# Android 动态模块

# 介绍

动态功能交付与资产交付类似，同样是需要单独拆分出一个模块，而且这个模块与常规的功能模块存在很大的不同。我们通常添加一个模块，是希望这个模块中的代码能被多个项目复用，所以app模块与这个feature模块的引用关系为：app引用feature模块。

这种引用关系决定了app模块中可以随意使用feature模块中的代码和资源，而feature模块无法访问app中的资源和代码。

在动态功能交付中，这种引用关系需要反过来，feature模块引用app模块。所以feature中可以随意访问app中的代码和资源，而反过来则不行。

# 步骤

1. 新建一个普通的android项目
2. 新建一个Module，选择Dynamic Feature Module。Create New Module->Dynamic Feature Module
3. 在app下的build.gradle会自动引入

dynamicFeatures = [':features']

1. 在features的build.gradle文件中会自动引用app模块

dependencies **{** implementation fileTree(dir: "libs", include: ["\*.jar"])  
 implementation project(":app")  
 …  
**}**

1. 在AndroidManifest.xml已经引入了下面内容

<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:dist="http://schemas.android.com/apk/distribution"  
 package="com.example.features">  
  
 <dist:module  
 dist:instant="false"  
 dist:title="@string/title\_features">  
 <dist:delivery>  
 <dist:install-time />  
 </dist:delivery>  
 <dist:fusing dist:include="true" />  
 </dist:module>  
   
</manifest>

确认以上内容后，一个动态功能交付的模块就创建完成了。接下来要做的工作就是将原app模块中需要拆分的代码迁移到新创建的这个模块中，

并消除app模块对这部分代码的引用（即能成功编译）。在这个过程中可能会发现一个问题，那就是app模块无法引用动态模块中的代码和资源后，

那动态模块我要如何使用呢？无法引用其中的类，也就无法调用其中的功能方法，那动态分发的意义在哪里？

这个问题的答案也很简答：使用反射。

对于调用动态模块中的代码，使用反射的方式去调用对应类中的方法。对于跳转动态模块中的activity，使用Intent指定类名和包名即可，示例如下。

Intent intent = new Intent();  
intent.setClassName(getPackageName(), "com.example.features.TestActivity");  
startActivityForResult(intent, 1);

1. 生成aab并安装

生成aab

gradlew bundleRelease

合并apks

java -jar bundletool-all-1.7.0.jar build-apks --bundle=app-debug.aab --output=test1.apks  
java -jar bundletool-all-1.7.0.jar build-apks --bundle=app-release.aab --output=test1.apks

安装测试aab

java -jar bundletool-all-1.7.0.jar install-apks --apks=test1.apks

1. 使用bundletool 工具本地测试时，需要把features模块下的AndroidManifest.xml文件dist改为install-time，如果还是on-demand则安装后运行会提示找不到对应的类。
2. 命令行生成全量apk的方式

gradlew packageV\_test1DebugUniversalApk

# 参考文档

https://blog.csdn.net/github\_38688162/article/details/114849855

https://developer.android.google.cn/guide/app-bundle/play-feature-delivery

https://juejin.cn/post/7049522872682807333

https://www.jianshu.com/p/57cccc680bb6

https://blog.csdn.net/weixin\_36139431/article/details/125858326

https://blog.csdn.net/qq\_21118431/article/details/106267607

https://blog.csdn.net/wu996489865/article/details/118539853

https://hanshuliang.blog.csdn.net/article/details/118912401?spm=1001.2101.3001.6650.4&utm\_medium=distribute.pc\_relevant.none-task-blog-2%7Edefault%7ECTRLIST%7ERate-4-118912401-blog-118539853.pc\_relevant\_recovery\_v2&depth\_1-utm\_source=distribute.pc\_relevant.none-task-blog-2%7Edefault%7ECTRLIST%7ERate-4-118912401-blog-118539853.pc\_relevant\_recovery\_v2&utm\_relevant\_index=5

https://blog.csdn.net/chzphoenix/article/details/119186609?spm=1001.2101.3001.6650.11&utm\_medium=distribute.pc\_relevant.none-task-blog-2%7Edefault%7EBlogCommendFromBaidu%7ERate-11-119186609-blog-118539853.pc\_relevant\_recovery\_v2&depth\_1-utm\_source=distribute.pc\_relevant.none-task-blog-2%7Edefault%7EBlogCommendFromBaidu%7ERate-11-119186609-blog-118539853.pc\_relevant\_recovery\_v2&utm\_relevant\_index=12

https://developer.android.com/codelabs/on-demand-dynamic-delivery#1

命令行生成全量apk

https://gist.github.com/Farbklex/dbf96c2426c59a37d681438ade63e416

式样

微软雅黑 Light 小四

段落 固定值 23磅

# 目录1

## 目录2

### 目录3

#### 目录4

##### 目录5