# 1.SpringBoot Maven插件打包



# 2.maven-jar-plugin+maven-dependency-plugin打包

正常的main函数程序，我们直接package生成的jar包无法直接运行，因为没有再/META-INF/MANIFEST.MF中指定主类。

1.指定主类有很多方法，这里介绍下最简单的使用maven-jar-plugin插件的方法。

只需要加一条mainClass的配置，指定主类名即可；

这样不出意外可以运行。

2.如果我们的项目有外部依赖，即定义了dependency，那么默认不会把依赖打入jar包中，这样，如果我们直接jar -jar运行，就会报错，找不到class。

解决方案是：

第一步，为maven-jar-plugin添加外部依赖的路径；其实这里使用了maven-jar-plugin插件配置了/META-INF/MANIFEST.MF中的两个元素。

第二步，使用maven-dependency-plugin把依赖打入路径；

完整的pom：

这样执行mvn clean package以后，就可以直接java -jar运行了。

***<?xml version="1.0" encoding="UTF-8"?>***

***<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"***

***xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">***

***<modelVersion>4.0.0</modelVersion>***

***<groupId>com.liyao</groupId>***

***<artifactId>testMaven</artifactId>***

***<version>1.0-SNAPSHOT</version>***

***<name>testMaven</name>***

***<dependencies>***

***<dependency>***

***<groupId>org.apache.commons</groupId>***

***<artifactId>commons-lang3</artifactId>***

***<version>3.8.1</version>***

***</dependency>***

***</dependencies>***

***<build>***

***<plugins>***

***<plugin>***

***<groupId>org.apache.maven.plugins</groupId>***

***<artifactId>maven-jar-plugin</artifactId>***

***<configuration>***

***<archive>***

***<manifest>***

***<addClasspath>true</addClasspath>***

***<classpathPrefix>lib/</classpathPrefix>//****class所在resource的目录*

***<mainClass>com.liyao.App</mainClass>//入口***

***</manifest>***

***</archive>***

***</configuration>***

***</plugin>***

***<plugin>***

***<groupId>org.apache.maven.plugins</groupId>***

***<artifactId>maven-dependency-plugin</artifactId>//将第三方包打入lib/eureka***

***<executions>***

***<execution>***

***<id>copy-dependencies</id>***

***<phase>prepare-package</phase>***

***<goals>***

***<goal>copy-dependencies</goal>//复制第三方依赖***

***</goals>***

***<configuration>***

***<outputDirectory>${project.build.directory}/lib</outputDirectory><!--第三方依赖路径-->***

***<overWriteReleases>false</overWriteReleases>***

***<overWriteSnapshots>false</overWriteSnapshots>***

***<overWriteIfNewer>true</overWriteIfNewer>***

***</configuration>***

***</execution>***

***</executions>***

***</plugin>***

***</plugins>***

***</build>***

***</project>***

看下.MF文件：

Manifest-Version: 1.0

Archiver-Version: Plexus Archiver

Built-By: miracle

Class-Path: lib/commons-lang3-3.8.1.jar

Created-By: Apache Maven 3.5.3

Build-Jdk: 1.8.0\_101

Main-Class: com.liyao.App

这样是比较简单直接的一种配置可执行jar包的方法，既可以指定主类，也可以管理外部依赖的位置。

————————————————

版权声明：本文为CSDN博主「绝世好阿狸」的原创文章，遵循CC 4.0 BY-SA版权协议，转载请附上原文出处链接及本声明。

原文链接：<https://blog.csdn.net/u010900754/article/details/90184250>

3.