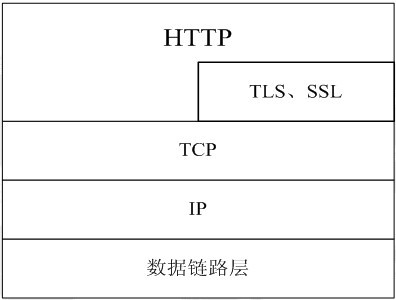
# HTTP学习笔记

## 基本概念

HTTP是Hyper Text Transfer Protocol（超文本传输协议）的缩写。RFC 2616定义了今天普遍使用的一个版本——HTTP 1.1。

HTTP是一个应用层协议，由请求和响应构成，是一个标准的客户端服务器模型。HTTP是一个无状态的协议。

### 在TCP/IP协议栈中的位置

HTTP协议通常承载于TCP协议之上，有时也承载于TLS或SSL协议层之上，这个时候，就成了我们常说的HTTPS。如下图所示：  
   

默认HTTP的端口号为80，HTTPS的端口号为443。

### HTTP的请求响应模型

HTTP协议永远都是客户端发起请求，服务器回送响应。见下图： ****

这样就限制了使用HTTP协议，无法实现在客户端没有发起请求的时候，服务器将消息推送给客户端。

HTTP协议是一个无状态的协议，同一个客户端的这次请求和上次请求是没有对应关系。

### **工作流程**

一次HTTP操作称为一个事务，其工作过程可分为四步：

1）首先客户机与服务器需要建立连接。只要单击某个超级链接，HTTP的工作开始。

2）建立连接后，客户机发送一个请求给服务器，请求方式的格式为：统一资源标识符（URL）、协议版本号，后边是MIME信息包括请求修饰符、客户机信息和可能的内容。

3）服务器接到请求后，给予相应的响应信息，其格式为一个状态行，包括信息的协议版本号、一个成功或错误的代码，后边是MIME信息包括服务器信息、实体信息和可能的内容。

4）客户端接收服务器所返回的信息通过浏览器显示在用户的显示屏上，然后客户机与服务器断开连接。

如果在以上过程中的某一步出现错误，那么产生错误的信息将返回到客户端，有显示屏输出。对于用户来说，这些过程是由HTTP自己完成的，用户只要用鼠标点击，等待信息显示就可以了。

## http报文结构组成

### 报文的组成部分：

所有的http报文都可以分为两类：请求报文和响应报文，请求报文会向服务器请求一个动作，响应报文会将请求结果返回给客户端。

每个报文由三部分组成：1、对报文进行描述的起始行（star line）、包含属性的首部（header）块、以及可选的包含数据的主体（body）部分。

起始行和首部就是由行分隔的ASCLL文本。每行以一个回车符和一个换行符作为行终止序列作为结束。

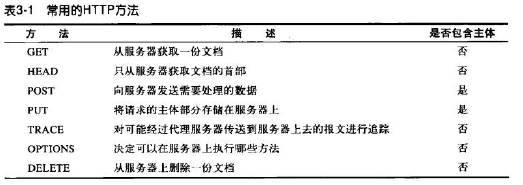
### 请求报文





#### 请求行

##### 请求方法



###### 安全方法：

GET和HEAD，即不会对服务器产生任何结果。

###### GET：

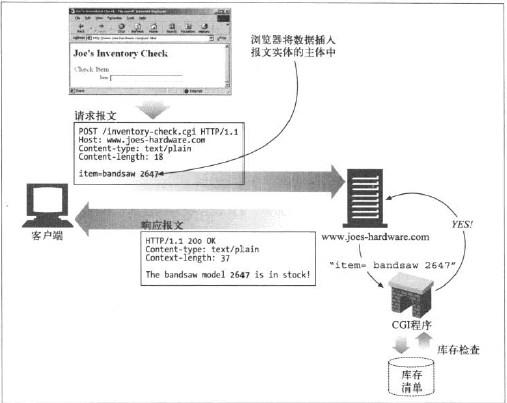
请求指定的页面信息，并返回实体主体。

###### HEAD：

只请求页面的首部。

###### POST：

向服务器输入数据，通常用来支持html表单数据发送。

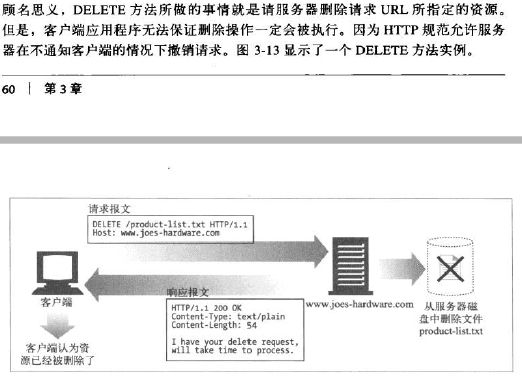


###### PUT：

从客户端向服务器传送的数据创建或取代指定的文档的内容。

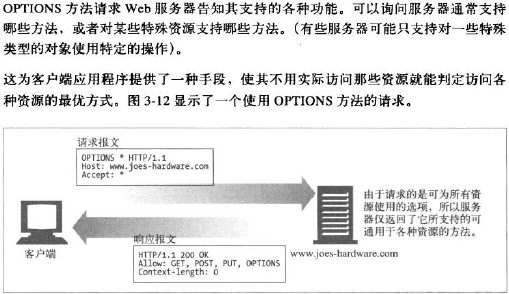
###### DELETE：

请求服务器删除指定的资源，不保证一定被执行，http规范运行服务器在不通知客户端情况下撤销请求。



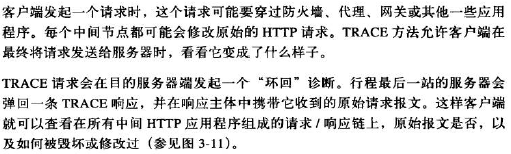
###### OPTIONS：

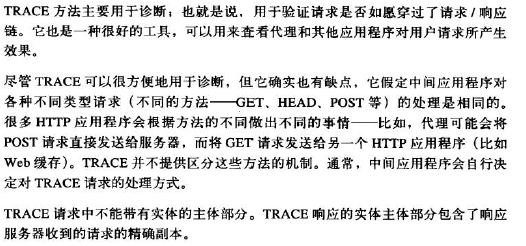
允许客户端查看服务器的支持的各种功能。

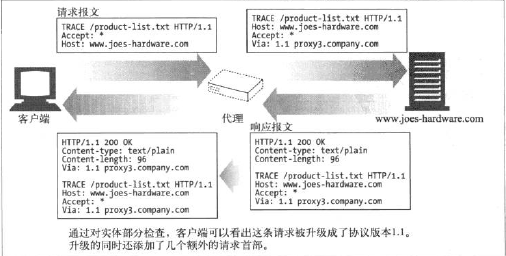


###### TRACE：

用于诊断请求过程中http请求的变化，服务器的trace响应中包含它接收到的原始请求报文。缺点：假定服务器对各种类型请求处理方式相同，真实情况可能不同。不能带有主体部分。

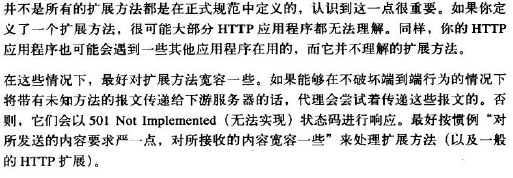






###### 拓展方法

http被设计为字段可拓展的。拓展方法的意思是在HTTP/1.1中没被定义的办法。



MOVE：

请求服务器移动资源

COPY：

请求服务器复制资源

LOCK：

允许用户“锁定”资源——例如在编辑某个资源时候锁定，防止别人同时进行修改。

MKCOL：

允许用户创建资源

##### URL地址

**统一资源定位符**（或称**统一资源定位器**/**定位地址**、**URL地址**等[[1]](https://zh.wikipedia.org/wiki/%E7%BB%9F%E4%B8%80%E8%B5%84%E6%BA%90%E5%AE%9A%E4%BD%8D%E7%AC%A6#cite_note-1)，英语：Uniform / Universal Resource Locator，常缩写为**URL**），有时也被俗称为**网页地址**（**网址**）。

##### HTTP协议及版本

超文本传输协议已经演化出了很多版本，它们中的大部分都是[向下兼容](https://zh.wikipedia.org/wiki/%E5%90%91%E4%B8%8B%E5%85%BC%E5%AE%B9)的。在 [RFC 2145](https://tools.ietf.org/html/rfc2145) 中描述了HTTP版本号的用法。客户端在请求的开始告诉服务器它采用的协议版本号，而后者则在响应中采用相同或者更早的协议版本。

###### 0.9

已过时。只接受GET一种请求方法，没有在通讯中指定版本号，且不支持请求头。由于该版本不支持POST方法，因此客户端无法向服务器传递太多信息。

###### HTTP/1.0

这是第一个在通讯中指定版本号的HTTP协议版本，至今仍被广泛采用，特别是在[代理服务器](https://zh.wikipedia.org/wiki/%E4%BB%A3%E7%90%86%E6%9C%8D%E5%8A%A1%E5%99%A8)中。

###### HTTP/1.1

持久连接被默认采用，并能很好地配合代理服务器工作。还支持以[管道方式](https://zh.wikipedia.org/wiki/HTTP%E7%AE%A1%E7%BA%BF%E5%8C%96)在同时发送多个请求，以便降低线路负载，提高传输速度。

HTTP/1.1相较于HTTP/1.0协议的区别主要体现在：

* 缓存处理
* 带宽优化及网络连接的使用
* 错误通知的管理
* 消息在网络中的发送
* 互联网地址的维护
* 安全性及完整性

###### HTTP/2

主条目：[HTTP/2](https://zh.wikipedia.org/wiki/HTTP/2)

当前版本，于2015年5月作为互联网标准正式发布。[[4]](https://zh.wikipedia.org/wiki/%E8%B6%85%E6%96%87%E6%9C%AC%E4%BC%A0%E8%BE%93%E5%8D%8F%E8%AE%AE#cite_note-4)

### 响应报文



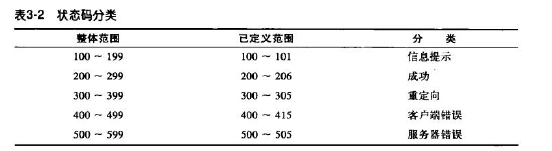
#### 响应行

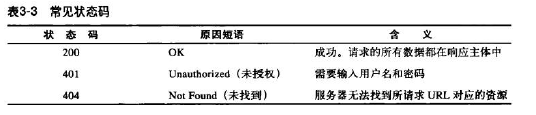
响应报文的起始行，或称为响应行，包含响应报文使用的HTTP版本、数字状态码、以及描述操作状态的文本形式的原因短语。

#### 状态码

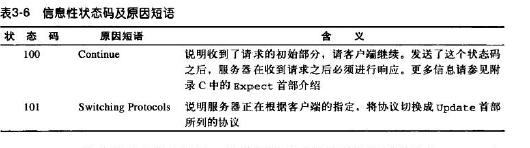
方法用来告诉服务器做什么事

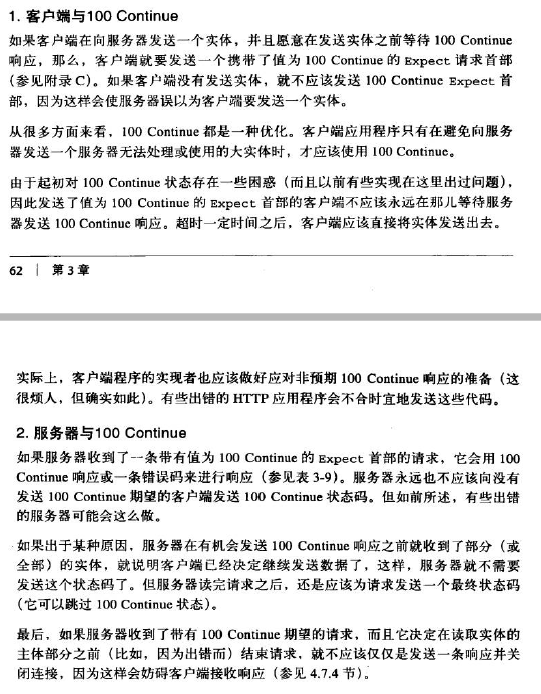
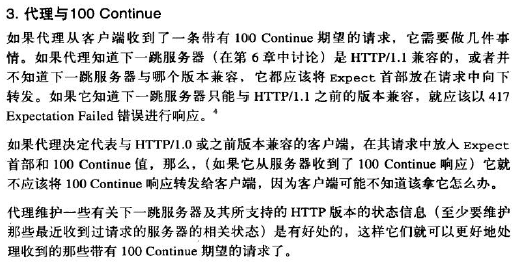
状态码用来告诉客户端发生了什么事情。



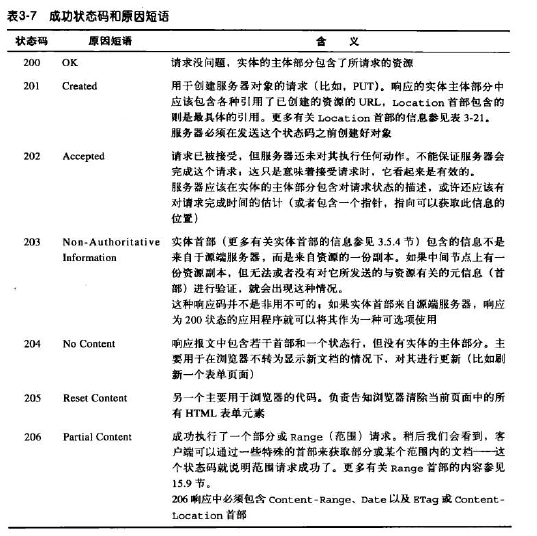


##### 100~199——信息性状态码

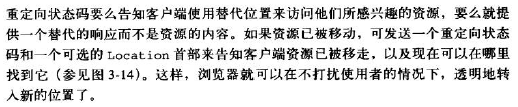


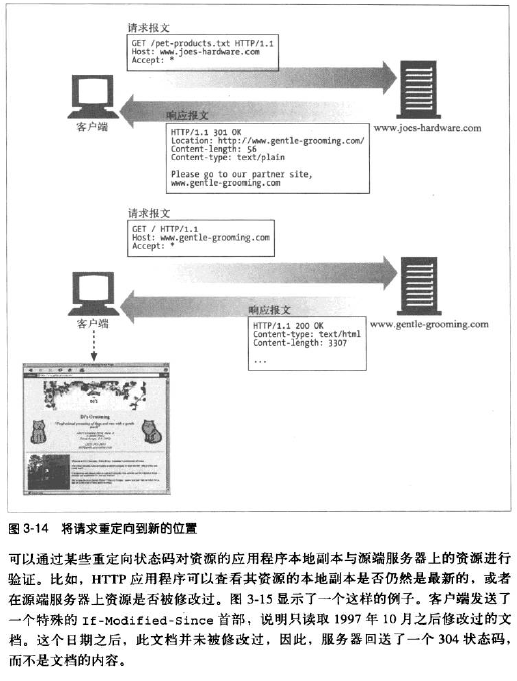
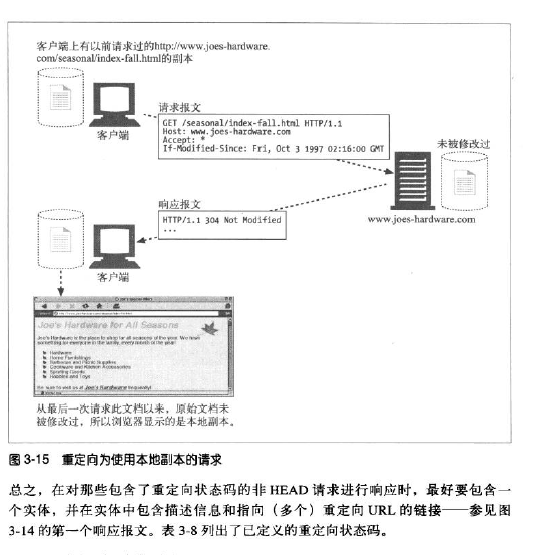
 

##### 200~299——成功状态码

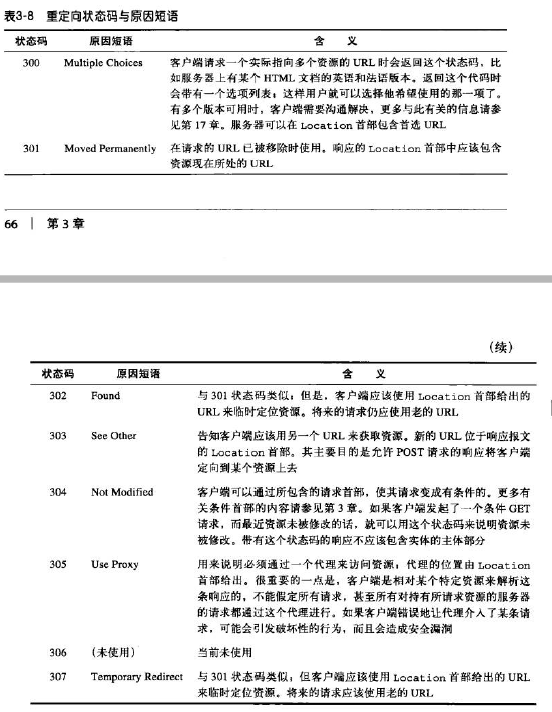


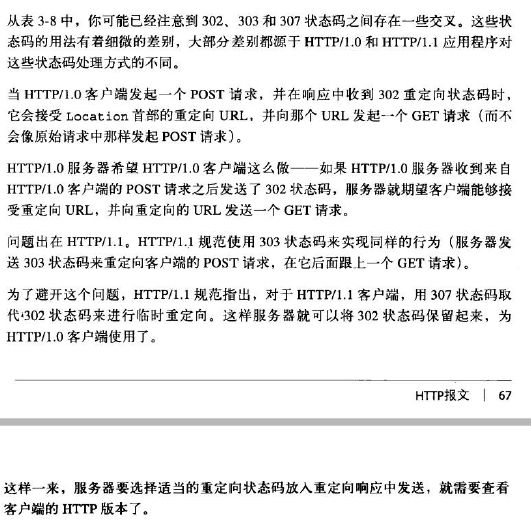
##### 300~399——重定向状态码



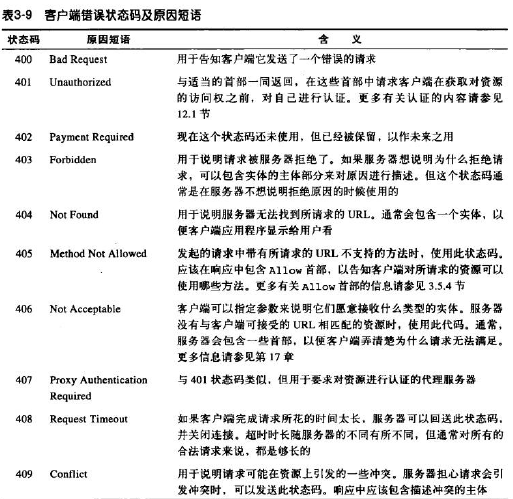
 

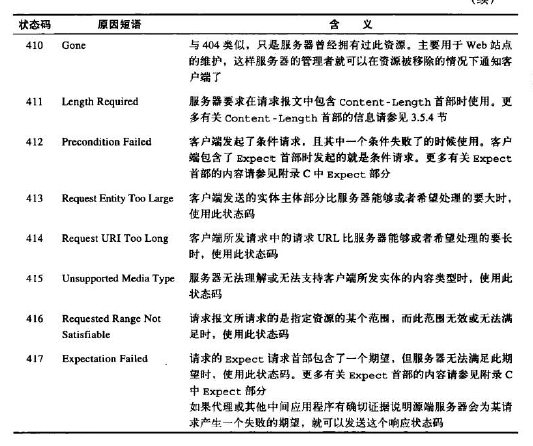
用于判断源本地副本和服务器上的资源相比是否最新，方便显示。除了head方法请求，最好给重定向实体信息，方便客户端查验。



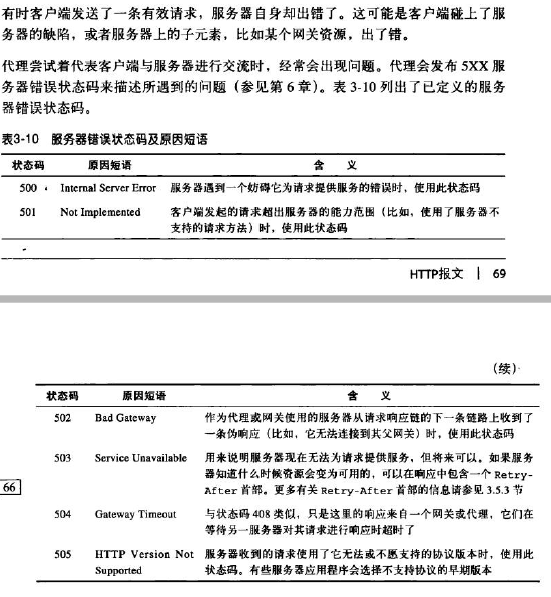


##### 400~499——客户端错误状态码





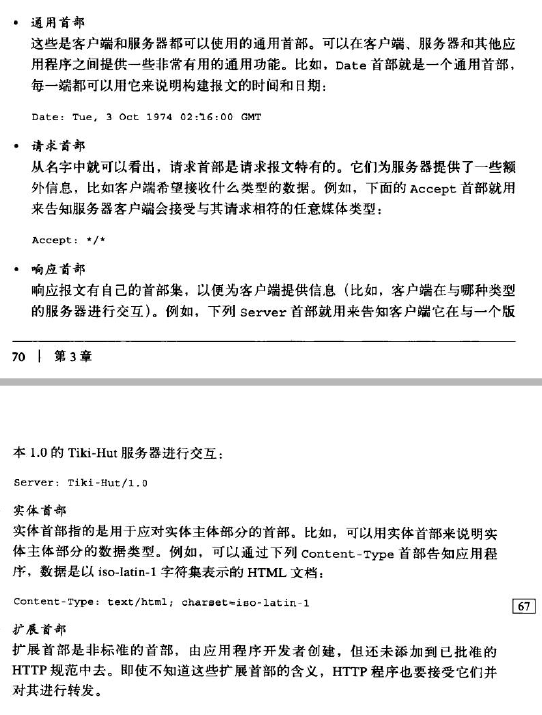
##### 500~599服务器错误状态码



### 首部信息

HTTP首部字段向请求和响应报文中添加了一些附加信息。本质上来说是一些名/值对的列表。

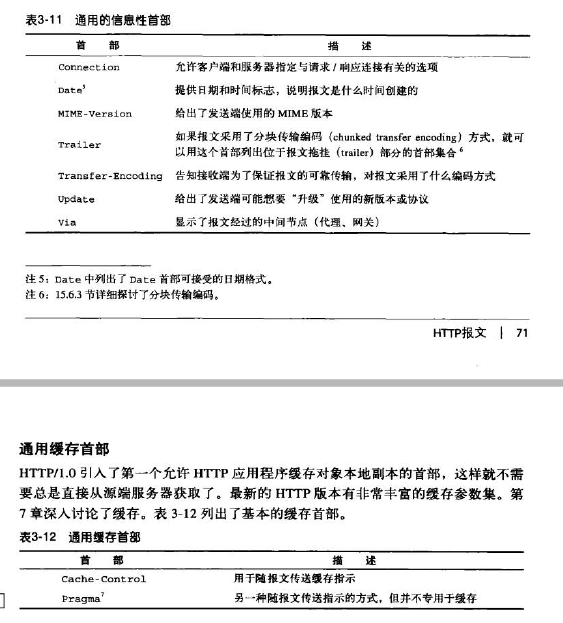
#### 首部分类：



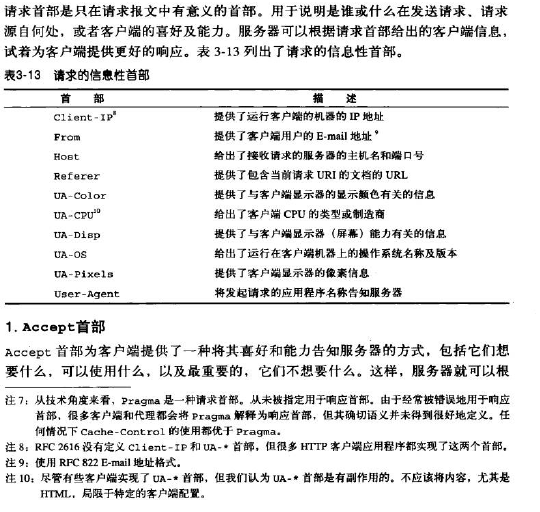
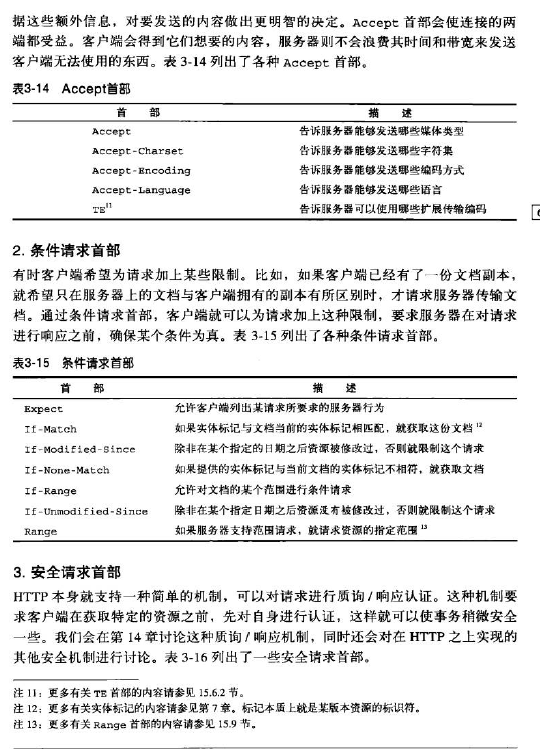
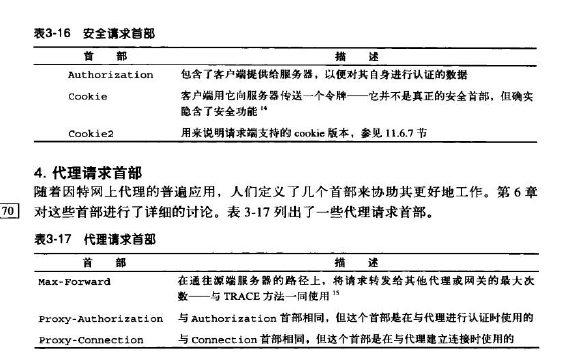
#### 通用首部

**多用途互联网邮件扩展**（**MIME**，**Multipurpose Internet Mail Extensions**）是一个[互联网标准](https://zh.wikipedia.org/wiki/%E4%BA%92%E8%81%94%E7%BD%91%E6%A0%87%E5%87%86)，它扩展了[电子邮件](https://zh.wikipedia.org/wiki/%E7%94%B5%E5%AD%90%E9%82%AE%E4%BB%B6)标准，使其能够支援：

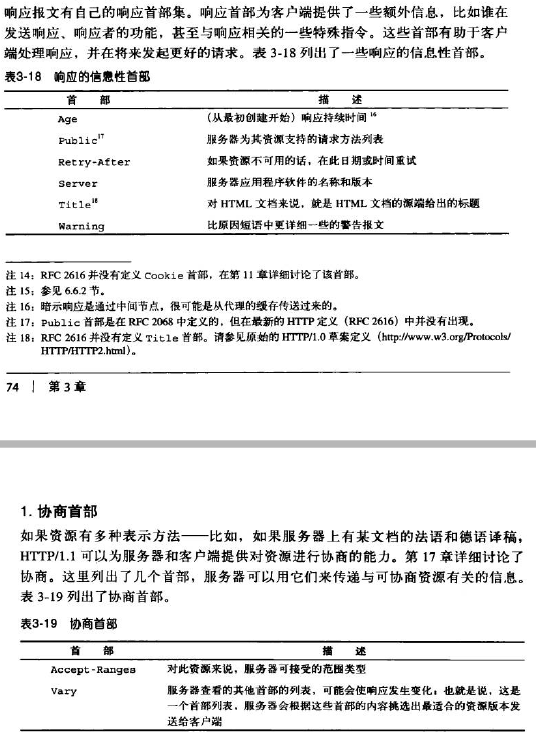
* 非[ASCII](https://zh.wikipedia.org/wiki/ASCII)字符文本；
* 非文本格式附件（[二进制](https://zh.wikipedia.org/wiki/%E4%BA%8C%E8%BF%9B%E5%88%B6)、声音、图像等）；
* 由多部分（multiple parts）组成的消息体；
* 包含非[ASCII](https://zh.wikipedia.org/wiki/ASCII)字符的头信息（Header information

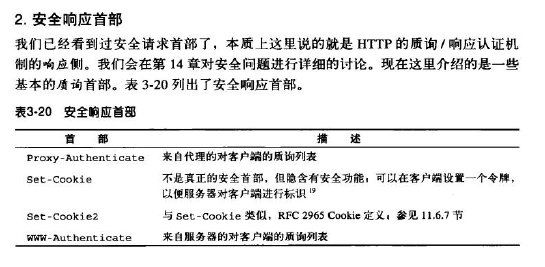


#### 请求首部

#### 响应头部





#### 实体首部

