

## 7.9.2 max-age响应首部

Cache-Control: max-age 首部表示的是从服务器将文档传来之时起，可以认为此文档处于新鲜状态的秒数。还有一个 s-maxage 首部（注意 maxage 的中间没有连字符），其行为与 max-age 类似，但仅适用于共享（公有）缓存：

```
Cache-Control: max-age=3600
Cache-Control: s-maxage=3600
```

服务器可以请求缓存不要缓存文档，或者将最大使用期设置为零，从而在每次访问的时候都进行刷新：

```
Cache-Control: max-age=0
Cache-Control: s-maxage=0
```

## 7.9.3 Expires响应首部

不推荐使用 Expires 首部，它指定的是实际的过期日期而不是秒数。HTTP 设计者后来认为，由于很多服务器的时钟都不同步，或者不正确，所以最好还是用剩余秒数，而不是绝对时间来表示过期时间。可以通过计算过期值和日期值之间的秒数差来计算类似的新鲜生存期：

```
Expires: Fri, 05 Jul 2002, 05:00:00 GMT
```

有些服务器还会回送一个 Expires:0 响应首部，试图将文档置于永远过期的状态，但这种语法是非法的，可能给某些软件带来问题。应该试着支持这种结构的输入，但不应该产生这种结构的输出。

## 7.9.4 must-revalidate响应首部

可以配置缓存，使其提供一些陈旧（过期）的对象，以提高性能。如果原始服务器希望缓存严格遵守过期信息，可以在原始响应中附加一个 Cache-Control: must-revalidate 首部。

```
Cache-Control: must-revalidate
```

Cache-Control: must-revalidate 响应首部告诉缓存，在事先没有跟原始服务器进行再验证的情况下，不能提供这个对象的陈旧副本。缓存仍然可以随意提供新鲜的副本。如果在缓存进行 must-revalidate 新鲜度检查时，原始服务器不可用，缓存就必须返回一条 504 Gateway Timeout 错误。