

Mastering Spring Boot 2.0

精通

Spring Boot 2.0

[印] 迪内什·拉吉普特 著 刘 璋 译



清华大学出版社

精通 Spring Boot 2.0

[印] 迪内什·拉吉普特 著

刘 璋 译

清华大学出版社

北 京

内 容 简 介

本书详细阐述了与 Spring Boot 2.0 相关的基本解决方案，主要包括定制 auto-configuration、Spring CLI 和 Actuator、Spring Cloud 和配置操作、Spring Cloud Netflix 和 Service Discovery、构建 Spring Boot RESTful 微服务、利用 Netflix Zuul 创建 API 网关、利用 Feign 客户端简化 HTTP API、构建事件驱动和异步响应式系统、利用 Hystrix 和 Turbine 构建弹性系统、测试 Spring Boot 应用程序、微服务的容器化、API 管理器、云部署（AWS）、生产服务监视和最佳实践等内容。此外，本书还提供了相应的示例、代码，以帮助读者进一步理解相关方案的实现过程。

本书适合作为高等院校计算机及相关专业的教材和教学参考书，也可作为相关开发人员的自学教材和参考手册。

Copyright © Packt Publishing 2018. First published in the English language under the title

Mastering Spring Boot 2.0.

Simplified Chinese-language edition © 2019 by Tsinghua University Press. All rights reserved.

本书中文简体字版由 Packt Publishing 授权清华大学出版社独家出版。未经出版者书面许可，不得以任何方式复制或抄袭本书内容。

北京市版权局著作权合同登记号 图字：01-2018-5178

本书封面贴有清华大学出版社防伪标签，无标签者不得销售。

版权所有，侵权必究。侵权举报电话：010-62782989 13701121933

图书在版编目（CIP）数据

精通 Spring Boot 2.0 /（印）迪内什·拉吉普特（Dinesh Rajput）著；刘璋译．—北京：清华大学出版社，2019

书名原文：Mastering Spring Boot 2.0

ISBN 978-7-302-53024-4

I. ①精… II. ①迪… ②刘… III. ①JAVA 语言-程序设计 IV. ①TP312.8

中国版本图书馆CIP数据核字（2019）第093916号

责任编辑：贾小红

封面设计：刘超

版式设计：魏远

责任校对：马子杰

责任印制：刘海龙

出版发行：清华大学出版社

网 址：<http://www.tup.com.cn>，<http://www.wqbook.com>

地 址：北京清华大学学研大厦A座 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

印装者：三河市铭诚印务有限公司

经 销：全国新华书店

开 本：185mm×230mm

印 张：21

字 数：418千字

版 次：2019年6月第1版

印 次：2019年6月第1次印刷

定 价：109.00元

译者序

Spring Boot 是市场上开发 Web、企业和云软件的最佳工具之一。Spring Boot 通过减少样板代码的数量，提供了现成的特性和简单的部署模型，从而大大简化了复杂软件的构建过程。

本书将解决与 Spring Boot 2.0 强大的可配置性和灵活性相关的应用程序设计问题。读者将了解 Spring Boot 2.0 配置如何在后台工作、如何覆盖默认配置，以及如何使用高级技术将 Spring Boot 2.0 应用程序投入至实际应用中。此外，本书还将向读者介绍 Spring 生态系统中相对较新的主题——响应式编程。在阅读完本书后，读者将精通于构建和部署 Spring Boot 2.0 应用程序。同时，本书还提供了相应的示例、代码，以帮助读者进一步理解相关方案的实现过程。

在本书的翻译过程中，除刘璋外，张博、刘晓雪、王辉、张华臻、刘祎等人也参与了部分翻译工作，在此一并表示感谢。

由于译者水平有限，难免有疏漏和不妥之处，恳请广大读者批评指正。

译者

前 言

Pivotal 最近发布了 Spring Boot 2.0，以支持响应式编程和云计算。Spring Boot 2.0 引入了诸多新特性和增强方案，本书也将对此进行逐一介绍。另外，本书还将引领读者深入理解 Spring Boot 和云微服务架构方面的知识。

当前，许多公司已经将 Spring Boot 作为企业应用程序开发的主要框架，对于采用微服务架构的 REST API 尤其如此。对于 Spring Boot 来说，我们并不需要使用外部企业服务器。本书旨在阐述本地云应用程序背后所采用的常见设计方案，以及如何在 Spring Boot 2.0 的 Spring Cloud 模块中对其予以实现。其间，作者还进一步总结了分布式设计日志记录机制和应用程序开发过程中的一些最佳实践方案。

本书共 15 章，涵盖了从基于微服务的云应用程序开发到微服务的部署（使用虚拟机或 Docker 等容器）的方方面面，包括如何使用 Rest 模板、Spring Cloud Netflix Feign 实现微服务架构中服务间的通信；如何使用 Spring Cloud Stream 和 Kafka 构建一个事件驱动的弹性系统。这一部分内容还向读者展示了如何使用 Hystrix 和 Turbine 进行监视。最后，本书还将解释如何测试和构建 API，并将其部署到容器（如 Docker）以及云中（如 AWS）。

适用读者

本书适用于各种层次的 Java 开发人员，他们希望学习 Spring Boot 和 Spring Cloud 并将其作为基于企业分布式云的应用程序。因此，当采用基于 Spring Boot 2.0 和 Spring Cloud 的微服务架构时，本书对企业级 Java 和 Spring 开发人员来说十分有用，进而帮助他们深入理解本地云设计模式，以及微服务体系结构如何解决分布式应用程序中本地云基础设施中的常见设计问题，并可将本书中的示例结合自己的项目加以使用。在阅读本书之前，读者应具备 Core Java、Spring Core Framework 以及 Spring Boot 方面的基础知识。

本书内容

第 1 章整体介绍了 Spring Boot 2.0 及其新特性，包括某些核心组件，以及 Spring Boot

的某些改进措施。

第 2 章阐述了 Spring Boot 的自动配置特性，同时进一步解释了如何覆盖默认的自动配置机制。

第 3 章通过多种方式创建 Spring Boot 应用程序，涉及 Spring Boot 的 Web 接口、STS IDE 以及 Spring Boot CLI。本章深入讨论了 Spring Boot CLI，以及如何在机器设备上安装 Spring Boot CLI，并以此创建 Spring Boot 应用程序。另外，本章还将通过 Actuator 介绍 Spring Boot 的生产环境特性。

第 4 章讨论如何构建配置服务器，并向客户端应用程序提供 Git 存储库中的一组配置文件。在本章中，读者将学习到与 Spring Cloud 配置服务相关的知识，以及如何构建和使用配置服务。

第 5 章介绍 Spring Cloud Netflix 和基于 Eureka 的 Service Discovery。

第 6 章构建一个 RESTful 原子微服务，该服务使用 Spring Cloud 和 Spring 数据在内存数据库（HSQL 或 H2）上执行 CRUD 操作，以使该服务能够向 Eureka 服务器进行服务发现注册。

第 7 章探讨微服务通信的 API 网关模式，无论是来自 UI 组件还是来自内部服务调用。另外，还将使用 Netflix API Zuul 实现一个 API 网关，并了解如何在应用程序中设置 Zuul 代理。

第 8 章对 Feign 及其工作方式加以介绍，其中包含了针对业务需求相关的、详细的 Feign 扩展/定制方式，其中展示了自定义编码器、解码器、Hystrix 和单元测试异常处理方面的参考实现。

第 9 章讲述了如何采用事件驱动架构并作为本地云应用程序构建事件驱动型微服务。对于分布式系统中的数据一致性处理，本章将考查一些重要的概念和主题。

第 10 章通过参考实现探讨断路器模式，其中涉及 Netflix Hystrix 库的使用，配置 Turbine 仪表盘以整合来自多项服务的 Hystrix 流。

第 11 章通过 JUnit 和 Mockito 讨论 Spring Boot Services 单元测试。其中，全部参考实现都将完成相应的单元测试。因此，本章内容更多地是整合了微服务的各种测试机制。

第 12 章介绍容器，并在 Docker 中运行第 11 章中构建的服务、编写 Dockerfile、使用 docker-compose 编排容器，并在 Kubernetes 中提供编排示例。

第 13 章探讨分布式系统中的 API 管理，设置 KONG 开源 API 管理器、在 KONG API 管理器中配置前述章节中的 API 端点、针对 API 标准引入 Swagger。最后，本章还将展示速率限制以及基于 KONG 的日志记录。

第 14 章介绍如何在 AWS EC2 实例中手动部署微服务，以及 CloudFormation 脚本的

应用方式。

第 15 章详细介绍构建分布式系统的一些最佳实践,并深入讨论生产环境下服务的性能监控方案。

软件环境和资源下载

本书内容可供读者独立阅读。但是,为了更好地理解书中的相关示例,读者需要安装 Java 8。对此,可访问 <http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html> 下载 Java 8。此外,读者还可根据个人喜好安装相应的 IDE,如 Software Spring Tool Suite。读者可访问 <https://spring.io/tools/sts/all>,并根据个人操作系统下载 Spring Tool Suite (STS) 的最新版本。Java 8 和 STS 也适用于其他平台,如 Windows、macOS 和 Linux。

读者可访问 <http://www.packtpub.com> 并通过个人账户下载示例代码文件。另外,在 <http://www.packtpub.com/support> 中注册成功后,我们将以电子邮件的方式将相关文件发与读者。

读者可根据下列步骤下载代码文件:

- (1) 利用电子邮件地址和密码登录或注册我们的网站。
- (2) 选择 SUPPORT 选项卡。
- (3) 单击 Code Downloads & Errata。
- (4) 在 Search 文本框中输入书名。

当文件下载完毕后,确保使用下列最新版本软件解压文件夹:

- Windows 系统下的 WinRAR/7-Zip。
- Mac 系统下的 Zipeg/iZip/UnRarX。
- Linux 系统下的 7-Zip/PeaZip。

另外,读者还可访问 GitHub 获取本书的代码包,对应网址为 <https://github.com/PacktPublishing/Mastering-Spring-Boot-2.0>。此外,读者还可访问 <https://github.com/PacktPublishing/> 以了解丰富的代码和视频资源。

本书约定

本书通过不同的文本风格区分相应的信息类型。下面通过一些示例对此类风格以及具

体含义的解释予以展示。

代码块如下所示：

```
@RestController
class HelloController {
    @GetMapping("/")
    String hello() {
        "Hello World!!!"
    }
}
```


当某个代码块希望引起读者的足够重视时，一般会采用黑体表示，如下所示：

```
<dependencies>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
  </dependency>
</dependencies>
```

命令行输入或输出则采用下列方式表达：

```
$ Spring run HelloController.groovy
```

 图标表示较为重要的说明事项。

 图标表示提示信息和操作技巧。

读者反馈和客户支持

欢迎读者对本书的建议或意见予以反馈。

对此，读者可向 feedback@packtpub.com 发送邮件，并以书名作为邮件标题。若读者对本书有任何疑问，均可发送邮件至 questions@packtpub.com，我们将竭诚为您服务。

若读者针对某项技术具有专家级的见解，抑或计划撰写书籍或完善某部著作的出版工作，则可访问 www.packtpub.com/authors。

勘误表

尽管我们在最大程度上做到尽善尽美，但错误依然在所难免。如果读者发现谬误之

处，无论是文字错误抑或是代码错误，还望不吝赐教。对此，读者可访问 <http://www.packtpub.com/submit-errata>，选取对应书籍，单击 Errata Submission Form 超链接，并输入相关问题的详细内容。

版权须知

一直以来，互联网上的版权问题从未间断，Packt 出版社对此类问题异常重视。若读者在互联网上发现本书任意形式的副本，请告知网络地址或网站名称，我们将对此予以处理。关于盗版问题，读者可发送邮件至 copyright@packtpub.com。

问题解答

若读者对本书有任何疑问，均可发送邮件至 questions@packtpub.com，我们将竭诚为您服务。

目 录

第 1 章 Spring Boot 2.0	1
1.1 Spring Boot 概述	2
1.2 利用 Spring Boot 简化应用程序开发	3
1.3 Spring Boot 中的核心组件	5
1.3.1 Spring Boot Starter	5
1.3.2 Spring Boot Starter Parent POM	7
1.3.3 Spring Boot auto-configuration	7
1.3.4 启用 Spring Boot auto-configuration	9
1.3.5 Spring Boot CLI	11
1.3.6 Spring Boot Actuator	11
1.4 设置 Spring Boot 工作区	12
1.4.1 利用 Maven 设置 Spring Boot	13
1.4.2 利用 Gradle 设置 Spring Boot	13
1.5 开发第一个 Spring Boot 应用程序	15
1.5.1 使用 Web 界面	15
1.5.2 利用 STS IDE 创建 Spring Boot 项目	17
1.6 实现 REST 服务	19
1.7 Spring Boot 2.0 中的新特性	22
1.8 本章小结	22
第 2 章 定制 auto-configuration	25
2.1 理解 auto-configuration	26
2.2 定制 Spring Boot	27
2.2.1 利用 Spring Boot 属性进行定制	27
2.2.2 替换已生成的 Bean	29
2.2.3 禁用特定的 auto-configuration 类	29
2.2.4 修改库的依赖关系	30
2.3 基于属性的配置外部化	31

2.3.1	属性的评估顺序	31
2.3.2	重命名 Spring 应用程序中的 application.properties	32
2.4	外部配置应用程序属性	33
2.5	基于日志记录的调优	35
2.6	YAML 配置文件.....	36
2.6.1	针对属性的 YAML.....	36
2.6.2	单一 YAML 文件中的多个属性.....	37
2.7	定制应用程序错误页面	37
2.8	本章小结	39
第 3 章	Spring CLI 和 Actuator	41
3.1	使用 Spring Boot CLI	41
3.1.1	安装 Spring Boot CLI	42
3.1.2	从安装文件中手动安装 Spring Boot CLI.....	42
3.1.3	使用 SDKMAN!安装 Spring Boot CLI.....	43
3.1.4	利用 OSX Homebrew 安装 Spring Boot CLI.....	43
3.2	使用 Initializr	44
3.3	Spring Boot Actuator.....	48
3.3.1	在应用程序中启用 Spring Boot Actuator	49
3.3.2	分析 Actuator 的端点	49
3.3.3	显示配置细节	51
3.3.4	显示指标端点	56
3.3.5	显示应用程序信息	57
3.3.6	关闭应用程序	59
3.3.7	自定义 Actuator 端点	59
3.3.8	创建一个自定义端点	64
3.4	Actuator 端点的安全性	66
3.5	Spring Boot 2.x 中的 Actuator	67
3.6	本章小结	68
第 4 章	Spring Cloud 和配置操作	69
4.1	原生云应用程序架构	69
4.1.1	微服务架构	71

4.1.2	微服务的优点	72
4.1.3	微服务面临的挑战	73
4.2	Spring Cloud 简介	74
4.2.1	云和微服务程序的构造块	74
4.2.2	Spring Cloud 应用	76
4.3	配置 Spring Cloud 应用程序	77
4.4	创建配置生成器——Spring Cloud Config Server	78
4.5	实现 Cloud Config Server	79
4.5.1	配置 application.properties 文件	80
4.5.2	创建 Git 存储库作为配置存储	80
4.6	利用模式配置多个存储库	83
4.6.1	身份验证	84
4.6.2	force-pull 属性	85
4.7	创建 Spring Cloud 客户端	85
4.8	本章小结	87
第 5 章	Spring Cloud Netflix 和 Service Discovery	89
5.1	Spring Cloud Netflix 简介	89
5.2	微服务架构中的 Service Discovery	90
5.3	实现 Service Discovery——Eureka Server	92
5.3.1	Maven 构建配置文件	92
5.3.2	Gradle 构建配置文件	93
5.3.3	启用 Eureka 服务器作为 Discovery Service 服务器	94
5.4	实现 Service Discovery——Eureka 客户端	96
5.4.1	添加 Maven 依赖关系配置	96
5.4.2	Gradle 构建配置	98
5.5	利用 Eureka 注册客户端	99
5.5.1	使用 REST 服务	102
5.5.2	使用 EurekaClient	102
5.5.3	Feign Client	107
5.6	本章小结	111

第 6 章	构建 Spring Boot RESTful 微服务	113
6.1	基于 Spring Boot 的微服务	113
6.1.1	bootstrap.yml 和 application.yml 简介	114
6.1.2	简单的微服务示例	115
6.2	Spring Data 简介	128
6.2.1	Apache Ignite 存储库	129
6.2.2	Spring Data MongoDB	129
6.2.3	Spring Data JPA	130
6.3	本章小结	130
第 7 章	利用 Netflix Zuul 创建 API 网关	133
7.1	API 网关模式需求	133
7.1.1	API Gateway 模式的优点	135
7.1.2	API Gateway 的一些缺点	135
7.1.3	API Gateway 模式组件	135
7.2	利用 Netflix Zuul Proxy 实现 API Gateway	136
7.2.1	利用 Maven 依赖关系包含 Zuul	137
7.2.2	启用 Zuul 服务代理	137
7.2.3	配置 Zuul 属性	138
7.2.4	添加过滤器	141
7.3	本章小结	144
第 8 章	利用 Feign 客户端简化 HTTP API	145
8.1	Feign 基础知识	145
8.2	在云应用程序中包含 Feign	148
8.2.1	重载 Feign 的默认配置	153
8.2.2	创建 Feign 客户端	155
8.2.3	Feign 继承机制	156
8.2.4	多重继承	156
8.3	Feign 客户端的高级应用	157
8.4	异常处理	158
8.5	自定义编码器和解码器	159
8.5.1	自定义编码器	160

8.5.2 自定义解码器	161
8.6 Feign 和 Hystrix	161
8.7 Feign 客户端单元测试	163
8.8 本章小结	164
第 9 章 构建事件驱动和异步响应式系统	165
9.1 事件驱动型架构模式	165
9.1.1 调停者拓扑	165
9.1.2 代理拓扑	166
9.2 响应式编程简介	167
9.2.1 Spring Reactive	167
9.2.2 ReactiveX	168
9.3 命令查询的责任分离简介	168
9.3.1 Event Sourcing 模式简介	170
9.3.2 最终一致性	171
9.4 构建事件驱动型响应式异步系统	172
9.5 Spring Cloud Streaming 简介	173
9.5.1 向应用程序中添加 Kafka	174
9.5.2 安装和运行 Kafka	175
9.5.3 Kafka 配置属性	175
9.5.4 用于写入 Kafka 的服务	176
9.5.5 Rest API 控制器	177
9.5.6 监听 Kafka 主题	177
9.6 本章小结	181
第 10 章 利用 Hystrix 和 Turbine 构建弹性系统	183
10.1 断路器模式	184
10.2 使用 Hystrix library	186
10.3 在应用程序中配置 Hystrix	187
10.3.1 Maven 依赖关系	188
10.3.2 启用断路器	188
10.3.3 向服务中添加 Hystrix 注解	189
10.3.4 错误传递	192

10.4	在客户服务中实现 REST 控制器.....	192
10.5	构建和测试客户服务.....	195
10.6	自定义默认的配置项.....	196
10.7	Hystrix Metrics Stream.....	198
10.8	在项目中实现 Hystrix Dashboard.....	199
10.9	Turbine 仪表盘.....	201
10.10	基于 Hystrix 和 Feign 的 REST 使用者.....	204
10.11	本章小结.....	206
第 11 章	测试 Spring Boot 应用程序.....	207
11.1	测试驱动开发.....	207
11.2	单元测试机制.....	208
11.2.1	单元测试的优点.....	211
11.2.2	单元测试的缺点.....	212
11.2.3	其他模拟库.....	212
11.3	集成测试.....	212
11.3.1	Spring 测试的优点.....	214
11.3.2	激活测试类的配置.....	214
11.4	Spring Boot 应用程序的 JUnit 测试.....	214
11.5	使用 Mockito 模拟服务.....	216
11.6	测试 RESTful 服务契约的 Postman.....	217
11.7	本章小结.....	220
第 12 章	微服务的容器化.....	221
12.1	微服务架构的容器.....	222
12.1.1	虚拟机和容器.....	222
12.1.2	容器方案的优点.....	224
12.1.3	面向容器方案的缺点.....	224
12.2	Docker 简介.....	225
12.2.1	安装 Docker.....	226
12.2.2	在 Linux 上安装 Docker.....	226
12.2.3	在 Windows 中安装 Docker.....	227
12.2.4	Docker 架构.....	229

12.2.5	Docker 引擎	231
12.2.6	Docker 容器	232
12.2.7	编写 Dockerfile	233
12.3	Docker 化 Spring Boot 应用程序	235
12.4	利用 Maven 创建 Docker 镜像	239
12.5	Docker Compose 简介	240
12.5.1	安装 Docker Compose	241
12.5.2	使用 Docker Compose	242
12.5.3	编写 docker-compose 文件	242
12.5.4	基于 docker-compose 文件的编排操作	244
12.5.5	利用 docker-compose 和负载均衡扩展容器	247
12.6	Kubernetes 简介	248
12.7	本章小结	249
第 13 章	API 管理器	251
13.1	API 管理	251
13.1.1	API 管理软件的优点	252
13.1.2	API 管理工具	252
13.2	速率限制	252
13.3	KONG 简介	253
13.3.1	基于 KONG 架构的微服务 REST API	254
13.3.2	未采用 KONG 架构的 API 应用	255
13.3.3	安装 KONG	255
13.3.4	使用 KONG API	257
13.4	Swagger	265
13.4.1	Swagger 应用	265
13.4.2	在微服务中使用 Swagger	266
13.4.3	Swagger 的优点	277
13.5	本章小结	278
第 14 章	云部署 (AWS)	279
14.1	AWS EC2 实例	279
14.2	AWS 上的微服务架构	284

14.3	在 AWS EC2 上安装 Docker.....	289
14.4	在 AWS EC2 上运行微服务.....	291
14.5	本章小结.....	293
第 15 章	生产服务监视和最佳实践.....	295
15.1	监视容器.....	295
15.2	日志机制所面临的挑战.....	295
15.3	微服务架构的中心日志方案.....	297
15.3.1	基于 ELK 栈的日志聚合.....	299
15.3.2	使用 Sleuth 的请求跟踪.....	306
15.3.3	基于 Zipkin 的请求跟踪.....	310
15.4	本章小结.....	315