ApexDataCollect-Android

集成步骤

1. **在Project的build.gradle下，添加AnalyticsAop的插件：**

```
buildscript {

    repositories {
        google()
        jcenter()
    }

    dependencies {
        classpath 'com.android.tools.build:gradle:3.3.0' // 最低2.3.0
        classpath 'com.chinapex.analytics.aop:AnalyticsAop:1.0.6'
    }
}
```

2. **在app的build.gradle下，添加：**

```
apply plugin: 'aop'

dependencies {
    // app 自身的各种依赖
    ...

    // 依赖ApexCollectSDK
    implementation 'com.chinapex.android.datacollect:ApexCollectSDK:1.1.3'
    releaseImplementation 'com.chinapex.android.monitor:ApexCollectMonitorNoOp:1.0.2'
    debugImplementation 'com.chinapex.android.monitor:ApexCollectMonitor:1.0.2'

    // sdk后续为了完善追踪事件，可能还会增加其它依赖，待补充
```

```
    ...
}
```

- **注意：若app本身依赖以下的库，请保证跟SDK所依赖的版本号一致，或之上**

```
implementation 'com.android.support:appcompat-v7:27.1.1'
implementation 'com.android.support:recyclerview-v7:27.1.1'
implementation 'com.squareup.okhttp3:okhttp:3.12.0'
implementation 'com.google.code.gson:gson:2.8.5'
```

3. **在app的AndroidManifest.xml下，添加：**

```
// 非敏感权限
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />

// 敏感权限，需用户授权
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
```

4. **混淆规则**

```
# ==========================ApexCollectSDK start=============================
# sdk
-keep class com.chinapex.android.datacollect.**{*;}

# okhttp
-dontwarn javax.annotation.**
-keepnames class okhttp3.internal.publicsuffix.PublicSuffixDatabase
-dontwarn org.codehaus.mojo.animal_sniffer.*
-dontwarn okhttp3.internal.platform.ConscryptPlatform

# sqlcipher
-keep  class net.sqlcipher.** {*;}
-keep  class net.sqlcipher.database.** {*;}

# Gson
-keepattributes Signature
-keepattributes *Annotation*
-dontwarn sun.misc.**
-keep class * implements com.google.gson.TypeAdapterFactory
-keep class * implements com.google.gson.JsonSerializer
-keep class * implements com.google.gson.JsonDeserializer
# ==========================ApexCollectSDK end=============================
```

5. **在app的application中，初始化：**

```
ApexAnalytics.getInstance().init(
        new AnalyticsSettings.SettingsBuilder(applicationContex)  // 必须为应用的context
              .setUuid("testUuid")                                // 可选，默认androidId
              .setChannelId("豌豆荚")                              // 可选，安装渠道ID
              .setLogLevel(ATLog.VERBOSE)                         // 可选，默认WARN
              .setDelayReportInterval(1000 * 60 * 2)              // 可选，默认5分钟
              .setCheckInstantErrInterval(1000 * 60)              // 可选，默认2分钟
              .setReportMaxNum(5)                                 // 可选，默认30条
              .setUrlDelay("https://www.baidu.com/")              // 可选，默认是测试url
              .setUrlInstant("https://www.baidu.com/")            // 可选，默认是测试url
              .setHostnameVerifier("www.baidu.com")               // 可选，域名过滤
              .build());
```

- **注意：若设置上报url，务必同时设置HostnameVerifier，且保证其与url的域名一致，否则网络请求会无法发送**

API

自定义代码埋点：void track(TrackEvent trackEvent)

```
ApexAnalytics.getInstance().track(new TrackEvent.EventBuilder()
                        .setMode(1)  // 0: delay延时上报(default), 1: instant即时上报
                        .setLabel("即时上报的label")
                        .setValue({
                                "custom1": "111111",
                                "custom2": "222222",
                                "custom3": "333333",
                                "custom4": "444444",
                                "custom5": "555555"})
                        .build());
```

- **注意：此处value必须为<String, String>的json键值对，可以通过以下两种方式转化：**

```
// 1. 自定义一个对象，其属性必须全部为String类型，例如：TestTracker，然后用Gson转成json串(推荐)
TestTracker testTracker = new TestTracker();

testTracker.setDate("2018.12.12");
testTracker.setName("张三");
testTracker.setAge("18");
testTracker.setWork("study");

String value = GsonUtils.toJsonStr(testTracker);
```

```
// 2. 使用Properties，之后同样用Gson转成json串
Properties properties = new Properties();

properties.put("lala1", "lala11");
properties.put("lala2", "lala22");
properties.put("lala3", "lala33");
properties.put("lala4", "lala44");
properties.put("lala5", "lala55");

String value = GsonUtils.toJsonStr(properties);
```

App用户登入：void signIn(String userId)

```
ApexAnalytics.getInstance().signIn("testUserId");
```

App用户登出：void signOut()

```
ApexAnalytics.getInstance().signOut();
```

全埋点务必复写的方法

Fragment

> onViewCreated (View view, Bundle savedInstanceState)
>
> onResume
>
> onPause
>
> onHiddenChanged (boolean hidden)
>
> setUserVisibleHint (boolean isVisibleToUser)

ListView / GridView

> onItemClick (AdapterView<?> parent, View view, int position, long id)
>
> setOnScrollListener
>
> - onScrollStateChanged (AbsListView view, int scrollState)

> - onScroll (AbsListView view, int firstVisibleItem, int visibleItemCount, int totalItemCount)

> 在资源文件String.xml中定义id：

```
<item name="apex_data_collect_list_data_path" type="id" />
```

> 同时在findViewById出列表控件时，需立即设置dataPath

```
mGridView.setTag(R.id.apex_data_collect_list_data_path,"item#mInfo#nameInner");
```

> 此处的"item#mInfo#nameInner"，意为：

```
1. item，必填，指列表条目
2. mInfo，指item所对应的列表对象Object中的"mInfo"字段
3. nameInner，若"mInfo"也为一个Object对象，指"mInfo"对象中的"nameInner"字段
```

ExpandableListView

> setOnGroupClickListener
>
> - onGroupClick(ExpandableListView parent, View v, int groupPosition, long id)
>
> setOnChildClickListener
>
> - onChildClick(ExpandableListView parent, View v, int groupPosition, int childPosition, long id)

> 在资源文件String.xml中定义id：

```
<item name="apex_data_collect_expandable_list_group_data_path" type="id" />
<item name="apex_data_collect_expandable_list_child_data_path" type="id" />
```

> 同时在findViewById出列表控件时，需立即设置dataPath

```
mExpandableListView.setTag(R.id.apex_data_collect_expandable_list_group_data_path,
"item");
mExpandableListView.setTag(R.id.apex_data_collect_expandable_list_child_data_path,
"item");
```

> 此处的"item#mInfo#nameInner"，意为：

```
1. item，必填，指列表条目
```

OptionsMenu

> 在资源文件String.xml中定义actionViewClass：

```
<menu xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    app:actionViewClass="android.support.v7.widget.SearchView"
    tools:context="com.example.menutest.MainActivity">

    <item
        android:id="@+id/start"
        android:orderInCategory="100"
        android:title="Start"
        app:actionViewClass="android.support.v7.widget.SearchView"
        app:showAsAction="never" />
```

圈选注意事项

1. 务必先定义页面，因为Fragment可以多层嵌套，请务必准确定义
2. 列表项的唯一id是由 listId + dataKey (依赖dataPath) 而成，代码中务必给列表控件设置Tag，才能对列表条目进行别名配置