



华章科技

全球知名Java技术专家（《How Tomcat Works》作者）亲自执笔，全面解读Servlet 和 JSP 最新技术

重点阐述Java Web开发的重要编程概念和设计模型，系统学习Servlet和JSP的必读著作



华章程序员书库

Servlet & JSP: A Tutorial

Servlet和JSP学习指南

（加）Budi Kurniawan 著

崔毅 俞哲皆 俞黎敏 译



机械工业出版社
China Machine Press

013032829

TP312JA
1477

序员书库

Servlet & JSP: A Tutorial

Servlet和JSP学习指南

(加) Budi Kurniawan 著
崔毅 俞哲皆 俞黎敏 译



北航

C1640837

TP312JA
1477



机械工业出版社
China Machine Press

03850010

图书在版编目 (CIP) 数据

Servlet 和 JSP 学习指南 / (加) 克尼亚万 (Kurniawan, B.) 著; 崔毅, 俞哲皆, 俞黎敏译. —北京: 机械工业出版社, 2013.4

(华章程序员书库)

书名原文: Servlet & JSP: A Tutorial

ISBN 978-7-111-41861-0

I. S… II. ①克… ②崔… ③俞… ④俞… III. ①JAVA 语言—程序设计—指南 ②JAVA 语言—网页制作工具—程序设计—指南 IV. ①TP312-62 ②TP393.092-62

中国版本图书馆 CIP 数据核字 (2013) 第 053366 号

版权所有·侵权必究

封底无防伪标均为盗版

本书法律顾问 北京市展达律师事务所

本书版权登记号: 图字: 01-2012-4858

本书是系统学习 Servlet 和 JSP 的必读之作。由全球知名的 Java 技术专家 (《How Tomcat Works》作者) 亲自执笔, 不仅全面解读 Servlet 和 JSP 的最新技术, 重点阐述 Java Web 开发的重要编程概念和设计模型, 而且包含大量可操作性极强的案例。

本书共 18 章: 第 1 章介绍 Servlet API 和几个简单的 Servlet; 第 2 章讨论 Session 追踪, 以及保持状态的 4 种技术; 第 3 章和第 4 章系统讲解 JSP 的语法以及 JSP 中的重要特性之一: Expression Language; 第 5~7 章分别阐述 JSTL 中最重要的类库、标签的具体编写方法和标签文件; 第 8~10 章讨论 Servlet 中的事件驱动编程、过滤器, 以及 Model 2 架构; 第 11 章展示如何利用 Servlet 3 的文件上传特性, 以及如何在客户端改善用户的体验; 第 12 章解释如何通过编程方式将资源发送到浏览器; 第 13 章介绍如何利用 Decorator 模式以及类来改变 Servlet 请求和响应的行为; 第 14 章讨论 Servlet 3 中的一项新特性, 用来处理异步的操作; 第 15 章阐述如何通过声明和编程方式来保护 Java 的 Web 应用程序; 第 16 章讨论 Servlet/JSP 应用程序的部署过程, 以及部署描述符中的元素; 第 17 章阐述 Servlet 3 中的两项新特性; 第 18 章介绍 Struts 2 的用法。

Authorized translation from the English language edition entitled by Servlet & JSP: A Tutorial
Budi Kurniawan, published by Brainy Software, Inc, Copyright © 2012.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanic, including photocopying, recording, or by any information storage retrieval system, without permission of Brainy Software, Inc.

Chinese simplified language edition published by China Machine Press.

Copyright © 2013 by China Machine Press.

本书中文简体字版由 Brainy Software 授权机械工业出版社独家出版。未经出版者书面许可, 不得以任何方式复制或抄袭本书内容。

机械工业出版社 (北京市西城区百万庄大街 22 号 邮政编码 100037)

责任编辑: 谢晓芳

北京市荣盛彩色印刷有限公司印刷

2013 年 4 月第 1 版第 1 次印刷

186mm×240mm·21.25 印张

标准书号: ISBN 978-7-111-41861-0

定 价: 59.00 元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

客服热线: (010) 88378991 88361066

投稿热线: (010) 88379604

购书热线: (010) 68326294 88379649 68995259

读者信箱: hzsj@hzbook.com

译者序

Sun 公司于 1996 年发布了 Java Servlet 技术，与 CGI（Common Gateway Interface，公共网关接口）形成竞争，之后，它成为在 Web 中生成动态内容的标准。自从 Servlet 面世以来，也开发出了许多基于 Java 的 Web 框架，以帮助程序员更迅速地编写 Web 应用程序。目前全世界拥有了众多的 Java Web 开发人员，也是最热门的编程技术。作者 Budi Kurniawan 是 BrainySoftware.com 的高级架构师，也是《How Tomcat Works》、《Java for the Web with Servlets, JSP and EJB》以及《Struts 2 Design and Programming》、《Java 7: A Beginner's Tutorial》的作者。他已经发表了上百篇技术文章，并编写过授权给全球著名的大公司使用的软件。

本书读者对象

本书是针对有 Java 编程语言基础的 Web 开发者的，Java Web 应用程序开发是一种很成熟并且很热门的编程技术。同时，它也汇集各种技术于一身，经常令初学者不知道从何入手。如果你也有同感，那么本书就很适合你，因为它就是一本专门为初学者量身定制的教程。作为一套面对初学者的教程，本书不是要教会你每一种 Servlet/JSP 技术。如果你是一名有经验的 Web 应用开发者，对 Servlet 3.0 的新特性感兴趣，那么本书涵盖新特性的章节正是你所需要阅读并实践的内容。

章节简介

本书介绍了最重要的编程概念，并教你如何编写 Servlet/JSP，以及如何使用 Servlet 核心类库。对于编写真实的 Web 应用程序提供了很好的指导。本书是你所能找到的最全面的 Java Web 应用程序开发之入门教程，其主题有：

- ❑ Java Web 编程的核心技术 Servlet
- ❑ 4 种 Session 管理技术
- ❑ JSP
- ❑ 表达式语言
- ❑ JSTL 以及定制标签
- ❑ 定制标签文件
- ❑ Servlet 事件驱动编程之监听器
- ❑ 拦截请求的 Web 对象过滤器

- ❑ Model 2 架构
- ❑ 利用 Servlet 3.0 新特性进行文件上传以改善用户的体验
- ❑ 通过编程方式将资源发送到浏览器
- ❑ 利用 Decorator 模式等来改变 Servlet 请求和响应的行为
- ❑ Servlet 3.0 新特性之异步处理
- ❑ 通过声明和编程方式来保护 Java 的 Web 应用程序
- ❑ 应用程序的部署及描述符说明
- ❑ Servlet 3.0 新特性之动态注册和 Servlet 容器初始化

技术范围

本书旨在使你更全面、更专业地掌握 Java Web 应用程序开发技术，但是它只是一本比较基础的书籍，如果你已经有多年的 Java Web 应用程序开发经验，或许可以通过快速的阅读或者直接找到自己所需要的新技术点。当阅读本书时，你会遇到许多需要动手进行验证的实例，可以利用本书附带的示例程序进行练习与实践。示例与答案下载地址为：<http://books.brainysoftware.com/download/servletjsp.zip>。

正如在翻译过程中发现原著的错误一样，虽然我们在翻译过程中竭力以求信、达、雅，但限于自身水平，必定仍会有诸多不足，还望各位读者不吝指正。大家可以通过访问我的博客 <http://YuLimin.ItEye.com> 或者发送电子邮件到 YuLimin@163.com 进行互动。

关于术语的翻译，仍然沿用翻译《Effective Java 中文（第 2 版）》时采用的术语表以及满江红开放技术研究组织翻译术语，请见 <http://yulimin.iteye.com/blog/272088>。

感谢崔毅（<http://cuiyi.javaeye.com/>）对我在翻译中碰到的问题进行的深入讨论，并对本书翻译时所采用的术语进行了认真的磋商；感谢“满江红开放技术研究组织”的翻译同仁们在术语表讨论中提出许多中肯的建议；感谢满江红开源组织的曹晓钢提供的一些翻译注意事项和热情的帮助；感谢机械工业出版社的编辑认真仔细地审稿，辛苦了，谢谢！

本书由我组织翻译，崔毅负责翻译第 1～8 章，俞哲皆负责翻译第 9～14 章，我负责翻译前言、第 15～18 章、附录并对全书所有章节进行全面审校，还负责对原文中的错误与作者进行沟通并加以修正。参与翻译与审校的还有：杨春花、崔毅、张琬滢、蒋凌锋、魏伟、万国辉等，在此再次深表感谢。

本书章节安排合理，内容承上启下，但是需要边看书边动手做实验，才能充分理解并掌握 Java Web 应用程序开发技术及新特性。快乐分享，实践出真知，最后，祝大家能够像我一样在阅读中享受本书带来的乐趣！

Read a bit and take it out, then come back read some more.

俞黎敏

前 言

欢迎你阅读本书，其内容涵盖了 Servlet 3.0 和 JSP 2.2 方面的技术。

Java Servlet 技术，或简称 Servlet，是 Java 中用于开发 Web 应用程序的基本技术。Sun 公司于 1996 年发布了 Java Servlet 技术，与 CGI（Common Gateway Interface，公网网关接口）形成竞争。之后，它成为在 Web 中生成动态内容的标准。CGI 的主要问题在于，它为每一个 HTTP 请求都创建一个新的进程。因为创建进程需要花费大量的 CPU 周期，这使得编写可扩展的 CGI 程序变得极为困难。另一方面，Servlet 程序也比 CGI 程序运行得更快，这是因为 Servlet 执行完它的第一个请求之后，就会驻留在内存中，等待后续的请求。

自从 Servlet 面世以来，也开发出了许多基于 Java 的 Web 框架，以帮助程序员更迅速地编写 Web 应用程序。这些框架可以使我们只关注业务逻辑，而不在编写样板代码（boilerplate code）上花费太多的时间。但你还是需要了解 Servlet 的基本知识。后来，JavaServer Pages（JSP）发布了，这使得编写 Servlet 变得更加轻松了。或许你正在使用一种很好的框架，如 Struts 2、Spring MVC，或者 JavaServer Faces。但是，如果没有充分理解 Servlet 和 JSP 方面的知识，你将无法进行高效的编码。顺便说一下，Servlets 是指在 Servlet 容器中运行的 Java 类。Servlet 容器或者 Servlet 引擎，就像是一个 Web 服务器，但它能够生成动态的内容，而不只是提供静态的资源。

目前的 Servlet 3.0 是在 JSR（Java Specification Request，Java 规范请求）315 中定义的（<http://jcp.org/en/jsr/detail?id=315>）。它需要用 Java Standard Edition 6 或者更新的版本。JSP 2.2 则是在 JSR 245 中定义的（<http://jcp.org/en/jsr/detail?id=245>）。本书假设你事先已经了解 Java 和面向对象编程。如果你是刚刚接触 Java，那么建议你先看一下笔者的另一本书：《Java 7: A Beginner's Tutorial》^①，ISBN 978-0-9808396-1-6。

下一节将讨论 Servlet/JSP 应用程序架构（Application Architecture）、HTTP 协议，以及书中各章节的主要内容。

Servlet/JSP 应用程序架构

Servlet 是一个 Java 程序。一个 Servlet 应用程序经常包含一个或者多个 Servlet。JSP 页面要被翻译成 Servlet，并进行编译。

Servlet 应用程序是在 Servlet 容器中运行的，它不能自动运行。Servlet 容器将用户的请

^① 中文版书名：《Java 7 程序设计》，由机械工业出版社引进并出版。——编辑注

求传给 Servlet 应用程序，并将 Servlet 应用程序的响应回传给用户。大多数 Servlet 应用程序至少都包含几个 JSP 页面。因此，将 Java 的 Web 应用程序称为“Servlet/JSP 应用程序”比“Servlet 应用程序”更为恰当一些。

Web 用户可以利用如 Internet Explorer、Mozilla Firefox 或者 Google Chrome 这类 Web 浏览器来访问 Servlet 应用程序。一个 Web 浏览器相当于一个 Web 客户端。图 1 中展示了一个 Servlet/JSP 应用程序的架构。

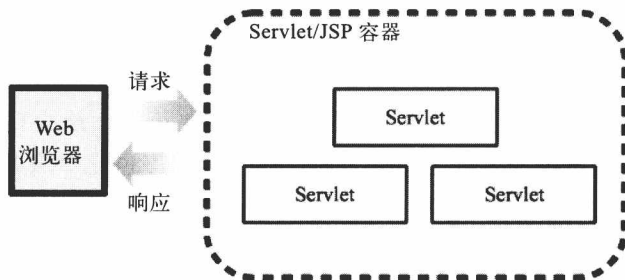


图 1 Servlet/JSP 应用程序架构

Web 服务器与 Web 客户端之间通过双方都熟悉的一种语言进行通信，即超文本转移协议（HyperText Transfer Protocol，HTTP）。为此，Web 服务器也称作 HTTP 服务器。关于 HTTP 的更多内容请参见下一节。

Servlet/JSP 容器是一种特殊的 Web 服务器，它可以处理 Servlet，并提供静态内容。在过去，人们更习惯于将 Servlet/JSP 容器作为 HTTP 服务器的一个模块来运行，如 Apache HTTP Server，因为他们认为 HTTP 服务器比 Servlet/JSP 容器更健壮。在这种情况下，Servlet/JSP 容器的任务是生成动态内容，HTTP 服务器则负责提供静态资源。如今，人们认为 Servlet/JSP 容器已经成熟，它们不需要 HTTP 服务器就可以进行广泛的部署。Apache Tomcat 和 Jetty 是其中最为盛行的两种 Servlet/JSP 容器，并且它们都是免费、开源的。可以分别从以下两个网站下载二者：<http://tomcat.apache.org> 和 <http://jetty.codehaus.org>。

Servlet 和 JSP 是 Java Enterprise Edition（EE）定义的众多技术当中的两种。其他的 Java EE 技术还包括 Java Message Service（JMS）、Enterprise JavaBeans（EJB）、JavaServer Faces（JSF），以及 Java Persistence。Java EE 6（当前的最新版本）的完整技术列表，可以在以下网站找到：

<http://www.oracle.com/technetwork/java/javaeetech/index.html>

运行 Java EE 应用程序时，需要一个 Java EE 容器，如 GlassFish、JBoss、Oracle WebLogic，以及 IBM WebSphere。也可以将 Servlet/JSP 应用程序部署在 Java EE 容器中，只不过用 Servlet/JSP 容器已经绰绰有余，它还比 Java EE 容器更轻量化。Tomcat 和 Jetty 则不属于

Java EE 容器，因此它们不能运行 EJB 或者 JMS。

超文本转移协议（HTTP）

HTTP 协议使得 Web 服务器与浏览器之间可以通过互联网（Internet）或者企业内部网（Intranet）来交换数据。万维网联盟（World Wide Web Consortium, W3C）是一个开发标准的国际化社区，它负责修订和维护这个协议。HTTP 的第一个版本是 HTTP 0.9，之后是 HTTP 1.0。再后来，HTTP 1.0 又被目前的 HTTP 1.1 取代。HTTP 1.1 是在 W3C 的 RFC（Request for Comments）2616 中定义的，它可以通过以下网址下载到：<http://www.w3.org/Protocols/HTTP/1.1/rfc2616.pdf>。

Web 服务器每天 24 小时，每周 7 天不间断地运行，随时等待 HTTP 客户端（一般是 Web 浏览器）的连接和资源请求。在 HTTP 中，总是由客户端发起连接，服务器从不主动联系客户端。在查找某一个资源时，互联网用户是通过单击一个包含 URL（Uniform Resource Locator，统一资源定位器）的链接，或在他/她的浏览器地址栏中输入一个 URL 进行的。下面举 URL 的两个例子：

```
http://google.com/index.html
http://facebook.com/index.html
```

URL 的第一部分是 `http`，它是协议的标识。并非所有的 URL 都用 HTTP。例如，下面这两个 URL 虽然没有使用 HTTP，但它们也都是有效的：

```
mailto:joe@example.com
ftp://marketing@ftp.example.org
```

一般来说，URL 的格式是这样的：

```
protocol://[host.]domain[:port][/context][/resource][?query string]
```

或者如下所示：

```
protocol://IP address[:port][/context][/resource][?query string]
```

以上方括号中的内容都是可选的，因此 URL 也可以像 `http://yahoo.ca` 或者 `http://192.168.1.9` 这么简单。顺便说一下，IP 地址其实是分配给某一台计算机或者另一种设备的一个数字标签。一台计算机可以有多个域名，因此，几个域名也可以共用一个 IP 地址。换句话说，如果不用 `http://google.com`，也可以用它的 IP 地址：`http://209.85.143.99`。为了查出某个域名的 IP 地址，可以在计算机的控制台或命令行中使用 `ping` 命令，如：


```
ping google.com
```

IP 地址很难记住，因此人们更喜欢使用域名。但你知道吗？像 `example.com` 和 `example.org` 这样的域名是无法购买到的，因为它们是留着备用的，比如编写文档时作为示例用。

主机名 `host` 部分可以有，但与没有主机名 `host` 的地址相比，它表示的则是互联网或者内部网中另一个完全不同的位置。例如，`http://yahoo.com`（没有主机名 `host`）与 `http://mail.yahoo.com`（有主机名，为 `mail`）表示的是两个完全不同的位置。在过去，`www` 是最盛行的主机名，因此它变成是默认的。一般来说，`http://www.domainName` 就是指 `http://domainName`。

80 是 HTTP 的默认端口。因此，如果 Web 服务器是在 80 端口上运行，那么不需要端口号也能到达服务器。但是，有的时候，如果 Web 服务器不是在端口 80 上运行的，那么就需要输入端口号。例如，Tomcat 默认是在端口 8080 上运行，因此需要提供端口号：

```
http://localhost:8080
```

`localhost` 是一个保留名称，一般用来表示本机，即正在运行 Web 浏览器的这台计算机。

URL 中的 `context` 部分是指应用程序的名称，但这个也是可选的。Web 服务器可以运行多个 `context` 或多个应用程序，其中一个可以设置为默认的 `context`。如果要请求的是默认 `context` 中的资源，则 URL 中的 `context` 部分就可以忽略。

最后，一个 `context` 可以有一个或者多个默认资源（一般为 `index.html`、`index.htm` 或者 `default.htm`）。一个没有资源名称的 URL，通常被当作是默认资源。当然，如果一个 `context` 中存在多个默认资源，那么当客户端没有指定资源名称时，将总是返回优先级最高的那一个。

在资源名称之后，一般是一个或者多个查询字符串。查询字符串是指可以传到服务器进行处理的一个键 / 值对。后面的章节会介绍到更多关于查询字符串的内容。

接下来的几个小节将详细介绍 HTTP 请求和响应。

HTTP 请求

一个 HTTP 请求中通常包含三个部分：

- 方法 / 统一资源标识符（Uniform Resource Identifier, URI）/ 协议 / 版本
- 请求标头
- 实体主体

下面是一个 HTTP 请求的示例：

```
POST /examples/default.jsp HTTP/1.1
Accept: text/plain; text/html
Accept-Language: en-gb
Connection: Keep-Alive
Host: localhost
```

```
User-Agent: Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10.5; en-US;
rv:1.9.2.6) Gecko/20100625 Firefox/3.6.6
Content-Length: 30
Content-Type: application/x-www-form-urlencoded
Accept-Encoding: gzip, deflate
```

```
lastName=Blanks&firstName=Mike
```

方法 /URI/ 协议版本号, 放在请求的第一行。如:

```
POST /examples/default.jsp HTTP/1.1
```

这里的 POST 是请求方法, /examples/default.jsp 是 URI, HTTP/1.1 是协议 / 版本号部分。

HTTP 请求可以利用 HTTP 标准中定义的其中一个请求方法。HTTP 1.1 支持 7 种请求类型: GET、POST、HEAD、OPTIONS、PUT、DELETE 以及 TRACE。GET 和 POST 是互联网应用程序中最常用的。

URI 用于指定一个互联网资源, 它通常解读为是相对于服务器的根目录。因此, 它始终应该以一个正斜线 (/) 开头。统一资源定位器 (URL) 实际上也是一种 URI (详情请查看 <http://www.ietf.org/rfc/rfc2396.txt>)。

在一个 HTTP 请求中, 请求标头包含关于客户端环境和请求实体主体的有用信息。例如, 它可以包含为浏览器设置的语言、实体主体的长度等。各标头之间用一个回车换行 (Carriage Return/LineFeed, CRLF) 序列符隔开。

在标头和实体主体之间是一个空行 (CRLF), 它对于 HTTP 请求格式是很重要的。CRLF 告诉 HTTP 服务器, 实体主体从这里开始。在有些互联网编程书籍中, 它们是把这个 CRLF 当作是 HTTP 请求的第四个部分。

在前一个 HTTP 请求中, 实体主体只有下面这一行:

```
lastName=Blanks&firstName=Mike
```

实体主体常常比一个典型的 HTTP 请求更长。

HTTP 响应

与 HTTP 请求类似, 一个 HTTP 响应中通常也包含三个部分:

- 协议 / 状态码 / 描述
- 响应标头
- 实体主体

下面是一个 HTTP 响应的示例:

```
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
```

```
Date: Thu, 5 Jan 2012 13:13:33 GMT
Content-Type: text/html
Last-Modified: Wed, 4 Jan 2012 13:13:12 GMT
Content-Length: 112
```

```
<html>
<head>
<title>HTTP Response Example</title>
</head>
<body>
Welcome to Brainy Software
</body>
</html>
```

响应标头的第一行与请求标头的第一行类似。它在告诉我们，所使用的协议版本是 HTTP 1.1，并且请求成功（200 是成功状态码）。

响应标头也包含与 HTTP 请求中的标头类似的有用信息。响应的实体主体是响应本身的 HTML 内容。标头和实体主体之间用一系列的回车换行符（CRLF）隔开。

当且仅当 Web 服务器能够找到所请求的资源时，才会发出状态码 200。如果无法找到某个资源，或者无法理解请求，服务器就会发出一个不同的请求代码。例如，401 是未授权访问的状态码，405 表示未经允许的 HTTP 方法。关于 HTTP 状态码的完整列表，请查看以下在线文档：

<http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html>

关于本书

下面介绍一下各章节的主要内容。

第 1 章介绍 Servlet API，还示范了几个简单的 Servlet。该章主要关注 Servlet API 4 个 Java 包当中的两个：`javax.servlet` 包和 `javax.servlet.http` 包。

第 2 章讨论 Session 追踪，或称作 Session 管理。由于 HTTP 的无状态性，因此这是 Web 应用程序开发中一个非常重要的主题。该章探讨保持状态的 4 种技术：改写 URL、隐藏域、cookie 以及 `HTTPSession` 对象。

JavaServer Pages（JSP）是 Servlet 的一种补充技术。第 3 章介绍 JSP 语法，包括它的指令、脚本元素以及动作。

第 4 章阐述 JSP 2.0 中增加的最重要特性之一：Expression Language（EL）。EL 旨在能够使我们设计出无脚本的 JSP 页面，并且能够协助你写出更简短、更高效的 JSP 页面。在该章中，你还会学到如何利用 EL 来访问 JavaBeans 和内置对象（Scoped Object）。

第 5 章阐述 JavaServer Pages Standard Tag Library（JSTL）中最重要的类库，这是一个

定制标签类库的集合，用于解决像迭代映射或集合、条件测试、XML 处理以及数据库访问和数据操作这类常见的问题。

通常情况下，是利用 JSTL 访问内置对象，以及完成 JSP 页面中的其他任务。但是，对于更为具体的任务，可能需要你自己编写定制标签。第 6 章就教你具体的编写方法。

第 7 章讨论标签文件，这是 JSP 2.0 中的一项新特性，它使得编写定制动作变得更加简单。该章讨论只利用标签文件来编写定制标签的几个方面。

第 8 章讨论 Servlet 中的事件驱动编程。其中讨论了 Servlet API 中的事件类和监听器，并介绍了如何编写监听器，以及如何在 Servlet/JSP 应用程序中使用它们。

第 9 章讲解过滤器，它们是拦截请求的 Web 对象。该章内容涵盖了 Filter API，其中包括 Filter、FilterConfig 及 FilterChain 接口，并介绍如何通过实现 Filter 接口来编写过滤器。

第 10 章解释 Model 2 架构，这是针对除最简单的 Java Web 应用程序之外的建议架构。该章还提供了几个范例，展示了 Model 2 应用程序中的不同组件。

第 11 章展示如何利用 Servlet 3 的文件上传特性，以及如何在客户端改善用户的体验。

第 12 章解释如何通过编程方式将一个资源发送到浏览器。

Servlet API 提供了用于包装 Servlet 请求和响应的类。在第 13 章中，将会学到如何利用 Decorator 模式及这些类来改变 Servlet 请求和响应的行为。

第 14 章讨论 Servlet 3.0 中的一项新特性，其用来处理异步的操作。如果 Servlet/JSP 应用程序中有一个或多个长时间运行的操作，那么这项特性就特别有帮助。该特性的原理是将那些操作分配给一个新的线程，从而将处理线程的请求释放回到池中，准备为另一个请求提供服务。

第 15 章阐述如何通过声明和编程方式来保护 Java 的 Web 应用程序。这里讨论了 4 个主要的安全性主题：验证、授权、保密性及数据完整性。

第 16 章讨论 Servlet/JSP 应用程序的部署过程，以及部署描述符中的元素。

第 17 章阐述 Servlet 3 中的两项新特性。动态注册是指不需要重启应用程序就可以动态地注册 Web 对象。框架开发者自然会喜欢 Servlet 容器初始化。

第 18 章介绍作为 MVC 框架的 Struts 2。该章阐述了 Struts 2 的基础组件和配置，并展示了一个简单的应用程序。

附录 A 阐述如何安装和配置 Tomcat，以及如何在各种操作系统中运行它。

附录 B 列出可以用来配置 Web 对象的所有注解，如 Servlet、监听器及过滤器。这些注解是 Servlet 3 的新特性，它们使部署描述符变成是可选的。

附录 C 阐述如何利用 KeyTool 程序生成公 / 私钥对，以及如何请一家可信任的证书颁发机构，将公钥变成一份数字证书。

下载应用程序范例

从以下网站可以下载到本书配套应用程序范例的压缩档：

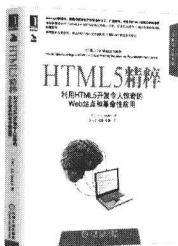
<http://books.brainysoftware.com/download>

选择框架

当你学完本书，并掌握了 Servlet 和 JSP 之后，最好至少选择学习一种 Web 框架。目前有许多优秀的框架可以免费使用，例如 Struts 2、JavaServer Faces、Spring MVC 及 Google Web Toolkit 等都不错。框架可以解决 Servlet/JSP 开发中常见的问题，大大缩短开发时间。

你可以根据工作需要，选择目前最需要的那些框架。

推荐阅读



HTML 5精粹

Amazon畅销书，被翻译成西班牙语等多种文字，广受好评，被誉为HTML 5领域的经典著作

详尽地讲解和分析了HTML 5中的所有新特性和核心技术，能为有一定HTML基础的读者迅速提升HTML 5开发技能提供绝佳指导



HTML 5游戏开发实战

清晰而全面地展示了如何使用最新的HTML 5和CSS 3标准来构建纸牌、绘图、物理等各种常见类型的游戏

以实例为导向，系统介绍网络游戏开发技术，结合具体示例的操作步骤讲解，浅显易懂，适合网络游戏开发人员、管理人员阅读



HTML 5游戏开发实践指南

来自美国硅谷的资深专家亲自执笔，包含大量实例，可操作性极强
全面讲解HTML 5游戏开发所需要掌握的各种最新技术、工具和框架（包含Canvas、SVG、WebGL等），以及开发的思维和方法



HTML 5开发精要与实例详解

畅销书《HTML 5与CSS 3权威指南》姊妹篇，HTML 5实战进阶必备

注重实战：包含28个中大型案例，每个案例既有直观的效果图，又有详尽的源代码分析

讲解透彻：对每个案例所涉及的理论精要进行深入剖析，理论与实践完美结合



HTML 5实战

依据HTML 5标准的最新草案编写，对HTML 5进行了系统、全面、透彻的讲解

106个精心设计的经典案例对各个知识点进行补充和阐释，理论与实践完美结合

目 录

译者序 前 言

第 1 章 Servlet / 1

- 1.1 Servlet API 概述 / 1
- 1.2 Servlet / 2
- 1.3 编写基础的 Servlet 应用程序 / 3
- 1.4 ServletRequest / 6
- 1.5 ServletResponse / 7
- 1.6 ServletConfig / 8
- 1.7 ServletContext / 10
- 1.8 GenericServlet / 11
- 1.9 HTTP Servlet / 13
- 1.10 处理 HTML 表单 / 15
- 1.11 使用部署描述符 / 21
- 1.12 小结 / 23

第 2 章 Session 管理 / 24

- 2.1 网址重写 / 24
- 2.2 隐藏域 / 29
- 2.3 cookie / 34
- 2.4 HttpSession 对象 / 42
- 2.5 小结 / 51

第 3 章 JSP / 52

- 3.1 JSP 概述 / 52
- 3.2 备注 / 57
- 3.3 隐式对象 / 57

- 3.4 指令 / 60
- 3.5 脚本元素 / 63
- 3.6 动作 / 68
- 3.7 小结 / 71

第 4 章 EL / 72

- 4.1 EL 语法 / 72
- 4.2 访问 JavaBean / 74
- 4.3 EL 隐式对象 / 75
- 4.4 使用其他 EL 运算符 / 78
- 4.5 使用 EL / 80
- 4.6 在 JSP 2.0 及更高版本中
配置 EL / 83
- 4.7 小结 / 85

第 5 章 JSTL / 86

- 5.1 下载 JSTL / 86
- 5.2 JSTL 类库 / 86
- 5.3 通用动作指令 / 87
- 5.4 条件式动作指令 / 91
- 5.5 iterator 动作指令 / 93
- 5.6 格式化动作指令 / 102
- 5.7 函数 / 109
- 5.8 小结 / 114

第 6 章 编写定制标签 / 115

- 6.1 定制标签概述 / 115
- 6.2 简单的标签处理器 / 116

- 6.3 SimpleTag 范例 / 116
- 6.4 处理属性 / 119
- 6.5 管理标签主体 / 122
- 6.6 编写 EL 函数 / 125
- 6.7 发布定制标签 / 126
- 6.8 小结 / 128

第 7 章 标签文件 / 129

- 7.1 标签文件简介 / 129
- 7.2 我们的第一个标签文件 / 130
- 7.3 标签文件指令 / 131
- 7.4 doBody / 139
- 7.5 invoke / 141
- 7.6 小结 / 143

第 8 章 监听器 / 144

- 8.1 监听器接口和注册 / 144
- 8.2 Servlet Context 监听器 / 145
- 8.3 Session 监听器 / 148
- 8.4 ServletRequest 监听器 / 153
- 8.5 小结 / 155

第 9 章 过滤器 / 156

- 9.1 Filter API / 156
- 9.2 过滤器的配置 / 157
- 9.3 范例 1: 日志过滤器 / 159
- 9.4 范例 2: 图片保护过滤器 / 163
- 9.5 范例 3: 下载计数过滤器 / 164
- 9.6 过滤器的顺序 / 168
- 9.7 小结 / 169

第 10 章 应用程序设计 / 170

- 10.1 Model 1 概述 / 170

- 10.2 Model 2 概述 / 170
- 10.3 基于 Servlet Controller 的 Model 2 / 172
- 10.4 基于 Filter Dispatcher 的 Model 2 / 181
- 10.5 验证器 / 184
- 10.6 数据库访问 / 189
- 10.7 依赖注入 / 199
- 10.8 小结 / 208

第 11 章 文件上传 / 209

- 11.1 客户端编程 / 209
- 11.2 服务器端编程 / 210
- 11.3 上传 Servlet 范例 / 212
- 11.4 多文件上传 / 214
- 11.5 上传客户端 / 217
- 11.6 小结 / 223

第 12 章 文件下载 / 224

- 12.1 文件下载概述 / 224
- 12.2 范例 1: 隐藏资源 / 225
- 12.3 范例 2: 防止跨站引用 / 230
- 12.4 小结 / 232

第 13 章 请求和响应的装饰 / 233

- 13.1 Decorator 模式 / 233
- 13.2 Servlet Wrapper 类 / 234
- 13.3 范例: AutoCorrect 过滤器 / 235
- 13.4 小结 / 242

第 14 章 异步处理 / 243

- 14.1 概述 / 243

- 14.2 编写异步的 Servlet 和 Filter / 243
- 14.3 编写异步的 Servlet / 244
- 14.4 异步监听器 / 249
- 14.5 小结 / 251

第 15 章 安全性 / 252

- 15.1 验证和授权 / 252
- 15.2 验证方法 / 256
- 15.3 SSL / 263
- 15.4 通过编程确保安全性 / 268
- 15.5 小结 / 271

第 16 章 部署 / 272

- 16.1 部署描述符概述 / 272
- 16.2 部署 / 284
- 16.3 Web Fragment / 285
- 16.4 小结 / 287

第 17 章 动态注册和 Servlet 容器初始化 / 288

- 17.1 动态注册 / 288
- 17.2 Servlet 容器初始化 / 291
- 17.3 小结 / 293

第 18 章 Struts 2 简介 / 294

- 18.1 Struts 2 的优势 / 294
- 18.2 Struts 2 工作原理 / 295
- 18.3 拦截器 / 297
- 18.4 Struts 2 的配置文件 / 299
- 18.5 简单的 Struts 应用程序 / 307
- 18.6 小结 / 311

附录 A Tomcat / 312

附录 B Web 注解 / 316

附录 C SSL 证书 / 320