spring cloud学习总结

目录

[微服务介绍 2](#_Toc522637527)

[服务注册和发现Eureka 3](#_Toc522637528)

[负载均衡Ribbon 8](#_Toc522637529)

[声明式调用Feign 12](#_Toc522637530)

[熔断器Hystrix 16](#_Toc522637531)

[路由网关Zuul 16](#_Toc522637532)

[配置服务Config 22](#_Toc522637533)

# 微服务介绍

微服务是系统架构上的一种设计风格，它的主旨是讲一个原本独立的系统拆分成多个小型服务，这些小型服务都在各自独立的进程中运行，服务之间通过基于HTTP的RESTful API进行通信。被拆分成的每一个小型服务都围绕着系统中的某一项或一些耦合度较高的业务功能进行构建，并且每个服务都维护着自身的数据存储、业务开发、自动化测试案例以及独立部署机制。由于有了轻量级的通信协作基础，所以这些位服务可以使用不同的语言来编写。

以下代码的运行环境：

spring cloud版本：2.0.4

JDK版本：1.8

IDE工具：IDEA ULTIMATE 2018.2

# 服务注册和发现Eureka

**服务器：**

pom.xml文件

*<?***xml version="1.0" encoding="UTF-8"***?>*<**project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd"**>  
 <**modelVersion**>4.0.0</**modelVersion**>  
  
 <**groupId**>com.liagn</**groupId**>  
 <**artifactId**>demo-liagn-eureka-server</**artifactId**>  
 <**version**>0.0.1-SNAPSHOT</**version**>  
 <**packaging**>jar</**packaging**>  
  
 <**name**>demo-liagn-eureka-server</**name**>  
 <**description**>Demo project for Spring Boot</**description**>  
  
 <**parent**>  
 <**groupId**>org.springframework.boot</**groupId**>  
 <**artifactId**>spring-boot-starter-parent</**artifactId**>  
 <**version**>2.0.4.RELEASE</**version**>  
 <**relativePath**/> *<!-- lookup parent from repository -->* </**parent**>  
  
 <**properties**>  
 <**project.build.sourceEncoding**>UTF-8</**project.build.sourceEncoding**>  
 <**project.reporting.outputEncoding**>UTF-8</**project.reporting.outputEncoding**>  
 <**java.version**>1.8</**java.version**>  
 <**spring-cloud.version**>Finchley.SR1</**spring-cloud.version**>  
 </**properties**>  
  
 <**dependencies**>  
 <**dependency**>  
 <**groupId**>org.springframework.cloud</**groupId**>  
 <**artifactId**>spring-cloud-starter-netflix-eureka-server</**artifactId**>  
 </**dependency**>  
  
 <**dependency**>  
 <**groupId**>org.springframework.boot</**groupId**>  
 <**artifactId**>spring-boot-starter-test</**artifactId**>  
 <**scope**>test</**scope**>  
 </**dependency**>  
 </**dependencies**>  
  
 <**dependencyManagement**>  
 <**dependencies**>  
 <**dependency**>  
 <**groupId**>org.springframework.cloud</**groupId**>  
 <**artifactId**>spring-cloud-dependencies</**artifactId**>  
 <**version**>${spring-cloud.version}</**version**>  
 <**type**>pom</**type**>  
 <**scope**>import</**scope**>  
 </**dependency**>  
 </**dependencies**>  
 </**dependencyManagement**>  
  
 <**build**>  
 <**plugins**>  
 <**plugin**>  
 <**groupId**>org.springframework.boot</**groupId**>  
 <**artifactId**>spring-boot-maven-plugin</**artifactId**>  
 </**plugin**>  
 </**plugins**>  
 </**build**>  
  
  
</**project**>

配置文件application.yml

**server**:  
 **port**: 8761  
**eureka**:  
 **client**:  
 **register-with-eureka**: **false  
 fetch-registry**: **false  
 serviceUrl**:  
 **defaultZone**: http://localhost:8761/eureka/

启动文件：

**package** com.liagn.demoliagneurekaserver;  
  
**import** org.springframework.boot.SpringApplication;  
**import** org.springframework.boot.autoconfigure.SpringBootApplication;  
**import** org.springframework.cloud.netflix.eureka.server.EnableEurekaServer;  
  
@SpringBootApplication  
@EnableEurekaServer  
**public class** DemoLiagnEurekaServerApplication {  
  
 **public static void** main(String[] args) {  
 SpringApplication.*run*(DemoLiagnEurekaServerApplication.**class**, args);  
 }  
}

在主启动类加上@EnableEurekaServer注解，开启Eureka server功能

**客户端：**

pom启动文件：

*<?***xml version="1.0" encoding="UTF-8"***?>*<**project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd"**>  
 <**modelVersion**>4.0.0</**modelVersion**>  
  
 <**groupId**>com.liagn</**groupId**>  
 <**artifactId**>demo-liagn-eureka-client</**artifactId**>  
 <**version**>0.0.1-SNAPSHOT</**version**>  
 <**packaging**>jar</**packaging**>  
  
 <**name**>demo-liagn-eureka-client</**name**>  
 <**description**>Demo project for Spring Boot</**description**>  
  
 <**parent**>  
 <**groupId**>org.springframework.boot</**groupId**>  
 <**artifactId**>spring-boot-starter-parent</**artifactId**>  
 <**version**>2.0.4.RELEASE</**version**>  
 <**relativePath**/> *<!-- lookup parent from repository -->* </**parent**>  
  
 <**properties**>  
 <**project.build.sourceEncoding**>UTF-8</**project.build.sourceEncoding**>  
 <**project.reporting.outputEncoding**>UTF-8</**project.reporting.outputEncoding**>  
 <**java.version**>1.8</**java.version**>  
 <**spring-cloud.version**>Finchley.SR1</**spring-cloud.version**>  
 </**properties**>  
  
 <**dependencies**>  
 <**dependency**>  
 <**groupId**>org.springframework.cloud</**groupId**>  
 <**artifactId**>spring-cloud-starter-netflix-eureka-client</**artifactId**>  
 </**dependency**>  
  
 <**dependency**>  
 <**groupId**>org.springframework.boot</**groupId**>  
 <**artifactId**>spring-boot-starter-web</**artifactId**>  
 </**dependency**>  
  
 <**dependency**>  
 <**groupId**>org.springframework.boot</**groupId**>  
 <**artifactId**>spring-boot-starter-test</**artifactId**>  
 <**scope**>test</**scope**>  
 </**dependency**>  
 </**dependencies**>  
  
 <**dependencyManagement**>  
 <**dependencies**>  
 <**dependency**>  
 <**groupId**>org.springframework.cloud</**groupId**>  
 <**artifactId**>spring-cloud-dependencies</**artifactId**>  
 <**version**>${spring-cloud.version}</**version**>  
 <**type**>pom</**type**>  
 <**scope**>import</**scope**>  
 </**dependency**>  
 </**dependencies**>  
 </**dependencyManagement**>  
  
 <**build**>  
 <**plugins**>  
 <**plugin**>  
 <**groupId**>org.springframework.boot</**groupId**>  
 <**artifactId**>spring-boot-maven-plugin</**artifactId**>  
 </**plugin**>  
 </**plugins**>  
 </**build**>  
  
  
</**project**>

配置文件：application.yml

**server**:  
 **port**: 8789  
**spring**:  
 **application**:  
 **name**: lianghua  
**eureka**:  
 **client**:  
 **serviceUrl**:  
 **defaultZone**: http://localhost:8761/eureka  
 **instance**:  
 **prefer-ip-address**: true

主启动文件：

**package** com.liagn.demoliagneurekaclient;  
  
**import** org.springframework.boot.SpringApplication;  
**import** org.springframework.boot.autoconfigure.SpringBootApplication;  
**import** org.springframework.cloud.client.discovery.EnableDiscoveryClient;  
  
@SpringBootApplication  
@EnableDiscoveryClient  
**public class** DemoLiagnEurekaClientApplication {  
  
 **public static void** main(String[] args) {  
 SpringApplication.*run*(DemoLiagnEurekaClientApplication.**class**, args);  
 }  
}

在启动类上增加@EnableDiscoveryClient注解，开启Eureka Client功能

接口类：HiController

**package** com.liagn.demoliagneurekaclient;  
  
**import** org.springframework.beans.factory.annotation.Value;  
**import** org.springframework.web.bind.annotation.GetMapping;  
**import** org.springframework.web.bind.annotation.RequestParam;  
**import** org.springframework.web.bind.annotation.RestController;  
  
@RestController  
**public class** HiController {  
 @Value(**"${server.port}"**)  
 String **port**;  
 @GetMapping(**"/hi"**)  
 **public** String index(@RequestParam String name){  
 **return "hi "**+name+**",i am from port: "**+**port**;  
 }  
}

Eureka 服务端的依赖为：

<**dependency**>  
 <**groupId**>org.springframework.cloud</**groupId**>  
 <**artifactId**>spring-cloud-starter-netflix-eureka-server</**artifactId**>  
</**dependency**>

Eureka 客户端的依赖为：

<**dependency**>  
 <**groupId**>org.springframework.cloud</**groupId**>  
 <**artifactId**>spring-cloud-starter-netflix-eureka-client</**artifactId**>  
</**dependency**>

# 负载均衡Ribbon

pom启动文件：

*<?***xml version="1.0" encoding="UTF-8"***?>*<**project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd"**>  
 <**modelVersion**>4.0.0</**modelVersion**>  
  
 <**groupId**>com.liagn</**groupId**>  
 <**artifactId**>eureka-ribbon-client</**artifactId**>  
 <**version**>0.0.1-SNAPSHOT</**version**>  
 <**packaging**>jar</**packaging**>  
  
 <**name**>eureka-ribbon-client</**name**>  
 <**description**>Demo project for Spring Boot</**description**>  
  
 <**parent**>  
 <**groupId**>org.springframework.boot</**groupId**>  
 <**artifactId**>spring-boot-starter-parent</**artifactId**>  
 <**version**>2.0.4.RELEASE</**version**>  
 <**relativePath**/> *<!-- lookup parent from repository -->* </**parent**>  
  
 <**properties**>  
 <**project.build.sourceEncoding**>UTF-8</**project.build.sourceEncoding**>  
 <**project.reporting.outputEncoding**>UTF-8</**project.reporting.outputEncoding**>  
 <**java.version**>1.8</**java.version**>  
 <**spring-cloud.version**>Finchley.SR1</**spring-cloud.version**>  
 </**properties**>  
  
 <**dependencies**>  
 <**dependency**>  
 <**groupId**>org.springframework.cloud</**groupId**>  
 <**artifactId**>spring-cloud-starter-netflix-ribbon</**artifactId**>  
 </**dependency**>  
  
 <**dependency**>  
 <**groupId**>org.springframework.cloud</**groupId**>  
 <**artifactId**>spring-cloud-starter-netflix-hystrix</**artifactId**>  
 <**version**>2.0.1.RELEASE</**version**>  
 </**dependency**>  
  
 <**dependency**>  
 <**groupId**>org.springframework.cloud</**groupId**>  
 <**artifactId**>spring-cloud-starter-netflix-eureka-client</**artifactId**>  
 </**dependency**>  
  
 <**dependency**>  
 <**groupId**>org.springframework.boot</**groupId**>  
 <**artifactId**>spring-boot-starter-web</**artifactId**>  
 </**dependency**>  
  
 <**dependency**>  
 <**groupId**>org.springframework.boot</**groupId**>  
 <**artifactId**>spring-boot-starter-test</**artifactId**>  
 <**scope**>test</**scope**>  
 </**dependency**>  
 </**dependencies**>  
  
 <**dependencyManagement**>  
 <**dependencies**>  
 <**dependency**>  
 <**groupId**>org.springframework.cloud</**groupId**>  
 <**artifactId**>spring-cloud-dependencies</**artifactId**>  
 <**version**>${spring-cloud.version}</**version**>  
 <**type**>pom</**type**>  
 <**scope**>import</**scope**>  
 </**dependency**>  
 </**dependencies**>  
 </**dependencyManagement**>  
  
 <**build**>  
 <**plugins**>  
 <**plugin**>  
 <**groupId**>org.springframework.boot</**groupId**>  
 <**artifactId**>spring-boot-maven-plugin</**artifactId**>  
 </**plugin**>  
 </**plugins**>  
 </**build**>  
  
  
</**project**>

配置文件：application.yml

**spring**:  
 **application**:  
 **name**: eureka-ribbon-client  
**server**:  
 **port**: 8900  
**eureka**:  
 **client**:  
 **serviceUrl**:  
 **defaultZone**: http://localhost:8761/eureka/

主启动文件：

**package** com.liagn.eurekaribbonclient;  
  
**import** org.springframework.boot.SpringApplication;  
**import** org.springframework.boot.autoconfigure.SpringBootApplication;  
**import** org.springframework.cloud.client.discovery.EnableDiscoveryClient;  
**import** org.springframework.cloud.client.loadbalancer.LoadBalanced;  
**import** org.springframework.cloud.netflix.hystrix.EnableHystrix;  
**import** org.springframework.context.annotation.Bean;  
**import** org.springframework.web.client.RestTemplate;  
  
@SpringBootApplication  
@EnableDiscoveryClient  
@EnableHystrix  
**public class** EurekaRibbonClientApplication {  
  
 @Bean  
 @LoadBalanced  
 **public** RestTemplate restTemplate(){  
 **return new** RestTemplate();  
 }  
 **public static void** main(String[] args) {  
 SpringApplication.*run*(EurekaRibbonClientApplication.**class**, args);  
 }  
}

在restTemplate上增加@Bean和@LoadBalanced注解，此时RestTemplate就结合了Ribbon开启了负载均衡功能

接口类：RibbonController

**package** com.liagn.eurekaribbonclient;  
  
**import** com.netflix.hystrix.contrib.javanica.annotation.HystrixCommand;  
**import** org.springframework.beans.factory.annotation.Autowired;  
**import** org.springframework.web.bind.annotation.\*;  
**import** org.springframework.web.client.RestTemplate;  
  
@RestController  
**public class** RibbonController {  
  
 @Autowired  
 RestTemplate **restTemplate**;  
  
 @HystrixCommand(fallbackMethod = **"hiError"**)  
 @RequestMapping(value=**"/hi"**,method = RequestMethod.***GET***)  
 **public** String hi(@RequestParam(required = **false**,defaultValue = **"liagn"**)String name){  
 **return restTemplate**.getForEntity(**"http://lianghua/hi?name={1}"**,String.**class**,**"liagn"**).getBody();  
 }  
  
 **public** String hiError(String name){  
 **return "hi,"**+name+**",sorry,error!"**;  
 }  
}

Ribbon的依赖为：

<**dependency**>  
 <**groupId**>org.springframework.cloud</**groupId**>  
 <**artifactId**>spring-cloud-starter-netflix-ribbon</**artifactId**>  
</**dependency**>

# 声明式调用Feign

POM文件

*<?***xml version="1.0" encoding="UTF-8"***?>*<**project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd"**>  
 <**modelVersion**>4.0.0</**modelVersion**>  
  
 <**groupId**>com.liagn</**groupId**>  
 <**artifactId**>eureka-feign-client</**artifactId**>  
 <**version**>0.0.1-SNAPSHOT</**version**>  
 <**packaging**>jar</**packaging**>  
  
 <**name**>eureka-feign-client</**name**>  
 <**description**>Demo project for Spring Boot</**description**>  
  
 <**parent**>  
 <**groupId**>org.springframework.boot</**groupId**>  
 <**artifactId**>spring-boot-starter-parent</**artifactId**>  
 <**version**>2.0.4.RELEASE</**version**>  
 <**relativePath**/> *<!-- lookup parent from repository -->* </**parent**>  
  
 <**properties**>  
 <**project.build.sourceEncoding**>UTF-8</**project.build.sourceEncoding**>  
 <**project.reporting.outputEncoding**>UTF-8</**project.reporting.outputEncoding**>  
 <**java.version**>1.8</**java.version**>  
 <**spring-cloud.version**>Finchley.SR1</**spring-cloud.version**>  
 </**properties**>  
  
 <**dependencies**>  
 <**dependency**>  
 <**groupId**>org.springframework.boot</**groupId**>  
 <**artifactId**>spring-boot-starter-web</**artifactId**>  
 </**dependency**>  
 <**dependency**>  
 <**groupId**>org.springframework.cloud</**groupId**>  
 <**artifactId**>spring-cloud-starter-netflix-eureka-client</**artifactId**>  
 </**dependency**>  
 <**dependency**>  
 <**groupId**>org.springframework.cloud</**groupId**>  
 <**artifactId**>spring-cloud-starter-openfeign</**artifactId**>  
 </**dependency**>  
  
 <**dependency**>  
 <**groupId**>org.springframework.boot</**groupId**>  
 <**artifactId**>spring-boot-starter-test</**artifactId**>  
 <**scope**>test</**scope**>  
 </**dependency**>  
 </**dependencies**>  
  
 <**dependencyManagement**>  
 <**dependencies**>  
 <**dependency**>  
 <**groupId**>org.springframework.cloud</**groupId**>  
 <**artifactId**>spring-cloud-dependencies</**artifactId**>  
 <**version**>${spring-cloud.version}</**version**>  
 <**type**>pom</**type**>  
 <**scope**>import</**scope**>  
 </**dependency**>  
 </**dependencies**>  
 </**dependencyManagement**>  
  
 <**build**>  
 <**plugins**>  
 <**plugin**>  
 <**groupId**>org.springframework.boot</**groupId**>  
 <**artifactId**>spring-boot-maven-plugin</**artifactId**>  
 </**plugin**>  
 </**plugins**>  
 </**build**>  
  
  
</**project**>

配置文件：application.properties

**spring.application.name**=**eureka-feign-client  
server.port**=**8901  
eureka.client.serviceUrl.defaultZone**=**http://localhost:8761/eureka/**

主启动文件：

**package** com.liagn.eurekafeignclient;  
  
**import** org.springframework.boot.SpringApplication;  
**import** org.springframework.boot.autoconfigure.SpringBootApplication;  
**import** org.springframework.cloud.client.discovery.EnableDiscoveryClient;  
**import** org.springframework.cloud.openfeign.EnableFeignClients;  
  
@SpringBootApplication  
@EnableDiscoveryClient  
@EnableFeignClients  
**public class** EurekaFeignClientApplication {  
  
 **public static void** main(String[] args) {  
 SpringApplication.*run*(EurekaFeignClientApplication.**class**, args);  
 }  
}

在主启动类中加上@EnableFeignClients开启Feign Client功能

feign接口文件：EurekaClientFeign

**package** com.liagn.eurekafeignclient;  
  
**import** org.springframework.cloud.openfeign.FeignClient;  
**import** org.springframework.stereotype.Component;  
**import** org.springframework.web.bind.annotation.GetMapping;  
**import** org.springframework.web.bind.annotation.RequestParam;  
  
@FeignClient(value=**"lianghua"**)  
@Component  
**public interface** EurekaClientFeign {  
 @GetMapping(value = **"/hi"**)  
 String sayHiFromClientEurka(@RequestParam(value=**"name"**) String name);  
}

Feign配置类

**package** com.liagn.eurekafeignclient;  
  
**import** feign.Retryer;  
**import** org.springframework.context.annotation.Bean;  
**import** org.springframework.context.annotation.Configuration;  
  
@Configuration  
**public class** FeignConfig {  
 @Bean  
 **public** Retryer feignRetryer(){  
 **return new** Retryer.Default();  
 }  
}

配置类中加上@Configuration注解，表明该类是一个配置内；

HiService类

**package** com.liagn.eurekafeignclient;  
  
**import** org.springframework.beans.factory.annotation.Autowired;  
**import** org.springframework.stereotype.Service;  
  
  
@Service  
**public class** HiService {  
 @Autowired  
 EurekaClientFeign **eurekaClientFeign**;  
 **public** String sayHi(String name){  
 **return eurekaClientFeign**.sayHiFromClientEurka(name);  
 }  
}

接口类：HiController

**package** com.liagn.eurekafeignclient;  
  
**import** org.springframework.beans.factory.annotation.Autowired;  
**import** org.springframework.web.bind.annotation.GetMapping;  
**import** org.springframework.web.bind.annotation.RequestParam;  
**import** org.springframework.web.bind.annotation.RestController;  
  
@RestController  
**public class** HiController {  
 @Autowired  
 HiService **hiService**;  
 @GetMapping(**"/hi"**)  
 **public** String sayHi(@RequestParam(defaultValue = **"liagn"**,required = **false**) String name){  
 **return hiService**.sayHi(name);  
 }  
}

# 熔断器Hystrix

1，代码集成到Ribbon里面，见Ribbon的代码

2，Hystrix的依赖为：

<**dependency**>  
 <**groupId**>org.springframework.cloud</**groupId**>  
 <**artifactId**>spring-cloud-starter-netflix-hystrix</**artifactId**>  
 <**version**>2.0.1.RELEASE</**version**>  
</**dependency**>

3，在主启动类中增加@EnableHystrix

4，在接口类中方法前加入@HystrixCommand（fallbackMethod=”方法名”）

# 路由网关Zuul

POM文件：

*<?***xml version="1.0" encoding="UTF-8"***?>*<**project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd"**>  
 <**modelVersion**>4.0.0</**modelVersion**>  
  
 <**groupId**>com.liagn</**groupId**>  
 <**artifactId**>eureka-zuul-client</**artifactId**>  
 <**version**>0.0.1-SNAPSHOT</**version**>  
 <**packaging**>jar</**packaging**>  
  
 <**name**>eureka-zuul-client</**name**>  
 <**description**>Demo project for Spring Boot</**description**>  
  
 <**parent**>  
 <**groupId**>org.springframework.boot</**groupId**>  
 <**artifactId**>spring-boot-starter-parent</**artifactId**>  
 <**version**>2.0.4.RELEASE</**version**>  
 <**relativePath**/> *<!-- lookup parent from repository -->* </**parent**>  
  
 <**properties**>  
 <**project.build.sourceEncoding**>UTF-8</**project.build.sourceEncoding**>  
 <**project.reporting.outputEncoding**>UTF-8</**project.reporting.outputEncoding**>  
 <**java.version**>1.8</**java.version**>  
 <**spring-cloud.version**>Finchley.SR1</**spring-cloud.version**>  
 </**properties**>  
  
 <**dependencies**>  
 <**dependency**>  
 <**groupId**>org.springframework.boot</**groupId**>  
 <**artifactId**>spring-boot-starter-web</**artifactId**>  
 </**dependency**>  
 <**dependency**>  
 <**groupId**>org.springframework.cloud</**groupId**>  
 <**artifactId**>spring-cloud-starter-netflix-eureka-client</**artifactId**>  
 </**dependency**>  
 <**dependency**>  
 <**groupId**>org.springframework.cloud</**groupId**>  
 <**artifactId**>spring-cloud-starter-netflix-zuul</**artifactId**>  
 </**dependency**>  
  
 <**dependency**>  
 <**groupId**>org.springframework.boot</**groupId**>  
 <**artifactId**>spring-boot-starter-test</**artifactId**>  
 <**scope**>test</**scope**>  
 </**dependency**>  
 </**dependencies**>  
  
 <**dependencyManagement**>  
 <**dependencies**>  
 <**dependency**>  
 <**groupId**>org.springframework.cloud</**groupId**>  
 <**artifactId**>spring-cloud-dependencies</**artifactId**>  
 <**version**>${spring-cloud.version}</**version**>  
 <**type**>pom</**type**>  
 <**scope**>import</**scope**>  
 </**dependency**>  
 </**dependencies**>  
 </**dependencyManagement**>  
  
 <**build**>  
 <**plugins**>  
 <**plugin**>  
 <**groupId**>org.springframework.boot</**groupId**>  
 <**artifactId**>spring-boot-maven-plugin</**artifactId**>  
 </**plugin**>  
 </**plugins**>  
 </**build**>  
  
  
</**project**>

Zuul网关依赖：

<**dependency**>  
 <**groupId**>org.springframework.cloud</**groupId**>  
 <**artifactId**>spring-cloud-starter-netflix-zuul</**artifactId**>  
</**dependency**>

配置文件：

**eureka**:  
 **client**:  
 **serviceUrl**:  
 **defaultZone**: http://localhost:8761/eureka/  
**zuul**:  
 **routes**:  
 **hiapi**:  
 **path**: /hiapi/\*\*  
 **serviceId**: lianghua  
 **ribbonapi**:  
 **path**: /ribbonapi/\*\*  
 **serviceId**: eureka-ribbon-client  
 **feignapi**:  
 **path**: /feignapi/\*\*  
 **serviceId**: eureka-feign-client  
 *#prefix: /v1* **max**:  
 **host**:  
 **connections**: 500  
 **host**:  
 **connect-timeout-millis**: 60000  
 **socket-timeout-millis**: 60000  
  
**ribbon**:  
 **eureka**:  
 **enabled**: true  
 **MaxAutoReTries**: 0  
 **MaxAutoRetriesNextServer**: 1  
 **ReadTimeout**: 600000  
 **ConnectTimeout**: 600000  
  
**hystrix**:  
 **command**:  
 **default**:  
 **execution**:  
 **timeout**:  
 **enable**: true  
 **isolation**:  
 **thread**:  
 **timeout-in-milliseconds**: 100000

主启动类：

**package** com.liagn.eurekazuulclient;  
  
**import** org.springframework.boot.SpringApplication;  
**import** org.springframework.boot.autoconfigure.SpringBootApplication;  
**import** org.springframework.cloud.client.discovery.EnableDiscoveryClient;  
**import** org.springframework.cloud.netflix.zuul.EnableZuulProxy;  
**import** org.springframework.context.annotation.Bean;  
  
@SpringBootApplication  
@EnableDiscoveryClient  
@EnableZuulProxy  
  
**public class** EurekaZuulClientApplication {  
  
 **public static void** main(String[] args) {  
 SpringApplication.*run*(EurekaZuulClientApplication.**class**, args);  
 }  
}

在主启动类中加入@EnableZuulProxy注解，开启Zuul功能

自定义熔断器类：UserfallbackProvider

**package** com.liagn.eurekazuulclient;  
  
**import** org.springframework.cloud.netflix.zuul.filters.route.FallbackProvider;  
**import** org.springframework.http.HttpHeaders;  
**import** org.springframework.http.HttpStatus;  
**import** org.springframework.http.MediaType;  
**import** org.springframework.http.client.ClientHttpResponse;  
**import** org.springframework.stereotype.Component;  
  
**import** java.io.ByteArrayInputStream;  
**import** java.io.IOException;  
**import** java.io.InputStream;  
**import** java.util.logging.Logger;  
  
@Component  
**public class** UserfallbackProvider **implements** FallbackProvider {  
  
 **private final** Logger **logger**= Logger.*getLogger*(**"UserfallbackProvider"**);  
 @Override  
 **public** String getRoute(){  
 **return "\*"**;  
 }  
  
 **public** ClientHttpResponse fallbackResponse(){  
 **return new** ClientHttpResponse() {  
 @Override  
 **public** HttpStatus getStatusCode() **throws** IOException {  
 **return** HttpStatus.***OK***;  
 }  
  
 @Override  
 **public int** getRawStatusCode() **throws** IOException {  
 **return** 200;  
 }  
  
 @Override  
 **public** String getStatusText() **throws** IOException {  
 **return "OK"**;  
 }  
  
 @Override  
 **public void** close() {  
  
 }  
  
 @Override  
 **public** InputStream getBody() **throws** IOException {  
 **return new** ByteArrayInputStream(**"service is out of work"**.getBytes());  
 }  
  
 @Override  
 **public** HttpHeaders getHeaders() {  
 HttpHeaders headers=**new** HttpHeaders();  
 headers.setContentType(MediaType.***APPLICATION\_JSON***);  
 **return** headers;  
 }  
 };  
 }  
 @Override  
 **public** ClientHttpResponse fallbackResponse(String route,Throwable cause) {  
 **if** (cause != **null** && cause.getCause() != **null**) {  
 String reason = cause.getCause().getMessage();  
 **logger**.info(**"Excption {}"**+reason);  
 }  
 **return** fallbackResponse();  
 }  
}

记得在自己定义的类的前面加上@Component注解

自定义过滤器：MyFilter

**package** com.liagn.eurekazuulclient;  
  
**import** com.netflix.zuul.ZuulFilter;  
**import** com.netflix.zuul.context.RequestContext;  
**import** org.apache.commons.lang.StringUtils;  
**import** org.springframework.stereotype.Component;  
  
  
**import** javax.servlet.http.HttpServletRequest;  
**import** java.util.logging.Logger;  
  
@Component  
**public class** MyFilter **extends** ZuulFilter {  
  
 **private final** Logger **logger**= Logger.*getLogger*(**"MyFilter"**);  
 @Override  
 **public** String filterType(){  
 **return "pre"**;  
 }  
  
 @Override  
 **public int** filterOrder(){  
 **return** 0;  
 }  
  
 @Override  
 **public boolean** shouldFilter(){  
 **return true**;  
 }  
  
 @Override  
 **public** Object run(){  
 RequestContext ctx=RequestContext.*getCurrentContext*();  
 HttpServletRequest request=ctx.getRequest();  
 **logger**.info(**"-->>> MyFilter: "**+request.getMethod()+**" "**+request.getRequestURL().toString());  
 String token=request.getParameter(**"token"**);*//获取请求的参数* **if**(StringUtils.*isNotBlank*(token)){  
 ctx.setSendZuulResponse(**true**);  
 ctx.setResponseStatusCode(200);  
 ctx.set(**"isSuccess"**,**true**);  
 **return null**;  
 }**else**{  
 ctx.setSendZuulResponse(**false**);  
 ctx.setResponseStatusCode(400);  
 ctx.setResponseBody(**"token is empty"**);  
 ctx.set(**"isSuccess"**,**false**);  
 **return null**;  
 }  
 }  
}

# 配置服务Config

Config server可以从本地仓库读取配置文件，也可以从远处git仓库读取。本地仓库是指将所有配置文件统一下写在Config server目录下，Config Server暴露Http API接口，Config Client通过调用Config Server的Http API接口读取配置文件。

**服务器端：**

POM文件:

*<?***xml version="1.0" encoding="UTF-8"***?>*<**project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd"**>  
 <**modelVersion**>4.0.0</**modelVersion**>  
  
 <**groupId**>com.liagn</**groupId**>  
 <**artifactId**>config-server</**artifactId**>  
 <**version**>0.0.1-SNAPSHOT</**version**>  
 <**packaging**>jar</**packaging**>  
  
 <**name**>config-server</**name**>  
 <**description**>Demo project for Spring Boot</**description**>  
  
 <**parent**>  
 <**groupId**>org.springframework.boot</**groupId**>  
 <**artifactId**>spring-boot-starter-parent</**artifactId**>  
 <**version**>2.0.4.RELEASE</**version**>  
 <**relativePath**/> *<!-- lookup parent from repository -->* </**parent**>  
  
 <**properties**>  
 <**project.build.sourceEncoding**>UTF-8</**project.build.sourceEncoding**>  
 <**project.reporting.outputEncoding**>UTF-8</**project.reporting.outputEncoding**>  
 <**java.version**>1.8</**java.version**>  
 <**spring-cloud.version**>Finchley.SR1</**spring-cloud.version**>  
 </**properties**>  
  
 <**dependencies**>  
 <**dependency**>  
 <**groupId**>org.springframework.cloud</**groupId**>  
 <**artifactId**>spring-cloud-config-server</**artifactId**>  
 </**dependency**>  
  
 <**dependency**>  
 <**groupId**>org.springframework.cloud</**groupId**>  
 <**artifactId**>spring-cloud-starter-netflix-eureka-client</**artifactId**>  
 </**dependency**>  
  
 <**dependency**>  
 <**groupId**>org.springframework.boot</**groupId**>  
 <**artifactId**>spring-boot-starter-test</**artifactId**>  
 <**scope**>test</**scope**>  
 </**dependency**>  
 </**dependencies**>  
  
 <**dependencyManagement**>  
 <**dependencies**>  
 <**dependency**>  
 <**groupId**>org.springframework.cloud</**groupId**>  
 <**artifactId**>spring-cloud-dependencies</**artifactId**>  
 <**version**>${spring-cloud.version}</**version**>  
 <**type**>pom</**type**>  
 <**scope**>import</**scope**>  
 </**dependency**>  
 </**dependencies**>  
 </**dependencyManagement**>  
  
 <**build**>  
 <**plugins**>  
 <**plugin**>  
 <**groupId**>org.springframework.boot</**groupId**>  
 <**artifactId**>spring-boot-maven-plugin</**artifactId**>  
 </**plugin**>  
 </**plugins**>  
 </**build**>  
  
  
</**project**>

依赖为

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

配置文件：

**spring**:  
 **cloud**:  
 **config**:  
 **server**:  
 **native**:  
 **search-locations**: classpath:/shared  
 **profiles**:  
 **active**: native  
 **application**:  
 **name**: config-server  
**server**:  
 **port**: 8762  
**eureka**:  
 **client**:  
 **serviceUrl**:  
 **defaultZone**: http://localhost:8761/eureka/

spring.profiles.active=native配置config server从本地读取配置，读取配置的路径为classpath下的shared目录

主启动文件：

**package** com.liagn.configserver;  
  
**import** org.springframework.boot.SpringApplication;  
**import** org.springframework.boot.autoconfigure.SpringBootApplication;  
**import** org.springframework.cloud.client.discovery.EnableDiscoveryClient;  
**import** org.springframework.cloud.config.server.EnableConfigServer;  
  
@SpringBootApplication  
@EnableConfigServer  
@EnableDiscoveryClient  
**public class** ConfigServerApplication {  
  
 **public static void** main(String[] args) {  
 SpringApplication.*run*(ConfigServerApplication.**class**, args);  
 }  
}

在主启动文件上加上@EnableConfigServer注解，开启Config Server功能

客户端配置文件：

**server**:  
 **port**: 8790  
**foo**: foo version 1

**客户端：**

POM文件：

*<?***xml version="1.0" encoding="UTF-8"***?>*<**project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd"**>  
 <**modelVersion**>4.0.0</**modelVersion**>  
  
 <**groupId**>com.liagn</**groupId**>  
 <**artifactId**>config-client</**artifactId**>  
 <**version**>0.0.1-SNAPSHOT</**version**>  
 <**packaging**>jar</**packaging**>  
  
 <**name**>config-client</**name**>  
 <**description**>Demo project for Spring Boot</**description**>  
  
 <**parent**>  
 <**groupId**>org.springframework.boot</**groupId**>  
 <**artifactId**>spring-boot-starter-parent</**artifactId**>  
 <**version**>2.0.4.RELEASE</**version**>  
 <**relativePath**/> *<!-- lookup parent from repository -->* </**parent**>  
  
 <**properties**>  
 <**project.build.sourceEncoding**>UTF-8</**project.build.sourceEncoding**>  
 <**project.reporting.outputEncoding**>UTF-8</**project.reporting.outputEncoding**>  
 <**java.version**>1.8</**java.version**>  
 <**spring-cloud.version**>Finchley.SR1</**spring-cloud.version**>  
 </**properties**>  
  
 <**dependencies**>  
 <**dependency**>  
 <**groupId**>org.springframework.boot</**groupId**>  
 <**artifactId**>spring-boot-starter-web</**artifactId**>  
 </**dependency**>  
 <**dependency**>  
 <**groupId**>org.springframework.cloud</**groupId**>  
 <**artifactId**>spring-cloud-starter-config</**artifactId**>  
 </**dependency**>  
  
 <**dependency**>  
 <**groupId**>org.springframework.cloud</**groupId**>  
 <**artifactId**>spring-cloud-starter-netflix-eureka-client</**artifactId**>  
 </**dependency**>  
  
 <**dependency**>  
 <**groupId**>org.springframework.boot</**groupId**>  
 <**artifactId**>spring-boot-starter-test</**artifactId**>  
 <**scope**>test</**scope**>  
 </**dependency**>  
 </**dependencies**>  
  
 <**dependencyManagement**>  
 <**dependencies**>  
 <**dependency**>  
 <**groupId**>org.springframework.cloud</**groupId**>  
 <**artifactId**>spring-cloud-dependencies</**artifactId**>  
 <**version**>${spring-cloud.version}</**version**>  
 <**type**>pom</**type**>  
 <**scope**>import</**scope**>  
 </**dependency**>  
 </**dependencies**>  
 </**dependencyManagement**>  
  
 <**build**>  
 <**plugins**>  
 <**plugin**>  
 <**groupId**>org.springframework.boot</**groupId**>  
 <**artifactId**>spring-boot-maven-plugin</**artifactId**>  
 </**plugin**>  
 </**plugins**>  
 </**build**>  
  
  
</**project**>

关键依赖为：

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

配置文件：bootstrap.yml

**spring**:  
 **application**:  
 **name**: config-client  
 **cloud**:  
 **config**:  
 **uri**: http://localhost:8762  
 **fail-fast**: **true  
 profiles**:  
 **active**: dev  
**eureka**:  
 **client**:  
 **serviceUrl**:  
 **defaultZone**: http://localhost:8761/eureka/

配置文件中的变量{spring.application.name}和变量{spring.profiles.active}，两者以“-”相连，构成了向Config Server读取的配置文件。

主启动类：

**package** com.liagn.configclient;  
  
**import** org.springframework.beans.factory.annotation.Value;  
**import** org.springframework.boot.SpringApplication;  
**import** org.springframework.boot.autoconfigure.SpringBootApplication;  
**import** org.springframework.cloud.client.discovery.EnableDiscoveryClient;  
**import** org.springframework.web.bind.annotation.RequestMapping;  
**import** org.springframework.web.bind.annotation.RestController;  
  
@SpringBootApplication  
@RestController  
@EnableDiscoveryClient  
**public class** ConfigClientApplication {  
  
 **public static void** main(String[] args) {  
 SpringApplication.*run*(ConfigClientApplication.**class**, args);  
 }  
  
 @Value(**"${foo}"**)  
 String **foo**;  
 @RequestMapping(value = **"/foo"**)  
 **public** String index(){  
 **return foo**;  
 }  
}