Gamemaker Studio: AppsFlyer

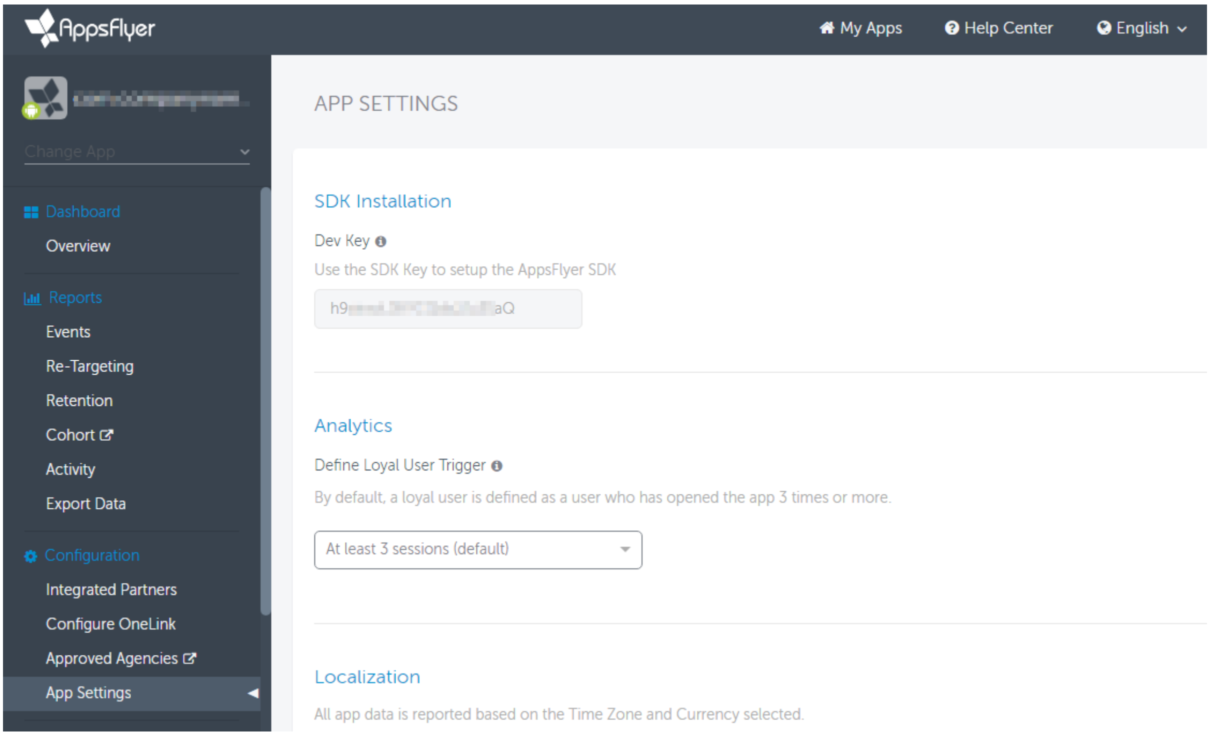
Reference Guide Revision 1.0.1

Contents

Set up

AppsFlyer Backend

If you haven't already, sign up for AppsFlyer and create an app. You must select either "available in the app store" or "Pending approval or unpublished", otherwise the extension will not work. Select Google Play, and ensure that the Android package name is the same that it is in Gamemaker.



APP SETTINGS

SDK Installation

Dev Key

Use the SDK Key to setup the AppsFlyer SDK

h9[redacted]aQ

Analytics

Define Loyal User Trigger

By default, a loyal user is defined as a user who has opened the app 3 times or more.

At least 3 sessions (default)

Localization

All app data is reported based on the Time Zone and Currency selected.

Once you have created the app, on the sidebar, click "App Settings". Copy the Dev Key, and in the extensions section in Gamemaker, under the AppsFlyer extension, change the AppsFlyerID constant to your developer ID.

SDK Manager/Licensing Agreement

If you have problems when compiling to do with accepting a license agreement, open the Android SDK manager, and update:

- Android support library
- Android support repository
- Google Play services

Minimum SDK

You must set your minimum SDK in Android global game configuration to 14

Java Ram Allocation

You will most likely require a larger heap size than default to compile the extension. In Gamemaker, go to File > Preferences > Android, set the heap allocation to something like 4 (=4096mb of ram), physical ram permitting.

Automatic Tracking

The extension automatically tracks installs and deep linking events, there is no need to do any extra work for this.

Events

Events are a way to track the user’s actions after they install the app. An event consists of an event type, and several parameters that describe the event. See the sections below for more types of events and parameters. Check the AppsFlyer documentation to see what parameters are required for each type.

To call an event say when the user passes a level in a game:

```
appsflyer_event("af_level_achieved", dm(
      "af_level", global.user_level,
      "af_score", global.user_score
));
```

This will send an event that the user passed the level stored in global.user_level and scored global.user_score.

When calling the appsflyer_event() function, in the example above, you will notice the second parameter is a function named dm(). This function is a shortcut to create a ds map in the format of dm(key,value,[key2,value2,etc…]), and the appsflyer_event() function frees the memory used by the map after it has been called.

Special Events

The extension comes with special events built in.

In-app purchases

In-app purchases and their specific information can be tracked automatically. Simply call the following function in the In-App asynchronous event of an object that persists throughout the game:

```
appsflyer_event_automatic_iap()
```

Additional Events

There are many more custom events you can implement. See the list of strings below and check the following article to read about their requirements: https://support.appsflyer.com/hc/en-us/articles/207551843 .

Event Types

af_level_achieved
af_add_payment_info
af_add_to_cart
af_add_to_wishlist
af_complete_registration
af_tutorial_completion
af_initiated_checkout
af_purchase
af_rate
af_search
af_spent_credits
af_achievement_unlocked
af_content_view
af_travel_booking
af_share
af_invite
af_login
af_re_engage
af_update
af_opened_from_push_notification
af_location_changed
af_location_coordinates

Event Parameters/Values

af_level
af_score
af_success
af_price
af_content_type
af_content_id
af_content_list
af_currency
af_quantity
af_registration_method
af_payment_info_available
af_max_rating_value
af_rating_value
af_search_string
af_date_a
af_date_b
af_destination_a
af_destination_b
af_description
af_class
af_event_start
af_event_end
af_lat
af_long
af_customer_user_id
af_validated
af_revenue
af_receipt_id
af_param_1
af_param_2
af_param_3
af_param_4
af_param_5
af_param_6
af_param_7
af_param_8
af_param_9
af_param_10