

表7-3 缓存再验证中使用的两个条件首部

| 首 部 | 描 述 |
|--------------------------|---|
| If-Modified-Since:<date> | 如果从指定日期之后文档被修改过了，就执行请求的方法。可以与 Last-Modified 服务器响应首部配合使用，只有在内容被修改后与已缓存版本有所不同时才去获取内容 |
| If-None-Match:<tags> | 服务器可以为文档提供特殊的标签（参见 ETag），而不是将其与最近修改日期相匹配，这些标签就像序列号一样。如果已缓存标签与服务器文档中的标签有所不同，If-None-Match 首部就会执行所请求的方法 |

7.8.5 If-Modified-Since:Date再验证

最常见的缓存再验证首部是 If-Modified-Since。If-Modified-Since 再验证请求通常被称为 IMS 请求。只有自某个日期之后资源发生了变化的时候，IMS 请求才会指示服务器执行请求：

- 如果自指定日期后，文档被修改了，If-Modified-Since 条件就为真，通常 GET 就会成功执行。携带新首部的新文档会被返回给缓存，新首部除了其他信息之外，还包含了一个新的过期日期。
- 如果自指定日期后，文档没被修改过，条件就为假，会向客户端返回一个小的 304 Not Modified 响应报文，为了提高有效性，不会返回文档的主体。¹⁶ 这些首部是放在响应中返回的，但只会返回那些需要在源端更新的首部。比如，Content-Type 首部通常不会被修改，所以通常不需要发送。一般会发送一个新的过期日期。

If-Modified-Since 首部可以与 Last-Modified 服务器响应首部配合工作。原始服务器会将最后的修改日期附加到所提供的文档上去。当缓存要对已缓存文档进行再验证时，就会包含一个 If-Modified-Since 首部，其中携带有最后修改已缓存副本的日期：

```
If-Modified-Since: <cached last-modified date>
```

如果在此期间内容被修改了，最后的修改日期就会有所不同，原始服务器就会回送新的文档。否则，服务器会注意到缓存的最后修改日期与服务器文档当前的最后修改日期相符，会返回一个 304 Not Modified 响应。

注 16：如果有一个不认识 If-Modified-Since 首部的老服务器收到了条件请求，它会将其作为一个普通的 GET 解释。在这种情况下，系统仍然能够工作，但由于要对未修改的文档数据进行不必要的传输，所以效率会比较低。