

一 微服务大量重复配置

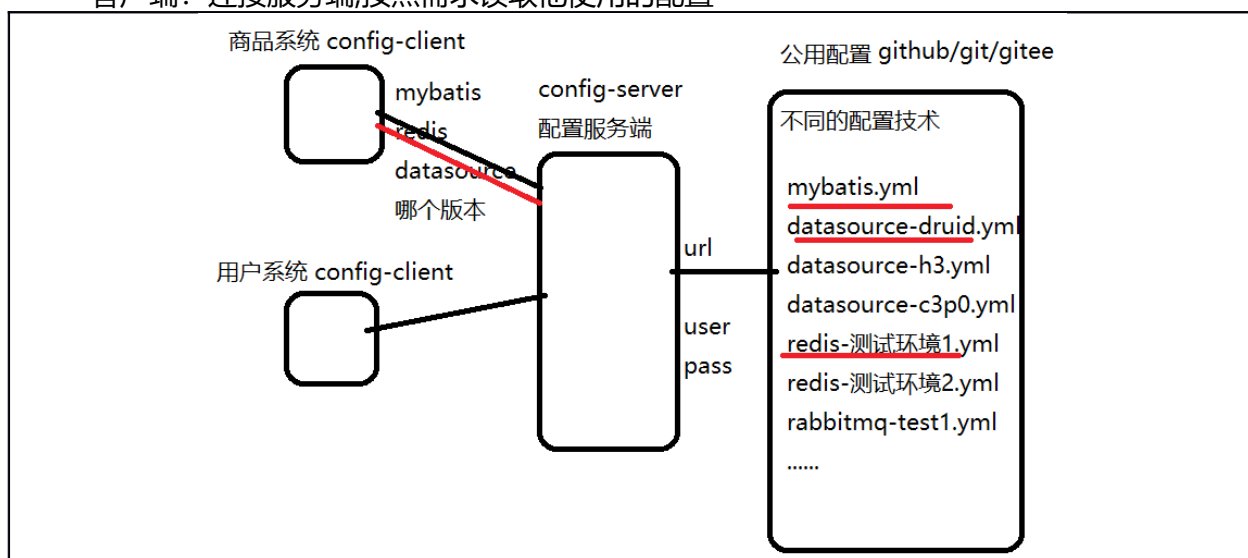
大量的微服务工程,application 全局配置文件,非常非常多的重复配置的.并且不好管理---这些全局配置文件的问题可以交给 spring cloud config.

二 springcloud config 结构

服务端和客户端

服务端：连接公共网络资源读取公用的配置属性

客户端：连接服务端,按照需求读取他使用的配置



三 client 从 server 获取配置的方式

可以通过访问 config-server 配置中心，决定拿到 git 上哪些 properties/yml 的配置文件.

配置中心提供了多种不同的接口地址(mapping 映射地址)，让 config-client 选择性的获取远程 git 的不同配置文件.

例：

localhost:8098/redis-test1.properties//读取 redis-test1.properties 一个文件

localhost:8098/redis/test1// （一次读取到多个文件,包括 redis-test1.properties）

1 访问格式:

```
/application/profile  
/application-profile.[json/yml/properties]
```

application:

应用名称: 使用应用名字区分各种技术,

例如: redis.properties, mybatis.properties

profile:

配合 application,给同一个应用提供不同的配置内容.因为即使都是 redis 的配置,也区分测试环境和线上生产环境.

例如: redis-test1.properties,redis-test2.properties,redis-product.properties

所以为了迎合这样的格式,准备 git 上的公用配置文件一般都是

`application-profile.yml/json/properties`

2 特殊情况

2.A 只有 application 没有 profile 的配置文件

读取的特点, 只要 application 同名就能读到里面的内容.

做同一个应用同一个技术的公用配置

2.B 名称不是严格的 application-profile 格式

例如: demo01-test02-version01.properties

下例都能访问该配置文件:

demo01-test02==可解析为 application

demo01==可解析为 application

3 结论

配置中心一旦搭建 config-server 就会按照他提供的接口接收我们的请求,解析请求中的参数,读取 git 的不同配置文件.

四 搭建 config-server

可以通过访问这个 config-server 配置中心, 决定拿到 git 上哪些.properties/.yml 的配置文件.


配置中心提供了多种不同的接口地址(mapping 映射地址), 让 config-client 选择性的获取远程 git 的不同配置文件.

- maven 项目
 - pom.xml
 - 继承:springboot
 - 导入:springcloud
 - 依赖:
 - config-server
 - application.properties
 - 端口:8098
 - git 服务器仓库地址
 - git 仓库下的文件夹
- server.port=8098

1 导入依赖

```
<!--config 服务器-->
<dependency>
    <groupId>org.springframework.cloud</groupId>
    <artifactId>spring-cloud-config-server</artifactId>
</dependency>
```

2 application.properties



The screenshot shows the Gitee repository page for 'easymall-test / easymall-config'. The repository is in the 'master' branch and has 3 commits, 1 branch, 0 tags, 0 releases, and 1 contributor. The commit history shows three commits: 'README.en.md', 'README.md', and 'datasource-mysql.properties', all committed 3 hours ago. The commit message for 'datasource-mysql.properties' is '重命名 datasource-mysql 为 datasource-mysql.properties'.

```
server.port=8098
#git 服务器的仓库 gitee 配置
spring.cloud.config.server.git.uri=https://gitee.com/easymall-test/easymall-config
#私有仓库账号密码 公有仓库不需要
spring.cloud.config.server.git.username=15723468981
spring.cloud.config.server.git.password=Sxs411051

#git 的某个仓库下是否有多个文件夹，可以指定多个或一个
#spring.cloud.config.server.git.search-paths=configtest
#TODO 高可用，微服务注册
```

3 启动类

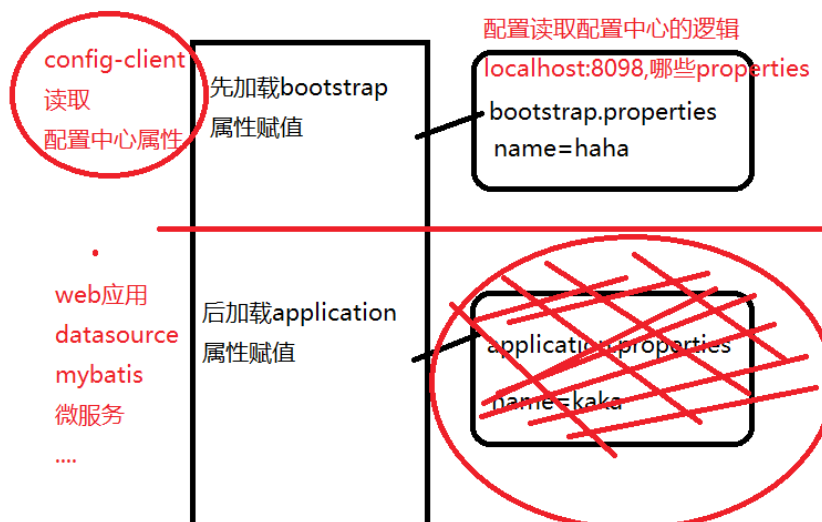
```
@SpringBootApplication
@EnableConfigServer
public class StarterConfig {
    public static void main(String[] args) {
        SpringApplication.run(StarterConfig.class, args);
    }
}
```

五 搭建 config-client

当一个 web 应用引入 config-client 之后,在 web 应用启动完整之前,最先做的事就是一个 client 进程读取 config-server 的所有配置属性,交给 web 应用去使用

- maven 项目
- pom:
 - 继承:springboot
 - 导入:springcloud
 - 依赖:
 - config-client
 - starter-web
- application.properties(不再使用)
- bootstrap.properties (使用这个配置文件)

1 application.properties 与 bootstrap.properties 区别



2 导入依赖

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

3 bootstrap.properties

```
server.port=8099
#config-client 继承指定读取远程公用配置内容
#redis-test1.properties
#当前客户端想要读取的 redis datasource mybatia 应用名称
spring.cloud.config.name=redis
spring.cloud.config.profile=test1
#发起的请求就是 localhost:8098/redis/test1
#指定一下 config-server 地址
spring.cloud.config.uri=http://localhost:8098
```

六 config-server 高可用

这个结构一旦 config-server 宕机,网络延迟,就无法让客户端访问获取必要的一些属性。

解决高可用访问思路: 服务治理组件,只要让 config-client 通过服务调用的方式, 访问多个配置中心服务端就行了。

1 config-server 作为 eureka 客户端在注册中心注册服务

1.A 增加依赖 eureka

```
<!--eureka 客户端-->
<dependency>
  <groupId>org.springframework.cloud</groupId>
  <artifactId>spring-cloud-starter-eureka</artifactId>
</dependency>
```

1.B 增加配置

```
spring.application.name=config-server
eureka.client.service-
url.defaultZone=http://127.0.0.1:8761/eureka
eureka.instance.prefer-ip-address=true
```

2 config-client 作为 eureka 客户端在注册中心抓取服务'

2.A 增加依赖 eureka

```
<!--eureka 客户端-->
<dependency>
  <groupId>org.springframework.cloud</groupId>
  <artifactId>spring-cloud-starter-eureka</artifactId>
</dependency>
```

2.B 完整配置

```
server.port=8099
#config-client 继承指定读取远程公用配置内容
#redis-test1.properties
#当前客户端想要读取的 redis datasource mybatia 应用名称
spring.cloud.config.name=redis
spring.cloud.config.profile=test1
#发起的请求就是 localhost:8098/redis/test1
#指定一下 config-server 地址
#spring.cloud.config.uri=http://localhost:8098
#微服务调用
spring.application.name=config-client
eureka.client.service-
url.defaultZone=http://127.0.0.1:8761/eureka
eureka.instance.prefer-ip-address=true
#通过服务访问配置中心 config-server
spring.cloud.config.discovery.enabled=true
spring.cloud.config.discovery.service-id=config-server
```