

计算机精品教材
Web开发工程师“零点起飞”系列图书



JSP Web

应用开发案例教程

主编 侯玉香 谭鸿健 郑旋



上海交通大学出版社
SHANGHAI JIAO TONG UNIVERSITY PRESS

计算机精品教材

Web 开发工程师“零点起飞”系列图书

JSP Web 应用开发 案例教程

主编 侯玉香 谭鸿健 郑 旋



上海交通大学出版社
SHANGHAI JIAO TONG UNIVERSITY PRESS

内容提要

全书共分 12 章, 内容涵盖: 初识 JSP、JSP 开发基础、JSP 的基础语法、JSP 中的内置对象、JavaBean、Java Servlet 基础、JSP 中的文件操作、MySQL 数据库基础、JDBC 的介绍与使用、JSP 的 XML 操作、MVC 模式和开发企业网站。

本书可作为各类院校及计算机教育培训机构专用教材, 也可作为程序爱好者的参考用书。

图书在版编目 (C I P) 数据

JSP Web 应用开发案例教程 / 侯玉香, 谭鸿健, 郑旋
主编. — 上海: 上海交通大学出版社, 2017 (2018 重印)
ISBN 978-7-313-17720-9

I. ①J… II. ①侯… ②谭… ③郑… III. ①JAVA 语言—网页制作工具—教材 IV. ①TP312②TP393.092

中国版本图书馆 CIP 数据核字 (2017) 第 183252 号

JSP Web 应用开发案例教程

主 编: 侯玉香 谭鸿健 郑 旋

出版发行: 上海交通大学出版社

地 址: 上海市番禺路 951 号

邮政编码: 200030

电 话: 021-64071208

出 版 人: 谈 毅

印 制: 北京市科星印刷有限责任公司

经 销: 全国新华书店

开 本: 787mm×1092mm 1/16

印 张: 19.25 字 数: 350 千字

版 次: 2017 年 9 月第 1 版

印 次: 2018 年 3 月第 2 次印刷

书 号: ISBN 978-7-313-17720-9/TP

定 价: 58.00 元

版权所有 侵权必究

告读者: 如发现本书有印装质量问题请与发行部联系

联系电话: 010-62137141



前言

JSP 全名为 Java Server Pages, 中文名为 java 服务器页面, 是由 Sun Microsystems 公司倡导、许多公司参与一起建立的一种动态网页技术标准。利用它可以创建安全、跨平台的动态网站。

本书是一本简单实用的教程, 不仅配备了大量实例, 且语言简洁、通俗易懂。书中所有实例都经过了精心的推敲和设计, 既有助于读者理解相关知识, 又具有启发性和实用性, 让读者边学边练, 学完即可上手操作。另外, 书中最后安排了一个综合实例, 以通俗易懂的语言和注释清晰的代码介绍了一个企业网站的开发, 让读者轻松做到学以致用。

本书特色

(1) **由浅入深, 循序渐进。**本书先从 JSP 开发基础学起, 然后由浅入深、循序渐进地学习其核心技术, 非常便于学生理解和掌握。

(2) **结构新颖, 轻松易学。**本书的大部分章节采用“知识点+小实例”的形式讲解, 在用通俗易懂的语言简单介绍知识点后, 紧接着安排了与当前知识点和实际应用相关的小实例, 从而使读者边学边练、学有所用。另外, 每章都安排有“本章总结”, 使学生学完各章后还能对所学知识和技能进行总结。

(3) **实例典型, 注释清晰。**书中的每一个实例都经过精心推敲和设计, 不仅能够准确反应相关知识点, 还能有效增强学生的实战能力。另外, 为方便读者阅读和学习程序代码, 我们为这些代码提供了详细的注释。

(4) **体例丰富, 形式活泼。**本书根据内容需要安排了很多“注意”和“知识库”, 可以让读者在学习过程中更轻松的理解相关知识点和概念。

(5) **提供精美的教学课件和源文件。**本书不仅提供精美的教学课件, 并且书中的每个实例都提供了源文件, 读者可直接拷贝到本地服务器上查看运行效果。

教学资源下载

读者可到网站 (www.bjjqe.com) 下载本书配套的教学课件、素材和实例源文件。如果读者在学习过程中有什么疑问, 也可登录该网站寻求帮助, 我们将会及时解答。

本书作者团队

本书由侯玉香、谭鸿健、郑旋任主编, 由李俊霞、楚志凯、谢艳芳、臧芳、邓春伟、王晖、汪普庆、陈爱国、闫红、生桂勇、肖丽萍、舒慧欣任副主编。

尽管我们在编写本书时已竭尽全力, 但书中疏漏及错误之处在所难免, 敬请广大读者批评指正。

编 者

2018 年 2 月

本书编委会

主 编：侯玉香 谭鸿健 郑 旋

副 主 编：李俊霞 楚志凯 谢艳芳

臧 芳 邓春伟 王 晖

汪普庆 陈爱国 闫 红

生桂勇 肖丽萍 舒慧欣



目录

第1章 初识 JSP	1
1.1 JSP 概述	2
1.1.1 什么是 JSP	2
1.1.2 Servlet	3
1.1.3 JSP 的运行原理	4
1.2 Windows 下 JSP 运行环境的搭建	5
1.2.1 安装与配置 JDK	5
1.2.2 安装与配置 Tomcat 服务器	9
1.3 JSP 开发工具	14
1.3.1 MyEclipse 简介	14
1.3.2 安装 MyEclipse	15
1.3.3 配置 MyEclipse	18
1.3.4 使用 MyEclipse 创建 JSP 项目	20
本章小结	25
思考与练习	25
第2章 JSP 开发基础	27
2.1 Java 基本程序设计	28
2.1.1 变量、常量与数据类型	28
2.1.2 流程控制语句	34
2.1.3 数组	37
2.1.4 面向对象程序设计	40
2.1.5 集合类的应用	43





2.1.6 异常处理语句	46
2.2 JavaScript 脚本语言	47
2.2.1 JavaScript 概述	47
2.2.2 在 JSP 中引入 JavaScript	48
2.2.3 JavaScript 的数据类型与运算符	48
2.2.4 JavaScript 流程控制语句	51
2.2.5 JavaScript 应用实例	52
本章小结	55
思考与练习	55
第 3 章 JSP 的基础语法	57
3.1 JSP 页面的基本结构	58
3.2 JSP 中的指令标记	60
3.2.1 page 指令	60
3.2.2 include 指令	63
3.3 Java 程序片	65
3.4 变量和方法的声明	66
3.5 Java 表达式	67
3.6 JSP 动作标记	69
3.6.1 include 动作标记	69
3.6.2 param 动作标记	70
3.6.3 forward 动作标记	72
3.6.4 plugin 动作标记	74
3.7 JSP 中的注释	75
本章小结	76
思考与练习	76
第 4 章 JSP 中的内置对象	77
4.1 request 对象	78
4.1.1 request 对象的常用方法	78
4.1.2 request 对象的应用	78
4.1.3 使用 request 对象获取用户提交的信息	80
4.2 response 对象	83
4.2.1 response 对象的常用方法	83
4.2.2 使用 response 对象动态响应 ContentType 属性	83



4.2.3 response 重定向	85
4.2.4 设置页面刷新频率	86
4.3 out 对象	87
4.4 application 对象	89
4.4.1 application 对象的常用方法	89
4.4.2 application 对象应用实例	89
4.5 session 对象	91
4.5.1 session 对象的常用方法	91
4.5.2 session 对象应用实例	92
本章小结	95
思考与练习	95
第5章 JavaBean	97
5.1 JavaBean 基础知识	98
5.1.1 什么是 JavaBean	98
5.1.2 JavaBean 的特点	98
5.2 JSP 与 JavaBean 结合	102
5.2.1 JSP 中与 JavaBean 相关的动作标记	102
5.2.2 调用 JavaBean	103
5.2.3 获取和修改 bean 的属性	104
5.2.4 JavaBean 的生命周期	105
5.3 使用 JavaBean 处理表单数据	106
5.3.1 使用 JavaBean 处理表单数据	107
5.3.2 设置中文编码	110
5.3.3 Post 与 Get	111
5.3.4 使用 JavaBean 制作简单用户注册页面	111
本章小结	114
思考与练习	114
第6章 Java Servlet 基础	117
6.1 什么是 Servlet	118
6.2 Servlet 的工作原理	119
6.2.1 Servlet 对象的生命周期	119
6.2.2 Servlet 的 init()方法	119
6.2.3 Servlet 的 service()方法	119





6.2.4 Servlet 的 destroy()方法	120
6.3 Servlet 的创建、部署与运行	120
6.3.1 Servlet 源文件和字节码文件	120
6.3.2 编写部署文件 web.xml	121
6.3.3 创建与运行 Servlet	122
6.4 通过 JSP 页面访问 Servlet	124
6.5 doGet()与 doPost()方法	129
6.6 重定向与转发	132
6.6.1 重定向	132
6.6.2 转发	133
本章小结	136
思考与练习	137
第 7 章 JSP 中的文件操作	139
7.1 File 类	140
7.2 I/O 流	142
7.2.1 字节输入流	143
7.2.2 字节输出流	145
7.2.3 字符流	147
7.2.4 缓冲流	147
7.3 文件上传和下载	149
7.3.1 文件上传	149
7.3.2 文件下载	154
本章小结	157
思考与练习	157
第 8 章 MySQL 数据库基础	159
8.1 MySQL 数据库介绍	160
8.2 在 Windows 上安装和配置 MySQL	160
8.2.1 下载 MySQL	160
8.2.2 安装与配置 MySQL	162
8.3 MySQL 服务的基本操作	166
8.3.1 启动 MySQL 服务	166
8.3.2 登录和退出 MySQL 服务	167
8.3.3 停止 MySQL 服务	169

8.4 MySQL 基本操作	169
8.4.1 数据库和数据表基本操作	169
8.4.2 数据记录基本操作	173
8.5 MySQL 图形化管理工具	175
8.5.1 Navicat for MySQL 简介	175
8.5.2 Navicat for MySQL 基本操作	176
8.5.3 使用 Navicat for MySQL 备份与恢复数据库	179
8.5.4 使用 Navicat for MySQL 创建数据表	181
本章小结	183
思考与练习	183
第 9 章 JDBC 的介绍与使用	185
9.1 JDBC 概述	186
9.1.1 JDBC 的作用	186
9.1.2 JDBC 的体系结构	187
9.1.3 JDBC 的驱动器类型	188
9.2 JDBC 程序访问数据库的过程	188
9.2.1 加载 JDBC 数据库驱动程序	189
9.2.2 创建数据库连接	190
9.2.3 执行 SQL 语句	191
9.2.4 处理结果	191
9.2.5 关闭 JDBC 对象	191
9.3 通过 Statement 执行 SQL 语句	193
9.3.1 ResultSet	194
9.3.2 executeQuery()方法	195
9.3.3 executeUpdate()方法	197
9.4 PreparedStatement	198
9.4.1 预编译功能	199
9.4.2 使用通配符	199
9.5 JDBC 事务控制	204
9.6 使用 JDBC 连接不同数据库	207
9.6.1 连接 Oracle 数据库	208
9.6.2 连接 SQL Server 数据库	208
9.6.3 连接 Access 数据库	208



本章小结	209
思考与练习	209
第 10 章 JSP 的 XML 操作	211
10.1 XML 文件的基本结构	212
10.2 DOM 解析器	216
10.2.1 应用 DOM 解析器的基本步骤	216
10.2.2 DOM 解析的原理	216
10.2.3 Document 对象	217
10.2.4 DOM 解析器应用实例	219
10.3 SAX 解析器	224
10.3.1 应用 SAX 解析器的基本步骤	224
10.3.2 SAX 解析的原理	224
10.3.3 SAX 解析器应用实例	226
本章小结	232
思考与练习	233
第 11 章 MVC 模式	235
11.1 MVC 模式简介	236
11.2 JSP 中的 MVC 模式	236
11.3 MVC 模式应用实例	237
11.3.1 使用 MVC 模式制作简易登录模块	237
11.3.2 使用 MVC 模式制作查询模块	242
本章小结	248
思考与练习	248
第 12 章 开发企业网站	249
12.1 开发背景	250
12.2 需求分析	250
12.3 系统功能结构	250
12.4 系统功能概览	251
12.5 数据库设计	253
12.5.1 数据库概念设计	253
12.5.2 数据库物理结构设计	254



12.6 系统实现	257
12.6.1 模块公用类	257
12.6.2 JavaBean 和 Dao	260
12.6.3 Servlet	266
12.6.4 后台页面的实现	270
12.6.5 前台页面的实现	283
参考文献	292





Servlet 的全称为 Servlet，是运行在服务器端的 Java 小程序，是运行在服务器端的程序。Servlet 的主要功能是接收客户端的请求，并返回响应。Servlet 的具体实现过程如下：

- (1) 客户端发送请求至服务器端。
- (2) 服务器将请求信息转发给 Servlet。
- (3) Servlet 生成响应内容并将其传回服务器。
- (4) 服务器将响应内容返回给客户端。

1.1 JSP 概述

1.1.1 什么是 JSP

JSP 全名为 Java Server Pages, 中文名为 Java 服务器页面, 是由 Sun Microsystems 公司倡导, 许多公司参与一起建立的一种动态网页技术标准。JSP 技术有点类似 ASP 技术。JSP 页面是在传统的网页 HTML 文件 (*.htm, *.html) 中插入 Java 程序段 (Scriptlet) 和 JSP 标记 (tag) (放在<% %>中) 而形成的文件。

知识库

HTML 即超文本标记语言, 其结构包括“头”部分和“主体”部分。其中, “头”部提供了关于网页的信息, “主体”部分提供了网页的具体内容。

ASP 即 Active Server Pages, 它是 Microsoft 公司开发的服务器端脚本环境, 可用来创建动态交互式网页, 并建立强大的 Web 应用程序。

JSP 页面按文本文件保存, 扩展名为.jsp。如果使用文本编辑器“记事本”编辑 JSP 页面, 在保存页面时必须在“保存类型”下拉列表中选择“所有文件”, 在“编码”下拉列表中选择“ANSI”, 并在文件名后加上扩展名.jsp, 如图 1-1 所示。



图 1-1 保存 JSP 文件

注意

在保存 JSP 页面时, 文件名必须符合标识符规定, 文件名可以由字母、数字、下划线和美元符号组成, 且首字符不能是数字。另外要注意, JSP 技术基于 Java 语言, 名字区分大小写。



下面为 1-1.jsp 的源代码，其运行结果如图 1-2 所示。

```
<%@ page language="java" import="java.util.*" pageEncoding="ISO-8859-1"%>
<html>
<head>
<title>myjsp</title>
</head>
<body>
This is my JSP page. <br>
</body>
</html>
```

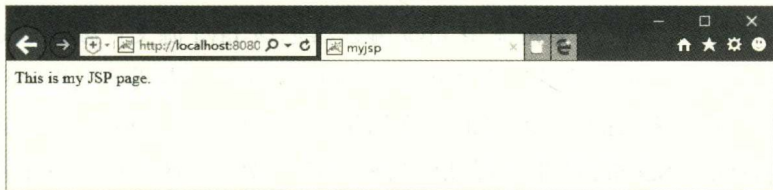


图 1-2 index.jsp 运行结果

用 JSP 开发的 Web 应用是跨平台的，既能在 Linux 下运行，也能在 Windows 等其他操作系统下运行。

1.1.2 Servlet

Servlet 的全称为 Server Applet，又称 Java Servlet，是用 Java 语言编写的服务器端程序。

Servlet 的主要功能是可交互式地浏览和修改数据，生成动态 Web 内容。具体实现过程如下：

- (1) 客户端发送请求至服务器端；
- (2) 服务器将请求信息发送至 Servlet；
- (3) Servlet 生成响应内容并将其传给服务器；
- (4) 服务器将响应内容返回给客户端。

Servlet 运行于支持 Java 的应用服务器中，一个 Servlet 就是 Java 编程语言中的一个类，它被用来扩展服务器的性能，服务器上驻留着可以通过“请求—响应”编程模型来访问的应用程序。虽然 Servlet 可以对任何类型的请求产生响应，但绝大多数情况下它只用于扩展基于 HTTP 协议的 Web 服务器。

在通信量大的服务器上，Java Servlet 的优点在于它的执行速度要快于 CGI 程序。各个

用户请求被激活成单个程序中的一个线程，而无需创建单独的进程，这意味着服务器端处理请求的系统开销将明显降低。

最早支持 Servlet 标准的是 JavaSoft 的 Java Web Server。此后，一些其他的基于 Java 的 Web 服务器也开始支持标准的 Servlet。

1.1.3 JSP 的运行原理

当一个 JSP 文件第一次被请求时，JSP 的运行过程如下：

(1) 将 JSP 页面转换成一个 Java 文件（又称 Servlet），同时也是一个完整的 Java 程序。如果在转换过程中发现 JSP 文件有语法错误，转换过程将中断，并向服务端和客户端输出出错信息。

(2) 转换成功后，JSP 引擎将调用 Java 编译器把该 Java 源文件编译成.class 文件（字节码文件）。该文件为二进制形式，但不能被系统直接执行，需要虚拟机解释执行。

(3) 创建一个该 Servlet（JSP 页面转换结果）的实例。该 Servlet 的 Init()方法将被执行，不过 Init()方法在 Servlet 的生命周期中只被执行一次，因此，可在 Init()中进行一些初始化工作，如建立网络连接、建立数据库连接等。

(4) 调用 Service()方法来处理客户端的请求。对于每一个新的请求，JSP 引擎都将创建一个新的线程来处理该请求。如果有多个客户端同时请求该 JSP 文件，则 JSP 引擎会创建多个线程。每个客户端请求对应一个线程。以多线程方式执行可以大大降低对系统的资源需求，提高系统的并发量及响应时间。Service()方法执行后会将 HTML 内容返回给客户端。

(5) 如果 JSP 文件被修改了，服务器将重新进行编译，并将编译结果取代内存中的 Servlet，并继续上述处理过程。

虽然 JSP 效率很高，但由于第一次调用时需要转换和编译而有一些轻微的延迟。另外，如果系统资源不足，JSP 引擎将会把 Servlet 从内存中移除。当这种情况发生时，Destroy()方法首先被调用，然后 Servlet 实例便被标记加入“垃圾收集”处理。一般在 Destroy()中释放相应的资源。

注意

如果对 JSP 页面进行了修改，那么 Tomcat 服务器会生成新的字节码文件。