

# Maven配置仓库

---

[Maven仓库介绍](#)

[Maven配置本地仓库](#)

[Maven配置远程仓库（公共或中央仓库）](#)

[Maven配置远程仓库（私有）](#)

[配置jdk](#)

## Maven仓库介绍

当使用 Maven 构建项目时，有三种仓库起着重要作用：

1. 本地仓库：就像你自己的书库。当你使用 Maven 下载依赖项时，它们会被保存在你电脑上的一个特定目录中，以后可以重复使用。这个本地仓库存储了项目所需的库文件和插件，使得构建过程更快速，因为不需要每次都从互联网下载。
2. 中央仓库：类似于一个全球共享的图书馆。中央仓库是 Maven 社区维护的一个公共存储库，其中包含了大量的开源库文件和插件。当你在 Maven 项目中声明依赖项时，如果没有指定其他远程仓库，Maven 会自动从中央仓库下载所需的依赖项。中央仓库是 Maven 默认的远程仓库，你可以从中央仓库获取到大量常见的开源库文件和插件。
3. 远程仓库：类似于其他人的书库。远程仓库是其他人或组织维护的存储库，可以用来存储和分享自定义的库文件和插件。当你在项目中需要特定的依赖项或插件，而这些不在中央仓库中，你可以配置远程仓库的位置，以便 Maven 可以从远程仓库中下载所需的文件。

## Maven配置本地仓库

- 1、在maven安装目录下新建文件夹 repository 作为本地仓库

bin	2023/5/8 9:04	文件夹	
boot	2023/5/8 9:04	文件夹	
conf	2023/5/8 9:04	文件夹	
lib	2023/5/8 9:04	文件夹	
repository	2023/6/15 15:15	文件夹	
LICENSE	2023/5/8 9:04	文件	19 KB
NOTICE	2023/5/8 9:04	文件	5 KB
README	2023/5/8 9:04	文本文档	3 KB

2、找到conf下的settings.xml文件进行配置

XML | 复制代码

- 1 <localRepository>.....<localRepository>中间内容替换成自己本地仓库的地址

```

-->
<settings xmlns="http://maven.apache.org/SETTINGS/1.2.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/SETTINGS/1.2.0 https://maven.apache.org/xsd/settings-1.2.0.xsd">
  <!-- localRepository
    | The path to the local repository maven will use to store artifacts.
    |
    | Default: ${user.home}/.m2/repository
    |>
    <localRepository>/path/to/local/repo</localRepository>
  -->
  <localRepository>D:\Maven\apache-maven-3.9.2-bin\apache-maven-3.9.2\repository</localRepository>
  <!-- interactiveMode
    | This will determine whether maven prompts you when it needs input. If set to false,
    | maven will use a sensible default value, perhaps based on some other setting, for
    | the parameter in question.
  -->

```

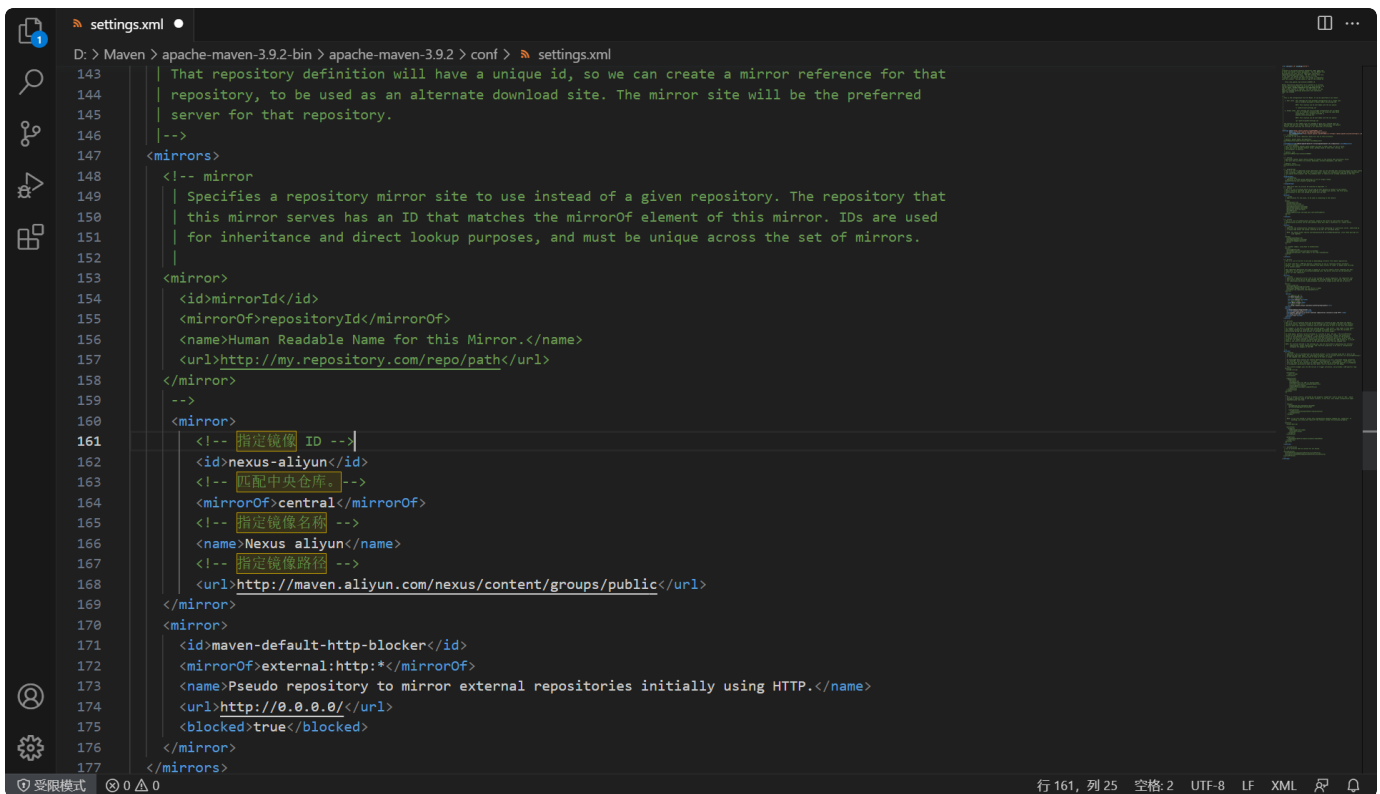
## Maven配置远程仓库（公共或中央仓库）

在 settings.xml 文件中配置仓库，下面是阿里云的中央仓库

```

1 <mirror>
2   <!-- 指定镜像 ID -->
3   <id>nexus-aliyun</id>
4   <!-- 匹配中央仓库。-->
5   <mirrorOf>central</mirrorOf>
6   <!-- 指定镜像名称 -->
7   <name>Nexus aliyun</name>
8   <!-- 指定镜像路径 -->
9   <url>http://maven.aliyun.com/nexus/content/groups/public</url>
10 </mirror>

```



## Maven配置远程仓库（私有）

确保将上面的URL替换为您实际的远程仓库URL。

在Maven的settings.xml文件中配置认证信息，在<servers>标签内添加一个<server>子标签，确保your-username和your-password替换为实际的认证信息。

```
1 <servers>
2   <server>
3     <id>remote-repo</id>
4     <url>http://example.com/repository</url>
5     <username>your-username</username>
6     <password>your-password</password>
7   </server>
8 </servers>
```

## 配置jdk

找到<profiles></profile>标签，在里面添加以下代码，配置JDK的版本，要与安装的版本对应

```
1 <profile>
2   <id>jdk-17</id>
3   <activation>
4     <activeByDefault>true</activeByDefault>
5     <jdk>17</jdk>
6   </activation>
7
8   <properties>
9     <maven.compiler.source>17</maven.compiler.source>
10    <maven.compiler.target>17</maven.compiler.target>
11    <maven.compiler.compilerVersion>17</maven.compiler.compilerVersion>
12  >
13  </properties>
14 </profile>
```