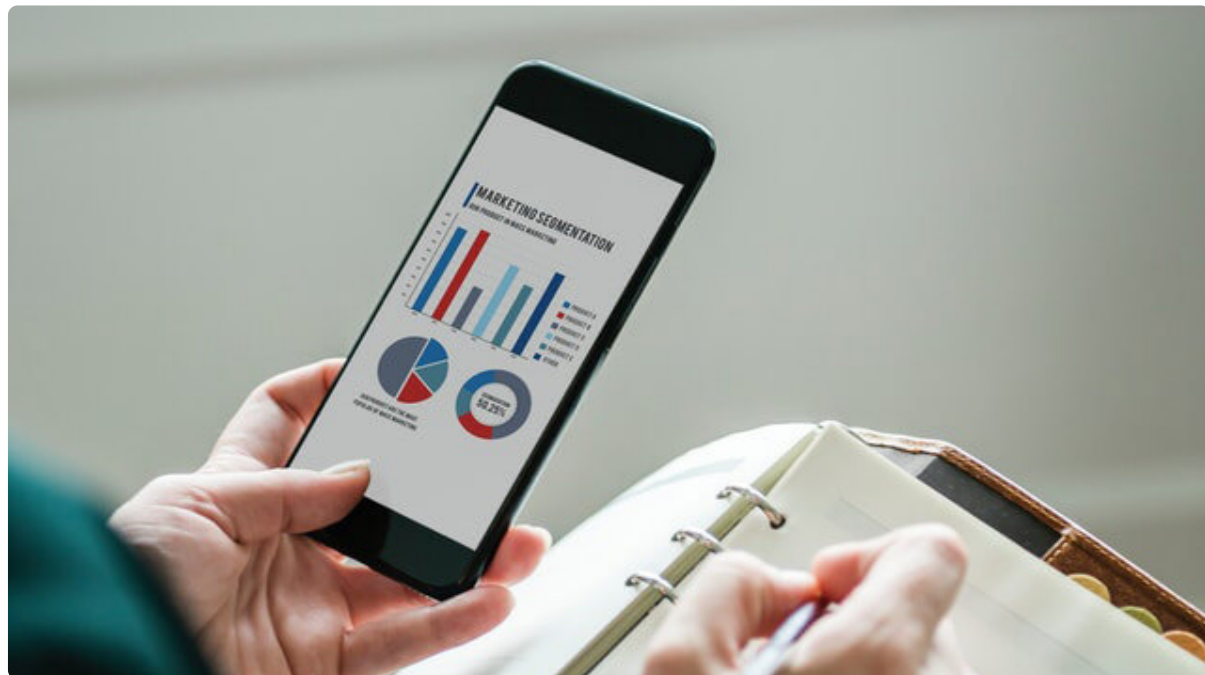


19 服务网关 Zuul 和 Spring Cloud Gateway

更新时间：2019-07-01 09:52:07



先相信你自己，然后别人才会相信你。

——屠格涅夫

从这节课开始我们学习服务网关——学习什么是服务网关，在微服务架构中服务网关会起到什么样的作用，以及 Spring Cloud 体系内都有哪些服务网关技术，分别都有哪些特点？

什么是服务网关

在微服务架构中，整个系统是由许多微服务组成，每个微服务负责一块独立的业务，当外部系统需要调用微服务架构中的某些功能时，我们如何对外提供接口。

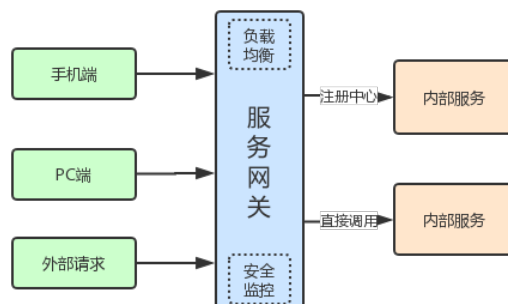
微服务架构中，服务实例地址经常会发生变化，所以我们无法直接将服务地址外放出来。如果每一个微服务都直接外放接口，会导致一系列的其它问题，比如调用链过于复杂、涉及到账户、权限时不能统一处理等。另外基于高内聚低耦合的设计准则来讲，我们也应该将内部系统和外部系统做切割。

于是在这样的场景下，就需要有一个独立的组件来处理外部的请求，这个组件就是服务网关。服务网关是为了简化前端的调用逻辑，通常情况下也会实现相关的认证逻辑，根据外部不同的请求响应不同的数据，从而简化内外部系统之间调用的复杂度。

总结一下服务网关的作用：

- 简化客户端调用复杂度，统一处理外部请求。
- 数据裁剪以及聚合，根据不同的接口需求，对数据加工后对外。
- 多渠道支持，针对不同的客户端提供不同的网关支持。
- 遗留系统的微服务化改造，可以作为新老系统的中转组件。
- 统一处理调用过程中的安全、权限问题。

下图为服务网关调用示意图：



服务网关负责服务请求路由、组合及协议转换。客户端的所有请求都首先经过服务网关，然后由它将请求路由到合适的微服务。服务网关经常会通过调用多个微服务并合并结果来处理一个请求，它可以在系统外部与内部响应之间友好的转换。

Spring Cloud 体系内的网关技术

Spring Cloud 体系内的网关技术主要有 Zuul 和 Spring Cloud Gateway。Zuul 是 Netflix 公司开源的产品，也是 Spring Cloud 前几个版本默认使用的服务网关组件，但是随着 Netflix 公司一系列的停更事件，在最新的 Spring Cloud Greenwich 版本中已经不建议采用 Zuul 技术，官方建议使用 Spring Cloud Gateway 作为默认的网关技术。

接下来详细介绍：

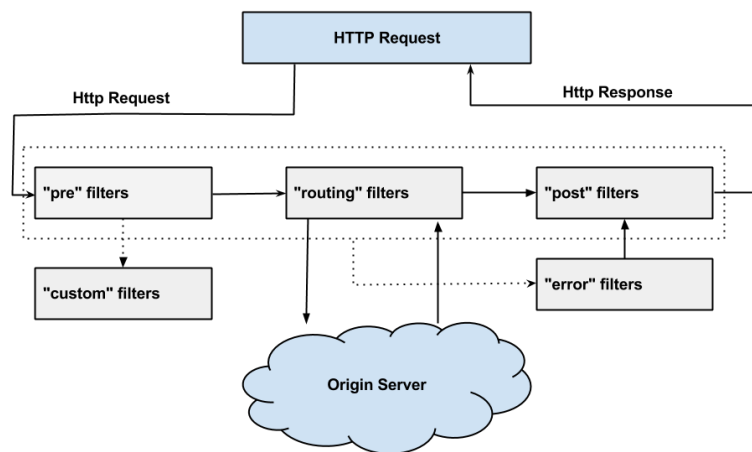
Zuul

Zuul 是 Netflix 开发的一款提供动态路由的网关服务。Zuul 接收所有设备或者网站请求，根据需求将请求转发到对应的后端系统。作为一个前置服务，Zuul 旨在实现动态路由，监控，弹性和安全性。

Zuul 提供了不同类型的 filter 用于处理请求，这些 filter 可以实现以下功能：

- 权限控制和安全性：可以识别认证需要的信息和拒绝不满足条件的请求。
- 监控：与边缘位置追踪有意义的数据和统计结果，从而带来精确的生产视图。
- 动态路由：根据需要动态地路由请求到后台的不同集群。
- 压力测试：逐渐增加指向集群的流量，以了解性能。
- 负载均衡：为每一种负载类型分配对应容量，并弃用超出限定值的请求。
- 静态资源处理：直接在 zuul 处理静态资源的响应，从而避免其转发到内部集群。

Filter 是 Zuul 的核心，用来实现对外服务的控制。Filter 的生命周期有4个，分别是“PRE”、“ROUTING”、“POST”、“ERROR”，整个生命周期可以用下图来表示。



图片来自 Netflix 博客: <https://medium.com/netflix-techblog/announcing-zuul-edge-service-in-the-cloud-ab3af5be08ee>

Zuul 大部分功能都是通过过滤器来实现的，这些过滤器类型对应于请求的典型生命周期。

- **PRE:** 这种过滤器在请求被路由之前调用。可利用这种过滤器实现身份验证、在集群中选择请求的微服务、记录调试信息等。
- **ROUTING:** 这种过滤器将请求路由到微服务。这种过滤器用于构建发送给微服务的请求，并使用 Apache HttpClient 或 Netflix Ribbon 请求微服务。
- **POST:** 这种过滤器在路由到微服务以后执行。这种过滤器可用来为响应添加标准的 HTTP Header、收集统计信息和指标、将响应从微服务发送给客户端等。
- **ERROR:** 在其他阶段发生错误时执行该过滤器。除了默认的过滤器类型，Zuul 还允许我们创建自定义的过滤器类型。例如，我们可以定制一种 STATIC 类型的过滤器，直接在 Zuul 中生成响应，而不将请求转发到后端的微服务。

在实际工作中可以根据业务的使用场景，来选择使用哪种类型的过滤器。

由于 Zuul2 不断推迟发布、以及 Netflix 内部对开源软件支持的不稳定性，Spring Cloud 自行研发了另外一款服务网关产品 Spring Cloud GateWay，并且在最新版本中推荐使用 Spring Cloud GateWay，本系列课程重点介绍 Spring Cloud GateWay 的使用。

Spring Cloud Gateway

Spring Cloud Gateway 是 Spring 体系内的一个全新项目，该项目是基于 Spring 5.0, Spring Boot 2.0 和 Project Reactor 等技术开发，它旨在为微服务架构提供一种简单有效的统一的 API 路由管理方式。

Spring Cloud Gateway 作为 Spring Cloud 生态系统中的网关，目标是替代 Netflix Zuul，其不仅提供统一的路由方式，并且基于 Filter 链的方式提供了网关基本的功能，例如：安全、监控/指标和限流。

特性

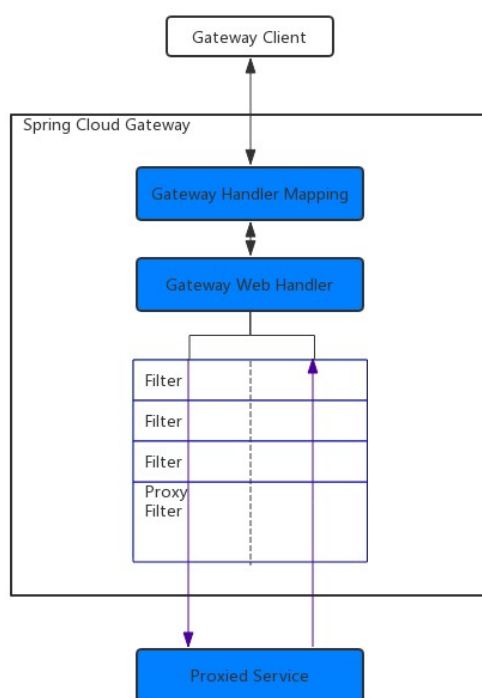
- 基于 Spring Framework 5, Project Reactor 和 Spring Boot 2.0
- 动态路由

- Predicates 和 Filters 作用于特定路由
- 集成 Hystrix 断路器
- 集成 Spring Cloud DiscoveryClient
- 易于编写的 Predicates 和 Filters
- 限流
- 路径重写

Spring Cloud Gateway 中的相关概念：

- **Route（路由）**：这是网关的基本构建块。它由一个 ID，一个目标 URI，一组断言和一组过滤器定义。如果断言为真，则路由匹配。
- **Predicate（断言）**：这是一个 Java 8 的 Predicate。输入类型是一个 ServerWebExchange。我们可以使用它来匹配来自 HTTP 请求的任何内容，例如 headers 或参数。
- **Filter（过滤器）**：这是 `org.springframework.cloud.gateway.filter.GatewayFilter` 的实例，我们可以使用它修改请求和响应。

工作原理



客户端向 Spring Cloud Gateway 发出请求。如果 Gateway Handler Mapping 中找到与请求相匹配的路由，将其发送到 Gateway Web Handler。Handler 再通过指定的过滤器链来将请求发送到我们实际的服务执行业务逻辑，然后返回。过滤器之间用虚线分开是因为过滤器可能会在发送代理请求之前（“pre”）或之后（“post”）执行业务逻辑。

虚线左边的是 pre 类型过滤器，虚线右边的是 post 类型过滤器，外部请求进来后先经过 pre 类型的过滤器，再发送到代理服务，代理服务响应请求，再次经过 post 过滤器链，最后传递给前端。

Spring Cloud Gateway 与 Zuul 对比

了解完两种网关技术之后，做一个简单的对比：

- Zuul 是 Netflix 开源的产品，Spring Cloud Gateway 是 Spring 体系内的产品，和 Spring 体系融合更好。
- Zuul1 不支持长连接，比如 WebSockets；Spring Cloud Gateway 支持 WebSockets 等协议。
- 作为一个服务网关产品，Spring Cloud Gateway 考虑更全面一些，增加了 Predicate、限流等技术。
- Zuul1 是基于 Servlet 框架构建，采用的是阻塞和多线程方式，即一个线程处理一次连接请求，这种方式在内部延迟严重、设备故障较多情况下会引起存活连接增多和线程增加的情况发生。
- Spring Cloud Gateway 基于 Spring 5.0、Spring Boot 2.0 和 Project Reactor 开发，采用了 Netty 实现了异步非阻塞模型，占用资源更小，在性能方面更有优势。

综合比较，Spring Cloud Gateway 在使用效率和性能方面更好一些，可以预期未来 Spring Cloud Gateway 的使用率会越来越高。

小结

本节为大家介绍了什么是服务网关，以及我们为什么要使用服务网关，同时介绍了 Spring Cloud 体系内的两种网关技术，以及对两种技术做了简单的对比。在最新版本中 Spring Cloud 推荐使用 Spring Cloud Gateway 作为默认的网关技术，在后面两节为大家介绍如何在项目中使用 Spring Cloud Gateway 技术。

参考：

<https://medium.com/netflix-techblog/announcing-zuul-edge-service-in-the-cloud-ab3af5be08ee>

<https://cloud.spring.io/spring-cloud-gateway/spring-cloud-gateway.html>

本文作者：纯洁的微笑、江南一点雨