

当前位置： 首页 > 编程社区 > Swift > Swift - JPush极光推送的使用7（发送自定义消息）

## Swift - JPush极光推送的使用7（发送自定义消息）

发布：hangge 阅读：1005

2016-07-18 08:50

我之前的一系列文章讲的都是发送通知的相关内容。JPush极光推送除了可以推送通知，还可以用来发送自定义消息。本文接着

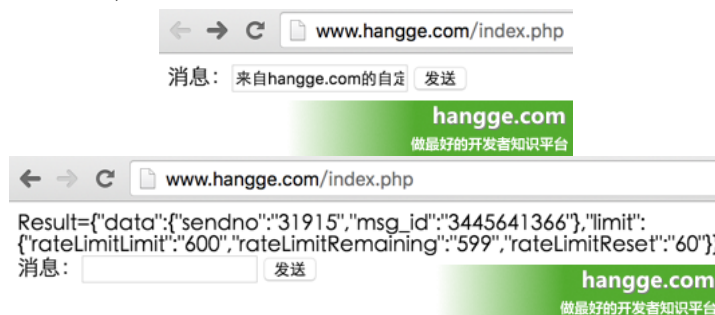
航歌 - 做最好的开发者知识平台

### 1，发送自定义消息与发送通知的异同

- (1) 客户端**App**只有在运行的时候才能收到自定义消息。而通知不同，不管客户端是否在运行都是能够收到推送过来的通知。
- (2) 发送自定义消息的话不需要通过 **APNs**，但相较于通知，可以发送更多的内容（当然还是有长度限制的）。
- (3) 虽然**App**退出后就没法收到自定义消息，但 **JPush** 服务器这时会将其保存成离线消息（具体保留时长可以设置）。当**App**启动后，会自动获取到这条离线消息。
- (4) 由于发送自定义消息不需要通过 **APNs**，且客户端在运行时才能接收。所以比较适合用在在线聊天室、即时通讯等相关应用上。
- (5) 同发送通知一样，发送自定义消息也可以根据别名、标签发送给指定用户，或者广播的形式发给所有人。
- (6) 自定义消息同样可以定时发送。

### 2，自定义消息的样例

- (1) 服务端页面上填写消息内容，点击“发送”。即可将自定义消息发送到客户端。



- (2) 客户端如果是运行状态，则会接收到并进行下一步处理（这里直接将收到的自定义消息弹出显示。）如果客户端此刻没有运行，等客户端下次运行时也会收到之前发送的这条信息。



## 文章类别

首页

编程社区

Swift

Flex / Flash

首页

留言

React / RN

其他

奇文共赏



## 本类相关

Swift - 使用反射将自定义对象数据序列化成JSON

Swift - Core Graphics绘图框架详解4（绘制图片、

Swift - 动态添加删除UITableView的单元格（以及内

Swift - 使用ijkplayer框架播放直播视频2（RTSP视

Swift - 从ALAsset中获取照片的原图并转换成NSC

Swift - 文本标签（UILabel）的用法

Swift - 邮件发送功能的实现

Swift - 多线程实现方式（1） - Thread

hangge.com通知 - 今日起开始休假，我们来年再！

Swift - 内存泄露原因（循环强引用）及解决办法

### 3, 样例代码

#### (1) 客户端代码: AppDelegate.swift

```
1  import UIKit
2
3  @UIApplicationMain
4  class AppDelegate: UIResponder, UIApplicationDelegate {
5
6      var window: UIWindow?
7
8      func application(_ application: UIApplication,
9                      didFinishLaunchingWithOptions
10                      launchOptions: [UIApplicationLaunchOptionsKey: Any]?) -> Bool {
11
12          //通知类型（这里将声音、消息、提醒角标都给加上）
13          let userSettings = UIUserNotificationSettings(types: [.alert, .badge, .sound],
14                                                         categories: nil)
15          if ((UIDevice.current.systemVersion as NSString).floatValue >= 8.0) {
16              //可以添加自定义categories
17              JPUSHService.register(forRemoteNotificationTypes: userSettings.types.rawValue,
18                                  categories: nil)
19          }
20          else {
21              //categories 必须为nil
22              JPUSHService.register(forRemoteNotificationTypes: userSettings.types.rawValue,
23                                  categories: nil)
24          }
25
26          // 启动JPushSDK
27          JPUSHService.setup(withOption: nil, appKey: "7b528331738ec719195798fd",
28                             channel: "Publish Channel", apsForProduction: false)
29
30          //监听自定义消息的接收
31          let defaultCenter = NotificationCenter.default
32          defaultCenter.addObserver(self, selector: #selector(networkDidReceiveMessage(notification:)),
33                                   name: Notification.Name.jpjfNetworkDidReceiveMessage, object: nil)
34
35          return true
36      }
37
38      //收到自定义消息
39      func networkDidReceiveMessage(notification: Notification){
40          var userInfo = notification.userInfo!
41          //获取推送内容
42          let content = userInfo["content"] as! String
43          //获取服务端传递的Extras附加字段, key是自己定义的
44          //let extras = userInfo["extras"] as! NSDictionary
45          //let value1 = extras["key1"] as! String
46
47          //显示获取到的数据
48          let alertController = UIAlertController(title: "收到自定义消息",
49                                                  message: content,
50                                                  preferredStyle: .alert)
51          alertController.addAction(UIAlertAction(title: "确定", style: .cancel, handler: nil))
52          self.window?.rootViewController!.present(alertController, animated: true, completion: nil)
53      }
54
55
56      func application(_ application: UIApplication,
57                      didRegisterForRemoteNotificationsWithDeviceToken deviceToken: Data) {
58          //注册 DeviceToken
59          JPUSHService.registerDeviceToken(deviceToken)
60      }
61
62      func application(_ application: UIApplication,
63                      didReceiveRemoteNotification userInfo: [AnyHashable: Any],
64                      fetchCompletionHandler
65                      completionHandler: @escaping (UIBackgroundFetchResult) -> Void) {
66          //增加IOS 7的支持
67          JPUSHService.handleRemoteNotification(userInfo)
68          completionHandler(UIBackgroundFetchResult.newData)
69      }
70
71      func application(_ application: UIApplication,
72                      didFailToRegisterForRemoteNotificationsWithError error: Error) {
```



```

73         //可选
74         NSLog("did Fail To Register For Remote Notifications With Error: \(error)")
75     }
76
77     //.....

```

## (2) 服务端代码: index.php

```

1  <?
2  //引入代码
3  require 'JPush/autoload.php';
4
5  use JPush\Client as JPush;
6
7  if(isset($_POST["message"])){
8      //初始化
9      $app_key = "7b528367738ec346395798fd";
10     $master_secret = "32da4a2c06dc7b24472c9828";
11     $client = new JPush($app_key, $master_secret);
12
13     //简单的消息发送样例
14     $result = $client->push()
15         ->setPlatform('ios', 'android')
16         ->addAllAudience()
17         ->message($_POST["message"])
18         ->options(array(
19             "apns_production" => true //true表示发送到生产环境(默认值), false为开发环境
20         ))
21         ->send();
22
23     echo 'Result=' . json_encode($result);
24 }
25 ?>
26 <html>
27 <head>
28 </head>
29 <body>
30 <form action="index.php" method="post">
31     消息: <input type="text" name="message"/>
32     <button type="submit">发送</button>
33 </form>
34 </body>
35 </html>

```

## 4, 更完整的发送样例

同推送通知一样, 发送自定义消息也可以添加附加字段, 以及定时发送。

```

1  // 发送自定义消息
2  $payload = $client->push()
3      ->setPlatform('ios', 'android')
4      ->addAllAudience()
5      ->message($_POST["message"], [
6          'title' => 'Hello',
7          'content_type' => 'text',
8          'extras' => [
9              'key1' => 'value1'
10         ]
11     ])
12     ->options(array(
13         "apns_production" => true //true表示发送到生产环境(默认值), false为开发环境
14     ))
15     ->build();
16
17 // 创建在指定时间点触发的定时任务
18 $response = $client->schedule()->createSingleSchedule("指定时间点的定时任务",
19     $payload, array("time"=>"2016-11-01 15:00:00"));
20 echo 'Result=' . json_encode($response);

```

上一篇 [Swift - JPush极光推送的使用6（定时推送通知）](#)

下一篇 [Swift - 页视图控制器（UIPageViewController）的使用1（基本用法）](#)



阿里大鱼  
阿里巴巴集团旗下

实时 稳定 简单



广告

评论 ( 0 )



在这里留下你想说的话。

昵称

邮箱

提交

移动版

给我留言

|

现在有 198 位访客在线

友情链接（申请）

GTmetrix

© copyright 2006-2016 航歌. All rights reserved. 苏ICP备14040777号

http://www.hangge.com/blog/cache/detail\_1277.html

第 4 页（共 4 页）