VaultSDK Documentation

Release 1

Dayanand

May 02, 2017

Contents:

CHAPTER 1

1. Embedding the SDK into Your Application (Mandatory)

The following steps are mandatory to measure installs and sessions. You can integrate the Vault SDK either automatically using Gradle's Dependency Management or manually as an SDK.aar.

1.1 Manually include JAR In App

- Download jar from here
- Copy jar to your project's libs folder
- Add following line to your_app -> build.gradle

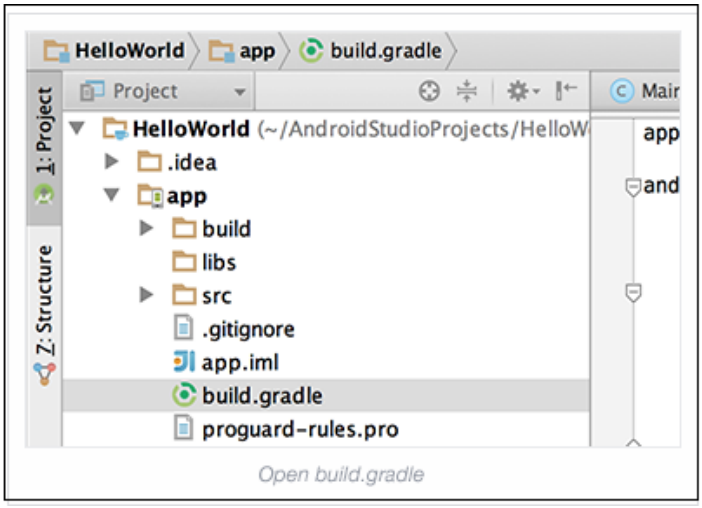
```
dependencies {
        compile fileTree(include: ['*.jar'], dir: 'libs')
}
```

- Sync and continue to *1.3* (No need to follow below step 1.2)

1.2 Add the Vault SDK to your Project

The simplest way to integrate the SDK into your project is by using Gradle's Dependency. Adding Vault's Android SDK Dependency:

1. Open your project (or create a new project), and then open your_app -> build.gradle



2. Add this to Module-level /app/build.gradle before dependencies:

```
repositories {
        mavenCentral()
}
```

3. Add the compile dependency with the latest version of the Vault SDK in the build.gradle file:

```
dependencies {
        compile 'com.rappier:vault_sdk:version_code+@aar'
}
```

version_code is latest version

1.3 Set the Required Permissions

- The AndroidManifest.xml should include the following permissions:

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

- ACCESS_WIFI_STATE and READ_PHONE_STATE permissions are optional.

```
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
```

Adding this permission enables the carrier to track IMEI.

1.4 Set an Install Referrer Broadcastreceiver in AndroidManifest.xml

receiver: Vault SDK provides a solution that broadcasts INSTALL_REFERRER to all other receivers automatically. In the AndroidManifest.xml, add the following receiver as the FIRST receiver for INSTALL_REFERRER, and ensure the receiver tag is within the application tag:

```
<receiver android:name="com.rappier.vaultsdk.vaultlib.mainsdkclasses.
↪TrackerEventReceiver" android:exported="true">
        <intent-filter>
                <action android:name="com.android.vending.INSTALL_REFERRER" />
        </intent-filter>
</receiver>
```

1.5 Embed Google Play Services into Your App

Collecting the Google Advertising ID (GAID) is essential for tracking campaigns across several channels including Facebook, Google and Twitter. To add Google Advertising ID:

Install the Google Play Services SDK and import it into your project. Add the following entry to the AndroidManifest.xml as the last entry under application (before </application>):

```
<meta-data android:name="com.google.android.gms.version"android:value="@integer/
↪google_play_services_version" />
```

OR add following in build.gradle to avoid multidex exception:

```
dependencies {
        compile 'com.google.android.gms:play-services-ads:google_play_services_version
↪'
}
```

Note:

- Vault uses the Google Play Services library for retrieving the Google Advertising ID.
- Google Play Services 7.5 is the minimum requirement for the GCM Registration Token (which is used for our Uninstall measurement). Vault recommends to always use the latest version.

CHAPTER 2

2. SDK Initialization and Installation Event (Minimum Requirement for Tracking)

Note: This is the minimum requirement to begin tracking your app installs.

2.1 Initializing the SDK

To initialize the SDK, add the following code to your Application's onCreate function:

```
Vault.initEvent(Context context, String APIkey, int AdvertiserId, String Password,␣
↪String GoogleAddId);
```

- context - use getApplicationContext()
- APIkey is unique key get after registration in Vault system
- AdvertiserId is a unique Id generated while registration in Vault system
- Password created while registration in Vault system
- GoogleAddId need to take and send from app.

Note: APIkey, AdvertiserId, Password will get while register on portal of Vault for tracking. If not already register then click here to register.

This API enables Vault to detect installations, sessions, and updates.

2.2 Reporting Background Sessions

If your app is a utility app running in the background you can use the API as described in here to measure sessions and retention within Vault.

CHAPTER 3

3. In-App Events Tracking API (Optional)

There are two types of optional events contains in Vault sdk, 1) Sdk defined 2) Custom events

3.1 Vault Sdk Defined Events

This API enables Vault to track some in app events that are already defined in sdk. App just need to call the methods of event and use it with specific events.

Events:

3.1.1 Register

If app contain registration/signup event this event will useful to track whether installed app made registration complete or not

Syntax:

```
Vault.setRegisterEvent(Context context, String APIkey, int AdvertiserId, String␣
↪Password, String GoogleAddId);
```

- context - use getApplicationContext()
- APIkey is unique key get after registration in Vault system
- AdvertiserId is a unique Id generated while registration in Vault system
- Password created while registration in Vault system
- GoogleAddId need to take and send from app.

3.1.2 Purchase

If app contain purchase event, this event will useful to track purchase

Syntax:

```
Vault.setPurchaseEvent(Context context, String APIkey, int AdvertiserId, String␣
↪Password, String GoogleAddId);
```

- context - use getApplicationContext()
- APIkey is unique key get after registration in Vault system
- AdvertiserId is a unique Id generated while registration in Vault system
- Password created while registration in Vault system
- GoogleAddId need to take and send from app.

3.2 Custom events(Advertiser defined events)

This API enables Vault to track post install events. These events are defined by the advertiser and include an event name, in addition to optional event values. Tracking in-app events helps you measure and analyze how loyal users discover your app, and attribute them to specific campaigns/media sources.

Syntax:

```
Vault setCustomEvent(Context context, String APIkey, int AdvertiserId, String␣
↪Password, String GoogleAddId, String customEventId)
```

- context - use getApplicationContext()
- APIkey is unique key get after registration in Vault system
- AdvertiserId is a unique Id generated while registration in Vault system
- Password created while registration in Vault system
- GoogleAddId need to take and send from app.
- customEventId event id generated while create custom event in Vault system