

TURING

图灵程序设计丛书 Java 系列

Apress®

Spring Recipes A Problem-Solution Approach

Spring 攻略

Gary Mak 著

丁雪丰 荣浩 译
杨春花 俞黎敏

丁雪丰 审校

亚马逊
五星图书

- Spring 专家力作
- 理论与实践完美结合
- 问题描述 → 解决方案 → 实现方法



人民邮电出版社
POSTS & TELECOM PRESS

TURING

图灵程序设计丛书 Java 系列

Spring Recipes A Problem-Solution Approach

Spring 攻略

Gary Mak 著

丁雪丰 荣浩 杨春花 俞黎敏 译

丁雪丰 审校

人民邮电出版社

样书

专用章

人民邮电出版社

北京

图书在版编目 (CIP) 数据

Spring 攻略 / 麦 (Mak,G.) 著; 丁雪丰等译. —北京: 人民邮电出版社, 2009.8

(图灵程序设计丛书)

书名原文: Spring Recipes: A Problem-Solution Approach

ISBN 978-7-115-20890-3

I. S… II. ①麦…②丁… III. JAVA 语言-程序设计
IV. TP312

中国版本图书馆CIP数据核字 (2009) 第080225号

内 容 提 要

Spring 着眼于 Java/Java EE 应用开发的诸多方面, 并针对它们提供了简单的解决方案。本书涵盖了 Spring 2.5 从基础概念到高级应用的所有主题, 介绍了几种常见的 Spring 项目, 重点关注 Spring 2.5 为构建企业级 Java 应用程序所增加的最新特性。本书的所有主题都是围绕完整真实的项目代码示例展开的, 读者可以按照这些示例一步一步地实践, 从而迅速掌握使用 Spring Framework 进行 Java/Java EE 开发。

本书适合具有丰富开发经验以及从零起步的 Java 开发人员阅读。

图灵程序设计丛书

Spring 攻略

-
- ◆ 著 Gary Mak
 - 译 丁雪丰 荣浩 杨春花 俞黎敏
 - 审 校 丁雪丰
 - 责任编辑 傅志红
 - 执行编辑 陈兴璐

 - ◆ 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: 39
 - 字数: 1044千字 2009年8月第1版
 - 印数: 1-3 000册 2009年8月北京第1次印刷
 - 著作权合同登记号 图字: 01-2009-2901号

ISBN 978-7-115-20890-3/TP

定价: 89.00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 *Spring 2 Recipes: A Problem-Solution Approach* by Gary Mak, published by Apress L.P., 2855 Telegraph Avenue, Suite 600, Berkeley, CA 94705 USA.

Copyright © 2008 by Gary Mak. 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。

译者简介



丁雪丰，网名DigitalSonic，满江红开放技术研究组织成员，Spring Framework 2.0 & 2.5文档翻译项目负责人，参与了Hibernate、Seam、OSGi R4等多份文档的翻译及校对工作。平时积极关注开源项目，是著名SFTP/SCP软件WinSCP的简体中文汉化者。参与翻译及编著的图书有《Spring专业开发指南》、《JRuby实战》、《开源技术选型手册》等。博客地址：<http://digitalsonic.javaeye.com>或 <http://blog.csdn.net/dingx>。



荣浩，现为ThoughtWorks咨询师，开发流程用户组社区（www.opug.org.cn）和jBPM-Side开源工作流项目（www.jbpm-side.org.cn）的发起者之一，关注敏捷和企业流程改进过程。参与翻译的图书有《AJAX企业级开发》。博客地址：<http://ronghao.javaeye.com>。



杨春花，北京聚众开源科技有限公司营销助理，负责社区活动的组织与沟通。满江红开放技术研究组织核心翻译与审校人员。参与翻译的图书有《Hibernate实战（第2版）》、《Spring 2企业应用开发》、《Effective Java中文版（第2版）》等。博客地址：<http://blog.csdn.net/Java2Class/>。



俞黎敏，网名阿敏总司令，目前在国际商业机器全球服务（中国）有限公司广州分公司（IBM广州），担任软件部高级信息工程师、技术顾问，主要负责IBM WebSphere系列产品的技术支持工作，专注于产品新特性、系统性能调优、疑难问题诊断与解决。参与翻译与审校的图书有《Ajax设计模式》、《CSS实战手册》、《Hibernate实战（第2版）》、《Java脚本编程》、《Effective Java中文版（第2版）》等。博客地址：<http://blog.csdn.net/YuLimin/> 或者 <http://YuLimin.javaeye.com>。

译者序

几年前，机缘巧合之下，我被一个朋友拉去参加技术活动，在那次活动上我第一次接触到了Spring。说实话，当时怎么都没有想到几年间Spring会有如此迅速的发展，更想不到自己会与Spring如此有缘。

从*Expert One-on-One J2EE Design and Development*到*Expert One-on-One J2EE Development without EJB*，Spring之父Rod Johnson一次又一次让大家眼前一亮。而Spring更是被广大Java开发者所接受，可以毫不夸张地说，提到Java企业级应用的开发，第一时间蹦出的词汇中肯定有Spring。

正是因为Spring如此大红大紫，越来越多的人希望了解它、学习它。每当有人让我推荐学习资料时，我都会让他去网上下载官方文档，这足以满足大多数人的需要。但如果你比较喜欢飘着墨香的纸质书，而又恰好捧起本书，相信它一样也不会让你失望的。书中以“问题描述/解决方案/实现方法”的形式介绍了Spring中的大多数内容和Spring Portfolio中的一些项目，读者既可以跟着作者的思路慢慢学习，也可以“对症下药”，立刻解决眼前的问题。要是你对Spring已经有了一定的了解，那么本书也能作为一本不错的参考手册，偶尔翻翻也会有意想不到的收获。

正如Spring背后有着一个优秀的团队一样，本书同样也有一个优秀的翻译团队，大家相互帮助、相互鼓励，唯一的目标就是为读者带来最好的作品。在本书翻译期间，荣浩负责第1~5章，俞黎敏负责第6、7、16、17、18章，杨春花负责第8~12章，而我则负责第13、14、15、19章以及全书的审稿、统稿。

最后，请允许我在此感谢那些在本书的翻译过程中给予我们帮助的朋友和图灵的编辑们，正是有了你们，本书才能最终呈现在读者面前。亲爱的读者朋友，如果你发现书中存在错误或者有什么意见，请毫不犹豫地告诉我们，我们不胜感激。

丁雪丰

2009年2月

前 言

我第一次使用Spring是在2004年，从那时候开始，我就成了Spring的忠实粉丝，在几乎所有的项目里都使用了它。它的简洁性和合理性深深吸引了我。Spring是我曾经使用过的最简单最强大的Java/Java EE应用程序框架，它使用简单方法解决棘手问题的能力给了我深刻的印象。Spring给出的解决方案也许不是最好的，但却是我能够想到的最合理的。

Spring着眼于Java/Java EE应用开发的绝大多数方面，并针对它们提供了简单的解决方案。通过Spring，许多工业级的最佳实践将会被自然而然地应用到应用程序的设计和实现中。相对于1.x版本，Spring 2.x的发布增加了许多改进和新特性。本书重点关注Spring 2.5为构建企业级Java应用程序所增加的最新特性。

作为一名教授编程技术的培训老师，我发现学生经常会面临的最大挑战是如何将实验性质的项目运行起来。许多编程图书都带有代码示例，但是绝大多数都只是代码片段，不是完整的项目。这类图书多数建议从它们的网站下载完整的项目代码，但是却没有提供让你自己一步一步构建项目的机会。我相信，你会从构建项目的过程中学到很多东西，而且项目一旦运行起来，你将获得极大的自信，这就是我撰写本书的初衷。

作为与时俱进的Java开发者，我们往往需要掌握新技术或者新框架。因为我们只是使用技术的开发者，不是需要参加考试的学生，所以没必要记住所有东西，而只需要能在必要时迅速找到参考资料就可以了。为了使经验丰富的读者和从头读到尾通读本书的初学者都能从本书中受益，本书每章都由多个“问题-方案”式的内容组成。采用这样的方式，你就可以轻松地查找特定问题的解决方案。

本书的主题是围绕完整的真实代码示例展开的，你可以按照这些示例一步一步地实践。你在本书中很少会看到对复杂概念的抽象叙述，你会发现大量生动的示例。在开始新项目时，可以考虑直接从复制本书提供的代码和配置文件，然后根据需要修改它们。比起从零开始创建项目，这能为你节省大量工作。

读者对象

本书的读者是那些想快速获得使用Spring Framework进行Java/Java EE开发经验的Java开发人员。如果你已经在项目里使用过Spring，那么也可以将本书作为参考，同时，你还会发现本书的代码示例非常有用。

阅读本书不需要很多的Java EE经验。不过，你需要了解使用Java进行面向对象编程的基本知识（例如，创建类/接口、实现接口、扩展基类、运行Main类、设置classpath等）。此外，你还需要知道关于Web的基础知识和数据库的相关概念，知道如何创建动态网页和使用SQL语句查询数据库。

本书结构

本书涵盖了Spring 2.5从基础概念到高级应用的所有主题。此外，也介绍了几种常见的Spring项目，这些项目会给你的应用程序开发带来巨大的参考价值。本书分为三个部分19个章节。

- 第一部分：核心概念。这部分重点讲述了Spring Framework的核心概念和实现机制。这部分内容旨在让你熟悉Spring的核心概念，如此一来你就能很快地掌握Spring的其他主题和用法了。
- 第1章：控制反转和容器。本章介绍了Spring的核心概念——IoC设计原理，以及容器的重要性。如果你对IoC已经非常熟悉，那么可以跳过这一章。
- 第2章：Spring简介。本章概述Spring的架构及其相关项目。此外，还介绍了如何在你的开发环境里建立Spring项目。
- 第3章：Spring中的Bean配置。本章介绍了Spring IoC容器里基本的Bean配置。理解本章所介绍的特性是阅读后续章节的基础。
- 第4章：高级Spring IoC容器。本章涵盖了Spring IoC容器的高级特性及其内部实现机制。尽管这些特性并不会像在第3章里介绍的特性那样被频繁地使用，但是，对于功能强大的容器来说，它们都是必需的。
- 第5章：动态代理和经典的Spring AOP。本章解释了需要AOP的原因，以及如何使用动态代理和经典的Spring AOP模块化横切关注点。如果你已经理解了AOP，并想直接使用Spring 2.x里的AOP，那么可以跳过本章，直接阅读第6章。
- 第6章：Spring 2.x AOP和AspectJ支持。本章涵盖了Spring 2.x AOP的使用方法以及一些高级的AOP主题，其中包括了如何在Spring应用程序里集成AspectJ框架。
- 第二部分：基础主题。这部分内容涉及Spring Framework的基础主题，在开发企业级应用程序时经常会用到它们。
- 第7章：Spring对JDBC的支持。本章说明了如何通过Spring的JDBC访问框架来简化JDBC的使用。它也可以作为对Spring数据访问模块的介绍。
- 第8章：Spring中的事务管理。本章讨论了Spring的不同事务管理方式并详细解释了事务属性。
- 第9章：Spring对ORM的支持。本章的重点是如何将流行的ORM框架（包括Hibernate和JPA）集成到Spring应用程序中。
- 第10章：Spring MVC框架。本章讨论了如何使用Spring Web MVC框架进行基于Web的应用程序开发，其中包括了传统的和新的基于注解的两种方式。
- 第11章：整合Spring与其他Web框架。本章介绍了如何将Spring Framework与Struts、JSF和DWR等流行的Web应用程序框架进行整合。
- 第12章：Spring对测试的支持。本章涉及Java应用程序里基本的测试技术以及Spring Framework所提供的测试支持特性。
- 第三部分：高级主题。这部分内容涵盖了Spring Framework的高级主题以及与之相关的项目。不过，想彻底讲清楚其中任何一个主题都需要一整本书的篇幅。所以，这些章节的目的在于提供一些特定于Spring的有用的基本知识和使用指南。
- 第13章：Spring Security框架。本章介绍了如何使用Spring Security框架来保护应用程序，这个框架的前身是著名的Acegi Security。本章的重点是使用Spring Security 2.0来保护Web应用程序。

- 第14章：Spring Portlet MVC框架。本章介绍了如何使用Spring Portlet MVC框架开发门户应用程序，重点是介绍该框架不同于Web MVC的特有特性。
- 第15章：Spring Web Flow。本章介绍了如何使用Spring Web Flow对Web应用程序的UI流程进行建模和管理，重点是在Spring MVC和JSF里使用Spring Web Flow 2.0。
- 第16章：Spring远程服务和Web服务。本章讨论Spring对各种远程服务技术的支持，这些技术包括RMI、Hessian、Burlap、HTTP Invoker和Web服务。此外，也介绍了如何使用Spring Web服务开发契约优先（Contract-First）的Web服务。
- 第17章：Spring对EJB和JMS的支持。本章讨论了如何利用Spring对EJB的支持开发EJB 2.x和3.0组件，以及如何利用Spring对JMS的支持简化JMS消息的发送、接收和监听。
- 第18章：Spring对JMX、电子邮件和调度的支持。本章讨论了如何将Spring Bean导出为JMX MBean以及访问远程MBean。此外，也介绍了如何利用Spring对电子邮件和调度的支持来发送邮件和调度任务。
- 第19章：Spring中的脚本编程。本章讨论了如何在Spring应用程序里使用流行的脚本语言，如JRuby、Groovy和BeanShell。

本书每章讨论一个Spring主题，每个主题又包括多个问题及这些问题的解决方案。你可以查找某个特定问题的解决方案，并在“实现方法”里查看解决方案是如何实现的。每章以一个完整真实的例子来说明主题。每一章的例子是连贯的，但各章之间的例子又保持独立。

代码约定

在需要提醒你注意代码示例里的某一特定部分时，我会加粗这部分代码。请注意，粗体部分并不是反映代码与上一个版本相比所发生的变化。当代码行过长超过页面宽度的限制时，我会使用代码续行符（↵）来换行。在运行这些代码时，你需要自己将这些折行的代码连接起来，中间不能留有空格。

配置环境

因为Java编程语言是平台无关的，所以你可以自由选择支持的操作系统。不过，因为本书里出现的文件系统路径是基于Windows的，所以在运行这些代码前，你可以简单地将这些路径转换为你的操作系统的格式。

要充分掌握本书的内容，你需要安装JDK1.5或更高版本。同时，为了方便开发，还应该安装Java IDE。本书的项目是使用Eclipse WTP（Web Tools Platform）开发的，我也建议你安装它。

下载代码

可以从Apress的网站下载本书的源代码^①，地址为<http://www.apress.com/book/downloadfile/4004>。源代码根据章节组织在一起，每章包括一到多个独立的Eclipse项目。要设置和运行源代码，请参阅位于根目录下的readme.txt文件。

^① 也可以在图灵教育网站（www.turingbook.com）上下载本书的源代码。——编者注

联系作者

非常欢迎你对本书内容提出问题或给出反馈。你可以把你的意见发送到springrecipes@metaarchit.com，同时可以在<http://www.metaarchit.com>上查看有关本书的讨论和更新内容。

致谢

在这里，我首先要感谢我的家人，特别是我的妻子Ivy。在本书的写作期间，是她给予了我精神上的支持，把家料理得井井有条。如果没有她的爱和支持，我不可能完成本书。

此外，还要感谢为本书付出辛勤劳动的Apress团队，特别是下面的几位。

Steve Anglin，我的策划编辑，是他找到我并鼓励我撰写本书。他为本书的提纲提供了很棒的想法，并为本书组织了一支伟大的团队。

Tom Welsh，我的项目编辑，他对本书的文字和技术方面都提供了极好的支持。在我写作遇到困难的时候，是他经常鼓励我并提供建议。

Kylie Johnston，我的项目负责人，她非常好地安排了本书的时间进度，对我的要求反馈也非常迅速。她的经验使得本书的时间安排非常灵活，让我能以一种非常舒适的方式完成我的工作。

Kris Lander，我的技术审稿人，他从技术角度审阅了本书，指出了其中存在的错误，并根据他丰富的Spring使用经验给出了非常有价值的建议。

Sam Brannen，我的另一位技术审稿人，他给本书提供了大量的专业意见，这些意见都直接提高了本书的质量，而且立竿见影。他是Spring的核心开发人员，我对他的意见深信不疑。

Damon Larson，我的文字编辑，他负责纠正我的语法错误，对文字表达进行润色，并保持术语的前后一致，他做得非常出色。

Laura Esterman，我的生产编辑，她的工作是本书看起来如此好的原因，她处理了我后期的修改要求。

Dina Quan、Ron Strauss和April Eddy，他们也为本书的成功出版提供了很多帮助。

目 录

第一部分 核心概念

第 1 章 控制反转和容器	2
1.1 使用容器管理组件	2
1.1.1 问题描述	2
1.1.2 解决方案	2
1.1.3 实现方法	3
1.2 使用服务定位器降低查找组件的复杂性	7
1.2.1 问题描述	7
1.2.2 解决方案	7
1.2.3 实现方法	7
1.3 应用控制反转和依赖注入	8
1.3.1 问题描述	8
1.3.2 解决方案	8
1.3.3 实现方法	9
1.4 理解不同类型的依赖注入	10
1.4.1 问题描述	10
1.4.2 解决方案	10
1.4.3 实现方法	11
1.5 使用配置文件配置容器	13
1.5.1 问题描述	13
1.5.2 解决方案	14
1.5.3 实现方法	14
1.6 小结	16
第 2 章 Spring 简介	17
2.1 Spring Framework	17
2.1.1 Spring 的模块介绍	17
2.1.2 Spring 的发布版本	19
2.1.3 Spring 的项目	20

2.2 安装Spring Framework	21
2.2.1 问题描述	21
2.2.2 解决方案	21
2.2.3 实现方法	21
2.3 建立Spring项目	22
2.3.1 问题描述	22
2.3.2 解决方案	22
2.3.3 实现方法	23
2.4 安装Spring IDE	24
2.4.1 问题描述	24
2.4.2 解决方案	24
2.4.3 实现方法	24
2.5 使用Spring IDE的Bean-supporting特性	26
2.5.1 问题描述	26
2.5.2 解决方案	26
2.5.3 实现方法	26
2.6 小结	31
第 3 章 Spring 中的 Bean 配置	32
3.1 在Spring IoC容器里配置Bean	32
3.1.1 问题描述	32
3.1.2 解决方案	32
3.1.3 实现方法	32
3.2 实例化Spring IoC容器	36
3.2.1 问题描述	36
3.2.2 解决方案	36
3.2.3 实现方法	36
3.3 解决构造器歧义	38
3.3.1 问题描述	38
3.3.2 解决方案	38
3.3.3 实现方法	38

3.4 指定Bean引用	41	3.13.2 解决方案	71
3.4.1 问题描述	41	3.13.3 实现方法	72
3.4.2 解决方案	41	3.14 小结	76
3.4.3 实现方法	41	第4章 高级 Spring IoC 容器	77
3.5 通过依赖检查来检查属性	44	4.1 通过调用构造器创建Bean	77
3.5.1 问题描述	44	4.1.1 问题描述	77
3.5.2 解决方案	45	4.1.2 解决方案	77
3.5.3 实现方法	45	4.1.3 实现方法	77
3.6 通过@Required注解检查属性	46	4.2 通过调用静态工厂方法创建Bean	80
3.6.1 问题描述	46	4.2.1 问题描述	80
3.6.2 解决方案	46	4.2.2 解决方案	80
3.6.3 实现方法	47	4.2.3 实现方法	80
3.7 XML配置里的Bean自动装配	48	4.3 通过调用实例工厂方法创建Bean	81
3.7.1 问题描述	48	4.3.1 问题描述	81
3.7.2 解决方案	49	4.3.2 解决方案	81
3.7.3 实现方法	49	4.3.3 实现方法	81
3.8 使用@Autowired和@Resource		4.4 使用Spring的工厂Bean创建Bean	83
自动装配Bean	52	4.4.1 问题描述	83
3.8.1 问题描述	52	4.4.2 解决方案	83
3.8.2 解决方案	52	4.4.3 实现方法	83
3.8.3 实现方法	52	4.5 根据静态字段声明Bean	85
3.9 继承Bean配置	58	4.5.1 问题描述	85
3.9.1 问题描述	58	4.5.2 解决方案	85
3.9.2 解决方案	58	4.5.3 实现方法	85
3.9.3 实现方法	58	4.6 根据对象属性声明Bean	86
3.10 为Bean属性定义集合	61	4.6.1 问题描述	86
3.10.1 问题描述	61	4.6.2 解决方案	86
3.10.2 解决方案	61	4.6.3 实现方法	86
3.10.3 实现方法	61	4.7 设置Bean的作用域	88
3.11 为集合元素指定数据类型	67	4.7.1 问题描述	88
3.11.1 问题描述	67	4.7.2 解决方案	88
3.11.2 解决方案	67	4.7.3 实现方法	88
3.11.3 实现方法	67	4.8 定制Bean的初始化和销毁过程	90
3.12 使用工厂Bean和Utility Schema定义		4.8.1 问题描述	90
集合	69	4.8.2 解决方案	90
3.12.1 问题描述	69	4.8.3 实现方法	91
3.12.2 解决方案	69	4.9 使Bean感知容器	95
3.12.3 实现方法	69	4.9.1 问题描述	95
3.13 在classpath中扫描组件	71	4.9.2 解决方案	95
3.13.1 问题描述	71		

4.9.3 实现方法	96	5.3 使用经典的Spring通知来模块化	
4.10 创建Bean后置处理器	96	横切关注点	127
4.10.1 问题描述	96	5.3.1 问题描述	127
4.10.2 解决方案	96	5.3.2 解决方案	127
4.10.3 实现方法	97	5.3.3 实现方法	128
4.11 外部化Bean配置	100	5.4 使用经典的Spring切入点匹配方法	135
4.11.1 问题描述	100	5.4.1 问题描述	135
4.11.2 解决方案	100	5.4.2 解决方案	135
4.11.3 实现方法	100	5.4.3 实现方法	136
4.12 解析文本信息	101	5.5 为Bean自动创建代理	138
4.12.1 问题描述	101	5.5.1 问题描述	138
4.12.2 解决方案	101	5.5.2 解决方案	138
4.12.3 实现方法	101	5.5.3 实现方法	138
4.13 使用应用程序事件进行通信	103	5.6 小结	140
4.13.1 问题描述	103		
4.13.2 解决方案	103	第6章 Spring 2.x AOP 和 AspectJ 支持	141
4.13.3 实现方法	104	6.1 在Spring中启用AspectJ注解支持	141
4.14 在Spring里注册属性编辑器	106	6.1.1 问题描述	141
4.14.1 问题描述	106	6.1.2 解决方案	142
4.14.2 解决方案	106	6.1.3 实现方法	142
4.14.3 实现方法	106	6.2 用AspectJ注解声明切面	144
4.15 创建定制的属性编辑器	108	6.2.1 问题描述	144
4.15.1 问题描述	108	6.2.2 解决方案	144
4.15.2 解决方案	109	6.2.3 实现方法	144
4.15.3 实现方法	109	6.3 访问连接点信息	149
4.16 加载外部资源	110	6.3.1 问题描述	149
4.16.1 问题描述	110	6.3.2 解决方案	149
4.16.2 解决方案	110	6.3.3 实现方法	149
4.16.3 实现方法	111	6.4 指定切面优先级	150
4.17 小结	113	6.4.1 问题描述	150
		6.4.2 解决方案	150
第5章 动态代理和经典的 Spring AOP	114	6.4.3 实现方法	150
5.1 非模块化的横切关注点所带来的问题	115	6.5 重用切入点定义	152
5.1.1 追踪方法	116	6.5.1 问题描述	152
5.1.2 验证参数	119	6.5.2 解决方案	152
5.1.3 找出问题	120	6.5.3 实现方法	152
5.2 使用动态代理模块化横切关注点	122	6.6 编写AspectJ切入点表达式	154
5.2.1 问题描述	122	6.6.1 问题描述	154
5.2.2 解决方案	122	6.6.2 解决方案	154
5.2.3 实现方法	122	6.6.3 实现方法	154

8.5.1	问题描述	222	9.1.3	用Hibernate作为JPA引擎持久化对象	252
8.5.2	解决方案	222	9.2	在Spring中配置ORM资源工厂	255
8.5.3	实现方法	222	9.2.1	问题描述	255
8.6	用事务通知声明式地管理事务	224	9.2.2	解决方案	256
8.6.1	问题描述	224	9.2.3	实现方法	256
8.6.2	解决方案	224	9.3	用Spring的ORM模板持久化对象	261
8.6.3	实现方法	225	9.3.1	问题描述	261
8.7	用@Transactional注解声明式地管理事务	226	9.3.2	解决方案	261
8.7.1	问题描述	226	9.3.3	实现方法	262
8.7.2	解决方案	226	9.4	用Hibernate的上下文会话持久化对象	267
8.7.3	实现方法	226	9.4.1	问题描述	267
8.8	设置传播事务属性	227	9.4.2	解决方案	267
8.8.1	问题描述	227	9.4.3	实现方法	267
8.8.2	解决方案	227	9.5	用JPA的上下文注入持久化对象	270
8.8.3	实现方法	228	9.5.1	问题描述	270
8.9	设置隔离事务属性	231	9.5.2	解决方案	270
8.9.1	问题描述	231	9.5.3	实现方法	270
8.9.2	解决方案	231	9.6	小结	273
8.9.3	实现方法	232	第 10 章	Spring MVC 框架	274
8.10	设置回滚事务属性	238	10.1	用Spring MVC开发简单的Web应用程序	274
8.10.1	问题描述	238	10.1.1	问题描述	274
8.10.2	解决方案	238	10.1.2	解决方案	274
8.10.3	实现方法	238	10.1.3	实现方法	275
8.11	设置超时和只读事务属性	239	10.2	将请求映射到处理程序	284
8.11.1	问题描述	239	10.2.1	问题描述	284
8.11.2	解决方案	239	10.2.2	解决方案	284
8.11.3	实现方法	239	10.2.3	实现方法	285
8.12	用加载时织入来管理事务	240	10.3	用Handler Interceptor拦截请求	287
8.12.1	问题描述	240	10.3.1	问题描述	287
8.12.2	解决方案	241	10.3.2	解决方案	287
8.12.3	实现方法	241	10.3.3	实现方法	288
8.13	小结	244	10.4	解析用户区域	290
第 9 章	Spring 对 ORM 的支持	245	10.4.1	问题描述	290
9.1	直接使用ORM框架的问题	245	10.4.2	解决方案	290
9.1.1	利用Hibernate API和Hibernate XML映射持久化对象	247	10.4.3	实现方法	290
9.1.2	利用Hibernate API和JPA注解持久化对象	250	10.5	外部化区分区域的文本信息	292
			10.5.1	问题描述	292

10.5.2 解决方案	292	第 11 章 整合 Spring 与其他 Web 框架	337
10.5.3 实现方法	292	11.1 在通用的 Web 应用程序中访问 Spring	337
10.6 按名称解析视图	293	11.1.1 问题描述	337
10.6.1 问题描述	293	11.1.2 解决方案	337
10.6.2 解决方案	293	11.1.3 实现方法	338
10.6.3 实现方法	293	11.2 整合 Spring 与 Struts 1.x	342
10.7 将异常映射到视图	296	11.2.1 问题描述	342
10.7.1 问题描述	296	11.2.2 解决方案	342
10.7.2 解决方案	296	11.2.3 实现方法	342
10.7.3 实现方法	296	11.3 整合 Spring 与 JSF	347
10.8 构造 ModelAndView 对象	297	11.3.1 问题描述	347
10.8.1 问题描述	297	11.3.2 解决方案	348
10.8.2 解决方案	297	11.3.3 实现方法	348
10.8.3 实现方法	298	11.4 整合 Spring 与 DWR	352
10.9 创建带有参数化视图的控制器	299	11.4.1 问题描述	352
10.9.1 问题描述	299	11.4.2 解决方案	352
10.9.2 解决方案	299	11.4.3 实现方法	352
10.9.3 实现方法	300	11.5 小结	355
10.10 用表单控制器处理表单	301	第 12 章 Spring 对测试的支持	356
10.10.1 问题描述	301	12.1 用 JUnit 和 TestNG 创建测试	357
10.10.2 解决方案	301	12.1.1 问题描述	357
10.10.3 实现方法	302	12.1.2 解决方案	357
10.11 用向导表单控制器处理多页表单	313	12.1.3 实现方法	357
10.11.1 问题描述	313	12.2 创建单元测试和集成测试	361
10.11.2 解决方案	313	12.2.1 问题描述	361
10.11.3 实现方法	313	12.2.2 解决方案	362
10.12 将多个动作分组到一个控制器	321	12.2.3 实现方法	362
10.12.1 问题描述	321	12.3 单元测试 Spring MVC 控制器	370
10.12.2 解决方案	321	12.3.1 问题描述	370
10.12.3 实现方法	321	12.3.2 解决方案	370
10.13 创建 Excel 和 PDF 视图	326	12.3.3 实现方法	370
10.13.1 问题描述	326	12.4 管理集成测试中的应用程序上下文	374
10.13.2 解决方案	326	12.4.1 问题描述	374
10.13.3 实现方法	326	12.4.2 解决方案	374
10.14 用注解开发控制器	330	12.4.3 实现方法	375
10.14.1 问题描述	330	12.5 将测试夹具注入到集成测试中	380
10.14.2 解决方案	330	12.5.1 问题描述	380
10.14.3 实现方法	330	12.5.2 解决方案	381
10.15 小结	336	12.5.3 实现方法	381

12.6	管理集成测试中的事务	384	13.7	处理领域对象的安全	427
12.6.1	问题描述	384	13.7.1	问题描述	427
12.6.2	解决方案	384	13.7.2	解决方案	427
12.6.3	实现方法	385	13.7.3	实现方法	427
12.7	在集成测试中访问数据库	389	13.8	小结	436
12.7.1	问题描述	389	第 14 章 Spring Portlet MVC 框架 437		
12.7.2	解决方案	390	14.1	用Spring Portlet MVC开发简单的Portlet	437
12.7.3	实现方法	390	14.1.1	问题描述	437
12.8	利用Spring的通用测试注解	393	14.1.2	解决方案	437
12.8.1	问题描述	393	14.1.3	实现方法	438
12.8.2	解决方案	393	14.2	将Portlet请求映射到处理程序	444
12.8.3	实现方法	394	14.2.1	问题描述	444
12.9	小结	395	14.2.2	解决方案	444
第三部分 高级主题			14.2.3	实现方法	444
第 13 章 Spring Security 框架 398			14.3	用简单表单控制器处理Portlet表单	453
13.1	保护URL访问	398	14.3.1	问题描述	453
13.1.1	问题描述	398	14.3.2	解决方案	453
13.1.2	解决方案	398	14.3.3	实现方法	453
13.1.3	实现方法	399	14.4	使用注解来开发Portlet控制器	460
13.2	登录Web应用程序	407	14.4.1	问题描述	460
13.2.1	问题描述	407	14.4.2	解决方案	460
13.2.2	解决方案	407	14.4.3	实现方法	460
13.2.3	实现方法	408	14.5	小结	465
13.3	验证用户	411	第 15 章 Spring Web Flow 466		
13.3.1	问题描述	411	15.1	用Spring Web Flow管理简单的UI流程	466
13.3.2	解决方案	411	15.1.1	问题描述	466
13.3.3	实现方法	412	15.1.2	解决方案	466
13.4	指定访问控制决策	419	15.1.3	实现方法	467
13.4.1	问题描述	419	15.2	用不同的流程类型进行Web流程建模	472
13.4.2	解决方案	419	15.2.1	问题描述	472
13.4.3	实现方法	420	15.2.2	解决方案	472
13.5	保护方法调用	422	15.2.3	实现方法	473
13.5.1	问题描述	422	15.3	保护Web流程	482
13.5.2	解决方案	422	15.3.1	问题描述	482
13.5.3	实现方法	422	15.3.2	解决方案	483
13.6	处理视图中的安全	425			
13.6.1	问题描述	425			
13.6.2	解决方案	425			
13.6.3	实现方法	425			