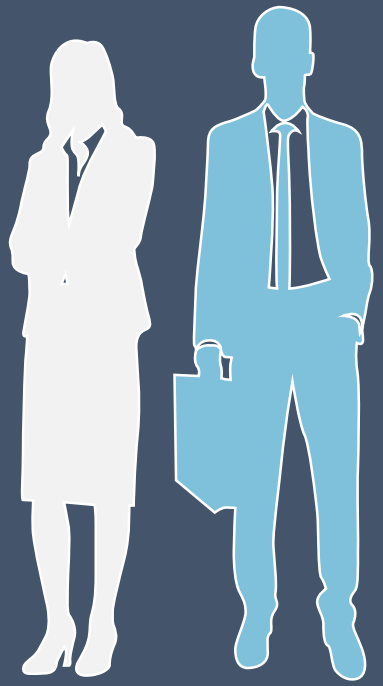
AQF Analyst of Quantitative Finance



HTTP请求方法与过程

朱映秋

CONTENTS

URI和URL

- URI的全称为Uniform Resource Identifier，统一资源标志符。
- URL的全称为Universal Resource Locator，统一资源定位符。URL是URI的一种常见形式，它指定了资源的位置以及访问该资源的方式。
- 即有这样的一个图标资源，我们用URL/URI来唯一指定了它的访问方式，这其中包括了访问协议https、访问路径（/即根目录）和资源名称favicon.ico。通过这样一个链接，我们便可以从互联网上找到这个资源，这就是URL/URI。

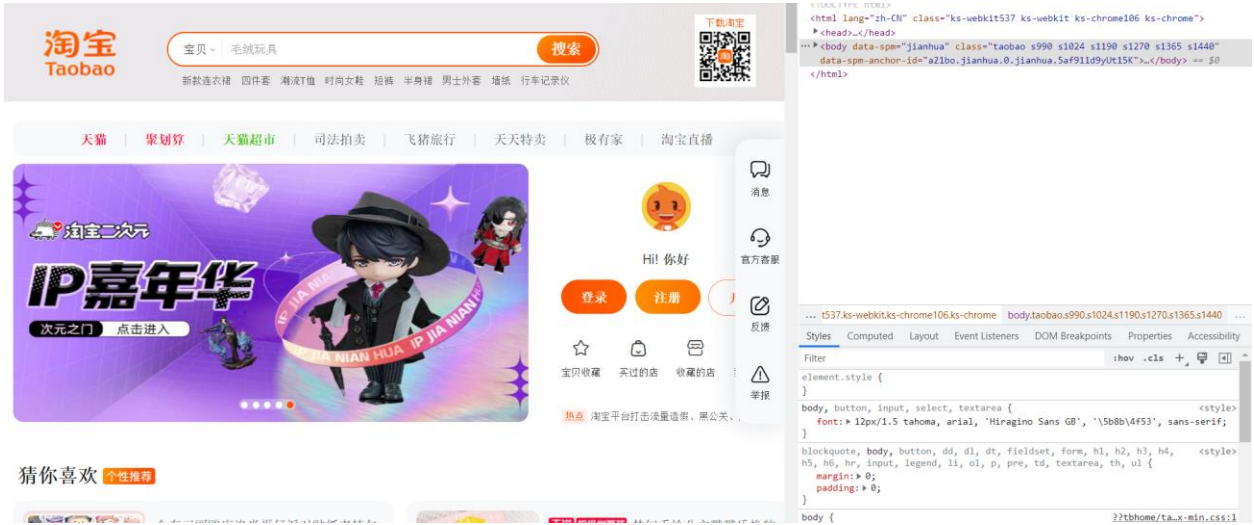
- URI还包括一个子类叫作URN，全称为Universal Resource Name，即统一资源名称。
- 它用于命名资源而不指定如何定位资源。与URL不同，URN仅通过一个唯一的名称来标识资源，而不涉及资源的位置或访问方式。
  - 例：urn:isbn:0325321627指定了某本书的ISBN，可以唯一标识这本书，但是没有指定到哪里定位这本书，这就是URN。

超文本

- 超文本英文名称叫作hypertext，是一种非线性的文本形式，我们在浏览器里看到的网页就是超文本解析而成的，其网页 源代码是一系列HTML代码，里面包含了一系列标签，比如<img> 标签用于在网页中显示图片。通过指定图片的源文件路径和其他属性，比如宽度和高度，可以将图片插入到网页中。

超文本

- 浏览器解析这些标签后，便形成了我们平常看到的网页
- 例：在Chrome浏览器里面打开任意一个页面。
  - 如淘宝首页，右击任一地方并选择“检查”项；
  - 打开浏览器的开发者工具，这时在Elements选项卡即可看到当前网页的 源代码；
  - 这些源代码就是超文本。



HTTP和HTTPS

- 在淘宝的首页https://www.taobao.com/ 中，URL的开头会有http或https，这就是访问资源需要的协议类 型。
- 有时还会看到ftp、sftp、smb开头的URL，它们都是协议类型。在爬虫中，我们抓取的页面通常就是http或 https协议的。
- HTTP
  - 全称： Hyper Text Transfer Protocol
  - 中文名：超文本传输协议
  - 作用：用于从网络传输超文本数据到本地浏览器的传送协议，能保证高效而准确地传送超文本文档。
  - HTTP由万维网协会（World Wide Web Consortium）和Internet工作小组IETF（Internet Engineering Task Force）共同合作制定的规范，目前广泛使用的是HTTP 1.1版本。

HTTP和HTTPS

- HTTPS
  - 全称：Hyper Text Transfer Protocol over Secure Socket Layer
  - 含义：以安全为目标的HTTP通道，简单讲是HTTP的安全版，即在HTTP下加入SSL层，简称为HTTPS。
- **HTTPS的安全基础是SSL，因此通过它传输的内容都是经过SSL加密的。**
- **HTTPS主要作用：**
  - 建立一个信息安全通道来保证数据传输的安全。
  - 确认网站的真实性，凡是使用了HTTPS的网站，都可以通过点击浏览器查看网站的认证信息，也 可查询其是否具有CA组织颁发的安全签名。

1. Cookie机制

- 为解决HTTP的无状态性带来的负面作用，Cookie机制应运而生。Cookie本质上是一段文本信息。当客户端请求服务器时，若服务器需要记录用户状态，就在响应用户请求时发送一段Cookie信息。
- 客户端浏览器会保存该Cookie信息，当用户再次访问该网站时，浏览器会把Cookie做为请求信息的一部分提交给服务器。
- 服务器对Cookie进行验证，以此来判断用户状态，当且仅当该Cookie合法且未过期时，用户才可直接登录网站。

2. Cookie的存储方式

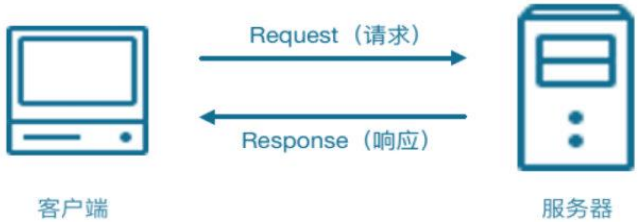
- Cookie由用户客户端浏览器进行保存，按其存储位置可分为内存式存储和硬盘式存储。
- 内存式存储：也称为会话Cookie或非持久Cookie。这种Cookie存储在客户端浏览器的内存中，随着浏览器的关闭而被删除。会话Cookie通常用于存储临时数据，例如用户在网站上的会话状态、购物车内容等。而当用户关闭浏览器时，这些Cookie将被丢弃，不会在下次访问时被重新发送。
- 硬盘式存储：也称为持久Cookie。这种Cookie会被保存在客户端浏览器的硬盘上，即使关闭了浏览器，它们仍然存在，并在下次访问相同的网站时被发送到服务器。持久Cookie具有过期时间，在到达过期时间之前，浏览器都会将其保留。持久Cookie通常用于实现记住登录状态、用户首选项、个性化设置等功能。

- 3. Cookie的实现过程
  - 客户端与服务器间的Cookie实现过程的具体步骤如下：
  - 客户端请求服务器：当客户端（浏览器）向服务器发送请求时，不会包含任何关于Cookie的信息。
  - 服务器响应请求：服务器接收到客户端的请求后，根据需要将相关的Cookie信息添加到响应中。服务器通过在响应头部中添加Set-Cookie字段来设置Cookie。Set-Cookie字段包含一个或多个Cookie的名称和值，以及其他可选的属性（例如过期时间、路径、域等）。
  - 客户端存储Cookie：客户端接收到服务器响应后，会将响应头中的Set-Cookie字段解析，并将其中的Cookie信息存储起来。浏览器会将这些Cookie保存在内存或硬盘中，具体存储位置取决于Cookie的属性（如是否是会话Cookie或持久Cookie）以及浏览器的设置。
  - 客户端再次请求服务器：当客户端再次向服务器发送请求时，会自动在请求头部中包含之前存储的Cookie信息。浏览器会将存储的Cookie添加到请求头的Cookie字段中，以便服务器可以读取并使用这些Cookie来获取客户端的状态和信息。
  - 服务器处理Cookie：服务器接收到客户端的请求后，可以通过读取请求头部中的Cookie字段来获取客户端发送的Cookie信息。服务器可以解析Cookie的名称和值，并根据这些信息来识别用户。

CONTENTS

HTTP请求过程

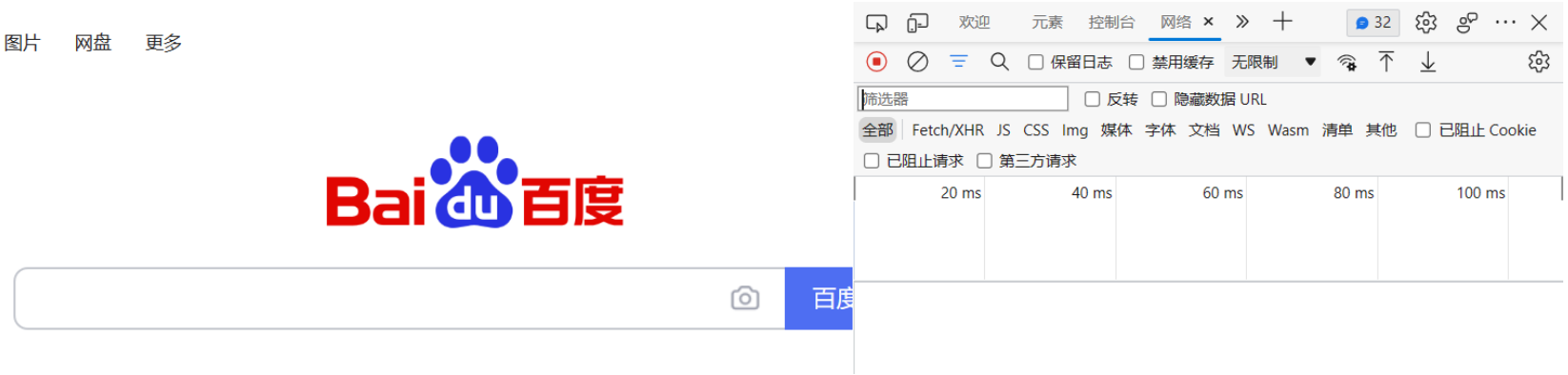
- 在浏览器中输入一个URL，浏览器便会显示页面内容。
- 实际上，这个过程是浏览器向网站所在的服务器发送请求后，服务器接收请求并进行解析和处理， 然后将对应的响应传回给浏览器。
- 响应里包含了页面的源代码等内容，浏览器会对其进行解析并显示内容。



*上图中所说的客户端指的是PC或手机浏览器，服务器指的是要访问的网站所在的服务器。

HTTP请求过程

- 打开Chrome浏览器，右击并选择“检查”项（或按快捷键F12），即可打开浏览器的开发者工具。 访问百度http://www.baidu.com/ ，输入该URL后，即可观察这个整个网络请求的过程。可以看到， 在Network标签下方出现了数行条目，每行条目代表一个发送请求和接收响应的完整过程。



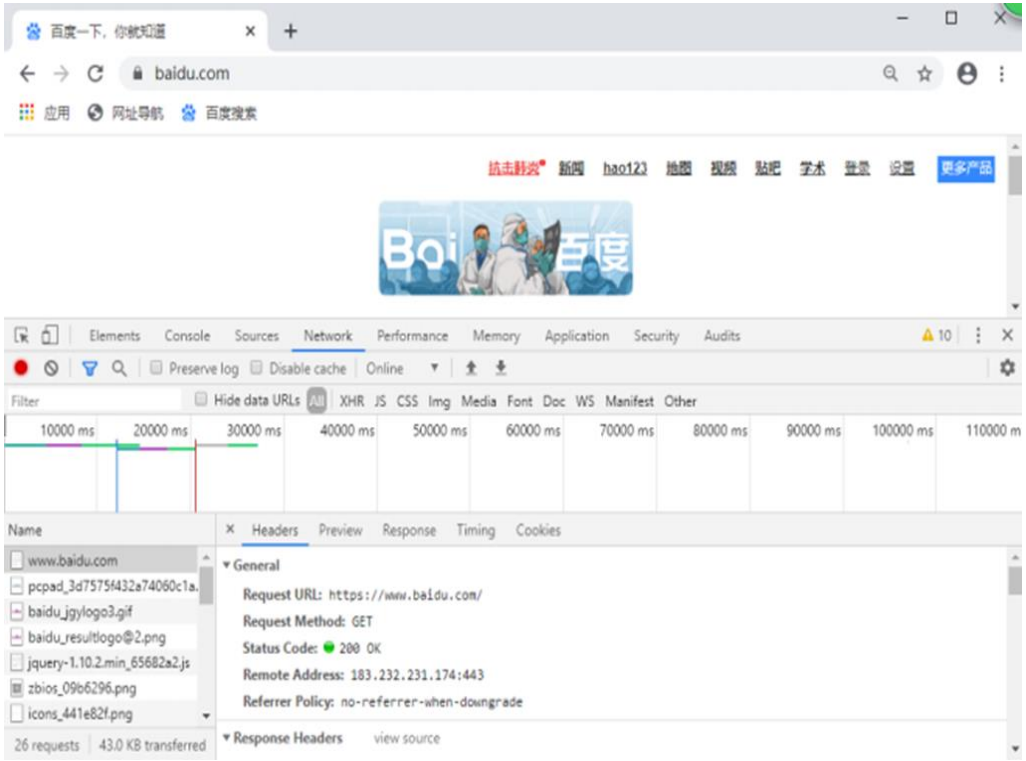
HTTP请求过程

Name	Status	Type	Initiator	Size	Time	Waterfall
www.baidu.com	200	document	Other	41.0 KB	12.03 s	
pcpad_3d7575f432a74060c1a5a0bfb71c8a2e....	200	png	www.baidu.com/:485	(memory ca...	0 ms	

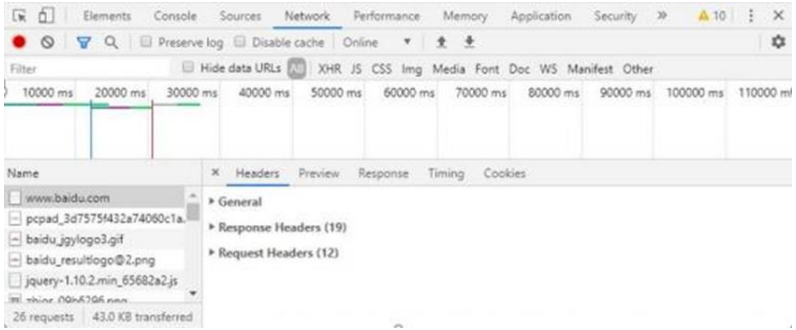
- 我们先观察第一个网络请求，即www.baidu.com。其中含义如下：
- 1）第一列Name：请求的名称，一般会将URL的最后一部分内容当作名称。
- 2）第二列Status：响应的状态码，这里显示为200，代表响应是正常的。通过状态码，我们可以判断 发送了请求之后是否得到了正常的响应。
- 3）第三列Type：请求的文档类型。这里为document，代表我们这次请求的是一个HTML文档，内容就是一些HTML代码。
- 4）第四列Initiator：请求源。用来标记请求是由哪个对象或进程发起的。
- 5）第五列Size：从服务器下载的文件和请求的资源大小。如果是从缓存中取得的资源，则该列会显示 from cache。
- 6）第六列Time：发起请求到获取响应所用的总时间。
- 7）第七列Waterfall：网络请求的可视化瀑布流。

HTTP请求过程

- **点击第一个条目，即可看到更为详细的信息：**
- 首先是General部分：
- Request URL为请求的URL，
- Request Method为请求的方法，Status Code为响应状态码，
- Remote Address为远程服务器的地址和端口，
- Referrer Policy为Referrer判别策略。



- 再继续往下看，可以看到：



➢ **有Response Headers和Request Headers，这分别代表响应头和请求头。**

- Request Headers请求头里带有许多请求信息，例如浏览器标识、Cookies、Host 等信息，这是请求的一部分，服务器会根 据请求头内的信息判断请求是否合法，进而作出相应的响应。
- Response Headers就是响应的一部分，
- 例如其中包含了服务器的类型、文档类型、日期等信息，浏览器接受到响应后会解析响应内容，并显示网页信息。

CONTENTS

请求方法

- **请求由客户端向服务端发出，常见方法：GET和POST**
- 当我们在浏览器中直接输入URL，并发起一个GET请求， 请求的参数会直接包含到URL当中。
- 例如：在百度中搜索Python，这就是一个GET请求，链接为https://www.baidu.com/s?wd=Python，其中URL中 包含了请求的参数信息，也就是wd表示要搜寻的关键字。
- POST请求大多在表单提交时发起。比如，对于一个登录表单，输入用户名和密码后，点击“登录”按 钮，这通常会发起一个POST请求，其数据通常以表单的形式传输，而不会体现在URL中。

请求方法

- GET和POST请求方法的区别：
  - GET请求中的参数包含在URL里面，数据可以在URL中看到，而POST请求的URL不会包含这些数据，数据 都是通过表单形式传输的，会包含在请求体中。
  - GET请求提交的数据最多只有1024字节，而POST方式没有限制。
  - 一般来说，登录时，需要提交用户名和密码，其中包含了敏感信息，使用GET方式请求的话，密码就会暴 露在URL里面，造成密码泄露，所以这里最好以POST方式发送。
  - 上传文件时，由于文件内容比较大，也会选用POST方式。

请求方法

- 平常遇到的绝大部分请求都是GET或POST请求，另外还有一些请求方法，如GET、HEAD、POST、PUT、DELETE、OPTIONS、CONNECT、TRACE等，我们简单将其总结为下表所示：

方法	描述
GET	请求页面，并返回内容
POST	大多用于提交表单或上传文件，数据包含在请求体中
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
PUT	从客户端向服务器传送的数据取代指定文档中的内容
DELETE	请求服务器删除指定的页面
CONNECT	把服务器当作跳板，让服务器代替客户端访问其他网页
OPTIONS	允许客户端查看服务器的性能
TRACE	回显服务器收到的请求，主要用于测试或诊断

请求的网址

- 请求的网址，即统一资源定位符URL，它可以唯一确定我们想请求的资源。

- https://www.taobao.com/

请求头

- Request Headers，用来说明服务器要使用的附加信息，比较重要的信息有Cookie、Referer、User-Agent 等。
- 请求头下一般包含的信息：
  - 1）Accept：指定客户端可以接受的响应内容类型。
  - 2）Accept-Language：指定客户端可以接受的语言类型。
  - 3）Accept-Encoding：指定客户端可以接受的内容编码方式。
  - 4）Host：指定请求资源的主机IP和端口号。
  - 5）Cookie：用于在客户端存储网站为了辨别用户进行会话跟踪而生成的数据。
  - 6）Referer：标识请求的来源页面，用于统计访问来源或进行防盗链处理等。
  - 7）User-Agent：识别客户端使用的操作系统、浏览器和版本等信息。
  - 8）Content-Type：指定请求中的媒体类型信息。
  - 以上是一些常见的请求头信息，可以根据实际需求添加或修改请求头中的其他字段。

请求体

- 请求体一般承载的内容是POST请求中的表单数据，而对于GET请求，请求体则为空。

Content-Type和POST提交数据方式的关系：

Content-Type	提交数据的方式
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

- 在爬虫中，如果要构造POST请求，需要使用正确的Content-Type，并了解各种请求库的各个参数设置时使用的是 哪种Content-Type，不然可能会导致POST提交后无法正常响应。

CONTENTS

响应状态码

- 响应由服务端返回给客户端
- 响应状态码表示服务器的响应状态，如200代表服务器正常响应，404代表页面未找到，500代表服务器内部发生错误。
- 在爬虫中，我们可以根据状态码来判断服务器响应状态，如状态码为200，则证明成功返回数据，再进 行进一步的处理，否则直接忽略。

100	继续	请求者应当继续提出请求。服务器已收到请求的一部分， 正在等待其余部分
101	切换协议	请求者已要求服务器切换协议，服务器已确认并准备切换
200	成功	服务器已成功处理了请求
201	已创建	请求成功并且服务器创建了新的资源
202	已接受	服务器已接受请求，但尚未处理
203	非授权信息	服务器已成功处理了请求，但返回的信息可能来自另一个源
204	无内容	服务器成功处理了请求，但没有返回任何内容
205	重置内容	服务器成功处理了请求，内容被重置
206	部分内容	服务器成功处理了部分请求

响应状态码

300	多种选择	针对请求，服务器可执行多种操作
301	永久移动	请求的网页已永久移动到新位置，即永久重定向
302	临时移动	请求的网页暂时跳转到其他页面，即暂时重定向
303	查看其他位置	如果原来的请求是POST，重定向目标文档应该通过GET提取
304	未修改	此次请求返回的网页未修改，继续使用上次的资源
305	使用代理	请求者应该使用代理访问该网页
307	临时重定向	请求的资源临时从其他位置响应
400	错误请求	服务器无法解析该请求
401	未授权	请求没有进行身份验证或验证未通过
403	禁止访问	服务器拒绝此请求
404	未找到	服务器找不到请求的网页
405	方法禁用	服务器禁用了请求中指定的方法
406	不接受	无法使用请求的内容响应请求的网页

响应状态码

407	需要代理授权	请求者需要使用代理授权
408	请求超时	服务器请求超时
409	冲突	服务器在完成请求时发生冲突
410	已删除	请求的资源已永久删除
411	需要有效长度	服务器不接受不含有效内容长度标头字段的请求
412	未满足前提条件	服务器未满足请求者在请求中设置的其中一个前提条件
500	服务器内部错误	服务器遇到错误，无法完成请求
501	未实现	服务器不具备完成请求的功能
502	错误网关	服务器作为网关或代理，从上游服务器收到无效响应
503	服务不可用	服务器目前无法使用
504	网关超时	服务器作为网关或代理，但是没有及时从上游服务器收到请求
505	HTTP版本不支持	服务器不支持请求中所用的HTTP协议版本

响应头

- **常用的头信息：**
  - （1）Date：标识响应产生的时间。
  - （2）Last-Modified：指定资源的最后修改时间。
  - （3）Content-Encoding：指定响应内容的编码。
  - （4）Server：包含服务器的信息，比如名称、版本号等。
  - （5）Content-Type：文档类型，指定返回的数据类型是什么，如text/html代表返回HTML文档，application/x-javascript则代表返回JavaScript文件， image/jpeg则代表返回图片。
  - （6）Set-Cookie：设置Cookies。响应头中的Set-Cookie告诉浏览器需要将此内容放在Cookies中，下 次请求携带Cookies请求。
  - （7）Expires：指定响应的过期时间，可以使代理服务器或浏览器将加载的内容更新到缓存中。如果再次 访问时，就可以直接从缓存中加载，降低服务器负载，缩短加载时间。

响应体

- 响应体是响应中最重要的内容。
- 响应的正文数据都在响应体中，比如请求网页时，它的响应体就是网页的HTML代码；请求一张图片时，它的响应体就是图片的二进制数据。做爬虫请求网页后，要解析的内容就是响应体。
- 在浏览器开发者工具中点击Response，就可以看到网页的源代码，也就是响应体的内容，它是解析的目标。
- 在做爬虫时，我们主要通过响应体得到网页的源代码、JSON数据等，然后从中做相应内容的提取。

Thank you!