基础篇

一、安装

1. 官网下载解压
2. 系统变量新建`MAVEN_HOME`，路径为MAVEN安装目录
3. path变量配置`%MAVEN_HOME%\bin`
4. 测试安装结果：cmd输入`mvn -v`

二、 仓库种类和彼此的关系

maven仓库：本地仓库、中央仓库、远程仓库（私服）、中央仓库。

1. 默认情况maven工程通过jar包坐标查找本地仓库，本地没有去中央仓库下载。

```
<!-- localRepository
 | The path to the local repository maven will use to store artifacts.
 |
 | Default: ${user.home}/.m2/repository
<localRepository>/path/to/local/repo</localRepository>
```

2.公司中：maven去本地仓库找jar包，本地找不到去远程仓库查找，远程仓库没有，可以本地上传或者去中央仓库查找。

3.使用`<localRepository>/path/to/local/repo</localRepository>`

修改本地jar仓库路径。

三、maven标准目录结构

maven项目标准目录结构：

- src/main/java目录： 核心代码部分
- src/main/resources: 配置文件部分
- src/test/java目录： 测试代码部分
- src/test/testsources: 测试配置文件
- src/main/webapp: 页面资源（js、css、图片等）

四、 基本命令

`mvn clean`：删除上次编译信息。

`mvn compile`: 编译

- 生成targeet目录

`mvn test`: 编译测试代码

- 生成的target目录里test-classes目录
- 编译源码和测试代码

`mvn package`： 打包

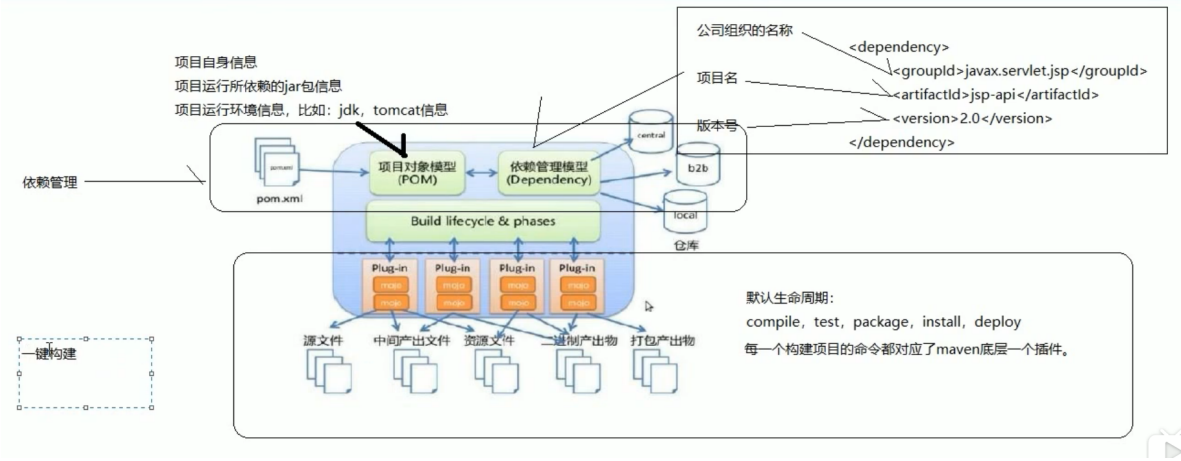
- java目录下的代码编译，test目录下的代码也被编译
- 生成war包

`mvn install`:

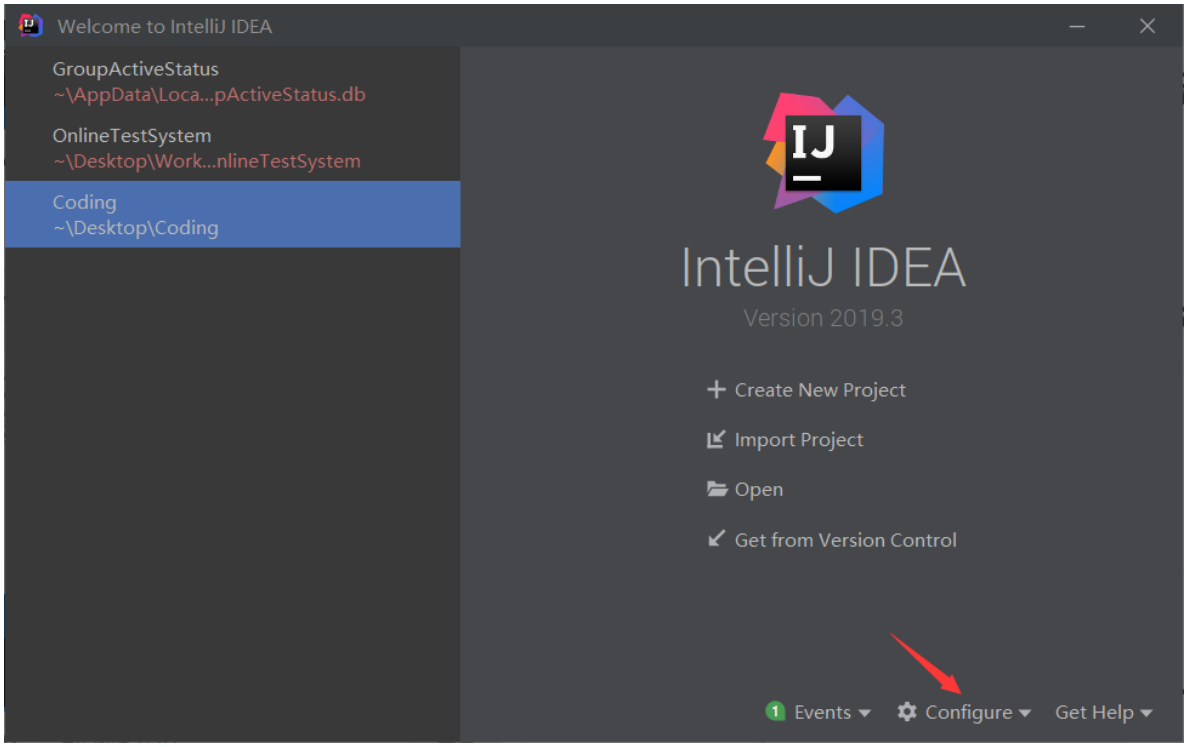
- 将项目打包
- 将项目jar包加入本地仓库

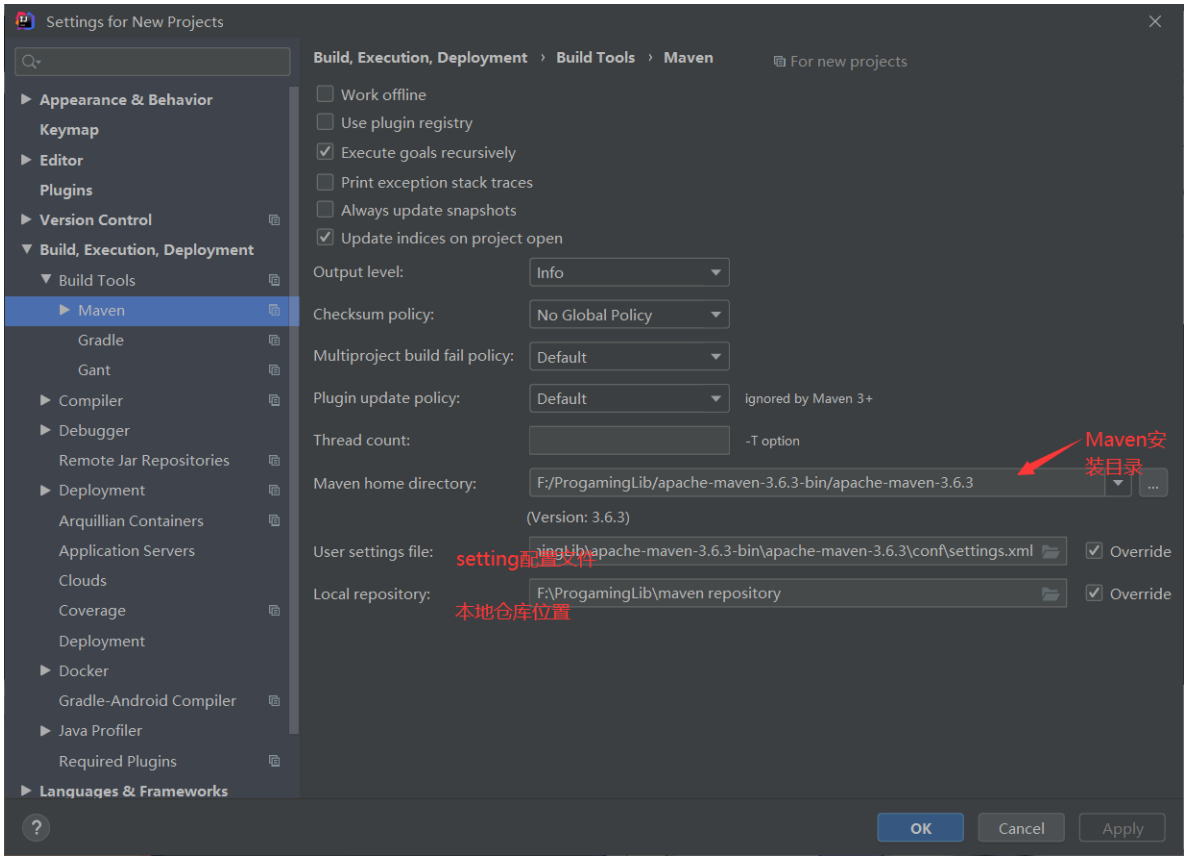
五、maven项目生命周期

六、maven概念模型图

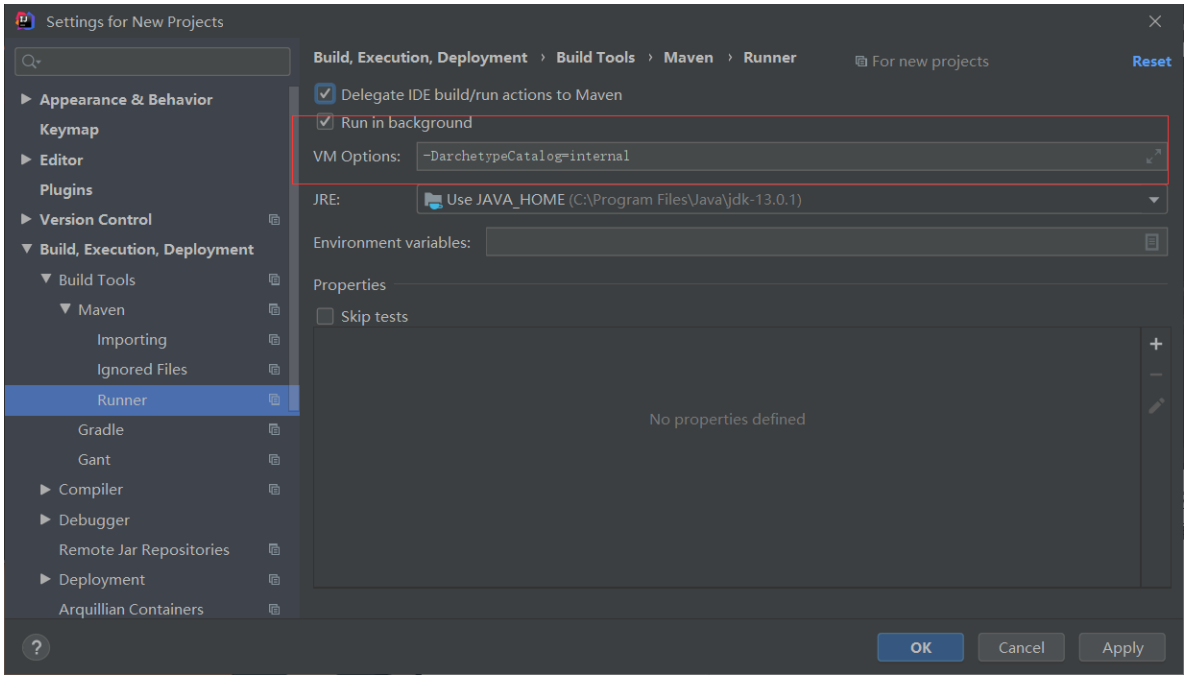


七、IDEA 集成Maven插件





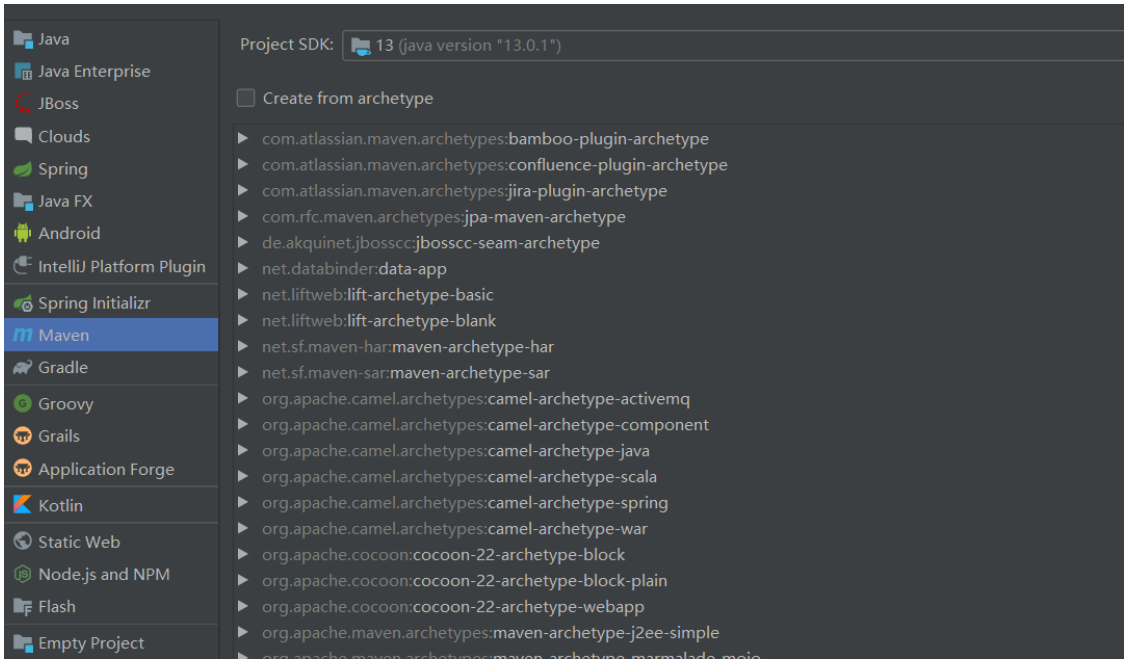
如果使用Maven骨架创建工程的话默认是需要联网的。配置此属性后，如果以前下载够配置工程相关的插件，就不用再联网（-DarchetypeCatalog=internal）。



八、 创建Java工程

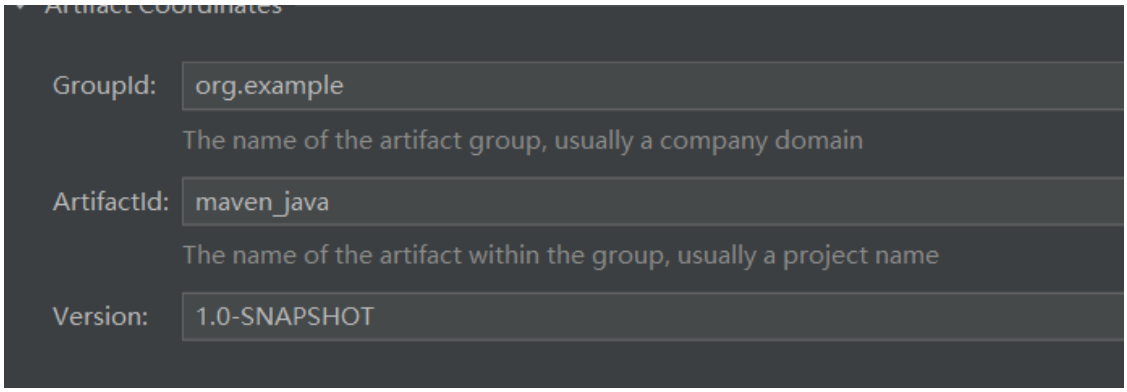
1.使用骨架创建Maven项目

1. 新建项目

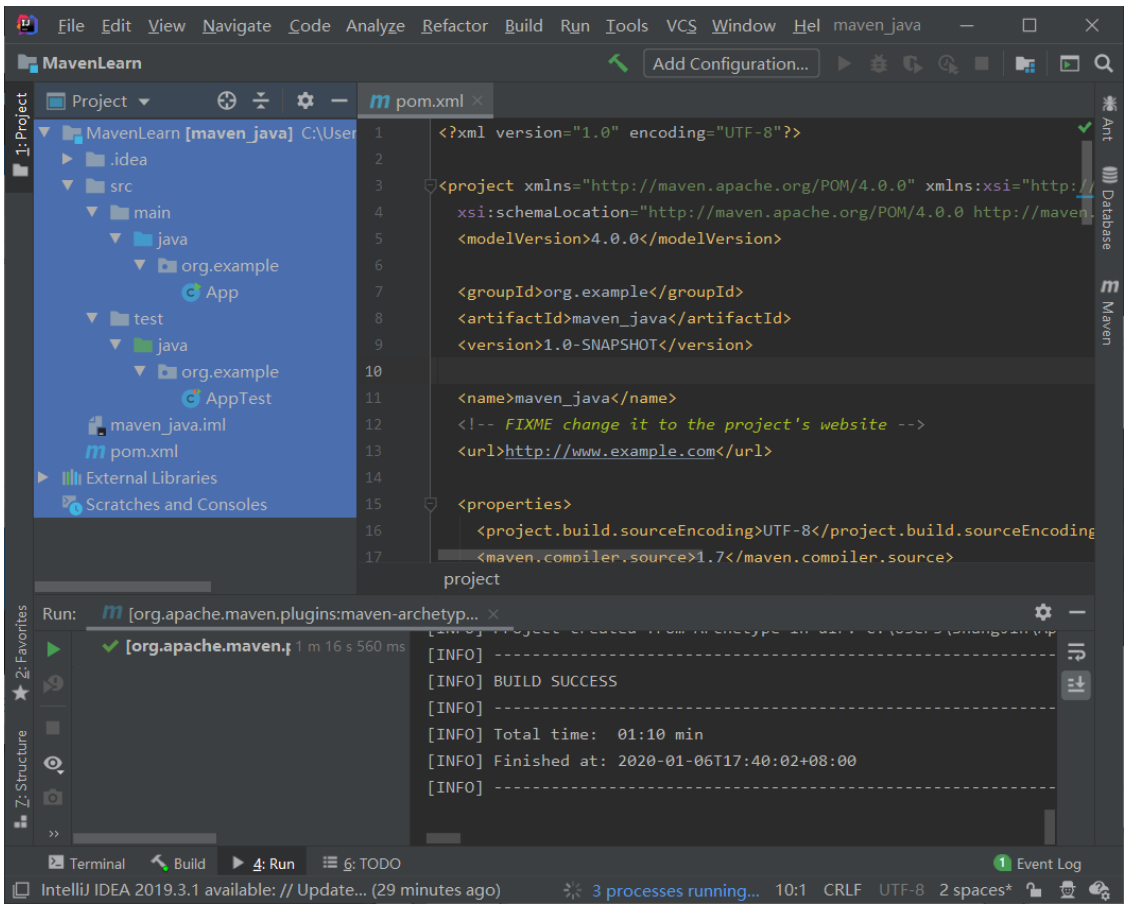


- 右侧为IDEA提供的骨架/模板（视频使用的quickstart模板）

2. 键入坐标



3. 具体包路径需要自己创建

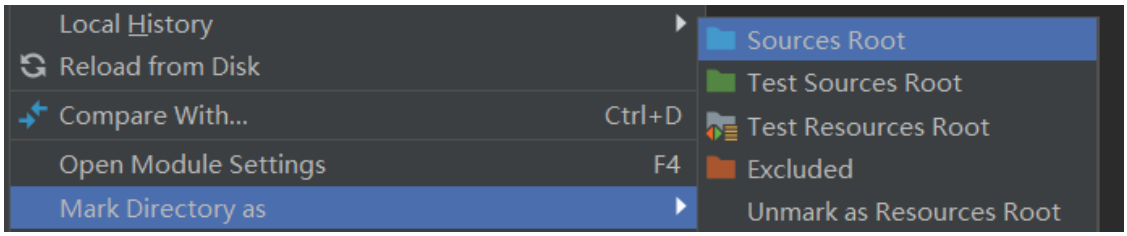


2.不适用骨架创建Maven项目（推荐）

在使用骨架创建Maven项目第一步时去掉勾选使用骨架创建选项即可。

3.使用骨架创建Web工程

1. 勾选以webapp结尾的骨架
2. 需手动在main目录下新建java目录，并右击java目录将该目录作为sources root



3. 本地仓库导入包文件

   在pom.xml文件中标签下加入 标签。

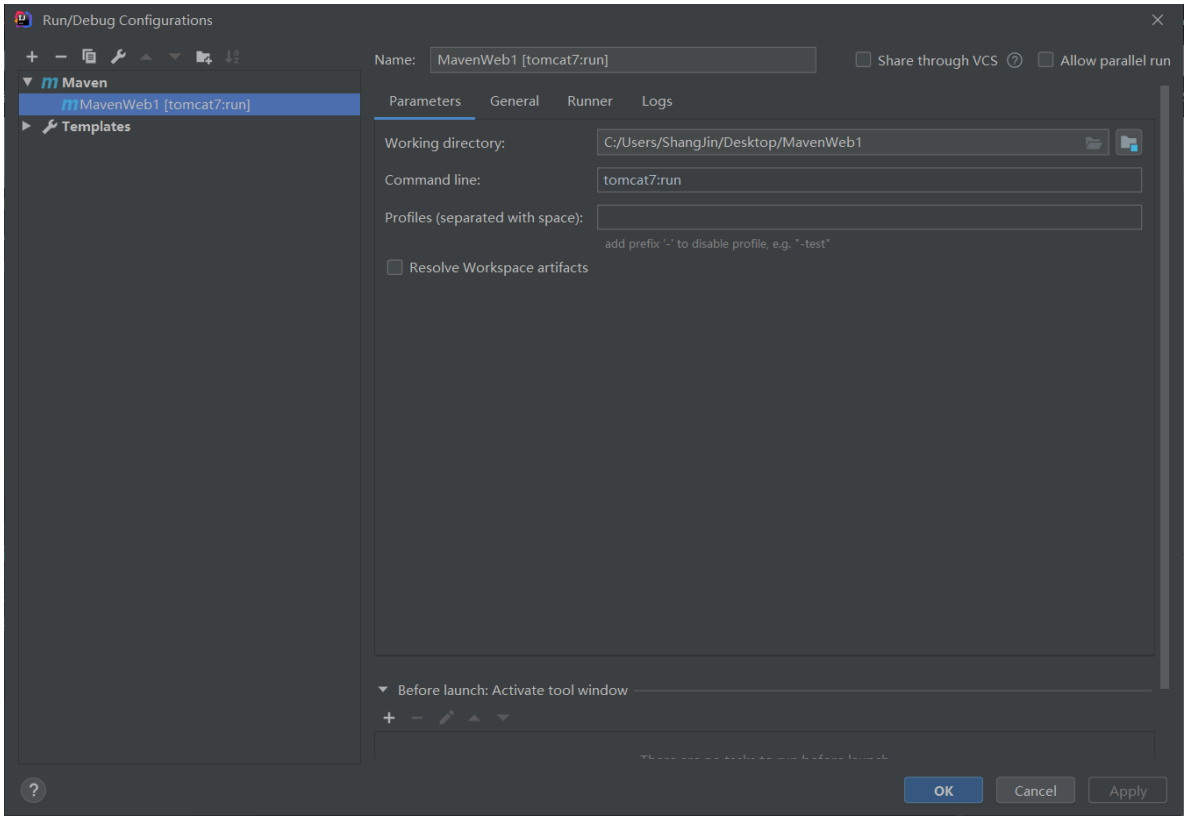
```
<!-- 加入依赖示例 -->
<!-- 添加servlet包 --->
<dependency>
  <groupId>javax.servlet</groupId>
  <artifactId>servlet-api</artifactId>
  <version>3.0-alpha-1</version>
</dependency>
<!-- 添加jsp ---->
<dependency>
  <groupId>javax.servlet.jsp</groupId>
  <artifactId>jsp-api</artifactId>
  <version>2.2</version>
</dependency>
```

4. IDEA使用Tomcat插件启动Maven项目

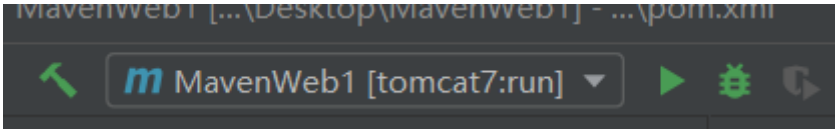
   在pom.xml文件标签中插入如下

```
<plugin>
<groupId>org.apache.tomcat.maven</groupId>
<artifactId>tomcat7-maven-plugin</artifactId>
<version>2.2</version>
<configuration>
  <path>/</path>
  <!-- 端口号 -->
  <port>
    8080
  </port>
</configuration>
</plugin>
```

配置Tomcat:



运行即可



4.运行产生的问题（jar包冲突）

本地Tomcat中的jsp包和servlet包与Maven导入的jar包冲突，运行时会出现错误。

解决：在冲突的包标签中加上

```
<scope>provided</scope>
```

标签。这样Maven导入的包只在写代码的时候起作用，项目运行时不起作用（Tomcat插件中的包起作用）。

同理

```
<scope>test</scope>
```

表示只在测试时起作用。（适合导入的测试类用的包）

九、 Maven输出中文乱码问题

加入

```
    <properties>
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
        <maven.compiler.encoding>UTF-8</maven.compiler.encoding>
        <java.version>1.8</java.version>
        <maven.compiler.source>1.8</maven.compiler.source>
        <maven.compiler.target>1.8</maven.compiler.target>
    </properties>
```