



链滴

# SpringBoot 应用打包

作者: [lucianlv](#)

原文链接: <https://ld246.com/article/1581225956837>

来源网站: [链滴](#)

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

# 单个 Jar 包

全部依赖集成到一个 jar 包中，且 jar 包可以直接执行。

## 方式一：parent 依赖

```
<?xml version="1.0" encoding="UTF-8"?>
<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/xsd/ma
en-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>2.1.1.RELEASE</version>
    <relativePath/> <!-- lookup parent from repository -->
  </parent>
  <groupId>geektime.demo</groupId>
  <artifactId>helloworld</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <name>HelloWorld</name>
  <description>Demo project for Spring Boot</description>

  <properties>
    <java.version>1.8</java.version>
  </properties>

  <dependencies>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-web</artifactId>
    </dependency>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-test</artifactId>
      <scope>test</scope>
    </dependency>
  </dependencies>

  <build>
    <plugins>
      <plugin>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-maven-plugin</artifactId>
      </plugin>
    </plugins>
  </build>

</project>
```

## 方式二：dependencyManagement

```
<?xml version="1.0" encoding="UTF-8"?>
<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/xsd/ma
en-4.0.0.xsd">

  <modelVersion>4.0.0</modelVersion>

  <groupId>geektime.demo</groupId>
  <artifactId>helloworld</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <name>HelloWorld</name>
  <description>Demo project for Spring Boot</description>

  <properties>
    <java.version>1.8</java.version>
  </properties>

  <dependencyManagement>
    <dependencies>
      <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-dependencies</artifactId>
        <version>2.1.1.RELEASE</version>
        <type>pom</type>
        <scope>import</scope>
      </dependency>
    </dependencies>
  </dependencyManagement>

  <dependencies>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-web</artifactId>
    </dependency>

    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-test</artifactId>
      <scope>test</scope>
    </dependency>
  </dependencies>

  <build>
    <plugins>
      <plugin>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-maven-plugin</artifactId>
        <version>2.1.1.RELEASE</version>
      </plugin>
    </plugins>
    <executions>
      <execution>
        <goals>
          <goal>repackage</goal>
        </goals>
      </execution>
    </executions>
  </build>
</project>
```

```
    </execution>
  </executions>
</plugin>
</plugins>
</build>

</project>
```

## 分目录部署包

jar 包，依赖包，配置文件分目录独立部署，修改配置文件无需重新打包。

## appassembler-maven-plugin

### pom 中配置

```
<plugin>
  <groupId>org.codehaus.mojo</groupId>
  <artifactId>appassembler-maven-plugin</artifactId>
  <version>1.10</version>
  <configuration>
    <configurationDirectory>conf</configurationDirectory>
    <configurationSourceDirectory>src/main/resources</configurationSourceDirectory>
    <copyConfigurationDirectory>true</copyConfigurationDirectory>
    <includeConfigurationDirectoryInClasspath>true</includeConfigurationDirectoryInClass
ath>
    <assembleDirectory>${project.build.directory}/app</assembleDirectory>
    <extraJvmArguments>-Xms128m</extraJvmArguments>
    <binFileExtensions>
      <unix>.sh</unix>
    </binFileExtensions>
    <platforms>
      <platform>windows</platform>
      <platform>unix</platform>
    </platforms>
    <repositoryName>lib</repositoryName>
    <programs>
      <program>
        <id>app</id>
        <mainClass>com.app.Main</mainClass>
        <jvmSettings>
          <extraArguments>
            <extraArgument>-server </extraArgument>
            <extraArgument>-Xmx2G</extraArgument>
            <extraArgument>-Xms2G</extraArgument>
          </extraArguments>
        </jvmSettings>
      </program>
    </programs>
  </configuration>
</plugin>
```

## maven 命令

```
mvn package appassembler:assemble
```

这样子就可以自动生成整个依赖文件，配置文件和运行脚本了。

配置说明

- configurationDirectory: 生成配置文件路径
- configurationSourceDirectory: 配置文件原路径，默认为 src/main/config
- assembleDirectory: 整体包目录
- extraJvmArguments: jvm 参数
- binFileExtensions: 生成脚本的后缀
- platforms: 生成哪几种平台
- repositoryName: 依赖包目录，默认 repo
- programs: 这个必须参数，启动的主 class

## jsw 生成

```
<plugin>
  <groupId>org.codehaus.mojo</groupId>
  <artifactId>appassembler-maven-plugin</artifactId>
  <version>1.2.1</version>
  <configuration>
    <repositoryLayout>flat</repositoryLayout>
    <repositoryName>lib</repositoryName>
    <includeConfigurationDirectoryInClasspath>true</includeConfigurationDirectoryInClasspath>
  </configuration>
</plugin>
<copyConfigurationDirectory>src/main/resources</copyConfigurationDirectory>
<target>${project.build.directory}</target>
<daemons>
  <daemon>
    <id>client</id>
    <mainClass>com.demo.client.App</mainClass>
    <commandLineArguments>
      <commandLineArgument>start</commandLineArgument>
    </commandLineArguments>
    <platforms>
      <platform>jsw</platform>
    </platforms>
    <generatorConfigurations>
      <generatorConfiguration>
        <generator>jsw</generator>
        <includes>
          <include>linux-x86-32</include>
          <include>linux-x86-64</include>
          <include>windows-x86-32</include>
          <include>windows-x86-64</include>
        </includes>
      </generatorConfiguration>
    </generatorConfigurations>
  </daemon>
</daemons>
</configuration>
```

```

    <property>
      <name>configuration.directory.in.classpath.first</name>
      <value>etc</value>
    </property>
    <property>
      <name>set.default.REPO_DIR</name>
      <value>lib</value>
    </property>
    <property>
      <name>wrapper.logfile</name>
      <value>../logs/wrapper.log</value>
    </property>
    <property>
      <name>run.as.user.envvar</name>
      <value>johndoe</value>
    </property>
  </configuration>
</generatorConfiguration>
</generatorConfigurations>
<jvmSettings>
  <initialMemorySize>256M</initialMemorySize>
  <maxMemorySize>512M</maxMemorySize>
  <systemProperties>
    <systemProperty>java.security.policy=conf/policy.all</systemProperty>
    <systemProperty>com.sun.management.jmxremote</systemProperty>
    <systemProperty>com.sun.management.jmxremote.port=8999</systemPro
erty>
    <systemProperty>com.sun.management.jmxremote.authenticate=false</sys
emProperty>
    <systemProperty>com.sun.management.jmxremote.ssl=false</systemPrope
ty>
  </systemProperties>
  <extraArguments>
    <extraArgument>-server</extraArgument>
  </extraArguments>
</jvmSettings>
</daemon>
</daemons>
</configuration>
<executions>
  <execution>
    <id>generate-jsw-scripts</id>
    <phase>package</phase>
    <goals>
      <goal>generate-daemons</goal>
    </goals>
  </execution>
</executions>
</plugin>

```