



Spring Cloud Microservice Development

Spring Cloud微服务

入门、实战与进阶

尹吉欢 著

从技术原理、工程实践、进阶提升3个维度详解Spring Cloud微服务的架构与开发

指导零基础读者快速入门并掌握工程实践能力，最终进阶为Spring Cloud微服务技术达人



机械工业出版社
China Machine Press

云计算与虚拟化技术丛书

Spring Cloud Microservice Development

Spring Cloud 微服务

入门、实战与进阶

尹吉欢 著



机械工业出版社
China Machine Press

图书在版编目 (CIP) 数据

Spring Cloud 微服务：入门、实战与进阶 / 尹吉欢著. —北京：机械工业出版社，2019.5
(云计算与虚拟化技术丛书)

ISBN 978-7-111-62731-9

I.S… II.尹… III.互连网络-网络服务器 IV. TP368.5

中国版本图书馆 CIP 数据核字 (2019) 第 091644 号

Spring Cloud 微服务：入门、实战与进阶

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：张锡鹏

责任校对：殷虹

印刷：北京市荣盛彩色印刷有限公司

版次：2019 年 6 月第 1 版第 1 次印刷

开本：186mm×240mm 1/16

印张：26

书号：ISBN 978-7-111-62731-9

定价：89.00 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

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 Cloud 是开发分布式系统的“全家桶”，它实现了很多分布式应用中的“套路”。目前 Spring Cloud 正被越来越多的企业用于生产。本书知识体系非常全面，涵盖了微服务、Spring Cloud、分布式事务、缓存、存储等话题，让读者能够快速上手构建自己的分布式系统，值得一读。

——《Spring Cloud 与 Docker 微服务架构实战》作者 周立

Spring Cloud 对于中小型互联网公司来说是一种福音，因为这类公司往往没有实力或者足够的资金投入去开发自己的分布式系统基础设施，使用 Spring Cloud 一站式解决方案能在企业从容应对业务发展的同时大大减少开发成本。本书详细介绍了 Spring Cloud 各模块的使用，并且对 Spring Cloud 没有涉及的安全认证、服务限流、一致性事务等解决方案进行了详细讲解，我相信开发者在读完本书后会快速掌握 Spring Cloud 的相关知识。

——海科融通研发中心副总监 张强（纯洁的微笑）

架构的演进永无止境。2000 年 WebService 出现后，SOA 被誉为下一代 Web 服务的基础架构，已经成为计算机信息领域的一个新的发展方向。SOA 经过十几年的发展，逐渐趋于成熟。微服务架构这一术语在 2012 年横空出世，用于描述一种特定的软件设计方法，即以若干组可独立部署的服务的方式进行软件应用系统的设计。每个小型服务都运行在自己的进程中。这些服务围绕业务、功能进行构建，并通过全自动的部署机制来进行独立部署。这些微服务可以使用不同的语言来编写，并且可以使用不同的数据存储技术。

容器及其相关技术的快速发展加速了微服务架构的成熟和普及，同时也涌现出一批新的微服务落地实施的解决方案，Spring Cloud 便是其中的佼佼者。本书内容全面详尽，文字浅显易懂。如果你想快速学习和实战 Spring Cloud，本书是一个不错的选择。

——红瓦科技 CTO 刘夕波

本书行文流畅，由浅入深。不仅介绍了分布式开发的市场布局，更是对 Spring Cloud 进行了一次全方位的实践与对话。本书结合作者多年的开发和项目管理经验，为读者徐徐展开了一幅优雅的技术画卷。

——中科韬睿技术总监、《轻松搞定 Extjs》《Android 自定义组件开发详解》作者 李赞红

本书很好地将微服务中的重要话题 Spring Cloud 与其他开源组件融合起来，这其中也包括了 Elastic-Job 和 Sharding-JDBC。作为这两款开源产品的作者之一，我很高兴地看到它们已逐渐走进技术人员的视野中。希望读者阅读本书后，可以快速熟悉当今微服务技术栈的整体结构，并提升技术选型的能力。

——京东金融数据研发负责人 张亮

微服务日趋流行，随着部署数量的不断增长，如何管理如此众多的微服务就成为一个亟待解决的问题。本书根植于项目中的实践经验，很多代码都是作者在实际项目中的提炼，并且以通俗易懂的方式描述了 Spring Cloud 管理微服务的方法。读者阅读本书后亲自实践，定能获得累累的收获。

——房价网 CTO 虞继恩

本书以作者真正的线上实战示例为基础，详细介绍了其对 Spring Cloud 的理解和应用。Spring Cloud 虽然被称为微服务的“全家桶”方案，但在实际使用时，我们依然会面临各种问题需要去解决。本书的可贵之处在于，作者在讲解 Spring Cloud 自身内容的同时也深入分享了一些实战中的问题、解决思路以及扩展内容，这些都是非常珍贵的实战经验，所以我推荐正在使用 Spring Cloud 的朋友们阅读本书。

——spring4all 社区发起人、《Spring Cloud 微服务实战》作者 翟永超

为什么要写这本书

在互联网时代，互联网产品的最大特点就是需要快速发布新功能，支持高并发和大数据。传统的架构已经慢慢不能支撑互联网业务的发展，这时微服务架构便顺势而出。

最开始，国内很多公司都是基于阿里开源的 Dubbo 框架来构建微服务的，由于阿里内部的原因，Dubbo 已经几年没进行维护了，不过在 2018 年，阿里宣布重新开始维护。反观 Spring Cloud，其在国外发展得很好，但在国内，由于 Dubbo 的存在导致 Spring Cloud 鲜为人知。不过从 2017 年开始，Spring Cloud 在国内的普及度逐渐变高，很多中小型互联网公司都开始拥抱 Spring Cloud。

Spring Cloud 提供一整套微服务的解决方案，基于 Spring Boot 可实现快速集成，且开发效率很高，堪称中小型互联网公司微服务开发的福音。而且 Spring Cloud 发布新功能的频率非常高，目前仅大版本就有很多个，同时还有庞大的社区支持，照这样的发展势头，我相信未来几年国内互联网公司的公布式系统开发一定是 Spring Cloud 的天下。

我一直在使用 Spring Boot、Spring Data 等一系列框架来进行开发，作为一名 Spring Cloud 的忠实粉丝，自然希望能够有更多开发者参与进来，于是自己坚持写 Spring Cloud 相关的文章，并且将文章涉及的代码整理好放在 GitHub 上面进行分享。在这个过程中我得到了很多开发者的关注，他们向我咨询一些微服务方面的问题，我也会在研究和解决了一些问题后，通过文章分享给各位开发者。在有幸结识了华章的杨老师后，我决定将这些文章整理成书，目的是想推广 Spring Cloud 在国内的使用和发展，并分享自己在微服务领域的一些小经验。

读者对象

- Java 开发工程师
- Spring Cloud 用户和爱好者
- 微服务爱好者

本书的读者对象主要是 Java 开发人员，特别是工作 1 ~ 3 年的开发人员，这个阶段开发人员资历尚浅，需要一些实用的技术和经验来提升自己，Spring Cloud 正是一门符合提升要求的技术。因为它现在正处于快速发展的阶段，越来越多的企业也开始使用 Spring Cloud。相信在不久的将来，熟练掌握 Spring Cloud 将会成为 Java 开发人员面试的门槛。

本书内容

本书主打的是与微服务相关的实战体系。第一部分是准备篇，可以帮助各位读者了解微服务以及 Spring Cloud 的概念。第二部分是基础篇，会对 Spring Cloud 中常用的模块进行详细讲解。第三部分是实战篇，开始实战性质的内容讲解，包括选择配置中心、自研配置中心、分布式跟踪、微服务安全认证、Spring Boot Admin 管理微服务、快速生成 API 文档等实用内容。

最后一部分是高级篇，也是难度比较大的一部分，主要内容如下：

- 对 Zuul 进行扩展，即对认证、限流、降级、灰度发布等内容进行讲解。
- 讲解缓存框架的使用，解决缓存穿透、缓存雪崩等问题。
- 数据存储的选型，比如对 MySQL、MongoDB、ElasticSearch 的使用进行讲解。
- 分布式事务的解决方案，重点是利用消息队列开发可靠性消息服务来实现数据的最终一致性。
- 讲解分布式任务调度框架 Elastic-Job。
- 讲解分库分表的解决方案 Sharding-JDBC。

勘误和支持

由于水平有限，书中难免会出现一些不准确的地方，恳请读者批评指正。为此，特贴出本书源码地址 <https://github.com/yinjihuan/spring-cloud>。如果你遇到任何问题或者有其他宝贵意见，欢迎发送邮件至邮箱 jihuan900@126.com，期待能够得到你们的真挚反馈。

致谢

首先要感谢 Spring Cloud 的各位开发人员，感谢你们开发出这样一个好用的框架。

感谢机械工业出版社华章公司的杨福川老师，是你在这半年多的时间中始终支持我的写作，正因为有你的鼓励和帮助，我才能顺利完成全部书稿。

感谢机械工业出版社华章公司的张锡鹏老师，是你在本书的审稿过程中给了我很多实用的建议，让我学习到了很多写作方面的技巧。

最后感谢家人的支持和理解，让我能够把全部精力投入到本书的写作中。谨以此书献给我最亲爱的家人，以及众多热爱 Spring Cloud 的朋友们！

目 录 Contents

赞誉
前言

第一部分 准备篇

第1章 Spring Cloud 与微服务概述 ... 2

- 1.1 传统的单体应用 2
 - 1.1.1 改进单体应用的架构 2
 - 1.1.2 向微服务靠拢 3
- 1.2 什么是微服务 4
 - 1.2.1 使用微服务架构的优势和劣势 4
 - 1.2.2 重构前的准备工作 5
- 1.3 什么是 Spring Cloud 5
 - 1.3.1 Spring Cloud 模块介绍 6
 - 1.3.2 Spring Cloud 版本介绍 6
- 1.4 本章小结 7

第2章 实战前的准备工作 8

- 2.1 开发环境的准备 8
- 2.2 Spring Boot 入门 9

- 2.2.1 Spring Boot 简介 9
- 2.2.2 搭建 Spring Boot 项目 9
- 2.2.3 编写第一个 REST 接口 11
- 2.2.4 读取配置文件 11
- 2.2.5 profiles 多环境配置 13
- 2.2.6 热部署 13
- 2.2.7 actuator 监控 15
- 2.2.8 自定义 actuator 端点 17
- 2.2.9 统一异常处理 18
- 2.2.10 异步执行 20
- 2.2.11 随机端口 22
- 2.2.12 编译打包 24
- 2.3 Spring Boot Starter 自定义 25
 - 2.3.1 Spring Boot Starter 项目创建 25
 - 2.3.2 自动创建客户端 26
 - 2.3.3 使用 Starter 27
 - 2.3.4 使用注解开启 Starter 自动构建 27
 - 2.3.5 使用配置开启 Starter 自动构建 28
 - 2.3.6 配置 Starter 内容提示 29
- 2.4 本章小结 29

第二部分 基础篇

第3章 Eureka 注册中心	32
3.1 Eureka	32
3.2 使用 Eureka 编写注册中心 服务	33
3.3 编写服务提供者	35
3.3.1 创建项目注册到 Eureka	35
3.3.2 编写提供接口	36
3.4 编写服务消费者	37
3.4.1 直接调用接口	37
3.4.2 通过 Eureka 来消费接口	38
3.5 开启 Eureka 认证	38
3.6 Eureka 高可用搭建	39
3.6.1 高可用原理	39
3.6.2 搭建步骤	40
3.7 常用配置讲解	41
3.7.1 关闭自我保护	41
3.7.2 自定义 Eureka 的 InstanceID	41
3.7.3 自定义实例跳转链接	42
3.7.4 快速移除已经失效的服务 信息	43
3.8 扩展使用	44
3.8.1 Eureka REST API	44
3.8.2 元数据使用	46
3.8.3 EurekaClient 使用	47
3.8.4 健康检查	49
3.8.5 服务上下线监控	50
3.9 本章小结	51

第4章 客户端负载均衡 Ribbon	52
4.1 Ribbon	52
4.1.1 Ribbon 模块	52
4.1.2 Ribbon 使用	53
4.2 RestTemplate 结合 Ribbon 使用	54
4.2.1 使用 RestTemplate 与整合 Ribbon	54
4.2.2 RestTemplate 负载均衡 示例	57
4.2.3 @LoadBalanced 注解原理	58
4.2.4 Ribbon API 使用	62
4.2.5 Ribbon 饥饿加载	63
4.3 负载均衡策略介绍	64
4.4 自定义负载策略	65
4.5 配置详解	66
4.5.1 常用配置	66
4.5.2 代码配置 Ribbon	67
4.5.3 配置文件方式配置 Ribbon	67
4.6 重试机制	68
4.7 本章小结	69

第5章 声明式REST客户端Feign	70
5.1 使用 Feign 调用服务接口	70
5.1.1 在 Spring Cloud 中集成 Feign	71
5.1.2 使用 Feign 调用接口	71
5.2 自定义 Feign 的配置	72
5.2.1 日志配置	72
5.2.2 契约配置	73
5.2.3 Basic 认证配置	74

5.2.4	超时时间配置	75	6.4	整合 Dashboard 查看监控数据	98
5.2.5	客户端组件配置	75	6.5	Turbine 聚合集群数据	100
5.2.6	GZIP 压缩配置	76	6.5.1	Turbine 使用	100
5.2.7	编码器解码器配置	77	6.5.2	context-path 导致监控 失败	101
5.2.8	使用配置自定义 Feign 的 配置	78	6.6	本章小结	102
5.2.9	继承特性	78	第7章	API网关	103
5.2.10	多参数请求构造	80	7.1	Zuul 简介	103
5.3	脱离 Spring Cloud 使用 Feign	80	7.2	使用 Zuul 构建微服务网关	104
5.3.1	原生注解方式	81	7.2.1	简单使用	104
5.3.2	构建 Feign 对象	82	7.2.2	集成 Eureka	105
5.3.3	其他配置	83	7.3	Zuul 路由配置	105
5.4	本章小结	83	7.4	Zuul 过滤器讲解	106
第6章	Hystrix 服务容错处理	84	7.4.1	过滤器类型	106
6.1	Hystrix	84	7.4.2	请求生命周期	107
6.1.1	Hystrix 的简单使用	84	7.4.3	使用过滤器	108
6.1.2	回退支持	85	7.4.4	过滤器禁用	109
6.1.3	信号量策略配置	86	7.4.5	过滤器中传递数据	110
6.1.4	线程隔离策略配置	86	7.4.6	过滤器拦截请求	111
6.1.5	结果缓存	87	7.4.7	过滤器中异常处理	113
6.1.6	缓存清除	88	7.5	Zuul 容错和回退	115
6.1.7	合并请求	89	7.5.1	容错机制	115
6.2	在 Spring Cloud 中使用 Hystrix	91	7.5.2	回退机制	116
6.2.1	简单使用	91	7.6	Zuul 使用小经验	118
6.2.2	配置详解	92	7.6.1	/routes 端点	118
6.2.3	Feign 整合 Hystrix 服务 容错	95	7.6.2	/filters 端点	118
6.2.4	Feign 中禁用 Hystrix	97	7.6.3	文件上传	119
6.3	Hystrix 监控	97	7.6.4	请求响应信息输出	121
			7.6.5	Zuul 自带的 Debug 功能	124
			7.7	Zuul 高可用	126
			7.8	本章小结	127

第三部分 实战篇

第8章 API 网关之Spring Cloud

Gateway 130

8.1 Spring Cloud Gateway 介绍 130

8.2 Spring Cloud Gateway 工作 原理 131

8.3 Spring Cloud Gateway 快速 上手 131

8.3.1 创建 Gateway 项目 131

8.3.2 路由转发示例 132

8.3.3 整合 Eureka 路由 133

8.3.4 整合 Eureka 的默认路由 133

8.4 Spring Cloud Gateway 路由 断言工厂 134

8.4.1 路由断言工厂使用 134

8.4.2 自定义路由断言工厂 136

8.5 Spring Cloud Gateway 过滤器 工厂 137

8.5.1 Spring Cloud Gateway 过滤 器工厂使用 137

8.5.2 自定义 Spring Cloud Gateway 过滤器工厂 138

8.6 全局过滤器 140

8.7 实战案例 143

8.7.1 限流实战 143

8.7.2 熔断回退实战 145

8.7.3 跨域实战 145

8.7.4 统一异常处理 147

8.7.5 重试机制 150

8.8 本章小结 151

第9章 自研分布式配置管理 152

9.1 自研配置管理框架 Smconf 简介 152

9.2 Smconf 工作原理 153

9.3 Smconf 部署 154

9.3.1 Mongodb 安装 154

9.3.2 Zookeeper 安装 155

9.3.3 Smconf Server 部署 156

9.4 项目中集成 Smconf 157

9.4.1 集成 Smconf 157

9.4.2 使用 Smconf 158

9.4.3 配置更新回调 159

9.5 Smconf 详细使用 160

9.5.1 源码编译问题 160

9.5.2 后台账号管理 160

9.5.3 REST API 161

9.6 Smconf 源码解析 163

9.6.1 Client 启动 163

9.6.2 启动加载配置 165

9.6.3 配置修改推送原理 166

9.7 本章小结 167

第10章 分布式配置中心Apollo 168

10.1 Apollo 简介 168

10.2 Apollo 的核心功能点 168

10.3 Apollo 核心概念 170

10.4 Apollo 本地部署 171

10.5 Apollo Portal 管理后台使用 172

10.6 Java 中使用 Apollo 174

10.6.1 普通 Java 项目中使用 174

10.6.2 Spring Boot 中使用 177

10.7	Apollo 的架构设计	179	12.2.1	表结构	210
10.7.1	Apollo 架构设计介绍	179	12.2.2	JWT 工具类封装	210
10.7.2	Apollo 服务端设计	181	12.2.3	认证接口	212
10.7.3	Apollo 客户端设计	188	12.3	服务提供方进行调用认证	212
10.7.4	Apollo 高可用设计	195	12.4	服务消费方申请 Token	214
10.8	本章小结	196	12.5	Feign 调用前统一申请 Token 传递到调用的服务中	216
第11章	Sleuth 服务跟踪	197	12.6	RestTemplate 调用前统一 申请 Token 传递到调用的 服务中	217
11.1	Spring Cloud 集成 Sleuth	197	12.7	Zuul 中传递 Token 到路由的 服务中	218
11.2	整合 Logstash	198	12.8	本章小结	219
11.2.1	ELK 简介	198	第13章	Spring Boot Admin	220
11.2.2	输出 JSON 格式日志	198	13.1	Spring Boot Admin 的使用 方法	220
11.3	整合 Zipkin	200	13.1.1	创建 Spring Boot Admin 项目	220
11.3.1	Zipkin 数据收集服务	200	13.1.2	将服务注册到 Spring Boot Admin	221
11.3.2	项目集成 Zipkin 发送调用 链数据	201	13.1.3	监控内容介绍	223
11.3.3	抽样采集数据	203	13.1.4	如何在 Admin 中查看各个 服务的日志	225
11.3.4	异步任务线程池定义	203	13.2	开启认证	226
11.3.5	TracingFilter	204	13.3	集成 Eureka	227
11.3.6	监控本地方法	205	13.4	监控服务	228
11.3.7	过滤不想跟踪的请求	206	13.4.1	邮件警报	228
11.3.8	用 RabbitMq 代替 Http 发送 调用链数据	206	13.4.2	自定义钉钉警报	229
11.3.9	用 Elasticsearch 存储调用 链数据	207	13.5	本章小结	232
11.4	本章小结	208			
第12章	微服务之间调用的安全 认证	209			
12.1	什么是 JWT	209			
12.2	创建统一的认证服务	210			

第14章 服务的API文档管理 233

14.1 Swagger 简介 233

14.2 集成 Swagger 管理 API
文档 234

14.2.1 项目中集成 Swagger 234

14.2.2 使用 Swagger 生成文档 234

14.2.3 在线测试接口 235

14.3 Swagger 注解 236

14.4 Eureka 控制台快速查看
Swagger 文档 240

14.5 请求认证 240

14.6 Zuul 中聚合多个服务
Swagger 241

14.7 本章小结 242

第四部分 高级篇**第15章 API 网关扩展** 244

15.1 用户认证 244

15.1.1 动态管理不需要拦截的
API 请求 244

15.1.2 创建认证的用户服务 246

15.1.3 路由之前的认证 247

15.1.4 向下游微服务中传递认证
之后的用户信息 24815.1.5 内部服务间的用户信息
传递 248

15.2 服务限流 250

15.2.1 限流算法 250

15.2.2 单节点限流 251

15.2.3 集群限流 255

15.2.4 具体服务限流 258

15.2.5 具体接口限流 258

15.3 服务降级 262

15.4 灰度发布 264

15.4.1 原理讲解 264

15.4.2 根据用户做灰度发布 265

15.4.3 根据 IP 做灰度发布 268

15.5 本章小结 268

第16章 微服务之缓存 269

16.1 Guava Cache 本地缓存 269

16.1.1 Guava Cache 简介 269

16.1.2 代码示例 270

16.1.3 回收策略 270

16.2 Redis 缓存 271

16.2.1 用 Redistemplate 操作
Redis 27116.2.2 用 Repository 操作
Redis 272

16.2.3 Spring Cache 缓存数据 274

16.2.4 缓存异常处理 278

16.2.5 自定义缓存工具类 279

16.3 防止缓存穿透方案 282

16.3.1 什么是缓存穿透 282

16.3.2 缓存穿透的危害 282

16.3.3 解决方案 282

16.3.4 布隆过滤器介绍 283

16.3.5 代码示例 283

16.4 防止缓存雪崩方案 284

16.4.1 什么是缓存雪崩 284

16.4.2	缓存雪崩的危害	284	17.4.1	集成 Spring Data Elasticsearch	312
16.4.3	解决方案	284	17.4.2	Repository 示例	312
16.4.4	代码示例	285	17.4.3	ElasticsearchTemplate 示例	315
16.4.5	分布式锁方式	285	17.4.4	索引构建方式	318
16.5	本章小结	286	17.5	本章小结	319
第17章	微服务之存储	287	第18章	微服务之分布式事务解决 方案	320
17.1	存储选型	287	18.1	两阶段型	320
17.2	Mongodb	288	18.2	TCC 补偿型	321
17.2.1	集成 Spring Data Mongodb	288	18.3	最终一致性	321
17.2.2	添加数据操作	288	18.3.1	原理讲解	321
17.2.3	索引使用	290	18.3.2	创建可靠性消息服务	323
17.2.4	修改数据操作	291	18.3.3	消息存储表设计	324
17.2.5	删除数据操作	293	18.3.4	提供服务接口	325
17.2.6	查询数据操作	294	18.3.5	创建消息发送系统	329
17.2.7	GridFS 操作	295	18.3.6	消费消息逻辑	332
17.2.8	用 Repository 方式操作 数据	296	18.3.7	消息管理系统	335
17.2.9	自增 ID 实现	300	18.4	最大努力通知型事务	335
17.2.10	批量更新扩展	303	18.5	本章小结	335
17.3	Mysql	304	第19章	分布式任务调度	336
17.3.1	集成 Spring Jdbc- Template	304	19.1	Elastic-Job	336
17.3.2	JdbcTemplate 代码示例	305	19.1.1	Elastic-Job 介绍	336
17.3.3	封装 JdbcTemplate 操作 Mysql 更简单	305	19.1.2	任务调度目前存在的 问题	336
17.3.4	扩展 JdbcTemplate 使用 方式	306	19.1.3	为什么选择 Elastic-Job	337
17.3.5	常见问题	310	19.2	快速集成	338
17.4	Elasticsearch	312	19.3	任务使用	339

19.3.1	简单任务	339	20.1.2	功能列表	359
19.3.2	数据流任务	340	20.1.3	相关概念	359
19.3.3	脚本任务	340	20.2	快速集成	360
19.4	配置参数讲解	341	20.3	读写分离实战	362
19.4.1	注册中心配置	341	20.3.1	准备数据	362
19.4.2	作业配置	342	20.3.2	配置读写分离	363
19.4.3	dataflow 独有配置	343	20.3.3	验证读从库	363
19.4.4	script 独有配置	343	20.3.4	验证写主库	365
19.5	多节点并行调度	344	20.3.5	Hint 强制路由主库	366
19.5.1	分片概念	344	20.4	分库分表实战	367
19.5.2	任务节点分片策略	344	20.4.1	常用分片算法	367
19.5.3	业务数据分片处理	345	20.4.2	使用分片算法	368
19.6	事件追踪	347	20.4.3	不分库只分表实战	368
19.7	扩展功能	349	20.4.4	既分库又分表实战	372
19.7.1	自定义监听器	349	20.5	分布式主键	375
19.7.2	定义异常处理	349	20.6	本章小结	377
19.8	运维平台	350	第21章	最佳生产实践经验	378
19.8.1	功能列表	350	21.1	开发环境和测试环境共用	
19.8.2	部署运维平台	351		Eureka	378
19.8.3	运维平台使用	351	21.2	Swagger 和 Actuator 访问进行	
19.9	使用经验分享	355		权限控制	379
19.9.1	任务的划分和监控	355	21.3	Spring Boot Admin 监控被保护	
19.9.2	任务的扩展性和节点			的服务	380
	数量	355	21.4	Apollo 配置中心简化版搭建	
19.9.3	任务的重复执行	355		分享	380
19.9.4	overwrite 覆盖问题	356	21.5	Apollo 使用小经验	382
19.9.5	流水式任务	356	21.5.1	公共配置	382
19.10	本章小结	357	21.5.2	账号权限	383
第20章	分库分表解决方案	358	21.5.3	环境配置和项目配置	385
20.1	Sharding-JDBC	358	21.6	Apollo 动态调整日志级别	385
20.1.1	介绍	358			

21.7	Apollo 存储加密	387	21.11	Elastic-Job 的 Spring-Boot- Starter 封装	394
21.8	扩展 Apollo 支持存储加 解密	390	21.12	Spring Boot 中 Mongoddb 多 数据源封装	396
21.9	Apollo 结合 Zuul 实现动态 路由	391	21.13	Zuul 中对 API 进行加解密 ...	398
21.10	Apollo 整合 Archaius	393	21.14	本章小结	400