



链滴

集中配置组件 SpringCloud Config

作者: [96XL](#)

原文链接: <https://ld246.com/article/1564733962580>

来源网站: [链滴](#)

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

前言

  在分布式系统中，由于服务数量巨大，为了方便服务配置文件统一管理，实时更新，所

以需要分布式配置中心组件。

正文

  在SpringCloud中，有分布式配置中心组件SpringCloud

Config，它支持配置服务放在配置服务的内存中（即本地），也支持放在远程Git仓库中。在SpringCloud Config 组件中，分两个角色，一是Config Server，二是Config

Client。

  Config Server是一个可横向扩展、集中式的配置服务器，它用于集中管理应用程序各个环境下的配置，默认使用Git存储配置文件内容，也可以使用SVN存储，或者是本地文件存储。

  Config Client是Config Server的客户端，用于操作存储在Config Server中配置内容。

微服务在启动时会请求Config Server获取配置文件的内容，请求到后再启动容器。

代码实现

• 配置服务端

  1. 在码云上创建仓库(点击右上角的加号，下拉菜单选择新建仓库);



  2. 上传配置文件，将某个工程的application.yml改名后上传（文件规则为application-profile.yml或application-profile.properties，其中application为应用名称，profile指的开发环境，如base-dev.yml）；



  3. 创建配置中心工程模块，引入依赖；

```
<dependencies>
  <dependency>
    <groupId>org.springframework.cloud</groupId>
    <artifactId>spring-cloud-config-server</artifactId>
  </dependency>
</dependencies>
```

  4.创建启动类ConfigServerApplication;

```
// 开启配置服务
@EnableConfigServer
@SpringBootApplication
public class ConfigServerApplication {
    public static void main(String[] args) {
        SpringApplication.run(ConfigServerApplication.class, args);
    }
}
```

  5.编写配置文件application.yml（注意缩进）；

```
spring:
  application:
    name: tensquare-config
  cloud:
    config:
      server:
        git:
          uri: https://gitee.com/wmb3609/tensquare_config.git #码云仓库的地址
server:
  port: 12000
```

  6.启动浏览器访问http://localhost:12000/base-dev.yml 可以看到配置内（base-dev.yml为之前上传的文件名）。

● 配置客户端

  1.在之前上传配置文件的工程中引入依赖；

```
<dependency>
  <groupId>org.springframework.cloud</groupId>
  <artifactId>spring-cloud-starter-config</artifactId>
</dependency>
```

  2.删除已上传的application.yml，添加bootstrap.yml；

```
spring:
  cloud:
    config:
      name: base #上传的配置文件文件名前缀
      profile: dev #上传的配置文件文件名后缀
      label: master #指定仓库分支
      uri: http://127.0.0.1:12000 #配置中心工程地址
```

  3.启动配置中心工程和客户端工程，测试能否正常启动。