一次访问的页面上跳转获得，在一定程度上说明了请求头有可能伪造。

## DNT

是 do not track 的缩写，告诉服务器，浏览器客户端是否禁止第三方网站追踪。这一条主要是用来保护浏览器用户隐私的，通过此功能，用户可以检测到跨站跟踪、cookie跟踪等等。在爬虫时一般都是禁止的。数字1代表禁止追踪，0代表接收追踪，null代表空置，没有规定。

## Connection

请求头的 header 字段指的是当 client 浏览器和 server 通信时对于长链接如何处理。由于http请求是无记忆性的，长链接指的是在 client 和server 之间建立一个通道，方便两者之间进行多次数据传输，而不用来回传输数据。有 close, keep-alive 等几种赋值，close表示不想建立长连接在操作完成后关闭链接，而keep-alive 表示希望保持畅通来回传输数据。爬虫时一般都建立一个长链接。

## Proxy-Connection

当使用代理服务器的时候，这个就指明了代理服务器是否使用长链接。但是，数据在从client到代理服务器，和从代理服务器到被请求的服务器之间如果存在信息差异的话，会造成信息请求不到，但是在大多数情况下，都还是能够成立的。

## Pragma

防止页面被缓存, 和 cache-control类似的一个字段, 一般爬虫都写成 no-cache。

## Cookie

同样是一个比较关键的字段, Cookie是 client 请求 服务器时, 服务器会返回一个键值对样的数据给浏览器, 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, 用来跟踪浏览器用户的访问前后路径。

在爬虫时, 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。

## Host

访问的服务器主机名, 比如百度的 [www.baidu.com](http://www.baidu.com)。这个值在爬虫时可以从 访问的 URI 中获得。

## If-Modified-Since

只有当所请求的内容在指定的日期之后又经过修改才返回它, 否则返回304。其目的是为了提高访问效率。但是在爬虫时, 不设置这个值, 而在增量爬取时才设置一个这样的值, 用以更新信息。

## Authorization

当客户端接收到来自WEB服务器的 WWW-Authenticate 响应时, 该头部来回应自己的身份验证信息给WEB服务器。主要是授权验证, 确定符合服务器的要求。这个在爬虫时按需而定。

一个典型的适用于爬虫爬取数据的伪造请求头如下所示:

```
"Proxy-Connection": "keep-alive",  
"Pragma": "no-cache",  
"Cache-Control": "no-cache",  
"User-Agent": "Mozilla/5.0 (Windows NT 6.3; WOW64) AppleWebKit/537.36 (KHTML, like  
Gecko) Chrome/52.0.2743.116 Safari/537.36",  
"Accept": "text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,/;q=0.8",  
"DNT": "1",  
"Accept-Encoding": "gzip, deflate, sdch",  
"Accept-Language": "zh-CN,zh;q=0.8,en-US;q=0.6,en;q=0.4",  
"Referer": "https://www.baidu.com/s?  
wd=%BC%96%E7%A0%81&rsv\_spt=1&rsv\_iqid=0x9fcbc99a0000b5d7&issp=1&f=8&rsv\_  
bp=1&rsv\_idx=2&ie=utf-8&rqlang=cn&tn=baiduhome\_pg&rsv\_enter=0&oq=If-None-  
Match&inputT=7282&rsv\_t=3001MIX2aUzape9perXDW%2FezcxidTWU4Bt%2FciwbikdOL"
```

QHYY98rhPyD2LDNevDKyLLg2&rsv\_pq=c4163a510000b68a&rsv\_sug3=24&rsv\_sug1=14  
&rsv\_sug7=100&rsv\_sug2=0&rsv\_sug4=7283",  
"Accept-Charset": "gb2312,gbk;q=0.7,utf-8;q=0.7,\*;q=0.7",

## 请求体 (Request Body)

POST请求体有内容，GET请求，请求体为空  
设置Request Header Content-Type

application/x-www-form-urlencoded 表单数据

multipart/form-data 表单文件上传

application/json 序列化json数据

text/xml xml数据

## 响应

### 响应状态码 (Response Status Code)



响应头，其中包含了服务器对请求的应答信息，如 Content-Type、Server、Set-Cookie 等，下面将一些常用的头信息说明如下：

- Date，标识 Response 产生的时间。
- Last-Modified，指定资源的最后修改时间。
- Content-Encoding，指定 Response 内容的编码。
- Server，包含了服务器的信息，名称，版本号等。



- Content-Type, 文档类型, 指定了返回的数据类型是什么, 如`text/html` 则代表返回 *HTML* 文档, `application/javascript` 则代表返回 JavaScript 文件, `image/jpeg` 则代表返回了图片。
- Set-Cookie, 设置Cookie, Response Headers 中的 Set-Cookie即告诉浏览器需要将此内容放在 Cookies 中, 下次请求携带 Cookies 请求。
- Expires, 指定 Response 的过期时间, 使用它可以控制代理服务器或浏览器将内容更新到缓存中, 如果再次访问时, 直接从缓存中加载, 降低服务器负载, 缩短加载时间。

## 响应体 Resposne Body

即响应体, 最重要的当属响应体内容了, 响应的正文数据都是在响应体中, 如请求一个网页, 它的响应体就是网页的HTML 代码, 请求一张图片, 它的响应体就是图片的二进制数据。所以最主要的数据都包含在响应体中了, 我们做爬虫请求网页后要解析的内容就是解析响应体。

## 网页的组成

html, css, javascript

## 网页结构和节点关系

### CSS 选择器和XPath选择器

[CSS 选择器参考手册](#)

目标	CSS 3	XPath
所有元素	*	//*
所有的P元素	p	//p
所有的p元素的子元素	p > *	//p/*
根据ID获取元素	#foo	//*[@id='foo']
根据Class获取元素	.foo	//*[contains(@class,'foo')] <sup>1</sup>
拥有某个属性的元素	*[title]	//*[@title]
所有P元素的第一个子元素	p > *:first-child	//p/*[0]
所有拥有子元素a的P元素	无法实现	//p[a]
下一个兄弟元素	p + *	//p/following-sibling::*[0]

## 会话和cookie

无状态http

## 爬虫代理

免费、付费

高度匿名代理

会将数据包原封不动转发，服务端记录的是代理服务器的ip

普通匿名代理

代理服务器通常会加入http头 HTTP\_VIA HTTP\_X\_FORWARDED\_FOR，可能能查到客户端的IP

## urllib库

urllib是基于http的高层库，它有以下三个主要功能：

1. request处理客户端的请求
2. response处理服务端的响应
3. parse会解析url
4. 主要用来识别网站的robots.txt文件，用得较少

## 获取响应信息

```
# 获取网页内容
import urllib.request
response = urllib.request.urlopen('http://www.baidu.com/')
html = response.read().decode("utf-8")
print(html)

# 取响应状态码和头信息
print(response.status)
print(response.getheaders())
print(response.getheader("Server"))
```



## 设置超时时间

```
import urllib.request
response = urllib.request.urlopen("http://2018.sina.com.cn/", timeout=1)
html = response.read().decode("utf-8")
print(html)
```

## 设置请求头和参数

```
from urllib import request, parse

url = "http://2018.sina.com.cn/"
headers = {
    "User-Agent": "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; 360SE)",
    "Host": "2018.sina.com.cn",
}
dict = {
    "name": "Question"
}
data = bytes(parse.urlencode(dict), encoding="utf8")
req = request.Request(url=url, data=data, headers=headers, method="GET")
response = request.urlopen(req)
print(response.read().decode("utf-8"))
```

## 异常处理

```
from urllib import request, error

try:
    response = request.urlopen("https://cuiqingcai.com/index.htm")
except error.URLError as e:
    print(e.reason)
```

## 用爬虫下载图片

pip install request

```
import requests

r = requests.get("http://www.baidu.com")
print(r.status_code)
print(r.text)
print(r.cookies)
```