验证码直接网关异步生成，基于webflux

[**#**](https://pig4cloud.com/#webflux-%E7%94%9F%E6%88%90%E9%AA%8C%E8%AF%81%E7%A0%81%E7%9A%84handler) **webflux 生成验证码的handler**

public class ImageCodeHandler implements HandlerFunction<ServerResponse> {

private final Producer producer;

private final RedisTemplate redisTemplate;

@Override

public Mono<ServerResponse> handle(ServerRequest serverRequest) {

//生成验证码

String text = producer.createText();

BufferedImage image = producer.createImage(text);

//保存验证码信息

String randomStr = serverRequest.queryParam("randomStr").get();

redisTemplate.opsForValue().set(CommonConstants.DEFAULT\_CODE\_KEY + randomStr, text, 60, TimeUnit.SECONDS);

// 转换流信息写出

FastByteArrayOutputStream os = new FastByteArrayOutputStream();

try {

ImageIO.write(image, "jpeg", os);

} catch (IOException e) {

log.error("ImageIO write err", e);

return Mono.error(e);

}

return ServerResponse

.status(HttpStatus.OK)

.contentType(MediaType.IMAGE\_JPEG)

.body(BodyInserters.fromResource(new ByteArrayResource(os.toByteArray())));

}

}

[**#**](https://pig4cloud.com/#webflux-%E8%AF%B7%E6%B1%82%E5%A4%84%E7%90%86%E5%85%A5%E5%8F%A3) **webflux 请求处理入口**

public class RouterFunctionConfiguration {

private final HystrixFallbackHandler hystrixFallbackHandler;

private final ImageCodeHandler imageCodeHandler;

@Bean

public RouterFunction routerFunction() {

return RouterFunctions.route(RequestPredicates.GET("/code")

.and(RequestPredicates.accept(MediaType.TEXT\_PLAIN)), imageCodeHandler);

}

}

[**#**](https://pig4cloud.com/#%E6%A0%A1%E9%AA%8C%E9%80%BB%E8%BE%91%2C%E9%80%9A%E8%BF%87oauth2-%E7%BB%88%E7%AB%AF%E7%9A%84client-id-%E6%9D%A5%E7%A1%AE%E5%AE%9A%E6%98%AF%E5%90%A6%E6%A0%A1%E9%AA%8C%E9%AA%8C%E8%AF%81%E7%A0%81) **校验逻辑,通过oauth2 终端的client-id 来确定是否校验验证码**

public class ValidateCodeGatewayFilter extends AbstractGatewayFilterFactory {

@Override

public GatewayFilter apply(Object config) {

return (exchange, chain) -> {

ServerHttpRequest request = exchange.getRequest();

// 终端设置不校验， 直接向下执行

String[] clientInfos = WebUtils.getClientId(request);

if (filterIgnorePropertiesConfig.getClients().contains(clientInfos[0])) {

return chain.filter(exchange);

}

//校验验证码

checkCode(request);

return chain.filter(exchange);

};

}

}

[**#**](https://pig4cloud.com/#%E9%85%8D%E7%BD%AE%E7%BB%88%E7%AB%AF%E4%B8%8D%E6%A0%A1%E9%AA%8C%E9%AA%8C%E8%AF%81%E7%A0%81) **配置终端不校验验证码**

pig-gateway-dev.yml

# 不校验验证码终端

ignore:

clients:

- 不校验验证码的终端client-id