

在移动应用中，推送已经成为不可缺少的重要功能。本文档主要介绍有关 iOS 推送的基础知识，并从几个典型问题出发，分析如何解决在实现推送中出现的一些问题。

本地通知和远程通知简介

在 iOS 设备上（模拟器无法使用推送），系统收到通知后这样处理：

- 在屏幕上弹出一些选项，或者在屏幕顶部显示横幅（banner）如下图左
- App 的角标数值发生变化，具体表现为 App icon 右上角的小红点及数字，如邮件中的红点
- 伴随推送消息的提示声音

当应用处于前台运行时，系统是不会在屏幕上显示通知，但是仍会调用相应的 API。

只有真机可以使用推送功能。

用户可以设置每一个 App 的通知权限，如下图最后一个：



用户可以选择关闭某个应用的推送功能。还可以设置通知是否在通知中心显示、通知到达时是否发出声音、通知能否改变 App 角标以及锁屏时是否显示该 App 的通知，还可以设置通知到达时的提醒样式。当然，这些都可以分别对每一个应用进行单独设置。

本地通知

本地通知是一种基于时间的提醒方式。

本地通知最多向系统注册 64 个，当超过这个数量后，最早注册的本地通知会被丢弃。

本地通知在 iOS 设备上的显示与远程推送通知一样。

远程通知

iOS App 运行在后台时，无法主动进行网络连接。

远程通知可以用来提醒用户。发送远程通知时，服务器首先需要使用推送证书与 APNs（Apple Push Notification service）建立安全连接，然后将消息传递给它。当 APNs 收到消息后，会通过与手机之间的长连接下发到对应的手机上，然后 iOS 弹出这条消息来提醒用户。

用户看通过点击或滑动通知来运行 App。可通过程序中相应的方法可以获取通知信息，然后做相应的逻辑处理。

如果不是通过通知启动 App，那么无法在程序中获取通知信息。例如通过点击应用图标启动 App。

推送通知都包含 Payload：一个 Apple 已经定义好的属性列表，操作系统根据它决定使用哪一种方式来提醒用户。还可以在 Payload 中加入一些自定义数据。

通知并不能一定到达。如果在用户无法收到推送通知时（关机或网络不可用），这种情况下 APNs 收到了多条发往这台设备的通知，那么只会保留最后一条，最早的会被丢弃。

APNs 首先通过移动蜂窝网络发送通知，只有当移动蜂窝网络不可用时才使用 Wi-Fi。

通知的两种推送环境

在使用 iOS 远程推送功能时，有两种不同的环境。开发环境（Development）以及生产环境（Production）。

App 当前使用的推送环境与 Xcode - Build Settings - Code Signing - Provisioning Profile 文件的模式一致。

证书与证书校验

与 APNs 之间是加密的连接，因此需要使用证书来加密连接。每个的推送环境有自己单独的推送证书，即开发证书和生产证书。

在将证书最终转为 pem 格式后，可通过与 APNs 连接来测试证书是否有效。

开发环境：

```
openssl s_client -connect gateway.sandbox.push.apple.com:2195 -cert MyApnsDev.pem
```

生产环境：

```
openssl s_client -connect gateway.push.apple.com:2195 -cert MyApnsPro.pem
```

当输入完命令回车后，终端首先会输出很多相关信息。

当连接建立失败时，会直接 close 掉。

当连接建立成功时，终端会停止输出，并等待你输入，你可以随便输入一些字符后摁回车，然后连接才会关闭。

以上命令在 Mac 下没有问题，在其他操作系统下需要指定服务器当前的 CA 根证书，具体可以从网站下载：

[entrust_2048_ca.cer 下载](#)

下载完成之后，在以上命令后面加上 -CAfile entrust2048ca.cer 即可

DeviceToken

通知需推送到具体某一台设备，而 DeviceToken 就是这台设备的标识符。在向 APNs 发送推送通知时，需要使用 DeviceToken 来指定这条通知将要到达的设备。

每一台设备，不同的推送环境下分别有一个 DeviceToken。即 DeviceToken 也分开发环境和生产环境。

应用在向 APNs 注册推送通知时，会根据当前 Xcode 工程中 target 对应的 Provisioning Profile 决定请求对应环境的 DeviceToken。即系统返回的 DeviceToken 其环境与 Provisioning Profile 的环境对应。

一般在使用 Xcode 开发时，Provisioning Profile 为 Development，因此获取的 DeviceToken 也是开发模式。

在 App 需要打包为 ipa 文件或者需要上传到 AppStore 时，会将 Provisioning Profile 文件修改为 Distribution，这时获取到的 DeviceToken 是生产模式。

在应用程序中注册远程推送功能

App 必须要向 APNs 请求注册以实现推送功能，在请求成功后，APNs 会返回一个设备的标识符即 DeviceToken 给 App，服务器在推送通知的时候需要指定推送通知目的设备的 DeviceToken。在 iOS 8 以及之后，注册推送服务主要分为四个步骤：

1. 使用 `registerUserNotificationSettings:` 注册应用程序想要支持的推送类型
2. 通过调用 `registerForRemoteNotifications` 方法向 APNs 注册推送功能
3. 请求成功时，系统会在应用程序委托方法中返回 DeviceToken，请求失败时，也会在对应的委托方法中给出请求失败的原因。
4. 将 DeviceToken 上传到服务器，服务器在推送时使用。

上述第一个步骤注册的 API 是 iOS 8 新增的，因此在 iOS 7，前两个步骤需更改为 iOS 7 中的 API。

DeviceToken 有可能会更改，因此需要在程序每次启动时都去注册并且上传到你的服务器端。

“

注意：iOS 设备与 APNs 需要建立一条长连接，之后的推送注册以及推送获取都需要通过这条长连接。

当 iOS 设备无网络可用时，在向 APNs 请求注册后，既不会有注册成功的回调，也不会有注册失败的回调。如果发生这种情况，需要检查设备的网络连接。

开发环境和生产环境建立不同的长连接，且设备上至少要有一个开发环境的 App 并且注册推送，才会建立开发环境下的长连接。

当已经注册过推送服务后，以后系统会立刻返回给你 DeviceToken。还有一点是，返回 DeviceToken 的回调方法并不一定是只有在你注册或者再次注册时才被系统调用，当 DeviceToken 改变时也会被系统直接调用。

在程序中处理通知

我们来看一下当系统传递本地或远程通知的时候都有哪些情况。

通知到达时，应用不在前台。

这种情况下，操作系统将呈现通知，弹出一个选项或者在屏幕上部显示 banner 提醒，给 App 角标设置对应的数字，可能播放提示音，当用户稍微往下滑动通知时显示更多的 action 按钮（如果有设置）。

通知到达时，应用在前台运行。

这种情况下，不会弹出 banner。系统会根据当前的通知是本地通知还是远程通知，调用不同的方法，并且在方法中传递通知的 Payload 数据。

在 iOS8 的通知中，用户点击通知上的自定义 action 按钮进入应用。

这种情况下，系统会调用 iOS8 中对 action 按钮新增的处理方法。

用户点击通知进入应用。

系统会根据当前的通知是本地通知还是远程通知，调用不同的方法，并且在方法中传递通知的 Payload 数据。

“

注意：只有当通过通知进入到 App 时，才可以获取到通知信息。

自定义通知提示音

你可以在 App 的 Bundle 中加入一段自定义提示音文件。然后当通知到达时可以指定播放这个文件。必须为以下几种数据格式：

- Linear PCM
- MA4 (IMA/ADPCM)
- μLaw
- aLaw

你可以将它们打包为 **aiff**、**wav** 或 **caf** 文件。自定义的声音文件时间必须小于 30 秒，如果超过了这个时间，将被系统声音代替。

Payload

Payload 是通知的一部分，每一条推送通知都包含一个 Payload。它包含了系统提醒用户通知到达的方式，还可以添加自定义的数据。即通知主要传递的数据为 Payload。

Payload 本身为 JSON 格式的字符串，它内部必须要包含一个键为 **aps** 的字典。aps 中可以包含以下字段中的一个或多个：

- alert：其内容可以为字符串或者字典，如果是字符串，那么将会在通知中显示这条内容
- badge：其值为数字，表示当通知到达设备时，应用的角标变为多少。如果没有使用这个字段，那么应用的角标将不会改变。设置为 0 时，会清除应用的角标。
- sound：指定通知展现时伴随的提醒音文件名。如果找不到指定的文件或者值为 default，那么默认的系统音将会被使用。如果为空，那么将没有声音。
- content-available：此字段为 iOS 7 silent remote notification 使用。不使用此功能时无需包含此字段。

百度云推送故障排查指南

使用百度云推送时，可能出现一些问题，你可以根据下面的项目进行排查。

使用百度云推送实现 iOS 推送主要分为三步：

- 制作推送证书。需要分别制作开发环境和生产证书两个证书。然后测试证书是否正确创建并导出。在制作证书的过程中，会为 App 开启推送功能。
- 在百度云推送官网控制台中创建应用。需要为安卓版本和 iOS 版本分别创建应用，然后为 iOS 应用上传已经制作成功的 pem 格式推送证书。
- 在 App 中集成百度云推送 SDK，按照文档中的步骤即可。集成完成后，App 与百度云推送服务器做一次绑定，App 收到绑定成功的回复后，保存回复字典中的 channelId，它是设备的唯一标识。userId 暂时不被使用。

完成以上步骤即可在官网的控制台中对安装你应用的 iOS 设备进行推送。如果还需服务端推送的支持，按照文档集成服务端推送 SDK 即可。

注册远程推送通知

在获取推送消息之前，需要先向 Apple Push Notification service (APNs) 注册推送服务。

1. 应用程序调用注册推送服务的 API，需要注意 iOS 8 之前和之后需要使用不同的 API。
2. 应用程序实现 **UIApplicationDelegate** 委托的 **application:didRegisterForRemoteNotificationsWithDeviceToken:** 回调方法，用于获取 APNs 返回给设备的 DeviceToken。
3. 应用程序实现 **UIApplicationDelegate** 委托的 **application:didFailToRegisterForRemoteNotificationsWithError:** 回调方法，用于当获取不到 DeviceToken 时，查看其错误原因。

然后应用程序将 DeviceToken 传递给云推送。

注册推送服务比较简单，但是仍然有几点比较重要你需要注意。

注册百度云推送服务

在注册百度云推送服务时，一定要根据当前应用所处的环境，正确设置注册方法的 PushMode 参数：

```
(void)registerChannel:(NSDictionary *)launchOptions apiKey:(NSString *)apikey pushMode:(BPushMode)mode withFirstAction:(NSString *)leftAction withSecondAction:(NSString *)rightAction withCategory:(NSString *)category isDebug:(BOOL)isdebug;
```

在开发时使用 Development 模式，在上线前一定需要修改为 Production 模式。如果上面方法的 PushMode 设置错误，那么无法收到通知。

如果在开发时发现设置错误，那么在修改模式之后，需要做一次解除绑定操作再进行绑定。即 PushMode 参数每一次修改后，都需要调用一次 unBind 方法来清除缓存。

“

requestid 为 88888888 时，表示此次请求的回复是从缓存中获取的。在修改 PushMode 参数后，需要调用 unbind 方法或删除应用重新安装来清除缓存。

没有委托回调

当含有推送功能的 App 第一次安装时，iOS 尝试与 APNs 建立一条长连接供所有操作系统上具有推送能力的 app 使用。如果

application:didRegisterForRemoteNotificationsWithDeviceToken 和 application:didFailToRegisterForRemoteNotificationsWithError 这两条回调方法都没有被调用。这意味着 iOS 设备与 APNs 之间没有成功建立起长连接。

这可能没有任何错误提示。系统可能由于蜂窝网络无信号或距离 Wi-Fi 点太远，或者开启了飞行模式。

要注意到网络状态是经常变化的，一旦长连接建立起来了，之前提到的回调方法就会被调用。

在 iOS 上，推送通知首先检查移动蜂窝网络是否能正常使用，即使当前已经连接了 Wi-Fi，只有当移动蜂窝网络无法使用时才会使用 Wi-Fi。

如果你的设备支持移动网络，那么将检查它是否有一个活动的蜂窝数据计划。在设备的【设置】中，关闭 Wi-Fi 选项，然后查看是否能够使用 Safari 浏览器打开网页等。另一方面，如果推送服务使用 Wi-Fi 服务，那么需要确保 Wi-Fi 热点防火墙允许 端口为 5223 的 TCP 通信。

“

注意：不同的推送环境使用单独的长连接，操作系统建立与 Sandbox 环境的长连接用于开发。ad hoc 和生产模式会连接到生产环境下。

回调错误的 aps-environment：需重新生成 provisioning profile 文件

支持推送服务的 iOS App 在第一次运行时，iOS 会询问用户是否想要接收这个 app 的推送通知。如果这个 App 之前没有提供接收推送通知的选项，那么它将失去 aps-environment 代码签名授权的机会。这种资格是在 building app 的时候从 provisioning profile 中得到，它控制了为了接收推送通知时，app 应该使用哪一种推送环境去建立长连接。

如果 iOS app 运行在设备上，在获取推送服务时发现代码签名授权获取失败，则 application:didFailToRegisterForRemoteNotificationsWithError 方法会被调用。如果你没有实现这个方法，那么你应该使用它来检查你的 aps-environment 授权是否在运行时获取到。

如果没有发现对应的代码签名授权，那么说明你还没有配置 app ID 开启推送服务，或者开启后没有更新 Provisioning Profile 文件，需要去 Member Center 去更新。

你需要在 Member Center 中为 App ID 配置开启推送服务后重新创建 provisioning profile，如果你 Xcode 中的 provisioning profile 是未配置前的，那么需要将其删除后重新生成新的并添加到 Xcode。

更新完这些后，你需要 rebuild 你的 app。

注册成功但收不到通知

如果你的 app 注册推送服务成功，并且能获取到 DeviceToken，但是收不到通知。那么有可能是你设备的问题，也有可能是发送通知的服务器的的问题。下面简述了如何排查这些问题。

设备有可能断掉了与 APNs 的长连接且不能重新建立，因而无法通过长连接将通知发送到你的 iOS 设备上。尝试关闭 app 并且重启设备看是否解决了问题。（在 iOS4 之后，app 可以后台运行，因此你需要强制杀死 app）。如果重启后 app 的注册服务不能完成，即获取不到 DeviceToken，那么 iOS 设备还是不能与 APNs 建立长连接。你需要按照之前章节中介绍的内容再次排查问题。

如果可以获取 DeviceToken，你需要上传对应的 DeviceToken 到百度云推送，使用云推送 SDK 中的 RegisterDeviceToken 方法即可。在 App 每次启动注册推送服务的时候，它总是会向 APNs 请求 DeviceToken，所以你不需要保存 DeviceToken 并且尝试复用它，因为它有时候会改变。

“

注意：系统每次启动都会通过 application:didRegisterForRemoteNotificationsWithDeviceToken 方法返回设备的 DeviceToken，你需要在这个方法中调用云推送的 RegisterDeviceToken 方法。

只有部分通知到达

你如果在很短的一段时间内，发送大量通知到同一台设备，那么 Apple 的 Push 服务将只会发送最后一个通知。原因如下，设备收到的每一条通知后都会进行确认。在 APNs 收到这个确认信息前，它只能假设设备已经离线，然后将通知存储在质量服务（QoS）队列中并尝试再次发送。往返的网络延迟是一个主要原因。

QoS 队列为每一台设备中每一个 App 仅保留一条通知，如果保留的这条通知还没发出去时又收到了一条新的通知，那么旧的通知将被丢弃。

所有的一切指出，推送通知是提示 App 它所感兴趣的一些事情已经改变，通知不应该包含数据，因为它并不是无论如何都能到达的，并且通知也不应

该是具有状态性的。

你的设备无法连接到 APNs 时，任何通知都不能立刻发送成功，需要入队等待以后再次发送。需要考虑网络延迟，最长的情况下 60 秒后 APNs 会认为是超时的。

单播收不到

在使用百度云推送进行单播推送时，如果一直收不到。首先检查手机的网络连接是否正常。

确认你当前的推送环境，具体可参照推送环境一项。针对不同的推送环境，需要在百度云推送 SDK 中的

```
+ (void)registerChannel:(NSDictionary *)launchOptions apiKey:(NSString *)apikey pushMode:(BPushMode)mode  
withFirstAction:(NSString *)leftAction withSecondAction:(NSString *)rightAction withCategory:(NSString *)category  
isDebug:(BOOL)isdebug; 方法中，修改 BPushMode为正确的环境。
```

卸载应用并且重新创建，根据绑定成功后返回的 channelId 再次进行单播推送。

广播收不到

在测试广播推送时，如果有部分能收到，那么你的推送服务配置是没有问题的。对于这部分没有收到的设备，首先检查其网络连接，然后尝试对这部分设备推单播，单播收不到时可以按照上面的方法进行排查。

如果广播中所有的用户都无法收到，可以尝试对其中部分用户推送单播，然后按照单播推送的排查方法解决。

APNs 与 iOS 设备之间的长连接

在 iOS 推送中，通知是通过 Apple 的 APNs 与设备之间的长连接进行推送的。开发环境和生产环境对应不同的长连接，设备上至少有一个开发环境的 App，并且这个 App 注册推送服务后，设备才会与 APNs 建立开发环境下的长连接。