链滴

Http 缓存 4：论某一资源被缓存和使用缓存的条件

作者：Lord-X

原文链接：https://ld246.com/article/1587990330589

来源网站：链滴

<img src="https://ld246.com/images/img-loading.svg" alt="" data-src="https://b3logfile
com/bing/20181204.jpg?imageView2/1/w/960/h/540/interlace/1/q/100">
<h2 id="Http缓存-论某一资源被缓存和使用缓存的条件">Http 缓存：论某一资源被缓存和使用缓
的条件</h2>
某一请求使用缓存的基本步骤有两个，一个是先将响应的资源缓存下来，另一个是在发起请求时
足使用缓存的条件。下面先来看看什么样的响应会被缓存下来。
<h3 id="什么样的响应会被缓存">什么样的响应会被缓存</h3>
这里引用 RFC7234 原文做简要说明（原文：page6，第三章节）
A cache MUST NOT store a response to any request, unless:

The request method is understood by the cache and defined as being
cacheable, and
the response status code is understood by the cache, and
the "no-store" cache directive (see Section 5.2) does not appear
in request or response header fields, and
the "private" response directive (see Section 5.2.2.6) does not
appear in the response, if the cache is shared, and
the Authorization header field (see Section 4.2 of [RFC7235]) does
not appear in the request, if the cache is shared, unless the
response explicitly allows it (see Section 3.2), and
the response either:

contains an Expires header field (see Section 5.3), or
contains a max-age response directive (see Section 5.2.2.8), or
contains a s-maxage response directive (see Section 5.2.2.9)
and the cache is shared, or
contains a Cache Control Extension (see Section 5.2.3) that
allows it to be cached, or
has a status code that is defined as cacheable by default (see
Section 4.2.2), or
contains a public response directive (see Section 5.2.2.5).

大概解释一下：
一个响应满足以下条件，即可被缓存：

请求的方法必须能被缓存理解，而且必须是可以被缓存的方法。例如 GET、HEAD 方法可以被缓
，POST 和 PATCH 如果设置了合适的 Header(Content-Location)也可以被缓存，但 PUT 和 DELETE
方法不可被缓存。
响应码可以被缓存理解，以下这些响应码是可以被缓存的：200、203、204、206、300、301、
04、405、410、414、501。
请求头和响应头中都没有指定 "no-store" 头部
对于共享缓存来说（代理服务器），在响应头中没有指定为 "private"
对于共享缓存来说（代理服务器），请求中没有 "Authorization" 头部
响应头中含有 "Expires"、"max-age"、"s-maxage"、"public"，或通过 "Cache Control Extens
on" 明确指明要缓存，或返回的响应码指明要缓存时

<h3 id="如何命中缓存">如何命中缓存</h3>
同样引用 RFC7234 原文做简要说明（原文：page8，第四章节）
When presented with a request, a cache MUST NOT reuse a stored
response, unless:

The presented effective request URI (Section 5.5 of [RFC7230]) and

that of the stored response match, and
the request method associated with the stored response allows it
to be used for the presented request, and
selecting header fields nominated by the stored response (if any)
match those presented (see Section 4.1), and
the presented request does not contain the no-cache pragma
(Section 5.4), nor the no-cache cache directive (Section 5.2.1),
unless the stored response is successfully validated
(Section 4.3), and
the stored response does not contain the no-cache cache directive
(Section 5.2.2.2), unless it is successfully validated
(Section 4.3), and
the stored response is either:

fresh (see Section 4.2), or
allowed to be served stale (see Section 4.2.4), or
successfully validated (see Section 4.3).

大概解释一下：
发起请求时，满足一下条件即可使用缓存：

URI 匹配，如果一个 URI 有多个缓存，则使用时间最近的
缓存的响应允许我们当前请求的 METHOD 使用缓存
头部匹配，指缓存的响应中，Vary 指定的头部必须与本次请求的头部匹配
提交的请求不包含 "no-cache" 头部（Pragma 和 Cache-Control 都不可包含）
缓存的响应不包含 "no-cache" 头部（Pragma 和 Cache-Control 都不可包含）
缓存未过期，或允许使用过期缓存(max-stale)，或针对过期缓存已经到源服务器成功验证(源服器响应 304)

<h3 id="Varying-Response">Varying Response</h3>
这里对命中缓存时提到的头部匹配做一个介绍。<code>Vary</code> 是 Http 响应头的一个 Hader，他决定了一个请求在命中缓存时的 Header 匹配规则。当缓存的响应中指定了 Vary 时，新的求必须满足 Vary 指定的所有 Header 规则才可使用缓存。例如：
<img src="https://ld246.com/images/img-loading.svg" alt="HttpCache8.png" data-src="ttps://b3logfile.com/file/2020/04/HttpCache8-6beb8091.png?imageView2/2/interlace/1/format/jpg">
上图中的这个 js 资源 Response 中的 Vary 指明，想要使用我这个缓存，必须验证 <code>Accpt-Encoding</code> 头的值，再看 Request Header 中，请求的 <code>Accept-Encoding</coe> 头的值为 <code>gzip, deflate</code>。所以，新的请求想要命中这个缓存，也必须带这个头值能够匹配才行。
再来看一个来自 Mozilla 官网的例子。
<img src="https://ld246.com/images/img-loading.svg" alt="HttpCache9.png" data-src="ttps://b3logfile.com/file/2020/04/HttpCache9-74385c05.png?imageView2/2/interlace/1/format/jpg">
Client1 向代理服务器 Cache 发起/doc 的请求，Cache 发现没有缓存，向源服务器 Server 请，Server 将结果响应给 Cache，并带有一个 Vary 头部，指定要校验 <code>Content-Encoding</ode> 头部，值为 gzip 的请求才能使用这份共享缓存。然后再响应给 Client1。
Client2 向代理服务器 Cache 发起/doc 请求，并带有 <code>Accept-Encoding</code> 头值为 br。Cache 服务器跟本地缓存校对，发现 Vary 校验不通过，因此不能使用共享缓存，再次向源务器发起请求，此时源服务器会返回响应，并将请求的 br 加到限制条件中。再返回给 Client2。
Client3 向代理服务器 Cache 发起/doc 请求，并带有 <code>Accept-Encoding</code> 头值为 br。Cache 服务器跟本地缓存校对，发现 Vary 中指定的值包含 br，校验通过，直接返回共享

存，不再请求源服务器。</p>
<h2 id="参考">参考</h2>

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