

一、是Apache下的一个开源的项目，是纯java编写，并且只是用来管理java项目

- 统一开发规范与工具
- 统一管理jar包

下载最新版：

在Files下选择如图所示的压缩文件，这是官网推荐的最新版本。

Files

Maven is distributed in several formats for your convenience. Simply pick a ready-made binary distribution archive and follow the [installation instructions](#). Use a source archive if you intend to build Maven yourself. In order to guard against corrupted downloads/installations, it is highly recommended to [verify the signature](#) of the release bundles against the public [KEYS](#) used by the Apache Maven developers.

	Link	Checksums	Signature
Binary tar.gz archive	apache-maven-3.6.2-bin.tar.gz	apache-maven-3.6.2-bin.tar.gz.sha512	apache-maven-3.6.2-bin.tar.gz.asc
Binary zip archive	apache-maven-3.6.2-bin.zip	apache-maven-3.6.2-bin.zip.sha512	apache-maven-3.6.2-bin.zip.asc
Source tar.gz archive	apache-maven-3.6.2-src.tar.gz	apache-maven-3.6.2-src.tar.gz.sha512	apache-maven-3.6.2-src.tar.gz.asc
Source zip archive	apache-maven-3.6.2-src.zip	apache-maven-3.6.2-src.zip.sha512	apache-maven-3.6.2-src.zip.asc

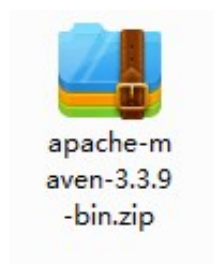
怎么玩：

Maven的核心配置文件 E:\maven\apache-maven-3.3.9-bin\apache-maven-

3.3.9\conf\settings.xml

配置：

下载安装：下载并准备maven安装压缩包



准备本地仓库：下载并准备本地仓库安装压缩包



配置本地仓库：

- 1、将maven的安装包和仓库包解压到一个没有中文路径下的目录中，
- 2、打开Maven的核心配置文件settings.xml

3、修改配置文件中定义的仓库地址为本地仓库地址

```
--
46 <settings xmlns="http://maven.apache.org/SETTINGS/1.0.0"
47     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
48     xsi:schemaLocation="http://maven.apache.org/SETTINGS/1.0.0
        http://maven.apache.org/xsd/settings-1.0.0.xsd">
49   <!-- localRepository
50       The path to the local repository maven will use to store artifacts.
51       Default: ${user.home}/.m2/repository
52   </localRepository>E:\maven\repository</localRepository>
53   -->
54   <!-- interactiveMode
55       This will determine whether maven prompts you when it needs input. If set to false,
56       maven will use a sensible default value, perhaps based on some other setting, for
57       the parameter in question.
58   -->
```

修改此处为本地仓库的地址

maven国内镜像

```
<?xml version="1.0" encoding="UTF-8"?>
<settings xmlns="http://maven.apache.org/SETTINGS/1.0.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/SETTINGS/1.0.0
http://maven.apache.org/xsd/settings-1.0.0.xsd">
  <mirrors>
    <!-- 阿里云仓库 -->
    <mirror>
      <id>alimaven</id>
      <mirrorOf>central</mirrorOf>
      <name>aliyun maven</name>
      <url>http://maven.aliyun.com/nexus/content/repositories/central/</url>
    </mirror>

    <!-- 中央仓库1 -->
    <mirror>
      <id>repo1</id>
      <mirrorOf>central</mirrorOf>
      <name>Human Readable Name for this Mirror.</name>
      <url>http://repo1.maven.org/maven2/</url>
    </mirror>

    <!-- 中央仓库2 -->
    <mirror>
      <id>repo2</id>
      <mirrorOf>central</mirrorOf>
      <name>Human Readable Name for this Mirror.</name>
      <url>http://repo2.maven.org/maven2/</url>
    </mirror>
  </mirrors>
</settings>
```

<mirrors>

<!-- 阿里云仓库 -->

<mirror>

<id>alimaven</id>

<mirrorOf>central</mirrorOf>

<name>aliyun maven</name>

<url>http://maven.aliyun.com/nexus/content/repositories/central/</url>

</mirror>

<!-- 中央仓库1 -->

<mirror>

<id>repo1</id>

<mirrorOf>central</mirrorOf>

<name>Human Readable Name for this Mirror.</name>

<url>http://repo1.maven.org/maven2/</url>

</mirror>

<!-- 中央仓库2 -->

<mirror>

<id>repo2</id>

<mirrorOf>central</mirrorOf>

<name>Human Readable Name for this Mirror.</name>

<url>http://repo2.maven.org/maven2/</url>

</mirror>

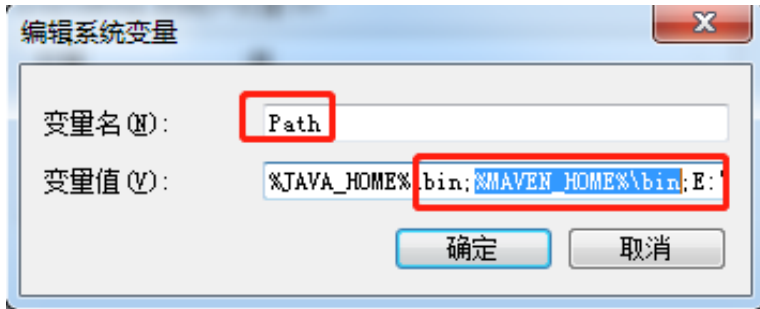
</mirrors>

配置环境变量:

1、在系统变量环境中新建一个MAVEN_HOME, 指定MAVEN的解压路径



2、将加入系统环境中



验证maven环境是否配置成功:

在cmd控制台 执行mvn -v命令

控制台显示以下内容说明配置成功

```
命令提示符
Microsoft Windows [版本 10.0.14393]
(c) 2016 Microsoft Corporation。保留所有权利。

C:\Users\syl>mvn -v
Apache Maven 3.3.9 (bb52d8502b132ec0a5a3f4c09453c07478323dc5; 2015-11-11T00:41:47+08:00)
Maven home: D:\class37\apache-maven-3.3.9\bin\..
Java version: 1.7.0_72, vendor: Oracle Corporation
Java home: D:\ITCAST\jdk1.7.0_72\jre
Default locale: zh_CN, platform encoding: GBK
OS name: "windows 8.1", version: "6.3", arch: "x86", family: "windows"
```

maven的三种仓库

本地仓库—————部署在本地电脑上的

远程仓库（私服）—————公司或维护的一个私有有仓库

中央仓库—————由Maven团队维护

Maven中pom.xml（项目对象模型）中的一些配置说明:

1、modelVersion

指定了当前Maven模型的版本号，对于Maven2和Maven3来说，它只能是4.0.0

2、groupId

顾名思义，这个应该是公司名或是组织名。

3、artifactId

可以认为是Maven构建的项目名

4、version

版本号，SNAPSHOT意为快照，说明该项目还在开发中，是不稳定的版本。

5、packing

项目打包的类型，可以使jar、war、rar、ear、pom，默认是jar

6、dependencies和dependency

前者包含后者。前面说了，Maven的一个重要作用就是统一管理jar包，为了一个项目可以build或运行，项目中不可避免的，会依赖很多其他的jar包，在Maven中，这些依赖就被称为dependency。

7、scope的分类

1.compile：默认值 他表示被依赖项目需要参与当前项目的编译，还有后续的测试，运行周期也参与其中，是一个比较强的依赖。打包的时候通常需要包含进去

2.test：依赖项目仅仅参与测试相关的工作，包括测试代码的编译和执行，不会被打包，例如：junit

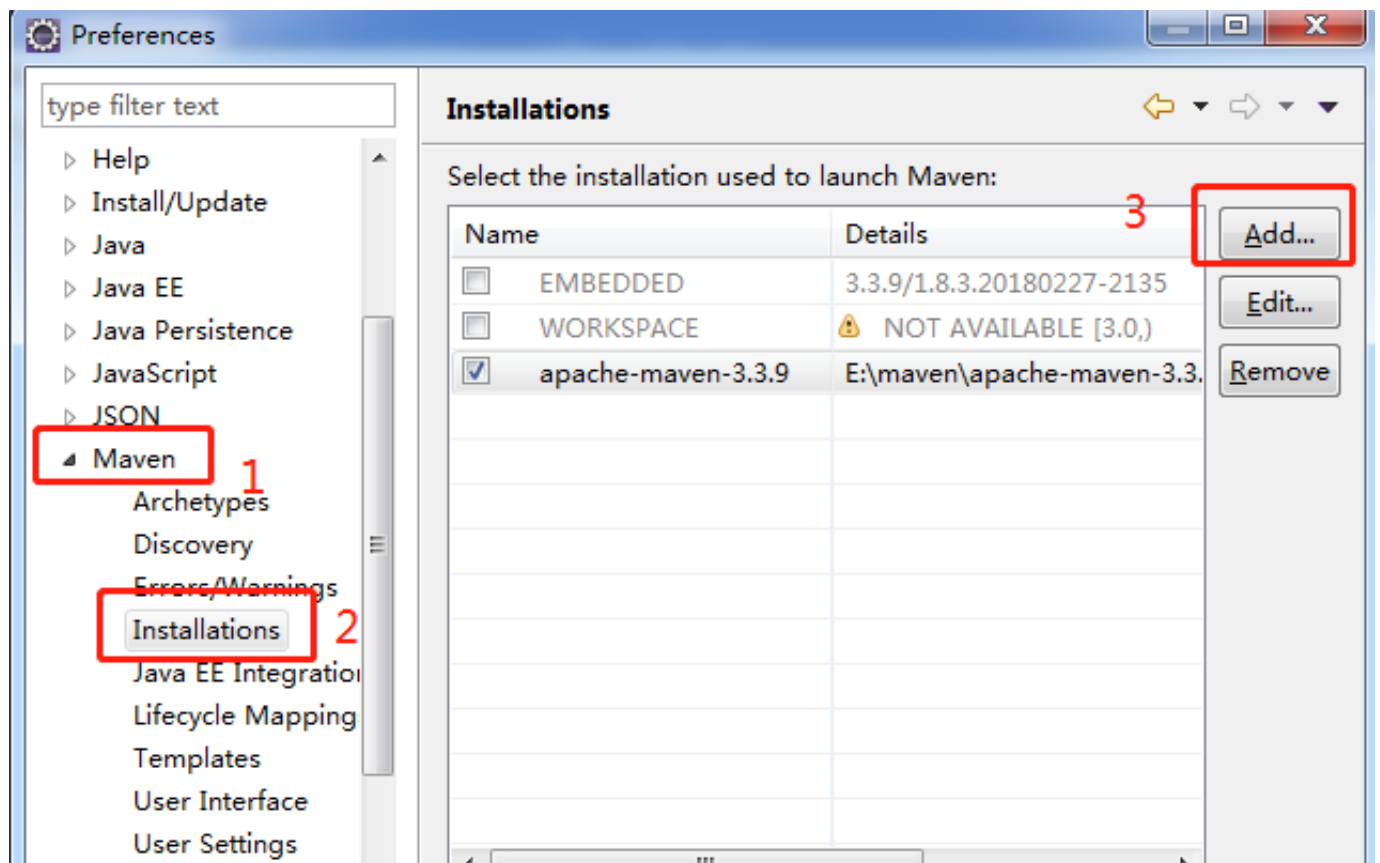
3.runtime：表示被依赖项目无需参与项目的编译，不过后期的测试和运行周期需要其参与。与compile相比，跳过了编译而已。例如JDBC驱动，适用运行和测试阶段

4.provided：打包的时候可以不用包进去，别的设施会提供。事实上该依赖理论上可以参与编译，测试，运行等周期。相当于compile，但是打包阶段做了exclude操作

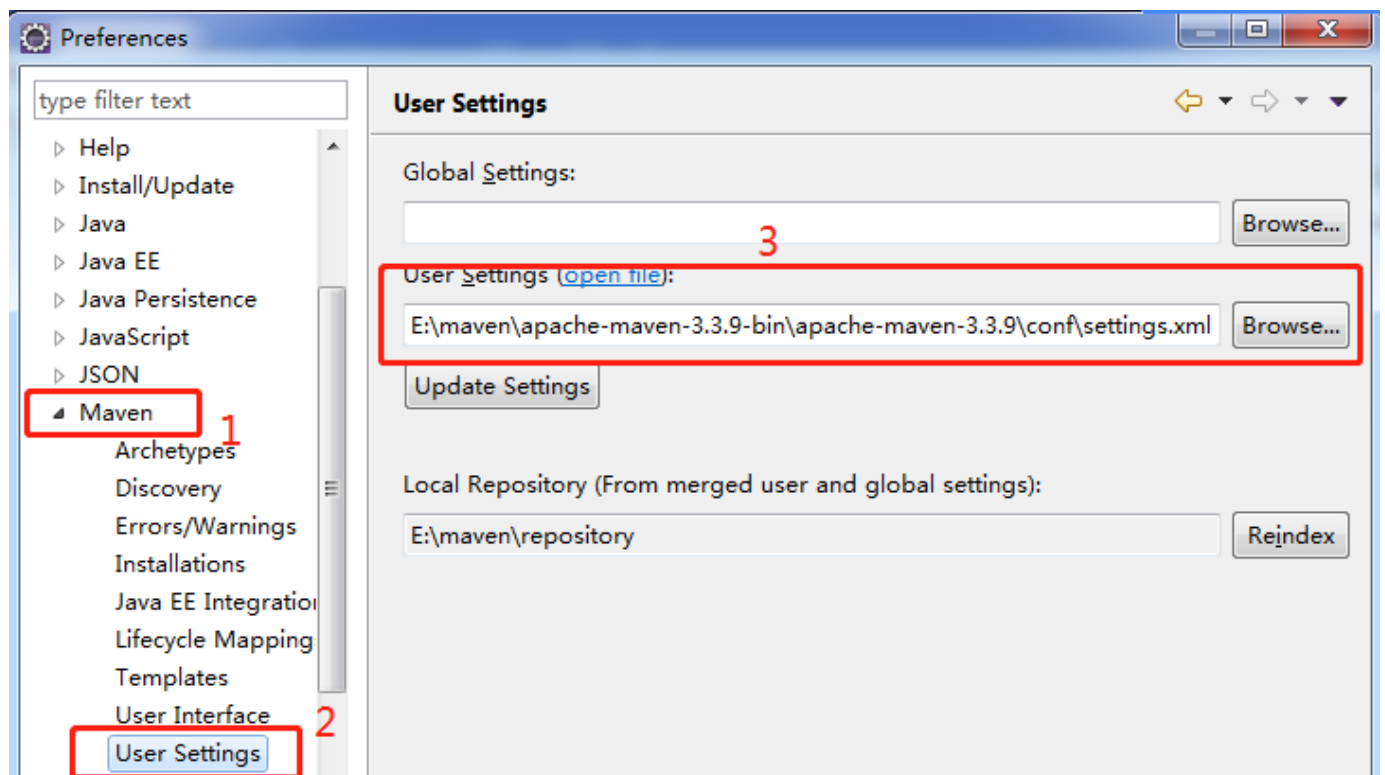
5.system：从参与度来说，和provided相同，不过被依赖项不会从maven仓库下载，而是从本地文件系统拿。需要添加systemPath的属性来定义路径

集成Maven

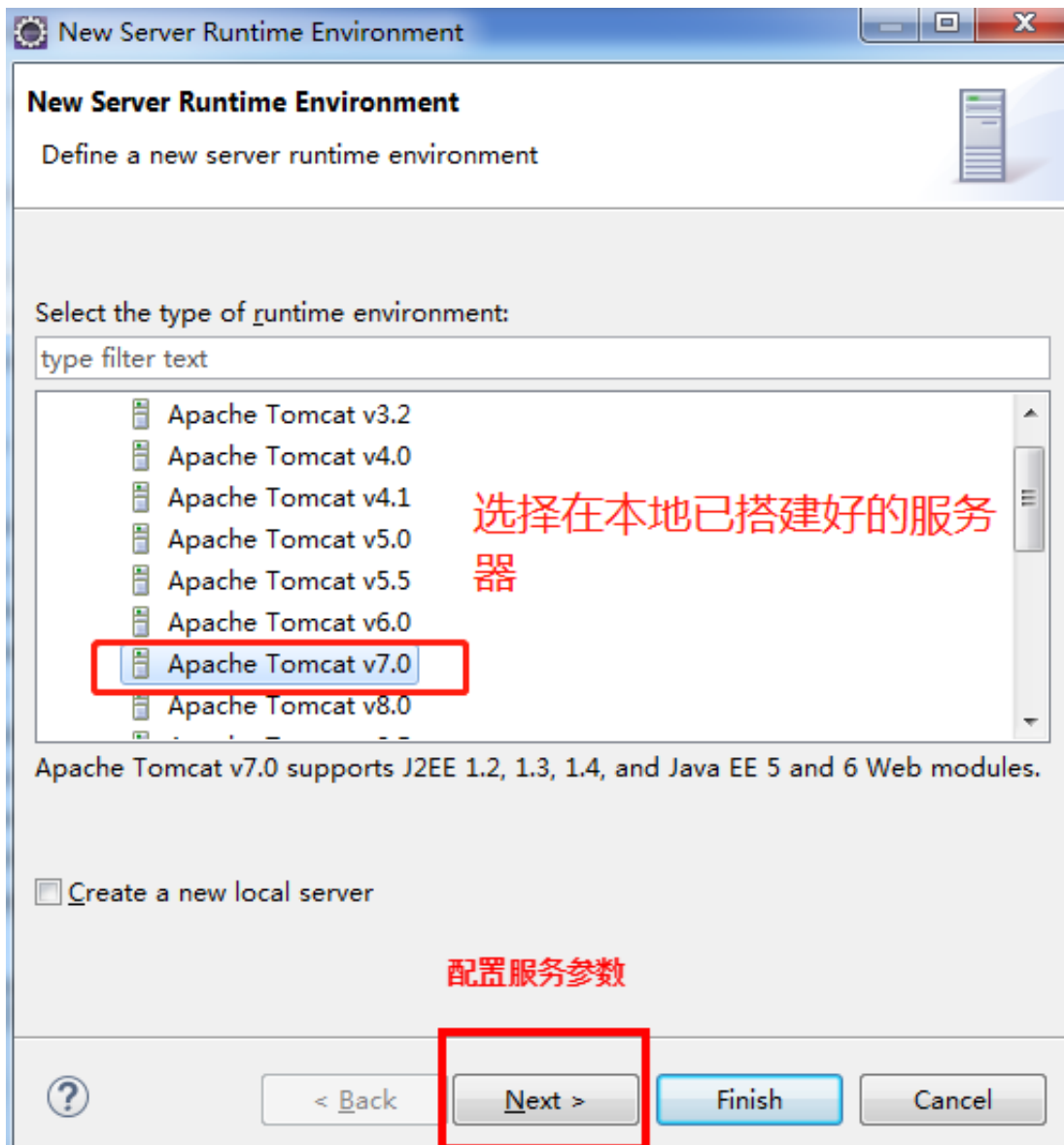
1、集成maven，找到安装的目录添加 E:\maven\apache-maven-3.3.9-bin\apache-maven-3.3.9



- 2、指定MAVEN的主配置文件的位置E:\maven\apache-maven-3.3.9-bin\apache-maven-3.3.9\conf\settings.xml



- 3、配置本地的tomcat服务器



New Server Runtime Environment

Tomcat Server

Specify the installation directory

Name:
Apache Tomcat v7.0 (2)

Tomcat installation directory:
C:\TomcatServer\apache-tomcat-7.0.85 指定tomcat的安装位置

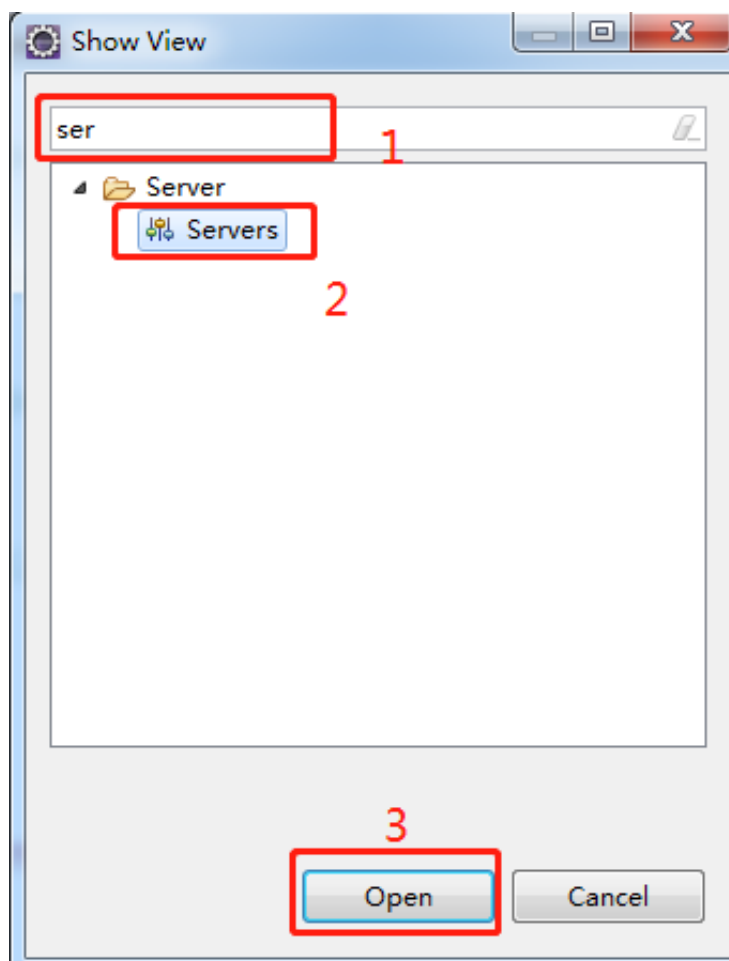
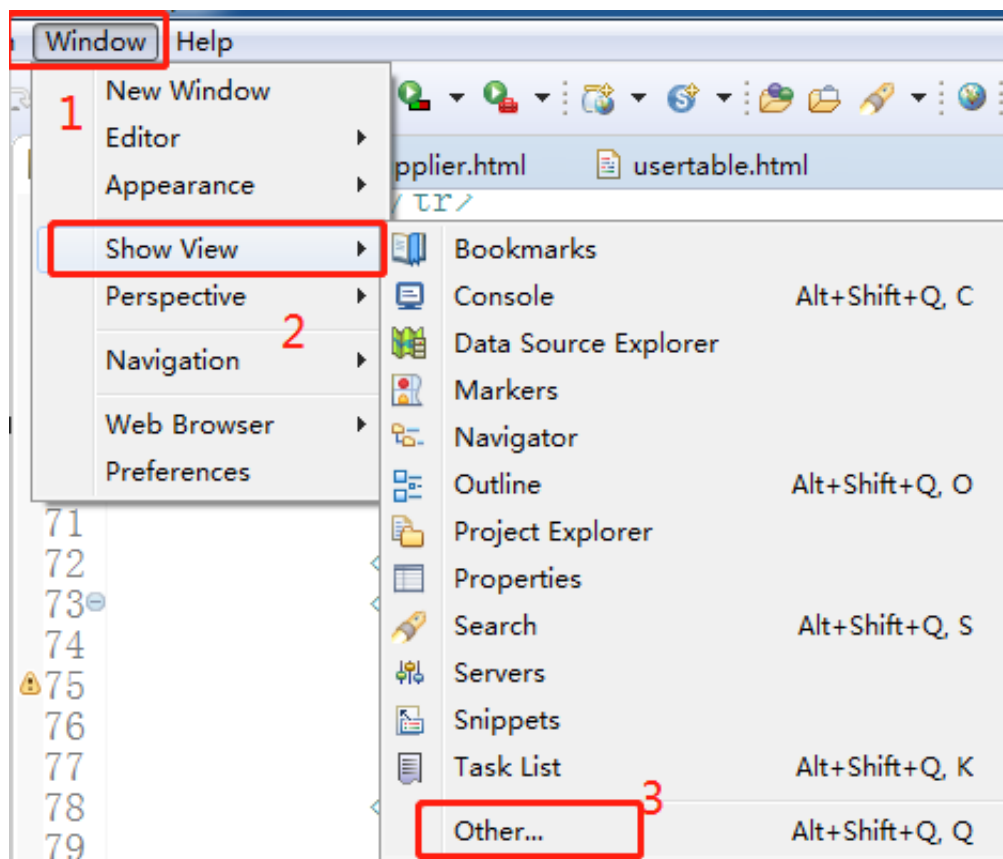
apache-tomcat-7.0.47 Download and Install...

JRE:
Workbench default JRE 指定本地的JDK

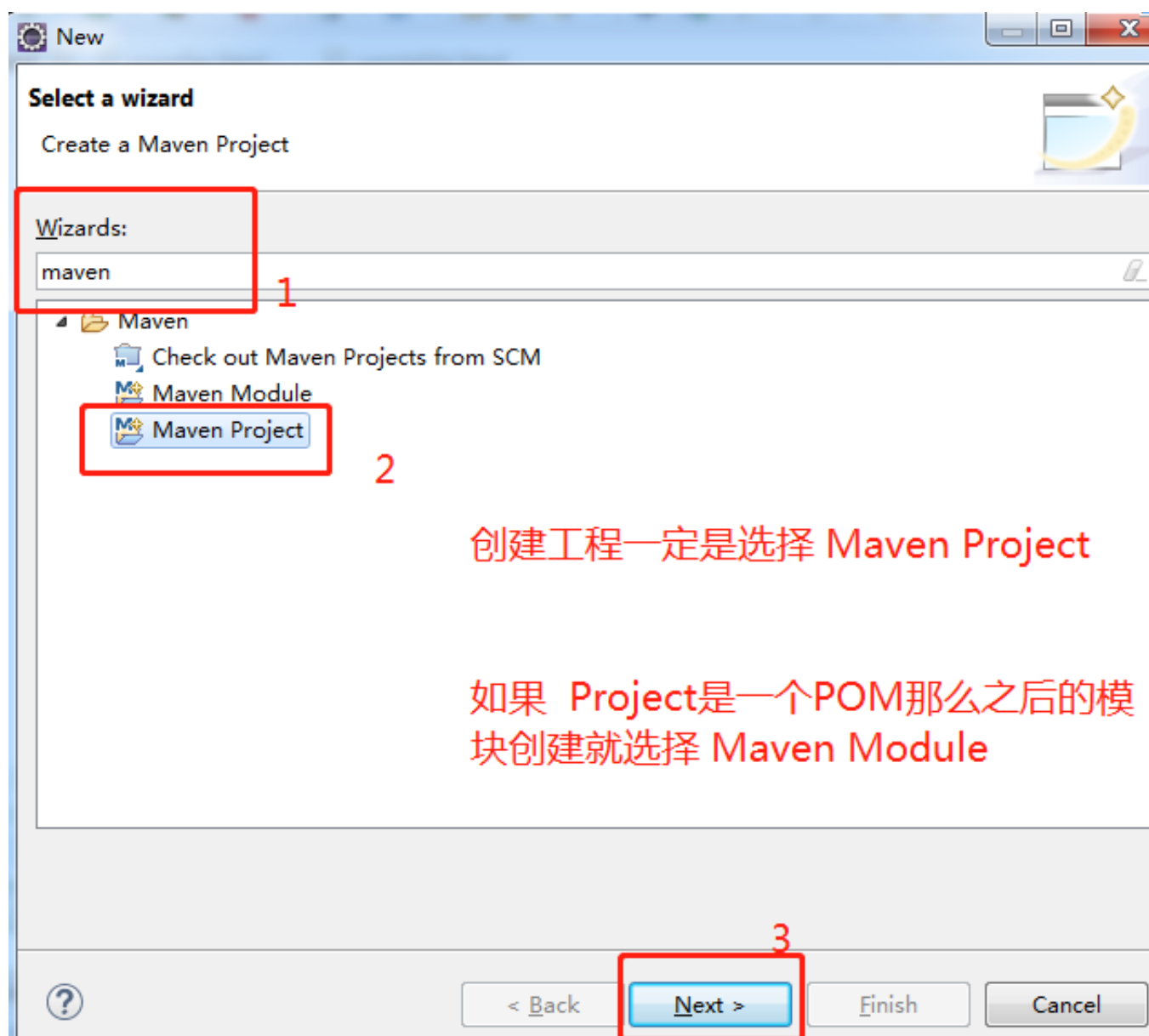
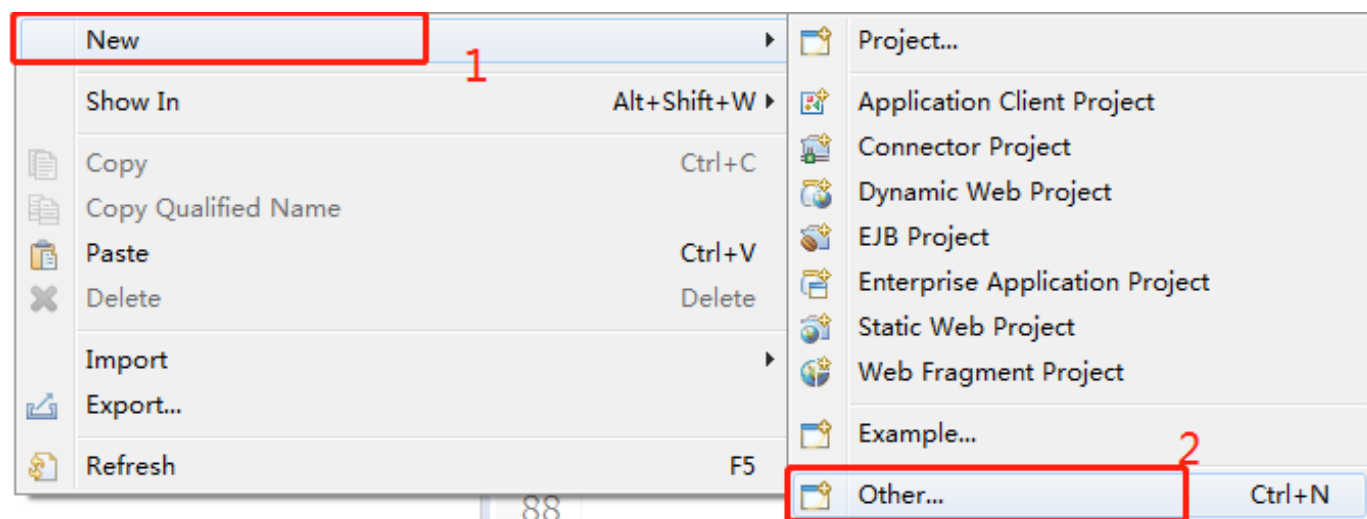
Installed JREs...

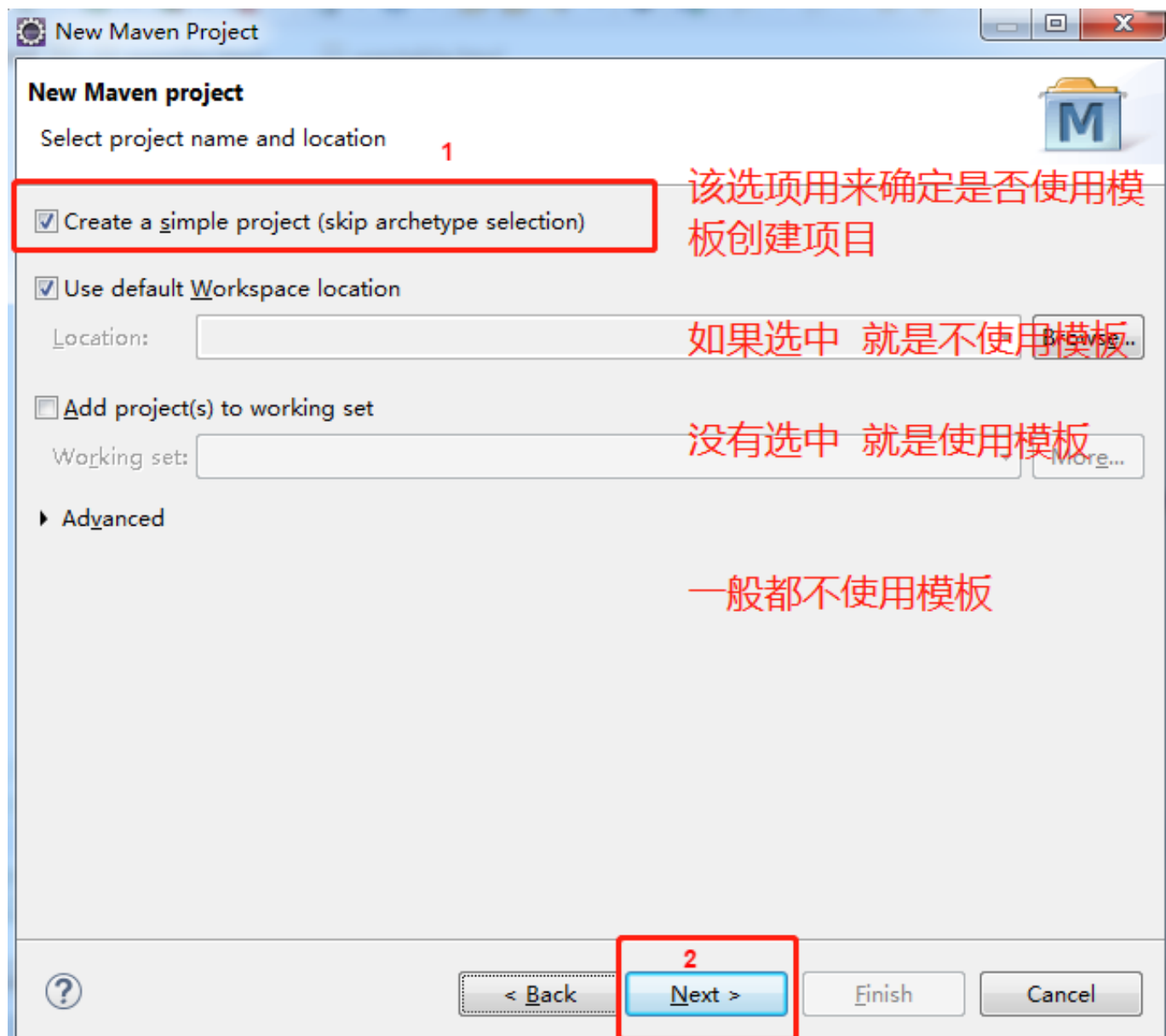
? < Back Next > Finish Cancel

显示服务



4、使用模板或不使用模板创建工程（[创建项目](#)）





New Maven Project

New Maven project

Enter a group id for the artifact.

Artifact

Group Id: 企业名 英文标识

Artifact Id: 项目名

Version: 0.0.1-SNAPSHOT

Packaging: 项目结构确定

Name:

Description:

Parent Project

Group Id:

Artifact Id: jar : java打包方式
一般如dao,service,实体类包都选择jar

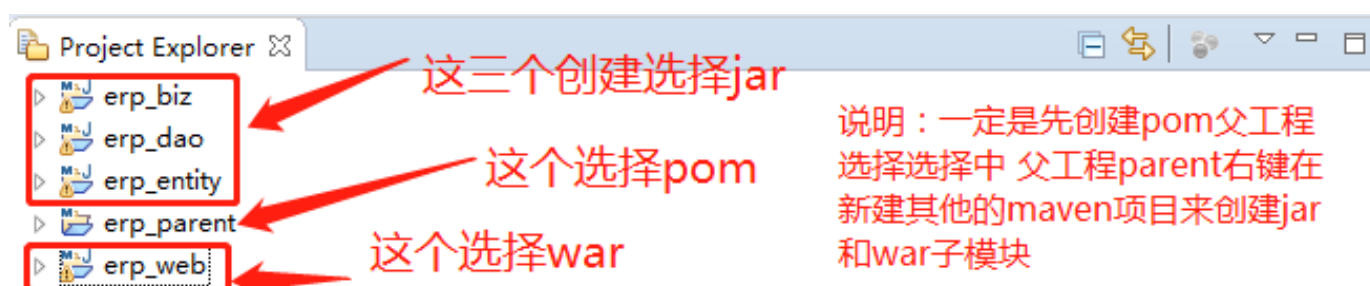
Version:

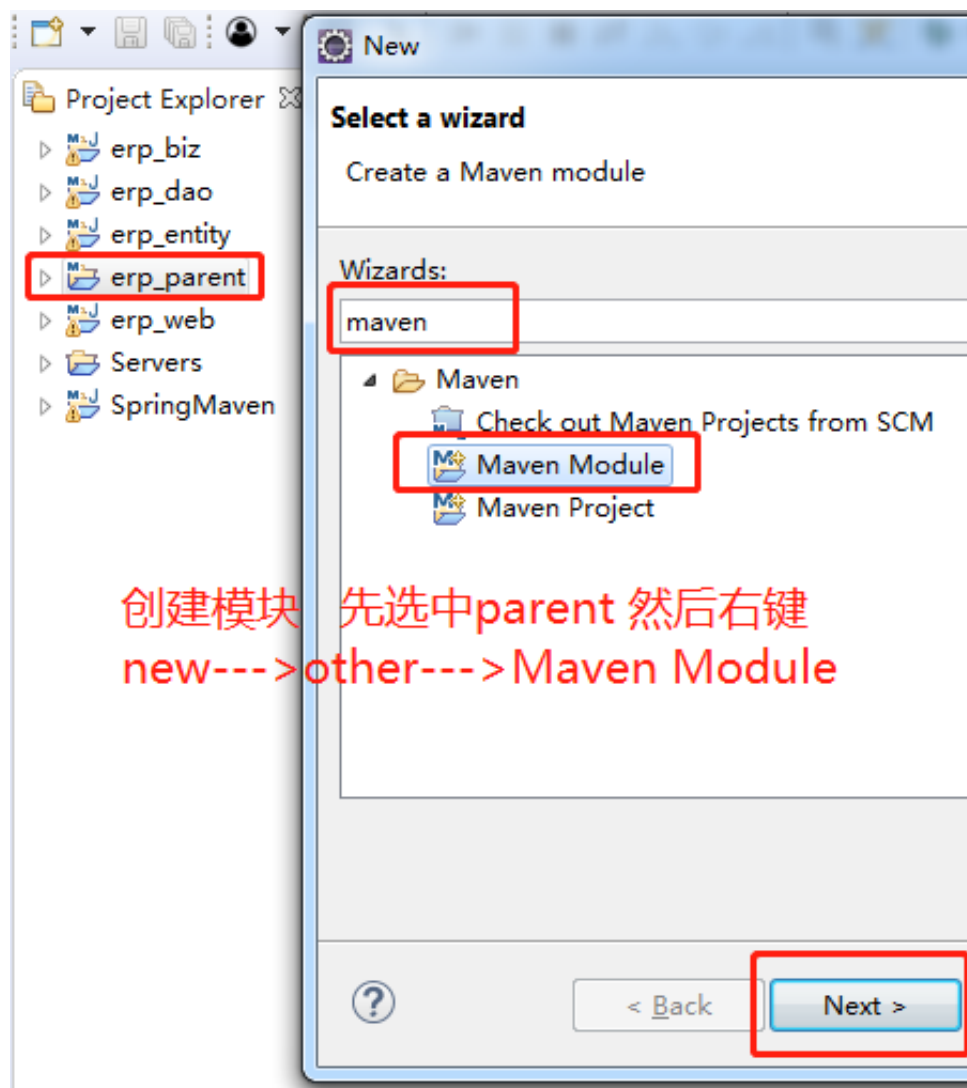
Advanced

war : web打包方式

< Back Next > Finish Cancel

带父目录工程的创建结构





New Maven Module

Enter a module name.

☒ Create a simple project (skip archetype selection) 1 选中

Module Name: 2 输入模块名 如 : dao

Parent Project: Browse...

☒ Add project(s) to working set

Working set: More...

Advanced

3

Artifact

Group Id:

Artifact Id:

Version:

Packaging: 只需要改打包方式

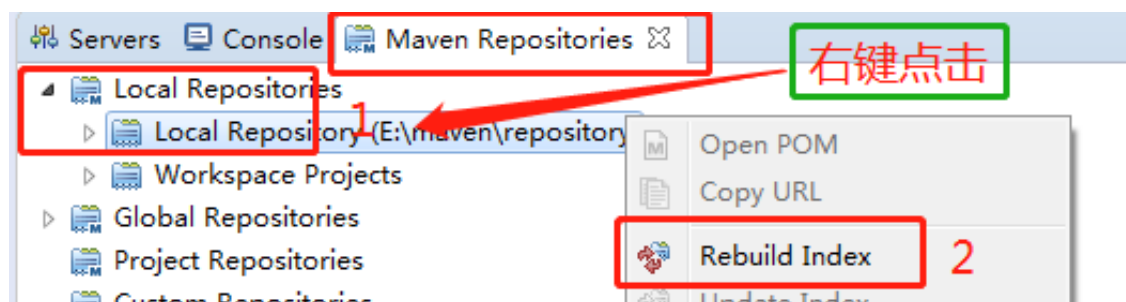
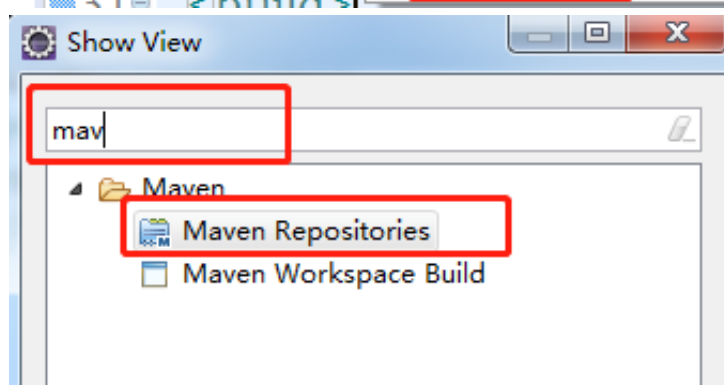
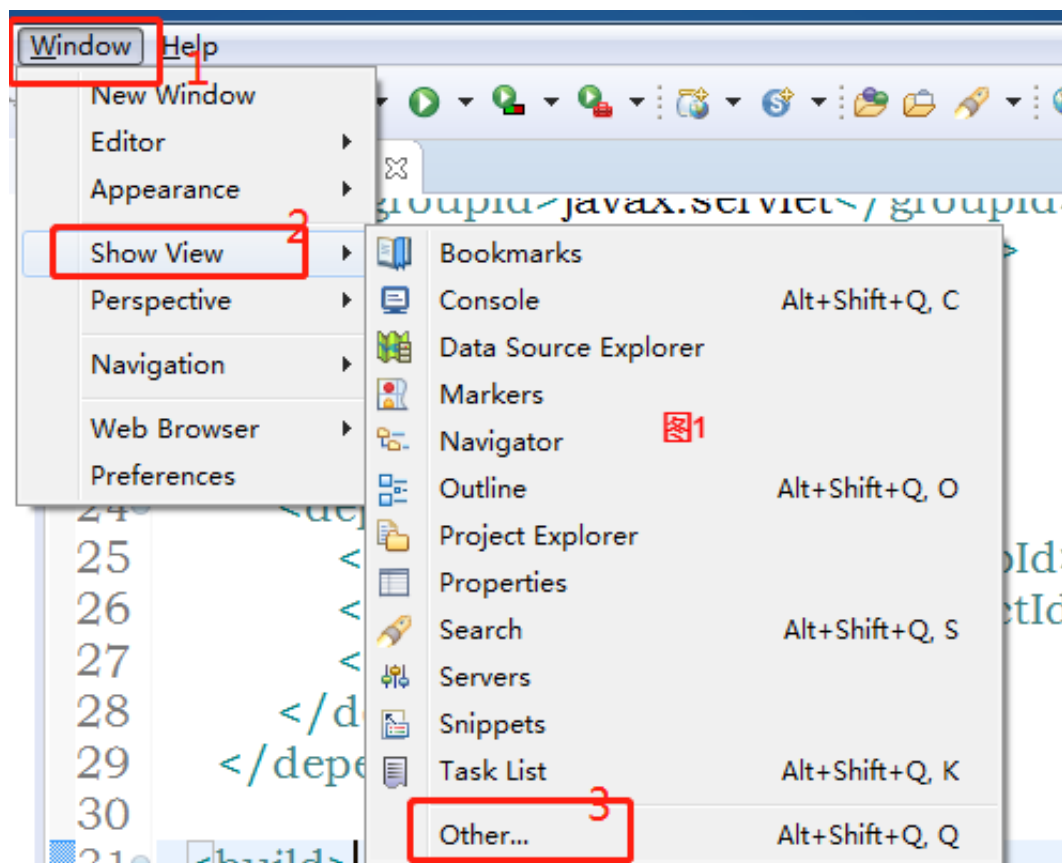
Name:

Description:

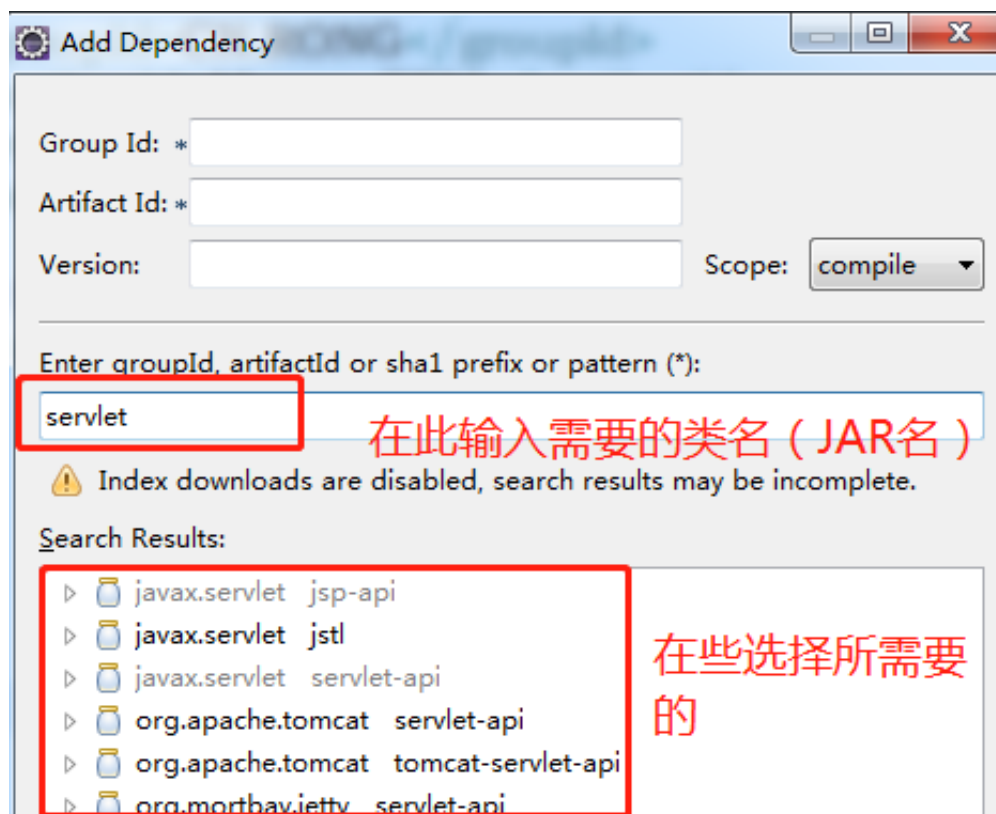
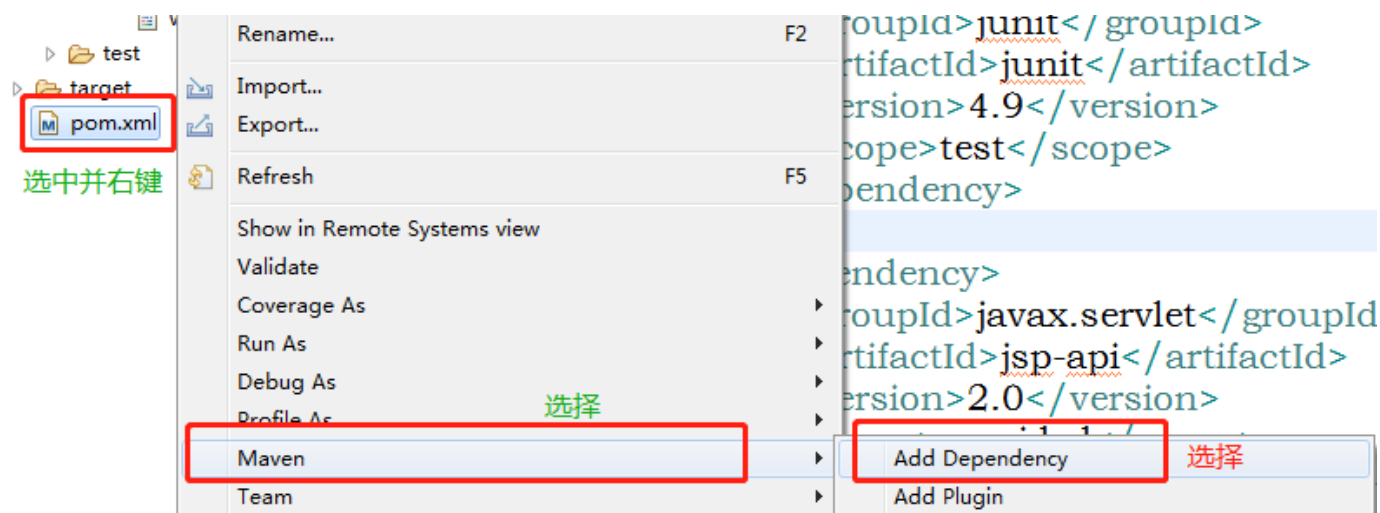
5、导相应的JAR包依赖

第一次搭建一定要干的事，搭建好以后就不管了（告诉项目依赖的JAR在哪里去找）

Show View---->输入maven---->选择Maven Repositories---->Local Repositories---->Rebuild Index---->OK



POM.xml配置: Add Dependency----->输入需要的依赖



最终可以使用ERP的**代码生成器**完成Maven的创建

1、这个工具需要放在一个没有中文的目录下才可以使用

2、它是根据数据库中的表来生成项目的，所以生成代码前必须保证数据库有要使用的数据表