链滴

Spring Cloud Gateway- 路由谓词工厂详解 (Route Predicate Factories)

作者：jianzh5

原文链接：https://ld246.com/article/1579138532220

来源网站：链滴

Spring Cloud Gateway 有三大核心概念：Route（路由）、Predicate（谓词）、Filter（过滤器）。

本章我们主要来聊聊路由谓词工厂（Route Predicate Factories）。路由谓词工厂的作用是：**符合Predicate的条件，就使用该路由的配置，否则就不管。**

Spring Cloud Gateway 内置十大谓词工厂，包括：

- **After**
- **Before**
- **Between**
- **Cookie**
- **Header**
- **Host**
- **Method**
- **Path**
- **Query**
- **RemoteAddr**

路由配置的两种形式

路由到指定URL

示例1：通配

```
spring:
  cloud:
    gateway:
      routes:
      - id: {唯一标识}
        uri: http://www.javadaily.cn
```

表示访问 GATEWAY_URL/** 会转发到 http://www.javadaily.cn/**

示例2：精确匹配

```
spring:
  cloud:
    gateway:
      routes:
      - id: {唯一标识}
        uri: http://www.javadaily.cn/account-service/
```

表示访问 GATEWAY_URL/account-service/ 会转发到 http://www.javadaily.cn/account-service/

路由到服务发现组件上的微服务

示例1：通配

```
spring:
  cloud:
    gateway:
      routes:
      - id: {唯一标识}
        uri: lb://account-service
```

表示访问 GATEWAY_URL/** 会转发到 account-service 微服务的 /**

示例2：精确匹配

```
spring:
  cloud:
    gateway:
      routes:
      - id: {唯一标识}
        uri: lb://account-service/account/1
```

表示访问 GATEWAY_URL/account/1 会转发到 account-service 微服务的 /account/1

谓词工厂详解

下面正式探讨路由谓词工厂。Spring Cloud Gateway提供了十来种路由谓词工厂。为网关实现灵活转发提供了基石。

After

示例：

```
spring:
  cloud:
    gateway:
      routes:
        - id: after_route
          uri: lb://account-service
          predicates:
            # 当且仅当请求时的时间After配置的时间时，才会转发到用户微服务
            # 目前配置不会进该路由配置，所以返回404
            # 将时间改成 < now的时间，则访问localhost:8040/** -> account-service/**
            # eg. 访问http://localhost:8090/account/1 -> account-service/account/1
            - After=2030-01-20T17:42:47.789-07:00[America/Denver]
```

TIPS

- 技巧：时间可使用 System.out.println(ZonedDateTime.now()); 打印，然后即可看到时区。如：2019-08-10T16:50:42.579+08:00[Asia/Shanghai]
- 时间格式的相关逻辑：

● 默认时间格式：org.springframework.format.support.DefaultFormattingConversionService#addDefaultFormatters

● 时间格式注册：org.springframework.format.datetime.standard.DateTimeFormatterRegistrar#registerFormatters

Before

示例：

```
spring:
  cloud:
    gateway:
      routes:
        - id: before_route
          uri: lb://user-center
          predicates:
            # 当且仅当请求时的时间Before配置的时间时，才会转发到用户微服务
            # 目前配置不会进该路由配置，所以返回404
            # 将时间改成 > now的时间，则访问localhost:8090/** -> account-service/**
            # eg. 访问http://localhost:8090/account/1 -> account-service/account/1
            - Before=2018-01-20T17:42:47.789-07:00[America/Denver]
```

Between

示例：

```
spring:
  cloud:
    gateway:
      routes:
        - id: between_route
          uri: lb://user-center
          predicates:
            # 当且仅当请求时的时间Between配置的时间时，才会转发到用户微服务
            # 因此，访问localhost:8090/** -> account-service/**
            # eg. 访问http://localhost:8090/account/1 -> account-service/account/1
            - Between=2017-01-20T17:42:47.789-07:00[America/Denver], 2027-01-21T17:42:47.78
-07:00[America/Denver]
```

Cookie

示例：

```
spring:
  cloud:
    gateway:
      routes:
        - id: cookie_route
```

```
        uri: lb://account-service
        predicates:
          # 当且仅当带有名为somecookie，并且值符合正则ch.p的Cookie时，才会转发到用户微服务
          # 如Cookie满足条件，则访问http://localhost:8090/** -> account-service/**
          # eg. 访问http://localhost:8090/account/1 -> account-service/account/1
          - Cookie=somecookie, ch.p
```

Header

```
spring:
  cloud:
    gateway:
      routes:
        - id: header_route
          uri: lb://account-service
          predicates:
            # 当且仅当带有名为X-Request-Id，并且值符合正则\d+的Header时，才会转发到用户微服

            # 如Header满足条件，则访问http://localhost:8090/** -> account-service/**
            # eg. 访问http://localhost:8090/account/1 -> account-service/account/1
            - Header=X-Request-Id, \d+
```

Host

```
spring:
  cloud:
    gateway:
      routes:
        - id: host_route
          uri: lb://account-service
          predicates:
            # 当且仅当名为Host的Header符合**.somehost.org或**.anotherhost.org时，才会转发用
微服务
            # 如Host满足条件，则访问http://localhost:8090/** -> account-service/**
            # eg. 访问http://localhost:8090/account/1 -> account-service/account/1
            - Host=**.somehost.org,**.anotherhost.org
```

Method

```
spring:
  cloud:
    gateway:
      routes:
        - id: method_route
          uri: lb://account-service
          predicates:
            # 当且仅当HTTP请求方法是GET时，才会转发用户微服务
            # 如请求方法满足条件，访问http://localhost:8090/** -> account-service/**
            # eg. 访问http://localhost:8090/account/1 -> account-service/account/1
```

```
        - Method=GET
```

Path

```
spring:
  cloud:
    gateway:
      routes:
        - id: path_route
          uri: lb://account-service
          predicates:
            # 当且仅当访问路径是/account/*或者/some-path/**，才会转发用户微服务
            # segment是一个特殊的占位符，单层路径匹配
            # eg. 访问http://localhost:8090/account/1 ->account-service/account/1
            - Path=/account/{segment},/some-path/**
```

Query

示例1：

```
spring:
  cloud:
    gateway:
      routes:
        - id: query_route
          uri: lb://account-service
          predicates:
            # 当且仅当请求带有baz的参数，才会转发到用户微服务
            # eg. 访问http://localhost:8090/account/1?baz=xx -> account-service的/account/1
            - Query=baz
```

示例2：

```
spring:
  cloud:
    gateway:
      routes:
        - id: query_route
          uri: lb://account-service
          predicates:
            # 当且仅当请求带有名为foo的参数，且参数值符合正则ba.，才会转发到用户微服务
            # eg. 访问http://localhost:8090/account/1?baz=baz -> account-service的/account/1?b
z=baz
            - Query=foo, ba.
```

RemoteAddr

示例：

```yaml
spring:
  cloud:
    gateway:
      routes:
        - id: remoteaddr_route
          uri: lb://account-service
          predicates:
            # 当且仅当请求IP是192.168.1.1/24网段，例如192.168.1.10，才会转发到用户微服务
            # eg. 访问http://localhost:8090/account/1 -> account-service的/account/1
            - RemoteAddr=192.168.1.1/24
```

相关代码

原文链接：Spring Cloud Gateway- 路由谓词工厂详解（Route Predicate Factories）