

以经验为后盾 以实用为目标 以实例为导向 以实践为指导

JSP网站 开发详解

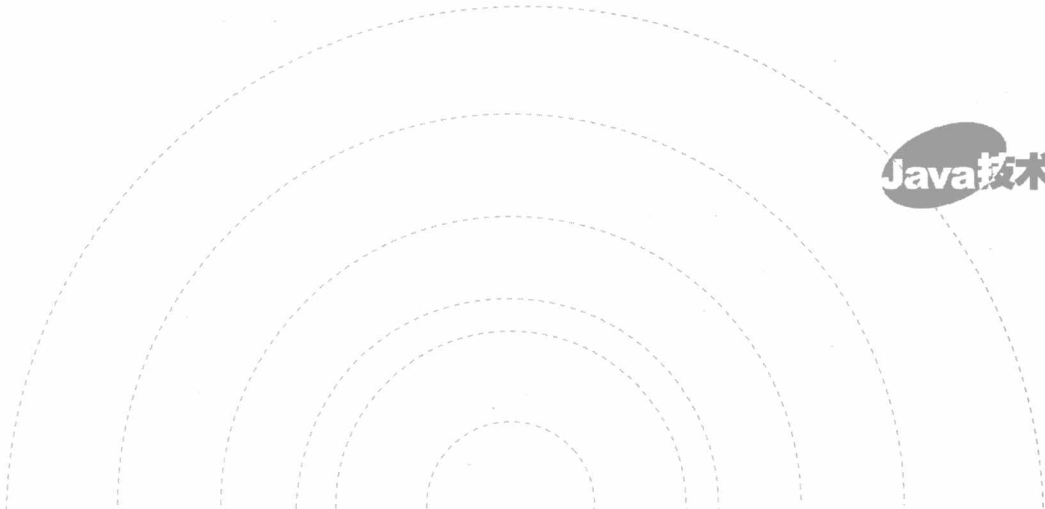
(JSP+Servlet+Tomcat+Eclipse+SSH)

唐友国 湛洪波 编著

光盘包含：
本书所有实例代码



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>



Java技术大系

JSP网站 开发详解

(JSP+Servlet+Tomcat+Eclipse+SSH)

唐友国 湛洪波 编 著

电子工业出版社

Publishing House of Electronics Industry

北京•BEIJING

内 容 简 介

本书介绍了使用 JSP 及其相关技术开发网络应用程序的基础知识和方法, 全书共分为 3 个部分。其中第 1 部分介绍了使用 JSP 开发网络应用系统的必备知识; 第 2 部分介绍了将 JSP 与 Struts、Hibernate 和 Spring 等技术结合的基础知识和集成方法; 第 3 个部分精选了实用性很强的 6 个典型模块和 3 个小型的网络应用程序, 分别是新闻发布系统、电子相册系统和在线购物系统, 书中从系统需求分析、代码设计到方案实施全面地介绍了这些实例。本书的随附光盘中包含了所有这些网络应用程序实例的源程序, 读者稍加修改, 即可将其应用于自己的开发项目中。

本书适用于使用 JSP 开发网络应用程序的初、中级技术人员, 以及对相关技术感兴趣的读者, 也可以作为高等院校相关专业的辅导教材。

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 网站开发详解/唐友国, 湛洪波编著. —北京: 电子工业出版社, 2008.10

(Java 技术大系)

ISBN 978-7-121-07126-3

I. J… II. ①唐…②湛… 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(2008)第 107485 号

责任编辑: 朱沐红

印 刷: 北京市通州大中印刷厂

装 订: 三河市鹏成印业有限公司

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

开 本: 787×1092 1/16 印张: 45.75 字数: 1035 千字

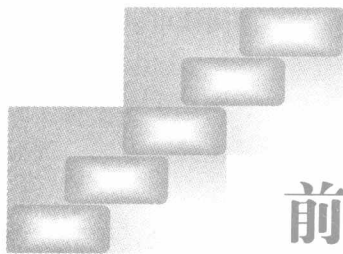
印 次: 2008 年 10 月第 1 次印刷

印 数: 4000 册 定价: 79.00 元(含光盘 1 张)

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, 联系及邮购电话: (010) 88254888。

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

服务热线: (010) 88258888。

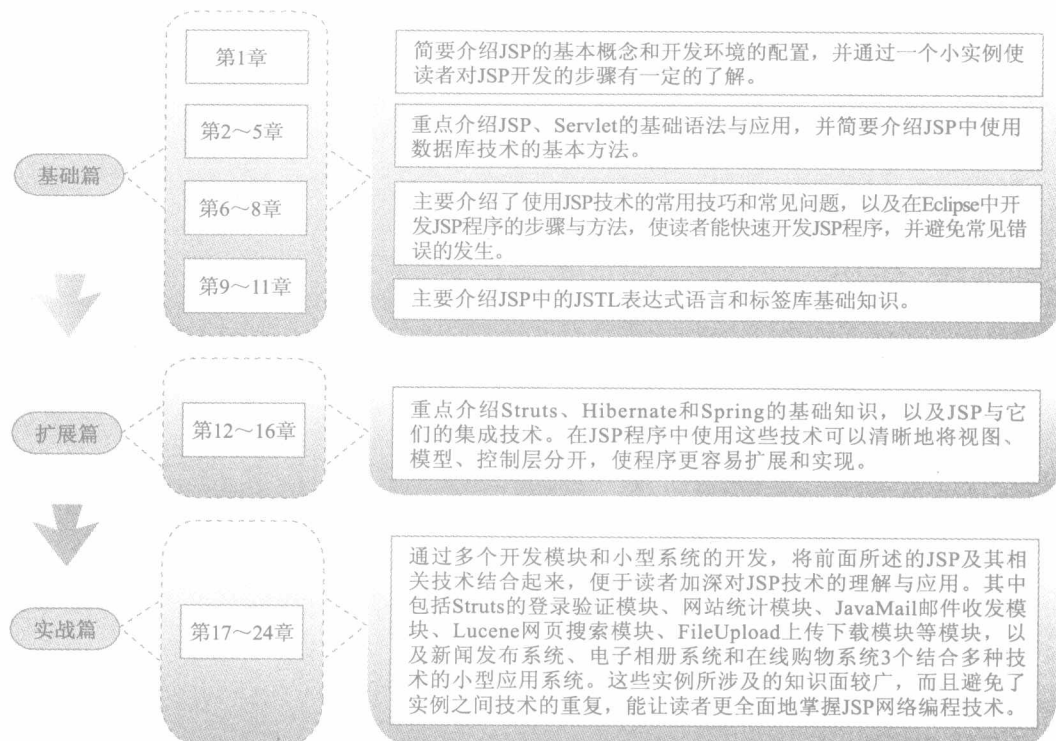


前言

计算机软件技术的日新月异推动了社会信息化的进程，这也使得开发人员不得不学习新的技术以适应这种局面。在 Web 应用领域，动态网页技术作为 Web 系统开发的核心，越来越受到业内人士的青睐。目前主流的动态网页技术主要是指 ASP、PHP 和 JSP，本书全面讲述 JSP 技术及其相关技术，如与 Struts、Hiberante 和 Spring 的集成等。

考虑到 Java Web 应用开发的知识和技能需要和读者的知识结构，本书将 JSP 基础知识和 Java Web 应用开发经验融为一体。内容上做到全面而深入，囊括 JSP 技术的核心和精髓，以及 Java Web 应用开发整体解决方案知识。通过由浅入深，实例举证的分析讲解，使得读者在了解 JSP 基础知识的同时，具备全面的 Java Web 应用系统开发知识。

本书主要分 3 个部分介绍 JSP 技术及与其他相关技术集成的知识。



目 录

第 1 篇 基础篇

第 1 章 JSP 开发基础 1

- 1.1 JSP 的概念和作用 1
 - 1.1.1 什么是 JSP 1
 - 1.1.2 JSP 和 Servlet 扮演的角色 2
 - 1.1.3 JSP 和其他框架的集成 3
 - 1.1.4 学习 JSP 网站编程 7
- 1.2 安装和配置 JDK 8
 - 1.2.1 安装 JDK 8
 - 1.2.2 配置 JDK 11
- 1.3 安装和配置 Tomcat 12
 - 1.3.1 安装 Tomcat 服务器 13
 - 1.3.2 测试安装是否成功 15
- 1.4 开发第 1 个 JSP 页面 16
 - 1.4.1 创建 Form 提交表单 16
 - 1.4.2 创建表单处理 Servlet 16
 - 1.4.3 创建 JSP 显示页面 17
 - 1.4.4 部署第 1 个 Web 应用 18
- 1.5 小结 19

第 2 章 JSP 基本语法 20

- 2.1 JSP 基础 20
 - 2.1.1 JSP 和 Servlet 的关系 20
 - 2.1.2 典型的 JSP 文件 24
 - 2.1.3 JSP 文件组成结构 25
 - 2.1.4 JSP 的生命周期 27
- 2.2 JSP 页面元素 28
 - 2.2.1 页面指令元素 29

2.2.2 脚本元素 37

2.2.3 动作元素 38

2.3 JSP 内建对象 45

2.3.1 输出对象 out 46

2.3.2 请求对象 request 47

2.3.3 响应对象 response 50

2.3.4 会话对象 session 52

2.3.5 页面上下文对象 pageContext 55

2.3.6 全局对象 application 56

2.3.7 Servlet 初始化参数对象 config 57

2.3.8 页面对象 page 58

2.4 使用 JSP 实现简单登录实例 59

2.4.1 创建登录 HTML 表单 59

2.4.2 使用 Request 获得数据 60

2.4.3 处理表单数据 61

2.4.4 使用 Session 保存用户对话 62

2.5 小结 63

第 3 章 Servlet 基础 64

3.1 Servlet 概念 64

3.1.1 什么是 Servlet 64

3.1.2 Servlet 技术的特点 65

3.1.3 Servlet 的生命周期 66

3.1.4 开发并部署一个简单的 Servlet 68

3.2 基本 Servlet 结构 69

3.2.1 init 方法 70

3.2.2 service 方法 72

3.2.3 doGet、doPost 和 doXxx 方法 72

3.2.4 destroy 方法 75

3.3 Servlet 的基本配置 77

3.3.1 Servlet 的名称及路径配置 79

3.3.2 初始化参数 80

3.3.3	启动装入优先级	80
3.3.4	Servlet 映射	81
3.4	Servlet 请求和响应	83
3.4.1	Request 实现请求	83
3.4.2	Response 处理响应	90
3.5	Servlet 请求转发	96
3.5.1	使用 sendRedirect 转发	96
3.5.2	使用 Dispatch 转发	98
3.5.3	Redirect 和 Request Dispatch 的区别	104
3.6	使用 Servlet 改进登录实例	104
3.6.1	在 Servlet 中获得数据	105
3.6.2	使用 Servlet 将用户转向到 指定页面	107
3.6.3	在 Servlet 中使用 Session	108
3.7	小结	109

第 4 章 JSP 和 Servlet 的集成 110

4.1	JSP 和 Servlet 结合的两模式	110
4.1.1	模式一	110
4.1.2	模式二	113
4.1.3	两种模式的比较	117
4.1.4	MVC 模式	117
4.1.5	JSP 和 Servlet 实现的 MVC 模式	119
4.2	使用 MVC 编程改进登录实例	120
4.2.1	JSP 实现表现层	120
4.2.2	JavaBean 模型层	121
4.2.3	Servlet 控制层 Control	123
4.3	小结	124

第 5 章 JDBC 数据库访问技术 125

5.1	JDBC 和驱动程序	125
5.1.1	为什么需要 JDBC	126
5.1.2	数据库驱动程序	126
5.1.3	使用 JDBC 访问数据库的过程	127
5.2	JDBC 处理数据库方法	129
5.2.1	加载并注册数据库驱动	129
5.2.2	建立到数据库的连接	130
5.2.3	访问数据库	131

5.2.4	Statement 和 PreparedStatement	133
5.2.5	ResultSet 结果集	140
5.2.6	事务处理	142
5.3	连接不同的数据库	145
5.3.1	连接 MySQL 数据库	145
5.3.2	连接 MSSQL 数据库	146
5.3.3	连接 Oracle 数据库	147
5.4	存取二进制文件	147
5.4.1	二进制文件的存取过程	147
5.4.2	在数据库中存取图片	148
5.5	使用 JDBC 操作数据库示例	151
5.5.1	在 JSP 页面中查询数据	151
5.5.2	分页	155
5.5.3	增、删及改数据	161
5.6	使用 JDBC 改进用户登录模块	167
5.6.1	在 JSP 中访问数据库	167
5.6.2	在 Servlet 中访问数据库	169
5.6.3	在 JavaBean 中访问数据库	170
5.7	小结	172

第 6 章 JSP 常用技巧 173

6.1	会话跟踪技巧	173
6.1.1	使用 session 的会话跟踪	174
6.1.2	使用 cookie 跟踪会话	176
6.2	数据共享技巧	179
6.2.1	使用 Java static 属性	179
6.2.2	基于 session 会话	181
6.2.3	基于 Web 范围	182
6.3	错误处理技巧	185
6.3.1	JSP 错误的常见类型	185
6.3.2	定向错误到友好页面	186
6.3.3	Web.xml 中的配置 HTTP 错误	188
6.3.4	在 JSP 页面中处理错误	189
6.4	JSP 文件上传技巧	190
6.4.1	使用 JSP SmartUpload 上传	190
6.4.2	SmartUpload 文件上传实例	193
6.5	在 JSP 中使用 JavaBean	196
6.5.1	使用 JavaBean 的原因	196
6.5.2	JavaBean 的基本概念	196

6.5.3	在 JSP 中使用 JavaBean	198
6.5.4	访问 JavaBean 属性	200
6.5.5	设置 JavaBean 属性	201
6.6	小结	202

第 7 章 JSP 中常见问题及其处理 203

7.1	中文乱码问题	203
7.1.1	出现中文乱码的原因	204
7.1.2	JSP 显示中文乱码处理	205
7.1.3	JSP 页面传递中文乱码处理	206
7.1.4	Servlet 接收参数中文乱码处理	208
7.1.5	URL 参数中文乱码处理	209
7.1.6	数据库中文乱码处理	210
7.2	数据库处理问题	211
7.2.1	找不到数据库驱动 Driver	212
7.2.2	无法连接数据库	212
7.2.3	数据库连接线程不足的问题	213
7.2.4	数据库查询性能差的问题	215
7.3	国际化问题	219
7.3.1	在 JSP 中实现国际化	219
7.3.2	时间格式的国际化	222
7.3.3	语言国际化	225
7.4	JSP 安全问题	227
7.4.1	防范 SQL 注入式攻击	227
7.4.2	JSP 中使用认证授权	229
7.5	小结	235

第 8 章 在 Eclipse 中开发 JSP 和 Servlet 236

8.1	MyEclipse 的下载与安装	236
8.1.1	安装 Eclipse	237
8.1.2	安装 Eclipse 多国语言包	237
8.1.3	下载和安装 MyEclipse	238
8.1.4	绑定 Tomcat 服务器与 MyEclipse	241
8.1.5	在 Eclipse 中启动 Tomcat 服务器	241
8.2	在 Eclipse 中创建 Web 工程	242
8.2.1	新建工程向导	242
8.2.2	Web 工程目录结构	244

8.2.3	在 Eclipse 中添加类库	245
8.3	新建 JSP 页面	247
8.3.1	使用向导新建 JSP 页面	247
8.3.2	在 Eclipse 中编辑 JSP 页面	249
8.3.3	添加表单 Form 元素	251
8.4	新建 Servlet	253
8.4.1	使用向导新建 Servlet	253
8.4.2	查看 Web.xml	256
8.4.3	编写处理代码	256
8.5	连接数据库	257
8.5.1	添加 MySQL 数据库驱动程序	258
8.5.2	建立数据库表	258
8.5.3	新建连接数据库类	259
8.5.4	修改 Servlet 处理数据库	260
8.6	在 Eclipse 中调试 JSP 和 Servlet	261
8.6.1	部署 Web 项目	261
8.6.2	运行 JSP	263
8.6.3	在 Servlet 中设置断点	263
8.6.4	在 JSP 中设置断点	263
8.6.5	跟踪监控变量	264
8.7	小结	266

第 9 章 表达式语言 EL 267

9.1	表达式语言概述	267
9.2	语法	268
9.2.1	[] 和操作符	268
9.2.2	算术运算符	269
9.2.3	关系运算符	270
9.2.4	逻辑运算符	271
9.2.5	Empty 运算符	272
9.2.6	条件运算符	272
9.2.7	改变运算符的优先级	272
9.3	隐含对象	273
9.4	命名变量	278
9.5	保留字	279
9.6	表达式语法实例	279
9.7	其它功能	280
9.8	小结	281

第 10 章 标准标签库 JSTL 282

10.1	JSTL 概念	282
------	---------	-----

10.1.1	JSTL 概述	283
10.1.2	安装 JSTL 标签库	283
10.2	一般用途的标签	285
10.2.1	<c:out> 标签	285
10.2.2	<c:set> 标签	286
10.2.3	<c:remove> 标签	288
10.2.4	<c:catch> 标签	288
10.3	条件标签	291
10.3.1	<c:if> 标签	291
10.3.2	<c:choose> 标签	292
10.3.3	<c:when> 标签	293
10.3.4	<c:otherwise> 标签	293
10.4	迭代标签	294
10.4.1	<c:forEach> 标签	294
10.4.2	<c:forEachTokens> 标签	297
10.5	与 URL 有关的标签	299
10.5.1	<c:import> 标签	299
10.5.2	<c:redirect> 标签	302
10.5.3	<c:url> 标签	303
10.5.4	<c:param> 标签	304
10.6	函数标签	305
10.6.1	<fn:startsWith> 标签	305
10.6.2	<fn:endsWith> 标签	305
10.6.3	<fn:indexOf> 标签	306
10.6.4	<fn:replace> 标签	306
10.6.5	<fn:substring> 标签	306
10.6.6	<fn:split>	306
10.7	使用标签库改进用户登录实例	307
10.8	小结	309

第 11 章 定义标签扩展..... 310

11.1	JSP 标签扩展	310
11.1.1	自定义标签的原因	310
11.1.2	标签类型及结构	312
11.2	自定义标签接口	313
11.2.1	Tag 接口	314
11.2.2	IteratorTag 接口	315
11.2.3	BodyTag 接口	316
11.2.4	SimpleTag 接口	317
11.3	标签开发实例	318

11.3.1	实现 Tag 接口	319
11.3.2	从 TagSupport 继承	321
11.4	标签库配置	322
11.4.1	创建标签库描述文件	322
11.4.2	在 Web 中使用标签	324
11.5	小结	325

第 2 篇 扩展篇

第 12 章 基于 MVC 的 Struts 框架..... 326

12.1	JSP 集成 Struts 框架	326
12.1.1	Struts 体系结构	326
12.1.2	Struts 的 MVC 模式	327
12.1.3	Struts 流行的原因	329
12.1.4	在 Struts 中集成 JSP	329
12.2	Struts 核心组件	330
12.2.1	控制组件 Action	330
12.2.2	模型组件	331
12.2.3	视图组件 ActionForm	335
12.3	Struts HTML 标签库	336
12.3.1	Struts 的标签体系	336
12.3.2	HTML 标签库	337
12.3.3	HTML 标签	338
12.3.4	Form 标签	340
12.4	Struts Bean 标签库	342
12.5	Struts Logic 标签库	344
12.6	Struts 验证 Validator 框架	347
12.6.1	Validator 框架	347
12.6.2	安装与配置及应用 Validator 框架	349
12.7	小结	351

第 13 章 Eclipse 中开发 Struts 登录实例..... 352

13.1	MyEclipse 对 Struts 的支持	352
13.1.1	文件创建向导支持	353
13.1.2	struts-config.xml 可视化编 辑器	353
13.2	在 Eclipse 中创建 Struts 工程	355

13.2.1	新建 Struts 工程向导	355
13.2.2	Struts 项目结构	356
13.2.3	Struts 类的包结构	356
13.2.4	查看 web.xml 和 struts-config.xml 文件	357
13.3	利用向导创建 JSP	359
13.3.1	创建 JSP 页面	359
13.3.2	创建 ActionForm	359
13.3.3	创建 Action	362
13.3.4	创建 Forward	363
13.3.5	查看 Struts-config.xml 文件	364
13.4	在 Eclipse 中编辑 JSP 页面	365
13.4.1	Struts 标签输入支持	366
13.4.2	编写 struts-config.xml 文件	367
13.5	Struts 验证支持	367
13.5.1	添加验证支持配置	367
13.5.2	添加页面输入验证	369
13.6	在 Eclipse 中调试 Struts	371
13.6.1	部署 Struts 项目	371
13.6.2	运行入口页面	372
13.6.3	在 Action 中设置断点	372
13.6.4	跟踪 ActionForm 变量	372
13.7	小结	374

第 14 章 使用 Hibernate 实现 JSP 持久层 375

14.1	Hibernate ORM 技术	375
14.1.1	Hibernate 概念和体系结构	375
14.1.2	Hibernate 数据库处理 API	377
14.2	Hibernate 基本配置	380
14.2.1	Hibernate 的下载和安装	380
14.2.2	Hibernate 开发的基本步骤	381
14.2.3	Hibernate 配置文件	382
14.2.4	持久化类	382
14.2.5	对象/关系映射文件	384
14.3	Hibernate 基本操作	385
14.3.1	持久化对象	385
14.3.2	查询对象	385
14.3.3	更新对象	387
14.3.4	删除对象	388
14.3.5	Hibernate 基本操作实现实例	388

14.4	Hibernate 对象标识符	392
14.5	Hibernate 查询 HQL 语言	393
14.5.1	HQL 语言特点	394
14.5.2	HQL 查询构成	394
14.5.3	子查询	396
14.6	在 JSP 中使用 Hibernate	396
14.6.1	使用 Hibernate 的 JSP 项目结构	396
14.6.2	在 JSP 中使用 Hibernate 访问数据库	397
14.7	小结	402

第 15 章 使用 Spring 集成 JSP 403

15.1	Spring 概述	403
15.1.1	控制反转 IoC 容器	404
15.1.2	面向方面的 AOP 编程	405
15.1.3	Spring 如何集成 JSP	407
15.2	基于 Spring IoC 容器开发	407
15.2.1	Spring 的依赖注入	407
15.2.2	Bean 封装机制	409
15.2.3	Bean 的基本配置	410
15.3	Spring 集成 JDBC	414
15.3.1	Spring 及 JDBC 支持	414
15.3.2	JdbcTemplate API 技术	415
15.3.3	JdbcTemplate 与 Template 设计模式	419
15.3.4	JdbcTemplate 对 JDBC 的改进	420
15.3.5	JdbcTemplate 与数据库事务管理	422
15.3.6	在 JSP 中使用 Spring 集成 JDBC	424
15.4	Spring 集成 Hibernate	429
15.4.1	Spring 的 Hibernate 支持	429
15.4.2	HibernateDaoSupport 支持	429
15.4.3	Hibernate SessionFactory 注入	431
15.4.4	使用 HibernateTemplate 模板	432
15.4.5	HibernateTemplate 的简捷方法	435
15.4.6	在 JSP 中使用 Spring 集成 Hibernate	435
15.5	小结	440

第 16 章 在 Eclipse 中开发 Hibernate 和 Spring..... 441

- 16.1 在 Eclipse 中开发 Hibernate..... 441
 - 16.1.1 为已有 JSP 项目添加 Hibernate 支持.....442
 - 16.1.2 Hibernate 项目结构443
 - 16.1.3 Hibernate 类包结构444
 - 16.1.4 hibernate.cfg.xml 文件.....445
- 16.2 创建映射文件 446
 - 16.2.1 连接到数据库447
 - 16.2.2 新建 Java 类和映射文件449
 - 16.2.3 完善 Java 类及映射文件450
- 16.3 应用 DAO 模式..... 451
 - 16.3.1 编写业务层及其方法451
 - 16.3.2 编写 DAO 业务接口452
 - 16.3.3 编写 DAO Hibernate 实现.....453
 - 16.3.4 在 JSP 和 Servlet 中调用 Hibernate458
 - 16.3.5 调试 Hibernate 应用460
- 16.4 在 Eclipse 中开发 Spring..... 461
 - 16.4.1 为已有 JSP 项目添加 Spring 支持461
 - 16.4.2 Spring 项目结构462
 - 16.4.3 Spring 类包结构462
 - 16.4.4 applicationContext.xml 文件.....463
- 16.5 使用 Spring ORM 改进 Hibernate..... 464
 - 16.5.1 修改 Spring 配置文件464
 - 16.5.2 创建 Hibernate sessionFactory 连接.....465
 - 16.5.3 创建 pojo 映射466
 - 16.5.4 创建 HibernateTemplate.....467
- 16.6 小结 468

第 3 篇 实战篇

第 17 章 登录验证模块469

- 17.1 在 MySQL 中建立用户信息表 ..469
 - 17.1.1 确定用户字段470

- 17.1.2 创建数据库和用户表..... 470
- 17.2 登录验证的建模分析.....471
 - 17.2.1 活动图..... 471
 - 17.2.2 登录与注册的运行流程分析..471
- 17.3 创建用户数据连接缓冲池472
 - 17.3.1 下载与设置数据连接缓冲池类包..... 472
 - 17.3.2 创建数据连接缓冲池..... 473
 - 17.3.3 重载 ActionServlet 以应用连接缓冲池.....474
 - 17.3.4 配置 web.xml 应用重载的 DataActionServlet..... 476
- 17.4 实现用户登录.....476
 - 17.4.1 创建登录视图组件..... 476
 - 17.4.2 创建登录的控制器组件.....478
 - 17.4.3 创建模型组件.....480
 - 17.4.4 配置 web.xml 和 struts-config.xml 文件481
 - 17.4.5 验证用户输入数据的有效性...484
 - 17.4.6 登录演示..... 485
- 17.5 实现新用户注册.....485
 - 17.5.1 创建注册视图组件.....486
 - 17.5.2 创建注册的控制器组件.....487
 - 17.5.3 创建模型组件.....489
 - 17.5.4 验证用户输入数据的有效性..489
 - 17.5.5 配置 web.xml 和 struts-config.xml 文件490
 - 17.5.6 配置资源包属性文件..... 491
 - 17.5.7 注册演示.....492
- 17.6 小结492

第 18 章 网站统计模块 493

- 18.1 需求分析及数据库设计.....493
 - 18.1.1 需求分析..... 494
 - 18.1.2 数据库设计..... 494
- 18.2 获取统计数据.....495
 - 18.2.1 获取客户端信息.....495
 - 18.2.2 截取用户操作系统和浏览器信息..... 497
 - 18.2.3 保存信息到数据库.....498

18.2.4	防止重复记录	500
18.3	显示网站访问计数器	502
18.3.1	显示当前在线人数计数器	502
18.3.2	显示网站访问量计数器	503
18.3.3	使用图片样式显示计数器	504
18.4	生成统计分析结果	505
18.4.1	使用 SQL 命令统计分析	505
18.4.2	JFreeChart 简介	507
18.4.3	生成日流量统计折线图	509
18.4.4	生成月流量统计柱状图	513
18.4.5	生成访问时段分布统计 柱状图	516
18.4.6	生成浏览器类型统计 饼状图	518
18.4.7	生成操作系统类型统计立体 饼状图	520
18.5	小结	523

第 19 章 邮件收发模块 524

19.1	邮件相关协议简介	524
19.2	JavaMail 简介	525
19.2.1	配置 JavaMail 开发环境	526
19.2.2	JavaMail 邮件处理过程	526
19.2.3	JavaMail 常用类	526
19.2.4	一个简单的邮件发送程序	529
19.3	发送 HTML 格式的邮件	531
19.3.1	撰写邮件页面	531
19.3.2	邮件发送 Servlet	532
19.3.3	配置 web.xml 文件及运行	534
19.3.4	处理主题与正文中的乱码 问题	535
19.4	发送带附件的邮件	536
19.4.1	发送本地文件形式的附件	537
19.4.2	发送远程文件形式的附件	537
19.5	实现邮件接收	538
19.5.1	接收邮件的一般过程	538
19.5.2	一个简单的邮件接收 Servlet 程序	539
19.6	小结	541

第 20 章 网页搜索模块 542

20.1	搜索引擎概述	542
20.1.1	搜索引擎分类	543
20.1.2	搜索引擎的实现原理	545
20.1.3	搜索引擎的相关技术	546
20.2	用第三方 API 实现站内搜索	547
20.2.1	使用 Google API	547
20.2.2	使用百度 API	548
20.3	基于 Java 的全文搜索引擎 Lucene	549
20.3.1	Lucene 概述	550
20.3.2	搜索应用程序与 Lucene 之间的关系	550
20.3.3	Lucene 类包简介	551
20.3.4	建立索引	551
20.3.5	搜索索引	554
20.4	构建 Lucene Web 搜索程序	556
20.4.1	搜索程序的工作流程	556
20.4.2	建立搜索页面	557
20.4.3	建立搜索控制 Servlet	558
20.4.4	创建索引	559
20.4.5	搜索索引	561
20.4.6	运行程序	564
20.5	小结	565

第 21 章 上传和下载模块 566

21.1	上传和下载简介	566
21.2	一个简单的文件下载程序	566
21.3	通过 Servlet 实现文件上传	568
21.3.1	建立文件上传表单	568
21.3.2	利用 Servlet 实现文件上传	569
21.3.3	配置 web.xml 并运行	570
21.4	使用 FileUpload 组件上传文件	571
21.4.1	FileUpload 上传处理过程	571
21.4.2	一个简单的文件上传实例	573
21.4.3	配置 web.xml 并运行	576
21.4.4	查看上传进度	577
21.5	小结	578

第 22 章 新闻发布系统 579

22.1 系统分析与总体设计	579
22.1.1 需求分析	580
22.1.2 总体设计	580
22.2 数据库设计与实现	581
22.2.1 数据库需求分析	581
22.2.2 数据库逻辑设计	582
22.2.3 创建数据表	583
22.3 编写数据库处理类	585
22.4 编写 JavaBean 实现	587
22.4.1 处理管理员数据 JavaBean	587
22.4.2 处理一级栏目数据 JavaBean	588
22.4.3 处理二级栏目数据 JavaBean	588
22.4.4 处理新闻数据 JavaBean	589
22.5 设计后台页面与功能实现	590
22.5.1 实现管理员登录	591
22.5.2 实现一级栏目的功能	597
22.5.3 实现二级栏目的功能	604
22.5.4 实现栏目下的相关新闻功能	607
22.5.5 后台管理主页面集成	614
22.5.6 演示后台功能	619
22.6 前台页面设计与功能实现	624
22.6.1 主页面设计与功能实现	624
22.6.2 新闻显示页面设计与功能实现	626
22.7 小结	627

第 23 章 在线相册系统 628

23.1 系统分析与总体设计	628
23.1.1 系统功能描述	628
23.1.2 选择平台与架构	629
23.2 数据库设计与实现	629
23.2.1 数据库需求分析	629
23.3 实现 DAO 层	631
23.3.1 DAO 组件的定义	632
23.3.2 实现 DAO 组件	633
23.4 实现 Web 层	651
23.4.1 相册管理	651

23.4.2 相册列表	655
23.4.3 上传相片	659
23.4.4 查看相片	664
23.4.5 管理相片	667
23.5 小结	667

第 24 章 在线购物系统 668

24.1 系统分析与总体设计	668
24.2 数据库设计与实现	669
24.2.1 数据库需求分析	669
24.2.2 数据库逻辑设计	670
24.3 设计与搭建架构	673
24.3.1 设计视图层	674
24.3.2 设计控制层	674
24.3.3 设计对象模型	676
24.3.4 设计数据访问层	678
24.4 在线购物页面	682
24.4.1 实现购书主页	682
24.4.2 实现二级分类页	685
24.4.3 实现图书详细页	686
24.5 商品查询	687
24.5.1 实现表示层	687
24.5.2 实现业务逻辑	688
24.5.3 实现数据访问层	689
24.6 管理购物车	689
24.6.1 添加商品到购物车	690
24.6.2 更新图书数量	696
24.6.3 取消购物车中的商品	699
24.7 生成订单导航页面	700
24.7.1 确认收货及定货地址	701
24.7.2 选择送货时间及运费	706
24.7.3 选择包装类型及费用	708
24.7.4 选择付款方式	709
24.7.5 选择缺货处理及余款处理 方式	709
24.7.6 确认订单	711
24.7.7 成功生成订单	715
24.8 小结	717

第1章 JSP 开发基础



1

JSP (Java Server Pages, Java 服务器页面) 是 SUN Microsystems 公司于 1999 年末在企业级 Java 工具集中加入的一个新元素, 它建立在 Java Servlet 之上。其设计目的是使开发人员, 甚至非开发人员都能高效率地创建 Web 内容。在过去几年里, 使用 JSP 技术的开发人员飞速增加, 它已经变成 Web 开发的主流技术之一。

本章将介绍 JSP 开发基础, 包括它与 Servlet 及其他 Web 框架的关系, 以及安装和配置开发 JSP 所需的软件环境, 最后将通过创建第 1 个 JSP 页面使读者对其有一个初步的认识。

本章要点

- (1) JSP 的概念及作用。
- (2) JSP 和 Servlet 及其他 Web 框架的关系。
- (3) 安装配置 JDK 及 Tomcat 应用服务器。
- (4) 开发第 1 个 JSP 页面。

1.1 JSP 的概念和作用

按照脚本语言是服务于某一个子系统的语言这种论述, JSP 应当被看做是一种脚本语言。然而作为一种脚本语言, JSP 又显得过于强大, 因为在其中几乎可以使用全部的 Java 类。

本节我们将介绍 JSP 的基础知识及学习建议。

1.1.1 什么是 JSP

简要地说, JSP 是用来开发含有动态内容网页的一种技术。纯 HTML 网页只会包含

静态的内容，即其内容通常保持不变。而 JSP 页面则不同，它可以根据任意数量的变量来动态调整自己页面中的内容，这些变量可以是身份验证信息、用户提供的对象信息、页面颜色，还可以是用户所做的一切选择等。这些功能对于创建像淘宝网这类在线购物及旅店、机票预订，以及对内容进行多国语言的国际化和个性化的 Web 应用程序来说是非常关键的。

JSP 虽然功能强大，但其页面和常规的 HTML 网页一样，也包含标准的标记语言元素，如 HTML 的标签。同时，JSP 页面还包括自身特殊的 JSP 元素，这些元素将允许服务器把动态内容添加到页面上。JSP 自己的元素用途非常广泛，它可以读取数据库的数据信息，以及记录用户需要保存的信息等。

现在来看一个简单的 JSP 页面执行过程。当用户请求一个 JSP 页面时，JSP 执行过程如图 1-1 所示。

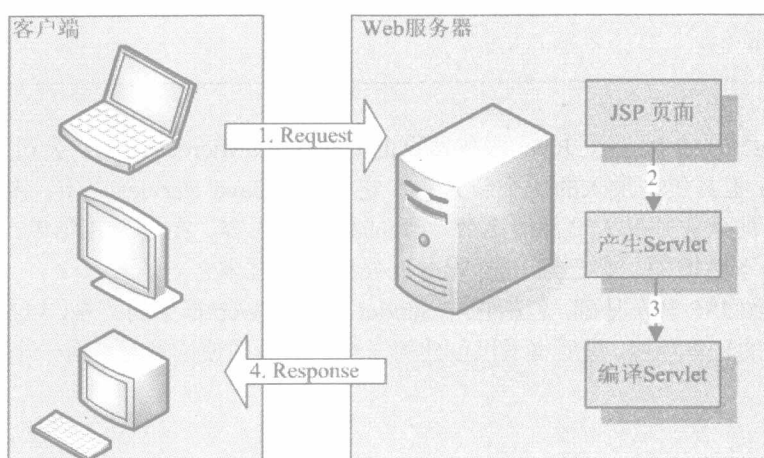


图 1-1 JSP 的执行过程

- (1) 客户端发出 Request (请求)。
- (2) JSP 容器 (服务器) 将 JSP 转译成 Servlet 的源代码。
- (3) 将产生的 Servlet 的源代码经过编译，并加载到内存执行。
- (4) 把结果 Response (响应) 至客户端。

JSP 除定义了自身特殊的元素外，还定义了许多用于 Web 应用程序的标准元素，如引用 JavaBeans 的元素及共享信息的元素。开发人员也可以通过使用应用程序的专用元素来扩展 JSP 语法，从而完成诸如访问数据库和 EJB、发送电子邮件，以及生成表示应用程序专用数据的 HTML 这样的任务。JSTL 规范 (与 JSP 规范相关的一种规范) 也定义了一组这种常用的自定义元素，标准元素和自定义元素的组合可以让开发人员开发出强大的 Web 应用程序。

1.1.2 JSP 和 Servlet 扮演的角色

Sun 公司首先发展出 Servlet，其功能强大且体系设计很完善。但其输出 HTML 语法时必须使用 `out.println()` 逐句输出，如下面一段简单的程序所示：

```

out.println("<html>");
out.println("<head><title>First Servlet</title></head>");
out.println(" Hello World <br>");
out.println("<body>");
out.println("您好");
out.println("</body>");
out.println("</html>")

```

由于这是一段简单的 Hello World 程序，还看不出其复杂性。但是当整个网页内容非常复杂时，Servlet 程序可能大部分都是用 `out.println()` 输出 HTML 的标签了！

后来 Sun 公司推出类似于 ASP 的嵌入型脚本语言并且命名为“JavaServer Pages”，简称为“JSP”，于是上面那段程序改为：

```

<html>
<head><title>www.csdn.cn -中国软件开发网 </title></head>
<body>
<%
    out.println(" Hello World <br>");
    out.println("您好");
%>
</body>
</html>

```

虽然表面上 JSP 代替了 Servlet 完成了请求响应，但实际上 JSP 仍然需要结合 Servlet 技术才能完成自己的任务。因为 JSP 页面在 Web 服务器端执行时首先转译成 Servlet 类，然后通过执行转译后的 Servlet 把响应结果送到客户端，读者可参考图 1-1。

可以说 JSP 注重于 Web 应用程序的前端视图，而 Servlet 则注重于 Web 应用程序的逻辑控制。现在大多数 Web 应用程序把这两种强大的技术结合起来，因为二者的结合是一个强大的工具，使用这种工具可以开发出易于维护且能够随着新的需求进行扩展的伸缩性强的应用程序。由于 Servlet 是一个普通的 Java 类，因此可以借用 Java 语言的全部能力，通过标准的 Java 开发和调试环境来实现请求处理。然后可以把 JSP 页面用在最适合它的地方，如前端视图中，使用 Servlet 收集或生成的信息把应答通过 JSP 显示。

自从 JSP 技术产生以来，将其与 Servlet 结合起来的 Web 开发框架有很多，如 JSP+Servlet+JavaBeans 模式、基于 JSP 和 Servlet 的 MVC（Model View Control，模型视图控制器）模式等。这些典型的框架技术目前都很受开发人员的青睐，在后面的章节中我们将会详细介绍。

1.1.3 JSP 和其他框架的集成

JSP 作为 J2EE 的一部分，既可以用于开发小型的 Web 站点，也可以用于开发大型企业级的应用程序。

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及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，就像使用 PHP 或 ASP 开发动态网页一样。

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，在这种站点中，不能

将所有的内容全部交给 JSP 页面来处理。在单纯的 JSP 中加入 Java Beans 技术将有助于这种中型网站的开发。利用 Java Beans 将很容易完成如数据库连接、用户登录与注销及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 Java Beans，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。

然而对于大中型网站来说，最复杂的就是如何把业务、逻辑控制和视图这几个方面完全清晰地区分开来，而不是由传统的 JSP+Java Beans 方式来实现。因为那样会让商业逻辑、数据库及视图控制方面很混乱，不易维护及扩展，并且可伸缩性差。这样就产生了用于区别商业逻辑、业务及视图控制的技术及其开发框架，典型的有 EJB、Struts、Hibernate 及 Spring 等框架技术。下面我们简单来看几个典型 JSP 和其他技术及框架相集成的技术。

(1) JSP+Java Beans+DataBase。

这个模型主要的特点就是 JSP 全权接收请求并发送响应，而 Java Beans 的作用是和数据库服务器进行通信，读取数据。然后将数据信息发送至 JSP 页面，如图 1-2 所示。

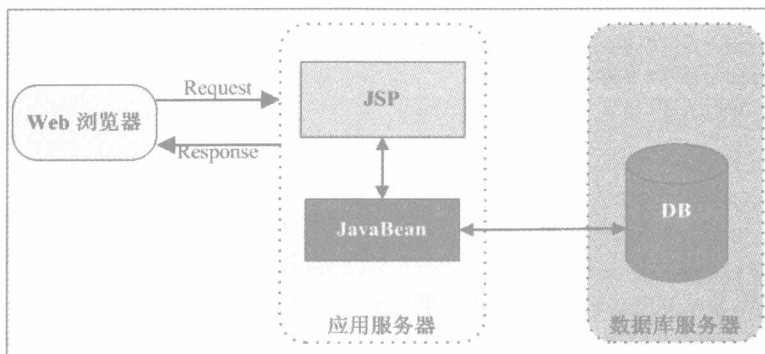


图 1-2 JSP+JavaBean 的框架实现

(2) JSP+JavaBeans+Servlet+DAO。

该模型集成使用了 Servlets 和 JSP 页面，如图 1-3 所示。其中 JSP 页面用于表示层，并且 Servlets 负责处理各类任务。Servlet 作为一个控制器，负责处理请求并创建 JSP 页面所需的任何 Bean，并确定将该请求传递到哪个 JSP 页面。JSP 页面检索 Servlet 创建的对象，并提取动态内容插入在一个模板中。

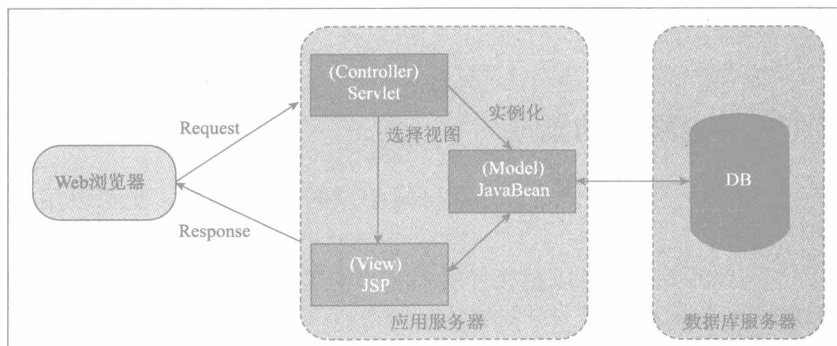


图 1-3 JSP+Servlet+JavaBean 的 MVC 框架实现

该模型促进了 MVC 体系结构风格设计模式的使用,早就存在多个框架能够实现 MVC 这个有用的设计模式,并将内容和表示真正地分离开来。Apache Struts 是 MVC 的形式化框架,非常适用于复杂的应用程序。在这些复杂的应用程序中,单个请求或表单提交会产生看起来截然不同的结果。

(3) JSP+Servlet+EJB。

EJB (Enterprise JavaBeans) 带给我们的是一种基本的事务管理和一种与客户端类型无关的组件模型,它加强了对模型 (Model)、视图 (View) 和控制器 (Controller) 的划分,从而使整个应用程序更易于扩展和维护。

EJB 组件有两种基本类型,即实体 Bean 和会话 Bean。实体 Bean 表示的是特定的一部分商业数据,例如,一个员工或一个客户的数据。每个实体 Bean 都有一个唯一的标识符,要访问这个实体的所有客户都将使用同一个 Bean 实例;而会话 Bean 主要用来处理商业逻辑,而且只能由创建它的那个客户使用。通常,会话 Bean 代表其客户操作实体 Bean。

在一个 EJB 应用程序的 Web 界面中,请求被发送给一个 Servlet,这与 JSP+Servlet 方案一样。该 Servlet 将不再处理这个请求,而是请求一个 EJB 会话 Bean (或者一个用到 EJB 会话 Bean 接口的 Web 层组件) 来完成对请求的处理。因此控制器角色跨越了 Servlet 和 EJB 会话 Bean,模型也可以跨越多个组件。通常我们使用 Web 层之间的通信,当完成对请求的处理后会话 Bean 可能更新了许多 EJB 实体 Bean。这时,Web 层中的 Java Beans 组件将会接到通知,从而自动地更新其状态。随后一个 JSP 页面将使用这些组件来生成一个应答,这样模型角色由 EJB 实体 Bean 和 Web 层中的 Java Beans 组件共同承担。

基于 EJB 的应用程序通常被看做是无所不能的,然而它是最复杂的模型,因此需要在开发、配置、操作和管理领域多做许多事情。尽管 EJB 可能是一种开发某种类型应用程序的比较好的方式,但它对许多人来说过于复杂了。因此在决定使用 EJB 之前,应仔细考虑到底是否真的需要 EJB。

(4) JSP+Struts+JavaBeans+DAO。

JSP 和 Struts 框架的集成实际上就是 MVC 框架 (如图 1-3 所示) 技术的延展,可用于快速开发 Java Web 应用。Struts 实现的重点在控制器,包括 ActionServlet/RequestProcessor,我们定制的 Action 也为视图提供了一系列定制标签 (Custom Tag)。但 Struts 几乎没有涉及模型,所以它可以采用 JAVA 实现的任何形式的商业逻辑。

Struts 本身负责将 MVC 中的用户数据传入业务层,并且将业务层处理的结果返回给用户。

(5) JSP+Struts+Hibernate。

Hibernate 是一个面向 Java 环境的对象/关系数据库映射工具,这种映射工具用来把对象模型表示的对象映射到基本 SQL 的关系模型数据结构中。

Hibernate 使用数据库和配置文件来为应用程序提供持久化服务,如图 1-4 所示为其概览。

Hibernate 不仅仅管理 Java 类到数据库表的映射 (包括 Java 数据类型到 SQL 数据类型的映射),还提供数据查询和获取数据的方法,可以大幅度减



图 1-4 Hibernate 概览