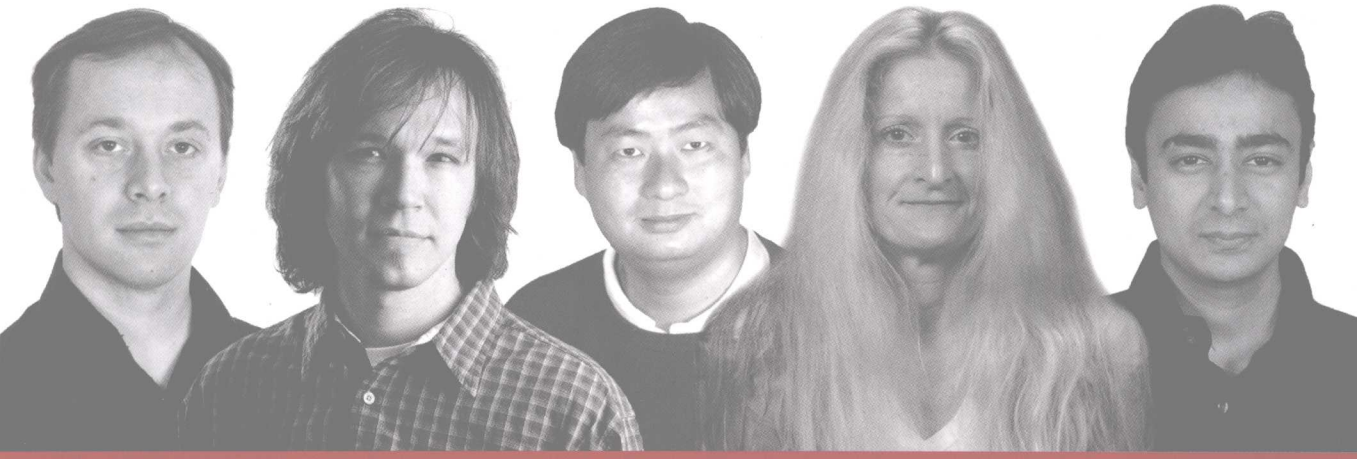
PROGRAMMER TO PROGRAMMER™



Beginning Spring Framework 2

Spring Framework 2 入门经典

(美) Thomas Van de Velde
Bruce Snyder
赵利通
等著
译

wrox™
WILEY
清华大学出版社

Spring Framework 2 入门经典

(美) Thomas Van de Velde 等著
Bruce Snyder
赵利通 译

清华大学出版社
北 京

Thomas Van de Velde, Bruce Snyder, et al
Beginning Spring Framework 2
EISBN: 978-0-470-10161-2

图书在版编目(CIP)数据

Spring Framework 2 入门经典/(美)威尔德(Velde, T.V.)，斯尼德(Snyder, B.)　等著；赵利通　译.
—北京：清华大学出版社，2009.7
书名原文：Beginning Spring Framework 2
ISBN 978-7-302-20208-0

Ⅰ. S…　Ⅱ.①威…②斯…③赵…　Ⅲ. JAVA 语言—程序设计　Ⅳ.TP312

中国版本图书馆 CIP 数据核字(2009)第 078936 号

责任编辑：王　军　谢晓芳
装帧设计：孔祥丰
责任校对：成凤进
责任印制：杨　艳
出版发行：清华大学出版社　　**地　　址**：北京清华大学学研大厦 A 座
http://www.tup.com.cn　　**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：三河市溧源装订厂
经　　销：全国新华书店
开　　本：185×260　**印　张**：28　**字　数**：681 千字
版　　次：2009 年 7 月第 1 版　　**印　　次**：2009 年 7 月第1 次印刷
印　　数：1～4000
定　　价：58.00 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题，请与清华大学出版社出版部联系调换。联系电话：(010)62770177 转 3103　　产品编号：026504-01

作者简介

Thomas Van de Velde 在开发众多行业内的高流量、面向公众的 Web 站点方面具有丰富经验。作为一家全球领先的技术咨询公司的顾问和项目经理，他已经完成了法国网上税务申报系统和美国一个最大的体育网站的创建工作。Thomas 对于探索如何在企业中利用开放源代码颇具热情，而在闲暇时，他会与妻子和女儿在一起居住的 California 南部进行冲浪。

Bruce Snyder 已经多年从事企业软件开发，他被公认为开放源代码软件方面的佼佼者。Bruce 熟悉多种技术(包括 Java EE、消息传输和面向服务体系结构)。除了是 IONA Technologies 的主要工程师以外，他还是 Apache Geronimo 的创建者之一，并且是 Apache ActiveMQ、Apache ServiceMix 和 Castor 等技术的开发人员。Bruce 是多个 JCP 专家组的成员，也是 Wrox 出版社出版的 *Professional Apache Geronimo* 一书的合著者。Bruce 还经常在一些产业研讨会上发言，这些会议包括 Colorado Software Summit、TheServerSide Java Symposium、Java in Action、JavaOne、ApacheCon、JAOO、SOA Web Services Edge、No Fluff Just Stuff 和各种 Java 用户组。Bruce 和他的家人居住在美丽的 Colorado 州 Boulder 地区。

Christian Dupuis 就职于一家世界领先的咨询公司，他也是 Technical Architecture 能力组的一名成员。Christian 的身份是技术体系结构设计者和实现领导，负责设计和实现多通道的、任务关键类型的、在各个层面都利用 Spring 及其他开放源代码架构的财务应用程序。Christian 是 Spring IDE 开放源代码项目(http://springide.org)的共同领导之一，这个项目为 Spring Portfolio 提供工具支持。

Sing Li 成长在微处理器时代，在 20 世纪 70 年代晚期就迷上了微型计算机。Sing Li 的第一台个人计算机是其通过在 Popular Electronics 杂志后面刊登广告的厂商邮购而获得的 DIY 型 Netronics COSMIC ELF 计算机，该计算机价值 99 美元，拥有 256 字节的内存。Sing Li 是一名已经具有 25 年从业经验的系统开发人员，他致力于为开放源代码软件做出贡献，并且是 Java 技术和嵌入式及分布式系统体系结构方面的自由撰稿人。Sing Li 定期为几个流行的技术杂志和电子杂志撰稿。他创建了 Internet Global Phone，这是最早的 Internet 电话之一。Sing Li 自己创作或与其他人合著了多个技术领域——包括 Geronimo、Tomcat、JSP、servlets、XML、Jini、媒体流、设备驱动器和 JXTA 等——的大量书籍。

Anne Horton 在软件行业已经工作了 24 年，从事过软件工程师、教材技术编辑、撰稿人和 Java 体系结构设计师等工作。目前，Anne Horton 就职于 Lockheed Martin 公司，而在周末时则经常与 Sing Li(作者)和 Sydney Jones(编辑)一起工作，撰写关于前沿技术的书籍，比如本书。您可以通过如下电子邮箱向她发送电子邮件：abhorton@comcast.net。

Naveen Balani 是 IBM India Software Labs(ISL)的一名体系结构设计师，他负责领导 ISL 开发的 WebSphere Business Service Fabric 产品的设计和开发工作。Naveen Balani 喜欢研究新兴技术，并为 IBM developer works 定期撰稿，文章主题包括 Web 服务、ESB、JMS、SOA、体系结构、开放源代码架构、语义网络、J2ME、劝导式计算、Spring 系列、AJAX 和多种 IBM 产品。您可以向 naveenbalani@rediffmail.com 发送电子邮件与其进行联系。

前　言

开发基于服务器的系统一向比较复杂。现在，一场悄悄兴起的革命正在试图简化基于服务器的系统的开发工作。这场革命关注一些脆弱标准(比如 Java 2 Enterprise Edition，即 J2EE)的轻量级替换方法，它的核心是基于简单旧式 Java 对象(Plain Old Java Object，POJO)进行设计，对处理与业务逻辑(比如登录和事务)正交的关注点的代码进行解耦，简化编码，以及实现长期可维护性。这场革命的中心就是称为 Spring 架构的 Java 架构。

本书讲解 Spring 架构。事实上，本书是一本关于 Spring 架构的介绍性书籍。与其他架构不同，Spring 架构对于已经具有解决商业问题经验的开发人员(可能他们使用的是传统服务器架构，如 J2EE 1.4)最为有用。与其他介绍性书籍不同的是，本书面向已经具有 Java 应用程序开发经验、但是还不熟悉 Spring 架构的开发人员。

从许多方面来讲，本书都是一本与众不同的技术书籍，因为我们现在生活在一个与以往不同的技术世界。现今，由于网络的可用性、可访问性及功能的不断增强，许多围绕数据驱动、基于服务器的系统的活动和应用程序应运而生。我们不再能够假定每个基于 Web 的系统都通过某种方法被绑定到大型主机的数据上；同样也不能假定每个基于服务器的应用程序都会带有面向企业的功能，比如复杂目录服务、事务服务器或者庞大的安全子系统。事实上，多数现代服务器端的 Java 开发都围绕一个需求——设计小型而敏捷的系统，这样的系统可以在业务改变时适应业务的需要。Java 技术的用户基础，特别是基于服务器的 Java 技术的用户基础已经具有临界规模，并且呈现出多样化态势。Spring 架构就是应对这种多样化用户分布需要的一种架构。

在当前的新时代中，Spring 架构并不仅仅是一组 API。它是一个不断进化的技术平台，正在迅速适应多样化的用户。Spring 架构在许多方面都领先于开发技术社区，并且它更加贴近于使用这种技术并从中获益的人员。

本书的作者是一组行业专家，他们将提供实际操作经验，以此来讲解 Spring 架构的功能。本书首先集中讨论这种架构的基础知识，以便您能尽快上手。然后，本书将讨论目前使用的 Spring 架构的各个应用领域，这是本书最重要的一部分，因为您很可能会从中找到一些可以立即应用到您的系统中的内容。本书将提供结合各个应用领域的实际应用程序示例，从中可以深入了解如何开发、测试和部署一个基于 Spring 的完整系统。

本书的作者组成了难得一见的 Spring 实践专家的合作团队。Thomas 刚刚完成了有史以来最庞大、最复杂的 Web 网站之一 NFL.com 的体系结构和实现，该网站基于 Spring；此外，他也使用 Spring 为法国政府完成了网上税务申报系统。Bruce 在空闲时间帮助将 Spring 架构的基础结构移植到 Apache Geronimo(领先的开放源代码应用程序服务器，也是 IBM 的 WebSphere Server Community Edition 的标志性服务器)。Christian 是开放源代码的 Spring IDE——Spring 开发人员首选的 IDE 工具——的合作创建者之一。Anne 自 Bell Labs 的 AT&T Unix 时代以来就一直从事于开发多用户商业应用程序系统。Naveen 每天都在为

新出现的市场创建和维护支持 Spring 的、基于 Web 的服务和应用程序系统。Sing 在为 SUN 工作很久以前就开始撰写有关开发技术的材料，他设法将 J-A-V-A 这几个字母连接到一起并读成 Java。将这些与 Rod Johnson——Spring 的创建者——富有启示性的指导结合起来，您就拥有了一本适时讨论不断发展的 Spring 技术平台的相关背景知识的书籍。本书可以帮助您快速学会使用这种功能丰富的软件技术。

0.1 本书结构

要在最短的时间内学会使用 Spring 架构，您所需要的只是本书的下载源代码、第 1 章、以及附录 A 和 C。这些章节和附录提供了足够的技术信息，可以帮助您快速掌握如何使用 Spring 2 架构。如果您是一名有经验的 Java 开发人员，那么可以在工作时根据需要学习其他内容。

然而，如果想要迅速地探究和学习其他当前关注的、支持 Spring 的应用领域，那么可以阅读本书的其余部分。

您可以自己确定阅读本书的速度以及阅读各章的顺序。我们尽力使本书适合各种读者的需要。下面对本书各章进行了简要的描述。

- **第 1 章， Spring 2 快速入门**：这一章是快速介绍的章节。通过一系列实际示例，您可以立即开始使用 Spring 架构。这里使用有效的代码介绍了一些基础概念，比如依赖注入(dependency injection，DI)、控制反转(inversion of control，IoC)和面向切面编程(aspect-oriented programming，AOP)，但是您也可以修改这些代码。本章一步步地讲解了 Spring 的配置特性，如 bean 连接。在阅读完本章后，您就可以轻松编写自己的基于 Spring 的应用程序。
- **第 2 章，设计 Spring 应用程序**：开发服务器端的业务应用程序应该十分简单，就如同创建执行必要的业务逻辑操作的 Java 对象一样。至少我们在大学里学到的知识是这么描述的。然而遗憾的是，多数实际的商业应用程序都要求定制访问数据库、安全子系统、事务、目录服务等的复杂 API 的编程。使用 Spring 时，您可以完全地追根究底并且由复杂回归简单。本章将揭示如何编码和测试只实现了业务逻辑的简单 POJO，以及如何将 Spring 用作基于服务器的应用程序的核心。POJO 是一个称为 Pix 的、基于 Web 的照片-相册-管理系统的核心对象(域对象)。这个 Pix 系统是实际的 Spring 系统，本书通篇都将使用这个系统作为示例。
- **第 3 章，使用 JPA 的 Spring 持久性**：在使用 Spring 时，向关系数据库添加存储和获取数据的功能并不是痛苦的工作。本章将揭示通过添加一些 Java 注释，您可以轻松地使用第 2 章中创建的 POJO 支持数据库。这里称这种支持技术为 Java 持久化 API(Java Persistence API)，本章将介绍大量关于 JPA 及其与脆弱的 Java EE 5 规范之间关系的内容。此外，您将了解如何将一般意义上的数据访问——这里主要讨论了 JPA——集成到 Spring 架构中。您将把 Pix 中的域对象 POJO 转换成驻留在关系数据库服务器的表中的持久性记录。

- **第 4 章，使用 Spring MVC 构建 Web 页面：**当您想要创建基于 Web 的应用程序时，需要创建可以与用户交互的 Web 页面和表单。当然，这些表单和页面与存储在关系数据库服务器的后台上的数据绑定在一起。通过 Spring，您可以使用在第 3 章中创建的支持数据库的 POJO 来处理后端。我们将揭示在使用 Spring MVC 时，基于这些 POJO 可以轻松地创建表单和 Web 页面。Spring MVC 的设计遵循模型-视图-控制器这种模式，并从实际数据(POJO)解耦了表示技术。在本章中您将学到 Spring MVC 体系结构，并且将练习使用这种体系结构。您将创建使用 Java 服务器页面(Java Server Page)进行表示的 Pix Web 页面。这些页面使用 PIX 持久性 POJO 注册用户及创建相册。
- **第 5 章，高级 Spring MVC：**Spring MVC 用于从底层模型数据解耦表示技术。在这个有关高级话题的一章中，您将生成相册的一个 Adobe 便携文档格式(portable document format，PDF)的副本，此时这种解耦的灵活性将很明显。通过一些简单的配置改动，您将学到如何通过生成简易供稿(Real Simple Syndication，RSS)表示您的相册，这样其他的用户和 Web 站点就可以与(您通过 Pix 发布的)新照片关联起来。此外，本章还讲解了其他一些高级的 Spring MVC 技术，包括支持文件上传、处理多页面入口表单、提供个性化服务以使每个用户可以定制其体验，以及向您的 Spring MVC 应用程序添加国际支持。
- **第 6 章，Spring Web 流程：**当您使用 Spring 和 Spring MVC 时，创建基于 Web 的服务器应用程序可以十分简单。但是并非所有基于 Web 的服务器应用程序都很简单——许多这样的应用程序都包含复杂的业务逻辑和多个相互关联的用户接口页面。此时停止使用 Spring MVC，而开始使用 Spring Web 流程(Spring Web Flow)。使用 Spring Web 流程时，您可以创建具备高度复杂的业务逻辑流程的应用程序，但是却不必为了支持它们而在服务器端累积杂乱而复杂的代码。通过 Spring Web 流程，您可以将复杂的应用程序逻辑分解为应用程序页面和业务逻辑的流程和子流程。您可以在 XML 描述符文件里设计和维护这些流程，从而简化其改动和升级。因为您将会把一些 Pix 应用程序的应用场合从 Spring MVC 迁移到 Spring Web 流程，所以本章中您将执行许多 Spring Web Flow 动作。您将亲自学到如何使用 Spring Web 流程对数据验证和会话管理的内置支持，以便极大地简化在支持这些生产级功能时通常需要使用的复杂编码。
- **第 7 章，Ajax 和 Spring：DWR 集成：**Web 2.0 指的是一种具备高度交互性的用户体验，这种体验可以使一个基于 Web 的应用程序让人感觉是本地运行的应用程序。这是与标准的基于 Web 的应用程序的一个极大区别，因为使用后者的用户必须在每个页面切换之间等待服务器的响应。使 Web 2.0 成为可能的神奇因素被称为 Ajax，或者异步 JavaScript 和 XML(Asynchronous JavaScript and XML)。在以前，创建 Ajax 应用程序要求具备浏览器端 JavaScript 编码、数据格式变换和网络编程方面的专门知识，但是，新的软件架构极大地简化了这个任务。本章说明了通过使用一个支持 Spring 的、称为 DWR(Direct Web Remoting，直接 Web 远程处理)的 Ajax 架构，如何使您的 Web 应用程序支持 Ajax。在本章中，通过使用 DWR 和 Spring，您将为查看 Pix 相册而创建一个支持 Ajax 的、具备高交互性的 Web 用户

界面。您将利用 Spring MVC 的能力从模型(域对象)解耦表示技术(如基于 Ajax 的 DWR)，并提供具备高度交互性的 Web 2.0 用户体验，以访问在前面几章中操作的相同的服务器端 Pix POJO 集。

- **第 8 章，Spring 和 JMS——消息驱动的 POJO**：当子系统之间相互发送数据时，发送方和接收方并不总是同时可用。在商业系统中您不希望在子系统不可用时丢失数据。例如，想象一个网上订购系统，在子系统不可用时它会损失订单；显然您不想设计这样一个系统。软件行业早就通过使用消息队列解决了这个问题，但是早期管理健壮队列(称为 MQ 代理程序，MQ broker)的服务器只有大型企业才可以负担得起。随着开放源代码的 ActiveMQ 代理程序(ActiveMQ broker)的出现，实现了大范围的可靠消息传输。Spring 通过使 POJO 支持队列来支持消息队列。这些 POJO 被称为消息驱动的 POJO，它们极易创建。在本章中，将介绍 Java Message Service(Java 消息服务，JMS)API——它如何与 MQ 代理程序关联，以及如何使某些 POJO 支持 MQ 以实现 PIX 系统的可靠操作。
- **第 9 章，Spring Web 服务和远程处理**：现在，在使用信息技术时几乎不可能不遇到 Web 服务。Web 服务的独特之处在于，只要可以使用浏览器，您就可以访问 Web 服务。这意味着通过 Internet 可以很容易地使用 Web 服务，即使存在可能会阻止其他服务调用机制的干涉性安全防火墙。这样，商业系统可以通过 Internet 与其他商业系统进行交流，而不需要额外的硬件或网络投入。创建 Web 服务通常是一个复杂的多步骤过程。通过一个称为 XFire 的开放源代码 API 库的协助，Spring 使您可以在非常短的时间内创建 Web 服务。事实上，只要改变一些 XML 配置文件，就可以将已有的 POJO 接口作为 Web 服务提供。本章将介绍 Web 服务的概念和术语，并使您通过使用 Spring 和 XFire 实现一个基于 Pix 的成员注册 Web 服务。
- **第 10 章，Web 服务的使用者及与.NET 的互操作性**：因为 Web 服务可以通过 Internet 访问，所以很可能您需要创建调用他人创建的 Web 服务的客户端。有了 Spring 和 XFire 的帮助，您可以轻松地完成这项任务。本章将逐步地说明如何创建这样的客户端——它们常被称为 Web 服务的使用者(Web service consumer)。Web 服务并不是 Java 开发人员的唯一领域。事实上，可以使用运行在完全不同的平台上的完全不同的技术实现使用者和服务，这是 Web 服务的一个优秀特性。这意味着 Java 用户可以调用基于 Microsoft .NET 的服务，反之亦然。本章将显示如何创建可以互操作的系统。将为 Pix 创建一个基于 Java Spring 的使用者，使其访问通过.NET 创建的、执行远程电子邮件验证的 Web 服务。此外，还将创建一个基于.NET 的 Web 服务使用者，这个使用者使用 C#来访问 Pix 成员注册 Web 服务。
- **第 11 章，使用 Spring IDE 进行快速 Spring 开发**：本章将介绍 Spring IDE，这是一个用于简化 Spring 开发的工具。Spring IDE 是用于 Eclipse 平台的一个插件的集合，它添加了对编辑 Spring XML 配置文件以及对这些文件添加验证和可视化的支持。此外，它还提供了综合工具来帮助您学习 Spring AOP 和 Spring Web 流程。本章将提供一个逐步的安装向导和如何在日常工作中使用 Spring 架构不同功能的详细描述，以此显示如何开始使用 Spring IDE。

- **第 12 章，Spring AOP 和 AspectJ**：面向切面编程(AOP)隔离了横切代码模块的关注点，并且解耦了处理这些关注点的紧密代码。在生产中，处理安全、登录、事务等的代码被混杂进所有的代码模块中，从而使理解、修改和维护它们变得十分困难。AOP 使您可以隔离这类代码并独立维护它们，从而使业务逻辑代码保持简单状态并远离那些让人困惑的元素。AspectJ 是首选的 AOP 编程平台，它一问世就强烈地冲击了世人的想象力。Spring 支持的 AOP 集成了 Aspect 的最佳特性，并对所有的 Spring 应用程序开放这些功能(Aspect 的开发人现在正式成为 Spring 社区的活跃的领导，这一点确实提供了帮助)。本章将介绍 AOP，您将看到它如何有助于提高甚至于最大型的项目的敏捷性和适应性，以及它如何极大地简化长期维护。您将练习使用 Spring AOP 和 AspectJ，并且将在 Pix 系统内富有成效地使用它们来解决横切关注点的问题。
- **第 13 章，更多的 AOP：事务**：同时执行多个相互关联的操作的系统经常会失败。当一个或多个这样的相关操作失败时，可以选择编写非常复杂的失败处理代码来应对所有不同的情况，或者选择停止操作并重新开始。当然，要重新开始通常是说起来容易，做起来难。但是不必担心：软件行业已经为此专门发明出了事务。在事务中，所有相关联的操作都必须成功，否则底层的事务管理系统会撤消事务中所有相关联操作的结果。Spring 通过一个统一的模型和 API 支持所有类型的事务。事务可以是局部的专属于服务器(比如一个 RDBMS 服务器)的某个实例，也可以是全局的——分布在几个网络化的服务器上。Spring 通过同样的统一化模型和 API 提供对局部和全局事务的支持。当向 POJO 添加事务时，Spring 允许您使用声明的方式执行添加操作，而不必修改任何 POJO 代码。本章将介绍事务的有关概念，并描述 Spring 如何使用拦截和应用 AOP 来简化事务处理，此外本章还揭示了如何通过修改 XML 配置文件轻松地使 PIX POJO 支持事务。
- **附录 A，Maven 2 基础**：Maven 2 是进行 Java 项目时首选的一个开放源代码的构建管理系统，对于支持 Spring 的 Java 项目尤其如此。本附录将介绍 Maven，描述如何安装 Maven，并且提供关于相关的 Maven 使用场合的详细选择，以便使您可以轻松而正确地使用这个功能丰富的工具。
- **附录 B，Spring 和 Java EE**：对于 Java EE 5 如何与 Spring 架构关联或与 Spring 架构竞争，一直以来都存在许多种说法，也存在不少困惑。本附录将讲解一些确切的事实，并具体比较这两种重要的平台。
- **附录 C，为代码示例做准备**：本附录提供关于安装和设置 Pix 服务器示例的详细指令。这个示例贯穿全书。

0.2　源代码

在读者学习本书中的示例时，可以手工输入所有的代码，也可以使用本书附带的源代码文件。本书使用的所有源代码都可以从本书合作站点 http://www.wrox.com/ 或 www.tupwk.com.cn/downpage 上下载。登录到站点 http://www.wrox.com/，使用 Search 工具

或使用书名列表就可以找到本书。接着单击本书细目页面上的 Download Code 链接，就可以获得所有的源代码。

注释：

由于许多图书的标题都很类似，所以按 ISBN 搜索是最简单的，本书英文版的 ISBN 是 978-0-470-10161-2。

在下载了代码后，只需用自己喜欢的解压缩软件对它进行解压缩即可。另外，也可以进入 http://www.wrox.com/dynamic/books/download.aspx 上的 Wrox 代码下载主页，查看本书和其他 Wrox 图书的所有代码。

0.3 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。

0.4 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 按扭。

(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 按扭。

(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。

注释：

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 页面上单击 FAQ 链接。

目　录

第1章

Spring 2 快速入门

第一次使用一种新的软件架构是非常令人兴奋的。Spring 2 就是这样一种令人兴奋的软件架构。但同时，它也是一种相当庞大的架构。为了能够在日常工作中有效地应用 Spring 2，必须先对以下问题有一定的了解：

- Spring 架构存在的原因
- Spring 架构要解决的问题
- Spring 架构的工作原理
- Spring 架构包含的新技术或新概念
- Spring 架构的最佳使用方法

本章可以帮助您尽快了解这些要点，并且能够尽快通过一些代码来使用 Spring 2 Framework。

本章内容如下：

- Spring 架构的历史和设计原理
- Spring 架构的典型应用
- 利用 Spring 架构连接 Java 组件来创建应用程序
- 了解 Spring 架构的自动连接(autowiring)能力
- 了解控制反转和依赖注入
- 了解 Spring 2 可用的 API 模块

读完这一章，您就有了基础从而为深入了解后面章节介绍的 Spring 架构的特定领域做好了准备。

1.1 Spring 简介

2002 年，Wrox 出版社在 J2EE 最为流行的时候出版了 Rod Johnson 编写的 *Expert One on One Java J2EE Design and Development*(ISBN：1861007841)。Rod Johnson 首次将 Spring 作为示例代码的主体，Spring 由此诞生。在那之前，创建一个大型企业级 Java 应用程序的常规的方法是利用 Java 2 企业版 1.3/1.4，然后遵循当时流行的 EJB 2.x 规范来创建复杂的软件组件——Enterprise JavaBean(EJB)。

注意：

更多有关 Spring 架构相对于 Java EE 的区别和改进，详见附录 B “Spring 和 Java EE”。

虽然 J2EE 很流行，但利用它来创建基于组件的 Java 服务器应用程序并不是一件容易的事。即使是对于一个很小的项目来说，构建 EJB 和创建没有 EJB 的应用程序都是一个复杂的过程。它涉及到许多冗长的代码，并且要求对一大堆的源代码进行管理。

Rod 对轻量级容器的阐述最大程度地降低了构建服务器端应用程序的复杂性，并且为沉闷的 J2EE 开发团体注入了新的活力。Spring——结合了既简单又充满开创性的概念，例如依赖注入(将会在以后的章节中讨论)——发挥了众多 Java 服务器端的开发者的想象力。

为了研究这些代码，诞生了一个基于 Internet 开发的活跃团队。他们主要对 Rod 公司的网站(interface21.com/)，以及相关的 Spring 架构的网站(springframework.org/)进行研究。

随着 Rod 和他团队的不懈努力，越来越多的开发者发现这种除了 J2EE 之外更加实用的和轻量级的开放源代码软件，该软件架构逐渐在世界范围内广泛采用。由于第三方插件的使用，该软件架构本身也日益丰富。

1.2 追求简洁

2007 年，Spring Framework 发布了第 2 版。此时，用 Spring 架构来代替服务器端的企业应用程序的开发已不仅仅只是一个概念，而是一个在世界范围内日益实践化的现实。Spring 对层次清晰的分离和应用程序组件的解耦的重视，以及其追求轻量级的理念和对降低开发复杂性的执着，为它在 Java 企业应用程序开发者的心中赢得了永久的一席之地。

Spring 的出现对全社会的开发者产生了如此重大的影响，以至于 Java Enterprise Edition (Java 企业版)的专家小组不得不对它进行重新设计。为了让用户更加满意，在 Java EE 5 中大大简化了创建 EJB 的工作。包括依赖注入在内的 J2EE 中的许多轻量级原理和方法已经不断与时俱进。

1.3 Spring 的应用

您很快会发现，使用 Spring 架构创建应用程序总是需要收集可重用的软件组件，然后再把它们组装成应用程序。在 Spring 中，这种组装组件的过程称为连接，该术语从电子硬件组件的类比中得出。连接的组件可以是为应用程序编写的 Java 对象，或者是 Spring API 库中许多预先编制的组件之一(或是由第三方供应商提供的组件，例如来自于 Hibernate 的事务管理器)。

理解 Spring 中组件的实例化以及连接的工作方式具有极为重要的意义。最好的方法是通过一个实际的示例来介绍 Spring 对象的工作原理。

Spring 架构操作模块化应用程序。第一步就是要创建这样一种应用程序。下一节将介绍如何把一个一体化的应用程序分解成各个组件，然后介绍 Spring 是如何通过灵活地连接组件来求和的。