HTTP 协议精解

WEB 服务器和浏览器之间的一问一答的交互过程也得遵循一定的规则，这个规则就是 HTTP 协议。它是 TCP/IP 协议集中的一个应用层协议，用于定义浏览 器和 WEB 服务器之间交换数据过程以及数据本身的格式。现在被广泛使用的 HTTP/1.1 相对 HTTP/1.0 而言，最大的特点就是支持持续连接。

1.HTTP/1.0 的会话方式：

建立连接，发送请求信息

客户机 ----------->服务器

客户机<-----------服务器

关闭连接，回送响应消息

每次连接只处理一个请求，即使是对同一个网站的每一个页面的访问，浏览器和 WEB 服务器都要建立一次单独的连接。

2.HTTP/1.1 的会话方式：

建立连接，第 1 个请求信息

---------->

.....

发送第 N 请求信息

---------->

客户机 回送第 1 次响应 服务器

<----------

.....
回应第 n 次响应

<----------

发送关闭连接请求

-------->

关闭连接

<----------

HTTP/1.1 支持持久连接，在一个TCP连接上可以传送多个HTTP请求和响应，减少了建立和关闭连接的消耗和延迟。一个包含有许多图象的网页文件的多个请求和响应可以在一个连接中传输，但每个单独的网页文件的请求和响应仍然需要使用各自的连接，还允许客户端不用等待上一次请求结果返回就可以发送下一个请求。

- **HTTP 消息的格式：**

一． 一个完整的请求消息包括：一个请求行、若干消息头、以及实体内容。

请求的格式为：

请求消息(组成):请求行(通用信息|请求头|实体头) CRLF[实体内容]

请求 行 = 方法 请求URL HTTP版本号 CRLF

方 法 = GET|HEAD|POST|扩展方法

U R L = 协议名称+宿主名+目录与文件名

HEAD——要求服务器查找某对象的元信息，而不是对象本身。

POST——从客户机向服务器传送数据，在要求服务器和CGI做进一步处理时会用到POST方法。POST主要用于发送HTML文本中FORM的内容，让CGI程序处理。

一个请求的例子为：

GET http://networking.zju.edu.cn/zju/index.htm HTTP/1.0

1.请求行(发送请求)：包括三个部分，即请求方式、资源路径、以及使用的 HTTP 协议版本。语法如下：请求方式资源路径 HTTP 版本号<CRLF>,其中<CRLF>表示回车和换行这两个字符的组合。HTTP 请求方式包括 POST、GET、HEAD、OPTIONS、DELETE、TRACE 和 PUT 几种。常用的是前两种。

1.1 使用 GET 和 POST 传递参数：

在 URL 地址后面可以附加一些参数，每个参数都由参数名和参数值组成，中间用＝分隔，各个参数用&分隔，URL 地址和整个参数之间用？分隔，如下所示：

http://www.it315.org/servlet/ParamsServlet?param1=aaa&param2=bbb

使用 **GET** 传递参数的**数据量是有限的**，一般限制在 **1KB** 以下。使用 **POST** 比 GET 要大的多。是**没有限制**的。但是必须设置 Content=Type

消息头为‘application/x-www-form-urlencoded’和设置 Content-Length 消息头为实体内容的长度。

二． 一个完整的响应消息包括：一个状态行、若干消息头、以及实体内容。

返回格式为

状态行(服务器返回客户端请求信息)：包括 HTTP 协议的版本号、一个状态码、以及对状态码进行描述的文本信息。

语法如下：HTTP 版本号 状态码 原因叙述<CRLF>

Ex: HTTP/1.1 200 OK

2．响应状态码：

2.1 常见如：

200：表示一切正常，返回的是正常请求结果。

404：表示服务器上不存在客户机上所请求的资源，这个状态码在浏览网页时最常见的。

2.2 具体如下：

HTTP 响应码

响应码由三位十进制数字组成，它们出现在由 HTTP 服务器发送的响应的第一行。

响应码分五种类型，由它们的第一位数字表示：

1.1xx：信息，请求收到，继续处理

2.2xx：成功，行为被成功地接受、理解和采纳

3.3xx：重定向，为了完成请求，必须进一步执行的动作

4.4xx：客户端错误，请求包含语法错误或者请求无法实现

5.5xx：服务器错误，服务器不能实现一种明显无效的请求

（后附详细响应码列表）

三。 通用信息头：即可用于请求，也可用于响应，是作为一个整体而不是特定资源与事务相关联。

Cache-Control：此字段用于通知客户机和服务器之间的代理服务器如何使用已缓存的页面。

Connection：用于指定处理完本次请求/响应后，客户端和服务器是否还要继续保持连接。

Date：用于表示 HTTP 消息产生的当前时间。

Transfer-Encoding：用于指定实体内容的传输编码方式。

请求头：允许客户端传递关于自身的信息和希望的响应形式。

Accept：用于指定客户端程序能够处理的 MIME 类型。有多个时用逗号隔开。

Accept-Charset：指出客户端程序可以使用的字符集。有多个时用逗号隔开。

Accept-Encoding：指定客户机能够进行解码的数据编码方式。有多个时用逗号隔开。

Accept-Language：指定客户机期望服务器返回哪个国家语言的文档。有多个时用逗号隔开。

Host：指定资源所在的主机名和端口号。

响应头：服务器用于传递自身信息的响应。

Accept-Range：用于说明当前 WEB 服务器是否接受 Range 请求和 Range 请求中指定的数据的单位。

Location：用于通知客户机应该到哪个新的地址去获取文档。由于当前响应并没有直接返回内容给客户机，所以使用 Location 头的 HTTP 消息不应该有实体内容，由此可见，在消息头中不能同时出现 Location 和 Contect-Type 这两个头自段。

实体头：定义被传送资源的信息。即可用于请求，也可用于响应。

Allow：用于指定客户机请求的资源所支持的请求方法。

Content-Encoding：用于指定实体内容的压缩编码方式。

Content-Language：用于指定返回的网页文档的国家语言类型。

Content-Length：用于指定实体内容的长度。

Content-Location：用于指定响应消息中所封装的实体内容的实际位置路径。

Content-Type：用于指定实体内容的 MIME 类型。客户机通过检查服务器响应消息的此字段中的 MIME 类型就能知道实体内容的数据格式和知道以 何种方式来进行处理了。<Tomcat>安装目录下的 conf 目录下的 web.xml 文件里面就定义了很多的类型。

Last-Modified：用于指定文档的最后修改时间。

扩展头：

Refresh：此字段告诉浏览器隔多长时间刷新。

下面简要介绍一下 HTTP 协议的请求头和应答头，这将有助于你进一步了解某些测试工具录制脚本的原理，便于编辑性能测试脚本。请求头和应答头都将用实例来说明。

请求消息举例：

POST /bbs/login2.asp?action=chk HTTP/1.1
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, application/vnd.ms-excel, application/vnd.ms-powerpoint, application/msword, application/x-shockwave-flash, */*
Referer: http://127.0.0.1/inpage.asp
Accept-Language: zh-cn
Content-Type: application/x-www-form-urlencoded
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; windows NT 5.0; (R1 1.3))
Host: 127.0.0.1Content-Length: 69
Connection: Keep-Alive
Cache-Control: no-cache
Cookie: aspsky=StatUserID=2200878793; Cookie1=y; ASPSESSIONIDQQTBSRCB=BHHIOGHAGFCKKNCGLLHDFHBM
username=snappyboy&CookieDate=0&password=123456&Submit3=%B5%C7%C2%BC

请求消息的第一行为下面的格式：

Method Request-URI HTTP-Version CRLF

例如：POST /bbs/login2.asp?action=chk HTTP/1.1

Method 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，向新闻组、BBS、邮件群组和数据库发送消息。

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。

Host 头域

Host 头域指定请求资源的 Intenet 主机和端口号，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域，否则系统会以 400 状态码返回。

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。

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（Partial Content）返回而不是以 200（OK）。

User-Agent 头域

User-Agent 头域的内容包含发出请求的用户信息。

Cache-Control 头域（请求和应答通用头域）

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。

应答消息举例：

```
HTTP/1.1 200 OK
server: microsoft-IIS/5.0
Date: Sat, 15 Jan 2005 15:36:26 GMT
Content-Length: 31401
Content-Type: text/html
Cache-control: private

<html>
<body>
省略内容
</body>
</html>
```

响应消息的第一行为下面的格式：

HTTP-Version Status-Code Reason-Phrase CRLF

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。

Location 响应头

Location 响应头用于重定向接收者到一个新 URI 地址。

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

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未接受方识别。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。

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

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-unit SP first-byte-pos -last-byte-pos/entity-legth

例如，传送头 500 个字节次字段的形式：Content-Range: bytes 0-499/1234 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。

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。

如何获取这些信息呢？

有许多的工具可以获得这些信息，一方面可以使用脚本录制工具，比如 WAS、LoadRunner、Jmeter 等，但是他们都是必须通过浏览器来录制（针对 web 协议部分的）。另外还可以借助于网络拦截软件来检测 HTTP 协议的活动情况，我使用的是 Visual Sniffer（是用于拦截通过网络传输的 TCP/IP/UDP/ICMP 等数据包的一个工具），它可以监测到通过本机的使用 HTTP 协议交互的数据。

例如：www-Authenticate: Basic realm=zxm.mgmt

HTTP/1.1 中用到的头标

头标格式：<name>:<value><CRLF>

Accept

定义客户端可以处理的媒体类型，按优先级排序；

在一个以逗号为分隔的列表中，可以定义多种类型和使用通配符。

例如：Accept: image/jpeg,image/png,*/*

Accept-Charset

定义客户端可以处理的字符集，按优先级排序；

在一个以逗号为分隔的列表中，可以定义多种类型和使用通配符。

例如：Accept-Charset: iso-8859-1,*,utf-8

Accept-Encoding

定义客户端可以理解的编码机制。

例如：Accept-Encoding:gzip,compress

Accept-Language

定义客户端乐于接受的自然语言列表。

例如：Accept-Language: en,de

Accept-Ranges

一个响应头标，它允许服务器指明：

将在给定的偏移和长度处，为资源组成部分的接受请求。

该头标的值被理解为请求范围的度量单位。

例如 Accept-Ranges: bytes 或 Accept-Ranges: none

Age

允许服务器规定自服务器生成该响应以来所经过的时间长度，

以秒为单位。该头标主要用于缓存响应。

例如：Age：30

Allow

一个响应头标，它定义一个由位于请求 URI 中的次源所支持的 HTTP 方法列表。

例如：Allow：GET,PUT

aUTHORIZATION

一个响应头标，用于定义访问一种资源所必需的授权

（域和被编码的用户 ID 与口令）。

例如：Authorization：Basic YXV0aG9yOnBoaWw=

Cache-Control

一个用于定义缓存指令的通用头标。

例如：Cache-Control：max-age=30

Connection

一个用于表明是否保存 socket 连接为开放的通用头标。

例如：Connection：close 或 Connection：keep-alive

Content-Base

一种定义基本 URI 的实体头标，为了在实体范围内解析相对 URLs。

如果没有定义 Content-Base 头标解析相对 URLs，

使用 Content-Location URI（存在且绝对）或使用 URI 请求。

例如：Content-Base: Http://www.myweb.com

Content-Encoding

一种介质类型修饰符，标明一个实体是如何编码的。

例如：Content-Encoding: zip

Content-Language

用于指定在输入流中数据的自然语言类型。

例如：Content-Language: en

Content-Length

指定包含于请求或响应中数据的字节长度。

例如：Content-Length:382

Content-Location

指定包含于请求或响应中的资源定位（URI）。

如果是一绝。对 URL 它也作为被解析实体的相对 URL 的出发点。

例如：Content-Location:

http://www.myweb.com/news

Content-MD5

实体的一种 MD5 摘要，用作校验和。

发送方和接受方都计算 MD5 摘要，接受方将其计算的值与

此头标中传递的值进行比较。

例如：Content-MD5: <base64 of 128 MD5 digest>

Content-Range

随部分实体一同发送；标明被插入字节的低位与高位字节偏移，

也标明此实体的总长度。

例如：Content-Range: 1001-2000/5000

Contern-Type

标明发送或者接收的实体的 MIME 类型。

例如：Content-Type: text/html

Date

发送 HTTP 消息的日期。

例如：Date: Mon,10PR 18:42:51 GMT

ETag

一种实体头标，它向被发送的资源分派一个唯一的标识符。

对于可以使用多种 URL 请求的资源，ETag 可以用于确定实际被发送的资源是否为同一资源。

例如：ETag: "208f-419e-30f8dc99"

Expires

指定实体的有效期。

例如：Expires: Mon,05 Dec 2008 12:00:00 GMT

Form

一种请求头标，给定控制用户代理的人工用户的电子邮件地址。

例如：From: webmaster@myweb.com

Host

被请求资源的主机名。对于使用 HTTP/1.1 的请求而言，此域是强制性的。

例如：Host: www.myweb.com

If-Modified-Since

如果包含了 GET 请求，导致该请求条件性地依赖于资源上次修改日期。

如果出现了此头标，并且自指定日期以来，此资源已被修改，

应该反回一个 304 响应代码。

例如：If-Modified-Since: Mon,10PR 18:42:51 GMT

If-Match

如果包含于一个请求，指定一个或者多个实体标记。

只发送其 ETag 与列表中标记区配的资源。

例如：If-Match: "208f-419e-308dc99"

If-None-Match

如果包含一个请求，指定一个或者多个实体标记。

资源的 ETag 不与列表中的任何一个条件匹配，操作才执行。

例如：If-None-Match: "208f-419e-308dc99"

If-Range

指定资源的一个实体标记，客户端已经拥有此资源的一个拷贝。

必须与 Range 头标一同使用。如果此实体自上次被客户端检索以来，

还不曾修改过，那么服务器只发送指定的范围，否则它将发送整个资源。

例如：Range: byte=0-499<CRLF>If-Range:"208f-419e-30f8dc99"

If-Unmodified-Since

只有自指定的日期以来，被请求的实体还不曾被修改过，才会返回此实体。

例如：If-Unmodified-Since:Mon,10PR 18:42:51 GMT

Last-Modified

指定被请求资源上次被修改的日期和时间。

例如：Last-Modified: Mon,10PR 18:42:51 GMT

Location

对于一个已经移动的资源，用于重定向请求者至另一个位置。

与状态编码 302（暂时移动）或者 301（永久性移动）配合使用。

例如：Location: http://www2.myweb.com/index.jsp

Max-Forwards

一个用于 TRACE 方法的请求头标，以指定代理或网关的最大数目，该请求通过网关才得以路由。

在通过请求传递之前，代理或网关应该减少此数目。

例如：Max-Forwards: 3

Pragma

一个通用头标，它发送实现相关的信息。

例如：Pragma: no-cache

Proxy-Authenticate

类似于 WWW-Authenticate，便是有意请求只来自请求链（代理）

的下一个服务器的认证。

例如：Proxy-Authenticate: Basic realm-admin

Proxy-Proxy-Authorization

类似于授权，但并非有意传递任何比在即时服务器链中更进一步的内容。

例如：Proxy-Proxy-Authorization: Basic

YXV0aG9yOnBoaWw=

Public

列表显示服务器所支持的方法集。

例如：Public: OPTIONS,MGET,MHEAD,GET,HEAD

Range

指定一种度量单位和一个部分被请求资源的偏移范围。

例如：Range: bytes=206-5513

Refener

一种请求头标域，标明产生请求的初始资源。对于 HTML 表单，它包含此表单的 Web 页面的地址。

例如：Refener:

http://www.myweb.com/news/search.html

Retry-After

一种响应头标域，由服务器与状态编码 503（无法提供服务）配合发送，

以标明再次请求之前应该等待多长时间。此时间即可以是一种日期，也可以是一种秒单位。

例如：Retry-After: 18

server

一种标明 Web 服务器软件及其版本号的头标。

例如：Server: Apache/2.0.46(Win32)

Transfer-Encoding

一种通用头标，标明对应被接受方反向的消息体实施变换的类型。

例如：Transfer-Encoding: chunked

Upgrade

允许服务器指定一种新的协议或者新的协议版本，与响应编码 101（切换协议）配合使用。

例如：Upgrade: HTTP/2.0

User-Agent

定义用于产生请求的软件类型（典型的如Web浏览器）。

例如：User-Agent: Mozilla/4.0(compatible; MSIE 5.5; windows NT; DigExt)

Vary

一个响应头标，用于表示使用服务器驱动的协商从可用的响应表示中选择响应实体。

例如：Vary: *

Via

一个包含所有中间主机和协议的通用头标，用于满足请求。

例如：Via: 1.0 fred.com, 1.1 wilma.com

Warning

用于提供关于响应状态补充信息的响应头标。

例如：Warning: 99 www.myweb.com Piano needs tuning

www-Authenticate

一个提示用户代理提供用户名和口令的响应头标，与状态编码 401（未授权）

配合使用。响应一个授权头标。

HTTP 响应码

响应码由三位十进制数字组成，它们出现在由 HTTP 服务器发送的响应的第一行。

响应码分五种类型，由它们的第一位数字表示：

1.1xx：信息，请求收到，继续处理

2.2xx：成功，行为被成功地接受、理解和采纳

3.3xx：重定向，为了完成请求，必须进一步执行的动作

4.4xx：客户端错误，请求包含语法错误或者请求无法实现

5.5xx：服务器错误，服务器不能实现一种明显无效的请求

下表显示每个响应码及其含义：

100　　继续

101　　分组交换协

200	OK
201	被创建
202	被采纳
203	非授权信息
204	无内容
205	重置内容
206	部分内容
300	多选项
301	永久地传送
302	找到
303	参见其他
304	未改动
305	使用代理
307	暂时重定向
400	错误请求
401	未授权
402	要求付费
403	禁止
404	未找到
405	不允许的方法
406	不被采纳
407	要求代理授权

408	请求超时
409	冲突
410	过期的
411	要求的长度
412	前提不成立
413	请求实例太大
414	请求URI太大
415	不支持的媒体类型
416	无法满足的请求范围
417	失败的预期
500	内部服务器错误
501	未被使用
502	网关错误
503	不可用的服务
504	网关超时
505	HTTP版本未被支持