



Android PUSH-SDK API 接入文档

注意：

1. Vivo 推送服务 SDK 支持的最低 android 版本为 Android 6.0。
2. 由于项目架构变动，可能会导致您在更新 sdk 版本时类的路径错误，重新导入类的路径即可。

1 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。

1.1 PushClient 功能类

`PushClient` 提供 Push 服务的对外 API 列表，通过 `PushClient.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 开关。
<code>turnOffPush</code>	关闭 Push 开关。
<code>getRegId</code>	获取当前设备的当前应用的唯一标识，需在 <code>turnOnPush</code> 成功后使用。
<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。



setTopic	设置应用标签。
delTopic	删除应用标签。
getTopics	获取应用标签。
isSupport	当前系统是否支持 Push。

1.2 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 中静态注册，详情见 [PushSDK 集成指南](#)。

方法	描述
onReceiveRegId	当首次 turnOnPush 成功或 regId 发生变化时回调。
onNotificationMessageClicked	通知被点击时回调。

1.3 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。

1.4 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。

2 API 接口

2.1 PushClient 类

initialize

功能介绍

初始化 push 服务。

接口定义

```
void initialize()
```

补充

无。

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，返回结果见上方[公共状态码](#)。

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，返回结果见上方[公共状态码](#)。

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。

接口定义

```
void checkManifest()
```

setTopic

功能介绍

设置标签，设置标签后应用将会接收到后台针对该标签发送的消息。

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。

delTopic

功能介绍

删除标签，删除标签成功后后台将无法通过该标签的方式向应用发消息。

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。

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别名。

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。

2.2 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

统一推送联盟 API 介绍

turnOnPush

功能介绍

做了相应的初始化操作，建议用户在自己应用的 Application 中的 onCreate() 方法中调用 turnOnPush 操作。

接口定义

```
void turnOnPush(Context context, UPSTurnCallback callback)
```

参数说明

context：应用上下文。

Callback：UPSTurnCallback 回调结构，包含以下方法

- getReturnCode(): 获取返回状态，0 代表成功，其他值代表失败

registerToken

功能介绍

注册 push，获取申请的 regId



接口定义

```
void registerToken(final Context context, String appId, String appKey, String appSecret, final UPSRegisterCallback callback)
```

参数说明

context：应用上下文。

appId：应用在 Vivo 开放平台申请的 APPId

appKey：应用在 Vivo 开放平台申请的 APPKey

appSecret：应用在 Vivo 开放平台申请的 APPSecret

callback: UPSRegisterCallback 回调结构；包含以下方法

- getReturnCode(): 获取返回状态，0 代表成功，其他值代表失败
- getToken(): 获取应用申请的 regId

unRegisterToken

功能介绍

接注册 push，关闭 push 功能

接口定义

```
void unRegisterToken(Context context, final UPSRegisterCallback callback)
```

参数说明

context：应用上下文。

Callback: UPSRegisterCallback 回调结构，包含以下方法

- getReturnCode(): 获取返回状态，0 代表成功，其他值代表失败



turnOffPush

功能介绍

该功能暂时未具体实现，要关闭 push 功能，可调用上面的 unRegisterToken 接口

接口定义

```
void turnOffPush(Context context, final UPSTurnCallback callback)
```

参数说明

context：应用上下文。

Callback：UPSTurnCallback 回调结构，包含以下方法

- `getReturnCode()`：获取返回状态，0 代表成功，其他值代表失败