**网关降级**

当网关流量向后转发的时候，如果微服务模块不可用，则会触发降级事件

[**#**](https://pig4cloud.com/#pig-%E4%B8%AD%E7%9A%84%E4%BD%BF%E7%94%A8%E8%AE%B2%E8%A7%A3%EF%BC%8C%E4%BB%A5upms%E8%B7%AF%E7%94%B1%E9%85%8D%E7%BD%AE%E4%B8%BA%E4%BE%8B) **Pig 中的使用讲解，以UPMS路由配置为例**

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:

- name: Hystrix #断路器降级策略

args:

name: default

fallbackUri: 'forward:/fallback' # 降级接口的地址

[**#**](https://pig4cloud.com/#%E5%8E%9F%E7%90%86) **原理**

Spring Cloud Gateway 会自动寻找配置Hystrix的Filter，这个功能是内置的，然后回调我们提供fallbackUri,Pig 中的降级接口是使用weblflux 实现的

* 降级业务很简单，输出那个接口的异常日志

@Slf4j

@Component

public class HystrixFallbackHandler implements HandlerFunction<ServerResponse> {

@Override

public Mono<ServerResponse> handle(ServerRequest serverRequest) {

Optional<Object> originalUris = serverRequest.attribute(GATEWAY\_ORIGINAL\_REQUEST\_URL\_ATTR);

originalUris.ifPresent(originalUri -> log.error("网关执行请求:{}失败,hystrix服务降级处理", originalUri));

return ServerResponse.status(HttpStatus.INTERNAL\_SERVER\_ERROR.value())

.contentType(MediaType.TEXT\_PLAIN).body(BodyInserters.fromObject("服务异常"));

}

}

* 降级入口。这里的意思类似于SpringMVC 定义一个 @GetMapping("/fallback") 接口

@Slf4j

@Configuration

@AllArgsConstructor

public class RouterFunctionConfiguration {

private final HystrixFallbackHandler hystrixFallbackHandler;

private final ImageCodeHandler imageCodeHandler;

@Bean

public RouterFunction routerFunction() {

return RouterFunctions.route(

RequestPredicates.path("/fallback")

.and(RequestPredicates.accept(MediaType.TEXT\_PLAIN)), hystrixFallbackHandler)

}

}