APP 关机重启权限及静默安装权限说明

1. 背景目的:

由于部分权限需要系统 APP 才能获取，第 3 方 APP 无法调用此类接口，故为了方便客户开发，系统层开发新的接口给客户 APP 调用，实现关机重启及静默安装等功能。

2. APP 调用 API 方法

2.1 关机重启权限调用

App 可以通过反射机制来获取系统提供的 API，如下代码截图

```
private void setShundownOrReboot(String appName , Boolean isReboot){
    if (android.os.Build.VERSION.SDK_INT >= android.os.Build.VERSION_CODES.TIRAMISU) {
        try {
            Object iMinServiceManager = getSystemService( name: "iminservice");
            Class<?> c = Class.forName("android.app.iMinServiceManager");
            Method method = c.getMethod( name: "setShutdownOrRebootforApp", String.class, boolean.class);
            method.invoke(iMinServiceManager, appName, isReboot);
        } catch (Exception e) {
            e.printStackTrace();
        }
    } else {
        try {
            Class<?> c = Class.forName("android.app.ActivityManager");
            Method method = c.getMethod( name: "setShutdownOrRebootforApp", String.class, boolean.class);
            method.invoke(am, appName, isReboot);
            Log.d( tag: "iminSF",  msg: "setShundownOrReboot : " + appName);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

AppName: kit 平台授权 app 的包名， kit 平台-系统定制-动态权限-添加规则，添加 APP 及对应的权限

isReboot: ture 代表重启, false 代表关机

备注： c 是系统 api 所在的包名，不能写错；setShutdownOrRebootforApp 是 api 调用的方法,后面是带的参数，特别注意 boolean.class 里面的 boolean 不能大写，否则识别不了

2.2 静默安装权限调用

App 可以通过反射机制来获取系统提供的 API，如下代码截图

```java
private void setSilenceInstall(String apkName , String apkFilePath){
    if (android.os.Build.VERSION.SDK_INT >= android.os.Build.VERSION_CODES.TIRAMISU) {
        try {
            boolean isInstallPermission = getPackageManager().canRequestPackageInstalls();
            if(!isInstallPermission){
                //权限没有打开则提示用户去手动打开
                openInstallPermission();
            }
            Object iMinServiceManager = getSystemService( name: "iminservice");
            Class<?> c = Class.forName("android.app.iMinServiceManager");
            Method method = c.getMethod( name: "setSilenceInstallforApp",String.class,String.class);
            method.invoke(iMinServiceManager,apkName,apkFilePath);
        } catch (Exception e) {
            e.printStackTrace();
        }
    } else {
        try {
            Class<?> c = Class.forName("android.app.ActivityManager");
            Method method = c.getMethod( name: "setSilenceInstallforApp", String.class, String.class);
            method.invoke(am, apkName, apkFilePath);
            Log.d( tag: "iminSF",  msg: "setSilenceInstall : " + apkName);
        } catch (Exception e) {
            e.printStackTrace();
        }
```

apkName:　kit 平台授权 app 的包名，　kit 平台-系统定制-动态权限-添加规则，添加 APP 及对应的权限

apkFilePath:　需要静默安装 APK 的路径，比如 String apkFilePath = "/sdcard/test.apk";

备注：　c 是系统 api 所在的包名，不能写错；setSilenceInstallforApp 是 api 调用的方法,后面是带的参数.

3、关于在 Android13 上申请访问需要安装的 apk 权限

由于需要安装的 APK 存储在 sdcard 路径下，app 在 onCreate 需要先检查是否具有访问权限，如下代码所示

```
private void checkPermissions() {
    File apkFile = new File(apkPath);
    if (apkFile.canWrite() && apkFile.canRead()) {
        return;
    }
    if (Build.VERSION.SDK_INT >= 33) {
        Intent intent = new Intent(Settings.ACTION_MANAGE_APP_ALL_FILES_ACCESS_PERMISSION)
                .setData(Uri.parse("package:" + getPackageName()));
        startActivity(intent);
    } else if (Build.VERSION.SDK_INT >= 23) {
        int REQUEST_CODE_CONTACT = 101;
        String[] permissions = {Manifest.permission.WRITE_EXTERNAL_STORAGE,
                Manifest.permission.MANAGE_EXTERNAL_STORAGE,
                Manifest.permission.READ_EXTERNAL_STORAGE,
                Manifest.permission.MOUNT_UNMOUNT_FILESYSTEMS};
        //验证是否许可权限
        for (String str : permissions) {
            if (ActivityCompat.checkSelfPermission( context: this, str) != PackageManager.PERMISSION_GRANTED) {
                //申请权限
                this.requestPermissions(permissions, REQUEST_CODE_CONTACT);
                return;
            }
        }
    }
}
```

Android13 访问权限的方式较为特殊，需要跳转进行授权。