AF Android SDK version 2.3.1.17

Table of Contents

1. What Is New In this Version - IMPORTANT 2
2. Compatibility 2

Initial Steps 2

3. Embed the SDK Into Your Application 2
4. Set Permissions 2
5. Set a Receiver AndroidManifest.xml (Mandatory) 2
6. Embed Google Play Services Into your Application (Mandatory) 3

Android SDK APIs 3

7. SDK Initialization 3
   - 7.1. Set Developer Key (Dev-Key) in the SDK 3
   - 7.2. Set Currency Code (Optional) 4
   - 7.3. Enable HTTPS to HTTP fallback (Optional) 4
8. Installation Event (Minimum requirement for tracking) 4
9. In-App Conversion Events Tracking API (Optional) 4
   - 9.1. Adding Registration (Login) Event 5
   - 9.2. Adding Purchase Event 5
   - 9.3. Adding Custom In-App Event 5
   - 9.4. Rich In-App Events API 5
10. Customer User ID (Optional) 6
11. Get AppsFlyer's UID (Optional) 6
12. Accessing AppsFlyer Attribution / Conversion Data from the SDK (Deferred Deep-linking) 6
13. Track App Installs Out Of Google Play (Advanced) 6

14. End User Opt-out (Optional) 6
15. Testing the SDK Integration 7
Appendix A: AppsFlyer Rich In-App Events 7

1. What Is New In this Version - IMPORTANT

1.1 The rich in-app events API was added. See Section 9.4 and Appendix A for more info.

2. Compatibility

The AppsFlyer Android SDK is compatible with Android 2.3 and above.

Initial Steps

3. Embed the SDK Into Your Application

Add AF-Android-SDK.jar (download here) to the project's class path.

4. Set 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" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" /> *
```

* READ_PHONE_STATE permission is optional.
Adding this permission will enable carrier tracking of IMEI (required for tracking out of Google Play).

5. Set a Receiver AndroidManifest.xml (Mandatory)

Android apps cannot have multiple receivers that have the same intent-filtered action.
AppsFlyer provides a solution that broadcasts INSTALL_REFERRER to all other receivers automatically.
In the AndroidManifest.xml, please add the following receiver as the **FIRST** for **INSTALL_REFERRER:**

******** **PLEASE MAKE SURE THE RECEIVER TAG IS WITHIN THE APPLICATION TAG**

```
<receiver android:name="com.appsflyer.MultipleInstallBroadcastReceiver" android:exported="true">
  <intent-filter>
    <action android:name="com.android.vending.INSTALL_REFERRER" />
  </intent-filter>
</receiver>
```

If you would like to use multiple receivers, the Manifest.xml should look like the following:

```
<!—The AppsFlyer Install Receiver is first and will broadcast to all receivers placed below it -->
<receiver android:name="com.appsflyer.MultipleInstallBroadcastReceiver" android:exported="true">
  <intent-filter>
    <action android:name="com.android.vending.INSTALL_REFERRER" />
  </intent-filter>
</receiver>
<!—All other receivers should follow right after -->
<receiver android:name="com.google.android.apps.analytics.AnalyticsReceiver" android:exported="true">
  <intent-filter>
```

```
    <action android:name="com.android.vending.INSTALL_REFERRER" />
  </intent-filter>
</receiver>
<receiver android:name="com.admob.android.ads.analytics.InstallReceiver" android:exported="true">
     <intent-filter>
        <action android:name="com.android.vending.INSTALL_REFERRER" />
     </intent-filter>
</receiver>
```

If you would like to add an additional receiver to access app install data from your app context, please click ***here***.

Note:

If you are using other mechanisms to broadcast multiple receivers, please use the ***com.appsflyer.AppsFlyerLib*** class below. This class will not broadcast other receivers.

```
<receiver android:name="com.appsflyer.AppsFlyerLib" android:exported="true">
        <intent-filter>
          <action android:name="com.android.vending.INSTALL_REFERRER" />
        </intent-filter>
 </receiver>
```

6. Embed Google Play Services Into your Application (Mandatory)

AppsFlyer can track Google advertising ID to improve tracking. To add Google advertising ID:

6.1 Install the Google Play Services SDK and import it into your project.

6.2 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" />
```

Note:

AppsFlyer uses the Google Play Services library for retrieving the Android Advertiser ID. Although the library provides additional services, it has a light footprint if you use ProGuard as part of your build process.

Source: https://developer.android.com/google/play-services/setup.html

Android SDK APIs

7. SDK Initialization

You need to initialize the SDK on the application's first launch. Please make sure the SDK is initialized before sending the tracking events below.

7.1. Set Developer Key (Dev-Key) in the SDK

AppsFlyerLib.setAppsFlyerKey("Dev_Key ");

Replace ***Dev_Key*** with your own Dev-Key (accessible from your account. See "SDK integration" on your app screen)

7.2. Set Currency Code (Optional)

USD is the default value. Please find acceptable ISO currency codes ***here***.

Examples:

USD: ***AppsFlyerLib.setCurrencyCode("USD")***

GBP: ***AppsFlyerLib.setCurrencyCode("GBP");***

7.3. Enable HTTPS to HTTP fallback (Optional)

AF Android SDK uses HTTPS at all times, without HTTP fallback.
If you wish to allow the SDK to use HTTP fallback, please add the following setting:
AppsFlyerLib.setUseHTTPFalback(true);

Default value is false, which prevents fallback to HTTP in cases HTTPS is not available.

8. Installation Event (Minimum requirement for tracking)

This API enables AppsFlyer to detect **installations, sessions, and updates.**
This is the minimum requirement to start tracking your app installs.

Add the following line to the first launched activity's onCreate.
AppsFlyerLib.sendTracking(getApplicationContext());

Important Notes:

- The sendTracking call must be called from the first activity in launch sequence. For example, if you have a splash screen activity or a tutorial activity which is called before the main activity, the sendTracking API must be called at the beginning of this activity's onCreate function.
- In order to track app sessions (opens), Please make sure to call this API upon every app session.

9. In-App Conversion Events Tracking API (Optional)

Note: Section 9.4 has been added to the SDK and enables rich in-app events tracking. Please refer to this section to start reporting in-app events to AppsFlyer.

These events help you track how loyal users discover your app and attribute them to specific campaign/source.

- These in-app events help you track how loyal users discover your app, and attribute them to specific campaigns/media-sources. Please take the time define the event/s you would like to measure to allow you to track ROI (Return on Investment) and LTV (Lifetime Value).
- The "***sendTrackingWithEvent***" method allows you to send in-app events to AppsFlyer analytics. This method allows you to add events dynamically by adding them directly to the application code.
- Syntax:

AppsFlyerLib.sendTrackingWithEvent(getApplicationContext(),"EVENT_NAME","EVENT_REVENUE_VALUE");

- **EVENT_NAME** is any string to define the event name. For example: "registration" or "purchase"
- **EVENT_REVENUE_VALUE** is the sales value. For example: 0.99 or 0.79

9.1. Adding Registration (Login) Event

```
AppsFlyerLib.sendTrackingWithEvent(getApplicationContext(),"registration","");
```

9.2. Adding Purchase Event

Example 1: 0.99 “purchase” event:

```
AppsFlyerLib.sendTrackingWithEvent(getApplicationContext(),"purchase","0.99");
```

Example 2: 10.50 GBP “purchase” event and setting the device default currency to GBP:

```
AppsFlyerLib.setCurrencyCode("GBP");
AppsFlyerLib.sendTrackingWithEvent(getApplicationContext(),"purchase","10.50");
```

Example 3: “purchase” with no revenue:

```
AppsFlyerLib.sendTrackingWithEvent(getApplicationContext(),"purchase","");
```

9.3. Adding Custom In-App Event

Example 1: “hotel-booked” with 200 GBP revenue and setting device default currency to GBP:

```
[AppsFlyer setCurrencyCode:@"GBP"];
AppsFlyerLib.sendTrackingWithEvent(getApplicationContext(),"hotel-booked","200");
```

9.4. Rich In-App Events API

Rich in-app events help you track more complex events with multiple parameters. You can later map these rich events to the different media sources’ rich events for enhanced optimization and targeting.

Syntax:

```
public static void trackEvent(Context context, String eventName, Map<String,Object> eventValues)
```

context - use getApplicationContext()
eventName is any string to define the event name. You can find a list of recommended constant event names in Appendix A.
eventValues is a map of event parameters that comprise a rich event. You can find a list of recommended parameters in Appendix A.

Counting revenue as part of a rich in-app event: Use the ***af_revenue*** constant to count revenue as part of rich in-app events. You can populate it with any numeric value, positive or negative.

Example 1: “af_add_to_cart” of a single item with the value of 9.99 USD, content_id “234234” and of type “category_a”:

```
Map<String,Object> event = new HashMap<String,Object>();
event.put(InAppEventParameterName.PRICE,9.99);
event.put(InAppEventParameterName.CONTENT_TYPE,"category_a");
event.put(InAppEventParameterName.CONTENT_ID,"234234");
event.put(InAppEventParameterName.CURRENCY,"USD");
event.put(InAppEventParameterName.QUANTITY,1);
AppsFlyerLib.trackEvent(AppsFlyerTestActivity.this, InAppEventType.ADD_TO_CART,event);
```

For more information see **Appendix A: Rich In-App Events**

10. Customer User ID (Optional)

Setting your own customer ID will enable you to cross-reference your own unique ID with AppsFlyer's user ID and the other devices' IDs. This ID will be available at AppsFlyer CSV reports along with postbacks APIs for cross-referencing with you internal IDs.

The Customer User ID must be called during the SDK initialization before any call to the SDK. See Section #5.

Set:
AppsFlyerLib.setAppUserId("myId");
Get:
String id = AppsFlyerLib.getAppUserId();

11. Get AppsFlyer's UID (Optional)

If READ_PHONE_STATE permission is off, AppsFlyer will generate a unique ID. In order to get this ID, please use the following API:
String appsFlyerId = AppsFlyerLib.getAppsFlyerUID(this);

12. Accessing AppsFlyer Attribution / Conversion Data from the SDK (Deferred Deep-linking)

For information regarding this advanced functionality, read here.

13. Track App Installs Out Of Google Play (Advanced)

IMPORTANT NOTICE: GOOGLE PLAY IS THE DEFAULT STORE. IN CASE YOU ARE PUBLISHING YOUR APP ONLY ON GOOGLE PLAY, PLEASE SKIP THIS SECTION.

In order to track installs out of Google Play, you need to set the channel/store at the app AndroidManifest.xml with a unique channel/store name for each APK. The CHANNEL value is case sensitive.

For example:

Amazon: ***<meta-data android:name="CHANNEL" android:value="Amazon" />***

Stand alone: ***<meta-data android:name="CHANNEL" android:value="Standalone" />***

Verizon (Pre-Installed): ***<meta-data android:name="CHANNEL" android:value="Verizon" />***

You will have to configure the CHANNEL value at the AppsFlyer dashboard when setting up the app. **The meta-data tag should be placed before the </application> tag.**

14. End User Opt-out (Optional)

AppsFlyer provides you a method to opt-out specific users from AppsFlyer analytics.
This method complies with the latest privacy requirements and complies with Facebook data and privacy policies.

public static void setDeviceTrackingDisabled(boolean isDisabled)

Example:
In order to opt-out specific user from tracking please use the following method upon the first launch of the application during the SDK initialization.
AppsFlyerLib.setDeviceTrackingDisabled(true);

15. Testing the SDK Integration

How to test the SDK integration before submitting to the App Store - Read here.

Appendix A: AppsFlyer Rich In-App Events

AppsFlyer's rich in-app events provide advertisers with the ability to track any post-install event and attribute it to a Media Source and campaign.

An in-app event is comprised of a recommended event name, pre-defined recommended parameters and additional optional parameters.

Syntax:
public static void trackEvent(Context context, String eventName, Map<String,Object> eventValues)
context - use getApplicationContext()
eventName is any string to define the event name. It is recommended to use one of the constant event names defined below.
eventValues is a map of event parameters that comprise a rich event. It is recommended to use the constant events parameters as defined below (recommended structure per each defined event).

Example 1: "af_add_to_cart" of a single item with the value of 9.99 USD, content_id "234234" and of type "category_a":

```
Map<String,Object> event = new HashMap<String,Object>();
event.put(InAppEventParameterName.PRICE,9.99);
event.put(InAppEventParameterName.CONTENT_TYPE,"category_a");
event.put(InAppEventParameterName.CONTENT_ID,"234234");
event.put(InAppEventParameterName.CURRENCY,"USD");
event.put(InAppEventParameterName.QUANTITY,1);
AppsFlyerLib.trackEvent(AppsFlyerTestActivity.this, InAppEventType.ADD_TO_CART,event);
```

The following in-app events are defined as part of the ***AFInAppEventType*** interface and are recommended:

Event String Constant	Event String Name
LEVEL_ACHIEVED	af_level_achieved
ADD_PAYMENT_INFO	af_add_payment_info
ADD_TO_CART	af_add_to_cart

ADD_TO_WISH_LIST	af_add_to_wishlist
COMPLETE_REGISTRATION	af_complete_registration
TUTORIAL_COMPLETION	af_tutorial_completion
INITIATED_CHECKOUT	af_initiated_checkout
PURCHASE	af_purchase
RATE	af_rate
SEARCH	af_search
SPENT_CREDIT	af_spent_credits
ACHIEVEMENT_UNLOCKED	af_achievement_unlocked
CONTENT_VIEW	af_content_view
TRAVEL_BOOKING	af_travel_booking
SHARE	af_share
INVITE	af_invite
LOGIN	af_login
RE_ENGAGE	af_re_engage
UPDATE	af_update
OPENED_FROM_PUSH_NOTIFICATION	af_opened_from_push_notification

Custom Events:

- The Events API can receive any string as the event name.

The following section describes the recommended structure of each event type together with the parameters mappings to Facebook, Twitter and Criteo:

af_level_achieved

Description: Used to track game level events.
Recommended attributes: af_level, af_score
Facebook Mapped Event: fb_mobile_level_achieved
Twitter Mapped Event: LEVEL_ACHIEVED
Criteo Mapped Event: None
Attributes Mapping:

AppsFlyer	Facebook	Twitter	Criteo
af_level	fb_level	level	-
af_score	-	-	-

af_add_payment_info

Description: Used to track payment info configuration status.
Recommended attributes: af_success
Facebook Mapped Event: fb_mobile_add_payment_info

Twitter Mapped Event: ADDED_PAYMENT_INFO
Criteo Mapped Event: None
Attributes Mapping:

AppsFlyer	Facebook	Twitter	Criteo
af_success	fb_success	user_payment_info	-

af_add_to_cart

Description: Used to track add to cart events of specific items.
Recommended attributes: af_price, af_content_type, af_content_id, af_currency, af_quantity
Facebook Mapped Event: fb_mobile_add_to_cart
Twitter Mapped Event: ADD_TO_CART
Criteo Mapped Event: viewBasket
Attributes Mapping:

AppsFlyer	Facebook	Twitter	Criteo
af_price	_valueToSum	price_micro*	price**
af_content_type	fb_content_type	content_type	-
af_content_id	fb_content_id	content_id	item_id
af_currency	fb_currency	price_currency	-
af_quantity	-	number_items	quantity

af_add_to_wishlist

Description: Used to track add to wishlist events of specific items.
Recommended attributes: af_price, af_content_type, af_content_id, af_currency, af_quantity
Facebook Mapped Event: fb_mobile_add_to_wishlist
Twitter Mapped Event: ADD_TO_WISHLIST
Criteo Mapped Event: None
Attributes Mapping:

AppsFlyer	Facebook	Twitter	Criteo
af_price	_valueToSum	price_micro*	-
af_content_type	fb_content_type	content_type	-
af_content_id	fb_content_id	content_id	-
af_currency	fb_currency	price_currency	-
af_quantity	-	number_items	-

af_complete_registration

Description: Used to track user registration methods
Recommended attributes: af_registration_method
Facebook Mapped Event: fb_mobile_complete_registration
Twitter Mapped Event: SIGN_UP
Criteo Mapped Event: None
Attributes Mapping:

AppsFlyer	Facebook	Twitter	Criteo
af_registration_method	fb_registration_method	registration_method	-

af_tutorial_completion

Description: Used to track tutorial completions
Recommended attributes: af_success, af_content_id
Facebook Mapped Event: fb_mobile_tutorial_completion
Twitter Mapped Event: TUTORIAL_COMPLETE
Criteo Mapped Event: None
Attributes Mapping:

AppsFlyer	Facebook	Twitter	Criteo
af_success	fb_success	-	-
af_content_id	fb_content_id	content_id	-

af_initiated_checkout

Description: Used to track checkout events.
Recommended attributes: af_price, af_content_type, af_content_id, af_quantity, af_payment_info_available, af_currency
Facebook Mapped Event: fb_mobile_initiated_checkout
Twitter Mapped Event: CHECKOUT_INITIATED
Criteo Mapped Event: None
Attributes Mapping:

AppsFlyer	Facebook	Twitter	Criteo
af_price	_valueToSum	price_micro*	-
af_content_type	fb_content_type	content_type	-
af_content_id	fb_content_id	content_id	-
af_currency	fb_currency	price_currency	-
af_quantity	fb_num_items	number_items	-
af_payment_info_available	fb_payment_info_available	user_payment_info	

af_purchase

Description: Used to track purchase events (and associate revenue to them).
Recommended attributes: af_revenue, af_content_type, af_content_id, af_quantity, af_currency
Facebook Mapped Event: fb_mobile_purchase
Twitter Mapped Event: PURCHASE
Criteo Mapped Event: trackTransaction
Attributes Mapping:

AppsFlyer	Facebook	Twitter	Criteo
af_revenue*	_valueToSum	price_micro**	price***
af_content_type	fb_content_type	content_type	-
af_content_id	fb_content_id	content_id	item_id
af_currency	fb_currency	price_currency	-
af_quantity	fb_num_items	number_items	quantity
af_validated	-	-	-
af_receipt_id	-	-	-

* af_revenue will be counted as revenue in AppsFlyer's platform

af_rate

Description: Used to track app/item rating events.
Recommended attributes: af_rating_value, af_content_type, af_content_id, af_max_rating_value
Facebook Mapped Event: fb_mobile_rate
Twitter Mapped Event: RATED
Criteo Mapped Event: None
Attributes Mapping:

AppsFlyer	Facebook	Twitter	Criteo
af_rating_value	_valueToSum	price_micro	-
af_content_type	fb_content_type	content_type	-
af_content_id	fb_content_id	content_id	-
af_max_rating_value	fb_max_rating_value	max_rated_value	-

af_search

Description: Used to track search events.
Recommended attributes: af_content_type, af_search_string, af_success
Facebook Mapped Event: fb_mobile_search
Twitter Mapped Event: SEARCH
Criteo Mapped Event: viewSearch

Attributes Mapping:

AppsFlyer	Facebook	Twitter	Criteo
af_content_type	fb_content_type	content_type	-
af_search_string	fb_search_string	search_string	-
af_date_a	-	-	din
af_date_b	-	-	dout
af_destination_a	-	-	-
af_destination_b	-	-	-
af_success	fb_success	-	-

af_spent_credits

Description: Used to track credit spend events.
Recommended attributes: af_price, af_content_type, af_content_id
Facebook Mapped Event: fb_mobile_spent_credits
Twitter Mapped Event: SPENT_CREDITS
Criteo Mapped Event: trackTransaction
Attributes Mapping:

AppsFlyer	Facebook	Twitter	Criteo
af_price	_valueToSum	price_micro*	price**
af_content_type	fb_content_type	content_type	-
af_content_id	fb_content_id	content_id	item_id

af_achievement_unlocked

Description: Used to track achievement unlocking events.
Recommended attributes: af_description
Facebook Mapped Event: fb_mobile_achievement_unlocked
Twitter Mapped Event: ACHIEVEMENT_UNLOCKED
Criteo Mapped Event: None
Attributes Mapping:

AppsFlyer	Facebook	Twitter	Criteo
af_description	fb_description	description	-

af_content_view

Description: Used to track content view events.
Recommended attributes: af_price, af_content_type, af_content_id, af_currency

Facebook Mapped Event: fb_mobile_content_view
Twitter Mapped Event: CONTENT_VIEW
Criteo Mapped Event: viewProduct/viewListing
Attributes Mapping:

AppsFlyer	Facebook	Twitter	Criteo
af_price	_valueToSum	price_micro*	price
af_content_type	fb_content_type	content_type	-
af_content_id	fb_content_id	content_id	item_id
af_currency	fb_currency	price_currency	-

af_travel_booking

Description: Used to track travel booking events (and associate revenue to them).
Recommended attributes: af_revenue, af_destination_a, af_destination_b, af_class, af_description, af_customer_user_id, af_content_type, af_content_id, af_date_a, af_date_b
Facebook Mapped Event: fb_mobile_purchase
Twitter Mapped Event: PURCHASE
Criteo Mapped Event: trackTransaction
Attributes Mapping:

AppsFlyer	Facebook	Twitter	Criteo
af_revenue*	_valueToSum	price_micro**	price***
af_customer_user_id	-	-	cid
af_content_type	fb_content_type	content_type	-
af_content_id	fb_content_id	content_id	-
af_class	-	-	-
af_date_a	-	-	din
af_date_b	-	-	dout
af_destination_a	-	-	-
af_destination_b	-	-	-
af_success	fb_success	-	-

* af_revenue will be counted as revenue in AppsFlyer's platform

af_share

Description: Used to track sharing events.
Recommended attributes: af_description

Facebook Mapped Event: None
Twitter Mapped Event: SHARE
Criteo Mapped Event: None
Attributes Mapping:

AppsFlyer	Facebook	Twitter	Criteo
af_description	-	description	-

af_invite

Description: Used to track invite (social) events.
Recommended attributes: None
Facebook Mapped Event: None
Twitter Mapped Event: INVITE
Criteo Mapped Event: None
Attributes Mapping:

AppsFlyer	Facebook	Twitter	Criteo
af_description	-	description	-

af_login

Description: Used to track user login events.
Recommended attributes: None
Facebook Mapped Event: None
Twitter Mapped Event: LOGIN
Criteo Mapped Event: None

af_reengage

Description: Used to track user re-engagement events.
Recommended attributes: None
Facebook Mapped Event: None
Twitter Mapped Event: RE_ENGAGE
Criteo Mapped Event: None
Attributes Mapping:

AppsFlyer	Facebook	Twitter	Criteo
af_description	-	description	-

af_opened_from_push_notification

Description: Used to track app opens from push notification events.
Recommended attributes: None
Facebook Mapped Event: None
Twitter Mapped Event: None

Criteo Mapped Event: None

af_update

Description: Used to track update events.
Recommended attributes: af_content_id
Facebook Mapped Event: None
Twitter Mapped Event: UPDATE
Criteo Mapped Event: None
Attributes Mapping:

AppsFlyer	Facebook	Twitter	Criteo
af_content_id	-	content_id	-

Optional Parameters:

In addition to the recommended parameters that could be passed with each event, the parameters below are defined as part of the ***AFInAppEventParameterName*** interface and can be sent as part of the event dictionary value:

Parameter String Constant	Parameter String Name	Recommended Value Type
REVENUE	af_revenue*	Float
PRICE	af_price	Float
LEVEL	af_level	Int
SUCCESS	af_success	Boolean
CONTENT_TYPE	af_content_type	String
CONTENT_ID	af_content_id	String
CURRENCY	af_currency	String
REGISTRATION_METHOD	af_registration_method	String
QUANTITY	af_quantity	Int
PAYMENT_INFO_AVAILABLE	af_payment_info_available	Boolean
RATING_VALUE	af_rating_value	Float
MAX_RATING_VALUE	af_max_rating_value	Float
SEARCH_STRING	af_search_string	String
DESCRIPTION	af_description	String
SCORE	af_score	Int
DESTINATION_A	af_destination_a	String
DESTINATION_B	af_destination_b	String
CLASS	af_class	String
DATE_A	af_date_a	String
DATE_B	af_date_b	String
EVENT_START	af_event_start	Unixtime
EVENT_END	af_event_end	Unixtime

LATITUDE	af_lat	Int
LONGITUDE	af_long	Int
CUSTOMER_USER_ID	af_customer_user_id	String
VALIDATED	af_validated	String
RECEIPT_ID	af_receipt_id	String
PARAM_1	af_param_1	String
PARAM_2	af_param_2	String
PARAM_3	af_param_3	String
PARAM_4	af_param_4	String
PARAM_5	af_param_5	String
PARAM_6	af_param_6	String
PARAM_7	af_param_7	String
PARAM_8	af_param_8	String
PARAM_9	af_param_9	String
PARAM_10	af_param_10	String

* af_revenue is the only parameter that is used for revenue calculations. Use it for events that actually represent revenue generation in your business logic. You can use af_price as a monetary parameter that will not be counted as revenue (such as in an “Add to Cart” event).