

JSP 组建动态网站 基础与实例操作

李长林 等编著

JSP
JSP JSP JSP
JSP



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

JSP 组建动态网站 基础与实例操作

李长林 等编著

電子工業出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

JavaServer Pages (JSP) 是基于 Java 的技术, 用于创建跨平台及跨 Web 服务器的动态网页。本书详细讲述了 JSP 的语言规范、运行环境、基本语法、指令以及建立动态网站的各种知识, 并结合实例, 覆盖了许多高级主题, 如 Enterprise Java Beans、JDBC 数据库连接池和自定义标签库, 读者通过学习可以逐步掌握采用 JSP 技术制作动态网站的技能。

本书可作为从事网站编程的广大科技人员的参考书, 也可作为高等院校计算机专业师生教学参考书以及相关专业培训教材。

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

JSP 组建动态网站基础与实例操作/李长林等编著. —北京: 电子工业出版社, 2005.12

ISBN 7-121-01971-X

I. J… II. 李… III. JAVA 语言—主页制作—程序设计 IV. TP393.092

中国版本图书馆 CIP 数据核字 (2005) 第 137212 号

责任编辑: 沈艳波 刘镕骥

印 刷: 北京智力达印刷有限公司

出版发行: 电子工业出版社

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: 787×1 092 1/16 印张: 23.75 字数: 608 千字

印 次: 2005 年 12 月第 1 次印刷

印 数: 5 000 册 定价: 32.00 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。质量投诉请发邮件至 zltts@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

前 言

JSP 是 Sun 公司推出的一种开发语言, 和 ASP、PHP 相比, JSP 在以下几个方面有了重大的突破:

(1) 通过 JSP 的扩展标签库和 JavaBeans 功能, 网站逻辑和网站界面可以完美地分离;

(2) 使用 Enterprise JavaBeans, 可以轻松地在 JSP 开发的 Web 中实现事务、安全、会话等企业级应用所需要的功能;

(3) JDBC 提供了不同的数据库产品无关的数据库连接方式。更重要的是, 数据库连接池提供了一种比普通的数据库连接方式效率高得多的连接方式。

JSP 的语法基本上和 Java 语法是相同的, 有 Java 基础的读者可以很快学会如何使用 JSP, 而没有 Java 语言基础的读者, 只要循序渐进地阅读本书, 一样可以成为 JSP 编程的高手。

本书从最基础的 JSP 开发开始, 循序渐进地介绍了 JSP 开发技术, 并结合实例, 覆盖了许多高级主题, 如 Enterprise JavaBeans、JDBC 数据库连接池和自定义标签库。

本书的内容包括: 第 1 章 JSP 概述; 第 2 章 HTTP 和 Servlet 基础; 第 3 章 Java 和 JSP 基础; 第 4 章 脚本; 第 5 章 动态页面; 第 6 章 JavaBeans 组件; 第 7 章 JDBC 基础; 第 8 章数据库应用技术; 第 9 章配置与访问数据库; 第 10 章处理输入和输出; 第 11 章错误处理和调试; 第 12 章用户间共享数据; 第 13 章 HTML 和 XML; 第 14 章预编译; 第 15 章开发电子商务系统——网上购物车实例; 第 16 章网上拍卖系统。

本书既适合初学者阅读, 也适合具有一定 JSP 基础的开发人员深入研究使用。

本书主要由李长林编写, 参加编写的人员还有王波波、兰吉昌、姜艳波、顾正大、艾丽香、辛征、李志、张玉平、刘文涛等。

编著者

2005 年 11 月

目 录

第 1 章 JSP 概述	(1)
1.1 JSP 的特点	(1)
1.2 JSP 与 PHP、ASP 的比较	(3)
1.3 JSP 的运作模式	(4)
1.4 建立 JSP 运行环境	(5)
1.4.1 JSP 运行环境需要的组件	(5)
1.4.2 J2SDK 及 JSWDK 安装与配置	(8)
1.4.3 在 Linux 下建立 JSP 运行环境	(11)
1.4.4 在 UNIX 下建立 JSP 运行环境	(13)
1.4.5 在 Windows NT 下建立 JSP 运行环境 (jswdk)	(15)
1.4.6 在 Windows NT 下建立 JSP 运行环境 (Resin)	(16)
1.4.7 Windows NT 下建立 JSP 运行环境 (Tomcat)	(17)
1.5 建立 JSP 程序	(18)
1.5.1 编写代码	(18)
1.5.2 保存文件	(19)
第 2 章 HTTP 和 Servlet 基础	(21)
2.1 HTTP 请求/响应模型	(21)
2.2 Servlet	(25)
2.2.1 Servlet 基础知识	(26)
2.2.2 Servlet 程序的运行环境	(28)
2.2.3 使用 Servlet 编写程序 Hello World!	(29)
第 3 章 Java 和 JSP 基础	(31)
3.1 Java 基础知识	(31)
3.1.1 Java 语言的特点	(31)
3.1.2 Java 语言规则	(31)
3.1.3 Java 开发环境	(35)
3.1.4 实例	(36)
3.2 JSP 页面概述	(39)
3.3 JSP 处理	(40)
3.3.1 Java 语言中异常类	(41)
3.3.2 异常处理	(42)
3.3.3 实例	(44)
3.4 采用 MVC 模型的 JSP 应用设计	(48)
3.4.1 MVC 简介	(48)
3.4.2 MVC 开发模式	(50)

3.4.3	在 Web 应用中使用 MVC 模式.....	(51)
第 4 章	脚本	(53)
4.1	使用脚本	(53)
4.1.1	声明、表达式、脚本的比较.....	(53)
4.1.2	用标签组合脚本.....	(54)
4.1.3	示例.....	(54)
4.2	表达式	(56)
4.3	Scriptlet	(57)
4.4	声明	(58)
4.5	JSP 注释	(60)
4.5.1	HTML 注释.....	(60)
4.5.2	JSP 注释	(61)
4.6	动作和脚本元素	(62)
4.6.1	动作.....	(62)
4.6.2	脚本元素.....	(65)
4.7	JSP 内部对象	(67)
第 5 章	动态页面	(79)
5.1	创建一个 JSP 页面	(79)
5.1.1	获取用户请求.....	(79)
5.1.2	翻译 JSP 页面.....	(79)
5.2	JSP 页面的安装与运行.....	(80)
5.2.1	安装 JSP 页面.....	(80)
5.2.2	运行 JSP 页面.....	(81)
5.3	使用 JSP 指令元素	(83)
5.3.1	JSP 编译指令 Page.....	(84)
5.3.2	JSP 编译指令 Include.....	(86)
5.3.3	taglib 指令	(89)
5.4	使用模板文本	(91)
5.4.1	使用 JSP 模板来封装网页布局.....	(91)
5.4.2	使用模板.....	(92)
5.5	使用 JSP 动作元素	(94)
5.5.1	include 动作	(94)
5.5.2	useBean 动作	(97)
5.5.3	setProperty 动作	(101)
5.5.4	getProperty 动作	(101)
5.5.5	Forward 动作	(102)
5.5.6	plugin 动作	(104)
5.5.7	param 动作	(108)
第 6 章	JavaBeans 组件	(110)
6.1	Beans 概述.....	(110)

6.2	JavaBeans 的设计目标	(112)
6.3	JavaBeans 的任务	(113)
6.4	JavaBeans 的基本结构	(114)
6.4.1	JavaBeans 属性	(114)
6.4.2	JavaBeans 方法	(117)
6.4.3	JavaBeans 事件	(117)
6.5	JavaBeans 实例	(118)
6.5.1	JavaBeans 使用方案	(118)
6.5.2	Java 基本类型与 String 类型的转换方法	(119)
6.5.3	创建和使用 JavaBeans	(121)
6.5.4	服务器端验证客户端输入数据 Bean	(122)
第 7 章	JDBC 基础	(132)
7.1	JDBC 简介	(132)
7.2	JDBC 技术原理	(132)
7.3	JDBC 技术的发展	(134)
7.4	ODBC、JDBC 和 SQLJ 的比较	(136)
7.5	JDBC 的结构	(137)
7.6	主要的 JDBC 类介绍	(137)
7.6.1	JDBC 驱动程序管理器——DriverManager 类	(138)
7.6.2	Connection 类	(140)
7.6.3	Statement 类	(142)
7.6.4	PreparedStatement 类	(146)
7.6.5	CallableStatement 类	(149)
7.6.6	DatabaseMetaData 类	(151)
7.6.7	ResultSet 类	(155)
7.6.8	ResultSetMetaData 类	(159)
7.7	JDBC 应用	(159)
7.7.1	应用于 Applet	(159)
7.7.2	应用在 Application	(160)
第 8 章	数据库应用技术	(161)
8.1	数据库简介	(161)
8.2	SQL Server 2000 数据库	(162)
8.2.1	SQL Server 2000 简介	(162)
8.2.2	SQL Server 2000 版本	(163)
8.2.3	SQL Server 2000 的特性	(164)
8.2.4	SQL Server 构架	(165)
8.3	Oracle 数据库	(167)
8.3.1	Oracle 数据库的存储结构	(167)
8.3.2	创建 Oracle 数据库	(167)
8.3.3	手工创建 Oracle 数据库	(168)

8.3.4	数据库创建之后应考虑的事项	(173)
8.3.5	初始化调整指导	(173)
第 9 章	配置与访问数据库	(175)
9.1	配置数据库	(175)
9.1.1	使用连接和连接池	(175)
9.1.2	配置 SQL Server2000 数据库	(181)
9.1.3	配置 Oracle 数据库	(185)
9.2	典型 JSP 数据库连接	(192)
9.3	使用 SQL 语句	(197)
9.4	访问数据库	(199)
9.4.1	从 JSP 页面访问数据库	(199)
9.4.2	JSP 访问数据库的方法	(202)
第 10 章	处理输入和输出	(204)
10.1	数据的输入/输出	(204)
10.1.1	数据的输出	(204)
10.1.2	数据的输入	(206)
10.2	验证用户输入	(208)
10.3	使用 HTML 样式格式化文本	(211)
第 11 章	错误处理和调试	(212)
11.1	处理语法错误	(212)
11.2	调试 JSP 应用	(213)
11.3	处理运行时错误	(218)
第 12 章	用户间共享数据	(223)
12.1	JSP 页面元素的访问控制	(223)
12.2	会话状态管理和应用数据	(227)
12.2.1	会话状态	(227)
12.2.2	控制会话的结束时间	(229)
12.2.3	在会话对象中保存数据	(230)
12.3	实例操作	(232)
12.3.1	在多个 JSP 页面之间传递参数实例	(232)
12.3.2	意见箱	(234)
12.4	内存使用策略	(251)
第 13 章	HTML 和 XML	(253)
13.1	XML 基本概念	(253)
13.2	HTML 基础知识	(255)
13.2.1	HTML 标记	(256)
13.2.2	HTML 表格	(262)
13.2.3	HTML 表单	(263)
13.3	XML 的转换	(270)
13.3.1	将 XML 转换为 HTML	(270)

13.3.2	将 XML 转换为依赖于设备的格式	(271)
13.4	处理 XML 数据和文档	(272)
13.4.1	将 XML 数据放到 JSP 中	(272)
13.4.2	JSP 和 XML 语法	(272)
第 14 章	预编译	(275)
14.1	预编译概述	(275)
14.1.1	程序方式的预编译	(275)
14.1.2	执行 weblogic.jspc	(276)
14.2	避免 JSP 页面缓存	(277)
14.2.1	在 Servlet 的 init()方法中缓存数据	(277)
14.2.2	禁止 Servlet 和 JSP 自动重载 (auto-reloading)	(278)
14.2.3	提高系统的性能	(278)
14.2.4	将页面输出进行压缩	(278)
14.2.5	选择正确的页面包含机制	(279)
14.2.6	正确地确定 JavaBean 的生命周期	(279)
14.3	将 JSP 页面写做 XML 文档	(280)
14.3.1	用 Taglib 指令将 JSP 页面写做 XML 文档	(280)
14.3.2	示例	(281)
第 15 章	开发电子商务系统——网上购物车实例	(282)
15.1	开发环境	(282)
15.1.1	Weblogic7.0 的安装	(282)
15.1.2	设置 JSP 服务器	(286)
15.1.3	测试配置	(291)
15.2	功能简介及建表	(292)
15.2.1	目标设计	(292)
15.2.2	数据库连接	(292)
15.2.3	创建数据库	(293)
15.3	网上购物程序结构	(300)
15.3.1	放入购物车的 JavaBean 文件	(300)
15.3.2	系统首页面	(302)
15.3.3	用户注册页面	(308)
15.3.4	添加物品页面	(312)
15.3.5	商品查询页面	(314)
15.3.6	购物车信息页面	(319)
15.3.7	移出购物车页面	(326)
15.3.8	不得结账页面	(327)
15.4	补充	(332)
第 16 章	网上拍卖系统	(333)
16.1	网上拍卖概述	(333)
16.1.1	全球网上拍卖发展的现状	(333)

16.1.2	本章设计的网上拍卖系统概述	(334)
16.1.3	系统浏览	(334)
16.1.4	系统特点	(335)
16.1.5	消费者网上会员中心	(335)
16.1.6	网上商品“列表”	(336)
16.1.7	商品搜索引擎	(336)
16.1.8	竞拍申请	(336)
16.1.9	发布商品	(337)
16.2	系统设计	(337)
16.2.1	系统设计思想	(337)
16.2.2	功能模块的划分	(337)
16.2.3	系统开发手段与运行环境的配置	(337)
16.3	数据库设计	(338)
16.3.1	数据库需求分析	(338)
16.3.2	数据库概念结构设计	(338)
16.3.3	数据库逻辑结构设计	(340)
16.4	JSP 脚本与相关 Bean 组件的设计	(341)
16.4.1	数据库连接的设计	(341)
16.4.2	商品列表 Bean 的设计	(343)
16.5	网上拍卖系统的界面设计与 I/O 设计	(346)
16.5.1	系统界面的设计	(346)
16.5.2	输入/输出的设计	(347)
16.5.3	系统 I/O 的容错设计	(347)
16.6	网上拍卖系统的实现	(348)
16.6.1	界面头的设计	(348)
16.6.2	会员登录模块的设计	(348)
16.6.3	会员注册模块	(350)
16.6.4	个人信息模块的设计	(353)
16.6.5	商品列表模块的设计	(355)
16.6.6	发布商品模块的设计	(357)
16.6.7	竞拍申请模块的设计	(359)
16.6.8	个人资料模块的设计	(363)
16.6.9	管理员模块的设计	(366)
16.6.10	系统维护模块的设计	(368)

第1章 JSP 概述

JavaServer Pages (JSP) 是由 Sun Microsystems 公司于 1999 年 6 月开发出来的一种全新的动态网页标准。JavaServer Pages (JSP) 是基于 Java 技术, 用于创建可支持跨平台及跨 Web 服务器的动态网页。目前作为最热门的网络编程语言, JSP 在电子商务和电子政务开发中得到了广泛的应用。

在传统的 HTML 文件中加入 Java 程序片段和 JSP 标记, 就构成了 JSP 网页。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。程序片段可以操作数据库、重新定义网页以及发送 E-mail 等, 这就是建立动态网站所需要的功能。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, 对客户浏览器的要求很低, 可以实现无 Plugin、无 Activex、无 Java Applet, 甚至无 Frame。

目前的 Java2 platform Enterprise Edition 简称 J2EE, 是 Sun 公司为企业级应用建立的高水准的架构, 它获得了各大开发平台提供商的积极支持, 并已成为当今企业应用开发的主流产品。JSP 技术是 J2EE 平台上的一个关键性的技术。由于 JSP 技术被设计成独立于平台和服务器的, 因此, 可以利用这一技术建立先进、安全和跨平台的动态网站。它的建立来源于通用的工具、服务器和数据库供应商, 所以 JSP 有着非常广阔的发展前景。

另外, JSP 通过 JavaBean 实现了功能扩充, 能够执行更为复杂的处理。

1.1 JSP 的特点

JSP 是动态网页开发技术, 它的主要特性如下。

1. 将内容的生成和显示进行分离

使用 JSP 技术, Web 页面开发人员可以使用 HTML 或者 XML 标识来设计最终页面。使用 JSP 标识或者小脚本来生成页面上的动态内容 (内容是根据请求来变化的, 例如请求账户信息或者特定物品的价格)。生成内容的逻辑被封装在标识和 JavaBeans 组件中, 并且捆绑在小脚本中, 所有的脚本在服务器端运行。如果核心逻辑被封装在标识和 Beans 中, 那么其他人, 如 Web 管理人员和页面设计者, 能够编辑和使用 JSP 页面而不影响内容的生成。

在服务器端, JSP 引擎解释 JSP 标识和小脚本, 生成所请求的内容 (例如, 通过访问 JavaBeans 组件, 使用 JDBC 技术访问数据库), 并且将结果以 HTML (或者 XML) 页面的形式发送回浏览器。这既有助于保护自己的代码, 又保证了任何基于 HTML 和 Web 浏览器的完全可用性。

2. 生成可重用的组件

绝大多数 JSP 页面依赖于可重用的、跨平台的组件 (JavaBeans 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团体所使用。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。

3. 采用标识简化页面开发

Web 页面开发人员不都是熟悉脚本语言的编程人员。而 JavaServer 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时所需要的。标准的 JSP 标识能够访问和实例化 JavaBeans 组件、设置或者检索组件属性、下载 Applet，以及执行用其他方法更难于编程且耗时的功能。

通过开发制定标识库，JSP 技术是可以扩展的，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执行特定功能的构件来工作。

4. 具有“一次编写、各处运行”的特点

JSP 在设计时，充分考虑到应用平台的无关性。因此 JSP 完全与平台无关，可以在 Apache、Netscape、IIS 等服务器上执行。支持的操作系统平台有 Solaris、Windows、Mac OS、Linux 等。

作为 Java 平台的一部分，JSP 拥有 Java 编程语言“一次编写，各处运行”的特点。随着越来越多的供应商将 JSP 支持添加到他们的产品中，可以使用自己所选择的服务器和工具，更改工具或服务器且不影响当前的应用。

5. 程序执行的效率高

在 JSP 中，代码被编译成 Servlet 并由 Java 虚拟机执行，编译过程只在程序第一次执行时进行，不需要每次执行程序时都编译，因此，代码的执行效率很高。

由于 JSP 页面的内置脚本语言是基于 Java 编程语言的，而且所有的 JSP 页面都被编译成为 Java Servlet，所以 JSP 页面就具有 Java 技术的所有优点，包括强大的存储管理功能 and 安全性。与 SERVLETS 相比，JSP 能提供所有的 SERVLETS 功能，但它比用 println 书写和修改 HTML 更方便。

6. 强大的数据库支持

JSP 可以和任何与 JDBC 兼容的数据库建立连接，操纵数据库数据。Sun 公司还开发了 JDBC-ODBC bridge，用此技术，JSP 程序就可以访问带有 ODBC 驱动程序数据库。目前大多数数据库系统都带有 ODBC 驱动程序，所以 JSP 程序能访问诸如 Oracle、Sybase、MS SQL Server 和 MS Access 等主流数据库。

7. JSP 的效率和安全性更高

ASP 以源码形式存放，以解释方式运行，每次 ASP 网页调用都需要对源码进行解释，运行效率不高。另外，IIS 的漏洞曾使得许多网站的源程序暴露。

JSP 在执行以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 机制，能提高字节码的访问效率。第一次调用 JSP 网页可能稍慢，因为它被编译成 Cache，以后就快得多了。同

时, JSP 源程序不大可能被下载, 特别是 JavaBean 程序完全可以放到不对外的目录中。

8. JSP 的组件 (Component) 方式更方便

ASP 通过 COM 来扩充复杂的功能, 如文件上载、发送 E-mail 以及将业务处理或者复杂分离出来成为独立可重复利用的模块。JSP 通过 JavaBean 实现了同样的功能扩充。在开发方面, COM 的开发远比 JavaBean 复杂和烦琐。在维护方面, COM 必须在服务器上注册, 如果修改了 COM 程序, 就必须重新注册, 甚至必须关机和重新启动。JavaBean 则不需要注册, 放在 CLASSPATH 包含的目录中就可以了。如果对 JavaBean 进行修改, 则 JSDK 和 Tomcat 还需要关闭和重新运行 (但不是关机), 但将在以后的版本中可以做到不需要关闭服务器。

另外, JavaBean 是完全的 OOP, 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, 例如用户权限控制、E-mail 自动回复等。

9. JSP 的适应平台更广

ASP 目前仅适用于 NT 和 IIS。虽然 UNIX 下有 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, 而且 UNIX 下的 COM 实现起来非常困难。

JSP 则不同, 几乎所有平台都支持 Java, 所以 JSP+JavaBean 可以在所有平台下使用。NT 下 IIS 通过一插件, 例如 JRUN (<http://www3.allaire.com/products/jrun>), 或者 ServletExec (<http://www.newatlanta.com>), 就能支持 JSP。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。

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。

1.2 JSP 与 PHP、ASP 的比较

目前, 最常用的动态网页语言有 ASP (Active Server Pages), JSP (Java Server Pages), PHP (Hypertext Preprocessor)。

1. PHP 与 ASP 的特点

(1) ASP 全名 Active Server Pages (动态服务器主页), 内含于 Microsoft Internet Information Server (IIS) 服务器当中, 提供一个服务器端的 VBScript/JavaScript 脚本运行环境, 现在的 IIS5.0 已经支持 ASP3.0 了。Web 服务器 (IIS 3.0/4.0/5.0, Microsoft Site Server 2.0/3.0) 会自动将设计的 ASP 程序解释为标准的 HTML 格式的网页内容, 在客户端的浏览器上显示出来。用户只要使用常规的浏览器, 即可浏览 ASP 程序动态生成的网页。ASP 还可通过 ActiveX 服务器端组件 (ActiveX Server Component) 来扩充功能, ActiveX Server Component 可使用 VB, Java, C++, COBOL 等语言来实现。此外, ASP 技术可以和任何脚本语言兼容。除了可使用 VBscript 或 JScript 语言来作为脚本语言, 还可通过插件的方式, 使用由第三方所提供的其他脚本语言, 如 PerlScript 语言。ASP 的源程序代码不会被传到用户的浏览器, 可以保护源程序。

(2) 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，并耦合 PHP 自己的特性，使 Web 开发者能够快速地编写出动态程序。它支持目前绝大多数数据库系统。PHP 是完全免费的，可以自由下载，甚至可以不受限制地获得源码，因而可以加进自己需要的特色。

2. PHP 与 ASP 的优缺点

(1) ASP 的优点是程序简单，容易编写，缺点是安全性不是太好，漏洞比较多。

(2) PHP 的优点是支持各种类型的数据库系统，而且可以自己定制 PHP 解释器的功能，缺点是 PHP 提供的数据库接口支持不统一，难移植。

3. JSP 与 PHP、ASP 的比较

综合 JSP 与 PHP、ASP 各自的特点，可以看出由于 PHP 缺乏规模支持、多层结构支持及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。而 ASP 和 JSP 则没有以上缺陷，ASP 可以通过 Microsoft Windows 的 COM/DCOM 获得 ActiveX 规模支持，通过 DCOM 和 Transaction Server 获得结构支持；JSP 可以通过 JavaBeans 和 EJB 获得规模支持，通过 EJB/CORBA 以及众多厂商的 Application Server 获得结构支持。

目前，在国内 PHP 与 ASP 技术应用最为广泛。但在国外，JSP 已经是比较流行的一种技术，尤其是电子商务类的网站多采用 JSP。

总体来说，JSP 应该是未来技术发展的趋势之一，世界上一些大的电子商务解决方案提供商都采用 JSP/Servlet 架构。

1.3 JSP 的运作模式

JSP 能在 Web Server 端整合 Java 语言至 HTML 网页的环境中，然后利用 HTML 网页内含的 Java 程序代码取代原有的 CGI，ISAPI 或者 IDC 的程序，以便执行原有的 CGI/WinCGI，ISAPI 的功能。

相对于 Client 端内嵌的描述语言，Sun 公司提供的 JSDK1.0.1 也支持类似的描述语言，它便是 Java 语言。JSP 的运作模式，如图 1-1 所示。

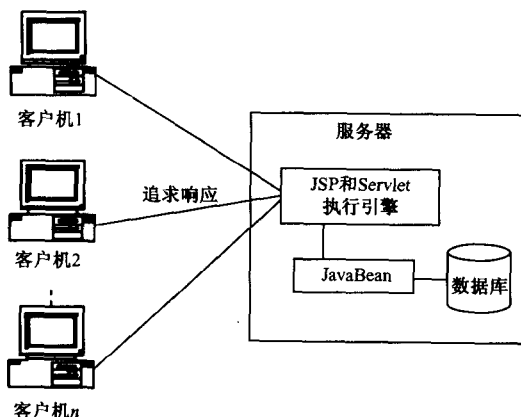


图 1-1 JSP 运作模式

在 JSP 第一次获得来自客户端浏览器的请求时, JSP 文件将被 JSP 引擎 (JSP engine) 转换成为一个 Servlet, 也就是把 JSP 文件编译成为 Java class 文件, 就是 Servlet。当 Servlet 引擎接收到请求后, 如果设置了使用最新的 JSP, 它就会去找 JSP 文件, 检查 JSP 文件在上次编译后是否改动过。如果改动过, 就会重新编译, 最终还是会把请求转交给编译好的 Servlet 引擎去运行。在编译时如果发现 JSP 文件有任何语法错误, 转换过程将中断, 并向客户端发出出错信息。如果编译成功, 则所转换产生的 Servlet 代码被编译, 然后该 Servlet 被 JSP 引擎加载到内存中。此时 JSP 引擎还请求了 JSPInit () 方法的执行, 并对此 Servlet 初始化。JSPInit () 方法在 Servlet 的周期中只被请求一次, 然后将被调用来处理客户端的请求和回复操作。对于所有随后对该 JSP 文件的请求, 服务器将检查该 JSP 文件自最后一次被存取后是否经过修改。如果没有修改, 则将请求交给还在内存中的 Servlet 的 JSPService () 方法, 执行回复操作。如果 JSP 文件被修改了, 服务器将自动对文件重新编译, 其结果将取代内存中的 Servlet, 并继续上述处理过程。

虽然 JSP 效率很高, 但是第 1 次调用时由于需要转换和编译而有一些暂短的延迟。此外, 在任何时候如果由于系统资源不足, JSP 引擎会以某种不确定的方式将 Servlet 从内存中清除。当这种情况发生时 JSPDestroy () 方法首先被调用, 然后 Servlet 实例便被标记加入“垃圾收集”处理。

1.4 建立 JSP 运行环境

为了编写和调试 JSP 程序, 需要建立一个 JSP 运行和开发环境。一个稳定和使用方便的工作平台可以极大地提高工作效率, 使程序开发更加得心应手。

由于 JSP 是跨平台的, 因此它支持 UNIX/Linux 和 Windows 系统。一般来说, JSP 程序开发工作可以从 Windows 开始, 源代码编写在 Windows 环境下进行, 然后在 UNIX/Linux 上做最后调试和运行, 这样配合可使开发效率达到最高。但是也需要了解一些 Linux 系统下的 JSP 程序开发及其工作环境的建立过程。

1.4.1 JSP 运行环境需要的组件

运行 JSP 之前, 需要安装 Java2 软件开发工具箱 (J2SDK), 以前称之为 Java 开发工具箱 (JDK)、JavaServer 网络开发工具箱 (JSWDK)、Tomcat, 或者其他能支持 JSP 的 Web 服务器。Sun 公司为 Windows、Solaris 以及 Linux 平台提供免费的 J2SDK 和 JSWDK。如果想在现有的 Web 服务器上使用 JSP, 而此服务器本身不支持 JSP 和 Java 代码片段, 可以试用 Allaire 公司的 Jrun, 它可以为 Netscape 的 Enterprise Server 和 Fasttrack Server、微软的 Internet Information Server (IIS) 和 Personal Web Server (PWS)、Apache 以及其他服务器充当 Web 服务器的附件。也可以使用 Apache 服务器的 Java 版, 其中包含最新的 JSWDK。

- J2SDK 可在 <http://java.sun.Com/jdk> 处下载;
- JSWDK 可在 <http://java.sun.Com/Products/jsp> 处下载;
- Tomcat 可在 <http://jakarta.apache.org> 处下载。

支持 JSP 的 Web 服务器必须要有相应的配置。目前流行的 Web 服务器有: JSWDK、Tomcat 及 Resin 等。

1. JSWDK

Java Server 网络开发工具箱 (JSWDK) 是 JSP 最容易使用的 Web 服务器。JSWDK 以压缩文档的方式在 Web 上提供。JSWDK 的安装只需将 jsjdk-1.4-1-Win.Zip 文件释放到硬盘目录下 (c:\等), 然后就可以在硬盘上找到\jsjdk-1.4.1\目录。如果不想保留 JSWDK 时, 删除这个目录就可以了, 没有任何系统文件和注册表遗留问题。

JSWDK 的 Web 服务器参数保存在\jsjdk-1.4.1\Web Server.Xml 中, 用 Windows 写字板打开并编辑这个文件就可以修改 JSWDK 的默认设置值。

JSWDK 默认的文档目录是\jsjdk-1.4.1\Webpages, 在此目录下可以建立子目录, 例如:\jsjdk-1.4.1\Webpages\test, 就能在浏览器中用 <http://localhost:8080/test> 访问这个目录, 并且执行其中的 JSP 程序。JSWDK 主文档在默认状态下为 index.html 和 index.jsp。也就是说访问 <http://localhost:8080> 等于访问\jsjdk-1.4.1\Webpages\index.html。

JSWDK 软件包中的示例页面分为两类, 一类是 JSP 文件; 另一类是包含一个表单的 HTML 文件, 这些表单均由 JSP 代码处理。与 ASP 一样, JSP 中的 Java 代码均在服务器端执行。因此, 在浏览器中使用“查看源文件”菜单命令是无法看到 JSP 源代码的, 而只能看到结果 HTML 代码。所有示例的源代码均通过一个单独的“examples”页面提供。JSWDK 的初始页面如图 1-2 所示。

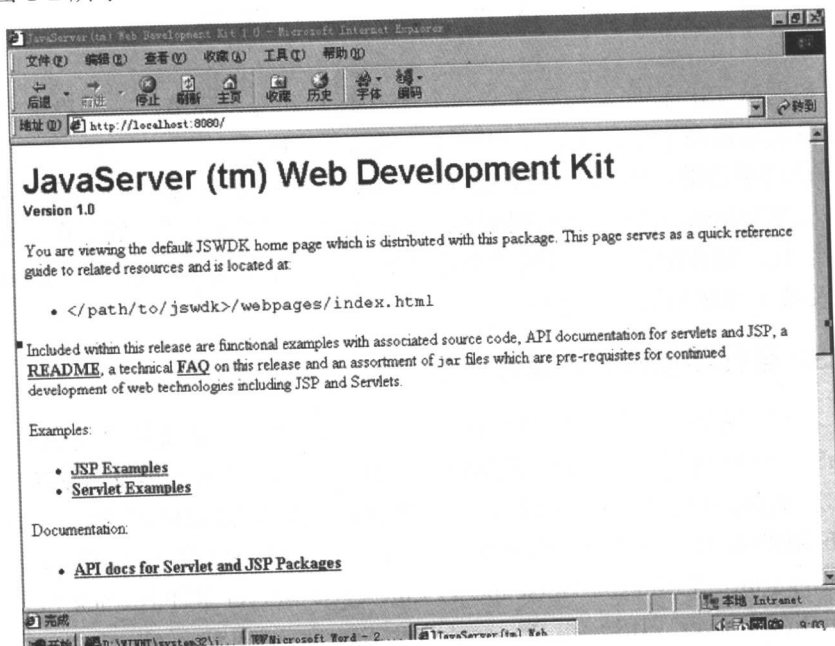


图 1-2 JSWDK 的初始页面

2. Tomcat

Tomcat 是一个 Servlet 容器和 JavaServer Pages (tm) 执行器, 它可以单独使用或和以下几个流行的网络服务器关联在一起使用。

- Apache, version 1.3 或更新版本。
- Microsoft Internet Information Server, version 4.0 或更新版本。

- Microsoft Personal Web Server, version 4.0 或更新版本。
- Netscape Enterprise Server, version 3.0 或更新版本。

Apache Tomcat 4.1/5.0 是 Apache Group Jakarta 小组开发的一个免费服务器软件，适合于嵌入 Apache 中使用。而且它的源代码可以免费获得，可以自由地对它进行扩充。不足之处是它的配置十分麻烦，而且有一些安全性的问题没有解决，例如，低版本的 Tomcat 不支持大写的 JSP 扩展名，如果用户请求大写扩展名的 JSP 程序，那么 Tomcat 可能会将 JSP 程序的源代码全部显示在客户端的浏览器上。

(1) 安装和运行 Tomcat。Tomcat 需要一个 Java 运行环境（包含在任何 Java2 平台系统里的 JRE1.1 或更新版本）。假如想开发应用，需要安装一个 Java 编译器，比如包括在 Java Development Kit1.1 或更新版本里的编译环境。

(2) 使用 Tomcat 进行应用开发。使用 Tomcat 进行 Servlet 或基于 JSP 的应用开发。

(3) 版本中的新特征。在 Tomcat5.0 版本后加入了一些新的特征，主要包括以下几方面：

- 可通过与 Microsoft's IIS 和 NetScape NES Web Server 连接来运行 Servlet，能获得与单独运行一样的效果。
- 可自动产生 Apache 的配置文件。
- 可自动调度一个应用并描述出一个 Web 文档（WAR）文件。配置一个应用，复制这个文件到 Tomcat\webapps 目录下，并且重新启动 Tomcat，WAR 文件将会自动扩展，移除一个应用，只要删除那个应用的目录以及相应的 WAR 文件即可。
- 当 Web-inf/classes 目录下的 class 文件或 Web-inf/lib/*.jar 文件有改动时自动重新装载。
- 最小限度的管理工具，允许一个新的 Web 应用被扩展到正在运行的 Tomcat 环境或从一个已经存正的应用中移除。
- 充分改进的安装和开发文档。

Tomcat 已经或者即将是一个较理想的 JSP & Servlet 开发和支撑平台。相对地，JSWDK 只是一个简化的服务器平台，性能和稳定程度都比较有限，而且实际上 Sun 并不许可将它作为互联网上的商业平台。Tomcat 的功能比 JWS 或 JSWDK 强大得多，可以访问 Tomcat 的站点 <http://jakarta.apache.org> 查看详细介绍。

Tomcat 有不同的工具来共同实现强大的功能。在 Jakarta 工程中有一些如下的相关子项目：

- Ant: 基于 Java 的跨平台开发工具，支持 XML。
- RegExp: 100%纯 Java 表达与调试工具包，可是 debug 和测试 Java 兼容性。
- Slide: 是一个包含了 Servlet API 的内容管理，用以推进 WebDAV 协议（Web-based Distributed Authoring and Versioning），从而使 servlet 可以在任何支持 API 2.2 或以上版本的容器内运行。
- Struts: 用于制作 jsp/servlet 的 web 通用应用开发框架，即 MVC（Model-View-Controller）。
- Taglibs: 是一个与 JSP1.1 标准兼容的自定义标记的库。作为一个强有力的 JSP 结构特性，这将为 JSP 开发注入更多功能，开发也变得更友好。
- Watchdog: 包含在 Tomcat5.0 版中，用于检测 Servlet 和 JSP 的兼容性。

3. Resin

Resin 提供了最快的 JSP/Servlets 运行平台。在 Java 和 JavaScript 的支持下，Resin 可以