

深入实践 Spring Boot

Deep into Spring Boot

陈韶健 著



深入实践 Spring Boot

Deep into Spring Boot

陈韶健 著



机械工业出版社
China Machine Press

图书在版编目 (CIP) 数据

深入实践 Spring Boot / 陈韶健著. —北京: 机械工业出版社, 2016.10

ISBN 978-7-111-55088-4

I. 深… II. 陈… III. JAVA 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2016) 第 244089 号

深入实践 Spring Boot

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: 李 艺

责任校对: 殷 虹

印 刷: 北京文昌阁彩色印刷有限责任公司

版 次: 2016 年 11 月第 1 版第 1 次印刷

开 本: 186mm×240mm 1/16

印 张: 17.25

书 号: ISBN 978-7-111-55088-4

定 价: 59.00 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

Spring Boot 作为 Java 编程语言的一个全新开发框架，在国内外才刚刚兴起，还未得到普及使用。相比于以往的一些开发框架，Spring Boot 不但使用更加简单，而且功能更加丰富，性能更加稳定而健壮。使用 Spring Boot 开发框架，不仅能提高开发速度，增强生产效率，从某种意义上，可以说是解放了程序员的劳动，而且一种新技术的使用，更能增强系统的稳定性和扩展系统的性能指标。本书就是本着提高开发效率，增强系统性能，促进新技术的普及使用这一目的而写的。

Spring Boot 是在 Spring 框架基础上创建的一个全新框架，其设计目的是简化 Spring 应用的搭建和开发过程，它不但具有 Spring 的所有优秀特性，而且具有如下显著的特点：

- 为 Spring 开发提供更加简单的使用和快速开发的技巧。
- 具有开箱即用的默认配置功能，能根据项目依赖自动配置。
- 具有功能更加强大的服务体系，包括嵌入式服务、安全、性能指标、健康检查等。
- 绝对没有代码生成，可以不再需要 XML 配置，即可让应用更加轻巧和灵活。

Spring Boot 对于一些第三方技术的使用，提供了非常完美的整合，使你在简单的使用中，不知不觉运用了非常高级和先进的技术。

虽然 Spring Boot 具有这么多优秀的特性，但它使用起来并不复杂，而且非常简单，所以不管是 Java 程序开发初学者，还是经验丰富的开发人员，使用 Spring Boot 都是一个理想的选择。

Spring Boot 发展迅速，自从 2014 年 4 月发布了 1.0.0 版本，至今已经发布了 1.4.0 正式版。现在，Spring Boot 正在不同的角落中悄然兴起，估计用不了多久，它将成为

Java 开发的又一个热潮，为众多 Java 开发者追捧。

本书将以一些非常切合生产实际的应用实例，带你一起使用 Spring Boot 框架，开始一段愉快的快速开发和探索之旅。

关于本书

本书以丰富的实例，介绍了如何使用 Spring Boot 开发框架进行基础应用和分布式应用等方面的开发，并且介绍了如何使用 Spring Boot 开发的应用搭建一个高性能的服务平台，同时还对 Spring Boot 的一些核心功能的源代码进行了分析，从而加深对 Spring Boot 的理解。书中对从最基本的入门知识，到数据库的使用，以及界面设计、安全设计等领域都做了详细的介绍和探讨，并在分布式应用系统领域，以平台级应用系统的实例，介绍了如何创建和使用 SSO 管理系统、分布式文件系统，如何使用 Spring Cloud 进行云应用方面的开发，以及如何使用 Docker 发布和构建高可用的分布式系统服务平台。同时，对 Spring Boot 的程序加载、自动配置、数据管理，和 Spring Cloud 的配置管理、发现服务和负载均衡服务等核心功能的源代码做了深入剖析，这样在认识其实现原理的基础上，能更好地使用其相应的功能。

全书分为三个部分：第一部分（第 1 ~ 5 章）介绍基础应用方面的开发，包含简单入门知识、数据库使用、界面设计和安全设计等内容；第二部分（第 6 ~ 9 章）介绍了 Spring Boot 在分布式系统开发和云应用开发等方面的应用以及使用微服务构建高可用的服务平台；第三部分（第 10 ~ 12 章）对 Spring Boot 的程序加载、自动配置和数据管理的实现原理，以及 Spring Cloud 的配置管理、发现服务和负载均衡服务等实现原理进行了深入的剖析。

本书章节编排

第 1 章为 Spring Boot 入门，介绍开发环境的搭建和开发工具的选择及安装配置，并使用一个非常简单的实例，说明如何轻易地使用 Spring Boot 开发框架。

第 2 章使用 Spring Boot 框架演示了以不同于以往的方式，以及如何轻易地使用数据库，并实际演示使用 MySQL、MongoDB、Redis 和 Neo4j 等数据库。

第 3 章使用 Thymeleaf 模板结合一些流行的 JavaScript 插件，介绍了使用 Spring Boot 进行界面设计的方法和技巧。

第 4 章对使用 Spring Boot 提高传统关系型数据库的性能方面做了一些探讨和尝试，

并扩展了使用 JPA 资源库的功能。

第 5 章介绍了如何使用 Spring Boot 结合 Spring Security 进行安全设计，包括登录认证和角色管理、权限管理等内容。

第 6 章介绍如何使用 Spring Security 结合 OAuth2 进行 SSO (Single Sign On) 的设计，并演示如何在分布式应用系统中使用认证授权和安全管理的功能。

第 7 章介绍如何使用 Spring Boot 框架结合分布式文件系统 FastDFS，并使用定制方式和富文本编辑器的方式演示了使用图片上传和建立本地图片库的方法。

第 8 章介绍云应用开发，包括配置管理、发现服务和监控服务的使用，以及如何使用动态路由和断路器的功能，创建高可用的微服务应用。

第 9 章介绍如何使用 Docker 引擎和 docker-compose 工具来发布应用和管理服务，以及如何构建一个高性能的服务平台和怎样使用 Docker 实施负载均衡。

第 10 章分析了 Spring Boot 的应用程序加载和自动配置原理，以及如何以改造加载配置的方式来提高应用的性能。

第 11 章分析了 Spring Boot 使用数据库的实现原理，并演示怎样利用一些技术手段提高和扩展访问数据库的功能。

第 12 章简要分析了微服务中配置管理、发现服务和负载均衡服务的实现原理和部分核心源代码，并使用一个实例说明配置管理中分布式消息的实现机制和原理。

附录 A ~ 附录 D 介绍了 Neo4j、MongoDB、Redis、RabbitMQ 等服务器的安装、配置和基本使用方法。

读者对象

本书适于所有 Java 编程语言开发人员，所有对 Spring Boot 感兴趣并希望使用 Spring Boot 开发框架进行开发的人员，已经使用过 Spring Boot 框架但希望更好地使用 Spring Boot 的开发人员，以及系统设计师、架构师等设计人员。同时，本书对运维人员和 DBA 等也具有一定的参考价值。

实例代码

本书的实例代码可以通过 <https://github.com/chenfromsz?tab=repositories> 查看和下载，推荐根据每章的提示使用 IntelliJ IDEA 通过 GitHub 检出各章的实例工程，这样可以保留原来工程的配置，并且能够直接使用。

反馈与勘误

读者如有反馈意见可以通过 <https://github.com/chenfromsz/correct/issues> 发起新话题与作者进行交互，在这也可能会发布一些勘误信息，以便纠正不可避免的错误。

致谢

首先，非常感谢华阳信通公司，虽然本书的编写过程大都在业余时间完成，但是公司强大的平台给本书的实例提供了更加方便的分享、验证和测试条件。同时在本书的编写过程中，也得到了我们的开发团队和众多朋友的大力支持和帮助，在此表示衷心的感谢！最后感谢华章公司的杨福川和李艺，他们在本书编辑的过程中，提出了一些宝贵而有益的建议，并为本书的出版做了许多工作。

由于时间仓促和水平有限，书中难免出现一些纰漏或不正确的地方，敬请大家批评指正！

前 言

第一部分 基础应用开发

第 1 章 Spring Boot 入门 3

- 1.1 配置开发环境 3
 - 1.1.1 安装 JDK 3
 - 1.1.2 安装 IntelliJ IDEA 4
 - 1.1.3 安装 Apache Maven 4
 - 1.1.4 安装 Git 客户端 5
- 1.2 创建项目工程 8
 - 1.2.1 使用 Maven 新建项目 8
 - 1.2.2 使用 Spring Initializr 新建项目 11
- 1.3 使用 Spring Boot 14
 - 1.3.1 Maven 依赖管理 14
 - 1.3.2 一个简单的实例 17
- 1.4 运行与发布 18
 - 1.4.1 在 IDEA 环境中运行 18
 - 1.4.2 将应用打包发布 19

- 1.5 关于 Spring Boot 配置 22

- 1.6 小结 23

第 2 章 在 Spring Boot 中使用

数据库 24

- 2.1 使用 MySQL 24
 - 2.1.1 MySQL 依赖配置 25
 - 2.1.2 实体建模 25
 - 2.1.3 实体持久化 27
 - 2.1.4 MySQL 测试 29
- 2.2 使用 Redis 33
 - 2.2.1 Redis 依赖配置 33
 - 2.2.2 创建 Redis 服务类 34
 - 2.2.3 Redis 测试 36
- 2.3 使用 MongoDB 38
 - 2.3.1 MongoDB 依赖配置 38
 - 2.3.2 文档建模 39
 - 2.3.3 文档持久化 40
 - 2.3.4 MongoDB 测试 41
- 2.4 使用 Neo4j 43
 - 2.4.1 Neo4j 依赖配置 43

2.4.2	节点和关系实体建模	43	4.1.1	配置 Druid 依赖	76
2.4.3	节点实体持久化	45	4.1.2	关于 XML 配置	76
2.4.4	Neo4j 测试	46	4.1.3	Druid 数据源配置	77
2.5	小结	49	4.1.4	开启监控功能	78
第 3 章	Spring Boot 界面设计	50	4.2	扩展 JPA 功能	80
3.1	模型设计	50	4.2.1	扩展 JPA 接口	81
3.1.1	节点实体建模	51	4.2.2	装配自定义的扩展接口	83
3.1.2	关系实体建模	51	4.2.3	使用扩展接口	85
3.1.3	分页查询设计	52	4.3	使用 Redis 做缓存	86
3.2	控制器设计	53	4.3.1	使用 Spring Cache 注解	86
3.2.1	新建控制器	53	4.3.2	使用 RedisTemplate	88
3.2.2	查看控制器	53	4.4	Web 应用模块	91
3.2.3	修改控制器	54	4.4.1	引用数据管理模块	91
3.2.4	删除控制器	55	4.4.2	Web 应用配置	92
3.2.5	分页查询控制器	55	4.5	运行与发布	94
3.3	使用 Thymeleaf 模板	56	4.6	小结	95
3.3.1	Thymeleaf 配置	56	第 5 章	Spring Boot 安全设计	96
3.3.2	Thymeleaf 功能简介	57	5.1	依赖配置管理	96
3.4	视图设计	60	5.2	安全策略配置	97
3.4.1	列表视图设计	60	5.2.1	权限管理规则	98
3.4.2	新建视图设计	64	5.2.2	登录成功处理器	99
3.4.3	查看视图设计	68	5.2.3	防攻击策略	100
3.4.4	修改视图设计	70	5.2.4	记住登录状态	102
3.4.5	删除视图设计	72	5.3	登录认证设计	103
3.5	运行与发布	73	5.3.1	用户实体建模	103
3.6	小结	74	5.3.2	用户身份验证	104
第 4 章	提高数据库访问性能	75	5.3.3	登录界面设计	106
4.1	使用 Druid	75	5.3.4	验证码验证	108
			5.4	权限管理设计	109

5.4.1	权限管理配置	109
5.4.2	权限管理过滤器	110
5.4.3	权限配置资源管理器	111
5.4.4	权限管理决断器	112
5.5	根据权限设置链接	113
5.6	运行与发布	116
5.6.1	系统初始化	116
5.6.2	系统运行与发布	118
5.7	小结	119

第二部分 分布式应用开发

第 6 章	Spring Boot SSO	123
6.1	模块化设计	123
6.2	登录认证模块	124
6.2.1	使用 OAuth2	124
6.2.2	创建数字证书	125
6.2.3	认证服务端配置	125
6.3	安全配置模块	128
6.4	SSO 客户端	129
6.4.1	客户端配置	129
6.4.2	登录登出设计	130
6.5	共享资源服务	132
6.5.1	提供共享资源接口	133
6.5.2	使用共享资源	134
6.5.3	查询登录用户的详细 信息	135
6.6	运行与发布	136
6.7	小结	138

第 7 章	使用分布式文件系统	139
7.1	FastDFS 安装	139
7.1.1	下载安装包	141
7.1.2	安装服务	141
7.1.3	Tracker Server 配置	142
7.1.4	Storage Server 配置	145
7.1.5	启动服务	148
7.1.6	客户端测试	148
7.2	FastFDS 客户端	149
7.2.1	客户端配置	150
7.2.2	客户端服务类	150
7.3	使用定制方式上传图片	151
7.3.1	实体建模	151
7.3.2	上传图片	152
7.4	使用富文本编辑器上传图片	156
7.4.1	使用富文本编辑器	156
7.4.2	实现文件上传	157
7.5	使用本地文件库	158
7.5.1	本地文件库建模	159
7.5.2	文件保存方法	159
7.5.3	文件库管理	161
7.6	运行与发布	163
7.7	小结	164

第 8 章	云应用开发	165
8.1	使用配置管理	166
8.1.1	创建配置管理服务器	167
8.1.2	使用配置管理的客户端	168
8.1.3	实现在线更新	171
8.1.4	更新所有客户端的配置	172

11.1.1	数据源类型和驱动	219	12.2	发现服务源代码剖析	235
11.1.2	支持的数据库种类	220	12.2.1	服务端的服务注册功能	236
11.1.3	与数据库服务器建立 连接	221	12.2.2	客户端注册和提取服务 列表	238
11.2	数据存取功能实现原理	222	12.3	负载均衡源代码剖析	240
11.2.1	实体建模源代码分析	222	12.4	分布式消息实现原理演示	244
11.2.2	持久化实现原理	225	12.4.1	消息生产者	244
11.3	扩展数据存取的功能	227	12.4.2	消息消费者	245
11.3.1	扩展 JPA 功能	227	12.5	小结	247
11.3.2	扩展 Neo4j 功能	228	附录 A	安装 Neo4j	248
11.4	小结	230	附录 B	安装 MongoDB	251
第 12 章 微服务核心技术实现			附录 C	安装 Redis	253
	原理	231	附录 D	安装 RabbitMQ	256
12.1	配置管理实现原理	232	结束语		262
12.1.1	在线更新流程	232			
12.1.2	更新消息的分发原理	233			



第一部分 *Part 1*

基础应用开发

- 第 1 章 Spring Boot 入门
 - 第 2 章 在 Spring Boot 中使用数据库
 - 第 3 章 Spring Boot 界面设计
 - 第 4 章 提高数据库访问性能
 - 第 5 章 Spring Boot 安全设计
- 

这一部分从搭建开发环境，简单入门，到使用数据库、界面设计、安全管理等一系列内容，介绍了使用 Spring Boot 框架进行基础应用开发的方法。

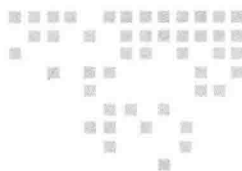
第 1 章介绍了开发环境的搭建和开发工具的选择和安装，并以一个非常简单的实例，演示了如何使用 Spring Boot 框架创建工程和发布应用。

第 2 章介绍了如何用 Spring Boot 特有的方式，使用当前流行的数据库：MySQL、Redis、MongoDB、Neo4j 等。

第 3 章介绍如何使用 Thymeleaf 模板结合一些流行的 JavaScript 插件，设计应用界面。

第 4 章使用 Druid 数据库连接池和 Redis 做缓存来尝试提升关系型数据库的访问性能，并扩展了 JPA 的资源库功能。

第 5 章在 Spring Boot 中使用 Spring Security 为应用系统进行安全设计，实现了登录认证和权限管理方面的功能。



Spring Boot 入门

在使用 Spring Boot 框架进行各种开发体验之前，要先配置好开发环境。首先安装 JDK，然后选择一个开发工具，如 Eclipse IDE 和 IntelliJ IDEA（以下简称 IDEA）都是不错的选择。对于开发工具的选择，本书极力推荐使用 IDEA，因为它为 Spring Boot 提供了许多更好和更贴切的支持，本书的实例都是使用 IDEA 创建的。同时，还需要安装 Apache Maven 和 Git 客户端。所有这些都准备好之后，我们就能开始使用 Spring Boot 了。

1.1 配置开发环境

下面的开发环境配置主要以使用 Windows 操作系统为例，如果你使用的是其他操作系统，请对照其相关配置进行操作。

1.1.1 安装 JDK

JDK (Java SE Development Kit) 需要 1.8 及以上版本，可以从 Java 的官网 <http://www.oracle.com/technetwork/java/javase/downloads/index.html> 下载安装包。如果访问官网速度慢的话，也可以通过百度搜索 JDK，然后在百度软件中心下载符合你的 Windows 版本和配置的 JDK1.8 安装包。

安装完成后，配置环境变量 `JAVA_HOME`，例如，使用路径 `D:\Program Files\Java\jdk1.8.0_25`（如果你安装的是这个目录的话）。`JAVA_HOME` 配置好之后，将 `%JAVA_HOME%\bin` 加入系统的环境变量 `path` 中。完成后，打开一个命令行窗口，输入命令 `java-version`，如果能正确输出版本号则说明安装成功了。输出版本的信息如下：

```
C:\Users\Alan>java-version
java version "1.8.0_25"
Java(TM) SE Runtime Environment (build 1.8.0_25-b18)
Java HotSpot(TM) 64-Bit Server VM (build 25.25-b02, mixed mode)
```

1.1.2 安装 IntelliJ IDEA

IDEA 需要 14.0 以上的版本，可以从其官网 <http://www.jetbrains.com/> 下载免费版，本书的实例是使用 IDEA 14.1.15 版本开发的。IDEA 已经包含 Maven 插件，版本是 3.0.5，这已经能够适用我们开发的要求。安装完成后，打开 IDEA，将显示如图 1-1 所示的欢迎界面，在这里可以看到 IDEA 的版本号。

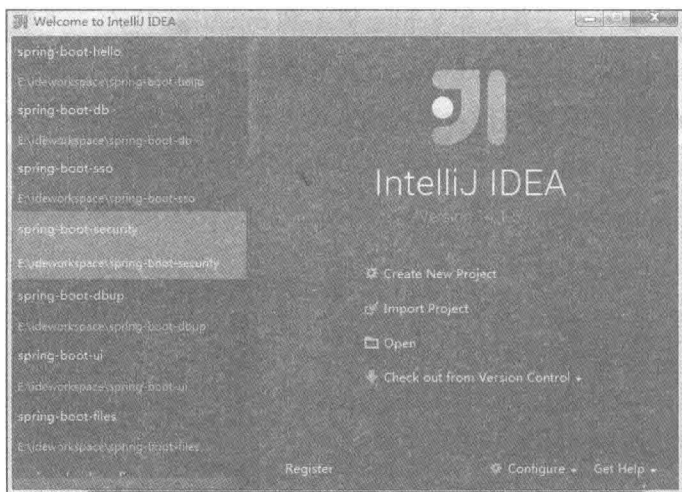


图 1-1 IntelliJ IDEA 欢迎界面

1.1.3 安装 Apache Maven

为了能够在命令行窗口中使用 Maven 来管理工程，可以安装一个 Maven 管理工具。通过 Maven 的官网 <http://maven.apache.org/download.cgi> 下载 3.0.5 以上的版本，下

载完后解压缩即可，例如，解压到 D: 盘上是不错的做法，然后将 Maven 的安装路径（如 D:\apache-maven-3.2.3\bin）也加入 Windows 的环境变量 path 中。安装完成后，在命令行窗口中执行指令：`mvn-v`，将输出如下的版本信息以及系统的一些环境信息。

```
C:\Users\Alan>mvn-v
Apache Maven 3.2.3 (33f8c3e1027c3ddde99d3cdebad2656a31e8fdf4; 2014-08-12T04:58:10+08:00)
Maven home: D:\apache-maven-3.2.3\bin\..
Java version: 1.8.0_25, vendor: Oracle Corporation
Java home: D:\Program Files\Java\jdk1.8.0_25\jre
Default locale: zh_CN, platform encoding: GBK
OS name: "windows 7", version: "6.1", arch: "amd64", family: "dos"
```

建议更改 IDEA 中 Maven 资源库的存放路径，可以先在 Maven 安装路径中创建一个资源库目录，如 `repository`。然后打开 Maven 的配置文件，即安装目录 `conf` 中的 `settings.xml`，找到下列代码，将路径更改为 `repository` 所在的位置，并保存在注释符下面。

例如找到下列代码行：

```
<localRepository>/path/to/local/repo</localRepository>
```

复制出来改为如下所示：

```
<localRepository>D:\apache-maven-3.2.3\repository</localRepository>
```

改好后可以拷贝一份 `settings.xml` 放置在 `${user.home}/.m2/` 下面，这样做可以不用修改 IDEA 的 Maven 这个配置。在图 1-2 所示的 Maven 配置界面中，User Settings File 保持了默认位置，Local Repository 使用了上面设置的路径 `D:\apache-maven-3.2.3\repository`，而 Maven 程序还是使用了 IDEA 自带的版本。

1.1.4 安装 Git 客户端

由于本书的实例工程都存放在 GitHub (<https://github.com/>) 中，所以还需要在 GitHub 中免费注册一个用户（可以通过 E-mail 直接注册免费用户），以方便在 IDEA 中从 GitHub 检出本书的实例工程。当然，如果不想注册，通过普通下载的方法也能取得实例工程的源代码。GitHub 是世界级的代码库服务器，如果你愿意，也可以将它作为你的代码库服务器，在这里还可以搜索到全世界的开发者分享出来的源程序。图 1-3 是打开 GitHub 的首页。