
dhooks Documentation

Release 1.1.0

kyb3r, fourjr

Aug 09, 2020

Table of Contents

1	Features	3
2	Contents	5
2.1	API Reference	5
2.1.1	Webhook	5
2.1.2	File	8
2.1.3	Embed	8
3	Indices	11
	Index	13

This **simple** library enables you to easily interact with discord webhooks, allowing you to easily format discord messages and discord embeds, retrieve webhook information, modify and delete webhooks. Asynchronous usage is also supported.

CHAPTER 1

Features

- Use the same client for sync/async operations.
- Implements the entire webhook API.
- Easy to use with an object oriented design.

2.1 API Reference

The following section outlines the API of dhooks.

2.1.1 Webhook

```
class dhooks.Webhook (url: str = "", session: Union[aiohttp.client.ClientSession, requests.sessions.Session, None] = None, is_async: bool = False, **options)
```

Class that represents a Discord webhook.

Parameters

- **url** (*str*, *optional*) – The webhook URL that the client will send requests to.

Note: the URL should contain the *id* and *token* of the webhook in the form of:

```
https://discord.com/api/webhooks/webhooks/{id}/{token}
```

Warning: If you don't provide *url*, you must provide both *id* and *token* keyword arguments.

- **session** (*requests.Session* or *aiohttp.ClientSession*, *optional*) – The HTTP session that will be used to make requests to the API. If *session* is not provided, a new *requests.Session* or *aiohttp.ClientSession* will be created, depending on *is_async*.
- **is_async** (*bool*, *optional*) – Defaults to `False`. Whether or not the API methods in the class should be asynchronous. If set to `True`, all methods will have the same interfaces, but returns a coroutine.
- ****id** (*int*, *optional*) – The Discord ID of the webhook. If not provided, it will be extracted from the webhook URL.

- ****token** (*str*, *optional*) – The Discord token of the webhook. If not provided, it will be extracted from the webhook URL.
- ****username** (*str*, *optional*) – The username that will override the default name of the webhook every time you send a message.
- ****avatar_url** (*str*, *optional*) – The URL of the avatar that will override the default avatar of the webhook every time you send a message.

id

The Discord ID of the webhook.

Type `int`

token

The Discord token of the webhook.

Type `str`

url

The webhook URL that the client will send requests to.

Type `str`

username

The username that will override the default name of the webhook every time you send a message. If *username* is not provided, the default name is used.

Type `str`

avatar_url

The avatar URL that will override the default avatar of the webhook every time you send a message. If *avatar_url* is not provided, the default avatar is used.

Type `str`

is_async

Whether or not to the API methods in the class should be asynchronous. If set to `True`, all methods will have the same interfaces, but returns a coroutine.

Type `bool`

session

The HTTP session that will be used to make requests to the API. *session* will be a `requests.Session` or `aiohttp.ClientSession` depending on *is_async*.

Type `requests.Session` or `aiohttp.ClientSession`

default_name

Warning: In order for some attributes to be filled, *get_info()* must be called prior.

The default name of the webhook, this can be changed via *modify()* or directly through discord server settings.

Type `str`

default_avatar

The *avatar string* of the webhook.

Type `str`

guild_id

Defaults to -1. The id of the webhook's guild.

Type `int`

channel_id

Defaults to -1. The id of the channel the webhook sends messages to.

Type `int`

classmethod Async (*url: str = "", session: Optional[aiohttp.client.ClientSession] = None, **options*) → `dhooks.client.Webhook`

Returns a new instance of `Webhook` with `is_async` set to `True`.

Equivalent to:

```
Webhook(url, session=session, is_async=True, **options)
```

delete () → `None`

Deletes the `Webhook` permanently.

edit (*name: str = "", avatar: bytes = b""*) → `dhooks.client.Webhook`

Alias for `modify()`.

execute (*content: str = "", embed: Optional[dhooks.embed.Embed] = None, embeds: Optional[List[dhooks.embed.Embed]] = None, file: Optional[dhooks.file.File] = None, username: str = "", avatar_url: str = "", tts: bool = False*) → `dhooks.client.Webhook`

Alias for `send()`.

get_info () → `dhooks.client.Webhook`

Updates `Webhook` with fresh data retrieved from discord.

The following attributes are retrieved and updated:

- `default_avatar`
- `default_name`
- `guild_id`
- `channel_id`

modify (*name: str = "", avatar: bytes = b""*) → `dhooks.client.Webhook`

Edits the webhook.

Parameters

- **name** (*str, optional*) – The new default name of the webhook.
- **avatar** (*bytes, optional*) – The new default avatar that webhook will be set to.

send (*content: str = "", embed: Optional[dhooks.embed.Embed] = None, embeds: Optional[List[dhooks.embed.Embed]] = None, file: Optional[dhooks.file.File] = None, username: str = "", avatar_url: str = "", tts: bool = False*) → `dhooks.client.Webhook`

Sends a message to discord through the webhook.

Parameters

- **content** (*str, optional*) – The message contents (up to 2000 characters)
- **embed** (*Embed, optional*) – Single embedded rich content.
- **embeds** (*List[Embed], optional*) – List of embedded rich content.
- **file** (*File, optional*) – The file that will be uploaded.

- **tts** (*bool*, *optional*) – Defaults to `False`. Whether or not the message will use text-to-speech.
- **username** (*str*, *optional*) – Defaults to `username`. Override the default username of the webhook.
- **avatar_url** (*str*, *optional*) – Defaults to `avatar_url`. Override the default avatar of the webhook.

2.1.2 File

class `dhooks.File` (*fp*: `Union[BinaryIO, str]`, *name*: `str` = "")

Data class that represents a file that can be sent to discord.

Parameters

- **fp** (*str* or `io.BytesIO`) – A file path or a binary stream that is the file. If a file path is provided, this class will open and close the file for you.
- **name** (*str*, *optional*) – The name of the file that discord will use, if not provided, defaults to the file name or the binary stream's name.

close (*force*=`False`) → `None`

Closes the file if the file was opened by `File`, if not, this does nothing.

Parameters **force** (*bool*) – If set to `True`, force close every file.

seek (*offset*: `int` = 0, **args*, ***kwargs*)

A shortcut to `self.fp.seek`.

2.1.3 Embed

class `dhooks.Embed` (***kwargs*)

Class that represents a discord embed.

Parameters

- ****title** (*str*, *optional*) – Defaults to `None`. The title of the embed.
- ****description** (*str*, *optional*) – Defaults to `None`. The description of the embed.
- ****url** (*str*, *optional*) – URL of the embed. It requires `title` to be set.
- ****timestamp** (*str*, *optional*) – ISO 8601 timestamp of the embed. If set to a “now”, the current time is set as the timestamp.
- ****color** (*int* (or *hex*), *optional*) – Color of the embed.
- ****image_url** (*str*, *optional*) – URL of the image.
- ****thumbnail_url** (*str*, *optional*) – URL of the thumbnail.

add_field (*name*: `str`, *value*: `str`, *inline*: `bool` = `True`) → `None`

Adds an embed field.

Parameters

- **name** (*str*) – Name attribute of the embed field.
- **value** (*str*) – Value attribute of the embed field.
- **inline** (*bool*) – Defaults to `True`. Whether or not the embed should be inline.

del_field (*index: int*) → None

Deletes a field by index.

Parameters **index** (*int*) – Index of the field to delete.

set_author (*name: str, icon_url: str = None, url: str = None*) → None

Sets the author of the embed.

Parameters

- **name** (*str*) – The author’s name.
- **icon_url** (*str, optional*) – URL for the author’s icon.
- **url** (*str, optional*) – URL hyperlink for the author.

set_footer (*text: str, icon_url: str = None*) → None

Sets the footer of the embed.

Parameters

- **text** (*str*) – The footer text.
- **icon_url** (*str, optional*) – URL for the icon in the footer.

set_image (*url: str*) → None

Sets the image of the embed.

Parameters **url** (*str*) – URL of the image.

set_thumbnail (*url: str*) → None

Sets the thumbnail of the embed.

Parameters **url** (*str*) – URL of the thumbnail.

set_timestamp (*time: Union[str, datetime.datetime] = None, now: bool = False*) → None

Sets the timestamp of the embed.

Parameters

- **time** (*str or datetime.datetime*) – The ISO 8601 timestamp from the embed.
- **now** (*bool*) – Defaults to False. If set to True the current time is used for the timestamp.

set_title (*title: str, url: str = None*) → None

Sets the title of the embed.

Parameters

- **title** (*str*) – Title of the embed.
- **url** (*str or None, optional*) – URL hyperlink of the title.

to_dict () → dict

Turns the *Embed* object into a dictionary.

CHAPTER 3

Indices

- `genindex`
- `search`

A

`add_field()` (*dhooks.Embed method*), 8
`Async()` (*dhooks.Webhook class method*), 7
`avatar_url` (*dhooks.Webhook attribute*), 6

C

`channel_id` (*dhooks.Webhook attribute*), 7
`close()` (*dhooks.File method*), 8

D

`default_avatar` (*dhooks.Webhook attribute*), 6
`default_name` (*dhooks.Webhook attribute*), 6
`del_field()` (*dhooks.Embed method*), 8
`delete()` (*dhooks.Webhook method*), 7

E

`edit()` (*dhooks.Webhook method*), 7
`Embed` (*class in dhooks*), 8
`execute()` (*dhooks.Webhook method*), 7

F

`File` (*class in dhooks*), 8

G

`get_info()` (*dhooks.Webhook method*), 7
`guild_id` (*dhooks.Webhook attribute*), 6

I

`id` (*dhooks.Webhook attribute*), 6
`is_async` (*dhooks.Webhook attribute*), 6

M

`modify()` (*dhooks.Webhook method*), 7

S

`seek()` (*dhooks.File method*), 8
`send()` (*dhooks.Webhook method*), 7
`session` (*dhooks.Webhook attribute*), 6

`set_author()` (*dhooks.Embed method*), 9
`set_footer()` (*dhooks.Embed method*), 9
`set_image()` (*dhooks.Embed method*), 9
`set_thumbnail()` (*dhooks.Embed method*), 9
`set_timestamp()` (*dhooks.Embed method*), 9
`set_title()` (*dhooks.Embed method*), 9

T

`to_dict()` (*dhooks.Embed method*), 9
`token` (*dhooks.Webhook attribute*), 6

U

`url` (*dhooks.Webhook attribute*), 6
`username` (*dhooks.Webhook attribute*), 6

W

`Webhook` (*class in dhooks*), 5