**一、实体首部**

HTTP实体首部描述了HTTP报名的内容。HTTP/1.1版定义了以下10个基本字体首部字段。

* Content-Type：实体中所承载对象的类型
* Content-Length：所传送实体的长度或大小
* Content-Language：与所传送对象最相配的人类语言
* Content-Encoding：对象数据所做的任意变换（例如压缩）
* Content-Location：一个备用位置，请求时可通过它获得对象
* Content-Range：如果这是部分实体，这个首部说明它是整体的哪个部分
* Content-MD5：实体主体内容和检验和
* Last-Modified：所传输内容在服务器上创建或最后修改日期时间
* Expires：实体数据将要失效的日期时间
* Allow：该资源所允许的各种请求方法，例如GET和HEAD
* ETag：文档特定的唯一验证码
* Cache-Control：指出应该如何缓存该文档

**二、内容编码**

内容编码的过程如下：

1、网站服务器生成原始响应报文，其中有原始的Content-Type和Content-Length首部

2、内容编码服务器（也可能就是原始的服务器或下行的代理）创建编码后的报文。编码后的报文有同样的Content-Type，但Content-Length可能不同（例如主体被压缩）。内容编码服务器在编码后的报文中增加Content-Encoding首部，这样接收的应用程序就可以进行解码了。

3、接收程序得到编码后的报文，进行解码，获得原始报文。

常用的内容编码类型有：gzip（效率最高）、compress、deflate、identity（前三者是无损压缩，不会导致信息损失）

同时，**为了避免服务器使用客户端不支持的编码方式，客户端就把自己支持的内容编码方式列表放在请求的Accept-Encoding首部里发出去。如果客户端HTTP请求没有Accept-Encoding首部，那么服务器可以假设客户端可以接受任何编码方式（等价于发送Accept-Encoding：\*）**

**客户端可以给每种编码附带Q（质量）值参数来说明编码的优先级。Q值范围从0.0到1.0，0.1说明客户端不想接受所说明的编码，1.0则表示最希望使用的编码。\*表示任何其它方法。**

**三、传输编码和分块编码**

HTTP协议中只定义了下面两个首部来描述和控制传输编码。

Transfer-Encoding：告知接收方为了可靠地传输报文，已经对其进行了何种编码。

TE：用在请求首部中，告知服务器可以使用哪些传输编码扩展。

最新的HTTP规范只定义了一种传输编码：分块编码

**分块编码**

分块编码把报文分割为若干个大小已知的块。块之间是紧挨着发送的，这样就不需要在发送之前知道整个报文的大小了。它是一种传输编码，因此是报文的属性，而不是文体的属性。

分块报文的拖挂：拖挂中可以包含附带的首部字段，它们的值在报文开始的时候可能是无法确定的（例如必须要先生成主体的内容）。Content-MD5首部就是一个可以在拖挂中发送的首部，因为在文档生成之前，很难算出它的MD5。

**四、范围请求**

假设你正通过慢速的调制解调器下载最新的热门电影，已经下载了四分之三，忽然因为一个网络故障连接中断了而要被迫全部重头再来有多么糟糕。

有了范围请求，HTTP客户端可以通过请求曾获取失败的实体的一个范围（一部分）来恢复下载该实体。当然这有一个前提，那就是客户端上一次请求该实体到这次发出范围请求的时间段内，该对象没有改变过。例如：Range:bytes=4000- （从4000字节之后的部分开始）

并不是所有的服务器都接受范围请求，但很多服务器可以。**服务器可以通过在响应中包含Accept-Ranges首部的形式向客户端说明可以接受的范围请求，这个首部通常以字节计算的。**

Range首部流行在点对点（Peer to Peer）文件共享客户端软件中得到广泛应用，它们从不同的对等实体同时下载多媒体文件的不同部分。

**四、差异编码**

如果服务器有页面更新的实例，当客户端请求该过期的副本时，哪怕页面上只有一小部分发生了改变，也要把完整的新页面实例发给客户端。若改变的地方比较少，与其发送完整的新页面给客户端，客户端更愿意服务器只发送页面发生改变的部分，这样就可以更快地得到最新的页面。差异编码是HTTP协议的一个扩展，它通过交换对象改变的部分而不是完整的对象来优化传输性能。

**如果客户端想告诉服务器它愿意接受该页面的差异，只要发送A－IM首部就可以了。A－IM是Accept-Instance-Manipulation（接受实例操控）的缩写。服务端发送回下面这些内容：一个特殊的响应代码－－226 IM Used，告知客户端它正在发送的是所请求对象的实例操控，而不是那个完整的对象自身；一个IM首部，说明用于计算差异的算法，新的ETag首部和Delta-Base首部，说明用于计算差异的基线文档的ETag，整个过程由实例操控、差异生成器、差异应用器共同完成。**

ETag：由服务器在响应中发送，客户端在后继请求中可以使用它。

If-None-Match：客户端发送请求首部，当且仅当客户端的文档版本与服务器不同时，才向服务器请求该文档。

A－IM：客户端请求首部，说明可以接受的实例操控类型

IM：服务器响应首部，说明作用在响应上的实例操控的类型。当响应代码是226 IM Used时，会发送这个首部。

Delta-Base：服务器响应首部，说明用于计算差异的基线文档的ETag值（应该与客户端请求中的If-None-Match首部里的ETag相同）