1. Ribbon简介

Eureka服务器实现了微服务的注册，而Ribbon正好实现了微服务实例的负载均衡(轮询的方式)。

2. 使用Ribbon、Feign和Eureka实现微服务实例之间的通信以及负载均衡。

⑴此处写了两个微服务，Service1和Service2。

Service1:

Request:

GET http://localhost:8002/service1

Response:

{

"min": 1,

"max": 99,

"port": 8002

}

Service2:

Request:

GET http://localhost:8003/service2

Response:

{

"port": 8002,

"second": 2

}

⑵微服务在Eureka上注册。

eureka:

client:

service-url:

register-with-eureka: true

fetch-registry: true

default-zone: http://localhost:8761/eureka

⑶使用Feign客户端调用REST服务

①创建代理Poxy接口，接口加注解@FeignClient(name = serviceName)。例子如下：

package com.longmao.service2.proxy;

import com.longmao.service2.dto.Service1Entity;

②在应用程序启动类加注解@EnableFeignClients，如@EnableFeignClients("com.longmao.service2.proxy")

import org.springframework.cloud.netflix.ribbon.RibbonClient;

import org.springframework.cloud.openfeign.FeignClient;

import org.springframework.web.bind.annotation.GetMapping;

@FeignClient(name = "service1")

@RibbonClient(name = "service1")

public interface Service1Proxy {

@GetMapping("/service1")

Service1Entity getMinAndMax();

}

⑷在代理接口中加入注解@RibbonClient(name = serviceName)，如⑶①中@RibbonClient(name = "service1")