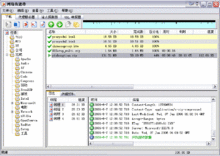
http

[超文本传输协议](https://baike.baidu.com/item/%E8%B6%85%E6%96%87%E6%9C%AC%E4%BC%A0%E8%BE%93%E5%8D%8F%E8%AE%AE)（HTTP，HyperText Transfer Protocol)是[互联网](https://baike.baidu.com/item/%E4%BA%92%E8%81%94%E7%BD%91)上应用最为广泛的一种[网络协议](https://baike.baidu.com/item/%E7%BD%91%E7%BB%9C%E5%8D%8F%E8%AE%AE)。所有的[WWW](https://baike.baidu.com/item/WWW)文件都必须遵守这个标准。设计HTTP最初的目的是为了提供一种发布和接收[HTML](https://baike.baidu.com/item/HTML)页面的方法。1960年美国人[Ted Nelson](https://baike.baidu.com/item/Ted%20Nelson)构思了一种通过[计算机](https://baike.baidu.com/item/%E8%AE%A1%E7%AE%97%E6%9C%BA)处理文本信息的方法，并称之为超文本（hypertext）,这成为了HTTP超文本传输协议标准架构的发展根基。Ted Nelson组织协调万维网协会（World Wide Web Consortium）和[互联网工程工作小组](https://baike.baidu.com/item/%E4%BA%92%E8%81%94%E7%BD%91%E5%B7%A5%E7%A8%8B%E5%B7%A5%E4%BD%9C%E5%B0%8F%E7%BB%84)（Internet Engineering Task Force ）共同合作研究，最终发布了一系列的[RFC](https://baike.baidu.com/item/RFC)，其中著名的RFC 2616定义了HTTP 1.1。

技术架构

HTTP是一个[客户端](https://baike.baidu.com/item/%E5%AE%A2%E6%88%B7%E7%AB%AF)和[服务器](https://baike.baidu.com/item/%E6%9C%8D%E5%8A%A1%E5%99%A8)端请求和应答的标准（TCP）。客户端是终端用户，服务器端是网站。通过使用[Web浏览器](https://baike.baidu.com/item/Web%E6%B5%8F%E8%A7%88%E5%99%A8)、[网络爬虫](https://baike.baidu.com/item/%E7%BD%91%E7%BB%9C%E7%88%AC%E8%99%AB)或者其它的工具，客户端发起一个到服务器上指定端口（默认[端口](https://baike.baidu.com/item/%E7%AB%AF%E5%8F%A3)为80）的HTTP请求。（我们称这个客户端）叫用户代理（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

[](https://baike.baidu.com/pic/http/243074/0/38403f3f7f4ad9a854e723ae?fr=lemma&ct=single)http和其他几种网络协议

多个中间层，比如代理，网关，或者隧道（tunnels）。尽管[TCP/IP协议](https://baike.baidu.com/item/TCP%2FIP%E5%8D%8F%E8%AE%AE)是互联网上最流行的应用，HTTP协议并没有规定必须使用它和（基于）它支持的层。 事实上，HTTP可以在任何其他互联网协议上，或者在其他网络上实现。HTTP只假定（其下层协议提供）可靠的传输，任何能够提供这种保证的协议都可以被其使用。

通常，由HTTP客户端发起一个请求，建立一个到服务器指定端口（默认是[80端口](https://baike.baidu.com/item/80%E7%AB%AF%E5%8F%A3)）的TCP连接。HTTP服务器则在那个端口监听客户端发送过来的请求。一旦收到请求，服务器（向客户端）发回一个状态行，比如"HTTP/1.1 200 OK"，和（响应的）消息，消息的消息体可能是请求的文件、错误消息、或者其它一些信息。

[HTTP协议的网页](https://baike.baidu.com/pic/http/243074/0/caae68094a44f3922eddd494?fr=lemma&ct=single)HTTP协议的网页

HTTP使用TCP而不是UDP的原因在于（打开）一个网页必须传送很多数据，而TCP协议提供传输控制，按顺序组织数据，和错误纠正。

通过HTTP或者HTTPS协议请求的资源由统一资源标示符（Uniform Resource Identifiers）（或者，更准确一些，URLs）来标识。

协议功能

HTTP协议（HyperText Transfer Protocol，超文本传输协议）是用于从WWW服务器传输超文本到本地浏览器的传输协议。它可以使浏览器更加高效，使网络传输减少。它不仅保证计算机正确快速地传输超文本文档，还确定传输文档中的哪一部分，以及哪部分内容首先显示(如文本先于图形)等。

HTTP是客户端浏览器或其他程序与[Web服务](https://baike.baidu.com/item/Web%E6%9C%8D%E5%8A%A1)器之间的应用层通信协议。在Internet上的Web服务器上存放的都是超文本信息，客户机需要通过HTTP协议传输所要访问的超文本信息。HTTP包含命令和传输信息，不仅可用于Web访问，也可以用于其他因特网/内联网应用系统之间的通信，从而实现各类应用资源超媒体访问的集成。

我们在浏览器的地址栏里输入的网站地址叫做URL (Uniform Resource Locator，[统一资源定位符](https://baike.baidu.com/item/%E7%BB%9F%E4%B8%80%E8%B5%84%E6%BA%90%E5%AE%9A%E4%BD%8D%E7%AC%A6))。就像每家每户都有一个门牌地址一样，每个网页也都有一个Internet地址。当你在

[](https://baike.baidu.com/pic/http/243074/0/4bd1e8038b9d50463912bbe0?fr=lemma&ct=single)http功用

浏览器的地址框中输入一个URL或是单击一个[超级链接](https://baike.baidu.com/item/%E8%B6%85%E7%BA%A7%E9%93%BE%E6%8E%A5)时，URL就确定了要浏览的地址。浏览器通过超文本传输协议(HTTP)，将Web服务器上站点的网页代码提取出来，并翻译成漂亮的网页。

协议基础

HTTP（HyperText Transport Protocol）是超文本传输协议的缩写，它用于传送WWW方式的数据，关于HTTP协议的详细内容请参考RFC2616。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，请求头包含请求的方法、URL、协议版本、以及包含请求修饰符、客户信息和内容的类似于MIME的消息结构。服务器以一个状态行作为响应，响应的内容包括消息协议的版本，成功或者错误编码加上包含服务器信息、实体元信息以及可能的实体内容。

通常HTTP消息包括客户机向服务器的请求消息和服务器向客户机的响应消息。这两种类型的消息由一个起始行，一个或者多个头域，一个指示头域结束的空行和可选的消息体组成。HTTP的头域包括通用头，请求头，响应头和实体头四个部分。每个头域由一个域名，冒号（:）和域值三部分组成。域名是大小写无关的，域值前可以添加任何数量的空格符，头域可以被扩展为多行，在每行开始处，使用至少一个空格或制表符。

通用头域

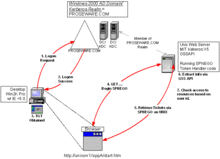
通用头域包含请求和响应消息都支持的头域，通用头域包含Cache-Control、Connection、Date、Pragma、Transfer-Encoding、Upgrade、Via。对通用头域的扩展要求通讯双方都支持此扩展，如果存在不支持的通用头域，一般将会作为实体头域处理。下面简单介绍几个在UPnP消息中使用的通用头域：

**1.Cache-Control头域**

Cache-Control指定请求和响应遵循的缓存机制。在请求消息或响应消息中设置Cache-Control并不会修改另一个消息处理过程中的缓存处理过程。请求时的缓存指令包括no-cache、no-store、max-age、max-stale、min-fresh、only-if-cached，响应消息中的指令包括public、private、no-cache、no-store、no-transform、must-revalidate、proxy-revalidate、max-age。各个消息中的指令含义如下：

Public指示响应可被任何缓存区缓存。

Private指示对于单个用户的整个或部分响应消息，不能被共享缓存处理。这允许服务器仅仅描述当用户

[](https://baike.baidu.com/pic/http/243074/0/ea85a94504e72062cffca3b0?fr=lemma&ct=single)http结构

的部分响应消息，此响应消息对于其他用户的请求无效。

no-cache指示请求或响应消息不能缓存

no-store用于防止重要的信息被无意的发布。在请求消息中发送将使得请求和响应消息都不使用缓存。

max-age指示客户机可以接收生存期不大于指定时间（以秒为单位）的响应。

min-fresh指示客户机可以接收响应时间小于当前时间加上指定时间的响应。

max-stale指示客户机可以接收超出超时期间的响应消息。如果指定max-stale消息的值，那么客户机可以接收超出超时期指定值之内的响应消息。

**HTTP Keep-Alive**

Keep-Alive功能使客户端到服务器端的连接持续有效，当出现对服务器的后继请求时，Keep-Alive功能避免了建立或者重新建立连接。市场上的大部分Web服务器，包括iPlanet、IIS和Apache，都支持HTTP Keep-Alive。对于提供静态内容的网站来说，这个功能通常很有用。但是，对于负担较重的网站来说，这里存在另外一个问题：虽然为客户保留打开的连接有一定的好处，但它同样影响了性能，因为在处理暂停期间，本来可以释放的资源仍旧被占用。当Web服务器和应用服务器在同一台机器上运行时，Keep- Alive功能对资源利用的影响尤其突出。

KeepAliveTime 值控制 TCP/IP 尝试验证空闲连接是否完好的频率。如果这段时间内没有活动，则会发送保持活动信号。如果网络工作正常，而且接收方是活动的，它就会响应。如果需要对丢失接收方敏感，换句话说，需要更快地发现丢失了接收方，请考虑减小这个值。如果长期不活动的空闲连接出现次数较多，而丢失接收方的情况出现较少，您可能会要提高该值以减少开销。缺省情况下，如果空闲连接 7200000 毫秒（2 小时）内没有活动，Windows 就发送保持活动的消息。通常，1800000 毫秒是首选值，从而一半的已关闭连接会在 30 分钟内被检测到。 KeepAliveInterval 值定义了如果未从接收方收到保持活动消息的响应，TCP/IP 重复发送保持活动信号的频率。当连续发送保持活动信号、但未收到响应的次数超出 TcpMaxDataRetransmissions 的值时，会放弃该连接。如果期望较长的响应时间，您可能需要提高该值以减少开销。如果需要减少花在验证接收方是否已丢失上的时间，请考虑减小该值或 TcpMaxDataRetransmissions 值。缺省情况下，在未收到响应而重新发送保持活动的消息之前，Windows 会等待 1000 毫秒（1 秒）。 KeepAliveTime 根据你的需要设置就行，比如10分钟，注意要转换成MS。 XXX代表这个间隔值得大小。

**2.Date头域**

Date头域表示消息发送的时间，时间的描述格式由rfc822定义。例如，Date:Mon,31Dec200104:25:57GMT。Date描述的时间表示世界标准时，换算成本地时间，需要知道用户所在的时区。

**3.Pragma头域**

Pragma头域用来包含实现特定的指令，最常用的是Pragma:no-cache。在HTTP/1.1协议中，它的含义和Cache-Control:no-cache相同。

请求消息

请求消息的第一行为下面的格式：

MethodSPRequest-URISPHTTP-VersionCRLFMethod表示对于Request-URI完成的方法，这个字段是大小写敏感的，包括OPTIONS、GET、HEAD、POST、PUT、DELETE、TRACE。方法GET和HEAD应该被所有的通用[WEB服务器](https://baike.baidu.com/item/WEB%E6%9C%8D%E5%8A%A1%E5%99%A8)支持，其他所有方法的实现是可选的。GET方法取回由Request-URI标识的信息。HEAD方法也是取回由Request-URI标识的信息，只是可以在响应时，不返回消息体。POST方法可以请求服务器接收包含在请求中的实体信息，可以用于提交[表单](https://baike.baidu.com/item/%E8%A1%A8%E5%8D%95)，向新闻组、BBS、邮件群组和数据库发送消息。

SP表示空格。Request-URI遵循URI格式，在此字段为星号（\*）时，说明请求并不用于某个特定的资源地址，而是用于服务器本身。HTTP-Version表示支持的HTTP版本，例如为HTTP/1.1。CRLF表示换行回车符。请求头域允许客户端向服务器传递关于请求或者关于客户机的附加信

[](https://baike.baidu.com/pic/http/243074/0/35da1d3b13a630db15cecbe2?fr=lemma&ct=single)http架构

息。请求头域可能包含下列字段Accept、Accept-Charset、Accept-Encoding、Accept-Language、Authorization、From、Host、If-Modified-Since、If-Match、If-None-Match、If-Range、If-Range、If-Unmodified-Since、Max-Forwards、Proxy-Authorization、Range、Referer、User-Agent。对请求头域的扩展要求通讯双方都支持，如果存在不支持的请求头域，一般将会作为实体头域处理。

典型的请求消息：

Host: download.\*\*\*\*\*\*\*.de

Accept: \*/\*

Pragma: no-cache

Cache-Control: no-cache

User-Agent: Mozilla/4.04[en](Win95;I;Nav)

Range: bytes=554554-

上例第一行表示HTTP客户端（可能是浏览器、下载程序）通过GET方法获得指定URL下的文件。棕色的部分表示请求头域的信息，绿色的部分表示通用头部分。

**1.Host头域**

Host头域指定请求资源的Intenet主机和[端口号](https://baike.baidu.com/item/%E7%AB%AF%E5%8F%A3%E5%8F%B7)，必须表示请求url的原始服务器或网关的位置。HTTP/1.1请求必须包含[主机头](https://baike.baidu.com/item/%E4%B8%BB%E6%9C%BA%E5%A4%B4)域，否则系统会以400状态码返回。

**2.Referer头域**

Referer头域允许客户端指定请求uri的源资源地址，这可以允许服务器生成回退链表，可用来登陆、优化cache等。他也允许废除的或错误的连接由于维护的目的被追踪。如果请求的uri没有自己的uri地址，Referer不能被发送。如果指定的是部分uri地址，则此地址应该是一个[相对地址](https://baike.baidu.com/item/%E7%9B%B8%E5%AF%B9%E5%9C%B0%E5%9D%80)。

**3.Range头域**

Range头域可以请求实体的一个或者多个子范围。例如，

表示头500个字节：bytes=0-499

表示第二个500字节：bytes=500-999

表示最后500个字节：bytes=-500

表示500字节以后的范围：bytes=500-

第一个和最后一个字节：bytes=0-0,-1

同时指定几个范围：bytes=500-600,601-999

但是服务器可以忽略此请求头，如果无条件GET包含Range请求头，响应会以状态码206（PartialContent）返回而不是以200（OK）。

**4.User-Agent头域**

User-Agent头域的内容包含发出请求的用户信息。

响应消息

响应消息的第一行为下面的格式：

HTTP-VersionSPStatus-CodeSPReason-PhraseCRLF

HTTP-Version表示支持的HTTP版本，例如为HTTP/1.1。Status-Code是一个三个数字的结果代码。Reason-Phrase给Status-Code提供一个简单的文本描述。Status-Code主要用于机器自动识别，Reason-Phrase主要用于帮助用户理解。Status-Code的第一个数字定义响应的类别，后两个数字没有分类的作用。第一个数字可能取5个不同的值：

1xx:信息响应类，表示接收到请求并且继续处理

2xx:处理成功响应类，表示动作被成功接收、理解和接受

3xx:[重定向](https://baike.baidu.com/item/%E9%87%8D%E5%AE%9A%E5%90%91)响应类，为了完成指定的动作，必须接受进一步处理

4xx:客户端错误，客户请求包含语法错误或者是不能正确执行

5xx:服务端错误，服务器不能正确执行一个正确的请求

响应头域允许服务器传递不能放在状态行的附加信息，这些域主要描述服务器的信息和Request-URI进一步的信息。响应头域包含Age、Location、Proxy-Authenticate、Public、Retry-After、Server、Vary、Warning、WWW-Authenticate。对响应头域的扩展要求通讯双方都支持，如果存在不支持的响应头域，一般将会作为实体头域处理。

典型的响应消息：

HTTP/1.0200OK

Date:Mon,31Dec200104:25:57GMT

Server:Apache/1.3.14(Unix)

Content-type:text/html

Last-modified:Tue,17Apr200106:46:28GMT

Etag:"a030f020ac7c01:1e9f"

Content-length:39725426

Content-range:bytes55\*\*\*\*\*\*/40279980

上例第一行表示HTTP服务端响应一个GET方法。棕色的部分表示响应头域的信息，绿色的部分表示通用头部分，红色的部分表示实体头域的信息。

**1.Location响应头**

Location响应头用于重定向接收者到一个新URI地址。

**2.Server响应头**

Server响应头包含处理请求的原始服务器的软件信息。此域能包含多个产品标识和注释，产品标识一般按照重要性排序。

实体信息

请求消息和响应消息都可以包含实体信息，实体信息一般由实体头域和实体组成。实体头域包含关于实体的原信息，实体头包括Allow、Content-Base、Content-Encoding、Content-Language、Content-Length、Content-Location、Content-MD5、Content-Range、Content-Type、Etag、Expires、Last-Modified、extension-header。extension-header允许客户端定义新的实体头，但是这些域可能无法被接受方识别。实体可以是一个经过编码的字节流，它的编码方式由Content-Encoding或Content-Type定义，它的长度由Content-Length或Content-Range定义。

**1.Content-Type实体头**

Content-Type实体头用于向接收方指示实体的介质类型，指定HEAD方法送到接收方的实体介质类型，或GET方法发送的请求介质类型

**2.Content-Range实体头**

Content-Range实体头用于指定整个实体中的一部分的插入位置，他也指示了整个实体的长度。在服务器向客户返回一个部分响应，它必须描述响应覆盖的范围和整个实体长度。一般格式：

Content-Range:bytes-unitSPfirst-byte-pos-last-byte-pos/entity-legth

例如，传送头500个字节次字段的形式：Content-Range:bytes0-499/1234如果一个http消息包含此节（例如，对范围请求的响应或对一系列范围的重叠请求），Content-Range表示传送的范围，Content-Length表示实际传送的字节数。

**3.Last-modified实体头**

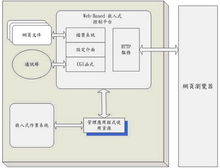
Last-modified实体头指定服务器上保存内容的最后修订时间。

例如，传送头500个字节次字段的形式：Content-Range:bytes0-499/1234如果一个http消息包含此节（例如，对范围请求的响应或对一系列范围的重叠请求），Content-Range表示传送的范围，Content-Length表示实际传送的字节数。

运作方式

在WWW中，“客户”与“服务器”是一个相对的概念，只存在于一个特定的连接期间，即在某个连接中的客户在另一个连接中可能作为服务器。基于HTTP协议的客户/服务器模式的信息交换过程，它分四个过程：建立连接、发送请求信息、发送响应信息、关闭连接。

HTTP协议是基于请求/响应[范式](https://baike.baidu.com/item/%E8%8C%83%E5%BC%8F)的。一个客户机与[服务器](https://baike.baidu.com/item/%E6%9C%8D%E5%8A%A1%E5%99%A8)建立连接后，发送一个请求给服务器，请求方式的格式为，统一资源标识符、协议版本号，后边是MIME信息包括请求修饰符、客户机信息和可能的内容。服务器接到请求后，给予相应的响应信息，其格式为一个状态行包括信息的协议版本号、一个成功或错误的代码，后边是[MIME](https://baike.baidu.com/item/MIME)信息包括服务器信息、实体信息和可能的内容。

[](https://baike.baidu.com/pic/http/243074/0/5202e5f21f483157b07ec5bb?fr=lemma&ct=single)http运作方式的一种

其实简单说就是任何服务器除了包括HTML文件以外，还有一个HTTP驻留程序，用于响应用户请求。你的浏览器是HTTP客户，向服务器发送请求，当浏览器中输入了一个开始文件或点击了一个超级链接时，浏览器就向服务器发送了HTTP请求，此请求被送往由[IP地址](https://baike.baidu.com/item/IP%E5%9C%B0%E5%9D%80)指定的URL。驻留程序接收到请求，在进行必要的操作后回送所要求的文件。在这一过程中，在网络上发送和接收的数据已经被分成一个或多个数据包（packet），每个数据包包括：要传送的数据；控制信息，即告诉网络怎样处理数据包。TCP/IP决定了每个数据包的格式。如果事先不告诉你，你可能不会知道信息被分成用于传输和再重新组合起来的许多小块。

许多HTTP通讯是由一个用户代理初始化的并且包括一个申请在源服务器上资源的请求。最简单的情况可能是在用户代理(UA)和源服务器(O)之间通过一个单独的连接来完成。

当一个或多个中介出现在请求/响应链中时，情况就变得复杂一些。中介有三种：代理(Proxy)、[网关](https://baike.baidu.com/item/%E7%BD%91%E5%85%B3)(Gateway)和通道(Tunnel)。一个代理根据URI的绝对格式来接受请求，重写全部或部分消息，通过URI的标识把已格式化过的请求发送到服务器。网关是一个接收代理，作为一些其它服务器的上层，并且如果必须的话，可以把请求翻译给下层的服务器协议。一个通道作为不改变消息的两个连接之间的中继点。当通讯需要通过一个中介(例如：防火墙等)或者是中介不能识别消息的内容时，通道经常被使用。

报文格式

HTTP报文由从客户机到服务器的请求和从服务器到客户机的响应构成。请求报文格式如下：

请求行 － 通用信息头 － 请求头 － 实体头 － 报文主体

请求行以方法字段开始，后面分别是 URL 字段和 HTTP 协议版本字段，并以 CRLF 结尾。SP 是分隔符。除了在最后的 CRLF 序列中 CF 和 LF 是必需的之外，其他都可以不要。有关通用信息头，请求头和实体头方面的具体内容可以参照相关文件。

应答报文格式如下：

状态行 － 通用信息头 － 响应头 － 实体头 － 报文主体

状态码元由3位数字组成，表示请求是否被理解或被满足。原因分析是对原文的状态码作简短的描述，状态码用来支持自动操作，而原因分析用来供用户使用。客户机无需用来检查或显示语法。有关通用信息头，响应头和实体头方面的具体内容可以参照相关文件。

工作原理

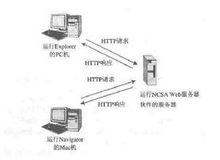
一次HTTP操作称为一个事务，其工作过程可分为四步：

首先客户机与服务器需要建立连接。只要单击某个超级链接，HTTP的工作就开始了。

建立连接后，客户机发送一个请求给服务器，请求方式的格式为：统一资源标识符（URL）、协议[版本号](https://baike.baidu.com/item/%E7%89%88%E6%9C%AC%E5%8F%B7)，后边是MIME信息包括请求修饰符、客户机信息和可能的内容。

服务器接到请求后，给予相应的响应信息，其格式为一个状态行，包括信息的协议版本号、一个成功或错误的代码，后边是MIME信息包括服务器信息、实体信息和可能的内容。

客户端接收服务器所返回的信息通过浏览器显示在用户的显示屏上，然后客

[](https://baike.baidu.com/pic/http/243074/0/3c2c4bfb9af4ba276d22ebe6?fr=lemma&ct=single)http工作流程图

户机与服务器断开连接。

如果在以上过程中的某一步出现错误，那么产生错误的信息将返回到客户端，由显示屏输出。对于用户来说，这些过程是由HTTP自己完成的，用户只要用鼠标点击，等待信息显示就可以了。

许多HTTP通讯是由一个用户代理初始化的并且包括一个申请在源服务器上资源的请求。最简单的情况可能是在用户代理和服务器之间通过一个单独的连接来完成。在Internet上，HTTP通讯通常发生在TCP/IP连接之上。缺省端口是TCP 80，但其它的端口也是可用的。但这并不预示着HTTP协议在Internet或其它网络的其它协议之上才能完成。HTTP只预示着一个可靠的传输。

这个过程就好像我们打电话订货一样，我们可以打电话给商家，告诉他我们需要什么规格的商品，然后商家再告诉我们什么商品有货，什么商品缺货。这些，我们是通过电话线用电话联系（HTTP是通过TCP/IP），当然我们也可以通过传真，只要商家那边也有传真。

状态消息

|  |  |
| --- | --- |
| **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。 |

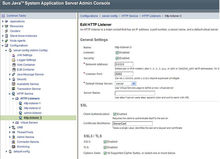
|  |  |
| --- | --- |
| **3xx:重定向** | |
| **消息** | **描述** |
| 300 Multiple Choices | 多重选择。链接列表。用户可以选择某链接到达目的地。最多允许五个地址。 |
| 301 Moved Permanently | 所请求的页面已经转移至新的url。 |
| 302 Found | 所请求的页面已经临时转移至新的url。 |
| 303 See Other | 所请求的页面可在别的url下被找到。 |
| 304 Not Modified | 未按预期修改文档。客户端有缓冲的文档并发出了一个条件性的请求（一般是提供If-Modified-Since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。 |

版本历史

超文本传输协议已经演化出了很多版本，它们中的大部分都是向下兼容的。在RFC 2145中描述了HTTP

[](https://baike.baidu.com/pic/http/243074/0/b3508d131f1568c46538db95?fr=lemma&ct=single)配置http通行证

版本号的用法。客户端在请求的开始告诉服务器它采用的协议版本号，而后者则在响应中采用相同或者更早的协议版本。

**0.9**已过时。只接受 GET 一种请求方法，没有在通讯中指定版本号，且不支持请求头。由于该版本不支持 POST 方法，所以客户端无法向服务器传递太多信息。

**HTTP/1.0**这是第一个在通讯中指定版本号的HTTP 协议版本，至今仍被广泛采用，特别是在[代理服务器](https://baike.baidu.com/item/%E4%BB%A3%E7%90%86%E6%9C%8D%E5%8A%A1%E5%99%A8)中。

**HTTP/1.1**当前版本。持久连接被默认采用，并能很好地配合代理服务器工作。还支持以管道方式同时发送多个请求，以便降低线路负载，提高传输速度。

HTTP/1.1相较于 HTTP/1.0 协议的区别主要体现在：

1 缓存处理

2 带宽优化及网络连接的使用

3 错误通知的管理

4 消息在网络中的发送

5 互联网地址的维护

6 安全性及完整性