



Cache-Control

Cache-Control 通用消息头字段，被用于在http请求和响应中，通过指定指令来实现缓存机制。缓存指令是单向的，这意味着在请求中设置的指令，不一定被包含在响应中。

Header type
General header
Forbidden header name
no
CORS-safelisted response-header
yes

语法

指令不区分大小写，并且具有可选参数，可以用令牌或者带引号的字符串语法。多个指令以逗号分隔。

缓存请求指令

客户端可以在HTTP请求中使用的标准 Cache-Control 指令。

```
Cache-Control: max-age=<seconds>  
Cache-Control: max-stale[=<seconds>]  
Cache-Control: min-fresh=<seconds>  
Cache-control: no-cache  
Cache-control: no-store  
Cache-control: no-transform  
Cache-control: only-if-cached
```

缓存响应指令

服务器可以在响应中使用的标准 **Cache-Control** 指令。

```
Cache-control: must-revalidate
Cache-control: no-cache
Cache-control: no-store
Cache-control: no-transform
Cache-control: public
Cache-control: private
Cache-control: proxy-revalidate
Cache-Control: max-age=<seconds>
Cache-control: s-maxage=<seconds>
```

扩展Cache-Control指令

拓展缓存指令不是核心HTTP缓存标准文档的一部分，使用前请注意检查[兼容性](#)！

```
Cache-control: immutable
Cache-control: stale-while-revalidate=<seconds>
Cache-control: stale-if-error=<seconds>
```

指令

可缓存性

public

表明响应可以被任何对象（包括：发送请求的客户端，代理服务器，等等）缓存，即使是通常不可缓存的内容。（例如：1.该响应没有 `max-age` 指令或 `Expires` 消息头；2.该响应对应的请求方法是 **POST** 。）

private

表明响应只能被单个用户缓存，不能作为共享缓存（即代理服务器不能缓存它）。私有缓存可以缓存响应内容，比如：对应用户的本地浏览器。

no-cache

在发布缓存副本之前，强制要求缓存把请求提交给原始服务器进行验证(协商缓存验证)。

no-store

缓存不应存储有关客户端请求或服务器响应的任何内容，即不使用任何缓存。

到期

max-age=<seconds>

设置缓存存储的最大周期，超过这个时间缓存被认为过期(单位秒)。与 `Expires` 相反，时间是相对于请求的时间。

s-maxage=<seconds>

覆盖 `max-age` 或者 `Expires` 头，但是仅适用于共享缓存(比如各个代理)，私有缓存会忽略它。

max-stale[=<seconds>]

表明客户端愿意接收一个已经过期的资源。可以设置一个可选的秒数，表示响应不能已经过时超过该给定的时间。

min-fresh=<seconds>

表示客户端希望获取一个能在指定的秒数内保持其最新状态的响应。

stale-while-revalidate=<seconds>

表明客户端愿意接受陈旧的响应，同时在后台异步检查新的响应。秒值指示客户愿意接受陈旧响应的时间长度。

stale-if-error=<seconds>

表示如果新的检查失败，则客户愿意接受陈旧的响应。秒数值表示客户在初始到期后愿意接受陈旧响应的的时间。

重新验证和重新加载

must-revalidate

一旦资源过期（比如已经超过 `max-age`），在成功向原始服务器验证之前，缓存不能用该资源响应后续请求。

proxy-revalidate

与 `must-revalidate` 作用相同，但它仅适用于共享缓存（例如代理），并被私有缓存忽略。

immutable

表示响应正文不会随时间而改变。资源（如果未过期）在服务器上不发生改变，因此客户端不应发送重新验证请求头（例如 `If-None-Match` 或 `If-Modified-Since`）来检查更新，即使用户显式地刷新页面。在 Firefox 中，`immutable` 只能被用在 `https://transactions`。有关更多信息，请参阅[这里](#)。

其他

no-transform

不得对资源进行转换或转变。`Content-Encoding`、`Content-Range`、`Content-Type` 等HTTP头不能由代理修改。例如，非透明代理或者如[Google's Light Mode](#)可能对图像格式进行转换，以便节省缓存空间或者减少缓慢链路上的流量。`no-transform` 指令不允许这样做。

only-if-cached

表明客户端只接受已缓存的响应，并且不要向原始服务器检查是否有更新的拷贝。

示例

禁止缓存

发送如下响应头可以关闭缓存。此外，可以参考 `Expires` 和 `Pragma` 消息头。

```
Cache-Control: no-store
```

缓存静态资源

对于应用程序中不会改变的文件，你通常可以在发送响应头前添加积极缓存。这包括例如由应用程序提供的静态文件，例如图像，CSS文件和JavaScript文件。另请参阅[Expires](#)标题。

```
Cache-Control: public, max-age=31536000
```

需要重新验证

指定 `no-cache` 或 `max-age=0` 表示客户端可以缓存资源，每次使用缓存资源前都必须重新验证其有效性。这意味着每次都会发起 HTTP 请求，但当缓存内容仍有效时可以跳过 HTTP 响应体的下载。

```
Cache-Control: no-cache  
Cache-Control: max-age=0
```

规范

Specification

Title

[RFC 7234](#)

Hypertext Transfer Protocol (HTTP/1.1): Caching

[RFC 5861](#)

HTTP Cache-Control Extensions for Stale Content

浏览器兼容性

[Report problems with this compatibility data on GitHub](#)

Cache-Control	
Chrome	Yes
Edge	12
Firefox	Yes
Internet Explorer	Yes
Opera	Yes
Safari	Yes
WebView Android	Yes
Chrome Android	Yes
Firefox Android	Yes
Opera Android	Yes
iOS Safari	Yes
Samsung Internet	Yes
immutable	
Chrome	No
Edge	15 — 79
Firefox	49
Internet Explorer	No
Opera	No
Safari	11
WebView Android	No
Chrome Android	No
Firefox Android	No

Opera Android	No
iOS Safari	11
Samsung Internet	No
stale-if-error	
Chrome	No
Edge	No
Firefox	No
Internet Explorer	No
Opera	No
Safari	No
WebView Android	No
Chrome Android	No
Firefox Android	No
Opera Android	No
iOS Safari	No
Samsung Internet	No
stale-while-revalidate	
Chrome	75
Edge	79
Firefox	68
Internet Explorer	No
Opera	No
Safari	No
WebView Android	75
Chrome Android	75
Firefox Android	68

Opera Android	No
iOS Safari	No
Samsung Internet	No

Full support

No support

Experimental. Expect behavior to change in the future.

Non-standard. Expect poor cross-browser support.

See implementation notes.

参见

- [HTTP Caching FAQ](#)
- [Caching Tutorial for Web Authors and Webmasters](#)
- Guide: *Cache-Control for civilians*
- [Age](#)
- [Expires](#)
- [Pragma](#)

Last modified: 2020年10月16日, by [MDN contributors](#)