

# Spring-Cloud 系列第 5 篇：spring-cloud-config-eureka-ribbon

作者：[xjtushilei](#)

原文链接：<https://ld246.com/article/1497579978998>

来源网站：[链滴](#)

许可协议：[署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

自学spring-cloud系列，越来越感觉spring-cloud很强大！

主要分为以下几篇：

1. [spring-cloud-config: 分布式配置管理](#)
2. [spring-cloud-eureka: 服务注册与发现](#)
3. [spring-cloud-eureka-consumer: 远程服务调用和及其负载均衡](#)
4. [spring-cloud-Hystrix: 熔断器保证服务高可用](#)
5. [spring-cloud-config-eureka-ribbon: 分布式配置管理的高可用](#)
6. [spring-cloud-bus: 配置信息的实时更新](#)
7. [spring-cloud-zuul: 网关基础服务](#)

## 介绍

spring cloud 的config server 如果挂掉了怎么办？那岂不是整个服务就瘫痪了，没关系，我们将其可用，利用其内嵌的 ribbon进行负载均衡。

## 准备

在开始之前，你需要明白：

- config server 和 config client 的基本使用（第一篇文章）
- eureka 的服务注册与发现（第二篇）
- 利用命令行 active profile 文件，启动多个实例（第三篇）

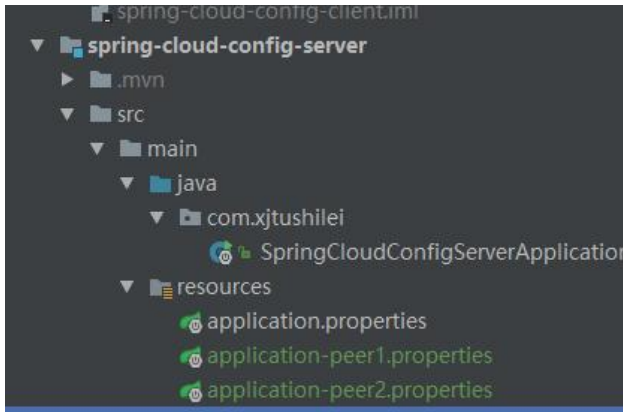
## 开始

### 1. server 端

配置文件：

```
spring.application.name=config-server  
  
server.port=8888  
spring.cloud.config.server.git.uri=file:///${user.home}/config-repo  
eureka.client.service-url.defaultZone=http://localhost:8761/eureka/
```

几乎没有变化，多启用几个实例就好了，在配置文件中只改变端口就行了。



## 2. client 端

### 2.1. 配置文件

在设置文件中去掉直接的 url 路径，加入自动发现功能即可。

坑：这三个配置文件都需要放到bootstrap.properties的配置中。因为在加载application.properties的时候，已经取到了git上的配置文件

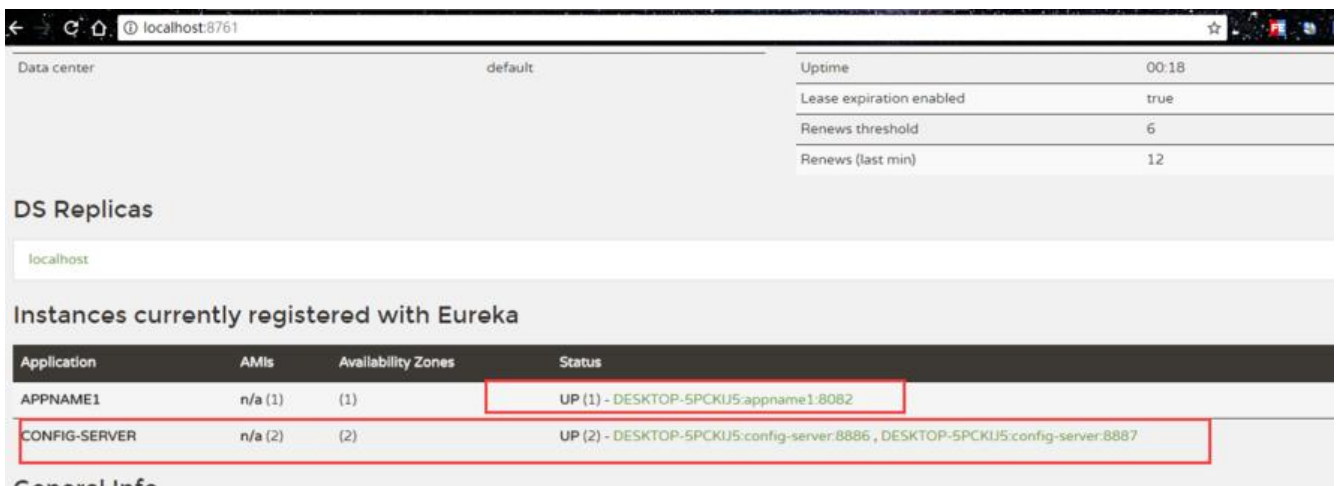
```
spring.cloud.config.discovery.enabled=true  
spring.cloud.config.discovery.serviceId=config-server
```

```
eureka.client.service-url.defaultZone=http://localhost:8761/eureka/
```

其中，serviceId 是你的服务的名字。

本质就是根据服务名称，来用ribbon选择使用哪个server服务器。

## 测试



如图，server有两个，client的应用有一个，当我们停掉一个server服务，重启client依然能够正常运

因为我们还没有学到热加载，即更改git的配置文件，立即热更新服务，不需要重启，所以必须关掉cli

在下一篇，我们将会介绍具体的使用 Spring Cloud Bus 来热加载配置信息。

## 示例源码

所有源码在我的github仓库里，传送门：<https://github.com/xjtushilei/spring-cloud-simples.git>

## 支持

如果你喜欢~给个星