flask-canvas Documentation

Release 0.1.0

Demian Brecht

September 15, 2014

Overview

This extension deals with authentication for a Facebook canvas-based application.

Installation

From PyPI:

```
pip install flask-canvas
```

From source:

```
git clone git@bitbucket.org:demianbrecht/flask-canvas.git
```

Usage

```python
import flask_canvas

# install will monkeypatch Flask in order to expose canvas_route
flask_canvas.install(my_flask_app)

# route your canvas-specific page
@my_flask_app.canvas_route('/app', methods=['POST'])
def canvas():
    return 'hello, world'

# route page requiring user data
@my_flask_app.canvas_route('/user', methods=['POST'])
def user(canvas_user):
    return canvas_user.request('/me')
```

Note: The user data parameter `must` be named `canvas_user`. If a parameter with that name exists, the user data will be passed to it. Otherwise, it is ignored. Meaning that you can still have canvas views that don't receive user data (or incur overhead of parsing the `signed_request` payload.

Configuration

The following app configuration values may be set:

- `CANVAS_CLIENT_ID`: Client ID supplied by Facebook

- `CANVAS_REDIRECT_URI`: The redirect URI specified in your app settings

- `CANVAS_CLIENT_SECRET`: Client secret supplied by Facebook

- `CANVAS_SCOPE`: Resources application requires access to

- `CANVAS_ERROR_URI` (optional, default: `"/"`: Where the user is redirected to on auth error (cancel)

- `CANVAS_SKIP_AUTH_CHECK` (optional, default: `False`): Useful if your application never changes the scope requested. If `True`, this will eliminate an extra request to the graph API to ensure that the users' current permission set matches a potentially updated list.

Running the example

To run the example project, simply run `make example` and direct your browser to https://apps.facebook.com/flask_canvas.

API

Flask extension for Facebook canvas-based applications

class flask_canvas.**User**

> **has_permissions**()
>> Check current user permission set
>>
>> Checks the current user permission set against the one being requested by the application.
>
> **request**(*path*, *data=None*, *method='GET'*)
>> Convenience Facebook request function.
>>
>> Utility function to request resources via the graph API, with the format expected by Facebook.

flask_canvas.**_authorize**()
> Redirect the user to Facebook's authorization page
>
> You can't just 302 a user as the app is rendered in an iframe

flask_canvas.**_canvas_route**(*self*, **args*, ***kwargs*)
> Decorator for canvas route

flask_canvas.**_decode**(*data*)
> Decodes the Facebook signed_request parts

flask_canvas.**_decode_signed_user**(*encoded_sig*, *encoded_data*)
> Decodes the ''POST''ed signed data

flask_canvas.**install**(*app*)
> Installs the Flask extension

f