上堂课回顾

1、传统springmvc项目拆分的演示 ---- 问题障碍

2、集群容错配置

---- cluster="failover(retries=2)/failfast/failsafe/failback/forking(forks=2)”

3、负载配置 ------ loadbalance="random,roundrobin,leastactive"，分别表示：随机，轮询，最少活跃调用

4、缓存配置 ----- cache="lru/threadlocal"

5、异步调用

本堂课内容：

1、dubbo标签常用，工作中使用---- 异步/事件触发/泛型调用/回声测

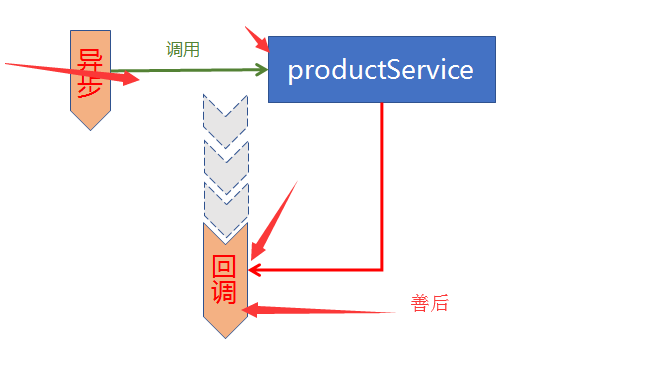
2、原理dubbo探究 ---- rpc整个调用过程

3、dubbo整个初始化过程

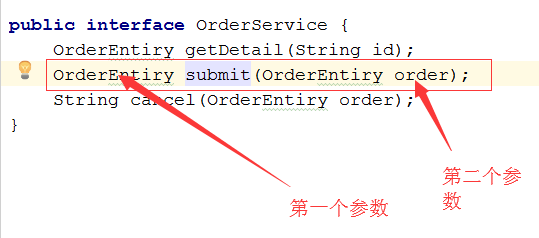
一、业务方法很长时

1、异步--- 多个请求并发

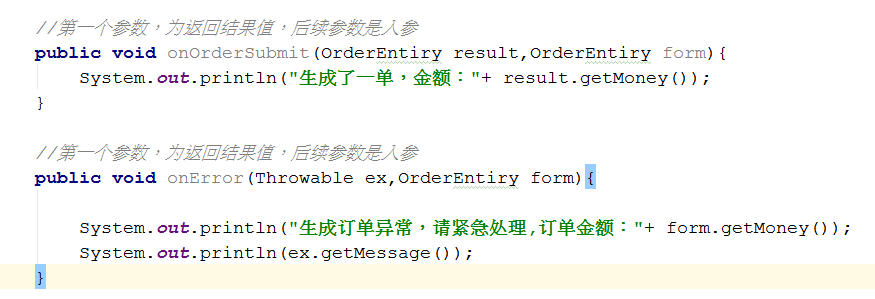
2、异步回调--- 时机（当前不需要返回值）---- 回调是在另一个线程中进行



使用方法



2.1、正常业务返回时，配置onreturn，异常抛出时，onthrow



二、回声测试，所有的reference代理，都实现 Echoservice接口



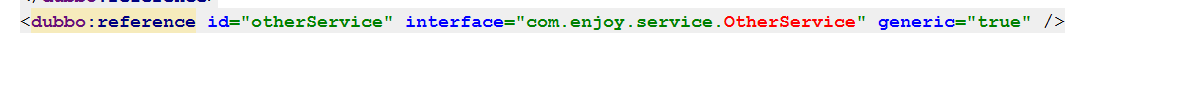
三、泛化调用

泛化 ---- 抽象

当A项目没有得到B项目接口描述，它还想要rpc调用它，有没有办法？

类似java反射

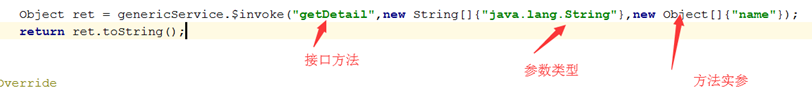
1、配置dubbo引入，设置generic = true



2、从IOC容器中取出代理对象，转为泛型接口对象

通过$invoke方法调用目标方法（传入方法名/参数类型/参数值）

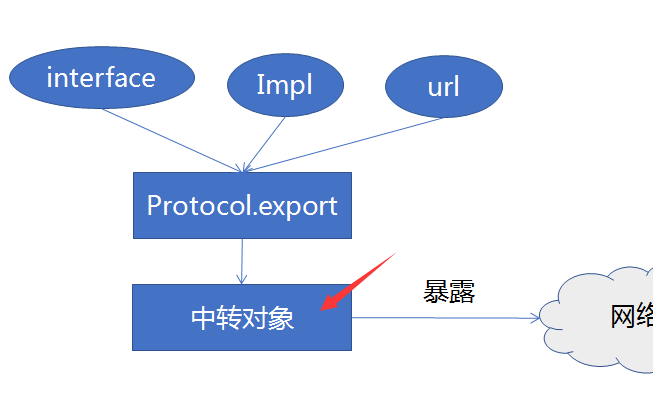




使用场景--------当项目未能得到目标服务的支持包时，用此方法救急（后续应当更正为常规方法）

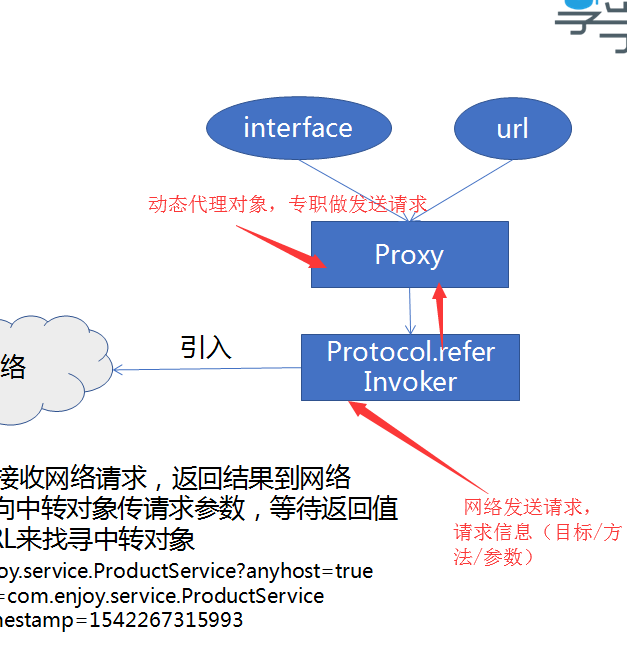
四、rpc调用过程

1、暴露，就是为某个服务创建好一个中转对象（参接触网络/能调用本地service）

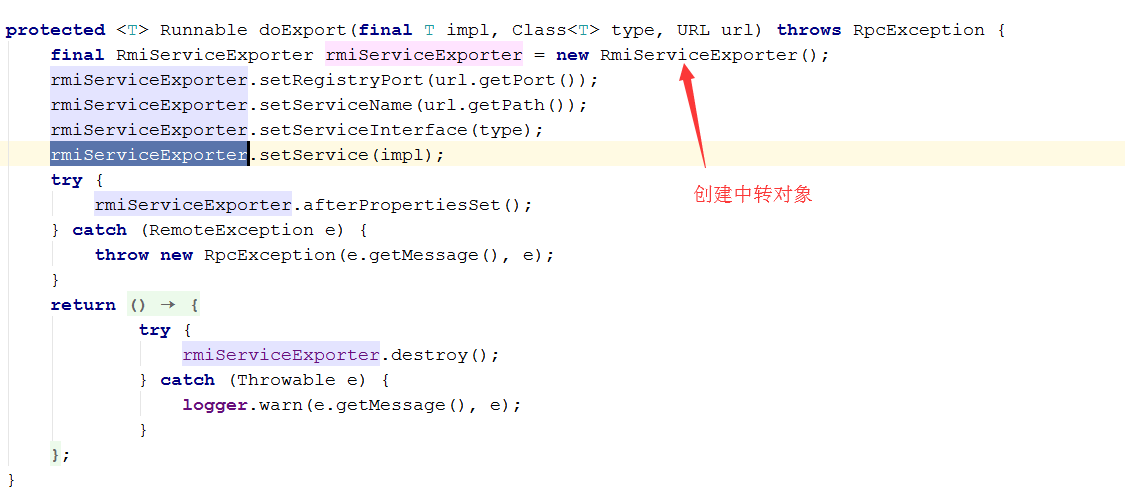


消费方，把请求的目标/方法/参数发送给中转对象，中转对象就能执行方法并返回

2、引入：建一个动态代理对象，专职来发送本接口请求信息

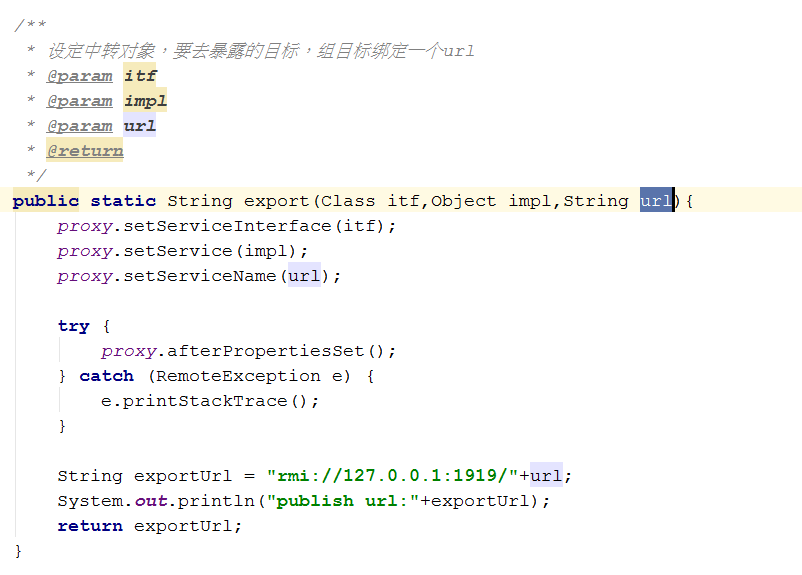


3、源码走读，rmi示例

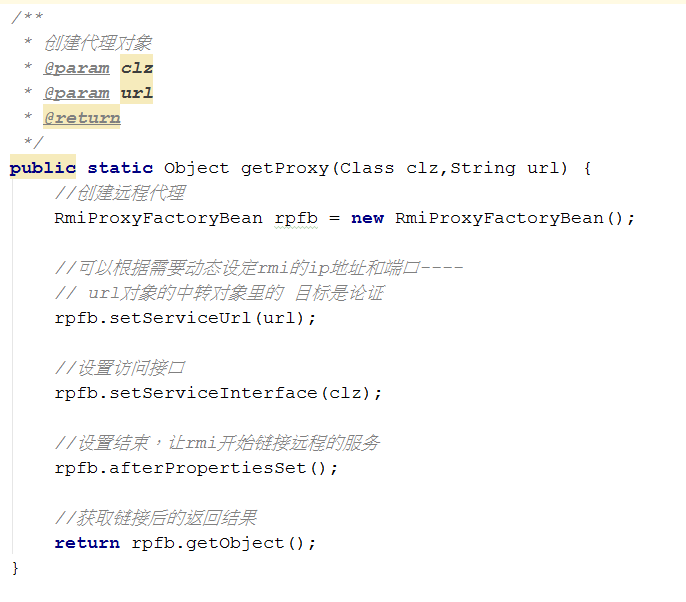


4、springrmi模拟：

服务端在中转对象上，设定--- 接口/实现/url

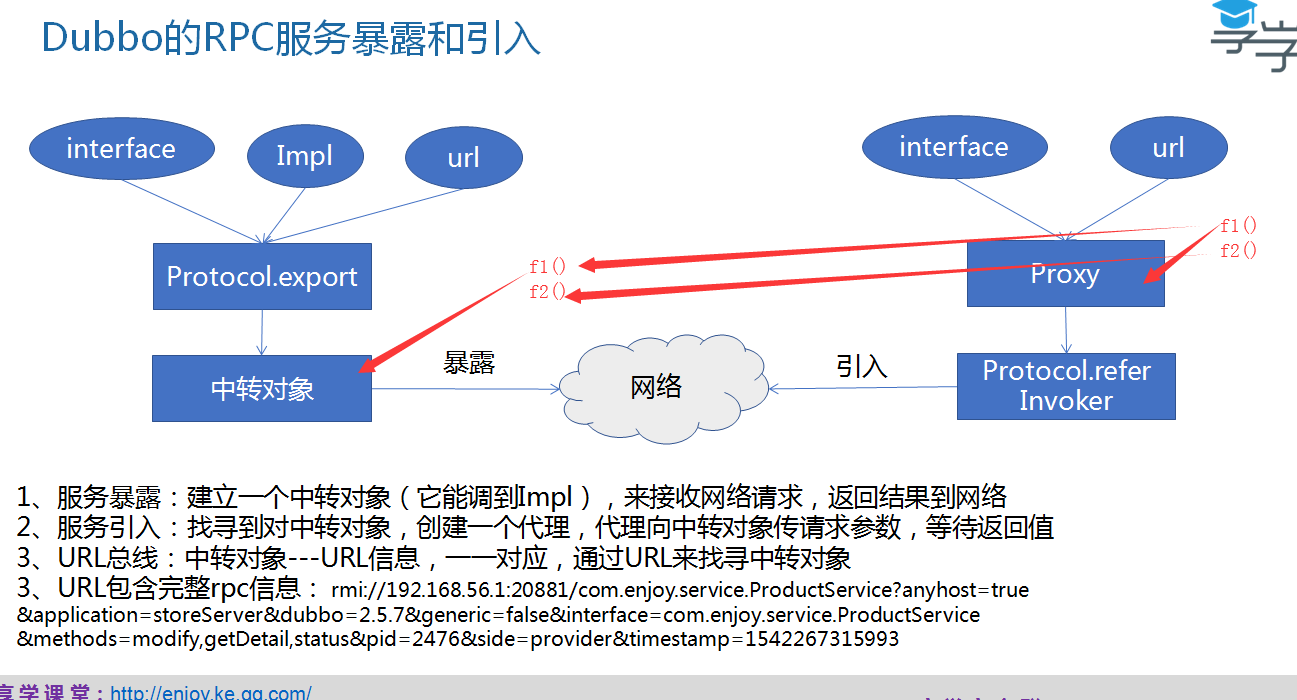


消费端创建代理对象，设定 接口/url



3、调用过程是这样的

中转对象，跟着代理对象动作，猴子学样

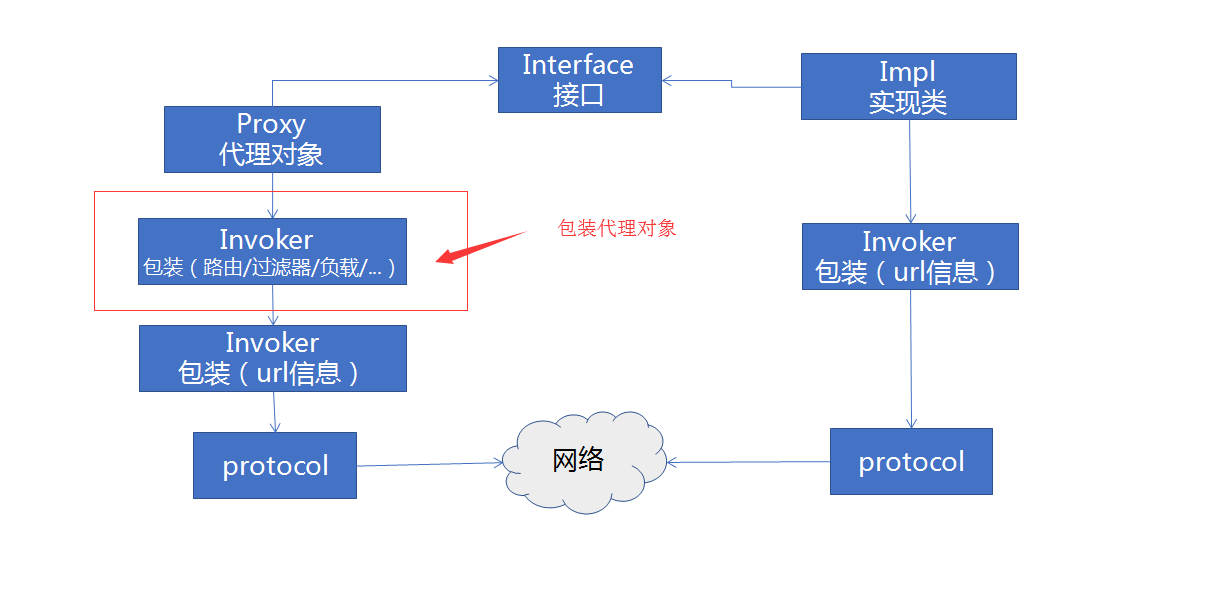


4、dubbo代理对象与中转对象（目标) 建立的关系，通过URL来传递的。

URL是整个dubbo里的总信息描述符。

消费方包装代理对象，做负载/路由/容错等等的策略

属性：



答疑

1、dubbo协议为什么不细讲？

dubbo协议，是阿里对netty协议做一些改造后，又加以封闭的一个协议，里面涉及大量的网络编程知识。我们dubbo里强调使用过程就行，在此只就rmi协议为例讲解一下流程。

2、dubbo一般选择使用哪个协议？

最常用，就是dubbo协议/hession协议

选择协议，一般就考量协议的性能，就数这两个协议效率高

3、zookeeper在dubbo里，处于什么位置

zk在dubbo里起的作用 --- 数据库 + 消息推送

同样的，使用redis一样能做数据的记录和消息投送，这对dubbo来说，没有区别。

4、dubbo学习，重点还是使用上面（拆解项目/规划项目结构）。

5、http调用，就是一个rpc调用