

HTTP是一个客户端终端（用户）和服务器端（网站）请求和应答的标准（TCP）。通过使用网页浏览器、网络爬虫或者其它的工具，客户端发起一个HTTP请求到服务器上指定端口（默认端口为80）。我们称这个客户端为用户代理程序（user agent）。应答的服务器上存储着一些资源，比如HTML文件和图像。我们称这个应答服务器为源服务器（origin server）。在用户代理和源服务器中间可能存在多个“中间层”，比如代理服务器、网关或者隧道（tunnel）。

\2. 发送HTTP请求  
通过TCP套接字，客户端向Web服务器发送一个文本的请求报文，一个请求报文由请求行、请求头部、空行和请求数据4部分组成。

\3. 服务器接受请求并返回HTTP响应  
Web服务器解析请求，定位请求资源。服务器将资源复本写到TCP套接字，由客户端读取。一个响应由状态行、响应头部、空行和响应数据4部分组成。

\4. 释放连接TCP连接  
若connection 模式为close，则服务器主动关闭TCP连接，客户端被动关闭连接，释放TCP连接;若connection 模式为keepalive，则该连接会保持一段时间，在该时间内可以继续接收请求;

\5. 客户端浏览器解析HTML内容  
客户端浏览器首先解析状态行，查看表明请求是否成功的状态代码。然后解析每一个响应头，响应头告知以下为若干字节的HTML文档和文档的字符集。客户端浏览器读取响应数据HTML，根据HTML的语法对其进行格式化，并在浏览器窗口中显示。

例如：在浏览器地址栏键入URL，按下回车之后会经历以下流程：

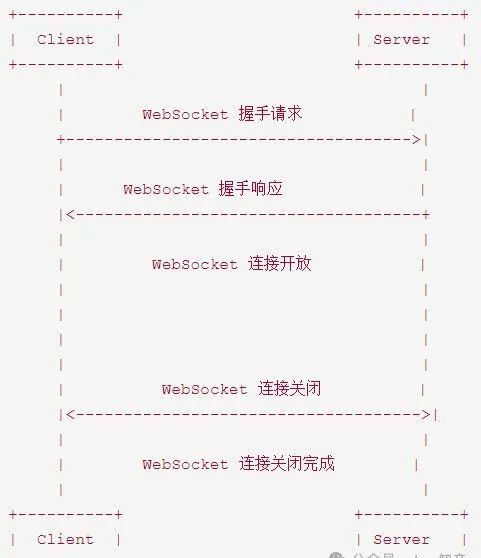
1. 浏览器向 DNS 服务器请求解析该 URL 中的域名所对应的 IP 地址;
2. 解析出 IP 地址后，根据该 IP 地址和默认端口 80，和服务器建立TCP连接;
3. 浏览器发出读取文件(URL 中域名后面部分对应的文件)的HTTP 请求，该请求报文作为 TCP 三次握手的第三个报文的数据发送给服务器;
4. 服务器对浏览器请求作出响应，并把对应的 html 文本发送给浏览器;
5. 释放 TCP连接;
6. 浏览器将该 html 文本并显示内容;

HTTPS 协议是 HyperText Transfer Protocol Secure（超文本传输安全协议）的缩写，是一种通过计算机网络进行安全通信的传输协议。

HTTP 本身是不安全的，因为传输的数据未经加密，可能会被窃听或篡改，为了解决这个问题，引入了 HTTPS，即在 HTTP 上加入 SSL/TLS 协议，为数据传输提供了加密和身份验证。

HTTPS 经由 HTTP 进行通信，但利用 SSL/TLS 来加密数据包，HTTPS 开发的主要目的，是提供对网站服务器的身份认证，保护交换资料的隐私与完整性。

socket：使用TCP/IP协议的应用程序通常采用应用[编程接口](https://zhida.zhihu.com/search?q=%E7%BC%96%E7%A8%8B%E6%8E%A5%E5%8F%A3&zhida_source=entity&is_preview=1)：UNIX BSD的套接字（socket）和UNIX System V的TLI（已经被淘汰），来实现网络进程之间的通信。就目前而言，几乎所有的应用程序都是采用socket，而现在又是网络时代，网络中进程通信是无处不在，这就是我为什么说“一切皆socket”。



websocket：用于web程序