

Oracle Java EE 7文档团队成员精准解说Java EE 7平台，为深入理解Java EE 7企业应用开发提供翔实指导

更简洁明确的概念介绍，更具体完整的应用示例，全面解读Java EE 7平台特性，手把手教你运行和测试示例，帮助你真正掌握Java EE 技术，成为Java EE企业应用开发的高手

PEARSON



Java EE 7

权威指南 卷1

(原书第5版)

The Java EE 7 Tutorial: Volume 1
(Fifth Edition)

埃里克·珍兆科 (Eric Jendrock)

里卡多·塞维拉-纳瓦罗 (Ricardo Cervera-Navarro)

[美] 伊恩·埃文斯 (Ian Evans)

著

金姆·哈泽 (Kim Haase)

威廉·马基特 (William Markito)

苏金国 江健 等译



机械工业出版社
China Machine Press



Java EE 7

权威指南 卷1

(原书第5版)

The Java EE 7 Tutorial: Volume 1
(Fifth Edition)

埃里克·珍兆科 (Eric Jendrock)

里卡多·塞维拉-纳瓦罗 (Ricardo Cervera-Navarro)

[美] 伊恩·埃文斯 (Ian Evans)

著

金姆·哈泽 (Kim Haase)

威廉·马基特 (William Markito)

苏金国 江健 等译



机械工业出版社
China Machine Press

图书在版编目 (CIP) 数据

Java EE 7 权威指南: 卷 1 (原书第 5 版)/(美) 珍兆科 (Jendrock, E.) 等著; 苏金国等译.
—北京: 机械工业出版社, 2015.5

(Java 核心技术系列)

书名原文: The Java EE 7 Tutorial: Volume 1, Fifth Edition

ISBN 978-7-111-49760-8

I. J… II. ①珍… ②苏… III. JAVA 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2015) 第 059224 号

本书版权登记号: 图字: 01-2014-4443

Authorized translation from the English language edition, entitled *The Java EE 7 Tutorial: Volume 1, Fifth Edition*, 9780321994929 by Eric Jendrock, Ricardo Cervera-Navarro, Ian Evans, Kim Haase and William Markito, published by Pearson Education, Inc., Copyright © 2014, Oracle and/or its affiliates.

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 China Machine Press Copyright © 2015.

本书中文简体字版由 Pearson Education (培生教育出版集团) 授权机械工业出版社在中华人民共和国境内 (不包括中国台湾地区和香港、澳门特别行政区) 独家出版发行。未经出版者书面许可, 不得以任何方式抄袭、复制或节录本书中的任何部分。

本书封底贴有 Pearson Education (培生教育出版集团) 激光防伪标签, 无标签者不得销售。

Java EE 7 权威指南: 卷 1 (原书第 5 版)

出版发行: 机械工业出版社 (北京市西城区百万庄大街 22 号 邮政编码: 100037)

责任编辑: 李 艺

责任校对: 董纪丽

印 刷: 北京市荣盛彩色印刷有限公司

版 次: 2015 年 5 月第 1 版第 1 次印刷

开 本: 186mm×240mm 1/16

印 张: 30.75

书 号: ISBN 978-7-111-49760-8

定 价: 99.00 元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

客服热线: (010) 88378991 88361066

投稿热线: (010) 88379604

购书热线: (010) 68326294 88379649 68995259

读者信箱: hzjsj@hzbook.com

版权所有·侵权必究

封底无防伪标均为盗版

本书法律顾问: 北京大成律师事务所 韩光/邹晓东

The Translator's Words 译者序

Java 平台素以提倡开放为宗旨，基于这种开放性，在全行业共同推动下，再加上 Java 社区进程和 GlassFish 社区的大力支持，Oracle 公司终于在 2013 年推出了 Java EE 7，而这距离 2008 年发布 Java EE 6 已经过去了 5 年之久。在这 5 年里，企业应用有了哪些发展？Java EE 又有了哪些变化？新增了哪些特性？对 Java 企业应用提供了哪些新的支持？在哪些领域有了新的探索？本书^①可以很好地回答我们的这些疑问。

实际上，伴随着 Java EE 的发展脚步，从 2001 年《The J2EE Tutorial》第 1 版问世，到现在本书与大家见面，也经过了十余年的演进历程，其间 2004 年改版为《J2EE™ Tutorial》第 2 版，2006 年出版了《The Java EE 5 Tutorial》第 3 版，从《The Java EE 6 Tutorial》第 4 版开始分为两册，分别介绍基本概念和高级主题，不过基本概念分册于 2010 年出版，而高级主题分册到 2013 年才姗姗到来，好在本书没有让我们久等，2014 年第 1 卷和第 2 卷同时出版。现在在你手上的这本就是其中的第 1 卷。

在第 1 卷中详细介绍了 Java EE 的基本概念以及 JSF、Java servlet、bean 验证、Java EE 上下文和依赖注入、Web 服务等内容。在这本书中，你不仅能看到简洁明确的概念介绍，还会看到具体完整的实用示例。此外，本书还会手把手地教你如何运行和测试这些示例，使你能真正掌握 Java EE 技术，成为 Java EE 企业应用开发的高手。

本书由苏金国、江健等主译，程芳、吴忠望、张练达、陈峰、杨健康、姚勇、张莹、牛亚峰、高强、赵蓓、荆涛、刘卫宏参与了全书的修改整理，朱涛江、杜志秀、杨春生、刘亮、王宇、张伶、王志淋、刘跃邦完善了关键部分的翻译。由于水平有限，译文肯定有不当之处，敬请批评指正。

译者

2014 年 12 月

① 实际上本书英文版和中文版都分两卷出版，你手上的这一本是卷 1。——编辑注

前 言 *Preface*

本书针对的是 Java 平台企业版 7 (Java Platform, Enterprise Edition 7), 即 Java EE 7, 是开发企业应用的实用指南。全书将使用 GlassFish 服务器开源版 (GlassFish Server Open Source Edition) 来介绍。

GlassFish 服务器开源版是目前业界领先的开源项目和开放社区平台, 可用来构建和部署下一代应用和服务。GlassFish 服务器开源版由 GlassFish 项目开源社区研发 (<https://glassfish.java.net/>), 是 Java EE 7 平台规范的第一个兼容实现。这个应用服务器具有轻量级、灵活和开源等特点, 使得企业不仅可以充分利用 Java EE 7 规范引入的诸多新功能, 还能通过更快捷、更优化的开发和部署流程扩展其已有功能。本书后面将把 GlassFish 服务器开源版简称为 GlassFish 服务器。

本书的读者对象

本书面向有兴趣开发和部署 Java EE 7 应用的程序员, 讨论构成 Java EE 平台的有关技术, 另外还会介绍如何开发 Java EE 组件并部署到 Java EE 软件开发包 (Software Development Kit, SDK)。

阅读须知

学习本书之前, 应当对 Java 编程语言有充分的了解。为此, 建议大家仔细通读《The Java Tutorials》(<http://docs.oracle.com/javase/tutorial/>)。

相关文档

第 2 卷会介绍本书 (《The Java EE 7 Tutorial》第 1 卷) 未涵盖的 Java EE 7 技术和主题, 包括企业 JavaBeans (Enterprise JavaBeans, EJB)、Java 持久化、Java 消息服务、安全性以及

其他一些技术。第 2 卷的最后还提供了 3 个案例研究，其中综合使用了多项 Java EE 7 技术。

GlassFish 服务器文档集介绍了部署计划和系统安装的有关内容。GlassFish 服务器开源版的相关文档可以从 <https://glassfish.java.net/docs/> 得到。

可以在 <http://docs.oracle.com/javaee/7/api/> 查阅 Java EE 7 API 规范，另外 Java EE 7 SDK 也提供了这个规范。

此外，Java EE 规范也很有用，参见 <http://www.oracle.com/technetwork/java/javaee/tech/>。

要在 NetBeans 集成开发环境（Integrated Development Environment，IDE）创建企业应用，有关信息参见 <https://netbeans.org/kb/>。

要了解 GlassFish 服务器使用的 Java DB 数据库，有关信息请访问 <http://www.oracle.com/technetwork/java/javadb/overview/>。

GlassFish Samples 项目提供了一组示例应用，展示了诸多 Java EE 技术。GlassFish Samples 与 Java EE 软件开发包（SDK）捆绑发行，也可以从 GlassFish Samples 项目网页（<https://glassfish-samples.java.net/>）获得。

Oracle 无障碍计划

关于 Oracle 无障碍承诺的有关信息请访问 Oracle 无障碍计划网站（<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>）。

默认路径和文件名

下表描述了本书中使用的默认路径和文件名。

占位变量	描 述	默认值
<i>as-install</i>	表示 GlassFish 服务器或 SDK（包含 GlassFish 服务器）的安装主目录	Solaris 操作系统、Linux 操作系统和 Mac 操作系统上的安装目录为： <i>user's-home-directory</i> /glassfish4/glassfish Windows 上的安装目录为： <i>SystemDrive</i> :\glassfish4\glassfish
<i>as-install-parent</i>	表示 GlassFish 服务器安装主目录的父目录	Solaris 操作系统、Linux 操作系统和 Mac 操作系统上的父目录为： <i>user's-home-directory</i> /glassfish4 Windows 上的父目录为： <i>SystemDrive</i> :\glassfish4
<i>tut-install</i>	表示安装 GlassFish 服务器或 SDK 并运行更新工具（Update Tool）之后 Java EE Tutorial 的安装主目录	<i>as-install-parent</i> /docs/javaee-tutorial
<i>domain-dir</i>	表示存储一个域的配置文件的目录	<i>as-install</i> /domains/domain1

致谢

首先要感谢 Java EE 规范的各位领导者: Linda DeMichiel、Bill Shannon、Emmanuel Bernard、Ed Burns、Shing Wai Chan、Kin-Man Chung、Danny Coward、Nigel Deakin、Rod Johnson、Roger Kitain、Jitendra Kotamraju、Anthony Lai、Bob Lee、Ron Monzillo、Rajiv Mordani、Pete Muir、Paul Parkinson、Santiago Pericas-Geertsen、Marek Potociar、Sivakumar Thyagarajan、Marina Vatkina 和 Chris Vignola。

另外还要感谢 Java EE 7 SDK 团队, 特别是 Snjezana Sevo-Zenzerovic、Adam Leftik、Michael Chen 和 John Clingan。

Manfred Riem 和 JavaServer Faces 规范领导者提出的建议对于有关 JSF 技术的各章有很大帮助。

要特别感谢我们的经理 Alan Sommerer, 感谢他一直以来的支持和影响。

还要感谢 Jordan Douglas 和 Dawn Tyler 制作并更新了本书的插图。感谢 Edna Elle 在工具方面提供了极大的帮助。感谢 Sheila Cepero 在很多方面帮助我们扫清障碍, 使出版能够顺利进行。

最后, 我们要向 Greg Doench、Elizabeth Ryan、Caroline Senay 和 Addison-Wesley 的制作团队致以最诚挚的谢意, 感谢大家审阅我们的初稿, 让本书得以顺利出版。

译者序
前 言

第一部分 引言

第 1 章 概述 2

- 1.1 Java EE 7 平台新增特性 3
- 1.2 Java EE 应用模型 3
- 1.3 分布式多层应用 4
 - 1.3.1 安全 4
 - 1.3.2 Java EE 组件 5
 - 1.3.3 Java EE 客户端 6
 - 1.3.4 Web 组件 7
 - 1.3.5 业务组件 8
 - 1.3.6 企业信息系统层 8
- 1.4 Java EE 容器 9
 - 1.4.1 容器服务 9
 - 1.4.2 容器类型 10
- 1.5 Web 服务支持 11
 - 1.5.1 XML 12
 - 1.5.2 SOAP 传输协议 12

- 1.5.3 WSDL 标准格式 12
- 1.6 Java EE 应用装配与部署 12
- 1.7 开发角色 13
 - 1.7.1 Java EE 产品提供商 13
 - 1.7.2 工具提供商 13
 - 1.7.3 应用组件提供商 13
 - 1.7.4 应用装配人员 14
 - 1.7.5 应用部署人员和管理员 14
- 1.8 Java EE 7 API 15
 - 1.8.1 EJB 技术 15
 - 1.8.2 Java Servlet 技术 17
 - 1.8.3 JSF 技术 18
 - 1.8.4 JSP 技术 19
 - 1.8.5 JSP 标准标记库 19
 - 1.8.6 Java 持久化 API 19
 - 1.8.7 Java 事务 API 20
 - 1.8.8 RESTful Web 服务 Java API 20
 - 1.8.9 托管 bean 20
 - 1.8.10 Java EE 的上下文和依赖注入 20
 - 1.8.11 Java 的依赖注入 21
 - 1.8.12 bean 验证 21
 - 1.8.13 Java 消息服务 API 21

第4章 注入 41

- 4.1 资源注入 41
- 4.2 依赖注入 42
- 4.3 资源注入和依赖注入的主要区别 43

第5章 打包 44

- 5.1 打包应用 44
- 5.2 打包企业 bean 46
 - 5.2.1 企业 bean 打包到 EJB JAR 模块中 46
 - 5.2.2 企业 bean 打包到 WAR 模块中 46
- 5.3 打包 Web 归档 47
- 5.4 打包资源适配器归档 48

第三部分 Web 层

第6章 Web 应用入门 50

- 6.1 Web 应用 50
- 6.2 Web 应用生命周期 51
- 6.3 使用 JSF 技术的 Web 模块：
hello1 示例 52
 - 6.3.1 使用 NetBeans IDE 查看
hello1 Web 模块 52
 - 6.3.2 打包和部署 hello1 Web 模块 56
 - 6.3.3 查看已部署的 Web 模块 56
 - 6.3.4 运行已部署的 hello1 Web 模块 57
 - 6.3.5 取消部署 hello1 Web 模块 57
- 6.4 使用 Java Servlet 技术的 Web
模块：hello2 示例 58
 - 6.4.1 URL 映射到 Web 组件 58

- 6.4.2 检查 hello2 Web 模块 59

- 6.4.3 运行 hello2 示例 60

6.5 配置 Web 应用 61

- 6.5.1 设置上下文参数 61
- 6.5.2 声明欢迎文件 62
- 6.5.3 将错误映射到错误屏幕 63
- 6.5.4 声明资源引用 64

6.6 关于 Web 应用的更多信息 65

第7章 JSF 技术 66

- 7.1 什么是 JSF 应用 67
- 7.2 JSF 技术优势 68
- 7.3 一个简单的 JSF 应用 68
- 7.4 用户界面组件模型 70
 - 7.4.1 用户界面组件类 70
 - 7.4.2 组件呈现模型 72
 - 7.4.3 转换模型 72
 - 7.4.4 事件和监听器模型 73
 - 7.4.5 验证模型 74
- 7.5 导航模型 74
- 7.6 JSF 应用的生命周期 77
 - 7.6.1 JSF 生命周期概述 77
 - 7.6.2 恢复视图阶段 79
 - 7.6.3 应用请求值阶段 80
 - 7.6.4 处理验证阶段 80
 - 7.6.5 更新模型值阶段 80
 - 7.6.6 调用应用阶段 81
 - 7.6.7 呈现响应阶段 81
- 7.7 部分处理和部分呈现 81
- 7.8 关于 JSF 技术的更多信息 82

第 8 章 Facelets 介绍 83

8.1 什么是 Facelets 83

8.2 Facelets 应用的生命周期 85

8.3 开发一个简单的 Facelets 应用:

guessnumber-jsf 示例应用 85

8.3.1 创建 Facelets 应用 86

8.3.2 配置应用 89

8.3.3 运行 guessnumber-jsf Facelets
示例 90

8.4 使用 Facelets 模板 91

8.5 复合组件 93

8.6 Web 资源 95

8.7 可重定位资源 96

8.8 资源库契约 96

8.8.1 hello1-rlc 示例应用 97

8.9 HTML5 友好的标记 99

8.9.1 使用直通元素 100

8.9.2 使用直通属性 101

8.9.3 reservation 示例应用 102

第 9 章 表达式语言 105

9.1 EL 概述 105

9.2 直接和延迟计算语法 106

9.2.1 直接计算 106

9.2.2 延迟计算 107

9.3 值表达式和方法表达式 107

9.3.1 值表达式 107

9.3.2 方法表达式 111

9.3.3 lambda 表达式 112

9.4 集合对象操作 112

9.5 操作符 114

9.6 保留字 115

9.7 EL 表达式示例 115

9.8 关于表达式语言的更多信息 116

**第 10 章 在 Web 页面中使用 JSF
技术** 117

10.1 建立页面 117

10.2 使用 HTML 标记库标记为

页面增加组件 118

10.2.1 常用组件标记属性 120

10.2.2 增加 HTML Head 和 Body
标记 122

10.2.3 增加表单组件 123

10.2.4 使用文本组件 123

10.2.5 使用命令组件标记完成
动作和导航 12710.2.6 使用 h:graphicImage 标记
增加图形图像 12810.2.7 使用 h:panelGrid 和 h:panelGroup
标记放置组件 129

10.2.8 显示选择单个值的组件 131

10.2.9 显示选择多个值的组件 132

10.2.10 使用 f:selectItem 和
f:selectItems 标记 133

10.2.11 显示选择组件的结果 135

10.2.12 使用数据绑定表组件 135

10.2.13 使用 h:message 和 h:messages
标记显示错误消息 13810.2.14 使用 h:button 和 h:link 标记
创建书签式 URL 139

10.2.15	使用视图参数配置书签式 URL	139	12.1.1	创建托管 bean	158
10.2.16	bookmarks 示例应用	140	12.1.2	使用 EL 引用托管 bean	159
10.2.17	使用 h:outputScript 和 h:outputStylesheet 标记 实现资源重定位	141	12.2	编写 bean 属性	160
10.3	使用核心标记	143	12.2.1	编写绑定到组件值的属性	161
第 11 章 使用转换器、监听器和 验证器			12.2.2	编写绑定到组件实例的 属性	166
11.1	使用标准转换器	146	12.2.3	编写绑定到转换器、监听器 或验证器的属性	167
11.1.1	转换组件值	147	12.3	编写托管 bean 方法	168
11.1.2	使用 DateTimeConverter	148	12.3.1	编写处理导航的方法	168
11.1.3	使用 NumberConverter	149	12.3.2	编写处理动作事件的方法	169
11.2	在组件上注册监听器	150	12.3.3	编写完成验证的方法	170
11.2.1	在组件上注册值改变 监听器	151	12.3.4	编写处理值改变事件的 方法	170
11.2.2	在组件上注册动作监听器	151	第 13 章 结合使用 Ajax 和 JSF 技术		
11.3	使用标准验证器	153	13.1	Ajax 概述	173
11.3.1	验证组件值	154	13.2	结合 JSF 技术使用 Ajax 功能	173
11.3.2	使用验证器标记	154	13.3	结合 Facelets 使用 Ajax	174
11.4	引用托管 bean 方法	155	13.3.1	使用 f:ajax 标记	174
11.4.1	引用完成导航的方法	156	13.4	发送 Ajax 请求	175
11.4.2	引用处理动作事件的方法	156	13.4.1	使用 event 属性	175
11.4.3	引用完成验证的方法	156	13.4.2	使用 execute 属性	176
11.4.4	引用处理值改变事件的 方法	157	13.4.3	使用 immediate 属性	176
第 12 章 使用 JSF 技术开发应用			13.4.4	使用 listener 属性	177
12.1	JSF 技术中的托管 bean	158	13.5	在客户端监视事件	177
			13.6	处理错误	177
			13.7	接收 Ajax 响应	178
			13.8	Ajax 请求生命周期	179
			13.9	组件分组	179

13.10 加载 JavaScript 资源	180	15.1.3 组件、呈现器和标记组合	194
13.10.1 在 Facelets 应用中使用 JavaScript API	180	15.2 了解图像地图示例	195
13.10.2 bean 类中使用 @Resource- Dependency 注解	181	15.2.1 为什么使用 JSF 技术实现 图像地图	195
13.11 ajaxguessnumber 示例应用	181	15.2.2 了解呈现的 HTML	195
13.11.1 ajaxguessnumber 源文件	181	15.2.3 了解 Facelets 页面	196
13.11.2 运行 ajaxguessnumber 示例	183	15.2.4 配置模型数据	197
13.12 关于 JSF 技术中 Ajax 的更多 信息	184	15.2.5 图像地图应用类总结	198
第 14 章 复合组件：高级主题与 示例	185	15.3 创建定制组件的步骤	199
14.1 复合组件的属性	185	15.4 创建定制组件类	199
14.2 调用托管 bean	186	15.4.1 指定组件族	202
14.3 验证复合组件值	186	15.4.2 完成编码	202
14.4 compositecomponentexample 示例应用	187	15.4.3 完成解码	204
14.4.1 复合组件文件	187	15.4.4 允许组件属性接受表达式	204
14.4.2 使用页面	188	15.4.5 保存和恢复状态	205
14.4.3 托管 bean	189	15.5 将呈现委托到呈现器	207
14.4.4 运行 compositecomponen- texample 示例	189	15.5.1 创建呈现器类	207
第 15 章 创建定制 UI 组件和其他 定制对象	191	15.5.2 标识呈现器类型	208
15.1 确定是否需要定制组件或 呈现器	192	15.6 实现事件监听器	209
15.1.1 何时使用定制组件	192	15.6.1 实现值改变监听器	209
15.1.2 何时使用定制呈现器	193	15.6.2 实现动作监听器	210
		15.7 处理定制组件的事件	210
		15.8 在标记库描述文件中定义定制 组件标记	211
		15.9 使用定制组件	212
		15.10 创建和使用定制转换器	214
		15.10.1 创建定制转换器	214
		15.10.2 使用定制转换器	216
		15.11 创建和使用定制验证器	218
		15.11.1 实现验证器接口	219
		15.11.2 指定定制标记	220

15.11.3 使用定制验证器	221	16.5.2 引用错误消息	250
15.12 将组件值和实例绑定到托管 bean 属性	222	16.6 使用默认验证器	250
15.12.1 将组件值绑定到属性	223	16.7 注册定制验证器	251
15.12.2 将组件值绑定到隐式 对象	224	16.8 注册定制转换器	251
15.12.3 将组件实例绑定到 bean 属性	225	16.9 配置导航规则	252
15.13 将转换器、监听器和验证器 绑定到托管 bean 属性	226	16.10 向呈现包注册定制呈现器	254
第 16 章 配置 JSF 应用	227	16.11 注册定制组件	256
16.1 使用注解配置托管 bean	228	16.12 JSF 应用的基本需求	257
16.1.1 使用托管 bean 作用域	228	16.12.1 使用 Web 部署描述文件 配置应用	257
16.2 应用配置资源文件	229	16.12.2 配置工程阶段	260
16.2.1 配置 eager 应用作用域 托管 bean	230	16.12.3 包含类、页面和其他资源	260
16.2.2 应用配置资源文件的顺序	230		
16.3 使用 Faces 流	231	第 17 章 Java Servlet 技术	261
16.3.1 应用中打包流	233	17.1 什么是 servlet	262
16.3.2 最简单的流: simple-flow 示例应用	233	17.2 servlet 生命周期	262
16.3.3 checkout-module 示例应用	235	17.2.1 处理 servlet 生命周期事件	262
16.4 配置托管 bean	241	17.2.2 处理 servlet 错误	263
16.4.1 使用 managed-bean 元素	242	17.3 共享信息	264
16.4.2 使用 managed-property 元素 初始化属性	243	17.3.1 使用作用域对象	264
16.4.3 初始化映射和列表	248	17.3.2 控制对共享资源的并发 访问	264
16.5 注册应用消息	248	17.4 创建和初始化 servlet	265
16.5.1 使用 FacesMessage 创建 消息	249	17.5 编写服务方法	265
		17.5.1 从请求获取信息	266
		17.5.2 构造响应	266
		17.6 过滤请求和响应	267
		17.6.1 编写过滤器	268
		17.6.2 编写定制请求和响应	269
		17.6.3 指定过滤器映射	269
		17.7 调用其他 Web 资源	270

17.7.1	在响应中包含其他资源	271
17.7.2	控制转移到另一个 Web 组件	271
17.8	访问 Web 上下文	272
17.9	维护客户端状态	272
17.9.1	访问会话	272
17.9.2	将对象与会话关联	272
17.9.3	会话管理	272
17.9.4	会话跟踪	273
17.10	终结 servlet	273
17.10.1	跟踪服务请求	274
17.10.2	通知方法关闭	274
17.10.3	创建周到的长时间运行方法	275
17.11	使用 Java Servlet 技术上传文件	276
17.11.1	@MultipartConfig 注解	276
17.11.2	getParts 和 getPart 方法	277
17.12	异步处理	277
17.12.1	servlet 中的异步处理	277
17.12.2	等待资源	278
17.13	非阻塞 I/O	280
17.13.1	使用非阻塞 I/O 读取大 HTTP POST 请求	281
17.14	协议升级处理	282
17.15	mood 示例应用	284
17.15.1	mood 示例应用的组件	284
17.15.2	运行 mood 示例	284
17.16	fileupload 示例应用	285
17.16.1	fileupload 示例应用的	

	体系架构	285
17.16.2	运行 fileupload 示例	288
17.17	dukeetf 示例应用	289
17.17.1	dukeetf 示例应用的体系架构	289
17.17.2	运行 dukeetf 示例应用	293
17.18	关于 Java Servlet 技术的更多信息	294

第 18 章 WebSocket Java API 295

18.1	WebSocket 介绍	296
18.2	在 Java EE 平台中创建 WebSocket 应用	297
18.3	可编程端点	297
18.4	注解端点	298
18.5	发送和接收消息	299
18.5.1	发送消息	299
18.5.2	接收消息	300
18.6	维护客户端状态	301
18.7	使用编码器和解码器	301
18.7.1	实现编码器将 Java 对象转换为 WebSocket 消息	301
18.7.2	实现解码器将 WebSocket 消息转换为 Java 对象	303
18.8	路径参数	304
18.9	处理错误	305
18.10	指定端点配置器类	305
18.11	dukeetf2 示例应用	306
18.11.1	dukeetf2 示例应用的体系架构	306

18.11.2	运行 dukeetf2 示例应用	309
18.12	websocketbot 示例应用	310
18.12.1	websocketbot 示例应用的 体系架构	310
18.12.2	运行 websocketbot 示例 应用	314
18.13	关于 WebSocket 的更多信息	315

第 19 章 JSON 处理 316

19.1	JSON 简介	316
19.1.1	JSON 语法	316
19.1.2	JSON 的使用	317
19.1.3	生成和解析 JSON 数据	317
19.2	Java EE 平台中的 JSON 处理	318
19.3	使用对象模型 API	319
19.3.1	从 JSON 数据创建对象 模型	319
19.3.2	从应用代码创建对象模型	319
19.3.3	导航对象模型	320
19.3.4	将对象模型写至一个流	321
19.4	使用流 API	322
19.4.1	使用解析器读取 JSON 数据	322
19.4.2	使用生成器写 JSON 数据	324
19.5	Java EE RESTful Web 服务中的 JSON	324
19.6	jsonpmodel 示例应用	325
19.6.1	jsonpmodel 示例应用的 组成	325
19.6.2	运行 jsonpmodel 示例应用	325
19.7	jsonpstreaming 示例应用	326

19.7.1	jsonpstreaming 示例应用的 组成	326
19.7.2	运行 jsonpstreaming 示例 应用	326

19.8	关于 JSON 处理 Java API 的 更多信息	327
------	-------------------------------	-----

第 20 章 Web 应用国际化和本地化 328

20.1	Java 平台本地化类	328
20.2	提供本地化消息和标签	329
20.2.1	建立本地化环境	329
20.2.2	设置资源包	330
20.2.3	获取本地化环消息	331
20.3	日期和数字格式化	331
20.4	字符集和编码	332
20.4.1	字符集	332
20.4.2	字符编码	332

第四部分 bean 验证

第 21 章 bean 验证介绍 334

21.1	使用 bean 验证约束	334
21.2	验证 Null 和空串	336
21.3	验证构造函数和方法	337
21.3.1	跨参数约束	338
21.3.2	标识违反参数约束	338
21.3.3	为方法返回值增加约束	338
21.4	关于 bean 验证的更多信息	339

第 22 章 bean 验证：高级主题 340

22.1	创建定制约束	340
------	--------	-----

22.1.1	使用内置约束建立一个 新约束	340
22.1.2	去除约束目标的二义性	341
22.2	定制验证器消息	342
22.2.1	ValidationMessages 资源包	342
22.3	组合约束	342
22.3.1	定制组验证顺序	342
22.4	在类型层次体系中使用方法 约束	343
22.4.1	在类型层次体系中使用 方法约束的规则	344

第五部分 Java EE 的上下文和依赖注入

第 23 章 Java EE 的上下文和依赖

注入介绍	346
23.1 入门	347
23.2 CDI 概述	348
23.3 关于 bean	349
23.4 关于 CDI 托管 bean	350
23.5 bean 作为可注入的对象	350
23.6 使用限定符	351
23.7 注入 bean	352
23.8 使用作用域	352
23.9 提供 bean EL 名	354
23.10 增加设置和获取方法	354
23.11 Facelets 页面中使用托管 bean	355
23.12 使用生成器方法注入对象	355
23.13 配置 CDI 应用	356

23.14	对 CDI 托管 bean 使用 @Post- Construct 和 @PreDestroy 注解	356
23.14.1	使用 @PostConstruct 初始化托管 bean	356
23.14.2	使用 @PreDestroy 注解 准备撤销托管 bean	357
23.15	关于 CDI 的更多信息	357

第 24 章 运行基本上下文和依赖

注入示例	358
24.1 simplegreeting CDI 示例	358
24.1.1 simplegreeting 源文件	358
24.1.2 Facelets 模板和页面	359
24.1.3 运行 simplegreeting 示例	360
24.2 guessnumber-cdi CDI 示例	361
24.2.1 guessnumber-cdi 源文件	362
24.2.2 Facelets 页面	366
24.2.3 运行 guessnumber-CDI 示例	367

第 25 章 Java EE 的上下文和依赖

注入：高级主题	369
25.1 打包 CDI 应用	369
25.2 CDI 应用中使用替代对象	370
25.2.1 使用特殊化	371
25.3 CDI 应用中使用生成器方法、 生成器字段和清除器方法	372
25.3.1 使用生成器方法	372
25.3.2 使用生成器字段生成资源	373
25.3.3 使用清除器方法	373