一、极光推送

**文件：**AppDelegate+Jpush

**接口：**

/\*

\* @param launchingOption 启动参数.

\* @param appKey 一个JPush 应用必须的,唯一的标识. 请参考 JPush 相关说明文档来获取这个标识.

\* @param channel 发布渠道. 可选.

\* @param isProduction 是否生产环境. 如果为开发状态,设置为 NO; 如果为生产状态,应改为 YES.

\*/

-(void)setupWithOption:(NSDictionary \*)launchingOption appKey:(NSString \*)appKey channel:(NSString \*)channel apsForProduction:(BOOL)isProduction;

**使用：**

1、pod 'JPush', '3.0.2'，导入极光推送

2、Info.plish文件添加项（具体配置参考项目的plist）

App Transport Security Settings

Required background modes

3、项目开启推送功能你Capabilities🡪PushNotifications

4、推送证书，描述文件的导入，在证书文件夹

5、到极光推送管理平台创建项目，导入推送证书（证书密码为1），这里会得到对应的appKey，用于下面的初始化（对于新项目，需要配置新的证书，具体可参考极光相关文档）

6、在AppDelegate中

#import "AppDelegate+Jpush.h"

- (BOOL)application:(UIApplication \*)application didFinishLaunchingWithOptions:(NSDictionary \*)launchOptions {

// Override point for customization after application launch.

[self setupWithOption:launchOptions appKey:@"c1609a5182c9faab5f5b4e8c" channel:@"push" apsForProduction:NO];

return YES;

}

运行，如果log如下(有**Device Token就说明成功)**

**2017-08-21 11:52:43.927445 PushServer[683:339286] | JIGUANG | I - [JIGUANGService]**

**--------------------------- JPush Log ----------------------------**

**--------------------JPush SDK Version:3.0.2--build:28----------**

**--------------------JCore Lib Version:1.1.3--build:21----------**

**-----------------AppKey:c1609a5182c9faab5f5b4e8c----------------**

**----------------------------------------------------------------**

**2017-08-21 11:52:44.001459 PushServer[683:339272] Device Token: <77ddd1eb ec0362e0 9fb9d9b5 1015c5ff c35f67bd dbb61e2a a3219c01 43648e14>**

7、测试。进入极光管理平台，选择发送通知，进行测试

//手机版本大于8.0的通知接收，app在后台不会执行，只有在点击通知进入app才会执行，如果app在前台，不会弹出通知框，但下面是会执行的

- (void)application:(UIApplication \*)application didReceiveRemoteNotification:(NSDictionary \*)userInfo fetchCompletionHandler:(void (^)(UIBackgroundFetchResult))completionHandler {

[JPUSHService handleRemoteNotification:userInfo];

NSLog(@"收到通知>8.0:%@", userInfo);

completionHandler(UIBackgroundFetchResultNewData);

}

正常情况下，这个回调接口是可以收到你刚才的测试通知。

**说明：**

这里使用的是极光推送，上面介绍的是极光通过Apns的推送，JPush 服务器端代理开发者向 Apple APNs 推送通知；

### 极光还拥有自己的一套推送，其实就是通过自己开发一套网络通讯机制，使用长连接。你app启动了，就可以和极光通讯了，要使用极光的应用内消息，进入到AppDelegate+Jpush.m开启注释掉的

//极光推送通过长连接网络的推送，即是极光的自定义消息，不需要网络推送的可以屏蔽

// NSNotificationCenter \*defaultCenter = [NSNotificationCenter defaultCenter];

// [defaultCenter addObserver:self

等等…

通知的回调

//极光通过自带网络发送的信息接收

- (void)networkDidReceiveMessage:(NSNotification \*)notification {

二、本地通知

接口：

//注册，最好在appDelegate里面调用这个

- (void)registerNotify;

/\*

发送本地通知

title：通知标题

alertBody：通知内容

infoDic：通知携带的内容

\*/

- (void)pushLocalNotification:(NSString \*)title alertBody:(NSString \*)alertBody infoDic:(NSDictionary \*)infoDic;

//flag通知的唯一标示，这个通知是唯一的，通知列表只会出现一种，不会重复

- (void)pushLocalNotification:(NSString \*)title alertBody:(NSString \*)alertBody flag:(NSString \*)flag infoDic:(NSDictionary \*)infoDic;

//清除特定的通知,flag通知的唯一标示

* (void)removeLocalNotification:(NSString \*)flag;

三、检查AppStore上的软件版本，用这个来判断是否要更新

接口

/\*

appStoreUrl： appStore上的url

isNeedUpdate：是否需要更新，succ：appStore版本比当前版本号大

\*/

+(void)checkAppFromAppStoreWithUrl:(NSString \*)appStoreUrl isNeedUpdate:(void(^)(BOOL succ))isNeedUpdate;

相对比较简单，具体调试请参考demo