配置环境变量

## 环境变量：

此电脑🡪右击属性🡪高级系统设置🡪环境变量

## 添加变量

系统变量选项组：新建 ->【变量名：JAVA\_HOME,变量值：jdk安装路径】

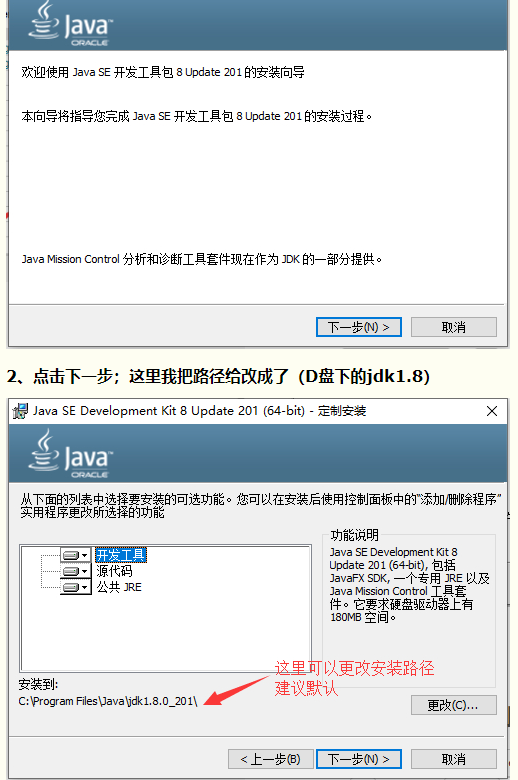
新建-> CLASSPATCH变量,变量值：.;% JAVA\_HOME %\lib; % JAVA\_HOME %\lib\tools.jar;

系统变量选项组中的Path：单击编辑 ;% JAVA\_HOME %\bin; % JAVA\_HOME %\jar\bin

测试：cmd -> javac

Java安装

官网安装：jdk（jdk1.8.exe）；







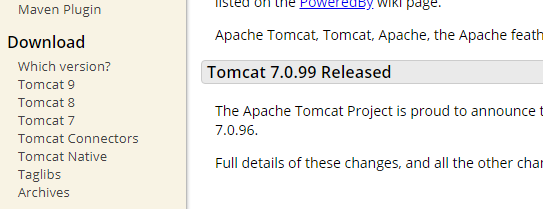
关闭

Tomcate

## 下载安装

<http://tomcat.apache.org>官网

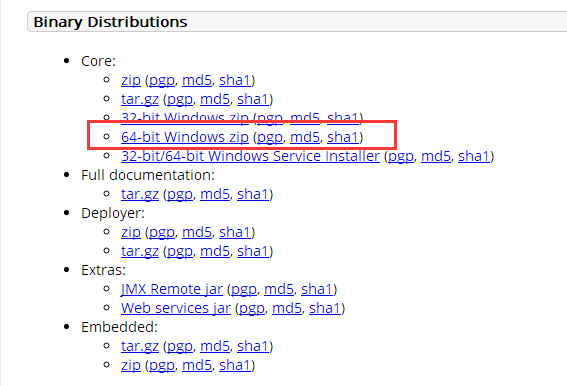
找版本



官网找到版本，下载

64-bit Windows zip(pgp,md5,sha1)免安装，解压就行

32-bit/64-bit Windows Service installer(pgp,md5,sha1)是安装的（注意版本）



目录bin下的startup.bat，点击启动Tomcat；shutdown.bat：关闭Tomcat。

## 配环境变量

启动Tomcat后，打开浏览器，键入 [http://localhost:8080](http://localhost:8080/) 进入如下页面则表示安装成功： 或者 cmd =>startup 能看见相关信息(先陪环境变量)

如果乱码：

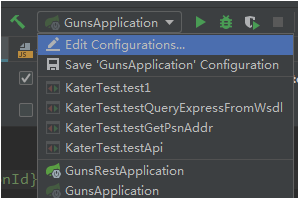
打开tomcat文件夹到conf目录下

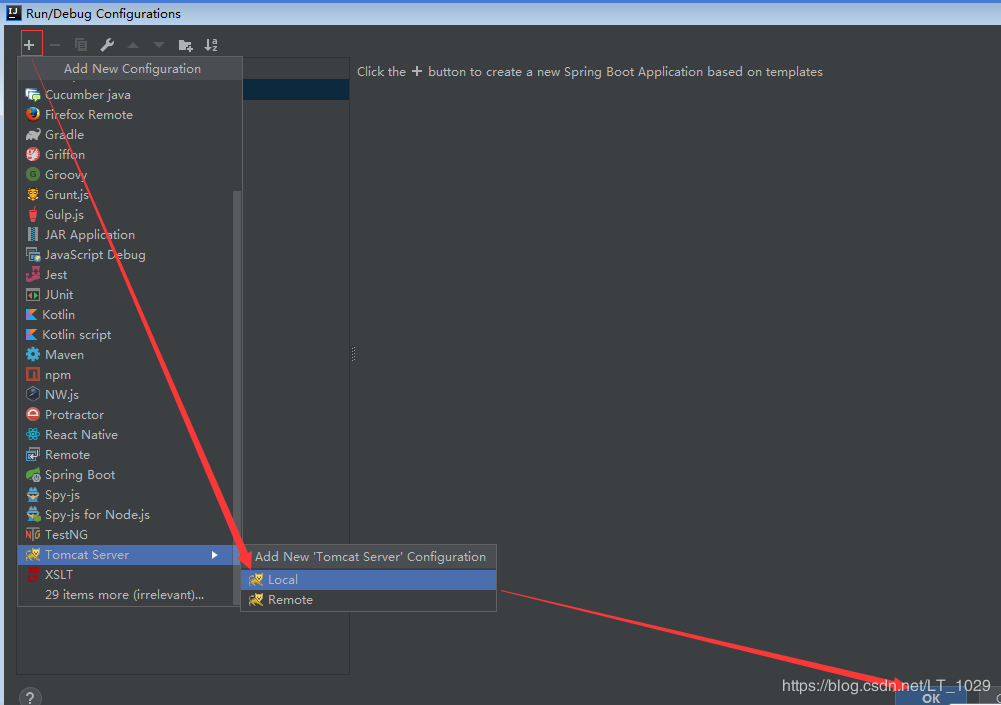
修改logging.properties

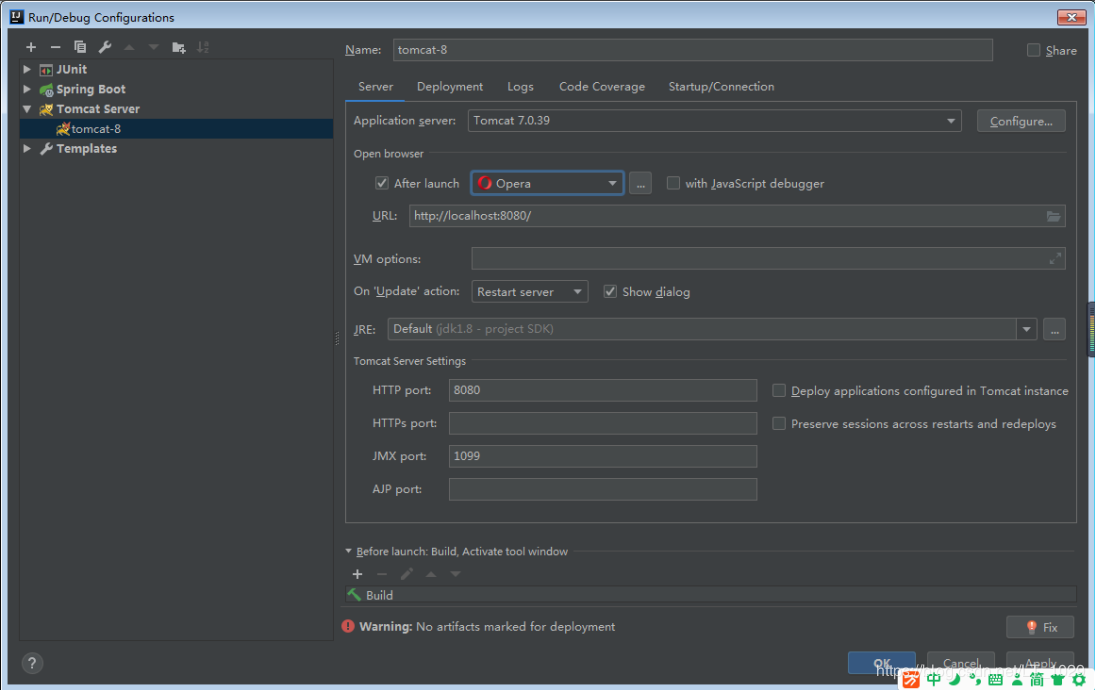
找到java.util.logging.ConsoleHandler.encoding = utf-8这行

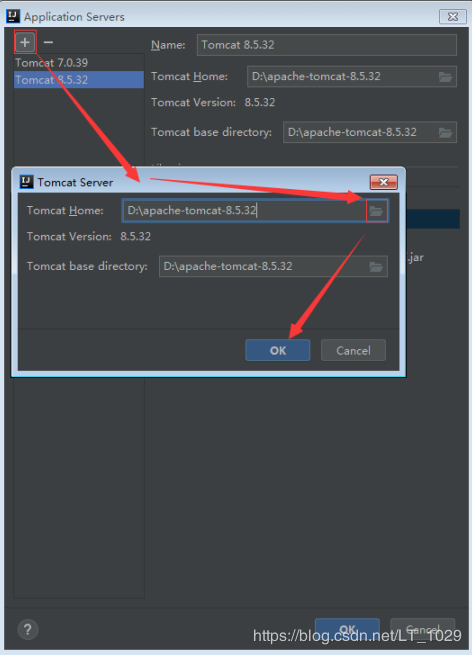
更改为java.util.logging.ConsoleHandler.encoding = GBK

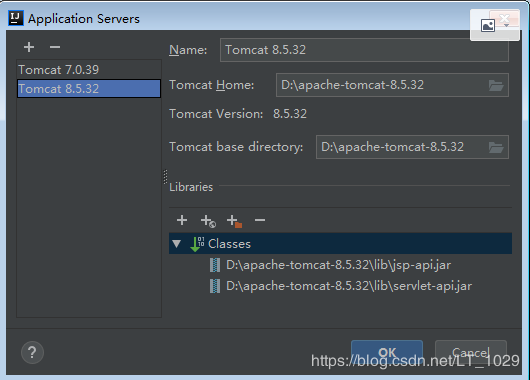
## Idea 配置tomcat

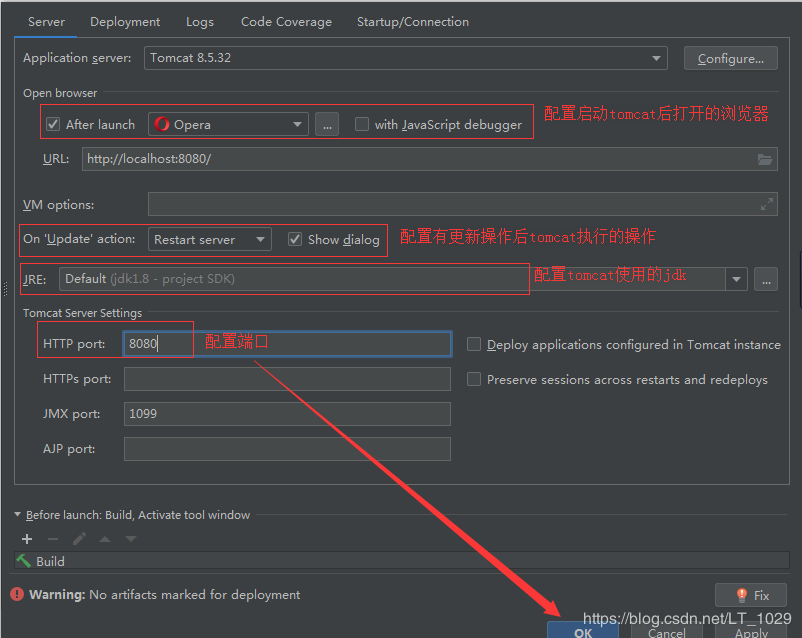












配置tomcat实现热部署与热加载的方式

(1)、在tomcat的webapps中新建一个与项目名相同的文件，并把项目下WebContent文件夹下的内容剪切到刚刚新建的文件夹中。这是tomcat自带的热部署。

(2)、在tomcat\conf\server.xml中的Host标签中加上一个配置。

<Context docBase="D:/test/web" path="/hot" privileged="true" reloadable="true"/>

path:指定访问该web应用的URL进口

docBase:指定web应用的文件路径，可以给定绝对路径，也可以给定相对于<Host>的appBase属性【默认

    指向tomcat的webapps】的相对路径；要是Web应用是个war文件，则指定war文件的路径。

reloadable:要是这个属性设置为true,Tomcat服务器在运行状况下会监视在WEB-INF/classess和WEB-

    INF/lib目次下的class文件的改动，以及监视web应用的WEB-INF/web.xml文件的改动。要是检测到的class

    文件或者web.xml文件被更新，服务器会自动加载Web应用。该属性的默认值为false.在web应用的开发和调

    试阶段，把reloadable设为true,可以方便对web应用的调试。在web应用正式发布阶段，把reloadable设为

    false,可以减低tomcat的运行负荷，提高Tomcat的运行性能。

(3)、在tomcat\conf\catalina\localhost中添加一个xml文件

hot.xml

<?xml version="1.0" encoding="UTF-8"?>

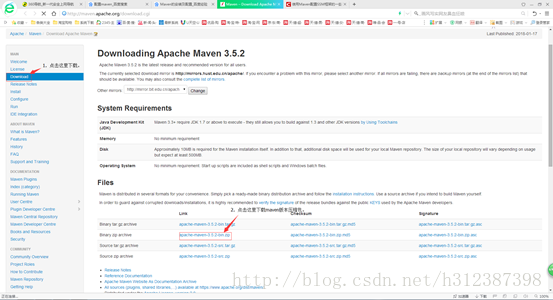
<Context docbase="D:/test/web" reloadable="true"/>

服务器会使用xml文件的名字作为URL路径

Maven

## 下载安装

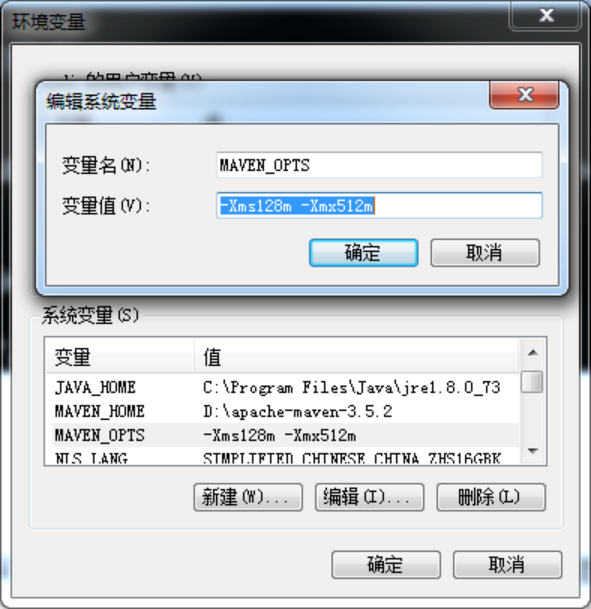
<http://maven.apache.org/download.cgi>官网



解压，确保安装了JDK cmd =>java -version

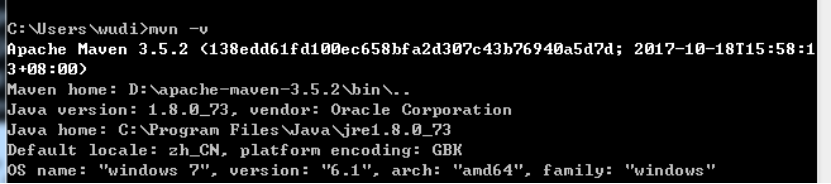
## 配置环境变量

多加一条

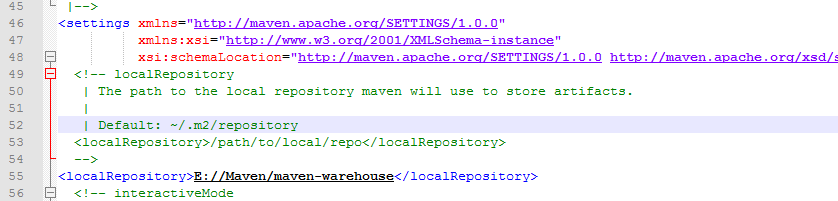


需要设置MAVEN\_OPTS的值为-Xms128m -Xmx512m，因为Java默认的最大可用内存往往不能够满足Maven运行的需要，比如在项目较大时，使用Maven生成项目站点需要占用大量的内存，如果没有该配置，则很容易得到java.lang.OutOfMemeoryError。

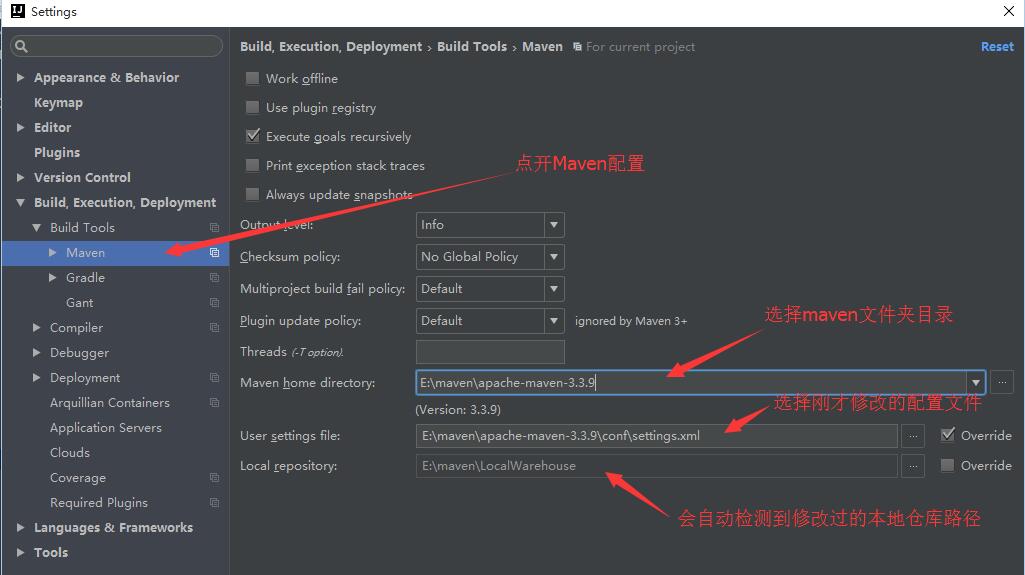
Mvn –v查看成功不。



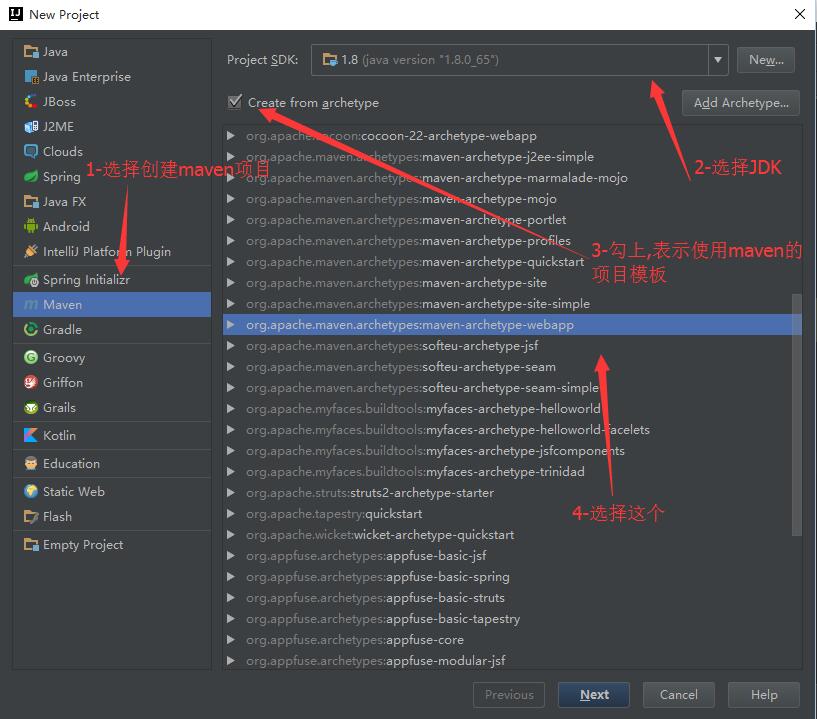
仓库配置：安装目录apache-maven-3.2.3\conf\setting.xml

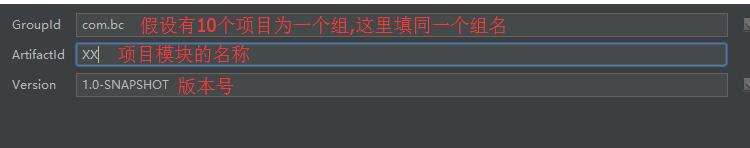


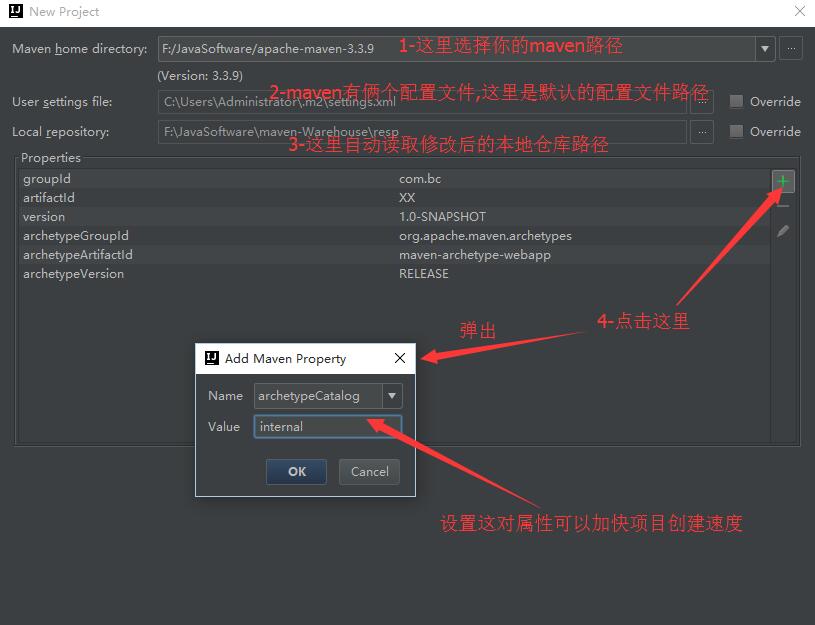
Idea配置和创建maven

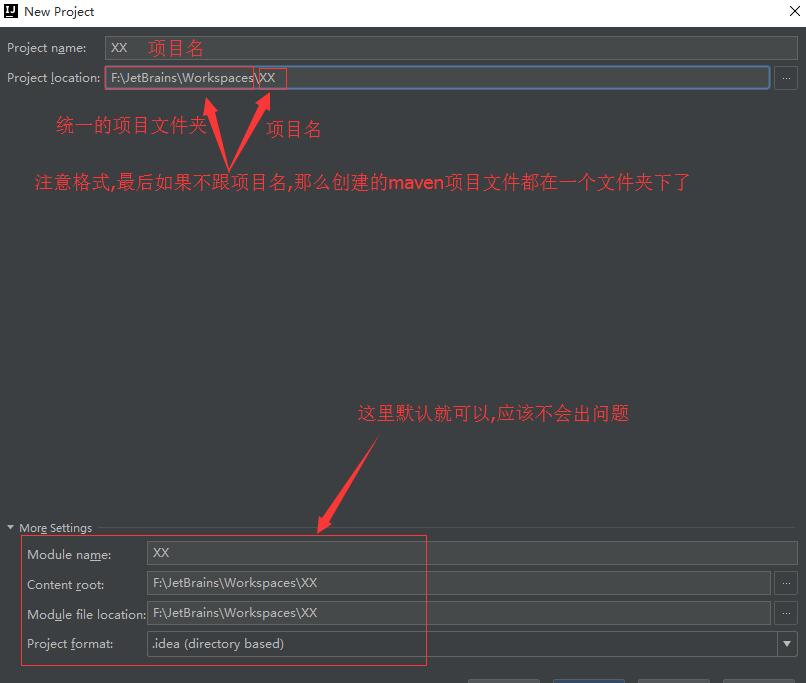


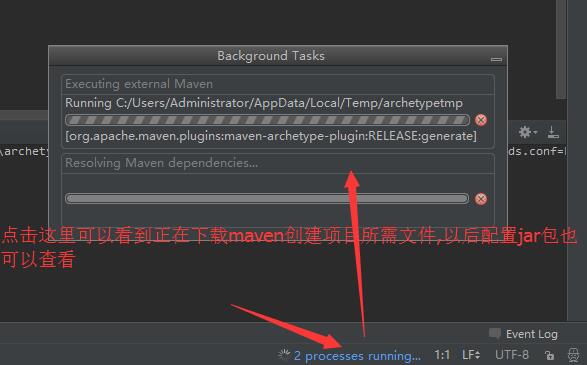
点击next



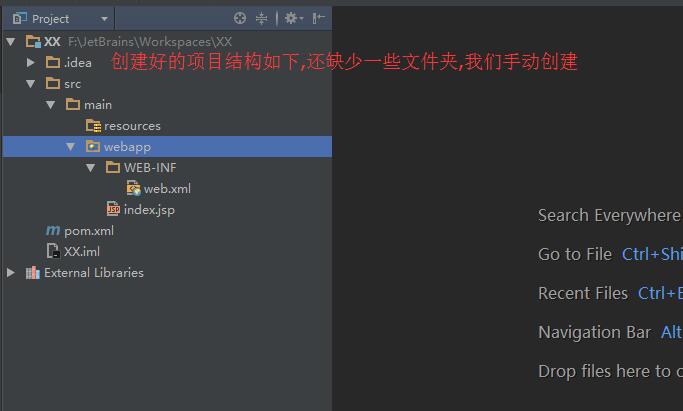




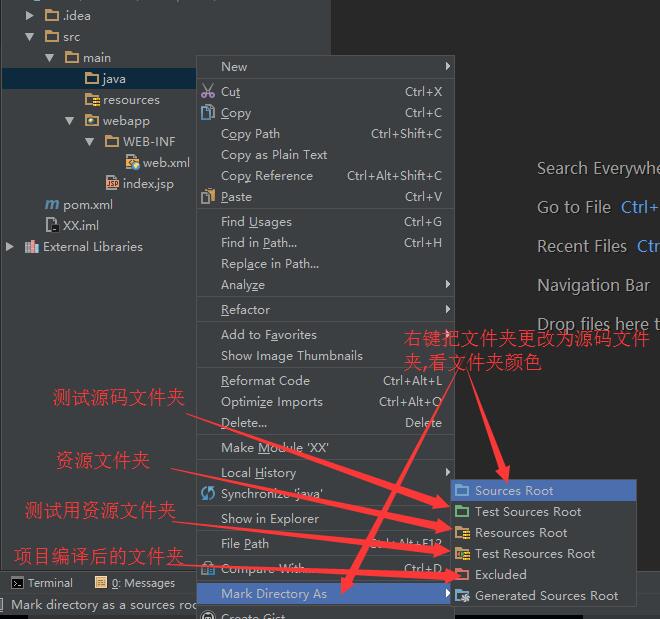


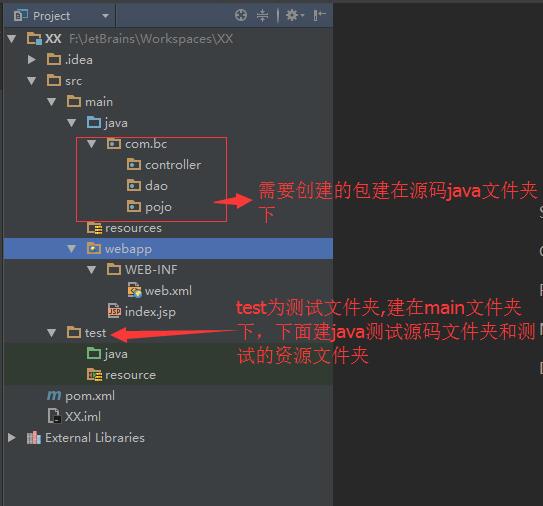


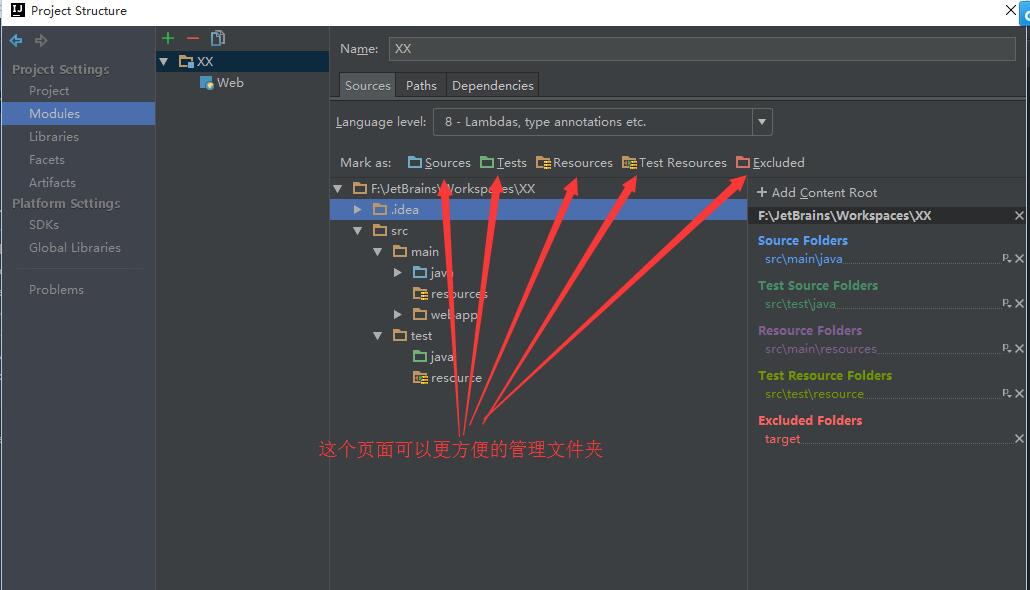
IDEA Maven web 模版项目结构

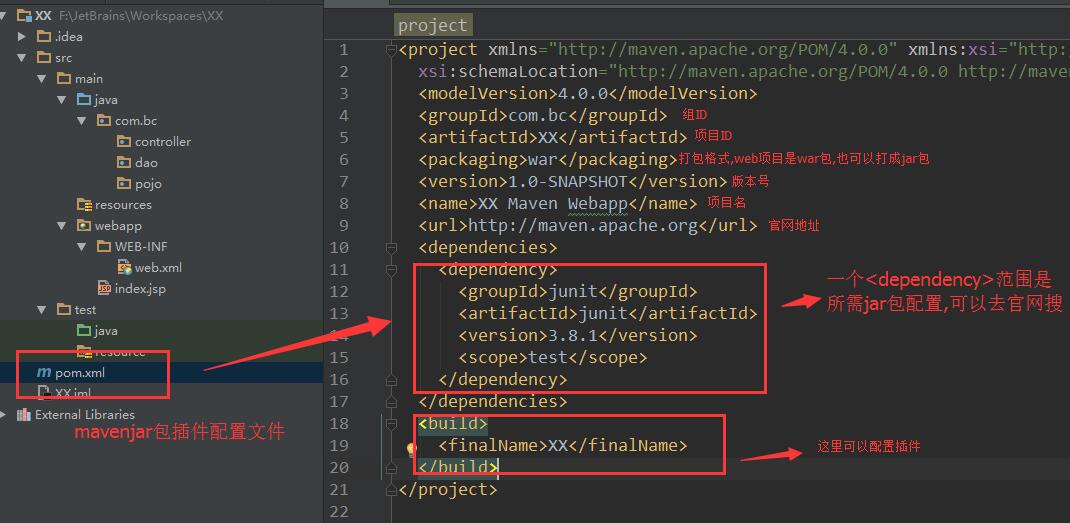




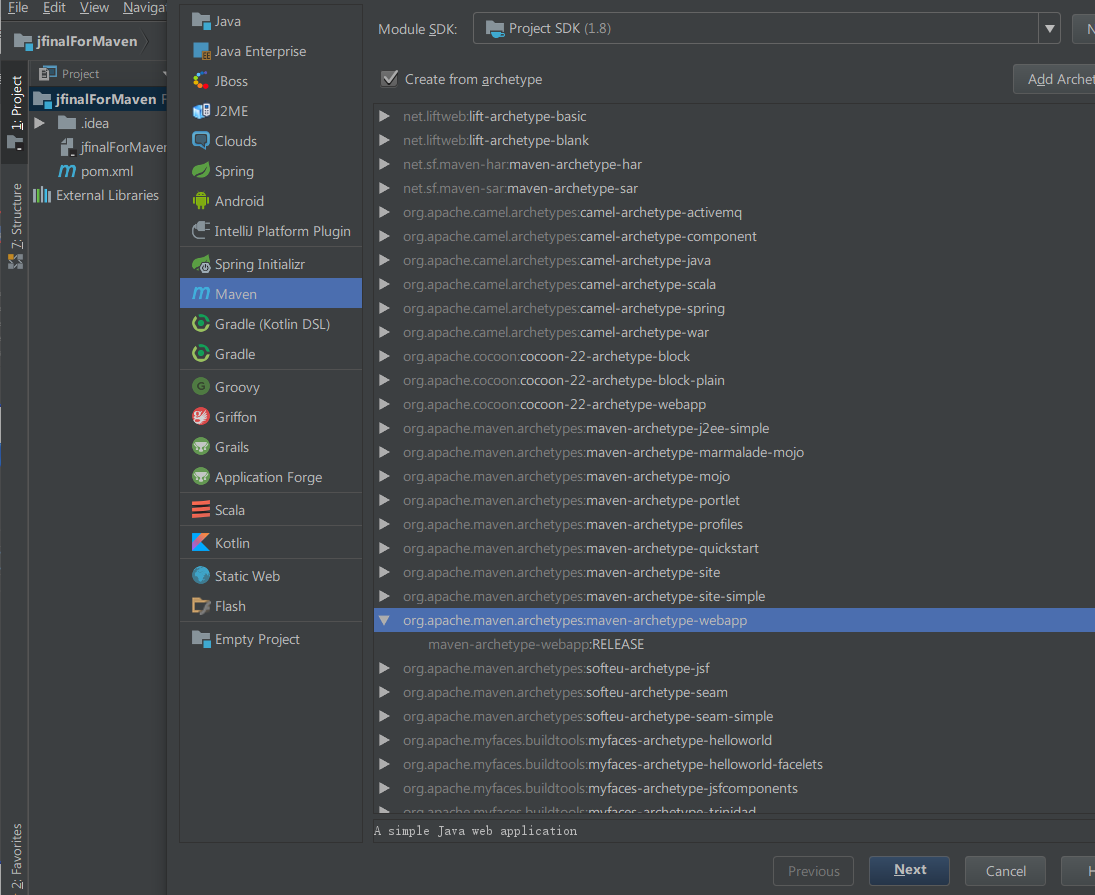








Jfinal2次创建maven使用模板



普通项目转为maven项目

在跟目录下创建pom.xml文件（和src同级目录），并在src目录下新建 **src/main/java**鼠标右键 Mark Directory as -->Sources Root

资源文件(配置)标准目录: **src/main/resources**鼠标右键 Mark Directory as -->Resources Root 再到pom.xml右键add as maven

Maven打包

GITHUB desktop操作

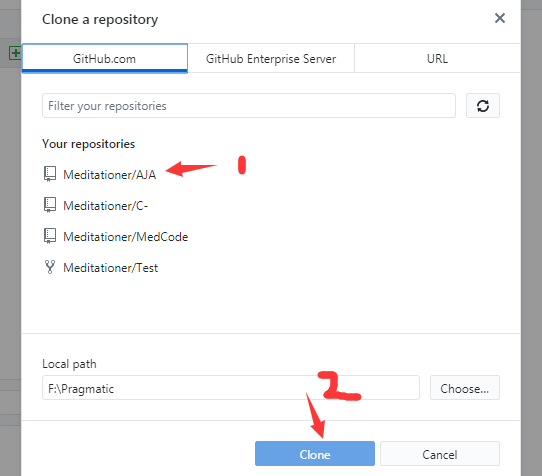
## 安装

百度云解压

## 操作

建立仓库： 1 选择文件夹右击 选择git bash here 输入git init初始化为git仓库 –>页面 –>file -> add ->选择刚才文件夹 –>点击 add repository –>点击fetch origin 更新到 github

File ->clone repository ->



GIT

GIT 安装 百度云（SVN开放源代码的版本控制系统）。一直next，到install的时候两个全选.

## 安装并查看是否安装好

查看安装：Win + R ;Cmd git

配置用户名和邮箱：

查看电脑上是不是配置过的： git config user.name git config user.email看邮箱

配置：电脑右击 git bash

修改(配置)就用git config --global user.name “xxx” git config --global user.email “xxx”

使用ssh 密钥登入 生成ssh命令：

生成ssh：ssh-keygen -t rsa 敲击回车键3次；查看是否生成：系统盘(太难；用everything搜索id\_rsa)



使用ssh 密钥登入 生成ssh命令：

没必要，太麻烦，可以用vscode 和github desktop 解决冲突和提交

Git version 出现版本号就对了

Git init将任何目录转化为 git 版本库

## 建立仓库

在Pragmatic中建立一个文件夹作为远程仓库，在远程仓库中建立一个裸存仓库（名+.git形式）

## 问题

桌面全部图标有蓝色问号。 找到c盘->用户->.git文件夹删除

Vs code

## 问题解决

中文乱码

文件 🡪 首选项 🡪 设置 🡪搜索框 输入：**autoGuessEncoding**

将设置中的**"files.autoGuessEncoding"**项的值改为**true**即可

## VsCode使用git

安装git history：gitlens两个插件 ；安装git终端（百度网盘）