

理解A .NET与客户端缓存之HTTP协议 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/252/2021_2022__E7_90_86_E8_A7_A3A_N_c97_252677.htm 前言 这个系列的文章要讨论的是如何通过ASP.net服务器端技术来优化客户端缓存策略，而且让这种策略变得可配置和可扩展。我们要了解的知识从HTTP协议中相关属性对客户端缓存的影响，到ASP.NET如何控制这些属性来实现我们的缓存策略。 HTTP基础知识 由于讨论涉及到客户端缓存，所以还是先要简单介绍一下HTTP协议是如何控制客户端缓存的，这里涉及到HTTP头的Last-Modified、ETag等属性。 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HttpServletResponse.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HttpServletResponse.ExpiresAbsolute在ASP.NET中是不建议使用的，现在我们应该使用的是HttpServletResponse.Cache.SetExpires。

Pragma 通常我们用到的值就是no-cache，这和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。

100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com