# Jar包打包插件maven-jar-plugin

## 应用背景

1. pom文件中<packaging>标签，设置打包类型为jar，该插件为扩展打包jar的功能。如果没有配置插件，只会将classes和resources打包成jar包。
2. 用于指定本项目生成的jar包中的MANIFEST.MF文件中的配置，如Class-Path和Main-Class

## Pom文件配置

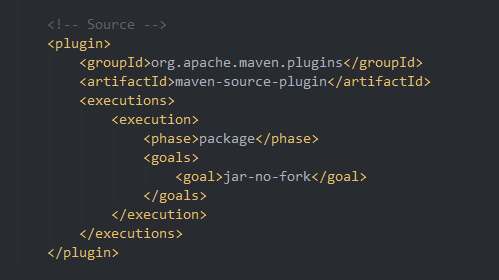


# 源码打包插件maven-source-plugin

## 应用背景

1. 在打包的时候，将源码打成source包,<packaging>标签不需要配置。

## 插件配置



# War包打包插件maven-war-plugin

## 应用背景

1. 配置<packaging>标签为war，用于web工程的打包。
2. 打包war包，java作为源码文件夹，resource作为静态资源，资源和源码打包在最上层目录。

## Pom文件配置



## servlet-api包注意事项

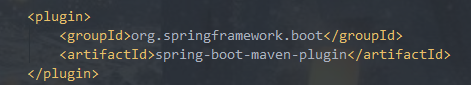
1. 由于servlet-api会和tomcat发生冲突，所以在引入依赖servlet-api的时候需要设置scope为provided，编译和测试时候有效。dependency-scope有四个值test(测试有效),compile(编译有效)，provided(编译和测试有效)，runtime(运行时依赖，打包只会将runtime打进去，其他三个都不会)

# Boot项目打包插件spring-boot-maven-plugin

## 应用背景

1. spring-boot-maven-plugin插件在Maven中提供了对Spring Boot的支持，可以帮助我们**打包出可执行的jar包或者war包**。其实spring-boot-maven-plugin所做的工作是在默认的maven-jar-plugin插件打包结束后，将项目依赖的jar包中的.class文件重新进行打包。

## Pom文件配置



1. 使用该插件打包会出现一个jar包，一个jar.original包。
2. jar.original是一个普通jar包，但是包含了之前打包的信息。