

## 11.6.9 cookie与缓存

缓存那些与 cookie 事务有关的文档时要特别小心。你不会希望给用户分配一个过去某些用户用过的 cookie，或者更糟糕的是，向一个用户展示其他人私有文档的内容。

cookie 和缓存的规则并没有很好地建立起来。下面是处理缓存时的一些指导性规则。

- 如果无法缓存文档，要将其标示出来

文档的所有者最清楚文档是否是不可缓存的。如果文档不可缓存，就显式地注明——具体来说，如果除了 Set-Cookie 首部之外文档是可缓存的，就使用 Cache-Control: no-cache="Set-Cookie"。另一种更通用的做法是为可缓存文档使用 Cache-Control: public，这样有助于节省 Web 中的带宽。

- 缓存 Set-Cookie 首部时要小心

如果响应中有 Set-Cookie 首部，就可以对主体进行缓存（除非被告知不要这么做），但要特别注意对 Set-Cookie 首部的缓存。如果向多个用户发送了相同的 Set-Cookie 首部，可能会破坏用户的定位。

有些缓存在将响应缓存起来之前会删除 Set-Cookie 首部，但这样也会引发一些问题，因为在没有缓存的时候，通常都会有 cookie 贴在客户端上，但由缓存提供服务的客户端就不会有 cookie 了。强制缓存与原始服务器重新验证每条请求，并将返回的所有 Set-Cookie 首部都合并到客户端的响应中去，就可以改善这种状况。原始服务器可以通过向缓存的副本中添加这个首部来要求进行这种再验证：

```
Cache-Control: must-revalidate, max-age=0
```

即便内容实际上是可以缓存的，比较保守的缓存可能也会拒绝缓存所有包含 Set-Cookie 首部的响应。有些缓存允许使用缓存 Set-Cookie 图片，但不缓存文本的模式。

- 小心处理带有 Cookie 首部的请求

带有 Cookie 首部的请求到达时，就在提示我们，得到的结果可能是私有的。一定要将私有内容标识为不可缓存的，但有些服务器可能会犯错，没有将此内容标识为不可缓存的。

有些响应文档对应于携带 Cookie 首部的请求，保守的缓存可能会选择不缓存这些响应文档。同样，有些缓存允许使用缓存 cookie 图片，而不缓存文本的模式。得到更广泛接受的策略是缓存带有 Cookie 首部的图片，将过期时间设置为零，强制每次都进行再验证。

273  
|  
274