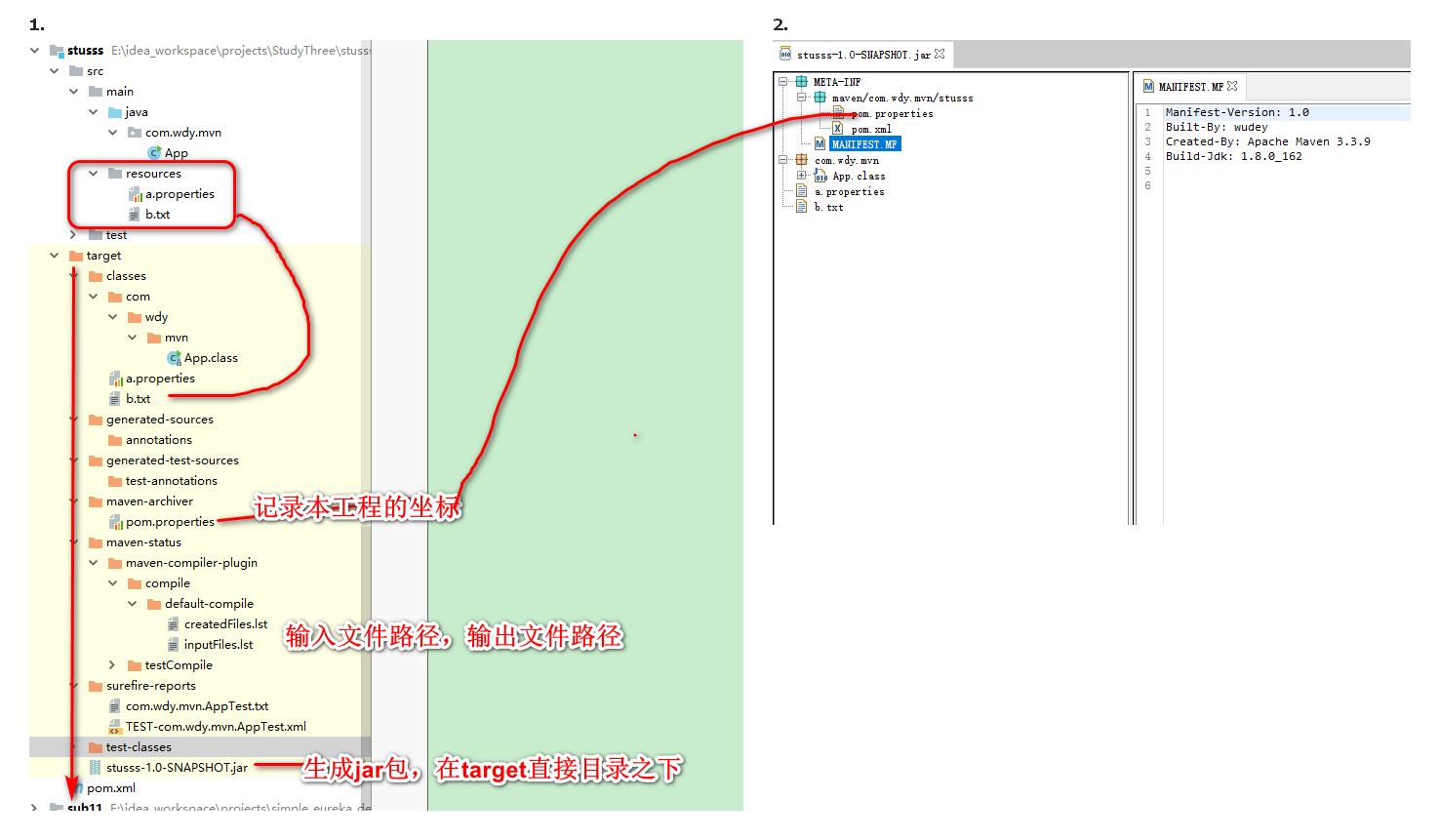
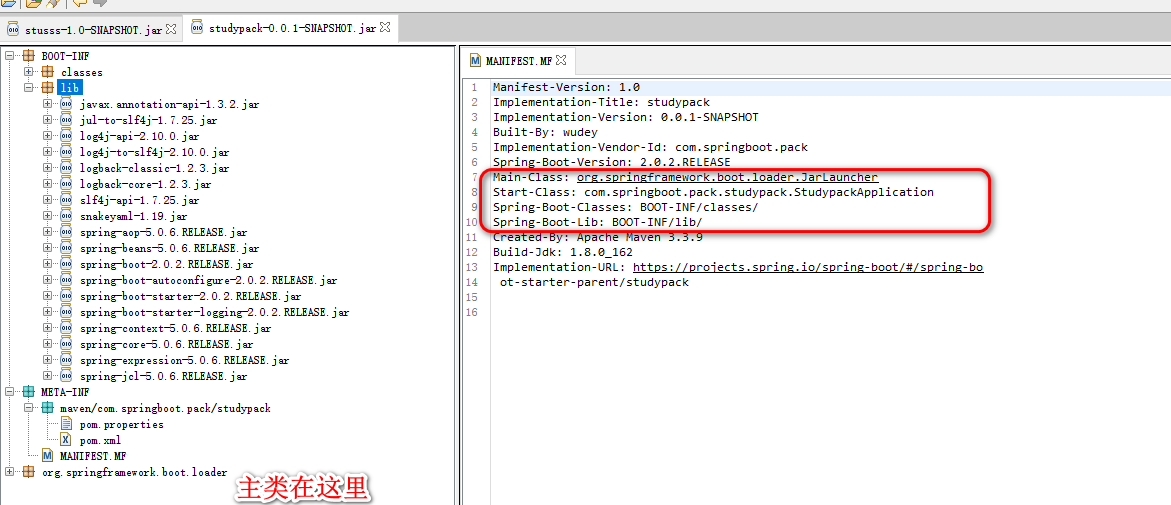
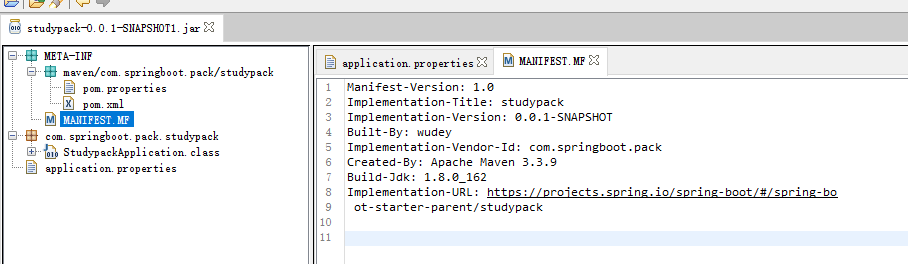
1普通的maven工程，依赖lib在打包的时候，并不会加进来，只是将pom加了进来，要正常运行必须要有maven环境



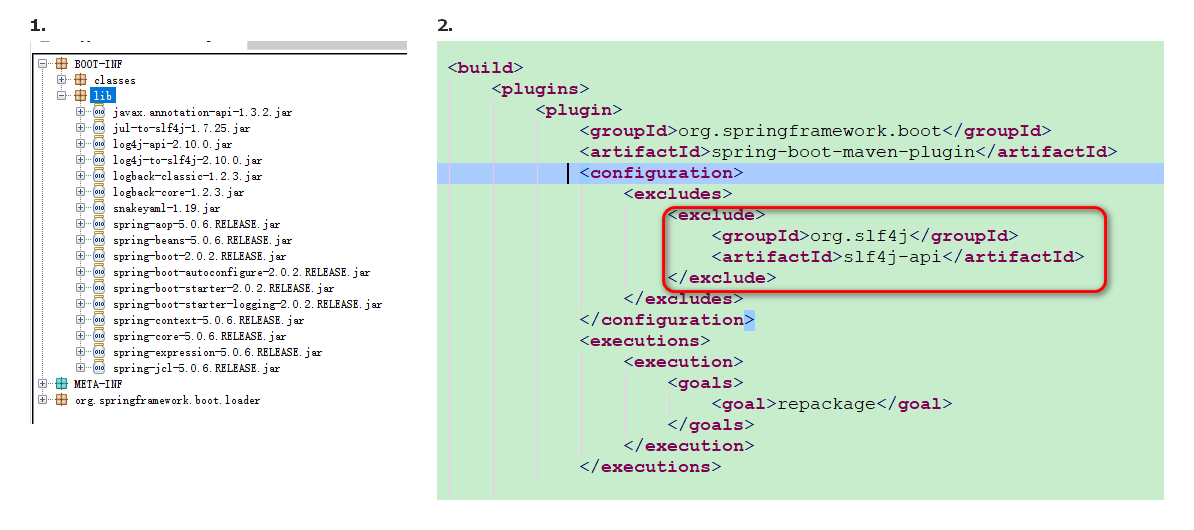
2.Spring boot maven插件打的包，除了scope为provided的依赖全部进来



它的原始包，跟普通的maven包是一样的：



3．Spring boot打包时排除某个包，除了scope为provided的依赖全部进来

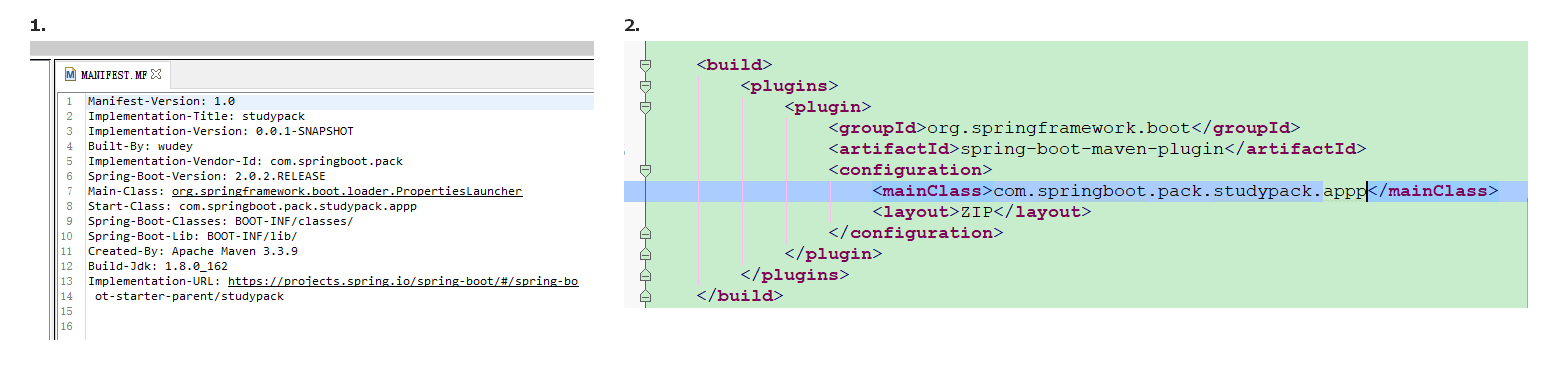


4.用artifactId排除



5.用groupID排除



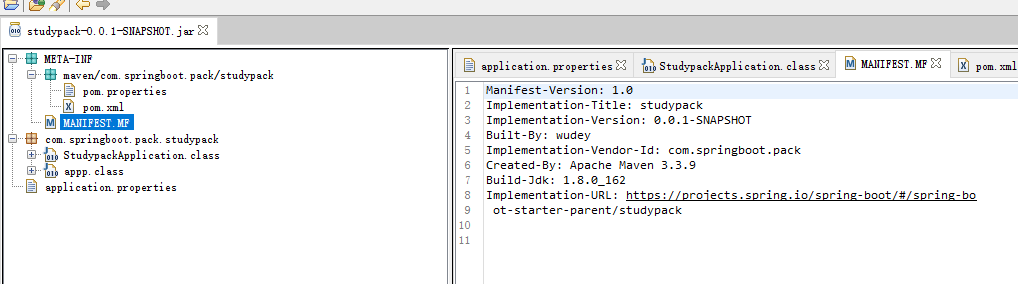
6.重写manifast文件，重写将start class重写了 

**Jar包布局：**

The layout property defaults to a guess based on the archive type (jar or war). The following layouts are available:

* JAR: regular executable JAR layout.
* WAR: executable WAR layout. provided dependencies are placed in WEB-INF/lib-provided to avoid any clash when the war is deployed in a servlet container.
* ZIP (alias to DIR): similar to the JAR layout using PropertiesLauncher.
* MODULE: Bundle dependencies (excluding those with provided scope) and project resources. Does not bundle a bootstrap loader.
* NONE: Bundle all dependencies and project resources. Does not bundle a bootstrap loader.

7.只打自己工程的包，依赖包不要



这个跟直接用maven插件差不多。

8.打包全部：

