python-flickr Documentation

Release latest

March 10, 2015

Contents

1 Features **3**

2 Installation **5**

3 Usage **7**

3.1 Authorization URL 7
3.2 Handling the Callback 7
3.3 Getting the Users recent activity feed 8
3.4 Add comment on a photo 8
3.5 Remove comment on a photo 8
3.6 Upload a photo 8
3.7 Catching errors 8

Python-Flickr is A Python library to interface with Flickr REST API & OAuth

CHAPTER 1

Features

- Photo Uploading
- Retrieve user information
- **Common Flickr methods**
  - Add/edit/delete comments
  - Add/edit/delete notes
  - And many more (very dynamic library)!!
- All responses return as nice dicts

CHAPTER 2

Installation

Installing Python-Flickr is simple:

```
$ pip install python-flickr
```

CHAPTER 3

Usage

3.1 Authorization URL

```
f = FlickrAPI(api_key='*your app key*',
          api_secret='*your app secret*',
          callback_url='http://www.example.com/callback/')

auth_props = f.get_authentication_tokens()
auth_url = auth_props['auth_url']

#Store this token in a session or something for later use in the next step.
oauth_token = auth_props['oauth_token']
oauth_token_secret = auth_props['oauth_token_secret']

print 'Connect with Flickr via: %s' % auth_url
```

Once you click "Allow" be sure that there is a URL set up to handle getting finalized tokens and possibly adding them to your database to use their information at a later date.

3.2 Handling the Callback

```
# oauth_token and oauth_token_secret come from the previous step
# if needed, store those in a session variable or something

f = FlickrAPI(api_key='*your app key*',
              api_secret='*your app secret*',
              oauth_token=oauth_token,
              oauth_token_secret=oauth_token_secret)

authorized_tokens = f.get_auth_tokens(oauth_verifier)

final_oauth_token = authorized_tokens['oauth_token']
final_oauth_token_secret = authorized_tokens['oauth_token_secret']

# Save those tokens to the database for a later use?
```

3.3 Getting the Users recent activity feed

```
# Get the final tokens from the database or wherever you have them stored

f = FlickrAPI(api_key='*your app key*',
              api_secret='*your app secret*',
              oauth_token=final_tokens['oauth_token'],
              oauth_token_secret=final_tokens['oauth_token_secret'])

recent_activity = f.get('flickr.activity.userComments')
print recent_activity
```

3.4 Add comment on a photo

```
# Assume you are using the FlickrAPI instance from the previous section

add_comment = f.post('flickr.photos.comments.addComment',
                     params={'photo_id': '6620847285', 'comment_text': 'This is a test comment.'})

#This returns the comment id if successful.
print add_comment
```

3.5 Remove comment on a photo

```
# Assume you are using the FlickrAPI instance from the previous section
# If the comment is already deleted, it will throw a FlickrAPIError (In this case, with code 2: Comme

del_comment = f.post('flickr.photos.comments.deleteComment', params={'comment_id':'45887890-66208472
print del_comment
```

3.6 Upload a photo

```
# Assume you are using the FlickrAPI instance from the previous section

files = open('/path/to/file/image.jpg', 'rb')
add_photo = f.post(params={'title':'Test Title!'}, files=files)

print add_photo  # Returns the photo id of the newly added photo
```

3.7 Catching errors

```
# Assume you are using the FlickrAPI instance from the previous section

try:
    # This comment was already deleted
    del_comment = f.post('flickr.photos.comments.deleteComment', params={'comment_id':'45887890-66208
except FlickrAPIError, e:
    print e.msg
```

```
    print e.code
    print 'Something bad happened :('
```