

网络工程师培训教材丛书

网络编程语言 JSP 实例教程

周 颢 编著
牛允鹏 审

電子工業出版社

Publishing House of Electronics Industry

北京 · BEIJING

内 容 简 介

本书结合作者长期从事 JSP 开发大型 Web 应用型网站的实际经验,综合作者在实际工作中的心得与体会,运用大量实例来讲解网络编程语言 JSP 及其实际使用。本书重点介绍 Java 的基本语法、Web 应用程序的概念与设计方法、JSP 操作 Oracle 数据库、Java Bean 的设计与使用、JavaMail 的应用等。本书的最后给出网上虚拟社区的综合实例,向读者详细地介绍如何按照需求分析、模块设计、页面设计、JSP 编程及页面合成的步骤完成整个网站的设计。

本书针对 JSP 实际运用中遇到的问题以实例的形式加以阐述,每章后还附有小结和习题,力求使读者能够在短期内掌握进行实际开发需要了解的知识。

本书面向即将或正在从事网站开发、设计、管理工作的大专院校学生和 IT 界人士。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,侵权必究。

图书在版编目(CIP)数据

网络编程语言 JSP 实例教程/周颢编著. —北京:电子工业出版社,2002.6

(网络工程师培训教材丛书)

ISBN 7-5053-7675-6

I. 网... II. 周... III. JAVA 语言—主页制作—程序设计—教材 IV. TP393.092

中国版本图书馆 CIP 数据核字(2002)第 034106 号

责任编辑:吴金生 特约编辑:牛 炎

印 刷:

出版发行:电子工业出版社 <http://www.phei.com.cn>

北京市海淀区万寿路 173 信箱 邮编 100036

经 销:各地新华书店

开 本:787×1092 1/16 印张:22.75 字数:580 千字

版 次:2002 年 6 月第 1 版 2002 年 6 月第 1 次印刷

印 数:6 000 册 定价:33.00 元

凡购买电子工业出版社的图书,如有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系。联系电话:(010)68279077

丛 书 序

近年来,Internet(因特网)技术得到迅速的发展,已经成为计算机产业的一个技术热点。促进 Internet 高速发展的因素之一就是 Web 技术。现在的 Web 技术已经不再局限于单纯地提供信息服务,而是日益成为一个操作平台,为用户提供强大的服务功能,例如网上电子商务、社会信息数据库等。

建立一个完善的网站,不是一个人或者某一种软件就能胜任的。任务的复杂性要求必需是团体合作、集体开发。一个好的 Web 开发团体,应当具有系统管理员、前台页面设计人员和后台编程人员等多种角色,他们应具有 Web 开发的不同方面的知识,以便各司其职、协同工作,完成网站的建设。

对于系统管理员,他要架设 Web 平台和管理服务器,为设计人员和 Web 浏览者提供稳定可靠的环境。他必需了解不同操作系统下 Web 服务器的设置与管理。

对于前台页面设计人员,他要完成页面编辑和设计,为 Web 浏览者提供最佳的视觉效果和良好的交互界面。他应通晓当前最为流行的 Dreamweaver,Fireworks 和 Flash 网页设计软件。

对于后台编程人员,他需处理各种大量的信息,因此需要他精通某种 Web 编程语言(如 ASP 和 JSP 等)熟练掌握网络数据库的管理和操作。

针对上述的需求,我们这个资深的网站建设团体,结合我们长期从事网站建设的经验,编写了本套丛书,奉献给正在或即将从事 Web 管理和开发的技术人员,特别是那些大学毕业后即将从事 Web 建设而又对 Web 的有关知识不甚熟悉的青年朋友们。

本套丛书共分 6 本,具有极强的针对性,涵盖 Web 开发基础、Web 服务器管理、前台页面设计和后台编程等,足以满足 Web 开发团体中各种角色的需要。

《Web 开发技术基础教程》作为本套丛书的基础,介绍网络和 Internet 的基本知识,给出 HTML 语法解释,描述了 DHTML(动态 HTML)技术(包括 CSS 层叠样式表、JavaScript 和 VBScript)。书中更以实例的形式讨论网站建设项目开发流程,包含项目分析、系统架构、数据库选取、前台设计与后台编程等各个知识点。

《基于 Windows 的网络服务器管理教程》向读者介绍如何在 Windows 环境下搭建一个稳定而又可靠的服务器平台。通过本书的学习,读者将学会在 Windows 环境下如何设置与管理 IIS、构建 Web 和 FTP 服务器,如何使用和配置微软代理服务器,如何架设 Exchange 2000 服务器实现 Mail 服务器。

《Web 前台设计实例教程》以丰富的实例介绍当前最流行的网页前台设计工具 Dreamweaver,Fireworks 和 Flash。通过本书的学习,你可学会使用 Dreamweaver 进行网页的可视化开发,使用 Fireworks 制作各种美观的网页图形,使用 Flash 制作炫目的网页动画。本书的最后,通过两个大型的综合实例介绍三种软件是如何协同工作、发挥三者的整合功能和实现真正美观靓丽的界面的。

《网络数据库应用与开发教程》介绍 Web 技术中最令人心动的通过浏览器实现各种对

数据库的操作,回答了在网站中选用什么数据库、如何对数据库进行管理和配置,以便通过 Web 技术访问数据库。本书涉及数据库的基本知识、SQL 语言,并对当前 Web 开发中用到的主流数据库 SQL Server 2000 和 Oracle 8i 的管理和配置进行详细的分析和讲解。

Web 强大的功能离不开后台程序的支持。目前 Windows 环境下,微软公司的 ASP 几乎成了网络编程语言的代名词。《网络编程语言 ASP 实例教程》一书根据作者亲自开发商业软件的经验,以实例的方式讲解如何使用 ASP 语言进行网站开发,内容涉及 ASP 基本对象、ASP 内置 Active X 组件、如何定制 ASP 组件及 ASP 如何访问数据库等。

这边 ASP 方兴未艾,那边 JSP 正迎头赶上。JSP 的后台是大名鼎鼎的 Java。广泛的适用平台、良好的可移植性和很高的执行效率,再加上 Sun 公司的支持,JSP 语言正在 Web 开发领域大展拳脚。《网络编程语言 JSP 实例教程》一书结合作者长期使用 JSP 开发大型商业软件的经验,以实例的方式系统地介绍 Java 的基本语法、Web 应用程序的概念与设计方法、JSP 操作 Oracle 数据库、Java Bean 的设计与使用,以及 JavaMail 的应用等。本书的后半部给出了一个大型的综合实例。

有一点提请读者注意,由于本套丛书针对的是已有一定计算机知识的人员,因此,为了文字上的叙述方便和简洁,同时也是为了大家以后能顺利地阅读其他的相关资料,对于一些译文较长的专业术语,在文中用了英文缩写。对于这些英文缩写,大多数在它第一次出现处一般都给出了它的英文全称和中译文,而其他地方则是直接引用。对于大家比较熟悉的专业术语,或者某些专有软件,我们就直接给出了英文,而未加译文。

参与本套丛书的组织、策划、编写和程序调试的人员有严洪华、牛允鹏、迟成文、胡学联、张宁、崔鸿、周颢、周继恩、张春阳、卢继军、马建辉、杨恩慧、王鹏、李震宇、陈明明、魏海林、牛炎和迟璐等人。

限于我们的水平,书中不足之处在所难免,恳请广大读者批评指正。

2002 年 5 月

前 言

JSP(Java Server Pages,Java 服务器页面)是在 Sun Microsystems 公司的倡导下,由许多公司共同参与建立的一种新的动态网页技术标准。它在动态网页的建设方面具有强大而特殊的功能。

Sun 公司应用组建“Java 社团”的思想开发 JSP 技术。在开发 JSP 规范的过程中,Sun 公司与许多主要的 Web 服务器、Web 应用服务器和开发工具供应商,以及各种各样富有经验的开发团体进行合作,找到了一种适合于应用和页面开发人员的开发方法,它具有极佳的可移植性和易用性。针对 JSP 的产品,Sun 公司授权予工具提供商(如 Macromedia)、结盟公司(如 Apache、Netscape)、最终用户、协作商及其他人。最近,Sun 公司将最新版本的 JSP 和 Java Servlet(JSP 1.1,Java Servlet 2.2)的源代码发放给 Apache,以求 JSP 和 Apache 紧密结合、共同发展。Apache、Sun 公司和许多其他的公司及个人公开成立的咨询机构更能方便任何公司和个人免费取得信息。

为了适应广大网络爱好者的需要,笔者在总结以往工作经验的基础上,编写了本书。我们强调编程思想和方法,淡化枯燥的语言讲解;内容的取舍以实用为主,讲述 JSP 运用中的问题与难点,并且描述了 Web 应用程序开发的方式与流程,让读者有法可循。本书中穿插了大量的实例,使读者在阅读这本书时,既能学到理论知识,又能掌握解决实际问题的方法。

本书从介绍网络技术的发展动态、JSP 的诞生背景开始,循序渐进地介绍 Java 的基本语法、Web 应用程序的概念与设计方法、JSP 操作 Oracle 数据库、Java Bean 的设计及使用及 JavaMail 的应用等。本书的最后给出了网上虚拟社区的综合实例,向读者介绍如何按照需求分析、模块设计、页面设计、JSP 编程及页面合成的步骤完成整个网站的设计。

本书重点在于如何使用 JSP 实际问题,而不在于介绍 Java 语言或 JSP 指令本身。但考虑到有些读者对 Java 或 JSP 本身不是很了解,因此在本书的最后给出了附录 A 和 B。附录 A 中给出了常用的 Java API,附录 B 给出了 JSP 指令快速索引。如果读者对 Java 或 JSP 有不清楚的地方,建议首先阅读附录,再学习正文中的内容。如果读者本身已有这方面的知识,则可以直接阅读正文内容,直接学习如何使用 JSP。

为使读者加深理解并掌握每章的重点,在每章的末了均附有小结和习题,并给出答案。

本书无论是对一个网络新手,还是一个久经沙场的网络高手,都能够起到一定的指导作用,都可以从本书中找到一些有益的知识。

本书在编写过程中得到很多人的帮助,特别是牛允鹏教授认真地审阅了全书,并提出了很多宝贵的修改意见,在此表示衷心感谢。

本书若有疏漏之处,敬请各位朋友指正。我们的 E-mail:kitewind@sina.com。

作 者
2002 年 5 月

目 录

| | |
|--------------------------------|--------|
| 第 1 章 JSP 简介 | (1) |
| 1.1 什么是 JSP | (1) |
| 1.1.1 JSP 的由来与 JSP 的实现方式 | (1) |
| 1.1.2 JSP 的应用模型 | (3) |
| 1.1.3 JSP 的特点 | (4) |
| 1.1.4 JSP 的发展现状 | (5) |
| 1.2 JSP 与其他动态网页技术的比较 | (6) |
| 1.2.1 CGI 程序接口 | (6) |
| 1.2.2 ASP 与 JSP | (7) |
| 1.3 JSP 开发环境的建立 | (8) |
| 1.3.1 Java 运行环境的安装 | (8) |
| 1.3.2 安装 Web 服务器——Resin | (9) |
| 1.3.3 Resin 服务器的配置 | (10) |
| 1.3.4 JSP 开发环境 | (13) |
| 1.4 JSP 页面示例 | (14) |
| 1.4.1 编写一个简单的 JSP 页面 | (14) |
| 1.4.2 JSP 引擎的工作 | (15) |
| 小结 | (18) |
| 第 2 章 JSP 基本语法 | (20) |
| 2.1 JSP 语法入门 | (20) |
| 2.1.1 一个简单的 JSP 网页 | (20) |
| 2.1.2 JSP 网页结构 | (21) |
| 2.1.3 网页命令 | (21) |
| 2.1.4 数据的输出 | (22) |
| 2.1.5 注释 | (22) |
| 2.1.6 JSP 文件的解译 | (23) |
| 2.1.7 JSP 网页的错误处理 | (25) |
| 2.2 数据类型、变量和常量概述 | (27) |
| 2.2.1 数据类型 | (27) |
| 2.2.2 变量 | (28) |
| 2.2.3 常量 | (29) |
| 2.3 运算操作 | (29) |
| 2.3.1 基本运算符 | (29) |
| 2.3.2 运算符的优先级 | (32) |
| 2.3.3 类型转换 | (33) |

| | | |
|-------|--|--------|
| 2.4 | 控制结构概述 | (34) |
| 2.4.1 | 判断结构 | (34) |
| 2.4.2 | 循环结构 | (40) |
| 2.4.3 | break 与 continue 语句 | (42) |
| 2.5 | 数组的使用 | (45) |
| 2.5.1 | 数组的声明与使用 | (45) |
| 2.5.2 | 多维数组 | (46) |
| 2.5.3 | 数组长度的获得 | (46) |
| 2.5.4 | 数组使用实例 | (47) |
| 2.6 | Java 和面向对象概念 | (48) |
| 2.6.1 | 对象的概念 | (48) |
| 2.6.2 | 类的概念 | (49) |
| 2.6.3 | 实例 | (50) |
| 2.6.4 | 在 JSP 中使用类与对象 | (52) |
| 2.7 | 时间的处理 | (54) |
| 2.7.1 | Date 类型声明 | (54) |
| 2.7.2 | 时间间隔的计算 | (55) |
| 2.7.3 | 时间字段的获取与设置 | (56) |
| 2.8 | 字符串的处理 | (58) |
| 2.8.1 | 字符串的基本操作 | (59) |
| 2.8.2 | 字符串与数字类型的转换 | (64) |
| 2.8.3 | 字符串与 Date 类型的转换 | (66) |
| | 小结 | (68) |
| 第 3 章 | JSP 网页深入分析 | (71) |
| 3.1 | 指令组件 | (71) |
| 3.1.1 | 网页指令 | (71) |
| 3.1.2 | 加载指令 | (74) |
| 3.1.3 | 标记库指令 | (75) |
| 3.2 | 动作组件 | (75) |
| 3.2.1 | <jsp useBean > , <jsp setProperty > 与 <jsp getProperty > | (76) |
| 3.2.2 | <jsp param > | (76) |
| 3.2.3 | <jsp include > | (76) |
| 3.2.4 | <jsp forward > | (77) |
| 3.2.5 | <jsp plugin > | (78) |
| 3.3 | 描述语言组件 | (80) |
| 3.3.1 | 声明语句 | (80) |
| 3.3.2 | 程序代码段 | (84) |
| 3.3.3 | 表达式语句 | (85) |
| | 小结 | (85) |

| | |
|---|---------|
| 第 4 章 Web 应用程序开发 | (88) |
| 4.1 网页表单的处理 | (88) |
| 4.1.1 数据传递过程 | (88) |
| 4.1.2 表单分析 | (90) |
| 4.1.3 request 对象讨论 | (92) |
| 4.2 缓冲区的控制与网页重导 | (94) |
| 4.2.1 缓冲区使用概述 | (94) |
| 4.2.2 网页重导 sendRedirect 方法 | (96) |
| 4.3 Cookie 的使用 | (99) |
| 4.3.1 Cookie 数据的存取 | (99) |
| 4.3.2 Cookie 变量的可存取范围 | (101) |
| 4.3.3 设置 Cookie 变量的有效时间 | (104) |
| 4.4 application session 与 request | (107) |
| 4.4.1 Web 应用程序的建立 | (107) |
| 4.4.2 application 对象的使用 | (109) |
| 4.4.3 session 对象的使用 | (112) |
| 4.4.4 将数据保存到 request 对象 | (117) |
| 4.5 文件的使用 | (119) |
| 4.5.1 文件的操作 | (120) |
| 4.5.2 从文件中取得数据 | (125) |
| 4.5.3 在文件中保存数据 | (128) |
| 4.6 异常处理机制 | (132) |
| 4.6.1 异常处理概述 | (132) |
| 4.6.2 异常处理机制 | (133) |
| 4.6.3 Java 的异常类 | (134) |
| 4.6.4 throw 语句 | (135) |
| 4.6.5 错误处理网页的指定与建立 | (137) |
| 小结 | (139) |
| 第 5 章 数据库应用 | (144) |
| 5.1 JSP 与数据库 | (144) |
| 5.1.1 WWW 与数据库 | (144) |
| 5.1.2 JSP 与 JDBC | (144) |
| 5.2 Oracle 数据库概述 | (145) |
| 5.2.1 Oracle 数据库的特点 | (145) |
| 5.2.2 Oracle 基本操作 | (147) |
| 5.3 链接 Oracle 数据库 | (155) |
| 5.3.1 Oracle 数据库链接的建立与关闭 | (155) |
| 5.3.2 Oracle 数据库链接实例 | (155) |
| 5.4 从数据库中取得数据 | (157) |

| | | |
|-------|-----------------------------------|---------|
| 5.4.1 | 记录集的概念 | (157) |
| 5.4.2 | 从数据源中取得数据 | (157) |
| 5.4.3 | 取得记录集中字段数与字段名称 | (162) |
| 5.4.4 | 记录指针的使用 | (163) |
| 5.5 | 编辑数据 | (164) |
| 5.5.1 | 添加数据 | (164) |
| 5.5.2 | 删除数据 | (168) |
| 5.5.3 | 更新数据 | (173) |
| | 小结 | (178) |
| 第 6 章 | Java Bean 的使用与设计 | (179) |
| 6.1 | Java Bean 的使用 | (179) |
| 6.1.1 | Java Bean 简介 | (179) |
| 6.1.2 | 在 JSP 中使用 Java Bean | (179) |
| 6.1.3 | Bean 对象存放目录的设定与引用 | (181) |
| 6.1.4 | 第一个 Java Bean 实例 | (182) |
| 6.2 | Java Bean 的设计 | (184) |
| 6.2.1 | Bean 的基本语法 | (184) |
| 6.2.2 | 网页链接维护的 Java Bean | (186) |
| 6.2.3 | 动态广告条 | (190) |
| 6.3 | session 事件的使用 | (193) |
| 6.3.1 | 编写响应 session 事件的 Java Bean | (193) |
| 6.3.2 | 在 session 事件中链接数据库 | (195) |
| 6.4 | Bean 的保存与读取 | (199) |
| 6.4.1 | Java Bean 对象的保存与读取 | (199) |
| 6.4.2 | Bean 的保存与读取实例 | (200) |
| 6.5 | 处理文件上传与下载的 Java Bean | (207) |
| 6.5.1 | 文件上传问题分析 | (207) |
| 6.5.2 | 解决方案 | (207) |
| 6.5.3 | 文件上传 | (208) |
| 6.5.4 | 文件下载 | (212) |
| | 小结 | (214) |
| 第 7 章 | JSP 与 JavaMail 电子邮件 | (217) |
| 7.1 | JavaMail 1.2 简介及其使用方法 | (217) |
| 7.2 | 使用 JavaMail 发送一般邮件 | (218) |
| 7.2.1 | 发信程序示例 | (218) |
| 7.2.2 | 账号申请与密码邮寄 | (221) |
| 7.3 | 使用 JavaMail 发送 HTML 格式邮件及附件 | (226) |
| | 小结 | (230) |
| 第 8 章 | 综合实例——虚拟社区系统开发 | (232) |

| | | |
|-------|-------------------|---------|
| 8.1 | 虚拟社区系统概述 | (232) |
| 8.2 | 图形计数器 | (233) |
| 8.2.1 | 图形计数器的机制 | (233) |
| 8.2.2 | 图形化计数器分析 | (235) |
| 8.3 | 留言板 | (236) |
| 8.3.1 | 系统机制 | (236) |
| 8.3.2 | 留言输入 | (236) |
| 8.3.3 | 处理留言数据 | (240) |
| 8.3.4 | 查看留言 | (244) |
| 8.4 | 用户管理 | (245) |
| 8.4.1 | 数据库表 | (246) |
| 8.4.2 | 用户登录窗体 | (248) |
| 8.4.3 | 账号确认与密码检查 | (251) |
| 8.4.4 | 限制未经登录的访问 | (255) |
| 8.5 | 讨论区 | (258) |
| 8.5.1 | 数据库准备工作 | (258) |
| 8.5.2 | 讨论区的框架 | (261) |
| 8.5.3 | 主题区的显示与操作 | (262) |
| 8.5.4 | 响应区的显示与操作 | (267) |
| 8.5.5 | 张贴文章的窗体 | (274) |
| 8.5.6 | 将文章数据添加到数据库 | (280) |
| 8.6 | 实时新闻 | (283) |
| 8.7 | 聊天室 | (288) |
| 8.7.1 | 聊天室的机制 | (288) |
| 8.7.2 | 用户登录 | (289) |
| 8.7.3 | 聊天室主画面 | (297) |
| 8.7.4 | 聊天信息的显示 | (298) |
| 8.7.5 | 输入聊天信息 | (300) |
| 8.7.6 | 聊天室设定 | (303) |
| 8.7.7 | 注销操作 | (306) |
| 8.8 | 电子贺卡 | (308) |
| 8.8.1 | 电子贺卡系统的机制 | (308) |
| 8.8.2 | 设计数据库 | (309) |
| 8.8.3 | 电子贺卡设计 | (310) |
| 8.8.4 | 预览 / 发送电子贺卡 | (315) |
| 8.8.5 | 收取贺卡 | (324) |
| | 小结 | (327) |
| 附录 A | 常用 Java API | (328) |
| 附录 B | JSP 指令快速索引 | (346) |

第 1 章 JSP 简介

WWW 是目前 Internet 上最主要的信息服务类型。随着网络的发展和日益普及,它的影响逐渐深入到我们生活的方方面面。WWW 服务的基础是 HTML 语言。本书将介绍的 JSP 是在 HTML 语言的基础上使用脚本语言对网页的对象模型进行编程,它是开发和维护 Web 站点的一种重要工具,为创建动态内容的 Web 页面提供了一种简捷而快速的方法。

1.1 什么是 JSP

JSP 是 Java Server Pages(Java 服务器页面)的缩写,是由 Sun Microsystems 公司倡导、由许多公司参与一起建立的一种动态网页技术标准(<http://www.javasoft.com/products/jsp>)。它的实现是在传统的网页 HTML 文件(*.htm, *.html)中加入 Java 程序片段(Scriptlet)和 JSP 标记(tag),从而构成了 JSP 网页(*.jsp)。Web 服务器在遇到访问 JSP 网页的请求时,首先执行其中的程序片段,然后将执行结果以 HTML 格式返回给客户。程序片段可以实现建立动态网站所需要的功能,包括操作数据库、重新定向网页及发送 E-mail 等等。所有程序操作都在服务器端执行,网络上传送给客户端的仅是得到的结果,从而对客户浏览器的要求最低。

在 Sun 公司正式发布 JSP 之后,这种新的 Web 应用开发技术很快便引起了人们的关注。JSP 为创建高度动态的 Web 应用提供了一个独特的开发环境,按照 Sun 公司的说法,JSP 能够适应市场上包括 Apache WebServer 和 IIS 在内的 85% 的服务器产品。

1.1.1 JSP 的由来与 JSP 的实现方式

提起 JSP,就不能不提及 Java。Java 技术是计算机技术的一次革命,从它诞生之日起就显示了强大的生命力。随着时间的推移,Java 在各个应用层面上都有了长足的进步。这里介绍一下 Java 在 Web 方面的应用。

1. Java Applet——Java 在浏览器端的应用

Java 在 Web 方面的应用是从 Applet 开始的。Applet 是 Java 的小应用程序,它是动态、安全和跨平台的网络应用程序。Java Applet 嵌入 HTML 语言,通过主页发布到 Internet。网络用户访问服务器的 Applet 时,这些 Applet 从网络上进行传输,然后在支持 Java 的浏览器中运行。

Applet 的功能主要是在浏览器端实现某些特殊效果,它并没有完全发挥 Java 的优势。Java Servlet 的出现才真正使 Java 在 Web 服务器方面发挥了强大的功能。

2. Java Servlet——Java 在服务器端的应用

Java Servlet 译为服务器小程序。与 Applet 类似,它也是使用 Java 语言编写的小型程序。Java Servlet 是一个与协议无关、跨平台的服务方构件,它被集成到服务器中,可以实现网络上远程动态加载。

Java Servlet 没有普通应用程序常见的 main 方法,而只有一些特定的方法用于启动、执行和退出。它可以与运行于客户端的 Applet 进行交互,也可以直接与客户端的 HTML 页交互,在许多大型应用中可以作为现有的 Internet 技术和 Java 的中间桥梁。

Java Servlet 在 Web 服务器端的应用相当成功,充分发挥了 Java 的优势。但是,由于它使用 Java 语言来编写程序,与传统的 HTML 语言有很大的区别,使得学习和掌握 Java Servlet 具有很大的难度,限制了 Servlet 的推广。JSP 的出现正是为了解决这个问题的。

3. JSP——Java 服务器页面

JSP 是在 HTML 代码中插入 Java 程序片段 (Scriptlet) 和 JSP 标记 (tag) 形成的。一方面,从外表看来,JSP 代码与 Java Servlet 代码相差很大,而和 HTML 代码很相似,这使得使用 JSP 比 Servlet 要容易的多;另一方面,从 JSP 的实现方式来讲,JSP 又和 Servlet 具有密切的联系。

在 JSP 的实现方式中,从一段纯文本的 JSP 代码到一个动态的 Web 页面的过程,取决于所使用的 JSP 运行系统,也就是 JSP 引擎的工作方式。各种不同的引擎中共同的、也是最重要的一步就是把 JSP 代码翻译为以“.java”结尾的 Servlet 代码,然后再调用 Java 编译器,把它编译成可执行的 Java 代码(.class 文件)。JSP 的实现方式如图 1-1 所示。

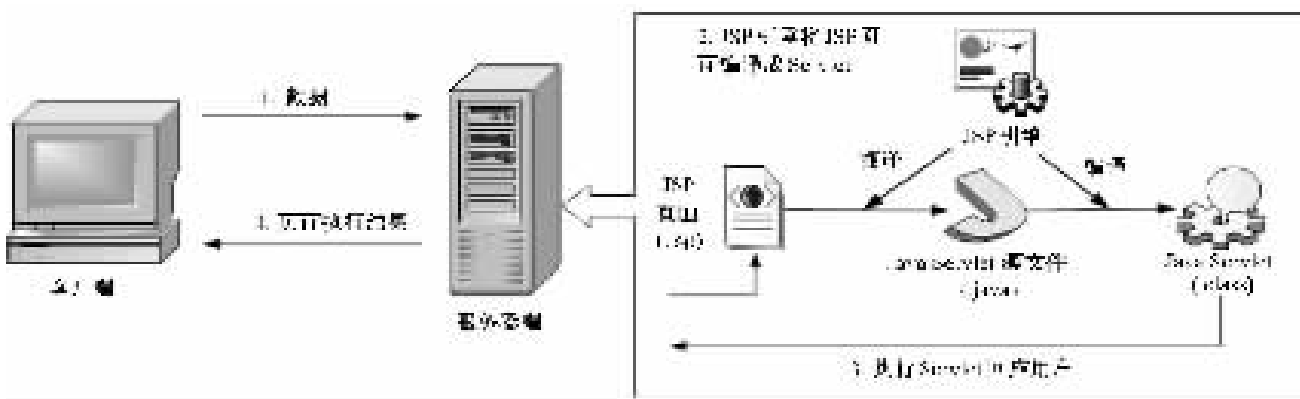


图 1-1 JSP 的实现方式

一般来讲,JSP 引擎都是构建于 Servlet 引擎上面,以 Servlet 的形式存在的。当 JSP 页面第一次被调用时,JSP 引擎首先把源代码编译为一个 Java Servlet 类,并且存储在服务器的内存中。这就是为什么在第一次执行 JSP 页面比较费时的原因,但这样避免了 CGI-BIN 为每一个 HTTP 请求生成一个新进程的问题,也避免了服务器端引用所引起的运行时语法分析。因而在以后调用该页面时有非常快的响应。除了第一次执行 JSP 时比较慢外,以后执行 JSP 的速度与 Servlet 相同,这也是 JSP 的优点之一。如果 JSP 源文件日期已发生变化,则首先 JSP 引擎会检查 JSP 文件是否比编译好的 JSP(这时实际上已经是 Servlet 了)新,如果是的

话,就表示 JSP 源文件已经更新,那么就要重新执行翻译和编译的过程。

JSP 引擎安装在 Web 服务器或使用 JSP 的应用服务器上,它可以接收客户端对 JSP 页面的请求,并且生成 JSP 页面响应客户端请求。

1.1.2 JSP 的应用模型

JSP 页面可以被包含在多种不同的应用体系结构或者模型中。JSP 页面可以用于有不同的协议、组件和格式组成的联合体中。下面是几种可能的情况。

1. 简单情况

在简单情况下,浏览器直接调用 JSP 页面,JSP 页面自己生成被请求的内容。JSP 页面可以调用 JDBC(Java DataBase Connectivity,Java 数据库连接)等组件来生成结果,创建标准的 HTML,并将结果发送回浏览器。JDBC 是 Java 应用程序访问关系数据库的接口,是可以跨平台的语言,在其上面的应用(Pure Java)可以在任何操作系统和应用服务器上运行。图 1-2 说明了这种简单的 JSP 访问模型。

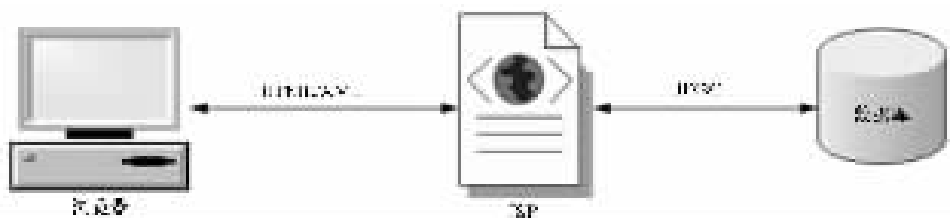


图 1-2 简单模型

使用这种方式,页面作者可以进行简单而快速的编程,很容易根据请求和资源状态生成动态内容。这种模型在许多实际应用中都能良好地工作。但当有大量并发的客户基于 Web 来访问企业资源时,由于每个客户都必须建立或者共享一个到资源的连接,大量的并发客户访问某资源时会建立大量的连接,这将影响到资源的性能。例如,利用 JSP 页面访问数据库时,可能生成许多到数据库的连接,影响到数据库的性能。

2. 使用 Servlet

基于 Web 的客户机可能直接对 Servlet 进行请求,由 Servlet 生成动态的内容,再将结果捆绑到一个结果对象中,并且调用 JSP 页面。JSP 页面从该对象中访问动态内容,并且将结果(例如 HTML)发送回浏览器。如图 1-3 所示。

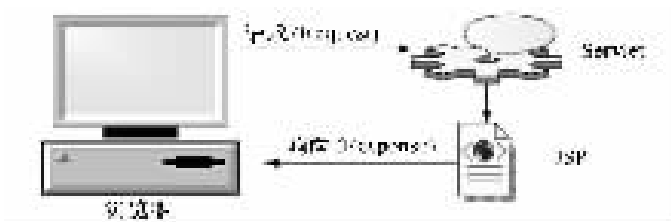


图 1-3 Servlet 的使用

采用这种方式,可以在应用程序之间创建共享的、可重用的组件,并且作为更大的应用的一部分来实现。

3. 采用企业版的 Java Bean 技术的可扩展处理

Java Bean 是一种可重用的 Java 组件,它可以被 Applet/Servlet/JSP 等 Java 应用程序调用,也可以可视化地被 Java 开发工具使用。它包含属性(Properties)、方法(Methods)和事件(Events)等特性。对于 Java Bean 的概念和应用,本书后面的章节中将进行详细地介绍。

JSP 页面可以作为企业版的 Java Bean(Enterprise Java Bean ,EJB)体系结构中的一个中间层次,如图 1-4 所示。在这种情况下,JSP 页面和后端资源通过 EJB 组件进行交互。

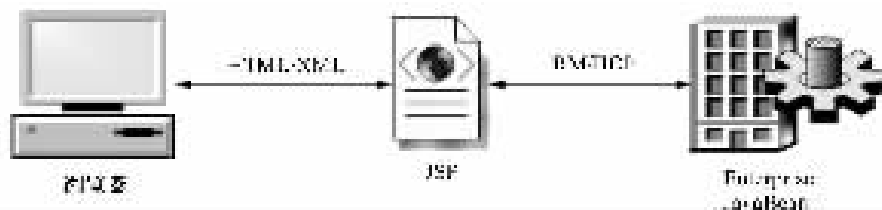


图 1-4 采用企业版的 Java Bean 技术的可扩展处理

EJB 组件管理对后端资源进行的访问,从而为众多的并发用户提供可扩展的性能。对于电子商务或者其他应用,EJB 可以管理商务并有效地控制潜在的安全性问题,这将简化 JSP 页面。Java 2 企业版(J2EE)平台支持这种模型。

1.1.3 JSP 的特点

JSP 技术有以下几个显著的优点。

1. 应用程序内容与页面显示分离

使用 JSP 技术,Web 页面开发人员可以使用 HTML 或者 XML 标记来设计和格式化最终页面,同时使用 JSP 标记或者 Scriptlet 来生成页面上根据请求而变化的动态内容(例如当前用户的信息)。生成内容的逻辑被封装在标记和 Java Bean 组件中,并且捆绑在 Scriptlet 中,所有的脚本程序在服务器端运行。这样,Web 管理人员和页面设计者,能够编辑和使用 JSP 页面,而不会影响到内容的生成。

在服务器端,由 JSP 引擎解释 JSP 标记和 Scriptlet,生成所请求的内容,并将结果以 HTML(或者 XML)页面的形式发送回浏览器。这样既可以保护作者自己的程序代码,又可以保证任何基于 HTML 的 Web 浏览器的完全可用性。

与 Servlet 相比,JSP 能够提供所有基于 Servlet 的功能,但它比用 println 编写和修改 HTML 方便得多。此外,还可以更明确地进行分工,Web 页面设计人员编写 HTML,只需要留出地方让 Servlet 程序员插入动态部分即可。

println 是 Java 中的基本输出语句。在 Servlet 中如果要输出一个 HTML 页面,则程序员需要不断调用 println 语句输出 HTML 页面的所有内容。对于一个大型网站来讲,这种工作量是很惊人的,幸好 JSP 的出现使我们不必进行这种重复而枯燥的工作。

2. 一次编写 ,到处运行

由于 JSP 页面的内置脚本语言是基于 Java 编程语言的 ,并且所有的 JSP 页面都要被编译成 Servlet ,因此 JSP 页面具有 Java 技术的所有优点 ,包括健壮的存储管理和安全性等。当然其中最重要的一点就是“一次编写 ,到处运行”。

JSP 技术是与设计平台完全无关的 ,包括动态 Web 页面、Web 服务器和底层的服务器组件。用户可以在任何平台上编写 JSP 页面 ,在任何 Web 服务器或者 Web 应用服务器上运行 ,通过任何 Web 浏览器访问。

有了这个优点 ,随着越来越多的供应商将支持 JSP 添加到他们的产品中 ,用户选择的服务器和工具的余地变得更大 ,并且更改工具或服务器并不会影响到当前的应用。

3. 强调可重用的组件

绝大多数 JSP 页面依赖于可重用的、跨平台的组件(Java Bean 或者企业版的 Java Bean 组件)来执行应用程序中所需要的复杂处理。开发人员能够共享和交换组件 ,使得这些组件能为更多的开发者或者客户团体所使用。这些组件有助于将网页和程序相互分离 ,节约了开发时间 ,同时充分发挥了 Java 和其他脚本语言的跨平台的能力和灵活性。

4. 采用标记简化页面的开发

Web 页面开发人员并不一定都是熟悉脚本语言的编程人员。JSP 技术封装了许多功能 ,提供了一系列的 JSP 标记。使用它们 ,能够完成访问 Java Bean 组件、设置或查询组件的属性及下载 Applet 等功能。并且通过开发定制化的标记库扩展 JSP 功能。今后 ,第三方开发人员和其他人员也可以为常用功能创建自己的标记库。这些标记有助于简化开发难度 ,减少开发时间。

1.1.4 JSP 的发展现状

JSP 技术被设计成为一个开放的、可扩展的及用于创建动态 Web 页面的标准。开发人员可以使用 JSP 技术来创建具有可移植性的 Web 应用程序。开发的结果可以在各种 Web 服务器和 Web 应用服务器上运行 ,而不用考虑具体的开发平台或开发工具。

为了给动态 Web 页面提供一个开放的工业标准 ,JSP 规范的 1.0 版本的提出 ,是向这个方向迈出的第一步。在 1.0 版本中 ,其基础由核心标记集、隐含对象以及开始创建动态 Web 页面所需的基本功能构成。已经有很多 Web 服务器 ,Web 应用服务器和工具开发商为他们的产品添加了对 JSP1.0 的支持。

JSP 1.0 版本中加入了一些新的特色 ,特别是加入了 XML 的语法和输出缓冲等功能 ,以解决重定向的错误 ,而同时一些在 0.92 版本中的功能被取消了。

随着 XML 越来越热门 ,在不久的将来 XML 将成为主流。因此 ,在 JSP 1.0 的语法中开始了对 XML 的支持 ,并希望能够与其兼容。不过 ,在 JSP 1.0 版本中对 XML 的支持还不是必须的 ,但在将来的 JSP 1.1 版本中对它的支持将成为必然。

JSP 1.1 与 JSP 1.0 两个版本基本上同时开发的。但是 ,JSP 1.1 需要新的 Servlet API

的支持,所以 JSP 1.1 将会等到新的 Servlet API 公布之后才会推出。除了 JSP 标准的语法标记之外, JSP 1.1 将会提供有可扩展性的自定义语法标记机制,这样使用者就可以定义自己的语法标记,并且可以在任何支持 JSP 1.1 的引擎上运行它们,而且供应商们也会有选择地扩展或扩充在 JSP 规范中的功能。JSP 引擎能够强有力地支持多种脚本语言和对象模型。在不断扩充和使用 JSP 技术的同时,也将保持 JSP 技术对平台和服务器的可移植性。

1.2 JSP 与其他动态网页技术的比较

飞速发展的基于 Web 的应用程序要求开发技术上的革新。在现阶段,新的发展方向在于创建交互式的基于 Web 的应用程序。在这些程序中,页面的内容基于用户的请求或者是系统的状态,而不是预先定义的文字,这就为动态网页技术的出现和发展创造了广泛的空间。

现在,已经出现了若干种动态网页技术, JSP 是其中的一种典型。它与其他的动态网页技术之间有什么关系和不同呢?有比较才能有鉴别,下面将其与 CGI 和 ASP 作一比较,从中不难发现 JSP 更为优异。

1.2.1 CGI 程序接口

对于创建动态 Web 页面的问题,一个最早的解决方案是使用 CGI-BIN 接口。

对 CGI(Common Gateway Interface, 公共网关接口)相信许多人都不会陌生。在动态网页出现的早期,几乎所有的网络上的复杂功能都是由 CGI 实现的。

它是在服务器端执行的程序,最主要的用途是处理从客户端送到服务器端的数据。客户端网页通常通过窗体(Form)将数据送至服务器端。服务器端再通过 CGI 接口将前端网页的数据转送给负责处理此信息的程序。当完成执行后,该程序便会将执行结果产生的网页输出给服务器,由服务器负责返回给客户端。由于程序取得数据的方式是通过 CGI 接口,因此这类程序又被称为 CGI 程序。其具体实现方式如图 1-5 所示。

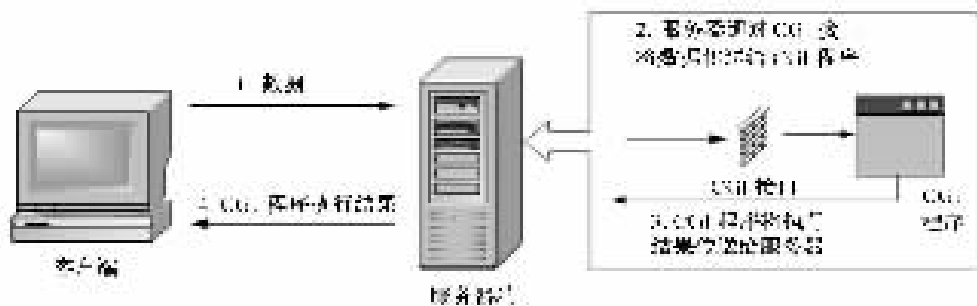


图 1-5 CGI 技术实现方式

CGI 程序的编写并不限于使用哪一种语言,可以使用诸如 C、VB 及 PERL(Practical Extraction and Report Language, 实用摘录与报告语言)等。不论使用哪种语言,该程序都必须从 CGI 接口读入一长串的字符串,再从字符串中取得所需要的数据并加以处理。解读这个

字符串的工作是编写 CGI 程序最繁琐的部分。CGI 程序的另一个缺点,就是除了 Perl 是解释式语言外,其他都是编译式的语言。每次修改 CGI 程序后,都必须重新编译一次,对开发人员提出了很高的要求。

此外,这种解决方案还存在严重的可扩展性问题。因为每个新的 CGI 都要求在服务器上新增加一个进程,如果多个用户并发地访问该程序,这些进程将耗尽 Web 服务器上的所有可用资源,使整个系统性能降低到极其低下的程度。现在的 JSP 和 ASP 技术,都直接与 Web 服务器相连接,不需要启动单独的进程,所以比 CGI 的运行效率要高得多。

CGI 的 Web 程序也难以维持他们的架构,因为 CGI 没有专门的技术可以将一组 CGI 程序组织成 Web 程序,并在互相之间共享资源,所以 CGI 不适于开发复杂的 Web 程序。而对于 ASP 和 JSP 技术,它们与 Web 服务器一起启动,一起执行直到关闭,因此很容易编写程序来管理和维护连接。

1.2.2 ASP 与 JSP

ASP(Active Server Pages,活动服务器页面)是由 Microsoft 提出的动态网页解决方案。它可以完成如动态网页生成、HTML 表单处理、创建数据库驱动的网页及跟踪用户会话等众多功能。

ASP 和 JSP 有很多相似之处,两者都是在 HTML 代码中混合某种程序代码,由语言引擎解释执行程序代码。由 HTML 代码负责信息的显示样式,而程序代码则用来描述处理逻辑。普通的 HTML 页面只依赖于 Web 服务器,而 ASP 和 JSP 页面需要附加的语言引擎来分析和执行程序代码。分析和执行完毕后,程序代码的执行结果被嵌入到 HTML 代码中,然后一起发送给浏览器。ASP 和 JSP 都是面向 Web 服务器的技术,客户端的浏览器不需要任何附加的软件支持。简言之,两者都是为基于 Web 应用实现动态交互网页制作提供的技术环境支持,两者都能够从逻辑上分离应用程序的编写和网页设计,两者都能够替代 CGI 使网站建设和发展变得较为简单和快捷。

尽管 JSP 技术和 ASP 技术在许多方面都很相似,但是它们仍然存在很多不同之处。其中本质的区别在于:两者来源于不同的技术规范组织,其实现的基础——Web 服务器平台要求不同。

ASP 的实现方式是由服务器通过 ISAPI(Internet Server Application Programming Interface,因特网服务器应用编程接口)将数据转发给 ASP 页面,然后将 ASP 页面的执行结果传给客户端。其实现方式如图 1-6 所示。

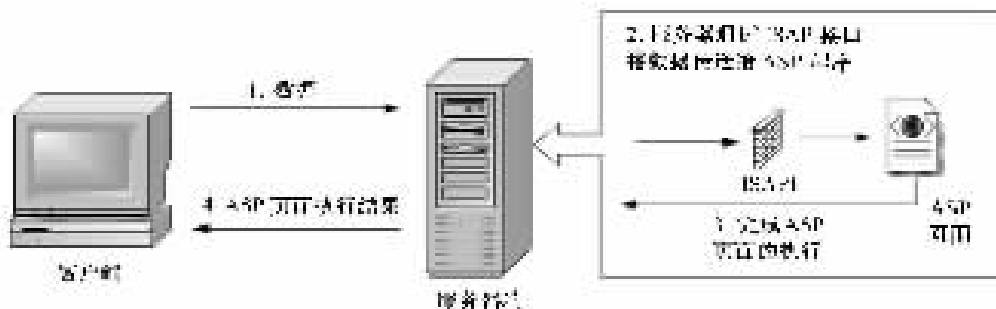


图 1-6 ASP 技术实现方式