1. 在config-server(服务端)中添加如下依赖:

<dependency>

   <groupId>org.springframework.cloud</groupId>

   <artifactId>spring-cloud-starter-eureka</artifactId>

</dependency>

1. 在config-server的入口类上添加@EnableDiscoveryClient注解,表示这是一个Eureka客户端
2. 在application.properties中配置eureka注册中心地址：

eureka.client.service-url.defaultZone=http://localhost:1111/eureka/

1. config-client(客户端)改造第一步也是先添加依赖

<dependency>

   <groupId>org.springframework.cloud</groupId>

   <artifactId>spring-cloud-starter-eureka</artifactId>

</dependency>

1. 入口类添加@EnableDiscoveryClient注解
2. 最后修改配置文件

spring.application.name=app

# dev根据具体情况来修改

spring.cloud.config.profile=dev

spring.cloud.config.label=master

eureka.client.service-url.defaultZone=http://localhost:1111/eureka/

spring.cloud.config.discovery.enabled=true

spring.cloud.config.discovery.service-id=config-server

server.port=2008

1.eureka.client.service-url.defaultZone设置了注册中心的地址，将config-client注册到eureka注册中心中去  
2.spring.cloud.config.discovery.enabled表示开启通过服务名来访问config-server  
3.spring.cloud.config.discovery.service-id=config-server则表示config-server的服务名

**失败快速响应**

不作任何额外配置的情况下，失败响应有点迟钝，举个简单的例子，关掉config-server，我们直接启动config-client，此时启动会报错，但是报错时间较晚，报错的时候系统已经打印了许多启动日志了，如果我们希望在启动失败时能够快速响应，方式很简单，config-client中添加如下配置即可：

spring.cloud.config.fail-fast=true

此时不启动config-server直接启动config-client依然会报错，但是我们看到报错时间较早，系统都没打印几条启动日志。

## 重试机制

如果由于网络抖动等原因导致config-client在启动时候访问config-server没有访问成功从而报错，这显然是不划算的，遇到这种情况我们希望config-client最好能重试几次，重试机制在这里也是受支持的，添加重试机制的方式很简单，引入如下两个依赖：

<dependency>

   <groupId>org.springframework.retry</groupId>

   <artifactId>spring-retry</artifactId>

</dependency>

<dependency>

   <groupId>org.springframework.boot</groupId>

   <artifactId>spring-boot-starter-aop</artifactId>

</dependency>

引入依赖就OK了，不用做任何额外配置（当然要确保失败快速响应已开启）

和重试机制相关的配置有如下四个：

# 配置重试次数，默认为6

spring.cloud.config.retry.max-attempts=6

# 间隔乘数，默认1.1

spring.cloud.config.retry.multiplier=1.1

# 初始重试间隔时间，默认1000ms

spring.cloud.config.retry.initial-interval=1000

# 最大间隔时间，默认2000ms

spring.cloud.config.retry.max-interval=2000

## 动态刷新配置

有的时候，我动态的更新了Git仓库中的配置文件，那么我如何让我的config-client能够及时感知到呢？方式很简单，首先在config-client中添加如下依赖：

<dependency>

   <groupId>org.springframework.boot</groupId>

   <artifactId>spring-boot-starter-actuator</artifactId>

</dependency>

该依赖中包含了/refresh端点的实现，我们将利用这个端点来刷新配置信息。然后需要在application.properties中配置忽略权限拦截：

management.security.enabled=false