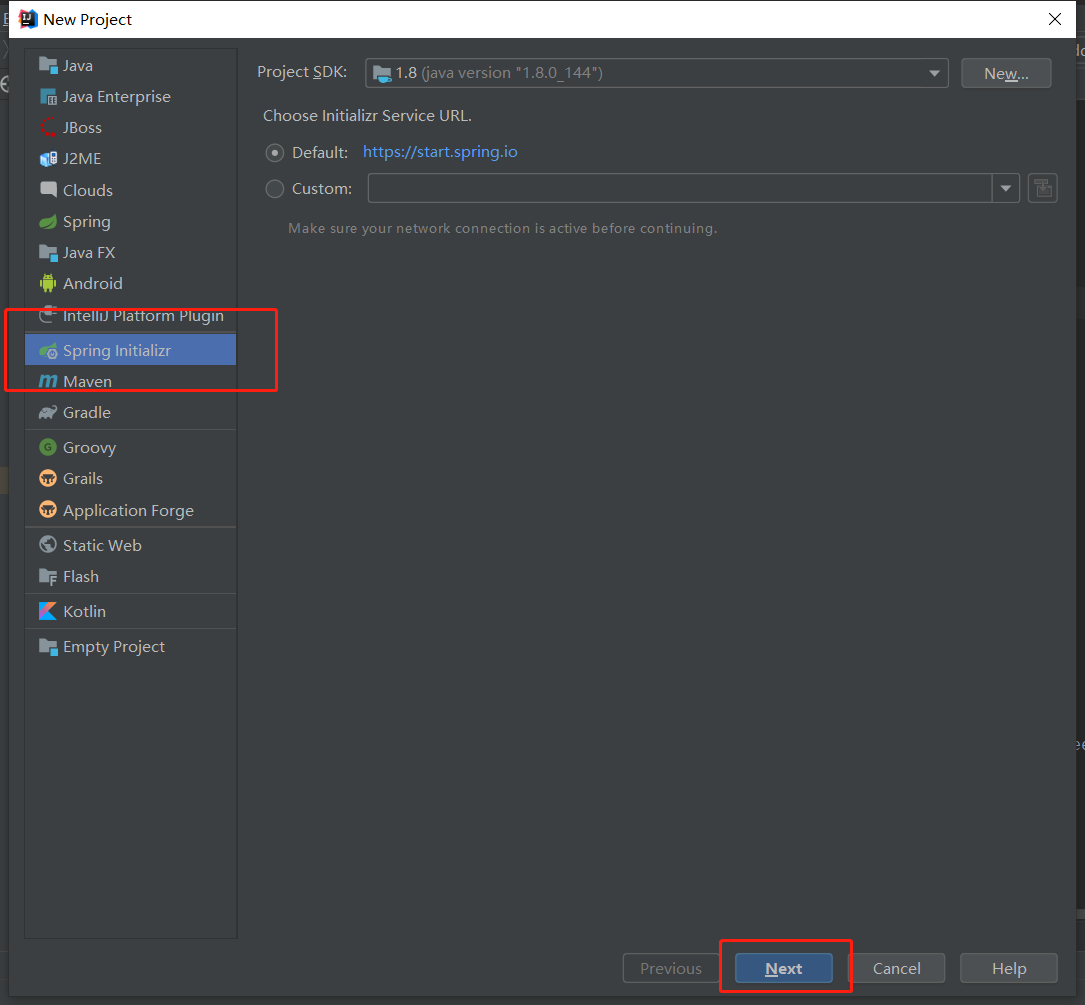
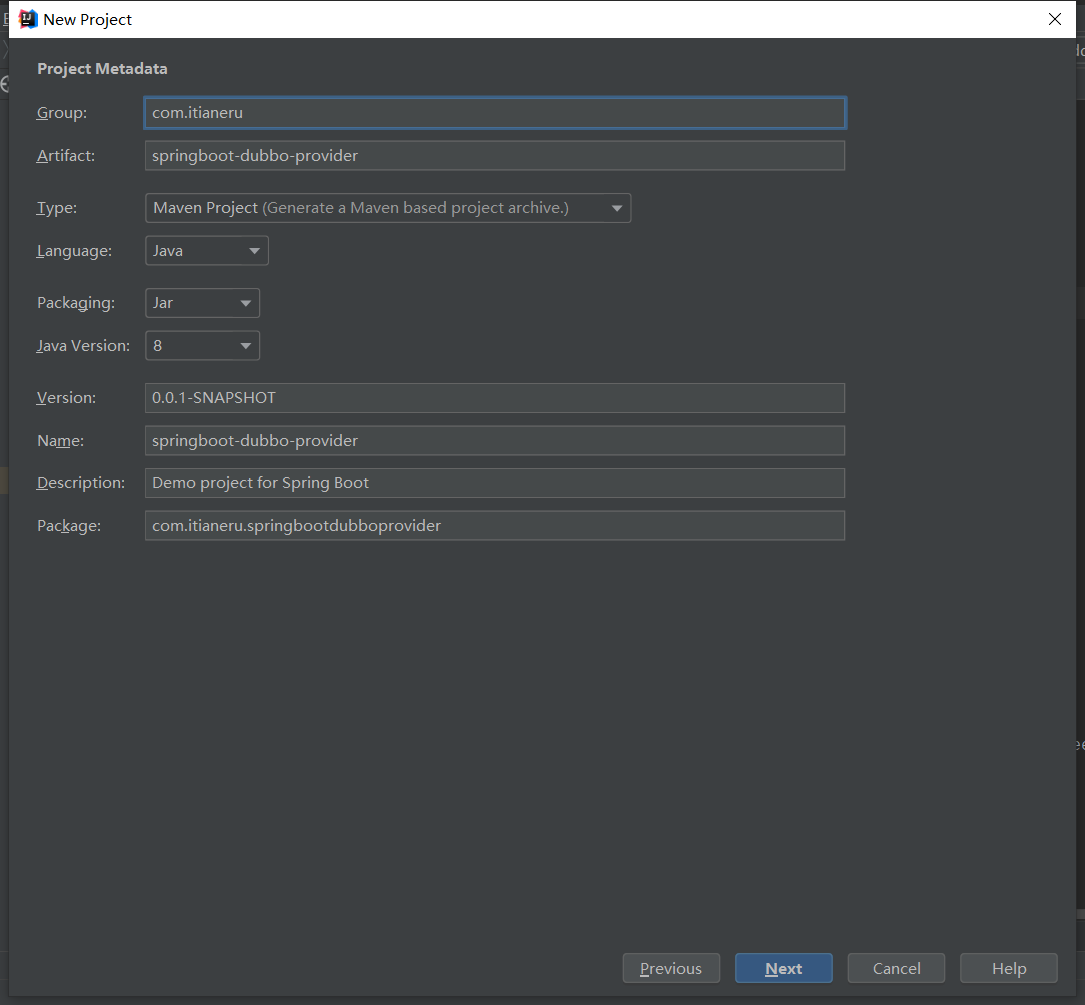
开发环境 jdk1.8，zookeeper 3.4，idea，apache-tomcat-7.0.93-dubbo-admin

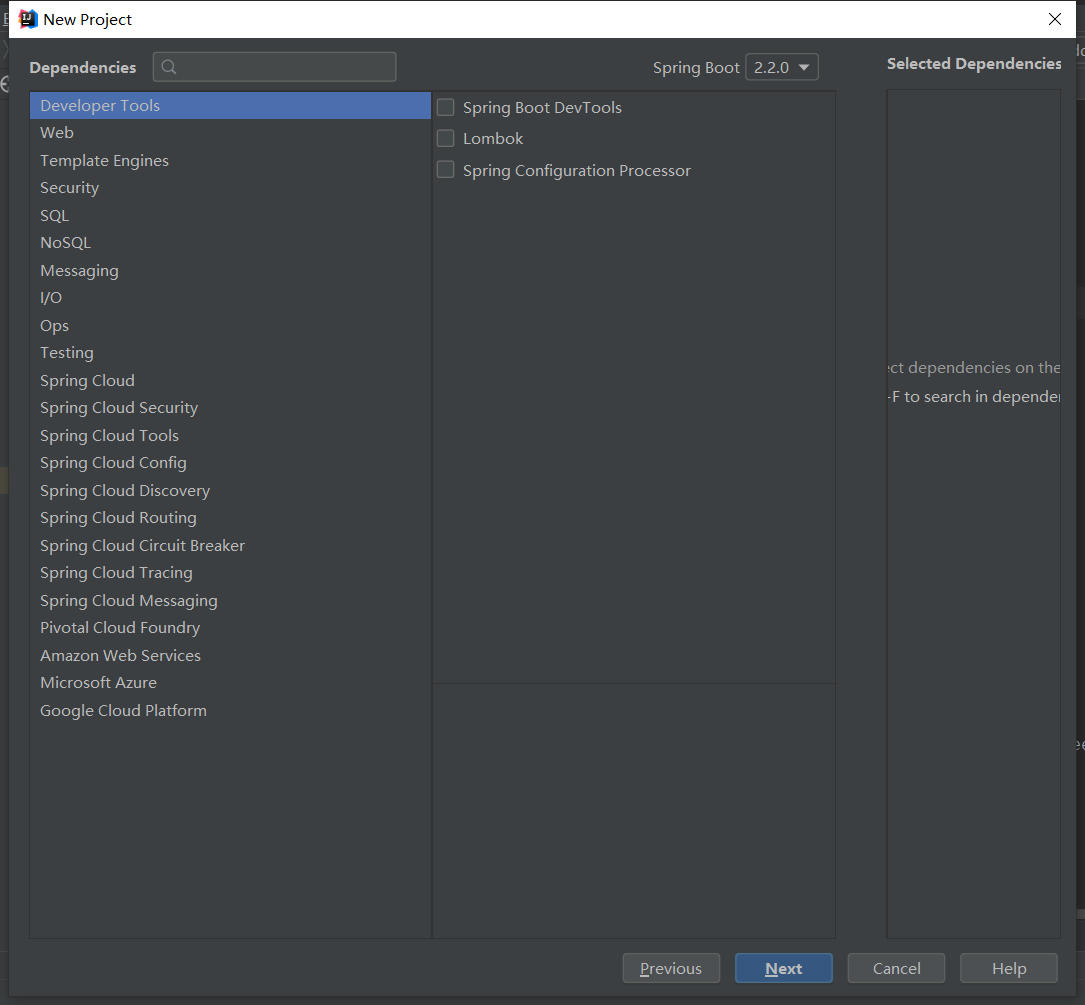
1. 创建提供者工程provider
2. 选择Spring Initializr，Project SDK选择1.8，点击next



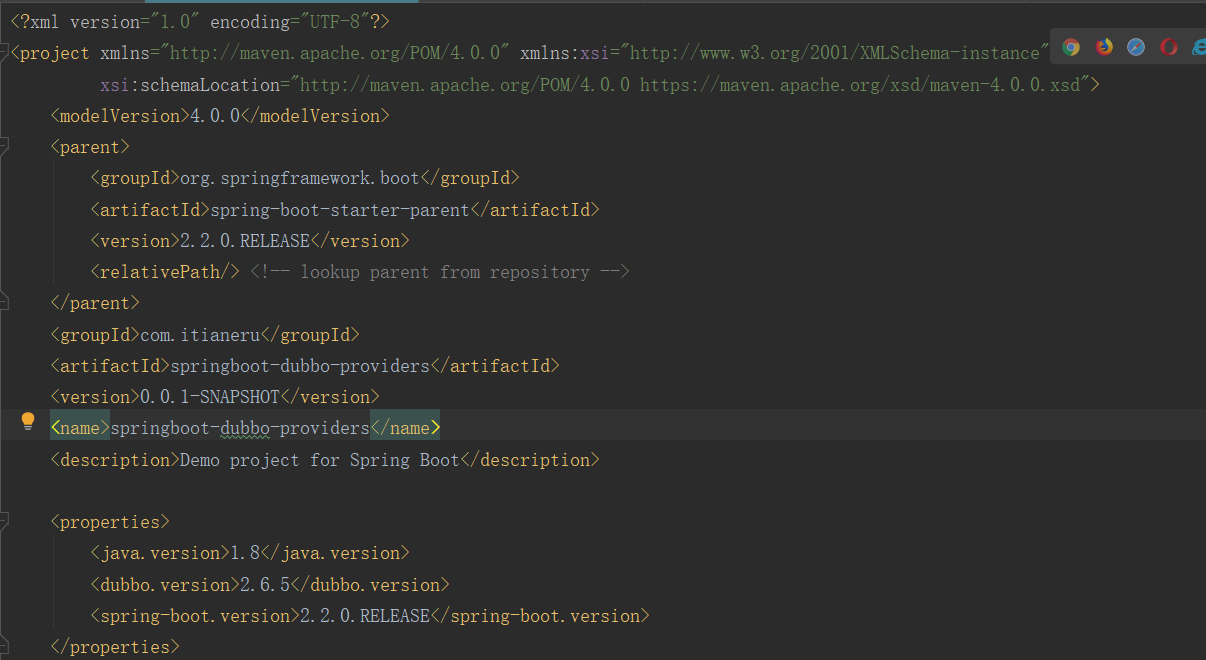
1. 填写工程名称（名称随便，最好带有provider），点击next

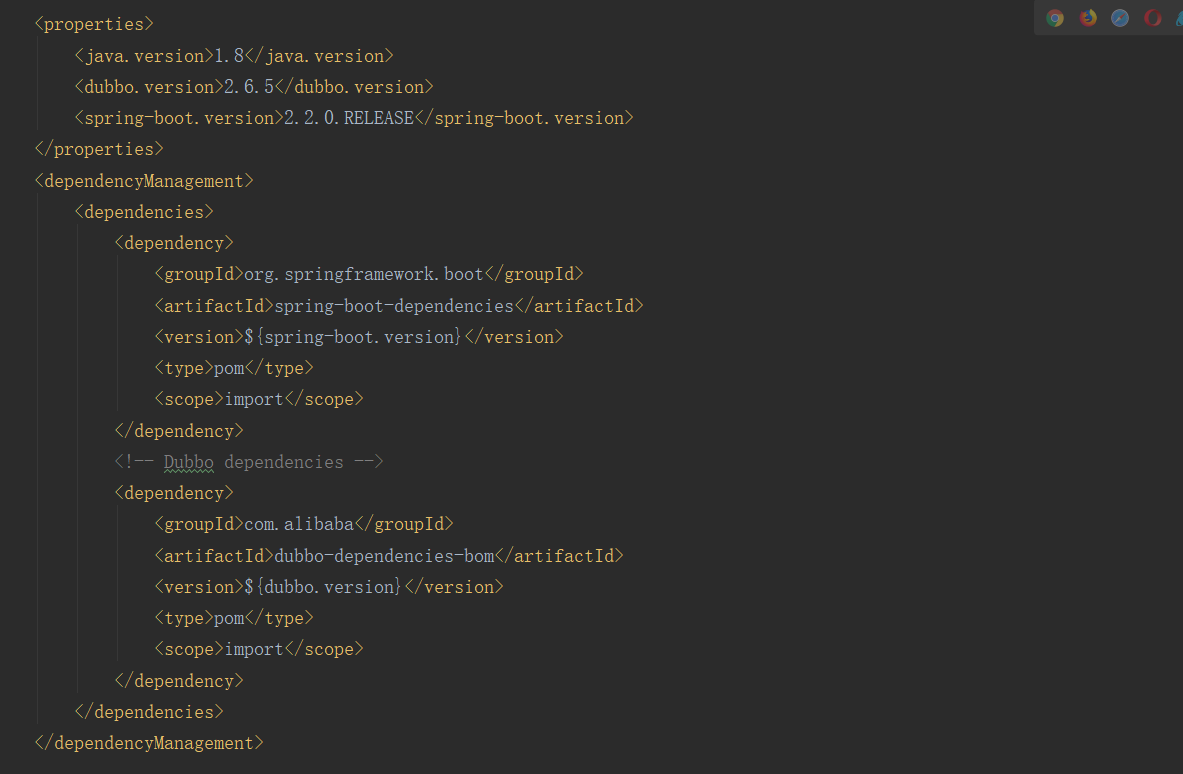


1. 不选依赖，直接下一步，后面点Finish直到完成创建

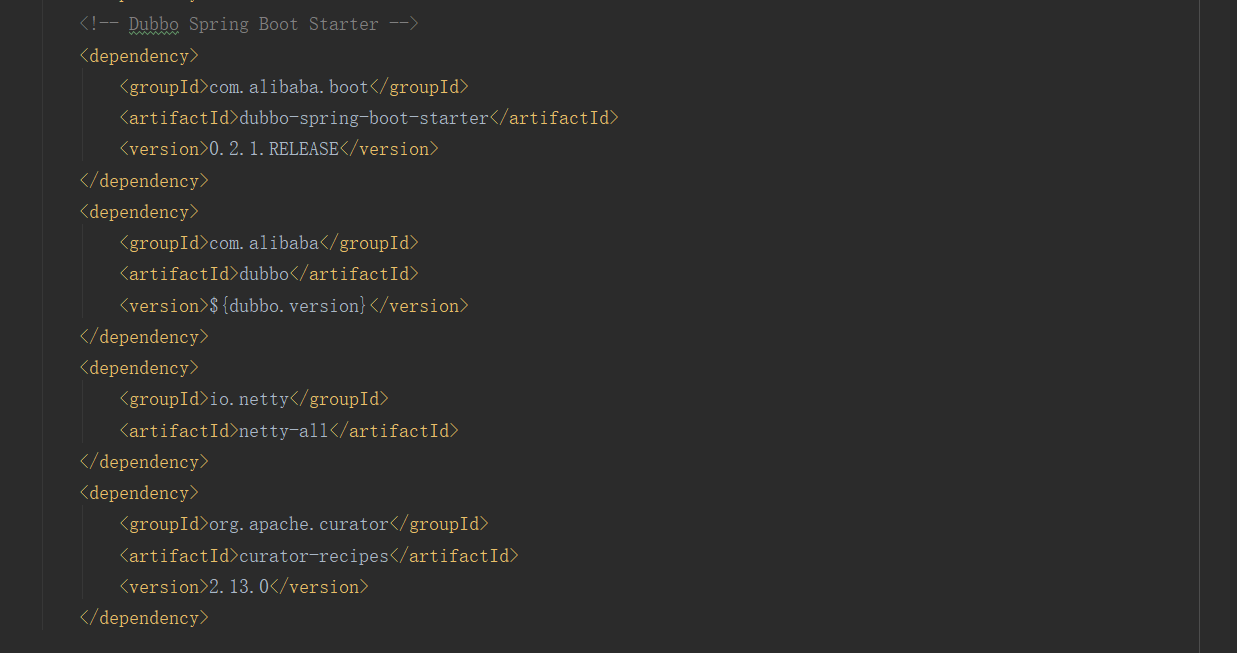


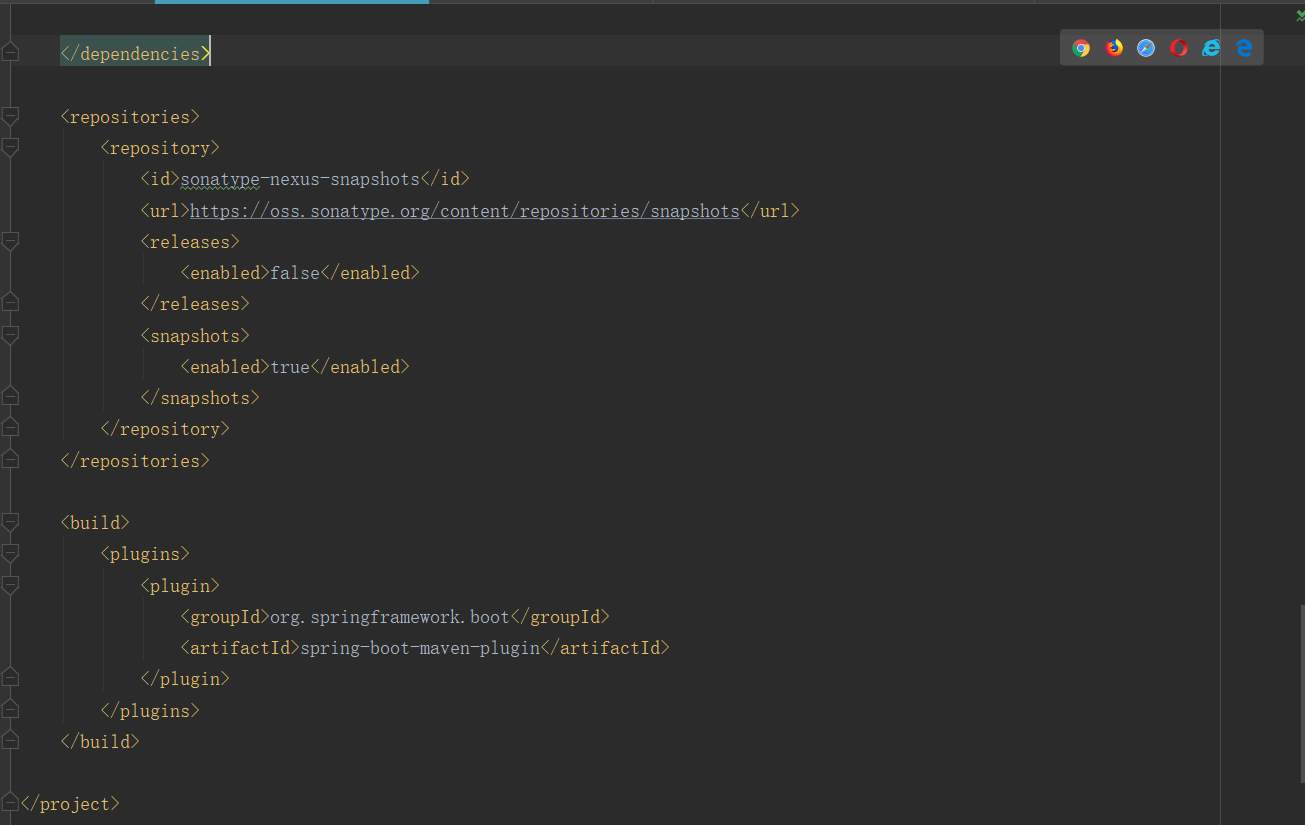
1. 选择pom.xml文件，添加dubbo依赖



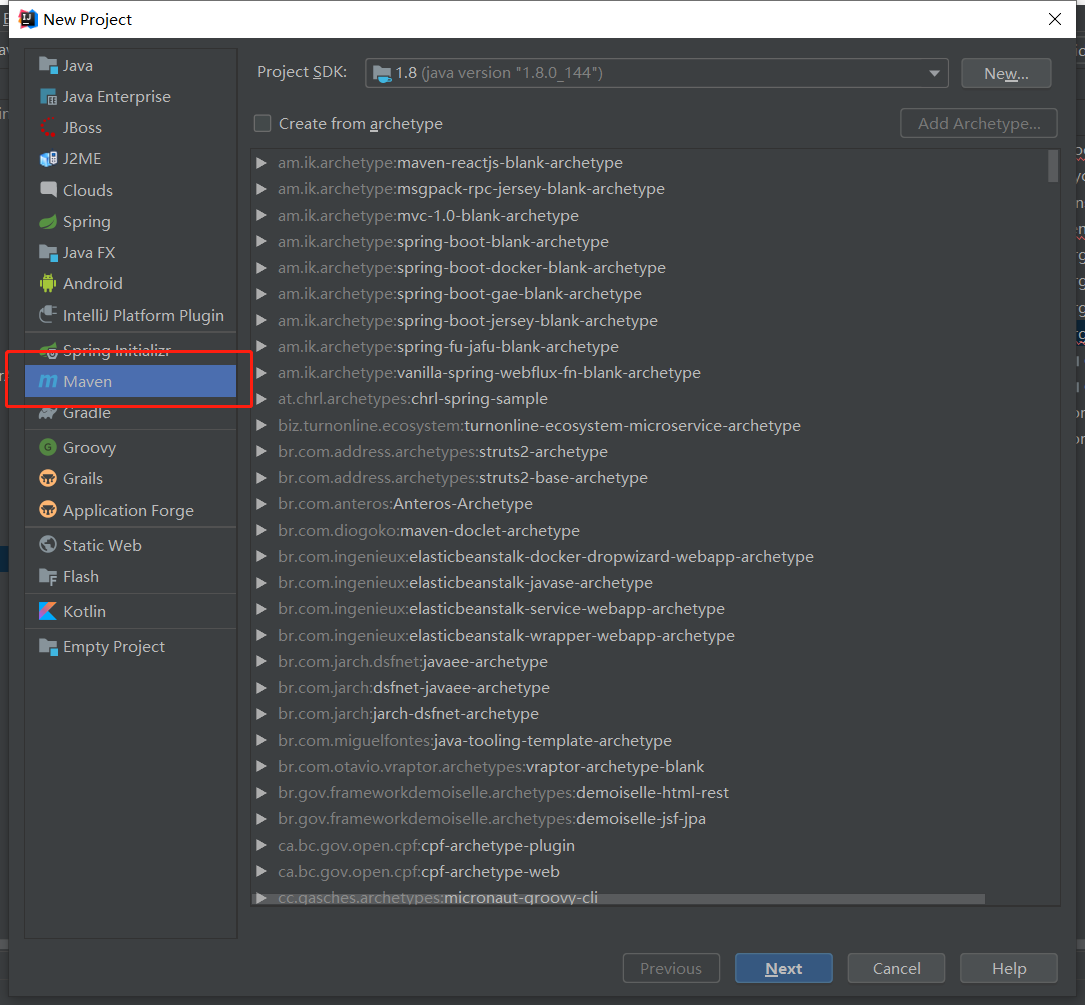




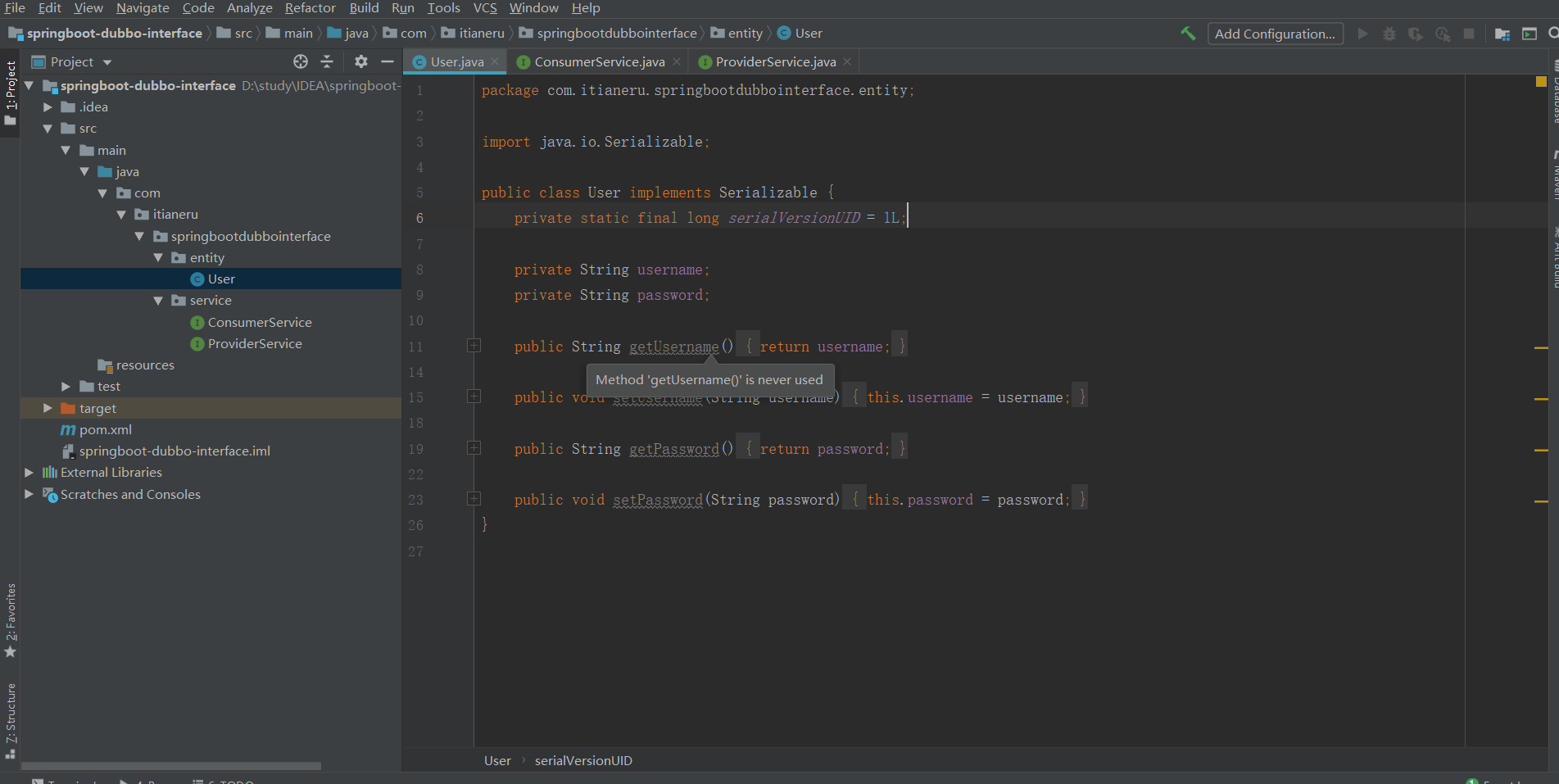


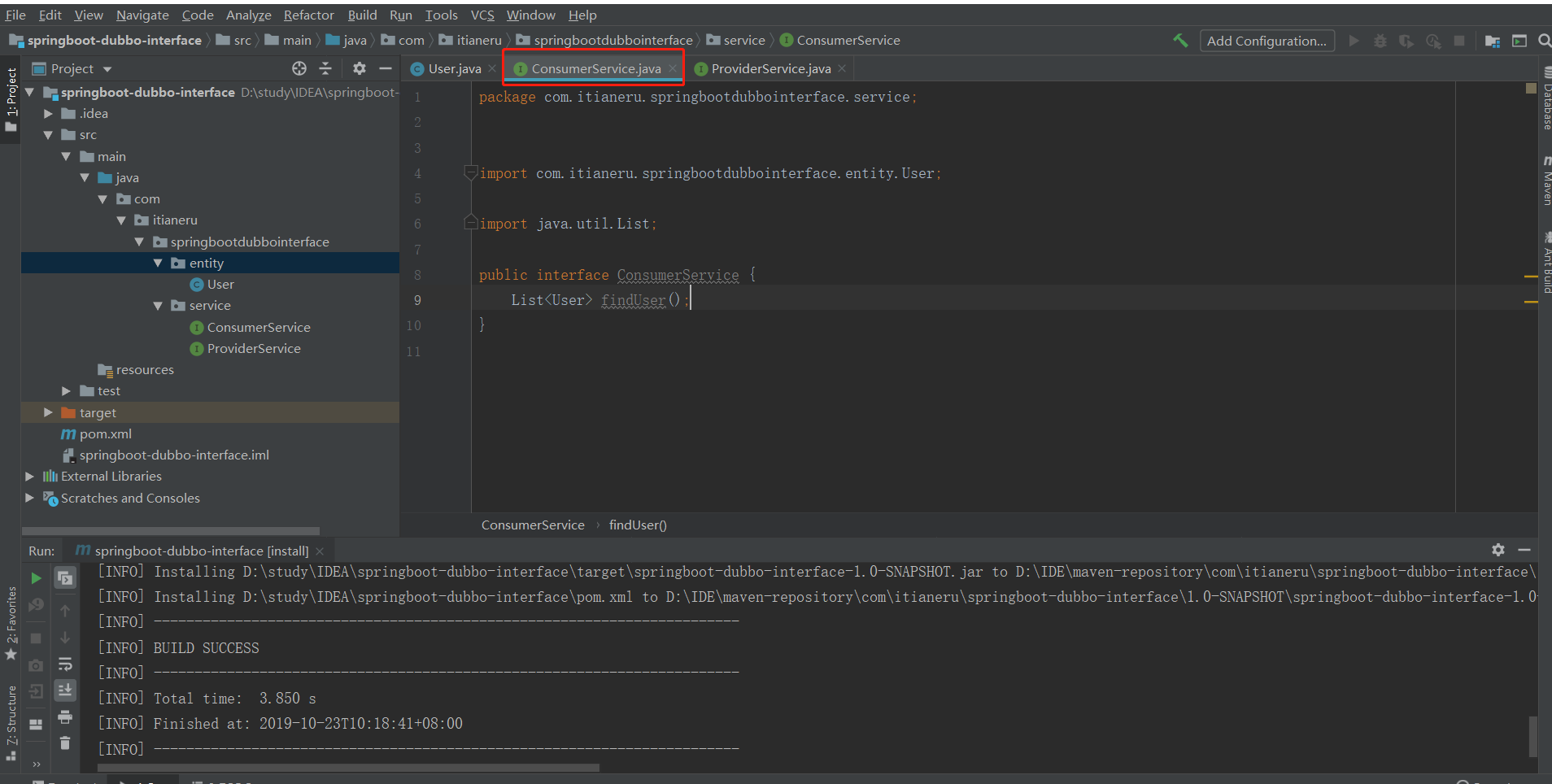


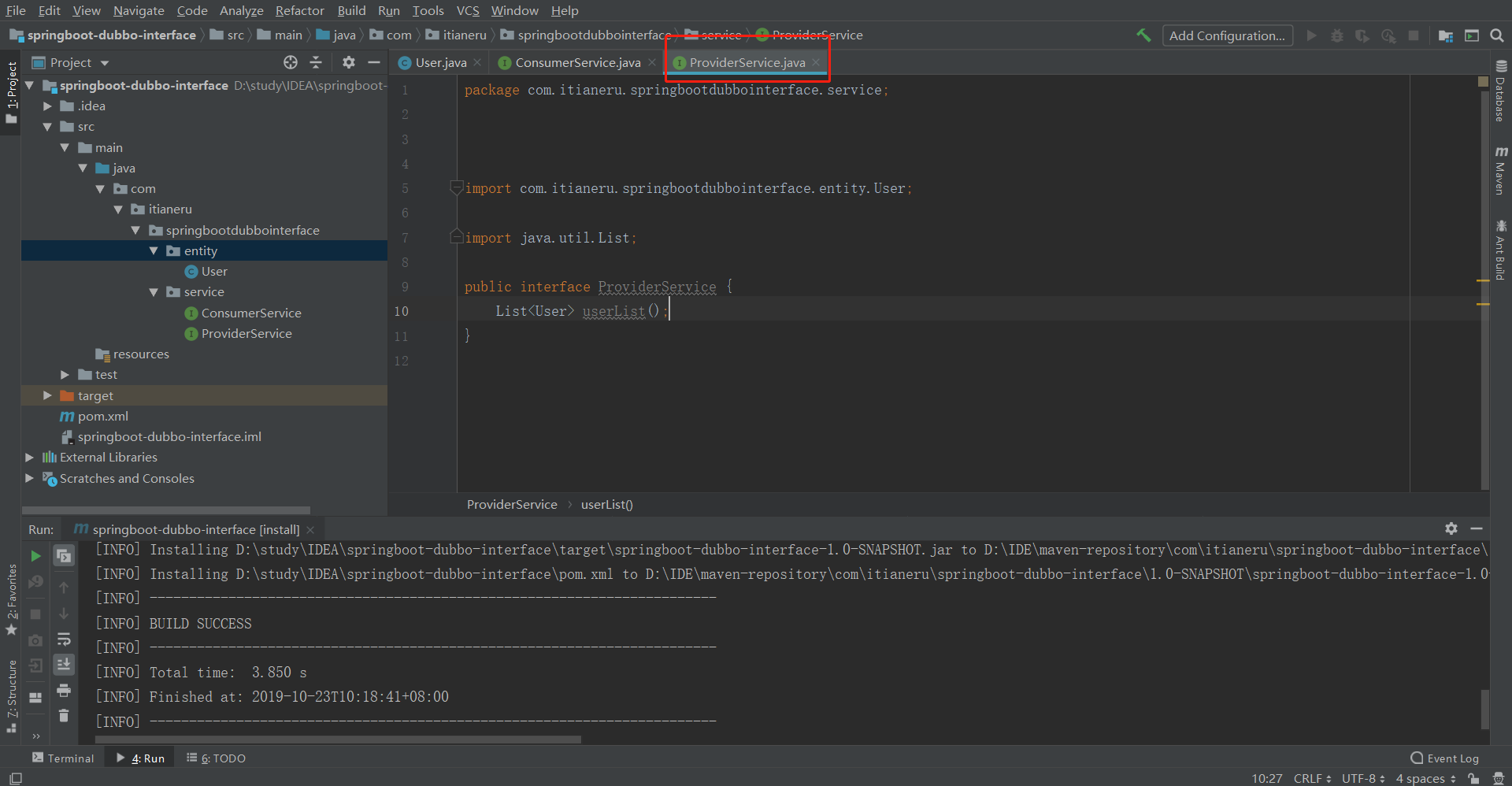
1. 创建接口工程interface，完成创建



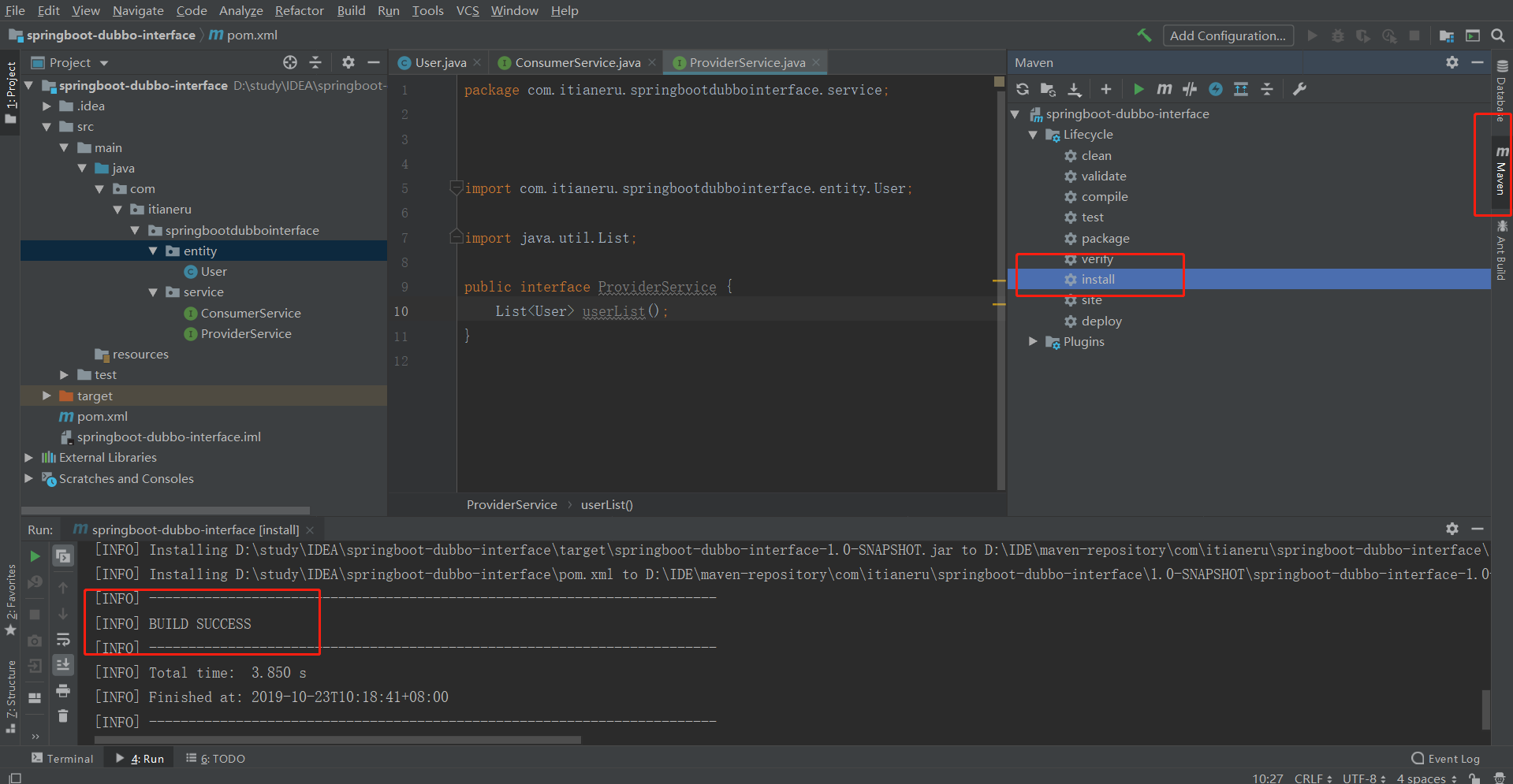
1. 在新建的接口工程，创建一个实体类User，两个接口ProviderService，ConsumerService



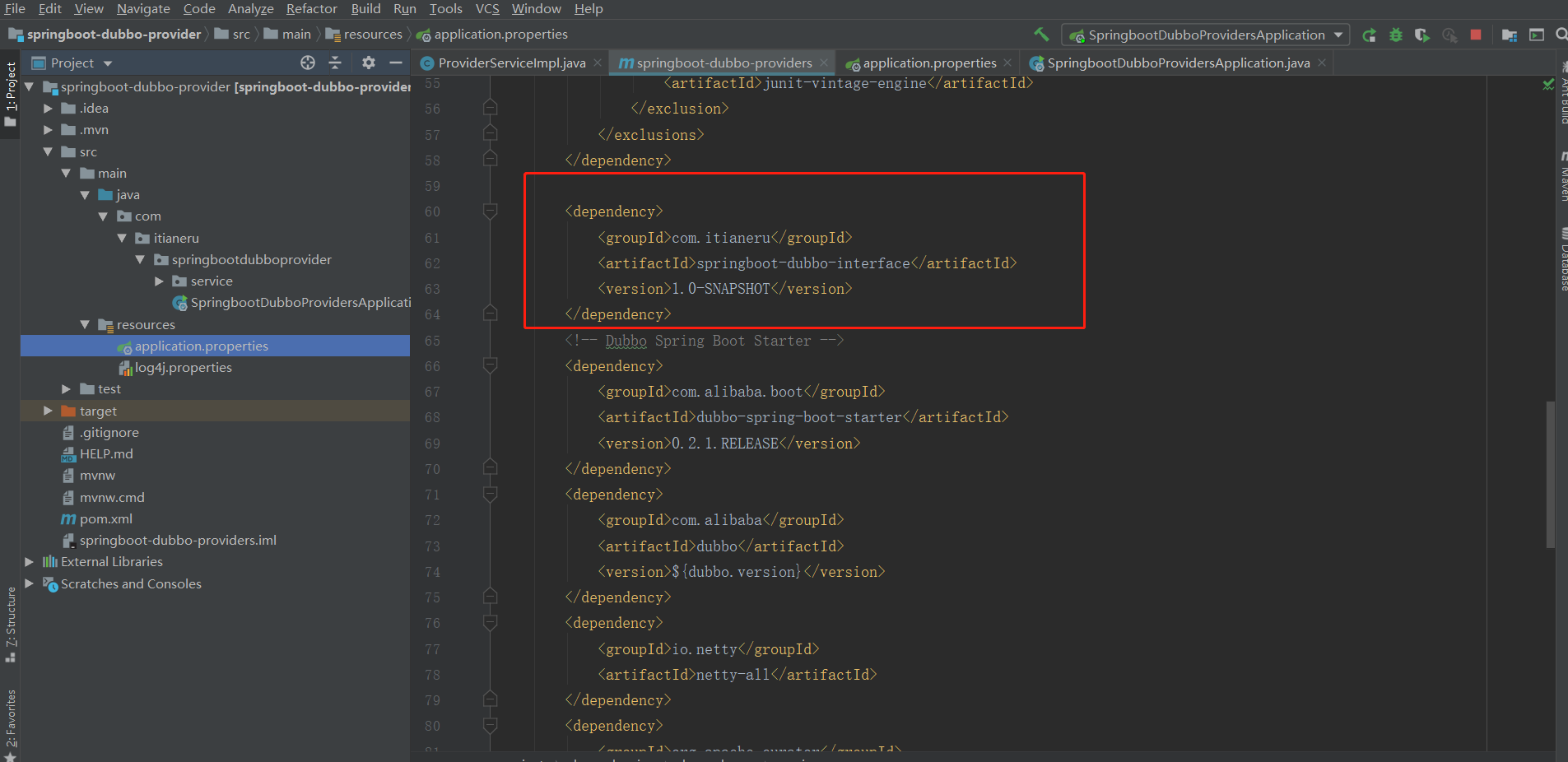




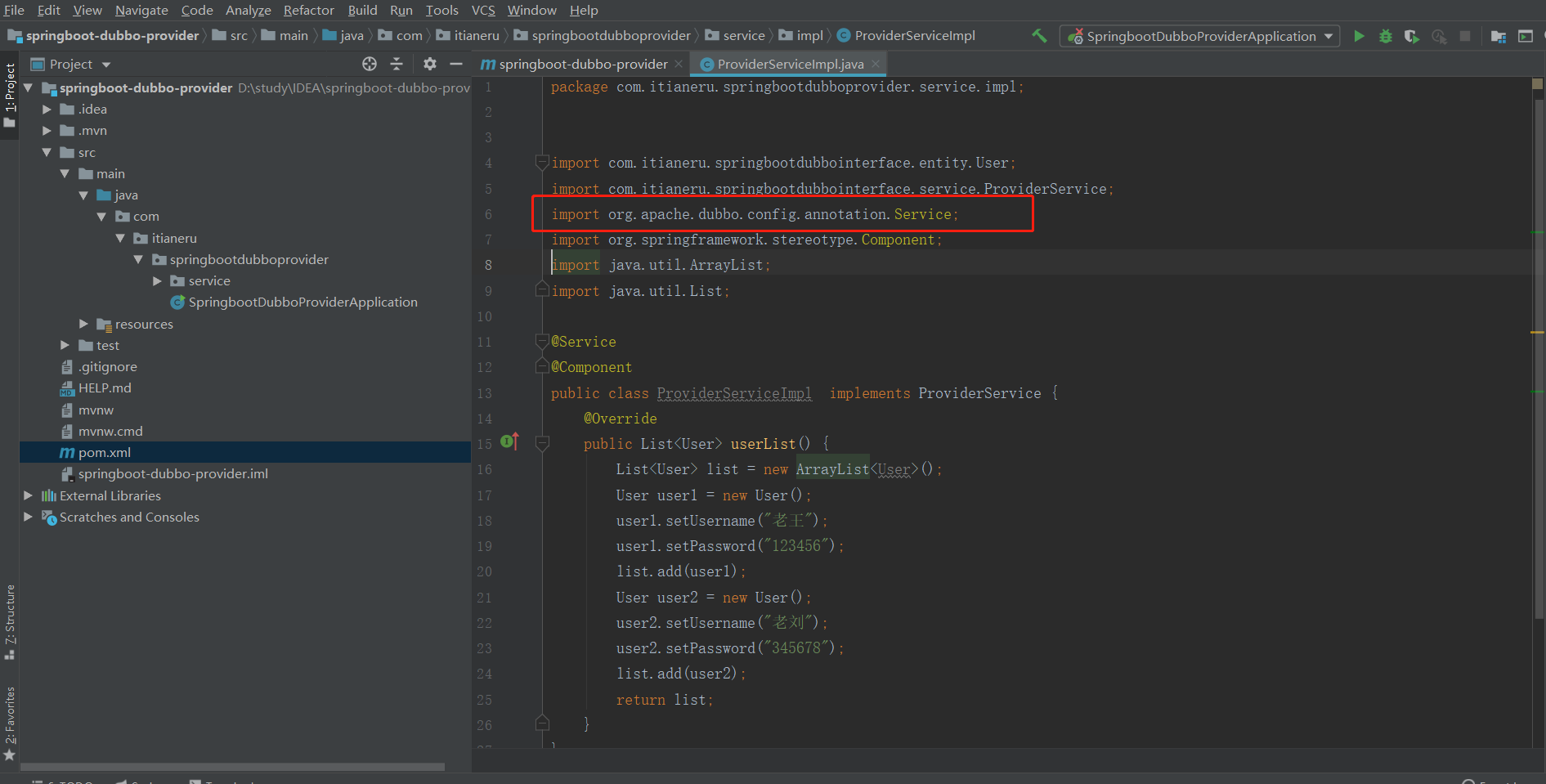
1. 使用maven将接口工程打成jar包，打的jar包会进入本地的maven仓库



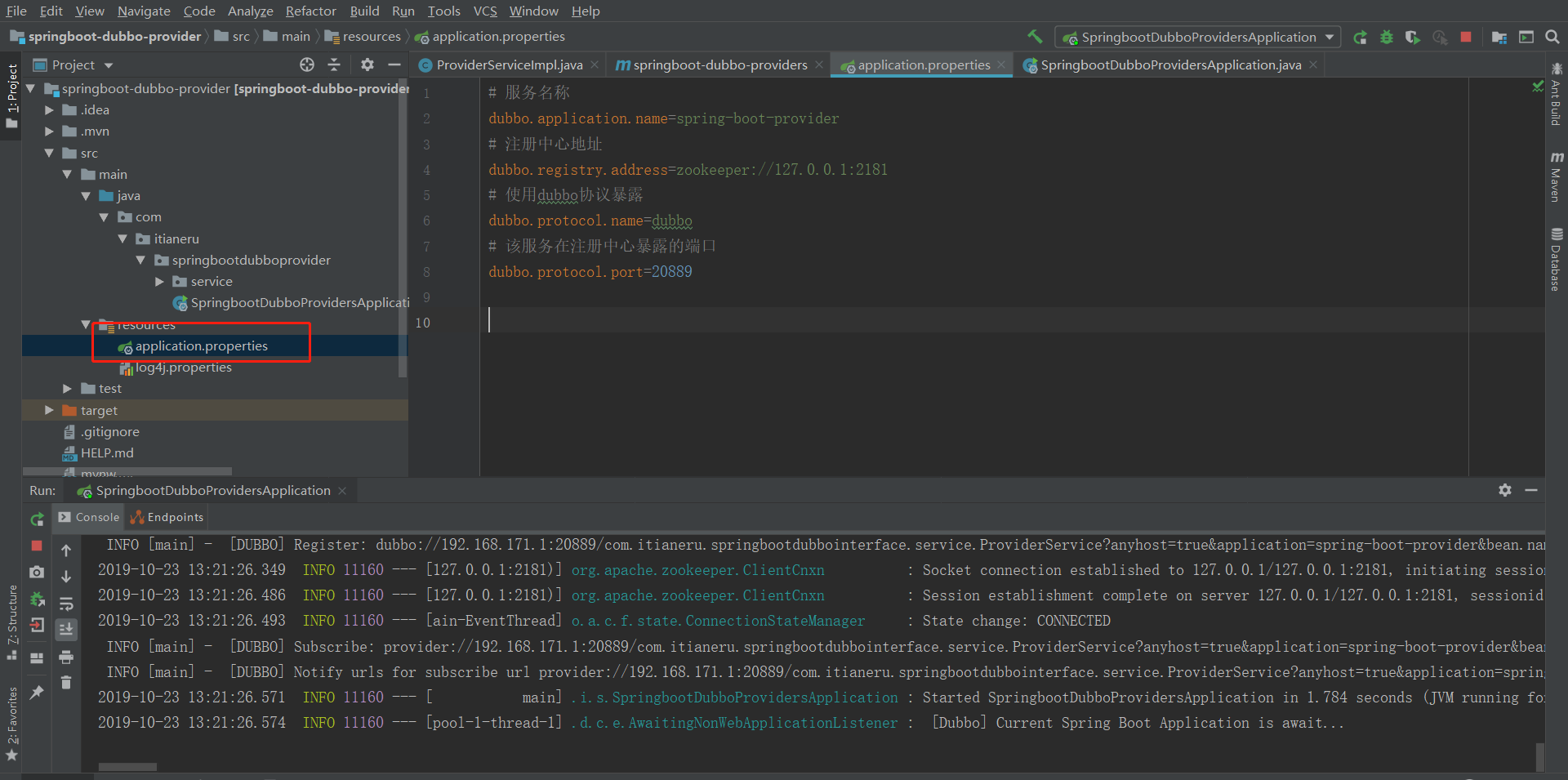
1. 打开之前创建的provider工程，添加接口工程jar包依赖



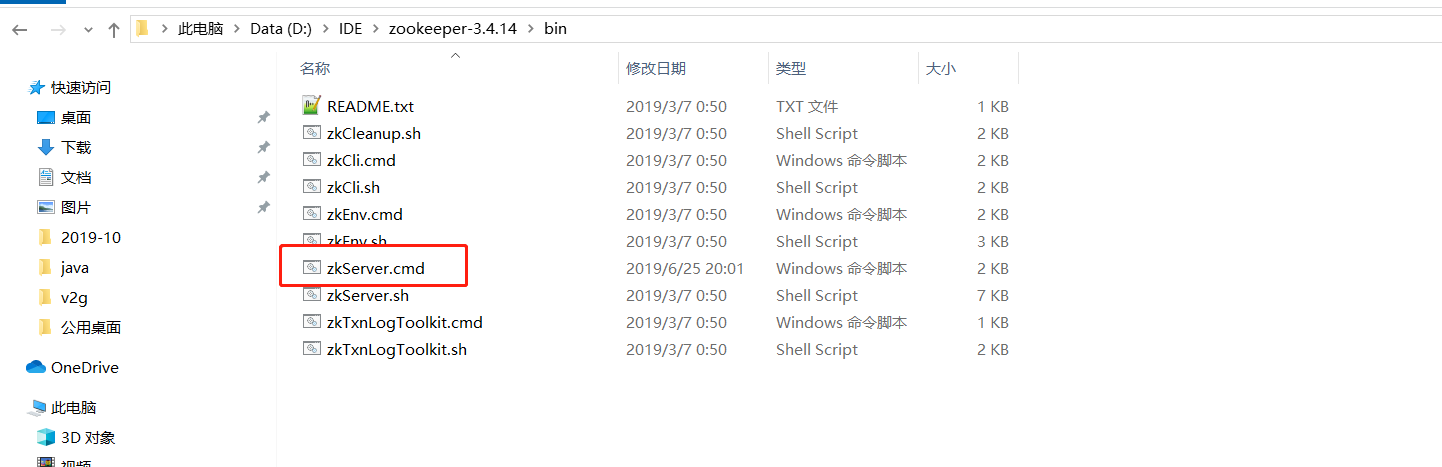
1. 创建接口的实现类，ProviderServiceImpl，实现接口工程中的接口，注意此处的@Service注解非spring的注解，而是dubbo的注解，功能是，将该服务暴露给zookeeper注册中心，供其他项目调用



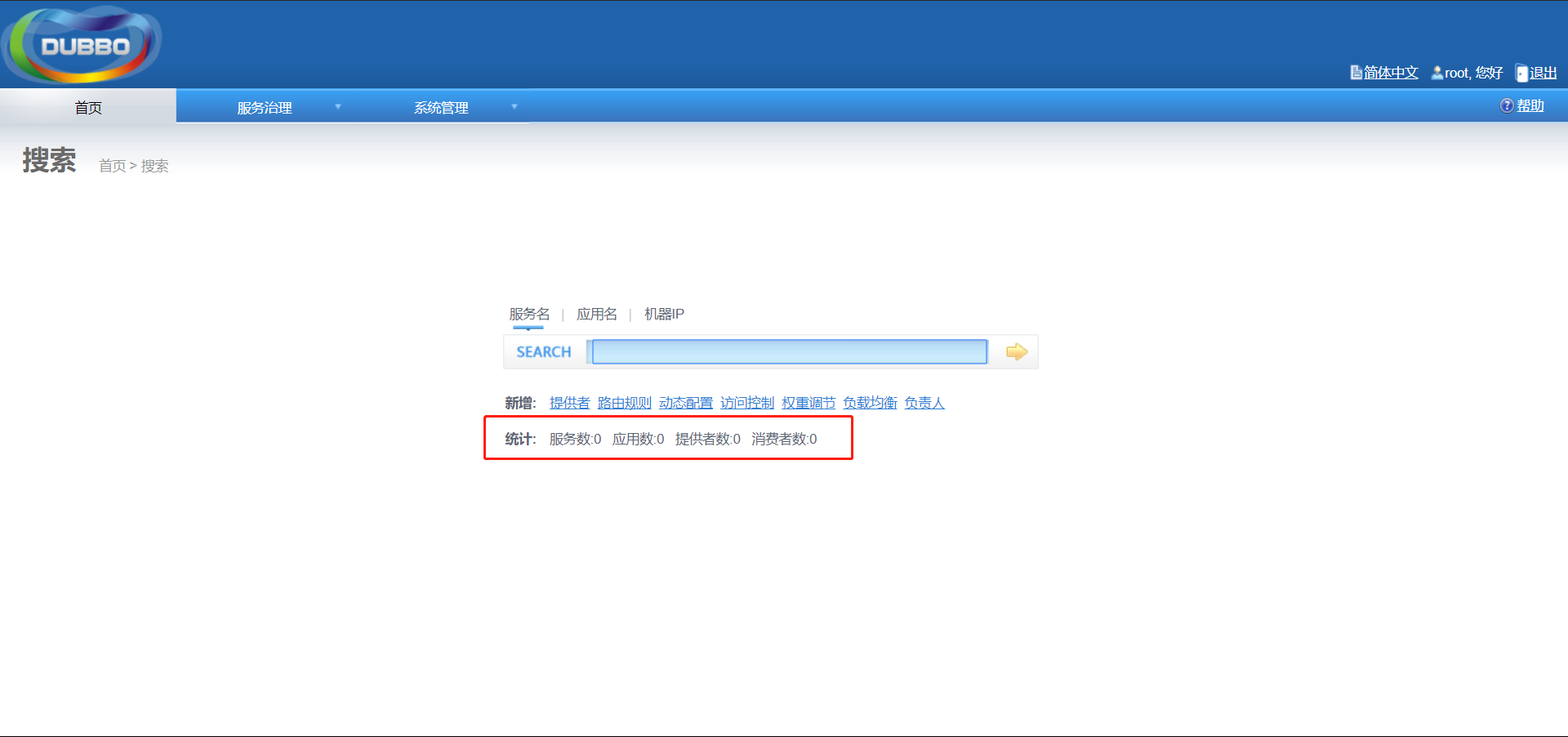
1. 在资源文件配置dubbo，添加log4j.properties，网上随变找一个就行。



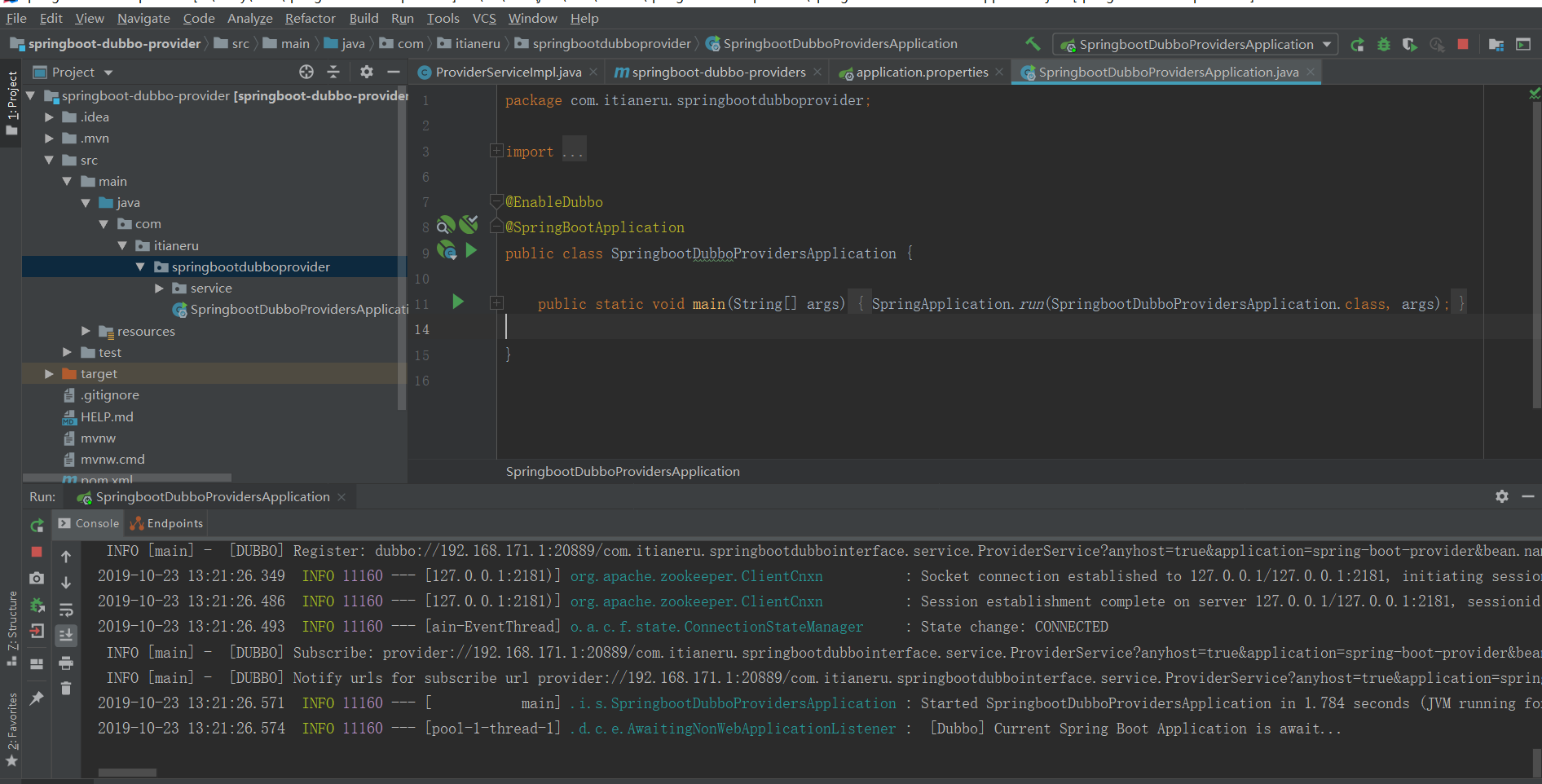
1. 启动项目测试，先启动zookeeper，点击zookeeper/bin/zkServer.cmd

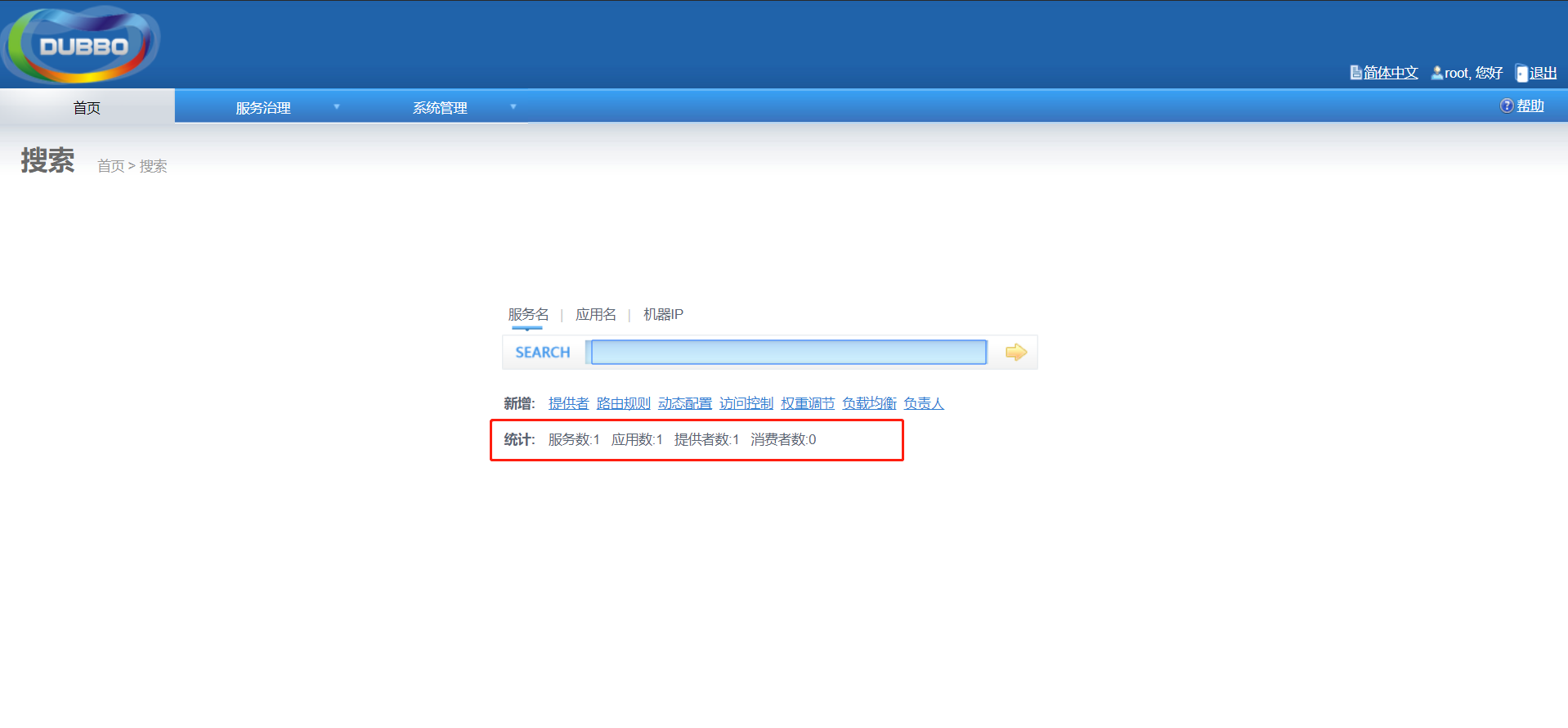


1. 启动dubbo-admin，点击apache-tomact-dubbo-admin/bin/startup.bat,启动后在浏览器输入<http://localhost:8088/>，账号&密码：root，root，进入管理平台，此时服务数等都为0.

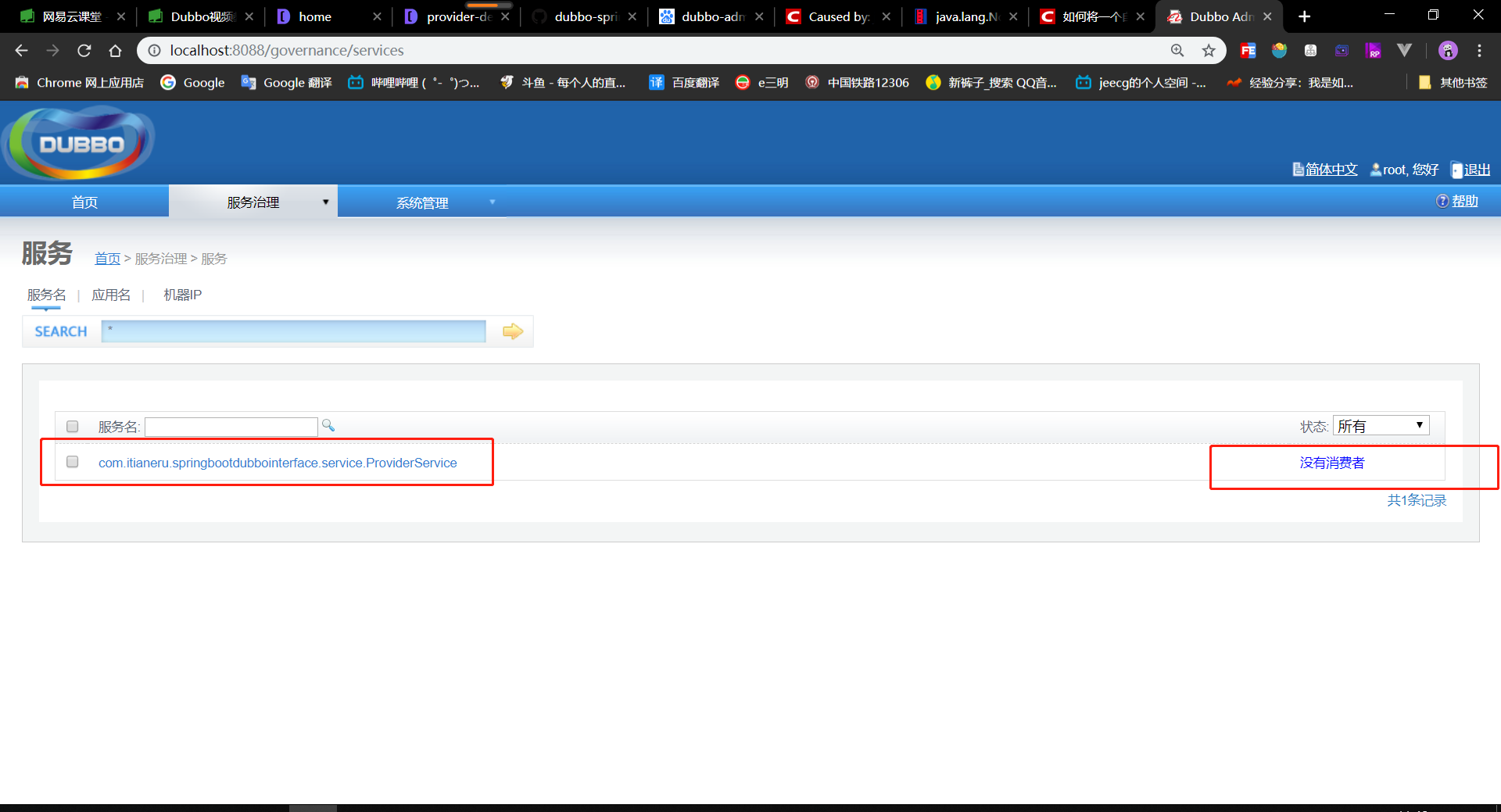


1. 在启动类上加入注解@EnableDubbo，然后启动provider项目，刷新dubbo-admin页面，下面的服务数变为1

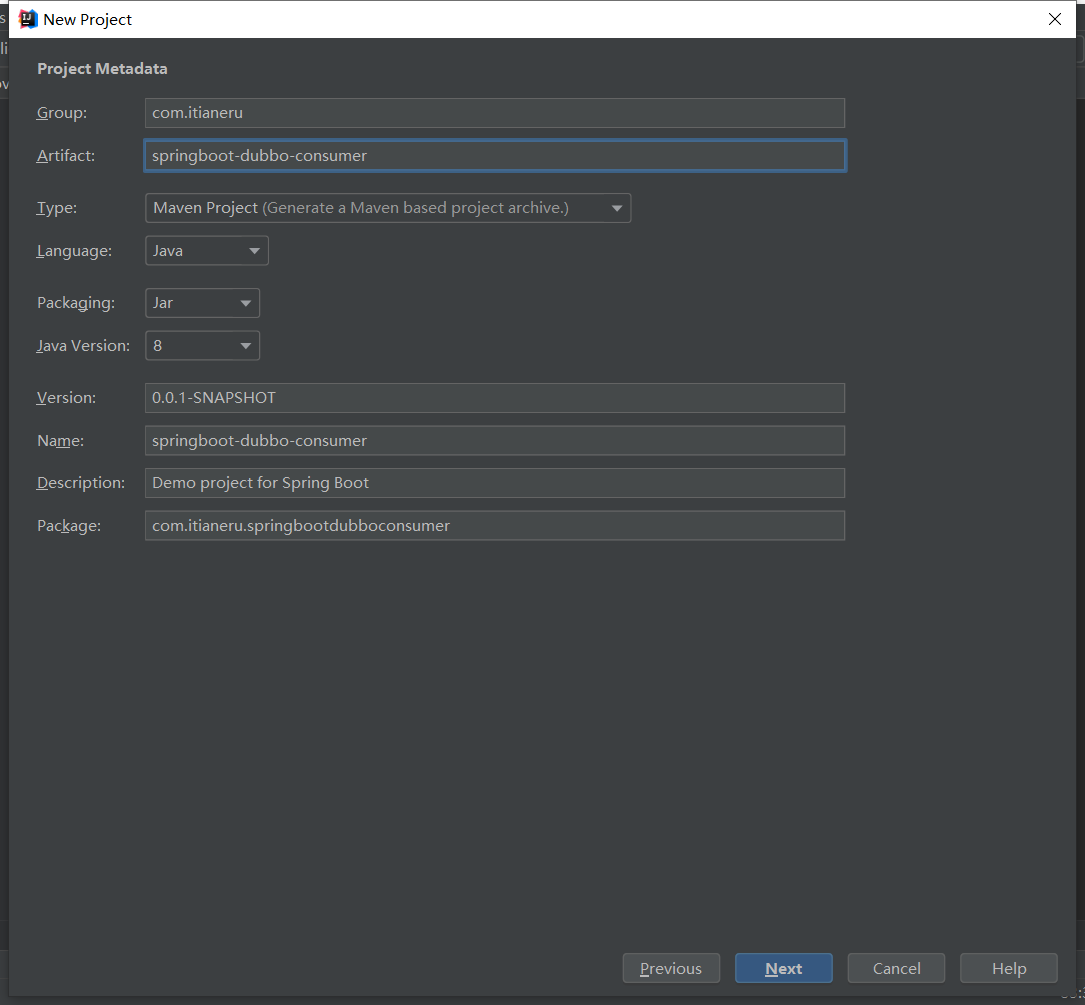


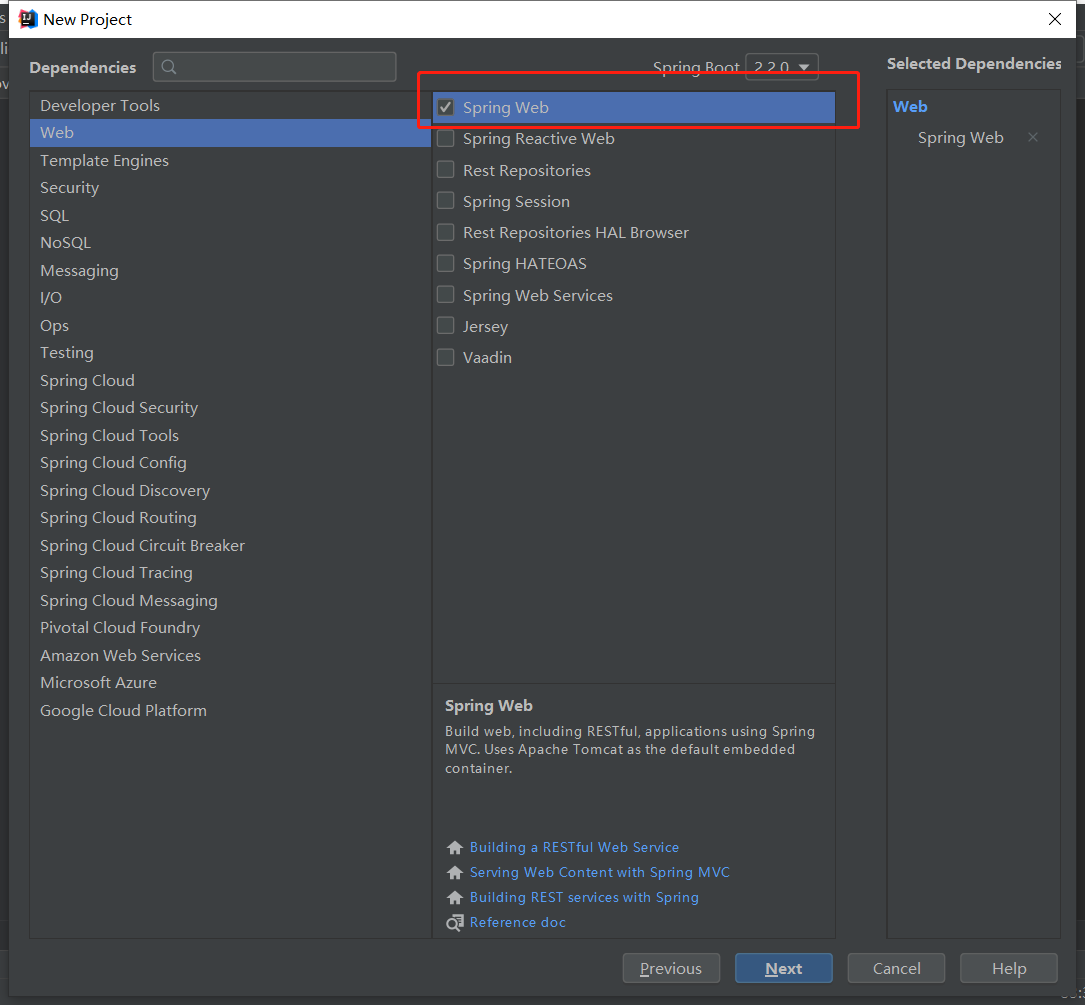


1. 在服务治理，选择服务，可看到我们注册的服务名，目前该服务没有消费者

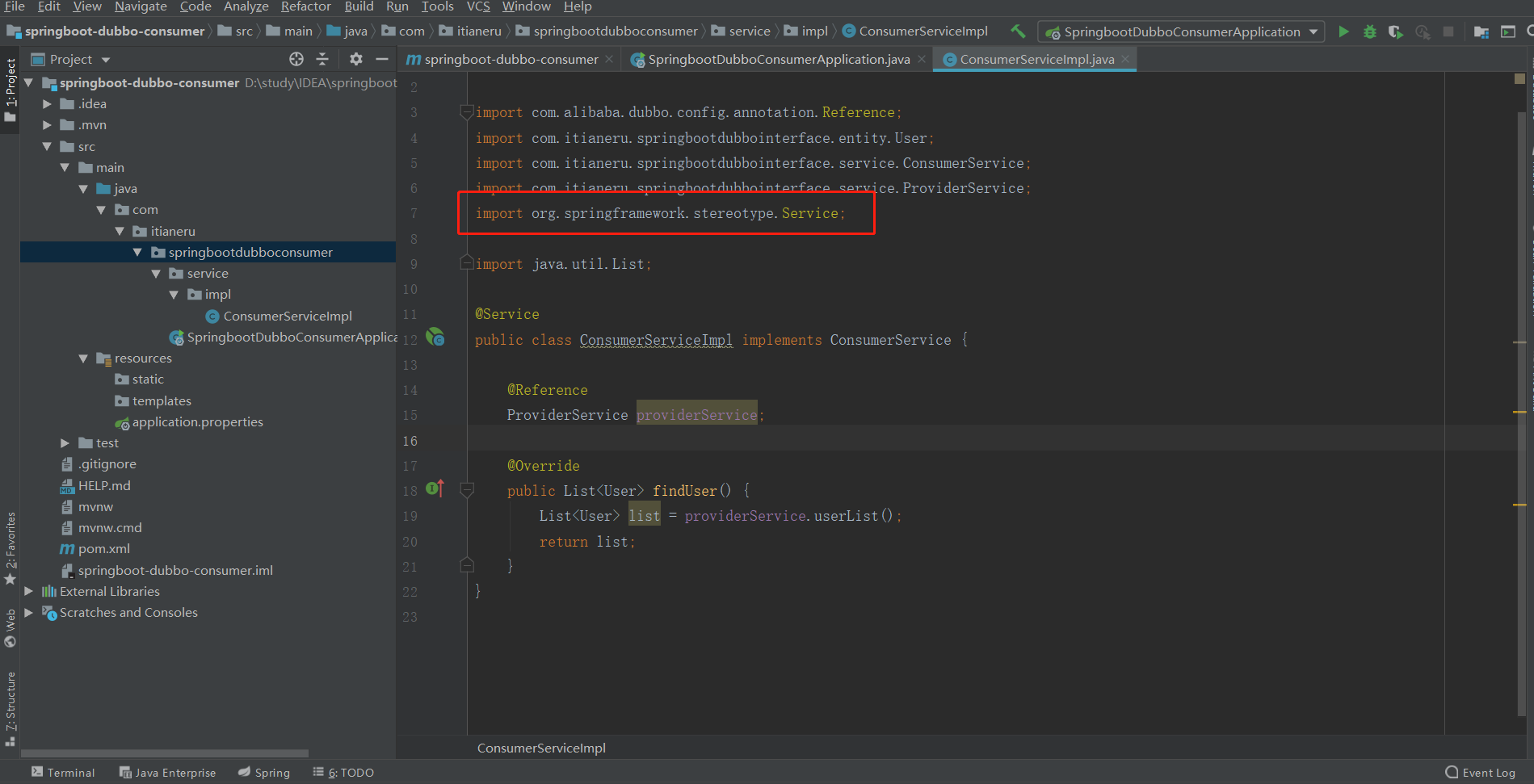


1. 接下来创建，消费者工程consumer，依赖中选择web，完成创建

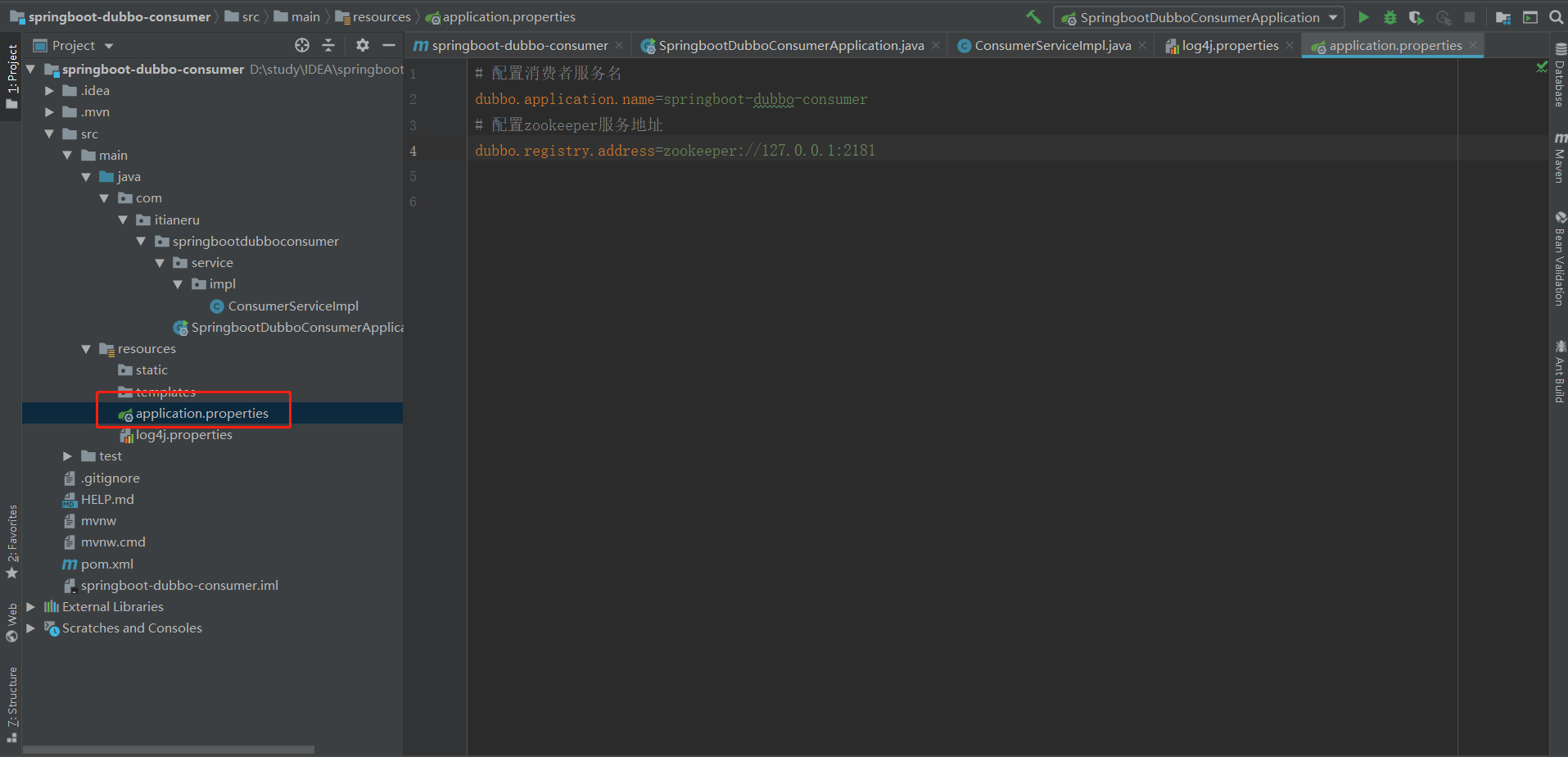




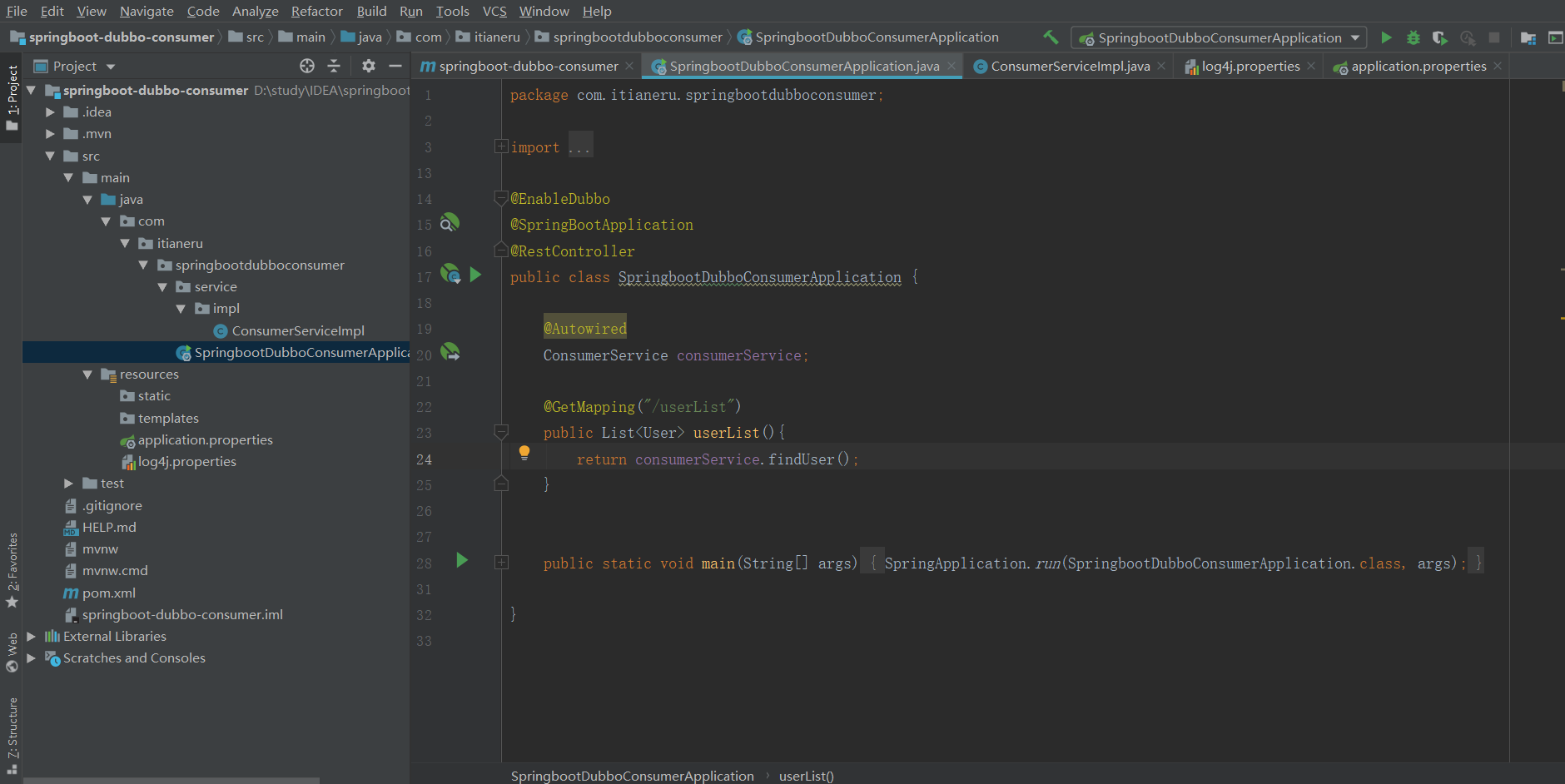
1. 在pom.xml中引入dubbo和接口工程依赖，和provider的相同(多了一个web的依赖)
2. 创建ConsumerService并实现对应接口，其中@Service是spring的注解不是dubbo的注解，@Reference会从注册中心找到对应的接口，注入进去，与dubbo的@Service注解对应。



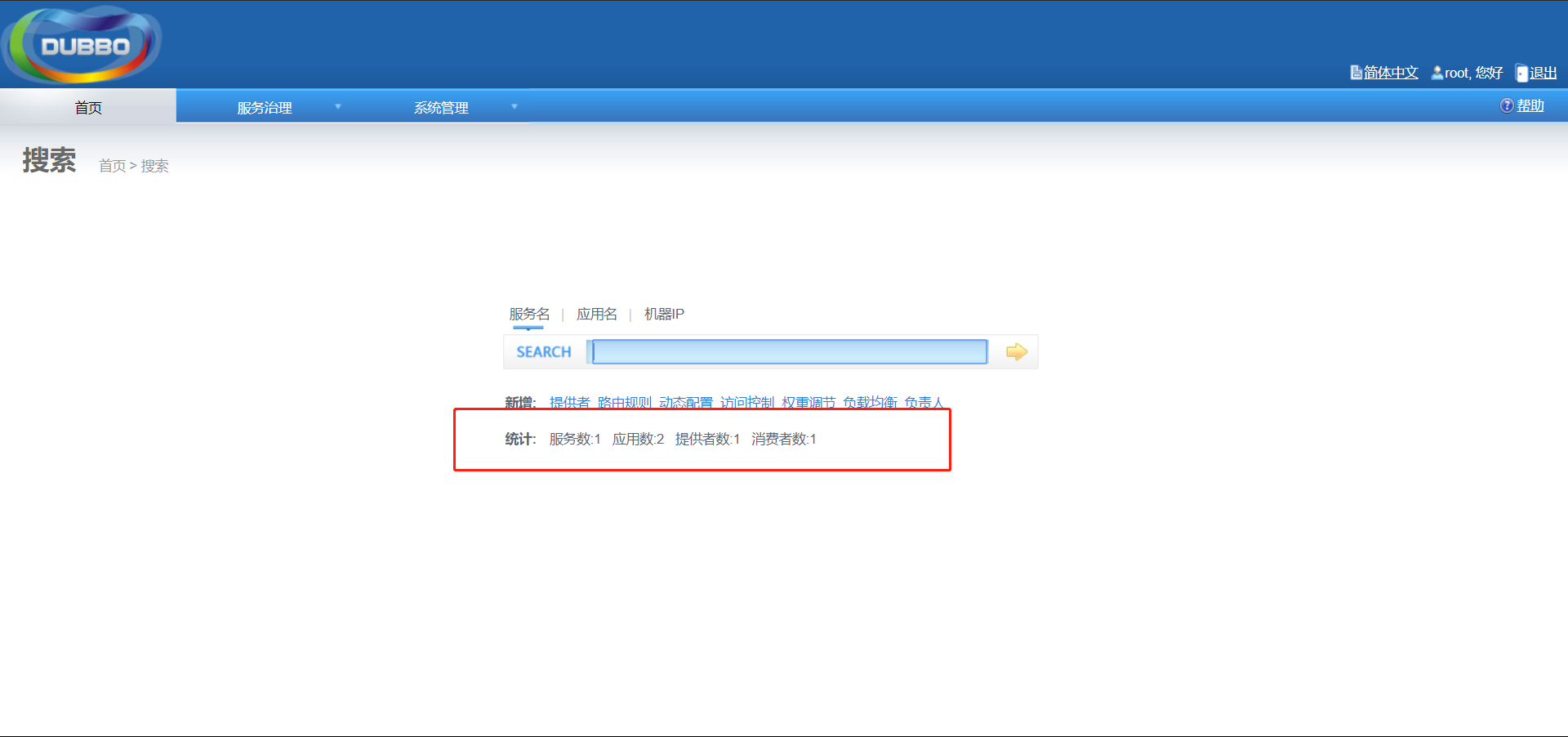
1. 在application.properties添加dubbo配置，添加log4j.properties配置文件

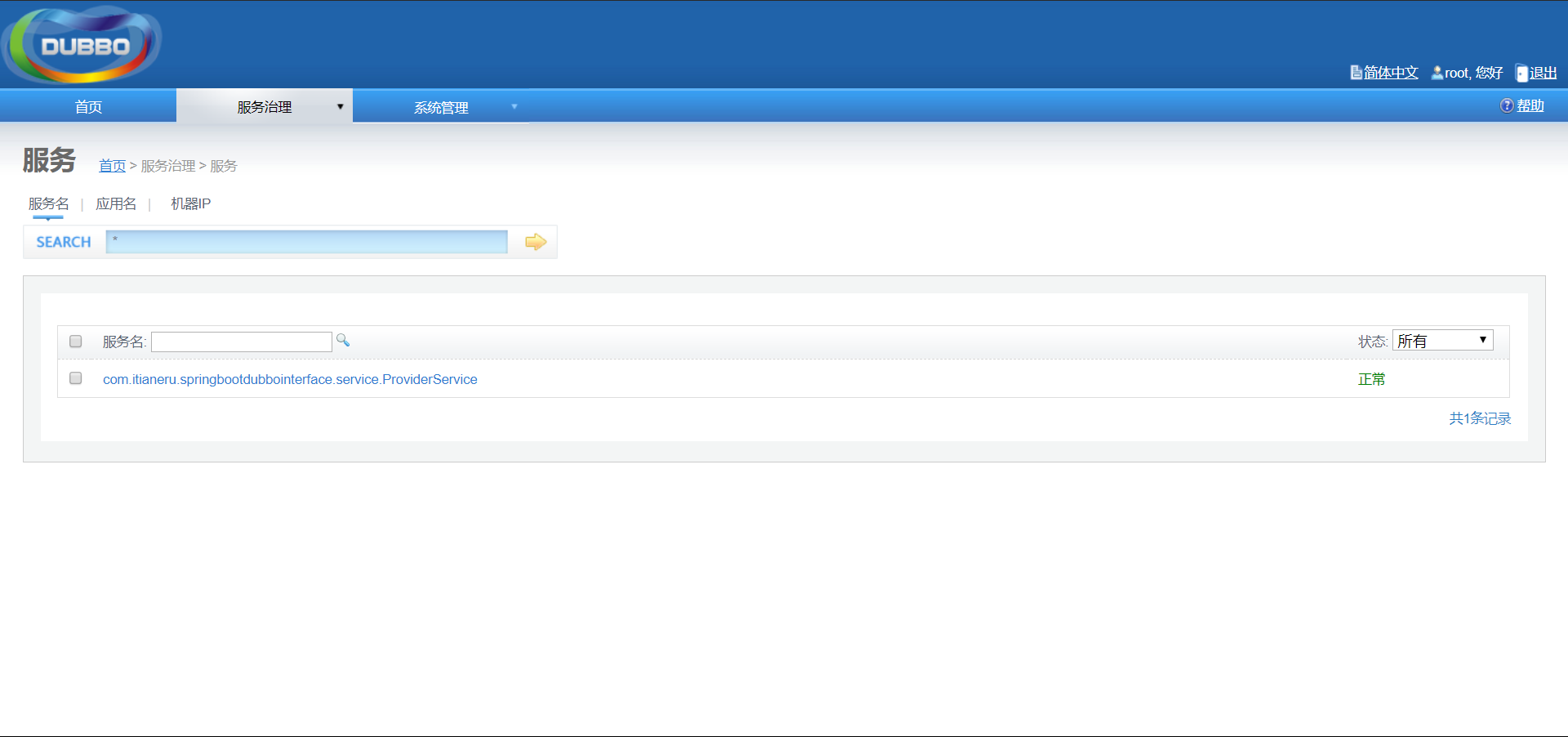


1. 在启动文件，添加@EnableDubbo注解开启dubbo，添加controller相关注解



1. 启动工程，刷新dubbo-admin，消费者数量变为1，前面注册的提供者服务的状态由没有消费者，变为正常，说明该服务，被后创建的消费者消费。





1. 在浏览器输入<http://localhost/userList>，测试服务，成功返回结果

