Mahmoud Abdrabo
Wed 27/3/13 11:13AM

# FACEBOOK LOGIN IN DJANGO

1. To use facebook login in a Django app, we need to register an app on facebook,

    a. Go to **developers.facebook.com**

    b. From the tabs at the top select "**Apps**"

    c. Then click "**Create New App**"

    d. In the modal, write the app's name, and click "**Continue**"

    e. Now we'll be redirected to the app's settings, we need to tell to state the url of the django app

        i. if working on localhost: click on "**Website with Facebook Login**" and enter in "**Site URL**": [http://127.0.0.1:8000/](http://127.0.0.1:8000/)

        ii. if working on server (ex. test.herokuapp.com): write in "**App Domains**": test.herokuapp.com. and in "**Website with Facebook Login**" and enter in "**Site URL**": [http://test.herokuapp.com](http://test.herokuapp.com)

    f. Take a look at the top of the app's settings "**App ID**" and "**App Secret**" we'll need them later.

2. Now our Facebook app is all set, we need to let our Django app know about it

    a. First we need to create a new .py file call it "**facebook.py"** in the same folder were "**Setting.py**" and "**views.py**" are

    b. This is the content of the file:

```
APP_ID = xxxxxxxxxxx  # From facebook app's settings

APP_SECRET = xxxxxxxxxxx  # From facebook app's settings

LOGIN_REDIRECT_URL = http://127.0.0.1:8000  # The url that the user will be redirected to
                                              after logging in with facebook

FACEBOOK_PERMISSIONS = ['email', 'user_about_me']  # facebook permissions

SCOPE_SEPARATOR = ' '

def facebook_login(request):

    if request.REQUEST.get("device"):
            device = request.REQUEST.get("device")
        else:
            device = "user-agent"

        params = {}
        params["client_id"] = APP_ID
        params["redirect_uri"] = request.build_absolute_uri(reverse("facebook_login_done"))
        params['scope'] = SCOPE_SEPARATOR.join(FACEBOOK_PERMISSIONS)
        params["device"] = device

        url = "https://graph.facebook.com/oauth/authorize?" + urllib.urlencode(params)
        request.session['next'] = request.META['HTTP_REFERER']
        return HttpResponseRedirect(url)

def facebook_login_done(request):
    result = authenticate(request=request)
    login(request, result)
    if 'next' in request.session:
        next = request.session['next']
    return HttpResponseRedirect(LOGIN_REDIRECT_URL)



class FacebookBackend:
    def authenticate(self, request):
        user = None
        access_token = None
        # assume logging in normal way
```

```
params = {}
params["client_id"] = APP_ID
params["client_secret"] = APP_SECRET
params["redirect_uri"] = request.build_absolute_uri(reverse("facebook_login_done"))
params["code"] = request.GET.get('code', '')

url = ("https://graph.facebook.com/oauth/access_token?"
        + urllib.urlencode(params))
from cgi import parse_qs
userdata = urllib.urlopen(url).read()
res_parse_qs = parse_qs(userdata)
# Could be a bot query
if not ('access_token') in res_parse_qs:
    return None
access_token = res_parse_qs['access_token'][-1]
url = "https://graph.facebook.com/me?access_token=" + access_token
import simplejson as json
fb_data = json.loads(urllib.urlopen(url).read())
uid = fb_data['id']
if not fb_data:
    return None

try:
        fb_user = UserProfile.objects.get(facebook_uid=uid, email=fb_data.get('email', None))
        fb_user.accesstoken = access_token
        fb_user.save()
        return fb_user
```

```
            except UserProfile.DoesNotExist:

                username = fb_data.get('username')

                if not username:

                    username = uid

                userProfile = UserProfile.objects.create(username=username)

                userProfile.first_name = fb_data['first_name']

                userProfile.last_name = fb_data['last_name']

                userProfile.email = fb_data.get('email', None)

                userProfile.save()

              return userProfile


    def get_user(self, user_id):

        try:

            return User.objects.get(pk=user_id)

        except:

            return None
```

To give short notes on how this works, first the user clicks the facebook login button, this button will call the “**facebook_login**” function which will send to faebook requesting user permission, when accepted, it will call the “**facebook_login_done**” function which will then ask the “**FacebookBackend**” to authenticate that user then redirect to the template specified at the top of the file. “**FacebookBackend**” is an Authentication backend that needs to be declared as so from the “**Settings.py**” in “**AUTHENTICATION_BACKENDS = (** ‘django-app.facebook.FacebookBackend’, ...).

This Authentication Backend first checks if that user is already in the system, if yes it will login that user and return, if not it will then create a new user with the information taken from his/her facebook profile, login that user and return.

3. To call “**facebook_login**” function from our template we need to have a url assigned to it, so in **urls.py** add the following:

```
urlpatterns += patterns('django-app.facebook',

    url(r'^facebook/login/$', 'facebook_login', name="facebook_login"),
```

4. Finally, create a button in our template, let it be:

```
<a href=”{% url facebook_login %}>Login using Facebook</a>
```

That was a fast walkthrough, if anything is not clear, or there are bugs in the code feel free to contact me on abdrabo.mahmoud@gmail.com