Maven项目管理工具

白面郎君

Svn eclipse maven量级

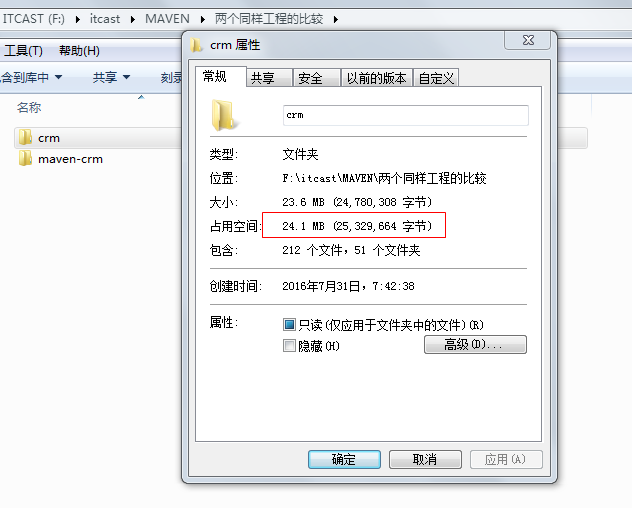
# Maven的简介

## 什么是maven

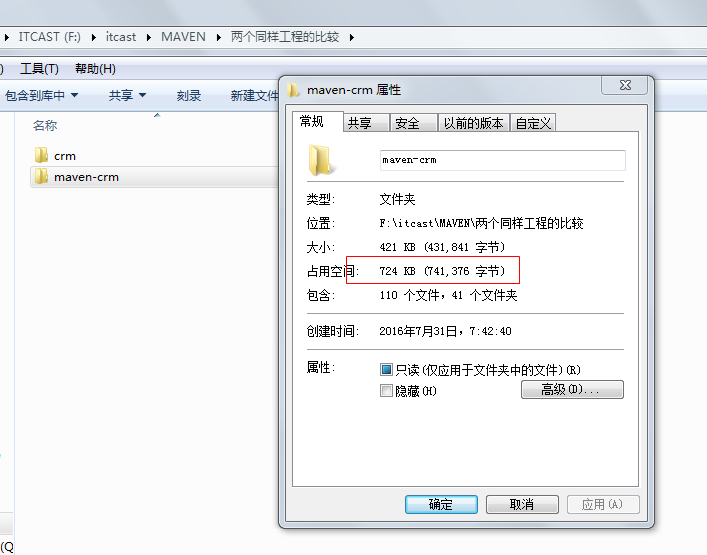
是apache下的一个开源项目，是纯java开发，并且只是用来管理java项目的

## Maven好处

普通的传统项目

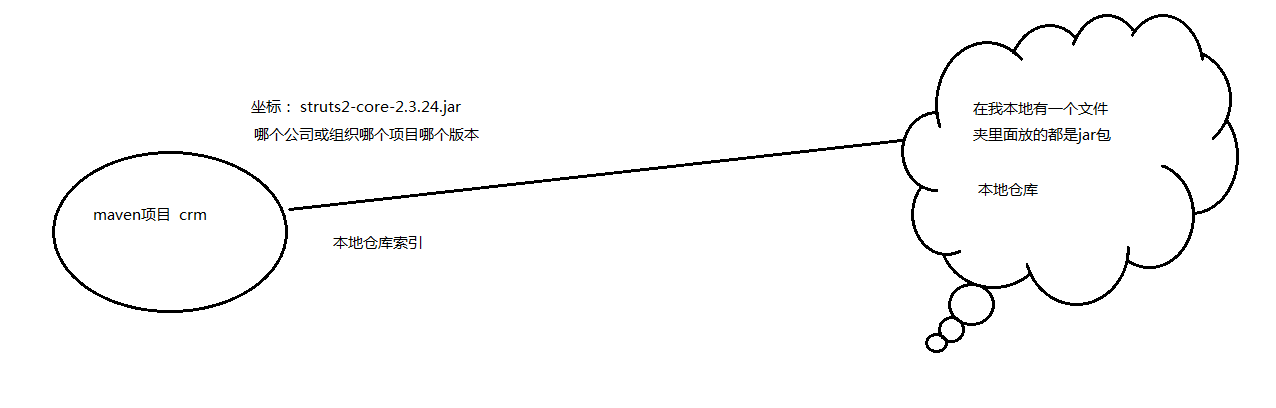


Maven项目



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？

## 依赖管理



## 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

## 小结：

Maven的好处：

### 依赖管理

就是对jar包的统一管理 可以节省空间

### 一键构建

### 可以跨平台

### 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

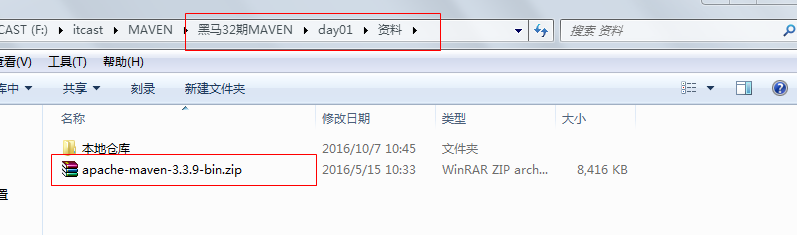
传统项目 按层分 entity dao service web

# Maven的安装配置

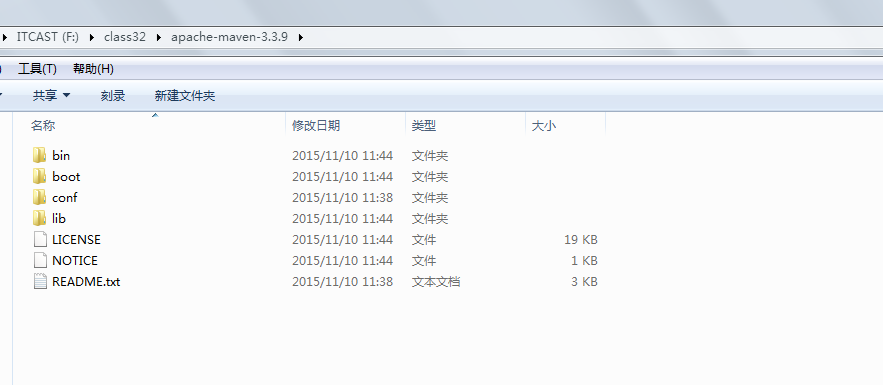
## 下载安装



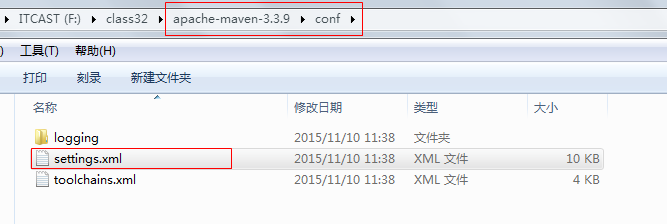
从此目录中拷贝文件：



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：



## Maven环境变量配置

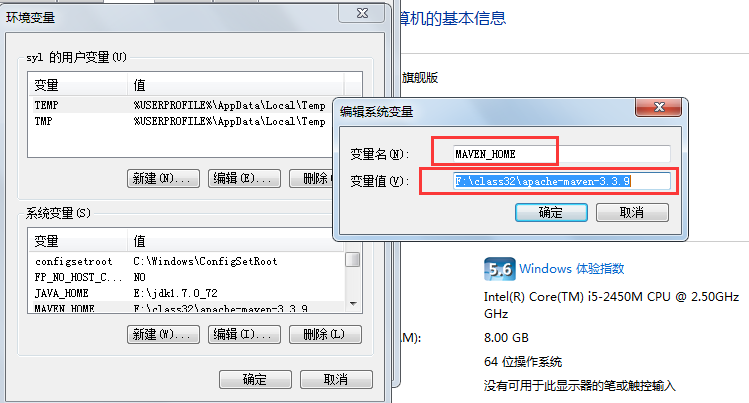
1. 要配置jdk， maven3.3.9这个版本所需的jdk版本必须要1.7以上
2. 最终要运行的是maven软件中bin目录的mvn命令

所以要配置maven的环境变量

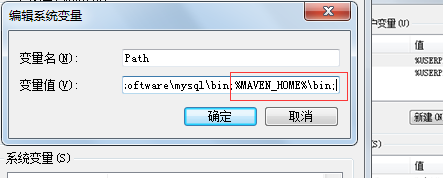
在系统变量添加

环境变量的名称：MAVEN\_HOME

变量值：就是maven软甲解压的目录F:\class32\apache-maven-3.3.9

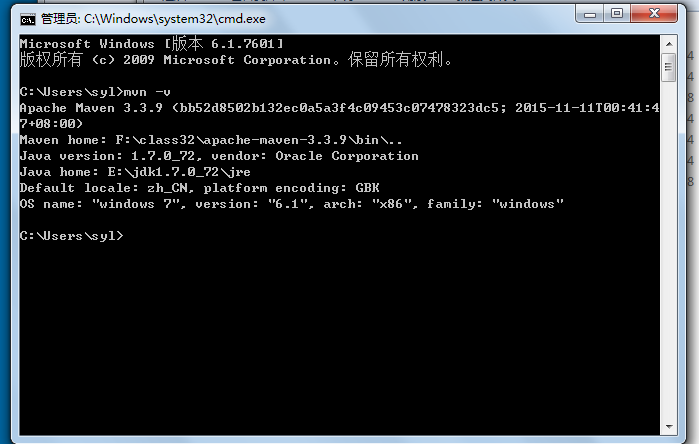


3、把MAVEN\_HOME添加到path里



4、验证maven是否配置成功：

打开dos窗口 输入： mvn –v

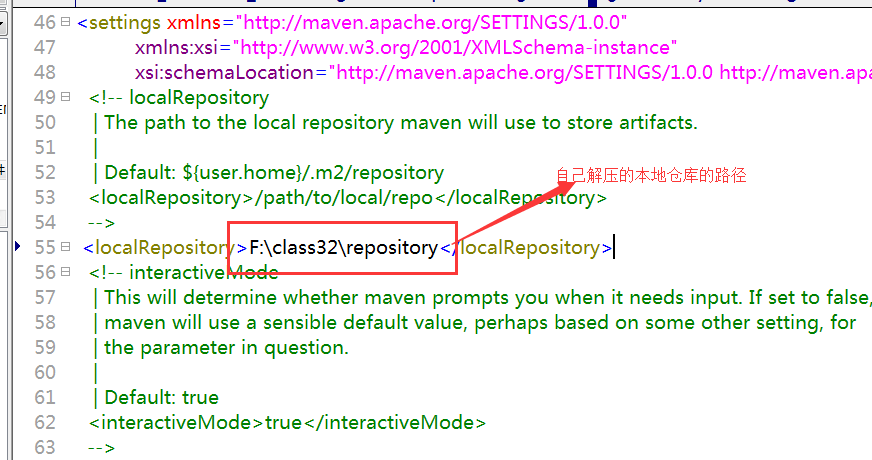


## Maven仓库

三种仓库

1、本地仓库 自己维护

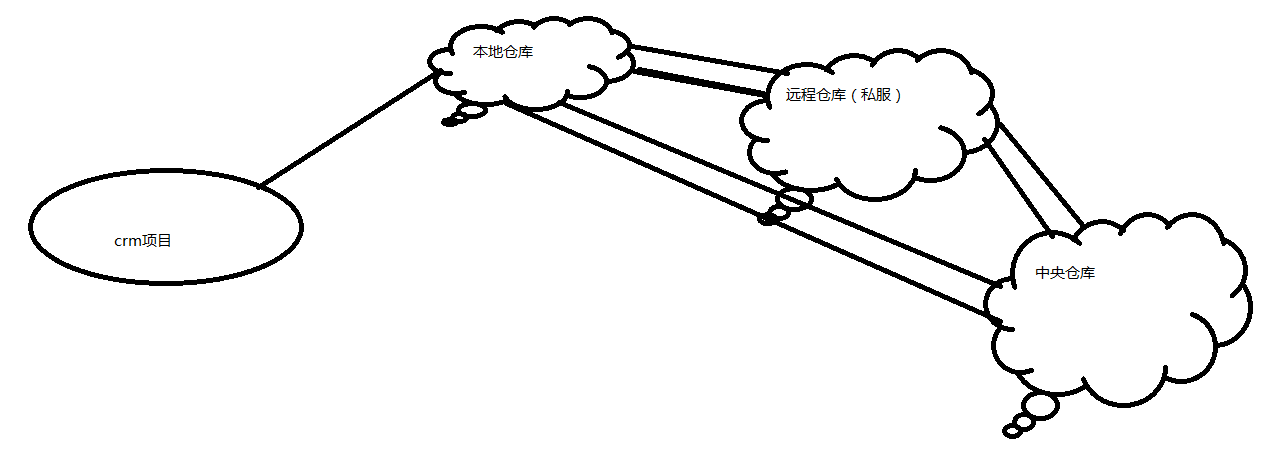
本地仓库的配置只需要修改settings.xml文件就可以



2、远程仓库（私服） 公司维护

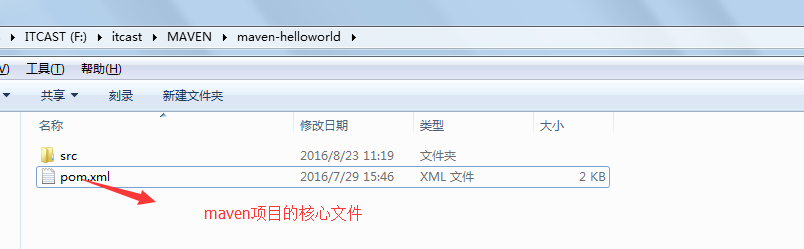
3、中央仓库 maven团队维护 两个亿

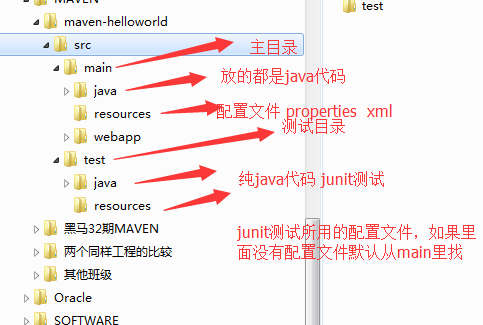
三种仓库的关系如下：



# 演示入门程序

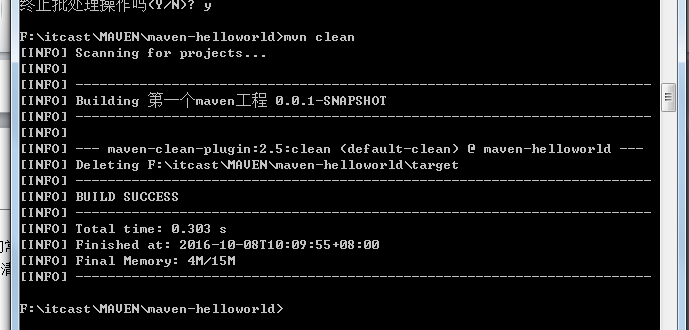
## Maven的目录结构



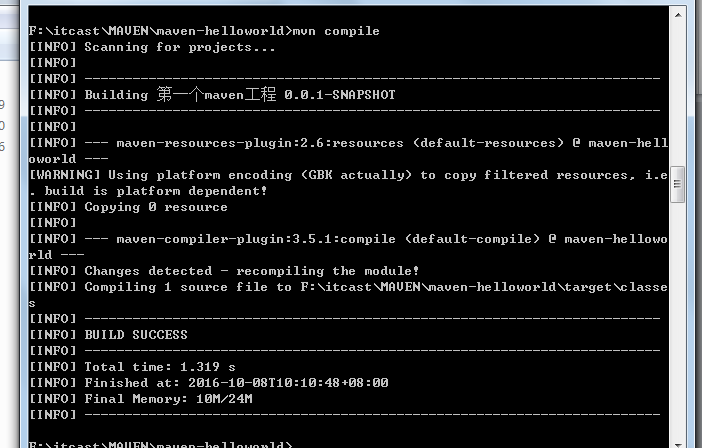


## Maven的常用命令

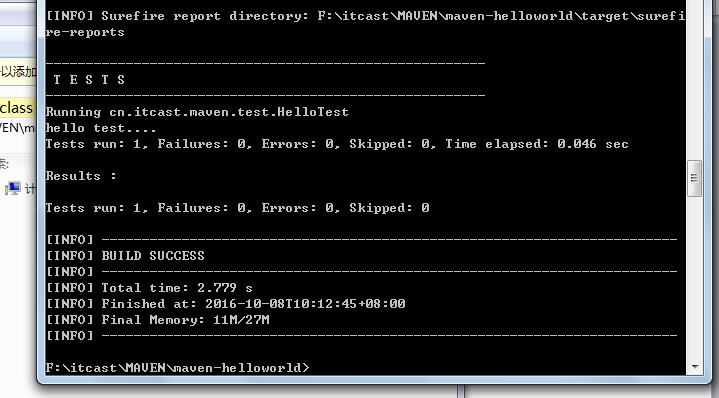
Clean 清理编译的文件



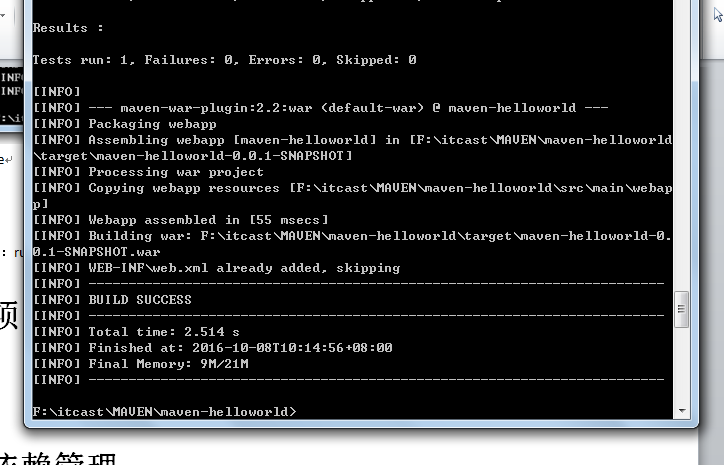
Compile 编译了主目录的文件



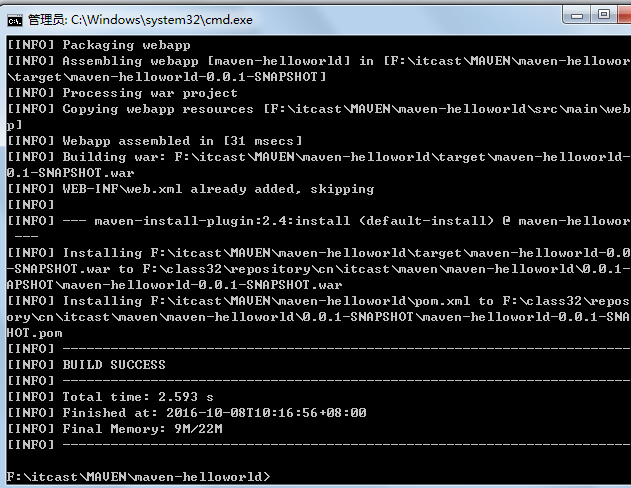
Test 编译并运行了test目录的代码



Package 打包



Install 就是把项目发布到本地仓库



Tomcat：run 一键启动

## 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

## 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

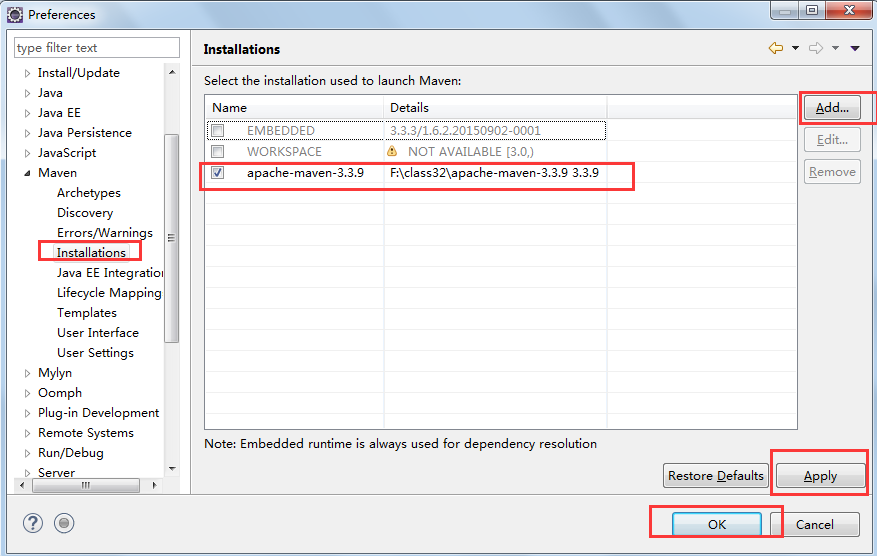
Mvn clean package

# 项目构建

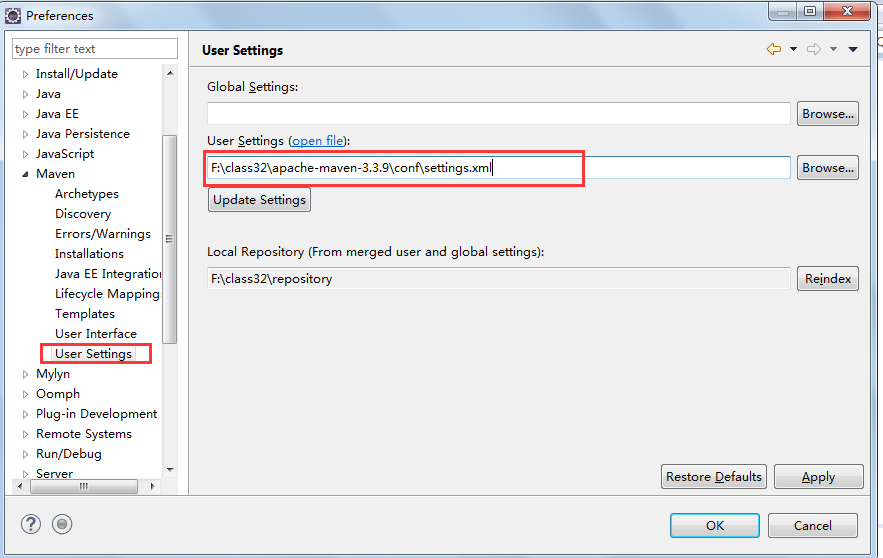
M2e插件的安装

Maven在eclipse的配置

1、选择3.3.9版本的maven软件

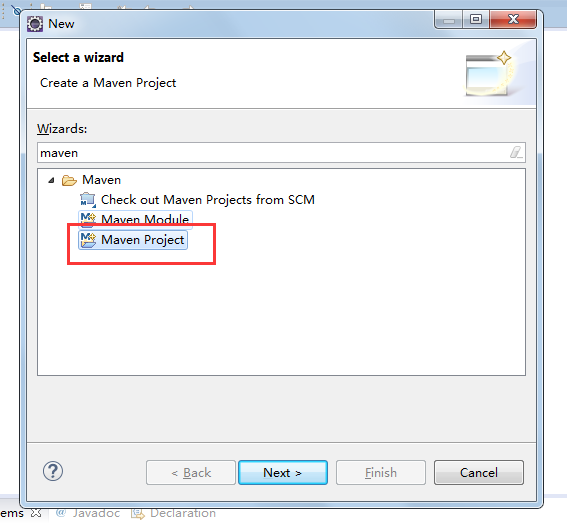


2、修改默认的本地仓库地址

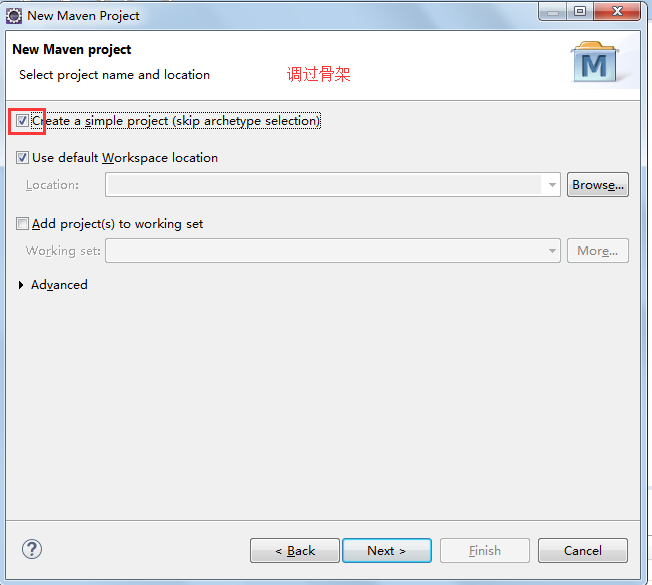


项目构建

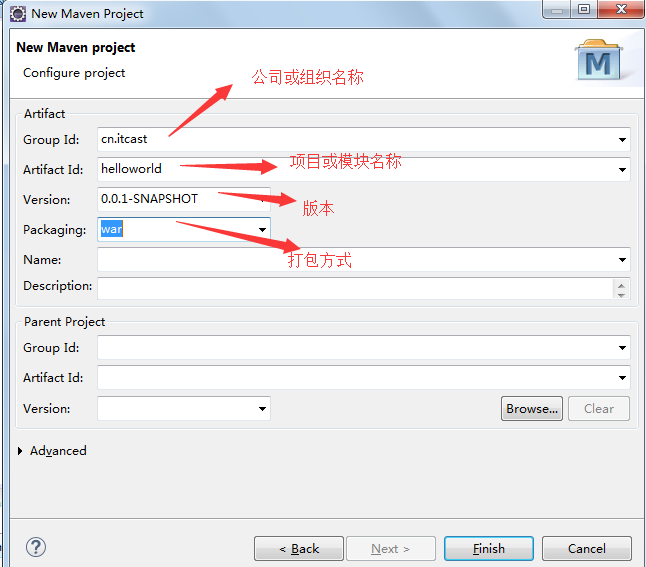
1. 新建maven项目



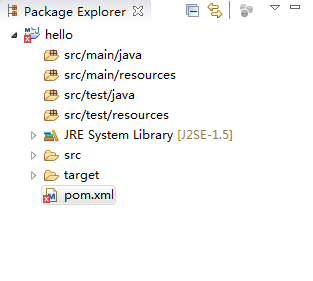
2、调过骨架，如果不跳过骨架选择创建出的项目目录是不全的



1. 填写坐标

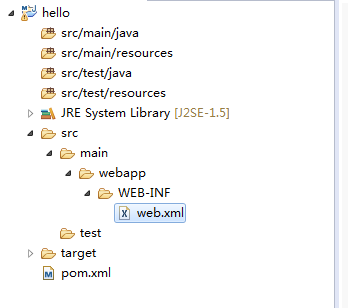


3、创建工程



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>

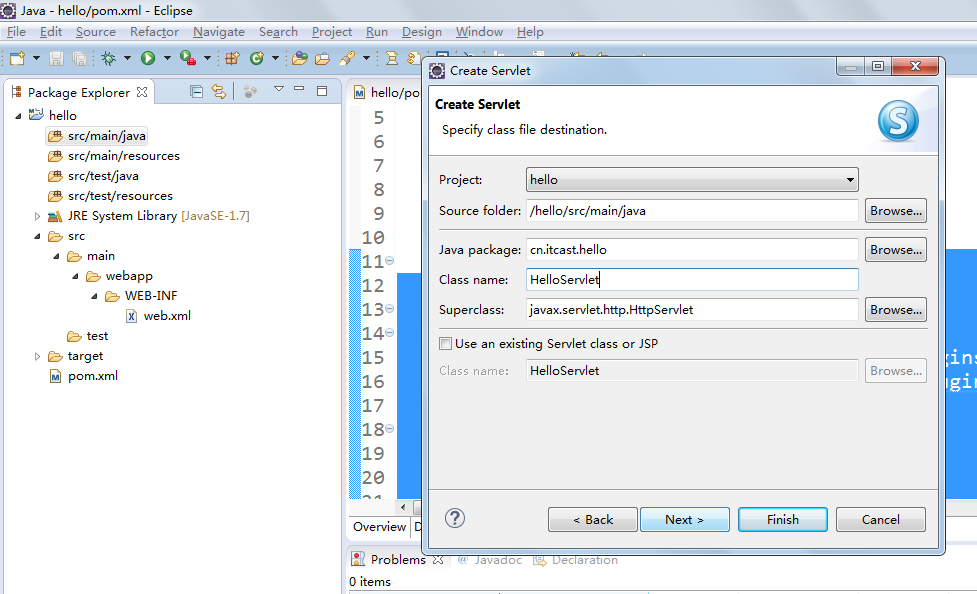
</configuration>

</plugin>

</plugins>

</build>

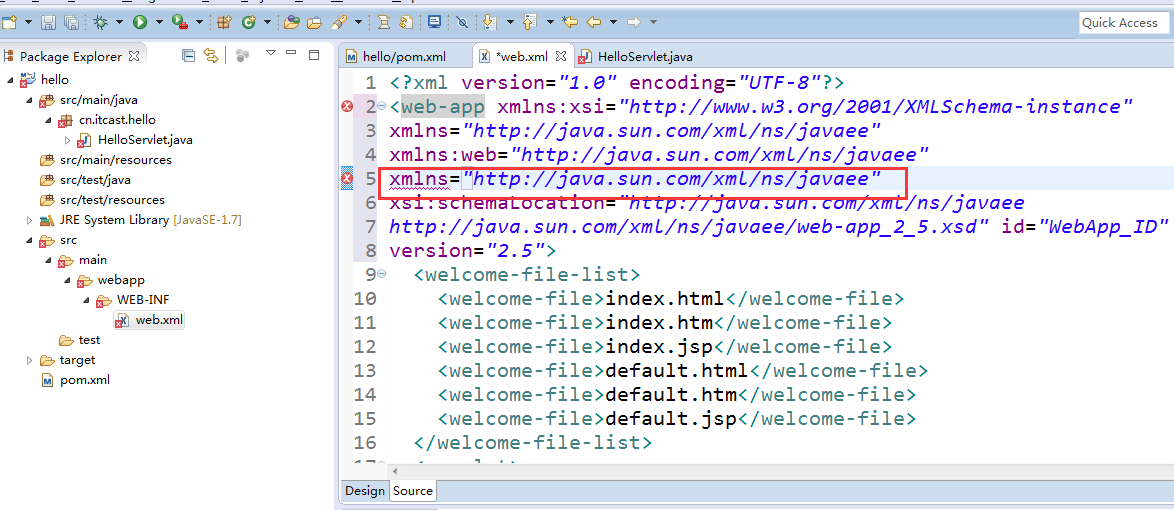
5、创建一个servlet



修改web.xml

删除重复的代码

xmlns=<http://java.sun.com/xml/ns/javaee>



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>

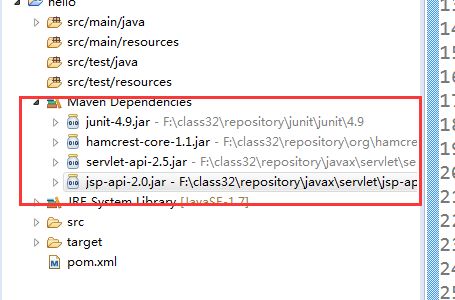
<version>2.0</version>

<scope>provided</scope>

</dependency>

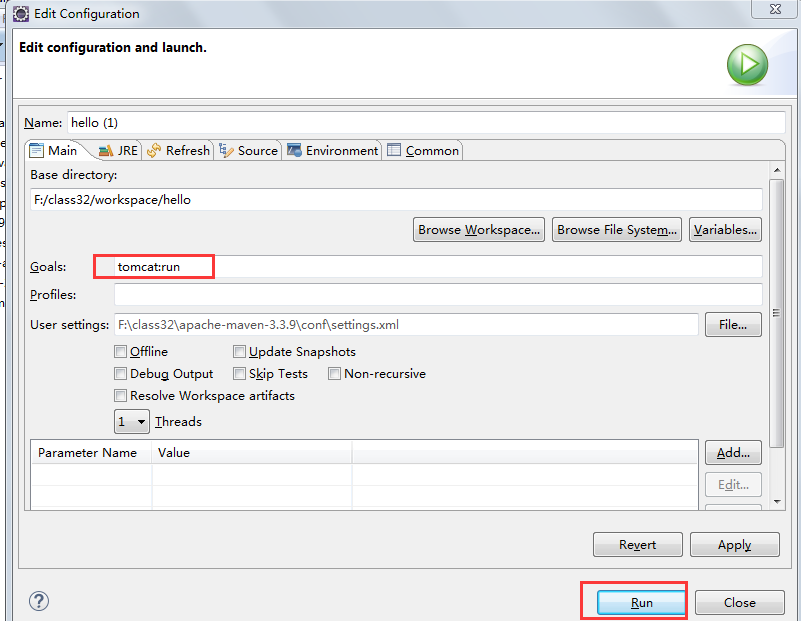
</dependencies>

会发现jar包已添加到项目中



启动项目

右击项目---run as –maven build ….



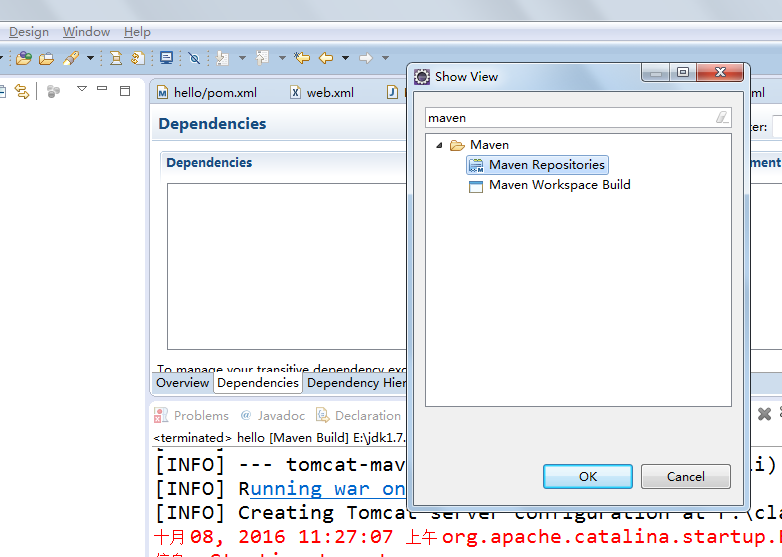
# 依赖管理

Jar包的管理

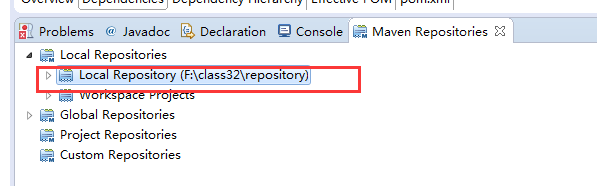
需求：整合struts2 页面上传一个客户id 跳转页面

## 添加依赖：

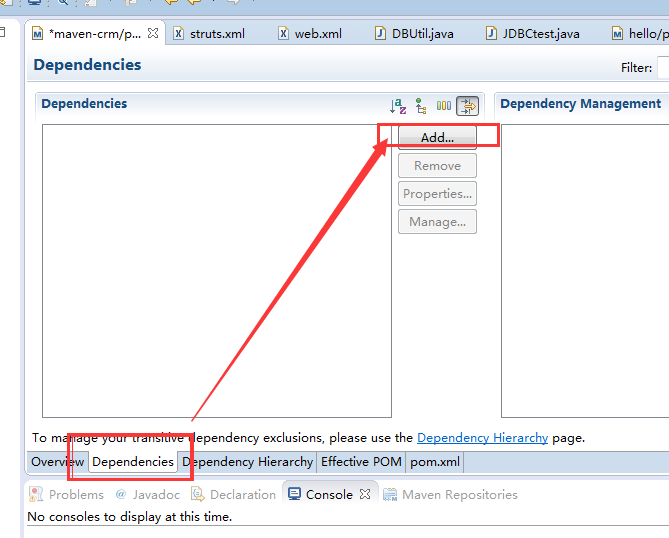
打开maven仓库的视图：



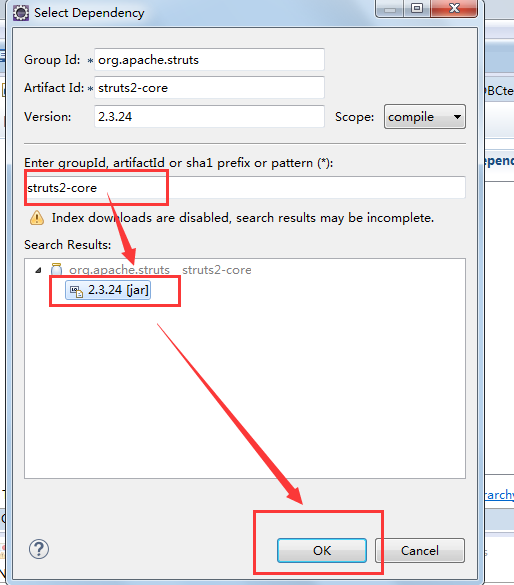
## 重建索引



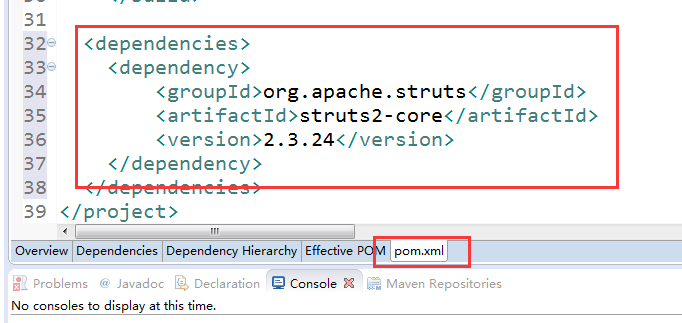
1. 创建maven项目（同上）
2. 跳过骨架（同上）
3. 填写坐标信息（同上）
4. 添加web.xml文件（同上）
5. 修改编译版本（同上）
6. 添加坐标 选择Dependencies标签 点击add



1. 手动输入要添加的坐标，选择版本



1. 可以看到 pom.xml文件中多出了如下代码



1. 同样的方式添加servlet-api.jar和jsp-api.jar **注意选择scope为provided**
2. 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;

}

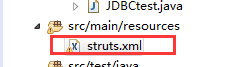
**public** String findById(){

**return** ***SUCCESS***;

}

}

1. 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=*"findById"*>

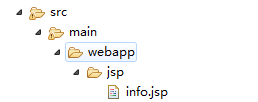
<result name=*"success"*>/jsp/info.jsp</result>

</action>

</package>

</struts>

1. 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>

## 依赖范围

### Compile struts2-core

编译（compile）时需要 测试时需要，，运行时需要，打包时需要

### 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

### Runtime 数据库驱动包

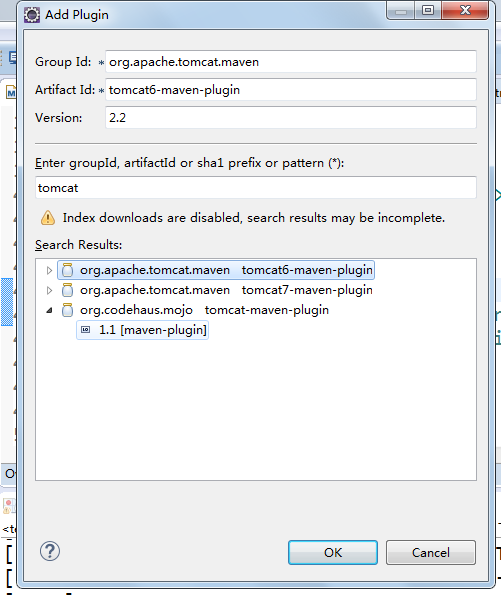
编译时不需要，测试时需要，，运行时需要，打包时需要

### 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：

