
SXCU

Release 4.1.0

Naveen M K

May 02, 2022

CONTENTS:

1	Installation	3
1.1	Examples	3
1.1.1	Uploading a Image to sxcu.net	3
1.1.2	Uploading Image with <code>og_properties</code>	3
1.1.2.1	What is <code>og_properties</code> ?	3
1.1.2.2	How to use it?	3
1.1.3	Creating a Short Link	4
1.2	Using Logging	4
1.3	Reference	4
1.3.1	Python API Reference	4
1.3.1.1	<code>sxcu.SXCU</code>	5
1.3.1.2	<code>sxcu.OGProperties</code>	8
1.3.1.3	<code>sxcu.exceptions.SXCUErrors</code>	8
1.3.1.4	<code>sxcu.__client__.RequestClient</code>	8
1.4	Changelog	9
1.4.1	<code>sxcu-v4.1.0</code>	9
1.4.1.1	New Features	9
1.4.1.2	Other changes	9
1.4.2	<code>sxcu-v4.0.0</code>	9
1.4.2.1	Breaking changes	10
1.4.2.2	New Features	10
1.4.2.3	Bug fixes	10
1.4.2.4	Other changes	10
1.4.3	<code>sxcu-v3.2.0</code>	10
1.4.4	<code>sxcu-v3.1.0</code>	10
1.4.5	<code>sxcu-v3.0.0</code>	11
1.4.6	<code>sxcu-v2.0.0</code>	11
1.4.7	<code>sxcu-v1.0.1</code>	11
1.4.8	<code>sxcu-v1.0.0</code>	11
1.4.8.1	New Features	11
1.4.8.2	For developers	11
1.4.9	<code>sxcu-v0.1.0-alpha.0</code>	12
2	Indices and tables	13
	Index	15

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.

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 accepted are

<code>upload_image(*args, **kwargs)</code>	This method is deprecated.
--	----------------------------

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

<code>og_properties</code>	OGProperties declaration.
----------------------------	---------------------------

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

<code>create_link(link)</code>	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 Using Logging

Logging are useful to know what really happened. All the logging are set to DEBUG. To view the DEBUG logs you would need to configure the logging module to show DEBUG logs as below.

```
import logging
logging.basicConfig(level=logging.DEBUG)
import sxcu
a=sxcu.SXCU()
```

Now all the debug logs are printed to `stdout`. You could also configure it to write to a file but that is out the scope of this documentation.

1.3 Reference

This it python API reference of the current things implemented.

1.3.1 Python API Reference

<code>SXCU(subdomain, upload_token, sxcu_config, ...)</code>	The Main class for sxcu.net request
<code>OGProperties(color, bool] = None, ...)</code>	This is a helper class for main SXCU function.
<code>SXCUError</code>	SXCUError A error which will be raised when the returned status code isn't right.
<code>RequestClient(headers)</code>	RequestClient is internally used to communicated with Requests Library.

1.3.1.1 sxcu.SXCU

class sxcu.SXCU (subdomain: str = None, upload_token: str = None, sxcu_config: Union[str, dict, io.StringIO] = None, *, file_sxcu: str = None)

The Main class for sxcu.net request

__init__ (subdomain: str = None, upload_token: str = None, sxcu_config: Union[str, dict, io.StringIO] = 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
- **sxcu_config** (str, optional) – The sxcu configuration you have. Parses only RequestURL and upload_token.

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

- **file_sxcu** (str, optional) – File location to sxcu configuration. Kept for backwards compatibility.

Methods

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

static collection_details (*args: Any, **kwargs: Any) → Union[dict, list]

This method is deprecated. Use `collection_meta()` instead.

static collection_meta (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: Optional[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 (**args: Any, **kwargs: Any*) → Union[dict, list]

This method is deprecated.

Use `list_subdomain()` instead.

static file_meta (*file_id: str = None, file_url: str = None, 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

- **file_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 always the id would be same.

- **file_url** (*str*) – The image URL returned of successfully upload. For example, `https://sxcu.net/QNeo92`. Either one of `:param:file_id`` or `:param:file_url`` is required.

Returns The returned JSON from the request.

Return type dict or list

static image_details (*args: Any, **kwargs: Any) → Union[dict, list]

This method is deprecated.

Use `file_meta()` instead.

static list_subdomain (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.
- **warning:** (.) – The returned list contains bytes encoded using `str.encode()`. Please use `bytes.decode()` for decoding it.

Returns The returned JSON from the request.

Return type list

upload_file (name: str = None, fileobj: io.BytesIO = None, collection: Optional[str] = None, collection_token: Optional[str] = None, noembed: Optional[bool] = False, og_properties: Optional[sxcu.og_properties.OGProperties] = None, self_destruct: bool = False, *, file: str = None) → Union[dict, list]

This uploads image to sxcu

Parameters

- **name** – The pathname of file to upload.
- **file** – aliased to name, for backwards compatibility.
- **fileobj** – If fileobj is given, it is used for reading the file.
- **collection** – The collection ID to which you want to upload to if you want to upload to a collection
- **collection_token** – The collection upload token if one is required by the collection you're uploading to.
- **noembed** – If True, the uploader will return a direct URL to the uploaded image, instead of a dedicated page.
- **og_properties** – This will configure the OpenGraph properties of the file's page, effectively changing the way it embeds in various websites and apps.
- **self_destruct** – If True, the file will be deleted automatically after 24 hours.

Returns The returned JSON from the request.

Return type dict or list

Raises **SXCUErrror** – Any error from the request side.

upload_image (*args: Any, **kwargs: Any) → Union[dict, list]

This method is deprecated. Use `upload_file()` instead.

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

1.3.1.2 sxcu.OGProperties

class `sxcu.OGProperties` (*color: Union[str, bool] = None, description: Union[str, bool] = None, title: Union[str, bool] = None, discord_hide_url: bool = False, site_name: Union[str, bool] = None*)

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

__init__ (*color: Union[str, bool] = None, description: Union[str, bool] = None, title: Union[str, bool] = None, discord_hide_url: bool = False, site_name: Union[str, bool] = None*) → *None*
 Initialize self. See help(type(self)) for accurate signature.

Methods

<code>__init__</code>	Initialize self.
<code>export</code>	Exports the Property set to a JSON file.
<code>from_json</code>	Import the Property set from parsing JSON.

export () → *str*

Exports the Property set to a JSON file.

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

Return type *str*

classmethod from_json (*contents: str*) → `sxcu.og_properties.OGProperties`

Import the Property set from parsing JSON.

Parameters contents (*str*) – The contents in JSON which needs to be parsed.

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

Return type *str*

1.3.1.3 sxcu.exceptions.SXCUErrror

exception `sxcu.exceptions.SXCUErrror`

SXCUErrror A error which will be raised when the returned status code isn't right.

See `error_codes` and message list for more information in <https://sxcu.net>.

1.3.1.4 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 initiate the handlers. :param headers: The extra header needed to be added to the Request. :type headers: *dict*, optional

Methods

<code>__init__</code>	This initiate the handlers.
<code>get</code>	Pass all the parameter to <code>requests.get()</code> .
<code>post</code>	Pass all the parameter to <code>requests.post()</code> .

get (*url: str, headers: Optional[dict] = None, **kwargs*) → `requests.models.Response`
 Pass all the parameter to `requests.get()`. Also, adding the necessary headers. Also, the newly passed header would override the default.

Parameters

- **headers** – The header needed to be added to the Request.
- **important :: (.)** – The headers would override the default header.

post (*url: str, headers: dict = None, **kwargs*) → `requests.models.Response`
 Pass all the parameter to `requests.post()`. Also, adding the necessary headers. Also, the newly passed header would override the default.

Parameters headers (*str, optional*) – The header needed to be added to the Request.

Important: The header would override the default header.

1.4 Changelog

1.4.1 sxcu-v4.1.0

1.4.1.1 New Features

- Allow uploading a file directly from `io.BytesIO`, see `fileobj` parameter in `sxcu.SXCU.upload_file()`
- Allow passing `io.StringIO` file as config file in `SXCU`

1.4.1.2 Other changes

- Deprecate `file_sxcu` in `SXCU`, use `sxcu_config` instead.
- Deprecate `file` parameter in `sxcu.SXCU.upload_file()`, use `name` instead.

1.4.2 sxcu-v4.0.0

This contains support for `sxcu.net` API v2.

1.4.2.1 Breaking changes

- Removed support for `sxcu.net` API v1 which is deprecated and would be removed soon.
- Remove `edit_collection` method. I was informed it was to be removed soon from the API.
- A default logging handler which print debug info is removed. Please set up your own handler. Details at [Using Logging](#).
- Deprecate `upload_image`. Use `upload_file` instead.
- Deprecate `collection_details`. Use `collection_meta` instead.
- Deprecate `image_details`. Use `file_meta` instead.
- Deprecate `domain_list`. Use `list_subdomain` instead.

1.4.2.2 New Features

- Added support for `site_name` in `OGProperties`
- Added support for `self_destruct` in `upload_file`.

1.4.2.3 Bug fixes

- Fix `sxcu.__version__` printing displaying wrong version.

1.4.2.4 Other changes

- Miscellaneous typo fixes.
- Slightly improve language in request handler.
- Changed the default user agent to contain a URL to this project.

1.4.3 sxcu-v3.2.0

- Update for new API changes

1.4.4 sxcu-v3.1.0

- Added `OGProperties.from_json()` method.
- Fix broken Error Handler. It raises `SXCUError` when there is an error from server now.
- Update API details. (slightly)

1.4.5 sxcu-v3.0.0

- Rename `og_properties` to `OGProperties`.
- Add support for `discord_hide_url`.

1.4.6 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.

1.4.7 sxcu-v1.0.1

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

1.4.8 sxcu-v1.0.0

1.4.8.1 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()`

1.4.8.2 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.

1.4.9 sxcu-v0.1.0-alpha.0

Initial release with basic structure.

INDICES AND TABLES

- `genindex`
- `modindex`
- `search`

Symbols

`__init__()` (*sxcu.OGProperties method*), 8
`__init__()` (*sxcu.SXCU method*), 5
`__init__()` (*sxcu.__client__.RequestClient method*), 8

C

`collection_details()` (*sxcu.SXCU static method*), 5
`collection_meta()` (*sxcu.SXCU static method*), 5
`create_collection()` (*sxcu.SXCU static method*), 5
`create_link()` (*sxcu.SXCU method*), 6

D

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

E

`export()` (*sxcu.OGProperties method*), 8

F

`file_meta()` (*sxcu.SXCU static method*), 6
`from_json()` (*sxcu.OGProperties class method*), 8

G

`get()` (*sxcu.__client__.RequestClient method*), 9

I

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

L

`list_subdomain()` (*sxcu.SXCU static method*), 7

O

`OGProperties` (*class in sxcu*), 8

P

`post()` (*sxcu.__client__.RequestClient method*), 9

R

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

S

`SXCU` (*class in sxcu*), 5
`SXCUError`, 8

U

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