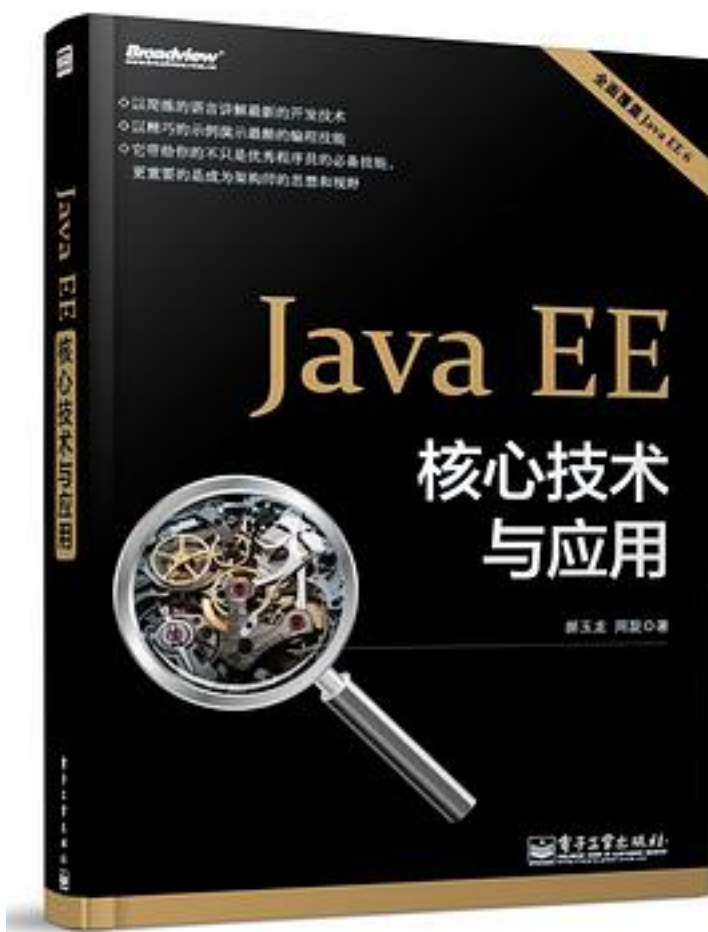


Java EE 核心技术与应用



[Java EE 核心技术与应用_下载链接1](#)

著者:郝玉龙

出版者:电子工业出版社

出版时间:2013-5

装帧:平装

isbn:9787121201752

本书基于最新的Java EE 6规范对Java EE应用开发技术进行系统讲解。书中主要包括四部分内容：第一部分介绍了Java

EE的定义、设计思想、技术架构和开发模式等，可使读者全面认识Java EE。第二部分以Java EE企业应用的表现层、数据持久化层和业务逻辑层的开发为主线，重点讲解Java EE 6规范的最新功能特性，包括JSF 2.0、Servlet3.0、EJB 3.1和JPA 2.0等，使读者掌握开发完整Java EE企业应用的基本技能。第三部分讲解了如何利用CDI实现组件的低耦合、如何利用Bean Validation框架实现统一校验、如何确保企业应用安全等高级知识，使读者掌握企业应用开发中的系统架构、安全防护等高级技能。第四部分讲解了如何利用JavaMail、Web服务、JMS消息服务和消息驱动Bean等技术来实现企业应用之间的集成和交互。

本书可作为高等学校计算机专业教材，也可作为相关人员的参考书。本书每一章都是一个完整独立的部分，因此教师在授课时可根据授课重点及课时数量进行灵活调整。

作者介绍:

目录: 第1章 走进Java EE	1
1.1 引言	1
1.2 为什么需要Java EE	1
1.2.1 企业级应用特征	1
1.2.2 企业级应用架构体系	2
1.3 什么是Java EE	3
1.4 Java EE设计思想	4
1.4.1 容器	4
1.4.2 组件	5
1.4.3 容器与组件的交互	5
1.5 Java EE 技术架构	5
1.5.1 组件技术	5
1.5.2 服务技术	7
1.5.3 通信技术	8
1.5.4 框架技术	9
1.6 Java EE核心开发模式	9
1.7 Java EE优点	10
1.8 小结	11
第2章 搭建开发环境	12
2.1 引言	12
2.2 安装JDK	12
2.3 安装NetBeans IDE	13
2.4 测试开发环境	14
2.5 小结	16
第3章 基于JSF构建表示逻辑层	17
3.1 引言	17
3.2 什么是JSF	17
3.2.1 什么是框架	17
3.2.2 为什么需要框架	17
3.2.3 JSF是什么样的框架	18
3.2.4 为什么学习JSF	19
3.3 第一个JSF应用	19
3.3.1 创建JSF项目	19
3.3.2 模型组件	21
3.3.3 视图组件	22
3.3.4 控制组件	22

- 3.3.5 运行演示 23
- 3.3.6 总结思考 23
- 3.4 利用JSF组件构建视图 24
 - 3.4.1 JSF标记库 24
 - 3.4.2 HTML标记 25
 - 3.4.3 Core标记 39
 - 3.4.4 使用JSTL标记 40
- 3.5 在视图中访问Web资源 41
- 3.6 利用页面模板提高视图可维护性 43
 - 3.6.1 布局 43
 - 3.6.2 装饰 48
- 3.7 利用Managed Bean封装业务逻辑 51
 - 3.7.1 定义Managed Bean 51
 - 3.7.2 生命周期范围 53
 - 3.7.3 Bean之间的依赖 56
 - 3.7.4 生命周期回调方法 58
- 3.8 使用EL访问服务器端数据 59
 - 3.8.1 范围 59
 - 3.8.2 访问对象 59
 - 3.8.3 值表达式和方法表达式 60
 - 3.8.4 延迟计算 61
- 3.9 实现灵活的导航控制 61
 - 3.9.1 视图ID 61
 - 3.9.2 利用Post请求实现导航 62
 - 3.9.3 导航约定 66
 - 3.9.4 导航规则 67
 - 3.9.5 重定向 69
 - 3.9.6 利用Get请求实现导航 71
 - 3.9.7 JSF框架外导航 74
 - 3.9.8 导航中的参数传递 74
 - 3.9.9 导航总结 77
- 3.10 实现国际化支持 77
 - 3.10.1 准备资源包 77
 - 3.10.2 配置资源包 79
 - 3.10.3 在JSF视图中使用资源 79
 - 3.10.4 设置应用程序本地属性 80
- 3.11 使用Ajax获得更好的用户体验 81
- 3.12 小结 83
- 第4章 扩展JSF 84
 - 4.1 引言 84
 - 4.2 JSF请求处理过程 84
 - 4.2.1 常规流程 84
 - 4.2.2 示例分析 85
 - 4.2.3 特殊流程 88
 - 4.2.4 异常处理 89
 - 4.2.5 总结思考 92
 - 4.3 利用监听器实现事件处理 93
 - 4.3.1 Value Change事件 93
 - 4.3.2 Action事件 96
 - 4.3.3 Phase事件 98
 - 4.3.4 System事件 99
 - 4.4 自定义类型转换 101
 - 4.4.1 标准转换器 102
 - 4.4.2 自定义转换器 102

4.5 自定义输入校验	105
4.5.1 标准校验器	105
4.5.2 自定义校验器	106
4.5.3 Bean方法校验	107
4.5.4 异常信息本地化	108
4.6 自定义复合组件	108
4.6.1 复合组件标记库	108
4.6.2 定制简单的复合组件	109
4.6.3 开发复杂的复合组件	111
4.7 自定义非UI组件	116
4.8 自定义UI组件	119
4.8.1 创建一个简单的UI组件	119
4.8.2 利用属性控制自定义组件行为	121
4.8.3 使用单独的渲染器	123
4.8.4 获取用户输入信息	125
4.8.5 保存组件状态	127
4.9 使用第三方组件	129
4.10 小结	130
第5章 利用Servlet处理复杂Web请求	131
5.1 引言	131
5.2 Servlet基础	131
5.2.1 什么是Servlet	131
5.2.2 Servlet工作流程	131
5.2.3 Servlet API	132
5.3 第一个Servlet	133
5.4 处理请求	138
5.5 生成响应	141
5.6 在JSF应用中处理非JSF请求	142
5.7 支撑自定义JSF组件	146
5.8 利用Filter过滤请求	149
5.9 小结	152
第6章 利用JPA访问企业信息	153
6.1 引言	153
6.2 数据库驱动与JDBC	153
6.3 连接池和数据源	154
6.3.1 基本概念	154
6.3.2 创建MySQL连接池	156
6.3.3 创建数据源	157
6.4 第一个JPA应用	158
6.4.1 持久化单元	158
6.4.2 Entity	159
6.4.3 EntityManager	161
6.4.4 运行演示	162
6.5 ORM	163
6.5.1 Entity	163
6.5.2 主键	164
6.5.3 复合主键	164
6.5.4 属性	167
6.5.5 关联映射	170
6.5.6 加载方式	173
6.5.7 顺序	174
6.5.8 继承映射	174
6.6 Entity管理	175
6.6.1 获取EntityManager	175

- 6.6.2 持久化上下文 176
- 6.6.3 Entity操作 176
- 6.6.4 级联操作 183
- 6.7 JPQL 184
 - 6.7.1 动态查询 185
 - 6.7.2 参数设置 186
 - 6.7.3 命名查询 186
 - 6.7.4 属性查询 187
 - 6.7.5 使用构造器 187
- 6.8 基于Criteria API的安全查询 188
- 6.9 缓存 190
- 6.10 并发控制 192
- 6.11 生命周期回调方法 194
- 6.12 小结 195
- 第7章 使用会话Bean实现业务逻辑 196
 - 7.1 引言 196
 - 7.2 EJB基础 196
 - 7.2.1 为什么需要EJB 196
 - 7.2.2 EJB容器 197
 - 7.2.3 EJB组件 198
 - 7.2.4 EJB接口 199
 - 7.2.5 EJB分类 199
 - 7.2.6 部署EJB 200
 - 7.2.7 EJB优点 200
 - 7.3 无状态会话Bean 200
 - 7.3.1 什么是无状态会话Bean 201
 - 7.3.2 开发一个无状态会话Bean 202
 - 7.3.3 利用Servlet测试无状态会话Bean 205
 - 7.3.4 利用远程客户端测试无状态会话Bean 207
 - 7.4 有状态会话Bean 208
 - 7.4.1 基本原理 208
 - 7.4.2 实现有状态会话Bean 209
 - 7.5 单例会话Bean 213
 - 7.5.1 基本原理 213
 - 7.5.2 利用JSF访问单例会话Bean 213
 - 7.5.3 并发控制 215
 - 7.5.4 依赖管理 216
 - 7.6 Time服务 217
 - 7.7 拦截器 219
 - 7.8 事务支持 221
 - 7.9 异步方法 222
 - 7.10 小结 227
- 第8章 利用CDI实现组件间低耦合 228
 - 8.1 引言 228
 - 8.2 CDI概述 228
 - 8.3 CDI下的受控Bean 228
 - 8.4 Bean的生命周期范围 232
 - 8.5 使用限定符注入动态类型 237
 - 8.6 使用替代组件实现部署时动态注入 239
 - 8.7 使用生产方法注入动态内容 241
 - 8.8 使用拦截器绑定类型注入功能服务 243
 - 8.9 利用构造型封装注入操作 246
 - 8.10 小结 247
- 第9章 使用Bean Validation校验数据 248

9.1 引言	248
9.2 Bean Validation概述	248
9.3 使用默认约束器	248
9.4 实现自定义约束器	251
9.5 约束的传递	254
9.5.1 继承	254
9.5.2 级联	255
9.6 小结	256
第10章 确保企业应用安全	257
10.1 引言	257
10.2 认证	257
10.2.1 配置文件安全域	257
10.2.2 配置JDBC安全域	258
10.2.3 声明认证配置	260
10.3 授权	260
10.3.1 授权声明	260
10.3.2 角色映射	261
10.4 测试Java EE容器的安全服务	262
10.5 定制FORM方式认证界面	262
10.6 在代码中获取用户身份信息	264
10.7 EJB安全控制	265
10.8 小结	267
第11章 为应用添加邮件发送功能	268
11.1 引言	268
11.2 JavaMail基础	268
11.2.1 JavaMail体系	268
11.2.2 JavaMail规范组成	269
11.3 配置JavaMail会话	270
11.4 发送邮件	270
11.5 发送带附件的邮件	272
11.6 小结	275
第12章 利用Web服务集成应用	276
12.1 引言	276
12.2 Web服务概述	276
12.2.1 什么是Web服务	276
12.2.2 Web服务技术体系	276
12.2.3 Web服务工作模型	278
12.3 Java EE平台下的Web服务实现	278
12.4 开发Web服务实例	278
12.4.1 创建Web服务组件	279
12.4.2 为Web服务组件添加业务逻辑	280
12.4.3 部署Web服务	280
12.4.4 测试Web服务	280
12.5 调用Web服务	282
12.5.1 添加Web服务客户端	282
12.5.2 调用Web服务	283
12.6 将会话Bean发布为Web 服务	285
12.7 RESTful Web服务	286
12.7.1 什么是REST	286
12.7.2 利用JAX-RS开发RESTful Web服务	287
12.7.3 与SOAP对比	290
12.8 Web服务的优缺点	291
12.9 小结	292
第13章 利用消息服务实现应用间异步交互	293

- 13.1 引言 293
- 13.2 JMS概述 293
 - 13.2.1 JMS消息模型 293
 - 13.2.2 JMS消息服务接口 293
 - 13.2.3 消息传递模式 294
- 13.3 配置消息服务资源和连接工厂 294
- 13.4 发送JMS消息 295
- 13.5 利用MDB处理消息 296
- 13.6 小结 298
- • • • • (收起)

[Java EE 核心技术与应用_下载链接1](#)

标签

Java

EE

编程

核心技术与应用

评论

一天过的入门读物

[Java EE 核心技术与应用_下载链接1](#)

书评

[Java EE 核心技术与应用_下载链接1](#)