1. **避免暴露实体类对象信息给外部**。
2. 不要**重复定义**返回的实体类**对象信息**，对象属于哪个服务就在哪里定义。
3. **调用方需要知道服务方的url再直接调用**，不合理。应该服务方自己定义接口暴露出来。

总结：以上几个，多个对象，多处调用，后期维护困难，代码冗余。Uri在调用方写，把服务方的服务给定义到自己的模块上，也不太合理，一但还是那种后期维护困难，比如调用方要修改url，要进行restful的风格改进，那么两边都要同时去维护，但是如果是暴露接口，接口就不需要改动，这就比较合理了。所以有了以下的解决方案：

解决方案：把服务再划分模块：

Server：处理业务逻辑

Client：对外暴露的接口（通过@FeignClient注解，暴露接口给外部调用）

Common：共用的对象信息（比如productinfo，这种服务方提供，调用方调用都需要返回的对象）

Zuul网关：

1.可以禁止接口暴露到外网，只允许内内部的服务间调用。

2.可以用于路由。