

小米应用商店 检查更新SDK 集成指南

小米应用商店检查更新SDK(下文简称为『检查更新SDK』)帮助您的用户尽快获得最新版本的应用！轻松增加『自动检查更新』功能，通过小米应用商店的CDN服务，快速的把最新版应用推送给用户！

功能特点：

- 支持增量更新：检查更新SDK会自动识别用户版本，并提供diff包给用户，加速省流
- 一次发布全面更新：只要在小米应用商店发布新版，所有的用户都可以收到新版发布提醒。
- 标准Android SDK：无论用户是否使用MIUI，都可以收到更新提醒。
- 方便省心：无需考虑配置服务器，部署CDN等问题，小米应用商店免费提供。
- 独家支持：在MIUI系统上支持调用系统商店更新，升级成功率更高！

立即开始集成：

1. 在工程中加入检查更新SDK

添加jar包

下载最新版SDK的zip包并解压缩，将其中的libs 文件夹合并到本地工程libs子目录下。
Eclipse用户右键工程根目录，选择Properties -> Java Build Path -> Libraries，然后点击Add External JARs... 选择指向xiaomi_sdk.jar的路径，点击OK，即导入成功。

添加资源文件

将SDK提供的res文件夹拷入工程目录下，和工程本身res目录合并。
请注意：小米SDK提供的资源文件都以xiaomi_开头。

2. 在AndroidManifest.xml文件中添加SDK信息

添加检查更新需要的权限

打开AndroidManifest.xml，添加SDK需要的权限到<manifest>标签下：

```
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-permission android:name="com.xiaomi.market.sdk.UPDATE" />
```

添加Receiver

```
<receiver android:name="com.xiaomi.market.sdk.DownloadCompleteReceiver" >
    <intent-filter>
        <action android:name="android.intent.action.DOWNLOAD_COMPLETE" />
    </intent-filter>
</receiver>
```

说明：SDK调用系统下载管理器进行下载，需要注册receiver，在下载完成后进行增量包合成与安装

3. 集成SDK代码

在应用程序入口Activity里的OnCreate() 方法中调用

```
public void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    XiaomiUpdateAgent.update(this);  
}
```

API:

```
- public void update(Context context)
```

Done! 集成成功!

现在只要在小米应用商店发布应用，您的用户就会收到更新提醒了！

Tips:

1. 新版本App要在小米应用商店上架之后，才能检查到更新
2. 考虑到集成检查更新SDK的版本分发也需要时间，建议您尽早集成

Changelog:

- **2013-07-08** alpha 第一个内测版发布
- **2013-07-17** ver 0.1 第一个公测版本，修复了alpha版上自行处理回调可能拿不到更新的问题，改善扩展性。

下页还有常见问题FAQ :)

常见问题 FAQ:

如何测试集成是否成功？

检查更新SDK依赖于应用的VersionCode判断新版本。如果您已经有App在小米应用商店发布，可以这样测试：

1. 完成上面的集成操作
2. 制作一个APK包，这个APK的VersionCode低于小米应用商店的上架版本
3. 在网络畅通的手机上，安装上面制作的APK并启动

如果集成成功，启动App后应该可以检测到新版本。

如何设置仅在WiFi下检查更新？

默认是在有网络条件下即检查，可设置为只在WiFi情况下检查更新：

```
XiaomiUpdateAgent.setCheckUpdateOnlyWifi(true);
XiaomiUpdateAgent.update(this);
```

API:

```
- public void setCheckUpdateOnlyWifi(boolean isOnlyWifi)
    isOnlyWifi 布尔值 false(默认) 在有网络情况下就检查更新，true，只在WiFi下检查更新
```

我希望自己设计提醒样式，如何手动处理回调？

如果想自己处理更新可以按如下步骤，设置updateAutoPopup=false，SDK将停止弹框，实现更新监听接口，处理更新事件：

```
XiaomiUpdateAgent.setUpdateAutoPopup(false);
XiaomiUpdateAgent.setUpdateListener(new XiaomiUpdateListener() {

    @Override
    public void onUpdateReturned(int updateStatus, UpdateResponse updateInfo) {
        switch (updateStatus) {
            case UpdateStatus.STATUS_UPDATE:
                // 有更新， UpdateResponse为本次更新的详细信息
                break;
            case UpdateStatus.STATUS_NO_UPDATE:
                // 无更新， UpdateResponse为null
                break;
            case UpdateStatus.STATUS_NO_WIFI:
                // 设置了只在WiFi下更新，WiFi， UpdateResponse为null
                break;
            case UpdateStatus.STATUS_NO_NET:
                // 没有网络， UpdateResponse为null
                break;
            case UpdateStatus.STATUS_FAILED:
                // 检查更新与服务器通讯失败，可稍后再试， UpdateResponse为null
                break;
            case UpdateStatus.STATUS_LOCAL_APP_FAILED:
                // 检查更新获取本地安装应用信息失败， UpdateResponse为null
                break;
            default:
                break;
        }
    }
});
XiaomiUpdateAgent.update(this);
```

API:

```
- public void setUpdateAutoPopup(boolean autopopup)
    autopopup 布尔值 true(默认) 弹出更新对话框, false 禁止弹出更新对话框
- public void setUpdateListener(XiaomiUpdateListener listener)
    listener 更新回调接口
- public void onUpdateReturned(int statusCode, UpdateResponse updateInfo)
    statusCode 更新回调接口返回状态, 与UpdateStatus里的常量相对应; updateInfo 更新回调返回数据, 包括App更新信息、属性; updateLog 更新日志; versionCode 最新版本的版本号 versionName最新版本的版本名称; path 最新版下载链接
```

提醒: 默认弹框方式下, 点确定升级时, 在装有小米应用商店的情况下会调起应用商店里对应的该应用的详情界面, 在没有小米应用商店的情况, 会直接调用系统下载管理器进行下载。

检查更新SDK如何检查新版本?

当调用SDK的update()方法时, SDK会获取当前应用的包名, 版本号, 签名和当前版本的MD5, 上传至服务器, 并带上相应的机型信息, 由服务器进行匹配过滤, 并将更新信息和下载地址等返回给SDK来处理。

混淆中遇到问题怎么处理?

如果您的应用使用了混淆, 由于SDK需要引用导入工程的资源文件, 通过了反射机制得到资源引用文件R.java, 但是在开发者通过proguard等混淆/优化工具处理apk时, proguard可能会将R.java删除, 如果遇到这个问题, 请在proguard配置文件中添加keep命令如:

```
-keep public class [您的应用包名].R${*}{
    public static final int *;
}
```

请把[您的应用包名] 替换成您自己的包名, 如com.yourcompany.example。

检查更新SDK申请的权限对应哪些功能?

- android.permission.WRITE_EXTERNAL_STORAGE 允许将下载的apk文件保存到sd卡中
- android.permission.ACCESS_NETWORK_STATE
- android.permission.INTERNET
- android.permission.ACCESS_WIFI_STATE 以上三个权限允许SDK访问手机的网络连接状态, 判断是否更新
- android.permission.READ_PHONE_STATE 用于读取手机硬件信息等, 用于机型过滤
- com.xiaomi.market.sdk.UPDATE 用于标明使用小米SDK

有时候测试发现并不是增量升级包是为什么?

由于增量更新需要通过比较当前包和更新包的差异来产生DIFF文件, 如果一直检测不到增量更新, 有可能是服务器上没有当前包的备份, 无法生成DIFF文件。

增量更新有哪些要求?

- 只要用户安装的App版本, 曾经在小米应用商店发布过, 在下载更新时, SDK就会自动下载增量包更新。无论与最新版之间相隔多少个版本, 小米应用商店的服务器都会自动计算出合适的增量包。
- 提醒1: 增量更新只在由SDK负责下载更新时生效
- 提醒2: 因为增量包生成的触发, 是首次有用户请求该增量包时发生的, 因此, 新版本刚刚发布时, 会有极少部分用户下载到的仍然是完整包, 后面我们会优化增量包的生成策略