



链滴

Spring Cloud 核心组件

作者: [xixiaoming](#)

原文链接: <https://ld246.com/article/1541587886167>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

Eureka (注册中心)

每个微服务都有一个Eureka Client组件，专门负责将这个服务的信息注册到Eureka Server中，也就告诉Eureka Server，自己在哪台机器上，监听着哪个端口。而Eureka Server是一个注册中心，里面一个注册表，保存了各服务所在的机器和端口号。

Feign (REST客户端)

Feign 是一个声明式REST客户端，主要是为了简便服务调用，更快捷、优雅地调用HTTP API。主要实现原理是用动态代理，你要是调用哪个接口，本质就是调用 Feign创建的动态代理。

Ribbon (负载均衡)

Ribbon的作用是负载均衡，会帮你在每次请求时选择一台机器，均匀的把请求分发到各个机器上，认使用的最经典的Round Robin轮询算法（如果发起10次请求，那就先让你请求第1台机器、然后是2台机器、第3台机器，接着再来一个循环，第1台机器、第2台机器。。。以此类推）

Hystrix (熔断器)

微服务框架是许多服务互相调用的，要是不做任何保护的话，某一个服务挂了，就会引起连锁反应，致别的服务也挂。Hystrix是隔离、熔断以及降级的一个框架。如果调用某服务报错（或者挂了），就 该服务熔断，在5分钟内请求此服务直接就返回一个默认值，不需要每次都卡几秒，这个过程，就是 谓的熔断。但是熔断了之后就会少调用一个服务，此时需要做下标记，标记本来需要做什么业务，但 因为服务挂了，暂时没有做，等该服务恢复了，就可以手工处理这些业务。这个过程，就是所谓的降 。

Zuul (服务网关)

Zuul，也就是微服务网关。这个组件是负责网络路由的。假设你后台部署了几百个服务，现在有个前 兄弟要来调用这些服务，难不成你让他把所有服务的名称和地址全部记住，这是不现实的，所以一般 服务架构中都必然会设计一个网关，所有请求都往网关走，网关会根据请求中的一些特征，将请求转 给后端的各个服务。而且有一个网关之后，还有很多好处，比如可以做统一的降级、限流、认证授权 安全，等等。

总结

1. 第一步：服务注册
2. 第二步：服务发现
3. 第三步：负载均衡
4. 第四步：服务调用
5. 第五步：隔离、熔断与降级
6. 第六步：网关路由

各个服务启动时，Eureka Client都会将服务注册到Eureka Server，并且Eureka Client还可以反过来 Eureka Server拉取注册表，从而知道其他服务在哪里。服务间发起请求的时候，基于Ribbon做负载 均衡，从一个服务的多台机器中选择一台。基于Feign的动态代理机制，根据注解和选择的机器，拼接 求URL地址，发起请求。发起请求是通过Hystrix的线程池来走的，不同的服务走不同的线程池，实现

不同服务调用的隔离，避免了服务雪崩的问题。如果前端、移动端要调用后端系统，统一从Zuul网关入，由Zuul网关转发请求给对应的服务。