



背景: □□□□□□□□

[日期: 2016-09-28]

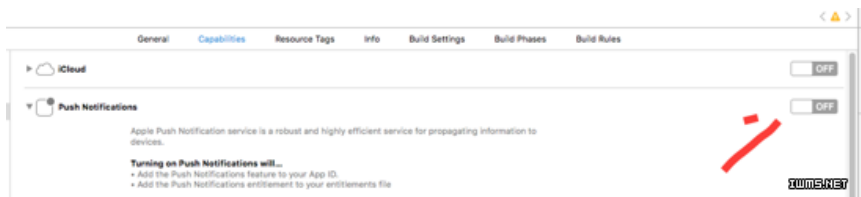
来源: Bison的技术博客 作者:

[字体: 大 中 小]

编辑: Bison

来源：徐不同

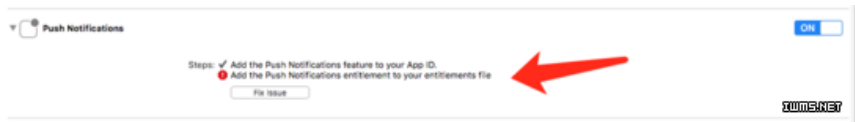
相对简单的推送证书以及环境的问题，我就不在这里讲啦，我在这里说的，是指原有工程的适配。1.首先我们需要打开下面的开关。所有的推送平台，不管是极光还是什么的，要想收到推送，这个是必须打开的哟~



msy.entitlements

Key	Type	Value
Entitlements File	Dictionary (1 item)	
APS.Environment	String	development

第 1 页 (共 7 页)



如果大家点击Fix issue之后，会惊奇的发现，APS Environment由 production 又变成 development 了。

解决办法：我的建议是不做任何修改。

经过我的测试，打包之后，生成的ipa包内，是没有这个.entitlements 文件的。经过测试，我发现是可以正常收到推送信息的。测试的方法如下，大家也可以测试一下。

测试方法：打包之后安装ipa文件，然后利用极光推送，选择生产环境，推送，即可。经过上面的操作，你就会惊奇的发现，推送已经适配完毕了，iOS10的系统，已经可以正常接收通知了。

二、中级篇

这里我会给大家讲一讲iOS10的推送，如何注册，通过什么代理，哪些方法可以用，哪些方法不可以用。

1.系统自带方法

大家不管是使用三方平台的推送，还是系统自带的推送，都先应该了解下系统自带方法，如何实现远程通知的实现。

第一步导入#import <UserNotifications/UserNotifications.h>

且要遵守 的协议，在AppDelegate.m中。

这里需要注意，我们最好写成这种形式

#ifdef NSFoundationVersionNumber_iOS_9_x_Max #import <UserNotifications/UserNotifications.h> #endif 第二步我们需要在 (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions中注册通知，代码如下： - (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions {

```
if ([[UIDevice currentDevice].systemVersion floatValue] >= 10.0) {
    //iOS10特有
    UNUserNotificationCenter *center = [UNUserNotificationCenter currentNotificationCenter];
    // 必须写代理，不然无法监听通知的接收与点击
    center.delegate = self;
    [center requestAuthorizationWithOptions:(UNAuthorizationOptionAlert | UNAuthorizationOptionBadge | UNAuthorizationOptionSound) completionHandler:^(BOOL granted, NSError * _Nullable error) {
        if (granted) {
            // 点击允许
            NSLog(@"注册成功");
            [center getNotificationSettingsWithCompletionHandler:^(UNNotificationSettings * _Nonnull settings) {
                NSLog(@"%@", settings);
            }];
        } else {
            // 点击不允许
            NSLog(@"注册失败");
        }
    }];
}
```

```

    }];
} else if ([[UIDevice currentDevice].systemVersion floatValue] > 8.0) {
    // iOS8 - iOS10
    [application registerUserNotificationSettings:[UIUserNotificationSettings settingsForTypes:UIUserNotificationTypeAlert | UIUserNotificationTypeSound | UIUserNotificationTypeBadge categories:nil]];

} else if ([[UIDevice currentDevice].systemVersion floatValue] < 8.0) {
    // iOS8系统以下
    [application registerForRemoteNotificationTypes:UIRemoteNotificationTypeBadge | UIRemoteNotificationTypeAlert | UIRemoteNotificationTypeSound];
}
// 注册获得device Token
[[UIApplication sharedApplication] registerForRemoteNotifications]; 其中,
获得Device Token的方法是没有改变的。

```

// 获得Device Token - (void)application:(UIApplication *)application didRegisterForRemoteNotificationsWithDeviceToken:(NSData *)deviceToken { NSLog(@"

, [NSString stringWithFormat:@"Device Token:, deviceToken]); } // 获得Device Token失败 - (void)application:(UIApplication *)application didFailToRegisterForRemoteNotificationsWithError:(NSError *)error { NSLog(@"did Fail To Register For Remote Notifications With Error:

, error); }

此次iOS10系统的更新, 苹果给了我们2个代理方法来处理通知的接收和点击事件, 这两个方法在 的协议中, 大家可以查看下。此外, 苹果把本地通知跟远程通知合二为一。区分本地通知跟远程通知的类是UNPushNotificationTrigger.h类中, UNPushNotificationTrigger的类型是新增加的, 通过它, 我们可以得到一些通知的触发条件, 在使用时, 我们不应该直接使用这个类, 应当使用它的子类。

我简单点说

- 1.UNPushNotificationTrigger (远程通知) 远程推送的通知类型
- 2.UNTimeIntervalNotificationTrigger (本地通知) 一定时间之后, 重复或者不重复推送通知。我们可以设置timeInterval (时间间隔) 和repeats (是否重复)。
- 3.UNCalendarNotificationTrigger (本地通知) 一定日期之后, 重复或者不重复推送通知 例如, 你每天8点推送一个通知, 只要dateComponents为8, 如果你想每天8点都推送这个通知, 只要repeats为YES就可以了。
- 4.UNLocationNotificationTrigger (本地通知) 地理位置的一种通知, 当用户进入或离开一个地理区域来通知。在CLRegion标识符必须是唯一的。因为如果相同的标识符来标识不同区域的UNNotificationRequests, 会导致不确定的行为。

接收通知的代码如下:

```

// iOS 10收到通知 - (void)userNotificationCenter:(UNUserNotificationCenter *)center willPresentNotification:(UNNotification *)notification withCompletionHandler:(void (^)(UNNotificationPresentationOptions options))completionHandler{ NSDictionary * userInfo = notification.request.content.userInfo; UNNotificationRequest *request = notification.request; // 收到推送的请求 UNNotificationContent *content = request.content; // 收到推送的消息内容 NSNumber *

```



同声译

跟外国人交流有困难? 您需要带上同声翻译官! 直接对着手机说话, 就能翻译并能说出来, 中译英, 英译中。您能听懂老外的话了, 老外能听懂你的话。接待外宾的神器! 出国旅行的神器!

```
badge = content.badge; // 推送消息的角标 NSString *body = content.body; // 推送消息体 U
NSNotificationSound *sound = content.sound; // 推送消息的声音 NSString *subtitle = content
.subtitle; // 推送消息的副标题 NSString *title = content.title; // 推送消息的标题
```

```
if([notification.request.trigger isKindOfClass:[UNPushNotificationTrigger cla
ss]]) {
    NSLog(@"iOS10 前台收到远程通知:%@", [self logDic:userInfo]);

}
else {
    // 判断为本地通知
    NSLog(@"iOS10 前台收到本地通知:{\\nbody:%@, \\ntitle:%@,\\nsubtitle:
%@,\\nbadge: %@, \\nsound: %@, \\nuserInfo: %@\\n}",body,title,subtitl
e,badge,sound,userInfo);
}
```

```
completionHandler(UNNotificationPresentationOptionBadge|UNNotificationPresent
ationOptionSound|UNNotificationPresentationOptionAlert); // 需要执行这个方法，选
择是否提醒用户，有Badge、Sound、Alert三种类型可以设置 } 下面的代码则是通知的点击事件
:
```

// 通知的点击事件 - (void)userNotificationCenter:(UNUserNotificationCenter *)center didRe
ceiveNotificationResponse:(UNNotificationResponse *)response withCompletionHandler:(voi
d(^)(void))completionHandler{

```
NSDictionary * userInfo = response.notification.request.content.userInfo;
UNNotificationRequest *request = response.notification.request; // 收到推送
的请求
UNNotificationContent *content = request.content; // 收到推送的消息内容
NSNumber *badge = content.badge; // 推送消息的角标
NSString *body = content.body; // 推送消息体
UNNotificationSound *sound = content.sound; // 推送消息的声音
NSString *subtitle = content.subtitle; // 推送消息的副标题
NSString *title = content.title; // 推送消息的标题
if([response.notification.request.trigger isKindOfClass:[UNPushNotification
Trigger class]]) {
    NSLog(@"iOS10 收到远程通知:%@", [self logDic:userInfo]);

}
else {
    // 判断为本地通知
    NSLog(@"iOS10 收到本地通知:{\\nbody:%@, \\ntitle:%@,\\nsubtitle:%@,
\\nbadge: %@, \\nsound: %@, \\nuserInfo: %@\\n}",body,title,subtitle,b
adge,sound,userInfo);
}

// Warning: UNUserNotificationCenter delegate received call to -userNotific
ationCenter:didReceiveNotificationResponse:withCompletionHandler: but the com
pletion handler was never called.
completionHandler(); // 系统要求执行这个方法
```

```
}
```

在点击事件中，如果我们不写completionHandler（）这个方法，可能会报一下的错误，希
望大家注意下~

Warning: UNUserNotificationCenter delegate received call to -userNotificationCenter:didR

ceiveNotificationResponse:withCompletionHandler: but the completion handler was never called.

最后最后，我们要大家补充一下，旧版本的一些方法，方便大家扩充iOS10的通知的通知，不影响原有逻辑。

- (void)application:(UIApplication *)application didReceiveRemoteNotification:(NSDictionary *)userInfo { NSLog(@"iOS6及以下系统，收到通知:");
[self logDic:userInfo]; }
- (void)application:(UIApplication *)application didReceiveRemoteNotification:(NSDictionary *)userInfo fetchCompletionHandler: (void (^)(UIBackgroundFetchResult))completionHandler {
NSLog(@"iOS7及以上系统，收到通知:");
[self logDic:userInfo]; completionHandler(UIBackgroundFetchResultNewData); }

2.极光推送（需要下载最新的版本）

如果用到三方的一些平台，做推送就会更为简单。

1.注册通知的代码如下

```
if ([[UIDevice currentDevice].systemVersion floatValue] >= 10.0) { #ifdef NSFoundationVersionNumber_iOS_9_x_Max JPUSHRegisterEntity * entity = [[JPUSHRegisterEntity alloc] init]; entity.types = UNAuthorizationOptionAlert|UNAuthorizationOptionBadge|UNAuthorizationOptionSound; [JPUSHService registerForRemoteNotificationConfig:entity delegate:self]; #endif } else 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]; }
```

注册完成之后，我们则需要加入极光推送更新后，新加入的2个方法,这两个方法在代理方法中。

```
/* * @brief handle UserNotifications.framework [willPresentNotification:withCompletionHandler:] * @param center [UNUserNotificationCenter currentNotificationCenter] 新特性用户通知中心 * @param notification 前台得到的通知对象 * @param completionHandler 该callback 中的options 请使用UNNotificationPresentationOptions / - (void)jpushNotificationCenter:(UNUserNotificationCenter *)center willPresentNotification:(UNNotification *)notification withCompletionHandler:(void (^)(NSInteger options))completionHandler; /* * @brief handle UserNotifications.framework [didReceiveNotificationResponse:withCompletionHandler:] * @param center [UNUserNotificationCenter currentNotificationCenter] 新特性用户通知中心 * @param response 通知响应对象 * @param completionHandler * / - (void)jpushNotificationCenter:(UNUserNotificationCenter *)center didReceiveNotificationResponse:(UNNotificationResponse *)response withCompletionHandler:(void (^)(void))completionHandler;
```

使用时，只需要在上面的代码中添加极光的处理方法就可以了，具体使用如下图：

```
if([response.notification.request.trigger isKindOfClass:[UNPushNotificationTrigger class]]) {  
// 这个方法，不管是收到通知代理还是点击通知的代理，如果使用极光推送，我们都是需要增
```


加这个方法的。[JPUSHService handleRemoteNotification:userInfo];

```

NSLog(@"iOS10 收到远程通知:%@", [self logDic:userInfo]);
[rootViewController addNotificationCount];
}
else {
    // 判断为本地通知
    NSLog(@"iOS10 收到本地通知:{\\nbody:%@, \\ntitle:%@, \\nsubtitle:%@, \\nbadge: %@, \\nsound: %@, \\nuserInfo: %@\\n}", body, title, subtitle, badge, sound, userInfo);
}

```

通过上面的文章，相信大家已经可以初步了解新版本的推送，要如何处理啦~

稍后,我会更新以下内容，希望大家支持： UNNotificationContentExtension - 通知内容扩展 UNNotificationServiceExtension- 通知服务扩展 UNNotificationAction - 通知响应的方法 U
NSErrorCode - 通知错误 UNNotificationAttachment - 附件通知 稍后我还会上传demo路径。



我的PM2.5
扫二维码下载APP

随时随地告诉你最精确的PM2.5值！走南闯北，随时查看，PM2.5值随你而变！不是城市的平均值，是用离你最近的数个PM2.5传感器，在云计算平台精确计算出来的实时值。还可接入酷炫的“我的PM2.5”，配套智能硬件，实时监测室内空气质量。

0

[顶一下](#)

分享 赶快成为第一个分享的人吧

收藏 推荐 打印 | 录入: Cstor | 阅读: 106 次

MySQL多线程复制问题处理之Error_code: 1872

多端平台发布：实时同步的 Realm 和完全开源的数据库实现

科技头条



扫二维码下载APP
(目前仅支持Android)

看新闻，发广告：每次评论均显示您的广告词和链接！

科技头条——科技爱好者的情报站，是一款挖掘互联网科技类新闻、微信公众账号热点文章的阅读客户端。酷炫科技，远见未来。精选微信朋友圈100000+的热点文章、知名微信公众账号，每日智能推送您关心的内容，帮助您更加高效地获取科技信息、追踪科技热点。



中国存储资讯



云创大数据



中国智慧城市资



大数据资讯



中国_物联网



中国云计算论坛



中国PM2.5资讯



专注云计算



笑看云存储

一键关注

注册微博

相关新闻 iOS

iOS应用UI线程卡顿监控 (今 09:13)

用Python为iOS10生成图标和截屏 (09月22日)

iOS 开发：14个技巧助你适配 (09月19日)

iOS函数响应式编程以及ReactiveCoc (08月12日)

iOS 中的 block 是如何持有对象的 (08月10日)

本文评论 查看全部评论 (0)

表情: 姓名: ☒ 匿名 字数 0

点评:

☒ 同意评论声明

评论声明

- 尊重网上道德，遵守中华人民共和国的各项有关法律法规
- 承担一切因您的行为而直接或间接导致的民事或刑事法律责任
- 本站管理人员有权保留或删除其管辖留言中的任意内容
- 本站有权在网站内转载或引用您的评论
- 参与本评论即表明您已经阅读并接受上述条款

版权所有:© 2013-2016 南京云创大数据科技股份有限公司 (股票代码: 835305), 保留一切权利。苏ICP备11060547号-4

本站所转载、链接、引用及翻译的内容等, 其版权属于各自合法的拥有者。

本站创建、管理、商务合作: zhujiguangjie#163.com(用@代替#)



Powered by iwms 6.0