

全面讲解基础知识、典型案例与实战进阶，帮助读者快速、全面提升技能

盛华 编著

JavaServer Faces 2.0

实用详解

本书特色

由浅入深：本书内容详略得当，循序渐进，适合初学者学习使用

面向实战：书中包括了大量实用又有代表性的实例和典型应用，内容丰富，系统性强

清华大学出版社

全面讲解基础知识、典型案例与实战进阶，帮助读者快速、全面提升技能

盛华 编著

JavaServer Faces 2.0

实用详解

本书特色

由浅入深：本书内容详略得当，循序渐进，适合初学者学习使用

面向实战：书中包括了大量实用又有代表性的实例和典型应用，内容丰富，系统性强

清华大学出版社
北 京

内 容 简 介

本书阐述使用 NetBean 7.0 整合开发环境 (IDE) 工具和 Glassfish 3.1 网络服务器开发动态跨浏览器的网络应用。全书系统而完整地介绍了 JSF 2.0 最新技术, 包括: 请求处理生命周期、三种托管 Bean、上下文和依赖注射 (CD)、网页隐式和显式导航、组件开发 (比如复合组件和自定义 UI 组件)、Ajax、数据转换与验证、国际化、JDBC 数据库连接池和持久架构 (JPA 2.0)、网络安全各种认证和安全连接 (SSL) 等; 最后介绍了综合实例, 即溧阳虚拟大学。

本书适合于开发 Java 企业应用的软件工程师, 也可作为高等学校 Java EE 6 的参考书, 以及作为相关培训和自学的教材。

本书封面贴有清华大学出版社防伪标签, 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 2.0 实用详解 / 盛华编著. —北京: 清华大学出版社, 2013. 1

ISBN 978-7-302-29117-6

I. ①J… II. ①盛… III. ①JAVA 语言-程序设计 IV. ①TP312

中国版本图书馆 CIP 数据核字 (2012) 第 132035 号

责任编辑: 袁金敏

封面设计: 陈晓兵

责任校对: 徐俊伟

责任印制: 何 芊

出版发行: 清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址: 北京清华大学学研大厦 A 座 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: 185mm×260mm 印 张: 32.25 字 数: 809 千字

版 次: 2013 年 1 月第 1 版 印 次: 2013 年 1 月第 1 次印刷

印 数: 1~4000

定 价: 69.00 元

产品编号: 044383-01

FOREWORD

前言

2009 年 12 月 Java 社区正式发布了 Java EE 6, Glassfish V3 和 NetBeans 6.8。它们分别是 Java 网络开发软件、网络服务器和整合开发环境工具,三者互相配合,以崭新的面貌呈现在 Java 网络开发者面前。“信息风云变幻”,Java 网络开发者要以“一万年太久,只争朝夕”的精神更新自己的知识,否则就会落伍,有可能被淘汰。尤其是在大学教 IT 的老师,要密切注视 IT 的发展动向,否则,就会误人子弟,造成自己的学生毕业后找不到工作的尴尬局面。这不是危言耸听,这是挑战,这是竞争。

其实,Java EE 6 包括两大部分内容:第一部分是网络表现层,它包括 Java Servlet 3.0, JavaServer pages (JSP 2.2), JavaServer Faces (JSF 2.0) 等;第二部分是商务逻辑层,它包括 Java 持久架构 (Java Persistence Application, 即 JPA 2.0), 企业 Beans (EJB 3.1), 和网络服务 (JAX-WS 2.2 和 JAX-RS 1.1) 等。

2004 年,JSF 专家组发布了 JSF 1.0 版本指南和参考文献,不久纠正了原版的错误,又发布了 JSF 1.1 版本。2006 年发布了 JSF 1.2 版本,突出了清晰而方便的特点。JSF 有优雅的框架,它具有高度可扩展性,因此吸引了框架开发者,他们把一些开放的资源加进 JSF,诸如 facelets, Ajax4jsf, Seam, JSF 模板, Pretty Faces, RichFaces, ICEFaces 等。几乎上述框架的原始作者都参加了 JSF 2.0 专家组,经过实践磨炼,JSF 2.0 已经十分成熟,于 2009 年 12 月正式发布 JSF 2.0 版本。

从 JSF 1.2 到 JSF 2.0,JSF 技术向前跨了一大步。JSF 2.0 的主要新特点有:(1) Ajax,这个特点使得 JSF 视图从浏览器直接与服务器通信,而不要求浏览器窗口整页刷新,Ajax 是当前高质量网络应用的一个重要元素。(2) 复合组件,这个特点能从其他 JSF 组件组合成真实的 JSF 组件,这让你很容易创建自定义组件,让你现有组件重用,包括组件监听器、属性和事件。(3) 部分状态存储,部分状态存储大大地减少了内存,当开发自定义组件时,它大大地简化了处理状态的 API。(4) 视图参数化,对旧版本另一个抱怨是,所有网页之间的导航坚持要用 POST 请求,而视图参数化改善了它。

JSF 2.0 还有一些特点是高级 JSF 开发者感兴趣的:(1) 加强导航能力,包括可书签化,不需要 XML 导航规则的导航,即隐式导航;条件导航,支持 PRG 模型、刷新和运行时导航侦察。(2) 异常处理,JSF 2.0 有一个中心的并举处理器,所有异常都归流到它处理。这样很容易构造使用 JSF 组件的出错网页。(3) 增强表达语言,引进了几

个新的隐式对象，现在，表达语言支持对任何 Java 方法的调用，包括带有参数的方法。(4) 验证，新的 Java 指南包括验证，该指南很好地整合了 JSF 2.0，新增了 Bean 验证。(5) 新的有效域，除了新增对话、视图有效域外，JSF 2.0 为开发者提供了自定义有效域机制。

总的趋势是，它让开发者使用起来越来越方便，而功能越来越健全。JSF 2.0 以 Facelets 的视图技术 (xhtml) 替代了原来的 JSP；通过使用注解 (annotation) 和默认值减少了 XML 的配置；提供了一种标准的资源机制。

何谓 JSF 技术呢？简单来说，JSF 就是“模式-视图-控制器”模型 (MVC) 加上用户接口组件 (UI Component)，在这些用户接口组件上绑定转换器、事件监听器和验证器。引进了导航规则，让你很容易从一个网页转移到另一个网页。JSF 技术更清楚地分离了表现层和商务逻辑层。说得抽象些，JSF 技术就是将 Java 网络应用在服务端的组件框架。概括起来，JSF 有三大优势：第一，JSF 是基于 UI 组件的框架；第二，JSF 是事件驱动模型；第三，JSF 提供了用户接口到业务逻辑的直接映射；总之，JSF 是新一代 Java 网络表现层。

本书采用 NetBean 7.0 整合开发环境工具，NetBean 容易编辑、布局和运行 Web 应用中各种文件。它既可以使用 Glassfish 3.1 服务器，也可以使用 Tomcat 7.0 服务器；既可以使用 Derby 数据库，也可以使用 MySQL 数据库。本书注重于基础知识的讲解，因为掌握了基础知识并培养了兴趣，读者就可以通过网络继续深造和创新。

本书遵循的原则

详而不繁。所谓详解，本书既告诉你怎样做 (how to)，又告诉你为什么要这样做 (why)，力求做到详而不繁。内容由浅入深，循序渐进。除了综合例题外，一般例子尽可能做到简单明了。Java Web 初学者完全可按书上的介绍，逐步实践 (hand on) 来掌握 JSF 的最新技术。对 Java Web 有一定基础的读者，也不会感到乏味，本书介绍了 JSF 2.0 的所有新特点，交代了问题的来龙去脉，站在制高点来观察 JSF，这有益于 Java 网络编程的开发者。对 JSF 请求处理生命周期这样疑难内容，不是像其他书那样，一开始就全盘托出，而是采用逐步引入，最后给读者一个清晰的概念。

少而精。本书以少而精为宗旨。Internet 自上世纪末发明以来，取得了快速发展和广泛的应用，几乎渗透到生活的各个方面。各个软件公司像雨后春笋般地涌现，新的技术日新月异、层出不穷。初学网络编程的读者如何在浩瀚的、五花八门的 IT 书籍中，找一本适合自己需求的启蒙书，当今，这是一件困难的事。本书力求内容简明扼要，条理清晰精练，迎合初学者的需求。

内容新颖。Java 语言诞生至今仅十多年，经历了不断发展和完善，JSF 技术从 2004 年发布至今，仅有 6 年时间，本书介绍的 JSF 2.0 是第三版 (2009 年 12 月)。本书例题所用 Java 版本是 JDK 1.7.0；J2EE 6.0，所有例题在 NetBeans 7.0 整合开发环境和 Glassfish 3.1 服务器运行。对太陈旧的内容尽量不讲，让读者集中精力和时间熟悉最新颖的内容，尽快赶上 IT 飞速发展的步伐，以适应 IT 实践的需要。

面向实战。本书列出了大量例题，在讲解中结合了大量实用而又有代表性的实例和典型应用，内容丰富、系统性强。在每章例题目录中，设置 README 文件，指出解题关键。第 15 章虚拟大学，是一个典型的综合性 JSF 网络编程例子。它使用了 JSF 的最新技术、

Facelets 和模板, 三种托管 Beans, 复合组件, 自定义转换器和验证器, Ajax, JPA 和数据库链接池, 认证和安全连接 (SSL), 发送确认电邮等技术。读者可参照该模式, 创建自己的电子商务项目, 这样可以节省大量时间和减少编码错误。

各章大纲 (两篇 15 章):

第一篇基础篇, 这篇共包含 10 章。

第 1 章 JSF 引论。介绍了什么是 JSF, 如何安装必备的软件, 如 Jdk 1.7.0、NetBeans 7.0.1, 包括 Glassfish 3.1.1 服务器等, 如何使用 NetBeans 创建一个简单的 JSF 应用, 同时简单讨论了 JSF 生命周期。

第 2 章托管 Bean 和表达式语言。本章引进了一个重要的概念“逆向控制”(IoC), 又叫“依赖注射”、托管 Bean 和上下文和依赖注射 (CDI)。Bean 就是依据于这概念。本章介绍 JavaBean 特性, JSF 2.0 增加的方法, 甚至这方法可以带有参数; 对比了三种不同的托管 Beans。JSF 2.0 部分增加了对话有效域、视图有效域和自定义有效域。本章采用的信息绑定, 可让你的 JSF 应用很容易实现国际化。

第 3 章 JSF 标准标签库。JSF 2.0 增加到六个标准库, 它们是 core, html, facelets, composite component, JSTL core, JSTL function。有两个借用 JSTL, 有一个借用 HTML, 其余三个是 JSF 独有的, 总共有 104 个标准标签。内容极其丰富, 其中 Facelets 库标签 (UI) 放在第 4 章讲述, 数据表格 (h:dataTable) 放在第 5 章讲述, 复合组件库标签 (cc) 放在第 10 章讲述。

第 4 章 Facelets 与模板。Facelets 是 JSF 专家组成员 Jacob Hookom 发明的, 在 JSF 2.0 版中, 它已结合进核心 JSF 指南中, 它被倡导为基于 JSF 技术的网络应用的表现技术。Facelets 用 XHTML 样式来创建 JSF 视图网页的声明语言, 它具有性能强大、简单易行的特点。在 JSF 2.0 以前的版本里, JSF 技术借用 JSP 作为它的视图声明语言。Facelets 让你把网页中的共同性封装在模板内。

第 5 章数据表格。JSF 使用 <h:dataTable> 标签来适应这种要求。它是标准组件中最复杂的组件, value 属性代表被迭代的数据, 它可以列举在数组、列表、结果集等六种类型中的每个数据, 数据名由 var 属性指定。<h:dataTable> 除了 value 和 var 属性外, 还有很多其他属性、功能齐全。<h:dataTable> 在制作 HTML 表格时配合 UIData 组件, 两者结合就提供了极其丰富的制表功能, 包括使用 CSS 样式表、访问数据库数据、自定义表格模式。

第 6 章导航模型。JSF 2.0 引进了隐式导航, 单击按钮或链接可以简单地返回 outcome, 直接指向下一个网页。显式导航需要 faces-config.xml 文件配置导航规则。讨论了静态导航和动态导航。由于 HTTP 协议, JSF 2.0 可以使用 flash 实现 PRG, 或用视参数实现 PRG, 用视参数的 PRG 可以书签化 URL。本章最后讨论了通配符导航和条件导航。

第 7 章用户接口组件模型。本章首先讲述了为什么要创建基于组件的网络开发, 详细讨论了 JSF 的用户接口组件架构, 介绍了 JSF 的基本 UI 组件, UI 组件树与客户在浏览器中所见建立一一对应关系, 介绍了 UI 组件、渲染器和标签处理器三者的关系, 最后讨论如何在 Facelets 使用 UI 组件。

第 8 章转换与验证数据。JSF 提供了一套标准转换器和验证器, 同时你自己也可自定义转换器和验证器。本章讨论了如何用标签或程序把转换器或验证器与组件联系起来。JSF

也提供了一套标准转换错误信息和标准验证错误信息，也可以修改这些标准信息。专题讨论了 Bean 验证。

第 9 章 JSF 事件模型。本章介绍了 JSF 的四种事件：动作事件，值变事件，阶段事件和系统事件。JSF 事件模型与请求处理生命周期有密切关系，你可以用阶段事件来跟踪生命周期的各个阶段。系统事件是 JSF 2.0 的新特点，它对应用提供了比阶段事件更深层的观察。

第 10 章复合组件。JSF 2.0 提供了一个程序包，能从现有组件合成一个新的组件，这个程序包就是标准标签库 `composite component(cc)`。复合组件是一种特殊的 Facelets 页面，它有接口区和实现区，放在资源库里；与一般组件一样使用它，必须在 `<html>` 元素下声明适当的命名空间；JSF 会自动创建一个顶级组件，当需要给复合组件加行为时，很容易用它提供自定义顶级组件类。

第二篇高级篇，共包含五章。

第 11 章 JSF 与 Ajax。JSF 2.0 提供了坚实的 Ajax 架构，可用它来实现丰富的用户接口，JSF 提供了抽象的 `f:ajax` 标签，可把 Ajax 行为附着在组件上，就像转换器和验证器一样，通过嵌入的 `f:validator` 和 `f:converter` 标签分别把验证和转换行为附着在组件上一样。也可以直接使用 Ajax 的 API，当然，这比直接使用 `f:ajax` 标签要复杂一些，但它提供更多灵活性，你可以把更多功能加到 Ajax 调用上。

第 12 章自定义 UI 组件。JSF 为创建基于 HTML 网络应用提供了一套基本的标签，诸如文本框、选择框和按钮等。第 10 章介绍了怎样从现有的组件合成一个复合组件，但它仅适合比较简单的布局，无法用复合组件来显示树状或表格状日历。幸运的是，JSF 提供了自定义 UI 组件的机制，满足各种各样的实际需求，实现自定义组件和相关的标签，其特点如同 JSF 标准标签一样。

第 13 章 JSF 与持久架构。利用 JDBC 的 API，可用 Java 语法去访问各种各样 SQL 数据库，重点介绍 Derby 和 MySQL 数据库引擎。Java 持久架构 (JPA) 是标准的对象关系映射 (ORM)，对象关系映射受到广大软件工程师的青睐，使用实体与数据库打交道既方便又准确。使用数据库连接池，通过 Web 应用中的所有用户共享对数据库的连接，数据库连接池大大改进了 Web 应用的表现力。

第 14 章 JSF 应用的安全问题。网络安全包括两个方面的内容：(1) 防止无授权用户访问敏感资料；这个处理涉及两个方面：a) 访问限制，即哪些资料需要受到保护，谁可以访问这些资料；b) 认证，即识别用户是否获得授权。(2) 资料传输过程中，防止黑客偷走资料：这个处理涉及使用安全套接层 (SSL)，给浏览器与服务器之间传输的资料加密。

第 15 章濮阳虚拟大学。演示了开发实际 JSF 应用所需的技能。它综合应用了本书所讲述的知识，包括模板、`dataTable` 和编辑表格格子内容、复合组件、数据转换和验证、网页导航、Ajax、JPA 和数据库链接池、认证和 SSL、发送确认电邮等技术。

本书读者对象

具有 Java 程序语言基础知识，对没有 Java 网络编程经验的读者，本书是你进入 Java Web 领域的最好导引；而对 Java 网络编程有基础的读者，比如，了解 Servlet 和 JSP 技术，想要更新 Java 网络知识的开发者，想成为一名开发 Java 企业应用的软件工程师，这是一本不可缺少的指导书；对学习计算机专业的大专院校师生，想掌握前沿 Java EE 6 技术，本书

肯定是本很好的参考书，也可作为 Java EE 6 培训和自学的教材。

致谢

本人感谢恩师中国科学院资深院士陶诗言教授。感谢在网络上发表对 JSF 技术学习心得和文章的作者们。部分例题参考第三版 Core JavaServer Faces 和 The Complete Reference JavaServer Faces 2.0。

本书献给夫人陈佩英和女儿盛晔。

联系

对于书中的疏漏和不妥之处，恳望读者批评指正。热烈欢迎读者对本书评论，请发送邮件到 huashengca@gmail.com。例程源始编码请到网站 <http://www.tup.com.cn> 下载。

CONTENTS

目 录

第一篇 基 础 篇

第 1 章	JavaServer Faces 引论	2
1.1	JSF 概述	2
1.1.1	JSF 应用是什么	2
1.1.2	JSF 的优势	3
1.2	JavaServer Faces 发展简史	4
1.3	必备的软件	6
1.3.1	JDK1.7.0	6
1.3.2	NetBeans7.0.1	7
1.4	一个简单的 JSF 应用	9
1.4.1	创建网页	10
1.4.2	Backing Beans	12
1.4.3	映射 Faces Servlet 实例	17
1.5	用 NetBeans IDE 创建、布局和运行 JSF 应用	18
1.6	JSF 应用生命周期	19
1.6.1	编码与解码	19
1.6.2	生命周期中的六个阶段	21
	本章小结	25
第 2 章	托管 Bean 和表达式语言	26
2.1	Bean 的定义	26
2.2	托管 Bean (Managed Bean)	27
2.2.1	举例说明托管 Bean	28
2.2.2	初始化托管 Bean 的特性	35
2.2.3	直接将 Lists 和 Maps 声明为托管 Bean	40
2.2.4	托管 Bean 的依赖	41
2.3	Backing Bean	42
2.4	上下文依赖注射 Bean (CDI)	43
2.5	信息绑定	44
2.5.1	properties 文件	44
2.5.2	国际化信息	46
2.6	Bean 的作用域	49
2.6.1	Bean 的各种作用域	49
2.6.2	Bean 生命周期的注解	52
2.7	表达式语言语法	53
2.7.1	EL 概况	53

2.7.2 及时计算和延迟计算.....	53	2.7.3 值表达式.....	54	2.7.4 方法表达式.....	59
2.8 程序访问托管 Bean.....	61	2.8.1 访问托管 Bean 的特性.....	61	2.8.2 访问托管 Bean 的方法.....	62
本章小结.....	63				
第 3 章 JSF 标准标签库.....	64				
3.1 JSF 核心标签库.....	64				
3.2 JSF HTML 标签.....	65				
3.2.1 HTML 组件的公共属性.....	66				
3.3 输入标签.....	68				
3.3.1 四个输入标签.....	68				
3.3.2 输入标签示例.....	69				
3.4 输出标签.....	72				
3.5 命令标签.....	73				
3.6 选择标签.....	80				
3.7 message 标签.....	92				
本章小结.....	95				
第 4 章 Facelets 与模板.....	96				
4.1 什么是 Facelets.....	96				
4.2 Facelets 标签.....	96				
4.2.1 模板标签.....	97				
4.2.2 非模板标签.....	98				
4.3 开发一个简单的 Facelets 应用.....	100				
4.4 一个模板 Facelets 应用.....	108				
4.5 Resources.....	114				
本章小结.....	114				
第 5 章 数据表格.....	115				
5.1 数据表格标签 (h:dataTable).....	115				
5.1.1 创建表格.....	115				
5.1.2 一个简单的数据表格.....	116				
5.1.3 h:dataTable 与 h:column 的属性.....	120				
5.1.4 表格标题, 列栏目名和列 注解.....	120				
5.2 样式表在表格中的使用.....	121				
5.3 编辑表格.....	124				
5.3.1 编辑单元格里的内容.....	124				
5.3.2 删除表格里的行.....	130				
5.4 数据库表格.....	133				
5.4.1 NetBeans IDE 创建数据库.....	133				
5.4.2 将数据库 mydb 连接到 连接池.....	135				
5.4.3 数据库表格.....	136				
5.5 数据表格模式.....	139				
5.5.1 显示数据表格行数.....	139				
5.5.2 寻找所选的行.....	143				
5.5.3 对数据进行分类检索.....	146				
本章小结.....	151				
第 6 章 导航模式.....	152				
6.1 JSF 导航系统概述.....	152				
6.1.1 对 MVC 模型的回顾.....	152				
6.1.2 导航处理器 (NavigationHandler).....	153				
6.1.3 Faces 动作方法.....	154				
6.2 隐式导航.....	154				
6.3 显式导航规则.....	158				
6.4 重定向.....	159				
6.4.1 问题的起因.....	159				
6.4.2 重定向的使用.....	160				
6.4.3 重定向与隐式导航.....	160				
6.5 PRG 模型与 JSF.....	164				
6.5.1 使用 flash 的 PRG.....	164				
6.5.2 使用视图参数的 PRG.....	170				
6.5.3 显式导航规则中配置 视图参数.....	174				
6.5.4 可书签化和视图参数.....	174				
6.6 通配符导航和条件导航.....	175				
6.6.1 通配符导航.....	175				
6.6.2 条件导航.....	176				
本章小结.....	179				

第 7 章 用户接口组件模型	180
7.1 什么是用户接口组件	180
7.2 JSF UI 组件引论	182
7.3 UI 组件树 (View)	184
7.4 UI 组件和其他辅助组件	189
7.5 UI 组件与 Facelets	190
本章小结	193

第 8 章 转换与验证数据	194
8.1 数据转换和数据验证概述	194
8.2 标准数据转换器	196
8.3 转换器与 UIComponent 实例的 联系	202
8.4 自定义转换器	205
8.4.1 如何自定义转换器	205
8.4.2 声明和配置自定义转换器	208
8.4.3 结果	209
8.5 标准数据验证器	210
8.6 验证器与 UIComponent 实例的 联系	213
8.6.1 用标签来联系验证器与 UI 组件实例	213
8.6.2 使用程序联系验证器与 组件实例	214
8.7 自定义验证器	216
8.7.1 实现 Validator 接口	216
8.7.2 在托管 Bean 中实现 validation()方法	219
8.8 JSF 中使用 Bean 验证	220
8.8.1 JSF 验证与 Bean 验证的 差别	220
8.8.2 在 JSF 框架内使用 Bean 验证的细节	222
8.8.3 信息与 Bean 验证	225
8.9 视图中的信息	227
8.9.1 FacesContext 与 FacesMessage	227
8.9.2 怎样渲染 FacesMessage	228
本章小结	231

第 9 章 事件模型	233
9.1 事件模型概况	233
9.1.1 事件分类	234
9.1.2 事件监听器接口和事件类	235
9.2 JSF 事件与请求处理生命周期	236
9.2.1 跟踪生命周期各个阶段	236
9.2.2 何时处理事件	238
9.3 动作事件	239
9.4 值变事件	244
9.5 阶段事件	247
9.6 系统事件	251
9.6.1 系统事件监听器方法	251
9.6.2 系统事件监听器类	257
9.6.3 发布与预订	262
本章小结	263

第 10 章 复合组件	264
10.1 复合标签库	264
10.2 logPanel 例子	266
10.2.1 定义网页	268
10.2.2 使用网页	268
10.2.3 资源库	270
10.2.4 复合组件实例	270
10.2.5 #{cc}隐式对象	271
10.3 属性类型和配置复合组件	271
10.3.1 属性类型	271
10.3.2 配置复合组件	272
10.4 操作服务端数据	274
10.5 复合组件上添加验证器和监听器	277
10.5.1 复合组件上添加验证器	277
10.5.2 复合组件绑定动作监听器	279
10.6 复合组件标签内插入其他标签	281
10.6.1 资源绑定	281
10.6.2 facets	282
10.6.3 插入子标签	282
10.7 复合组件加上 JavaScript	283
本章小结	284

第二篇 高级篇

第 11 章 JSF 与 Ajax	286	第 13 章 JSF 与 JDBC	350
11.1 Ajax 引论	286	13.1 JDBC 概述	350
11.2 一个简单的 Ajax 例子	287	13.2 Derby 数据库引擎	354
11.2.1 ajax-login 例子	287	13.3 MySQL 数据库引擎	358
11.2.2 实现 Ajax 的步骤	290	13.3.1 安装和测试 MySQL	359
11.2.3 Firebug 跟踪 Ajax	290	13.3.2 NetBeans 中连接 MySQL	360
11.3 <f:ajax>与它的属性	293	13.4 使用预先准备好的查询语句	363
11.3.1 <f:ajax>包装组件	293	13.5 建立和使用存储过程和函数	366
11.3.2 <f:ajax>的属性	293	13.6 商务合同	369
11.4 Ajax 监听事件	294	13.7 数据库连接池	372
11.5 监测 Ajax 请求进度	297	13.7.1 为什么要创建数据库连接池	372
11.6 处理 Ajax 出错	300	13.7.2 配置数据资源	373
11.7 JSF 2.0 中的 JavaScript 库	301	13.7.3 登录-注册-登录实例	376
11.7.1 标准 JavaScript 资源	301	13.8 Java 持久架构	387
11.7.2 使用 JavaScript 实现 Ajax	302	13.8.1 Java 持久架构引论	387
11.8 复合组件中使用 Ajax	305	13.8.2 怎样使用 JPA	388
本章小结	312	13.8.3 登录-注册-登录实例	389
第 12 章 自定义 UI 组件	313	13.8.4 JPA 使用无状态会话 Bean	394
12.1 UI 组件的构件	313	13.8.5 JPA 使用有状态会话 Bean	397
12.1.1 什么是 UI 组件	313	本章小结	403
12.1.2 UI 组件的构件	314	第 14 章 JSF 中的网络安全	405
12.2 创建自定义 UI 组件	315	14.1 网络安全引论	405
12.2.1 自定义组件实例 (MyHello)	315	14.1.1 安全概述	405
12.2.2 接受表单输入的自定义 UI 组件	319	14.1.2 几个重要的术语	406
12.2.3 分离的渲染器	322	14.1.3 Tomcat 怎样实现安全领域	407
12.3 处理自定义组件标签属性	326	14.2 表单认证	410
12.3.1 自定义标签处理器	326	14.2.1 表单认证的七个步骤	411
12.3.2 使用映射处理标签属性	327	14.2.2 表单认证实例	415
12.3.3 使用标签处理器处理标签属性	328	14.3 基本认证	422
12.4 自定义组件使用 JavaScript	330	14.3.1 基本认证概述	422
12.5 使用子组件和 Facet	334	14.3.2 基本认证实例	423
12.6 自定义组件的状态管理	345	14.4 证书认证	430
本章小结	349	14.4.1 不对称加密	430
		14.4.2 证书的作用	430
		14.4.3 证书认证	431

14.5 安全连接.....	432	15.2 师生论坛.....	451
14.5.1 SSL 引论.....	432	15.3 师生互动.....	457
14.5.2 Glassfish 服务器配置 SSL.....	434	15.3.1 学生更新“学生笔记”.....	458
14.5.3 为 Tomcat 服务器配置 SSL.....	435	15.3.2 导师发布告.....	462
14.6 应用管理安全.....	436	15.4 购物车应用.....	467
14.6.1 应用管理安全概述.....	436	15.4.1 JPA 和实体类.....	468
14.6.2 检查请求者身份.....	438	15.4.2 后台托管 Bean.....	470
14.6.3 全部用编程处理安全.....	441	15.4.3 购物车应用的用户接口.....	475
本章小结.....	445	15.4.4 购物车应用结构.....	480
第 15 章 综合实例——溧阳虚拟		15.5 发送确认电子邮件.....	494
大学	446	15.5.1 JavaMail 引论.....	494
15.1 溧阳虚拟大学概况.....	446	15.5.2 发送电子邮件步骤.....	495
15.1.1 模板.....	446	15.5.3 发送确认电子邮件.....	496
15.1.2 登录-注册-登录.....	449	本章小结.....	498
15.1.3 免费视听.....	449	附录 缩写字索引	499
		参考文献	502

第一篇

基础篇

第 1 章

JavaServer Faces 引论

JavaServer Faces (JSF) 是基于组件的网络开发框架, 是创建网络应用的用户接口的标准。组件上承载着验证器、数据转换器、事件监听器等, 同时也定义了隐式和显式导航规则。本章首先解释了 JSF 技术, 追溯了它的发展简史; JSF 技术的最大优点之一是把网络应用的行为与表现清楚地分离开, 将网络应用开发者划分为三个角色: 网页设计人员, 应用程序设计人员和 UI 组件开发者; 然后介绍了如何安装 Java 标准版 (JDK1.7.0)、整合开发工具 NetBeans 7.0.1 (它包含了服务器 Glassfish 3.1.1 和 Tomcat 7.0.14), 如何使用 NetBeans 整合工具创建、布局和运行 JSF 应用; 最后简述了 JSF 应用的生命周期。

1.1 JSF 概述

JavaServer Faces 技术是 Java 网络应用服务端的组件框架。说得具体些, 在 MVC 模型上添加了用户接口组件 (UIComponent), 在这些用户接口组件上承载着验证器、数据转换器、事件监听器, 定义了导航规则。如果你掌握了 UIComponent, 就掌握了九成的 JSF 技术。UIComponent 由表现组件和管理组件状态的 API 组成, 它们能处理事件, 转换和验证数据, 定义页面的导航, 支持国际化信息。把组件加入网页, 可联系组件与服务端对象的标签库。

JSP 技术提供了一个定义良好的程序模式和各种标签库, 这些特点使得有服务端用户接口的网络应用更容易创建和维护, JSF 技术有如下一些功能:

- (1) 创建网页;
- (2) 把组件可以拖拽到网页上, 即加组件标签;
- (3) 把网页的组件与服务端的数据绑定在一起;
- (4) 把产生组件的事件与服务端的应用编码联系起来;
- (5) 在服务端请求生命史外, 还存储和恢复应用状态;
- (6) 通过自定义来重用和扩展组件。

1.1.1 JSF 应用是什么

典型 JavaServer Faces 应用包括如下几部分。

- (1) 一套布局着组件的网页。
- (2) 一套可将组件加进网页的标签。

(3) 一套后台 Beans (Backing Beans), 它们是 JavaBean 成员, 包含着特性和网页上组件的功能。

(4) 网络布局描述文件 (web.xml)。

(5) 一个或多个应用资源配置文件, 如 faces-config.xml, 用它来定义网页导航规则, 配置 Bean 和其他自定义对象, 如自定义组件, 此文件为可选项。

(6) 一套自定义对象, 包括自定义组件、验证器、转换器或监听器, 这些由应用开发者创建, 此对象也是可选项。

(7) 一套表现网页上自定义对象的自定义标签。

图 1-1 描述了在经典 JSF 应用中客户与服务器之间的互动, 对于一个客户请求的响应, 网页由网页容器来渲染, 网页容器也实现了 JSF 技术。

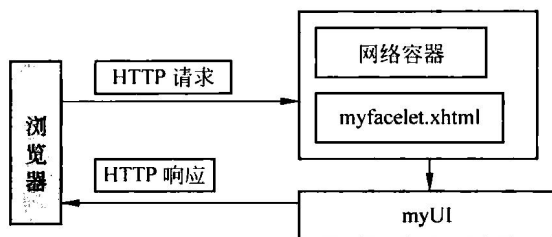


图 1-1 JSF 应用对客户请求的响应

网页 myfacelet.xhtml 是用 JSF 组件标签建成的。组件标签用来把组件加到视图上, 它是网页在服务端的表现。除了组件外, 网页也能引用对象, 比如:

- ☐ 任何注册在组件上的事件监听器、验证器、转换器。
- ☐ JavaBean 组件可捕获数据和处理应用指定的组件功能。

对于来自客户的请求, 视图作为响应而被渲染, 这种渲染是对服务端视图的处理。网页容器产生输出, 如 HTML 或 XHTML, 它们由浏览器阅读。

1.1.2 JSF 的优势

JSF 技术的最大优点之一是把网络应用的行为与表现清楚地分离开。一个 JSF 应用能把 HTTP 请求映射到指定组件的事件处理上, 管理组件如同服务器上有状态的对象那样, 这样, JSF 技术允许你建立网络应用实现更精细的行为与表现分离, 这种表现传统上是由客户端 UI 架构提供的。

JSF 技术清楚地将网络应用开发者划分为三个角色: 网页设计人员、应用程序设计人员和 UI 组件开发者。从使用角度来看, 网页设计人员和应用程序设计人员可以各自开发程序, 而不侵入彼此的工作范围; 而 UI 组件开发人员可以独立开发个别组件。也就是说, 即使没有很好的程序背景, 网页设计者也可以在自己网页上使用 JSF 标签, 把这些标签与服务端的对象连接起来, 而又不需要写任何脚本程序。

JSF 技术的另一个重要的目的是利用熟悉的组件和网络层概念, 而不是局限于特定的脚本技术或标记语言。JSF 技术的 API 直接位于 Servlet API 之上 (如图 1-2 所示)。这种分

层有很多好处，如可以使用不同表现技术，直接从组件类创建自定义组件，在不同客户设备上产生输出等。

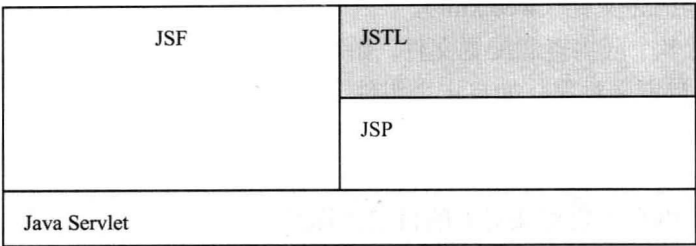


图 1-2 Java 网络应用技术

JSF 2.0 引进了 Facelets 技术，现在它已成为 JSF 网络应用的流行表现技术，Facelets 有如下的优点：

- ❑ Facelets 技术提供了代码重用，通过组件模板和复合组件的特点来扩充组件。
- ❑ 使用 JSF 注解（annotation）时，能自动把后台 Beans 注册为 JSF 应用的资源。此外，隐式导航规则允许开发者很快地配置网页导航，这些特点减少了应用的手工配置过程。

最重要的是，JSF 技术在管理组件状态、处理组件数据、验证用户输入等方面提供了一个丰富的架构。

1.2 JavaServer Faces 发展简史

像绝大多数其他重要的 IT 技术一样，JavaServer Faces 技术的发展，也经历了一个不断完善、不断更新的过程，新的技术不断淘汰旧的技术（如图 1-3 所示）。JSF 发展的宗旨是，要更简单、更有效地创建动态用户接口，而这些接口是基于设计良好和容易维护的架构之上的。

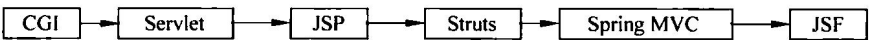


图 1-3 JSF 发展史

1. 公共网关接口（CGI）

早在上世纪 90 年代中期，那时刚开发网络应用，装配网络应用的主要技术就是 CGI，用 CGI 来产生动态的网络内容。CGI 是由 Rob 和 Mike McCool 引进的，他们是美国国家超级计算应用中心（NCSA）负责开发 HTTP 服务器的专家组成员，同时，NCSA 也创建了世界上第一代浏览器——Mosaic。

CGI 技术是调用服务器进程来产生网页的动态输出，比如股票查询、报告被浏览的次数等。而产生动态输出的程序通常是操作系统的 Shell 脚本程序，一个原始的编译程序，或者是一个解释了的程序，比如 Perl，对每次来自 HTML 页面的请求，产生一个新的进程，