

Pro Spring 2.5

Spring高级程序设计

Jan Machacek

Aleksa Vukotic

[英]

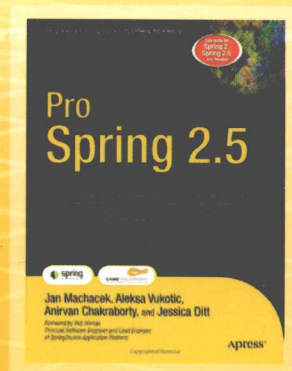
Anirvan Chakraborty

著

Jessica Ditt

马连浩 栗磊 张龙 张俊 译

- Spring框架创始人倾情推荐的权威开发指南
- 全面揭示Spring框架关键技术
- 深入了解Spring内部工作机制



人民邮电出版社
POSTS & TELECOM PRESS

TURING 图灵程序设计丛书 Java 系列



Pro Spring 2.5 Spring高级程序设计

Jan Machacek
[英] Aleksa Vukotic 著
Anirvan Chakraborty
Jessica Ditt

马连浩 栗磊 张龙 张俊 译

人民邮电出版社
北 京

图书在版编目(CIP)数据

Spring 高级程序设计 / (英) 马哈切克 (Machacek, J.)
等著; 马连浩等译. —北京: 人民邮电出版社, 2009.9
(图灵程序设计丛书)
书名原文: Pro Spring 2.5
ISBN 978-7-115-21204-7

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

中国版本图书馆CIP数据核字(2009)第143911号

内 容 提 要

本书对 Spring 开发框架的核心概念及关键模块进行了深入剖析, 并结合示例展示了它们的具体应用。本书共分 4 部分。第一部分介绍 Spring 框架的基础知识, 重点阐述了 AOP 和 IOC 的基本概念; 第二部分介绍 Spring 对数据持久化的支持, 从基础的 JDBC 开始, 讲解了 Spring 对应用最广泛的 ORM 框架 iBATIS 和 Hibernate 的支持; 第三部分阐述了 Spring 企业级组件, 介绍了任务调度、Mail 服务、动态语言等企业级组件相关的知识; 第四部分阐述如何使用 Spring 进行 Java EE 5 开发, 从远程调用和事务管理开始, 接着介绍了 Spring MVC、Web 工作流、AJAX、JMX、单元测试以及性能调优等 Java EE 相关开发知识。

本书由 Spring 框架核心开发人员撰写, 适合各层次 Java 开发人员, 特别是企业级 Java 开发人员阅读参考。

图灵程序设计丛书

Spring高级程序设计

- ◆ 著 [英] Jan Machacek Aleksa Vukotic
Anirvan Chakraborty Jessica Ditt
译 马连浩 栗磊 张龙 张俊
责任编辑 傅志红
执行编辑 马晓燕
- ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街14号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京鑫正大印刷有限公司印刷
- ◆ 开本: 800×1000 1/16
印张: 48.75
字数: 1304千字 2009年9月第1版
印数: 1—3 000册 2009年9月北京第1次印刷

著作权合同登记号 图字: 01-2009-3806号

ISBN 978-7-115-21204-7/TP

定价: 99.00元

读者服务热线: (010)51095186 印装质量热线: (010)67129223

反盗版热线: (010)67171154

版 权 声 明

Original English language edition, entitled *Pro Spring 2.5* by Jan Machacek, Aleksa Vukotic, Anirvan Chakraborty, and Jessica Ditt, published by Apress, 2855 Telegraph Avenue, Suite 600, Berkeley, CA 94705 USA.

Copyright © 2008 by Jan Machacek, Aleksa Vukotic, Anirvan Chakraborty, and Jessica Ditt. Simplified Chinese-language edition copyright © 2009 by Posts & Telecom Press. All rights reserved.

本书中文简体字版由Apress L.P.授权人民邮电出版社独家出版。未经出版者书面许可，不得以任何方式复制或抄袭本书内容。

版权所有，侵权必究。

感谢Marc，他仍认为bean（豆）是长在树上的。

——Jan

感谢我的父母Andja和Slobodan，他们指导我的人生，鼓励我走自己的路。

——Aleksa

感谢我的父母Sujata和Kalyan，谢谢他们给予了我无私的爱和支持。

——Anirvan

感谢Ioannis，在我最需要支持的时候不断地鼓励我。并且感谢我的家人，只因为他们是我的家人。

——Jessica

译者序

随着Spring框架的发展和逐渐成熟，它与Hibernate、Struts一起构成了国内Java Web开发的最佳组合。与EJB相比，Spring是一个轻量级的解决方案，用于快速建立企业级应用。Spring框架在企业开发中的大量应用，也催生了相关书籍的出版。本书是由Spring框架核心开发团队成员撰写的经典之作，为想要及时了解Spring技术趋势的读者提供了权威参考。

本书第一版自出版以来，深受读者喜爱，常年位居畅销书排行榜前列。第二版在保留第一版特点的同时，紧跟Spring框架的发展趋势，精心修订了原有内容并进行了大量扩充，例如新增了Spring对于AJAX技术和动态语言支持的介绍。本书基于Spring框架2.5版本，新版本的Spring框架全面支持Java 6和Java EE5，同时对原有功能进行了大量增强，例如基于注解的Spring MVC和已修订的测试框架。

本书的翻译工作由Redsaga (<http://www.redsaga.com/>) 翻译小组中的4名成员完成，各人具体负责的章节如下：张俊翻译前言和第1、7、8、20、22章，栗磊翻译第2~6章，马连浩翻译第9~14章，张龙翻译第15~19、21章。全书最后由马连浩统稿。另外，本书部分翻译内容参考了满江红网站的开放文档《Spring中文参考手册》（包含其在线版及可下载版的链接为http://bit.ly/Spring25_Ref_cn），在此表示感谢。

译者特别感谢人民邮电出版社图灵公司的编辑们，正是她们的积极推动和严谨工作保证了本书的顺利出版。

最后，祝读者能借助本书深入了解Spring框架，开发出更优秀的企业级应用。

作者简介

Jan Machacek Cake解决方案的首席软件架构师，在所有项目的架构讨论中他都是核心人物。除了软件架构，Jan也经常为实现一些最复杂和最具挑战性的功能提供帮助。自从加入了Cake项目以来，Jan不但参与了各种各样复杂的项目，还出版了不少技术专著，证明了他Java开发领域的高深造诣。闲暇之余，Jan热衷于自行车运动，是曼彻斯特自行车俱乐部会员。不管是计时赛还是公路赛，他都全力争胜！

Jan负责第1~4、6、9、11、14、16、22章的撰写。

Aleksa Vukotic Cake解决方案的软件架构师，负责大部分Cake项目的架构及实现。他在Java EE方面，特别是在Spring MVC应用开发和安全领域有着丰富的经验。他掌握了很多技术诀窍，这让他几乎可以解决所有技术难题。他还是一位出色的领导者，率领Cake解决方案开发团队，帮助团队解决最棘手的问题。除了醉心于Java和.NET平台外，他还喜欢体育、音乐和夜生活。他与Jan一起负责Cake解决方案的所有主要项目。

Aleksa负责第5、8、10、15、17、21章的撰写。

Anirvan Chakraborty Cake解决方案的高级开发工程师。由于在Spring框架方面有着丰富的经验并对细节实现有深入了解，他负责实现一些具有挑战性的项目。Anirvan为他的代码感到自豪，始终都确保他的代码能够作为范例示人。除了爱捣弄Java EE和Linux之外，他喜欢跟朋友们一起聚会，享用美食。他还热衷于板球，喜欢阅读侦探小说。

Anirvan负责第7、13、19、20章的撰写。

Jessica Ditt 自2008年初开始成为伦敦Avenue A | Razorfish公司的开发人员。在此之前她为Cake解决方案工作了两年半。她主要负责一些企业级项目，所有项目都用到了本书所介绍的Spring框架和Spring Web Flow。Jessica掌握了大量的专业知识，擅长使用Lucene获取更高效率的索引，使用Gigaspaces有效地提升Java EE应用的可扩展性。工作之余，Jessica热衷于排球运动和健身。

Jessica负责第12、18章的撰写。

序

我最终决定不参加编写本书时，心情万分沉重。在这里我要感谢Jan，是他挺身而出，组织编写了本书。在过去三年中，*Pro Spring*已成为我生活的重要组成部分，最后放弃绝不是挥一挥衣袖那么轻松。由于我和Juergen还在为Spring框架2.0努力，我知道我没有时间参与本书的写作，毕竟软件开发与写作不能同时进行。幸运的是，Jan帮我完成了这个心愿。

当Jan和Apress出版社还计划着发布*Pro Spring 2.0*时，我和Juergen已经对Spring框架进行了很多的变更，无意中导致他们很难跟上Spring的变化。我至今清楚地记得，在更新所有的JSP表单标签时我有些难为情，我知道我给Jan增加了多少工作量。

Spring 2.5刚发布，Jan就做出了一个明智的选择，放弃Spring 2.0而采用Spring 2.5。Spring框架2.5体现了Spring框架和企业级Java框架的最新进展。对所有勤勉认真的Java开发人员来说，这本关于这个重要工具的专业指南应该是必读的。

我至今仍记得，在Cake解决方案起始，我们决定再聘请一个开发人员。总的来说我们缺乏招聘的经验，招程序员更是难上加难。当时我们只是想招到个大学毕业生就不错了，但是没有想到就此找来的Jan是这么能干。

在开始工作的第一个星期，他从头编写了一个桌面邮件程序包，顺利地投入使用。在过去的5年中，Jan一直都是大部分Cake子项目的核心人物，其中多为基于Spring框架的大型Java产品。他的Spring知识来自于极其丰富的实践经验：自Spring 1.0版本以来，Jan一直在实际的项目中使用Spring并构建了成功的系统。

令我赞赏的是，Jan意识到编写本书的工作量很大，仅靠他一个人不能完成，所以他组织了Cake解决方案组的其他成员来协助他。我感到格外兴奋，有这样一群真正的程序员、一群真正体会到Spring精髓的人来与大家分享知识，毫无疑问，很多人都会发现本书是不可缺少的参考书。

因此，尽管无法参与本书的编写有些遗憾，但是我仍很高兴，Jan出色地完成了许多人一直都在期盼的*Pro Spring*的更新版本。

希望大家尽情享受阅读的乐趣！

Rob Harrop

(SpringSource核心开发人员兼首席工程师)

前言

最近，Java业界发生了巨变，从如EJB（Enterprise JavaBean）的“重量级”架构体系向如Spring的轻量级框架迁移。Hibernate和AOP（面向方面编程）等更简单的实现机制代替了复杂并且依赖容器的服务（如CMP）和事务管理系统。Spring核心层提供了一个完善的基于控制反转（IoC）的轻量级容器，用户可以使用它建立自己的应用程序。在容器上，Spring提供了大量实用的服务，将很多高质量开源项目集成到统一的框架上。

Spring框架的品质已经显现，它在很多应用中已经取代了传统的Java EE架构。因此，越来越多的开发者认识到需要对Spring技术有一个全面的了解。尽管Spring有着相当丰富的文档资源和案例，但我们还是发现，仍然有许多开发者为了解如何使用Spring，特别是如何有效地运用Spring而苦苦探究。因此，我们决定为*Pro Spring*写一个新版本。

起初，我们以为只需要在原有基础上更新几章即可推出*Pro Spring 2.5*。然而，我们很快就意识到，Spring 2.5带来了许多新的功能和改进，虽然我们一直保持着*Pro Spring*的名字，但这却是一本全新的书。

通过本书，你将学习如何使用Spring来建立更好的Web应用和单机应用，如何在框架中进行恰当的选择。我们的目标是为你提供全面的知识，帮助你在自己的应用程序中有效地使用Spring，同时让你更深入了解Spring的机制。

例如，你可以：

- 在AOP上下文中了解控制反转的相关基础知识；
- 通过查看简单易懂的应用实例，了解Spring的强大功能和无缝集成能力；
- 学习如何使用包括基于AOP的Spring声明式事务管理框架在内的Spring特性替换常用的EJB特性；
- 使用Spring内置的JMX引擎高效管理Spring组件和应用；
- 了解如何使用Quartz在Spring应用中实现任务调度。

阅读完本书后，你将获得使用Spring及其相关开源项目构建高效应用的所有知识。

致谢

任何书籍的编写都需要进行大量的幕后工作，在作者身后，是一支由编辑、校对和技术审稿人组成的杰出团队。本书也不例外，我们要感谢所有为本书付出心血的人。我们要感谢Rick，本书的技术审稿人，他尽心尽力的工作保证了本书的质量。感谢Apress出类拔萃的编辑团队，特别是Sofia Marchant、Tom Welsh编辑、Heather Lang、Kelly Winkvist等人。如果没有他们的帮助，我们无法完成本书。同时，我们还要感谢Rob Harrop为本书作序。最后，我衷心地感谢Cake解决方案的总经理Guy Remond，是他让我们的团队可以占用一些工作时间撰写本书！

目 录

第一部分 Spring 入门

第 1 章 Spring 简介 2

- 1.1 Spring是什么 2
 - 1.1.1 依赖注入之外的特性 4
 - 1.1.2 使用Spring进行面向方面编程 4
 - 1.1.3 数据访问 5
 - 1.1.4 简化与整合Java EE 5
 - 1.1.5 基于Spring的任务调度 6
 - 1.1.6 Spring对邮件的支持 6
 - 1.1.7 动态语言 6
 - 1.1.8 远程访问 6
 - 1.1.9 事务管理 7
 - 1.1.10 Spring MVC框架 7
 - 1.1.11 Spring Web Flow 7
 - 1.1.12 AJAX技术 7
 - 1.1.13 国际化 8
 - 1.1.14 简化异常处理 8
- 1.2 Spring项目 8
 - 1.2.1 Spring的起源 8
 - 1.2.2 Spring .NET 8
 - 1.2.3 Spring IDE 9
 - 1.2.4 Spring安全系统 (原Acegi) 9
 - 1.2.5 Spring的替代方案 9
- 1.3 示例代码 10
- 1.4 小结 10

第 2 章 Spring 入门 11

- 2.1 获取Spring框架 11
 - 2.1.1 从CVS上签出Spring 11
 - 2.1.2 从源码构建Spring 12

- 2.1.3 检查Spring发布包 13
- 2.1.4 Spring发布包 13
- 2.1.5 Spring依赖的包 14
- 2.1.6 示例应用 16
- 2.1.7 配置Spring和IDE 18
- 2.2 HelloWorld示例 20
- 2.3 Spring版本的“Hello, World”示例 23
 - 2.3.1 依赖注入 24
 - 2.3.2 Spring的侵入性 26
- 2.4 小结 26

第 3 章 控制反转 27

- 3.1 控制反转和依赖注入 27
- 3.2 控制反转的类型 27
 - 3.2.1 上下文依赖查找 29
 - 3.2.2 构造方法依赖注入 30
 - 3.2.3 设置方法依赖注入 31
 - 3.2.4 依赖注入与依赖查找 32
 - 3.2.5 设置方法注入与构造方法注入 33
- 3.3 Spring中的控制反转 34
- 3.4 基于Spring的依赖注入 34
 - 3.4.1 bean和BeanFactory 34
 - 3.4.2 BeanFactory实现 35
 - 3.4.3 XML bean定义 36
 - 3.4.4 构造方法注入 37
 - 3.4.5 注入参数 40
 - 3.4.6 理解bean的命名 50
 - 3.4.7 bean的实例化模式 52
 - 3.4.8 解析依赖 55
 - 3.4.9 bean的自动装配 57
 - 3.4.10 依赖检查 61

3.4.11 bean的继承	62	4.10 小结	134
3.5 小结	64	第5章 Spring AOP 基础	135
第4章 进阶	66	5.1 AOP概念	136
4.1 Spring对应用程序可移植性的影响	67	5.2 AOP的类型	136
4.2 管理bean的生命周期	67	5.2.1 静态AOP	137
4.2.1 嵌入bean的创建	68	5.2.2 动态AOP	137
4.2.2 嵌入bean的销毁	74	5.2.3 选择一种AOP类型	137
4.3 让bean可被Spring感知	79	5.3 Spring中的AOP	137
4.3.1 使用BeanNameAware接口	80	5.3.1 AOP联盟	138
4.3.2 使用BeanFactoryAware接口	81	5.3.2 AOP的Hello World	138
4.4 使用方法注入 (method injection)	83	5.3.3 Spring AOP架构	140
4.4.1 查找方法注入	83	5.3.4 ProxyFactory类	140
4.4.2 方法替换	88	5.3.5 在Spring中创建通知	141
4.5 使用FactoryBean接口	91	5.4 Spring里的通知者和切入点	155
4.5.1 MessageDigestFactoryBean类	91	5.4.1 切入点接口	156
4.5.2 直接访问FactoryBean	94	5.4.2 使用ComposablePointcut	172
4.6 BeanFactoryPostProcessor类	94	5.4.3 切入点总结	175
4.7 JavaBean的属性修改器	101	5.5 代理详解	176
4.7.1 内建的PropertyEditor	101	5.5.1 理解代理	176
4.7.2 创建自定义PropertyEditor	104	5.5.2 使用JDK动态代理	176
4.8 BeanPostProcessor类	108	5.5.3 使用CGLIB代理	177
4.8.1 实现一个BeanPostProcessor	110	5.5.4 两者的性能比较	177
4.8.2 使用BeanPostProcessor类的实 机选择	114	5.5.5 选用代理	180
4.9 Spring ApplicationContext	115	5.6 小结	180
4.9.1 ApplicationContext的实现类	115	第6章 AOP 进阶	182
4.9.2 使用ApplicationContext- Aware	116	6.1 @AspectJ注解	182
4.9.3 控制bean的初始化	117	6.2 @AspectJ方面详解	186
4.9.4 使用基于注解的配置	118	6.2.1 切入点	186
4.9.5 使用MessageSource进行 国际化	123	6.2.2 切入点表达式	189
4.9.6 在独立应用中使用 MessageSource	130	6.2.3 探讨切入点表达式	191
4.9.7 MessageSourceResolvable 接口	130	6.2.4 在XML中使用@Pointcuts	194
4.9.8 使用应用程序事件	130	6.2.5 通知的类型	194
4.9.9 对于事件用法的考虑	132	6.2.6 参数绑定	201
4.9.10 访问资源	133	6.2.7 引入	202
		6.2.8 方面的生命周期	208
		6.3 AOP的框架服务	209
		6.3.1 使用AOP命名空间创建第一个 方面	209

6.3.2 AOP命名空间中的切入点	211	8.3.1 分层设计	265
6.3.3 使用AOP命名空间创建通知	212	8.3.2 高性能分页	266
6.3.4 AOP命名空间中的引入	217	8.3.3 多错误报告	268
6.4 风格选择	219	8.3.4 用户界面事务	271
6.5 使用Spring AOP代理	220	8.3.5 后台进程	274
6.6 AspectJ集成	225	8.3.6 邮件通知	278
6.6.1 创建第一个AspectJ方面	225	8.3.7 错误收集和日志	280
6.6.2 编译示例程序	227	8.4 小结	283
6.6.3 AspectJ方面的作用域	229		
6.7 加载时织入	230	第二部分 数据访问	
6.7.1 第一个加载时织入示例	230	第9章 Spring对JDBC的支持	286
6.7.2 LoadTimeWeaver的查找策略	232	9.1 JDBC的主要概念	286
6.8 AOP实践	232	9.1.1 使用DriverManager和 Connection	288
6.9 小结	235	9.1.2 使用PreparedStatement	289
第7章 Spring schema与命名空间	236	9.1.3 使用CallableStatement类	292
7.1 新配置的缘由	236	9.1.4 其他JDBC概念	293
7.2 Spring 2.5包含的schema	238	9.2 Spring对数据访问支持的概念	293
7.2.1 beans schema	238	9.3 Spring对JDBC数据访问的支持	294
7.2.2 上下文schema	239	9.4 使用JdbcTemplate类	294
7.2.3 util schema	239	9.4.1 JdbcTemplate类的execute 方法	296
7.2.4 tx schema	242	9.4.2 JdbcTemplate类的query方法 和该方法的扩展	299
7.2.5 aop schema	242	9.4.3 JdbcTemplate类的update方法	303
7.2.6 jee schema	242	9.4.4 JdbcTemplate类的batchUpdate 方法	304
7.2.7 lang schema	243	9.5 RdbmsOperation子类	305
7.3 schema背后	244	9.5.1 SqlUpdate子类	306
7.4 自定义schema	246	9.5.2 BatchSqlUpdate子类	311
7.5 IDE配置	249	9.5.3 SqlCall类和StoredProcedure 子类	312
7.6 小结	252	9.5.4 SqlQuery类和它的子类	314
第8章 Spring模式	253	9.5.5 JdbcTemplate类和RdbmsOperation 类的比较	321
8.1 目录结构	253	9.6 大二进制对象	321
8.1.1 简单应用	253	9.7 JdbcDaoSupport类	324
8.1.2 复杂应用	255	9.8 简单的Spring JDBC	326
8.1.3 打包和命名	255	9.8.1 SimpleJdbcTemplate类	326
8.2 设计模式简介	255	9.8.2 SimpleJdbcCall类	329
8.2.1 面向接口编程	256		
8.2.2 创建模式	256		
8.2.3 结构模式	260		
8.2.4 行为模式	262		
8.3 Spring应用模式	264		

9.8.3 SimpleJdbcInsert类	331
9.8.4 SimpleJdbcDaoSupport类	332
9.9 小结	333
第10章 集成 iBATIS	334
10.1 iBATIS简述	334
10.1.1 iBATIS版本	334
10.1.2 基础架构和配置	335
10.2 映射文件	335
10.2.1 sqlMap文件	337
10.2.2 配置iBATIS和Spring	339
10.3 查询数据	341
10.3.1 简单查询操作	341
10.3.2 一对一查询操作	344
10.3.3 一对多查询操作	348
10.3.4 多对多查询操作	350
10.4 更新数据	350
10.5 删除数据	353
10.6 插入数据	354
10.7 iBATIS缺少的特性	356
10.8 整体性能	357
10.9 小结	358
第11章 Spring 对 Hibernate 的支持	359
11.1 Hibernate入门	359
11.2 Hibernate打包	360
11.3 Hibernate支持的介绍	361
11.3.1 使用Hibernate Session	363
11.3.2 使用HibernateDaoSupport类	366
11.3.3 HibernateTemplate和Session 间的选择	368
11.4 在企业级应用程序中使用 Hibernate	372
11.4.1 阻止更新脏数据	372
11.4.2 对象等价性	375
11.4.3 事务支持	378
11.4.4 延迟加载	382
11.5 处理大数据集	392
11.6 处理大对象	394
11.7 使用Hibernate处理其他DAO代码	397
11.8 小结	398

第三部分 企业级应用组件

第12章 基于 Spring 的任务调度	400
12.1 使用JDK Timer调度任务	401
12.1.1 Timer触发器类型	401
12.1.2 创建一个简单任务	401
12.1.3 Spring对JDK Timer调度的 支持	404
12.2 使用OpenSymphony Quartz来调度 任务	409
12.2.1 Quartz简介	410
12.2.2 Spring对Quartz的支持	418
12.3 任务调度时需考虑的因素	423
12.3.1 选择一个调度器	423
12.3.2 剥离Job类中的任务逻辑	424
12.3.3 任务执行和线程池	424
12.4 小结	428
第13章 Spring 的邮件支持	429
13.1 Spring Mail API结构	430
13.2 发送简单的电子邮件	430
13.2.1 编程式构造和发送电子邮件	431
13.2.2 声明式构造电子邮件	433
13.3 构造并发送MIME消息	436
13.3.1 发送基本的HTML消息	438
13.3.2 发送带有内嵌图片的HTML 消息	439
13.3.3 发送带有附件的消息	441
13.3.4 发送带有可替换纯文本的 HTML消息	442
13.3.5 发送复杂MIME消息	445
13.4 深入了解企业级电子邮件处理	449
13.5 小结	458
第14章 动态语言	459
14.1 支持的动态语言概览	459
14.1.1 BeanShell	459
14.1.2 Groovy	461
14.1.3 JRuby	462
14.2 使用动态语言定义Spring bean	463
14.2.1 动态语言支持的幕后	465

14.2.2 代理动态语言bean	466	15.7 小结	517
14.2.3 性能	466	第 16 章 事务管理	518
14.3 可刷新bean (refreshable bean)	468	16.1 Spring事务抽象层简介	518
14.4 基于BeanShell的bean	470	16.2 分析事务属性	519
14.5 基于JRuby的bean	472	16.2.1 探索TransactionDefinition 接口	519
14.6 基于Groovy的bean	473	16.2.2 使用TransactionStatus接口	520
14.7 动态语言在Spring程序中的典型 应用	473	16.2.3 PlatformTransactionManager 的实现	521
14.8 小结	477	16.3 对一个事务管理示例的探索	521
第四部分 Java EE 5 应用开发		16.4 程式事务管理	529
第 15 章 Spring 远程访问	480	16.4.1 使用TransactionTemplate类	531
15.1 Spring Remoting架构	481	16.4.2 程式事务管理小结	532
15.2 远程方法调用	482	16.5 声明性事务管理	532
15.2.1 开放任意的服务	482	16.5.1 使用TransactionProxy- FactoryBean	532
15.2.2 通过代理访问RMI服务	484	16.5.2 在事务管理中使用代理的含 义	534
15.2.3 开放CORBA服务	486	16.6 AOP事务管理	535
15.2.4 访问CORBA服务	488	16.6.1 使用基于注解的AOP事务管 理	535
15.3 使用JAX-RPC实现Web服务	490	16.6.2 使用XML AOP事务管理	537
15.3.1 Apache Axis简介	490	16.6.3 tx:advice标签简介	538
15.3.2 使用ServletEndpoint- Support创建Web服务	490	16.6.4 XML AOP小结	539
15.3.3 使用代理访问RPC风格的 Web服务	494	16.7 在多个事务性资源上使用事务	540
15.3.4 与Axis服务的JavaBean交互	497	16.8 实现你自己的事务同步	541
15.4 使用JAX-WS Web服务	500	16.9 小结	548
15.4.1 通过SimpleJaxWsService- Exporter公开Web服务	501	第 17 章 基于 Spring MVC 的 Web 应用 开发	549
15.4.2 使用XFire公开Web服务	501	17.1 MVC架构	549
15.4.3 访问JAX-WS Web服务	503	17.2 Spring MVC介绍	550
15.4.4 从其他客户端访问Java Web 服务	504	17.3 使用处理器映射	551
15.5 使用HTTP Invoker创建Web服务	507	17.4 Spring控制器	553
15.5.1 开发简单服务	508	17.4.1 AbstractController类	554
15.5.2 通过代理访问HTTP Invoker 服务	510	17.4.2 ParameterizableView- Controller类	555
15.5.3 在HTTP Invoker服务中使用 任意对象	511	17.4.3 MultiActionController类	555
15.5.4 使用HTTP基本认证	513	17.5 拦截器	558
15.6 远程架构的选择	516	17.6 视图、本地化和主题	559

17.6.1 以编程的方式使用视图	559	18.2 Hello, Web Flow!	642
17.6.2 使用视图解析器	561	18.3 探索状态	646
17.6.3 使用本地化消息	565	18.3.1 view状态	647
17.6.4 使用Locale	565	18.3.2 decision状态	647
17.6.5 使用主题	565	18.3.3 end状态	648
17.7 命令控制器	567	18.4 处理转换	648
17.7.1 使用表单控制器	568	18.5 高级概念	650
17.7.2 探索AbstractWizardForm- Controller 类	574	18.5.1 表达式语言和范围	650
17.7.3 文件上传	578	18.5.2 实现action	653
17.8 处理异常	581	18.5.3 模型数据绑定	654
17.9 Spring与其他Web技术	583	18.5.4 局部视图刷新	658
17.9.1 使用JSP	583	18.5.5 映射流的输入与输出参数	658
17.9.2 使用Velocity	600	18.5.6 使用子流	659
17.9.3 FreeMarker	604	18.6 幕后的Spring Web Flow	660
17.9.4 使用XSLT视图	608	18.6.1 流执行架构	660
17.9.5 使用PDF视图	609	18.6.2 流执行者	662
17.9.6 实现PDF视图	610	18.6.3 流定义注册表	662
17.9.7 使用Excel视图	611	18.6.4 流执行仓库	665
17.9.8 使用Tiles	613	18.7 集成Spring MVC	666
17.9.9 JasperReports报表引擎	623	18.7.1 流处理	666
17.10 Spring的约定优于配置	627	18.7.2 视图解析	667
17.10.1 控制器约定	628	18.8 使用Spring Security构建安全的流	668
17.10.2 MultiActionController 约定	628	18.8.1 第一步: 添加SecurityFlow- ExecutionListener	668
17.10.3 模型约定	629	18.8.2 第二步: 处理基本的认证和 授权	668
17.10.4 视图约定	630	18.8.3 第三步: 在流定义中定义安 全规则	670
17.11 使用注解配置控制器	631	18.9 问题解决	672
17.11.1 @Controller注解	631	18.9.1 带状态的导航控制	672
17.11.2 @RequestMapping注解	632	18.9.2 浏览器导航栏支持和双提交 (double submit)	672
17.11.3 @RequestParam注解	633	18.10 测试流定义	673
17.11.4 @ModelAttribute注解	633	18.11 小结	674
17.11.5 使用注解配置命令控制器	634	第18章 Spring Web Flow	636
17.12 小结	635	18.1 Spring Web Flow简介	637
第18章 Spring Web Flow	636	18.1.1 核心概念	637
18.1 Spring Web Flow简介	637	18.1.2 获取Spring Web Flow	640
18.1.1 核心概念	637	18.1.3 Spring Web Flow依赖	642
18.1.2 获取Spring Web Flow	640	18.2 Hello, Web Flow!	642
18.1.3 Spring Web Flow依赖	642	18.3 探索状态	646
18.2 Hello, Web Flow!	642	18.3.1 view状态	647
18.3 探索状态	646	18.3.2 decision状态	647
18.3.1 view状态	647	18.3.3 end状态	648
18.3.2 decision状态	647	18.4 处理转换	648
18.3.3 end状态	648	18.5 高级概念	650
18.4 处理转换	648	18.5.1 表达式语言和范围	650
18.5 高级概念	650	18.5.2 实现action	653
18.5.1 表达式语言和范围	650	18.5.3 模型数据绑定	654
18.5.2 实现action	653	18.5.4 局部视图刷新	658
18.5.3 模型数据绑定	654	18.5.5 映射流的输入与输出参数	658
18.5.4 局部视图刷新	658	18.5.6 使用子流	659
18.5.5 映射流的输入与输出参数	658	18.6 幕后的Spring Web Flow	660
18.5.6 使用子流	659	18.6.1 流执行架构	660
18.6 幕后的Spring Web Flow	660	18.6.2 流执行者	662
18.6.1 流执行架构	660	18.6.3 流定义注册表	662
18.6.2 流执行者	662	18.6.4 流执行仓库	665
18.6.3 流定义注册表	662	18.7 集成Spring MVC	666
18.6.4 流执行仓库	665	18.7.1 流处理	666
18.7 集成Spring MVC	666	18.7.2 视图解析	667
18.7.1 流处理	666	18.8 使用Spring Security构建安全的流	668
18.7.2 视图解析	667	18.8.1 第一步: 添加SecurityFlow- ExecutionListener	668
18.8 使用Spring Security构建安全的流	668	18.8.2 第二步: 处理基本的认证和 授权	668
18.8.1 第一步: 添加SecurityFlow- ExecutionListener	668	18.8.3 第三步: 在流定义中定义安 全规则	670
18.8.2 第二步: 处理基本的认证和 授权	668	18.9 问题解决	672
18.8.3 第三步: 在流定义中定义安 全规则	670	18.9.1 带状态的导航控制	672
18.9 问题解决	672	18.9.2 浏览器导航栏支持和双提交 (double submit)	672
18.9.1 带状态的导航控制	672	18.10 测试流定义	673
18.9.2 浏览器导航栏支持和双提交 (double submit)	672	18.11 小结	674
18.10 测试流定义	673	第19章 Spring 与 AJAX	675
18.11 小结	674	19.1 DWR	675
第19章 Spring 与 AJAX	675	19.2 安装DWR	676
19.1 DWR	675	19.3 针对DWR的Spring配置	676
19.2 安装DWR	676	19.4 关于完整示例	677
19.3 针对DWR的Spring配置	676		
19.4 关于完整示例	677		

19.5	测试DWR配置	682	21.2	单元测试	711
19.6	运行完整示例	682	21.3	集成测试	715
19.7	DWR脚本基础	683	21.3.1	使用AbstractSpring- ContextTests	722
19.7.1	使用简单的回调函数	683	21.3.2	使用AbstractDependency- InjectionSpringContext- Tests类	723
19.7.2	调用元数据对象	684	21.3.3	使用AbstractTransac- tionalSpringContextTest	726
19.8	engine.js文件	684	21.3.4	使用AbstractAnnotation- AwareTransactionalTest	728
19.8.1	调用批处理	684	21.3.5	JNDI	731
19.8.2	调用顺序	684	21.4	Spring TestContext Framework	733
19.8.3	错误和警告处理	684	21.4.1	应用上下文及使用 TestContext Framework进行 依赖注入	733
19.9	util.js脚本	685	21.4.2	使用TestContext Framework 进行依赖注入	735
19.10	DWR的安全性	685	21.4.3	TestContext Framework中的 事务	736
19.11	DWR的优缺点	686	21.4.4	支持类	738
19.12	小结	687	21.5	测试覆盖	740
第 20 章	使用 Spring JMX	688	21.6	小结	740
20.1	JMX进阶	688	第 22 章	Spring 性能调优	741
20.2	开放Java Bean	689	22.1	性能与响应	741
20.2.1	MBeanExporter类	689	22.2	探索企业应用的性能问题	741
20.2.2	MBeanServer FactoryBean类	691	22.3	测量Java EE应用的性能	742
20.2.3	在已有的MBean服务器上开 放Bean	692	22.3.1	测量目标	742
20.2.4	Bean注册行为	692	22.3.2	确定必要的数据集	747
20.3	控制对象名	693	22.3.3	改善数据访问层	748
20.4	控制管理接口	693	22.3.4	提高事务管理	757
20.4.1	MBeanInfoAssembler接口	693	22.3.5	控制远程调用的性能	758
20.4.2	MethodNameBasedMBean- InfoAssembler接口	694	22.3.6	了解视图性能	758
20.4.3	用Java接口来控制管理接口	696	22.4	使用缓存	759
20.4.4	使用源码级元数据	698	22.5	性能测试	760
20.5	远程使用Spring JMX	701	22.6	监控应用程序的状态和性能	762
20.5.1	开放远程MBean	701	22.7	更多的性能调优资源	764
20.5.2	访问远程MBean	702	22.8	小结	764
20.5.3	代理MBean	702			
20.6	Spring JMX通知	703			
20.6.1	通知监听器	703			
20.6.2	发布通知	704			
20.7	小结	708			
第 21 章	使用 Spring 进行测试	709			
21.1	进行单元测试	709			

Part 1

第一部分

Spring 入门

在本部分中，我们从下载相应的二进制文件和源代码开始来介绍 Spring 框架。我们会解释控制反转（IoC）模式和 AOP（面向方面编程）。控制反转和 AOP 是 Spring 框架的核心。在第一部分结束时，你将详细了解到配置 Spring 管理 bean 的各种方法。最后，我们还会了解一些重要的 Spring 设计模式。

本 部 分 内 容

- 第 1 章 Spring 简介
- 第 2 章 Spring 入门
- 第 3 章 控制反转
- 第 4 章 进阶
- 第 5 章 Spring AOP 基础
- 第 6 章 AOP 进阶
- 第 7 章 Spring schema 与命名空间
- 第 8 章 Spring 模式