

Spring框架高级编程



[Spring框架高级编程_下载链接1](#)

著者:约翰逊

出版者:机械工业出版社

出版时间:2006-4

装帧:简装本

isbn:9787111186380

Spring框架是主要的开源应用程序开发框架，它使得Java/J2EE开发更容易、效率更高。本书不仅向读者展示了Spring能做什么?而且揭示了Spring完成这些功能的原理，解释其功能和动机，以帮助读者使用该框架的所有部分来开发成功的应用程序。本书涵盖

Spring的所有特性，并且演示了如何将其构成一个连贯的整体，帮助读者理解Spring方法的基本原理、何时使用Spring以及如何效仿最佳实践。所有这些都以完整的样本应用程序来举例说明。读完本书，读者将能够有效地在各种应用中(从简单的web应用程序到复杂的企业级应用程序)使用Spring。

本书旨在指导读者何使用Spring框架来实现各种有效应用程序的开发(从简单的Web应用到复杂的企业级应用)。通过一个完整的样本应用程序贯穿全书，举例说明Spring如何提供一个清晰的、分层的体系结构的基础；以及如何适当地使用Spring来实现好的OO设计和体系结构设计，以及如何应用最佳实践。

本书面向欲深入学习Spring框架的Java/J2EE体系结构设计人员和开发人员，使他们能够把spring框架有效地用于从简单的Web应用到复杂的企业级应用；也非常适合Spring框架新手作为常备参考书。

作者介绍:

目录: 第1章 Spring框架概述 1

1.1 为什么要Spring? 1

1.1.1 J2EE传统方式的问题 1

1.1.2 轻量级框架 3

1.1.3 进入Spring 4

1.2 Spring的价值 5

1.3 上下文中的Spring 6

1.3.1 技术 7

1.3.2 面向方面的编程 9

1.3.3 一致性抽象 12

1.3.4 与其他框架的关系 16

1.4 使用Spring构造应用程序 18

1.4.1 关系图 18

1.4.2 持久性和集成 20

1.4.3 业务服务对象 23

1.4.4 表示 24

1.5 前景 25

1.5.1 发布日程 25

1.5.2 Java和J2EE的进展 26

1.5.3 技术趋势 27

1.5.4 标准与源码公开 27

1.6 Spring项目和社区 28

1.6.1 历史 28

1.6.2 模块摘要 29

1.6.3 支持的环境 32

1.7 小结 32

第2章 BEAN工厂与应用程序上下文 34

2.1 控制反转和依赖注入 34

2.1.1 不同形式的依赖注入 36

2.1.2 在设置器注入和构造函数注入之间决定 37

2.2 容器 38

2.2.1 Bean工厂 39

2.2.2 应用程序上下文 40

2.2.3 启动容器 41

2.2.4 使用来自工厂的Bean 42

2.2.5 XML Bean配置 43

- 2.2.6 基本的Bean定义 43
- 2.2.7 管理Bean生命周期 52
- 2.2.8 对服务和资源访问的抽象 54
- 2.2.9 重用Bean定义 57
- 2.2.10 使用后置处理器处理定制的Bean和容器 59
- 2.3 小结 63
- 第3章 高级容器概念 64
 - 3.1 低层资源的抽象 64
 - 3.1.1 应用程序上下文作为ResourceLoader 64
 - 3.1.2 应用程序上下文作为消息源 66
 - 3.2 应用程序事件 68
 - 3.3 管理容器 69
 - 3.3.1 ApplicationContext构造函数中的资源位置路径 69
 - 3.3.2 应用程序上下文的声明性用法 71
 - 3.3.3 将容器定义划分到多个文件 72
 - 3.3.4 处理组件的策略 73
 - 3.3.5 用于访问容器的单态 75
 - 3.4 一些方便的工厂BEAN 75
 - 3.4.1 PropertyPathFactoryBean 76
 - 3.4.2 FieldRetrievingFactoryBean 76
 - 3.4.3 MethodInvokingFactoryBean 77
 - 3.5 SPRING提供的属性编辑器 77
 - 3.6 测试策略 78
 - 3.6.1 单元测试 78
 - 3.6.2 使用Spring容器的测试 80
 - 3.7 XML的替代品 81
 - 3.7.1 来自Properties文件的定义 82
 - 3.7.2 编程化的Bean定义 82
 - 3.7.3 其他格式 83
 - 3.7.4 参考文献 83
 - 3.8 小结 83
- 第4章 SPRING与AOP 84
 - 4.1 目标 84
 - 4.2 假设 85
 - 4.3 例子 85
 - 4.4 SPRING的AOP框架 86
 - 4.4.1 拦截器链 87
 - 4.4.2 赞成与反对 87
 - 4.4.3 通知 88
 - 4.4.4 切点 92
 - 4.4.5 通知器 (Advisor) 96
 - 4.4.6 与Spring IoC容器的集成 97
 - 4.4.7 在运行时间检验并处理代理状态 107
 - 4.4.8 编程化的代理创建 107
 - 4.5 使用SPRING的AOP框架的高级特性 108
 - 4.5.1 TargetSources 108
 - 4.5.2 及早终止拦截器链 112
 - 4.5.3 使用介绍 112
 - 4.5.4 暴露当前的代理 114
 - 4.5.5 暴露当前的MethodInvocation 115
 - 4.5.6 理解代理类型 115
 - 4.5.7 调试与测试 117
 - 4.5.8 其他 118
 - 4.6 与其他AOP框架集成 120

- 4.6.1 目标 120
- 4.6.2 AspectJ集成 121
- 4.6.3 AspectWerkz 125
- 4.7 参考文献 125
- 4.8 小结 126
- 第5章 DAO支持与JDBC框架 127
- 5.1 数据访问对象模式 127
- 5.2 SPRING的JDBC框架概述 128
 - 5.2.1 起因：直接使用JDBC的问题 128
 - 5.2.2 Spring可以如何帮助 130
 - 5.2.3 简单的例子 130
- 5.3 建立样本应用程序的数据访问层 131
 - 5.3.1 样本应用程序的数据模型 131
 - 5.3.2 DataSource 132
 - 5.3.3 异常转译 133
- 5.4 使用JDBCTEMPLATE的操作 134
 - 5.4.1 回调方法的使用 134
 - 5.4.2 JdbcTemplate的方便方法 135
 - 5.4.3 使用JdbcTemplate的基本查询 136
 - 5.4.4 使用JdbcTemplate的基本更新 137
 - 5.4.5 JdbcTemplate的高级用法 137
 - 5.4.6 对RowSet的支持 138
- 5.5 使用RDBMS操作类 139
 - 5.5.1 SqlQuery和MappingSqlQuery 139
 - 5.5.2 使用SqlUpdate插入和更新 140
 - 5.5.3 使用UpdatableSqlQuery更新ResultSet 141
 - 5.5.4 生成主键 141
 - 5.5.5 检索数据库生成的键 142
 - 5.5.6 调用存储过程 143
- 5.6 高级概念 144
 - 5.6.1 在应用程序服务器中的运行Spring JDBC 144
 - 5.6.2 使用定制的异常转译 145
 - 5.6.3 读写LOB数据 147
 - 5.6.4 批量更新 149
 - 5.6.5 存储过程的高级用法 150
- 5.7 其他的考虑事项 152
 - 5.7.1 性能 152
 - 5.7.2 何时使用JDBC和O/R映射 152
 - 5.7.3 JDBC版本和J2EE版本 153
- 5.8 小结 153
- 第6章 事务和资源管理 154
- 6.1 背景 154
 - 6.1.1 什么是事务？ 154
 - 6.1.2 ACID属性 154
 - 6.1.3 并发控制 156
- 6.2 事务与J2EE 156
 - 6.2.1 局部事务 157
 - 6.2.2 全局/分布式事务 157
 - 6.2.3 事务传播 157
 - 6.2.4 事务划分 157
- 6.3 SPRING事务支持的例子 158
- 6.4 SPRING的事务抽象介绍 159
 - 6.4.1 事务控制选择的概述 159
 - 6.4.2 事务划分策略 162

- 6.4.3 事务管理策略 167
- 6.5 DATASOURCE声明 171
 - 6.5.1 本地非缓冲 171
 - 6.5.2 本地缓冲 172
 - 6.5.3 JNDI 172
 - 6.5.4 在本地DataSource和JNDI DataSource之间选择 173
- 6.6 小结 173
- 第7章 对象/关系映射 175
 - 7.1 背景知识 175
 - 7.1.1 基本的O/R映射 176
 - 7.1.2 对象查询语言 176
 - 7.1.3 透明持久性 177
 - 7.1.4 何时选择O/R映射 177
 - 7.2 SPRING中的O/R映射支持 178
 - 7.2.1 数据访问对象 178
 - 7.2.2 事务管理 179
 - 7.3 IBATIS SQL MAPS 179
 - 7.3.1 映射文件 180
 - 7.3.2 DAO实现 181
 - 7.3.3 在Spring上下文中设置 182
 - 7.3.4 事务管理 182
 - 7.3.5 iBATIS概要 183
 - 7.4 HIBERNATE 184
 - 7.4.1 映射文件 184
 - 7.4.2 DAO实现 185
 - 7.4.3 Spring上下文中的设置 186
 - 7.4.4 设置为JCA连接器 187
 - 7.4.5 事务管理 188
 - 7.4.6 Open Session in View 192
 - 7.4.7 BLOB/CLOB处理 194
 - 7.4.8 Hibernate: 小结 195
 - 7.5 JDO 196
 - 7.5.1 持久性对象的生命周期 197
 - 7.5.2 DAO实现 197
 - 7.5.3 在Spring上下文中设置 199
 - 7.5.4 事务管理 200
 - 7.5.5 PersistenceManager的生命周期 200
 - 7.5.6 Open PersistenceManager in View 201
 - 7.5.7 JDO方言 203
 - 7.5.8 JDO: 小结 203
 - 7.6 其他O/R映射工具 205
 - 7.6.1 Apache OJB 205
 - 7.6.2 TopLink 206
 - 7.6.3 Cayenne 207
 - 7.6.4 JSR-220持久性 207
 - 7.7 小结 207
- 第8章 轻量级远程处理 209
 - 8.1 概念和范围 209
 - 8.2 公共的配置风格 211
 - 8.3 HESSIAN和BURLAP 212
 - 8.3.1 访问一个服务 213
 - 8.3.2 导出一个服务 214
 - 8.4 HTTP调用器 215
 - 8.4.1 访问一个服务 216

- 8.4.2 导出一个服务 216
- 8.4.3 定制化选项 217
- 8.5 RMI 218
 - 8.5.1 访问一个服务 219
 - 8.5.2 Stub查询策略 220
 - 8.5.3 导出一个服务 221
 - 8.5.4 定制化选项 222
 - 8.5.5 RMI-IIOP 222
- 8.6 通过JAX-RPC的WEB服务 223
 - 8.6.1 访问一个服务 224
 - 8.6.2 导出一个服务 225
 - 8.6.3 定制的类型映射 226
- 8.7 小结 227
- 第9章 支持性服务 229
 - 9.1 JMS 229
 - 9.1.1 引言 229
 - 9.1.2 Spring的JMS支持的起因 230
 - 9.1.3 通过模板的JMS访问 230
 - 9.1.4 异常处理 232
 - 9.1.5 ConnectionFactory管理 232
 - 9.1.6 消息转换器 233
 - 9.1.7 目的管理 233
 - 9.1.8 事务管理 234
 - 9.1.9 JmsGatewaySupport 235
 - 9.1.10 前景 236
 - 9.2 使用SPRING的调度 236
 - 9.2.1 Timer与Quartz 236
 - 9.2.2 Timer 237
 - 9.2.3 Quartz 238
 - 9.3 使用SPRING发送电子邮件 241
 - 9.3.1 入门指南 242
 - 9.3.2 重用现有的邮件会话 242
 - 9.3.3 使用COS的邮寄发送 242
 - 9.3.4 通用的邮件管理器 243
 - 9.4 脚本 245
 - 9.4.1 一致性模型 245
 - 9.4.2 其他脚本语言 248
 - 9.5 小结 248
- 第10章 SPRING的ACEGI SECURITY系统 249
 - 10.1 企业应用程序的安全选择 249
 - 10.1.1 典型的需求 249
 - 10.1.2 Acegi Security简介 250
 - 10.1.3 Java身份验证和授权服务 252
 - 10.1.4 Servlet规范 254
 - 10.2 ACEGI SECURITY的基础 255
 - 10.2.1 身份验证 255
 - 10.2.2 存储身份验证对象 257
 - 10.2.3 授权 258
 - 10.2.4 域对象实例的安全 259
 - 10.3 范例代码 262
 - 10.3.1 范例介绍 262
 - 10.3.2 安全不知道的的实现 262
 - 10.3.3 安全方法 263
 - 10.3.4 身份验证 263

- 10.3.5 授权 264
- 10.4 小结 265
- 第11章 SPRING与EJB 267
 - 11.1 对EJB的需要做出决断 267
 - 11.2 访问EJB 268
 - 11.2.1 样板方法 268
 - 11.2.2 Spring的方法 269
 - 11.3 用SPRING实现EJB 273
 - 11.3.1 无状态会话Bean 273
 - 11.3.2 有状态会话Bean 275
 - 11.3.3 消息驱动的Bean 276
 - 11.3.4 XDoclet 276
 - 11.4 单态容器访问，好还是坏？ 277
 - 11.4.1 ContextSingletonBeanFactoryLocator和SingletonBeanFactoryLocator 278
 - 11.4.2 共享的上下文作为Web应用程序的应用程序上下文的双亲 280
 - 11.4.3 使用来自EJB的共享的上下文 280
 - 11.5 测试关注点 281
 - 11.5.1 在POJO委托中实现业务功能 281
 - 11.5.2 使用模仿的EJB容器 282
 - 11.5.3 应用程序服务器内部的集成测试 282
 - 11.6 小结 283
- 第12章 WEB MVC框架 284
 - 12.1 简单的例子 284
 - 12.2 一般的体系结构 285
 - 12.2.1 Web MVC概念 285
 - 12.2.2 使用分发器和控制器的通用Web MVC 286
 - 12.2.3 好的Web MVC框架的需求 286
 - 12.2.4 Spring的Web MVC的等价物 287
 - 12.3 基础设施组件 289
 - 12.3.1 DispatcherServlet 289
 - 12.3.2 WebApplicationContext 291
 - 12.4 处理请求有关的工作流 293
 - 12.5 SPRING MVC WEB应用程序的一般布局 294
 - 12.6 HANDLERMAPPING 295
 - 12.6.1 BeanNameUrlHandlerMapping 296
 - 12.6.2 SimpleUrlHandlerMapping 296
 - 12.6.3 CommonsPathMapUrlHandlerMapping 297
 - 12.6.4 多个HandlerMapping 298
 - 12.7 HANDLEREXECUTIONCHAIN和拦截器 298
 - 12.7.1 WebContentInterceptor 299
 - 12.7.2 UserRoleAuthorizationInterceptor 300
 - 12.7.3 Spring MVC提供的其他处理器拦截器 300
 - 12.8 处理器及其适配器 300
 - 12.9 MODELANDVIEW和VIEWRESOLVERS 301
 - 12.9.1 UrlBasedViewResolver 301
 - 12.9.2 BeanNameViewResolver和XmlViewResolver 302
 - 12.9.3 ResourceBundleViewResolver 302
 - 12.9.4 视图解析器成链 303
 - 12.10 地区值的改变和解析 304
 - 12.11 HANDLEREXCEPTIONRESOLVERS 305
 - 12.12 控制器 307
 - 12.12.1 WebContentGenerator 307
 - 12.12.2 AbstractController 308
 - 12.12.3 UrlFilenameViewController 308

- 12.12.4 ParameterizableViewController 309
- 12.12.5 MultiActionController 309
- 12.13 数据绑定 310
 - 12.13.1 使用数据绑定时的方便功能 310
- 12.14 使用控制器的实际例子 311
- 12.15 使用ABSTRACTCONTROLLER查看演出列表 311
 - 12.15.1 预订 313
- 12.16 向导功能 318
 - 12.16.1 基本配置 318
 - 12.16.2 模板方法 318
 - 12.16.3 向导流程 319
 - 12.16.4 页面改变、编号和其他动作 319
- 12.17 扩展SPRING处理器的基础设施 320
- 12.18 上传文件 321
 - 12.18.1 配置多部分解析器 321
 - 12.18.2 创建表单来上传文件 321
 - 12.18.3 处理上传的数据 322
- 12.19 测试控制器 322
 - 12.19.1 无需应用程序上下文的测试 323
 - 12.19.2 使用模仿对象的更广泛测试 323
 - 12.19.3 使用完全的应用程序上下文测试 323
 - 12.19.4 测试Web应用程序的其他方法 324
 - 12.19.5 小结 324
- 第13章 WEB视图技术 326
 - 13.1 一个例子 326
 - 13.1.1 通用配置 327
 - 13.1.2 JSP 327
 - 13.1.3 FreeMarker 327
 - 13.1.4 使用iText生成PDF 328
 - 13.2 选择技术时的注意事项 328
 - 13.3 视图对象和模型 329
 - 13.4 ABSTRACTVIEW提供的特性 330
 - 13.4.1 使用RedirectView发布新的请求 330
 - 13.4.2 使用视图前缀发布转向或者重定向 331
 - 13.5 JSP 332
 - 13.5.1 配置应用程序以使用JSP 332
 - 13.5.2 使用定制标记创建表单 333
 - 13.5.3 使用标记文件创建可重用的元素 337
 - 13.6 VELOCITY和FREEMARKER 338
 - 13.6.1 配置视图解析器 338
 - 13.6.2 使用表单简化宏 340
 - 13.7 TILES 342
 - 13.8 基于XML和XSLT的视图 344
 - 13.9 EXCEL和基于其他文档的视图 345
 - 13.9.1 从表演列表生成Excel文件 345
 - 13.9.2 以模板作为Excel文件的基础 346
 - 13.9.3 基于其他文档的视图 346
 - 13.9.4 使用HandlerInterceptor区分HTML和Excel 346
 - 13.10 实现定制视图 347
 - 13.10.1 View和AbstractView 347
 - 13.10.2 实现从数据对象生成XML的视图 348
 - 13.10.3 实现定制视图时的注意事项 348
 - 13.11 小结 349
- 第14章 与其他WEB框架集成 350

- 14.1 选择MVC框架时的注意事项 350
 - 14.1.1 比较传统的Web MVC框架 350
- 14.2 与SPRING集成：核心概念 359
- 14.3 WEBWORK 集成 361
 - 14.3.1 设置ObjectFactory 361
- 14.4 STRUTS集成 361
 - 14.4.1 使用ActionSupport 362
 - 14.4.2 使用DelegationRequestProcessor和DelegationActionProxy 362
 - 14.4.3 使用自动装配的基本动作 364
- 14.5 TAPESTRY集成 365
 - 14.5.1 为Tapestry准备Bean 365
 - 14.5.2 页面类 365
 - 14.5.3 页面定义 366
 - 14.5.4 页面模板 366
 - 14.5.5 Tapestry集成的最终思考 366
- 14.6 JSF集成 367
- 14.7 小结 368
- 第15章 样本应用程序 369
 - 15.1 服务器技术的选择 369
 - 15.2 应用层 370
 - 15.3 持久性层 370
 - 15.3.1 数据模型 370
 - 15.3.2 域对象模型 371
 - 15.3.3 对象/关系映射 371
 - 15.3.4 DAO实现 373
 - 15.3.5 数据访问上下文 374
 - 15.4 业务服务层 374
 - 15.4.1 服务 374
 - 15.4.2 应用程序上下文 375
 - 15.5 WEB层 375
 - 15.5.1 应用程序流程 376
 - 15.5.2 通过web.xml配置应用程序 376
 - 15.5.3 Web控制器 377
 - 15.5.4 视图技术 377
 - 15.6 与J2EE设计和开发实现的比较 378
 - 15.6.1 更简单的技术 378
 - 15.6.2 数据库变化 378
 - 15.7 服务器配置 378
 - 15.7.1 MySQL 379
 - 15.7.2 Tomcat 379
 - 15.8 组建和部署 379
 - 15.8.1 创建和载入数据库表 380
 - 15.8.2 组建应用程序并部署到Tomcat服务器 380
 - 15.9 小结 380
- 第16章 结论 381
 - 16.1 SPRING解决的问题 381
 - 16.2 SPRING解决方案 381
 - 16.3 SPRING开发的指导方针 383
 - 16.3.1 技术选择 383
 - 16.3.2 层与层 385
 - 16.3.3 构造应用程序 389
 - 16.3.4 测试应用程序 392
 - 16.4 相关项目 394
 - 16.4.1 Spring的Acegi Security 394

- 16.5 其他项目 395
- 16.6 非J2EE环境中的SPRING 395
- 16.7 发现更多 396
 - 16.7.1 书籍和文章 396
 - 16.7.2 在线资源 397
 - 16.7.3 样本应用程序 397
- 16.8 前景.. 398
- 附录A 样本应用程序的需求 399
 - A.1 概述 399
 - A.2 用户群体 399
 - A.2.1 公众Internet用户 399
 - A.2.2 售票处用户 400
 - A.2.3 系统管理员 400
 - A.3 假设 400
 - A.4 范围限制 401
 - A.5 交付时间表 401
 - A.6 INTERNET用户界面 401
 - A.6.1 基本工作流 402
 - A.6.2 错误处理 402
 - A.6.3 应用程序屏幕 402
 - A.7 非功能性需求 409
 - A.8 硬件和软件环境 410
 - • • • • [\(收起\)](#)

[Spring框架高级编程_下载链接1](#)

标签

spring

Java

J2EE

Spring

OpenSource

计算机

编程

工作

评论

对于老版本的内容来说还是挺全的，缺点的话如下： 1
aop这部分概念不少但是总觉的缺点啥 2 涉及主题多了，难免有点泛泛而谈 3
翻译质量的确……

最近需要用，周末又翻出来看了看，还是有不少收获。孔子曾经曰过：学而时习之，不亦乐乎。

内容比较旧，但是对spring讲解比较透彻。

翻译的难以置信的差

很不错的一本书。

[Spring框架高级编程 下载链接1](#)

书评

计算机的中译书一般都会挨骂,这本骂得尤为厉害,淘了一本,是正版的,纸张一般也就算了,但是这个翻译质量真的是不敢恭维,我虽然是个菜鸟,但是这样翻译不通顺,主次不分的,很令人纠结啊,不敢恭维,就看了两页,我就很郁闷了,其中有一句说"应用程序上下文也是一个Bean工厂,使用...(省去...

Wrox不出什么烂书 只有翻译的烂 重者的重点是在说Spring的设计原理
Spring是拿来对付哪些问题的 实际案例不多 适合深度研究Spring的孩子们看
AOP那章比Spring In action说的更深入

[Spring框架高级编程_下载链接1](#)