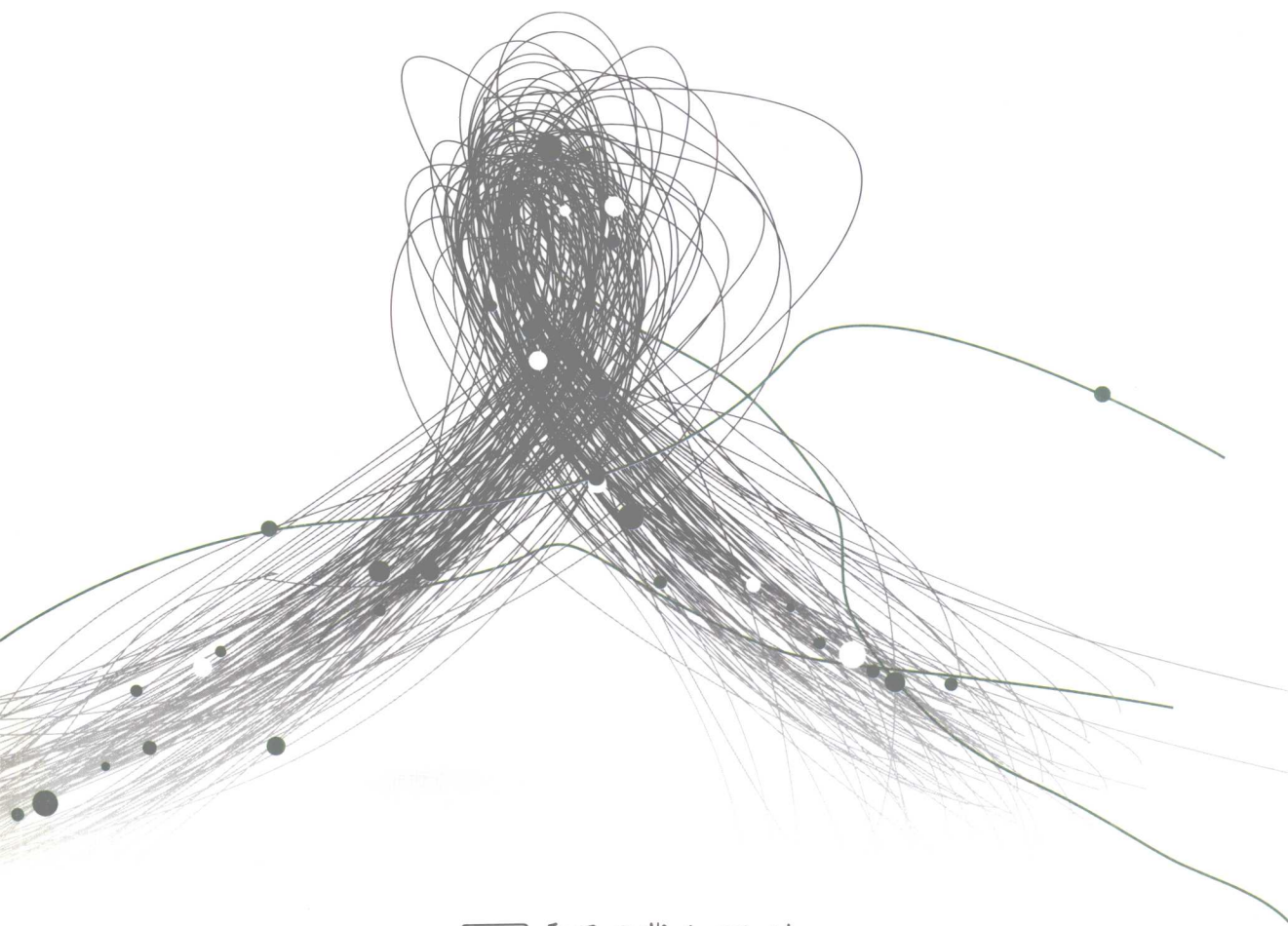


Spring 3.x

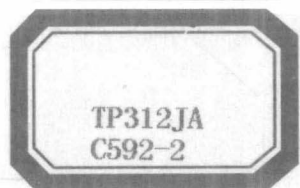
陈雄华 林开雄 著

企业应用开发实战

随书光盘附赠
两章原创内容及项目源码



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>



郑州大学 *04010800820N*

spring

Spring 3.x

陈雄华 林开雄 著

企业应用开发实战



TP312JA
C592-2

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

Spring 3.0 是 Spring 在积蓄了 3 年之久后,隆重推出的一个重大升级版本,进一步加强了 Spring 作为 Java 领域第一开源平台的翘楚地位。

Spring 3.0 引入了众多 Java 开发者翘首以盼的新功能和新特性,如 OXM、校验及格式化框架、REST 风格的 Web 编程模型等。这些新功能实用性强、易用性高,可大幅降低 Java 应用,特别是 Java Web 应用开发的难度,同时有效提升应用开发的优雅性。

本书是在《精通 Spring 2.x——企业应用开发详解》的基础上,经过历时一年的重大调整改版而成的,本书延续了上一版本追求深度,注重原理,不停留在技术表面的写作风格,力求使读者在熟练使用 Spring 的各项功能的同时,还能透彻理解 Spring 的内部实现,真正做到知其然知其所以然。此外,本书重点突出了“实战性”的主题,力求使全书“从实际项目中来,到实际项目中去”。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有,侵权必究。

图书在版编目(CIP)数据

Spring 3.x 企业应用开发实战 / 陈雄华, 林开雄著. —北京: 电子工业出版社, 2012.2
ISBN 978-7-121-15213-9

I. ①S… II. ①陈… ②林… III. ①JAVA 语言—程序设计 IV. ①TP312

中国版本图书馆 CIP 数据核字(2011)第 241383 号

责任编辑: 李 冰

文字编辑: 江 立

印 刷: 北京东光印刷厂

装 订: 三河市皇庄路通装订厂

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本: 787×1092 1/16 印张: 45.5 字数: 1158 千字

印 次: 2012 年 3 月第 2 次印刷

印 数: 3 000 册 定价: 90.00 元(含光盘 1 张)

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话:(010) 88254888。

质量投诉请发邮件至 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线:(010) 88258888。

前言

本书小述

Spring 为 Java 世界带来了一场震撼性的技术革新，它颠覆了传统 Java 应用开发笨重的方式，影响和正在影响着 Java 开发者思考问题的方法。从 2004 年发布第一个版本以来，Spring 逐渐占据了 Java 开发人员的视线，博得了开源社区一片赞誉之声，开源社区里“春”城无处不飞花。

在 2006 年发布 Spring 2.0 版本后，Spring 的易用性，扩展性和整合性得到了充分的展示，越来越多 Java 开发者争先恐后地投奔到 Spring 平台中来，Spring 已成为事实上的 Java 应用开发平台，成为 Java 一站式轻量级平台的无冕之王。

在历经长达 3 年的磨砺之后，2009 年 Spring 3.0 横空出世，挟带着 SpEL、OXM、REST、验证/格式化等众多令人惊艳的新功能再次掀起一场 Spring 的热潮。笔者在 2007 年曾撰写的拙作《精通 Spring 2.x》已经显得昨日黄花，有感于读者朋友的青睐和出版社朋友的力促，一直希望能与时俱进将本书翻版更新，但囿于这两年工作繁忙且笔者又不希望草率应付，遂使心中夙愿沉积日久渐成心病。由于在可预知的未来皆难有大段空暇的时间，想起刘墉之创作《萤窗小语》都是掇拾繁忙之空隙集腋成裘，因此，笔者亦学习效仿之，利用节假日，周末等时间完成原版的改造，从 2010 年国庆开始到 2011 年国庆结束，历时已一年矣。

本次改版，不但将全书内容更新同步到 Spring 3.0，还对原版内容进行了大面积的优化和调整。例如，对单元测试、WebService 等内容进行了全新重写，对 Spring 事务管理各种疑难困惑详细剖析，此外还引入一章逼真再现一个实战性 Web 项目的开发全过程等。所有这些调整的目的都是希望在延续前版的“深入”的同时，能让本书更贴近于“实战”。

本书的特点

- **揭示内幕、深入浅出：**笔者对 Spring 的源码进行了彻底的分析，深刻揭示了 Spring 框架的技术内幕，让读者知其然，更知其所以然。Spring 中许多设计经验、技巧、模式具有很高的借鉴性，在透彻学习 Spring 体系结构的同时，读者可以直接将这些方法引借到具体的应用开发中。
- **同步更新、与时俱进：**虽然在 2009 年 9 月就发布 Spring 3.0 第一个候选版本，后来又发布了多个 RC 版本，并最终于 2011 年 10 月发布了 Spring 3.1 的正式版本。新功能的添加以及旧功能的调整从来就没有停止过。笔者在本书写作过程中时刻关注 Spring 新版本功能的变化，并及时调整全书内容与其同步，保证全书内容与时俱进。

- **突出重点，淡化边缘：**虽然全书洋洋洒洒近 800 页，便本书没有片面追求内容的面面俱到，相反，我们特别注意内容的剪裁和取舍。对于实用性强的知识点深入分析、深度挖掘，而对于不常使用的知识点到为止，甚至不纳入本书的范围。举例来说，我们对使用 Spring-WS 开发基于 Spring 的 Web Service 应用、OXM、Spring MVC 等这些实用性强的技术都进行了深入的分析，而对如何集成 EJB、JMX、JCA 等这些不常使用的功能完全不涉及。很好地做到了实用性和深入性二者的统一。
- **理论透彻、面向实践：**本书在透彻分析原理、讲解技术知识点的同时，特别注意与实际应用的结合，笔者将自身丰富的实战经验糅合到全书的相关知识点上，很好地做到了知识讲解和实战经验的结合。让读者在掌握纯技术知识的同时，能够对如何活用技术做到胸有成竹。如笔者在第 13 章讲解任务调度的内容时，专门辟出 13.6 小节讲解实际应用中任务调度的使用经验；在第 18 章中讲述使用 JavaMail 发送电子邮件时，专门通过 18.4 小节讲述了在实际应用中发送邮件的各种注意事项。此外，我们还适时提供了“实战经验”的插文，它们在不影响上下文连贯性的同时让读者学习到了相关技术的实战经验。诸如此类以实际应用为导向的内容贯穿全书，这是本书区别于其他书籍特色之一。
- **代码简洁、图例丰富：**全书的代码在排版布局以及内容的剪裁上颇费心思，实例代码重点关注当前知识点涉及的内容，弱化边缘代码，并采用特殊的排版方式适时添加简明扼要的注释，方便程序代码的阅读和重点内容的把握。全书拥有大量精美的图表，这些图表很好地解构了上下文中一些难点的知识，大大提高了阅读性，降低了理解的难度。
- **注重趣味、轻松阅读：**由于技术书籍的严谨性、知识性的特点，阅读技术书籍往往是枯燥乏味的，更遑论趣味性。笔者对此深有感触，为寻求一些突破，我们在全书大部分章节都精心设计一个“轻松一刻”，它们和上下文内容存在某种程度的关联性，但其本身是一段趣味性的短文，它们在增强全书趣味性的同时，还为读者提供另外一个思考问题的角度。
- **相关知识、一网打尽：**Spring 不但本身涉及众多 Java 技术，其集成的第三方技术本身也涵盖了丰富的知识。我们在介绍 Spring 相关技术时，都会简明扼要地讲解相关的基础知识，这包括 JDK 5.0 的新知识和被集成技术的知识，准备好知识背景，而不是完全脱离背景知识的情况下孤立讲解 Spring 的知识。
- **历时一载、倾力打造：**本书从筹划到全书改版完成，历经近一年的时间，笔者充分利用所有可用的空闲时间，多次拖延完稿的计划，终于于 2011 年国庆才完成本书的所有稿件。

本书的结构

本书分为 5 大篇，其中第 1 篇为 Spring 概述性知识；第 2 篇讲解了 Spring 的 IoC 和 AOP 的知识；第 3 篇讲解 Spring 的各种数据访问技术的内容；第 4 篇讲解业务层和 Web 层的技术；第 5 篇讲解面向实践的单元测试及实战项目。由于篇幅所限，笔者将邮件发送及 WebService 的章节以电子文档的形式作为附件放到光盘中，请读者通过光盘进行阅读。

下面简要介绍一下每章的内容。

第 1 章: 对 Spring 框架进行宏观性的概述, 力图使读者建立起对 Spring 整体性的认识。

第 2 章: 通过一个简单的例子展现开发 Spring Web 应用的整体过程, 通过这个实例, 读者可以快速跨入 Spring Web 应用的世界。

第 3 章: 讲解 Spring IoC 容器的知识, 通过具体的实例详细地讲解 IoC 概念。同时, 对 Spring 框架的三个最重要的框架级接口进行了剖析, 并对 Bean 的生命周期进行讲解。

第 4 章: 讲解如何在 Spring 配置文件中使用 Spring 3.0 的 Schema 格式配置 Bean 的内容, 并对各个配置项的意义进行了深入的说明。

第 5 章: 对 Spring 容器进行解构, 从内部探究 Spring 容器的体系结构和运行流程。此外, 我们还将对 Spring 容器一些高级主题进行深入的阐述。

第 6 章: 我们从 Spring AOP 的底层实现技术入手, 一步步深入到 Spring AOP 的内核中, 分析它的底层结构和具体实现。

第 7 章: 对如何使用基于 AspectJ 配置 AOP 的知识进行了深入的分析, 这包括使用 XML Schema 配置文件、使用注解进行配置等内容。

第 8 章: 介绍了 Spring 所提供的 DAO 封装层, 这包括 Spring DAO 的异常体系、数据访问模板等内容。

第 9 章: 介绍了 Spring 事务管理的工作机制, 通过 XML、注解等方式进行事务管理配置, 同时还讲解了 JTA 事务配置知识。

第 10 章: 对实际应用中 Spring 事务管理各种疑难问题进行透彻的剖析, 让读者对 Spring 事务管理不再有云遮雾罩的感觉。

第 11 章: 讲解了如何使用 Spring JDBC 进行数据访问操作, 我们还重点讲述了 LOB 字段处理、主键产生和获取等难点知识。

第 12 章: 讲解了如何在 Spring 中集成 Hibernate、myBatis 等数据访问框架, 同时, 读者还将学习到 ORM 框架的混用和 DAO 层设计的知识。

第 13 章: 本章重点对在 Spring 中如何使用 Quartz 进行任务调度进行了讲解, 同时还涉及了使用 JDK Timer 和 JDK 5.0 执行器的知识。

第 14 章: 介绍 Spring 3.0 新增的 OXM 模块, 同时对 XML 技术进行了整体的了解。

第 15 章: 对 Spring MVC 框架进行详细介绍, 对 REST 风格编程方式进行重点讲解, 同时还对 Spring 3.0 的校验和格式化框架如果和 Spring MVC 整合进行讲解。

第 16 章: 有别于一般书籍的单元测试内容, 本书以当前最具实战的 JUnit4+Unitils+Mockito 复合测试框架对如何测试数据库、Web 的应用进行了深入的讲解。

第 17 章: 以一个实际的项目为蓝本, 带领读者从项目需求分析、项目设计、代码开发、单元测试直到应用部署经历整个实际项目的整体开发过程。

如何使用本书

读者应该在机器上安装 MyEclipse 8.5, 并下载 Spring 3.0 的最新发布包, 在机器上重现书中实例的开发过程。毕竟程序开发是实践性极强的工作, 只有亲身体验才能掌握其真谛。

配套光盘拥有本书所有实例的代码，读者也可以在此基础上重复本书的实例的开发过程，省去重新录入代码之苦。

本书的插文

本书会适时加入一些提示、实战经验和轻松一刻的小段插文，在不打断行文的同时提供一些有益的开发经验、使用技巧并增强阅读的趣味性。这些插文都带有一个小图标加以突显，说明如下：

	提示：在上下文中可能存在一些读者容易忽视或容易犯错的地方，在提示信息中给予针对性的帮助信息。
	实战经验：笔者将多年的开发实战经验适时介绍给大家。这些知识往往是不能从一般的书籍或资料中获得的。本书会适时地在行文中将这些实战经验分享出来，相信可以使读者朋友少走一些弯路。
	轻松一刻：为了增强技术书籍阅读的趣味性，全书每章几乎都有一到两个轻松一刻的短文，它们和上下文内容都存在某种程度的关联性，不但为阅读带来了趣味性，还可以启发读者的思考。

此外，由于 Spring 3.x 拥有多个版本，为了保持行文的简洁，除非特别指出，本书的 Spring 或 Spring 3.0 即代表当前最新的版本（Spring 3.1.x）。

如何与作者联系

由于 Spring 内容涵盖面宽广，涉及的内容非常多，同时由于作者水平有限，错误之处在所难免。我们不但欢迎读者朋友来信交流，更期待各界高手、专家就不足之处给予赐教和斧正。您可以通过 quickselect@yahoo.com.cn 与笔者联系。

陈雄华 厦门

目 录

第 1 篇 概述

第 1 章 Spring 概述	2
1.1 认识 Spring	3
1.2 关于 SpringSource	4
1.3 Spring 带给我们什么	5
1.4 Spring 体系结构	6
1.5 Spring 3.0 的新功能	8
1.5.1 核心 API 更新到 Java 5.0	8
1.5.2 Spring 表达式语言	8
1.5.3 可通过 Java 类提供 IoC 配置信息	9
1.5.4 通用类型转换系统和属性格式化系统	10
1.5.5 数据访问层新增 OXM 功能	10
1.5.6 Web 层的增强	10
1.5.7 其他	11
1.6 Spring 对 Java 版本的要求	11
1.7 如何获取 Spring	11
1.8 小结	12
第 2 章 快速入门	13
2.1 实例功能概述	14
2.1.1 比 Hello World 更适用的实例	14
2.1.2 实例功能简介	14
2.2 环境准备	16
2.2.1 创建库表	16
2.2.2 建立工程	17
2.2.3 类包及 Spring 配置文件规划	19
2.3 持久层	20

2.3.1 建立领域对象	20
2.3.2 UserDao	21
2.3.3 LoginLogDao	24
2.3.4 在 Spring 中装配 DAO	24
2.4 业务层	26
2.4.1 UserService	26
2.4.2 在 Spring 中装配 Service	27
2.4.3 单元测试	29
2.5 展现层	31
2.5.1 配置 Spring MVC 框架	31
2.5.2 处理登录请求	33
2.5.3 JSP 视图页面	35
2.6 运行 Web 应用	37
2.7 小结	38

第 2 篇 IoC 和 AOP

第 3 章 IoC 容器概述	40
3.1 IoC 概述	41
3.1.1 通过实例理解 IoC 的概念	41
3.1.2 IoC 的类型	43
3.1.3 通过容器完成依赖关系的注入	45
3.2 相关 Java 基础知识	46
3.2.1 简单实例	46
3.2.2 类装载器 ClassLoader	48
3.2.3 Java 反射机制	51
3.3 资源访问利器	53
3.3.1 资源抽象接口	53
3.3.2 资源加载	56
3.4 BeanFactory 和 ApplicationContext	58

3.4.1	BeanFactory 介绍	58	4.8	Bean 作用域	113
3.4.2	ApplicationContext 介绍	61	4.8.1	singleton 作用域	113
3.4.3	父子容器	68	4.8.2	prototype 作用域	114
3.5	Bean 的生命周期	68	4.8.3	Web 应用环境相关的 Bean 作用域	115
3.5.1	BeanFactory 中 Bean 的生命周期 ..	68	4.8.4	作用域依赖问题	117
3.5.2	ApplicationContext 中 Bean 的 生命周期	77	4.9	FactoryBean	118
3.6	小结	79	4.10	基于注解的配置	120
第 4 章 在 IoC 容器中装配 Bean					
4.1	Spring 配置概述	81	4.10.1	使用注解定义 Bean	120
4.1.1	Spring 容器高层视图	81	4.10.2	使用注解配置信息启动 Spring 容器	120
4.1.2	基于 XML 的配置	82	4.10.3	自动装配 Bean	122
4.2	Bean 基本配置	84	4.10.4	Bean 作用范围及生命过程方法 ..	125
4.2.1	装配一个 Bean	84	4.11	基于 Java 类的配置	127
4.2.2	Bean 的命名	85	4.11.1	使用 Java 类提供 Bean 定义 信息	127
4.3	依赖注入	86	4.11.2	使用基于 Java 类的配置信息 启动 Spring 容器	130
4.3.1	属性注入	86	4.12	不同配置方式比较	132
4.3.2	构造函数注入	89	4.13	小结	134
4.3.3	工厂方法注入	93	第 5 章 Spring 容器高级主题		
4.3.4	选择注入方式的考量	94	5.1	Spring 容器技术内幕	136
4.4	注入参数详解	95	5.1.1	内部工作机制	136
4.4.1	字面值	95	5.1.2	BeanDefinition	139
4.4.2	引用其他 Bean	96	5.1.3	InstantiationStrategy	140
4.4.3	内部 Bean	98	5.1.4	BeanWrapper	140
4.4.4	null 值	98	5.2	属性编辑器	141
4.4.5	级联属性	98	5.2.1	JavaBean 的编辑器	142
4.4.6	集合类型属性	99	5.2.2	Spring 默认属性编辑器	145
4.4.7	简化配置方式	103	5.2.3	自定义属性编辑器	146
4.4.8	自动装配	106	5.3	使用外部属性文件	149
4.5	方法注入	107	5.3.1	使用外部属性文件	149
4.5.1	lookup 方法注入	107	5.3.2	使用加密的属性文件	151
4.5.2	方法替换	108	5.3.3	属性文件自身的引用	155
4.6	<bean>之间的关系	109	5.4	引用 Bean 的属性值	156
4.6.1	继承	109	5.5	国际化信息	158
4.6.2	依赖	110	5.5.1	基础知识	158
4.6.3	引用	111			
4.7	整合多个配置文件	112			

5.5.2	MessageSource	163
5.5.3	容器级的国际化信息资源	166
5.6	容器事件	167
5.6.1	Spring 事件类结构	168
5.6.2	解构 Spring 事件体系的具体实现	169
5.6.3	一个实例	170
5.7	小结	172

第6章 Spring AOP 基础 173

6.1	AOP 概述	174
6.1.1	AOP 到底是什么	174
6.1.2	AOP 术语	176
6.1.3	AOP 的实现者	178
6.2	基础知识	178
6.2.1	带有横切逻辑的实例	178
6.2.2	JDK 动态代理	181
6.2.3	CGLib 动态代理	184
6.2.4	AOP 联盟	186
6.2.5	代理知识小结	186
6.3	创建增强类	187
6.3.1	增强类型	187
6.3.2	前置增强	188
6.3.3	后置增强	192
6.3.4	环绕增强	193
6.3.5	异常抛出增强	194
6.3.6	引介增强	196
6.4	创建切面	199
6.4.1	切点类型	200
6.4.2	切面类型	201
6.4.3	静态普通方法名匹配切面	203
6.4.4	静态正则表达式方法匹配切面	205
6.4.5	动态切面	208
6.4.6	流程切面	211
6.4.7	复合切点切面	213
6.4.8	引介切面	215
6.5	自动创建代理	216
6.5.1	实现类介绍	217

6.5.2	BeanNameAutoProxyCreator	217
6.5.3	DefaultAdvisorAutoProxyCreator	219
6.6	小结	220

第7章 基于@AspectJ 和 Schema 的 AOP 221

7.1	Spring 对 AOP 的支持	222
7.2	JDK 5.0 注解知识快速进阶	222
7.2.1	了解注解	222
7.2.2	一个简单的注解类	223
7.2.3	使用注解	224
7.2.4	访问注解	225
7.3	着手使用@AspectJ	226
7.3.1	使用前的准备	226
7.3.2	一个简单的例子	227
7.3.3	如何通过配置使用@AspectJ 切面	229
7.4	@AspectJ 语法基础	230
7.4.1	切点表达式函数	230
7.4.2	在函数入参中使用通配符	231
7.4.3	逻辑运算符	232
7.4.4	不同增强类型	232
7.4.5	引介增强用法	233
7.5	切点函数详解	235
7.5.1	@annotation()	235
7.5.2	execution()	237
7.5.3	args()和@args()	238
7.5.4	within()	240
7.5.5	@within()和@target()	240
7.5.6	target()的 this()	241
7.6	@AspectJ 进阶	243
7.6.1	切点复合运算	243
7.6.2	命名切点	244
7.6.3	增强织入的顺序	245
7.6.4	访问连接点信息	246
7.6.5	绑定连接点方法入参	247
7.6.6	绑定代理对象	249
7.6.7	绑定类注解对象	249
7.6.8	绑定返回值	250

7.6.9	绑定抛出的异常	251
7.7	基于 Schema 配置切面	252
7.7.1	一个简单切面的配置	252
7.7.2	配置命名切点	253
7.7.3	各种增强类型的配置	255
7.7.4	绑定连接点信息	257
7.7.5	Advisor 配置	258
7.8	混合切面类型	259
7.8.1	混合使用各种切面类型	260
7.8.2	各种切面类型总结	261
7.9	JVM Class 文件字节码转换基础知识	261
7.9.1	java.lang.instrument 包的工作原理	262
7.9.2	如何向 JVM 中注册转换器	263
7.9.3	使用 JVM 启动参数注册转换器的	265
7.10	使用 LTW 织入切面	265
7.10.1	Spring 的 LoadTimeWeaver	266
7.10.2	使用 LTW 织入一个切面	268
7.10.3	在 Tomcat 下的配置	270
7.10.4	在其他 Web 应用服务器下的配置	271
7.11	小结	271

第 3 篇 数据访问

第 8 章	Spring 对 DAO 的支持	274
8.1	Spring 的 DAO 理念	275
8.2	统一的异常体系	275
8.2.1	Spring 的 DAO 异常体系	276
8.2.2	JDBC 的异常转换器	278
8.2.3	其他持久技术的异常转换器	278
8.3	统一数据访问模板	279
8.3.1	使用模板和回调机制	279
8.3.2	Spring 为不同持久化技术所提供的模板类	281
8.4	数据源	282
8.4.1	配置一个数据源	282

8.4.2	获取 JNDI 数据源	287
8.4.3	Spring 的数据源实现类	287
8.5	小结	288

第 9 章 Spring 的事务管理 289

9.1	数据库事务基础知识	290
9.1.1	何为数据库事务	290
9.1.2	数据并发的问题	291
9.1.3	数据库锁机制	293
9.1.4	事务隔离级别	294
9.1.5	JDBC 对事务支持	294
9.2	ThreadLocal 基础知识	296
9.2.1	ThreadLocal 是什么	296
9.2.2	ThreadLocal 的接口方法	297
9.2.3	一个 ThreadLocal 实例	298
9.2.4	与 Thread 同步机制的比较	299
9.2.5	Spring 使用 ThreadLocal 解决线程安全问题	300
9.3	Spring 对事务管理的支持	301
9.3.1	事务管理关键抽象	302
9.3.2	Spring 的事务管理器实现类	305
9.3.3	事务同步管理器	307
9.3.4	事务传播行为	309
9.4	编程式的事务管理	309
9.5	使用 XML 配置声明式事务	310
9.5.1	一个将被实施事务增强的服务接口	312
9.5.2	使用原始的 TransactionProxyFactoryBean	313
9.5.3	基于 tx/aop 命名空间的配置	315
9.6	使用注解配置声明式事务	318
9.6.1	使用 @Transactional 注解	318
9.6.2	通过 AspectJ LTW 引入事务切面	322
9.7	集成特定的应用服务器	323
9.7.1	BEA WebLogic	324
9.7.2	BEA WebLogic	324
9.8	小结	324

第 10 章 Spring 的事务管理难点剖析.....325

10.1 DAO 和事务管理的牵绊.....	326
10.1.1 JDBC 访问数据库.....	326
10.1.2 Hibernate 访问数据库.....	328
10.2 应用分层的迷惑.....	330
10.3 事务方法嵌套调用的迷茫.....	334
10.3.1 Spring 事务传播机制回顾.....	334
10.3.2 相互嵌套的服务方法.....	335
10.4 多线程的困惑.....	338
10.4.1 Spring 通过单实例化 Bean 简化多线程问题.....	338
10.4.2 启动独立线程调用事务方法.....	338
10.5 联合军种作战的混乱.....	340
10.5.1 Spring 事务管理器的应对.....	340
10.5.2 Hibernate+Spring JDBC 混合框架的事务管理.....	341
10.6 特殊方法成漏网之鱼.....	345
10.6.1 哪些方法不能实施 Spring AOP 事务.....	345
10.6.2 事务增强遗漏实例.....	345
10.7 数据连接泄漏.....	349
10.7.1 底层连接资源的访问问题.....	349
10.7.2 Spring JDBC 数据连接泄漏.....	350
10.7.3 通过 DataSourceUtils 获取 数据连接.....	353
10.7.4 通过 DataSourceUtils 获取 数据连接.....	355
10.7.5 JdbcTemplate 如何做到对连接 泄漏的免疫.....	357
10.7.6 使用 TransactionAwareData SourceProxy.....	357
10.7.7 其他数据访问技术的等价类.....	358
10.8 小结.....	359

第 11 章 使用 Spring JDBC 访问数据库.....361

11.1 使用 Spring JDBC.....	362
11.1.1 JdbcTemplate 小试牛刀.....	362
11.1.2 在 DAO 中使用 JdbcTemplate.....	363

11.2 基本的数据操作.....	364
11.2.1 更改数据.....	364
11.2.2 返回数据库的表自增主键值.....	367
11.2.3 批量更改数据.....	369
11.2.4 查询数据.....	370
11.2.5 查询单值数据.....	373
11.2.6 调用存储过程.....	375
11.3 BLOB/CLOB 类型数据的操作.....	377
11.3.1 如何获取本地数据连接.....	377
11.3.2 相关的操作接口.....	379
11.3.3 插入 Lob 类型的数据.....	380
11.3.4 以块数据方式读取 Lob 数据.....	383
11.3.5 以流数据方式读取 Lob 数据.....	383
11.4 自增键和行集.....	384
11.4.1 自增键的使用.....	384
11.4.2 如何规划主键方案.....	386
11.4.3 以行集返回数据.....	388
11.5 其他类型的 JdbcTemplate.....	389
11.5.1 NamedParameterJdbcTemplate.....	389
11.5.2 SimpleJdbcTemplate.....	391
11.6 以 OO 方式访问数据库.....	391
11.6.1 使用 MappingSqlQuery 查询 数据.....	391
11.6.2 使用 SqlUpdate 更新数据.....	393
11.6.3 使用 StoredProcedure 执行存储 过程.....	394
11.6.4 SqlFunction 类.....	396
11.7 小结.....	396

第 12 章 整合其他 ORM 框架.....398

12.1 Spring 整合 ORM 技术.....	399
12.2 在 Spring 中使用 Hibernate.....	400
12.2.1 配置 SessionFactory.....	400
12.2.2 使用 HibernateTemplate.....	403
12.2.3 处理 LOB 类型数据.....	407
12.2.4 添加 Hibernate 事件监听器.....	409
12.2.5 使用原生 Hibernate API.....	409
12.2.6 使用注解配置.....	410

12.2.7	事务处理	412
12.2.8	延迟加载的问题	413
12.3	在 Spring 中使用 myBatis	414
12.3.1	配置 SqlMapClient	415
12.3.2	在 Spring 配置 myBatis	416
12.3.3	编写 myBatis 的 DAO	417
12.5	DAO 层设计	420
12.5.1	DAO 基类的设计	421
12.5.2	查询接口方法的设计	423
12.5.3	分页查询接口设计	424
12.6	小结	425

第 4 篇 业务层及 Web 层技术

第 13 章 任务调度和异步执行器 428

13.1	任务调度概述	429
13.2	Quartz 快速进阶	429
13.2.1	Quartz 基础结构	430
13.2.2	使用 SimpleTrigger	432
13.2.3	使用 CronTrigger	434
13.2.4	使用 Calendar	437
13.2.5	任务调度信息存储	439
13.3	在 Spring 中使用 Quartz	442
13.3.1	创建 JobDetail	442
13.3.2	创建 Trigger	444
13.3.3	创建 Scheduler	446
13.4	Spring 中使用 JDK Timer	447
13.4.1	Timer 和 TimerTask	448
13.4.2	Spring 对 JDK Timer 的支持	450
13.5	Spring 对 JDK 5.0 Executor 的支持	451
13.5.1	了解 JDK 5.0 的 Executor	452
13.5.2	Spring 对 Executor 所提供的抽象	454
13.6	实际应用中的任务调度	455
13.6.1	如何产生任务	456
13.6.2	任务调度对应用程序集群的影响	457
13.6.3	任务调度云	458

13.6.4	Web 应用程序中调度器的启动和关闭问题	460
13.7	小结	462

第 14 章 使用 OXM 进行对象 XML 映射 463

14.1	认识 XML 解析技术	464
14.1.1	什么是 XML	464
14.1.2	XML 的处理技术	464
14.2	XML 处理利器: XStream	466
14.2.1	XStream 概述	466
14.2.2	快速入门	466
14.2.3	使用 XStream 别名	469
14.2.4	XStream 转换器	470
14.2.5	XStream 注解	472
14.2.6	流化对象	474
14.2.7	持久化 API	475
14.2.8	额外功能: 处理 JSON	476
14.3	其他常见 O/X Mapping 开源项目	478
14.3.1	JAXB	478
14.3.2	XMLBeans	482
14.3.3	Castor	485
14.3.4	JiBX	490
14.3.5	总结比较	493
14.4	与 Spring OXM 整合	494
14.4.1	Spring OXM 概述	494
14.4.2	整合 OXM 实现者	496
14.4.3	如何在 Spring 中进行配置	497
14.4.4	Spring OXM 简单实例	499
14.5	小结	501

第 15 章 Spring MVC 503

15.1	Spring MVC 概述	504
15.1.1	体系结构	504
15.1.2	配置 DispatcherServlet	505
15.1.3	一个简单的实例	510
15.2	注解驱动的控制	514
15.2.1	使用 @RequestMapping 映射请求	514
15.2.2	请求处理方法签名概述	518

15.2.3	处理方法签名详细说明	519	16.1.1	为什么需要单元测试	593
15.2.4	使用 <code>HttpMessageConverter<T></code>	523	16.1.2	单元测试之误解	594
15.2.5	处理模型数据	532	16.1.3	单元测试之困境	595
15.3	处理方法的数据绑定	538	16.1.4	单元测试基本概念	596
15.3.1	数据绑定流程剖析	539	16.2	JUnit 4 快速进阶	600
15.3.2	数据转换	539	16.2.1	JUnit 4 概述	600
15.3.3	数据格式化	545	16.2.2	JUnit 4 生命周期	601
15.3.4	数据校验	549	16.2.3	使用 JUnit 4	601
15.4	视图和视图解析器	558	16.3	模拟利器 Mockito	608
15.4.1	认识视图	558	16.3.1	模拟测试概述	608
15.4.2	认识视图解析器	560	16.3.2	创建 Mock 对象	608
15.4.3	JSP 和 JSTL	561	16.3.3	设定 Mock 对象的期望行为及 返回值	609
15.4.4	模板视图	565	16.3.4	验证交互行为	611
15.4.5	Excel	569	16.4	测试整合之王 Unitils	612
15.4.6	PDF	570	16.4.1	Unitils 概述	612
15.4.7	输出 XML	572	16.4.2	集成 Spring	615
15.4.8	输出 JSON	573	16.4.3	集成 Hibernate	618
15.4.9	使用 <code>XmlViewResolver</code>	573	16.4.4	集成 Dbunit	619
15.4.10	使用 <code>ResourceBundle</code> <code>ViewResolver</code>	574	16.4.5	自定义扩展模块	620
15.4.11	混合使用多种视图技术	575	16.5	使用 Unitils 测试 DAO 层	620
15.5	本地化解析	577	16.5.1	数据库测试的难点	621
15.5.1	本地化概述	577	16.5.2	扩展 Dbunit 用 Excel 准备数据	621
15.5.2	使用 <code>CookieLocaleResolver</code>	578	16.5.3	测试实战	624
15.5.3	使用 <code>SessionLocaleResolver</code>	579	16.6	使用 unitils 测试 Service 层	634
15.5.4	使用 <code>LocaleChangeInterceptor</code>	579	16.7	测试 Web 层	639
15.6	文件上传	579	16.7.1	对 <code>LoginController</code> 进行单元 测试	640
15.6.1	配置 <code>MultipartResolver</code>	580	16.7.2	使用 Spring Servlet API 模拟 对象	641
15.6.2	编写控制器和文件上传表单页面	580	16.7.3	使用 Spring RestTemplate 测试	642
15.7	杂项	581	16.7.4	使用 Selenium 测试	644
15.7.1	静态资源处理	581	16.8	小结	647
15.7.2	装配拦截器	586			
15.7.3	异常处理	587			
15.8	小结	589			

第 5 篇 测试及实战

第 16 章 实战型单元测试 592

16.1 单元测试概述 593

第 17 章 实战案例开发 648

17.1 论坛案例概述 649

17.1.1 论坛整体功能结构 649

17.1.2 论坛用例描述 649

17.1.3 主要功能流程描述 651

17.2	系统设计	655	17.6.3	服务类 Bean 的装配	685
17.2.1	技术框架选择	655	17.7	对服务层进行测试	686
17.2.2	Web 目录结构及类包结构规划	656	17.7.1	编写 Service 测试基类	687
17.2.3	单元测试类包结构规划	657	17.7.2	编写 ForumService 测试用例	687
17.2.4	系统的结构图	657	17.8	Web 层开发	689
17.2.5	PO 的类设计	658	17.8.1	BaseController 的基类	689
17.2.6	持久层设计	659	17.8.2	用户登录和注销	691
17.2.7	服务层设计	659	17.8.3	用户注册	692
17.2.8	Web 层设计	660	17.8.4	论坛管理	694
17.2.9	数据库设计	661	17.8.5	论坛普通功能	696
17.3	开发前的准备	663	17.8.6	分页显示论坛版块的主题帖子	698
17.4	持久层开发	664	17.8.7	web.xml 配置	702
17.4.1	PO 类	664	17.8.8	Spring MVC 配置	704
17.4.2	DAO 基类	666	17.9	对 Web 层进行测试	705
17.4.3	通过扩展基类所定义 DAO 类	671	17.9.1	编写 Web 测试基类	705
17.4.4	DAO Bean 的装配	672	17.9.2	编写 ForumManageController 测试用例	706
17.4.5	使用 Hibernate 二级缓存	674	17.10	部署和运行应用	707
17.5	对持久层进行测试	676	17.11	小结	710
17.5.1	配置 Unitils 测试环境	676	以下内容详见本书配书光盘:		
17.5.2	准备测试数据库及测试数据	677	附录 A JavaMail 发送邮件		711
17.5.3	编写 DAO 测试基类	678	附录 B 在 Spring 中开发 Web Service		738
17.5.4	编写 BoardDao 测试用例	678			
17.6	服务层开发	680			
17.6.1	UserService 的开发	680			
17.6.2	ForumService 的开发	682			

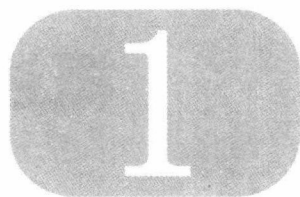
第 1 篇

概述



-
- ◆ 第 1 章 Spring 概述
 - ◆ 第 2 章 快速入门

第1章 Spring 概述



Spring 已经成为 Java 应用首选的 full-stack 开发框架，该框架本着“从实践中来，到实践中去”的原则，对传统 EJB 重量型框架的思想进行了颠覆式的革新，通过 Rod Johnson 天才般的演绎，使 Spring 在短短的时间内就成为用户众多、社群庞大、文档丰富、极具实用性的开源开发框架。Spring 拥有一支活跃的开发队伍，功能的改进和版本的升级从来没有停止过。目前，Spring 已经升级到 3.0 版本，Spring 3.0 充分拥抱 Java 5.0 技术，核心 API 对泛型、可变入参等语言特性提供了全面的支持。Bean 的配置得到了重大的升级，更加灵活和强大。此外还新增了 SpEL、OXM、通用类型转换和属性格式化等功能，全面支持 REST 风格的 Web 开发。Spring 家族系列的子项目更加丰富，特别是 Spring 在云计算、OSGI 这些引领世界风潮的技术上拥有令人惊羡的表现。本章对 Spring 框架进行宏观性的概述，力图使读者建立起对 Spring 整体性的认识。

本章主要内容：

- ◆ 认识 Spring
- ◆ Spring 体系及文档结构
- ◆ Spring 3.0 新特性
- ◆ Spring 的子项目
- ◆ 相似框架的介绍

本章亮点：

- ◆ 对 Spring 3.0 新特性进行分析
- ◆ 详细介绍 Spring 的第三方依赖类库
- ◆ 对 Spring 子项目进行点评