

Android 热修复框架: Sophix基本使用

版本号:2019/2/24-15:38

- Android 热修复框架: Sophix基本使用
 - 框架集成
 - 添加依赖
 - 添加SophixStubApplication
 - 修改AndroidManifest
 - 替换原Application为SophixStubApplication
 - 添加必备权限
 - 添加Sophix的APP ID、APP Secret、RSA密钥
 - 注册阿里云开发者
 - 完整 AndroidManifest.xml
 - 混淆配置
 - 基本使用
 - 查询并加载新补丁: queryAndLoadNewPatch
 - SophixPatchTool生成补丁: sophix-patch.jar
 - 生成并且下发补丁
 - 问题汇总
 - 参考资料

框架集成

添加依赖

1、build.gradle文件中添加热修复相关的依赖

根目录: build.gradle

```

buildscript {
    repositories {
        maven {
            url 'http://maven.aliyun.com/nexus/content/repositories/releases/'
        }
    }
}

allprojects {
    repositories {
        maven {
            url 'http://maven.aliyun.com/nexus/content/repositories/releases/'
        }
    }
}

```

module app: build.gradle

```

dependencies {
    // Sophix-阿里热修复
    api 'com.aliyun.ams:alicloud-android-hotfix:3.2.8'
}
// implementation、compile都可以，看实际情况。

```

添加SophixStubApplication

2、添加热修复需要的SophixStubApplication，不要添加任何内容，除非有特殊需要。

@SophixEntry(BaseApplication.class) 其中的Application就是原来自定义的Application(实现业务逻辑)。SophixStubApplication中绝对不要增加任何逻辑代码。

```

package com.hao.lib_base.base;

import android.content.Context;
import android.util.Log;

import com.taobao.sophix.PatchStatus;
import com.taobao.sophix.SophixApplication;
import com.taobao.sophix.SophixEntry;
import com.taobao.sophix.SophixManager;
import com.taobao.sophix.listener.PatchLoadStatusListener;

import androidx.annotation.Keep;

/**
 * Sophix入口类，专门用于初始化Sophix，不应包含任何业务逻辑。
 * 此类必须继承自SophixApplication，onCreate方法不需要实现。
 * 此类不应与项目中的其他类有任何互相调用的逻辑，必须完全做到隔离。
 * AndroidManifest中设置application为此类，而SophixEntry中设为原先Application类。
 * 注意原先Application里不需要再重复初始化Sophix，并且需要避免混淆原先Application类。
 * 如有其它自定义改造，请咨询官方后妥善处理。
 */
public class SophixStubApplication extends SophixApplication {
    private final String TAG = "SophixStubApplication";
    // 此处SophixEntry应指定真正的Application，并且保证RealApplicationStub类名不被混淆。
    @Keep
    @SophixEntry(BaseApplication.class)
    static class RealApplicationStub {}
    @Override
    protected void attachBaseContext(Context base) {
        super.attachBaseContext(base);
        // 如果需要使用MultiDex，需要在此处调用。
        // MultiDex.install(this);
        initSophix();
    }
    private void initSophix() {
        String appVersion = "0.0.0";
        try {
            appVersion = this.getPackageManager()
                .getPackageInfo(this.getPackageName(), 0)
                .versionName;
        } catch (Exception e) {
        }
        final SophixManager instance = SophixManager.getInstance();
        instance.setContext(this)
            .setAppVersion(appVersion)
            .setSecretMetaData(null, null, null)
            .setEnableDebug(true)
            .setEnableFullLog()
            .setPatchLoadStatusStub(new PatchLoadStatusListener() {
                @Override
                public void onLoad(final int mode, final int code, final String info, final
                    if (code == PatchStatus.CODE_LOAD_SUCCESS) {
                        Log.i(TAG, "sophix load patch success!");
                    } else if (code == PatchStatus.CODE_LOAD_RELAUNCH) {

```

```

        // 如果需要在后台重启，建议此处用SharedPreferences保存状态。
        Log.i(TAG, "sophix preload patch success. restart app to make effect");
    }
}
}).initialize();
}
}

```

修改AndroidManifest

替换原Application为SophixStubApplication

3、替换原Application为SophixStubApplication

1. 系统会先加载 SophixStubApplication，内部再加载我们自定义的Application
2. 要避免原Application被混淆，不然找不到！看后面的混淆部分。

```

<application
    android:name="xxx.SophixStubApplication">

    // xxx
</application>

```

添加必备权限

4、添加必备权限(高版本注意动态权限申请需要自己处理)

```

<!-- 网络权限 -->
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
<!-- 外部存储读权限，调试工具加载本地补丁需要 -->
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>

```

添加Sophix的APP ID、APP Secret、RSA密钥

5、在application中添加meta-data

```
<application
    android:name="com.hao.lib_base.base.SophixStubApplication"
    xxx>

    <meta-data
        android:name="com.taobao.android.hotfix.IDSECRET"
        android:value="2568....." />
    <meta-data
        android:name="com.taobao.android.hotfix.APPSECRET"
        android:value="b8c8d31de746b9a....." />
    <meta-data
        android:name="com.taobao.android.hotfix.RSASECRET"
        android:value="MIIEvQIBADANBgk....." />
</application>
```

注册阿里云开发者

6、要使用Sophix需要登录阿里云开发平台，开通热修复的功能。

1. [阿里云移动热修复官网](#)
2. 具体如何开通相关应用，自行百度。

完整 AndroidManifest.xml

7、修改AndroidManifest

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    package="com.hao.iday">

    <!-- 网络权限 -->
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
    <uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
    <!-- 外部存储读权限，调试工具加载本地补丁需要 -->
    <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>

    <application
        android:name="com.hao.lib_base.base.SophixStubApplication"
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme"
        tools:ignore="GoogleAppIndexingWarning">
        <activity
            android:name=".MainActivity"
            android:launchMode="singleTask">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>

        <meta-data
            android:name="com.taobao.android.hotfix.IDSECRET"
            android:value="25681463" />
        <meta-data
            android:name="com.taobao.android.hotfix.APPSECRET"
            android:value="b8c8d31de746b9a097a4027c093c07e7" />
        <meta-data
            android:name="com.taobao.android.hotfix.RSASECRET"
            android:value="MIIEvQIBADANBgkqhkiG9w0BAQEFAASCBKcwggSjAgEAAoIBAQCbtW80WQ/sF3TCEYOg
    </application>

</manifest>

```

混淆配置

8、进行混淆配置，文件是 proguard-rules

```
#基线包使用，生成mapping.txt
-printmapping mapping.txt
#生成的mapping.txt在app/build/outputs/mapping/release路径下，移动到/app路径下
#修复后的项目使用，保证混淆结果一致
#-applymapping mapping.txt
#hotfix
-keep class com.taobao.sophix.**{*;}
-keep class com.ta.utdid2.device.**{*;}
-dontwarn com.alibaba.sdk.android.utils.**
#防止inline
-dontoptimize

-keepclassmembers class com.my.pkg.MyRealApplication {
    public <init>();
}
# 如果不使用android.support.annotation.Keep则需加上此行
# -keep class com.my.pkg.SophixStubApplication$RealApplicationStub
```

基本使用

查询并加载新补丁: queryAndLoadNewPatch

1、在BaseApplication(自定义的)中进行补丁的获取和更新

```
public class BaseApplication extends Application {

    @Override
    public void onCreate() {
        super.onCreate();
        // 1. 查询并且加载补丁。queryAndLoadNewPatch不可放在attachBaseContext 中，否则无网络权限，
        SophixManager.getInstance().queryAndLoadNewPatch();
    }
}
```

SophixPatchTool生成补丁: sophix-patch.jar

2、使用阿里提供的SophixPatchTool工具，通过新旧apk差异生成补丁包。

1. 具体教程可以看参考资料的 1和2, [Sophix-阿里第三代非侵入式热修复](#)
2. SophixPatchTool工具在参考资料: 3-[Windows版本打包工具地址](#)

生成并且下发补丁

3、在阿里云开发者平台的[移动热修复-管理控制台](#)中可以

1. 移动热修复->补丁管理->上传补丁->发布

问题汇总

参考资料

1. [Sophix官方文档: 移动热修复-快速开始](#)
2. [Sophix-阿里第三代非侵入式热修复](#)
3. [Windows版本打包工具地址](#)
4. [调试工具地址](#)