

springcloud-gateway

Gateway Handler Mapping

Gateway web handler

filter

① 宏观处理流程

初始化

子主题 3

功能描述

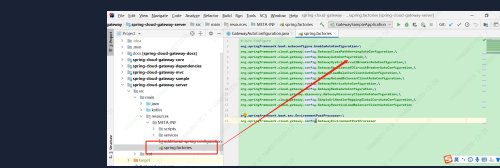
引入spring-cloud-starter-gateway包

GatewayAutoConfiguration (存在spring.factories中)

其中有工程spring-cloud-gateway-server, classpath中找所有spring.factories文件

classpath中找所有spring.factories文件

springboot启动注解@SpringBootApplication包含三个注解, 其中核心注解@EnableAutoConfiguration



从classpath中找所有spring.factories文件, 并将其中的配置项信息的类, 加载类上标注@Configuration通过反射实例化类, 然后汇总并加载到IOC容器中

RouteLocatorBuilder RouteLocator的建造者, 使用了建造者设计模式

PropertiesRouteDefinitionLocator 属性路由定义定位器. 通过getRouteDefinitions可获取RouteDefinition的list

GatewayProperties 从配置文件(例如, yml / Properties 等)

对 RouteDefinition的save和delete方法, 供GatewayWebfluxEndpoint调用, 保留接口对外

save方法: 保存线程安全的map中 Map<String, RouteDefinition> routes = synchronizedMap(new LinkedHashMap<String, RouteDefinition>());

使用 InMemoryRouteDefinitionRepository 来维护 RouteDefinition 信息, 在网关实例重启或者崩溃后, 数据就没有了, 此时我们可以实现 RouteDefinitionRepository 接口, 以实现例如MySQLRouteDefinitionRepository, 通过类似MySQL 等持久化、可共享的存储, 也可以带来 Spring Cloud Gateway 实例集群获得一致的、相同的 RouteDefinition 信息

RouteLocator 路由定位器 多个实现类

routeDefinitionLocator 路由配置信息赋值

configurationService 基础配置信息

RoutePredicateFactory 断言工厂类 子主题 1

GatewayFilterFactory 过滤器添加

gatewayProperties

loadFilters这个方法把上面这段代码收集了所有GlobalFilter其中包含框架默认提供的也包含我们自定义的, 进入到new方法可以看到在初始化的时候会根据order的值来进行排序

HandlerMapping接口的实现

FilteringWebHandler路由处理类

RouteLocator 路由定位器

GlobalCorsProperties

Environment环境变量信息

核心方法: lookupRoute 获取路由信息, 并将获取到的route放入当前请求上下文的属性中

WebSocketService

NettyConfiguration

netty相关 NettyWriteResponseFilter

子主题 3

HttpClient初始化

各种PredicateFactory

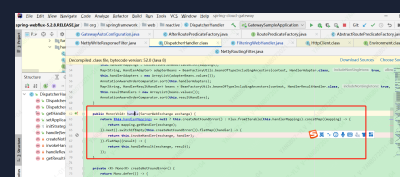
handler是gateway的核心控制部分, 其控制着request在gateway的整个生命周期, gateway是通过Gateway Web Handler来与web框架交接工作的

向后输入FilteringWebHandler 即gateway web handler

传递请求上下文ServerWebExchange

核心FilteringWebHandler 使用Filter链表处理请求的WebHandler, FilteringWebHandler负责组装Filter链表并调用链表处理请求

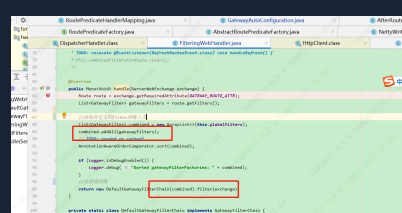
所有请求的调度器, 负责请求分发



DispatcherHandler DispatcherHandler 首先判断持有的handlerMappings是否为空, 为空则调用 createNotFoundError方法, 返回 No matching handler信息, 接着有序遍历 handlerMappings并调用 mapping的getHandler方法返回对应的处理handler, DispatcherHandler持有HandlerMapping列表如下

通过上面的mapping处理调用到RoutePredicateHandlerMapping的getHandlerInternal getHandlerInternal处理断言匹配逻辑

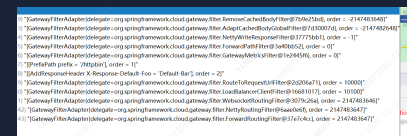
接着循环SimpleHandlerAdapter处理传入的到FilteringWebHandler的handle



RoutePredicateHandlerMapping: 路由谓词匹配器, 用于路由的查找, 以及找到路由后返回对应的WebHandler, RoutePredicateHandlerMapping处理将返回FilteringWebHandler



debug下



分类

GlobalFilter 全局的, 每个请求都会过滤

GatewayFilter 可实现接口自定义

将应用到单个路由或者一个分组的路由上

AdaptCachedBodyGlobalFilter 从exchange的attributes中获取cachedRequestBody属性值作为request的body

ForwardPathFilter

ForwardRoutingFilter 转发路由网关过滤器。其根据 forward:// 前缀( Scheme )过滤处理, 将请求转发到当前网关实例本地接口

GatewayMetricsFilter

LoadBalancerClientFilter 配置中的lb://{serviceId}前缀 替换服务的真实ip+port, 默认采用的netflix-ribbon作为负载均衡器

NettyRoutingFilter

WebClientWriteResponseFilter

通过HttpClient转发请求到真是的服务并将响应写回当前的响应中

OrderedGatewayFilter

ReactiveLoadBalancerClientFilter 获取服务实例, 然后从exchange中解析出uri, 拼接出requestUrl, 设置到属性

RemoveCachedBodyFilter

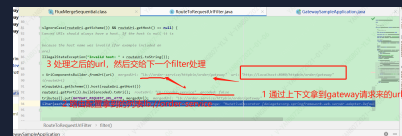
RouteToRequestUrlFilter

WebClientHttpRequestFilter 构造完整的负载均衡地址, 例如route配置的中转服务是lb://MY-SERVICE, 请求的路径是/hello/world, 则构建后的地址是lb://MY-SERVICE/hello/world

WebClientWriteResponseFilter

WebsocketRoutingFilter

WeightCalculatorWebFilter 过滤器实现了gateway对于websocket的支持, 内部通过websocketClient实现将一个http请求协议转换成websocket, ws://



通过各种GatewayFilter Factory生成

