

入 · 门 · 与 · 提 · 高 · 系 · 列 · 丛 · 书



恒

毅

下

二

人

JSP 动态网站技术

入门与提高

r u m e n y u t i g a o

◆ 太阳工作室 孙晓龙 赵莉 编著

人民邮电出版社
www.pptph.com.cn



附光盘
CD-ROM

入门与提高系列丛书

JSP 动态网站技术入门与提高

太阳工作室 孙晓龙 赵 莉 编著

人 民 邮 电 出 版 社

图书在版编目(CIP)数据

JSP 动态网站技术入门与提高/太阳工作室/孙晓龙,赵莉编著. —北京:人民邮电出版社,2001.7
(入门与提高系列丛书)

ISBN 7-115-09409-8

I .J... II .①太...②孙... III .计算机网络—程序设计 IV .TP393.092

中国版本图书馆 CIP 数据核字(2001)第 036995 号

内 容 提 要

本书在简单介绍 Web 技术、HTML 语言、JavaScript 脚本语言及 Java 程序设计后,对 JSP (Java Server Page) 技术进行了系统的讲述,包括 JSP 运行开发环境、基本语法、实战技巧、JSP 和 XML、JSP 数据库编程技术等,最后给出了几个综合实例,以提高读者对所学知识的使用技巧。

本书以讲述 Java Server Page (JSP) 技术为主,定位在 JSP 的入门与提高上,由浅及深,有理论讲解,也有实例练习,既适合没有互联网技术基础的初学者阅读,也适合已有 CGI、PHP 和 ASP 编程经验的读者阅读。

本书的配套光盘中附有本书实例的源代码和作者所在的网博公司应用 JSP/Servlet 技术开发的网站建设和管理平台 Website Builder Demo,希望对读者的学习有帮助。

入门与提高系列丛书

JSP 动态网站技术入门与提高

◆ 编 著 太阳工作室 孙晓龙 赵 莉
责任编辑 邹文波

◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@ pptph.com.cn
网址 <http://www.pptph.com.cn>
读者热线:010-67129212 010-67129211(传真)
北京汉魂图文设计有限公司制作
北京顺义振华印刷厂印刷
新华书店总店北京发行所经销

◆ 开本:787×1092 1/16

印张:23.75

字数:579 千字

印数:1-5 000 册

2001 年 7 月第 1 版

2001 年 7 月北京第 1 次印刷

ISBN 7-115-09409-8/TP·2296

定价:39.00 元(附光盘)

前 言

众所周知, Java 以其独有的开放性、跨平台性和面向网络的交互性席卷全球, 以其安全、易用和开发周期短等特点, 迅速从最初的编程语言发展成为全球性软件开发平台, 这些优点已引起国内外计算机界的极大关注。Java 公用规范 (PAS) 已被国际标准化组织 (ISO) 认定, Java 技术已被列为当今世界信息技术三大要点之一。

Java 在 Internet 领域内发展出来的 JSP (Java Server Page) 和 Servlet 技术是国际上一种先进的 Web 技术, 已被 Sun、IBM、BEA、BV 和 Viginette 等 Internet 软件厂商广泛采用。近一年来, JSP 和 Servlet 在国内也逐渐升温, 谈论和学习的人也多了起来, 但是应当注意到很多人都处于开始学习和摸索的阶段, 谈论的问题多局限于安装、语法等初级阶段, 加之国内很缺乏 JSP 和 Servlet 方面的中文资料, 很需要这方面的书籍和教程。

近年来, 由于互联网技术的飞速发展和电子商务的盛行, 很需要一种合适的工具语言, 提供企业同企业、企业同个人的电子商务所需要的功能, 并且前端能同已发展出来的 HTML、JavaScript 等技术配合, 后端可以与数据库紧密结合, JSP 不失为一种好的选择。JSP 继承了 Java 的面向对象的程序语言能力与跨平台的优势, 而且同传统的 CGI 程序比较, JSP 不仅编写容易, 而且大幅提升了系统的执行效率。

根据权威部门的预测, 2001 年, 我国对 Java 技术人员的需求量将高达 20 余万人。然而, 到目前为止, 实际人数和 20 万人的需求量相差甚远。因此, 早日掌握 Java 技术, 对每个有志在 IT 行业发展的人来说是尤为重要的。我们深信 Java 技术必将迅速发展和普及, 所以我们组织编写了本书。

本书的作者都是多年从事软件开发和互联网应用的技术人员, 并已成功地应用 JSP 和 Servlet 技术开发出完全自主知识产权的网站建设和管理平台 Website Builder。

本书共 4 部分。

第一部分 基础篇, 分别介绍 Web 技术基础、HTML 语言和 JavaScript 脚本语言。

第二部分 入门篇, 分别介绍 Java 程序设计语言、JSP 开发环境和 JSP 基本语法。

第三部分 深入篇, 分别介绍 JSP 实战技巧、JSP 和 XML、JSP 和 JavaBean 以及 JSP 数据库编程技术方面的知识。

第四部分 实战篇, 介绍了一个 JSP 的综合实例。

本书面向有志于从事和已经从事网站开发和互联网技术的大专院校学生和技术人员。

本书的配套光盘中附有本书实例的源代码和作者所在的网博公司应用 JSP/Servlet 技术开发的网站建设和管理平台 Website Builder Demo, 希望对读者的学习有帮助。

由于 JSP 是新兴的动态网站技术, 处在不断发展的过程中, 加上作者学识有限, 书中难免有错误之处, 恳请读者批评指正。联系方式: sun_xiaolong@263.net。

作者

目 录

第一部分 基础篇

第1章 Web 技术基础	3
1.1 Web 的产生和发展	3
1.2 Web 的基本概念	3
1.2.1 客户机/服务器模式	3
1.2.2 超文本模式	4
1.2.3 HTML 语言	4
1.2.4 统一资源定位符 URL	5
1.2.5 Java 语言	5
1.2.6 JavaScript 脚本语言	5
1.3 HTTP 超文本传输通信协议	6
1.4 CGI	10
1.4.1 什么是 CGI	10
1.4.2 CGI 的传送方式	11
1.4.3 CGI 环境变量	12
第2章 HTML 语言入门	15
2.1 HTML 的概念	15
2.2 HTML 标记使用说明	16
2.2.1 head 中的标记	16
2.2.2 文本标记	18
2.2.3 超链接	22
2.2.4 定义列表	23
2.2.5 列表标记	24
2.2.6 图片标记	26
2.3 Table 表格	26
2.4 HTML 表单 (Form)	27
2.4.1 表单的概念	27
2.4.2 输入域	28
2.4.3 文本框	30
2.4.4 下拉框	31

2.4.5 选项	32
第3章 JavaScript 脚本语言	33
3.1 JavaScript 的概念	33
3.1.1 什么是 JavaScript	33
3.1.2 JavaScript 和 Java 的区别	34
3.1.3 第一个 JavaScript 程序	34
3.2 JavaScript 的数据类型	36
3.3 JavaScript 的变量	36
3.4 JavaScript 的常量	37
3.5 JavaScript 的表达式和运算符	38
3.5.1 表达式	38
3.5.2 运算符	38
3.5.3 范例	40
3.6 JavaScript 的函数	41
3.6.1 JavaScript 函数定义	41
3.6.2 函数中的形式参数	42
3.6.3 JavaScript 中的系统函数	42
3.7 JavaScript 的程序流程	43
3.7.1 if 条件语句	43
3.7.2 for 循环语句	43
3.7.3 while 循环	43
3.7.4 break 和 continue 语句	44
3.8 JavaScript 对象的使用	44
3.8.1 对象的基本概念	44
3.8.2 对象属性的引用	46
3.8.3 对象的方法的引用	46
3.8.4 常用对象的属性和方法	47
3.9 JavaScript 的事件驱动	51
3.9.1 基本概念	51
3.9.2 范例	52

第二部分 入门篇

第4章 Java 程序设计语言	59
4.1 面向对象的程序设计概念	59
4.2 Java 快速入门	60
4.3 Java 的基本数据类型	63
4.4 Java 的表达式	66
4.5 Java 的流程控制	67

4.6	Java 的类别与对象	70
4.7	Java 的类别库——包 (Package)	74
4.8	Java 的异常	75
第 5 章	JSP 概述	81
5.1	JSP 简介	81
5.2	JSP 与 CGI 的比较	82
5.3	JSP 与 ASP 的比较	83
5.4	JSP 与 PHP 的比较	86
5.5	JSP 与 Servlet 的关系	87
5.6	JSP 发展前景	88
第 6 章	JSP 运行和开发环境	91
6.1	JSP 运行和开发环境概述	91
6.2	安装自己的 JSP 开发运行环境	93
6.2.1	安装 JDK1.3	93
6.2.2	安装 Apache	97
6.2.3	安装 Tomcat	100
6.3	开发工具	106
第 7 章	JSP 基本语法	109
7.1	基本语法原理	109
7.2	标签和转义字符	111
7.3	注 释	113
7.4	表达式	115
7.5	程序码片段	116
7.6	声 明	117
7.7	内建对象	118
7.8	指令元素	119
7.8.1	JSP page 指令	119
7.8.2	JSP include 指令	122
7.8.3	脚本元素和指令元素的应用举例	123
7.9	动作元素	124
7.9.1	jsp:include 动作	125
7.9.2	jsp:useBean 动作	126
7.9.3	jsp:setProperty 动作	129
7.9.4	jsp:getProperty 动作	129
7.9.5	jsp:forward 动作	130
7.9.6	jsp:plugin 动作	131
7.9.7	jsp:param 动作	132

第三部分 深入篇

第 8 章 JSP 实战技巧	135
8.1 重定向页面	135
8.2 HTML Encoder 和 URL Encoder	136
8.2.1 HTML Encoder	136
8.2.2 URL Encoder	138
8.3 在 JSP 中读写文件	139
8.3.1 读文件	139
8.3.2 写文件	142
8.3.3 追加数据到文件	145
8.4 URL 重组和表单隐藏域	147
8.4.1 URL 重组	148
8.4.2 表单隐藏域	151
8.5 Cookie	155
8.6 session 对象的使用	158
8.7 处理 JSP 的中文问题	161
8.8 JSP 的错误处理	167
8.8.1 异常处理	167
8.8.2 转译错误与客户端请求错误处理	168
8.8.3 错误显示页	168
第 9 章 JSP 和 XML	171
9.1 XML 介绍	171
9.1.1 XML 的优点	172
9.1.2 标准 XML 范例	173
9.2 XML 语法	174
9.2.1 XML 声明	174
9.2.2 XML 元素	175
9.2.3 字符数据	175
9.2.4 XML 标记	176
9.2.5 CDATA	177
9.2.6 XML 注释	177
9.3 JSP 的 XML 语法	178
第 10 章 JSP 和 JavaBean	181
10.1 JavaBean 的概念	181
10.2 JavaBean 的生命周期	181
10.2.1 Page Scope	183

10.2.2 Request Scope	184
10.2.3 Session Scope	191
10.2.4 Application Scope	193
10.3 JavaBean 属性设置	195
第 11 章 JSP 数据库编程技术	201
11.1 关系数据库.....	201
11.2 SQL 简介	203
11.3 使用 JSP 连接数据库.....	206
11.3.1 JDBC 简介	206
11.3.2 连接数据库	206
11.4 查询数据库.....	212
11.5 ResultSet 深入	215
11.6 浏览 ResultSet	220
11.7 更新数据库.....	222
11.8 使用预编译语句 PreparedStatement	224
11.9 执行存储过程.....	225
11.10 使用事务	227
11.11 数据库连接池	228

第四部分 实战篇

第 12 章 JSP 综合实例	243
12.1 网上 CD 订购系统.....	247
12.1.1 系统结构	247
12.1.2 建立数据库表格	248
12.1.3 JavaBean 程序	250
12.1.4 JPS 程序和执行结果	258
12.2 图形计数器	312
12.3 Web 文件管理器	314
附录	370

第一部分

基 础 篇

第 1 章 Web 技术基础

1.1 Web 的产生和发展

World Wide Web 简称为 WWW 或 Web，是 Tim Berners-Lee 在欧洲粒子物理实验室（CERN）时发明的。1990 年 3 月，Tim Berners-Lee 建议开发一个超文本系统，以帮助分散在世界各地的高能物理研究人员更有效地共享最新的研究信息。

Tim Berners-Lee 使用以超文本为中心的管理方式来组织网络上的资料。一组资料通过页面方式展示出来，称为一个网页，并提出了建立、存取和浏览网页的方法。首先，从 SGML（Standardized Generalized Markup Language）语言中摘录所需的标记，建立了一个称之为 HTML 语言的 SGML 子集合，在其中增加了超链接，并设计了 HTTP 协议，通过超链接来取得超文本文件。最后，HTTP 和 HTML 使用统一资源定位符 URL 来定位网络上的文件，以提供一个分布式的客户机/服务器系统的基本结构。这个系统 Tim Berners-Lee 称之为 World Wide Web，在这个系统之中，用来保存资料并处理客户请求的是 Web 服务器，而用来查询并浏览资料的客户端就是浏览器。

1990 年底，按照最初构想设计的基于字符界面的 Web 客户浏览程序在 NextStep 计算机操作系统上实现了，并于 1991 年 3 月开始在 Internet 上运行。1991 年底，CERN 向高能物理界正式宣布了 Web 服务。

到 1993 年 1 月，在 Internet 上已有 50 多个 Web 服务器运行。1993 年 2 月，发生了一件 Web 发展过程中的大事，美国国家超级计算应用中心 NCSA 研制的图形浏览器 Mosaic 问世，这促进了 Web 在世界范围内的风行。到 1993 年 10 月，在 Internet 上运行的 Web 服务器达 500 多个。据统计，至 2000 年全世界 Web 用户已达到 1.3 亿。

Web 技术正迅猛发展，它不仅提供一种在组织内部进行业务活动的手段，而且还可促进商家和消费者之间的交易。此外，Web 的发展趋势是帮助发展一种新型的资源——信息，商家、学校和 Internet 用户正越来越依赖它。

互联网发展到今天，正逐步渗透到社会生活的各个层面。Web 网站的开发与管理技术也随着不断进步，本书介绍的 JSP 技术就是其中的佼佼者。

1.2 Web 的基本概念

1.2.1 客户机/服务器模式

在 Internet 上，Web 是以客户机/服务器（C/S 或 Client/Server）方式工作的。客户机/服务器模式是近几年来伴随网络发展而迅速发展起来的新型分布式信息处理体系结构，这种模式已开始支配所有形式的网络计算，从内部局域网到广域网和 Internet 都应用此模式进行计算。

所有的 C/S 系统都可以分为三部分：客户机、中间件和服务端。客户机涉及用户的前端部分，它常常使用图形用户界面，客户机有时也被称为客户端，浏览器就是典型的客户机。中间件处于客户机与服务端中间，通常对用户是透明的。中间件一般有三层：硬件、协议和 API。服务端通常是作为核心的程序或机器提供对客户端的服务。当用户需要服务时，与客户机的用户界面交互，客户机将这些交互事件翻译成服务端能接受的命令，通过中间件发送给服务端。服务端检查并执行这些命令，将结果通过中间件传回给客户机。客户机再将这些结果转换为用户界面的格式，提供给用户。

Web 充分发挥了 Internet 和 C/S 两方面的优势，并将两者有机结合，在 Internet 上建立各种服务端，为用户提供各种服务。

1.2.2 超文本模式

Web 在 Internet 上是基于 HTML 超文本标记语言、HTTP 超文本传输协议和统一资源定位符 URL 的。

超文本 (HyperText) 是一种新的联机信息表示和管理技术，它将以往基于线性方式的信息表示改为非线性的、非顺序的、联想式的文本组织方式，这种方式更符合人们的阅读习惯，同时用户可以动态地重新构建链接，以组成满足自己需要的阅读文本。

超文本将零星信息组织在一系列离散的节点之中，并通过链接 (Link) 来建立起节点与节点之间的联系，形成由节点链构成的网状信息结构。超文本的节点内容并不一定限于文本，可能是图像、图形、声音动画等多媒体。

超文本技术可以简单定义为收集、存储和浏览离散信息，以及建立和表示信息之间关系的技术。超文本技术不同于传统的计算机技术，它不仅注重信息本身，更注重表示信息之间的关系。用超文本技术建立和浏览信息，主要是根据关系，而不是传统的索引。它可以把任何相关的事物联系在一起。由于信息是按照关系组织起来的，所以整个系统是网状的，而不是树状的，也就是说，采用超文本技术的信息系统无所谓头和尾。

1.2.3 HTML 语言

目前，Web 的超文本是使用 HTML 超文本标记语言来编写的。当用户使用 Web 客户浏览器通过 Internet 阅读这些超文本时，浏览器负责解释嵌入其中的 HTML 语言，并按照 HTML 标记将文本中的信息显示给用户。

HTML 是一种标记语言，通过在普通文本中嵌入各种标签 (Tag)，使普通文本具有了超文本的功能。早期的 HTML 基本上是一种排版语言，它告诉计算机要使用的文字颜色、文字大小以及文字的形狀，这和大部分的排版语言没有多大差别。人们不断对 HTML 进行完善，例如，可以加入多个窗口，可以使用表格细致地安排网页格式，可以使用 Imagemap 对图形的部分区域建立链接等。HTML 语言和其他标记语言不同点在于：它可以和相关内容建立超链接，并且可以通过表单 (Form) 产生互动的页面，而且在浏览器中还加入了 JavaScript、VBScript、Java Applet、ActiveX 等新技术。

HTML 语言的详细内容请参阅第 2 章的内容。

1.2.4 统一资源定位符 URL

对于用户而言, URL 是一种统一格式的 Internet 信息资源地址的表达方法, 它将 Internet 提供的各类服务统一编址, 以便用户通过 Web 客户程序进行访问。URL 即 Universal Resource Locator 的缩写, 一般直接称之为 URL。URL 的格式分为 3 个基本部分, 基本格式如下:

信息服务类型: //信息资源地址: 端口号/文件路径

例如:

http://www.ChinaWebber.com

telnet://sun1.net.com:70

ftp://ftp.w3.org/pub/Web/doc

1.2.5 Java 语言

1996 年由于 Internet 应用迅速升温, Sun 开发的 Java 技术在 Internet 应用开发方面获得了巨大的成功。Java 技术的核心是 Java 语言。Java 语言本身仅仅是一种普通的面向对象的程序设计语言, 它的成功很大程度归功于它天生的异构平台互操作能力。采用 Java 虚拟机的概念, 为 Java 应用程序提供了一个与平台无关的环境。在此环境下, 开发人员只要用 Java 语言编写一个应用程序模块, 这个模块就可以在任何一种类型的计算机上运行。正是这种跨平台的能力加上 Java 其他的优越特性, 顺应了互联网发展的潮流, 因此才能够在极短的时间内成为 Internet 应用中的佼佼者。

Java 语言主要特征如下:

- 面向对象的程序设计语言
- 分布式的语言
- 既可解释执行, 也可编译执行
- 与操作系统和平台无关
- 支持多线程
- 简单而执行效率较高
- 健壮而且安全

1.2.6 JavaScript 脚本语言

JavaScript 是一种容易学习的脚本语言, 其本身与 Java 语言并没有直接的联系, 只是语法与 Java 语言类似, 其学习难度与 Basic 语言相仿。

JavaScript 具有在客户端浏览器被解释执行, 提供响应并处理用户交互事件的能力。因此 JavaScript 是客户端执行的脚本语言, 此语言可以被嵌入 HTML 的文件之中。

通过 JavaScript 可以做到响应使用者的需求事件 (如表单 (Form) 的输入) 而不用任何的网路来回传输资料, 所以当一位使用者输入一项资料时, 它不用经过传给服务器处理, 再传回来的过程, 而直接可以被客户端的应用程式所处理, 这也可以想像成有一个可执行程序在客户端上执行一样。

1.3 HTTP 超文本传输通信协议

Web 在 Internet 上是基于超文本传输通信协议 HTTP (Hyper Text Transfer Protocol) 进行工作的。Web 客户端与服务器通过 HTTP 建立连接和完成超文本在 Internet 上的正确传输。

HTTP 协议是一种很简单的通信协议,HTTP 定义浏览器和服务器如何通信并将信息传来传去。

HTTP 协议采用请求 (Request) 和响应 (Response) 方式,HTTP 协议是无状态的,每一次客户端请求都包括以下 4 个步骤:

(1) 连接 客户端与指定的服务器建立连接。

(2) 请求 客户端 (浏览器) 发送一个指定的请求到服务器。这个请求一般包括下述信息: 客户端使用的通信协议, 要请求的对象名称, 对象在服务器上的位置, 服务器应当如何回应, 以及客户端使用何种方式 (POST 或 GET) 来去掉这个对象。

(3) 响应 服务器收到请求后经过处理, 取得对象并送到客户端 (浏览器)。

(4) 关闭 这个“请求和响应”过程结束后, HTTP 协议关闭连接。

HTTP 协议通常用来给浏览器传送 HTML 文件, 但也可以传送其他任何文件。如果浏览器不能显示这一文件, 它会启动另一个应用程序来显示, 如果仍不能显示, 它会提示用户将其保存到磁盘上。

1. HTTP 请求头

HTTP 客户端浏览器, 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。

表 1-1 列出了一些最常见的请求头。

表 1-1 常见的 HTTP 请求头

请求头名	含 义
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
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), 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
Content-Length	表示请求消息正文的长度

续 表

请求头名	含 义
Cookie	这是最重要的请求头信息之一，参见后面的讨论
From	请求发送者的 E-mail 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

有关 HTTP 请求头完整、详细的说明，请参见 <http://www.w3.org/Protocols/> 的 HTTP 规范。

2. HTTP 应答状态码

Web 服务器响应浏览器或其他客户程序的请求时，其应答一般由以下几个部分组成：一个状态行、几个应答头、一个空行和内容文档。下面是一个最简单的应答：

```
HTTP/1.1 200 OK
Content-Type: text/plain
Hello World
```

状态行包含 HTTP 版本、状态代码和与状态代码对应的简短说明信息。在大多数情况下，除了 Content-Type 之外的所有应答头都是可选的。Content-Type 是必需的，它描述的是后面文档的 MIME 类型。虽然大多数应答都包含一个文档，但也有些应答不包含文档，例如对 HEAD 请求的应答永远不会附带文档。有许多状态代码实际上用来标识一次失败的请求，这些应答也不包含文档（或只包含一个简短的错误信息说明）。

表 1-2 列出了常见的 HTTP 1.1 状态代码以及它们对应的状态信息和含义。

应当谨慎地使用那些只有 HTTP 1.1 支持的状态代码，因为许多浏览器还只能够支持 HTTP 1.0。如果使用了 HTTP 1.1 特有的状态代码，最好能够检查一下请求的 HTTP 版本号。

表 1-2 常见 HTTP 状态代码

状态代码	状态信息	含 义
100	Continue	初始的请求已经接受，客户应当继续发送请求的其余部分（HTTP 1.1 新）
101	Switching Protocols	服务器将遵从客户的请求转换到另外一种协议（HTTP 1.1 新）

续 表

状态代码	状态信息	含 义
200	OK	一切正常, 对 GET 和 POST 请求的应答文档跟在后面
201	Created	服务器已经创建了文档, Location 头给出了它的 URL
202	Accepted	已经接受请求, 但处理尚未完成
203	Non-Authoritative Information	文档已经正常地返回, 但一些应答头可能不正确, 因为使用的是文档的拷贝 (HTTP 1.1 新)
204	No Content	没有新文档, 浏览器应该继续显示原来的文档。如果用户定期地刷新页面, 这个状态代码是很有用的
205	Reset Content	没有新的内容, 但浏览器应该重置它所显示的内容。用来强制浏览器清除表单输入内容 (HTTP 1.1 新)
206	Partial Content	客户发送了一个带有 Range 头的 GET 请求, 服务器完成了它 (HTTP 1.1 新)
300	Multiple Choices	客户请求的文档可以在多个位置找到, 这些位置已经在返回的文档内列出。如果服务器要提出优先选择, 则应该在 Location 应答头指明
301	Moved Permanently	客户请求的文档在其他地方, 新的 URL 在 Location 头中给出, 浏览器应该自动地访问新的 URL
302	Found	类似于 301, 但新的 URL 应该被视为临时性的替代, 而不是永久性的。注意, 在 HTTP1.0 中对应的状态信息是 “Moved Temporarily”。出现该状态代码时, 浏览器能够自动访问新的 URL, 因此它是一个很有用的状态代码。注意这个状态代码有时候可以和 301 替换使用。例如, 如果浏览器错误地请求 <code>http://host/~user</code> (缺少了后面的斜杠), 有的服务器返回 301, 有的则返回 302。严格地说, 我们只能假定只有当原来的请求是 GET 时浏览器才会自动重定向。请参见 307
303	See Other	类似于 301/302, 不同之处在于, 如果原来的请求是 POST, Location 头指定的重定向目标文档应该通过 GET 提取 (HTTP 1.1 新)
304	Not Modified	客户端有缓冲的文档并发出了一个条件性的请求 (一般是提供 <code>if-Modified-Since</code> 头, 表示客户只想获取比指定日期更新的文档)。服务器告诉客户, 原来缓冲的文档还可以继续使用。305 Use Proxy 客户请求的文档应该通过 Location 头所指明的代理服务器提取 (HTTP 1.1 新)
307	Temporary Redirect	和 302 (Found) 相同。许多浏览器会错误地响应 302 应答进行重定向, 即使原来的请求是 POST, 即使它实际上只能在 POST 请求的应答是 303 时才能重定向。由于这个原因, HTTP 1.1 新增了 307, 以便更加清晰地区分几个状态代码。当出现 303 应答时, 浏览器可以跟随重定向的 GET 和 POST 请求; 如果是 307 应答, 则浏览器只能跟随对 GET 请求的重定向 (HTTP 1.1 新)