

Core JavaServer™ Faces, Third Edition

PEARSON

JavaServer Faces

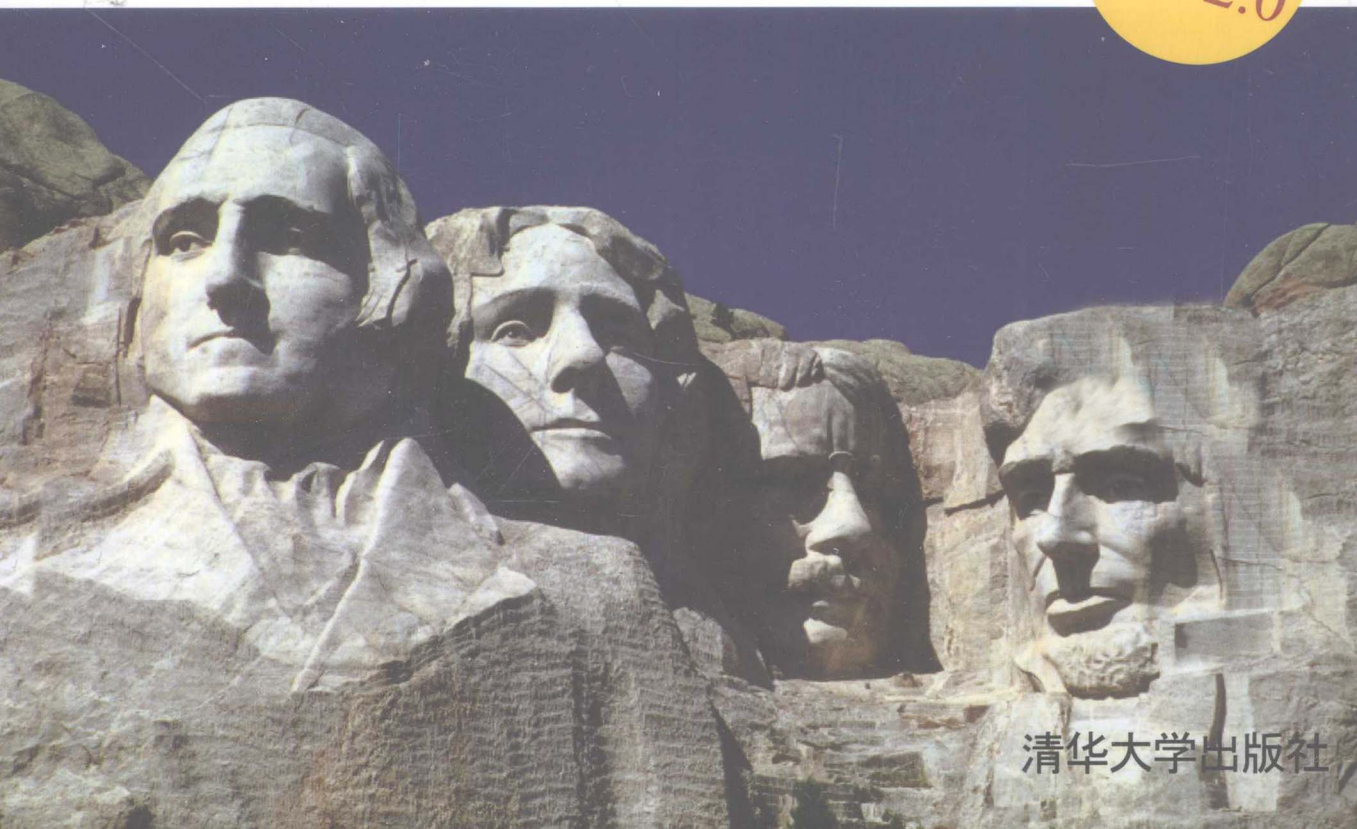
核心编程(第3版)



(美) David Geary
Cay Horstmann
王 超

著
译

更新至
JSF 2.0



清华大学出版社

JavaServer Faces核心编程

(第3版)

(美) David Geary 著
Cay Horstmann
王 超 译

清华大学出版社

北 京

Authorized translation from the English language edition, entitled Core JavaServer™ Faces, Third Edition, 978-0-13-701289-3 by David Geary, Cay Horstmann, published by Pearson Education, Inc, publishing as Prentice Hall, Copyright © 2010 Sun Microsystems, Inc.

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 PEARSON EDUCATION ASIA LTD., and TSINGHUA UNIVERSITY PRESS Copyright © 2011.

北京市版权局著作权合同登记号 图字: 01-2010-6800

本书封面贴有 Pearson Education(培生教育出版集团)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

JavaServer Faces 核心编程(第3版)/(美)吉尔里(Geary, D.), (美)霍斯特曼(Horstmann, C.) 著; 王超 译.

—北京: 清华大学出版社, 2011.10

书名原文: Core JavaServer™ Faces, Third Edition

ISBN 978-7-302-25839-1

I. J… II. ①吉… ②霍… ③王… III. Java 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(2011)第 103944 号

责任编辑: 王 军 韩宏志

装帧设计: 牛艳敏

责任校对: 胡雁翎

责任印制: 何 芊

出版发行: 清华大学出版社

地 址: 北京清华大学学研大厦 A 座

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

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 刷 者: 北京密云胶印厂

装 订 者: 北京市密云县京文制本装订厂

经 销: 全国新华书店

开 本: 185×260 印 张: 33.25 字 数: 809 千字

版 次: 2011 年 10 月第 1 版 印 次: 2011 年 10 月第 1 次印刷

印 数: 1~4000

定 价: 68.00 元

产品编号: 034126-01

致 谢

首先感谢 Prentice Hall 出版社的编辑 Greg Doench, 他指导我们完成了本项目, 尽管遇到不少困难, 但他从未灰心。非常感谢 Vanessa Moore 将我们的手稿变为这本引人入胜的书籍, 还有她的耐心和精益求精的态度也值得称道。

我们非常感谢本版本以及前两个版本的审阅人员, 他们的工作十分出色, 在各种草稿中查漏补缺, 并提出了改进意见。他们是:

- Gail Anderson, Anderson Software Group 有限公司
- Larry Brown, LMBrown.com 有限公司
- Damodar Chetty, Software Engineering Solution 有限公司
- Frank Cohen, PushToTest
- Brian Goetz, Sun Microsystems 有限公司
- Rob Gordon, Crooked Furrow Farm
- Marty Hall, *Core Servlets and JavaServer Pages™*, Second Edition (Prentice Hall, 2008)一书的作者
- Steven Haines, GeekCap 有限公司 CEO/创始人
- Charlie Hunt, Sun Microsystems 有限公司
- Jeff Langr, Langr Software Solutions 公司
- Jason Lee, Sun Microsystems 有限公司高级 Java 开发人员
- Bill Lewis, Tufts 大学
- Kito Mann, *JavaServer Faces in Action*(Manning, 2005)一书的作者, JSFCentral.com 创始人
- Jeff Markham, Markham 软件公司
- Angus McIntyre, IBM 公司
- John Muchow, *Core J2ME™*(Prentice Hall, 2001)的作者
- Dan Shellman, BearingPoint
- Sergei Smirnov, Exadel JSF Studio 的首席架构师
- Roman Smolgovsky, Flytecomm
- Stephen Stelting, Sun Microsystems 有限公司
- Christopher Taylor, Nanshu Densetsu
- Kim Topley, Keyboard Edge Limited

- Michael Yuan, *JBoss® Seam: Simplicity and Power Beyond Java™ EE*(Prentice Hall, 2007)的合著者之一

最后感谢一直默默支持我们的家人和朋友，他们帮助我们减轻压力并使本书最终得以完成。

前 言

David 和 Cay 曾参加 2002 年度 JavaOne 大会,当二人在大会上聆听到 JavaServer Faces (JSF)技术时都十分激动,心情久久不能平静。David 和 Cay 在客户端 Java 编程方面都拥有丰富经验;此前 David 曾撰写 *Graphic Java*TM, Cay 曾撰写 *Core Java*TM(均由 Sun Microsystems 出版社出版)。两人发现,用 servlet 和 JSP(JavaServer Pages)编写 Web 程序不够简明直观,令人感到厌烦。而 JSF 为 Web 应用程序提供了十分友好的界面,使程序员能够聚精会神地考虑文本字段和菜单,而不必围绕“页面跳转”和“请求参数”而苦思冥想。当 David 和 Cay 向 Sun Microsystems 出版社提出著书的想法时,该出版社立即给予热情回应,并建议二人合著这本 JSF 书籍。

JSF 专家组(David 便是该专家组的成员之一)于 2004 年发布了 JSF 1.0 规范和参考实现,随即很快发布了修订版本 1.1。2006 年发布的增量版本 JSF 1.2 对修订版本进行了诸多完善和整理,新增了大量便捷功能。

最初的 JSF 规范平凡无奇,与“完美”相去甚远,程序员在实际工作中根本没兴趣使用它所提供的用例。最初的规范未对 API 设计给予足够的重视,迫使程序员硬着头皮编写枯燥无味的复杂代码。对 GET 请求的支持也僵硬笨拙。错误处理方式令人痛苦不堪,开发人员们默默地诅咒着“炼狱般的堆栈跟踪”。

当时的 JSF 也并非一无是处:它具有高度的可扩展能力;而可扩展能力是框架开发人员十分看重的特点。这些人员开发了嵌入到 JSF 中的最前沿开源软件,如 Facelets、Ajax4jsf、Seam、JSF Templates、Pretty Faces、RichFaces 和 ICEFaces 等。

JSF 2.0 于 2009 年发布。JSF 2.0 吸取了开源框架的经验,将这些宝贵经验作为基石;上述框架几乎所有的创作者都加入 JSF 2 专家组中。因此 JSF 2.0 与 JSF 1.0 具有很大的不同,JSF 2.0 是由历经长期严峻考验而走向成熟的实用开源项目融合而成的。

JSF 2.0 的使用比 JSF 1.0 更简单,并更好地集成在 Java EE 技术栈中。JSF 2.0 几乎对 JSF 1.0 的每一部分都做了改进和完善。另外,JSF 2.0 规范目前支持 Ajax 和 REST 等新 Web 技术。

JSF 现已蜕变为卓越的服务器端 Java Web 框架,兑现了自己的大部分承诺。在设计 Web 用户界面时,程序员确实可在表单上放置组件,然后将它们与 Java 对象关联起来,未必需要结合使用代码和标签。JSF 的一个强大之处在于,它使用可扩展的组件模型,而且现有大量的第三方组件可供使用。灵活的框架设计使其能充分扩展,以便吸纳新技术。

JSF 是规范而非产品,用户可从多个供应商那里获得 JSF 实现、组件和工具,不必完全依赖于某个供应商。我们对 JSF 2.0 的问世感到欢欣鼓舞,我们也希望,当你通过学习这门技术来提高 Web 编程效率时,能与我们共享无尽的乐趣。

关于本书

本书面向负责实现用户界面和业务逻辑的 Web 开发人员。官方 JSF 规范文档内容丰富,堪称鸿篇巨制,主要供框架实现人员和撰写大篇幅书籍的作者参阅;本书的编排方式与官方 JSF 规范文档明显不同。JSF 建立在 servlet 基础之上,但在 JSF 开发人员看来,servlet 仅是底层技术而已。熟悉 servlet、JSP 或 Struts 等技术对学习本书有帮助,但不熟悉这些技术的读者也可以利用本书收到圆满的学习效果。

本书前半部分(即第 1 章~第 7 章)浓墨重彩地描述 JSF 标签。JSF 标签与 HTML 表单标签类似,是 JSF 用户界面的基本构建块。只要你掌握了基本 HTML 技能(网页设计范畴)和标准 Java 编程(应用程序逻辑范畴),就可以使用 JSF 标签构建 Web 应用程序。

本书前半部分包含以下主题:

- 设置编程环境(第 1 章)
- 将 JSF 标签连接到应用程序逻辑(第 2 章)
- 页面间导航(第 3 章)
- 使用标准 JSF 标签(第 4 章)
- 将 Facelets 标签用于模板(第 5 章)(新增内容)
- 数据表(第 6 章)
- 转换和验证输入(第 7 章)

从第 8 章起开始详细讨论 JSF 编程,包括如何执行高级任务以及如何扩展 JSF 框架。后半部分的主题如下:

- 事件处理(第 8 章)
- 构建复合组件——由较简单组件构成的具有复杂行为的可重用组件(第 9 章)(新增内容)
- Ajax(第 10 章)(新增内容)
- 实现自定义组件(第 11 章)
- 连接到数据库和其他外部服务(第 12 章)

本书的收官章节“问答”(第 13 章)旨在解答常见问题,我们建议你在熟悉掌握 JSF 的基础知识之后立即浏览一下该章的内容。该章包含有关调试和日志记录的有用说明,也提供了 JSF 未纳入的功能(如文件上传、弹出菜单和长表的分页器组件等)的实现细节和工作代码。

第 3 版中的所有章节都经过全面修订和完善,强调了 JSF 2.0 中的新功能和改进功能。第 3 版新增了第 5 章、第 9 章和第 10 章。

所需软件

本书需要的所有软件都可以免费获得。可使用支持 Java EE 6(如 GlassFish v3)的应用服务器,或带有 JSF 实现的 `servlet` 运行器(例如 Tomcat 6)。应用服务器软件可在 Linux、Mac OS X、Solaris 和 Windows 上运行。Eclipse 和 NetBeans 都对 GlassFish 或者 Tomcat 环境中的 JSF 开发提供了全面支持。

Web 支持

可以登录 <http://www.tupwk.com.cn/downpage> 来获得以下信息:

- 本书示例的源代码
- 一个 JSF Overview 文档,其中包含一些十分有用的信息

目 录

第 1 章 JSF 简介	1
1.1 选用 JSF 的原因	1
1.2 一个简单例子	2
1.2.1 组成部分	4
1.2.2 目录结构	5
1.2.3 构建 JSF 应用程序	6
1.2.4 部署 JSF 应用程序	7
1.3 JSF 开发环境	9
1.4 分析示例应用程序	11
1.4.1 bean	12
1.4.2 JSF 页面	13
1.4.3 servlet 配置	14
1.5 简单了解 Ajax JSF 2.0	17
1.6 JSF 框架服务	19
1.7 技术内幕	21
1.7.1 呈现页面	21
1.7.2 请求解码	22
1.7.3 生命周期	23
1.8 小结	25
第 2 章 受管 bean	27
2.1 bean 的定义	27
2.1.1 bean 属性	29
2.1.2 值表达式	31
2.1.3 backing bean	31
2.2 CDI bean CDI	32
2.3 消息包	33
2.3.1 具有可变部分的消息	35
2.3.2 应用程序的区域设置	36
2.4 示例应用程序	37
2.5 bean 作用域	42
2.5.1 会话作用域	43

2.5.2 请求作用域	44
2.5.3 应用程序作用域	45
2.5.4 对话作用域 CDI	45
2.5.5 视图作用域 JSF 2.0	46
2.5.6 自定义作用域 JSF 2.0	46
2.6 配置 bean	46
2.6.1 注入 CDI bean CDI	46
2.6.2 注入受管 bean JSF 2.0	47
2.6.3 bean 生命周期批注	48
2.6.4 使用 XML 配置受管 bean	48
2.7 表达式语言语法	52
2.7.1 左值和右值模式	53
2.7.2 使用方括号	53
2.7.3 映射和列表表达式	54
2.7.4 调用方法和函数 JSF 2.0	55
2.7.5 解析初项	55
2.7.6 复合表达式	57
2.7.7 方法表达式	57
2.7.8 方法表达式参数 JSF 2.0	58
2.8 小结	59
第 3 章 导航	61
3.1 静态导航	61
3.2 动态导航	62
3.2.1 将结果映射到视图 ID	63
3.2.2 JavaQuiz 应用程序	64
3.3 重定向	73
3.4 REST 式导航和书签	
URL JSF 2.0	75
3.4.1 视图参数	75
3.4.2 获取请求链接	76
3.4.3 指定请求参数	76

3.4.4 将书签链接添加到 Quiz 应用程序中	78	5.4.3 处理空白	168
3.5 高级导航规则	81	5.5 小结	168
3.5.1 通配符	82	第6章 数据表	169
3.5.2 使用 from-action	83	6.1 数据表标签——h:dataTable	169
3.5.3 条件导航情形 JSF 2.0	84	6.2 一个简单的表	170
3.5.4 动态目标视图 ID JSF 2.0	84	6.2.1 h:dataTable 特性	173
3.6 小结	84	6.2.2 h:column 特性	174
第4章 标准 JSF 标签	85	6.3 列头、列尾和标题	174
4.1 JSF 核心标签概述	86	6.4 样式	177
4.2 JSF HTML 标签概述	88	6.4.1 列样式	178
4.3 面板	96	6.4.2 行样式	179
4.4 标题、正文和表单标签	98	6.5 表中的 JSF 组件	181
4.5 文本字段和文本区域	103	6.6 编辑表格	183
4.5.1 隐藏字段	106	6.6.1 编辑表格单元格	184
4.5.2 使用文本字段和文本 区域	106	6.6.2 删除行 JSF 2.0	186
4.5.3 显示文本和图像	108	6.7 数据库表	189
4.6 按钮和链接	111	6.8 表模型	193
4.6.1 使用按钮	113	6.8.1 呈现行号	193
4.6.2 使用命令链接	117	6.8.2 查找所选行	194
4.7 选择标签	120	6.8.3 排序和过滤	195
4.7.1 复选框和单选按钮	123	6.9 滚动技术	202
4.7.2 菜单和列表框	125	6.9.1 用滚动条滚动	202
4.7.3 项目	127	6.9.2 用分页器小组件滚动	203
4.8 消息	143	6.10 小结	203
4.9 小结	148	第7章 转换和验证	205
第5章 Facelets JSF 2.0	149	7.1 转换和验证过程概述	205
5.1 Facelets 标签	149	7.2 使用标准转换器	206
5.2 Facelets 模板	150	7.2.1 数字和日期的转换	206
5.2.1 使用公共模板生成页面	152	7.2.2 转换错误	210
5.2.2 组织视图	155	7.2.3 一个完整的转换器例子	215
5.2.3 装饰器	160	7.3 使用标准验证器	218
5.2.4 参数	162	7.3.1 验证字符串长度和值 范围	218
5.3 自定义标签	162	7.3.2 检查必需值	219
5.4 其他标签	165	7.3.3 显示验证错误	220
5.4.1 <ui:debug>	165	7.3.4 跳过验证	222
5.4.2 <ui:remove>	167	7.3.5 一个完整的验证例子	222
		7.4 bean 验证 JSF 2.0	225

7.5 使用自定义转换器和验证器编程	230
7.5.1 实现自定义转换器类	230
7.5.2 指定转换器 JSF 2.0	233
7.5.3 报告转换错误	234
7.5.4 从资源包中获取 错误消息	235
7.5.5 自定义转换器示例程序	239
7.5.6 为转换器提供特性	242
7.5.7 实现自定义验证器类	243
7.5.8 注册自定义验证器	243
7.5.9 使用 bean 方法验证	247
7.5.10 验证多个组件之间的 关系	247
7.6 实现自定义转换器和验证器 标签	249
7.7 小结	255
第 8 章 事件处理	257
8.1 事件和 JSF 生命周期	258
8.2 值更改事件	259
8.3 动作事件	263
8.4 事件监听器标签	269
8.5 直接组件	270
8.5.1 使用直接输入组件	271
8.5.2 使用直接命令组件	272
8.6 从 UI 向服务器传递数据	273
8.6.1 方法表达式参数 JSF 2.0	274
8.6.2 f:param 标签	275
8.6.3 f:attribute 标签	275
8.6.4 f:setPropertyActionListener 标签	276
8.7 阶段事件	277
8.8 系统事件 JSF 2.0	278
8.8.1 多组件验证	280
8.8.2 在呈现视图之前做出 决定	281
8.9 合为一体	285
8.10 小结	291
第 9 章 复合组件	293
9.1 复合标签库	294
9.2 使用复合组件	295
9.3 实现复合组件	297
9.4 配置复合组件	298
9.5 特性类型	299
9.6 必需的特性和默认特性值	299
9.7 处理服务器端数据	301
9.8 复合组件的本地化	303
9.9 公开复合组件	304
9.10 facet	308
9.11 子组件	309
9.12 JavaScript	310
9.13 支持组件	315
9.14 在 JAR 中打包复合组件	323
9.15 小结	323
第 10 章 Ajax JSF 2.0	325
10.1 Ajax 和 JSF	326
10.2 JSF 生命周期和 Ajax	327
10.3 JSF Ajax 方法	328
10.4 f:ajax 标签	329
10.5 Ajax 组	331
10.6 Ajax 字段验证	333
10.7 Ajax 请求监视	334
10.8 JavaScript 名称空间	336
10.9 处理 Ajax 错误	338
10.10 Ajax 响应	338
10.11 JSF 2.0 JavaScript 库	341
10.12 传递额外的 Ajax 请求 参数	342
10.13 事件队列	344
10.14 合并事件	344
10.15 截获 jsf.ajax.request()	345
10.16 在复合组件中使用 Ajax	346
10.17 小结	351
第 11 章 自定义组件	353
11.1 实现组件类	354
11.2 编码：生成标记	357

11.3	解码: 处理请求值	359	12.3.3	使用受管 bean 和无状态 会话 bean	434
11.4	标签库描述符 JSF 2.0	365	12.3.4	有状态会话 bean CDI	437
11.5	使用外部呈现器	368	12.4	容器管理的身份验证和 授权	440
11.6	处理标签特性 JSF 2.0	371	12.5	发送邮件	450
11.6.1	支持值更改监听器	372	12.6	使用 Web 服务	456
11.6.2	支持方法表达式	373	12.7	小结	462
11.6.3	事件队列	374			
11.6.4	示例应用程序	375			
11.7	JavaScript 编码	382	第 13 章	问答	463
11.8	使用子组件和 facet	385	13.1	如何找到更多组件	463
11.8.1	处理 SelectItem 子组件	388	13.2	如何支持文件上传	464
11.8.2	处理 facet	389	13.3	如何显示图像映射	471
11.8.3	使用隐藏字段	390	13.4	如何在 JSF 页面中生成 二进制数据	473
11.9	保存和还原状态	396	13.5	如何以一次一个页面的方式 显示大型数据集	482
11.10	构建 Ajax 组件 JSF 2.0	400	13.6	如何生成一个弹出窗口	487
11.11	在自定义组件中实现独立的 Ajax	401	13.7	如何酌情显示和隐藏页面 部分	494
11.12	小结	410	13.8	如何自定义错误页面	495
第 12 章	外部服务	411	13.9	如何编写自己的客户端验证 标签	500
12.1	使用 JDBC 访问数据库	411	13.10	如何配置应用程序	506
12.1.1	发出 SQL 语句	411	13.11	如何扩展 JSF 表达式 语言	507
12.1.2	连接管理	413	13.12	如何在 JSF 表达式语言中 添加函数 JSF 2.0	510
12.1.3	消除连接泄漏问题	413	13.13	如何监视浏览器和服务器 之间的流量	511
12.1.4	使用预定义语句	415	13.14	如何调试问题页面	513
12.1.5	事务	416	13.15	当开发 JSF 应用程序时 如何使用测试工具	514
12.1.6	使用 Derby 数据库	417	13.16	如何结合使用 JSF 和 Scala	515
12.2	配置数据源	418	13.17	如何结合使用 JSF 和 Groovy	516
12.2.1	访问容器管理的资源	418	13.18	小结	517
12.2.2	在 GlassFish 中配置 数据库资源	419			
12.2.3	在 Tomcat 中配置数据库 资源	421			
12.2.4	一个完整的数据库 例子	422			
12.3	使用 Java 持久性架构	429			
12.3.1	JPA 简介	429			
12.3.2	在 Web 应用程序中使用 JPA	430			

第 1 章

JSF 简介

本章主题

- 选用 JSF 的原因
- 一个简单示例
- JSF 开发环境
- 分析示例应用程序
- 简单了解 Ajax
- JSF 框架服务
- 技术内幕

1.1 选用 JSF 的原因

如今，我们能在众多框架中选择一种来开发 Web 应用程序的用户界面。JavaServer Faces(JSF)是一种基于组件的框架。例如，如果想要显示一个带有行和列的表格，就不必在循环结构中为行和单元格生成 HTML 标签，只须将表格组件添加到页面中即可(读者如果熟悉客户端 Java 开发，可将 JSF 想象成“用于服务器端应用程序的 Swing”)。通过使用组件，读者能够在超越原始 HTML 的更高级别思考用户界面，也能够重用自己的应用程序组件并使用第三方组件集。还可以酌情使用一种可视化开发环境，在此环境中读者能够将组件拖放到表单上。

JSF 包含三个部分：

- 一套预置的 UI(用户界面)组件
- 一个事件驱动的编程模型
- 一个组件模型，它支持第三方开发商提供额外的组件

有些 JSF 组件很简单，例如输入字段和按钮。另一些则相当复杂，例如数据表和树。

JSF 包含了所有用于事件处理和组织组件的代码。应用程序编程人员完全可以忽略这些细节，而将精力投入到应用程序逻辑上。

JSF 并不是唯一一种基于组件的 Web 框架,但它是 Java EE 标准中的视图层。每个 Java EE 应用服务器都包含 JSF,而且用户可以很轻松地将其添加到独立的 Web 服务器(例如 Tomcat)中。

与大多数 Web 框架不同的是,JSF 是一种可通过多个方法来实现的标准框架。这使用户可以选择自己心仪的供应商,而不被某个供应商牢牢牵制。另一个优势是标准委员会已就 JSF 框架设计进行了深思熟虑,并在持续改进和更新 JSF。

本书将浓墨重彩地描述 JSF 2.0,JSF 2.0 在旧版本的基础上做了重大改进。JSF 2.0 使用起来比 JSF 1.x 要简单得多,并提供了强大的新功能,例如简单的 Ajax 集成乃至复合组件程序编写。

1.2 一个简单例子

下面分析一个简单的 JSF 应用程序的例子。这个例子由一个登录屏幕开始,如图 1-1 所示。



图 1-1 登录屏幕

描述登录屏幕的文件实际上是一个包含一些额外标签的 HTML 文件(如程序清单 1-1 所示)。即使不掌握任何编程技能的美工人员也可以轻松地修改这个登录屏幕的外观。

程序清单 1-1 login/web/index.xhtml

```
1. <?xml version="1.0" encoding="UTF-8"?>
2. <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
3. "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
4. <html xmlns="http://www.w3.org/1999/xhtml"
5.       xmlns:h="http://java.sun.com/jsf/html">
6.   <h:head>
7.     <title>Welcome</title>
8.   </h:head>
9.   <h:body>
10.    <h:form>
```

```
11.         <h3>Please enter your name and password.</h3>
12.         <table>
13.             <tr>
14.                 <td>Name:</td>
15.                 <td><h:inputText value="#{user.name}"/></td>
16.             </tr>
17.             <tr>
18.                 <td>Password:</td>
19.                 <td><h:inputSecret value="#{user.password}"/></td>
20.             </tr>
21.         </table>
22.         <p><h:commandButton value="Login" action="welcome"/></p>
23.     </h:form>
24. </h:body>
25. </html>
```

本章“JSF 页面”一节将详细讨论这个文件的内容。目前要注意以下几个要点：

- 大量标签是标准 HTML 标签，如 p、table 等。
- 有些标签是具有前缀，例如 h:head 和 h:inputText。这些都是 JSF 标签。Xmlns 特性声明了 JSF 名称空间。
- h:inputText、h:inputSecret 和 h:commandButton 标签对应于图 1-1 中的文本字段、密码字段和提交按钮。
- 输入字段被链接到对象属性。例如，特性 value="#{user.name}"告诉 JSF 实现将文本字段链接到 user 对象的 name 属性。后面的“bean”一节将详细讨论这个链接。

当用户输入了用户名和密码，然后单击 Login 按钮后，屏幕将显示 welcome.xhtml 文件，h:commandButton 标签的 action 特性指定了该操作(请参阅图 1-2 和程序清单 1-2)。



图 1-2 欢迎页面

JSF 2.0

注意

在 JSF 2.0 发布之前，只有将“导航规则”添加到 WEB-INF/faces-config.xml 文件中，才能指定当用户单击按钮时应当显示的页面。在 JSF 2.0 中，可直接在按钮的 action 特性中指定页面名称(读者仍然可以使用导航规则；见第 3 章)。

本应用程序的第二个 JSF 页面比第一个页面更简单(参见程序清单 1-2)。用#{user.name} 表达式显示在第一个页面设置的 user 对象的 name 属性。这里忽略了密码。

程序清单 1-2 login/web/welcome.xhtml

```
1. <?xml version="1.0" encoding="UTF-8"?>
2. <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
3. "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
4. <html xmlns="http://www.w3.org/1999/xhtml"
5.       xmlns:h="http://java.sun.com/jsf/html">
6.   <h:head>
7.     <title>Welcome</title>
8.   </h:head>
9.   <h:body>
10.    <h3>Welcome to JavaServer Faces, #{user.name}!</h3>
11.  </h:body>
12. </html>
```

这个应用程序尚无法生成赏心悦目的效果,但可以演示创建一个 JSF 应用程序所需的各个部分。

1.2.1 组成部分

本示例应用程序的组成部分如下:

- 定义登录和欢迎屏幕的页面。我们称之为 index.xhtml 和 welcome.xhtml。
- 管理用户数据(在本例中即用户名和密码)的 bean。bean 是一个公开属性的 Java 类,通常在属性后面紧跟的是遵守简单命名约定的 getter 和 setter 方法。代码位于 UserBean.java(参见程序清单 1-3)文件中。注意, @Named 或 @ManagedBean 批注指定了在 JSF 页面中引用该类的对象时使用的名称(出于兼容目的,目前使用两种替代批注来命名 bean。@Named 是符合 Java EE6 标准的应用服务器的最佳选择。@ManagedBean 则用于旧版应用服务器和独立的 servlet 运行器)。
- 保证应用服务器正常运行所需的配置文件: web.xml 和 beans.xml。



注意

在 JSF 2.0 发布之前,我们需要在 WEB-INF/faces-config.xml 文件中声明 bean。如今不再需要该操作,而且此应用程序也不需要 faces-config.xml 文件。

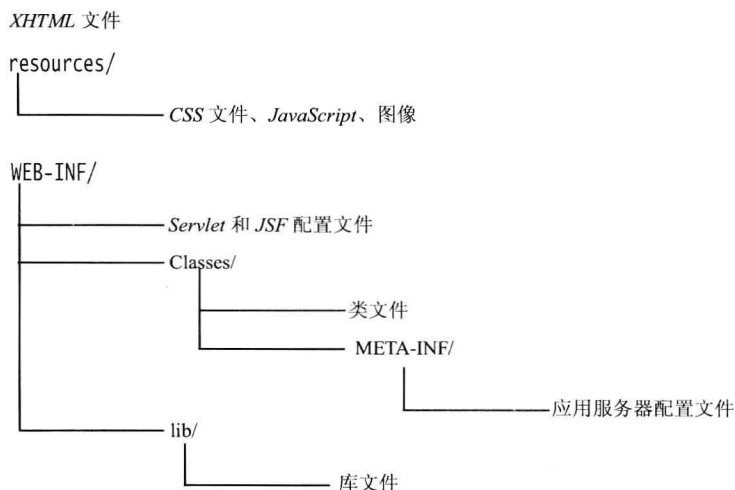
更高级的 JSF 应用程序的结构与此相同,但是它们可以包含额外的 Java 类,例如事件处理程序、验证器和自定义组件。可将额外的配置参数放在 WEB-INF/faces-config.xml 文件中,下一章将描述该文件。对一个简单应用程序而言,并不需要这一文件。

程序清单 1-3 login/src/java/com/corejsf/UserBean.java

```
1.package com.corejsf;
2.
3.import java.io.Serializable;
4.import javax.inject.Named;
5.    // or import javax.faces.bean.ManagedBean;
6.import javax.enterprise.context.SessionScoped;
7.    // or import javax.faces.bean.SessionScoped;
8.
9.@Named("user") // or @ManagedBean(name="user")
10.@SessionScoped
11.public class UserBean implements Serializable {
12.    private String name;
13.    private String password;
14.
15.    public String getName() { return name; }
16.    public void setName(String newValue) { name = newValue; }
17.
18.    public String getPassword() { return password; }
19.    public void setPassword(String newValue) { password = newValue; }
20. }
```

1.2.2 目录结构

JSF 应用程序作为一个 WAR 文件部署：一个带有扩展名为.war 的压缩文件，并且具有符合标准布局要求的目录结构：



例如，本示例应用程序的 WAR 文件的目录结构如图 1-3 所示。注意，UserBean 类位于 com.corejsf 程序包中。