

实际的区别，这只是意味着到了要进行核对的时间了。这种情况被称为“服务器再验证”，说明缓存需要询问原始服务器文档是否发生了变化。

- 如果再验证显示内容发生了变化，缓存会获取一份新的文档副本，并将其存储在旧文档的位置上，然后将文档发送给客户端。
- 如果再验证显示内容没有发生变化，缓存只需要获取新的首部，包括一个新的过期日期，并对缓存中的首部进行更新就行了。

这是个很棒的系统。缓存并不一定要为每条请求验证文档的有效性——只有在文档过期时它才需要与服务器进行再验证。这样不会提供陈旧的内容，还可以节省服务器的流量，并拥有更好的用户响应时间。

HTTP 协议要求行为正确的缓存返回下列内容之一：

- “足够新鲜”的已缓存副本；
- 与服务器进行过再验证，确认其仍然新鲜的已缓存副本；
- 如果需要与之进行再验证的原始服务器出故障了，就返回一条错误报文¹⁴；
- 附有警告信息说明内容可能不正确的已缓存副本。

7.8.4 用条件方法进行再验证

HTTP 的条件方法可以高效地实现再验证。HTTP 允许缓存向原始服务器发送一个“条件 GET”，请求服务器只有在文档与缓存中现有的副本不同时，才回送对象主体。通过这种方式，将新鲜度检测和对象获取结合成了单个条件 GET。向 GET 请求报文中添加一些特殊的条件首部，就可以发起条件 GET。只有条件为真时，Web 服务器才会返回对象。

HTTP 定义了 5 个条件请求首部。对缓存再验证来说最有用的 2 个首部是 If-Modified-Since 和 If-None-Match。¹⁵ 所有的条件首部都以前缀“If-”开头。表 7-3 列出了在缓存再验证中使用的条件请求首部。

注 14：如果原始服务器不可访问，但缓存需要进行再验证，那么缓存就必须返回一条错误或一条用来描述通信故障的警告报文。否则，来自已移除服务器上的页面未来可能会在网络的缓存中存留任意长的时间。

注 15：其他条件首部包括 If-Unmodified-Since（在进行部分文件的传输时，获取文件的其余部分之前要确保文件未发生变化，此时这个首部是非常有用的）、If-Range（支持对不完整文档的缓存）和 If-Match（用于与 Web 服务器打交道时的并发控制）。