21世纪高等学校规划教材 | 计算机科学与技术

Java Web开发教程
（第2版）

孙霞 主编
党海峰 冯筠 贺小伟 副主编

清华大学出版社

内 容 简 介

本书是关于 Java Web 开发的入门教材。全书内容由浅入深，首先回顾 Web 开发技术的发展，讲解 HTTP、HTML 等 Web 系统开发技术基础知识，然后讲解 Servlet 和 JSP 等开发的相关技术，最后引入了 MVC 设计模式的理念，详细讲述一个完整实际的 Java Web 开发项目，逐步引领读者从基础到各个知识点的学习，帮助读者较为全面地掌握 Java Web 开发技术。

本书适合作为高校计算机以及信息管理等相关专业在校大学生的 Java Web 开发课程的教材，也可以作为 Java Web 初学者的参考书目，还可供社会 Java Web 技术培训班作为教材使用。

图书在版编目(CIP)数据

Java Web 开发教程/孙霞主编. —2 版. —北京：清华大学出版社，2012.9
21 世纪高等学校规划教材·计算机科学与技术
ISBN 978-7-302-29111-4

Ⅰ. ①J… Ⅱ. ①孙… Ⅲ. ①JAVA 语言—程序设计—高等学校—教材 Ⅳ. ①TP312

中国版本图书馆 CIP 数据核字(2012)第 131029 号

责任编辑：郑寅堃　薛　阳
封面设计：傅瑞学
责任校对：白　蕾
责任印制：何　芊

出版发行：清华大学出版社
网　　址：http://www.tup.com.cn，http://www.wqbook.com
地　　址：北京清华大学学研大厦 A 座　　邮　　编：100084
社 总 机：010-62770175　　邮　　购：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质 量 反 馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
课 件 下 载：http://www.tup.com.cn，010-62795954

印 装 者：三河市李旗庄少明印装厂
经　　销：全国新华书店
开　　本：185mm×260mm　　**印　张**：20.75　　**字　数**：504 千字
版　　次：2008 年 10 月第 1 版　2012 年 9 月第 2 版　　**印　次**：2012 年 9 月第 1 次印刷
印　　数：1～3000
定　　价：34.50 元

产品编号：041385-01

前言

1. 编写目的

Java Web 在整个 Web 开发领域占有重要地位。目前许多大型企业和公司都将 Java Web 作为首选的 Web 应用开发技术。然而很多高校课程中却缺少这门课程的身影，大多数高校只开设“Java 程序设计”课程，这使得培养出的学生与社会脱节，需要企业和公司花大量时间和精力培养毕业生的 Web 开发能力。为此我们决定编写一部 Java Web 开发的教材。

2. 章节介绍

在本教材中，从最基础的概念入手，循序渐进地将 Java Web 开发中的每个技术点展现在学习者面前，力求让学习者掌握 Java Web 开发的基础概念及技术要点。本书第 1 章为 Java Web 开发基础，从 Web 开发的基础网络技术 HTTP 谈起，分别介绍了 Web 基础技术和 Java Web 开发基础技术。第 2 章和第 3 章分别介绍了 Tomcat 工具和 Eclipse 相关知识。对 Tomcat 的介绍具体包括 Tomcat 服务器结构、Tomcat 配置和 Tomcat 中的 Web 应用。对 Eclipse 的介绍具体包括 Eclipse 体系结构、Eclipse 常用配置以及如何使用 Eclipse 开发 Java Web 应用。第 4 章主要介绍了两个简单的 Java Web 开发实例，通过这两个简单实例揭开开发 Web 应用的神秘面纱，让读者对 Web 开发有一个感性认识。第 5 章和第 6 章介绍了 Java Web 开发中最为关键的两个技术——Servlet 技术和 JSP 技术，包括 Servlet 关键概念、JSP 基本语法以及隐含对象。第 7 章介绍了 JDBC 技术，主要包括 JDBC 体系结构、4 种驱动程序和常用的 API，然后详细介绍了数据库连接过程和简单的数据库操作，最后介绍了数据库连接池技术。第 7 章还给出了两个具体连接数据库的实例，帮助读者理解 JDBC 关键技术点。第 8 章实现一个较为完整的、实际的 Java Web 开发项目——皮影制作网站，采用 MVC 开发理念将本书介绍的技术串联起来，给读者一个 Java Web 开发的全景图。最后，在本书附录中给出了开发程序过程中经常容易出现的各种错误，并针对不同的错误类型，总结了不同的应对措施和解决方法。

3. 本书特点

本教材最大特色是根据所讲解知识的特点采用不同的组织和描述方式：对于 JSP、Servlet 等技术性较强的内容，从技术的深层次实现细节进行剖析，深入浅出，使读者更加容易理解；对于 Eclipse、Tomcat 等工具性知识，采用工具书式的组织方式，使读者更加容易查阅。

本教材另一个特色是通过项目实践性的案例教学，化繁为简、化难为易、深入浅出地介

绍基本概念和理论。本书始终以皮影制作网站项目实例贯穿始末,从最简单的静态网页——皮影制作网站Logo展示,到采用JDBC技术实现皮影制作网站用户注册与登录,再到完整的皮影制作网站实现,这些实例的讲解无疑能够帮助读者更好地理解书中所讲的技术难点。该实例是来自于西北大学的大学生国家创新项目"基于互联网的交互式皮影的制作与表演系统"。随书的配套电子课件中含有全书所有项目案例的源代码,供读者学习参考使用,所有程序均通过了实际环境中的运行和测试。

本教材其他特色体现在:①每章开头都有导论,让零基础的读者了解相关概念,顺利入门。每章结尾都有总结,用适合初学者学习的语言总结本章重点内容。②代码解析。将范例代码中的关键代码行逐一解释,有助于读者掌握相关概念和知识。

4. 适用范围

本书既适合作为高校计算机以及信息管理等相关专业在校大学生的Java Web开发课程的教材,也可以作为Java Web开发初学者的参考书目,还可供社会Java Web技术培训班作为教材使用。

本书主编为西北大学信息科学与技术学院孙霞,副主编为IBM西安研发中心党海峰、西北大学信息科学与技术学院冯筠、贺小伟。研究生牛维、钟杰 、李倩以及王浩参与本书部分代码的撰写。吴云峰、郑盼盼、王小东、赵晓龙、杨小敏、赵翊凯、李耀琳、崔磊、仝鑫龙、林永锋和安苏阳参与校对工作。

由于作者水平有限,书中的疏漏和不妥之处在所难免,敬请读者批评指正。

5. 再版说明

本书与第一版《贯通Java Web开发三剑客:Eclipse+Tomcat+Ant整合开发》在编排上和内容上有了很大区别。经过第二版的修订,本书更适合高等学校本科生和初学者学习。第二版强调Java Web基础的技术,从基本的概念入手,循序渐进地将Java Web开发中的每个技术点展现在学习者面前,并且本书始终以我们承担的国家大学生创新项目为教学案例,化繁为简、化难为易、深入浅出地介绍本书涵盖的技术难点。第二版补充了Java Web开发中最常用到的JDBC技术,删除了第一版中高级应用部分,使初学者不会一开始就陷入纷繁复杂的Java Web技术中,望而却步,继而放弃Java Web技术的学习。

编 者

2012年7月

目 录

第1章 Java Web开发基础

【本章导读】

Web开发，顾名思义就是开发应用于Web之上的系统。而随着Internet成为现今覆盖面最大和应用最广泛的网络，Web开发技术也主要集中在开发基于Internet的网络应用系统。Internet是一系列网络结构和网络协议等网络技术的集合，这些技术也是基于Internet的Web应用的基础，了解基本的Internet网络技术对于深入理解Web开发技术是不可或缺的。本章简单介绍一些Internet中常用的而且也是与Web开发紧密相关的网络基础知识，包括HTTP、HTML。除此之外，本章在回顾了Web开发技术发展历史之后，简单介绍了Java Web开发中常见的技术：Servlet技术、JSP技术、JavaBean技术以及JDBC技术，便于读者对Java Web的开发有一个整体初步的认识。

1.1 Web应用概述

典型的Web应用是B/S模式(浏览器/服务器模式)，即Internet上的两台主机，一台充当服务器，另一台充当客户机，客户机通过本机的浏览器与服务器进行通信，如图1.1所示。

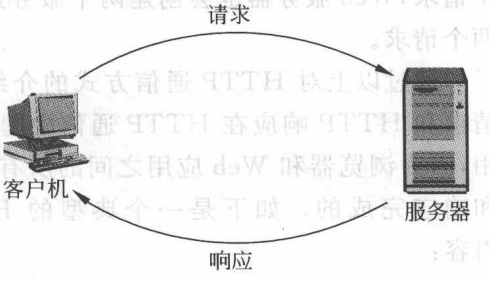

图1.1 B/S模式示意图

下面以访问西北大学主页为例，阐述客户机与服务器之间的通信过程。读者在浏览器中输入西北大学的主页地址www.nwu.edu.cn，按Enter键后浏览器就会向西北大学的服务器发送一个请求并且将自己的IP地址连同请求一块发送，该请求要求浏览西北大学的主页；西北大学的服务器接收到该请求后取出客户机的IP地址，以客户机的IP地址作为目的地址，将西北大学的主页内容作为数据包发出；当数据包传送到客户机后，读者的浏览器就可以显示西北大学的主页了。上例中西北大学的Web服务器处理客户机响应的程序就是一个典型的Web应用，是运行在服务器上的一段程序。接收请求、分析请求、构造响应、发送响应都是由该Web应用完成的，这几项工作也是大多数Web应用的主要工作。客户机上的浏览器是客户端的应用程序，从上例中可以看出，浏览器的主要功能有如下几项。

(1) 用户通过在浏览器的地址栏输入地址向服务器发送请求。

(2) 建立与服务器的连接,把用户在客户端输入的信息提交到服务器。

(3) 接收从服务器传递回来的信息。

(4) 解析并显示从服务器返回的内容。

1.2 HTTP协议

从1.1节中可以看出,Web应用的核心就是分析请求、完成相应动作并构造响应。而这其中的分析请求和构造响应都是与Internet的一种传输协议——HTTP——紧密相关,它规定了Web应用中的数据在网络中的传输方式和传输格式。目前使用的HTTP协议是HTTP 1.1版。

HTTP的全称是HyperText Transfer Protocal,即超文本传输协议。它是Internet的应用层协议,它定义了客户机的浏览器与服务器的Web应用之间如何进行通信,以及通信时用于传递数据的数据包的格式等内容。HTTP协议是采用请求/响应模式的无状态协议。客户机浏览器和服务器Web应用采用HTTP协议进行通信时,通信由浏览器发起。浏览器向Web应用发送一个请求,Web应用接收并处理该请求,然后向浏览器发回响应。在请求/响应过程中,Web应用不保存与任何一个客户机通信的状态,它只对当前到来的请求进行处理,处理完后返回对应于该请求的响应。任何两个请求的处理都是独立的,无论这两个请求是来自同一个客户机还是不同的客户机。

如图1.2所示为Web服务器同时响应多个客户机浏览器请求的示意图。当同时有多个客户机向同一个Web应用发出请求时,Web服务器就为每一个请求创建一个服务进程/线程用以处理这一请求;即使是同一个客户机发送的两个请求,Web服务器也会创建两个服务进程/线程用于处理两个请求。

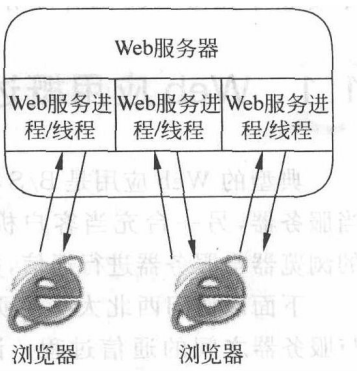

图1.2 Web服务器与客户浏览器交互示意图

通过以上对HTTP通信方式的介绍可以发现,HTTP请求和HTTP响应在HTTP通信中起到了至关重要的作用,因为浏览器和Web应用之间的所有通信都是依靠请求和响应完成的。如下是一个典型的HTTP请求消息的内容:

```
GET / HTTP/1.1
Accept: image/gif, image/x - xbitmap, image/jpeg, image/pjpeg, application/x - shockwave -
flash, application/vnd.ms - excel, application/vnd.ms - powerpoint, application/msword, * / *
Accept - Language: zh - cn
Accept - Encoding: gzip, deflate
User - Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1)
Host: www.nwu.edu.cn
...
```

该消息用于请求http://www.nwu.edu.cn的主页。如下是对该请求的响应消息(HTML页面内容部分用"…"省略):

```
HTTP/1.1 200 OK
Server: Microsoft - IIS/5.0
Content - Location: http://www.nwu.edu.cn
Date: Mon, 24 Dec 2010 08:31:08 GMT
Content - Type: text/html
Accept - Ranges: bytes
Last - Modified: Fri, 24 Dec 2010 02:48:20 GMT
Content - Length: 9744

<!DOCTYPE html PUBLIC " - //W3C//DTD XHTML 1.0 Transitional//EN" " http://www.w3.org/TR/
xhtml1/DTD/xhtml1 - transitional.dtd">
<html>
<head>
<title>西北大学</title>
...
</body>
</html>
```

这一对请求/响应消息是使用 IE 浏览器访问西北大学主页时产生的 HTTP 消息流。在 IE 浏览器的地址栏中输入西北大学主页的地址 http://www.nwu.edu.cn,按 Enter 键后,IE 浏览器便会将这一段请求消息以文本的形式发送出去,经过网络传递到西北大学的 Web 服务器上,Web 服务器经过分析发现该客户端请求的是西北大学的主页,于是将西北大学的主页放在响应消息中发送回客户机的浏览器。下面对 HTTP 请求和响应消息分别进行详细介绍。

1.2.1 HTTP 请求消息

HTTP 请求消息由 Request-Line(请求行)、Header Field(头域)和 Message-Body(消息体)组成,如图 1.3 所示。

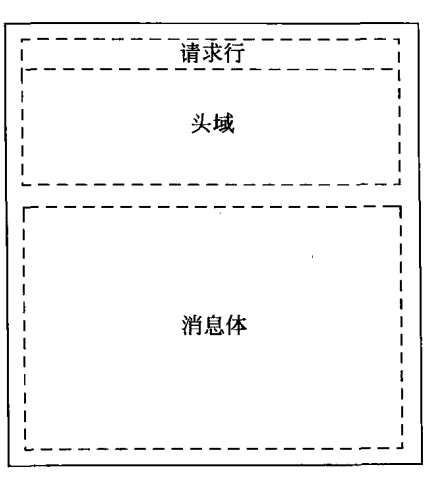

图 1.3 HTTP 请求消息格式

Request-Line 在 HTTP 请求消息的第一行,一般格式是:

```
Request - Line = Method[SP]Request - URI[SP]HTTP - Version CRLF
```

其中 Method 称为 HTTP 方法(HTTP Method),它表示该请求所要进行的操作类型;Request-URI 称为请求 URI,它表示与该请求有关的 Web 服务器中的资源定位符;HTTP-Version 表示该请求使用的 HTTP 协议的版本号,一般是 HTTP 1.0 或 HTTP 1.1,目前使用的 HTTP 版本大部分都是 HTTP 1.1。[SP]表示空格,CRLF 表示回车换行,它们都是格式信息,用于分隔各部分信息。例如:

```
GET /index.htm HTTP/1.1
```

就是一个典型的 Request-Line,其中 GET 是 HTTP 方法,/index.htm 是请求 URI,HTTP 1.1 是 HTTP 版本号。

头域紧跟在 Request-Line 的后面,每个域一行,本节后面部分将会详细介绍头域。消息体在头域后面,与头域隔一个空行,不过并不是所有 HTTP 请求消息都有消息体,有些就没有,这由该 HTTP 请求消息的 HTTP 方法类型决定。

1. HTTP 方法

HTTP 请求消息通过使用不同的 HTTP 方法来向接收到请求的主机说明其请求所期望执行的操作。HTTP 1.1 总共定义了 OPTIONS、GET、HEAD、POST、PUT、DELETE、TRACE 和 CONNECT 8 种 HTTP 方法,其中 GET 方法和 POST 方法是最常见的也是使用最多的 HTTP 方法,而其他方法使用得很少,甚至有些方法在很多服务器中都会被屏蔽或者忽略,所以本书将只重点针对 GET 方法和 POST 方法进行详细介绍。

上网浏览网页时基本上使用的都是 GET 方法。GET 方法向服务器申请得到请求 URI 指定的资源。请求 URI 可能指向的是一个 Web 服务器路径下的文件,那么接收到请求后 Web 服务器会将该文件的内容作为 HTTP 响应的内容返回给浏览器;请求 URI 指向的也可能是一个数据处理过程(比如一个 Servlet),那么 Web 服务器会执行该过程并将该过程执行结束后向客户端反馈的结果信息加入到 HTTP 响应中返回。可见使用 GET 方法进行的请求响应过程中,数据流向主要是从服务器向客户机,所以 GET 请求消息的消息体通常不包含任何内容。一般在如下场景中会使用 GET 方法:

在浏览器中输入网页地址后,会从 Web 服务器上获取网页中的所有内容,例如 HTML、图片、Flash、JavaScript 等。客户机请求每一项内容时都会将一个 GET 请求提交给服务器,然后服务器会处理每一个请求并将请求的内容作为响应返回给浏览器。单击网页上的一个图片链接打开一个图片,浏览器会将图片的 URI 构造成一个请求消息,并将请求消息提交给服务器,服务器接收到请求消息,解析请求 URI,然后将 URI 指向的图片返回给浏览器。

POST 方法则恰好与 GET 方法相反,POST 方法主要用于向服务器提交数据内容,所以一般来说 POST 消息的消息体中会包含提交的数据内容。POST 消息中请求 URI 也可以是一个文件位置或者数据处理过程。假如指向的是一个文件位置,那么 Web 服务器会将 POST 消息体中携带的数据作为一个文件保存在指定的位置;如果指向的是一个数据处理

过程，那么 Web 服务器会将 POST 消息体中携带的数据传递给该数据处理过程，并启动该数据处理过程对数据进行处理。通常 POST 方法会被使用到如下场景：

(1) 提交登录信息。当输入完用户名和密码后，单击“登录”按钮时，浏览器就会将登录信息(用户名和密码，为了安全起见，很多系统会对密码加密)作为 POST 消息的消息体提交给 Web 服务器。

(2) 在论坛中发帖子。帖子的标题和内容会作为 POST 消息的消息体提交给 Web 服务器。

(3) 发送 E-mail。E-mail 的各项信息(发件人、收件人、抄送、密送、标题、正文等)会组织成一定的格式，然后作为 POST 消息的消息体提交给 Web 服务器。

2. Request-URI

Request-URI 称为请求 URI，它是一个不含空白字符的字符串，符合 URI(统一资源定位符)的格式规范，表示了 Web 服务器上的一个资源位置。它可以是四种格式：

```
Request - URI = "*" | absoluteURI | abs_path | authority
```

“*”表示该 Request-URI 并不指向某个特定的位置，说明该 HTTP 请求消息所请求的操作是针对整个 Web 服务器，而不是针对某个特定资源的。当然并不是所有的 HTTP 方法都能够使用“*”作为 Request-URI，只有某些特定的 HTTP 方法才可以，比如 OPTIONS。

absoluteURI 是一个用绝对形式表示的 URI，即以协议开头的 URI，比如：http://www.nwu.edu.cn/images/bg.jpg，这种表示形式单独就能指定唯一的网络资源位置。

abs_path 是一个用相对形式表示的 URI，但是它必须是一个 Web 服务器上的绝对路径，必须以一个 / 开头，例如：/images/bg.jpg。这种表示形式指定了一个从 Web 服务器根目录开始的相对路径。Web 服务器根目录是服务器设置的所有 Web 资源的顶层目录。假设，域名为 nwu.edu.cn 的 Web 服务器设置的根目录是 D:\webroot，那么请求 URL “http://www.nwu.edu.cn/index.htm”就是请求 Web 服务器上的文件 D:\webroot\index.htm。可见，使用 abs_path 的 Request-URI 只是指定了 Web 服务器内部的路径，而并没有指定 Web 服务器的主机地址，所以它不能单独用于指定一个网络位置。使用这种 Request-URI 的 HTTP 请求消息都会有一个名为 Host 的头域，它的值就用于指定一个主机的地址，比如：Host 头域值为 www.nwu.edu.cn，Request-URI 为/images/bg.jpg 的 HTTP 请求消息所指定资源的位置也是 http://www.nwu.edu.cn/images/bg.jpg。

authority 仅能被 CONNECT 方法使用。

1.2.2 HTTP 响应消息

HTTP 响应消息是 Web 服务器在处理完 HTTP 请求消息后返回给客户机浏览器的消息，它也由状态行、头域和消息体组成，如图 1.4 所示。

状态行的一般格式如下：

```
Status - Line = HTTP - Version[SP]Status - Code[SP]Reason - Phrase CRLF
```

其中，HTTP-Version、[SP]和 CRLF 的意义与请求消息中的一样。Status-Code 是状

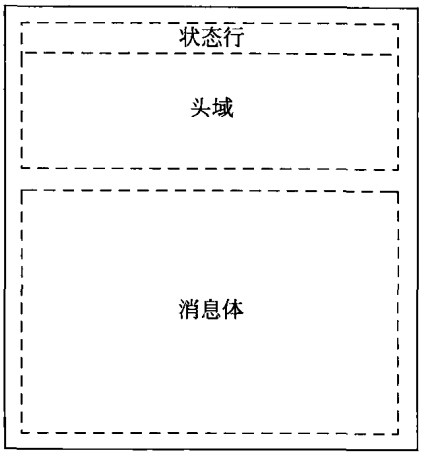

图 1.4 HTTP 响应消息格式

态码,它是 3 位十进制数,HTTP 1.1 预定义了很多状态码用于表示服务器处理请求的状态;Reason-Phrase 是一个简短的文字,它对响应状态码进行文字性说明。Status-Code 根据首位数字的不同可分为以下几大类。

1xx:信息响应类,表示接收到请求并且继续处理。例如"100 Continue"表示服务器已接收并开始处理请求,然后要求客户机继续发送请求的剩余部分,如果请求已被发送完全,客户机可以忽略该消息。

2xx:处理成功响应类,表示动作被成功接收、理解和接受。例如"200 OK"表示请求的操作已成功完成,对于 GET 请求则表示请求的资源已附在响应消息中,对于 POST 请求则表示提交的内容已被处理。

3xx:重定向响应类,为了完成指定的动作,必须接受进一步处理。例如"301 Moved Permanently"表示请求的资源已被永久地移到另外一个 URI,往后对该资源的请求应该都替换成新的 URI,新的 URI 将由响应消息的 Location 头域说明;"302 Found"表示请求应该暂时被重定向为另外一个 URI,以后对该资源的请求应该还是使用当前的 URI。

4xx:客户端错误,客户请求包含语法错误或者是不能被正确执行。例如"400 Bad Request"表示客户端提交的请求无法被服务器理解,客户端需要对请求重新改动后再提交;"403 Forbidden"表示服务器已理解客户端的请求,但是服务器拒绝执行客户端请求的操作;"404 Not Found"表示客户端请求中的 Request-URI 指定的资源位置不存在。

5xx:服务端错误,服务器不能正确执行一个正确的请求。例如"500 Internal Server Error"表示服务器遭遇一个非预期错误而导致无法完成请求的操作。

1.2.3 Header Field

如前所述,在 HTTP 请求消息和响应消息中都包含的有 Header Field(头域),这些头域用于说明一些辅助信息,以便于丰富客户机和服务器之间的通信。有些头域用于说明一些通用信息,称为 General Header Field(通用头域),既可以用于请求消息也可以用于响应消息;有些头域只被用于请求消息,称为 Request Header Field(请求头域);有些头域只被

用于响应消息，称为 Response Header Field(响应头域)；有些头域用于说明传输内容的信息，称为 Entity Header Field(实体头域)，它们可以被用于请求消息也可以被用于响应消息。整个头域由多条头域项组成，每条头域项占一行。一条头域项的一般格式为：

```
Field-Name: Field-Value
```

其中 Field-Name 是头域名，Field-Value 是头域值。

1. General Header Field

这类头域既可以出现在请求消息中也可以出现在响应消息中，它们只描述了传递消息的一些属性，而不能用于描述传送文件的信息。常见的有以下几项。

Cache-Control：用于指定一种缓冲机制，这种缓冲机制在整个请求/响应过程中必须被遵守，这个头域中指定的缓冲机制将覆盖默认的缓冲机制。例如：

```
Cache-Control: no-cache
```

Date：表示消息生成时的日期时间，该域所使用的日期格式必须符合 HTTP 日期格式，例如：

```
Date: Tue, 13 Nov 2007 08:12:31 GMT
```

Pragma：用于指定一些与实现相关的参数，在 HTTP 协议中并没有规定该头域所携带参数的意义。例如：

```
Pragma: "string"
```

其中“string”表示一个由引号括起的字符串，各种对 HTTP 协议的不同实现(例如不同的浏览器和服务器)可以利用该头域定义一系列字符串用于传递特定的信息。

Transfer-Encoding：如果该头域被指定，那就说明消息体采用了所指定的传输类型进行传输。例如最常见的：

```
Transfer-Encoding: chunked
```

表示消息体采用分块的方式进行传输。

2. Request Header Field

这类头域只出现在请求消息中，它们通常被客户机用于向服务器传递一些客户机的信息或者请求消息的信息。常见的有以下几种。

Accept：用来说明客户机浏览器能够接收的媒体格式，例如：

```
Accept: text/html,text/plain, image/*
```

表示客户机浏览器接受 html 和纯文本以及各种图片格式。

Accept-Charset：用来说明客户机浏览器能够接受的字符编码格式，例如：

```
Accept-Charset: iso-8859-1, gb2312
```

表示客户机浏览器接受的字符编码格式有 iso-8859-1(也就是 ASCII 编码)和 gb2312(一种

简体中文编码)。

Accept-Encoding：用来说明客户机浏览器能够接受的内容编码方法，通常是用来指定内容的压缩方法，例如：

```
Accept-Encoding: gzip, identity
```

表示客户机浏览器接受 gzip 压缩方式和不压缩。

Accept-Language：用来说明客户机浏览器能够接受的语言，例如：

```
Accept-Language: zh-CN
```

表示客户机浏览器接受简体中文。

From：表示提交该请求的终端用户的电子邮件，例如：

```
From: user@company.com
```

表示提交该请求的终端用户的电子邮件地址为 user@company.com。

Host：指示 Internet 上的一个主机和端口号，主机通常是域名或者 IP 地址，例如：

```
Host: www.nwu.edu.cn
```

表示该请求访问的主机的域名为 www.nwu.edu.cn。

If-Match：如果 HTTP 请求中含有该头域或者后面将要提到的 If-Modified-Since，If-None-Match，If-Range 或 If-Unmodified-Since 头域时，那么该请求就变成了“条件请求”，即只有满足上述描述的条件时请求的操作才需要被执行，这样可以减少不必要的资源浪费。该域的值是一个匹配字符串，如果该匹配字符串匹配成功则执行操作，否则不执行。在匹配字符串中 * 表示任意。例如：

```
If-Match: *
```

表示匹配任何资源。

If-None-Match：意义与 If-Match 恰好相反，表示匹配不成功则执行操作，否则不执行。

If-Modified-Since：值是一个日期，表示请求的资源如果从给定的日期后被修改过则执行操作，否则不执行。例如：

```
If-Modified-Since: Tue, 13 Nov 2007 08:12:31 GMT
```

表示如果请求的文件在 2007-11-13 08:12:31 后被修改过则执行操作。

If-Unmodified-Since：意义与 If-Modified-Since 恰好相反，表示请求的资源如果从给定的日期后没有被修改过则执行操作，否则不执行。

If-Range：假如客户机的缓冲池中已有了资源实体的一部分，而期望获得剩余部分，则客户机的请求可以使用该头域。它表示“如果指定的资源实体没有被修改则将我缺少的发给我，否则发给我整个资源实体”。

Max-Forwards：在 TRACE 和 OPTIONS 方法中使用，用于限制消息在网络中传播的跳数，即消息被代理或者网关转发的次数，以此来限制消息的生命期。

Range：用于指定一个范围，它表示请求的资源实体的范围，可以使用字节数指定。If-Range 需要的范围就是通过该头域指定的。