

关于本书

《Spring 微服务实战》是专门为使用 Java 或 Spring 工作，并且需要实践的建议和举例说明如何建立和实施微服务应用的开发者而编写的。在编写本书的时候，我是围绕核心微服务模式，使用 Spring Boot 和 Spring Cloud 例子在实战中对模式进行验证。因此，你会找到具体的微服务设计模式，它几乎在每章都讨论，并且有使用 Spring Boot 和 Spring Cloud 实现该模式的例子。

本书适合读者

- 你是一位拥有 java 构建分布式应用程序经验的开发人员（1-3 年）。
- 你了解或使用 Spring（1 年以上）。
- 你有兴趣学习如何构建微服务的应用。
- 你对如何使用微服务构建云应用有兴趣。
- 你想知道构建微服务应用，与 Java 和 Spring 相关的技术。
- 你对将微服务应用部署到云涉及的内容有兴趣。

章节安排

《Spring 微服务实战》包含 10 个章节和两个附录：

- 第 1 章向您介绍为什么微服务架构是一个构建应用程序重要的、实质性的方法，特别是构建基于云的应用程序。
- 第 2 章逐步引导您使用 Spring Boot 构建第一个基于 REST 的微服务。通过架构师、应用工程师和运维工程师的视角，指导你如何看待微服务。
- 第 3 章介绍使用 Spring Cloud Config 如何管理微服务配置。Spring Cloud Config 能够帮助将服务配置信息集中存储在一个单一的仓库，并对配置信息版本实现有效的控制，且能够让所有的服务实例可重复的访问。

- 第 4 章向您介绍了第一种微服务路由模式：服务发现。在这一章中，您将学习到如何使用 Spring Cloud 和 Netflix 的 Eureka 服务从客户消费服务时抽象服务的具体位置。
- 第 5 章讨论当一个或多个微服务实例关闭或处于被降级的状态时，保护微服务消费者的方法。本章将演示如何使用 Spring Cloud 和 Netflix Hystrix (和 Netflix Ribbon)实现调用的客户端负载均衡，断路器模式，后退模式和舱壁模式。
- 第 6 章涵盖了微服务路由模式：服务网关。使用 Spring Cloud 与 Netflix 的 Zuul 服务器，您将所有的微服务创建单一的访问入口。我们将讨论如何使用 Zuul 的过滤器 API 对所有流经服务网关的服务，建立强制的访问策略。
- 第 7 章介绍了使用 Spring Cloud security 和 OAuth2 如何实现服务的认证和授权。我们将介绍建立一个 OAuth2 服务来保护您的服务以及在 OAuth2 实施过程中，如何使用 JavaScript Web Tokens (JWT) 。
- 第 8 章介绍使用 Spring Cloud Stream 和 Apache Kafka，如何将异步消息传递到微服务。
- 第 9 章展示了如何实现日志记录的常用日志模式，如日志关联分析，日志聚合和使用 Spring Cloud Sleuth 和 Zipkin 进行追踪。
- 第 10 章是本书最重要的部分。您将本书中构建的服务，部署到 Amazon 弹性容器服务 (ECS) 中。我们还将讨论如何使用自动化工具，如 Travis CI，实现微服务的自动化构建和部署。
- 附录 A 介绍了如何设置桌面开发环境，以便运行本书中的所有代码示例。本附录包括如何在本地构建应用程序和如果你想在本机运行代码示例，如何启动本地的 Docker 容器。

- 附录 B 是 OAuth2 补充材料。OAuth2 是一个非常灵活的认证模型，本章简要概述了 OAuth2 被用于保护应用程序和对应微服务的各种方式。

关于代码

本书包含了每一章的示例代码。所有的代码示例都可用。在我的 GitHub 仓库，每一章都有自己的独立库。在 https://github.com/carnellj/spmia_overview 概述页面，您可以找到链接到每一章代码库的链接。您可以在 Manning 出版社的网站上下载书中应用程序的完整源代码，地址是 www.manning.com/books/spring-microservices-in-action。

在这本书中的所有代码是运行在 java 8 环境下，使用 Maven 作为主要的构建工具。请参阅本书附录 A，详细了解编译和运行代码示例所需的软件工具。

我写这本书时所遵循的原则是，每个章节中的代码示例应该独立于其他章节中的代码示例。因此，我在每章中创建的每个服务都构建一个对应的 Docker 镜像。当代码被前面的章节使用，它包括源代码和一个已构建的 Docker 镜像。我们使用 Docker compose 构建 Docker 镜像来保证在每章中拥有一个可以重复运行的环境。

本书包含很多源代码示例，它们使用编号列表和正常文本行显示。在这两种情况下，源代码是以固定宽度字体格式化的，以便将其与普通文本分隔开来。有时代码也会加粗突出显示区别于本章前面步骤的代码，如新功能添加到现有代码行时。

在很多情况下，源代码已被格式化；我们添加了换行符和重做缩进来调整书中可用的页面空间。在少数情况下，列表中包含了行延续标记(↵)。此外，在文本中描述代码时，源代码中的注释常常被从清单中删除。代码注释伴随许多的列表，突出重要概念。

作者在线

购买本书的读者还能免费访问 Manning 出版社的私有 Web 论坛，在那里你能就本书发表评论，询问技术问题，向作者以及其他用户寻求帮助。如需访问并订阅该论坛，请打开

浏览器访问 www.manning.com/books/spring-microservices-in-action。该页面提供了详细的信息，告诉你在注册后如何访问论坛，论坛里都能提供哪些帮助，以及论坛的管理规则。

Manning 向读者承诺，为读者与读者之间以及读者与作者之间的沟通建立桥梁。但 Manning 并不保证作者在论坛中的参与程度，他们在论坛上投入多少精力是全凭自愿的（并且是无偿的）。我们强烈建议你向作者问些有挑战性的问题，让他有兴趣留在论坛里。