
ECL Facebook Documentation

Release 1.4.1

Dan Loewenherz

October 31, 2014

1	Installation	3
2	Configuration	5
3	Authentication	7
4	Integrating with Django	9
5	TODO	11
6	Contributing, feedback, and questions	13
7	Indices and tables	15

ECL Facebook is a Facebook library for Python 2.7+. It makes the Facebook API a joy to use and has built-in integration for [Django](#) and [Flask](#).

If you have an issue to report or a feature request, add it at <https://github.com/elmcitylabs/ECL-Facebook/issues>.

Installation

ECL Facebook is on [PyPi](#), and we recommend installing via `pip`

```
$ pip install ecl-facebook
```

Configuration

If you'd like to use ECL Facebook for a stand alone application (e.g., in a script you're writing to download your tweets), you'll need to set the environment variables `FACEBOOK_KEY`, `FACEBOOK_SECRET`, `FACEBOOK_REDIRECT_URL`, and `FACEBOOK_SCOPE` with the values appropriate for your Facebook application.

```
export FACEBOOK_KEY="256064624431781"  
export FACEBOOK_SECRET="4925935cb93e3446eff851ddaf5fad07"  
export FACEBOOK_REDIRECT_URL="http://example.com/oauth/complete"  
export FACEBOOK_SCOPE="email"
```

If you're only interested in integration with Django, read the section below.

Authentication

We've made authentication very simple. Probably too simple, to be honest.

```
>>> from ecl_facebook.settings import DIALOG_URL
>>> DIALOG_URL
https://www.facebook.com/dialog/oauth?scope=email&redirect_uri=http%3A%2F%2Fexample.com%2Fredirect&c
```

After opening this URL in your browser and allowing the application, you'll be redirected to a page with a URL similar to the following.

```
http://example.com/redirect?code=AQDOvI5wqlwNXQ6AK9jepHW4LUKboJk7v9yLGeaFNCDcs1hchWpCYoqDF0FZFLS03YOZ
```

You'll need to paste this code in the code variable below.

```
>>> from ecl_facebook import Facebook
>>> code = "AQDOvI5wqlwNXQ6AK9jepHW4LUKboJk7v9yLGeaFNCDcs1hchWpCYoqDF0FZFLS03YOZJ1lLhrzQrQ7PNWD2iiZZ"
>>> facebook = Facebook()
>>> data = facebook.oauth.access_token(code=code)
>>> data
<Objectifier#dict access_token=str expires=str>
```

Congratulations, you have successfully authenticated with Facebook. data is an Objectifier object which should contain your token and its expiration time.

To call the API, use your newly-acquired access token and access token secret.

```
>>> facebook = Facebook(data.access_token)
>>> facebook.me()
<Objectifier#dict username=unicode first_name=unicode last_name=unicode verified=bool name=unicode l
```

So, yeah. That's it. Be fruitful and multiply.

Integrating with Django

What we did above is easy. For Django projects, we've made it even easier. In your views file,

```

from django.contrib.auth import authenticate, login
from django.http import HttpResponseRedirect

from ecl_facebook.django_decorators import facebook_begin, facebook_callback
from ecl_facebook import Facebook

from .models import User

# ...

@facebook_begin
def oauth_facebook_begin(request):
    pass

@facebook_callback
def oauth_facebook_complete(request, access_token, error):
    if error is None:
        facebook = Facebook(token)
        fbuser = facebook.me()
        user, _ = User.objects.get_or_create(
            facebook_id=fbuser.id, defaults={'access_token': access_token})
        user = authenticate(id=user.id)
        login(request, user)
        return HttpResponseRedirect(reverse('home'))
    else:
        # handle authentication exception
        pass

```

Of course, you'll need to have a URL with the name `home` defined in your URLs file. Now, add these values to your settings.

```

# The User model that you'll be using to authenticate with Facebook.
PRIMARY_USER_MODEL = "app.User"

AUTHENTICATION_BACKENDS = (
    # ...
    'ecl_facebook.backends.FacebookAuthBackend',
)

FACEBOOK_KEY = "256064624431781"
FACEBOOK_SECRET = "4925935cb93e3446eff851ddaf5fad07"

```

```
FACEBOOK_REDIRECT_URL = "http://example.com/oauth/complete"  
FACEBOOK_SCOPE = "email"
```

There's also setting called `FACEBOOK_CSRF_TOKEN_REQUIRED`, which is `True` by default. We don't suggest you change this one unless you have a really good reason.

Then map the above views in your `urls.py`.

```
# ...  
  
urlpatterns = patterns('app.views',  
    # ...  
    url(r'^oauth/facebook/begin$', 'oauth_facebook_begin'),  
    url(r'^oauth/facebook/complete$', 'oauth_facebook_complete'),  
)
```

You're done. Oh, you might also want to add some fields for storing the Facebook-related fields in your user model.

TODO

- Decorators for other popular Python frameworks.
- More comprehensive test suite.
- More users!

Contributing, feedback, and questions

- Bitbucket: <http://bitbucket.com/elmcitylabs/ecl-facebook>
- Github: <https://github.com/elmcitylabs/ecl-facebook>
- Email: opensource@elmcitylabs.com.
- Twitter: [@elmcitylabs](https://twitter.com/elmcitylabs)

Indices and tables

- *genindex*
- *modindex*
- *search*