HTTP请求头概述 （HttpServletRequest） HTTP客户程序（例如浏览器），向服务器发送请求的时候必须指明请求类型（一般是GET或者POST或者HEAD）。   
  
　　如有必要，客户程序还可以选择发送其他的请求头。大多数请求头并不是必需的，但Content-Length除外。对于POST请求来说Content-Length必须出现。 下面是一些最常见的请HTTP请求头概述 （HttpServletRequest）   
  
　　HTTP客户程序（例如浏览器），向服务器发送请求的时候必须指明请求类型（一般是GET或者POST）。如有必要，客户程序还可以选择发送其他的请求头。大多数请求头并不是必需的，但Content-Length除外。对于POST请求来说Content-Length必须出现。   
  
　　下面是一些最常见的请求头   
  
　　Accept：浏览器可接受的MIME类型。   
  
　　Accept-Charset：浏览器可接受的字符集。   
  
　　Accept-Encoding：浏览器能够进行解码的数据编码方式，比如gzip。Servlet能够向支持gzip的浏览器返回经gzip编码的HTML页面。许多情形下这可以减少5到10倍的下载时间。   
  
　　Accept-Language：浏览器所希望的语言种类，当服务器能够提供一种以上的语言版本时要用到。   
  
　　Authorization：授权信息，通常出现在对服务器发送的WWW-Authenticate头的应答中。   
  
　　Connection：表示是否需要持久连接。如果Servlet看到这里的值为“Keep-Alive”，或者看到请求使用的是HTTP 1.1（HTTP 1.1默认进行持久连接），它就可以利用持久连接的优点，当页面包含多个元素时（例如Applet，图片），显著地减少下载所需要的时间。要实现这一点，Servlet需要在应答中发送一个Content-Length头，最简单的实现方法是：先把内容写入ByteArrayOutputStream，然后在正式写出内容之前计算它的大小。   
  
　　Content-Length：表示请求消息正文的长度。   
  
　　Cookie：这是最重要的请求头信息之一   
  
　　From：请求发送者的email地址，由一些特殊的Web客户程序使用，浏览器不会用到它。   
  
　　Host：初始URL中的主机和端口。   
  
　　If-Modified-Since：只有当所请求的内容在指定的日期之后又经过修改才返回它，否则返回304“Not Modified”应答。   
  
　　Pragma：指定“no-cache”值表示服务器必须返回一个刷新后的文档，即使它是代理服务器而且已经有了页面的本地拷贝。   
  
　　Referer：包含一个URL，用户从该URL代表的页面出发访问当前请求的页面。   
  
　　User-Agent：浏览器类型，如果Servlet返回的内容与浏览器类型有关则该值非常有用。   
  
　　UA-Pixels，UA-Color，UA-OS，UA-CPU：由某些版本的IE浏览器所发送的非标准的请求头，表示屏幕大小、颜色深度、操作系统和CPU类型。   
  
　　HTTP应答头概述（HttpServletResponse）   
  
　　Web服务器的HTTP应答一般由以下几项构成：一个状态行，一个或多个应答头，一个空行，内容文档。设置HTTP应答头往往和设置状态行中的状态代码结合起来。例如，有好几个表示“文档位置已经改变”的状态代码都伴随着一个Location头，而401（Unauthorized）状态代码则必须伴随一个WWW-Authenticate头。   
  
　　然而，即使在没有设置特殊含义的状态代码时，指定应答头也是很有用的。应答头可以用来完成：设置Cookie，指定修改日期，指示浏览器按照指定的间隔刷新页面，声明文档的长度以便利用持久HTTP连接，……等等许多其他任务。   
  
　　设置应答头最常用的方法是HttpServletResponse的setHeader，该方法有两个参数，分别表示应答头的名字和值。和设置状态代码相似，设置应答头应该在发送任何文档内容之前进行。   
  
　　setDateHeader方法和setIntHeadr方法专门用来设置包含日期和整数值的应答头，前者避免了把Java时间转换为GMT时间字符串的麻烦，后者则避免了把整数转换为字符串的麻烦。   
  
　　HttpServletResponse还提供了许多设置   
  
　　setContentType：设置Content-Type头。大多数Servlet都要用到这个方法。   
  
　　setContentLength：设置Content-Length头。对于支持持久HTTP连接的浏览器来说，这个函数是很有用的。   
  
　　addCookie：设置一个Cookie（Servlet API中没有setCookie方法，因为应答往往包含多个Set-Cookie头）。   
  
　　另外，如上节介绍，sendRedirect方法设置状态代码302时也会设置Location头。   
  
　　HTTP应答头 说明   
  
　　Allow 服务器支持哪些请求方法（如GET、POST等）。   
  
　　Content-Encoding 文档的编码（Encode）方法。只有在解码之后才可以得到Content-Type头指定的内容类型。利用gzip压缩文档能够显著地减少HTML文档的下载时间。Java的GZIPOutputStream可以很方便地进行gzip压缩，但只有Unix上的Netscape和Windows上的IE 4、IE 5才支持它。因此，Servlet应该通过查看Accept-Encoding头（即request.getHeader("Accept-Encoding")）检查浏览器是否支持gzip，为支持gzip的浏览器返回经gzip压缩的HTML页面，为其他浏览器返回普通页面。   
  
　　Content-Length 表示内容长度。只有当浏览器使用持久HTTP连接时才需要这个数据。如果你想要利用持久连接的优势，可以把输出文档写入ByteArrayOutputStram，完成后查看其大小，然后把该值放入Content-Length头，最后通过byteArrayStream.writeTo(response.getOutputStream()发送内容。   
  
　　Content-Type 表示后面的文档属于什么MIME类型。Servlet默认为text/plain，但通常需要显式地指定为text/html。由于经常要设置Content-Type，因此HttpServletResponse提供了一个专用的方法setContentTyep。   
  
　　Date 当前的GMT时间。你可以用setDateHeader来设置这个头以避免转换时间格式的麻烦。   
  
　　Expires 应该在什么时候认为文档已经过期，从而不再缓存它？   
  
　　Last-Modified 文档的最后改动时间。客户可以通过If-Modified-Since请求头提供一个日期，该请求将被视为一个条件GET，只有改动时间迟于指定时间的文档才会返回，否则返回一个304（Not Modified）状态。Last-Modified也可用setDateHeader方法来设置。   
  
　　Location 表示客户应当到哪里去提取文档。Location通常不是直接设置的，而是通过HttpServletResponse的sendRedirect方法，该方法同时设置状态代码为302。   
  
　　Refresh 表示浏览器应该在多少时间之后刷新文档，以秒计。除了刷新当前文档之外，你还可以通过setHeader("Refresh", "5; URL=http://host/path")让浏览器读取指定的页面。注意这种功能通常是通过设置HTML页面HEAD区的<META HTTP-EQUIV="Refresh" CONTENT="5;URL=http://host/path">实现，这是因为，自动刷新或重定向对于那些不能使用CGI或Servlet的HTML编写者十分重要。但是，对于Servlet来说，直接设置Refresh头更加方便。注意Refresh的意义是“N秒之后刷新本页面或访问指定页面”，而不是“每隔N秒刷新本页面或访问指定页面”。因此，连续刷新要求每次都发送一个Refresh头，而发送204状态代码则可以阻止浏览器继续刷新，不管是使用Refresh头还是<META HTTP-EQUIV="Refresh" ...>。注意Refresh头不属于HTTP 1.1正式规范的一部分，而是一个扩展，但Netscape和IE都支持它。   
  
　　Server 服务器名字。Servlet一般不设置这个值，而是由Web服务器自己设置。   
  
　　Set-Cookie 设置和页面关联的Cookie。Servlet不应使用response.setHeader("Set-Cookie", ...)，而是应使用HttpServletResponse提供的专用方法addCookie。参见下文有关Cookie设置的讨论。   
  
　　WWW-Authenticate 客户应该在Authorization头中提供什么类型的授权信息？在包含401（Unauthorized）状态行的应答中这个头是必需的。例如，response.setHeader("WWW-Authenticate", "BASIC realm=\"executives\"")。注意Servlet一般不进行这方面的处理，而是让Web服务器的专门机制来控制受密码保护页面的访问（例如.htaccess）。