**服务网关**

# Zuul介绍

Zuul的主要功能是路由转发和过滤器。路由功能是微服务的一部分，比如／api/user转发到到user服务，/api/shop转发到到shop服务。zuul默认和Ribbon结合实现了负载均衡的功能。

# Gateway介绍

Spring Cloud Gateway是Spring Cloud官方推出的第二代网关框架，取代Zuul网关。网关作为流量的，在微服务系统中有着非常作用，网关常见的功能有路由转发、权限校验、限流控制等作用

# 路由转发

# 过滤

|  |
| --- |
| @Component  public class MyFilter extends ZuulFilter {  private static Logger log = LoggerFactory.getLogger(MyFilter.class);  @Override  public String filterType() {  return "pre";  }  @Override  public int filterOrder() {  return 0;  }  @Override  public boolean shouldFilter() {  return true;  }  @Override  public Object run() {  RequestContext ctx = RequestContext.getCurrentContext();  HttpServletRequest request = ctx.getRequest();  log.info(String.format("%s >>> %s", request.getMethod(), request.getRequestURL().toString()));  Object accessToken = request.getParameter("token");  if(accessToken == null) {  log.warn("token is empty");  ctx.setSendZuulResponse(false);  ctx.setResponseStatusCode(401);  try {  ctx.getResponse().getWriter().write("token is empty");  }catch (Exception e){}  return null;  }  log.info("ok");  return null;  }  } |

filterType：返回一个字符串代表过滤器的类型，在zuul中定义了四种不同生命周期的过滤器类型，具体如下：

pre：路由之前

routing：路由之时

post： 路由之后

error：发送错误调用

filterOrder：过滤的顺序

shouldFilter：这里可以写逻辑判断，是否要过滤，本文true,永远过滤。

run：过滤器的具体逻辑。可用很复杂，包括查sql，nosql去判断该请求到底有没有权限访问