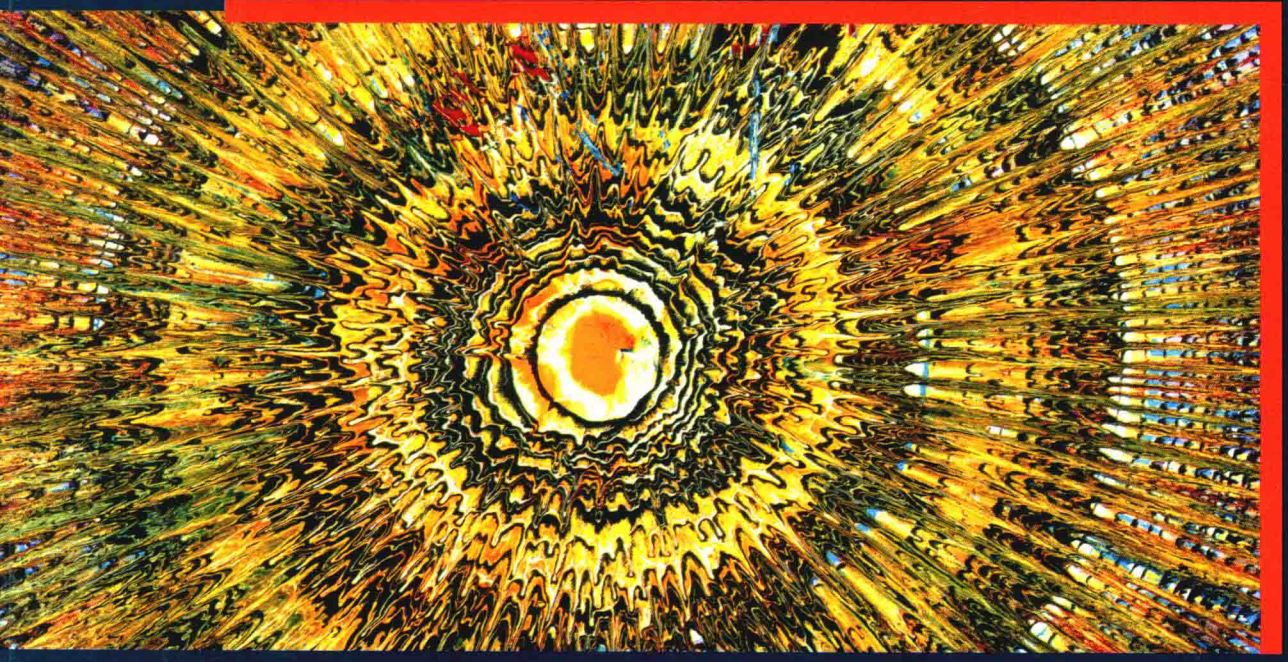
华章程序员书库

HZ BOOKS
华章IT

Spring Boot 开发实战

Introduction to Spring Boot: A Hands-On Approach

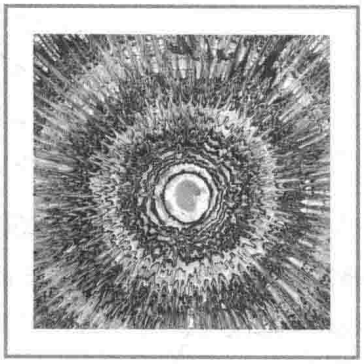


基于Spring Boot 2.0，零基础学习Web开发
从企业级应用的场景出发，手把手带领入门，包含大量案例代码

陈光剑 编著

机械工业出版社
China Machine Press

华章程序员书库



Spring Boot 开发实战

图书馆

陈光剑 编著

机械工业出版社
China Machine Press

图书在版编目（CIP）数据

Spring Boot 开发实战 / 陈光剑编著 . —北京：机械工业出版社，2018.7
（华章程序员书库）

ISBN 978-7-111-60333-7

I. S… II. 陈… III. JAVA 语言 – 程序设计 IV. TP312.8

中国版本图书馆 CIP 数据核字（2018）第 145768 号

Spring Boot 开发实战

出版发行：机械工业出版社（北京市西城区百万庄大街 22 号 邮政编码：100037）
责任编辑：吴 怡 责任校对：李秋荣
印 刷：北京文昌阁彩色印刷有限责任公司 版 次：2018 年 8 月第 1 版第 1 次印刷
开 本：186mm × 240mm 1/16 印 张：23.5
书 号：ISBN 978-7-111-60333-7 定 价：89.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换
客服热线：（010）88379426 88361066 投稿热线：（010）88379604
购书热线：（010）68326294 88379649 68995259 读者信箱：hzit@hzbook.com

版权所有 • 侵权必究
封底无防伪标均为盗版
本书法律顾问：北京大成律师事务所 韩光 / 邹晓东

Preface 前　言

Spring Boot 是由 Pivotal 团队提供的全新框架，其设计目的是简化新 Spring 应用的初始搭建以及开发过程。在 Java 开发领域中，有很多著名框架都是 Pivotal 团队的产品，如：Spring 框架及其衍生框架、缓存 Redis、消息队列框架 RabbitMQ、Greenplum 数据库等。还有 Tomcat、Apache Http Server、Groovy 里的一些顶级开发者、DevOps 理论的提出者都属于 Pivotal 团队。Spring 团队在现有 Spring 框架的基础上，开发了一个新框架 :Spring Boot，用来简化配置和部署 Spring 应用程序的过程，去除了那些烦琐的开发步骤和样板代码及其配置，使得基于 Spring 框架的 Java 企业级应用开发“极简化”。相比于传统的 Spring/Spring MVC 框架的企业级应用开发 (Spring 的各种配置太复杂了，我们之前是用“生命”在搞这些配置)，Spring Boot 用简单的注解和 application.properties 配置文件，避免了烦琐而且容易出错的 XML 配置文件，极大地简化了基于 Spring 框架的企业级应用开发的配置。

Kotlin 是由 JetBrains 团队开发的多平台、静态类型、强工程实用性的编程语言，Kotlin 100% 兼容 Java，比 Java 更强大、更安全、更简洁、更优雅。Kotlin 是 Google 公司的 Android 官方支持的开发语言。Spring 官方也正式支持 Kotlin 语言，Spring Boot 2.0 版本中为 Kotlin 提供了一流的支持。其实，在 Spring Boot 2.0 和 Spring 5.0 框架源代码中，已经可以看到 Kotlin 代码。

本书可以说是我对使用 Spring Boot + Kotlin 进行服务端开发的实战和思考过程的粗浅总结。通过本书的写作，加深了我对 Spring Boot 框架和 Kotlin 编程语言的理解，我深刻体会到了学无止境的含义。写书的过程也是我系统学习与思考的过程，如果本书能够对你有所帮助，将不胜欣慰。

如何阅读本书

本书系统介绍了使用 Spring Boot 2.0 框架，并基于 Gradle + Kotlin 来开发企业级应用。希望通过简练的表述，系统全面地介绍如何使用 Spring Boot 2.0 框架开发项目，每章的关联度不

大，读者可根据自己的需求阅读本书。

全书共分三大部分：

- ❑ 第Ⅰ部分　Spring Boot 框架基础（第 1 ～ 3 章）
- ❑ 第Ⅱ部分　Spring Boot 项目综合实战（第 4 ～ 17 章）
- ❑ 第Ⅲ部分　Spring Boot 系统监控、测试与运维（第 18 ～ 20 章）

建议初学者最好按照章节顺序来阅读本书。如果想直接使用 Spring Boot 框架进行项目的实战，可以直接进入第Ⅱ部分，如果对 Spring Boot 应用的监控、测试与运维感兴趣，那么可以从第Ⅲ部分直接开始阅读。

本书共 20 章，各个章节内容简介如下。

第 1 章：简单介绍了 Spring Boot 框架的历史、组成、特性等。

第 2 章：使用 Spring Boot 2.0 快速实现一个基于 Kotlin 和 Gradle 的 HelloWorld 应用。

第 3 章：介绍 Spring Boot 是怎样通过自动配置实现“极简化配置”的应用开发。

第 4 章：介绍如何使用 Spring Boot 集成 MyBatis 来进行数据库层开发。

第 5 章：介绍如何使用 Spring Boot 集成 Spring Data JPA 来进行数据库层开发。

第 6 章：介绍如何开发一个 Gradle 插件，以及如何简化开发过程中样板代码的编写。

第 7 章：介绍 Kotlin 编程语言，以及如何集成 Spring Boot 和 Spring MVC 进行服务端开发。

第 8 章：介绍在 Spring Boot 项目中怎样自定义 Web MVC 配置。

第 9 章：介绍基于 Spring Boot + Spring MVC，使用 AOP + Filter 如何实现一个简单的用户登录鉴权与权限控制系统。

第 10 章：介绍如何使用 Spring Boot 集成 Spring Security 开发一个自动化测试平台。

第 11 章：介绍 Spring Boot 集成 React.js 开发前后端分离项目的实战案例。

第 12 章：介绍如何开发任务调度、邮件服务等系统功能。

第 13 章：介绍如何用 Spring Boot 集成 WebFlux 开发响应式 Web 应用。

第 14 章：介绍在 Spring Boot 项目开发中怎样使用 Spring Cache 实现数据的缓存。

第 15 章：介绍如何使用 Spring Session 集成 Redis 实现 Session 共享，从而实现水平扩展。

第 16 章：介绍如何使用 Netflix Zuul 实现一个微服务 API Gateway 来完成简单代理转发和过滤器功能。

第 17 章：详细介绍 Spring Boot 应用的日志配置与使用，主要介绍 Logback 日志框架。

第 18 章：介绍如何使用 Spring Boot Actuator 和 Spring Boot Admin 实现监控与管理。

第 19 章：介绍 Spring Boot 应用的测试，以及如何在实际项目中进行分层测试。

第 20 章：介绍如何使用 Docker 来构建部署运行 Spring Boot 应用。

试读结束：需要全本请在线购买：www.ertongbook.com

谁适合阅读本书

本书适合于所有 Java、Kotlin 程序员，以及任何对编程感兴趣的朋友。如果你目前还不是程序员，但想进入企业级应用开发的编程世界，那么你也可以尝试从本书开始学习。

虽然书中的部分内容需要一定的 Java 和 Kotlin 编程基础，还需要了解 Spring 框架，但是如果你想快速开始企业级应用开发，不妨从这里开始——Spring Boot 2.0 + Kotlin ，这种方式的极简特性定能激发你对编程的兴趣。

代码下载

每章末尾基本上都附了该章示例工程源代码地址。这些源码都在 https://github.com/Easy-SpringBoot 。可以根据需要，自由克隆下载学习。

致谢

在本书的写作出版过程中，得到了很多人的帮助和陪伴。首先要感谢的是我的妻子和两个可爱的孩子。正是有了你们的陪伴，我的生活才更加有意义。我始终感谢我的父母，虽然你们可能不知道我写的东西是什么，但是因为有了你们的辛勤养育，我才能长成今天的我。我要衷心地感谢吴怡编辑。在本书的写作修改过程中，她耐心细致地对稿件进行了详尽、细致的审阅和批注，还提出了很多宝贵的修改建议。感谢本书出版过程中所有付出辛勤劳动的工作人员。我还要感谢在我的工作学习生活中认识的，所有朋友和同事们，能够认识你们并跟你们一起学习共事，是我的荣幸。

请联系我

虽然在本书写作与修改的过程中，我竭尽全力追求简单正确、清晰流畅地表达内容，但是限于自身水平和有限的时间，也许仍有错误与疏漏之处，还望各位读者不吝指正。

关于本书的任何问题、意见或者建议都可以通过邮件 universsky@163.com 与我交流。

快乐生活，快乐学习，快乐分享，快乐实践出真知。

最后，祝大家阅读愉快！

陈光剑

2018 年 4 月于杭州

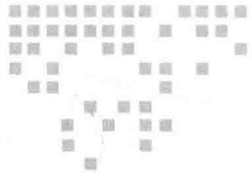
目　录 *Contents*

试读结束：需要全本请在线购买： www.ertongbook.com

第 I 部分 *Part 1*

Spring Boot 框架基础

- 第 1 章　Spring Boot 简介
- 第 2 章　快速开始 HelloWorld
- 第 3 章　深入理解 Spring Boot 自动配置



Chapter 1 第 1 章

Spring Boot 简介

认识一个事物最好的方式就是首先去了解它的历史。

Spring 框架是由 Rod Johnson 在 2001 年开始开发的一个开源框架，主要为了解决企业级应用程序开发的复杂性。Spring 提倡“零”侵入设计原则，颠覆了传统的编程模式。Spring 引入控制反转（Inversion of Control，IoC）的核心编程思想，控制反转还有一个名字叫作依赖注入（Dependency Injection，DI），就是由容器来管理协同 Bean 之间的关系，而非传统实现中，由程序代码直接操控。同时，Spring 还把面向切面编程（AOP）集成进来，使得 AOP 的编程范式发扬光大。

Spring 从 IoC 容器发展而来，通过不断集成 AOP、MVC、OR/Mapping 以及几乎你能想到的各项服务而提供完善的企业应用框架。目前大多数 J2EE 项目都已经采用 Spring 框架。

随着 Spring 功能的不断丰富，版本的不断迭代发展，Spring 框架渐渐暴露出了一些问题和弊端。例如太多样板化的配置、烦琐复杂的使用过程等，我们不仅需要维护程序代码，还需要额外去维护相关的配置文件。Spring 项目的配置越来越复杂，让人难以承受。大量的 XML 配置以及复杂的依赖管理使得人们不得不去解决这个问题——Spring Boot 由此应运而生。

在本章中，我们先来简单了解一下 Spring Boot 框架的历史、组成、特性等。

1.1 从 Spring 到 Spring Boot

本节将介绍 Spring Boot 的产生背景。我们先来回顾一下 Spring 框架的前世今生。

1.1.1　从 EJB 到 Spring

EJB（Enterprise Java Bean）最初的设计思想是为分布式应用服务的。分布式是针对大型应用构造的跨平台的协作计算，EJB 最初的目的就是为这种计算服务的。使用 EJB 技术的系统整体架构如图 1-1 所示。

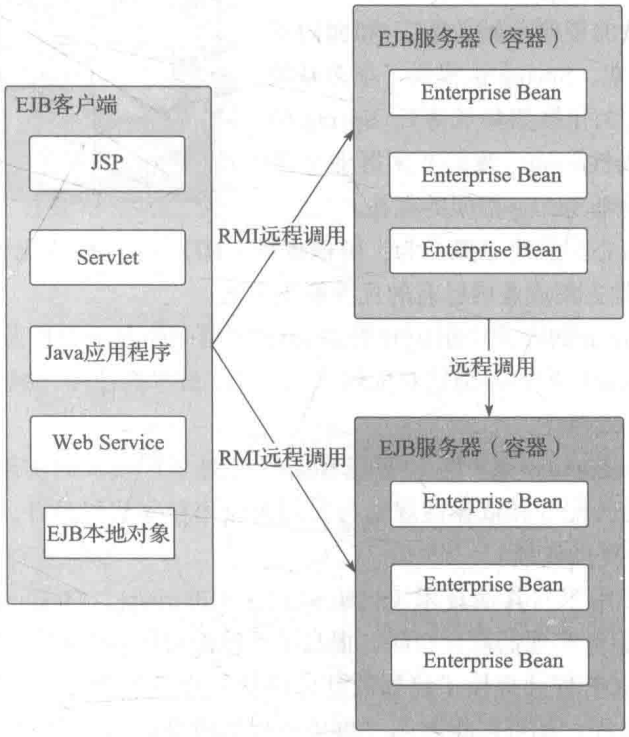


图 1-1　使用 EJB 技术的系统架构图

EJB 的基础是 RMI（Remote Method Invocation，远程方法调用），RMI 利用 Java 对象序列化的机制实现分布式计算，实现远程类对象的实例化以及调用。通过 RMI，J2EE 将 EJB 组件创建为远程对象。RMI 将各种任务与功能的类放到不同的服务器上，然后通过各个服务器间建立的调用规则实现分布式的运算。通过 RMI 的通信（底层仍然是 Socket），连接不同功能模块的服务器，以实现一个完整的功能。

EJB 规范定义了 EJB 组件在何时如何与它们的容器进行交互作用，容器负责提供公用的服务，例如目录服务、事务管理、安全性、资源缓冲池以及容错性。但这里值得注意的是，EJB 并不是实现 J2EE 的唯一途径。但是软件发展到目前为止，大多数应用不需要采用这么重的解决方案，因此用 EJB 显得太臃肿了。

更多关于 J2EE 的内容，可以参考：https://github.com/javaee。

对于中小型的应用项目而言，基本不采用分布式的解决方案，那么为什么要采取一个为分布式设计的方案来解决非分布式的问题呢？Spring 就是为了解决这个问题而诞生的。

Spring 的目的是为了解决企业应用开发的复杂性，它的主要功能是使用基本的 Java Bean 代替 EJB，并提供了更多的企业应用功能。Spring 使得已存在的技术更加易用。简单来说，Spring 是一个轻量级的控制反转（IoC）和面向切面（AOP）的容器框架。Spring 也提供了很多基础功能（事务管理、持久化框架集成等）。Spring 的设计原则是“非侵入性”的，我们在实际业务逻辑代码中几乎感觉不到 Spring 框架的存在。

Spring 框架的核心功能简单概括为：解耦依赖（DI）、系统模块化（AOP）。Spring“不重复发明轮子”，而是去集成业内已有的优秀解决方案。

Spring 容器以 Bean 的方式来组织和管理 Java 应用中的各个组件及其组件之间的关系。基于 Java Beans 的配置管理，特别是对依赖注入（DI）技术的使用，减少了各组件间对业务逻辑具体实现的相互依赖性。

Spring 使用 BeanFactory 来产生和管理 Bean，它是工厂模式的实现。BeanFactory 使用控制反转模式将应用的配置和依赖性规范与实际的应用程序代码分开。BeanFactory 使用依赖注入的方式给组件提供依赖。

Spring 框架主要用于与其他技术（例如 Struts，Hibernate，MyBatis 等）进行整合，将应用程序中的 Bean 组件实现低耦合关联，提高了系统的可扩展性和维护性。

Spring 集成的 AOP 框架提供了诸如数据库声明式事务等服务。通过使用 Spring AOP，我们无须依赖 EJB 组件，就可以将声明式事务管理集成到应用程序中。AOP 的目的是提高系统的模块化程度。

当然，作为一个完整的 J2EE 框架，Spring 生态中也给出了完整的分布式系统架构的解决方案，那就是 Spring Boot + Spring Cloud，这个解决方案中包含了服务发现（Service Discovery）、断路器（Circuit Breaker）、OAuth2（实现 SSO、登录 token 的管理）、服务配置（Configuration Server）、消费者驱动契约（Consumer-Driven Contracts）、API Gateway 等。

Spring 的微服务系统架构如图 1-2 所示。

1.1.2 Spring 框架发展简史

Spring 框架首次在 2003 年 6 月的 Apache2.0 使用许可中发布。第一个具有里程碑意义的版本是 2004 年 3 月发布的 1.0。

下面是 Spring 框架的发展简史：

- 2003 年，Spring0.9 发布。2003 年 11 月，Ben Alex 将 Acegi Security 的代码贡献给 Rod 和 Juergen，2006 年 5 月发布 Acegi Security。

- 2006 年 6 月发布 Spring Webflow 1.0。2006 年 8 月发布 Spring LDAP。2006 年 10 月发布 Spring 2.0。
- 2007 年 5 月发布 Spring Batch。2007 年 11 月发布 Spring 2.5。Spring 2.5 是 Spring 2.1 各个里程碑版本的终结。
- 2011 年 6 月发布 Spring Data JPA 1.0。2011 年 12 月发布 Spring 3.1
- 2014 年 4 月发布 Spring Boot 1.0。2014 年 12 月发布 Spring 4.1.3
- 2015 年 7 月发布 Spring 4.2
- 2016 年 6 月发布 Spring 4.3
- 2017 年 9 月发布 Spring 5.0。2017 年 11 月发布 Spring Boot v2. 0.0.M7
- 2018 年 3 月 1 日发布 Spring Boot v2.0.0.Release；2018 年 4 月 5 日发布 Spring Boot 2.0.1.Release 版本，是目前最新版本。

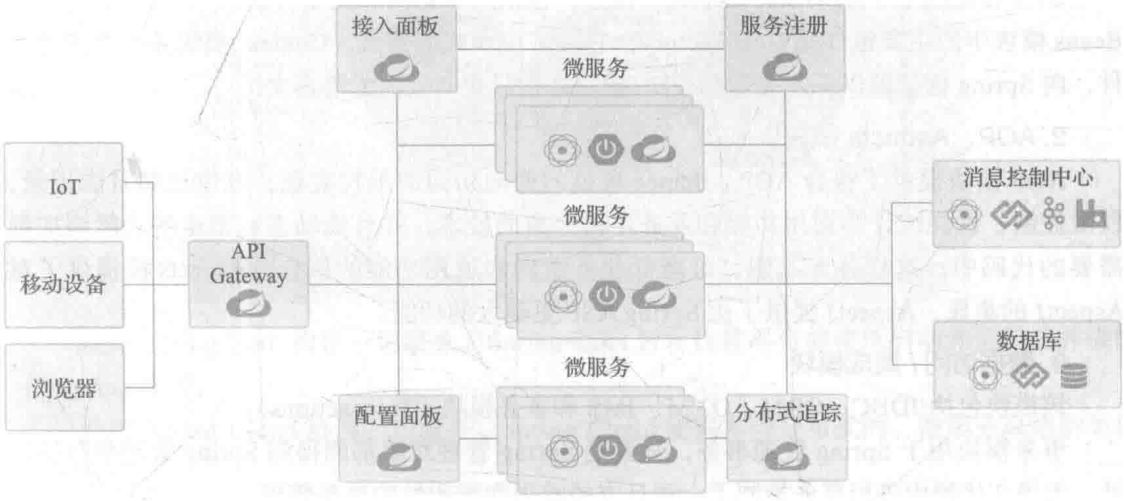


图 1-2　Spring 的微服务系统架构图

提示　详细的发布日志参考 https://github.com/spring-projects/spring-boot/releases。

1.1.3　Spring 框架的核心模块

Spring 框架如图 1-3 所示。组成 Spring 框架的每个模块（或组件）都可以单独存在，或者与其他一个或多个模块联合实现。下面我们分别介绍。

1. 核心容器模块

核心容器提供 Spring 框架的基本功能，包括 Core、Beans、Context、EL 模块。

试读结束：需要全本请在线购买：　www.ertongbook.com