

目录	
第1章 优化的概念	
1 开篇词：你的前端性能还能再抢救一下	
2 解读雅虎35条军规（上）	
3 解读雅虎35条军规（下）	
4 你要不要看这些优化指标？	
第2章 性能工具介绍	
5 性能优化百宝箱（上）	
6 性能优化百宝箱（下）	
第3章 网络部分	
7 聊聊 DNS Prefetch	
8 Webpack 性能优化两三事	
9 图片加载优化（上）	
10 图片加载优化（下）	
第4章 缓存部分	
11 十八般缓存	
12 CDN 缓存	
13 本地缓存（Web Storage）	
14 浏览器缓存（上）	最近阅读
15 浏览器缓存（下）	
第5章 渲染部分	
16 渲染原理与性能优化	
17 如何应对首屏“一片空白”（上）	
18 如何应对首屏“一片空白”（下）	
19 不容小觑的 DOM 性能优化	

# 14 浏览器缓存（上）

更新时间：2019-08-22 11:54:40



“生活永远不像我们想像的那样好，但也不会像我们想像的那样糟。”  
——莫泊桑

**合理利用缓存是前端工程师的必修课。**上一节我们讲了浏览器本地的一些存储方案，那么这一节我们主要讲HTTP缓存策略。HTTP缓存策略不是前端单方面的事情，它需要服务端和我们共同配合才能完成，常见的服务器软件如：Apache、Nginx都可以为资源设置不同的HTTP缓存策略。在概念层面HTTP缓存策略又细分为强制缓存和协商缓存，同时这一部分也是面试当中的高频考点，因此，这一节内容大家一定要重视起来。

## 强制缓存

首先，不管是强制缓存还是协商缓存，主要针对的都是前端的静态资源，最完美的效果就是我们发起请求然后取到相应的静态资源。如果服务端的静态资源没有更新，我们就下次访问的时候直接从本地读取即可；如果服务端的静态资源已经更新，那么我们再次请求的时候就要到服务器请求最新的资源，然后进行拉取。

想要达到上述这种效果，需要强制缓存和协商缓存的共同配合才可以，具体的做法我们放到最后来讲，首先我们来讲强制缓存当中的字段。

### expires

expires是强制缓存策略的关键字段，expires是HTTP1.0的字段，通过指定一个具体的绝对时间值作为缓存资源的过期时间，具体的设置方法如下：

<div><div>← 慕课专栏</div><div>三 你不知道的前端性能优化技巧 / 14 浏览器缓存（上）</div></div>	
目录	
第1章 优化的概念	
1 开篇词：你的前端性能还能再抢救一下	
2 解读雅虎35条军规（上）	
3 解读雅虎35条军规（下）	
4 你要不要看这些优化指标？	
第2章 性能工具介绍	
5 性能优化百宝箱（上）	
6 性能优化百宝箱（下）	
第3章 网络部分	
7 聊聊 DNS Prefetch	
8 Webpack 性能优化两三事	
9 图片加载优化（上）	
10 图片加载优化（下）	
第4章 缓存部分	
11 十八般缓存	
12 CDN 缓存	
13 本地缓存（Web Storage）	
14 浏览器缓存（上） <div>最近阅读</div>	
15 浏览器缓存（下）	
第5章 渲染部分	
16 渲染原理与性能优化	
17 如何应对首屏“一片空白”（上）	
18 如何应对首屏“一片空白”（下）	
19 不容小觑的 DOM 性能优化	

我们在首次发起请求的时候，服务端会在Response Header当中设置expires字段，可以看到我们设置的过期时间是Tue, 09 Jul 2019 06:16:29 GMT。那么如果在这个时间之前我们发起请求去请求资源，我们就不会发起新的请求，直接使用本地已经缓存好的资源，这样我们可以有效减少了不必要的HTTP请求，不仅提升了性能，而且节省了流量，减少网络资源的消耗。

expires作为最开始的强制缓存解决方案，看起来没什么问题，但它的时间和服务端的时间是保持一致的，可是我们最终比较的时候是用本地时间和expires设置的时间进行比较。如果服务端的时间和我们本地的时间存在误差，那么缓存这个时候很容易就失去了效果，这个时候功能更强大的Cache-Control出现了。

### Cache-Control

Cache-Control同样也是强制缓存的关键字段。Cache-Control是HTTP1.1才有的字段，Cache-Control设置的是一个相对时间，可以更加精准地控制资源缓存。如下：

可以看到这里cache-control设置的值为max-age=315360000，这里的单位是秒，315360000代表缓存的时间跨度。由于这个是相对时间，所以不会受到服务端和本地时间不统一造成的缓存问题。315360000秒是一年，现在大多网站的静态资源设置的跨度都是315360000秒，也就是一年的时间跨度。

### 用法

expires的字段值是一个时间戳，而Cache-Control可设置的字段值较多，下面我们——来介绍。

- public：设置了该字段值的资源表示可以被任何对象（包括：发送请求的客户端、代理服务器等等）缓存。这个字段值不常用，一般还是使用\*\*max-age=\*\*来精确控制；
- private：设置了该字段值的资源只能被用户浏览器缓存，不允许任何代理服务器缓存。在实际开发当中，对于一些含有用户信息的HTML，通常都要设置这个字段值，避免代理服务器(CDN)缓存；
- no-cache：设置了该字段需要先和服务端确认返回的资源是否发生了变化，如果资源未发生变化，则直接使用缓存好的资源；

<div>← 慕课专栏</div>	<div>≡ 你不知道的前端性能优化技巧 / 14 浏览器缓存（上）</div>
<div>目录</div>	
<div>第1章 优化的概念</div>	<div><ul style="list-style-type: none"><li>• max-age=：设置缓存的最大有效期，单位为秒；</li><li>• s-maxage=：优先级高于max-age=，仅适用于共享缓存(CDN)，优先级高于 max-age 或者 Expires 头；</li><li>• max-stale=[=]：设置了该字段表明客户端愿意接收已经过期的资源，但是不能超过给定的时间限制。</li></ul></div>
<div>1 开篇词：你的前端性能还能再抢救一下</div>	
<div>2 解读雅虎35条军规（上）</div>	<div>上面就是强制缓存的常用字段，实际开发当中expires和Cache-Control一般都要进行设置，这是为了兼容不支持HTTP1.1的环境。两者同时存在，Cache-Control的优先级要高于expires。</div>
<div>3 解读雅虎35条军规（下）</div>	<div>协商缓存</div>
<div>4 你要不要看这些优化指标？</div>	<div>如果命中强制缓存，我们无需发起新的请求，直接使用缓存内容，如果没有命中强制缓存，如果设置了协商缓存，这个时候协商缓存就会发挥作用了。下面我就来介绍协商缓存中的字段。</div>
<div>第2章 性能工具介绍</div>	<div>Last-Modified/If-Modified-Since</div>
<div>5 性能优化百宝箱（上）</div>	<div>Last-Modified从字面意思就可以看出是最后一次的修改时间，设置方法和我们上面讲的强制缓存的设置方法一样，都是设置一个时间戳，同样它也是由服务端放到Response Headers返回给我们，如下：</div>
<div>6 性能优化百宝箱（下）</div>	<div></div>
<div>第3章 网络部分</div>	
<div>7 聊聊 DNS Prefetch</div>	
<div>8 Webpack 性能优化两三事</div>	
<div>9 图片加载优化（上）</div>	
<div>10 图片加载优化（下）</div>	
<div>第4章 缓存部分</div>	
<div>11 十八般缓存</div>	<div>如果有设置协商缓存，我们在首次请求的时候，返回的Response Headers会带有Last-Modified。当再次请求没有命中强制缓存的时候，这个时候我们的Request Headers就会携带If-Modified-Since字段，它的值就是我们第一次请求返回给我们的Last-Modified值。服务端接收到资源请求之后，根据If-Modified-Since的字段值和服务端资源最后的修改时间是否一致来判断资源是否有修改。<b>如果没有修改，则返回的状态码为304；如果有修改，则返回新的资源，状态码为200。</b></div>
<div>12 CDN 缓存</div>	
<div>13 本地缓存（Web Storage）</div>	
<div>14 浏览器缓存（上）最近阅读</div>	
<div>15 浏览器缓存（下）</div>	
<div>第5章 渲染部分</div>	
<div>16 渲染原理与性能优化</div>	
<div>17 如何应对首屏“一片空白”（上）</div>	
<div>18 如何应对首屏“一片空白”（下）</div>	<div>缺陷</div> <div><ol style="list-style-type: none"><li>1. 服务端对Last-Modified标注的最后修改时间只能精确到秒级，如果某些文件在1秒钟以内被修改多次的话，这个时候服务端无法准确标注文件的修改时间。</li></ol></div>
<div>19 不容小觑的 DOM 性能优化</div>	

目录	Etag/If-None-Match
第1章 优化的概念	可以看到Last-Modified/If-Modified-Since是有一定缺陷的，因此后来又增加了 Etag/If-None-Match，用法与Last-Modified/If-Modified-Since相似，但是Etag更准确。它通常是根据文件的具体内容计算出一个hash值，只要文件的内容不变，它就不会发生改变，保证了唯一性，这一点可以类比人的指纹。
1 开篇词：你的前端性能还能再抢救一下	Etag/If-None-Match的用法这里对应Last-Modified/If-Modified-Since。如果我们有设置协商缓存，在首次请求的时候，返回的Response Headers会带有Etag值。当再次请求没有命中强制缓存的时候，这个时候我们的Request Headers就会携带If-None-Match字段，它的值就是我们第一次请求返回给我们的Etag值。服务端再用Etag来进行比较，如果相同就直接使用缓存，如果不同再从服务端拉取新的资源。
2 解读雅虎35条军规（上）	
3 解读雅虎35条军规（下）	
4 你要不要看这些优化指标？	
第2章 性能工具介绍	
5 性能优化百宝箱（上）	
6 性能优化百宝箱（下）	
第3章 网络部分	
7 聊聊 DNS Prefetch	
8 Webpack 性能优化两三事	
9 图片加载优化（上）	
10 图片加载优化（下）	
第4章 缓存部分	
11 十八般缓存	
12 CDN 缓存	
13 本地缓存（Web Storage）	
14 浏览器缓存（上）最近阅读	
15 浏览器缓存（下）	
第5章 渲染部分	
16 渲染原理与性能优化	
17 如何应对首屏“一片空白”（上）	
18 如何应对首屏“一片空白”（下）	
19 不容小觑的 DOM 性能优化	

<div><div>← 慕课专栏</div><div>≡ 你不知道的前端性能优化技巧 / 14 浏览器缓存（上）</div></div>	
目录	欢迎在这里发表留言，作者筛选后可公开显示
第1章 优化的概念	
1 开篇词：你的前端性能还能再抢救一下	<div><div>Amor\ 安于命</div><div>下次再来</div><div><div>👍 0</div><div>回复</div></div><div>2020-06-20</div></div>
2 解读雅虎35条军规（上）	
3 解读雅虎35条军规（下）	
4 你要不要看这些优化指标？	<div><div>嫫嫫个性</div><div>这篇看懂了，不错，不错，不错。。。</div><div><div>👍 0</div><div>回复</div></div><div>2019-08-24</div></div>
第2章 性能工具介绍	
5 性能优化百宝箱（上）	
6 性能优化百宝箱（下）	
第3章 网络部分	
7 聊聊 DNS Prefetch	
8 Webpack 性能优化两三事	
9 图片加载优化（上）	
10 图片加载优化（下）	
第4章 缓存部分	
11 十八般缓存	
12 CDN 缓存	
13 本地缓存（Web Storage）	
14 浏览器缓存（上）	<div>最近阅读</div>
15 浏览器缓存（下）	
第5章 渲染部分	
16 渲染原理与性能优化	
17 如何应对首屏“一片空白”（上）	
18 如何应对首屏“一片空白”（下）	
19 不容小觑的 DOM 性能优化	