maven

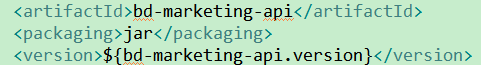
服务器对内接口

假设有个一个订单服务 OrderService，可能app、h5基本的订单信息数据来自于此台服务器。是通过外网http请求。

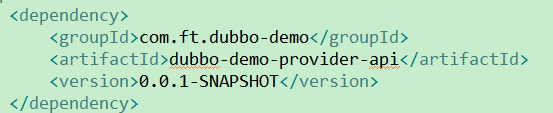
当然系统大了，模块增多，比如现在财务系统就需要查询订单系统中某个时间段的交易额，我们显然肯定是通过某种协议走内网。

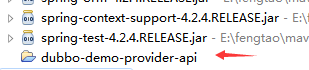
这个时候我们可以这样，我们生成一个项目OrderService-api，定义一个接口RemoteOrderService，里面定义各种对外（其他模块系统）的方法，抽象出所有的方法之后，我们mvn install打包到本地，然后我们可以在OrderService引入这个jar包，并增加对应的实现类，最后我们在需要RemoteOrderService的方法时，我们也只需要外部引入这个jar包，然后通过某些协议(如hessian、dubbo)拿到对应的服务，这样我们就可以在任何服务器、任何模块都能访问OrderService的服务。并且是通过内网，效率就得到很大提升

流程：

创建OrderServiceApi模块，定义好所需要的数据类型 bean 和对外方法，，打包方式以jar形式。

定完所有方法之后，右击pom.xml文件，执行打包到本地库命令 maven install，为之后其他模块引用做准备。



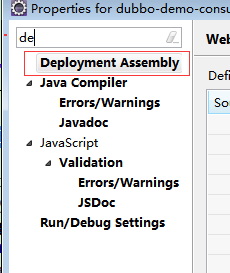
 —— 成功引入 maven依赖如左图

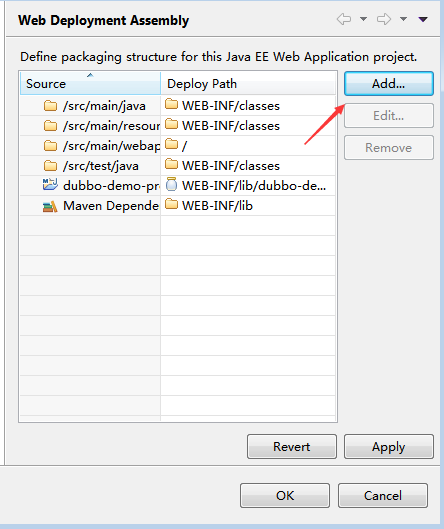
通过 groupId、artifactId和version三个坐标找到我们之前install的jar包，这样我们就能引用之前定义的RemoteOrderService。

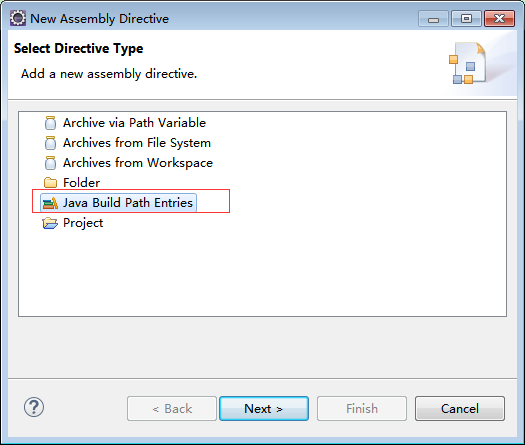
通过某种协议获取到对应的服务，这样我们就不用本地在生成service及对应的bean了，通通都已经被打包到jar。然后通过这些协议我们可以在客户端直接调用服务器的方法，拿到对应的数据。

发布时要注意把相关依赖的jar添加到发布路径中去，具体操作如下，以eclipse ee为例

右键web工程，点击最下面的Properties，快捷栏输入de，点选搜索结果中出现的







选择如上图 按照next操作 即可把maven依赖加入到发布路径。

但是由于maven优先引入本地库，因此你之前所引入的RemoreOrderService，maven会以工程的形式依赖进来，所以成功依赖之后的图标并不是jar的图标，因此我们还需要一步操作，把真正的jar加入到发布路径（好像无需下面的步骤了）

依然是点击add

选择project，把之前打包的工程选中 ok就行了，这样就会出现



这个时候，依赖的工程会以jar的形式发布。

