



案例素材和源代码、电子课件、习题参考答案

JSP

应用与开发技术

(第3版)

马建红 李学相 主 编
韩 颖 王瑞娟 张 晗 副主编



非外借



清华大学出版社

JSP 应用与开发技术

(第3版)

马建红 李学相 主 编

韩 颖 王瑞娟 张 晗 副主编

清华大学出版社

北 京

内 容 简 介

JSP(Java Server Pages)是由 Sun Microsystems 公司倡导的、许多公司参与建立的一种动态网页技术标准。JSP 被赋予了 Java 技术的强大功能,能够为用户提供强大的技术支持;同时,JSP 继承了 Java 的优势,可以建立安全的、跨平台的动态网站。

本书详细讲解了 JSP 的基本语法和 Web 程序设计方法。全书共 16 章,可分为 7 个部分。第 1 章和第 2 章是第 1 部分,介绍了 JSP 与 Web 技术概论和开发环境的搭建;第 3~5 章是第 2 部分,详细讲解了 JSP 技术的基本语法;第 6 章和第 7 章是第 3 部分,是 JSP 应用开发的进阶,讲述了 JavaBean、表单处理及文件的操作;第 8 章和第 9 章是第 4 部分,以 MySQL 数据库为例详细讲解了 JSP 中使用数据库的操作;第 10 章和第 11 章是第 5 部分,详细讲述了 Servlet 技术;第 12 章和第 13 章是第 6 部分,介绍了表达式语言(EL)、标准标签库(JSTL)及自定义标签库;第 14~16 章是第 7 部分,以 3 个实例讲述了 Web 开发的实际应用。另外,本书还给出了 6 个实验,以指导读者进行上机操作。

本书语言简洁,内容丰富,既可作为 JSP 初学者的入门教材,也可作为高等院校相关专业的教材和辅导用书,而且对 JSP 开发人员的自学也具有较高的参考价值。

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 13701121933

图书在版编目(CIP)数据

JSP 应用与开发技术 / 马建红, 李学相 主编. —3 版. —北京: 清华大学出版社, 2019
ISBN 978-7-302-51373-5

I. ①J… II. ①马… ②李… III. ①JAVA 语言—程序设计 IV. ①TP312.8

中国版本图书馆 CIP 数据核字(2018)第 232137 号

责任编辑: 王 定
封面设计: 孔祥峰
版式设计: 思创景点
责任校对: 成凤进
责任印制: 丛怀宇

出版发行: 清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址: 北京清华大学学研大厦 A 座

邮 编: 100084

社总机: 010-62770175

邮 购: 010-62786544

投稿与读者服务: 010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈: 010-62772015, zhiliang@tup.tsinghua.edu.cn

印 装 者: 三河市金元印装有限公司

经 销: 全国新华书店

开 本: 185mm×260mm

印 张: 32.5

字 数: 811 千字

版 次: 2011 年 2 月第 1 版

2019 年 1 月第 3 版

印 次: 2019 年 1 月第 1 次印刷

印 数: 26501~30000

定 价: 78.00 元

产品编号: 079635-01

前言

P R E F A C E

JSP(Java Server Pages)是目前动态网站开发技术中最典型的一种,它继承了 Java 语言的优势,是一种与平台无关的开发技术,而 Java 技术也赋予了 JSP 为用户提供强大功能的技术支持。JSP 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,提高了执行效率而成为互联网上的主流开发技术,已经越来越受到编程者的关注和喜爱。

JSP 虽然综合性地包括了 Java 和 HTML 这两类语法,但不能通过简单地使用 JSP,让它集显示、业务逻辑和流程控制于一身,因为用这种方式开发出来的 Web 应用程序难以维护。所以对 JSP 使用观念的建立,以及 JavaBean、数据库、Servlet 等技术的了解运用,是利用 JSP 开发复杂的商业级网站的重点。为了让读者在学习的过程中能够彻底掌握相关概念,除了基本语法介绍外,本书同时将重点集中在面向对象的观点和 JSP 程序架构方面。

《JSP 应用与开发技术(第 3 版)》仍保持了第 2 版实用、新颖和经验总结的特点,从基本的语法和规范入手,以经验为后盾,以实用为目标,以实例为导向,以实践为指导,深入浅出地讲解了 JSP 开发中的种种问题。以精简的内容介绍了 JSP 的语法和 Servlet、JDBC、EL、标签库、MVC 等技术。本书每一章节的实例读者可以直接使用,实例讲解过程条理清晰、循序渐进,符合程序设计的自然思路,读者学完一个章节,也就相应地掌握了相关的 JSP 开发思想和技术。并且通过几个较完整的综合实例,让读者对 JSP+Servlet+JavaBean+JDBC 这样的 Web 架构有一个整体认识。和第 2 版相比,本书不仅采用了最新版本的开发工具,而且增加了最新 Servlet 版本中出现的新特性,调整了相关实例,竭尽所能展现本书的实用性。另外,对第 2 版的章节进行了细致的修订,对部分内容进行了重大改动。

本书共 16 章,可以分成 7 个部分。

第 1 部分介绍了 JSP 编程基础与环境配置,包括第 1 章和第 2 章:第 1 章, JSP 与 Web 技术概论;第 2 章, JSP 的开发和运行环境。和第 2 版相比,增加了最新版的 Eclipse Java EE、MyEclipse 集成开发环境,以及其他 JSP 开发环境。新版本删除了第 2 版的第 2 章 HTML 和 JavaScript 技术的介绍。

第 2 部分介绍了 JSP 应用开发基础,包括第 3~5 章:第 3 章, JSP 基本语法;第 4 章, JSP 内置对象;第 5 章, Cookie 及会话追踪。和第 2 版相比,增加了 session 会话与会话追踪相关内容。

第 3 部分介绍了 JSP 应用开发进阶,包括第 6 章和第 7 章:第 6 章, JavaBean 和表单处理;第 7 章, JSP 中的文件操作。这部分主要介绍了 JavaBean 分离表示与实现,使用 JSP 处理 HTML 表单,使用 Java 类库里的 I/O 类,开发具备文件存取功能的网页程序。和第 2 版的不同之处是,增加了 Apache Commons FileUpload 文件的上传。

第 4 部分介绍了 JSP 数据库编程基础,包括第 8 章和第 9 章:第 8 章,应用 JDBC 进行数据库开发,以 MySQL 数据库为例详细讲解了 JSP 中使用数据库的操作,包括 JDBC、连接池、

分页处理等技术;第 9 章,JSP 与 JavaBean 应用实例。和第 2 版相比,本书删除了 MySQL 数据库安装配置等基本操作,并且更换了第 2 版第 11 章中的电商管理实例,改为一个简洁的商品管理系统,对 Model1 模式的应用进行详细的讲解。

第 5 部分讲述了 Servlet 技术,包括第 10 章和第 11 章:第 10 章,Servlet 基础;第 11 章,使用 Servlet 过滤器和监听器。新版本中增加了 Servlet 3.0 程序的步骤,监听器新增@WebListener 注解。

第 6 部分讲述了标签库的应用,包括第 12 和第 13 章:第 12 章,JSTL 标准标签库;第 13 章,自定义标签库。新版本中对每部分内容都进行了细致的修订。

第 7 部分即第 14~16 章,讲解了 Web 开发的具体应用。和第 2 版不同的是,新版中更新了部分实例,运用 MVC 技术,采用三层架构,并采用了自定义标签和 EL 等技术展示了三个门户网站综合实例,使读者对这些技术的使用有更深刻的了解。

本书由浅入深、循序渐进,全面系统地介绍了 JSP 程序设计的原理、方法和技术,还提供了大量的 JSP 应用开发实例,给出了相应的实用技巧、操作步骤及优化思路,可以帮助读者很快地进行实际开发。在本书每章的最后还提供了习题,让读者能够检验自己对各章内容的学习、消化程度,并巩固所学到的知识。

本书由马建红、李学相任主编,韩颖、王瑞娟、张晗任副主编。参与编写的人员还有卫权岗、郭红艳、谷保平、何保锋、程凤娟、尹辉、赵玉娟等。此外,代瑾、张文亮、王若楠、任东冉、杨卢阳、张天壮、石刘帅、申清源在整理材料方面给予了编者很大的帮助,在此,对他们表示衷心的感谢。

由于时间仓促,加之水平有限,书中不足之处在所难免,敬请读者批评指正。

本书每章的案例源代码可通过扫描对应章节前的二维码下载,课件、实验参考答案及课后习题参考答案下载如下:



课件



实验参考答案



课后习题参考答案

编者

2018 年 9 月

目录

C O N T E N T S

第 1 章 JSP 与 Web 技术概论	1
1.1 Web 应用程序开发基础知识	2
1.1.1 HTTP 通信协议概述	2
1.1.2 传统的 Web 应用程序	3
1.1.3 Web 结构	3
1.2 静态网页与 HTML	5
1.3 交互式动态网页技术	6
1.3.1 动态网页的程序语言	6
1.3.2 在客户端执行的网页语言	6
1.3.3 在服务器端执行的网页语言	7
1.4 网站数据库技术	8
1.5 JSP 与相关技术	9
1.5.1 什么是 JSP	10
1.5.2 JSP 技术原理	11
1.5.3 JSP 本质与 Servlet	12
1.5.4 JSP 与 JavaBean	12
1.6 软件编程体系简介	13
1.6.1 C/S 结构编程体系	14
1.6.2 B/S 结构编程体系	14
1.7 企业应用开发架构	15
1.7.1 两层架构	15
1.7.2 三层架构	15
1.7.3 N 层架构	15
1.7.4 开发架构比较	15
1.7.5 J2EE 简介	16
1.8 JSP 知识体系及学习之路	18
1.8.1 JSP 知识体系	18
1.8.2 JSP 程序员学习路径	18
1.9 小结	20

1.10 习题	20
第 2 章 JSP 的开发和运行环境	22
2.1 JSP 的开发和应用平台介绍	23
2.1.1 Caucho 公司的 Resin 平台	23
2.1.2 Apache 公司的 Tomcat 平台	23
2.1.3 BEA 公司的 WebLogic 平台	24
2.1.4 IBM WebSphere Application Server 平台	24
2.2 Eclipse Java EE 集成开发环境	25
2.2.1 安装和配置 JDK	25
2.2.2 Tomcat 服务器	27
2.2.3 Eclipse Java EE 开发环境搭建	32
2.3 MyEclipse 开发环境	38
2.3.1 MyEclipse 简介与下载	38
2.3.2 MyEclipse 安装与使用	39
2.4 其他 JSP 开发环境	45
2.4.1 IDEA 开发环境	45
2.4.2 Adobe Dreamweaver	46
2.4.3 UltraEdit 编辑器	47
2.5 小结	47
2.6 习题	48
第 3 章 JSP 基本语法	50
3.1 JSP 文件的结构	51
3.1.1 创建第一个 JSP 文件	51
3.1.2 分析 JSP 文件的组成元素	52
3.2 JSP 的脚本元素	53
3.2.1 隐藏注释	53

3.2.2	HTML 注释	53	4.5.5	其他 session 对象的常用方法	110
3.2.3	声明	55	4.5.6	session 对象应用实例	110
3.2.4	脚本代码	57	4.6	application 内置对象	116
3.2.5	表达式	58	4.6.1	存取 application 中的数据	116
3.3	JSP 指令元素	59	4.6.2	使用 application 对象取得信息	117
3.3.1	page 指令	59	4.6.3	application 对象应用实例	118
3.3.2	include 指令	64	4.7	其他 JSP 内置对象	122
3.3.3	taglib 指令	66	4.7.1	pageContext 对象	122
3.4	JSP 动作元素	66	4.7.2	config 对象	124
3.4.1	<jsp:include>	67	4.7.3	page 对象	127
3.4.2	<jsp:forward>	70	4.7.4	exception 对象	127
3.4.3	<jsp:param>	72	4.8	小结	128
3.4.4	<jsp:useBean>、<jsp:setProperty>和<jsp:getProperty>动作	73	4.9	习题	129
3.4.5	<jsp:plugin>	81	第 5 章	Cookie 及会话追踪	132
3.5	小结	83	5.1	Cookie 的概念和特性	133
3.6	习题	83	5.1.1	什么是 Cookie	133
第 4 章	JSP 内置对象	86	5.1.2	Cookie 的常见用途	134
4.1	JSP 内置对象概述	87	5.2	在 JSP 中使用 Cookie	135
4.2	request 对象	88	5.2.1	创建 Cookie	135
4.2.1	request 对象常用方法	88	5.2.2	读写 Cookie	136
4.2.2	request 对象应用实例	89	5.2.3	Cookie 中的主要方法	137
4.3	response 对象	95	5.2.4	操作 Cookie 的常用方法	137
4.3.1	response 对象常用方法	96	5.3	Cookie 对象的应用实例	139
4.3.2	response 对象应用实例	96	5.4	会话与会话追踪	146
4.4	out 对象	103	5.5	实现会话追踪的 4 种方式	147
4.4.1	out 对象方法成员与数据输出	104	5.5.1	Cookies 和 URL 重写	147
4.4.2	缓冲区操作	105	5.5.2	隐藏表单字段	148
4.4.3	out 对象应用实例	105	5.5.3	使用内建 session 对象	148
4.5	session 对象	107	5.6	内建 session 对象的生命周期	150
4.5.1	session 的概念	108	5.6.1	在 web.xml 中配置内建 session 对象的过期时间	150
4.5.2	session 对象的 ID	108	5.6.2	设置内建 session 对象的过期时间	150
4.5.3	session 的有效期限	109			
4.5.4	访问 session 中的数据	109			

5.7	利用 session 存取功能实现简单购物车	152
5.8	小结	158
5.9	习题	159
第 6 章	JavaBean 和表单处理	162
6.1	非 MVC 模式(Model1)	163
6.1.1	Model1 的特点	163
6.1.2	Model1 的应用范围	164
6.2	MVC 编程模式(Model2)	164
6.2.1	什么是 MVC 模式	164
6.2.2	MVC 模式在 Web 编程中的应用	165
6.3	剖析 JavaBean	165
6.3.1	什么是 JavaBean	166
6.3.2	JavaBean 的特征	167
6.3.3	创建一个 JavaBean	168
6.4	在 JSP 中使用 JavaBean	169
6.4.1	调用 JavaBean	170
6.4.2	访问 JavaBean 属性	170
6.4.3	设置 JavaBean 属性	170
6.4.4	JavaBean 的生命周期	170
6.4.5	类型自动转换规则	174
6.5	使用 JavaBean 处理表单数据	174
6.5.1	JSP 处理与 form 相关的常用标签简单实例	175
6.5.2	设置中文编码	181
6.6	小结	181
6.7	习题	181
第 7 章	JSP 中的文件操作	184
7.1	数据流和 File 类	185
7.1.1	数据流	185
7.1.2	File 类	185
7.2	读写文本文件	187
7.3	文件的浏览	189

7.4	创建和删除目录	190
7.5	文件的上传和下载	191
7.6	使用 jspSmartUpload 上传包	195
7.7	使用 Commons FileUpload 上传包	197
7.8	小结	199
7.9	习题	199
第 8 章	应用 JDBC 进行数据库开发	202
8.1	JDBC 概述	203
8.1.1	JDBC 的用途	203
8.1.2	JDBC 的典型用法	203
8.1.3	JDBC 的体系结构	204
8.1.4	驱动器类型	204
8.1.5	安装驱动器	206
8.2	JDBC 连接数据库的方法	206
8.3	使用 JDBC 操作数据库	207
8.3.1	使用 JDBC 访问数据库的过程	208
8.3.2	使用 Statement 执行 SQL 语句	210
8.3.3	PreparedStatement 接口	217
8.3.4	CallableStatement 对象	220
8.3.5	使用 ResultSet 处理结果集	225
8.4	Java 与 SQL 的数据类型转换	229
8.5	使用 JDBC 连接不同的数据库	231
8.5.1	连接 Oracle 数据库	231
8.5.2	连接 DB2 数据库	231
8.5.3	连接 SQL Server 数据库	231
8.5.4	连接 Sybase 数据库	232
8.5.5	连接 Access 数据库	232
8.6	连接池	232
8.6.1	连接池的实现原理	232
8.6.2	在 Tomcat 上配置数据源与连接池	233

8.6.3	配置连接池时需要注意的问题	235	10.3.1	GenericServlet	293
8.7	存取二进制文件	236	10.3.2	HttpServlet	294
8.7.1	图像文件存取到数据库的过程	236	10.3.3	Servlet 实现相关实例	295
8.7.2	声音文件存取到数据库的过程	240	10.4	Servlet 请求和响应相关	298
8.7.3	视频文件存取到数据库的过程	244	10.4.1	HttpServletRequest 接口	298
8.7.4	保存图片文件路径到数据库的存取图片过程	248	10.4.2	HttpServletResponse 接口	300
8.8	实现分页显示	251	10.4.3	Servlet 请求和响应相关实例	301
8.8.1	分页显示技术的优劣比较	251	10.5	Servlet 配置相关	303
8.8.2	分页显示的 JavaBean 实现	252	10.5.1	ServletConfig 接口	303
8.9	小结	258	10.5.2	获取 Servlet 配置信息的例子	303
8.10	习题	258	10.6	Servlet 中的会话追踪	307
第 9 章	JSP 与 JavaBean 应用实例	261	10.6.1	HttpSession 接口	307
9.1	需求和设计	262	10.6.2	HttpSession 应用实例	309
9.1.1	功能介绍	262	10.7	Servlet 上下文	311
9.1.2	文件结构	262	10.7.1	ServletContext 接口	311
9.1.3	数据库设计	263	10.7.2	ServletContext 接口的应用实例	312
9.2	使用 JavaBean 封装数据库的访问	263	10.8	Servlet 协作	313
9.3	项目页面实现	265	10.8.1	RequestDispatcher	313
9.4	小结	282	10.8.2	forward()控制页面跳转	314
第 10 章	Servlet 基础	283	10.8.3	include()控制页面包含	315
10.1	Servlet 介绍	284	10.9	Servlet 异常相关	316
10.1.1	什么是 Servlet	284	10.9.1	声明式异常处理	316
10.1.2	Servlet 技术特点	285	10.9.2	程序式异常处理	319
10.1.3	JSP 与 Servlet 的关系	285	10.10	Servlet 3.0 注解	322
10.1.4	Servlet 的工作原理	286	10.10.1	开发 Servlet 3.0 程序所需的环境	322
10.1.5	Servlet 常用接口和类	286	10.10.2	开发 Servlet 3.0 程序	322
10.2	开发部署一个简单的 Servlet	288	10.11	Servlet 应用实例	325
10.2.1	创建 Servlet 文件	289	10.12	小结	333
10.2.2	Servlet 的配置文件	291	10.13	习题	333
10.3	Servlet 实现相关的接口和类	292	第 11 章	使用 Servlet 过滤器和监听器	337
			11.1	过滤器在 Web 开发中的应用	338

11.1.1 过滤器概述	338	12.3 设置 JSTL 运行环境	375
11.1.2 Filter API	338	12.3.1 JSTL 的安装	375
11.1.3 Filter 接口	339	12.3.2 JSTL 应用示例	375
11.1.4 FilterConfig 接口	340	12.4 使用核心标签	376
11.1.5 FilterChain 接口	340	12.4.1 表达式操作	377
11.1.6 编写过滤器类	340	12.4.2 建立 URL	380
11.1.7 过滤器的部署	342	12.4.3 条件控制	383
11.1.8 对请求数据进行处理 过滤器	345	12.4.4 迭代—运行循环	385
11.1.9 过滤器新增@WebFilter 注解	350	12.5 使用 JSTL 的数据库标签	387
11.2 Servlet 监听器	351	12.5.1 指定数据源	388
11.2.1 监听器接口	351	12.5.2 进行查询或更新操作	388
11.2.2 ServletRequestListener 接口	352	12.5.3 对返回的结果进行处理	391
11.2.3 ServletRequestAttributeListener 接口	353	12.5.4 其他 SQL 标签库的标签	392
11.2.4 ServletContextListener 接口	353	12.6 i18n 与国际化	393
11.2.5 ServletContextAttributeListener 接口	355	12.6.1 国际化设置标签	393
11.2.6 HttpSessionAttributeListener 接口	357	12.6.2 消息标记库	394
11.2.7 HttpSessionBindingListener 接口	359	12.6.3 数字、日期格式化	396
11.2.8 监听器新增@WebListener 注解	364	12.7 函数标签	399
11.3 小结	365	12.8 小结	399
11.4 习题	365	12.9 习题	400
第 12 章 JSTL 标准标签库	368	第 13 章 自定义标签库	403
12.1 EL 表达式语言	369	13.1 自定义标签体系介绍	404
12.1.1 EL 与 EL 隐含对象	369	13.1.1 标签的形式	404
12.1.2 在 EL 中访问 JSP 隐含对象 的 getXXX() 方法	373	13.1.2 标签类相关接口和类	405
12.1.3 用 EL 访问 JavaBean 中的 属性	373	13.1.3 标签库描述文件	409
12.2 JSTL 标签库简介	374	13.1.4 在 Web 部署描述符中引入 标签库文件	411
		13.1.5 在页面中使用标签	411
		13.1.6 标签在 Web 页面中的 作用	412
		13.2 传统标签的开发	412
		13.2.1 带属性标签的开发	412
		13.2.2 带 Body 标签的开发	415
		13.2.3 嵌套标签的开发	417
		13.2.4 迭代标签的开发	420
		13.3 Simple 标签的开发	423

13.3.1	SimpleTag 接口	423	15.5.4	用户管理	448
13.3.2	Simple 标签的开发示例	424	15.5.5	调查问卷管理	454
13.4	小结	425	15.5.6	问题管理	460
13.5	习题	425	15.6	小结	467
第 14 章	网上书店	427	第 16 章	Web 应用开发实践	468
14.1	快乐购书网介绍	428	16.1	系统介绍	469
14.2	系统需求 and 设计	428	16.2	系统需求分析	469
14.3	数据库表设计	428	16.3	系统功能结构	469
14.4	JSP 页面开发	429	16.4	系统功能描述	470
14.5	JavaBean 开发	431	16.4.1	游客用户浏览模块	470
14.5.1	使用 Java Bean 封装数据库的 访问	431	16.4.2	管理员登录模块	473
14.5.2	购物车 JavaBean	432	16.4.3	管理员管理模块	473
14.6	Servlet 开发	432	16.5	数据库设计	477
14.6.1	普通 Servlet 开发	433	16.5.1	数据库逻辑结构设计	477
14.6.2	Listener 开发	434	16.5.2	数据库表的设计	478
14.6.3	Filter 开发	435	16.5.3	数据库相关脚本	479
14.7	其他 bean 类——Util	435	16.6	系统实现	480
14.8	部署描述符	436	16.6.1	模块公用类	480
14.9	小结	438	16.6.2	JavaBean	481
14.10	习题	439	16.6.3	Servlet	484
第 15 章	调查问卷管理系统	440	16.6.4	自定义标签	485
15.1	系统介绍	441	16.6.5	前台界面的实现	490
15.2	系统需求分析	441	16.6.6	后台管理页面的实现	493
15.3	系统功能结构	442	16.7	小结	494
15.4	数据库设计	442	16.8	习题	494
15.4.1	数据库逻辑结构设计	442	附录	实验	496
15.4.2	数据库表的设计	443	实验一	JSP 应用开发基础(一)	496
15.4.3	数据库相关脚本	444	实验二	JSP 应用开发基础(二)	498
15.5	系统实现	445	实验三	JSP 应用开发进阶	502
15.5.1	JavaBean	445	实验四	JSP 数据库编程基础	504
15.5.2	Servlet	445	实验五	Servlet 技术实验	506
15.5.3	系统界面	447	实验六	Web 应用开发	508

第 1 章

JSP与Web技术概论

本章主要对 JSP 与 Web 技术进行概要介绍，并为读者提出了一些学习 JSP 的建议。为了让读者在开始学习之前能对 JSP 技术有一个清晰与完整的概念，本章首先介绍了 Web 应用程序开发基础知识，还介绍了静态网页和动态网页技术，通过 JSP 技术原理以及与其他主流动态网页技术的比较，进一步了解 JSP 技术是一种功能强大、可以实现跨平台操作的动态网页开发技术；然后通过编写一个简单的 JSP 页面实例，让读者对 JSP 技术有一个直观的感性认识；最后介绍了软件编程体系和企业应用开发架构。本章对 JSP 知识体系的剖析，有助于读者学习和掌握 JSP 知识体系中的各个模块，对 JSP 技术有一个总体了解。

本章学习目标

- ◎ 掌握 Web 应用程序开发基础知识
- ◎ 了解静态网页与动态网页技术
- ◎ 了解网站数据库技术
- ◎ 掌握 JSP 的基本概念
- ◎ 掌握 JSP 的知识体系
- ◎ 了解软件编程体系
- ◎ 了解企业应用开发架构
- ◎ 了解 JSP 的学习之路

1.1 Web 应用程序开发基础知识

WWW(World Wide Web)即全球广域网，也称为万维网。它是一种基于超文本和 HTTP 的、全球性的、动态交互的、跨平台的分布式图形信息系统，是建立在 Internet 上的一种网络服务，为浏览者在 Internet 上查找和浏览信息提供了图形化的、易于访问的直观界面，其中的文档及超级链接将 Internet 上的信息节点组织成一个互为关联的网状结构。万维网并不等同互联网，万维网只是互联网提供的服务之一。

从表现形式上来看，基于 Web 的应用程序是许多网页的集合，这些网页可以与其他网页进行交互，也可以与 Web 服务器上的其他各种资源(如数据库或 Web 服务)进行交互。

为了规范这种 Web 页面之间的交互流程，保证客户端和服务端之间能很好地通信，在网络上必须采用一个规范性的协议(HTTP 通信协议)来定义各种 Web 应用上的服务细节。

1.1.1 HTTP 通信协议概述

HTTP(Hypertext Transfer Protocol)通信协议是目前在 Internet 上应用最广泛的通信协议之一。HTTP 通信协议允许客户端向服务器提出基于 HTTP 格式的“请求”(Request)，而服务器解析请求和完成请求的处理后，将根据实际的处理结果向请求端传回基于 HTTP 的“回应”(Response)。根据 HTTP 通信协议，客户端和服务端端的交互主要由以下四个步骤组成。

- (1) 当客户端向 Web 服务器发出请求时，Web 服务器将会为该客户开启一个新的连接。
- (2) 通过这个连接，用户(服务请求端)可以将 HTTP 请求通过网络，传送给 Web 服务器。
- (3) 当 Web 服务器收到 HTTP 请求时，将根据请求内容进行相应的处理，并将处理结果包装成 HTTP 回应。
- (4) 服务器会将 HTTP 回应传送给用户。只要用户接收到 HTTP 回应，Web 服务器就会关闭同客户端的连接，从而结束本次通信。

图 1-1 说明了客户端与 Web 服务器通信的整个流程，从图中可以看到，HTTP 通信协议用来规范在客户端和服务端之间的数据传输格式，更重要的是，HTTP 通信协议定义了从客户端发送 HTTP 请求到服务器端返回 HTTP 响应的整个流程。

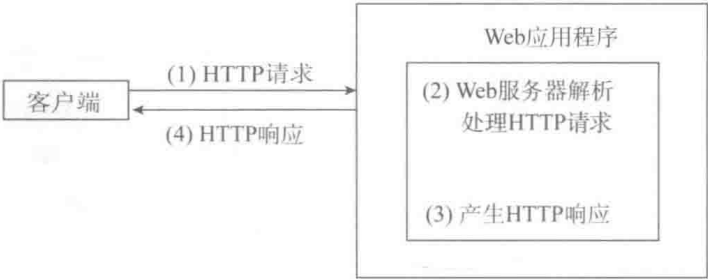


图 1-1 客户端(客户)与服务器端(Web 服务器)之间的交互关系

1.1.2 传统的 Web 应用程序

目前,大多数的 Web 应用程序都是基于 HTTP 通信协议的,即一个典型的基于 Web 的服务器和客户端应用程序必须具备“接受客户端的请求”和“将处理结果回应给客户端”这两种能力。

在 Web 开发的起步阶段,程序员是用 CGI(Common Gateway Interface, 公共网关接口)程序来编写 Web 应用程序组件的,即 Web 服务器之间及客户端和服务端之间是通过 CGI 来互相交换信息。当客户端向 Web 服务器送出一个 HTTP 请求时,Web 服务器将根据其中的 CGI 内嵌服务程序,执行如下动作:

- 如果客户端请求的内容是静态的 HTML 网页数据(即这些数据不是从数据库等数据源动态获取生成),Web 服务器将会根据事先编写好的处理程序自行处理,产生回应信息并返回。
- 如果涉及动态数据(如查询在服务器上的数据),则由内嵌的 CGI 程序负责处理,处理结束后,CGI 程序会先把结果回送给 Web 服务器,再由 Web 服务器传回用户端。

整个流程如图 1-2 所示。

从实际的运行效果来看,虽然 CGI 程序能在一定程度上解决客户端与服务器端之间的交互问题,但是基于 CGI 程序本身的特点,这样的开发模式隐含了一些可能导致系统效率降低和项目可维护性变差的缺点。

- 从运行方式上来看,CGI 程序属于“操作系统进程”,因此对于每个 HTTP 的请求,服务器端都必须对应地开启一个 CGI 的服务,这加重了系统整体资源负担。
- CGI 程序通常不具有“平台独立性”,它甚至可以用多种语言来编写,如果转换到其他系统平台,程序可能要做适当的改写或重写。
- 最为重要的是,CGI 无法很好地形成一种通用的规范。事实上,不同类型的服务器 CGI 代码往往从请求风格到服务方式,都是不同的。

在实际的应用中,以上三个特点确实影响到 Web 应用程序的执行性能,因此需要一个能够取代 CGI 的 Web 开发技术,来满足“低负载、高重用和规范统一”等通用要求。

JSP 技术恰好可以完美地取代 CGI 程序,Java 的“平台无关性”可以保证用 JSP 编写出来的代码能够在多个平台之间很好地迁移。同时,JSP 语言是基于 Java 虚拟机的,不会对系统产生严重的资源负载。而且,不同平台上的 Java 语言及其虚拟机均采用了同一类标准(基于 Java 的标准),因此,JSP 可以满足“规范统一”这个要求。

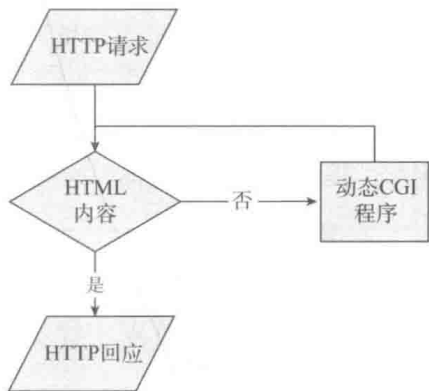


图 1-2 CGI 程序基本运行流程

1.1.3 Web 结构

Web 主要由两个部分组成,其结构如图 1-3 所示:提供 Web 信息服务的服务器端网站及向网站提出信息内容浏览要求的客户端浏览器。服务器端网站存放包含各种形态的多媒体信息网页,客户端主要包含各种可以浏览网页内容的浏览器软件,目前

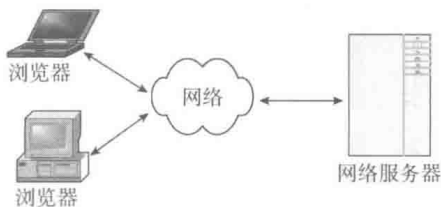


图 1-3 Web 结构

通用的浏览软件为 Windows 操作系统的 Internet Explorer(IE)浏览器。

了解 Web 结构的初步概念后，下面继续针对其中几个重要的技术内容进行讲解。

1. 客户端与浏览器

要打开和浏览网络上的网页文件，必须通过浏览器程序，如 IE、Netscape 等。而使用浏览器打开网页的这一端，称为客户端。因为网站为上线用户提供打开网页的服务，因此用户也可称为客户。

浏览器的功能主要是解释 HTML 文件中的内容，若 HTML 中同时含有客户端执行的描述语言，例如 VBScript 或 JavaScript，则浏览器同样会对其进行解释的操作，最后将整份网页的执行结果呈现在用户的浏览器窗口中。

2. 服务器端与服务器

与客户端相比，提供浏览网页服务的一方称作“服务器端”，而用来放置这些网页信息的计算机，则称为服务器。

服务器的功能并不只是单纯地存放网页信息，任何可提供网络服务的计算机都是服务器。例如，提供网页信息的称为网页服务器，而提供文件上传与下载功能的则称为文件服务器。

3. 通信协议

在网络上要能彼此互通信息就必须遵循一定的沟通方式，这些沟通方式即所谓的通信协议，表 1-1 所示为 WWW 中较为常用的通信协议。

表 1-1 WWW 常用通信协议

通 信 协 议	说 明
http	最常用的通信协议之一，主要用来传送文字、图片、声音等多媒体类型的数据，也是 Web 信息网传输网页所使用的通信协议
ftp	可用来上传文件至远程主机，或从指定的远程主机下载文件至本地计算机
mail	主要用来建立邮件发送服务
telnet	使用于远程登录，例如登录 BBS 系统
news	用于取得网络新闻论坛的协议

只有双方都使用相同的通信协议，才能建立起连接通道。

4. 全球资源定位器——URL

URL 的英文全称为 Uniform Resource Locator(全球资源定位器)，是对可以从互联网上得到的资源的位置和访问方法的一种简洁的表示，是互联网上标准资源的地址。互联网上的每个文件都有一个唯一的 URL，它包含的信息指出文件的位置及浏览器应该怎么处理它。当用户想要打开位于远程网站主机上的网页时，必须指定其 URL 位置，也就是通常所讲的网址，它从左到右由下述部分组成。

- Internet 资源类型(scheme): 指出 WWW 客户程序用来操作的工具。如“http://”表示 WWW 服务器，“ftp://”表示 FTP 服务器，“gopher://”表示 Gopher 服务器，而“new:”表示 Newgroup 新闻组。
- 服务器地址(host): 指出 WWW 页所在的服务器域名。
- 端口(port): 有时需要，对某些资源的访问来说，需给出相应的服务器提供端口号。
- 路径(path): 指明服务器上某资源的位置(其格式与 DOS 系统中的格式一样，通常由“目

录/子目录/文件名”这样的结构组成)。与端口一样,路径并非总是需要的。

URL 地址格式排列为: scheme://host:port/path。例如, http://www.oracle.com/technetwork/java/index.html 就是一个典型的 URL 地址。当浏览器接收了用户输入的一个网址后,便会根据其中所提供的信息,向服务器提出网页浏览请求。

1.2 静态网页与 HTML

HTML 称为超文本标记语言,是一种用于开发静态网页的技术语言,主要提供创建网页文件所需的标准语法,以及描述网页数据的呈现方式,其中包含超级链接、图形或声音影像等多媒体内容。

静态网页主要由各种规范数据展现格式的标签组成,图 1-4 所示为显示“JSP 动态网页技术”的 HTML 网页文件 jsp.html 在浏览器上的显示结果。

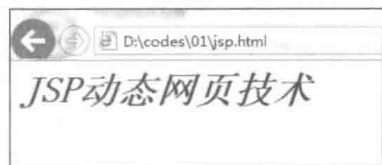


图 1-4 静态网页在浏览器中的显示示例

使用记事本打开并查看这份网页的 HTML 如下:

```
<html>
<head>
<title>JSP</title>
</head>

<body>
  <b><font size="6"><i>JSP 动态网页技术</i></font></b>
</body>

</html>
```

可以看到,其内容由各种 HTML 标签组成,这些标签各有其特定的意义。例如, ... 表示其中包含的文字要以粗体的外观显示, <i>...</i> 则是将其中包含的文字设置为斜体格式。

由于 HTML 大量使用于网页的建构当中,因此编写 HTML 的相关软件也发展得相当成熟,网页设计人员几乎可以不用了解 HTML 标签即能设计出各种出色的网页。当然,学习 JSP 网页程序设计,了解 HTML 是不可缺少的必备知识。在本书后面的内容中,对于 HTML 与 JSP 的相关应用,将会有详细的介绍。

静态网页的 HTML 技术对于文件内容的展示,表现得相当称职。用户在浏览网页的时候,可以轻松查看网页设计者预先设计好的内容,网页一旦设计完成,内容就不会再变动,无论使用的网页浏览器、浏览用户的身份如何不同,他们所见到的内容都是相同的。

在 Web 发展初期,HTML 所建构的静态网页的确达到了信息传递的主要目的。然而随着 Web 网站各种应用服务的快速增长,静态网页单向呈现信息的特性,很快便无法满足实际的应用需求。尤其是越来越多的商业活动在网络上进行,网站与用户的互动需求开始浮现,如何创建出更加吸引用户的网站内容,这对静态网页的发展提出了新的要求。

一个购物网站根据到访的用户身份显示出不同的信息,以提供个性化的网页内容;一个门户网站也可能根据用户的习惯,将其喜好的信息内容显示在每次访问的首页。为了满足这些需

求, 具备动态特性的交互式网页开始出现, CGI、ASP 及 JSP 便是用于建构动态网页的相关技术, 在下一节的内容当中, 我们继续来介绍这一方面的相关概念。

1.3 交互式动态网页技术

由于静态 HTML 网页不能与网站用户进行互动, 因此出现了各种提供交互式功能的网页技术。本节将介绍交互式网页的概念, 同时说明交互式网页技术领域的主流技术 JSP 与其他动态网页技术。

1.3.1 动态网页的程序语言

HTML 网页只能够提供各种静态的多媒体信息, 却不能与联机用户进行数据交换等动态交流。为了让网页根据用户不同的行为作出动态响应, 因此产生了内嵌于 HTML 网页的程序语言, 这一类程序语言依解释方式的不同, 可以区分为两类: 客户端 Script 语言和服务器端网页语言。下面分别对这两种网页语言进行说明。

1.3.2 在客户端执行的网页语言

在客户端执行的网页语言内嵌在 HTML 中, 而包含这类客户端执行程序的网页扩展名同样是 .htm。当浏览器向服务器请求打开网页时, 服务器会将整份网页传送到客户端, 由浏览器进行网页程序解释操作, 并且将结果显示在浏览器窗口中, 其过程如图 1-5 所示。

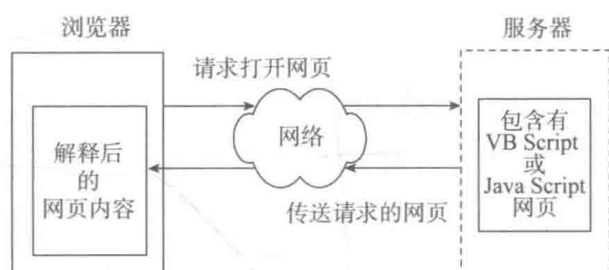


图 1-5 在客户端执行的网页语言

该过程说明如下:

- (1) 用户通过浏览器指定 URL, 向网页服务器请求特定的网页内容。
- (2) 服务器加载指定的 HTML 网页, 返回至客户端的网页浏览器进行解释。
- (3) 浏览器取得 HTML 文件, 解释其中的 HTML 及 Script 网页程序代码。
- (4) 显示 HTML 网页。

在客户端执行的网页语言可在网页中产生动态的效果, 如各类网页特效, 同时也能够在客户端与服务器端之间作数据交换时, 先行处理一些事前的准备操作。例如, 一般提供会员登录功能的网页, 通常都会利用客户端的 Script 提供输入数据的校验功能, 当用户输入不正常的的数据(如不合法的身份证信息)时, 登录网页的操作将会失败, 相关的资料则无法返回至服务器端进行处理。

客户端的 Script 语言, 可以直接在浏览器这一端完成一些工作, 而不需要将所有的工作都返回至服务器端, 这样可以降低服务器的负担并提高执行的效率, 为一些大型网站减轻负担提供了很好的解决方案。

目前可以提供动态网页的 Script 语言有两种, 分别是 JavaScript 和 VBScript。其中, JavaScript