SDK使用文档：

1．导入jar包（避免生成jar包过大，仅需将src打入即可）或直接导入library工程。

2．在你的项目的AndroidManifest中添加：

<uses-permission android:name=*"android.permission.INTERNET"* />

<uses-permission android:name=*"android.permission.ACCESS\_NETWORK\_STATE"* />

<uses-permission android:name=*"android.permission.ACCESS\_WIFI\_STATE"* />

<uses-permission android:name=*"android.permission.CHANGE\_WIFI\_STATE"* />

<uses-permission android:name=*"android.permission.WAKE\_LOCK"* />

<service

android:name=*"org.openudid.OpenUDID\_service"* android:exported=*"false"* >

<intent-filter>

<action android:name=*"org.OpenUDID.GETUDID"* />

</intent-filter>

</service>

<receiver

android:name=*"com.countly.android.sdk.ReferrerReceiver"* android:exported=*"true"* >

<intent-filter>

<action android:name=*"com.android.vending.INSTALL\_REFERRER"* />

</intent-filter>

</receiver>

3.

在application中初始化

设置log是否开启：

Countly.*sharedInstance*().setLoggingEnabled(**true**);

设置是否自动统计视图（Activity、Fragment）

Countly.*sharedInstance*().setViewTracking(**true**);

初始化服务器地址、app\_key以及模式（测试数据、正式数据）

Countly.*sharedInstance*().init(**this**,"YOUR\_SERVEL\_URL","YOUR\_APPKEY",Countly.CountlyMode.*TEST*);

设置开启崩溃统计(需服务器支持post请求)

Countly.sharedInstance().enableCrashReporting();

4.

统计activity访问次数及时常(自动统计、手动统计)

Countly.*sharedInstance*().setViewTracking(**true**);时：

只需在onStart()中调用

Countly.sharedInstance().onStart(this);

在onStop()中调用

Countly.sharedInstance().onStop(this);

Countly.*sharedInstance*().setViewTracking(**false**);时

手动统计：

Countly.sharedInstance().recordView(Activity act);

或Countly.sharedInstance().recordView(String name);

5．统计Fragment访问次数及时常（自动统计、手动统计）

Countly.*sharedInstance*().setViewTracking(**true**);时：

让你的Fragement继承CountlyFragment即可。

Countly.*sharedInstance*().setViewTracking(**false**);时：

手动统计：

Countly.sharedInstance().recordView(String name);

6.事件统计（支持自定义事件）

Countly.sharedInstance().recordEvent(final String key);

需细分值及数量统计

Countly.sharedInstance().recordEvent(final String key, final int count)

Countly.sharedInstance().recordEvent(final String key, final int count, final double sum)

Countly.sharedInstance().recordEvent(final String key, final Map<String, String> segmentation, final int count)