

# 使用RePlugin时wiki上没说的一些坑

## 使用RePlugin时wiki上没说的一些坑

RePlugin的GitHub链接

作为一个新手学习新的东西,当然是看wiki;团队很友好wiki做的很详细  
于是打开wiki选择了快速上手,按照步骤发现并不顺利,  
在这里总结了两个简单的原因(也是Issues上提问比较多的)  
初次研究后续还有坑会继续补充...

### 坑(1):暂时支持的gradle版本

```
classpath 'com.android.tools.build:gradle:2.1.3'  
distributionUrl=https://services.gradle.org/distributions/gradle-2.14.1-all.zip
```

否则宿主App会出现空指针,插件app会出现不识别id问题

```
! Execution failed for task ':host:processDebugResources'.  
> java.lang.NullPointerException (no error message)
```

```
! Error:(54, 0) Failed to apply plugin [id 'replugin-plugin-gradle']  
Open File
```

### 坑(2):外置插件安装的正确姿势

栗子:在存储器中存放插件apk进行外置插件安装

- 按照官方实例自定义一个自己的插件app并打包,这里取名为 plugin01  
对应需要改造点地方是

```
<meta-data  
    android:name="com.qihoo360.plugin.name"  
    android:value="plugin01" />  
<meta-data  
    android:name="com.qihoo360.plugin.version.ver"  
    android:value="1" />
```

只需要更改value值

即可,详见 PluginInfo的parseFromPackageInfo 方法

- 将自定义的插件apk存放到想要加载的位置,这里保存路径为 /storage/emulated/0/plugin/plugin01.apk

```
private static final String PLUGIN_PATH =  
    Environment.getExternalStorageDirectory().getPath() + File.separator + "plugin";
```

```
private static final String PLUGIN_NAME = "plugin01";
private static final String PLUGIN_SUFFIX = ".apk";
```

- 在宿主app中配置Application类进行设置按照官方实例可以用两种方式

### 1.直接继承自 RePluginApplication

### 2.“非继承式”配置Application

这里说下第二种情况,需要注意的是

不允许在attachBaseContext调用完成之后再修改RePluginConfig对象中的内容

详见RePluginConfig的checkAllowModify方法

## 宿主app的attachBaseContext

这里需要设置 `setVerifySign`为false 不进行签名检查,回调事件函数可不写

```
@Override
protected void attachBaseContext(Context base) {
    super.attachBaseContext(base);
    mApp = this;
    RePluginConfig pluginConfig = new RePluginConfig();
    pluginConfig.setVerifySign(false)
        .setEventCallbacks(new RePluginEventCallbacks() {
            @Override
            public void onInstallPluginFailed(String path, InstallResult code) {
                Log.d(TAG, "onInstallPluginFailed() called with: path = [" + path + "], code = [" + code + "]");
            }

            @Override
            public void onInstallPluginSucceed(PluginInfo info) {
                Log.d(TAG, "onInstallPluginSucceed() called with: info = [" + info + "]);");
            }

            @Override
            public void onStartActivityCompleted(String plugin, String activity, boolean result) {
                Log.d(TAG, "onStartActivityCompleted() called with: plugin = [" + plugin + "], activity = [" + activity + "]);");
            }
        });
    // ===== REPLUGIN =====
    RePlugin.App.attachBaseContext(this, pluginConfig);
    // =====
}
```

## 安装插件

```
PluginInfo pluginInfo = RePlugin.install(PLUGIN_PATH + File.separator + PLUGIN_NAME + PLUGIN_SUFFIX);
```

执行完之后再控制台打印出日志 输出插件信息

```
D/SampleApplication: onInstallPluginSucceed() called with: info = [PInfo { <plugin01:2(4)> [APK] js={"pkgname":"com.kuangye.plugin.one"}]
D/MainActivity: pluginInfo = PInfo { <plugin01:2(4)> [APK] js={"pkgname":"com.kuangye.plugin.one","ali":"plugin01","name":"plugin01","l
```

同时会发现指定路径下的自定义插件apk消失,在 `data/data/包名/app_p_a` 下有一个jar包生成,这是因为在插件安装时,将文件“移动”到app\_p\_a目录下

可以在RePluginConfig中调用setMoveFileWhenInstalling方法进行修改;

这也就是大部分小白(like me)调用install方法时pluginInfo为null的原因之一,因为加载过后原有路径文件不存在了