

高等院校课程设计案例精编

赠送光盘
中附有完
整的案例
源代码

JSP 课程设计 案例精编

申吉红 廖学峰 余 健 编著

- 用户注册登录系统 •
- 聊天室系统 • BBS系统 •
- 在线商务沟通系统 • 新闻频道系统 •
- 网上书店系统 • 学生信息管理系统 •

赠送光盘

清华大学出版社



高等院校课程设计案例精编

JSP 课程设计案例精编

申吉红 廖学峰 余 健 编著

清华大学出版社

北 京

内 容 简 介

JSP(Java Server Pages)是一门动态网页开发语言。在众多的动态网页开发语言中选择 JSP,是因为它具有以下 3 大优点:一是 JSP 采用 Java 技术,具有优秀的跨平台性,几乎在所有的服务器系统上都可以运行;二是 JSP 采用 JavaBean 技术,对程序进行了很好的封装和重复调用,实现了程序的可重用性;三是 JSP 具有 Java 语言支持,可以开发完成各种复杂的应用。

本书主要以实例的方式来讲解 JSP 的应用,全书内容共分为 10 章。第 1 章和第 2 章着重介绍了 JSP 入门知识,并回顾 JSP 开发涉及到的重要理论。第 3 章到第 10 章中的每一章都是一个完整的编程实例,内容覆盖了注册登录系统、聊天室、BBS、在线商务沟通、新闻频道系统、网上书店和学生信息管理系统。每个实例都采用系统分析和模块划分的方法,讲解了各个系统是如何设计和实现的。

由于本书不是以理论介绍为重点,而更多侧重于实践应用,因此本书适合对 JSP 有一定了解或有一定 Java 编程基础的读者。本书可以作为高等院校的课程设计和实践的指导用书,也非常适用于作为 JSP 开发的参考书。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13501256678 13801310933

图书在版编目(CIP)数据

JSP 课程设计案例精编/申吉红,廖学峰,余健编著. —北京:清华大学出版社,2007.2
(高等院校课程设计案例精编)
ISBN 978-7-302-14623-0

I. J… II. ①申… ②廖… ③余… III. JAVA 语言—主页制作—程序设计—课程设计—高等学校—教学参考资料 IV. TP393.092

中国版本图书馆 CIP 数据核字(2007)第 014507 号

责任编辑:李春明 邹 杰

封面设计:山鹰工作室

版式设计:杨玉兰

责任校对:周剑云

责任印制:李红英

出版发行:清华大学出版社 地 址:北京清华大学学研大厦 A 座

<http://www.tup.com.cn> 邮 编:100084

c-service@tup.tsinghua.edu.cn

社 总 机:010-62770175 邮购热线:010-62786544

投稿咨询:010-62772015 客户服务:010-62776969

印 装 者:北京嘉实印刷有限公司

经 销:全国新华书店

开 本:185×260 印 张:25.5 字 数:611 千字

附光盘 1 张

版 次:2007 年 2 月第 1 版 印 次:2007 年 2 月第 1 次印刷

印 数:1~5000

定 价:39.00 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:(010)62770177 转 3103 产品编号:021537-01

前 言

如果说互联网将我们引入一个迷人的世界,那么 JSP 就是构建这个世界的重要工具之一。JSP、ASP(Active Server Pages)和 PHP(Personal Home Pages)一起并称为当今 Web 服务器程序开发的三大利器。它们都可以实现用户与服务器之间的交互以及数据库连接和内容读取的操作。然而,与 ASP、PHP 不同的是,JSP 拥有 Java 语言作为支撑,这使得它几乎拥有 Java 语言的所有优点:

- 优秀的跨平台性,这使得 JSP 可以运行在几乎所有的操作平台上。
- 可扩展可重用性,Java 语言可以完成任何复杂的应用,而 JSP 可以使用 JavaBean 等形式很好地封装和调用 Java 程序。
- 企业支持,JSP、Servlet 和 EJB(Enterprise Java Bean)一同构成了 J2EE(Java2 企业版)企业级解决方案,既拥有 IBM 的 WebSphere、BEA 的 WebLogic 等企业级商业平台的支持,而且还获得了 JBoss 等开源服务的拥护。

正是由于 JSP 以上的优点,使得它在前述 3 种动态开发语言中脱颖而出,成为近年来较为流行的一种语言。而且企业的支持为它注入了持久的生命力,使它当前的耀眼光辉可以延伸至更远的将来。

回顾 JSP 的发展历史,不容忽视的是 Java 语言及 J2EE 方案的成长。正是由于它们的成长,为 JSP 提供了良好的内部环境和外部环境,促进了 JSP 的成长与发展。1995 年在 Sun Microsystems 公司研发下,Java 语言问世。由于 Java 语言的面向对象、跨平台性、安全性和简单易操作等特点,使得 Java 语言迅速走红。1996 年,Netscape 浏览器在 2.0 版本上增加了对 Java Applets 和 JavaScript 的支持。同年,微软的 IE 3.0 也开始支持 Java 技术。1998 年,Sun 公司发布了 EJB 1.0 标准;1999 年 6 月,又推出了 JSP;1999 年,相继推出了 J2EE 的第一个版本。至此,Sun 完成了它与 Microsoft 的 .NET 体系对抗的 J2EE 体系。

许多人问到 JSP 的学习方法。对于这个问题,因为每个人学习特点的不同会有不同的答案。笔者总结了一个由易到难的大众性学习步骤,希望能对读者有所帮助。工欲善其事,必先利其器。在使用 JSP 编程之前,某些基础知识的掌握是非常必要的,它们是:

- HTML 语言 静态网页开发必备。如果要开发漂亮界面,还要掌握 CSS(层叠风格页面)。
- JavaScript 语言 可以完成客户端表单验证、按钮提示和回退等功能。
- Java 基本语法和面向对象 编写 JavaBean 的基础。
- JSP 的语法知识 JSP 开发基础。

这样就可以进行简单系统的开发了。这时,需要着重学习 3 个方面的知识:JavaBean、数据库和 Servlet。在开发过程中,可以同时参考几本实例书。由于不同的作者思路不同,所以对同一个功能的实现可能会表现为不同的方法,总结这些方法并比较优劣,可以快速提高实战水平。

在能够使用 JSP 开发简单系统后,可以学习 EJB 标准,特别是 Struts 架构。

本书具有以下特点：

- 运用大量实例来讲解 JSP 的应用。
- 使用系统分析、模块划分和流程图的方法讲解。
- 实例中标注了知识点、技巧和注意的地方，帮助读者更好地学习和掌握。

本书由申吉红、廖学峰、余健主要执笔，参与本书编写工作的还有张广彬、宛磊、韩阳、黄晓燕、金鑫、宋卿、陈凯、邹辉。由于作者水平有限，疏漏之处在所难免，敬请读者指正。

申吉红

angnes_7_cn@163.com

目 录

第 1 章 JSP 入门知识.....	1	2.3.3 Java Servlet	58
1.1 什么是 JSP	1	2.4 数据库知识.....	63
1.2 为什么使用 JSP.....	3	2.4.1 数据库语句.....	63
1.2.1 ASP 简介	3	2.4.2 数据库连接.....	65
1.2.2 PHP 简介	4	小结	67
1.2.3 JSP 的优势	4	第 3 章 简单的 Web 应用	
1.3 JSP 的构成	5	——MyfirstApp.....	68
1.4 HTTP 通信协议基础.....	6	3.1 Web 应用的目录结构	68
1.4.1 HTTP 协议的基本概念		3.2 部署描述文件 web.xml	69
及特点	6	3.3 Web 应用程序的打包	72
1.4.2 HTTP 协议的原理与		3.4 代码实现.....	72
工作流程	7	3.4.1 index.htm 文件的源代码.....	73
1.4.3 HTTP 协议的请求和响应.....	8	3.4.2 login.jsp 文件的源代码	73
1.5 Web 容器.....	9	3.4.3 hello.jsp 文件的源代码	74
1.6 JSP 开发与运行环境搭建.....	10	3.4.4 MyServlet.java 文件的	
1.6.1 常用开发工具.....	10	源代码.....	75
1.6.2 安装 Java 开发工具包.....	11	3.5 编译并发布.....	76
1.6.3 安装与运行 Tomcat	14	3.6 MyfirstApp 的运行与发布	77
1.7 Web 编程基础.....	16	3.6.1 运行.....	77
1.7.1 HTML	16	3.6.2 发布.....	77
1.7.2 JavaScript.....	19	小结	78
小结	23	第 4 章 用户注册登录系统.....	79
第 2 章 JSP 预备知识.....	24	4.1 学习目标和预备知识.....	79
2.1 HTML 快速入门	24	4.1.1 学习目标.....	79
2.1.1 HTML 基本结构	24	4.1.2 预备知识.....	79
2.1.2 HTML 常用标记	26	4.2 需求分析与总体设计	82
2.1.3 XML 与 XHTML.....	32	4.2.1 需求分析.....	82
2.2 Java 基础知识	36	4.2.2 总体设计.....	83
2.2.1 Java 语言简介	36	4.3 界面设计.....	86
2.2.2 Java 语法	37	4.4 数据库设计.....	89
2.2.3 Java 异常处理	42	4.4.1 建立数据库文件.....	89
2.3 JSP 基础知识	45	4.4.2 建立 ODBC 数据源.....	90
2.3.1 JSP 语法	45	4.4.3 建立 JDBC-ODBC 连接.....	92
2.3.2 JavaBean 知识.....	55		

4.5 代码设计	93	5.5 代码设计	140
4.5.1 index.jsp 代码及详解	94	5.5.1 online.java 代码及详解	140
4.5.2 action.jsp 代码及详解	97	5.5.2 index.jsp 关键 代码及详解	143
4.5.3 reg1.jsp 关键代码 及详解	99	5.5.3 action.jsp 关键 代码及详解	144
4.5.4 reg21.jsp 关键代码 及详解	100	5.5.4 select_room.jsp 关键代码及详解	148
4.5.5 reg_action.jsp 代码 及详解	101	5.5.5 main.jsp 关键 代码及详解	150
4.5.6 reg22.jsp 关键代码 及详解	105	5.5.6 user.jsp 代码及详解	152
4.5.7 reg_action2.jsp 代码 及详解	106	5.5.7 control.jsp 代码及详解	158
4.5.8 lost2.jsp 代码及详解	107	5.5.8 message.jsp 代码及详解	161
4.5.9 lost_action.jsp 代码及 详解	110	5.5.9 leave_action.jsp 代码及详解	166
4.5.10 lost3.jsp 关键代码 及详解	112	5.6 代码知识点总结	169
4.5.11 lost_action2.jsp 关键 代码及详解	112	5.7 运行调试	170
4.5.12 login.jsp 关键代码 及详解	113	小结	171
4.6 代码知识点总结	115	第 6 章 BBS 系统	172
4.6.1 表单提交	115	6.1 学习目标和预备知识	172
4.6.2 表单验证	117	6.1.1 学习目标	172
4.7 运行调试	122	6.1.2 预备知识	172
小结	123	6.2 需求分析与总体设计	173
第 5 章 聊天室系统	124	6.2.1 需求分析	173
5.1 学习目标和预备知识	124	6.2.2 总体设计	174
5.1.1 学习目标	124	6.3 界面设计	179
5.1.2 预备知识	124	6.4 数据库设计	183
5.2 需求分析与总体设计	130	6.4.1 表设计	183
5.2.1 需求分析	130	6.4.2 建立 JDBC 连接	184
5.2.2 总体设计	131	6.5 代码设计	189
5.3 界面设计	135	6.5.1 login.jsp 关键代码及详解	189
5.4 数据库设计	136	6.5.2 action.jsp 关键代码 及详解	190
5.4.1 表设计	136	6.5.3 reg.jsp 关键代码及详解	192
5.4.2 建立 JDBC 连接	138	6.5.4 reg_check.jsp 关键 代码及详解	193
		6.5.5 sort.jsp 关键代码及详解	194
		6.5.6 topic.jsp 代码及详解	197

6.5.7 topic_detail.jsp 关键 代码及详解	201	7.4.1 表设计	230
6.5.8 new_check.jsp 关键 代码及详解	203	7.4.2 建立数据库连接池	232
6.5.9 admin/login.jsp 关键 代码及详解	205	7.5 代码设计	232
6.5.10 admin/login_check.jsp 关键代码及详解	205	7.5.1 wares.java 代码及详解	232
6.5.11 admin/sort_manage.jsp 关键代码及详解	206	7.5.2 waresList.java 代码及详解	233
6.5.12 admin/addsort_check.jsp 关键代码及详解	208	7.5.3 wares_list.jsp 关键代码及详解	237
6.5.13 admin/edit_sort.jsp 代码及详解	208	7.5.4 wares_detail.jsp 关键代码及详解	240
6.5.14 admin/editsort_check.jsp 关键代码及详解	210	7.5.5 reg.jsp 关键代码及详解	241
6.5.15 admin/topic_manage.jsp 关键代码及详解	210	7.5.6 reg_check.jsp 代码及详解	243
6.5.16 admin/edit_topic.jsp 代码及详解	212	7.5.7 lost.jsp 关键代码及详解	244
6.5.17 admin/edittopic_check.jsp 代码及详解	213	7.5.8 lost_check.jsp 代码及详解	246
6.5.18 admin/user_manage.jsp 代码及详解	213	7.5.9 login_check.jsp 关键代码及详解	247
6.6 代码知识点总结	215	7.5.10 cartwares.java 代码及详解	248
6.6.1 JDBC 常用的接口	215	7.5.11 cart.java 代码及详解	250
6.6.2 实现数据库访问的步骤	215	7.5.12 cart.jsp 代码及详解	254
6.7 运行调试	217	7.5.13 clear_cart.jsp 代码及详解	255
小结	219	7.5.14 order.jsp 关键代码及详解	256
第 7 章 在线商务沟通系统	220	7.5.15 order2.jsp 关键代码及详解	257
7.1 学习目标和预备知识	220	7.5.16 order_check.jsp 关键代码及详解	258
7.1.1 学习目标	220	7.5.17 admin/login.jsp 关键代码及详解	259
7.1.2 预备知识	220	7.5.18 admin/login_check.jsp 关键代码及详解	260
7.2 需求分析与总体设计	221	7.5.19 admin/add_ware.jsp 关键代码及详解	262
7.2.1 需求分析	221	7.5.20 admin/add_check.jsp 代码及详解	264
7.2.2 总体设计	221	7.5.21 admin/manage_order.jsp 关键代码及详解	265
7.3 界面设计	226	7.5.22 admin/order_detail.jsp 关键代码及详解	267
7.4 数据库设计	230		

7.5.23	admin/execute_order.jsp	
	关键代码及详解	268
7.5.24	admin/manage_user.jsp	
	关键代码及详解	268
7.5.25	admin/del_user.jsp	
	关键代码及详解	270
7.6	代码知识点总结	270
7.7	运行调试	273
小结		274

第 8 章 新闻频道系统 275

8.1	学习目标和预备知识	275
8.1.1	学习目标	275
8.1.2	预备知识	275
8.2	系统设计	275
8.2.1	需求分析	275
8.2.2	总体设计	276
8.3	数据库设计与实现	278
8.3.1	数据库需求分析	278
8.3.2	数据库逻辑设计	278
8.3.3	创建数据库脚本	280
8.4	系统实现	281
8.4.1	实现效果	281
8.4.2	系统组成	285
8.5	Servlet 概述	286
8.5.1	什么是 Servlet	286
8.5.2	Servlet 与 JSP	287
8.5.3	Servlet 的生命周期	288
8.5.4	Servlet 结构	289
8.5.5	Servlet 响应	
	客户端请求	289
8.6	常用 Servlet API 概述	290
8.6.1	javax.servlet 包	290
8.6.2	javax.servlet.http 包	294
8.7	开发 Servlet	298
8.7.1	开发一个简单 Servlet	298
8.7.2	如何使用 Servlet	301
8.7.3	Servlet 开发中常用技术	302
8.8	代码实现	305

8.8.1	新闻频道前台	
	显示模块实现	305
8.8.2	新闻频道后台	
	管理模块实现	310

小结		311
----	--	-----

第 9 章 网上书店系统 312

9.1	学习目标和预备知识	312
9.1.1	学习目标	312
9.1.2	预备知识	312
9.2	系统设计	312
9.2.1	需求分析	312
9.2.2	总体设计	313
9.2.3	功能设计	314
9.3	数据库设计与实现	315
9.3.1	数据库的需求分析	315
9.3.2	数据库的逻辑设计	315
9.3.3	创建数据库的脚本	319
9.4	系统实现	320
9.4.1	系统组成	320
9.4.2	实现效果	320
9.5	技术要点	325
9.5.1	共享数据	325
9.5.2	Web 组件的三种	
	关联关系	326
9.5.3	使用 cookie	327
9.5.4	会话与会话跟踪	329
9.6	代码分析	330
9.6.1	用户表示层	330
9.6.2	业务逻辑层	340
小结		351

第 10 章 学生信息管理系统 352

10.1	系统设计	352
10.1.1	需求分析	352
10.1.2	总体设计	355
10.1.3	功能设计系统组成	357
10.2	数据库设计与实现	359
10.2.1	数据库的需求分析	359

10.2.2 数据库的逻辑设计	359	10.4.2 具体实现	376
10.2.3 创建数据库的脚本	364	10.5 学生选课模块实现	381
10.2.4 数据库连接池的 访问类实现	368	10.5.1 实现效果	381
10.3 新闻动态模块实现	369	10.5.2 具体实现	385
10.3.1 实现效果	369	10.6 留言板模块实现	390
10.3.2 具体实现	371	10.6.1 实现效果	390
10.4 学生信息与成绩查询模块实现	374	10.6.2 具体实现	391
10.4.1 实现效果	374	小结	393

第 1 章 JSP 入门知识

学习目标

- 了解 JSP 的基本概念以及与其他动态网页语言的区别。
- 了解 HTTP 协议的基本工作原理和工作流程。
- 了解 Web 容器的基本概念。
- 了解基本的 JSP 开发工具。
- 掌握 JDK 的安装方法和环境变量的设置。
- 掌握 Tomcat 的安装和测试方法。

1.1 什么是 JSP

JSP(Java Server Pages)是由 Sun Microsystems 公司倡导、许多公司一起参与建立的一种动态网页技术标准。JSP 技术有些类似于 ASP 技术，它是在传统的网页 HTML 文件(*.htm,*.html)中插入 Java 脚本小程序(Scriptlet)和 JSP 标记(tag)，从而形成 JSP 文件(*.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段，然后将执行结果连同 JSP 文件中的 HTML 代码一起返回给客户。插入的 Java 程序段可以操作数据库、重新定向网页等，以实现建立动态网页所需要的功能。所有程序操作都在服务器端执行，经网络上传送给客户端的仅仅是得到的结果，对客户浏览器的要求比较低，可以实现无 Plugin，无 ActiveX，无 Java Applet，甚至无 Frame。

JSP 与 Java Servlet 一样，是在服务器端执行的，通常返回给客户端的是一个 HTML 文本，因此客户端只要有浏览器就可以浏览。

在代码编写过程中，工作人员可以将网页中的动态部分和静态的 HTML 相分离。我们可以先正常地使用平常得心应手的编辑工具来书写 HTML 语句，然后将动态部分用特殊的标记嵌入，这些标记常常以“<%”开始并以“%>”结束。

下面举一个最简单的例子：

```
<html>
<head><title>jsp 示例</title></head>
<body>
<br>
<%
out.println( "A JSP Example");
%>
</br>
</body></html>
```

运行结果为：输出“A JSP Example”。



说起 JSP 的语法,可以概括为如表 1.1 所示,其详细的信息与使用规则我们将会在后面的章节中进行讲解。

表 1.1 JSP 的部分语法

JSP 元素	语 法	解 释
脚本元素: JSP 表达式	<%=表达式%>	Expression 用于计算并用于输出 <jsp:expression>表达式</jsp:expression>, 可使用的预定义的变量有 request, response, out, session, application, config 和 pageContext(在 Scriptlet 中也可使用)
脚本元素: JSP 脚本小程序	<% 代码 %>	插入用于服务的代码 <jsp:scriptlet>代码</jsp:scriptlet>
脚本元素: JSP 声明	<!% 代码 %>	属于 Servlet 部分代码但不是服务方法 <jsp:declaration>代码</jsp:declaration>
指令: JSP page 命令	<%@ page att="val" %>	指向 Servlet 引擎的路径 <jsp:directive.page att= "val" >。以下是其合法的属性(默认值加粗): import="package.class" contentType="MIME-Type" isThreadSafe="true false" session="true false" buffer="sizekb none" autoFlush="true false" extends="package.class" info="message" errorPage="url" isErrorPage="true false" language="java"
指令: JSP include 命令	<%@ include file ="URL" %>	当 JSP page 被翻译成 Servlet 时, 将包含本地系统上的文件 <jsp:directive.include file="URL"> 这个 URL 必须是相对的。当页面被请求时才使用 "jsp:include action" 调入
注释: JSP 注释	<%--注释--%>	当 JSP 转换为 Servlet 时将被忽略 <-- 注释 -->
动作: The JSP: include 动作	<jsp:include page="relative URL" flush="true"/>	在页面被请求(Requested)时调入文件。如果要想在页面被转化(Translated)时将文件包含进来, 则请使用上面所提到的 directive 来代替。警告: 在一些服务器上(Servers), 被包含的文件只能是 HTML 或 JSP, 一般以文件的后缀名来判定

续表

JSP 元素	语 法	解 释
动作: The jsp:useBean 动作	<code><jsp:useBean att=val*/></code> 或 <code><jsp:useBean att=val*>.....</code> <code></jsp:useBean></code>	寻找或生成一个 Java Bean 可能的属性是: <code>id="name"</code> <code>scope="page request session application"</code> <code>class="package.class"</code> <code>type="package.class"</code> <code>beanName="package.class"</code>
动作: The jsp:setProperty 动作	<code><jsp:setProperty att=val*/></code>	设置 bean 的属性, 通过明确的指定或使用 request 得到参数 合法的属性: <code>name="beanName"</code> <code>property="propertyName *"</code> <code>param="parameterName"</code> <code>value="val"</code>
动作: The jsp:getProperty 动作	<code><jsp:getProperty name="propertyName" value="val"/></code>	检索并输出 bean 的属性
动作: The jsp:forward 动作	<code><jsp:forward page="relative URL"/></code>	向前请求(request)另一个页面
动作: The jsp:plugin 动作	<code><jsp:plugin attribute="value"*></jsp:plugin></code>	生成特定的浏览器 OBJECT 或 EMBED 标签, 用来明确运行 Applet 所使用的 Java 插件(plugin)

1.2 为什么使用 JSP

1.2.1 ASP 简介

ASP 全名为 Active Server Pages, 是一个 Web 服务器端的开发环境, 利用它可以产生和运行动态的、交互的、高性能的 Web 服务应用程序。ASP 采用脚本语言 VB Script 或 Java Script 作为自己的开发语言。

ASP 是 Microsoft 开发的动态网页语言, 也继承了微软产品的一贯传统——只能运行在微软的服务器产品 IIS (Internet Information Server)和 PWS(Personal Web Server)上。UNIX 下也有 ChiliSoft 的插件来支持 ASP, 但是 ASP 本身的功能有限, 必须通过 ASP+COM 的组合来扩充, UNIX 下的 COM 实现起来非常困难。

1.2.2 PHP 简介

PHP 是一种跨平台的服务器端的嵌入式脚本语言。它大量地借用 C、Java 和 Perl 语言的语法,并配合 PHP 自己的特性,使 Web 开发者能够快速写出动态生成页面,它支持目前绝大多数数据库。还有一点,PHP 是完全免费的,不用花钱,可以从 PHP 官方站点 <http://www.php.net> 上自由下载。而且可以不受限制地获得开放源代码,甚至可以从其中加进自己所需要的特色。

采用 PHP 的网站如新浪网(sina)、中国人(Chinaren)等,但由于 PHP 本身存在一些缺点,使得它不适合应用于大型电子商务站点,而更适合一些小型的商业站点。

首先,PHP 缺乏规模支持;其次,PHP 缺乏多层结构支持。对于大负荷站点,解决方法只有一个,那就是分布计算。数据库、应用逻辑层和表示逻辑层彼此分开,而且同层也可以根据流量分开,组成二维阵列。而 PHP 则缺乏这种支持。此外,PHP 提供的数据库接口支持不统一,这就使得它不适合运用在电子商务中。

1.2.3 JSP 的优势

1. JSP 的效率和安全性更高

JSP 在执行之前先被编译成字节码(byte code),字节码由 Java 虚拟机(Java Virtual Machine)解释执行,比源码解释的效率高;服务器上还有字节码的 Cache(缓存)机制,能提高字节码的访问效率。第一次调用 JSP 网页速度可能稍慢,因为它被编译成缓存内容,以后就快得多了。同时,JSP 源程序一般不可能被下载,特别是 JavaBean 程序完全可以放到不对外的目录中。

2. JSP 的组件(Component)方式更方便

绝大多数 JSP 页面依赖于可重用的、跨平台的组件(JavaBean 或者 Enterprise JavaBeans™ 组件)来执行应用程序所要求的更为复杂的处理。开发人员能够共享和交换执行普通操作的组件,或者使得这些组件为更多的使用者或者客户所使用。基于组件的方法加速了总体开发过程,并且使得各种组织在他们现有的技能和优化结果的开发努力中得到平衡。

JavaBean 是完全的 OOP(面向对象程序设计),可以针对不同的业务处理功能方便地建立一整套可重复利用的对象库,如用户权限控制、E-mail 自动回复等。

3. JSP 的适应平台更广

几乎所有平台都支持 Java,JSP+JavaBean 可以在所有平台下畅通无阻。在 NT 下 IIS 通过一个插件,如 JRUN 或者 ServletExec,就能支持 JSP。著名的 Web 服务器 Apache 已经能够支持 JSP。由于 Apache 广泛应用在 NT、UNIX 和 Linux 上,因此 JSP 有更广泛的运行平台。虽然现在 NT 操作系统已经占有了很大的市场份额,但是在服务器方面 UNIX 的优势仍然很大,而新崛起的 Linux 更是来势不小。

从一个平台移植到另外一个平台,JSP 和 JavaBean 甚至不用重新编译,因为 Java 字

节码都是标准的与平台无关的。笔者将在 NT 下实验的 JSP 网页原封不动地拿到 Linux 下就运行起来了, 感觉非常满意。

4. 采用标识简化页面开发

Web 页面开发人员不一定是熟悉脚本语言的编程人员。Java Server Page 技术封装了许多功能, 这些功能是在易用的、与 JSP 相关的 XML 标识中进行动态内容生成所需要的。标准的 JSP 标识能够访问和实例化 JavaBean 组件, 设置或者检索组件属性, 下载 Applet 以及执行使用其他方法更难于编码和耗时的功能。

通过开发定制化标识库, JSP 技术是可以扩展的。今后, 第三方开发人员和其他人员可以为常用功能创建自己的标识库。这使得 Web 页面开发人员能够使用熟悉的工具和如同标识一样的执行特定功能的构件来工作。

有人做过试验, 对这三种语言分别做循环性能测试和存取 Oracle 数据库测试。

在循环性能测试中, JSP 只用了令人吃惊的 4 秒钟就结束了 20000×20000 的循环。而使用 ASP、PHP 测试的是 2000×2000 循环(少一个数量级), 却分别用了 63 秒和 84 秒(参考 PHPLIB)。

在数据库测试中, 三者分别对 Oracle 8 进行 1000 次 Insert、Update、Select 和 Delete 测试, JSP 需要 13 秒, PHP 需要 69 秒, ASP 则需要 73 秒。

在三者中, JSP 应该是未来发展的趋势。世界上一些大的电子商务解决方案提供商都采用 JSP/Servlet。比较出名的如 IBM 的 E-business, 它的核心是采用 JSP/Servlet 的 WebSphere; 西方另外一个非常著名的电子商务软件提供商 Intershop, 它原来的产品 Intershop 1, 2, 3, 4 占据了主要的电子商务软件份额, 它们都是通过 CGI(通用网关接口)来提供支持的。但是去年 10 月后 Intershop 推出了 Enfinity, 一个采用 JSP/Servlet 的电子商务 Application Server, 而且将不再开发传统软件。

总之, ASP、PHP 和 JSP 三者都拥有相当数量的支持者, 由此也可以看出三者各有所长。目前, 正在学习或使用动态页面的朋友可根据三者的特点选择一种适合自己的语言。

1.3 JSP 的构成

JSP 作为一种动态网页编程语言, 有其自身的语言特点。其 JSP 语法元素, 可以划分成注释、指令、脚本和动作四类。

1. 注释

JSP 有两种注释: 输出注释和隐藏注释。

输出注释就是 HTML 注释, 用于在客户端显示一个注释。其语法为:

```
<!--comment+<%=expression%>-->
```

隐藏注释写在 JSP 文件中, 但不会发到客户端显示。实际上, 隐藏注释在 JSP 编译时会被忽略掉, 它不会在发到客户端的代码中出现, 其语法为:

```
<%--comment--%>
```

2. 指令

指令包括页面指令、include 指令、taglib 指令和表达式语言。

3. 脚本

脚本主要有声明(Declaration)、表达式(Expression)以及脚本小程序(Scriptlet)等。

- 声明(Declaration): 在 JSP 中, 声明就是一段 Java 代码, 用来定义由 JSP 产生 Java 类的文件中的类范围和方法, 通常用于声明变量和方法, 在声明中定义的任何东西在整个 JSP 页面内均可以使用, 但其他 JSP 页面不可使用, 声明放在 JSP 的标记<%!.....%>中。
- 表达式(Expression): 用于在返回给客户的响应流中产生输出值的 Scriptlet 的简单表示法。
- 脚本小程序(Scriptlet): 是一段可以在处理请求时间执行的 Java 代码, 其包含在 JSP 标记<%.....%>之间, 其代码可以产生输出, 并将输出发到客户端。

4. 动作

动作指令在请求处理阶段起作用, 它们是以 jsp 作为前缀, 如<jsp:include>、<jsp:useBean>等。

1.4 HTTP 通信协议基础

1.4.1 HTTP 协议的基本概念及特点

HTTP(Hypertext Transfer Protocol, 超文本传输协议)是目前在 Internet 上应用最广泛的通信协议之一。HTTP 是一种应用级的协议, 轻便并能满足分布式的协同工作的超媒体信息系统所要求的速度。它是一种通用的, 无状态的面向对象的协议, 通过扩展它的请求方法(命令)可以用于很多系统。

HTTP 设计得简单而灵活, 它具有 6 个重要的特点。

1. 以 Client/Server 模型为基础

HTTP 支持客户与服务器之间通信及相互传送数据, 一个服务器可以为分布在世界各地的许多客户服务。

2. 简易性

HTTP 被设计成一个非常简单的协议, 使得 Web 服务器能高效地处理大量请求, 客户机要连接到服务器, 只需发送请求方式和 URL 路径等少量信息。HTTP 规范定义了 7 种请求方式, 最常用的有 3 种: GET、HEAD 和 POST。每一种请求方式都允许客户以不同类型的消息与 Web 服务器进行通信, Web 服务器也因此可以是简单小巧的程序。由于 HTTP 简单, HTTP 的通信与 FTP、Telnet 等协议的通信相比, 速度快而且开销小。

3. 灵活性与内容-类型(content-type)标识

HTTP 允许任意类型数据的传送, 因此可以利用 HTTP 传送任何类型的对象, 并让客户程序能够恰当地处理它们, 内容-类型标识指示了所传输数据的类型。

4. “无连接”性

HTTP 是“无连接”的协议, 但值得特别注意的是, 这里的“无连接”是建立在 TCP/IP 协议之上的, 与建立在 UDP 协议之上的无连接不同。这里的“无连接”意味着每次连接只能处理一个请求。客户要建立连接需要先发出请求, 收到响应, 然后断开连接, 这实现起来效率十分高。采用这种“无连接”协议, 在没有请求提出时, 服务器就不会在那里空闲等待。完成一个请求之后, 服务器即不会继续为这个请求负责, 从而不用为保留历史请求而耗费宝贵的资源。这在服务器的一方实现起来是非常简单的, 因为只需保留活动的连接(Active Connection), 不用为请求间隔而浪费时间。

5. 无状态性

HTTP 是无状态的协议, 这既是优点也是缺点。一方面, 由于缺少状态使得 HTTP 累赘少, 系统运行效率高, 服务器应答快; 另一方面, 由于没有状态, 协议对事务处理没有记忆能力, 若后续事务处理需要有关前面处理的信息, 那么这些信息必须在协议外面保存。

6. 元信息(meta information)

收到数据的浏览器可以根据元信息确定服务器发来的是什么内容, 预料有多少数据, 明确是否接收到完整的数据以及发送过程中是否有错, 这样客户就可以知道传输对象的类型。元信息被定义为“关于信息的信息”, 元信息允许服务器提供所发送数据的信息, 例如, HTTP 可以提供所发送的对象的语言和类型, 也可以用元信息来实现有条件请求和报告事务的完成。元信息的引入使得 HTTP 协议所能做的工作更多了。

1.4.2 HTTP 协议的原理与工作流程

HTTP 通信协议允许用户提出 HTTP “请求”(request), 然后由服务器根据实际处理结果传回 HTTP “响应”(response), 其基本运行方式如图 1.1 所示。

- (1) 当用户向 Web 服务器送出请求时, Web 服务器将会开启一个新连接。
- (2) 通过这个连接, 用户可以将 HTTP 请求传送给 Web 服务器。
- (3) 当 Web 服务器收到 HTTP 请求时, 将进行解析与处理, 并将处理结果包装成 HTTP 响应。
- (4) 最后, Web 服务器会将 HTTP 响应传送至用户。只要用户接收到 HTTP 响应, Web 服务器就会关闭这个连接, 用户的执行状态将不会被保存。