

CarNotification Android SDK 接入文档

目录

CarNotification Android SDK 接入文档.....	错误!未定义书签。
1. Android SDK 概述.....	4
1.1.产品简介.....	4
1.2.功能简介.....	4
2. SDK 接入与初始化.....	4
2.1.导入 SDK.....	4
2.2.配置 Gradle.....	4
2.3.初始化 SDK.....	4
3. 接口说明.....	5
3.1.发送消息.....	5
3.2.删除消息.....	8
3.3.删除所有消息.....	9

文档说明	CarNotification Android SDK 接入文档
创建日期	2018/11/02
适用项目	福特 phase6
文档变更	
变更时间	变更内容
2018/11/02	新增 1. 产品及功能简介 2. SDK 接入流程（配置 / 初始化） 3. 发送消息，删除消息
2018/12/06	增加 mMode 字段，修改消息级别
2018/12/11	增加广播发送 toast 类型消息
2019/1/15	增加查看是否有未读消息以及回调
2019/04/24	增加附录使用例子，修改 level，type 值，增加 query，

	futureTime, replay, carCondition 等字段
2022/12/06	针对 phase5/6 删减部分内容

1. Android SDK 概述

1.1.产品简介

CarNotification Android SDK 是一套基于 Android 5.0 及以上版本设备的应用程序接口。通过该套 SDK 接口，您可实现发送消息通知用户相关事情。

该套 SDK 提供给具有一定 Android 编程经验和了解面向对象概念的读者使用，此外，读者还应该对车联网服务有一定的了解。

1.2.功能简介

提供接口绑定服务，解绑服务，可以发送多种类型消息通知。

2. SDK 接入与初始化

2.1.导入 SDK

将以下 aar 文件拷贝到 libs 目录下。

car-notification-v5.1.0.aar

2.2.配置 Gradle

✧ 引入aar

```
api(name: 'car-notification-v5.1.0', ext: 'aar')
```

2.3.初始化 SDK

✧ 绑定服务

```
CarNotificationManager.getInstance().init(this);
```

请在 Application onCreate()方法中进行

2.4.混淆

proguard-rules.pro 中加入下面两行

#消息中心

```
-dontwarn com.baidu.xiaoduos.messageserverlib.**  
-keep class com.baidu.xiaoduos.messageserverlib.** { *; }
```

3. 接口说明

管理类： CarNotificationManager 负责管理所有消息处理，包括发送、更新以及删除。

3.1.发送消息

✧ 方法定义

```
public void sendNotification(String id,CarNotification notification)
```

✧ 参数详情

notification	消息主体
id	消息唯一标识，用于更新或者删除同一通知

CarNotification:

```
private CarPendingIntent mContentPendingIntent
```

点击消息整体意图	自定义，比如点击打开app，或者详情页，可参考CarPendingInten提供的多种方法
----------	---

```
private CarPendingIntent mBtnLeftttPendingIntent
```

左按钮意图	自定义，比如可设置点击后拨号
-------	----------------

```
private CarPendingIntent mBtnRightPendingIntent;
```

左按钮意图	比如点击跳往其他应用
-------	------------

`private Bitmap mAppIconBitmap;`

App 图标	用于显示在消息左上角，app自身图标
--------	--------------------

`private int mNotificationType;`

消息类型	共 2 个类型： TYPE_NORMAL_BANNER = 2: Banner 弹框； TYPE_NORMAL_DIALOG = 8: 带按钮对话框，可选设置 1-2 个按钮 (备注：数字后续可能修改，调用请直接采用命名)
------	--

`private int mLevel;`

消息等级	共五个等级： 消息优先级初步分为 5 级别，1 级优先级最低，5 级优先级最高，1, 2, 3 级代表低、中、高。 各应用可定义每个消息的优先级别，只能选择 1, 2, 3 级；当应用需要定义 4, 5 级消息时，需由百度审核。 LEVEL_ONE= 1; LEVEL_TWO = 2; LEVEL_THREE = 3; LEVEL_FOUR = 4; LEVEL_FIVE = 5;
------	---

`private int mDisplayPosition;`

显示位置	共 7 种情况： DISPLAY_ONLY_IN_DROPDOWN = 1: 消息只显示在下拉列表； DISPLAY_ONLY_IN_DIALOG = 2: 消息只显示弹框； DISPLAY_ONLY_IN_CENTER = 3: 消息只显示在历史消息；
------	--

	<p>DISPLAY_DROPDOWN_AND_DIALOG = 4: 消息弹框与显示下拉列表;</p> <p>DISPLAY_DROPDOWN_AND_CENTER = 5: 消息显示在下拉列表与历史消息;</p> <p>DISPLAY_DIALOG_AND_CENTER = 6: 消息弹框与显示在历史消息;</p> <p>DISPLAY_ALL_WAYS = 0: 消息显示所有位置(默认)。</p>
--	---

private String **mContentTitle**;

消息标题	标题，最多显示一行，最多 20 个字符
------	---------------------

private int **mContentText**;

消息内容	内容，可以多行，最多显示100字符
------	-------------------

private int **mAppName**;

app 名	发送通知的app名，最长10字符
-------	------------------

private int **mBtnLeft**;

左按钮	左按钮，最长6字符
-----	-----------

private int **mBtnRight**

右按钮	右按钮，最长6字符
-----	-----------

private int **mPackageName**;

包名	发出通知的app包名
----	------------

private int **mTargetPackageName**;

目标包名	推送专用，为目标app发送消息
------	-----------------

private int **mTargetAppName**;

包名	推送专用，app名
----	-----------

`private String mTTS`

语音播报	空表示该消息不需要语音播报，需要播报则传入播报的内容，最多60字符
------	-----------------------------------

`private int mBtnLeftAutoCountDown`

左边按钮倒计时	例：1000 = 1s
---------	-------------

`private int mBtnRightAutoCountDown`

右边按钮倒计时	例：1000 = 1s
---------	-------------

`private int mDuration`

自定义弹框消失时间	单位毫秒，例：1000 = 1s
-----------	------------------

`private boolean withAudio`

音频播放	True/False 消息弹出时是否播放音频
------	------------------------

`private boolean mEnableVoiceControl`

语音操作	True/False 消息是否支持语音操作（打开、关闭等），如不需要，请指定false
------	---

3.2.删除消息

消息展示在下拉列表或者非下拉列表时，可以调用此接口删除 id 相对应的消息。

✧ 方法定义

`public void cancel (String id)`

✧ 参数说明

id 消息 id

3.3.删除所有消息

消息展示在下拉列表或者非下拉列表时，可以调用此接口删除该 app 相对应的所有消息。

✧ 方法定义

```
public void cancelAll ()
```

附录

使用例子：

初始化：

```
CarNotificationManager.getInstance(this).init();
```

弹框：

```
// 左边按钮跳转(自定义)
```

```
Intent intent = new Intent(this, LeftActivity.class);
```

```
CarPendingIntent leftPI = CarPendingIntent.getActivity(MainActivity.this, 0,  
    intent, PendingIntent.FLAG_UPDATE_CURRENT);
```

```
// 右边按钮跳转(自定义)
```

```
intent = new Intent(this, RightActivity.class);
```

```
CarPendingIntent rightPI = CarPendingIntent.getActivity(this, 0,  
    intent, PendingIntent.FLAG_UPDATE_CURRENT);
```

```
// 内容跳转(自定义)
```

```
intent = new Intent(this, ContentActivity.class);
```

```
CarPendingIntent contentPI = CarPendingIntent.getActivity(this, 0,
```

```
intent, PendingIntent.FLAG_UPDATE_CURRENT);
```

```
CarNotification notification = new CarNotification.Builder(this)
    .setAppIconBitmap(bitmap)
    .setAppName("应用名称")
    .setContentTitle("标题文案")
    .setContentText("内容文案") // 内容不能为空，否则无效。
    .setContentPendingIntent(contentPI)
    .setBtnLeft("左边按钮文案")
    .setBtnLeftPendingIntent(leftPI)
    .setBtnRight("右边按钮文案")
    .setBtnRightPendingIntent(rightPI)
    .setNotificationType(CarNotification.TYPE_NORMAL_BANNER) // 必须设置那 2 种
    类型弹框，否则无效。
    .setBtnLeftAutoCountDown(0)
    .setBtnRightAutoCountDown(0)
    .setTTS("tts 内容")
    .setLevel(5) // 1-5 级，不设置默认等级 1
    .setDuration(5000) // 默认 5s
    .setDisplayPosition(0) // 默认 0
    .withAudio(true) // 默认 false
    .build();
```

```
CarNotificationManager.getInstance(this).sendNotification("msgid", notification);
```

消息长链接推送 json 字段:

```
{
  "contentIntent": {
    "cls": "",
    "type": 1,
    "extra": "",
    "action": "",
    "data": "",
    "pkg": ""
  },
  "leftIntent": {
    "cls": "",
    "type": 1,
    "extra": "",
    "action": "",
    "data": "",
    "pkg": ""
  },
  "rightIntent": {
    "cls": "",
    "type": 1,
    "extra": "",
    "action": "",
    "data": "",
    "pkg": ""
  },
  "type": 2,
  "level": 3,
  "displayPosition": 0,
  "title": "标题",
  "content": "content 内容",
```

```
"appName": "",  
"btnLeft": "左边",  
"btnRight": "右边",  
"pkg": "包名",  
"tts": "测试语音",  
"mBtnLeftAutoCountDown": 5000,  
"mBtnRightAutoCountDown": 5000,  
"duration": 10000,  
"withAudio": true,  
"mEnableVoiceControl": false,  
"id": "",  
"msgId": "消息 id"  
}
```