简单的一个provide 和cosumer

# 准备工作

## 1、两个mave项目，一个provider 一个cosumer，

## 2、搭建好spring

# 1、提供者

## 1、pom依赖

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">  
 <modelVersion>4.0.0</modelVersion>  
 <groupId>com.wpb.dubbo</groupId>  
 <artifactId>dubbo-provide</artifactId>  
 <version>0.0.1-SNAPSHOT</version>  
   
 <properties>  
 <spring.version>4.1.3.RELEASE</spring.version>  
 </properties>  
 <dependencies>  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-context</artifactId>  
 <version>${spring.version}</version>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-beans</artifactId>  
 <version>${spring.version}</version>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-webmvc</artifactId>  
 <version>${spring.version}</version>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-jdbc</artifactId>  
 <version>${spring.version}</version>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-aspects</artifactId>  
 <version>${spring.version}</version>  
 </dependency>  
  
 <dependency>  
 <groupId>com.alibaba</groupId>  
 <artifactId>dubbo</artifactId>  
 <version>2.4.10</version>  
 <exclusions>  
 <exclusion>  
 <artifactId>spring</artifactId>  
 <groupId>org.springframework</groupId>  
 </exclusion>  
 </exclusions>  
 </dependency>  
  
 <dependency>  
 <groupId>com.101tec</groupId>  
 <artifactId>zkclient</artifactId>  
 <version>0.3</version>  
 </dependency>  
 </dependencies>  
</project>

## 2、接口和实现类

### 1、接口

public interface DubboProvide {  
  
 void sayHello();  
}

### 2、实现类

public class DubboProvideImpl implements DubboProvide {  
  
 public void sayHello() {  
 System.out.println("this is my first dubbo program");  
 }  
  
}

## 3、spring 配置dubbo zookeeper

### 1、<dubbo:application 给服务器起的名字

### 2、 <dubbo:registry用zookeeper注册服务中心暴露服务地址 –

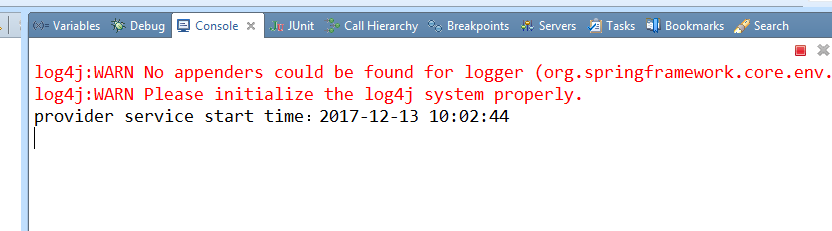
### 3、 <dubbo:service 提供服务的接口，以及实现类进行注入（有点类似webservice）

*<?*xml version="1.0" encoding="UTF-8"*?>*<beans xmlns="http://www.springframework.org/schema/beans"   
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"   
 xmlns:dubbo="http://code.alibabatech.com/schema/dubbo"   
 xsi:schemaLocation="http://www.springframework.org/schema/beans   
 http://www.springframework.org/schema/beans/spring-beans.xsd   
 http://code.alibabatech.com/schema/dubbo   
 http://code.alibabatech.com/schema/dubbo/dubbo.xsd   
 ">   
 <!-- 给dubbo服务者起个名 -->   
 <dubbo:application name="productor"/>   
   
 <!-- 用zookeeper注册服务中心暴露服务地址 -->   
 <dubbo:registry protocol="zookeeper" address="zookeeper://127.0.0.1:2181"/>   
 <!-- 暴露dubbo的通信端口 -->   
 <dubbo:protocol name="dubbo" port="20880"/>   
   
 <!-- 提供服务的实现类 -->   
 <bean id="bubboProdutor" class="com.dubbo.service.impl.DubboProvideImpl"></bean>   
 <!-- 给消费者提供服务的接口 -->   
 <dubbo:service ref="bubboProdutor" interface="com.dubbo.service.DubboProvide"></dubbo:service>   
 </beans>

## 4、加载这个spring配置文件

public class DubboProvideStart {  
  
 public static void main(String[] args) throws Exception {   
 ClassPathXmlApplicationContext cfg = new ClassPathXmlApplicationContext("classpath:config/ApplicationContextProducror.xml");   
 cfg.start();   
 SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");   
 Date date = new Date();   
 System.out.println("provider service start time��"+sdf.format(date));   
   
 System.in.read();   
 }   
}

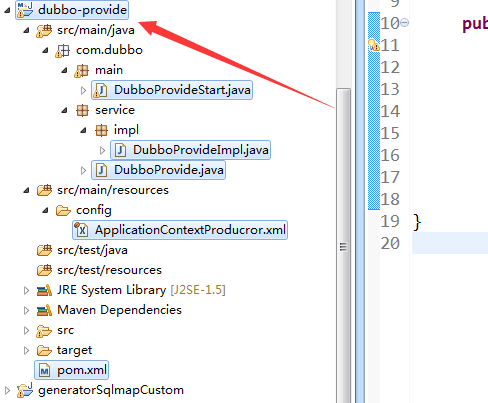
### 1、控制台查看



## 5、在dubbo-admin浏览器的提供者中会看到有一个provide



## 6、代码位置



# 2、消费者

## 1、pom依赖

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">  
 <modelVersion>4.0.0</modelVersion>  
 <groupId>com.wpb.dubbo</groupId>  
 <artifactId>dubbo-consumer</artifactId>  
 <version>0.0.1-SNAPSHOT</version>  
   
 <properties>  
 <spring.version>4.1.3.RELEASE</spring.version>  
 </properties>  
 <dependencies>  
 <dependency>  
 <groupId>com.wpb.dubbo</groupId>  
 <artifactId>dubbo-provide</artifactId>  
 <version>0.0.1-SNAPSHOT</version>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-context</artifactId>  
 <version>${spring.version}</version>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-beans</artifactId>  
 <version>${spring.version}</version>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-webmvc</artifactId>  
 <version>${spring.version}</version>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-jdbc</artifactId>  
 <version>${spring.version}</version>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-aspects</artifactId>  
 <version>${spring.version}</version>  
 </dependency>  
  
 <dependency>  
 <groupId>com.alibaba</groupId>  
 <artifactId>dubbo</artifactId>  
 <version>2.4.10</version>  
 <exclusions>  
 <exclusion>  
 <artifactId>spring</artifactId>  
 <groupId>org.springframework</groupId>  
 </exclusion>  
 </exclusions>  
 </dependency>  
  
 <dependency>  
 <groupId>com.101tec</groupId>  
 <artifactId>zkclient</artifactId>  
 <version>0.3</version>  
 </dependency>  
 </dependencies>  
</project>

## 2、只要提供者的接口（可以通过将提供者制作成依赖jar，然后pom进行引入，也可以直接复制过来提供者的接口，笔者建议使用第一种）

## 3、spring配置

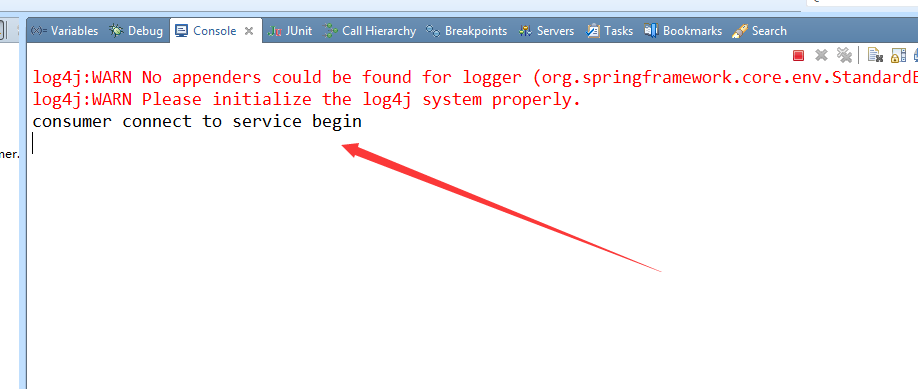
### 1、dubbo:reference <!-- 调用远程的接口 -->

*<?*xml version="1.0" encoding="UTF-8"*?>*<beans xmlns="http://www.springframework.org/schema/beans"   
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"   
 xmlns:dubbo="http://code.alibabatech.com/schema/dubbo"   
 xsi:schemaLocation="http://www.springframework.org/schema/beans   
 http://www.springframework.org/schema/beans/spring-beans.xsd   
 http://code.alibabatech.com/schema/dubbo   
 http://code.alibabatech.com/schema/dubbo/dubbo.xsd   
 ">   
   
 <!-- 消费者名称 -->   
 <dubbo:application name="customer"/>   
 <!-- 用zookeeper注册服务中心发现服务地址 -->   
 <dubbo:registry protocol="zookeeoer" address="zookeeper://127.0.0.1:2181"/>   
 <!-- 调用远程的接口 -->   
 <dubbo:reference id="dubboProvider" interface="com.dubbo.service.DubboProvide"/>   
   
</beans>

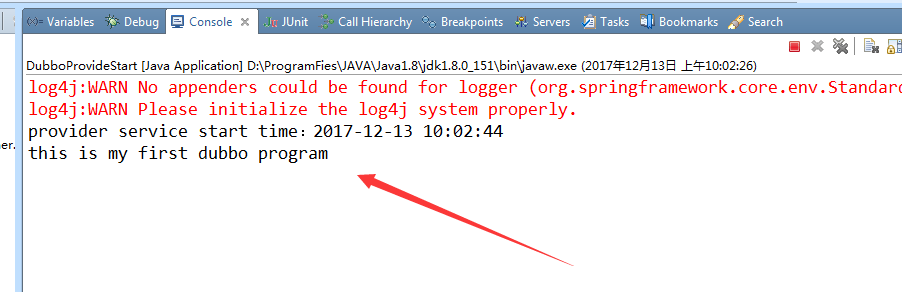
## 4、加载配置文件，并main函数进行调用测试。

public class CustomerMain {  
 public static void main(String[] args) throws InterruptedException{   
 ClassPathXmlApplicationContext cfig = new ClassPathXmlApplicationContext("classpath:config/ApplicationContextCustomer.xml");   
 DubboProvide test = (DubboProvide) cfig.getBean("dubboProvider");   
 System.out.println("consumer connect to service begin");   
 test.sayHello();   
 Thread.sleep(100000);   
 System.out.println("consumer connect to service end");   
 }   
}

## 5、关照消费者端的控制台，发现没有结果，那当然不会有结果了，这个类似于webservice。会在提供者端打印结果



## 6、观察提供者端控制台



## 7、观察dubbo-admin浏览器



## 8、代码位置

