

Maven 仓库能帮助我们管理构件（主要是JAR），它就是放置所有JAR文件（WAR，ZIP，POM等等）的地方。

本地仓库

安装 Maven 后并不会创建，在第一次执行 maven 命令的时候才被创建。运行 Maven 的时候，Maven 所需要的任何构件都是直接从本地仓库获取的。如果本地仓库没有，它会首先尝试从远程仓库下载构件至本地仓库，然后再使用本地仓库的构件。Maven 本地仓库默认被创建在 %USER_HOME% 目录下。

中央仓库

Maven 中央仓库是由 Maven 社区提供的仓库，包含了大量常用的库。大多数流行的开源Java构件，以及源码、作者信息、SCM、信息、许可证信息等。

- 这个仓库由 Maven 社区管理。
- 不需要配置。
- 需要通过网络才能访问。

<http://search.maven.org/#browse>

远程仓库

如果 Maven 在中央仓库中也找不到依赖的文件，它会停止构建过程并输出错误信息到控制台。为避免这种情况，Maven 提供了远程仓库的概念，它是开发人员自己定制仓库

搜索顺序

本地仓库 --> 中央仓库（下载到本地仓库） --> 一个或多个远程仓库（若未设置，简单的停滞处理并抛出错误。找到则下载到本地仓库）

阿里云 (Aliyun) 仓库

第一步:修改 maven 根目录下的 conf 文件夹中的 setting.xml 文件,在 mirrors 节点上,添加内容如下:

```
<mirrors>
  <mirror>
    <id>alimaven</id>
    <name>aliyun maven</name>
    <url>http://maven.aliyun.com/nexus/content/groups/public/</url>
    <mirrorOf>central</mirrorOf>
  </mirror>
</mirrors>
```

```
<mirrors>
  <!-- mirror
    | Specifies a repository mirror site to use instead of a given repository. The
    | this mirror serves has an ID that matches the mirrorOf element of this mirror
    | for inheritance and direct lookup purposes, and must be unique across the
    |
  <mirror>
    <id>mirrorId</id>
    <mirrorOf>repositoryId</mirrorOf>
    <name>Human Readable Name for this Mirror.</name>
    <url>http://my.repository.com/repo/path</url>
  </mirror>
  -->
  <mirror>
    <id>alimaven</id>
    <name>aliyun maven</name>
    <url>http://maven.aliyun.com/nexus/content/groups/public/</url>
    <mirrorOf>central</mirrorOf>
  </mirror>
</mirrors>
```

第二步：pom.xml文件里添加：

```
<repositories>
  <repository>
    <id>alimaven</id>
    <name>aliyun maven</name>
    <url>http://maven.aliyun.com/nexus/content/groups/public/</url>
    <releases>
      <enabled>true</enabled>
    </releases>
    <snapshots>
      <enabled>>false</enabled>
    </snapshots>
  </repository>
</repositories>
```

```
        </snapshots>
    </repository>
</repositories>
```