

# Convert profile to page

Due to Facebook's privacy policy, we are not able to use the addon to get all of your posts from the personal profile, only from a public page, as posts from a personal profile are protected for privacy reasons of Facebook.

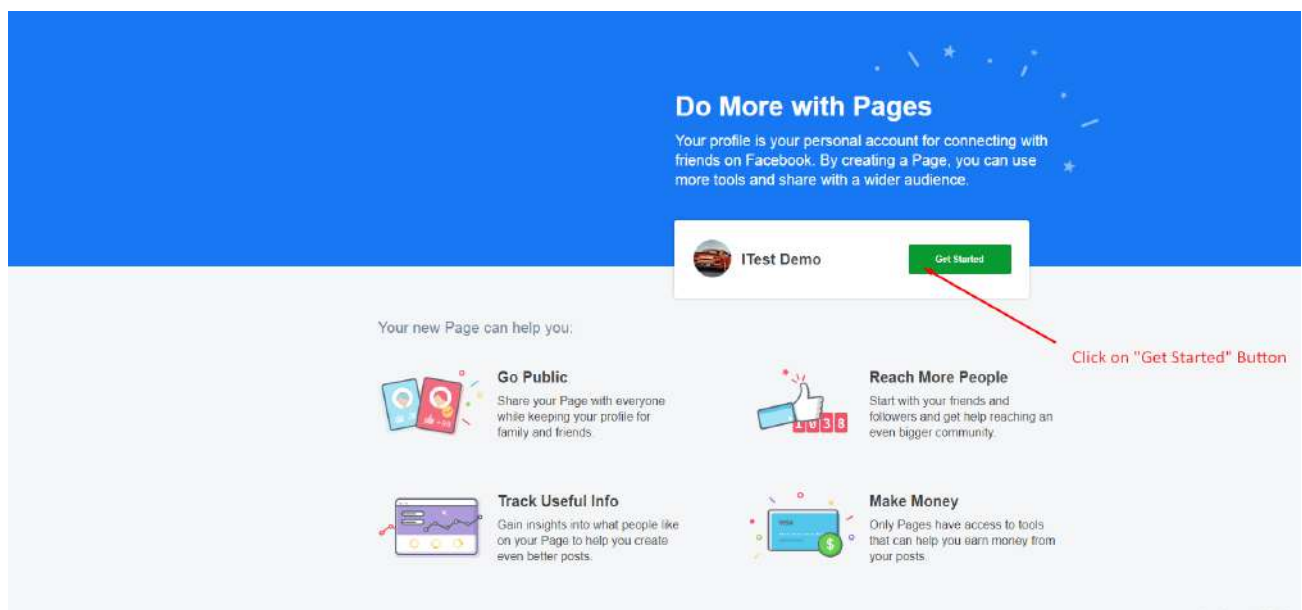
If you're using the profile to represent a business, product, organization, public figure, or something else other than yourself, then we had advise converting your profile to a page per Facebook's recommendation:

<http://www.facebook.com/help/175644189234902/>

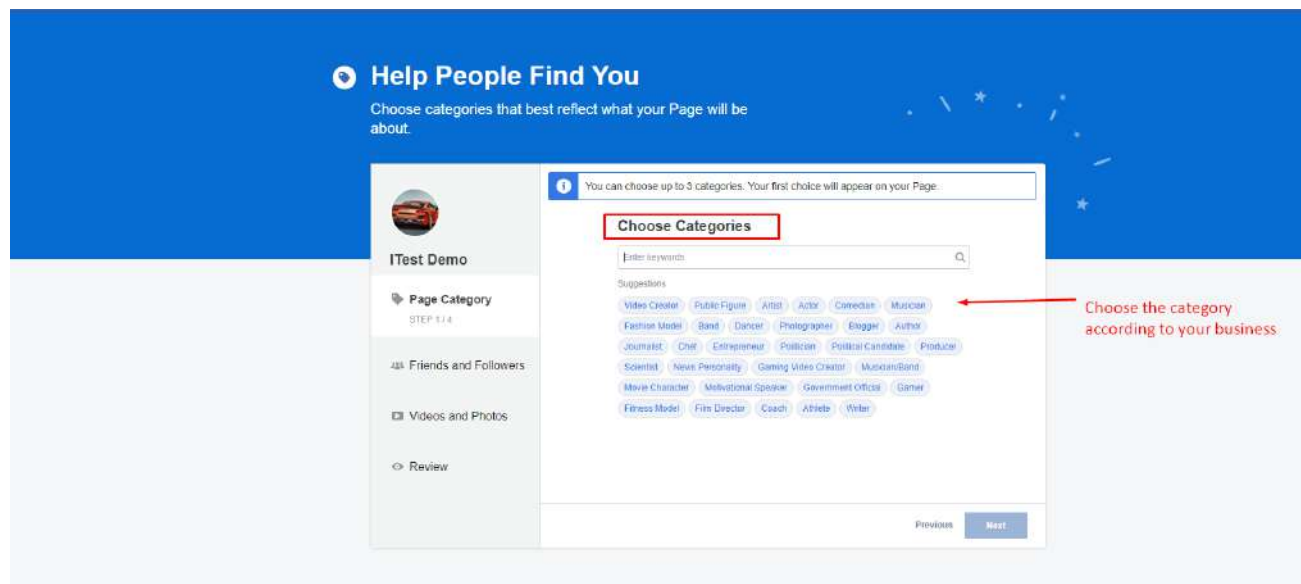
There are many advantages to using pages over the profiles. It is in fact against Facebook policies to use a profile to represent something other than yourself. See "Why should I convert my profile to a page" in the above link.

## Steps to create the page from Profile.

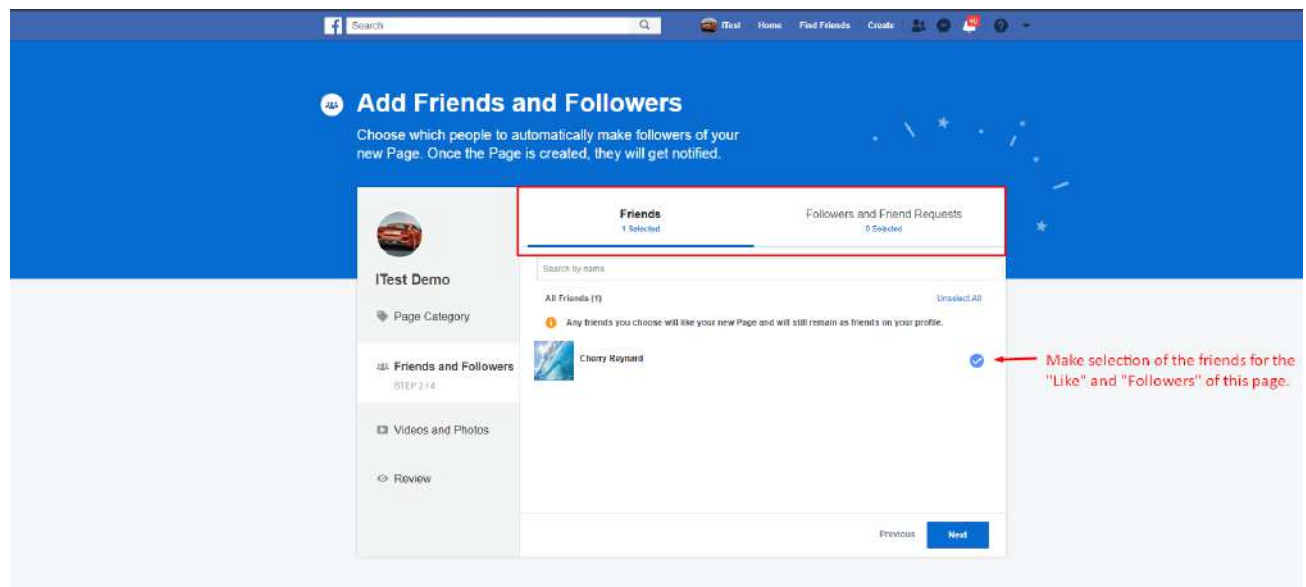
1. Go to [Create a Facebook Page Based on Your Profile](#).
2. Click on the "Get Started" button.



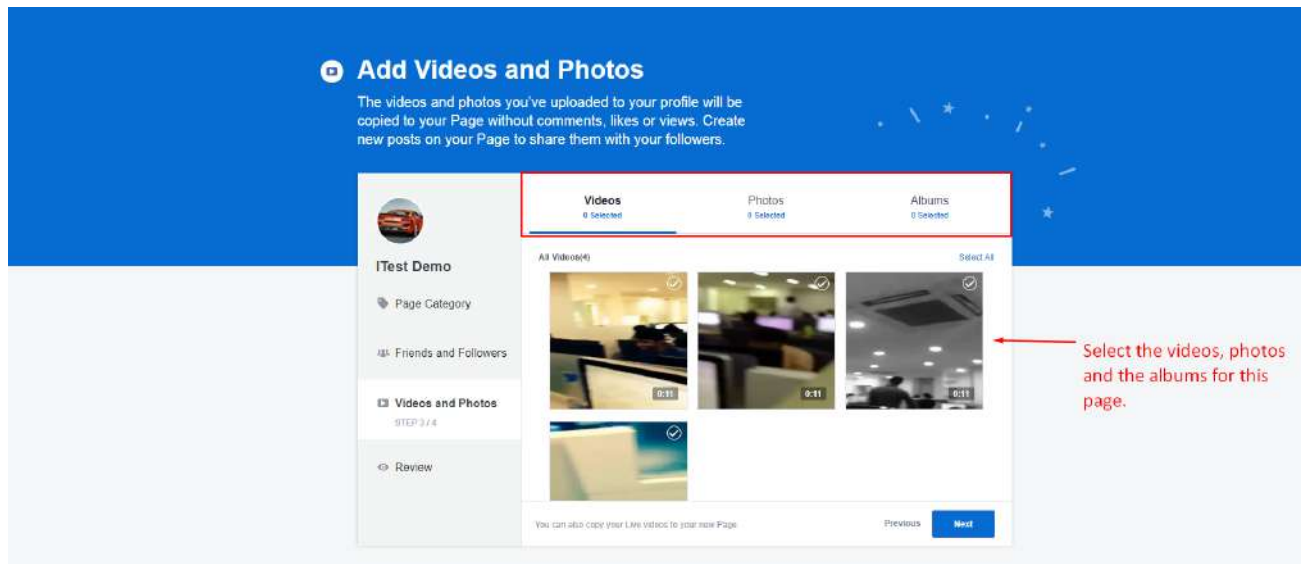
### 3. Select the appropriate category for your business page.



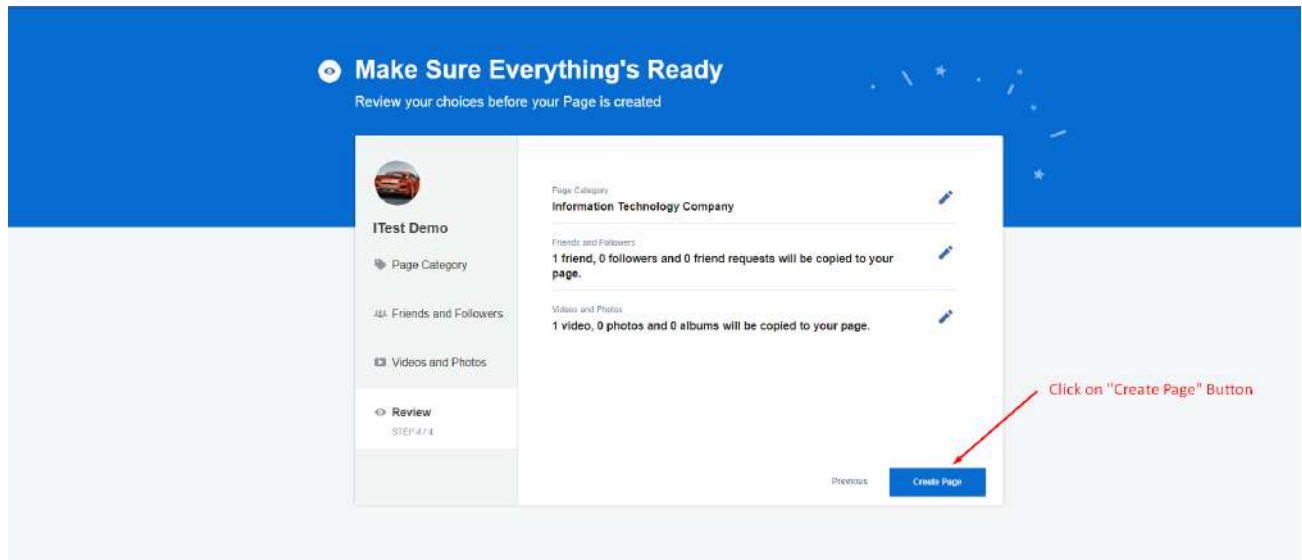
### 4. Select the "Friends" and "Followers" for your page.



5. Select the "Videos", "Photos" and the "Albums" for your page.



6. Finally, click on the "Create Page" button.



Once you have converted to a page then the addon will be able to retrieve and display all of your posts.

## Link your instagram to facebook

**Before you start you need to do something very first to work with the Instagram feed.**

Essentials setups you need.

1. You have an Instagram creator account or Instagram business account.
2. Need to connect the Facebook page to that account.



## **Set up a Business Account on Instagram.**

First of all, download the Instagram app on your mobile and create an account.

Follow the steps to create the Instagram account.

If you have a personal account then convert it to a business account. It is a must to have a business account to link with Facebook.

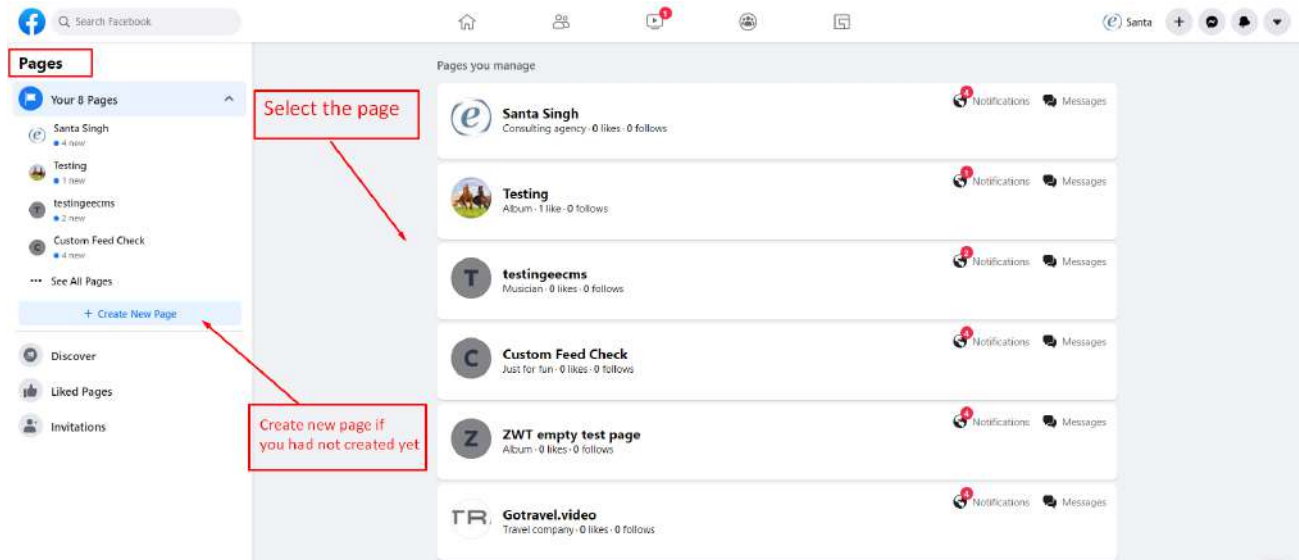
To convert your personal profile to a business account.

1. First, go to the profile and tap on the  icon in upper right corner.
2. Tap on settings  icon.
3. Tap on the account.
4. Tap switch on Switch to professional account.
5. Tap on business.
6. If you'd like, you can follow the steps to connect your business account to a Facebook Page associated with your business. This step is optional and will make it easier to use all of the features available for businesses across the Facebook family of apps. At this time, only one Facebook Page can be connected to your business account.
7. Add your business category and contact information.

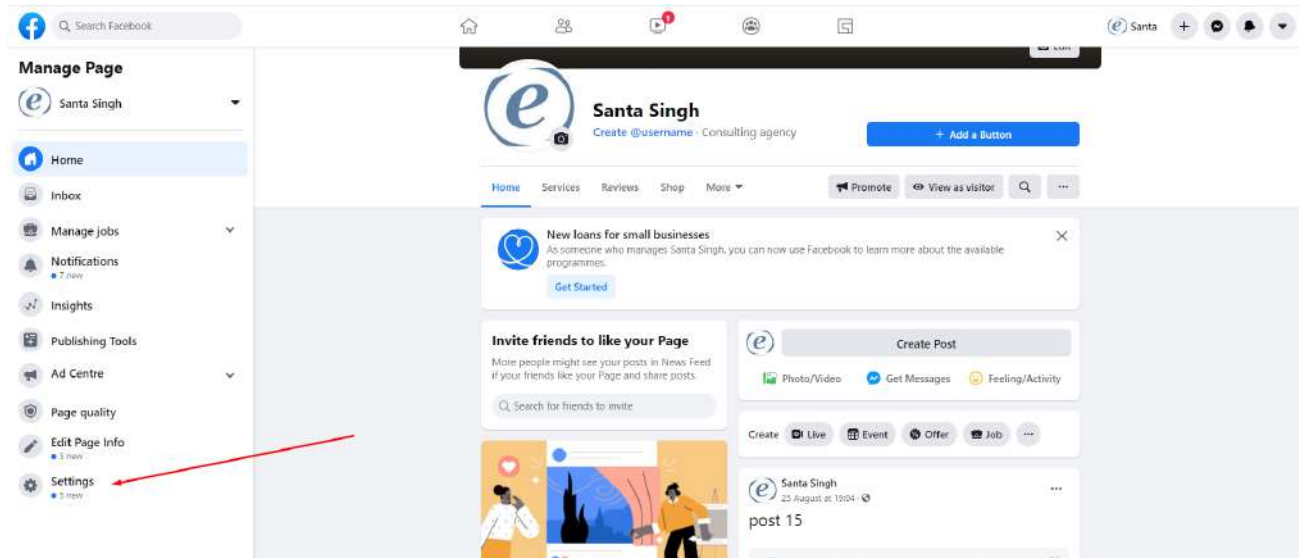
Now Instagram setup is completed. Move to the further steps to connect with the Facebook account.

## **Connect Facebook page to the account.**

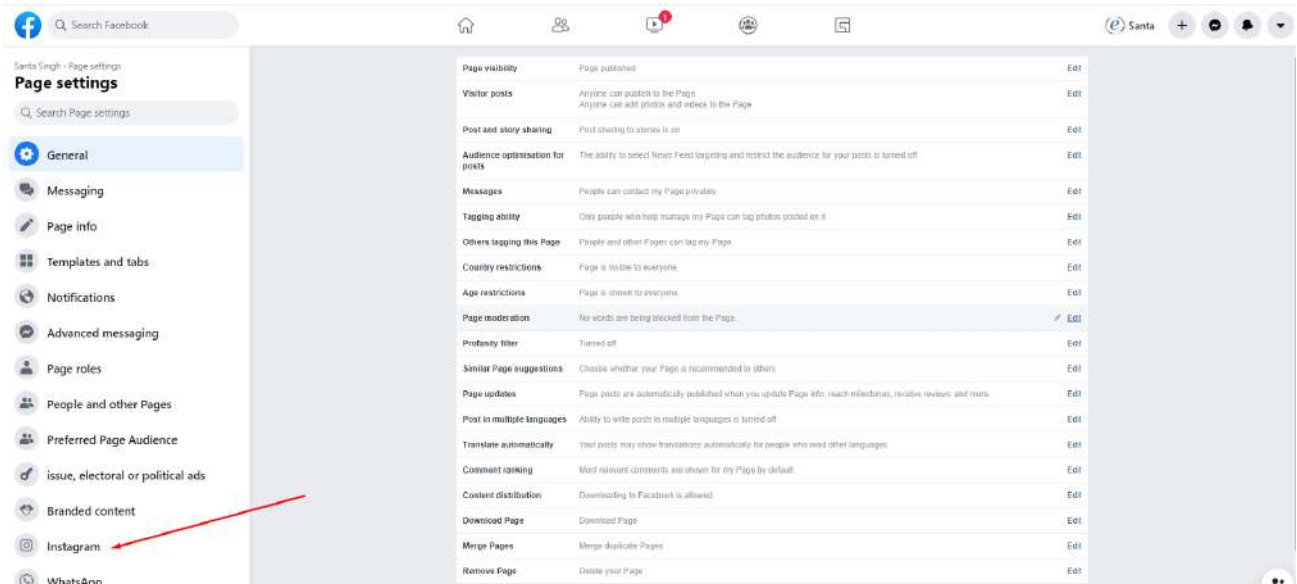
1. First, Select your Facebook page Or Create new page if you had not created yet.



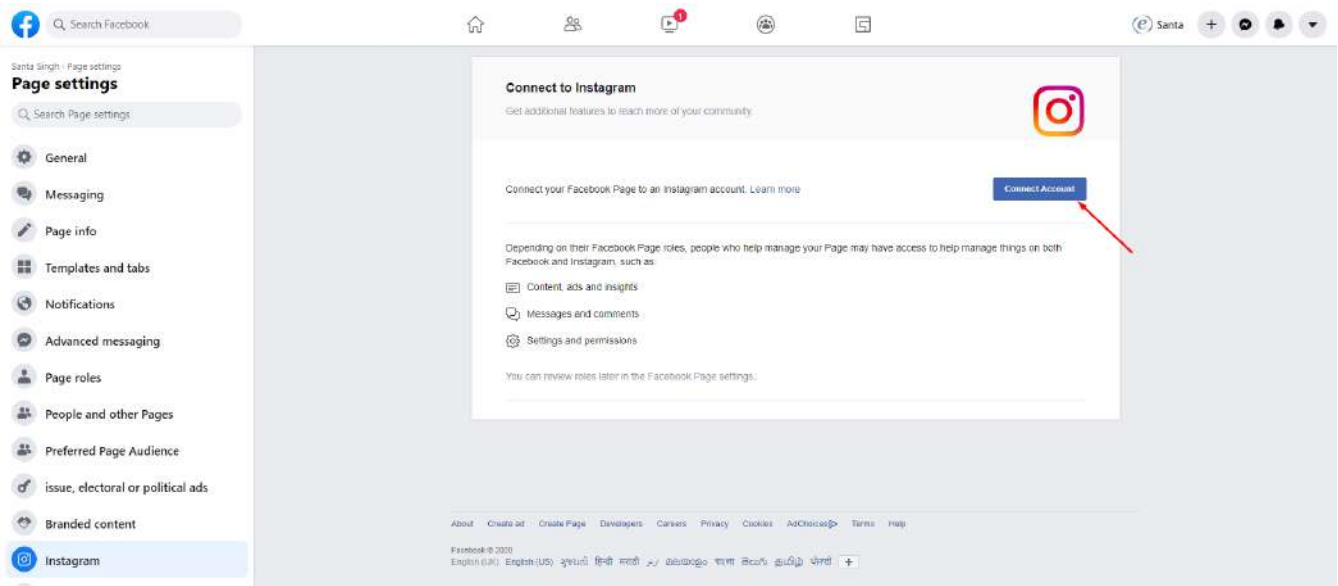
2. Select "Settings" from the menu.



### 3. Select "Instagram" from the menu.



### 4. Now select "Connect Account" to add an Instagram account. Login into your Instagram account. Till now you have not converted your personal account into the business you will be prompted.

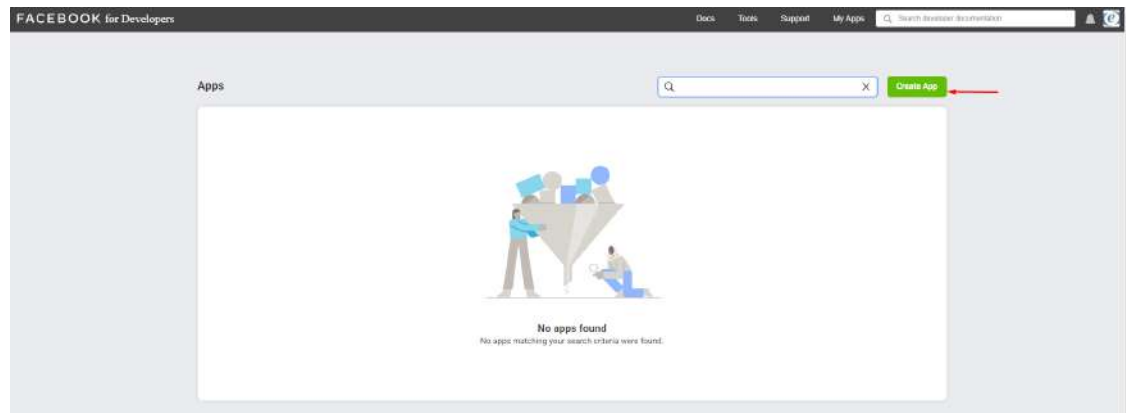


Also, you can remove your Instagram account anytime by selecting the "Disconnect Account".

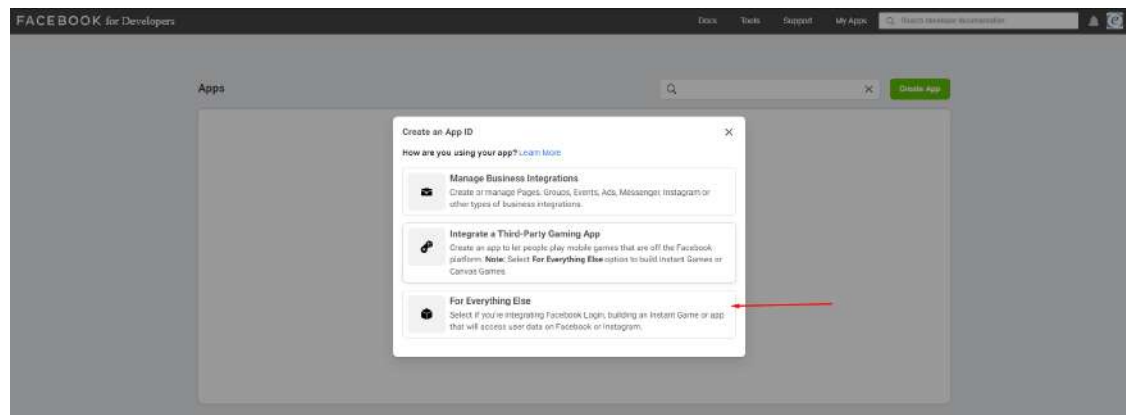
# Authenticate Instagram Facebook Graph API

## Steps to create the App and authenticate Instagram Facebook Graph API.

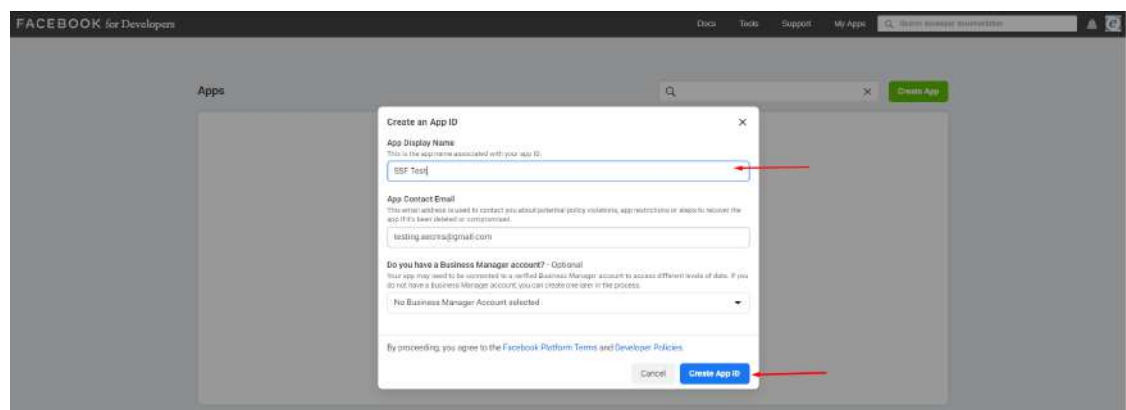
1. Go to [All Apps – Facebook for Developers](#), Click on the “Create App” button.



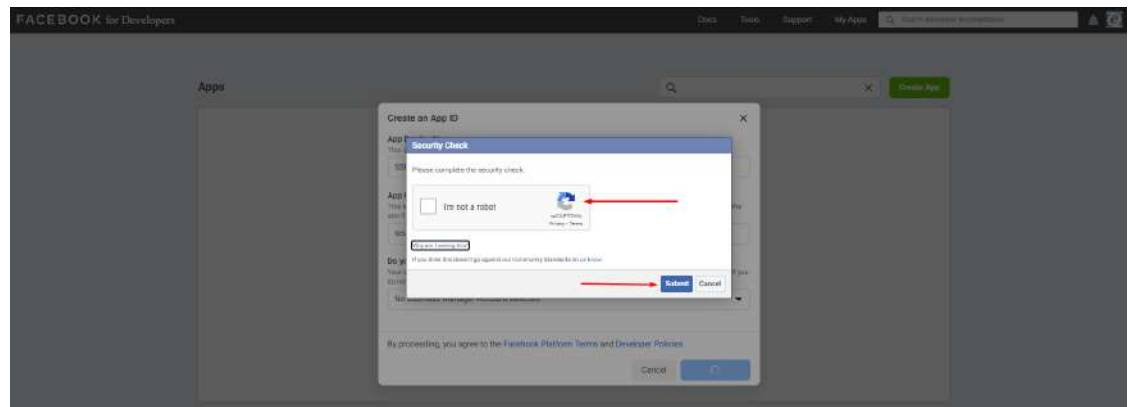
2. Select the “For Everything Else” option.



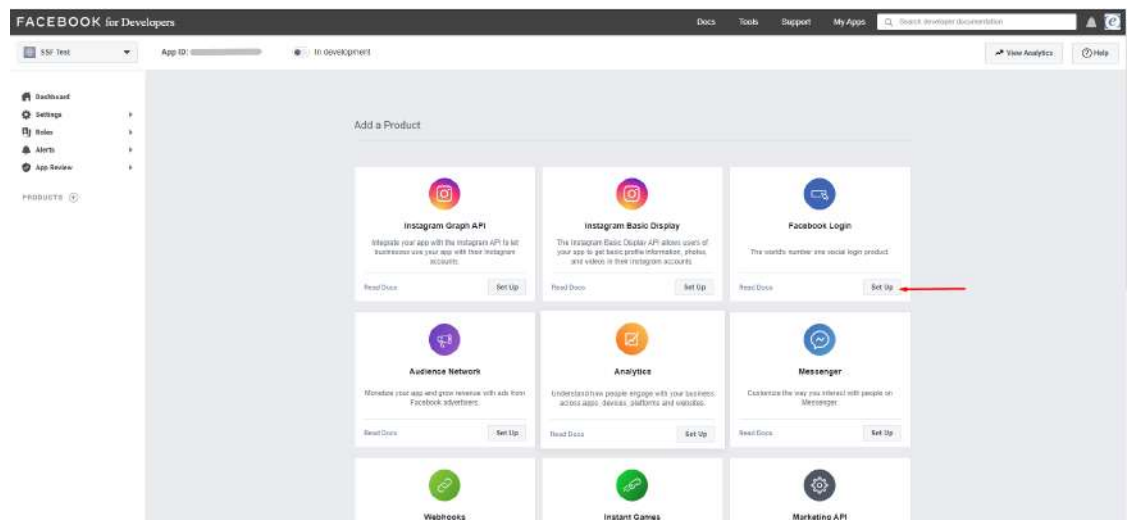
3. Enter the name of the app and click on “Create App ID”.



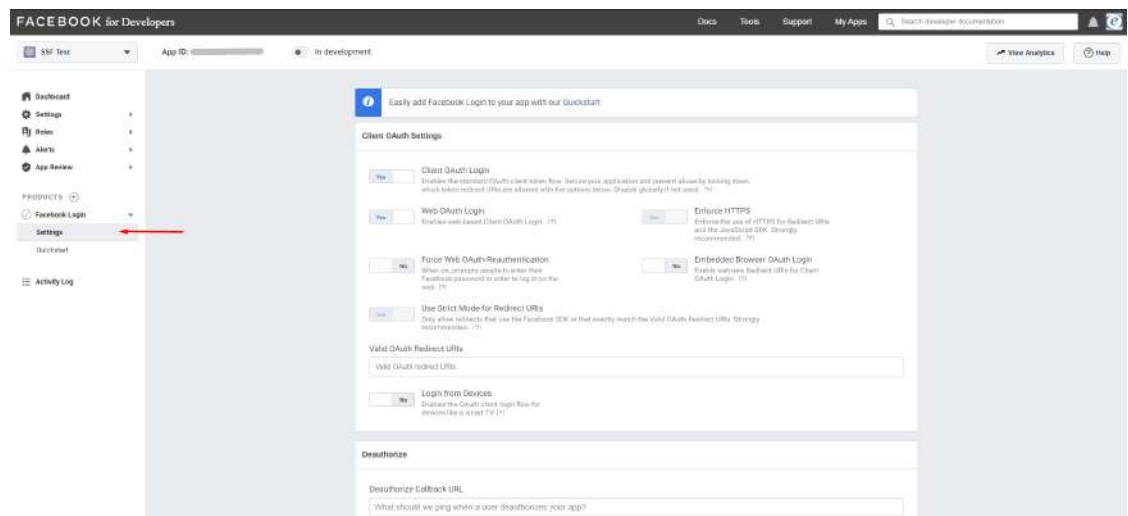
#### 4. Complete the security check and click on the “Submit” button.



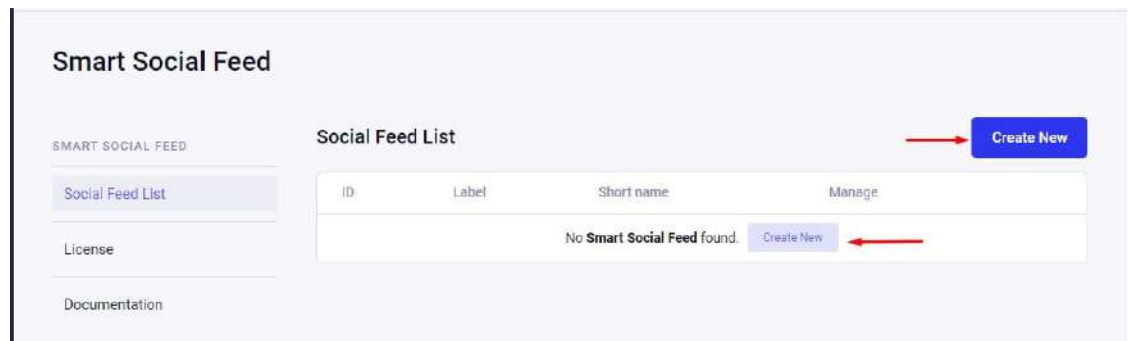
#### 5. Add the “Facebook Login” product so click on the “Set Up” button of that product.



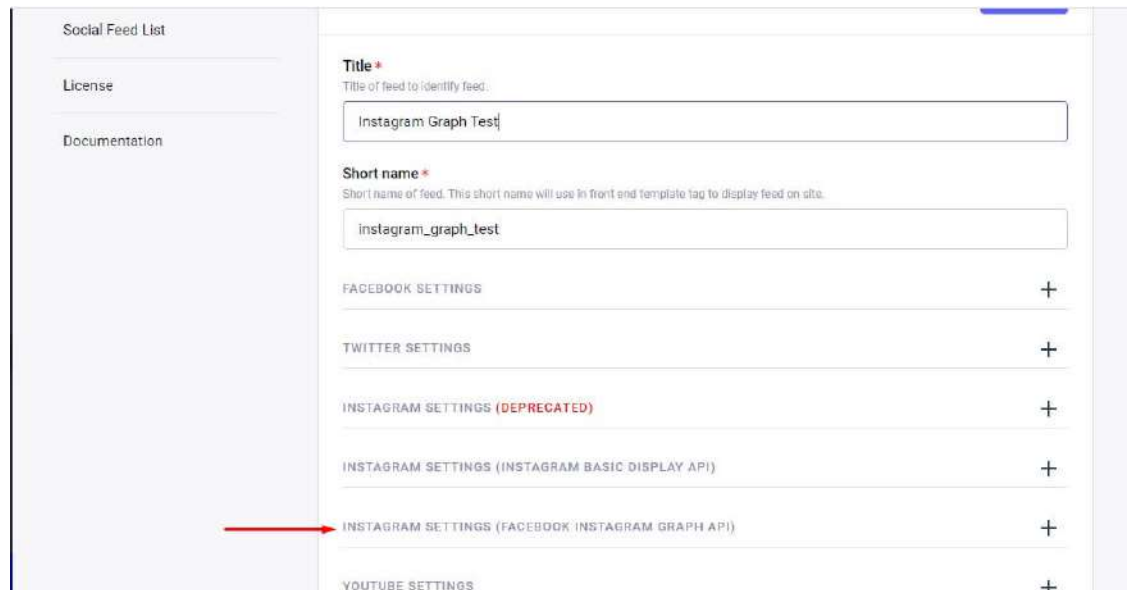
#### 6. Go to the “PRODUCTS > Facebook Login > Settings”.



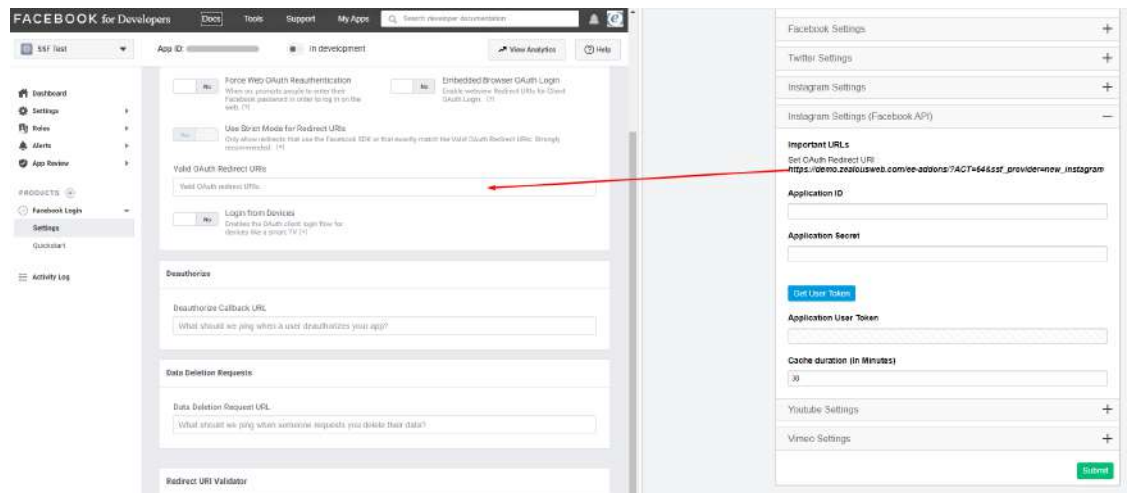
#### 7. Open smart social feed addon in the ExpressionEngine admin. Click on the “Create New” button.



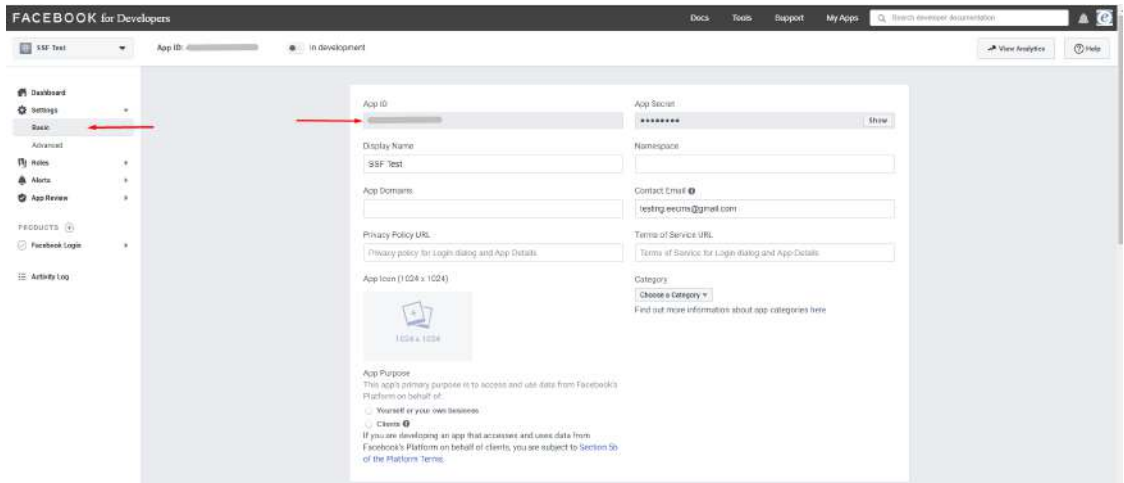
8. Enter the title for the setting and click on the “INSTAGRAM SETTINGS (FACEBOOK INSTAGRAM GRAPH API)” tab.



9. Copy the URL of “Set OAuth Redirect URI” of addon setting and paste to the “Valid OAuth Redirect URIs” textbox of the settings of the “Facebook Login” product.



10. Go to the “Settings > Basic” of the app. Copy “App ID” from the basic page of the app.



## INSTAGRAM SETTINGS (FACEBOOK INSTAGRAM GRAPH API)

### Important URLs

Generate credentials from [Here](#)

More information about this [Here](#)

Set OAuth Redirect URI :

[https://eedemo.zealousweb.com/ee6/index.php?ACT=45&ssf\\_provider=new\\_instagram](https://eedemo.zealousweb.com/ee6/index.php?ACT=45&ssf_provider=new_instagram)

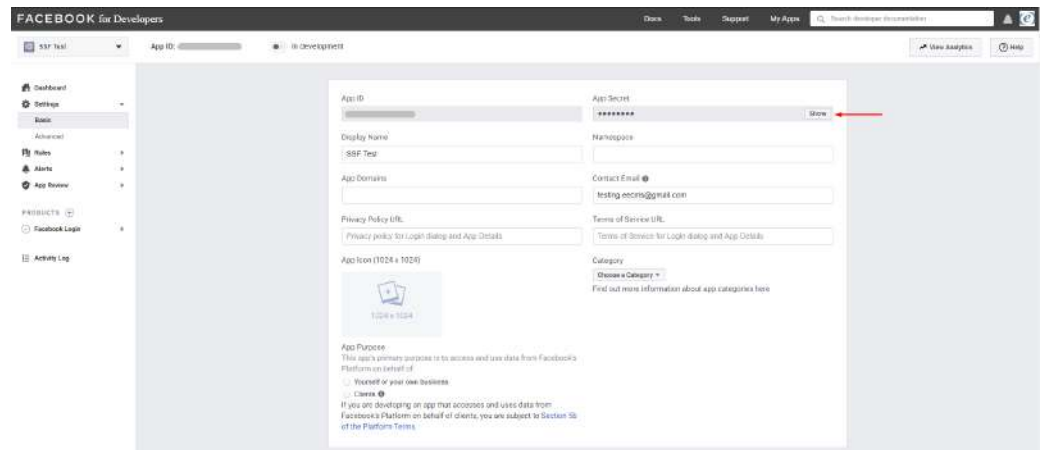
[https://eedemo.zealousweb.com/ee6/index.php?ACT=45&ssf\\_provider=new\\_instagram\\_accounts](https://eedemo.zealousweb.com/ee6/index.php?ACT=45&ssf_provider=new_instagram_accounts)

### Application ID

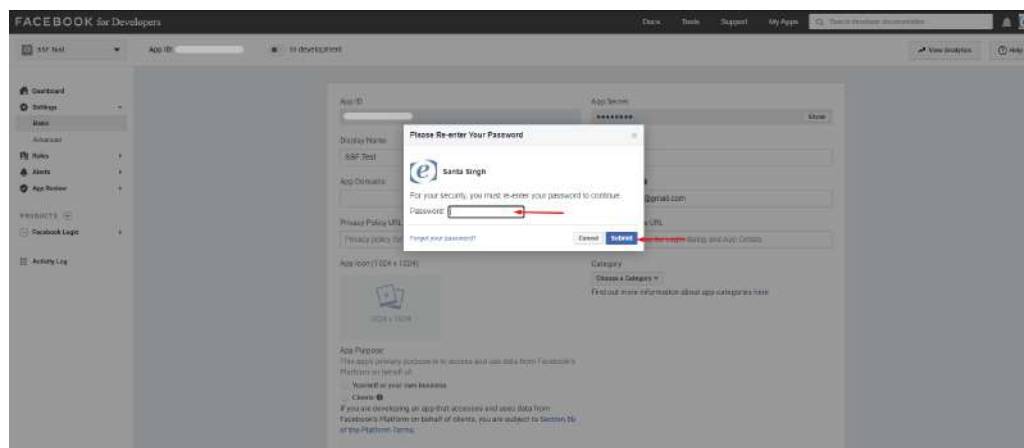
### Application Secret

Get User Token

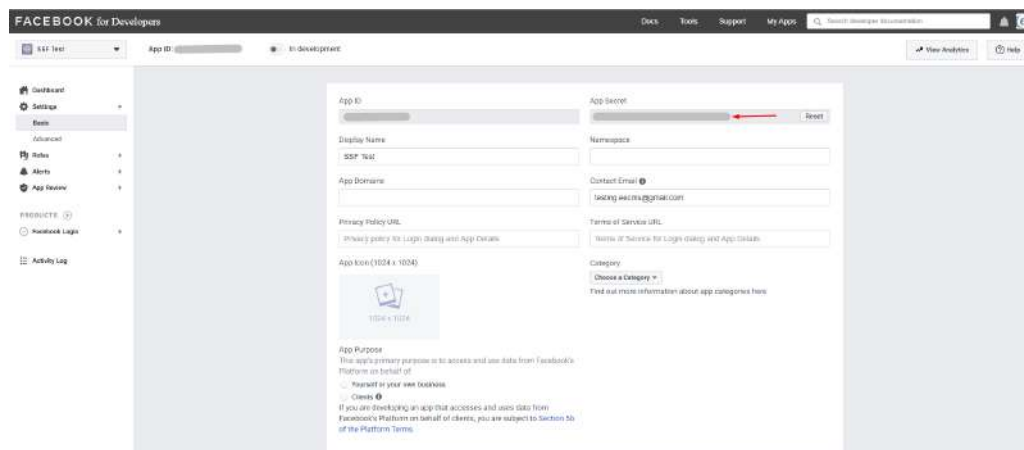
1. Click on the "Show" button of the "App Secret" input field.



2. Enter the password of your Facebook account and click on the “Submit” button.



3. Copy “App Secret” from the basic page of the app.



4. Paste “App Secret” to the “Application Secret” of the addon setting.

INSTAGRAM SETTINGS (FACEBOOK INSTAGRAM GRAPH API)

**Important URLs**

Generate credentials from [Here](#)

More information about this [Here](#)

Set OAuth Redirect URI :

[https://eedemo.zealousweb.com/ee6/index.php/?ACT=45&ssf\\_provider=new\\_instagram](https://eedemo.zealousweb.com/ee6/index.php/?ACT=45&ssf_provider=new_instagram)

[https://eedemo.zealousweb.com/ee6/index.php/?ACT=45&ssf\\_provider=new\\_instagram\\_accounts](https://eedemo.zealousweb.com/ee6/index.php/?ACT=45&ssf_provider=new_instagram_accounts)

**Application ID**

**Application Secret**

**Get User Token**

**Application User Token**

5. Click on the "Get User Token" button of the addon setting. One popup is opened and provides permission in it so the "Application User Token" input field fills up automatically.

**Important URLs**

Generate credentials from [Here](#)

More information about this [Here](#)

Set OAuth Redirect URI :

[https://eedemo.zealousweb.com/ee6/index.php/?ACT=45&ssf\\_provider=new\\_instagram](https://eedemo.zealousweb.com/ee6/index.php/?ACT=45&ssf_provider=new_instagram)

[https://eedemo.zealousweb.com/ee6/index.php/?ACT=45&ssf\\_provider=new\\_instagram\\_accounts](https://eedemo.zealousweb.com/ee6/index.php/?ACT=45&ssf_provider=new_instagram_accounts)

**Application ID**

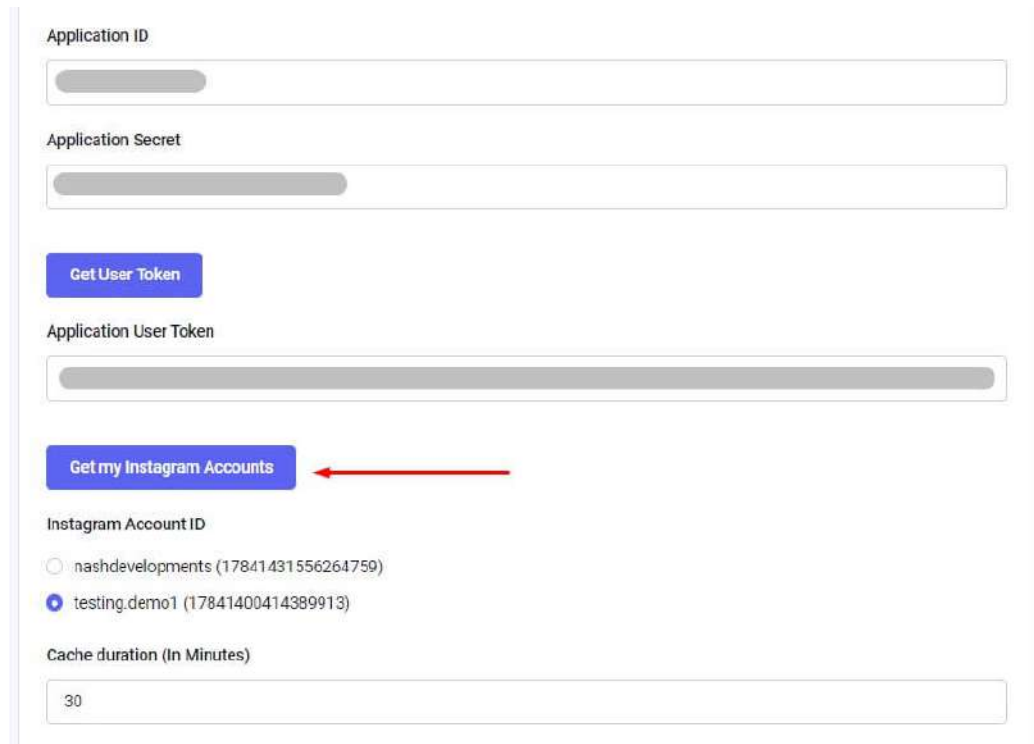
**Application Secret**

**Get User Token**

**Application User Token**

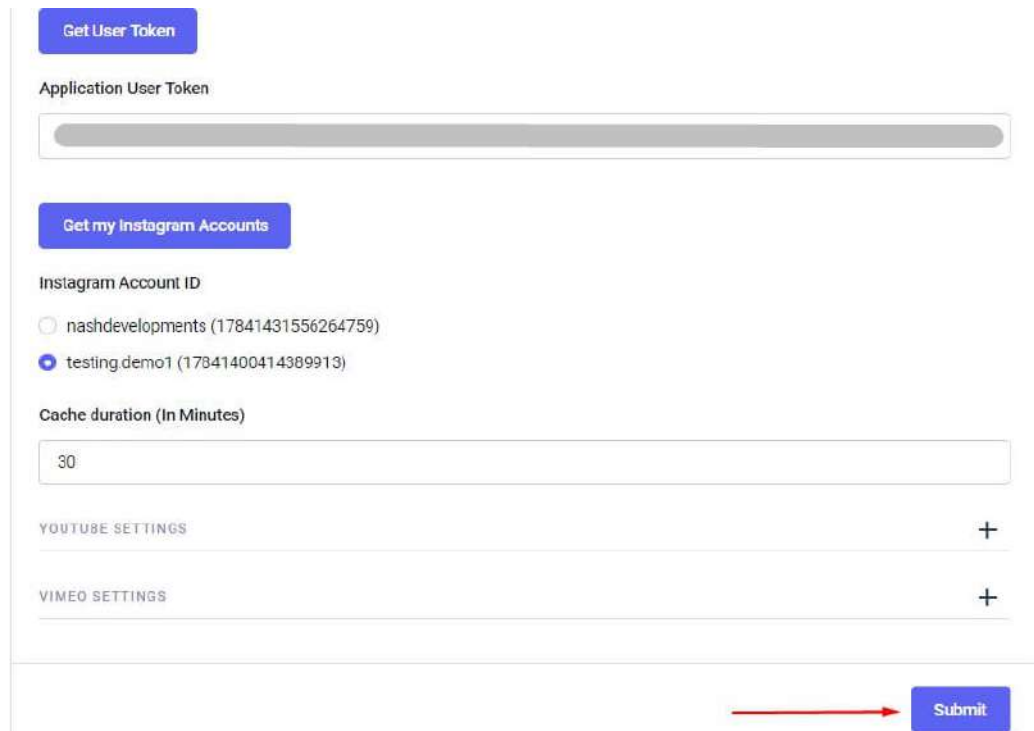
**Get my Instagram Accounts**

6. Click on the “Get my Instagram Accounts” button of the addon setting. Instagram account IDs options are added in the form so you can select any appropriate account ID.



This screenshot shows the 'Get my Instagram Accounts' button highlighted with a red arrow. The form includes fields for 'Application ID', 'Application Secret', 'Application User Token', and 'Cache duration (In Minutes)'. Below the 'Get my Instagram Accounts' button, there are two radio button options for 'Instagram Account ID': 'nashdevelopments (17841431556264759)' and 'testing.demo1 (17841400414389913)'. The 'testing.demo1' option is selected.

7. Click on the “Submit” button of the addon setting.



This screenshot shows the 'Submit' button highlighted with a red arrow. The form includes fields for 'Application User Token', 'Cache duration (In Minutes)', and 'YOUTUBE SETTINGS'. Below the 'Submit' button, there are two radio button options for 'Instagram Account ID': 'nashdevelopments (17841431556264759)' and 'testing.demo1 (17841400414389913)'. The 'testing.demo1' option is selected.

## Instruction

1. Your app must be in the “In development” mode.
2. Your site must have SSL.

# Authenticate Instagram Basic Display API

The Instagram Basic Display API allows your app users to get photos and videos in their Instagram public type of accounts.

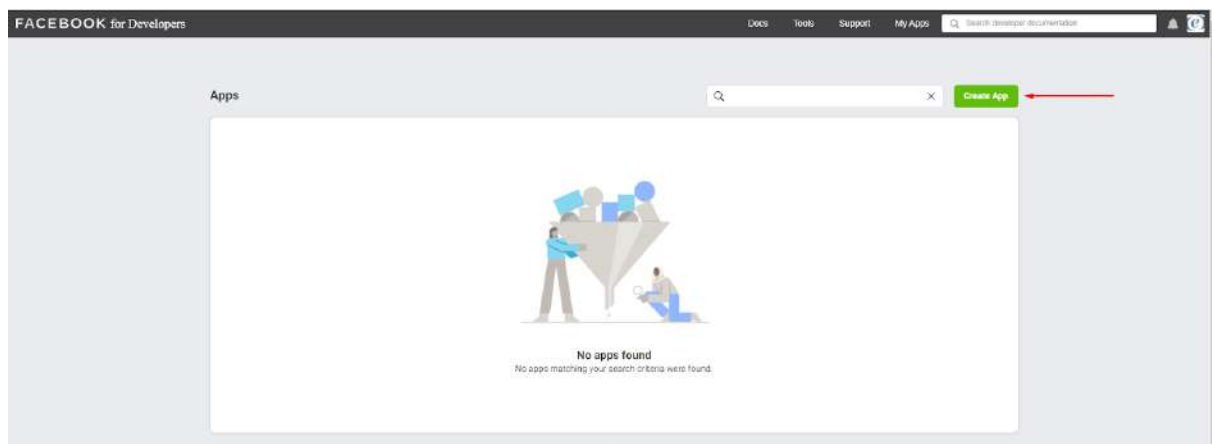
## Before You Start

You will need:

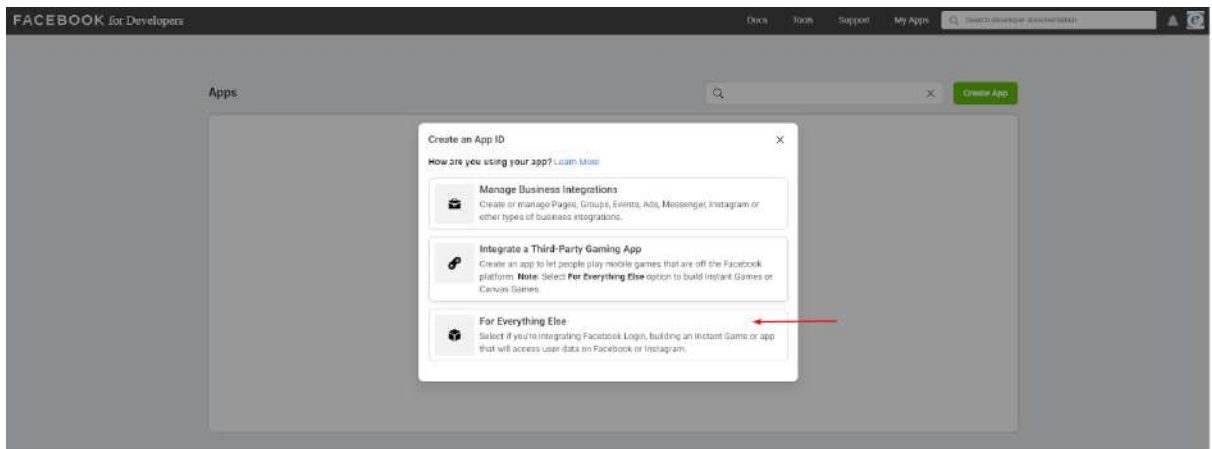
1. [A Facebook Developer Account](#)
2. [An Instagram account](#). (Type: Public)
3. A public website that you own. (HTTPS on websites are required if you want to authorize Instagram accounts used by authorized windows on your website.)

## Step to authorize API

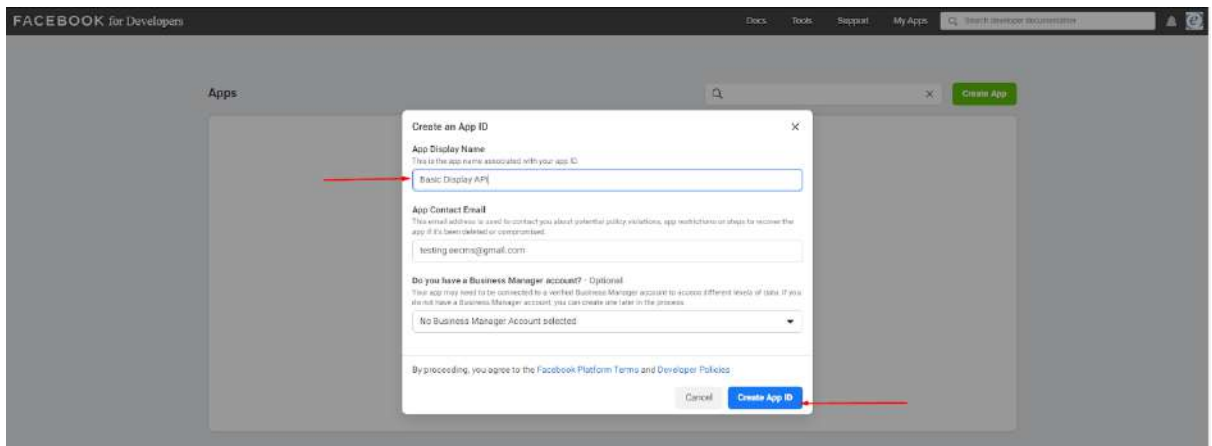
1. Go to [Facebook Developer Account](#), Click on **My Apps** button, and create a new app.



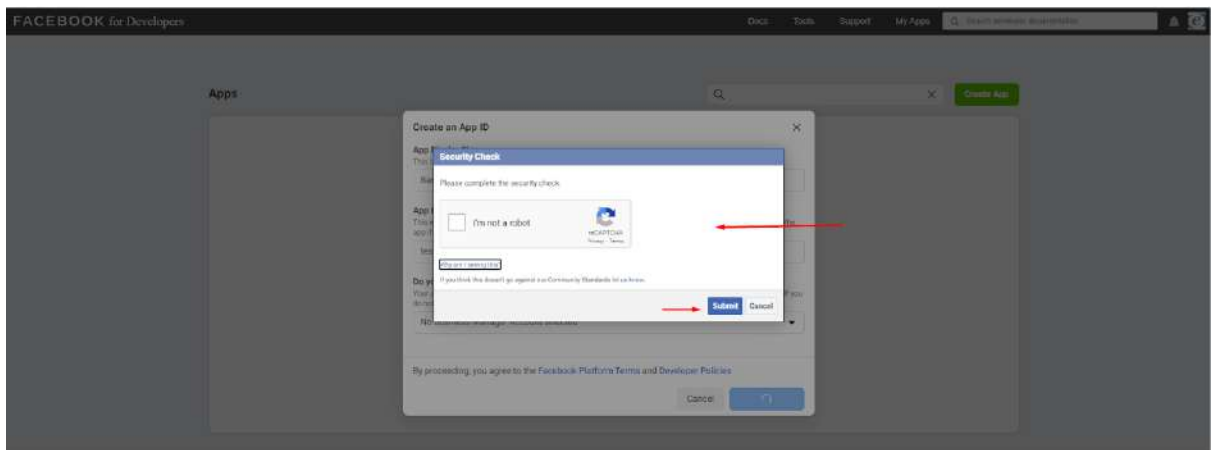
2. Select the **For Everything Else** option from the popup.



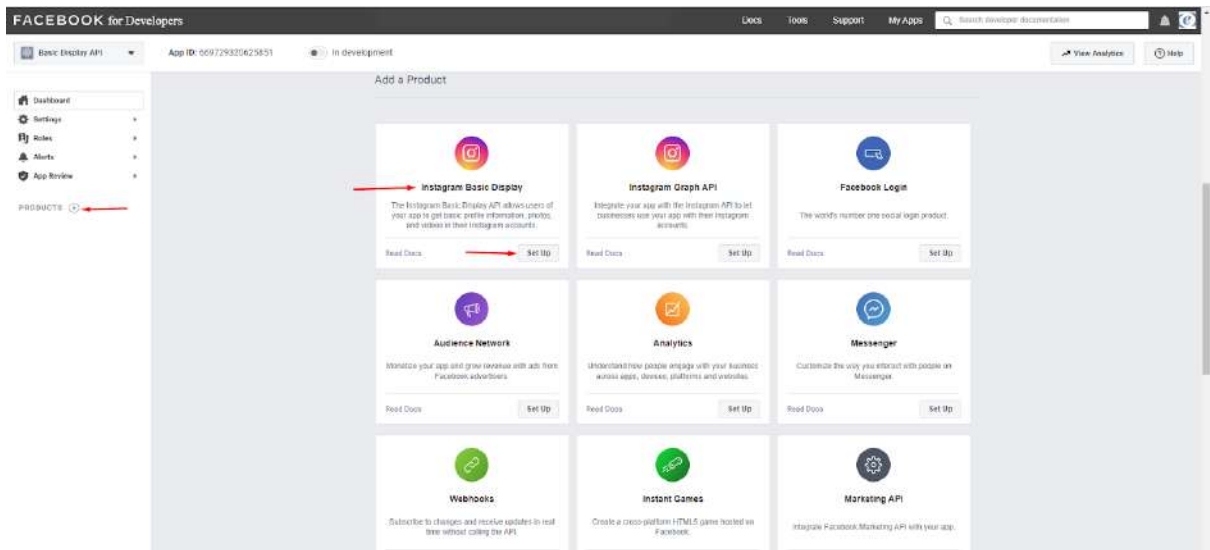
3. Enter the name in the **App Display Name** field that you want. Click on the **Create App ID** button.



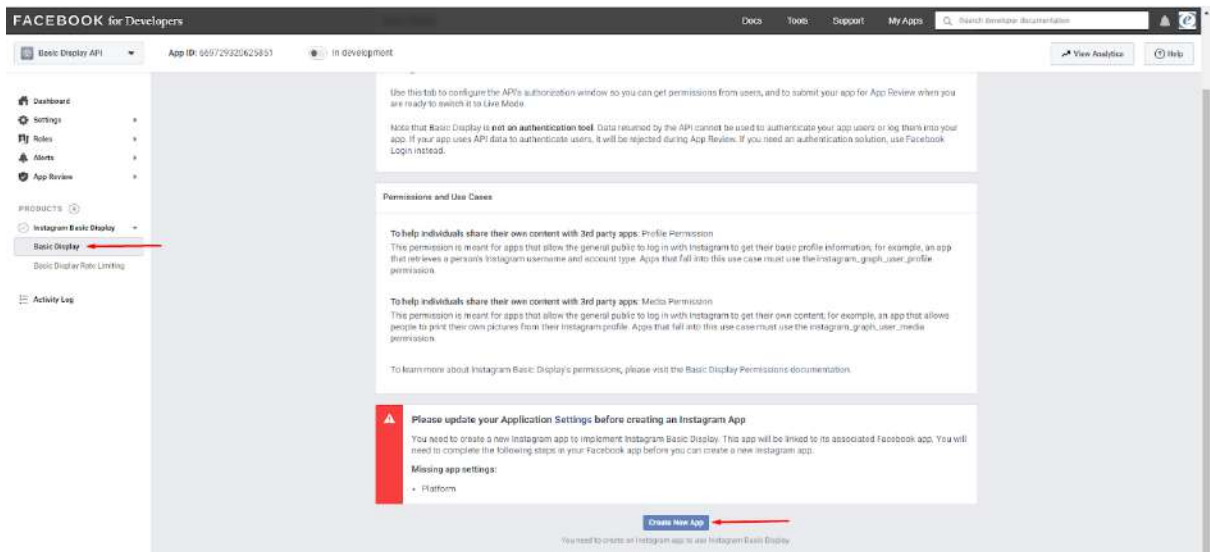
4. Complete the security check of google recaptcha and submit the form.



5. Finally, the app is created. Add the **Instagram Basic** product in the app created. To do so, click on the **Set Up** button of this product.

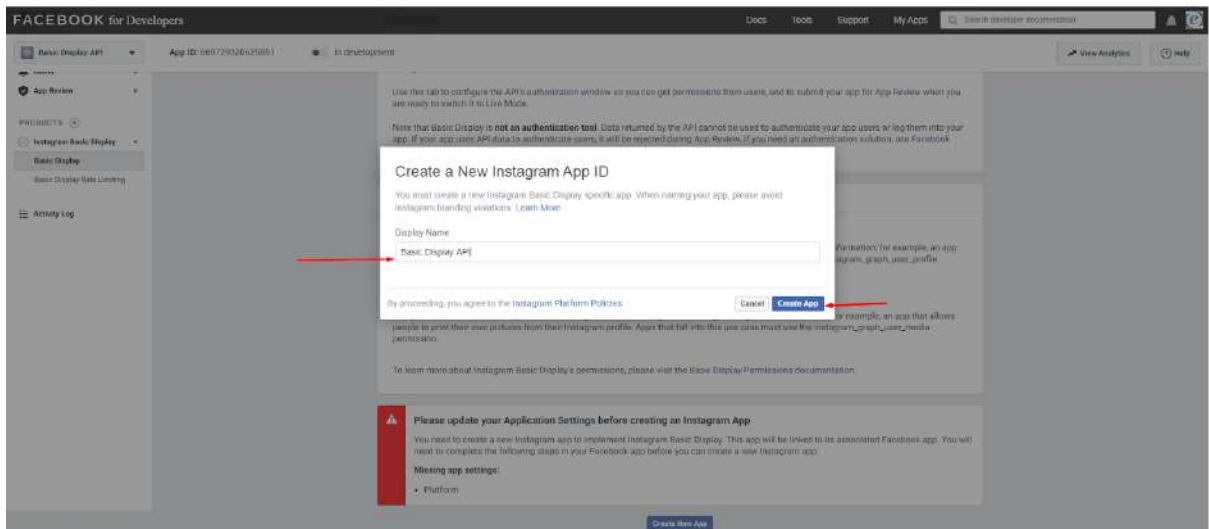


6. Click on the **Create New App** button of the **Basic Display** page of the added **Instagram Basic Display** product.

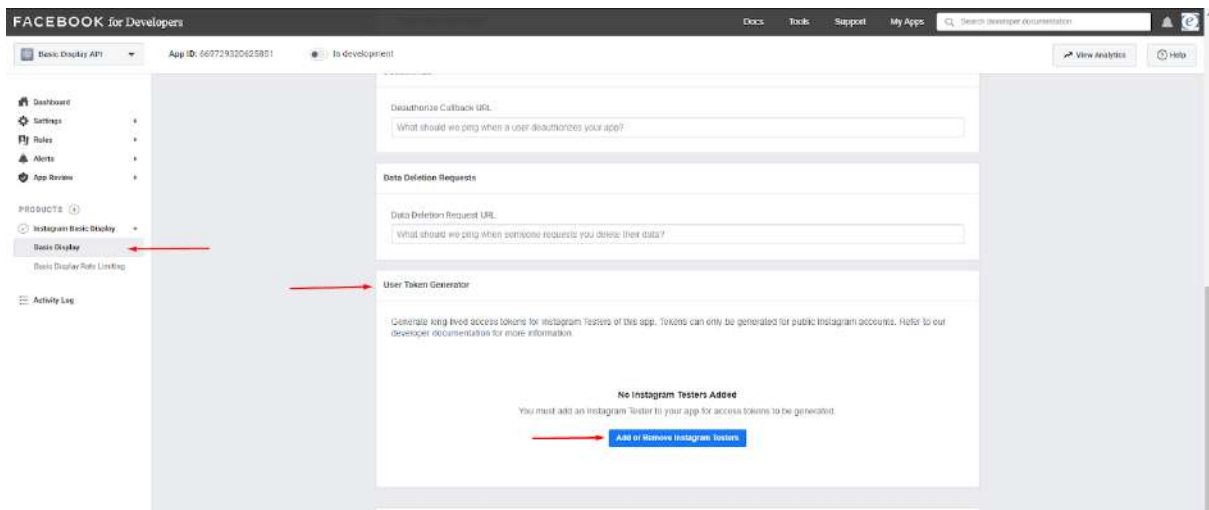


7. You can change the name in the **Display Name** field in the form of **Create a New Instagram App ID** if you want to change, otherwise you can keep

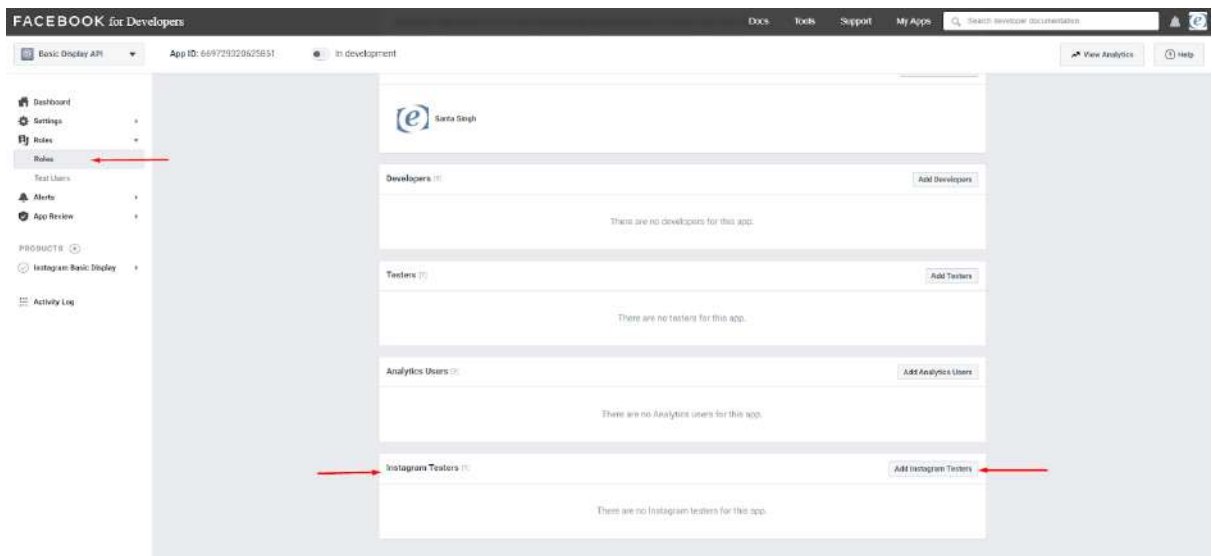
the same name. Click on the **Create App** button.



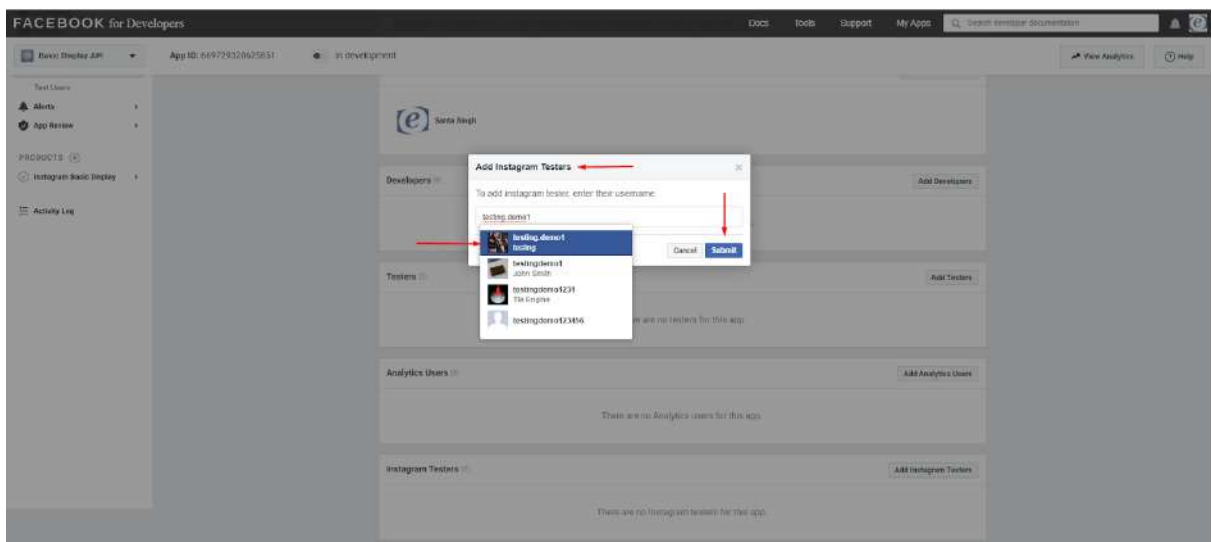
8. We must add an Instagram Tester to the app created to access tokens to be generated. For this, click on the **Add or Remove Instagram Testers** button of the **User Token Generator** tab of the **Basic Display** page.



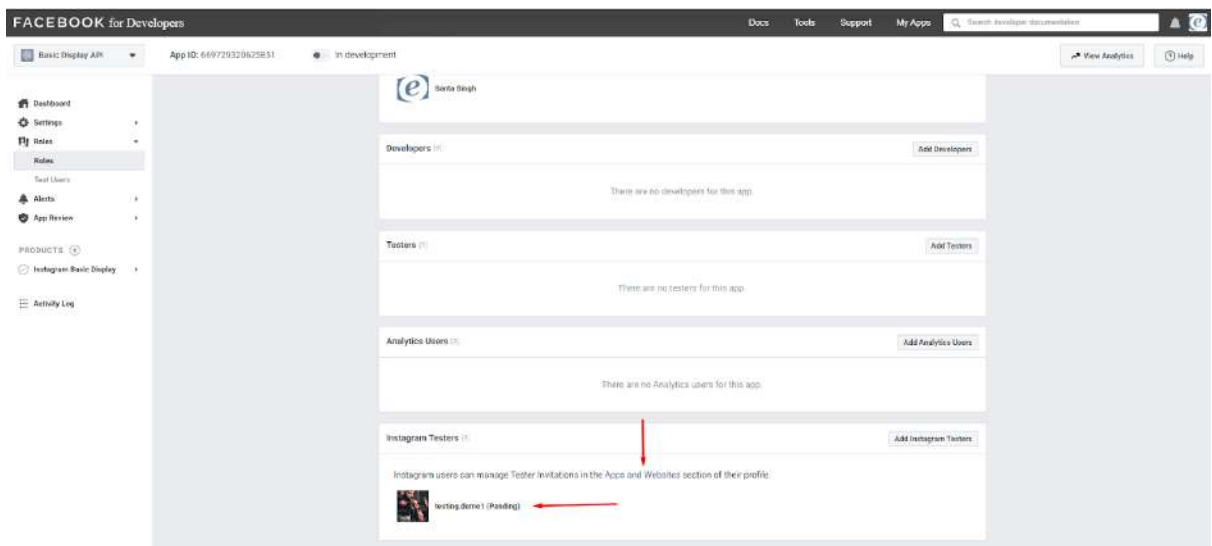
9. The **Roles** page is now open. Click on the **Add Instagram Testers** button of the **Instagram Testers** tab of this page.



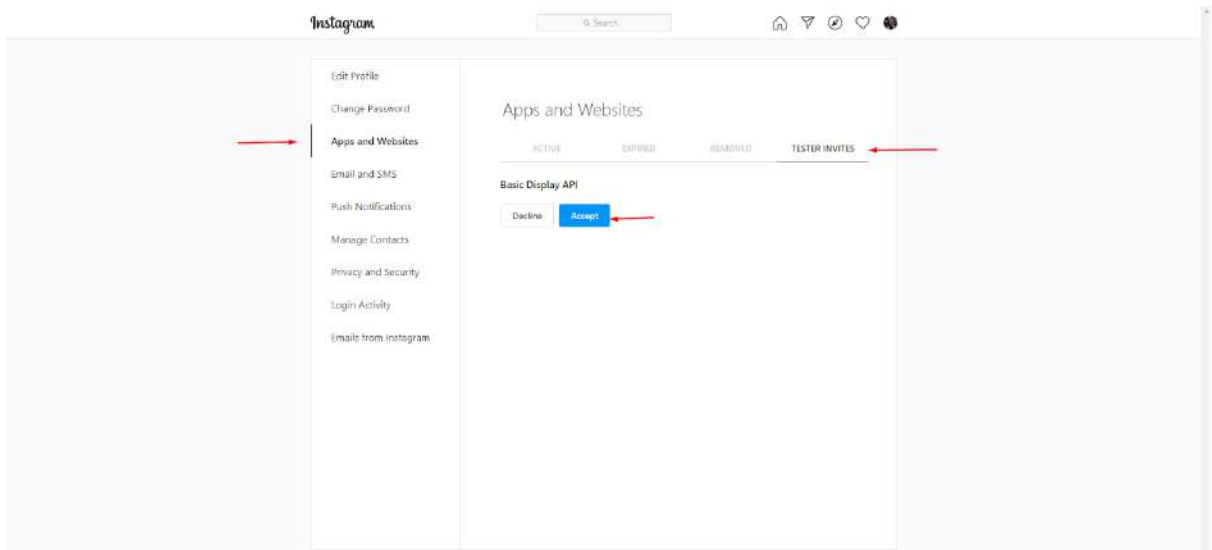
10. Enter the username of the Instagram account you want to add in the form of the **Add Instagram Testers** form and select desired account and click on the Submit button.



11. Instagram tester is added but it will show as pending. The Instagram tester must accept the invitation in the **App and Websites** section of their profile.



12. Click on the **Accept** button of the specific invited app in the **TESTER INVITES** tab of the **Settings > Apps and Websites** of the invited Instagram account. Now Instagram account is successfully added as tester and activated.



13. Go to the **Smart Social Feed** addon in EE admin and create the new setting and click on the **Instagram Settings (Instagram Basic Display API)** tab.

SMART SOCIAL FEED

Social Feed List

License

Documentation

**Create new social feed**

**Title \***  
Title of feed to identify feed.

**Short name \***  
Short name of feed. This short name will use in front end template tag to display feed on site.

FACEBOOK SETTINGS +

TWITTER SETTINGS +

INSTAGRAM SETTINGS (DEPRECATED) +

INSTAGRAM SETTINGS (INSTAGRAM BASIC DISPLAY API) +

INSTAGRAM SETTINGS (FACEBOOK INSTAGRAM GRAPH API) +

14. This is the important step to create the token.

1. We can create the token using the **Authorised Window** using the EE admin panel.
2. We can create the token using the **User Token Generator** of the app.

15. First, we will see how to create a token using the **Authorised Window** method. Copy the **Instagram App ID** and **Instagram App Secret** from the **Basic Display** page.

FACEBOOK for Developers

Basic Display API

App ID: [redacted]

In development

View Analytics

Help

Dashboard

Settings

Roles

Alerts

App Review

PRODUCTS

Instagram Basic Display

Basic Display

Basic Display Rate Limiting

Activity Log

Instagram Basic Display allows apps to access the Instagram Basic Display API, which provides read-only access to basic data in app users' Instagram accounts.

Use this tab to configure the API's authorization window so you can get permissions from users, and to submit your app for App Review when you are ready to switch it to Live Mode.

Note that Basic Display is **not an authentication tool**. Data returned by the API cannot be used to authenticate your app users or log them into your app. If your app uses API data to authenticate users, it will be rejected during App Review. If you need an authentication solution, use Facebook Login instead.

**Permissions and Use Cases**

**To help individuals share their own content with 3rd party apps: Profile Permission**  
This permission is meant for apps that allow the general public to log in with Instagram to get their basic profile information, for example, an app that removes a person's Instagram username and account type. Apps that fall into this use case must use the `instagram_graph_user_profile` permission.

**To help individuals share their own content with 3rd party apps: Media Permission**  
This permission is meant for apps that allow the general public to log in with Instagram to get their own content, for example, an app that allows people to print their own pictures from their Instagram profile. Apps that fall into this use case must use the `instagram_graph_user_media` permission.

To learn more about Instagram Basic Display's permissions, please visit the [Basic Display Permissions documentation](#).

Instagram App ID: [redacted]

Instagram App Secret: [redacted]

Instagram Display Name: [redacted]

Basic Display API

16. Paste the **Instagram App ID** to **Application ID (Instagram App ID)** and **Instagram App Secret** to **Application Secret (Instagram App Secret)** of

the addon setting.

INSTAGRAM SETTINGS (INSTAGRAM BASIC DISPLAY API)

**Important URLs**  
 Generate credentials from [Here](#)  
 More information about this [Here](#)  
 Set OAuth Redirect URI :  
<https://eedemo.zealousweb.com/ee6/index.php/insta-basic-auth>

**Application ID (Instagram App ID)**

**Application Secret (Instagram App Secret)**

**Get User Token**

**Application User Token**

**Cache duration (In Minutes)**

17. Copy the **Set OAuth Redirect URI** from the addon setting and paste to the **Valid OAuth Redirect URIs** field of the Client OAuth Settings tab of the **Basic Display** page.

**Note:** Your website URL must with https.

The screenshot shows the 'Basic Display API' settings page. On the left, the 'Instagram Settings (Instagram Basic Display API)' tab is expanded, showing the 'Set OAuth Redirect URI' field with the value 'https://eedemo.zealousweb.com/ee6/index.php/insta-basic-auth'. A red arrow points from this field to the 'Valid OAuth Redirect URIs' field in the 'Client OAuth Settings' tab on the right. The 'Valid OAuth Redirect URIs' field is currently empty.

18. We should add a website URL or **Set OAuth Redirect URI** to **Deauthorize Callback URL** field of the **Deauthorize** and **Data Deletion Request URL** field of the **Data Deletion Requests** tab of the **Basic Display** page. We need to add these two URLs in the app because it is a required field for this form.

Click on the **Submit Changes** button to save the details.

**Note:** Our addon takes any action on both URL (Deauthorize Callback URL, Data Deletion Request URL).

Facebook for Developers

Basic Display API | App ID: 669725320625851 | in development

Dashboard | Settings | Roles | Alerts | App Review

PRODUCTS

- Instagram Basic Display
- Basic Display
- Basic Display Rate Limiting

Activity Log

Instagram Display Name

basic Display API

Client OAuth Settings

Valid OAuth Redirect URIs

<https://eedemo.zealousweb.com/ee6/index.php/insta-basic-auth>

Deauthorize

Deauthorize Callback URL

<https://eedemo.zealousweb.com/ee6/index.php/insta-basic-auth>

Data Deletion Requests

Data Deletion Request URL

<https://eedemo.zealousweb.com/ee6/index.php/insta-basic-auth>

User Token Generator

Generate long-lived access tokens for Instagram Testers of this app. Tokens can only be generated for public Instagram accounts. Refer to our developer documentation for more information.

Instagram Test Users

Tokens

Discard Save Changes

19. Click on the **Get User Token** in the addon setting.

INSTAGRAM SETTINGS (INSTAGRAM BASIC DISPLAY API)

Important URLs

Generate credentials from [Here](#)

More information about this [Here](#)

Set OAuth Redirect URI :

**<https://eedemo.zealousweb.com/ee6/index.php/insta-basic-auth>**

Application ID (Instagram App ID)

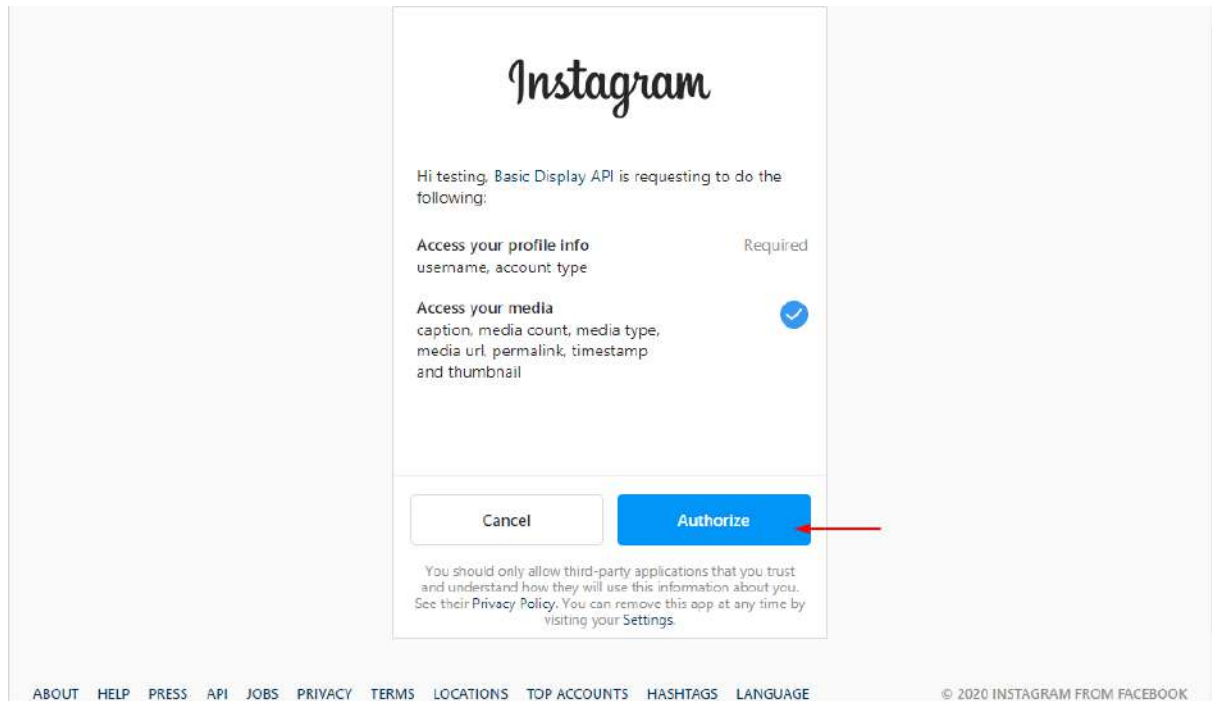
Application Secret (Instagram App Secret)

Get User Token

Application User Token

Cache duration (In Minutes)

20. One Authorization Window will open to ask permission to access the media of the Instagram account. Click on the **Authorize** button.



21. If everything is perfect then a token is created and is auto placed to the **Application User Token** field of the addon setting.

INSTAGRAM SETTINGS (INSTAGRAM BASIC DISPLAY API)

### Important URLs

Generate credentials from [Here](#)

More information about this [Here](#)

Set OAuth Redirect URI :

**https://eedemo.zealousweb.com/ee6/index.php/insta-basic-auth**

### Application ID (Instagram App ID)

### Application Secret (Instagram App Secret)

### Application User Token

### Cache duration (In Minutes)

1. If you don't want to create the token using the Authorization Window using your site then you can use the second method to create the token using the **User Token Generator** of the app. For this method, You need to click on the **Generate Token** button of the **User Token Generator** tab

FACEBOOK for Developers
Docs
Tools
Support
My Apps
Search developer documentation

Basic Display API
App ID:
In development
View Analytics
Help

- Dashboard
- Settings
- Rules
- Alerts
- App Review

PRODUCTS

- Instagram Basic Display
- Basic Display
- Basic Display Rate Limiting

- Activity Log

### Data Deletion Requests

Data Deletion Request URL:

https://eedemo.zealousweb.com/ee6/index.php/insta-basic-auth

### User Token Generator

Generate long-lived access tokens for Instagram Testers of this app. Tokens can only be generated for public Instagram accounts. Refer to our developer documentation for more information.

Instagram Test Users	Tokens
testing demo1	--

[Add or Remove Instagram Testers](#)

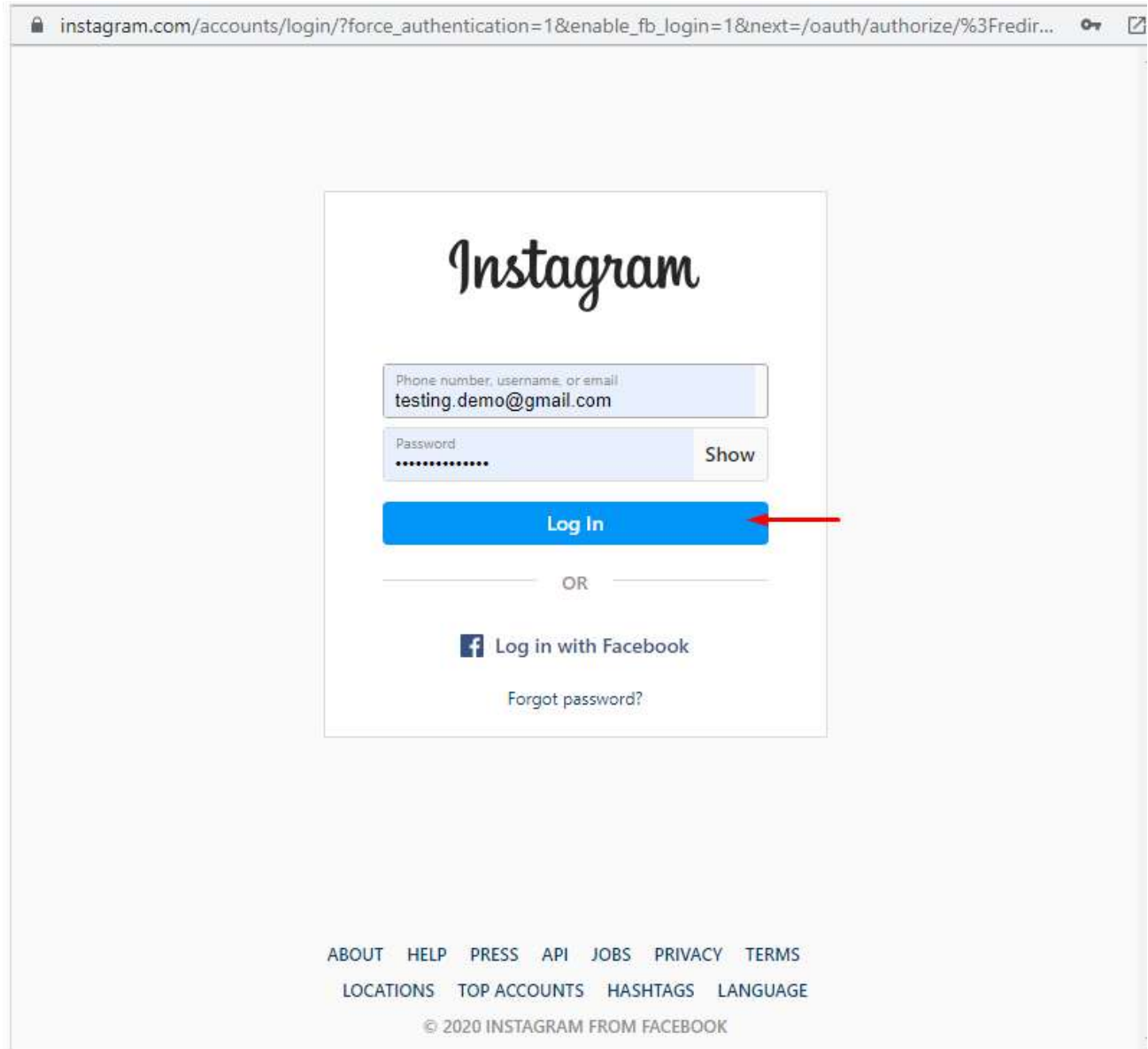
### App Review for Instagram Basic Display

To use Instagram platform, your app needs to be approved for `instagram_graph_user_profile` and `instagram_graph_user_media` permission.

☐ `instagram_graph_user_profile`  
Grants the ability to read an Instagram user's profile info.

☐ `instagram_graph_user_media`  
Grants the ability to read an Instagram user's media.

2. In this method, a popup window will open and ask for a login to the account in instagram.



instagram.com/accounts/login/?force\_authentication=1&enable\_fb\_login=1&next=/oauth/authorize/%3Fredir...

Instagram

Phone number, username, or email  
testing.demo@gmail.com

Password  
..... Show

Log In

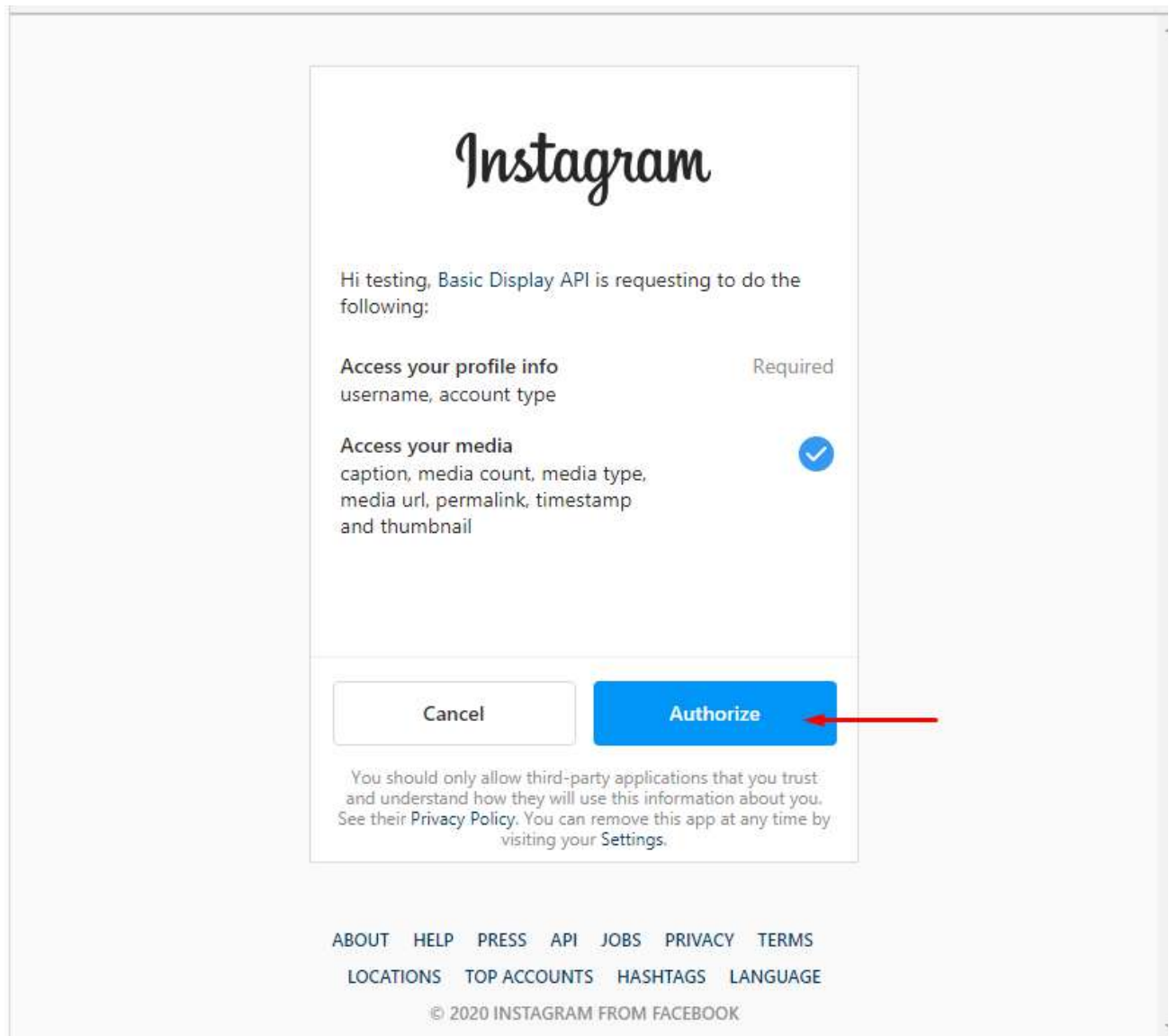
OR

Log in with Facebook

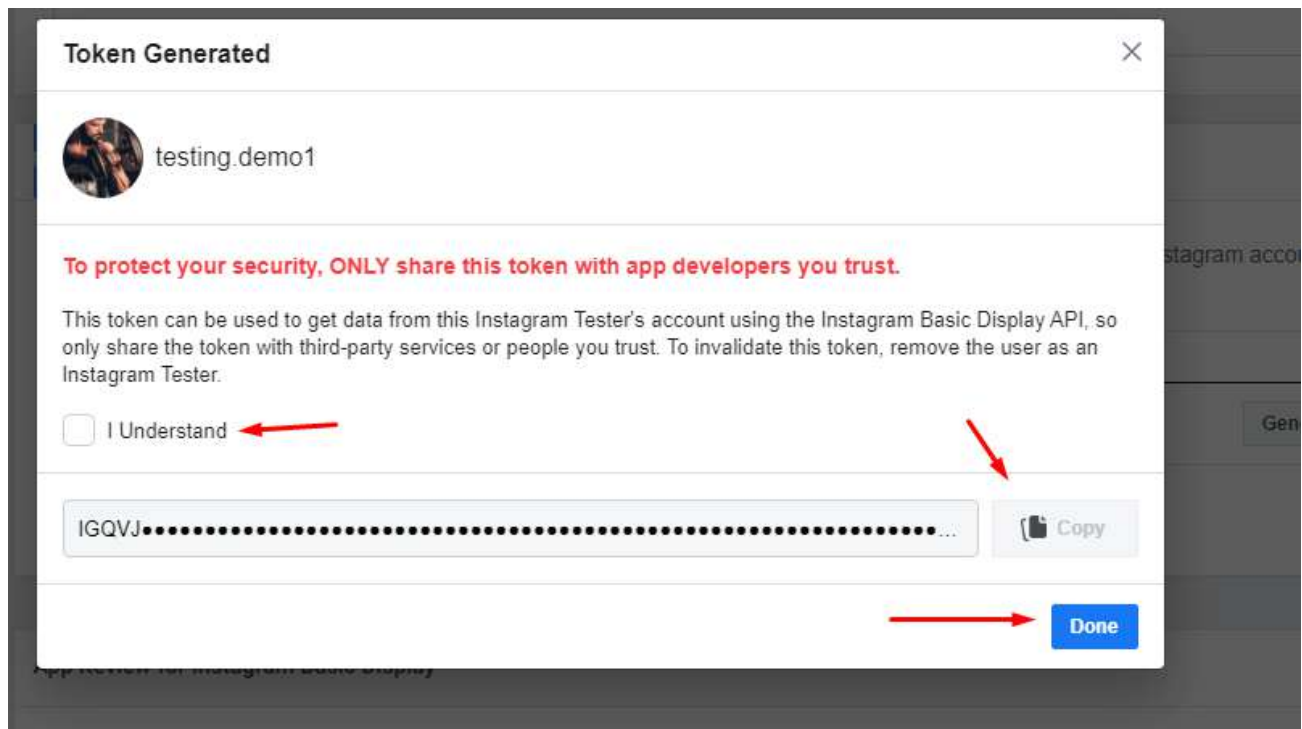
Forgot password?

ABOUT HELP PRESS API JOBS PRIVACY TERMS  
LOCATIONS TOP ACCOUNTS HASHTAGS LANGUAGE  
© 2020 INSTAGRAM FROM FACEBOOK

3. Click on the Authorize button to give the access to the media of the Instagram account.



4. Your token is generated. Check the **I Understand** checkbox field, copy the token, and click on the **Done** button.



5. Then, you can paste the token to the **Application User Token** field of the addon setting. In this method, you have to add **Application ID (Instagram App ID)** and **Application Secret (Instagram App Secret)** in the addon setting page. The advantage of this method is that we don't add **Set OAuth Redirect URI** to the app setting.
6. Now finally, we have got the token. We then provide the name of the addon setting and save this setting.

**Title \***  
Title of feed to identify feed:  
dev-insta-basic

**Short name \***  
Short name of feed. This short name will use in front end template tag to display feed on site:  
dev\_insta\_basic

INSTAGRAM SETTINGS (INSTAGRAM BASIC DISPLAY API)

**Important URLs**  
Generate credentials from [Here](#)  
More information about this [Here](#)  
Set OAuth Redirect URI :  
<https://eedemo.zealousweb.com/ee6/index.php/insta-basic-auth>

Application ID (Instagram App ID)  
[Redacted]

Application Secret (Instagram App Secret)  
[Redacted]

[Get User Token](#)

Application User Token  
[Redacted]

Cache duration (In Minutes)  
30

[Submit](#)

7. Your setting is displayed on the **Social Feed List** page. Now you can click the sync Icon of your setting to get the feed.

**Smart Social Feed**

Smart Social Feed

☒ Social Feed List

[Documentation](#)

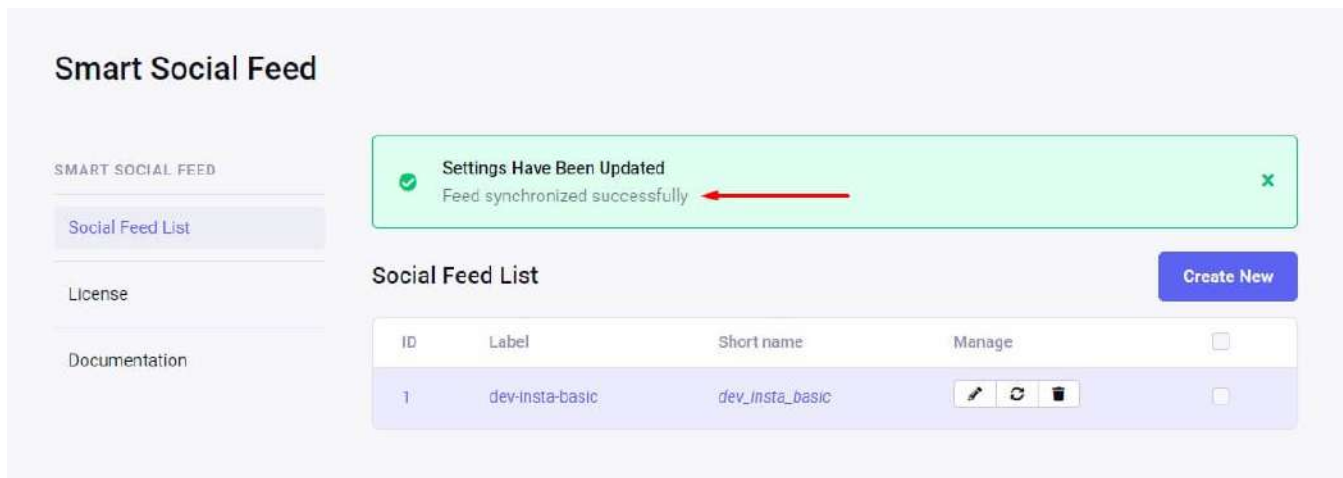
Add-On Manager / Smart Social Feed / Social Feed List

**Settings Have Been Updated**  
Settings successfully updated.

[Create New](#)

ID	Label	Short Name	Manage
1	Instagram Basic Display	instagram_basic_display	<a href="#">Edit</a> <a href="#">Sync</a> <a href="#">Delete</a>

8. If setting details are valid then you get a message saying, **Feed synchronized** successfully, which means that means you have got the feed successfully from the account.



# Installation

## Follow below steps to install Smart Social Feed version: EE4, EE5, EE6:

1. Download and Extract ZIP file of the module. (We recommend to use the latest version.)
2. Copy Smart\_Social\_Feed > system > user > addons > smart\_social\_feed folder to "Your Site > system > user > addons" Folder.
3. Copy Smart\_Social\_Feed > themes > user > smart\_social\_feed folder to "Your Site > themes > user" Folder.
4. Login to backend panel with Super admin credentials.
5. Go to "Developer > Add-Ons" and Install Smart Social Feed module.

**Note:** If you purchased the addon from Devot:ee then you can get the license key from the [licenses tab in your profile](#).

## Social Feed

Social Feed List allows to create or edit Social media feeds as per your requirements.

You can create a new feed just by clicking on CREATE NEW button.

Also, you can edit and manage the previously created feeds with ease.

Create feeds for social platforms like Facebook, Twitter, Instagram, YouTube, Vimeo with ease. Once created it will be displayed in the row of the table.

## Social Feed List (Default view)

**Smart Social Feed**

SMART SOCIAL FEED

- Social Feed List
- License
- Documentation

**Social Feed List** Create New

ID	Label	Short name	Manage	
1	Facebook	facebook		<input type="checkbox"/>
2	Twitter	twitter		<input type="checkbox"/>
3	Instagram Basic Display	instagram_basic_display		<input type="checkbox"/>
4	Facebook Instagram Graph	facebook_instagram_graph		<input type="checkbox"/>
5	Youtube	youtube		<input type="checkbox"/>
6	Vimeo	vimeo		<input type="checkbox"/>

**Smart Social Feed**

SMART SOCIAL FEED

- Social Feed List
- License
- Documentation

**Social Feed List** Create New

**Settings Have Been Updated**  
 Settings successfully updated.

ID	Label	Short name	Manage	
1	dev-insta-basic	dev_insta_basic		<input type="checkbox"/>

It contains five columns. These are listed below in detail.

### 1. ID

This column contains the ID of the created feed. This is used for parameter

feed\_id in tags for creating feed.

## 2. Label

This column consists of the name of the already created feeds.

## 3. Short Name

This column contains the short name of the created feed. This is used for parameter feed\_name in tags for creating feed.

## 4. Manage

It has three sub elements:

### 1. Edit:

This is used to edit the previously created feed.

### 2. Synchronize feed:

This is used to call API again and save data to the database.

So every time synchronize is clicked the api is called and data is fetched again and save to the database by removing previously saved data.

### 3. Clear cache :

It helps in removing the cached data from the database which was saved during the call to API.

## 5. Multiple Delete

The already created feeds can be deleted using this option.

Check the feeds that need to be omitted. now, select "remove" which will appear as soon as the feed is checked.

Apply submit to complete the process.

# Facebook

Social Feed List allows you to create or edit Facebook feeds in an easy way. CREATE NEW button helps in creating the required feed. Here, you can easily manage the previously created feed.

Due to Facebook's privacy policy you're not able to get posts from a profile, only

from a public Facebook Page. [More Info.](#)

You can edit or modify the fields whenever required.

# Smart Social Feed:Facebook (Default view)

## FACEBOOK SETTINGS

## Important URLs

Generate credentials from [Here](#)

More information about this [Here](#)

Set OAuth Redirect URI :

**[https://eedemo.zealousweb.com/ee6/index.php?ACT=45&ssf\\_provider=facebook](https://eedemo.zealousweb.com/ee6/index.php?ACT=45&ssf_provider=facebook)**

**[https://eedemo.zealousweb.com/ee6/index.php?ACT=45&ssf\\_provider=facebook\\_pages](https://eedemo.zealousweb.com/ee6/index.php?ACT=45&ssf_provider=facebook_pages)**

## Application ID

## Application Secret

Get User Token

## Application User Token

## Get Facebook Feed of

☒ User Feed

Due to Facebook's privacy policy you're not able to get posts from a profile, only from a public Facebook Page. [More Info](#)

☐ Page Feed

## Type of feed

☐ Photo

☐ Status

☐ Video

☐ Link

☐ Event

## Cache duration (In Minutes)

## Limit

Limit of total facebook feed you want to fetch from your facebook page.

There are multiple fields that can be changed as per your requirements. Details about those fields are mentioned below:

### 1. Important URLs:

Important URLs contains two sub-links.

#### 1. Generate Credentials from [here](#).

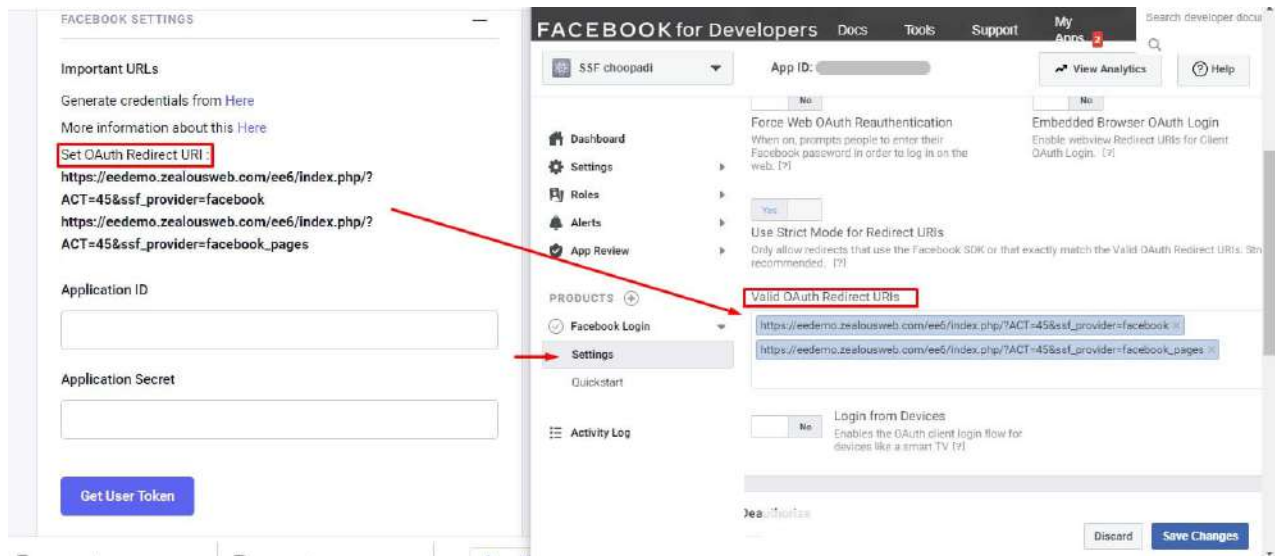
Click on this link to generate your Application ID and Application Secret for Facebook app.

#### 1. More information about this [here](#)

This section provides further details regarding the usage of the Facebook feature in this addon.

### 2. Set OAuth Redirect URI:

Copy the URL of "Set OAuth Redirect URI" of addon setting and paste to the "Valid OAuth Redirect URIs" textbox of the settings of the "Facebook Login" product.



### 1. Application ID:

This field contains your Facebook Application ID which is generated by your Facebook app.

### 2. Application Secret:

This field must contain your Facebook Application Secret which is generated

by your facebook app.

3. Facebook page ID:

This section contains your facebook page ID whose feed you want to fetch. In case, you do not have a Facebook ID, get one with the help of the link present below this field. .

4. Limit to page posts:

This field allows you to fetch data which is only posted or published by the page. In case you want this feature, select yes or no otherwise.

5. Types of feed:

This field contains multiple options to fetch the selected type of feed from the facebook page. For Instance, select videos option, in case, only the posts related to videos is required. You can select multiple options.

6. Cache Duration:

Duration in minutes after which data will be fetched from social sites api call instead of data cached in database. For given duration the data will be return from database that is saved when making first api call to social sites. Default value will be 30 minutes.

7. Limit:

Limit to number of posts or data you want to fetch. Default value will be 10.

## Twitter

Social Feed List allows you to create or edit Twitter feeds for a given Twitter user in an easy way. CREATE NEW button helps in creating the required feed.

Here, you can easily manage the previously created feed.

## Smart Social Feed:Twitter (Default view)

TWITTER SETTINGS

Important URLs

Generate credentials from [Here](#)

More information about this [Here](#)

Consumer Key (API Key)

Consumer Secret (API Secret)

Twitter Username

Include Retweets

Check this if you want to include retweets too

☒

Cache duration (In Minutes)

30

Limit

Limit of total tweets you want to fetch from your twitter page

10

You can edit or modify the fields whenever required. Details about these fields are mentioned below.

### 1. Important URLs:

Important URLs contains two sub-links.

1. Generate Credentials from [here](#).
2. Click on this link to generate your Consumer Key(API Key) and Consumer Secret(API Secret) for a Twitter a
  1. More information about this [here](#)

To get further information related to the usage of this addon for Twitter, click on this.

### 1. Consumer Key (API Key):

This field contains your Twitter Consumer Key which is generated from your Twitter app.

## 2. Consumer Secret(API Secret):

This field contains your Twitter Consumer Secret which is generated from your Twitter app.

## 3. Twitter Username:

To get the information related to the tweets of a certain user, this field is used. Enter the Twitter username of that user here.

## 4. Include Retweets:

Select Yes in case you want to fetch the retweets too ,otherwise, go for No option.

## 5. Cache Duration:

Mention the cache duration here only in minutes. During this period, the data is fetched from the database where it was saved while making the first API call to the social media api. After that period, data will again be fetched from API and will be saved to the database. This cycle goes on. The default value of this field is 30 minutes.

## 6. Limit:

It puts a limit on the number of posts or data you want to fetch. The default value is 10 posts.

# Instagram (Deprecated)

Social Feed List allows you to create or edit Instagram feeds for a given Instagram user in an easy way and also according to your requirement. CREATE NEW button helps in creating the required feed.

Here, you can easily manage the previously created feed.

# Smart Social Feed:Instagram (Default view)

[Social Feed List](#)  
[Licence](#)  
[Documentation](#)

**Feed Title**  
**Title \***   
Title of feed to identify feed.  
**Short Name \***   
Short name of feed. This short name will use in front end template tag to display feed on site.  
 Facebook Settings +  
 Twitter Settings +  
 Instagram Settings -  
**Important URLs** [Generate credentials from Here](#)  
More information about this [Here](#)  
**Instagram Access Token**   
Don't have an access token? Generate [Here](#)  
**Instagram User ID**   
Get Your Instagram User ID from [Here](#)  
**Cache duration (in Minutes)**   
**Limit**   
Limit of total Instagram posts you want to fetch from your Instagram account.  
 Youtube Settings +  
 Vimeo Settings +

ExpressionEngine 3.5.10
 [Report Bug](#) [New Ticket](#) [User Guide](#)

©2017 EllisLab, Inc.  
[scroll to top](#)

Edit or modify the fields whenever required. Details about these fields are mentioned below.

### 1. Important URLs:

Important URLs contains two sub-links.

1. Generate Credentials from [here](#).

You can generate your CLIENT ID and get REDIRECT URI for Instagram by following this link.

1. More information about this [here](#).

To get further information related to the usage of this addon for Instagram , click on this.

### 2. Instagram Access Token:

This field consists of your Instagram access token. In case you do not have your token, generate one from the link mentioned below this label as 'https://api.instagram.com/oauth/authorize/?client\_id=CLIENT-

ID&redirect\_uri=REDIRECT-URI&response\_type=token' where 'CLIENT-ID' and 'REDIRECT-URI'. Here, replace the 'CLIENT-ID' and 'REDIRECT-URI' with your Client-ID and Redirect-URI respectively. Get your access token in the url.

### 3. Instagram User ID:

This field contains Instagram User ID of the user whose feeds and posts you want to fetch.

NOTE: In case you are using sandbox mode for Instagram, then add that user to the sandbox account.

### 4. Cache Duration:

Mention the cache duration here only in minutes. During this period, the data is fetched from the database where it was saved while making the first API call to the social media api. After that period, data will again be fetched from API and will be saved to the database. This cycle goes on. The default value of this field is 30 minutes.

### 5. Limit:

It puts a limit on the number of posts or data you want to fetch. The default value is 10 posts.

## Instagram Basic Display API

Social Feed List allows you to create or edit Instagram feeds for the Instagram user easily and according to the user's requirement. The CREATE NEW button helps to create the required feed.

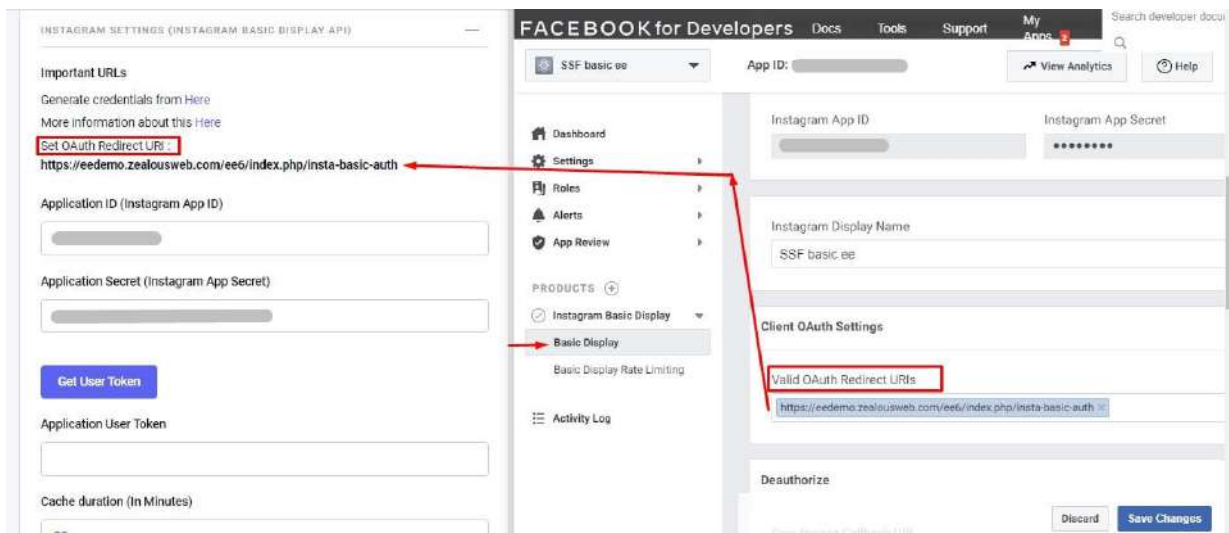
Check this page "[Authenticate Instagram Basic Display API](#)" to get step-wise authentication for Instagram basic display API.

## Smart Social Feed: Instagram Facebook Basic Display API(Default view)

Edit or modify the fields whenever required. Details about these fields are as follows.

1. **“Set OAuth Redirect URI”** Copy the Set OAuth Redirect URI from the addon setting and paste to the Valid OAuth Redirect URIs field of the Client OAuth Settings tab of the Basic Display page.

Note: Your website URL is a must with https.



2. Application ID (Instagram App ID):

This field contains the Instagram App ID of the user's app whose feeds and posts you want to fetch.

3. Application Secret (Instagram App Secret): This field contains the Instagram App Secret of the user's app whose feeds and posts you want to fetch.

4. Cache Duration:

Mention the cache duration here only in minutes.

During this period, the data is saved in the database and is fetched while making the first API call to the social media api.

After that duration, again the data will be fetched from the API and will be saved to the database. The cycle goes on.

The default value of this field is 30 minutes.

## Limitation

1. You get only the latest 25 feeds. We are working on it to get more feeds and it will be provided in the next version.
2. We provide only two types of feed (IMAGE, VIDEO). We do not provide a CAROUSEL\_ALBUM type of post. We are working on it to get the CAROUSEL\_ALBUM type of feed, and it will be provided in the next version.

# Instagram Facebook Graph API

You must have your Instagram Account linked to a Facebook Business Page. If you had not linked yet then you can get more detail about this thing from this [“Link your instagram to facebook”](#) page.

Social Feed List allows you to create or edit Instagram feeds for a given Instagram user in an easy way and also according to your requirement. CREATE NEW button helps in creating the required feed.

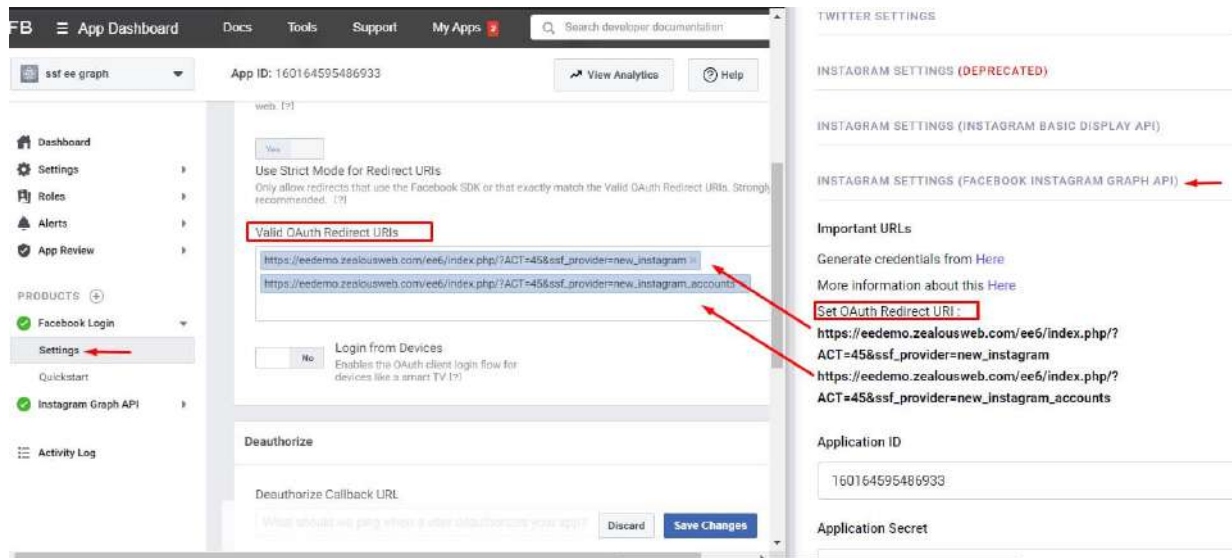
Check this page [“Authenticate Instagram Facebook Graph API”](#) to get each step of the process of authentication for Instagram Facebook graph API.

Here, you can easily manage the previously created feed.

Edit or modify the fields whenever required. Details about these fields are mentioned below.

1. Set OAuth Redirect URI: You need to add **“Set OAuth Redirect URI”** URL in **“Facebook Login-settings-> Valid OAuth Redirect URLs”** of facebook app.

Please see below screenshot.



2. Application ID: This field contains Facebook Application ID of the user's app whose feeds and posts you want to fetch.
3. Application Secret: This field contains Facebook Application Secret of the user's app whose feeds and posts you want to fetch.
4. Application User Token: You can get this token value once you click on the "Get User Token" button.
5. Instagram Account ID: You can select the Instagram account ID once you click on the "Get my Instagram Accounts" button.
6. Cache Duration: Mention the cache duration here only in minutes.  
 During this period, the data is fetched from the database where it was saved while making the first API call to the social media API.  
 After that period, data will again be fetched from API and will be saved to the database.  
 This cycle goes on. The default value of this field is 30 minutes.

## YouTube

Social Feed List allows you to create or edit YouTube feeds for a given YouTube channel in an easy way. CREATE NEW button helps in creating the required feed.

Here, you can easily manage the previously created feed also.

# Smart Social Feed:YouTube (Default view)

**YOUTUBE SETTINGS**

**Important URLs**  
Generate credentials from [Here](#)  
More information about this [Here](#)

**Channel ID**  
Find your Channel ID from [Here](#)

**API Key**  
Youtube library must be enabled

**Cache duration (In Minutes)**

**Limit**  
Limit of total videos you want to fetch from your youtube channel.

Edit or modify the fields whenever required. Details about these fields are mentioned below.

## 1. Important URLs:

Important URLs contains two sub-links.

### 1. Generate Credentials from [here](#).

Click here to generate your Google API Key.

### 2. More information about this [here](#)

Get further details about usage of this addon for YouTube from here.

## 2. Channel ID:

This field must contain your YouTube channel ID. If you don't have channel ID you can fetch it from the link mentioned below.

## 3. API Key:

This field must contain Google API Key.

## 4. Cache Duration:

Mention the cache duration here only in minutes. During this period, the data is fetched from the database where it was saved while making the first API call to the social media api. After that period, data will again be fetched from API and will be saved to the database. This cycle goes on. The default value of this field is 30 minutes.

## 5. Limit:

It puts a limit on the number of posts or data you want to fetch. The default value is 10 posts.

# Vimeo

Social Feed List allows you to create or edit Vimeo feeds for a given Vimeo user in an easy way. CREATE NEW button helps in creating the required feed.

Here, you can easily manage the previously created feed also.

## Smart Social Feed:Vimeo (Default view)

VIMEO SETTINGS

Important URLs

Generate credentials from [Here](#)

More information about this [Here](#)

Vimeo Access Token

Vimeo Feed Type

☐ User Feed

☒ Channel Feed

Vimeo Channel ID

Include Private Videos

For Channel Feeds only

☒

Cache duration (In Minutes)

Limit

Limit of total vimeo videos you want to fetch from your Vimeo account

Edit or modify the fields whenever required. Details about these fields are mentioned below.

### 1. Important URLs:

Important URLs contains two sub-links.

#### 1. Generate Credentials from [here](#).

Generate your Access token for Vimeo using this link.

#### 2. More information about this [here](#)

Get further details about usage of this addon for Vimeo here.

### 2. Vimeo Feed Type:

It gives two options. They are

#### 1. User Feeds

#### 2. Channel Feeds

The selection of an option depends on the type of videos you want to fetch. If you need user-uploaded videos then go for the User Feed, otherwise, select

Channel Feed for channel videos.

### 3. Vimeo User ID/Vimeo Channel ID:

This field must contain Vimeo User ID or Vimeo Channel ID which is entirely based on Vimeo Feed Type that you select

How to get User ID or Channel ID?

## Vimeo:User ID



## Vimeo:Channel ID



### 1. Include Private Videos:

Select Yes if you also want to include private videos while fetching Videos of the related channel, otherwise select No.

### 2. Cache Duration:

Mention the cache duration here only in minutes. During this period, the data is fetched from the database where it was saved while making the first API call to the social media api. After that period, data will again be fetched from API and will be saved to the database. This cycle goes on. The default value of this field is 30 minutes.

### 3. Limit:

Limit the number of posts or data you want to fetch. The default value is 10.

## Custom Sidebar

We provide one "customSidebar" function in the MCP file. You can add more menus in the sidebar of the addon using this function.

```
/* To get dynamic menu in the addon */  
function customSidebar(){  
    $sidebar = ee('CP/Sidebar')->make();  
    $this->navSettings = $sidebar->addHeader('Test Li  
}
```

# Facebook

You can create or edit a smart social feed for facebook for fetching facebook feeds or posts. Details for the same are shared below.

Tag for facebook feed will look like this.

```
{exp:smart_social_feed:pull} Content data {/exp:smart_sc
```

# Parameters

1. [feed\\_id](#)
2. [feed\\_name](#)
3. [platform](#)
4. [order\\_by](#)
5. [sort](#)

# Variables

1. `ssf_id`
2. `post_identifier`
3. `post_saved_time`

4. post\_from\_name
5. post\_from\_id
6. post\_id
7. platform
8. post\_name
9. post\_message
10. post\_description
11. post\_created\_time
12. post\_total\_likes
13. post\_total\_comments
14. post\_total\_shares
15. post\_thumbnail\_image
16. post\_full\_image
17. post\_url
18. post\_story
19. post\_type
20. post\_relative\_date
21. page\_source

Following Parameters can be use in Facebook feed

## **feed\_id**

Feed id of Feeds made from backend.

Example:

```
feed_id = "5"
```

## **feed\_name**

Name of feed being created from backend.

Example:

```
feed_name = "social_feed"
```

## platform

Select platforms if data to be displayed of specific platform.

Example:

```
platform = "facebook|twitter|instagram|youtube|vimeo"
```

## order\_by

This parameter is used to order the data by certain column .

Example:

```
order_by = "ssf_id"
```

## sort

This parameter is used to sort the fetched data. Default sorting is descending. You can either sort ascending, descending or random. ASC for ascending , DESC for descending and RAND for random.

Example:

```
{exp:smart_social_feed:pull feed_id="1" }
  {if no_results}<p>No results!</p>{/if}
  {if count == 1}
  <div class="feed-wrapper">
  {/if}
    <p>count                : {count}</p>
    <p>Total Results        : {total_results}</p>
    <p>id                    : {id}</p>
    <p>platform              : {platform}</p> //current
```

```

<p>SSF Id           : {ssf_id}</p>
<p>Post Identifier   : {post_identifier}</p>
<p>Post  Saved Time  : {post_saved_time format="
<p>Post Id           : {post_id}</p>
<p>Post From Name    : {post_from_name}</p>
<p>Post From Id      : {post_from_id}</p>
<p>Post Name         : {post_name}</p>
<p>Post Message      : {post_message}</p>
<p>Post Description   : {post_description}</p>
<p>Post Created Time  : {post_created_time}</p>
<p>Relative Date     : {post_relative_date}</p>
<p>Total Likes       : {post_total_likes}</p>
<p>Total Comments    : {post_total_comments}</p>
<p>Total Shares      : {post_total_shares}</p>
{if post_thumbnail_image} // get thumbnail image
<p>Thumbnail Image   : <
{/if}
<p>Post Url          : {post_url}</p>
<p>Post Story        : {post_story}</p>
<p>Post Type         : {post_type}</p>
{if post_type == "video" && post_source != ""} //
<div>
    <iframe src="{post_source}"></iframe>
</div>

```

```
        {/if}  
    {if count == total_results}  
    </div>  
    {/if}  
{/exp:smart_social_feed:pull}
```

# Twitter

You can create or edit a smart social feed for twitter for fetching twitter feeds or tweets. Details for the same are shared below.

Tag for twitter feed will look like this.

```
{exp:smart_social_feed:pull} Content data {/exp:smart_sc
```

# Parameters

1. [feed\\_id](#)
2. [feed\\_name](#)
3. [platform](#)
4. [order\\_by](#)
5. [sort](#)

# Variables

1. [ssf\\_id](#)
2. [post\\_identifier](#)
3. [post\\_saved\\_time](#)

4. platform
5. post\_from\_name
6. post\_from\_id
7. post\_id
8. post\_description
9. post\_created\_time
10. favorite\_count
11. retweet\_count
12. geo
13. coordinates
14. place
15. post\_relative\_date
16. post\_video

Following Parameters can be use in Twitter feed

## **feed\_id**

Feed id of Feeds made from backend.

Example:

```
feed_id = "5"
```

## **feed\_name**

Name of feed being created from backend.

Example:

```
feed_name = "social_feed"
```

## **platform**

Select platforms if data to be displayed of specific platform.

Example:

```
platform = "facebook|twitter|instagram|youtube|vimeo"
```

## order\_by

This parameter is used to order the data by certain column .

Example:

```
order_by = "ssf_id"
```

## sort

This parameter is used to sort the fetched data. Default sorting is descending. You can either sort ascending,descending or random. ASC for ascending , DESC for descending and RAND for random.

Example:

```
sort = "ASC"
```

## Example:

```
{exp:smart_social_feed:pull feed_id="1" }
  {if no_results}<p>No results!</p>{/if}
  {if count == 1}
    <div class="feed-wrapper">
      {/if}
      <p>count                : {count}</p>
      <p>Total Results       : {total_results}</p>
      <p>id                  : {id}</p>
      <p>platform            : {platform}</p> //current
      <p>SSF Id              : {ssf_id}</p>
      <p>Post Identifier      : {post_identifier}</p>
      <p>Post Saved Time     : {post_saved_time format="
```

```

<p>Post Id           : {post_id}</p>
<p>Post From Name    : {post_from_name}</p>
<p>Post From Id      : {post_from_id}</p>
<p>Post Description   : {post_description}</p>
<p>Post Created Time  : {post_created_time}</p>
<p>Total Favorites    : {favorite_count}</p>
<p>Total Retweets     : {retweet_count}</p>
<p>Geo                : {geo}</p>
<p>Coordinates        : {coordinates}</p>
<p>Place              : {place}</p>
<p>Post type          : {post_type}</p>

{if platform == "twitter"} // Give you proper re
<p>Relative Date      : {post_relative_date}</p>
{/if}

{if post_full_image} // If any image uploaded in
<p>Post full image    : 
    <iframe src="{post_video}"></iframe>

```

```
        </div>
      {/if}
    {if count == total_results}
  </div>
{/if}
{/exp:smart_social_feed:pull}
```

## Instagram (Deprecated)

You can create or edit a smart social feed for instagram for fetching instagram feeds or posts. Details for the same are shared below.

Tag for instagram feed will look like this.

```
{exp:smart_social_feed:pull} Content data {/exp:smart_sc
```

## Parameters

1. [feed\\_id](#)
2. [feed\\_name](#)
3. [platform](#)
4. [order\\_by](#)
5. [sort](#)

## Variables

1. [ssf\\_id](#)
2. [post\\_identifier](#)

3. post\_saved\_time
4. platform
5. post\_from\_name
6. post\_from\_id
7. post\_id
8. post\_created\_time
9. post\_total\_likes
10. post\_total\_comments
11. post\_thumbnail\_image
12. post\_full\_image
13. post\_url
14. post\_relative\_date
15. post\_video
16. post\_description
17. post\_type
18. post\_carousel

Following Parameters can be use in Instagram feed

## **feed\_id**

Feed id of Feeds made from backend.

Example:

```
feed_id = "5"
```

## **feed\_name**

Name of feed being created from backend.

Example:

```
feed_name = "social_feed"
```

## platform

Select platforms if data to be displayed of specific platform.

Example:

```
platform = "facebook|twitter|instagram|youtube|vimeo"
```

## order\_by

This parameter is used to order the data by certain column .

Example:

```
order_by = "ssf_id"
```

## sort

This parameter is used to sort the fetched data. Default sorting is descending. You can either sort ascending, descending or random. ASC for ascending , DESC for descending and RAND for random.

Example:

```
sort = "ASC"
```

## Example:

```
{exp:smart_social_feed:pull feed_id="1" }
  {if no_results}<p>No results!</p>{/if}
  {if count == 1}
    <div class="feed-wrapper">
      {/if}
      <p>count                                : {count}</p>
      <p>Total Results                        : {total_results}</p>
      <p>id                                  : {id}</p>
```

```

<p>platform                      : {platform}</p> //cu
<p>SSF Id                        : {ssf_id}</p>
<p>Post Identifier               : {post_identifier}</p>
<p>Post Saved Time              : {post_saved_time for
<p>Post Id                      : {post_id}</p>
<p>Post From Name               : {post_from_name}</p>
<p>Post From Id                : {post_from_id}</p>
<p>Post Created Time           : {post_created_time}<
<p>Relative Date                : {post_relative_date}
<p>Total Likes                  : {post_total_likes}</
<p>Total Comments              : {post_total_comments
<p>Post thumbnail Image         : {post_thumbnail_imag
<p>Post Full Image              : {post_full_image}</p>
<p>Post URL                     : {post_url}</p>

```

```
//New Variables added in v1.1.7
```

```

<p>Post Video                   : {post_video}</p>
<p>Post Description             : {post_description}</
<p>Post Type                    : {post_type}</p> // v

```

```
<div>
```

```
    Post Carousel              :
```

```
    {post_carousel}
```

```
        {if post_carousel:count == 1}<div>{/if} /
```

```
        {post_carousel:type} // video, image
```

```
        {post_carousel:url} // Video or Image URI
```

```
        {if post_carousel:count == post_carousel:
```

```
        {/post_carousel}
```

```
</div>
{if count == total_results}
</div>
{/if}
{/exp:smart_social_feed:pull}
```

# Instagram Basic Display API

You can create or edit a smart social feed for instagram to fetch Instagram feeds or posts. Details for the same are shared below.

Tag for instagram feed will look like this.

```
{exp:smart_social_feed:pull} Content data {/exp:smart_sc
```

## Parameters

1. [feed\\_id](#)
2. [feed\\_name](#)
3. [platform](#)
4. [order\\_by](#)
5. [sort](#)
6. [limit](#)
7. [post\\_type](#)

## Variables

1. [ssf\\_id](#)

2. post\_saved\_time
3. platform
4. post\_from\_name
5. post\_id
6. post\_created\_time
7. post\_thumbnail\_image: This variable gives result only for "VIDEO" type of feed.
8. post\_full\_image
9. post\_relative\_date
10. post\_video
11. post\_description
12. post\_type

The following parameters can be used for the Instagram feed

## **feed\_id**

Feed id of Feeds made from backend.

Example:

```
feed_id = "1"
```

## **feed\_name**

Name of feed being created from backend.

Example:

```
feed_name = "instagram_basic_display"
```

## **platform**

Select platforms if data to be displayed on a specific platform.

Example:

```
platform = "facebook|twitter|basic_instagram|youtube|vi
```

## order\_by

This parameter is used to order the data by certain column .

Example:

```
order_by = "ssf_id"
```

## sort

This parameter is used to sort the data that is fetched. Default sorting is descending. You can either sort ascending,descending or random. ASC for ascending , DESC for descending and RAND for random.

Example:

```
sort = "ASC"
```

## limit

This parameter is used to get only a specified number of feeds. You can set a limit from 1 to 25.

Example:

```
limit = "4"
```

## post\_type

This parameter is used to get only specific types of feeds. You can set a specific type like IMAGE, VIDEO.

Example:

```
post_type = "IMAGE"
```

```
post_type = "VIDEO"
```

```
post_type = "IMAGE|VIDEO"
```

## Example:

```

{exp:smart_social_feed:pull feed_id="1" }

{if no_results}<p>No results!</p>{/if}

{if count == 1}

<div class="feed-wrapper">

{/if}

<p>count                                : {count}</p>

<p>Total Results                        : {total_results}</p>

<p>id                                   : {id}</p>

<p>platform                            : {platform}</p> //cu

<p>SSF Id                              : {ssf_id}</p>

<p>Post Saved Time                     : {post_saved_time for

<p>Post Id                              : {post_id}</p>

<p>Post From Name                      : {post_from_name}</p>

<p>Post Created Time                   : {post_created_time f

```

```
<p>Relative Date           : {post_relative_date}

<p>Post thumbnail Image    : {post_thumbnail_imag

<p>Post Full Image         : {post_full_image}</p>

<p>Post Video              : {post_video}</p>

<p>Post Description        : {post_description}</

<p>Post Type               : {post_type}</p> // v

{if count == total_results}

</div>

{/if}

{/exp:smart_social_feed:pull}
```

# Instagram Facebook Graph API

You can create or edit a smart social feed for instagram for fetching instagram feeds or posts. Details for the same are shared below.

Tag for instagram feed will look like this.

```
{exp:smart_social_feed:pull} Content data {/exp:smart_so
```

## Parameters

1. `feed_id`
2. `feed_name`
3. `platform`
4. `order_by`
5. `sort`

## Variables

1. `ssf_id`
2. `post_identifier`
3. `post_saved_time`
4. `platform`
5. `post_from_name`
6. `post_from_id`
7. `post_id`
8. `post_created_time`
9. `post_total_likes`
10. `post_total_comments`
11. `post_thumbnail_image`: This variable gives result only for "VIDEO" type of feed.
12. `post_full_image`
13. `post_url`

14. post\_relative\_date

15. post\_video

16. post\_description

17. post\_type

18. post\_carousel

Following Parameters can be use in Instagram feed

## **feed\_id**

Feed id of Feeds made from backend.

Example:

```
feed_id = "5"
```

## **feed\_name**

Name of feed being created from backend.

Example:

```
feed_name = "social_feed"
```

## **platform**

Select platforms if data to be displayed of specific platform.

Example:

```
platform = "facebook|twitter|instagram|youtube|vimeo"
```

## **order\_by**

This parameter is used to order the data by certain column .

Example:

```
order_by = "ssf_id"
```

## **sort**

This parameter is used to sort the fetched data. Default sorting is descending. You can either sort ascending, descending or random. ASC for ascending , DESC for descending and RAND for random.

Example:

```
sort = "ASC"
```

## Example:

```
{exp:smart_social_feed:pull feed_id="1" }

{if no_results}<p>No results!</p>{/if}

{if count == 1}

<div class="feed-wrapper">

{/if}

<p>count                                : {count}</p>

<p>Total Results                        : {total_results}</p>

<p>id                                   : {id}</p>

<p>plateform                           : {plateform}</p> //cu

<p>SSF Id                              : {ssf_id}</p>
```

<p>Post Identifier : {post\_identifier}</p>

<p>Post Saved Time : {post\_saved\_time for

<p>Post Id : {post\_id}</p>

<p>Post From Name : {post\_from\_name}</p>

<p>Post From Id : {post\_from\_id}</p>

<p>Post Created Time : {post\_created\_time}<

<p>Relative Date : {post\_relative\_date}

<p>Total Likes : {post\_total\_likes}</

<p>Total Comments : {post\_total\_comments

<p>Post thumbnail Image : {post\_thumbnail\_imag

<p>Post Full Image : {post\_full\_image}</p>

<p>Post URL : {post\_url}</p>

//New Variables added in v1.1.7

```

        <p>Post Video                : {post_video}</p>

        <p>Post Description          : {post_description}</p>

        <p>Post Type                : {post_type}</p> // v

    {if count == total_results}

</div>

{/if}

{/exp:smart_social_feed:pull}

```

## YouTube

You can create or edit a smart social feed for YouTube for fetching YouTube feeds or videos. Details for the same are shared below.

Tag for YouTube feed will look like this.

```
{exp:smart_social_feed:pull} Content data {/exp:smart_sc
```

## Parameters

1. [feed\\_id](#)
2. [feed\\_name](#)

3. `platform`
4. `order_by`
5. `sort`

# Variables

1. `ssf_id`
2. `post_identifier`
3. `post_saved_time`
4. `platform`
5. `post_from_name`
6. `post_from_id`
7. `post_id`
8. `post_name`
9. `post_description`
10. `post_created_time`
11. `post_thumbnail_image`
12. `post_full_image`
13. `post_url`
14. `youtube_embed`
15. `post_relative_date`
16. `post_video`

Following Parameters can be used in YouTube feed

## **feed\_id**

Feed id of Feeds made from backend.

Example:

```
feed_id = "5"
```

## feed\_name

Name of feed being created from backend.

Example:

```
feed_name = "social_feed"
```

## platform

Select platforms if data to be displayed of specific platform.

Example:

```
platform = "facebook|twitter|instagram|youtube|vimeo"
```

## order\_by

This parameter is used to order the data by certain column .

Example:

```
order_by = "ssf_id"
```

## sort

This parameter is used to sort the fetched data. Default sorting is descending. You can either sort ascending,descending or random. ASC for ascending , DESC for descending and RAND for random.

Example:

```
sort = "ASC"
```

## Example:

```
{exp:smart_social_feed:pull feed_id="1" }  
  {if no_results}<p>No results!</p>{/if}  
  {if count == 1}
```

```

<div class="feed-wrapper">
  {/if}

    <p>count                                : {count}</p>
    <p>Total Results                        : {total_results}</p>
    <p>id                                  : {id}</p>
    <p>platform                            : {platform}</p> //c
    <p>SSF Id                             : {ssf_id}</p>
    <p>Post Identifier                     : {post_identifier}</p>
    <p>Post Saved Time                    : {post_saved_time for
    <p>Post Id                            : {post_id}</p>
    <p>Post From Name                     : {post_from_name}</p>
    <p>Post From Id                      : {post_from_id}</p>
    <p>Post Created Time                  : {post_created_time}<
    <p>Relative Date                      : {post_relative_date}
    <p>Post Description                   : {post_description}</
    <p>Post thumbnail Image              : {post_thumbnail_imag
    <p>Post Full Image                   : {post_full_image}</p>
    <p>Post URL                          : {post_url}</p>
    <p>Youtube Embed                     : {post_video}</p>

    {if count == total_results}
  </div>

  {/if}

{/exp:smart_social_feed:pull}

```

# Vimeo

You can create or edit a smart social feed for YouTube for fetching YouTube feeds or videos. Details for the same are shared below.

Tag for youtube feed will look like this.

```
{exp:smart_social_feed:pull} Content data {/exp:smart_sc
```

## Parameters

1. [feed\\_id](#)
2. [feed\\_name](#)
3. [platform](#)
4. [order\\_by](#)
5. [sort](#)

## Variables

1. `ssf_id`
2. `post_identifier`
3. `post_saved_time`
4. `platform`
5. `post_from_id`
6. `post_id`
7. `post_name`
8. `post_description`
9. `post_created_time`
10. `post_total_likes`
11. `post_total_comments`
12. `post_thumbnail_image`

13. post\_full\_image

14. vimeo\_embed

15. post\_url

16. post\_relative\_date

17. post\_video

Following Parameters can be used in Vimeo feed

## **feed\_id**

Feed id of Feeds made from backend.

Example:

```
feed_id = "5"
```

## **feed\_name**

Name of feed being created from backend.

Example:

```
feed_name = "social_feed"
```

## **platform**

Select platforms if data to be displayed of specific platform.

Example:

```
platform = "facebook|twitter|instagram|youtube|vimeo"
```

## **order\_by**

This parameter is used to order the data by certain column .

Example:

```
order_by = "ssf_id"
```

## **sort**

This parameter is used to sort the fetched data. Default sorting is descending. You can either sort ascending, descending or random. ASC for ascending , DESC for descending and RAND for random.

Example:

```
sort = "ASC"
```

## Example:

```
{exp:smart_social_feed:pull feed_id="1" }
  {if no_results}<p>No results!</p>{/if}
  {if count == 1}
  <div class="feed-wrapper">
  {/if}

    <p>count                                : {count}</p>
    <p>Total Results                        : {total_results}</p>
    <p>id                                   : {id}</p>
    <p>platform                             : {platform}</p>
    <p>SSF Id                               : {ssf_id}</p>
    <p>Post Identifier                       : {post_identifier}</p>
    <p>Post Saved Time                      : {post_saved_time for
    <p>Post Id                              : {post_id}</p>
    <p>Post Name                            : {post_name}</p>
    <p>Post From Id                         : {post_from_id}</p>
    <p>Post Description                     : {post_description}</p>
    <p>Post Created Time                   : {post_created_time}<
    <p>Relative Date                        : {post_relative_date}
    <p>Post Likes                           : {post_total_likes}</p>
    <p>Post Comments                       : {post_total_comments
```

```
<p>Post thumbnail Image      : {post_thumbnail_imag
<p>Post Full Image          : {post_full_image}</p>
<p>Post URL                  : {post_url}</p>
<p>Post Vimeo Embed         : {post_video}</p>
{if count == total_results}
</div>
{/if}
{/exp:smart_social_feed:pull}
```