Webmention Tools

Release 0.4.1

Panayotis Vryonis, André Jaenisch

Nov 16, 2019

CONTENTS:

CHAPTER

ONE

WEBMENTION-TOOLS

Some simple tools in python to deal with Webmentions.

Note, that this package was formerly known as webmentiontools, but had to be renamed due to PEP-541. (Namely, not classified as abandoned project, because the author was reachable).

Currently:

- webmentiontools.send implements WebmentionSend that sends Webmentions.
- webmentiontools.urlinfo implements UrlInfo() that will rerurn usefull information about a web page, like title, the existance of an "in-reply-to" link, the author name, the author image, etc.
- webmentiontools.webmentionio provides a class to query webmention.io

There is also the corresponting command line tool, webmention-tools (which is also a simple example on how to use the library.

Check bin/demo.py on how to use the library to query webmention.io and present information for all URLs that mentioned http://indiewebcamp.com/webmention

1.1 Installation

pip install webmention-tools

1.2 Usage

Command line:

```
webmention-tools send `source` `target`
webmention-tools urlinfo `url`
```

or

Python code to send a Webmention:

```
from webmentiontools.send import WebmentionSend
source = 'URL of page sending the Webmention'
target = 'URL of page to receive the Webmention'
mention = WebmentionSend(source, target)
mention.send()
```

Python code to get info about a webpage.

```
from webmentiontools.urlinfo import UrlInfo
url = 'a link to a web page'
i = UrlInfo(url)
if i.error:
    print('There was an error getting %s' % url)
else:
    print('in-reply-to link: %s' % i.inReplyTo())
    print('publication date: %s' % i.pubDate())
    print('page title: %s' % i.title())
    print('image link: %s' % i.image())
```

1.3 Development

1. Create a virtualenv with python3
2. Change into that directory and clone the repository
3. Activate the virtualenv by `source`ing `bin/activate`
4. Change into the cloned repository and install dependencies via 'pip install -r requirements.txt'
5. Run `pytest --cov=webmentiontools` for unit tests with code coverage

CHAPTER

TWO

WEBMENTIONTOOLS

2.1 webmentiontools package

2.1.1 Submodules

2.1.2 webmentiontools.discover module

2.1.3 webmentiontools.parser module

2.1.4 webmentiontools.request module

Wrapper around requests

webmentiontools.request.**is_successful_response**(*response: requests.models.Response*) → bool

Checks status code of response for success.

Parameters **response** (*requests.models.Response*) – The response to check.

Returns Was response successful?

Return type bool

webmentiontools.request.**request_get_url**(*url: str*) → requests.models.Response

Makes a GET request against the url.

Parameters **url** (*str*) – The URL to send request to.

Returns Network response.

Return type requests.models.Response

webmentiontools.request.**request_head_url**(*url: str*) → requests.models.Response

Makes a HEAD request against the url.

Parameters **url** (*str*) – The URL to send request to.

Returns Network response.

Return type requests.models.Response

webmentiontools.request.**request_post_url**(*endpoint: str*, *source_url: str*, *target_url: str*) → requests.models.Response

Makes a POST request against the endpoint.

Parameters

- **endpoint** (*str*) – The URL to send request to.

- **source_url** (*str*) – URL of page containing a Webmention.
- **target_url** (*str*) – URL of reference in source_url

Returns Network response.

Return type requests.models.Response

2.1.5 webmentiontools.send module

2.1.6 webmentiontools.urlinfo module

2.1.7 webmentiontools.webmentionio module

Provides interface to interact with https://webmention.io/

class webmentiontools.webmentionio.**WebmentionIO**(*access_token=None*)
Bases: object

Wrapper for interacting.

Example: webmention_io_token = None # or set your token. wio = WebmentionIO(webmention_io_token) ret = wio.links_to_url('http://indiewebcamp.com/webmention') if not ret:

print(wio.error)

else:

for l in ret['links']: print(l['id'], l['source'], l['verified_date'])

api_links_req(*key*, *value*)
Queries API for WebMentions

links_to_domain(*domain*)
Queries API for results with the given domain

links_to_url(*url*)
Queries API for results with the given target

2.1.8 Module contents

This is Webmention tools!

CHAPTER

THREE

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

W

INDEX