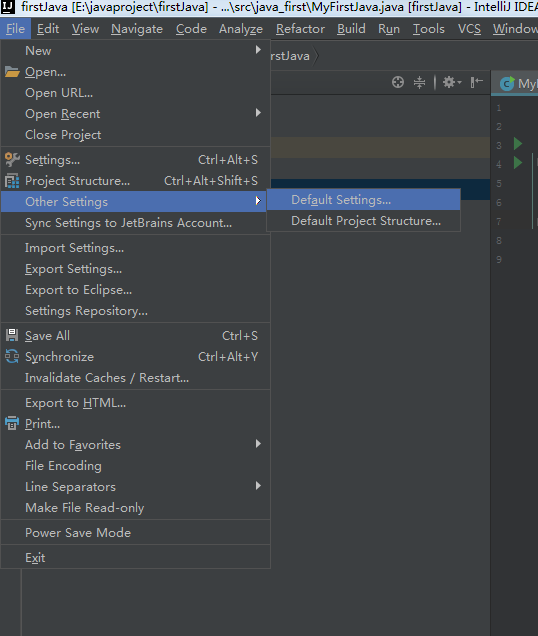
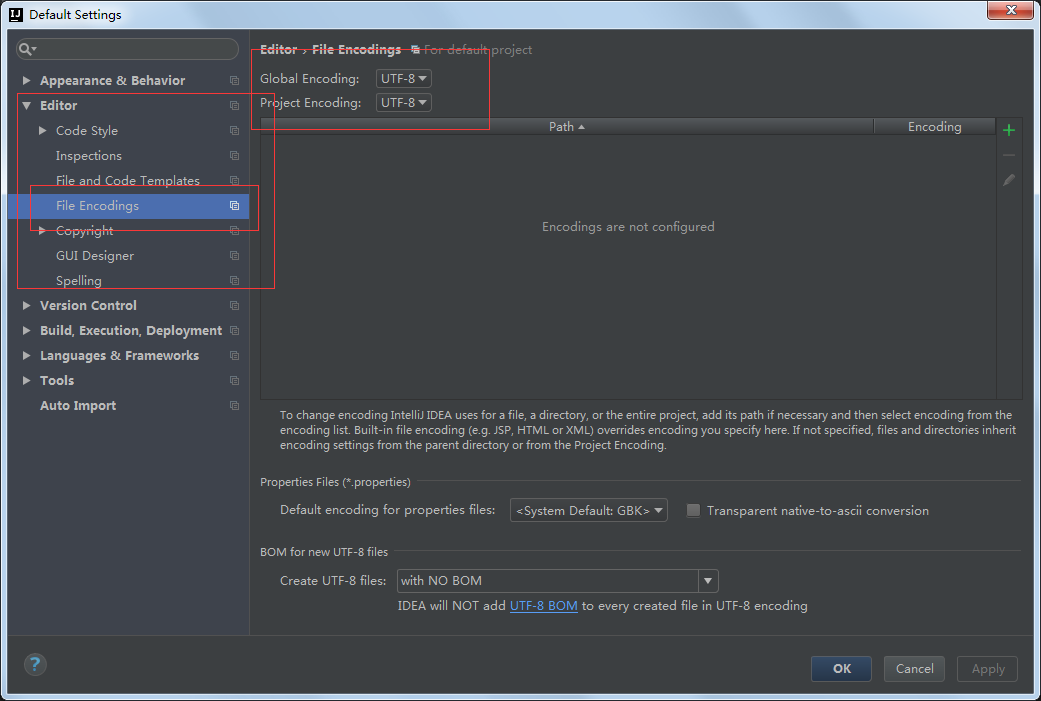
## 字符编码设置

1. 全局编码设置

File -> Other Settings -> Default Settings

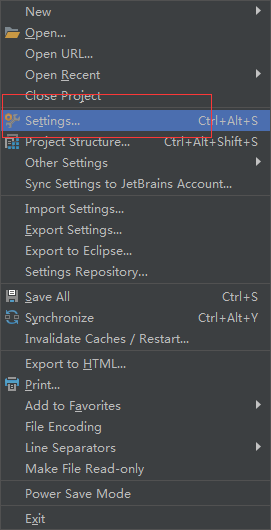


Editor -> File Encodings

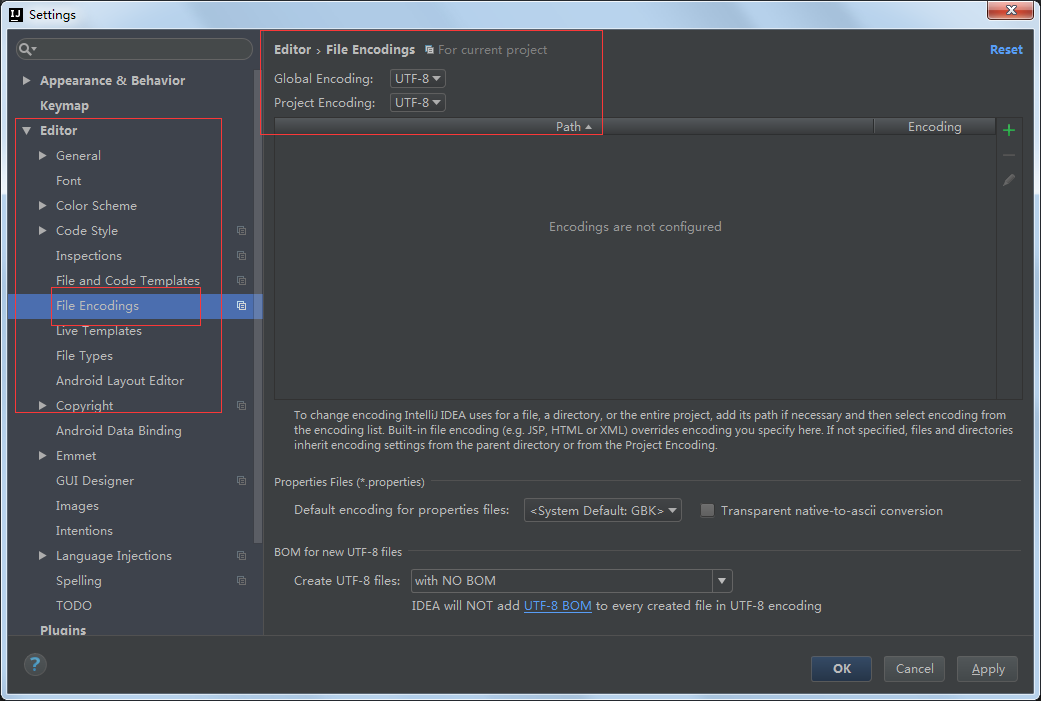


1. 工程编码设置

File -> Settings

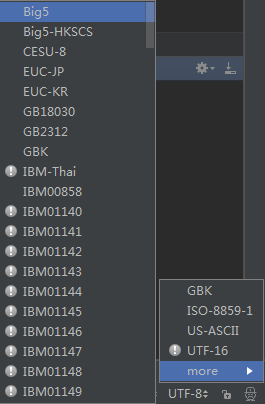


Editor -> File Encodings



1. 文件编码设置

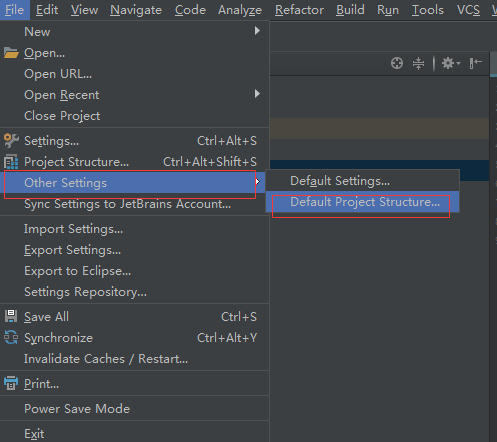
打开需要设置编码的文件，在右下角进行设置



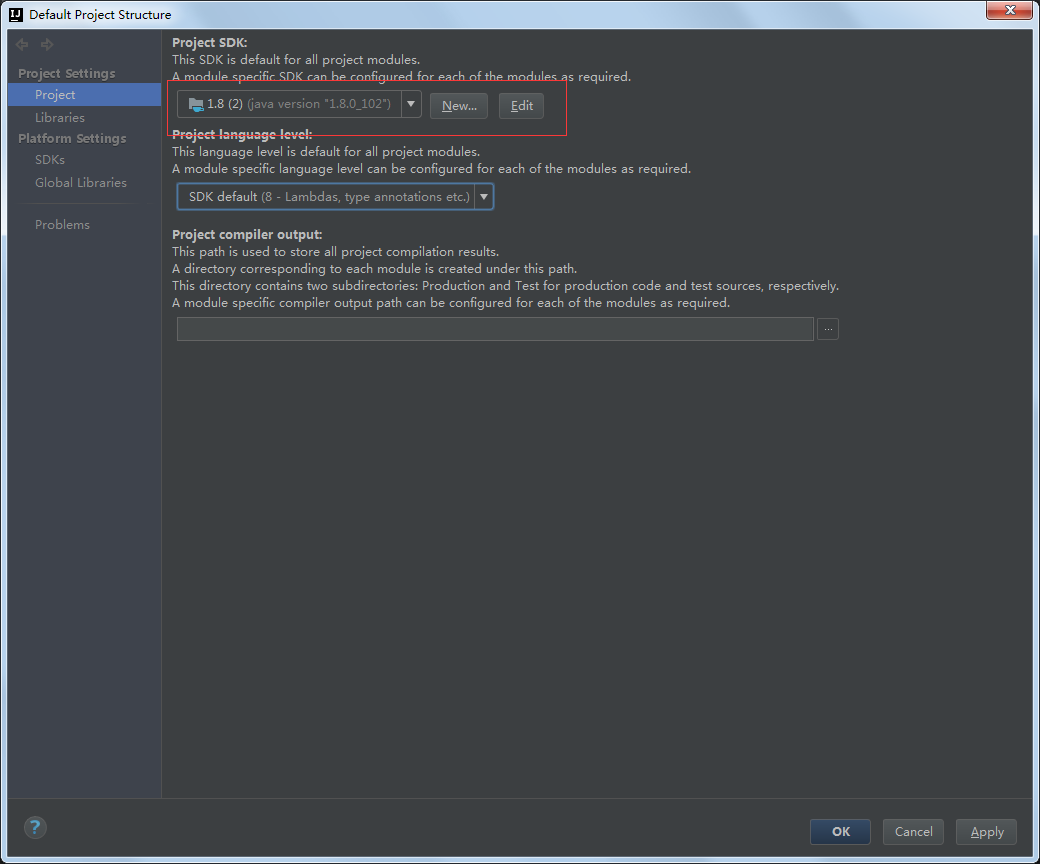
在more中找到utf-8编码

## Jdk配置

1. File -> Other Settings -> Default Project Structure

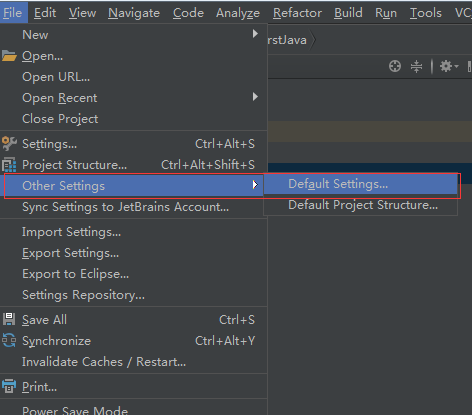


1. Project Settings -> Project

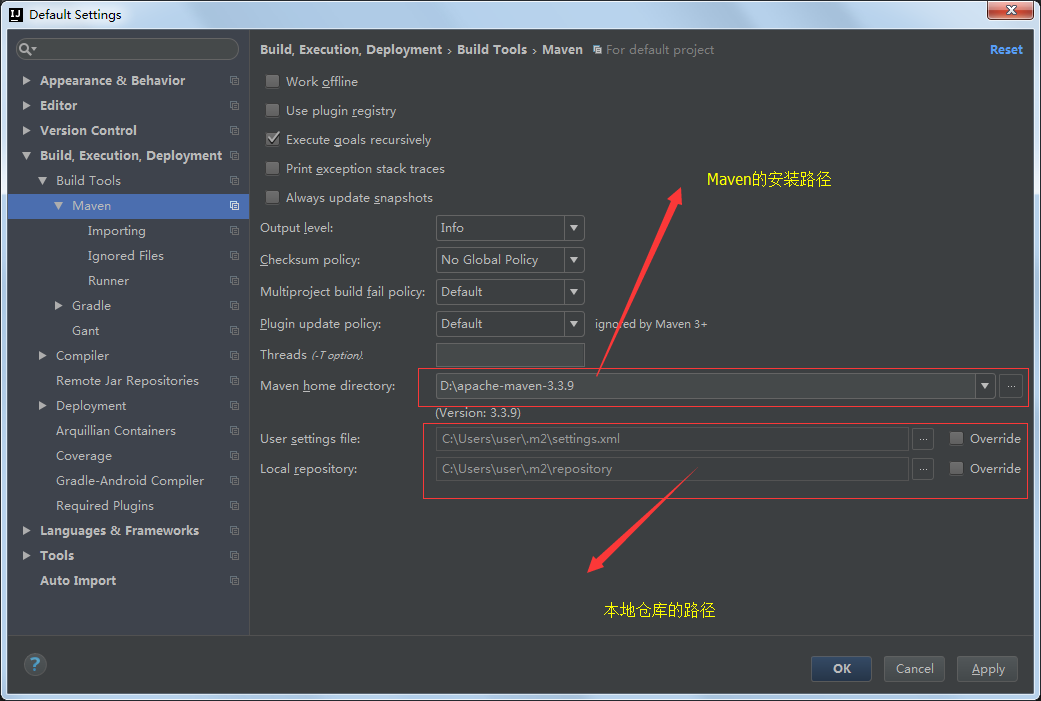


## Maven配置

1. File -> Other Settings -> Default Settings

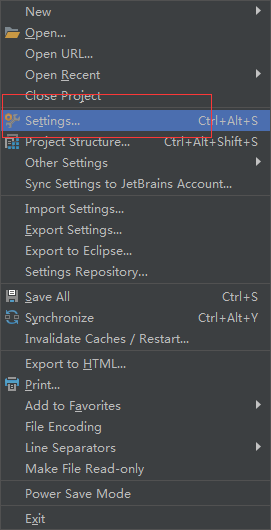


1. Build, Execution, Deployment ->Build Tools -> Maven

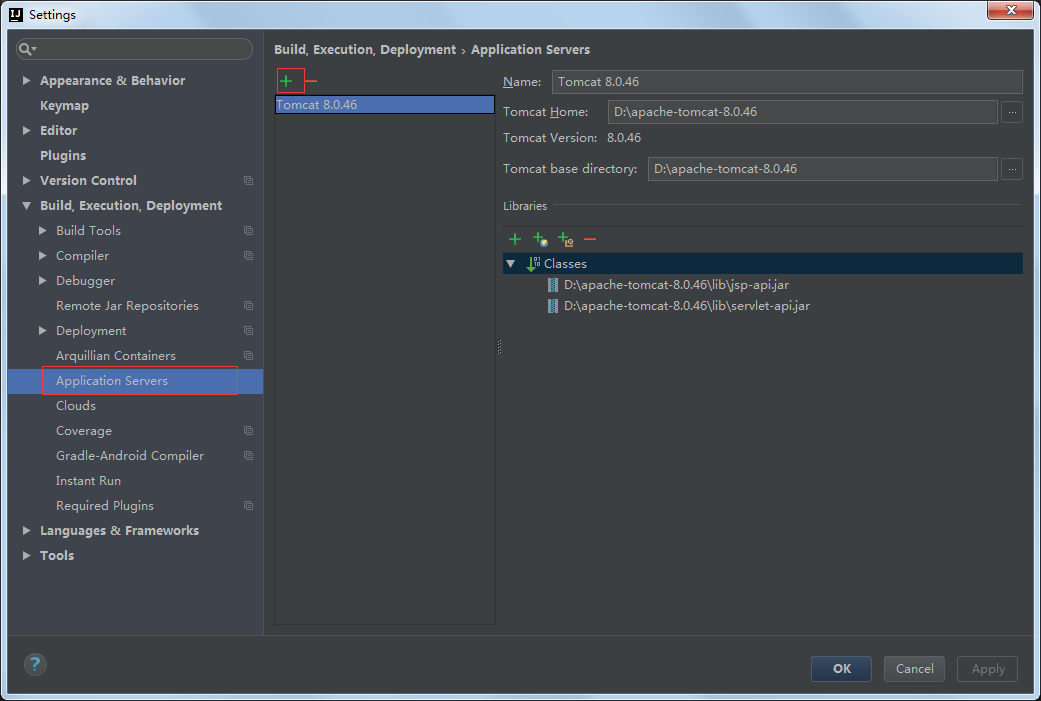


## Tomcat配置

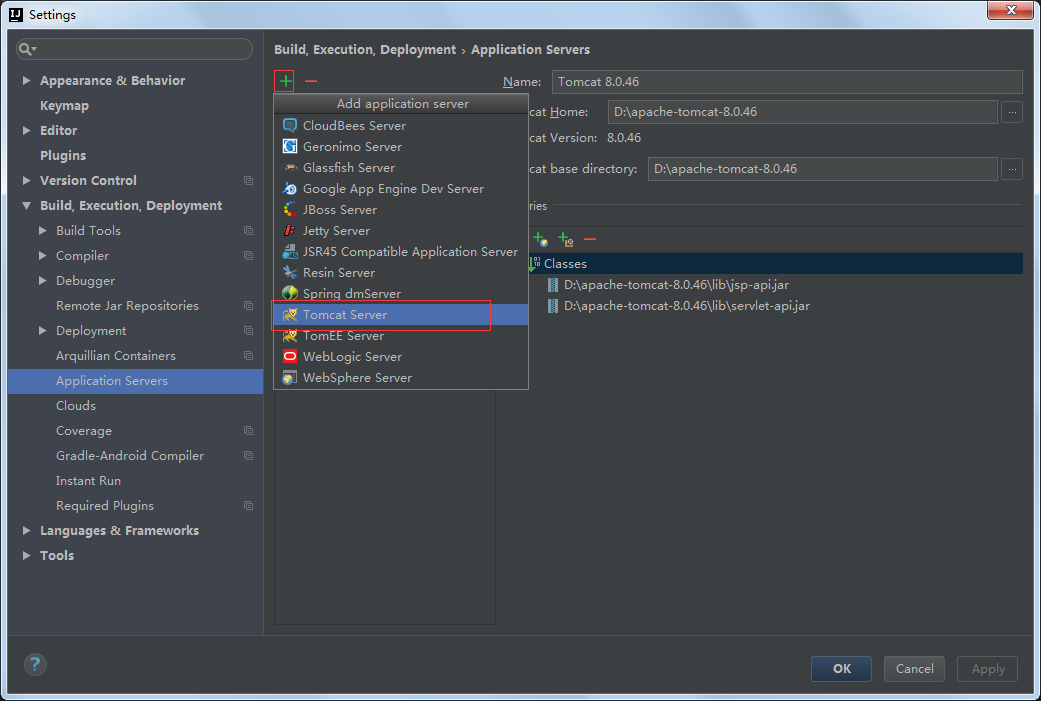
1）File -> Settings



1. Build, Execution, Deployment -> Application Servers



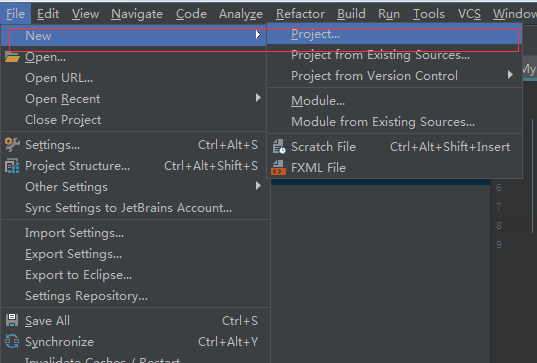
1. 之后点击加号



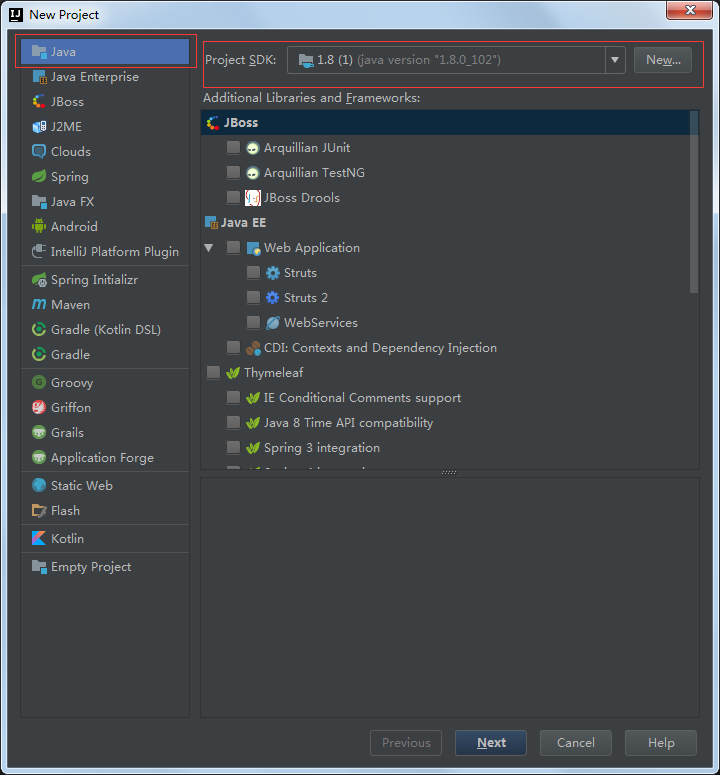
1. 选择tomcat的本地安装包的路径，点击ok就行了

## 创建一个简单的java项目

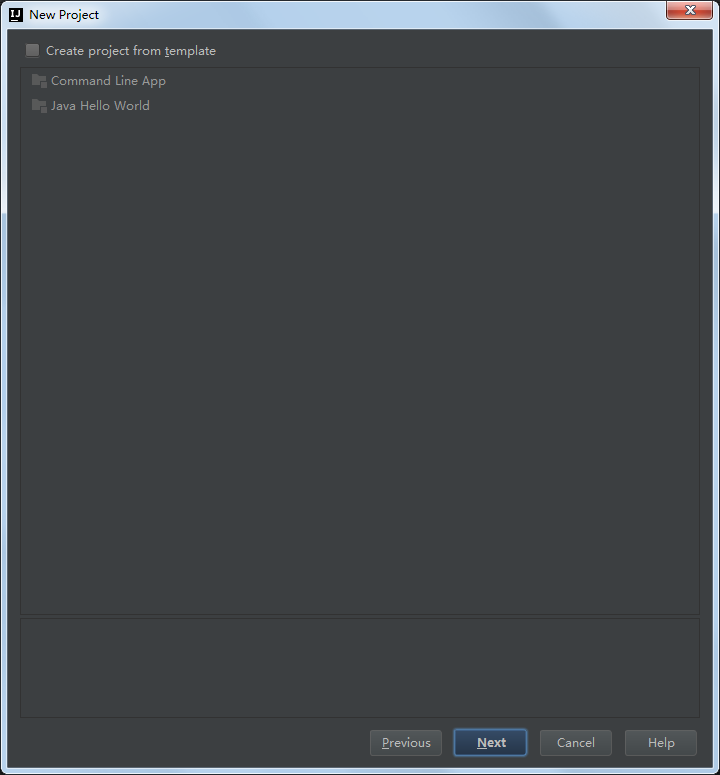
1）File -> New -> Project



1. Java

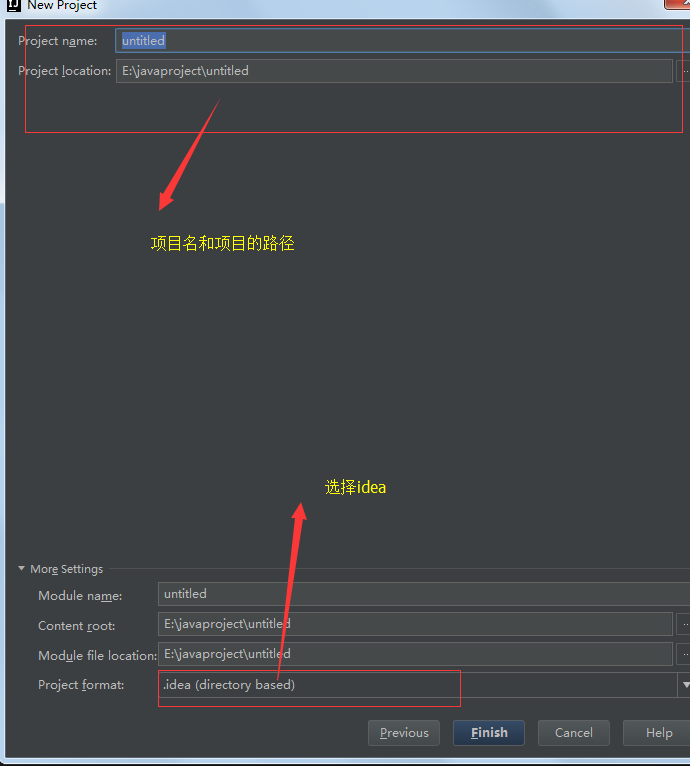


1. 点击下一步，进入下一个窗口

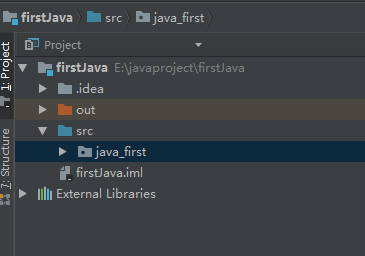


这是一个模板，我们不做操作，直接下一步

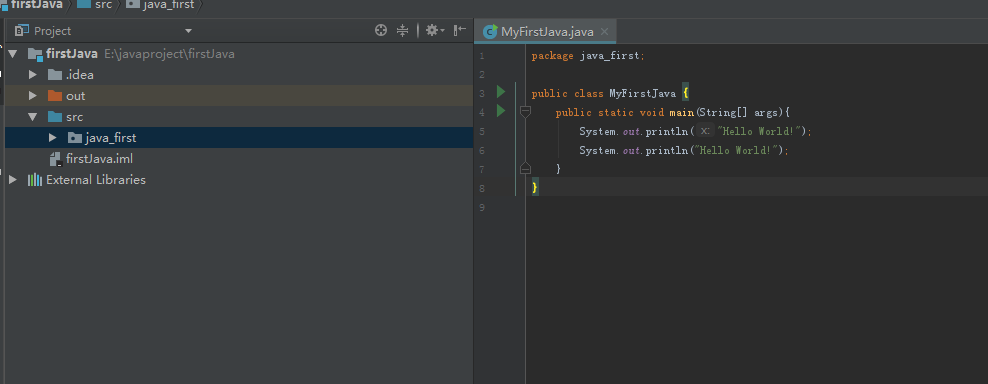
1. Next



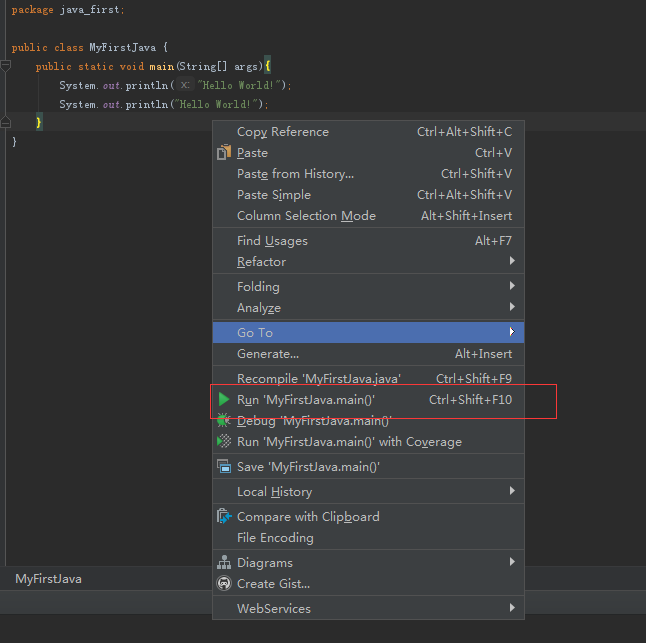
1. 看一下项目的包



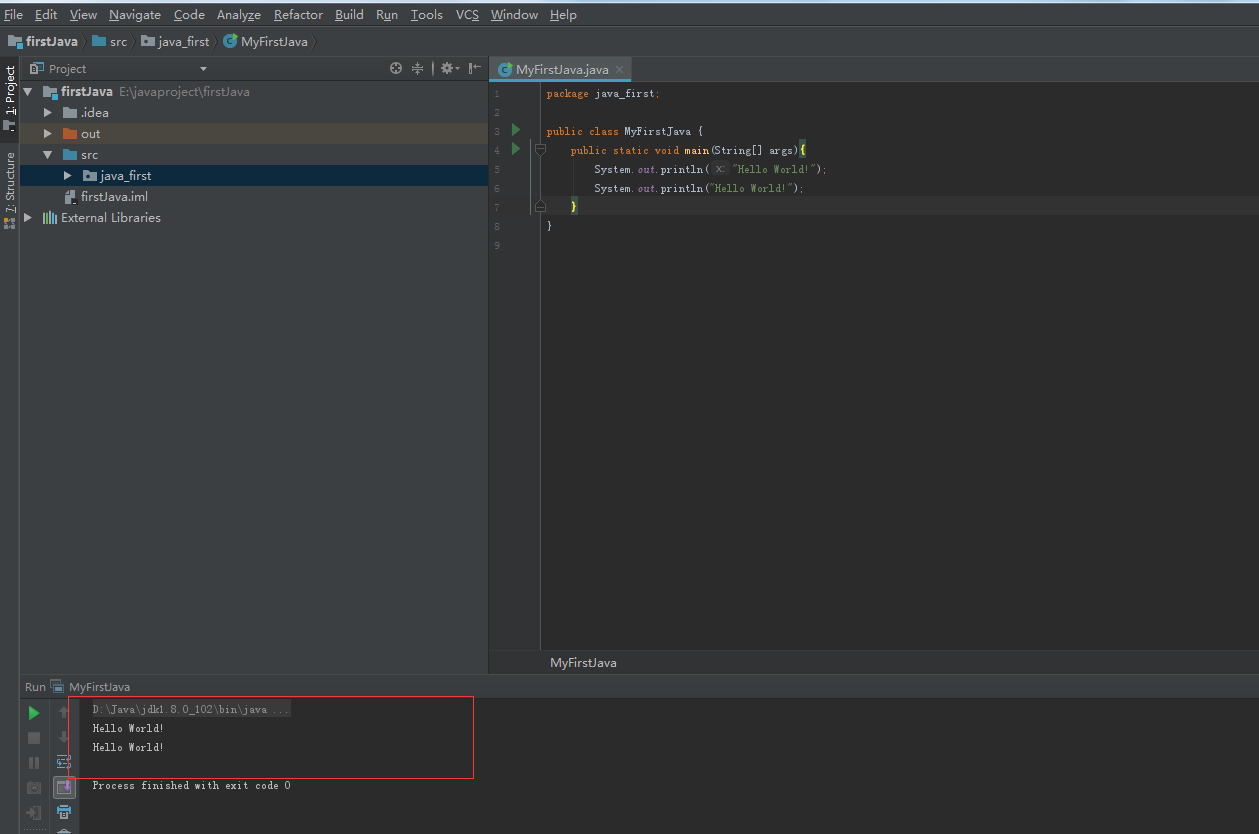
这里面out是run以后自动生成的文件，我们在src下创建一个包，并创建一个class，如下：



我们在class中写一个main函数，输出一句话，右键点击



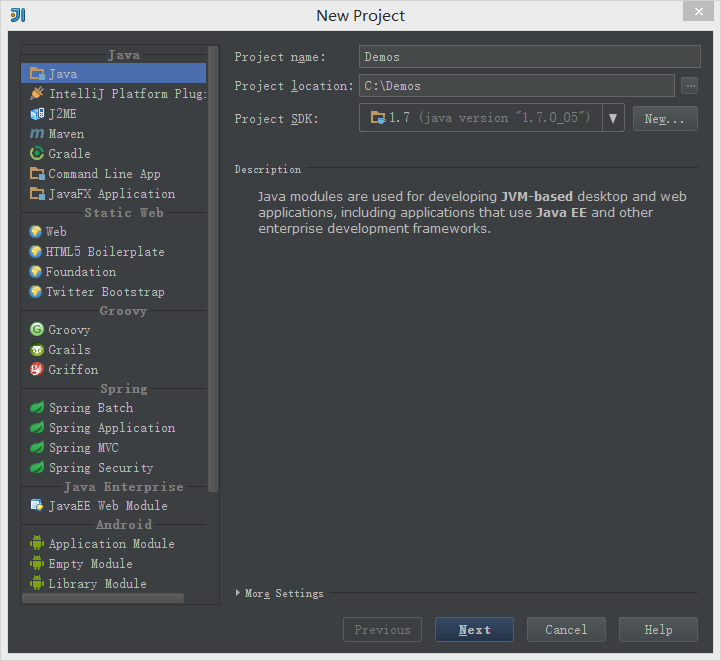
运行main函数，看到控制台输出hello world说明一个简单的java项目已经创建成功了



## 创建一个javaweb项目

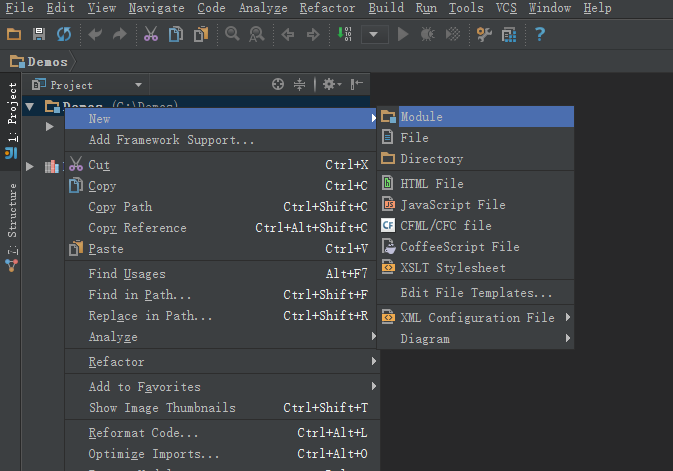
1、创建一个项目

File -> New Project -> 输入项目名称“Demos” ，选择Project SDK为1.7 -> Next -> Finish

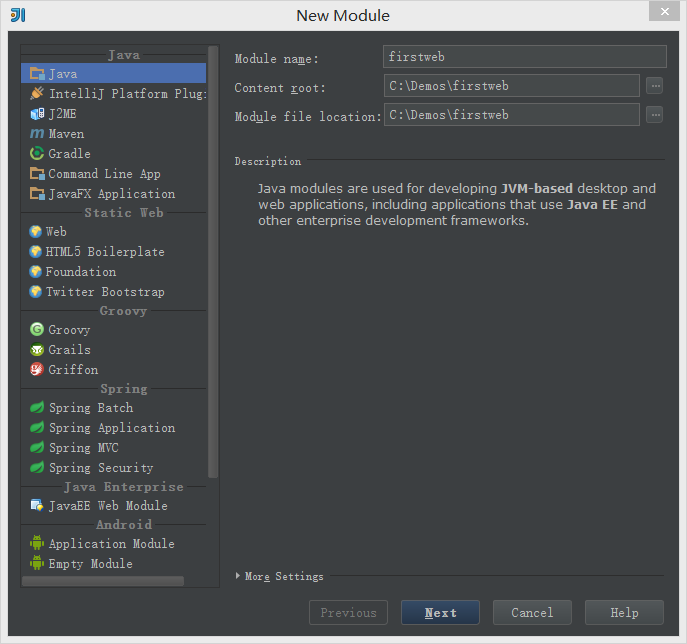


2、创建一个Module

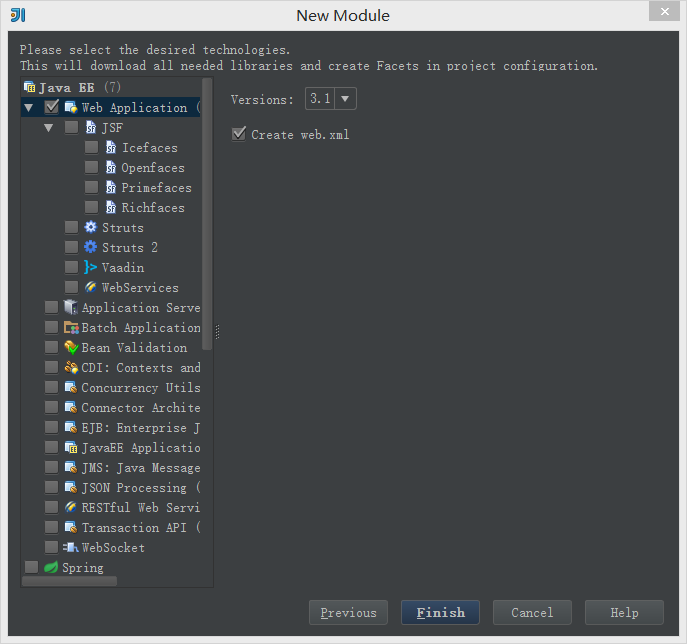
鼠标点中项目名称demos -> 右键选择New，选择Module



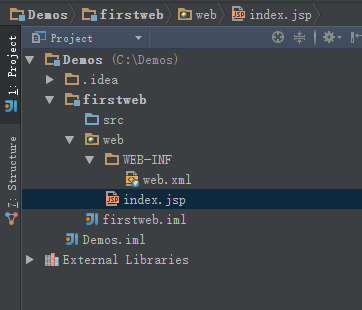
输入Module名称“firstweb” -> 点击Next



勾选“Web Application” -> 确认已勾选“Create web.xml” -> Finish

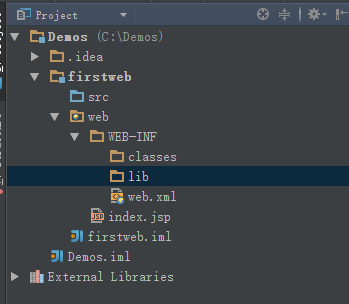


3、创建好的web项目



4、在web/WEB-INF下创建两个文件夹：classes和lib

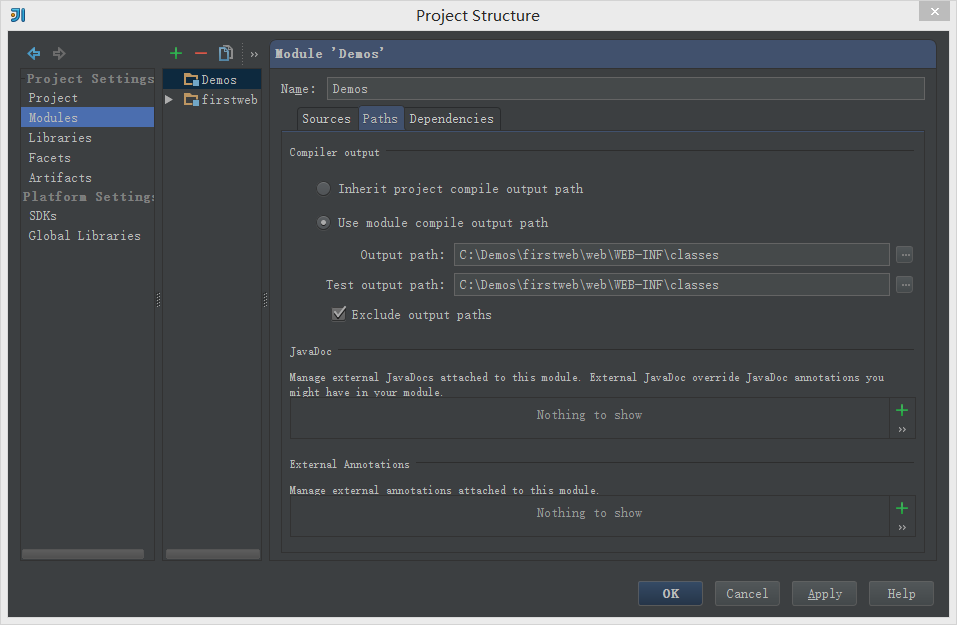
classes用来存放编译后输出的class文件，lib用于存放第三方jar包。



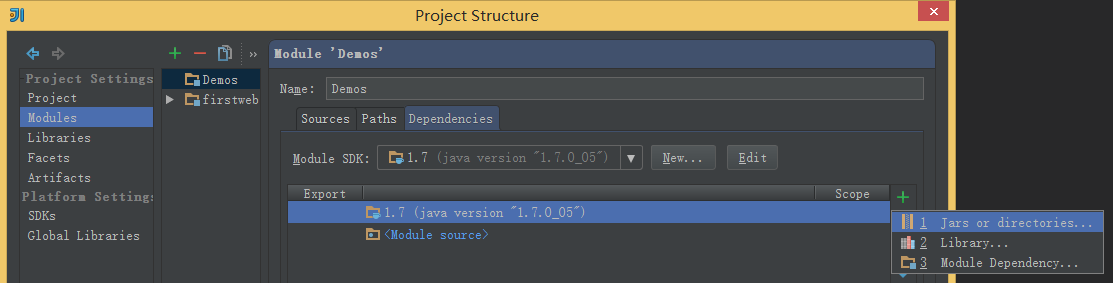
5、配置文件夹路径

File -> Project Structure (快捷键：Ctrl + Shift + Alt + S) -> 选择Module ：

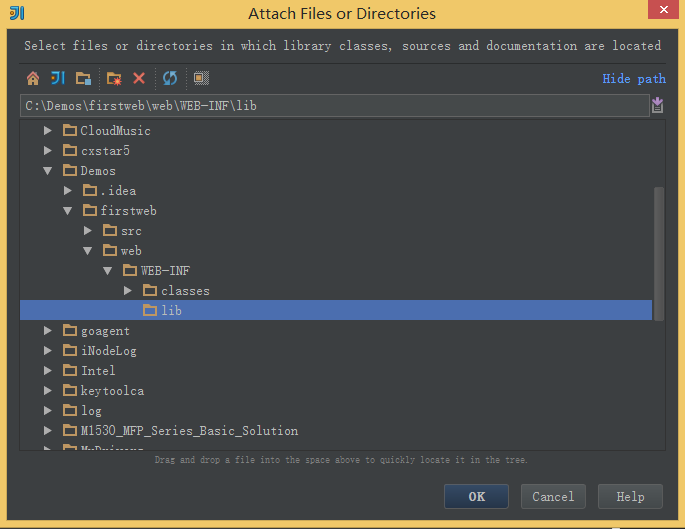
 选择 Paths -> 选择"Use module compile output path" -> 将Output path和Test output path都选择刚刚创建的classes文件夹。



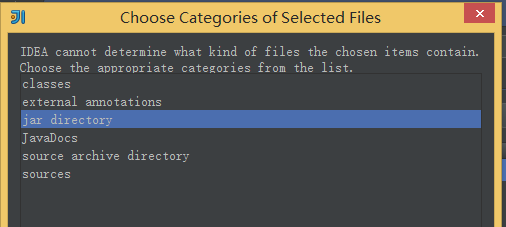
接着选择Dependencies ->  将Module SDK选择为1.7 -> 点击右边的“+”号  -> 选择1 “Jars or Directories”



 -> 选择刚刚创建的lib文件夹

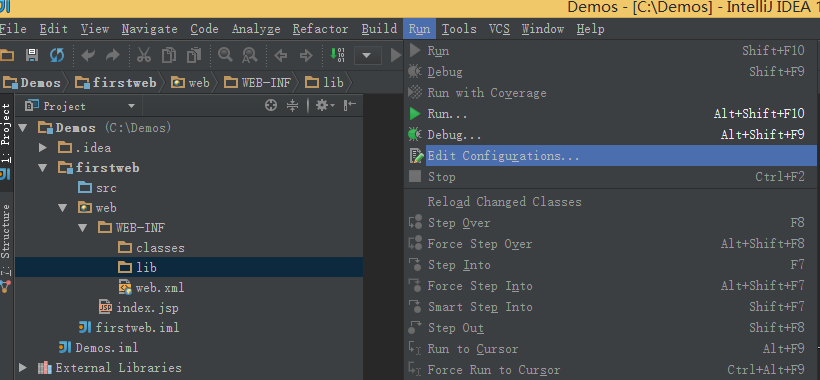


-> 选择“jar directory” -> 接着返回一路OK就行了~~

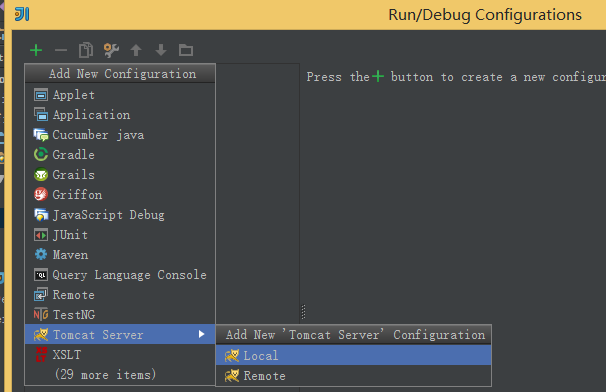


6、配置Tomcat容器

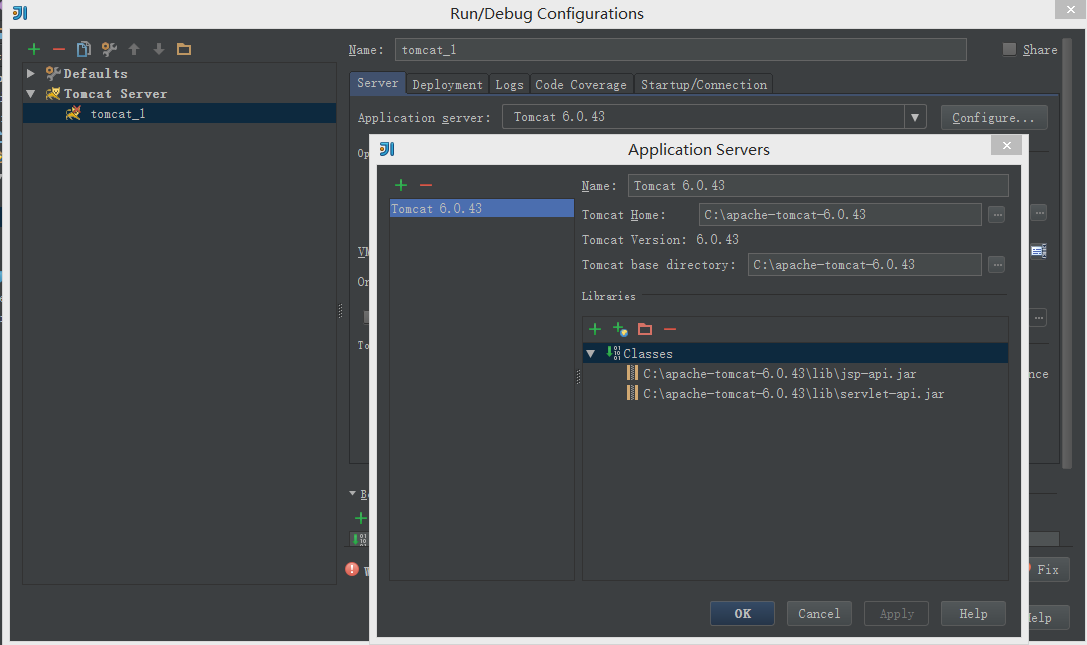
打开菜单Run -> 选择Edit Configuration



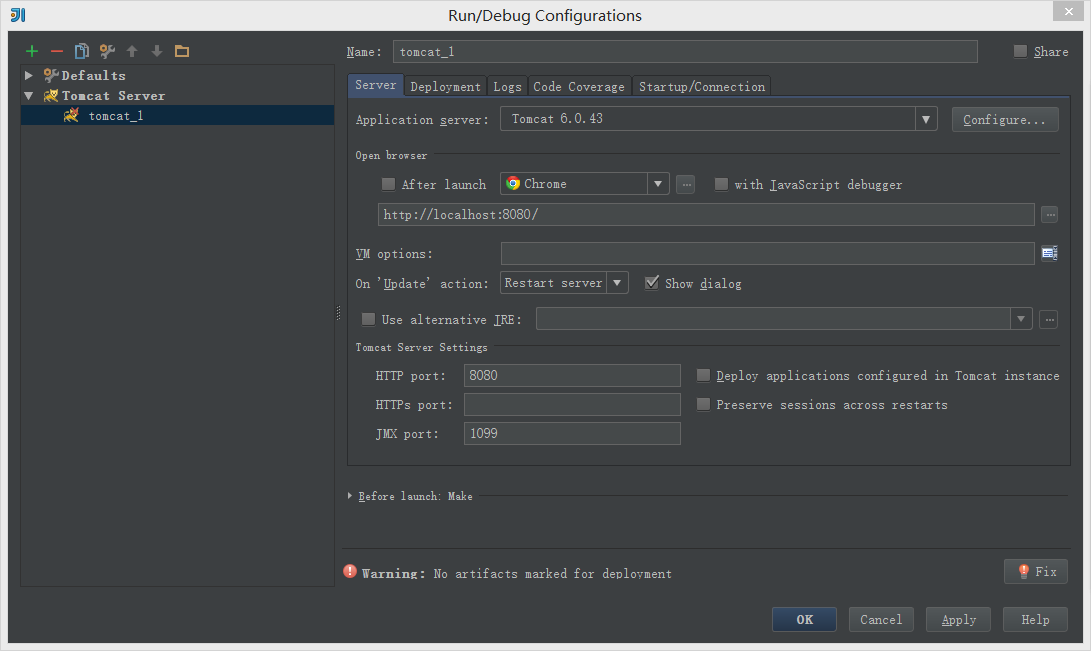
点击“+”号 -> 选择“Tomcat Server” -> 选择“Local”



在"Name"处输入新的服务名，点击"Application server"后面的"Configure..."，弹出Tomcat Server窗口，选择本地安装的Tomcat目录 -> OK

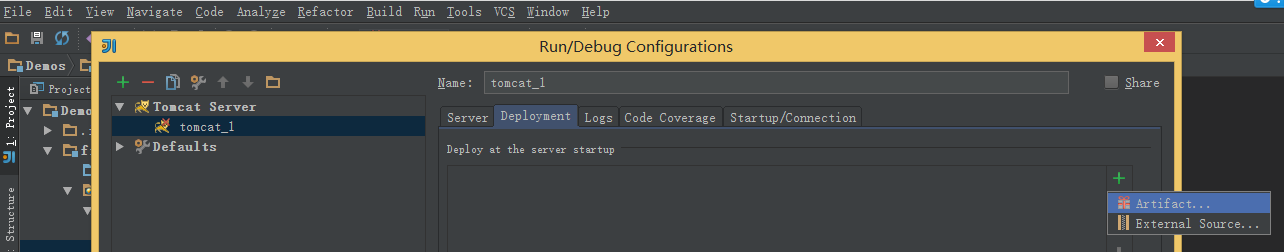


在"Run/Debug Configurations"窗口的"Server"选项板中，取消勾选"After launch"，设置"HTTP port"和"JMX port"（默认值即可），点击 Apply -> OK， 至此Tomcat配置完成。

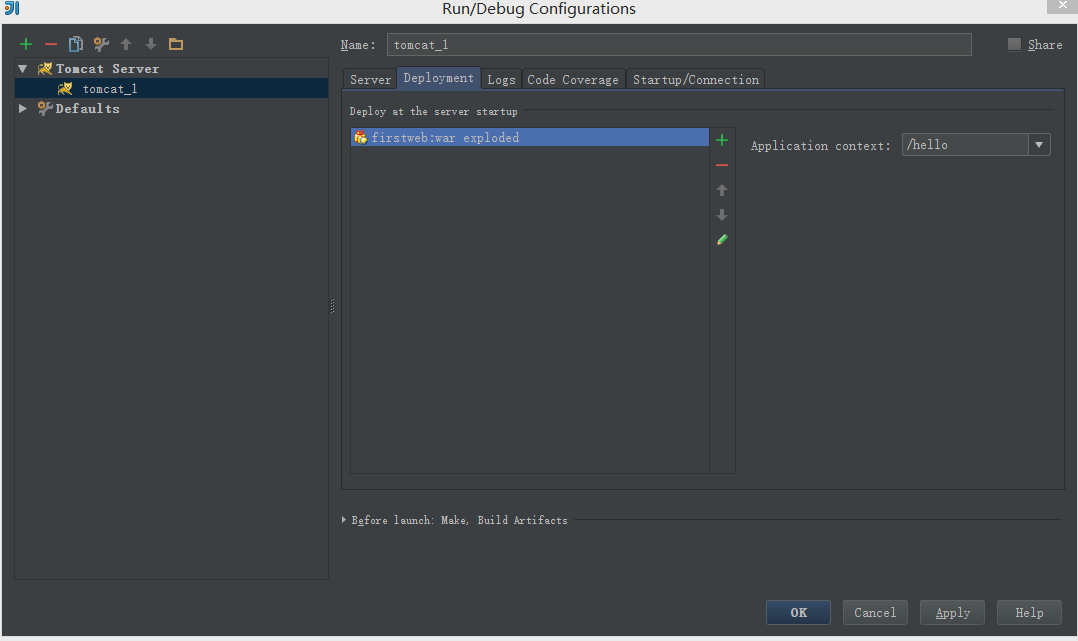


7、在Tomcat中部署并运行项目

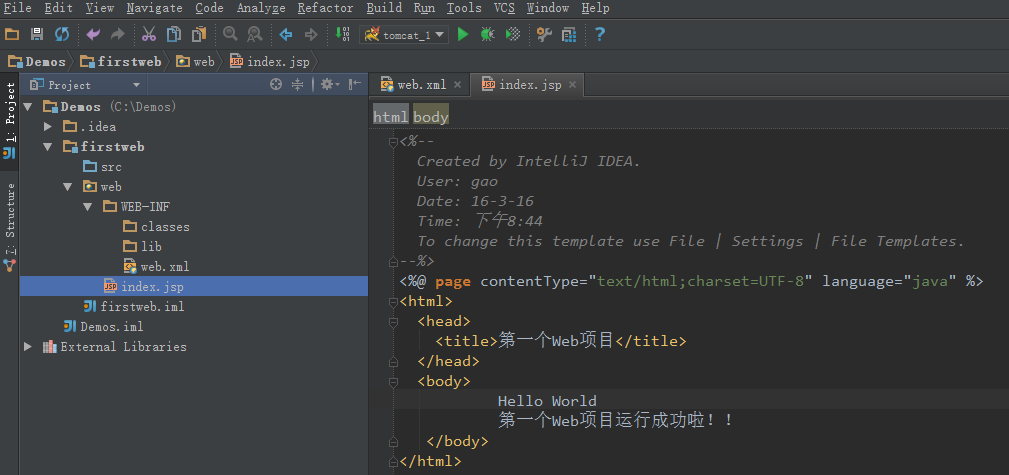
Run -> Edit Configurations，进入"Run/Debug Configurations"窗口 -> 选择刚刚建立的Tomcat容器 -> 选择Deployment -> 点击右边的“+”号 -> 选择Artifact



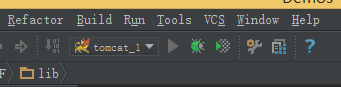
->选择web项目 -> Application context可以填“/hello”(其实也可以不填的~~) -> OK



8、编辑index.jsp文件

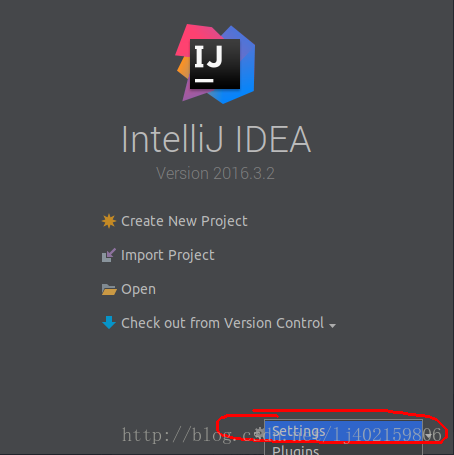


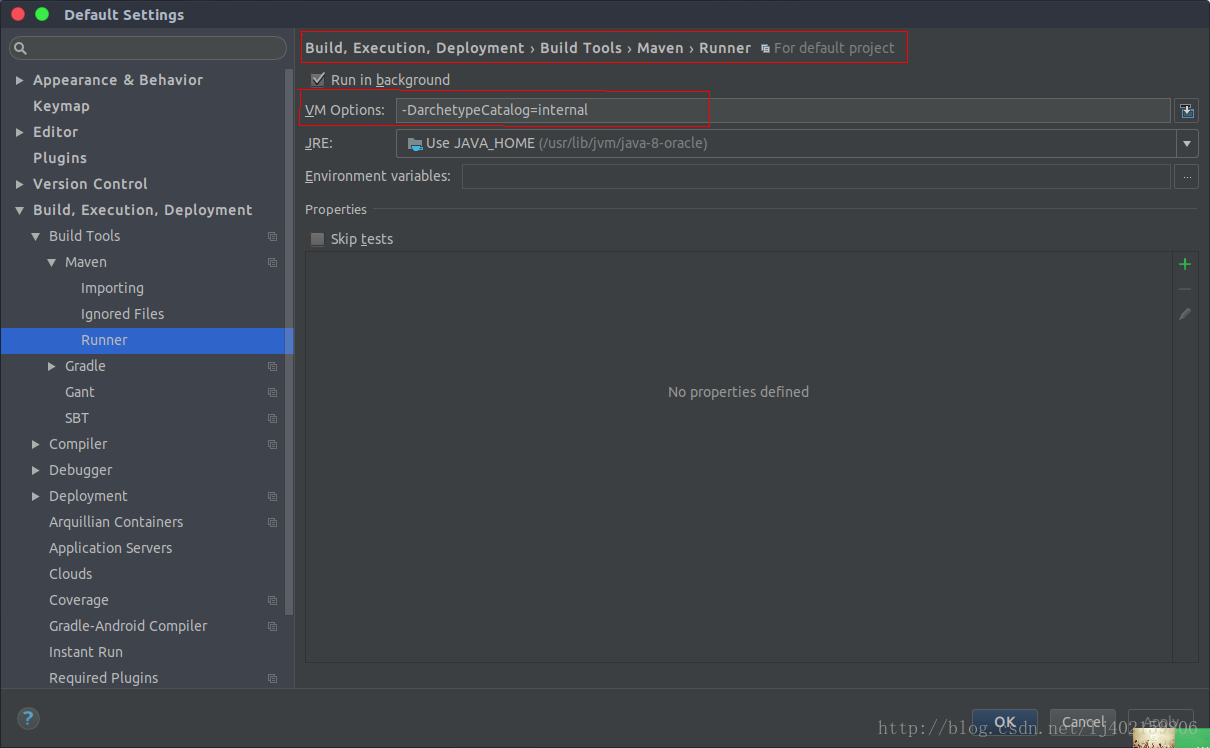
9、运行Tomcat,在浏览器中查看运行结果

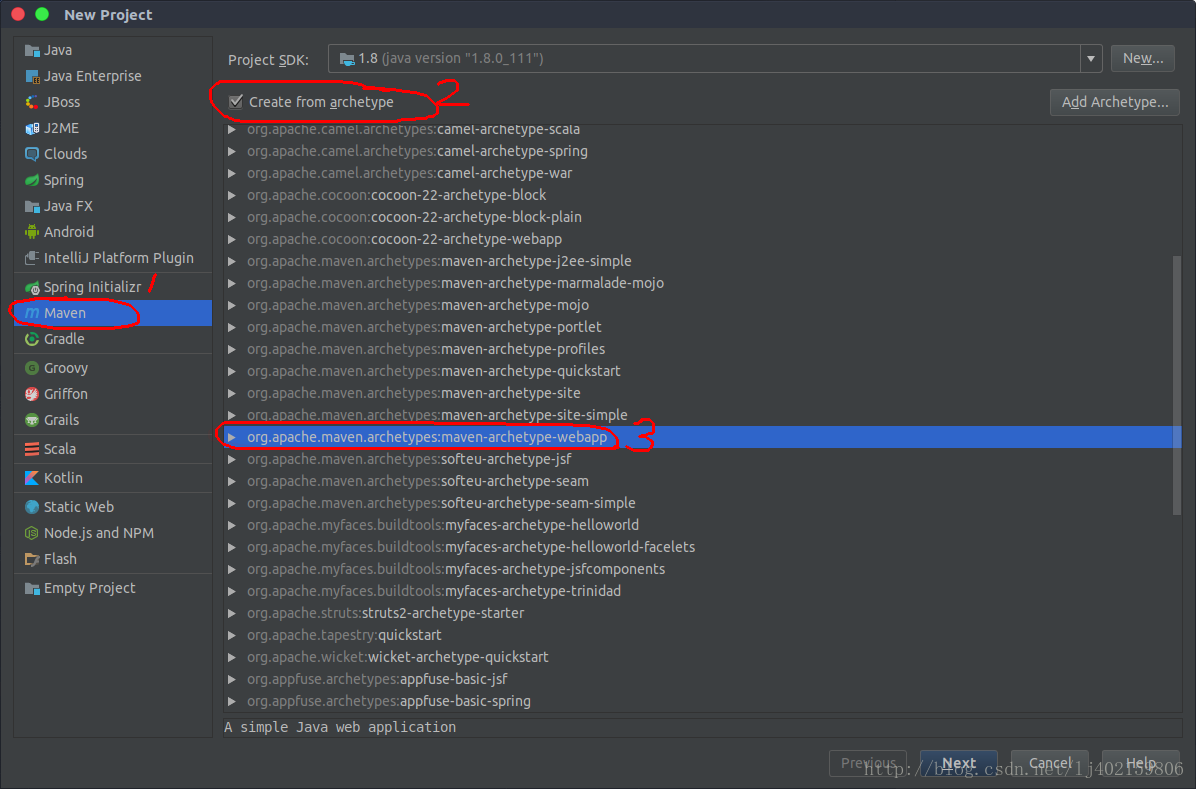


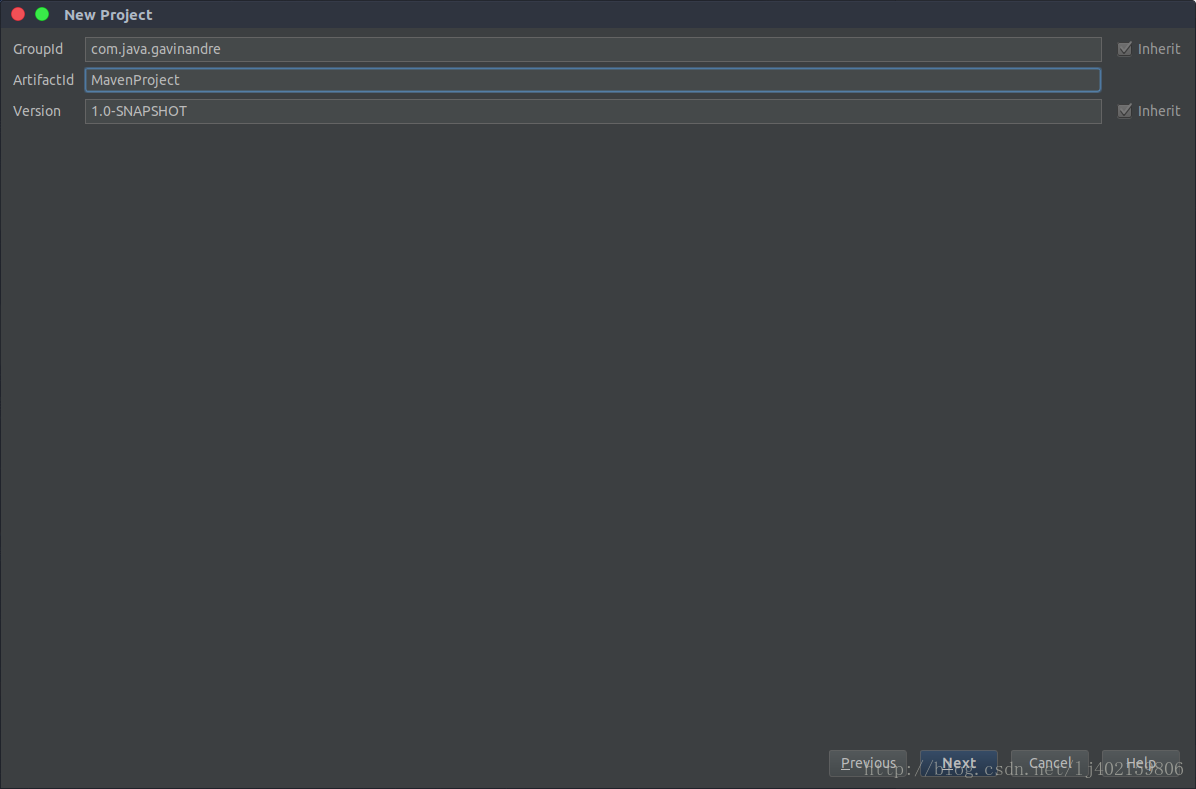


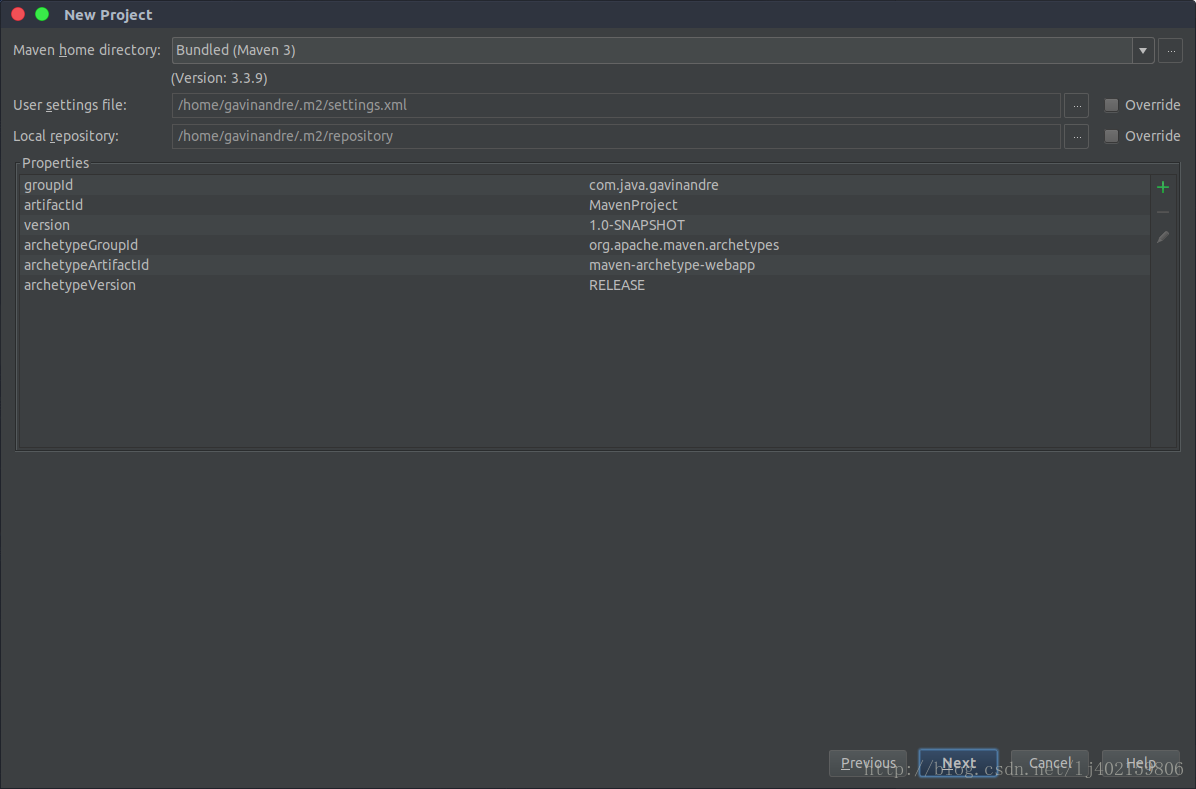
## 创建一个maven项目

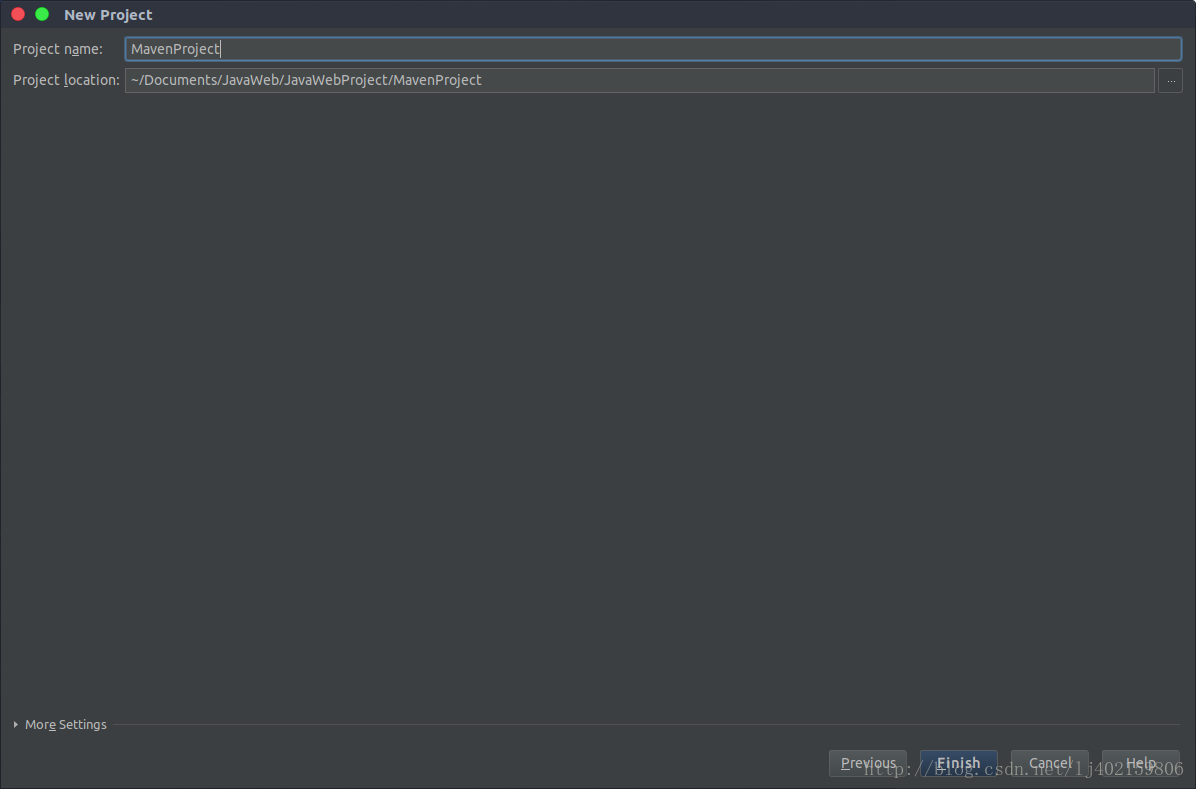
首先打开IDEA后点击settings   


然后在VM Options内添加-DarchetypeCatalog=internal 运行参数   
  
****(不做上面操作的话会导致maven 骨架生成速度缓慢到令人发指,卡在running tmp archetypexxxtmp这一步)****

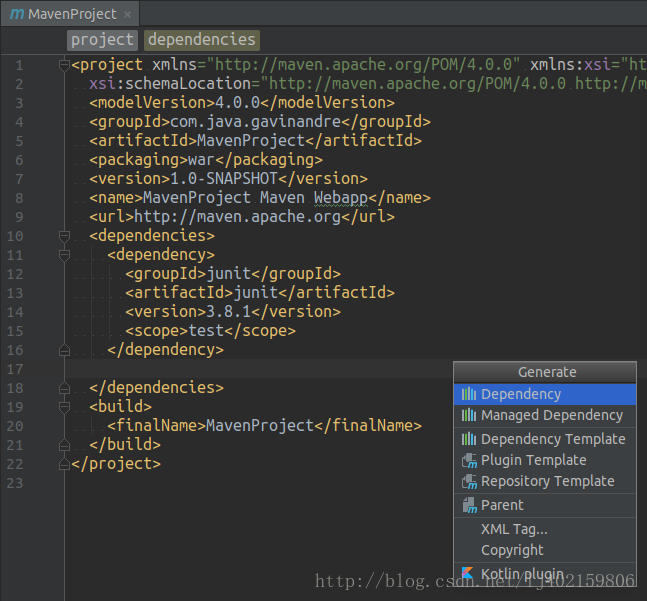
创建maven web项目,按照下图顺序点击后点击下一步   


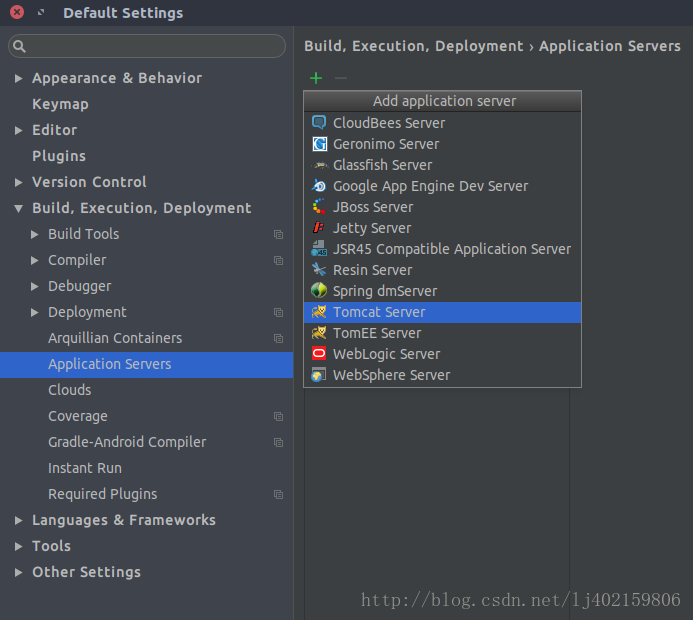
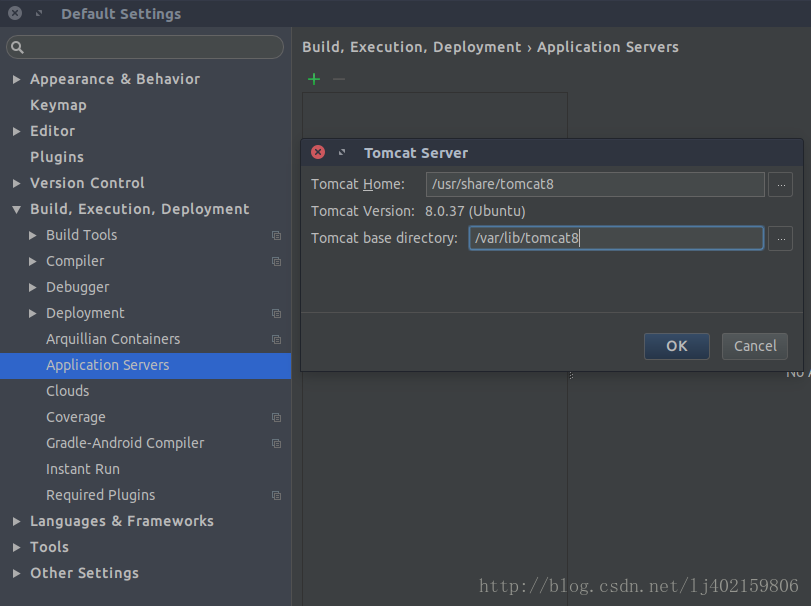
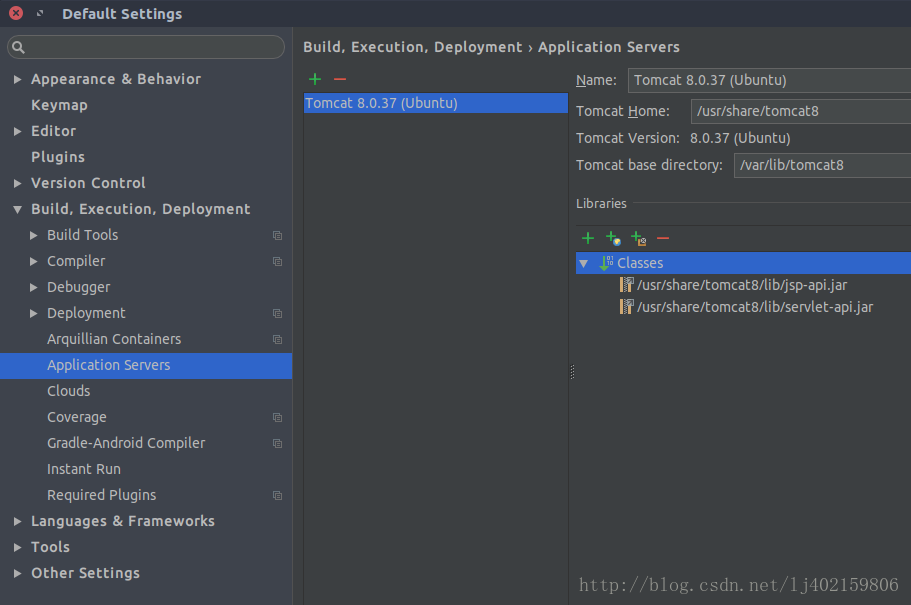
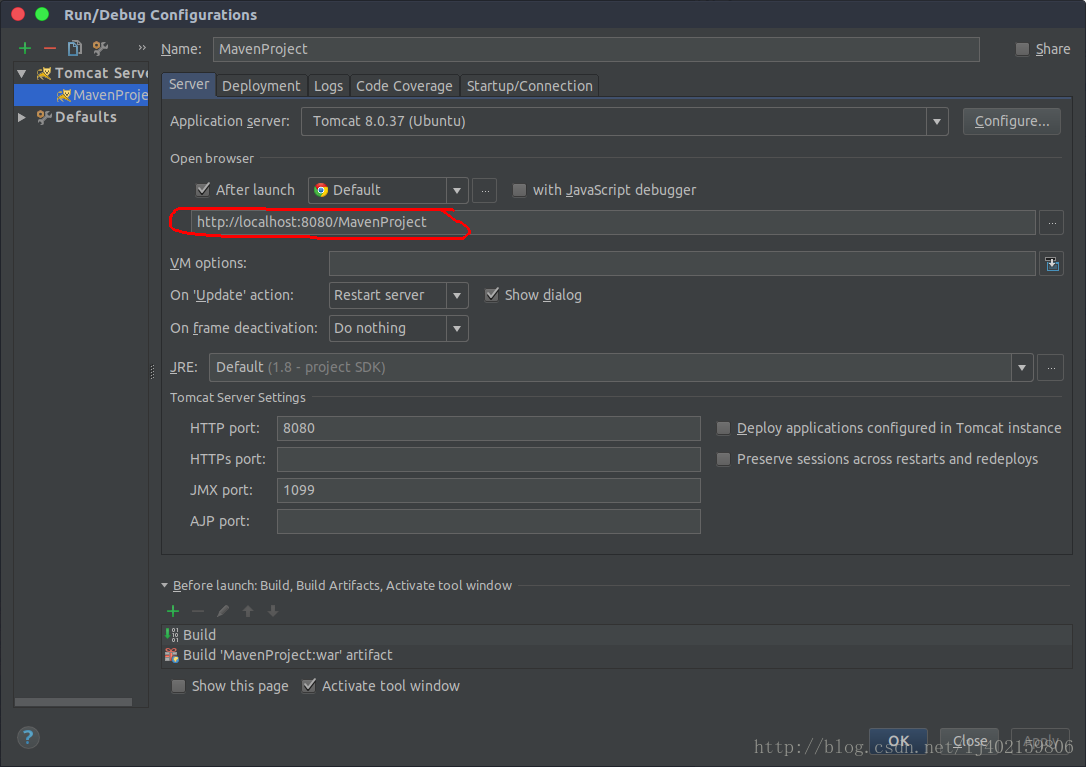
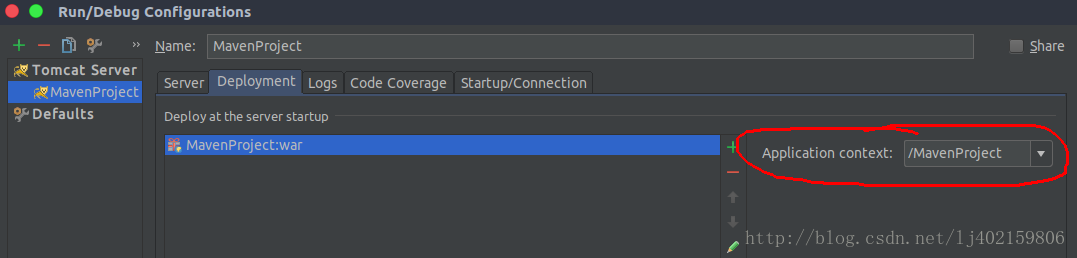
groupid表示项目的包名,artifactid表示项目名   


选择maven版本和配置文件等   


工程名和工程目录,点击finish完成创建   


* 为了在pom.xml文件中添加依赖之后自动引入jar，点击右下角红圈的Enable Auto-Import选项,左上角红圈可以看到mavenproject的项目结构

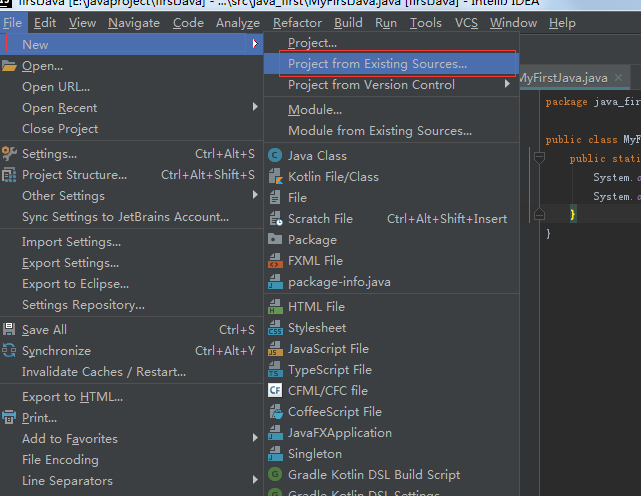
然后就能在pom.xml文件中自行添加maven依赖   


添加tomcat容器   
先打开全局设置,添加一个tomcat容器   
   
设置tomcat目录,仅供参考   
   
添加完成后确定   
  
然后就能在Run/Debug Configurations里设置tomcat了   
****(注意:如果指定了项目的url路径那么application context也要指定路径,见下图红圈)****   
  


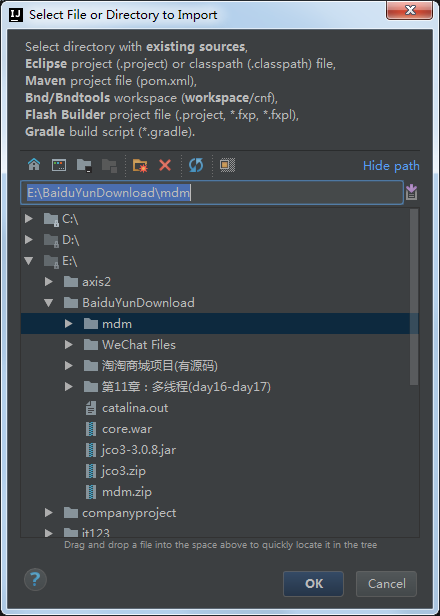
添加完成tomcat后点击运行,运行成功后如下图所示,项目部署成功   
****(注意:请先在命令行停止已经运行的系统的tomcat服务,可使用命令systemctl stop tomcat8,否则无法启动idea的tomcat服务)****   


## 8.导入一个maven项目

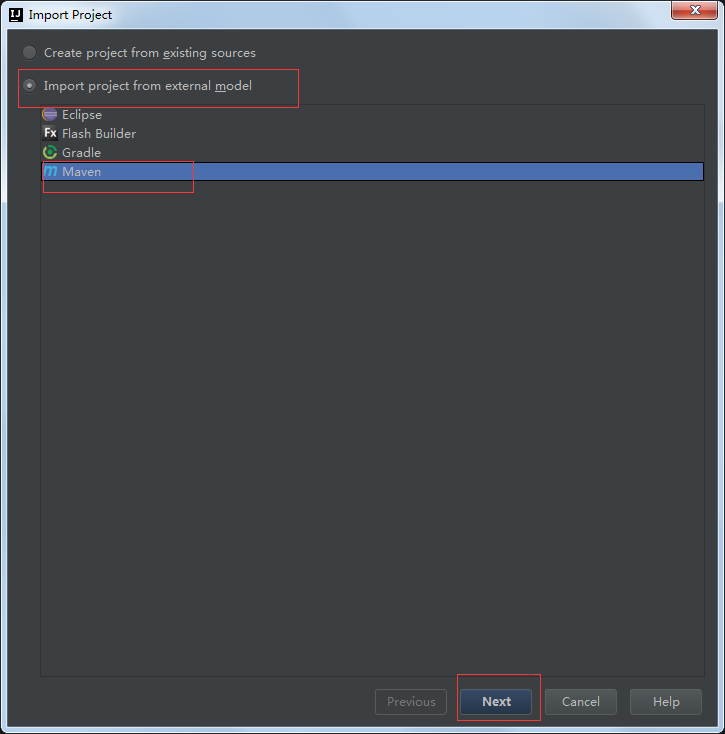
1）File -> Project from Existing Sources...



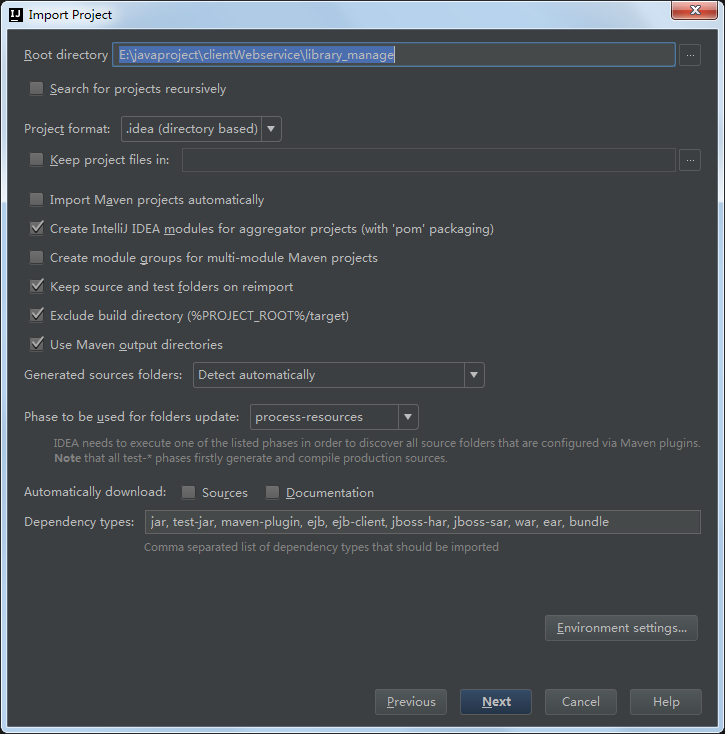
1. 选择项目路径



1. 进入下一个窗口



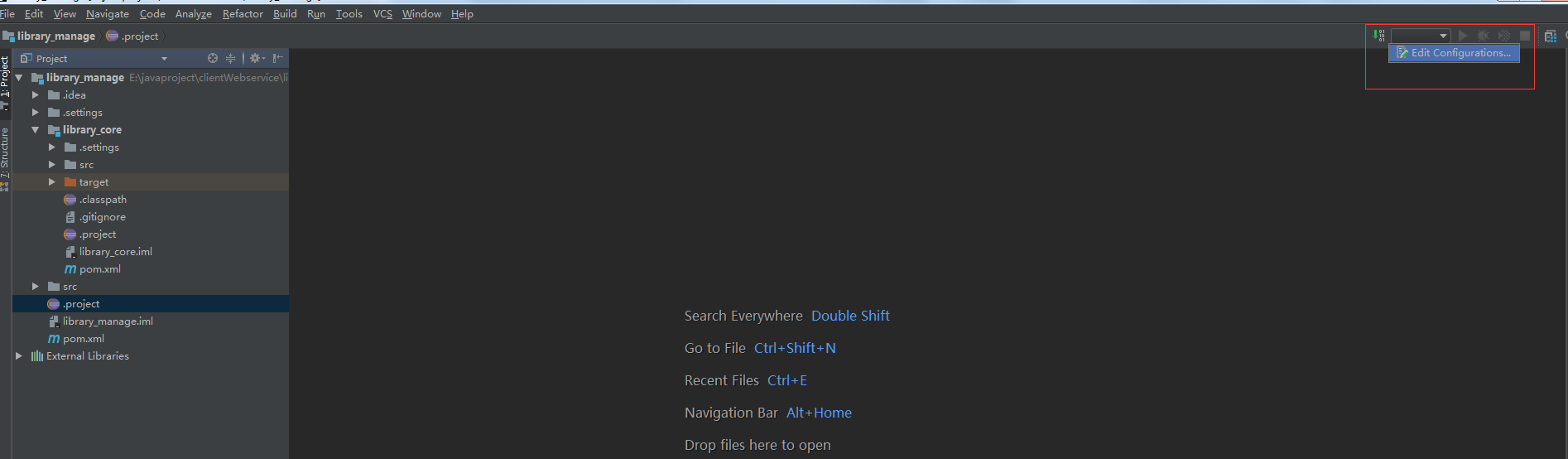
1. 点击next



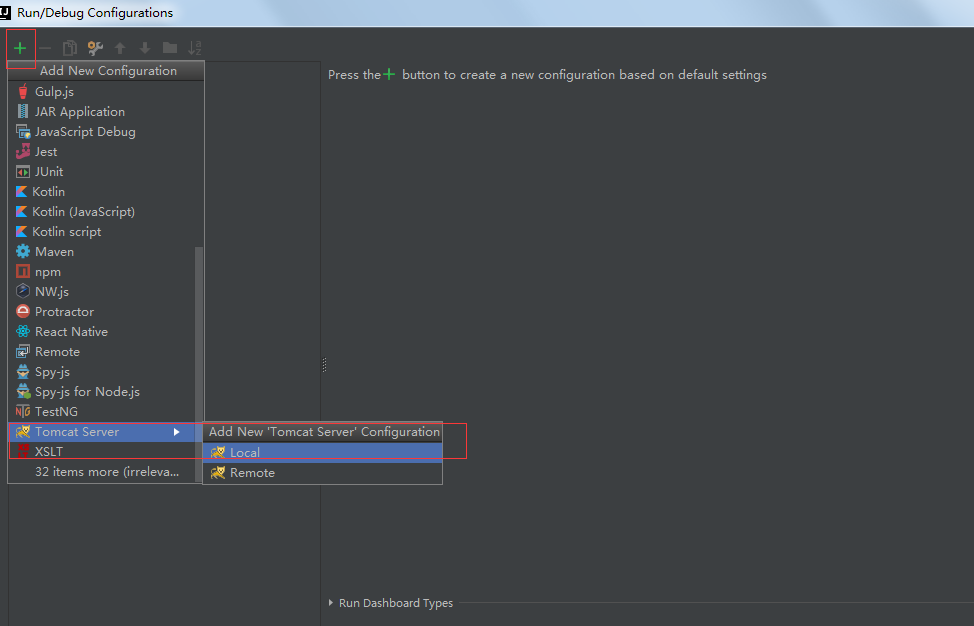
不用修改，默认就行了，next下一个

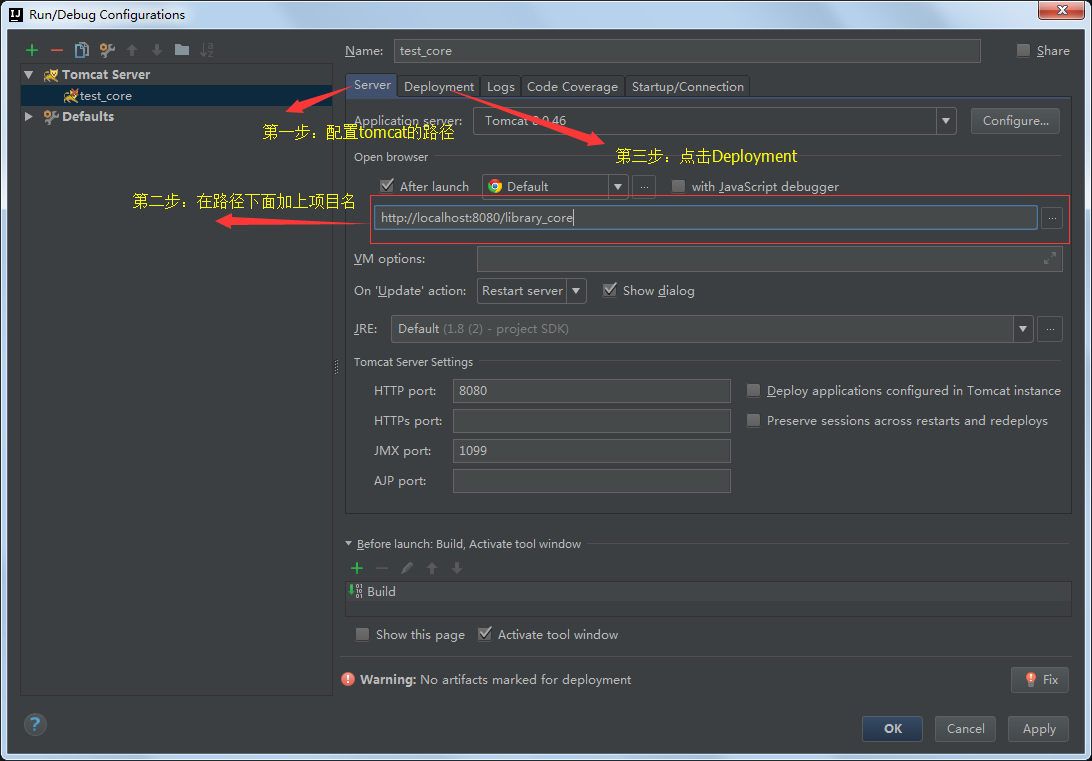
之后一路next就行了，直到finish，加载pom.xml后就可以了

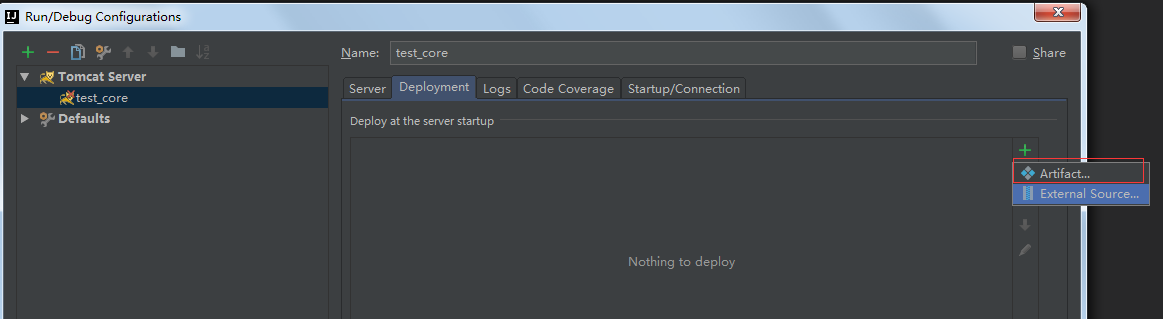
1. 配置tomcat

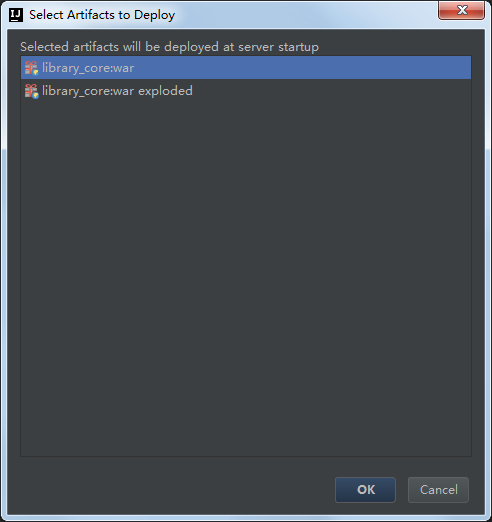


点击红色区域，进入tomcat的配置窗口









上图中我们选择第二个，之后回到server

