

# 目录

## 第一部分 Spring 的核心

1	第 1 章 Spring 之旅 2	
	1.1 简化 Java 开发 3	
	1.1.1 激发 POJO 的潜能 4	
	1.1.2 依赖注入 5	
	1.1.3 应用切面 9	
	1.1.4 使用模板消除样板式代码 13	
	1.2 容纳你的 Bean 15	
	1.2.1 与应用上下文共事 16	
	1.2.2 Bean 的生命周期 17	
	1.3 俯瞰 Spring 风景线 19	
	1.3.1 Spring 模块 19	
	1.3.2 Spring Portfolio 22	
	1.4 Spring 新功能 25	
	1.4.1 Spring 2.5 新特性 26	
	1.4.2 Spring 3.0 新特性 26	
	1.4.3 Spring Portfolio 新特性 27	
	1.5 小结 28	
2	第 2 章 装配 Bean 29	
	2.1 声明 Bean 30	
	2.1.1 创建 Spring 配置 30	
	2.1.2 声明一个简单 Bean 31	
	2.1.3 通过构造器注入 33	
	2.1.4 Bean 的作用域 37	
	2.1.5 初始化和销毁 Bean 38	
	2.2 注入 Bean 属性 40	
	2.2.1 注入简单值 41	
	2.2.2 引用其他 Bean 42	
	2.2.3 使用 Spring 的命名空间 p 装配属性 45	
	2.2.4 装配集合 46	
	2.2.5 装配空值 50	
	2.3 使用表达式装配 51	
	2.3.1 SpEL 的基本原理 51	
	2.3.2 在 SpEL 值上执行操作 54	
	2.3.3 在 SpEL 中筛选集合 58	
	2.4 小结 62	
3	第 3 章 最小化 Spring XML 配置 63	
	3.1 自动装配 Bean 属性 64	
	3.1.1 4 种类型的自动装配 64	
	3.1.2 默认自动装配 68	
	3.1.3 混合使用自动装配和显式装配 68	
	3.2 使用注解装配 69	
	3.2.1 使用 @Autowired 70	
	3.2.2 借助 @Inject 实现基于标准的 自动装配 74	
	3.2.3 在注解注入中使用表达式 76	
	3.3 自动检测 Bean 77	
	3.3.1 为自动检测标注 Bean 78	
	3.3.2 过滤组件扫描 79	
	3.4 使用 Spring 基于 Java 的 配置 80	
	3.4.1 创建基于 Java 的配置 80	
	3.4.2 定义一个配置类 81	
	3.4.3 声明一个简单的 Bean 81	
	3.4.4 使用 Spring 的基于 Java 的配 置进行注入 82	

4

3.5 小结 83

**第4章 面向切面的 Spring 84**

4.1 什么是面向切面编程 85

4.1.1 定义 AOP 术语 86

4.1.2 Spring 对 AOP 的支持 88

4.2 使用切点选择连接点 90

4.2.1 编写切点 91

4.2.2 使用 Spring 的 bean() 指示器 92

4.3 在 XML 中声明切面 93

4.3.1 声明前置和后置通知 94

4.3.2 声明环绕通知 96

4.3.3 为通知传递参数 98

4.3.4 通过切面引入新功能 100

4.4 注解切面 102

4.4.1 注解环绕通知 104

4.4.2 传递参数给所标注的通知 105

4.4.3 标注引入 105

4.5 注入 AspectJ 切面 107

4.6 小结 109

## 第二部分 Spring 应用程序的核心组件

5

**第5章 征服数据库 112**

5.1 Spring 的数据访问哲学 113

5.1.1 了解 Spring 的数据访问异常体系 114

5.1.2 数据访问模板化 116

5.1.3 使用 DAO 支持类 118

5.2 配置数据源 119

5.2.1 使用 JNDI 数据源 119

5.2.2 使用数据源连接池 120

5.2.3 基于 JDBC 驱动的数据源 121

5.3 在 Spring 中使用 JDBC 122

5.3.1 应对失控的 JDBC 代码 122

5.3.2 使用 JDBC 模板 125

5.4 在 Spring 中集成 Hibernate 130

5.4.1 Hibernate 概览 131

5.4.2 声明 Hibernate 的 Session 工厂 132

5.4.3 构建不依赖于 Spring 的 Hibernate 代码 134

5.5 Spring 与 Java 持久化 API 136

5.5.1 配置实体管理器工厂 136

5.5.2 编写基于 JPA 的 DAO 140

5.6 小结 142

6.2.1 JDBC 事务 149

6.2.2 Hibernate 事务 149

6.2.3 Java 持久化 API 事务 150

6.2.4 JTA (Java Transaction API) 事务 151

6.3 在 Spring 中的编码事务 151

6.4 声明式事务 153

6.4.1 定义事务属性 154

6.4.2 在 XML 中定义事务 157

6.4.3 定义注解驱动的事务 159

6.5 小结 160

7

**第6章 事务管理 144**

6.1 理解事务 145

6.1.1 用4个词来表示事务 146

6.1.2 理解 Spring 对事务管理的支持 147

6.2 选择事务管理器 147

**第7章 使用 Spring MVC 构建 Web 应用程序 162**

7.1 Spring MVC 起步 163

7.1.1 跟踪 Spring MVC 的请求 163

7.1.2 搭建 Spring MVC 165

7.2 编写基本的控制器 166

7.2.1 配置注解驱动的 Spring MVC 167

7.2.2 定义首页的控制器 168

7.2.3 解析视图 171

7.2.4 定义首页的视图 175

7.2.5 完成 Spring 应用上下文 177

7.3 处理控制器的输入 178

7.3.1 编写处理输入的控制器 179

7.3.2 渲染视图 181

7.4 处理表单 183

7.4.1 展现注册表单 183

7.4.2 处理表单输入 185

7.4.3 校验输入 187

6



- 7.5 处理文件上传 191
  - 7.5.1 在表单上添加文件上传域 191
  - 7.5.2 接收上传的文件 192
  - 7.5.3 配置 Spring 支持文件上传 195
- 7.6 小结 196

8

## 第 8 章 使用 Spring Web Flow 197

- 8.1 安装 Spring Web Flow 198
  - 8.1.1 在 Spring 中使用 WebFlow 198
- 8.2 流程的组件 201
  - 8.2.1 状态 201
  - 8.2.2 转移 204
  - 8.2.3 流程数据 205
- 8.3 组合起来：披萨流程 207
  - 8.3.1 定义基本流程 207
  - 8.3.2 收集顾客信息 211
  - 8.3.3 构建订单 216
  - 8.3.4 支付 219
- 8.4 保护 Web 流程 211
- 8.5 小结 221

9

## 第 9 章 保护 Spring 应用 223

- 9.1 Spring Security 介绍 224
  - 9.1.1 Spring Security 起步 224

- 9.1.2 使用 Spring Security 配置命名空间 225
- 9.2 保护 Web 请求 226
  - 9.2.1 代理 Servlet 过滤器 226
  - 9.2.2 配置最小化的 Web 安全性 227
  - 9.2.3 拦截请求 231
- 9.3 保护视图级别的元素 234
  - 9.3.1 访问认证信息的细节 234
  - 9.3.2 根据权限渲染 235
- 9.4 认证用户 237
  - 9.4.1 配置内存用户存储库 238
  - 9.4.2 基于数据库进行认证 239
  - 9.4.3 基于 LDAP 进行认证 240
  - 9.4.4 启用 remember-me 功能 244
- 9.5 保护方法调用 245
  - 9.5.1 使用 @Secured 注解保护方法调用 245
  - 9.5.2 使用 JSR-250 的 @RolesAllowed 注解 246
  - 9.5.3 使用 SpEL 实现调用前后的安全性 246
  - 9.5.4 声明方法级别的安全性切点 250
- 9.6 小结 251

## 第三部分 Spring 集成

10

## 第 10 章 使用远程服务 254

- 10.1 Spring 远程调用概览 255
- 10.2 使用 RMI 257
  - 10.2.1 发布一个 RMI 服务 257
  - 10.2.2 装配 RMI 服务 260
- 10.3 使用 Hessian 和 Burlap 发布远程服务 262
  - 10.3.1 使用 Hessian 和 Burlap 发布 Bean 的功能 263
  - 10.3.2 访问 Hessian/Burlap 服务 266
- 10.4 使用 Spring 的 HttpInvoker 267
  - 10.4.1 将 Bean 发布为 HTTP 服务 267
  - 10.4.2 通过 HTTP 访问服务 269
- 10.5 发布和使用 Web 服务 270

11

## 第 11 章 为 Spring 添加 REST 功能 277

- 10.5.1 创建 JAX-WS 端点 271
- 10.5.2 在客户端代理 JAX-WS 服务 275
- 10.6 小结 276
- 11.1 了解 REST 278
  - 11.1.1 REST 的基本原理 278
  - 11.1.2 Spring 是如何支持 REST 的 279
- 11.2 编写面向资源的控制器 279
  - 11.2.1 剖析 RESTless 的控制器 280
  - 11.2.2 处理 RESTful URL 281
  - 11.2.3 执行 REST 动作 284
- 11.3 表述资源 287
  - 11.3.1 协商资源表述 287

- 11.3.2 使用 HTTP 信息转换器 290
- 11.4 编写 REST 客户端 293
  - 11.4.1 了解 RestTemplate 的操作 295
  - 11.4.2 GET 资源 296
  - 11.4.3 PUT 资源 298
  - 11.4.4 DELETE 资源 300
  - 11.4.5 POST 资源数据 301
  - 11.4.6 交换资源 303
- 11.5 提交 RESTful 表单 305
  - 11.5.1 在 JSP 中渲染隐藏的方法域 306
  - 11.5.2 发布真正的请求 307
- 11.6 小结 308

## 12 第 12 章 Spring 消息 310

- 12.1 JMS 简介 311
  - 12.1.1 构建 JMS 312
  - 12.1.2 评估 JMS 的优点 314
- 12.2 在 Spring 中搭建消息代理 316
  - 12.2.1 创建连接工厂 316
  - 12.2.2 声明 ActiveMQ 消息目的地 317
- 12.3 使用 Spring 的 JMS 模板 318
  - 12.3.1 处理失控的 JMS 代码 318
  - 12.3.2 使用 JMS 模板 319
- 12.4 创建消息驱动的 POJO 324
  - 12.4.1 创建消息监听器 325
  - 12.4.2 配置消息监听器 326
- 12.5 使用基于消息的 RPC 327
  - 12.5.1 使用 Spring 基于消息的 RPC 328
  - 12.5.2 使用 Lingo 实现异步 RPC 330
- 12.6 小结 332

## 13 第 13 章 使用 JMX 管理

### Spring Bean 333

- 13.1 将 Spring Bean 导出为 MBean 334
  - 13.1.1 通过名称发布方法 337
  - 13.1.2 使用接口定义 MBean 的操作和属性 339
  - 13.1.3 使用注解驱动的 MBean 340
  - 13.1.4 处理 MBean 冲突 342
- 13.2 远程 MBean 343
  - 13.2.1 发布远程 MBean 343
  - 13.2.2 访问远程 MBean 344
  - 13.2.3 代理 MBean 346
- 13.3 处理通知 347
  - 13.3.1 监听通知 348
- 13.4 小结 349

## 14 第 14 章 其他 Spring 技巧 350

- 14.1 外部化配置 351
  - 14.1.1 替换属性占位符 351
  - 14.1.2 重写属性 354
  - 14.1.3 加密外部属性 355
- 14.2 装配 JNDI 对象 357
  - 14.2.1 JNDI 的传统用法 357
  - 14.2.2 装配 JNDI 对象 359
  - 14.2.3 将 EJB 装配到 Spring 中 362
- 14.3 发送邮件 363
  - 14.3.1 配置邮件发送器 363
  - 14.3.2 构建邮件 365
- 14.4 调度和后台任务 370
  - 14.4.1 声明调度方法 371
  - 14.4.2 声明异步方法 372
- 14.5 小结 373
- 14.6 结束语 374