

如果 HTTP/1.1 缓存或服务器收到的请求既带有 If-Modified-Since, 又带有实体标签条件首部, 那么只有这两个条件都满足时, 才能返回 304 Not Modified 响应。

## 7.9 控制缓存的能力

服务器可以通过 HTTP 定义的几种方式来指定在文档过期之前可以将其缓存多长时间。按照优先级递减的顺序, 服务器可以:

- 附加一个 Cache-Control: no-store 首部到响应中去;
- 附加一个 Cache-Control: no-cache 首部到响应中去;
- 附加一个 Cache-Control: must-revalidate 首部到响应中去;
- 附加一个 Cache-Control: max-age 首部到响应中去;
- 附加一个 Expires 日期首部到响应中去;
- 不附加过期信息, 让缓存确定自己的过期日期。

本节描述了缓存控制首部。下一节, 也就是 7.10 节介绍了如何为不同的内容分配不同的缓存信息。

### 7.9.1 no-Store与no-Cache响应首部

HTTP/1.1 提供了几种限制对象缓存, 或限制提供已缓存对象的方式, 以维持对象的新鲜度。no-store 首部和 no-cache 首部可以防止缓存提供未经证实的已缓存对象:

```
Pragma: no-cache
Cache-Control: no-store
Cache-Control: no-cache
```

标识为 no-store 的响应会禁止缓存对响应进行复制。缓存通常会像非缓存代理服务器一样, 向客户端转发一条 no-store 响应, 然后删除对象。

标识为 no-cache 的响应实际上是可以存储在本地缓存区中的。只是在与原始服务器进行新鲜度再验证之前, 缓存不能将其提供给客户端使用。这个首部使用 do-not-serve-from-cache-without-revalidation 这个名字会更恰当一些。

HTTP/1.1 中提供 Pragma: no-cache 首部<sup>17</sup>是为了兼容于 HTTP/1.0+。除了与只理解 Pragma: no-cache 的 HTTP/1.0 应用程序进行交互时, HTTP 1.1 应用程序都应该使用 Cache-Control: no-cache。

182

注 17: 从技术上来讲, Pragma: no-cache 首部只能用于 HTTP 请求, 但在实际中它作为扩展首部已被广泛地用于 HTTP 请求和响应之中。