

Spring实战（第4版）

作者：[美] Craig Walls 沃尔斯

[目录](#)

[版权信息](#)

[作者简介](#)

[版权声明](#)

[内容提要](#)

[关于本书](#)

[路线图](#)

[代码规范与下载](#)

[作者在线](#)

[封面插图简介](#)

[前言](#)

[译者序](#)

[致谢](#)

[第1部分 Spring的核心](#)

[第1章 Spring之旅](#)

[1.1 简化Java开发](#)

[1.1.1 激发POJO的潜能](#)

[1.1.2 依赖注入](#)

[1.1.3 应用切面](#)

[1.1.4 使用模板消除样板式代码](#)

[1.2 容纳你的Bean](#)

[1.2.1 使用应用上下文](#)

[1.2.2 bean的生命周期](#)

[1.3 俯瞰Spring风景线](#)

[1.3.1 Spring模块](#)

[1.3.2 Spring Portfolio](#)

[1.4 Spring的新功能](#)

[1.4.1 Spring 3.1新特性](#)

[1.4.2 Spring 3.2新特性](#)

[1.4.3 Spring 4.0新特性](#)

[1.5 小结](#)

[第2章 装配Bean](#)

[2.1 Spring配置的可选方案](#)

[2.2 自动化装配bean](#)

[2.2.1 创建可被发现的bean](#)

[2.2.2 为组件扫描的bean命名](#)

[2.2.3 设置组件扫描的基础包](#)

[2.2.4 通过为bean添加注解实现自动装配](#)

[2.2.5 验证自动装配](#)

[2.3 通过Java代码装配bean](#)

[2.3.1 创建配置类](#)

[2.3.2 声明简单的bean](#)

[2.3.3 借助JavaConfig实现注入](#)

[2.4 通过XML装配bean](#)

[2.4.1 创建XML配置规范](#)

[2.4.2 声明一个简单的<bean>](#)

[2.4.3 借助构造器注入初始化bean](#)

[2.4.4 设置属性](#)

[2.5 导入和混合配置](#)

[2.5.1 在JavaConfig中引用XML配置](#)

[2.5.2 在XML配置中引用JavaConfig](#)

[2.6 小结](#)

[第3章 高级装配](#)

[3.1 环境与profile](#)

[3.1.1 配置profile bean](#)

[3.1.2 激活profile](#)

- [3.2 条件化的bean](#)
- [3.3 处理自动装配的歧义性](#)
 - [3.3.1 标示首选的bean](#)
 - [3.3.2 限定自动装配的bean](#)
- [3.4 bean的作用域](#)
 - [3.4.1 使用会话和请求作用域](#)
 - [3.4.2 在XML中声明作用域代理](#)
- [3.5 运行时值注入](#)
 - [3.5.1 注入外部的值](#)
 - [3.5.2 使用Spring表达式语言进行装配](#)
- [3.6 小结](#)
- [第4章 面向切面的Spring](#)
 - [4.1 什么是面向切面编程](#)
 - [4.1.1 定义AOP术语](#)
 - [4.1.2 Spring对AOP的支持](#)
 - [4.2 通过切点来选择连接点](#)
 - [4.2.1 编写切点](#)
 - [4.2.2 在切点中选择bean](#)
 - [4.3 使用注解创建切面](#)
 - [4.3.1 定义切面](#)
 - [4.3.2 创建环绕通知](#)
 - [4.3.3 处理通知中的参数](#)
 - [4.3.4 通过注解引入新功能](#)
 - [4.4 在XML中声明切面](#)
 - [4.4.1 声明前置和后置通知](#)
 - [4.4.2 声明环绕通知](#)
 - [4.4.3 为通知传递参数](#)
 - [4.4.4 通过切面引入新的功能](#)
 - [4.5 注入AspectJ切面](#)
 - [4.6 小结](#)
- [第2部分 Web中的Spring](#)
- [第5章 构建Spring Web应用程序](#)
 - [5.1 Spring MVC起步](#)
 - [5.1.1 跟踪Spring MVC的请求](#)
 - [5.1.2 搭建Spring MVC](#)
 - [5.1.3 Spitttr应用简介](#)
 - [5.2 编写基本的控制器](#)
 - [5.2.1 测试控制器](#)
 - [5.2.2 定义类级别的请求处理](#)
 - [5.2.3 传递模型数据到视图中](#)
 - [5.3 接受请求的输入](#)
 - [5.3.1 处理查询参数](#)
 - [5.3.2 通过路径参数接受输入](#)
 - [5.4 处理表单](#)
 - [5.4.1 编写处理表单的控制器](#)
 - [5.4.2 校验表单](#)
 - [5.5 小结](#)
- [第6章 渲染Web视图](#)
 - [6.1 理解视图解析](#)
 - [6.2 创建JSP视图](#)
 - [6.2.1 配置适用于JSP的视图解析器](#)
 - [6.2.2 使用Spring的JSP库](#)
 - [6.3 使用Apache Tiles视图定义布局](#)
 - [6.3.1 配置Tiles视图解析器](#)
 - [6.4 使用Thymeleaf](#)
 - [6.4.1 配置Thymeleaf视图解析器](#)
 - [6.4.2 定义Thymeleaf模板](#)
 - [6.5 小结](#)
- [第7章 Spring MVC的高级技术](#)
 - [7.1 Spring MVC配置的替代方案](#)
 - [7.1.1 自定义DispatcherServlet配置](#)

- [7.1.2 添加其他的Servlet和Filter](#)
- [7.1.3 在web.xml中声明DispatcherServlet](#)
- [7.2 处理multipart形式的的数据](#)
 - [7.2.1 配置multipart解析器](#)
 - [7.2.2 处理multipart请求](#)
- [7.3 处理异常](#)
 - [7.3.1 将异常映射为HTTP状态码](#)
 - [7.3.2 编写异常处理的方法](#)
- [7.4 为控制器添加通知](#)
- [7.5 跨重定向请求传递数据](#)
 - [7.5.1 通过URL模板进行重定向](#)
 - [7.5.2 使用flash属性](#)
- [7.6 小结](#)
- [第8章 使用Spring Web Flow](#)
 - [8.1 在Spring中配置Web Flow](#)
 - [8.1.1 装配流程执行器](#)
 - [8.1.2 配置流程注册表](#)
 - [8.1.3 处理流程请求](#)
 - [8.2 流程的组件](#)
 - [8.2.1 状态](#)
 - [8.2.2 转移](#)
 - [8.2.3 流程数据](#)
 - [8.3 组合起来：披萨流程](#)
 - [8.3.1 定义基本流程](#)
 - [8.3.2 收集顾客信息](#)
 - [8.3.3 构建订单](#)
 - [8.3.4 支付](#)
 - [8.4 保护Web流程](#)
 - [8.5 小结](#)
- [第9章 保护Web应用](#)
 - [9.1 Spring Security简介](#)
 - [9.1.1 理解Spring Security的模块](#)
 - [9.1.2 过滤Web请求](#)
 - [9.1.3 编写简单的安全性配置](#)
 - [9.2 选择查询用户详细信息的服务](#)
 - [9.2.1 使用基于内存的用户存储](#)
 - [9.2.2 基于数据库表进行认证](#)
 - [9.2.3 基于LDAP进行认证](#)
 - [9.2.4 配置自定义的用户服务](#)
 - [9.3 拦截请求](#)
 - [9.3.1 使用Spring表达式进行安全保护](#)
 - [9.3.2 强制通道的安全性](#)
 - [9.3.3 防止跨站请求伪造](#)
 - [9.4 认证用户](#)
 - [9.4.1 添加自定义的登录页](#)
 - [9.4.2 启用HTTP Basic认证](#)
 - [9.4.3 启用Remember-me功能](#)
 - [9.4.4 退出](#)
 - [9.5 保护视图](#)
 - [9.5.1 使用Spring Security的JSP标签库](#)
 - [9.5.2 使用Thymeleaf的Spring Security方言](#)
 - [9.6 小结](#)
- [第3部分 后端中的Spring](#)
- [第10章 通过Spring和JDBC征服数据库](#)
 - [10.1 Spring的数据访问哲学](#)
 - [10.1.1 了解Spring的数据访问异常体系](#)
 - [10.1.2 数据访问模板化](#)
 - [10.2 配置数据源](#)
 - [10.2.1 使用JNDI数据源](#)
 - [10.2.2 使用数据源连接池](#)
 - [10.2.3 基于JDBC驱动的数据源](#)

- [10.2.4 使用嵌入式的数据源](#)
- [10.2.5 使用profile选择数据源](#)
- [10.3 在Spring中使用JDBC](#)
 - [10.3.1 应对失控的JDBC代码](#)
 - [10.3.2 使用JDBC模板](#)
- [10.4 小结](#)
- [第11章 使用对象-关系映射持久化数据](#)
 - [11.1 在Spring中集成Hibernate](#)
 - [11.1.1 声明Hibernate的SessionFactory](#)
 - [11.1.2 构建不依赖于Spring的Hibernate代码](#)
 - [11.2 Spring与Java持久化API](#)
 - [11.2.1 配置实体管理器工厂](#)
 - [11.2.2 编写基于JPA的Repository](#)
 - [11.3 借助Spring Data实现自动化的JPA Repository](#)
 - [11.3.1 定义查询方法](#)
 - [11.3.2 声明自定义查询](#)
 - [11.3.3 混合自定义的功能](#)
 - [11.4 小结](#)
- [第12章 使用NoSQL数据库](#)
 - [12.1 使用MongoDB持久化文档数据](#)
 - [12.1.1 启用MongoDB](#)
 - [12.1.2 为模型添加注解，实现MongoDB持久化](#)
 - [12.1.3 使用MongoTemplate访问MongoDB](#)
 - [12.1.4 编写MongoDB Repository](#)
 - [12.2 使用Neo4j操作图数据](#)
 - [12.2.1 配置Spring Data Neo4j](#)
 - [12.2.2 使用注解标注图实体](#)
 - [12.2.3 使用Neo4jTemplate](#)
 - [12.2.4 创建自动化的Neo4j Repository](#)
 - [12.3 使用Redis操作key-value数据](#)
 - [12.3.1 连接到Redis](#)
 - [12.3.2 使用RedisTemplate](#)
 - [12.3.3 使用key和value的序列化器](#)
 - [12.4 小结](#)
- [第13章 缓存数据](#)
 - [13.1 启用对缓存的支持](#)
 - [13.1.1 配置缓存管理器](#)
 - [13.2 为方法添加注解以支持缓存](#)
 - [13.2.1 填充缓存](#)
 - [13.2.2 移除缓存条目](#)
 - [13.3 使用XML声明缓存](#)
 - [13.4 小结](#)
- [第14章 保护方法应用](#)
 - [14.1 使用注解保护方法](#)
 - [14.1.1 使用@Secured注解限制方法调用](#)
 - [14.1.2 在Spring Security中使用JSR-250的@RolesAllowed注解](#)
 - [14.2 使用表达式实现方法级别的安全性](#)
 - [14.2.1 表述方法访问规则](#)
 - [14.2.2 过滤方法的输入和输出](#)
 - [14.3 小结](#)
- [第4部分 Spring集成](#)
- [第15章 使用远程服务](#)
 - [15.1 Spring远程调用概览](#)
 - [15.2 使用RMI](#)
 - [15.2.1 导出RMI服务](#)
 - [15.2.2 装配RMI服务](#)
 - [15.3 使用Hessian和Burlap发布远程服务](#)
 - [15.3.1 使用Hessian和Burlap导出bean的功能](#)
 - [15.3.2 访问Hessian/Burlap服务](#)
 - [15.4 使用Spring的HttpInvoker](#)
 - [15.4.1 将bean导出为HTTP服务](#)

- [15.4.2 通过HTTP访问服务](#)
- [15.5 发布和使用Web服务](#)
 - [15.5.1 创建基于Spring的JAX-WS端点](#)
 - [15.5.2 在客户端代理JAX-WS服务](#)
- [15.6 小结](#)
- [第16章 使用Spring MVC创建REST API](#)
 - [16.1 了解REST](#)
 - [16.1.1 REST的基础知识](#)
 - [16.1.2 Spring是如何支持REST的](#)
 - [16.2 创建第一个REST端点](#)
 - [16.2.1 协商资源表述](#)
 - [16.2.2 使用HTTP信息转换器](#)
 - [16.3 提供资源之外的其他内容](#)
 - [16.3.1 发送错误信息到客户端](#)
 - [16.3.2 在响应中设置头部信息](#)
 - [16.4 编写REST客户端](#)
 - [16.4.1 了解RestTemplate的操作](#)
 - [16.4.2 GET资源](#)
 - [16.4.3 检索资源](#)
 - [16.4.4 抽取响应的元数据](#)
 - [16.4.5 PUT资源](#)
 - [16.4.6 DELETE资源](#)
 - [16.4.7 POST资源数据](#)
 - [16.4.8 在POST请求中获取响应对象](#)
 - [16.4.9 在POST请求后获取资源位置](#)
 - [16.4.10 交换资源](#)
 - [16.5 小结](#)
- [第17章 Spring消息](#)
 - [17.1 异步消息简介](#)
 - [17.1.1 发送消息](#)
 - [17.1.2 评估异步消息的优点](#)
 - [17.2 使用JMS发送消息](#)
 - [17.2.1 在Spring中搭建消息代理](#)
 - [17.2.2 使用Spring的JMS模板](#)
 - [17.2.3 创建消息驱动的POJO](#)
 - [17.2.4 使用基于消息的RPC](#)
 - [17.3 使用AMQP实现消息功能](#)
 - [17.3.1 AMQP简介](#)
 - [17.3.2 配置Spring支持AMQP消息](#)
 - [17.3.3 使用RabbitTemplate发送消息](#)
 - [17.3.4 接收AMQP消息](#)
 - [17.4 小结](#)
- [第18章 使用WebSocket和STOMP实现消息功能](#)
 - [18.1 使用Spring的低层级WebSocket API](#)
 - [18.2 应对不支持WebSocket的场景](#)
 - [18.3 使用STOMP消息](#)
 - [18.3.1 启用STOMP消息功能](#)
 - [18.3.2 处理来自客户端的STOMP消息](#)
 - [18.3.3 发送消息到客户端](#)
 - [18.4 为目标用户发送消息](#)
 - [18.4.1 在控制器中处理用户的消息](#)
 - [18.4.2 为指定用户发送消息](#)
 - [18.5 处理消息异常](#)
 - [18.6 小结](#)
- [第19章 使用Spring发送Email](#)
 - [19.1 配置Spring发送邮件](#)
 - [19.1.1 配置邮件发送器](#)
 - [19.1.2 装配和使用邮件发送器](#)
 - [19.2 构建丰富内容的Email消息](#)
 - [19.2.1 添加附件](#)
 - [19.2.2 发送富文本内容的Email](#)

- [19.3 使用模板生成Email](#)
- [19.3.1 使用Velocity构建Email消息](#)
- [19.3.2 使用Thymeleaf构建Email消息](#)
- [19.4 小结](#)
- [第20章 使用JMX管理Spring Bean](#)
- [20.1 将Spring bean导出为MBean](#)
- [20.1.1 通过名称暴露方法](#)
- [20.1.2 使用接口定义MBean的操作和属性](#)
- [20.1.3 使用注解驱动的MBean](#)
- [20.1.4 处理MBean冲突](#)
- [20.2 远程MBean](#)
- [20.2.1 暴露远程MBean](#)
- [20.2.2 访问远程MBean](#)
- [20.2.3 代理MBean](#)
- [20.3 处理通知](#)
- [20.3.1 监听通知](#)
- [20.4 小结](#)
- [第21章 借助Spring Boot简化Spring开发](#)
- [21.1 Spring Boot简介](#)
- [21.1.1 添加Starter依赖](#)
- [21.1.2 自动配置](#)
- [21.1.3 Spring Boot CLI](#)
- [21.1.4 Actuator](#)
- [21.2 使用Spring Boot构建应用](#)
- [21.2.1 处理请求](#)
- [21.2.2 创建视图](#)
- [21.2.3 添加静态内容](#)
- [21.2.4 持久化数据](#)
- [21.2.5 尝试运行](#)
- [21.3 组合使用Groovy与Spring Boot CLI](#)
- [21.3.1 编写Groovy控制器](#)
- [21.3.2 使用Groovy Repository实现数据持久化](#)
- [21.3.3 运行Spring Boot CLI](#)
- [21.4 通过Actuator获取了解应用内部状况](#)
- [21.5 小结](#)
- [看完了](#)

版权信息

书名：Spring实战（第4版）

ISBN：978-7-115-41730-5

本书由人民邮电出版社发行数字版。版权所有，侵权必究。

您购买的人民邮电出版社电子书仅供您个人使用，未经授权，不得以任何方式复制和传播本书内容。

我们愿意相信读者具有这样的良知和觉悟，与我们共同保护知识产权。

如果购买者有侵权行为，我们可能对该用户实施包括但不限于关闭该帐号等维权措施，并可能追究法律责任。

• 著 [美] Craig Walls

译 张卫滨

责任编辑 陈冀康

• 人民邮电出版社出版发行 北京市丰台区成寿寺路11号

邮编 100164 电子邮件 315@ptpress.com.cn

网址 <http://www.ptpress.com.cn>

• 读者服务热线：(010)81055410

反盗版热线：(010)81055315

作者简介

Craig Walls是Pivotal的高级工程师，是Spring Social和Spring Sync的项目领导者，同时也是Manning出版社《Spring In Action》的作者，目前这本书已经更新到了第四版。他非常热心于Spring框架的推广，经常在当地的用户组和会议上演讲并在博客上撰写Spring相关的内容。在不琢磨代码的时候，Craig Walls会尽可能多地陪伴他的妻子、两个女儿、两只小鸟以及两只小狗。

本书特色

全球有超过100 000的开发者使用本书来学习Spring

中文版累计销售超10万册，畅销经典Spring 技术图书，针对Spring 4 全新升级

作者Craig Walls，SpringSource的软件开发人员，也是一位畅销书作者。

第3版译者继续翻译新版，品质保障！

版权声明

Original English language edition, entitled Spring in Action, 4th Edition by Craig Walls Bibeault published by Manning Publications Co., 209 Bruce Park Avenue, Greenwich, CT 06830. Copyright ©2015 by Manning Publications Co.

Simplified Chinese-language edition copyright ©2016 by Posts & Telecom Press. All rights reserved.

本书中文简体字版由Manning Publications Co.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本书内容。

版权所有，侵权必究。

内容提要

本书是经典的、畅销的Spring学习和实践指南。

第4版针对Spring 4进行了全面更新。全书分为4部分。第1部分介绍Spring框架的核心知识。第2部分在此基础上介绍了如何使用Spring构建Web应用程序。第3部分告别前端，介绍了如何在应用程序的后端使用Spring。第4部分描述了如何使用Spring与其他的应用和服务进行集成。

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。

欢迎访问：电子书学习和下载网站 (<https://www.shgis.cn>)

文档名称：Spring实战（第4版） - [美] Craig Walls 沃尔斯.epub

请登录 <https://shgis.cn/post/1832.html> 下载完整文档。

手机端请扫码查看：

