



JWorld@TW技术论坛版主
Java权威技术顾问与专业讲师

全新改版!

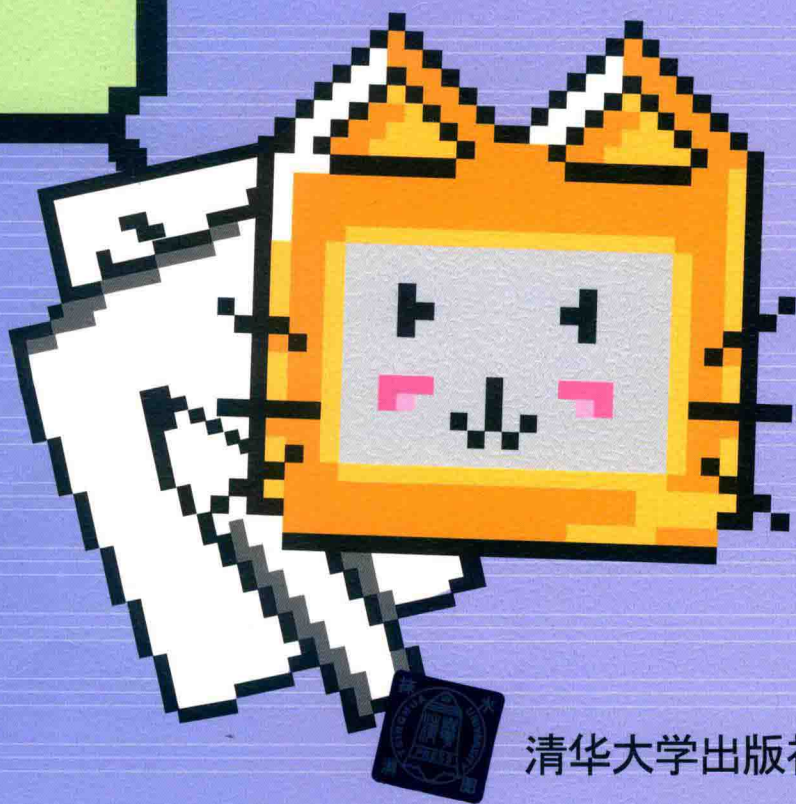
JSP & Servlet

学习笔记(第3版)

——从Servlet到Spring Boot

林信良 著

- 涵盖SCWCD考试范围
- Servlet 4.0新功能介绍
- 加强Web安全基本观念
- 衔接Spring MVC、Spring Boot
- 提供Lab操作文档



碁峯
www.gotop.com.tw

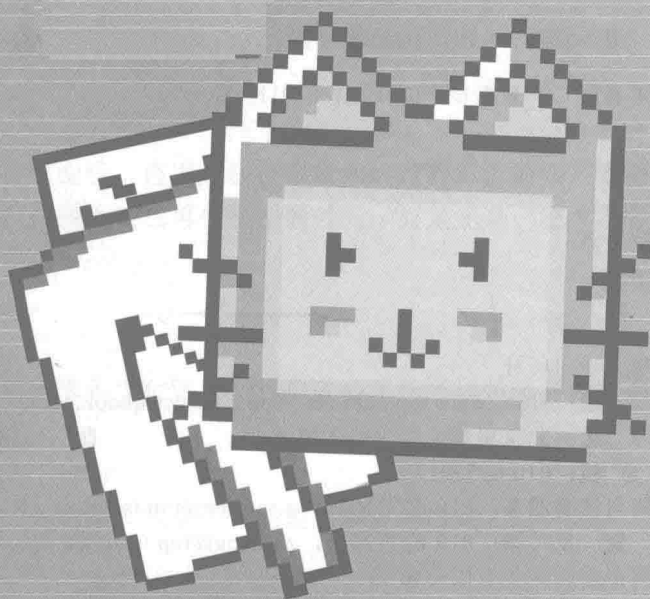
清华大学出版社

JSP & Servlet

学习笔记(第3版)

——从Servlet到Spring Boot

林信良 著



清华大学出版社

北 京

内 容 简 介

本书是作者多年来教学实践经验的总结,汇集了学员在学习 JSP & Servlet 或认证考试时遇到的概念、操作、应用等各种问题及解决方案。

本书基于 Servlet 4.0/Java SE 8 重新改版,无论章节架构还是范例程序代码,都做了全面更新。书中详细介绍了 OWASP TOP10、CWE、CVE,讨论了会话安全、密码管理、Java EE 安全机制、CSRF 等 Web 安全基本概念,增加了对 Spring、Spring MVC、Spring Boot 的入门介绍,认识 Web MVC 框架与快速开发工具的使用。本书还涵盖了文本处理、图片验证、自动登录、验证过滤器、压缩处理、线上文件管理、邮件发送等实用范例。

本书在讲解过程中,以“微博”项目贯穿全书,将 JSP & Servlet 技术应用于实际项目开发之中,并使用重构方式来改进应用程序架构。

本书适合 JSP & Servlet 初学者以及广大 JSP & Servlet 技术应用人员。

北京市版权局著作权合同登记号 图字:01-2018-4946

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 & Servlet 学习笔记:从 Servlet 到 Spring Boot:第3版 / 林信良 著. —北京:清华大学出版社, 2019

ISBN 978-7-302-52245-4

I. ①J... II. ①林... III. ①JAVA 语言—程序设计 IV. ①TP312.8

中国版本图书馆 CIP 数据核字(2019)第 020117 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:牛艳敏

责任印制:沈 露

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:三河市君旺印务有限公司

经 销:全国新华书店

开 本:185mm×260mm

印 张:30.25

字 数:755 千字

版 次:2010 年 4 月第 1 版 2019 年 4 月第 3 版

印 次:2019 年 4 月第 1 次印刷

定 价:98.00 元

产品编号:080994-01

序

时间可以是最大的敌人，也可以是最强的盟友，然而多数人选择了前者。

2009/2/21

人生在世，要赚到钱也要赚到时间。

2010/7/21

金钱投资得宜可以赢得更多金钱，时间投资得当可以赢得更多时间，后者远比前者重要，但前者往往比后者得到更多的关注。

2012/2/18

只要每日认真投资(时间)，有时需要的只是等待。

2012/3/17

存起来的钱用掉就没了，存起来的时间却可反复使用。

2012/3/18

人啊！多半用时间换取金钱，再用金钱换取时间，过程遗留的残烬叫等待；有些人的时间越来越难换取金钱；有些人的钱再也换不到时间，而是无奈、空虚、怨念、固执……或者是无法再复燃的灰烬……

2013/4/8

人们多关心工作可以带来多少金钱，却鲜少关心工作可以带来多少时间！

2013/4/28

人的一生中两个最大的财富其实是：你的健康和你的时间。

2013/9/24

花费的时间成本是用来获得时间回馈，不是金钱回馈，不能混为一谈。

2013/10/18

时间不是金钱，时间就是时间，时间要值回更多的时间。

2014/6/16

有些事，现在不付出时间，将来需要付出更多时间。

2015/11/29

其实时间并不是一个定量，除了变少还可能变多。

2016/10/9

致投资时间在此的你！

2018/3/18

导 读

这份导读可以让你更了解如何使用本书。

字型

本书内文中与程序代码相关的文字，都用固定宽度字体来加以呈现，以与一般名词作区别。例如，JSP 是一般名词，而 `HttpServlet` 为程序代码相关文字，使用了固定宽度字体。

新旧版差异

本书是从《JSP & Servlet 学习笔记(第 2 版)》改版而来的，因此这里说明一下与《JSP & Servlet 学习笔记(第 2 版)》之间的差异。

就目录上可以看出的主要差异是，删除了《JSP & Servlet 学习笔记(第 2 版)》第 12 章“从模式到框架”，并由新撰写的 3 个 Spring 相关章节取代，这是为了从实际的框架中学习，而不是空谈概念；然而，Spring 那些章节并不是作为全面探讨 Spring 之用，而是作为一个衔接，希望从实际的应用程序重构中筛选出对应用程序有益的框架特性，以便逐步掌握框架的本质。

当然，照例要谈一些 Java EE 8 的功能，相关讨论会放在各章节中适当的地方。由于《JSP & Servlet 学习笔记(第 2 版)》是基于 Java EE 6，为了便于查找 Java EE 7/8 的功能介绍，如果发现页左侧有如  图示，就表示提及 Java EE 7 或 Java EE 8 功能，本书还提供了 Java EE 7/8 功能快速查询目录。

各章节的范例都做了全面改写，由于 Java EE 8 是基于 Java SE 8，范例程序代码会适当使用 Java SE 8 的特性，例如 Lambda 与 Stream API 等。

时至今日，撰写应用程序时必须要有相关的安全防护概念，作为一本谈论 Web 应用程序的书，适时地提及安全概念是必要的，书中谈到了 OWASP TOP 10，讨论了 Session 防护、注入攻击、Cookie 安全、密码加盐哈希、跨域伪造请求(Cross-Site Request Forgery, CSRF)等安全基本观念，并在适当的地方介绍了 OWASP Java Encoder、Java HTML Sanitizer 等项目的使用。

程序范例

本书大多数范例使用完整的程序实作来展现，如果是用以下方式示范程序代码：



FirstServlet Hello.java

```
package cc.openhome;

import java.io.IOException;
import java.io.PrintWriter;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

@WebServlet("/hello")
public class Hello extends HttpServlet { ← ❶ 继承 HttpServlet
    @Override
    protected void doGet( ← ❷ 重新定义 doGet()
        HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html; charset=UTF-8"); ← ❸ 设置响应内容类型

        String name = request.getParameter("name"); ← ❹ 取得请求参数

        PrintWriter out = response.getWriter(); ← ❺ 取得响应输出对象
        out.print("<!DOCTYPE html>");
        out.print("<html>");
        out.print("<head>");
        out.print("<title>Hello</title>");
        out.print("</head>");
        out.print("<body>");
        out.printf("<h1> Hello! %s!\n</h1>", name); ← ❻ 跟用户说 Hello!
        out.print("</body>");
        out.print("</html>");
    }
}
```

范例开始的左边名称为 FirstServlet，表示可以在范例文件的 samples 文件夹中查找相应章节目录，即可找到对应的 FirstServlet 项目，而右边名称为 Hello.java，表示可以在项目找到 Hello.java 文件。如果程序代码中出现标号与提示文字，表示后续的内文中会有对应于标号及提示的更详细说明。

原则上，建议每个项目范例都亲自动手撰写，如果由于教学时间或实现时间上的限制，本书有建议进行的练习。在范例开始前有  图示的，表示建议动手实践，而且在范例文件的 labs 文件夹中有练习项目的基础内容，可以在导入项目后，完成项目中遗漏或必须补齐的程序代码或设置。

如果文中使用以下程序代码，则表示它是一个完整的程序内容，但不是项目的一部分，主要用来展现如何撰写一个完整的文件。

```
<%@page import="java.time.LocalDateTime"%>
<%@page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>JSP 范例文件</title>
```

```
</head>
<body>
<!-- 这里会依 Web 网站的时间而产生不同的响应 -->
<%= LocalDateTime.now() %>
</body>
</html>
```

如果使用以下程序代码，则表示它是一个代码段，主要展现程序撰写时需要特别注意的片段。

```
// 略 ...
public void _jspService(HttpServletRequest request,
                        HttpServletResponse response)
    throws java.io.IOException, ServletException {
    // 略...
    try {
        response.setContentType("text/html;charset=UTF-8");
        //略...
        out = pageContext.getOut();
        // 略...
    } catch (Throwable t) {
        // 略 ...
    } finally {
        // 略 ...
    }
}
```

操作步骤

本书将 IDE 设定的相关操作步骤，也作为练习的一部分，你会看到如下的操作步骤说明：

- (1) 运行 eclipse 文件夹中的 eclipse.exe。
- (2) 出现 Eclipse Launcher 对话框时，将 Workspace 设定为 C:\workspace，单击 Launch 按钮。
- (3) 执行菜单 Window|Preferences 命令，在弹出的 Preferences 对话框中，展开左边的 Server 节点，选择其中的 Runtime Environment 节点。
- (4) 单击右边 Server Runtime Environments 中的 Add 按钮，在弹出的 New Server Runtime Environment 对话框中选择 Apache Tomcat v9.0，单击 Next 按钮。
- (5) 单击 Tomcat installation directory 旁的 Browse 按钮，选取 C:\workspace 中解压缩的 Tomcat 文件夹，单击“确定”按钮。

提示框

在本书中会出现以下提示框：

提示 >>> 针对课程中提到的观点，提供一些额外的资源或思考方向，暂时忽略这些提示对课程进行的影响，但有时间的话，针对这些提示多作阅读、思考或讨论是有帮助的。

注意 >>> 针对课程中提到的观点，以提示框方式特别呈现出必须注意的一些使用方式、陷阱或避开问题的方法，看到这个提示框时请集中精力阅读。

综合练习

本书以“微博”项目的实现过程贯穿全书，随着每一章的进行，会在适当的时候将新介绍的技术应用至“微博”程序之中并作适当的修改，以了解完整的应用程序基本上是如何建构出来。

附录

本书配套资源中的范例文件包括本书全部范例，提供 Eclipse 范例项目，部分范例是 Gradle 项目，附录 A 说明如何使用这些范例项目。本书也说明如何在 Web 应用程序中整合数据库，实现范例时使用的数据库为 H2，使用方式可见 9.1 节的内容，范例若包含 H2 数据库文件*.mv.db 的话，联机时的名称与密码都是 caterpillar 与 12345678。

联系作者

若有与本书相关的勘误反馈等问题，可通过网站与作者联系：

<http://openhome.cc>

资源下载

本书配套资源下载：



目 录

Chapter 1 Web 应用程序简介	1
1.1 Web 应用程序基础	2
1.1.1 关于 HTML	2
1.1.2 URL、URN 与 URI	3
1.1.3 关于 HTTP	5
1.1.4 HTTP 请求方法	6
1.1.5 有关 URI 编码	9
1.1.6 后端与前端	11
1.1.7 Web 安全概念	13
1.2 Servlet/JSP 简介	14
1.2.1 何谓 Web 容器	14
1.2.2 Servlet 与 JSP 的关系	16
1.2.3 关于 MVC/Model 2	19
1.2.4 Java EE 简介	22
1.3 重点复习	23
Chapter 2 编写与设置 Servlet	24
2.1 第一个 Servlet	25
2.1.1 准备开发环境	25
2.1.2 第一个 Servlet 程序	27
2.2 在 Hello 之后	29
2.2.1 关于 HttpServlet	30
2.2.2 使用 @WebServlet	32
2.2.3 使用 web.xml	33
2.2.4 文件组织与部署	36
2.3 进阶部署设置	37
2.3.1 URL 模式设置	37
2.3.2 Web 文件夹结构	40
2.3.3 使用 web-fragment.xml	41
2.4 重点复习	44
2.5 课后练习	45

Chapter 3 请求与响应	46
3.1 从容器到 HttpServlet	47
3.1.1 Web 容器做了什么	47
3.1.2 doXXX()方法	49
3.2 关于 HttpServletRequest	52
3.2.1 处理请求参数	52
3.2.2 处理请求标头	55
3.2.3 请求参数编码处理	56
3.2.4 getReader()、getInputStream() 读取内容	58
3.2.5 getPart()、getParts()取得 上传文件	62
3.2.6 使用 RequestDispatcher 调派 请求	67
3.3 关于 HttpServletResponse	73
3.3.1 设置响应标头、缓冲区	73
3.3.2 使用 getWriter()输出字符	75
3.3.3 使用 getOutputStream()输出 二进制字符	78
3.3.4 使用 sendRedirect()、 sendError()	80
3.4 综合练习	81
3.4.1 微博应用程序功能概述	82
3.4.2 实现会员注册功能	83
3.4.3 实现会员登录功能	88
3.5 重点复习	89
3.6 课后练习	90
Chapter 4 会话管理	92
4.1 会话管理基本原理	93
4.1.1 使用隐藏域	93
4.1.2 使用 Cookie	96
4.1.3 使用 URI 重写	100

4.2	HttpSession 会话管理	102
4.2.1	使用 HttpSession	103
4.2.2	HttpSession 会话管理 原理	107
4.2.3	HttpSession 与 URI 重写	109
4.3	综合练习	111
4.3.1	登录与注销	111
4.3.2	会员信息管理	112
4.3.3	新增与删除信息	116
4.4	重点复习	118
4.5	课后练习	119
Chapter 5 Servlet 进阶 API、过滤器与监听器 120		
5.1	Servlet 进阶 API	121
5.1.1	Servlet、ServletConfig 与 GenericServlet	121
5.1.2	使用 ServletConfig	123
5.1.3	使用 ServletContext	126
5.1.4	使用 PushBuilder	128
5.2	应用程序事件、监听器	130
5.2.1	ServletContext 事件、 监听器	130
5.2.2	HttpSession 事件、 监听器	135
5.2.3	HttpServletRequest 事件、 监听器	141
5.3	过滤器	142
5.3.1	过滤器的概念	142
5.3.2	实现与设置过滤器	144
5.3.3	请求封装器	149
5.3.4	响应封装器	153
5.4	异步处理	157
5.4.1	AsyncContext 简介	158
5.4.2	异步 Long Polling	160
5.4.3	更多 AsyncContext 细节	163

5.4.4	异步 Server-Sent Event	164
5.4.5	使用 ReadListener	167
5.4.6	使用 WriteListener	169
5.5	综合练习	172
5.5.1	创建 UserService	172
5.5.2	设置过滤器	177
5.5.3	重构微博	179
5.6	重点复习	183
5.7	课后练习	185
Chapter 6 使用 JSP 186		
6.1	从 JSP 到 Servlet	187
6.1.1	JSP 生命周期	187
6.1.2	Servlet 至 JSP 的简单 转换	191
6.1.3	指示元素	194
6.1.4	声明、Scriptlet 与表达式 元素	197
6.1.5	注释元素	201
6.1.6	隐式对象	201
6.1.7	错误处理	204
6.2	标准标签	208
6.2.1	<jsp:include>、<jsp:forward> 标签	208
6.2.2	<jsp:useBean>、 <jsp:setProperty>与 <jsp:getProperty>简介	209
6.2.3	深入<jsp:useBean>、 <jsp:setProperty>与 <jsp:getProperty>	211
6.2.4	谈谈 Model 1	214
6.2.5	XML 格式标签	216
6.3	表达式语言(EL)	217
6.3.1	EL 简介	218
6.3.2	使用 EL 取得属性	220
6.3.3	EL 隐式对象	222

6.3.4	EL 运算符	223
6.3.5	自定义 EL 函数	224
6.3.6	EL 3.0	226
6.4	综合练习	227
6.4.1	改用 JSP 实现视图	228
6.4.2	重构 UserService 与 member.jsp	231
6.4.3	创建 register.jsp、index.jsp、 user.jsp	234
6.5	重点复习	242
6.6	课后练习	243
Chapter 7	使用 JSTL	244
7.1	JSTL 简介	245
7.2	核心标签库	246
7.2.1	流程处理标签	246
7.2.2	错误处理标签	249
7.2.3	网页导入、重定向、URI 处理标签	250
7.2.4	属性处理与输出标签	252
7.3	II8N 兼容格式标签库	254
7.3.1	II8N 基础	254
7.3.2	信息标签	257
7.3.3	地区标签	259
7.3.4	格式标签	264
7.4	XML 标签库	267
7.4.1	XPath、XSLT 基础	267
7.4.2	解析、设置与输出标签	270
7.4.3	流程处理标签	271
7.4.4	文件转换标签	272
7.5	函数标签库	274
7.6	综合练习	275
7.6.1	修改 index.jsp、 register.jsp	275
7.6.2	修改 member.jsp	277
7.6.3	修改 user.jsp	278

7.7	重点复习	278
7.8	课后练习	280
Chapter 8	自定义标签	281
8.1	Tag File 自定义标签	282
8.1.1	Tag File 简介	282
8.1.2	处理标签属性与 Body	285
8.1.3	TLD 文件	287
8.2	Simple Tag 自定义标签	288
8.2.1	Simple Tag 简介	288
8.2.2	了解 API 架构与生命 周期	290
8.2.3	处理标签属性与 Body	293
8.2.4	与父标签沟通	296
8.2.5	TLD 文件	300
8.3	Tag 自定义标签	301
8.3.1	Tag 简介	301
8.3.2	了解架构与生命周期	302
8.3.3	重复执行标签 Body	304
8.3.4	处理 Body 运行结果	306
8.3.5	与父标签沟通	309
8.4	综合练习	311
8.4.1	重构/使用 DAO	312
8.4.2	加强 user.jsp	315
8.5	重点复习	317
8.6	课后练习	319
Chapter 9	整合数据库	320
9.1	JDBC 入门	321
9.1.1	JDBC 简介	321
9.1.2	连接数据库	327
9.1.3	使用 Statement、 ResultSet	331
9.1.4	使用 PreparedStatement、 CallableStatement	335
9.2	JDBC 进阶	338

9.2.1 使用 DataSource 取得 连接	338
9.2.2 使用 ResultSet 卷动、更新 数据	341
9.2.3 批次更新	343
9.2.4 Blob 与 Clob	344
9.2.5 事务简介	350
9.2.6 metadata 简介	356
9.2.7 RowSet 简介	358
9.3 使用 SQL 标签库	363
9.3.1 数据源、查询标签	363
9.3.2 更新、参数、事务标签	364
9.4 综合练习	366
9.4.1 使用 JDBC 实现 DAO	366
9.4.2 设置 JNDI 部署描述	369
9.4.3 实现首页最新信息	370
9.5 重点复习	374
9.6 课后练习	375
Chapter 10 Web 容器安全管理	376
10.1 了解与实现 Web 容器安全 管理	377
10.1.1 Java EE 安全基本 概念	377
10.1.2 声明式基本身份验证	379
10.1.3 容器基本身份验证 原理	384
10.1.4 声明式窗体验证	385
10.1.5 容器窗体验证原理	386
10.1.6 使用 HTTPS 保护 数据	387
10.1.7 程式化安全管理	389
10.1.8 标注访问控制	391
10.2 综合练习	393
10.2.1 使用容器窗体验证	393
10.2.2 设置 DataSource- Realm	395
10.3 重点复习	396
10.4 课后练习	397
Chapter 11 JavaMail 入门	398
11.1 使用 JavaMail	399
11.1.1 发送纯文字邮件	399
11.1.2 发送多重内容邮件	401
11.2 综合练习	405
11.2.1 发送验证账号邮件	405
11.2.2 验证用户账号	411
11.2.3 发送重设密码邮件	412
11.2.4 重新设置密码	415
11.3 重点复习	418
11.4 课后练习	419
Chapter 12 Spring 起步走	420
12.1 使用 Gradle	421
12.1.1 下载和设置 Gradle	421
12.1.2 简单的 Gradle 项目	422
12.1.3 Gradle 与 Eclipse	423
12.2 认识 Spring 核心	425
12.2.1 相依注入	425
12.2.2 使用 Spring 核心	427
12.3 重点复习	430
12.4 课后练习	430
Chapter 13 整合 Spring MVC	431
13.1 初识 Spring MVC	432
13.1.1 链接库或框架	432
13.1.2 初步套用 Spring MVC	433
13.1.3 注入服务对象与 属性	440
13.2 逐步善用 Spring MVC	444
13.2.1 简化控制器	444
13.2.2 建立窗体对象	449
13.2.3 关于 Thymeleaf 模板	452

13.3 重点复习	455
13.4 课后练习	456
Chapter 14 简介 Spring Boot	457
14.1 初识 Spring Boot	458
14.1.1 哈喽! Spring Boot!	458
14.1.2 实现 MVC	461
14.1.3 使用 JSP	464
14.2 整合 IDE	465

14.2.1 导入 Spring Boot 项目	465
14.2.2 Spring Tool Suite	466
14.3 重点复习	467
14.4 课后练习	468
Appendix A 如何使用本书项目	469
A.1 项目环境配置	470
A.2 范例项目导入	470

Java EE 7/8 新功能索引

web.xml 版本变动	33
web.xml 新增<default-context-path>	34
HttpServletRequest 新增 getHttpServletMapping()	39
web.xml 新增<request-character-encoding>	57
Part 新增 getSubmittedFileName()	64
web.xml 新增<response-character-encoding>	76
HttpServletRequest 新增 changeSessionId()	104
ServletContext 新增 setSessionTimeout()	109
新增 PushBuilder	128
新增 HttpSessionIdListener	141
新增了 GenericFilter、HttpFilter 类别	145
ServletInputStream 非阻塞输入	168
ServletOutputStream 非阻塞输出	170
Expression Language 3.0	226

Web 应用程序简介

Chapter

1

学习目标:

- 认识 HTTP 基本特性
- 了解 GET、POST 使用时机
- 了解何为 URL/URI 编码
- 认识 Web 容器角色
- 了解 Servlet 与 JSP 的关系
- 认识 MVC/Model 2

1.1 Web 应用程序基础

在正式学习 Servlet/JSP 相关技术之前,要先花点时间了解一些 Web 应用程序基础知识,虽然是基础知识,却很重要。在我这几年的教学中,发现有些学员并不具备这些基础知识,或者忽略了这些基础知识中的一些细节,如 HTML(HyperText Markup Language)、HTTP(HyperText Transfer Protocol)、URI(Uniform Resource Identifier)甚至文字编码的问题等。

当然,谈这些内容并不是要你成为这几个名词的专家,而是在以后学习 Servlet/JSP 相关技术时,若有这些基础知识,就能真正理解相关技术背后的原理,而不会沦落到死背 API(Application Programming Interface)的窘境。

1.1.1 关于 HTML

本书介绍的 Web 应用程序,是由客户端(Client)与服务器端(Server)两个部分组成的,客户端基本是浏览器,服务器端是指 HTTP 服务器及运行在其上的相关资源,后面会使用浏览器来作为客户端代表,以 Web 网站来代称 HTTP 服务器及运行在其上的相关资源。

浏览器会请求 Web 网站,就本书来说,Web 网站必须产生 HTML,HTML 是以标签的方式来定义文件结构的。下面是一个简单的 HTML 范例:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>HTML 范例文件</title>
  </head>
  <body>
    哈喽! 请输入...<br><br>
    <form method="get" action="echo" name="message">
      名称: <input type="text" name="name"><br><br>
      <button>发送</button>
    </form>
  </body>
</html>
```

整份 HTML 文件的定义编写在<html>与</html>标签之间。在文件开始之前,浏览器必须先解析所有标签,<head>与</head>标签间用来定义文件主体前,提供给浏览器的一般信息,像是该范例的文件编码与标题信息等,就分别定义在<meta>以及<title>与</title>标签之间的内容。

接着浏览器针对文件主体内容解析,以便进行画面呈现与定义行为,文件主体内容定义在<body>标签中,例如
告诉浏览器换行,范例文件中有个代表图片的标签,这告知浏览器下载指定图片并显示。HTML 标签可以拥有属性,用来定义标签的额外信息,像是图片的来源(src 属性)。<form>标签定义了窗体,可让用户填写一些将要发送至 Web 网站的信息,其中还使用了<input>标签,分别定义输入字段及发送按钮。

简而言之,浏览器从 Web 网站取得这份 HTML 文件之后,就可以按照 HTML 定义的结构等信息进行画面的绘制。图 1.1 所示为大致的 HTML 标签与对应的画面呈现。