# SpringCould[feign]

|  |
| --- |
| @SpringBootApplication  // @EnableDiscoveryClient  @EnableEurekaClient // 表示服务提供者，如果服务中心是eureka的话，那么推荐使用这个注解.@EnableEurekaClient基于spring-cloud-netflix  // 如果是其他的注册中心，那么推荐使用@EnableDiscoveryClient。  @EnableFeignClients // 注解开启Feign的功能  /\*\*  \* Feign的功能 Feign 采用的是基于接口的注解 Feign 整合了ribbon，  \* 具有负载均衡的能力 整合了Hystrix，  \* 具有熔断的能力  \*  \* @author 25772  \*  \*/  public class ServiceFeignApplication {  public static void main(String[] args) {  SpringApplication.run(ServiceFeignApplication.class, args);  }  } |
| /\*\*  \* Created by fangzhipeng on 2017/4/6.  \*/  @RestController  **public** **class** HiController {  @Autowired  SchedualServiceHi schedualServiceHi;      @RequestMapping(value = "/hi",method = RequestMethod.***GET***)  **public** String sayHi(@RequestParam String name){  **return** schedualServiceHi.sayHiFromClientOne(name);  }  } |
| /\*\*  \* **@FeignClient**:表示这个类是一个FeignClient客户端,  \* 具有负载均衡,熔断路由的效果  \* value指定对哪一个服务接口进行调用。  \*  \* **@author** 25772  \*  \*/  @FeignClient(value = "service-hi",fallback = SchedualServiceHiHystric.**class**)  **public** **interface** SchedualServiceHi {  @RequestMapping(value = "/hi",method = RequestMethod.***GET***)  String sayHiFromClientOne(@RequestParam(value = "name") String name);  } |
| /\*\*  \* 服务器短路后的逻辑处理  \*/  @Component  **public** **class** SchedualServiceHiHystric **implements** SchedualServiceHi {  @Override  **public** String sayHiFromClientOne(String name) {  **return** "sorry "+name;  }  } |

## [feign==Ribbon+Hystrix]