

Java 微服务实践 - [Spring Cloud](#) 系列

简介

Spring Cloud 为开发人员提供快速构建分布式系统的一些通用模式，其中包括：配置管理、服务发现、服务短路、智能路由、微型网关、控制总线、一次性令牌、全局锁、领导选举、分布式会话和集群状态。分布式系统间的协调导向样板模式，并且使用 Spring Cloud 的开发人员能够快速构建实现这些模式的服务和应用。这些服务和应用也将任何环境下工作良好，无论是开发者的笔记本、还是数据中心裸机或者管控平台。

讲师信息

小马哥，十余年 Java EE 从业经验，架构师、微服务布道师。目前主要负责一线互联网公司微服务技术实施、架构演进、基础设施构建等。重点关注云计算、微服务以及软件架构等领域。通过 SUN Java（SCJP、SCWCD、SCBCD）以及 Oracle OCA 等的认证。

课程特点

Spring Cloud 系列课程致力于以实战的方式覆盖 Spring Cloud 的功能特性，更为重要的是，小马哥希望通过“授人以渔”的方式，不仅让小伙伴们能够认识到技术的演进并非凭空遐想，而是在其特定的场景下“生根发芽”，并且结合自身十余年的学习方法和工作经验，将技术的发展脉络贯穿其中。循序渐进式地引导朋友们，站在哲学的高度，体会 Spring Cloud 的作者设计意图。同时，结合 Spring Cloud 的源码加深理解，最终达到形成系统性的知识和技术体系的目的。

运行环境

Spring Cloud 系列实战运行环境如下：

容器/框架

Spring Cloud Dalston.SR3

Spring Boot 1.5.x

Spring Framework 4.3.x

Java 运行时

Java 8

构建工具

Maven 3.5.0

课程详情

第一节 Spring Cloud Config Client

- 课程内容（1.5 - 2 小时）
 - 技术回顾：回顾提及的 Environment、以及 Spring Boot 配置相关的事件和监听器，如 `ApplicationEnvironmentPreparedEvent` 和 `ConfigFileApplicationListener`，
 - Bootstrap 配置属性：解密 Bootstrap 配置属性与 Spring Framework / Spring Boot 配置架构的关系，介绍如何调整 Bootstrap 配置文件路径、覆盖远程配置属性、自定义 Bootstrap 配置以及自定义 Bootstrap 配置属性源
 - Environment 端点：介绍 `/env` 端点的使用场景，并且解读其源码，了解其中奥秘
 - 安全：介绍客户端配置安全相关议题

第二节 Spring Cloud Config Server

- 课程内容（1.5 - 2 小时）
 - 基本使用：介绍 `@EnableConfigServer`、`Environment` 仓储、秘钥管理等基本使用方法
 - 分布式配置官方实现：介绍 Spring 官方标准分布式配置实现方式：Git实现 和 文件系统实现
 - 动态配置属性 Bean：介绍 `@RefreshScope` 基本用法和使用场景，并且说明其中的局限性
 - 健康指标：介绍 Spring Boot 标准端口（`/health`）以及 健康指标（Health Indicator）
 - 分布式配置自定义实现：基于配置管理容器 Zookeeper，自定义实现分布式配置能力
 - 健康指标自定义实现：实现分布式配置的健康指标自定义实现

第三节 Spring Cloud Netflix Eureka

- 课程内容（1.5 - 2 小时）
 - 前微服务时代：介绍前微服务时代，服务发现和注册在 SOA 甚至是更早的时代的实现和实施方式，如 WebService 中的 UDDI、REST 中的 HEATOAS
 - 高可用架构：简介高可用架构的基本原则，计算方法和系统设计
 - Eureka 客户端：介绍 Spring Cloud Discovery 结合 Netflix Eureka 客户端的基本使用方法，包括服务发现激活、Eureka 客户端注册配置 以及 API 使用等
 - Eureka 服务器：介绍 Eureka 服务器作为服务注册中心的搭建方法，以及内建 Dashboard 基本运维手段

第四节 Spring Cloud Netflix Ribbon

- 课程内容（1.5 - 2 小时）
 - 理论基础：简介负载均衡客户端和服务端的相关理论，包括调度算法：如先来先服务、轮训、多级队列等。基本特性：非对称负载、健康检查、优先级队列等
 - 技术回顾：回顾 Spring Framework HTTP 组件 `RestTemplate` 的使用方法，结合 `ClientHttpRequestInterceptor` 实现简单负载均衡客户端
 - 整合 Netflix Ribbon：作为 Spring Cloud 客户端负载均衡实现，Netflix Ribbon 提供了丰富的组件，包括负载均衡器、负载均衡规则、PING 策略等，根据前章所积累的经验，实现客户端负载均衡

第五节 Spring Cloud Hystrix

- 课程内容（1.5 - 2 小时）
 - 核心理念：介绍服务短路的名词由来、目的，以及相关的类似概念。随后讲述其中设计哲学、触发条件、处理手段以及客户端和服务端实现方法
 - Spring Cloud Hystrix：作为服务端服务短路实现，介绍 Spring Cloud Hystrix 常用限流的功能，同时，说明健康指标以及数据指标在生产环境下的现实意义
 - 生产准备特性：介绍聚合数据指标 Turbine、Turbine Stream，以及整合 Hystrix Dashboard

第六节 Spring Cloud Feign

- 课程内容（1.5 - 2 小时）
 - 核心理念：回顾远程服务调用（RPC）的核心理念，介绍接口定义语言（IDL）以及服务存根（Stubs）以及通讯协议，如二进制协议 RMI、文本协议 REST 等
 - Spring Cloud Feign：介绍声明式客户端REST实现 Spring Cloud Feign的使用方式（如 `@EnableFeignClients`、`@FeignClient`），结合 Eureka 构建分布式服务应用
 - 整合支持：Spring Cloud Feign 整合 Hystrix 以及 Ribbon

第七节 Spring Cloud Stream

- 课程内容（1.5 小时）
 - Spring Cloud Stream
 - Kafka 绑定实现
 - RabbitMQ 绑定实现

第八节 Spring Cloud Bus

- 课程内容（1.5 小时）
 - 面向消息服务器
 - Kafka 实现
 - RabbitMQ 实现

第九节 Spring Cloud Sleuth

- 课程内容（1.5 小时）
 - 分布式应用跟踪
 - ZipKin 整合