

Maven

纯 `java` 代码开发出来的一个依赖版本工具。查看 `maven` 的目录结构。

```
mvn -v // mvn -version
```

1. Maven 的仓库分类

1. 本地仓库：放在自己本地的仓库叫本地仓库。
2. 远程仓库（私服）：放在类似于公司区域网上的仓库叫做远程仓库。但是仅仅只局限于公司内网访问。
3. 中央仓库：放在互联网上，供全球的项目共享。

中央仓库的地址

```
http://repo.maven.apache.org
```

2. 本地仓库的配置

进入 `maven` 的安装目录，打开 `maven/conf/settings.xml` 文件，添加如下代码即完成了本地仓库的配置：

```
<localRepository>/Users/liqiwen/Repository</localRepository>
```

3. Maven 项目的构建过程

3.1 项目的一键运行

进入项目的 `pom.xml` 目录中，运行：

```
//默认运行的是 tomcat 6，如需运行指定的 tomcat 版本，需要配置插件  
mvn tomcat:run
```

打开 `http://localhost:8080/xxx` 即可看到运行的项目。

3.2 项目的一键构建流程

maven 将项目构建的过程进行标准化，每个阶段使用一个命令完成。

4. Maven 常用命令

4.1 mvn compile

`compile` 是 `maven` 工程的编译命令，作用是将 `src/main/java` 目录下的 `java` 代码编译成 `class` 文件。输出的目录是 `target` 目录下。

4.2 mvn test

执行 `test` 目录下的文件。会执行 `src/main/test` 下的单元测试类。

4.3 mvn clean

`clean` 是 `maven` 工程的清理命令，执行 `clean` 会清理 `target` 目录下的内容。

4.4 mvn package

`package` 是 `maven` 工程的打包命令，对于 `java` 工程执行 `package` 打成 `jar` 包，对于 `web` 工程打包成 `war` 包。

4.5 mvn install

`install` 是 `maven` 工程的安装命令，执行 `install` 将 `maven` 打成 `jar` 包或者 `war` 包发布到本地仓库。

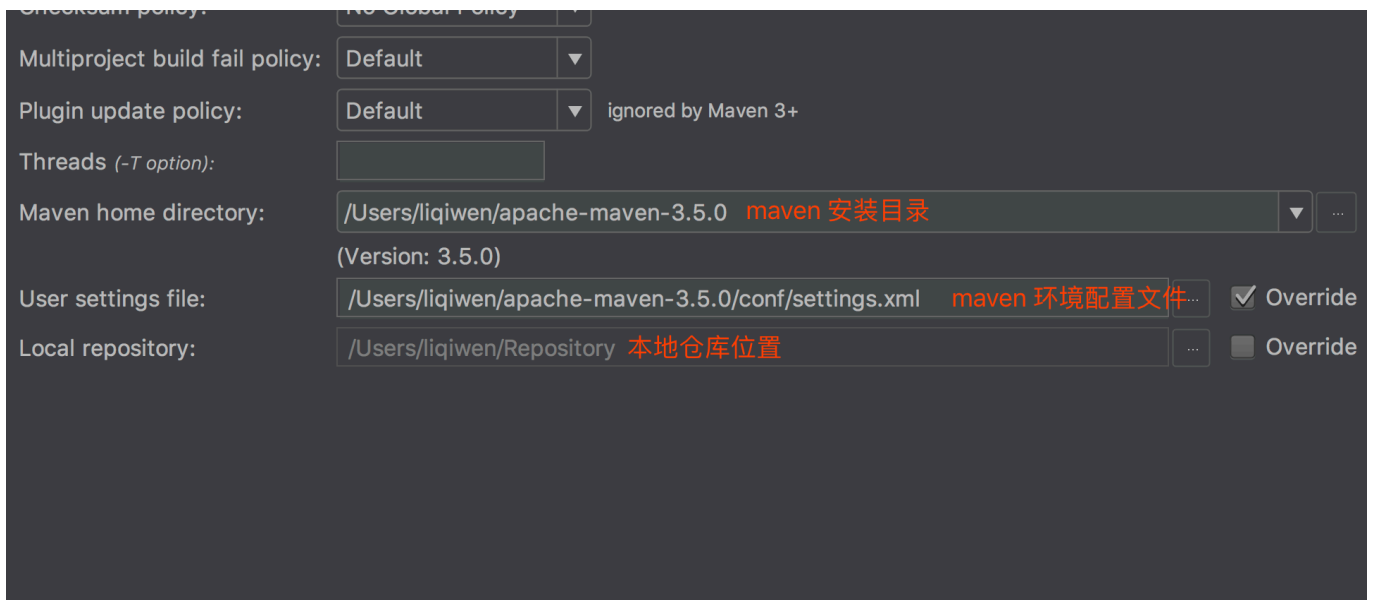
`maven` 严格按照一键构建的流程进行操作，也就是说，如果你直接执行 `maven install`，前面的命令都会全部执行一遍。

4. Maven 的生命周期

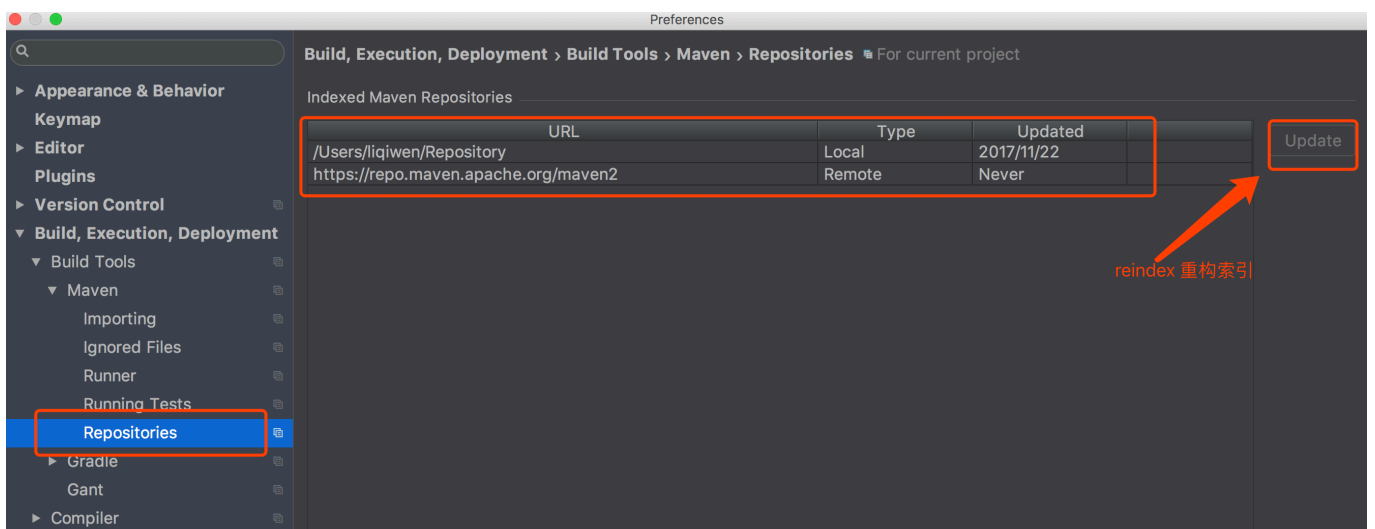
当后面的命令被执行时，前面的命令也会被执行。

5. 使用 IntelliJ 开发 maven 项目

1. 配置 maven 目录和本地仓库的位置

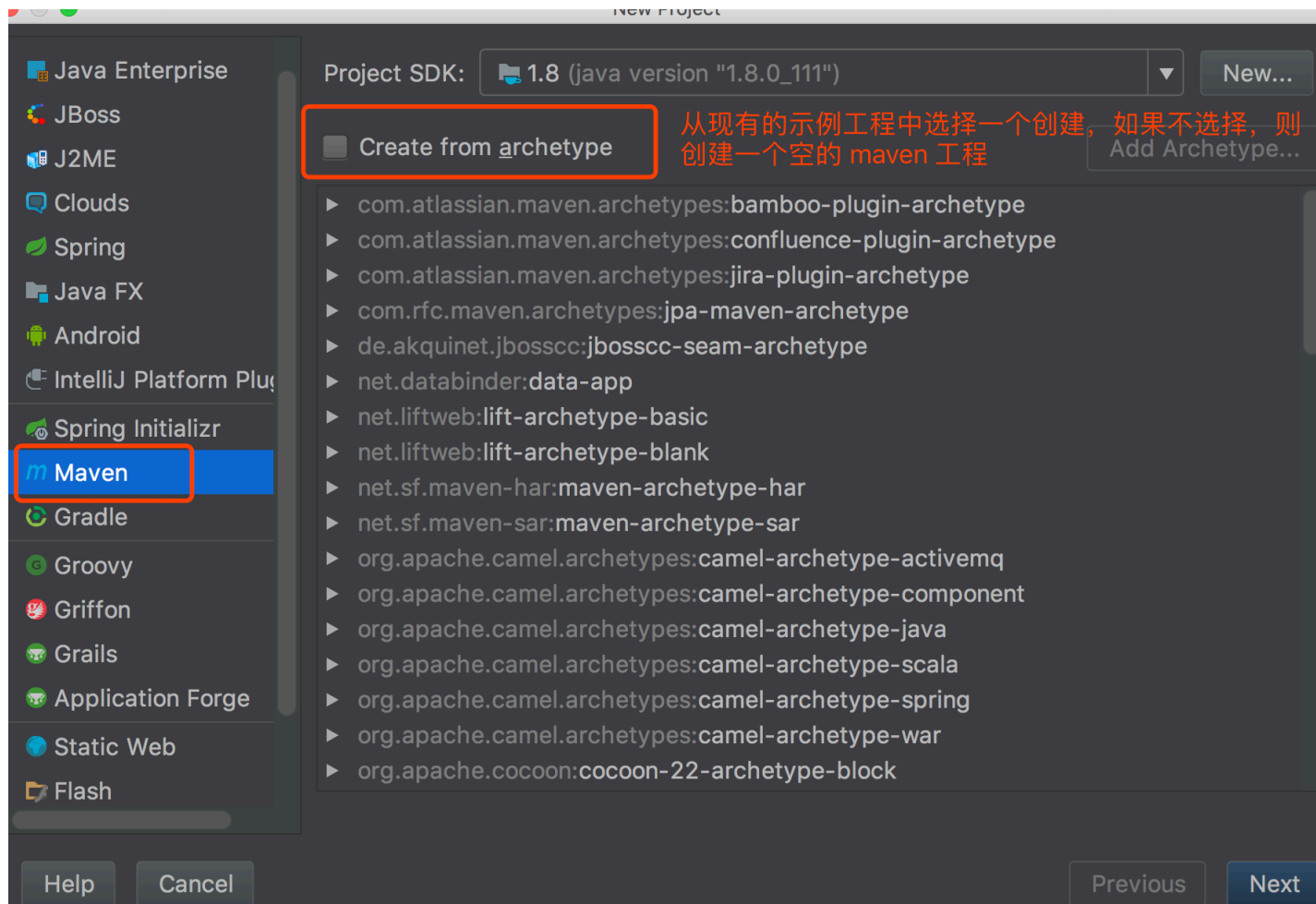


2. 构建索引



3. 在 IntelliJ 创建 maven 工程

new --> project --> maven



New Project

GroupId	<input type="text" value="com.lqwit"/>	公司, 组织名称	<input checked="" type="checkbox"/> Inherit
ArtifactId	<input type="text" value="mavenDemo"/>	项目名称	
Version	<input type="text" value="1.0-SNAPSHOT"/>	项目版本	<input checked="" type="checkbox"/> Inherit

Help Cancel Previous Next

Project name:

mavenDemo

工程名称

Project location:

~/Codes/mavenDemo

工程目录位置

...

maven 项目有三种打包方式:

jar: 一般针对 java 工程

war: 一般针对 web 工程

pom: 一般针对工程聚合

没看到配置的地方, 需要手动到 pom.xml 映射文件中更改

More Settings

Module name:

mavenDemo

Content root:

/Users/liqiwen/Codes/mavenDemo

...

Module file location:

/Users/liqiwen/Codes/mavenDemo

...

Project format:

.idea (directory based)

▼

Help

Cancel

Previous

Finish