
YARSAW

Release 2.1.0

Bruce

Mar 27, 2022

DOCUMENTATION

1	Client	1
2	Utils	5
3	Frequently Asked Questions	7
4	Changelog	9
5	Overview	11
6	Installation	13
7	Getting Started	15
	Index	17

CLIENT

This is the main async client class you will use to make requests to the API. All methods are available here. You can find helper functions for some of these methods in the [utils page](#).

You may create an instance of this class and use it to make requests.

class yarsaw.**Client**(*authorization, key*)

Represents a client object used to interact with the Random Stuff API.

Parameters

- **authorization** (*str*) – Your API Key for the Random Stuff API used to authenticate your requests.
- **key** (*str*) – Your RapidAPI-Key for the Random Stuff API used to authenticate your requests.

async **get_ai_response**(*message, **kwargs*)

Gets AI responses from the API.

Parameters

- **message** (*str*) – The message you want to get the AI response for.
- **id** (Optional[Union[*str*, *int*]]) – Assign an unique ID for customized response for each user.
- **bot_name** (Optional[*str*]) – Set a name for the AI replying to your message.
- **bot_gender** (Optional[*str*]) – Set a gender for the AI replying to your message.
- **bot_master** (Optional[*str*]) – The creator/master of the AI replying to your message.
- **bot_age** (Optional[*str*]) – The age of the AI replying to your message.
- **bot_company** (Optional[*str*]) – The company that owns the AI replying to your message.
- **bot_location** (Optional[*str*]) – The location of the AI replying to your message.
- **bot_email** (Optional[*str*]) – The email of the AI replying to your message.
- **bot_build** (Optional[*str*]) – The build of the AI replying to your message.
- **bot_birth_year** (Optional[*str*]) – The birth year of the AI replying to your message.
- **bot_birth_date** (Optional[*str*]) – The birth date of the AI replying to your message.
- **bot_birth_place** (Optional[*str*]) – The birth place of the AI replying to your message.
- **bot_favorite_color** (Optional[*str*]) – The favorite color of the AI replying to your message.

- **bot_favorite_book** (Optional[[str](#)]) – The favorite book of the AI replying to your message.
- **bot_favorite_band** (Optional[[str](#)]) – The favorite band of the AI replying to your message.
- **bot_favorite_artist** (Optional[[str](#)]) – The favorite artist of the AI replying to your message.
- **bot_favorite_actress** (Optional[[str](#)]) – The favorite actress of the AI replying to your message.
- **bot_favorite_actor** (Optional[[str](#)]) – The favorite actor of the AI replying to your message.

Returns An object containing the AI response and its details.

Return type [AIResponse](#)

async get_animal_image(*animal, amount=1*)

Gets animal images from the API.

Parameters

- **animal** ([str](#)) – The animal you want to get images for. Supported Animals: Dog, Cat, Wolf, Fox
- **amount** (Optional[[int](#)]) – The amount of images you want to get.

Returns An object containing the image.

Return type [Image](#)

async get_anime_gif(*query, channel*)

Get anime gifs from the API.

Parameters

- **query** ([str](#)) – The query you want to get gifs for.
- **channel** ([str](#)) – The channel you want to get gifs from, could be 1, 2 or 3.

Returns A list of AnimeGIF objects.

Return type [list](#)

async canvas(*method, save_to=None, txt=None, text=None, img1=None, img2=None, img3=None*)

Edit Images with the API.

Parameters

- **method** ([str](#)) – The method to be used to edit the image.

Allowed Methods:

- Method(s) in which only 1 image is required: `affect`, `beautiful`, `wanted`, `delete`, `trigger`, `facepalm`, `blur`, `hitler`, `kiss`, `jail`, `invert`, `jokeOverHead`
- Method(s) in which 2 images are required: `bed`, `fuse`, `kiss`, `slap`, `spank`
- Method(s) in which 3 images are required: `distracted`
- Method(s) in which only Text is required: `changemymind`
- **save_to** (Optional[[str](#)]) – The path to save the edited image to. If not specified, the edited image will be returned as bytes.
- **txt** (Optional[[str](#)]) – The text required for your method.

- **text** (Optional[[str](#)]) – The text required for your method. Alias of txt.
- **img1** (Optional[[str](#)]) – The path/link to the first image.
- **img2** (Optional[[str](#)]) – The path/link to the second image.
- **img3** (Optional[[str](#)]) – The path/link to the third image.

Returns If save_to is not specified, the edited image will be returned as a Response object containing the base64 encoded image. If save_to is specified, the edited image will be saved to the specified path, and will 200.

Return type Union[[CanvasResponse](#), int]

async get_joke(tags=None, blacklist=None)

Get random jokes from the API.

Parameters

- **tags** (Optional[[list](#)]) – A list of tags to filter the jokes by.
- **blacklist** (Optional[[list](#)]) – Blacklisted tags to filter the jokes by. Blacklist “dirty” if you want safe jokes.

Returns A Joke object containing the joke and the tags.

Return type [Joke](#)

async fetch_subreddit_post(subreddit, search_type='hot')

Fetches a random post from a subreddit.

Parameters

- **subreddit** ([str](#)) – The subreddit to fetch a post from.
- **search_type** (Optional[[str](#)]) – This is how it sorts the posts. Allows: “hot”, “new”, “rising”, “top”

Returns An object containing the post and its details.

Return type [RedditPost](#)

async fetch_post(subreddit, search_type='hot')

Fetches a random post from a subreddit. This is an alias of [fetch_subreddit_post\(\)](#).

Parameters

- **subreddit** ([str](#)) – The subreddit to fetch a post from.
- **search_type** (Optional[[str](#)]) – This is how it sorts the posts. Allows: “hot”, “new”, “rising”, “top”

Returns An object containing the post and its details.

Return type [RedditPost](#)

async random_meme(search_type='hot')

Gets a random meme from reddit.

Parameters **search_type** (Optional[[str](#)]) – This is how it sorts the posts. Allows: “hot”, “new”, “rising”, “top”

Returns An object containing the post and its details.

Return type [RedditPost](#)

async fetch_random_post(search_type='hot')

Fetches a random post from reddit.

Parameters **search_type** (Optional[[str](#)]) – This is how it sorts the posts. Allows: “hot”, “new”, “rising”, “top”

Returns An object containing the post and its details.

Return type [RedditPost](#)

async fetch_post_by_id(*post_id*, *search_type*='hot')

Fetch a reddit post by its ID.

Parameters

- **post_id** ([str](#)) – The ID of the post to fetch.
- **search_type** (Optional[[str](#)]) – This is how it sorts the posts. Allows: “hot”, “new”, “rising”, “top”

Returns An object containing the post and its details.

Return type [RedditPost](#)

async get_weather(*city*)

Gets the weather for a city.

Parameters **city** ([str](#)) – The city to get the weather for.

Returns A list containing the weather details.

Return type [list](#)

async get_fact(*fact_type*='all')

Fetches a random fact from the API. PREMIUM ENDPOINT.

Parameters **fact_type** (Optional[[str](#)]) – The type of fact you want to fetch.

Returns An object containing the fact.

Return type [Fact](#)

async disconnect()

Closes the Client Session

async reconnect()

Restarts the Client Connection

Documentation Last Updated on Mar 27, 2022

2.1 Helper Functions

Support functions for the *Client* methods.

`yarsaw.format_joke()`

Support function for *Client.get_joke()*. Auto-format a joke. If its a single type of joke, it returns the joke itself. If its a two-part joke, it returns the setup and delivery, separated by a newline or a character you choose.

Parameters

- **joke** (*Joke*) – The joke to format.
- **format_as** (Optional[*str*]) – The format to use. Defaults to "{setup}\n{delivery}".

Returns The formatted joke.

Return type *str*

`yarsaw.generate_uid()`

Support function for *Client.get_ai_response()*. Generates a random string of characters to be used as a unique identifier for a user.

Parameters

- **chars** (Optional[*int*]) – The number of characters to generate. Defaults to 8.
- **letters** (Optional[*bool*]) – Whether or not to include letters. Defaults to True.
- **special_chars** (Optional[*bool*]) – Whether or not to include special characters. Defaults to False.

Returns The generated UID.

Return type *str*

2.2 Data Classes

```
class yarsaw.APIInfo(requests_limit: int, requests_remaining: int, reset_time: int)
```

```
class yarsaw.BotDetails(bot_name: str, bot_master: str, bot_age: str, bot_location: str, bot_company: str,  
                        bot_birth_year: str, bot_birth_date: str, bot_birth_place: str)
```

```
class yarsaw.AIResponse(response: str, bot_details: yarsaw.data_classes.BotDetails, api_info:  
                        yarsaw.data_classes.APIInfo)
```

```
class yarsaw.Joke(joke: str, tags: list, api_info: yarsaw.data_classes.APIInfo)
```

```
class yarsaw.RedditPost(id: str, type: str, title: str, author: str, url: str, image: str, gallery: bool, text: str,
                        thumbnail: str, subreddit: str, nsfw: bool, spoiler: bool, created_at: int, upvotes: int,
                        downvotes: int, upvote_ratio: float, api_info: yarsaw.data_classes.APIInfo)

class yarsaw.AnimalImage(images: list, api_info: yarsaw.data_classes.APIInfo)

class yarsaw.AnimeGIF(title: str, thumbnail: str, image: str, api_info: yarsaw.data_classes.APIInfo)

class yarsaw.CanvasResponse(base64: str, decoded_base64: str, api_info: yarsaw.data_classes.APIInfo)

class yarsaw.Fact(fact: str, api_info: yarsaw.data_classes.APIInfo)
```

2.3 Exceptions

```
exception yarsaw.InvalidPlan
exception yarsaw.InvalidAPIKey
exception yarsaw.RateLimited
exception yarsaw.BadGateway
```

Documentation Last Updated on Mar 27, 2022

FREQUENTLY ASKED QUESTIONS

3.1 How do I use this?

1. Install the package

```
python3 -m pip install yarsaw
```

2. Register to get an API Key at the [Random Stuff API resgistration page](#). This is used for authentication.
3. Register at [RapidAPI](#) for a RapidAPI Key and Account, and subscribe to the Random Stuff API. This is used to make requests to the Random Stuff API and keep track of them. You can go to [The RapidAPI Developer Dashboard](#) after logging in, select an application, head over to security, and copy its key. This is your RapidAPI Key.
4. Import the module and create an instance of Client

```
import yarsaw

client = yarsaw.Client("your_rsa_api_key", "your_rapidapi_key")
```

5. Use the client to get a random joke (to get familiar with the module)

```
import yarsaw
import asyncio # builtin, used for asynchronous calls

client = yarsaw.Client("your_api_key", "your_rapidapi_key")

async def joke():
    joke = await client.get_joke() # get the joke in form of a dict
    formatted_joke = yarsaw.Utills().format_joke(joke) # format the joke (optional)
    print(formatted_joke) # print the joke

asyncio.get_event_loop().run_until_complete(joke()) # run the joke() function
```

Now just start reading the documentation!

3.2 Why does Random Stuff API Require two keys?

The developer of the Random Stuff API decided to switch to the RapidAPI servers because they are more reliable and faster. The RapidAPI servers require a RapidAPI Key and an Account. The Random API key is used for authentication and the RapidAPI key is used to make requests to the Random Stuff API. Now, several issues were fixed, such as:

- Premium plans were only applicable to the premium endpoints. This meant that even if you had bought premium, you will still only be able to post x requests to the free endpoints instead of xxx requests.
- The API had bad error messages.
- The API was slow.
- The API did not have many examples available.

3.3 Does this wrapper still work?

YES! Unlike most other wrappers, YARSAW still works! Although the API was shifted to RapidAPI and had tons of changes, the wrapper was updated accordingly to accomodate the changes.

3.4 I have an issue

If its an issue with the wrapper, please open an issue at [GitHub](#). If there's a question about the API itself, please join the [PGamerX Studio Discord Server](#) and ask there.

3.5 Who is behind this wrapper?

YARSAW is developed by [BruceDev](#). The API itself is developed by [PGamerX](#).

Documentation Last Updated on Mar 27, 2022

CHANGELOG

4.1 2.1

- **Documentation update.**
 - Change theme to furo.
- **Edit dataclasses.**
 - Dataclasses now have more pythonic parameters for ease of use.
 - Replaced headers with APIInfo.
- Updated anime and joke methods to accomadate the latest API Changes.
- Most methods now return dataclass objects.
- Remove nested return type from `get_animal_image()` and `get_anime_gif()`
- Various changes.

4.2 2.0.1

- Added support for `/weather`
- Parameter `base` was removed from class `Client`
- Error handling for incorrect API keys was improved

4.3 2.0

This is a major change. The Random Stuff API was completely rewritten, and so was this module. Aside from new functions,

- **Docs were updated**
 - Uses the ReadTheDocs theme for documentation
 - The documentation is no longer a single page, but a collection of pages.
- Since the API now has more use of headers than ever (it returns the number of daily requests left), all methods return headers along with other data.
- The `generate_uid` and `format_joke` methods are no longer async - them being async was useless.
- New Docstrings and comments

- `RawClient` was terminated.
- A lot more updates

4.4 1.3

- The `Client` class now returns objects which are easier to work with.
- `RawClient` class can be used to return the raw JSON response from the API.

4.5 1.2

- **Updated Docs**
 - Uses `ReadTheDocs`.
 - Uses `Sphinx` instead of `MKDocs`
 - More readable.
 - Is updated automatically with docstrings.
 - Covers all the methods. Fixed minor mistakes.
- Created a `Utils` class, added `format_joke()` and `generate_uid()` to the `Utils`.
- Added docstrings to all the methods.
- Better code style (`black`).
- Updated many other things.

4.6 1.0

- Updated docs
- Renamed functions in `Client`
- Added new function `generate_uid()`
- Many more bug fixes and changes.

YARSAW is an open source, free and easy to use API Wrapper for the [Random Stuff API](#).

OVERVIEW

Features:

- Wraps all of the [Random Stuff API](#)
- Async-ready
- Easy to use
- Saves you a lot of time

INSTALLATION

To install the latest stable version of YARSAW, run the following command:

```
python -m pip install yarsaw
```

To install a specific version of YARSAW, run the following command:

```
python -m pip install yarsaw==<version>
```

To install the beta version of YARSAW, run the following command:

```
python -m pip install git+https://github.com/BruceCodesGithub/yarsaw --upgrade
```


GETTING STARTED

7.1 Get your API Keys

1. Register to get an API Key at the [Random Stuff API resgistration page](#). This is used for authentication.
2. Register at [RapidAPI](#) for a RapidAPI Key and Account, and subscribe to the Random Stuff API. This is used to make requests to the Random Stuff API and keep track of them. You can go to [The RapidAPI Developer Dashboard](#) after logging in, select an application, head over to security, and copy its key. This is your RapidAPI Key.

7.2 Examples

```
import yarsaw
import asyncio # builtin, used for asynchronous calls

client = yarsaw.Client("your_api_key", "your_rapidapi_key")

async def joke():
    joke = await client.get_joke() # get the joke in form of a dict
    formatted_joke = yarsaw.Utils().format_joke(joke) # format the joke (optional)
    print(formatted_joke) # print the joke

asyncio.get_event_loop().run_until_complete(joke()) # run the joke() function
```

Now just start reading the documentation!

Tip: The [Client](#) page contains all of the methods you can use to interact with the Random Stuff API, so we recommend reading that first.

Documentation Last Updated on Mar 27, 2022

INDEX

A

`AIResponse` (class in yarsaw), 5
`AnimalImage` (class in yarsaw), 6
`AnimeGIF` (class in yarsaw), 6
`APIInfo` (class in yarsaw), 5

B

`BadGateway`, 6
`BotDetails` (class in yarsaw), 5

C

`canvas()` (yarsaw.Client method), 2
`CanvasResponse` (class in yarsaw), 6
`Client` (class in yarsaw), 1

D

`disconnect()` (yarsaw.Client method), 4

F

`Fact` (class in yarsaw), 6
`fetch_post()` (yarsaw.Client method), 3
`fetch_post_by_id()` (yarsaw.Client method), 4
`fetch_random_post()` (yarsaw.Client method), 3
`fetch_subreddit_post()` (yarsaw.Client method), 3
`format_joke()` (in module yarsaw), 5

G

`generate_uid()` (in module yarsaw), 5
`get_ai_response()` (yarsaw.Client method), 1
`get_animal_image()` (yarsaw.Client method), 2
`get_anime_gif()` (yarsaw.Client method), 2
`get_fact()` (yarsaw.Client method), 4
`get_joke()` (yarsaw.Client method), 3
`get_weather()` (yarsaw.Client method), 4

I

`InvalidAPIKey`, 6
`InvalidPlan`, 6

J

`Joke` (class in yarsaw), 5

R

`random_meme()` (yarsaw.Client method), 3
`RateLimited`, 6
`reconnect()` (yarsaw.Client method), 4
`RedditPost` (class in yarsaw), 5