

Core JavaServer Faces Second Edition

JavaServer

Faces

核心编程

(第2版)



(美) David Geary 著 马振晗 译
Cay Horstmann

清华大学出版社

JavaServer Faces 核心编程

(第 2 版)

David Geary
(美) Cay Horstmann 著

马振晗 译

清华大学出版社

北 京

Authorized translation from the English language edition, entitled Core JavaServer Faces Second Edition, 978-0-13-173886-7 by David Geary, Cay Horstmann, published by Pearson Education, Inc, publishing as Prentice Hall PTR, Copyright © 2007 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 © 2008.

北京市版权局著作权合同登记号 图字：01-2008-2288

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2版)/(美)吉瑞(Geary, D.), (美)霍斯特曼(Horstmann, G.)著; 马振晗译.
—北京: 清华大学出版社, 2008.10

书名原文: Core JavaServer Faces Second Edition

ISBN 978-7-302-18549-9

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

中国版本图书馆 CIP 数据核字(2008)第 140769 号

责任编辑: 王 军 李松峰

装帧设计: 孔祥丰

责任校对: 成凤进

责任印制: 孟凡玉

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: 清华大学印刷厂

装 订 者: 三河市金元印装有限公司

经 销: 全国新华书店

开 本: 185×260 印 张: 39.75 字 数: 968 千字

版 次: 2008 年 10 月第 1 版 印 次: 2008 年 10 月第 1 次印刷

印 数: 1~4000

定 价: 79.80 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话: (010)62770177 转 3103 产品编号: 021271-01

致 谢

首先,我们非常感谢 Greg Doench, Prentice Hall 公司的编辑,他带领我们完成了本书,尽管经历了许多艰辛,但是他从没有灰心。

我们非常感谢所有版本的审阅人员,他们都做得非常好,在各种手稿的草稿中找出了错误,并提出了改进意见。他们是:

- Gail Anderson, Anderson 软件公司
- Larry Brown, LMBrown.com, Inc.
- Frank Cohen, PushToTest
- Brian Goetz, Sun 微系统公司
- Rob Gordon, Crooked Furrow Farm
- Marty Hall, Core Java Servlets and JavaServer Pages 作者
- Charlie Hunt, Sun 微系统公司
- Jeff Langr, Langr Software Solutions
- Bill Lewis, Tufts 大学
- Jeff Markham, Markham 软件公司
- Angus McIntyre, IBM 公司
- John Muchow, Core J2ME 的作者
- Dan Shellman, BearingPoint
- Sergei Smirnov, Exadel JSF Studio 的首席架构师
- Roman Smolgovsky, Flytecomm
- Stephen Stelting, Sun 微系统公司
- Christopher Taylor, Nanshu Densetsu
- Kim Topley, Keyboard Edge Limited
- Michael Yuan, JBoss Seam: Simplicity and Power Beyond Java EE 的合著者

最后,感谢一直支持我们的家人和朋友,是他们减轻我们的压力才使本书最终得以完成。

前 言

2002年在JavaOne大会上第一次发布JavaServer Faces (JSF)的时候,这消息令我们非常兴奋。我们在客户端Java开发方面都具有丰富的经验,并已出版了David编写的*Graphic Java™*和Cay编写的*Core Java™*,由Sun Microsystems出版社出版。第一次使用servlet和JavaServer Pages (JSP)编写Web程序的时候,我们发现它不是很直观并且很乏味。现在JavaServer Faces在Web应用程序前端提供了友好的外观,使程序员只需考虑文本字段和菜单而不用为页面跳转和请求参数烦恼。当我们提出写一本JSF书的想法时,出版商迅速作出回应,建议我们一起就这一技术为Sun Microsystems出版社写本书。

直到2004年JSF专家组(David是成员之一)才发布JSF 1.0规范和参考实现。之后在很短时间内,又发布了漏洞修补版本1.1。在2006年,增量1.2版发布,添加了许多清理和方便的特性。

现在,JSF是服务器端最杰出的Java Web框架,它兑现了自己的大部分承诺。开发人员确实可以通过在表单上放置组件,将它们链接到Java对象来设计Web用户界面,而不必混合代码和标签。JSF的一个强大之处在于它是可拓展的组件模型,并且有大量的第三方组件可用。灵活的框架设计使它能够更好地扩展,并能容纳新技术,例如Ajax。框架的设计用于支持工具,最终产生了支持拖放的GUI构建器。最后,对于其他竞争技术,一旦深入表面,就立刻会跌下深崖;与这些竞争技术不同,JSF支持最困难的部分——表示和业务逻辑的分离、导航、连接外部服务和配置管理。

我们现在仍然因JSF而兴奋不已,而且也希望读者通过学习这个技术成为更加高效的Web应用程序开发人员,也能同我们一起分享这种快乐。

关于本书

本书面向Web开发人员(主要关注用户界面设计)和程序员(实现Web应用程序可重用的组件)。这与官方JSF规范形成了鲜明的对比,官方规范是一本厚重、庞大的文本文档,它的主要读者是框架实现人员和撰写书籍的作者。

本书的前半部分,直到第6章,着重讲解JSF标签。这些标签类似于HTML表单标签。它们是JSF用户界面的基本构建块。不是所有的程序都需要使用这些标签。我们假设网页仅需要基本的HTML技术,而业务逻辑需要标准的Java编程。

本书第一部分涵盖下面的主题:

- 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 和第 5 章)
- 转换和验证输入(第 6 章)

从第 7 章起, 我们开始详细讨论 JSF 编程, 包括如何完成高级任务和扩展 JSF 框架。

下面是第二部分的主题:

- 事件处理(第 7 章)
- 在多个页面中包含公共内容(第 8 章)
- 实现自定义组件、转换器和验证器(第 9 章)
- 连接到数据库和其他外部服务(第 10 章)
- Ajax(第 11 章)
- 开源技术, 重点放在 Facelets、Seam 和 Shale(第 12 章)

本书最后一章回答各种“如何实现……”的常见问题(第 13 章)。我们建议读者在熟悉了 JSF 的基础之后, 就快速地浏览这一章。其中包含关于调试和日志的有用说明, 也提供了 JSF 缺少功能的实现细节和工作代码, 例如文件上传、弹出菜单和长表的分页组件。

JSF 是在 servlet 和 JSP 基础上建立的, 但是从 JSF 开发者的角度看, 这些技术仅组成了低层系统。虽然熟悉其他网络技术例如 servlet、JSP 或 Struts 没有坏处, 但我们会假设读者不熟悉这些知识。

所需软件

本书需要的所有软件都可以免费获得。包括 Sun Microsystems 的 Java 软件开发工具包和支持 JSF 的应用程序服务器, 例如优秀的开源项目 GlassFish。该软件可以运行在 Linux、Mac OS X、Solaris 和 Windows 上。我们在 Linux 和 Mac OS X 上使用 Java 5 来开发本书中的示例代码。

如果读者正在寻找支持 JSF 的开发环境, 我们可以真诚地向您推荐免费软件 NetBeans IDE。较好支持 JSF 的 Eclipse 可以从销售 Eclipse 增强版的几个供应商那里得到。

网络支持

本书的支持网站是 <http://www.horstman.com/corejsf/>, 其中包含:

- 本书中所有例子的源代码(从 www.tupwk.com.cn 中也可下载源代码)
- 我们认为浏览形式比打印形式更有效的参考资料
- 本书和代码中已知的错误列表
- 用于提交更正和建议的表单

另外, 大家可以通过 wkservice@vip.163.com 向我们反馈建议和堪误信息。

目 录

第 1 章 入门	1
1.1 为什么要选择 JavaServer Faces	1
1.2 软件安装	2
1.3 一个简单的例子	4
1.3.1 组成部分	6
1.3.2 目录结构	7
1.3.3 构建说明	8
1.4 示例应用程序分析	9
1.4.1 Beans	9
1.4.2 JSF 页面	10
1.4.3 导航	13
1.4.4 Servlet 配置	15
1.4.5 欢迎文件	17
1.5 JSF 开发环境	17
1.5.1 集成开发环境	17
1.5.2 可视构建器工具	18
1.5.3 使用 Ant 构建自动化	20
1.6 JSF 框架服务	23
1.7 内幕	25
1.7.1 呈现页面	26
1.7.2 解码请求	27
1.7.3 生命周期	28
第 2 章 受管理 Bean	31
2.1 Bean 的定义	31
2.1.1 Bean 属性	33
2.1.2 值表达式	34
2.2 消息包	35
2.2.1 具有可变部分的消息	37
2.2.2 设置应用程序的本地化	37
2.3 示例应用程序	39

2.4 支撑 Bean	45
2.5 Bean 作用域	46
2.5.1 会话作用域	46
2.5.2 应用程序作用域	47
2.5.3 请求作用域	47
2.5.4 生命周期说明	48
2.6 配置 Bean	48
2.6.1 设置属性值	49
2.6.2 初始化列表和映射	49
2.6.3 链接 Bean 定义	51
2.6.4 字符串转换	53
2.7 值表达式的语法	54
2.7.1 使用方括号	55
2.7.2 映射和列表表达式	55
2.7.3 解析初始术语	56
2.7.4 复合表达式	57
2.7.5 方法表达式	58
第 3 章 导航	61
3.1 静态导航	61
3.2 动态导航	63
3.3 高级导航问题	73
3.3.1 重定向	74
3.3.2 通配符	75
3.3.3 使用 from-action	75
3.3.4 导航算法	76
第 4 章 标准 JSP 标签	79
4.1 JSF 核心标签概述	80
4.2 JSF HTML 标签概述	82
4.3 表单	89
4.4 文本字段和文本区域	94
4.4.1 隐藏字段	97

4.4.2	使用文本字段和文本区域	97	6.3.1	验证字符串长度和数值范围	202
4.4.3	显示文本和图片	101	6.3.2	必需值的检查	203
4.5	按钮和链接	103	6.3.3	显示验证错误	204
4.5.1	使用命令按钮	104	6.3.4	跳过验证	205
4.5.2	使用命令链接	108	6.3.5	一个完整的验证例子	205
4.6	选择标签	113	6.4	使用自定义转换器和验证器编程	207
4.6.1	复选框和单选按钮	115	6.4.1	实现自定义转换器类	207
4.6.2	菜单和列表框	118	6.4.2	实现自定义验证器类	220
4.6.3	项目	120	6.4.3	注册自定义验证器	223
4.7	消息	137	6.4.4	使用 Bean 方法验证	225
4.8	面板	142	6.4.5	为转换器提供属性	226
第 5 章	数据表	147	6.4.6	在多个组件间验证关系	226
5.1	数据表标签——h:dataTable	148	第 7 章	事件处理	231
5.2	一个简单的表	148	7.1	生命周期事件	232
5.2.1	h:dataTable 属性	151	7.2	值变化事件	233
5.2.2	h:column 属性	153	7.3	动作事件	238
5.3	表头、表尾和标题	153	7.4	事件监听器标签	247
5.4	JSF 组件	156	7.5	直接组件	249
5.5	编辑表元	160	7.5.1	使用直接输入组件	250
5.6	样式	163	7.5.2	使用直接命令组件	251
5.6.1	列样式	163	7.6	从 UI 到服务器传递数据	252
5.6.2	行样式	164	7.6.1	f:param 标签	253
5.7	数据库表	165	7.6.2	f:attribute 标签	254
5.8	表模型	170	7.6.3	f:setPropertyActionListener 标签	255
5.8.1	编辑表模型	171	7.7	阶段事件	256
5.8.2	排序和过滤	176	7.8	把所有组件放在一起	265
5.9	滚动技术	186	第 8 章	子视图和 Tiles	275
5.9.1	使用滚动条滚动表	186	8.1	常见布局	275
5.9.2	使用分页组件滚动	187	8.2	书籍阅读器和图书馆	276
第 6 章	转换和验证	189	8.3	书籍阅读器	278
6.1	转换和验证过程概述	189	8.3.1	单一 JSF 页面	279
6.2	使用标准转换器	190	8.3.2	公共内容包含	285
6.2.1	数字和日期的转换	190	8.3.3	基于 JSP 程序的包含内容	285
6.2.2	转换错误	194	8.3.4	JSF 特定的考虑	286
6.2.3	一个完整的转换器的例子	199			
6.3	使用标准验证器	202			

- 8.4 书籍阅读器中的内容包含287
 - 8.4.1 了解 Tiles290
 - 8.4.2 安装 Tiles290
 - 8.4.3 为书籍阅读器使用 Tiles291
 - 8.4.4 参数化 Tiles292
 - 8.4.5 扩展 Tiles293
- 8.5 图书馆297
 - 8.5.1 嵌入 Tiles298
 - 8.5.2 Tiles 控制器298
- 第 9 章 自定义组件、转换器
和验证器311**
 - 9.1 用于实现自定义组件的类312
 - 9.1.1 标签和组件314
 - 9.1.2 自定义组件开发者的
工具箱316
 - 9.2 编码：产生标记317
 - 9.3 解码：处理请求值320
 - 9.4 实现自定义组件标签325
 - 9.4.1 TLD 文件326
 - 9.4.2 标签处理程序类329
 - 9.4.3 微调控制器程序333
 - 9.4.4 在 JSF 1.1 中定义标签
处理程序336
 - 9.5 深入理解微调控制器341
 - 9.5.1 使用外部呈现器341
 - 9.5.2 从外部呈现器中调用
转换器346
 - 9.5.3 支持值变化监听器347
 - 9.5.4 支持方法表达式348
 - 9.5.5 示例程序350
 - 9.6 编码 JavaScript 以避免到
服务器往返356
 - 9.7 使用子组件和 facets359
 - 9.7.1 处理 SelectItem 子组件363
 - 9.7.2 处理 facets363
 - 9.7.3 编码 CSS 样式364
 - 9.7.4 使用隐藏字段366
 - 9.7.5 保存和重建状态367
 - 9.7.6 触发动作事件369
 - 9.7.7 使用多页选项卡376
- 9.8 实现自定义转换器和
验证器382
 - 9.8.1 自定义转换器标签382
 - 9.8.2 自定义验证器标签391
- 第 10 章 外部服务401**
 - 10.1 使用 JDBC 访问数据库401
 - 10.1.1 发送 SQL 语句401
 - 10.1.2 连接管理403
 - 10.1.3 填补连接泄漏403
 - 10.1.4 使用预编译语句405
 - 10.2 配置数据源406
 - 10.2.1 在 GlassFish 中配置数据库
资源406
 - 10.2.2 在 Tomcat 中配置数
据库资源408
 - 10.2.3 访问容器管理的资源410
 - 10.2.4 一个完整的数据库
例子412
 - 10.3 LDAP 介绍421
 - 10.3.1 LDAP 目录421
 - 10.3.2 配置 LDAP 服务器423
 - 10.3.3 访问 LDAP 目录信息426
 - 10.4 管理配置信息431
 - 10.4.1 配置 Bean431
 - 10.4.2 配置外部上下文433
 - 10.4.3 配置容器管理资源434
 - 10.4.4 创建 LDAP 应用程序438
 - 10.5 容器管理的验证和授权450
 - 10.6 使用 Web 服务461
- 第 11 章 Ajax471**
 - 11.1 Ajax 基础472
 - 11.2 JavaScript 库474
 - 11.2.1 Prototype 库475
 - 11.2.2 褪色技术库475
 - 11.3 表单自动完成475
 - 11.4 实时验证478

11.5	传送客户端视图状态	483
11.6	直接 Web 远程调用	484
11.7	Ajax 组件	486
11.7.1	混合组件	487
11.7.2	将 JavaScript 放在呈现器外面	490
11.7.3	传输 JSP 标签属性到 JavaScript 代码	492
11.8	Ajax4jsf	493
11.8.1	使用 Ajax4jsf 实现表单自动完成	494
11.8.2	使用 Ajax4jsf 实现实时验证	498
第 12 章	开源	509
12.1	Web 流——Shale	510
12.1.1	对话框配置	513
12.1.2	进入对话框	513
12.1.3	对话框导航	514
12.1.4	对话框作用域	515
12.1.5	对话框上下文敏感	517
12.1.6	子对话框	519
12.2	另一种视图技术——Facelets	521
12.2.1	XHTML 视图	522
12.2.2	标记通过使用 JSF 组件替换: jsfc 属性	523
12.2.3	使用 JSF 标签	526
12.2.4	使用模板的页面合成	528
12.2.5	Facelets 自定义标签	530
12.3	EJB 集成——Seam	531
12.3.1	地址簿	532
12.3.2	配置	535
12.3.3	实体 Bean	536
12.3.4	有状态会话 Bean	538
12.3.5	JSF DataModel 集成	540
12.3.6	对话作用域	542
第 13 章	如何实现	545
13.1	Web 用户界面设计	545
13.1.1	如何找到更多的组件?	545
13.1.2	如何支持文件上传?	548
13.1.3	如何显示图像映射?	556
13.1.4	如何在 MyPage 中包含一个 applet?	557
13.1.5	在 JSF 页面中如何产生二进制数据?	559
13.1.6	如何一次在一个页面中显示大型数据集?	569
13.1.7	如何产生一个弹出窗口?	575
13.1.8	如何有选择地显示和隐藏组件?	583
13.1.9	如何自定义错误页面?	584
13.2	验证	588
13.2.1	如何编写自己的客户端验证标签?	588
13.2.2	如何为客户端验证使用 Shale 验证器?	595
13.2.3	如何验证组件间的关系?	596
13.3	编程	597
13.3.1	在 Eclipse 中如何使用 JSF?	597
13.3.2	如何查找配置文件?	600
13.3.3	JSF 组件如何访问 JAR 文件中的资源?	600
13.3.4	如何将一系列标签打包到 JAR 文件中?	604
13.3.5	在 JavaScript 中, 如何为产生 document.forms[id] 取得表单 ID?	605
13.3.6	如何使 JavaScript 函数每页只出现一次?	605
13.3.7	如何执行初始化或清理工作?	606

- 13.3.8 如何存储一个受管理 Bean, 使它的生命周期比请求作用域长但比会话作用域短?606
- 13.3.9 如何扩展 JSF 表达式语言?607
- 13.4 调试和日志610
 - 13.4.1 如何解释栈跟踪?611
 - 13.4.2 如何避免“复杂的堆栈跟踪”?614
 - 13.4.3 如何“热部署”应用程序?614
 - 13.4.4 如何注释掉 JSF 页面的一部分?615
 - 13.4.5 如何找到日志?616
 - 13.4.6 如何查看页面接收了什么参数?618
 - 13.4.7 如何开启 JSF 容器的日志功能?619
 - 13.4.8 如何调试问题页?622
 - 13.4.9 如何找到库源代码?623

本章主题

- 为什么要选择 JavaServer Faces?
- 软件安装
- 一个简单的例子
- 示例应用程序分析
- JSF 开发环境
- JSF 框架服务
- 内幕

1.1 为什么要选择 JavaServer Faces

从招聘网站上发布的职位信息可以看出，当前开发 Web 应用程序有两种流行的方式：

- “快速开发”方式。在这种方式中，使用可视化开发环境，例如 Microsoft ASP.NET。
- “硬编码”方式。在这种方式中，要编写许多代码，以支持高性能的后端，例如 Java EE(Java 企业版本)。

许多开发团队面临这样一个艰难的选择。Java EE 是一个充满吸引力的平台。它具有高度的可扩展性，易于跨平台使用，并由许多供应商支持。另一方面，ASP.NET 易于创建吸引人的用户界面而无需烦琐的编码。对程序员来说，当然希望兼有高性能的后端和容易的编程用户界面。JSF(JavaServer Faces)的目的就是将快速用户界面开发引入服务器端 Java 中。

读者如果熟悉客户端 Java 开发，可以将 JSF 想像成“适用于服务器端应用程序的 Swing”。如果以前有开发 JSP 的经验，则会发现 JSF 提供了许多 JSP 开发人员必须自己实现的组件，例如网页导航和验证。可以将 servlet 和 JSP(Java 服务器网页)看作是在高层 JSF 框架覆盖下的“汇编语言”。如果熟悉服务器端框架，例如 Struts，则会发现 JSF 使用了类似的架构，但是提供了许多额外的服务。



说明

要学习本书，不必熟知 Swing、JSP 或者 Struts 的内容。读者只需了解 Java 和 HTML 即可。

JSF 包含如下部分：

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

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，数据表和树形列表。

JSF 包含了所有处理事件和构成组件时需要的代码。应用程序编程人员可以忽略这些细节，把精力放在应用程序的逻辑上。

或许最重要的是，JSF 是 Java EE 标准的一部分。JSF 包含在每个 Java EE 应用程序服务器中，并且可以很轻松地将它添加到独立的 Web 服务器中，例如 Tomcat。

更多信息，见第 1.6 节“JSF 框架服务”。许多 IDE(集成开发环境)都支持 JSF，并支持从代码自动生成到可视页面设计的特性。参考第 1.5 节“JSF 开发环境”可以获得更多的信息。在以下的章节中，我们将展示如何手工开发一个 JSF 应用程序，以便让读者理解 IDE 在该框架下是如何工作的，以便更有效地解决问题。

1.2 软件安装

需要如下软件包以便开始学习：

- JDK (Java SE 开发包) 5.0 或者更高版本(<http://java.sun.com/j2se>)
- JSF 1.2
- 本书的示例代码，在 <http://corejsf.com> 下载

我们假定读者已经安装了 JDK，并熟悉 JDK 工具。关于 JDK 的更多信息，请参考 Horstmann, Cay 和 Cornell, Gary 编著的 *Core Java™ 2, vol.2-Advanced Features(7thed.)*。Santa Clara, CA:Sun Microsystems Press/ Prentice Hall。

由于 JSF1.2 是 Java EE 5 规范的一部分，试用 JSF 的最简单方法是使用一个兼容 Java EE 5 的应用服务器。在这一节中，我们将描述 GlassFish 应用服务器 (<http://glassfish.dev.java.net>)。读者可以在相应的网站(<http://corejsf.com>)上找到其他应用服务器的使用说明。



说明

在本书出版的时候,主要有两种版本的 JSF。最新的版本 JSF 1.2,作为 Java EE 5 的一部分在 2006 年发布。原来的版本,JSF 1.0,是在 2004 年发布的。一个漏洞修复版本,称为 JSF 1.1,之后不久也发布了。本书包括 JSF 1.1 版和 1.2 版,但是重点是 JSF 1.2 版。



说明

要学习本书,不必熟知 Swing、JSP 或者 Struts 的内容。读者只需了解 Java 和 HTML 即可。

如果不想安装完整的应用服务器,也可将 Tomcat (<http://tomcat.apache.org>)与 Sun 公司的 JSF 库(<http://javaserverfaces.dev.java.net>)一起使用。参考本书的配套网站(<http://corejsf.com>)获得安装说明。

按照网站上的说明安装 GlassFish,然后启动应用服务器。在 UNIX/Linux 上,使用下面的命令:

```
glassfish/bin/asadmin start-domain
```

(如图 1-1 所示)这里, glassfish 是安装 GlassFish 软件的目录。

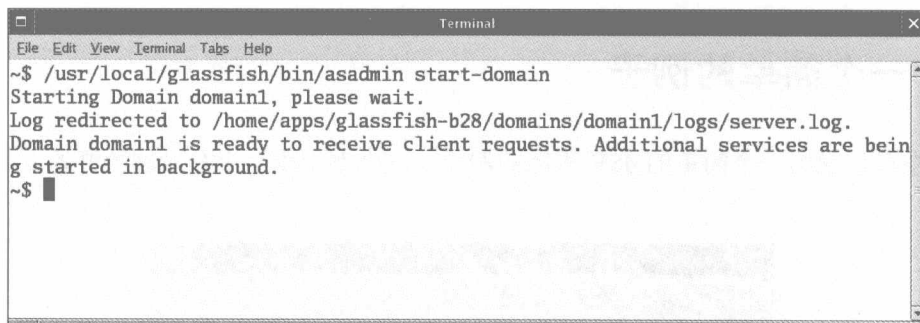


图 1-1 启动 GlassFish

在 Windows 上,使用下面的命令:

```
glassfish\bin\asadmin start-domain
```

为了测试 GlassFish 是否正常运行,在浏览器地址栏中输入 <http://localhost:8080>。应该看到 GlassFish 欢迎页,如图 1-2 所示。

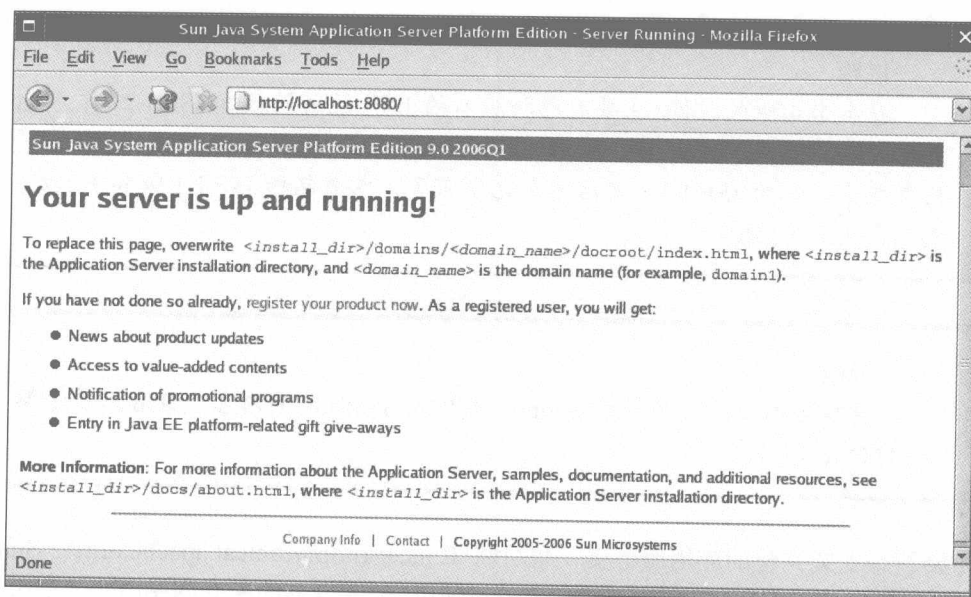


图 1-2 GlassFish 欢迎页

使用下面的命令关闭 GlassFish:

```
glassfish/bin/asadmin stop-domain
```

或者在 Windows 上, 使用:

```
glassfish\bin\asadmin stop-domain
```

1.3 一个简单的例子

现在我们学习一个简单的 JSF 程序的例子。这个例子由一个登录屏幕开始, 如图 1-3 所示。

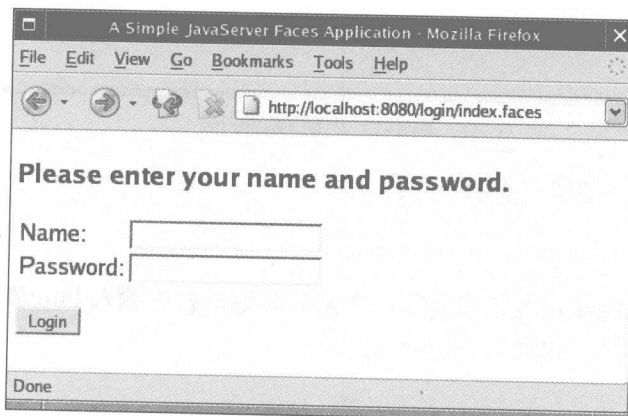


图 1-3 登录屏幕

当然, 在实际的 Web 应用程序中, 这个屏幕会由熟练的美工人员来美化。

描述登录屏幕的文件实际上是一个带有一些额外标签的 HTML 文件(如程序清单 1-1 所示)。其可视外观可以很容易地由无任何编程技能的美工人员来改善。

程序清单 1-1 login/web/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.   <f:view>
5.     <head>
6.       <title>A Simple JavaServer Faces Application</title>
7.     </head>
8.     <body>
9.       <h:form>
10.        <h3>Please enter your name and password.</h3>
11.        <table>
12.          <tr>
13.            <td>Name:</td>
14.            <td>
15.              <h:inputText value="#{user.name}"/>
16.            </td>
17.          </tr>
18.          <tr>
19.            <td>Password:</td>
20.            <td>
21.              <h:inputSecret value="#{user.password}"/>
22.            </td>
23.          </tr>
24.        </table>
25.        <p>
26.          <h:commandButton value="Login" action="login"/>
27.        </p>
28.      </h:form>
29.    </body>
30.  </f:view>
31. </html>
```

在本章 1.4.2 节的“JSF 页面”部分，我们将详细讨论这个文件的内容。现在，注意以下几个要点：

- 大量标签是标准的 HTML 标签，如 body、table 等。
- 有些标签是具有前缀的，例如 f:view 和 h:inputText。这些是 JSF 标签。两个 taglib 声明声明了 JSF 标签库。
- h:inputText、h:inputSecret 和 h:commandButton 标签与图 1-3 中的 Name 字段、Password 字段和提交按钮相对应。

- 输入字段被链接到对象属性。例如，属性 `value="#{user.name}"` 告诉 JSF 实现链接文本字段到用户对象的 `name` 属性。在本章 1.4.1 节的“Beans”部分，我们将详细讨论这个链接。

当用户输入姓名和密码，单击 Login 按钮之后，屏幕将显示欢迎界面(见图 1-4)。第 1.4.2 节的程序清单 1-3 显示了这个屏幕的源代码。第 1.4.3 节的“导航”部分解释程序如何从登录页面导航进入欢迎界面。

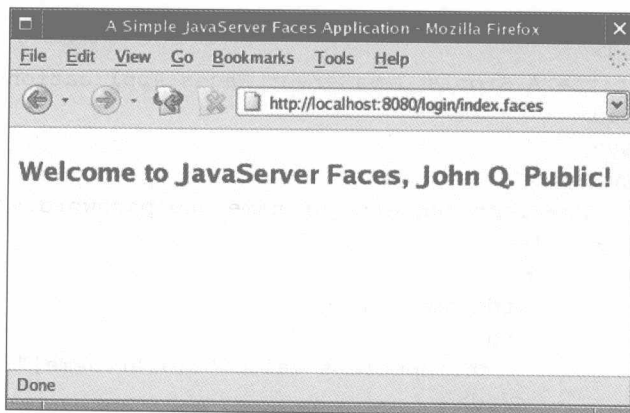


图 1-4 欢迎界面

欢迎消息包含用户名。这里，密码被忽略。

这个程序的目的当然不是为了给任何人留下深刻印象，而是为了演示创建一个 JSF 程序必需的各个部分。

1.3.1 组成部分

我们的示例应用程序包含下列组成部分：

- 定义登录和欢迎屏幕的页面。我们称之为 `index.jsp` 和 `welcome.jsp`。
- 管理用户数据(在我们的例子中，是用户名和密码)的 bean。bean 是一个提供属性的 Java 类，通常属性后面紧跟遵守简单命名约定的 `getter` 和 `setter` 方法。这个类的代码在 `UserBean.java` (程序清单 1-2)文件中。注意该类包含在 `com.corejsf` 包中。
- 程序的配置文件列出了 bean 资源和导航规则。这个文件默认为 `faces-config.xml`。
- 保证 `servlet` 容器正常运行必需的其他文件：`web.xml` 和 `index.html`，后者负责将用户重定向到正确的登录页 URL。

更高级的 JSF 程序也具有相同的结构，但是它们可以包含其他的 Java 类，例如事件处理程序、验证器和自定义组件。

程序清单 1-2 login/src/java/com/corejsf/UserBean.java

```
1. package com.corejsf;
2.
3. public class UserBean {
4.     private String name;
5.     private String password;
```