1. zookeeper集群搭建

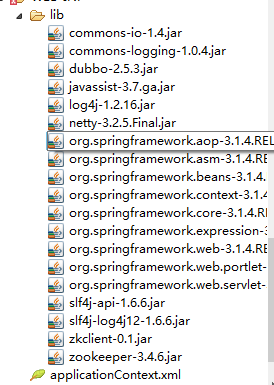
参考《zookeeper单节点安装和伪分布式集群安装和完全分布式集群安装.doc》

1. Spring集成dubbo

Dubbo采用全Spring配置方式，透明化接入应用，对应用没有任何API侵入，只需用Spring加载Dubbo的配置即可，Dubbo基于Spring的Schema扩展进行加载。

## Provider部分

新建一个Web工程 ，导入以下包：



新建一个DemoService.java类，提供者和消费者都要有这个Service，

**[java]** [view plain](http://blog.csdn.net/qduningning/article/details/41445937/" \o "view plain) [copy](http://blog.csdn.net/qduningning/article/details/41445937/" \o "copy) [IMG_257](https://code.csdn.net/snippets/531246)[IMG_258](https://code.csdn.net/snippets/531246/fork)

1. **package** com.znn.provider;
3. **public** **interface** DemoService {
4. String sayHello(String name);
5. }

新建一个服务提供方实现接口，对服务消费方隐藏实现  
DemoServiceImpl.java

**[java]** [view plain](http://blog.csdn.net/qduningning/article/details/41445937/" \o "view plain) [copy](http://blog.csdn.net/qduningning/article/details/41445937/" \o "copy) [IMG_259](https://code.csdn.net/snippets/531246)[IMG_260](https://code.csdn.net/snippets/531246/fork)

1. **package** com.znn.provider;
3. **public** **class** DemoServiceImpl **implements** DemoService{
5. @Override
6. **public** String sayHello(String name) {
7. **return** "Hello Dubbo,Hello " + name;
8. }
10. }

用Spring配置声明暴露服务：

新建一个dubbo-provider.xml文件对dubbo进行配置：

**[html]** [view plain](http://blog.csdn.net/qduningning/article/details/41445937/" \o "view plain) [copy](http://blog.csdn.net/qduningning/article/details/41445937/" \o "copy) [IMG_261](https://code.csdn.net/snippets/531246)[IMG_262](https://code.csdn.net/snippets/531246/fork)

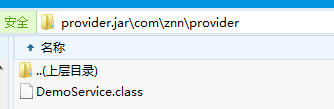
1. **<?xml** version="1.0" encoding="UTF-8"**?>**
2. **<beans** xmlns="http://www.springframework.org/schema/beans"
3. xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4. xmlns:dubbo="http://code.alibabatech.com/schema/dubbo"
5. xsi:schemaLocation="http://www.springframework.org/schema/beans
6. http://www.springframework.org/schema/beans/spring-beans.xsd
7. http://code.alibabatech.com/schema/dubbo
8. http://code.alibabatech.com/schema/dubbo/dubbo.xsd"**>**
9. <!-- 提供方应用信息，用于计算依赖关系 -->
10. **<dubbo:application** name="hello-world-app"  **/>**
11. <!-- 使用zookeeper注册中心暴露服务地址 -->
12. **<dubbo:registry** address="zookeeper://192.168.0.181:2181" **/>**
13. <!-- 用dubbo协议在20880端口暴露服务 -->
14. **<dubbo:protocol** name="dubbo" port="20880" **/>**
15. <!-- 声明需要暴露的服务接口 -->
16. **<dubbo:service** interface="com.znn.provider.DemoService" ref="demoService" **/>**
17. <!-- 和本地bean一样实现服务 -->
18. **<bean** id="demoService" class="com.znn.provider.DemoServiceImpl" **/>**
19. **</beans>**

注：自己搭了一个zookeeper，使用zookeeper来管理。

## 消费者部分

另外新建一个工程：znn-service-consumer

导入第三方包，和提供者一样。另外可以把提供者打成一个Jar，在消费者部分添加，如果用的是Eclipse并且提供者和消费者在同一个workplace下，可以直接通过右键->propertities->Java Build Path->Project->Add来添加。此处打jar包，如下：



将此jar包加入到消费者工程的lib里。

新建一个dubbo-consumer.xml，对dubbo进行配置：

**[html]** [view plain](http://blog.csdn.net/qduningning/article/details/41445937/" \o "view plain) [copy](http://blog.csdn.net/qduningning/article/details/41445937/" \o "copy) [IMG_263](https://code.csdn.net/snippets/531246)[IMG_264](https://code.csdn.net/snippets/531246/fork)

1. **<?xml** version="1.0" encoding="UTF-8"**?>**
2. **<beans** xmlns="http://www.springframework.org/schema/beans"
3. xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4. xmlns:dubbo="http://code.alibabatech.com/schema/dubbo"
5. xsi:schemaLocation="http://www.springframework.org/schema/beans
6. http://www.springframework.org/schema/beans/spring-beans.xsd
7. http://code.alibabatech.com/schema/dubbo
8. http://code.alibabatech.com/schema/dubbo/dubbo.xsd"**>**
9. <!-- 消费方应用名，用于计算依赖关系，不是匹配条件，不要与提供方一样 -->
10. **<dubbo:application** name="consumer-of-helloworld-app"**/>**
11. **<dubbo:registry** address="zookeeper://192.168.0.181:2181" **/>**
12. <!-- 生成远程服务代理，可以和本地bean一样使用demoService -->
13. **<dubbo:reference** id="demoService" interface="com.znn.provider.DemoService" **/>**
14. **</beans>**

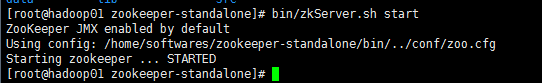
新建一个测试类，当然具体用的时候会写一个专门的类使用IOC注入：

**[java]** [view plain](http://blog.csdn.net/qduningning/article/details/41445937/" \o "view plain) [copy](http://blog.csdn.net/qduningning/article/details/41445937/" \o "copy) [IMG_265](https://code.csdn.net/snippets/531246)[IMG_266](https://code.csdn.net/snippets/531246/fork)

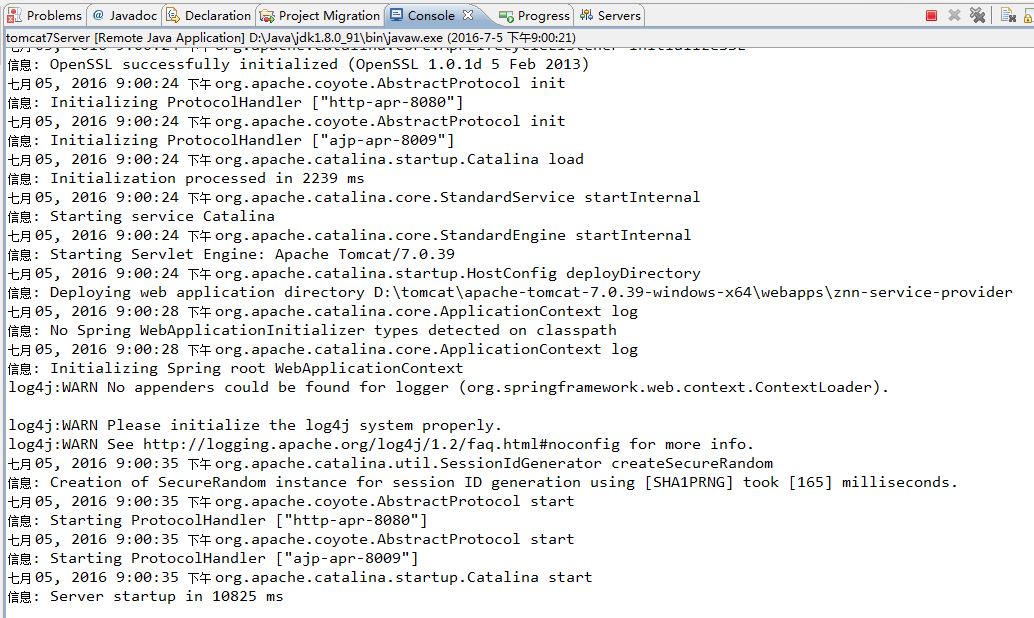
1. **package** com.znn.consumer;
3. **import** org.springframework.context.support.ClassPathXmlApplicationContext;
4. **import** com.znn.provider.DemoService;
6. **public** **class** Consumer {
7. **public** **static** **void** main(String[] args) {
8. ClassPathXmlApplicationContext context = **new** ClassPathXmlApplicationContext("file:D:\\Workspaces\\znn-service-consumer\\WebRoot\\WEB-INF\\applicationContext.xml");
9. context.start();
10. DemoService demoService = (DemoService)context.getBean("demoService"); // 获取远程服务代理
11. String hello = demoService.sayHello("world"); // 执行远程方法
12. System.out.println(hello);
13. }
14. }

运行步骤：

1. 运行zookeeper服务，可以单机模式，或者伪分布模式，或者集群模式，为演示方便，这里采用单机模式。



1. 将提供者部署到web容器tomcat中，启动tomcat，运行提供者：



1. 运行Consumer中的main方法，运行消费者：

