

HTTP

request:

Cache 头域

If-Modified-Since

作用：把浏览器端缓存页面的最后修改时间发送到服务器去，服务器会把这个时间与服务器上实际文件的最后修改时间进行对比。如果时间一致，那么返回304，客户端就直接使用本地缓存文件。如果时间不一致，就会返回200和新的文件内容。客户端接到之后，会丢弃旧文件，把新文件缓存起来，并显示在浏览器中。

If-None-Match

作用：If-None-Match和ETag一起工作，工作原理是在HTTP Response中添加ETag信息。当用户再次请求该资源时，将在HTTP Request 中加入If-None-Match信息(ETag的值)。如果服务器验证资源的ETag没有改变（该资源没有更新），将返回一个304状态告诉客户端使用本地缓存文件。否则将返回200状态和新的资源和Etag。使用这样的机制将提高网站的性能

Pragma

作用：防止页面被缓存，在HTTP/1.1版本中，它和Cache-Control:no-cache作用一模一样

Cache-Control

作用：这个是非常重要的规则。这个用来指定Response-Request遵循的缓存机制。各个指令含义如下

Cache-Control:Public 可以被任何缓存所缓存（）

Cache-Control:Private 内容只缓存到私有缓存中

Cache-Control:no-cache 所有内容都不会被缓存

Client 头域

Accept

作用：浏览器端可以接受的媒体类型，

例如：Accept: text/html 代表浏览器可以接受服务器回发的类型为text/html 也就是我们常说的html文档，

如果服务器无法返回text/html类型的数据，服务器应该返回一个406错误(non acceptable)

通配符 * 代表任意类型

例如 Accept: /* 代表浏览器可以处理所有类型，（一般浏览器发给服务器

都是发这个)

Accept-Encoding:

作用：浏览器申明自己接收的编码方法，通常指定压缩方法，是否支持压缩，支持什么压缩方法（gzip, deflate），（注意：这不是只字符编码）；

Accept-Language

作用：浏览器申明自己接收的语言。

语言跟字符集的区别：中文是语言，中文有多种字符集，比如big5, gb2312, gbk等等；

User-Agent

作用：告诉HTTP服务器，客户端使用的操作系统和浏览器的名称和版本。

Accept-Charset

作用：浏览器申明自己接收的字符集，这就是本文前面介绍的各种字符集和字符编码，如gb2312, utf-8（通常我们说Charset包括了相应的字符编码方案）

Cookie/Login 头域

Cookie:

作用：最重要的header, 将cookie的值发送给HTTP 服务器

Entity头域

Content-Length

作用：发送给HTTP服务器数据的长度。

Content-Type

作用：

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

Miscellaneous 头域

Referer:

作用：提供了Request的上下文信息的服务器，告诉服务器我是从哪个链接过来的，比如从我主页上链接到一个朋友那里，他的服务器就能够从HTTP Referer中统计出每天有多少用户点击我主页上的链接访问他的网站。

Transport 头域

Connection

例如： Connection: keep-alive 当一个网页打开完成后，客户端和服务端之间用于传输HTTP数据的TCP连接不会关闭，如果客户端再次访问这个服务器上的网页，会继续使用这一条已经建立的连接

例如： Connection: close 代表一个Request完成后，客户端和服务端之间

用于传输HTTP数据的TCP连接会关闭，当客户端再次发送Request，需要重新建立TCP连接。

Host（发送请求时，该报头域是必需的）

作用：请求报头域主要用于指定被请求资源的Internet主机和端口号，它通常从HTTP URL中提取出来的

HTTP Response header

Cache头域

Date

作用：生成消息的具体时间和日期

例如：Date: Sat, 11 Feb 2012 11:35:14 GMT

Expires

作用：浏览器会在指定过期时间内使用本地缓存

Vary

作用：

例如：Vary: Accept-Encoding

Cookie/Login 头域

P3P

作用：用于跨域设置Cookie，这样可以解决iframe跨域访问cookie的问题

例如：P3P: CP=CURa ADMa DEVa PSAo PSDo OUR BUS UNI PUR INT
DEM STA PRE COM NAV OTC NOI DSP COR

Set-Cookie

作用：非常重要的header，用于把cookie 发送到客户端浏览器，每一个写入cookie都会生成一个Set-Cookie.

Entity头域

ETag

作用：和If-None-Match 配合使用。

Last-Modified:

作用：用于指示资源的最后修改日期和时间。（实例请看上节的If-Modified-Since的实例）

Content-Type

作用：WEB服务器告诉浏览器自己响应的对象的类型和字符集，

Miscellaneous 头域

Server:

作用：指明HTTP服务器的软件信息

X-AspNet-Version:

作用：如果网站是用ASP.NET开发的，这个header用来表示[ASP.NET](#)的版本

X-Powered-By:

作用：表示网站是用什么技术开发的

例如：X-Powered-By: [ASP.NET](#)

Transport头域

Connection

例如：Connection: keep-alive 当一个网页打开完成后，客户端和服务端之间用于传输HTTP数据的TCP连接不会关闭，如果客户端再次访问这个服务器上的网页，会继续使用这一条已经建立的连接

例如：Connection: close 代表一个Request完成后，客户端和服务端之间用于传输HTTP数据的TCP连接会关闭，当客户端再次发送Request，需要重新建立TCP连接。

Location头域

Location

作用：用于重定向一个新的位置，包含新的URL地址

HTTP协议是无状态的和Connection: keep-alive的区别

无状态是指协议对于事务处理没有记忆能力，服务器不知道客户端是什么状态。从另一方面讲，打开一个服务器上的网页和你之前打开这个服务器上的网页之间没有任何联系。

HTTP是一个无状态的面向连接的协议，无状态不代表HTTP不能保持TCP连接，更不能代表HTTP使用的是UDP协议（无连接）。

从HTTP/1.1起，默认都开启了Keep-Alive，保持连接特性，简单地说，当一个网页打开完成后，客户端和服务端之间用于传输HTTP数据的TCP连接不会关闭，如果客户端再次访问这个服务器上的网页，会继续使用这一条已经建立的连接。

Keep-Alive不会永久保持连接，它有一个保持时间，可以在不同的服务器软件（如Apache）中设定这个时间。

http缓存机制

Expires：响应 一个绝对时间，在这个时间之前都有效，

Cache-Control: 请求/响应 相对值

可以取: Public: 可以被任何缓存所缓存

Private: 内容只缓存到私有缓存中

no-cache: 不建议使用本地缓存, 其仍然会缓存数据到本地

max-age: 相对时间, 在这个相对时间内都有效, 此时相当于

Expires = 当前时间+max-age

no-store: 不会在客户端缓存任何响应数据

no-cache相当于max-age = 0, 立即向浏览器请求

注意: 1、Cache-Control 中指定的缓存过期策略优先级高于 Expires, 当它们同时存在的时候, 后者会被覆盖掉。

2、缓存数据标记为已过期只是告诉客户端不能再直接从本地读取缓存了, 需要再发一次请求到服务器去确认, 并不等同于本地缓存数据从此就没用了, 有些情况下即使过期了还是会被再次用到, 具体下面会讲到。

Last-Modified: 响应 资源的最后修改日期和时间

If-Modified-Since 请求 浏览器缓存页面的最后修改时间

这种组合只能精确到秒

Tag 响应

If-None-Match 请求 etag的值