

Android PUSH-SDK API 集成指南

1. 集成 SDK

1.1. 导入推送 jar 包

将解压后的 libs 文件夹中 vivo_pushsdk-VERSION.jar(vivo_pushsdk-VERSION.jar 为集成的 jar 包名字, VERSION 为版本名称)拷贝到您的工程的 libs 文件夹中。

1.2. 配置 AndroidManifest 文件

在当前工程的 AndroidManifest.xml 文件中, 添加权限和声明信息, 详见

```
1. <!--推送需要的权限列表-->
2. <uses-permission android:name="android.permission.INTERNET"/>
3. <!--推送服务需要配置的 service、activity-->
4. <service
5.     android:name="com.vivo.push.sdk.service.CommandClientService"
6.     android:exported="true"/>
7. <activity
8.     android:name="com.vivo.push.sdk.LinkProxyClientActivity"
9.     android:exported="false"
10.    android:screenOrientation="portrait"
11.    android:theme="@android:style/Theme.Translucent.NoTitleBar"/>
12. <!--推送配置项-->
13. <meta-data
14.     android:name="com.vivo.push.api_key"
15.     android:value="xxxxxxx"/>
16. <meta-data
17.     android:name="com.vivo.push.app_id"
18.     android:value="xxxx"/>
```

备注:

- com.vivo.push.app_id , com.vivo.push.api_key 由开放平台生成, 详见 vivo push 操作手册。
- 推送服务 SDK 支持的最低安卓版本为 Android 4.0 系统。

1.3. 启动云推送

```
1. //      在当前工程入口函数, 建议在 Application 的 onCreate 函数中, 添加以下代码:
2.      PushClient.getInstance(getApplicationContext()).initialize ();
3. //      当需要打开推送服务时, 调用以下代码:
4.      PushClient.getInstance(getApplicationContext()).turnOnPush(new IPush
      ActionListener() {
5.          @Override
6.          public void onStateChanged(int state) {
7.              // TODO: 开关状态处理
8.          }
9.      });
```

1.4. 自定义回调类

在当前工程中新建一个类 `PushMessageReceiverImpl` 并实现 `OpenClientPushMessageReceiver` 中的 `onNotificationMessageClicked` 方法。

在当前工程的 `AndroidManifest.xml` 文件中, 添加自定义 Receiver 信息如下:

```
1. <!-- push 应用定义消息 receiver 声明 -->
2. <receiver android:name="xxx.xxx.xxx.PushMessageReceiverImpl" >
3.     <intent-filter>
4.         <!-- 接收 push 消息 -->
5.         <action android:name="com.vivo.pushclient.action.RECEIVE" />
6.     </intent-filter>
7. </receiver>
```

2. 处理推送消息

接收通知消息

当设备接收到通知消息后, 查看手机的通知栏, 可以看到通知栏内的该新通知展示。当点击通知时, 回调 `PushMessageReceiverImpl#onNotificationMessageClicked` 方法。

3. 混淆说明



若需要混淆 app，请在混淆文件中添加以下说明，防止 SDK 内容被二次混淆，自定义回调类切勿混淆。

```
1. -dontwarn com.vivo.push.**
2. -keep class com.vivo.push.**{*;}
3. -keep class com.vivo.vms.**{*;}
4. -keep class xxx.xxx.xxx.PushMessageReceiverImpl{*;}
```