



APEX TECHNOLOGIES

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:3.2.1' // 最低2.3.0  
        classpath 'com.chinapex.analytics.aop:AnalyticsAop:1.0.4'  
    }  
}
```

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:1.0.5'

    // 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(applicationContext) // 必须为应用的context
        .setUuid("testUuid") // 可选，默认androidId
        .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("") // 可选，默认是测试url
        .setUrlInstant("") // 可选，默认是测试url
        .build());

```

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());

```

App用户登入：void signIn(String userId)

```

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

```

App用户登出: void signOut()

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