pydiscourse Documentation

Release 1.1.1

Marc Sibson

Nov 10, 2020

Contents

Contents:

CHAPTER 1

Introduction

A Python library for working with Discourse.

This is a fork of the original Tindie version. It was forked to include fixes, additional functionality, and to distribute a package on PyPI.

CHAPTER 2

Goals

- Exceptional documentation
- Support all supported Python versions
- Provide functional parity with the Discourse API, for the currently supported version of Discourse (something of a moving target)

The order here is important. The Discourse API is itself poorly documented so the level of documentation in the Python client is critical.

CHAPTER 3

Installation

```
pip install pydiscourse
```

CHAPTER 4

Examples

Create a client connection to a Discourse server:

```
from pydiscourse import DiscourseClient
client = DiscourseClient(
        'http://example.com',
        api_username='username',
        api_key='areallylongstringfromdiscourse')
```

Get info about a user:

```
user = client.user('eviltrout')
print user

user_topics = client.topics_by('johnsmith')
print user_topics
```

Create a new user:

```
user = client.create_user('The Black Knight', 'blacknight', 'knight@python.org',
↪'justafleshwound')
```

Implement SSO for Discourse with your Python server:

```
@login_required
def discourse_sso_view(request):
    payload = request.GET.get('sso')
    signature = request.GET.get('sig')
    nonce = sso_validate(payload, signature, SECRET)
    url = sso_redirect_url(nonce, SECRET, request.user.email, request.user.id,␣
↪request.user.username)
    return redirect('http://discuss.example.com' + url)
```

CHAPTER 5

Command line

To help experiment with the Discourse API, pydiscourse provides a simple command line client:

```
export DISCOURSE_API_KEY=your_master_key
pydiscoursecli --host-http://yourhost --api-user-system latest_topics
pydiscoursecli --host-http://yourhost --api-user-system topics_by johnsmith
pydiscoursecli --host-http://yourhost --api-user-system user eviltrout
```

CHAPTER 6

Development

For patches, please ensure that all existing tests pass, that you have adequate tests added as necessary, and that all code is documented! The latter is critical. If you add or update an existing function, class, or module, please ensure you add a docstring or ensure the existing docstring is up-to-date.

Please use Google docstring format.

This *will* be enforced.

CHAPTER 7

Pull requests

Reviewing and merging pull requests is work, so whatever you can do to make this easier for the package maintainer not only speed up the process of getting your changes merged but also ensure they are. These few guidelines help significantly. If they are confusing or you need help understanding how to accomplish them, please ask for help in an issue.

- Please do make sure your chnageset represents a *discrete update*. If you would like to fix formatting, by all means, but don't mix that up with a bug fix. Those are separate PRs.
- Please do make sure that both your pull request description and your commits are meaningful and descriptive. Rebase first, if need be.
- Please do make sure your changeset does not include more commits than necessary. Rebase first, if need be.
- Please do make sure the changeset is not very big. If you have a large change propose it in an issue first.
- Please do make sure your changeset is based on a branch from the current HEAD of the fork you wish to merge against. This is a general best practice. Rebase first, if need be.

CHAPTER 8

Testing

The best way to run the tests is with tox:

```
pip install tox
detox
```

Or it's slightly faster cousin detox which will parallelize test runs:

```
pip install detox
detox
```

Alternatively, you can run the self test with the following commands:

```
pip install -r requirements.dev.txt
pip install -e .
python setup.py test
```

CHAPTER 9

Live Testing

You can test against a Discourse instance by following the [Official Discourse developement instructions][discoursedev]. For the impatient here is the quick and dirty version:

```
git clone git@github.com:discourse/discourse.git
cd discourse
vagrant up
vagrant ssh
cd /vagrant
bundle install
bundle exec rake db:migrate
bundle exec rails s
```

Once running you can access the Discourse install at http://localhost:4000.

[discoursedev]: https://github.com/discourse/discourse/blob/master/docs/VAGRANT.md "Discourse Vagrant"

CHAPTER 10

TODO

For a list of all operations:

> you can just run rake routes inside of the discourse repo to get an up to date list

Or check the old [*routes.txt*](https://github.com/discourse/discourse_api/blob/aa75df6cd851f0666f9e8071c4ef9dfdd39fc8f8/routes.txt) file, though this is certainly outdated.

CHAPTER 11

Indices and tables

- genindex
- modindex
- search