SXCU

Release v0.1.0a0

Naveen M K

Oct 16, 2020

CONTENTS:

1 Installation **3**
- 1.1 Examples . . . 3
  - 1.1.1 Uploading a Image to sxcu.net . . . 3
  - 1.1.2 Uploading Image with `og_properties` . . . 3
    - 1.1.2.1 What is `og_properties`? . . . 3
    - 1.1.2.2 How to use it? . . . 3
  - 1.1.3 Creating a Short Link . . . 4
- 1.2 Reference . . . 4
  - 1.2.1 Python API Reference . . . 4
    - 1.2.1.1 sxcu.SXCU . . . 4
    - 1.2.1.2 sxcu.og_properties . . . 7
    - 1.2.1.3 sxcu.__client__.RequestClient . . . 8
- 1.3 Changelog . . . 8
  - 1.3.1 sxcu-v2.0.1 . . . 8
  - 1.3.2 sxcu-v2.0.0 . . . 8
  - 1.3.3 sxcu-v1.0.1 . . . 9
  - 1.3.4 sxcu-v1.0.0 . . . 9
    - 1.3.4.1 New Features . . . 9
    - 1.3.4.2 For developers . . . 9
  - 1.3.5 sxcu-v0.1.0-alpha.0 . . . 9

2 Indices and tables **11**

Index **13**

SCXU is an API wrapper for api located at sxcu.net. Pretty much this documentation covers with some example and API reference. Currently the documentation is in building stage.

CHAPTER

ONE

INSTALLATION

SXCU can be installed using `pip`

```
pip install sxcu
```

1.1 Examples

1.1.1 Uploading a Image to sxcu.net

To upload an image *foo.png* located in the same directory as the script and print the url:

```
import sxcu
connector = sxcu.SXCU()
result = connector.upload_image("foo.png")
print(result["url"])
```

Parameters acceped are

upload_image(file[, collection, . . .])	This uploads image to sxcu

1.1.2 Uploading Image with `og_properties`

1.1.2.1 What is `og_properties`?

The Open Graph protocol The Open Graph protocol enables any web page to become a rich object in a social graph.

sxcu.net allows the following properties to be changed for changing the way it embed in apps and websites.

1.1.2.2 How to use it?

Using the class

og_properties(color, description, title)	This is a helper class for main SXCU function.

A small example using this can be

```
import sxcu
connector = sxcu.SXCU()
```

(continues on next page)

(continued from previous page)

```
props = sxcu.og_properties(color="#000",description="some random post",title="I love␣
↪SXCU")
result = connector.upload_image("foo.png")
print(result["url"])
```

1.1.3 Creating a Short Link

The reference for it is

create_link(link)	Creates a new link.

A small example illustrating it can be.

```
import sxcu
connector = sxcu.SXCU()
result = connector.create_link("https://sxcu.syrusdark.website")
print(result["url"])
```

should print a shortened URL.

1.2 Reference

This it python API reference of the current things implemented.

1.2.1 Python API Reference

SXCU(subdomain, upload_token, file_sxcu)	The Main class for sxcu.net request
og_properties(color, description, title)	This is a helper class for main SXCU function.
RequestClient(headers)	`RequestClient` is internally used to communicated with `Requests` Library.

1.2.1.1 sxcu.SXCU

class sxcu.**SXCU**(*subdomain: str = None, upload_token: str = None, file_sxcu: str = None*)

The Main class for sxcu.net request

__init__(*subdomain: str = None, upload_token: str = None, file_sxcu: str = None*) → None

This initialise the handler

Parameters

- **subdomain** (`str`, optional) – The subdomain you get from sxcu.net
- **upload_token** (`str`, optional) – The upload token that comes along with subdomain
- **file_sxcu** (`str`,optional) – The sxcu file you have got. Parses only `RequestURL` and `upload_token`.

Note: The content in `.scxu` file has more priority than passed parameters.

Methods

__init__	This initialise the handler
collection_details	Get collection details and list of images (if any are uploaded) for a given collection
create_collection	Create a new collection on sxcu.net.
create_link	Creates a new link.
delete_image	Deletes images from sxcu.net
domain_list	This lists all the public domains available, sorted by upload count.
edit_collection	Edit an existing collection.
image_details	Get basic details about an image on sxcu.net or any of its subdomain
upload_image	This uploads image to sxcu
upload_text	Uploads an text to sxcu.net (via cancer-co.de)

static collection_details(*collection_id: str*) → Union[dict, list]

Get collection details and list of images (if any are uploaded) for a given collection

Parameters **collection_id** (`str`) – collection_id returned when creating a collection.

Returns The returned JSON from the request.

Return type `dict` or `list`

static create_collection(*title: str*, *private: bool = False*, *unlisted: bool = False*, *desc: str = None*) → Union[dict, list]

Create a new collection on sxcu.net.

Note: If you are creating one time / bot collections you must make them unlisted!

Parameters

- **title** (`str`) – The title of the collection.
- **private** (`bool`, optional) – Whether the collection should be private or not.
- **unlisted** (`bool`, optional) – Whether the collection should be unlisted or not.
- **desc** (`str`, optional) – The description of the collection.

Returns The returned JSON from the request.

Return type `dict` or `list`

create_link(*link: str*) → Union[dict, list]

Creates a new link.

Parameters **link** (`str`) – The link to which you want to redirect.

Returns The returned JSON from the request.

Return type `dict` or `list`

static delete_image(*delete_url: str*) → bool

Deletes images from sxcu.net

Parameters **delete_url** (str) – The delete URL returned from sxcu.net while uploading.

Returns Deleted or not

Return type bool

static domain_list(*count: int = - 1*) → list

This lists all the public domains available, sorted by upload count.

Parameters **count** (int, optional) – Number of domains to return. If count=``-1`` it lists all.

:param .. warning::: The returned list contains bytes. Using str.encode(). Please use bytes.decode() for decoding it.

Returns The returned JSON from the request.

Return type list

static edit_collection(*collection_id: str, collection_token: str, title: str = None, desc: str = None, unlisted: bool = False, regen_token: bool = False, empty_collection: bool = False, delete_collection: bool = False*) → dict

Edit an existing collection.

Parameters

- **collection_id** (str) – The ID of the collection to be edited.
- **collection_token** (str) – The current token of that collection.
- **title** (str, optional) – The new title of the collection.
- **desc** (str, optional) – The new description of the collection.
- **unlisted** (bool, optional) – If True the collection will be made unlisted.
- **regen_token** (bool, optional) – If True, it will generate a new token for the collection and return it in the response.
- **empty_collection** (bool, optional) – If True it will disassociate all of the images in the collection from it.
- **delete_collection** (bool, optional) – If True it will disassociate all of the images in the collection from it and delete the collection.

Returns The returned JSON from the request.

Return type dict

static image_details(*image_id: str = None, image_url: str = None*) → Union[dict, list]

Get basic details about an image on sxcu.net or any of its subdomain

Parameters

- **image_id** (str) – The id of the image. For example, if https://sxcu.net/QNeo92 is the image URL then QNeo92 will be the image_id.

Note: The image_id can be from any subdomain also as alway the id would be same.

- **imageUrl** (str) – The image URL returned of sucessful upload.For example, https://sxcu.net/QNeo92.

Returns The returned JSON from the request.

Return type dict or list

upload_image (*file: str, collection: str = None, collection_token: str = None, noembed: bool = False, og_properties: sxcu.sxcu.og_properties = None*) → Union[dict, list]

This uploads image to sxcu

Parameters

- **file** (str, optional) – The path of File to Upload
- **collection** (str, optional) – The collection ID to which you want to upload to if you want to upload to a collection
- **collection_token** (str, optional) – The collection upload token if one is required by the collection you're uploading to.
- **noembed** (bool, optional) – If True, the uploader will return a direct URL to the uploaded image, instead of a dedicated page.
- **og_properties** (*og_properties*, optional) – This will configure the OpenGraph properties of the file's page, effectively changing the way it embeds in various websites and apps.

Returns The returned JSON from the request.

Return type dict or list

static upload_text (*text: str*) → Union[dict, list]

Uploads an text to sxcu.net (via cancer-co.de)

Parameters **text** (str) – The text being uploaded.

Returns The returned JSON from the request.

Return type dict or list

1.2.1.2 sxcu.og_properties

class sxcu.**og_properties** (*color: str = None, description: str = None, title: str = None*)

This is a helper class for main SXCU function. This helps you to reuse the og_properties.

__init__ (*color: str = None, description: str = None, title: str = None*) → None

Initialize self. See help(type(self)) for accurate signature.

Methods

__init__	Initialize self.
export	Exports the Property set to a JSON file.

export () → str

Exports the Property set to a JSON file.

Returns Using json.dumps the content of JSON file is dumped.

Return type str

1.2.1.3 sxcu.__client__.RequestClient

class sxcu.__client__.**RequestClient**(*headers: dict = None*)

`RequestClient` is internally used to communicated with `Requests` Library.

__init__(*headers: dict = None*) → None

This initaite the handlers. :param headers: The extra header needed to be added to the Request. :type headers: `str`, optional

Methods

__init__	This initaite the handlers.
get	Pass all the parameter to `requests.get()`.
post	Pass all the parameter to `requests.post()`.

get(*url: str, headers: dict = None, **kwargs*) → requests.models.Response

Pass all the parameter to `requests.get()`. Also, adding the neccessary headers. Also, the newly passed header would overide the default. :param headers: The header needed to be added to the Request.

> **Important:** The header would overide the default header.

post(*url: str, headers: dict = None, **kwargs*) → requests.models.Response

Pass all the parameter to `requests.post()`. Also, adding the neccessary headers. Also, the newly passed header would overide the default. :param headers: The header needed to be added to the Request.

> **Important:** The header would overide the default header.

1.3 Changelog

1.3.1 sxcu-v2.0.1

[TOB]

1.3.2 sxcu-v2.0.0

- Introduced a logger class so that users can know what really happened. User's can just set a logging handle and it will automatically write logs as in any other library.
- Improved Interface with Requests. Instead of Directly calling it now logs them and goes through `__client__`.
- Moved version to `__version__` and added other meta data to it.
- Now it handle's server response codes. Previously it was rising JSON decode Error which was unexpected and could cause problems.

1.3.3 sxcu-v1.0.1

- Fixed an issue in reading `.sxcu` file. (PR-10)
- Fixed `__version__` and removed a dependency.

1.3.4 sxcu-v1.0.0

1.3.4.1 New Features

- Create a new logo.
- Added a missing API method `image_details()`.
- Fix a bug due to subdomain parsing while using `.sxcu` files.
- Add a missing endpoint of `edit_collection()`.
- Fixed a bug in `create_link()`

1.3.4.2 For developers

- Added docs at https://sxcu.syrusdark.website.
- Added a few Tests.
- Enforce formatting with pre-commit.
- Added test Coverage
- Linting For Pull Requests Added.

1.3.5 sxcu-v0.1.0-alpha.0

Initial release with basic structure.

CHAPTER

TWO

INDICES AND TABLES

- genindex
- modindex
- search

INDEX

Symbols

`__init__()` (*sxcu.SXCU method*), 4
`__init__()` (*sxcu.__client__.RequestClient method*), 8
`__init__()` (*sxcu.og_properties method*), 7

C

`collection_details()` (*sxcu.SXCU static method*), 5
`create_collection()` (*sxcu.SXCU static method*), 5
`create_link()` (*sxcu.SXCU method*), 5

D

`delete_image()` (*sxcu.SXCU static method*), 5
`domain_list()` (*sxcu.SXCU static method*), 6

E

`edit_collection()` (*sxcu.SXCU static method*), 6
`export()` (*sxcu.og_properties method*), 7

G

`get()` (*sxcu.__client__.RequestClient method*), 8

I

`image_details()` (*sxcu.SXCU static method*), 6

O

`og_properties` (*class in sxcu*), 7

P

`post()` (*sxcu.__client__.RequestClient method*), 8

R

`RequestClient` (*class in sxcu.__client__*), 8

S

`SXCU` (*class in sxcu*), 4

U

`upload_image()` (*sxcu.SXCU method*), 7
`upload_text()` (*sxcu.SXCU static method*), 7