

Maven 项目管理工具

白面郎君

Svn eclipse maven 量级

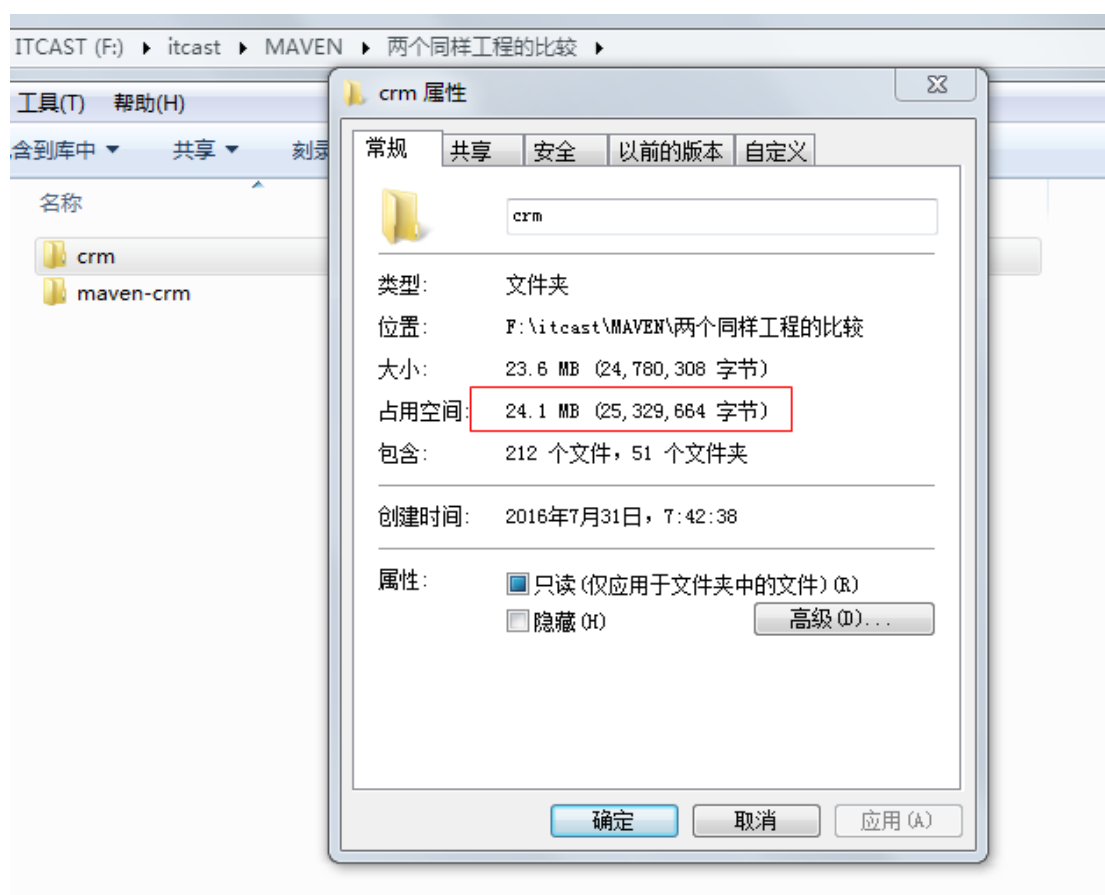
1 Maven 的简介

1.1 什么是 maven

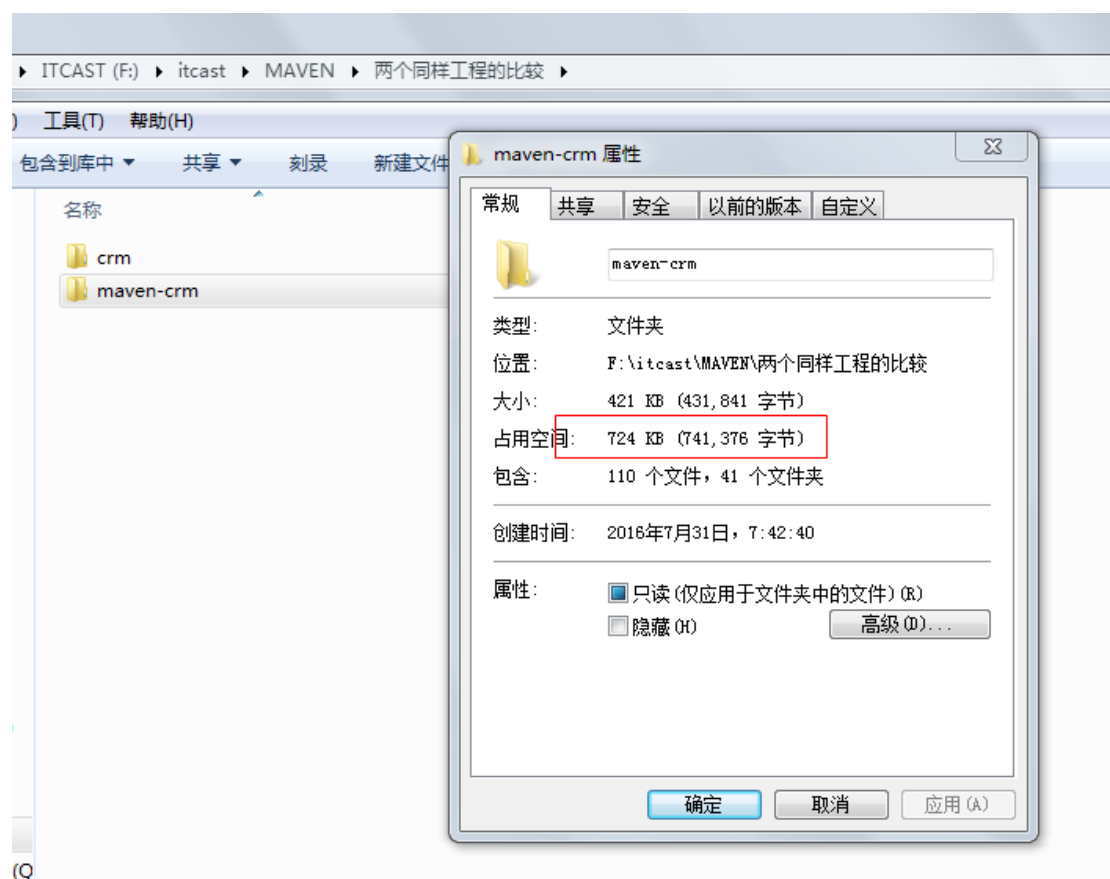
是 apache 下的一个开源项目，是纯 java 开发，并且只是用来管理 java 项目的

1.2 Maven 好处

普通的传统项目

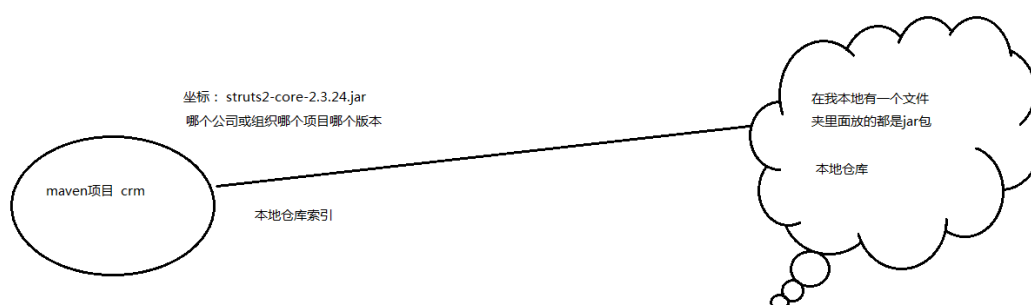


Maven 项目



分析：maven 项目为什么这么小？没有 jar。需要 jar 吗？肯定需要。没有存在于 maven 项目里面，jar 存在于哪？

1.3 依赖管理



1.4 项目一键构建

编码 编译 测试(junit) 运行 打包 部署

一个 `tomcat:run` 就能把项目运行起来

Maven 能干的事:

编译 测试(junit) 运行 打包 部署

1.5 小结:

Maven 的好处:

1.5.1 依赖管理

就是对 jar 包的统一管理 可以节省空间

1.5.2 一键构建

1.5.3 可以跨平台

1.5.4 应用于大型项目 可以提高开发效率

分析:

用户管理 订单管理 支付管理

Maven 的分模块开发

互联网项目 按业务分

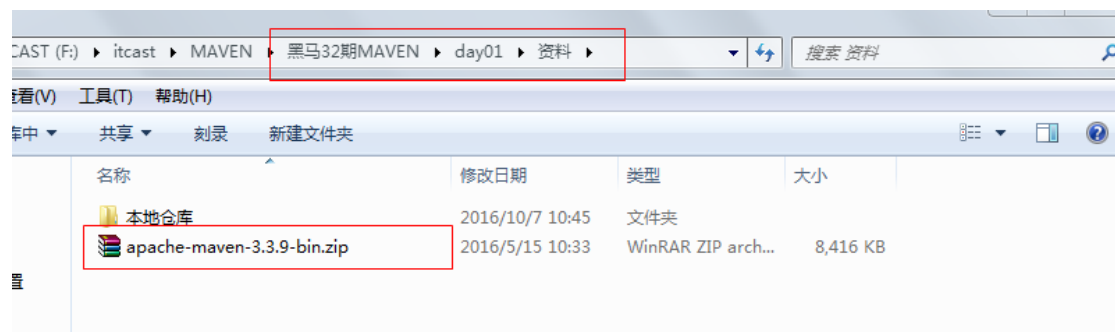
传统项目 按层分 entity dao service web

2 Maven 的安装配置

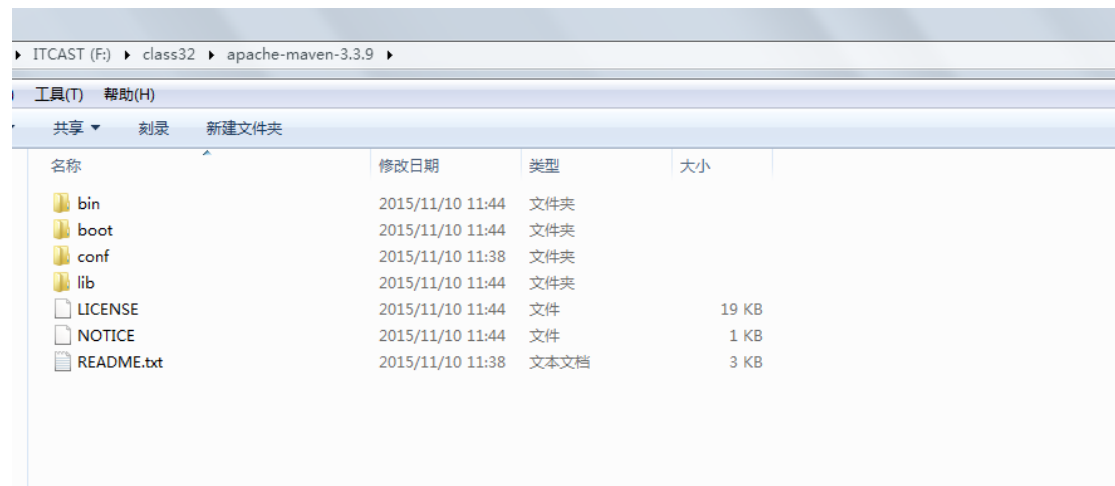
2.1 下载安装

从该网站 <http://maven.apache.org/download.cgi> 下载 maven3.3.9 版本

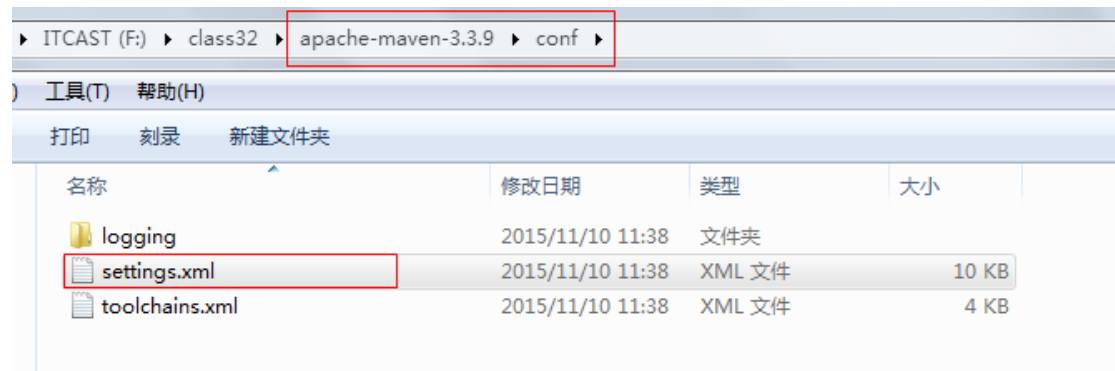
从此目录中拷贝文件:



直接解压到一个没有中文没有空格的文件夹里解压后的目录如下：



Maven 软件的核心文件：



2.2 Maven 环境变量配置

1、要配置 jdk， maven3.3.9 这个版本所需的 jdk 版本必须要 1.7 以上

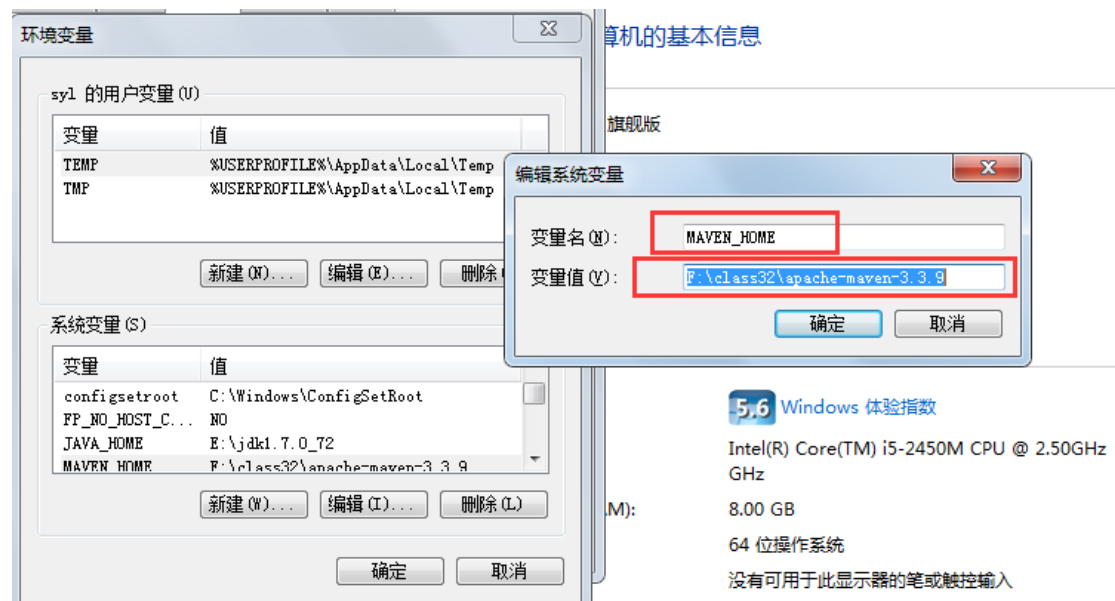
2、最终要运行的是 maven 软件中 bin 目录的 mvn 命令

所以要配置 maven 的环境变量

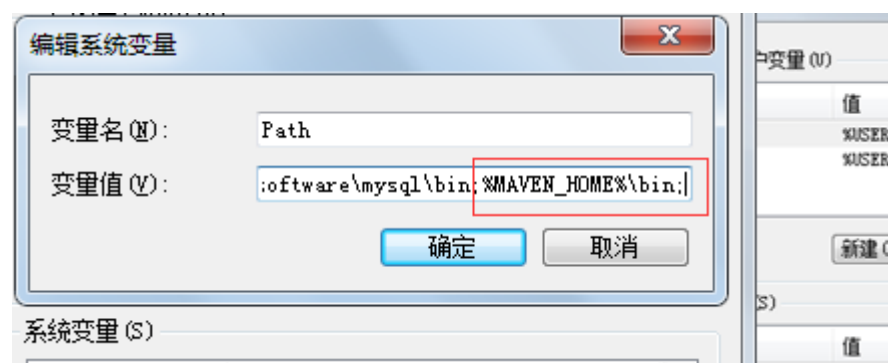
在系统变量添加

环境变量的名称：MAVEN_HOME

变量值：就是 maven 软件解压的目录 F:\class32\apache-maven-3.3.9

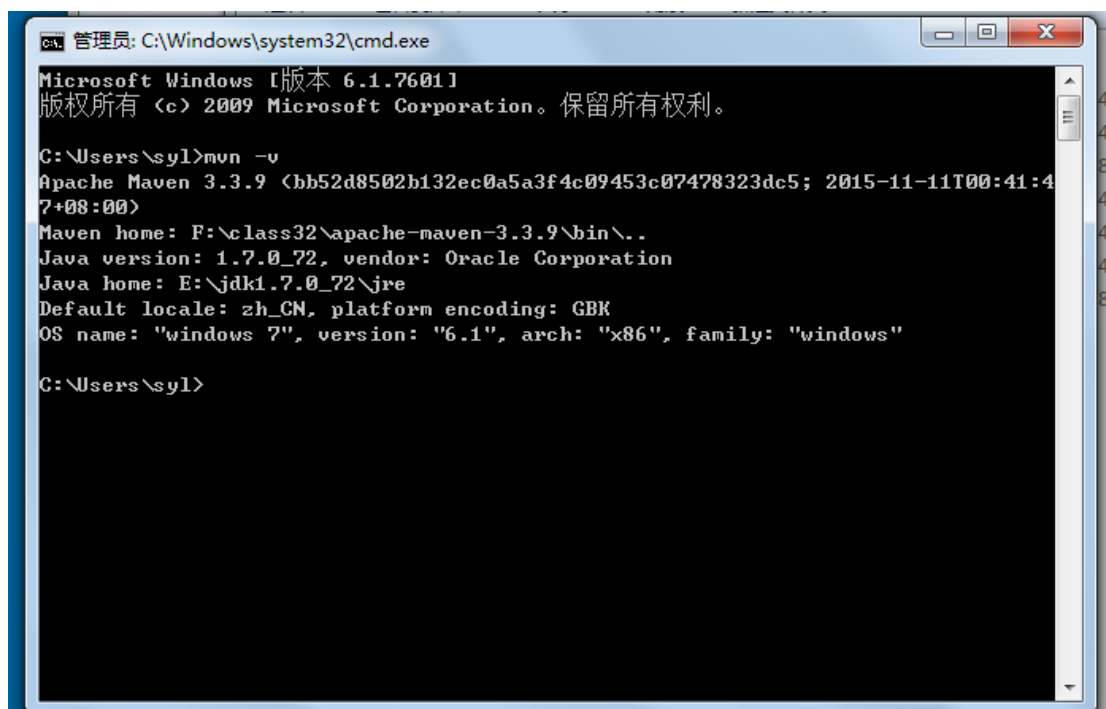


3、把 MAVEN_HOME 添加到 path 里



4、验证 maven 是否配置成功:

打开 dos 窗口 输入: `mvn -v`

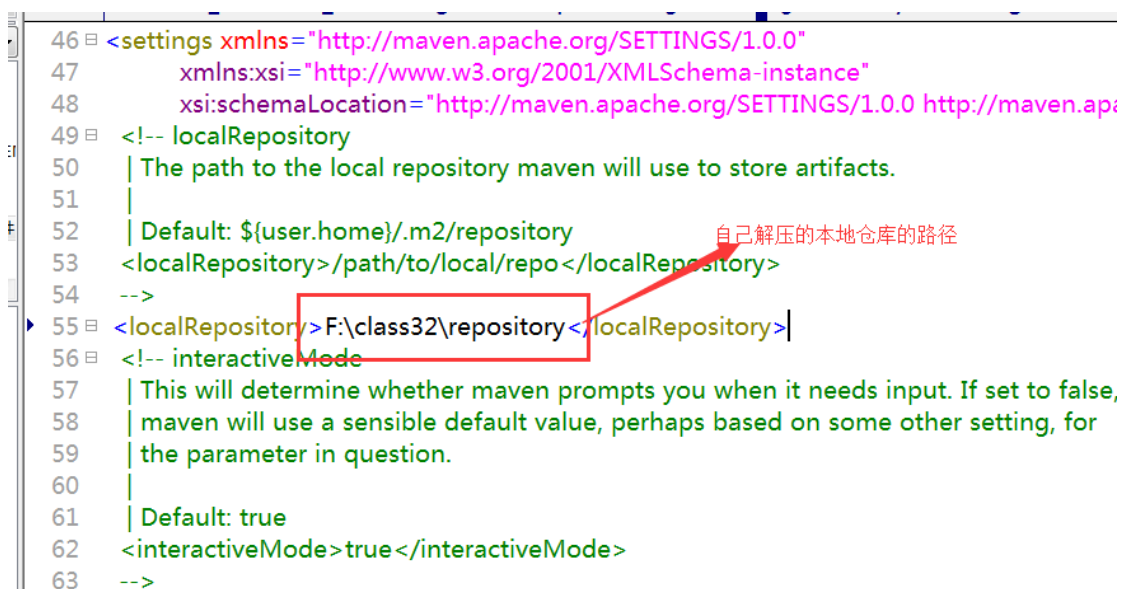


2.3 Maven 仓库

三种仓库

1、本地仓库 自己维护

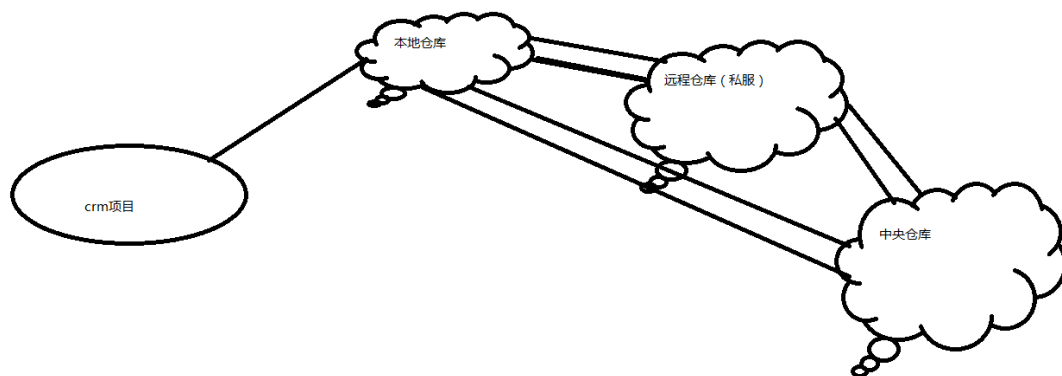
本地仓库的配置只需要修改 settings.xml 文件就可以



2、远程仓库（私服） 公司维护

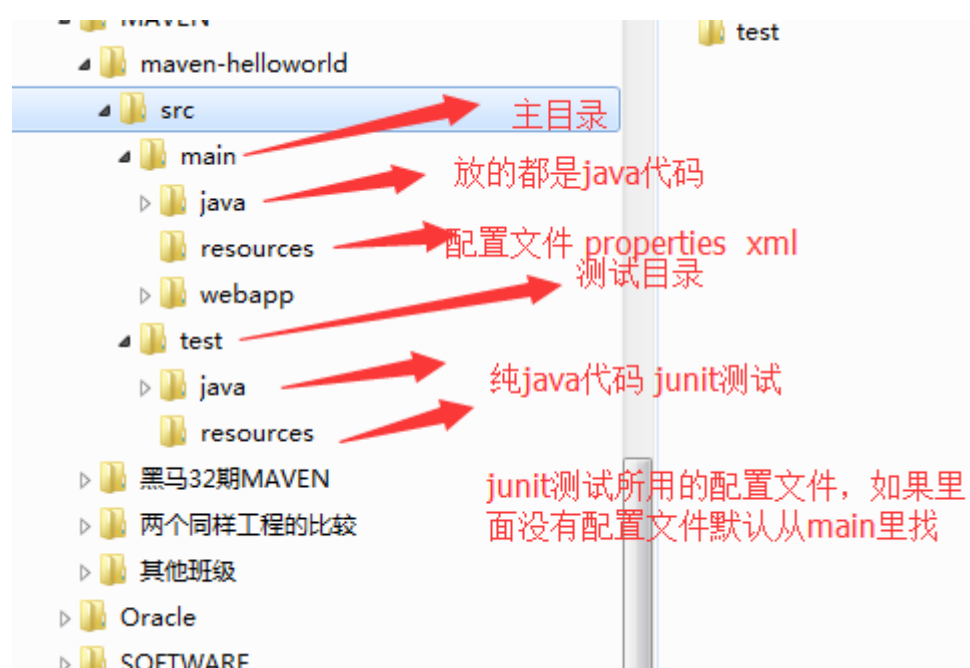
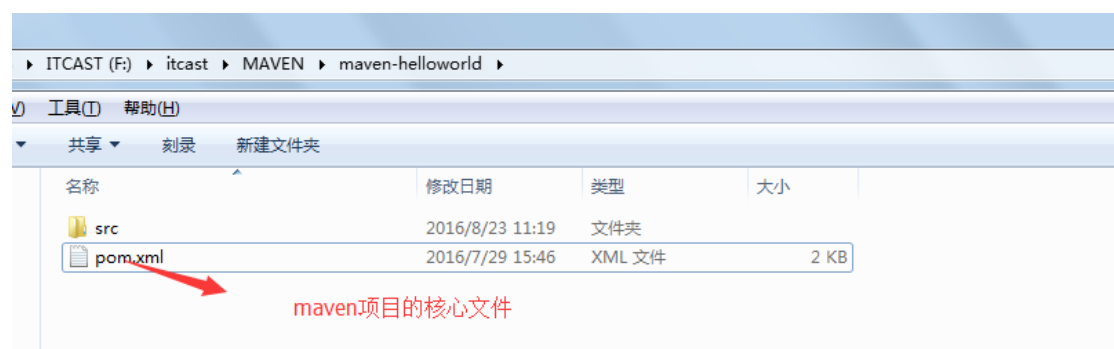
3、中央仓库 maven 团队维护 两个亿

三种仓库的关系如下：



3 演示入门程序

3.1 Maven 的目录结构



3.2 Maven 的常用命令

Clean 清理编译的文件

```
终止批处理操作吗(Y/N)? y
F:\itcast\MAVEN\maven-helloworld>mvn clean
[INFO] Scanning for projects...
[INFO]
[INFO] -----
[INFO] Building 第一个maven工程 0.0.1-SNAPSHOT
[INFO] -----
[INFO]
[INFO] --- maven-clean-plugin:2.5:clean (default-clean) @ maven-helloworld ---
[INFO] Deleting F:\itcast\MAVEN\maven-helloworld\target
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO]
[INFO] Total time: 0.303 s
[INFO] Finished at: 2016-10-08T10:09:55+08:00
[INFO] Final Memory: 4M/15M
[INFO] -----
F:\itcast\MAVEN\maven-helloworld>
```

Compile 编译了主目录的文件

```
F:\itcast\MAVEN\maven-helloworld>mvn compile
[INFO] Scanning for projects...
[INFO]
[INFO] -----
[INFO] Building 第一个maven工程 0.0.1-SNAPSHOT
[INFO] -----
[INFO]
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ maven-helloworld ---
[WARNING] Using platform encoding (GBK actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] Copying 0 resource
[INFO]
[INFO] --- maven-compiler-plugin:3.5.1:compile (default-compile) @ maven-helloworld ---
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 1 source file to F:\itcast\MAVEN\maven-helloworld\target\classes
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO]
[INFO] Total time: 1.319 s
[INFO] Finished at: 2016-10-08T10:10:48+08:00
[INFO] Final Memory: 10M/24M
[INFO] -----
F:\itcast\MAVEN\maven-helloworld>
```

Test 编译并运行了 test 目录的代码


```

[INFO] Surefire report directory: F:\itcast\MAVEN\maven-helloworld\target\surefi
re-reports

-----
T E S T S
-----
Running cn.itcast.maven.test.HelloTest
hello test....
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.046 sec

Results :

Tests run: 1, Failures: 0, Errors: 0, Skipped: 0

[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 2.779 s
[INFO] Finished at: 2016-10-08T10:12:45+08:00
[INFO] Final Memory: 11M/27M
[INFO] -----
F:\itcast\MAVEN\maven-helloworld>

```

Package 打包

```

Results :

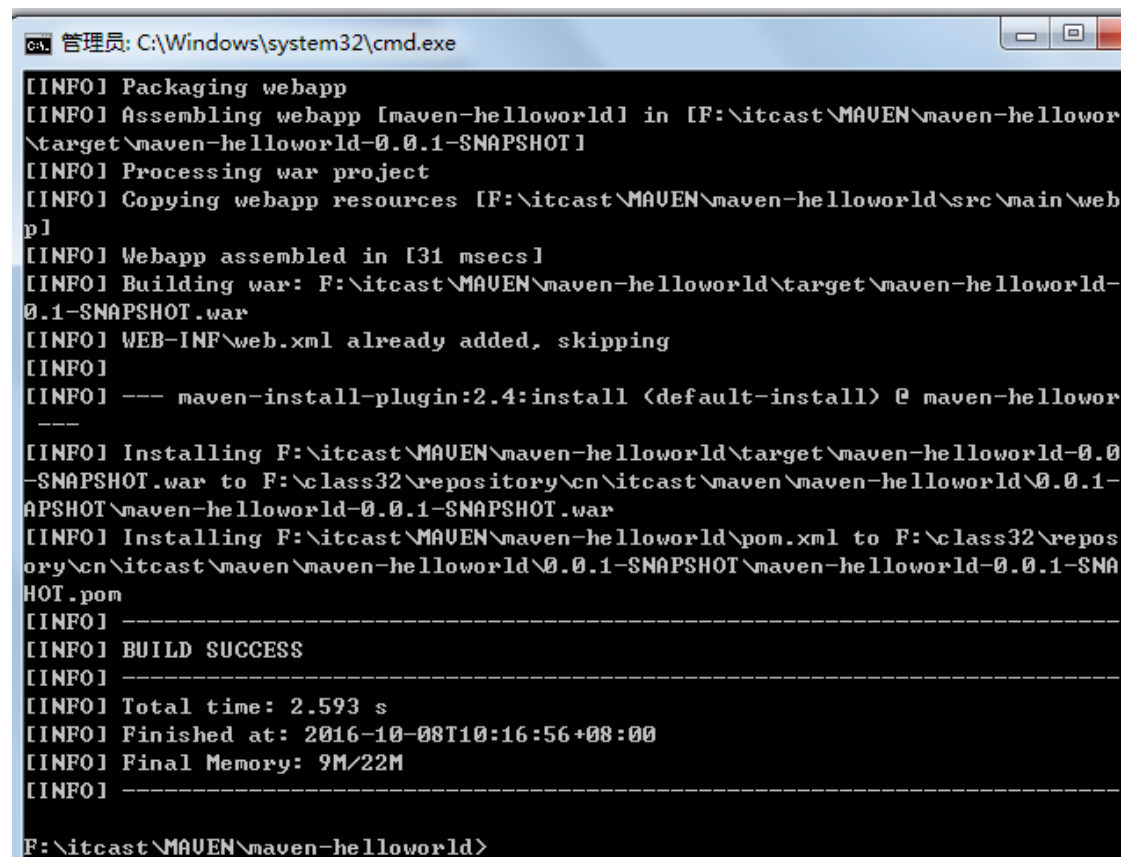
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0

[INFO] --- maven-war-plugin:2.2:war (default-war) @ maven-helloworld ---
[INFO] Packaging webapp
[INFO] Assembling webapp [maven-helloworld] in [F:\itcast\MAVEN\maven-helloworld
\target\maven-helloworld-0.0.1-SNAPSHOT]
[INFO] Processing war project
[INFO] Copying webapp resources [F:\itcast\MAVEN\maven-helloworld\src\main\webap
p]
[INFO] Webapp assembled in [55 msecs]
[INFO] Building war: F:\itcast\MAVEN\maven-helloworld\target\maven-helloworld-0.
0.1-SNAPSHOT.war
[INFO] WEB-INF\web.xml already added, skipping
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 2.514 s
[INFO] Finished at: 2016-10-08T10:14:56+08:00
[INFO] Final Memory: 9M/21M
[INFO] -----
F:\itcast\MAVEN\maven-helloworld>

```

发布到本地仓库

Install 就是把项目发布到本地仓库



```
管理员: C:\Windows\system32\cmd.exe
[INFO] Packaging webapp
[INFO] Assembling webapp [maven-helloworld] in [F:\itcast\MAUEN\maven-helloworld\target\maven-helloworld-0.0.1-SNAPSHOT]
[INFO] Processing war project
[INFO] Copying webapp resources [F:\itcast\MAUEN\maven-helloworld\src\main\web
p]
[INFO] Webapp assembled in [31 msecs]
[INFO] Building war: F:\itcast\MAUEN\maven-helloworld\target\maven-helloworld-0.0.1-SNAPSHOT.war
[INFO] WEB-INF\web.xml already added, skipping
[INFO]
[INFO] --- maven-install-plugin:2.4:install (default-install) @ maven-hellowor
---
[INFO] Installing F:\itcast\MAUEN\maven-helloworld\target\maven-helloworld-0.0.1-SNAPSHOT.war to F:\class32\repository\cn\itcast\maven\maven-helloworld\0.0.1-SNAPSHOT\maven-helloworld-0.0.1-SNAPSHOT.war
[INFO] Installing F:\itcast\MAUEN\maven-helloworld\pom.xml to F:\class32\repository\cn\itcast\maven\maven-helloworld\0.0.1-SNAPSHOT\maven-helloworld-0.0.1-SNAPSHOT.pom
[INFO]
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 2.593 s
[INFO] Finished at: 2016-10-08T10:16:56+08:00
[INFO] Final Memory: 9M/22M
[INFO] -----
F:\itcast\MAUEN\maven-helloworld>
```

Tomcat: run 一键启动

3.3 Maven 的生命周期（了解）

Compile test package install deploy（发布到私服）

三种生命周期

Clean 生命周期

Clean

Default 生命周期

Compile test package install deploy

Site 生命周期

Site

3.4 命令和生命周期的阶段的关系

不同的生命周期的命令可以同时执行

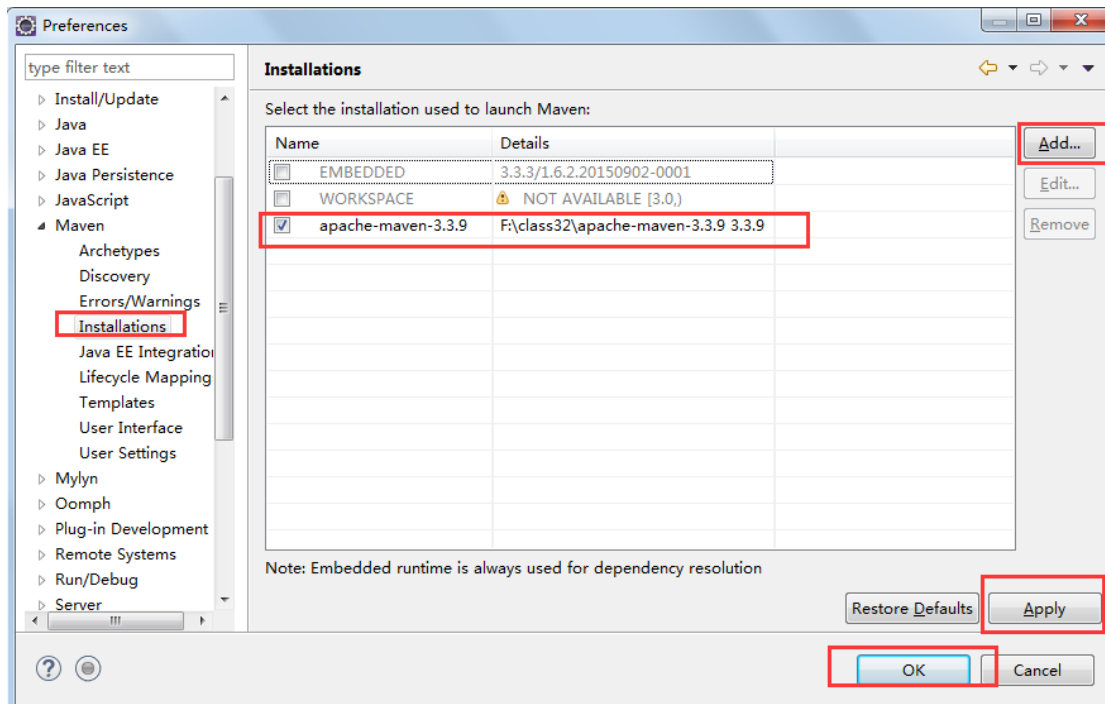
Mvn clean package

4 项目构建

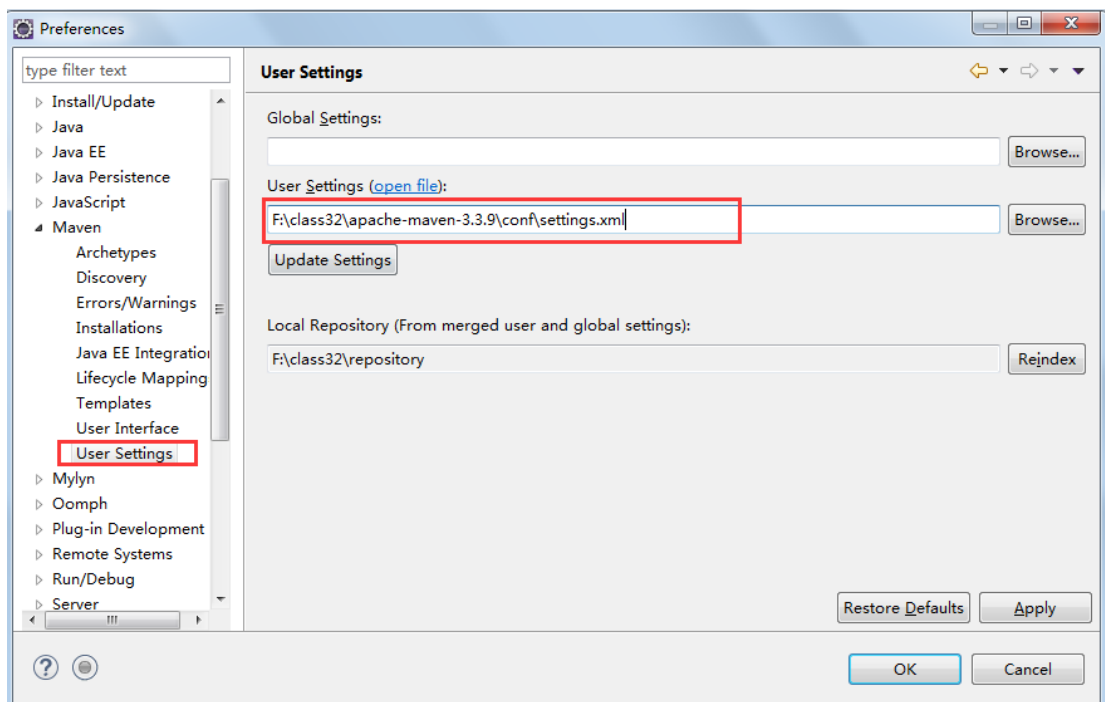
M2e 插件的安装

Maven 在 eclipse 的配置

1、选择 3.3.9 版本的 maven 软件

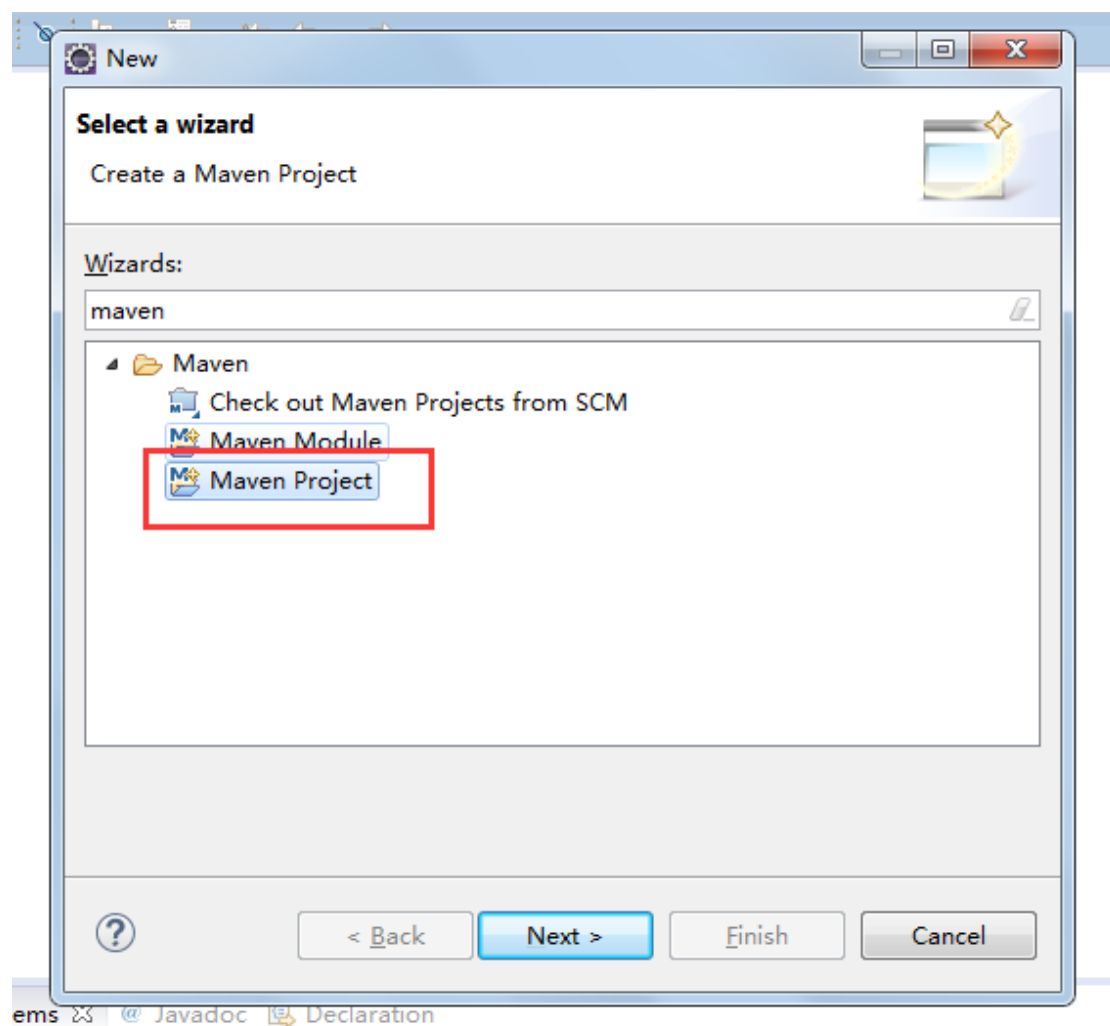


2、修改默认的本地仓库地址

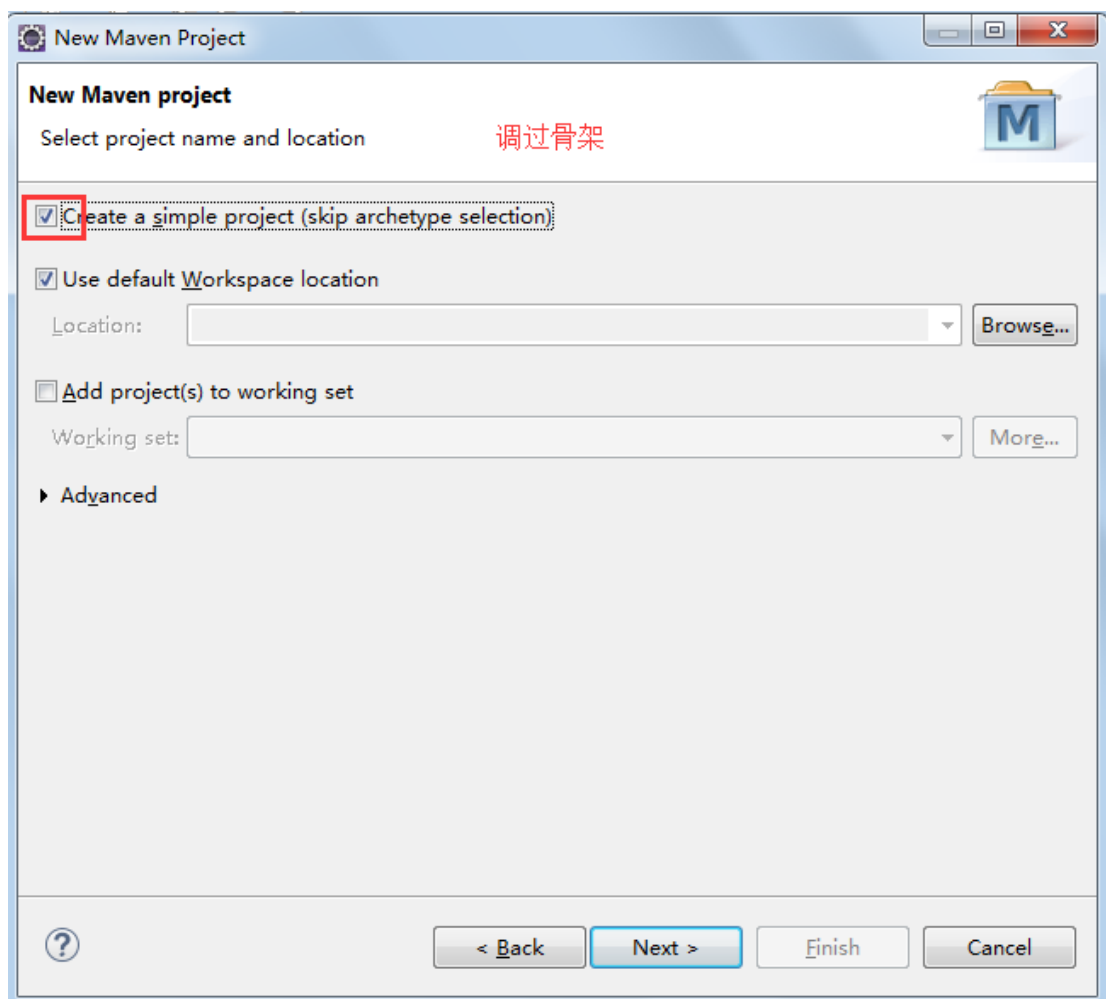


项目构建

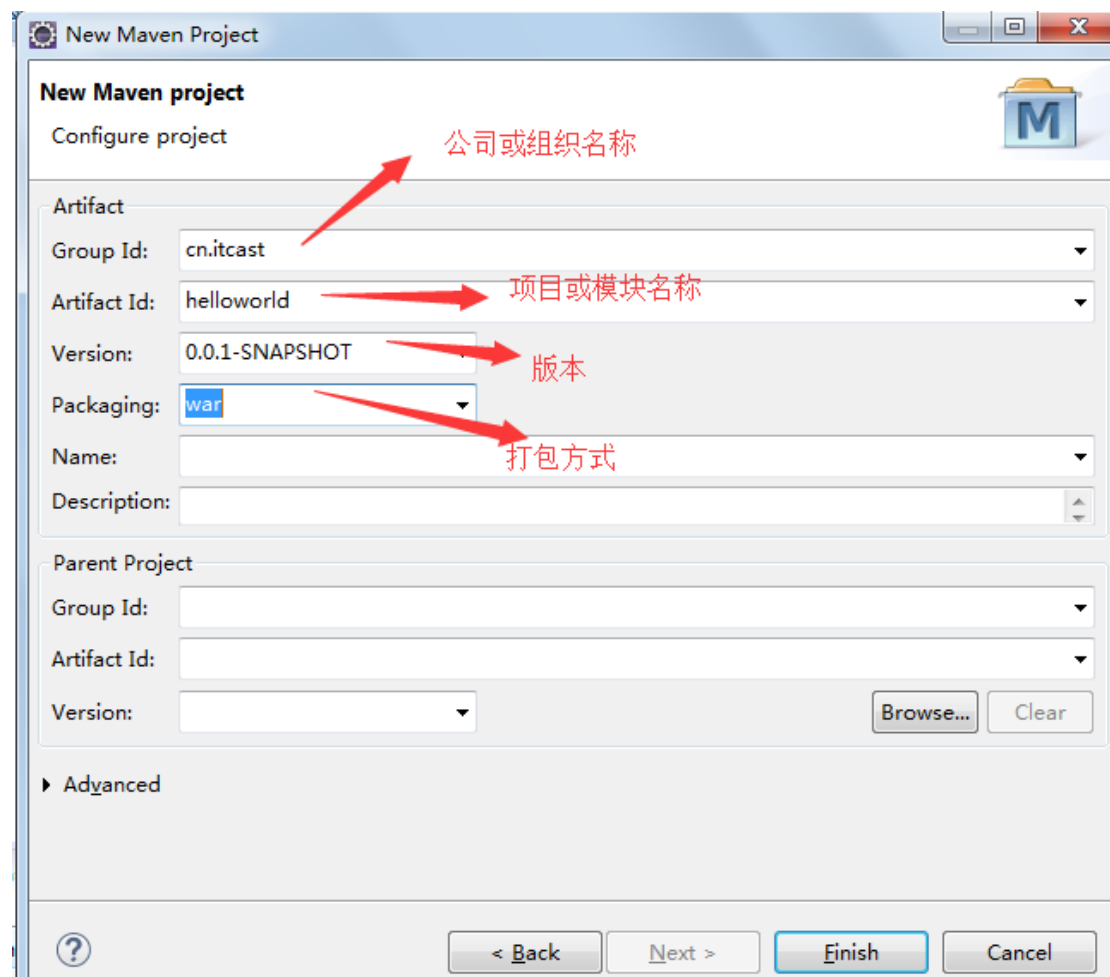
1、新建 maven 项目



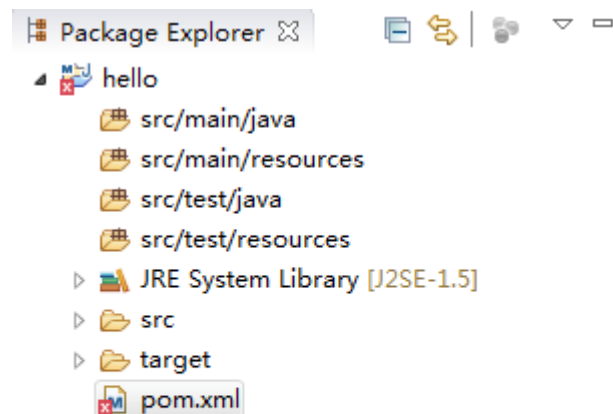
2、调过骨架，如果不跳过骨架选择创建出的项目目录是不全的



2、填写坐标

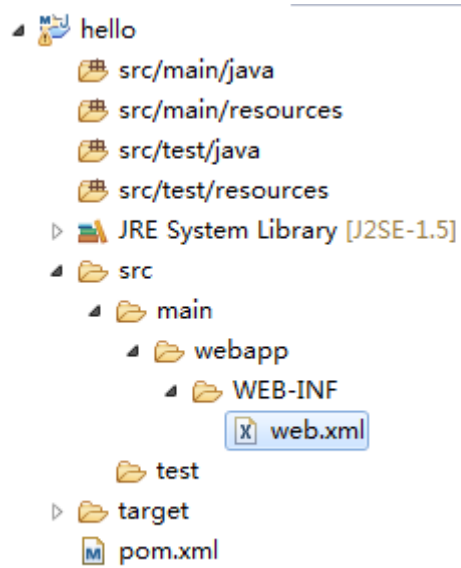


3、创建工程



4、处理红色叉号

手动在 webapp 文件夹下创建一个 WEB-INF 文件夹，在里面放一个 web.xml 文件

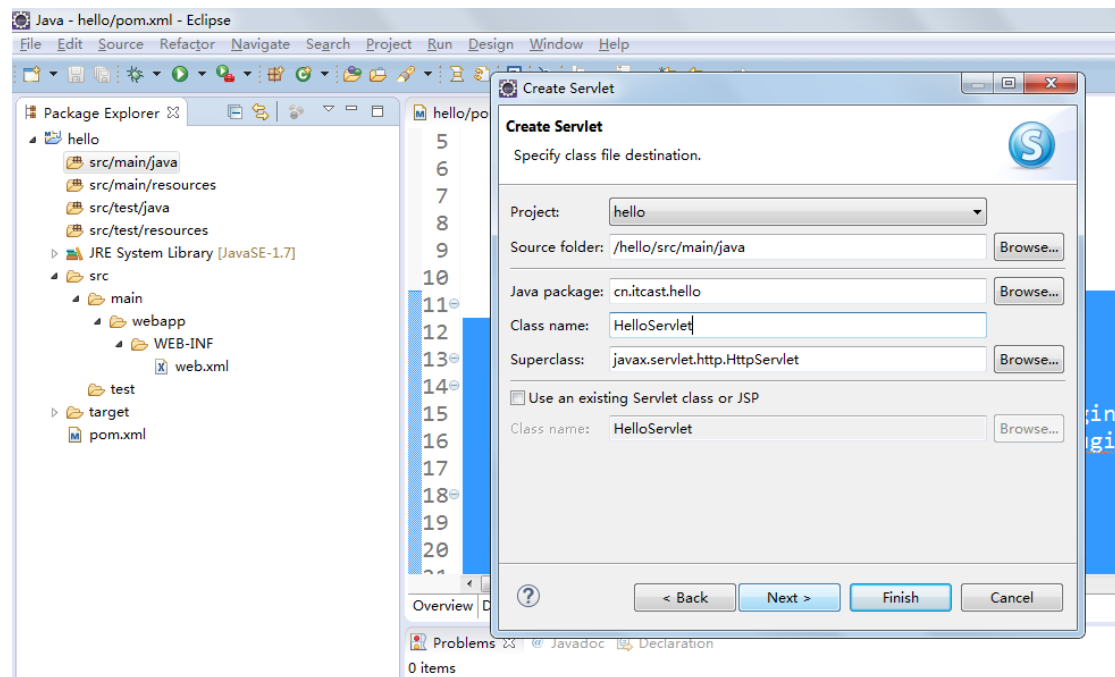


5、处理编译版本

在 pom.xml 中添加如下代码

```
<build>
    <!-- 配置了很多插件 -->
    <plugins>
        <plugin>
            <groupId>org.apache.maven.plugins</groupId>
            <artifactId>maven-compiler-plugin</artifactId>
            <version>3.5.1</version>
            <configuration>
                <source>1.7</source>
                <target>1.7</target>
                <encoding>UTF-8</encoding>
            </configuration>
        </plugin>
    </plugins>
</build>
```

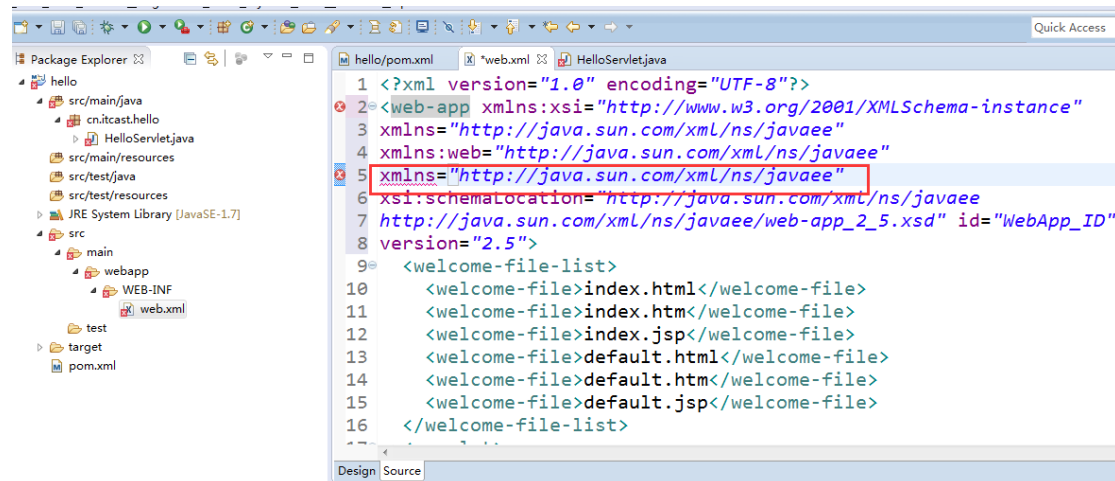
5、创建一个 servlet



修改 web.xml

删除重复的代码

[xmlns=http://java.sun.com/xml/ns/javaee](http://java.sun.com/xml/ns/javaee)



添加 jar 包

在 pom 中添加如下代码:

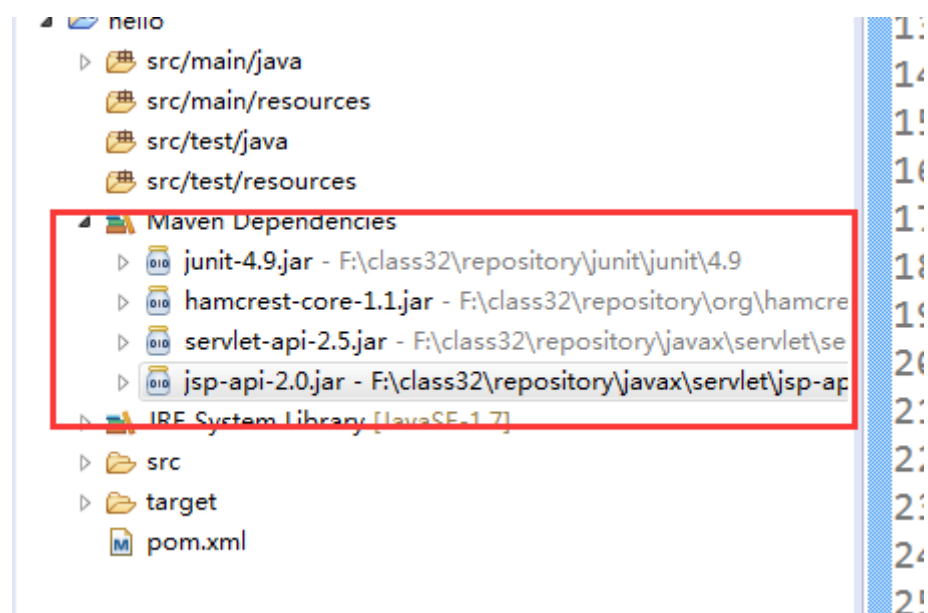
```
<dependencies>
  <dependency>
    <groupId>junit</groupId>
    <artifactId>junit</artifactId>
    <version>4.9</version>
```



```
<scope>test</scope>
</dependency>
<dependency>
  <groupId>javax.servlet</groupId>
  <artifactId>servlet-api</artifactId>
  <version>2.5</version>
  <scope>provided</scope>
</dependency>
<dependency>
  <groupId>javax.servlet</groupId>
  <artifactId>jsp-api</artifactId>
  <version>2.0</version>
  <scope>provided</scope>
</dependency>

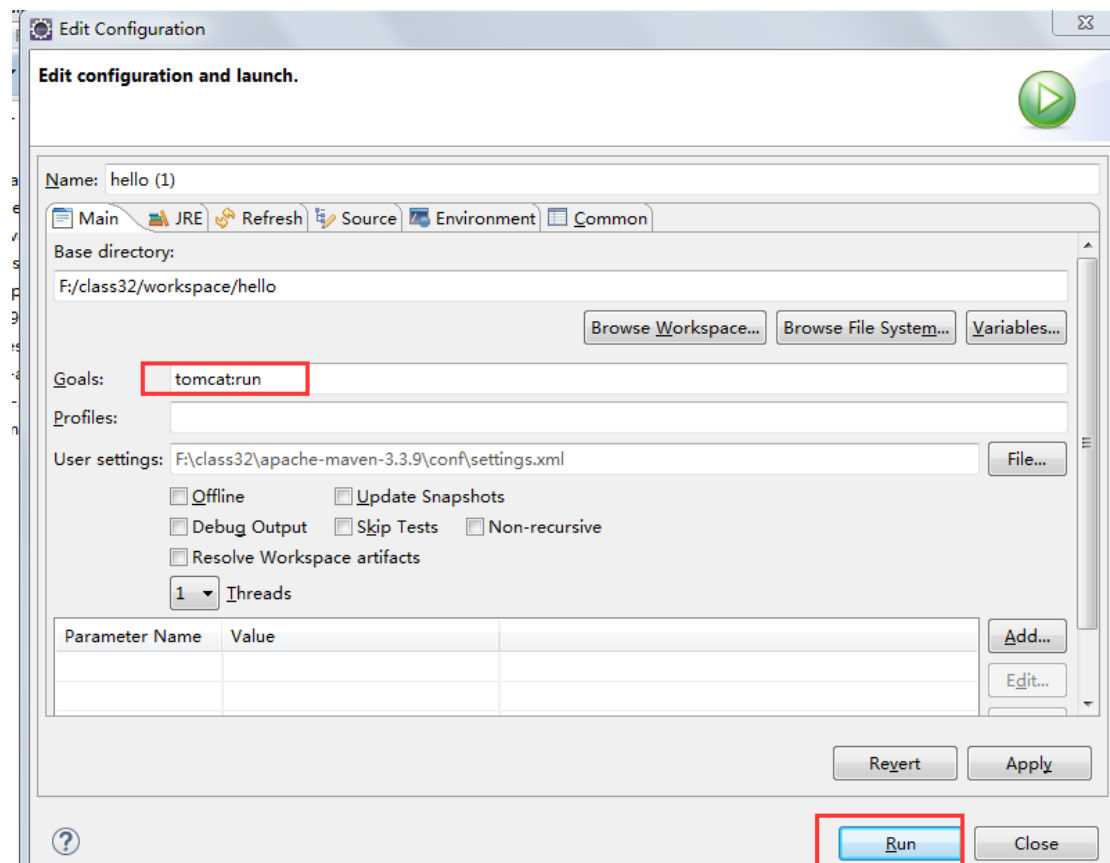
</dependencies>
```

会发现 jar 包已添加到项目中



启动项目

右击项目---run as --maven build



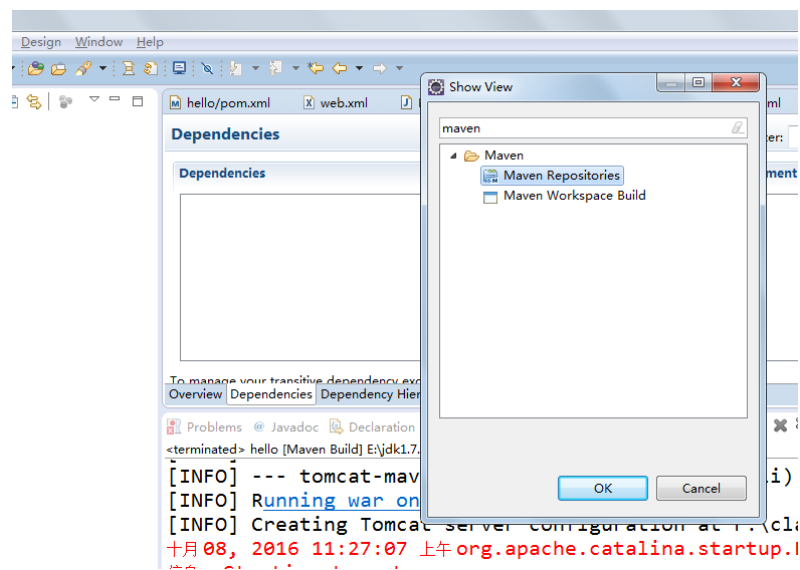
5 依赖管理

Jar 包的管理

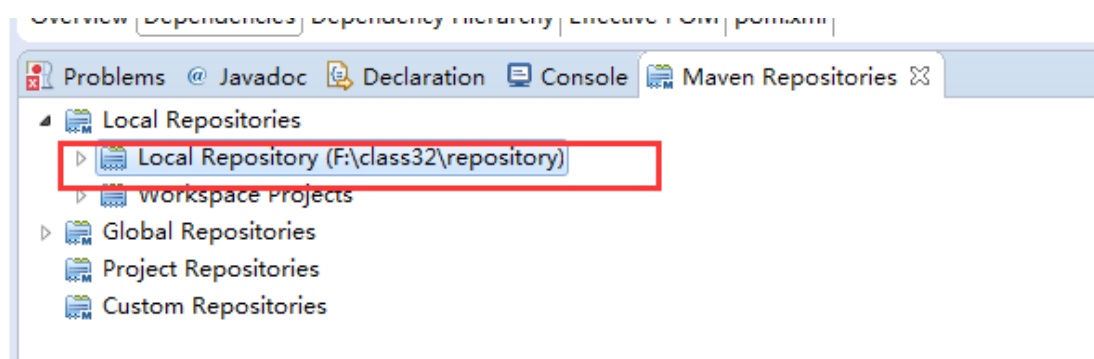
需求：整合 struts2 页面上上传一个客户 id 跳转页面

5.1 添加依赖：

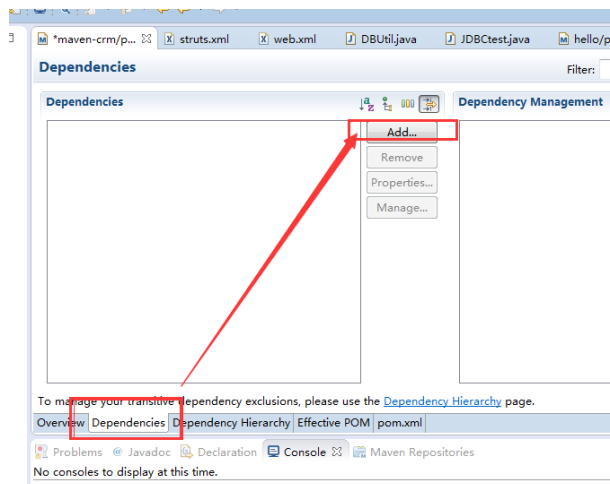
打开 maven 仓库的视图：



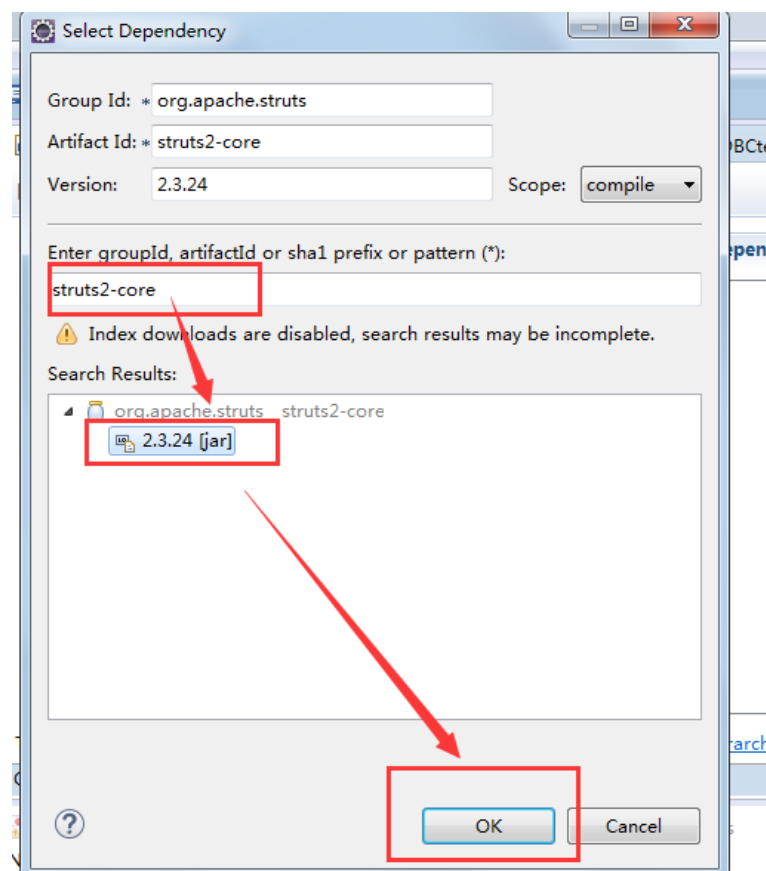
5.2 重建索引



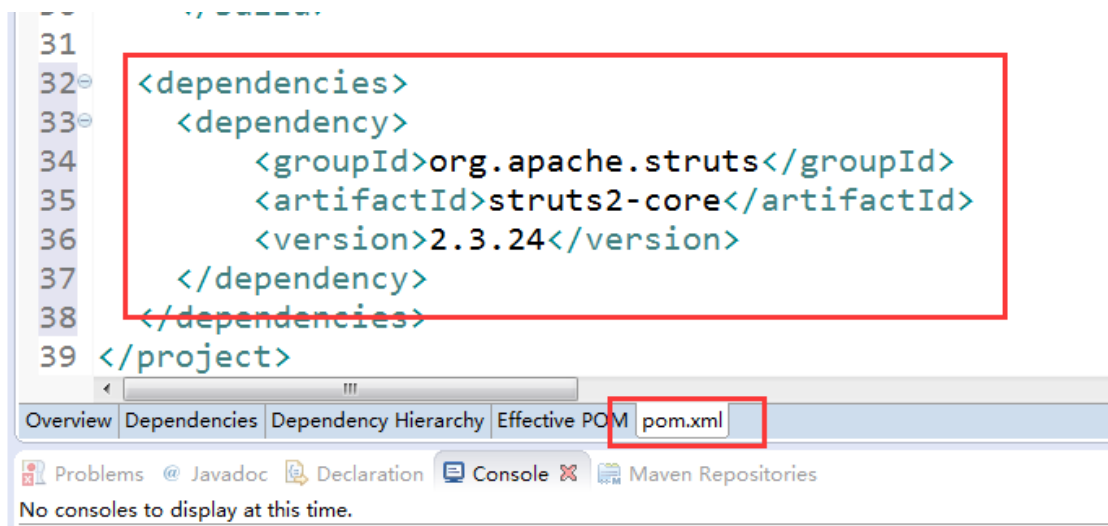
- 1、创建 maven 项目（同上）
- 2、跳过骨架（同上）
- 3、填写坐标信息（同上）
- 4、添加 web.xml 文件（同上）
- 5、修改编译版本（同上）
- 6、添加坐标 选择 Dependencies 标签 点击 add



7、手动输入要添加的坐标，选择版本



8、可以看到 pom.xml 文件中多出了如下代码



9、同样的方式添加 servlet-api.jar 和 jsp-api.jar 注意选择 scope 为 provided

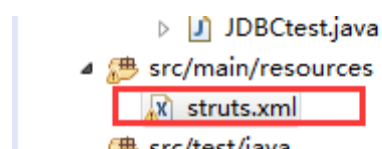
10、写 action 代码

```

import com.opensymphony.xwork2.ActionSupport;
public class CutomerAction extends ActionSupport {
    private Long custId;
    public Long getCustId() {
        return custId;
    }
    public void setCustId(Long custId) {
        this.custId = custId;
    }
    public String findById(){
        return SUCCESS;
    }
}

```

11、添加 struts.xml 文件放到 resources 目录中



内容:

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE struts PUBLIC
    "-//Apache Software Foundation//DTD Struts Configuration 2.3//EN"
    "http://struts.apache.org/dtds/struts-2.3.dtd">

<struts>

```

```

<!-- 配置常量 -->
<!-- 字符集 -->
<constant name="struts.i18n.encoding" value="UTF-8"></constant>
<!-- 开发模式 -->
<constant name="struts.devMode" value="true"></constant>

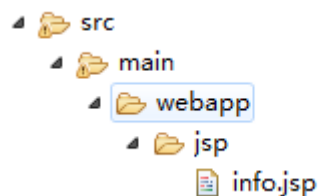
<!-- 通用 package -->
<package name="customer" namespace="/" extends="struts-default">

    <action name="find" class="cn.itcast.action.CutomerAction"
        method="findByld">
        <result name="success">/jsp/info.jsp</result>
    </action>

</package>
</struts>

```

12、 添加 jsp 页面



15、修改 web.xml 文件 添加过滤器

```

<filter>
    <filter-name>struts2</filter-name>
    <filter-class>org.apache.struts2.dispatcher.ng.filter.StrutsPrepareAndExecuteFilter</filter-class>
</filter>
<filter-mapping>
    <filter-name>struts2</filter-name>
    <url-pattern>/*</url-pattern>
</filter-mapping>

```

5.3 依赖范围

5.3.1 Compile struts2-core

编译（compile）时需要 测试时需要，， 运行时需要， 打包时需要

5.3.2 Provided `jsp-api.jar` `servlet-api.jar`

编译（`compile`）时需要，测试（`test`）时也需要 ，运行时不需要，打包时不需要

5.3.3 Runtime 数据库驱动包

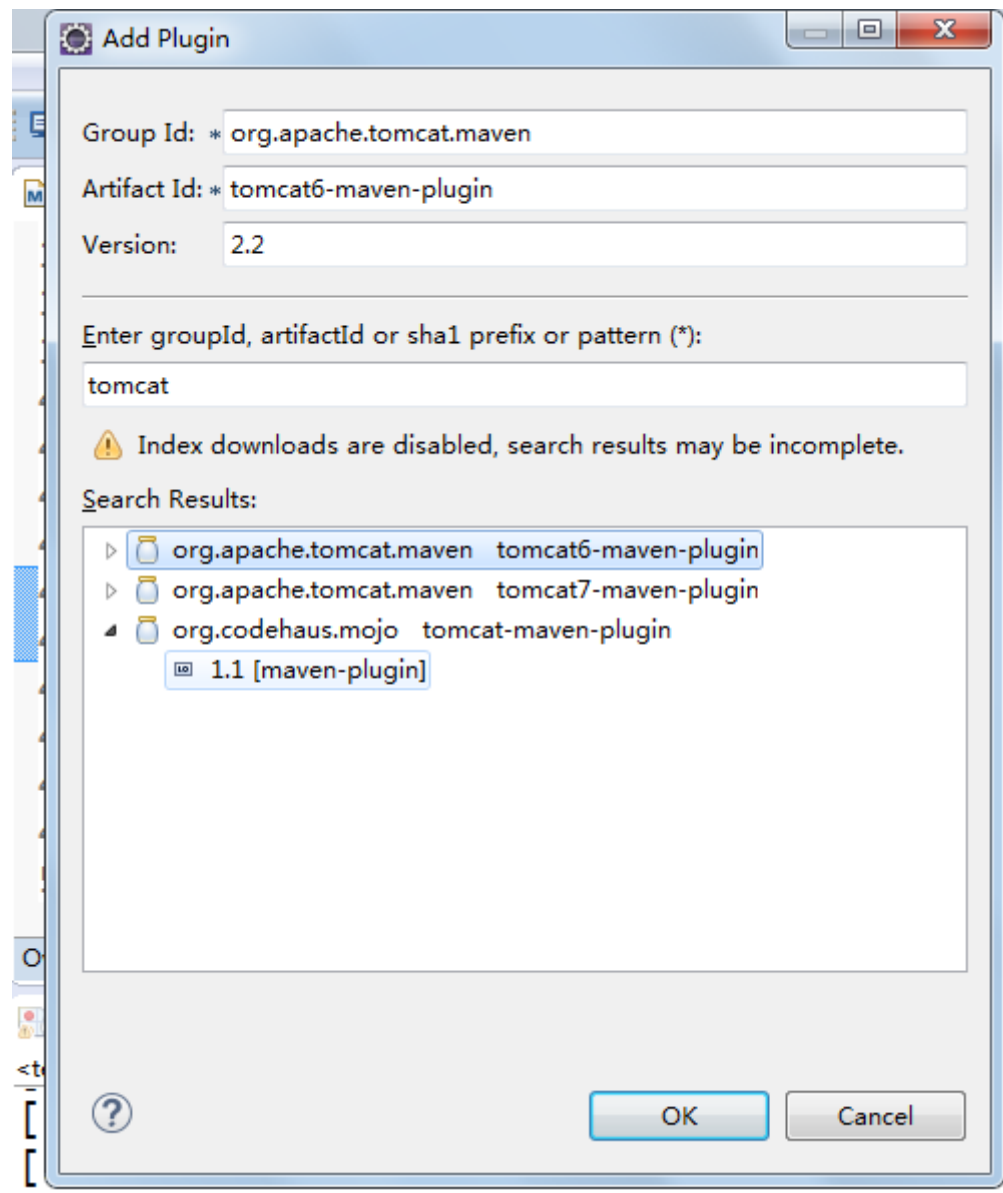
编译时不需要，测试时需要，运行时需要，打包时需要

5.3.4 Test `junit.jar`

编译时不需要，测试时需要，运行时不需要，打包也不需要

添加插件

Maven `add` `plugin`



如果用 tomcat7 运行用命令：

Tomcat7:run

常见问题：


```
<terminated> hello [Maven Build] C:\Program Files (x86)\Java\jre7\bin\javaw.exe (2016年7月29日 下午5:00:32)
[INFO] Scanning for projects...
[INFO]
[INFO] -----
[INFO] Building helloworld_project 0.0.1-SNAPSHOT
[INFO]
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ hello ---
[WARNING] Using platform encoding (GBK actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] Copying 0 resource
[INFO]
[INFO] --- maven-compiler-plugin:3.5.1:compile (default-compile) @ hello ---
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 1 source file to F:\class0613\workspace\hello\target\classes
[INFO]
[ERROR] COMPILATION ERROR :
[INFO]
[ERROR] No compiler is provided in this environment. Perhaps you are running on a JRE rather than a JDK?
[INFO] 1 error
[INFO]
[INFO] -----
[INFO] BUILD FAILURE
[INFO]
[INFO] Total time: 0.932 s
[INFO] Finished at: 2016-07-29T17:00:34+08:00
[INFO] Final Memory: 7M/18M
[INFO]
[ERROR] Failed to execute goal org.apache.maven.plugins:maven-compiler-plugin:3.5.1:compile (default-compile) on project hello: Compilation failure
[ERROR] No compiler is provided in this environment. Perhaps you are running on a JRE rather than a JDK?
[ERROR] -> [Help 1]
[ERROR]
[ERROR] To see the full stack trace of the errors, re-run Maven with the -e switch.
[ERROR] Re-run Maven using the -X switch to enable full debug logging.
[ERROR]
[ERROR] For more information about the errors and possible solutions, please read the following articles:
[ERROR] [Help 1] http://wiki.apache.org/confluence/display/MAVEN/MojoFailureException
```