

---

# **dblpy Documentation**

***Release 0.3.3***

**Assanali Mukhanov**

**Jan 28, 2020**



---

## Contents

---

|          |                                |           |
|----------|--------------------------------|-----------|
| <b>1</b> | <b>What's New</b>              | <b>1</b>  |
| 1.1      | v0.3.3 . . . . .               | 1         |
| 1.2      | v0.3.2 . . . . .               | 1         |
| 1.3      | v0.3.1 . . . . .               | 1         |
| 1.4      | v0.3.0 . . . . .               | 1         |
| 1.5      | v0.2.1 . . . . .               | 2         |
| 1.6      | v0.2 . . . . .                 | 2         |
| 1.7      | v0.1.6 . . . . .               | 2         |
| 1.8      | v0.1.4 . . . . .               | 2         |
| 1.9      | v0.1.3 . . . . .               | 2         |
| 1.10     | v0.1.2 . . . . .               | 2         |
| <b>2</b> | <b>API Reference</b>           | <b>5</b>  |
| 2.1      | Version Related Info . . . . . | 5         |
| 2.2      | Client . . . . .               | 5         |
| 2.3      | Event reference . . . . .      | 8         |
| 2.4      | Exceptions . . . . .           | 9         |
| <b>3</b> | <b>Indices and tables</b>      | <b>11</b> |
|          | <b>Index</b>                   | <b>13</b> |



# CHAPTER 1

---

## What's New

---

This page keeps a detailed human friendly rendering of what's new and changed in specific versions.

### 1.1 v0.3.3

- Internal changes regarding support of top.gg migration
- Fixed errors raised when using `DBLClient.close()` without built-in webhook

### 1.2 v0.3.2

- **BREAKING CHANGE:** `Client` class has been renamed to `DBLClient`

### 1.3 v0.3.1

- Added `on_guild_post`, an event that is called when autoposter successfully posts guild count
- Renamed `get_upvote_info` to `get_bot_upvotes`
- Added `get_user_vote`

### 1.4 v0.3.0

- `DBLClient` now has `autopost` kwarg that will post server count automatically every 30 minutes
- Fixed code 403 errors
- Added `on_dbl_vote`, an event that is called when you test your webhook
- Added `on_dbl_test`, an event that is called when someone tests your webhook

## 1.5 v0.2.1

- Added webhook
- Removed support for discord.py versions lower than 1.0.0
- Made `DBLClient.get_weekend_status()` return a boolean value
- Added webhook example in README
- **BREAKING CHANGE:** Removed `post_server_count` and `get_server_count`

## 1.6 v0.2

- Added `post_guild_count`
- Made `post_server_count` an alias for `post_guild_count`
- Added `get_guild_count`
- Made `get_server_count` an alias for `get_guild_count`
- Added `DBLClient.get_weekend_status()`
- Removed all parameters from `DBLClient.get_upvote_info()`
- Added limit to `DBLClient.get_bots()`
- Fixed example in README

## 1.7 v0.1.6

- Bug fixes & improvements

## 1.8 v0.1.4

- Initial ratelimit handling

## 1.9 v0.1.3

- Added documentation
- Fixed some minor bugs

## 1.10 v0.1.2

Initial release

- Working
  - POSTing server count

- GET bot info, server count, upvote count, upvote info
  - GET all bots
  - GET specific user info
  - GET widgets (large and small) including custom ones. See [top.gg/api/docs](https://top.gg/api/docs) for more info.
- Not Working / Implemented
  - Searching for bots via the api





The following section outlines the API of dblpy.

### 2.1 Version Related Info

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

`dbl.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'.

`dbl.__version__`

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

### 2.2 Client

**class** `dbl.DBLClient` (*bot*, *token*, *\*\*kwargs*)

Represents a client connection that connects to top.gg. This class is used to interact with the DBL API.

#### Parameters

- **token** – An API Token
- **bot** – An instance of a discord.py Bot or Client object
- **\*\*loop** (*Optional[event loop]*) – The `event loop` to use for asynchronous operations. Defaults to `bot.loop`.
- **\*\*session** (*Optional*) – The `aiohttp session` to use for requests to the API.
- **\*\*autopost** (*Optional[bool]*) – Whether to automatically post bot's guild count every 30 minutes. This will dispatch `on_guild_post()`.

- **\*\*webhook\_auth** (*Optional*) – The string for Authorization you set on the site for verification.
- **\*\*webhook\_path** (*Optional*) – The path for the webhook request.
- **\*\*webhook\_port** (*Optional[int]*) – The port to run the webhook on. Will activate webhook when set.

**guild\_count** ()

Gets the guild count from the Client/Bot object

**close** ()

This function is a coroutine.

Closes all connections.

**generate\_widget\_large** (*bot\_id: int = None, top: str = '2C2F33', mid: str = '23272A', user: str = 'FFFFFF', cert: str = 'FFFFFF', data: str = 'FFFFFF', label: str = '99AAB5', highlight: str = '2C2F33'*)

This function is a coroutine.

Generates a custom large widget. Do not add # to the color codes (e.g. #FF00FF become FF00FF).

#### Parameters

- **bot\_id** (*int*) – ID of the bot you wish to make a widget for.
- **top** (*str*) – The hex color code of the top bar.
- **mid** (*str*) – The hex color code of the main section.
- **user** (*str*) – The hex color code of the username text.
- **cert** (*str*) – The hex color code of the certified text (if applicable).
- **data** (*str*) – The hex color code of the statistics (numbers only e.g. 44) of the bot.
- **label** (*str*) – The hex color code of the description (text e.g. guild count) of the statistics.
- **highlight** (*str*) – The hex color code of the data boxes.

#### Returns URL of the widget

Return type `str`

**generate\_widget\_small** (*bot\_id: int = None, avabg: str = '2C2F33', lcol: str = '23272A', rcol: str = '2C2F33', ltxt: str = 'FFFFFF', rtxt: str = 'FFFFFF'*)

This function is a coroutine.

Generates a custom large widget. Do not add # to the color codes (e.g. #FF00FF become FF00FF).

#### Parameters

- **bot\_id** (*int*) – ID of the bot you wish to make a widget for.
- **avabg** (*str*) – The hex color code of the background of the avatar (if the avatar has transparency).
- **lcol** (*str*) – The hex color code of the left background color.
- **rcol** (*str*) – The hex color code of the right background color.
- **ltxt** (*str*) – The hex color code of the left text.
- **rtxt** (*str*) – The hex color code of the right text.

#### Returns URL of the widget

**Return type** str

**get\_bot\_info** (*bot\_id: int = None*)

This function is a coroutine.

Gets information about a bot from top.gg

**Parameters** **bot\_id** (*int [Optional]*) – ID of the bot you want to lookup.

**Returns** **bot info** – Information on the bot you looked up. <https://top.gg/api/docs#bots>

**Return type** dict

**get\_bot\_upvotes** ()

This function is a coroutine.

Gets information about last 1000 votes for your bot on top.gg

---

**Note:** This API endpoint is available to the owner of the bot only.

---

**Returns** **users** – Users who upvoted your bot.

**Return type** list

**get\_bots** (*limit: int = 50, offset: int = 0, sort: str = None, search: dict = None, fields: list = None*)

This function is a coroutine.

Gets information about listed bots on top.gg

**Parameters**

- **limit** (*int [Optional]*) – The number of results you want to lookup. Defaults to 50. Max 500.
- **offset** (*int [Optional]*) – The amount of bots to skip. Defaults to 0.
- **sort** (*str [Optional]*) – The field to sort by. Prefix with – to reverse the order.
- **search** (*dict [Optional]*) – The search data.
- **fields** (*list [Optional]*) – Fields to output.

**Returns** **bots** – Returns info on the bots on DBL. <https://top.gg/api/docs#bots>

**Return type** dict

**get\_guild\_count** (*bot\_id: int = None*)

This function is a coroutine.

Gets a guild count from top.gg

**Parameters** **bot\_id** (*int [Optional]*) – ID of the bot you want to lookup. Defaults to the discord.py Bot/Client provided in Client init.

**Returns** **stats** – The guild count and shards of a bot. The date object is returned in a date-time.datetime object.

**Return type** dict

**get\_user\_info** (*user\_id: int*)

This function is a coroutine.

Gets information about a user on top.gg

**Parameters** **user\_id** (*int*) – ID of the user you wish to lookup.

**Returns user data** – Info about the user. <https://top.gg/api/docs#users>

**Return type** dict

**get\_user\_vote** (*user\_id: int*)

This function is a coroutine.

Gets information about the user's upvote for your bot on top.gg

**Parameters** **user\_id** (*int*) – ID of the user you wish to lookup.

**Returns** **vote status** – Info about the user's upvote.

**Return type** bool

**get\_weekend\_status** ()

This function is a coroutine.

Gets weekend status from top.gg

**Returns** **weekend status** – The boolean value of weekend status.

**Return type** bool

**get\_widget\_large** (*bot\_id: int = None*)

This function is a coroutine.

Generates the default large widget.

**Parameters** **bot\_id** (*int*) – ID of the bot you wish to make a widget for.

**Returns** **URL of the widget**

**Return type** str

**get\_widget\_small** (*bot\_id: int = None*)

This function is a coroutine.

Generates the default small widget.

**Parameters** **bot\_id** (*int*) – ID of the bot you wish to make a widget for.

**Returns** **URL of the widget**

**Return type** str

**post\_guild\_count** (*shard\_count: int = None, shard\_no: int = None*)

This function is a coroutine.

Posts your bot's guild count to top.gg

**Parameters**

- **shard\_count** (*int [Optional]*) – The total number of shards.
- **shard\_no** (*int [Optional]*) – The index of the current shard. DBL uses 0 based indexing for shards.

## 2.3 Event reference

**dbl.on\_guild\_post** ()

Called when guild count is posted on top.gg

**dbl.on\_dbl\_vote** (*data*)

Called when someone votes for your bot on top.gg

**Parameters** `data` – The data with vote info returned in dict object

Example:

```
@bot.event
async def on_dbl_vote(data):
    print(data)
```

The data returned can be found [here](#).

`dbl.on_dbl_test(data)`

Called when someone tests webhook system for your bot on top.gg

**Parameters** `data` – The data with test info returned in dict object

Example:

```
@bot.event
async def on_dbl_test(data):
    print(data)
```

The data returned can be found [here](#).

## 2.4 Exceptions

The following exceptions are thrown by the library.

**exception** `dbl.DBLException`

Base exception class for dblpy

Ideally speaking, this could be caught to handle any exceptions thrown from this library.

**exception** `dbl.UnauthorizedDetected`

Exception that's thrown when no API Token is provided

Subclass of `DBLException`

**exception** `dbl.ClientException`

Exception that's thrown when an operation in the `Client` fails.

These are usually for exceptions that happened due to user input.

**exception** `dbl.HTTPException(response, message)`

Exception that's thrown when an HTTP request operation fails.

**response**

The response of the failed HTTP request. This is an instance of `aiohttp.ClientResponse`.

**text**

The text of the error. Could be an empty string.

**Type** `str`

**exception** `dbl.Unauthorized(response, message)`

Exception that's thrown for when status code 401 occurs.

Subclass of `HTTPException`

**exception** `dbl.Forbidden(response, message)`

Exception that's thrown for when status code 403 occurs.

Subclass of `HTTPException`

**exception** `dbl.NotFound(response, message)`

Exception that's thrown for when status code 404 occurs.

Subclass of *HTTPException*

**exception** `dbl.InvalidArgument`

Exception that's thrown when an argument to a function is invalid some way (e.g. wrong value or wrong type).

This could be considered the analogous of `ValueError` and `TypeError` except derived from *ClientException* and thus *DBLException*.

**exception** `dbl.ConnectionClosed(original)`

Exception that's thrown when the gateway connection is closed for reasons that could not be handled internally.

**code**

The close code of the websocket.

**Type** `int`

**reason**

The reason provided for the closure.

**Type** `str`

## CHAPTER 3

---

### Indices and tables

---

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





## Symbols

`__version__` (*in module dbl*), 5

## C

`ClientException`, 9

`close()` (*dbl.DBLCClient method*), 6

`code` (*dbl.ConnectionClosed attribute*), 10

`ConnectionClosed`, 10

## D

`DBLClient` (*class in dbl*), 5

`DBLException`, 9

## F

`Forbidden`, 9

## G

`generate_widget_large()` (*dbl.DBLCClient method*), 6

`generate_widget_small()` (*dbl.DBLCClient method*), 6

`get_bot_info()` (*dbl.DBLCClient method*), 7

`get_bot_upvotes()` (*dbl.DBLCClient method*), 7

`get_bots()` (*dbl.DBLCClient method*), 7

`get_guild_count()` (*dbl.DBLCClient method*), 7

`get_user_info()` (*dbl.DBLCClient method*), 7

`get_user_vote()` (*dbl.DBLCClient method*), 8

`get_weekend_status()` (*dbl.DBLCClient method*), 8

`get_widget_large()` (*dbl.DBLCClient method*), 8

`get_widget_small()` (*dbl.DBLCClient method*), 8

`guild_count()` (*dbl.DBLCClient method*), 6

## H

`HTTPException`, 9

## I

`InvalidArgument`, 10

## N

`NotFound`, 9

## O

`on_dbl_test()` (*in module dbl*), 9

`on_dbl_vote()` (*in module dbl*), 8

`on_guild_post()` (*in module dbl*), 8

## P

`post_guild_count()` (*dbl.DBLCClient method*), 8

## R

`reason` (*dbl.ConnectionClosed attribute*), 10

`response` (*dbl.HTTPException attribute*), 9

## T

`text` (*dbl.HTTPException attribute*), 9

## U

`Unauthorized`, 9

`UnauthorizedDetected`, 9

## V

`version_info` (*in module dbl*), 5