**[Http 1.1 Etag 与 Last-Modified](http://www.dbanotes.net/web/http_11_etag_lastmodified.html)**

本文地址：[www.dbanotes.net/web/http\_11\_etag\_lastmodified.html](http://www.dbanotes.net/web/http_11_etag_lastmodified.html)

我们可以大致得知，Entity tags 本质上说是一种"强校验器"，但是 HTTP 协议提供了一种通过给 Entity tags 打标签的"弱"的机制(类似于**内容的校验码**)．虽然这段话后面通过两种方式进行了解释，但是还是有些晦涩．我看了这段话之后只是得出了 Etag 的 "E" 代表 "Entity" 而已.

Etag 与 Last-Modified 是从 HTTP 1.0 到 HTTP 1.1 才有的概念．当我们从 Web 服务器获取文件的时候,只需要读取 HTTP 响应头的　Etag 与 Last-Modified　字段即可，这两个字段里面的具体内容是什么可以不管(可能会千奇百怪,RFC 2616 对 Etag 没有具体值的定义)，把这两个值 Cache 在本地，下次检查文件是否更新的时候比对这两个值即可．如果没有变化，服务器的响应代码不是 HTTP 200 (OK) , 而是 304.

Last-Modified   
     
　　在浏览器第一次请求某一个URL时，服务器端的返回状态会是200，内容是你请求的资源，同时有一个Last-Modified的属性标记此文件在服务期端最后被修改的时间，格式类似这样：   
     
Last-Modified: Fri, 12 May 2006 18:53:33 GMT   
     
　　客户端第二次请求此URL时，会在头部加入一个属性，询问该时间之后文件是否有被修改过：   
     
If-Modified-Since: Fri, 12 May 2006 18:53:33 GMT   
     
　　如果服务器端的文件没有被修改过，则返回状态是304，内容为空，这样就节省了传输数据量。如果服务器端的文件被修改过，则返回和第一次请求时类似。

ETag   
     
　　和Last-Modified类似，由于RFC2616(也就是HTTP/1.1)中没有说明ETag该是什么格式的，只要确保用双引号括起来就行了，所以你可以用文件的hash，甚至是直接用Last-Modified，以下是服务器端返回的格式：   
     
ETag: "50b1c1d4f775c61:df3"   
     
　　客户端的查询更新格式是这样的：   
     
If-None-Match: W/"50b1c1d4f775c61:df3"   
     
　　如果ETag没改变，则返回状态304然后不返回，这也和Last-Modified一样。

Expires   
     
　　这个属性就如我们在ASP中使用HttpResponse.ExpiresAbsolute一样直接，声明某某时刻过期之后浏览器就应该重新请求该URL，使用格式为：   
     
Expires: Sun, 10 Feb 2002 16:00:00 GMT   
     
　　注意HttpResponse.ExpiresAbsolute在ASP.NET中是不建议使用的，现在我们应该使用的是HttpResponse.Cache.SetExpires。   
     
　　Pragma   
     
　　通常我们用到的值就是no-cache，这和在Cache-Control中使用no-cache值是一样的，Cache-Control在下面讲。Pragma的使用格式如下：   
     
　　Pragma: no-cache   
　　Cache-Control   
     
　　这是一个集合型属性，它里面能够包含很多子属性，并且允许用户扩展新的子属性。常见的子属性包括：   
     
　　max-age - 以秒为单位的超时，覆盖Expires属性。   
　　public - 允许保存在共享缓存中。   
　　private - 只允许保存在私有缓存中。   
　　no-cache - 不允许缓存。   
　　no-store - 不允许缓存在持久介质中。   
　　no-transform - 不允许转换存储系统。