

- (1) 如果特定的 HTTP 报文类型中不允许带有主体，就忽略 Content-Length 首部，它是对（没有实际发送出来的）主体进行计算的。这种情况下，Content-Length 首部是提示性的，并不说明实际的主体长度。（考虑不周的 HTTP 应用程序会认为有了 Content-Length 就有主体存在，这样就会出问题。）

最重要的例子就是 HEAD 响应。HEAD 方法请求服务器发送等价的 GET 请求中会出现的首部，但不要包括主体。因为对 GET 的响应会带有 Content-Length 首部，所以 HEAD 响应里面也有；但和 GET 响应不同的是，HEAD 响应中不会有主体。1XX、204 以及 304 响应也可以有提示性的 Content-Length 首部，但是也都没有实体主体。那些规定不能带有实体主体的报文，不管带有什么首部字段，都必须在首部之后的第一个空行终止。

- (2) 如果报文中含有描述传输编码的 Transfer-Encoding 首部（不采用默认的 HTTP “恒等” 编码），那实体就应由一个称为“零字节块”（zero-byte chunk）的特殊模式结束，除非报文已经因连接关闭而结束。我们将在本章后面讨论传输编码和分块编码。
- (3) 如果报文中含有 Content-Length 首部（并且报文类型允许有实体主体），而且没有非恒等的 Transfer-Encoding 首部字段，那么 Content-Length 的值就是主体的长度。如果收到的报文中既有 Content-Length 首部字段又有非恒等的 Transfer-Encoding 首部字段，那就必须忽略 Content-Length，因为传输编码会改变实体主体的表示和传输方式（因此可能就会改变传输的字节数）。
- (4) 如果报文使用了 multipart/byteranges（多部分 / 字节范围）媒体类型，并且没有用 Content-Length 首部指出实体主体的长度，那么多部分报文中的每个部分都要说明它自己的大小。这种多部分类型是唯一的一种自定界的实体主体类型，因此除非发送方知道接收方可以解析它，否则就不能发送这种媒体类型。³
- (5) 如果上面的规则都不匹配，实体就在连接关闭的时候结束。实际上，只有服务器可以使用连接关闭来指示报文的结束。客户端不能用关闭连接来指示客户端报文的结束，因为这样会使服务器无法发回响应。⁴

346

为了和使用 HTTP/1.0 的应用程序兼容，任何带有实体主体的 HTTP/1.1 请求都必须带有正确的 Content-Length 首部字段（除非已经知道服务器兼容 HTTP/1.1）。

注 3：因为 Range 首部可能会被不理解多部分 / 字节范围的更原始的代理所转发，所以如果发送方不能确定接收方是否理解这种自定界的格式的话，就必须用本节的方法（1）、（3）或（5）来对报文定界。

注 4：客户端可以使用半关闭，也就是只把连接的输出端关闭，但很多服务器应用程序设计的时候没有考虑到处理这种情况，会把半关闭当作客户端要从服务器断开连接来处理。HTTP 没有对连接管理进行良好的规范。详情请参见第 4 章。