

maven3 创建项目并打包可执行 jar 包

作者: [fanhongwei](#)

原文链接: <https://ld246.com/article/1506396656353>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

参考地址:

[[maven3常用命令、java项目搭建、web项目搭建详细图解](#)]

[[Maven生成可以直接运行的jar包的多种方式](#)]

一、创建maven项目

创建java项目:

```
mvn archetype:generate -DgroupId=com.mycompany.app -DartifactId=my-app -DarchetypeArtifactId=maven-archetype-quickstart -DinteractiveMode=false
```

命令说明

mvn archetype:generate	固定格式
-DgroupId	组织标识 (包名)
-DartifactId	项目名称
-DarchetypeArtifactId	指定ArchetypeId, maven-archetype-quickstart, 创建一个Java project; maven-archetype-webapp, 创建一个Web Project
-DinteractiveMode	是否使用交互模式

标准的Maven项目结构

src/main/java 存放项目的源代码

src/test/java 存放测试源代码

如果要存放一些配置文件, 可以再建立一个目录src/main/resource存放, 如存放log4j.properties等

二、maven生成可执行jar包(多种方式)

Maven可以使用mvn package指令对项目进行打包, 如果使用java -jar xxx.jar执行运行jar文件, 会出现"no main manifest attribute, in xxx.jar" (没有设置Main-Class)、ClassNotFoundException (找不到依赖包) 等错误。

要想jar包能直接通过java -jar xxx.jar运行, 需要满足:

- 1、在jar包中的META-INF/MANIFEST.MF中指定Main-Class, 这样才能确定程序的入口在哪里;
- 2、要能加载到依赖包。

使用Maven有以下几种方法可以生成能直接运行的jar包, 可以根据需要选择一种合适的方法。

方式一、使用maven-assembly-plugin插件打包 (本人测试打包成)

maven项目创建成功后生成的pom.xml 原文件

```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/
MLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-
4_0_0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>hello</groupId>
  <artifactId>test</artifactId>
  <packaging>jar</packaging>
  <version>1.0-SNAPSHOT</version>
  <name>test</name>
  <url>http://maven.apache.org</url>
  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>3.8.1</version>
      <scope>test</scope>
    </dependency>
  </dependencies>
</project>
```

添加springframework框架(第二个dependency,根据需求自由配置)和打包插件(build标签内容)后的pom.xml

```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/
MLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-
4_0_0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>hello</groupId>
  <artifactId>test</artifactId>
  <packaging>jar</packaging>
  <version>1.0-SNAPSHOT</version>
  <name>test</name>
  <url>http://maven.apache.org</url>
  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>3.8.1</version>
      <scope>test</scope>
    </dependency>

    <dependency>
      <groupId>org.springframework</groupId>
      <artifactId>spring-context</artifactId>
      <version>4.3.11.RELEASE</version>
    </dependency>
  </dependencies>

  <build>
```

```
<plugins>
  <plugin>
    <groupId>org.apache.maven.plugins</groupId>
    <artifactId>maven-assembly-plugin</artifactId>
    <version>2.5.5</version>
    <configuration>
      <archive>
        <manifest>
          <mainClass>hello.Application</mainClass>
        </manifest>
      </archive>
      <descriptorRefs>
        <descriptorRef>jar-with-dependencies</descriptorRef>
      </descriptorRefs>
    </configuration>
  </plugin>
</plugins>
</build>

</project>
```

添加成功后,cmd中进入maven项目目录,执行命令

```
mvn package assembly:single
```

之后会自动将项目依赖打入jar包,

执行jar包: cmd进入项目目录下的target目录,执行命令:java -jar XXX.jar