

# iOS新手开发互助群:529052159

多一个朋友，多一条路！相聚，相识，互相帮助，只因有你这个世界才变的美好！

昵称：iOS开发群529052159

园龄：10个月

粉丝：16

关注：4

+加关注

博客园 首页 新随笔 联系 管理 订阅 XML

随笔- 6 文章- 124 评论- 0

## iOS极光推送的基本使用

昨天花了一下午的时间研究了极光推送，也前也是没做过，不知道从何下手！才开始的时候一看官方的SDK感觉好难，不过经过一系列的捣鼓之后，手机收到了推送信息，感觉其实并没有那么难！

## 1.配置开发证书（得有开发者账号，个人，企业的都可以）

## 搜索

## 常用链接

我的随笔  
我的评论  
我的参与  
最新评论  
我的标签  
更多链接

## 我的标签

iOS开发(122)  
iOS(119)  
iOS动画(5)  
iOS面试题(5)  
iOS性能优化(3)  
iOS蓝牙(2)  
iOS推送(2)  
GCD(2)  
iOS内存管理(2)  
iOS多线程(2)  
更多

### 开发环境测试

在对 JPush iOS 开发环境进行测试前，请确保 3 个统一：

- App 是开发环境打包（开发证书 Development）
- 上传了开发证书并验证通过
- Portal 上的应用设置为开发环境

### 发布环境测试

在对 JPush iOS 生产环境进行测试前，请确保 3 个统一：

- App 是 ad-hoc 打包或者App Store 版本（发布证书 Production）
- 上传了发布证书并验证通过
- Portal 上的应用设置为生产环境
- 

## iOS 证书 设置指南

### 创建应用程序ID

- 登陆 [iOS Dev Center](#) 选择进入iOS Provisioning Portal。



- 在 [iOS Provisioning Portal](#)中，点击App IDs进入App ID列表。

## 随笔档案

2016年3月 (2)

2016年1月 (4)

## 文章分类

Android(1)

C语言

HTML5(3)

iOS开发(116)

OC

PHP

## 阅读排行榜

1. iOS极光推送的基本使用(1570)

2. iOS常用的封装方法(1551)

3. iOS百度推送的基本使用(512)

4. iOS极光推送(377)

5. OC字符串的常用方法(259)

## 推荐排行榜

1. iOS极光推送(1)

2. iOS百度推送的基本使用(1)



- 创建 App ID，如果 ID 已经存在可以直接跳过此步骤



- 为 App 开启 Push Notification 功能。如果是已经创建的 App ID 也可以通过设置开启 Push Notification 功能。



根据实际情况完善 App ID 信息并提交,注意此处需要指定具体的 Bundle ID 不要使用通配符。



### 配置和下载证书

- 如果你之前没有创建过 Push 证书或者是要重新创建一个新的，请在证书列表下面新建。



- 新建证书需要注意选择证书种类（开发证书用于开发和调试使用，生产证书用于 App Store 发布）



- 点击 Continue 后选择证书对应的应用ID，然后继续会出现“About Creating a Certificate Signing Request (CSR)”。



- 根据它的说明创建打开KeychainAccess 创建 Certificate Signing Request。



- 填写“User Email Address”和“Common Name” 后选择 Saved to disk 进行保存 。



- 继续返回Apple developer 网站点击 Continue ，上传刚刚生成的 .certSigningRequest 文件生成 APNs Push Certificate。

- 下载并双击打开证书，证书打开时会启动“钥匙串访问”工具。

- 在“钥匙串访问”中你的证书会显示在“我的证书”中，注意选择“My Certificates” 和“login”



### 导出 .p12 证书文件

注意要选“login”和“My Certificates” 导出证书时要选中证书文件，不要展开private key。



- 将文件保存为Personal Information Exchange (.p12)格式。



- 将文件保存为Personal Information Exchange (.p12)格式。

### 上传证书

在 JPush 管理 Portal 上，针对某应用程序，上传上面步骤得到 .p12 证书文件。这是 iOS SDK 能够接收到 JPush 推送消息的必要步骤。

### Provisioning Profile的创建

- 创建Provisioning Profile的前提，已在Apple Developer网站创建待发布应用所使用的Bundle ID的App ID ，且为该App ID创建了APNs证书，如下图：



- 创建App ID、APN证书和p12证书的导出的具体步骤请看 :[iOS 证书 设置指南](#)
- 在苹果开发者账号的Provisioning Profile页面点击下图按钮，创建Provisioning Profile



- 选择此Provisioning Profile的环境后点击[Continue]:



- 选择要创建Provisioning Profile的App ID后点击[Continue]:



- 选择所属的开发者证书，（这里创建了多个开发者证书，建议只创建一个，方便管理）为了方便，选择了[Select All]，再点击[Continue]进入下一步:



- 为该Provisioning Profile选择将要安装的设备（一般选择[Select All]），点击[Continue]:



- 给该Provisioning Profile填写Profile Name，点击[generate]完成创建。



- 填写完Profile Name后点击[generate]完成创建，之后点击[Download]下载Provisioning Profile



- 双击下载下来的Provisioning Profile，添加到xcode。

## 2. Xcode的证书配置教程

参照[iOS SDK 集成指南](#)集成JPush SDK 和上传了推送用到的p12证书后在编译运行前需要先配置一下证书，步骤如下:

- 打开xxx-info.plist的Bundle identifier项把上传到JPush 控制台的bundle id填写进去:



- 点击项目，选择目标TARGETS后进入Build Setting 界面，搜索“Code signing”，按照下图配置



### 客户端设置

#### 开启Remote notifications

需要在Xcode 中修改应用的 Capabilities 开启Remote notifications，请参考下图:



## 3.SDK集成步骤

## 集成压缩包内容

包名为JPush-iOS-SDK-{版本号}

- lib文件夹：包含头文件 JPUSHService.h，静态库文件jpush-ios-x.x.x.a，支持的iOS版本为 5.0 及以上版本。（请注意：模拟器不支持APNs）
- pdf文件：集成指南
- demo文件夹：示例

## 开发环境

- 使用Xcode 6及以上版本可以使用新版Push SDK，XCode 5环境下需要运行旧版本SDK(1.7.4)

## 1、在JPush Portal上创建应用

- 在 JPush的管理Portal 上创建应用并上传APNs证书。如果对APNs证书不太了解 请参考：[iOS 证书设置指南](#)



- 创建成功后自动生成 AppKey 用以标识该应用。



## 2、导入API开发包到应用程序项目

- 将SDK包解压，在XCode中选择"Add files to 'Your project name'..."，将解压后的lib子文件夹（包含JPU SHService.h、jpush-ios-x.x.x.a）添加到你的工程目录中。

## 3、必要的框架

- CFNetwork.framework
- CoreFoundation.framework
- CoreTelephony.framework
- SystemConfiguration.framework
- CoreGraphics.framework
- Foundation.framework
- UIKit.framework
- Security.framework
- Xcode7需要的是libz.tbd；Xcode7以下版本是libz.dylib

## 4、Build Settings

如果你的工程需要支持小于7.0的iOS系统，请到Build Settings 关闭 bitCode 选项，否则将无法编译通过。

- 设置 Search Paths 下的 User Header Search Paths 和 Library Search Paths，比如SDK文件夹（默认为lib）与工程文件在同一级目录下，则都设置为"\${SRCROOT}/{静态库所在文件夹名称}"即可。

## 5、创建并配置PushConfig.plist文件

2.1.0 版本开始，新增了带参数的setupWithOptions初始化方法，可通过此方法等参数传入AppKey等信息。

1.8.8及之前版本的 JPush SDK只能通过PushConfig.plist配置AppKey等信息。

在你的工程中创建一个新的Property List文件，并将其命名为PushConfig.plist，文件所含字段如下：

- CHANNEL
  - 指明应用程序包的下载渠道，为方便分渠道统计，具体值由你自行定义，如：App Store。
- APP\_KEY
  - 填写管理Portal上创建应用后自动生成的AppKey值。请确保应用内配置的 AppKey 与第1步在 Portal 上创建应用后生成的 AppKey 一致。
- APS\_FOR\_PRODUCTION
  - 1.3.1版本新增，用于标识当前应用所使用的APNs证书环境。
  - 0 (默认值)表示采用的是开发证书，1 表示采用生产证书发布应用。
  - 注：此字段的值要与Build Settings的Code Signing配置的证书环境一致。
- 在1.2.2或之前版本的配置文件中，有 TEST\_MODE 这个键，新版的SDK不再使用，可以将它删除。

PushConfig.plist文件示例图：



## 6、添加代码

2.1.0版本开始,API类名为JPUSHService，不再使用原先的APService。

### 允许XCode7支持Http传输方法

如果用的是Xcode7时，需要在App项目的plist手动加入以下key和值以支持http传输：

```
<key>NSAppTransportSecurity</key>
<dict>
<key>NSAllowsArbitraryLoads</key>
<true/>
</dict>
```

### 集成所需API

APIs 主要集中在 JPUSHService 接口类里。

- 初始化JPush方法分为两个：
  - 1.8.8及以下版本使用的是已过期的初始化方法。升级到2.1.0的老用户仍可继续使用旧的初始化方法。
  - 2.1.0版本开始提供带appkey等参数的新初始化方法。使用此方法无需再添加PushConfig.plist配置JPUSH的AppKey等字段。

## 4.项目代码：

AppDelegate.h里面的代码：

```
#import<UIKit/UIKit.h>

staticNSString *appKey = @"4fd48a0712a3fde75eb1c7423";//申请应用成功以后官方会提供给你
staticNSString *channel = @"Publish channel";

staticBOOL isProduction = FALSE;

@interface AppDelegate :UIResponder <UIApplicationDelegate>

@property (strong, nonatomic) UIWindow *window;

@end
```

#import "AppDelegate.m"中的代码

```
#import "AppDelegate.h"
```

```
#import "JPUSHService.h"
```

```
@interface AppDelegate ()
```

```
@end
```

```
@implementation AppDelegate
```

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

```
    // Override point for customization after application launch.
```

```
    self.window.backgroundColor = [UIColor whiteColor];
```

```
    [self.window makeKeyAndVisible];
```

```
    if ([[UIDevice currentDevice].systemVersion floatValue] >= 8.0) {
```

```
        //可以添加自定义categories
```

```
        [JPUSHService registerForRemoteNotificationTypes:(UIUserNotificationTypeBadge |
```

```
                                                            UIUserNotificationTypeSound |
```

```
                                                            UIUserNotificationTypeAlert)
```

```
                categories:nil];
```

```
    } else {
```

```
        //categories 必须为nil
```

```
        [JPUSHService registerForRemoteNotificationTypes:(UIRemoteNotificationTypeBadge |
```

```
                                                            UIRemoteNotificationTypeSound |
```

```
                                                            UIRemoteNotificationTypeAlert)
```

```
                categories:nil];
```

```
    }
```

```
    [JPUSHService setupWithOptions:launchOptions appKey:appKey
```

```
                                channel:channelapsForProduction:NO];
```

```
    return YES;
```

```
}
```

```
- (void)application:(UIApplication *)application didRegisterForRemoteNotificationsWithDeviceToken:
(NSData *)deviceToken {
```

```
// Required
[JPUSHService registerDeviceToken:deviceToken];
}

- (void)application:(UIApplication *)application didReceiveRemoteNotification:
(NSDictionary *)userInfo {

    // Required, For systems with less than or equal to iOS6
    [JPUSHService handleRemoteNotification:userInfo];
}

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

    // iOS 7 Support Required
    [JPUSHService handleRemoteNotification:userInfo];
    completionHandler(UIBackgroundFetchResultNewData);
}

- (void)application:(UIApplication *)application didFailToRegisterForRemoteNotificationsWithError:
(NSError *)error {

    //Optional
    NSLog(@"did Fail To Register For Remote Notifications With Error: %@", error);
}

- (void)applicationWillResignActive:(UIApplication *)application {

    // Sent when the application is about to move from active to inactive state. This can occur for
    certain types of temporary interruptions (such as an incoming phone call or SMS message) or when
    the user quits the application and it begins the transition to the background state.

    // Use this method to pause ongoing tasks, disable timers, and throttle down OpenGL ES frame
    rates. Games should use this method to pause the game.
}

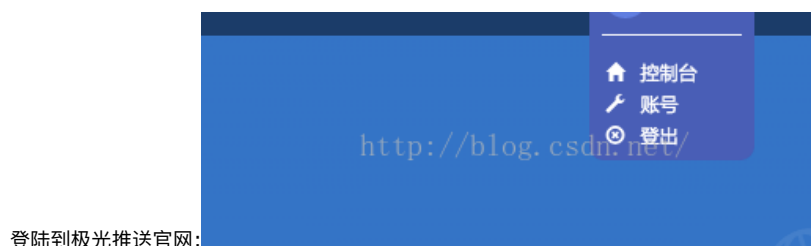
- (void)applicationDidEnterBackground:(UIApplication *)application {

    // Use this method to release shared resources, save user data, invalidate timers, and store
    enough application state information to restore your application to its current state in case it is
    terminated later.

    // If your application supports background execution, this method is called instead of
    applicationWillTerminate: when the user quits.
}
```

```
- (void)applicationWillEnterForeground:(UIApplication *)application {  
    // Called as part of the transition from the background to the inactive state; here you can undo  
    many of the changes made on entering the background.  
}  
  
- (void)applicationDidBecomeActive:(UIApplication *)application {  
    // Restart any tasks that were paused (or not yet started) while the application was inactive. If  
    the application was previously in the background, optionally refresh the user interface.  
}  
  
- (void)applicationWillTerminate:(UIApplication *)application {  
    // Called when the application is about to terminate. Save data if appropriate. See also  
    applicationDidEnterBackground:..  
}  
  
@end
```

#### 4.真机测试：



点击控制台：





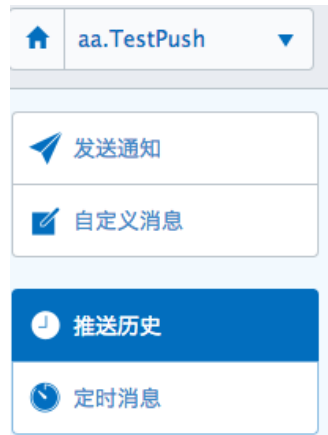
选择应用，也就是你创建测试的项目：我的项目是aa.TestPush



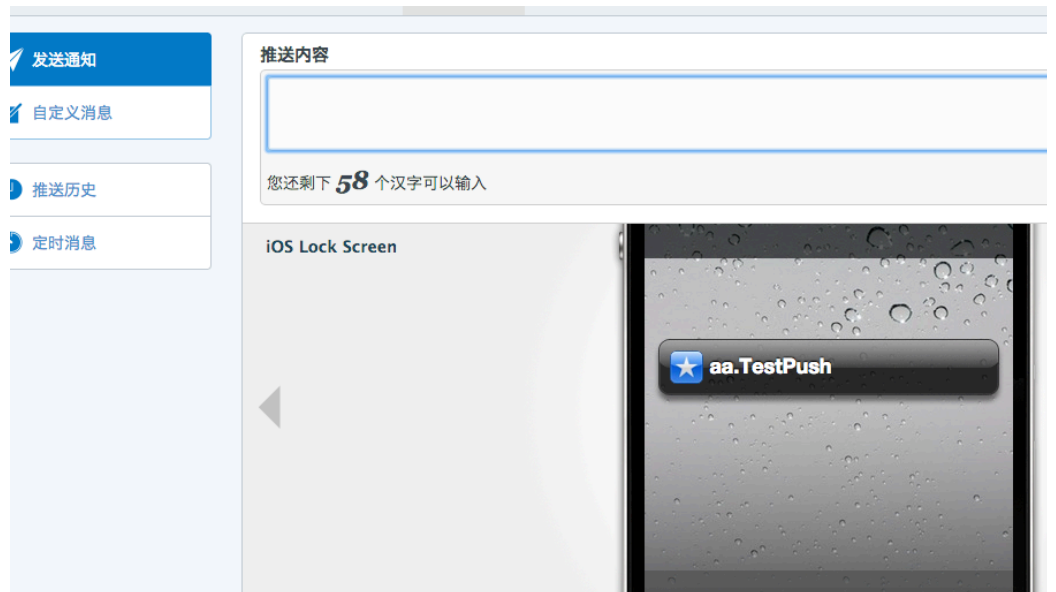
选中以后点击进入到下面这里：



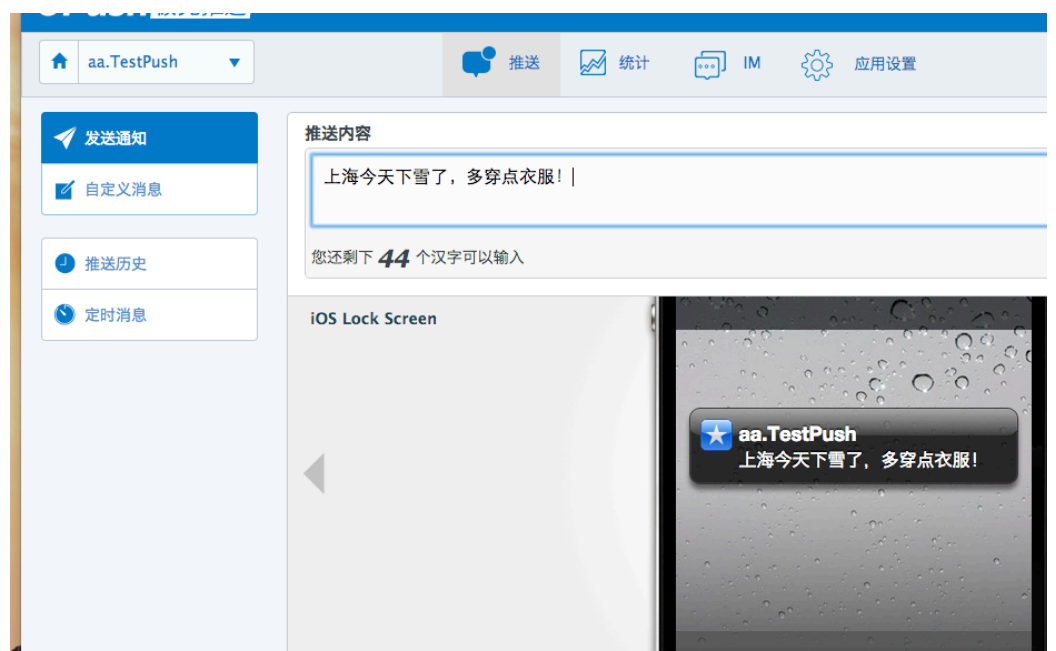
点击上面的推送：



点击左边的发送通知，也可以选择自定义消息(这里我选择的发送通知)：



这里我随便写点什么用我手机测试：



推送对象

☒ iOS 开发环境 ☐ iOS 生产环境 ☐ Android ☐ WinPhone

☒ 广播(所有人) ☐ 设备标签(Tag) ☐ 设备别名(Alias) ☐ Registration ID

将向应用所有的注册用户推送此通知。

发送时间

☒ 立即 ☐ 定时

是否启用定速推送?

☐ 定速推送

▶ 可选设置

点击立即发送：



JPush极光推送

aa.TestPush

aa.TestPush

选择分组

全部应用

自定义消息

推送历史

定时消息

推送

统计

IM

应用设置

搜索(tag/alias/Reg.ID)

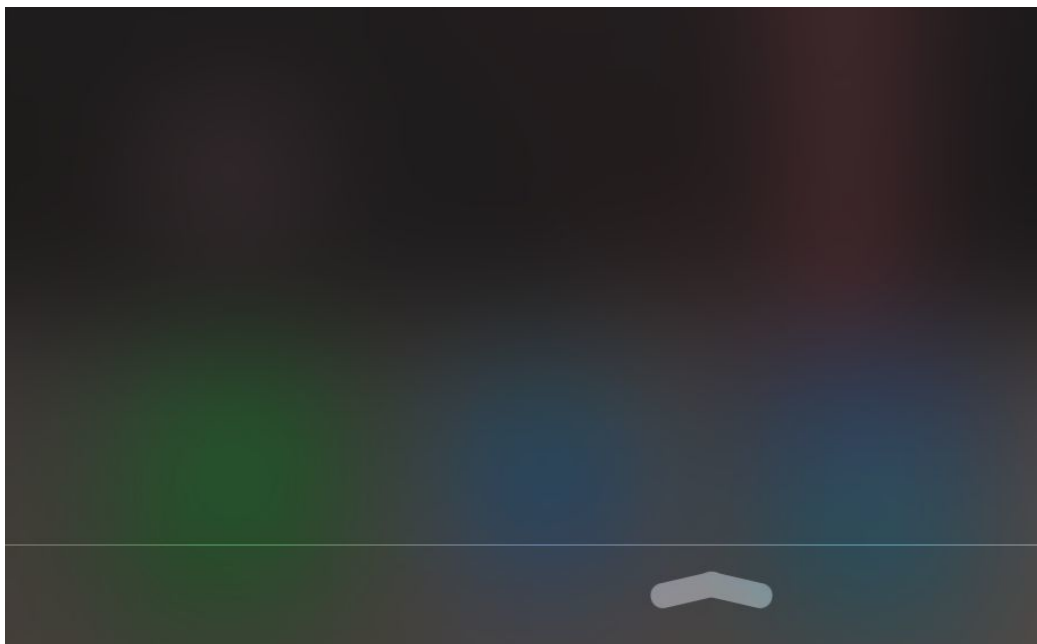
通知

Web

发送时间	内容	类型	iOS 目标   成功
2016-01-21 09:59	今天下雪了，好冷呀！	广播	1   1
2016-01-21 09:58	上海今天下雪了，多穿点衣服！	广播	2   2

手机上的显示：



[好文要顶](#)[关注我](#)[收藏该文](#)[iOS开发群529052159](#)

关注 - 4

粉丝 - 16

[+加关注](#)

0

0

» 下一篇: [iOS多线程的基本使用](#)posted @ 2016-01-26 11:23 [iOS开发群529052159](#) 阅读(1569) 评论(0) [编辑](#) [收藏](#)[刷新评论](#) [刷新页面](#) [返回顶部](#)注册用户登录后才能发表评论, 请 [登录](#) 或 [注册](#), [访问网站首页](#)。

【推荐】50万行VC++源码: 大型组态工控、电力仿真CAD与GIS源码库

【推荐】融云发布 App 社交化白皮书 IM 提升活跃超 8 倍



最新IT新闻:

- 英科学家研发智能绷带 随伤口感染变色
- 贾跃亭壮士断腕: 开启战略收缩 乐视或裁员10%
- 中国移动首发: 一次完整的井盖基站建设全过程
- Ubuntu 17.04路线图公布: 2017年4月13日发布
- 外媒: 阿里YunOS将超iOS 成中国第二大手机操作系统

» 更多新闻...



最新知识库文章:

- 循序渐进地代码重构
  - 技术的正宗与野路子
  - 陈皓：什么是工程师文化？
  - 没那么难，谈CSS的设计模式
  - 程序猿媳妇儿注意事项
- » 更多知识库文章...

Copyright ©2016 iOS开发群529052159