1. http：超文本传输协议（[英文](https://zh.wikipedia.org/wiki/%E8%8B%B1%E6%96%87)：HyperText Transfer Protocol，[缩写](https://zh.wikipedia.org/wiki/%E7%B8%AE%E5%AF%AB)：HTTP）
2. https：超文本传输安全协议

## http

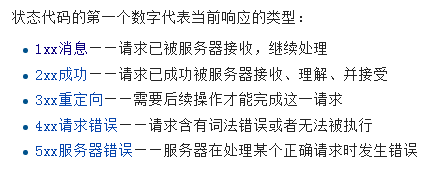
HTTP的发展是由[蒂姆·伯纳斯-李](https://zh.wikipedia.org/wiki/%E6%8F%90%E5%A7%86%C2%B7%E6%9F%8F%E7%B4%8D-%E6%9D%8E)于1989年在[欧洲核子研究组织](https://zh.wikipedia.org/wiki/%E6%AD%90%E6%B4%B2%E6%A0%B8%E5%AD%90%E7%A0%94%E7%A9%B6%E7%B5%84%E7%B9%94)（CERN）所发起。由[万维网协会](https://zh.wikipedia.org/wiki/%E5%85%A8%E7%90%83%E8%B3%87%E8%A8%8A%E7%B6%B2%E5%8D%94%E6%9C%83)（World Wide Web Consortium，W3C）和[互联网工程任务组](https://zh.wikipedia.org/wiki/%E4%BA%92%E8%81%94%E7%BD%91%E5%B7%A5%E7%A8%8B%E4%BB%BB%E5%8A%A1%E7%BB%84)（Internet Engineering Task Force，IETF）制定标准，最终发布了一系列的[RFC](https://zh.wikipedia.org/wiki/RFC)，其中最著名的是1999年6月公布的 [RFC 2616](https://tools.ietf.org/html/rfc2616)，定义了HTTP协议中现今广泛使用的一个版本——HTTP 1.1。

2014年12月，[互联网工程任务组](https://zh.wikipedia.org/wiki/%E4%BA%92%E8%81%94%E7%BD%91%E5%B7%A5%E7%A8%8B%E4%BB%BB%E5%8A%A1%E7%BB%84)（IETF）的Hypertext Transfer Protocol Bis（httpbis）工作小组将[HTTP/2](https://zh.wikipedia.org/wiki/HTTP/2)标准提议递交至[IESG](https://zh.wikipedia.org/w/index.php?title=Internet_Engineering_Steering_Group&action=edit&redlink=1)进行讨论，于2015年2月17日被批准。 HTTP/2标准于2015年5月以RFC 7540正式发表，替换HTTP 1.1成为HTTP的实现标准。

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](https://zh.wikipedia.org/wiki/TCP)）。

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：

* 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。
* 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）。
* 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。其中一个原因是GET可能会被[网络蜘蛛](https://zh.wikipedia.org/wiki/%E7%BD%91%E7%BB%9C%E8%9C%98%E8%9B%9B)等随意访问。
* 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。
* PUT：向指定资源位置上传其最新内容。
* DELETE：请求服务器删除Request-URI所标识的资源。
* TRACE：回显服务器收到的请求，主要用于测试或诊断。
* 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）。



|  |  |
| --- | --- |
| **1xx:信息** | |
| **消息** | **描述** |
| 100 Continue | 服务器仅接收到部分请求，但是一旦服务器并没有拒绝该请求，客户端应该继续发送其余的请求。 |
| 101 Switching Protocols | 服务器转换协议：服务器将遵从客户的请求转换到另外一种协议。 |

|  |  |
| --- | --- |
| **2xx:成功** | |
| **消息** | **描述** |
| 200 OK | 请求成功（其后是对GET和POST请求的应答文档。） |
| 201 Created | 请求被创建完成，同时新的资源被创建。 |
| 202 Accepted | 供处理的请求已被接受，但是处理未完成。 |
| 203 Non-authoritative Information | 文档已经正常地返回，但一些应答头可能不正确，因为使用的是文档的拷贝。 |
| 204 No Content | 没有新文档。浏览器应该继续显示原来的文档。如果用户定期地刷新页面，而Servlet可以确定用户文档足够新，这个状态代码是很有用的。 |
| 205 Reset Content | 没有新文档。但浏览器应该重置它所显示的内容。用来强制浏览器清除表单输入内容。 |
| 206 Partial Content | 客户发送了一个带有Range头的GET请求，服务器完成了它。 |

|  |  |
| --- | --- |
| **3xx:重定向** | |
| **消息** | **描述** |
| 300 Multiple Choices | 多重选择。链接列表。用户可以选择某链接到达目的地。最多允许五个地址。 |
| 301 Moved Permanently | 所请求的页面已经转移至新的url。 |
| 302 Found | 所请求的页面已经临时转移至新的url。 |
| 303 See Other | 所请求的页面可在别的url下被找到。 |
| 304 Not Modified | 未按预期修改文档。客户端有缓冲的文档并发出了一个条件性的请求（一般是提供If-Modified-Since头表示客户只想比指定日期更新的文档）。服务器告诉客户，原来缓冲的文档还可以继续使用。 |
| 305 Use Proxy | 客户请求的文档应该通过Location头所指明的代理服务器提取。 |
| 306 *Unused* | 此代码被用于前一版本。目前已不再使用，但是代码依然被保留。 |
| 307 Temporary Redirect | 被请求的页面已经临时移至新的url。 |

|  |  |
| --- | --- |
| **4xx:客户端错误** | |
| **消息** | **描述** |
| 400 Bad Request | 服务器未能理解请求。 |
| 401 Unauthorized | 被请求的页面需要用户名和密码。 |
| 401.1 | 登录失败。 |
| 401.2 | 服务器配置导致登录失败。 |
| 401.3 | 由于 ACL 对资源的限制而未获得授权。 |
| 401.4 | 筛选器授权失败。 |
| 401.5 | ISAPI/CGI 应用程序授权失败。 |
| 401.7 | 访问被 Web 服务器上的 URL 授权策略拒绝。这个错误代码为 IIS 6.0 所专用。 |
| 402 Payment Required | 此代码尚无法使用。 |
| 403 Forbidden | 对被请求页面的访问被禁止。 |
| 403.1 | 执行访问被禁止。 |
| 403.2 | 读访问被禁止。 |
| 403.3 | 写访问被禁止。 |
| 403.4 | 要求 SSL。 |
| 403.5 | 要求 SSL 128。 |
| 403.6 | IP 地址被拒绝。 |
| 403.7 | 要求客户端证书。 |
| 403.8 | 站点访问被拒绝。 |
| 403.9 | 用户数过多。 |
| 403.10 | 配置无效。 |
| 403.11 | 密码更改。 |
| 403.12 | 拒绝访问映射表。 |
| 403.13 | 客户端证书被吊销。 |
| 403.14 | 拒绝目录列表。 |
| 403.15 | 超出客户端访问许可。 |
| 403.16 | 客户端证书不受信任或无效。 |
| 403.17 | 客户端证书已过期或尚未生效。 |
| 403.18 | 在当前的应用程序池中不能执行所请求的 URL。这个错误代码为 IIS 6.0 所专用。 |
| 403.19 | 不能为这个应用程序池中的客户端执行 CGI。这个错误代码为 IIS 6.0 所专用。 |
| 403.20 | Passport 登录失败。这个错误代码为 IIS 6.0 所专用。 |
| 404 Not Found | 服务器无法找到被请求的页面。 |
| 404.0 | （无）–没有找到文件或目录。 |
| 404.1 | 无法在所请求的端口上访问 Web 站点。 |
| 404.2 | Web 服务扩展锁定策略阻止本请求。 |
| 404.3 | MIME 映射策略阻止本请求。 |
| 405 Method Not Allowed | 请求中指定的方法不被允许。 |
| 406 Not Acceptable | 服务器生成的响应无法被客户端所接受。 |
| 407 Proxy Authentication Required | 用户必须首先使用代理服务器进行验证，这样请求才会被处理。 |
| 408 Request Timeout | 请求超出了服务器的等待时间。 |
| 409 Conflict | 由于冲突，请求无法被完成。 |
| 410 Gone | 被请求的页面不可用。 |
| 411 Length Required | "Content-Length" 未被定义。如果无此内容，服务器不会接受请求。 |
| 412 Precondition Failed | 请求中的前提条件被服务器评估为失败。 |
| 413 Request Entity Too Large | 由于所请求的实体的太大，服务器不会接受请求。 |
| 414 Request-url Too Long | 由于url太长，服务器不会接受请求。当post请求被转换为带有很长的查询信息的get请求时，就会发生这种情况。 |
| 415 Unsupported Media Type | 由于媒介类型不被支持，服务器不会接受请求。 |
| 416 Requested Range Not Satisfiable | 服务器不能满足客户在请求中指定的Range头。 |
| 417 Expectation Failed | 执行失败。 |
| 423 | 锁定的错误。 |

|  |  |
| --- | --- |
| **5xx:服务器错误** | |
| **消息** | **描述** |
| 500 Internal Server Error | 请求未完成。服务器遇到不可预知的情况。 |
| 500.12 | 应用程序正忙于在 Web 服务器上重新启动。 |
| 500.13 | Web 服务器太忙。 |
| 500.15 | 不允许直接请求 Global.asa。 |
| 500.16 | UNC 授权凭据不正确。这个错误代码为 IIS 6.0 所专用。 |
| 500.18 | URL 授权存储不能打开。这个错误代码为 IIS 6.0 所专用。 |
| 500.100 | 内部 ASP 错误。 |
| 501 Not Implemented | 请求未完成。服务器不支持所请求的功能。 |
| 502 Bad Gateway | 请求未完成。服务器从上游服务器收到一个无效的响应。 |
| 502.1 | CGI 应用程序超时。　· |
| 502.2 | CGI 应用程序出错。 |
| 503 Service Unavailable | 请求未完成。服务器临时过载或当机。 |
| 504 Gateway Timeout | 网关超时。 |
| 505 HTTP Version Not Supported | 服务器不支持请求中指明的HTTP协议版本。 |

在HTTP1.0，单一TCP连接内仅执行一个“客户端发送请求—服务器发送应答”周期，之后释放TCP连接。在HTTP1.1优化支持持续活跃连接：客户端连续多次发送请求、接收应答；批量多请求时，同一TCP连接在活跃（Keep-Live）间期内复用，避免重复TCP初始握手活动，减少网络负荷和响应周期。此外支持应答到达前继续发送请求（通常是两个），称为“流线化”（stream）。

## HTTPS

超文本传输安全协议（英语：Hypertext Transfer Protocol Secure，[缩写](https://zh.wikipedia.org/wiki/%E7%B8%AE%E5%AF%AB)：HTTPS，也被称为HTTP over TLS，HTTP over SSL或HTTP Secure）是一种[网络安全](https://zh.wikipedia.org/wiki/%E7%B6%B2%E7%B5%A1%E5%AE%89%E5%85%A8)传输协议。

与[HTTP](https://zh.wikipedia.org/wiki/HTTP)的[URL](https://zh.wikipedia.org/wiki/%E7%BB%9F%E4%B8%80%E8%B5%84%E6%BA%90%E5%AE%9A%E4%BD%8D%E7%AC%A6)由“http://”起始且默认使用[端口](https://zh.wikipedia.org/wiki/TCP/UDP%E7%AB%AF%E5%8F%A3%E5%88%97%E8%A1%A8)80不同，HTTPS的URL由“https://”起始且默认使用[端口](https://zh.wikipedia.org/wiki/TCP/UDP%E7%AB%AF%E5%8F%A3%E5%88%97%E8%A1%A8)443。

HTTP协议和安全协议同属于应用层（[OSI模型](https://zh.wikipedia.org/wiki/OSI%E6%A8%A1%E5%9E%8B)的最高层），具体来讲，安全协议工作在HTTP之下，运输层之上：安全协议向运行HTTP的进程提供一个类似于TCP的套接字，供进程向其中注入报文，安全协议将报文加密并注入运输层套接字；或是从运输层获取加密报文，解密后交给对应的进程。严格地讲，HTTPS并不是一个单独的协议，而是对工作在一加密连接（[TLS](https://zh.wikipedia.org/wiki/%E4%BC%A0%E8%BE%93%E5%B1%82%E5%AE%89%E5%85%A8)或SSL）上的常规HTTP协议的称呼。

HTTPS报文中的任何东西都被加密，包括所有报头和荷载。除了可能的[选择密文攻击](https://zh.wikipedia.org/wiki/%E9%80%89%E6%8B%A9%E5%AF%86%E6%96%87%E6%94%BB%E5%87%BB)（参见[限制](https://zh.wikipedia.org/wiki/%E8%B6%85%E6%96%87%E6%9C%AC%E4%BC%A0%E8%BE%93%E5%AE%89%E5%85%A8%E5%8D%8F%E8%AE%AE#.E9.99.90.E5.88.B6)小节）之外，一个攻击者所能知道的只有在两者之间有一连接这一事实。