Maven入门

Apache开源项目，是项目的构建工具。用来进行依赖管理。

# Maven优点

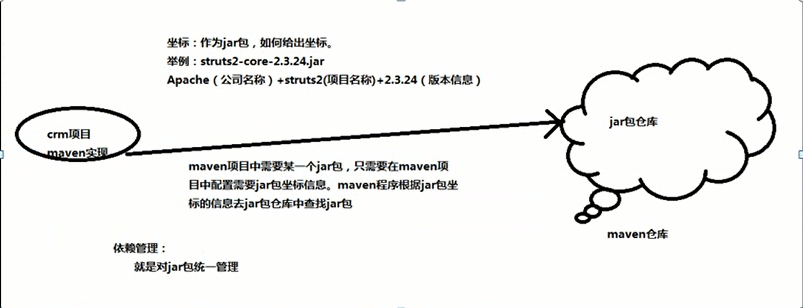


图 1‑1 Maven找jar包的过程

# Maven优点的实现

## 依赖管理

对jar包的管理

## 项目构建

项目编码完成后，对项目进行编译、测试、打包、部署（通过命令来实现）

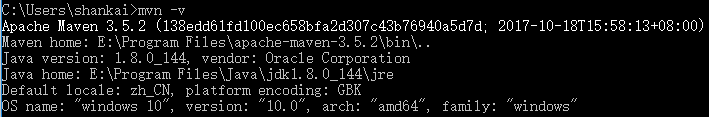
# Maven安装、配置本地仓库

Jdk是前提

## Maven的下载安装

### 配置环境变量

### 查看安装情况



## 配置本地仓库

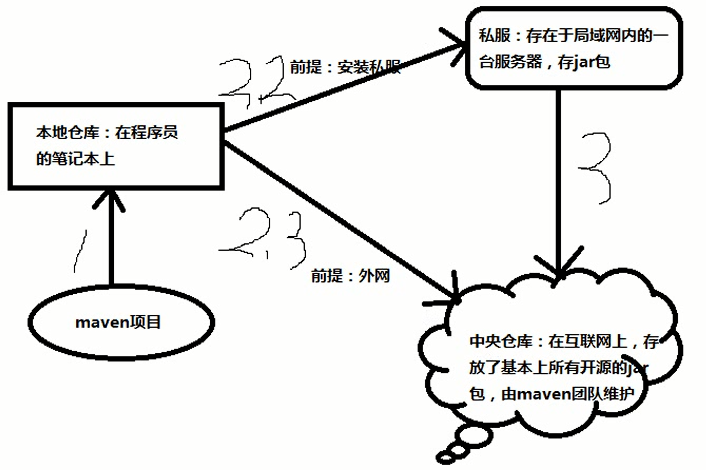
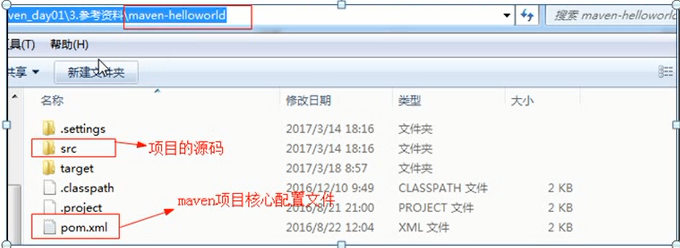


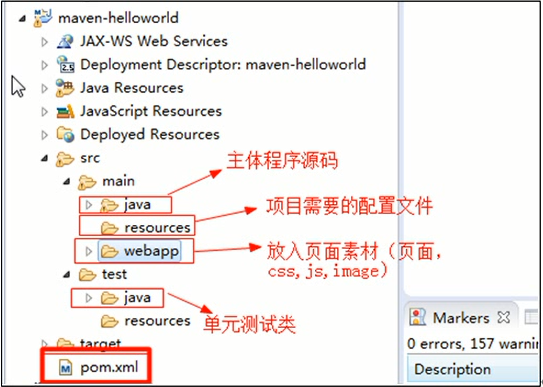
图 3‑1 仓库的类型

### 配置仓库路径

<localRepository>/path/to/local/repo</localRepository>

# Maven项目标准的目录结构





# Maven的常用命令

## clean

## complie

## test

单元测试的类名有要求，\*\*\*Test.java

## package

web project 🡪war

java project 🡪jar

将项目打包到项目根目录下的target目录下面

## install

打包到本地仓库

## maven项目的生命周期

在maven中存在“三套”生命周期，没一套生命周期相互独立，互不影响。在同一套生命周期内，执行后面的命令，前面的命令会自动执行。

cleanLifeCycle：清理生命周期

clean

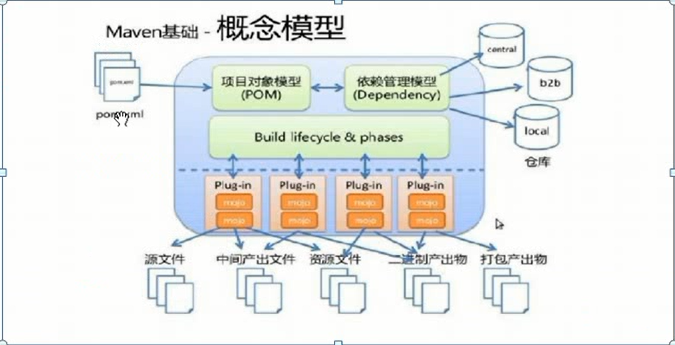
defaultLifeCycle：默认生命周期

complie test package install deploy

siteLifeCycle：站点生命周期

site

## 概念模型

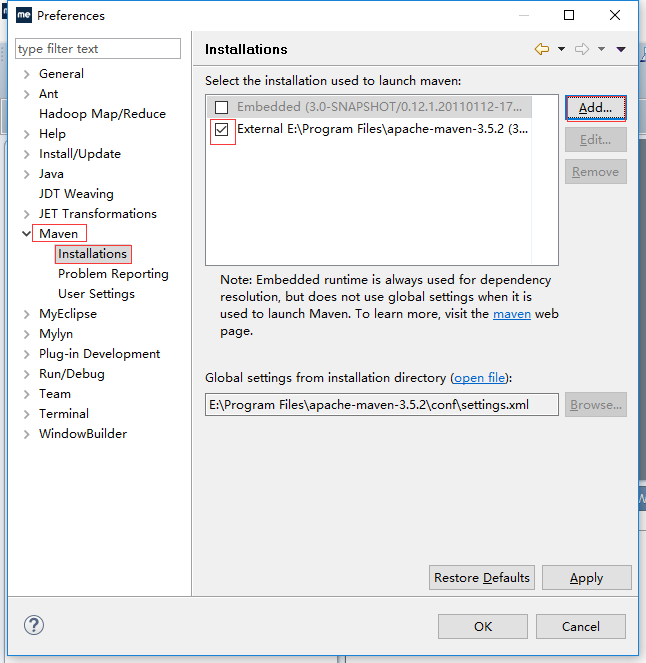


# maven整合web项目

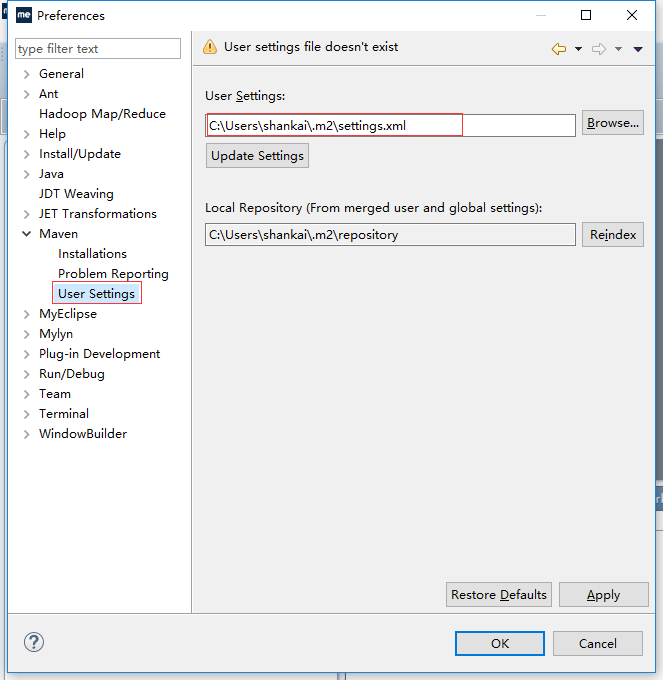
## eclipse中配置maven环境

### 配置m2e插件

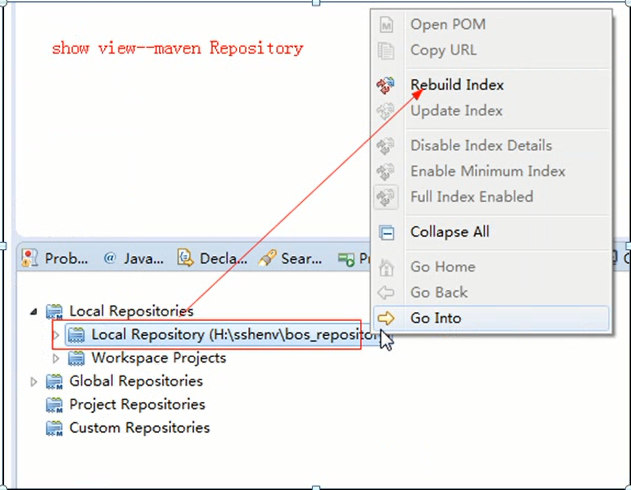
### 配置maven



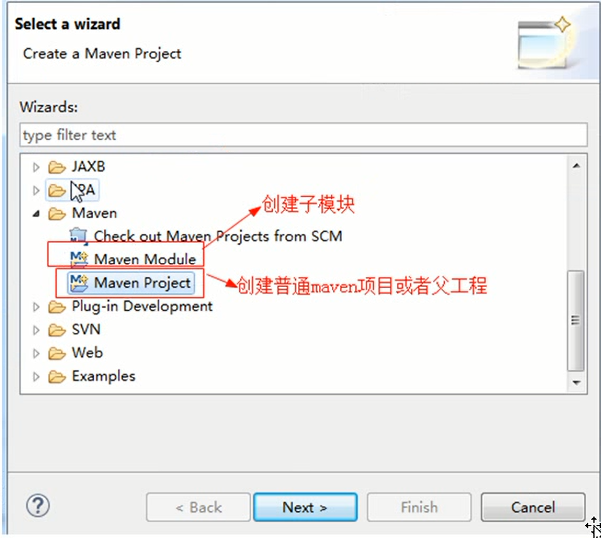
### 指定settings文件

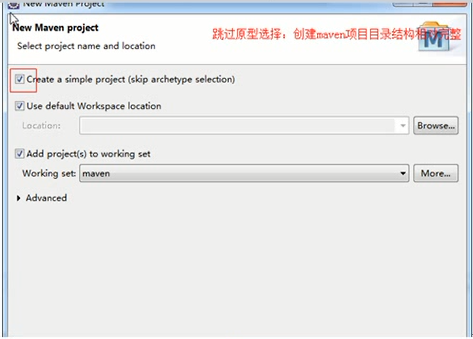


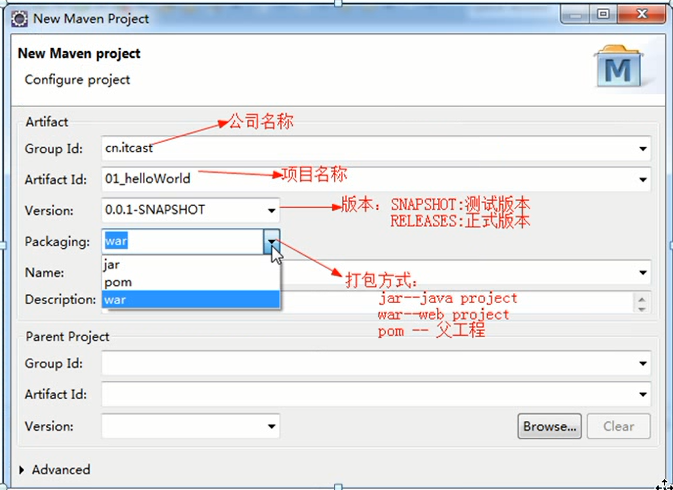
### 构建索引



## Maven整合servlet

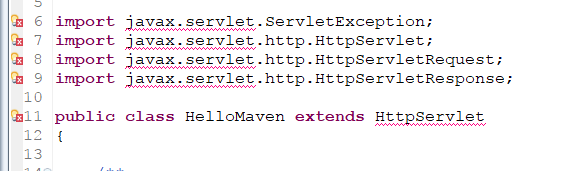


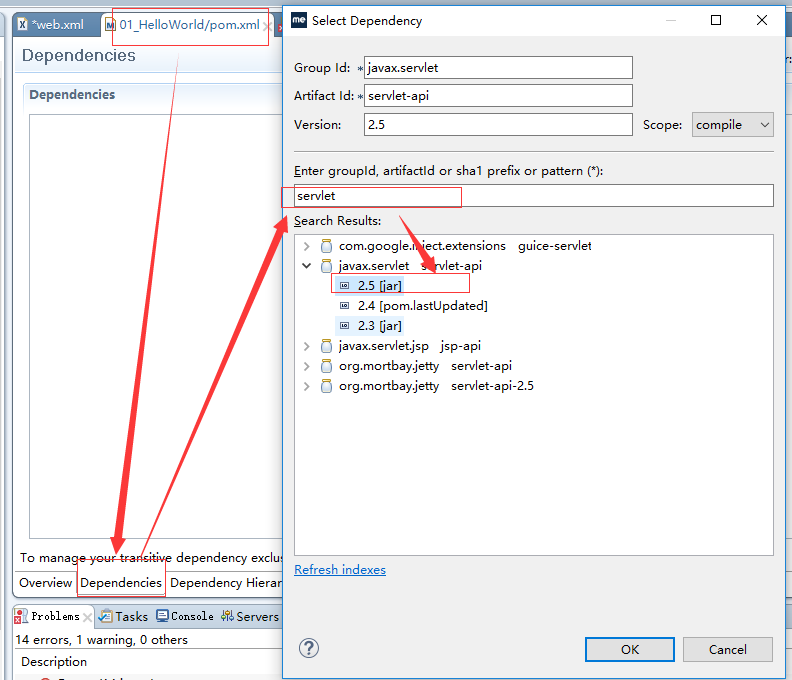




### 查找依赖

现象：创建servlet之后编译报错，原因缺少jar包



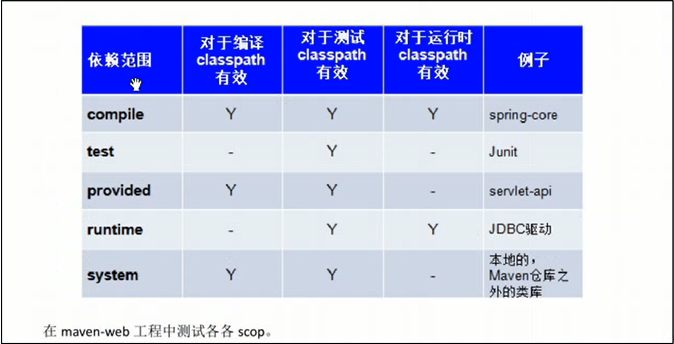


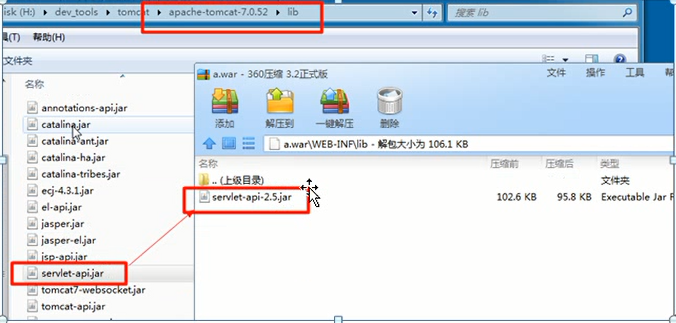
### 依赖范围

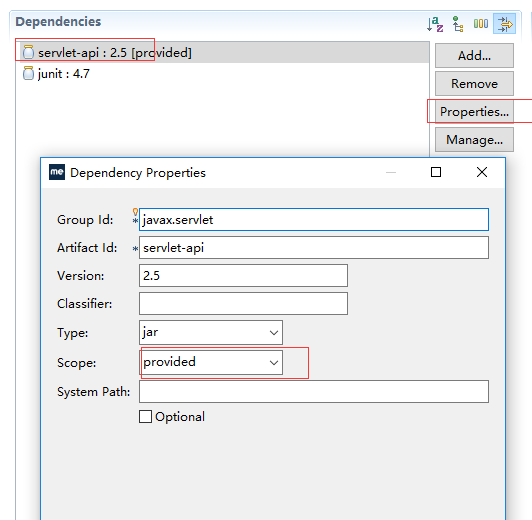
添加的依赖范围默认的就是compile

Provided：表示部署到tomcat的时候就不需要了

如果将servlet-api.jar设置为compile，打包后将包含servlet-api.jar,war包部署到tomcat之后会跟tomcat中存在的servlet-api.jar冲突，导致运行失败

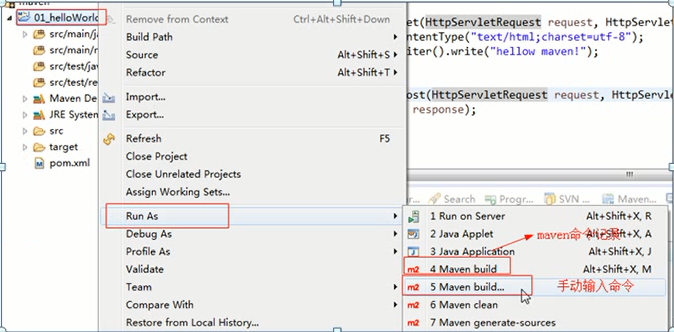




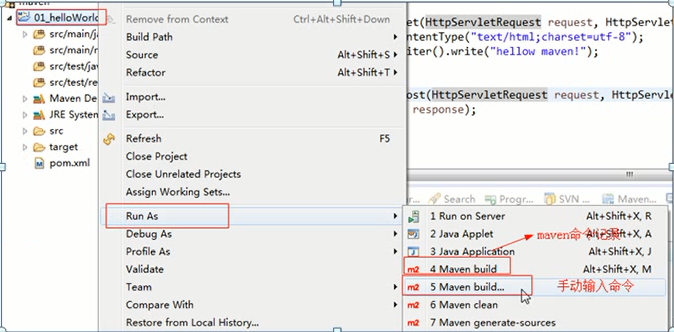


总结：如果使用到tomcat中自带的lib，将项目中的依赖项的作用范围设置为provided。尤其是：

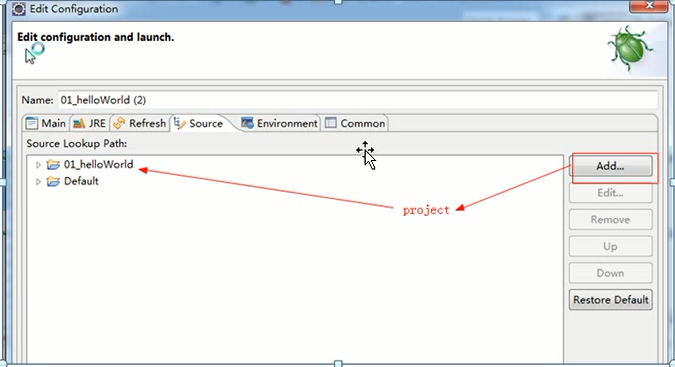




### 运行servlet



### 调试servlet



## Maven整合struts2

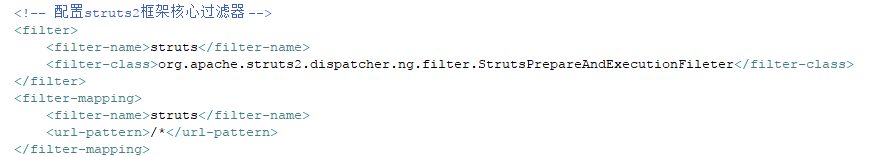
### 生成web.xml文件

### 添加struts2的依赖

### 创建struts.xml

### 创建action类

### 在web.xml中配置struts2框架核心过滤器



### 编写action中方法

### 在struts.xml中配置action节点

# 总结

## 安装

## 目录结构

ProjectName

Src

Main

Java

Resources

[webapp]

WEB-INF

Web.xml

Test

Java

Resources

Pom.xml

## Maven常用命令

Clean

Compile test package（项目根目录下target目录下） install（本地仓库）

## 使用eclipse开发maven项目

区别：不再拷贝jar包了；项目目录结构不同

## Pom.xml:项目对象模型

1. 项目坐标信息
2. Jdk版本信息
3. 项目运行的依赖信息