在像 HTTP 这样的Client-Server（客户端-服务器）协议中，会话分为三个阶段：

1. 客户端建立一条 TCP 连接（如果传输层不是 TCP，也可以是其他适合的连接）。
2. 客户端发送请求并等待应答。
3. 服务器处理请求并送回应答，回应包括一个状态码和对应的数据。

从 HTTP/1.1 开始，连接在完成第三阶段后不再关闭，客户端可以再次发起新的请求。这意味着第二步和第三步可以连续进行数次。

**建立连接**[**Edit**](https://developer.mozilla.org/zh-CN/docs/Web/HTTP/Session$edit#%E5%BB%BA%E7%AB%8B%E8%BF%9E%E6%8E%A5)

在客户端-服务器协议中，连接是由客户端发起建立的。在HTTP中打开连接意味着在底层传输层启动连接，通常是 TCP。

使用 TCP 时，HTTP 服务器的默认端口号是 80，另外还有 8000 和 8080 也很常用。页面的 URL 会包含域名和端口号，但当端口号为 80 时可以省略。前往 [标识互联网上的内容](https://developer.mozilla.org/zh-CN/docs/) 获取更多内容。

**注意:** 客户端-服务器模型不允许服务器在没有显式请求时发送数据给客户端。为了解决这个问题，Web 开发者们使用了许多技术：例如，使用 [XMLHTTPRequest](https://developer.mozilla.org/zh-CN/docs/Web/API/XMLHTTPRequest) 或 [Fetch](https://developer.mozilla.org/zh-CN/docs/Web/API/Fetch) API 周期性地请求服务器，使用 HTML [WebSockets API](https://developer.mozilla.org/en/WebSockets)，或其他类似协议。

**发送客户端请求**[**Edit**](https://developer.mozilla.org/zh-CN/docs/Web/HTTP/Session$edit#%E5%8F%91%E9%80%81%E5%AE%A2%E6%88%B7%E7%AB%AF%E8%AF%B7%E6%B1%82)

一旦连接建立，用户代理就可以发送请求 (用户代理通常是 Web 浏览器，但也可以是其他的（例如爬虫）。客户端请求由一系列文本指令组成，并使用 CRLF 分隔，它们被划分为三个块：

1. 第一行包括请求方法及请求参数：
   * 文档路径，不包括协议和域名的绝对路径 URL
   * 使用的 HTTP 协议版本
2. 接下来的行每一行都表示一个 HTTP 首部，为服务器提供关于所需数据的信息（例如语言，或 MIME 类型），或是一些改变请求行为的数据（例如当数据已经被缓存，就不再应答）。这些 HTTP 首部组成以一个空行结束的一个块。
3. 最后一块是可选数据块，包含更多数据，主要被 POST 方法所使用。

请求示例

访问 developer.mozilla.org 的根页面，即 [http://developer.mozilla.org/](https://developer.mozilla.org/)，并告诉服务器用户代理倾向于该页面使用法语展示：

GET / HTTP/1.1

Host: developer.mozilla.org

Accept-Language: fr

注意最后的空行，它把首部与数据块分隔开。由于在 HTTP 首部中没有 Content-Length，数据块是空的，所以服务器可以在收到代表首部结束的空行后就开始处理请求。

例如，发送表单的结果：

POST /contact\_form.php HTTP/1.1

Host: developer.mozilla.org

Content-Length: 64

Content-Type: application/x-www-form-urlencoded

name=Joe%20User&request=Send%20me%20one%20of%20your%20catalogue

请求方法

HTTP 定义了一组 [请求方法](https://developer.mozilla.org/zh-CN/docs/Web/HTTP/Methods) 用来指定对目标资源的行为。它们一般是名词，但这些请求方法有时会被叫做 HTTP 动词。最常用的请求方法是 GET 和 POST：

* [GET](https://developer.mozilla.org/zh-CN/docs/Web/HTTP/Methods/GET) 方法请求指定的资源。GET 请求应该只被用于获取数据。
* [POST](https://developer.mozilla.org/zh-CN/docs/Web/HTTP/Methods/POST) 方法向服务器发送数据，因此会改变服务器状态。这个方法常在 [HTML 表单](https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/Forms) 中使用。

**服务器响应结构**[**Edit**](https://developer.mozilla.org/zh-CN/docs/Web/HTTP/Session$edit#%E6%9C%8D%E5%8A%A1%E5%99%A8%E5%93%8D%E5%BA%94%E7%BB%93%E6%9E%84)

当收到用户代理发送的请求后，Web 服务器就会处理它，并最终送回一个响应。与客户端请求很类似，服务器响应由一系列文本指令组成, 并使用 CRLF 分隔，它们被划分为三个不同的块：

1. 第一行是 *状态行，*包括使用的 HTTP 协议版本，状态码和一个状态描述（可读描述文本）。
2. 接下来每一行都表示一个 HTTP 首部，为客户端提供关于所发送数据的一些信息（如数据大小，使用的压缩算法，缓存指示）。与客户端请求的头部块类似，这些 HTTP 首部组成一个块，并以一个空行结束。
3. 最后一块是数据块，包含了响应的数据 （如果有的话）。

响应示例

成功的网页响应：

HTTP/1.1 200 OK

Date: Sat, 09 Oct 2010 14:28:02 GMT

Server: Apache

Last-Modified: Tue, 01 Dec 2009 20:18:22 GMT

ETag: "51142bc1-7449-479b075b2891b"

Accept-Ranges: bytes

Content-Length: 29769

Content-Type: text/html

<!DOCTYPE html... (这里是 29769 字节的网页HTML源代码)

请求资源已被永久移动的网页响应：

HTTP/1.1 301 Moved Permanently

Server: Apache/2.2.3 (Red Hat)

Content-Type: text/html; charset=iso-8859-1

Date: Sat, 09 Oct 2010 14:30:24 GMT

Location: https://developer.mozilla.org/ (目标资源的新地址, 服务器期望用户代理去访问它)

Keep-Alive: timeout=15, max=98

Accept-Ranges: bytes

Via: Moz-Cache-zlb05

Connection: Keep-Alive

X-Cache-Info: caching

X-Cache-Info: caching

Content-Length: 325 (如果用户代理无法转到新地址，就显示一个默认页面)

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">

<html><head>

<title>301 Moved Permanently</title>

</head><body>

<h1>Moved Permanently</h1>

<p>The document has moved <a href="https://developer.mozilla.org/">here</a>.</p>

<hr>

<address>Apache/2.2.3 (Red Hat) Server at developer.mozilla.org Port 80</address>

</body></html>

请求资源不存在的网页响应：

HTTP/1.1 404 Not Found

Date: Sat, 09 Oct 2010 14:33:02 GMT

Server: Apache

Last-Modified: Tue, 01 May 2007 14:24:39 GMT

ETag: "499fd34e-29ec-42f695ca96761;48fe7523cfcc1"

Accept-Ranges: bytes

Content-Length: 10732

Content-Type: text/html

<!DOCTYPE html... (包含一个站点自定义404页面, 帮助用户找到丢失的资源)

响应状态码

[HTTP 响应状态码](https://developer.mozilla.org/en-US/docs/Web/HTTP/Status) 用来表示一个 HTTP 请求是否成功完成。响应被分为 5 种类型：信息型响应，成功响应，重定向，客户端错误和服务端错误。

* [200](https://developer.mozilla.org/zh-CN/docs/Web/HTTP/Status/200): OK. 请求成功。
* [301](https://developer.mozilla.org/zh-CN/docs/Web/HTTP/Status/301): Moved Permanently. 请求资源的 URI 已被改变。
* [404](https://developer.mozilla.org/zh-CN/docs/Web/HTTP/Status/404): Not Found. 服务器无法找到请求的资源。