记得录屏

# 集群容错

**Failover cluster 失败的时候自动切换并重试其他服务器。 通过retries=2。 来设置重试次数**

failfast cluster 快速失败，只发起一次调用 ;写操作。比如新增记录的时候， 非幂等请求

failsafe cluster 失败安全。 出现异常时，直接忽略异常

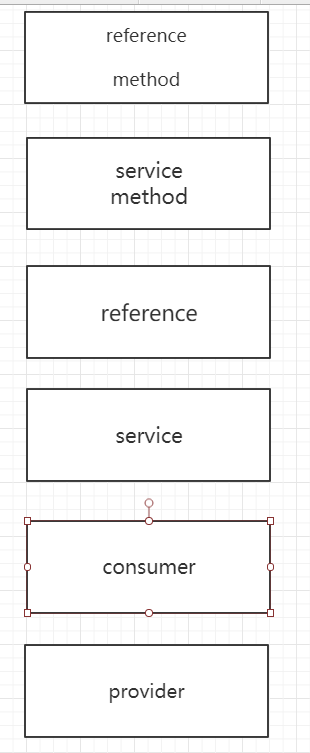
failback cluster 失败自动恢复。后台记录失败请求，定时重发

forking cluster 并行调用多个服务器，只要一个成功就返回。 只能应用在读请求

broadcastcluster广播调用所有提供者，逐个调用。其中一台报错就会返回异常

# 配置的优先级

消费端有限最高 – 服务端



# 服务改造

|  |  |
| --- | --- |
|  |  |

# 服务的最佳实践

## 分包

1. 服务接口、请求服务模型、异常信息都放在api里面，符合重用发布等价原则，共同重用原则
2. api里面放入spring的引用配置。 也可以放在模块的包目录下。 com.gupao.vip.mic.order/\*\*\*-reference.xml

## 粒度

1. 尽可能把接口设置成粗粒度，每个服务方法代表一个独立的功能，而不是某个功能的步骤。否则就会涉及到分布式事务
2. 服务接口建议以业务场景为单位划分。并对相近业务做抽象，防止接口暴增
3. 不建议使用过于抽象的通用接口 T T<泛型>，接口没有明确的语义，带来后期的维护

## 版本

1. 每个接口都应该定义版本，为后续的兼容性提供前瞻性的考虑 version（maven-snapshot）
2. 建议使用两位版本号，因为第三位版本号表示的兼容性升级，只有不兼容时才需要变更服务版本
3. 当接口做到不兼容升级的时候，先升级一半或者一台提供者为新版本，再将消费全部升级新版本，然后再将剩下的一半提供者升级新版本

预发布环境

# 推荐用法

## 在provider端尽可能配置consumer端的属性

比如timeout、retires、线程池大小、LoadBalance

## 配置管理员信息

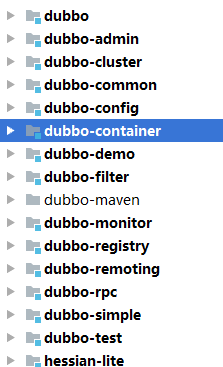
application上面配置的owner、 owner建议配置2个人以上。因为owner都能够在监控中心看到

# 配置dubbo缓存文件

注册中心的列表

服务提供者列表

# 源码分析



基于spring配置文件的扩展的话

NamespaceHandler: 注册BeanDefinitionParser， 利用它来解析

BeanDefinitionParser： 解析配置文件的元素

spring会默认加载jar包下/META-INF/spring.handlers找到对应的NamespaceHandler

initializingBean,当spring容器初始化完以后，会调用afterPropertiesSet方法

DisposableBean,

ApplicationContextAware,

ApplicationListener,

BeanNameAware

# 课后作业

如下，pay-center做了四台机器的集群，每个机器上都运行了一个定时任务。

需求是：

按照如下架构搭建一个分布式应用架构，使用dubbo做为rpc

保证定时任务的触发只在其中一台机器上进行，其他机器不执行，通过zookeeper去实现

