链滴

Http 缓存 2：如何判断缓存是否过期

作者：Lord-X

原文链接：https://ld246.com/article/1587967243232

来源网站：链滴

<img src="https://ld246.com/images/img-loading.svg" alt="" data-src="https://b3logfile
com/bing/20180610.jpg?imageView2/1/w/960/h/540/interlace/1/q/100">
<h2 id="Http缓存-如何判断缓存是否过期">Http 缓存：如何判断缓存是否过期</h2>
缓存是否过期主要与 Response Header 的两类头部有关，一个是缓存最大有效时间（记为 fres
ness_lifetime），另一个是缓存已经存在的时间（记为 current_age）。
有个上面两个值后，就可以判断缓存是否过期了，逻辑如下：
<pre><code class="language-java highlight-chroma"><span c
ass="highlight-cl">if (</spa
> current_age) <spa
class="highlight-o">{
    <span class="h
ghlight-err">未过期
<span class="high
ight-o">} else {</sp
n>
    <span class="h
ghlight-err">已过期
<span class="high
ight-o">}
</code></pre>
接下来分别看一下这两个值怎么计算。
<h3 id="freshness-lifetime的计算">freshness_lifetime 的计算</h3>
freshness_lifetime 的值取自 Response Header，优先级如下：
<pre><code class="language-text highlight-chroma"><span cl
ss="highlight-cl">s-maxage > max-age > Expires > 预估的过期时间(下文中解释)
</code></pre>
下面针对前三种情况，举两个例子。

栗子 1：s-maxage 和 max-age

<img src="https://ld246.com/images/img-loading.svg" alt="HttpCache4.png" data-src="
ttps://b3logfile.com/file/2020/04/HttpCache4-6b9ea541.png?imageView2/2/interlace/1/form
t/jpg">
如上图，某一资源这两个值同时出现，则 <code>freshness_lifetime</code> 的值取 <code>s
maxage</code> 。

栗子 2：max-age 和 Expires

<img src="https://ld246.com/images/img-loading.svg" alt="HttpCache5.png" data-src="
ttps://b3logfile.com/file/2020/04/HttpCache5-16759f97.png?imageView2/2/interlace/1/forma
/jpg">
如上图，某一资源这两个值同时出现，则 <code>freshness_lifetime</code> 的值取 <code>
ax-age</code> 。
<h4 id="预估过期时间">预估过期时间</h4>
接下来说说 <code>预估的过期时间</code>。由于网络中某些资源没有通过 max-age 等头部
诉浏览器缓存这个资源（可能是服务器配置有问题），但这些资源通常是一些不长变化的静态资源，
如 js、css 等，这种情况下，浏览器为了性能考虑还是决定把它缓存。那缓存多久呢？现代浏览器通
是基于 RFC7234 推荐的计算方法，即：
<pre><code class="language-text highlight-chroma"><span cl
ss="highlight-cl">(DownloadTime - LastModified) * 10%
</code></pre>

DownloadTime：浏览器获取到响应的时间

LastModified：服务端资源上次修改时间

这个值的优先级是最低的，只有当 Response Header 没有返回前三个头部，并且浏览器决定缓这个资源的时候，才会使用他。
<h3 id="current-age的计算">current_age 的计算</h3>
current_age 的计算涉及到 Response Header 的 <code>age</code> 头部，所以我们先来确一下 <code>age</code> 的含义。
<code>age</code> 表示自源服务器发出资源的响应，到客户端使用这个资源的缓存时，经过秒数。
这里一定要注意的是：自源服务器发出响应的时间。举个例子：
一个资源从源服务器发出后，可能会经过多层代理服务器，最后到达客户端，而每一层代理服务也有可能把这个资源缓存，所以 age 的计算是要加上每一层代理服务器的缓存时间的，例如下图：</>

由上图可知，虽然 Browser 是直接从 <code>代理服务器1</code> 获取到的 <code>aaa.js</ode>，但是 <code>aaa.js</code> 从源服务器发出的时间是 12:00:00，所以 <code>age</code> 应该是 Browser 接收到响应的时间（12:02:10）减去源服务器发出的时间（12:00:00），等于 130。
PS：上例中的 age 计算过程是粗略的计算，实际计算时，还要计算每一个代理间的响时延。
以上就是 <code>current_age</code> 的计算方式。
<h2 id="参考">参考</h2>

极客时间 - Web 协议详解与抓包实战
RFC7234
Cacheable/li>
Http Cachin
