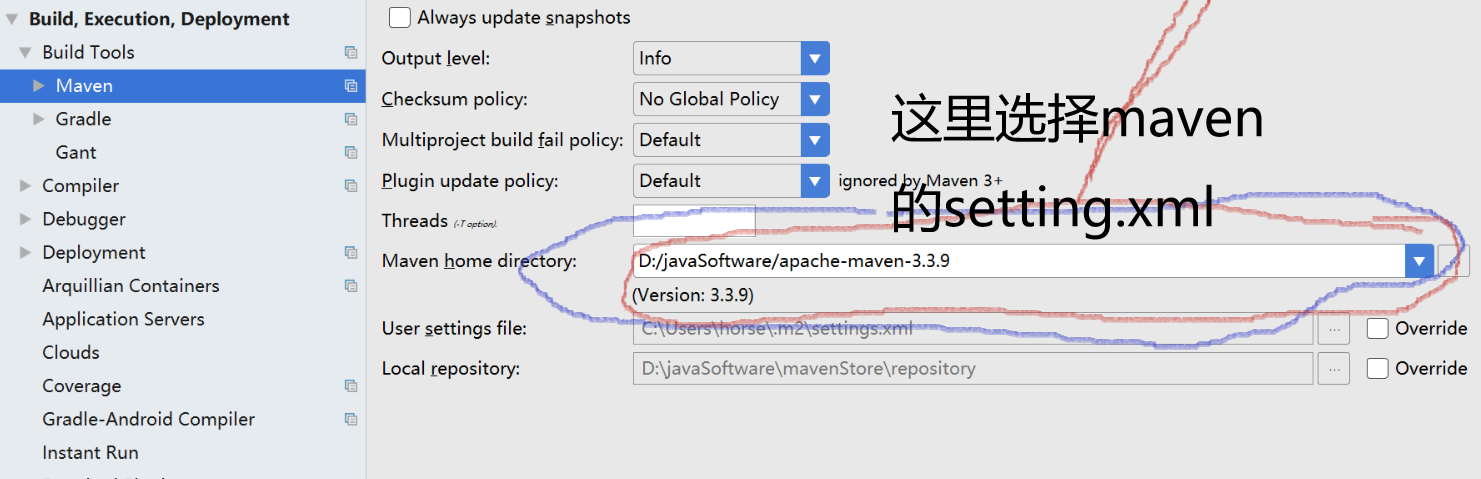
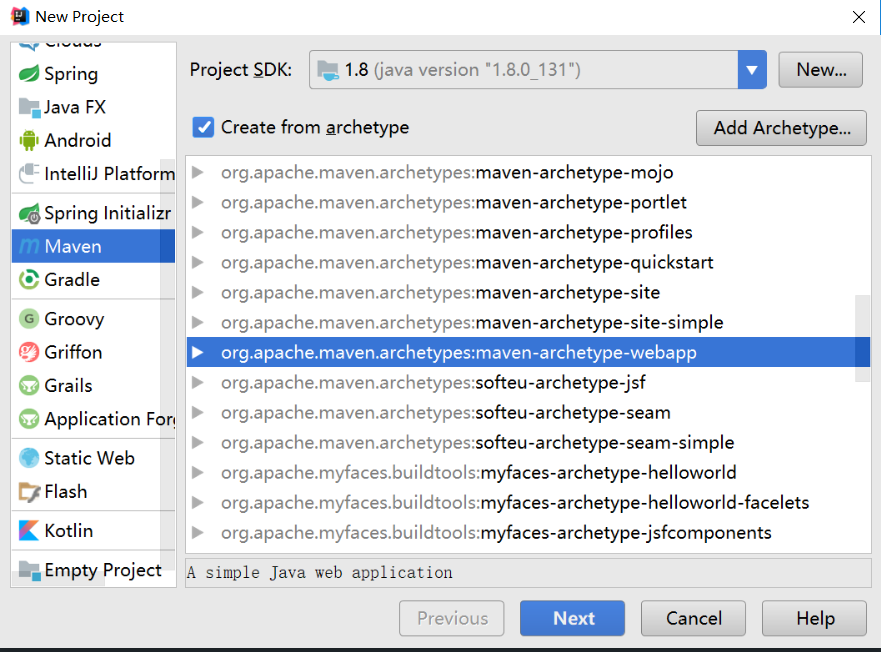
idea中maven的配置：

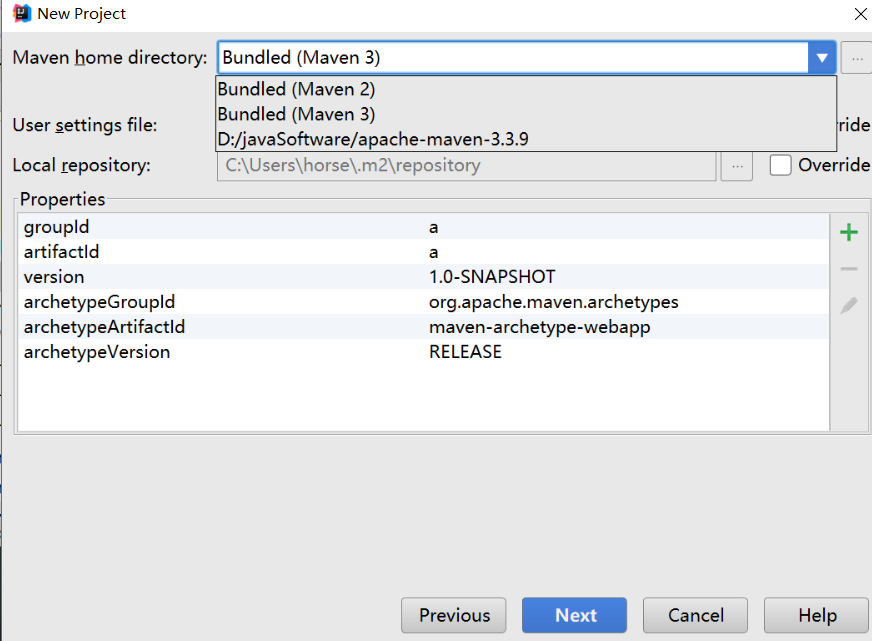
导入maven：



新建maven中web项目：



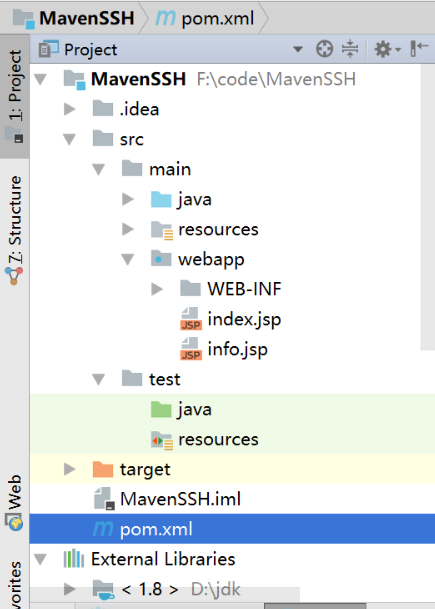
选择webapp骨架



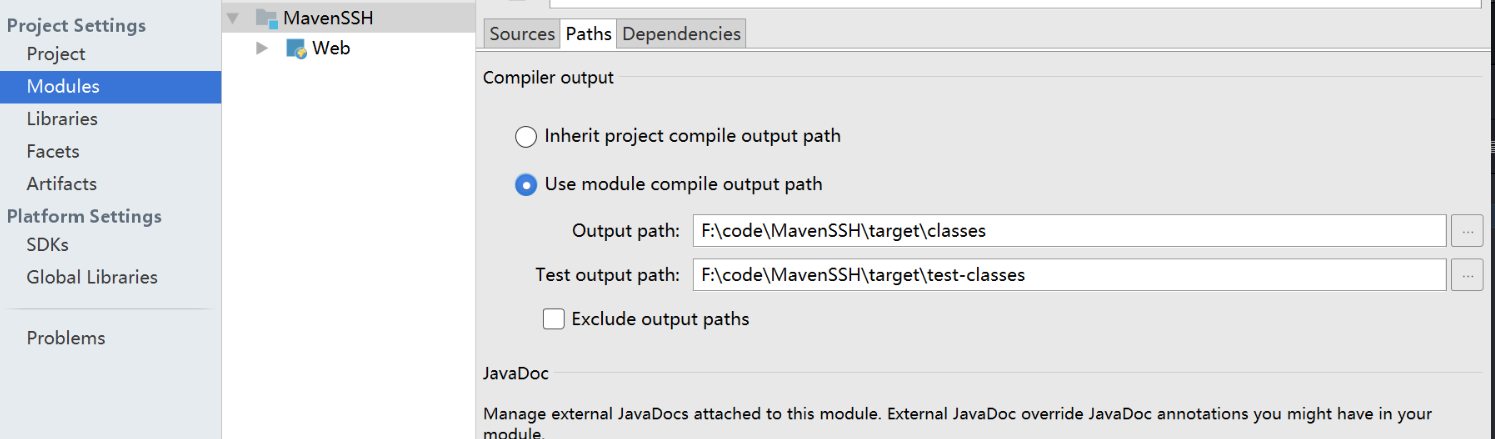
注意：要想生成快，加一个archetypeCatalog internal

选择自己的本机的maven，而不是idea中自带的。

maven的文件夹样式：



src下分为源程序文件夹main和测试文件夹test，又各分为程序文件夹java/test 和 放配置文件的resources文件夹。main文件夹中还有webapp文件夹。



这里可以选择把.java文件编译后的存放位置，maven项目会编译到target文件夹下，如图。

maven中服务器的配置

maven是什么东西？

Maven是跨平台的项目管理工具。主要为了项目构建和依赖管理。

项目构建：

编译—>测试—>运行—>打包—>部署

通过maven都可以完成。

依赖管理：

maven项目中可以不含jar包，通过一个依赖仓库来实现。

maven是怎么进行项目构建的

编码，通过idea来完成。

编译，通过maven的指令 mvn compile 来完成。

测试，通过maven的指令 mvn test 来完成。

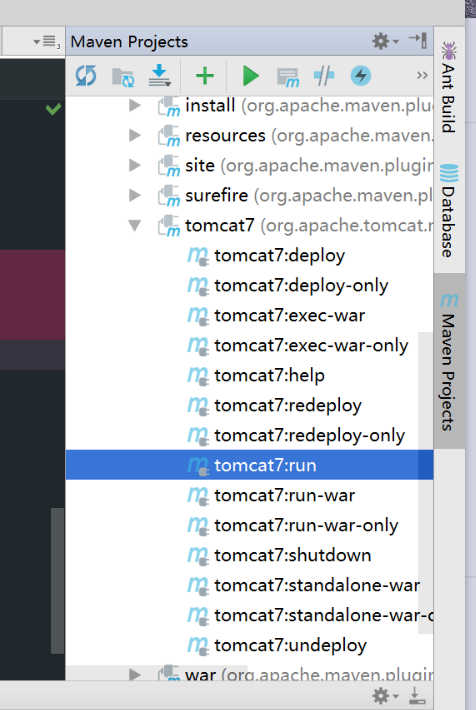
运行？运行一个web项目是需要web服务器的，那么maven是一个服务器吗？答案是否定的，那么maven是怎么运行的呢？

我们之前的非maven项目都是通过 idea 配置我们本地的 tomcat服务器来发布运行的。那现在如果我们没有本地服务器，那怎么运行？

maven中默认带了一个tomcat6的插件，所以可以直接通过tomcat:run来运行。但是tomcat6这个插件有问题，所以现在一般都安装一个tomcat7的插件，在pom.xml中添加配置：

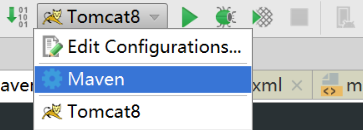
<**plugin**>  
 <**groupId**>org.apache.tomcat.maven</**groupId**>  
 <**artifactId**>tomcat7-maven-plugin</**artifactId**>  
 <**version**>2.2</**version**>  
 <**configuration**>  
 <!--<url>http://localhost:8080/manager/text</url>-->  
 <!--<server>myServer</server>-->  
 <!--<username>zk</username>-->  
 <!--<password>123</password>-->  
 <**path**>/${project.artifactId}</**path**>  
 </**configuration**>  
</**plugin**>

这样就直接得到了一个tomcat7的插件，我们就可以直接通过：

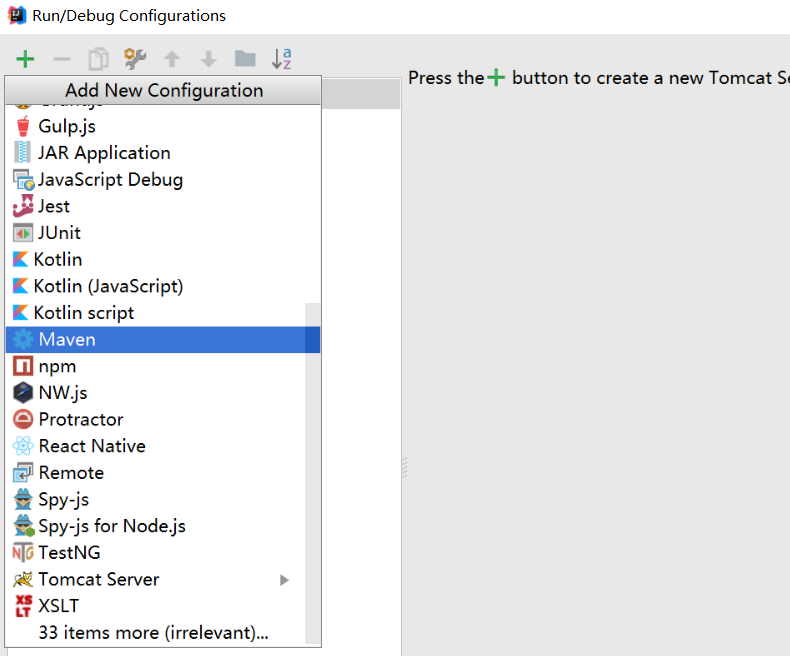


这里是idea的Maven管理模块，我们直接点击tomcat7:run运行就可以了。

我们也可以配置一个Maven，就像配置tomcat模块一样。

 图1

进入配置：

图2

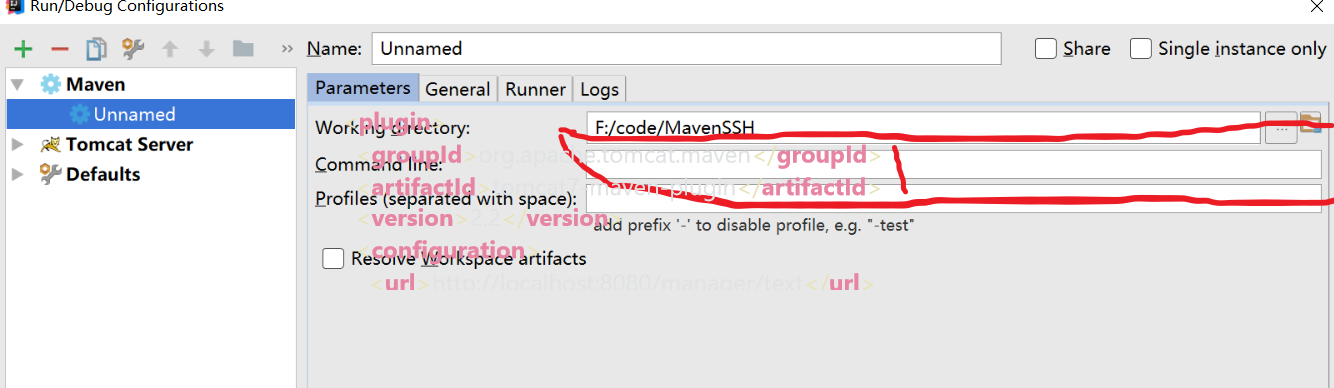


图 3

只要在上图3 画线 处 键入 tomcat7:run 或者其他命令。就可以了。

然后在图1点击那个maven就可以执行tomcat7:run ,

本质上就是配置了一个maven的命令！！

maven部署到本地tomcat服务器

非maven项目，我们通过配置tomcat服务器，对程序进行发布运行，发布到本地的tomcat服务器上。而在maven项目中采用了tomcat的插件（如上面那样配置），可以直接通过tomcat7:run运行，但并不能发布，因为这个插件并没有提供服务器仓库的功能，而上面的配置也没有绑定一个tomcat服务器。

要通过maven把项目部署到本地服务器中，我们必须建立maven插件和tomcat服务器的关系。

首先在本地tomcat服务器下的conf文件夹中的tomcat-users.xml中添加

<**role** rolename="manager" />  
<**role** rolename="manager-gui"/>   
<**role** rolename="manager-script"/>   
<**user** username="zk" password="123" roles="manager-gui, manager-script,manager" />

role标签的作用是规定权限。

user标签的作用是为tomcat创建一个用户。

然后通过点击 <http://localhost:8080/manager/html> 进行验证。

|  |
| --- |
| Tomcat Manager是Tomcat自带的、用于对Tomcat自身以及部署在Tomcat上的应用进行管理的web应用。Tomcat是Java领域使用最广泛的服务器之一，因此Tomcat Manager也成为了使用非常普遍的功能应用。  在默认情况下，Tomcat Manager是处于禁用状态的。准确地说，Tomcat Manager需要以用户角色进行登录并授权才能使用相应的功能，不过Tomcat并没有配置任何默认的用户，因此需要我们进行相应的用户配置之后才能使用Tomcat Manager。 |

然后在本地maven下的conf中的setting.xml中添加：

<**server**>  
 <**id**>myServer</**id**>  
 <**username**>zk</**username**>  
 <**password**>123</**password**>  
</**server**>

因为maven中很多setting.xml，首先idea自带的setting.xml，然后maven本地仓库位置处也有一个setting.xml.

如果不使用idea的自带setting.xml那么里面就不要加这段代码。

使用本地maven，添加代码不是仓库那个setting.xml而是conf文件夹下的。

最后在pom.xml中插件配置中改成：

<**plugin**>  
 <**groupId**>org.apache.tomcat.maven</**groupId**>  
 <**artifactId**>tomcat7-maven-plugin</**artifactId**>  
 <**version**>2.2</**version**>  
 <**configuration**>  
 <**url**>http://localhost:8080/manager/text</**url**>  
 <**server**>myServer</**server**> <!--在setting.xml中配置的server id-->  
 <**path**>/${project.artifactId}</**path**>  
 </**configuration**>  
</**plugin**>

通过server标签绑定tomcat。

然后运行tomcat7:deploy 就可以在webapps文件夹中见到部署的项目了。

maven部署到远程tomcat服务器

原理和部署到本地一样，但有很多细节不同，具体如下：

<http://www.cnblogs.com/xyb930826/p/5725340.html> 博客地址

maven的多模块开发以及继承聚合概念：

实际使用maven开发的项目，都是多模块聚合而成的，是通过继承和聚合来完成的。

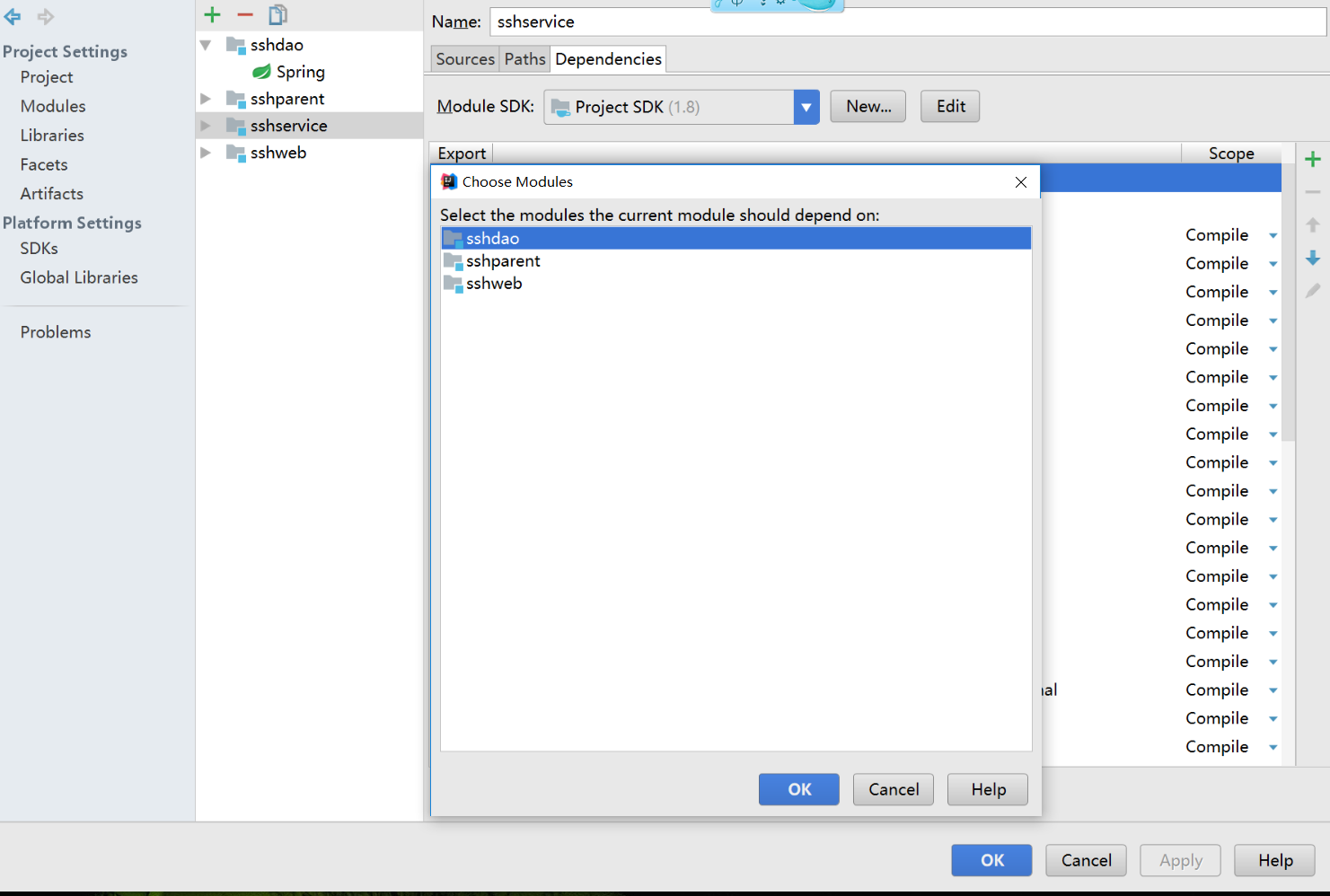
每一个POM对象就代表一个maven项目。一些相同的配置，可以放在父工程中，其他一些子工程，就可以通过依赖传递，拿到父工程。

推荐博客：

<http://juvenshun.iteye.com/blog/305865>

另外，如果模块项目没有发布，那么需要依赖关系的话，有两种方法：

第一种idea的特性：



在这里添加模块的依赖。

第二种：

在pom.xml中添加

<**dependency**>  
 <**groupId**>ncu</**groupId**>  
 <**artifactId**>ssh-dao</**artifactId**>  
 <**version**>1.0-SNAPSHOT</**version**>  
</**dependency**>

另：idea还有私服的概念。就不记载了。