华为+小米推送

一:华为官方文档

1:链接:

http://developer.huawei.com/consumer/cn/service/hms/catalog/huaweipush.html?page=hmssdk\_huaweipush\_introduction

2:接入流程



3:长连接成功后, 开始注册推送服务, 判断是小米推送还是华为推送服务

4:public class HuaweiPushReceiver extends PushReceiver

/\*\*

\* 应用需要创建一个子类继承com.huawei.hms.support.api.push.PushReceiver，

\* 实现onToken，onPushState ，onPushMsg，onEvent，这几个抽象方法，用来接收token 返回，push连接状态，透传消息和通知栏点击事件处理。

\* onToken 调用getToken方法后，获取服务端返回的token结果，返回token以及belongId

\* onPushState 调用getPushState方法后，获取push连接状态的查询结果

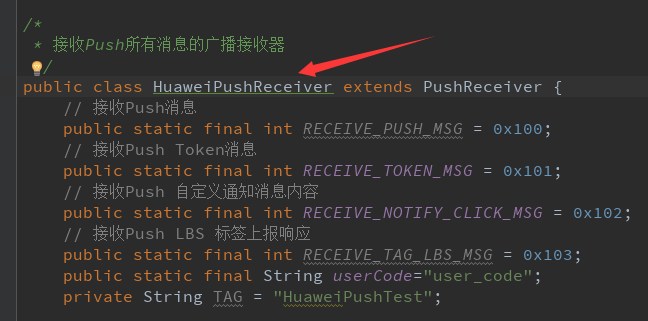
\* onPushMsg 推送消息下来时会自动回调onPushMsg方法实现应用透传消息处理。本接口 必须被实现。 在开发者网站上发送push消息分为通知和透传消息

\*通知为直接在通知栏收到通知，通过点击可以打开网页，应用 或者富媒体，不会收到 onPushMsg消息

\*透传消息不会展示在通知栏，应用会收到onPushMsg

\* onEvent 该方法会在设置标签、点击打开通知栏消息、点击通知栏上的按钮之后被调用。 由业务决定是否调用该函数。

\*/



收到的信息都在这个地方做的处理;

二:小米推送

1:小米推送官方文档:

<https://dev.mi.com/console/doc/detail?pId=230>

2:小米推送支持消息栏通知和透传通知

小米推送目前支持两种消息传递方式：透传方式和通知栏方式。

透传消息到达手机端后，SDK会将消息通过广播方式传给

AndroidManifest中注册的PushMessageReceiver

的子类的onReceivePassThroughMessage；

对于通知栏消息，SDK会根据消息中设置的信息弹出通知栏通知，

通知消息到达时会到达PushMessageReceiver子类的

onNotificationMessageArrived方法,用户点击之后再传给您

的PushMessageReceiver的子类的

onNotificationMessageClicked方法；对于应用在前台时不

弹出通知的通知消息，SDK会将消息通过广播方式传给

AndroidManifest中注册的PushMessageReceiver的子类的

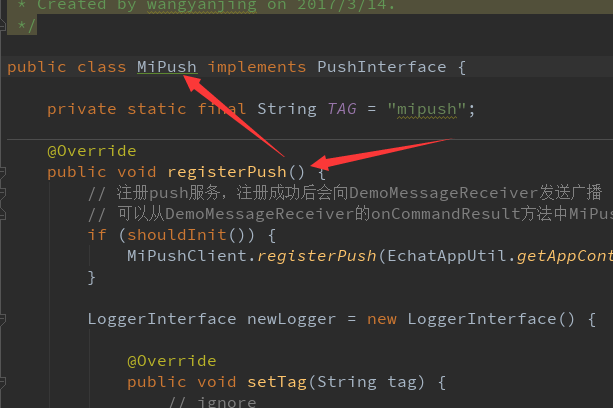
onNotificationMessageArrived方法

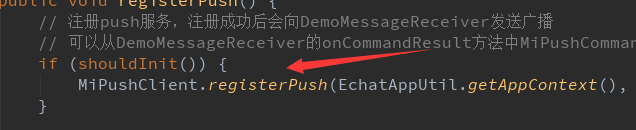
(在MIUI上，如果没有收到onNotificationMessageArrived回调，

是因为使用的MIUI版本还不支持该特性，需要升级到MIUI7之后。

非MIUI手机都可以收到这个回调)。

3:注册推送服务





4:public class OnMiPushMessageReceiver extends PushMessageReceiver

为了接收消息，您需要自定义一个继承自PushMessageReceiver

类的BroadcastReceiver，实现其中的

onReceivePassThroughMessage，

onNotificationMessageClicked，

onNotificationMessageArrived，

onCommandResult和

onReceiveRegisterResult方法，

然后把该receiver注册到AndroidManifest.xml文件中。

onReceivePassThroughMessage用来接收服务器发送的透传消息，

onNotificationMessageClicked用来接收服务器发来的通知栏

消息（用户点击通知栏时触发），

onNotificationMessageArrived用来接收服务器发来的通知栏

消息（消息到达客户端时触发，并且可以接收应用在前台时不

弹出通知的通知消息），

onCommandResult用来接收客户端向服务器发送命令消息后返回的响应，

onReceiveRegisterResult用来接受客户端向服务器发送注册命令消息后返回的响应。

5:上面所说的方法都是运行在子线程的, 所以更新的时候要注意;