

《Java Web高级编程》

图书基本信息

书名：《Java Web高级编程》

13位ISBN编号：9787302400954

出版时间：2015-6-1

作者：威廉斯 (Nicholas S.Williams)

页数：797

译者：王肖锋

版权说明：本站所提供下载的PDF图书仅提供预览和简介以及在线试读，请支持正版图书。

更多资源请访问：www.tushu000.com

《Java Web高级编程》

内容概要

Java成为世界上编程语言之一是有其优势的。熟悉JavaSE的程序员可以轻松地进入到Java EE开发中，构建出安全、可靠和具有扩展性的企业级应用程序。编写《Java Web高级编程——涵盖WebSockets

、Spring Framework、JPA Hibernate和Spring Security》一书的目的正是如此。

《Java Web高级编程：涵盖WebSockets、Spring Framework、JPA》面向的读者是已经了解Java SE、SQL和基本的HTML，准备将他们的Java编码技能提升到更高水平的程序员。软件开发者可以按顺序阅读本书或者在遇到特定的编程问题时将某个章节用作参考。

主要内容

为企业级Web应用的Java编码提供了自我指导、自我学习的方法

帮助Web应用架构师在开发团队的项目中或者开发过程中应用新的概念

对Java EE平台7和其中的许多技术提供了详细的介绍

讲解了Servlet、JSP、WebSockets、Spring Framework、AMQP、JPA和O/RM、Spring Data、全文搜索

、Apache Lucene和Hibernate Search、Spring Security和OAuth

重点介绍Java SE 8中新增加的常用功能，如lambda表达式和新的JSR 310 Java 8 Date and Time API等

作者简介

Nicholas S. Williams是Java和相关技术领域的一位著名专家。在2010年，他被授予中部田纳西州年度软件工程师的称号。Nick大量参与了开源社区工作，为Apache Log4J、Apache Tomcat、Jackson Mapper、Spring Framework和Spring Security等项目贡献了bug修复、新特性和文档。

书籍目录

第 部分 创建企业级应用程序

第1章 介绍Java EE平台 3

1.1 Java平台时间线 3

1.1.1 起始 3

1.1.2 企业级Java的诞生 4

1.1.3 Java SE和Java EE共同发展 5

1.1.4 了解最新的平台特性 7

1.1.5 持续发展 11

1.2 了解基本的Web应用程序结构 11

1.2.1 Servlet、过滤器、监听器和JSP 11

1.2.2 目录结构和WAR文件 12

1.2.3 部署描述符 13

1.2.4 类加载器架构 14

1.2.5 企业级应用程序归档文件 14

1.3 小结 15

第2章 使用Web容器 17

2.1 选择Web容器 17

2.1.1 Apache Tomcat 18

2.1.2 GlassFish 19

2.1.3 JBoss和WildFly 19

2.1.4 其他容器和应用服务器 20

2.1.5 本书使用Tomcat的原因 20

2.2 在个人计算机中安装Tomcat 21

2.2.1 将Tomcat安装为Windows服务 21

2.2.2 将Tomcat安装为命令行应用程序 21

2.2.3 配置自定义的JSP编译器 23

2.3 在Tomcat中部署和卸载应用程序 24

2.3.1 手动部署和卸载 24

2.3.2 使用Tomcat管理器 25

2.4 通过IDE调试Tomcat 26

2.4.1 使用IntelliJ IDEA 27

2.4.2 使用Eclipse 30

2.5 小结 34

第3章 创建第一个Servlet 35

3.1 创建Servlet类 36

3.1.1 选择要继承的Servlet类 36

3.1.2 使用初始化方法和销毁方法 38

3.2 配置可部署的Servlet 39

3.2.1 向描述符中添加Servlet 39

3.2.2 将Servlet映射到URL 40

3.2.3 运行和调试Servlet 42

3.3 了解doGet、doPost和其他方法 44

3.3.1 在service方法执行的过程中 44

3.3.2 使用HttpServletRequest 44

3.3.3 使用HttpServletResponse 47

3.4 使用参数和接受表单提交 49

3.5 使用初始化参数配置应用程序 53

- 3.5.1 使用上下文初始化参数 53
- 3.5.2 使用Servlet初始化参数 54
- 3.6 通过表单上传文件 56
 - 3.6.1 介绍客户支持项目 56
 - 3.6.2 配置Servlet支持文件上传 56
 - 3.6.3 接受文件上传 59
- 3.7 编写多线程安全的应用程序 60
 - 3.7.1 理解请求、线程和方法执行 60
 - 3.7.2 保护共享资源 61
- 3.8 小结 62
- 第4章 使用JSP显示页面内容 63
 - 4.1 使用替代output.println(" ") 64
 - 4.1.1 使用JSP的原因 65
 - 4.1.2 JSP在运行时的处理 66
 - 4.2 创建第一个JSP 67
 - 4.2.1 了解文件结构 67
 - 4.2.2 指令、声明、脚本和表达式 69
 - 4.2.3 注释代码 70
 - 4.2.4 在JSP中导入类 71
 - 4.2.5 使用指令 72
 - 4.2.6 使用标签 75
 - 4.3 在JSP中使用Java(以及不鼓励使用Java的原因) 76
 - 4.3.1 使用JSP中隐式的变量 76
 - 4.3.2 不应该在JSP中使用Java的原因 82
 - 4.4 结合使用Servlet和JSP 82
 - 4.4.1 配置部署描述符中的JSP属性 82
 - 4.4.2 将Servlet中的请求转发给JSP 85
 - 4.5 关于JSP文档(JSPX)的注意事项 89
 - 4.6 小结 91
- 第5章 使用会话维持状态 93
 - 5.1 需要会话的原因 93
 - 5.1.1 维持状态 94
 - 5.1.2 记住用户 94
 - 5.1.3 启动应用程序工作流 94
 - 5.2 使用会话cookie和URL重写 95
 - 5.2.1 了解会话cookie 96
 - 5.2.2 URL中的会话ID 98
 - 5.2.3 会话的漏洞 100
 - 5.3 在会话中存储数据 102
 - 5.3.1 在部署描述符中配置会话 103
 - 5.3.2 存储和获取数据 105
 - 5.3.3 删除数据 109
 - 5.3.4 在会话中存储更复杂的数据 111
 - 5.4 使用会话 114
 - 5.4.1 为客户支持应用程序添加登录功能 114
 - 5.4.2 使用监听器检测会话的变化 118
 - 5.4.3 维护活跃会话列表 120

- 5.5 将使用会话的应用程序群集化 124
 - 5.5.1 在群集中使用会话ID 124
 - 5.5.2 了解会话复制和故障恢复 126
- 5.6 小结 127
- 第6章 在JSP中使用表达式语言 129
 - 6.1 了解表达式语言 129
 - 6.1.1 表达式语言的用途 130
 - 6.1.2 了解基本语法 130
 - 6.1.3 添加EL表达式 132
 - 6.2 使用EL语法 133
 - 6.2.1 保留关键字 133
 - 6.2.2 操作符优先级 134
 - 6.2.3 对象属性和方法 139
 - 6.2.4 EL函数 140
 - 6.2.5 静态字段和方法访问 141
 - 6.2.6 枚举 142
 - 6.2.7 lambda表达式 142
 - 6.2.8 集合 143
 - 6.3 在EL表达式中使用作用域变量 144
 - 6.3.1 使用隐式的EL作用域 145
 - 6.3.2 使用隐式的EL变量 148
 - 6.4 使用流API访问集合 151
 - 6.4.1 了解中间操作 151
 - 6.4.2 使用终结操作 153
 - 6.4.3 使用流API 154
 - 6.5 使用表达式语言替换Java代码 155
 - 6.6 小结 157
- 第7章 使用Java标准标签库 159
 - 7.1 JSP标签和JSTL简介 159
 - 7.2 使用核心标签库(C命名空间) 163
 - 7.2.1 163
 - 7.2.2 164
 - 7.2.3 165
 - 7.2.4、和 166
 - 7.2.5 167
 - 7.2.6 168
 - 7.2.7 168
 - 7.2.8 169
 - 7.2.9 和 170
 - 7.2.10 使用核心库标签 170
 - 7.3 使用国际化和格式化标签库(FMT命名空间) 173
 - 7.3.1 国际化和本地化组件 173
 - 7.3.2 175
 - 7.3.3 176
 - 7.3.4 和 176
 - 7.3.5 177
 - 7.3.6 和 177
 - 7.3.7 和 178
 - 7.3.8 和 179

- 7.3.9 使用i18n和格式化库标签 180
- 7.4 使用数据库访问标签库(SQL命名空间) 182
- 7.5 使用XML处理标签库(X命名空间) 184
- 7.6 使用JSP标签替换Java代码 184
- 7.7 小结 186
- 第8章 编写自定义标签和函数库 189
 - 8.1 了解TLD、标签文件和标签处理器 189
 - 8.1.1 读取Java标准标签库TLD 190
 - 8.1.2 比较JSP指令和标签文件指令 196
 - 8.2 创建标签文件用作HTML模板 198
 - 8.3 创建日期格式化标签处理器 199
 - 8.4 创建EL函数简写字符串 203
 - 8.5 使用自定义JSP标签替换Java代码 204
 - 8.6 小结 210
- 第9章 使用过滤器改进应用程序 211
 - 9.1 了解过滤器的目的 211
 - 9.1.1 日志过滤器 212
 - 9.1.2 验证过滤器 212
 - 9.1.3 压缩和加密过滤器 212
 - 9.1.4 错误处理过滤器 212
 - 9.2 创建、声明和映射过滤器 213
 - 9.2.1 了解过滤器链 213
 - 9.2.2 映射到URL模式和Servlet名称 213
 - 9.2.3 映射到不同的请求派发器类型 214
 - 9.2.4 使用部署描述符 214
 - 9.2.5 使用注解 215
 - 9.2.6 使用编程式配置 215
 - 9.3 过滤器排序 216
 - 9.3.1 URL模式映射和Servlet名称映射 217
 - 9.3.2 演示过滤器顺序 218
 - 9.3.3 使用过滤器处理异步请求 220
 - 9.4 调查过滤器的实际用例 224
 - 9.4.1 添加简单的日志过滤器 225
 - 9.4.2 使用过滤器压缩响应内容 226
 - 9.5 使用过滤器简化认证 230
 - 9.6 小结 231
- 第10章 在应用程序中使用WebSocket进行交互 233
 - 10.1 演变：从AJAX到WEBSOCKET 234
 - 10.1.1 问题：从服务器获得新数据到浏览器 234
 - 10.1.2 解决方案1：频繁轮询 235
 - 10.1.3 解决方案2：长轮询 236
 - 10.1.4 解决方案3：分块编码 237
 - 10.1.5 解决方案4：Applet和Adobe Flash 238
 - 10.1.6 WebSocket：一种无人知道但已经存在的解决方案 239
 - 10.2 了解WebSocket API 242
 - 10.2.1 HTML5(JavaScript)客户端API 243
 - 10.2.2 Java WebSocket API 245
 - 10.3 使用WebSocket创建多人游戏 247
 - 10.3.1 实现基本的三连棋游戏策略 247

- 10.3.2 创建服务器终端 248
- 10.3.3 编写JavaScript游戏控制台 252
- 10.3.4 WebSocket三连棋游戏试玩 256
- 10.4 在群集中使用WebSocket进行通信 257
 - 10.4.1 使用两个Servlet实例模拟简单的群集 257
 - 10.4.2 发送和接收二进制消息 259
 - 10.4.3 测试模拟群集应用程序 261
- 10.5 在客户支持应用程序中添加"支持与客户聊天"功能 262
 - 10.5.1 使用编码器和解码器转换消息 262
 - 10.5.2 创建聊天服务器终端 264
 - 10.5.3 编写JavaScript聊天应用程序 267
- 10.6 小结 269
- 第11章 使用日志监控应用程序 271
 - 11.1 了解日志的概念 272
 - 11.1.1 记录日志的原因 272
 - 11.1.2 在日志中记录的内容 273
 - 11.1.3 日志的写入方式 274
 - 11.2 使用日志级别和分类 276
 - 11.2.1 使用不同日志级别的原因 276
 - 11.2.2 定义的日志级别 276
 - 11.2.3 日志分类的工作方式 277
 - 11.2.4 筛选的工作方式 277
 - 11.3 选择日志框架 277
 - 11.3.1 API和实现 278
 - 11.3.2 性能 278
 - 11.3.3 Apache Commons Logging和SLF4J 279
 - 11.3.4 Log4j 2简介 280
 - 11.4 在应用程序中集成日志 283
 - 11.4.1 创建Log4j 2配置文件 284
 - 11.4.2 在Web过滤器中使用鱼标签 287
 - 11.4.3 在Java代码中编写日志语句 288
 - 11.4.4 在JSP中使用日志标签库 290
 - 11.4.5 客户支持应用程序中的日志 290
 - 11.5 小结 291
- 第 部分 添加Spring Framework
- 第12章 介绍Spring Framework 295
 - 12.1 Spring Framework简介 296
 - 12.1.1 反转控制和依赖注入 296
 - 12.1.2 面向切面编程 297
 - 12.1.3 数据访问和事务管理 297
 - 12.1.4 应用程序消息 297
 - 12.1.5 Web应用程序的模型-视图-控制器模式 298
 - 12.2 使用Spring Framework的原因 298
 - 12.2.1 逻辑代码分组 298
 - 12.2.2 使用同一代码库的多个用户界面 298
 - 12.3 了解应用上下文 299
 - 12.4 启动Spring Framework 300
 - 12.4.1 使用部署描述符启动Spring 301
 - 12.4.2 在初始化器中使用编程的方式启动Spring 303

- 12.5 配置Spring Framework 306
 - 12.5.1 创建XML配置 308
 - 12.5.2 创建混合配置 310
 - 12.5.3 使用@Configuration配置Spring 314
- 12.6 使用bean definition profile 318
 - 12.6.1 了解profile的工作原理 319
 - 12.6.2 考虑反模式和安全问题 321
- 12.7 小结 322
- 第13章 使用控制器替代Servlet 323
 - 13.1 了解@RequestMapping 323
 - 13.1.1 使用@RequestMapping特性缩小请求匹配的范围 324
 - 13.1.2 指定控制器方法参数 328
 - 13.1.3 为控制器方法选择有效的返回类型 335
 - 13.2 使用Spring Framework的模型和视图模式 337
 - 13.2.1 使用显式的视图和视图名称 338
 - 13.2.2 使用含有模型特性的隐式视图 340
 - 13.2.3 返回响应实体 341
 - 13.3 使用表单对象简化开发 346
 - 13.3.1 在模型中添加表单对象 347
 - 13.3.2 使用Spring Framework 标签 347
 - 13.3.3 获得被提交的表单数据 349
 - 13.4 更新客户支持应用程序 350
 - 13.4.1 启用Multipart支持 350
 - 13.4.2 将Servlet转换成Spring MVC控制器 351
 - 13.4.3 创建自定义下载视图 352
 - 13.5 小结 353
- 第14章 使用服务和仓库支持控制器 355
 - 14.1 了解模型-视图-控制器模式与控制器-服务-仓库模式 355
 - 14.1.1 识别程序逻辑的不同类型 356
 - 14.1.2 使用仓库提供持久化逻辑 357
 - 14.1.3 使用服务提供业务逻辑 357
 - 14.1.4 使用控制器提供用户界面逻辑 358
 - 14.2 使用根应用上下文替代Web应用上下文 359
 - 14.2.1 在多用户界面中重用根应用上下文 359
 - 14.2.2 将业务逻辑从控制器移动到服务 360
 - 14.2.3 使用仓库存储数据 364
 - 14.3 使用异步和计划执行改进服务 368
 - 14.3.1 了解执行器和调度器 369
 - 14.3.2 配置调度器和异步支持 369
 - 14.3.3 创建和使用@Async方法 371
 - 14.3.4 创建和使用@Scheduled方法 372
 - 14.4 使用WebSocket实现逻辑层分离 373
 - 14.4.1 在Spring应用上下文中添加由容器管理的对象 373
 - 14.4.2 使用Spring WebSocket配置器 375
 - 14.4.3 记住：WebSocket只是业务逻辑的另一个界面 376
 - 14.5 小结 380
- 第15章 使用Spring Framework i18n国际化应用程序 381
 - 15.1 使用Spring Framework i18n的原因 381

- 15.1.1 使国际化变得更容易 382
- 15.1.2 直接本地化错误消息 382
- 15.2 使用基本的国际化和本地化API 382
 - 15.2.1 了解资源包和消息格式 383
 - 15.2.2 使用消息源进行挽救 385
 - 15.2.3 使用消息源国际化JSP 386
- 15.3 在Spring Framework中配置国际化 387
 - 15.3.1 创建消息源 387
 - 15.3.2 了解区域设置解析器 388
 - 15.3.3 使用处理拦截器修改区域设置 390
 - 15.3.4 提供一个用户Profile区域设置 390
 - 15.3.5 包含时区支持 391
 - 15.3.6 了解主题如何改进国际化 392
- 15.4 国际化代码 392
 - 15.4.1 使用标签 393
 - 15.4.2 以更干净的方式处理应用程序错误 395
 - 15.4.3 更新客户支持应用程序 398
 - 15.4.4 直接使用消息源 399
- 15.5 小结 402
- 第16章 使用JSR 349、Spring Framework和Hibernate Validator执行Bean验证 403
 - 16.1 Bean验证的概念 404
 - 16.1.1 使用Hibernate Validator的原因 405
 - 16.1.2 了解注解元数据模型 406
 - 16.1.3 使用Spring Framework实现Bean验证 406
 - 16.2 在Spring Framework容器中配置验证 406
 - 16.2.1 配置Spring验证Bean 407
 - 16.2.2 创建错误代码本地化 409
 - 16.2.3 使用方法验证Bean后处理器 410
 - 16.2.4 在Spring MVC中使用相同的验证Bean 411
 - 16.3 在Bean中添加约束验证注解 411
 - 16.3.1 了解内建的约束注解 411
 - 16.3.2 了解常见的约束特性 412
 - 16.3.3 使用约束 413
 - 16.3.4 使用@Valid实现递归验证 415
 - 16.3.5 使用验证组 416
 - 16.3.6 在编译时检查约束合法性 417
 - 16.4 为方法验证配置Spring Bean 418
 - 16.4.1 标注接口，而非实现 418
 - 16.4.2 在方法参数上使用限制和递归验证 419
 - 16.4.3 验证方法返回值 420
 - 16.4.4 表示一个类是否适用于方法验证 420
 - 16.4.5 在Spring MVC控制器中使用参数验证 421
 - 16.4.6 为用户显示验证错误 423
 - 16.5 编写自己的验证约束 425
 - 16.5.1 在自定义限制中继承其他限制 425
 - 16.5.2 创建限制验证器 426
 - 16.5.3 了解限制验证器的生命周期 428
 - 16.6 在客户支持应用程序中集成验证 429
 - 16.7 小结 431

- 第17章 创建RESTful和SOAP Web服务 433
 - 17.1 了解Web服务 433
 - 17.1.1 最初的SOAP 434
 - 17.1.2 RESTful Web服务提供了一种更简单的方式 435
 - 17.2 在Spring MVC中配置RESTful Web服务 442
 - 17.2.1 使用原型注解分离控制器 442
 - 17.2.2 创建单独的Web和REST应用上下文 443
 - 17.2.3 处理RESTful Web服务中的错误条件 446
 - 17.2.4 将RESTful请求映射到控制器方法 449
 - 17.2.5 使用索引终端改进发现机制 453
 - 17.3 测试Web服务终端 454
 - 17.3.1 选择测试工具 454
 - 17.3.2 请求Web服务 455
 - 17.4 使用Spring Web Service创建SOAP Web服务 457
 - 17.4.1 编写契约优先的XSD和WSDL 458
 - 17.4.2 添加SOAP派发器Servlet配置 460
 - 17.4.3 创建SOAP终端 462
 - 17.5 小结 465
- 第18章 使用消息传送和群集实现灵活性和可靠性 467
 - 18.1 识别需要消息传送和群集的时机 467
 - 18.1.1 应用程序消息传送的定义 468
 - 18.1.2 群集的定义 470
 - 18.1.3 消息传送和群集的协作方式 474
 - 18.2 为应用程序添加消息传送支持 477
 - 18.2.1 创建应用程序事件 477
 - 18.2.2 订阅应用程序事件 478
 - 18.2.3 发布应用程序事件 479
 - 18.3 在群集中分布消息传送 481
 - 18.3.1 更新事件以支持分布 482
 - 18.3.2 创建并配置一个自定义事件多播器 483
 - 18.3.3 使用WebSocket发送和接收事件 485
 - 18.3.4 通过多播数据包发现节点 487
 - 18.3.5 部署多个应用程序模拟群集 489
 - 18.4 使用AMQP分布事件 490
 - 18.4.1 配置AMQP代理 491
 - 18.4.2 创建AMQP多播器 492
 - 18.4.3 运行使用了AMQP的应用程序 494
 - 18.5 小结 495
- 第 部分 使用JPA和Hibernate ORM持久化数据
- 第19章 介绍Java Persistence API和 Hibernate ORM 499
 - 19.1 数据持久化的定义 499
 - 19.1.1 平面文件实体存储 500
 - 19.1.2 结构化文件存储 500
 - 19.1.3 关系数据库系统 501
 - 19.1.4 面向对象数据库 501
 - 19.1.5 无模式数据库系统 502
 - 19.2 对象-关系映射的定义 502
 - 19.2.1 了解持久化实体的问题 503
 - 19.2.2 O/RM使实体持久化更简单 504

- 19.2.3 JPA提供了一种标准O/RM API 505
- 19.3 使用Hibernate ORM的原因 507
- 19.4 Hibernate ORM简介 507
 - 19.4.1 使用Hibernate映射文件 507
 - 19.4.2 了解会话API 509
 - 19.4.3 从SessionFactory中获得会话 511
 - 19.4.4 使用Spring Framework创建SessionFactory 512
- 19.5 准备关系数据库 513
 - 19.5.1 安装MySQL和MySQL Workbench 513
 - 19.5.2 安装MySQL JDBC驱动 515
 - 19.5.3 在Tomcat中创建连接资源 516
 - 19.5.4 注意Maven依赖 517
- 19.6 小结 517
- 第20章 使用JPA注解将实体映射到表 519
 - 20.1 使用简单实体 520
 - 20.1.1 创建实体并将它映射到表 521
 - 20.1.2 指示JPA使用实体字段的方式 523
 - 20.1.3 映射代理键 523
 - 20.1.4 使用基本数据类型 529
 - 20.1.5 指定列名和其他细节 532
 - 20.2 创建和使用持久化单元 534
 - 20.2.1 设计数据库表 534
 - 20.2.2 了解持久化单元作用域 536
 - 20.2.3 创建持久化配置 536
 - 20.2.4 使用持久化API 539
 - 20.3 映射复杂数据类型 543
 - 20.3.1 使用枚举作为实体属性 543
 - 20.3.2 了解JPA如何处理日期和时间 544
 - 20.3.3 将大属性映射为CLOB和BLOB 546
 - 20.4 小结 548
- 第21章 在Spring Framework仓库中使用JPA 549
 - 21.1 使用Spring仓库和事务 550
 - 21.1.1 了解事务范围 550
 - 21.1.2 为事务和实体管理器使用线程 551
 - 21.1.3 使用异常转换 552
 - 21.2 在Spring Framework中配置持久化 553
 - 21.2.1 查找数据源 553
 - 21.2.2 在代码中创建持久化单元 554
 - 21.2.3 创建事务管理 557
 - 21.3 创建和使用JPA仓库 560
 - 21.3.1 注入持久化单元 560
 - 21.3.2 实现标准CRUD操作 561
 - 21.3.3 为所有的实体创建一个基础仓库 563
 - 21.3.4 在服务中标记事务范围 568
 - 21.3.5 使用事务服务方法 572
 - 21.4 在DTO和实体之间转换数据 573
 - 21.4.1 为客户支持应用程序创建实体 574

- 21.4.2 使用BCrypt保护用户密码 578
- 21.4.3 在服务中将数据传输到实体中 579
- 21.5 小结 581
- 第22章 使用Spring Data JPA消除公式化的仓库 583
 - 22.1 了解Spring Data的统一数据访问 584
 - 22.1.1 避免代码重复 584
 - 22.1.2 使用Stock仓库接口 587
 - 22.1.3 为搜索实体创建查询方法 588
 - 22.1.4 提供自定义方法实现 591
 - 22.2 配置和创建Spring Data JPA仓库 594
 - 22.2.1 启用仓库自动生成 595
 - 22.2.2 编写和使用Spring Data JPA接口 601
 - 22.3 重构客户支持应用程序 603
 - 22.3.1 转换现有仓库 603
 - 22.3.2 在支持票据中添加评论 605
 - 22.4 小结 609
- 第23章 使用JPA和Hibernate Search搜索数据 611
 - 23.1 搜索介绍 612
 - 23.1.1 了解索引的重要性 612
 - 23.1.2 采取三种不同的方式 613
 - 23.2 使用高级条件定位对象 614
 - 23.2.1 创建复杂条件查询 614
 - 23.2.2 在查询中使用OR 621
 - 23.2.3 创建有用的索引改进性能 622
 - 23.3 使用JPA的全文索引 623
 - 23.3.1 在MySQL表中创建全文索引 624
 - 23.3.2 创建和使用可搜索的仓库 625
 - 23.3.3 使全文搜索可迁移 630
 - 23.4 使用Apache Lucene和Hibernate Search索引任意数据 630
 - 23.4.1 了解Lucene全文索引 631
 - 23.4.2 使用索引元数据标注实体 632
 - 23.4.3 结合使用Hibernate Search和JPA 634
 - 23.5 小结 637
- 第24章 创建高级映射和自定义数据类型 639
 - 24.1 JPA的相关内容 639
 - 24.2 转换非标准数据类型 640
 - 24.2.1 了解特性转换器 641
 - 24.2.2 了解转换注解 642
 - 24.2.3 创建和使用特性转换器 643
 - 24.3 在实体中内嵌POJO 645
 - 24.3.1 表示嵌套的类型 645
 - 24.3.2 使属性成为可嵌入属性 645
 - 24.3.3 覆盖可内嵌列列名 647
 - 24.4 定义实体间的关系 648
 - 24.4.1 了解一对一关系 648
 - 24.4.2 使用一对多和多对一关系 650
 - 24.4.3 创建多对多关系 653
 - 24.5 处理其他常见的情况 654

- 24.5.1 使用修订和时间戳版本化实体 654
- 24.5.2 定义公共属性的抽象实体 655
- 24.5.3 映射基本的和内嵌的集合 656
- 24.5.4 持久化含有键值对的Map 659
- 24.5.5 在多个表中存储实体 660
- 24.6 创建编程式触发器 661
 - 24.6.1 在CRUD操作之前或之后执行 661
 - 24.6.2 使用实体监听器 663
- 24.7 简化客户支持应用程序 664
 - 24.7.1 映射附件的集合 664
 - 24.7.2 使用加载时织入延迟加载简单属性 666
- 24.8 小结 669
- 第 部分 使用Spring Security保护应用程序
- 第25章 介绍Spring Security 673
 - 25.1 认证的概念 673
 - 25.1.1 集成认证 674
 - 25.1.2 了解授权 682
 - 25.2 选择Spring Security的原因 685
 - 25.2.1 了解Spring Security基础 686
 - 25.2.2 使用Spring Security的授权服务 686
 - 25.2.3 配置Spring Security 687
 - 25.3 小结 687
- 第26章 使用Spring Security验证用户 689
 - 26.1 选择并配置认证提供者 690
 - 26.1.1 配置用户细节提供者 690
 - 26.1.2 使用LDAP和活动目录提供者 700
 - 26.1.3 使用OpenID进行认证 703
 - 26.1.4 remember-me认证 705
 - 26.1.5 学习其他认证提供者 706
 - 26.2 编写自己的认证提供者 707
 - 26.2.1 以正确的顺序启动 707
 - 26.2.2 创建和配置提供者 710
 - 26.2.3 缓解跨站请求伪装攻击 715
 - 26.3 小结 718
- 第27章 使用授权标签和注解 719
 - 27.1 通过声明进行授权 719
 - 27.1.1 在方法代码中检查权限 720
 - 27.1.2 采用URL安全 722
 - 27.1.3 使用注解声明权限 725
 - 27.1.4 定义方法切点规则 732
 - 27.2 了解授权决策 733
 - 27.2.1 使用访问决策投票者 733
 - 27.2.2 使用访问决策管理器 734
 - 27.3 为对象安全创建访问控制列表 736
 - 27.3.1 了解Spring Security的ACL 736
 - 27.3.2 配置访问控制列表 738
 - 27.3.3 为实体填充ACL 740
 - 27.4 在客户支持应用程序中添加授权 741

- 27.4.1 切换到自定义用户细节 742
- 27.4.2 保护服务方法 746
- 27.4.3 使用Spring Security的标签库 750
- 27.5 小结 751
- 第28章 使用OAuth保护RESTful Web服务 753
- 28.1 了解Web服务安全 754
 - 28.1.1 比较Web GUI和Web服务安全 754
 - 28.1.2 选择认证机制 754
- 28.2 介绍OAuth 756
 - 28.2.1 了解关键参与者 756
 - 28.2.2 起始：OAuth 1.0 757
 - 28.2.3 标准：OAuth 1.0a 757
 - 28.2.4 演化：OAuth 2.0 762
- 28.3 使用Spring Security OAuth 769
 - 28.3.1 创建OAuth 2.0提供者 769
 - 28.3.2 创建OAuth 2.0客户端 773
- 28.4 完成客户端支持应用程序 775
 - 28.4.1 生成请求随机数和签名 776
 - 28.4.2 实现客户端服务 777
 - 28.4.3 实现随机数服务 780
 - 28.4.4 实现令牌服务 782
 - 28.4.5 自定义资源服务器过滤器 785
 - 28.4.6 重新配置Spring Security 787
- 28.5 创建OAuth客户端应用程序 791
 - 28.5.1 自定义REST模板 792
 - 28.5.2 配置Spring Security OAuth客户端 793
 - 28.5.3 使用REST模板 795
 - 28.5.4 同时测试提供者和客户端 796
- 28.6 小结 797

精彩短评

- 1、非常好的一本讲java web的书,比起国内的讲java web的书不知道高到哪儿去了(深入分析 java web 说这个锅我不背)
- 2、新手向 Java EE、Spring 入门教程
- 3、非常契合工作中的应用
- 4、一本书搞定JAVA WEB后台编程
- 5、这书讲的还是挺全的..
- 6、果然技术还是国外的先进，国内的说实话就是代码的堆砌，这本书我看的第一感觉居然是特别舒服，很爽
- 7、等会，这个豆瓣的评分怎么回事！没人评四星以下总分怎么只有7.8
- 8、蛮好的
- 9、能让我对java web后台开发更有系统性。值得一看。

精彩书评

1、这本书的内容非常新，完全是工作中需要用到的技能，而且有很多例子是实际项目中可以直接拿来用的。这本书书写风格大气、一致，每种技术介绍前，会将历史来龙去脉以及为何如此发展书写清楚不足之处也很明显，由于JAVA EE范畴相当大，本书对具体实例介绍非常快，甚至很多概念和内容讲解得相当简略。如果读者没有基础，甚至没有相关经验，那么读来就会感觉不知所云。总的来看此本书值得推荐

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:www.tushu000.com