weebapi Documentation

Release 0.1.5b

AndyTempel

Mar 02, 2021

Contents:

This is a API wrapper library for invite only service called weeb.sh. This library provides a modern, pythonic and easy to use, object driven, interface.

This API wrapper also provides full discord.py rewrite integration. Check examples or Client object for more information.

CHAPTER 1

API Reference

The following section outlines the API of weeb.sh.

Note: This module uses the Python logging module to log diagnostic and errors in an output independent way. If the logging module is not configured, these logs will not be output anywhere.

1.1 Version Related Info

There are two main ways to query version information about the library.

`version_info`

A named tuple that is similar to sys.version_info.

Just like sys.version_info the valid values for `releaselevel` are 'alpha', 'beta', 'candidate' and 'final'.

`__version__`

A string representation of the version. e.g. `'0.1.0-alpha0'`.

1.2 Weeb.sh Client

class `weebapi.client.`**`Client`**(*api_key: str, wolke_token: bool = True, v2_api: bool = False, base_url: str = 'https://api.weeb.sh/', bot=None, loop=<_UnixSelectorEventLoop running=False closed=False debug=False>*)

Client object for Weeb.sh V2

This is a client object for Weeb.sh API. Here are two versions. Basic without discord.py bot and a pluggable version that inserts this client object directly into your discord.py bot.

Represents a client connection that connects to weeb.sh. It works in two modes:

1. As a standalone variable.

2. Plugged-in to discord.py Bot or AutoShardedBot, see `Client.pluggable`

Parameters

- **`api_key`** (`str`) – Your weeb.sh api token.
- **`wolke_token`** (`bool[Optional]`) – *Not required.* Is the api token type Wolke or Bearer. **Default:** True
- **`v2_api`** (`bool[Optional]`) – *Not required.* Use v2 API. **Default:** False
- **`**base_url`** (`str[Optional]`) – *Not required.* Specify different DBL API url.
- **`**base_url`** – *Not required.* Specify different base url.
- **`**bot`** (`Bot or AutoShardedBot`) – Your bot client from discord.py
- **`**loop`** (`asyncio loop`) – Your asyncio loop.

`classmethod pluggable`(*bot*, *api_key: str*, **args*, ***kwargs*)

Pluggable version of Client. Inserts Client directly into your Bot client. Called by using *bot.dbl*

Parameters

- **`bot`** (`discord.ext.commands.Bot or discord.ext.commands.AutoShardedBot`) – Your bot client from discord.py
- **`api_key`** (`str`) – Your weeb.sh api token.

Note: Takes the same parameters as `Client` class. Usage changes to `bot.weebsh`. (`bot` is your bot client variable)

`get_image`(*image: str*) → weebapi.data_objects.Image

This function is a *coroutine*.

This function gets image by it's ID.

Parameters **`image`** (`str` or `weebapi.data_objects.Image`) – Image ID from string or `weebapi.data_objects.Image` object.

Returns `weebapi.data_objects.Image`

`get_preview`(*type_name: str*, *hidden: bool = False*, *nsfw: int = 1*) → weebapi.data_objects.Preview

This function is a *coroutine*.

This function gets a preview of the specified image type.

`1 (Default)`

If the value of the nsfw parameter is `1`, content with nsfw will be hidden.

`2`

If the value of the nsfw parameter is `2`, both nsfw and non-nsfw content will be shown.

`3`

If the vaule of the nsfw parameter is `3`, only content marked with nsfw will be shown.

Parameters

- **`type_name`** (`str`) – Name of the image type for which you want the preview.
- **`**hidden`** (`bool`) – Search in hidden image types. Defaults to False.

- **`**nsfw`** (`int`) – Display options for NSFW content. Look for the valid values in the box above.

Returns *`weebapi.data_objects.Preview`*

`get_random`(*tags: str = None*, *image_type: str = None*, *nsfw: int = 1*, *hidden: bool = False*, *file_type: str = None*) → weebapi.data_objects.Image

This function is a *coroutine*.

This function gets a random image of the specified tags or the image type.

Note: Make sure you either use comma separated or list of tags and/or one image type. You cannot leave both `tags` nor `image_type` variable blank, one of them must be used.

`1 (Default)`
If the value of the nsfw parameter is `1`, content with nsfw will be hidden.

`2`
If the value of the nsfw parameter is `2`, both nsfw and non-nsfw content will be shown.

`3`
If the vaule of the nsfw parameter is `3`, only content marked with nsfw will be shown.

Parameters

- **`**tags`** (`str` or `list`) – One tag or comma separated tags or list of image tags.
- **`**image_type`** (`str`) – Specify image type.
- **`**hidden`** (`bool`) – Get hidden image tags. Defaults to False.
- **`**nsfw`** (`int`) – Display options for NSFW content. Look for the valid values in the box above.
- **`**file_type`** (`str`) – One of `jpg, jpeg, png, gif` file types.

Returns Returns the *`weebapi.data_objects.Image`* object.

`get_tags`(*hidden: bool = False*, *nsfw: int = 1*) → list

This function is a *coroutine*.

This function gets the list of all available image tags.

`1 (Default)`
If the value of the nsfw parameter is `1`, content with nsfw will be hidden.

`2`
If the value of the nsfw parameter is `2`, both nsfw and non-nsfw content will be shown.

`3`
If the vaule of the nsfw parameter is `3`, only content marked with nsfw will be shown.

Parameters

- **`**hidden`** (`bool`) – Get hidden image tags. Defaults to False.
- **`**nsfw`** (`int`) – Display options for NSFW content. Look for the valid values in the box above.

Returns Returns the list of *`weebapi.data_objects.Tag`* objects.

get_types (*hidden: bool = False*, *nsfw: int = 1*) → list

This function is a *coroutine*.

This function gets the list of all available image types.

1 (Default)
If the value of the nsfw parameter is `1`, content with nsfw will be hidden.

2
If the value of the nsfw parameter is `2`, both nsfw and non-nsfw content will be shown.

3
If the vaule of the nsfw parameter is `3`, only content marked with nsfw will be shown.

Parameters

- ****hidden** (`bool`) – Get hidden image types. Defaults to False.
- ****nsfw** (`int`) – Display options for NSFW content. Look for the valid values in the box above.

Returns Returns the list of *`weebapi.data_objects.ImageType`* objects.

1.3 Data objects

Here is the collection of all data objects that this API wrapper uses.

1.3.1 Image

class weebapi.data_objects.**Image** (*snowflake: str*, *image_type: str*, *base_type: str*, *nsfw: bool*, *file_type: str*, *mime_type: str*, *tags: list*, *url: str*, *hidden: bool*, *account: str*, *client*, *source: str = ''*)

Represents an image from Weeb.sh.

snowflake
`str` – The image ID.

image_type
`ImageType` – Type of the image.

base_type
`str` – Base type of the image.

nsfw
`bool` – Is image NSFW (not safe for work).

file_type
`str` – Image file type.

mime_type
`str` – Image MIME type.

hidden
`bool` – Is image private.

account
`str` – ID of the poster.

source
Optional[`str`] – Source of the image.

tags
`list` – List of *Tag* objects.

url
`str` – CDN URL of the image.

add_tags(*tags: list*)
This function is a *coroutine*.

Adds tags to this image.

Parameters **tags** (`list`) – List of strings.

Raises Forbidden: – If required permissions are absent.

delete()
This function is a *coroutine*.

Removes current image.

Raises Forbidden: – If required permissions are absent.

remove_tags(*tags: list*)
This function is a *coroutine*.

Removes tags from this image.

Parameters **tags** (`list`) – List of strings.

Raises Forbidden: – If required permissions are absent.

1.3.2 Preview

class weebapi.data_objects.**Preview**(*snowflake: str, url: str, file_type: str, base_type: str, image_type: str, client*)
This class represents a preview object.

str()
By itself it returns the image URL.

snowflake
`str` – Unique image identifier.

url
`str` – URL of the preview image.

image_type
ImageType – *ImageType* object.

base_type
`str` – Image base type.

file_type
`str` – Image file type.

1.3.3 ImageType

class weebapi.data_objects.**ImageType**(*name: str, client*)
This object represents an image type. By itself returns the name.

str()
Returns the name of the image type.

name

str – Name of the image type.

get_preview

This function is a *coroutine*.

Returns a *Preview* object of the image type.

Returns *Preview*

1.3.4 Tag

class weebapi.data_objects.**Tag**(*tag: dict*, *client*)

This class represents a Tag object.

str()

Returns name of the tag.

__get__

By it self it returns the name of the tag.

is_nsfw

This function is a *coroutine*.

Checks if the tag is NSFW (not safe for work).

Returns bool

CHAPTER 2

Image generation

This module represents the image generation interface.

Note: All images retrieved with this module are placed in system default temporary folder. If the system doesn't have a temporary folder images are placed in a current directory. **Please note that you should practice file deletion after usage.**

2.1 Image file object

To simplify file manipulation here is a *weebapi.data_objects.ImageFile* object. This also integrates nicely with discord.py

Note: By accessing the `discord_file` property you can get `discord.File` object.

class weebapi.data_objects.**ImageFile**(*file_path: str*)

Represents a image file object usually retrieved from image generation.

file_path

`str` – Returns file path.

discord_file

`discord.File` – If discord.py is installed it returns a `discord.File` object, else raises an exception.

Raises *weebapi.errors.DiscordPyNotInstalled* – Raised if discord.py is not installed, but `discord.File` object is requested.

delete()

This function deletes the file.

2.2 Generators

Here is the list and the description of all available generators.

Note: Thus this is represented as a separate class, it's loaded automatically into the `weebapi.Client`. If we assume that your client object is `weeb = Client(token="SomeToken123")` then you access simple image generator like this: `generator = await weeb.img_gen.get_simple("won")`.

class `weebapi.img_gen.`**`ImgGen`**(*client*)

`discord_status`(*status: str = 'online'*, *avatar: str = None*) → weebapi.data_objects.ImageFile

This function is a *coroutine*.

Generates a discord status image.

Available types:

- `online`
- `idle`
- `dnd`
- `streaming`
- `offline`

Parameters

- **`status`** (`str`) – Type of the image. See available types above.
- **`avatar`** (`str`) – URL of the avatar.

Returns *`weebapi.data_objects.ImageFile`*

`get_simple`(*img_type: str*, *face_color: str = None*, *hair_color: str = None*) → weebapi.data_objects.ImageFile

This function is a *coroutine*.

Generates simple image.

Available types:

- `awooo`
- `eyes`
- `won`

Parameters

- **`img_type`** (`str`) – Type of the image. See available types above.
- **`face_color`** (`str`) – Only for `awooo` type. HEX string for color.
- **`hair_color`** (`str`) – Only for `awooo` type. HEX string for color.

Returns *`weebapi.data_objects.ImageFile`*

license(*title: str*, *avatar: str*, *badges: list = []*, *widgets: list = []*) → weebapi.data_objects.ImageFile

This function is a *coroutine*.

Generates a spook license.

Parameters

- **title** (str) – Title of the card.
- **avatar** (str) – URL of the avatar.
- **badges** (list Optional) – List of string URLs of the images. Max 3.
- **widgets** (list Optional) – List of strings. Max 3.

Returns *weebapi.data_objects.ImageFile*

love(*avatar1: str*, *avatar2: str*) → weebapi.data_objects.ImageFile

This function is a *coroutine*.

Generates a love image.

Parameters

- **avatar1** (str) – URL of the avatar displayed on the left.
- **avatar2** (str) – URL of the avatar displayed on the right.

Returns *weebapi.data_objects.ImageFile*

waifu_insult(*avatar: str*) → weebapi.data_objects.ImageFile

This function is a *coroutine*.

Generates a waifu insult.

Parameters **avatar** (str) – URL of the avatar.

Returns *weebapi.data_objects.ImageFile*

CHAPTER 3

Errors

exception weebapi.errors.**RequireFormatting**
Bases: Exception

exception weebapi.errors.**MissingRequiredArguments**
Bases: Exception

exception weebapi.errors.**WeirdResponse**
Bases: Exception

exception weebapi.errors.**SpecifyClient**
Bases: Exception

exception weebapi.errors.**Forbidden**
Bases: Exception

exception weebapi.errors.**DiscordPyNotInstalled**
Bases: Exception

exception weebapi.errors.**NotFound**
Bases: Exception

CHAPTER 4

Indices and tables

- genindex
- modindex
- search

Python Module Index

Index