

CSDN新首页上线啦，邀请你来立即体验！(http://blog.csdn.net/)

CSDN

博客 (//blog.csdn.net?ref=toolbar) 学院 (//edu.csdn.net?ref=toolbar)

下载 (//download.csdn.net?ref=toolbar) GitChat (//gitbook.cn/?ref=csdn)

更多 

2



登录 (https://passport.csdn.net/account/login?ref=toolbar) 注册 (https://passport.csdn.net/account/mobile/register?ref=toolbar&action=mobile) 注册 (https://passport.csdn.net/account/mobile/register?ref=toolbar&action=mobile) 注册 (https://passport.csdn.net/account/mobile/register?ref=toolbar&action=mobile)

Android studio Maven仓库使用

转载

2017年02月14日 10:12:36

6543

转自：http://www.cnblogs.com/sihaixuan/p/4852974.html

原文：How to distribute your own Android library through jCenter and Maven Central from Android Studio (http://inthecheesefactory.com/blog/how-to-upload-library-to-jcenter-maven-central-as-dependency/en)

转自：翻译 http://www.jcodecraeer.com/a/anzhuokaifa/androidkaifa/2015/0623/3097.html

(http://www.jcodecraeer.com/a/anzhuokaifa/androidkaifa/2015/0623/3097.html)

如果你想在Android Studio中引入一个library到你的项目，你只需添加如下的一行代码到模块的build.gradle文件中。

```
dependencies {
    compile 'com.inthecheesefactory.thecheeselibrary:fb-like:0.9.3'
}
```

就是如此简单的一行代码，你就可以使用这个library了。

酷呆了。不过你可能很好奇Android Studio是从哪里得到这个library的。这篇文章将详细讲解这是怎么回事，包括如何把你的库发布出去分享给世界各地的其他开发者，这样不仅可以让世界更美好，还可以要一

次酷。加入CSDN，享受更精准的内容推荐，与5000万程序员共同成长！

立即体验



望京soho



羊驼租赁



jinyip (http://blog.csdn.net/)

+ 关注

(http://blog.csdn.net/jinyip)

码云

未开通

(https://gitee.com/utm_source)

原创
3

粉丝
3

喜欢
0

他的最新文章

更多文章 (http://blog.csdn.net/jinyip)

Android studio使用JNI实例 (http://blog.csdn.net/jinyip/article/details/78362209)

什么时候使用CountDownLatch (http://blog.csdn.net/jinyip/article/details/73614281)

像大牛一样写代码: 31个Android 开发者工具 (http://blog.csdn.net/jinyip/article/details/72820332)

登录

注册

内容举报

返回顶部



Android studio 是从哪里得到库的？

先从这个简单的问题开始，我相信不是每个人都完全明白Android studio 是从哪里得到这些library的。莫非就是Android studio 从google搜索然后下载了一个合适的给我们？

呵呵，没那么复杂。Android Studio是从build.gradle里面定义的Maven 仓库服务器上下载library的。Apache Maven是Apache开发的一个工具，提供了用于贡献library的文件服务器。总的来说，只有两个标准的Android library文件服务器：jcenter 和 Maven Central。

2

jcenter

jcenter是一个由 bintray.com维护的Maven仓库。你可以在这里 (<http://jcenter.bintray.com/>)看到整个仓库的内容。

我们在项目的build.gradle 文件中如下定义仓库，就能使用jcenter了：

```
allprojects {
    repositories {
        jcenter()
    }
}
```

Maven Central

Maven Central 则是由sonatype.org (<https://sonatype.org/>)维护的Maven仓库。你可以在这里 (<https://oss.sonatype.org/content/repositories/releases/>)看到整个仓库。

注：不管是jcenter还是Maven Central，两者都是Maven仓库

我们在项目的build.gradle 文件中如下定义仓库，就能使用Maven Central了：

```
allprojects {
    repositories {
        mavenCentral()
    }
}
```

加入CSDN，享受更精准的内容推荐，与5000万程序员共同成长！

大数据浪潮下，前端工程师眼中的完整support.csdn.net/a
数据链图 (<http://blog.csdn.net/details/72465765>)

在windows环境下编译hadoop
og.csdn.net/jinyp/article/details/55095310



望京soho



羊驼租赁

相关推荐

AndroidStudio关联Maven仓库
n.net/u010680097/article/details/75039653

AndroidStudio 使用本地Maven仓库 (<http://blog.csdn.net/fyfcauc/article/details/70174960>)

Android Studio之maven Central, JCenter (http://blog.csdn.net/lu_xin_/article/details/51134849)

AndroidStudio使用进阶二：搭建自己的maven私服，并使用Gradle统一依赖管理 (http://blog.csdn.net/JF_1994/article/details/51228560)



内容举报



返回顶部

登录

注册



注意，虽然jcenter和Maven Central 都是标准的 android library仓库，但是它们维护在完全不同的服务器上，由不同的人提供内容，两者之间毫无关系。在jcenter上有的可能 Maven Central 上没有，反之亦然。

除了两个标准的服务器之外，如果我们使用的library的作者是把该library放在自己的服务器上，我们还可以自己定义特有的Maven仓库服务器。Twitter的Fabric.io 就是这种情况，它们在<https://maven.fabric.io/public> (<https://maven.fabric.io/public>)上维护了一个自己的Maven仓库。如果你想使用Fabric.io的library，你必须自己如下定义仓库的url。

```
repositories {
    maven { url 'https://maven.fabric.io/public' }
}
```

然后在里面使用相同的方法获取一个library。

```
dependencies {
    compile 'com.crashlytics.sdk.android:crashlytics:2.2.4@aar'
```

但是将library上传到标准的服务器与自建服务器，哪种方法更好呢？当然是前者。如果将我们的library公开，其他开发者除了一行定义依赖名的代码之外不需要定义任何东西。因此这篇文章中，我们将只关注对开发者更友好的jcenter 和 Maven Central。

实际上可以在Android Studio上使用的除了Maven 仓库之外还有另外一种仓库：Ivy 仓库 (<http://ant.apache.org/ivy/>)。但是根据我的经验来看，我还没看到任何人用过它，包括我，因此本文就直接忽略了。

理解jcenter和Maven Central

为何有两个标准的仓库？

事实上两个仓库都具有相同的使命：提供Java或者Android library服务。上传到哪个（或者都上传）取决于开发者。

起初，Android Studio 选择Maven Central作为默认仓库。如果你使用老版本的Android Studio创建一个新加入项目，你会发现它会自动添加Maven Central仓库。随着项目的增长，你会发现Maven Central仓库已经无法满足你的需求，这时你可以选择添加jcenter仓库，这样你就可以使用更多的第三方库了。Maven Central和jcenter仓库都是由Google维护的，它们都位于Google的服务器上，因此它们的速度非常快。Maven Central和jcenter仓库都是由Google维护的，它们都位于Google的服务器上，因此它们的速度非常快。

(<https://passport.csdn.net/a>)

IT 外包
 望京soho

羊驼租赁
 羊驼租赁

agv搬运机器人
 仓库仓储

小型仓库出租
 公寓出租

他的热门文章

Android studio Maven仓库使用 (<http://blog.csdn.net/jinyp/article/details/55095310>) 内容举报

6477

使用 TeamCity 实现持续集成 (CI) (<http://blog.csdn.net/jinyp/article/details/54406794>) 返回顶部

1228

登录

注册



但是Maven Central的最大问题是对开发者不够友好。上传library异常困难。上传上去的开发者都是某种程度的极客。同时还因为诸如安全方面的其他原因，Android Studio团队决定把默认的仓库替换成jcenter。正如你看到的，一旦使用最新版本的Android Studio创建一个项目，jcenter()自动被定义，而不是mavenCentral()。

有许多将Maven Central替换成jcenter的理由，下面是几个主要的原因。

- jcenter通过CDN发送library，开发者可以享受到更快的下载体验。
- jcenter是全世界最大的Java仓库，因此在Maven Central上有的，在jcenter上也极有可能有。换句话说jcenter是Maven Central的超集。
- 上传library到仓库很简单，不需要像在Maven Central上做很多复杂的事情。
- 友好的用户界面
- 如果你想把library上传到Maven Central，你可以在bintray网站上直接点击一个按钮就能实现。

基于上面的原因以及我自己的经验，可以说替换到jcenter是明智之举。

所以我们这篇文章将把重心放在jcenter，反正如果你能成功把library放在jcenter，转到Maven Central是非常容易的事情。

gradle是如何从仓库上获取一个library的？

在讨论如何上传library到jcenter之前，我们先看看gradle是如何从仓库获取library的。比如我们在build.gradle输入如下代码的时候，这些库是如果奇迹般下载到我们的项目中的。

```
compile 'com.inthecheesefactory.thecheeselibrary:fb-like:0.9.3'
```

一般来说，我们需要知道library的字符串形式，包含3部分

```
GROUP_ID:ARTIFACT_ID:VERSION
```

上面的例子中，GROUP_ID是com.inthecheesefactory.thecheeselibrary，ARTIFACT_ID是fb-like，VERSION是0.9.3。

GROUP_ID定义了library的group。有可能在同样的上下文中存在多个不同功能的library。如果library具有相同的group，那么它们将共享一个GROUP_ID。通常我们以开发者包名紧跟着library的group名称来命名，比如com.inthecheesefactory.thecheeselibrary:fb-like:0.9.3。至于VERSION，就是版本号

RHEL6.3_x86_64 下载源码 <http://blog.csdn.net/a25014168>
(2.5) MySQL(5.5.25a) PHP
<http://blog.csdn.net/jinyp/article/details/917>

1036

Gson 使用 (<http://blog.csdn.net/details/9193531>)

815

<http://www.bloges.com/article/152141688018066> alizati

502



望京soho



羊驼租赁

广告

内容举报

返回顶部

登录

注册

×

而已，虽然可以是任意文字，但是我建议设置为x.y.z的形式，如果喜欢还可以加上beta这样的后缀。

下面是Square library的一个例子。你可以看到每个都可以很容易的分辨出library和开发者的名称。

```
dependencies {  
    compile 'com.squareup:otto:1.3.7'  
    compile 'com.squareup.picasso:picasso:2.5.2'  
    compile 'com.squareup.okhttp:okhttp:2.4.0'  
    compile 'com.squareup.retrofit:retrofit:1.9.0'  
}
```



那么在添加了上面的依赖之后会发生什么呢？简单。Gradle会询问Maven仓库服务器这个library是否存在，如果是，gradle会获得请求library的路径，一般这个路径都是这样的形式：

GROUP_ID/ARTIFACT_ID/VERSION_ID。比如可以在<http://jcenter.bintray.com/com/squareup/otto/1.3.7>

(<http://jcenter.bintray.com/com/squareup/otto/1.3.7>) 和

<https://oss.sonatype.org/content/repositories/releases/com/squareup/otto/1.3.7/>

(<https://oss.sonatype.org/content/repositories/releases/com/squareup/otto/1.3.7/>)

下获得com.squareup:otto:1.3.7的library文件。

然后Android Studio 将下载这些文件到我们的电脑上，与我们的项目一起编译。整个过程就是这么简单，一点都不复杂。

我相信你应该清楚的知道从仓库上下载的library只是存储在仓库服务器上的jar 或者aar文件而已。有点类似于自己去下载这些文件，拷贝然后和项目一起编译。但是使用gradle依赖管理的最大好处是你除了添加几行文字之外啥也不做。library一下子就可以在项目中使用了。

了解aar文件

等等，我刚才说了仓库中存储的有两种类型的library：jar 和 aar。jar文件大家都知道，但是什么是aar文件呢？

aar文件时在jar文件之上开发的。之所以有它是因为有些Android Library需要植入一些安卓特有的文件，比如AndroidManifest.xml，资源文件，Assets或者JNI。这些都不是jar文件的标准。

因此aar文件就时发明出来包含所有这些东西的。总的来说它和jar一样只是普通的zip文件，不过具有不同的文件结构。jar文件以classes.jar的名字被嵌入到aar文件中。其余的文件罗列如下：

加入CSDN，享受更精准的内容推荐，与5000万程序员共同成长！

(<https://passport.csdn.net/a>



望京soho



羊驼租赁

广告



内容举报



返回顶部

登录

注册



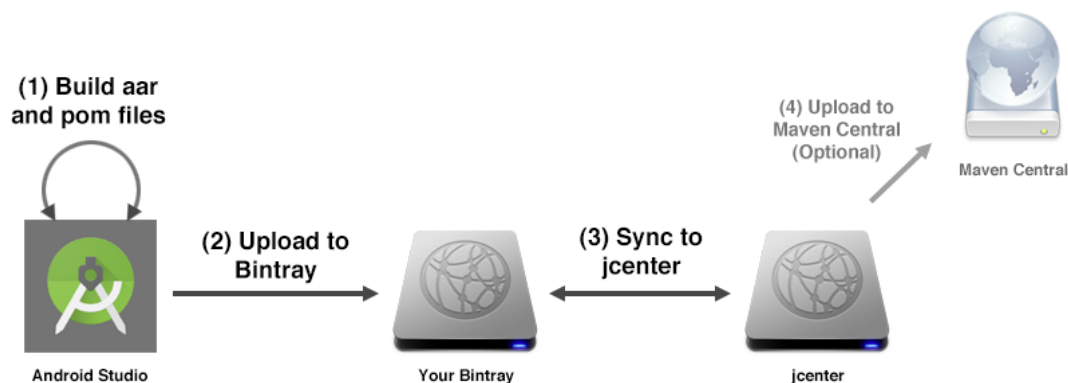
- /AndroidManifest.xml (mandatory)
- /classes.jar (mandatory)
- /res/ (mandatory)
- /R.txt (mandatory)
- /assets/ (optional)
- /libs/*.jar (optional)
- /jni/<abi>/*.so (optional)
- /proguard.txt (optional)
- /lint.jar (optional)

可以看到.aar文件是专门为安卓设计的。因此这篇文章将教你如何创建与上传一个aar形式的library。

如何上传library到jcenter

我相信你已经知道了仓库系统的大体工作原理。现在我们来开始最重要的部分：上传。这个任务和如何上传library文件到<http://jcenter.bintray.com> (<http://jcenter.bintray.com/>)一样简单。如果做到，这个library就算发布了。好吧，有两个需要考虑：如何创建aar文件以及如何上传构建的文件到仓库。

虽然需要若干步骤，但是我还是想强调这事并不复杂，因为已经准备好了所有事情。整个过程如下图：



加入CSDN，享受更精准的内容推荐，与5000万程序员共同成长！
因为细节比较多，我分为7部分，一步一步的详细解释清楚。

[登录](#)[注册](#)

内容举报

返回顶部

(<https://passport.csdn.net/a>



望京soho



羊驼租赁

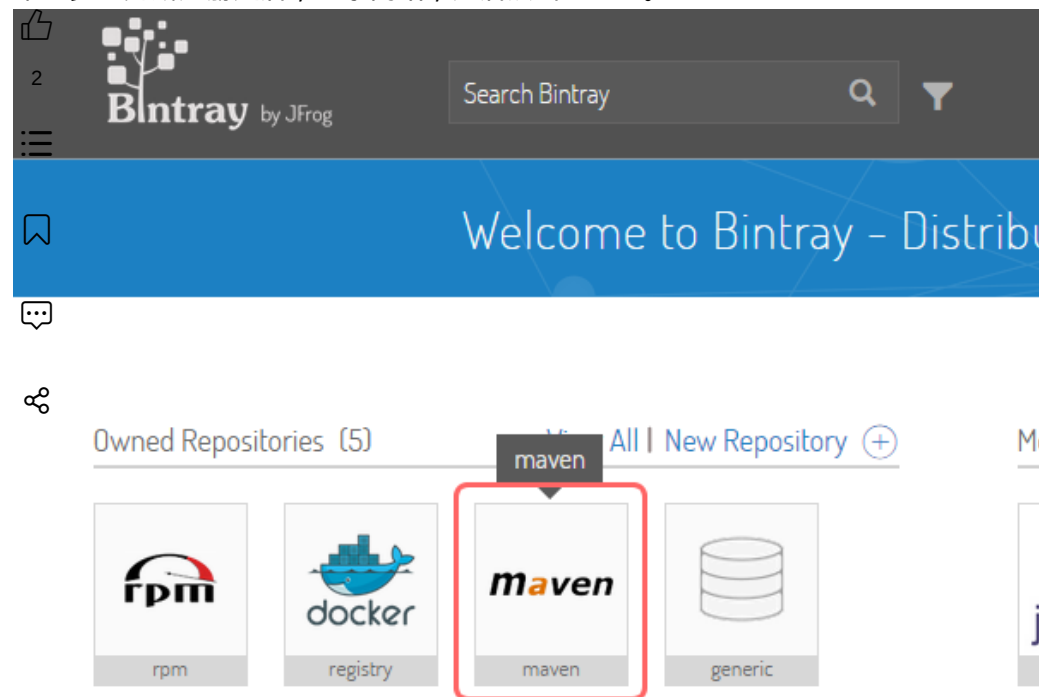
广告

第一部分：在bintray上创建package

首先，你需要在bintray上创建一个package。为此，你需要一个bintray账号，并在网站上创建一个package。

第一步：在bintray.com (<https://bintray.com/>)上注册一个账号。（注册过程很简单，自己完成）

第二步：完成注册之后，登录网站，然后点击maven。



第三步：点击Add New Package，为我们的library创建一个新的package。

加入CSDN，享受更精准的内容推荐，与5000万程序员共同成长！

登录

注册



(<https://passport.csdn.net/a>



望京soho

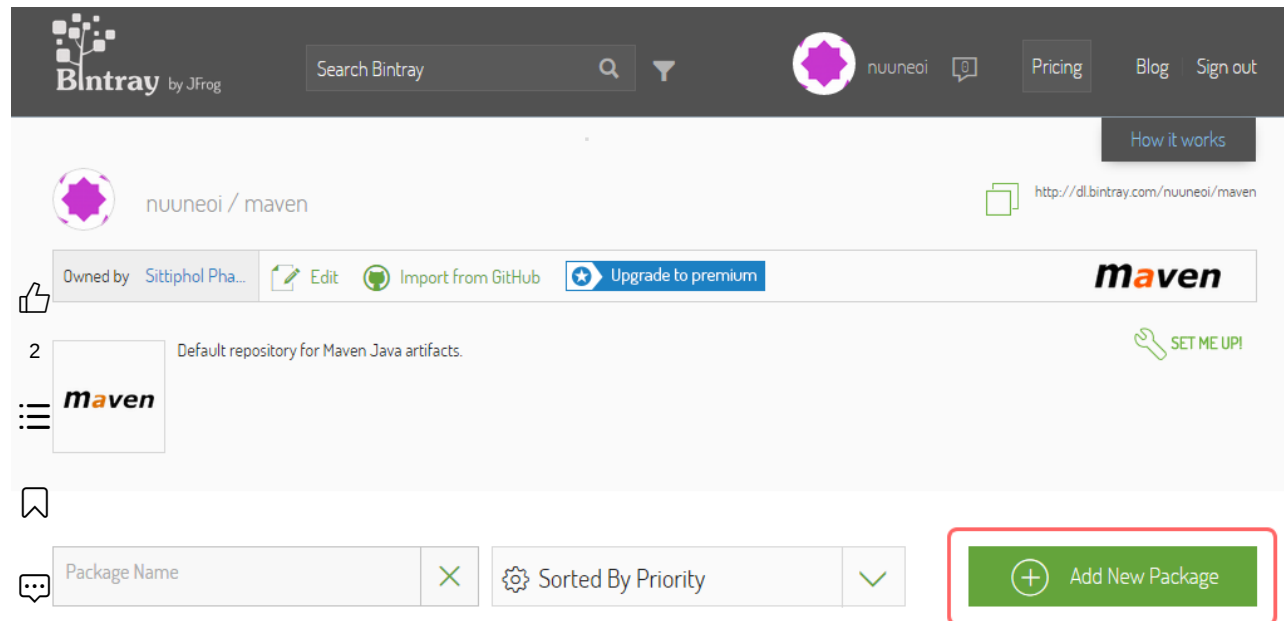


羊驼租赁

广告

内容举报

返回顶部



第四步：输入所有需要的信息

(https://passport.csdn.net/a



望京soho



羊驼租赁

广告

内容举报

返回顶部

加入CSDN，享受更精准的内容推荐，与5000万程序员共同成长！

登录


注册



Create New Package

Package Details

Package avatar

 [Upload a file](#)

Name*

fb-like

Description

A wrapper for Facebook Native Like Button (LikeView)

Licenses *

Apache-2.0

Start typing to see available license types

Request a new license type

Tags

android facebook like likeview like button

Enter comma-separated description tags

Website

https://github.com/nuuneoi/FBLikeAndroid

Issues tracker

https://github.com/nuuneoi/FBLikeAndroid/issues

Version control *

https://github.com/nuuneoi/FBLikeAndroid

☒ Make download numbers in stats public

Create Package Cancel

<https://passport.csdn.net/a>

望京soho



羊驼租赁

广告

虽然如何命名包名没有什么限定，但是也有一定规范。所有字母应该为小写，单词之间用 - 分割，比如，fb-like。

当每项都填完之后，点击Create Package。

第五步：网页将引导你到 Package编辑页面。点击 Edit Package文字下的Package名字，进入Package详情界面。




内容举报



返回顶部


加入CSDN，享受更精准的内容推荐，与5000万程序员共同成长！

[登录](#)[注册](#)




Edit Package
nuuneoi / maven / **fb-like**


fb-like




2 Package Details >




Attributes List




Versions List





Package Details

Package avatar



Upload a file

Name*

fb-like

Description

A wrapper for Facebook Native Like Button (LikeView) on Android

完工！现在你有了自己在Bintray上的Maven仓库，可以准备上传library到上面了。

(<https://passport.csdn.net/a>



望京soho



羊驼租赁

广告



内容举报



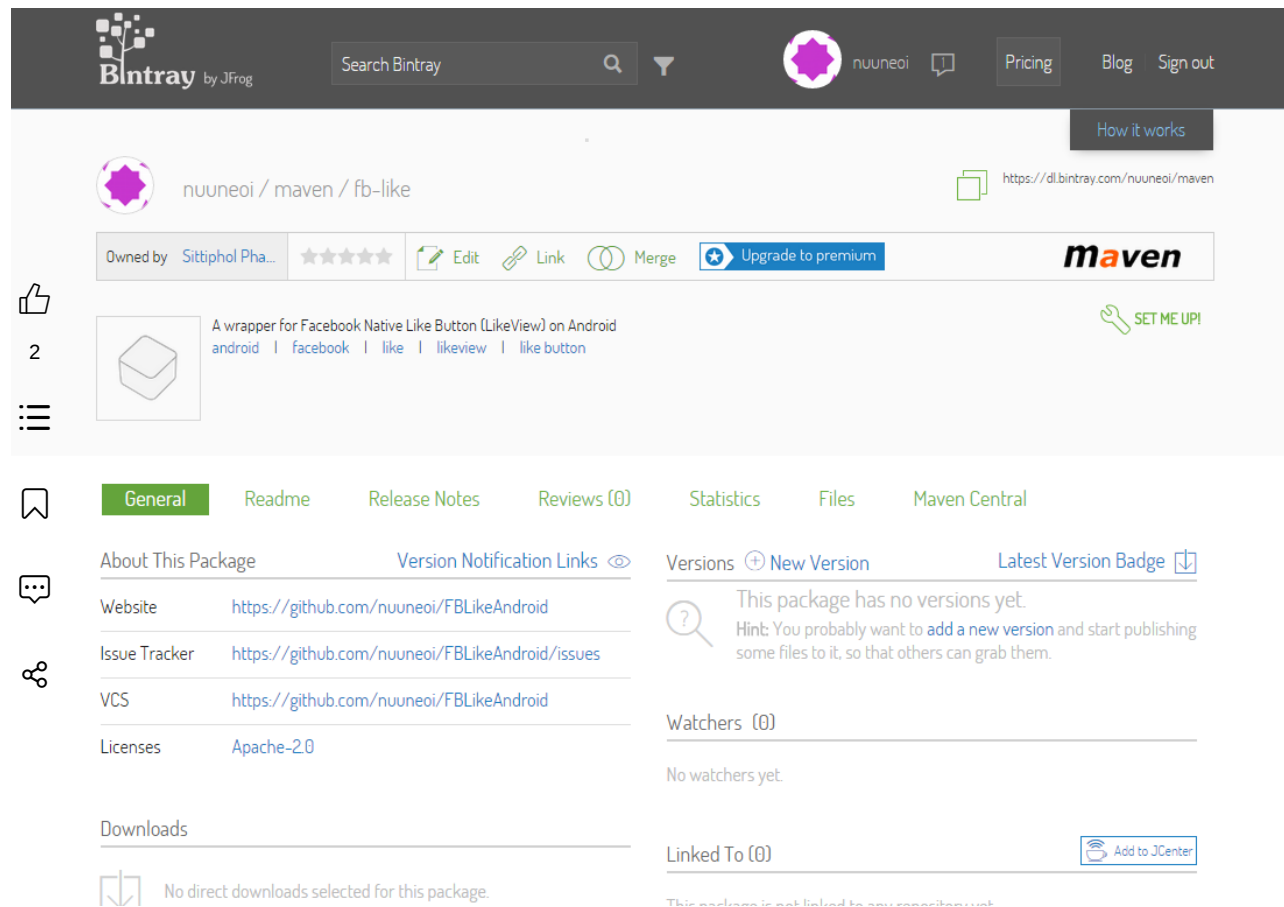
返回顶部

加入CSDN，享受更精准的内容推荐，与5000万程序员共同成长！

登录

注册





Bintray by JFrog

Search Bintray

nuuneoi / maven / fb-like

Owned by Sittiphol Pha... ★★★★★ [Edit](#) [Link](#) [Merge](#) [Upgrade to premium](#) **maven**

A wrapper for Facebook Native Like Button (LikeView) on Android
[android](#) | [facebook](#) | [like](#) | [likeview](#) | [like button](#)

General Readme Release Notes Reviews (0) Statistics Files Maven Central

About This Package Version Notification Links [Version Notification Links](#)

Website <https://github.com/nuuneoi/FBLikeAndroid>

Issue Tracker <https://github.com/nuuneoi/FBLikeAndroid/issues>

VCS <https://github.com/nuuneoi/FBLikeAndroid>

Licenses [Apache-2.0](#)

Downloads

No direct downloads selected for this package.

Versions [+ New Version](#) [Latest Version Badge](#)

This package has no versions yet.
Hint: You probably want to [add a new version](#) and start publishing some files to it, so that others can grab them.

Watchers (0)

No watchers yet.

Linked To (0) [Add to JCenter](#)

Bintray账户的注册就完成了。下一步是Sonatype，Maven Central 的提供者。

第二部分：为Maven Central创建个Sonatype帐号

注：如果你不打算把library上传到Maven Central，可以跳过第二和第三部分。不过我建议你不要跳过，因为仍然有许多开发者在使用这个仓库。

加入CSDN，享受更精准的内容推荐，与5000万程序员共同成长！

(<https://passport.csdn.net/a>



望京soho



羊驼租赁

广告

内容举报

返回顶部

登录

注册



和jcenter一样，如果你想通过Maven Central贡献自己的library，你需要在提供者的网站Sonatype上注册一个帐号。

你需要知道的就是这个帐号，你需要在Sonatype网站上创建一个IRA Issue Tracker 帐号。请到Sonatype Dashboard (<https://issues.sonatype.org/secure/Dashboard.jspa>) 注册这个帐号。

完成之后。你需要请求得到贡献library到Maven Central的权限。不过这个过程对我来说有点无厘头，因为你需要做的就是JIRA中创建一个issue，让它们允许你上传匹配Maven Central提供的GROUP_ID的library。

要创建上述所讲到的issue，访问Sonatype Dashboard

(<https://issues.sonatype.org/secure/Dashboard.jspa>)，用创建的帐号登录。然后点击顶部菜单的Create。

填写如下信息：

Project: Community Support - Open Source Project Repository Hosting

Issue Type: New Project

Summary: 你的 library名称的概要，比如The Cheese Library。

Group Id: 输入根GROUP_ID，比如，com.inthecheeselibrary。一旦批准之后，每个以com.inthecheeselibrary开始的library都允许被上传到仓库，比如com.inthecheeselibrary.somelib。

Project URL: 输入任意一个你想贡献的library的URL，比如，<https://github.com/nuuneoi/FBLikeAndroid> (<https://github.com/nuuneoi/FBLikeAndroid>)。

SCM URL: 版本控制的URL，比如 <https://github.com/nuuneoi/FBLikeAndroid.git> (<https://github.com/nuuneoi/FBLikeAndroid.git>)。

其余的不用管，然后点击Create。现在是最难的部分...耐心等待...平均大概1周左右，你会获准把自己的library分享到 Maven Central。

最后一件事是在Bintray Profile (<https://bintray.com/profile/edit>)的帐户选项中填写自己的Sonatype OSS用户名。

(<https://passport.csdn.net/a>



望京soho



羊驼租赁

广告



内容举报



返回顶部

加入CSDN，享受更精准的内容推荐，与5000万程序员共同成长！

登录

注册



<https://passport.csdn.net/a>

望京soho



羊驼租赁

广告

点击Update，完成。

第三部分：启用bintray里的自动注册

就如我上面提到的，我们可以通过jcenter上传library到Maven Central，不过我们需要先注册这个library。

bintray提供了通过用户界面让library一旦上传后自动注册的机制。

第一步是使用下面的命令行产生一个key。（如果你用的是windows，请在cygwin

(<https://www.cygwin.com/>)下做这件事情）

```
gpg --gen-key
```

有几个必填项。部分可以采用默认值，但是某些项需要你自己输入恰当的内容，比如，你的真实名字，密码等等。

创建了key之后，调用如下的命令查看被创建key的信息。

加入CSDN，享受更精准的内容推荐，与5000万程序员共同成长！

```
gpg --list-keys
```



内容举报



返回顶部

登录

注册



如果没问题的话，可以看到下面的信息：

```
pub 2048R/01ABCDEF 2015-03-07
uid Sittiphol Phanvilai <yourmail@email.com>
sub 2048R/98765432 2015-03-07
```



现在你需要把key上传到keyserver让它发挥作用。为此，请调用如下的命令并且将其中的PUBLIC_KEY_ID替换成上面pub一行中2048R/ 后面的 8位16进制值，譬如本例是01ABCDEF。

```
gpg --keyserver hkp://pool.sks-keyservers.net --send-keys PUBLIC_KEY_ID
```



然后，使用如下的命令以ASCII形式导出公共和私有的key，请将yourmail@email.com替换成你前面用于创建key的email。

```
gpg -a --export yourmail@email.com > public_key_sender.asc
gpg -a --export-secret-key yourmail@email.com > private_key_sender.asc
```



打开Bintray的Edit Profile (<https://bintray.com/profile/edit>)页面点击GPG 注册。分别在Public Key和 Private Key中填入上一步导出的public_key_sender.asc和 private_key_sender.asc文件中的内容。

(<https://passport.csdn.net/a>



望京soho



羊驼租赁

广告



内容举报



返回顶部

加入CSDN，享受更精准的内容推荐，与5000万程序员共同成长！

登录

注册



<https://passport.csdn.net/a>

望京soho



羊驼租赁

广告

General

Password

Organizations

Teams

Repositories

Accounts

2 GPG Signing >

API Key

Public Key

-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG v1.4.13 (Mingw32)

ABCDEFABCDEFABCDEFABCDEFABCDEFABCDEFABCDEFABCDEF
ABCDEFABCDEFABCDEFABCDEFABCDEFABCDEFABCDEFABCDEF
ABCDEFABCDEFABCDEFABCDEFABCDEFABCDEFABCDEFABCDEF
ABCDEFABCDEFABCDEFABCDEFABCDEFABCDEFABCDEFABCDEF
-----END PGP PUBLIC KEY BLOCK-----

Private Key

Update

Cancel

Clear GPG Key

点击Update保存这些key。

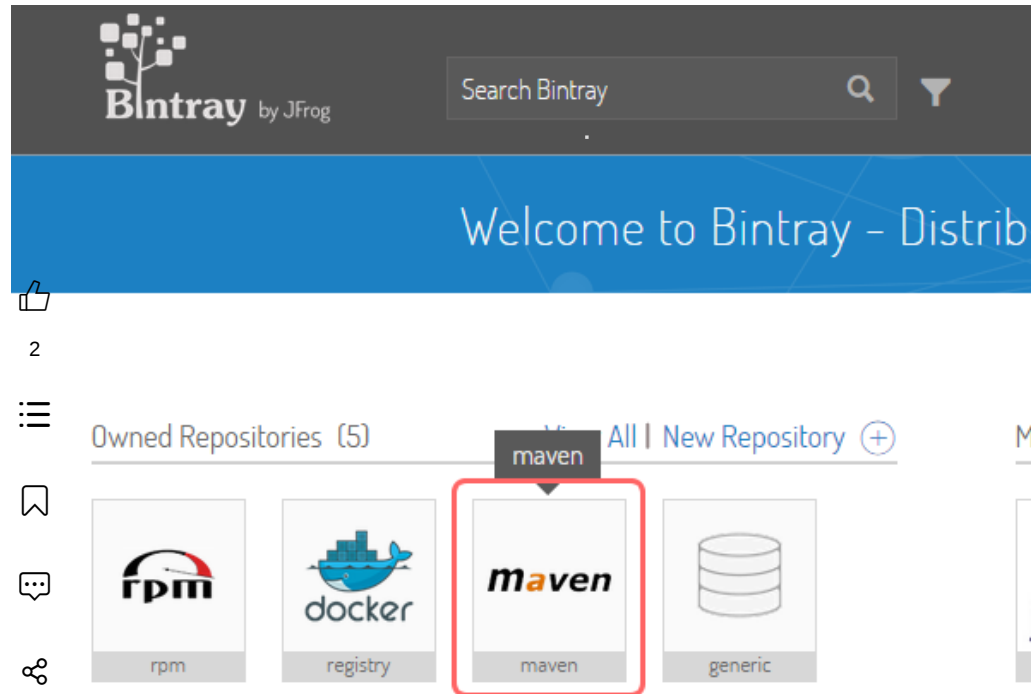
最后一步就是启用自动注册。到Bintray (<https://bintray.com/>)的主页点击maven。

加入CSDN，享受更精准的内容推荐，与5000万程序员共同成长！

[登录](#)[注册](#)

内容举报

返回顶部



点击编辑



加入CSDN，享受更精准的内容推荐，与5000万程序员共同成长！

(<https://passport.csdn.net/a>



望京soho



羊驼租赁

广告



内容举报



返回顶部

登录

注册



勾选中GPG Sign uploaded files automatically以启用自动注册。

Tags

Enter comma-separated description tags

☐ GPG sign uploaded files using Bintray's public /private key pair.

☒ GPG Sign uploaded files automatically

Note: Automatic content signing will not run since your private key is passphrase-protected. You can sign files by using the REST API, providing your key passphrase as input.

Update

Cancel

Delete Repository

点击Update保存这些步骤。完成。现在只需点击一下，每个上传到我们Maven仓库的东西都会自动注册并做好转向Maven Central。

请注意这是一次性的操作，以后创建的每一个library都要应用此操作。

Bintray和Maven Central 已经准备好了。现在转到Android Studio部分。

第四部分：准备一个Android Studio项目

很多情况下，我们需要同时上传一个以上的library到仓库，也可能不需要上传东西。因此我建议最好将每部分分成一个Module。最好分成两个module，一个Application Module一个Library Module。Application Module用于展示库的用法，Library Module是library的源代码。如果你的项目有一个以上的library，尽管创建另外的module：1个 module对应1 个library。

加入CSDN，享受更精准的内容推荐，与5000万程序员共同成长！

(<https://passport.csdn.net/a>



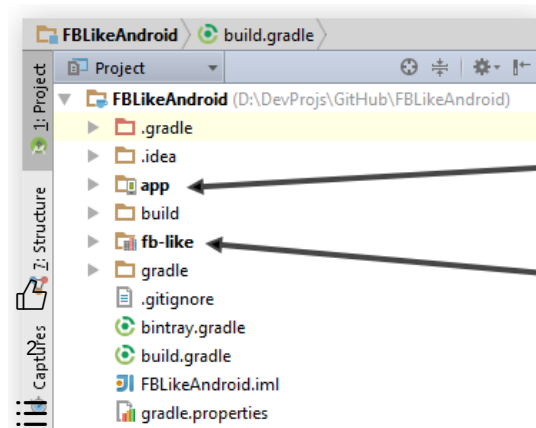
内容举报

返回顶部

登录

注册





Application Module

```
apply plugin: 'com.android.application'
```

(Will not be uploaded to repository)

Library Module

```
apply plugin: 'com.android.library'
```

(Will be uploaded to repository,
1 module = 1 package)

我相信大家知道如何创建一个新的module，因此就不会深入讲解这个问题了。其实很简单，基本就是选择 **File -> New -> Module -> New module**，然后选择 **creating an Android Library module**，然后就完了。



(<https://passport.csdn.net/a>



望京soho



羊驼租赁

广告



内容举报

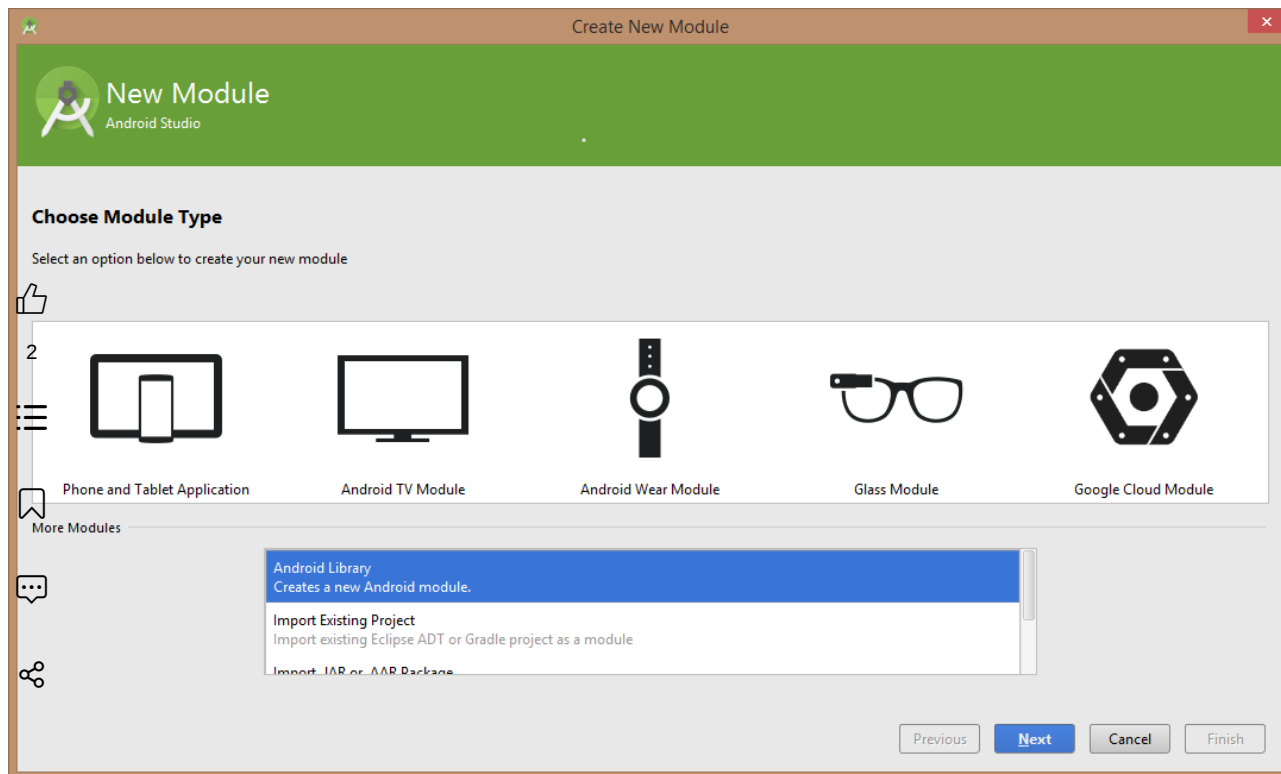
[返回顶部](#)

加入CSDN，享受更精准的内容推荐，与5000万程序员共同成长！

登录

注册





下一步是把bintray插件应用在项目中。我们需要修改项目的build.gradle文件中的依赖部分，如下：

```
dependencies {  
    classpath 'com.android.tools.build:gradle:1.2.3'  
    classpath 'com.jfrog.bintray.gradle:gradle-bintray-plugin:1.2'  
    classpath 'com.github.dcendents:android-maven-plugin:1.2'  
}
```

有一点非常重要，那就是gradle build tools的版本设置成1.1.2以上，因为以前的版本有严重的bug，我们将使用的是最新的版本1.2.3。

接下来我们将修改local.properties。在里面定义api key的用户名以及被创建key的密码，用于bintray的认证。之所以要把这些东西放在这个文件是因为这些信息时比较敏感的，不应该到处分享，包括版本控制里面。幸运的是在创建项目的时候local.properties文件就已经被添加到.gitignore了。因此这些敏感数据不会加入代码库，从而保证内容推荐，与5000万程序员共同成长！

(<https://passport.csdn.net/a>



望京soho



羊驼租赁

广告



内容举报



返回顶部

登录

注册



下面是要添加的三行代码：


```
bintray.user=YOUR_BINTRAY_USERNAME
bintray.apikey=YOUR_BINTRAY_API_KEY
bintray.gpg.password=YOUR_GPG_PASSWORD
```

bintray username 放在第一行，API Key放在第二行，API Key可以在Edit Profile

(<https://bintray.com/profile/edit>)页面的API Key 选项卡中找到。

最后一行是创建 GPG key的密码。保存并关闭这个文件。

最后要修改的是module的build.gradle文件。注意前面修改的是项目的build.gradle文件。打开它，在apply plugin: 'com.android.library'之后添加这几行，如下：

 复制代码

```
apply plugin: 'com.android.library'

ext {
    bintrayRepo = 'maven'
    bintrayName = 'fb-like'

    publishedGroupId = 'com.inthecheesefactory.thecheeselibrary'
    libraryName = 'FBLike'
    artifact = 'fb-like'

    libraryDescription = 'A wrapper for Facebook Native Like Button (LikeView) on Android'

    siteUrl = 'https://github.com/nuuneoi/FBLikeAndroid'
    gitUrl = 'https://github.com/nuuneoi/FBLikeAndroid.git'

    libraryVersion = '0.9.3'

    developerId = 'nuuneoi'
    developerName = 'Sittiphol Phanvilai'
    developerEmail = 'sittiphol@gmail.com'

    licenseName = 'The Apache Software License, Version 2.0'
    licenseUrl = 'http://www.apache.org/licenses/LICENSE-2.0.txt'
    allLicenses = ["Apache-2.0"]
}
```

加入CSDN，享受更精准的内容推荐，与5000万程序员共同成长！

(<https://passport.csdn.net/a>



望京soho



羊驼租赁

广告



内容举报

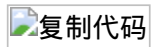


返回顶部

登录

注册





bintrayRepo使用默认的，即maven。bintrayName修改成你上面创建的 package name。其余的项也修改成和你library信息相匹配的值。有了上面的脚本，每个人都能通过下面的一行gradle脚本使用这个library。

```
compile 'com.inthecheesefactory.thecheeselibrary:fb-like:0.9.3'
```



最后在文件的后面追加两行如下的代码来应用两个脚本，用于构建library文件和上传文件到bintray（为了方便，我直接使用了github上连接到相关文件的链接）：

```
apply from: 'https://raw.githubusercontent.com/nuuneoi/JCenter/master/installv1.gradle'
apply from: 'https://raw.githubusercontent.com/nuuneoi/JCenter/master/bintrayv1.gradle'
```



完成！你的项目现在设置好了，准备上传到bintray吧！



第五部分：把library上传到你的bintray空间

现在是上传library到你自己的bintray仓库上的时候了。请到Android Studio的终端（Terminal）选项卡。

(<https://passport.csdn.net/a>



望京soho



羊驼租赁

广告



内容举报



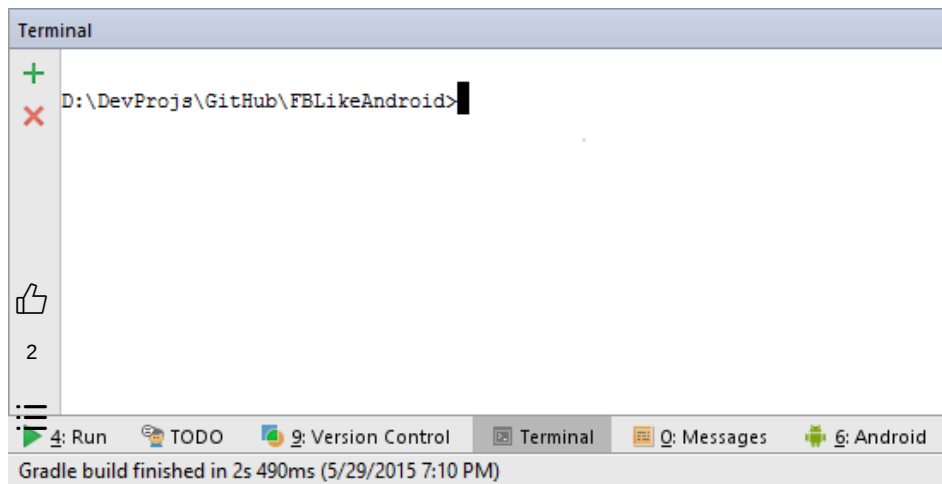
返回顶部

加入CSDN，享受更精准的内容推荐，与5000万程序员共同成长！

登录

注册





第一步是检查代码的正确性，以及编译library文件（aar，pom等等），输入下面的命令：

```
gradlew install
```



如果没有什么问题，会显示：

```
BUILD SUCCESSFUL
```

现在我们已经成功一半了。下一步是上传编译的文件到bintray，使用如下的命令：

```
gradlew bintrayUpload
```

如果显示如下你就大喊一声eureka吧！

```
SUCCESSFUL
```

在bintray的网页上检查一下你的package。你会发现在版本区域的变化。

加入CSDN，享受更精准的内容推荐，与5000万程序员共同成长！

登录

注册



(<https://passport.csdn.net/a>



望京soho



羊驼租赁

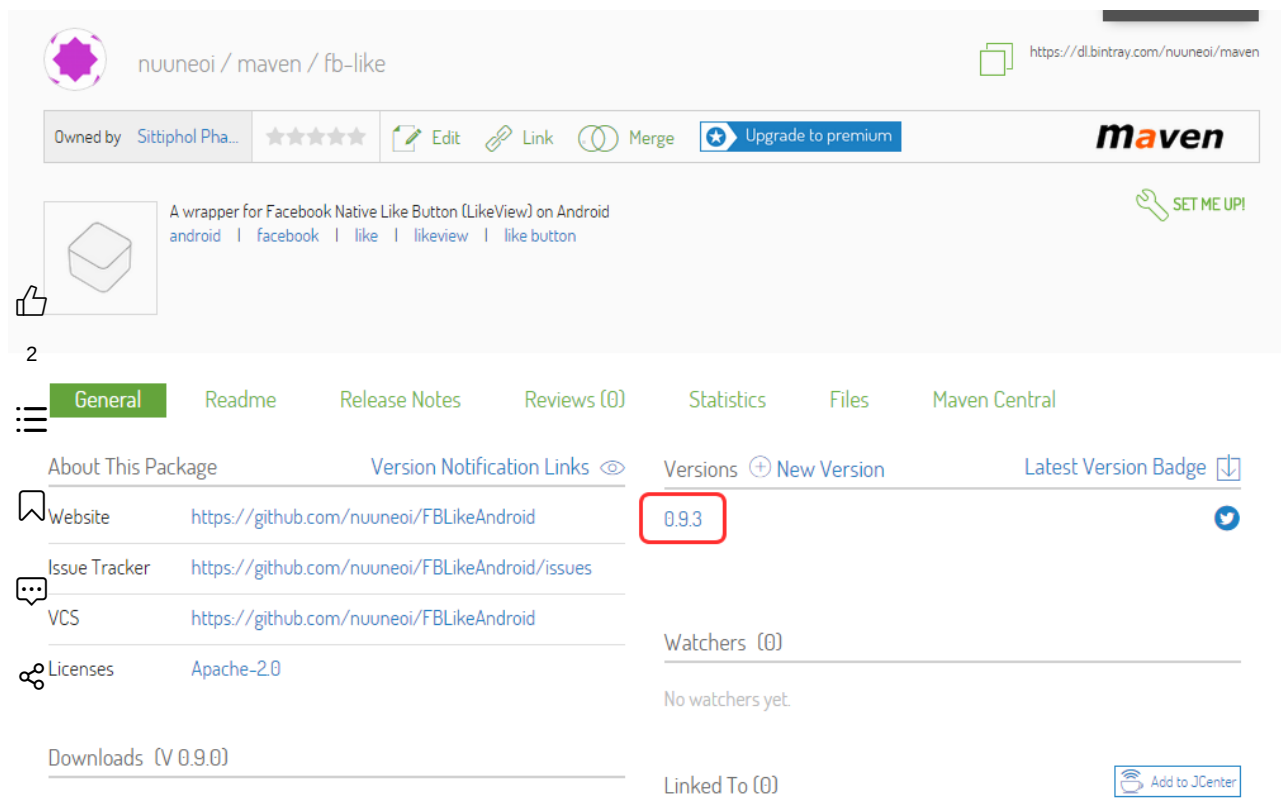
广告



内容举报



返回顶部



nuuneoi / maven / fb-like

Owned by Sittiphon Pha... ★★★★★ Edit Link Merge Upgrade to premium

A wrapper for Facebook Native Like Button (LikeView) on Android
android | facebook | like | likeview | like button

2

General Readme Release Notes Reviews (0) Statistics Files Maven Central

About This Package Version Notification Links Versions + New Version Latest Version Badge

Website <https://github.com/nuuneoi/FBLikeAndroid> 0.9.3

Issue Tracker <https://github.com/nuuneoi/FBLikeAndroid/issues>

VCS <https://github.com/nuuneoi/FBLikeAndroid>

Licenses Apache-2.0

Downloads (V 0.9.0)

Watches (0)
No watchers yet.

Linked To (0) Add to JCenter

点击进去，进入Files选项卡，你会看见那里有我们所上传的library文件。

(<https://passport.csdn.net/a>



望京soho



羊驼租赁

广告

内容举报

返回顶部

加入CSDN，享受更精准的内容推荐，与5000万程序员共同成长！

登录

注册



General Readme Release Notes Reviews (0) Statistics **Files** Maven Central

/ com / inthecheesefactory / thecheeselibrary / fb-like / 0.9.3

Name	Updated	Size
..	-	-
fb-like-0.9.3-javadoc.jar	1 month ago	50.9 KB
fb-like-0.9.3-javadoc.jar.asc	1 month ago	475 B
fb-like-0.9.3-sources.jar	1 month ago	4.3 KB
fb-like-0.9.3-sources.jar.asc	1 month ago	475 B
fb-like-0.9.3.aar	1 month ago	11.8 KB
fb-like-0.9.3.aar.asc	1 month ago	475 B
fb-like-0.9.3.pom	1 month ago	1.6 KB
fb-like-0.9.3.pom.asc	1 month ago	475 B

(https://passport.csdn.net/a



望京soho



羊驼租赁

广告

恭喜，你的library终于放在了互联网上，任何人都可以使用了！

不过也别高兴过头，library现在仍然只是在你自己的Maven仓库，而不是在jcenter上。如果有人想使用你的library，他必须定义仓库的url，如下：

复制代码

```
repositories {
    maven {
        url 'https://dl.bintray.com/nuuneoi/maven/'
    }
}

...

dependencies {
    compile 'com.inthecheesefactory.thecheeselibrary:fb-like:0.9.3'
```

加入CSDN，享受更精准的内容推荐，与5000万程序员共同成长！

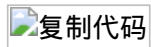
内容举报

返回顶部

登录

注册





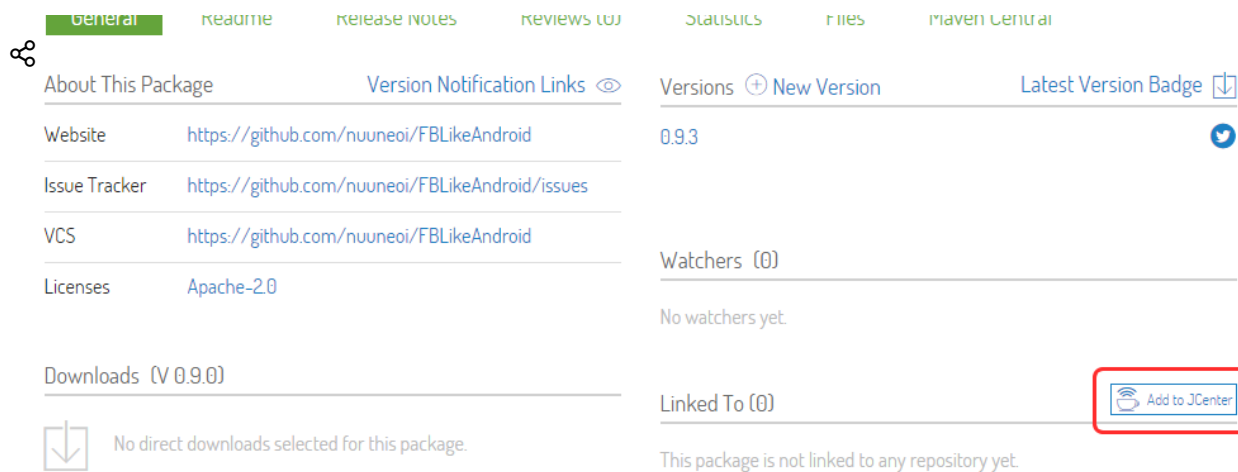
译者注：前面都没怎么看懂，看到上面的代码之后一下子全懂了，呵呵。

你可以在bintray的web界面找到自己Maven仓库的url，或者直接吧nuuneoi替换成你的bintray用户名（因为前面部分其实都是一样的）。我还建议你直接访问那个链接，看看里面到底是什么。

但是，就如我们前面所讲的那样，让开发者去定义url这种复杂的事情并不是分享library的最佳方式。想象一下，使用10个library不得添加10个url？所以为了更好的体验，我们把library从自己的仓库传到jcenter上。

第六部分：同步bintray用户仓库到jcenter

把library同步到jcenter非常容易。只需访问网页并点击Add to JCenter



什么也不做直接点击Send。

加入CSDN，享受更精准的内容推荐，与5000万程序员共同成长！

登录

注册



(https://passport.csdn.net/a



望京soho



羊驼租赁

广告



内容举报



返回顶部

(https://passport.csdn.net/a

Request to include the package 'fb-like' in 'jcenter'

☐ Host my **snapshot** build artifacts on the OSS Artifactory at<https://oss.jfrog.org>

This allows you to have your project's builds snapshots deployed to
<https://oss.jfrog.org> and to release and publish them to Bintray in one click.



2



Send



Cancel



现在我们所能做的就是等待bintray团队审核我们的请求，大概2-3个小时。一旦同步的请求审核通过，你会收到一封确认此更改的邮件。现在我们去网页上确认，你会在 Linked To 部分看到一些变化。

Linked To (1)

[Stage snapshots on oss.jfrog.org](#)

从此之后，任何开发者都可以使用jcenter() repository 外加一行gradle脚本来使用我们的library了

加入CSDN，享受更精准的内容推荐，与5000万程序员共同成长！
compile 'com.inthecheesefactory.thecheeselibrary:fb-like:0.9.3'



望京soho



羊驼租赁

广告



内容举报



返回顶部

登录

注册



想检查一下自己的library在jcenter上是否存在？你可以直接访问<http://jcenter.bintray.com> (<http://jcenter.bintray.com/>)，然后进入和你library的group id 以及artifact id匹配的目录。在本例中就是 com -> inthecheesefactory -> thecheeselibrary -> fb-like -> 0.9.3。



请注意链接到jcenter是一个只需做一次的操作。如果你对你的package做了任何修改，比如上传了一个新版本的binary，删除了旧版本的binary等等，这些改变也会影响到jcenter。不过毕竟你自己的仓库和jcenter在不同的地方，所以需要等待2 - 3分钟让jcenter同步这些修改。

同时注意，如果你决定删除整个package，放在jcenter仓库上的library不会被删除。它们会像僵尸一样的存在，没有人再能删除它了。因此我建议，如果你想删除整个package，请在移除package之前先在网页上删除每一个版本。

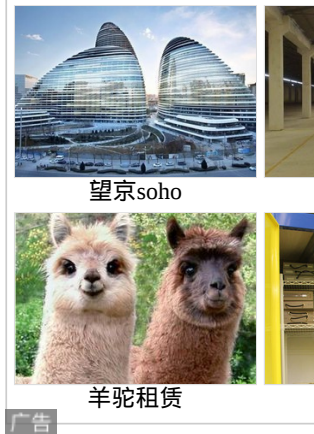
第七部分：上传library到Maven Central

并不是每个安卓开发者都使用jcenter。仍然有部分开发者还在使用mavenCentral()，因此让我们也把library上传到Maven Central 吧。

要从jcenter到Maven Central，首先需要完成两个任务：

加入CSDN，享受更精准的内容推荐，与5000万程序员共同成长！
1) Bintray package 已经连接到jcenter。

(<https://passport.csdn.net/a>



内容举报



返回顶部

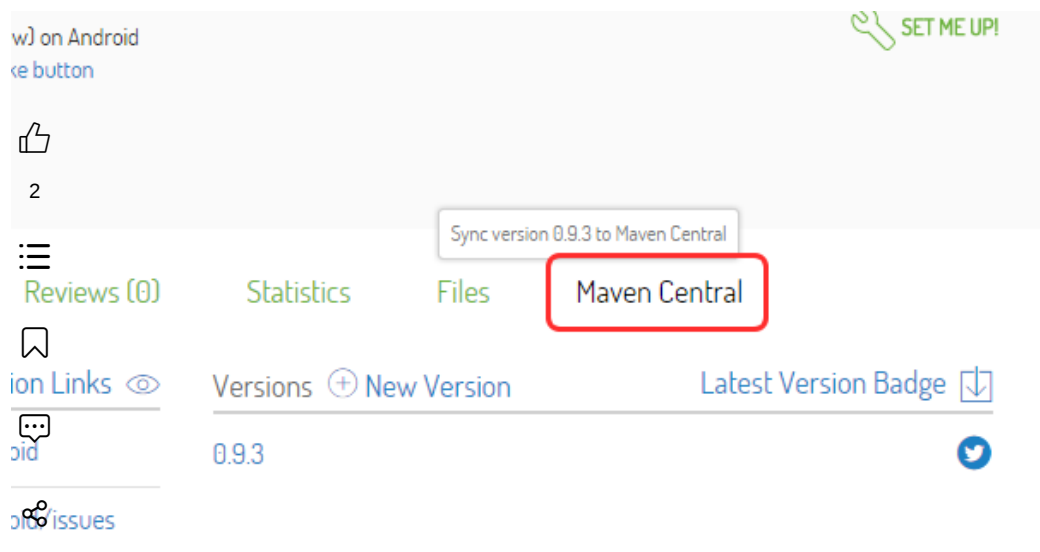
登录

注册



2) Maven Central上的仓库已经认证通过

如果你已经通过了这些授权，上传library package到Maven Central就异常简单了，只需在package的详情页面点击Maven Central 的链接。



输入你的Sonatype用户名和密码并点击Sync。

GeneralReadmeRelease NotesReviews (0)StatisticsFilesMaven Central

ⓘ To have your package artifacts synced to the Maven Central repository, your package must comply with the [Maven Central requirements](#).

User token key:

nuuneoi

User token password:

ⓘ We do not save this password.

☒ Close and release repository when done.

Sync

Sync Status

Last Synced: Mon May 25 17:08:32 UTC 2015

Last Sync Status: Successfully synced and closed repo.

Last Sync Errors: N/A

Note: The responsiveness of the Maven Central gateway at Sonatype fluctuates. This operation may take several minutes depending on the weather.

(<https://passport.csdn.net/a>

加入CSDN，享受更精准的内容推荐，与5000万程序员共同成长！

登录

注册

⚠
内容举报

⬆
返回顶部

<http://blog.csdn.net/jinyp/article/details/55095310>

28/36

如果成功，在Last Sync Status中会显示Successfully synced and closed repo（见图），但是如果遇到任何问题，则会在Last Sync Errors显示出来。你需要根据情况修复问题，能上传到Maven Central 的library的条件是相当严格的，比如+ 号是不能在ibrary版本的依赖定义中使用的。
完成之后，你可以在 Maven Central Repository
(<https://oss.sonatype.org/content/repositories/releases/>) 上找到你的library。在那些匹配你ibrary的group id以及artifact id的目录中。比如本例中就是com -> inthecheesefactory -> thecheeselibrary -> fb-like -> 0.9.3。

恭喜！虽然需要许多步骤，但是每一步都很简单。而且大部分操作都是一劳永逸的。

2



AndrExpert (/AndrExpert) 2017-11-02 16:37

3楼

(/AndrExpert) 很详细不错

回复



qq_30742989 (/qq_30742989) 2017-09-29 11:38

2楼

(/qq_30742989) 从桌面拖aar到这个Jcenter,我点击add to Jcenter按钮时为什么不能上传，需要pom文件

回复



itlavn (/itlavn) 2017-04-18 19:12

1楼

(/itlavn) 管用,多谢,不过库的版本需要更新了:

加入CSDN，享受更精准的内容推荐，与5000万程序员共同成长！
classpath 'com.github.dcendents:android-maven-gradle-plugin:1.5'

(<https://passport.csdn.net/a>



望京soho



羊驼租赁

广告



内容举报



返回顶部

登录

注册



回复

相关文章推荐

AndroidStudio关联Maven (<http://blog.csdn.net/u010680097/article/details/75039853>)



1.首先maven客户端可以安装也可以不安装 2.首先在引入的lib本地库中的gradle中顶部添加：apply plugin:'maven' //maven仓库上传插件然后：其中：然后...



u010680097 (<http://blog.csdn.net/u010680097>) 2017年07月12日 18:46 231



AndroidStudio 使用本地Maven仓库 (<http://blog.csdn.net/fyfcauc/article/details/70174960>)



Android项目直接使用本地aar。开发过程中有直接使用本地aar的调试需求，Android Gradle插件目前不支持直接以compile file的形式将aar引入(目前似乎只支持Jar)。...



fyfcauc (<http://blog.csdn.net/fyfcauc>) 2017年04月14日 18:17 3674



广告

惊呆了！微博和阿里背后的数据库有多厉害？

想不到！数据库作为最关键的基础设施，渗透技术领域的方方面面，我阿里和微博的师哥们是这么分享的...

(http://www.baidu.com/cb.php?c=lgF_pyfqHmknjTzrjb0IZ0qnfK9ujYzP1nsrjD10Aw-5Hc3rHnYnHb0TAq15HfLPWRznjb0T1YsPH99P1Rsn16zmhu-rAf30AwY5HDdnHn4rHbvnWD0lgF_5y9YIZ0IQzq-uZR8mLPbUB48ugfEpZNGXy-jULNzTvRETVNzpyN1gvw-IA7GUatLPjqdIAdxTvqdThP-5yF_UvTkn0KzujYk0AFV5H00TZcqn0KdpYfqHRLPjnvnfKEpyfqHc4rj6kP0KWpyfqP1cvrHnz0AqLUWYs0ZK45HcsP6KWThnqPj6krjn)

Android Studio之maven Central、JCenter (http://blog.csdn.net/lu_xin_/article/details/51...)

加入CSDN，享受更精准的内容推荐，与5000万程序员共同成长！

登录

注册



(<https://passport.csdn.net/a>



望京soho



羊驼租赁

广告




内容举报




返回顶部

Android studio 是从哪里得到库的？Android Studio是从build.gradle里面定义的Maven 仓库服务器上下载library的。Apache Maven是Apache开...

 lu_xin_ (http://blog.csdn.net/lu_xin_) 2016年04月12日 16:49 8227


AndroidStudio使用进阶二：搭建自己的maven私服，并使用Gradle统一依赖管理 ([http://blog...](http://blog.csdn.net/lu_xin_))

配置maven私服，使用gradle统一依赖管理，已项目演示的方式，将配置文件贴出，一看就懂。达到的目的：依赖关系在config.gradle中一目了然，需要修改时非常方便，使用maven私服更加灵活...

 JF_1994 (http://blog.csdn.net/JF_1994) 2016年04月23日 21:54 6290

android studio的maven私服使用配置踩坑实录 ([http://blog.csdn.net/cdecde111/article/deta...](http://blog.csdn.net/cdecde111/article/details/521174))

本文主要介绍在android studio上使用maven私服的实践过程及踩过的一些坑，相关工作...

 cdecde111 (<http://blog.csdn.net/cdecde111>) 2016年06月28日 16:52 2174



7
3.10/只
厂家直销ABS防水接线盒85x85x50mm 电线



8
5类非屏蔽网线
cat5e超五类网线八芯300米非屏蔽双绞线电



9
5.70/条
网线厂家屏蔽双绞线万兆网线高性能纯铜

Android Studio 引入第三方jar包和lib工程【一】-- 初识maven仓库 ([http://blog.csdn.net/A_...](http://blog.csdn.net/A_AsinCEO))

由于Studio比eclipse多了一步对工程构建的步骤，即为build.gradle这个文件运行，因此其引入第三方开发jar包与lib工程对比Eclipse已完成不同，引入第三方jar与lib工程显...

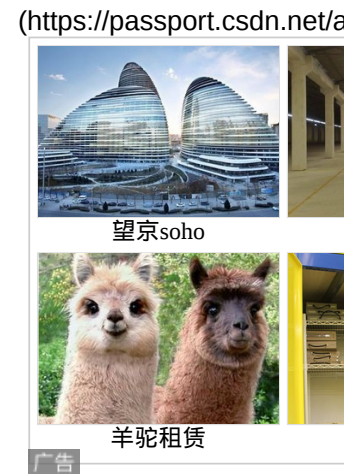
 A_AsinCEO (http://blog.csdn.net/A_AsinCEO) 2015年07月14日 11:20 28363

加入CSDN，享受更精准的内容推荐，与5000万程序员共同成长。[http://blog.csdn.net/u013325929/article/details/...](http://blog.csdn.net/u013325929/article/details/55095310)

登录

注册

×



内容举报

返回顶部

1.从File-project structure -dependencies -加号 -library搜索添加 或者 从github上找到这个项目的gradle-dependencies , 添加到主项...



u013325929 (<http://blog.csdn.net/u013325929>) 2015年09月11日 17:51 3607

30分钟搭建一个android的私有Maven仓库 (<http://blog.csdn.net/u012548313/article/details/...>)



30分钟搭建一个android的私有Maven仓库 原文链接 : A PRIVATE MAVEN REPOSITORY FOR ANDROID IN 30 MIN 原作者 : JERO...



u012548313 (<http://blog.csdn.net/u012548313>) 2015年09月10日 13:21 1429



Android Studio 配置多个代码仓库(maven) (<http://blog.csdn.net/uisoul/article/details/7748...>)



收集了一些。方便copy使用。repositories { mavenCentral() maven { url "https://jitpack.io" } maven { ...



uisoul (<http://blog.csdn.net/uisoul>) 2017年08月22日 16:58 1049

Android Studio 如何使用自定义(本地)maven仓库中的库 (<http://blog.csdn.net/fangziyi19911...>)

最近这两天一直在研究这个，网上找了很多别人写的博客，总是显得一头雾水，不知所措，可能我只是一个菜鸟，很多东西理解不了吧，不过我还是要把他分享出来，或许能够帮到一些人。我就不说那么多，不然这篇文章就跟别...



fangziyi199110 (<http://blog.csdn.net/fangziyi199110>) 2016年12月27日 17:22 929

JCenter下载太慢？教你修改Maven仓库地址为国内镜像 (<http://blog.csdn.net/wuqilianga/art...>)

解决方法一：https：改成 http协议下载 下文介绍部分来源自：https://my.oschina.net/lif/blog/754861 下面是build.gradle的内容...



wuqilianga (<http://blog.csdn.net/wuqilianga>) 2017年02月16日 16:38 4665

(<https://passport.csdn.net/a>)



望京soho



羊驼租赁

广告



内容举报



返回顶部

登录

注册



Android Studio 更换国内源下载依赖库 (<http://blog.csdn.net/coolwxb/article/details/508021...>)

我的博客：<http://daycoding.com> 小程序猿 由于国内GFW的原因，经常导致android studio 莫名其妙的编译不了，多数原因是由于不能下载依赖库 Gradle支持三种...



coolwxb (<http://blog.csdn.net/coolwxb>) 2016年03月04日 13:57 13367

2

Android Studio导入本地jar包，本地library库，maven库相关方法 (<http://blog.csdn.net/u01...>)

Android Studio导入本地jar包，本地library库，maven库相关方法导入本地jar包步骤 将jar拷贝到项目libs目录 Ctrl+Alt+Shift+s 进入project str...



u010859720 (<http://blog.csdn.net/u010859720>) 2015年08月22日 15:42 1226

AndroidStudio用maven导入项目步骤 (<http://blog.csdn.net/jackuhan/article/details/337398...>)

新鲜的工程，不要从eclipse中export 导入时候选择maven 可能会出现default activity不见，需要从原项目中copy AndroidManifest.xml 附...



jackuhan (<http://blog.csdn.net/jackuhan>) 2014年06月23日 16:46 2887

Android Studio Maven仓库使用 (<http://blog.csdn.net/ldld1717/article/details/52294509>)

Android studio Maven仓库使用 原文：How to distribute your own Android library through jCenter and Ma...



ldld1717 (<http://blog.csdn.net/ldld1717>) 2016年08月23日 21:14 1251

Android Studio发布项目到Maven仓库 (http://blog.csdn.net/H_Zhang/article/details/51558...)
加入CSDN，享受更精准的内容推荐，与5000万程序员共同成长！

(<https://passport.csdn.net/a>)



望京soho



羊驼租赁

广告



内容举报



返回顶部

登录

注册



为了方便别人使用我们开发的Android Library，一般我们都会把Android Library打成aar包，并将aar包发布到Maven仓库当中。如果是开源项目就可以把包发布到Maven Ce...



H_Zhang (http://blog.csdn.net/H_Zhang) 2016年06月01日 20:34 4179

android studio的maven私服使用 (<http://blog.csdn.net/cdecde111/article/details/51777427>)

本文主要介绍在android studio中使用artifactory进行maven配置方案的最终实现



cdecde111 (<http://blog.csdn.net/cdecde111>) 2016年06月28日 17:17 534



Android Gradle上传Maven仓库 (<http://blog.csdn.net/wei7017406/article/details/72084762>)

在Android的gradle配置上追加上Maven的相关操作 maven的gradle插件Android里内置了maven插件 只需要apply plugin: "maven" 即可使用mave...



wei7017406 (<http://blog.csdn.net/wei7017406>) 2017年05月14日 23:51 598

android studio远程仓库依赖 (<http://blog.csdn.net/lxxlxx888/article/details/56325493>)

远程仓库依赖 以LemonBubble4Android为例 你可以使用Gradle，需要首先在你的Project build.gradle文件中（allprojects -> repository...



lxxlxx888 (<http://blog.csdn.net/lxxlxx888>) 2017年02月21日 22:32 1020

Android Studio 中的maven仓库使用 (http://blog.csdn.net/Cui_Cui_666/article/details/7501...)

Android Studio是从哪里得到的库？



Cui_Cui_666 (http://blog.csdn.net/Cui_Cui_666) 2017年07月12日 14:32 488

加入CSDN，享受更精准的内容推荐，与5000万程序员共同成长！

(<https://passport.csdn.net/a>



望京soho



羊驼租赁

广告



内容举报



返回顶部

登录

注册



- 👍
- 2
- ☰
- 🔖
- 💬
- 🔗

(<https://passport.csdn.net/a>



望京soho



羊驼租赁

广告

- ⚠️
内容举报
- ⬆️
TOP
返回顶部

加入CSDN，享受更精准的内容推荐，与5000万程序员共同成长！

登录 注册 ✕

(<https://passport.csdn.net/a>