# 通过Hystrix实现服务端的服务降级

# 在pom.xml文件中添加依赖

<!--新增hystrix客户端插件-->

<dependency>

<groupId>org.springframework.cloud</groupId>

<artifactId>spring-cloud-starter-netflix-hystrix</artifactId>

</dependency>

# 在启动类中添加注解

@EnableCircuitBreaker

# 在降级的方法上添加注解

@HystrixCommand(fallbackMethod = "selectOneBackup",groupKey = "selectOneBackupThread",threadPoolKey = "selectOneBackupThreadPoolKey",

commandProperties = {

@HystrixProperty(name = "execution.isolation.thread.timeoutInMilliseconds", value = "3000")})

同时定义一个处理服务降级的方法: selectOneBackup

# 案例说明

1. 案例项目的名称: cloud-provider-payment-hystrix-8004

# Hystrix项目使用说明

Hystrix可以用在客户端也可以使用在服务端。

# 全局降级方法

1. 在controller的头部添加如下注解

@DefaultProperties(defaultFallback = "全局降级方法")

备注:方法是没有参数的

案例:

/\*\*

\* 容错的时候使用的方法

\*

\* @return CommonResult<Payment>

\*/

public CommonResult<Payment> selectOneWithFeignHBackupGlobal() {

log.info("服务器繁忙,port:80----》消费端");

return new CommonResult<Payment>(500, "全局方法:服务器繁忙,port:80", new Payment());

}

1. 在降级的方法上面添加如下注解

@HystrixCommand

1. 定制降级说明

如果某个方法想有自己定制的内容，可以在@HystrixCommand上面指定自己的降级的属性和方法，案例如下:

/\*\*

\* 通过主键查询支付单条数据

\*

\* @param id 主键

\* @return 单条数据

\*/

@GetMapping("selectOneWithFeignH")

@HystrixCommand(fallbackMethod = "selectOneWithFeignHBackup", groupKey = "selectOneBackupThread", threadPoolKey = "selectOneBackupThreadPoolKey",

commandProperties = {

@HystrixProperty(name = "execution.isolation.thread.timeoutInMilliseconds", value = "3000")})

public CommonResult<Payment> selectOneWithFeignH(Long id) {

log.info("selectOneWithFeign->id:{}", id);

try {

TimeUnit.SECONDS.sleep(4);

} catch (InterruptedException e) {

e.printStackTrace();

}

return paymentFeignService.selectOne(id);

}