

*JavaServer Faces*



JavaServer<sup>TM</sup> Faces

交互式网站界面设计

**O'REILLY®**

东南大学出版社

*Hans Bergsten* 著

O'Reilly Taiwan公司 编译

---

# JavaServer™ Faces

## 交互式网站界面设计

*Hans Bergsten* 著

*O'Reilly Taiwan* 公司 编译

**O'REILLY®**

*Beijing • Cambridge • Farnham • Köln • Paris • Sebastopol • Taipei • Tokyo*

O'Reilly Media, Inc. 授权东南大学出版社出版

东南大学出版社

## 图书在版编目 (CIP) 数据

JavaServer™ Faces 交互式网站界面设计 / (美) 伯格  
斯腾 (Bergsten, H.) 著; O'Reilly Taiwan 公司编译. —南  
京: 东南大学出版社, 2006.1

书名原文: JavaServer™ Faces

ISBN 7-5641-0161-X

I. J... II. ①伯... ②O... III. JAVA 语言—程序设计  
IV. TP312

中国版本图书馆 CIP 数据核字 (2005) 第 057382 号

江苏省版权局著作权合同登记

图字: 10-2005-087 号

©2004 by O'Reilly Media, Inc.

Simplified Chinese Edition, jointly published by O'Reilly Media, Inc. and Southeast University Press, 2005. Authorized translation of the English edition, 2004 O'Reilly Media, Inc., the owner of all rights to publish and sell the same.

All rights reserved including the rights of reproduction in whole or in part in any form.

英文原版由 O'Reilly Media, Inc. 出版 2004。

简体中文版由东南大学出版社出版 2005。英文原版的翻译得到 O'Reilly Media, Inc. 的授权。此简体中文版的出版和销售得到出版权和销售权的所有者——O'Reilly Media, Inc. 的许可。

版权所有, 未得书面许可, 本书的任何部分和全部不得以任何形式重制。

书 名 / JavaServer™ Faces 交互式网站界面设计

书 号 / ISBN 7-5641-0161-X

责任编辑 / 张烨

封面设计 / Emma Colby, 张健

出版发行 / 东南大学出版社

地 址 / 南京四牌楼 2 号 邮编 210096

印 刷 / 扬中市印刷有限公司

开 本 / 787 毫米 × 980 毫米 16 开本 41.75 印张

版 次 / 2006 年 1 月第 1 版 2006 年 1 月第 1 次印刷

印 数 / 0001-4000 册

定 价 / 65.00 元 (册)

# O'Reilly Media, Inc.介绍

为了满足读者对网络 and 软件技术知识的迫切需求，世界著名计算机图书出版机构 O'Reilly Media, Inc. 授权东南大学出版社，翻译出版一批该公司久负盛名的英文经典技术专著。

O'Reilly Media, Inc. 是世界上在 Unix、X、Internet 和其他开放系统图书领域具有领导地位的出版公司，同时也是联机出版的先锋。

从最畅销的《The Whole Internet User's Guide & Catalog》（被纽约公共图书馆评为二十世纪最重要的 50 本书之一）到 GNN（最早的 Internet 门户和商业网站），再到 WebSite（第一个桌面 PC 的 Web 服务器软件），O'Reilly Media, Inc. 一直处于 Internet 发展的最前沿。

许多书店的反馈表明，O'Reilly Media, Inc. 是最稳定的计算机图书出版商——每一本书都一版再版。与大多数计算机图书出版商相比，O'Reilly Media, Inc. 具有深厚的计算机专业背景，这使得 O'Reilly Media, Inc. 形成了一个非常不同于其他出版商的出版方针。O'Reilly Media, Inc. 所有的编辑人员以前都是程序员，或者是顶尖级的技术专家。O'Reilly Media, Inc. 还有许多固定的作者群体——他们本身是相关领域的技术专家、咨询专家，而现在编写著作，O'Reilly Media, Inc. 依靠他们及时地推出图书。因为 O'Reilly Media, Inc. 紧密地与计算机业界联系着，所以 O'Reilly Media, Inc. 知道市场上真正需要什么图书。

# 目录

|                               |    |
|-------------------------------|----|
| 前言 .....                      | 1  |
| 第一章 JavaServer Faces 概论 ..... | 7  |
| 何谓 JavaServer Faces .....     | 7  |
| JSF 与传统技术的比较 .....            | 9  |
| JSF 在大结构中的定位 .....            | 14 |
| 准备工作 .....                    | 15 |
| 第二章 JSF 开发程序 .....            | 17 |
| 开发具有 JSF 用户界面的应用系统 .....      | 18 |
| 开发后台程序 .....                  | 19 |
| 开发组件与集成程序 .....               | 20 |
| 开发用户界面网页 .....                | 23 |
| 第三章 准备 JSF 环境 .....           | 27 |
| 安装 Java 软件开发工具 .....          | 28 |
| 安装 Tomcat Server .....        | 29 |
| 测试 Tomcat .....               | 33 |

|                                                |            |
|------------------------------------------------|------------|
| 安装本书范例 .....                                   | 34         |
| 关于本书的范例 .....                                  | 36         |
| <b>第四章 Servlet 与 JavaServer Pages 概论 .....</b> | <b>38</b>  |
| HTTP .....                                     | 38         |
| 网站应用系统的配置以及运行时环境 .....                         | 45         |
| Servlet、Filter 与 Listener .....                | 48         |
| JavaServer Pages .....                         | 56         |
| 访问应用系统数据 .....                                 | 69         |
| <b>第五章 事务逻辑与身份验证 .....</b>                     | <b>72</b>  |
| 示范应用系统的大体结构 .....                              | 72         |
| 事务逻辑类的实现 .....                                 | 74         |
| 验证与授权 .....                                    | 87         |
| <b>第六章 组件的创建与绘制 .....</b>                      | <b>94</b>  |
| JSF 的基础 .....                                  | 94         |
| 组件与模型属性之间的绑定 .....                             | 110        |
| 选择性地绘制组件 .....                                 | 119        |
| <b>第七章 验证外界输入 .....</b>                        | <b>127</b> |
| 处理语法错误 .....                                   | 127        |
| 使用标准验证器 .....                                  | 130        |
| 定义自定义错误信息 .....                                | 138        |
| 使用自定义验证器 .....                                 | 139        |
| 其他验证输入值的办法 .....                               | 150        |
| <b>第八章 事件处理 .....</b>                          | <b>151</b> |
| JSF 事件模型 .....                                 | 152        |

|                                                |            |
|------------------------------------------------|------------|
| 处理后台事件 .....                                   | 156        |
| UI 事件 .....                                    | 164        |
| <b>第九章 浏览控制 .....</b>                          | <b>172</b> |
| 变换 JSF 视图 .....                                | 172        |
| 返回非 JSF-view 响应视图 .....                        | 179        |
| 对 Non-JSF Request 返回一个 JSF View Response ..... | 181        |
| 结论 .....                                       | 183        |
| <b>第十章 表格数据的处理 .....</b>                       | <b>184</b> |
| 显示只读数据表 .....                                  | 184        |
| 表行事件的处理 .....                                  | 188        |
| 超大表格的处理 .....                                  | 193        |
| 编辑表格数据 .....                                   | 202        |
| <b>第十一章 国际化 .....</b>                          | <b>209</b> |
| 输出信息的区域化 .....                                 | 210        |
| 处理区域化应用系统的输入 .....                             | 220        |
| 非西方语系的处理 .....                                 | 222        |
| <b>第十二章 精雕细琢 .....</b>                         | <b>230</b> |
| 使用多个 JSP 文件构成一个视图 .....                        | 230        |
| JSF View 与其他内容的结合 .....                        | 239        |
| Struts 与 JSF 的集成 .....                         | 243        |
| 程序化修改组件 .....                                  | 251        |
| 使用 PhaseListener .....                         | 255        |
| 关于调试与错误处理的构想 .....                             | 257        |
| <b>第十三章 开发自定义绘制器与其他外挂类 .....</b>               | <b>264</b> |
| 开发自定义的绘制器 .....                                | 265        |

|                                        |            |
|----------------------------------------|------------|
| 使用其他自定义类 .....                         | 290        |
| 自定义类的封装 .....                          | 291        |
| <b>第十四章 开发自定义组件 .....</b>              | <b>292</b> |
| 扩充现有的组件 .....                          | 292        |
| 从头开发新组件 .....                          | 301        |
| <b>第十五章 开发自定义表示层 .....</b>             | <b>325</b> |
| ViewHandler 类 .....                    | 325        |
| 使用 Java 类作为 View .....                 | 326        |
| 使用纯 HTML 模板搭配 XML 视图定义文件 .....         | 335        |
| <b>附录一 标准 JSF 标记库 .....</b>            | <b>349</b> |
| <b>附录二 JSF EL 语法参考 .....</b>           | <b>409</b> |
| <b>附录三 标准 JSF 组件与 Render Kit .....</b> | <b>417</b> |
| <b>附录四 基础 API 参考数据 .....</b>           | <b>525</b> |
| <b>附录五 JSF 配置文件参考数据 .....</b>          | <b>598</b> |
| <b>附录六 网站应用系统的结构与配置描述文件参考数据 .....</b>  | <b>612</b> |
| <b>索引 .....</b>                        | <b>637</b> |



---

# 前言

JavaServer™ Faces (JSF) 是 Java 最新的网站应用系统技术，它是 Java Servlet、Java Server™ Pages (JSP) 以及多种 webapp framework (包括商用与 open source) 的经验累积成果。长久以来，对于独立 GUI 应用程序的开发，已形成一个很成功的标准模型，但是对于网站应用系统，至今仍缺乏这么一套标准。近年来，有越来越多的厂商开始关注此问题，在高度期盼下，Java Community Process (JCP) 终于发表了一份满足这方面需求的规范说明书，也就是 JavaServer™ Faces (后文简称 JSF)。JSF 是一个以组件为基础的事件驱动式模型，它很类似于独立 GUI 应用程序的开发模型。在 JSF 模型下，厂商可以专注于发展第一流的开发工具，而不必另外研发特殊的 API，或是重新发明别人已经发明的玩意儿，而你将获得市场规模扩大所带来的好处。

## 本书内容

本书涵盖 2004 年 3 月发表的 JSP 1.0 规范。

前四章介绍 JSF 提供了些什么以及它与 Java 的其他企业技术之间的关系。你也会学到基础技术的概念，包括 HTTP、Java Web Container、Servlet 与 JSP 以及如何架设一个 Java Web Container。

接着，你将学到如何使用 JSF 的所有标准组件及其最常用的典型功能（如：检验输入数据、转换数据类型、网页浏览、国际化），以及如何开发符合特殊需求的自定义组件。本书以一个网站应用系统为例，于每一章分别解说如何使用特定功能来改良或实现该应用系统，除了示范如何着手进行之外，还解释功能背后发生了什么事，借此协助读者了解 JSF 的运作原理以及如何在自己的应用系统中运用相关技术。

本书引用的范例是一个费用核报系统，可供用户建立、编辑、提交费用报告；而管理员可以决定要接受还是拒绝。这是一个可以展示 JSF 各项机制的真实例子，提供了常见设计问题的一般解法，像是如何验证用户身份、既将事务逻辑对象连接到 UI 又保持各环节不密合、区别 UI 事件与 APP 事件、输出信息的调整与翻译、以模块化手法建构完整的 UI。最后三章属于高级内容，示范如何开发自定义组件以及自定义的表示层。

## 读者

开发一个网站应用系统需要集合不同技术的专家，而 JSF 规范说明书赋予每种专家一个角色名称。如网页设计者（page author）是知道客户端技术（例如 HTML、CSS 与 JavaScript）但不是程序员的人员；应用系统开发者（application developer）是了解应用领域、也知道如何开发事务对象来构成解决方案的 Java 程序员；组件作者（component writer）是有能力开发自定义组件的专家，他们能将网页设计者提供的 UI 连接到应用系统开发者提供的对象。组件作者必须懂 Java 与 JSF API，而且熟悉 JSF 如何处理 request 以及如何产生 response。本书涵盖了上述三种角色所掌握的所有基本技术。

## 必备基础知识

假设本书读者已经有使用 HTML 语言的经验，这表示我可以不用解释范例中用到的标准 HTML 标记。如果你需要 HTML 方面的参考数据，我推荐《HTML and XHTML: The Definitive Guide》（Chuck Musciano 与 Bill Kennedy 合著，O'Reilly 出版）。但即使你已经是 HTML 大师，仍有可能这是你第一次接触到网站应用系统的领域，因此，对于驱动所有网站应用系统的 HTTP 协议，我会提供一个通盘的介绍。

再假设你已经熟悉 Java 程序设计，特别是 JavaBeans 技术与面向对象的概念。如果你不熟悉，那么必须先学好 Java 才能看这本书。关于 Java 的书太多了，像是《Head First Java》（Kathy Sierra 与 Bert Bates 合著，O'Reilly 出版）、《Learning Java》（Patrick Niemeyer 与 Jonathan Knudsen 合著，O'Reilly 出版）。

JSF 规范以 Java Servlet 为基础，并使用 JSP 来布置网页的版面。所以，本书也提供了这两项技术的介绍，其分量足以让你了解本书的范例。不过，在你进一步着手开发真实世界的大型网站应用系统之前，我建议你彻底搞懂 JSP 与 Servlet 的细节。我写的《JavaServer Pages》以及 Jason Hunter 与 William Crawford 合著的《Java Servlet Programming》（都是 O'Reilly 出版）是有关这两项内容的公认好书。

## 章节编排

本书共计十五章，六篇附录。

### 第一章 JavaServer Faces 概论

解释 JSF 于网站应用系统的定位以及与其他同类技术的比较。

### 第二章 JSF 开发程序

对于使用 JSF 开发网站应用系统的程序，提供一个整体概念。

### 第三章 准备 JSF 环境

示范如何安装 Tomcat 5 Web container 与本书范例，包括 JSF 参考实现。

### 第四章 Servlet 与 JavaServer Pages 概论

描述 HTTP、Servlet 与 JSP 的基本概念。这些都是了解 JSF 所必需的基础知识。

### 第五章 事务逻辑与身份验证

本书大部分的范例片段都是来自同一个网站应用系统样本，本章对此网站应用系统做个简介，并解说该应用系统所用到的事务逻辑类。

### 第六章 组件的创建与绘制

初窥 JSF 内层的风貌，着重介绍 API 与 JSP 层是如何创建、绘制组件的。

### 第七章 检验外界输入

示范如何使用 converter 与 validator 来验证用户输入的有效性，以及如何发展自己的 validator，并调整错误信息。

### 第八章 事件处理

解释如何处理各种不同类型的事件（如用户按下按钮、点击某链接、改变某输入组件的值……），以及这些事件如何影响 UI、如何触发后端程序。

### 第九章 浏览控制

介绍 JSF 的网页浏览功能，以及事件处理的结果如何控制下次要显示的网页。

### 第十章 表格数据的处理

提供显示、编辑表格数据的另类办法，包括如何处理大表格。

### 第十一章 国际化

解释国际化与地区化，Java 使应用程序国际化的有用功能，以及 JSF 如何发挥这方面的功能来支持多语种网站。

### 第十二章 精雕细琢

弥补前文未讨论到的内容，包括复合网页（由多个文件构成的一个网页）、与 Struts application framework 的集成、调试技巧等。

### 第十三章 开发自定义绘制器与其他外挂类

探讨如何为标准组件开发特殊的 JSF renderer, 借此提供替代的成形与输入鉴别能力; 讨论如何以自定义版本取代其他外挂类。

### 第十四章 开发自定义组件

示范开发 JSF 自定义组件的各种方式, 包括修改现成组件以及制作全新组件。

### 第十五章 开发自定义的表示层

解说 JSF 如何支持 JSP 以外的其他表示层技术, 以及如何开发类似 Tapestry (一个 open source 软件) 之类的特殊表示层。

### 附录一 标准 JSF 标记库

解说 core 和 HTML 这两套 JSF 标准标记库所提供的 JSP 自定义动作组件。

### 附录二 JSF EL 语法参考

提供 JSF EL 的语法与规则的说明。

### 附录三 标准 JSF 组件与 Render Kits

解说所有 JSF 标准组件类, 以及标准的绘制工具组。

### 附录四 基础 API 参考

说明所有 JSF 基础类, 包括 converters、validators 以及错误信息。

### 附录五 JSF 配置文件参考

解说 JSF 配置文件的所有声明元素。

### 附录六 网站应用系统的结构与配置描述文件

解说网站应用系统的标准结构, 以及配置描述文件的所有声明元素。

所有章节都是互相关联的, 所以建立顺序阅读。

## 关于范例

JSF-based 网站应用系统所需的大部分技术, 本书都提供了具体范例。这些范例除了印在书本上之外, 还可从 O'Reilly 的网站自由下载 (<http://www.oreilly.com/catalog/jsrfaces/>)。此外, 作者自己的网站 (<http://www.hansbergsten.com/>) 也提供范例下载与回复技术问题的服务。

虽然 JSF 规范只要求到 Servlet 2.3 与 JSP 1.2, 但是本书范例使用了 Servlet 2.4 与 JSP 2.0 规范的功能 (2003 年 11 月之前的最新版本), 因为这样可让我的日子好过些。有鉴于此, 我希望读者的系统中最好安装新版规范的实现。所有范例都已经过完整的测试, 以下是我的测试环境: JSF1.0 正式参考实现, Sun 的 Java 2 SDK, Standard Edition 1.4.2

版 (Windows ME、2000、Red Hat Linux 7.2), BlackDown Java 1.4.1-01 (Gentoo Linux/Sparc64), Tomcat 5.0.19。

## 排版约定

斜体字 (*Italic*)

表示路径、文件名、程序名、编译器及其他选项, 新名词, 网址 (域名及 URL)。

等宽字 (*Constant Width*)

用于表示应该以原貌出现在 JSP 网页或 Java 程序中的东西, 包括关键字、数据类型、常量、方法名称、变量名称、类名、接口名称、命令以及所有 JSP 和 Java 程序列表、HTML 网页内容、标记、属性。

等宽斜体字 (*Constant Width Italic*)

用于表示语法中应该被替换成实际值的部分。

等宽黑体字 (*Constant Width Bold*)

在用户界面中, 凸显人工输入的部分。在程序列表中, 强调与当时说明主题相关的部分。

## 建议与评论

本书的内容都经过测试, 尽管我们做了最大的努力, 但错误和疏忽仍然是在所难免的。如果你发现有什么错误, 或者是对将来的版本有什么建议, 请通过下面的地址告诉我们:

美国:

O'Reilly Media, Inc.  
1005 Gravenstein Highway North  
Sebastopol, CA 95472

中国:

100080 北京市海淀区知春路 49 号希格玛公寓 B 座 809 室  
奥莱理软件 (北京) 有限公司

与本书有关的在线信息 (譬如, 勘误、相关的链接):

<http://www.oreilly.com/catalog/jsvrfaces/>

询问技术问题或对本书的评论, 请发电子邮件到:

*info@mail.oreilly.com.cn*

*bookquestions@oreilly.com*

最后，您可以在以下站点找到我们：

*<http://www.oreilly.com>*

*<http://www.oreilly.com.cn>*

## 致谢

没有 JSF 规范就没有本书存在的必要。JSF 规范是 Java Community Process (JCP) 参与者努力工作的成果，这团队里有大公司的代表、小公司的代表，以及许多像我这样的个人。我要感谢积极参与规范制定工作的 JSF Expert Group (EG) 的所有成员，特别是 Oracle 公司的 Adam Winer —— 没有你，我真不知道我们会做出什么东西来。Kudos 是规范说明书的推手。Ed Burns 细心记录并追查我们必须解决的所有议题。当我开始糊涂时，幸好有 Craig McClanahan 的耐心。

我要感谢校阅本书并修正错误的高手：Adam Winer、Craig McClanahan、Steve Bang、Pierre Delisle 以及 Janne Andersson，你们的提议肯定改善了本书呈现的内容；感谢 Dan Malks 将我的设计构想融入他的《Core J2EE Patterns: Best Practices and Design Strategies》(Prentice Hall 出版) 中所描述的 J2EE 典范，很可惜，现在已经来不及在本书纳入 J2EE 设计典范的参考内容，只能期待未来改版时有机会。

将我的话变成真正的书需要许多人的努力。我要感谢 O'Reilly 公司里的每个人，为他们的辛勤工作，并给我写作机会，特别是我的编辑 —— Mike Loukides，给我深思熟虑的建议；Colleen Gorman，深厚的文字修润功夫；Robert Romano，将我的简单草图化为艺术。

感谢我的家人与世界各地的朋友对我的支持与鼓励，以及周五夜晚在 Vesica Pisces 的 Lighthouse 聚会的朋友 (<http://www.vp-unleashed.com/> —— Kelly、Brian、Adam 和 Nuss)、Rob、Dave、Becky、James、Tamie 以及其他死党。干杯！

—— Hans Bergsten

# JavaServer Faces

## 概论

近年来, Java 已成为开发网站应用系统 (Web Application) 的领导技术。Servlet 与 JSP 被成功地运用于无数的应用系统, 让开发者能写出有弹性的网页用户界面以及稳健的后台程序。然而, 随着网站应用系统越来越复杂, 开发者开始怀念昔日使用 GUI framework 的美好时光。传统的 GUI 开发环境提供了许多方便好用的 UI 组件, 而且事件驱动模型 (event-driven model) 也比较符合网站应用系统的结构。问题在于 HTTP 是一种简单、粗糙的 request/response 通信协议, 它的 stateless 本质 (译注 1) 迫使开发者必须亲自处理原本可由 AWT/Swing (Java 的标准 GUI framework) 默默处理掉的琐碎细节, 而这些细节都是 Servlet 与 JSP 无法解决的。

为了更容易开发出精致的前台用户界面, 开始有人模拟传统的 GUI framework, 其中有些是 open source 计划, 也有一些是商业产品, 比较著名的例子包括 Enhydra Barracuda、Apache Tapestry、Oracle UIX 以及 Sun JATO。2001 年春天, 百家争鸣的情况开始有了结合的迹象, 各家公司派代表成立了一个 Java Community Process (JCP) 团队 (其中包括笔者), 着手研发一套能涵括各家 framework 的标准解决方案。2004 年 3 月, JCP 正式发表首次的合作成果 —— JavaServer Faces 1.0。

## 何谓 JavaServer Faces

JavaServer Faces (JSF) 定义了 UI 组件的模型, 以及一套严谨的处理作业流程, 组件模型与处理作业流程的紧密结合, 可大幅简化网站应用系统的开发工作。JSF 实现了以下几个梦想:

---

译注 1: stateless 是指 HTTP 没有 session-tracking 的能力。因为 HTTP 将每次 request 都视为独立事件, 所以服务器端无法辨认不同 request 之间的关联性 (无法判断是否来自同一个客户端、是否与前次 request 有关等)。

- Java程序员不必顾虑HTTP的细节,而可以按照原本熟悉的事件驱动模型来设计后台系统,并通过一个能担保数据类型无误的数据传递接口将后台系统与前台界面结合在一起。
- 没有程序设计知识的网页美工人员也可以使用程序员提供的组件来布置用户界面的“观感”(look & feel),这些组件已封装了所有互动操作的逻辑,将网页本身必须包含的程序逻辑降到最低。
- 厂商可以研制更好用、更一致的前后台开发工具。

更精确地说,JSF规范综合了各家厂商现有产品的精粹。它定义了一组用户界面组件(User Interface component, 后文简称“UI 组件”),以及一组标准的应用程序接口(Application Programming Interface, API)。所有UI组件都可以直接用在网页里,而且大部分组件几乎都是HTML form系列标记的翻版。API可用来扩充原有的标准组件,也可以开发全新的组件。连接在组件上的validators,不仅能够检验用户输入的数据,还能自动将输入数据传递给应用对象。每当用户点击链接或按下按钮,诸如此类的操作便会触发event handler,而handler可以改变其他组件的状态,或是运行某段后台程序。借由一个可插入的navigation handler, event handler可以控制接下来要显示哪一个网页。

大多数网站应用系统选择的标记语言是HTML,本书的范例自然也不例外。不过,JSF并未硬性规定非得使用HTML或任何其他标记语言不可。实际会传送什么标记给客户端,其实决定于绘制器(renderer)。Renderer独立于UI组件之外,同一个UI组件,只要搭配不同的renderer,便可产生迥然不同的输出(例如输出HTML或WML)。如果你熟悉Swing,不妨将renderer想象成“可插入的观感组合”(pluggable look & feel, PLAF)。

JSF对于表示层(presentation layer, 实际提供UI页面的软件层)的包容性非常宽大。原则上,JSF 1.0只规定JavaServer Pages (JSP)为必须支持的表示层技术;在JSP网页中,JSF组件是以“JSP自定义动作标记”(也称为“自定义标记”)的形式出现。除了JSP之外,JSF API也支持其他表示层技术。举例来说,你可以使用纯Java程序来制作JSF组件(很像是在制作新的Swing UI),也可以将JSF组件嵌入HTML模板中的特殊节点(这是Barracuda/XMLC与Tapestry所采用的技术)。本书大部分范例都是以JSP为表示层的,因为JSP是JSF 1.0中唯一完整规范的技术,但是本节也会示范如何搭配其他表示层。

如果你已经用Java开发后台系统好一阵子了,想必你听说过(或已经使用过)“应用程序框架”(application framework)之类的东西,像是Apache Struts (<http://jakarta.apache.org/struts/>)、Maverick (<http://mav.sourceforge.net/>)等。JSF能够搭配这类框架,虽然在功能上有部分重叠(例如,同样都支持validation与navigation)。在



第十二章介绍 Struts 时, 你会见到 JSF API 与组件带来了怎样的额外选择, 届时你会见识到 JSF 如何搭配其他热门的 application framework。

## JSF 与传统技术的比较

JSF 将原本运用于 GUI 应用程序的组件模型, 引入了网站应用系统的开发。让我们来看看, 相较于传统的模型, 新模型带来了哪些优点。

### 减少 UI 模板中的程序代码

Java 网站应用系统的用户界面, 原本是以一组 JSP 网页构成 (或 Velocity、FreeMarker 之类的模板引擎), 在处理网页时, 静态内容 (呈现版面的文字与 HTML 标记) 与产生动态内容的标记混杂在一起。这种手法的问题在于网页势必要含有处理 UI 状态的逻辑。举例来说, 具有输入字段的网页, 需要含有能够维护目前输入状态 (如复选框、已选择的项目……) 的程序代码。

使用 JSF 模型时, 若以 JSP 为表示层技术, 则 JSF 组件是以特殊标记 (也就是 “JSP 自定义动作标记”) 的形式出现在网页中。下面的网页程序代码片段, 以一组 HTML 复选框 (checkbox) 显示访客的饮食偏好:

```
...
<h:form>
  <table>
    ...
    <tr>
      <td>Favorite Foods:</td>
      <td>
        <h:selectManyCheckbox value="{cust.foodSelections}">
          <f:selectItem itemValue="z" itemLabel="Pizza" />
          <f:selectItem itemValue="p" itemLabel="Pasta" />
          <f:selectItem itemValue="c" itemLabel="Chinese" />
        </h:selectManyCheckbox>
      </td>
    </tr>
    ...
  </table>
</h:form>
...
```

现阶段还不必考虑细节, 只要注意在这段网页程序代码中没有循环, 也没有条件判断。因为这些程序逻辑都已经封装在 <h:selectManyCheckbox> 与 <f:selectItem> 这两个特殊标记所代表的 JSF 组件里。用户提交此窗体后, JSF framework 将目前的选择清单存放在服务器上的一个应用对象 (即 value 属性所指的 #{cust.foodSelection}) 中。