

一、HTTP 协议概述

- > HTTP 是 HyperText Transfer Protocol(超文本传输协议)的简写, 传输 HTML 文件。
- > 用于定义 WEB 浏览器与 WEB 服务器之间**交换数据**的过程及数据本身的**格式**。

二、请求部分

```
POST /day08_02/1.html HTTP/1.1
Accept: application/x-ms-application, image/jpeg, application/xaml+xml, image/gif, ima
Referer: http://localhost:8080/day08_02/1.html
Accept-Language: zh-CN
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; WOW64; Trident/4.0; SLC
Content-Type: application/x-www-form-urlencoded
Accept-Encoding: gzip, deflate
Host: localhost:8080
Content-Length: 17
Connection: Keep-Alive
Cache-Control: no-cache

uName=tom&pwd=123
```

请求包括三部分:

1.请求消息行

GET /day08_02/1.html HTTP/1.1

请求方式 uri 请求协议

请求方式: Get (默认) POST DELETE HEAD 等

GET: 明文传输 不安全, 数据量有限, 不超过 1kb

GET /day08_02/1.html?uName=tom&pwd=123 HTTP/1.1

POST: 暗文传输, 安全。数据量没有限制。

URI: 统一资源标识符。去协议和 IP 地址。

协议/版本 :

2.请求消息头

从第 2 行到空行处, 都叫消息头

(1) **Accept:**浏览器可接受的 **MIME** 类型

告诉服务器客户端能接收什么样类型的文件。

(2) **Accept-Charset:** 浏览器通过这个头告诉服务器, 它支持哪种字符集

(3) **Accept-Encoding:**浏览器能够进行解码的数据编码方式, 比如 **gzip**

(4) **Accept-Language:**浏览器所希望的语言种类, 当服务器能够提供一种以上的语言版本时要用到。 可以在浏览器中进行设置。

(5) **Host:**初始 URL 中的主机和端口

(6) **Referrer:**包含一个 **URL**, 用户从该 **URL** 代表的页面出发访问当前请求的页面

(7) **Content-Type:**内容类型: 告诉服务器浏览器传输数据的 **MIME** 类型, 文件传输的类型, 如 **application/x-www-form-urlencoded** 表示使用 **post** 提交数据

(8) **If-Modified-Since:** Wed, 02 Feb 2011 12:04:56 GMT 利用这个头与服务器的文件进行比对, 如果一致, 则从缓存中直接读取文件。

(9) **User-Agent:**浏览器类型。

(10) **Content-Length:**表示请求消息正文的长度

(11) **Connection:**表示是否需要持久连接。如果服务器看到这里的值为 “Keep-Alive”, 或者看到请求使用的是 **HTTP 1.1** (**HTTP 1.1** 默认进行持久连接);

(12) Cookie:这是最重要的请求头信息之一 (在讲会话时解析)

(13) Date: Date: Mon, 22 Aug 2011 01:55:39 GMT 请求时间 GMT

3.消息正文

当请求方式是 POST 方式时，才能看见消息正文

uName=tom&pwd=123

三、响应部分

```
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Accept-Ranges: bytes
ETag: W/"188-1434960808873"
Last-Modified: Mon, 22 Jun 2015 08:13:28 GMT
Content-Type: text/html
Content-Length: 188
Date: Mon, 22 Jun 2015 08:22:41 GMT
```

<html> 服务器给浏览器发送的附加信息
<head>

响应包含三部分数据:

1.响应消息行

第一行:

HTTP/1.1 200 OK

协议/版本 响应状态码 对响应码的描述 (一切正常)

响应状态码:

常用的就 40 多个。

200(正常):一切正常

302/307(临时重定向)

304(未修改):表示客户机缓存的版本是最新的，客户机可以继续使用它，无需到服务器请求。

404(找不到):服务器上不存在客户机所请求的资源。

500(服务器内部错误)

2.响应消息头

Location: <http://www.it315.org/index.jsp> 指示新的资源的位置

通常和 302/307 一起使用，完成请求重定向

Server:apache tomcat 指示服务器的类型

Content-Encoding: gzip 服务器发送的数据采用的编码类型

Content-Length: 80 告诉浏览器正文的长度

Content-Language: zh-cn 服务发送的文本的语言

Content-Type: text/html; charset=GB2312 服务器发送的内容的 MIME 类型

Last-Modified: Tue, 11 Jul 2000 18:23:51 GMT 文件的最后修改时间

Refresh: 1;url=http://www.it315.org 指示客户端刷新频率。单位是秒

Content-Disposition: attachment; filename=aaa.zip 指示客户端下载文件

Set-Cookie:SS=Q0=5Lb_nQ; path=/search 服务器端发送的 Cookie

Expires: -1

Cache-Control: no-cache (1.1)

Pragma: no-cache (1.0) 表示告诉客户端不要使用缓存

Connection: close/Keep-Alive

Date: Tue, 11 Jul 2000 18:23:51 GMT

3.响应正文

和网页右键“查看源码”看到的内容一样。

四、区分请求重定向和请求转发

在浏览器中设置请求头信息，meta 标签：

http://www.w3school.com.cn/tags/tag_meta.asp

<meta> 元素可提供有关页面的元信息（meta-information），比如针对搜索引擎和更新频度的描述和关键词。标签位于文档的头部，它的属性定义了与文档相关联的名称/值对。

属性	值	描述
http-equiv	content-type expires refresh set-cookie	把 content 属性关联到 HTTP 头部。
name	author description keywords generator revised others	把 content 属性关联到一个名称。
scheme	some_text	定义用于翻译 content 属性值的格式。

（一）name 属性

name 属性提供了名称/值对中的名称。HTML 和 XHTML 标签都没有指定任何预先定义的 <meta> 名称。通常情况下，您可以自由使用对自己和源文档的读者来说富有意义的名称。

“keywords” 是一个经常被用到的名称。它为文档定义了一组关键字。某些搜索引擎在遇到这些关键字时，会用这些关键字对文档进行分类。类似这样的 meta 标签可能对于进入搜索引擎的索引有帮助：

```
<meta name="keywords" content="HTML, ASP, PHP, SQL">
```

如果没有提供 name 属性，那么名称/值对中的名称会采用 http-equiv 属性的值。

（二）http-equiv 属性

http-equiv 属性为名称/值对提供了名称。并指示服务器在发送实际的文档之前先在要传送给浏览器的 MIME 文档头部包含名称/值对。

当服务器向浏览器发送文档时，会先发送许多名称/值对。虽然有些服务器会发送许多这种名称/值对，但是所有服务器都至少要发送一个：

content-type:text/html。这将告诉浏览器准备接受一个 HTML 文档。

使用带有 http-equiv 属性的 <meta> 标签时，服务器将把名称/值对添加到发送给浏览器的内容头部。例如，添加：

```
<meta http-equiv="charset" content="iso-8859-1">
```

```
<meta http-equiv="expires" content="31 Dec 2008">
```

这样发送到浏览器的头部就应该包含：

```
content-type: text/html
```

```
charset:iso-8859-1
```

expires:31 Dec 2008

当然，只有浏览器可以接受这些附加的头部字段，并能以适当的方式使用它们时，这些字段才有意义。

(三) content 属性

content 属性提供了名称/值对中的值。该值可以是任何有效的字符串。

content 属性始终要和 name 属性或 http-equiv 属性一起使用。

(四) scheme 属性

scheme 属性用于指定要用来翻译属性值的方案。此方案应该在由 <head> 标签的 profile 属性指定的概况文件中进行了定义。

五、区分请求重定向和请求转发

请求重定向：浏览器请求两次；

请求转发：浏览器请求一次。