

iOS SDK 接口说明

- [SDK接口详细说明](#)
 - [远程推送API](#)
 - [本地推送API](#)
 - [短信验证API](#)
 - [推送回调API](#)
 - [DeviceToken回调API](#)
 - [检查版本更新](#)
 - [设置为测试设备](#)
 - [别名和标签API](#)

mPush SDK 提供的 API 主要集中在 `PushManager.h` 接口类中。

远程推送API

```
/
开启推送服务并设置推送代理以及DeviceToken获取回调代理、自定义展示页面数据
pushDelegate 推送代理，必须，参数对象必须实现onMessage:content:extention:方法
deviceTokenDelegate DeviceToken获取代理，如果无需获取DeviceToken，此项可为nil。
(此DeviceToken为苹果原生DeviceToken的32位hex字符串。)
customViews 为自定义展示页面数据字典，如果无需自定义，此项可为nil
/

+ (void)startPushService:(NSDictionary *)launchOptions
    pushDelegate:(id <PushManagerDelegate>) pushDelegate
    deviceTokenDelegate:(id <DeviceTokenDelegate>) deviceTokenDelegate
    customViews:(NSDictionary *)displaysMap;
```

示例代码：

```
// Optional 自定义展示页
// 添加后，新建推送时，可指定页面，注意新建推送时，看到的是自定义的界面的别名Key。
// key:设置预打开界面的别名 value:自定义展示界面的类名

NSMutableDictionary dictionary = [NSMutableDictionary dictionary];
[dictionary setObject:NSStringFromClass([DemoViewController class])
 forKey:@"DemoPage"];

[PushManager startPushService:launchOptions
    pushDelegate:self
    deviceTokenDelegate:self
    customViews:dictionary];
```

收到通知后是否弹框

```
/**
 设置是否允许SDK当应用在前台运行收到Push时弹出Alert框（默认开启）
@param warning 透传消息始终不会弹出Alert框，如需要弹出，请在消息回调方法中自定义！
@param value 是否开启弹出框
/

+ (void)setAutoShowAlert:(BOOL)value;
```

本地推送 API

- 创建本地通知

```
/**
 创建本地通知

@param title      通知标题
@param content    通知内容
@param action     详见 LocalPushAction 类
@param extention  自定义参数集合
@param fireDate   通知发生时间

@return SDK分配给本通知的标识
/

+ (NSString *)sendLocalPushMsg:(NSString *)title
                        content:(NSString *)content
                        action:(LocalPushAction *)action
                        extensions:(NSDictionary *)extention
                        fireDate:(NSDate *)fireDate;
```

- 取消指定标识的单个本地通知

```
/**
 取消指定标识的单个本地通知

@param key 通知标识，为创建时所分配。
/

+ (void)cancelLocalPushForKey:(NSString *)key;
```

- 取消所有的本地通知

```
/
    一次性取消所有的本地通知
/

+ (void)cancelAllLocalPush;
```

短信验证API

```
/
    获取验证码接口（调用此接口前，请确保手机号正常可用。）

@param mobile        手机号
@param timeInterval 验证码有效时间，单位分钟，默认为3分钟
@param finishBlock   接口请求成功后的回调 成功是返回code，失败或者其他错误是返回errorMsg
/

+ (void)authSms:(NSString *)mobile
    timeInterval:(NSTimeInterval)timeInterval
    finished:(void(^)(NSString code, NSString errorMsg))finishBlock;
```

推送回调 API

```
/**
    推送消息代理回调方法 （包含远程推送和本地推送）

@param title    消息title内容
@param content  消息content内容
@param extention 参数字典,此字典包含用户自定义参数和应用状态参数。
    获取应用状态参数可通过 AppStateKey 来获取，状态类型参见 UIApplicationState

@return BOOL 当返回YES时，仅处理至当前事件处，后续事件将不再执行
    当返回NO时，按照事件链继续执行，直至返回YES或者所有事件执行完。
/

- (BOOL)onMessage:(NSString *)title
    content:(NSString *)content
    extention:(NSDictionary *)extention;
```

DeviceToken回调 API

```
/
获取DeviceToken，此DeviceToken为苹果原生DeviceToken 的32位hex字符串。
如果需要原生的DeviceToken，可使用系统
didRegisterForRemoteNotificationsWithDeviceToken：方法获取

@return deviceToken
/

- (void)didReciveDeviceToken:(NSString *)deviceToken;
```

检查版本更新

```
/
版本更新,无需按渠道更新时，请设置为nil。按渠道更新时，需要传入渠道名称

@param isNewest 是否已经是最新版本回调，开发者可以根据此返回值做界面提示等
/

+ (void)checkNewUpdateWithChannel:(NSString )channelName success:
(void(^)(BOOL isNewest))isNewest;
```

设置为测试设备

“

如果需要将终端设置为测试设备，需要在集成了SDK，并成功调用
*startPushService: pushDelegate: deviceTokenDelegate: customViews:*方法，注册
*mPush*服务后有效。

手机连续摇5次后，SDK将自动将终端设置为测试设备，再次连续摇5次后，即可注销测试设备。

别名和标签 API

- 别名 **alias**

上报用户的别名(例如用户昵称)，以便支持按别名推送消息。

```
/**
 为设备设置别名或移除别名,设置的前提条件是设备的DeviceToken已经获取成功。

@param alias 别名名称,当为nil或者@""时,为移除别名,否则为设置别名。
重复调用会自动更新至最后一次调用时的别名。
/

+ (void)setAlias:(NSString *)alias;
```

- **标签 tag**

开发者可以针对不同的用户设置标签,以便根据标签进行指向性推送消息。

- **设置标签**

```
/**
 为设备设置tag,可以为多个。

@param tags tag名称。
/

+ (void)setTags:(NSMutableArray *)tags;
```

- **删除标签**

```
/*
 删除设置过的tag,可以为多个。

@param tags tag名称。
/

+ (void)deleteTags:(NSMutableArray *)tags;
```