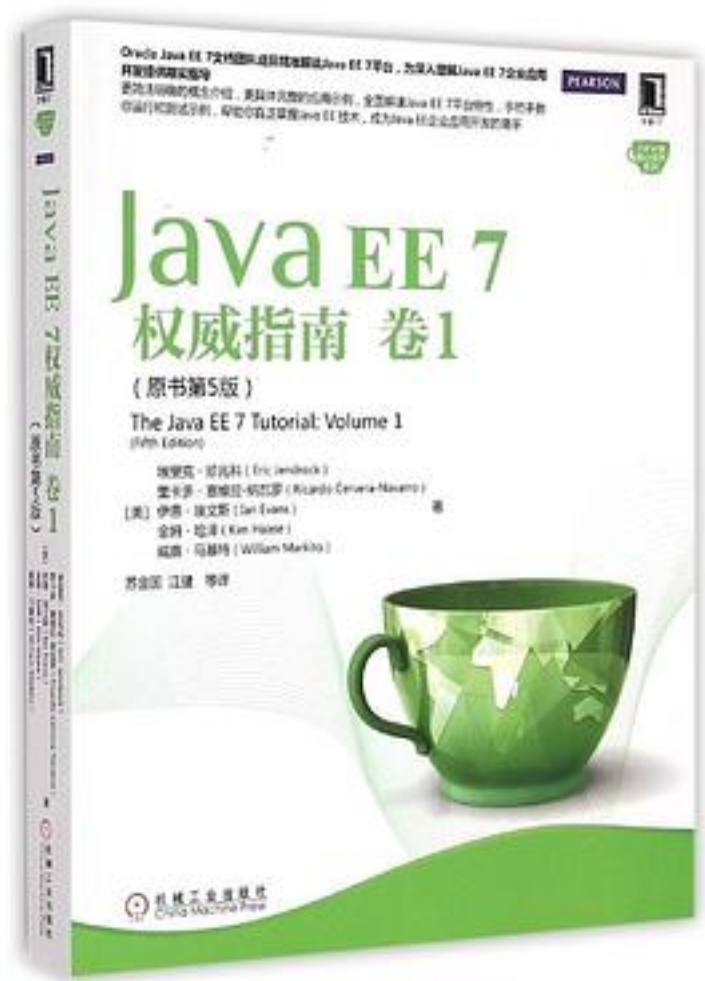


Java EE 7权威指南： 卷1



[Java EE 7权威指南： 卷1_ 下载链接1](#)

著者:[美] 埃里克·珍兆科

出版者:机械工业出版社

出版时间:2015-5

装帧:平装

isbn:9787111497608

本书是一个面向任务的实用指南，通过一系列实用示例来介绍如何开发Java EE 7(java平台企业版7)企业应用。这本书由Oracle的Java

EE文档小组成员倾力编写，会让初学者和中缀Java程序员深入了解这个平台。

这个指南包括平台特性的全面描述，并提供使用最新版本NetBeans IDE和GlassFish服务器开源版的有关说明书中不仅介绍了平台的基础知识，包括资源创建、资源注入和打包，还涵盖了多项相关技术，包括JavaServer Faces(JSF)、Java Servlet、WebSocket Java API、JSON处理Java API(JSON—P)、国际化和本地化、bean验证、Java EE上下文和依赖注入(CDI)以及Web服务JAX—WS和JAX—RS。

作者介绍:

本书作者均为Oracle Java EE 6文档团队成员，其中Eric Jendrock是Java EE Tutorial团队负责人，并编写了Java Servlet技术和Java安全性技术方面的文档，Ian Evan负责编写Enterprise JavaBeans、Java Persistence API、Bean Validation、Java Transaction API、JAX-RS和JAX-WS等方面的文档，Devika Gollapudi负责编写JavaServer Faces技术和Facelets等方面的文档，Kim Hasse负责编写面向Java EE平台的上下文和依赖注入、Java安全、JMS（Java Message Service）等方面的文档，Chinmayee Srivathsa负责编写资源连接方面的文档。

目录: 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
- 1.8.14 Java EE连接器架构 21
- 1.8.15 JavaMail API 22
- 1.8.16 Java容器授权契约 22
- 1.8.17 Java容器认证服务提供者接口 22
- 1.8.18 WebSocket Java API 22
- 1.8.19 JSON处理Java API 22
- 1.8.20 Java EE的并发工具 23
- 1.8.21 Java平台的批处理应用 23
- 1.9 Java平台标准版7的Java EE 7 API 23
- 1.9.1 Java数据库互连API 23
- 1.9.2 Java命名和目录接口API 23
- 1.9.3 JavaBeans激活框架 24
- 1.9.4 XML处理Java API 24
- 1.9.5 Java XML绑定架构 24
- 1.9.6 XML Web服务Java API 25
- 1.9.7 带附件SOAP Java API 25
- 1.9.8 Java认证和授权服务 25
- 1.9.9 Java平台的公共注解 25
- 1.10 GlassFish服务器工具 25
- 第2章 使用教程示例 27
- 2.1 必要软件 27
- 2.1.1 Java EE 7软件开发包 28
- 2.1.2 Java平台标准版 28
- 2.1.3 Java EE 7教程组件 28
- 2.1.4 NetBeans IDE 29
- 2.1.5 Apache Maven 30
- 2.2 启动和停止GlassFish服务器 30
- 2.2.1 使用NetBeans IDE启动GlassFish服务器 30
- 2.2.2 使用NetBeans IDE停止GlassFish服务器 30
- 2.2.3 使用命令行启动GlassFish服务器 30
- 2.2.4 使用命令行停止GlassFish服务器 31
- 2.3 启动管理控制台 31
- 2.3.1 使用NetBeans IDE启动管理控制台 31
- 2.4 启动和停止Java DB服务器 31
- 2.4.1 使用NetBeans IDE启动数据库服务器 32
- 2.5 构建示例 32
- 2.6 教程示例目录结构 32
- 2.7 教程中的Java EE 7 Maven原型 33
- 2.7.1 安装教程Maven原型 33
- 2.8 获取教程最新更新 33
- 2.8.1 使用NetBeans IDE更新教程 33
- 2.8.2 使用命令行更新教程 34
- 2.9 调试Java EE应用 34
- 2.9.1 使用服务器日志 34
- 2.9.2 使用调试器 35
- 第二部分 平台基础知识

- 第3章 资源创建 38
 - 3.1 资源和JNDI命名 38
 - 3.2 数据源对象和连接池 39
 - 3.3 以管理方式创建资源 40
- 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	127
10.2.6 使用h:graphicImage标记增加图形图像	128
10.2.7 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	138
10.2.14 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
10.2.16 bookmarks示例应用	140
10.2.17 使用h:outputScript和h:outputStylesheet标记实现资源重定位	141
10.3 使用核心标记	143

- 第11章 使用转换器、监听器和验证器 146
 - 11.1 使用标准转换器 146
 - 11.1.1 转换组件值 147
 - 11.1.2 使用DateTimeConverter 148
 - 11.1.3 使用NumberConverter 149
 - 11.2 在组件上注册监听器 150
 - 11.2.1 在组件上注册值改变监听器 151
 - 11.2.2 在组件上注册动作监听器 151
 - 11.3 使用标准验证器 153
 - 11.3.1 验证组件值 154
 - 11.3.2 使用验证器标记 154
 - 11.4 引用托管bean方法 155
 - 11.4.1 引用完成导航的方法 156
 - 11.4.2 引用处理动作事件的方法 156
 - 11.4.3 引用完成验证的方法 156
 - 11.4.4 引用处理值改变事件的方法 157
- 第12章 使用JSF技术开发应用 158
 - 12.1 JSF技术中的托管bean 158
 - 12.1.1 创建托管bean 158
 - 12.1.2 使用EL引用托管bean 159
 - 12.2 编写bean属性 160
 - 12.2.1 编写绑定到组件值的属性 161
 - 12.2.2 编写绑定到组件实例的属性 166
 - 12.2.3 编写绑定到转换器、监听器或验证器的属性 167
 - 12.3 编写托管bean方法 168
 - 12.3.1 编写处理导航的方法 168
 - 12.3.2 编写处理动作事件的方法 169
 - 12.3.3 编写完成验证的方法 170
 - 12.3.4 编写处理值改变事件的方法 170
- 第13章 结合使用Ajax和JSF技术 172
 - 13.1 Ajax概述 173
 - 13.2 结合JSF技术使用Ajax功能 173
 - 13.3 结合Facelets使用Ajax 174
 - 13.3.1 使用f:ajax标记 174
 - 13.4 发送Ajax请求 175
 - 13.4.1 使用event属性 175
 - 13.4.2 使用execute属性 176
 - 13.4.3 使用immediate属性 176
 - 13.4.4 使用listener属性 177
 - 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
 - 13.10.1 在Facelets应用中使用 JavaScript API 180
 - 13.10.2 bean类中使用@Resource-Dependency注解 181
 - 13.11 ajaxguessnumber示例应用 181
 - 13.11.1 ajaxguessnumber源文件 181
 - 13.11.2 运行ajaxguessnumber示例 183
 - 13.12 关于JSF技术中Ajax的更多信息 184
- 第14章 复合组件：高级主题与示例 185
 - 14.1 复合组件的属性 185
 - 14.2 调用托管bean 186

- 14.3 验证复合组件值 186
- 14.4 compositecomponentexample示例应用 187
 - 14.4.1 复合组件文件 187
 - 14.4.2 使用页面 188
 - 14.4.3 托管bean 189
 - 14.4.4 运行compositecomponentexample示例 189
- 第15章 创建定制UI组件和其他定制对象 191
 - 15.1 确定是否需要定制组件或呈现器 192
 - 15.1.1 何时使用定制组件 192
 - 15.1.2 何时使用定制呈现器 193
 - 15.1.3 组件、呈现器和标记组合 194
 - 15.2 了解图像地图示例 195
 - 15.2.1 为什么使用JSF技术实现图像地图 195
 - 15.2.2 了解呈现的HTML 195
 - 15.2.3 了解Facelets页面 196
 - 15.2.4 配置模型数据 197
 - 15.2.5 图像地图应用类总结 198
 - 15.3 创建定制组件的步骤 199
 - 15.4 创建定制组件类 199
 - 15.4.1 指定组件族 202
 - 15.4.2 完成编码 202
 - 15.4.3 完成解码 204
 - 15.4.4 允许组件属性接受表达式 204
 - 15.4.5 保存和恢复状态 205
 - 15.5 将呈现委托到呈现器 207
 - 15.5.1 创建呈现器类 207
 - 15.5.2 标识呈现器类型 208
 - 15.6 实现事件监听器 209
 - 15.6.1 实现值改变监听器 209
 - 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
 - 15.12 将组件值和实例绑定到托管bean属性 222
 - 15.12.1 将组件值绑定到属性 223
 - 15.12.2 将组件值绑定到隐式对象 224
 - 15.12.3 将组件实例绑定到bean属性 225
 - 15.13 将转换器、监听器和验证器绑定到托管bean属性 226
- 第16章 配置JSF应用 227
 - 16.1 使用注解配置托管bean 228
 - 16.1.1 使用托管bean作用域 228
 - 16.2 应用配置资源文件 229
 - 16.2.1 配置eager应用作用域托管bean 230
 - 16.2.2 应用配置资源文件的顺序 230
 - 16.3 使用Faces流 231
 - 16.3.1 应用中打包流 233
 - 16.3.2 最简单的流：simple-flow示例应用 233

- 16.3.3 checkout-module示例应用 235
- 16.4 配置托管bean 241
 - 16.4.1 使用managed-bean元素 242
 - 16.4.2 使用managed-property元素初始化属性 243
 - 16.4.3 初始化映射和列表 248
- 16.5 注册应用消息 248
 - 16.5.1 使用FacesMessage创建消息 249
 - 16.5.2 引用错误消息 250
- 16.6 使用默认验证器 250
- 16.7 注册定制验证器 251
- 16.8 注册定制转换器 251
- 16.9 配置导航规则 252
- 16.10 向呈现包注册定制呈现器 254
- 16.11 注册定制组件 256
- 16.12 JSF应用的基本需求 257
 - 16.12.1 使用Web部署描述文件配置应用 257
 - 16.12.2 配置工程阶段 260
 - 16.12.3 包含类、页面和其他资源 260
- 第17章 Java Servlet技术 261
 - 17.1 什么是servlet 262
 - 17.2 servlet生命周期 262
 - 17.2.1 处理servlet生命周期事件 262
 - 17.2.2 处理servlet错误 263
 - 17.3 共享信息 264
 - 17.3.1 使用作用域对象 264
 - 17.3.2 控制对共享资源的并发访问 264
 - 17.4 创建和初始化servlet 265
 - 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
 - 25.4 CDI应用中使用预定义bean 374
 - 25.5 CDI应用中使用事件 375
 - 25.5.1 定义事件 375
 - 25.5.2 使用观察者方法处理事件 376
 - 25.5.3 触发事件 377
 - 25.6 CDI应用中使用拦截器 378
 - 25.7 CDI应用中使用装饰器 379
 - 25.8 CDI应用中使用构造型 380
- 第26章 运行高级上下文和依赖注入示例 382
 - 26.1 encoder示例：使用替代对象 382
 - 26.1.1 Coder接口和实现 383
 - 26.1.2 encoder Facelets页面和托管bean 383
 - 26.1.3 运行encoder示例 385
 - 26.2 producermethods示例：使用生成器方法选择bean实现 387
 - 26.2.1 producermethods示例的组成 387
 - 26.2.2 运行producermethods示例 388
 - 26.3 producerfields示例：使用生成器字段生成资源 389
 - 26.3.1 producerfields示例的生成器字段 389
 - 26.3.2 producerfields实体和会话bean 391
 - 26.3.3 producerfields Facelets页面和托管bean 392
 - 26.3.4 运行producerfields示例 394
 - 26.4 billpayment示例：使用事件和拦截器 395
 - 26.4.1 PaymentEvent事件类 395
 - 26.4.2 PaymentHandler事件监听器 396
 - 26.4.3 billpayment Facelets页面和托管bean 396
 - 26.4.4 LoggedInterceptor拦截器类 399
 - 26.4.5 运行billpayment示例 400
 - 26.5 decorators示例：装饰bean 401
 - 26.5.1 decorators示例的组成 401
 - 26.5.2 运行decorators示例 402
- 第六部分 Web服务
- 第27章 Web服务介绍 406
 - 27.1 什么是Web服务 406
 - 27.2 Web服务类型 406
 - 27.2.1 大 Web服务 407
 - 27.2.2 RESTful Web服务 407
 - 27.3 确定使用何种类型的Web服务 408
- 第28章 用JAX-WS构建Web服务 409
 - 28.1 使用JAX-WS创建一个简单的Web服务和客户端 410
 - 28.1.1 JAX-WS端点的需求 411
 - 28.1.2 编写服务端点实现类 411

- 28.1.3 构建、打包和部署服务 412
- 28.1.4 测试Web服务端点的方法 412
- 28.1.5 一个简单的JAX-WS应用客户端 413
- 28.1.6 一个简单的JAX-WS Web客户端 414
- 28.2 JAX-WS支持的类型 417
 - 28.2.1 模式-Java映射 417
 - 28.2.2 Java-模式映射 418
- 28.3 Web服务互操作性和JAX-WS 419
- 28.4 关于JAX-WS的更多信息 419
- 第29章 用JAX-RS构建RESTful Web服务 420
 - 29.1 什么是RESTful Web服务 420
 - 29.2 创建RESTful根资源类 421
 - 29.2.1 用JAX-RS开发RESTful Web服务 421
 - 29.2.2 JAX-RS应用概览 422
 - 29.2.3 @Path注解和URI路径模板 424
 - 29.2.4 响应HTTP方法和请求 425
 - 29.2.5 使用@Consumes和@Produces定制请求和响应 428
 - 29.2.6 抽取请求参数 430
 - 29.2.7 配置JAX-RS应用 433
 - 29.3 JAX-RS示例应用 434
 - 29.3.1 创建一个简单的RESTful Web服务 434
 - 29.3.2 rsvp示例应用 435
 - 29.3.3 真实示例 438
 - 29.4 关于JAX-RS的更多信息 438
- 第30章 用JAX-RS客户端API访问REST资源 439
 - 30.1 客户端API概述 439
 - 30.1.1 使用客户端API创建一个基本的客户端请求 439
 - 30.2 JAX-RS示例应用中使用客户端API 442
 - 30.2.1 rsvp示例应用中的客户端API 442
 - 30.2.2 customer示例应用中的客户端API 443
 - 30.3 客户端API的高级特性 445
 - 30.3.1 配置客户端请求 445
 - 30.3.2 客户端API中的异步调用 447
- 第31章 JAX-RS: 高级主题与示例 449
 - 31.1 资源类字段和bean属性的注解 449
 - 31.1.1 抽取路径参数 450
 - 31.1.2 抽取查询参数 450
 - 31.1.3 抽取表单数据 451
 - 31.1.4 抽取请求或响应的Java类型 451
 - 31.2 用bean验证来验证资源数据 452
 - 31.2.1 对资源方法使用约束注解 452
 - 31.2.2 验证实体数据 453
 - 31.2.3 验证异常处理和响应码 455
 - 31.3 子资源和运行时资源解析 455
 - 31.3.1 子资源方法 455
 - 31.3.2 子资源定位符 456
 - 31.4 JAX-RS与EJB技术和CDI集成 457
 - 31.5 条件HTTP请求 458
 - 31.6 运行时内容协商 459
 - 31.7 结合使用JAX-RS和JAXB 460
 - 31.7.1 使用Java对象为数据建模 462
 - 31.7.2 从现有的XML模式定义开始 463
 - 31.7.3 结合JAX-RS和JAXB使用JSON 465
 - 31.8 customer示例应用 466

31.8.1 customer示例应用概述 466
31.8.2 Customer和Address实体类 466
31.8.3 CustomerService类 469
31.8.4 CustomerBean类中使用JAX-RS客户端 470
31.8.5 运行customer示例 472
• • • • • ([收起](#))

[Java EE 7权威指南：卷1_下载链接1](#)

标签

JavaEE

Java

j2ee

J2EE

技术

Buy

评论

是官方文档的翻译。讲解很明了，不过不够深度，都是入门级别的。但是读完自己的脑中可以构建出完整的 JAVAEE 体系，很不错，需要和卷2配合阅读。

[Java EE 7权威指南：卷1_下载链接1](#)

书评

首先，这本书就是Java EE 6

Tutorial的一个实体书版本，所以有英文基础的朋友还是直接看文档吧，毕竟不花钱而且还能方便的存放到pad上。

连接给出：<http://docs.oracle.com/javaee/6/tutorial/doc/index.html>

然后吐槽一下翻译，Tutorial居然能给翻译成权威指南，真不知道该说...

内容太单薄，总体给人的感觉是，我知道的它讲了。我不知道的，或关心的，它没讲。举个简单的例子，ManagedBean如果设置为@ConversationScoped，那么如何管理Conversation的起始和结束。书中没有，还得网上去查资料。再比如，JSF的页面导航，书中居然没有这方面的内容。 还...

[Java EE 7权威指南：卷1 下载链接1](#)