

学茶商城项目打包步骤

1. 关于Spring Boot项目打包

当你通过Spring Boot的创建向导来创建项目后，在 `pom.xml` 文件中，都会有这样一段代码，它通常在文件末尾的位置：

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

这个就是Spring Boot支持Maven打包的插件，关于这个插件：

- 仅当你需要把整个项目打成一个jar包时，你才需要这个插件，如果你只是在IntelliJ IDEA或类似的开发工具中，通过执行启动类来启动整个项目，你并不需要这个插件
- 很多时候，这段代码可能正在提示错误，此时，你可以选择：
 - 如果你不需要将项目打包，你可以直接删除这段代码
 - 如果你需要将项目打包，你可以在 `<artifactId>` 标签的下方，添加 `<version>` 标签，以配置这个插件的版本，版本号应该与这个项目的Spring Boot版本相同，例如添加：
`<version>2.5.0</version>`，然后刷新即可，甚至很多时候在不报错之后，再删除 `<version>` 标签仍不会报错
- 即使没有这个插件，你也可以通过Maven对项目进行打包，但是，默认的打包结果中只会包含这个项目的代码的编译结果，并不会包含项目的各个依赖项，以至于打出来的包并不能直接运行，你还需要将所有依赖项都找出来放在一起才能运行，当有了这个插件后，会将这个项目的代码的编译结果和所有依赖项都打成一个包，这样的包是可以直接运行的，当然，这样的包会大很多，Spring官方将其称为“Fat jar”，另外，除了Fat jar以外，还会出现一个 `xxx.jar.original` 文件，这是没有添加依赖项的包，可以不使用
- 打包结果将出现在项目的 `target` 文件夹下
 - 如果你的IntelliJ IDEA不会在项目文件夹下显示 `target` 文件夹，请在左侧的项目结构面板（Project面板）的配置中勾选 `Show Excluded Files`
- 不需要被执行的项目，不要添加此插件

2. 关于Spring Boot的配置文件

Spring Boot项目的配置文件通常位于 `src/main/resources` 下，这意味着打包后，相关的配置文件也会集成到jar包内部，是无法修改的，实际上，开发时编写的配置往往并不适用于项目的生产环境，当你尝试把项目的jar包部署到服务器上时，许多配置项都是需要修改的，例如连接数据库的参数、服务的端口号等等。

Spring Boot支持在多个不同的位置自动查找配置文件，并且，在 `src/main/resources` 下的配置文件的优先级是最低的，也就是说，你只需要在Spring Boot指定的其它位置编写配置文件，就可以覆盖 `src/main/resources` 下的配置！当然，你必须事先知道各配置的属性名称，例如配置端口的属性名是 `server.port`。

比较推荐的位置是在jar包同级位置存放你的配置文件，例如：

```
[你存放jar包文件的文件夹]
-- -- project.jar
-- -- application.properties
```

这样的话，你在如上所示的 `application.properties` 中编写的配置就可以覆盖jar包里的配置（覆盖 `src/main/resources/application.properties` 中的配置）。

另外，如果你在 `src/main/resources/application.properties` 中编写了10项配置，在jar包同级的 `application.properties` 中只覆盖了3项配置，那么，在 `src/main/resources/application.properties` 中的另7个配置仍是生效的！

3. 关于pom.xml中的配置

3.1. 关于<relativePath> 标签

Spring Boot项目默认的 `pom.xml` 文件中，在 `<parent>` 标签的子级就存在 `<relativePath/>` 标签。

此标签是用于配置父级项目的 `pom.xml` 文件相对子级项目的 `pom.xml` 的位置的。

用法1：配置为空值

当配置为 `<relativePath/>` 或 `<relativePath></relativePath>`，表示将从本地库（例如 `.m2/repository`）下查找父级项目。

用法2：不配置此标签

如果不配置 `<relativePath>` 标签，默认值是 `<relativePath>../pom.xml</relativePath>`，所以，如果你的父子项目的文件夹结构本身就符合默认查找父级 `pom.xml` 的规则，并不需要使用此标签，例如：

```
[项目文件夹]
-- -- [父级项目]
-- -- -- -- pom.xml
-- -- -- -- [子级项目]
-- -- -- -- -- pom.xml
```

如上所示，在子级项目的 `pom.xml` 中，在 `<parent>` 标签的子级并不需要 `<relativePath>` 标签。

用法3：显式的指定值

如果默认的位置无法查找父级项目的 `pom.xml` 文件（父子级项目在文件夹结构上不符合），就需要通过此标签进行配置，例如：

```
[项目文件夹]
-- -- [父级项目]
-- -- -- -- pom.xml
-- -- [子级项目]
-- -- -- -- pom.xml
```














如上所示，父级项目和子级项目在文件夹结构上是同级的，此时你需要在子级项目的 `pom.xml` 中通过 `<relativePath>../父级项目/pom.xml</relativePath>` 指定父级项目的 `pom.xml` 的位置。











3.2. 关于执行Maven命令



你可以展开IntelliJ IDEA的Maven面板，找到你需要操作的项目，双击其下的 Lifecycle 下的命令，即可执行对应的Maven命令，如下图所示：

Maven

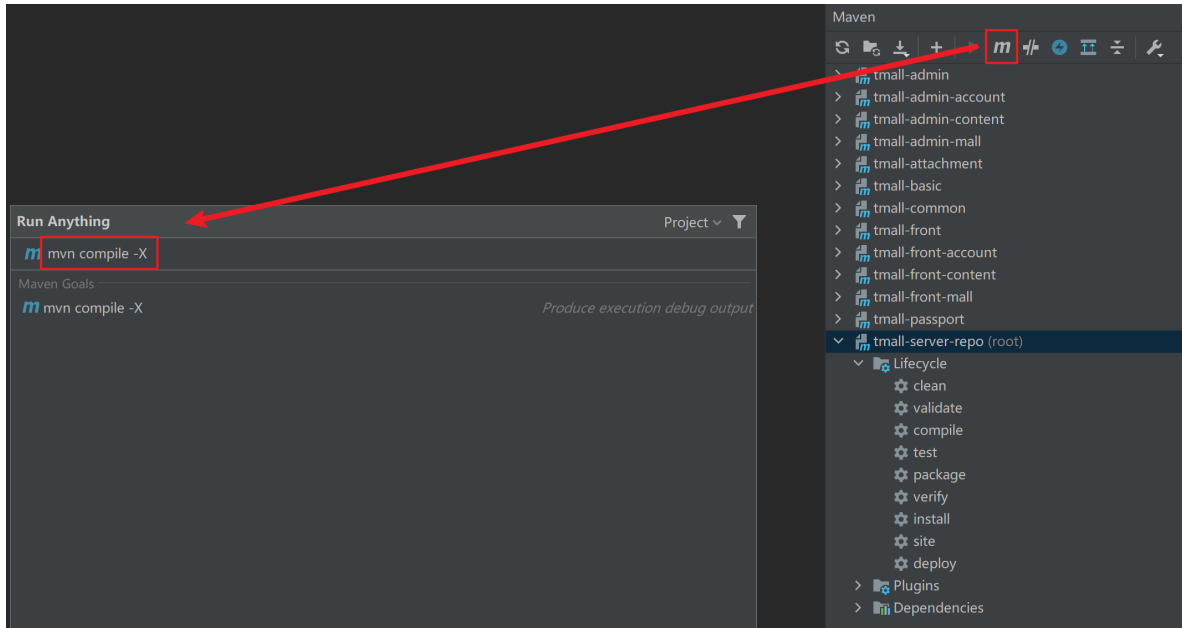


- >  tmall-admin
- >  tmall-admin-account
- >  tmall-admin-content
- >  tmall-admin-mall
- >  tmall-attachment
- >  tmall-basic
- >  tmall-common
- >  tmall-front
- >  tmall-front-account
- >  tmall-front-content
- >  tmall-front-mall
- >  tmall-passport
- ▼  tmall-server-repo (root)

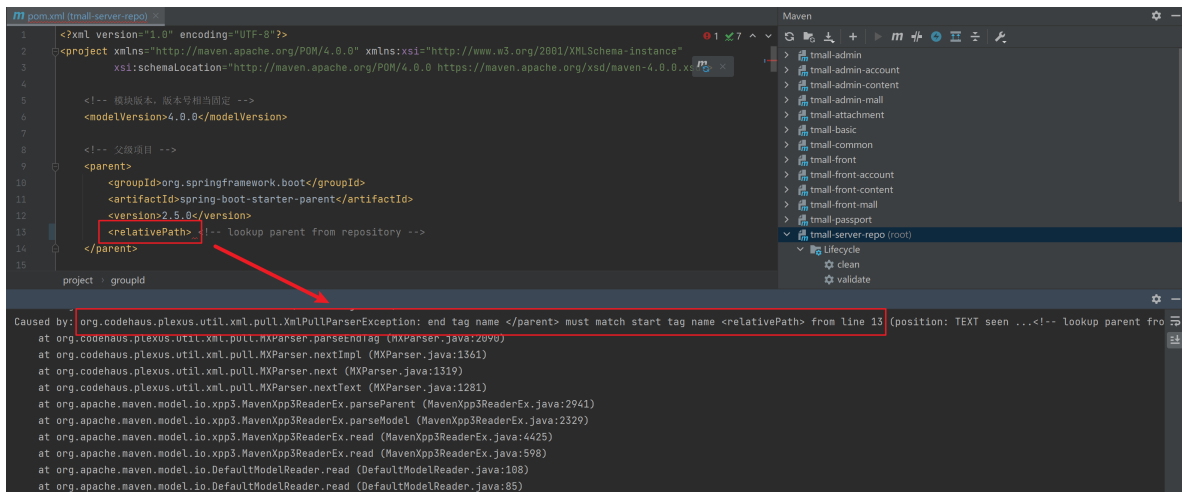
- ▼  Lifecycle
 -  clean
 -  validate
 -  compile
 -  test
 -  package
 -  verify
 -  install
 -  site
 -  deploy

- >  Plugins
- >  Dependencies

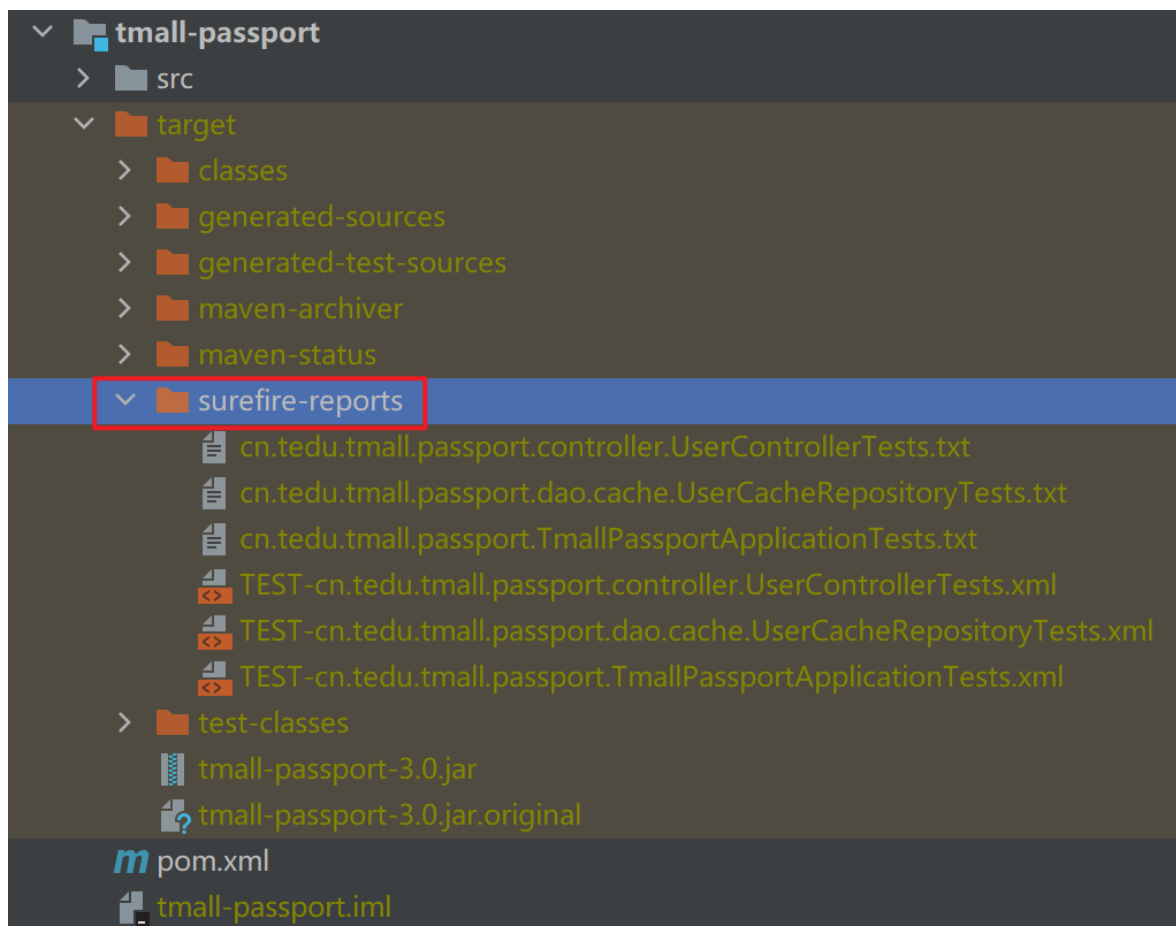
提示，如果你执行命令出错，你可以点击此面板上的 **M** 图标按钮，并执行 `mvn compile -X` 命令，就可以看到编译时的详细信息，如下图所示：



如果编译时出现异常，相关的异常信息将会显示在控制台，例如，故意将 `<relativePath/>` 改为 `<relativePath>` 使得标签没有正常关闭，在控制台提示了对应的错误信息：



注意：执行Maven的许多命令时，都会自动执行你的项目中的所有测试（即 `src/test/java` 下的所有测试类中的所有测试方法），任何测试方法执行失败，都会导致Maven命令执行失败，此时，你可能需要在 `target/surefire-reports` 文件夹下找到每个测试类对应的报告文件，以找出失败的原因。



注意：由于Spring Boot中的许多测试都是集成测试，可能在测试类上使用了`@SpringBootTest`注解，这些测试类中的测试方法被执行时，还会加载Spring环境！所以，如果你的项目中存在预热类（例如实现了`ApplicationRunner`的类），或配置了执行频率或执行间隔的计划任务，它们都会被执行，你需要了解执行的风险，例如这些组件可能对你的数据库（MySQL / Redis / Elasticsearch / 其它）进行了数据的写入等。

3.3. 关于 `mvn install`

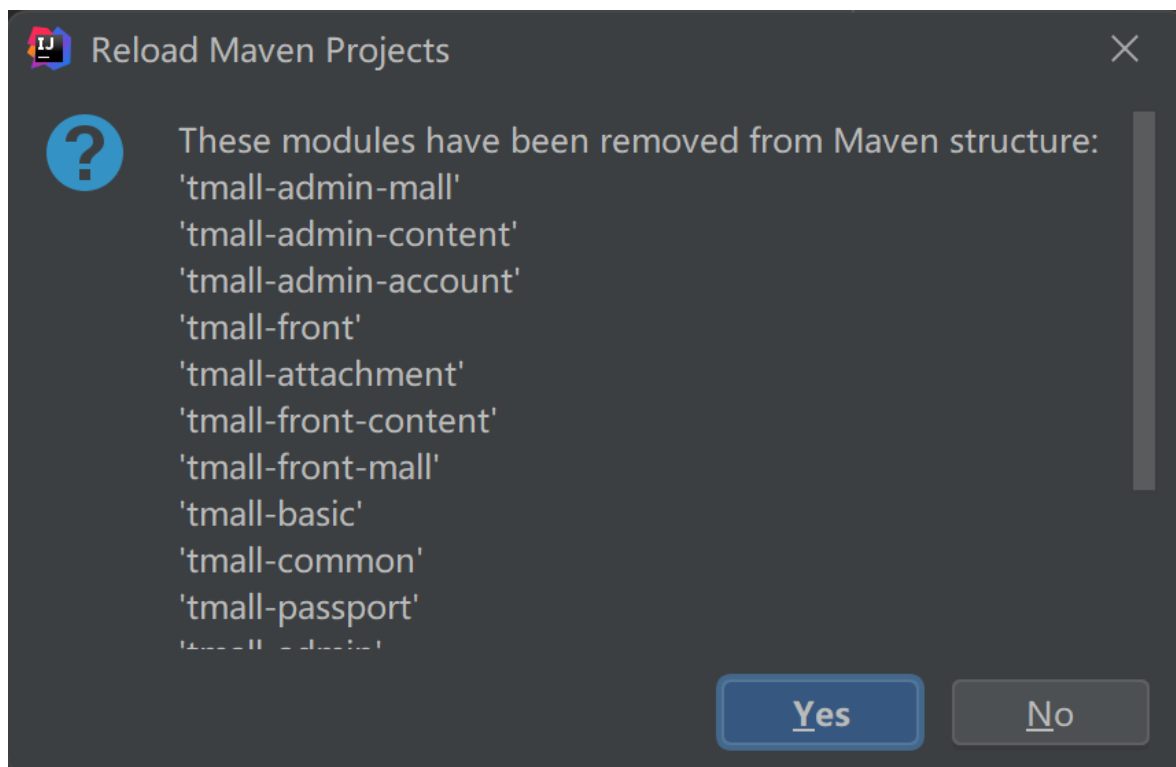
在聚合的项目中，被依赖的项目必须先“安装”到本地库，才可以对使用到它的项目进行打包！

例如 `ProjectX` 项目依赖了 `ProjectY`，你必须先对 `ProjectY` 项目执行 `mvn install`，才可以对 `ProjectX` 进行打包！

典型的被依赖项目就是被别的项目写到 `<dependency>` 中的项目，需要注意，父级项目也是各子级模块项目的依赖项，所以，对子级打包之前，应该先对父级项目执行 `mvn install`。

如果父级项目中使用了 `<modules>` 标签配置各子级模块项目，对父级项目执行 `mvn install` 可能会失败，此时，你可以将父级项目中的 `<modules>` 标签删除，再执行 `mvn install`，当成功执行后，再恢复 `<modules>` 标签的配置。

注意：当你删除 `<modules>` 时，IntelliJ IDEA会提示移出各子级模块项目（如下图所示），此时，**不要选择YES，一定要选择No！**



4. 正确的配置Maven聚合项目

以当前《学茶商城》项目为例，为了保证能够正确的打包，在配置时需要注意：

- 根级项目的 `tmall-server-repo` 的 `pom.xml` 文件，父级为 `spring-boot-starter-parent`，对应的 `<parent>` 标签的子级需要存在 `<relativePath/>`，后续，打包时，将从本地库中查找 `spring-boot-starter-parent`
- 根级项目的 `tmall-server-repo` 的 `pom.xml` 文件，不能添加 `spring-boot-maven-plugin` 插件，因为此项目是用于管理各子级模块项目的，本身不直接运行
- 对于 `tmall-common` 的 `pom.xml` 文件，不能添加 `spring-boot-maven-plugin` 插件，因为此项目是用于存放公共使用的类的，本身不直接运行
- 其它各项目（包含 `tmall-common`）的 `pom.xml` 文件中，配置父级项目的 `<parent>` 标签中，不添加 `<relativePath>` 标签，因为这些项目都应该在父级文件夹中查找父级项目的 `pom.xml` 文件，而不是从本地库或其它位置查找
- 其它各项目（不包含 `tmall-common`）的 `pom.xml` 文件中，均配置 `spring-boot-maven-plugin` 插件，因为这些项目都是需要运行的
 - `tmall-admin` 和 `tmall-front` 除外，这2个项目只起到“文件夹”的作用，或便于后续统一管理，本身没有其它作用

5. 将学茶商城项目打包

操作步骤如下（按照顺序操作）：

- 打开根级项目 `tmall-server-repo` 的 `pom.xml` 文件，删除 `<modules>` 标签，如果弹出提示框，一定要选择 `No`
- 对根级项目 `tmall-server-repo` 执行 `mvn install`
- 将根级项目 `tmall-server-repo` 的 `pom.xml` 文件中的 `<modules>` 标签还原回来
- 对 `tmall-common` 项目执行 `mvn install`
 - 它是其它各项目的依赖项，必须先安装到本地库

- 对其它各项目分别执行 `mvn package`
 - 不区分先后顺序

全部完成后，你可以在各项目的 `target` 文件夹下找到对应的Fat jar，共计9个。

你可以通过 `java -jar xxx.jar` 逐一执行这些Fat jar，例如：

```
Terminal Local +
C:\Users\Administrator\IdeaProjects\tmall-server-repo\tmall-passport\target>java -jar tmall-passport-3.0.jar

  ____  _
 / __ \| | | |
(  __/| |_| |
 \__ \|  __/
  ___/| | | |
 /___ \| |_| |
      |_|_|_|

:: Spring Boot ::
                (v2.5.0)

2023-09-19 22:06:20.449 INFO 1604 --- [           main] c.t.t.passport.TmallPassportApplication : Starting TmallPassportApplication v3.0 using Java 1.8.0_312 on PC-CHENGHENG with PID 1604 (C:\Users\Administrator\IdeaProjects\tmall-server-repo\tmall-passport\target\tmall-passport-3.0.jar started by Administrator in C:\Users\Administrator\IdeaProjects\tmall-server-repo\tmall-passport\target)
2023-09-19 22:06:20.452 INFO 1604 --- [           main] c.t.t.passport.TmallPassportApplication : The following profiles are active: prod
```

另外，启动项目的相关说明（例如需要事先安装各数据库，创建数据库，及配置参数等）请参阅本项目的 `README.md` 中的说明，以保证项目可以正常启动。