
Flask-Micropub Documentation

Release 0.2.4

Kyle Mahan

Nov 05, 2017

Contents

1	Flask-Micropub	3
1.1	Authentication	3
1.2	Authorization	3
1.3	CSRF	3
2	Example Usage	5
3	API Reference	7
3.1	Flask-Micropub	7
	Python Module Index	9

A Flask extension to support IndieAuth and Micropub clients.

A Flask extension to support IndieAuth and Micropub clients.

1.1 Authentication

Authentication uses the [IndieAuth](#) flow to confirm a user controls a particular URL, without requesting any sort of permissions or access token. Annotate an endpoint with `@micropub.authenticated_handler` and then call `micropub.authenticate` to initiate the login.

1.2 Authorization

Authorization uses the full [Micropub](#) flow to authenticate a user and then request an access token with which to make micropub requests. Annotate an endpoint with `@micropub.authorized_handler` and then call `micropub.authorize` to initiate the login.

1.3 CSRF

MicropubClient provides a simple mechanism to deter Cross-Site Request Forgery. Based on [this Flask snippet](#), we generate a random string, pass it to the indieauth service via the state parameter, and then confirm we get the same random string back later.

This helps prevent malicious sites from sending users to your indieauth endpoint against their will.

CHAPTER 2

Example Usage

```
from flask import Flask, request, url_for
from flask.ext.micropub import MicropubClient

app = Flask(__name__)
micropub = MicropubClient(app)

@app.route('/login')
def login():
    return micropub.authorize(
        me, scope=request.args.get('scope'))

@app.route('/micropub-callback')
@mikropub.authorized_handler
def micropub_callback(resp):
    print('success!', resp.me, resp.access_token, resp.next_url, resp.error)
```

See `example.py` for a more thorough example. Protocol details at <https://indiewebcamp.com/IndieAuth> and <https://indiewebcamp.com/Micropub>

3.1 Flask-Micropub

This extension adds the ability to login to a Flask-based website using [IndieAuth](https://indiewebcamp.com/IndieAuth), and to request an [Micropub](https://indiewebcamp.com/Micropub) access token.

class flask_micropub.AuthResponse (*me=None, micropub_endpoint=None, access_token=None, state=None, scope=None, error=None*)

Authorization response, passed to the `authorized_handler` endpoint.

me

string – The authenticated user’s URL. This will be non-None if and only if the user was successfully authenticated.

micropub_endpoint

string – The endpoint to POST micropub requests to.

access_token

string – The authorized user’s micropub access token.

state

string – The optional state that was passed to authorize.

scope

string – The scope that comes with the micropub access token

error

string – describes the error encountered if any. It is possible that the authentication step will succeed but the access token step will fail, in which case `me` will be non-None, and `error` will describe this condition.

class flask_micropub.MicropubClient (*app=None, client_id=None*)

Flask-Micropub provides support for IndieAuth/Micropub authentication and authorization.

authenticate (*me, state=None, next_url=None*)

Authenticate a user via IndieAuth.

Parameters

- **me** (*string*) – the authing user’s URL. if it does not begin with `https?://`, `http://` will be prepended.
- **state** (*string*, *optional*) – passed through the whole auth process, useful if you want to maintain some state, e.g. the starting page to return to when auth is complete.
- **next_url** (*string*, *optional*) – deprecated and replaced by the more general “state”. still here for backward compatibility.

Returns a redirect to the user’s specified authorization url, or <https://indieauth.com/auth> if none is provided.

authenticated_handler (*f*)

Decorates the authentication callback endpoint. The endpoint should take one argument, a `flask.ext.micropub.AuthResponse`.

authorize (*me*, *state=None*, *next_url=None*, *scope='read'*)

Authorize a user via Micropub.

Parameters

- **me** (*string*) – the authing user’s URL. if it does not begin with `https?://`, `http://` will be prepended.
- **state** (*string*, *optional*) – passed through the whole auth process, useful if you want to maintain some state, e.g. the starting page to return to when auth is complete.
- **next_url** (*string*, *optional*) – deprecated and replaced by the more general “state”. still here for backward compatibility.
- **scope** (*string*, *optional*) – a space-separated string of micropub scopes. ‘read’ by default.

Returns a redirect to the user’s specified authorization <https://indieauth.com/auth> if none is provided.

authorized_handler (*f*)

Decorates the authorization callback endpoint. The endpoint should take one argument, a `flask.ext.micropub.AuthResponse`.

init_app (*app*, *client_id=None*)

Initialize the Micropub extension if it was not given app in the constructor.

Parameters

- **app** (*flask.Flask*) – the flask application to extend.
- **client_id** (*string*, *optional*) – the IndieAuth client id, will be displayed when the user is asked to authorize this client. If not provided, the app name will be used.

f

`flask_micropub`, [7](#)

A

`access_token` (`flask_micropub.AuthResponse` attribute), 7
`authenticate()` (`flask_micropub.MicropubClient` method), 7
`authenticated_handler()` (`flask_micropub.MicropubClient` method), 8
`authorize()` (`flask_micropub.MicropubClient` method), 8
`authorized_handler()` (`flask_micropub.MicropubClient` method), 8
`AuthResponse` (class in `flask_micropub`), 7

E

`error` (`flask_micropub.AuthResponse` attribute), 7

F

`flask_micropub` (module), 7

I

`init_app()` (`flask_micropub.MicropubClient` method), 8

M

`me` (`flask_micropub.AuthResponse` attribute), 7
`micropub_endpoint` (`flask_micropub.AuthResponse` attribute), 7
`MicropubClient` (class in `flask_micropub`), 7

S

`scope` (`flask_micropub.AuthResponse` attribute), 7
`state` (`flask_micropub.AuthResponse` attribute), 7