

Broadview
www.broadview.com.cn

[PACKT] open source
PUBLISHING community experience freedom

Mastering Microservices with Java

Java微服务

掌握在生产环境下轻松实现微服务的艺术

[美] Sourabh Sharma 著
卢涛 译



中国工信出版集团



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

Mastering Microservices with Java

Java 微服务



[美] Sourabh Sharma 著
卢涛 译

电子工业出版社
Publishing House of Electronics Industry
北京·BEIJING

内 容 简 介

微服务是利用云平台开发企业应用程序的最新技术，它是小型、轻量和过程驱动的组件。微服务适合设计可扩展、易于维护的应用程序。它可以使开发更容易，还能使资源得到最佳利用。本书帮助你用 Java 构建供企业使用的微服务架构，内容包括微服务核心概念和框架、大型软件项目的高层次设计、开发环境设置和前期配置、对微服务架构持续集成的部署、实现微服务的安全性、有效地执行测试、微服务设计的最佳做法和一般原则，以及如何检测和调试问题。

本书适合想要了解微服务架构，以及想要深入了解如何有效地实施企业级微服务的 Java 开发人员。

Copyright © 2016 Packt Publishing. First published in the English language under the title 'Mastering Microservices with Java'.

本书简体中文版专有出版权由 Packt Publishing 授予电子工业出版社。未经许可，不得以任何方式复制或抄袭本书的任何部分。专有出版权受法律保护。

版权贸易合同登记号 图字：01-2016-7945

图书在版编目（CIP）数据

Java 微服务 / (美) 沙鲁巴·夏尔马 (Sourabh Sharma) 著；卢涛译. —北京：电子工业出版社，2017.1

书名原文：Mastering Microservices with Java

ISBN 978-7-121-30493-4

I. ①J… II. ①沙… ②卢… III. ①JAVA 语言—程序设计 IV. ①TP312.8

中国版本图书馆 CIP 数据核字（2016）第 288000 号

责任编辑：张春雨

印 刷：北京中新伟业印刷有限公司

装 订：北京中新伟业印刷有限公司

出版发行：电子工业出版社

北京市海淀区万寿路 173 信箱

邮编：100036

开 本：787×980 1/16 印张：15.5

字数：322.4 千字

版 次：2017 年 1 月第 1 版

印 次：2017 年 1 月第 1 次印刷

定 价：69.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888，88258888。

质量投诉请发邮件至 zltts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式：（010）51260888-819，faq@phei.com.cn。

作者简介

Sourabh Sharmahas 具有十年以上的产品/应用程序开发经验。他的专长是开发、部署和测试多层 web 应用程序。他喜欢解决复杂的问题，并寻找最佳的解决方案。

在他的职业生涯中，他已成功地为财富 500 强的客户开发和交付了各种独立应用程序和云应用程序，给他们带来很多收益。

Sourabh 还为他的总部设在美国的顶尖企业产品公司发起并开发了一种基于微服务的产品。他在大学时代，即 20 世纪 90 年代后期，开始编写 Java 程序，而且至今仍然热爱这项工作。

审阅者简介

Guido Grazioli 担任过种类繁多的业务应用程序的开发人员、软件架构师和系统集成人员,他的工作跨越多个领域。他是一位复合型软件工程师,对 Java 平台和工具,以及 Linux 系统管理都有深入了解;对 SOA、EIP、持续集成和交付,以及在云环境中的服务业务流程尤其感兴趣。

前言

微服务 (Microservices) 架构是软件架构风格的一种。随着云平台的采用，企业应用程序的开发从整体应用程序转移到小型、轻量和过程驱动的组件，这种组件称为微服务。顾名思义，微服务是指小型服务。它们是设计可扩展、易于维护的应用程序的下一个重大事件。它不但使应用程序开发起来更容易，而且还提供了极大的灵活性来以最佳方式利用各种资源。

本书是帮助你构建供企业使用的微服务实现的实践指南。它还解释了领域驱动设计及其在微服务中的采用。它讲述了怎样构建更小型、更轻量、更快速的服务，同时确保其可以很方便地在生产环境中实施。它也讲述了企业应用程序开发从设计与开发，到部署、测试和实现安全性的完整生命周期。

本书包含的内容

第 1 章，一种解决方法，涉及大型软件项目的高层次设计，在生产环境中所面临的共同问题和解决问题的方法。

第 2 章，设置开发环境，讲述了如何设置开发环境，包括 IDE 和其他开发工具，以及不同的库。本章涉及创建基本项目到设置 spring 引导配置，以建立和发展第一个微服务。

第 3 章，领域驱动设计，通过引用一个示例项目为其余的章节设定基调。它使用此示例项目来驱动服务或应用程序的不同功能和领域组合来解释领域驱动设计。

第 4 章，实现微服务，讲述示例项目从设计到实现的过程。本章不仅涉及编码，还涉及微服务的不同方面——构建、单元测试和包装。在本章末尾，将完成一个可用于部署和使用的示例微服务项目。

第 5 章，部署和测试，讲述了如何采用不同的形式，包括独立部署和使用诸如 Docker

的容器来部署微服务。本章还将演示如何用 Docker 把我们的示例项目部署到诸如 AWS 的云服务上面。你还将掌握使用 REST Java 客户端和其他工具来测试微服务的知识。

第 6 章，实现微服务的安全性，解释如何利用身份验证和授权来保证微服务的安全。身份验证将使用基本身份验证和身份验证令牌来讲述。同样，授权将使用 Spring Security 来解释。本章还将解释常见的安全问题及对策。

第 7 章，利用微服务 Web 应用程序来使用服务，解释了如何利用 Knockout、Require 和 Bootstrap JS 库开发 web 应用程序（UI），构建使用微服务来显示数据的 web 应用程序的原型和一个小型实用程序项目（示例项目）的流程。

第 8 章，最佳做法和一般原则，讲述微服务设计的最佳做法和一般原则。本章还提供了有关使用行业做法进行微服务开发的详细信息和范例。本章还包含微服务实现会产生的错误，以及如何才能避免这类问题的几个例子。

第 9 章，故障排除指南，解释了在微服务及其解决方案的开发过程中会遇到的常见问题。这将帮助你顺利地掌握本书内容，并使学习过程轻松。

学习本书需要具备的条件

为了学习本书，可以使用至少具备 2GB 内存的安装了任何操作系统（Linux、Windows 或 Mac）的计算机；还需要 NetBeans with Java、Maven、Spring Boot、Spring Cloud、Eureka Server、Docker 和 CI/CD 的应用程序。对于 Docker 容器，可能需要一个单独的虚拟机或一个云主机，最好拥有 16GB 或更大的内存。

本书的受众

本书面向熟悉微服务架构，并对核心要素和微服务应用程序有一个合理的知识水平和理解，但现在想要深入了解如何有效地实施企业级微服务的 Java 开发人员。

版式约定

你会发现，本书采用了大量的文本样式，用以区分不同类型的信息。下面是这些样式和解释它们的含义的一些例子。

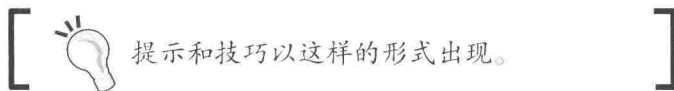
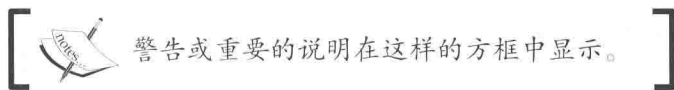
正文中的代码、数据库表名称、文件夹名称、文件名、文件扩展名、路径名、虚拟的 URL、用户输入和 Twitter 句柄如下所示：“可以使用下面的实现创建 Table 实体，并且可以根据自己的需要添加属性”。

代码块的设置，如下所示：

```
public class Table extends BaseEntity<BigInteger> {  
  
    private int capacity;  
  
    public Table(String name, BigInteger id, int capacity) {  
        super(id, name);  
        this.capacity = capacity;  
    }  
}
```

任何命令行输入或输出采用的格式如下：

```
docker push localhost:5000/sourabhh/restaurant-service:PACKT-SNAPSHOT  
docker-compose pull
```



下载示例代码

从 <http://www.broadview.com.cn> 下载所有已购买的博文视点书籍的示例代码文件。

勘误表

虽然我们已经尽力谨慎地确保内容的准确性，但错误仍然存在。如果你发现了书中的错误，包括正文和代码中的错误，请告诉我们，我们会非常感激。这样，你不仅帮助了其他读者，也帮助我们改进后续的出版。如发现任何勘误，可以在博文视点网站相应图书的

页面提交勘误信息。一旦你找到的错误被证实，你提交的信息就会被接受，我们的网站也会发布这些勘误信息。你可以随时浏览图书页面，查看已发布的勘误信息。

参与本书翻译的人员有卢涛、李颖、卢林、陈克非、李洪秋、张慧珍、李又及、卢晓瑶、李阳、陈克翠、刘雯、汤有四。

目录

前言	XV
1 一种解决方法.....	1
微服务的演变.....	2
整体式架构概述.....	3
整体式架构的局限性与它的微服务解决方案的对比.....	3
一维的可扩展性.....	6
在出故障时回滚版本.....	7
采用新技术时的问题.....	7
与敏捷实践的契合.....	8
减轻开发工作量——可以做得更好.....	9
微服务的构建管道.....	10
使用诸如 Docker 的容器部署.....	11
容器.....	11
Docker.....	12
Docker 的架构.....	13
Docker 容器.....	14
部署.....	14
小结.....	14
2 设置开发环境.....	17
Spring Boot 配置.....	18
Spring Boot 概述.....	18
把 Spring Boot 添加至 REST 示例.....	19

添加一个嵌入式 Jetty 服务器.....	21
示例 REST 程序.....	22
编写 REST 控制器类.....	24
@RestController.....	25
@RequestMapping.....	25
@RequestParam.....	25
@PathVariable.....	26
制作一个示例 REST 可执行应用程序.....	29
设置应用程序构建.....	30
运行 Maven 工具.....	30
用 Java 命令执行.....	31
使用 Postman Chrome 扩展测试 REST API.....	31
更多的正向测试场景.....	34
反向的测试场景.....	35
NetBeans IDE 安装和设置.....	37
参考资料.....	42
小结.....	42
3 领域驱动设计.....	43
领域驱动设计基本原理.....	44
组成部分.....	45
普遍存在的语言.....	45
多层架构.....	45
表示层.....	46
应用程序层.....	46
领域层.....	46
基础架构层.....	47
领域驱动设计的工件.....	47
实体.....	47

值对象	48
服务	49
聚合	50
存储库	52
工厂	53
模块	54
战略设计和原则	55
有界上下文	55
持续集成	56
上下文映射	57
共享内核模式	58
客户和供应商模式	58
顺从者模式	59
防腐层	59
独立方法	59
开放主机服务	60
精馏	60
示例领域服务	60
实体的实现	61
存储库的实现	63
服务的实现	66
小结	67
4 实现微服务	69
OTRS 概述	70
开发和实现微服务	71
餐馆微服务	72
控制器类	73
服务类	76

存储库类.....	79
实体类	82
预订和用户服务	85
注册和发现服务（Eureka 服务）	85
执行.....	87
测试.....	87
参考资料.....	92
小结.....	92
5 部署和测试	93
使用 Netflix OSS 的微服务架构概述	93
负载均衡.....	95
客户端的负载均衡	95
服务器端的负载均衡	98
电路断路器与监控.....	102
使用 Hystrix 的回退方法	102
监控.....	103
设置 Hystrix 仪表盘	105
设置 Turbine	107
使用容器部署微服务	109
安装和配置.....	109
具有 4 GB 内存的 Docker 机器.....	110
使用 Maven 构建 Docker 映像	110
使用 Maven 运行 Docker	114
使用 Docker 执行集成测试	115
把映像推送到注册表.....	118
管理 Docker 容器.....	119
参考资料.....	121
小结.....	121

6 实现微服务的安全性	123
启用安全套接字层	123
身份验证和授权	127
OAuth 2.0	127
OAuth 的用法	128
OAuth 2.0 规范——简明详细信息	128
OAuth 2.0 角色	129
OAuth 2.0 客户端注册	131
OAuth 2.0 协议端点	135
OAuth 2.0 授权类型	137
使用 Spring Security 的 OAuth 实现	144
授权码许可	150
隐式许可	153
资源所有者密码凭据许可	154
客户端凭据许可	155
参考资料	155
小结	156
7 利用微服务 Web 应用程序来使用服务	157
AngularJS 框架概述	157
MVC	158
MVVM	158
模块	158
提供程序和服务	160
作用域	161
控制器	161
过滤器	161
指令	162
UI-Router	162

OTRS 功能的开发	163
主页/餐馆列表页	163
index.html	164
app.js	169
restaurants.js	172
restaurants.html	179
搜索餐馆	180
餐馆详细信息与预订选项	181
restaurant.html	181
登录页面	183
login.html	184
login.js	185
预订确认	186
设置 web 应用程序	187
小结	201
8 最佳做法和一般原则	203
概述和心态	203
最佳做法和原则	205
Nanoservice (不推荐)、规模和整体性	205
持续集成和部署	206
系统/端到端测试自动化	207
自我监控和记录	207
每个微服务都使用独立的数据存储区	209
事务边界	210
微服务框架和工具	210
Netflix 开放源码软件 (OSS)	210
构建——Nebula	211
部署和交付——Spinnaker 与 Aminator	211

服务注册和发现——Eureka.....	211
服务沟通——Ribbon.....	212
电路断路器——Hystrix.....	212
边缘（代理）服务器——Zuul.....	212
业务监控——Atlas.....	213
可靠性监控服务——Simian Army.....	213
AWS 资源监控——Edda.....	214
主机性能监控——Vector.....	215
分布式配置管理——Archaius.....	215
Apache Mesos 调度器——Fenzo.....	215
成本和云利用率——Ice.....	216
其他安全工具——Scumblr 和 FIDO.....	216
参考资料.....	217
小结.....	218
9 故障排除指南.....	219
日志记录和 ELK 环境.....	219
简要概述.....	221
Elasticsearch.....	221
Logstash.....	221
Kibana.....	222
ELK 环境安装.....	222
安装 Elasticsearch.....	223
安装 Logstash.....	224
安装 Kibana.....	225
服务调用关联 ID 的使用.....	226
让我们看看怎样解决这个问题.....	226
依赖项和版本.....	227
循环依赖关系及其影响.....	227

设计系统时需要分析它	227
维护不同版本	227
让我们了解更多	228
参考资料	228
小结	228