



Android PUSH-SDK API 接入文档

接口概述

本 SDK 中有三个开放类，分别为：Push 操作类 `PushClient`、回调接收类 `OpenClientPushMessageReceiver`/ `IPushActionListener`。

| 类 | 描述 |
|--|-----------------------|
| <code>PushClient</code> | 功能接口，用于调用 Push 相关功能。 |
| <code>OpenClientPushMessageReceiver</code> | 回调接口，用于 Push 的数据结果返回。 |
| <code>IPushActionListener</code> | 操作回调接口，用于功能接口状态回调。 |

1. PushClient 功能类

`PushClient` 提供 Push 服务的对外 API 列表，采用单例模型，通过 `getInstance(Context context)` 获得该对象。

| 方法 | 功能 |
|--|--|
| <code>initialize</code> | 初始化 Push 服务。 |
| <code>checkManifest</code> | 检查 <code>AndroidManifest</code> 是否正确声明组件和权限。 |
| <code>turnOnPush</code> | 打开 PUSH。 |
| <code>turnOffPush</code> | 关闭 PUSH。 |
| <code>getRegId</code> | 获取当前设备的当前应用的唯一标识。 |
| <code>bindAlias</code> | 设置应用别名。 |
| <code>unBindAlias</code> | 取消应用别名。 |
| <code>getAlias</code> | 获取应用别名。 |
| <code>setTopic</code> | 设置主题。 |
| <code>delTopic</code> | 删除主题。 |
| <code>getTopics</code> | 获取主题。 |



| | |
|---------------------------|----------------|
| isSupport | 当前系统是否支持 PUSH。 |
|---------------------------|----------------|

2. OpenClientPushMessageReceiver 回调接收类

OpenClientPushMessageReceiver 类提供基础消息的接收和操作结果的反馈，需要开发者继承本类，并重载相关的方法；同时，还需要在 AndroidManifest.xml 静态注册。

| 方法 | 描述 |
|--|-------------|
| onReceiveRegId | RegId 结果返回。 |
| onNotificationMessageClicked | 通知被点击结果返回。 |

3. IPushActionListener 操作回调类

IPushActionListener 类提供 PushClient 的操作回调。

| 方法 | 描述 |
|--------------------------------|---------------|
| onStateChanged | 操作状态返回，详见操作码。 |

公共状态码

状态码用于返回各项操作的结果，常用于回调接收类中状态返回，以下为公共状态码：

| 状态码 | 描述 |
|-------------|-------------------------|
| 0 | 操作成功。 |
| 1 | 操作成功，此动作在未操作前已经设置成功。 |
| 101 | 系统不支持。 |
| 102 | PUSH 初始化异常，请重现初始化 PUSH。 |
| 1001 | 一天内调用次数超标。 |
| 1002 | 操作频率过快。 |
| 1003 | 操作超时。 |



| | |
|-------|----------------|
| 1004 | 应用处于黑名单。 |
| 1005 | 当前 push 服务不可用。 |
| 10000 | 未知异常。 |

API 接口

PushClient 类

initialize

功能介绍

初始化 push 服务，可以提高后台发送消息的实时性。

接口定义

```
void initialize()
```

补充

建议每次手机启动后，都调用一次，确保 Push 服务能够正常运行。

turnOnPush

功能介绍

打开应用 push 开关，绑定应用，成功后便可接收到当前应用的推送消息。

接口定义

```
void turnOnPush(IPushActionListener listener)
```

参数说明

listener: 状态监听。

补充

打开 push 成功后无需重复调用。

turnOffPush

功能介绍

关闭应用 push 开关，解除绑定应用，解绑后将收不到当前应用的推送消息。

接口定义

```
void turnOffPush(IPushActionListener listener)
```

参数说明

listener: 状态监听。

补充

关闭 push 成功后无需重复调用。

getRegId

功能介绍

获取当前设备的当前应用的唯一标识，后台可基于此标识发送通知。



接口定义

```
String getRegId()
```

返回值

当前设备的当前应用的唯一标识。

补充

此方法仅在打开 PUSH 成功后可以获取，否则为 null。

bindAlias

功能介绍

在注册成功，收到 **regId** 之后，即可设置别名，绑定当前设备的当前应用的别名，可以理解为 **regId** 的别名，开发者可以将别名设置为自己应用帐号系统的帐号，或者设备标识等。然后通过此别名发送消息。

接口定义

```
bindAlias(String alias, IPushActionListener listener)
```

参数说明

alias：别名。

listener：状态监听。

| 状态码 | 描述 |
|-------|---------------------|
| 30001 | 设置别名失败：请打开 push 开关。 |
| 30002 | 设置别名失败：订阅别名为空。 |



| | |
|-------|--------------------------|
| 30003 | 设置别名失败：别名设置超长，字符长度超过 70。 |
|-------|--------------------------|

补充

绑定别名，同一个别名仅能绑定一个 regId。

当 regId 已绑定了别名 A，若调用此接口绑定别名 B，则与别名 A 的绑定关系会自动解除。

此接口与 unBindAlias 一天内最多调用 100 次，两次调用的间隔需大于 2s。

unBindAlias

功能介绍

删除别名，解除绑定当前应用别名，解绑成功后后台将无法通过别名的方式向应用发消息。

接口定义

```
void unBindAlias(String alias)
void unBindAlias(String alias, IPushActionListener listener)
```

参数说明

alias：别名。

listener：状态监听。

| 状态码 | 描述 |
|-------|--------------------------|
| 30001 | 删除别名失败：请打开 push 开关。 |
| 30002 | 删除别名失败：订阅别名为空。 |
| 30003 | 删除别名失败：别名设置超长，字符长度超过 70。 |

补充

此接口与 `bindAlias` 一天内最多调用 100 次，两次调用的间隔需大于 2s。

getAlias

功能介绍

获取当前设备的应用别名，后台可基于此别名发送通知。

接口定义

```
String getAlias()
```

返回值

当前设备的应用别名。

checkManifest

功能介绍

校验接入是否异常。通过调用该 API 来检查 `AndroidManifest.xml` 是否正确声明了本 SDK 要求的所有组件和权限。当有组件和权限未定义，或者组件的属性声明错误时抛出异常 `VivoPushException`。

接口定义

```
void checkManifest()
```

setTopic

功能介绍

在注册成功, 收到 `regId` 之后, 即可设置主题; 开发者可根据用户订阅的主题实现分组。

接口定义

```
setTopic(String topic, IPushActionListener listener)
```

参数说明

`topic`: 需要订阅的主题名。

`listener`: 状态监听。

| 状态码 | 描述 |
|-------|---------------------------|
| 20001 | 设置主题失败: 请打开 push 开关。 |
| 20002 | 设置主题失败: 订阅主题为空。 |
| 20003 | 设置主题失败: 别名设置超长, 字符长度超过 70 |
| 20004 | 设置主题失败: 订阅次数太频繁或已订阅数过多。 |

补充

同一个 `regId` 可设置多个主题, 最多可订阅 500 个主题。

当 `regId` 已设置了 `topicA`, 若调用此接口设置 `topicB`, 则 `regId` 同时订阅了 `topicA` 和 `topicB`。

此接口与 `delTopic` 一天内最多调用 500 次, 两次调用的间隔需大于 2s。



delTopic

功能介绍

删除已经设置的主题。

接口定义

delTopic(String topic, IPushActionListener listener)

参数说明

topic：需要取消订阅的主题名。

listener: 状态监听。

| 状态码 | 描述 |
|-------|--------------------------|
| 20001 | 删除主题失败：请打开 push 开关。 |
| 20002 | 删除主题失败：订阅主题为空。 |
| 20003 | 删除主题失败：别名设置超长，字符长度超过 70。 |
| 20004 | 删除主题失败：取消主题次数太频繁。 |

补充

此接口与 setTopic 一天内最多调用 500 次，两次调用的间隔需大于 2s。

getTopics

功能介绍

获取当前设备的应用主题，后台可基于此主题发送通知。



接口定义

List<String> getTopics()

返回值

当前设备已经设置的应用主题。

isSupport

功能介绍

用于判断当前系统是否支持 PUSH 服务。

接口定义

boolean isSupport()

返回值

true:系统支持。

false:系统不支持。

OpenClientPushMessageReceiver 类

onReceiveRegId

功能介绍

RegId 结果返回。当开发者首次调用 turnOnPush 成功或 regId 发生改变时会回调该方法。



接口定义

```
void onReceiveRegId(Context context, String regId);
```

参数说明

context：应用上下文。

regId：当前设备的当前应用的唯一标识。

onNotificationMessageClicked

功能介绍

通知被点击后的结果返回。当 push 发出的通知被点击后便会触发 onNotificationClicked 通知应用。

接口定义

```
void onNotificationMessageClicked (Context context, UPSNotificationMessage msg)
```

参数说明

context：应用上下文。

msg: UPSNotificationMessage 通知消息结构体，包含以下字段：

- msgid: 通知 id
- title: 通知标题
- content: 通知内容
- skipContent: 通知自定义内容
- params: 自定义键值对