[Methodology](https://blog.theodo.com/methodology)

**Automatically publish to Facebook pages with Python**

February 11, 2019[Sam Bunce](https://blog.theodo.com/sam-bunce)4 min read



Publishing to Facebook programmatically can seem like a complex affair, with several tricky steps, particularly authentication. Documentation of these steps can be scattered, obtuse, and deterring to the would-be poster.

In this article I’ll break down each step, with code snippets, from authentication to posting, as well as an overview of the concepts behind posting, and the administrative steps needed to go live. I’ll be using Python, but the concepts are quite simple and could be applied in most major languages.

First we’ll look at the technical steps to create a post from your app, then the steps needed to take your app through review and go live. This is by no means a definitive guide, but it’s a good starting point that will allow you to post automatically within a few minutes.

**Permissions & tokens**

The Facebook API represents any software system that interacts with it as an ‘app’. This may be an actual mobile or desktop app, website backend, AWS Lambda, or pretty much any code. The primary purpose is to identify your system for automated actions and verify its permissions. All of this is managed via [***Facebook for Developers***](https://developers.facebook.com/), which requires a Facebook account.

To identify your system, you’ll need to create an 'app' on *Facebook for Developers* and make sure you have an “Administrator” (the default) or “Developer” role. This means you can view test posts while the app is in development, and put the app through review. You will also need to be an administrator of the page that’s being posted to.

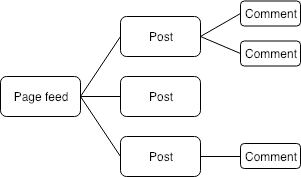
Access tokens are used to validate the app’s permissions whenever an action is attempted. They last for a set amount of time, and come in three main types:

* *User Access Tokens*, for interactions made as a user (posts, comments, seeing friend list). We’ll need one of these to generate a Page Access Token (below).
* *Page Access Tokens*, for interactions as a page (posts, moderation, advertising)
* *App Access Tokens*, for changing app settings

For today, we’ll focus on the first two. The Page Access Token will be used to post on the page, but in order to get it we must use the administrator's User Access Token.

**The Graph API**

Facebook’s Graph API represents pages, posts, comments, and other data as nodes in a graph.

**[](https://blog.theodo.com/static/c87ff0e10e48fb9887e63a38d80f8bf8/fb933/Graph-API.png)**A simplified example of the Graph API

In order to post to a page, we must create a new post node connected to our page’s node.

We can do this using the Facebook API for authentication, and an official or third-party SDK to handle the graph manipulation. In this example the snippets will be in Python, but the concepts are similar across all the SDKs.

Before you start, you’ll need the Page ID of the page you want to post on, and a short-term User Access Token for the page administrator. The former can be found in the page's settings, and the latter can be generated [**here**](https://developers.facebook.com/tools/explorer). In production, you'll need to get this using a "Continue with Facebook" button to get the user to authorise posting on their pages. Various libraries and techniques are available, but on our project we used [**python-social-auth**](https://github.com/python-social-auth/social-core).

**How it works**

You’ll need the facebook graph API url, which is currently https://graph.facebook.com/v3.0  
You’ll also need your *App ID* and *App Secret*, available via the app’s dashboard.

First we get the user's ID from the following URL:

https://graph.facebook.com/v3.0/me

?access\_token=YOUR\_USER\_ACCESS\_TOKEN

This can be used to retrieve the list of all the pages they're admin of, including access tokens:

https://graph.facebook.com/v3.0/YOUR\_USER\_ID/accounts

?access\_token=YOUR\_USER\_ACCESS\_TOKEN

Which looks something like this:

[

{

'access\_token': 'a\_long\_string',

'name': 'Your page name',

'id': '123456789',

...

},

...

]

These access tokens are not permanent, so the calls will need to be made each time we want to post.  
With the list of pages,  
The Graph API itself is easy to use. We need to get the graph node for the page:

import facebook

graph = facebook.GraphAPI(access\_token=fb\_page['access\_token'], version="3.0")

And then add the post object to the page

graph.put\_object(

parent\_object=YOUR\_PAGE\_ID,

connection\_name="feed",

message='I just posted automatically with python!',

)

After this, the test post appears on the page:

**[](https://blog.theodo.com/static/0dad2ab6660376c8ee72d9432769e24e/50383/Screenshot-2019-02-11-at-14.35.41.png)**

**App Review**

In order to post for real, we’ll need 2 permissions:

* manage\_pages
* publish\_pages

These permissions will let us create posts on Facebook pages. To get them we’ll need to go through App Review, which is performed by a human at Facebook. Until they are granted (and the app is made live), posts will only be visible to Administrators and Developers in the app.

The bulk of app review is demonstrating how your app is used. You’ll need a screencast of the activities that use Facebook, and instructions to help the reviewer test it. You’ll also need to make sure your app conforms to Facebook’s policies and guidelines, and of course data protection laws.

After this, you’ll need to wait for a team member at Facebook to approve the review, which may take several days. Once it’s been approved, all you need to do is flip the switch and make it live. Voila!