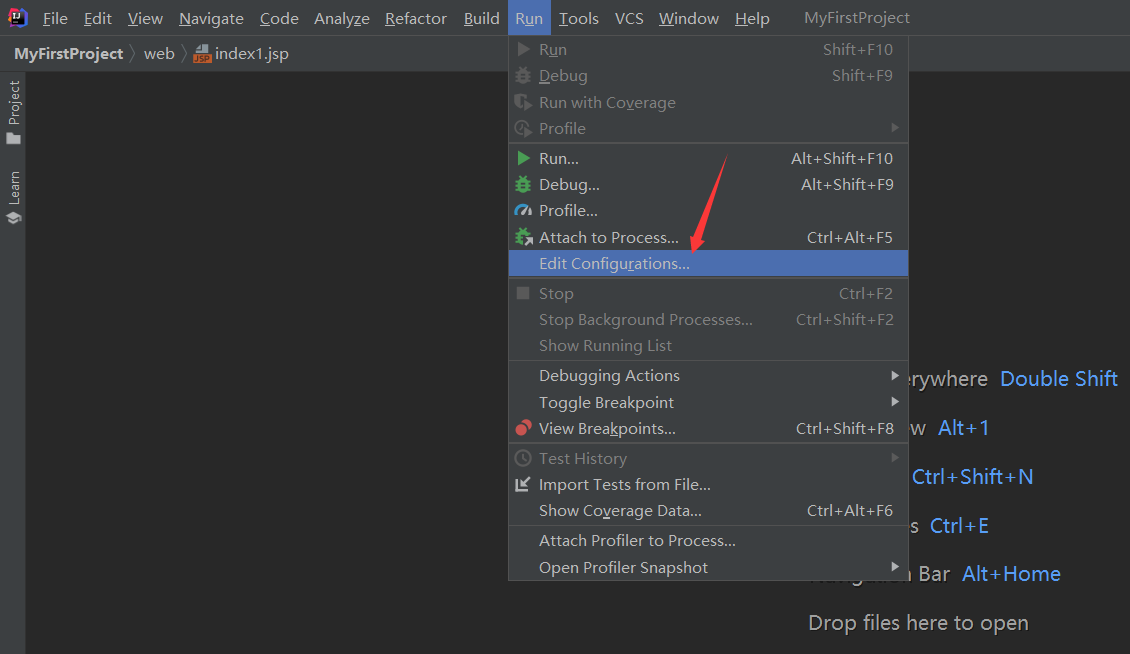
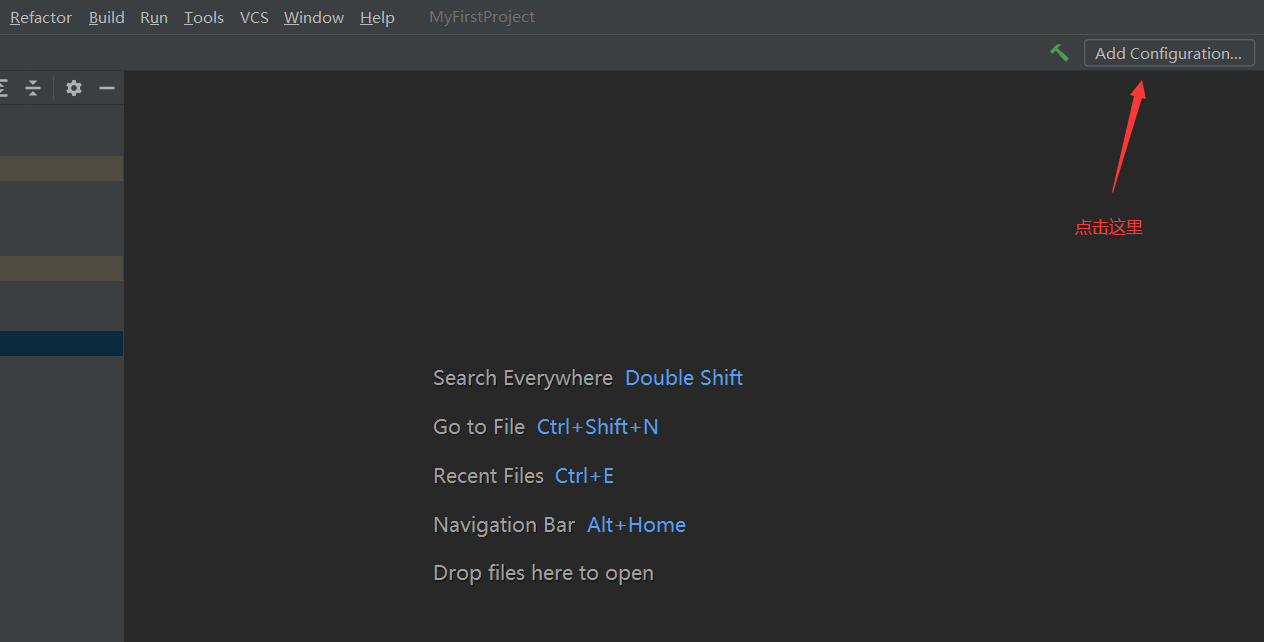
**在IDEA2021中配置tomcat服务器,再将web项目部署到tomcat服务器上**

**在IEDA2021中配置tomcat**

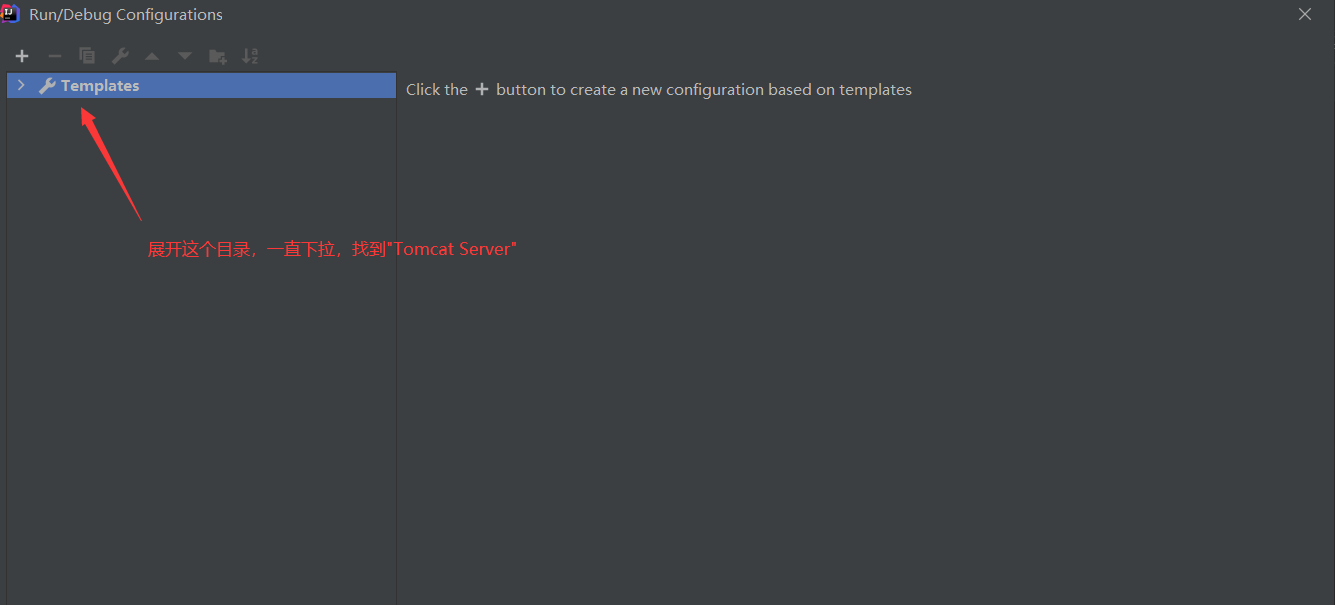
第一步:

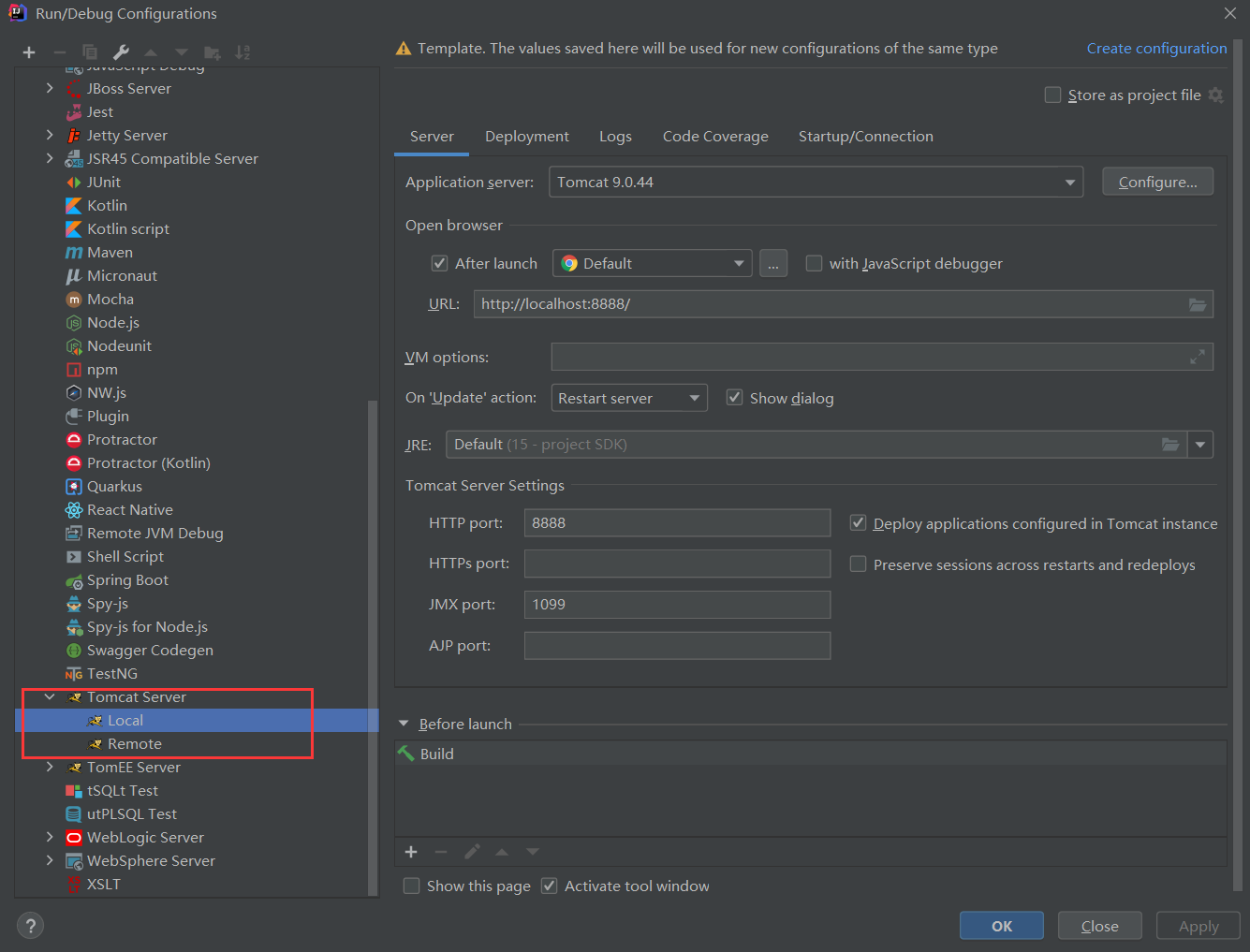


或者

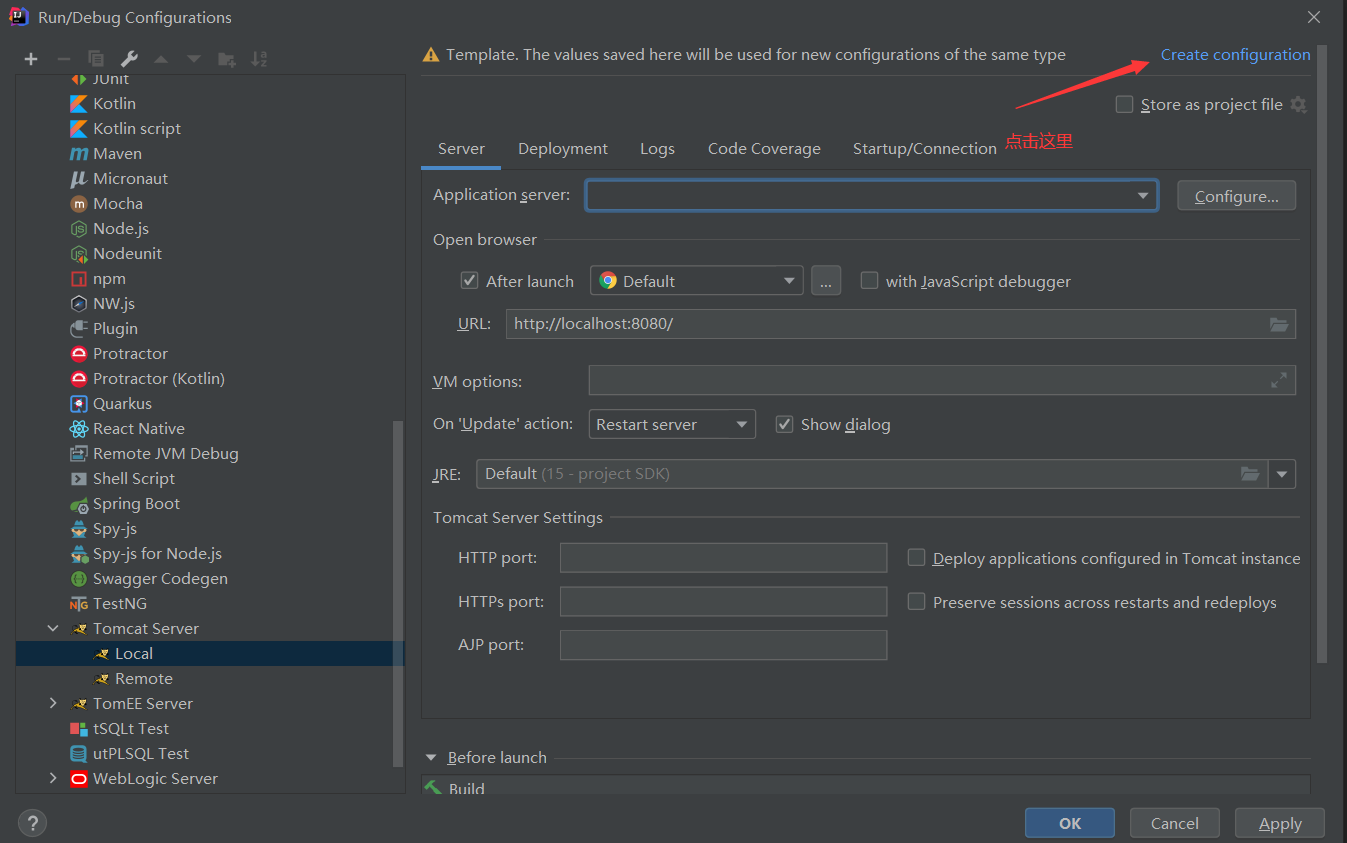


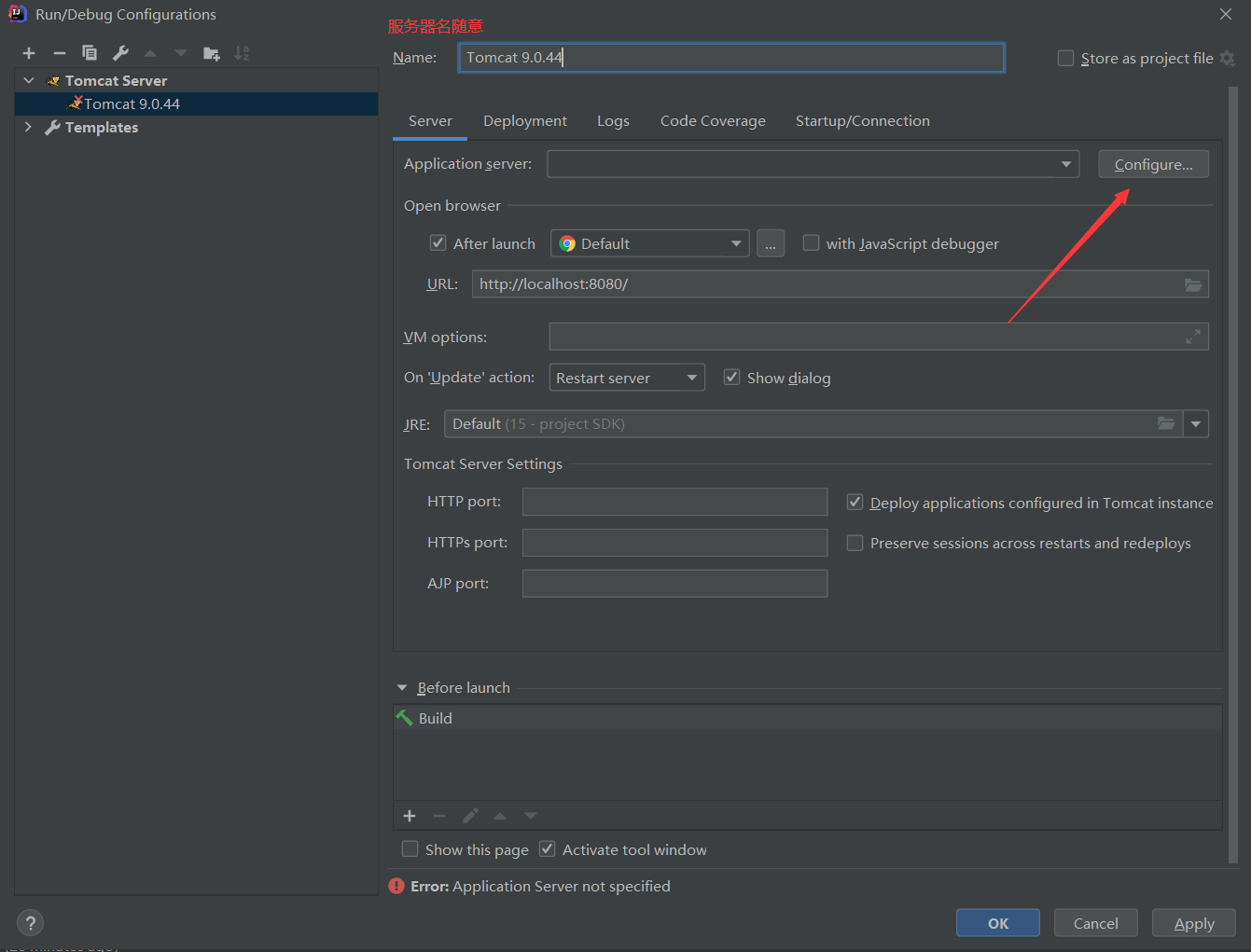
第二步:

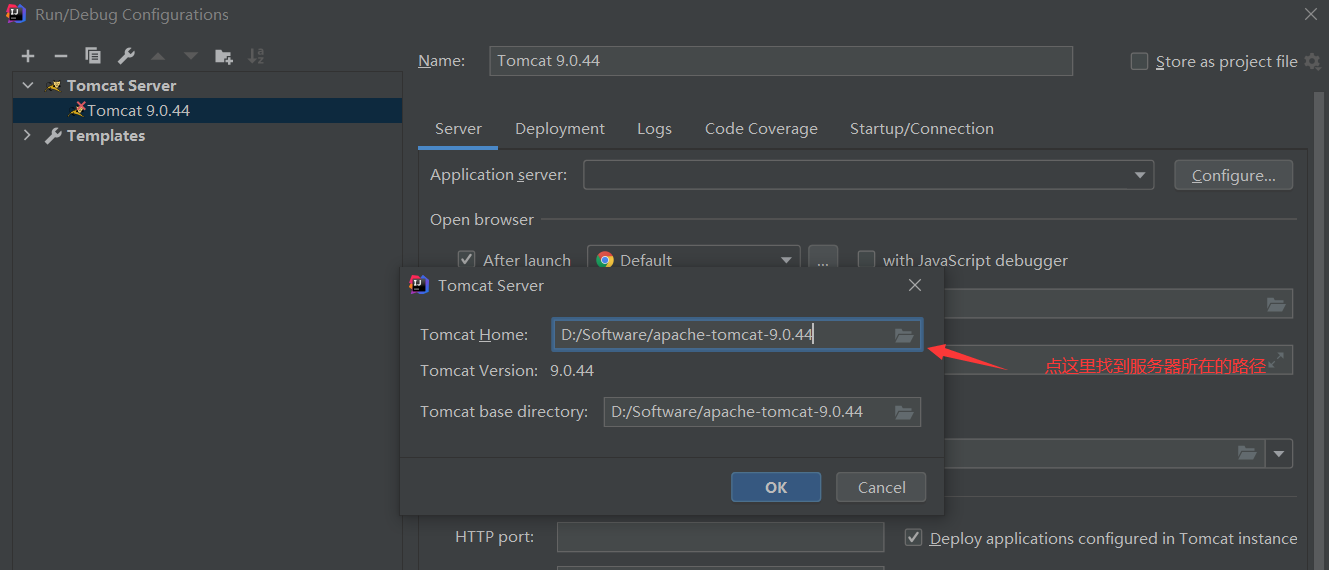


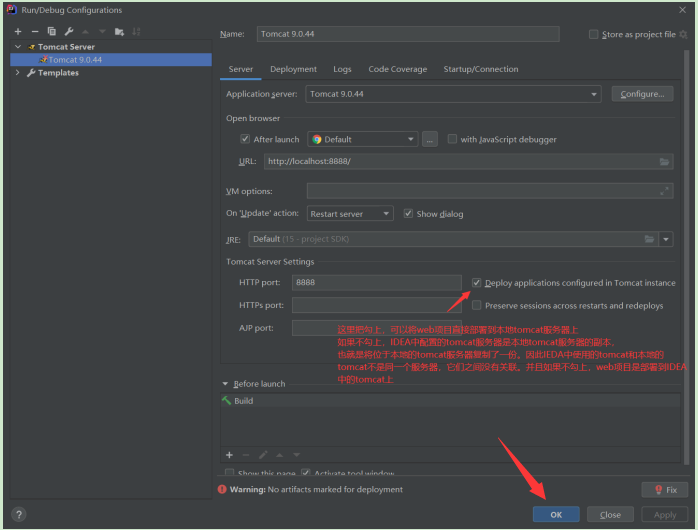


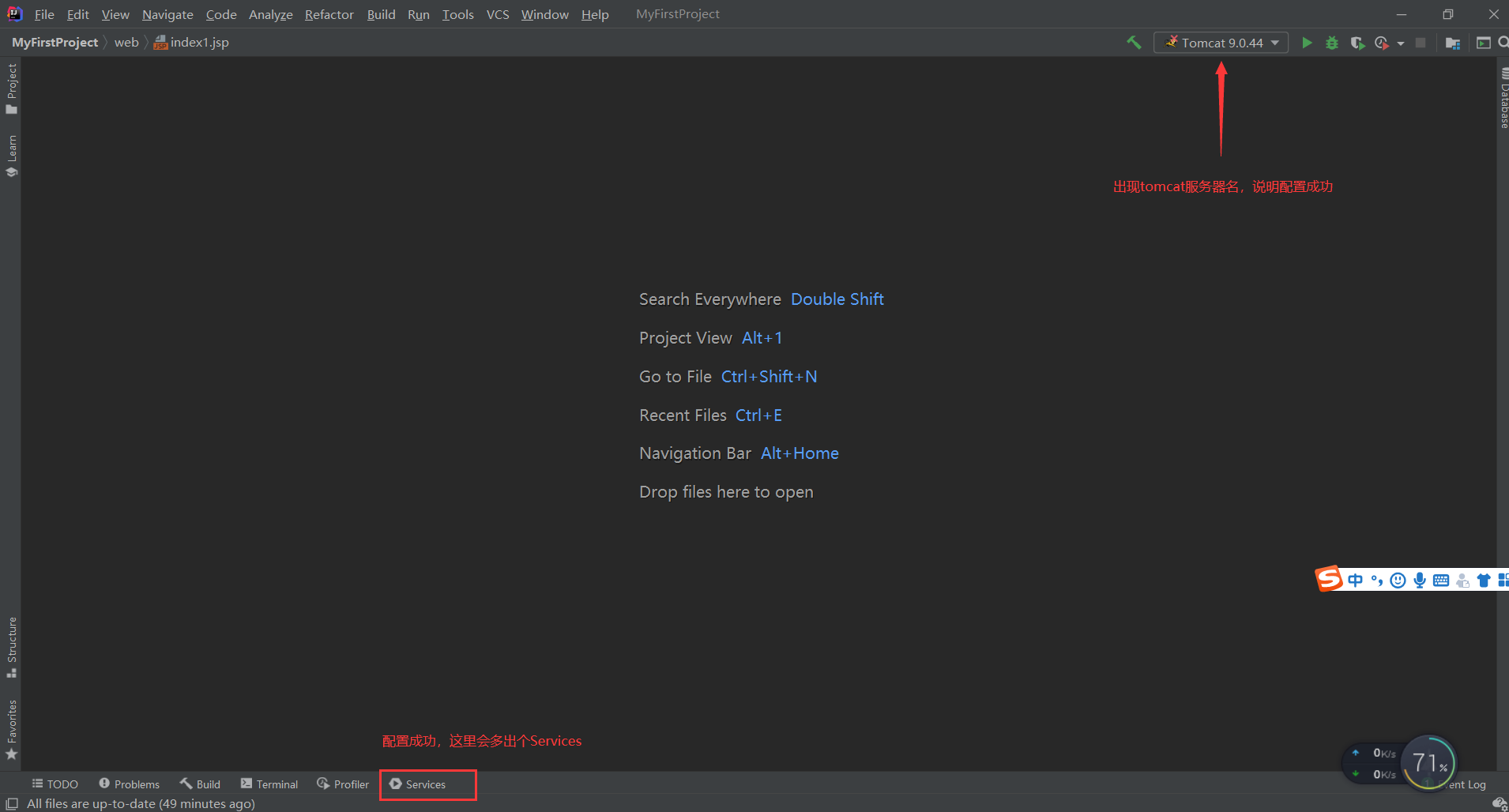
第三步:



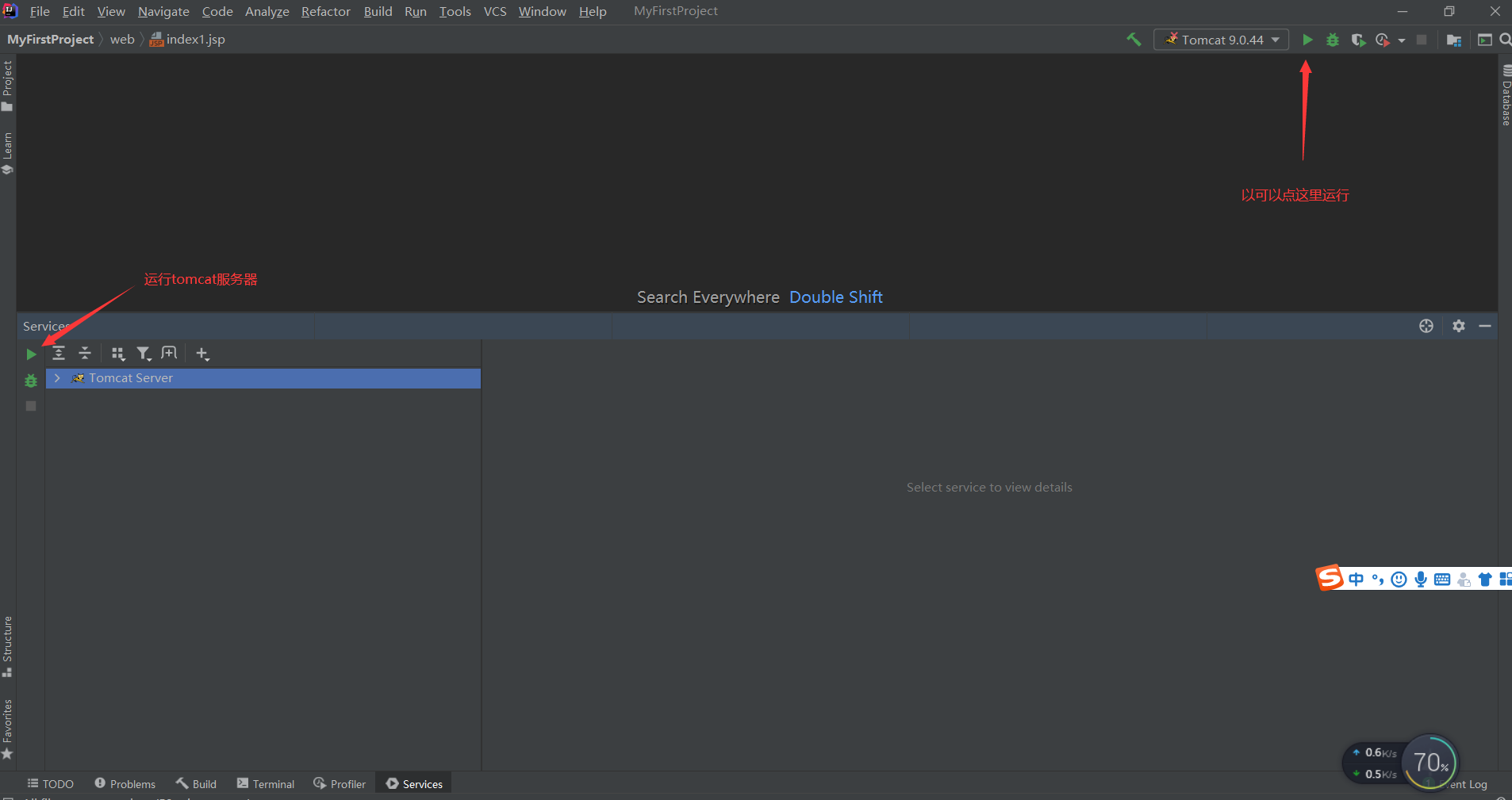


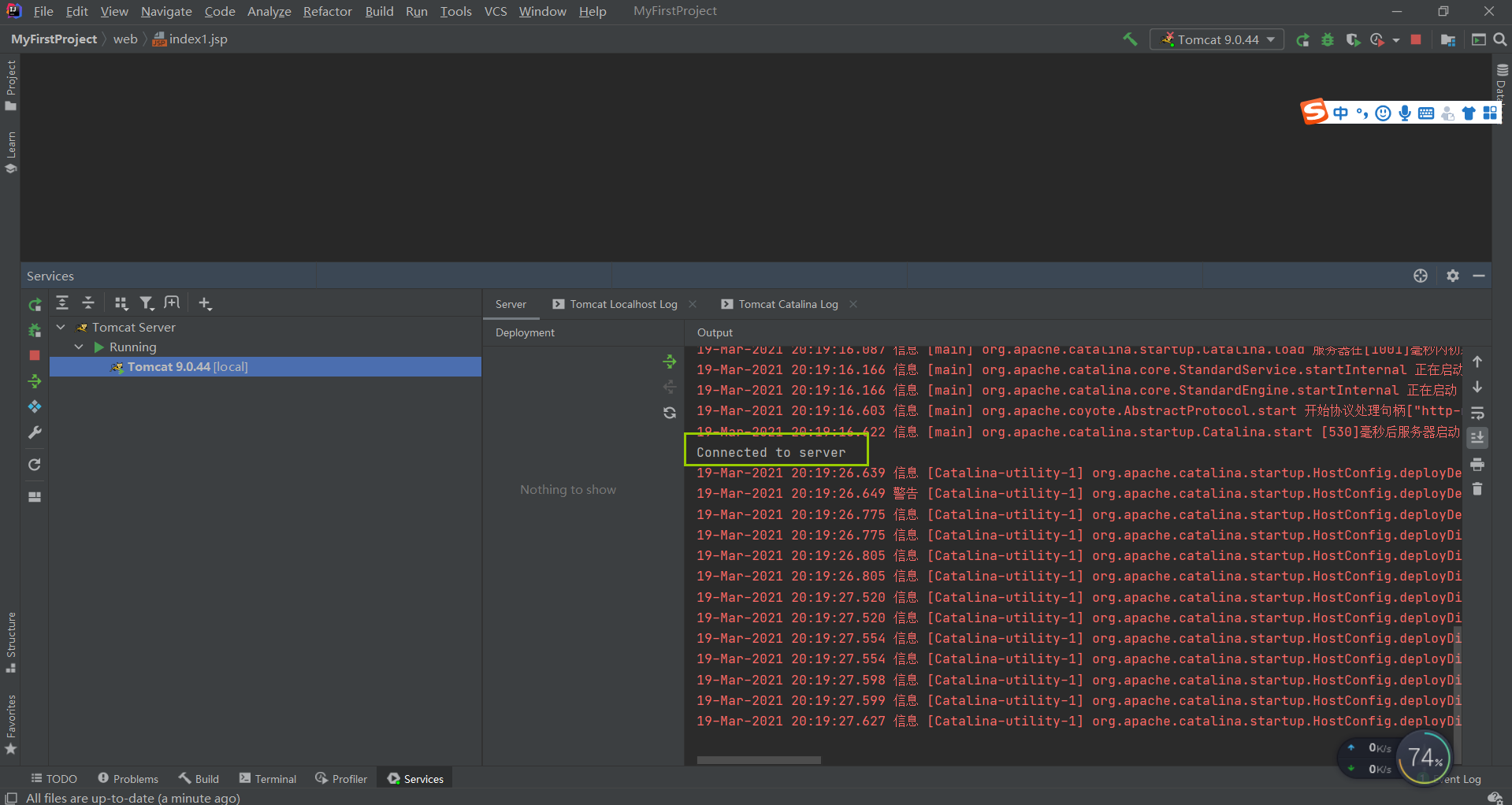






测试在IDEA中能否成功运行刚配置好的tomcat服务器

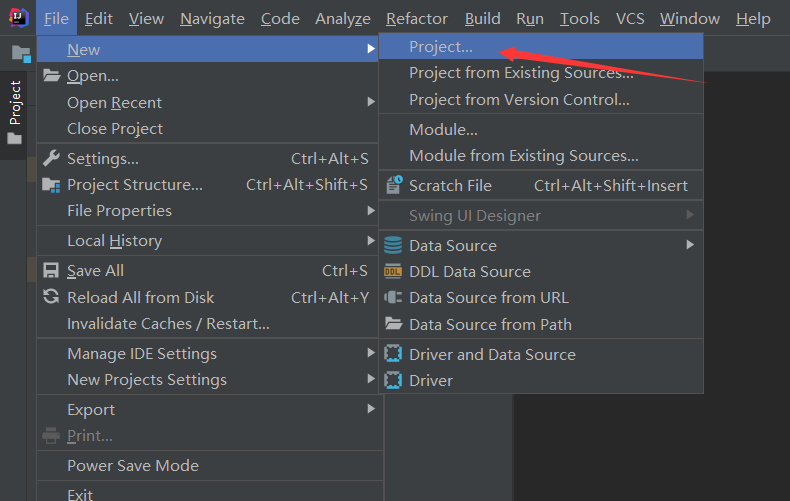


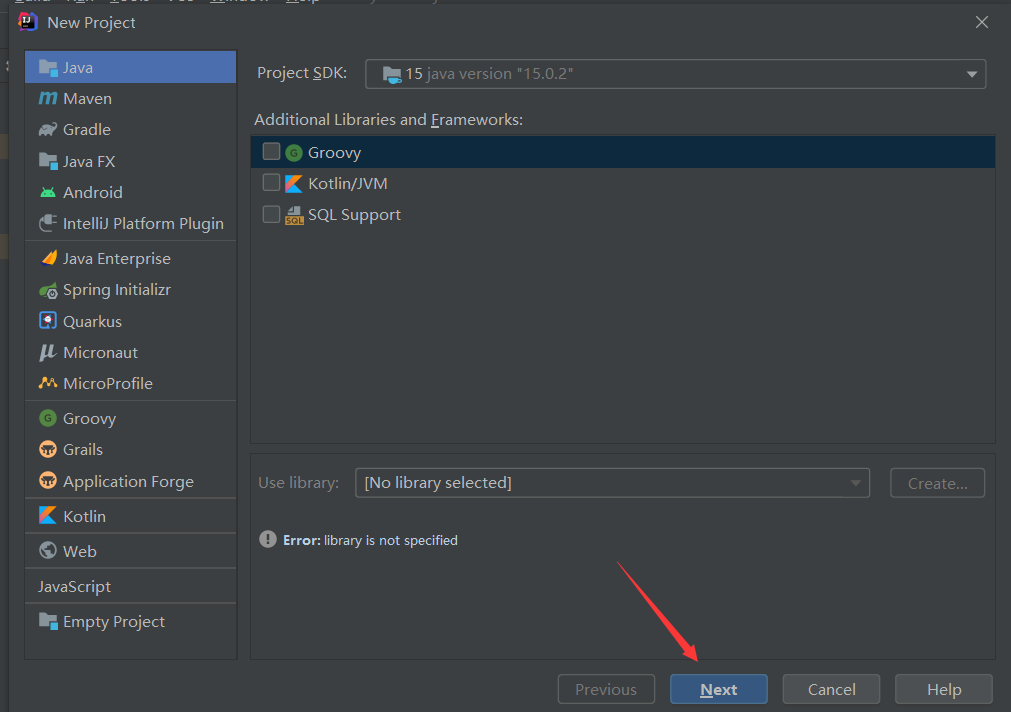


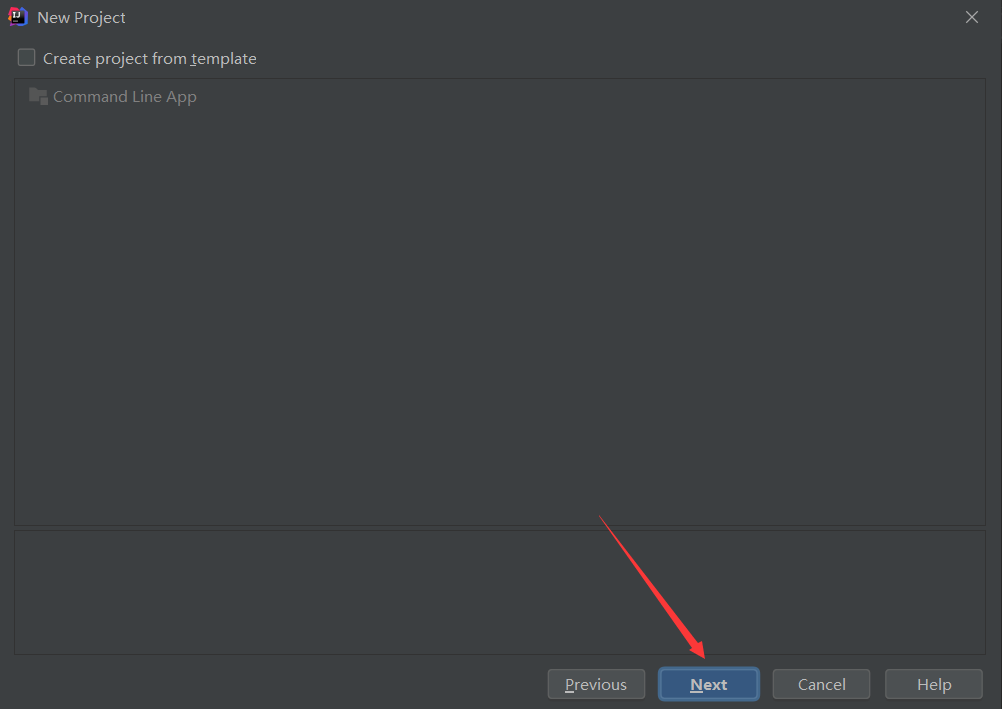
出现Connected to server，说明成功运行tonmcat。（这里并未部署web项目到tomcat中）

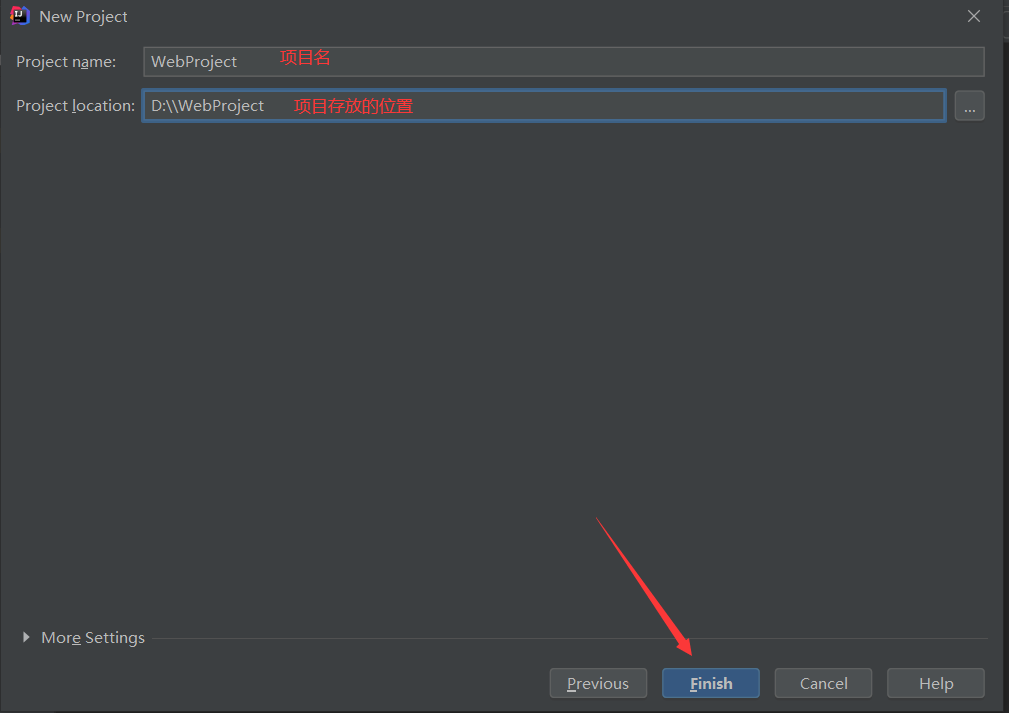
**创建web项目，并部署到tomcat中**

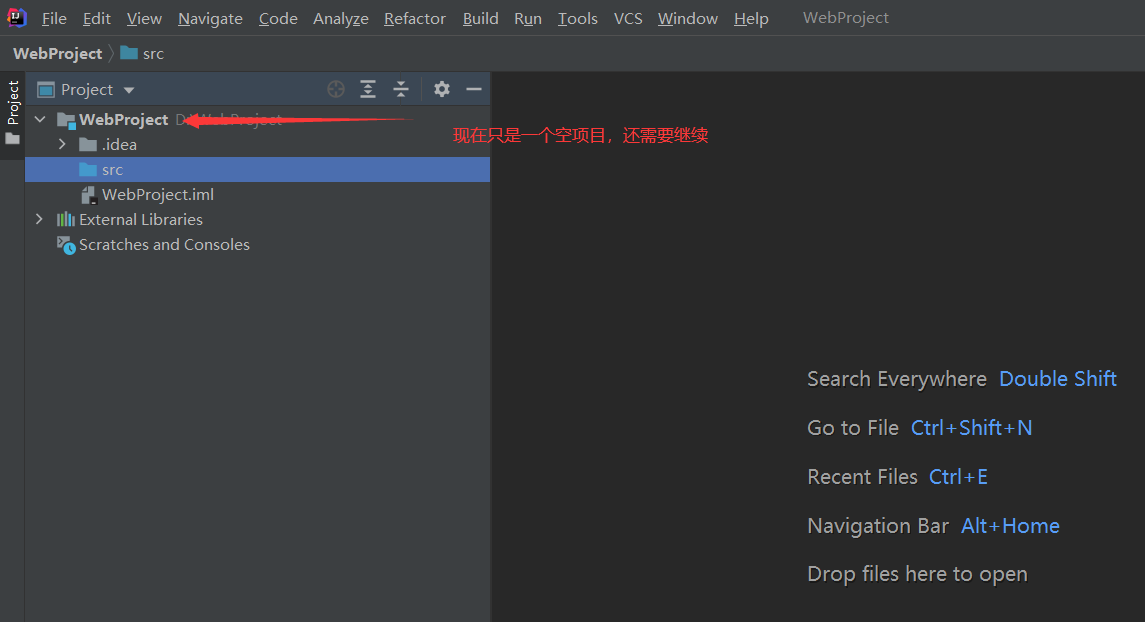
第一步: 先创建web项目

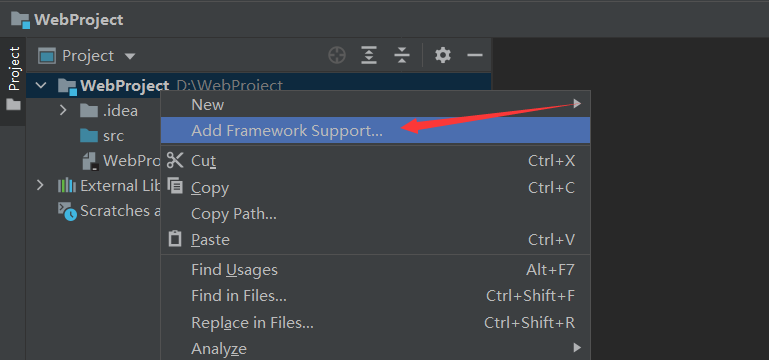




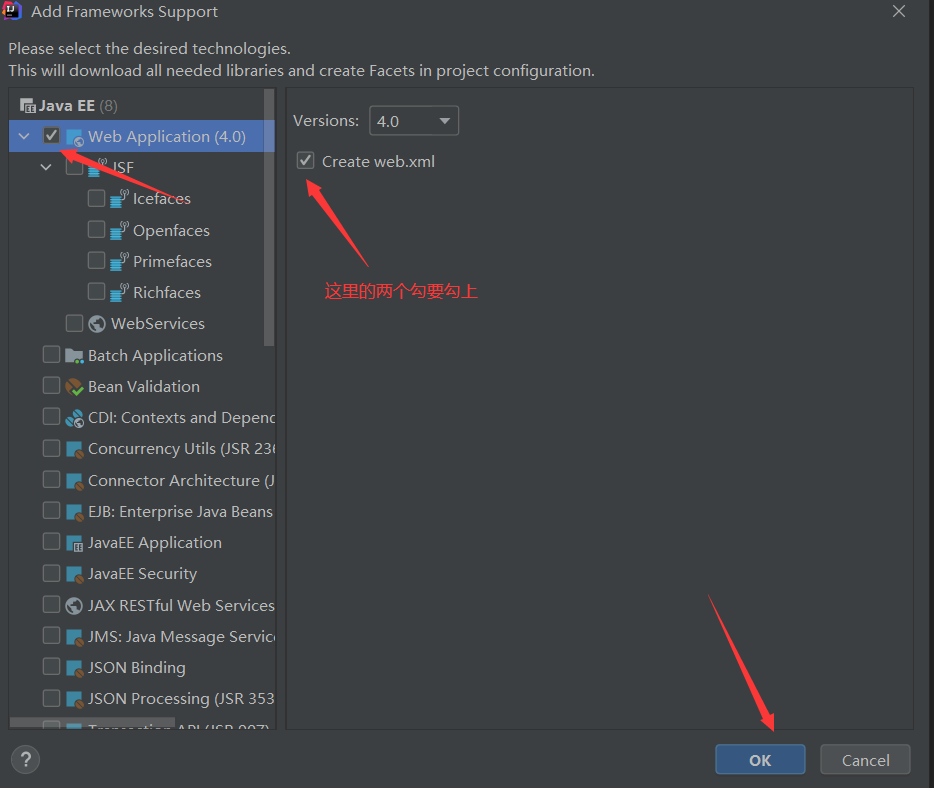


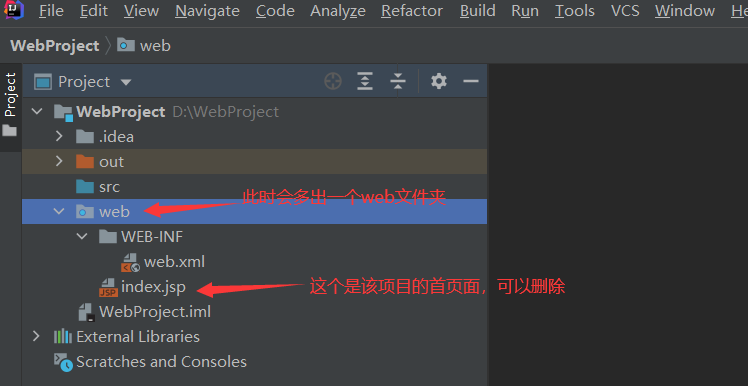






右键项目点击Add Framework Support

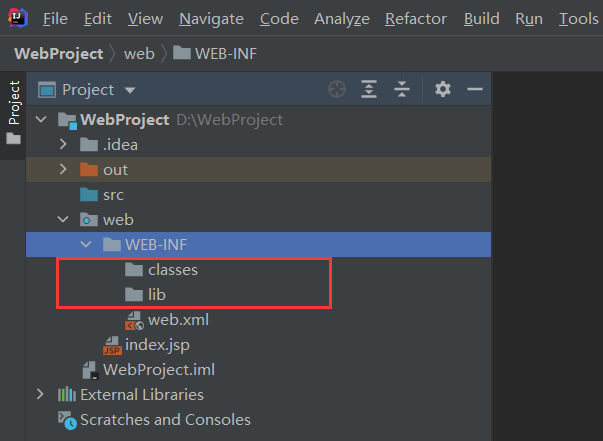




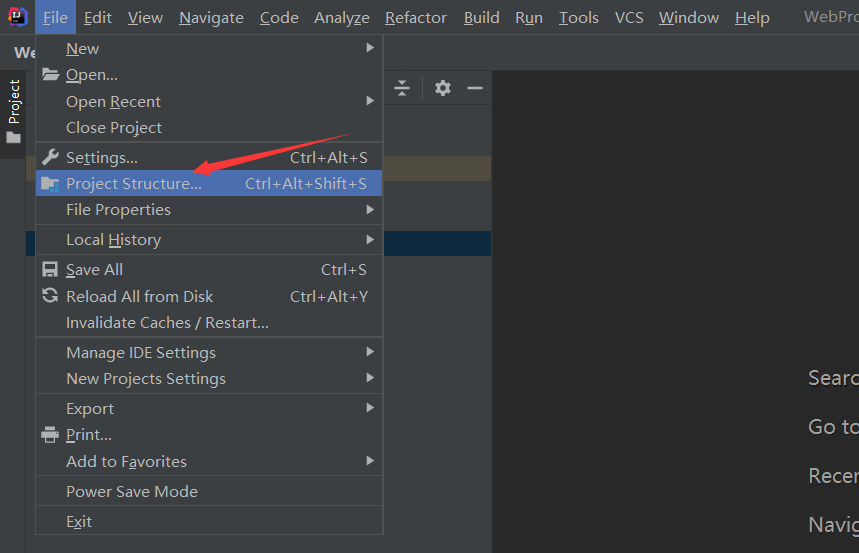
注: 到这里一个web项目就算创建完成，但最好在WEB-INF下新建两文件夹：classes和lib

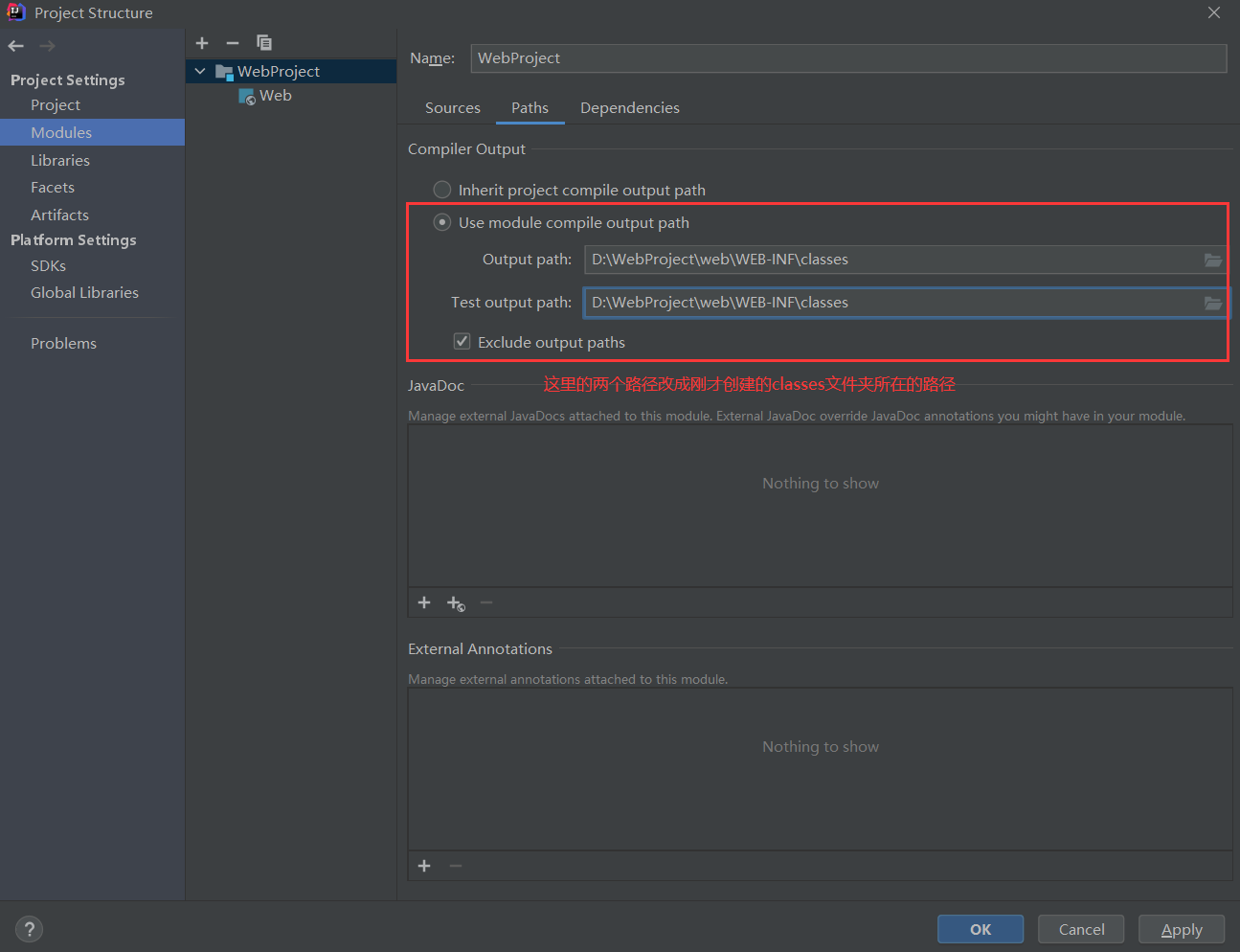
classes目录：放置该项目所产生的字节码文件

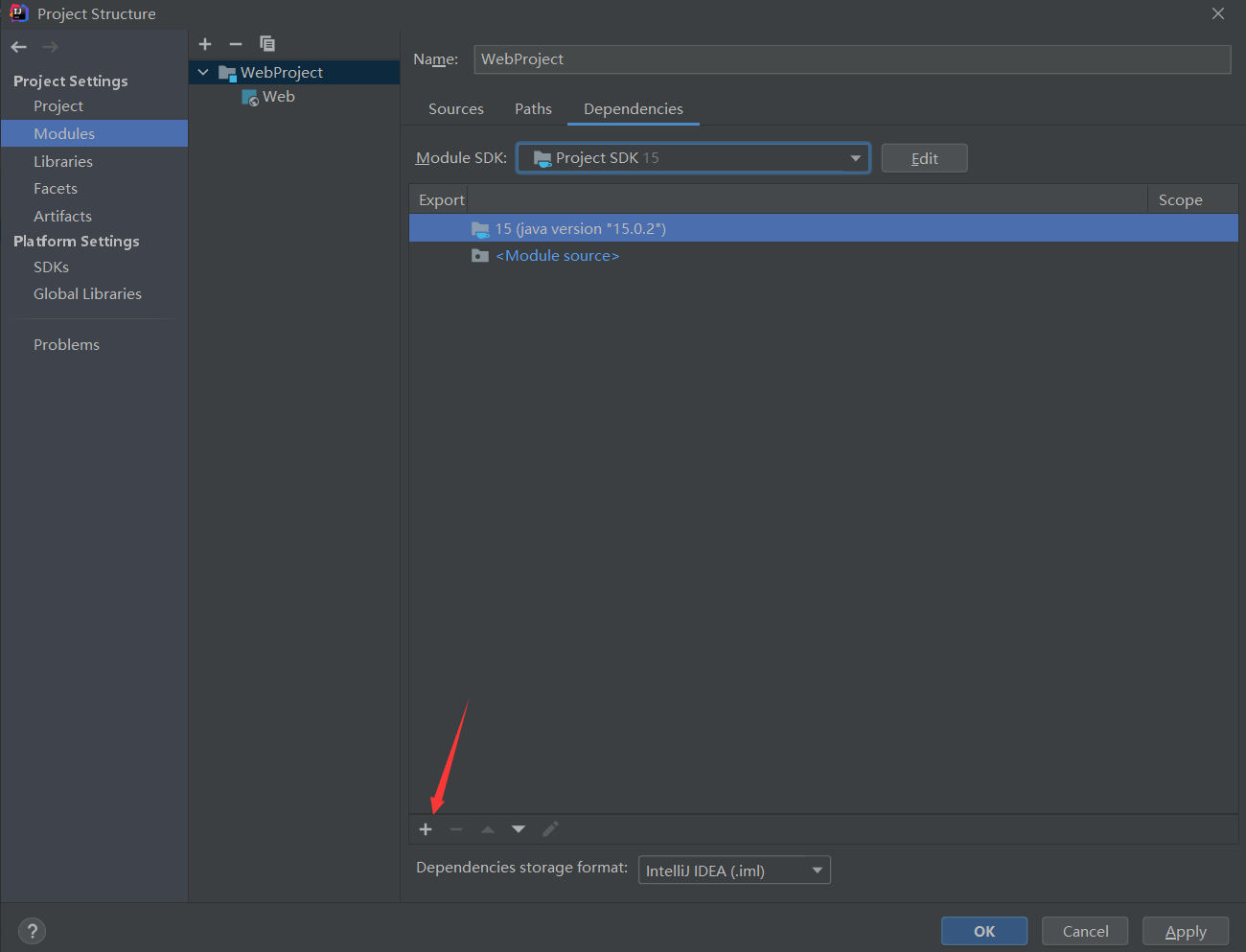
lib目录:：放置项目依赖的jar包

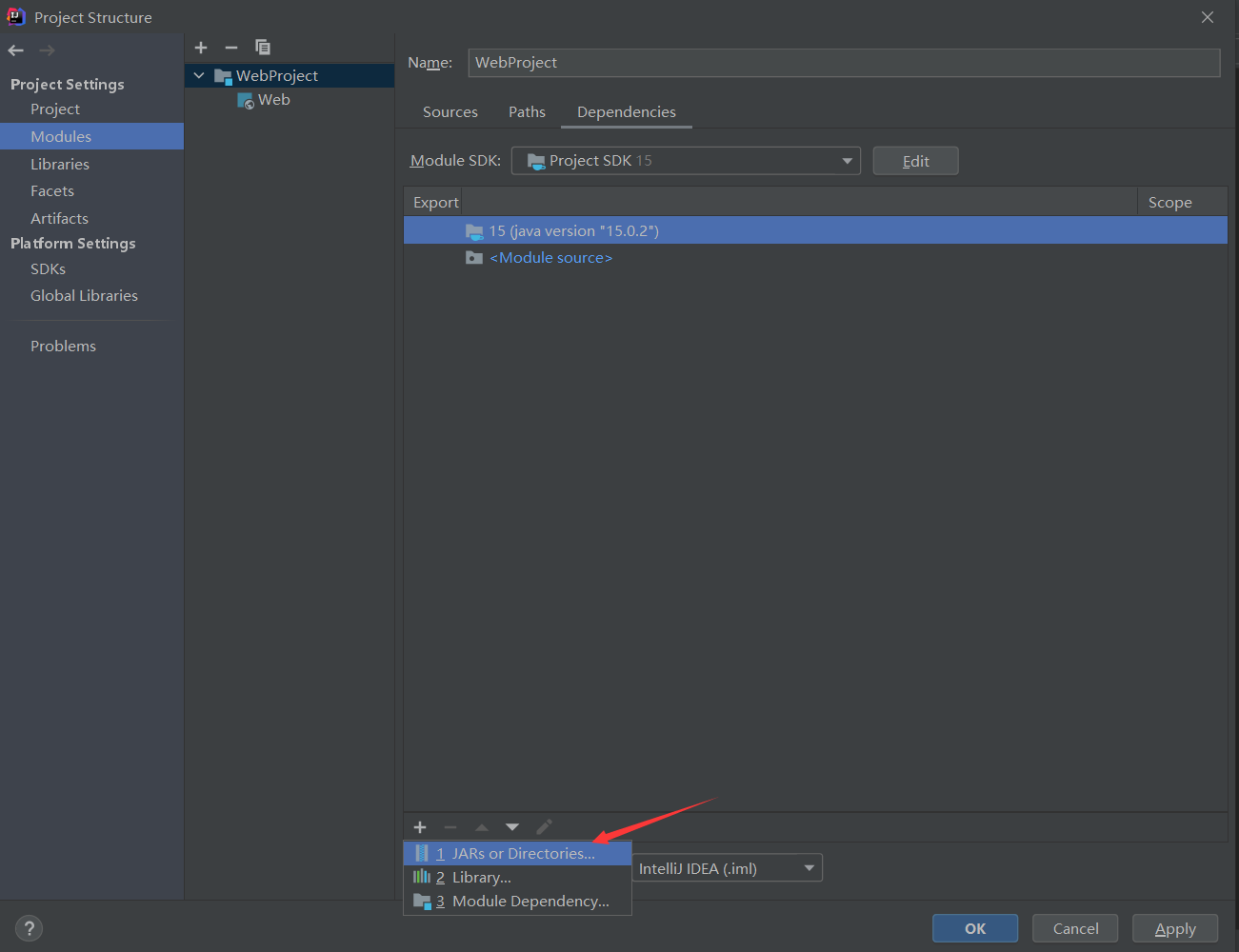


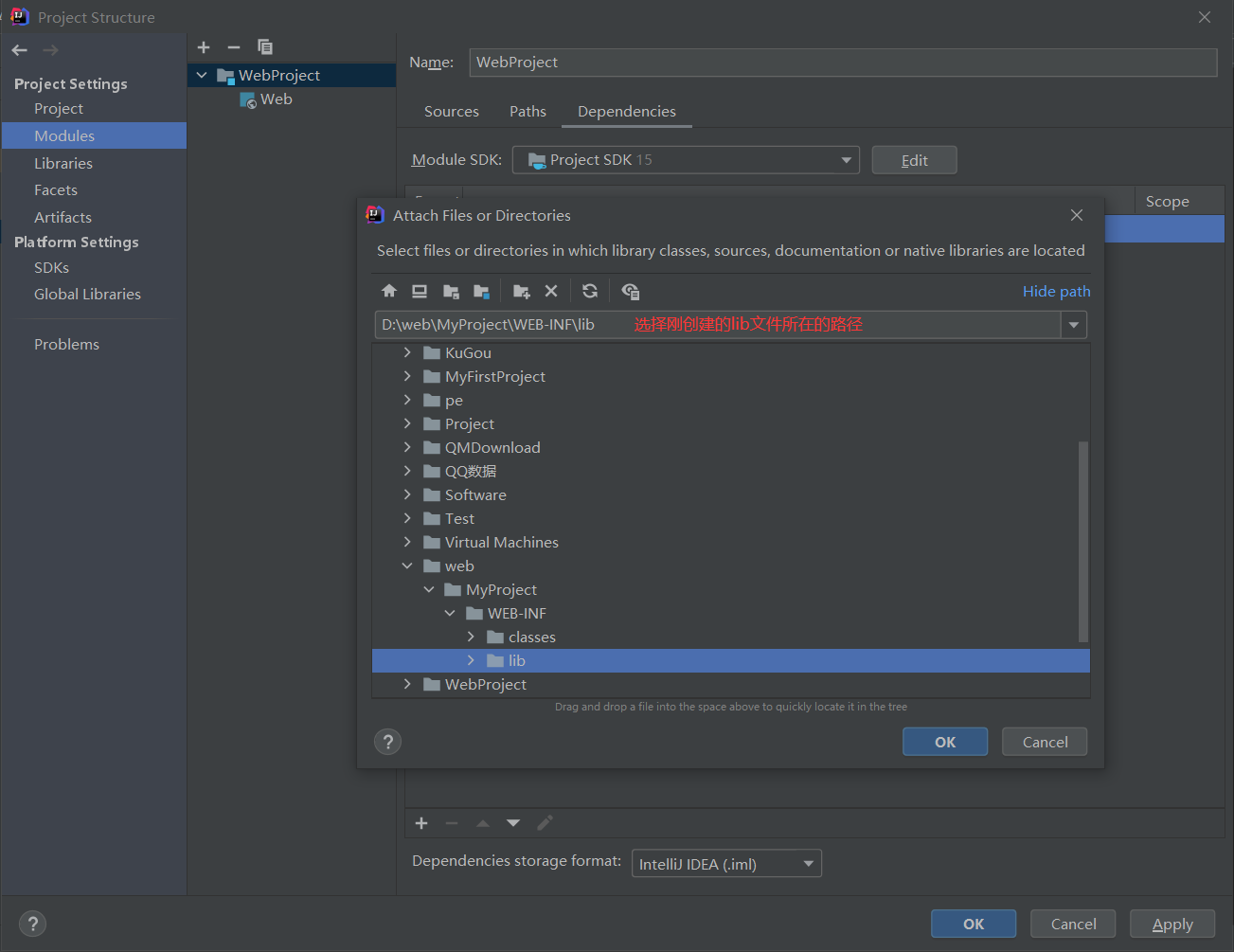
这两个文件创建好后，再配置下这两个文件

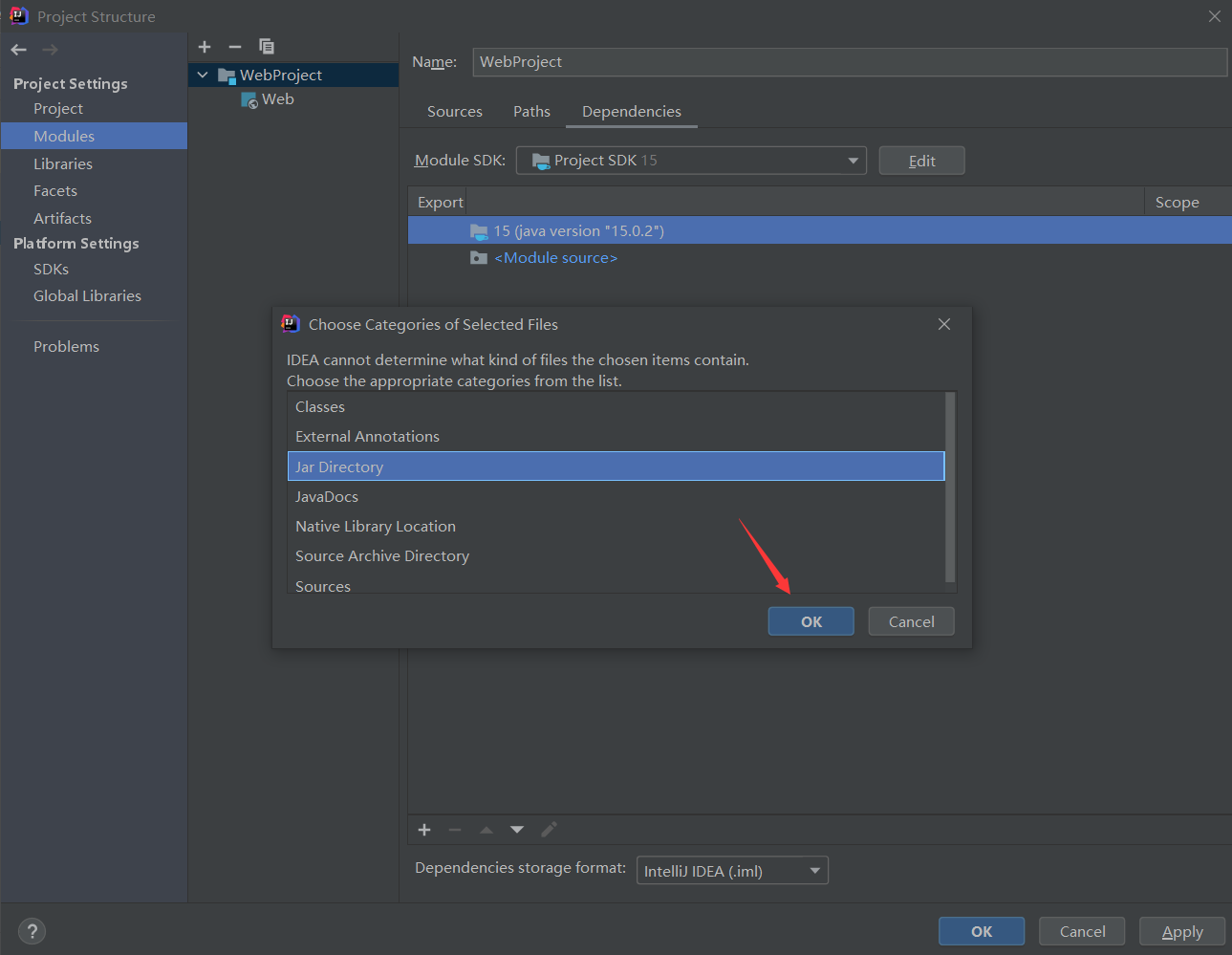


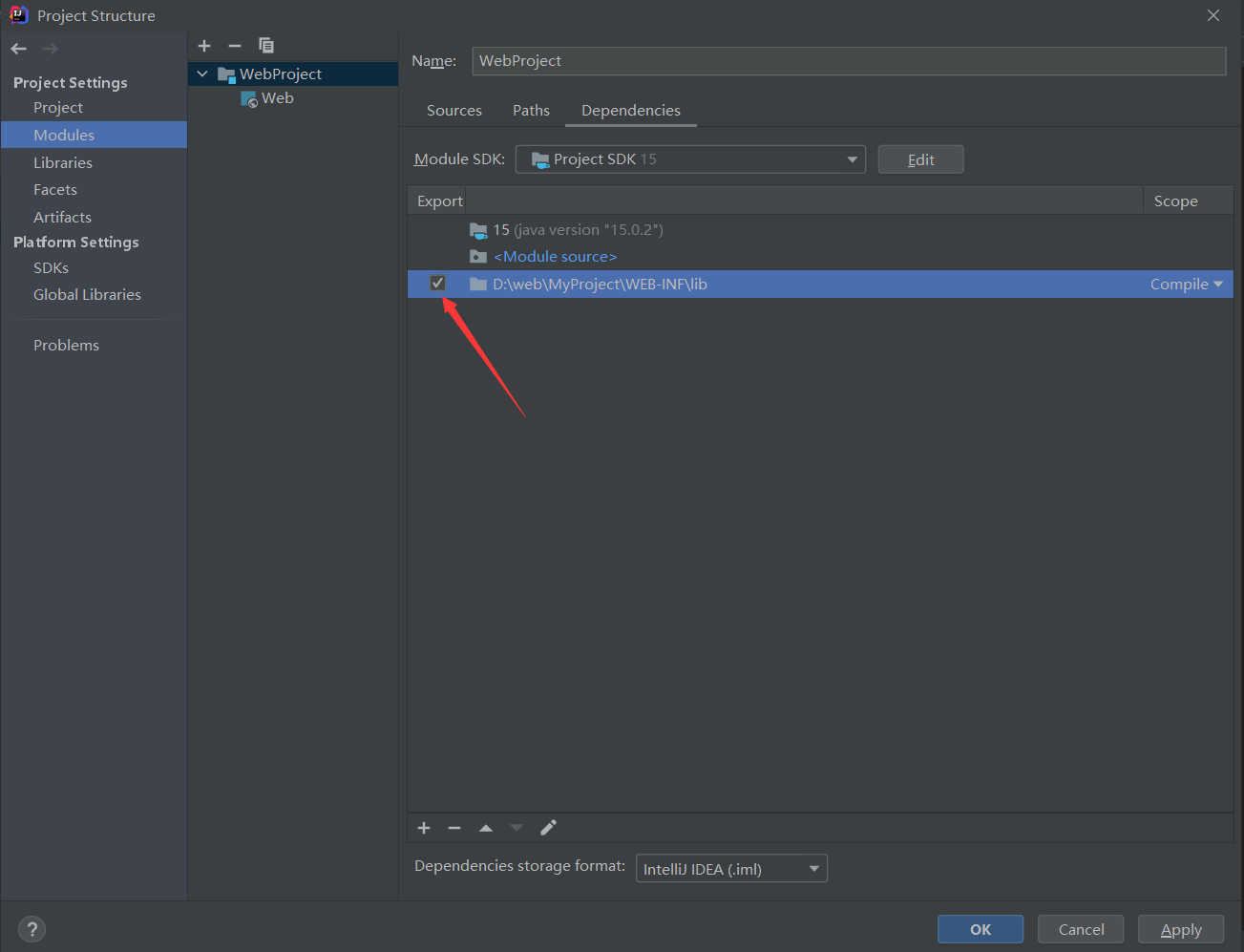


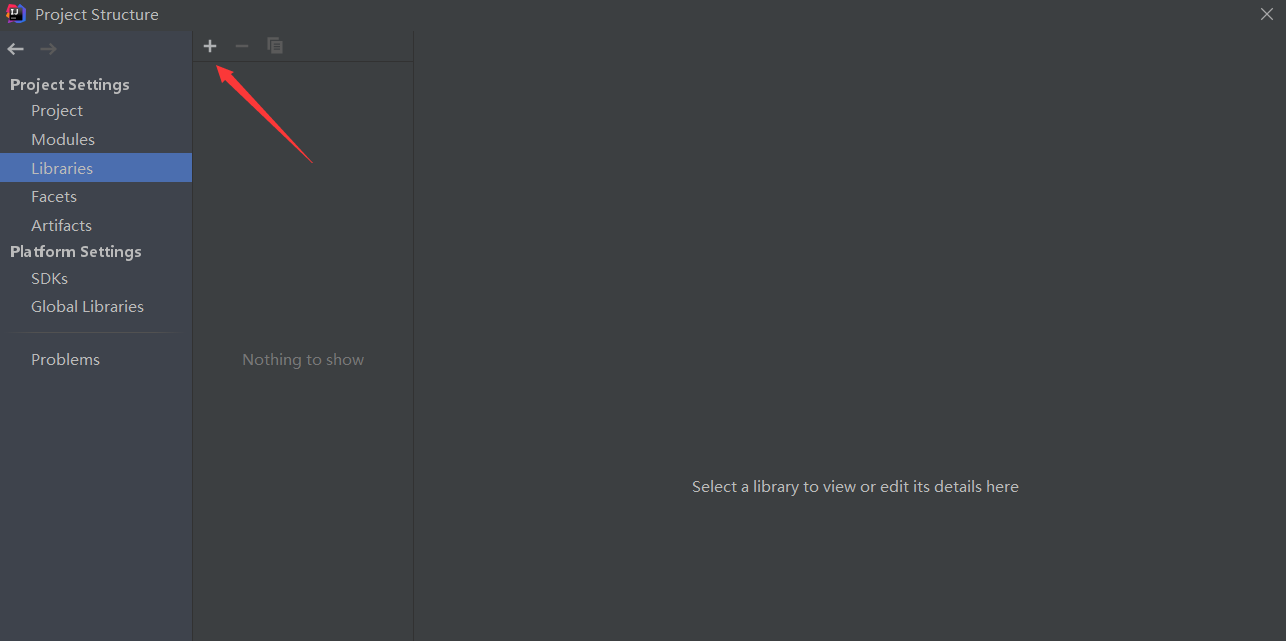


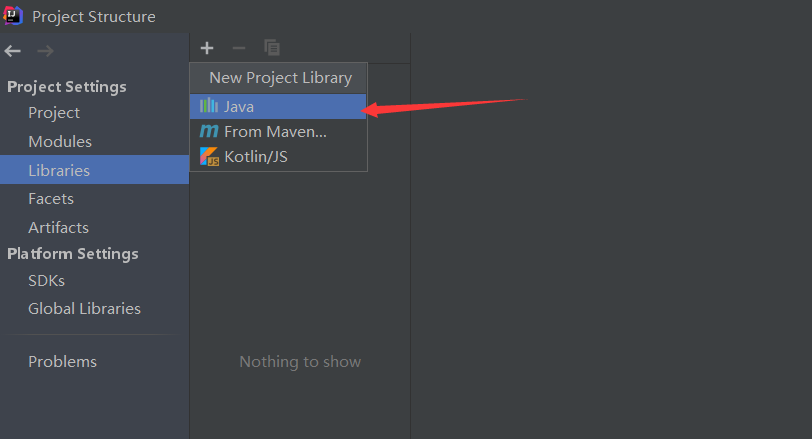


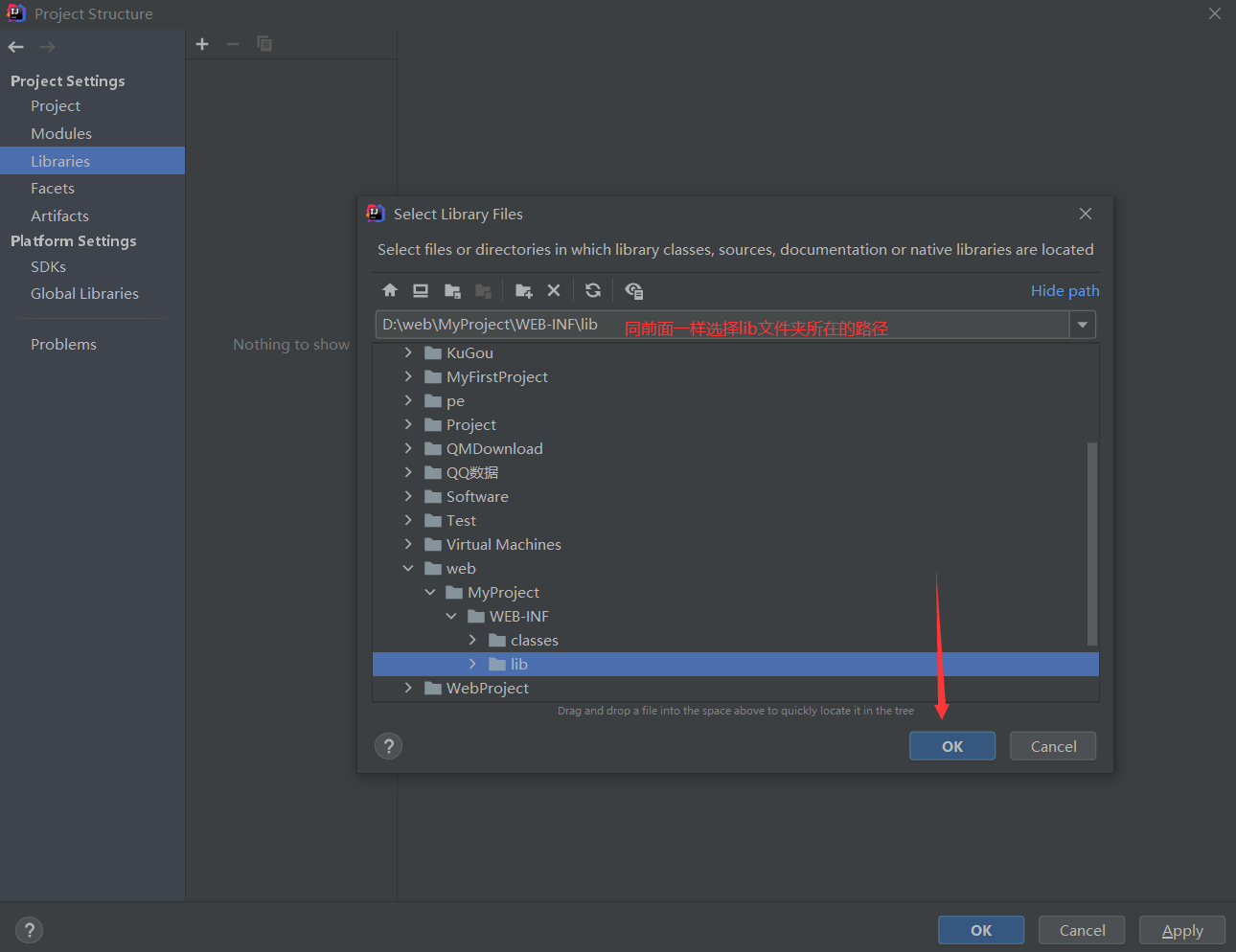


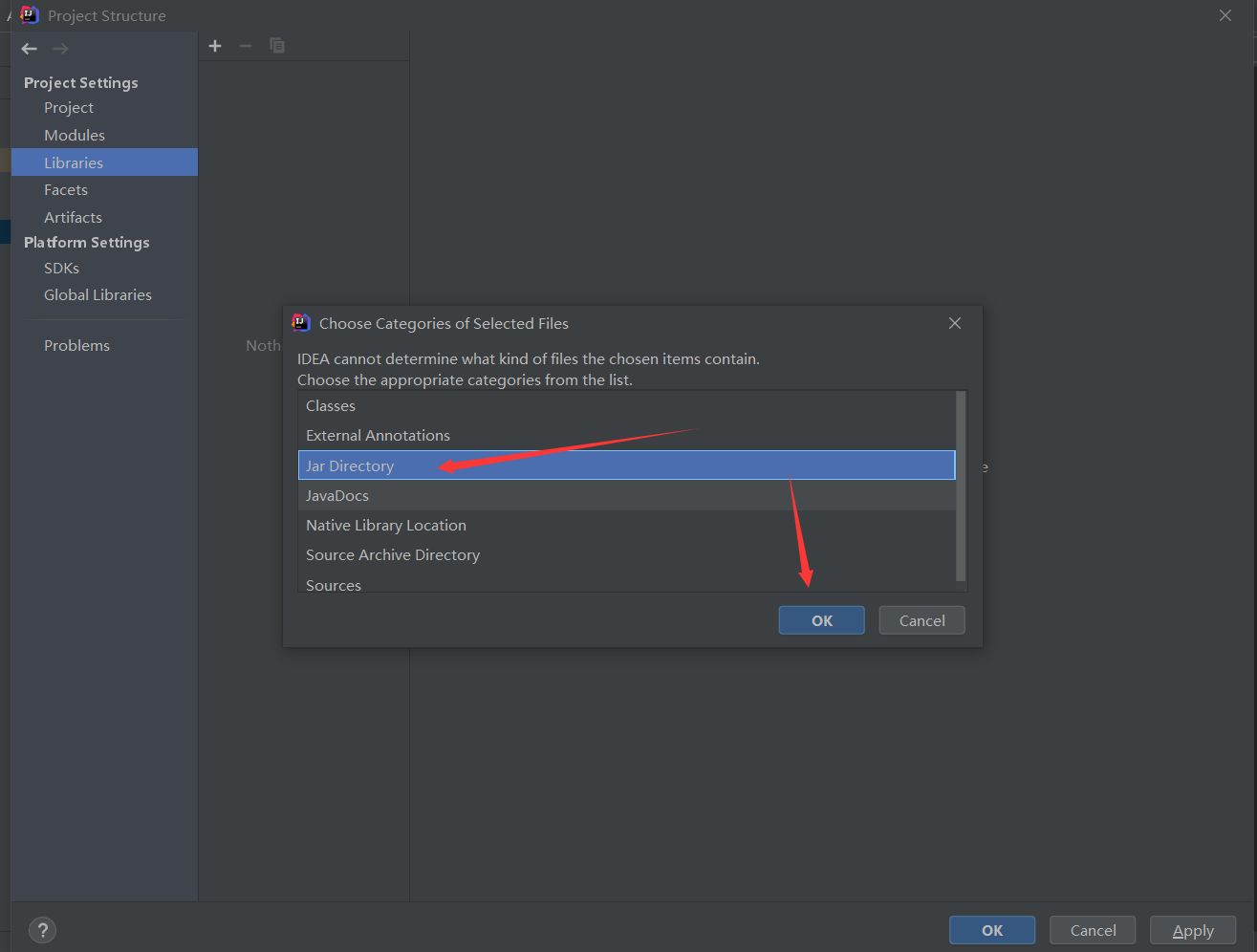


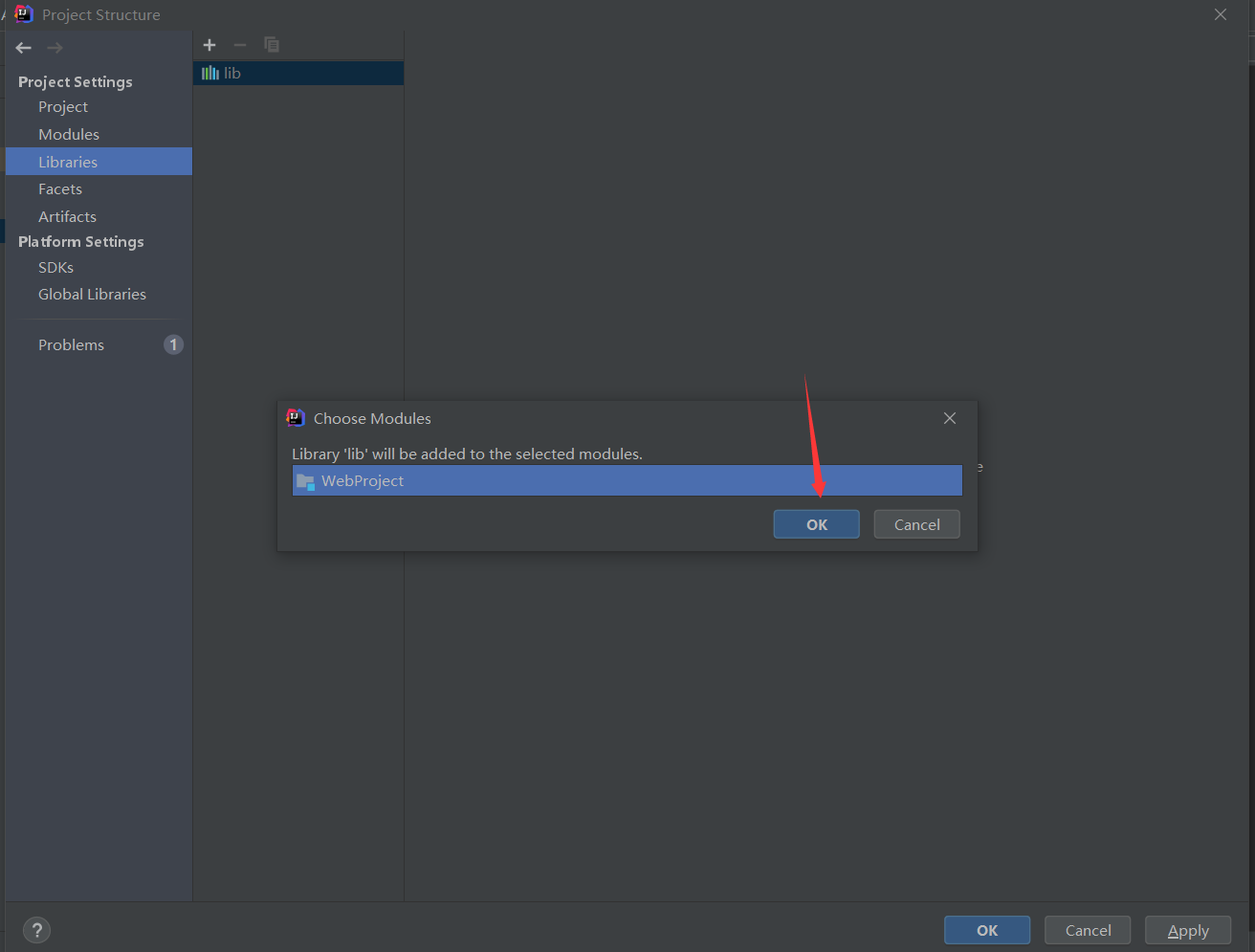


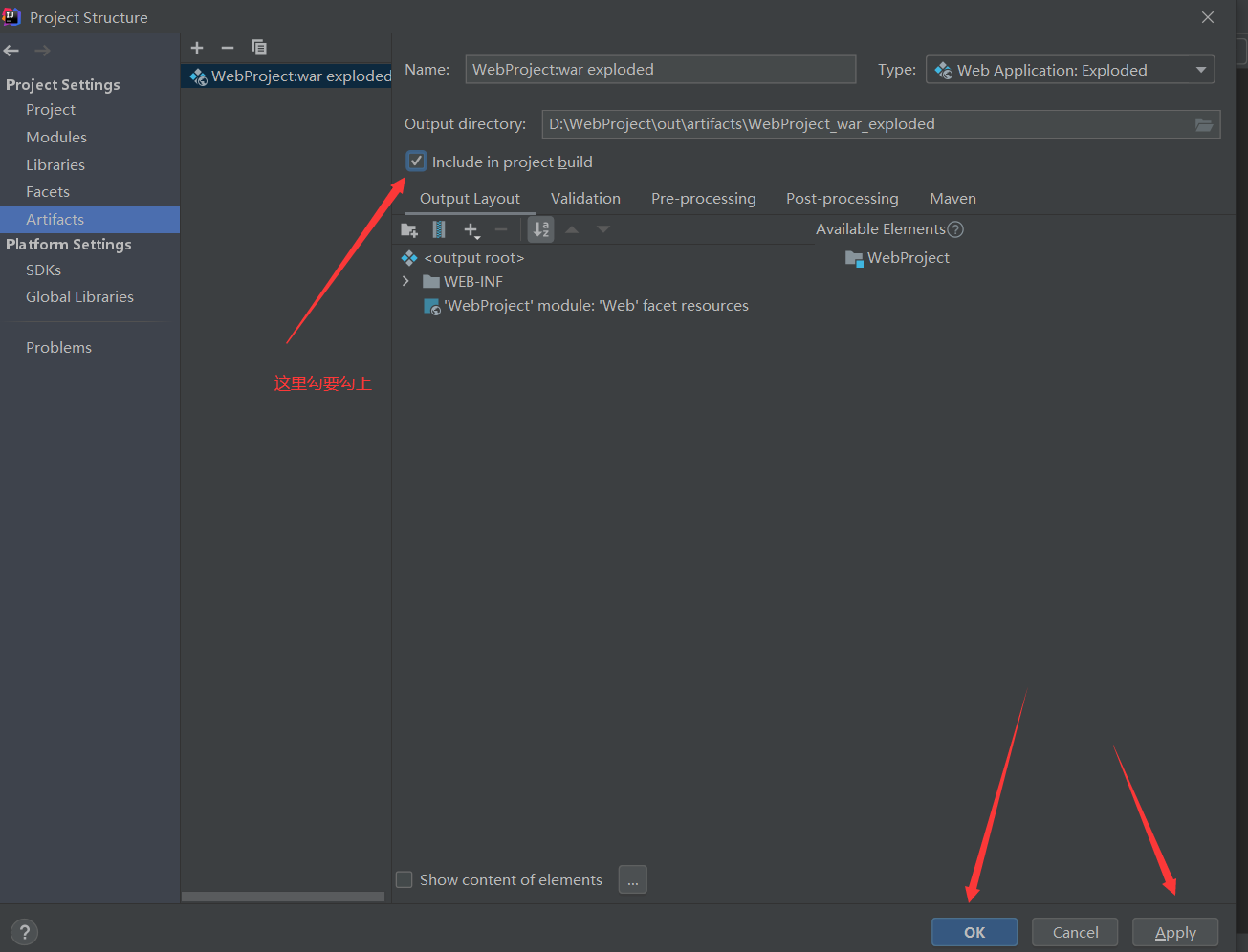






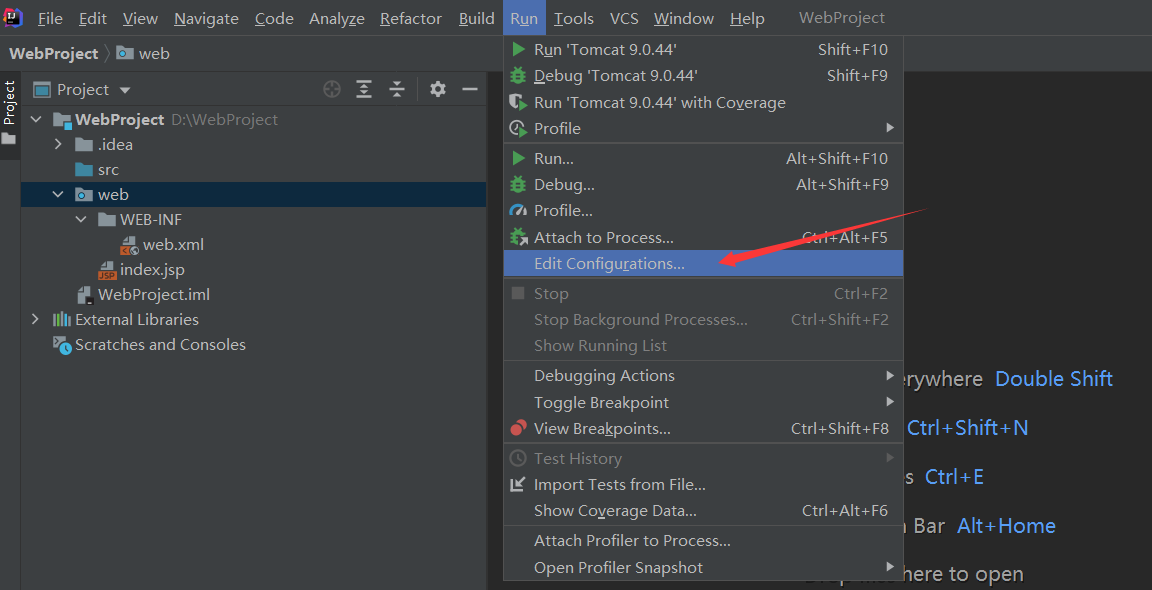


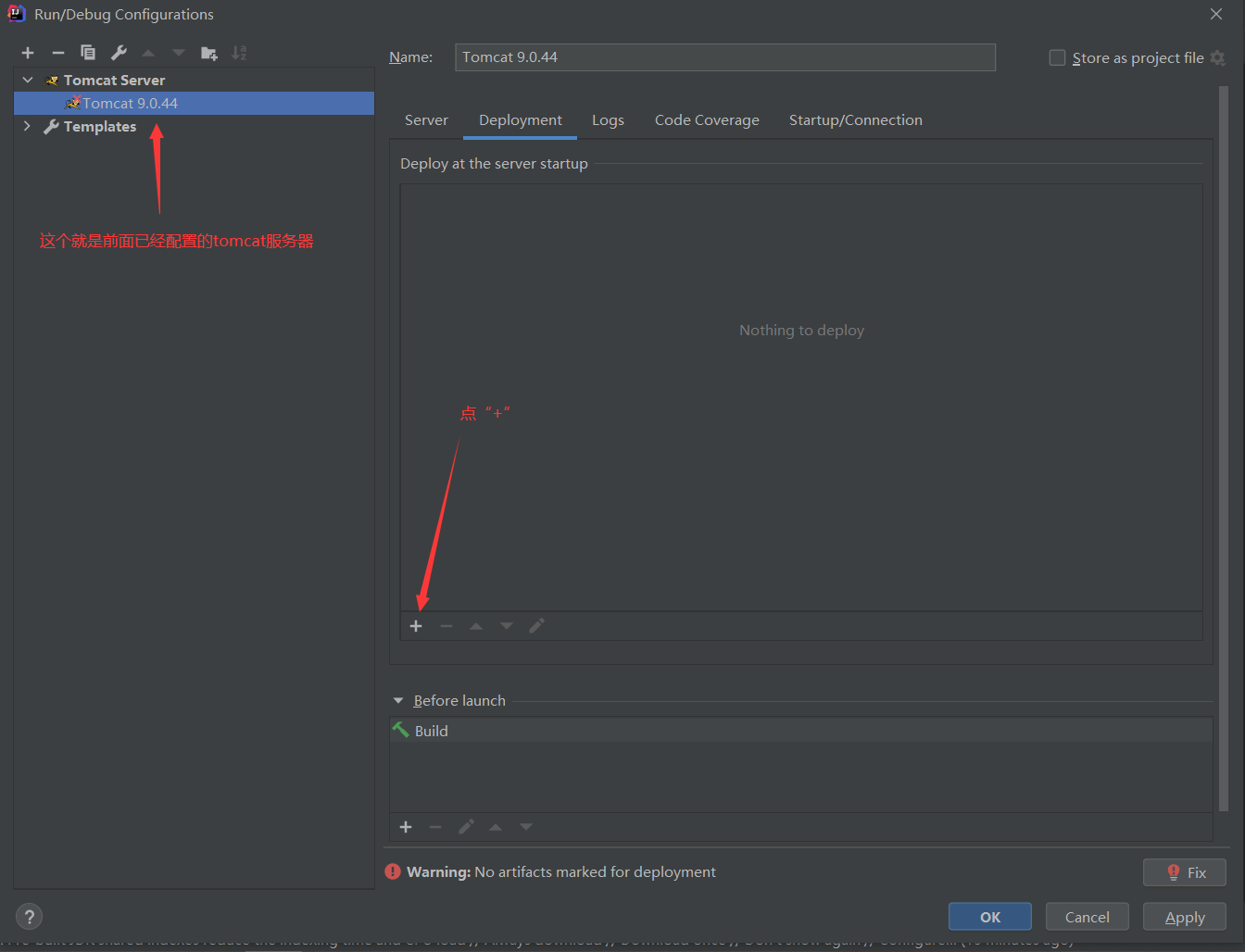


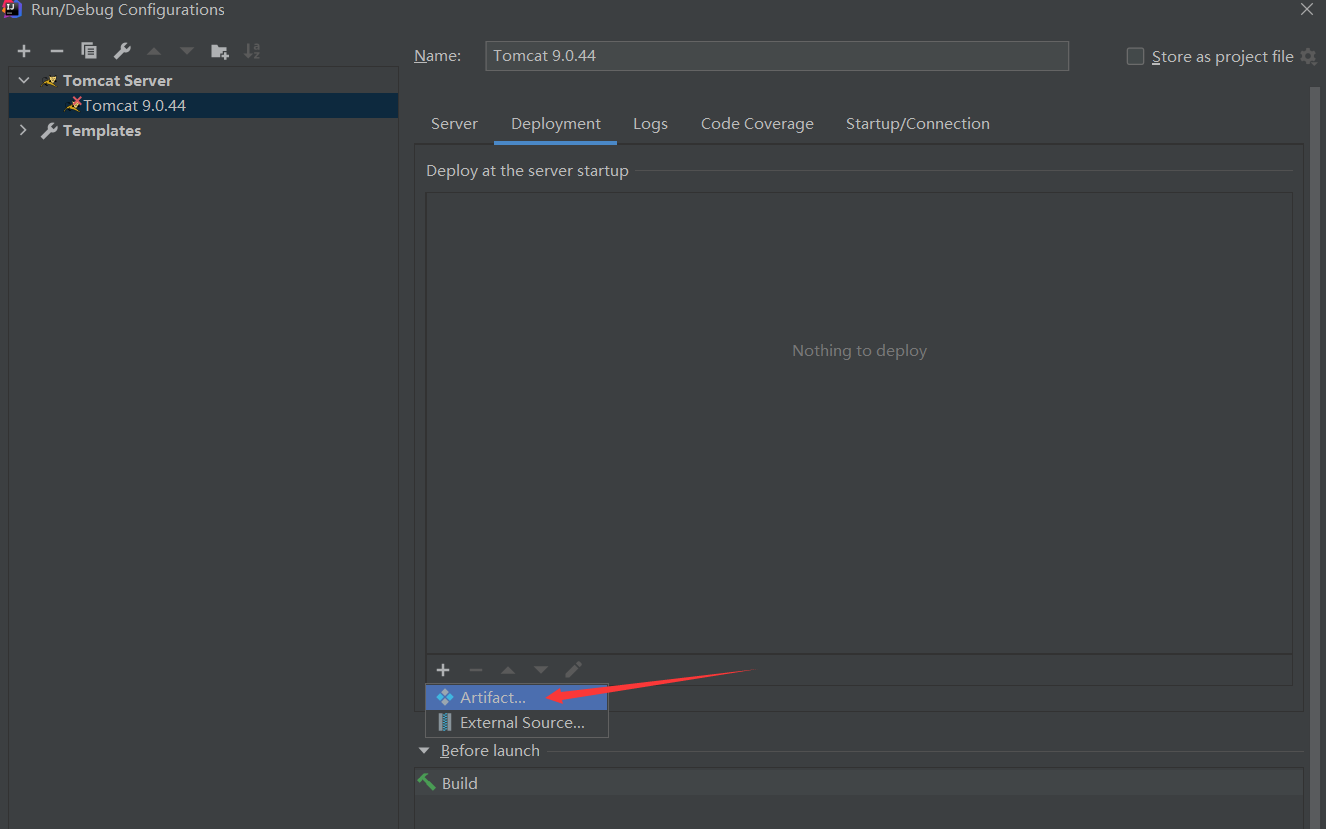


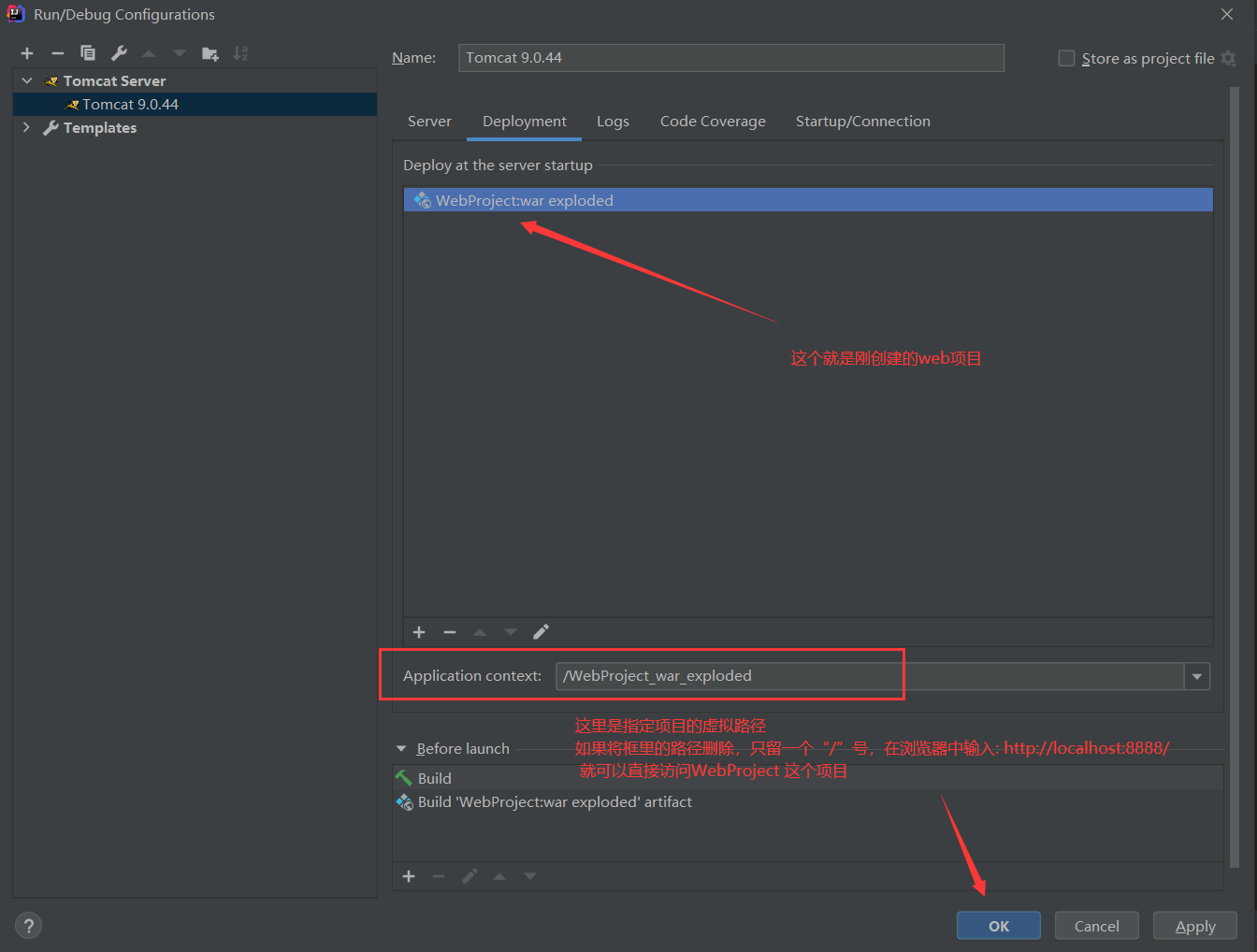
最后点击apply，再点击ok，到此结束，classes和lib就配置好。

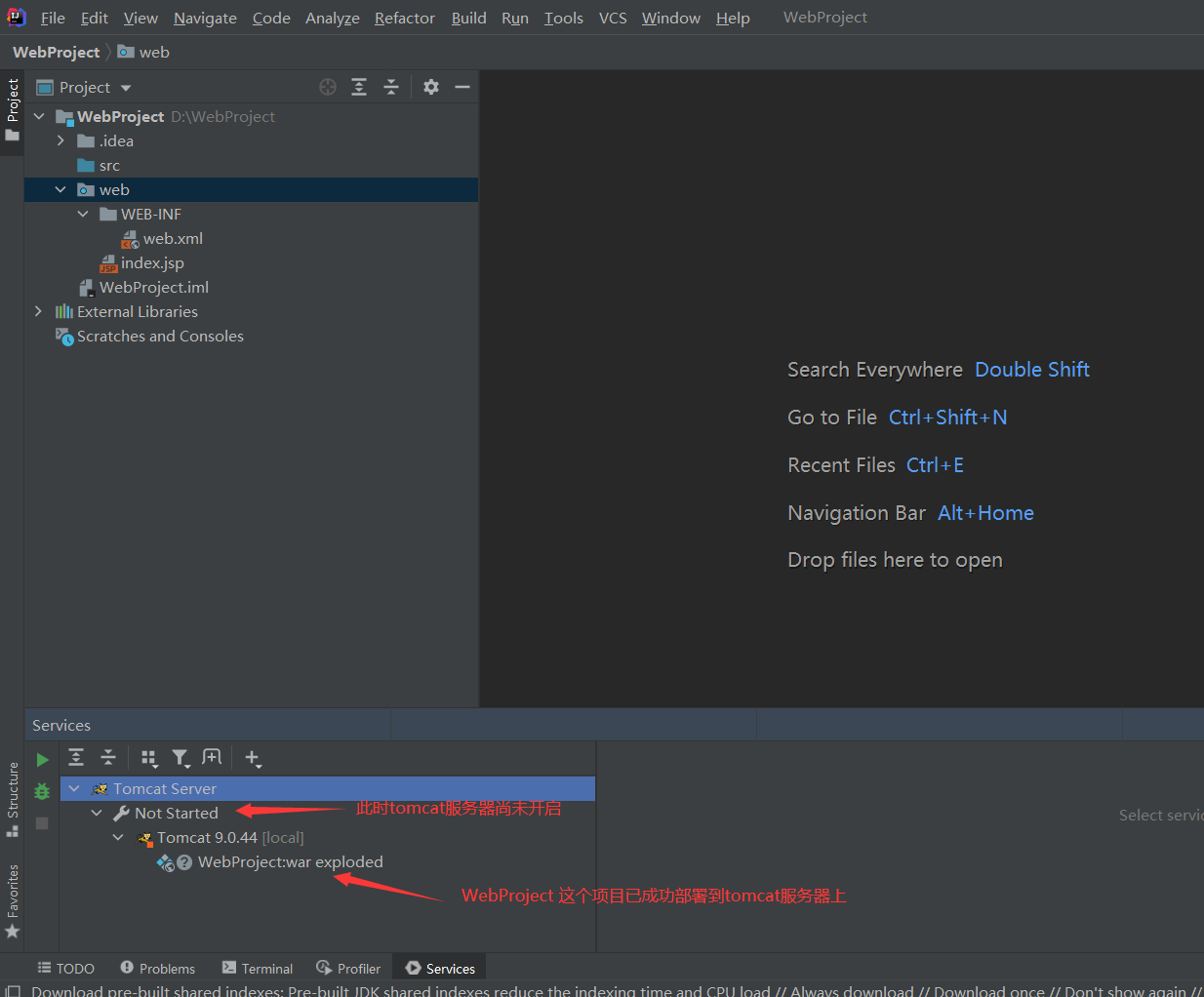
第二步: 将刚创建的web项目部署到tomcat中





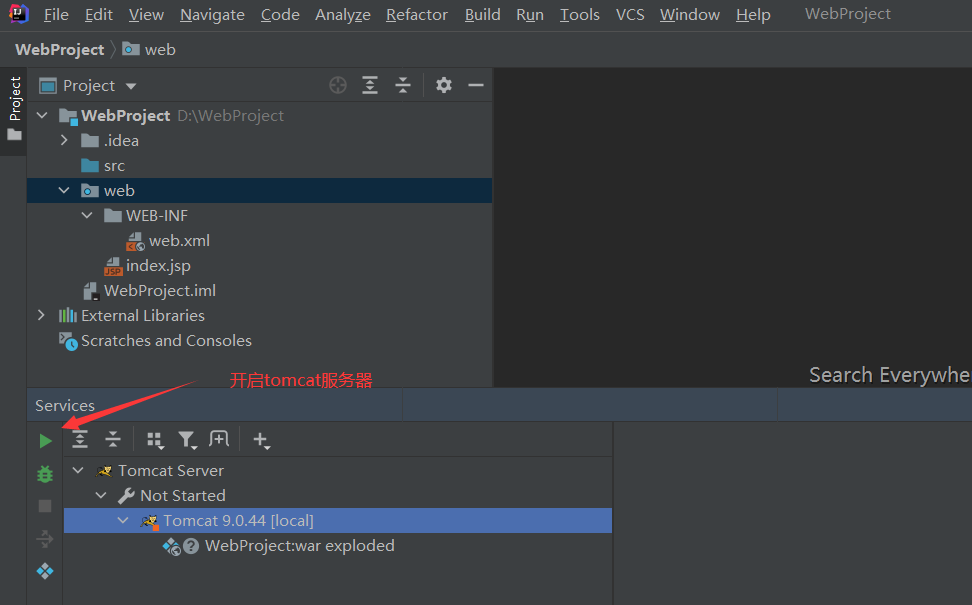


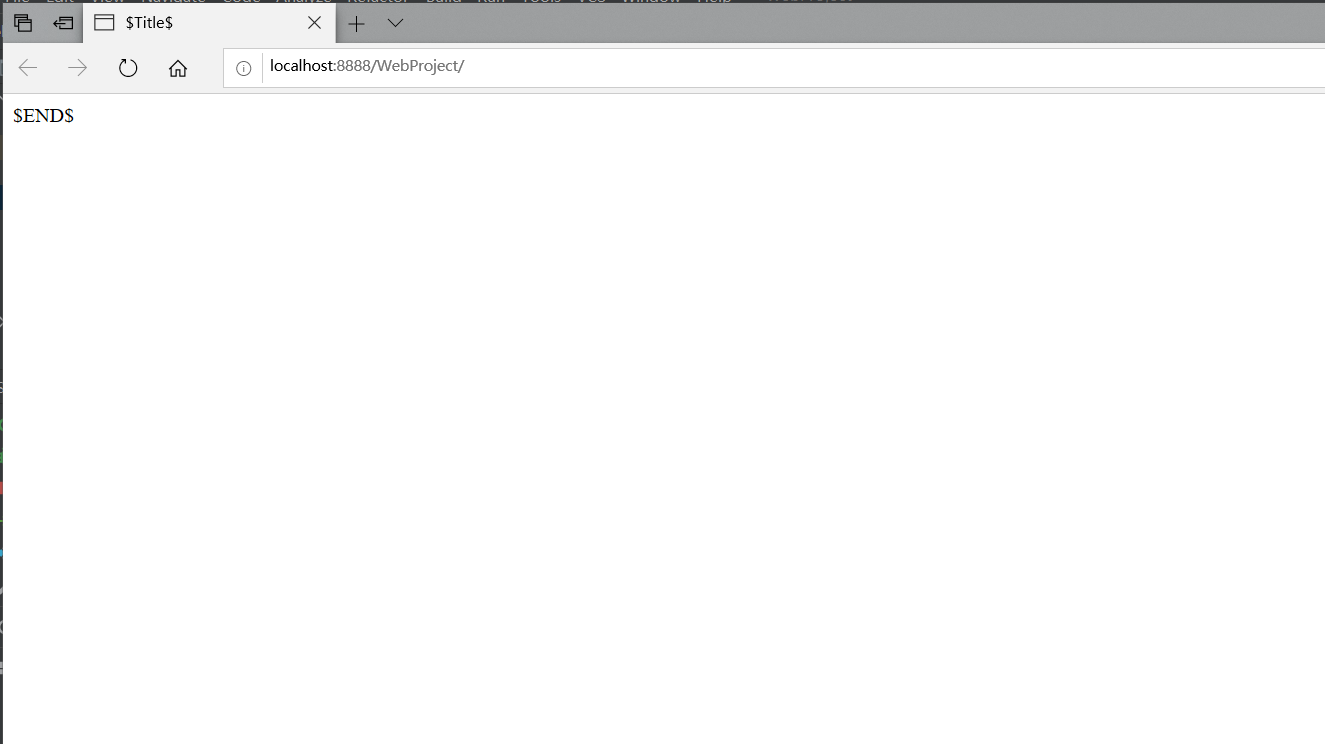




注: 如果web项目之前就已经创建好，可以直接在配置tomcat服务器的时候，就将web项目部署到tomcat中

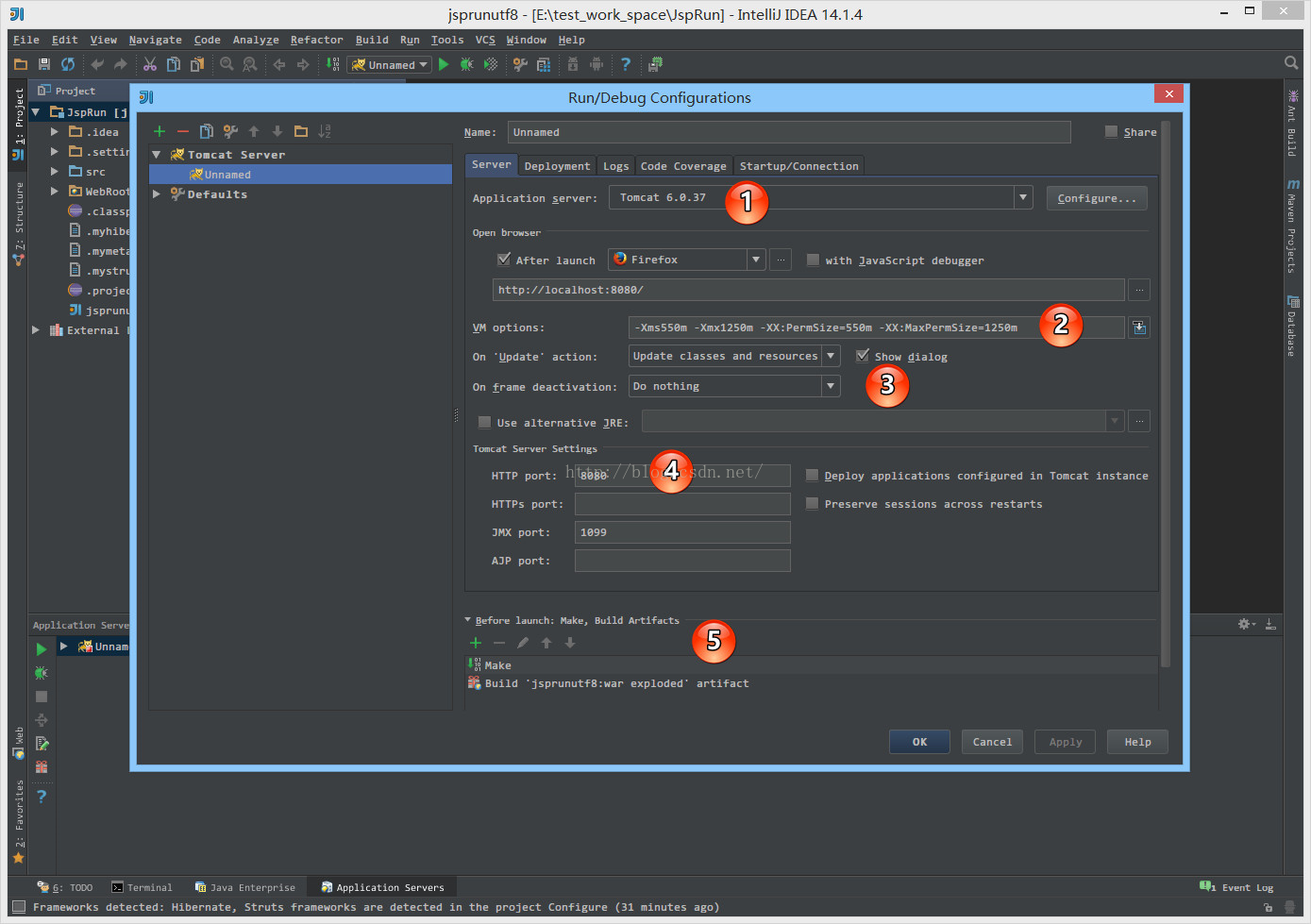
开启tomncat服务器，测试刚部署好的web项目(WebProject)能否成功运行





开启tomcat后，浏览器中会自动打开该项目的首页面(index.jsp)，说明部署的项目运行成功。

**拓展1：**

****

1 所示，我们可以切换随时为项目切换不同的容器

2 所示，我们可以指定给运行的容器设置 VM 参数

3 所示，这分别是在两种命令下的触发的事件。这个是 IntelliJ IDEA 特有的，也是重点

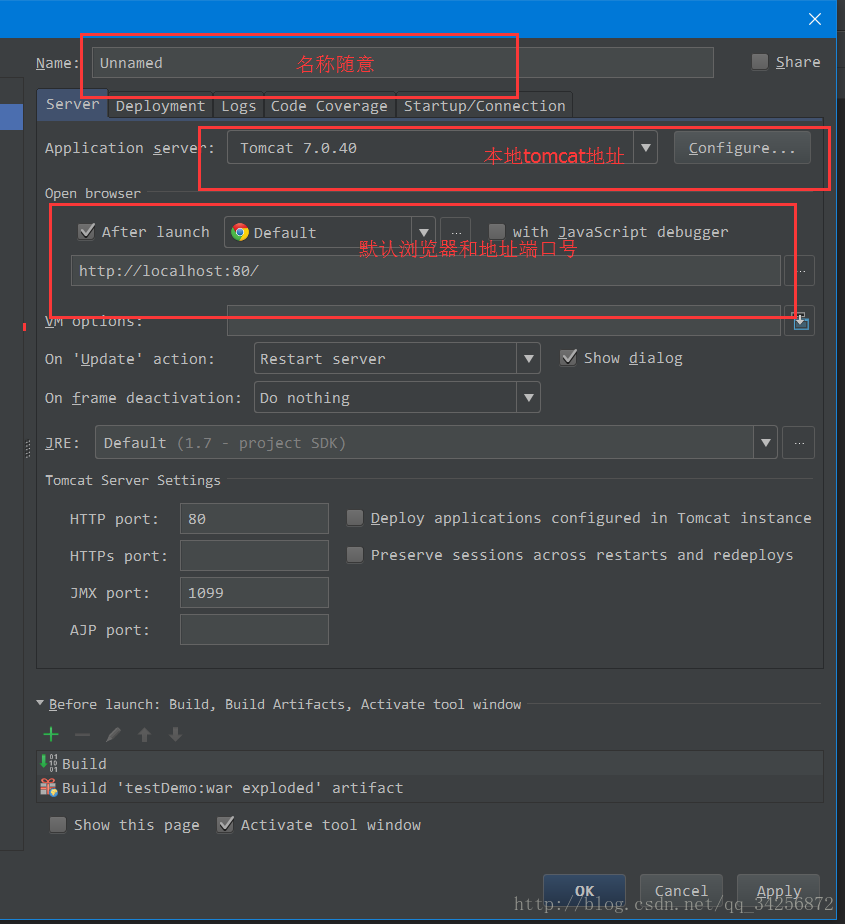
On Update action 当我们按 Ctrl + F10 进行容器更新的时候，可以根据我们配置的这个事件内容进行容器更新。其中我选择的 Update classes and resources 事件是最常用的，表示我们在按 Ctrl + F10 进行容器更新的时候，我们触发更新编译的类和资源文件到容器中。在 Debug 模式下，这个也就是所谓的 HotSwap。这是这种热部署没有 JRebel 插件那样好用

On frame deactivation 当我们切换 IntelliJ IDEA 到浏览器的时候进行指定事件更新，这个一般是因为 Web 开发的时候，我们需要经常在 IntelliJ IDEA 和各个浏览器之间来回切换测试，所以才有了这种需求。IntelliJ IDEA 是为了帮我们在做这种无聊切换的时候做一些指定事情。当然了，如果切换过于频繁，这个功能还是很耗资源的，所以我设置的是 Do nothing 表示切换的时候什么都不做

4 所示，默认 Tomcat 的 HTTP 端口是 8080，如果你需要改其端口可以在这里设置

5 所示，这个知识点在前面的文章已经有讲过了。这里表示在 Tomcat 容器运行前做什么事情，这里分别了：Make 和 Build Artifacts 操作。如上面 Gif 演示，这里的 Build Artifacts 是我们在 Deployment 选项卡中添加了 Artifact 之后自动出现的

拓展2:



**拓展3：**

