



教育部职业教育与成人教育司推荐教材  
(5年制)高等职业学校计算机技术专业教学用书

# JSP-Servlet

# 基础案例教程

◎ 向传杰 主编



电子工业出版社  
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

<http://www.phei.com.cn>

教育部职业教育与成人教育司推荐教材  
(5年制) 高等职业学校计算机技术专业教学用书

# JSP-Servlet 基础案例教程

向传杰 主编

電子工業出版社

Publishing House of Electronics Industry

北京·BEIJING

## 内 容 简 介

很多基于网络的多层结构信息系统都采用 Java 系列的 JSP-Servlet 进行开发。JSP-Servlet 程序编写简单。本书注重实用, 兼顾基础, 介绍 JSP-Servlet 的基础理论知识和实际编程知识。全书共分 11 章, 介绍 Java 技术, Java Servlet 及其特点, JSP 及其特点, 用 JSP 开发 Web 的几种主要方式, 常用开发工具的选择, JBuilder 9 的安装, Web 服务器 Tomcat, Servlet 程序设计, 表单, 处理 HTTP 实例, JSP 语法实例, Servlet 处理 Cookie 和会话, JSP-Servlet 数据库编程, 以及一个用 JSP-Servlet 开发的完整的交互式电子政务系统(企业外出网上备案系统)的开发和发布运行。

本书可作为高等职业教育计算机及相关专业的教材, 也可供需要了解或实际应用 JSP-Servlet 开发交互式网站系统的读者参考。

为了方便教师教学, 本书还配有电子教学参考资料包(包括教学指南、电子教案和习题答案), 详见前言。

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-Servlet 基础案例教程 / 向传杰主编. —北京: 电子工业出版社, 2006.12

教育部职业教育与成人教育司推荐教材. (5 年制) 高等职业学校计算机技术专业教学用书  
ISBN 7-121-03560-X

I. J… II. 向… III. JAVA 语言—程序设计—高等学校: 技术学校—教材 IV. TP312

中国版本图书馆 CIP 数据核字(2006)第 142294 号

责任编辑: 李 影 刘真平

印 刷: 北京季蜂印刷有限公司

装 订: 三河市万和装订厂

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

开 本: 787×1092 1/16 印张: 12.5 字数: 320 千字

印 次: 2006 年 12 月第 1 次印刷

印 数: 4000 册 定价: 18.00 元

凡所购买电子工业出版社图书有缺损问题, 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; 邮购电话: (010) 88254888。

质量投诉请发邮件至 zlt@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线: (010) 88258888。

# 前 言



近年来,随着全国各行业信息化的普及,各层次的计算机人才需求旺盛,特别是对具有熟练编程技能的程序员的需求越来越大。目前,在各行业对计算机各类人才的需求中,对交互式网站开发人才的需求更为迫切,各类信息系统的开发基本上都基于交互式 Web 技术,交互式 Web 成为很多信息系统中不可或缺的一部分。

三层甚至多层体系结构的交互式 Web 技术开发涉及面很广,本书试图对其中的基本技术 JSP-Servlet 进行较为系统的理论介绍,在此基础上详细介绍 JSP-Servlet 的具体开发技术,最后用 JSP-Servlet 开发一个完整的电子政务系统。全书共分 11 章,系统介绍了 Java 技术基础,Java Servlet 及其特点, JSP 及其特点,用 JSP 开发 Web 的几种主要方式,常用开发工具的选择, JBuilder 9 的安装, Web 服务器 Tomcat, Servlet 程序设计, 表单, 处理 HTTP 实例, JSP 语法实例, Servlet 处理 Cookie 和会话, JSP-Servlet 数据库编程, 以及一个用 JSP-Servlet 开发的完整的交互式电子政务系统(企业外出网上备案系统)的开发和发布运行。

本书由向传杰主编。毕业于北京理工大学计算机应用专业的向传杰博士长期从事网站开发,企业信息化建设,电子商务/电子政务、ERP、计算机网络相关技术的开发、管理和培训的工作,有较强的实战经验。

本书由天津师范大学马希荣、天津电子信息职业技术学院傅连仲主审,通过教育部认定,作为教育部职业教育与成人教育司推荐教材。

由于作者水平有限,书中不当和谬误之处在所难免,敬请读者批评指正。

为了方便教师教学,本书还配有教学指南、电子教案及习题答案(电子版),请有此需要的教师登录华信教育资源网([www.huaxin.edu.cn](http://www.huaxin.edu.cn) 或 [www.hxedu.com.cn](http://www.hxedu.com.cn))免费注册后进行下载。有问题时请在网站留言板留言或与电子工业出版社联系(E-mail:[hxedu@phei.com.cn](mailto:hxedu@phei.com.cn))。

编 者

2006 年 10 月



# 目 录



<b>第 1 章 Servlet 和 JSP 概述</b> .....	(1)
1.1 Java 技术 .....	(1)
1.1.1 Java 技术的历史 .....	(2)
1.1.2 JDBC .....	(3)
1.2 Java Servlet 及其特点 .....	(3)
1.3 JSP 及其特点 .....	(4)
1.3.1 JSP 技术概述 .....	(4)
1.3.2 JSP 的优势 .....	(6)
1.3.3 用 JSP 开发 Web 的几种主要方式 .....	(7)
本章小结 .....	(8)
习题 .....	(8)
<b>第 2 章 安装相关软件</b> .....	(9)
2.1 常用开发工具的选择 .....	(9)
2.1.1 IBM VisualAge for Java .....	(9)
2.1.2 JBuilder .....	(11)
2.1.3 JRun .....	(13)
2.2 Servlet 和 JSP 开发工具 JBuilder .....	(14)
2.2.1 选择 JBuilder 作为开发工具 .....	(14)
2.2.2 JBuilder 9 的安装 .....	(15)
2.2.3 JBuilder 9 的开发环境介绍 .....	(19)
2.3 支持 Servlet 和 JSP 的 Web 服务器 Tomcat .....	(27)
2.3.1 Tomcat 介绍 .....	(27)
2.3.2 Tomcat 安装 .....	(28)
2.3.3 Tomcat 系统运行 .....	(30)
本章小结 .....	(30)
习题 .....	(30)
<b>第 3 章 Java 技术基础</b> .....	(31)
3.1 Java 语言规则 .....	(31)
3.2 基本数据类型、标识符及关键字 .....	(32)
3.2.1 Java 基本数据类型 .....	(32)
3.2.2 Java 语言的标识符 .....	(33)
3.2.3 Java 关键字 .....	(33)
3.3 表达式和运算符 .....	(34)
3.3.1 赋值运算符 .....	(34)
3.3.2 算术运算符 .....	(35)

3.3.3	关系运算符与逻辑运算符	(35)
3.4	流程控制	(36)
3.4.1	条件控制语句	(36)
3.4.2	循环控制语句	(38)
3.5	其他 Java 基础知识	(39)
3.5.1	子类	(39)
3.5.2	类的类型	(40)
3.5.3	接口	(41)
3.5.4	包	(41)
3.6	Java 编码约定	(42)
	本章小结	(43)
	习题	(43)
<b>第 4 章</b>	<b>Servlet 程序设计</b>	<b>(44)</b>
4.1	Servlet 基本结构	(44)
4.1.1	Servlet 概述	(44)
4.1.2	用 JBuilder 开发 Java Servlet	(44)
4.1.3	Java Servlet API 2.2 简介	(50)
4.2	能简单回复的 Servlet	(53)
4.2.1	需求	(53)
4.2.2	代码修改	(54)
4.2.3	发布 Servlet 到服务器	(55)
4.2.4	运行 Servlet	(55)
	本章小结	(56)
	习题	(56)
<b>第 5 章</b>	<b>表单</b>	<b>(57)</b>
5.1	表单基础知识	(57)
5.1.1	表单概述	(57)
5.1.2	表单的组成	(58)
5.2	制作表单	(63)
5.2.1	表单的属性	(63)
5.2.2	构造表单	(63)
5.3	运行页面	(68)
5.4	构造 Servlet 来处理表单	(68)
5.4.1	构造一个基本的 Servlet Ser_Chulibiaodan	(69)
5.4.2	添加处理语句	(73)
5.4.3	测试	(74)
	本章小结	(75)
	习题	(75)
<b>第 6 章</b>	<b>处理 HTTP 实例</b>	<b>(76)</b>
6.1	HTTP 请求头	(76)

6.1.1	HTTP 请求头概述	(76)
6.1.2	在 Servlet 中读取请求头	(77)
6.1.3	用 Servlet 输出所有的请求头	(77)
6.2	HTTP 应答状态	(79)
6.2.1	状态代码的组成和作用	(79)
6.2.2	设置状态代码	(80)
6.2.3	HTTP 1.1 状态代码及其含义	(80)
6.2.4	可访问多个搜索引擎的 Servlet	(82)
6.3	设置 HTTP 应答头	(88)
6.3.1	HTTP 应答头概述	(88)
6.3.2	常见应答头及其含义	(88)
	本章小结	(89)
	习题	(89)
<b>第 7 章</b>	<b>JSP</b>	(90)
7.1	JSP 概述	(90)
7.2	在 JBuilder 环境下开发 JSP	(91)
7.2.1	生成基本 JSP 代码	(91)
7.2.2	在 JBuilder 环境下运行 JSP	(95)
7.2.3	把 JSP 发布到 Tomcat 服务器上	(96)
7.3	JSP 语法概要	(97)
7.4	JavaBean 组件	(98)
	本章小结	(98)
	习题	(99)
<b>第 8 章</b>	<b>JSP 语法</b>	(100)
8.1	JSP 的语法和语义	(100)
8.2	JSP 语法实例	(101)
8.2.1	显示时间	(101)
8.2.2	Scriptlet	(101)
8.2.3	Scriptlet 和 HTML 的混合	(102)
8.2.4	JSP Tag	(104)
8.2.5	JSP Session	(105)
8.2.6	Bean 的处理	(108)
8.3	JSP 内置组件	(110)
8.4	JSP 实例	(111)
8.4.1	JSP 读文件/写文件的例子	(111)
8.4.2	JSP 计数器例子	(112)
8.4.3	用 JSP 显示进度条例子	(115)
	本章小结	(121)
	习题	(121)

<b>第 9 章 Servlet 处理 Cookie 和会话</b> .....	(122)
<b>9.1 Cookie</b> .....	(122)
9.1.1 Cookie 概述 .....	(122)
9.1.2 Servlet 的 Cookie API .....	(122)
9.1.3 在应答头中设置 Cookie .....	(123)
9.1.4 读取保存到客户端的 Cookie .....	(124)
9.1.5 处理 Cookie 的 Servlet .....	(124)
<b>9.2 会话</b> .....	(127)
9.2.1 会话状态概述 .....	(127)
9.2.2 会话状态跟踪 API .....	(128)
9.2.3 在会话对象中保存数据 .....	(129)
9.2.4 显示会话信息 .....	(130)
本章小结 .....	(132)
习题 .....	(132)
<b>第 10 章 JSP/Servlet 数据库编程</b> .....	(133)
<b>10.1 Java 数据库编程基础</b> .....	(133)
10.1.1 JDBC API .....	(134)
10.1.2 Java 操作数据库的基本过程 .....	(136)
10.1.3 常用数据库的 JSP/Servlet 连接基本操作 .....	(137)
<b>10.2 编写一个读取数据库数据的程序</b> .....	(144)
10.2.1 创建数据库和表 .....	(144)
10.2.2 配置数据源 .....	(146)
10.2.3 JSP 把数据写入数据库表 .....	(148)
10.2.4 JSP 读取数据库表中数据 .....	(151)
10.2.5 Servlet 读取数据库表中数据 .....	(152)
<b>10.3 SQL</b> .....	(154)
本章小结 .....	(155)
习题 .....	(156)
<b>第 11 章 企业外出网上备案系统开发</b> .....	(157)
<b>11.1 需求分析</b> .....	(157)
<b>11.2 方案设计</b> .....	(157)
11.2.1 基本方案 .....	(157)
11.2.2 数据库方案 .....	(158)
<b>11.3 代码开发</b> .....	(158)
11.3.1 基本文件组成 .....	(158)
11.3.2 各个文件说明 .....	(159)
<b>11.4 发布运行</b> .....	(175)
本章小结 .....	(175)
习题 .....	(175)
<b>附录 A 习题答案</b> .....	(176)



# 第 1 章 Servlet 和 JSP 概述



知识要点:

- Java 技术、Java 技术的历史和 JDBC;
- Java Servlet 及其特点;
- JSP 及其特点, JSP 技术概述, JSP 的优势, 用 JSP 开发 Web 的几种主要方式。

## 1.1 Java 技术

在 Java 出现以前, Internet (因特网) 上的信息内容都是一些乏味死板的 HTML 文档。这对于那些迷恋于 Web 浏览的人们来说简直不可容忍。他们迫切希望能在 Web 中看到一些交互式的内容, 开发人员也极希望能在 Web 上创建一类无须考虑软硬件平台就可以执行的应用程序, 同时这些程序还要有极大的安全保障。对于用户的这种要求, 传统的编程语言显得无能为力, 而 Sun 公司的工程师们敏锐地察觉到这一点。他们开始将 Java 技术应用于 Web 上, 并且开发出 HotJava 的第一个版本。当 Sun 公司 1995 年正式将其以 Java 这个名字推出的时候, 几乎所有的 Web 开发人员都感觉这正是他们想要的。于是 Java 成了一颗耀眼的明星, 丑小鸭一下变成了白天鹅。有人预言 Java 将是网络上的“世界语”, 今后所有用其他语言开发的软件统统都要用 Java 语言来改写。

Java 是新一代面向对象程序设计语言, 特别适合于 Internet 应用程序开发。它的平台无关性直接威胁到 Windows 技术的垄断地位。Java 作为软件开发的一种革命性的技术, 其地位已被确立。一些著名的计算机公司纷纷购买 Java 语言的使用权, 如 IBM, Netscape, Novell, Apple, DEC 和 SGI 等, 以及最不情愿的 Microsoft 公司。Java 已得到了工业界的认可, 众多的软件开发商开始支持 Java 的软件产品。Intranet 正在成为企业信息系统最佳的解决方案, 而其中 Java 将发挥不可替代的作用。世界上成千上万的软件开发人员都转向 Java。Java 语言正在不断发展和完善, Sun 公司是主要的发展推动者, 较通用的编译环境有 JDK (Java Development Kit) 与 JWS (Java Workshop)。还有很多其他公司正在开发 Java 语言的编译器与集成环境, 如 IBM 的 Visual age for Java, 以及 Borland 公司的 JBuilder 等。Java 语言的正确性与效率都在不断提高, 用户用 Java 编程和用 C++ 编程一样方便和高效。

Java 有着广泛的应用前景, 包括所有面向对象的应用开发, 计算过程的可视化的软件开发, 动态画面图形图像开发, 交互式操作的设计, Internet 的系统管理开发, Intranet 上的软件开发, 数据库开发, 以及其他类型的程序开发等。

Java 是一种简单易用, 完全面向对象, 具有平台无关性且安全可靠的主要面向 Internet 的开发工具。自 1995 年正式问世以来, Java 的快速发展已经让整个 Web 世界发生了翻天覆地的变化。随着 Java Servlet 的推出, Java 在电子商务方面开始崭露头角: 最新的 JavaServer



Pages (JSP) 技术的推出, 更让 Java 成为基于 Web 的应用程序的首选开发工具。要学习 Java 技术中的 JSP 和 Servlet, Java 基础是必不可少的, 须掌握 Java 的基础知识。

### 1.1.1 Java 技术的历史

Java 技术是由美国 Sun 公司倡导和推出的, Java 技术包括 Java 语言和 Java Media APIs, Security APIs, Management APIs, Java Applet, Java RMI, JavaBean, Java OS, Java Servlet, JDBC, JNDI 和 Enterprise JavaBean 等。1990 年, Sun 公司 James Gosling 领导的小组设计了一种平台独立的语言 Oak, 主要用于为各种家用电器编写程序。1995 年 1 月, Oak 被改名为 Java。1995 年 5 月 23 日, Sun 公司在 Sun World 95 上正式发布 Java 和 Hot Java 浏览器。1995 年 8 月至 12 月, Netscape 公司、Oracle 公司、Borland 公司、SGI 公司、Adobe 公司、IBM 公司、AT&T 公司、Intel 公司获得 Java 许可证。1996 年 1 月, Sun 公司宣布成立新的业务部门, 即 Java 软件部, 以开发、销售并支持基于 Java 技术的产品, 由 Alan Baratz 任总裁。同时, Sun 公司推出 Java 开发工具包 JDK 1.0, 为开发人员提供编制 Java 应用软件所需的工具。1996 年 2 月, Sun 公司发布 Java 芯片系列, 包括 picoJava, MicroJava 和 UltraJava, 并推出 Java 数据库连接 JDBC。1996 年 3 月, Sun 公司推出 Java WorkShop。1996 年 4 月, Microsoft 公司、SCO 公司、Apple (苹果电脑) 公司、NEC 公司等获得 Java 许可证。Sun 公司宣布苹果电脑、HP、日立、IBM、微软、Novell、SGI、SCO、Tandem 等公司将把 Java 平台嵌入其操作系统中。1996 年 5 月, HP 公司、Sybase 公司获得 Java 许可证, 北方电讯公司宣布了把 Java 技术和 Java 微处理器应用到其下一代电话机中的计划。5 月 29 日, Sun 公司在旧金山举行第一届 JavaOne 世界 Java 开发者大会, 业界人士踊跃参加。Sun 公司在大会上推出一系列 Java 平台新技术。1996 年 8 月, Java WorkShop 成为 Sun 公司通过因特网提供的第一个产品。1996 年 9 月, Addison-Wesley 和 Sun 公司推出 Java 虚拟机规范和 Java 类库。1996 年 10 月, 德州仪器等公司获得 Java 许可证。Sun 公司提前完成 JavaBean 规范并发布, 发布第一个 Java JIT (Just-In-Time) 编译器, 并打算在 Java WorkShop 和 Solaris 操作系统中加入 JIT。10 月 29 日, Sun 公司发布 Java 企业计算技术, 包括 Java Station 网络计算机, 65 家公司发布的 85 个 Java 产品和应用, 7 个新的 Java 培训课程和 Java 咨询服务, 基于 Java 的 Solstice 互联网邮件软件, 新的 Java 开发者支持服务, 演示 HotJava Views、Java Tutor, 完成 Java Card API 等。Sun 公司宣布完成 Java Card API 规范, 这是智能卡使用的第一个开放 API。Java Card 规范将把 Java 能力赋予全世界的亿万张智能卡。1996 年 11 月, IBM 公司获得 Java OS 和 HotJava 许可证; Novell 公司获得 Java WorkShop 许可证; Sun 和 IBM 公司宣布双方就提供 Java 化的商业解决方案达成一项广泛协议, IBM 公司同意建立第一个 Java 检验中心。1996 年 12 月, Xerox 等公司获得 Java 或 Java OS 许可证; Sun 公司发布 JDK 1.1、Java 商贸工具包、JavaBean 开发包及一系列 Java APIs, 并推出一个新的 Java Server 产品系列, 其中包括 Java Web Server, Java NC Server 和 Java Server Toolkit; Sun 公司发布 100% 纯 Java 计划, 得到百家公司的支持。1997 年 1 月, SAS 等公司获得 Java 许可证; Sun 公司交付完善的 JavaBean 开发包, 这是在确定其规范后不到 8 个月内完成的。1997 年 2 月, Sun 和 ARM 公司宣布同意使 Java OS 运行在 ARM 公司的 RISC 处理器架构上; Informix 公司宣布在其 Universal Server 和其他数据库产品上支持 JDK 1.1; Netscape 公司宣布其 Netscape Communicator 支持所有 Java 化的应用软件和核心 APIs。1997 年 3 月, HP 公司获得 Java WorkShop 许可证, 用于 HP-UX 操作系统。稍后, 西门子 AG 公司等相继获得 Java 许可证; 日立半导体公司、Informix 公司等获得 Java OS 许可



证；Novell 公司获得 Java Studio 许可证。继而，Sun 公司发售 Java OS 1.0 操作系统（这是一种在微处理器上运行 Java 环境的最小、最快的方法）提供给 Sun 公司的 Java OS 许可证持有者使用；发售 HotJava Browser 1.0，这是一种 Java 浏览环境，可以方便地按剪裁来编制专用的信息应用软件，如客户自助台和打上公司牌号的网络应用软件；推出 JDK 1.1.1。1999 年 6 月，Sun 公司发布 JDK 1.3 和 Java Web Server 2.0。

### 1.1.2 JDBC

JDBC 是用于执行 SQL 语句的 Java 应用程序接口，由一组用 Java 语言编写的类与接口组成。在 JSP 中将使用 JDBC 来访问数据库。JDBC 是一种规范，它让各数据库厂商为 Java 程序员提供标准的数据库访问类和接口，这样就使得独立于 DBMS 的 Java 应用程序的开发工具和产品成为可能。一般的 Java 开发工具都带有 JDBC-ODBC 桥驱动程序，只要能够使用 ODBC 访问的数据库系统，也就能使用 JDBC 访问了。有趣的是，不同于 ODBC 是 Open Database Connectivity 的简称，JDBC 并不是 Java Database Connectivity 的简称，而是 Sun 公司的注册商标，至少官方说法是这样的。

## 1.2 Java Servlet 及其特点

Java Servlet 是 JSP 技术的基础，而且大型的 Web 应用程序的开发需要 Java Servlet 和 JSP 配合才能完成。这里简单介绍 Servlet 的相关知识。

Java Servlet 这个名称大概源于 Applet，现在国内的翻译方式很多，可以称之为“服务小程序”。Servlet 其实与传统的 CGI 程序，以及 ISAPI, NSAPI 等 Web 程序开发工具的作用是相同的。在使用 Java Servlet 以后，用户不必再使用效率低下的 CGI 方式，也不必使用只能在某个固定 Web 服务器平台运行的 API 方式来动态生成 Web 页面。许多 Web 服务器都支持 Servlet，即使不直接支持 Servlet 的 Web 服务器，也可以通过附加的应用服务器和模块来支持 Servlet。得益于 Java 的跨平台的特性，Servlet 也是与平台无关的。实际上，只要符合 Java Servlet 规范，Servlet 是完全与平台和 Web 服务器无关的。由于 Java Servlet 内部以线程方式提供服务，不必对每个请求都启动一个进程，并且利用多线程机制可以同时为多个请求服务，因此 Java Servlet 的效率非常高。

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  $N$  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。

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。

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 程序则不能；Servlet 还能在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。



Servlet 用 Java 编写, Servlet API 具有完善的标准。因此,为 I-Planet Enterprise Server 写的 Servlet 无须任何实质上的改动,即可移植到 Apache、Microsoft IIS 等服务器上。几乎所有的主流服务器都直接或通过插件支持 Servlet。

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,更容易使用,功能更强大,具有更好的可移植性,更节省投资。

## 1.3 JSP 及其特点

### 1.3.1 JSP 技术概述

JSP 技术为创建显示动态生成内容的 Web 页面提供一个简捷而快速的方法。JSP 技术的设计目的是使得构造基于 Web 的应用程序更加容易和快捷,而这些应用程序能够与各种 Web 服务器、应用服务器、浏览器和开发工具共同工作。

在万维网短暂的历史中,它已经从一个大部分显示静态信息的网络演化到可以对股票进行交易和进行购书操作的一个基础设施。在各种各样的应用程序中,对于可能使用的基于 Web 的客户端,看上去没有任何限制。

基于浏览器客户端的应用程序相对传统的基于客户机/服务器的应用程序有几个好处。这些好处包括几乎没有限制的客户端访问和极其简化的应用程序部署、管理(要更新一个应用程序,管理人员只需更改一个基于服务器的程序,而不是成千上万的安装在客户端的应用程序)。这样,软件工业正迅速地向着基于浏览器客户端的多层次应用程序迈进。

这些快速增长的精巧的基于 Web 的应用程序要求开发技术上的改进。静态 HTML 对于显示相对静态的内容是不错的选择;新的挑战在于创建交互的基于 Web 的应用程序,在这些程序中,页面的内容是基于用户的请求或者系统的状态,而不是预先定义的文字。

对于这个问题的一个早期解决方案是使用 CGI-BIN 接口。开发人员编写与接口相关的单独的程序,以及基于 Web 的应用程序,后者通过 Web 服务器来调用前者。这个方案有着严重的扩展性问题——每个新的 CGI 要求在服务器上新增一个进程。如果多个用户并发地访问该程序,则这些进程将消耗该 Web 服务器所有的可用资源,并且系统性能会降低到极其低下的地步。

某些 Web 服务器供应商已经尝试通过为他们的服务器提供“插件”和 API 来简化 Web 应用程序的开发。这些解决方案是与特定的 Web 服务器相关的,不能解决跨多个供应商的解决方案的问题。例如,微软的 Active Server Pages™ (ASP) 技术使得在 Web 页面上创建动态内容更加容易,但它也只能工作在微软的 IIS 和 Personal Web Server 上。

还存在其他的解决方案,但是都不能使一个普通的页面设计者轻易地掌握。而像 Java Servlet 这样的技术就可以使得用 Java 语言编写交互的应用程序的服务器端的代码变得容易。一个 Java Servlet 就是一个基于 Java 技术的运行在服务器端的程序(与 Applet 不同,后者运行在浏览器端)。开发人员能够编写出这样的 Servlet,以接收来自 Web 浏览器的 HTTP 请求,动态地生成响应(可能要查询数据库来完成这项请求),然后发送包含 HTML 或 XML 文档的响应到浏览器。

采用这种方法,整个网页必须都在 Java Servlet 中制作。如果开发人员或者 Web 管理人员想要调整页面显示,则不得不编辑并重新编译该 Java Servlet,即使它在逻辑上已经能够运



行了。采用这种方法，生成带有动态内容的页面仍然需要应用程序的开发技巧。

很显然，目前所需要的是一个业界范围内的创建动态内容页面的解决方案。这个方案将解决当前方案所受到的限制，即能够在任何 Web 或应用程序服务器上运行，将应用程序逻辑和页面显示分离，能够快速地开发和测试，简化开发基于 Web 的交互式应用程序的过程。

JSP 技术就是被设计用来满足上面的要求的。JSP 规范是 Web 服务器、应用服务器、交易系统，以及开发工具供应商间广泛合作的结果。Sun 公司开发出这个规范来整合和平衡已经存在的对 Java 编程环境进行支持的技术和工具。其结果是产生一种新的开发基于 Web 应用程序的方法，给予使用基于组件应用逻辑的页面设计者以强大的功能支持。

在开发 JSP 规范的过程中，Sun 公司与许许多多主要的 Web 服务器、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，其结果是找到一种为应用和页面开发人员平衡了可移植性和易用性的开发方法。

按照脚本语言是服务于某个子系统的语言这种论述，JSP 应当被看做一种脚本语言。然而，作为一种脚本语言，JSP 又显得过于强大，在 JSP 中几乎可以使用全部的 Java 类。

作为一种基于文本的、以显示为中心的开发技术，JSP 具有 Java Servlet 的所有好处。而且，当与一个 JavaBean 类（JavaBean 是可复用的组件，对 JavaBean 并没有严格的规范，理论上讲，任何一个 Java 类都可以是一个 Bean）结合在一起时，JSP 提供一种使内容和显示逻辑分开的简单方式。分开内容和显示逻辑的好处是，更新页面外观的人员不必懂得 Java 代码，而更新 JavaBean 类的人员也不必是设计网页的行家。他们可以用带 JavaBean 类的 JSP 页面来定义 Web 模板，以建立一个由具有相似外观的页面组成的网站。JavaBean 类完成数据提供，这样在模板中就没有 Java 代码，这意味着这些模板可以由一个 HTML 编写人员来维护。当然，也可以利用 Java Servlet 来控制网站的逻辑，通过 Java Servlet 调用 JSP 文件的方式来将网站的逻辑和内容分离。

在选择使用一个 Java Servlet，还是一个 JSP 页面时，要记住 Java Servlet 是一个程序设计工具，它最适用于不需要频繁修改的低级应用功能；而 JSP 页面则通过以显示为中心的描述性方法，将动态内容和逻辑结合在一起。对于使用一个 JSP 页面的简单的基于 Web 的应用程序，可以使用定制标记或者 Scriptlet 直接嵌入 JSP 页面中的很小的 Java 代码段，而不使用 JavaBean 类来将内容与应用逻辑结合起来。定制标记被打包到一个标记库中，并被引入一个 JSP 页面中。

一般说来，在实际的 JSP 引擎中，JSP 页面在执行时是编译式，而不是解释式的。解释式的动态网页开发工具如 ASP, PHP3 等，由于速度等原因，已经满足不了当前大型电子商务应用的需要了，传统的开发技术都在向编译执行的方式改变，如 ASP→ASP+, PHP3→PHP4。尽管 JSP 的规范书中并没有要求实际的 JSP 引擎要使用编译式的执行方式，但一般是不会使用解释的方式来执行 JSP 页面的。通常来说，JSP 页面要先翻译为 Servlet 的 Java 源文件，再经过 Java 编译器编译为 Servlet 的 class 文件。为什么要编译为 Servlet 呢？这是为了让原先的 Servlet 引擎可以直接服务于 JSP，而 JSP 引擎仅仅需要将 JSP 编译为 Servlet 就可以了。这里要注意的是，JSP 规范书中并没有规定如何将 JSP 页面编译为 Servlet。因此，不同的 JSP 引擎编译的结果也是不一样的。在 JSP 文件编译为 Servlet 以后，每次客户机（通常是用户的 Web 浏览器）向服务器请求这一个 JSP 文件的时候，服务器将检查自上次编译后 JSP 文件是否有改变。如果没有改变，就直接执行 Servlet，而不用再重新编译，其效率是相当高的。一般说来，JSP 文件的编译是在第一个用户访问到这个 JSP 页面时发生的，而这第一个用户通



常是开发人员自己。这样，正式放在服务器上让用户访问的 JSP 文件一般都已经有了对应的编译好的 Servlet 了。许多服务器都有设置，可以使 JSP 文件在第一个用户访问之前就预先编译好，这样看来，效率就更高了。

在 JSP 规范书中，并没有明确要求 JSP 中的程序代码部分（称为 Scriptlet）一定要用 Java 来写。实际上，有一些 JSP 引擎就采用其他的脚本语言，如 WebL 等，但实际上这种脚本语言也是构建在 Java 上面，编译为 Servlet 来实现的。按照 JSP 规范书，使用完全和 Java 没有任何关系的 Scriptlet 也是可以的。由于 JSP 的强大功能主要在于能和 Java Beans, Enterprise JavaBean 一起工作，所以即使是 Scriptlet 部分不使用 Java，编译成的执行代码也应该是与 Java 相关的。

JSP 是一种实现普通静态 HTML 和动态 HTML 混合编码的技术，许多由 CGI 程序生成的页面大部分仍旧是静态 HTML。动态内容只在页面中有限的几个部分出现。包括 Servlet 在内的大多数 CGI 技术及其变种，总是通过程序生成整个页面。JSP 可以分别创建这两个部分。下面就是一个简单的 JSP 页面。

```
<%@ page contentType="text/html; charset=GBK" %>
<html>
<head>
<title>
Jsp2
</title>
</head>
<body>
<h1>
欢迎使用 JSP 编写网络程序
</h1>
<form method="post">
<br><br>
<% out.println("第一个简单的 JSP 程序"); %>
<hr>
<input type="submit" name="Submit" value="Submit">
<input type="reset" value="Reset">
</form>
</body>
</html>
```

这个页面运行的效果如图 1-1 所示。

### 1.3.2 JSP 的优势

和传统的 CGI 相比，JSP 具有相当的优势。首先，传统的 CGI 程序在速度上需要使用系统的标准输入、输出设备来实现动态网页的生成，而 JSP 是直接和服务器相关联的。而且对于 CGI 来说，每一个访问都需要新增加一个进程来处理；进程不断地建立和销毁，对于作为 Web 服务器的计算机将是不小的负担。其次，JSP 是专门为 Web 开发而设计的，其目的是为了建立基于 Web 的应用程序，包含一整套的规范和工具。使用 JSP 技术可以很方便地将一大



堆 JSP 页面组合成为一个 Web 应用程序。JSP 的开发速度快,开发难度小,在编译为 Java Servlet 以后,配合目前最新的 JIT (Just In Time) 的 Java 解释器,其执行速度也慢不了多少。



图 1-1 一个简单的 JSP 运行界面

Microsoft 公司的 ASP 是一种和 JSP 类似的技术。JSP 和 ASP 相比具有两方面的优点。首先,动态部分用 Java 编写,而不是 VB Script 或其他 Microsoft 语言,不仅功能更强大,而且更易于使用。其次,JSP 应用可以移植到其他操作系统和非 Microsoft 的 Web 服务器上。

JSP 并没有增加任何本质上不能用 Servlet 实现的功能。但是,在 JSP 中编写静态 HTML 更加方便,不必再用 `println` 语句来输出每一行 HTML 代码。更重要的是,借助内容和外观的分离,页面制作中不同性质的任务可以方便地分开,比如由页面设计专家进行 HTML 设计,同时留出供 Servlet 程序员插入动态内容的空间。

JSP 和服务器端包含 (Server-Side Include, SSI) 相比,SSI 是一种受到广泛支持的在静态 HTML 中引入外部代码的技术。JSP 在这方面的支持更为完善,因为它可以用 Servlet 而不是独立的程序来生成动态内容。另外,SSI 实际上只用于简单的包含,而不是面向那些能够处理表单数据、访问数据库的“真正的”程序。

JavaScript 能够在客户端动态地生成 HTML。虽然 JavaScript 很有用,但它只能处理以客户端环境为基础的动态信息。除了 Cookie 之外,HTTP 状态和表单提交数据对 JavaScript 来说都是不可用的。另外,由于是在客户端运行,JavaScript 不能访问服务器端资源,比如数据库、目录信息等。

### 1.3.3 用 JSP 开发 Web 的几种主要方式

JSP 作为 J2EE 的一部分,既可用于开发小型的 Web 站点,也可用于开发大型的、企业级的应用程序。本节将讲述对于不同规模的 Web 系统,使用 JSP 进行开发的不同方式。

#### 1. 直接使用 JSP

对于最小型的 Web 站点,可以直接使用 JSP 来构建动态网页。这种站点最为简单,所需要的仅仅是简单的留言板、动态日期等基本功能。对于这种开发模式,一般可以将所有的动态处理部分都放置在 JSP 的 Scriptlet 中,就像一般使用 PHP 或 ASP 开发动态网页一样。



## 2. JSP+JavaBean

中型站点面对的是数据库查询、用户管理和小量的商业业务逻辑。对于这种站点，不能将所有的东西全部交给 JSP 页面来处理。在单纯的 JSP 中加入 JavaBean 技术将有助于这种中型网站的开发。利用 JavaBean，将很容易完成如数据库连接、用户登录与注销、商业业务逻辑封装的任务。例如，将常用的数据库连接写为一个 JavaBean，既方便了使用，又可以使 JSP 文件简单而清晰；通过封装，还可以防止一般的开发人员直接获得数据库的控制权。

## 3. JSP+JavaBean+Servlet

无论用 ASP 还是 PHP 开发动态网站，长期以来都有一个比较重要的问题，就是网站的逻辑关系和网站的显示页面不容易分开。常常可以看见一些夹杂着 if...then...、case select 或 if{...}，以及大量显示用的 HTML 代码的 ASP、PHP 页面，即使其作者是有着良好的程序写作习惯的程序员，这些页面也几乎无法阅读。另一方面，动态 Web 的开发人员也在抱怨，将网站美工设计的静态页面和动态程序合并是一个异常痛苦的过程。

如何解决这个问题呢？在 JSP 问世以后，有人认为 Servlet 已经完全可以被 JSP 代替。然而，事实是 Servlet 在不再担负动态页面生成的任务以后，开始担负起决定整个网站逻辑流程的任务。在逻辑关系异常复杂的网站中，借助于 Servlet 和 JSP 良好的交互关系，以及 JavaBean 的协助，完全可以将网站的整个逻辑结构放在 Servlet 中，而将动态页面的输出放在 JSP 页面中。在这种开发方式中，一个网站可以有有一个或几个核心的 Servlet 来处理网站的逻辑，通过调用 JSP 页面来完成客户端（通常是 Web 浏览器）的请求。



### 本章小结

本章是全书的开篇，主要介绍一些最基本的概况信息，包括 Java 技术、Java 技术的历史、JavaBean、JDBC、J2EE、EJB、Java Servlet 及其特点、JSP 及其特点、JSP 技术概述、JSP 的优势，以及用 JSP 开发 Web 的几种主要方式。



### 习题

1. 简述 Java 技术。
2. 什么是 JDBC？
3. 简单介绍 Java Servlet 和 JSP 的特点。
4. 用 JSP 开发 Web 程序有哪些方式？



## 第2章 安装相关软件



知识要点:

- 常用开发工具介绍;
- 安装 Servlet 和 JSP 开发工具 JBuilder 9;
- 安装支持 Servlet 和 JSP 的 Web 服务器 Tomcat。

### 2.1 常用开发工具的选择

现在流行的 JSP/Servlet 开发工具主要有 IBM VisualAge, JBuilder, NetBean, JRun, Urledit, Dreamweaver 和 EditPlus 等。这里主要介绍 IBM VisualAge, JBuilder 和 JRun 三种工具的一些基本情况。

#### 2.1.1 IBM VisualAge for Java

IBM 的 Web 开发产品可谓同类之中的庞然大物, 体积庞大, 功能众多, 使用对象是整个开发团队, 而设计目标则是处理视野中能看到的一切。

IBM WebSphere 软件平台 (全套的集成电子商务解决方案, 包括 VisualAge for Java 1.2 版) 是电子商务应用构架的关键部分, 该构架是 IBM 在同一编程模型上开发的统一的端对端体系结构。作为公共编程模型, EJB (企业 Java Bean) 规范得到了行业范围内的广泛支持, 它也是电子商务应用构架的集成部件模型。VisualAge for Java 是提供向导功能的第一个 Java 集成编译环境, 它可以生成大量的 EJB 基础结构代码, 从而简化 EJB 开发过程。另外, 对话和实体 bean 的创建向导, 把 bean 映射到现有关系数据存储, 以及先进的测试环境, 极大地减少了创建、测试和部署 EJB 所花的时间。

VisualAge for Java 持续性工具能让开发人员把 EJB 映射到现有的关系数据库结构。利用对对话和实体 bean (CMP) 的支持, 开发人员能从现有的数据库表生成 EJB, 或从现有的 EJB 生成数据库结构。VisualAge for Java 还能提供把对象模型映射到数据模型的手段, 从而把数据库管理员和 Java 开发人员联系在一起。另外, VisualAge for Java 3.5 版还支持 IBM DB2 S/390 和 Sybase, 以及支持 IBM DB2 通用数据库和 Oracle。

如果在 VisualAge for Java 中主机运行 IBM WebSphere 测试环境 (WTE), 开发人员还能在把 EJB 和 Servlet 部署到远程服务器前对它们进行预测试。由于开发人员能启动和停止服务器及应用程序, 并能在 VisualAge for Java IDE 中任意改变应用, 所以开发人员不必远程部署和测试代码, 从而节省了大量时间。

生成测试客户机能让开发人员测试他们服务器端的 EJB, 而无须编写一个用户界面或等待图形制作人员给出的强制性界面。当用户界面完成时, 它还能作为一个测试工具, 决定因