

# EaseMob Push SDK User Guide

## 1. 工程设置

- 将easemobpush-sdk-1.0.0.jar 拷贝到应用的libs 目录
- 修改AndroidManifest.xml 添加 需要的权限如下

```
<!-- easemob push required permission -->

<uses-permission android:name="android.permission.VIBRATE" />

<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" />

<uses-permission android:name="android.permission.CHANGE_WIFI_STATE" />

<uses-permission android:name="android.permission.READ_PHONE_STATE"/>

<uses-permission android:name="android.permission.GET_TASKS" />

<uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED" />

<uses-permission android:name="android.permission.WAKE_LOCK" />
```

- 继续修改AndroidManifest.xml 添加service 和 appkey （appkey是从 easemob 网站申请的应用的appkey）

```
<!-- easemob push sdk service -->

<service android:name="com.easemob.EaseMobService" />

<!-- easemob push appkey -->

<meta-data android:name="EASEMOB_APPKEY" android:value="pushtest1"/>
```

- 修改AndroidManifest.xml 添加message receiver

```
<!-- receiver to receive message from easemob push service -->

<receiver android:name="com.easemob.push.example.PushMessageReceiver" >

    <intent-filter>

        <action android:name="com.easemob.push.action" />

    </intent-filter>

</receiver>
```

## 2. 初始化 EaseMob Push SDK

- 在创建应用时初始化 EaseMob SDK

```
public void onCreate() {  
    super.onCreate();  
    Log.i("app", "initialize EaseMob Push Service");  
    EaseMobPush.init(this);  
}
```

- 如果应用退出后，不想继续接收推送，可以在应用结束时，终止推送服务

(如果想一直接收推送，可以忽略此步)

```
public void onTerminate () {  
    EaseMobPush.stopPush(this);  
    super.onTerminate();  
}
```

## 3. 接收推送消息

Push SDK 会broadcast 三类消息：

- Registration: 注册消息: SDK 注册成功，会返回应用一个registration ID. 应用可以根据需要，把这个registryID 传到server，生成应用的用户名到推送 registration ID的映射关系。
- Notification: 通知消息: SDK会生产系统通知，同时发送broadcast 给应用
- Message: 透穿消息。SDK直接发给应用，不会通过系统通知

示例代码：

```
public class PushMessageReceiver extends BroadcastReceiver {  
  
    private final static String TAG = "pushMsgReceiver";  
  
    @Override  
    public void onReceive(Context context, Intent intent) {  
        int type = intent.getIntExtra(EaseMobPush.PUSH_INTENT_TYPE, 0);  
        switch (type) {
```

```

        case EaseMobPush.PUSH_TYPE_REGISTRATION:

            //for registration intent, will have device registration id

            String registrationId = intent.getStringExtra
(EaseMobPush.PUSH_REGISTRATION_ID);

            Log.d(TAG, "application received push registration id:" +
registrationId);

            MainActivity.printDebugMsg("application received push registration id:"
+ registrationId);

            break;

        case EaseMobPush.PUSH_TYPE_NOTIFICATION:

            //for notification intent, will have notification id

            String notificationId = intent.getStringExtra
(EaseMobPush.PUSH_NOTIFICATION_ID);

            Log.d(TAG, "application received push notification, id:" +
notificationId);

            MainActivity.printDebugMsg("application received push notification, id:"
+ notificationId);

            break;

        case EaseMobPush.PUSH_TYPE_MESSAGE:

            //for notification intent, will have push message

            String message = intent.getStringExtra(EaseMobPush.PUSH_MESSAGE);

            Log.d(TAG, "application received push msg:" + message);

            Toast.makeText(context, "push msg:" + message, Toast.LENGTH_LONG).show
();

            MainActivity.printDebugMsg("application received push msg:" + message);

            break;

    }

}

}

```