

# HTTP 协议中的缓存

作者: [someone38063](#)

原文链接: <https://ld246.com/article/1558353689304>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

## 缓存的概念与分类

- 

- 

- 浏览器端的缓存，服务器端的缓存，代理服务器的缓存，ASP.NET 页面缓存，对象缓存，数据缓存

- 

- 

- HTTP 具有缓存功能的是**浏览器缓存**和**代理服务器缓存**

- 

- 

- HTTP 缓存是指当 Web 请求抵达缓存时，若本地有"已缓存的"副本，就可以从本地存储设备而是原始服务器提取这个文档

- 

- 

## 缓存的优点

- 

- 

- 减少了冗余的数据传输，节省了传输时间

- 

- 

- 减少了服务器的负担，大大提高了网站的性能

- 

- 

- 加快了客户端增加网页的速度

- 

- 

## 如何判断缓存新鲜度

- 

- 

- Web 服务器通过以下两种方式来判断浏览器缓存是否最新

- 

- 

- 浏览器把缓存文件的最后修改时间通过 Header "If-Modified-Since" 告诉 Web 服务器

- 

- 

- 

- 

- 如果客户端想请求一个文档，它首先检查本地缓存，发现存在这个文档的缓存，获取缓存文档的后修改时间，通过 "If-Modified-Since" 发送 HTTP 请求给服务器

- 

- 

- Web 服务器收到 HTTP 请求，将服务器的文档修改时间 (Last-Modified) 跟 HTTP 请求 Header 的 If-Modified-Since 相比较。如果时间是一样的，说明缓存是最新的，Web 服务器将发送状态码 304 (Not Modified) 给浏览器客户端，告诉客户端直接使用缓存里的版本

- 

- 

- 假如文档已经更新了，Web 服务器将发送该文档的最新版本给浏览器客户端

- 

- 

- 

- 浏览器把缓存文件的 Etag 通过 Header "If-None-Match" 告诉 Web 服务器

- 

## 与客户端有关的 Header

```
<table>
<thead>
<tr>
<th>名称</th>
<th>释义</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cache-Control: max-age=0</td>
<td>以秒为单位</td>
</tr>
<tr>
<td>If-Modified-Since: Monday, 19 Nov 2012 08:38:01 GMT</td>
<td>缓存文件的最后修改时间</td>
</tr>
<tr>
<td>If-None-Match: "0693f67a67cc1:0"</td>
<td>缓存文件的 Etag 值</td>
</tr>
<tr>
<td>Cache-Control: no-cache</td>
<td>不使用缓存</td>
</tr>
<tr>
<td>Pragma: no-cache</td>
<td>不使用缓存</td>
</tr>
<tr>
<td>Cache—Control: public</td>
<td>响应被缓存, 并且在多用户间共享</td>
</tr>
<tr>
<td>Cache-Control: private</td>
<td>相应只能作为私有缓存, 不能在用户之间共享</td>
</tr>
<tr>
<td>Cache-Control: no-cache</td>
<td>提醒浏览器要从服务器提取文档进行验证</td>
</tr>
<tr>
<td>Cache-Control: no-store</td>
<td>绝对禁止缓存(用于机密, 敏感文件)</td>
</tr>
<tr>
<td>Cache-Control: max-age=60</td>
<td>60s 后缓存过期(相对时间)</td>
</tr>
<tr>
<td>Date: Mon, 19 Nov 2012 08:40:01 GMT</td>
<td>当前响应发送的时间</td>
</tr>
<tr>
<td>Expires: Mon, 19 Nov 2012 08:38:01 GMT</td>
```

<td>服务器端文件的最后修改时间</td>

</tr>

<tr>

<td>Etag: "20b1add7ec1cd1:0" </td>

<td>服务器端文件的 Etag 值</td>

</tr>

</tbody>

</table>

<ul>

<li><strong>如果同时存在 cache-control 和 Expires, 浏览器总是优先使用 cache-control, 如果有 cache-control 才考虑 Expires</strong></li>

</ul>

<h2 id="Etag">Etag</h2>

<p>Entity Tag(实体标签), 是根据实体内容生成的一段 hash 字符串(类似于 MD5 或者 SHA1 之后结果), 可以识别资源的状态。当资源发生改变时, ETag 也发生变化</p>

<ul>

<li>

<p>ETag 是 Web 服务端产生的, 然后发给浏览器客户端, 浏览器客户端不用关心 Etag 是如何产生</p>

</li>

<li>

<p>使用 ETag 是为了解决一些 Last-Modified 无法解决的问题</p>

</li>

</ul>

<ol>

<li>

<p>某些服务器不能精确的得到文件最后的修改时间, 这样就无法通过修改最后时间来判断文件是否新了</p>

</li>

<li>

<p>某些文件的修改非常频繁, 以秒为单位一下的时间内进行修改, 而 Last-Modified 只能精确到秒</p>

</li>

<li>

<p>一些文件的最后修改时间改变了, 但是内容未改变, 我们不希望客户端认为这个文件修改了</p>

</li>

</ol>

<ul>

<li>

<p>某些网站很多图片或者 CSS 文件都使用了缓存, 这些都是通过比较 ETag 的值来判断文件是否新了</p>

</li>

<li>

<p>Request 中的 If-None-Match 和 Response 中的 ETag 的值一样, 说明文件没有更新, 告诉客户端使用本地缓存文件</p>

</li>

</ul>

<h2 id="浏览器不使用缓存">浏览器不使用缓存</h2>

<p>使用【ctrl+F】强制刷新浏览器, 可以让浏览器不使用缓存</p>

<ol>

<li>

<p>浏览器发送 HTTP 请求给 Web 服务器, Header 中带有 Cache-Control: no-cache, 明确告诉 Web 服务器客户端不使用缓存</p>

</li>

```
<li>
<p>Web 服务器把最新的文档发送给浏览器客户端</p>
</li>
</ol>
<ul>
<li>
<p>实例，打开某网页使用【Ctrl+F5】快捷键强制刷新浏览器，将看到浏览器发送的 HTTP 请求中有
Cache-Control: no-cache"</p>
</li>
<li>
<p><strong>"Pragma: no-cache"的作用和"Cache-Control: no-cache"一模一样，都是不使用
存</strong></p>
</li>
<li>
<p><strong>"Pragma: no-cache"是 HTTP1.0 中定义的，所以为了兼容 HTTP1.0 会同时使用"Pra
ma:no-cache"和"Cache-Control:no-cache"</strong></p>
</li>
</ul>
<h2 id="直接使用缓存-不去服务器验证">直接使用缓存，不去服务器验证</h2>
<ul>
<li>
<p>按【F5】快捷键刷新浏览器，浏览器会去 Web 服务器验证缓存</p>
</li>
<li>
<p>在地址栏输入网址然后按回车键，浏览器会"直接使用有效的缓存"，而不会发送 HTTP 请求去服
器验证缓存，这种情况叫<strong>缓存命中</strong></p>
</li>
</ul>
<h2 id="公有缓存和私有缓存的区别">公有缓存和私有缓存的区别</h2>
<ul>
<li>
<p>"Cache-Control: public"指可以公有缓存，缓存可以有数千名用户共享</p>
</li>
<li>
<p>"Cache-Control: private"指只支持私有缓存，私有缓存是单个用户专用的</p>
</li>
</ul>
```