**课程概要：**

1. 分布式项目开发与联调
2. 控制管理后台使用
3. Dubbo注册中心详解

一、分布式项目开发与联调

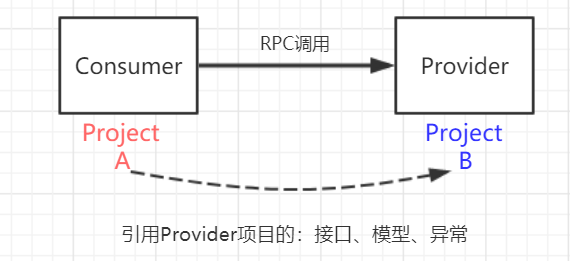
**接口暴露与引用**

在一个RPC场景中 ，调用方是通过接口来调用服务端，传入参数并获得返回结果。这样服务端的接口和模型必须暴露给调用方项目。服务端如何暴露呢？客户端如何引用呢？

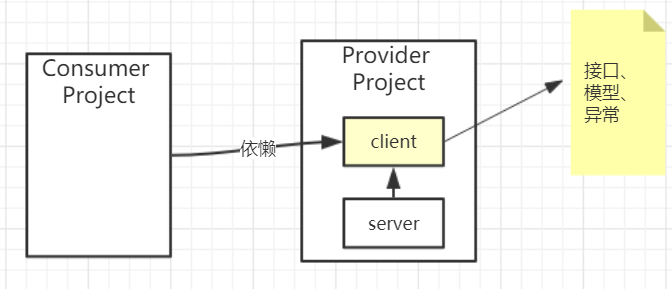
**接口信息**

**、模型信息**

**、异常**

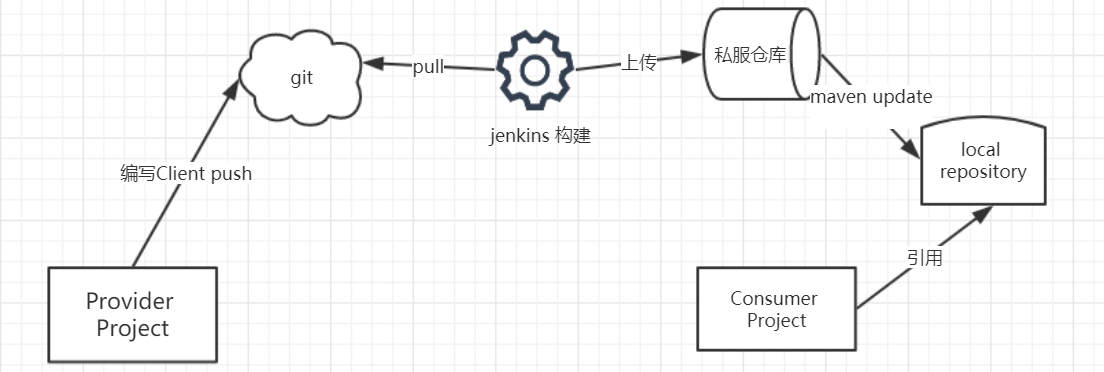


暴露接口的通常做法是 接口与实现分离，服务端将 接口、模型、异常 等统一放置于一个模块，实现置于另一个模块。调用方通过Maven进行引用。



**自动化构建与协作**

当项目越来越多，服务依懒关系越发复杂的时候，为了提高协作效率，必须采用自动化工具 完成 接口从编写到构建成JAR包，最后到引用的整个过程。

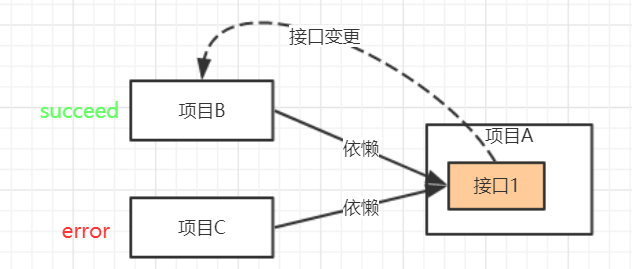


流程描述：

1. 服务提供者项目发人员编写Client 接口
2. push 至远程仓库
3. jenkins 构建指定版本
4. jenkins Deploye 至私服仓库 nexus
5. 服务消费者项目开发人员基于maven 从私服务仓库下载

**接口平滑升级：**

在项目迭代过程当中， 经常会有多个项目依懒同一个接口，如下图 项目B、C都依懒了项目A当中的接口1，此时项目B业务需要，需要接口1多增加一个参数，升级完成后。项目B能正确构建上线，项目C却不行。



解决办法与原则：

1. 接口要做到向下兼容：接口参数尽量以对象形式进行封装。Model属性只增不删，如果需要作废，可以添加@Deprecated  标识。
2. 如果出现了不可兼容的变更，则必须通知调用方整改，并制定上线计划。

**开发联调：**

在项目开发过程当中，一个开发或测试环境的注册中心很有可能会同时承载着多个服务，如果两组服务正在联调，如何保证调用的是目标服务呢？

**1、基于临时分组联调**

group 分组

 在reference 和server 当中采用相同的临时组 ,通过group 进行设置

**2、直连提供者：**

在reference 中指定提供者的url即可做到直连

<dubbo:reference  url="dubbo://127.0.0.1:20880" id="demoService"

                  timeout="2000"

                  interface="com.tuling.teach.service.DemoService" check="false"/>

**3、只注册：**

一个项目有可能同是为即是服务提供者又消费者，在测试时需要调用某一服务同时又不希望正在开发的服务影响到其它订阅者如何实现？

通过修改 register=false 即可实现

<dubbo:registry address="multicast://224.5.6.7:1234" register="false"/>

二、Dubbo控制管理后台使用

**Dubbo 控制后台版本说明：**

dubbo 在2.6.0 以前 使用dubbo-admin 作为管理后台，2.6 以后已经去掉dubbo-admin 并采用 incubator-dubbo-ops  作为新的管理后台，目前该后台还在开发中还没有发布正式的版本  ，所以本节课还是采用的旧版的dubbo-admin 来演示。

**Dubbo 控制后台的安装：**

#从github 中下载dubbo 项目

git clone https://github.com/apache/incubator-dubbo.git

#更新项目

git fetch

#临时切换至 dubbo-2.5.8 版本

git checkout dubbo-2.5.8

#进入 dubbo-admin 目录

cd dubbo-admin

#mvn 构建admin war 包

mvn clean pakcage -DskipTests

#得到 dubbo-admin-2.5.8.war 即可直接部署至Tomcat

#修改 dubbo.properties 配置文件

dubbo.registry.address=zookeeper://127.0.0.1:2181

注：如果实在懒的构建 可直接下载已构建好的：

链接：<https://pan.baidu.com/s/1zJFNPgwNVgZZ-xobAfi5eQ> 提取码：gjtv

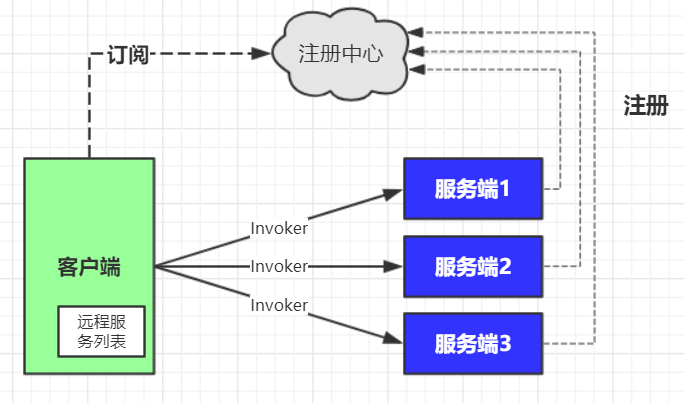
**控制后台基本功能介绍 ：**

* 服务查找：
* 服务关系查看:
* 服务权重调配：
* 服务路由：
* 服务禁用

三、Dubbo注册中心详解

注册中心的作用

为了到达服务集群动态扩容的目的，注册中心存储了服务的地址信息与可用状态信息，并实时推送给订阅了相关服务的客户端。



**一个完整的注册中心需要实现以下功能：**

1. 接收服务端的注册与客户端的引用，即将引用与消费建立关联，并支持多对多。
2. 当服务非正常关闭时能即时清除其状态
3. 当注册中心重启时，能自动恢复注册数据，以及订阅请求
4. 注册中心本身的集群

Dubbo所支持的注册中心

1. **Multicast 注册中心**
   1. 基于组网广播技术，只能用在局域网内，一般用于简单的测试服务
2. **Zookeeper 注册中心(推荐)**
   1. [Zookeeper](http://zookeeper.apache.org/) 是 Apacahe Hadoop 的子项目，是一个树型的目录服务，支持变更推送，适合作为 Dubbo 服务的注册中心，工业强度较高，可用于生产环境，并推荐使用
3. **Redis 注册中心**
   1. 基于Redis的注册中心
4. **Simple 注册中心**
   1. 基于本身的Dubbo服务实现（SimpleRegistryService），不支持集群可作为自定义注册中心的参考，但不适合直接用于生产环境。

**Redis 注册中心**

关于Redis注册中心我们需要了解两点，

1. 如何存储服务的注册与订阅关系
2. 是当服务状态改变时如何即时更新

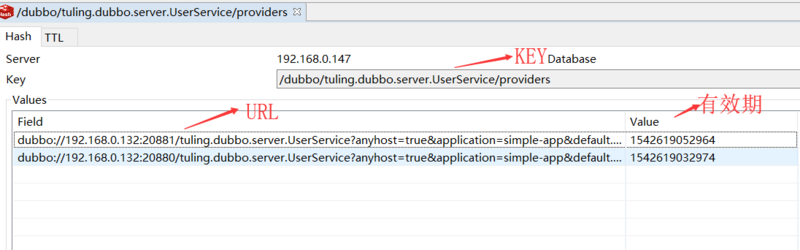
演示使用Redis 做为注册中心的使用。

* 启动Redis服务
* 服务端配置注册中心
* 启动两个服务端
* 通过RedisClient 客户端观察Redis中的数据

redis 注册中心配置：

<dubbo:registry protocol="redis" address="192.168.0.147:6379"/>

当我们启动两个服务端后发现，Reids中增加了一个Hash 类型的记录，其key为/dubbo/tuling.dubbo.server.UserService/providers。Value中分别存储了两个服务提供者的URL和有效期。



**同样消费者也是类似其整体结构如下：**

//服务提供者注册信息

/dubbbo/com.tuling.teach.service.DemoService/providers

dubbo://192.168.246.1:20880/XXX.DemoService=1542619052964

dubbo://192.168.246.2:20880/XXX.DemoService=1542619052964

//服务消费订阅信息

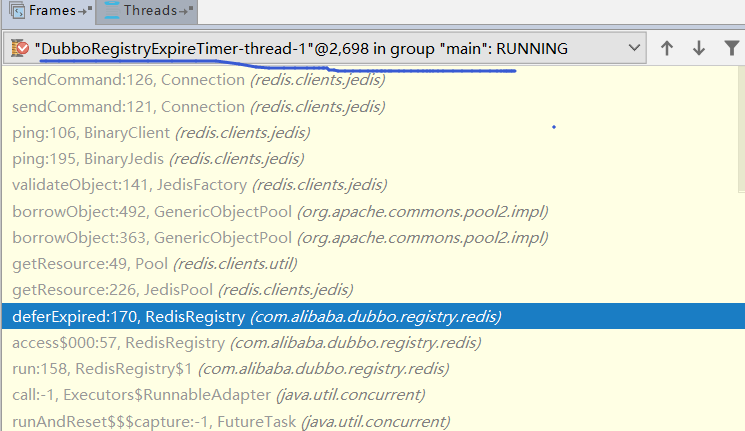
/dubbbo/com.tuling.teach.service.DemoService/consumers

dubbo://192.168.246.1:20880/XXX.DemoService=1542619788641

* 主 Key 为服务名和类型
* Map 中的 Key 为 URL 地址
* Map 中的 Value 为过期时间，用于判断脏数据，脏数据由监控中心删除

接下来回答第二个问题 **当提供者突然 宕机状态能即里变更吗**？

这里Dubbo采用的是定时心跳的机制 来维护服务URL的有效期，默认每30秒更新一次有效期。即URL对应的毫秒值。具体代码参见：com.alibaba.dubbo.registry.redis.RedisRegistry#expireExecutor



com.alibaba.dubbo.registry.redis.RedisRegistry#deferExpired

com.alibaba.dubbo.registry.integration.RegistryDirectory

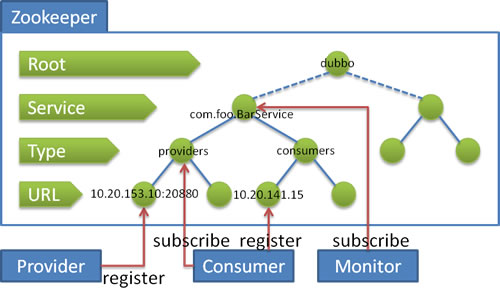
com.alibaba.dubbo.registry.support.ProviderConsumerRegTable

**Zookeeper 注册中心**

关于Zookeeper 注册中心同样需要了解其存储结构和更新机制。

Zookeper是一个树型的目录服务，本身支持变更推送相比redis的实现Publish/Subscribe功能更稳定。

结构：



**失败重连**

com.alibaba.dubbo.registry.support.FailbackRegistry

**提供者突然断开：**

基于Zookeeper 临时节点机制实现，在客户端会话超时后 Zookeeper会自动删除所有临时节点，默认为40秒。

// 创建临时节点

com.alibaba.dubbo.remoting.zookeeper.curator.CuratorZookeeperClient#createEphemeral

提问：

在zookeeper 断开的40秒内 如果 有客户端加入 会调用 已失效的提供者连接吗？

答：不会，提供者宕机后 ，其与客户端的链接也随即断开，客户端在调用前会检测长连接状态。

// 检测连接是否有效

com.alibaba.dubbo.rpc.protocol.dubbo.DubboInvoker#isAvailable

创建 configurators与routers 会创建持久节点

// 创建持久节点

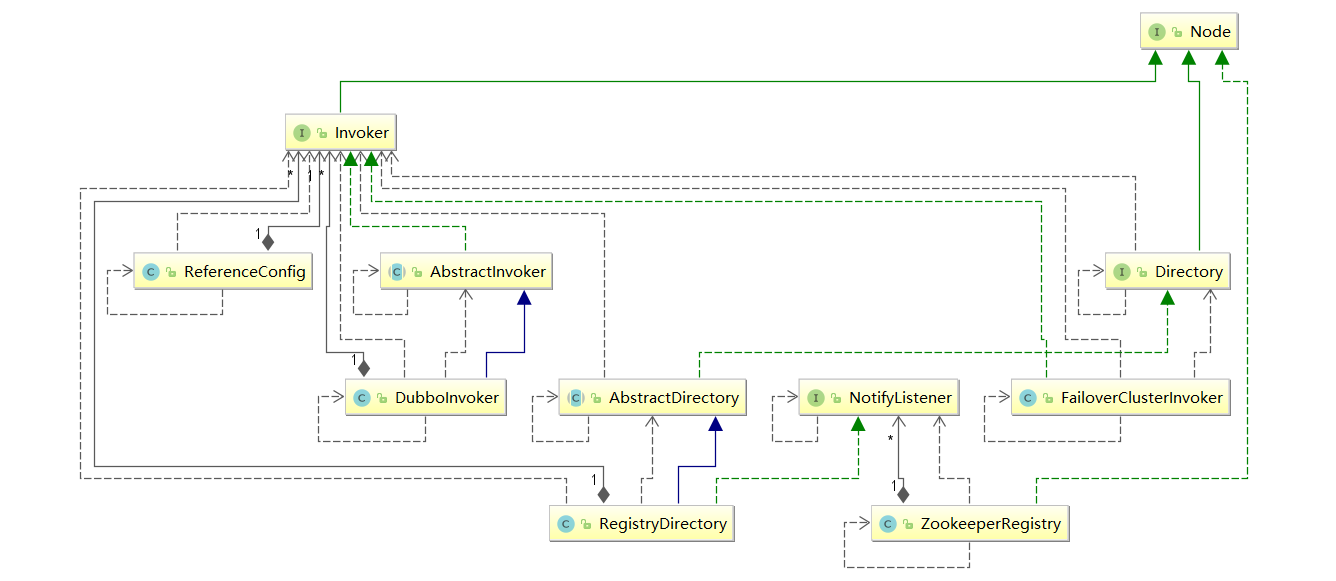
com.alibaba.dubbo.remoting.zookeeper.curator.CuratorZookeeperClient#createPersistent

**服务订阅机制实现：**

// 注册目录

com.alibaba.dubbo.registry.integration.RegistryDirectory

源码解析：



com.alibaba.dubbo.registry.integration.RegistryDirectory