**[#](https://pig4cloud.com/" \l "spring-cloud-gateway) Spring Cloud Gateway**

Pig 2.0 采用的是spring官方的网关组件，通过异步背压的高性能网关。 路由配置是整个微服务中最为核心的功能

[**#**](https://pig4cloud.com/#%E9%85%8D%E7%BD%AE%E8%B7%AF%E7%94%B1) **配置路由**

我们以UPMS 的路由为例子，注意注释

spring:

cloud:

gateway:

locator:

enabled: true

routes:

#UPMS 模块

- id: pig-upms # 唯一的服务ID

uri: lb://pig-upms # 注册中心的服务名称，实现负载均衡

predicates:

- Path=/admin/\*\* #所有业务的请求前缀

filters:

# 限流配置

- name: RequestRateLimiter #限流策略

args:

key-resolver: '#{@remoteAddrKeyResolver}'

redis-rate-limiter.replenishRate: 10

redis-rate-limiter.burstCapacity: 20

# 降级配置

- name: Hystrix #断路器降级策略

args:

name: default

fallbackUri: 'forward:/fallback'

[**#**](https://pig4cloud.com/#pig-%E9%BB%98%E8%AE%A4%E6%8F%90%E4%BE%9B%E4%BA%86%E5%85%A8%E5%B1%80%E7%9A%84%E8%B7%AF%E7%94%B1%E8%BF%87%E6%BB%A4%E5%99%A8%E5%8E%9F%E7%90%86) **pig 默认提供了全局的路由过滤器原理**

PigRequestGlobalFilter,对全部的微服务提供了安全过滤（这个后边会讲）和全局StripPrefix=1配置，**意味着你在使用Pig的时候，网关转发到业务模块时候会自动截取前缀，不用再每个微服务路由配置了StripPrefixFilter**

public class PigRequestGlobalFilter implements GlobalFilter, Ordered {

private static final String HEADER\_NAME = "X-Forwarded-Prefix";

@Override

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

// 1. 清洗请求头中from 参数

ServerHttpRequest request = exchange.getRequest().mutate()

.headers(httpHeaders -> httpHeaders.remove(SecurityConstants.FROM))

.build();

// 2. 重写StripPrefix

addOriginalRequestUrl(exchange, request.getURI());

String rawPath = request.getURI().getRawPath();

String newPath = "/" + Arrays.stream(StringUtils.tokenizeToStringArray(rawPath, "/"))

.skip(1L).collect(Collectors.joining("/"));

ServerHttpRequest newRequest = request.mutate()

.path(newPath)

.build();

exchange.getAttributes().put(GATEWAY\_REQUEST\_URL\_ATTR, newRequest.getURI());

return chain.filter(exchange.mutate()

.request(newRequest.mutate()

.build()).build());

}

@Override

public int getOrder() {

return -1000;

}

}