

码农明明桑

导入开源库到基于Android Studio构建的项目中

Dec 12, 2014

前两天，谷歌发布了Android Studio 1.0的正式版，也有更多的人开始迁移到Android Studio进行开发。然而，网上很多的开源库,控件等还是以前的基于Eclipse进行开发，很多人不知道怎么导入到自己的基于Android Studio项目中来，微博上也有人私信我，让我来写写，正好今天回来的比较早，就写写吧。主要介绍一下常见的一些导包的场景。

前言

```
--project    //项目目录
|
build.gradle //项目的gradle配置文件
|
settings.gradle // gradle设置，会保存所有的module
|
app          //module 目录
|__build.gradle module的配置
|
module2      //module2目录
|__build.gradle module的配置
```

JAVA

同eclipse中的项目一样，gradle/android studio 构建也可以有module，将moudle放到项目目录下面,然后在settings.gradle中增加该module，最简单的方法是用文件夹名。比如我们上面的结构，build.gradle文件应该如下：

```
include ':app', ':module2'
```

更多关于gradle的知识可以看我以前的文章:

使用gradle构建android项目（续）

使用Gradle构建Android项目

导入Jar文件

这种可能很常见，可以下载到别人搞好的jar包，这样可以直接在自己的主module下创建libs文件夹(我这里这样，只是为了兼容eclipse方式),然后把jar文件放进去,然后在module的 `build.gradle` 文件中的 `dependencies{}` 添加如下代码：

```
compile files('libs/name.jar')
```

当libs文件夹下面有多个文件时，可以用一句代码包含这些包:

```
compile fileTree(dir: 'libs', include: ['*.jar'])
```

当有文件不需要被包含时，可以这样:

```
compile fileTree(dir: 'libs', exclude: ['android-support*.jar'], include: ['*.jar'])
```

从上面的代码中可以看到我们可以使用通配符，`+` 表示一个字符，`*` 表示 0 到多个字符。

导入maven中的库

如果开源库作者有将代码放到Maven库中，我们可以在gradle配置中直接引入,类似如下:

```
compile 'com.github.dmytropanylyk.android-process-button:library:1.0.1'
```

一般我们可以在开源库的github页面上看有没有这样一个地址，或者到maven库中根据包名搜索有没有,我们前面这个引入的项目分三个部分 *group:name:version*，我们引入其他的包也有遵守这个规则。

导入gradle构建的开源库

这种情况的比较少用到，因为这张的开源库，作者一般都有放到maven库中，但是偶尔也会用到这里也提一下。

首先下载文件，将我们需要的这个库的module文件夹拷贝到我们的项目的目录下面，然后在setting.gradle文件中增加文件夹名称,然后在我们需要依赖这个模块的module中的build.gradle文件中加入如下代码:

```
compile project(':libmodule')
```

这样就可以了。

导入基于Eclipse构建的开源库

基于Eclipse构建的项目，和基于Android Studio构建的项目的很大区别是目录结构不同。

我们首先将module文件夹拷贝到我们的项目目录下面，然后在settings.gradle文件中增加这个module，然后在要使用的module中的build.gradle文件中引入依赖，这样看的话，似乎和引入基于gradle构建的没什么不同。但是，基于Eclipse构建的项目中，没有build.gradle文件，所以我们需要自己新建一个放到module下面,下面是一个模版：

```
apply plugin: 'android-library'

repositories {
    mavenCentral()
}

android {
    compileSdkVersion 19
    buildToolsVersion "20.0.0"
```

```
defaultConfig {
    minSdkVersion 9
    targetSdkVersion 19
}

sourceSets {
    main {
        manifest.srcFile 'AndroidManifest.xml'
        java.srcDirs = ['src']
        resources.srcDirs = ['src']
        aidl.srcDirs = ['src']
        res.srcDirs = ['res']
        assets.srcDirs = ['assets']
        jniLibs.srcDirs = ['libs']
    }
}

lintOptions {
    abortOnError false
}

dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
}
```

当然，根据各自的sdk和buildtools版本等等，以及其他，配置会有变化，可以看我一之前的文章。

其他

以上就是主要的集中导入场景，自己可以根据自己的实际情况然后改变配置等等。

另外,我们导入的仓库可能不是maven中心仓库，或者可能是我们自己搭建的仓库,我们可以自定义仓库地址的,修改build.gradle文件中的repositories就可以了，例如:

```
buildscript {  
    repositories {  
        jcenter()  
        mavenCentral()  
        maven {  
            url "https://oss.sonatype.org/content/repositories/snapshots"  
        }  
    }  
}
```

另外，project层的buildscript在module层也是会生效的，所以不用在每个module都配置。

原文地址：<http://blog.isming.me/2014/12/12/import-library-to-android-studio/>，转载请注明出处。

[PREV](#)[NEXT](#)