

# 8 - Spring Cloud Gateway

Spring Cloud Gateway implements the matching rules of Route routes through Predicate assertions. To put it simply, Predicate is the judgment condition of route forwarding. The request will be forwarded to the specified service for processing only if it meets the condition of Predicate.

Let's use an example to demonstrate how Predicate is used.

1. Create a Spring Boot module named micro-service-cloud-gateway-9527 under the parent project DataEngineSwarm, and introduce relevant dependencies in its pom.xml. The configuration is as follows.

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3     xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
  https://maven.apache.org/xsd/maven-4.0.0.xsd">
4     <modelVersion>4.0.0</modelVersion>
5     <parent>
6         <groupId>com.luxbp</groupId>
7         <artifactId>DataEngineSwarm</artifactId>
8         <version>0.0.1-SNAPSHOT</version>
9     </parent>
10    <groupId>com.luxbp</groupId>
11    <artifactId>micro-service-cloud-gateway-9527</artifactId>
12    <version>0.0.1-SNAPSHOT</version>
13    <name>micro-service-cloud-gateway-9527</name>
14    <description>Demo project for Spring Boot</description>
15    <properties>
16        <java.version>1.8</java.version>
17    </properties>
18    <dependencies>
19        <dependency>
20            <groupId>org.springframework.boot</groupId>
21            <artifactId>spring-boot-starter</artifactId>
22        </dependency>
23        <dependency>
24            <groupId>org.springframework.boot</groupId>
25            <artifactId>spring-boot-starter-test</artifactId>
26            <scope>test</scope>
27        </dependency>
28        <!--Note: don't introduce spring-boot-starter-web into Gateway, otherwise it will
  bring up a error-->
29        <!-- Spring cloud gateway dependency-->
30        <dependency>
31            <groupId>org.springframework.cloud</groupId>
```

```

32         <artifactId>spring-cloud-starter-gateway</artifactId>
33     </dependency>
34     <!--Eureka -->
35     <dependency>
36         <groupId>org.springframework.cloud</groupId>
37         <artifactId>spring-cloud-starter-netflix-eureka-client</artifactId>
38     </dependency>
39     <dependency>
40         <groupId>org.springframework.boot</groupId>
41         <artifactId>spring-boot-devtools</artifactId>
42     </dependency>
43     <dependency>
44         <groupId>org.projectlombok</groupId>
45         <artifactId>lombok</artifactId>
46     </dependency>
47 </dependencies>
48 <build>
49     <plugins>
50         <plugin>
51             <groupId>org.springframework.boot</groupId>
52             <artifactId>spring-boot-maven-plugin</artifactId>
53         </plugin>
54     </plugins>
55 </build>
56 </project>

```

2. In the classpath (/resources directory) of micro-service-cloud-gateway-9527, create a new configuration file application.yml, with the configuration content as follows.

```

1 server:
2   port: 9527
3 spring:
4   application:
5     name: microServiceCloudGateway
6   cloud:
7     gateway:
8       routes:
9         #hide micro-service-cloud-provider-dept-8001
10        - id: provider_dept_list_routh
11          uri: http://localhost:8001
12          predicates:
13            # Predicate
14            - Path=/dept/list/**
15            - Method=GET
16 eureka:
17   instance:
18     instance-id: micro-service-cloud-gateway-9527
19     hostname: micro-service-cloud-gateway
20   client:
21     fetch-registry: true
22     register-with-eureka: true

```

```

23     service-url:
24     defaultZone:
http://eureka7001.com:7001/eureka/,http://eureka7002.com:7002/eureka/,http://eureka7003.com:
7003/eureka/

```

3. On the main startup class of micro-service-cloud-gateway-9527, use the `@EnableEurekaClient` annotation to enable the Eureka client function

By default, Spring Cloud Gateway will create a dynamic route for forwarding based on the service list maintained in the service registry (such as Eureka Server) and the service name (spring.application.name) as the path, so as to realize the dynamic routing function.

We can modify the uri address of Route to the following form in the configuration file.

```
lb://service-name
```

Modify the configuration of application.yml in micro-service-cloud-gateway-9527 and use the microservice name in the registry to create a dynamic route for forwarding. The configuration is as follows.

```

1  server:
2    port: 9527
3  spring:
4    application:
5      name: microServiceCloudGateway
6
7  cloud:
8    gateway:
9      discovery:
10     locator:
11       enabled: true
12     routes:
13       - id: provider_dept_list_routh
14         uri: lb://MICROSERVICECLOUDPROVIDERDEPT # dynamic routing
15         predicates:
16           - Path=/dept/list/**
17           - Method=GET
18  eureka:
19    instance:
20      instance-id: micro-service-cloud-gateway-9527
21      hostname: micro-service-cloud-gateway
22    client:
23      fetch-registry: true
24      register-with-eureka: true
25      service-url:
26      defaultZone:
http://eureka7001.com:7001/eureka/,http://eureka7002.com:7002/eureka/,http://eureka7003.com:
7003/eureka/

```

# Filter

wait to be updated