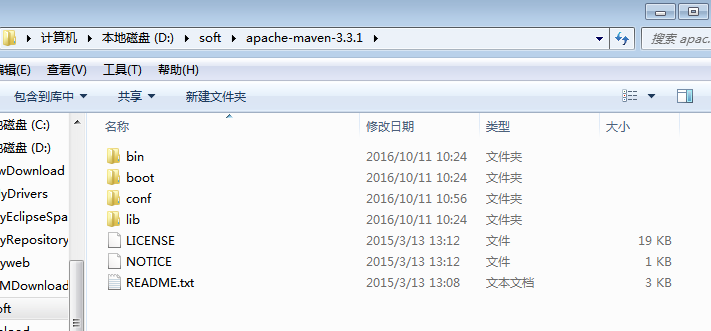
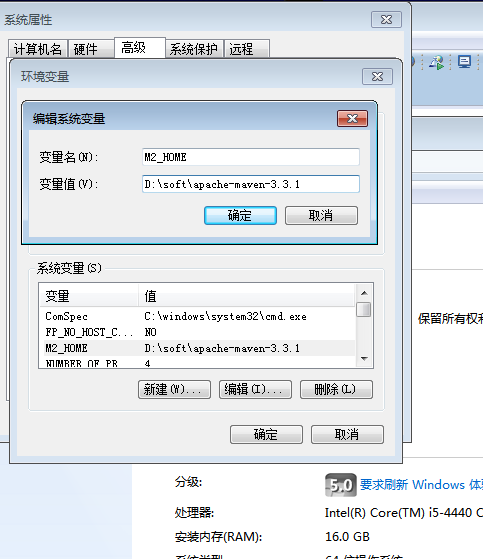
# O1、maven配置

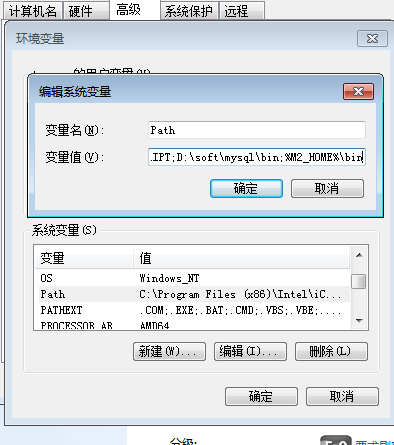
# 1 maven的安装与配置

## 1.1 下载maven 3.3.1 版本 并解压

安装后需要配置环境变量，如下所示：



Path中添加 %M2\_MOME%\bin



## 1.2 修改settings.xml

### 1.2.1添加本地仓库

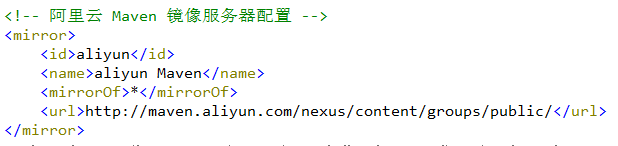
打开conf/setting.xml文件，修改其中的localRpository

路径地址为自己的本地仓库地址

### 

### 1.2.2 添加镜像服务器

修改settings.xml文件, 添加达内镜像服务器配置信息:



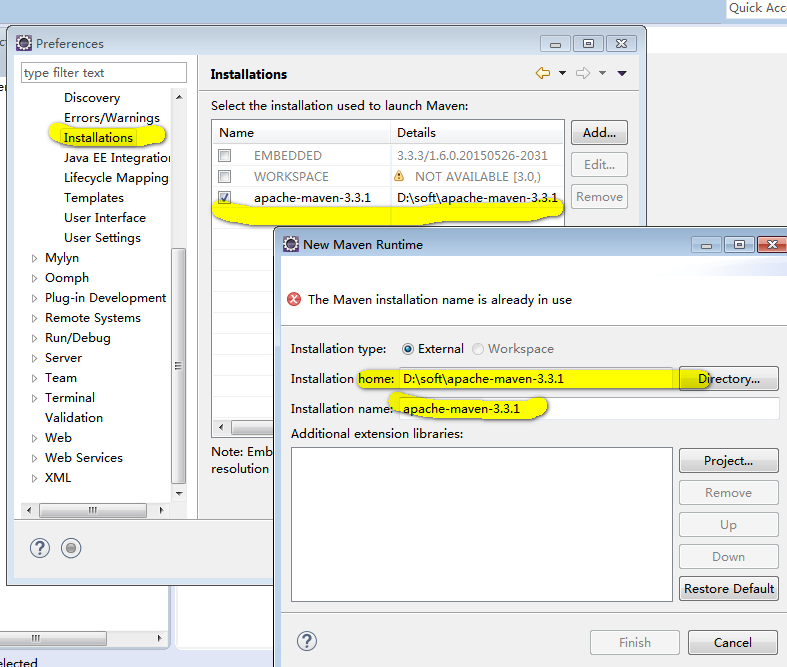
阿里云镜像服务器URL: <http://maven.aliyun.com/nexus/content/groups/public/>

<**mirror**>  
 <**id**>nexus</**id**>  
 <**name**>Tedu Maven</**name**>  
 <**mirrorOf**>\*</**mirrorOf**>  
 <**url**>http://maven.tedu.cn/nexus/content/groups/public/</**url**>  
</**mirror**>

达内镜像服务器URL: <http://maven.tedu.cn/nexus/content/groups/public/>

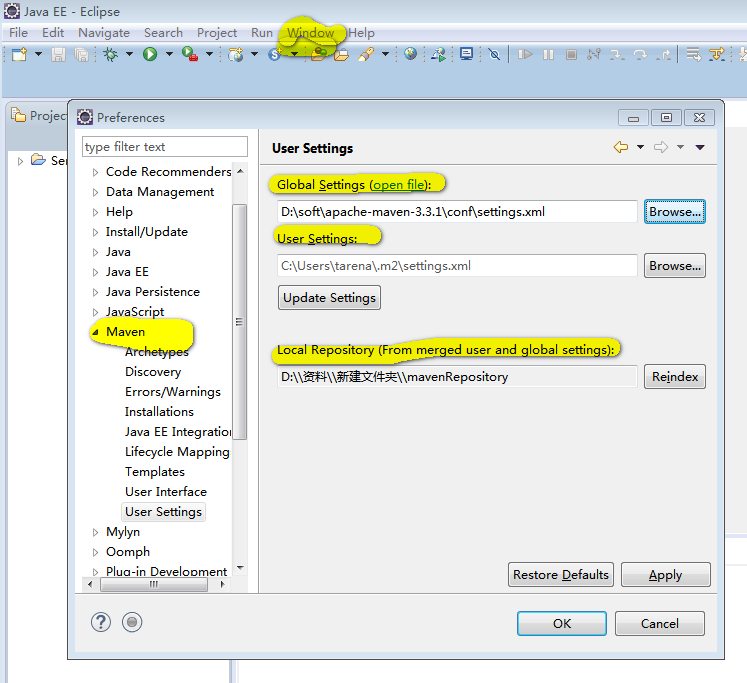
## 1.3打开eclipse window下的 Preference下的maven 下的Installations

添加刚刚配置的maven

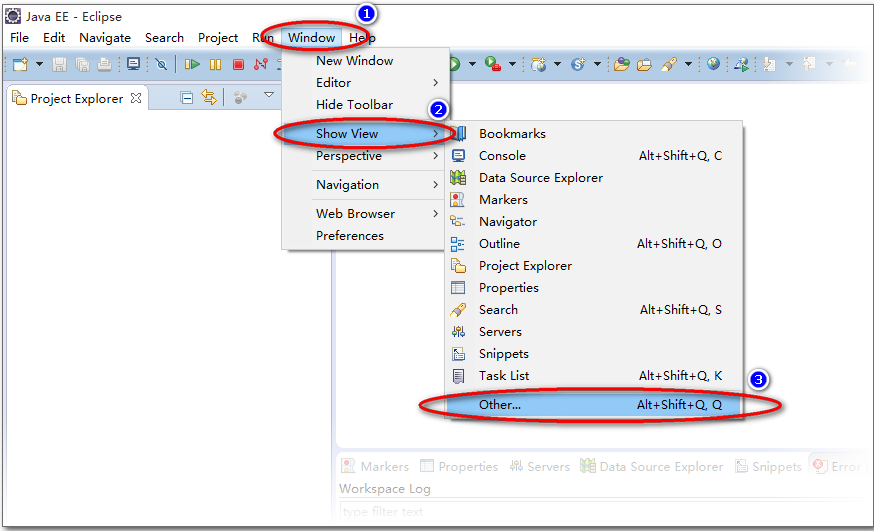


## 1.4打开eclipse window下的 Preference下的maven 的UserSettings

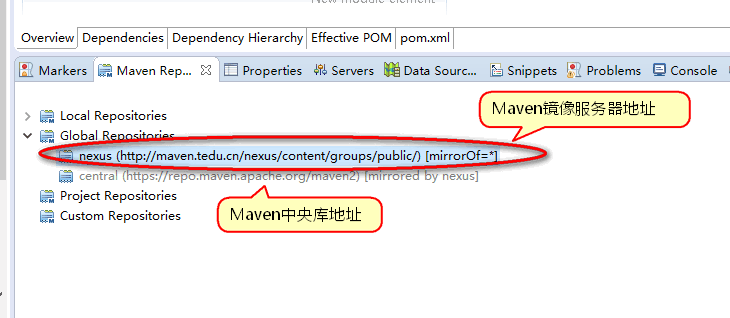
Global Settings为 maven的全局配置文件，路径为上面配置的setting的路径



## 1.5找到window菜单

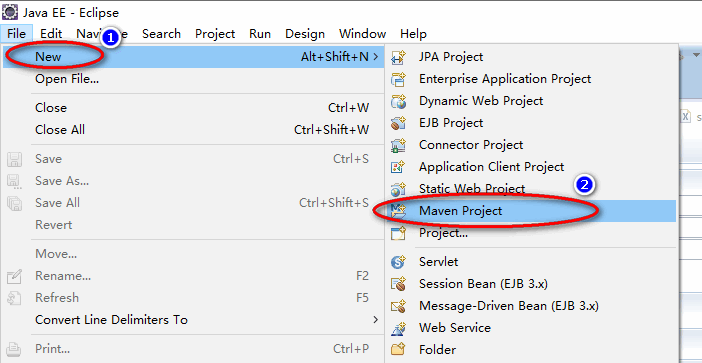


## 1.6开启Maven的视图, 检查是否已经配置了镜像库

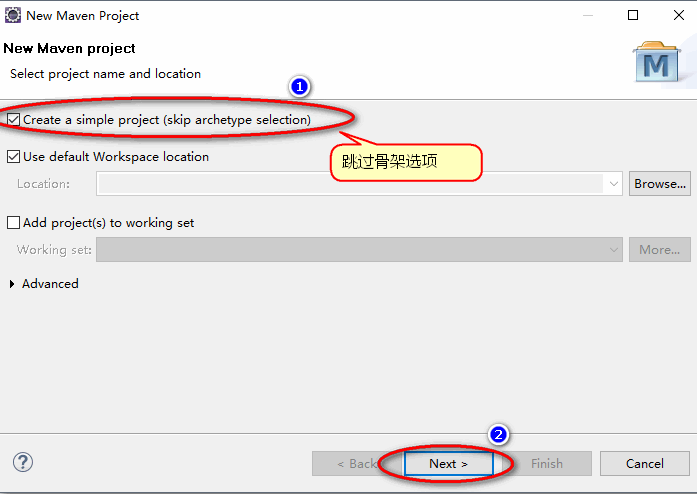


# 2 maven创建jar工程

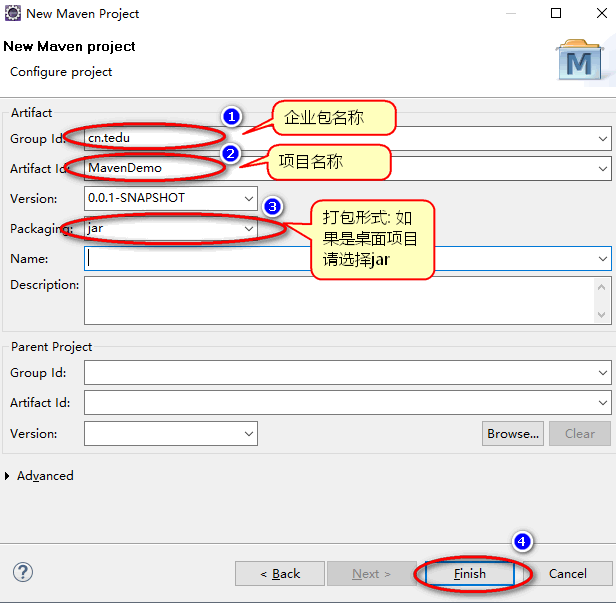
## 2.1选择菜单创建Maven项目



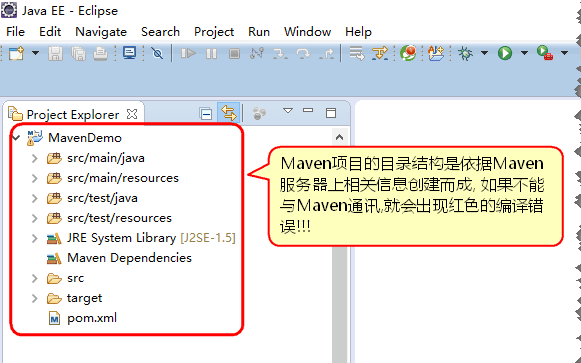
## 2.2 选择项目目录结构的骨架



## 2.3 输入项目相关信息



## 2.4maven jar的目录结构

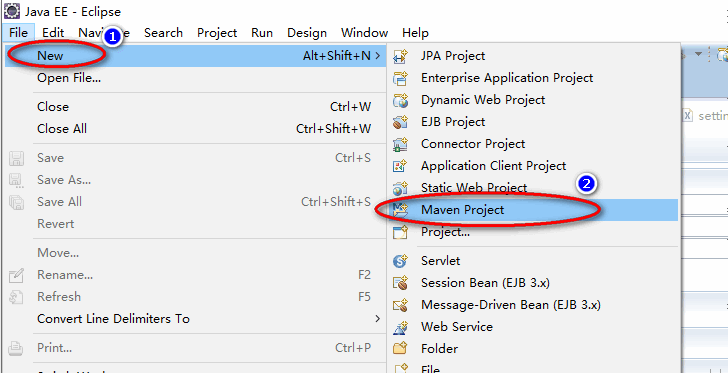


## 2.5在taotao-manager工程的pom文件中添加如下内容：

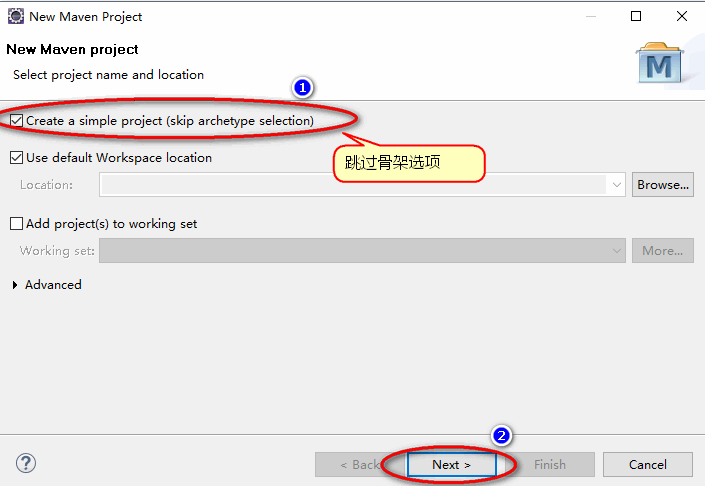
|  |
| --- |
| <build>  <!-- 配置插件 -->  <plugins>  <plugin>  <groupId>org.apache.tomcat.maven</groupId>  <artifactId>tomcat7-maven-plugin</artifactId>  <configuration>  <port>8080</port>  <path>/</path>  </configuration>  </plugin>  </plugins>  </build> |

# 3 创建Maven Web项目

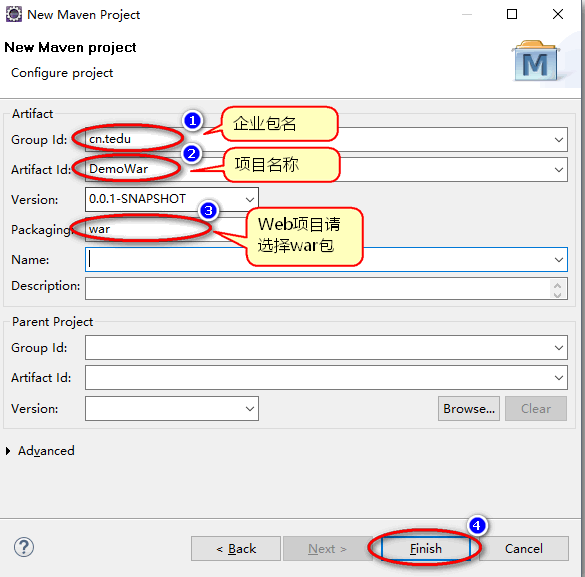
## 3.1选择菜单创建Maven项目



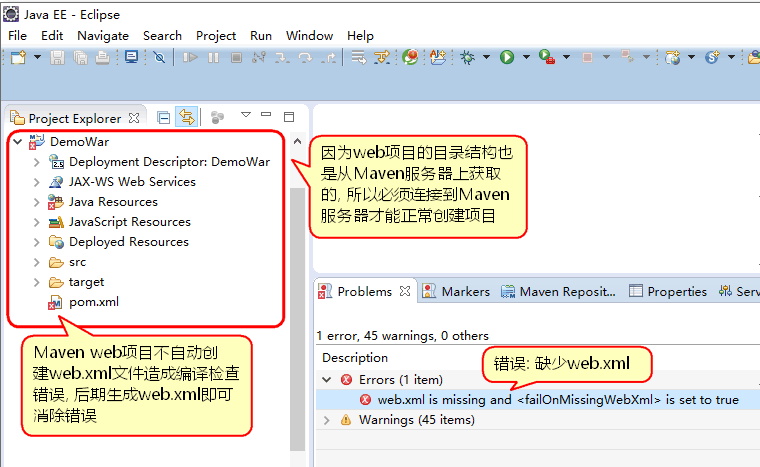
## 3.2选择项目目录结构的骨架



## 3.3输入项目相关信息

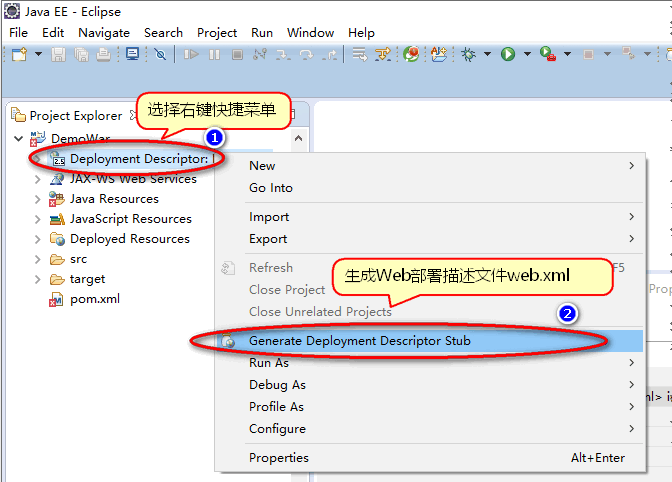


## 3.4 Maven war项目的目录结构



因为这个步骤必须要链接到maven服务器获取相关信息，如果配置配置和网路有故障则这个步骤就会出现错误，如果出现错误请使用浏览器监测能否访问Maven服务器的url

## 3.5 创建web部署描述符web.xml消除检测错误



## 3.6 maven中xml文件的加载

在idea中的maven项目的java文件夹下的xml文件默认时不加载的，只会加载resources中的文件

所以需要在pom文件的build中配置

|  |
| --- |
| <**resources**>  <**resource**>  <**directory**>src/main/java</**directory**>  <**includes**>  <**include**>\*\*/\*.xml</**include**>  </**includes**>  </**resource**>  <**resource**>  <**directory**>src/main/resources</**directory**>  <**includes**>  <**include**>\*\*/\*.xml</**include**>  <**include**>\*\*/\*.properties</**include**>  </**includes**>  </**resource**> </**resources**> |

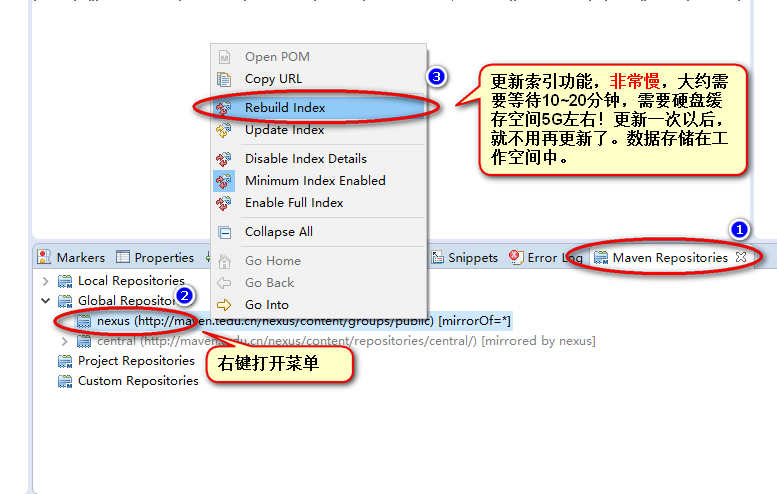
## 3.7 java编译插件

|  |
| --- |
| <**plugin**>  <**groupId**>org.apache.maven.plugins</**groupId**>  <**artifactId**>maven-compiler-plugin</**artifactId**>  <**version**>3.2</**version**>  <**configuration**>  <**source**>1.7</**source**>  <**target**>1.7</**target**>  <**encoding**>utf-8</**encoding**>  </**configuration**> </**plugin**> |

# 4 在Eclipse中搜索添加Jar组件"坐标"

## 4.1 更新Maven库的索引文件

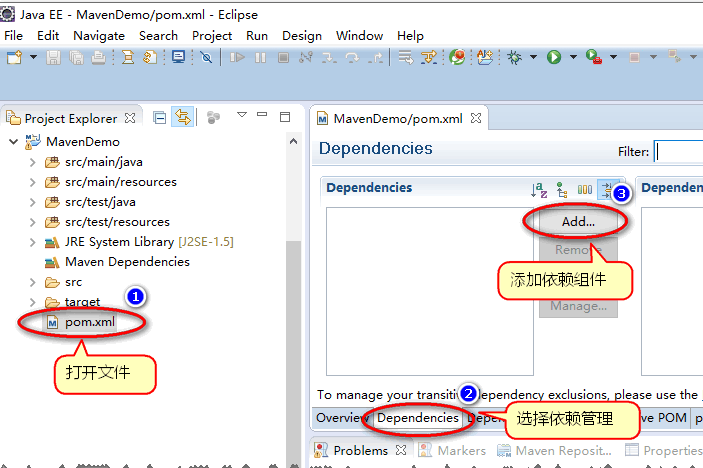
### Mavne索引用于搜索jar组件"坐标", 更新索引以后就可以在Eclipse中搜索jar组件"坐标", 但这个功能不是必须的, 使用 4. 在线搜索添加Jar组件"坐标"



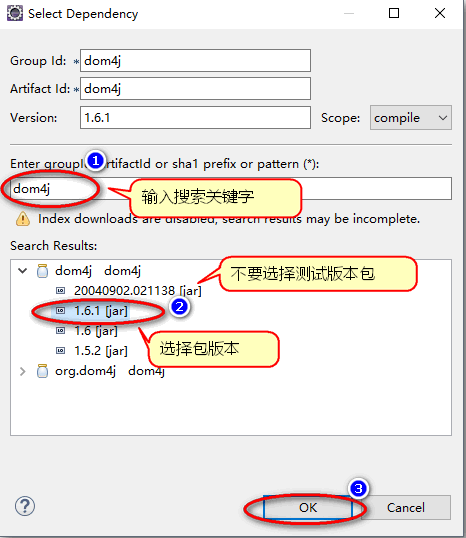
**Eclipse工作空间中已经更新过Maven索引, 一般无需更新这个索引**, 但是如果更换过Eclipse工作空间, 就需要再次更新Maven索引, 如果无法更新索引, 使用 [4. 在线搜索添加Jar组件](http://doc.tedu.cn/maven/" \l "online).

阿里云 Maven 镜像服务器没有提供Mavne索引功能, 无法下载Maven索引! 使用 [4. 在线搜索添加Jar组件](http://doc.tedu.cn/maven/" \l "online).

## 4.2 打开maven中的 pom.xml

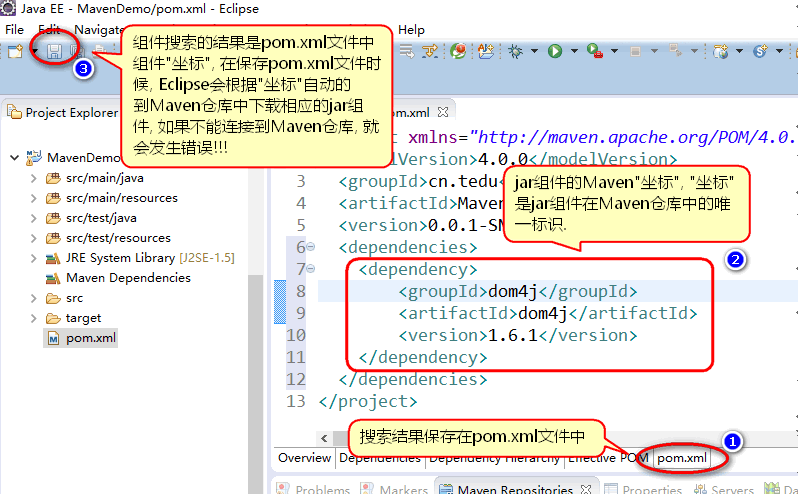


## 4.3搜索需要jar组件包"坐标"



这个搜索功能是利用 [3.1 更新Maven库的索引文件](http://doc.tedu.cn/maven/" \l "update-index)的索引完成, 如果没有更新过索引, 这个功能失效!

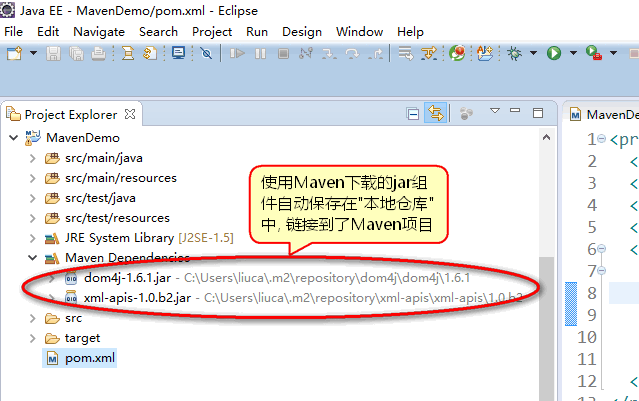
## 4.4搜索结果就是在pom.xml文件中"坐标", 保存pom.xml时候会自动下载jar组件



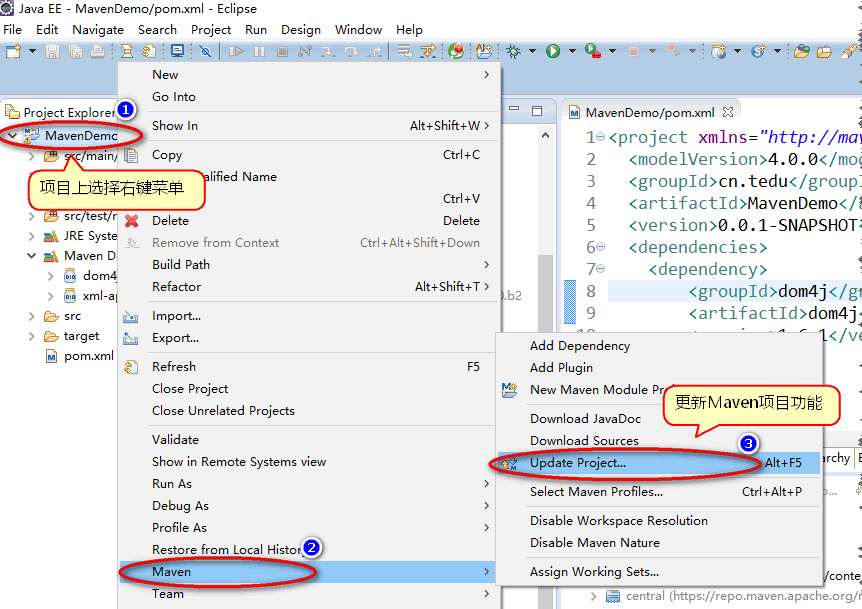
这个搜索功能是利用 Maven 仓库服务器完成的, 如果不能连接到Maven仓库服务器, 这是会发生错误!! 如果出现问题请参考: [Maven 的配置](http://doc.tedu.cn/maven/" \l "maven)

由于版权的问题, 无法在公共Maven服务器上下载Oracle JDBC Driver

## 4.5使用Maven下载的jar组件. 组件保存在本地仓库".m2"文件夹中



## 3.6 如果出现下载故障, 可以在检查Maven仓库正常联通后, 再删除本地仓库".m2"以后"更新Maven 项目"

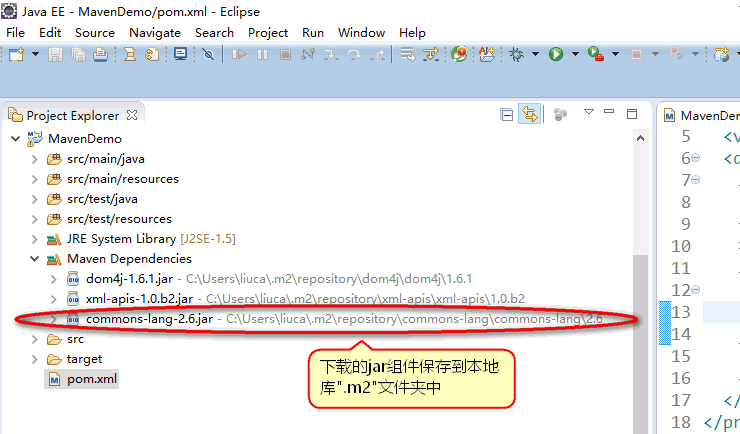


3.7 在线搜索添加Jar组件"坐标"

Maven组件坐标可以在Eclipse中搜索, 也可以利用在线工具搜索

达内jar组件"坐标"搜索: http://maven.tedu.cn/nexus/

阿里云jar组件"坐标"搜索: <http://maven.aliyun.com/nexus/>



# 5 maven 指令

## 5.1 必须在pom.xml的路径下运行

### 5.1.1 **mvn clean**

能够清除target文件夹

### 5.1.2 **mvn compile** 能够将java文件编译.class文件

### 5.1.3 **mvn site**

会生成一个maven工程的手册

### 5.1.4 **mvn package**

能够实现打包

### 5.1.5 **mvn install**

### 包括clean-compile-package一气呵成

### 5.1.6 **mvn tomcat1:run**

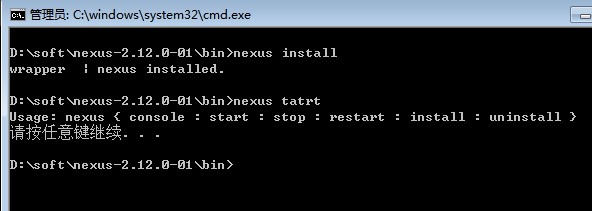
# 6 私服搭建

## 6.1、传统方式搭建私服

把war放到tomcat中（不推荐使用）

## 6.2 以用户服务的方式搭建私服

1. 解压nexus文件
2. 在bin问路下执行
3. **exus install** 安装nexus
4. **nexus start** 开启服务

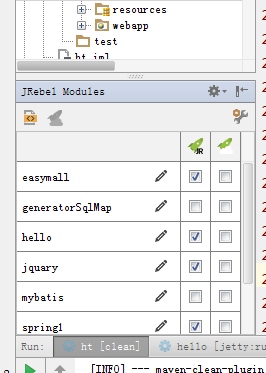


# 7 maven 配置jrebel热部署

1、

|  |
| --- |
| *<!-- jetty插件 -->* <**plugin**>  <**groupId**>org.mortbay.jetty</**groupId**>  <**artifactId**>maven-jetty-plugin</**artifactId**>  <**version**>6.1.8</**version**>  <**configuration**>  <**webAppConfig**>  <**contextPath**>/$</**contextPath**>  </**webAppConfig**>  <**scanIntervalSeconds**>0</**scanIntervalSeconds**>  </**configuration**> </**plugin**> *<!-- jrebel插件，自动生成rebel.xml -->* <**plugin**>  <**groupId**>org.zeroturnaround</**groupId**>  <**artifactId**>jrebel-maven-plugin</**artifactId**>  <**version**>1.1.5</**version**>  <**executions**>  <**execution**>  <**id**>generate-rebel-xml</**id**>  <**phase**>process-resources</**phase**>  <**goals**>  <**goal**>generate</**goal**>  </**goals**>  </**execution**>  </**executions**> </**plugin**> |

2



3

