

11

SUN核心技术丛书
Core JavaServer Faces



JavaServer Faces

核心编程

[美] David Geary 著
Cay Horstmann
王 军 马振萍 等译



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

<http://www.phei.com.cn>



内 容 简 介

JavaServer Faces (JSF) 在Java的Web编程中已经被认为是下一个重大的事件。通过JSF, 你可以在网页上使用Web组件来捕获用户行为产生的事件。JavaServer Faces将快速成为标准的Web应用程序框架。本书可帮助你掌握这个功能强大的技术。

本书适合用户界面设计的Web开发者, 以及实现Web应用程序中可重用组件的程序员。这与官方的JSF规范形成鲜明的对比, 官方的JSF规范主要适合于框架实现者。



Authorized translation from the English language edition, entitled CORE JAVASERVER FACES, 1st Edition, 0131463055 by GEARY DAVID; HORSTMANN CAY, published by Pearson Education, Inc., publishing as Prentice Hall PTR/Sun Microsystems Press, Copyright © 2004 by Sun Microsystems.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc. CHINESE SIMPLIFIED language edition published by PUBLISHING HOUSE OF ELECTRONICS INDUSTRY & BEIJING MEDIA ELECTRONIC INFORMATION CO. LTD, Copyright © 2005

本书英文版由Pearson教育公司出版, Pearson教育公司已将简体中文独家版权授予中国电子工业出版社及北京美迪亚电子信息有限公司。未经许可, 不得以任何形式和手段复制或抄袭本书内容。

版权贸易合同登记号 图字: 01-2005-1638

图书在版编目(CIP)数据

JavaServer Faces核心编程/(美)吉瑞(Geary, D)等著;王军等译.一北京:电子工业出版社,2005.4
(SUN核心技术丛书)

书名原文:Core JavaServer Faces

ISBN 7-121-01034-8

I. J… II. ①吉… ②王… III. JAVA语言—程序设计 IV. TP312

中国版本图书馆CIP数据核字(2005)第021380号

责任编辑:吴源朱巍

印刷:北京天竺颖华印刷厂

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

北京市海淀区翠微东里甲2号 邮编:100036

经销:各地新华书店

开本:787×1092 1/16 印张:35.125 字数:890千字

印次:2005年4月第1次印刷

定价:55.00元

凡购买电子工业出版社的图书,如有缺损问题,请向购买书店调换,若书店售缺,请与本社发行部联系。联系电话:010-68279077。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。

译 者 序

在Java的Web编程中，JavaServer Faces (JSF) 已经被认为是下一个重大的技术。通过JSF，你可以在网页上使用Web组件，来捕获用户行为产生的事件。JavaServer Faces技术为开发基于网络用户界面的Java开发者提供了标准的编程接口API以及标签库。

本书前半部分，一直到第6章，主要讨论JSF标签。从第6章的最后一节开始，我们将重点介绍JSF编程，讨论如何执行高级的任务，以及如何扩展JSF框架。最后一章以“如何实现”的形式来回答一些常见的问题。

本书在翻译过程中力求忠实于原著，但由于水平和时间有限，译作中的不当与疏漏在所难免，恳请广大专家和读者批评指正。参加本书翻译工作的人员有王军、马振萍、刘芳、刘斌、王健、周涛、王珏、张吉祥、郭颖浩、陈旌、何文、陈雪松、方勇、张雯静、宋爱华和周小暄等，在此向他们深表感谢！

致 谢

首先也是最重要的，我们非常感谢Greg Doench, Prentice Hall的编辑，他带领我们完成了本书，尽管经历了很多艰辛，但是他从没有灰心。感谢Mary Lou Nohr编辑了本书的手稿。而且，感谢Patti Guerrieri的产品制作。我们非常感谢完成了大量工作、发现了很多错误并对多个版本的手稿提出改善建议的批评家：

- Larry Brown, Johns Hopkins University
- Frank Cohen, PushToTest
- Rob Gordon, Crooked Furrow Farm
- Marty Hall, Core Java Servlets and JavaServer Pages的作者
- Jeff Markham, Markham Software Company
- Angus McIntyre, IBM的合作伙伴
- John Muchow, Core J2ME的作者
- Sergei Smirnov, Exadel JSF Studio的主要设计师
- Roman Smolgovsky, Flytecomm
- Stephen Stelting, Sun Microsystems, Inc.
- Christopher Taylor, Nanshu Densetsu
- Michael Yuan, Enterprise J2ME的作者

最后，感谢一直支持我们的家人和朋友。

前 言

当在2002 Java One会议上听到JavaServer™ Faces (JSF) 时, 我们非常兴奋。我们在客户端编程方面具有很多经验, 如David编写的“Graphic Java”和Cay编写的“Core Java”, 这两本书都是由Sun Microsystems Press出版的。当我们第一次使用servlet和JavaServer Pages (JSP) 来进行Web编程时, 我们发现它不是很直观并且有点乏味。JavaServer Faces在Web应用程序之前提供了友好的框架, 允许程序员考虑文本字段和菜单, 而无需操心页面翻转和请求参数。当我们提出写一本书时, 出版商建议我们联合起来编写一本有关该技术的Sun Microsystems Press书籍。

这本书花费了JSF专家组 (David也是其成员之一) 近两年的时间来了解JSF 1.0规范和参考实现 (其后发布维护版本称为JSF 1.1, 它并没有引入新的功能)。这个版本实现了很多最初的承诺。通过将组件放到一个表单中并将它们链接到Java对象, 以此设计Web用户界面, 而无需编写任何代码。这个框架支持工具, 现在已经出现支持拖拉GUI的创建器。该框架可以扩展——不局限于使用HTML组件的标准集, 甚至可以使用完全不同的呈现技术来支持无线设备。最后, JSF支持表示和业务逻辑的分离、导航、与外部服务的连接和配置管理。

当然, JSF的最初版本还不是很完美, 有些API很难使用。com.corejsf.util包提供了一些工具类, 可以减少程序员的工作。而且, 组件要比所期望的少很多。虽然JSF具有功能强大和便捷的数据表格组件, 但是有些有用的组件 (例如页嵌式面板、滚动条、文件上传, 等等) 都没有包含在内。在本书中, 我们介绍了如何实现这些功能。当然, 我们期望JSF的下一个版本能够弥补这些缺点。

我们仍然对JSF很感兴趣, 并且我们希望能够与读者分享这种快乐: 这种技术能使读者成为更有效的Web应用程序开发者。

关于本书

本书适合用户界面设计的Web开发者, 以及实现Web应用程序中可重用组件的程序员。这与官方的JSF规范形成鲜明的对比, 官方的JSF规范主要适合于框架实现者。

本书前半部分, 一直到第6章, 主要讨论JSF标签。这些标签类似于HTML表单标签。它们是JSF用户界面的基本构件。不需要任何编程技术就可以使用这些标签。我们认为读者只需要具备Web页面的基本HTML技术和业务逻辑的标准Java编程技术。

本书的前半部分包含下列内容:

- 建立编程环境 (第1章)
- 连接JSF标签到应用程序逻辑 (第2章)
- 在页面之间导航 (第3章)

- 使用标准的JSF标签（第4章和第5章）
- 转换和验证输入（第6章）

从第6章的最后一节开始，我们将重点介绍JSF编程，讨论如何执行高级的任务，以及如何扩展JSF框架。下面是第二部分的主要内容：

- 实现自定义转换器和验证器（第6章）
- 事件处理（第7章）
- 在多个页面之间包含公共内容（第8章）
- 实现自定义组件（第9章）
- 连接到数据库和其他外部服务（第10章）
- 支持无线客户端（第11章）

本书最后一章以“如何实现”的形式来回答一些常见的问题。我们鼓励读者在熟悉了JSF的基本知识之后浏览一下最后一章。在该章中，有很多关于调试和日志记录方面的说明，并且还给出了JSF 1.0/1.1中所没有提供功能的实现细节和代码，例如文件上传、弹出菜单和大型表格的页面调度器组件。

JSF是建立在servlet和JSP之上的，但是从JSF开发者的观点看来，这些技术只是形成了低层系统。虽然它不影响熟悉其他Web技术（例如servlet、JSP或Struts），但是我们假设你具有了这样的知识。

所需要的软件

本书所需要的所有软件都可以免费获得。读者需要从Sun Microsystems获得Java Software Development Kit，此外还需要一个servlet容器（例如Tomcat）和JSF实现，例如Sun的参考实现。该软件在Linux、Mac OS X、Solaris和Windows上的运行是相同的。我们在Linux和Mac OS X上使用J2SE 1.4.2和Tomcat 5.0.19来开发本书中的代码示例。

我们还期望不久的将来能够有集成的开发环境可用。

Web支持

本书的Web页面是<http://corejsf.com>，其中包含：

- 本书所有示例的源代码
- 我们认为浏览要比打印出来更有效的有用参考资料
- 本书中已经发现的错误
- 提交更正和建议的表单

目 录

1	开始	2
	为什么要使用JavaServer Faces	3
	软件安装	4
	一个简单的示例	6
	示例应用程序分析	11
	可视化开发环境	17
	JSF框架服务	19
	内幕	21
	使用Ant自动化创建过程	25
2	受管理的Bean	30
	Bean的定义	31
	消息包	34
	示例程序	36
	Backing Bean	42
	Bean范围	43
	配置bean	44
	值绑定表达式的语法	50
3	导航	56
	静态导航	57
	动态导航	58
	高级导航问题	67
4	标准的JSF标签	72
	概述JSF核心标签	73
	概述JSF HTML标签	75
	表单	81
	文本字段和文本区域	85
	按钮和链接	94
	选择标签	104

消息	128
面板	132
5 数据表格	138
数据表格标签	139
一个简单的表格	140
h:dataTable属性	143
标题和脚注	144
表格单元中的JSF组件	146
编辑表格单元	150
行和列的样式	153
表格模型	160
滚动技术	174
6 转换和验证	176
概述转换和验证过程	177
使用标准转换器	178
使用标准的验证器	186
编写自定义转换器和验证器	190
实现自定义标签	212
7 事件处理	232
生命周期事件	233
值变换事件	235
动作事件	240
事件监听器标签	247
即时组件	249
阶段事件	252
实践	259
8 子视图和Tiles	268
共同布局	269
图书浏览器和图书馆	269
图书浏览器	271
图书浏览器中的内容包含	280
图书馆	289

9	自定义组件	302
	使用类来实现自定义组件	304
	编码: 产生标记	308
	解码: 处理请求值	311
	实现自定义组件标签	316
	再次回到微调控制器程序	325
	编码JavaScript来避免服务器往返	342
	使用子组件和Facets	346
10	外部服务	372
	访问数据库	373
	发送SQL语句	373
	使用LDAP进行身份验证	387
	管理配置信息	396
	使用Web服务	424
11	无线客户端	436
	移动客户端的呈现技术	437
	MIDP基础	438
	移动通讯和控制流	445
	战舰游戏	455
12	如何实现	486
	Web用户界面设计	487
	验证	521
	编程	536
	调试和记录日志	546

Core JavaServer Faces

JavaServer Faces

核心编程

〔美〕 David Geary 著
Cay Horstmann

王 军 马振萍 等译

電子工業出版社

Publishing House of Electronics Industry

北京 • BEIJING

开 始

本章内容包括:

- 为什么要使用JavaServer Faces
- 软件安装
- 一个简单的示例
- 应用程序示例分析
- 可视化开发环境
- JSF框架服务
- 内幕
- 使用Ant自动化创建过程

为什么要使用JavaServer Faces

从招聘网站上发布的职位招聘信息可以看出，当前用于开发Web应用程序的两种技术是：

- “快速开发”风格，使用可视化开发环境，例如Microsoft ASP.NET。
- “硬核编码”风格，编写大量支持高性能后端的代码，例如J2EE（即Java 2企业版本）。

在编写本书的时候，开发小组面临两难的选择。J2EE是一种很吸引人的平台，它高度可扩展，可以移植到多个平台，而且获得了很多厂商的支持。另一方面，ASP.NET无需乏味的编程就可以很容易地创建漂亮的用户界面。对于程序员来说，当然希望两者都兼有：高性能的后端和容易编程的用户界面。

JavaServer Faces的目的就是将快速用户界面开发引入到服务器端Java。

如果用户熟悉客户端Java开发，可以将JSF认为是“适用于服务器端应用程序的Swing”。如果以前使用过JavaServer Pages（JSP），则可发现JSF提供了很多JSP开发人员必须自己实现的应用。如果熟悉服务器端框架，例如Struts，则可发现JSP使用的是类似的结构。



说明：要学习本书，不必熟知Swing、JSP或Struts等内容。读者只需熟悉Java和HTML。

JSP具有下列部件：

- 一套预制的UI组件
- 事件驱动的编程模型
- 使得第三方开发者能够支持其他组件的组件模型

JSP提供了事件处理和组件构成所需要的所有代码。很幸运的是，应用程序员可以忽略这些细节，而专注于应用程序的逻辑。

要完全实现JSP的目标，需要生成JSF应用程序的集成开发环境。在编写本章的时候，已经开始开发这些IDE。为此，我们首先将介绍如何手动编写JSF应用程序。在阅读本章中介绍的指令时，可以认为这些步骤在将来是自动完成的。

软件安装

要开始学习JSF，需要安装下列软件包：

- Java SDK 1.4.1或更高版本 (<http://java.sun.com/j2se>)
- Tomcat servlet容器 (<http://jakarta.apache.org/tomcat/>)
- JavaServer Faces参考实现 (<http://java.sun.com/j2ee/javaserverfaces>)
- 本书的示例代码，可以从<http://corejsf.com>上获得

我们假设用户已经安装了Java SDK并熟悉SDK工具。有关Java SDK的更多信息，可以参考Horstmann & Cornell, Core Java, Sun Microsystems Press, 2003。

在本章中，我们将介绍如何使用JSF和Tomcat 5。Tomcat是一种servlet容器：一种用于服务Web页面和执行servlet的程序——处理Web请求的Java程序。JavaServer Faces建立在servlet技术上，但是不必知道有关使用servlet来建立JSP应用程序的知识。下载和解压Tomcat、JSF参考实现和示例代码，将它们保存到某个目录中（一般来说，路径名称之间最好避免使用空格，例如c:\Program Files）。

在本书的其他章节中，我们将这些目录指定为tomcat、jsf和corejsfexamples。例如，我们提及tomcat/conf/server.xml，则需要查看像usr/local/jakarta-tomcat-5.0.19/conf或c:\jakarta-tomcat-5.0.19\conf这样的目录。

如果使用Windows操作系统，可以选择安装Tomcat的Windows安装包，来替代独立平台的ZIP文件。下载和执行这个安装程序。当提示选择安装目录时，我们建议使用c:\tomcat来替代默认的安装位置c:\Program Files目录。



说明：可以在任何支持Servlet 2.3和JSP 1.2的servlet容器中运行JSF应用程序。为了使得指令保持简单，我们只讨论Tomcat 5。如果喜欢其他servlet容器，只需按照标准的过程来在服务器上部署Web应用程序即可。



说明：还可以从其他资源上获得JSF，例如J2EE SDK (<http://java.sun.com/j2ee/download.html>) 或Java Web Services Development Pack (Java Web服务开发包) (<http://java.sun.com/webservices/webservicespack.html>)。但是，这些软件包还包含其他软件组件，并需要更复杂的配置。我们建议用户在熟悉了Tomcat和JSF参考实现之后，再使用其他更复杂的环境。

现在，准备开始。打开命令窗口并启动Tomcat。在UNIX/Linux上可以使用该命令：

```
tomcat/bin/startup.sh
```

（如图1-1所示）在Windows中启动

```
tomcat\bin\startup.bat
```

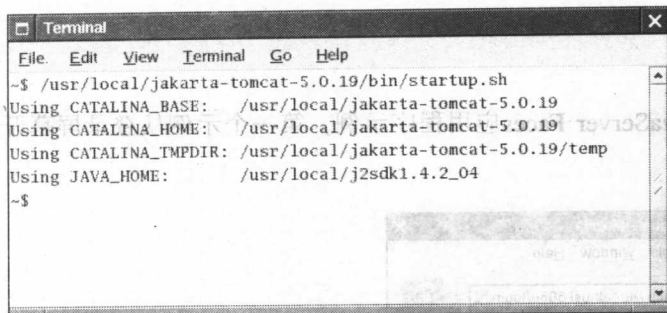


图1-1 启动Tomcat

或者，如果使用Windows安装程序来安装Tomcat，可以使用开始菜单中的“启动Tomcat”菜单项目。

要测试Tomcat是否能够正常运行，可以将浏览器指向<http://localhost:8080>。此时应该可以看到一个欢迎页面（如图1-2所示）。

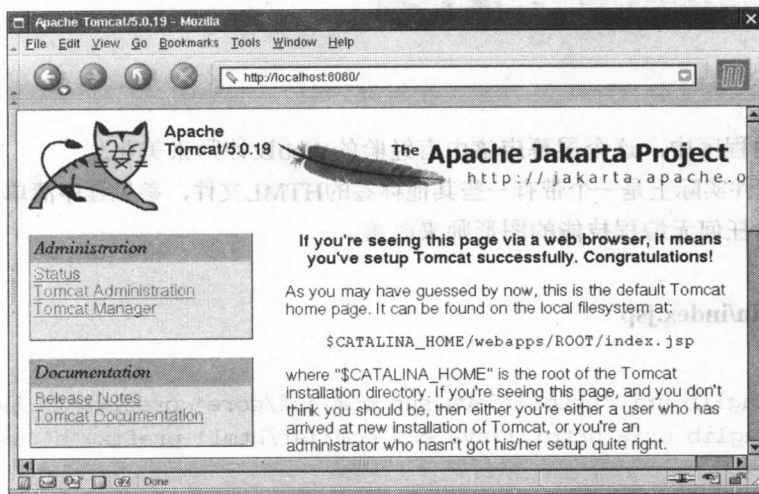


图1-2 Tomcat欢迎页面

使用下列命令来关闭Tomcat:

```
tomcat/bin/shutdown.sh
```

或者，在Windows中，使用下列命令:

```
tomcat\bin\shutdown.bat
```

或者，如果使用Windows安装程序来安装Tomcat，可以通过右键单击系统托盘上的Tomcat图标来关闭运行中的服务器。

一个简单的示例

下面将介绍一个简单的JavaServer Faces应用程序示例。第一个示例从登录屏幕开始，如图1-3所示。

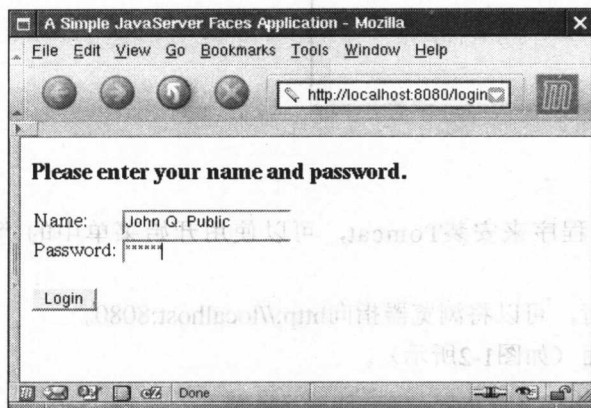


图1-3 登录屏幕

在实际的Web应用程序中，这个屏幕应该由有经验的Web设计师来美化。

描述登录屏幕的文件实际上是一个带有一些其他标签的HTML文件，参见程序清单1-1。其外观可以很容易地由任何无编程技能的图形师来改善。

程序清单1-1 login/index.jsp

```
1. <html>
2.   <%@ taglib uri="http://java.sun.com/jsf/core" prefix="f" %>
3.   <%@ taglib uri="http://java.sun.com/jsf/html" prefix="h" %>
4.
5.   <f:view>
6.     <head>
7.       <title>A Simple JavaServer Faces Application</title>
8.     </head>
9.     <body>
10.      <h:form>
11.        <h3>Please enter your name and password.</h3>
12.        <table>
13.          <tr>
14.            <td>Name:</td>
15.            <td>
16.              <h:inputText value="#{user.name}" />
17.            </td>
18.          </tr>
```

```
19.         <tr>
20.             <td>Password:</td>
21.         <td>
22.             <h:inputSecret value="#{user.password}"/>
23.         </td>
24.     </tr>
25. </table>
26. <p>
27.     <h:commandButton value="Login" action="login"/>
28. </p>
29. </h:form>
30. </body>
31. </f:view>
32. </html>
```

我们在本章后面将详细讨论这个文件的内容。现在，需要注意以下几个要点：

- 大量标签是标准的HTML标签：**body**，**table**，等等。
- 有些标签具有前缀，例如**f:view**和**h:inputText**。这些是JSF标签，两个taglib声明说明了JSF标签库。
- **h:inputText**、**h:inputSecret**和**h:commandButton**标签对应于文本字段、密码字段和提交按钮，如图1-3所示。
- 输入字段被连接到对象属性。例如，属性**value="#{user.name}"**告诉JSF实现将该文本字段与**user**对象的**name**属性相连接。我们将在本章后面详细讨论这些连接。

当用户输入名称和密码并且单击登录按钮时，将显示欢迎屏幕（如图1-4所示）。

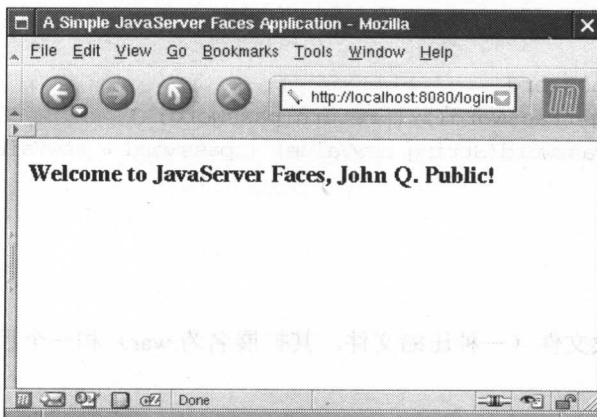


图1-4 欢迎屏幕

欢迎消息包含用户名称。当前我们忽略密码。

这个应用程序的目的只是演示用于创建JSF应用程序所必需的多个部分。

组成部分

我们的应用程序示例包含下列组成部分：

- 定义登录和欢迎屏幕的页面，我们称其为index.jsp和welcome.jsp。
- 管理用户数据的bean（在这里就是用户名和密码）。一个bean就是一个暴露属性的Java类，getter和setter方法通常遵循一个简单的命名规范。该代码位于UserBean.java文件中——参见程序清单1-2。注意该类包含在com.corejsf包中。
- 一个列出bean资源和导航规则的应用程序配置文件。默认时，这个文件被称为faces-config.xml。
- 使得servlet容器保持正确的其他必要文件：web.xml文件和一个index.html文件，该文件重定向用户到登录页面的正确URL。

多数高级的JSF应用程序都有相同的结构，但是它们可以包含其他Java类，例如事件驱动器、验证器和自定义组件。

程序清单1-2 login/WEB-INF/classes/com/corejsf/UserBean.java

```
1. package com.corejsf;
2.
3. public class UserBean {
4.     private String name;
5.     private String password;
6.
7.     // PROPERTY: name
8.     public String getName() { return name; }
9.     public void setName(String newValue) { name = newValue; }
10.
11.    // PROPERTY: password
12.    public String getPassword() { return password; }
13.    public void setPassword(String newValue) { password = newValue; }
14. }
```

目录结构

JSP应用程序被部署为一个WAR文件（一种压缩文件，其扩展名为.war）和一个目录结构，它遵循下列标准化的布局：

HTML和JSP文件：

WEB-INF/

configuration文件

classes/

class文件

lib/

library文件