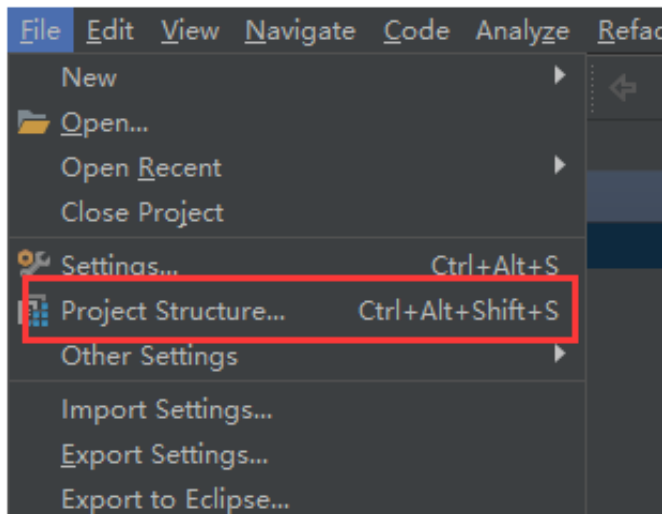
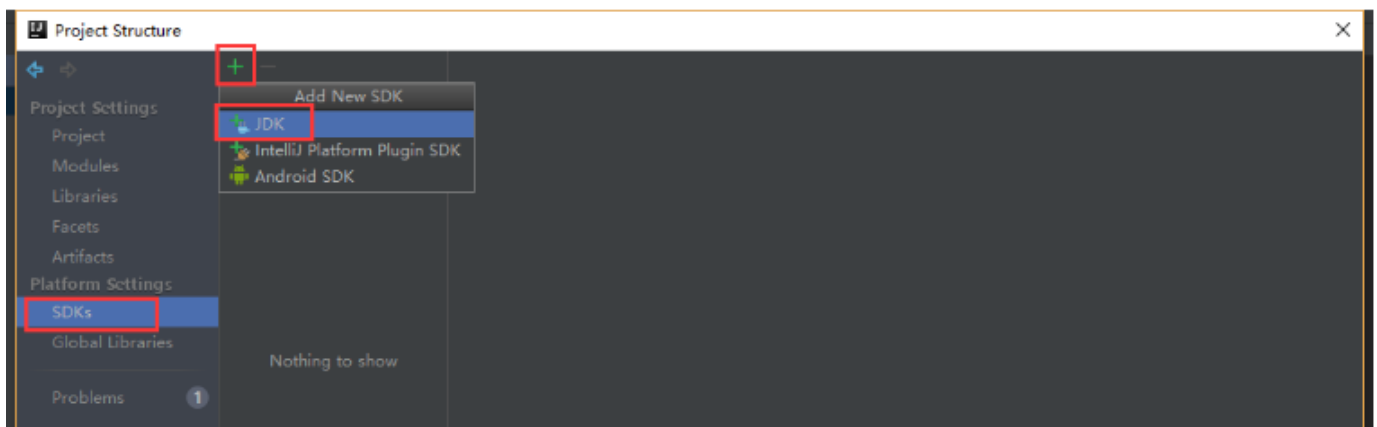


1、JDK配置 **新建项目前配置JDK**

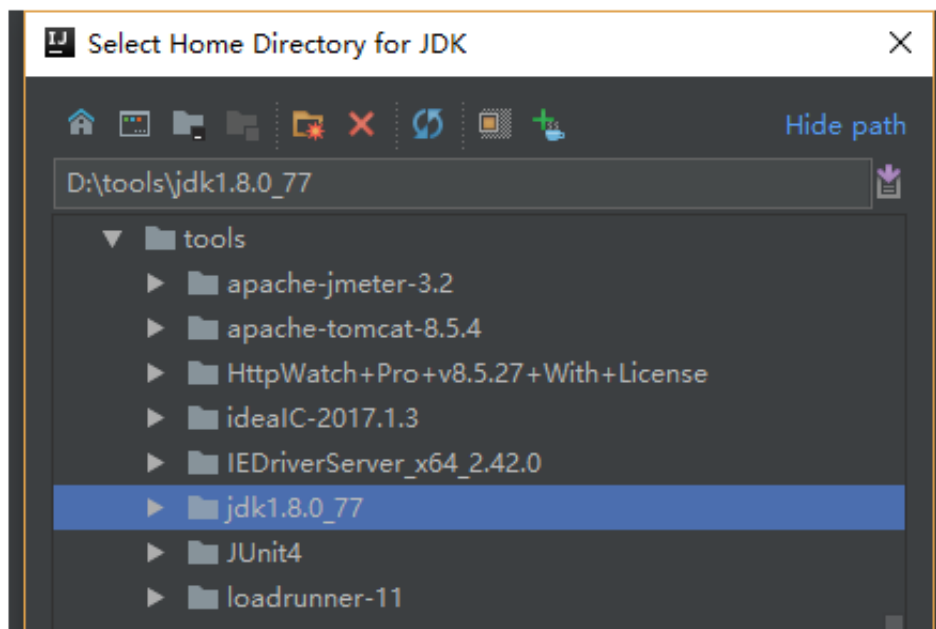
1、点击File -->Project Structure ;



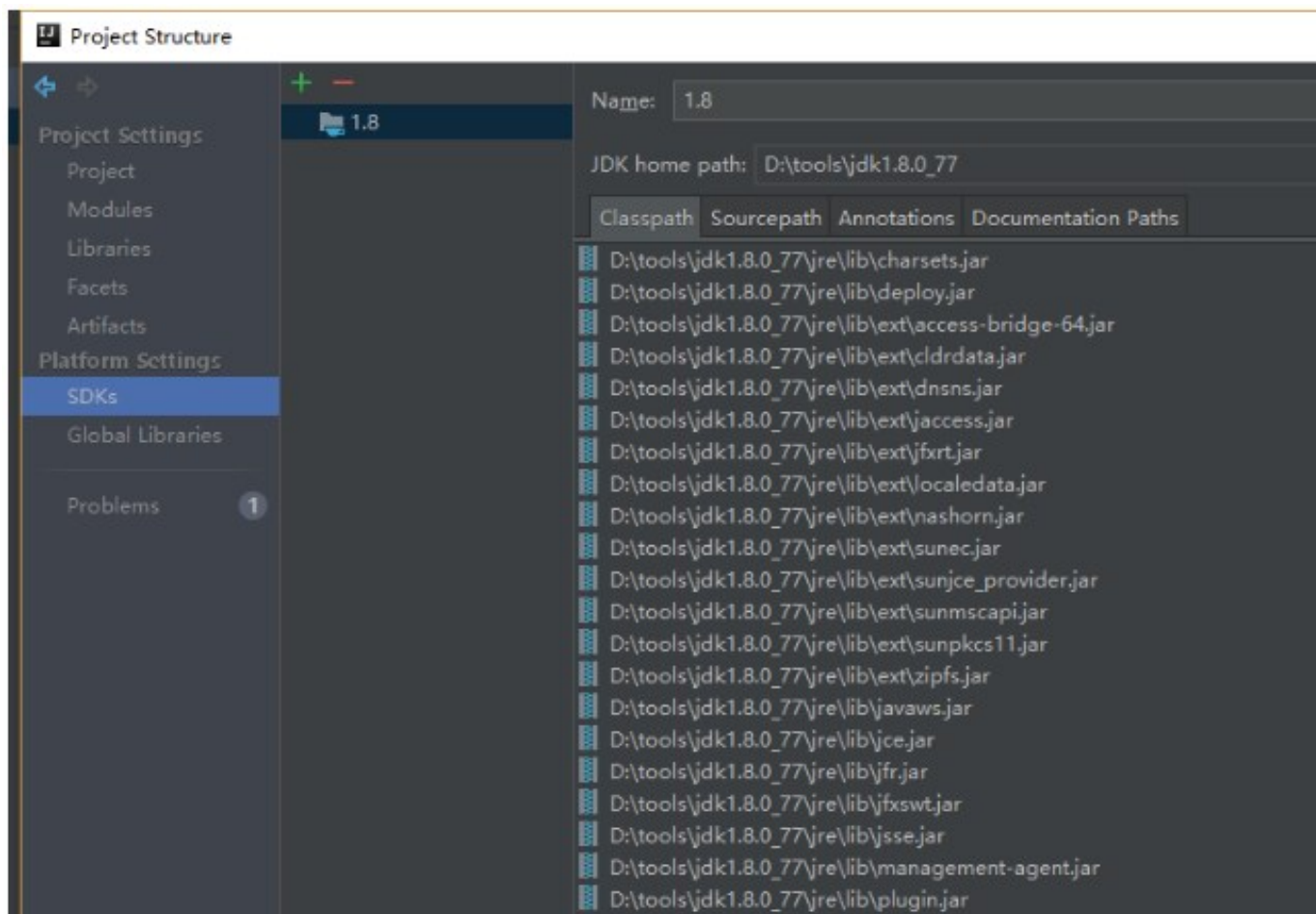
2、点击左侧标签页SDKs选项，再点击左上角“+”，选择JDK



3、在弹出框选择JDK安装路径，点击OK即可配置成功。

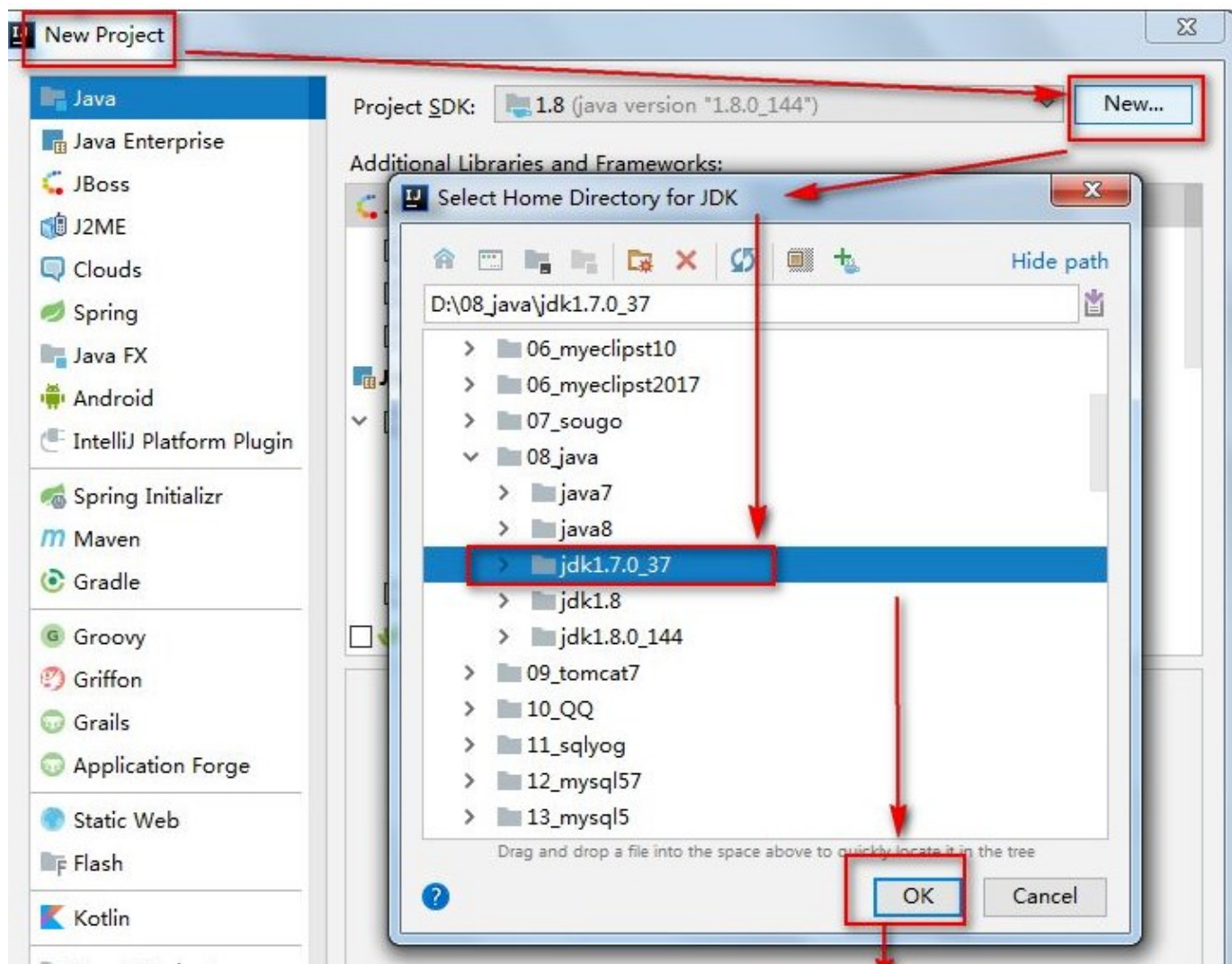


看到JDK已经在IDEA中配置好了



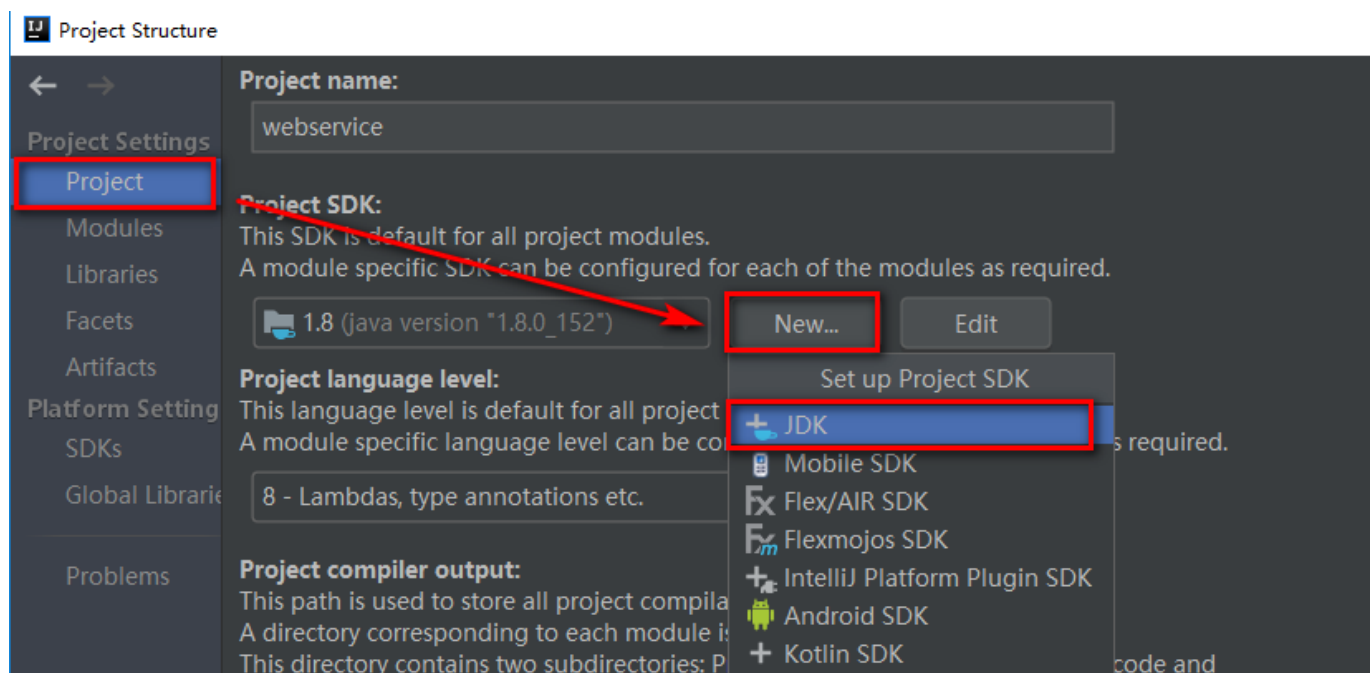
新建项目中配置JDK

打开IDEA集成开发环境工具，点击：File--New--Project



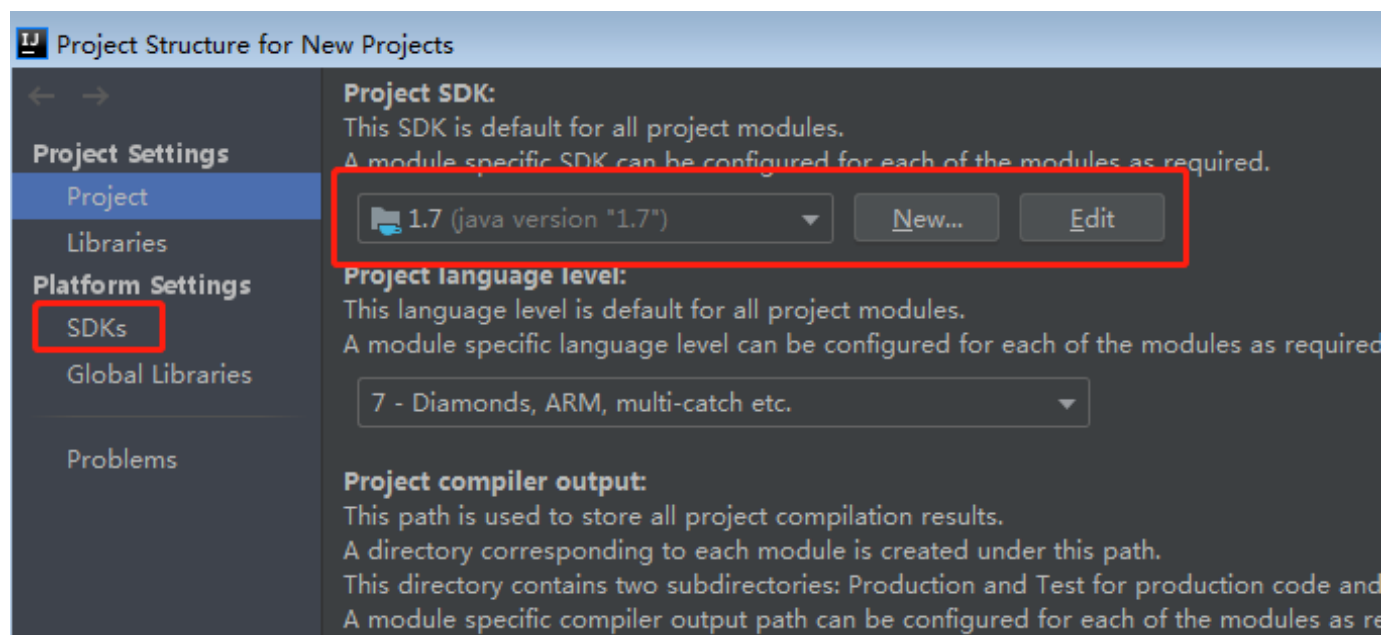
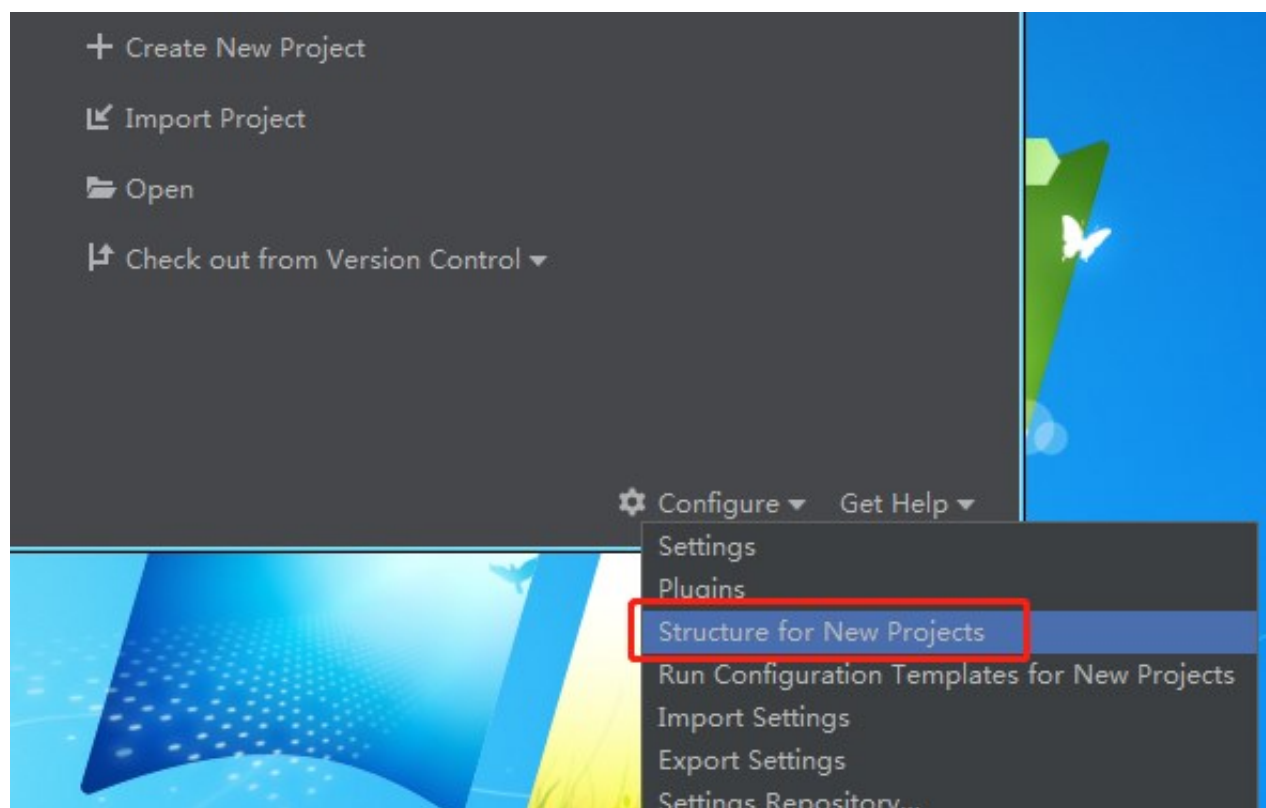
切换项目的JDK

打开IDEA集成开发环境工具，点击：File--Project Structure



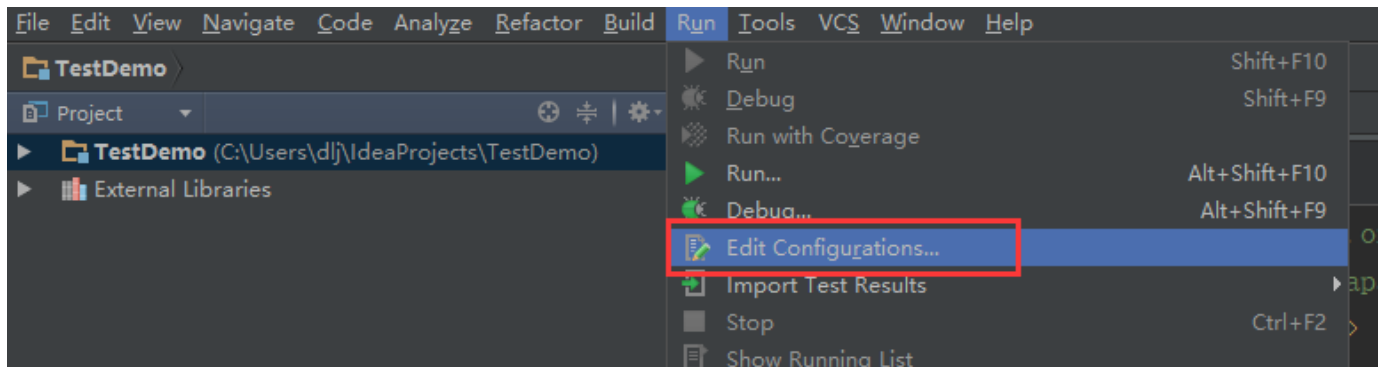
启动页面中配置JDK

在IDEA启动页面中，下拉Configure，选择Project Defaults -- Project Structure

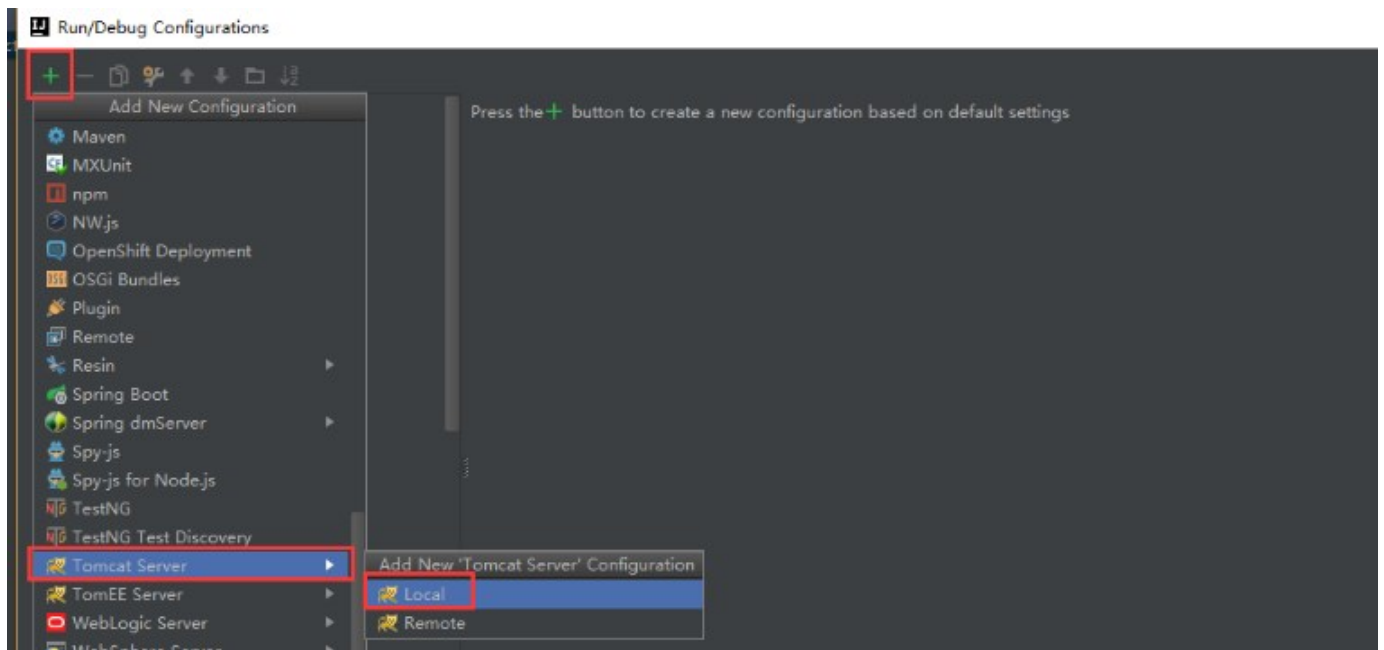


2、在IntelliJ IDEA配置Tomcat

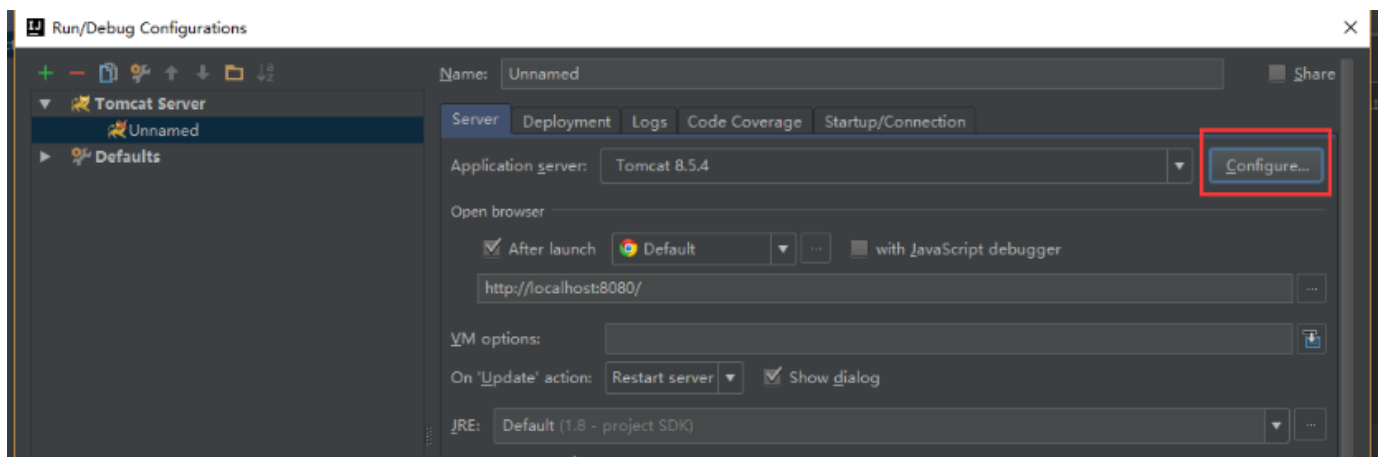
1、点击Run-Edit Configurations...



2、点击左侧“+”，选择Tomcat Server--Local，



3、在Tomcat Server -> Unnamed -> Server -> Application server项目下，点击 Configuration，找到本地 Tomcat 服务器，再点击 OK按钮



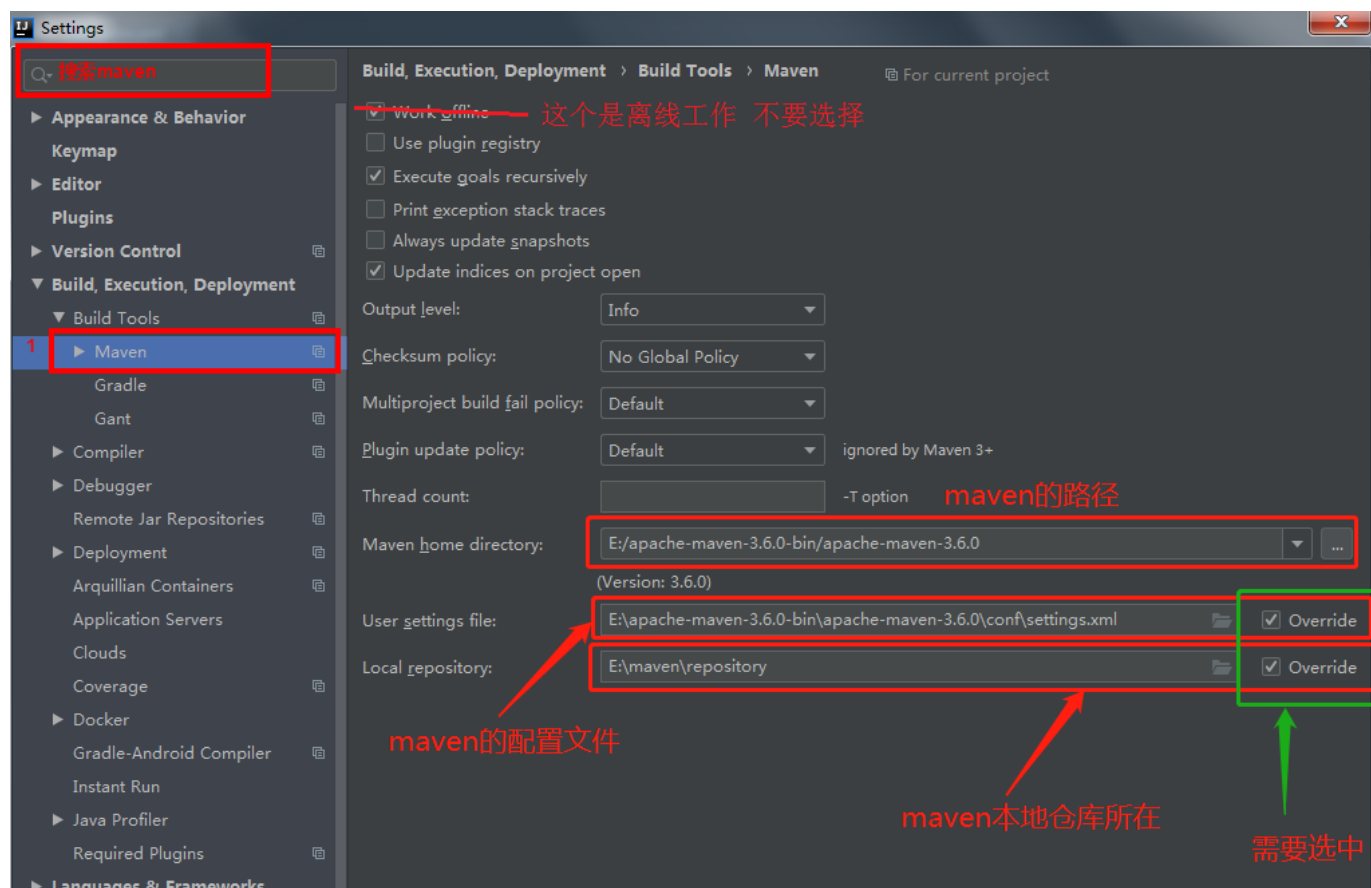
3、在IntelliJ IDEA中配置maven

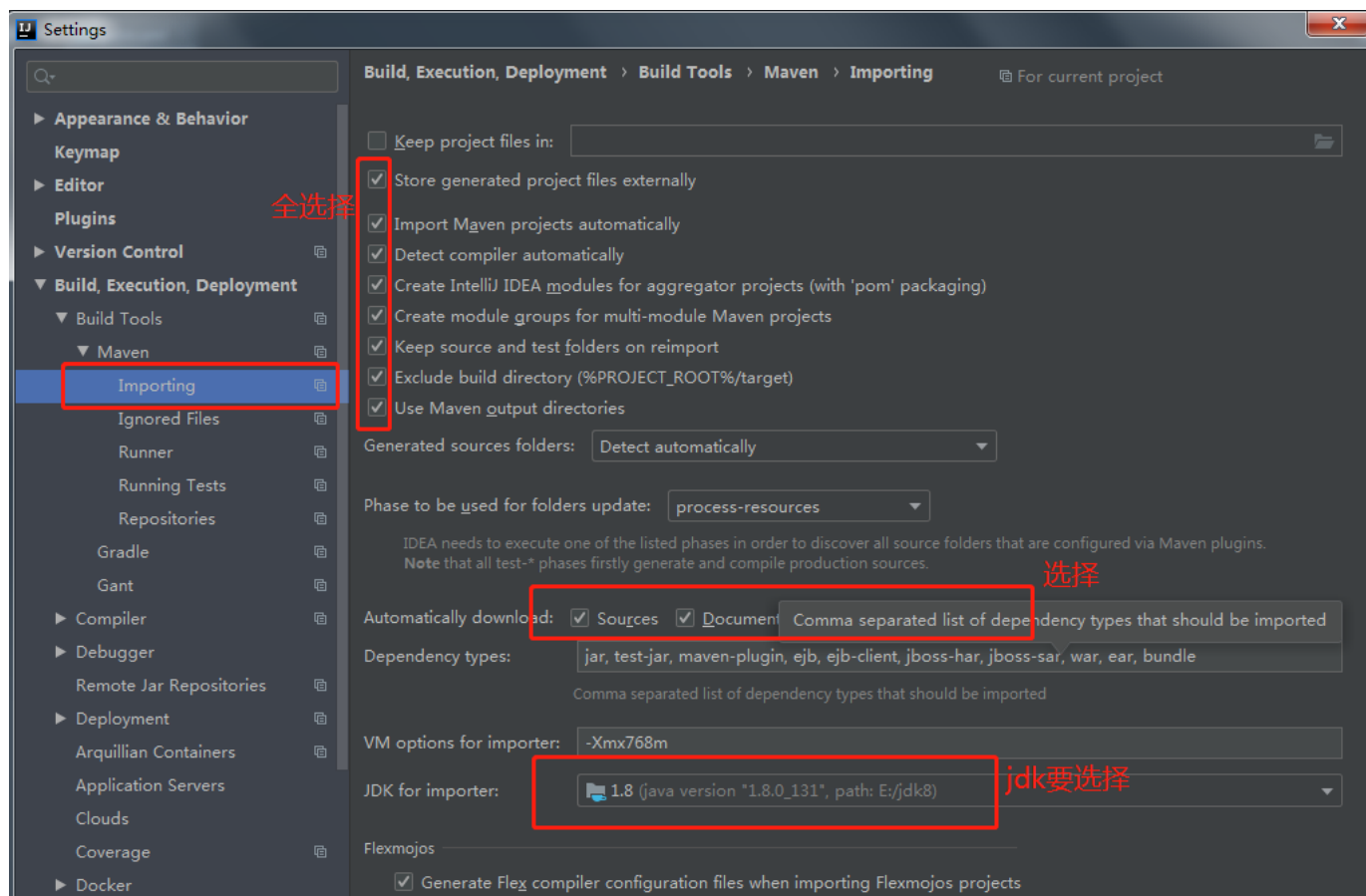
打开-File-Settings

E:/maven/apache-maven-3.3.9-bin/apache-maven-3.3.9 目录

E:\maven\apache-maven-3.3.9-bin\apache-maven-3.3.9\conf\settings.xml 配置文件

E:\maven\repository 本地仓库





idea设置Maven仓库的地址为国内镜像

第一步：找到idea中maven中的“settings.xml”位置

一般在idea安装目录下的“D:\IntelliJ IDEA 2017.2.5\plugins\maven\lib\maven3\conf”（参照左面地址）

第二步：找到“<mirrors>”标签

在“<mirrors>”标签下加下面语句：

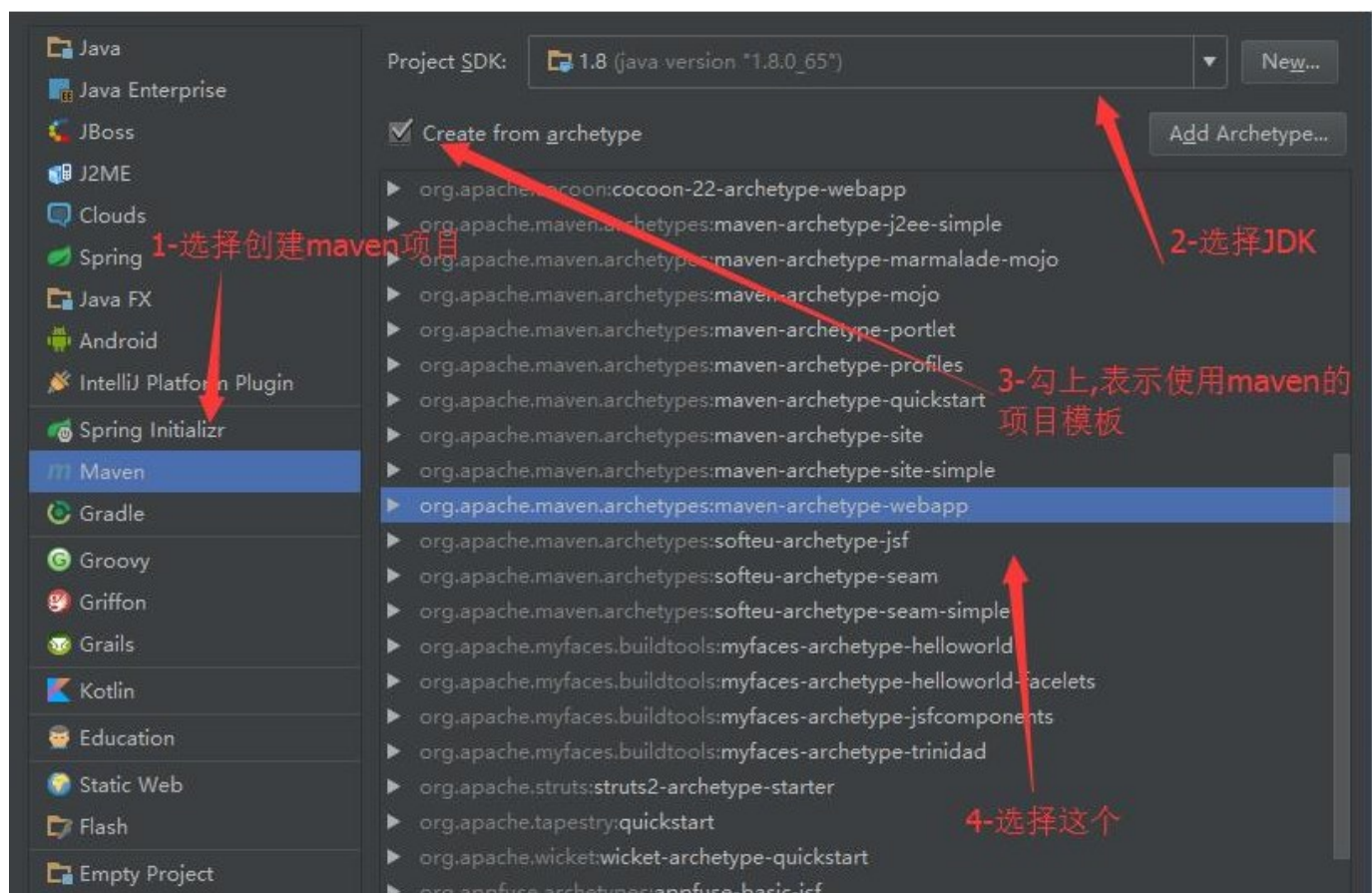
```
1 <mirror>
2     <id>nexus-aliyun</id>
3     <mirrorOf>central</mirrorOf>
4     <name>Nexus aliyun</name>
5     <url>http://maven.aliyun.com/nexus/content/groups/public</url>
6 </mirror>
```

第三步：保存

5.新建maven WEB项目

打开-File-New-Project

点击NEXT

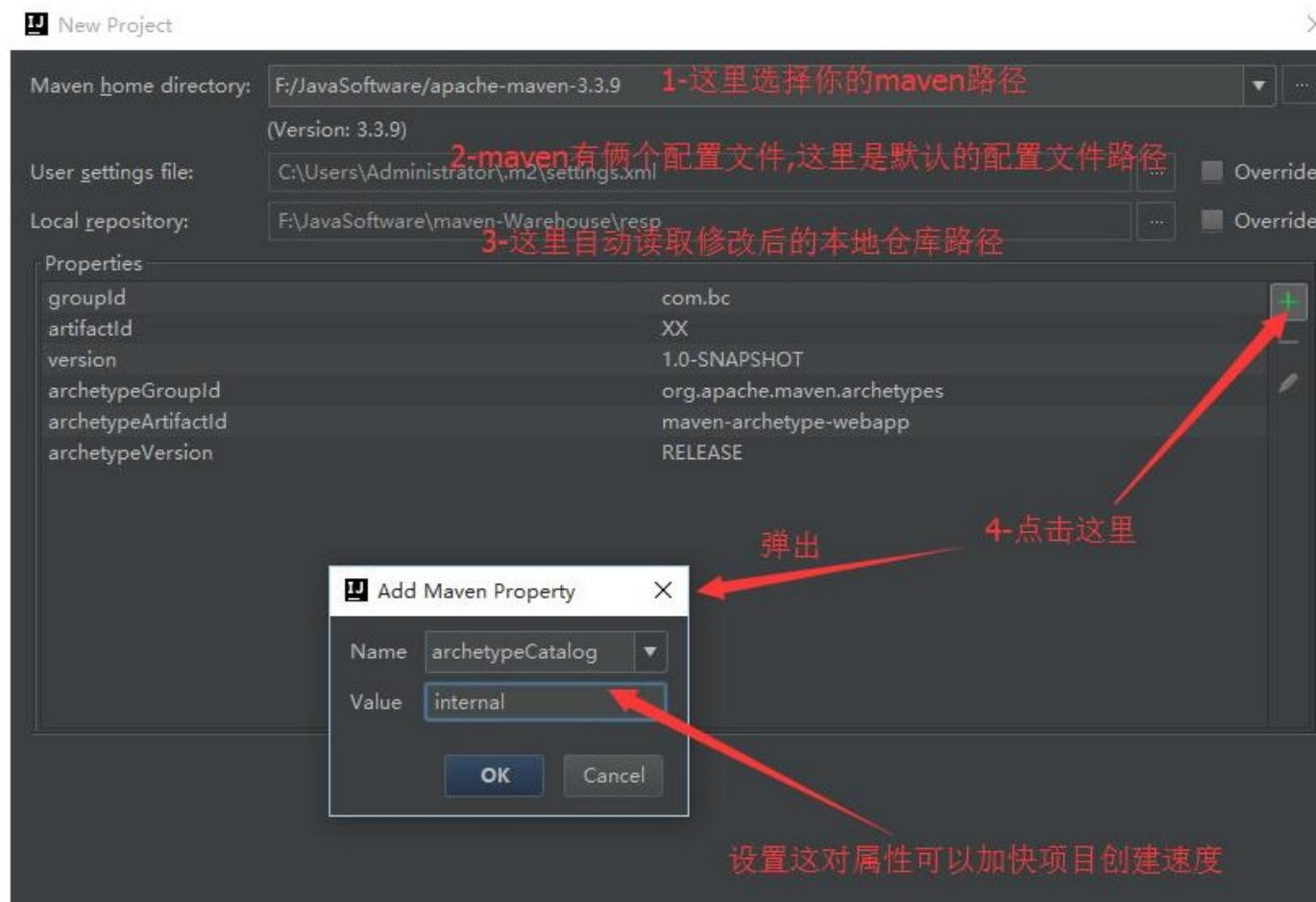


点击NEXT

GroupId	com.bc	假设有10个项目为一个组,这里填同一个组名
ArtifactId	xxj	项目模块的名称
Version	1.0-SNAPSHOT	版本号

添加的配置为 archetypeCatalog=internal

点击NEXT



点击NEXT

 New Project

Project name: XX 项目名

Project location: F:\JetBrains\Workspaces\XX

统一的项目文件夹

项目名

注意格式,最后如果不跟项目名,那么创建的maven项目文件都在一个文件夹下了

这里默认就可以,应该不会出问题

▼ More Settings

Module name:

XX

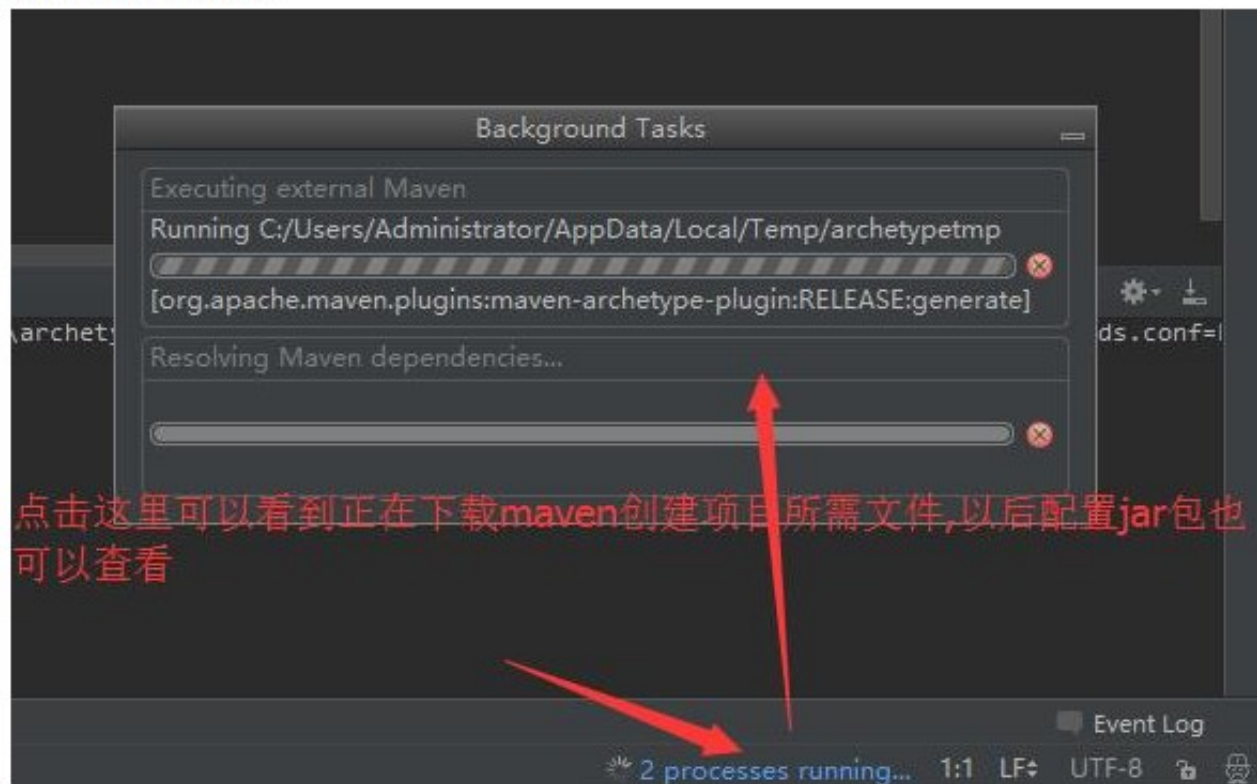
Content root:

F:\JetBrains\Workspaces\XX

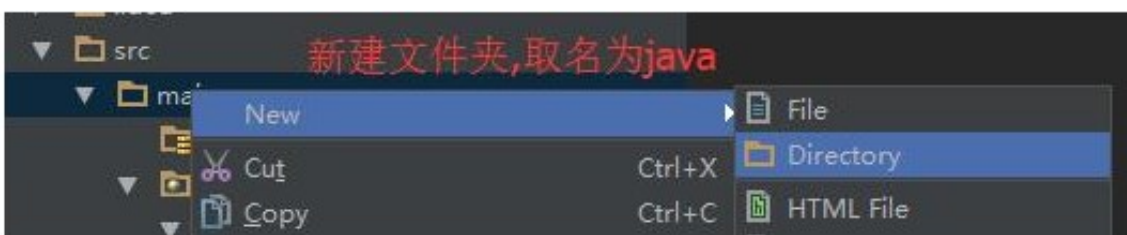
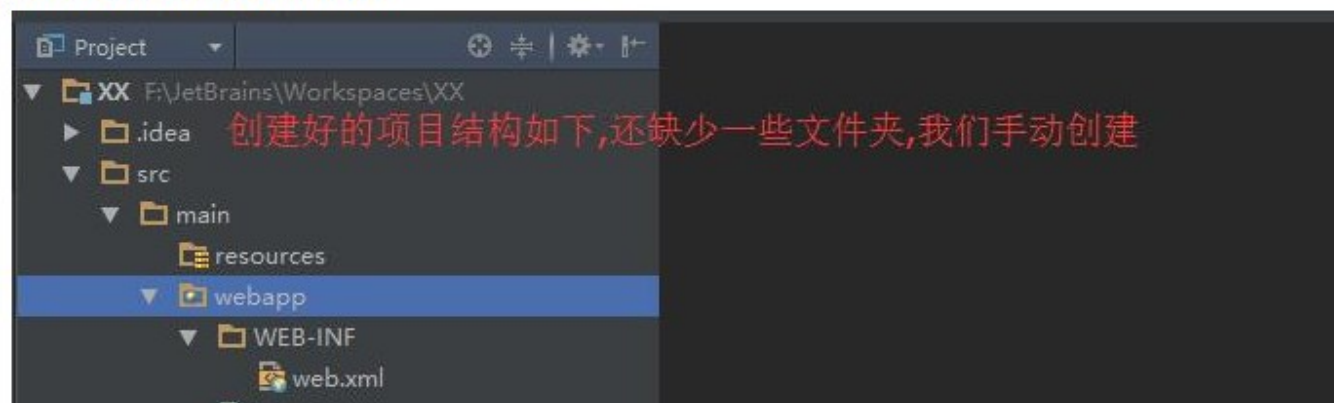
Module name: F:\JetBrains\Workspaces\XX

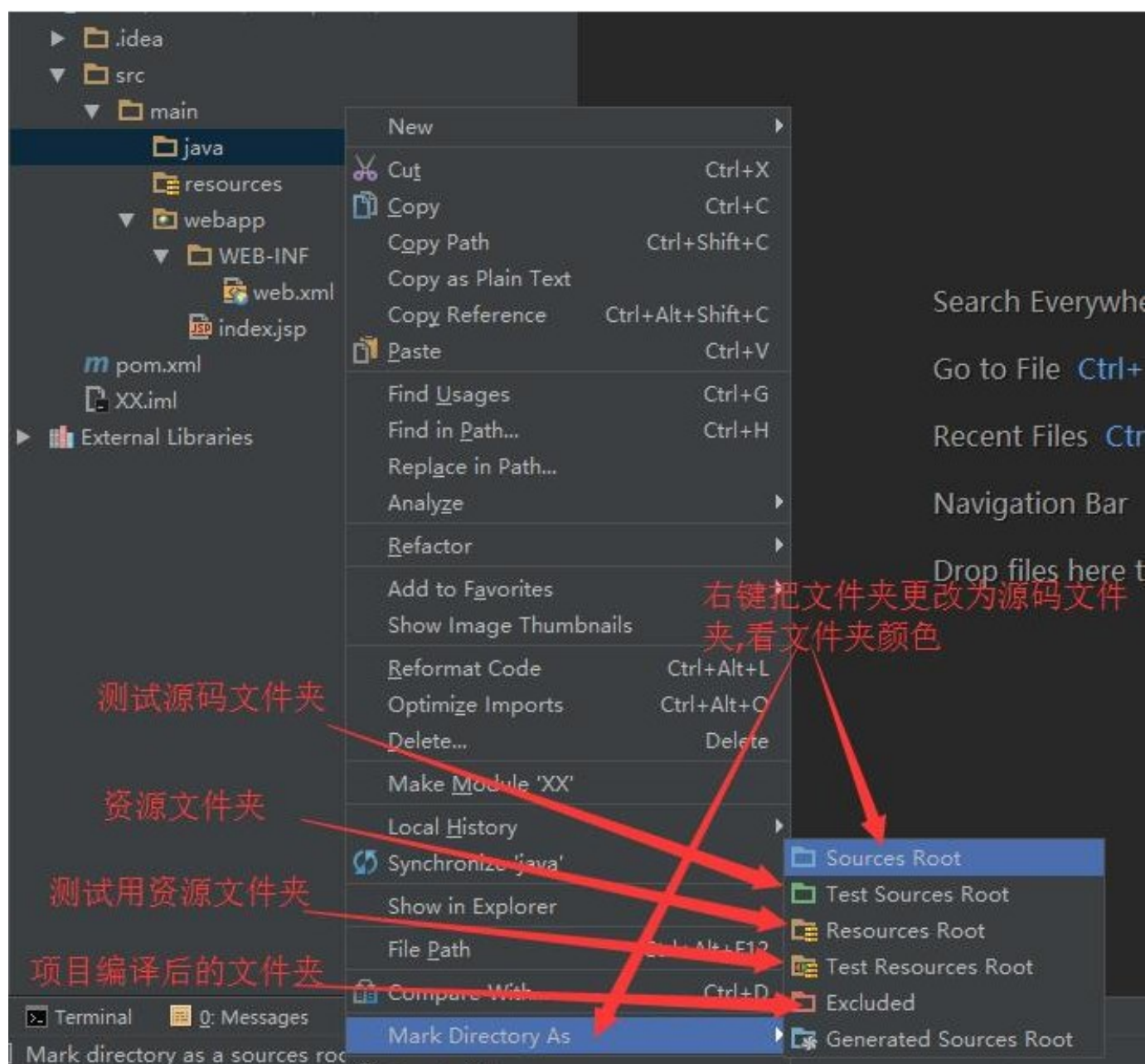
点击Finish后项目开始创建

点击右下角查看进去

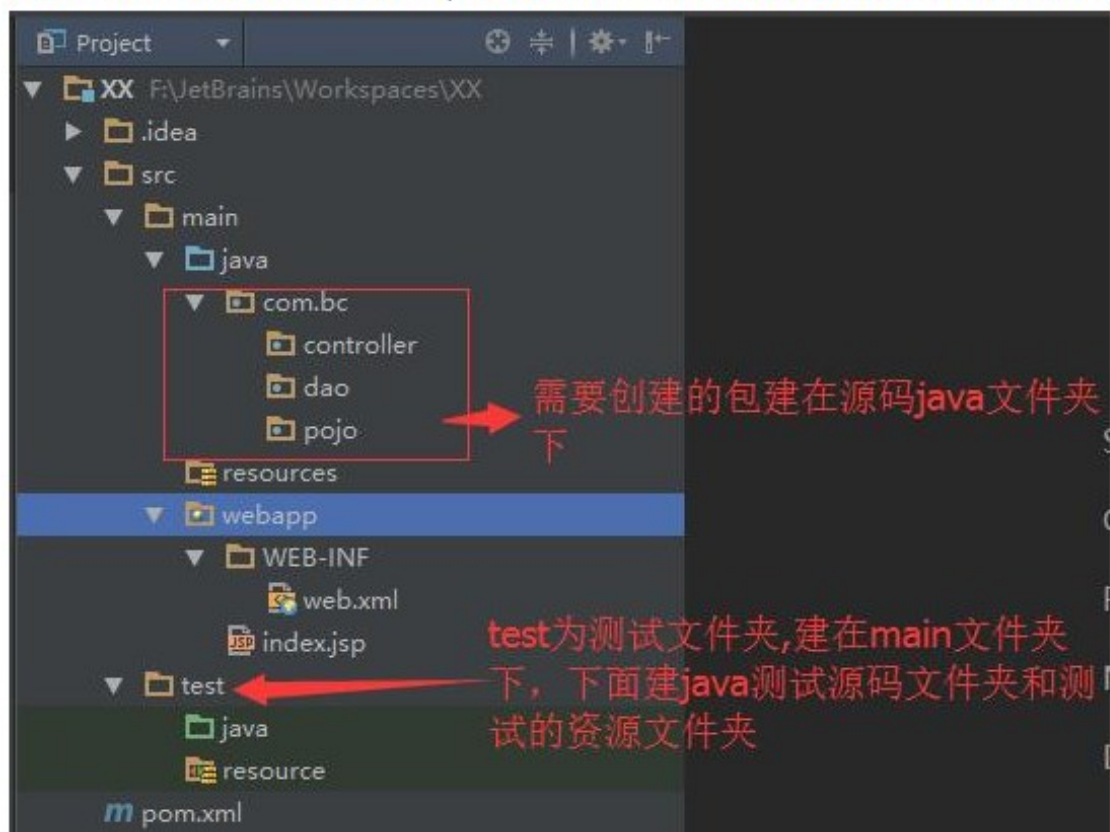


6.maven web模板项目结构

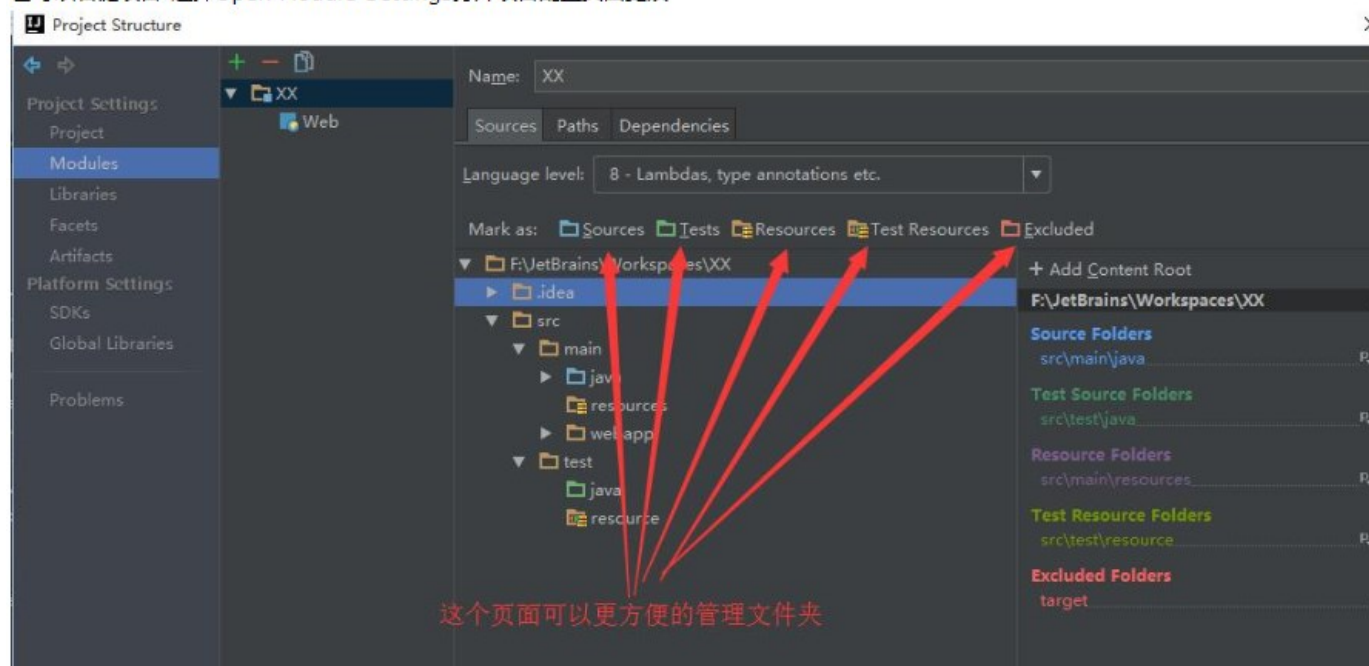




同样在main下新建test测试文件夹,再在此文件夹下新建Java测试源码文件夹和resource测试资源文件夹



也可以右键项目-选择Open Module Settings打开项目配置页面更改



7.配置依赖jar包

The screenshot displays the Maven project configuration in an IDE. On the left, the project structure shows the following hierarchy:

- XX F:\JetBrains\Workspaces\XX
 - .idea
 - src
 - main
 - java
 - com.bc
 - controller
 - dao
 - pojo
 - resources
 - webapp
 - WEB-INF
 - web.xml
 - index.jsp
 - test
 - java
 - resources
 - External Libraries

The `pom.xml` file is highlighted in the project structure, with a red box around it and the text "mavenjar包插件配置文件" (Maven jar package plugin configuration file) below it. The `pom.xml` content is shown on the right, with the following XML structure:

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://maven.apache.org/xsi:schemaLocation" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.bc</groupId> 组ID
  <artifactId>XX</artifactId> 项目ID
  <packaging>war</packaging> 打包格式,web项目是war包,也可以打成jar包
  <version>1.0-SNAPSHOT</version> 版本号
  <name>XX Maven Webapp</name> 项目名
  <url>http://maven.apache.org</url> 官网地址
  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>3.8.1</version>
      <scope>test</scope>
    </dependency>
  </dependencies>
  <build>
    <finalName>XX</finalName>
  </build>
</project>
```

Annotations in the image provide additional context:

- A red box highlights the `<dependency>` section, with a red arrow pointing to it and the text "一个<dependency>范围是所需jar包配置,可以去官网搜" (A <dependency> scope is the configuration for the required jar package, you can search on the official website).
- A red box highlights the `<build>` section, with a red arrow pointing to it and the text "这里可以配置插件" (Here you can configure plugins).

