

● 高等学校21世纪计算机教材

JSP

程序设计教程

肖金秀 冯沃辉 施鸿翔 编著

冶金工业出版社

JSP 程序设计教程

肖金秀 冯沃辉 施鸿翔 编著

内 容 简 介

本书是一本详细介绍 JSP 编程技术的实用教程,其重点是 JSP 和其他技术的综合使用。本书内容包括 JSP 概述及应用基础、Java 应用基础、JSP 语法基础、JDBC、Servlet、JavaBean 在 JSP 中的应用、JSP 与 XML 联合开发技术、网上投票制作实例、网上测试制作实例、留言板制作实例、论坛制作实例等。

本书语言简洁,内容丰富,范例具有理论与实践相结合的特点,可作为高等学校相关专业的教材和 JSP 的培训用书,也可作为 JSP 开发人员和广大用户的自学或参考用书。

图书在版编目(CIP)数据

JSP 程序设计教程 / 肖金秀等编著. —北京:冶金工业出版社, 2003.9
ISBN 7-5024-3343-0

I. J... II. 肖... 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(2003)第 072324 号

出版人 曹胜利(北京沙滩嵩祝院北巷 39 号,邮编 100009)

责任编辑 程志宏

广东惠阳印刷厂印刷;冶金工业出版社发行;各地新华书店经销

2003 年 10 月第 1 版,2003 年 10 月第 1 次印刷

787mm×1092mm 1/16; 25 印张; 612 千字; 392 页; 1-2600 册

39.00 元

冶金工业出版社发行部 电话:(010) 64044283 传真:(010) 64027893

冶金书店 地址:北京东四西大街 46 号(100711) 电话:(010) 65289081

(本社图书如有印装质量问题,本社发行部负责退换)

前 言

一、关于 JSP

JSP (Java Server Pages) 是目前十分流行的一种技术, 主要运用于开发服务端的脚本程序和动态网站的建设。它与目前流行的 ASP 技术、PHP 技术属同一性质、同一层次的, 它们在网站的建设中所起到的作用大致相同。但是 JSP 技术与后面两种技术相比, 有着自身独有的优点。

关于 JSP 技术与 ASP 技术、PHP 技术的比较在书中另有论述。这里需强调的一点是: JSP 技术有 J2EE 平台支持, 发展前途不可限量。众所周知, J2EE 平台提供了 Java 企业应用编程接口 (Java Enterprise APIs), 为企业计算以及电子商务应用系统提供了有关的技术和强大的类库支持。J2EE 平台包含十几种技术, JSP 技术正是其中的一种核心技术。J2EE 的发展势头十分迅猛, 在可以预见的将来, Sun 的 J2EE 平台可能是惟一可以与微软的 .Net 构架相互抗衡的平台。从这个意义上来说, 基于 J2EE 平台的 JSP 技术与基于 .Net 平台的 ASP、ASP+ 技术之争, 正好就是 J2EE 平台与 .Net 平台之争的折射。因此, JSP 技术以及它的基础——J2EE 技术十分值得去关注。

在国外, 采用 JSP+J2EE 技术构架电子商务网站已经开始大行其道, 应用得十分普遍。在国内, 这项技术的运用已经开始展露头角, 虽然采用这一项技术架构的网站还不多, 但这种趋势已经出现。所以, 了解、学习好 JSP/J2EE 技术, 将使自己受益非浅。

二、本书内容结构

全书共分为 12 章, 具体内容结构如下:

第 1 章: JSP 概述。主要介绍了 Java Servlet 技术, ASP、PHP 与 JSP 的比较, JSP 的结构等内容。

第 2 章: JSP 应用基础。主要介绍了 JSP 运行环境的要求、JSP 运行环境的配置、应用服务器的启动与关闭等内容, 并介绍了一个简单的 JSP 实例。

第 3 章: Java 应用基础。主要介绍了面向对象编程、Java 语言基础、对象和简单数据对象、类和继承、接口和包等内容。

第 4 章: JSP 语法基础。主要介绍了 JSP 编译器、JSP 注释、JSP 脚本元素、JSP 指令、JSP 预定义变量、JSP 动作等内容。

第 5 章: JDBC。主要介绍了 JDBC 驱动程序的类型、JDBC 应用程序的基本流程、JDBC 编程等内容。

第 6 章: Java Servlet。主要介绍了 Servlet 基本结构、Servlet API 中的几个对象、Servlet 与表单交互、读取 HTTP 请求头、访问 CGI 变量、HTTP 应答状态、HTTP 应答头、Cookie、会话状态、使用 JSP 和 Servlet 实现分页等内容。

第 7 章: JavaBean 在 JSP 中的应用。主要介绍了在 JSP 中使用 JavaBean、JavaBean 的 Scope 属性、JSP+JavaBean 的数据库操作应用、JSP+Bean 读取数据库实例、文件上传 JavaBean

等内容。

第 8 章：JSP 与 XML 联合开发技术。主要介绍了 XML 概述、XML 与 JSP 技术联合的模式、在 JSP 中应用 XML、Tag Library 开发与应用实例等内容。

第 9 章：网上投票制作实例。本章通过一个网上投票的制作实例来介绍 JSP 的简单应用。

第 10 章：网上测试制作实例。本章通过一个网上测试的制作实例来介绍 JavaBean 的具体应用。

第 11 章：留言本制作实例。本章通过制作留言本来介绍 JSP 对文件及数据库的写入操作。

第 12 章：论坛制作实例。本章通过详细介绍论坛制作的实例来加深读者对 JSP 的全面了解。

三、本书特点

本书语言简洁，内容丰富，举例典型，讲解详细。在介绍知识时，本书注重理论联系实际，通过大量的实例来加深读者对理论知识的了解。此外，本书的基础知识部分都附有练习题，书末还有参考答案，读者可以通过练习来巩固所学知识。

四、本书适用对象

本书可作为高等学校相关专业的教材和 JSP 的培训用书，也可作为 JSP 开发人员和广大用户的自学或参考用书。

由于作者水平有限，时间仓促，书中缺点和错误在所难免，恳请读者批评指正。

读者如果有好的意见或建议，可以发 E-mail 到 service@cnbook.net，也可以登录网站：<http://www.cnbook.net>，在该网站的论坛进行探讨。此外，该网站还提供了有关本书的一些学习资料，读者可以免费下载。

编 者

2003 年 8 月

目 录

第1章 JSP 概述.....	1	第3章 Java 应用基础.....	19
1.1 JSP 简介	1	3.1 Java 概述.....	19
1.1.1 JSP 的起源与背景	1	3.1.1 Java 的发展史.....	19
1.1.2 JSP 的优点.....	2	3.1.2 Java 的影响.....	20
1.2 Java Servlet 技术.....	3	3.2 面向对象编程.....	21
1.3 ASP、PHP 与 JSP 的比较.....	4	3.2.1 对象的定义.....	21
1.3.1 结构.....	4	3.2.2 消息的定义.....	22
1.3.2 性能和平台无关性	4	3.2.3 类的定义.....	23
1.3.3 Session 管理	5	3.2.4 实例和类成员.....	23
1.3.4 Application 管理.....	5	3.2.5 继承的定义.....	27
1.3.5 ServerSideIncludes	5	3.2.6 接口的定义.....	28
1.3.6 JavaBean	5	3.2.7 面向对象应用实例.....	28
1.3.7 JSP 和数据库.....	6	3.3 Java 语言基础	32
1.4 JSP 的结构	7	3.3.1 变量	32
小结.....	10	3.3.2 操作符	36
综合练习一	10	3.3.3 表达式、语句和块.....	41
一、基础题.....	10	3.3.4 控制流程语句	43
二、上机实习.....	10	3.4 对象和简单数据对象	51
第2章 JSP 应用基础.....	11	3.4.1 对象的生命周期.....	51
2.1 JSP 运行环境的要求	11	3.4.2 字符和字符串	56
2.1.1 JSP 对硬件的基本要求	11	3.4.3 数字	60
2.1.2 JSP 对软件的基本要求	11	3.4.4 数组	62
2.1.3 JSP 对操作系统的基本要求	12	3.5 类和继承.....	66
2.2 JSP 运行环境的配置.....	12	3.5.1 创建类.....	66
2.2.1 JDK 的配置.....	12	3.5.2 执行嵌套类.....	77
2.2.2 应用服务程序的配置	13	3.6 接口和包.....	78
2.3 应用服务器的启动与关闭.....	14	3.6.1 创建接口	78
2.4 一个简单的 JSP 实例.....	15	3.6.2 创建和使用包.....	80
2.4.1 测试纯 JSP 页面实例	15	小结.....	81
2.4.2 测试 Servlet	16	综合练习三.....	81
小结.....	18	一、基础题	81
综合练习二	18	二、上机实习	82
一、基础题.....	18	第4章 JSP 语法基础.....	83
二、上机实习.....	18	4.1 JSP 编译器.....	83

4.2 JSP 注释	84	5.1.4 JDBC API	112
4.2.1 HTML 注释	84	5.1.5 与 CGI 的比较	113
4.2.2 隐藏注释	84	5.1.6 JDBC 的互操作性	113
4.3 JSP 脚本元素	85	5.1.7 JDBC-ODBC 桥	114
4.3.1 声明	85	5.1.8 JDBC 的优点和缺点	114
4.3.2 表达式	86	5.2 JDBC 驱动程序的类型	115
4.3.3 Scriptlet	87	5.2.1 类型 1: JDBC-ODBC 桥	115
4.3.4 应用实例	89	5.2.2 类型 2: Java 到本地 API	115
4.4 JSP 指令	92	5.2.3 类型 3: Java 到专有网络协议	116
4.4.1 Include 指令	93	5.2.4 类型 4: Java 到本地数据库协议	117
4.4.2 Page 指令	94	5.3 JDBC 应用程序的基本流程	117
4.4.3 Taglib 指令	96	5.3.1 建立与数据库的连接	117
4.4.4 JSP 指令应用实例	96	5.3.2 执行 SQL 语句	118
4.5 JSP 预定义变量	98	5.3.3 处理结果	118
4.5.1 request	98	5.3.4 从数据库断开连接	119
4.5.2 response	98	5.4 JDBC 编程	119
4.5.3 out	98	5.4.1 程序基本结构	119
4.5.4 session	99	5.4.2 Statement 类及其子类	124
4.5.5 application	100	5.4.3 结果集 ResultSet	132
4.5.6 config	100	5.4.4 DatabaseMetaData	136
4.5.7 pageContext	100	小结	139
4.5.8 JSP 预定义变量应用实例	100	综合练习五	140
4.6 JSP 动作	102	一、基础题	140
4.6.1 <jsp:forward>动作	102	二、上机实习	140
4.6.2 <jsp:getProperty>动作	103	第 6 章 Java Servlet	141
4.6.3 <jsp:include>动作	103	6.1 Servlet 概述	141
4.6.4 <jsp:plugin>动作	104	6.2 Servlet 基本结构	142
4.6.5 <jsp:setProperty>动作	105	6.2.1 输出纯文本的简单 Servlet	143
4.6.6 <jsp:useBean>动作	107	6.2.2 Servlet 的编译和安装	143
小结	108	6.2.3 运行 Servlet	144
综合练习四	109	6.2.4 输出 HTML 的 Servlet	144
一、基础题	109	6.2.5 几个 HTML 工具函数	144
二、上机实习	109	6.3 Servlet API 中的几个对象	145
第 5 章 JDBC	110	6.4 Servlet 与表单交互	146
5.1 JDBC 概述	110	6.4.1 实例: 读取三个表单变量	147
5.1.1 JDBC 的工作机制	110	6.4.2 实例: 输出所有的表单数据	148
5.1.2 Java 作为数据库前台的优势	111	6.5 读取 HTTP 请求头	150
5.1.3 JDBC 与 SQL	111	6.6 访问 CGI 变量	152

6.7 HTTP 应答状态	155	二、上机实习	215
6.7.1 设置状态代码	155	第8章 JSP 与 XML 联合开发技术	216
6.7.2 HTTP 1.1 状态代码及其含义	155	8.1 XML 概述	216
6.7.3 实例: 访问多个搜索引擎	158	8.1.1 XML 简介	216
6.8 HTTP 应答头	160	8.1.2 XML 的语法规则	218
6.8.1 常见应答头及其含义	161	8.1.3 DTD 的应用	220
6.8.2 实例: 内容改变时自动刷新页面	162	8.1.4 XSL 简介	225
6.9 Cookie	164	8.1.5 XSL 语法	226
6.9.1 Cookie API	164	8.2 XML 与 JSP 技术联合的模式	230
6.9.2 Cookie 工具函数	166	8.3 在 JSP 中应用 XML	231
6.9.3 实例: 定制的搜索引擎界面	166	8.3.1 taglib 编译指令	231
6.10 会话状态	169	8.3.2 Tag Library 和 Tag 的原理	232
6.10.1 会话状态跟踪 API	170	8.4 Tag Library 开发与应用实例	236
6.10.2 实例: 显示会话信息	171	8.4.1 Application Tag Library 的 开发目标	236
6.11 使用 JSP 和 Servlet 实现分页	173	8.4.2 定义 TLD 文档	237
小结	177	8.4.3 编写 Tag Handler	239
综合练习六	177	8.4.4 部署 Application Tag Library	245
一、基础题	177	8.4.5 测试 Application Tag Library	245
二、上机实习	177	小结	247
第7章 JavaBean 在 JSP 中的应用	178	综合练习八	247
7.1 JavaBean 简介	178	一、基础题	247
7.1.1 JavaBean 属性	179	二、上机实习	247
7.1.2 JavaBean 的事件	182	第9章 网上投票制作实例	248
7.2 在 JSP 中使用 JavaBean	186	9.1 投票表单的设计	248
7.2.1 <jsp:useBean>操作指令	187	9.2 投票处理过程	249
7.2.2 <jsp:setProperty>操作指令	189	9.3 源程序展示	250
7.2.3 <jsp:getProperty>操作指令	190	小结	253
7.3 JavaBean 的 Scope 属性	190	第10章 网上测试制作实例	254
7.3.1 Application Scope	190	10.1 概述	254
7.3.2 Session Scope	192	10.2 实例步骤	254
7.3.3 Request Scope	194	10.2.1 试题	254
7.3.4 Page Scope	196	10.2.2 调用 JavaBean	254
7.4 JSP+JavaBean 的数据库操作应用	201	10.2.3 答题页面	254
7.5 JSP+Bean 读取数据库实例	209	10.2.4 答题结果处理	256
7.6 文件上传 JavaBean	211	10.3 实例程序展示	256
小结	214	小结	262
综合练习七	214		
一、基础题	214		

第 11 章 留言本制作实例	263	12.3 论坛版面.....	309
11.1 文件留言本.....	263	12.4 帖子模块.....	332
11.1.1 留言本填写页面.....	263	12.4.1 显示帖子.....	332
11.1.2 留言处理.....	264	12.4.2 发布帖子.....	339
11.1.3 留言显示页面.....	268	12.5 管理功能.....	345
11.2 数据库留言本.....	269	12.5.1 版主管理功能.....	345
11.2.1 创建数据库.....	269	12.5.2 论坛管理功能.....	352
11.2.2 留言本首页.....	272	12.6 其他功能.....	371
11.2.3 填写留言页面.....	276	12.6.1 查看积分功能.....	372
11.2.4 删除留言页面.....	279	12.6.2 帖子查询功能.....	378
小结.....	280	12.6.3 论坛的版面结构.....	382
第 12 章 论坛制作实例	281	小结.....	388
12.1 相关模块分析与论坛数据结构.....	281	参考答案	389
12.1.1 用户信息模块.....	281	第 1 章.....	389
12.1.2 论坛版面模块.....	281	第 2 章.....	389
12.1.3 显示帖子模块.....	282	第 3 章.....	389
12.1.4 管理模块.....	282	第 4 章.....	390
12.1.5 存储数据.....	283	第 5 章.....	390
12.2 用户信息.....	283	第 6 章.....	390
12.2.1 新用户注册.....	283	第 7 章.....	390
12.2.2 更改个人资料和查询用户资料.....	295	第 8 章.....	391

第 1 章 JSP 概述

本章对 JSP 进行了简单的介绍,让读者了解到什么是 JSP、JSP 的起源与背景、JSP 运行的优点、Java Servlet 技术、JSP 具备的功能以及 JSP 的结构等知识。

1.1 JSP 简介

如今 IT 行业涌现出一大批能够构建 Web 的应用系统、支持动态处理的程序设计技术,例如:ASP、CGI 等。当前,新技术和新的程序开发软件层出不穷,可谓是“百舸争流”。

20 世纪末才匆匆登场的 Java Server Pages (JSP) 技术,一出世就吸引了世界各大软件开发者和程序设计者的浓厚兴趣,表现出无比强大的生命力。由于 Java Server Pages 技术不但能够实现跨平台和兼容各种 Web 服务器产品,而且还把强大的服务器端的 Java 技术 (Server-side Java Technology) 有机地融合到了 HTML 网页构造中去。因此,利用 Java Server Pages 技术可以建立先进、安全和跨平台的动态网站。

在 Sun 正式发布 JSP (Java Server Pages) 之后,这种新的 Web 应用开发技术很快引起了人们的关注。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 的说法,JSP 能够适应市场上包括 Apache Web Server、IIS 4.0 在内的 85% 的服务器产品。

JSP (Java Server Page) 是一种在服务器端运行的页面脚本,类似于 ASP (Active Server Page)。JSP 提供了一种简单快速的基于动态内容显示的建立站点的技术。在 JSP 文件里可以包含 HTML 标志、NCSA 标志、Servlet 标志和 JSP 语法,使得 HTML 的编码从 Web 页面的业务逻辑中有效地分离出来。

JSP 不但拥有 Servlet 的所有特性与优点,更增加了程序开发上的弹性,除了 Java 程序本身跨平台的优点,它是直接在 HTML 中内嵌 JSP 程序代码,使得程序开发更为容易、方便。JSP 是完全架构在 Servlet 程序上,不同的是,JSP 程序是由 JSP Engine 首先将之转换成相对应的 Servlet 程序代码,接着将它编译成类文件载入执行,这样如同 Servlet 一样了。

JSP 技术很容易整合到多种应用体系结构中,以利用现存的工具和技巧,并且扩展到能够支持企业级的分布式应用。作为采用 Java 技术家族的一部分,以及 Java 2 (企业版体系结构) 的一个组成部分,JSP 技术能够支持高度复杂的基于 Web 的应用。

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,因此 JSP 页面便具有 Java 技术的所有好处,包括健壮的存储管理和安全性。

作为 Java 平台的一部分,JSP 拥有 Java 编程语言“一次编写,到处都可运行”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,用户可以使用自己所选择的服务器和工具,更改工具或服务器并不影响当前的应用。

当与 Java 2 平台、企业版 (J2EE) 和 Enterprise JavaBean 技术整合时,JSP 页面将提供企业级的扩展性和性能,这对于在虚拟企业中部署基于 Web 的应用是必需的。

1.1.1 JSP 的起源与背景

溯源 Java,可追至 1991 年 4 月 Sun 公司的“Green Project (绿包计划)”。早期研究小组

成员包括 Patrick Navghon, Jame Gosling 和 Mike Sheridan 等科学家。“Green Project (绿包计划)”的最初目的是开发出一种适用于家用电器的操作软件。设计者 James Gosling 当时引进一个新语言 Oak, 保留 C++ 的语法, 但省略了明确的资源引用、指针算法与操作符过载等潜在的危害系统的特性。由于项目需要该语言具有可移植性, 即应当能够运行在多种不同的微处理器芯片和环境中, Oak 被实现为平台无关性; 而这恰好适合 Internet 编程, 很快被应用于 Web 页的查看与制作等其他网络编程。1995 年, Oak 更名为 Java 并在 Sun World 95 中正式发布。

Java 是一种软件编程语言, 旨在为 Internet 发展而使网页由静态转为动态, 并可以生成“小应用程序”。它具有阻碍电脑病毒传播的功能, 使用者无需花费许多时间学习; Java 突破了客户机器环境和 CPU 结构的语言, 可做到“编写一次, 哪里都可运行 (Writing Once, Running Everywhere)”。

开始命名这种软件语言时, 有若干选择供考虑: 例如 Oak (橡树)、Silk (丝绸) 等, 但最终还是由早期研发小组成员们所喜爱的热腾香浓的咖啡之中获得灵感, 据说他们在研发期间饮用了大量的咖啡, 而一致推崇 Java (爪哇) 所产咖啡为最佳。

但是在把 Java 推向程序设计和开发人员时, 总是有一些问题: 首先是 Java 缺少主流系统的广泛支持, 而在所有的平台和技术上都缺少可以直接参照的应用方式。这时的 Java 就开始注意到各个应用层的工作, 在所有可能的应用层上积极进取, 并在自己擅长的领域走出了坚定的脚步。

让高级程序开发人员感到兴奋的就是 Java Servlet Pages 技术的出现和逐渐成熟, 这关键性的一步, 让 Java 正式走向了 Web 技术和应用的绚丽舞台。Java 利用了它的跨平台技术, 使 Java 应用能快速部署在各个平台上, 一处编写哪里都可运行。它有效地利用了这个优势, 巧妙地将 Java 的通用性、JSP 的易用性、Servlet 的速度结合在一起, 利用平台部署上的技巧有效地屏蔽了 Java 的速度问题, 使 Java 能够迅速地在 Web 程序设计和开发人员中推广开来。

携着 Java Bean、EJB、Java Application Server、Java 事务服务器等强大的后台应用功能, Java 语言开始真正地走向了成熟的应用。

1.1.2 JSP 的优点

JSP 在多个方面加速了动态 Web 页面的开发。

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 标识来设计和格式化最终页面。

使用 JSP 标识或者小脚本来生成页面上的动态内容。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, 或者包含文件), 并且将结果以 HTML (或者 XML) 页面的形式发送回浏览器。这有助于作者保护自己的代码, 而又保证任何基于 HTML 的 Web 浏览器的完全可用性。

2. 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 (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 页面开发人员不会都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。与 JSP 相关的 XML 标识能够访问 JavaBeans 组件, 并且能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自己熟悉的工具以及由自己定制的、用于执行特定功能构件的标识库相结合进行工作。

JSP 技术很容易整合到多种应用体系结构中, 以利用现存的工具和技巧, 并且扩展到能够支持企业级的分布式应用。

作为采用 Java 技术家族的一部分, 以及 Java 2 (企业版体系结构) 的一个组成部分, JSP 技术能够支持高度复杂的基于 Web 的应用。

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, JSP 页面就具有 Java 技术的所有好处, 包括健壮的存储管理和安全性。

作为 Java 平台的一部分, JSP 拥有 Java 编程语言“一次编写, 到处都可运行”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, 用户可以使用自己所选择的服务器和工具, 更改工具或服务器并不影响当前的应用。

1.2 Java Servlet 技术

Servlet 是 Java 技术对 CGI 编程的回答。Servlet 程序在服务器端运行, 动态地生成 Web 页面。与传统的 CGI 和许多其他类似 CGI 的技术相比, Java Servlet 具有更高的效率、更容易的使用、更强大的功能、更好的可移植性和更能节省投资等优点, 下面详细介绍它的优点:

(1) 高效。在传统的 CGI 中, 每个请求都要启动一个新的进程, 如果 CGI 程序本身的执行时间较短, 启动进程所需要的开销很可能反而超过实际执行时间。而在 Servlet 中, 每个请求由一个轻量级的 Java 线程处理 (而不是重量级的操作系统进程)。在传统 CGI 中, 如果有 N 个并发的对同一 CGI 程序的请求, 则该 CGI 程序的代码在内存中重复装载了 N 次; 而对于 Servlet, 处理请求的是 N 个线程, 只需要一份 Servlet 类代码。在性能优化方面, Servlet 也比 CGI 有着更多的选择, 比如缓冲以前的计算结果, 保持数据库连接的活动等等。

(2) 方便。Servlet 提供了大量的实用工具例程, 例如自动地解析和解码 HTML 表单数据、读取和设置 HTTP 头、处理 Cookie、跟踪会话状态等。

(3) 功能强大。在 Servlet 中, 许多使用传统 CGI 程序很难完成的任务都可以轻松地完成。例如: Servlet 能够直接和 Web 服务器交互, 而普通的 CGI 程序不能。Servlet 还能够各个程序之间共享数据, 使得数据库连接池之类的功能很容易实现。

(4) 可移植性好。Servlet 用 Java 编写, Servlet API 具有完善的标准。因此, 为 I-Planet Enterprise Server 写的 Servlet 无需任何实质上的改动即可移植到 Apache、Microsoft IIS 或者 WebStar。几乎所有的主流服务器都直接或通过插件支持 Servlet。

(5) 节省投资。不仅有许多廉价甚至免费的 Web 服务器可供个人或小规模网站使用, 而且对于现有的服务器来说, 如果它不支持 Servlet 的话, 要加上这部分功能也往往是免费的 (或只需要极少的投资)。

1.3 ASP、PHP 与 JSP 的比较

从微软推出了 ASP (Active Server Page) 后, 它就以其强大的功能, 简单易学的特点而受到广大 Web 开发人员的喜欢。但是它却有微软产品的通病, 只能在 Windows 平台下使用, 虽然它可以通过增加控件而在 Linux 下使用, 但是其功能最强大的 DCOM 控件却不能使用。PHP (Hypertext Preprocessor) 是一种跨平台的服务器端嵌入式语言, 在 UNIX 平台应用比较广泛。

PHP 是完全免费的, 也支持大多数的数据库。但是, PHP 缺乏规模支持而且对多层结构也缺乏支持, 对于大型网站来说, 应用 PHP 会造成负担过重。而 Sun 公司在 Java 的基础下开发出的 JSP (Java Server Pages) 实现了动态页面与静态页面的分离, 脱离了硬件平台的束缚, 以及编译后运行等方式大大提高了其执行效率而逐渐成为因特网上的主流开发工具。

1.3.1 结构

JSP 和 ASP 在结构上类似, 都是以 “<%” 和 “%>” 作为标志符, 不同的是在标志符之间的代码 ASP 为 JavaScript 或 VBScript 脚本, 而 JSP 为 Java 代码。而 PHP 是以 “<?” 和 “?>” 作为标志符, 所用脚本可以为多种语言, 如 Perl、C、C++ 等。JSP 将网页的表现形式和服务器的代码逻辑分开。作为服务器进程的 JSP 页面, 首先被转换成 Servlet (一种服务器端运行的 java 程序)。

Servlet 支持 HTTP 协议的请求和响应。当 JSP 被转换成纯 Java 代码。多个用户同时请求一个 JSP 页面时, 应用实例化线程来响应请求。

这些线程由 Web 服务器进程来管理, 和 ASP 的线程管理器功能类似。同 CGI 为每个请求创建一个进程的模式比较, 这种方式效率高得多。

1.3.2 性能和平台无关性

与 C++、VB 等语言比较, Java 被看作是效率不太高的语言, 但它以牺牲效率换来了平台无关性, 使 Java 可以在大多数操作系统上运行而不需要重新编译——许多人愿意为此而付出代价。

Java 的编译代码是一种字节代码, 在运行时由操作系统上的一个 Java Virtual Machine (JVM) 虚拟机解释。字节代码可以在所有平台上迁移, 而不需要任何改动。交叉平台编程倾向于采用所有操作系统的 “最小公分母”, 尤其是使用图形化界面时 (Servlets 不需要任何图形界面), Servlets 比传统的 Java 程序 (Applets, javaapp) 要快, 因为它们在服务器端运行, 不需要加载沉重的 GUI (HTML 的 GUI 是非常少的)。另外, Servlets 的字节代码只有在客户请求时才执行, 所以尽管当首次调用 Servlets 时会有几秒钟的加载时间, 但后续的请求

相应非常迅速,因为服务器已经缓存了运行的 Servlets。当前的 JSP 服务器,都带有 Java 即时编译器 (JIT),因此, JSP 的执行比每次都要解释执行的 ASP 代码要快,尤其是在代码中存在循环操作时, JSP 的速度要快 1 到 2 个数量级。

1.3.3 Session 管理

为了跟踪用户的操作状态, ASP 应用 Session 对象。JSP 使用一个叫 HttpSession 的对象实现同样的功能。Session 的信息保存在服务器端, Session 的 id 保存在客户机的 Cookie 中。如果客户机禁止 Cookie, Session Id 就必须附在 URL 后面。Session 一般在服务器上设置了一个 30 分钟的过期时间,当客户停止活动后自动失效。Session 中保存和检索的信息数据类型不能是基本数据类型,如 int、double 等数据类型不能作为 Session 中保存和检索的信息,而必须是 Java 的相应的 Object (对象) 如 (Integer, Double)。例如下面的程序代码示例:

```
HttpSession session=request.getSession(); //得到一个 session 对象或创建一个
session.getId(); //得到 sessionID 数
Integer id=(Integer)session.getValue("id"); //检索出 session 的值并转化为整型
session.putValue("ItemValue", itemName); //ItemValue 必须不是 must 简单类型
```

1.3.4 Application 管理

有时服务器需要管理面向整个应用的参数,使得每个客户都能获得同样的参数值。和 Session 一样, ASP 使用 Application 对象,而 JSP 使用 ServletContext 对象,操作的方法和 Session 一样。

```
getServletContext().setAttribute("Item", ItemValue); //设置一个应用变量
Integer i=(Integer)getServletContext().getAttribute("ItemName"); //得到//item
```

1.3.5 ServerSideIncludes

在服务器端引用 ASP 和 JSP 有着相同之处。Include 功能在服务器端引用 ASP 和 JSP 有着相同之处,即 ASP 和 JSP 都支持服务器上实现服务器端包含虚文件。但 JSP 是将动态页面的结果包含进来,而不是包含文件的代码本身。当包含的文件在另一个服务器上时,不包含任何代码和对象是一个非常有效的功能。

```
ASP: <!--#includefile="subfile.asp"-->
JSP: <%@includefile="subfile.jsp"%>
```

1.3.6 JavaBean

JavaBean 是一些完成预定义功能的封装的对象数据。JavaBeans 和 JSP 接合与 COM 和 ASP 相比如下:

(1) COM 对象常用来封装商业逻辑和为 ASP 页面完成高强度计算。重用的组件使得页面简单快速,因为组件由编译语言 (C++, VB) 构成,而不是解释型的 Scripting 语言 (VBScript, JScript)。

(2) JavaBean 只能用 Java 语言开发, COM 可以由符合标准对象模型的任何语言开发 (包括 Visual J++)。另一方面, JavaBean 更容易开发,因为一旦掌握了 Java, 了解 JavaBean 的结构便非常容易。由于 COM 的复杂性,需要大量的学习和培训才能掌握 (即使你具有 C++ 和 VB 的基础也需要额外的努力)。COM 组件需要在服务器上注册,如果你修改了一个现有

的组件,服务器需要重新启动才能使用它。而创建或修改 JavaBean 不需要重新注册及重新启动服务器即可使用,对于不需要得到对服务器完全访问权限的开发者来说,这是一个巨大的优点。

(3) 可以创建满足商业逻辑的完整的 JavaBean 库,让非编程人员使用库来开发动态网站。

(4) Bean 可以在服务器端管理数据库连接。

(5) JavaBean 符合结构化对象模型:每个 Bean 由一个不带参数的构造函数,控制它的 Servlet 可以使用内置的(Introspection)来设置其属性。要设置 Bean 的内置属性,必须使用带有属性名的 setProperty 标签。如果一个属性可以设置,Bean 需要有一个 setXxxx 方法,Xxxx 用实际的属性名来代替。

1.3.7 JSP 和数据库

ASP 使用 ODBC 通过 ADO 连接数据库,而 Java 通过一个叫 JDBC 的技术连接数据库。目标数据库需要一个 JDBC 驱动程序——一个数据库与 Java 的接口,使得 Java 可以用标准的方式访问数据库。JDBC 不使用服务器端的数据源。只要有 JDBC 驱动程序,Java 就可以访问数据库了。

如果一个特定的数据库没有 JDBC 驱动程序,而只有 ODBC 驱动程序,Java 提供一个 JDBC-ODBC 桥来将 JDBC 调用转化为 ODBC 调用。所有的 Java 编译器都带有一个免费的 JDBC-ODBC 桥。理论上,桥可以访问任何常见的数据库产品。

JSP 模型是在 ASP 之后定义的,它借用了 ASP 的许多优点,如 Session、Application 等对象。

同时 JSP 使用灵活而强大的 Java 语言,而不是低效的 Scripting 语言。ASP 开发者只能使用基于 Windows 平台的技术,而 Java 和 JSP 是跨平台的。

JSP 与 Microsoft 的 ASP 技术非常相似。两者都提供在 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的能力。在 ASP 或 JSP 环境下,HTML 代码主要负责描述信息的显示样式,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分析和执行程序代码。程序代码的执行结果被重新嵌入到 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,客户端浏览器不需要任何附加的软件支持。

ASP 的编程语言是 VBScript 之类的脚本语言,JSP 使用的是 Java,这是两者最明显的区别。此外,ASP 与 JSP 还有一个更为本质的区别:两种语言引擎用完全不同的方式处理页面中嵌入的程序代码。在 ASP 下,VBScript 代码被 ASP 引擎解释执行;在 JSP 下,代码被编译成 Servlet 并由 Java 虚拟机执行,这种编译操作仅在对 JSP 页面的第一次请求时发生。

为了简明起见,表 1-1 将 JSP、ASP、PHP 等的性能列表做了一个比较。

表 1-1 JSP 与 ASP 等几种语言的比较

名称 特性	JSP	ASP	PHP	ISAPI	CGI(Perl)
易学性	容易	很容易	很容易	难	较难
运行速度	快	较快	较快	很快	慢

续表 1-1

名称 特性	JSP	ASP	PHP	ISAPI	CGI(Perl)
运行开销	小	较大	较大	小	较大
平台	均可	Windows	Windows/UNIX	Windows	均可
扩展性	很好	好	不好	不好	不好
安全性	好	不好	好	一般	好
分布式处理	支持	支持	不支持	支持	不支持
函数支持	广	不广	广	广	不广
数据库支持	多	多	多	多	多
应用程度	较广泛	较广泛	较广泛	少	少
厂商支持程度	广泛	少	较广泛	少	广泛
对 XML 的支持	支持	不支持	支持	不支持	不支持
对组件的支持	支持	支持	不支持	支持	不支持

1.4 JSP 的结构

JSP 技术推出的目的其实是起源于如何解决使用一些更通用、更富有表现力的工具和方法来发展 Servlets 技术。

一般来说, JSP 程序的运行有两个阶段: 编译阶段和请求过程阶段。如果 JSP 页面没有变化, 编译阶段将只运行一次。如果页面中没有任何语法错误, JSP 页面的编译阶段完成后将变成一个 class 文件, 实现 Servlet 接口界面的功能, 如图 1-1 所示。

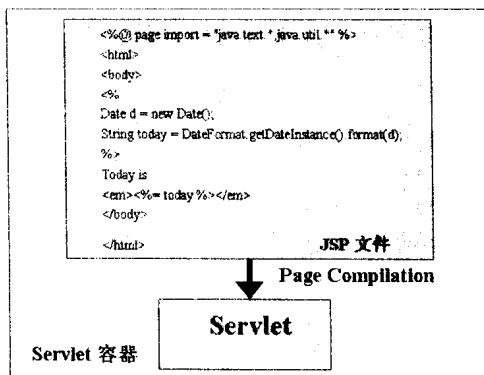


图 1-1

编译阶段的工作一般由 JSP 引擎自己来实现, 当 JSP 引擎第一次接收到来自 JSP 网页的请求时, JSP 引擎将自动启动实现编译。JSP1.1 及以上的版本还支持预编译, JSP 网页可以被预编译为 class 文件。预编译在有些时候是非常有用的。比如, 在一个页面中嵌入 JSP 部分要响应一个来自客户端的请求时, 如果先对 JSP 部分进行预编译, 将可以尽可能地抵消启动编译带来的程序在时间上滞后, 给用户的感受将是页面响应速度更快、界面变得更友好。在编译阶段有许多细节问题(如源文件和类文件存放的位置)是和具体程序的调试及运行很有关系的。

例如, 一个 helloworld.jsp 的源程序如下:

```
<%@ page language="Java" %>
```

```

<%@ page info="a hello world example" %>
<html>
<head>
<title>Hello, World</title>
</head>
<body>
<h1>
<%out.println(" <center><br>Hello, World! <br>");%>
<%out.println(" This is my first JSP Program. </center>");%>
</h1>
</body>
</html>

```

程序运行的效果如图 1-2 所示。

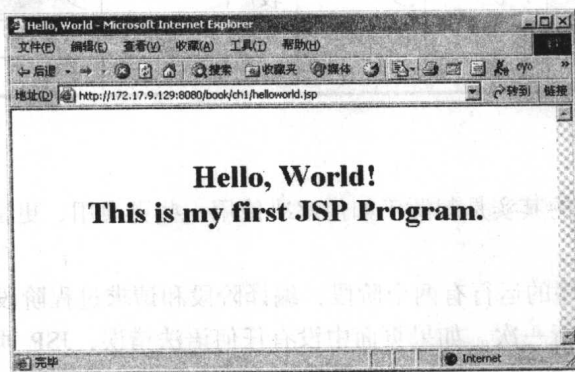


图 1-2

以上程序在 JSP 引擎上所得到的 Java 文件的源代码，如下面的程序所示：

```
package ch_00031;
```

```

import javax.servlet.*;
import javax.servlet.http.*;
import javax.servlet.jsp.*;
import javax.servlet.jsp.tagext.*;
import java.io.PrintWriter;
import java.io.IOException;
import java.io.FileInputStream;
import java.io.ObjectInputStream;
import java.util.Vector;
import org.apache.jasper.runtime.*;
import java.beans.*;
import org.apache.jasper.JasperException;

public class _0002fch_00031_0002fhelloworld_0002ejspelloworld_jsp_5 extends
    HttpJspBase {
    //begin [file="C:\\JBuilder6\\jakarta-tomcat-3.2.3\\webapps\\book\\ch1
    \\helloworld.jsp";from=(1,0);to=(1,40)]
    public String getServletInfo() {
        return "a hello world example";
    }
}

//end
static {

```