

《JSF第一步》

图书基本信息

书名：《JSF第一步》

13位ISBN编号：9787302159735

10位ISBN编号：7302159734

出版时间：2007-10

出版社：清华大学出版社

作者：罗会波

页数：551

版权说明：本站所提供下载的PDF图书仅提供预览和简介以及在线试读，请支持正版图书。

更多资源请访问：www.tushu000.com

《JSF第一步》

内容概要

《JSF第一步:JSF+Spring+Hibernate+AJAX编程》讲述JSF是表示层框架的标准，Hibernate是一个比较完善的对象关系映射工具，Spring则提供了一个Web应用的轻量级的解决方案。在开发一个多层的Java EE应用程序时，这些框架可谓是相辅相成、相得益彰，可以称得上是开发轻量级Java EE应用的三剑客。另外，AJAX是一种非常流行的改善用户体验的技术，但目前国内外还没有一本完整地介绍组合这三种框架及AJAX来开发应用程序的书籍，《JSF第一步:JSF+Spring+Hibernate+AJAX编程》是在这方面的一个尝试。《JSF第一步:JSF+Spring+Hibernate+AJAX编程》详尽地介绍了JSF的基本结构和工作原理、标签、验证与转换、事件处理、国际化编程及IoC；对象关系映射的概念、Hibernate的结构及与JSF的集成；Spring的结构、AOP、Spring与JSF和Hibernate的集成；常用IDE（Eclipse和MyEclipse）的使用方法等内容。

作者简介

罗会波 高级工程师，IBM认证DB2数据库系统管理员。早年从事中学物理教学工作，后在亿利达（Elite）集团明泰R/D部（深圳）从事通信产品研发工作。1996年底进入税务系统从事信息化工作至今。在亿利达工作期间开始接触汇编语言编程；在税务系统主持或参与了“湖北省国税局纳税大厅Web查询系统”、“全国税务征管软件Ctais纳税人查询系统”等数个省局及市局税务信息化项目的开发工作；参与了湖北省国税局内部教材《国税系统办公自动化实用指南》一书的编撰工作；曾在《计算机世界》、《电脑爱好者》及《电脑编辑技巧与维护》等报刊杂志及赛迪网上发表过包括《Struts原理与实践》系列讲座（该系列曾长期占据赛迪网技术文章排行榜TOP10）在内的多篇原创及翻译的IT技术文章，文风严谨，又不失幽默，受到广大读者的一致好评。

书籍目录

第1章 为什么要学习框架及学习什么样的框架

11.1 Java的发展与变化

11.1.1 版本与称谓的变化

11.1.2 应用程序开发方式的变迁

11.2 两种形式的多层结构的J2EE应用程序

21.2.1 胖客户端J2EE应用程序

21.2.2 Web应用程序

31.3 Web应用的两种实现方式

41.3.1 传统的重量级实现方式

51.3.1.1 运行环境

51.3.1.2 EJB

51.3.1.3 其他特点

61.3.1.4 适应的场合

61.3.2 方兴未艾的轻量级实现方式

71.3.2.1 运行环境

71.3.2.2 POJO

71.3.2.3 其他特点

71.3.2.4 适应场合

71.4 风云际会的框架世界

71.4.1 框架的出现与流行

81.4.2 大器晚成展新颜——JSF

81.4.3 层林深处好冬眠——Hibernate

91.4.4 七音合奏春光曲——Spring

91.5 为什么是JSF而不是Struts

91.5.1 比Struts更高的抽象

91.5.2 早先暴露的缺陷不断被弥补

111.5.2.1 解决了与JSP集成中的一些突出问题

111.5.2.2 统一表达式语言与JSTL迭代标签的支持

111.5.2.3 其他新的特征

121.5.3 与客户端技术的完美结合

121.5.3.1 可以借助JavaScript改善用户体验

121.5.3.2 可与AJAX无缝集成

12第2章 JSF与Web应用程序

132.1 JSF的设计目标及与Web应用的关系

132.2 Web应用程序概览

132.2.1 基本的Web组件

132.2.2 Web部署描述器文件

132.2.3 Web应用程序的物理结构

142.3 Web应用程序的开发环境配置

142.3.1 Java 2 SDK的下载与安装

142.3.2 Web容器的下载与安装

152.3.3 环境变量的设置

162.3.4 其他准备工作

162.4 Servlet

182.4.1 Servlet的基本概念

182.4.2 Servlet的体系结构

182.4.3 你的第一个Servlet应用程序——Hello World

- 192.4.3.1 建立示例应用目录结构
- 192.4.3.2 编写Servlet类文件
- 192.4.3.3 编译Servlet类文件
- 202.4.3.4 编写部署描述器文件web.xml
- 202.4.3.5 部署与运行示例应用
- 212.4.3.6 原理分析
- 222.4.4 Servlet的生命周期
 - 222.4.4.1 初始化
- 232.4.4.2 处理请求
- 232.4.4.3 销毁
- 242.4.5 Servlet的作用域对象
- 242.4.6 过滤请求与响应
- 242.5 JSP
 - 242.5.1 JSP如何表示静态内容
 - 252.5.2 JSP如何表示动态内容
 - 252.5.3 JSP的注释
 - 252.5.3.1 HTML注释
 - 262.5.3.2 JSP注释
 - 262.5.4 JSP示例
 - 262.5.4.1 建立应用程序目录
 - 262.5.4.2 编写JSP示例代码
 - 262.5.4.3 部署和运行JSP
 - 262.5.5 JSP与Servlet的关系
 - 272.5.5.1 JSP转换为Servlet后执行
 - 272.5.5.2 JSP转化后的Servlet的简要分析
 - 272.5.5.3 JSP为Servlet提供了模板机制
 - 292.5.6 JSP的元素
 - 292.5.6.1 指令
 - 292.5.6.2 动作
 - 302.5.6.3 脚本元素
 - 302.5.7 JSP的标签库
 - 322.5.7.1 标签库的组成
 - 322.5.7.2 自定义标签库
 - 332.5.7.3 JSP标准标签库
 - 352.5.8 JSP的内建对象
 - 352.5.8.1 out
 - 362.5.8.2 request
 - 372.5.8.3 session
 - 392.5.8.4 application
 - 412.5.8.5 config
 - 412.5.8.6 pageContext
 - 412.5.8.7 page
 - 412.5.8.8 response
 - 412.5.8.9 exception
 - 412.5.9 JSP中的异常处理
 - 422.5.10 Web组件间的关联关系
 - 422.5.10.1 包含
 - 422.5.10.2 请求转发
 - 432.5.10.3 重定向

- 452.5.11 JavaBean
 - 452.5.11.1 JavaBean的概念
 - 452.5.11.2 为什么要引入JavaBean
 - 462.5.11.3 JavaBean的编码约定
 - 462.5.11.4 一个简单的JavaBean
 - 482.5.11.5 在JSP中使用JavaBean
- 502.5.12 JavaBean在JSP中的有效范围
 - 502.5.12.1 演示Page范围的Bean
 - 502.5.12.2 演示Request范围的Bean
 - 522.5.12.3 演示Session范围的Bean
 - 532.5.12.4 演示Application范围的Bean
- 542.6 早期框架Struts
 - 552.6.1 应用程序的分层
 - 552.6.2 表单的处理与验证
 - 552.6.3 类型转换
 - 562.6.4 导航
 - 562.6.5 资源管理
 - 562.6.6 国际化编程
 - 562.6.7 异常处理的加强
 - 562.6.8 模板的重用管理与布局
- 572.7 JSF
 - 572.7.1 具有早期框架的一些特性
 - 572.7.2 有状态的用户界面组件模型
 - 572.7.3 用户界面事件的服务器端的处理
 - 582.7.4 自动生成各种标志语言
 - 582.7.5 与业务对象的声明式集成
- 58第3章 JSF应用程序开发入门
 - 593.1 JSF开发环境的配置
 - 593.1.1 JSF参考实现的下载与安装
 - 593.1.2 ANT的下载与安装
 - 603.1.3 环境变量的设置
 - 603.2 开发您的第一个JSF应用程序
 - 603.2.1 Hello World应用程序的需求分析
 - 613.2.2 应用程序的目录结构
 - 613.2.3 应用程序的页面文件
 - 613.2.3.1 index.jsp
 - 613.2.3.2 hello.jsp
 - 623.2.3.3 howDoYouDo.jsp
 - 653.2.3.4 howAreYou.jsp
 - 663.2.4 应用程序的Backing Bean
 - 663.2.4.1 Backing Bean及其作用
 - 663.2.4.2 示例中的Backing Bean类文件
 - 673.2.4.3 Backing Bean的特点
 - 693.2.5 JSF的配置文件faces-config.xml
 - 693.2.5.1 示例应用的导航规则
 - 703.5.5.2 示例应用的托管JavaBean配置
 - 713.2.6 应用程序的配置文件web.xml
 - 713.2.7 手工编译应用程序
 - 723.2.8 应用程序上下文

- 733.2.9 建立应用程序上下文及部署应用程序
 - 733.2.9.1 直接修改Tomcat的配置文件server.xml
 - 733.2.9.2 使用Tomcat页面形式的管理工具
- 733.2.10 运行应用程序
- 743.3 用ANT工具构建应用程序
 - 753.3.1 ANT工具简介
 - 753.3.2 创建build.xml文件
 - 763.3.2.1 设置属性
 - 763.3.2.2 设置classpath
 - 773.3.2.3 定义clean目标
 - 773.3.2.4 定义prepare目标
 - 773.3.2.5 定义deploy.copyJars目标
 - 773.2.2.6 定义build目标
 - 783.3.2.7 定义build.war目标
 - 793.3.2.8 定义deploy目标
 - 793.3.3 用ANT工具自动构建和部署应用程序
- 79第4章 JSF中的请求处理
 - 804.1 JSF处理请求的基本过程
 - 804.1.1 将请求映射到FacesServlet
 - 804.1.2 FacesServlet类
 - 804.1.3 获得配置Servlet的配置信息
 - 814.1.4 初始化时获得Faces上下文工厂和生命周期实例
 - 814.1.5 FacesServlet对请求的处理过程
 - 814.1.5.1 获得Faces上下文并用它处理当前请求
 - 824.1.5.2 调用请求处理生命周期的execute()方法
 - 824.1.5.3 调用请求处理生命周期的render()方法
 - 824.1.5.4 释放Faces上下文对象
 - 834.1.5.5 与普通Servlet对请求处理的比较
 - 834.1.6 处理完成后的资源释放
 - 834.2 Faces上下文
 - 834.2.1 Faces上下文简介
 - 834.2.2 Faces上下文的主要方法
 - 844.2.2.1 获取应用程序实例
 - 844.2.2.2 获取外部上下文
 - 844.2.2.3 获取与设置视图根
 - 854.2.2.4 处理消息队列
 - 854.2.2.5 获取呈现器工具包
 - 864.2.2.6 流程控制方法
 - 864.2.2.7 获取表达式语言上下文
 - 874.2.2.8 响应
 - 884.2.2.9 获取Faces上下文的当前实例
 - 884.3 请求处理的生命周期
 - 884.3.1 获得生命周期对象
 - 884.3.2 生命周期的场景
 - 894.3.2.1 非Faces请求生成Faces响应
 - 894.3.2.2 Faces请求生成非Faces响应
 - 904.3.2.3 Faces请求生成Faces响应
 - 904.3.3 标准的JSF请求处理生命周期
 - 904.3.3.1 重建视图

- 924.3.3.2 应用请求值
- 934.3.3.3 处理验证
- 944.3.3.4 更新模型值
- 954.3.3.5 调用应用程序
- 954.3.3.6 呈现响应
- 954.4 Application及相关的类
 - 964.4.1 Application简介
 - 964.4.2 Application的主要属性
 - 974.4.2.1 动作监听器
 - 974.4.2.2 导航处理器
 - 984.4.2.3 视图处理器
 - 984.4.2.4 缺省呈现包标识
 - 984.4.2.5 状态管理器
 - 984.4.2.6 表达式语言解析器
 - 984.4.3 Application动作
 - 984.4.4 Application作为组件工厂
 - 994.4.5 Application支持国际化编程的方法
- 99第5章 JSF的体系结构
 - 1015.1 MVC架构模式
 - 1015.1.1 耦合与解耦
 - 1015.1.2 MVC架构模式
 - 1025.1.3 Web应用中的MVC——Model2
 - 1045.2 MVC架构模式的实现策略
 - 1055.2.1 Struts的实现策略
 - 1055.2.2 JSF的实现策略
 - 1055.2.2.1 JSF的控制器部分
 - 1065.2.2.2 JSF的视图部分
 - 1075.2.2.3 JSF的模型部分
 - 1075.3 JSF的主要组成部分
 - 1085.3.1 JSF主要组成部分及其关系
 - 1085.3.1.1 组成部分概览
 - 1085.3.1.2 各部分之间的关系
 - 1095.3.2 用户界面组件
 - 1095.3.2.1 对组件的直观认识
 - 1095.3.2.2 JSF的组件模型
 - 1095.3.2.3 组件模型中的接口
 - 1105.3.2.4 组件模型中的类
 - 1115.3.2.5 组件的作用
 - 1125.3.2.6 组件的属性
 - 1125.3.2.7 组件在服务器和客户端的表现形式
 - 1135.3.2.8 组件的标识
 - 1145.3.2.9 通过标识引用组件
 - 1155.3.2.10 组件是有状态的
 - 1165.3.2.11 标准组件与自定义组件
 - 1165.3.3 呈现器
 - 1165.3.4 Backing Bean
 - 1175.3.5 事件与事件监听器
 - 1185.3.6 导航系统
 - 1195.3.7 类型转换器

- 1205.3.8 验证器
- 1205.3.9 消息
- 120第6章 标准用户界面组件
- 1226.1 标准组件概述
- 1226.1.1 组件的类型
- 1226.1.2 组件的家族
- 1226.1.3 组件的呈现器类型
- 1236.2 JSF的呈现模型
- 1236.2.1 直接实现方式
- 1236.2.2 委托实现方式
- 1236.2.3 呈现包工厂
- 1246.2.4 呈现包
- 1246.2.5 呈现器
- 1256.3 JSF实现的缺省呈现器类型与标准组件HTML具体类的关系
- 1266.4 组件在JSP页面上的表示
- 1276.4.1 在JSP页面中使用用户界面组件的定制动作
- 1286.4.2 核心标签库
- 1286.4.3 HTML标签库
- 1286.4.4 组件的HTML属性
- 1306.5 标准组件数据模型
- 1306.5.1 标准组件与呈现无关的属性
- 1306.5.2 标准组件模型Bean
- 1316.5.2.1 SelectItem
- 1316.5.2.2 SelectItemGroup
- 1316.5.2.3 DataModel
- 1316.6 标准组件及使用方法
- 1326.6.1 用UIViewRoot控制页面
- 1326.6.1.1 UIViewRoot简介
- 1326.6.1.2 在JSP页面上的应用
- 1336.6.1.3 的使用
- 1346.6.2 用UIForm处理表单
- 1366.6.3 用UIParameter传递参数
- 1376.6.4 用Facet制作表头、表脚及列头
- 1376.6.5 用UIInput处理输入
- 1386.6.5.1 用HtmlInputText输入单行文本
- 1396.6.5.2 用HtmlInputSecret输入密码
- 1396.6.5.3 用HtmlInputTextarea输入多行文本
- 1396.6.5.4 用HtmlInputHidden作为隐含字段
- 1406.6.6 用UIOutput处理输出
- 1406.6.6.1 用HtmlOutputText输出简单文本
- 1406.6.6.2 用HtmlOutputFormat输出格式化文本
- 1416.6.6.3 用HtmlOutputLabel作为组件的标注
- 1426.6.6.4 用HtmlOutputLink作为超链接
- 1426.6.7 用UISelectItem提供单个选项
- 1426.6.8 用UISelectItems提供多个选项
- 1436.6.9 用UISelectBoolean表示逻辑选项
- 1436.6.10 用UISelectOne作为单选组件
- 1446.6.10.1 单选列表框HtmlSelectOneListbox
- 1446.6.10.2 单选下拉菜单HtmlSelectOneMenu

- 1446.6.10.3 单选按钮HtmlSelect-OneRadio
- 1446.6.11 用UISelectMany作为复选组件
- 1456.6.11.1 复选列表框HtmlSelectManyListbox
- 1456.6.11.2 复选下拉菜单HtmlSelectManyMenu
- 1456.6.11.3 复选按钮HtmlSelectManyCheckbox
- 1456.6.12 用UICommand调用应用程序
- 1466.6.12.1 命令按钮HtmlCommandButton
- 1466.6.12.2 命令超链接HtmlCommandLink
- 1466.6.13 用UIMessage显示特定组件的消息
- 1476.6.14 用UIMessages显示多条消息
- 1476.6.15 用UIGraphic显示图片
- 1486.6.16 用UIPanel分组与布局组件
- 1486.6.16.1 用HtmlPanelGroup分组组件
- 1486.6.16.2 用HtmlPanelGrid布局组件
- 1486.6.17 用UIColumn表示数据列
- 1496.6.18 用UIData显示数据集中的数据
- 1496.6.19 到哪里获得组件更详细的信息
- 150第7章 表达式语言与托管Bean
- 1517.1 表达式语言的出现
- 1517.1.1 统一表达式语言出现的背景
- 1517.1.1.1 JSP的表达式语言
- 1517.1.1.2 JSF自己的表达式语言
- 1527.1.1.3 JSP与JSF表达式语言的冲突
- 1537.1.2 统一表达式语言
- 1537.1.2.1 值表达式
- 1537.1.2.2 方法表达式
- 1547.1.2.3 解决原来的JSP与JSF表达式的冲突问题
- 1547.2 托管Bean
- 1557.2.1 托管Bean创建工具与托管Bean
- 1557.2.2 托管Bean的配置
- 1557.2.2.1 基本的配置元素
- 1567.2.2.2 配置托管Bean的属性
- 1567.2.2.3 托管Bean属性的初始化与JSF的IoC
- 1577.2.3 List配置为托管Bean
- 1587.2.4 Map配置为托管Bean
- 1587.3 托管Bean的有效范围
- 1597.3.1 请求范围
- 1597.3.2 会话范围
- 1607.3.3 应用程序范围
- 1607.3.4 NONE范围
- 1607.3.5 不同范围间的托管Bean的引用
- 161第8章 JSF中的导航
- 1638.1 JSF导航概述
- 1638.1.1 应用程序动作与导航
- 1638.1.2 导航处理器的行为是可以配置的
- 1638.2 JSF支持的导航形式
- 1648.2.1 动态导航
- 1648.2.1.1 何为动态导航
- 1648.2.1.2 简单动态导航示例回顾

- 1648.2.2 静态导航
- 1658.2.2.1 何为静态导航
- 1658.2.2.2 静态导航示例
- 1658.3 导航规则
- 1678.3.1 导航规则需要配置的元素
- 1678.3.2 导航规则的匹配算法
- 1678.3.3 复杂导航规则的配置示例
- 1688.4 动作事件监听器方法对导航的影响
- 1698.4.1 动作事件监听器方法间接影响导航
- 1698.4.2 动作事件监听器方法对导航影响的示例
- 1698.4.2.1 导航出发页面
- 1698.4.2.2 包含动作事件监听器方法和动作方法的Backing Bean
- 1708.4.2.3 导航目的地页面
- 1718.4.2.4 配置导航规则
- 1728.4.2.5 部署和运行示例应用程序
- 172第9章 JSF与应用程序模型层的集成
- 1749.1 JSF作为表现层框架
- 1749.1.1 JSF结构回顾
- 1749.1.2 Backing Bean扮演的角色
- 1749.1.3 编写Backing Bean代码时的不良实践
- 1749.2 登录示例应用程序
- 1759.2.1 示例登录应用的需求分析
- 1759.2.2 登录示例应用的界面
- 1759.2.3 目录结构
- 1769.2.4 业务服务接口
- 1769.2.5 业务服务实现
- 1769.2.6 示例应用的Backing Bean
- 1789.2.7 示例中的页面文件
- 1799.2.8 在Faces配置文件中配置导航规则
- 1809.2.9 在Faces配置文件中配置托管Bean
- 1809.2.10 部署和运行示例应用
- 1819.3 JSF中的控制反转 (IoC)
- 1829.3.1 登录示例应用的不足
- 1829.3.2 控制反转 (IoC) 及依赖注入 (DI)
- 1829.3.2.1 控制反转及依赖注入简介
- 1829.3.2.2 依赖注入的形式
- 1839.4 在JSF应用程序中使用依赖注入
- 1859.4.1 在Backing Bean使用type2型依赖注入
- 1859.4.2 在Faces配置文件中配置依赖关系
- 1869.4.3 部署和运行示例应用
- 187第10章 JSF的事件处理
- 18810.1 JSF事件及事件监听器模型
- 18810.1.1 事件类
- 18810.1.2 监听器类
- 18910.1.3 事件广播及处理
- 18910.2 JSF中的事件分类
- 19010.2.1 值改变事件
- 19010.2.2 动作事件
- 19010.2.3 阶段事件

- 19010.3 事件处理示例
 - 19010.3.1 值改变事件的处理示例
 - 19010.3.1.1 示例应用的页面文件
 - 19110.3.1.2 在Backing Bean中编写处理值改变事件方法
 - 19210.3.1.3 在Faces配置文件中配置Backing Bean
 - 19210.3.1.4 部署和运行
 - 19210.3.1.5 用专门的监听器来处理值改变事件
 - 19310.3.2 动作事件的处理示例
 - 19410.3.2.1 在页面中使用命令超链接
 - 19410.3.2.2 在Backing Bean中编写事件处理方法
 - 19510.3.2.3 部署和运行示例应用
 - 19610.3.3 Immediate组件使用示例
 - 19710.3.3.1 在页面上使用Immediate组件
 - 19710.3.3.2 在Backing Bean中编写事件监听器方法
 - 19910.3.3.3 观察示例应用的运行效果
- 199第11章 JSF中的转换
 - 20111.1 JSF的转换模型
 - 20111.1.1 什么是JSF中的转换
 - 20111.1.2 JSF中为什么需要转换
 - 20111.1.3 JSF中何时发生转换
 - 20211.1.4 JSF的转换器接口
 - 20211.2 使用JSF的标准转换器
 - 20311.2.1 JSF的标准转换器
 - 20311.2.2 使用标准转换器的一般方式
 - 20311.3 自动转换方式示例
 - 20411.3.1 在注册页面增加一个表示年龄的文本输入框
 - 20411.3.2 在注册成功的页面上显示年龄
 - 20411.3.3 Backing Bean年龄属性的数据类型为int
 - 20411.3.4 示例应用的业务数据对象
 - 20511.3.5 示例应用中的自定义异常
 - 20611.3.6 示例应用的业务服务
 - 20611.3.7 示例应用的Faces配置
 - 20611.3.8 自动转换的效果
 - 20711.3.8.1 自动转换失败的效果
 - 20711.3.8.2 自动转换成功的效果
 - 20711.4 使用组件标签的属性
 - 20811.5 使用转换器标签
 - 20811.6 Java中的日期与时间
 - 20911.6.1 Java中的Date类
 - 20911.6.2 Calendar类
 - 21011.6.3 GregorianCalendar类
 - 21011.6.4 日期、时间的格式化
 - 21111.6.5 访问数据库时使用的日期类
 - 21111.7 日期时间转换器示例
 - 21211.7.1 建立示例应用的目录结构
 - 21211.7.2 新建一个可以输入日期的页面
 - 21311.7.3 在页面上使用显示日期的组件
 - 21311.7.4 Backing Bean中的日期型属性
 - 21311.7.5 业务对象中的日期属性

- 21411.7.6 业务服务接口中的日期型参数
- 21511.7.7 在业务服务实现中处理日期型数据
- 21511.7.8 更改示例应用的Faces配置
- 21611.7.9 日期转换的效果
- 21711.8 数字转换器
- 21811.9 使用自定义转换器
 - 21811.9.1 创建自定义转换器
 - 21811.9.2 注册自定义转换器
- 22011.10 自定义的转换器示例
 - 22011.10.1 建立示例应用的目录结构
 - 22011.10.2 示例应用的Backing Bean
 - 22011.10.3 新建自定义格式的业务对象
 - 22211.10.4 修改原业务对象
 - 22311.10.5 在业务服务接口的方法中加入新参数
 - 22311.10.6 更改业务服务实现类
 - 22411.10.7 在Faces中配置自定义转换器
 - 22511.10.8 在用户注册页面上使用自定义转换器
 - 22511.10.9 在注册成功页面上使用自定义转换器
 - 22611.10.10 检验自定义转换器的使用效果
- 226第12章 JSF中的验证
 - 22712.1 JSF的验证模型
 - 22712.1.1 验证器类
 - 22712.1.2 验证器的注册
 - 22712.1.3 验证处理过程
 - 22812.1.4 标准验证器实现
 - 22812.2 使用标准验证器示例
 - 22912.2.1 建立示例应用目录结构
 - 22912.2.2 在页面中使用标准验证器
 - 22912.2.3 部署和运行应用程序
 - 23012.3 正则表达式简介
 - 23112.4 使用自定义验证器示例
 - 23312.4.1 建立示例应用目录
 - 23312.4.2 创建自定义验证器类
 - 23412.4.3 在Faces配置文件中注册验证器
 - 23512.4.4 在页面中使用自定义验证器
 - 23512.4.5 自定义验证器的使用效果
 - 23512.5 使用Backing Bean的验证方法
 - 23712.5.1 建立示例应用目录结构
 - 23712.5.2 在Backing Bean中编写验证方法
 - 23712.5.3 在Faces配置中删除自定义验证器
 - 23812.5.4 在页面中引用验证方法
 - 23812.5.5 运行应用程序查看验证效果
- 239第13章 JSF的国际化编程
 - 24013.1 问题的由来
 - 24013.2 字符集的基础知识
 - 24013.2.1 计算机表示字符的方式与字符集
 - 24013.2.2 字符集的发展
 - 24113.2.3 UTF-8字符集
 - 24213.3 Java及Web中的编码

- 24313.3.1 Java中的编码
- 24313.3.2 Web中的编码
- 24313.4 中文乱码的几种表现形式及成因浅析
 - 24413.4.1 字符变问号
 - 24413.4.2 部分汉字是乱码
 - 24713.4.3 显示乱码（有些是汉字但并不是你预期的）
 - 24813.4.4 读写数据库时出现乱码
- 25013.5 JSF国际化编程的思路
- 25013.6 JSF国际化编程的实现
 - 25113.6.1 支持国际化编程的应用的基本特征
 - 25113.6.2 Java平台的本地化类
 - 25113.6.3 资源束
 - 25213.6.4 日期与数字的格式化
- 25313.7 登录示例程序的国际化与本地化
 - 25413.7.1 收集区域敏感数据
 - 25513.7.2 制作缺省资源文件
 - 25513.7.3 改编页面文件
 - 25613.7.4 编写和配置Servlet过滤器
 - 25813.7.5 构建和运行应用程序
 - 25913.7.6 编写中文资源文件
 - 25913.7.7 检验应用程序的运行效果
- 26013.8 图片的国际化
 - 26113.8.1 在项目中建立存放图片的目录
 - 26213.8.2 制作对应各种不同语言的图片
 - 26213.8.3 在页面文件上添加图片
 - 26213.8.4 在资源束文件中加上对应图片键-值对
 - 26313.8.5 在build.xml中添加拷贝图片的内容
 - 26313.8.6 观察应用程序的运行结果
- 26413.9 异常消息的国际化
 - 26413.9.1 异常消息的种类
 - 26513.9.2 标准异常消息
 - 26513.9.3 自定义异常消息的处理过程
- 26613.10 异常消息国际化示例
 - 26613.10.1 在资源文件中加上消息的键-值对
 - 26613.10.2 创建消息工厂
 - 26713.10.3 业务模型中的消息处理方式
 - 26713.10.4 Backing Bean中的消息处理方式
 - 26713.10.5 示例应用的运行效果
- 26913.11 日期的国际化示例
 - 26913.11.1 示例应用要实现的效果
 - 26913.11.2 建立示例应用的目录结构
 - 26913.11.3 修改页面文件
 - 27013.11.4 准备资源文件
 - 27013.11.5 修改业务对象
 - 27113.11.6 修改业务模型中的类文件
 - 27113.11.7 修改Backing Bean
 - 27213.11.8 国际化编程后的效果检验
- 274第14章 JSF与Tiles框架的集成
 - 27514.1 页面布局的几种可选方案

- 27514.1.1 使用JSP指令的方法
- 27514.1.2 使用JSP标签
- 27914.1.3 使用Tiles框架
- 27914.2 Tiles框架简介
- 28014.3 用Tiles布局页面的示例
 - 28014.3.1 建立示例应用的目录结构
 - 28014.3.2 下载和安装与Tiles有关的库文件
 - 28014.3.3 编写tiles的配置文件
 - 28114.3.4 在web.xml文件中配置tiles
 - 28114.3.5 编写布局模板文件
 - 28214.3.6 编写布局的各组成部分的页面文件
 - 28314.3.7 编写应用模板的页面文件
 - 28414.3.8 示例应用的运行效果
- 28414.4 用Tiles布局Frame页面
 - 28514.4.1 Frame页面的基础知识
 - 28514.4.2 一个简单的Frame页面例子
 - 28614.4.3 让Frame页面居中显示
- 28814.5 用Tiles布局Frame页面的示例
 - 28914.5.1 建立示例应用的目录结构
 - 28914.5.2 新建图片文件
 - 28914.5.3 更改tiles配置
 - 28914.5.4 新建frame布局模板页面
 - 28914.5.5 新建用于导航的菜单页面
 - 29014.5.6 新建带frame的页面
 - 29114.5.7 应用程序的运行效果
- 291第15章 JSF访问数据库
 - 29315.1 MS SQL Server数据库的安装
 - 29315.1.1 SQL Server数据库的安装过程
 - 29315.1.2 数据库补丁包的安装
 - 29615.2 JDBC简介
 - 29715.2.1 JDBC的发展历程
 - 29715.2.2 JDBC的结构和工作原理
 - 29815.2.3 JDBC驱动器
 - 30015.2.4 JDBC的使用场景
 - 30015.2.5 两层模式
 - 30015.2.6 三层模式
 - 30115.2.6.1 客户层
 - 30115.2.6.2 中间层
 - 30115.2.6.3 数据源层
 - 30215.3 数据访问对象模式简介
 - 30215.4 让登录程序访问数据库中用户信息
 - 30215.4.1 建立数据库
 - 30215.4.2 建立数据库表
 - 30315.4.3 在表中输入实验用的数据
 - 30415.4.4 下载和安装JDBC驱动程序
 - 30415.4.5 建立应用程序的目录结构
 - 30515.4.6 增加应用程序访问数据库的部分
 - 30515.4.7 修改应用程序的业务逻辑部分
 - 30715.4.8 修改backing bean使其使用新的service

- 30815.4.9 创建与数据访问相关的自定义异常类
- 30915.4.10 修改资源文件，添加新的键-值对
- 31115.4.11 配置应用程序
- 31115.4.12 运行应用程序
- 313第16章 应用程序中的异常处理
 - 31516.1 Java异常的分类
 - 31516.2 Java处理异常的一般方法
 - 31716.3 创建自己的异常子类
 - 32016.4 成链异常
 - 32116.5 JSF应用程序的异常处理
- 322第17章 应用程序的日志
 - 32417.1 为什么要记日志
 - 32417.2 用什么记日志
 - 32417.3 怎么记日志
 - 32417.4 在登录示例程序中记日志
 - 32617.4.1 建立示例应用目录结构
 - 32617.4.2 下载通用日志包
 - 32617.4.3 下载日志器实现Log4J
 - 32717.4.4 写日志有关的配置文件
 - 32717.4.5 创建一个空的日志文件
 - 32917.4.6 编写访问日志接口的应用程序代码
 - 32917.4.7 部署和运行应用程序
- 331第18章 JSF与AJAX集成
 - 33418.1 AJAX概述
 - 33418.1.1 什么是AJAX
 - 33418.1.2 AJAX的原则
 - 33418.1.3 AJAX包含那些技术
 - 33418.1.4 AJAX的工作原理
 - 33518.2 Ajax4jsf
 - 33618.2.1 Ajax4jsf概述
 - 33618.2.2 Ajax4jsf工作原理
 - 33618.2.3 下载和安装Ajax4jsf
 - 33718.3 示例应用程序
 - 33818.3.1 建立应用程序的目录结构
 - 33918.3.2 需要的库文件
 - 33918.3.3 在web.xml文件中添加Ajax4jsf的过滤器
 - 33918.3.4 在注册页面中使用Ajax4jsf标签
 - 33918.3.5 在Backing Bean中添加动作监听器方法
 - 34018.3.6 部署和运行应用
- 340第19章 ORM及其Hibernate实现
 - 34119.1 ORM简介
 - 34119.1.1 对象模型与关系范例的失配
 - 34119.1.1.1 粒度问题
 - 34119.1.1.2 子类型问题
 - 34119.1.1.3 标识符问题
 - 34219.1.1.4 关联问题
 - 34219.1.1.5 对象图导航问题
 - 34319.1.2 集成层的可选方案
 - 34319.1.2.1 使用JDBC

- 34319.1.2.2 使用EJB
- 34319.1.2.3 使用面向对象的数据库系统
- 34419.1.2.4 使用对象/关系映射 (ORM)
- 34419.1.3 ORM面临的问题
- 34419.2 Hibernate的结构简介
 - 34419.2.1 核心应用程序编程接口
 - 34519.2.2 回调接口
 - 34719.2.3 扩展接口
- 34819.3 映射对象
 - 34819.3.1 基本映射
 - 34819.3.2 标识的属性
 - 34919.3.3 标识生成器
 - 35019.3.4 多对一映射/一对多映射
 - 35019.3.5 多对一中的类示例
 - 35019.3.6 多对一/一对多的表间关系
 - 35319.3.7 多对一/一对多的映射文件
 - 35319.3.8 多对一元素的主要属性
 - 35419.3.9 集合映射
 - 35519.3.10 外键元素
 - 35519.3.11 一对多元素的主要属性
 - 35519.3.12 多对多映射
- 35519.4 检索对象
 - 35719.4.1 Hibernate查询语言
 - 35719.4.2 条件查询
 - 35819.4.3 SQL查询
- 35819.5 报表查询
- 358第20章 JSF与Hibernate的集成
 - 36020.1 Hibernate的配置
 - 36020.1.1 Hibernate的下载
 - 36020.1.2 需要的库文件
 - 36120.1.3 持久化类的映射文件
 - 36120.1.4 Hibernate配置文件
 - 36220.2 用Hibernate重写登录程序的数据库访问部分
 - 36220.2.1 辅助工具类
 - 36220.2.2 项目使用的数据库
 - 36320.3 具体的实现
 - 36320.3.1 创建应用程序项目目录
 - 36320.3.2 修改build.xml文件及调整部分配置文件的目录
 - 36320.3.3 准备必要的Java库文件
 - 36320.3.4 创建辅助工具类
 - 36320.3.5 创建持久类及映射文件
 - 36420.3.6 创建Hibernate配置文件
 - 36420.3.7 创建DAO接口
 - 36520.3.8 创建DAO的Hibernate实现类
 - 36620.3.9 创建应用服务的Hibernate实现类
 - 36720.3.10 修改Backing Bean
 - 36820.3.11 用ANT构建和部署应用程序
 - 36820.3.12 运行应用程序
 - 36920.4 对应用程序的回顾

- 36920.4.1 关于Hibernate Session的关闭问题
- 36920.4.2 关于示例程序中的修改问题
- 36920.4.3 关于DAO代码简化的问题
- 37020.5 进一步简化DAO代码
 - 37020.5.1 DAO中的样板代码
 - 37020.5.2 去除样板代码的方法
- 37120.5.3 部署和运行新项目
- 373第21章 JSF与Spring和Hibernate的集成
 - 37421.1 Spring的组成
 - 37421.2 Spring中的IoC以及用Spring管理JSF托管bean
 - 37521.2.1 JSF与Spring集成的原理
 - 37521.2.2 集成所需要做的配置工作
 - 37521.2.3 早期版本的JSF参考实现在实际集成中会遇到的问题
 - 37721.2.4 项目使用的数据库
 - 37821.3 用JSF、Spring和Hibernate编写注册登录示例
 - 37821.3.1 建立应用程序项目的目录结构
 - 37821.3.2 准备Spring库文件
 - 37821.3.3 准备JSF与Spring的源代码库文件
 - 38021.3.4 创建数据访问接口
 - 38121.3.5 创建数据访问实现类
 - 38121.3.6 创建业务服务接口
 - 38221.3.7 创建业务服务实现类
 - 38221.3.8 创建持久化类及映射文件
 - 38421.3.9 创建Backing Bean
 - 38521.3.10 创建注册页面及注册成功页面
 - 38521.3.11 在Spring上下文中配置服务及数据访问对象
 - 38621.3.12 配置Faces上下文
 - 38821.3.13 配置web.xml
 - 38821.3.14 构建、部署及运行示例应用程序
- 389第22章 使用Eclipse和MyEclipse工具
 - 39022.1 下载与配置Eclipse和MyEclipse
 - 39022.1.1 下载IDE软件
 - 39022.1.2 配置开发环境
 - 39122.2 用IDE开发示例
 - 39422.2.1 建立新项目
 - 39422.2.2 加入JSF Capabilities
 - 39622.2.3 加入Spring Capabilities
 - 39722.2.4 加入Hibernate Capabilities
 - 39822.2.5 业务层与集成层对象的编写与配置
 - 40222.2.5.1 引入访问数据库所需要的包文件
 - 40222.2.5.2 配置数据源
 - 40322.2.5.3 创建持久化对象及其映射文件
 - 40422.2.5.4 修改Hibernate sessionFactory的配置
 - 40522.2.5.5 创建数据访问对象的接口及实现类
 - 40822.2.5.6 创建应用服务接口及实现类和自定义异常类
 - 40822.2.5.7 配置其他对象
 - 40822.2.6 应用程序表示层对象的编写与配置
 - 40922.2.6.1 配置Application
 - 41022.2.6.2 配置和生成Backing Bean

- 41022.2.6.3 在Backing Bean中注入依赖
- 41122.2.6.4 在Backing Bean中编写动作方法
- 41222.2.6.5 可视化方式制作页面文件
- 41222.2.6.6 配置导航规则
- 41222.2.7 配置应用程序的部署描述器文件web.xml
- 41322.2.8 部署和运行项目
- 41522.3 用IDE打开示例应用的方法
- 41722.4 在IDE中使用Hibernate的反向工程
 - 41722.4.1 创建一个新项目
 - 41722.4.2 创建数据库
 - 41722.4.3 创建数据库浏览器窗口
- 41822.4.4 建立必要的包
- 42022.4.5 加入JSF和Spring的Capabilities
- 42022.4.6 加入Hibernate的Capabilities
- 42022.4.7 应用反向工程生成类和配置文件
- 42322.4.8 反向工程生成的文件样例
- 425第23章 AOP及事务管理
 - 42723.1 AOP简介
 - 42723.1.1 什么是AOP
 - 42723.1.1.1 对AOP的一些常见的误解
 - 42723.1.1.2 AOP的一些常用术语
 - 42823.1.2 Spring实现的AOP
 - 42823.1.2.1 Spring实现的AOP的特点
 - 42823.1.2.2 Spring中AOP的应用
 - 42923.2 事务简介
 - 42923.2.1 事务的基本属性
 - 42923.2.2 Spring的事务策略
 - 42923.2.3 Spring支持的事务隔离度
 - 43023.2.4 Spring支持的事务传播类型
 - 43023.3 事务管理示例
 - 43123.3.1 示例应用要演示的情形
 - 43123.3.2 数据库表的建立
 - 43123.4 没有事务管理的例子
 - 43223.4.1 编写持久化对象的类文件
 - 43223.4.2 编写映射文件
 - 43324.4.3 编写DAO接口
 - 43423.4.4 编写DAO实现类
 - 43523.4.5 编写业务服务接口
 - 43623.4.6 编写没有事务处理的业务服务
 - 43623.4.7 编写没有事务管理的Spring配置
 - 43823.4.8 示例应用的Backing Bean
 - 43923.4.9 配置web.xml
 - 44123.4.10 应用程序的页面文件
 - 44123.4.11 示例应用的运行效果
 - 44123.5 编程式事务管理的例子
 - 44223.5.1 编程式事务管理的一般方法
 - 44223.5.2 编写编程式事务的业务服务
 - 44223.5.3 编写编程式事务管理的Spring配置
 - 44423.5.4 修改web.xml

- 44523.5.5 应用的运行效果
- 44523.6 声明式事务管理的例子
 - 44523.6.1 声明式事务管理的方法
 - 44523.6.2 在Spring的配置文件中声明事务
 - 44622.6.3 修改web.xml
 - 44723.6.4 应用的运行效果
- 447第24章 权限系统示例应用程序
 - 44824.1 基于角色的访问控制
 - 44824.2 应用程序的界面
 - 44824.2.1 登录界面
 - 44924.2.2 管理员用户界面
 - 44924.2.3 普通用户界面
 - 45024.3 数据库设计
 - 45124.3.1 表的结构
 - 45124.3.2 表之间的关系
 - 45424.4 持久化对象
 - 45524.4.1 持久化对象的类文件
 - 45524.4.2 持久化对象的映射文件
 - 46124.5 数据访问对象
 - 46324.5.1 数据访问对象接口
 - 46424.5.2 数据访问对象实现
 - 46524.6 业务服务
 - 46924.6.1 业务服务接口
 - 46924.6.2 业务服务实现
 - 47024.6.3 编程式事务在示例中的应用
 - 47324.7 Backing Bean
 - 47424.7.1 使用Backing Bean的策略
 - 47424.7.2 示例Backing Bean的编码
 - 47524.7.3 安全退出系统的动作
 - 47824.7.4 在Backing Bean中验证Email地址
 - 47824.7.5 在Backing Bean中引用其他Backing Bean
 - 48024.8 配置文件
 - 48224.8.1 应用程序部署描述器文件
 - 48224.8.2 JSF配置文件
 - 48324.8.3 Spring配置文件
 - 48624.8.4 Hibernate的配置文件
 - 48924.9 页面文件
 - 48924.9.1 index.jsp
 - 48924.9.2 登录页面文件
 - 48924.9.3 Frameset文件
 - 49024.9.4 树型菜单文件
 - 49124.9.5 添加新用户的页面文件
 - 49224.9.6 指派用户角色的页面文件
 - 49224.9.7 安全登出系统的页面文件
 - 49524.9.8 显示操作结果的页面文件
 - 49524.9.9 页面间的导航关系
 - 49624.10 辅助文件
 - 49624.10.1 字符集过滤器类文件
 - 49624.10.2 资源文件

- 49724.10.3 消息工厂类文件
- 49724.11 怎样让示例程序运行起来
 - 49724.11.1 创建示例数据库
 - 49724.11.2 更改配置文件中的数据库密码
 - 49824.11.3 用IDE部署示例应用
- 49824.12 为已有的JSF项目增加AJAX支持
 - 49824.12.1 导入库文件
 - 49824.12.2 更改页面文件
 - 49824.12.3 更改Backing Bean
 - 50024.12.4 更改业务服务接口及实现
 - 50124.12.5 更改应用程序部署描述器文件
- 502第25章 办公用品管理系统示例程序
 - 50325.1 示例程序的需求分析
 - 50325.1.1 需求分析简述
 - 50325.1.2 用例 (Use Case) 图
 - 50325.1.3 用例文档的书写
 - 50425.1.4 系统需要用到的页面
 - 50525.1.5 业务规则的说明
 - 50625.1.6 其他要求
 - 50625.2 示例程序的架构设计
 - 50625.2.1 多层架构
 - 50625.2.2 选择合适的技术
 - 50625.2.3 表示层设计
 - 50725.2.4 业务逻辑层设计
 - 50725.2.5 数据映射层设计
 - 50725.2.6 数据库设计
 - 50725.3 示例程序的实现
 - 50825.3.1 表示层实现
 - 50825.3.2 业务层实现
 - 51025.3.3 数据映射层实现
 - 51225.4 其他实用功能的实现
 - 51225.4.1 文件上传
 - 51225.4.1.1 Java中的输入输出与流
 - 51225.4.1.2 文件的路径与目录
 - 51225.4.1.3 MyFaces的文件上传标签
 - 51325.4.2 文件上传的实现
 - 51325.4.3 处理分页
 - 51825.4.4 发送电子邮件
 - 52125.4.4.1 Spring对电子邮件的抽象
 - 52125.4.4.2 编写Backing Bean
 - 52125.4.4.3 配置邮件发送器和邮件消息
 - 52325.4.4.4 编写发送邮件的方法
 - 52425.4.4.5 发送邮件时需要加入的库文件
 - 52525.5 示例应用程序的测试
 - 52525.5.1 测试简介
 - 52525.5.2 测试的分类
 - 52625.5.3 模拟测试与集成测试
 - 52625.5.4 常用的测试框架JUnit
 - 52625.5.5 测试用例的运行方式

- 52725.5.6 数据访问层测试示例
 - 52825.5.6.1 创建测试基类
 - 52825.5.6.2 创建测试具体类
 - 52925.5.6.3 修改向导生成的测试类文件
 - 53125.5.6.4 测试过程
 - 53225.5.6.5 在IDE中测试易出现的问题
 - 53325.6 部署和运行应用程序
 - 53325.6.1 数据库的建立
 - 53325.6.2 配置文件中的参数更改
 - 534第26章 应用程序的安全性
 - 53526.1 应用程序的安全性简介
 - 53526.2 权限管理示例程序的安全性考虑
 - 53526.2.1 控制对资源的访问
 - 53526.2.2 保护敏感数据
 - 53626.3 防止用户绕过登录的方法
 - 53726.3.1 登录时保存信息到HttpSession中
 - 53726.3.2 制作验证HttpSession中内容的页面
 - 53926.3.3 在需要保护的页面前部包含验证页面
 - 53926.4 Java密码架构和加密服务提供者
 - 54026.4.1 Java密码架构
 - 54026.4.2 加密服务提供者
 - 54126.4.3 Java密码架构的使用方式
 - 54126.5 消息摘要与口令认证
 - 54126.5.1 什么是消息摘要
 - 54126.5.2 口令的加密存储
 - 54226.5.3 加密后的口令的验证
 - 54326.6 权限管理应用的口令加密示例
 - 54426.6.1 新建示例项目
 - 54426.6.2 更改人员表中口令字段的数据类型
 - 54426.6.3 改变持久化类中口令字段的数据类型
 - 54426.6.4 更改映射文件中口令字段的数据类型
 - 54526.6.5 在业务服务实现中增加口令加密方法
 - 54626.6.6 在业务服务实现中增加口令验证方法
 - 54726.6.7 更改业务服务实现的注册新用户方法
 - 54826.6.8 更改业务服务实现的登录方法
 - 54926.6.9 更改业务服务实现的用户改变口令方法
 - 55126.6.10 在数据库表中查看口令数据
- 552

《JSF第一步》

编辑推荐

《JSF第一步：JSF+Spring+Hibernate+AJAX编程》的特点主要是从编程实践的角度来介绍这些框架的基本原理，以及用示例演示如何具体应用这些框架来开发一个Web应用程序。在介绍各种框架时，对涉及到的Web开发中容易遇到的疑难问题尽可能给出浅显易懂的解释，其目的是帮助想利用这些框架进行开发的人员尽快进入实战状态。 《JSF第一步：JSF+Spring+Hibernate+AJAX编程》附光盘1张。

精彩短评

- 1、老子的教科书——||
- 2、怎么说呢 中国人写的书 还算行了.....凑合能看 毕竟解决了我的一些疑惑.....

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:www.tushu000.com