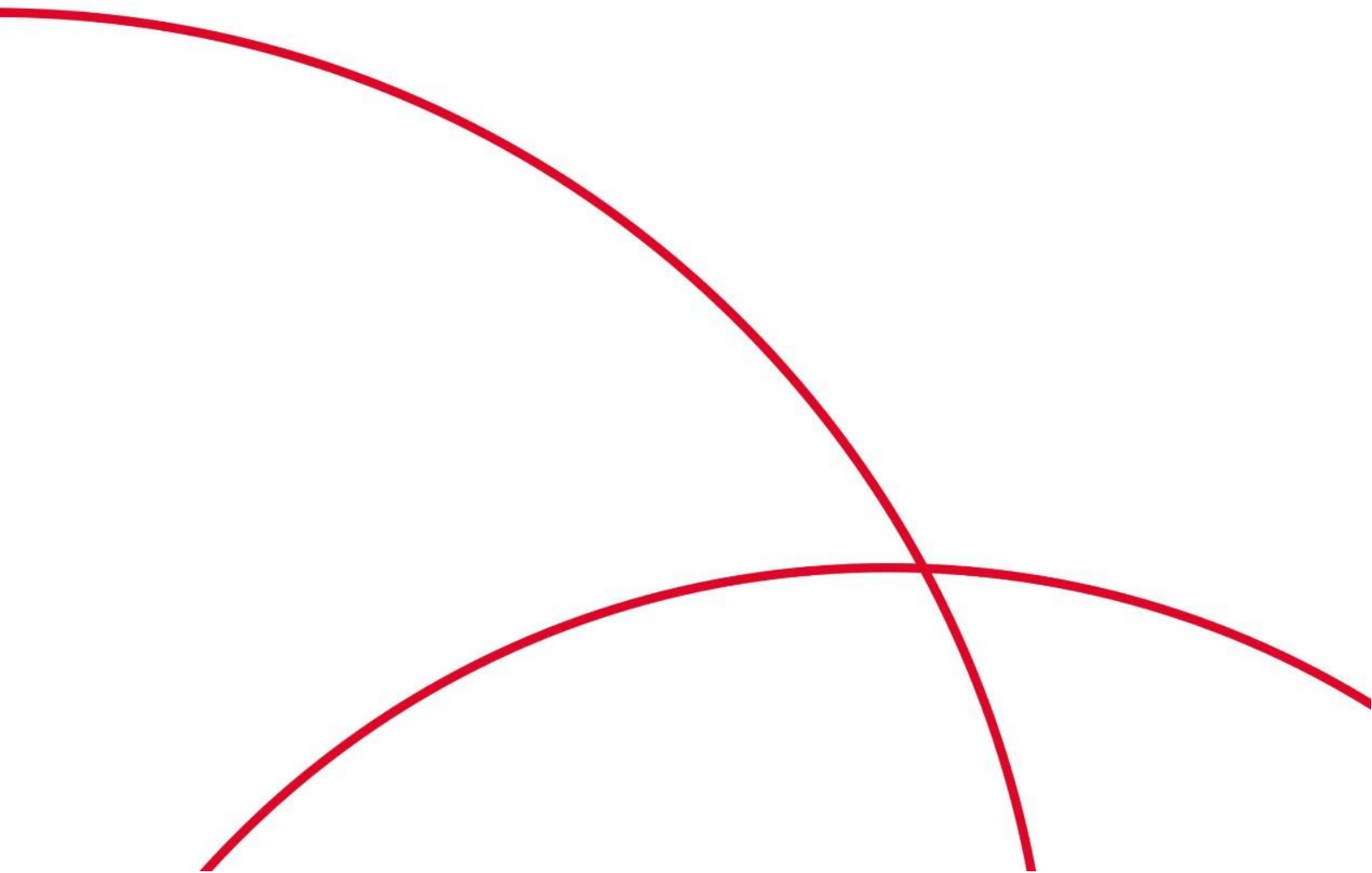




# 软件开发生产线 CodeArts

## 制品仓库常见问题

天翼云科技有限公司



---

# 目 录

---

<b>1 软件发布库</b> .....	<b>3</b>
1.1 制品仓库服务下软件发布库首页为什么不能上传文件、创建目录等? .....	3
1.2 是否可以通过直接更改 pom.xml 文件中的依赖 ID 来调用软件发布库中的 jar 包? .....	3
<b>2 私有依赖库</b> .....	<b>4</b>
2.1 如何将 snapshot 组件上传到 Maven 私有依赖库? .....	4
2.2 如何正确调用 Maven 私有依赖库的私有组件? .....	6
2.3 支持本地编译时调用私有依赖库中的软件包吗? .....	7
2.4 仓库接收不到请求 (Connection Reset) .....	7
2.5 无法下载依赖的 war、jar 文件 .....	8
2.6 本地构建 Maven 组件上传到私有依赖库, 返回 401 错误提示 .....	8
2.7 私有依赖库回收站中文件无法还原 .....	9

# 1 软件发布库

制品仓库服务下软件发布库首页为什么不能上传文件、创建目录等？

是否可以通过直接更改 pom.xml 文件中的依赖 ID 来调用软件发布库中的 jar 包？

## 1.1 制品仓库服务下软件发布库首页为什么不能上传文件、创建目录等？

软件发布库页面中，列表中的顶层目录名对应租户下的项目名，用于区分归档软件包所属项目。

在该目录下不允许进行上传文件、创建目录等操作，只能浏览。

单击项目名称进入项目文件夹后即可进行上传文件、创建目录等操作。

## 1.2 是否可以通过直接更改 pom.xml 文件中的依赖 ID 来调用软件发布库中的 jar 包？

不可以。

因为发布库的软件包是用于部署，而不是用于编译构建过程中的依赖。

若需要引用依赖，请将依赖包上传至私有依赖库中。

# 2 私有依赖库

- 如何将 snapshot 组件上传到 Maven 私有依赖库？
- 如何正确调用 Maven 私有依赖库的私有组件？
- 支持本地编译时调用私有依赖库中的软件包吗？
- 仓库接收不到请求（Connection Reset）
- 无法下载依赖的 war、jar 文件
- 本地构建 Maven 组件上传到私有依赖库，返回 401 错误提示
- 私有依赖库回收站中文件无法还原

## 2.1 如何将 snapshot 组件上传到 Maven 私有依赖库？

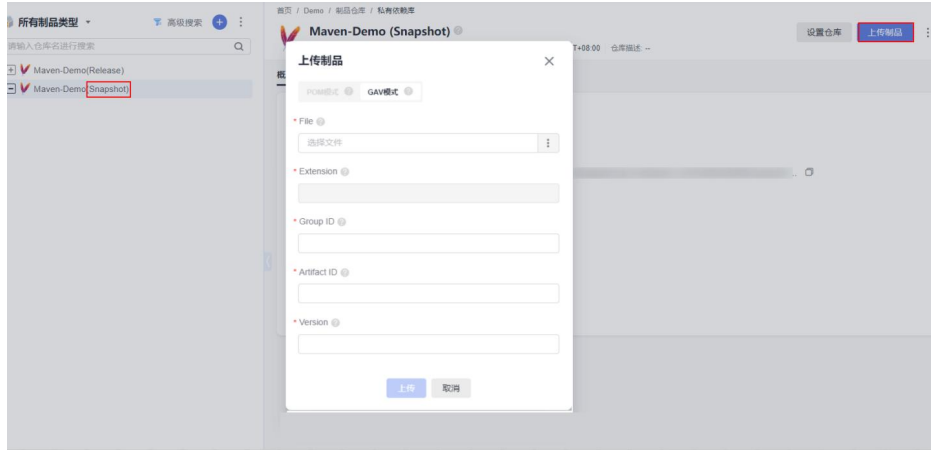
### 背景信息

制品仓库服务支持通过几种方式上传 snapshot 组件：

- 通过页面上上传 snapshot 组件
- 通过 Maven CLI 命令上传 snapshot 组件
- 通过编译构建将 snapshot 组件发布到 Maven 私有依赖库

### 通过页面上上传 snapshot 组件

- 步骤 1 登录软件开发生产线首页。
- 步骤 2 单击导航“服务 > 制品仓库”，选择“私有依赖库”页签，找到对应的仓库。
- 步骤 3 在页面左侧仓库列表中选择 snapshot 仓库。单击“上传制品”，弹出“上传 snapshot 组件”对话框，请根据需要选择 GAV 定义来源。



GAV 定义来源有以下两种：


GAV 定义来源模式	描述
POM 模式	从 POM 文件中解析出 Maven 的 GAV 坐标信息，然后再上传。
GAV 模式	直接指定 Maven 包的坐标信息再上传。

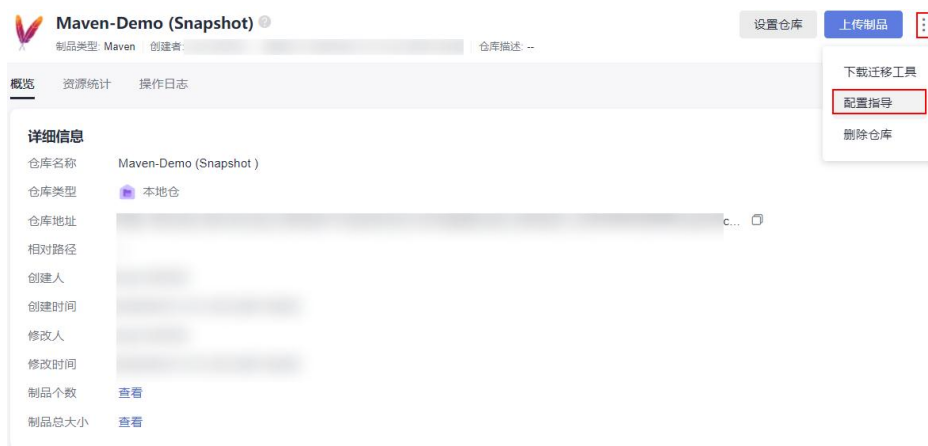
步骤 4 根据界面提示填写相关参数、上传对应文档完成组件上传。

----结束

## 通过 Maven CLI 命令上传 snapshot 组件

步骤 1 进入“Maven 私有依赖库”，在页面左侧导航中选择“snapshot”类型仓库。

步骤 2 单击右上角，在下拉栏中单击“配置指导”弹出“私有依赖库使用配置”对话框。




步骤 3 根据配置指导配置本地 Maven 工具。

步骤 4 执行 `mvn deploy` 命令上传 Maven 项目。

在命令窗口进入 maven 项目“pom.xml”所在目录，输入如下命令，上传本地 jar 包：

```
mvn deploy:deploy-file -DgroupId=com. -DartifactId=aopalliance -Dversion=1.0-SNAPSHOT -Dpackaging=jar -Dfile=D:\aopalliance-1.0-SNAPSHOT.jar -Durl={Maven Snapshot 地址} -DrepositoryId=snapshots
```

### 📖 说明

- DgroupId、DartifactId、Dversion、Dpackaging：可根据需要自定义。
- Dfile：本地 jar 包所在位置的绝对路径。
- Durl：Maven Snapshot 地址。单击上图中的图标  即可获取。

----结束

## 通过编译构建将 snapshot 组件发布到 Maven 私有依赖库

步骤 1 进入代码仓库，打开 pom.xml 文件，定义待上传组件的坐标信息（GAV 信息）。

### 📖 说明

- 构建任务执行时会根据定义识别上传到私有依赖库的组件属性信息。
- version：版本号，默认上传 release 组件。如果需要上传 snapshot 组件，version 值以“-SNAPSHOT”结尾即可，如“1.0-SNAPSHOT”。

步骤 2 编辑编译构建任务，在“构建步骤 Maven 构建”中：

- 在“命令”框中：使用注释 mvn package 命令（在命令前加#），取消注释 mvn deploy（删除其命令前的#）。
- 单击“发布依赖包到 CodeArts 私有依赖库”，勾选“配置所有 pom”。

步骤 3 执行构建任务，任务执行成功后即可在私有依赖库中找到生成的 Maven 组件。

----结束

## 2.2 如何正确调用 Maven 私有依赖库的私有组件？

步骤 1 进入私有依赖库，单击需要私有组件名称，进入文件属性页面。




步骤 2 获取依赖下载地址，复制后添加至“pom.xml”文件即可。

----结束

## 2.3 支持本地编译时调用私有依赖库中的软件包吗？

支持。

进入依赖包所在的仓库，单击页面右上角，在下拉栏中单击“配置指导”下载配置文件，参考弹框中的说明完成配置即可。

## 2.4 仓库接收不到请求（Connection Reset）

### 问题现象

本地构建失败，提示“Connection Reset”，日志信息提示如下：

```
[ERROR] Failed to execute goal org.apache.maven.plugins:maven-deploy-plugin:2.7:deploy (default-deploy) on project base-parent: Failed to retrieve remote metadata [REDACTED].framework:base-parent:4.1.200-SNAPSHOT/maven-metadata.xml: Could not transfer metadata [REDACTED].framework:base-parent:4.1.200-SNAPSHOT/maven-metadata.xml from/to snapshots (https://devrepo.devcloud.cn-north-4.huaweicloud.com/[REDACTED]) [REDACTED].framework:base-parent:4.1.200-SNAPSHOT/maven-metadata.xml: Connection reset -> [Help 1]
[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:
```

### 原因分析

用户使用的 Java 版本低，默认不支持 TLS 1.2。

### 处理方法

- 若您使用的是 Java 6 版本：请升级到 Java 8 以上的版本。
- 若您使用的是 Java 7 版本：Java 7 支持 TLS 1.2，但在 1.7.0\_131-b31 之前的版本中默认禁用，可以使用以下命令启用。

```
mvn -Dhttps.protocols= TLSv1.2 <goals>
```

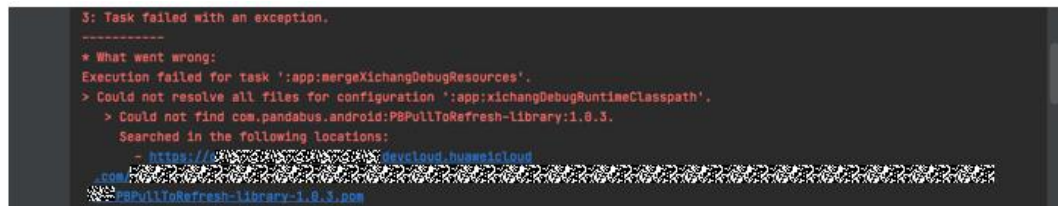
或将以下命令添加到您的环境或构建脚本中。

```
export MAVEN_OPTS=-Dhttps.protocols= TLSv1.2
```

## 2.5 无法下载依赖的 war、jar 文件

### 问题现象

本地工具通过私有依赖库无法下载组件，报错为找不到对应的 pom 文件，日志信息提示如下：



```
3: Task failed with an exception.
-----
* What went wrong:
Execution failed for task ':app:mergeXichangDebugResources'.
> Could not resolve all files for configuration ':app:xichangDebugRuntimeClasspath'.
> Could not find com.pandabus.android:PBPullToRefresh-Library:1.8.3.
Searched in the following locations:
- https://.../devcloud_huaweicloud...com:PBPullToRefresh-Library-1.8.3.pom
```

### 原因分析

依赖中缺失对应的 pom 文件。

无论是通过 gradle 还是 maven 下载依赖，其顺序都为先下载 pom 文件，然后才是 jar 或者 war，若云端不存在 pom 文件，则会下载失败。

### 处理方法

建议按照组件上传标准再次重新上传无法下载的组件。

## 2.6 本地构建 Maven 组件上传到私有依赖库，返回 401 错误提示

### 问题现象

通过本地 IDE 向私有依赖库上传 Maven 组件时报错：401-权限不足。

### 原因分析


代码仓库文件“pom.xml”中配置的私有依赖库信息，与私有依赖库配置文件“settings.xml”不符。

### 处理方法

上传组件时，将文件“pom.xml”中“distributionManagement”元素下“repository\_id”的值，替换为文件“settings.xml”中“server”元素下“repository\_id”的值。



上传组件操作步骤如下：

- 步骤 1 进入“Maven 私有依赖库”，在页面左侧导航中选择 Maven 类型仓库。
- 步骤 2 单击页面右上角，在下拉栏中单击“配置指导”，弹出“私有依赖库使用配置”对话框。
- 步骤 3 根据配置指导配置本地 Maven 工具。
- 步骤 4 执行 `mvn deploy` 命令上传 Maven 项目。

- 在命令窗口进入 maven 项目“pom.xml”所在目录，检查文件“pom.xml”中“distributionManagement”元素下的“repository\_id”的值是否与文件“settings.xml”中“server”元素下“repository\_id”的值是否匹配。

setting.xml

```
<server>
  <id>release_cn-north-1</id>
  <username>cn-north-1</username>
  <password></password>
</server>
<server>
  <id>snapshot_cn-north-1</id>
  <username>cn-north-1</username>
  <password></password>
</server>
<server>
  <id>z_mirrors</id>
</server>
```

pom.xml

```
<dependencies>
  <dependency>
    <groupId></groupId>
    <artifactId>a</artifactId>
    <version>a</version>
  </dependency>
</dependencies>
<distributionManagement>
  <repository>
    <id>release_cn-north-1</id>
    <url></url>
  </repository>
  <snapshotRepository>
    <id></id>
    <url></url>
  </snapshotRepository>
</distributionManagement>
</project>
```

- 执行如下命令，上传本地 jar 包：

```
mvn deploy:deploy-file -DgroupId=com. -DartifactId=aopalliance -Dversion=1.0 -Dpackaging=jar
```

----结束

## 2.7 私有依赖库回收站中文件无法还原

### 问题现象

在回收站页面上对文件执行还原操作，文件无法还原。

### 原因分析

仓库中的被还原位置上存在一个同名的文件，还原时会跳过该文件。

## 处理方法

仓库中被还原的原位置已存在同名文件，建议忽略该文件的还原操作。若必须还原该文件，需要先将仓库中的同名文件删除。