

# Android Gradle的依赖仓库



Joker\_Lee

关注

 0.091

2020.05.11 22:54:43

字数 1,005

阅读 1,352

Gradle构建工具结合了Ant 和 Maven的优点，它具有Ant的强大和灵活，也有maven的生命周期管理和易用性，又沿用并改进了Maven的依赖管理体系。所以介绍Gradle的依赖仓库离不开Maven仓库！

## Maven仓库

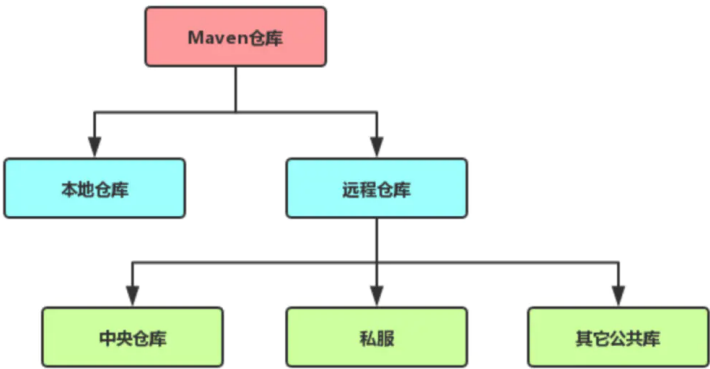
Maven最核心的改进就在于提出仓库这个概念。可以把所有依赖的包，都放到网络仓库里去，在工程里标明我需要什么什么包，什么版本。在构建的时候，maven就自动帮我把这些包引入了，不需要自己去管理几十上百个jar文件了。

Maven为了改进依赖包引入方式提出了网络中央仓库（类似AppleStore）里面存储所有第三方依赖包；工程里面只要加入仓库路径+依赖包的唯一标识即可自动通过网络加载引入依赖包。

Maven仓库整个可以分为两大类：本地和远程仓库，远程仓库又定义了个特殊的仓库-->中央仓库。先在本地仓库中寻找，如果找不到，再去远程仓库寻找。

本地仓库 默认地址为~/.m2/, 一个构件只有在本地仓库存在之后, 才能由Maven项目使用。因为Android项目不属于Maven项目，所以通常没有本地仓库，但是它会有本地缓存，在~/.gradle/caches/modules-2/files-2.1目录下面。

远程仓库分为中央仓库、私库，中央仓库是一些公司或开源组织提供的全球所有用户都能使用的仓库（也有说法：中央仓库特指Maven社区提供的MavenCentral），私库是局域网或者指定组织才能使用的仓库。很多公司都搭有自己的私库，当用户想私库请求一个构件，私库中不存在，它通常会先去中央仓库请求，然后缓存下来再为用户提供下载。



仓库类型

## Maven的坐标体系

Maven制定了标识一个依赖库的标准，一个maven仓库内的依赖包必须包含groupId、artifactId、version三个信息；

groupId是工程组的标识，通常是一个项目的唯一标识，类似Android项目的包名

artifactId是工程的标识。它通常是工程的名称，groupId 和 artifactId 一起定义了 artifact 在仓库中的位置。

这是工程的版本号。在 artifact 的仓库中，它用来区分不同的版本。

```
1 //gradle 沿用的这套坐标体系
2 compile 'com.alibaba:fastjson:1.2.68'
3 //groupId -> com.alibaba
4 //artifactId -> fastjson
5 //version -> 1.2.68
```

## Android常用的Maven仓库

### google仓库

在gradle4.1之后，添加了新的语法google()，用于引用google自有的仓库。

浏览器查看：<https://maven.google.com/web/index.html>

```
classpath 'com.android.tools.build:gradle:3.6.3'
```

比如studio版本对不上gradle插件版本，也不知道当前gradle插件有哪些版本号可以去浏览查看

### Google's Maven Repository

com.android.tools.build Find Clear

```
- com.android.tools.build
+ aapt2
+ aapt2-proto
+ aaptcompiler
+ apksig
+ apkzlib
+ builder
+ builder-model
+ builder-test-api
+ bundletool
- gradle
  3.0.0-alpha1
  3.0.0-alpha2
  3.0.0-alpha3
  3.0.0-alpha4
  3.0.0-alpha5
  3.0.0-alpha6
  3.0.0-alpha7
  3.0.0-alpha8
  3.0.0-alpha9
  3.0.0-beta1
  3.0.0-beta2
  3.0.0-beta3
  3.0.0-beta4
```

google仓库查询

jcenter仓库

jcenter是由 bintray.com维护的Maven仓库, 通过地址查看仓库内容:  
<https://bintray.com/bintray/jcenter>

mavenCentral

早期的android使用了mavenCentral作为主要的仓库, 现在逐步使用Jcenter替代, 内容查看地址:  
<https://search.maven.org/>

Jcenter使用https & CDN, 性能和占存储大小方面比mavenCentral更优。

```
1 //目前Android默认的构建脚本会自动添加google & jcenter作为默认仓库
2 allprojects {
3     repositories {
4         google() //--> https://dl.google.com/dl/android/maven2/
5         jcenter() // --> https://jcenter.bintray.com/
6     }
7 }
```

国内镜像

这些常用的仓库服务器基本都在海外, 终所周知的原因, 我们需要更快的访问速度只能借助镜像, 幸好阿里大佬基本都免费镜像了这些仓库:

[阿里云-公共代理库](#)  
[阿里云-公共代理库在线搜索查看](#)

代理的仓库列表		
仓库名称	代理源地址	使用地址
central	<a href="https://repo1.maven.org/maven2/">https://repo1.maven.org/maven2/</a>	<a href="https://maven.aliyun.com/repository/central">https://maven.aliyun.com/repository/central</a> 或 <a href="https://maven.aliyun.com/nexus/content/repositories/central">https://maven.aliyun.com/nexus/content/repositories/central</a>
jcenter	<a href="http://jcenter.bintray.com/">http://jcenter.bintray.com/</a>	<a href="https://maven.aliyun.com/repository/jcenter">https://maven.aliyun.com/repository/jcenter</a> 或 <a href="https://maven.aliyun.com/nexus/content/repositories/jcenter">https://maven.aliyun.com/nexus/content/repositories/jcenter</a>
public	central仓和jcenter仓的聚合仓	<a href="https://maven.aliyun.com/repository/public">https://maven.aliyun.com/repository/public</a> 或 <a href="https://maven.aliyun.com/nexus/content/groups/public">https://maven.aliyun.com/nexus/content/groups/public</a>
google	<a href="https://maven.google.com/">https://maven.google.com/</a>	<a href="https://maven.aliyun.com/repository/google">https://maven.aliyun.com/repository/google</a> 或 <a href="https://maven.aliyun.com/nexus/content/repositories/google">https://maven.aliyun.com/nexus/content/repositories/google</a>
gradle-plugin	<a href="https://plugins.gradle.org/m2/">https://plugins.gradle.org/m2/</a>	<a href="https://maven.aliyun.com/repository/gradle-plugin">https://maven.aliyun.com/repository/gradle-plugin</a> 或 <a href="https://maven.aliyun.com/nexus/content/repositories/gradle-plugin">https://maven.aliyun.com/nexus/content/repositories/gradle-plugin</a>

阿里代理的仓库

```
1 //阿里镜像仓库的使用
2 allprojects {
3     repositories {
4
5         maven { url 'https://maven.aliyun.com/repository/public/' }
```

```
6 |           //jcenter()
7 |
8 |           maven { url 'https://maven.aliyun.com/repository/google' }
9 |           //google()
10 |       }
11 |   }
```

参考

- Maven基础教程之仓库
- Android Studio中的mavenCentral、jcenter、google仓库
- Gradle 缓存目录结构 缓存策略

<https://www.jianshu.com/p/cbb706470648>

1人点赞 >

Gradle系列



**Joker\_Lee**  
总资产7 共写了1.8W字 获得35个赞 共11个粉丝

关注