

3.4 状态码

如前面的表 3-2 所示，HTTP 状态码被分成了五大类。本节对这五类 HTTP 状态码中的每一类都进行了总结。

状态码为客户端提供了一种理解事务处理结果的便捷方式。尽管并没有实际的规范对原因短语的确切文本进行说明，本节还是列出了一些原因短语示例。我们所列的是 HTTP/1.1 规范推荐使用的原因短语。

3.4.1 100 ~ 199——信息性状态码

HTTP/1.1 向协议中引入了信息性状态码。这些状态码相对较新，由于对其复杂性和感知价值存在一些争论，而受到限制。表 3-6 列出了已定义的信息性状态码。

表3-6 信息性状态码及原因短语

状 态 码	原因短语	含 义
100	Continue	说明收到了请求的初始部分，请客户端继续。发送了这个状态码之后，服务器在收到请求之后必须进行响应。更多信息请参见附录 C 中的 Expect 首部介绍
101	Switching Protocols	说明服务器正在根据客户端的指定，将协议切换成 Update 首部所列的协议

100 Continue 状态码尤其让人糊涂。它的目的是对这样的情况进行优化：HTTP 客户端应用程序有一个实体的主体部分要发送给服务器，但希望在发送之前查看一下服务器是否会接受这个实体。这可能会给 HTTP 程序员带来一些困扰，因此在这里进行了比较详细（它如何与客户端、服务器和代理进行通信）的讨论。

1. 客户端与 100 Continue

如果客户端在向服务器发送一个实体，并且愿意在发送实体之前等待 100 Continue 响应，那么，客户端就要发送一个携带了值为 100 Continue 的 Expect 请求首部（参见附录 C）。如果客户端没有发送实体，就不应该发送 100 Continue Expect 首部，因为这样会使服务器误以为客户端要发送一个实体。

从很多方面来看，100 Continue 都是一种优化。客户端应用程序只有在避免向服务器发送一个服务器无法处理或使用的大实体时，才应该使用 100 Continue。

由于起初对 100 Continue 状态存在一些困惑（而且以前有些实现在这里出过问题），因此发送了值为 100 Continue 的 Expect 首部的客户端不应该永远在那儿等待服务器发送 100 Continue 响应。超时一定时间之后，客户端应该直接将实体发送出去。