

激活-长连接推送-数据统计接口文档

激活/推送/埋点 SDK 接口文档 V0.2.15 (2020.11.23)

[最新版本 syncclient-sdk 下载](#) (对于福特项目, rom 版本请使用 1 月 8 日以后的版本)

百度国际科技(深圳)有限公司

车联网事业部

作者: 李大青 (lidaqing@baidu.com) 刘正德 (v_liuzhengde@baidu.com)

1	修改日期	改动说明
2	V0.1	初版
3	V0.2.1	增加 getVin, getDeviceID 接口 修改 add/remove listener 接口 (IPushMessageListener 参数改为 OnPushMessageListener)
4	V0.2.2	增加 getAK, getCN, getSignPrefix, getSignSuffix 接口
5	V0.2.4	合并 setUserID, setBduss 接口为 updateAccountInfo 增加数据埋点接口 onEvent(int event, int type, String module)
6	V0.2.5	1. 增加 getBaiduAccountUID 接口 2. 修改 onEvent(int event, int type, String module) 为 onEvent(int event, int type) 3. 修复非首次开机获取 UUID 很慢的问题
7	V0.2.6	修改 onEvent(int event, int type) 的 TYPE 类型, 增加进入页面/退出页面事件
8	V0.2.7	添加客户进程对服务进程的监听

9	V0. 2. 8	增加 <code>void onEvent(int event, EventType type,String attachMent)</code> 接口
10	V0. 2. 9	增加 <code>void onPageEvent(int eventId, EventType type, String pageName)</code> 接口
11	V0. 2. 10	增加三个类型的 EventType 1. KEY_TYPE, /*物理按键类型*/ 2.DISPLAY_TYPE, /*展现类型*/ 3.BACKSTAGE_TYPE; /*后台类型*/
12	V0. 2. 11	增加三个接口： <code>void onEvent(String eventId, EventType type)</code> <code>void onEvent(String eventId, EventType type,String attachMent)</code> <code>void onPageEvent(String eventId, EventType type, String pageName)</code>
13	V0. 2. 12	统计项版本名称上传改为版本号上传
14	V0. 2. 13	更新数据接口
15	V0. 2. 14	解决上报丢失的问题
16	V0. 2. 15	修改 EventType 类型

1. SDK 功能简介

目录:

2. SDK 接口说明

3. 如何接入 SDK

1. SDK 功能简介

- 获取本车机的 UUID（本地获取）
- 注册/注销推送消息监听器
- 百度账户 UID（账户登录时主动通知到 SDK）
- 百度账户 BDUSS（账户登录时主动通知到 SDK）
- 判断是否小度车载 OS 系统
- 获取当前车机 DuerOS 的 AK，渠道号，应用签名等信息
- 数据埋点

2. SDK 接口说明

2.1 SDK 接口初始化

所有接口都需要导入包：`import com.baidu.xiaoduos.syncclient.Entry`

- `static Entry getInstance()`

作用： 获取 Entry 实例对象

- `void init(Context context, Entry.Callback callback)`

作用： 初始化 SDK，异步调用，直接返回

参数：@context: caller 的上下文（Context）

@callback: 初始化回调函数

`public interface Callback {`

`//初始化成功时回调（连接上系统服务）`

```

                                void onSuccess();    //初始化失败时回调（未连接上系统服务）

                                void

onFailure();

                                }

```

2.2 SDK 数据统计埋点接口

规则： 用户行为数据统计每 10 条数据保存一次，每 17 分钟上传一次，每次上传最多读取 1000 条，读完为止，单次 POST 不多于 40 条数据到服务器

- `void onEvent(int eventId, EventType type)`

数据埋点采集数据接口：

参数： @eventId: // Number(int); 自定义事件 id (PM 定义)

@type: //Number(int);从 EventType.java 中获取交互类型

声控 0 (EventType.VOICE_TYPE); 触控 1 (EventType.TOUCH_TYPE) 消息类型 2 (EventType.MESSAGE_TYPE); 物理按键类型 3 (EventType.KEY_TYPE)

展现类型 4 (EventType.DISPLAY_TYPE); 后台类型 5 (EventType.BACKSTAGE_TYPE) -1 缺省值 (EventType.OTHER_TYPE)

根据需求页面平均时长统计(目前新接入埋点的请用下面新接口 onPageEvent)， 新增页面时长埋点参数 4 个，

(该参数只针对页面时长埋点) 如下：

声控进入 6 (EventType.VOICE_START_TYPE); 声控退出 7 (EventType.VOICE_END_TYPE) 触控进入 8 (EventType.TOUCH_START_TYPE); 触控退出 9 (EventType.TOUCH_END_TYPE)

示例：mSyncEntry.onEvent(30100000, EventType.TOUCH_TYPE);

页面时长埋点示例：埋点需求——>用户每天使用【意见反馈】时长(目前新接入埋点的请使用下面新接口 onEventPage)

进入意见反馈页面事件
mSyncEntry.onEvent(30100000, EventType.TOUCH_START_TYPE);

退出意见反馈页面事件
mSyncEntry.onEvent(30100000, EventType.TOUCH_END_TYPE);

- void onEvent(String eventId, EventType type)
数据埋点采集数据接口：
参数： @eventId: //String 提供字面+数字的形式的 event 参数
其他说明同上
- void onEvent(int eventId, EventType type, String attachMent)
自定义上传数据 AttachMent 接口： event 与 type 参数说明同上，在接口上面增加 attachMent 的字段
@attachMent // String; 扩展其他参数供调用，传入与服务端约定好的 Json 串（SDK 不参与定义具体字段组装，目前云端解析负责的是罗中杰）
- void onEvent(String eventId, EventType type, String attachMent)
自定义上传数据 AttachMent 接口： type 参数说明同上，在接口上面增加 attachMent 的字段
@eventId: //String 提供字面+数字的形式的 event 参数
@attachMent // String; 扩展其他参数供调用，传入与服务端约定好的 Json 串（SDK 不参与定义具体字段组装，目前云端解析负责的是罗中杰）
- void onEventPage(int eventId, EventType type, String pageName)
用户使用时长时间上报接口： event 与 type 参数说明同上，pageName 时长统计的页面或者功能描述
@pageName // String; 用户页面名称或者功能名称描述等
- void onEventPage(String eventId, EventType type, String pageName)
用户使用时长时间上报接口： type 参数说明同上，pageName 时长统计的页面或者功能描述
@eventId: //String 提供字面+数字的形式的 event 参数
@pageName // String; 用户页面名称或者功能名称描述等

2.3 SDK 其他接口（获取 uuid, vin, deviceId 等）

- `String getUUID()`

作用：获取 UUID

返回：UUID 字符串（0x 开头的 17 个数字的字串）或者空值

注意：当车机刷机后第一次启动时，getUUID 函数可能返回空值（因为首次开机或者将车机安装到新的车上时，需要联网获取 UUID），调用者需要考虑返回空值情况（当网络正常时重试）

- `boolean isXiaoduOS()`

作用：获取当前系统是否小度车载 OS

返回：@true: xiaoduos @false: 非 xiaoduos

- `void addPushMessageListener(String packageName, OnPushMessageListener listener)`

```
import com.baidu.xiaoduos.syncclient.OnPushMessageListener
```

作用：注册推送消息监听器，注意每个 packageName 名字只能注册一个监听器，若多次调用，只有最后一次的监听器起作用

参数：@PackageName: caller 所在应用的包名

@OnPushMessageListener: 监听器回调函数

```
Public abstract class OnPushMessageListener  
{
```

```

                                void onPushMessageListener(String
pkgName,   String content);

                                @pkgName: caller 所在
应用的包名

                                @content: 推送消息内容

                                (sdk 透传)

                                }

```

- **void** removePushMessageListener(String packageName, OnPushMessageListener listener)

```
import com.baidu.xiaoduos.syncclient.OnPushMessageListener
```

参数: @PackageName: caller 所在应用的包名

@OnPushMessageListener: 监听器回调函数

- **void** updateAccountInfo(String packageName, String uid, String bduss)

作用: 当百度账户登录成功时, 需要通知 sdk 百度账户的 uid, bduss, sdk 会将该值回传到服务器

参数: @PackageName: caller 所在应用的包名

@id: 百度账户的 uid
@bduss: 百度账户的 bduss

- **String** getVIN()

作用: 获取 VIN 返回: VIN 字符串, 如果没有获取到则返回 null

- **String** getDeviceID()

作用：获取 DeviceID 返回：Device ID 字符串，如果没有获取到则返回 null

- **String getCN()**

作用：获取 DuerOS 应用在当前车型的渠道号返回：渠道号字符串，如果没有获取到则返回 null

- **String getAK()**

作用：获取 DuerOS 应用在当前车型的 AK 值 返回：AK 字符串，如果没有获取到则返回 null

- **String getSignPrefix()**

作用：获取 DuerOS 应用在当前车型的签名前缀 返回：签名前缀字符串，如果没有获取到则返回 null

- **String getSignSuffix()**

作用：获取 DuerOS 应用在当前车型的签名后缀 返回：签名后缀字符串，如果没有获取到则返回 null

3. 如何接入 SDK?

SDK 接口需要依赖车机上集成的百度系统服务（BaiduSyncService），在 2018 年 11 月 30 日及以后的系统版本集成有该服务

- 将 sdk（以 syncclient-0.2.jar 为例子）到 AndroidStudio 工程的 libs 目录，然后在 gradle 加入：


```
Dependencies {... implementation files('libs/syncclient-0.2.jar')}
```

- 初始化:

```
private Entry mSyncEntry;

...

mSyncEntry = Entry.getInstance();

mSyncEntry.init(getApplicationContext(),

                                new Entry.Callback() {

                                    @Override

                                    public void

onSuccess() {...}

                                    @Override

                                    public void onFailure()

{...}});
```

- 调用（获取 UUID）：

```
mSyncEntry.getUUID()

....
```

- 注册/注销消息推送监听器

```
IPushMessageListener mListener = new IPushMessageListener.Stub() {

    @Override

    public void onPushMessageListener(String pkgName, String

content) {
```

```

        ...
    }

};

```

```

mSyncEntry.addPushMessageListener(getPackageName(), mListener);

```

```

protected void onDestroy() {
    ...
    mSyncEntry.removePushMessageListener(getPackageName(), mL
    istener);
    .....
}

```

推送测试接口（推送需要找服务端

RD[chenbinwen@baidu.com, v_zengheji@baidu.com] 注册自己 Package，注册后会分配一个 Token）

https://sandbox-vehicle.baidu.com/system/commitmsg?toid=0x82081156530798592&qq=test_test&pkg=com.baidu.xiaoduos.testsyncclient&content=content232132&token=2e4fd00ca7a96721f86c67bef3c75aa5646ee8b9&fr=com.baidu.xiaoduos.testsyncclient

pkg=需要接收推送消息的包名

token=服务端分配给 pkg 的 token

content=服务端推送消息_9A_内容

toid=从 sdk 获取的 uuid （需要以 0x 开头）