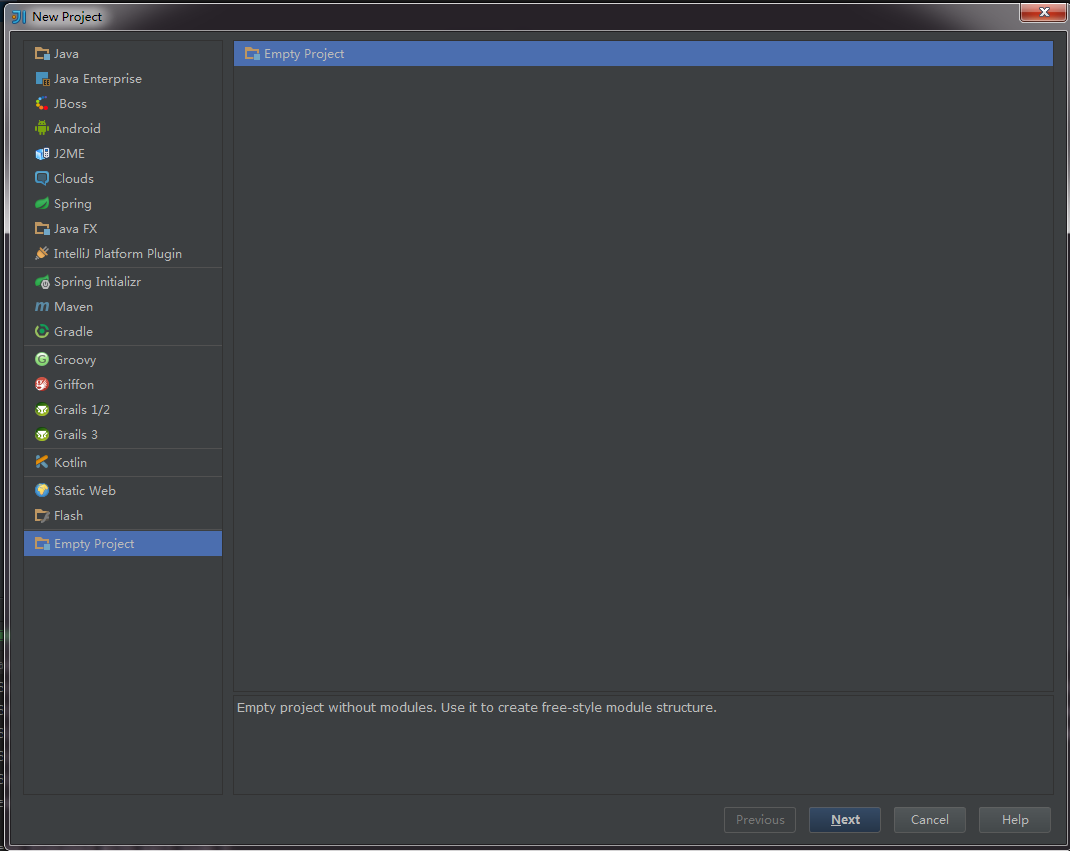
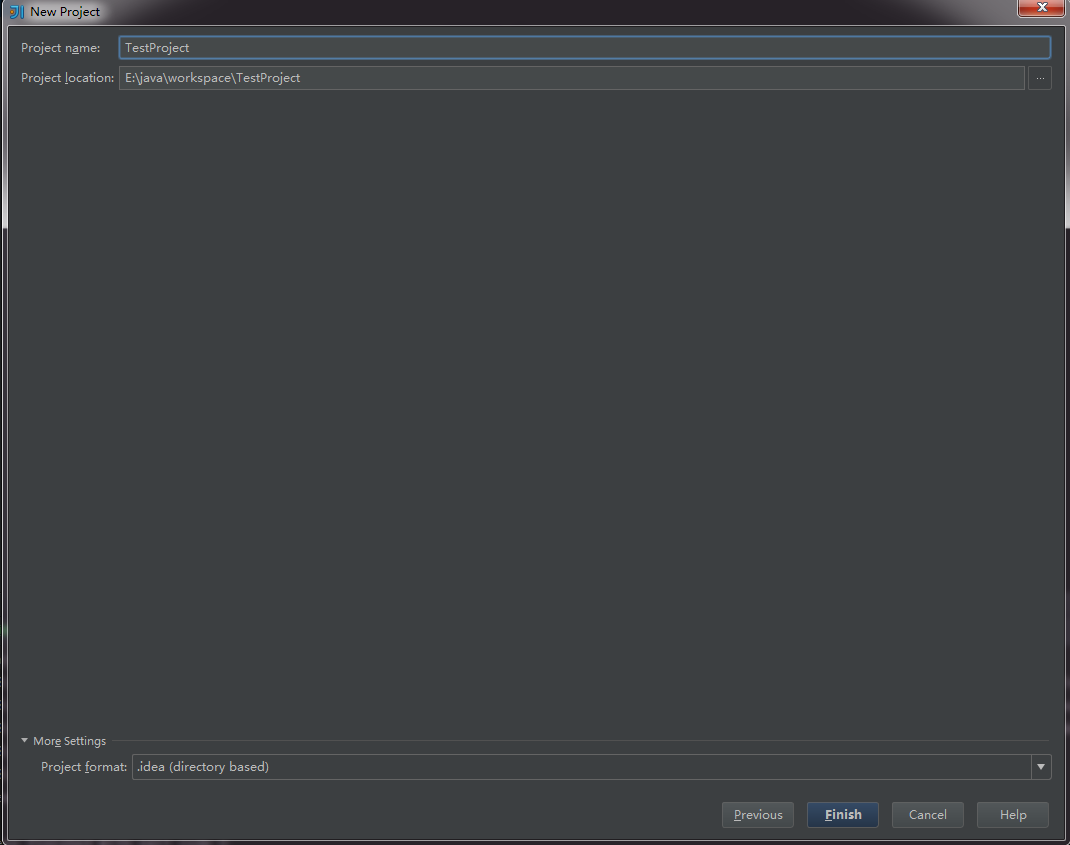
# 从eclipse转Idea

## 使用idea创建普通多模块的java项目：

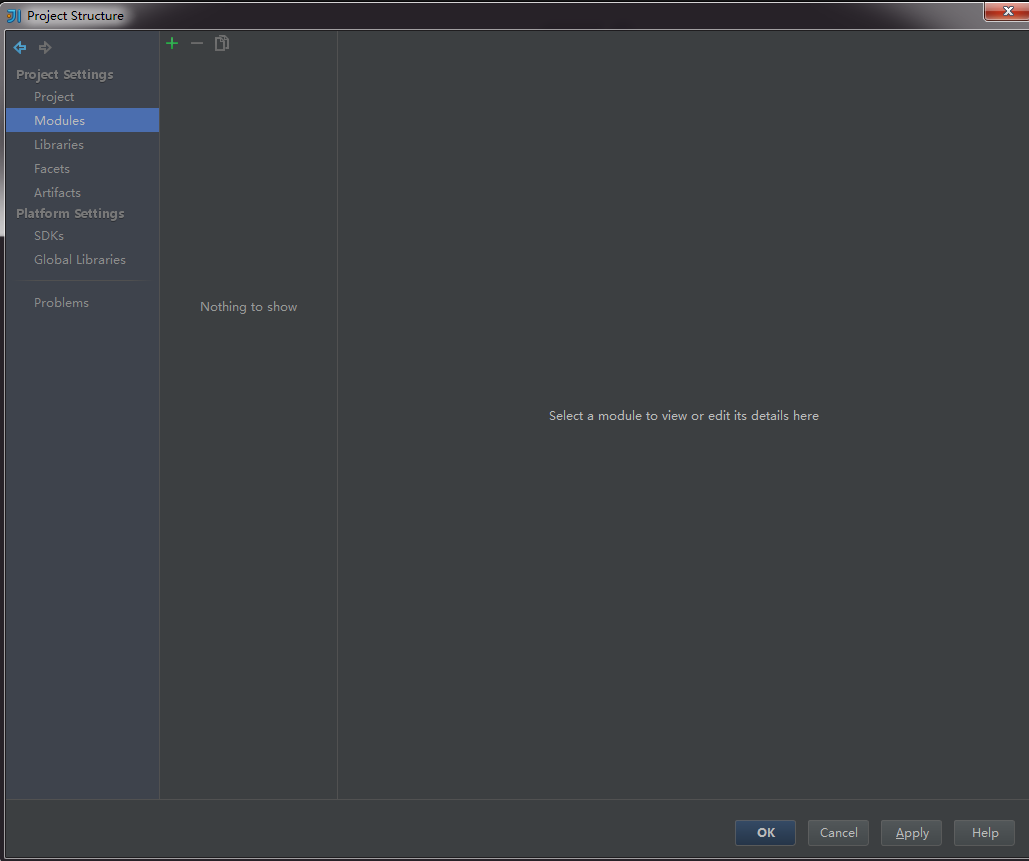
### 打开idea新建Project:选择第一个空的project



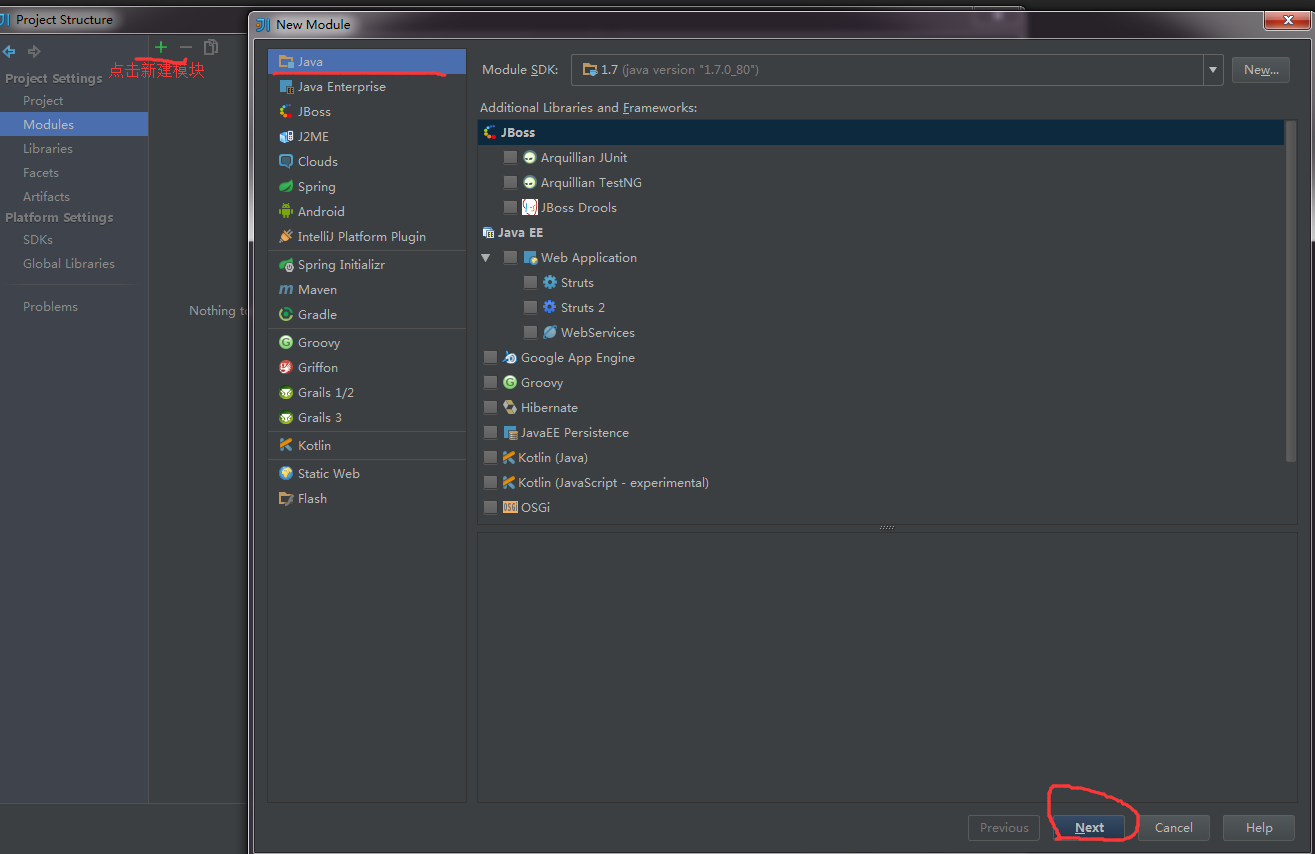
### 填项目名称：



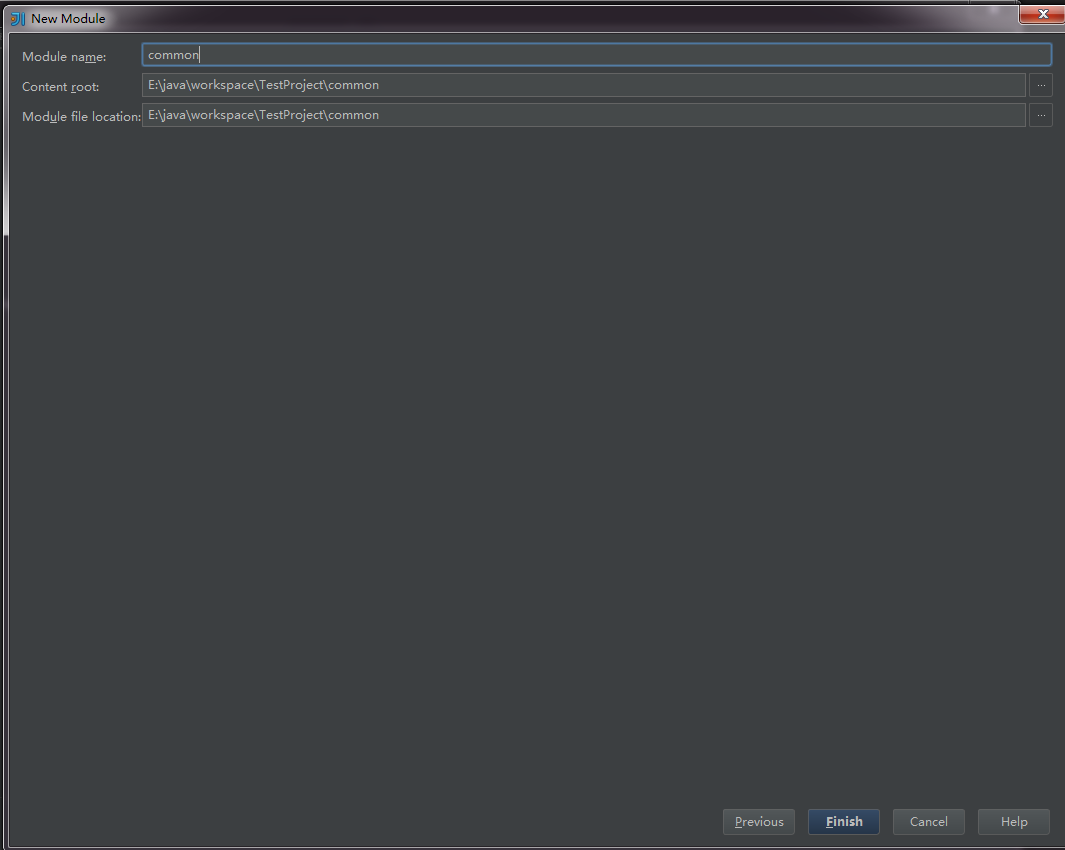
### 点击确定之后，会弹出新建模块的窗口



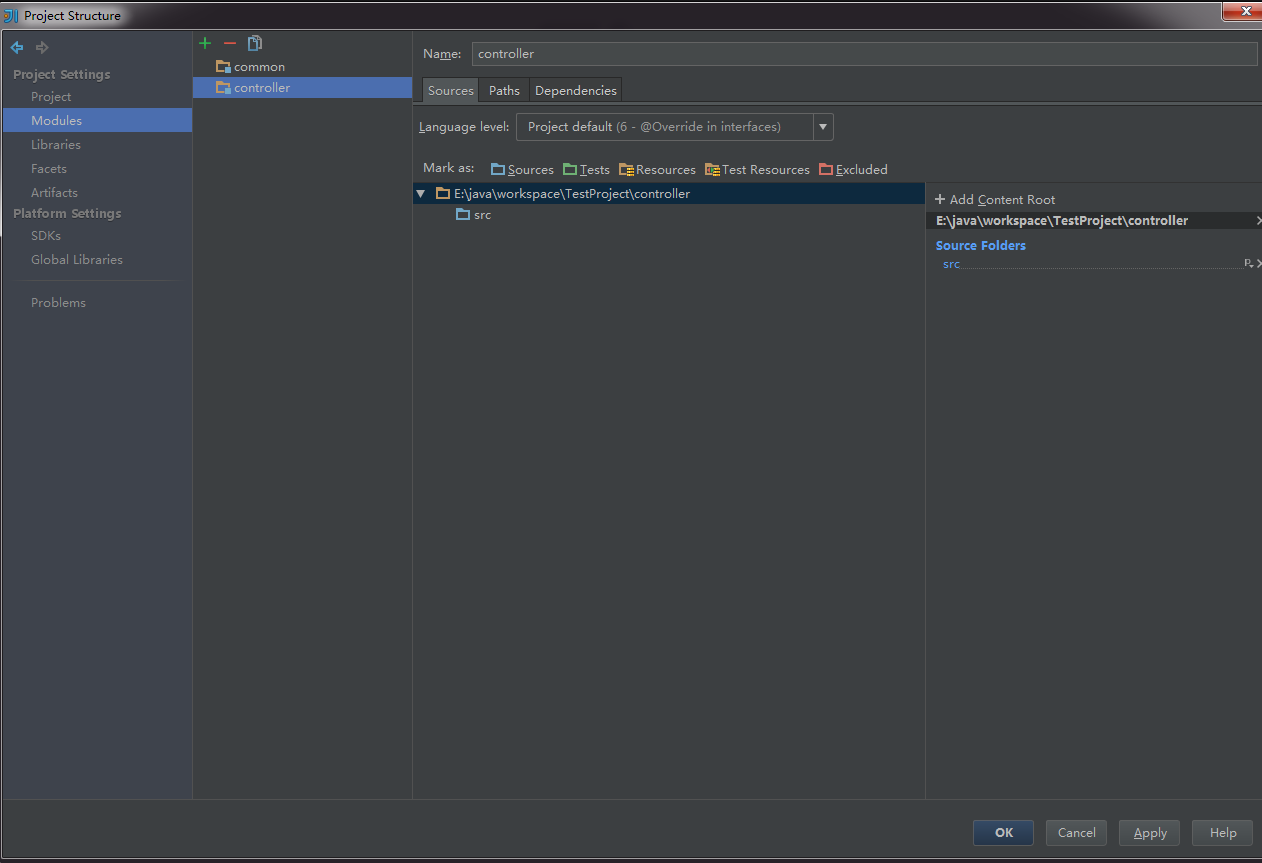
### 新建一个module，红线标记操作



### 填入module名称，点击确定：



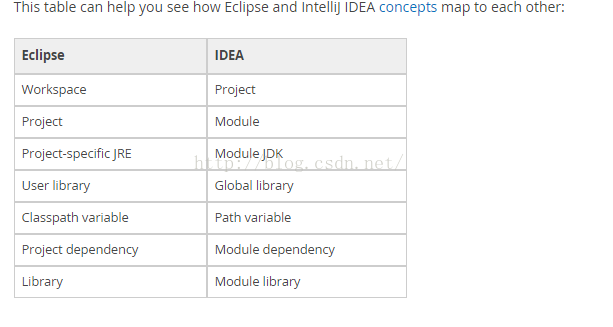
### 同理在新建一个module：controller，可以看到下面:



### 点击确定后打开项目是这样的：



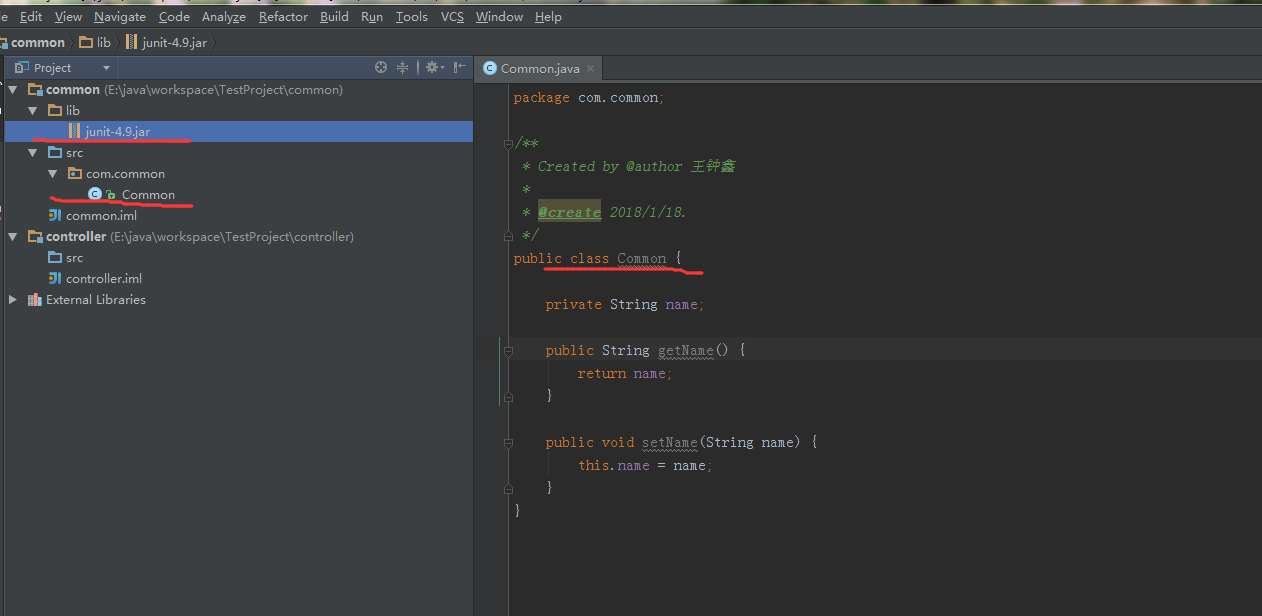
### Idea和eclipse两者在文件结构上的不同：



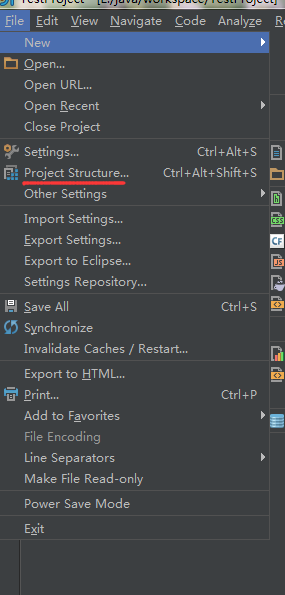
Idea没有工作空间的概念，所以无法创建多个project，只能在project上面创建多个module。

### 开始操作项目：

##### 在common模块上面新建lib目录用来放jar，并且在src目录下新建一个类Common

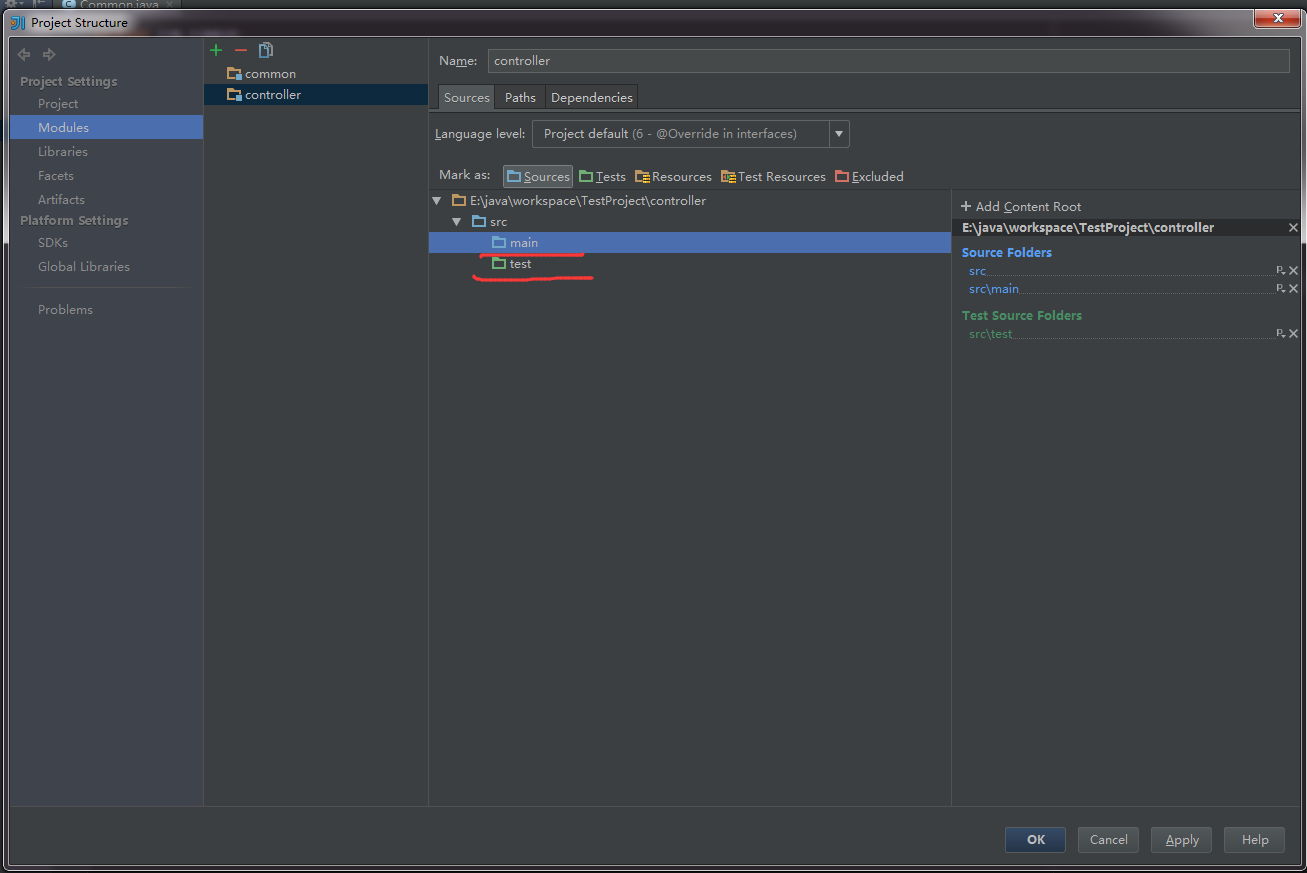


打开：**Project Struction**

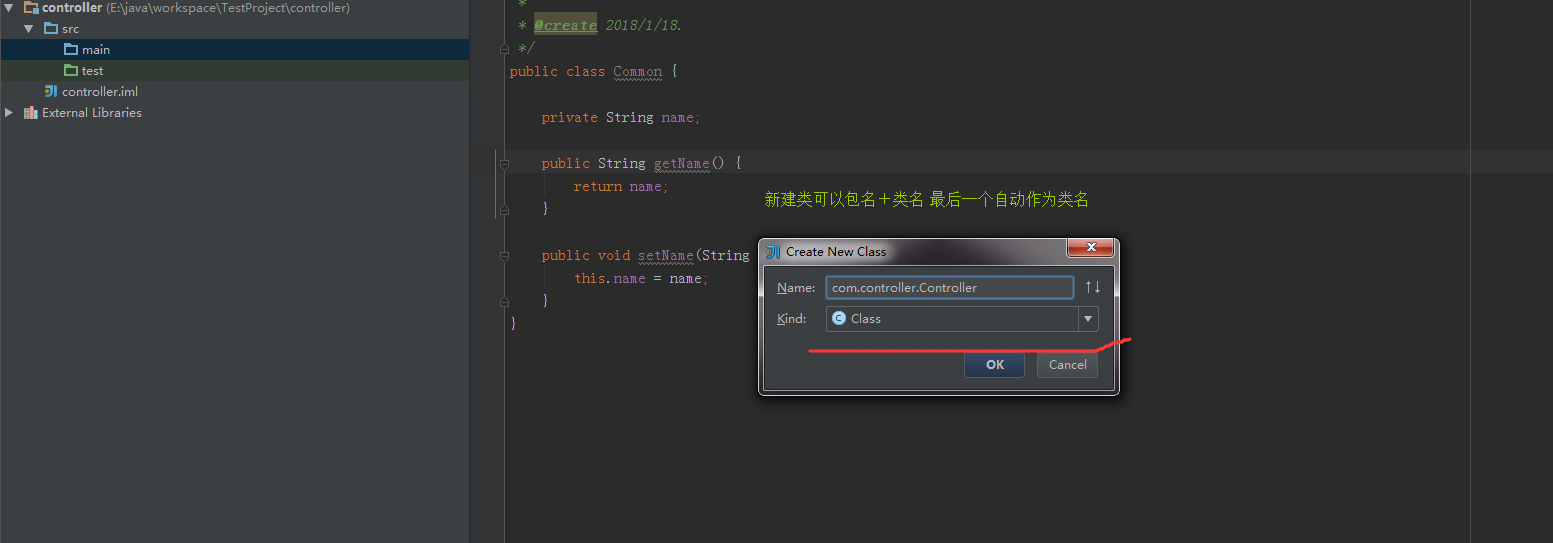


#### 选择controller模块：

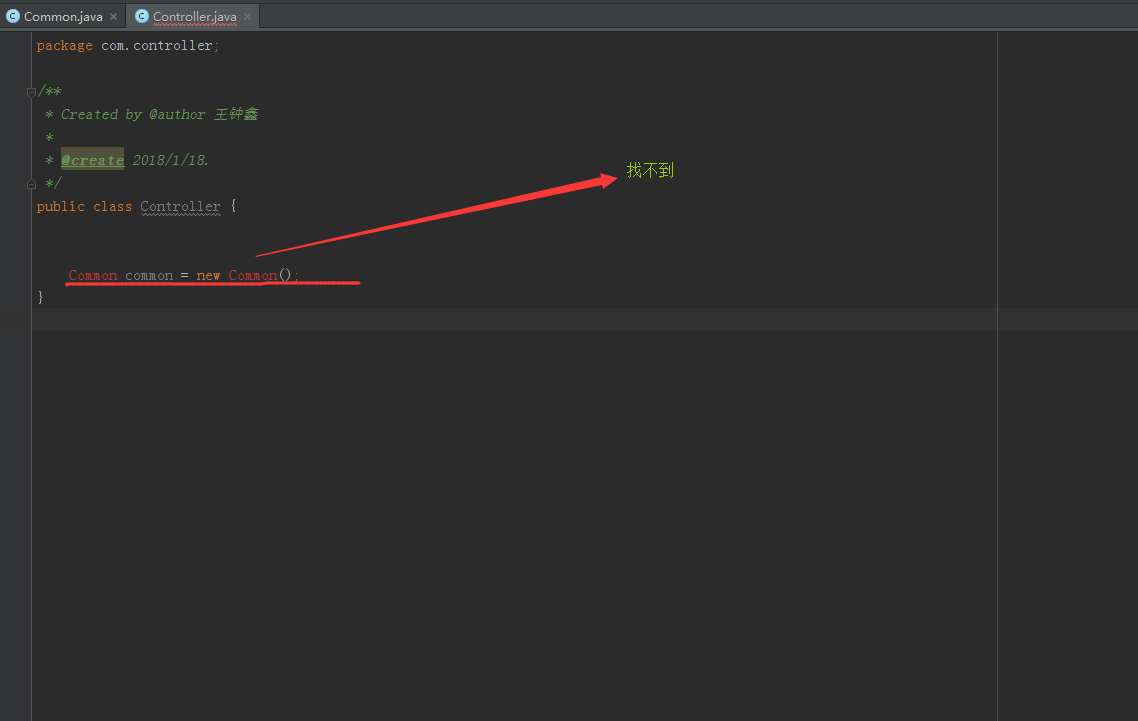
在src下面新建main和test目录:



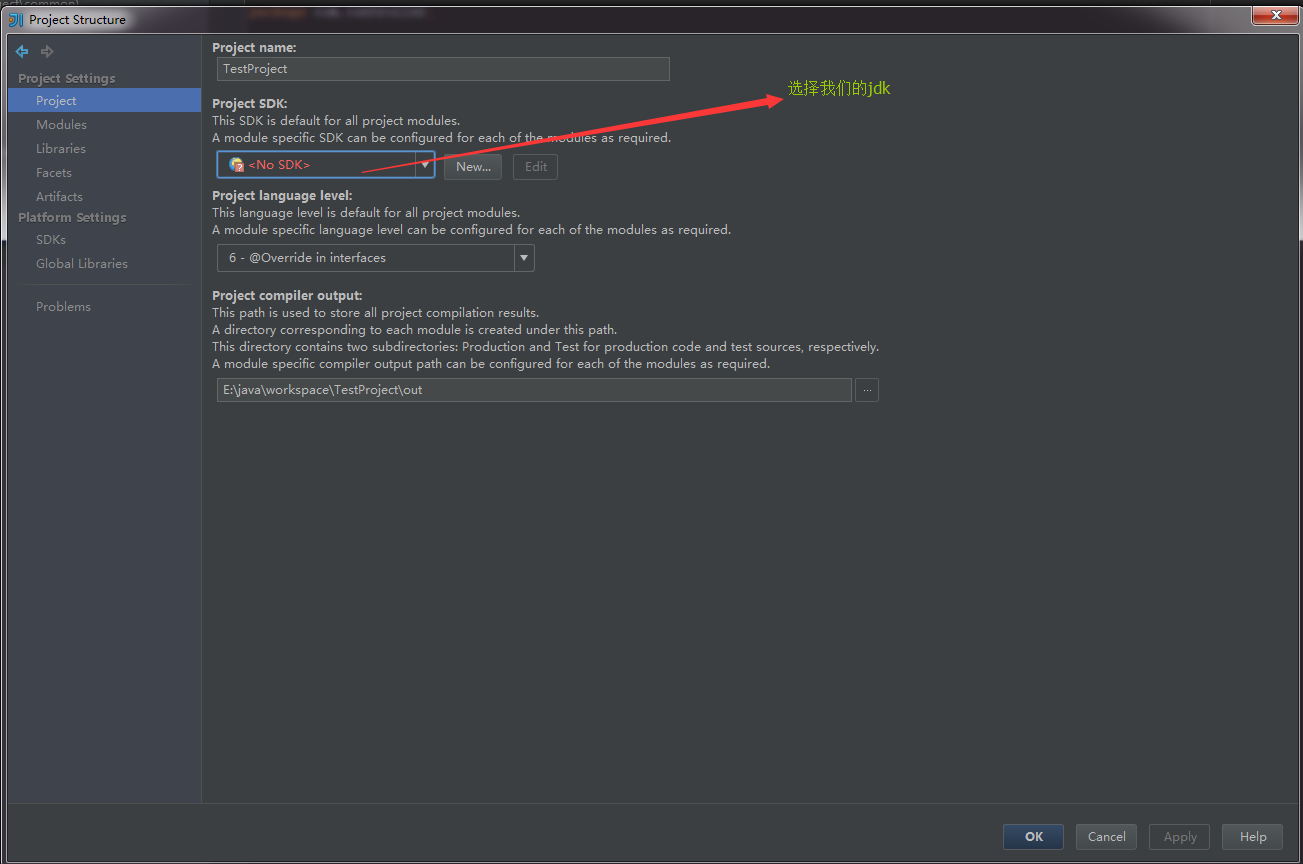
#### 此刻我们在main下面新建Controller类：



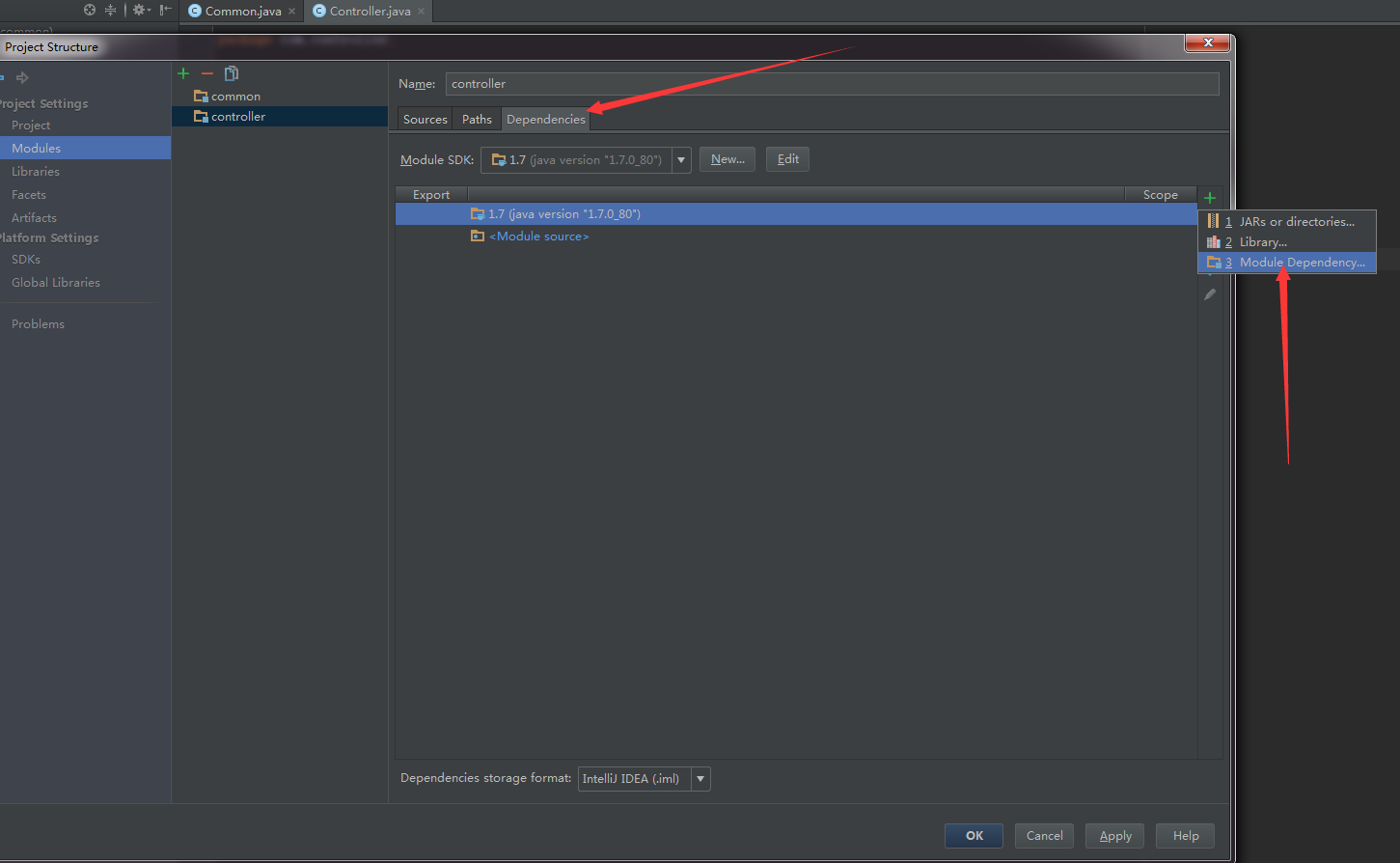
### 在这里访问common模块的对象会出现找不到对象：

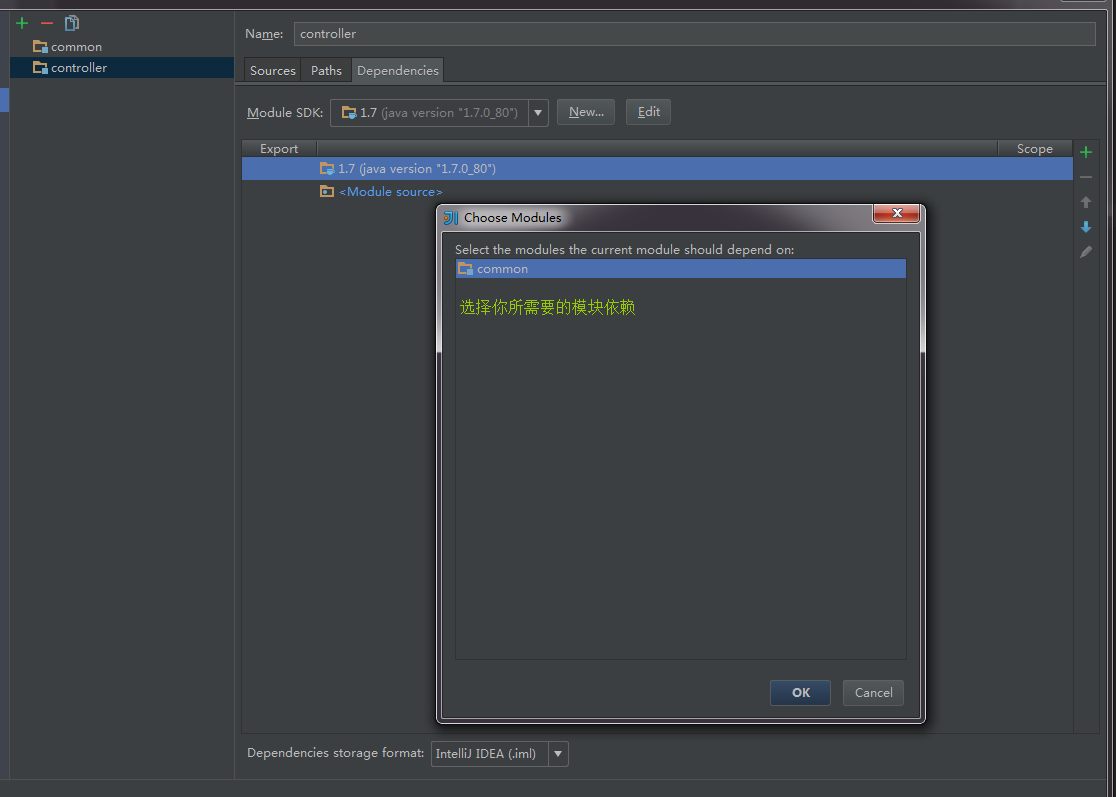


### 此刻需要使用模块的关联，把每个模块都给关联起来：打开：Project Struction

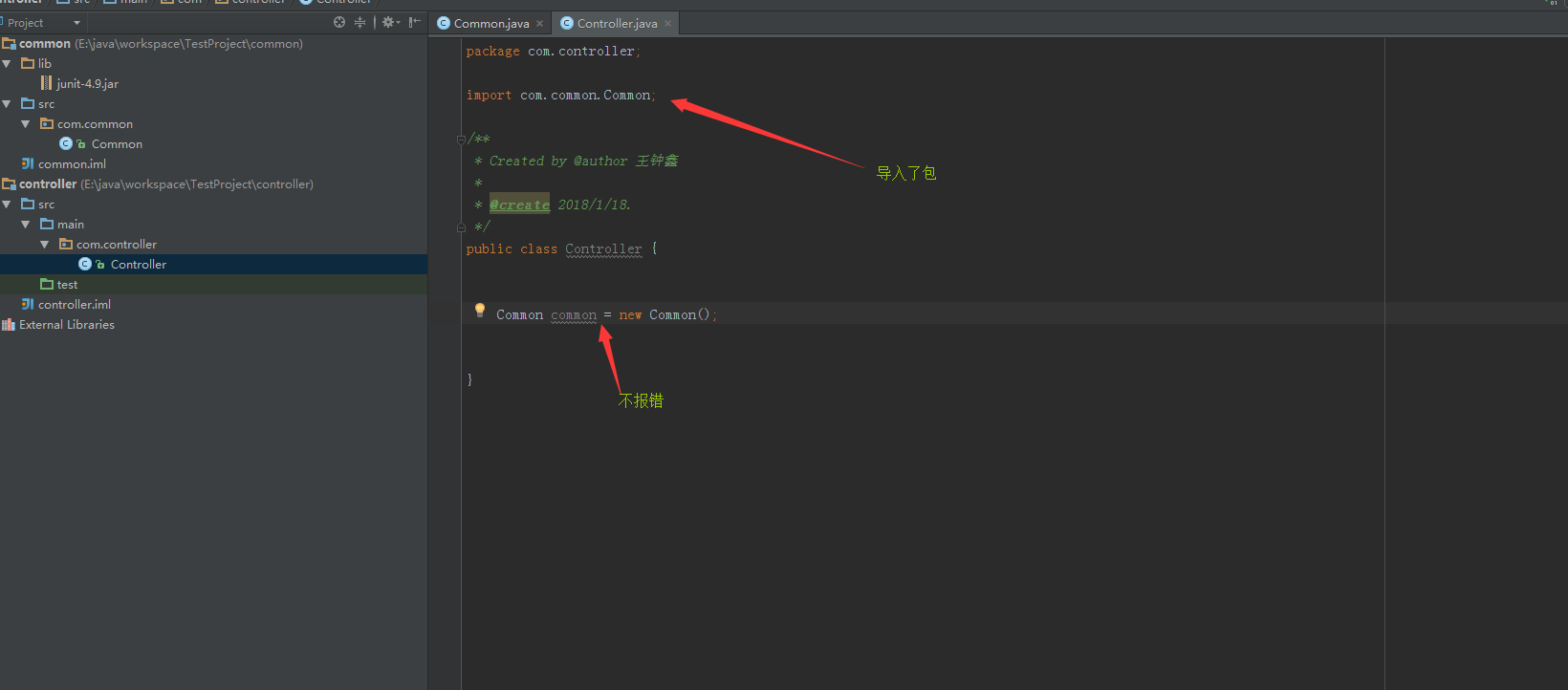


### 选中对应的模块，添加对应的模块依赖

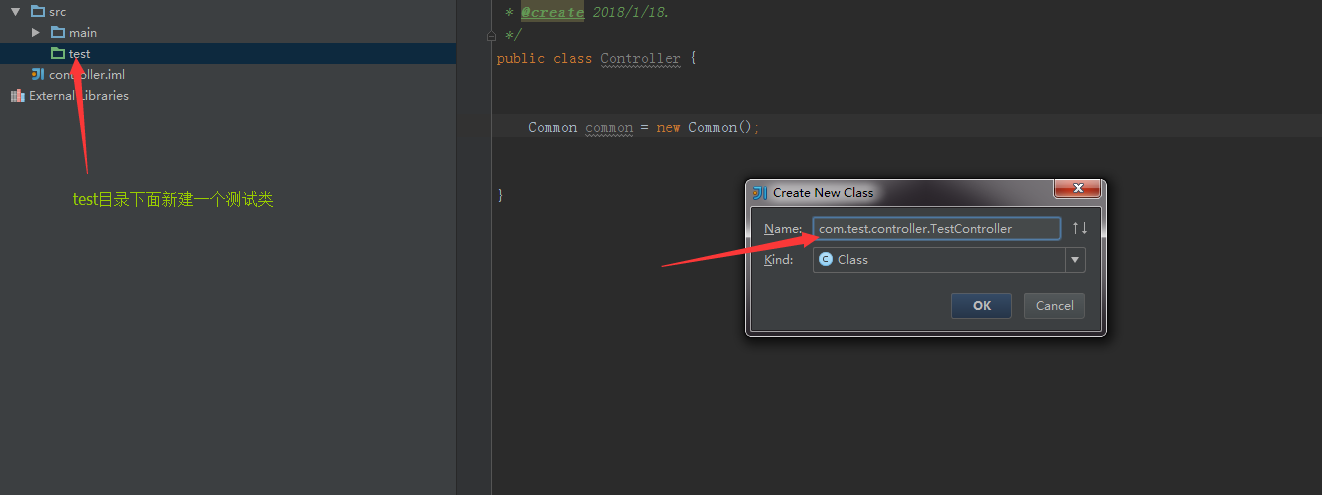


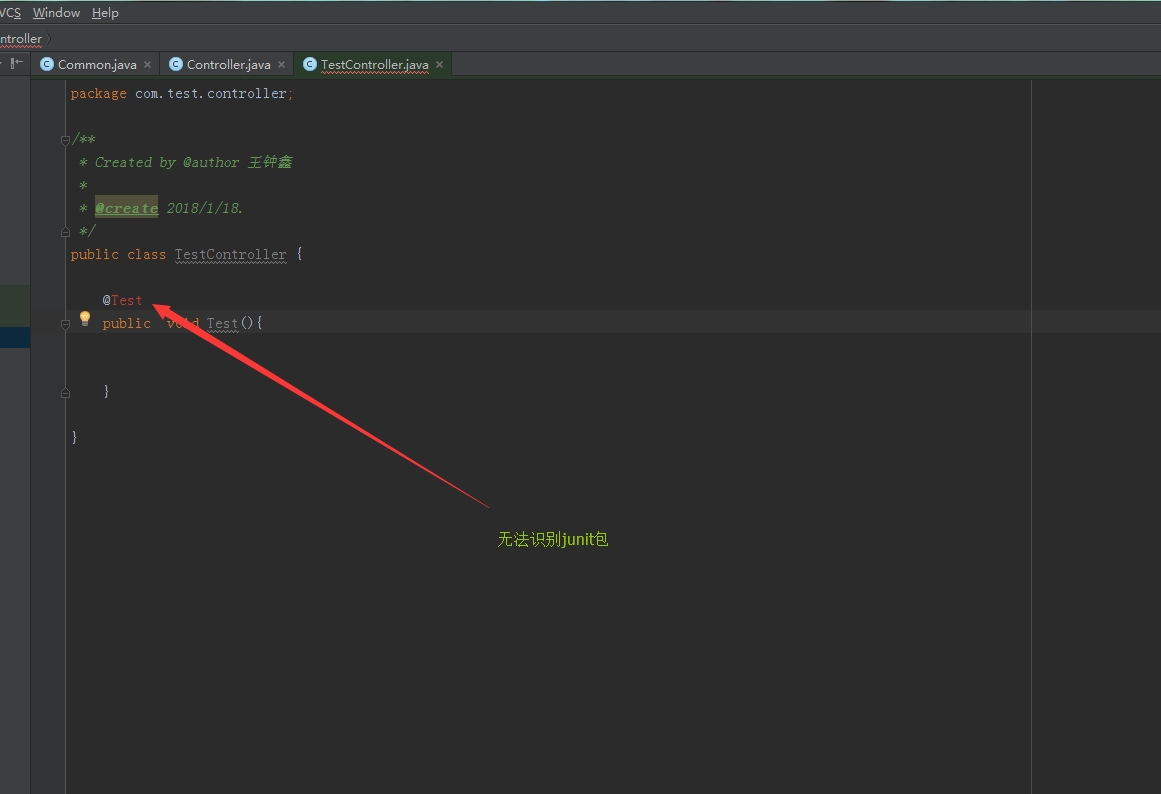


### 点击确定后，Apply一下，重回界面中：

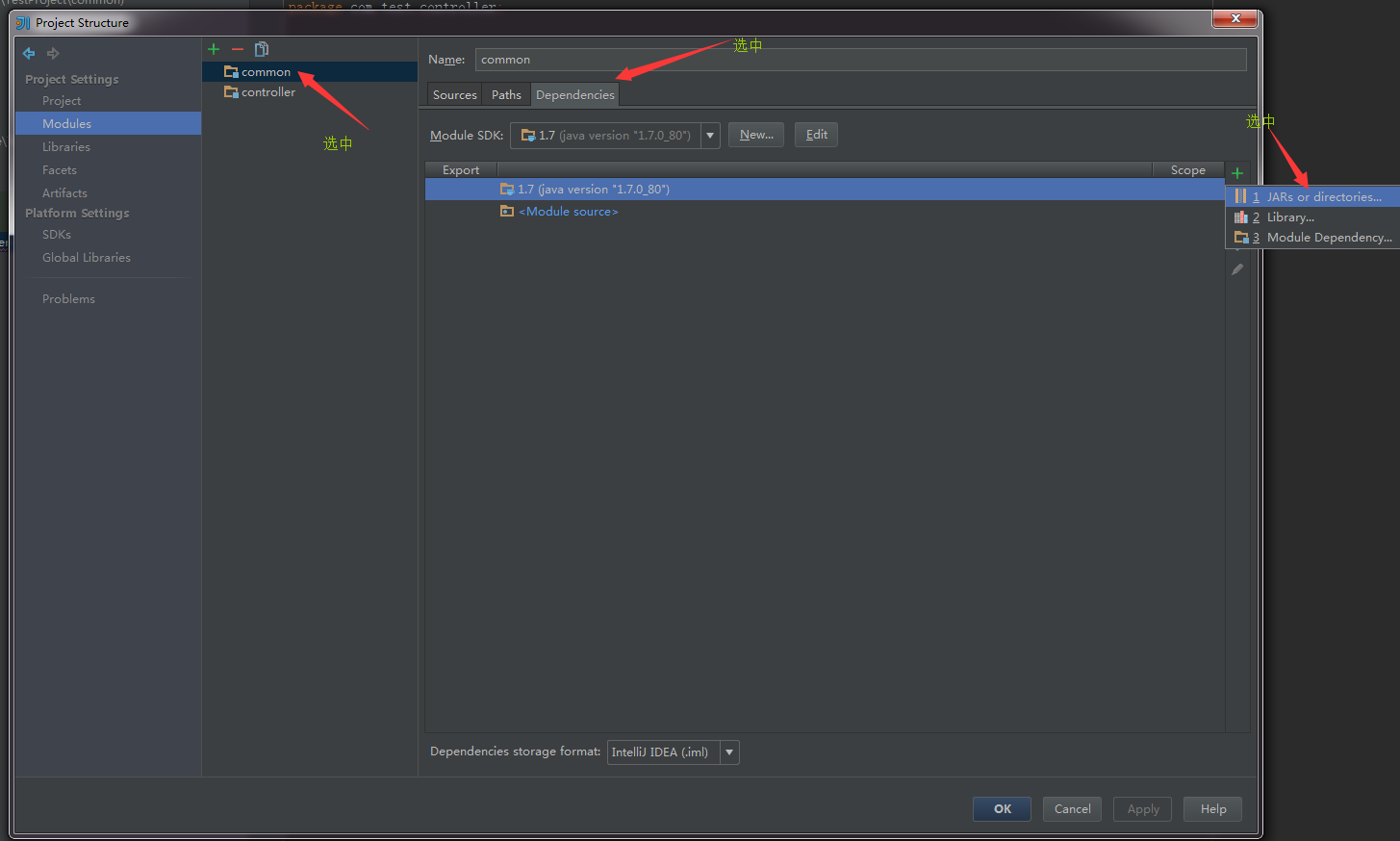


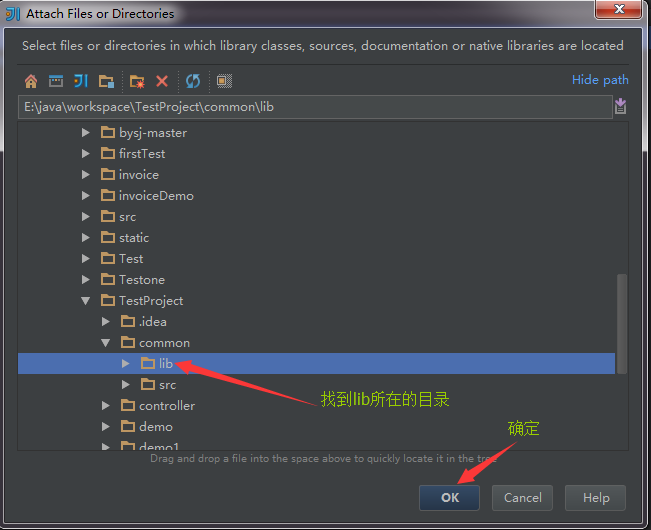
### 操作controller下面的test，新建一个测试类：

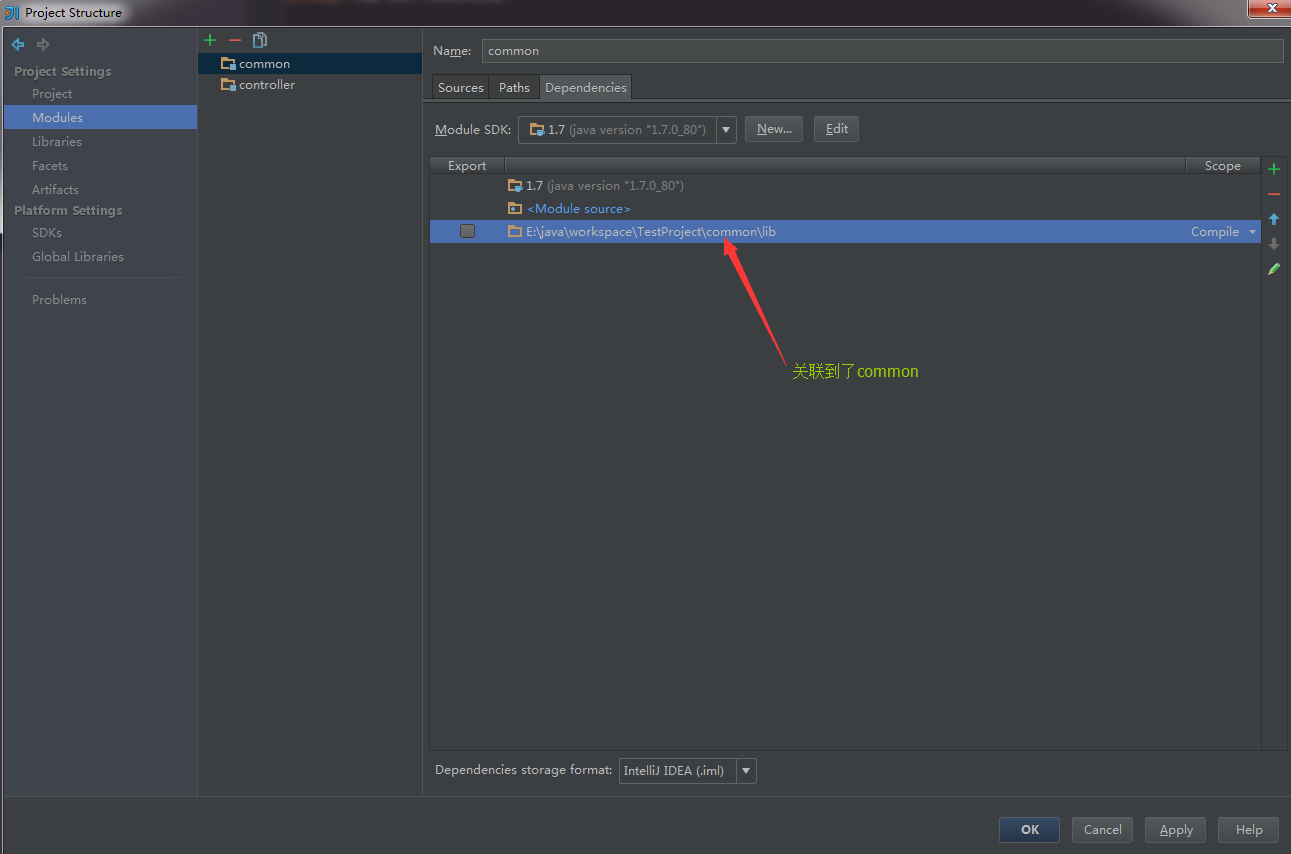


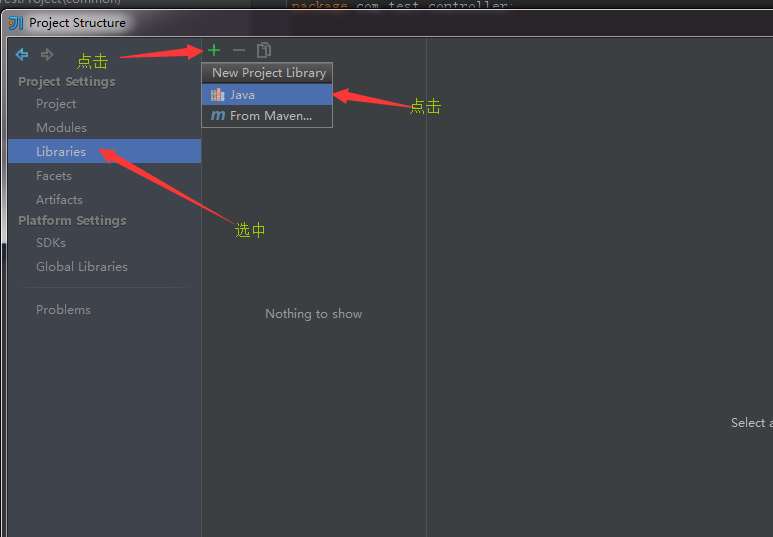


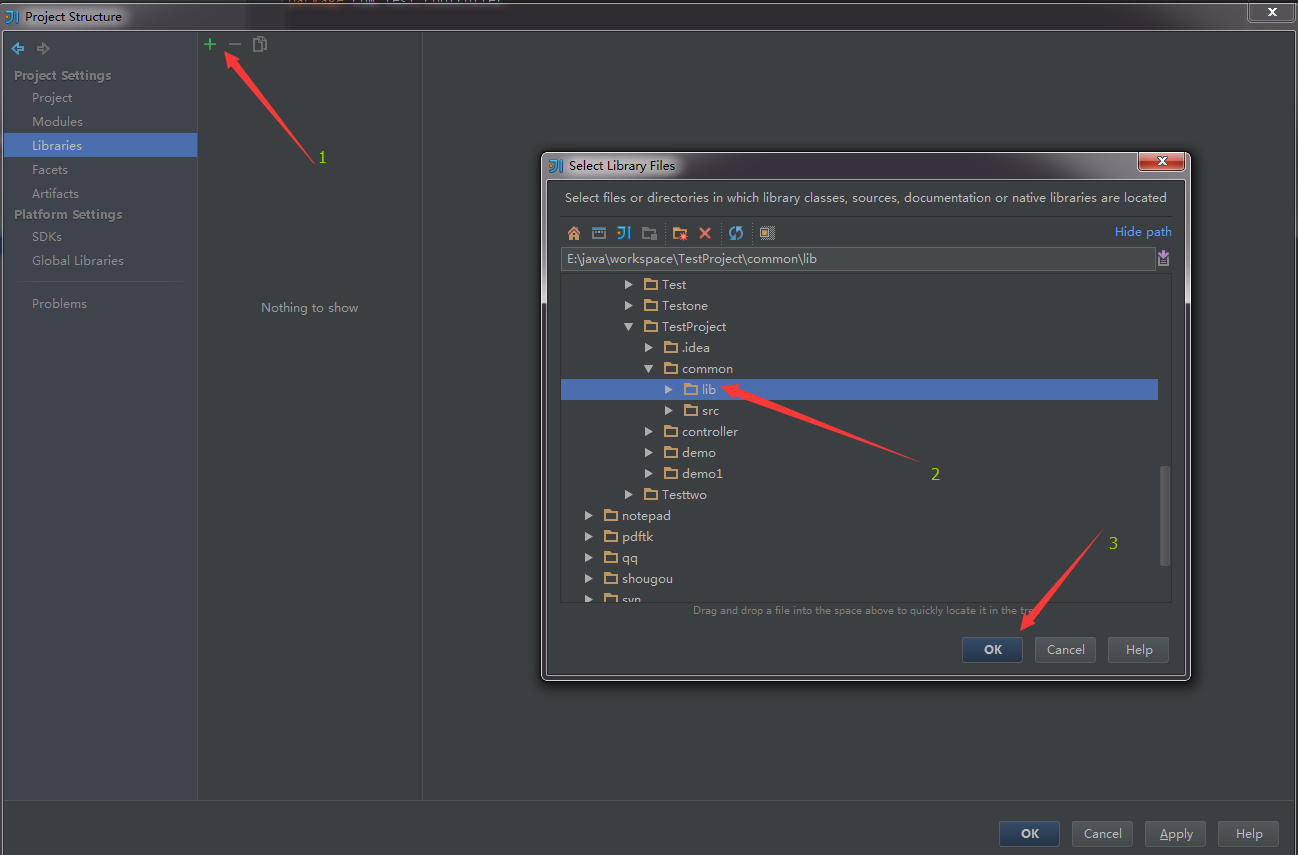
### 引入项目的外部jar包依赖：打开Project Struction

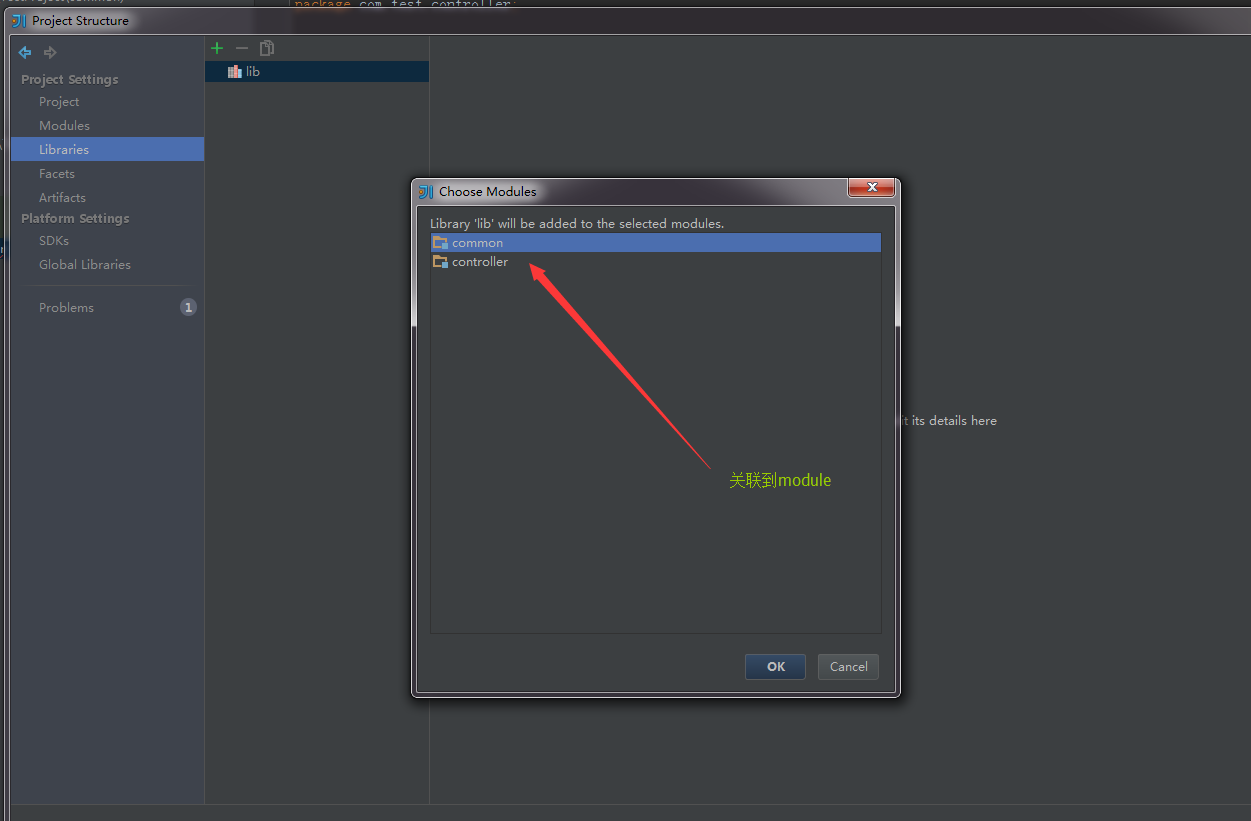




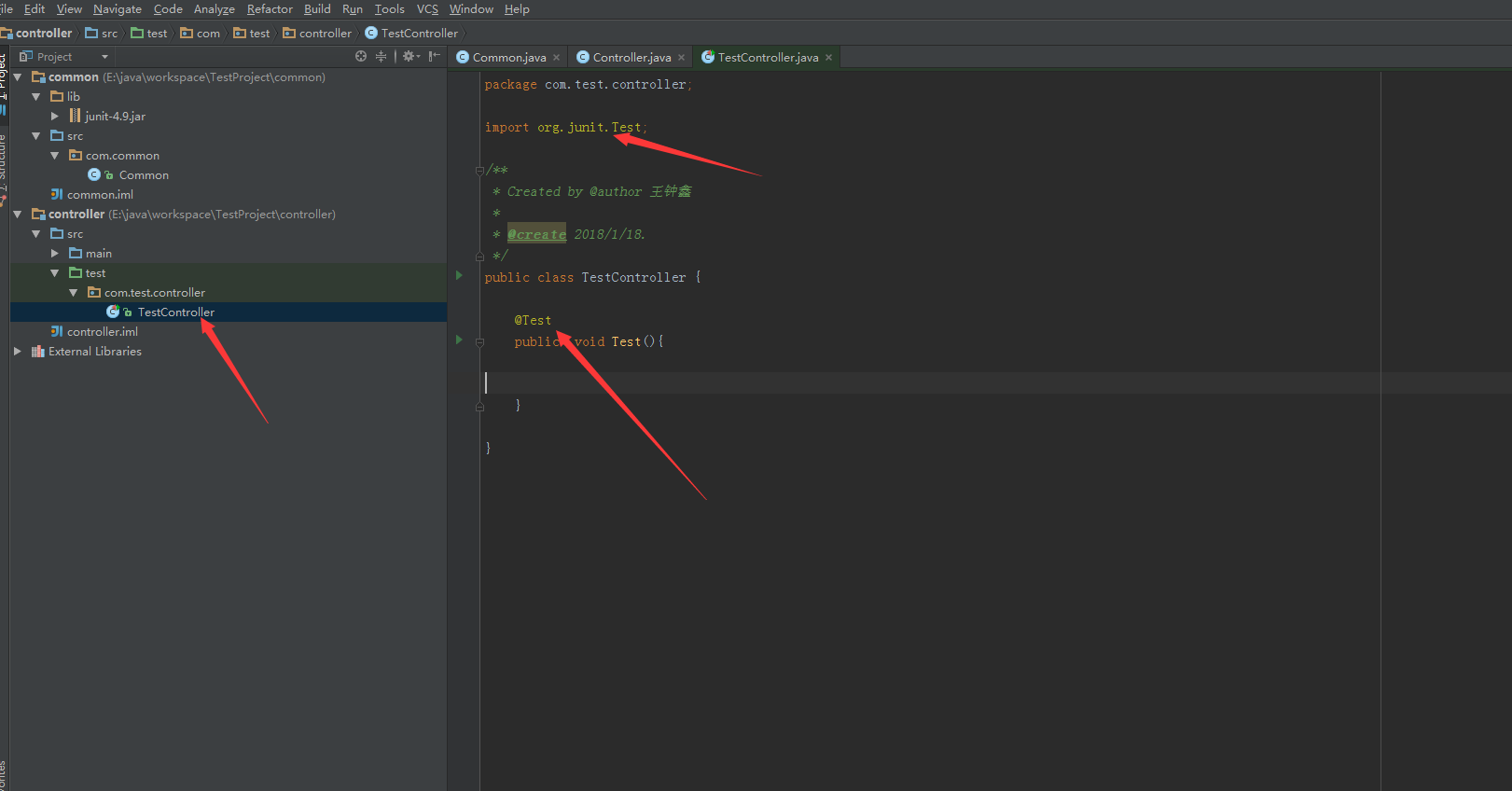








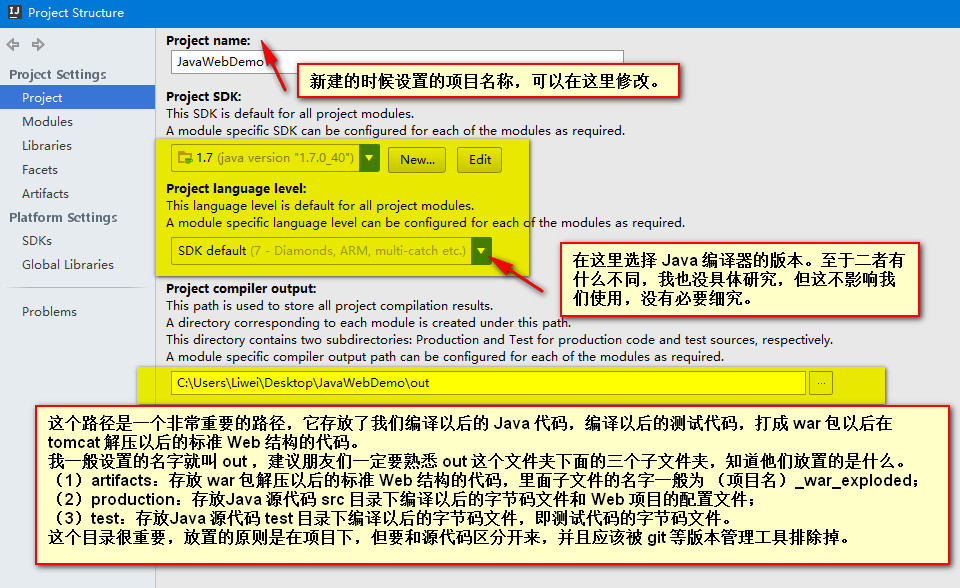
### 重回界面测试：成功不报错了



### 详解 Project Struction

然后，我们打开 Project Struction （打开 Project Struction 的方法很多，例如通过菜单 File -> Project Struction… 或者使用快捷键 ctrl + shift + alt + S，或者通过菜单栏的快捷按钮）。   
我们首先看 【Project】部分。

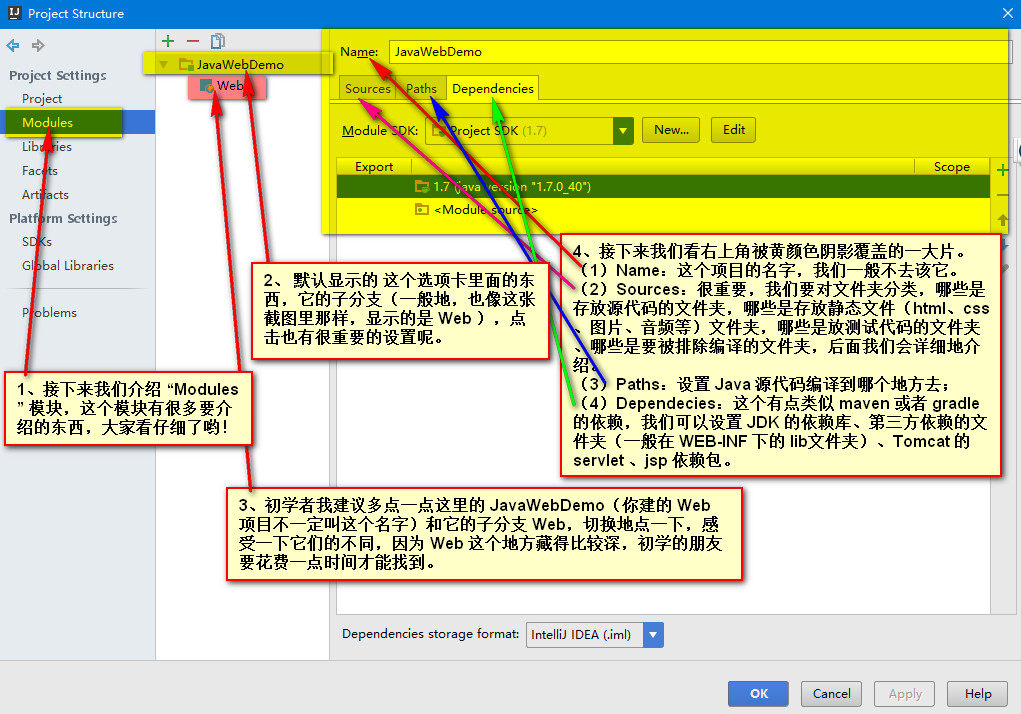
#### 详解 Project Struction 之 Project 模块



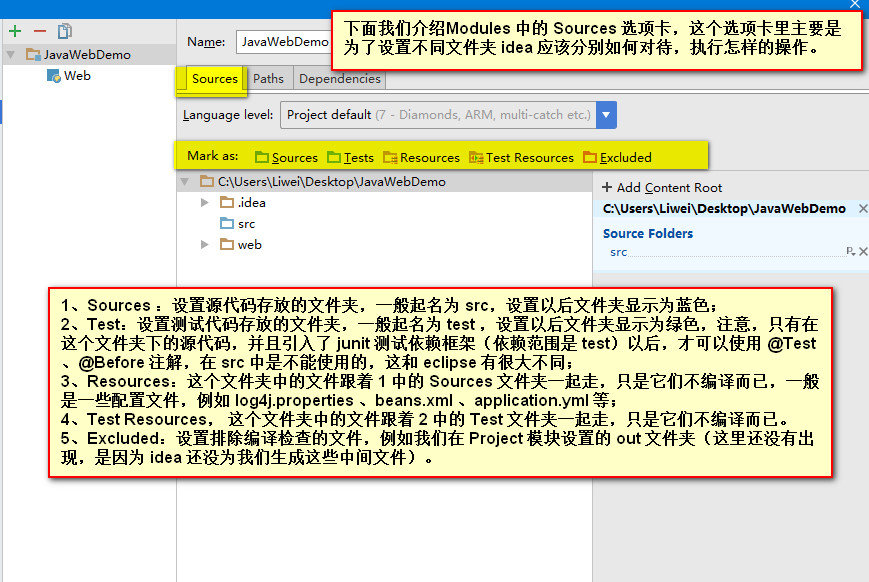
对 Project compiler output 的解释：   
这个路径是一个非常重要的路径，它存放了我们编译以后的 Java 代码，编译以后的测试代码，打成 war 包以后在 tomcat 解压以后的标准 Web 结构的代码。   
我一般设置的名字就叫 out ，建议朋友们一定要熟悉 out 这个文件夹下面的三个子文件夹，知道他们放置的是什么。   
（1）artifacts：存放 war 包解压以后的标准 Web 结构的代码，里面子文件的名字一般为 （项目名）\_war\_exploded；   
（2）production：存放Java 源代码 src 目录下编译以后的字节码文件和 Web 项目的配置文件；   
（3）test：存放Java 源代码 test 目录下编译以后的字节码文件，即测试代码的字节码文件。   
这个目录很重要，放置的原则是在项目下，但要和源代码区分开来，并且应该被 git 等版本管理工具排除掉。

#### 详解 Project Struction 之 Modules 模块

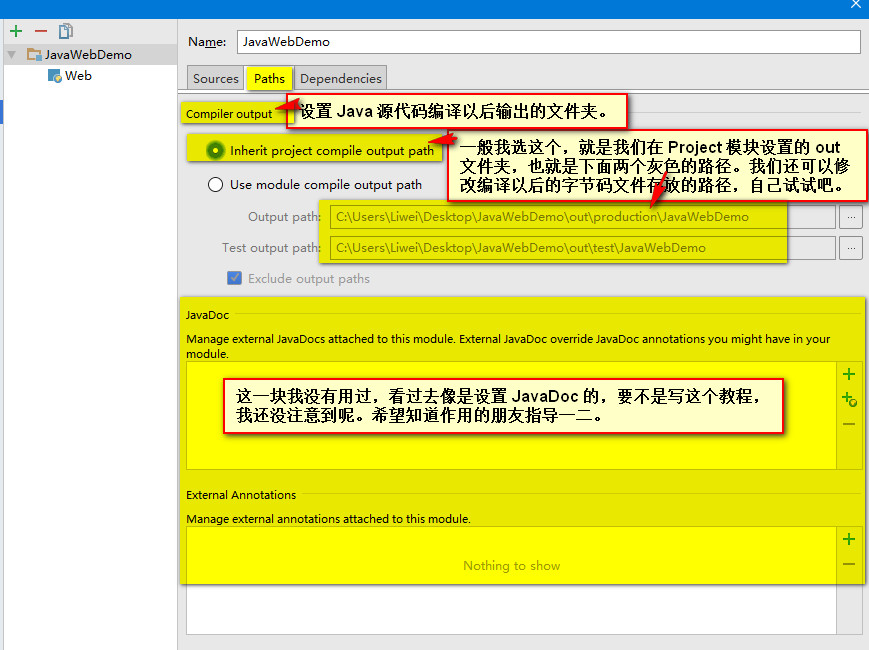
##### （1）大体认识



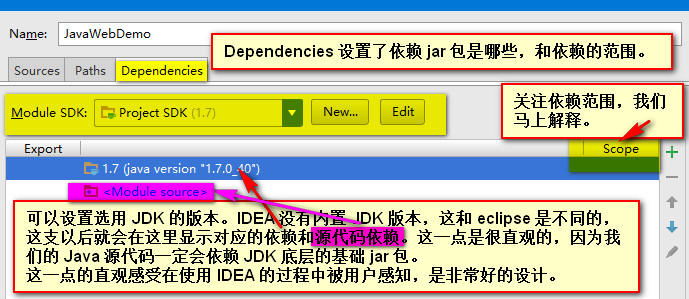
##### （2）详解 Sources 选项卡



##### （3）详解 Paths 选项卡

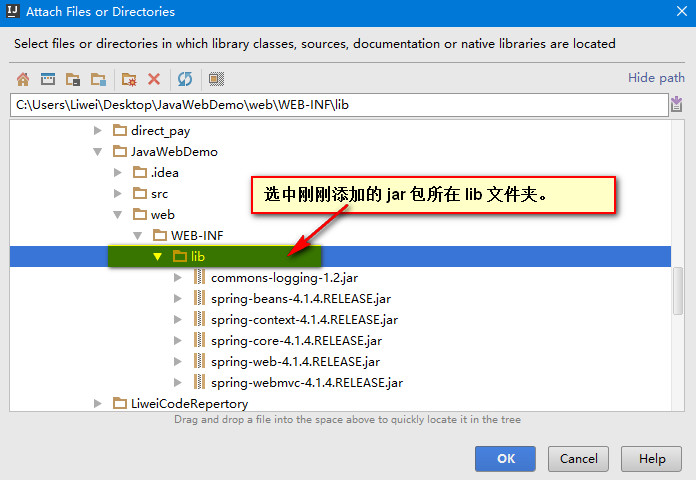


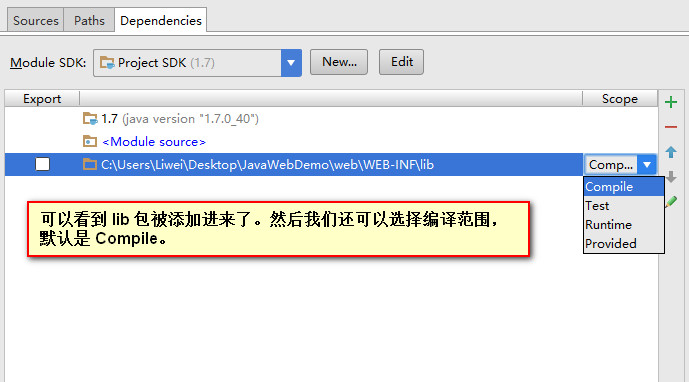
##### （4）详解 Dependencies 选项卡

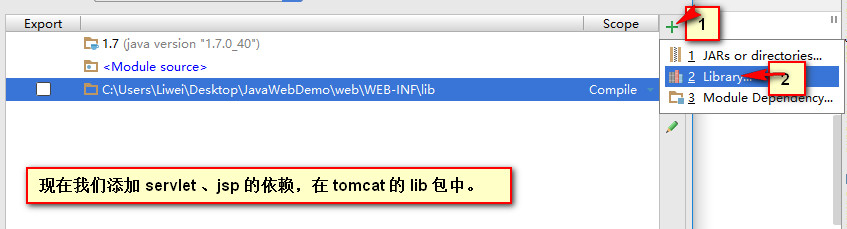


##### （5）详解如何添加依赖

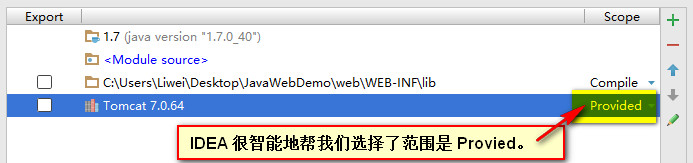
第 1 种情况：添加 lib 包下的依赖：   

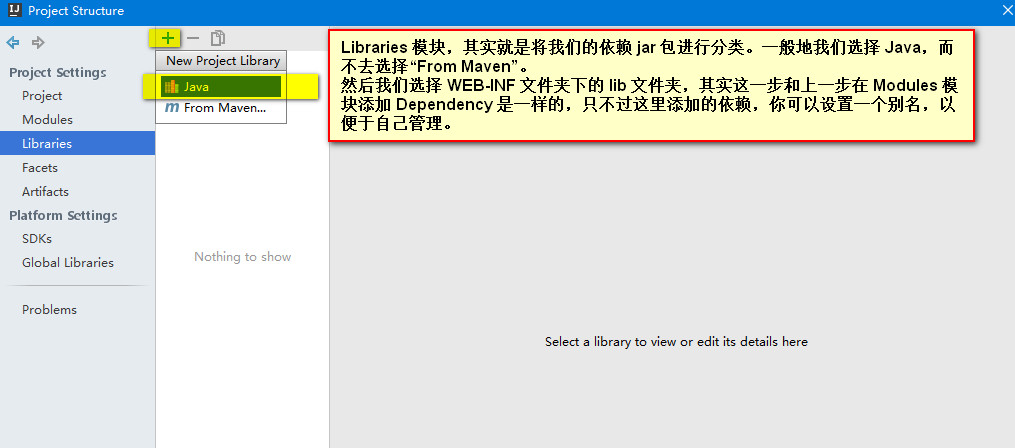


第 2 种情况：添加 tomcat 依赖，这种依赖是一种 provided 的依赖：   






#### 详解 Project Struction 之 Libraries 模块

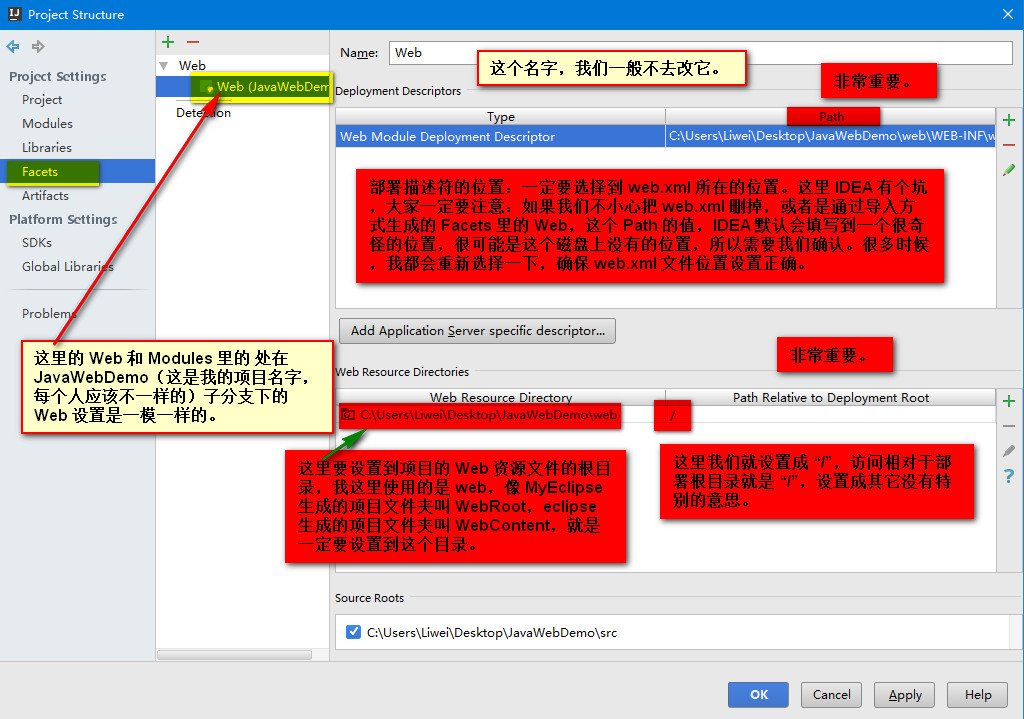


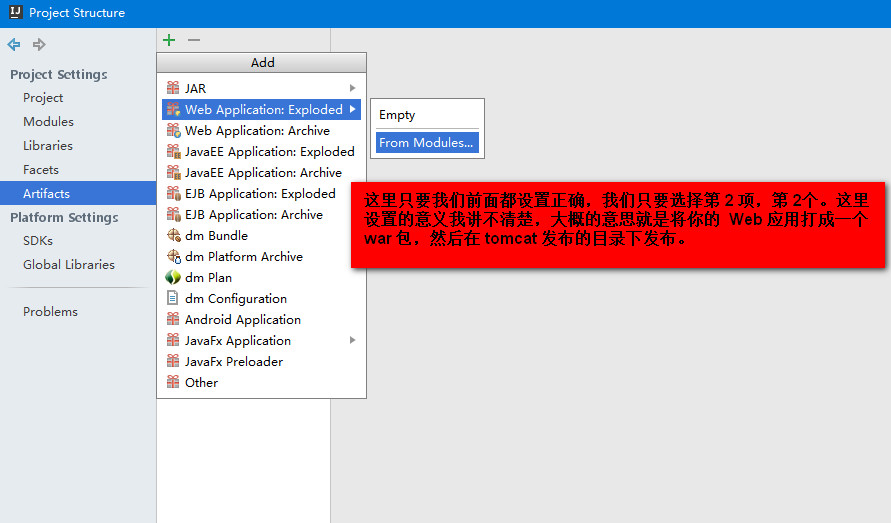
可以看到：   

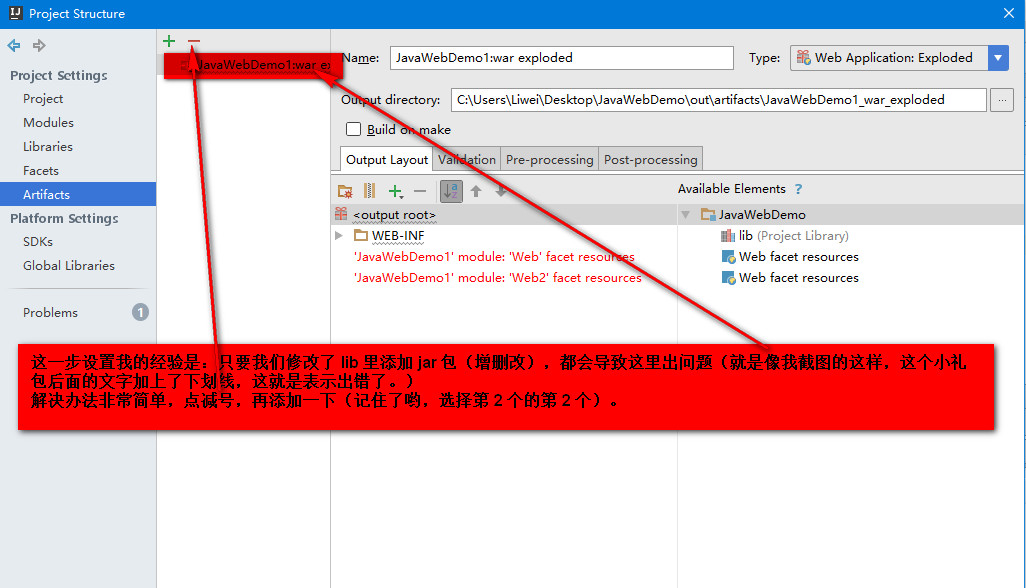

#### 详解 Project Struction 之 Facets 模块

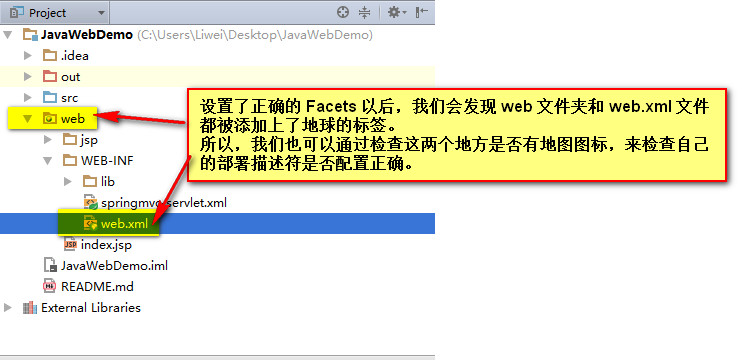
重要的事情说三遍。

**注意：这一部分的配置非常重要，一定要设置对。而且如果不认真检查， IDEA 会“自作聪明”地帮我们设置了一个根本不存在的路径。如果设置不对，非常影响我们的情绪。**   
**注意：这一部分的配置非常重要，一定要设置对。而且如果不认真检查， IDEA 会“自作聪明”地帮我们设置了一个根本不存在的路径。如果设置不对，非常影响我们的情绪。**   
**注意：这一部分的配置非常重要，一定要设置对。而且如果不认真检查， IDEA 会“自作聪明”地帮我们设置了一个根本不存在的路径。如果设置不对，非常影响我们的情绪。**







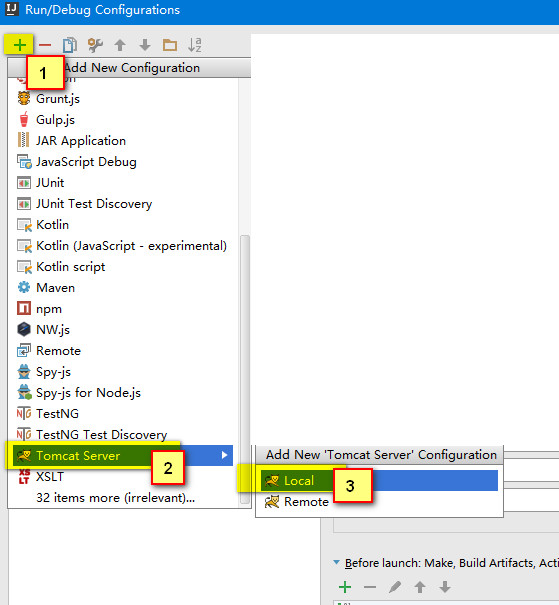


## 第 3 部分：配置本地关联的 tomcat 服务器位置

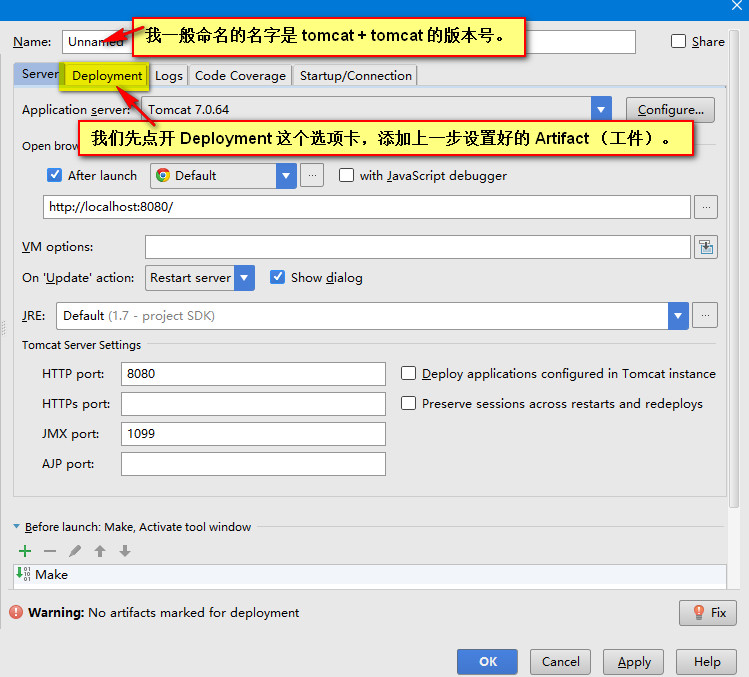
### 1、依次点击 Run -> Edit Configurations…

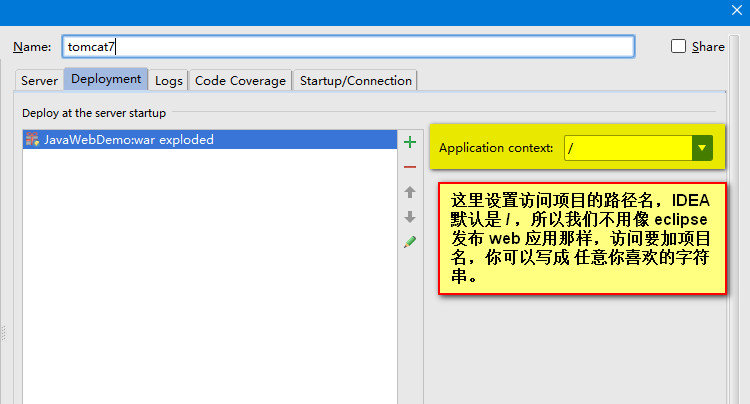


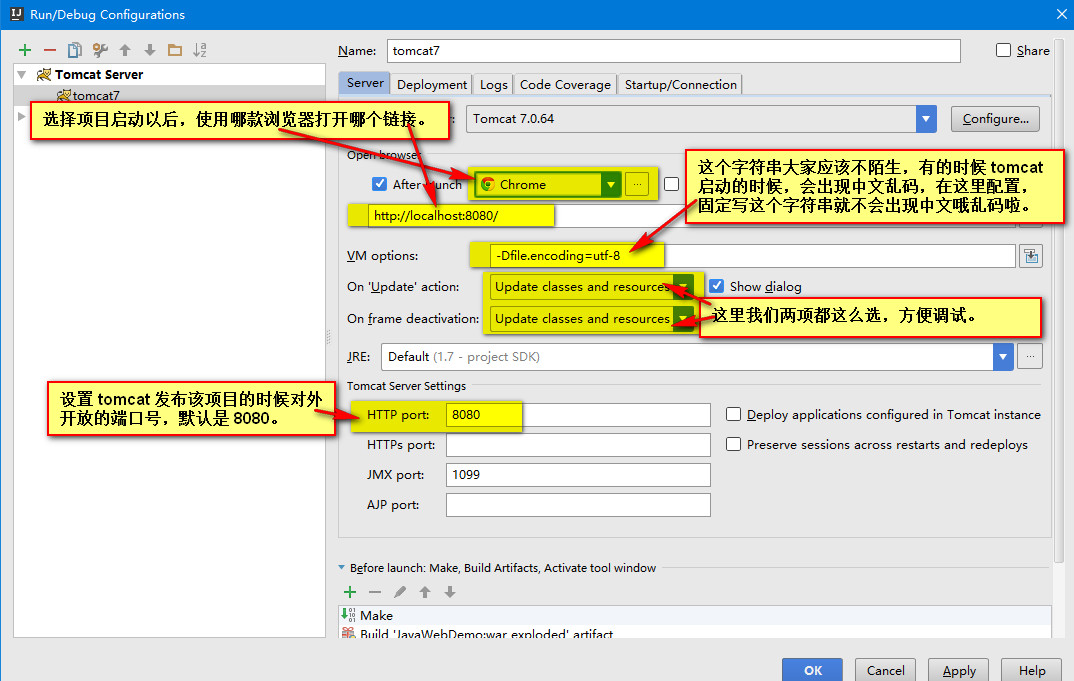
### 2、添加 Tomcat Server -> Local

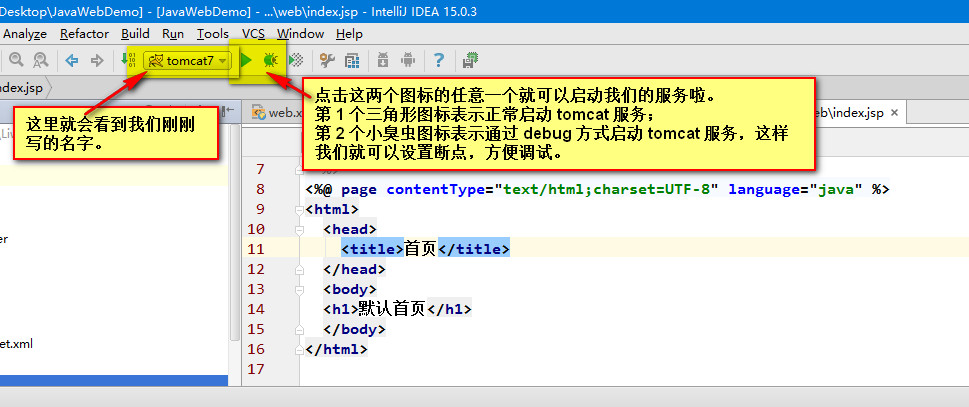


### 3、添加 deployment （前提是在第 2 部分的各种配置里要把 Artifact 添加上）

还回忆得起是怎么添加 Artifact 的吗，选择第 2 个的第 2 个。   


我们还可以设置访问路径：   


配置启动以后默认打开的链接、浏览器、端口号、tomcat 控制台显示编码等（这个界面很重要）。   


最后，我们启动配置好的服务器，大功告成了！   


### Idea设置编码和内存：

|  |
| --- |
| -server  -XX:PermSize=128M  -XX:MaxPermSize=256m  -Dfile.encoding=UTF-8 |

