

Beginning Spring

Spring入门经典

Rod Johnson和Jürgen Höller作序推荐!



[美] Mert Caliskan
Kenan Sevindik

著

王净 范园芳 田洪

译

清华大学出版社

Spring 入门经典

[美] Mert Caliskan 著
Kenan Sevindik

王 净 范园芳 田 洪 译

清华大学出版社

北 京

Mert Caliskan, Kenan Sevindik

Beginning Spring

EISBN: 978-1-118-89292-3

Copyright © 2015 by John Wiley & Sons, Inc., Indianapolis, Indiana

All Rights Reserved. This translation published under license.

Trademarks: Wiley, the Wiley logo, Wrox, the Wrox logo, Programmer to Programmer, and related trade dress are trademarks or registered trademarks of John Wiley & Sons, Inc. and/or its affiliates, in the United States and other countries, and may not be used without written permission. All other trademarks are the property of their respective owners. John Wiley & Sons, Inc., is not associated with any product or vendor mentioned in this book.

本书中文简体字版由 Wiley Publishing, Inc. 授权清华大学出版社出版。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。

北京市版权局著作权合同登记号 图字: 01-2015-2179

Copies of this book sold without a Wiley sticker on the cover are unauthorized and illegal.

本书封面贴有 Wiley 公司防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话: 010-62782989 13701121933

图书在版编目(CIP)数据

Spring 入门经典/(美)卡利斯科(Caliskan, M.), (美)塞温迪克(Sevindik, K.)著;王净,范园芳,田洪译.

—北京:清华大学出版社,2015

书名原文: Beginning Spring

ISBN 978-7-302-41248-9

I. ①S… II. ①卡… ②塞… ③王… ④范… ⑤田… III. ①JAVA 语言—程序设计 IV. ①TP312

中国版本图书馆 CIP 数据核字(2015)第 186483 号

责任编辑:王 军 韩宏志

装帧设计:孔祥峰

责任校对:成凤进

责任印制:何 芊

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 邮 编:100084

社总机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

印 刷 者:北京密云胶印厂

装 订 者:三河市漂源装订厂

经 销:全国新华书店

开 本:185mm×260mm

印 张:27.25

字 数:646 千字

版 次:2015 年 9 月第 1 版

印 次:2015 年 9 月第 1 次印刷

印 数:1~4000

定 价:59.80 元

产品编号:064481-01

译者序

Spring是一种多层的J2EE应用程序框架，是以Rod Johnson撰写的*Expert One-on-One J2EE Design and Development*一书的代码为基础发展而来的。Spring的核心是提供一种新的机制来管理业务对象及其依赖关系。例如，可以指定一个DAO(数据访问对象)类依赖一个DataSource类。此外，还允许开发人员通过接口编程，使用XML文件来简单地定义其实现。Spring拥有许多类来支持其他框架(如Hibernate和Struts)，这使得集成变得易如反掌。

Spring不仅解决了开发人员的实际问题，也促使他们养成良好的编程习惯，例如接口编程可减少耦合以及提供简易的测试。在当今的编程领域特别是在Java中，一些优秀的开发人员正在实践TDD(Test Driven Development，测试驱动开发)。TDD可以让开发人员用测试类或是client类驱动其他类的设计。而Spring本身提供了丰富的测试套件，这使得测试类变得轻松自如。

《Spring入门经典》深入浅出地介绍了Spring的相关知识。首先重点介绍了Spring框架的核心概念和功能。读者将首先了解基本概念，比如非入侵性的POJO(Plain Old Java Object)编程模型方法，然后给出了依赖注入模式(即控制反转)的定义。此外，Spring还为构建Web应用程序提供了MVC实现。除了讲解Spring Framework所提供的事务机制之外，还介绍了Spring框架的Java Database Connectivity和Java Persistence API功能。随后讲述Spring为进行测试驱动开发所提供的功能，介绍面向方面编程，从而使读者了解Spring如何处理系统横切关注点的实现。Spring表达式语言、缓存和REST风格的Web服务提供了Spring Framework的大量功能。最后学习了Spring Security以及框架4.0版本所提供的最新功能。此外，本书通过提供丰富实用的代码示例，来真正展示Spring框架的强大功能——能够使企业级应用程序的开发变得更简单。

本书适用于已具有一定Java编程基础的读者，以及在Java平台下进行各类软件开发的开发人员、测试人员，尤其适用于企业级Java开发人员。本书既可以被刚开始学习Spring的读者当作学习指南，也可以被那些想深入了解Spring某方面功能的资深用户作为参考用书。

参与本书翻译活动的有王净、田洪、范园芳、范桢、胡训强、纪红、彭洁、晏峰、余佳隼、张洁以及赵翊含，最终由王净负责统稿。此外，还要感谢我的家人，她们总是无怨无悔地支持我的一切工作，我为有这样的家庭而感到幸福。

译者在翻译过程中，尽量保持原书的特色，并对书中出现的术语和难词难句进行了仔细推敲和研究。但毕竟有少量技术是译者在自己的研究领域中不曾遇到过的，所以疏漏和

争议之处在所难免，望广大读者提出宝贵意见。

最后，希望广大读者能多花些时间细细品味这本凝聚作者和译者大量心血的书籍，为将来的职业生涯奠定良好的基础。

王净
于广州

作者简介

Mert Caliskan 是一位首席软件架构师，居住在土耳其安卡拉，从事企业级 Java Web 应用程序架构设计的软件开发已有十多年。他是软件项目(比如 PrimeFaces)的源代码开放倡导者，此外还是 *PrimeFaces Cookbook* 第 1 版和第 2 版(Packt Publishing, 2013)的合著者。他是 AnkaraJUG 的创始人，而 AnkaraJUG 是土耳其最活跃的 JUG。Caliskan 是 Hacettepe 大学关于企业级 Web 应用程序体系和 Web 服务方面的兼职讲师。在 2014 年，他获得了 Java Champion 头衔，并且在国内和国际会议上分享自己的知识，比如 JDays 2015、JavaOne 2013、JDC2010 和 JSFDays'08。可以在 Twitter @mertcal 上了解他的相关信息。

Kenan Sevindik 擅长使用不同的 Java 技术设计和开发企业级应用程序。他使用 Java 的经历可以追溯到 1998 年，当时他在大学为在线教育项目开发 Java 小程序。在 Spring Application Framework 和 Spring Security Framework 的起始阶段，Sevindik 就开始使用它们了。他拥有计算机工程学士学位。目前，主要从事 Hareami IT Solution，开发企业软件以及在世界各地提供关于 Java、OOP、AOP、Spring、Spring Security 和 Hibernate 的培训、指导和咨询服务。可以从 <http://blog.harezmi.com.tr> 阅读他的技术著作，并通过 ksevindik@harezmi.com.tr 与他取得联系。

技术编辑简介

Chád (Shod) Darby 是 Java 开发方面的作者、讲师和演讲者。作为一名 Java 应用程序和体系结构方面公认的权威，他在全球软件开发会议上发表技术演讲。在他作为一名专业软件架构师 15 年期间，曾经为 Blue Cross/Blue Shield、Merck、Boeing、Red Hat 以及少量的新公司工作。Chád 是多部 Java 图书的特约作者，包括 *Professional Java E-Commerce*(Wiley, 2001)以及 *Beginning Java Networking*(Wiley, 2010)。此外，他还获得了来自 Sun Microsystems 和 IBM 的 Java 认证，并从 Carnegie Mellon 大学获得了计算机理学学士学位。你可以访问他的微博 www.luv2code.com，观看关于 Java 的免费视频教程。可以在 Twitter @darbyluv2code 上了解他的相关信息。

致 谢

首先，我想要感谢我的朋友 Kenan Sevindik，感谢他参与本书的编写。此外，还要特别感谢 Spring Framework 的创建者 Rod Johnson 和 Jurgen Holler，感谢他们为本书作序。如果没有他们的想法和灵感点燃了我们，这本书将不会存在。

其次，还要感谢副社长 Jim Minatel、项目编辑 Charlotte Kughen、技术编辑 Chád Darby 和 Krishna Srinivasan、编审 Kim Cofer 以及制作编辑 Rebecca Anderson。在整个编写过程中，这些人始终陪伴着我们，在他们的支持、建议和评论下，本书才得以顺利出版。

最后感谢我们的妈妈、爸爸、Tugce，特别是我深爱的 Funda，她给了我永无止境的支持和热情。

—Mert Caliskan

首先要感谢妻子 Betul，在我撰写本书期间，她一直默默地给予支持。还要感谢我的同事 Muammer Yucel 给予的鼓励和动力。本书的问世得益于我广泛的 Spring 应用程序框架经验，也得益于以下各位编辑的帮助以及他们提出的宝贵意见：项目编辑 Charlotte Kughen、技术编辑 Chad Darby 和 Krishna Srinivasan、文字编辑 Kim Cofer 以及制作编辑 Rebecca Anderson。

—Kenan Sevindik

序 1

我对 Wrox 的图书具有美好的回忆。12 年前，一本 Wrox 的图书就开始介绍 Spring。在那之前，我参与了多本 Wrox 图书相关章节的编写工作，并且作为评论家对许多 Wrox 图书进行了评论。

许多的回忆都在我的脑海中留下了深深的烙印，其中许多都涉及哲学和格式问题，最令我兴奋的是在 2000 年成为一名 Wrox 作者。独特的编排方式为传授知识、鼓励作者突出重要知识点以及促进系统解释提供了一种有效的结构。它强调实际的代码示例——这也是传授编程知识最有效的工具。

在 Mert Caliskan 和 Kenan Sevindik 撰写的《Spring 入门经典》一书中，这些优点显而易见。本书的结构非常合理，并且提供了大量示例，其中包括使代码正确运行所需的清晰指令以及一步一步详细的解释。

就像构建企业级 Java 应用程序一样，Spring 也是一个内容广泛的主题，想在一本入门级的图书中全部覆盖 Spring Framework 的核心内容几乎是不可能的。Mert 和 Kenan 在选择重点内容方面做了很好的工作。如果想要成为一名 Spring Web 开发人员，掌握本书所介绍的内容是非常重要的，包括核心 Dependency Injection 容器、MVC 框架(特别强调了 REST)、关系数据访问和事务管理、用来自定义应用程序行为的 AOP 和 Spring EL 的使用，以及 Spring 4.0 如何包含了 Java 8 中的重要语言扩展。虽然 Spring Security 是唯一的 Spring 子项目，但本书还是为构建 Spring 系统知识提供了坚实基础。

内容的级别同样有很强的针对性。本书对重要的背景信息进行了明智的搭配(例如，事务的 ACID 属性)，并且对如何使用 Spring 完成相关操作进行了详细讲述。虽然本书假设读者没有任何 Spring 的相关知识，但并没有浪费过多的篇幅来介绍编程问题，因为这些问题在很多的入门书籍中都有详细的介绍。

虽然我没有亲自参与本书的撰写工作，但我一直在关注 Spring 的发展。读了这本书，可以让读者了解到为什么 Spring 会如此重要。在过去 12 年里，Spring 的核心概念逐步建立起来：例如，一致性、事务管理的轻量级方法以及依赖注入的中心原则。当读者理解了“Spring 路线”之后，就可以快速掌握其他 Spring 技术。

如果你是一名 Java Web 开发人员，但仍然不熟悉 Spring，那么你会发现 Spring 可以让工作更加轻松。所以我推荐本书作为学习 Spring 的起点。

序 2

Java 的前景让我感到非常惊讶！即使在领导 Spring Framework 项目开发 11 年之后，我仍然在学习新的应用领域、新的优化方案以及新的系统体系结构。软件开发经常在发生变化，而 Spring 不仅是跟随变化——Spring 也在推动着变化。Spring 社区经常在相关的趋势变成大家讨论的焦点之前就已经预见了这些趋势；例如，在术语“微服务”产生之前，人们就已经开始使用 Spring 构建微服务体系结构。非常有趣的是，在 2014 年，Spring 框架背后的原始设计决策再次显示了其力量：引入用来将该框架与应用程序服务器体系结构相分离的机制被证明在适应现代嵌入式中间件方面是非常有用的。

Spring 之后的一项关键任务将最新的 Java 编程模型样式应用到尽可能广泛的部署环境中。在 Java 社区，我们往往给自己套上一件“紧身衣”，而不允许自己使用 API 和框架的当前版本。这些约束的主要原因在于应用程序服务器平台，这些平台包含了 JDK 和 Java EE API 的过时版本，且落后了太多年。在使用 Java 8 时，该问题就显得尤为突出：使用 lambda 表达式、新的日期时间类型、集合流等，此外，Java 8 还提供了许多用来跨代码块改变源代码风格的功能。作为社区，我们需要将这些新功能带入当前的环境中——不仅是用来开发，而是完整的生产支持。

通过使用 Spring 4，为 2014 以及以后创建了一条新的基线。Spring 的全面面向注解的编程模型是经过严格测试和精细调整的，为尽可能保持一致，多年来，Spring 被设计为稳定的基础。在每年发布的 Spring Framework 4.x 版中，都会对框架的功能进行扩展和重定义，从而作为自然与 Spring 4 基础完全兼容的组件。此外，该功能在 JDK 6+环境中也可用：用于更新现有应用程序，用于部署到现有的数据中心以及用于使用了保守 JDK 策略的环境。同时，你还会看到面向 JDK 8+的相关功能，特别是针对面向数据流的编程和反应体系架构。

最后，我提一些个人建议，以便帮助你在自己新开发的项目中充分利用 Spring：

- 首先，购买一本介绍核心 Spring 及其设计理念的书籍。通过阅读该书，你将学习更多相关的知识。
- 不要轻易妥协。应该基于你对应用程序域的理解来设计清晰的应用程序体系结构。
- 使用最新的 Java 基础结构：JDK 8、Spring 4、Tomcat 8、Jetty 9 等。
- 在开发自己的项目时不要忘记更新框架和库，特别是 Spring，其设计目的就是易于在应用程序中进行升级，而独立于任何服务器安装。

《Spring入门经典》重点介绍了Spring Framework 4背后的关键原则，是进入现代Spring开发殿堂的非常棒的一本书。所以请从人为约束中解放自己，充分享受使用Spring进行开发的乐趣！

—Jürgen Höller

前 言

Spring Framework 是一个得到了 Apache License 2.0 许可的开源的企业级应用程序框架，为构建满足企业级需求的应用程序提供了大量的工具集。推出该框架的原因是在使用 J2EE(Java Enterprise Edition 的前一版本)进行开发时会提高复杂性。使用 EJB(Enterprise Java Bean)创建和部署 Bean 是沉重的负担，原因是为了创建一个 Bean，除了 Bean 定义之外，还需要创建主程序和组件接口。在当时，由于在 EJB 世界中没有依赖注入的概念，因此查找方法是找到对象或资源的唯一方法。

Spring Framework 的第一个版本是基于 Rod Johnson 和 Jorgen Holler 在 *Expert One-on-One J2EE Design and Development without EJB*(Wrox, 2004)一书中所公布的代码。该书的主要思想是减少使用 EJB 进行企业级应用程序开发的复杂性。Spring 的第一个版本于 2002 年发布，随后在 2004 和 2005 年分别发布了里程碑式版本。1.0 版本引入了轻量级应用程序框架以及相关功能，比如 JDBC 抽象、对象关系映射工具支持、事务管理、调度和邮寄抽象实现以及 MVC Web 框架。

2.0 和 3.0 版本在框架中引入了一些重要功能，比如 AspectJ 支持和 REST 支持。在撰写本书的时候，4.0 版本已经发布，并且符合由 Java Enterprise Edition 7 和 Java 8 Standard Edition(Java 的最新版本)提供的规范。在本书中，将尝试覆盖包含框架最新版本的相关功能。

Spring 因为其核心 DI(Dependency Injection, 依赖注入)模式而逐步流行起来，而该模式也被称为控制反转(Inversion of Control, IoC)。因为面向对象编程在对象之间引入了关联，所以 DI 方法尝试通过提取这些依赖项的管理而实现松耦合设计，从而导致简单且可管理的实现。Spring 的另一个引人注目的特征由 AOP(Aspect-Oriented Programming, 面向方面编程)提供，AOP 提供了一种在应用程序中实现横向关注点的非常好的方法。每一个 Web 应用程序都需要诸如异常管理、身份验证、授权、登录以及缓存之类的功能。DI 和 AOP 的概念将分别在第 2 章和第 8 章详细介绍。

借助这些主要功能，Spring 提供了易于测试和重复使用的代码，并且与供应商无关，因为它可以很容易地在应用程序服务器之间移植，比如 WebLogic、JBoss 和 Tomcat。通过使用分层的体系结构，可以解决复杂企业级应用程序开发的不同部分。

可以肯定地讲，Spring Framework 已经成为开发 Java 企业级应用程序的事实标准。在本书中，我们将学习该框架所带来的所有主要功能。

本书读者对象

如本书书名所示，本书在初学者水平上介绍了Spring Framework，并且在需要的地方介绍了一些中等难度的概念。解释了该框架提供了什么，并通过一些易于理解的示例演示了如何实际使用这些功能。书中介绍的内容适用那些还没有尝试使用Spring Framework或者还不是完全了解其核心及其子项目的企业级应用程序开发人员。此外，本书对于从事中层管理的人也非常有帮助，可以使它们对企业级应用程序开发的组成部分有更深刻的了解，并且体会到Spring Framework的不同风格。

我们假设读者了解Java语言原则的一些基本知识，并能够使用集成开发环境(IDE)开发Java代码，比如Eclipse、NetBeans或其他环境。

本书并不适用于那些不熟悉Java语言和基本面向对象编程概念的读者。此外，对于那些并不热衷于使用Java进行企业级应用程序开发的读者来说，本书也没有什么吸引力。

本书主要内容

Spring 通过使用 XML 和 Java 注解提供了一个全面的配置模型。在本书中，演示了基于注解配置和基于 XML 配置，以便为读者提供所有可能的配置方案。

本书首先重点介绍了 Spring Framework 的核心概念和功能。使读者首先了解基本概念，比如非入侵性的 POJO(Plain Old Java Object)编程模型方法，然后给出了依赖注入模式(即控制反转)的定义。此外，Spring 还为构建 Web 应用程序提供了 MVC 实现；我们将详细介绍所提供的相关功能。除了讲解 Spring Framework 所提供的事务机制之外，还将介绍 Spring Framework 的 Java Database Connectivity 和 Java Persistence API 功能。随后，我们将讲述 Spring 为进行测试驱动开发所提供的功能。然后介绍面向方面编程，从而让读者了解 Spring 如何处理系统横切关注点的实现。Spring 表达式语言、缓存和 REST 风格的 Web 服务提供了 Spring Framework 的大量功能，我们将通过一些真实世界的例子介绍这些功能，读者可以在几分钟之内完成开发。我们还将研究 Spring Security，它已经成为在企业级应用程序中应用安全约束的事实标准。

最后，本书介绍框架4.0版本所提供的最新功能。我们介绍了最好的版本，因为本书通过关注框架的最新版本介绍了Spring的主要功能。

下面将简要介绍本书13个章节中的主要内容。

Spring 的依赖注入和配置

本书涵盖了Spring Framework的核心概念，比如DI (也称为IoC)及其配置模型。面向对象编程原则描述了实例化的对象之间的关联。DI模式的主要目的是这些对象不再彼此相互

依赖。这样就可以达到松耦合的目的，从而提供代码的可用性和可维护性。DI的想法源自 Martin Fowler，并且变得非常流行而被广泛采用。此外，在一些流行的框架中也使用了DI，比如PicoContainer、Guice和Spring。

DI 模式与其他模式类似，比如 Factory 或 Strategy。我们可以说，如果使用 Factory 模式，那么对象的实例化仍然由 Factory 定义来完成，但如果使用 DI，则由外部的其他组件或框架完成。另一方面，如果使用 Strategy 模式，则通过相同接口的多个对象(这些对象内部包含了具体的实现过程)的帮助，当前的实现过程被替换。然而，如果使用 DI，则包含这些实现过程的对象将被装配，而不管定义的实现过程是什么。

从 Spring Framework 2.0 版本开始，除了支持自定义命名空间外，还可以使用 XML 架构配置 DI 机制。而从 2.5 版本开始，Spring 利用 Java 5 和相关的注解支持组件的自动发现以及基于注解的组件的自动装配，并利用生命周期注解来挂钩初始化和销毁回调。我们将介绍该框架所引入的所有注解。

Spring 4.0

第 13 章介绍 Spring 最新版本的前沿功能。4.0 版本借助相关的功能提供了对 Java 8 的支持，比如 lambda 表达式和方法表达式。此外，它还为新的 DateTime API 提供了注解驱动的时间格式。4.0 版本还符合 Java EE7 规范，比如 JPA 2.1、JMS 2.0、JTA 1.2、Bean Validation 1.1 等。我们讲述这些功能是为了帮助读者更深入地了解最新版本。

Spring 注解

Spring 主要为依赖注入以及框架的其他功能提供了基于注解和基于 XML 的配置。我们首选通过一些示例给出传统的 XML 配置，从而展示 XML 配置带给开发人员的复杂性和负担。在本书中，我们尽可能使用注解，以便减轻读者的工作量并以一种实用的方法使示例更易于理解。此外，我们还尽量在示例中覆盖该框架所提供的所有注解。

从 Spring 2.5 开始，JSR-250 Annotations API 也被支持。它是 Java Specification Request，可用于注解开发，使 Spring 与 Java 的 EE 版本更加兼容。

使用 JDBC、JPA 和 ORM 提供 Spring 持久性支持

Spring 向开发人员提供了针对 JDBC 数据库访问的抽象。它通过减少用于连接管理的样板代码、异常/事务处理以及语句的准备和执行，从而简化了开发。

虽然 Spring 没有提供内置的对象关系映射(ORM)框架，但它支持那些遵守 JPA 规范的知名 ORM 框架，比如 Hibernate、EclipseLink、TopLink 等。本书将通过一步一步的工作示

例介绍关于 JDBC、JPA 和 ORM 的所有相关功能。

Spring 表达式语言

Spring 表达式语言(Spring Expression Language, SpEL)是一种功能强大的表达式语言，可以在运行时在对象图中进行导航。所有的 SpEL 表达式都可以使用 XML 配置或注解进行定义。而 SpEL 使用的语法与统一的 EL 相似，却提供了更多增强的功能。

SpEL 可用来执行属性或 Bean 注入以及方法调用。除了正则表达式之外，它还支持数学和关系操作符。Spring 的其他子项目也可以非常容易地使用 SpEL，比如安全性或缓存。此外，SpEL 也是与技术无关的，所以可以单独使用而不必依赖于 Spring 上下文。本书将通过一些示例介绍这些功能。

Spring 事务管理和 AOP

数据完整性是每个处理数据的系统的关键点之一。某些系统行为是会被系统用户所接受的，比如不正确的账户余额、丢失订单或者文档中缺少条目等。而事务管理提供了一种方法来实现数据的完整性。此外，Spring 还提供了扩展的事务管理，以确保企业级应用程序的数据完整性和一致性。第 6 章将使用示例代码来演示 Spring 如何提供声明式和编程式事务机制、本地事务和全局事务的区别、事务上的传播规则以及框架的提交/回滚体系结构。

为了实现横切关注点，Spring AOP 使用 XML 配置提供了 AOP 元素的简单定义，此外，它还支持 AspectJ 框架注解，从而使应用程序的配置变得更容易。

Spring MVC 和开发 REST 风格的 Web 服务

借助于 MVC(Model View Controller, 模型-视图-控制器)模式，可以非常容易地使用三层来定义企业级应用程序。其中“模型”是应用程序中用来处理应用程序数据逻辑的部分。“视图”则是用来显示数据的部分，而“控制器”则是用来处理用户交互的部分。Spring 提供了一个 MVC 子项目来处理 HTTP 请求，从而减少了访问请求参数、验证以及对话和模型更新所需的样板代码。此外，它还提供了一种便捷的方法来定义 REST 风格的 Web 服务(这些服务将通过先进的用户界面框架来使用)，从而使 Web 应用程序更加灵活。

Spring Security

Spring Security 是 Spring 的一个子项目，为基于 Java 的企业级应用程序提供了一流的

身份验证和授权支持。该项目的最初代号为 Acegi，随后合并到 Spring 项目组合。它符合 Servlet API，所以可以非常容易地将其与一个基于 Java EE 的 Web 应用程序相集成。此外，还可以与其他 Spring 框架相集成，比如 MVC，所以在功能使用上保持了一致性。第 12 章将为这些功能提供一些巧妙的示例。

Spring 缓存

从 3.1 版本开始，Spring Framework 通过在后台使用不同的缓存框架(比如 Ehcache 和 Hazelcast)，为企业级应用程序提供了一种透明的缓存抽象。

缓存的主要目的是通过执行参数对方法返回的结果进行缓存，从而减少目标方法的执行次数。

注意：

Spring Framework 是一个全面的工具集，想要在一本初学者水平的图书中介绍所有的子项目以及所有的功能集是不可能的。本书并不适用于那些想要通过详细解释对 Spring 子项目进行全面了解的读者。

此外，本书主要关注于框架的 4.0.5.RELEASE 版本，在撰写本书时这是最新的版本。因此，本书并没有介绍框架的较旧版本及其相关功能。

运行本书示例所需的条件

为了运行本书中的示例，需要以下条件：

- Java Development Kit (JDK) 8.0
- Maven项目构建和管理工具：通过基于Maven的项目结构，可以在www.wrox.com/go/beginningspring的Download Code选项卡中找到本书中的所有示例。

注意：

Maven 需要在本地环境(而不是 Java 运行时环境)中安装 Java Development Kit。

- Java Web 容器：为了运行本书中的 Web 项目示例，我们使用了 Tomcat 8.0.12 和 Jetty 9.2.3。
- Java IDE：本书中的示例通过 Eclipse IDE 实现，但因为使用 Maven 作为项目结构，所以可以非常容易地使用其他 IDE(比如 NetBeans 或 IntelliJ IDEA)来构建示例。

勘误表

尽管我们已经尽了各种努力来保证文章或代码中不出现错误，但是错误总是难免的，

如果您在本书中找到了错误，例如拼写错误或代码错误，请告诉我们，我们将非常感激。通过勘误表，可以让其他读者避免受挫，当然，这还有助于提供更高质量的信息。

请给 wkservice@vip.163.com 发电子邮件，我们就会检查您的反馈信息，如果是正确的，我们将在本书的后续版本中采用。

要在网站上找到本书英文版的勘误表，可以登录 <http://www.wrox.com>，通过 Search 工具或书名列表查找本书，然后在本书的细目页面上，单击 Book Errata 链接。在这个页面上可以查看到 Wrox 编辑已提交和粘贴的所有勘误项。完整的图书列表还包括每本书的勘误表，网址是 www.wrox.com/misc-pages/booklist.shtml。

p2p.wrox.com

要与作者和同行讨论，请加入 p2p.wrox.com 上的 P2P 论坛。这个论坛是一个基于 Web 的系统，便于您张贴与 Wrox 图书相关的消息和相关技术，与其他读者和技术用户交流心得。该论坛提供了订阅功能，当论坛上有新的消息时，它可以给您传送感兴趣的论题。Wrox 作者、编辑和其他业界专家和读者都会到这个论坛上来探讨问题。

在 <http://p2p.wrox.com> 上，有许多不同的论坛，它们不仅有助于阅读本书，还有助于开发自己的应用程序。要加入论坛，可以遵循下面的步骤：

- (1) 进入 p2p.wrox.com，单击 Register 链接。
- (2) 阅读使用协议，并单击 Agree 按钮。
- (3) 填写加入该论坛所需要的信息和自己希望提供的其他信息，单击 Submit 按钮。
- (4) 您会收到一封电子邮件，其中的信息描述了如何验证账户，完成加入过程。

注意：

不加入 P2P 也可以阅读论坛上的消息，但要张贴自己的消息，就必须加入该论坛。

加入论坛后，就可以张贴新消息，响应其他用户张贴的消息。可以随时在 Web 上阅读消息。如果要想让该网站给自己发送特定论坛中的消息，可以单击论坛列表中该论坛名旁边的 Subscribe to this Forum 图标。

关于使用 Wrox P2P 的更多信息，可阅读 P2P FAQ，了解论坛软件的工作情况以及 P2P 和 Wrox 图书的许多常见问题。要阅读 FAQ，可以在任意 P2P 页面上点击 FAQ 链接。

源代码

在理解本书中的示例时，你可能会选择手动输入所有的数据和代码或者使用本书提供的源代码和数据对象文件。本书中使用的所有数据和源代码都可以从 <http://www.wrox.com/go/beginningspring> 下载。你会发现每个示例练习活动所需的数据集都具有一个下载图标和表明数据文件名称的注释，这样你就能知道它可供下载并且能够在下载文件中轻易定位到。