
discordwebhook Documentation

Release latest

Aug 06, 2022

Contents

1	Setup	3
1.1	Installation:	3
1.2	Import	3
2	API Reference	5
2.1	discordwebhook.Webhook	5
2.2	discordwebhook.WebhookMessage	8
2.3	discordwebhook.File	9
2.4	discordwebhook.Embed	10
3	Examples	13
3.1	Synchronous (Basic) Example	13
3.2	Asynchronous Example	14
3.3	Fetch example	14
4	Version History	17
4.1	1.1.2- 6th August 2022	17
4.2	1.1.0 - 5th Feb 2022	17
4.3	1.0.2 - 30th January 2022	17
4.4	1.0.1 - 30th January 2022	17
4.5	0.1.2 - 29th July 2020	17
4.6	0.1.1 - 29th July 2020	18
4.7	0.1.0 - 29th July 2020	18
4.8	0.0.8 - 28th July 2020	18
4.9	0.0.7 - 28th July 2020	18
4.10	0.0.6 - 28th July 2020	18
4.11	0.0.5 - 28th July 2020	18
4.12	0.0.4 - 28th July 2020	19
4.13	0.0.3 - 28th July 2020	19
4.14	0.0.2 - 28th July 2020	19
4.15	0.0.1 - 28th July 2020	19

[PyPi page](#) | [GitHub page](#)

discordwebhook.py is a Python Library for easily using discord webhooks in python

It is a lightweight, sync and async ready wrapper for Discord API Webhooks.

Main features:

- Modern
- Asynchronous and synchronous support
- In-built embed support - no need for another package such as discord.py
- Full coverage of the webhook API. Read, Modify, Delete and Execute webhooks (and delete/edit/fetch messages)
- Easy to use, fully documented
- Rate limit handling

1.1 Installation:

```
pip install discordwebhook.py
```

Or download it from the [PyPi page](#)

1.2 Import

```
import discordwebhook
```


2.1 discordwebhook.Webhook

Table 1: discordwebhook.Webhook

Parameters	Methods
url : Optional[str]	.send_sync
	.send_async
	.modify_sync
	.modify_async
	.delete_sync
	.delete_async
	.fetch_message_sync
	.fetch_message_async

2.1.1 Parameters

url : Optional[str] = None - An optional URL to overwrite the Webhook object

Webhook object

All synchronous and async webhook usage functions are in this class

.fetch_data_sync(url = None)

2.1.2 Parameters

url : Optional[str] = None - An optional URL to overwrite the Webhook object

Returns **discordwebhook.Webhook** and updates current Webhook object

`await .fetch_data_async(url = None)`

2.1.3 Parameters

url : Optional[str] = None - An optional URL to overwrite the Webhook object

Returns **discordwebhook.Webhook** and updates current Webhook object

`.send_sync(content = None, url = None, username = None, avatar_url = None, is_tts = False, embed = None, embeds = None, allowed_mentions = allowedmentions.AllowedMentions(), file = None, files = [])`

2.1.4 Parameters

content : Optional[str] = None - The content to add to the message

url : Optional[str] = None - The url to send the message to. Overwrites Webhook.url if url already exists

username : Optional[str] = None - Username to overwrite the webhook name

avatar_url : Optional[str] = None - The avatar url to overwrite the icon

is_tts : Optional[bool] = False - Whether the message sends with TTS enabled

embed : Optional[discordwebhook.Embed] - Singular embed to add to the message

embeds : Optional[List[discordwebhook.Embed]] - Optional list of embeds

allowed_mentions : Optional[discordwebhook.AllowedMentions] - The allowed mentions for the message

file : Optional[discordwebhook.File] = None - A single file to add to a message

files : Optional[List[discordwebhook.File]] = [] - A list of files to attach

Synchronously send a message to the webhook.

Throws error if no url is provided (through the class or method)

Returns **discordwebhook.WebhookMessage**

`await .send_async(content = None, url = None, username = None, avatar_url = None, is_tts = False, embed = None, embeds = None, allowed_mentions = allowedmentions.AllowedMentions(), file = None, files = [])`

2.1.5 Parameters

content : Optional[str] = None - The content to add to the message

url : Optional[str] = None - The url to send the message to. Overwrites Webhook.url if url already exists

username : Optional[str] = None - Username to overwrite the webhook name

avatar_url : Optional[str] = None - The avatar url to overwrite the icon

is_tts : Optional[bool] = False - Whether the message sends with TTS enabled

embed : Optional[discordwebhook.Embed] - Singular embed to add to the message

embeds : Optional[List[discordwebhook.Embed]] - Optional list of embeds

allowed_mentions : `Optional[discordwebhook.AllowedMentions]` - The allowed mentions for the message

file : `Optional[discordwebhook.File] = None` - A single file to add to a message

files : `Optional[List[discordwebhook.File]] = []` - A list of files to attach

Asynchronously send a message to the webhook.

Throws error if no url is provided (through the class or method)

Returns `discordwebhook.WebhookMessage`

`.modify_sync(name, channel_id)`

2.1.6 Parameters

name : `str` - The updated name for the Webhook

channel_id : `int` - The channel ID to move the webhook to.

Synchronously modify the name and channel of the webhook

Returns `discordwebhook.Webhook` and updates current Webhook object

`await .modify_async(name, channel_id)`

2.1.7 Parameters

name : `str` - The updated name for the Webhook

channel_id : `int` - The channel ID to move the webhook to.

Asynchronously modify the name and channel of the webhook

Returns `discordwebhook.Webhook` and updates current Webhook object

`.delete_sync()`

Synchronously delete the webhook

`await .delete_async()`

Asynchronously delete the webhook

`.fetch_message_sync(message_id)`

2.1.8 Parameters

message_id : `int` - The message ID to fetch information for

Synchronously get message data from an ID

Returns `discordwebhook.WebhookMessage`

await `.fetch_message_async(message_id)`

2.1.9 Parameters

message_id : int - The message ID to fetch information for

Asynchronously get message data from an ID

Returns **discordwebhook.WebhookMessage**

2.2 discordwebhook.WebhookMessage

Table 2: discordwebhook.WebhookMessage

Parameters	Methods
webhook : discordwebhook.Webhook	<code>.delete_sync</code>
data : dict	<code>.delete_async</code>
	<code>.edit_sync</code>
	<code>.edit_async</code>

2.2.1 Parameters

webhook : discordwebhook.Webhook - The webhook this message belongs to

data : dict - The raw message data

WebhookMessage object

Returned from **Webhook.fetch_message** and **Webhook.send**

`.delete_sync()`

Deletes the message synchronously

await `.delete_async()`

Deletes the message asynchronously

`.edit_sync(content = None, embed = None, embeds = None, allowed_mentions = discordwebhook.AllowedMentions(), file = None, files = [])`

2.2.2 Parameters

content : Optional[str] = None - The content to update the message with

embed : Optional[discordwebhook.Embed] = None - Single embed for the message

embeds : Optional[List[discordwebhook.Embed]] = None - An array of embeds

allowed_mentions : **Optional[discordwebhook.AllowedMentions]** = **discordwebhook.AllowedMentions()** - The mentions that will work with the message

file : Optional[discordwebhook.File] = None - A single file to add to a message

files : `Optional[List[discordwebhook.File]] = []` - A list of files to attach

Synchronously edit the message

Returns `discordwebhook.WebhookMessage`

await `.edit_async(content = None, embed = None, embeds = None, allowed_mentions = discordwebhook.AllowedMentions(), file = None, files = [])`

2.2.3 Parameters

content : `Optional[str] = None` - The content to update the message with

embed : `Optional[discordwebhook.Embed] = None` - Single embed for the message

embeds : `Optional[List[discordwebhook.Embed]] = None` - An array of embeds

allowed_mentions : `Optional[discordwebhook.AllowedMentions]` = `discordwebhook.AllowedMentions()` - The mentions that will work with the message

file : `Optional[discordwebhook.File] = None` - A single file to add to a message

files : `Optional[List[discordwebhook.File]] = []` - A list of files to attach

Asynchronously edit the message

Returns `discordwebhook.WebhookMessage`

2.3 discordwebhook.File

Table 3: discordwebhook.File

Parameters	Methods
<code>path</code> : <code>str</code>	<code>.open</code>
<code>fp</code> : <code>File</code>	<code>.close</code>
<code>name</code> : <code>str</code>	

2.3.1 Parameters

path : `str` - The file path to the file

fp : `File` - The file data of the file. Use instead of path

name : `str` - Custom name for the file. Required if using fp

`.open()`

Open the file from the path or fp

`.close()`

Close the file if it has been opened

2.4 discordwebhook.Embed

Table 4: discordwebhook.Embed

Parameters	Methods
title : Optional[str]	.set_footer
description : Optional[str]	.set_image
timestamp : Optional[datetime.datetime]	.set_video
color	.set_thumbnail
	.set_author
	.add_field

2.4.1 Parameters

title : Optional[str] = None - Add a title to the embed

description : Optional[str] = None - Add a description to the embed

timestamp : Optional[datetime.datetime] = None - Add a timestamp to the embed footer

color = None - Set embed color

Embed object

Create embeds in the exact same way as discord.py with this class

.set_footer(text, icon_url = None, proxy_icon_url = None)

Set the embed footer

2.4.2 Parameters

text : str - Set footer text. Required argument

icon_url : Optional[str] = None - Set footer icon url. Not required

proxy_icon_url : Optional[str] = None - Proxy icon url. Not required

.set_image(url, proxy_url = None, height = None, width = None)

Add an image to the embed

2.4.3 Parameters

url : str - Set footer text. Required argument

proxy_url : Optional[str] = None - Proxy image url. Not required

height : Optional[int] = None - Custom image height. Not required

width : Optional[int] = None - Custom image width. Not required

.set_video(url, proxy_url = None, height = None, width = None)

Add a video to the embed

2.4.4 Parameters

url : str - Set footer text. Required argument

proxy_url : Optional[str] = None - Proxy image url. Not required

height : Optional[int] = None - Custom image height. Not required

width : Optional[int] = None - Custom image width. Not required

.set_thumbnail(url, proxy_url = None, height = None, width = None)

Set embed thumbnail

2.4.5 Parameters

url : str - Set footer text. Required argument

proxy_url : Optional[str] = None - Proxy image url. Not required

height : Optional[int] = None - Custom image height. Not required

width : Optional[int] = None - Custom image width. Not required

.set_author(name, url = None, icon_url = None, proxy_icon_url = None)

Set the embed footer

2.4.6 Parameters

name : str - Set footer text. Required argument

url : Optional[str] = None - User URL. Not required

icon_url : Optional[str] = None - Set footer icon url. Not required

proxy_icon_url : Optional[str] = None - Proxy icon url. Not required

.add_field(name, value, inline = True)

Add a field to the embed. Same as discord.py

2.4.7 Parameters

name : str - Field name.

value : str - Value of the embed field. Goes under the name

inline : Optional[bool] = True - Whether the field is inline. Defaults True

Multiple different examples

3.1 Synchronous (Basic) Example

```
import discordwebhook

# Create the webhook. Parameter url="webhook_url" can be added here instead of in_
↳method .sendSync()
webhook = discordwebhook.Webhook(
    url="webhook_url"
)

# OPTIONAL - Get webhook username and avatar (sends request to discord)
webhook_data = webhook.fetch_data_sync()

# Add embed with title "Embed title", same as discord.py
embed = discordwebhook.Embed(
    title="Embed title",
)

# Add a field to the embed, exactly the same as discord.py
embed.add_field(name="Field title", value="Exact same as discord.py, however can be_
↳used synchronously", inline=False)

# Post webhook to URL synchronously
webhook.send_sync(
    f"This webhook's original username was **{webhook_data.username}**", # Webhook_
↳message content
    username="Coolo2", # Overwrite webhook username, can also be defined when class_
↳is initialized
    embed=embed # Embeds can also be set with embeds=[embed]
)
```

3.2 Asynchronous Example

```
import discordwebhook

async def doWebhook():

    # Create the webhook. Parameter url="webhook_url" can be added here instead of in_
    ↪method .sendSync()
    webhook = discordwebhook.Webhook(
        url="webhook_url"
    )

    # OPTIONAL - Get webhook username and avatar (sends request to discord)
    webhook_data = await webhook.fetch_data_async()

    # Add embed with title "Embed title", same as discord.py
    embed = discordwebhook.Embed(
        title="Embed title",
    )

    # Add a field to the embed, exactly the same as discord.py
    embed.add_field(name="Field title", value="Exact same as discord.py, however can_
    ↪be used synchronously", inline=False)

    # Post webhook to URL synchronously
    await webhook.send_async(
        f"This webhook's original username was **{webhook_data.username}**", #_
    ↪Webhook message content
        username="Coolo2", # Overwrite webhook username, can also be defined when_
    ↪class is initialized
        embed=embed # Embeds can also be set with embeds=[embed]
    )
```

3.3 Fetch example

```
import discordwebhook

webhook = discordwebhook.Webhook(
    url="webhook_url"
)

# Can be used synchronously and asynchronously with fetch_data_async. Returns current_
    ↪Webhook class
webhook.fetch_data_sync()

print(webhook.id)
print(webhook.url)

print(webhook.name)
print(webhook.icon_url)

print(webhook.channel_id)
print(webhook.guild_id)
```

More examples can be found at the [GitHub page Examples folder](#) .

CHAPTER 4

Version History

4.1 1.1.2- 6th August 2022

- Fix editing message embeds

4.2 1.1.0 - 5th Feb 2022

- Added discordwebhook.File - Attach a file to a webhook
- Added Webhook.modify - Modify a webhook
- Added Webhook.delete - Delete a webhook
- Added discordwebhook.WebhookMessage
- This allows messages to be edited and deleted. See documentation

4.3 1.0.2 - 30th January 2022

- Fixed bugs with 1.0.1

4.4 1.0.1 - 30th January 2022

- Rewritten the whole library. Too many changes to show, see documentation.

4.5 0.1.2 - 29th July 2020

- Added ability to set username and avatar_url in Webhook().send() with alias author

4.6 0.1.1 - 29th July 2020

- Fix version number issues with 0.1.0

4.7 0.1.0 - 29th July 2020

- Fixed asyncCreate not returning any values
- Added error handling for invalid token in fetching webhooks
- Added error handling for no provided url
- Added ability for setting webhook link prior to sending it, adding a link to the create.Webhook object
- Added alias *discordwebhook.use* for *discordwebhook.create* and *discordwebhook.asyncUse* for *discordwebhook.asyncCreate*
- Many changes to documentation and other things

4.8 0.0.8 - 28th July 2020

- Fixed fatal error which occurred with all embeds after 0.0.7

4.9 0.0.7 - 28th July 2020

- Added *discordwebhook.discordwebhook*
- *discordwebhook.fetch.Webhook()* now supports a *Webhook()* object instead of just string
- Changed *discordwebhook.discordwebhook.version.recent* to *current* with alias *recent*
- When using webhooks set information is stored in variables and can be accessed later by using *(webhook).setData()*, for example *webhook.message*
- Added message and tts kwarg to *.send* to allow for one line webhook sends
- Added kwargs to embed creation to create an experience identical to discord.py
- Added docstrings at the top of files
- Started error handling for embeds, checks on send to keep projects working

4.10 0.0.6 - 28th July 2020

- Fixed fatal errors with 0.0.5

4.11 0.0.5 - 28th July 2020

- Changed embed class so *embed.embed* can be replaced with just *embed* - simplifying sends
- Additions and fixes to documentation

- Documentation addition to PyPi page

4.12 0.0.4 - 28th July 2020

- Added mention permissions (if mentions will work)
- Reorganised some functions
- Created documentation - Not listed on PyPi page

4.13 0.0.3 - 28th July 2020

- Fixed fatal bugs with the previous release with asyncio

4.14 0.0.2 - 28th July 2020

- Moved to a different name
- Fixed bugs with original release

4.15 0.0.1 - 28th July 2020

- Original release on another name
- Added main features such as Webhook post and creation
- Added embeds
- Added asynchronous and synchronous functions