

状态码	原因短语	含 义
302	Found	与 301 状态码类似；但是，客户端应该使用 Location 首部给出的 URL 来临时定位资源。将来的请求仍应使用老的 URL
303	See Other	告知客户端应该用另一个 URL 来获取资源。新的 URL 位于响应报文的 Location 首部。其主要目的是允许 POST 请求的响应将客户端定向到某个资源上去
304	Not Modified	客户端可以通过所包含的请求首部，使其请求变成有条件的。更多有关条件首部的内容请参见第 3 章。如果客户端发起了一个条件 GET 请求，而最近资源未被修改的话，就可以用这个状态码来说明资源未被修改。带有这个状态码的响应不应该包含实体的主体部分
305	Use Proxy	用来说明必须通过一个代理来访问资源，代理的位置由 Location 首部给出。很重要的一点是，客户端是相对某个特定资源来解析这条响应的，不能假定所有请求，甚至所有对持有所请求资源的服务器的请求都通过这个代理进行。如果客户端错误地让代理介入了某条请求，可能会引发破坏性的行为，而且会造成安全漏洞
306	(未使用)	当前未使用
307	Temporary Redirect	与 301 状态码类似；但客户端应该使用 Location 首部给出的 URL 来临时定位资源。将来的请求应该使用老的 URL

从表 3-8 中，你可能已经注意到 302、303 和 307 状态码之间存在一些交叉。这些状态码的用法有着细微的差别，大部分差别都源于 HTTP/1.0 和 HTTP/1.1 应用程序对这些状态码处理方式的不同。

当 HTTP/1.0 客户端发起一个 POST 请求，并在响应中收到 302 重定向状态码时，它会接受 Location 首部的重定向 URL，并向那个 URL 发起一个 GET 请求（而不会像原始请求中那样发起 POST 请求）。

HTTP/1.0 服务器希望 HTTP/1.0 客户端这么做——如果 HTTP/1.0 服务器收到来自 HTTP/1.0 客户端的 POST 请求之后发送了 302 状态码，服务器就期望客户端能够接受重定向 URL，并向重定向的 URL 发送一个 GET 请求。

问题出在 HTTP/1.1。HTTP/1.1 规范使用 303 状态码来实现同样的行为（服务器发送 303 状态码来重定向客户端的 POST 请求，在它后面跟上一个 GET 请求）。

为了避开这个问题，HTTP/1.1 规范指出，对于 HTTP/1.1 客户端，用 307 状态码取代 302 状态码来进行临时重定向。这样服务器就可以将 302 状态码保留起来，为 HTTP/1.0 客户端使用了。