

<<JavaServer Faces核心编程>>

图书基本信息

书名：<<JavaServer Faces核心编程>>

13位ISBN编号：9787302185499

10位ISBN编号：7302185492

出版时间：2008-10

出版时间：清华大学出版社

作者：（美）吉瑞（Gearym,D），（美）霍斯特曼（Horstmann,G） 著；马振晗 译

页数：623

译者：马振晗

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

<<JavaServer Faces核心编程>>

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。

<<JavaServer Faces核心编程>>

内容概要

本书由著名畅销书作家、JSF 1.0专家组成员David Geary主笔，是JSF编程图书中的绝对“头号指南”，自第一版出版以来，一直是广大JSF学习者的首选教程。

本书全面深入地讲解了JSF 1.2编程的各个方面，包括JSF的各种增强特性、强大的Ajax开发技术，以及使JSF更具价值的开源创新内容，为创建高性能的应用程序提供系统级的最佳实用方案。两位传奇作者利用自身对Java平台的透彻理解，为读者提供了大量解决方案、技巧、提示和编写优秀JSF 1.2产品代码的方法，无论读者是JSF或者servlet方面的新手还是老手，都将大受裨益。

作者简介

David Geary, 从1994年到1997年在Sun Microsystems公司工作, 曾任JSF 1.0专家组成员。目前, 他担任Clarity Training Inc.公司的董事长, 该公司是一家从事服务器端Java技术的培训咨询机构。

David还是8本Java技术书籍的作者, 其中包括最畅销的Graphic, Java 2系列、AdvancedJavaServer Pages和Google Web Toolkit Solutions。

此外, David也是JSTL专家组成员、Apache Struts项目的执行委员, 他还曾为Sun的Web Developer Certification Exam编写试题。

作为一名演讲者, David在全球四大Java研讨会之一的No Fluff Just Stuff tour上经常作例行演讲。2005年, David和CraigMcClanahan合作的Shale Presentation, 获得了JavaOne Rock Star的称号。

<<JavaServer Faces核心编程>>

书籍目录

第1章 入门 1.1 为什么要选择JavaServer Faces 1.2 软件安装 1.3 一个简单的例子 1.3.1 组成部分 1.3.2 目录结构 1.3.3 构建说明 1.4 示例应用程序分析 1.4.1 Beans 1.4.2 JSF页面 1.4.3 导航 1.4.4 Servlet配置 1.4.5 欢迎文件 1.5 JSF开发环境 1.5.1 集成开发环境 1.5.2 可视构建器工具 1.5.3 使用Ant构建自动化 1.6 JSF框架服务 1.7 内幕 1.7.1 呈现页面 1.7.2 解码请求 1.7.3 生命周期第2章 受管理Bean 2.1 Bean的定义 2.1.1 Bean属性 2.1.2 值表达式 2.2 消息包 2.2.1 具有可变部分的消息 2.2.2 设置应用程序的本地化 2.3 示例应用程序 2.4 支撑Bean 2.5 Bean作用域 2.5.1 会话作用域 2.5.2 应用程序作用域 2.5.3 请求作用域 2.5.4 生命周期说明 2.6 配置Bean 2.6.1 设置属性值 2.6.2 初始化列表和映射 2.6.3 链接Bean定义 2.6.4 字符串转换 2.7 值表达式的语法 2.7.1 使用方括号 2.7.2 映射和列表表达式 2.7.3 解析初始术语 2.7.4 复合表达式 2.7.5 方法表达式第3章 导航 3.1 静态导航 3.2 动态导航 3.3 高级导航问题 3.3.1 重定向 3.3.2 通配符 3.3.3 使用from-action 3.3.4 导航算法第4章 标准JSP标签 4.1 JSF核心标签概述 4.2 JSF HTML标签概述 4.3 表单 4.4 文本字段和文本区域 4.4.1 隐藏字段 4.4.2 使用文本字段和文本区域 4.4.3 显示文本和图片 4.5 按钮和链接 4.5.1 使用命令按钮 4.5.2 使用命令链接 4.6 选择标签 4.6.1 复选框和单选按钮 4.6.2 菜单和列表框 4.6.3 项目 4.7 消息 4.8 面板第5章 数据表 5.1 数据表标签——h:dataTable 5.2 一个简单的表 5.2.1 h:dataTable属性 5.2.2 h:column属性 5.3 表头、表尾和标题 5.4 JSF组件 5.5 编辑表元 5.6 样式 5.6.1 列样式 5.6.2 行样式 5.7 数据库表 5.8 表模型 5.8.1 编辑表模型 5.8.2 排序和过滤 5.9 滚动技术 5.9.1 使用滚动条滚动表 5.9.2 使用分页组件滚动第6章 转换和验证 6.1 转换和验证过程概述 6.2 使用标准转换器 6.2.1 数字和日期的转换 6.2.2 转换错误 6.2.3 一个完整的转换器的例子 6.3 使用标准验证器 6.3.1 验证字符串长度和数值范围 6.3.2 必需值的检查 6.3.3 显示验证错误 6.3.4 跳过验证 6.3.5 一个完整的验证例子 6.4 使用自定义转换器和验证器编程 6.4.1 实现自定义转换器类 6.4.2 实现自定义验证器类 6.4.3 注册自定义验证器 6.4.4 使用Bean方法验证 6.4.5 为转换器提供属性 6.4.6 在多个组件间验证关系第7章 事件处理 7.1 生命周期事件 7.2 值变化事件 7.3 动作事件 7.4 事件监听器标签 7.5 直接组件 7.5.1 使用直接输入组件 7.5.2 使用直接命令组件 7.6 从UI到服务器传递数据 7.6.1 f:param标签 7.6.2 f:attribute标签 7.6.3 f:setPropertyActionListener标签 7.7 阶段事件 7.8 把所有组件放在一起第8章 子视图和Tiles 8.1 常见布局 8.2 书籍阅读器和图书馆 8.3 书籍阅读器 8.3.1 单一JSF页面 8.3.2 公共内容包含 8.3.3 基于JSP程序的包含内容 8.3.4 JSF特定的考虑 8.4 书籍阅读器中的内容包含 8.4.1 了解Tiles 8.4.2 安装Tiles 8.4.3 为书籍阅读器使用Tiles 8.4.4 参数化Tiles 8.4.5 扩展Tiles 8.5 图书馆 8.5.1 嵌入Tiles 8.5.2 Tiles控制器 第9章 自定义组件、转换器和验证器 用于实现自定义组件的类 9.1.1 标签和组件 9.1.2 自定义组件开发者的工具箱 9.2 编码：产生标记 9.3 解码：处理请求值 9.4 实现自定义组件标签 9.4.1 TLD文件 9.4.2 标签处理程序类 9.4.3 微调控制器程序 9.4.4 在JSF 1.1中定义标签处理程序 9.5 深入理解微调控制器 9.5.1 使用外部呈现器 9.5.2 从外部呈现器中调用转换器 9.5.3 支持值变化监听器 9.5.4 支持方法表达式 9.5.5 示例程序 9.6 编码JavaScript以避免到服务器往返 9.7 使用子组件和facets 9.7.1 处理SelectItem子组件 9.7.2 处理facets 9.7.3 编码CSS样式 9.7.4 使用隐藏字段 9.7.5 保护和重建状态 9.7.6 触发动作事件 9.7.7 使用多页选项卡 9.8 实现自定义转换器和验证器 9.8.1 自定义转换器标签 9.8.2 自定义验证器标签 第10章 外部服务 10.1 使用JDBC访问数据库 10.1.1 发送SQL语句 10.1.2 连接管理 10.1.3 填补连接泄漏 10.1.4 使用预编译语句 10.2 配置数据源 10.2.1 在GlassFish中配置数据库资源 10.2.2 在Tomcat中配置数据库资源 10.2.3 访问容器管理的资源 10.2.4 一个完整的数据库例子 10.3 LDAP介绍 10.3.1 LDAP目录 10.3.2 配置LDAP服务器 10.3.3 访问LDAP目录信息 10.4 管理配置信息 10.4.1 配置Bean 10.4.2 配置外部上下文 10.4.3 配置容器管理资源 10.4.4 创建LDAP应用程序 10.5 容器管理的验证和授权 10.6 使用Web服务 第11章 Ajax 11.1 Ajax基础 11.2 JavaScript库 11.2.1 Prototype 11.2.2 褪色技术库 11.3 表单自动完成 11.4 实时验证 11.5 传送客户端视图状态 11.6 连接Web远程调用 11.7 Ajax组件 11.7.1 混合组件 11.7.2 将JavaScript放在呈现器外面 11.7.3 传输JSP标签属性到JavaScript代码 11.8 Ajax4jfs 11.8.1 使用Ajax4jfs实现表单自动完成 11.8.2 使

<<JavaServer Faces核心编程>>

- 用Ajax4jsf实现实时验证 第12章 开源 12.1 Web流——Shale 12.1.1 对话框配置 12.1.2 进入对话框
- 12.1.3 对话框导航 12.1.4 对话框作用域 12.1.5 对话框上下文敏感 12.1.6 子对话框
- 12.2 另一种视图技术——Facelets 12.2.1 XHTML视图 12.2.2 标记通过使用JSF组件替换：jsfc
- 属性 12.2.3 使用JSF标签 12.2.4 使用模板的页面合成 12.2.5 Facelets自定义标签 12.3 EJB集成——Seam
- 12.3.1 地址簿 12.3.2 配置 12.3.3 实体Bean 12.3.4 有状态会话Bean
- JSF DataModel集成 12.3.6 对话作用域 第13章 如何实现…… 13.1 Web用户界面设计 13.1.1 如何找到更多的组件？
- 13.1.2 如何支持文件上传？
- 13.1.3 如何显示图像映射？
- 13.1.4 如何在MyPage中包含一个applet？
- 13.1.5 在JSF页面中如何产生二进制数据？
- 13.1.6 如何一次在一个页面中显示大型数据集？
- 13.1.7 如何产生一个弹出窗口？
- 13.1.8 如何有选择地显示和隐藏组件？
- 13.1.9 如何自定义错误页面？
- 13.2 验证 13.2.1 如何编写自己的客户端验证标签？
- 13.2.2 如何为客户端验证使用Shale验证器？
- 13.2.3 如何验证组件间的关系？
- 13.3 编程 13.3.1 在Eclipse中如何使用JSF？
- 13.3.2 如何查找配置文件？
- 13.3.3 JSF组件如何访问JAR 文件中的资源？
- 13.3.4 如何将一系列标签打包到JAR文件中？
- 13.3.5 在JavaScript中，如何为产生document.forms[id]取得表单ID？
- 13.3.6 如何使JavaScript函数每页只出现一次？
- 13.3.7 如何执行初始化或清理工作？
- 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？
- 13.3.9 如何扩展JSF表达式语言？
- 13.4 调试和日志 13.4.1 如何解释栈跟踪？
- 13.4.2 如何避免“复杂的堆栈跟踪”？
- 13.4.3 如何“热部署”应用程序？
- 13.4.4 如何注释掉JSF页面的一部分？
- 13.4.5 如何找到日志？
- 13.4.6 如何查看页面接收了什么参数？
- 13.4.7 如何开启JSF容器的日志功能？
- 13.4.8 如何调试问题页？
- 13.4.9 如何找到库源代码？

<<JavaServer Faces核心编程>>

章节摘录

第1章 入门 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，ASENET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。

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JEF组件很简单，例如，输入字段和按钮。

编辑推荐

本书由著名畅销书作者、JSF 1.0专家组成员David Geary主笔，是JSF编程图书中的“头号指南”。作为升级版，本书内容全面覆盖JSF 1.2，囊括了各种增强特性、强大的Ajax开发技术，以及使JSF更有价值的开源创新内容。

作者David Geary和Cay Horstmann在本书中深入讲解了JSF 1.2开发的各个方面，为创建高性能应用程序提供了系统级的最佳实用方案。

他们利用自身对Java平台内部知识的透彻理解，为读者提供了大量解决方案、技巧、提示和编写优秀JSF 1.2产品代码的方法。

即使读者是JSF或servlet新手，本书也能提供很大的帮助。

本版新增的内容包括：JSF 1.2与Java EE 5平台的更好结合；增强的JSF API；使用Shale控制Web流；使用Facelets将JSP替换为XHTML标签。

此外，本版还涵盖了通过JSF进行Ajax开发的主题——从实时验证、DWR，到在JSF组件中包装Ajax和使用流行的Ajax4jsf框架。

自动化低层细节，消除服务器端开发中不必要的复杂性从有效的UI设计和样式表到国际化，发现JSF最佳实践使用JSF和Tiles构建一致、可重用的用户界面利用外部服务，如数据库、LDAP目录、认证/授权，以及Web服务使用JBoss Seam大大简化数据库后端应用程序的开发实现自定义组件、转换器和验证器掌握JSF 1.2标签库和使用额外的标签库扩展JSF。

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>