



下载APP



25 | 微服务网关：Gateway 中的路由和谓词有何应用？

2022-02-09 姚秋辰

《Spring Cloud 微服务项目实战》

课程介绍 >

**讲述：姚秋辰**

时长 17:30 大小 16.03M



你好，我是姚秋辰。

在上节课中，我们了解了 Spring Cloud Gateway 网关在微服务架构中的定位，我还介绍了 Gateway 的三大核心组件路由、谓词和过滤器的基本概念。今天，我们就来进一步认识 Gateway 的内置功能，了解在 Gateway 中如何声明一个路由，以及路由中的谓词判断逻辑有什么作用。

领资料

Spring Cloud Gateway（以下简称 Gateway）提供了非常丰富的内置谓词，你可以通过内置谓词来构建复杂的路由条件，甚至连“整点秒杀”这个场景都能在网关层做控制。这些内置谓词就像乐高积木一样，你可以随意组合在自己的业务逻辑中，构建五花八门的网关层判断逻辑。如果这还不够，那么 Gateway 还提供了自定义的谓词工厂扩展点，让你构建自定义谓词。

由于这些个谓词都要附着于一个路由之上，所以在介绍谓词之前，我得先和你聊一下怎么声明一个路由。这一节不涉及微服务项目改造，只是让你能够用最直观的方式体验 Gateway 的功能特点。

声明路由的几种方式

在上一节中我们讲到，路由是 Gateway 中的一条基本转发规则。网关在启动的时候，必须将这些路由规则加载到上下文中，它才能正确处理服务转发请求。那么网关可以从哪些地方加载路由呢？

Gateway 提供了三种方式来加载路由规则，分别是 Java 代码、yaml 文件和动态路由。让我们先来一睹为快，近距离感受一下这三种风格迥异的加载方式。

第一种加载方式是 Java 代码声明路由，它是可读性和可维护性最好的方式，也是我比较喜欢使用的方式。你可以使用一种链式编程的 Builder 风格来构造一个 route 对象，比如在下面的例子里，相信就算我不解释，你也能看明白这段代码做的事情。它声明了两个路由，根据 path 的匹配规则将请求转发到不同的地址。

[复制代码](#)

```
1 @Bean
2 public RouteLocator declare(RouteLocatorBuilder builder) {
3     return builder.routes()
4         .route("id-001", route -> route
5             .path("/geekbang/**")
6             .uri("http://time.geekbang.org")
7         ).route(route -> route
8             .path("/test/**")
9             .uri("http://www.test.com")
10        ).build();
11 }
```

第二种方式是通过配置文件来声明路由，你可以在 application.yml 文件中组装路由规则。我把前面定义的 Java 路由规则改写成了 yaml 版，你可以参考一下。

[复制代码](#)

```
1 spring:
2   cloud:
3     gateway:
4       routes:
```

```
5      - id: id-001
6        uri: http://time.geekbang.org
7        predicates:
8          - Path=/geekbang2/**
9      - uri: http://www.test.com
10       predicates:
11         - Path=/test2/**
```

不管是 Java 版还是 yaml 版，它们都是通过“hardcode”的方式声明的静态路由规则，这些 Route 只会在项目启动后被加载一次。如果你想要在 Gateway 运行期更改路由逻辑，那么就要使用第三种方式：动态路由加载。

动态路由也有不同的实现方式。如果你在项目中集成了 actuator 服务，那么就可以通过 Gateway 对外开放的 actuator 端点在运行期对路由规则做增删改查。但这种修改只是临时性的，项目重新启动后就会被打回原形，因为这些动态规则并没有持久化到任何地方。

动态路由还有另一种实现方式，是我比较推荐的，那就是借助 Nacos 配置中心来存储路由规则。Gateway 通过监听 Nacos Config 中的文件变动，就可以动态获取 Nacos 中配置的规则，并在本地生效了。我将在后面的课程中带你落地一套 Nacos+Gateway 的动态路由。

了解了如何加载路由规则之后，我们再来看一看，有哪些构建在路由之上的、功能丰富的内置谓词吧。

Gateway 的内置谓词都有哪些

Gateway 的内置谓词可真不少，我这里捡一些比较常用的谓词，为你介绍下它们的用法。我把这些谓词大致分为三个类型：寻址谓词、请求参数谓词和时间谓词。我将使用基于 Java 代码的声明方式，带你挨个来看下如何在路由中配置谓词。

寻址谓词，顾名思义，就是针对请求地址和类型做判断的谓词条件。比如这里我们用到的 path，其实就是一个路径匹配条件，当请求的 URL 和 Path 谓词中指定的模式相匹配的时候，这个谓词就会返回一个 True 的判断。而 method 谓词则是根据请求的 Http Method 做为判断条件，比如我这里就限定了只有 GET 和 POST 请求才能访问当前 Route。

 复制代码

```
1 .route("id-001", route -> route
```

```
2     .path("/geekbang/**")
3     .and().method(HttpMethod.GET, HttpMethod.POST)
4     .uri("http://time.geekbang.org")
```

在上面这段代码中，我添加了不少一个谓词。在谓词与谓词之间，你可以使用 `and`、`or`、`negate` 这类“与或非”逻辑连词进行组合，构造一个复杂判断条件。

接下来是**请求参数谓词**，这类谓词主要对服务请求所附带的参数进行判断。这里的参数不单单是 Query 参数，还可以是 Cookie 和 Header 中包含的参数。比如下面这段代码，如果请求中没有包含指定参数，或者指定参数的值和我指定的 regex 表达式不匹配，那么请求就无法满足当前路由的谓词判断条件。

[复制代码](#)

```
1 .route("id-001", route -> route
2     // 验证cookie
3     .cookie("myCookie", "regex")
4     // 验证header
5     .and().header("myHeaderA")
6     .and().header("myHeaderB", "regex")
7     // 验证param
8     .and().query("paramA")
9     .and().query("paramB", "regex")
10    .and().remoteAddr("远程服务地址")
11    .and().host("pattern1", "pattern2")
```

如果你要对原始服务请求的远程地址或 Header 中的 Host 参数做些文章，那么你也可以通过 `remoteAddr` 和 `host` 谓词进行判断。

在实际项目中，非必要情况下，我并不推荐把过多的参数谓词条件定义在网关层，因为这些参数往往携带了业务层的逻辑。如果这些业务参数被大量引入到网关层，从职责分离的角度来讲，并不合适。网关层的逻辑一般来说比较“轻薄”，主要只是一个请求转发，最多再夹带一些简单的鉴权和登录态检查就够了。

最后一组是时间谓词。你可以借助 `before`、`after`、`between` 这三个时间谓词来控制当前路由的生效时间段。

[复制代码](#)

```
1 .route("id-001", route -> route
```

```
2 // 在指定时间之前
3 .before(ZonedDateTime.parse("2022-12-25T14:33:47.789+08:00"))
4 // 在指定时间之后
5 .or().after(ZonedDateTime.parse("2022-12-25T14:33:47.789+08:00"))
6 // 或者在某个时间段以内
7 .or().between(
8     ZonedDateTime.parse("起始时间"),
9     ZonedDateTime.parse("结束时间"))
```

拿一项秒杀活动来说，如果开发团队做了一个新的秒杀下单入口，我要限定该入口的生效时间在秒杀时间点之后，那么我就可以使用 `after` 谓词。对于固定时间窗口的秒杀活动来说，你还可以使用 `between` 来限定生效时间窗口。再结合前面我们讲到的请求参数谓词，你还可以实现更加复杂的路由判断逻辑，比如通过 `query` 谓词针对特定商品开放不同的秒杀时段。

如果 Gateway 的内置谓词还差那么点意思，你想要实现自定义的谓词逻辑，那么你可以通过 Gateway 的可扩展谓词工厂来实现自定义谓词。Gateway 组件提供了一个统一的抽象类 `AbstractRoutePredicateFactory` 作为谓词工厂，你可以通过继承这个类来添加新的谓词逻辑。

我把实现一个自定义谓词的代码框架放到了这里，你可以参考一下。

[复制代码](#)

```
1 // 继承自通用扩展抽象类AbstractRoutePredicateFactory
2 public class MyPredicateFactory extends
3     AbstractRoutePredicateFactory<MyPredicateFactory.Config> {
4
5     public MyPredicateFactory() {
6         super(Config.class);
7     }
8
9     // 定义当前谓词所需要用到的参数
10    @Validated
11    public static class Config {
12        private String myField;
13    }
14
15    @Override
16    public List<String> shortcutFieldOrder() {
17        // 声明当前谓词参数的传入顺序
18        // 参数名要和Config中的参数名称一致
19        return Arrays.asList("myField");
20    }
21}
```



```
22 // 实现谓词判断的核心方法
23 // Gateway会将外部传入的参数封装为Config对象
24 @Override
25 public Predicate<ServerWebExchange> apply(Config config) {
26     return new GatewayPredicate() {
27
28         // 在这个方法里编写自定义谓词逻辑
29         @Override
30         public boolean test(ServerWebExchange exchange) {
31             return true;
32         }
33
34         @Override
35         public String toString() {
36             return String.format("myField: %s", config.myField);
37         }
38     };
39 }
40 }
```

这个实现的过程非常简单，相信看了上面的源码就能明白。这里面的关键步骤就两步，一是定义 Config 结构来接收外部传入的谓词参数，二是实现 apply 方法编写谓词判断逻辑。我将会留一道课后作业让你自己动手实现一个专属谓词。

到这里，我们就了解了 Gateway 的路由和谓词是如何完成请求转发的。接下来我来带你回顾一下这一节的重点内容吧。

总结

今天我们了解了 Gateway 中声明路由的三种不同方式。对于静态路由来说，我推荐你使用可读性更强的 Java 代码方式来配置路由；至于动态路由呢，就等到后面的课程，我再教你如何使用 Nacos 定义 JSON 格式动态路由吧。

但是这里要注意的是，一般来讲，路由规则是不受开发团队控制的。暴露什么 URL 给到外部网关，那可是涉及到安全性的一个决策，在大厂中，所有的对外接口都要经过严格的漏扫和渗透测试，然后再经由相关团队审批才能上线路由规则。

在实际工作中，最最常用的谓词当属 path，其它大部分内置谓词都用不太上，如果你想要使用这些谓词在网关层判断登录状态或者做权限验证，那么我更推荐你使用 Gateway 的

Filter 机制，也就是过滤器。我在下节课将基于 Gateway 限流的场景，跟你讲一下如何在路由规则中添加 Filter。

思考题

结合这节课的内容，你能自己写一个自定义谓词实现某个简单逻辑吗？比如说恶搞的“春节炸弹”，在春节这一天将所有请求转发到一个特定的 URL（不要使用 between 谓词来实现）。这里你需要思考一个问题，如果某个请求同时满足两个路由的判断条件，如何设置其中一个路由先行生效。

好啦，这节课就结束啦。欢迎你把这节课分享给更多对 Spring Cloud 感兴趣的朋友。我是姚秋辰，我们下节课再见！

分享给需要的人，Ta 购买本课程，你将得 20 元

 生成海报并分享

 赞 1  提建议

© 版权归极客邦科技所有，未经许可不得传播售卖。页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

上一篇 24 | 为什么微服务架构少不了微服务网关？

下一篇 26 | 微服务网关：如何设置请求转发、跨域和限流规则？

精选留言 (2)

 写留言



Charles

2022-02-10

请教老师，是否推荐使用谓词实现金丝雀测试方案？



peter

2022-02-09

请教老师几个问题啊：

Q1：SpringCloud Gateway和zuul相比哪个更好

Q2：Gateway只对用户的请求起作用吗？对微服务之间的请求起作用吗？

Q3：gateway路由的具体效果是什么？

比如这个路由：...

展开 ✓

