AF iOS SDK version 2.5.3.15

Table of Contents

1. What Is New In this Version - IMPORTANT

1.1 Added in-app events constants

1.2 Receipt validation update to a rich in-app event

2. Compatibility

The AppsFlyer iOS SDK is compatible with all iOS devices (iPhone, iPod, iPad) with iOS version 6.0 and above.

3. Introduction

AppsFlyer's SDK provides application installation and event tracking functionality. While installs, updates, and sessions tracking are done by default (see Section 6), developers may also track additional in-app events beyond app installs (including in-app purchases, game levels, etc.) to evaluate ROI and user engagement levels.

Initial Steps

4. Embed the SDK into Your Application

The simplest way to integrate the iOS SDK is using cocoapods:

4.1 Add the following line to your pod file: ***pod 'AppsFlyer-SDK'***

This command will download AppsFlyer's latest SDK (***libAppsFlyerLib.a*** & ***AppsFlyerTracker.h***) and copy it to the your workspace (project).

If you don't use cocoapods, follow the steps below:

4.2 Add the header and lib files to your project.
AppsFlyer SDK components (***AppsFlyerTracker.h & libAppsFlyerLib.a***) are available here.

4.3 Add the AppsFlyer header import:

```
#import "AppsFlyerTracker.h"
```

4.4 Add the **AdSupport.framework** to your project and set it as "Optional".
Please make sure you add **AdSupport.framework** to your project and follow the submission instructions **here.**

Note: AppsFlyer will collect IDFA only if you include AdSupport.framework.

4.5 Add the **iAd.framework** to your project

iOS SDK APIs

5. SDK Initialization

You need to initialize the SDK on the application first launch. Please make sure that the SDK is initialized before sending the tracking events below.
Add the following ***code*** to your **"didFinishLaunchingWithOptions"** function:

5.1. Set Your App_ID and Dev_Key

// Replace App_ID & Dev_Key with your values

// **App_ID** = Your iTunes App You can get your AppsFlyer dev key on our dashboard. See "SDK integration" on your app screen.

ID, e.g. **34320065**

// **Dev_Key** = Your unique developer ID, which is accessible from your account, e.g. R5AKpQjiER6LzSQ6iX

```
// For example:

// [AppsFlyerTracker sharedTracker].appsFlyerDevKey = @"R5AKpQjiER6LzSQ6iX";
// [AppsFlyerTracker sharedTracker].appleAppID = @"343200656";

[AppsFlyerTracker sharedTracker].appsFlyerDevKey = @"REPLACE THIS WITH YOUR Dev_Key";
[AppsFlyerTracker sharedTracker].appleAppID = @"REPLACE THIS WITH YOUR App_ID";
```

5.2. Set Currency Code (Optional)

// USD is default value. Acceptable ISO Currency codes here. Examples:

```
// British Pound:   [[AppsFlyerTracker sharedTracker].currencyCode = @"GBP"];
// US Dollar:   [[AppsFlyerTracker sharedTracker].currencyCode = @"USD"];
 [[AppsFlyerTracker sharedTracker].currencyCode = @"USD"];
```

5.3. Set Customer User ID (Advanced)

You can set your own customer id, which will be added to the reports. **See Section 8 for more details**.

```
// [[AppsFlyerTracker sharedTracker].customerUserID =@"YOUR_CUSTOM_DEVICE_ID"];
```

5.4. Set HTTPS (Advanced)

AppsFlyer's SDK communicates with its servers over HTTPS. In case you chose to disable that (not recommended), you can set the **isHTTPS** property to NO. The default is YES.

```
// [AppsFlyerTracker sharedTracker].isHTTPS = YES;
```

6. Installation Event (Minimum requirement to enable tracking)

This API enables AppsFlyer to detect **installations, sessions (app opens), and updates.**

This is the minimum requirement to start tracking your app installs.

// Add the following ***code*** to your *AppDelegate.m* source file at "**applicationDidBecomeActive"** function

```
#import "AppsFlyerTracker.h"
-(void)applicationDidBecomeActive:(UIApplication *)application
{
        // Track Installs, updates & sessions(app opens) (You must include this API to enable tracking)
         [[AppsFlyerTracker sharedTracker] trackAppLaunch];
}
```

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

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

- 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 "trackEvent" 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.
- You are also able to track more complex events which contain multiple parameters.

Syntax:

```
- (void) trackEvent:(NSString *)eventName withValues:(NSDictionary*)values
```

eventName is any string to define the event name. You can find a list of recommended constant event names in Appendix A.
values is a dictionary 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":

```
[[AppsFlyerTracker sharedTracker] trackEvent:AFEventAddToCart withValues:@{
                AFEventParamPrice: @9.99,
                AFEventParamContentType : @"category_a",
                AFEventParamContentId: @"234234",
                AFEventParamCurrency : @"USD",
                AFEventParamQuantity : @1
}];
```

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

8. Customer User ID (Advanced)

Setting your own custom 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's CSV reports along with postbacks APIs for cross-referencing with your internal IDs. The ID must be set during the first launch of the app at the SDK initialization.

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

```
[[AppsFlyerTracker sharedTracker].customerUserID =@"YOUR_CUSTOM_DEVICE_ID"];
```

9. Get AppsFlyer's Unique Device UID (Advanced)

Get AppsFlyer's proprietary device ID. AppsFlyer's device ID is the main ID used by AppsFlyer in the Reports and APIs.

```
(NSString *) [AppsFlyerTracker sharedTracker] getAppsFlyerUID
```

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

For information regarding this advanced functionality, read here

11. End User Opt-out (Advanced)

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.

[AppsFlyerTracker sharedTracker].deviceTrackingDisabled = YES;
Default is NO, meaning tracking is enabled by default.
Please use this API during the SDK initialization in Section 5.

12. Explicit opt-out from ID for Advertisers – IDFA/IFA (Optional)

AF SDK will collect IDFA only if AdSupport.framework library is included in your project. No need to explicitly opt-in or opt-out. However, in case you would like to explicitly opt-out IDFA please use the following API during the SDK initialization in Section 5:

```
// Not recommended unless instructed otherwise.
 // [AppsFlyerTracker sharedTracker].disableAppleAdSupportTracking = YES;
```

13. Testing the SDK Integration

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

14. In-App Purchase Receipt Validation

AppsFlyer's SDK can provide in-app purchase server verification. To set receipt validation tracking you need to call the **validateAndTrackInAppPurchase** method inside the SKStoreKit's **completeTransaction:** callback. This call will automatically generate an "af_purchase" in-app event.

```
[[AppsFlyerTracker sharedTracker] validateAndTrackInAppPurchase:@"in-app-purchase-success"
            eventNameIfFailed:@"in-app-purchase-failed"
            withValue:@”YOUR_VALUE”
            withProduct:@”YOUR_PRODUCT_IDENTIFIER”
            price:@”PRODUCT_PRICE”
            success:^(NSDictionary *result){ }
            failure:^(NSError *error, id response) { }
}];
```

This call has two callback blocks, one for 'success' and one for 'failure' (for any reason, including validation fail). On success, a dictionary will be returned with the receipt validation data provided by Apple's servers.

Important: For testing purposes, we recommend to set the *useReceiptValidationSandbox* **flag to YES, as this will redirect the requests to Apple sandbox servers.**

```
[AppsFlyerTracker sharedTracker].useReceiptValidationSandbox = YES;
```

Example:

```
- (void)completeTransaction:(SKPaymentTransaction*)transaction {
            //getProductByIdFromCache is your implementation for  caching your SKProduct items

            SKProduct *product = [self getProductByIdFromCache: transaction.payment.productIdentifier];

            [[AppsFlyerTracker sharedTracker] validateAndTrackInAppPurchase:@"in-app-purchase-success"
                              eventNameIfFailed:@"in-app-purchase-failed"
                              withValue:[NSString stringWithFormat:@"price: %@", product.price]
                             withProduct:product.productIdentifier
                             price:product.price
                             success:^(NSDictionary *result){

            } failure:^(NSError *error, id response) {
                          NSLog(@"response = %@", response);
            }];
}
```

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:

```
- (void) trackEvent:(NSString *)eventName withValues:(NSDictionary*)values
```

eventName is any string to define the event name. You can find a list of recommended constant event names in Appendix A.
values is a dictionary of event parameters that comprise a rich event. You can find a list of recommended parameters in Appendix A.

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":

```
[[AppsFlyerTracker sharedTracker] trackEvent:AFEventAddToCart withValues:@{
            AFEventParamPrice: @9.99,
            AFEventParamContentType : @"category_a",
            AFEventParamContentId: @"234234",
            AFEventParamCurrency : @"USD",
            AFEventParamQuantity : @1
}];
```

The following in-app events types are defined and recommended:

Event String Constant	Event String Name
AFEventLevelAchieved	af_level_achieved
AFEventAddPaymentInfo	af_add_payment_info
AFEventAddToCart	af_add_to_cart
AFEventAddToWishlist	af_add_to_wishlist
AFEventCompleteRegistration	af_complete_registration
AFEventTutorialCompletion	af_tutorial_completion
AFEventInitiatedCheckout	af_initiated_checkout
AFEventPurchase	af_purchase
AFEventRate	af_rate
AFEventSearch	af_search
AFEventSpentCredits	af_spent_credits
AFEventAchievementUnlocked	af_achievement_unlocked
AFEventContentView	af_content_view
AFEventTravelBooking	af_travel_booking
AFEventShare	af_share
AFEventInvite	af_invite
AFEventLogin	af_login
AFEventReEngage	af_re_engage

AFEventUpdate	af_update
AFEventOpenedFromPushNotification	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 and can be sent as part of the event dictionary value:

Parameter String Constant	Parameter String Name	Recommended Value Type
AFEventParamRevenue	af_revenue*	Float
AFEventParamPrice	af_price	Float
AFEventParamLevel	af_level	Int
AFEventParamSuccess	af_success	Boolean
AFEventParamContentType	af_content_type	String
AFEventParamContentId	af_content_id	String
AFEventParamCurrency	af_currency	String
AFEventParamRegistrationMethod	af_registration_method	String
AFEventParamQuantity	af_quantity	Int
AFEventParamPaymentInfoAvailable	af_payment_info_available	Boolean
AFEventParamRatingValue	af_rating_value	Float
AFEventParamMaxRatingValue	af_max_rating_value	Float
AFEventParamSearchString	af_search_string	String
AFEventParamDescription	af_description	String
AFEventParamScore	af_score	Int
AFEventParamDestinationA	af_destination_a	String
AFEventParamDestinationB	af_destination_b	String
AFEventParamClass	af_class	String
AFEventParamDateA	af_date_a	String
AFEventParamDateB	af_date_b	String

AFEventParamEventStart	af_event_start	Unixtime
AFEventParamEventEnd	af_event_end	Unixtime
AFEventParamLat	af_lat	Int
AFEventParamLong	af_long	Int
AFEventParamCustomerUserId	af_customer_user_id	String
AFEventParamValidated	af_validated	String
AFEventParamReceiptId	af_receipt_id	String
AFEventParam1	af_param_1	String
AFEventParam2	af_param_2	String
AFEventParam3	af_param_3	String
AFEventParam4	af_param_4	String
AFEventParam5	af_param_5	String
AFEventParam6	af_param_6	String
AFEventParam7	af_param_7	String
AFEventParam8	af_param_8	String
AFEventParam9	af_param_9	String
AFEventParam10	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).