# SpringGateWay的配置

# 基本环境配置

1、在pom.xml文件中添加如下配置:

<dependencies>

<dependency>

<groupId>org.projectlombok</groupId>

<artifactId>lombok</artifactId>

<optional>true</optional>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-test</artifactId>

<scope>test</scope>

<exclusions>

<exclusion>

<groupId>org.junit.vintage</groupId>

<artifactId>junit-vintage-engine</artifactId>

</exclusion>

</exclusions>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-devtools</artifactId>

</dependency>

<dependency>

<groupId>org.jliang.apps.cloud</groupId>

<artifactId>cloud\_api\_commons</artifactId>

</dependency>

<!--增加EUREKA的客户端-->

<dependency>

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

<artifactId>spring-cloud-starter-netflix-eureka-client</artifactId>

</dependency>

<!-- spring cloud 的gateway -->

**<dependency>**

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

**<artifactId>spring-cloud-starter-gateway</artifactId>**

**</dependency>**

**</dependencies>**

**主要是上述标红的配置。**

2、需要配置eureka的服务注册，这个首先在配置文件添加如下：

#eueka客户端的配置

eureka:

client:

#表示当前项目要注册到eureka server上面

fetch-registry: true

#是否抓取已注册的信息，true表示抓取

fetchRegistry: true

#eureka server的地址

service-url:

defaultZone: http://eureka7001.com:7001/eureka/,http://eureka7002.com:7002/eureka/

instance:

instance-id: order80

#ip地址显示

prefer-ip-address: true

3、在服务启动类上面添加如下注解:

@SpringBootApplication

@EnableEurekaClient

例如:

package org.jliang.apps.cloud;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

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

/\*\*

\* 网关项目的启动类

\*

\* @author Lenovo

\*/

@SpringBootApplication

@EnableEurekaClient

public class CloudConsumerOrder80GatewayMain {

/\*\*

\* 主启动方法

\*

\* @param args 命令行入参

\*/

public static void main(String[] args) {

SpringApplication.run(CloudConsumerOrder80GatewayMain.class, args);

}

}

基本环境搭建完毕

# 静态路由配置

在application.yml文件添加如下配置:

spring:

application:

name: cloud-consumer-order80-gateway

cloud:

gateway:

routes:

#路由的名字，要求唯一

**- id: payment\_route**

#匹配后提供服务的地址

**uri: http://localhost:8001**

#断言，提供服务的路由

**predicates:**

**- Path=/payment/\*\***

在请求的时候，只需要访问http://localhost:9527/payment/selectOne?id=1

# 动态路由

spring:

application:

name: cloud-consumer-order80-gateway

cloud:

**gateway:**

**discovery:**

**locator:**

**enabled: true #利用服务名进行动态路由**

routes:

#路由的名字，要求唯一

- id: payment\_route

#匹配后提供服务的地址

uri: **lb**://cloud-provider-payment #匹配后提供路由的地址,注意路由配置的开始是由lb:开头的

#断言，提供服务的路由

predicates:

- Path=/payment/\*\*

# 自定义全局过滤器

package org.jliang.apps.cloud.filter;

import lombok.extern.slf4j.Slf4j;

import org.springframework.cloud.gateway.filter.GatewayFilterChain;

import org.springframework.cloud.gateway.filter.GlobalFilter;

import org.springframework.core.Ordered;

import org.springframework.stereotype.Component;

import org.springframework.web.server.ServerWebExchange;

import reactor.core.publisher.Mono;

/\*\*

\* 定义过滤器

\*

\* @author Lenovo

\*/

@Component("myGateWayFilter")

@Slf4j

public class MyGateWayFilter implements **GlobalFilter, Ordered** {

@Override

public Mono<Void> filter(ServerWebExchange exchange, GatewayFilterChain chain) {

log.info("我已经进入自定义的过滤器了");

String id = exchange.getRequest().getQueryParams().getFirst("id");

log.info("我是在过滤器中id={}", id);

return chain.filter(exchange);

}

@Override

public int getOrder() {

return 0;

}

}

备注说明:

需要定义一个实现接口**GlobalFilter, Ordered的类，然后添加spring的组件标签**@Component。

# 参考项目

cloud-consumer-order80-gateway

cloud-consumer-order80-gateway-dynamic

# Spring GateWay总结

Spring GateWay包含三方面主要的内容:

1. 路由
2. 断言
3. 过滤器