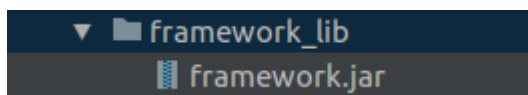


Eclipse 中添加系统 jar 包：

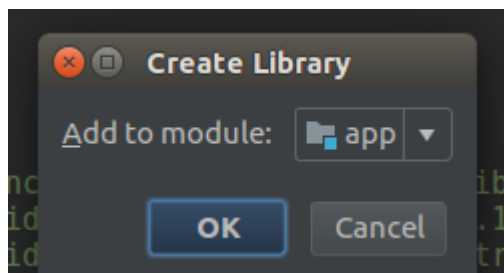
1. 右键项目名称-->properties
2. 点击 java build path 选项
3. 选择 libraries 标签页
4. 点击 Add External JARS 按钮
5. 选择存放 framework.jar 目录下的 framework.jar
6. 点击确定即可

Android Studio 中添加系统 jar 包：

1. 在 app 下新建一个目录 framework_lib
2. 将 framework.jar 包复制到 framework_lib 目录下



3. 右键 framework.jar 点击 Add As Library 然后选择默认的 app module



4. 添加成功后，在 module 下的 build.gradle 里的 dependencies 下多了一行 implementation files('framework_lib/framework.jar')

```
dependencies {  
    implementation fileTree(include: ['*.jar'], dir: 'libs')  
    implementation 'com.android.support:appcompat-v7:26.1.0'  
    implementation 'com.android.support.constraint:constraint-layout:1.0.2'  
    testImplementation 'junit:junit:4.12'  
    androidTestImplementation 'com.android.support.test:runner:1.0.1'  
    androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.1'  
    implementation files('framework_lib/framework.jar')  
}
```

5. 将 implementation files('framework_lib /framework.jar')更改为 provided files('framework_lib/framework.jar')

```
dependencies {
    implementation fileTree(include: ['*.jar'], dir: 'libs')
    implementation 'com.android.support:appcompat-v7:26.1.0'
    implementation 'com.android.support.constraint:constraint-layout:1.0.2'
    testImplementation 'junit:junit:4.12'
    androidTestImplementation 'com.android.support.test:runner:1.0.1'
    androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.1'
    // implementation files('framework_lib/framework.jar')
    provided files('framework_lib/framework.jar')
}
```

6. 在 project 的 build.gradle 里的 allprojects 下增加如下段（作用是将 framework.jar 优先 android.jar 编译）

```
gradle.projectsEvaluated {
    tasks.withType(JavaCompile) {
        options.compilerArgs.add('-Xbootclasspath/p:app/framework_lib/framework.jar')
    }
}
```

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

    gradle.projectsEvaluated {
        tasks.withType(JavaCompile) {
            options.compilerArgs.add('-Xbootclasspath/p:app/framework_lib/framework.jar')
        }
    }
}
```

注意：app/framework_lib/framework.jar 是相对路径，而且我这里是 linux 路径，如果是 windows 相对路径是 app\\framework_lib\\framework.jar

完成以上步骤，即可进行编译安装。

但是(IDCardManager)getSystemService(Context.IDCARD_SERVICE);此处 Context.IDCARD_SERVICE 仍会提示找不到

解决办法：

打开 app 下的 app.iml 文件（该文件是自动生成文件），将鼠标滑到最后

找到 `<orderEntry type="jdk" jdkName="Android API 26 Platform" jdkType="Android SDK" />`行，将这行移动到最后

```
<orderEntry type="library" scope="TEST" name="Gradle: net.sf.kxml:kxml2:2.3.0@jar" level="project" />
<orderEntry type="library" name="Gradle: android.arch.lifecycle:runtime-1.0.0" level="project" />
<orderEntry type="jdk" jdkName="Android API 26 Platform" jdkType="Android SDK" />
</component>
```

此时 Context.IDCARD_SERVICE 就不会报错。但是该更改会随着 sync gradle 变化而变化，这样手动更改，很容易自动还原,可以使用以下办法:

****如下是可以自动将<orderEntry type="jdk" jdkName="Android API 26 Platform" jdkType="Android SDK" />行移动到最后的脚本代码，将这段话复制到 app/build.gradle 文件根节点最后即可：**

```
preBuild {
    doLast {
        def imlFile = file(project.name + ".iml")
        println 'Change ' + project.name + '.iml order'
        try {
            def parsedXml = (new XmlParser()).parse(imlFile)
            def jdkNode = parsedXml.component[1].orderEntry.find { it.'@type' == 'jdk' }
            parsedXml.component[1].remove(jdkNode)
            def sdkString = "Android API " +
                android.compileSdkVersion.substring("android-".length()) + " Platform"
            new Node(parsedXml.component[1], 'orderEntry', ['type': 'jdk', 'jdkName':
                sdkString, 'jdkType': 'Android SDK'])
            groovy.xml.XmlUtil.serialize(parsedXml, new FileOutputStream(imlFile))
        } catch (FileNotFoundException e) {
            // nop, iml not found
        }
    }
}
```

```

dependencies {
    implementation fileTree(include: ['*.jar'], dir: 'libs')
    implementation 'com.android.support:appcompat-v7:26.1.0'
    implementation 'com.android.support.constraint:constraint-layout:1.0.2'
    testImplementation 'junit:junit:4.12'
    androidTestImplementation 'com.android.support.test:runner:1.0.1'
    androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.1'
    // implementation files('framework_lib/framework.jar')
    provided files('framework_lib/framework.jar')
}

preBuild {
    doLast {
        def imlFile = file(project.name + ".iml")
        println 'Change ' + project.name + '.iml order'
        try {
            def parsedXml = (new XmlParser()).parse(imlFile)
            def jdkNode = parsedXml.component[1].orderEntry.find { it.@type == 'jdk' }
            parsedXml.component[1].remove(jdkNode)
            def sdkString = "Android API " + android.compileSdkVersion.substring("android-".length()) + " Platform"
            new Node(parsedXml.component[1], 'orderEntry', ['type': 'jdk', 'jdkName': sdkString, 'jdkType': 'Android SDK'])
            groovy.xml.XmlUtil.serialize(parsedXml, new FileOutputStream(imlFile))
        } catch (FileNotFoundException e) {
            // nop, iml not found
        }
    }
}

```