

# iOS9和iOS10推送通知处理小结

今天对项目中处理推送通知的代码进行了整理，如有错漏，欢迎指出。

APP状态	iOS9	iOS10
前台	收到通知时执行方法2	收到通知时执行方法3，点击通知时执行方法4
后台	点击通知执行方法2	点击通知执行方法4
杀死进程	点击通知执行方法1	点击通知执行方法4，然后执行方法1
方法1	<code>- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions</code>	
方法2	<code>- (void)application:(UIApplication *)application didReceiveRemoteNotification:(NSDictionary *)userInfo</code>	
方法3	<code>- (void)userNotificationCenter:(UNUserNotificationCenter *)center willPresentNotification:(UNNotification *)notification withCompletionHandler:(void (^)(UNNotificationPresentationOptions options))completionHandler</code>	
方法4	<code>- (void)userNotificationCenter:(UNUserNotificationCenter *)center didReceiveNotificationResponse:(UNNotificationResponse *)response withCompletionHandler:(void (^)(void))completionHandler</code>	

参照上图，由于资源有限，没有对iOS8进行测试，情况应该是跟iOS9是一样的。

以我们的项目为例，因为iOS系统9和10的区别为APP处于前台时，iOS9不

会有通知提醒，为了避免用户错过重要的推送提醒，因此做了弹框处理。所以在写代码的时候，要区分iOS9和iOS10。

iOS9系统的处理代码在方法1和方法2实现，分别对应APP进程被杀死状态点击通知和APP处于前/后台状态点击通知这两种情况。iOS10系统的处理代码在方法4实现，需要注意的是iOS10系统APP进程被杀死状态下，同样会调用方法1，因此在方法1里要判断一下系统版本，避免重复执行。

要区别APP当前处于睡眠状态，可以使用下面的代码来获取当前的状态：

```
[UIApplication sharedApplication].applicationState
```

返回值为以下数据类型。通过测试发现，APP处于前台时返回的是UIApplicationStateActive，值为0。而APP处于后台和进程杀死状态下，均返回1。

```
typedef NS_ENUM(NSInteger, UIApplicationState) {  
    UIApplicationStateActive,  
    UIApplicationStateInactive,  
    UIApplicationStateBackground  
} NS_ENUM_AVAILABLE_IOS(4_0);
```

在收到通知的时候，我们的APP有可能处于任意一个界面，在AppDelegate类里想要跳转到其它ViewController，需要获取到当前所在的ViewController对象，可以通过下面的代码获取到该对象。

```
- (UIViewController*)getCurrentVC {  
    //获得当前显示viewController所在的tabBarController  
    UINavigationController*nav = [[self.tabBarController viewControllers  
    //获得当前显示viewController所在的tabBarController所包含的viewController  
    NSIntegercountOfVC = [nav.viewControllerscount];  
    //获得APP当前或者退到后台前显示的viewController  
    UIViewController*currentViewController = [nav.viewControllersobject  
    return currentViewController;  
}
```

当APP处于进程被杀死状态下，点击通知进入APP，此时由于ViewController的对象还没有创建完成，获取当前ViewController会导致闪退，因此我们需要将获取当前ViewController的代码放在下面的block中执行：

```
dispatch_after(dispatch_time(DISPATCH_TIME_NOW, (int64_t)(0.5*NSEC_PER_SEC)
));
```

这样会将Block里的代码延时提交到主队列中，保证ViewController对象创建成功后再运行。