12.1 简介

特点：

- 限流
- 路径重写
- 动态路由
- 集成 Spring Cloud DiscoveryClient
- 集成 Hystrix 断路器

和 Zuul 对比：

1. Zuul 是 Netflix 公司的开源产品，Spring Cloud Gateway 是 Spring 家族中的产品，可以和 Spring 家族中的其他组件更好的融合。
2. Zuul1 不支持长连接，例如 websocket。
3. Spring Cloud Gateway 支持限流。
4. Spring Cloud Gateway 基于 Netty 来开发，实现了异步和非阻塞，占用资源更小，性能强于 Zuul。

12.2 基本用法

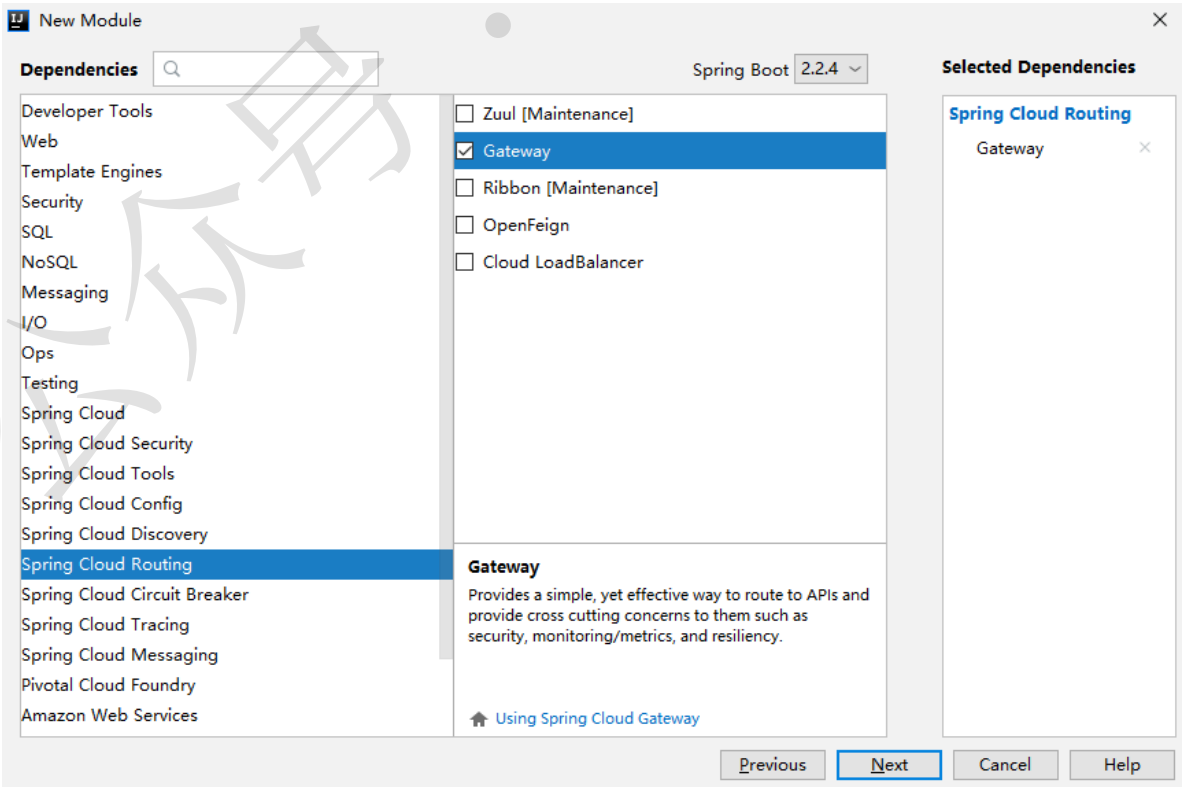
Spring Cloud Gateway 支持两种不同的用法：

- 编码式
- yml 配置

两种都来看下。

编码式

首先创建 Spring Boot 项目，添加 Spring Cloud Gateway 模块：



项目创建成功后，直接配置一个 RouteLocator 这样一个 Bean，就可以实现请求转发。

```
@Bean
RouteLocator routeLocator(RouteLocatorBuilder builder) {
    return builder.routes()
            .route("javaboy_route", r ->
r.path("/get").uri("http://httpbin.org"))
            .build();
}
```

这里只需要提供 RouteLocator 这个 Bean，就可以实现请求转发。配置完成后，重启项目，访问：http://localhost:8080/get

properties 配置

```
spring.cloud.gateway.routes[0].id=javaboy_route
spring.cloud.gateway.routes[0].uri=http://httpbin.org
spring.cloud.gateway.routes[0].predicates[0]=Path=/get
```

YML 配置

```
spring:
  cloud:
    gateway:
      routes:
        - id: javaboy_route
          uri: http://httpbin.org
          predicates:
            - Path=/get
```

12.2.1 服务化

首先给 Gateway 添加依赖，将之注册到 Eureka 上。

加依赖：

```
<dependency>
    <groupId>org.springframework.cloud</groupId>
    <artifactId>spring-cloud-starter-netflix-eureka-client</artifactId>
</dependency>
```

加配置：

```
spring:
  cloud:
    gateway:
      routes:
        - id: javaboy_route
          uri: http://httpbin.org
          predicates:
            - Path=/get
  application:
    name: gateway
eureka:
  client:
    service-url:
      defaultZone: http://localhost:1111/eureka
```

配置路由转发：

```
spring:
  cloud:
    gateway:
      routes:
        - id: javaboy_route
          uri: http://httpbin.org
          predicates:
            - Path=/get
      discovery:
        locator:
          enabled: true # 开启自动代理
  application:
    name: gateway
eureka:
  client:
    service-url:
      defaultZone: http://localhost:1111/eureka
logging:
  level:
    org.springframework.cloud.gateway: debug
```

接下来，就可以通过 Gateway 访问到其他注册在 Eureka 上的服务了，访问方式和 Zuul 一样。

12.3 Predicate

通过时间匹配：

```
spring:
  cloud:
    gateway:
      routes:
        - id: javaboy_route
          uri: http://httpbin.org
          predicates:
            - After=2021-01-01T01:01:01+08:00[Asia/Shanghai]
```

表示，请求时间在 2021-01-01T01:01:01+08:00[Asia/Shanghai] 时间之后，才会被路由。

除了 After 之外，还有两个关键字：

- Before，表示在某个时间点之前进行请求转发
- Between，表示在两个时间点之间，两个时间点用 , 隔开

也可以通过请求方式匹配，就是请求方法：

```
spring:
  cloud:
    gateway:
      routes:
        - id: javaboy_route
          uri: http://httpbin.org
          predicates:
            - Method=GET
```

这个配置表示只给 GET 请求进行路由。

通过请求路径匹配：

```
spring:
  cloud:
    gateway:
      routes:
        - id: javaboy_route
          uri: http://www.javaboy.org
          predicates:
            - Path=/2019/0612/{segment}
```

表示路径满足 /2019/0612/ 这个规则，都会被进行转发，例如：

http://www.javaboy.org/2019/0612/git-install.html
http://www.javaboy.org/2019/0612/git-basic.html

通过参数进行匹配：

```
spring:
  cloud:
    gateway:
      routes:
        - id: javaboy_route
          uri: http://httpbin.org
          predicates:
            - Query=name
```

表示请求中一定要有 name 参数才会进行转发，否则不会进行转发。

也可以指定参数和参数的值。

例如参数的 key 为 name，value 必须要以 java 开始：

```
spring:
  cloud:
    gateway:
      routes:
        - id: javaboy_route
          uri: http://httpbin.org
          predicates:
            - Query=name,java.*
```

多种匹配方式也可以组合使用。

```
spring:
  cloud:
    gateway:
      routes:
        - id: javaboy_route
          uri: http://httpbin.org
          predicates:
            - Query=name,java.*
            - Method=GET
            - After=2021-01-01T01:01:01+08:00[Asia/Shanghai]
```

12.4 Filter

Spring Cloud Gateway 中的过滤器分为两大类：

- GatewayFilter
- GlobalFilter

AddRequestParameter 过滤器使用：

```
spring:
  cloud:
    gateway:
      routes:
        - id: javaboy_route
          uri: lb://provider
          filters:
            - AddRequestParameter=name,javaboy
          predicates:
            - Method=GET
```

这个过滤器就是在请求转发路由的时候，自动额外添加参数。