

本文档旨在说明如果快速的接入集成推送 SDK, 快速实现小米, 华为, 魅族的推送接入, 有关详细的设计文档参考集成推送设计说明文档

准备工作

编译本项目, 配置完 android sdk 后执行以下命令, 或者你可以将本工程导入到 android studio 中

```
./gradlew clean assemble
```

一 AndroidManifest 配置

由于三方 SDK 的权限, 组件全部配置在 ups-push-sdk 的 aar 的 AndroidManifest 中, 因此开发者只需关注各个平台与应用相关的配置即可, 更加详细的配置参考 Ups_PushDemo

1.1 权限声明

NOTE: 请将 \${PACKAGE_NAME} 换成实际应用的包名

```
<!-- 小米个性化配置 需要到应用的主 manifest 中去配置-->
<!-- the following 2 ${PACKAGE_NAME} should be changed to your
package name -->
<permission
android:name="${PACKAGE_NAME}.permission.MIPUSH_RECEIVE"
        android:protectionLevel="signature" />
<uses-permission
android:name="${PACKAGE_NAME}.permission.MIPUSH_RECEIVE" />
```

```
<!-- 魅族个性化配置 需要到应用的主 manifest 中去配置-->
<!-- the following 4 ${PACKAGE_NAME} should be changed to your
package name -->
<uses-permission
android:name="com.meizu.flyme.push.permission.RECEIVE"></uses-
permission>
<permission
android:name="${PACKAGE_NAME}.push.permission.MESSAGE"
android:protectionLevel="signature"/>
<uses-permission
android:name="${PACKAGE_NAME}.push.permission.MESSAGE"></uses-
permission>
```

```

    <uses-permission android:name="com.meizu.c2dm.permission.RECEIVE"
/>
    <permission android:name="${PACKAGE_NAME}.permission.C2D_MESSAGE"
        android:protectionLevel="signature"></permission>
    <uses-permission
android:name="${PACKAGE_NAME}.permission.C2D_MESSAGE"/>

```

1.2 基础组件

```

<!-- 华为移动服务配置,将${PACKAGE_NAME}替换为实际包名 -->
<provider
    android:name="com.huawei.hms.update.provider.UpdateProvider"
    android:authorities="${PACKAGE_NAME}.hms.update.provider"
    android:exported="false"
    android:grantUriPermissions="true" >
</provider>

```

实现 UpsPushReceiver

应用只需要继承 UpsPushMessageReceiver, 覆盖其中的方法即可, 如下

```

public class UpsReceiver extends UpsPushMessageReceiver{
    @Override
    public void onThroughMessage(Context context, UpsPushMessage
upsPushMessage) {

    }

    @Override
    public void onNotificationClicked(Context context, UpsPushMessage
upsPushMessage) {

    }

    @Override
    public void onNotificationArrived(Context context, UpsPushMessage
upsPushMessage) {

    }

    @Override

```

```

    public void onNotificationDeleted(Context context, UpsPushMessage
upsPushMessage) {

        }

    @Override
    public void onUpsCommandResult(Context context, UpsCommandMessage
upsCommandMessage) {

        }

```

并在 AndroidManifest.xml 中配置此组件，此组件的实际路径以此类的实际路径为准，下面包名只是示例说明

```

<!-- 统一推送配置 -->
<!-- the following 4 ${PACKAGE_NAME} should be changed to your actual
package name where the implemented UpsReceiver belong to -->
    <receiver android:name="${PACKAGE_NAME}.UpsReceiver">
        <intent-filter>
            <!-- 接收 push 消息 -->
            <action android:name="com.meizu.ups.push.intent.MESSAGE" />
        </intent-filter>
    </receiver>

```

1.3 应用配置信息

NOTE: 华为需要在 AndroidManifest 自行配置

```

<!-- APPID 华为移动服务配置 请将 value 替换成实际的 appid -->
    <meta-data
        android:name="com.huawei.hms.client.appid"
        android:value="${HUAWEI_APP_ID}" />

```

1.4 库引入说明

魅族，华为的包默认依赖相关的 artifactory 库，需要在你的工程根目录加入如下 maven url 配置

```

//魅族的 pushsdk 存放在 jcenter 中
jcenter()
//华为的库存放其私有仓库中
maven {
    url 'http://developer.huawei.com/repo/'
}

```

```
}
```

小米的库需要手动将其 jar 放到工程的 lib 目录下，小米的 pushsdk jar [下载](#)

最后在你的工程 build.gradle 配置 ups-push 的依赖

```
compile 'com.meizu.flyme.internet:push-ups:1.1.+'
```

二 统一推送平台配置

2.1 应用信息平台设置

推荐到[魅族统一推送平台](#)应用配置-> 添加多渠道添加各个平台的 AppId, AppKey, AppSecret, 如下图所示:

添加多渠道

渠道	App ID	App Key
小米	2882303761517631454	5581763188454
魅族	111684	a16517ce04d64c3abdfbedb48
华为	100125059	1111

2.2 应用信息本地设置测试[可选]

应用也可以在 AndroidManifest 配置 AppID, AppKey 信息方便本地测试，ups-pushsdk 会优先读取本地 AndroidManifest 中的配置信息

NOTE: 小米的 APP_ID, APP_KEY 需要进行字符转义, 不然无法正确读出, 例如小米的 appid 为 2882303761517631454 需要如下配置
android:value="\02882303761517631454"

```
<meta-data
    android:name="XIAOMI_APP_ID"
    android:value="{XIAOMI_APP_ID}"/>
```

```
<meta-data
    android:name="XIAOMI_APP_KEY"
    android:value="{XIAOMI_APP_KEY}"/>
```

```
<meta-data
    android:name="MEIZU_APP_ID"
    android:value="{MEIZU_APP_ID}"/>
```

```
<meta-data
    android:name="MEIZU_APP_KEY"
    android:value="{MEIZU_APP_KEY}"/>
```

三 通知栏自定义行为使用方式

统一推送平台目前由五种方式，[详情参见](#)

问题说明

- 华为无法订阅成功
 - 升级华为移动服务至最新版本,最好为此版本 2.5.3.302,到华为应用商店更新即可
 - [华为消息通用错误码](#)
- Ups 平台报签名错误
 - 无法获取手机唯一识别码,需要到各个平台的手机管家中,打开获取手机信息的权限
- 通知栏问题
 - OPPO ColorOS 需要打开通知栏权限才可展示通知栏
 - Android O 平台没有设置 channelId,无法展示通知栏问题
- 判定厂商
 - OPPO MANUFACTURER:OPPO model:R7Plusm
 - VIVO MANUFACTURER:vivo model:vivo X6D
 - HUAWEI MANUFACTURER:HUAWEI model:MHA-AL00 brand:MHA
 - MEIZU MANUFACTURER:MEIZU
- ups-pushsdk 的依赖关系./gradlew -p UpsPushSDKLib/ -q dependencies --configuration compile

```
+--- com.huawei.android.hms:push:2.5.2.300
|   \--- com.huawei.android.hms:base:2.5.2.300
+--- com.meizu.flyme.internet:push-internal:3.6.3
\--- com.meizu.flyme.internet:push-open:2.0.0
```

```
\--- com.squareup.wire:wire-runtime:2.3.0-RC1
  \--- com.squareup.okio:okio:1.11.0
```