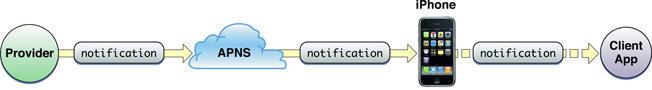
# 消息推送

**APNS的推送机制**

首先我们看一下苹果官方给出的对IOS推送机制的解释。如下图



Provider就是我们自己程序的后台服务器，APNS是Apple Push Notification Service的缩写，也就是苹果的推送服务器。

上图可以分为三个阶段：

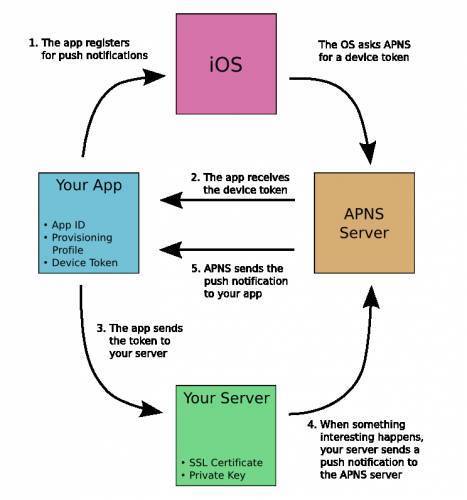
第一阶段：应用程序的服务器端把要发送的消息、目的iPhone的标识打包，发给APNS。

第二阶段：APNS在自身的已注册Push服务的iPhone列表中，查找有相应标识的iPhone，并把消息发送到iPhone。

第三阶段：iPhone把发来的消息传递给相应的应用程序，并且按照设定弹出Push通知。

**APNS推送通知的详细工作流程**

下面这张图是说明APNS推送通知的详细工作流程：



根据图片我们可以概括一下：

1、应用程序注册APNS消息推送。

2、IOS从APNS Server获取devicetoken，应用程序接收device token。

3、应用程序将device token发送给程序的PUSH服务端程序。

4、服务端程序向APNS服务发送消息。

5、APNS服务将消息发送给iPhone应用程序。

客户端实现起来相对容易，但后台要支持APNS的话还是比较复杂的，因此，我使用口碑较好的第三方推送-极光推送。

**准备工作**

1. 申请推送证书

http://www.2cto.com/kf/201604/500620.html

1. 将推送证书导出为p12文件，并上传至极光推送
2. 导入JPUSH框架

xcode 8以下 建议手动导入JPUSH，由于缺少系统库UserNotifications.framework会导致框架无法使用

* 在极光官网下载[最新SDK](http://docs.jiguang.cn/jpush/resources/)
* 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JPUSHService.h、jpush-ios-x.x.x.a、jcore-ios-x.x.x.a）添加到你的工程目录中。
* 添加Framework
  + CFNetwork.framework
  + CoreFoundation.framework
  + CoreTelephony.framework
  + SystemConfiguration.framework
  + CoreGraphics.framework
  + Foundation.framework
  + UIKit.framework
  + Security.framework
  + libz.tbd (Xcode7以下版本是libz.dylib)
  + AdSupport.framework (获取IDFA需要；如果不使用IDFA，请不要添加)
  + UserNotifications.framework (Xcode8及以上)
  + libresolv.tbd (JPush 2.2.0及以上版本需要, Xcode7以下版本是libresolv.dylib)
* 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。

**允许Xcode7支持Http传输方法**

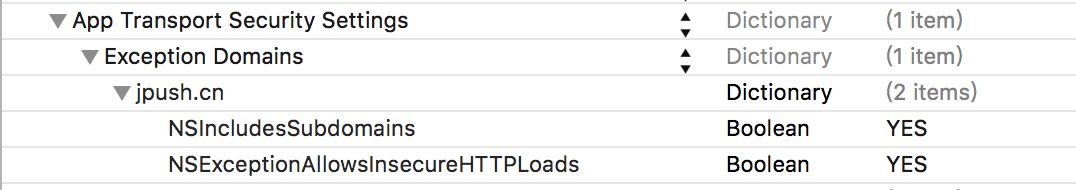
如果您使用的是2.1.9及以上的版本则不需要配置此步骤

如果用的是Xcode7或更新版本，需要在App项目的plist手动配置下key和值以支持http传输:

**选择1：根据域名配置**

* 在项目的info.plist中添加一个Key：NSAppTransportSecurity，类型为字典类型。
* 然后给它添加一个NSExceptionDomains，类型为字典类型；
* 把需要的支持的域添加給NSExceptionDomains。其中jpush.cn作为Key，类型为字典类型。
* 每个域下面需要设置2个属性：NSIncludesSubdomains、NSExceptionAllowsInsecureHTTPLoads。 两个属性均为Boolean类型，值分别为YES、YES。

如图：



**选择2：全局配置**

<key>NSAppTransportSecurity</key>

<dict>

<key>NSAllowsArbitraryLoads</key>

<true/>

</dict>

**代码集成**

初始化JPUSH, types属性设置收到通知时，APP的一些相应方式，包括bageValue,提示音或者震动，来提示用户接收到通知。在XCODE8以下，由于缺少UserNotifications.framework框架，所以无法使用IOS的特性，而极光的代理属性洽洽需要配合UserNotifications才能使用，因此此处传nil即可。

appKey:极光开发者后台提供，你可以通过注册账号并创建应用获得

channel:通常为Publish channel

func setPushNoti(launchOptions:[NSObject: AnyObject]?){

//setup apns

let entity = JPUSHRegisterEntity()

entity.types = (1 << 0) | (1 << 1) | (1 << 2)

JPUSHService.registerForRemoteNotificationConfig(entity, delegate: nil)

//set jpush

JPUSHService.setupWithOption(launchOptions, appKey: K\_APP\_KEY, channel: K\_CHANNEL, apsForProduction: false)

setJPUSH()

}

依次添加以下代码

func application(application: UIApplication, didRegisterForRemoteNotificationsWithDeviceToken deviceToken: NSData){

JPUSHService.registerDeviceToken(deviceToken)

}//将苹果通知中心的deviceToken返回给极光推送后台

func application(application: UIApplication, didFailToRegisterForRemoteNotificationsWithError error: NSError){

print("注册apns失败 \n \(error.localizedDescription)")

}//当然了，你需要关键的LOG

func application(application: UIApplication, didReceiveRemoteNotification userInfo: [NSObject : AnyObject], fetchCompletionHandler completionHandler: (UIBackgroundFetchResult) -> Void) {

JPUSHService.handleRemoteNotification(userInfo)

completionHandler(UIBackgroundFetchResult.NewData)

print("ios7, 及以上系统收到通知\(userInfo)")

urlString = userInfo["url"] as? String

if urlString != nil && (application.applicationState == .Active){

NSNotificationCenter.defaultCenter().postNotification(NSNotification(name: "NewVersion", object: urlString, userInfo: nil))

}

}//XCODE7 的环境下，消息的解析在这里实现。

至此IOS的推送的集成基本完成了。但是要做到接收通知，打开通知栏就打开一个URL，还需要进一步的研究。（推送的消息中，content-alive字段必须为true否则将无法从服务器获取通知的内容）

1. 当程序进程被杀死，通过通知栏启动程序（注意，如果收到通知，仍然通过ICON去启动程序，通知的内容是无法获取到的，必须通过通知才可以）

func application(application: UIApplication, didFinishLaunchingWithOptions launchOptions: [NSObject: AnyObject]?) -> Bool {

print("launchOptions ----",launchOptions)

if launchOptions != nil{

let remoteNoti = launchOptions![UIApplicationLaunchOptionsRemoteNotificationKey]

urlString = remoteNoti!["url"] as? String

}

}//获取通知内容

2, 当程序 处于后台，程序可以正常通过通知进入，并触发一个特定的操作

3. 当程序处于前台

ios 10以下，是没有什么反应的，不弹出横幅，也无通知的声音。因此，可以通过添加这么一个判断来触发界面提示更新

if urlString != nil && (application.applicationState == .Active){

NSNotificationCenter.defaultCenter().postNotification(NSNotification(name: "NewVersion", object: urlString, userInfo: nil))

}

IOS10 及以上则既可以弹出横幅又可以有声音，相关的逻辑可以在JPUSH的代理方法里去实现

**关于分组推送和精确推送**

极光推送里面的精准推送是通过idfa去实现的，但前提是app里面需要有明显的广告功能，如果没有的话，app将无法上架，再者，这个东西也不是完全的可靠，如果用户在设置里面停止广告跟踪的话，精准推送还是会失效的。

思路：jpush中又一个registuserID这个id有个特点，就是程序如果被卸载，则重新生成一个新的来唯一表示设备，我们可以将用户ID作为key registuserID为value，如果重新安装，只需更新这个registuserID的值，即可唯一表示设备

分组推送

思路：先给用于一个别名alias，再给这个用户打上他所具备的标签，如 alias:liming, tag1:汽车 tag2:游戏 tag3：旅游

laias:jone tag1:美食 tag2:游戏 tag3：电影

推送的时候，可以通过对tag的交并补等集合操作，实现对不同群体用户的信息的分类推送，如游戏这条推送，liming和jone都能收到