https://blog.csdn.net/abbc7758521/article/details/77987608

**Maven打包Spring boot，把依赖和配置文件及前端代码提取到jar文件外**

2017年09月15日 09:11:31 [缘灭由你](https://me.csdn.net/abbc7758521) 阅读数：5110

运用到maven如下插件：

maven-jar-plugin，负责将应用程序打包成可执行的jar文件   
maven-assembly-plugin，负责将整个项目按照自定义的目录结构打成最终的压缩包，方便实际部署

maven-compiler-plugin，负责编译项目

将依赖的jar提取到可运行的jar文件之外，使用maven-jar-plugin来实现

打包后目录如上，依赖都在lib文件夹中

代码目录结构如上，最终的可运行文件jar文件并不包含依赖的jar包，所有依赖的jar包都放在和XM003.jar平行的lib文件夹内，

这样部署的文件比较规整，如果有文件上传功能，程序可以找到前端目录，放置进去。

在pom文件中maven-jar-plugin的配置文件如下：

1. <plugin>
2. <groupId>org.apache.maven.plugins</groupId>
3. <artifactId>maven-jar-plugin</artifactId>
4. <version>2.6</version>
5. <configuration>
6. <archive>
7. *<!-- 添加index则不从mainfest中读取classpath，而是从Index.list中读取 -->*
8. *<!-- <index>true</index> -->*
9. <manifest>
10. <mainClass>com.sd.xm003.XM003Application</mainClass>
11. <addClasspath>true</addClasspath>
12. <classpathPrefix>lib/</classpathPrefix>
13. </manifest>
14. <manifestEntries>
15. <Class-Path>./</Class-Path>
16. </manifestEntries>
17. </archive>
18. <excludes>
19. <exclude>mapper/\*\*</exclude>
20. <exclude>static/\*\*</exclude>
21. <exclude>templates/\*\*</exclude>
22. <exclude>\*.txt</exclude>
23. <exclude>\*.xml</exclude>
24. <exclude>\*.properties</exclude>
25. </excludes>
26. </configuration>
27. </plugin>

其中manifest的部分是核心，在可执行的jar文件中，打包后会在jar文件内的META-INF文件夹下，生成一个MANIFEST.MF文件，里面记录了可执行文件的一些相关配置，比如像上面一段代码中所配置的内容，这里面就配置了可执行jar文件未来读取classpath的相对目录位置在什么地方，以及引入的jar文件都有哪些，上面的配置就是classpath目录是./  
mainClass配置表示，哪个class作为程序的入口来执行   
addClasspath配置表示，是否将依赖的classpath一起打包   
classpathPrefix配置表示，依赖的classpath的前缀，也就是打包后生成的MANIFEST.MF文件里，引入的jar文件都会加上前缀，lib/，比如fastjson-1.2.7.jar，在mainfest文件里就会是lib/fastjson-1.2.7.jar   
excludes配置表示，排除哪些文件夹不被打包进去   
  
其实maven-jar-plugin主要就是配置了MANIFEST.MF这个文件而已，就是让可执行文件知道自己怎么执行，加载哪些文件执行的描述，剩下的工作交由maven-assembly-plugin来处理   
在pom文件中maven-assembly-plugin配置如下：

1. <plugin>
2. <artifactId>maven-assembly-plugin</artifactId>
3. <configuration>
4. *<!-- not append assembly id in release file name -->*
5. <appendAssemblyId>false</appendAssemblyId>
6. <descriptors>
7. <descriptor>src/main/resources/package.xml</descriptor>
8. </descriptors>
9. </configuration>
10. <executions>
11. <execution>
12. <id>make-assembly</id>
13. <phase>package</phase>
14. <goals>
15. <goal>single</goal>
16. </goals>
17. </execution>
18. </executions>
19. </plugin>

重点的就是package.xml的路径了，使用maven-assembly-plugin的相关配置实际上都在这个文件里面   
package.xml的文件内容 ：

1. <?xml version="1.0" encoding="UTF-8"?>
2. <assembly xmlns="http://maven.apache.org/plugins/maven-assembly-plugin/assembly/1.1.3" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3. xsi:schemaLocation="http://maven.apache.org/plugins/maven-assembly-plugin/assembly/1.1.3 http://maven.apache.org/xsd/assembly-1.1.3.xsd">
4. <id>package</id>
5. <formats>
6. <format>zip</format>
7. </formats>
8. <includeBaseDirectory>true</includeBaseDirectory>
9. <fileSets>
10. <fileSet>
11. <directory>bin</directory>
12. <outputDirectory>/</outputDirectory>
13. </fileSet>
14. <fileSet>
15. <directory>src/main/resources</directory>
16. <outputDirectory>/</outputDirectory>
17. </fileSet>
18. <fileSet>
19. <directory>${project.build.directory}</directory>
20. <outputDirectory>/</outputDirectory>
21. <includes>
22. <include>\*.jar</include>
23. </includes>
24. </fileSet>
25. </fileSets>
26. <dependencySets>
27. <dependencySet>
28. <outputDirectory>lib</outputDirectory>
29. <scope>runtime</scope>
30. *<!-- <unpack>false</unpack> -->*
31. <excludes>
32. *<!-- <exclude>${project.name}-${project.version}</exclude> -->*
33. <exclude>${groupId}:${artifactId}</exclude>
34. </excludes>
35. </dependencySet>
36. </dependencySets>
37. </assembly>

配置了，最终压缩的文件格式，为zip，也就是最终打包出来的是一个zip的文件，然后发布到服务器上进行解压部署，相关我要的配置都在这个压缩包内，解压即可直接使用

下面的fileSets中配置了我需要将那些文件打包到我的最终压缩包中，    
相关的配置文件src/main/resources，里面放着整个程序提取的properties等相关的配置文件   
最终可运行的jar文件，使用了${project.build.directory}变量，也就是通过maven-jar-plugin生成的那个jar文件   
dependencySets里面配置了依赖库最终输出到lib文件夹下，与上面的maven-jar-plugin配置生成的manifest文件路径相对应，这样可运行jar就会按照manifest的路径来找相应的文件进行加载 

 收藏