

# [今日课程大纲]

**Maven 简介**

**配置 Maven 运行环境**

**Maven 项目创建**

**Ant 和 Maven 项目简单对比**

**Maven 项目之间的关系**

**War 类型项目创建(tomcat 插件和资源拷贝插件)**

**分布式介绍**

**聚合项目演示**

**热部署**

# [知识点详解]

## 一.Ant

- 1.Ant 和 Maven 都是项目构建工具
- 2.在项目的运行编辑打包等过程都需要依赖于 ant 构建工具

## 二. 分布式

1. 传统项目部署:
  - 1.1 新建一个项目,通过不同的包区分不同的模块.
  - 1.2 把这—个项目发布服务器的 tomcat 中
2. 分布式项目部署.(适用于高负载情况下)

2.1 把一个完整的项目拆分成多个项目,把拆分后的项目分别部署到对应的服务器(tomcat)中的过程就叫做分布式项目部署.

2.1.1 把传统项目包换成一个单独的项目.

## 三.maven 简介

1.基于 Ant 的构建工具,Ant 有的功能 Maven 都有,额外添加了其他功能.

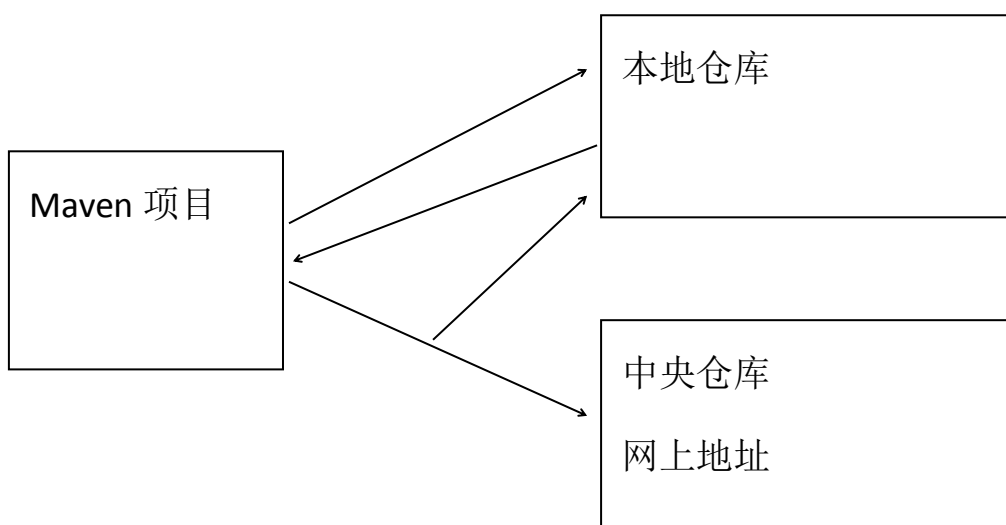
2.运行原理图.

2.1 本地仓库:计算机中一个文件夹,自己定义是哪个文件夹.

2.2 中央仓库:网上地址 <https://repo1.maven.org/maven2/>

2.2.1 下载速度慢--> 配置国内镜像

2.3 保证 JDK 版本和开发环境一致,如果不配置 1.4 或 1.5



### 3.坐标

3.1 每一 jar 文件都有一个唯一坐标.通过坐标可以精确确定是哪个 jar

#### 3.2 坐标组成

3.2.1 Group ID : 公司名.公司网址倒写

3.2.2 Artifact ID : 项目名

3.2.3 Version : 版本

### 4.pom

4.1 英文名称(Project Object Mode)

#### 4.2 项目对象模型

4.2.1 把 project 当作对象看待.

4.2.2 通过 maven 构建工具可以让对象(项目)和对象(项目)之间产生关系

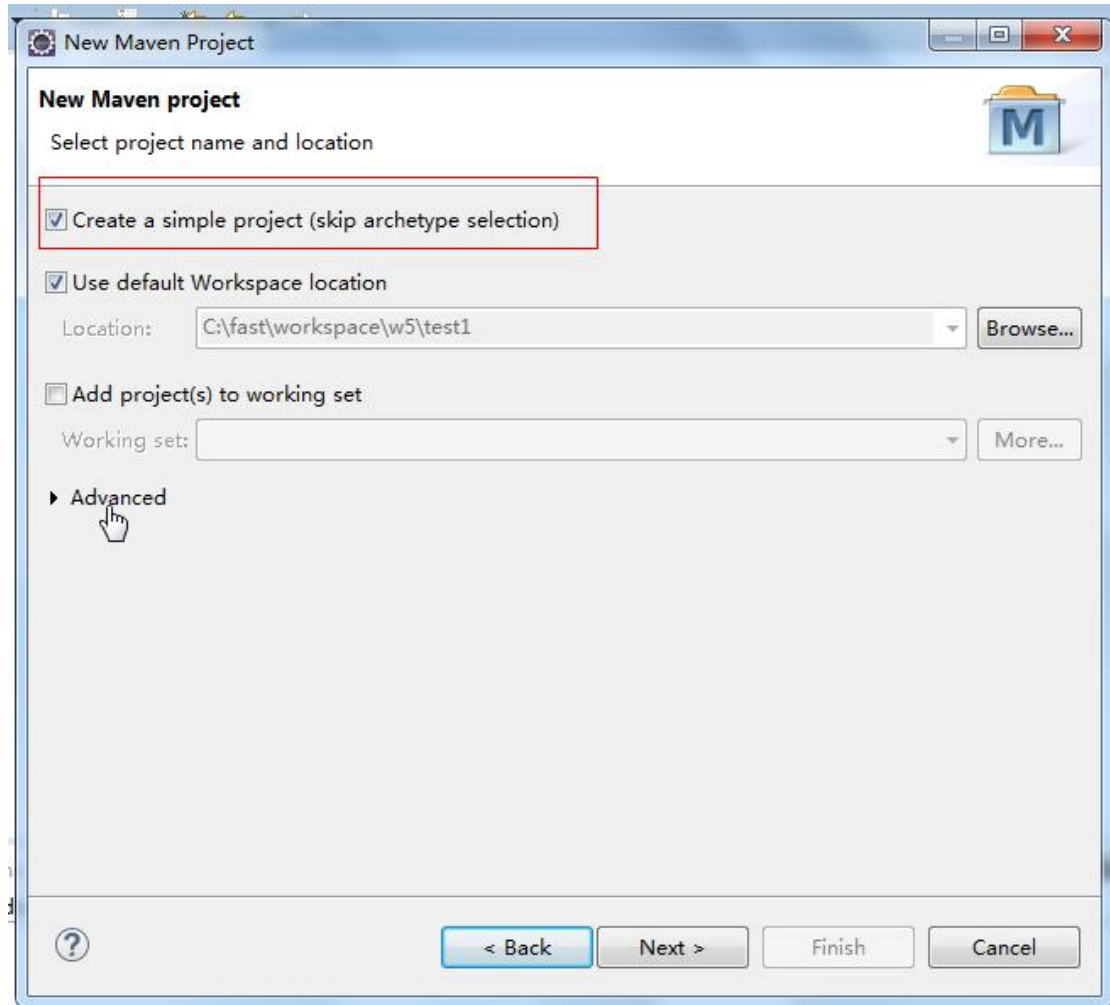
### 5.网址(查看坐标)

<http://mvnrepository.com/>

## 四.新建 Maven 项目

1. New--> Maven project

2. 勾选复选框,表示创建一个简单 Maven 项目(不应用任何模板)



### 3. 填写项目信息

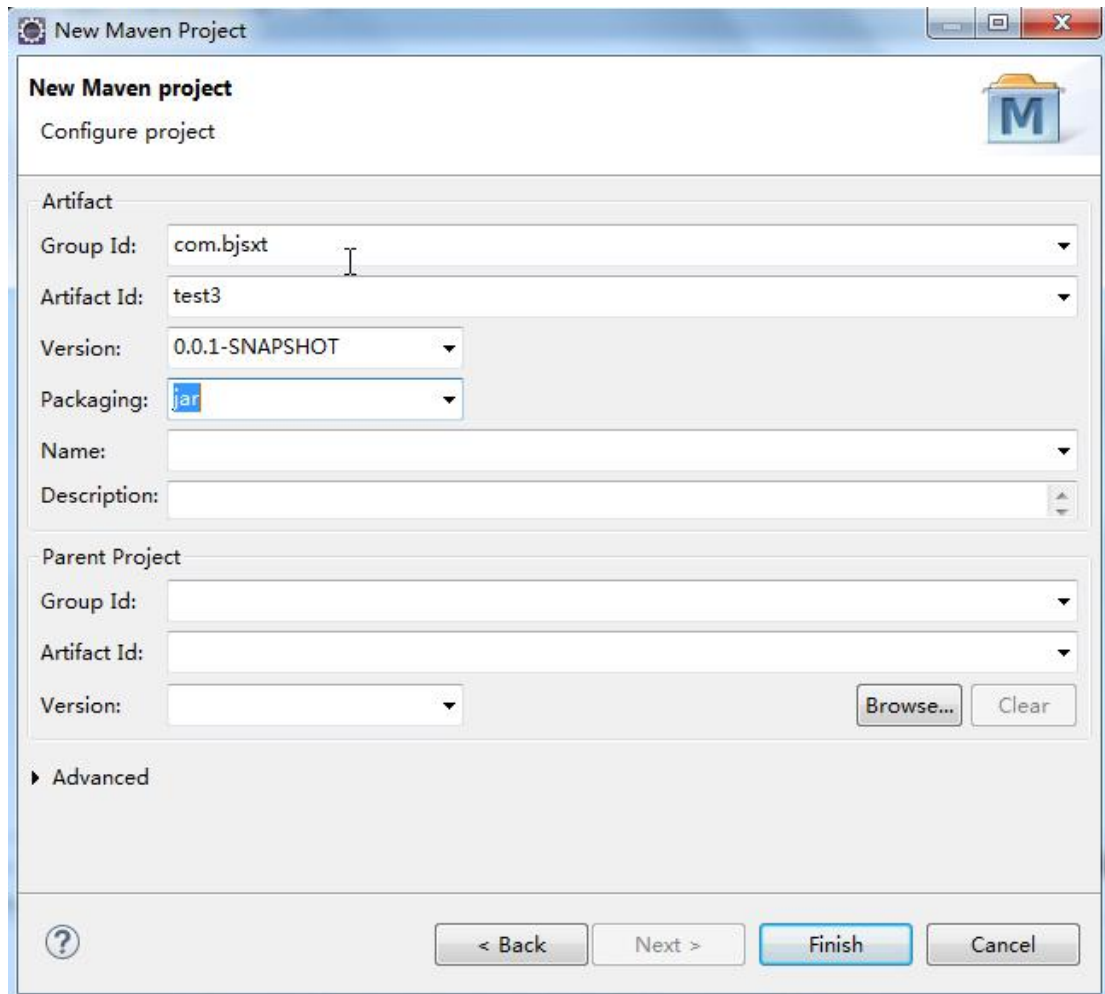
3.1 Packaging: 项目的类型,最终会被打包成什么类型.

3.1 jar : java 项目

3.2 war: web 项目

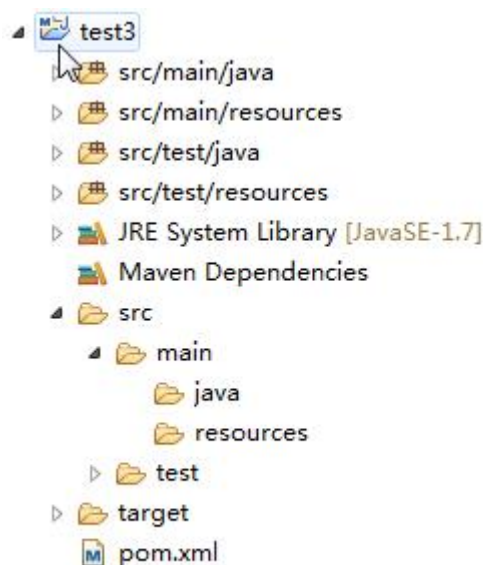
3.3 pom:逻辑父项目

3.3.1 只要一个项目有子项目必须是 pom 类型.



## 五. Maven 项目目录结构(jar 类型)

### 1. 截图



2. src/main/java 真实目录的快捷目录,写 java 代码
3. src/main/resources 快捷目录
  - 3.1 存放配置文件.
  - 3.2 虽然看见 resources 但是里面所有配置文件最终会被编辑放入到 classes 类路径.
4. src/test/java 写测试 java 代码
5. src/text/resources 测试的配置文件夹
6. pom.xml maven 的配置文件
  - 6.1 当前项目所依赖的其他项目或 jar 或插件等

## 六. Maven 项目之间的关系

1. 依赖关系
  - 1.1 标签<dependency>把另一个项目的 jar 引入到当前项目
  - 1.2 自动下载另一个项目所依赖的其他项目

## 2. 继承关系.

### 2.1 父项目是 pom 类型

2.2 子项目 jar 或 war,如果子项目还是其他项目的父项目,子项目也是 pom 类型.

### 2.3 有继承关系后,子项目中出现<parent>标签

2.3.1 如果子项目和<groupId>和<version>与父项目项目,在子项目中可以不配置<groupId>和<version>

2.4 父项目 pom.xml 中是看不到有哪些子项目.在逻辑上具有父子项目关系.

```
<parent>

  <groupId>com.bjsxt</groupId>

  <artifactId>parent</artifactId>

  <version>0.0.1-SNAPSHOT</version>

</parent>
```

## 3. 聚合关系.

3.1 前提是继承关系.父项目会把子项目包含到父项目中.

3.2 子项目的类型必须是 Maven Module 而不是 maven project

3.3 新建聚合项目的子项目时,点击父项目右键新建 Maven Module

3.4 具有聚合关系的父项目,在 pom.xml 中<modules>

```
<modules>

  <module>child2</module>
```

```
</modules>
```

### 3.5 具有聚合关系的子项目,在 pom.xml 中<parent>

```
<parent>

    <groupId>com.bjsxt</groupId>

    <artifactId>parent</artifactId>

    <version>0.0.1-SNAPSHOT</version>

</parent>
```

## 4. 聚合项目和继承项目区别

4.1 在语意上聚合项目父项目和子项目关系性较强

4.2 在语意上单纯继承项目父项目和子项目关系性较弱

## 5. <dependencyManagement> 写在父项目

5.1 作用:声明可能使用到的所有 jar

5.2 子项目中只需要有坐标的<groupid>和<artifactid>,<version>继承父项目

5.3 在父项目中<properties>把所有版本好进行统一管理

5.4 父项目 pom.xml

5.4.1 <properties>子标签名称自定义

5.4.2 \${名字} 引用标签的值

```
<properties>

    <spring-version>4.1.6.RELEASE</spring-version>

</properties>
```



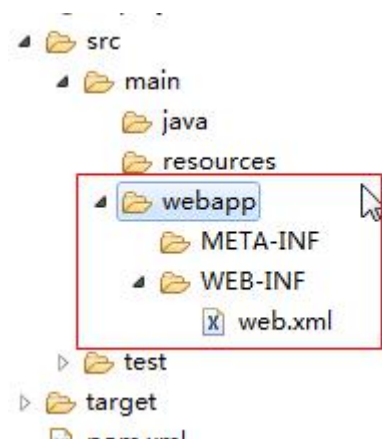
```
<dependencyManagement>
  <dependencies>
    <dependency>
      <groupId>org.springframework</groupId>
      <artifactId>spring-webmvc</artifactId>
      <version>${spring-version}</version>
    </dependency>
  </dependencies>
</dependencyManagement>
```

## 5.5 子项目

```
<dependencies>
  <dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-webmvc</artifactId>
  </dependency>
</dependencies>
```

# 七. 创建 war 类型项目

1. 创建 maven project 时选择 packaging 为 war
2. 在 webapp 文件夹下新建 META-INF 和 WEB-INF/web.xml



3. 在 pom.xml 中添加 java ee 相关的三个 jar

3.1 <scope> jar 有效范围 provided 表示编译期生效,不会打包发布到 tomcat 中

```
<dependencies>

    <dependency>

        <groupId>javax.servlet</groupId>

        <artifactId>javax.servlet-api</artifactId>

        <version>3.0.1</version>

        <scope>provided</scope>

    </dependency>

    <dependency>

        <groupId>javax.servlet.jsp</groupId>

        <artifactId>jsp-api</artifactId>

        <version>2.2</version>

        <scope>provided</scope>

    </dependency>
```

```
<dependency>

    <groupId>jstl</groupId>

    <artifactId>jstl</artifactId>

    <version>1.2</version>

</dependency>

</dependencies>
```

#### 4. 配置 tomcat 插件

```
<build>

    <plugins>

        <plugin>

            <groupId>org.apache.tomcat.maven</groupId>

            <artifactId>tomcat7-maven-plugin</artifactId>

            <version>2.2</version>

            <configuration>

                <!-- 控制 tomcat 端口号 -->

                <port>80</port>

                <!-- 项目发布到 tomcat 后的名称 -->

                <!-- / 相当于把项目发布名称为 ROOT -->

                <!-- /abc -->

                <path>/</path>

            </configuration>

        </plugin>

    </plugins>

</build>
```

```
        </plugin>

    </plugins>

</build>
```

5. 右键项目--> run as --> maven build-->Goals 中输入

clean tomcat7:run

## 八. 资源拷贝插件

1.maven 默认只把 src/main/resources 里面的非.java 文件进行编译到 classes 中

2.如果希望 src/main/java 下的文件也被编辑到 classes 中,在 pom.xml 中配置

2.1 下面内容配置在<build>中

```
<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>
```

## 九. 热部署

1. 修改 tomat/conf/tomcat-users.xml 添加

```
<role rolename="manager-gui"/>

<role rolename="manager-script"/>

<user          username="tomcat"          password="tomcat"
roles="manager-gui,manager-script"/>
```

2. 在 maven 项目的 pom.xml 中 tomcat 插件的<configuration>里配置

```
<plugins>

  <plugin>

    <groupId>org.apache.tomcat.maven</groupId>

    _____

    <artifactId>tomcat7-maven-plugin</artifactId>

    _____

    <version>2.2</version>

    _____

    <configuration>

      <!-- 控制 tomcat 端口号 -->
```

```
<port>80</port>

<!-- 项目发布到 tomcat 后的名称 -->

<!-- / 相当于把项目发布名称为 ROOT -->

<!-- /abc -->

<path>/jpk</path>

<username>tomcat</username>

<password>tomcat</password>

<url>http://192.168.139.128:8080/manager/text</url>

</configuration>

</plugin>

</plugins>
```

3. 右键项目--> run as --> maven build(以前写过,选择第二个) -->输入

3.1 tomcat7:deploy 第一次发布

3.2 tomcat7:redploy 不是第一次发布.