# HTTP协议：

HTTP协议：超文本传输协议（英文：HyperText Transfer Protocol，缩写：HTTP）是一种用于分布式、协作式和超媒体信息系统的应用层协议。HTTP是万维网的数据通信的基础。

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）。

通常，由HTTP客户端发起一个请求，创建一个到服务器指定端口（默认是80端口）的TCP连接。HTTP服务器则在那个端口监听客户端的请求。一旦收到请求，服务器会向客户端返回一个状态，比如"HTTP/1.1 200 OK"，以及返回的内容，如请求的文件、错误消息、或者其它信息。

HTTP协议定义Web客户端如何从Web服务器请求Web页面，以及服务器如何把Web页面传送给客户端。HTTP协议采用了请求/响应模型。客户端向服务器发送一个请求报文，请求报文包含请求的方法、URL、协议版本、请求头部和请求数据。服务器以一个状态行作为响应，响应的内容包括协议的版本、成功或者错误代码、服务器信息、响应头部和响应数据。

**无状态：**HTTP是一种不保存状态,即无状态(stateless)协议。HTTP协议 自身不对请求和响应之间的通信状态进行保存。也就是说在HTTP这个 级别,协议对于发送过的请求或响应都不做持久化处理。HTTP/1.1虽然是无状态协议,但为了实现期望的保持状态功能, 于是引入了Cookie技术。有了Cookie再用HTTP协议通信,就可以管 理状态了。有关Cookie的详细内容稍后讲解。

**无连接：**无连接的含义是限制每次连接只处理一个请求。服务器处理完客户的请求，并收到客户的应答后，即断开连接。无连接有两种方式，早期的http协议是一个请求一个响应之后，直接就断开了，但是现在的http协议1.1版本不是直接就断开了，而是等几秒钟，如果用户在这几秒钟之内有新的请求，那么还是通过之前的连接通道来收发消息，如果过了这几秒钟用户没有发送新的请求，那么就会断开连接，减少短时间内建立连接的次数，建立连接也是耗时的，默认的好像是3秒中现在，但是这个时间是可以通过咱们后端的代码来调整的。

以下是 HTTP 请求/响应的步骤：

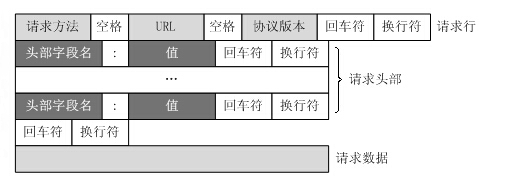
1. 客户端连接到Web服务器：一个HTTP客户端，通常是浏览器，与Web服务器的HTTP端口（默认为80）建立一个TCP套接字连接。例如，http://www.luffycity.com。
2. 发送HTTP请求：通过TCP套接字，客户端向Web服务器发送一个文本的请求报文，一个请求报文由请求行、请求头部、空行和请求数据4部分组成。
3. 服务器接受请求并返回HTTP响应：Web服务器解析请求，定位请求资源。服务器将资源复本写到TCP套接字，由客户端读取。一个响应由状态行、响应头部、空行和响应数据4部分组成。
4. 释放连接TCP连接：若connection 模式为close，则服务器主动关闭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;

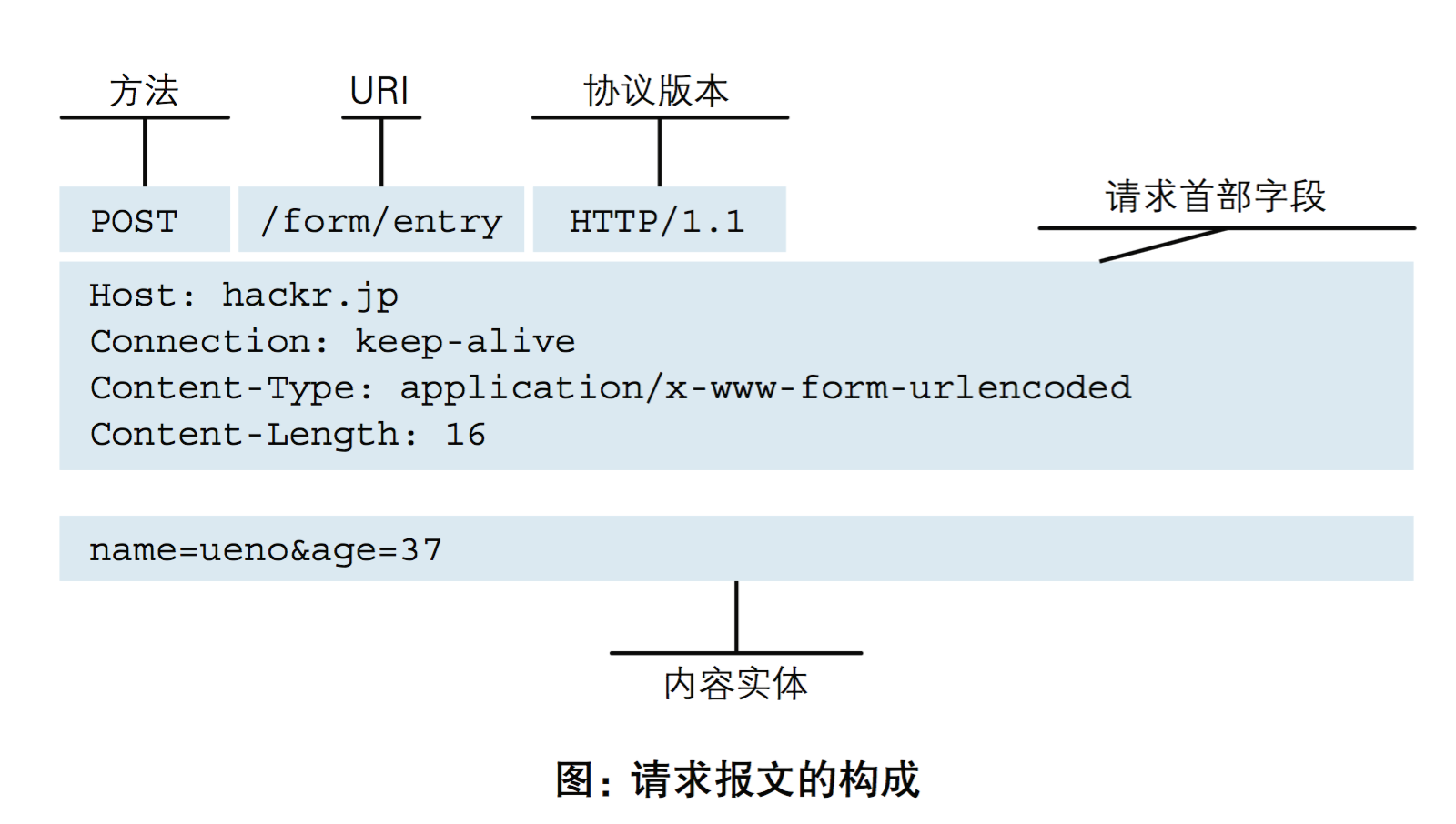
5. 客户端浏览器解析HTML内容：客户端浏览器首先解析状态行，查看表明请求是否成功的状态代码。然后解析每一个响应头，响应头告知以下为若干字节的HTML文档和文档的字符集。客户端浏览器读取响应数据HTML，根据HTML的语法对其进行格式化，并在浏览器窗口中显示。

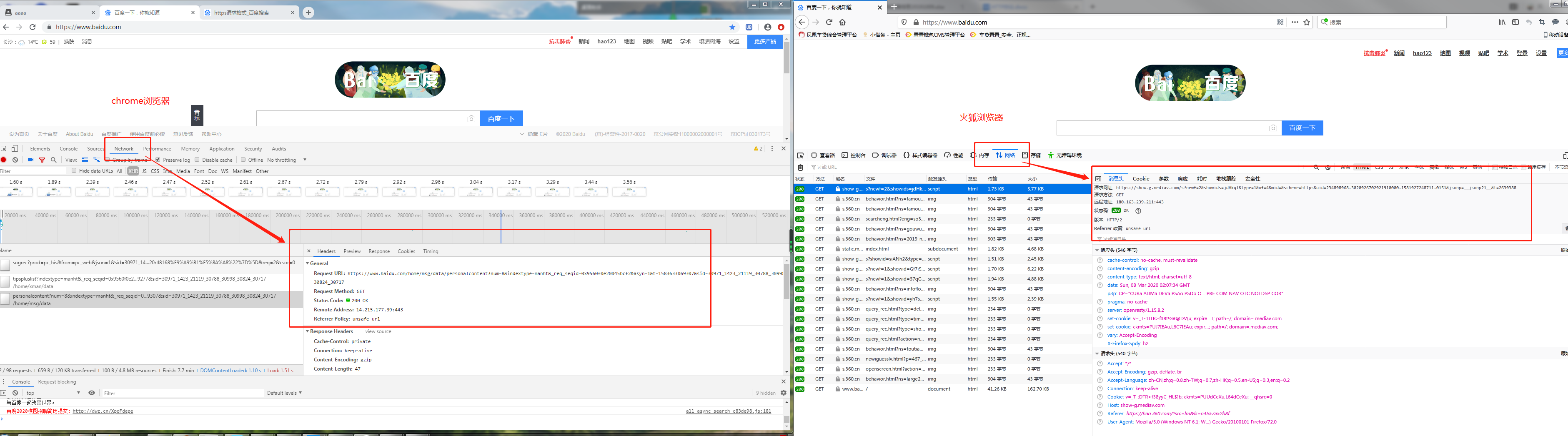
**例如：在浏览器地址栏键入URL，按下回车之后会经历以下流程：**

* 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;
* 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;
* 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;
* 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;
* 释放 TCP连接;
* 浏览器将该 html 文本并显示内容;

**HTTP请求格式(请求协议)**

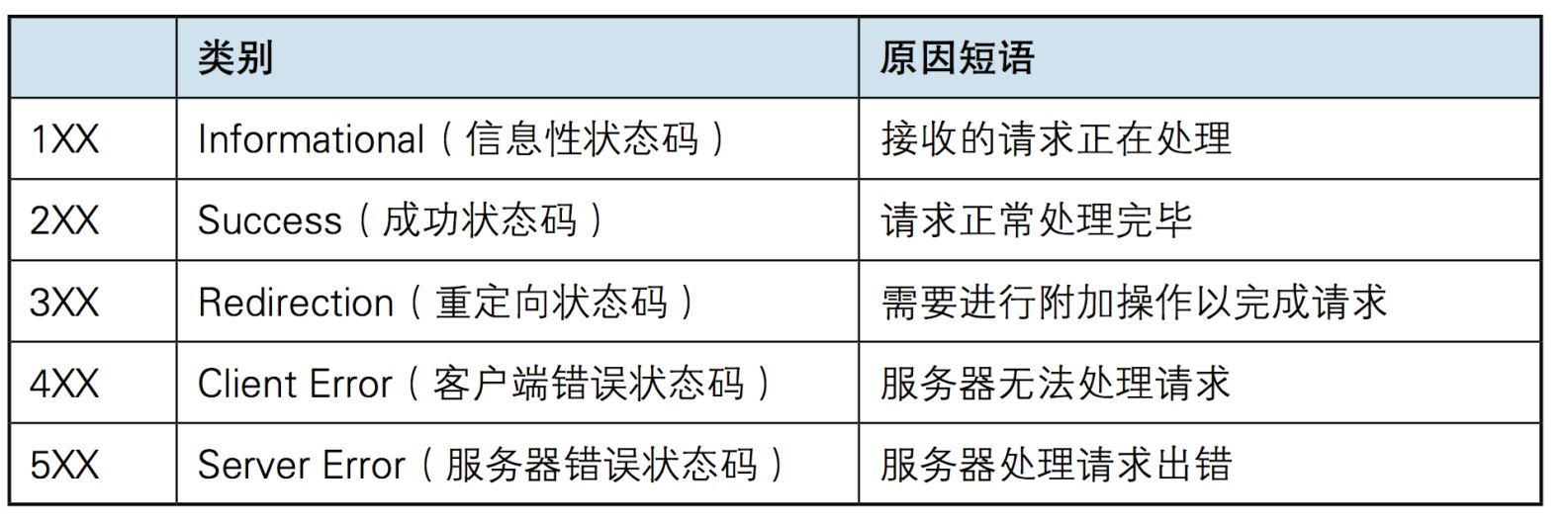






HTTP请求方法HTTP/1.1协议中共定义了八种方法（也叫“动作”）来以不同方式操作指定的资源：

1. GET：向指定的资源发出“显示”请求。使用GET方法应该只用在读取数据，而不应当被用于产生“副作用”的操作中，例如在Web Application中。其中一个原因是GET可能会被网络蜘蛛等随意访问。
2. HEAD：与GET方法一样，都是向服务器发出指定资源的请求。只不过服务器将不传回资源的本文部分。它的好处在于，使用这个方法可以在不必传输全部内容的情况下，就可以获取其中“关于该资源的信息”（元信息或称元数据）。
3. POST：向指定资源提交数据，请求服务器进行处理（例如提交表单或者上传文件）。数据被包含在请求本文中。这个请求可能会创建新的资源或修改现有资源，或二者皆有。
4. PUT：向指定资源位置上传其最新内容。
5. DELETE：请求服务器删除Request-URI所标识的资源。
6. TRACE：回显服务器收到的请求，主要用于测试或诊断。
7. OPTIONS：这个方法可使服务器传回该资源所支持的所有HTTP请求方法。用'\*'来代替资源名称，向Web服务器发送OPTIONS请求，可以测试服务器功能是否正常运作。
8. 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。通常用于SSL加密服务器的链接（经由非加密的HTTP代理服务器）。



成功

200 正常;请求已完成。

201 正常;紧接POST命令。

202 正常;已接受用于处理，但处理尚未完成。

203 正常;部分信息 — 返回的信息只是一部分。

204 正常;无响应 — 已接收请求，但不存在要回送的信息。

重定向

301 永久重定向 — 请求的数据具有新的位置且更改是永久的。

302 暂时重定向 — 请求的数据临时具有不同URI。

303 请参阅其它 — 可在另一URI下找到对请求的响应，且应使用 GET方法检索此响应。

304 未修改 — 未按预期修改文档。

305 使用代理 — 必须通过位置字段中提供的代理来访问请求的资源。

306 未使用 — 不再使用;保留此代码以便将来使用。

代码中的错误

400 错误请求 — 请求中有语法问题，或不能满足请求。

401 未授权 — 未授权客户机访问数据。

402 需要付款 — 表示计费系统已有效。

403 禁止— 即使有授权也不需要访问。

404 找不到—服务器找不到给予的资源;文档不存在。

406 不可接受 — 根据此请求中所发送的“接受”标题，此请求所标识的资源只能生成内容特征为“不可接受”的响应实体。

407 代理认证请求 — 客户机首先必须使用代理认证自身。

410 请求的网页不存在(永久);

415 介质类型不受支持 —服务器拒绝服务请求，因为不支持请求实体的格式。

500 内部错误 — 因为意外情况，服务器不能完成请求。

501 未执行 —服务器不支持请求的工具。服务器不具备完成请求的功能。例如，服务器无法识别请求方法时可能会返回此代码。

502 错误网关—服务器接收到来自上游服务器的无效响应。已经执行，但是由于某种原因而没有执行完毕，最终导致终止

503 无法获得服务 — 由于临时过载或维护，服务器无法处理请求。

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 版本不受支持） 服务器不支持请求中所用的 HTTP 协议版本。