# 客户端项目集成OpenFeign的步骤

## 修改客户端的pom.xml文件

<!-- 增加open feign的依赖 -->

<dependency>

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

<artifactId>spring-cloud-starter-openfeign</artifactId>

</dependency>

## 修改spring boot的主启动类

需要在主启动类上面新增注解: @EnableEurekaClient

如下代码:

package org.jliang.apps.cloud;

import lombok.extern.slf4j.Slf4j;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.cloud.netflix.eureka.EnableEurekaClient;

import org.springframework.cloud.openfeign.EnableFeignClients;

/\*\*

\* 订单消费端

\*

\* @author Lenovo

\*/

@SpringBootApplication

@Slf4j

@EnableEurekaClient

@EnableFeignClients

public class CloudConsumerOrder80Openfeign {

/\*\*

\* 主启动类

\*

\* @param args 命令行参数

\*/

public static void main(String[] args) {

SpringApplication.run(CloudConsumerOrder80Openfeign.class, args);

}

}

## 新增openfeign的客户端接口

package org.jliang.apps.cloud.service;

import org.jliang.apps.cloud.entity.CommonResult;

import org.jliang.apps.cloud.entity.Payment;

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;

/\*\*

\* feign的客户端调用服务端的接口类

\*

\* @author Lenovo

\*/

@Component("paymentFeignService")

@FeignClient(value = "CLOUD-PROVIDER-PAYMENT")

public interface PaymentFeignService {

/\*\*

\* 通过主键查询单条数据

\*

\* @param id 主键

\* @return 单条数据

\*/

@GetMapping("/payment/selectOne")

CommonResult<Payment> selectOne(@RequestParam(value = "id") Long id);

}

备注:

**1、当使用feign传参数的时候,需要加上@RequestParam注解,否则对方服务无法识别参数**

**2、参照项目:** **cloud-consumer-order80-openfeign**

## 在controller层调用feign的接口

下面程序中标红就行调用所需要的代码:

/\*\*

\* 定义paymentFeignService句柄

\*/

@Resource

**private PaymentFeignService paymentFeignService;**

/\*\*

\* 通过主键查询支付单条数据

\*

\* @param id 主键

\* @return 单条数据

\*/

@GetMapping("selectOneWithFeign")

public CommonResult<Payment> selectOneWithFeign(Long id) {

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

return **paymentFeignService.selectOne(id);**

}

## Openfeign超时设置

在application.yml文件添加如下设置：

#设置openfeign的超时时间

feign:

client:

config:

remote-service:

connectTimeout: 2000 #表示超时时间是2秒

readTimeout: 2000

## Openfeign添加详细日志

1、在config文件中添加如下配置:

@Bean

public Logger.Level levelLogger() {

return Logger.Level.FULL;

}

2、在application.yml配置文件添加详细日志打印的feign接口

logging:

level:

org.jliang.apps.cloud.service.PaymentFeignService: debug