来自一线开发者的实战经验总结
赠送作者讲解的部分配套视频课程

深入浅出 Spring Boot 2.x

杨开振 著

从**理论**到**实践**，全面介绍 **Spring Boot** 的原理和应用
以**高并发场景**作为实践案例，**循序渐进**阐述 **Spring Boot** 实用技巧
结合主流**持久层框架 MyBatis**，讲述企业级 **Spring Boot** 开发要点

中国工信出版集团
人民邮电出版社
POSTS & TELECOM PRESS

杨开振

精通 Java 互联网技术开发和实践，拥有十余年一线企业开发经验，著有业内畅销书《深入浅出 MyBatis 技术原理与实战》和《Java EE 互联网轻量级框架整合开发——SSM 框架（Spring MVC+Spring+MyBatis）和 Redis 实现》，目前成为自由职业人，在淘宝教育平台上从事 Java EE 互联网相关的在线培训工作。

本书免费视频

作者在线课程

深入浅出 Spring Boot 2.x

杨开振 著

人 民 邮 电 出 版 社
北 京

图书在版编目（CIP）数据

深入浅出Spring Boot 2.x / 杨开振著. -- 北京 : 人民邮电出版社, 2018.8
ISBN 978-7-115-48638-7

Ⅰ. ①深… Ⅱ. ①杨… Ⅲ. ①JAVA语言－程序设计 Ⅳ. ①TP312.8

中国版本图书馆CIP数据核字(2018)第124270号

内 容 提 要

Spring 框架是 Java EE 开发的强有力的工具和事实标准，而 Spring Boot 采用“约定优于配置”的原则简化了 Spring 的开发，从而成为业界最流行的微服务开发框架，已经被越来越多的企业采用。2018 年 3 月 Spring Boot 的版本正式从 1.x 升级到了 2.x，为了适应新潮流，本书将对 Spring Boot 2.x 技术进行深入讲解。

本书从一个最简单的工程开始讲解 Spring Boot 企业级开发，其内容包含全注解下的 Spring IoC 和 AOP、数据库编程（JDBC、JPA 和 MyBatis）、数据库事务、NoSQL（Redis 和 MongoDB）技术、Spring MVC、Spring 5 新一代响应式框架 WebFlux、互联网抢购业务、部署与监控、REST 风格和 Spring Cloud 分布式开发等。

本书内容紧扣互联网企业的实际要求，从全注解下 Spring 知识讲到 Spring Boot 的企业级开发，对于 Java 开发人员，尤其是初学 Spring Boot 的人员和需要从传统 Spring 转向 Spring Boot 开发的技术人员，具有很高的参考价值。

◆ 著　　杨开振
责任编辑　杨海玲
责任印制　焦志炜

◆ 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
北京鑫正大印刷有限公司印刷

◆ 开本：800×1000　1/16
印张：27.75
字数：686 千字　　2018 年 8 月第 1 版
印数：1－4 000 册　　2018 年 8 月北京第 1 次印刷

定价：99.00 元

读者服务热线：(010)81055410　印装质量热线：(010)81055316
反盗版热线：(010)81055315
广告经营许可证：京东工商广登字 20170147 号

前　言

本书的缘起

当前互联网后端开发中 Java EE 占据了主导地位。对于 Java EE 开发，首选框架和事实标准是 Spring 框架。在传统的 Spring 开发中需要使用大量的 XML 配置才能使 Spring 框架运行起来，这备受许多开发者诟病。随着 Spring 4.x 的发布，Spring 已经完全可以脱离 XML，只使用注解就可以运行项目。近两三年里，互联网世界掀起了“微服务”热潮。“微服务”将一个大的系统拆分为多个子系统，然后通过 REST 风格的请求将它们集成起来，进一步简化了分布式系统的开发。为了进一步简化 Spring 的开发，2014 年 Spring Boot 诞生了，它是一个由 Pivotal 团队提供的全新框架，其设计目的是简化 Spring 应用的搭建以及开发过程，并迎合时下流行的微服务思维，越来越多的企业选择了 Spring Boot。随着 2017 年 9 月 Spring 5.x 的推出，2018 年 Spring Boot 也推出了 2.x 版本，进入 2.x 版本时代。

基于这样的趋势，在我和朋友合作创作完成《Java EE 互联网轻量级框架整合开发：SSM 框架（Spring MVC+Spring+MyBatis）和 Redis 实现》后，收到了许多的读者、前同事和业内朋友的建议，他们希望我创作一本关于 Spring Boot 的书，来给需要学习 Spring Boot 的从业人员提供参考，这就是创作本书的缘起。Spring Boot 采用了“约定优于配置”的规则，大部分情况下依赖它提供的 starter 后，就可以使用默认的约定，加上属性文件，做大量的自定义配置，使开发更为简单；对于部署，Spring Boot 提供了内嵌服务器，和 Maven（或 Grandle）打包，进一步降低了企业部署的难度；对于测试，它提供了快速测试的环境，进一步提高了开发效率，因此它渐渐成为中小型企业甚至是一些大型企业开发的主流选择。加之在互联网世界中，分布式已经是一种必然的趋势，而分布式的治理和组件研发成本并非一般公司所能承担，为此 Spring 社区还在 Spring Boot 的基础上提供了 Spring Cloud 分布式开发组件，从而进一步简化了企业级分布式开发，这让 Spring Boot 和 Spring Cloud 都站到了互联网后端开发的主流方向上，越来越受到企业的青睐。

本书的安排

Spring Boot 不是代替 Spring，而是使 Spring 项目可以更加快速地开发、部署和测试。它采用了“约定优于配置”的理念，在其内部提供了大量的 starter，而这些 starter 又提供了许多自动配置类，让开发者可以奉行“拿来主义”，开箱即用。虽然这样能够快速地开发、部署和测试，但是也会带来很大的问题，那就是，如果不懂 Spring 的原理，一旦出现开发的问题，开发者就很容易陷入困境，难以找到问题的根源，造成开发者的困扰。所以要学习 Spring Boot 就必须掌握 Spring 的基础知识。基于这种情况，本书会结合 Spring 的原理讨论 Spring Boot 的应用。

为了更好地讨论 Spring Boot 的相关知识，本书内容安排如下。

- 第 1 章和第 2 章先讲 Spring Boot 和传统 Spring 开发的区别，以及如何搭建 Spring Boot 开发环境。
- 第 3 章和第 4 章讨论在全注解下的 Spring 基础 IoC 和 AOP，让初学者可以无缝对接 Spring Boot 的全注解开发方式。
- 第 5 章和第 6 章讲述数据库的开发、基于 SSM 框架（Spring MVC+Spring+MyBatis）的流行以及数据库事务的重要性，除了讨论传统的 JDBC 和 JPA 开发，还会重点讨论和 MyBatis 框架的整合，以及 Spring 数据库事务的编程。
- 第 7 章和第 8 章主要讲互联网中广泛使用的两种 NoSQL 数据库（即 Redis 和 MongoDB），使用它们可以极大地提高系统的性能。
- 第 9 章和第 10 章讲解在 Spring Boot 和全注解下的 Spring MVC 开发，从 Spring MVC 的基础讲到实际的开发和应用，让读者能够掌握各种 Spring Web 后端的开发技巧。
- 第 11 章讲构建 REST 风格的网站。因为当前各个微服务是以 REST 风格请求相互融合的，所以时下它已经成为一种广泛使用的风格。
- 第 12 章讲 Spring Security，通过它可以保护我们的站点，使其远离各种各样的攻击，保证网站安全，这是互联网应用必须做到的。
- 第 13 章讲一些 Spring 常用的技术，如异步线程、定时器、消息机制和 WebSocket 等，以满足企业的其他开发需要。
- 第 14 章讲解 Spring 5 推出的新的非阻塞框架 WebFlux，介绍非阻塞编程的技巧，通过它可以构建非阻塞的网站。
- 第 15 章讲 SSM 整合，并通过抢购场景讲述互联网中的高并发与锁的应用。
- 第 16 章讲 Spring Boot 的打包、部署、测试和监控。
- 第 17 章讲基于 Spring Cloud 的分布式开发入门知识，使用它可以构建企业级分布式系统。

上述内容可以让读者对 Spring Boot 有深入的了解，并且通过进一步学习掌握企业级应用的开发技巧。

阅读本书的要求和目标读者

阅读本书前，读者需要具备 Java 编程语言基础、Java EE（Servlet 和 JSP）基础、前端（HTML、JavaScript 和 JQuery）基础和数据库（MySQL、Redis 和 MongoDB）基础。当然读者也可以根据自己感兴趣的技术选择部分章节来学习。

本书使用全注解讲解 Spring 基础技术（IoC 和 AOP），因此适合从事或者即将使用 Spring Boot 开发的人员阅读和学习，也适合基于传统 Spring 需要转向 Spring Boot 开发方式的开发者阅读，当然也适合作为大中专院校作为教材，帮助在校师生贴近企业级 Java EE 开发。读者通过本书的学习可以有效地提高自身的技术能力，并能将这些技术应用于实际学习和工作当中，当然读者也可以把本书当作工作手册来查阅。

本书使用的 Spring Boot 版本

Spring Boot 作为一个被市场高度关注的微服务开发框架，版本迭代十分频繁，这给我创作本书带来了极大的挑战。本书出版前还有一个有趣的插曲，在本书初创时 Spring Boot 的最新正式版是 1.5.4，到我最初定稿时更新到了 1.5.9，都是基于 Spring Boot 的 1.x 版本。在 2018 年 3 月初，在书稿进入复审环节之前，Spring Boot 发生了重大的版本更替，正式更新到了 2.x 的正式（GA）版本。为了与时俱进，保证本书更有参考价值，我决定将本书采用的 Spring Boot 版本从最初定稿的 1.5.9 更新到 2.0.0。因此，本书采用版本 2.0.0.RELEASE 进行讲解。Spring Boot 2.x 和 Spring Boot 1.x 在使用上有很多地方存在很多不同，并且只能支持 JDK 8 或者以上版本，这些是读者在阅读本书和实践中需要注意的。

致谢

本书得以顺利出版要感谢人民邮电出版社的编辑们，尤其是杨海玲编辑，她以编辑的专业精神时常鞭策我，并给予我很多帮助和支持，没有编辑的付出就不会有本书的出版。

感谢我的家人对我的支持和理解，当在电脑桌前写书、写代码和录视频课程时，牺牲了很多本该好好陪伴他们的时光。

感谢鼓励我编写本书的读者和朋友们，没有他们的鼓励，就不会有本书的缘起。

纠错、源码和课程

Spring 和 Spring Boot 技术的使用和涉及面十分广泛，一些技术博大精深，版本更替也十分频繁，加上本人能力有限，所以书中错误之处在所难免。但是，正如没有完美的技术一样，也没有完美的书籍。尊敬的读者，如果你对本书有任何意见或建议，欢迎给我发送邮件（ykzhen2013@163.com），或者在我的博客（https://blog.csdn.net/ykzhen2015）上留言，以便于及时修订本书的错漏。

为了更好地帮助读者学习和理解本书内容，本书还提供免费的基础入门视频课程（扫封面上的二维码在线观看）和源代码下载，相关信息会发布到异步社区（https://www.epubit.com）和作者博客上，欢迎读者关注。

杨开振

2018 年 5 月

资源与支持

本书由异步社区出品，社区（https://www.epubit.com/）为您提供相关资源和后续服务。

配套资源

本书提供如下资源：

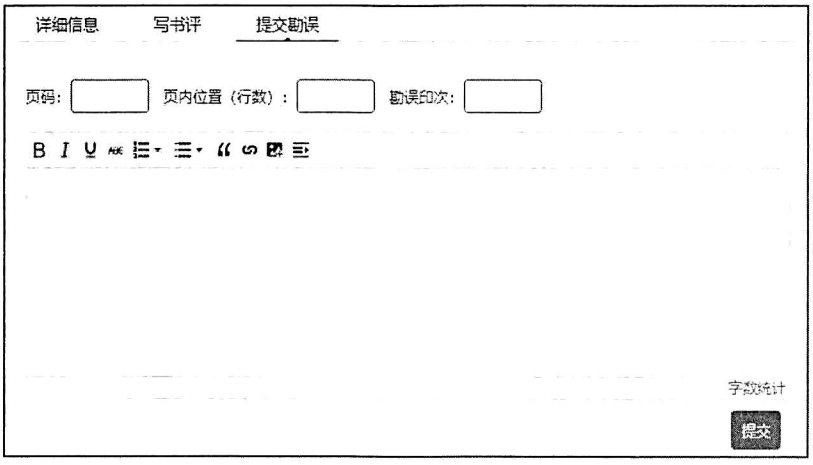
- 本书源代码；
- 书中彩图文件。

要获得以上配套资源，请在异步社区本书页面中点击 配套资源，跳转到下载界面，按提示进行操作即可。注意：为保证购书读者的权益，该操作会给出相关提示，要求输入提取码进行验证。

提交勘误

作者和编辑尽最大努力来确保书中内容的准确性，但难免会存在疏漏。欢迎您将发现的问题反馈给我们，帮助我们提升图书的质量。

当您发现错误时，请登录异步社区，按书名搜索，进入本书页面，点击“提交勘误”，输入勘误信息，点击“提交”按钮即可。本书的作者和编辑会对您提交的勘误进行审核，确认并接受后，您将获赠异步社区的 100 积分。积分可用于在异步社区兑换优惠券、样书或奖品。



扫码关注本书

扫描下方二维码，您将会在异步社区微信服务号中看到本书信息及相关的服务提示。

与我们联系

我们的联系邮箱是 contact@epubit.com.cn。

如果您对本书有任何疑问或建议，请您发邮件给我们，并请在邮件标题中注明本书书名，以便我们更高效地做出反馈。

如果您有兴趣出版图书、录制教学视频，或者参与图书翻译、技术审校等工作，可以发邮件给我们；有意出版图书的作者也可以到异步社区在线提交投稿（直接访问 www.epubit.com/selfpublish/submission 即可）。

如果您是学校、培训机构或企业，想批量购买本书或异步社区出版的其他图书，也可以发邮件给我们。

如果您在网上发现有针对异步社区出品图书的各种形式的盗版行为，包括对图书全部或部分内容的非授权传播，请您将怀疑有侵权行为的链接发邮件给我们。您的这一举动是对作者权益的保护，也是我们持续为您提供有价值的内容的动力之源。

关于异步社区和异步图书

“异步社区”是人民邮电出版社旗下 IT 专业图书社区，致力于出版精品 IT 技术图书和相关学习产品，为作译者提供优质出版服务。异步社区创办于 2015 年 8 月，提供大量精品 IT 技术图书和电子书，以及高品质技术文章和视频课程。更多详情请访问异步社区官网 https://www.epubit.com。

“异步图书”是由异步社区编辑团队策划出版的精品 IT 专业图书的品牌，依托于人民邮电出版社近 30 年的计算机图书出版积累和专业编辑团队，相关图书在封面上印有异步图书的 LOGO。异步图书的出版领域包括软件开发、大数据、AI、测试、前端、网络技术等。

异步社区

微信服务号

目 录

第 1 章

Spring Boot 来临

当今许多互联网企业采用 Java EE 的技术开发自己的后端服务器，其原因在于 Java 语言的简单、安全、支持多线程、高性能和多年 Java EE 的技术积累，能够快速、安全、高性能地构建互联网项目。而如果你身处于 Java EE 的领域，那么你一定听过 Spring 的大名，它是当今 Java EE 开发的事实标准，也是绝大部分企业构建 Java EE 应用的基础。开启 Spring Boot 讲解之前，让我们先回顾 Spring Framework 的历史。

1.1 Spring 的历史

在 Spring 框架没有开发出来时，Java EE 是以 Sun 公司（已经被 Oracle 公司收购，不复存在，但为了纪念其对 Java 发展进程的巨大影响力，全书还是保留其名称，以表致敬之意）所制定的 EJB（Enterprise Java Bean）作为标准的。在“遥远”的 EJB 年代，开发一个 EJB 需要大量的接口和配置文件，直至 EJB 2.0 的年代，开发一个 EJB 还需要配置两个文件，其结果就是配置的工作量比开发的工作量还要大。其次 EJB 是运行在 EJB 容器中的，而 Sun 公司定义的 JSP 和 Servlet 却是运行在 Web 容器中的，于是你可以想象得到，你需要使用 Web 容器去调用 EJB 容器的服务。这就意味着存在以下的弊端：需要增加调用的配置文件才能让 Web 容器调用 EJB 容器；与此同时需要开发两个容器，非常多的配置内容和烦琐的规范导致开发效率十分低下，这非常让当时的开发者诟病；对于 Web 容器调用 EJB 容器的服务这种模式，注定了需要通过网络传递，造成性能不佳；对于测试人员还需要了解许多 EJB 烦琐的细节，才能进行配置和测试，这样测试也难以进行。

就在大家诟病 EJB 的时候，2002 年澳大利亚工程师 Rod Johnson（论学历他应该是音乐家，因为他是音乐博士）在其著名的著作 *Expert One-on-One J2EE Design and Development* 中提出了 Spring 的概念。按书中的描述，Spring 是如下的框架。

We believe that:

J2EE should be easier to use.

It is best to program to interfaces, rather than classes. Spring reduces the complexity cost of using

interfaces to zero.

JavaBean offers a great way of configuring applications.

OO design is more important than any implementation technology, such as J2EE.

Checked exceptions are overused in Java. A platform should not force you to catch exceptions you are unlikely to recover from Testability is essential and a platform such as spring should help make your code easier to test.

We aim that:

Spring should be a pleasure to use.

Your application codes should not depend on Spring APIs.

Spring should not compete with good existing solutions, but should foster integration.

然后在 2004 年由 Rod Johnson 主导的 Spring 项目推出了 1.0 版本，这彻底地改变了 Java EE 开发的世界，很快人们就抛弃了繁重的 EJB 的标准，迅速地投入到了 Spring 框架中，于是 Spring 成为了现实中 Java EE 开发的标准。Spring 以强大的控制反转（IoC）来管理各类 Java 资源，从而降低了各种资源的耦合；并且提供了极低的侵入性，也就是使用 Spring 框架开发的编码，脱离了 Spring API 也可以继续使用；而 Spring 的面向切面的编程（AOP）通过动态代理技术，允许我们按照约定进行配置编程，进而增强了 Bean 的功能，它擦除了大量重复的代码，如数据库编程所需大量的 try...catch...finally...语句以及数据库事务控制代码逻辑，使得开发人员能够更加集中精力于业务开发，而非资源功能性的开发；Spring 还提供许多整合了当时非常流行的框架的模板，如持久层 Hibernate 的 HibernateTemplate 模板、iBATIS 的 SqlMapClientTemplate 模板等，极大地融合并简化了当时主流技术的使用，使得其展示了强有力的生命力，并延续至今。

值得一提的是，EJB 3.0 的规范也引入了 Spring 的理念，而且整合了 Hibernate 框架的思想，但是也未能挽回其颓势，主要原因在于它的规范还是比较死板，而且比较难整合其他开源框架。其次，它运行在 EJB 容器之中，使用上还是比较困难，性能也不高。

1.2　注解还是 XML

只是在 Spring 早期的 1.x 版本中，由于当时的 JDK 并不能支持注解，因此只能使用 XML。而很快随着 JDK 升级到 JDK5，它加入了注解的新特性，这样注解就被广泛地使用起来，于是 Spring 的内部也分为了两派，一派是使用 XML 的赞同派，一派是使用注解的赞同派。为了简化开发，在 Spring 2.x 之后的版本也引入了注解，不过只是少量的注解，如@Component、@Service 等，但是功能还不够强大，因此对于 Spring 的开发，绝大部分的情况下还是以使用 XML 为主，注解为辅。

到了 Spring 3.0 后，引入了更多的注解功能，于是在 Spring 中产生了这样一个很大的分歧，即是使用注解还是使用 XML？对于 XML 的引入，有些人觉得过于繁复，而对于注解的使用，会使得注解分布得到处都是，难以控制，有时候还需要了解很多框架的内部实现才能准确使用注解开发所需的功能。这个时候大家形成了这样的一个不成文的共识，对于业务类使用注解，例如，对于 MVC 开发，控制器使用@Controller，业务层使用@Service，持久层使用@Repository；而对于一些公用的 Bean，例如，对于数据库（如 Redis）、第三方资源等则使用 XML 进行配置，直至今时今日这样的配置方式还在

企业中广泛地使用着。也许使用注解还是 XML 是一个长期存在的话题，但是无论如何都有道理。

随着注解的功能增强，尤其是 Servlet 3.0 规范的提出，Web 容器可以脱离 web.xml 的部署，使得 Web 容器完全可以基于注解开发，对于 Spring 3.x 和 Spring 4.x 的版本注解功能越来越强大，对于 XML 的依赖越来越少，到了 4.x 的版本后甚至可以完全脱离 XML，因此在 Spring 中使用注解开发占据了主流的地位。与此同时，Pivotal 团队在原有 Spring 的基础上主要通过注解的方式继续简化了 Spring 框架的开发，它们基于 Spring 框架开发了 Spring Boot，所以 Spring Boot 并非是代替 Spring 框架，而是让 Spring 框架更加容易得到快速的使用。Pivotal 团队在 2014 年推出 Spring Boot 的 1.0 版本，该版本使用了特定的方式来进行配置，从而使开发人员不再需要定义样板化的配置。在 2018 年 3 月 Spring Boot 推出了 2.0.0 GA 版本，该版本是基于 Spring 5 的，并引入其最新的功能，能够有效支持 Java 9 的开发。Spring Boot 致力于在蓬勃发展的快速应用开发领域（rapid application development）借助 Java EE 在企业互联网的强势地位成为业界领导者，它也是近年来 Java 开发最令人感到惊喜的项目之一。

随着近年来微服务的流行，越来越多的企业需要快速的开发，而 Spring Boot 除了以注解为主的开发，还有其他的绑定，例如，对服务器进行了绑定和默认对 Spring 的最大化配置，所以开发者能够尽快进行开发代码、发布和测试自己的项目。这符合了现今微服务快速开发、测试和部署的需要，于是越来越多的企业选择 Spring Boot 作为开发的选型，进而使得 Spring Boot 更加兴旺起来。本书主要就是论述 Spring Boot 这一令人激动的开发工具。

1.3 Spring Boot 的优点

谈到 Spring Boot，就让我们先来了解它的优点。依据官方的文档，Spring Boot 的优点如下：

- 创建独立的 Spring 应用程序；
- 嵌入的 Tomcat、Jetty 或者 Undertow，无须部署 WAR 文件；
- 允许通过 Maven 来根据需要获取 starter；
- 尽可能地自动配置 Spring；
- 提供生产就绪型功能，如指标、健康检查和外部配置；
- 绝对没有代码生成，对 XML 没有要求配置。

这段描述告诉我们，首先 Spring Boot 是一个基于 Spring 框架搭建起来的应用，其次它会嵌入 Tomcat、Jetty 或者 Undertow 等服务器，并且不需要传统的 WAR 文件进行部署，也就是说搭建 Spring Boot 项目并不需要单独下载 Tomcat 等传统的服务器；同时提供通过 Maven（或者 Grandle）依赖的 starter，这些 starter 可以直接获取开发所需的相关包，通过这些 starter 项目就能以 Java Application 的形式运行 Spring Boot 的项目，而无须其他服务器配置；对于配置，Spring Boot 提供 Spring 框架的最大自动化配置，大量使用自动配置，使得开发者对 Spring 的配置尽量减少；此外还提供了一些监测、自动检测的功能和外部配置，与此同时没有附加代码和 XML 的配置要求。

约定优于配置，这是 Spring Boot 的主导思想。对于 Spring Boot 而言，大部分情况下存在默认配置，你甚至可以在没有任何定义的情况下使用 Spring 框架，如果需要自定义也只需要在配置文件配置一些属性便可以，十分便捷。而对于部署这些项目必需的功能，Spring Boot 提供 starter 的依赖，