

Android Library就是一个没有界面的应用程序，一般很少单独存在，一般我们是把经常用到的应用层的逻辑抽出来放在Library里面，当然一些常用的第三方的库也会采用这种方式。

## 打包jar

1. 新建一个Library，这个在studio里面很简单就可以做到。
2. 当逻辑写完之后，需要配置grade文件，代码如下：

```
task makeJar(type: Copy) {
    //删除存在的
    delete 'build/libs/mysdk.jar'
    //设置拷贝的文件
    from('build/intermediates/bundles/release/')
    //打进jar包后的文件目录
    into('build/libs/')
    //将classes.jar放入build/libs/目录下
    //include ,exclude参数来设置过滤
    //(我们只关心classes.jar这个文件)
    include('classes.jar')
    //重命名
    rename ('classes.jar', 'mysdk.jar')
}
```

1. 配置之后，在AndroidStudio中的Terminal中输入：

```
./gradlew makeJar
```

1. 完成之后，生成的jar包就会出现在libs路径下面了。
2. 引用jar包，添加如下代码：

```
repositories {
    flatDir {      //添加在android() 里面
        dirs 'libs'
    }
}

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

以上便可以开始使用jar包了，简单说一下jar包里面最好是不要有静态资源文件的，因为是访问不到的，如果要访问静态文件需要利用java的反射机制，来获取添加依赖项目的静态资源。当然，如果我们的Library里面有动态库(用c写的so文件)，也是访问不到的，这时我们可以采用aar文件。

## 打包aar文件

- 1.新建一个Library 2.Rebuild一下代码，就可以在Library -> build -> outputs -> aar -> xxx.aar 找到了 3.添加到想要依赖的项目的libs目录下 4.修改gradle代码

```
repositories {
    flatDir {      //添加在android() 里面
        dirs 'libs'
    }
}

compile (name:'DotEngine-debug', ext:'aar')
```

以上我们便可以使用了。