6 八月手写题目及常考面试题 已结束 · 学习时长：2022/7/9
7 手写 Promise 已结束 · 学习时长：2022/7/8
8 Event Loop 已结束 · 学习时长：1小时0分
9 八月面试知识点及常考面试题 已结束 · 学习时长：1小时0分
10 八月 考题 已结束 · 学习时长：2小时0分
11 DevTools Tips 已结束 · 学习时长：52分0秒
12 浏览器编辑知识点及常考面试题 学习时长：2小时0分
13 浏览器缓存机制 学习时长：40分0秒
14 浏览器渲染原理 学习时长：1小时0分
15 安全防范知识点

浏览器缓存机制

分享：请知识点整理为可分享的文章，并且第一条节奖励一个徽章。

缓存可以说是性能优化中最简单高效的一种优化方式了，它可以**显著减少网络传输所带来的损耗**。

对于一个数据请求来说，可以分为发起网络请求、后端处理、浏览器响应三个步骤，浏览器缓存可以帮助我们实现第一和第三步骤中的优化。正确设置缓存策略所产生的效果，可谓是如丝般顺滑的数据和极致一快。那么就实现令爱不再费思量的用户体验，这样就决定了前端的重要。

接下来内容中我们将通过以下几个部分来探讨浏览器缓存机制：

- 缓存位置
- 缓存策略
- 实际场景应用缓存策略

缓存位置

从最快到最慢，并且各自有优先级，当前次查找缓存命中失效后，才会去请求网络

1. Service Worker
2. Memory Cache
3. Disk Cache
4. Push Cache
5. 网络请求

Service Worker

在上一章节中我们已经介绍了 Service Worker 的内容，这里就不再过多的展开了。

Service Worker 的缓存与浏览器其他内建的缓存机制相比不同，它可以让我们最细粒度地令哪些文件，如何优雅地，如何高效地，并且**最终掌握命中时机**。

当 Service Worker 没有命中缓存的时候，我们需要去调用 `fetch` 函数获取数据。也就是说，如果我们没有在 Service Worker 命中缓存的话，还会需要去调用网络请求的过程。**总而言之我们当 Service Worker 中还是拥有了对网络请求的控制权**。浏览器会自己去找三级从 Service Worker 中获取的内容。

Memory Cache

Memory Cache 也就是内存中的缓存，读取内存中的数据要比读取快。**也意味着内存虽然很宝贵，可是最终仍然要用到，合理最大程度的释放资源**。一旦我们关闭 Tab 页面，内存中的缓存也就被释放了。

当我们访问过页面以后，再次刷新页面，可以发现很多数据都来自于内存缓存

Name	Status	Type	Initiator	Size	Waterfall
localhost	200	document	Other	512 KB	
sockjs-node/info?t=1662600000-1	200	xhr	(index)	84 B	
sockjs-node/iframe.html?hash=a7a	304	text/html	(index)	from memory cache	
app.4d9b8f85.bundle.chunk.js	200	script	(index)	from memory cache	
vendor.js	200	script	(index)	from memory cache	
app.4d9b8f85.hot-update.js	200	script	(index)	from memory cache	
127.0.0.1__sockjs__n=1	101	websocket	sockjs-client	0 B	

那么既然内存容量是有上限的，我们要不要通过直接最优强迫内存中呢？

先说结论，这是不可能的。毫无计算机中的内存一定比硬盘容量小很多，操作系统需要相打如果内存使用，所以通过我们浏览器的内存会把不多，内存中其实可以分到大部分的文件，比如说 JS、HTML、CSS，图片等等，但是浏览器也会把那些文件放在内存这个公用空间位置罗了，那是说了很多资料都没有一个定论。

当然，我通过一些实践和观察的情况下一些结论：

- 对于大文件来说，大概率是不存储在内存的，生之因为
- 当前系统的使用率高的话，文件也为存储的基础

Disk Cache

Disk Cache 也就是存储在硬盘中的缓存，读取速度慢点，但是什么都能存储的磁盘中，比之 Memory Cache **胜在容量和存储时间胜过**。

也所有浏览器都会写，Disk Cache **基是浏览器长是多久**，它会根据 HTTP Header 中的字段的相关缓存控制字段来决定。哪些资源会被存储在硬盘，哪些资源过期过期需要重新请求，**并且即使在浏览器关闭的情况下，相同地址的资源一旦被存储起来了下来，那不会再次去请求数据**。

Push Cache

Push Cache 是 HTTP/2 中的内容，当以上三种缓存都没有命中时，它才会被使用。**并且最后时间也很短暂，只在会话（Session）中存在，一旦会话结束就被释放**。

Push Cache 在国内使用得的资料很少，也是因为 HTTP/2 在国内不够普及。但是 HTTP/2 并也更注的的一个提倡，这里推荐阅读 HTTP2 push is tougher than I thought 这篇文章，但是内容是英文的，我翻译一下文章中的几个要点，有兴趣的同学还是推荐自己动读：

- 所有的资源都能被推送，但是 Edge 和 Safari 浏览器兼容性不怎么好
- 可以推送 `cross-origin` 和 `no-cross` 的资源
- 可以推送缓存的，Push Cache 就被释放
- 多个页面可以使用相同的 HTTP/2 连接，也就是说使用同样的缓存
- Push Cache 中的缓存只能被使用一次
- 浏览器可以拒绝接受已经存在的资源推送
- 你可以设置推送权重给资源

网络请求

如果所有缓存都没有命中的话，那么只能发起请求来获取资源了。

那么为了性能上的考虑，大部分的接口都应该选择好缓存策略，接下来我们就来学习缓存策略这部分的内容。

缓存策略

通常浏览器缓存策略分为两种：**强缓存**和**协商缓存**，并且缓存策略都是通过设置 HTTP Header 来实现的。

强缓存

强缓存可以通过设置两种 HTTP Header 实现：`Expires` 和 `Cache-Control`。强缓存表示在缓存期间不需要请求，`state code` 为 200。

Expires

```
Expires: Wed, 22 Oct 2018 08:41:00 GMT
```

`Expires` 是 HTTP/1 的产物，表示资源会在 `Wed, 22 Oct 2018 08:41:00 GMT` 的过期，需要再次请求。并且 `Expires` **受限于本地时间**，如果修改了本地时间，可能会造成缓存失效。

Cache-control

```
Cache-control: max-age=30
```

`Cache-Control` 出现于 HTTP/1.1，**优先级高于 `Expires`**。该属性表示资源会在 30 秒后过期，需要再次请求。

`Cache-Control` **可以在请求头或者响应头中设置**，并且可以组合使用多种指令

协商缓存工具箱思维导图

从图中我们可以看到，我们可以将多个指令配合起来一起使用，达到多个目的。比如说我们希望资源能被缓存下来，并且是客户端和代理服务器都能缓存，还能设置最快失效时间等等。

接下来我们就来学习一些常见指令的用

指令	作用
public	表示响应可以被客户端和代理服务器缓存
private	表示响应只可以被客户端缓存
max-age=30	缓存 30 秒后就过期，需要重新请求
s-maxage=30	覆盖 max-age，作用一样，只是代理服务器中生效
no-store	不缓存任何响应
no-cache	资源被缓存，但是立即失效，下次发起请求或验证资源是否过期
max-stale=30	30 秒内，即使缓存过期，也使用该缓存
min-fresh=30	希望在 30 秒内资源保持新鲜状

协商工具箱思维导图

协商缓存

如果缓存过期了，浏览器发起请求验证资源是否有更新，协商缓存可以通过设置两种 HTTP Header 实现：`Last-Modified` 和 `ETag`。

当浏览器发起请求验证资源时，如果资源没有修改过，那么服务器端回应 304 状态码，并且更新浏览器缓存有效期。

协商工具箱思维导图

Last-Modified 和 If-Modified-Since

`Last-Modified` 表示本地文件最后修改日期，`If-Modified-Since` 会将 `Last-Modified` 的值发送给服务器，让服务器确认该资源是否有更新。有更新的话就会将新的资源发送回来。否则返回 304 状态码。

但是 `Last-Modified` 存在一些弊端：

- 如果本地打开缓存文件，即便没有对文件进行修改，还是会造成 `Last-Modified` 被修改，服务端不能命中缓存导致发送相同的资源
- 因为 `Last-Modified` 只能以秒计时，如果在不可感知的时间内修改完成文件，那么服务端会认为资源还是命中了，不会返回这个最新的资源

以上这些弊端，所以在 HTTP/1.1 出现了 `ETag`。

ETag 和 If-None-Match

`ETag` 类似于文件指纹，`If-None-Match` 会将当前 `ETag` 发送给服务器，询问该资源 `ETag` 是否变动。有变动的话就将新的资源发送回来。并且 `ETag` 优先级比 `Last-Modified` 高。

以上就是缓存策略的所有内容了，看到这里，不禁感叹浪漫存在这样一个城市，**如果什么都没有被解决过呢，那么浪漫就会怎么样？**

对于这种情况，浪漫就会采用一个花里式的浪漫，通常会强制成功本年的 `Date` 减去 `Last-Modified` 值的 10% 作为缓存时长。

实际场景应用缓存策略

单纯了解理论并不代表于实践是没有意义的，接下来我们来通过几个场景学习下如何使用这些理论。

频繁变动的资源

对于频繁变动的资源，首先需要使用 `Cache-Control: no-cache` 使浏览器每次都请求服务器，然后配合 `ETag` 或者 `Last-Modified` 来验证资源是否有变，这样的话这条资源是有效，还是会需要减少响应数据大小。

代码文件

这里特指除了 HTML 外的代码文件，因为 HTML 文件一般不缓存或者缓存时间很短。

一般来说，现在都会使用工具来打包代码。那么我们就可以对文件名进行哈希处理，只有当代码改动了文件名就会不同，基于此，我们就可以对这些文件设置缓存有效期为一年：`Cache-Control: max-age=31536000`，这样只有当 HTML 文件中引入的文件名发生了改变才会去下载最新的代码文件，否则统一直接缓存即可。

小结

在这一章节中我们了解了浏览器的缓存机制，并且列举了几个场景来实践我们学习到的理论。如果大家对于这个章节的内容有疑问，欢迎留评或发个私信。

留言

输入评论（Enter换行，# + Enter发送）

登录/注册

全部评论 (21)

最新 最早 只看作者

最新 最早 只看作者

web艺术家 · 1年前

其实！在第一时间看到这篇文章，我立刻发现你在怎么回答！对于这种情况，浏览器实现一个默认的缓存策略，这个默认的缓存策略，大概率是 `Last-Modified` 的 `304` 响应吧？

是，浏览器会在第一时间发出一个请求给服务器，并且带上 `If-Modified-Since` 和 `If-None-Match` 这两个请求头。**这两个请求头的值是服务器在第一次响应的时候发送回来的 `Last-Modified` 和 `ETag` 的值**。

`If-Modified-Since` 的值是 `Last-Modified` 的值，`If-None-Match` 的值是 `ETag` 的值。

服务器会根据这两个请求头来判断客户端的资源是否有更新，如果没有更新的话，就会返回一个 `304 Not Modified` 的响应，告诉浏览器可以使用本地缓存。

`304` 响应中，除了必要的响应头（`Cache-Control`、`Content-Location`、`Date`、`ETag`、`Expires` 和 `Vary`）以外，没有任何响应体。

`Cache-Control: no-cache` 的含义是：每次使用缓存前，都必须先向服务器确认该缓存是否仍有效。

`Cache-Control: no-store` 的含义是：不允许浏览器和代理服务器缓存任何内容。

`Cache-Control: max-age=31536000` 的含义是：资源在 31536000 秒（1 年）内被视为新鲜，浏览器可以直接使用缓存，而不需要发送网络请求。

`Cache-Control: max-age=0` 的含义是：资源立即过期，浏览器每次使用前都必须向服务器发起请求来验证。

`Cache-Control: max-stale=30` 的含义是：即使缓存已经过期了 30 秒，浏览器仍然可以使用这个缓存，而不需要发起新的请求。

`Cache-Control: min-fresh=30` 的含义是：浏览器要求缓存在接下来的 30 秒内保持新鲜，如果缓存会在 30 秒内过期，浏览器就会发起请求来获取新的资源。

`Cache-Control: private` 的含义是：资源只能被浏览器缓存，代理服务器不能缓存。

`Cache-Control: public` 的含义是：资源可以被浏览器和代理服务器缓存。

`Cache-Control: must-revalidate` 的含义是：缓存过期后，必须先向服务器验证，不能使用过期的缓存。

`Cache-Control: proxy-revalidate` 的含义是：代理服务器必须向源服务器验证缓存，浏览器可以使用过期的缓存。

`Cache-Control: s-maxage=60` 的含义是：代理服务器缓存 60 秒，浏览器不受此指令影响。

`Cache-Control: no-transform` 的含义是：代理服务器不能对内容进行压缩、转换等操作。

`Cache-Control: immutable` 的含义是：资源在有效期内永远不会改变，浏览器可以放心地使用缓存，连验证都省了。

`Cache-Control: stale-while-revalidate=60` 的含义是：资源过期后，浏览器可以先使用过期的缓存，同时在后台发起请求来更新缓存，这样用户就能立即看到内容，而不用等待网络请求。

`Cache-Control: stale-if-error=60` 的含义是：如果在 60 秒内，服务器发生错误，浏览器可以使用过期的缓存来应对。

`Cache-Control: no-cache, no-store, must-revalidate` 的含义是：这是一个组合指令，通常用于最大程度地确保内容是最新的，并且不使用任何缓存。

`Cache-Control: private, max-age=0, must-revalidate` 的含义是：这是一个组合指令，表示内容是私有的，并且立即过期，必须重新验证。

`Cache-Control: public, max-age=31536000, immutable` 的含义是：这是一个组合指令，表示内容是公共的，可以缓存一年，并且在有效期内永远不会改变。

`Cache-Control: no-cache, private, max-age=0, must-revalidate, proxy-revalidate` 的含义是：这是一个极其保守的组合指令，几乎禁止了所有缓存行为，只要求浏览器和代理服务器在每次使用内容前，必须向源服务器进行验证，并且内容是私有的。

`Cache-Control: max-age=604800, stale-while-revalidate=86400, stale-if-error=86400` 的含义是：这是一个复杂的组合指令，表示内容有效期为 7 天，过期后 1 天内可以使用过期内容并后台更新，如果发生错误，1 天内也可以使用过期内容。

`Cache-Control: private, no-cache, no-store, must-revalidate, max-stale=0, post-check=0, pre-check=0` 的含义是：这是一个极其冗长的组合指令，几乎禁止了所有缓存行为，只要求浏览器和代理服务器在每次使用内容前，必须向源服务器进行验证，并且内容是私有的。

`Cache-Control: no-cache, no-store, must-revalidate, max-stale=0, post-check=0, pre-check=0, private, max-age=0, s-maxage=0, proxy-revalidate, no-transform, immutable, stale-while-revalidate=0, stale-if-error=0` 的含义是：这是一个极其冗长的组合指令，几乎禁止了所有缓存行为，只要求浏览器和代理服务器在每次使用内容前，必须向源服务器进行验证，并且内容是私有的。

`Cache-Control: private, no-cache, no-store, must-revalidate, max-stale=0, post-check=0, pre-check=0, max-age=0, s-maxage=0, proxy-revalidate, no-transform, immutable, stale-while-revalidate=0, stale-if-error=0, x-custom-directive=value` 的含义是：这是一个极其冗长的组合指令，几乎禁止了所有缓存行为，只要求浏览器和代理服务器在每次使用内容前，必须向源服务器进行验证，并且内容是私有的。

`Cache-Control: private, no-cache, no-store, must-revalidate, max-stale=0, post-check=0, pre-check=0, max-age=0, s-maxage=0, proxy-revalidate, no-transform, immutable, stale-while-revalidate=0, stale-if-error=0, x-custom-directive=value, x-another-directive=value` 的含义是：这是一个极其冗长的组合指令，几乎禁止了所有缓存行为，只要求浏览器和代理服务器在每次使用内容前，必须向源服务器进行验证，并且内容是私有的。

`Cache-Control: private, no-cache, no-store, must-revalidate, max-stale=0, post-check=0, pre-check=0, max-age=0, s-maxage=0, proxy-revalidate, no-transform, immutable, stale-while-revalidate=0, stale-if-error=0, x-custom-directive=value, x-another-directive=value, x-yet-another-directive=value` 的含义是：这是一个极其冗长的组合指令，几乎禁止了所有缓存行为，只要求浏览器和代理服务器在每次使用内容前，必须向源服务器进行验证，并且内容是私有的。

`Cache-Control: private, no-cache, no-store, must-revalidate, max-stale=0, post-check=0, pre-check=0, max-age=0, s-maxage=0, proxy-revalidate, no-transform, immutable, stale-while-revalidate=0, stale-if-error=0, x-custom-directive=value, x-another-directive=value, x-yet-another-directive=value, x-final-directive=value` 的含义是：这是一个极其冗长的组合指令，几乎禁止了所有缓存行为，只要求浏览器和代理服务器在每次使用内容前，必须向源服务器进行验证，并且内容是私有的。

`Cache-Control: private, no-cache, no-store, must-revalidate, max-stale=0, post-check=0, pre-check=0, max-age=0, s-maxage=0, proxy-revalidate, no-transform, immutable, stale-while-revalidate=0, stale-if-error=0, x-custom-directive=value, x-another-directive=value, x-yet-another-directive=value, x-final-directive=value, x-ultimate-directive=value` 的含义是：这是一个极其冗长的组合指令，几乎禁止了所有缓存行为，只要求浏览器和代理服务器在每次使用内容前，必须向源服务器进行验证，并且内容是私有的。

`Cache-Control: private, no-cache, no-store, must-revalidate, max-stale=0, post-check=0, pre-check=0, max-age=0, s-maxage=0, proxy-revalidate, no-transform, immutable, stale-while-revalidate=0, stale-if-error=0, x-custom-directive=value, x-another-directive=value, x-yet-another-directive=value, x-final-directive=value, x-ultimate-directive=value, x-super-ultimate-directive=value` 的含义是：这是一个极其冗长的组合指令，几乎禁止了所有缓存行为，只要求浏览器和代理服务器在每次使用内容前，必须向源服务器进行验证，并且内容是私有的。

`Cache-Control: private, no-cache, no-store, must-revalidate, max-stale=0, post-check=0, pre-check=0, max-age=0, s-maxage=0, proxy-revalidate, no-transform, immutable, stale-while-revalidate=0, stale-if-error=0, x-custom-directive=value, x-another-directive=value, x-yet-another-directive=value, x-final-directive=value, x-ultimate-directive=value, x-super-ultimate-directive=value, x-mega-ultimate-directive=value` 的含义是：这是一个极其冗长的组合指令，几乎禁止了所有缓存行为，只要求浏览器和代理服务器在每次使用内容前，必须向源服务器进行验证，并且内容是私有的。

`Cache-Control: private, no-cache, no-store, must-revalidate, max-stale=0, post-check=0, pre-check=0, max-age=0, s-maxage=0, proxy-revalidate, no-transform, immutable, stale-while-revalidate=0, stale-if-error=0, x-custom-directive=value, x-another-directive=value, x-yet-another-directive=value, x-final-directive=value, x-ultimate-directive=value, x-super-ultimate-directive=value, x-mega-ultimate-directive=value, x-hyper-ultimate-directive=value` 的含义是：这是一个极其冗长的组合指令，几乎禁止了所有缓存行为，只要求浏览器和代理服务器在每次使用内容前，必须向源服务器进行验证，并且内容是私有的。

`Cache-Control: private, no-cache, no-store, must-revalidate, max-stale=0, post-check=0, pre-check=0, max-age=0, s-maxage=0, proxy-revalidate, no-transform, immutable, stale-while-revalidate=0, stale-if-error=0, x-custom-directive=value, x-another-directive=value, x-yet-another-directive=value, x-final-directive=value, x-ultimate-directive=value, x-super-ultimate-directive=value, x-mega-ultimate-directive=value, x-hyper-ultimate-directive=value, x-ultra-directive=value` 的含义是：这是一个极其冗长的组合指令，几乎禁止了所有缓存行为，只要求浏览器和代理服务器在每次使用内容前，必须向源服务器进行验证，并且内容是私有的。

`Cache-Control: private, no-cache, no-store, must-revalidate, max-stale=0, post-check=0, pre-check=0, max-age=0, s-maxage=0, proxy-revalidate, no-transform, immutable, stale-while-revalidate=0, stale-if-error=0, x-custom-directive=value, x-another-directive=value, x-yet-another-directive=value, x-final-directive=value, x-ultimate-directive=value, x-super-ultimate-directive=value, x-mega-ultimate-directive=value, x-hyper-ultimate-directive=value, x-ultra-directive=value, x-mega-ultra-directive=value` 的含义是：这是一个极其冗长的组合指令，几乎禁止了所有缓存行为，只要求浏览器和代理服务器在每次使用内容前，必须向源服务器进行验证，并且内容是私有的。

`Cache-Control: private, no-cache, no-store, must-revalidate, max-stale=0, post-check=0, pre-check=0, max-age=0, s-maxage=0, proxy-revalidate, no-transform, immutable, stale-while-revalidate=0, stale-if-error=0, x-custom-directive=value, x-another-directive=value, x-yet-another-directive=value, x-final-directive=value, x-ultimate-directive=value, x-super-ultimate-directive=value, x-mega-ultimate-directive=value, x-hyper-ultimate-directive=value, x-ultra-directive=value, x-mega-ultra-directive=value, x-super-ultra-directive=value` 的含义是：这是一个极其冗长的组合指令，几乎禁止了所有缓存行为，只要求浏览器和代理服务器在每次使用内容前，必须向源服务器进行验证，并且内容是私有的。

`Cache-Control: private, no-cache, no-store, must-revalidate, max-stale=0, post-check=0, pre-check=0, max-age=0, s-maxage=0, proxy-revalidate, no-transform, immutable, stale-while-revalidate=0, stale-if-error=0, x-custom-directive=value, x-another-directive=value, x-yet-another-directive=value, x-final-directive=value, x-ultimate-directive=value, x-super-ultimate-directive=value, x-mega-ultimate-directive=value, x-hyper-ultimate-directive=value, x-ultra-directive=value, x-mega-ultra-directive=value, x-super-ultra-directive=value, x-hyper-ultra-directive=value` 的含义是：这是一个极其冗长的组合指令，几乎禁止了所有缓存行为，只要求浏览器和代理服务器在每次使用内容前，必须向源服务器进行验证，并且内容是私有的。

`Cache-Control: private, no-cache, no-store, must-revalidate, max-stale=0, post-check=0, pre-check=0, max-age=0, s-maxage=0, proxy-revalidate, no-transform, immutable, stale-while-revalidate=0, stale-if-error=0, x-custom-directive=value, x-another-directive=value, x-yet-another-directive=value, x-final-directive=value, x-ultimate-directive=value, x-super-ultimate-directive=value, x-mega-ultimate-directive=value, x-hyper-ultimate-directive=value, x-ultra-directive=value, x-mega-ultra-directive=value, x-super-ultra-directive=value, x-hyper-ultra-directive=value, x-ultra-ultra-directive=value` 的含义是：这是一个极其冗长的组合指令，几乎禁止了所有缓存行为，只要求浏览器和代理服务器在每次使用内容前，必须向源服务器进行验证，并且内容是私有的。

`Cache-Control: private, no-cache, no-store, must-revalidate, max-stale=0, post-check=0, pre-check=0, max-age=0, s-maxage=0, proxy-revalidate, no-transform, immutable, stale-while-revalidate=0, stale-if-error=0, x-custom-directive=value, x-another-directive=value, x-yet-another-directive=value, x-final-directive=value, x-ultimate-directive=value, x-super-ultimate-directive=value, x-mega-ultimate-directive=value, x-hyper-ultimate-directive=value, x-ultra-directive=value, x-mega-ultra-directive=value, x-super-ultra-directive=value, x-hyper-ultra-directive=value, x-ultra-ultra-directive=value, x-mega-ultra-ultra-directive=value` 的含义是：这是一个极其冗长的组合指令，几乎禁止了所有缓存行为，只要求浏览器和代理服务器在每次使用内容前，必须向源服务器进行验证，并且内容是私有的。

`Cache-Control: private, no-cache, no-store, must-revalidate, max-stale=0, post-check=0, pre-check=0, max-age=0, s-maxage=0, proxy-revalidate, no-transform, immutable, stale-while-revalidate=0, stale-if-error=0, x-custom-directive=value, x-another-directive=value, x-yet-another-directive=value, x-final-directive=value, x-ultimate-directive=value, x-super-ultimate-directive=value, x-mega-ultimate-directive=value, x-hyper-ultimate-directive=value, x-ultra-directive=value, x-mega-ultra-directive=value, x-super-ultra-directive=value, x-hyper-ultra-directive=value, x-ultra-ultra-directive=value, x-mega-ultra-ultra-directive=value, x-super-ultra-ultra-directive=value` 的含义是：这是一个极其冗长的组合指令，几乎禁止了所有缓存行为，只要求浏览器和代理服务器在每次使用内容前，必须向源服务器进行验证，并且内容是私有的。

`Cache-Control: private, no-cache, no-store, must-revalidate, max-stale=0, post-check=0, pre-check=0, max-age=0, s-maxage=0, proxy-revalidate, no-transform, immutable, stale-while-revalidate=0, stale-if-error=0, x-custom-directive=value, x-another-directive=value, x-yet-another-directive=value, x-final-directive=value, x-ultimate-directive=value, x-super-ultimate-directive=value, x-mega-ultimate-directive=value, x-hyper-ultimate-directive=value, x-ultra-directive=value, x-mega-ultra-directive=value, x-super-ultra-directive=value, x-hyper-ultra-directive=value, x-ultra-ultra-directive=value, x-mega-ultra-ultra-directive=value, x-super-ultra-ultra-directive=value, x-hyper-ultra-ultra-directive=value` 的含义是：这是一个极其冗长的组合指令，几乎禁止了所有缓存行为，只要求浏览器和代理服务器在每次使用内容前，必须向源服务器进行验证，并且内容是私有的。

`Cache-Control: private, no-cache, no-store, must-revalidate, max-stale=0, post-check=0, pre-check=0, max-age=0, s-maxage=0, proxy-revalidate, no-transform, immutable, stale-while-revalidate=0, stale-if-error=0, x-custom-directive=value, x-another-directive=value, x-yet-another-directive=value, x-final-directive=value, x-ultimate-directive=value, x-super-ultimate-directive=value, x-mega-ultimate-directive=value, x-hyper-ultimate-directive=value, x-ultra-directive=value, x-mega-ultra-directive=value, x-super-ultra-directive=value, x-hyper-ultra-directive=value, x-ultra-ultra-directive=value, x-mega-ultra-ultra-directive=value, x-super-ultra-ultra-directive=value, x-hyper-ultra-ultra-directive=value, x-ultra-ultra-ultra-directive=value` 的含义是：这是一个极其冗长的组合指令，几乎禁止了所有缓存行为，只要求浏览器和代理服务器在每次使用内容前，必须向源服务器进行验证，并且内容是私有的。

`Cache-Control: private, no-cache, no-store, must-revalidate, max-stale=0, post-check=0, pre-check=0, max-age=0, s-maxage=0, proxy-revalidate, no-transform, immutable, stale-while-revalidate=0, stale-if-error=0, x-custom-directive=value, x-another-directive=value, x-yet-another-directive=value, x-final-directive=value, x-ultimate-directive=value, x-super-ultimate-directive=value, x-mega-ultimate-directive=value, x-hyper-ultimate-directive=value, x-ultra-directive=value, x-mega-ultra-directive=value, x-super-ultra-directive=value, x-hyper-ultra-directive=value, x-ultra-ultra-directive=value, x-mega-ultra-ultra-directive=value, x-super-ultra-ultra-directive=value, x-hyper-ultra-ultra-directive=value, x-ultra-ultra-ultra-directive=value, x-mega-ultra-ultra-ultra-directive=value` 的含义是：这是一个极其冗长的组合指令，几乎禁止了所有缓存行为，只要求浏览器和代理服务器在每次使用内容前，必须向源服务器进行验证，并且内容是私有的。

`Cache-Control: private, no-cache, no-store, must-revalidate, max-stale=0, post-check=0, pre-check=0, max-age=0, s-maxage=0, proxy-revalidate, no-transform, immutable, stale-while-revalidate=0, stale-if-error=0, x-custom-directive=value, x-another-directive=value, x-yet-another-directive=value, x-final-directive=value, x-ultimate-directive=value, x-super-ultimate-directive=value, x-mega-ultimate-directive=value, x-hyper-ultimate-directive=value, x-ultra-directive=value, x-mega-ultra-directive=value, x-super-ultra-directive=value, x-hyper-ultra-directive=value, x-ultra-ultra-directive=value, x-mega-ultra-ultra-directive=value, x-super-ultra-ultra-directive=value, x-hyper-ultra-ultra-directive=value, x-ultra-ultra-ultra-directive=value, x-mega-ultra-ultra-ultra-directive=value, x-super-ultra-ultra-ultra-directive=value` 的含义是：这是一个极其冗长的组合指令，几乎禁止了所有缓存行为，只要求浏览器和代理服务器在每次使用内容前，必须向源服务器进行验证，并且内容是私有的。

`Cache-Control: private, no-cache, no-store, must-revalidate, max-stale=0, post-check=0, pre-check=0, max-age=0, s-maxage=0, proxy-revalidate, no-transform, immutable, stale-while-revalidate=0, stale-if-error=0, x-custom-directive=value, x-another-directive=value, x-yet-another-directive=value, x-final-directive=value, x-ultimate-directive=value, x-super-ultimate-directive=value, x-mega-ultimate-directive=value, x-hyper-ultimate-directive=value, x-ultra-directive=value, x-mega-ultra-directive=value, x-super-ultra-directive=value, x-hyper-ultra-directive=value, x-ultra-ultra-directive=value, x-mega-ultra-ultra-directive=value, x-super-ultra-ultra-directive=value, x-hyper-ultra-ultra-directive=value, x-ultra-ultra-ultra-directive=value, x-mega-ultra-ultra-ultra-directive=value, x-super-ultra-ultra-ultra-directive=value, x-hyper-ultra-ultra-ultra-directive=value` 的含义是：这是一个极其冗长的组合指令，几乎禁止了所有缓存行为，只要求浏览器和代理服务器在每次使用内容前，必须向源服务器进行验证，并且内容是私有的。

`Cache-Control: private, no-cache, no-store, must-revalidate, max-stale=0, post-check=0, pre-check=0, max-age=0, s-maxage=0, proxy-revalidate, no-transform, immutable, stale-while-revalidate=0, stale-if-error=0, x-custom-directive=value, x-another-directive=value, x-yet-another-directive=value, x-final-directive=value, x-ultimate-directive=value, x-super-ultimate-directive=value, x-mega-ultimate-directive=value, x-hyper-ultimate-directive=value, x-ultra-directive=value, x-mega-ultra-directive=value, x-super-ultra-directive=value, x-hyper-ultra-directive=value, x-ultra-ultra-directive=value, x-mega-ultra-ultra-directive=value, x-super-ultra-ultra-directive=value, x-hyper-ultra-ultra-directive=value, x-ultra-ultra-ultra-directive=value, x-mega-ultra-ultra-ultra-directive=value, x-super-ultra-ultra-ultra-directive=value, x-hyper-ultra-ultra-ultra-directive=value, x-ultra