

<<Java EE 6权威指南>>

图书基本信息

书名：<<Java EE 6权威指南>>

13位ISBN编号：9787115290434

10位ISBN编号：7115290431

出版时间：2012-9

出版时间：人民邮电出版社

作者：[美]Eric Jendrock,[美] Ian Evan,[美]Devika Gollapudi,[美]Kim Hasse,[美]Chinmayee Srivathsa

页数：404

字数：626000

译者：李 鹏,韩 智

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>

<<Java EE 6权威指南>>

前言

本书基于GlassFish服务器开源版讲述如何在Java EE 6平台上开发企业应用。

Oracle GlassFish Server是一个Java EE兼容的应用服务器，它基于GlassFish服务器开源版开发而成。

GlassFish服务器开源版是目前业界领先的开源项目和开放平台，可用来构建和部署下一代应用和服务。

GlassFish服务器开源版是由GlassFish项目开源社区研发的，其官方网站为<http://glassfish.java.net>，它是第一个兼容Java EE 6平台规范的应用服务器，具备轻量、灵活和开源等特点，这使得企业不仅可以利用Java EE 6带来的新功能，还能通过更快捷的开发和部署流程在已有功能的基础上添加新功能。

Oracle GlassFish Server又称GlassFish服务器商业版，它和GlassFish服务器开源版在本书中统称为GlassFish服务器。

阅读须知Oracle GlassFish Server文档集相关文档符号约定字体约定默认路径和文件名方面的约定文档、支持以及培训查找Oracle产品文档第三方网站参考内容阅读须知读者应当事先熟悉Java这门编程语言，如果并不熟悉，建议大家先通读Addison-Wesley在2006年出版、由Sharon Zakhour等人所著的The Java Tutorial, Fourth Edition。

Oracle GlassFish Server文档集GlassFish服务器文档集讲述的是部署计划以及系统的安装。

GlassFish服务器文档的网址是<http://docs.sun.com/coll/1343.13>。

有关GlassFish服务器的介绍，可以按顺序参阅表1列出的文档。

表1 GlassFish服务器文档集里的文档文档名 描述信息Release Notes（版本说明）提供有关软件和文档的最新信息，以表格的形式全面列出它所支持的硬件、操作系统、Java开发包（JDK）以及数据库的驱动程序等信息Quick Start Guide（快速入门指南）讲述如何使用GlassFish服务器Installation Guide（安装指南）讲述如何安装GlassFish服务器软件以及它的各个组件Upgrade Guide（升级指南）讲述如何升级到最新版本的GlassFish服务器，并介绍相邻版本之间的差异、配置选项的差异，这些差异可能会导致与产品规范不兼容Administration Guide（管理指南）讲述如何使用asadmin（1 M）实用工具，以命令行的方式配置、监控和管理GlassFish服务器的各个子系统和组件。

管理控制台（Administration Console）的在线帮助里介绍了如何执行这些任务Application Deployment Guide（应用部署指南）讲述如何把应用装配和部署到GlassFish服务器上，还提供部署描述文件的相关信息Your First Gup：An Introduction to the Java EE Platform（Java EE平台简介）为Java EE的初学者提供简短的教程，讲述开发简单的企业应用的完整过程。

示例程序是一个Web程序，包括一个基于EJB规范的组件、一个基于JAX-RS的Web服务，以及一个用于Web前端的JavaServer Faces组件Application Development Guide（应用开发指南）讲述如何创建和实现运行在GlassFish服务器上的Java EE应用。

这些应用遵循开放Java标准模型，使用的是Java EE组件以及API（Application Programming Interface，应用编程接口）。

本指南还提供有关开发者工具、安全以及调试的信息Add-On Component Development Guide（插件开发指南）讲述如何使用GlassFish服务器已发布的接口为GlassFish服务器开发插件。

本文档只讲述那些能让插件应用于GlassFish服务器的必要步骤Embedded Server Guide（嵌入式服务器指南）讲述如何在嵌入式GlassFish服务器环境下运行应用及如何开发嵌入GlassFish服务器的应用Scripting Framework Guide（脚本框架指南）讲述如何使用Ruby on Rails和Groovy on Grails等脚本语言为GlassFish服务器开发脚本应用Troubleshooting Guide（疑难排解指南）讲述使用GlassFish服务器过程中的常见问题及解决办法Error Message Reference（出错消息参考）讲述在使用GlassFish服务器过程中可能遇到的出错消息Reference Manual（参考手册）提供手册页（manual page）格式的参考信息，讲述GlassFish服务器的管理命令、实用工具命令以及相关的概念Domain File Format Reference（域配置文件格式参考）讲述GlassFish服务器的配置文件domain.xml的格式Java EE 6 Tutorial（Java EE 6教程）讲述如何使用Java EE 6平台技术和API开发Java EE应用Message Queue Release Notes（消息队列版本说明）讲述GlassFish消息队列的新特性、兼容性问题以及已知bugMessage Queue Administration Guide（消息队列管理指南）讲述如何创建和管理消息队列系统Message Queue Developer's Guide for JMX Clients（JMX客户端

<<Java EE 6权威指南>>

的消息队列开发指南) 讲述消息队列相关的API, 以便开发人员在遵循JMX (Java Management Extensions, Java管理扩展) 的前提下, 通过编程的方式配置和监控消息队列的资源情况。

相关文档GlassFish服务器还提供了基于Javadoc工具的开发包参考文档, 可以以如下方式访问。

Java EE 6的API的规范位于<http://docs.oracle.com/javaee/6/api/>。

GlassFish服务器 3.0.1版的API规范, 包括Java EE 6平台的包以及GlassFish服务器专用的、非平台的包, 位于<http://glassfish.java.net/nonav/docs/v3/api/>。

此外, Java EE规范也有可能用得着, 其网址为<http://www.oracle.com/technetwork/java/javaee/tech/index.html>。

关于如何用NetBeans这个IDE (Integrated Development Environment, 集成开发环境) 创建企业应用, 参见<http://www.netbeans.org/kb/>。

关于如何为GlassFish服务器采用Java DB数据库, 参见<http://www.oracle.com/technetwork/java/javadb/overview/index.html>。

GlassFish的示例项目包括许多应用, 这些应用展示了广泛的Java EE技术。

GlassFish示例项目和Java EE SDK打包到了一起, 可以在GlassFish示例项目页<http://glassfish-samples.java.net/>获得。

符号约定表2展示了本书中可能用到的符号。

表2 符号约定符号 描述 例子 意义 [] 括号里是可选参数或者命令的选项 ls [-l] -l选项不是必需的, 可以不用{ | }

对于一个必选的命令选项, 必须从括号里的几个备选项中选择一个 -d {y|n} -d选项要求必须用y或n作为参数 \${ } 指对变量的引用 \${com.sun.javaRoot} 表示com.sun.javaRoot这个变量的值- 同时按下多个按键 Control+A 同时按下Ctrl键和A键+ 顺序按下多个按键 Ctrl+A+N 按下Ctrl键, 然后松开, 接着顺序按下A键和N键→ 指的是在图形用户界面里菜单项的选择 File→New→Templates 从File菜单里选择New, 在New子菜单里选择Templates 字体约定表3展示的是字体约定。

表3 字体约定字体 意义 例子 AaBbCc123 命令以及计算机屏幕上的输出信息 使用ls -a列出所有文件 machine_name% you have mail. (续) 字体 意义 例子 AaBbCc123 用户的输入用粗体, 以区别于计算机的输出 machine_name% suPassword : AaBbCc123 占位符, 会被真正的名字或者值替换掉 删除文件的命令是rm filename 楷体 新的术语以及需要强调的重点术语 (注意, 有一些强调的项以粗体形式出现) 缓存是指本地保存的一份数据副本 不要保存文件默认路径和文件名方面的约定表4列出了本书会用到的默认路径和文件名。

表4 默认路径和文件名占位符 描述信息 默认值 as-install 表示GlassFish服务器的安装路径或者SDK的安装路径 (如果GlassFish服务器打包至SDK中) 在Solaris操作系统、Linux操作系统或者Mac操作系统下时为: user’s-home-directory/glassfishv3/glassfish 在Windows系统下时为: SystemDrive : \glassfishv3\glassfish as-install-parent 表示GlassFish服务器安装路径的上一级路径 在Solaris操作系统、Linux操作系统或者Mac操作系统下时为: user’s-home-directory/glassfishv3 在Windows系统下时为: SystemDrive : \glassfishv3\glassfish tut-install 表示安装完GlassFish服务器或者SDK后运行Update工具时, “Java EE Tutorial” (Java EE教程) 所在的路径 as-install/docs/javaee-tutorial domain-root-dir 表示创建域 (domain) 的默认保存位置 as-install/domains/domain-dir 保存域配置文件的地方在配置文里, domain-dir表示为: \$(com.sun.aas.instanceRoot) domain-root-dir/domain-name 文档、支持以及培训 Oracle网站提供了如下一些辅助资源: 文档信息, 参见<http://docs.sun.com/>; 支持信息, 参见<http://www.sun.com/support/>; 培训信息, 参见<http://education.oracle.com/>。

查找Oracle产品文档除了可以在<http://docs.sun.com>网站查找Oracle产品的文档以外, 还可以使用搜索引擎, 在搜索框里按如下语法格式输入信息: 搜索关键词 site: docs.sun.com 举例来说, 想查找“broker”的相关内容, 就在搜索框里输入: broker site: docs.sun.com 为了查找到Oracle其他网站的信息, 比如说Oracle技术网络中的Java开发者网站 (<http://www.oracle.com/technetwork/java/index.html>), 就在输入框里把上述docs.sun.com替换成oracle.com

。第三方网站参考内容本书会引用一些第三方网站的参考内容, 主要是为了提供相关的辅助信息。

注意 Oracle不对本书所引用第三方网站的可访问性负责。

<<Java EE 6权威指南>>

对于第三方网站上及可通过其访问的任何内容、广告、产品以及其他材料，Oracle并不表示认可，也不对其负责。

任何对上述内容（内容、产品或者服务）的使用，Oracle不对其导致的实际或者潜在的损害或损失负任何责任。

<<Java EE 6权威指南>>

内容概要

《Java EE 6权威指南：基础篇(第4版)》是任务导向、示例驱动的Java EE 6基础教程，讲述如何开发企业应用，并作为第4版加入了很多新内容，详细介绍了最新的API和平台特性。本书分8部分阐述了Java EE 6基础知识、Web层技术(包括JavaServer Faces和Facelets)，以及如何使用JAX-WS和JAX-RS构建Web服务、使用JavaBeans组件开发业务逻辑、使用Java Persistence API访问数据库、为Web应用或企业应用增强安全性、在Java EE平台使用上下文和依赖注入。

《Java EE 6权威指南：基础篇(第4版)》适合初中级Java编程人员学习参考。

<<Java EE 6权威指南>>

作者简介

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。

<<Java EE 6权威指南>>

书籍目录

目 录

第一部分 简介

第1章 综述 2

- 1.1 Java EE 6平台新特性 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 5
 - 1.3.2 Java EE组件 5
 - 1.3.3 Java EE客户端 5
 - 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 9
- 1.4 Java EE容器 9
 - 1.4.1 容器服务 10
 - 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应用程序装配与部署 13
- 1.7 打包应用程序 13
- 1.8 开发中的角色分工 14
 - 1.8.1 Java EE产品提供商 15
 - 1.8.2 工具提供商 15
 - 1.8.3 应用组件提供商 15
 - 1.8.4 应用程序装配者 15
 - 1.8.5 应用程序部署者和管理员 16
- 1.9 Java EE 6 API 16
 - 1.9.1 企业JavaBeans技术 19
 - 1.9.2 Java Servlet技术 20
 - 1.9.3 JavaServer Faces技术 20
 - 1.9.4 JavaServer Pages技术 20
 - 1.9.5 JavaServer Pages标准标签库 21
 - 1.9.6 Java Persistence API 21
 - 1.9.7 Java Transaction API 21
 - 1.9.8 支持REST式Web服务的Java API 21
 - 1.9.9 Managed Beans 21
 - 1.9.10 Java EE平台(JSR 299)的上下文与依赖注入 22
 - 1.9.11 Java Dependency Injection(JSR 330) 22
 - 1.9.12 Bean Validation 22
 - 1.9.13 Java Message Service API 22
 - 1.9.14 Java EE Connector架构 22
 - 1.9.15 JavaMail API 23
 - 1.9.16 Java容器授权合约 23
 - 1.9.17 Java容器认证服务提供商接口 23

<<Java EE 6权威指南>>

- 1.10 Java标准版6.0中的Java EE6 API 23
 - 1.10.1 Java Database ConnectivityAPI 24
 - 1.10.2 Java命名和目录接口API 24
 - 1.10.3 JavaBeans ActivationFramework 24
 - 1.10.4 Java XML API 24
 - 1.10.5 Java XML绑定架构 25
 - 1.10.6 支持带附件的SOAP消息API 25
 - 1.10.7 基于XML的Java WebServices API 25
 - 1.10.8 Java认证与授权服务 25
- 1.11 GlassFish服务器工具 26
- 第2章 使用教程示例 27
 - 2.1 必备软件 27
 - 2.1.1 Java 平台标准版(J2SE) 27
 - 2.1.2 Java EE 6 SDK 28
 - 2.1.3 Java EE 6教程组件 28
 - 2.1.4 NetBeans集成开发环境 29
 - 2.1.5 Apache Ant 30
 - 2.2 启动和关闭GlassFish服务器 30
 - 2.3 启动管理控制台 31
 - 2.4 启动和关闭Java DB服务器 32
 - 2.5 构建示例 32
 - 2.6 教程示例的目录结构 32
 - 2.7 获取最新版的教程 33
 - 2.8 调试Java EE应用 33
 - 2.8.1 服务器日志 33
 - 2.8.2 调试器 34
- 第二部分 Web层
- 第3章 Web应用初步 36
 - 3.1 Web应用 36
 - 3.2 Web应用的生命周期 37
 - 3.3 Web模块示例——hello1 39
 - 3.3.1 hello1 Web模块分析 40
 - 3.3.2 打包Web模块 42
 - 3.3.3 部署Web模块 43
 - 3.3.4 运行已部署的Web模块 44
 - 3.3.5 查看已部署的Web模块 44
 - 3.3.6 更新Web模块 45
 - 3.3.7 动态加载 45
 - 3.3.8 卸载Web模块 46
 - 3.4 配置Web应用之hello2示例 46
 - 3.4.1 映射URL到Web组件 46
 - 3.4.2 hello2Web模块分析 47
 - 3.4.3 构建、打包、部署以及运行hello2应用 48
 - 3.4.4 设置welcome文件 49
 - 3.4.5 设置上下文以及初始化参数 49
 - 3.4.6 映射错误信息到出错页面 50
 - 3.4.7 资源引用声明 51

<<Java EE 6权威指南>>

- 3.5 Web应用的更多信息 52
- 第4章 JSF技术 54
 - 4.1 什么是JSF应用 54
 - 4.2 JSF技术的优势 55
 - 4.3 创建简单的JSF应用 56
 - 4.3.1 开发后台bean 57
 - 4.3.2 创建网页 57
 - 4.3.3 映射FacesServlet实例 58
 - 4.3.4 hello应用程序的生命周期 58
 - 4.4 有关JSF技术的更多信息 59
- 第5章 Facelets介绍 60
 - 5.1 什么是Facelets 60
 - 5.2 开发简单的Facelets应用 61
 - 5.2.1 创建Facelets应用 61
 - 5.2.2 配置应用 64
 - 5.2.3 构建、打包、部署以及运行guessnumber Facelets示例 65
 - 5.3 模板化 66
 - 5.4 复合组件 68
 - 5.5 资源 70
- 第6章 表达式语言 72
 - 6.1 EL综述 72
 - 6.2 即时求值和延后求值语法 73
 - 6.2.1 即时求值 73
 - 6.2.2 延后求值 73
 - 6.3 值表达式和方法表达式 74
 - 6.3.1 值表达式 74
 - 6.3.2 方法表达式 77
 - 6.4 定义标签属性类型 79
 - 6.5 文本表达式 80
 - 6.6 运算符 80
 - 6.7 保留字 81
 - 6.8 EL表达式的例子 81
- 第7章 在网页中使用JSF技术 83
 - 7.1 设置页面 83
 - 7.2 使用HTML标签为页面添加组件 84
 - 7.2.1 组件标签的通用属性 85
 - 7.2.2 添加HTML的head和body标签 87
 - 7.2.3 添加表单组件 88
 - 7.2.4 文本组件 88
 - 7.2.5 使用命令型组件标签执行动作和导航 92
 - 7.2.6 用h:graphicImage添加图像和图形 93
 - 7.2.7 用h:panelGrid和h:panelGroup实现组件布局 93
 - 7.2.8 显示选项组件 94
 - 7.2.9 显示多项选择组件 96
 - 7.2.10 使用f:selectItem和f:selectItems标签 97
 - 7.2.11 使用数据绑定表组件 98
 - 7.2.12 使用h:message和h:messages显示出错信息 101

<<Java EE 6权威指南>>

- 7.2.13 使用h:button和h:link标签创建可加入收藏夹的URL 102
- 7.2.14 使用视图参数配置可加入收藏夹的URL 102
- 7.2.15 使用h:output标签实现资源再定位 103
- 7.3 核心标签 104
- 第8章 转换器、监听器和验证器 106
 - 8.1 标准转换器 106
 - 8.1.1 转换组件的值 107
 - 8.1.2 DateTimeConverter 108
 - 8.1.3 NumberConverter 109
 - 8.2 为组件注册监听器 110
 - 8.2.1 为组件注册一个值变更监听器 110
 - 8.2.2 为组件注册动作监听器 111
 - 8.3 标准验证器 111
 - 8.3.1 验证组件的值 112
 - 8.3.2 LongRange Validator 112
 - 8.4 引用后台bean的方法 112
 - 8.4.1 引用执行页面导航的方法 113
 - 8.4.2 引用处理动作事件的方法 113
 - 8.4.3 引用执行验证逻辑的方法 114
 - 8.4.4 引用处理值变更事件的方法 114
- 第9章 用JSF技术开发Web应用 115
 - 9.1 后台bean 115
 - 9.1.1 创建后台bean 115
 - 9.1.2 使用EL引用后台bean 116
 - 9.2 为后台bean的属性编写代码 117
 - 9.2.1 为绑定到组件值的属性编写代码 118
 - 9.2.2 为绑定到组件实例的属性编写代码 122
 - 9.2.3 为绑定到转换器、监听器以及验证器的属性编写代码 123
 - 9.3 为后台bean的方法编写代码 124
 - 9.3.1 编写处理导航的方法 124
 - 9.3.2 编写处理动作事件的方法 125
 - 9.3.3 编写执行验证的方法 126
 - 9.3.4 编写处理值变更事件的方法 126
 - 9.4 使用Bean验证 127
- 第10章 Java Servlet技术 130
 - 10.1 什么是servlet 130
 - 10.2 servlet生命周期 131
 - 10.2.1 处理servlet生命周期内的事件 131
 - 10.2.2 处理servlet错误 132
 - 10.3 共享信息 132
 - 10.3.1 有作用域的对象 133
 - 10.3.2 控制对共享资源的并发访问 133
 - 10.4 创建以及初始化servlet 133
 - 10.5 编写service方法 134
 - 10.5.1 从请求里提取信息 134
 - 10.5.2 构造应答信息 135
 - 10.6 过滤请求和应答 136

<<Java EE 6权威指南>>

- 10.6.1 编程实现过滤器 136
- 10.6.2 通过编程定制请求和应答 137
- 10.6.3 设定过滤器映射 138
- 10.7 调用其他Web资源 139
 - 10.7.1 在应答里包含其他资源 139
 - 10.7.2 转交控制权给其他Web组件 140
- 10.8 访问Web上下文 140
- 10.9 维护客户端状态 140
 - 10.9.1 访问会话 141
 - 10.9.2 把对象关联到会话 141
 - 10.9.3 会话管理 141
 - 10.9.4 会话追踪 142
- 10.10 结束servlet 142
 - 10.10.1 追踪服务请求 142
 - 10.10.2 将关闭事件通知方法 143
 - 10.10.3 妥善处理长时方法 143
- 10.11 mood示例应用 144
 - 10.11.1 mood示例应用里的组件 144
 - 10.11.2 构建、打包、部署以及运行mood示例 144
- 10.12 有关Java Servlet技术的更多信息 145
- 第三部分 Web服务
- 第11章 Web服务简介 148
 - 11.1 什么是Web服务 148
 - 11.2 Web服务的类型 148
 - 11.2.1 “重量级”的Web服务 148
 - 11.2.2 REST式Web服务 149
 - 11.3 Web服务类型的选用 150
- 第12章 用JAX-WS构建Web服务 151
 - 12.1 用JAX-WS开发简单的Web服务和客户端 152
 - 12.1.1 对JAX-WS端点的要求 152
 - 12.1.2 编写服务端点实现类 153
 - 12.1.3 构建、打包及部署服务 153
 - 12.1.4 测试Web服务端点中的方法 154
 - 12.1.5 简单的JAX-WS应用客户端 155
 - 12.1.6 简单的JAX-WSWeb客户端 156
 - 12.2 JAX-WS支持的类型 159
 - 12.3 Web服务的互操作性与JAX-WS 159
 - 12.4 有关JAX-WS的更多信息 159
- 第13章 用JAX-RS构建REST式Web服务 160
 - 13.1 什么是REST式Web服务 160
 - 13.2 创建一个REST式根资源类 161
 - 13.2.1 用JAX-RS开发REST式Web服务 161
 - 13.2.2 JAX-RS应用概述 162
 - 13.2.3 @Path注解和URI路径模板 163
 - 13.2.4 响应HTTP资源 165
 - 13.2.5 使用@Consumes和@Produces定制请求和应答 167
 - 13.2.6 从请求里提取参数 169

<<Java EE 6权威指南>>

- 13.3 JAX-RS的示例应用 172
 - 13.3.1 REST式Web服务 172
 - 13.3.2 rsvp示例应用 174
 - 13.3.3 真实示例 176
- 13.4 有关JAX-RS的更多信息 176
- 第四部分 企业bean
- 第14章 企业bean 180
 - 14.1 什么是企业bean 180
 - 14.1.1 使用企业bean的好处 180
 - 14.1.2 何时使用企业bean 181
 - 14.1.3 企业bean的类型 181
 - 14.2 什么是会话bean 181
 - 14.2.1 会话bean的类型 181
 - 14.2.2 何时使用会话bean 182
 - 14.3 什么是消息驱动bean 183
 - 14.3.1 消息驱动bean与会话bean的区别 183
 - 14.3.2 何时使用消息驱动bean 184
 - 14.4 访问企业bean 184
 - 14.4.1 在客户端中使用企业bean 185
 - 14.4.2 远程还是本地访问 185
 - 14.4.3 本地客户端 186
 - 14.4.4 远程客户端 187
 - 14.4.5 Web服务客户端 188
 - 14.4.6 方法的参数和方法的访问 189
 - 14.5 企业bean的内容 189
 - 14.5.1 在EJB的JAR模块中打包企业bean 189
 - 14.5.2 在WAR模块中打包企业bean 190
 - 14.6 企业bean的命名规范 191
 - 14.7 企业bean的生命周期 191
 - 14.7.1 有状态会话bean的生命周期 191
 - 14.7.2 无状态会话bean的生命周期 192
 - 14.7.3 单件会话bean的生命周期 193
 - 14.7.4 消息驱动bean的生命周期 193
 - 14.8 有关企业bean的更多信息 194
- 第15章 企业bean应用初步 195
 - 15.1 创建企业bean 195
 - 15.1.1 编写企业bean 的类 195
 - 15.1.2 创建converter的Web客户端 196
 - 15.1.3 构建、打包、部署及运行converter示例 197
 - 15.2 修改Java EE应用 198
- 第16章 运行企业bean示例 200
 - 16.1 cart示例 200
 - 16.1.1 业务接口 201
 - 16.1.2 会话bean类 201
 - 16.1.3 @Remove方法 204
 - 16.1.4 辅助类 204
 - 16.1.5 构建、打包、部署及运行cart示例 204

<<Java EE 6权威指南>>

- 16.2 单件会话bean示例counter 206
 - 16.2.1 创建单件会话bean 206
 - 16.2.2 counter示例的架构 209
 - 16.2.3 构建、打包、部署及运行counter示例 211
 - 16.3 Web服务示例helloservice 212
 - 16.3.1 Web服务端点的实现类 212
 - 16.3.2 无状态会话bean的实现类 213
 - 16.3.3 构建、打包、部署及测试helloservice示例 213
 - 16.4 使用定时器服务 214
 - 16.4.1 创建日历型定时器表达式 215
 - 16.4.2 可编程定时器 217
 - 16.4.3 自动定时器 218
 - 16.4.4 取消及保存定时器 219
 - 16.4.5 获得定时器的信息 219
 - 16.4.6 事务和定时器 219
 - 16.4.7 timersession示例 219
 - 16.4.8 构建、打包、部署及运行timersession示例 221
 - 16.5 处理异常 222
- 第五部分 Java EE平台的上下文与依赖注入
- 第17章 Java EE平台的上下文与依赖注入入门 226
- 17.1 CDI概述 227
 - 17.2 关于bean 227
 - 17.3 关于托管bean 228
 - 17.4 可注入对象bean 228
 - 17.5 使用限定词 229
 - 17.6 注入bean 230
 - 17.7 使用作用域 230
 - 17.8 为bean设定EL名称 231
 - 17.9 增加存取方法 232
 - 17.10 在Facelets页面中使用托管bean 232
 - 17.11 使用Producer方法注入对象 233
 - 17.12 配置CDI应用 233
 - 17.13 有关CDI的更多信息 234
- 第18章 运行简单的上下文与依赖注入示例 235
- 18.1 CDI示例simplegreeting 235
 - 18.1.1 simplegreeting的源文件 235
 - 18.1.2 Facelets模板和页面 236
 - 18.1.3 配置文件 237
 - 18.1.4 构建、打包、部署及运行CDI示例simplegreeting 237
 - 18.2 CDI示例guessnumber 239
 - 18.2.1 guessnumber的源文件 239
 - 18.2.2 Facelets页面 243
 - 18.2.3 构建、打包、部署及运行CDI示例guessnumber 244
- 第六部分 持久化
- 第19章 Java Persistence API简介 248
- 19.1 实体 248
 - 19.1.1 实体类的需求 248

<<Java EE 6权威指南>>

- 19.1.2 实体类的持久化字段和属性 249
- 19.1.3 实体里的主键 253
- 19.1.4 实体关系的多样性 254
- 19.1.5 实体关系的方向性 255
- 19.1.6 实体里的可嵌入类 256
- 19.2 实体继承 257
 - 19.2.1 抽象实体 257
 - 19.2.2 映射超类 258
 - 19.2.3 非实体超类 258
 - 19.2.4 实体继承映射策略 259
- 19.3 管理实体 261
 - 19.3.1 EntityManager接口 261
 - 19.3.2 持久化单元 264
- 19.4 查询实体 265
- 19.5 有关Persistence的更多信息 266
- 第20章 运行Persistence示例 267
 - 20.1 order应用 267
 - 20.1.1 order示例中的实体关系 267
 - 20.1.2 order应用里的主键 269
 - 20.1.3 映射多个数据库表的实体 272
 - 20.1.4 order应用里的级联操作 272
 - 20.1.5 order应用里的BLOB和CLOB类型 273
 - 20.1.6 order应用里的时间类型 273
 - 20.1.7 管理order应用里的实体 273
 - 20.1.8 构建、打包、部署以及运行order应用 275
 - 20.2 roster应用 276
 - 20.2.1 roster应用里的关系 276
 - 20.2.2 roster应用里的实体继承关系 277
 - 20.2.3 roster里的Criteria查询 278
 - 20.2.4 roster应用里的自动建表特性 280
 - 20.2.5 构建、打包、部署以及运行roster应用 280
 - 20.3 address-book应用 282
 - 20.3.1 address-book应用里的Bean验证约束 282
 - 20.3.2 为address-book应用里的约束指定出错信息 283
 - 20.3.3 验证JSF应用中输入的Contact数据 283
 - 20.3.4 构建、打包、部署以及运行address-book应用 284
- 第21章 Java Persistence查询语言 286
 - 21.1 查询语言术语 286
 - 21.2 使用Java Persistence查询语言创建查询 287
 - 21.2.1 查询里的命名参数 287
 - 21.2.2 查询里的位置参数 288
 - 21.3 查询语言的简要语法 288
 - 21.3.1 选择语句 288
 - 21.3.2 更新和删除语句 289
 - 21.4 查询示例 289
 - 21.4.1 简单查询 289
 - 21.4.2 需定位到相关实体的查询 290

<<Java EE 6权威指南>>

- 21.4.3 使用其他条件表达式的查询 291
- 21.4.4 批量更新和删除 292
- 21.5 查询语言语法全本 293
 - 21.5.1 BNF符号 293
 - 21.5.2 Java Persistence查询语言的BNF语法 293
 - 21.5.3 FROM语句 297
 - 21.5.4 路径表达式 300
 - 21.5.5 WHERE语句 301
 - 21.5.6 SELECT语句 309
 - 21.5.7 ORDER BY 语句 310
 - 21.5.8 GROUP BY和HAVING语句 311
- 第22章 使用Criteria API构造查询 312
 - 22.1 Criteria和Metamodel API概述 312
 - 22.2 使用Metamodel API为实体类建模 313
 - 22.3 使用Criteria API和Metamodel API创建类型安全的基本查询 314
 - 22.3.1 创建Criteria查询 315
 - 22.3.2 查询根 315
 - 22.3.3 使用join查询关联关系 316
 - 22.3.4 Criteria查询中的路径定位 316
 - 22.3.5 过滤Criteria查询结果 317
 - 22.3.6 处理Criteria查询结果 319
 - 22.3.7 查询执行 320
- 第七部分 安全
- 第23章 Java EE平台安全入门 322
 - 23.1 Java EE安全性概述 322
 - 23.1.1 简单的安全应用示例 323
 - 23.1.2 安全机制的特性 325
 - 23.1.3 应用安全的特征 325
 - 23.2 安全机制 326
 - 23.2.1 Java SE安全机制 326
 - 23.2.2 Java EE安全机制 327
 - 23.3 为容器增加安全性 329
 - 23.3.1 用注解为应用增加安全性 329
 - 23.3.2 用部署描述文件为应用增加安全性 329
 - 23.3.3 使用编程式的安全机制 330
 - 23.4 为GlassFish服务器增加安全性 330
 - 23.5 使用域、用户、用户组和角色 330
 - 23.5.1 什么是域、用户、用户组和角色 331
 - 23.5.2 在GlassFish服务器中管理用户和用户组 333
 - 23.5.3 设置安全角色 334
 - 23.5.4 将角色映射至用户和用户组 335
 - 23.6 使用SSL建立安全连接 336
 - 23.6.1 验证及配置SSL 337
 - 23.6.2 使用数字证书 337
 - 23.7 有关安全性的更多信息 339
- 第24章 Web应用安全化入门 341
 - 24.1 Web应用安全性概述 341

<<Java EE 6权威指南>>

- 24.2 为Web应用增加安全性 342
 - 24.2.1 设定安全限制 343
 - 24.2.2 设定认证机制 345
 - 24.2.3 声明安全角色 350
- 24.3 在Web应用中使用程式安全机制 351
 - 24.3.1 以编程方式实现用户认证 351
 - 24.3.2 以编程方式检查发起者身份 353
 - 24.3.3 编程安全性的代码示例 353
 - 24.3.4 声明并关联角色引用 354
- 24.4 为Web应用增加安全性的示例 355
 - 24.4.1 在servlet中使用基本认证的示例 356
 - 24.4.2 在JSF中使用表单认证机制的示例 359
- 第25章 企业应用安全化入门 363
 - 25.1 为企业bean增加安全性 363
 - 25.1.1 使用声明方式为企业bean增加安全性 365
 - 25.1.2 使用编程方式为企业bean增加安全性 369
 - 25.1.3 用于身份传播的安全标识(run-as) 370
 - 25.1.4 部署经过安全加固的企业bean 371
 - 25.2 为企业bean增加安全性的一组示例 371
 - 25.2.1 使用声明方式为企业bean增加安全性的示例 371
 - 25.2.2 使用编程方式为企业bean增加安全性的示例 375
 - 25.3 为应用客户端增加安全性 377
 - 25.3.1 使用登录模块 377
 - 25.3.2 使用程式用户登录 378
 - 25.4 为企业信息系统应用增加安全性 378
 - 25.4.1 容器管理登录 378
 - 25.4.2 组件管理登录 379
 - 25.4.3 配置资源适配器安全性 379
- 第八部分 Java EE支持技术
- 第26章 Java EE支持技术简介 384
 - 26.1 事务 384
 - 26.2 资源 384
 - 26.2.1 Java EE连接器架构和资源适配器 384
 - 26.2.2 Java Message Service API 385
 - 26.2.3 Java数据库连接软件 385
- 第27章 事务 386
 - 27.1 什么是事务 386
 - 27.2 容器托管的事务 387
 - 27.2.1 事务属性 387
 - 27.2.2 回滚容器托管的事务 390
 - 27.2.3 同步会话bean的实例变量 390
 - 27.2.4 容器托管事务里不允许使用的方法 391
 - 27.3 bean托管事务 391
 - 27.3.1 JTA事务 392
 - 27.3.2 不提交的返回 392
 - 27.3.3 bean托管事务里不允许使用的方法 392
 - 27.4 事务超时 392

<<Java EE 6权威指南>>

- 27.5 更新多个数据库中的数据 393
- 27.6 Web组件里的事务 394
- 27.7 有关事务的更多信息 394
- 第28章 资源连接 395
- 28.1 资源和JNDI命名 395
- 28.2 DataSource对象和连接池 396
- 28.3 资源注入 397
- 28.3.1 字段级别的注入方式 398
- 28.3.2 方法级别的注入方式 398
- 28.3.3 类级别的注入方式 399
- 28.4 资源适配器和契约 399
- 28.4.1 管理契约 400
- 28.4.2 通用工作上下文契约 401
- 28.4.3 外向型和内向型契约 401
- 28.5 元数据注解 402
- 28.6 公共客户端接口 404
- 28.7 参考资源 404

章节摘录

版权页：插图：还需要使用部署描述文件进行安全设定。

认证机制必须在服务器上进行配置才能生效。

基本认证机制是GlassFish服务器的默认认证方法。

本书将介绍如何通过为企业应用的业务方法增加访问控制注解，来实现基于用户名和密码的认证机制。

为了简化部署人员的工作，应用开发人员可以定义安全角色。

安全角色是一组权限的组合，特定类型的用户必须具备这些权限才可以访问应用。

例如，在一个工资单管理系统中，某些用户可查看自己的工资信息（即员工），某些用户可以查看其他用户的工资信息（如经理），某些用户可以修改其他用户的工资信息（如人事部门）。

应用开发人员将负责确定应用的潜在用户，以及哪些用户可以访问哪些方法，即用户与方法访问的对应关系。

于是，应用开发人员为企业bean的类或方法增加注解，声明授权访问方法的用户类型。

关于如何通过注解的方式声明授权用户，将在25.1.1节中的“通过角色声明指定授权用户”进行介绍。

如果使用某种注解定义方法的访问权限，则部署系统在运行并收到请求时，将自动触发用户名和密码的验证。

在这种认证机制中，用户被提示输入用户名和密码。

这些数据将被传送至服务器端，与用户数据库中的已知用户信息进行比较。

如果找到用户且密码匹配，则服务器将比较用户的角色与被授权访问该方法的角色。

如果用户认证通过，且具备可访问该方法的角色，则数据将返回给用户。

25.1.1节将详细讨论如何实现声明式安全性。

程式安全性对于企业bean，嵌入至业务方法中的代码片段将以编程的方式访问调用者的用户信息，并使用该信息进行进一步的判定。

当声明式的安全机制不能完全满足应用的特定安全需求时，程式安全机制是一种重要的补充手段。

通常来说，应用的安全策略应该对企业bean中的业务方法透明，且由容器强制执行。

本章介绍的程式安全性API并不常用。

当企业bean的业务方法需要访问有关安全的上下文信息时，如按照时段设定访问权限，或为某个角色增加其他重要检查时，才会用到这些API。

25.1.2节将详细讨论程式安全性。

本章中讲述的某些内容，需要读者首先阅读并掌握第14章、第15章及第23章的内容。

如前所述，企业bean运行在EJB容器中，对于GlassFish服务器来说，EJB容器是一个应用运行时环境。

如图25—1所示。

本节将讨论如何为Java EE应用增加安全性。

Java EE应用通常由一个或多个模块组成，如EJBJAR文件，并打包至EAR文件中（保存应用的归档文件）。

安全注解用在Java的类文件中，设定授权用户，以及声明使用基本认证机制（基于用户名和密码）。

企业bean通常为Web应用提供业务逻辑。

在这种场景中，将企业bean打包至Web应用的WAR模块中，会简化应用的部署过程和组织方式。

企业bean可以以Java类文件的形式打包至wAR模块中，或先打包成JAR文件，再集成至WAR模块中。

<<Java EE 6权威指南>>

编辑推荐

Oracle Java EE 6文档团队成员精准解说Java EE 6平台以任务为导向，通过示例阐释企业应用开发最新API和平台特性详解

<<Java EE 6权威指南>>

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>