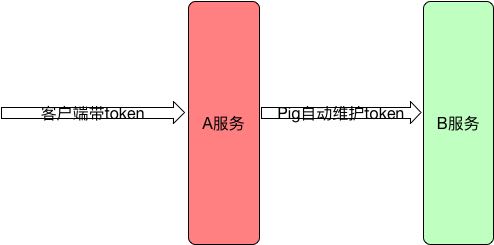
[**#**](https://pig4cloud.com/#%E5%AE%A2%E6%88%B7%E7%AB%AF%E5%B8%A6token-%E6%83%85%E5%86%B5) **客户端带Token 情况**

1. 如下图客户端携带token访问A服务。
2. A服务通过FeginClient 调用B服务获取相关依赖数据。
3. 所以只要带token 访问A 无论后边链路有多长 ABCD 都可以获取当前用户信息
4. 权限需要有这些整个链路接口的全部权限才能成功



[**#**](https://pig4cloud.com/#%E6%A0%B8%E5%BF%83%E4%BB%A3%E7%A0%81) **核心代码**

fein 拦截器将本服务的token 通过copyToken的形式传递给下游服务

public class PigFeignClientInterceptor extends OAuth2FeignRequestInterceptor {

@Override

public void apply(RequestTemplate template) {

Collection<String> fromHeader = template.headers().get(SecurityConstants.FROM);

if (CollUtil.isNotEmpty(fromHeader) && fromHeader.contains(SecurityConstants.FROM\_IN)) {

return;

}

accessTokenContextRelay.copyToken();

if (oAuth2ClientContext != null

&& oAuth2ClientContext.getAccessToken() != null) {

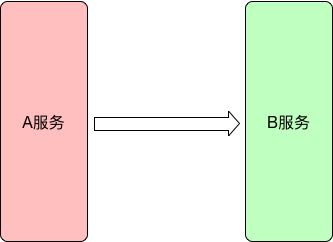
super.apply(template);

}

}

}

[**#**](https://pig4cloud.com/#%E6%97%A0token%E8%AF%B7%E6%B1%82%EF%BC%8C%E6%9C%8D%E5%8A%A1%E5%86%85%E9%83%A8%E5%8F%91%E8%B5%B7%E6%83%85%E5%86%B5%E5%A4%84%E7%90%86%E3%80%82) **无token请求，服务内部发起情况处理。**



很多情况下，比如定时任务。A服务并没有token 去请求B服务，pig也对这种情况进行了兼容。类似于A对外暴露API，但是又安全限制。参考日志插入情况

* FeignClient 需要带一个请求token,FROM\_IN 声明是内部调用

remoteLogService.saveLog(sysLog, SecurityConstants.FROM\_IN);

* 参考API 对外暴露章节对外暴露目标接口

ignore:

urls:

- 目标借口的Ant表达式即可

* 目标接口对内外调用进行限制 @Inner 注解，这样就避免接口对外暴露的安全问题。只能通过内部调用才能使用，浏览器不能直接访问该接口

@Inner

@PostMapping

public R save(@Valid @RequestBody SysLog sysLog) {

return new R<>(sysLogService.save(sysLog));

}

[**#**](https://pig4cloud.com/#%E5%90%8E%E8%BE%B9%E4%BC%9A%E4%B8%93%E9%97%A8%E8%AE%B2%40inner-%E6%B3%A8%E8%A7%A3) **后边会专门讲@Inner 注解**