

需求：

1. 将一些公共基础库（例如日志，UI控件等）封装成library的方式，提供给有赞内部其他Android项目使用
2. 将一些业务封装成SDK的形式（也是library的形式), 提供给有赞内部其他Android项目使用

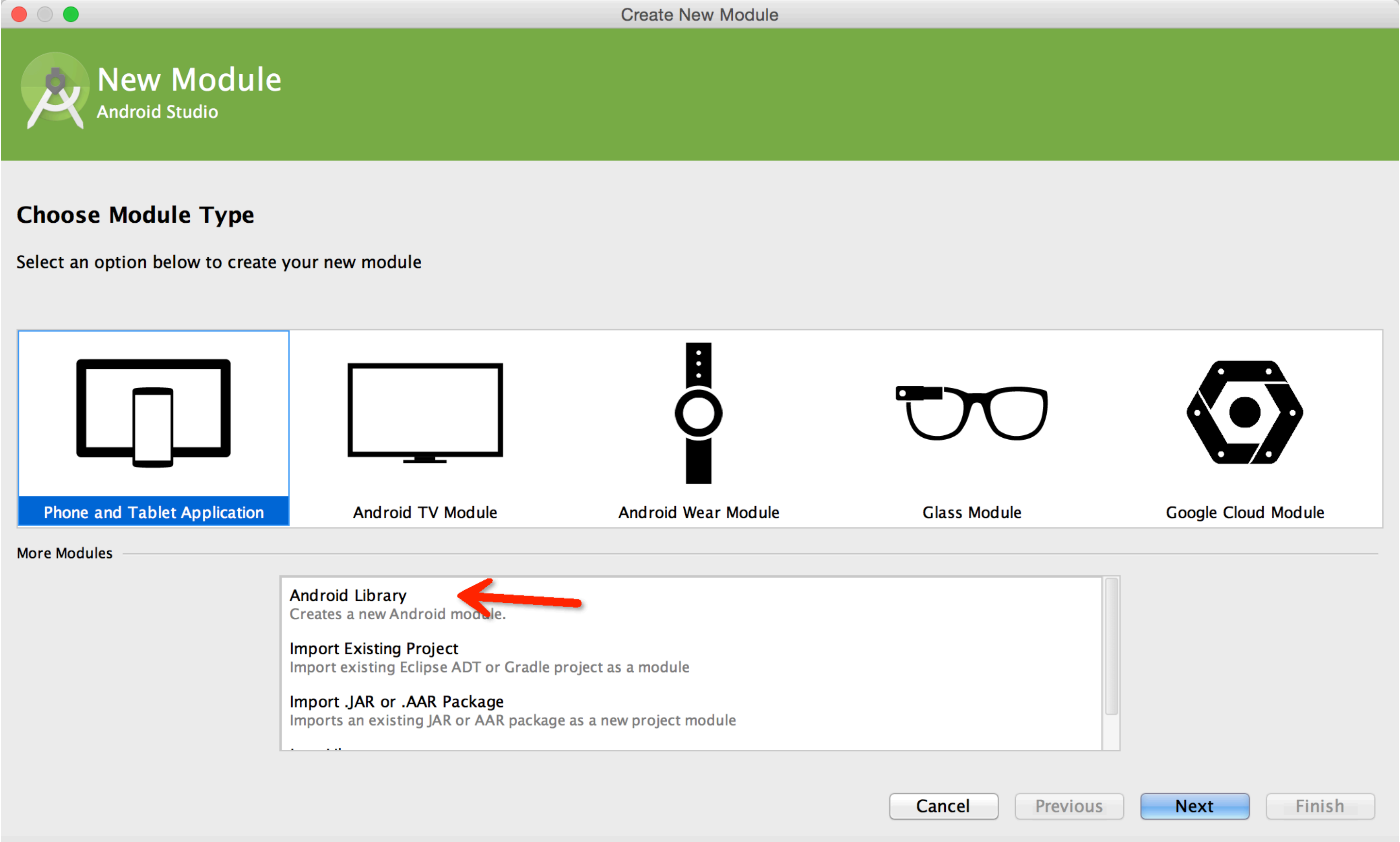
分析：

1. 将库打包
2. 将包上传到内部maven服务器
3. 使用方在build.gradle里配置内部服务器的maven仓库地址, 制定依赖的库，便可方便使用

分解：

将项目打包成库：

方式1：在Android Studio里，File->New Module->Android Library



方式2：将已有非Library的工程转换成库工程

修改工程的build.gradle, 将apply plugin: ‘com.android.application’ 修改成：
apply plugin: 'com.android.library'

Android库文件的后缀名：

.aar而非普通Java库的.jar格式
(android archive .aar) is not (java archive .jar)

. 将库上传到内部maven服务器

. maven服务器简单来说就是一个存储服务器，存储所有的libraries。既然只是一个存储服务器，没有业务逻辑，所以哪怕是本地文件系统，也可以当做maven服务器

这里介绍2种方式:

1. 将gitlab或者github这种SCM存放代码的地方用来做库的存储服务器（既然可以存放代码，当然也可以存放库这种格式的文件）

分析：

github上的任何项目(repo)， 要上传文件，必须在本机clone下来的文件夹内添加文件，然后git add/git commit/git push上传

可以参看我的github例子:

https://github.com/ericxu1983/AndroidPlayGround/tree/master/github_as_mvn_repo_example

因此方案如下：

1. 打包完成后， 将包“拷贝”到本地github文件夹
2. 在github文件夹内git add/git commit/git push， 上传到github服务器
3. 使用方添加github上库对应版本的url做为maven的额外仓库地址

. 打包上传如下， 在pom.project里制定pom.xml的一些信息（例如是哪个版本等）

```
apply plugin: 'maven'
uploadArchives {
    repositories {
        mavenDeployer {
            repository(url: 'file:///Users/ericxu/Documents/workspace/AndroidPL

            pom {
                project {
                    groupId 'com.example.ericxu'
                    version '1.0.2-snapshot'
                }
            }
        }
    }
}
```

然后在shell上调用：
gradle upload
进行上传

使用方如下方式使用：

```
repositories {
    maven {
        url 'https://raw.githubusercontent.com/ericxu1983/AndroidPlayGround/

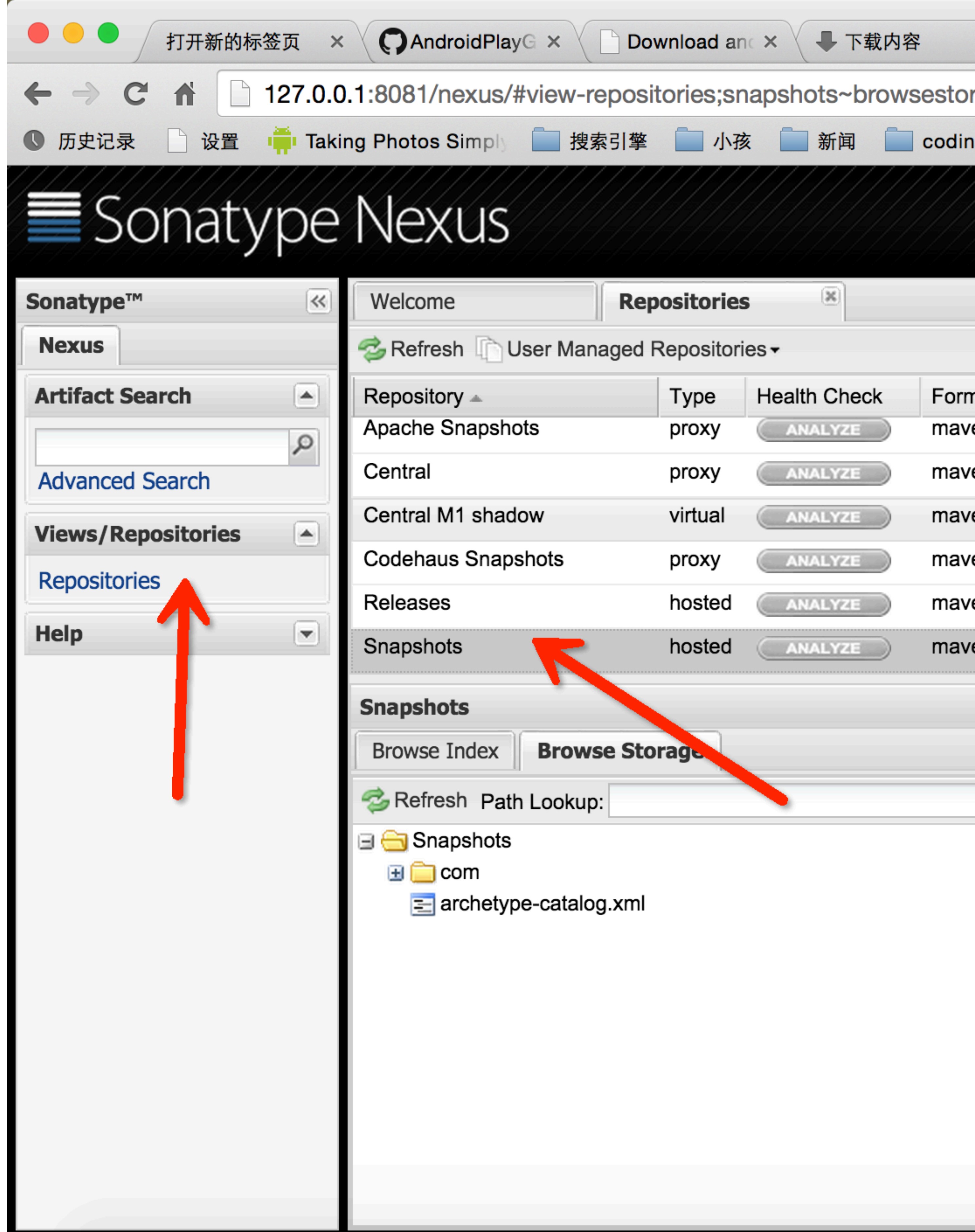
dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    compile 'com.android.support:appcompat-v7:22.0.0'
    compile 'com.example.ericxu:mylib2:1.0.2-snapshot@aar'
```

依赖的片

方式2：搭建真正的maven中央仓库server

- 1. 使用sonatype的nexus来搭建maven中央仓库，搭建方法非常简单
- 2. <http://www.sonatype.org/nexus/go/> 下载开源版本（OSS）的zip文件
- 3. 解压，在命令行上运行其中的bin/nexus便可以了
- 4. 访问127.0.0.1:8081，点击Repositories，可以查看所有仓库，假设这里我们上传的包是一个每日构建的非稳定版本的包（SNAPSHOT）

可以参看我的github例子：
https://github.com/ericxu1983/AndroidPlayGround/tree/master/upload_to_center_mvn_repo



mylib2/build.gradle - [mylib2] - MyLib - [~/Docu

MyLib mylib2 build.gradle

Project

Structure

MyLib (~ /Documents/workspace/AndroidPlayGrou

- .idea
- app
- build
- gradle
- mylib2
 - build
 - libs
 - src
 - .gitignore
- build.gradle
- mylib2.iml
- proguard-rules.pro
- .gitignore
- build.gradle
- gradle.properties
- gradlew
- gradlew.bat
- local.properties
- MyLib.iml
- settings.gradle
- External Libraries

Build Variants

Gradle Console

Favorites

TODO 6: Android Terminal

MainActivity.java x MyLib

```
}

apply plugin: 'maven'
uploadArchives {
    repositories {
        mavenDeployer {
            repository {
                authentic
            }
        }
    }
    pom {
        project
        group
    }
}

// fo
vers

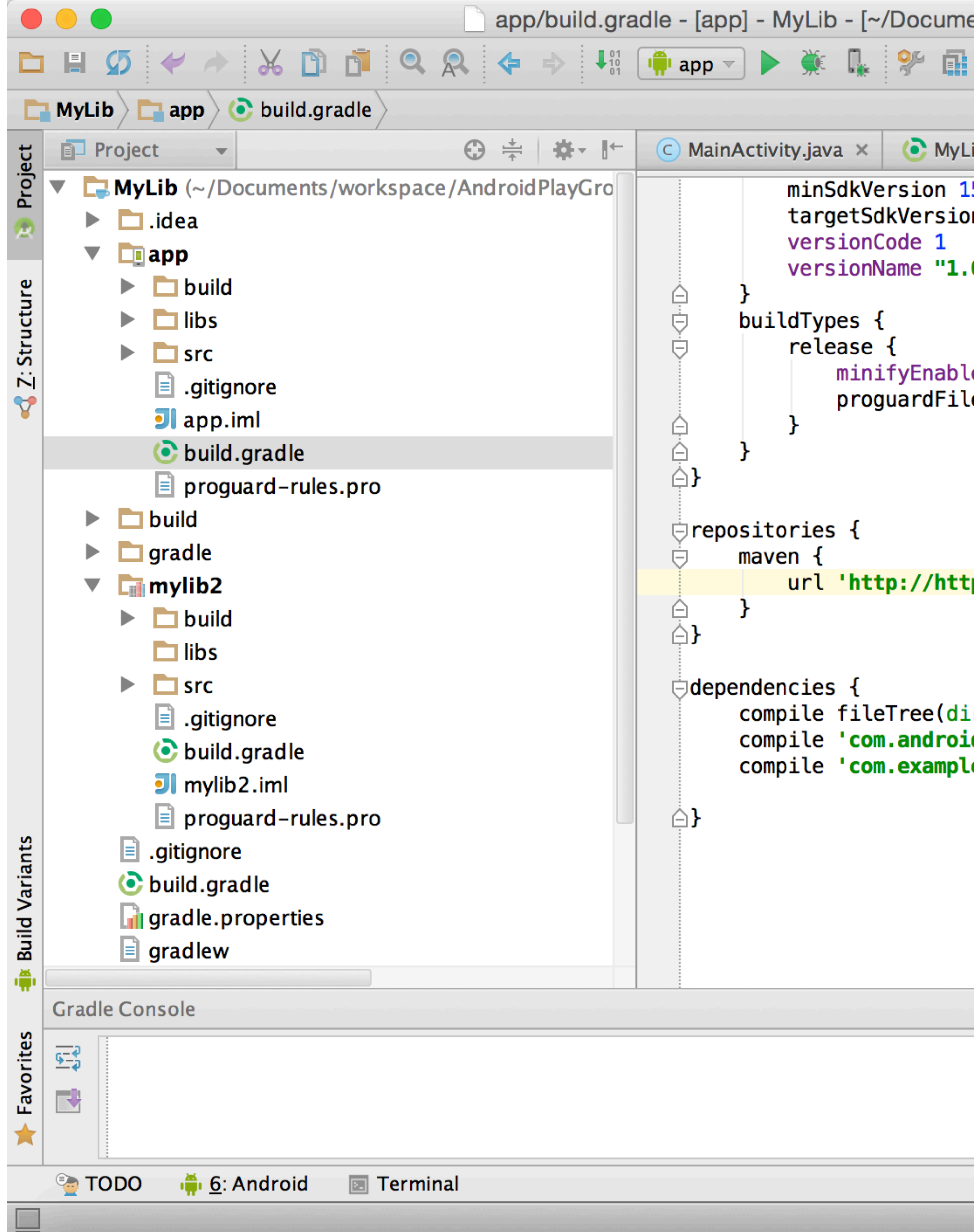
}

}

}

dependencies {
    compile fileTree(di
    compile 'com.android
```

使用方：



- 其他注意事项:
1. 每次修改库，必须更改版本号，否则使用方使用从使用方本地的cache中使用，不会去服务器上更新
 2. 如果是一个snapshot包，version name必须以-SNAPSHOT结尾