首先声明一点，该篇教程只是基于maven还是萌新的小白的我尝试了好多坑勉强让自己的工程打包成功的案例，不具备技术权威性，后期若有时间研究maven再来填坑（这个有时间就待定了，能学的辣么多不是，哪有时间专门研究maven）。

穿插一个打包前可能就会出现的智障问题，你问我为什么会出现，我不知道，这里只贴出解决方案。什么问题呢？工程运行时其实之前会将一大堆乱七八糟的.class啊，.xml呀，.properties呀等等这些文件塞到工程根路径的target目录下，这个过程我们专业说法就是编译，有时候会出现个智障问题，.xml等等会编译不过去，也就是好多人怎么找错都找不到的原因（Java开发得学会在编译路径找问题，2333）。辣么，怎么解决呢，既然我们在用maven开发，我们就用maven的方式去解决。简单地说，聚合工程中哪个模块出现了这个问题，就在那个模块的pom中的build下添加如下配置。

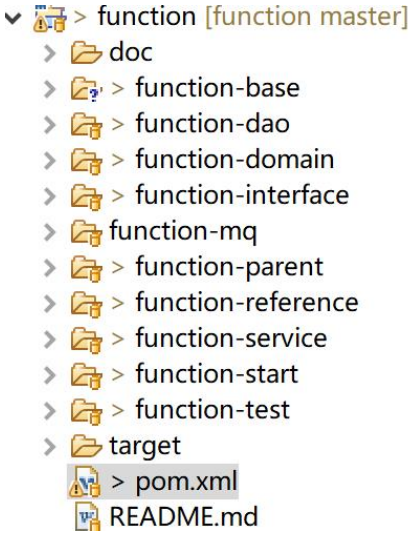
```
<resources>
 <resource>
  <directory>src/main/java</directory>
  <includes>
   <include>**/*.properties</include>
   <include>**/*.xml</include>
  </includes>
  <filtering>false</filtering>
 </resource>
 <resource>
  <directory>src/main/resources</directory>
  <includes>
   <include>**/*.properties</include>
   <include>**/*.xml</include>
  </includes>
  <filtering>false</filtering>
 </resource>
</resources>
```

简单解释下，主要配了俩东西，java和resources路径下的.properties和.xml就不会遇到上面描述的智障问题。我的意思呢是这样，如果遇到问题就加，哪里有问题加哪里，当然你乐意每个地方都加也没人阻拦。

好了，开始正题。单个工程打包可能没人需要教了，该篇是针对springboot的聚合工程打jar包，其它的聚合工程甚至打war的工程该篇可以不用找答案了，没写，可能有点参考性，愿意看就看呗。

嗯嗯，不废话了，辣么，开始说，一现在我们是要给一个springboot聚合工程打jar包，二我们要明确这个jar包打出来是要能运行能用的，所以说我们看似在给一个父工程打jar包，其实最终的jar就是你启动类所在那个模块的jar，因为所有依赖最终都是汇总到这里的。

辣么问题简单了，其实我们只需要给启动类模块配置打包插件，其它被依赖的自然而然会打的。展示一下我的测试项目结构。



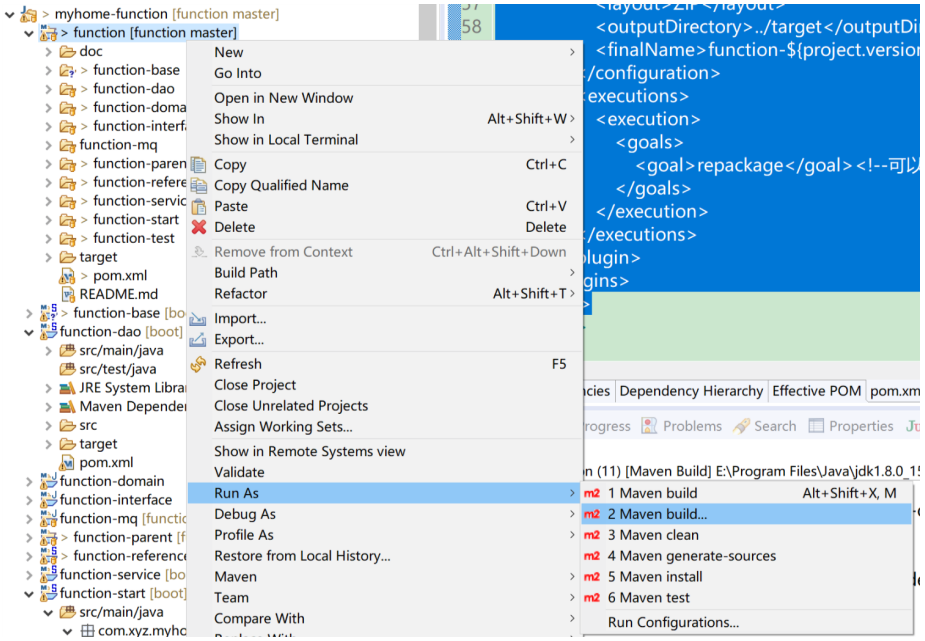
其中function-start就是这个聚合工程的启动类所在模块，我们去看看它的pom里面的build。

```
<build>
 <plugins>
  <!-- 打包 -->
  <plugin>
   <groupId>org.springframework.boot</groupId>
   <artifactId>spring-boot-maven-plugin</artifactId>
   <configuration>
    <!-- 指定该Main Class为全局的唯一入口 -->
    <mainClass>com.xyz.myhome.starter.ApplicationStarter</mainClass>
    <outputDirectory>../target</outputDirectory>
    <finalName>function-${project.version}</finalName>
   </configuration>
   <executions>
    <execution>
     <goals>
```

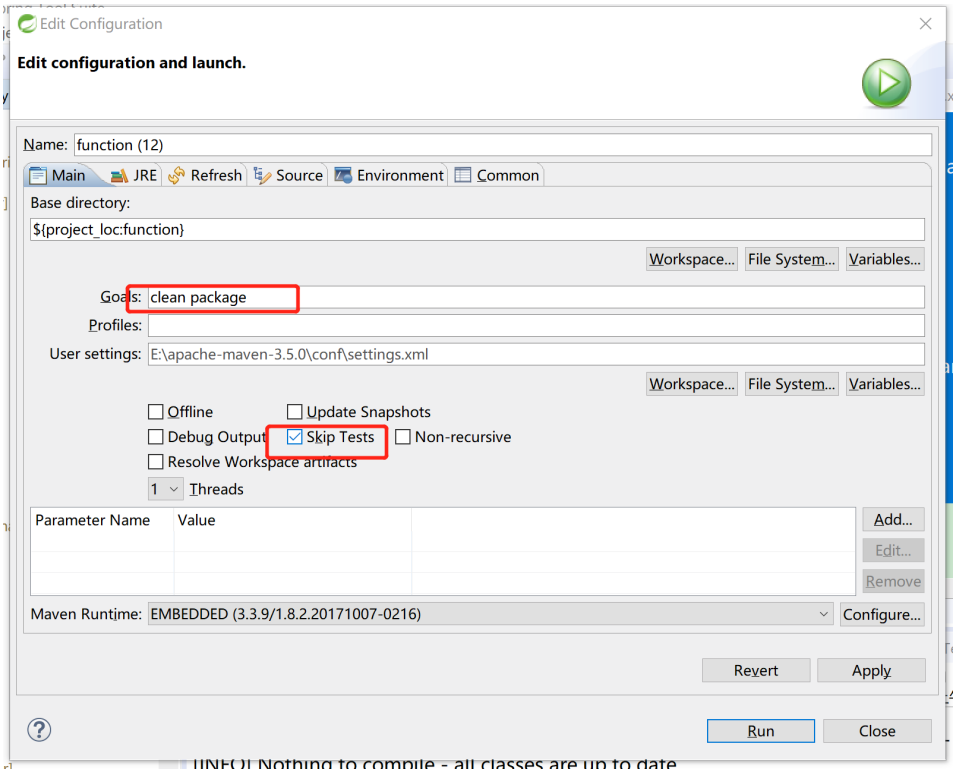
```
        <goal>repackage</goal>
        <!--可以把依赖的包都打包到生成的Jar包中 -->
      </goals>
    </execution>
  </executions>
 </plugin>
</plugins>
</build>
```

嗯，其实仔细看和我简单引入打包插件就是多了一个configuration和一个execution，前者设定了各种配置，嗯...应该都看得懂吧，依次指定了main类，打包地址（这样就会在父级工程目录下建一个target，并在里面生成一个jar包，其实就是function-start的jar，换个位置而已），jar包名称。后者就如注释一样，会将以来的jar包全部打包打进最终的jar包里。

然后就是开始打包了，对父工程执行如下操作。



然后在该处写上clean package（有时候这样父工程目录下的那个总jar包不会生成，现象很奇怪反正我遇到，只打package就解决了），还有将skip test勾上，点击run即可。



至此，针对springboot的聚合工程的简单打包就这样完成了，你会在这个目录下看到我们期待的那个jar。