

Mobile Social Plugin

Anyone can comment

[Short Overview](#)

[Getting Started](#)

[Twitter Setup](#)

[Facebook Setup](#)

[How to get support](#)

Short Overview

Plugin provides the functionality to use Twitter and Facebook api on IOS and Android devices. You can write code once and be sure that it will work the same on your IOS and Android app. However some API call may looks different, on IOS / Android even if it do the same for example posting/authorization. You can find out more about differences in API References.

Twitter API includes:

- User Aunification
- Loading User Data
 - id
 - name
 - description
 - screen_name
 - location
 - lang
 - status
 - profile images urls
 - profile images
 - and much more
- Posting
- Posting with image

Facebook API:

Provided facebook api as wrapper class around [Unity Facebook SDK](#). With means all Unity Facebook SDK will be completely available for you.

Wrapper include API for:

- User Aunification
- Loading User Data
 - id
 - name
 - first_name
 - last_name
 - email

- location
 - locale
 - profile images urls
 - profile images
 - and much more
- Posting
- Posting with image
- Loading friends information

Getting Started

Twitter Setup

In order to implement twitter oAuth in your application you need twitter **consumer key** and **consumer secret** which are used to make twitter API calls. So register a new twitter application and get the keys

1. Go to <https://dev.twitter.com/apps/new> and register new application. Fill application name, description and website.

2. Give some **dummy url in the callback url field** to make the app as browser app. (If you leave it as blank it will act as Desktop app which won't work in mobile device)

3. Under the **settings** tab upload icon and change the access type to **Read and Write**.

The screenshot shows the 'Application details' form on the Twitter developer portal. It includes fields for Name, Description, Website, and Callback URL, each with a red asterisk indicating it is required. The 'Name' field contains 'AndroidNativeApp', 'Description' contains 'test app', and 'Website' contains 'http://mywebsite.com/'. The 'Callback URL' field contains 'http://mywebsite.com/callback/'. A red arrow points to the 'Callback URL' field, and a red text annotation 'any url, we not going to use it' is placed next to it. Below the form, there is a note about OAuth 1.0a applications and a warning about restricting the application from using callbacks.

Application details

Name *

AndroidNativeApp

Your application name. This is used to attribute the source of a tweet and in user-facing authorization screens. 32 characters max.

Description *

test app

Your application description, which will be shown in user-facing authorization screens. Between 10 and 200 characters max.

Website *

http://mywebsite.com/

Your application's publicly accessible home page, where users can go to download, make use of, or find out more information about your application. This fully-qualified URL is used in the source attribution for tweets created by your application and will be shown in user-facing authorization screens. (If you don't have a URL yet, just put a placeholder here but remember to change it later.)

Callback URL

http://mywebsite.com/callback/


Where should we return after successfully authenticating? [OAuth 1.0a](#) applications should explicitly specify their oauth_callback URL on the request token step, regardless of the value given here. To restrict your application from using callbacks, leave this field blank.

any url, we not going to use it

4. Copy **Consumer Key & Consumer Secret key**

Application settings

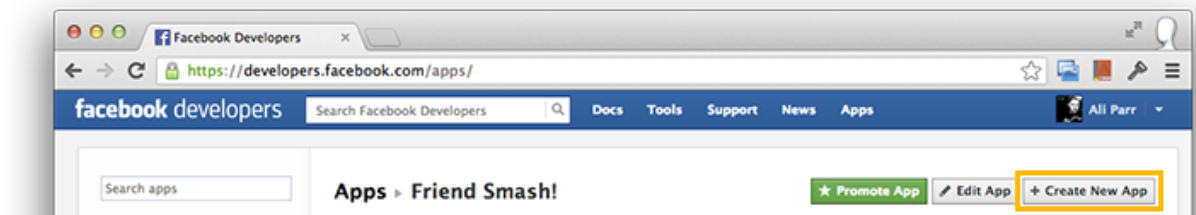
Keep the "API secret" a secret. This key should never be human-readable in your application.

API key	wEvDyAUr2QabVAsWPDiGwg
API secret	igRxZbOrkLQPNLSvibNC3mdNJ5tOIVOPH3HNNKDY0
Access level	Read, write, and direct messages (modify app permissions)
Owner	LacostSt
Owner ID	

You done. Now you can start using Twitter API in your project using copied secret keys.

Facebook Setup

Step 1: Navigate to [your App Dashboard](#) and click '+ Create New App'. In the popup dialog, give your new app a name and optionally a unique namespace, click 'Continue' and follow the subsequent instructions.



Create New App

App Name [?]

Unity Example App

Valid

App Namespace [?]

unityexample

Available

App Category [?]

Other

Choose a sub-category

Web Hosting [?]

☐ Yes, I would like free web hosting provided by Heroku ([Learn More](#))

By proceeding, you agree to the [Facebook Platform policies](#)

Continue

Cancel

Step 2: Once completed, you'll see your app's Basic Settings page. Here you can find your **App ID**, which is required within your Unity game's configuration, in order to integrate with Facebook.

Apps ▸ Unity Example App ▸ Basic



Unity Example App

App ID: 412806198828612

App Secret: 39102f9d731c6f33ca636575793e25b2 ([reset](#))

● This app is in Sandbox Mode (Only visible to Admins, Developers and Testers)

Basic info

Display Name: [?]

Unity Example App

Namespace: [?]

unityexample

Contact Email: [?]

App Domains: [?]

Enter your site domains and press enter

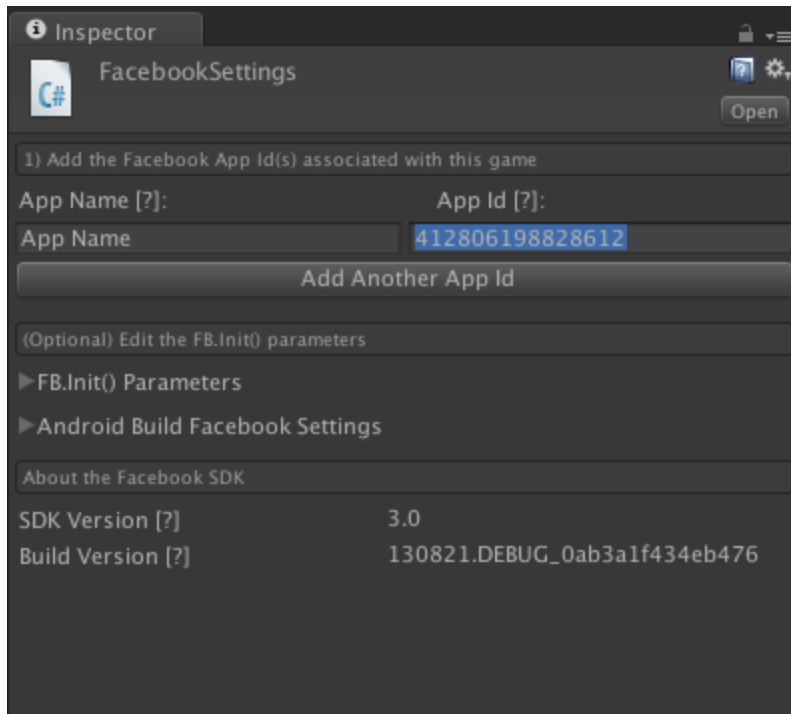
Hosting URL: [?]

You have not generated a URL through one of our partners ([Get one](#))

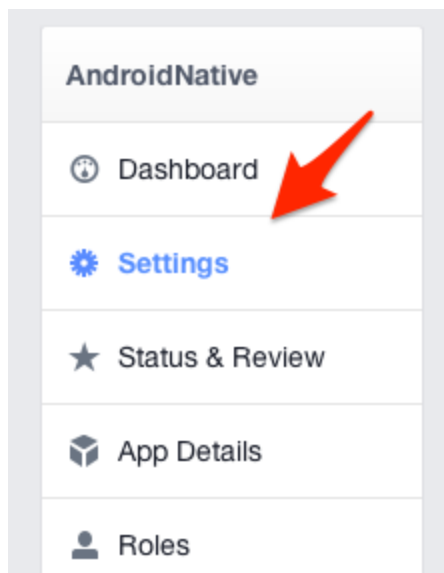
Sandbox Mode: [?]

☒ Enabled
 ☐ Disabled

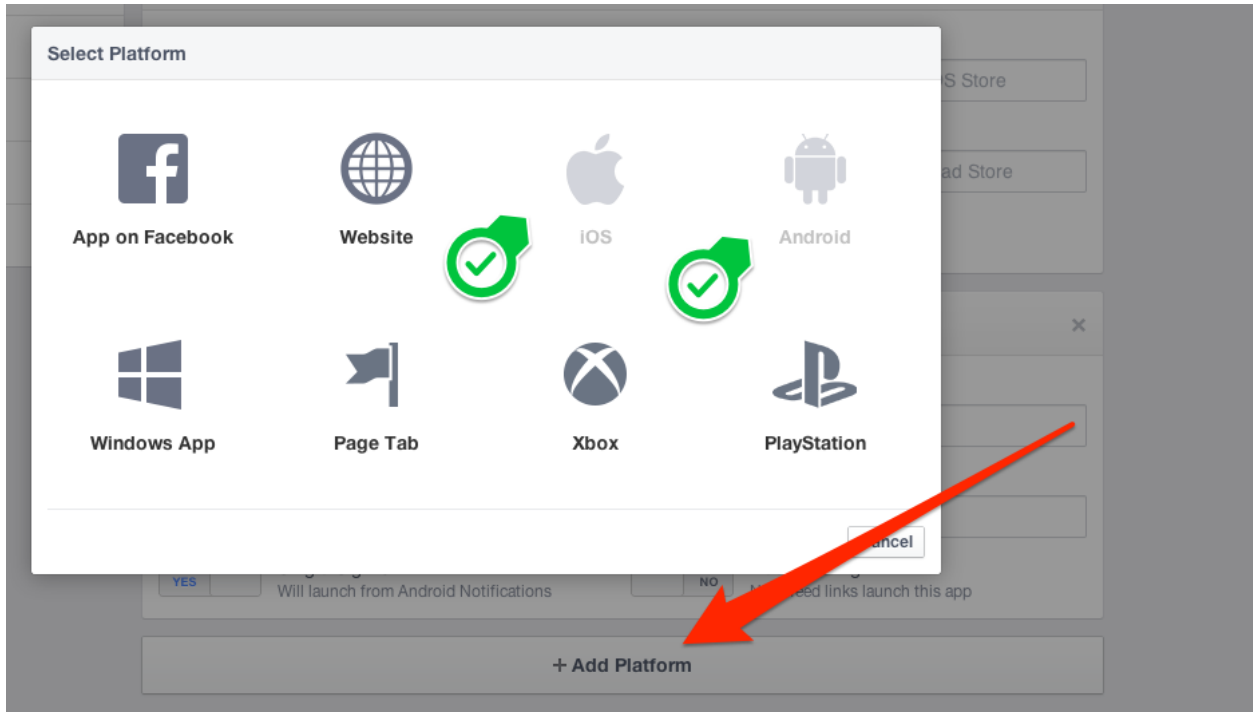
Step 3: Open project with imported plugin. Select 'Edit Settings' from the 'Facebook' menu.



Goro your app settings on facebook



Click on Add Platforms and add IOS and Android platform



IOS Set up Part:

Set your app bundle Id. You can also specify iPhone and iPad Store ID if you have one.

A screenshot of the 'IOS' configuration panel. It features several input fields: 'Bundle ID' with the value 'com.unionassets.android.plugin' and a green checkmark icon; 'URL Scheme Suffix (Optional)' which is empty; 'iPhone Store ID' with the placeholder 'The ID to identify your app in the iOS Store'; and 'iPad Store ID' with the placeholder 'The ID to identify your app in the iPad Store'. At the bottom, there is a 'Single Sign On' section with a 'NO' button and the text 'Will launch from iOS Notifications'.

Android Part.

Set your app bundle id (Package Name). Use **com.android.AndroidNativeBridge** as Class Name. Set Key hash for you app

Android

Package Name

com.unionassets.android.plugin

Class Name

com.android.AndroidNativeBridge

Key Hashes

PWD6Z/cftKr+7cm85XzST+U2Etg= x

YES

Single Sign On
Will launch from Android Notifications

NO

Deep Linking
Newsfeed links launch this app

You can use Debug Key Hash or get one from your app keystore Keystore.
To get Debug Key Hash use:

```
keytool -exportcert -alias androiddebugkey -keystore ~/.android/debug.keystore | openssl  
sha1 -binary | openssl base64
```

password: **android.**

If you already your keystore, use this to get Key Hash:

```
keytool -exportcert -alias <alias_name> -keystore <path_to_keystore> | openssl sha1  
-binary | openssl base64
```

If you don't you can use [this instruction](#) to create your app keystore.

That's it you can now go to API Reference section to find out how to use Facebook and Twitter API in your app.

