

Facebook oauth and Graph API with Django



[Laurent Luce](#) written 10 years ago



In this post, I am going to describe how I integrated Facebook into [Gourmious](#), so users could post their favorite dishes on Gourmious and on Facebook at the same time.

I wanted to keep the Gourmious login so the users could decide to login using their Gourmious credentials or using the Facebook login feature. I also wanted all users to have a Gourmious account so I didn't allow users to login using their Facebook credentials if they didn't have a Gourmious account. I decided to support the following 3 scenarios :

- user has a Django account and he logs in with Facebook. I associate his Facebook account to his Django account. This needs to be done only once.
- user does not have a Django account and tries to login using Facebook. I ask him first to create a Django account and I associate both accounts.
- user logs in using his Django credentials.

Facebook oauth is easier than the old Facebook connect. I was so happy to migrate to this new scheme.

I assume that you already have a server running your Django application and a Facebook app.

We are going to use the Django app `django_facebook_oauth` to make our life easier. Make sure you have `python simplejson` package installed.

Facebook Platform uses the OAuth 2.0 protocol for authentication and authorization. Read this first before continuing.
[Facebook authentication](#)

Clone the Django Facebook oauth app source code:

```
1 | git clone http://github.com/dickeytk/django_facebook_oauth.git
```

Copy the folder `django_facebook_oauth` to your Django project apps folder and rename it to `facebook`.

We will assume that your apps folder is called `apps`.

Django settings.py

We need to make the following changes:

- Add `'apps.facebook.backend.FacebookBackend'` to `AUTHENTICATION_BACKENDS`.
- Add `'apps.facebook'` to `INSTALLED_APPS`
- Add your Facebook API app ID: `API_ID = xxxx`.
- Add your Facebook secret key: `APP_SECRET = xxxx`.

Django urls.py

Add the following to `urlpatterns`:

```
1 | (r'', include('apps.facebook.urls')),
```


Database changes

Recent Posts

- [Least frequently used cache eviction scheme with complexity O\(1\) in Python](#)
- [Cambridge city geospatial statistics](#)
- [Massachusetts Census 2010 Towns maps and statistics using Python](#)
- [Python, Twitter statistics and the 2012 French presidential election](#)
- [Twitter sentiment analysis using Python and NLTK](#)
- [Python dictionary implementation](#)
- [Python string objects implementation](#)
- [Python integer objects implementation](#)
- [Python and cryptography with pycrypto](#)
- [Python list implementation](#)

Search

To search, type and hit ent



Meta

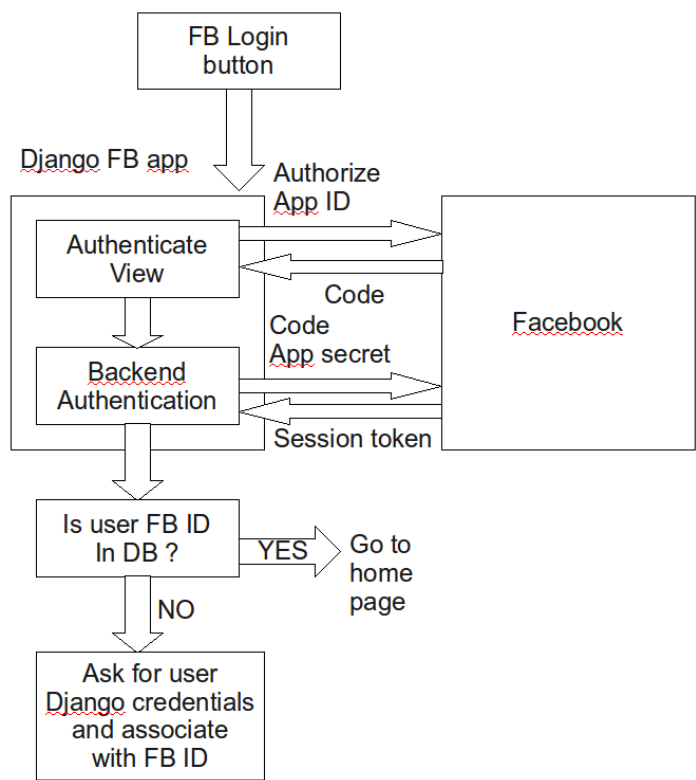
- [Log in](#)
- [Entries feed](#)
- [Comments feed](#)
- [WordPress.org](#)



You need to add the following table so you can associate Django user IDs with Facebook user IDs and store the Facebook session token. You can use the syncdb feature or create the table manually.

```
1 CREATE TABLE `facebook_facebookuser` (  
2   `id` int(11) NOT NULL auto_increment,  
3   `user_id` int(11) NOT NULL,  
4   `facebook_id` varchar(150) NOT NULL,  
5   `access_token` varchar(150) default NULL,  
6   PRIMARY KEY (`id`),  
7   UNIQUE KEY `user_id` (`user_id`),  
8   UNIQUE KEY `facebook_id` (`facebook_id`)  
9 ) ENGINE=MyISAM DEFAULT CHARSET=utf8
```

Facebook authentication flow



- 1- user clicks on the Facebook login button
- 2- Django Facebook app authenticate view is called
- 3- facebook open graph authorize url is called with application ID, redirect uri = /authenticate, scope = 'publish_stream' to ask for permission to post to user's wall
- 4- Dajgno Facebook app authenticate view is called with a parameter called code.
- 5- authentication using the Django Facebook app backend by passing the code back to Facebook along with the app secret.
- 6- Facebook returns a session token to be used for actions like posting messages to the user's wall.
- 7- if the user is already in facebook_facebookuser table, login and redirect to home page. If user is not in the table, ask him to login using your app credentials so you can associate his Django app user id with his Facebook user id.

Django Facebook app views.py

I modified views.py to use my login form to associate Django user id with Facebook ID instead of using the register form included in the Django Facebook app.

When the token is returned and the user does not exist in the facebook table, I add a parameter to the request session to indicate that the user needs to login first so I can join the accounts.

```
1 if user != None:  
2     login(request, user)  
3     return HttpResponseRedirect('/')  
4 else:  
5     # lines added  
6     request.session['ask_to_login_facebook'] = '1'  
7     return HttpResponseRedirect('/')  
8     #return HttpResponseRedirect('/register/')
```

I check for this session parameter in my template to popup a login form.

When the user enters his Django app credentials, I add an entry to the Facebook table before logging in the user. This operation just needs to be done once. After that, I know how to match a user Facebook id to the Django app user id so the user can directly login using

the Facebook connect button.

This the code in my Django app views.py

```
01 # check credentials
02 user = authenticate(username=username, password=password)
03 if user is not None:
04     if user.is_active:
05         if 'fb_id' in request.session:
06             fb_user = FacebookUser(user = user,
07                                     facebook_id = request.session['fb_id'],
08                                     access_token = request.session['fb_token'])
09             fb_user.save()
10             del request.session['fb_id']
11             del request.session['fb_token']
12             login(request, user)
13             status = 0
14         else:
15             status = 1
16     else:
17         status = 1
18     msg = _('Invalid password')
```

Django Facebook app backend.py

I modified this file to also add the token to the request session so I can use it when I add an entry to the facebook user table in views.py.

```
1 try:
2     fb_user = FacebookUser.objects.get(facebook_id =
3     str(profile["id"]))
4 except FacebookUser.DoesNotExist:
5     request.session['fb_id'] = profile['id']
6     # line added
7     request.session['fb_token'] = access_token
8     return None
9 fb_user.access_token = access_token
10 fb_user.save()
```

Django app template

This is what I added to my Django app login form to support the Facebook login button.

```
1 <form id="form_authenticate" action="/authenticate"
2 method="POST">
3     <p>blablabla</a></p>
4     <input id="form_authenticate_button" type="image"
5     src="link_to_facebook_button"
6     onClick="javascript:$('#form_authenticate').submit();">
7 </form>
```

I also added a ‘Post to Facebook’ check box to my ‘Post dish review’ form so the user can decide what gets posted to his Facebook wall.

Post a message on the user’s wall

I used to post messages using the Javascript SDK but I now do it on the server side using the Facebook python SDK.

Add the file facebook.py (<http://github.com/facebook/python-sdk/blob/master/src/facebook.py>) to the Facebook app folder and rename it facebook_sdk.py .

We call put_wall_post() to post a message to the user’s wall.

Here is the code I am using in my Django app views.py to format the parameters before calling put_wall_post() .

```
01 def post_to_facebook(request):
02     try:
03         fb_user = FacebookUser.objects.get(user = request.user)
04         # GraphAPI is the main class from facebook_sdp.py
05         graph = GraphAPI(fb_user.access_token)
06         attachment = {}
07         message = 'test message'
08         caption = 'test caption'
09         attachment['caption'] = caption
10         attachment['name'] = 'test name'
```



```
11 attachment['link'] = 'link_to_picture'
12 attachment['description'] = 'test description'
13 graph.put_wall_post(message, attachment)
14 return 0
15 except:
16 logging.debug('Facebook post failed')
```

That’s it for now.


If you enjoyed this article, check out my web app [Gourmious](#) to discover and share your favorite restaurant dishes. It would be cool if you could add some of your favorite restaurant dishes.

Last modified: May 30, 2020

 Django


 Python

Author




[Laurent Luce](#)

Comments



[Rory](#) commented 9 years ago


Just as a hint. If you’re using pip & virtualenv (which you should use), you can just “pip install git+git:.....”. It would be easier than “git clone and then copy etc.”



[codepoet](#) commented 9 years ago

Hi Laurent


Thanks for your response. As per the details you have given i have cloned the application and put it in my project and rename it to facebook. when i check this facebook’s view i only found two methods ‘login’ and ‘callback’. The ‘authenticate_view’ is a new function you have written?



[codepoet](#) commented 9 years ago


Hi Laurent,

The First code you have written ,for login, is in facebook app’s view or you have created your own view.



[Laurent Luce](#) commented 9 years ago Author

@codepoet The first code snippet is part of the function authenticate_view of the Facebook app view.



[Miguel](#) commented 9 years ago



Hi, I have to do what you did with Gourmious, but I'm not sure about something, every time you log into the web is going to ask me to connect to facebook?, or just the first time and then asueme that I publish on facebook too?



Laurent Luce commented 9 years ago

Author

@Miguel: The first step is to associate the facebook account to your web site account. I took this approach as I wanted every user to have an account on my side. You will have to look at the advantages and disadvantages of that versus only using their Facebook account. When this is done, the user just needs to click on the 'Connect with Facebook' button each time he wants to login using his facebook account. An access token will be received by the Django backend. You can use this access token to post on their wall.



patrick commented 10 years ago

I'm new to Django and could use some debugging pointers. Should I be able to get this to work if my server is localhost?

I'm seeing the following error:
<http://localhost:80/authenticate/>{
"error": {
"type": "OAuthException",
"message": "Missing redirect_uri parameter."
}
});

Thanks



laurent commented 10 years ago

Author

@patrick

It should work.

What is redirect_uri set to in authenticate_view() in the Facebook Django library ?
What is your Facebook application URL set to ?



Eugene commented 10 years ago

Hi,
I have a couple of questions:

1) Does the http://github.com/dickeytk/django_facebook_oauth.git refer to http://github.com/digvan/django_facebook_oauth or <http://github.com/vikalp/django-facebook-oauth>?

2) Does the fb_token found in # check credentials refer to the oauth access token ?

That's all for now!.



laurent commented 10 years ago

Author

1) it refers to the package: http://github.com/dickeytk/django_facebook_oauth

2) Yes.

Comments are closed.

[← Upload files from Python to Django](#)



➤ [Upload to Django with progress bar using Ajax and jQuery.](#)

Copyright © 2020 [Laurent Luce's Blog](#). All rights reserved. Theme by [Mynote](#).

