

制品库
常见问题
产品文档



腾讯云

【版权声明】

©2013-2024 腾讯云版权所有

本文档著作权归腾讯云单独所有，未经腾讯云事先书面许可，任何主体不得以任何形式复制、修改、抄袭、传播全部或部分本文档内容。

【商标声明】

及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。

【服务声明】

本文档意在向客户介绍腾讯云全部或部分产品、服务的当时的整体概况，部分产品、服务的内容可能有所调整。您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或模式的承诺或保证。

常见问题

最近更新时间：2024-01-02 17:36:37

制品库支持什么类型？

支持包括 Docker、Maven、npm、Generic、Pypi、Helm 等常见制品库类型。

制品库的层级关系是怎样的？

制品库的层级关系为：**仓库 > 包 > 版本**，每个层级描述如下：

仓库：用于管理不同类型的仓库和仓库下的包资源，可以设置仓库对外的访问权限。

包：构建产物对外提供访问的基础单元，用于介绍当前构建产物的用途和使用指引。

版本：列出某个包下的所有构建产物，详细记录了每次构建产物的版本迭代更新变化。

制品库对外的权限是怎样的？

项目内：本项目成员可读，写。其他成员不可读，写。

团队内：本项目成员可读，写。企业内其他成员可读不可写。其他成员不可读，写。

公开：本项目成员可读，写。非本项目成员和匿名成员可读不可写。

制品库包名称的规则是什么？

包名仅支持1 - 31位英文、数字、下划线（_）、中划线（-）、点（.）的组合。不可与本仓库其他包名称重复，可以其他仓库内包名重复；

包的设置项包含什么？

设置项包含：许可证、包描述、成熟度、Web 站点 URL、问题跟踪 URL、版本控制 URL 等。

Maven 相关问题

Maven 的 settings.xml 配置文件在哪？

在生成 Maven 类型制品时，您需要配置您的 `settings` 文件，通常这个文件存放的位置有如下几个地方，您都可以按需使用，只不过配置生效的范围和优先级不同：

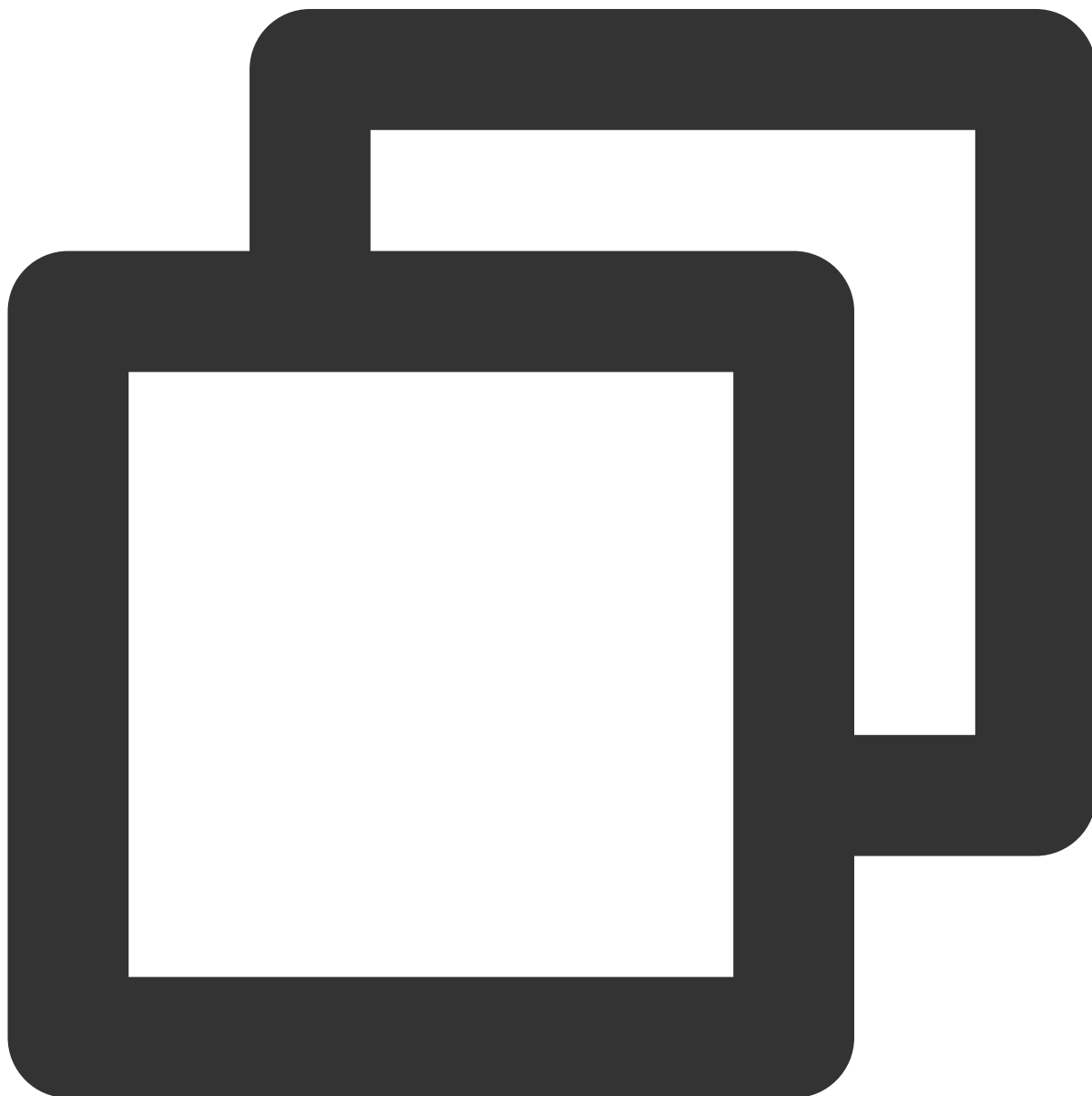
1. 全局配置：`${M2_HOME}/conf/settings.xml`

如果您不记得 Maven 的安装目录 `${M2_HOME}`，您可以在终端中执行 `echo ${M2_HOME}` 或者 `mvn -version` 就可以看到 Maven home 的路径。

2. 用户配置：`${user.home}/.m2/settings.xml`

您可以通过 `echo` 环境变量的方式找到该文件目录，有时候这个目录下是没有 `settings.xml` 文件，您可以去全部配置里拷贝一份 `settings.xml` 再进行修改。

3. 指定路径下的 `settings.xml`



```
mvn deploy --settings settings.xml
```

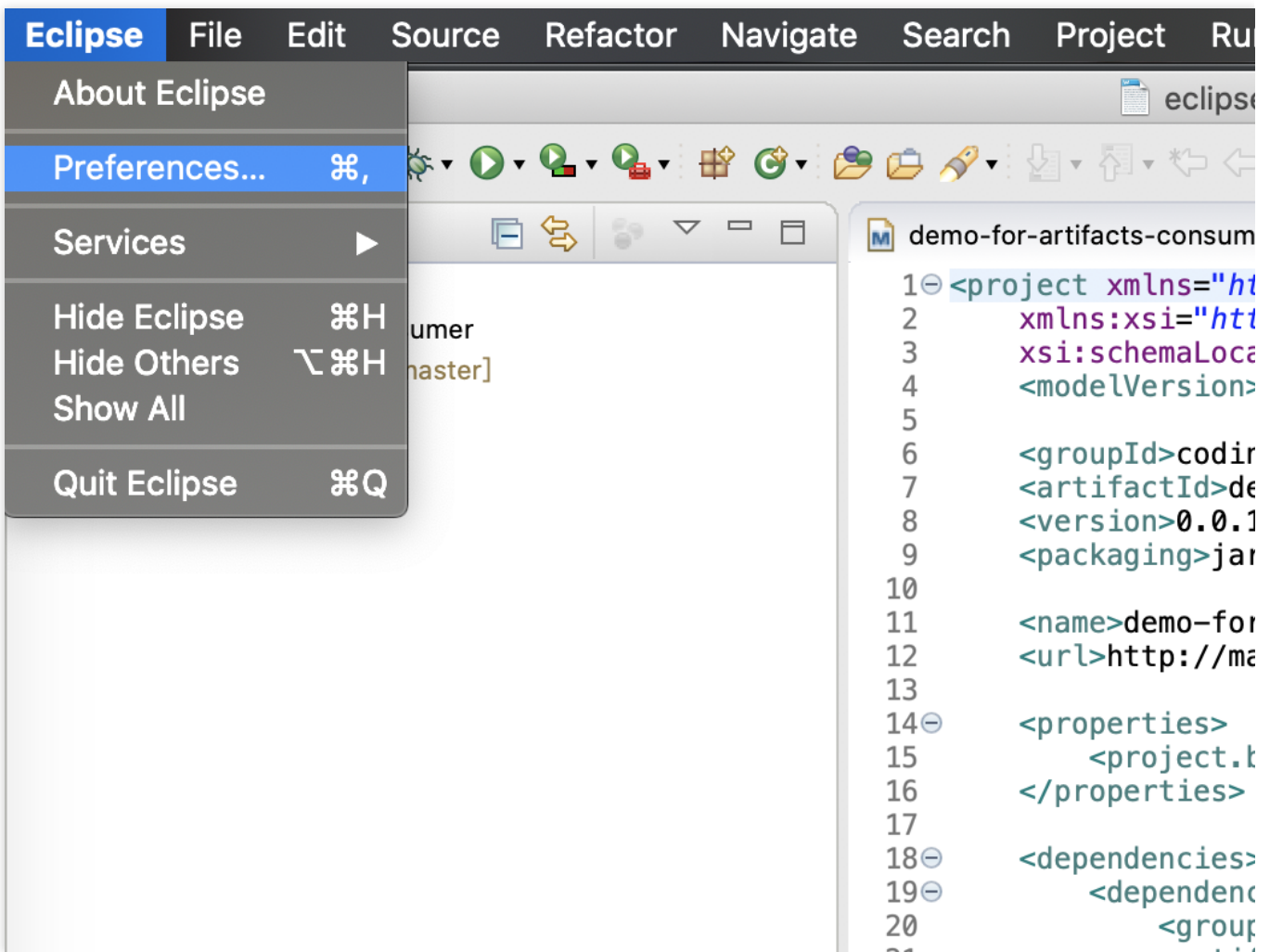
说明：

在终端执行 mvn 相关命令时， settings.xml 配置生效的优先级：指定路径 > 用户配置 > 全局配置。

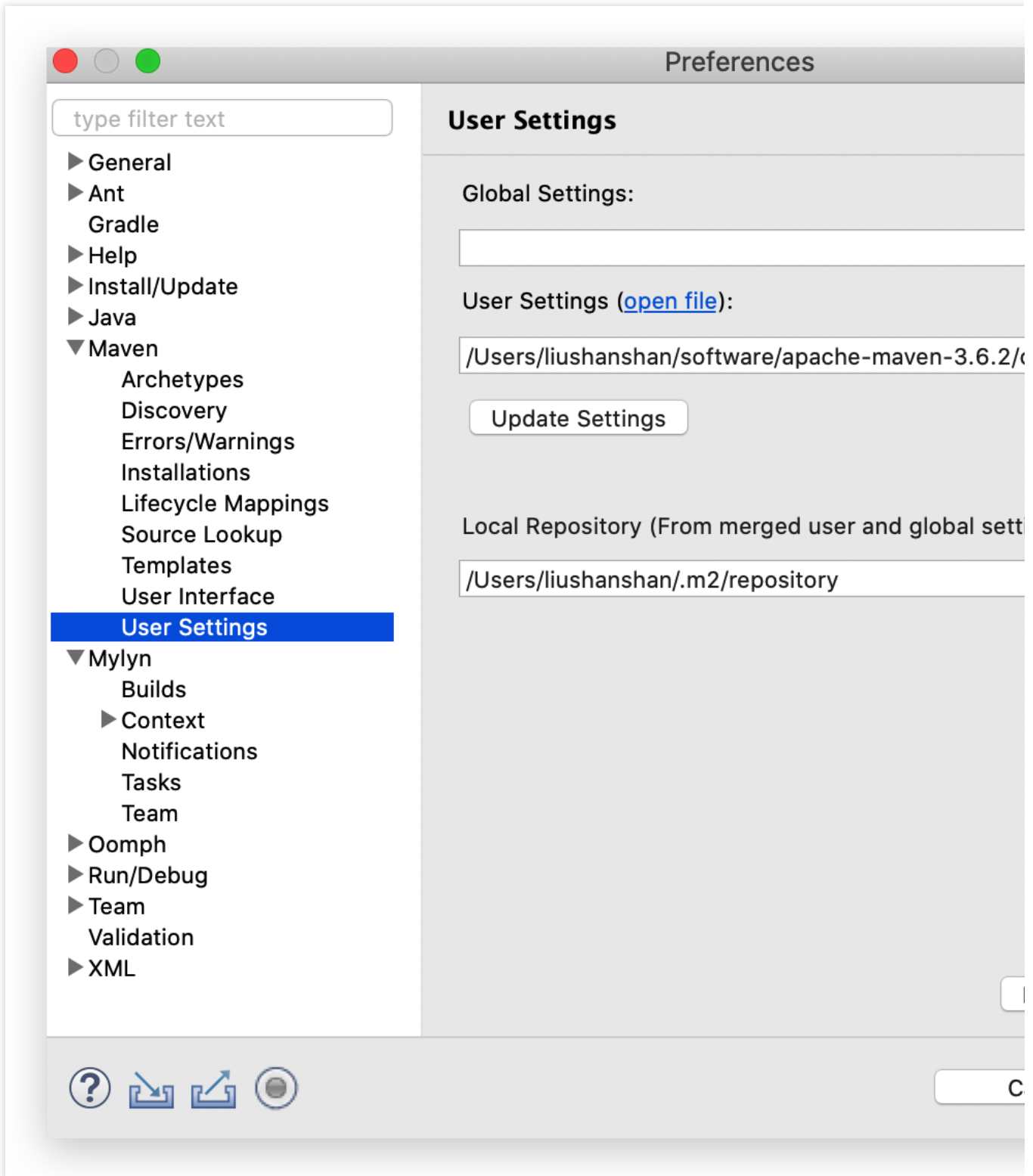
除了在终端当中执行 mvn 命令，有时候您在 Eclipse 等 IDE 中也会用 Maven，该怎么修改 settings.xml 文件的配置？

以 Eclipse 为例（其它类型 IDE 网上也有丰富的资料供参考）：

1. 单击 **Preferences** 进入偏好设置。



2. 在 **Maven > User Settings** 当中您就可以看到您使用的配置文件路径，并且修改配置文件。



npm 相关问题

如何将 npm @scope 指向 CODING 私有制品库？

1. 可以通过配置 `.npmrc` 来指定 `@scope` 的 registry。

例如: 有一个 npm 包, 位置信息如下:

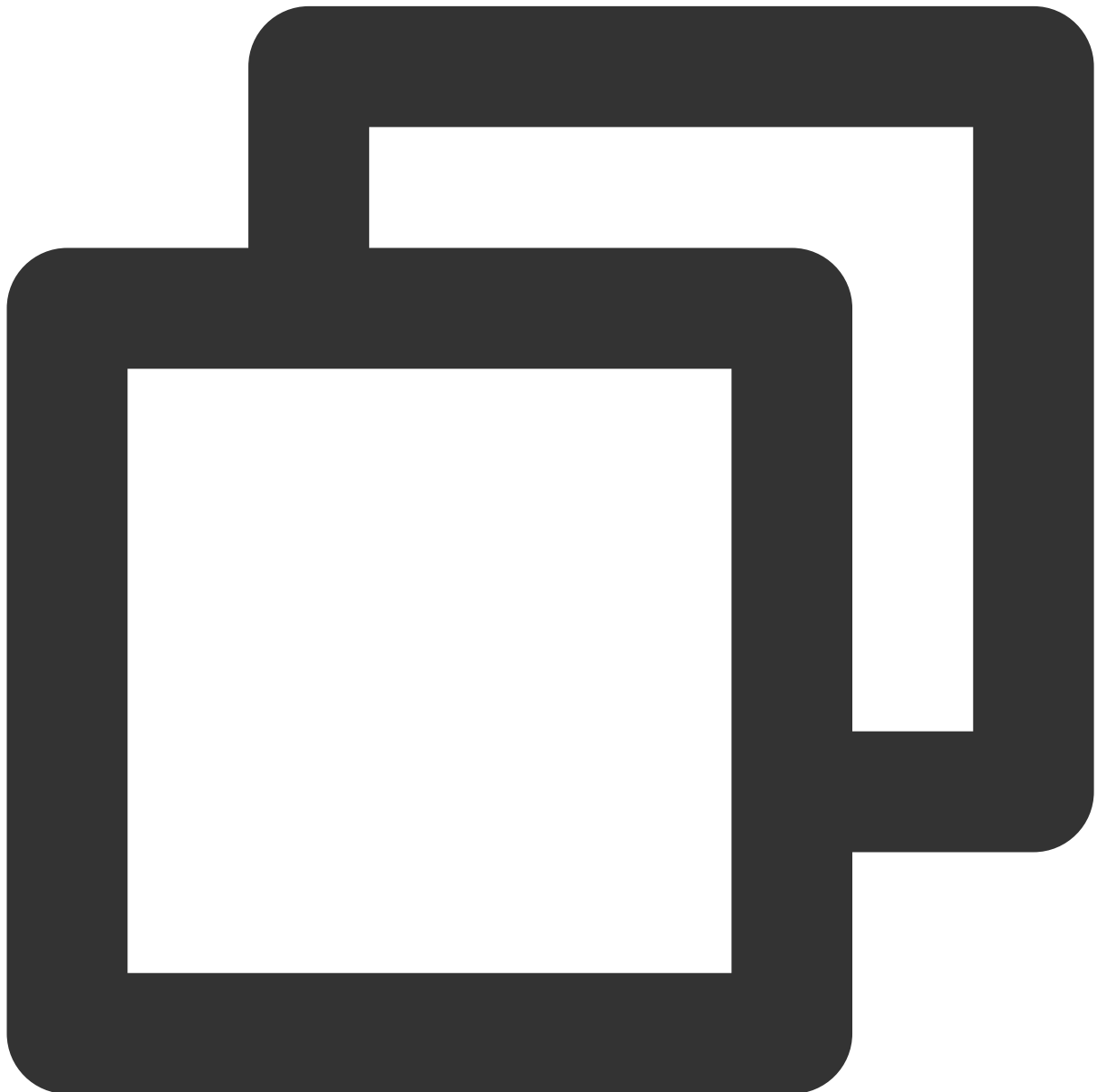
企业: my-team

项目: my-project

制品仓库: my-npm-repo

名: `@my-scope/my-pkg`。

可以通过配置 `.npmrc`, 让 `package.json` 中的 `@my-scope/my-pkg` 指向该链接地址:



```
https://my-team-npm.pkg.coding.net/my-project/my-npm-repo/
```

2. 直接将 npm 包的 registry 指向 CODING 制品库。

直接单击 npm 制品库指引页面中的[使用访问令牌生成配置](#)生成 .npmrc。

Operation Guide

npm

Configure Credentials

Push

Pull

Mirror Acceleration

Configure Access Token

Enter Password an access token is generated and inserted

Enter the password.

Set Credentials with Configuration File

Add the following configuration to the .npmrc file in the s package.json. For management functions, go to [Personal](#)

```
; Please keep your configuration safe
registry=https://StrayBirds-npm.pkg.coding.net
always-auth=true
//StrayBirds-npm.pkg.coding.net/coding-demo/n
//StrayBirds-npm.pkg.coding.net/coding-demo/n
//StrayBirds-npm.pkg.coding.net/coding-demo/n
```

Replace Text:

- PASSWORD: your login password
- If you choose to manually replace the login password encode the <PASSWORD>:

- Execute the following command:

```
node -e "require('readline').createInte"
```

请妥善保管生成的配置：



```
registry=https://my-team-npm.pkg.coding.net/my-project/my-npm-repo/  
always-auth=true  
//my-team-npm.pkg.coding.net/my-project/my-npm-repo/:username=xxxxxx  
//my-team-npm.pkg.coding.net/my-project/my-npm-repo/:_password=xxxxx  
//my-team-npm.pkg.coding.net/my-project/my-npm-repo/:email=xxxxx
```

由于 CODING 的 npm 制品库支持代理功能，可以直接将 npm registry 设置为 CODING 制品库，公共制品也可以被拉取到。

说明：

关于如何在 CODING 持续集成中使用 npm 制品库，可参见 [持续集成 > 构建 npm 类型制品](#)。

权限相关问题

如何拉取其它 CODING 项目制品库的制品？

您可以通过项目令牌的方式拉取其它 CODING 项目制品库的制品。

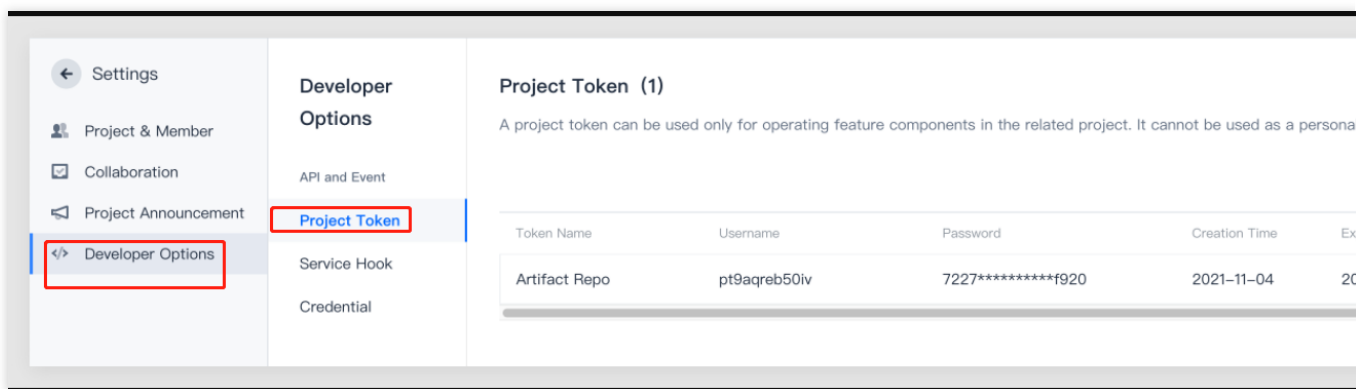
为了方便您区分即将要操作的两个不同项目，我们统一将：

需要被拉取制品的制品库所在项目称为“项目 A”

执行拉取的项目称为“项目 B”

步骤一：在项目 A 内创建项目令牌

1. 进入项目 A 项目设置 > 开发者选项 > 项目令牌，单击新建项目令牌。



2. 配置制品库权限。

Settings

- Project & Member
- Collaboration
- Project Announcement
- Developer Options

Publish API Documentation Create, query, edit, delete Read and manipulate project me... Read and operate the project per...

Code Repository Permission

Unified configuration all code warehouse permissions Appointed warehouse code configuration access

Repository Name	Access Permission	Operation Permission
* All Code Repositories in the Project		
	<input checked="" type="checkbox"/> Read Read Code Repository	<input type="checkbox"/> Read/Write Push to Code Repository

Artifact Repository Permission

United configure all products warehouse permissions Specified products warehouse configuration access

Artifact Repository Name	Access Permission	Operation Permission
* all products in the project library		
	<input type="checkbox"/> Read Pull Artifact Repository	<input type="checkbox"/> Read/Write Pull or Push Artifact Repository

Continuous Integration Permission

Build Node API Trigger Build Job

Allow the build nodes to access. Use APIs to trigger the build of continuous integration. Build job management / Trigger build (For Open API only)

Create
Cancel

3. 单击**确定**后创建成功。

Settings

- Project & Member
- Collaboration
- Project Announcement
- Developer Options

Developer Options

- API and Event
- Project Token
- Service Hook
- Credential

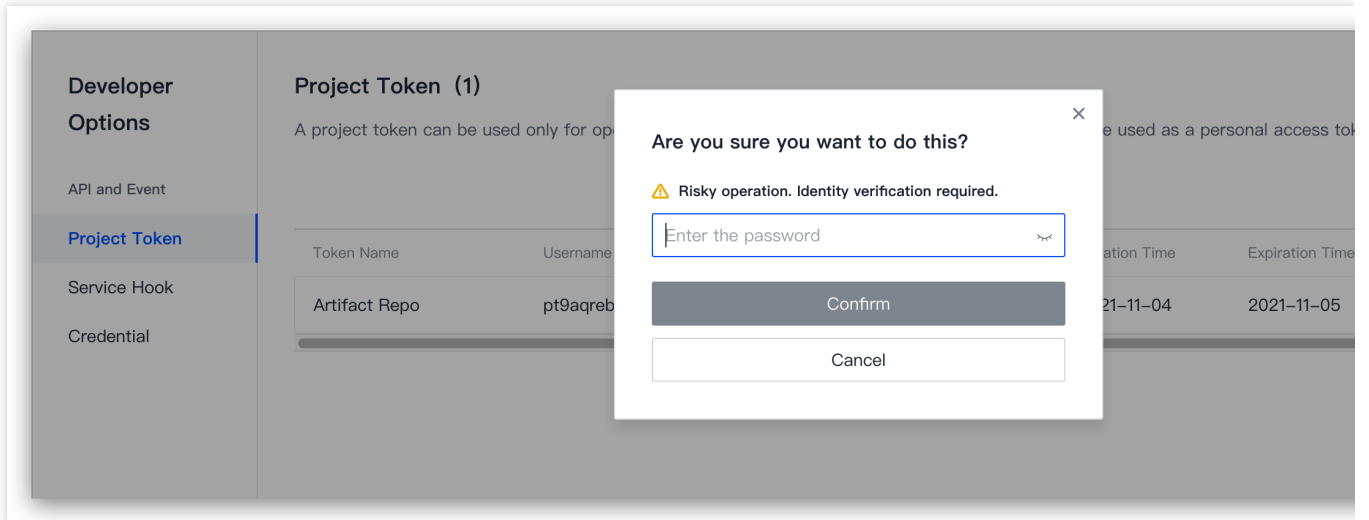
Project Token (1)

A project token can be used only for operating feature components in the related project. It cannot be used as a personal e

Token Name	Username	Password	Creation Time	Expire
Artifact Repo	pt9aqreb50iv	7227*****f920	2021-11-04	202

步骤二：在项目 B 中将刚才创建的项目 A 项目令牌作为用户名 + 密码拉取制品

1. 根据您的制品类型，配置认证信息。
2. 回到刚才创建的项目 A 项目令牌页面，单击“查看密码”。



3. 在项目 B 制品库配置认证信息时，将项目令牌用户名 + 项目令牌密码（token）作为用户名 + 密码填入信息。
4. 正确填入信息后，即可拉取成功其他 CODING 项目的制品库。

为什么制品仓库会出现没有主动推送的依赖包？

这是由于该制品仓库开启了制品代理功能，当制品库代理功能开启时，制品仓库会作为统一入口帮助您管理依赖的第三方制品，因此这些依赖包会出现在制品仓库。您可以在 CODING 制品库内追踪该依赖包的团队内成员的使用情况，也可以通过 CODING 制品扫描统一检测该依赖包的安全漏洞，方便您的团队依赖制品审计。

镜像相关问题

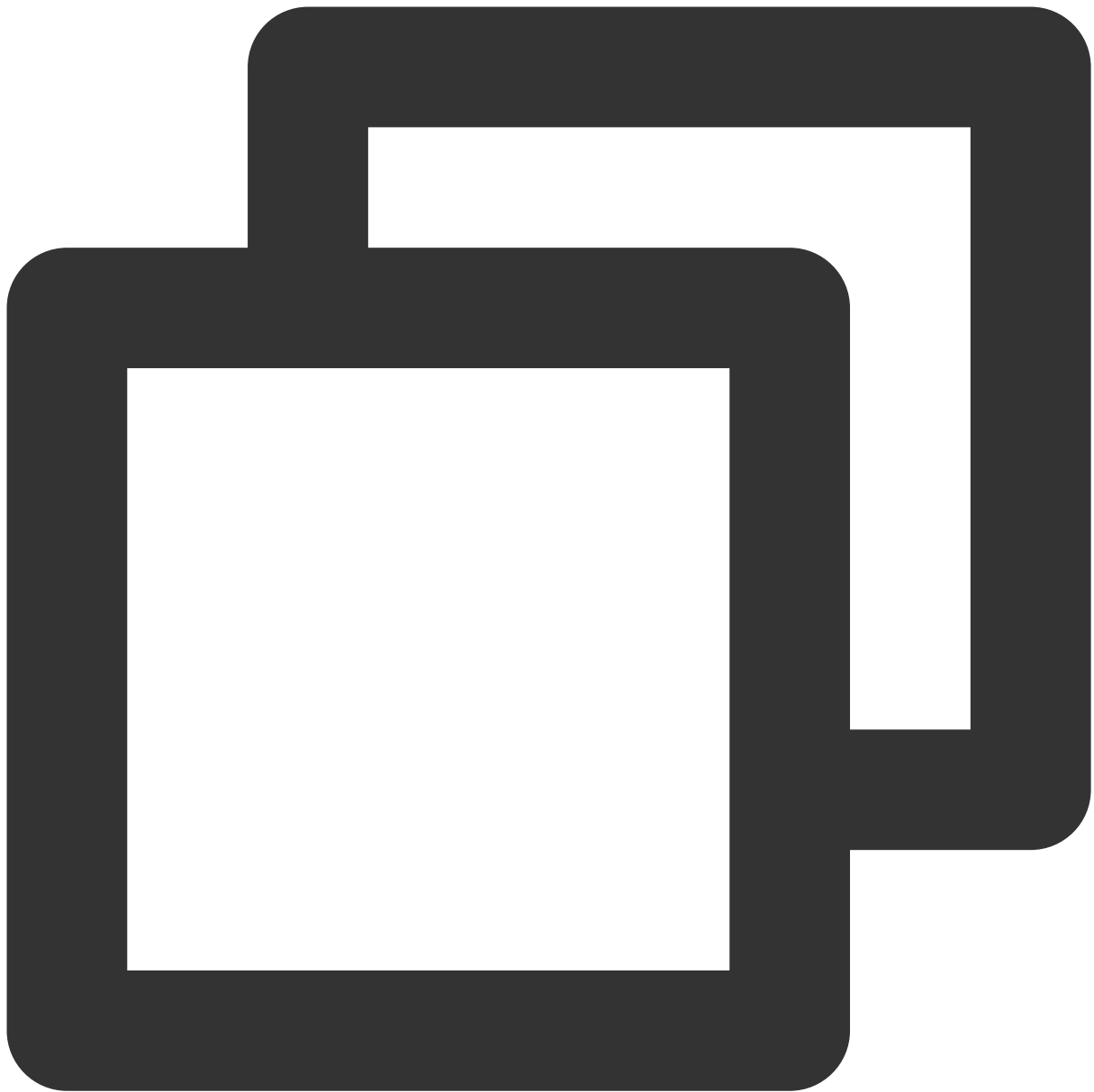
为什么无法从制品库拉取依赖包？

使用 mirrors 参数配置镜像源加速后，无法从制品库拉取依赖包进行构建，问题截图如下：

```
[INFO] Finished at: 2021-05-19T10:44:33+08:00
[INFO] -----
[ERROR] Failed to execute goal on project maven-mirror-test: Could not resolve dependencies for project org.example:maven-mirror-test:jar:2.0.0:
[ERROR]    not find artifact org.example:maven-child:jar:2.0.0 in nexus-tencentyun (http://mirrors.cloud.tencent.com/nexus)
[ERROR]
[ERROR] To see the full stack trace of the errors, re-run Maven with the -e switch.
[ERROR] Re-run Maven using the -X switch to enable full debug logging.
[ERROR]
[ERROR] For more information about the errors and possible solutions, please read the following articles:
[ERROR] [Help 1] http://cwiki.apache.org/confluence/display/MAVEN/DependencyResolutionException
```

导致此问题的原因可能是由于 `<mirror>` 配置中的 `<mirrorOf>*` 将所有流量切换到了镜像源中拉取，而镜像源中并没有保存在 CODING 制品仓库中的依赖包。此问题有两个解决方案。

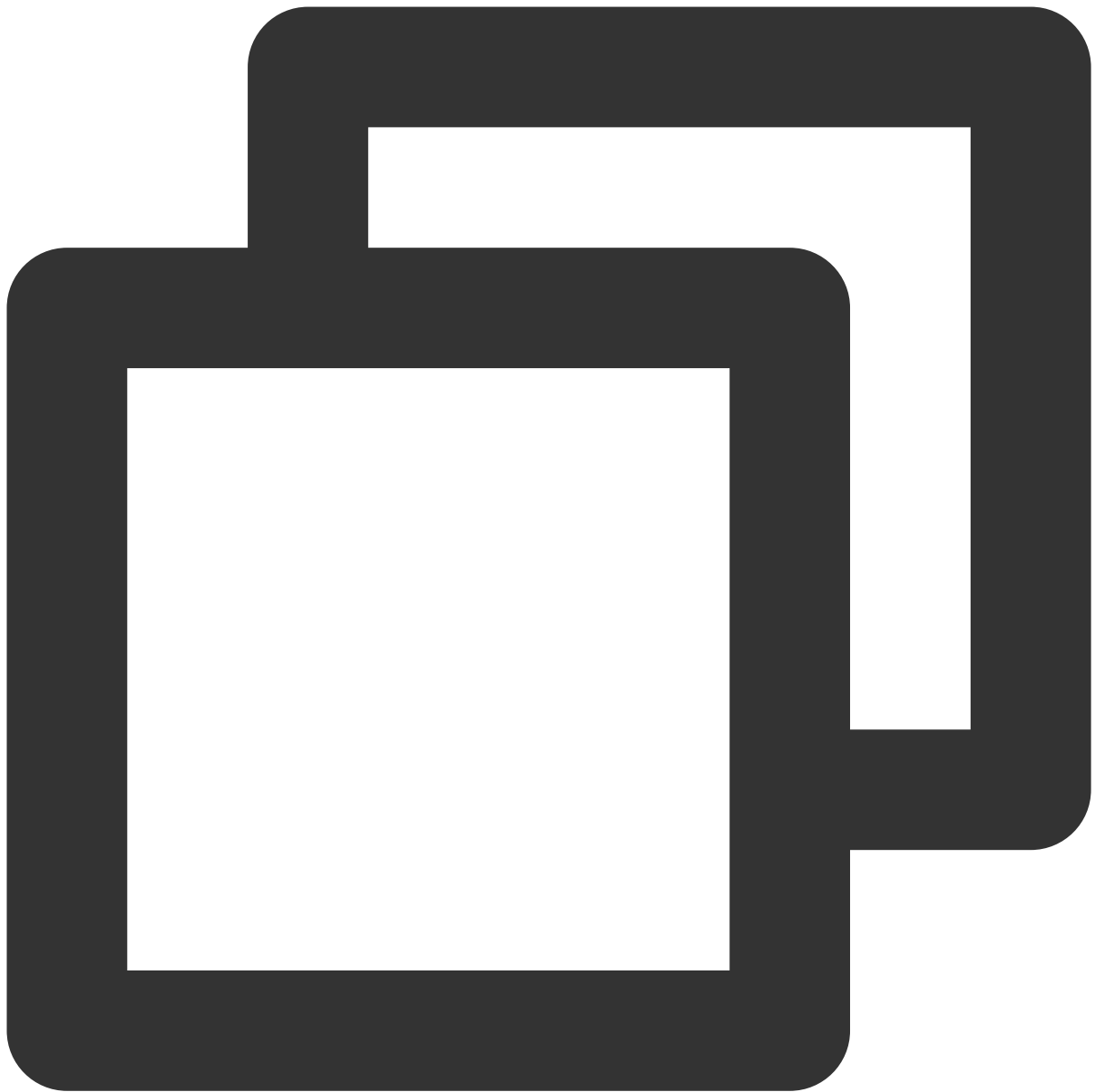
方法一：修改参数配置，仅允许非 CODING 制品仓库来源的制品从镜像源拉取。



```
<settings>
  <!-- profiles 配置根据 CODING 仓库中的拉取指引配置 -->
  <profiles>
    <profile>
      <id>Repository Proxy</id>
      <activation>
        <activeByDefault>>true</activeByDefault>
      </activation>
      <repositories>
        <repository>
          <id>coding-maven-repo-id</id>
```

```
<name>coding-maven-repo-name</name>
<url>https://coding-maven-repo-url</url>
<releases>
  <enabled>true</enabled>
</releases>
<snapshots>
  <enabled>true</enabled>
</snapshots>
</repository>
</repositories>
</profile>
</profiles>
<mirrors>
  <mirror>
    <id>nexus-tencentyun</id>
    <!-- 非 coding-maven-repo-id 来源的制品才会从镜像源中拉取制品 -->
    <mirrorOf>!coding-maven-repo-id</mirrorOf>
    <name>Nexus tencentyun</name>
    <url>http://mirrors.cloud.tencent.com/nexus/repository/maven-public/</u
  </mirror>
</mirrors>
</settings>
```

方法二：删除 `<mirrors>` 镜像源配置，打开 **CODING** 制品仓库代理并指向镜像源。此方案会将所有开源依赖包代理并保存至 **CODING** 制品仓库。



```
<settings>
  <!-- profiles 配置根据 CODING 仓库中的拉取指引配置 -->
  <profiles>
    <profile>
      <id>Repository Proxy</id>
      <activation>
        <activeByDefault>>true</activeByDefault>
      </activation>
      <repositories>
        <repository>
          <id>coding-maven-repo-id</id>
```

```
<name>coding-maven-repo-name</name>
<url>https://coding-maven-repo-url</url>
<releases>
  <enabled>true</enabled>
</releases>
<snapshots>
  <enabled>true</enabled>
</snapshots>
</repository>
</repositories>
</profile>
</profiles>
</settings>
```

注意：

持续集成中的镜像源加速配置写在了构建环境全局配置 `${M2_HOME}/conf/settings.xml` 中，需要使用项目下 `settings.xml` 配置覆盖；



```
mvn install -s "./settings.xml"
```