

<<JSF2和RichFaces4使用指南>>

图书基本信息

书名：<<JSF2和RichFaces4使用指南>>

13位ISBN编号：9787121177378

10位ISBN编号：7121177374

出版时间：2012-9

出版时间：电子工业出版社

作者：韩陆

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>

<<JSF2和RichFaces4使用指南>>

前言

前言本书是笔者在过去两年多的时间里，从事基于JSF和RichFaces开发的工作总结和心得。

作为一名开发人员，能有机会和同行分享技术，笔者非常珍惜和欢喜。

故而在动笔之前告诫自己要珍惜每一页纸，使用简练的语言讲述更丰富和更深刻的内容。

书中的每个技术点都提供了笔者的经验之谈和详实的示例；与此同时，在示例的文字部分笔者对代码做了斟酌处理，以示说明，而将完整的代码内容放到了本书的源代码部分中。

目的是使本书物有所值，相信读者能体会其中的良苦用心。

本书使用的JSF版本为2.1，RichFaces版本为4.2.1。

这份源代码会有后续更新，目的是保证其时效性和精准性。

一本技术书籍纵然是经典，也有很强的时效性，有时软件版本的升级会给讲解的软件版本已经陈旧的书带来灭顶之灾。

因此日后笔者会在JSF和RichFaces升级后，及时提供更新的源代码，以延长时效性。

对于精准性，每个人都会犯错。

笔者实是庸才，必然鄙陋之处甚多。

但笔者可以保证的是一旦发现源代码中有错误，将会在第一时间更新源代码。

若是书中文字有误，会在源代码中附上勘误表。

如果读者有兴趣，可以使用git通过gitHub管理的项目来更新本书源代码。

当然你也可以通过浏览器访问并直接下载，附录A中有关于本书源代码的详细说明。

本书是本经验谈，笔者竭尽全力并毫无保留地收录了两年多来一线工作中使用JSF和RichFaces的经验和教训。

希望能使读者少走弯路，快速掌握这门技术。

传递经验主要传递的是学习方法，笔者在编写本书时常问自己，一个新手如何能尽快学习繁多的组件？

首先是归类，一旦脑海中对一事物有了清晰的分类，就会快速各个击破；其次是统一，人们习惯在做一件新的事情时采用已有的经验作为锚点，因此笔者将使用如下统一的方式讲述每一个组件。

(1) 介绍该组件的功能和相关组件。

(2) 从感观上认识该组件。

(3) 给出使用样例，使读者快速了解如何使用该组件。

笔者会在组件第1次出现时介绍一遍其公共属性，书中知识点重现时会有导读，以使读者减少重复阅读，提升阅读质量。

笔者会将很多心得体会融入组件讲解之中，而不是简单地罗列其特性。

本书以JSF和RichFaces的特性为单位，逐一讲述其原理，以及基本和高级应用方法；同时，由点到面以3个环环相扣的示例为牵引，希冀读者融会贯通。

本书不是JSF和RichFaces的API大全或者速查手册，因此不会罗列每一个组件的每个属性。

而是从实用角度讲述组件的常用和重要的技术点，读者能看到的是如何最快并最准确地使用JSF和RichFaces。

本书只关注JSF和RichFaces，如果必须交代相关技术，则将其收录在附录中。

读者看不到庞杂的内容，笔者希望为读者提供一个简单且精力集中的学习氛围。

笔者最大的心愿就是让本书成为读者有机会还愿意回头再翻翻的书籍。

面向读者本书面向的读者需要至少了解如下知识。

(1) Java的基本使用方法、Eclipse和Tomcat的初级使用经验。

(2) HTML基本常识，包括CSS和JavaScript的基本常识。

读者最好具备的知识储备为servlet和JSP开发经验，以及http协议和Ajax基本常识。

阅读指南本书包括3个部分共7章。

第1部分是入门，分3章引领读者走进JSF和RichFaces。

第1章介绍JSF和RichFaces，并说明如何开始用其开发项目，读者可通过第1个实例学习如何快速地开发

<<JSF2和RichFaces4使用指南>>

一个Web项目。

第2章从JSF的组件讲起，这是因为JSF是面向组件开发的。

然后介绍JSF的Facelet视图和模板，这部分将展现JSF的快速开发和高度复用特点。

接下来说明容器管理，读者可从中了解JSF页面和Java类之间的联系。

最后讲解JSF的转换器和验证器等功能性组件，以及JSF的脚本库。

第3章从RichFaces的组件讲起，读者从中可以清晰地了解RichFaces组件的强大和易用性。

然后讲述RichFaces对JSF的增强，并针对RichFaces3.3.3升级到RichFaces4.x提供指导性说明。

在本章最后读者可看到一个全面使用JSF和RichFaces基本功能的示例，其中引入了许多笔者实际的技术成果。

第2部分讲解原理，其中只有一章，是本书的核心章节和笔者漫长的学习过程中总结的结晶。

本章讲述JSF如何工作，以及RichFaces如何与JSF结合并扩展。

第3部分是进阶部分，分为3章来讲述JSF和RichFaces的高级应用，其中的内容对实际开发项目更有指导意义。

第5章展现JSF自定义组件的能力，其中第1节从原理上讲述如何自定义一个组件；第2节讲述JSF2.0推出的composite component，这是一种使用Facelet页面定义组件的方式，其特点是快捷和易用；第3节讲述如何使用RichFaces提供的自定义组件工具CDK来创建组件。

3个小节都提供了示例，方便学习。

第6章首先讲述如何利用JSF提供的配置参数调优性能，接下来讲述RichFaces的性能调优。

然后讲述JSF错误处理机制，最后是锦上添花的皮肤、主题的使用，这部分对提升用户体验很有意义。

第7章收录了笔者在基于RichFaces开发中总结的实用技术，以及笔者开发、维护和使用近两年的JSF实例知识管理系统（KMS），希冀这一示例使读者在推敲过后掌握JSF和RichFaces。

学习是一种生活习惯，很多人走出校园后因为种种原因慢慢失去了这个习惯，也因此丧失了对新事物的热爱。

作为一名开发人员，学习是一种生存习惯。

学习会贯穿职业生涯的始终，甚至是毕生。

学习，对于开发人员不是一种技能，而是一种情怀。

我们不会因为掌握了一门新技术而骄傲，我们会产生发自心底的欣慰。

笔者才疏学浅，疏漏之处祈盼读者不吝指教。

<<JSF2和RichFaces4使用指南>>

内容概要

本书以JSF和RichFaces的特性为单位，逐一讲述其基本使用、原理和高级应用；同时，由点到面地以3个环环相扣的示例为牵引，希冀从实用角度，讲述该组件的常用和重要的技术点，书中的每个技术点都提供了作者的经验之谈和翔实的示例。

本书不仅是一本技术书，更是一本经验谈。

作者把自己两年多在一线工作中对JSF和RichFaces的使用经验和教训，以及学习方法通过本书与读者分享，希望能帮助读者最快和最准确地使用JSF和RichFaces，并且通过本书的学习方法能迅速学会更多有意思及有用的框架。

<<JSF2和RichFaces4使用指南>>

书籍目录

第1部分 入门

第1章 绪论2

1.1 JSF简介2

1.2 RichFaces简介4

1.3 第1个例子7

1.3.1 创建JSF工程7

1.3.2 创建RichFaces工程12

1.3.3 tinylibrary 1.013

1.4 本章小结17

第2章 JSF入门18

2.1 JSF组件库18

2.1.1 标准html库18

2.1.2 标准核心库19

2.1.3 页面组件21

2.1.4 输入组件26

2.1.5 输出组件33

2.1.6 行为组件38

2.1.7 元数据组件43

2.1.8 多语言组件43

2.1.9 功能性组件44

2.2 JSF视图52

2.2.1 视图定义语言 (VDL) 52

2.2.2 模板的设计和使用53

2.3 容器管理56

2.3.1 ManagedBean56

2.3.2 Unified EL59

2.3.3 flash62

2.4 JSF转换器63

2.4.1 转换器组件63

2.4.2 数字转换组件65

2.4.3 日期时间转换组件66

2.5 JSF验证器67

2.5.1 验证器组件67

2.5.2 必输项验证组件69

2.5.3 长度验证组件69

2.5.4 长整型范围验证组件70

2.5.5 浮点型范围验证组件70

2.5.6 正则验证组件70

2.5.7 JSR303和Bean验证组件70

2.6 JSF脚本75

2.6.1 jsf名字空间75

2.6.2 jsf.ajax名字空间75

2.6.3 jsf.util名字空间76

2.7 JSF变量76

2.8 本章小结78

<<JSF2和RichFaces4使用指南>>

第3章 RICHFACES入门79

3.1 RichFaces组件库79

3.1.1 行为组件79

3.1.2 jQuery82

3.1.3 轮询组件82

3.1.4 输入组件84

3.1.5 选择组件101

3.1.6 菜单组件106

3.1.7 验证组件112

3.1.8 输出组件116

3.1.9 迭代组件128

3.1.10 集合组件129

3.1.11 树组件136

3.1.12 拖动组件139

3.1.13 功能输出组件141

3.1.14 监听组件143

3.2 对JSF的增强144

3.2.1 异步请求组件144

3.2.2 异步输出容器147

3.2.3 异步队列148

3.2.4 js面向切面函数148

3.2.5 异步区域组件150

3.2.6 异步脚本函数组件150

3.2.7 请求队列组件151

3.2.8 RichFaces脚本函数155

3.2.9 其他组件156

3.3 推送技术158

3.3.1 实现模式158

3.3.2 配置环境160

3.3.3 TopicsContext实践162

3.3.4 CDI实践163

3.3.5 JMS实践164

3.3.6 浏览器支持166

3.4 从RichFaces 3.x升级167

3.4.1 Ajax相关167

3.4.2 样式表167

3.4.3 组件类167

3.4.4 组件标签及属性变化167

3.5 第2个例子167

3.6 本章小结173

第2部分 原理

第4章 JSF原理176

4.1 请求处理176

4.1.1 生命周期176

4.1.2 傲游全程177

4.1.3 JSF转换机制186

4.1.4 JSF验证机制190

<<JSF2和RichFaces4使用指南>>

- 4.1.5 立即属性192
- 4.2 视图导航192
 - 4.2.1 JSF与REST192
 - 4.2.2 隐式导航193
 - 4.2.3 基于规则的导航194
 - 4.2.4 forward和redirect198
 - 4.2.5 PRG200
- 4.3 事件模型204
 - 4.3.1 事件类型204
 - 4.3.2 监听器类型205
 - 4.3.3 事件触发206
- 4.4 异步请求206
 - 4.4.1 异步请求函数207
 - 4.4.2 全局函数208
 - 4.4.3 异步请求处理流程210
 - 4.4.4 异步应答对象211
 - 4.4.5 服务器端处理213
- 4.5 RichFaces原理214
 - 4.5.1 框架结构214
 - 4.5.2 异步增强216
- 4.6 下载应用220
- 4.7 本章小结221
- 第3部分 进阶
- 第5章 自定义组件224
 - 5.1 custom component224
 - 5.2 composite component231
 - 5.3 使用CDK自定义组件235
 - 5.3.1 UI组件类236
 - 5.3.2 Renderer类236
 - 5.3.3 tablib标签符236
 - 5.3.4 图画组件236
 - 5.3.5 CDK名字空间240
 - 5.3.6 package-info243
 - 5.4 本章小结244
- 第6章 配置与优化245
 - 6.1 JSF配置参数245
 - 6.2 JSF配置文件249
 - 6.3 资源管理251
 - 6.3.1 资源优化252
 - 6.3.2 资源映射253
 - 6.4 错误处理254
 - 6.4.1 统一错误导航255
 - 6.4.2 自定义异常处理257
 - 6.4.3 异步请求异常处理259
 - 6.5 皮肤和ECSS259
 - 6.5.1 皮肤主题259
 - 6.5.2 ECSS261

<<JSF2和RichFaces4使用指南>>

| | |
|----------------|-----|
| 6.5.3 自定义主题 | 262 |
| 6.5.4 配置参数 | 265 |
| 6.6 本章小结 | 266 |
| 第7章 技巧和总结 | 267 |
| 7.1 实战收录 | 267 |
| 7.1.1 分页实现 | 267 |
| 7.1.2 数据表多选 | 273 |
| 7.1.3 提示输入框 | 274 |
| 7.2 知识管理系统示例 | 275 |
| 7.2.1 依赖资源 | 275 |
| 7.2.2 功能 | 277 |
| 7.3 展望与总结 | 279 |
| 附录A 本书源代码 | 280 |
| 附录B 实例的开发和调试环境 | 281 |
| 附录C Maven简明指南 | 283 |
| 附录D 参考资料 | 288 |

<<JSF2和RichFaces4使用指南>>

章节摘录

版权页：插图：恢复视图阶段的主要任务是建立或恢复视图（UIViewRoot）组件树对象，建立视图对象的策略如下。

（1）视图对象不存在 该阶段会根据Faces的上下文变量facesContext创建一个视图组件树，参考com.sun.faces.application.view.ViewHandlingStrategy.createView方法。

刚创建的UIViewRoot对象是视图组件树的树根，这棵树还没有枝干。

如果元数据，如页面参数等变量在参数UIViewRoot.METADATA FACET NaME（javaxfaces_metadata）没有定义，视图处理流程将不再继续。

又因为组件树是空的，树上的组件不需要更新数据，没有验证的组件和需要更新的数据模型，也没有事件需要处理，所以流程应当直接进入请求处理生命周期的第6步。

此处的处理是调用上下文变量的渲染应答方法facesContext.renderResponse0，该方法的目的是告诉JSF直接进入渲染应答阶段。

生命周期控制类LifecycleImpl的execute方法在每执行下一阶段之前都要判断是否有必要让流程进入到第6阶段，即渲染阶段。

由此可知，第2阶段～第5阶段都是可选的，而非必须经历的。

LifecycleImpl类的execute方法片段2如下：（2）视图对象存在 判断是否是postback，如果不是postback的请求if（！

facesContext.isPostbackO），也会执行上下文变量的渲染应答方法facesContext.renderResponse（）；；否则恢复视图，完成本阶段事务进入第2阶段。

postback的概念可以理解为当前页面是否为首次进入，如果是首次进入，该值为false：如果从该页再次发起请求，由于页面已经被渲染过一次，所以该值为true。

判断依据是请求参数中是否包含javax.faces.ViewState参数，其实现可以参考ResponseStateManagerImpl类。

ViewState概念从JSF 2.0版本开始出现，其实这个概念在.net中已经存在多年。

其作用就是标识当前页面是否被刷新过，一旦被刷新，其值就会发生改变。

ViewState示例如下：根据以上分析，在本例中首次进入begin.xhtml时第1阶段会造一棵空树。

然后将流程送入第6阶段，即渲染应答阶段。

<<JSF2和RichFaces4使用指南>>

编辑推荐

《JSF2和RichFaces4使用指南》编辑推荐：实例实战，组件的调用详解。

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>