

---

**dbots**

*Release 3.0.0*

Oct 26, 2020



---

# Contents

---

<b>1</b>	<b>Documentation Contents</b>	<b>3</b>
1.1	API Reference . . . . .	3
1.1.1	Version Related Info . . . . .	3
1.1.2	Classes . . . . .	3
1.1.3	HTTP Classes . . . . .	5
1.1.4	Services . . . . .	6
<b>2</b>	<b>Links</b>	<b>21</b>
	<b>Index</b>	<b>23</b>



A poster/wrapper for Discord bot lists



### 1.1 API Reference

The following section outlines the API of dbots.

#### 1.1.1 Version Related Info

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

`dbots.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'.

`dbots.__version__`

A string representation of the version. e.g. '1.0.0rc1'. This is based off of [PEP 440](#).

#### 1.1.2 Classes

**class** `dbots.Poster` (*client\_id*, *server\_count*, *user\_count*, *voice\_connections*, *on\_custom\_post=None*,  
*\*\*options*)

A class that posts server count to listing sites.

##### Parameters

- **client\_id** (*str*) – The client ID used for posting to a service.
- **server\_count** (function or *coroutine*) – The function to use when retrieving the amount of servers.
- **user\_count** (function or *coroutine*) – The function to use when retrieving the amount of users.

- **voice\_connections** (function or *coroutine*) – The function to use when retrieving the amount of voice connections.
- **on\_custom\_post** (Optional[function or *coroutine*]) – The function to use when posting to a *custom* service. This function will be used with the named parameters being *server\_count*, *user\_count* and *voice\_connections*.
- **loop** (Optional[`asyncio.AbstractEventLoop`]) – The `asyncio.AbstractEventLoop` to use for asynchronous operations. Defaults to `None`, in which case the default event loop is used via `asyncio.get_event_loop()`.
- **sharding** (Optional[`bool`]) – Whether or not to use sharding values.
- **shard\_count** (Optional[`int`]) – The amount of shards the client has.
- **shard\_id** (Optional[`int`]) – The shard ID the poster us posting for.
- **proxy** (Optional[`str`]) – Proxy URL.
- **proxy\_auth** (Optional[`aiohttp.BasicAuth`]) – An object that represents proxy HTTP Basic Authorization.
- **api\_keys** (Optional[`dict`]) – A dictionary of API keys with the key being service keys and values being tokens.

**client\_id**

The client ID of the poster.

**get\_key** (*service: str*) → str

Gets an API key.

**Parameters** **service** (*str*) – The service key to get.

**kill\_loop** ()

Cancels the current posting loop.

**manual\_post** (*server\_count, service=None, user\_count=None, voice\_connections=None*) →

`dbots.http.HTTPResponse`

Manually posts a server count to a service.

**Parameters**

- **server\_count** (*int*) – The server count to post to the service.
- **service** (Optional[`str`]) – The service to post to. Can be *None* to post to all services or *custom* to use the custom post method.
- **user\_count** (Optional[`int`]) – The user count to post to the service.
- **voice\_connections** (Optional[`int`]) – The voice connection count to post to the service.

**post** (*service=None*) → `dbots.http.HTTPResponse`

Posts the current clients server count to a service.

**Parameters** **service** (Optional[`str`]) – The service to post to. Can be *None* to post to all services or *custom* to use the custom post method.

**remove\_key** (*service: str*) → str

Removes an API key.

**Parameters** **service** (*str*) – The service key to remove.

**set\_key** (*service: str, key: str*) → str

Sets an API key.



**Parameters**

- **service** (*str*) – The service key to set.
- **key** (*str*) – The API key to use.

**shard\_count**

The shard count of the poster.

**shard\_id**

The shard ID of the poster.

**start\_loop** (*interval=1800*) → `dbots.poster.AsyncLoop`

Creates a loop that posts to all services every *n* seconds.

**Parameters interval** (`Optional[int]`) – The amount of time (in seconds) to post to all services again.

**class** `dbots.Service` (*token=None, \*\*options*)

Represents any postable service.

**Parameters**

- **token** (`Optional[str]`) – The token that the class will use to authorize requests.
- **proxy** (`Optional[str]`) – Proxy URL.
- **proxy\_auth** (`Optional[aiohttp.BasicAuth]`) – An object that represents proxy HTTP Basic Authorization.

**BASE\_URL**

The base URL that the service uses for API requests.

**Type** `Optional[str]`

**token**

The token that will be used for the service.

**Type** `str`

**http**

The HTTP client the service is using.

**Type** `HTTPClient`

**static get** (*key: str*)

Gets a service from a key.

**Parameters key** (*str*) – The name of the service to get.

**has\_token**

Whether or not the service class has a token.

**class** `dbots.AsyncLoop` (*timeout, callback*)

### 1.1.3 HTTP Classes

**class** `dbots.HTTPClient` (*base\_url=None, proxy=None, proxy\_auth=None*)

Represents an HTTP client that can send requests.

**class** `dbots.HTTPResponse` (*response, text*)

A wrapped response from an `HTTPClient`.

**body**

The body of the response. The response is auto-parsed to json if available.

**Type** `dict` or `str`

**text**

The text body of the response.

**Type** `str`

**raw**

The raw response from `aiohttp`.

**Type** `aiohttp.ClientResponse`

**status**

The HTTP status code of the response.

**Type** `int`

**method**

The method the request used.

**Type** `str`

**url**

The URL of the response.

**Type** `str`

## 1.1.4 Services

**class** `dbots.Arcane` (*token=None, \*\*options*)

Represents the Arcane Bot Center service.

**See also:**

- [Arcane Bot Center Website](#)
- [Arcane Bot Center API Documentation](#)

**class** `dbots.Blist` (*token=None, \*\*options*)

Represents the Blist service.

**See also:**

- [Blist Website](#)
- [Blist API Documentation](#)

**get\_bot** (*bot\_id: str*) → `dbots.http.HTTPResponse`

This function is a *coroutine* returning a `HTTPResponse`.

Gets the bot listed on this service.

**Parameters** `bot_id` (`str`) – The bot's ID.

**get\_bot\_votes** (*bot\_id: str*) → `dbots.http.HTTPResponse`

This function is a *coroutine* returning a `HTTPResponse`.

Gets the list of people who voted this bot on this service.

**Parameters** `bot_id` (`str`) – The bot's ID.

**get\_user** (*user\_id: str*) → dbots.http.HTTPResponse  
 This function is a *coroutine* returning a HTTPResponse.

Gets the user listed on this service.

**Parameters** **user\_id** (*str*) – The user’s ID.

**get\_widget\_url** (*bot\_id: str, widget\_type: str = 'normal', \*\*query*) → str  
 Gets the widget URL for this bot.

**Parameters**

- **bot\_id** (*str*) – The bot’s ID.
- **widget\_type** (Optional[*str*]) – The type of widget to show.
- **\*\*query** – The query string to append to the URL.

**class** dbots.**BotListSpace** (*token=None, \*\*options*)  
 Represents the botlist.space service.

**See also:**

- [botlist.space Website](#)
- [botlist.space API Documentation](#)

**get\_bot** (*bot\_id: str*) → dbots.http.HTTPResponse  
 This function is a *coroutine* returning a HTTPResponse.

Gets the bot listed on this service.

**Parameters** **bot\_id** (*str*) – The bot’s ID.

**get\_bot\_uptime** (*bot\_id: str*) → dbots.http.HTTPResponse  
 This function is a *coroutine* returning a HTTPResponse.

Gets the uptime of a bot listed on this service.

**Parameters** **bot\_id** (*str*) – The bot’s ID.

**get\_bot\_votes** (*bot\_id: str*) → dbots.http.HTTPResponse  
 This function is a *coroutine* returning a HTTPResponse.

Gets the list of people who voted this bot on this service.

**Parameters** **bot\_id** (*str*) – The bot’s ID.

**get\_bots** () → dbots.http.HTTPResponse  
 This function is a *coroutine* returning a HTTPResponse.

Gets a list of bots on this service.

**get\_statistics** () → dbots.http.HTTPResponse  
 This function is a *coroutine* returning a HTTPResponse.

Gets the statistics of this service.

**get\_user** (*user\_id: str*) → dbots.http.HTTPResponse  
 This function is a *coroutine* returning a HTTPResponse.

Gets the user listed on this service.

**Parameters** **user\_id** (*str*) – The user’s ID.

**get\_user\_bots** (*user\_id: str*) → dbots.http.HTTPResponse

This function is a *coroutine* returning a HTTPResponse.

Gets the user's bots listed for this service.

**Parameters** **user\_id** (*str*) – The user's ID.

**get\_widget\_url** (*bot\_id: str, style=1, \*\*query*) → str

Gets the widget URL for this bot.

**Parameters**

- **bot\_id** (*str*) – The bot's ID.
- **style** (Optional[*int*]) – The style of the widget.
- **\*\*query** – The query string to append to the URL.

**class** dbots.**BotsDataBase** (*token=None, \*\*options*)

Represents the BotsDataBase service.

**See also:**

- [BotsDataBase Website](#)
- [BotsDataBase API Documentation](#)

**get\_bot** (*user\_id: str*) → dbots.http.HTTPResponse

This function is a *coroutine* returning a HTTPResponse.

Gets the bot listed on this service.

**Parameters** **bot\_id** (*str*) – The bot's ID.

**get\_bot\_votes** (*bot\_id: str*) → dbots.http.HTTPResponse

This function is a *coroutine* returning a HTTPResponse.

Gets the list of people who voted this bot on this service.

**Parameters** **bot\_id** (*str*) – The bot's ID.

**get\_user** (*user\_id: str*) → dbots.http.HTTPResponse

This function is a *coroutine* returning a HTTPResponse.

Gets the user listed on this service.

**Parameters** **user\_id** (*str*) – The user's ID.

**class** dbots.**BotsForDiscord** (*token=None, \*\*options*)

Represents the Bots For Discord service.

**See also:**

- [Bots For Discord Website](#)
- [Bots For Discord API Documentation](#)

**get\_bot** (*bot\_id: str*) → dbots.http.HTTPResponse

This function is a *coroutine* returning a HTTPResponse.

Gets the bot listed on this service.

**Parameters** **bot\_id** (*str*) – The bot's ID.

**get\_bot\_votes** (*bot\_id: str*) → dbots.http.HTTPResponse

This function is a *coroutine* returning a HTTPResponse.

Gets the list of people who voted this bot on this service.

**Parameters** **bot\_id** (*str*) – The bot’s ID.

**get\_user** (*user\_id: str*) → dbots.http.HTTPResponse

This function is a *coroutine* returning a HTTPResponse.

Gets the user listed on this service.

**Parameters** **user\_id** (*str*) – The user’s ID.

**get\_user\_bots** (*user\_id: str*) → dbots.http.HTTPResponse

This function is a *coroutine* returning a HTTPResponse.

Gets the user’s bots listed for this service.

**Parameters** **user\_id** (*str*) – The user’s ID.

**get\_widget\_url** (*bot\_id: str, \*\*query*) → str

Gets the widget URL for this bot.

**Parameters**

- **bot\_id** (*str*) – The bot’s ID.
- **\*\*query** – The query string to append to the URL.

**class** dbots.**BotsOnDiscord** (*token=None, \*\*options*)

Represents the Bots On Discord service.

**See also:**

- [Bots On Discord Website](#)
- [Bots On Discord API Documentation](#)

**check\_review** (*bot\_id: str, user\_id: str*) → dbots.http.HTTPResponse

This function is a *coroutine* returning a HTTPResponse.

Checks whether or not a user has reviewed a bot.

**Parameters**

- **bot\_id** (*str*) – The bot’s ID.
- **user\_id** (*str*) – The user’s ID.

**get\_widget\_url** (*bot\_id: str, \*\*query*) → str

Gets the widget URL for this bot.

**Parameters**

- **bot\_id** (*str*) – The bot’s ID.
- **\*\*query** – The query string to append to the URL.

**class** dbots.**Carbon** (*token=None, \*\*options*)

Represents the Carbonitex service.

**See also:**

- [Carbonitex Website](#)

**get\_bots** () → dbots.http.HTTPResponse  
This function is a *coroutine* returning a HTTPResponse.  
Gets a list of bots on this service.

**class** dbots.DBLista (token=None, \*\*options)  
Represents the DBLista service.

**See also:**

- [DBLista Website](#)
- [DBLista API Documentation](#)

**add\_bot** (data: dict) → dbots.http.HTTPResponse  
This function is a *coroutine* returning a HTTPResponse.  
Updates the bot's listing with the data provided.

**Parameters** **data** (dict) – The data being posted. This should include the ID of the bot.

**get\_bot** (bot\_id: str) → dbots.http.HTTPResponse  
This function is a *coroutine* returning a HTTPResponse.  
Gets the bot's stats on this service.

**Parameters** **bot\_id** (str) – The bot's ID.

**get\_bots** (page: int = 0) → dbots.http.HTTPResponse  
This function is a *coroutine* returning a HTTPResponse.  
Gets the bot's stats on this service.

**Parameters** **page** (int) – The page you want to get.

**get\_rejected\_bots** () → dbots.http.HTTPResponse  
This function is a *coroutine* returning a HTTPResponse.  
Gets a list of rejected bots on this service.

**get\_unverified\_bots** () → dbots.http.HTTPResponse  
This function is a *coroutine* returning a HTTPResponse.  
Gets a list of unverified bots on this service.

**rate\_bot** (bot\_id: str, data: dict) → dbots.http.HTTPResponse  
This function is a *coroutine* returning a HTTPResponse.  
Adds a rating to a bot on the service.

**Parameters**

- **bot\_id** (str) – The bot's ID.
- **data** (dict) – The data being posted. This should include the ID of the bot.

**remove\_bot** (bot\_id: str) → dbots.http.HTTPResponse  
This function is a *coroutine* returning a HTTPResponse.  
Removes a bot from the service.

**Parameters** **bot\_id** (str) – The bot's ID.

**remove\_rating** (bot\_id: str) → dbots.http.HTTPResponse  
This function is a *coroutine* returning a HTTPResponse.  
Removes a rating from a bot on the service.

**Parameters** `bot_id` (`str`) – The bot’s ID.

**search** (`query: str`) → `dbots.http.HTTPResponse`  
This function is a *coroutine* returning a `HTTPResponse`.

Searches for bots on the service.

**Parameters** `query` (`str`) – The query to search for.

**update\_bot** (`data: dict`) → `dbots.http.HTTPResponse`  
This function is a *coroutine* returning a `HTTPResponse`.

Updates the bot’s listing with the data provided.

**Parameters** `data` (`dict`) – The data being posted. This should include the ID of the bot.

**class** `dbots.DiscordBotsCo` (`token=None, **options`)  
Represents the DiscordBots.co service.

**See also:**

- [DiscordBots.co Website](#)
- [DiscordBots.co API Documentation](#)

**get\_bot** (`bot_id: str`) → `dbots.http.HTTPResponse`  
This function is a *coroutine* returning a `HTTPResponse`.

Gets the bot listed on this service.

**Parameters** `bot_id` (`str`) – The bot’s ID.

**class** `dbots.DiscordBotsGG` (`token=None, **options`)  
Represents the Discord Bots service.

**See also:**

- [Discord Bots Website](#)
- [Discord Bots API Documentation](#)

**get\_bot** (`bot_id: str, **query`) → `dbots.http.HTTPResponse`  
This function is a *coroutine* returning a `HTTPResponse`.

Gets the bot listed on this service.

**Parameters**

- `bot_id` (`str`) – The bot’s ID.
- `**query` – The query string to append to the URL.

**get\_bots** (`**query`) → `dbots.http.HTTPResponse`  
This function is a *coroutine* returning a `HTTPResponse`.

Gets a list of bots on this service.

**Parameters** `**query` – The query string to append to the URL.

**class** `dbots.DiscordAppsDev` (`token=None, **options`)  
Represents the Discord Apps service.

**See also:**

- [Discord Apps Website](#)

- [Discord Apps API Documentation](#)

**get\_apps** (\*\*query) → dbots.http.HTTPResponse  
This function is a *coroutine* returning a HTTPResponse.

Gets a list of applications on this service.

**get\_bot** (bot\_id: str) → dbots.http.HTTPResponse  
This function is a *coroutine* returning a HTTPResponse.

Gets the bot listed on this service.

**Parameters** **bot\_id** (str) – The bot’s ID.

**get\_bots** (\*\*query) → dbots.http.HTTPResponse  
This function is a *coroutine* returning a HTTPResponse.

Gets a list of bots on this service.

**get\_rpc\_apps** (\*\*query) → dbots.http.HTTPResponse  
This function is a *coroutine* returning a HTTPResponse.

Gets a list of RPC applications on this service.

**update\_bot** (bot\_id: str, data: dict) → dbots.http.HTTPResponse  
This function is a *coroutine* returning a HTTPResponse.

Updates the bot with the data provided.

**Parameters**

- **bot\_id** (str) – The bot’s ID.
- **data** (dict) – The data being posted.

**class** dbots.DiscordBoats (token=None, \*\*options)  
Represents the Discord Boats service.

**See also:**

- [Discord Boats Website](#)
- [Discord Boats API Documentation](#)

**get\_bot** (bot\_id: str) → dbots.http.HTTPResponse  
This function is a *coroutine* returning a HTTPResponse.

Gets the bot listed on this service.

**Parameters** **bot\_id** (str) – The bot’s ID.

**get\_user** (user\_id: str) → dbots.http.HTTPResponse  
This function is a *coroutine* returning a HTTPResponse.

Gets the user listed on this service.

**Parameters** **user\_id** (str) – The user’s ID.

**get\_widget\_url** (bot\_id: str, \*\*query) → str  
Gets the widget URL for this bot.

**Parameters**

- **bot\_id** (str) – The bot’s ID.
- **\*\*query** – The query string to append to the URL.



**user\_voted** (*bot\_id: str, user\_id: str*) → dbots.http.HTTPResponse

This function is a *coroutine* returning a HTTPResponse.

Checks whether or not a user has reviewed a bot.

**Parameters**

- **bot\_id** (*str*) – The bot’s ID.
- **user\_id** (*str*) – The user’s ID.

**class** dbots.**DiscordBotList** (*token=None, \*\*options*)

Represents the Discord Bot List service.

**See also:**

- [Discord Bot List Website](#)
- [Discord Bot List API Documentation](#)

**class** dbots.**DiscordBotWorld** (*token=None, \*\*options*)

Represents the Discord Bot World service.

**See also:**

- [Discord Bot World Website](#)
- [Discord Bot World API Documentation](#)

**get\_bot** (*bot\_id: str*) → dbots.http.HTTPResponse

This function is a *coroutine* returning a HTTPResponse.

Gets the bot listed on this service.

**Parameters** **bot\_id** (*str*) – The bot’s ID.

**get\_bot\_likes** (*bot\_id: str*) → dbots.http.HTTPResponse

This function is a *coroutine* returning a HTTPResponse.

Gets the list of people who liked this bot on this service.

**Parameters** **bot\_id** (*str*) – The bot’s ID.

**get\_bot\_stats** (*bot\_id: str*) → dbots.http.HTTPResponse

This function is a *coroutine* returning a HTTPResponse.

Gets the bot’s stats on this service.

**Parameters** **bot\_id** (*str*) – The bot’s ID.

**get\_bots** () → dbots.http.HTTPResponse

This function is a *coroutine* returning a HTTPResponse.

Gets a list of bots on this service.

**get\_user** (*user\_id: str*) → dbots.http.HTTPResponse

This function is a *coroutine* returning a HTTPResponse.

Gets the user listed on this service.

**Parameters** **user\_id** (*str*) – The user’s ID.

**class** dbots.**DiscordExtremeList** (*token=None, \*\*options*)

Represents the Discord Extreme List service.

**See also:**

- [Discord Extreme List Website](#)
- [Discord Extreme List API Documentation](#)

**get\_bot** (*bot\_id: str*) → dbots.http.HTTPResponse  
This function is a *coroutine* returning a HTTPResponse.

Gets the bot listed on this service.

**Parameters** **bot\_id** (*str*) – The bot’s ID.

**get\_bot\_likes** (*bot\_id: str*) → dbots.http.HTTPResponse  
This function is a *coroutine* returning a HTTPResponse.

Gets the list of people who liked this bot on this service.

**Parameters** **bot\_id** (*str*) – The bot’s ID.

**get\_bot\_stats** (*bot\_id: str*) → dbots.http.HTTPResponse  
This function is a *coroutine* returning a HTTPResponse.

Gets the bot’s stats on this service.

**Parameters** **bot\_id** (*str*) – The bot’s ID.

**get\_statistics** () → dbots.http.HTTPResponse  
This function is a *coroutine* returning a HTTPResponse.

Gets the statistics of this service.

**get\_user** (*user\_id: str*) → dbots.http.HTTPResponse  
This function is a *coroutine* returning a HTTPResponse.

Gets the user listed on this service.

**Parameters** **user\_id** (*str*) – The user’s ID.

**class** dbots.**DiscordLabs** (*token=None, \*\*options*)  
Represents the Discord Labs service.

**See also:**

- [Discord Labs Website](#)
- [Discord Labs API Documentation](#)

**get\_bot** (*bot\_id: str*) → dbots.http.HTTPResponse  
This function is a *coroutine* returning a HTTPResponse.

Gets the bot listed on this service.

**Parameters** **bot\_id** (*str*) – The bot’s ID.

**class** dbots.**DiscordListology** (*token=None, \*\*options*)  
Represents the DiscordListology service.

**See also:**

- [DiscordListology Website](#)
- [DiscordListology API Documentation](#)

**get\_bot\_stats** (*bot\_id: str*) → dbots.http.HTTPResponse

This function is a *coroutine* returning a HTTPResponse.

Gets the bot's stats listed on this service. :param bot\_id: The bot's ID. :type bot\_id: *str*

**get\_guild\_stats** (*guild\_id: str*) → dbots.http.HTTPResponse

This function is a *coroutine* returning a HTTPResponse.

Gets the guild's stats listed on this service. :param guild\_id: The guild's ID. :type guild\_id: *str*

**user\_voted\_bot** (*bot\_id: str, user\_id: str*) → dbots.http.HTTPResponse

This function is a *coroutine* returning a HTTPResponse.

Checks whether or not a user has voted for a bot on this service. :param bot\_id: The bot's ID. :type bot\_id: *str* :param user\_id: The user's ID. :type user\_id: *str*

**user\_voted\_guild** (*guild\_id: str, user\_id: str*) → dbots.http.HTTPResponse

This function is a *coroutine* returning a HTTPResponse.

Checks whether or not a user has voted for a guild on this service. :param guild\_id: The guild's ID. :type guild\_id: *str* :param user\_id: The user's ID. :type user\_id: *str*

**class** dbots.LBots (*token=None, \*\*options*)

Represents the LBots service.

**See also:**

- [LBots Website](#)
- [LBots API Documentation](#)

**get\_bot\_favorites** (*bot\_id: str*) → dbots.http.HTTPResponse

This function is a *coroutine* returning a HTTPResponse.

Gets the list of people who favorited this bot on this service.

**Parameters** *bot\_id* (*str*) – The bot's ID.

**get\_panel\_guild\_settings** (*bot\_id: str, guild\_id: str*) → dbots.http.HTTPResponse

This function is a *coroutine* returning a HTTPResponse.

Gets the list of people who favorited this bot on this service.

**Parameters**

- *bot\_id* (*str*) – The bot's ID.
- *guild\_id* (*str*) – The guild's ID.

**invalidate** (*bot\_id: str*) → dbots.http.HTTPResponse

This function is a *coroutine* returning a HTTPResponse.

Invalidates the token being used in the request.

**Parameters** *bot\_id* (*str*) – The bot's ID.

**update\_panel\_guilds** (*bot\_id: str, guild\_id: str, data: dict*) → dbots.http.HTTPResponse

This function is a *coroutine* returning a HTTPResponse.

Gets a guilds settings from the bot's panel.

**Parameters**

- *bot\_id* (*str*) – The bot's ID.
- *guild\_id* (*str*) – The guild's ID.

- **data** (*dict*) – The data being posted.

**user\_favorited** (*bot\_id: str, user\_id: str*) → *dbots.http.HTTPResponse*  
This function is a *coroutine* returning a *HTTPResponse*.

Checks whether or not a user has favorited a bot on this service.

**Parameters**

- **bot\_id** (*str*) – The bot’s ID.
- **user\_id** (*str*) – The user’s ID.

**class** *dbots.ListMyBots* (*token=None, \*\*options*)  
Represents the List My Bots service.

**See also:**

- [List My Bots Website](#)

**get\_bot** (*bot\_id: str*) → *dbots.http.HTTPResponse*  
This function is a *coroutine* returning a *HTTPResponse*.

Gets the bot listed on this service.

**Parameters** **bot\_id** (*str*) – The bot’s ID.

**get\_status\_widget\_url** (*bot\_id: str, \*\*query*) → *str*  
Gets the status widget URL for this bot.

**Parameters**

- **bot\_id** (*str*) – The bot’s ID.
- **\*\*query** – The query string to append to the URL.

**get\_user\_bots** (*user\_id: str*) → *dbots.http.HTTPResponse*  
This function is a *coroutine* returning a *HTTPResponse*.

Gets the user’s bots listed on this service.

**Parameters** **user\_id** (*str*) – The user’s ID.

**get\_widget\_url** (*bot\_id: str, \*\*query*) → *str*  
Gets the widget URL for this bot.

**Parameters**

- **bot\_id** (*str*) – The bot’s ID.
- **\*\*query** – The query string to append to the URL.

**class** *dbots.MythicalBots* (*token=None, \*\*options*)  
Represents the Mythical Bots service.

**See also:**

- [Mythical Bots Website](#)
- [Mythical Bots API Documentation](#)

**get\_bot** (*bot\_id: str*) → *dbots.http.HTTPResponse*  
This function is a *coroutine* returning a *HTTPResponse*.

Gets the bot listed on this service.

**Parameters** `bot_id` (`str`) – The bot’s ID.

`get_user` (`user_id: str`) → `dbots.http.HTTPResponse`  
This function is a *coroutine* returning a `HTTPResponse`.

Gets the user listed on this service.

**Parameters** `user_id` (`str`) – The user’s ID.

`get_widget_url` (`bot_id: str, **query`) → `str`  
Gets the widget URL for this bot.

**Parameters**

- `bot_id` (`str`) – The bot’s ID.
- `**query` – The query string to append to the URL.

**class** `dbots.SpaceBotsList` (`token=None, **options`)  
Represents the Space Bots List service.

**See also:**

- [Space Bots List Website](#)
- [Space Bots List API Documentation](#)

`get_bot` (`bot_id: str`) → `dbots.http.HTTPResponse`  
This function is a *coroutine* returning a `HTTPResponse`.

Gets the bot listed on this service.

**Parameters** `bot_id` (`str`) – The bot’s ID.

**class** `dbots.TopCord` (`token=None, **options`)  
Represents the TopCord service.

**See also:**

- [TopCord Website](#)
- [TopCord API Documentation](#)

`get_bot` (`bot_id: str`) → `dbots.http.HTTPResponse`  
This function is a *coroutine* returning a `HTTPResponse`.

Gets the bot listed on this service.

**Parameters** `bot_id` (`str`) – The bot’s ID.

**class** `dbots.TopGG` (`token=None, **options`)  
Represents the Top.gg service.

**See also:**

- [Top.gg Website](#)
- [Top.gg API Documentation](#)

`get_bot` (`bot_id: str`) → `dbots.http.HTTPResponse`  
This function is a *coroutine* returning a `HTTPResponse`.

Gets the bot listed on this service.

**Parameters** `bot_id` (`str`) – The bot’s ID.

**get\_bot\_stats** (*bot\_id: str*) → dbots.http.HTTPResponse  
This function is a *coroutine* returning a HTTPResponse.

Gets the bot's stats listed on this service.

**Parameters** **bot\_id** (*str*) – The bot's ID.

**get\_bot\_votes** (*bot\_id: str*) → dbots.http.HTTPResponse  
This function is a *coroutine* returning a HTTPResponse.

Gets the list of people who voted this bot on this service.

**Parameters** **bot\_id** (*str*) – The bot's ID.

**get\_bots** (*\*\*query*) → dbots.http.HTTPResponse  
This function is a *coroutine* returning a HTTPResponse.

Gets a list of bots on this service.

**Parameters** **\*\*query** – The query string to append to the URL.

**get\_user** (*user\_id: str*) → dbots.http.HTTPResponse  
This function is a *coroutine* returning a HTTPResponse.

Gets the user listed on this service.

**Parameters** **user\_id** (*str*) – The user's ID.

**get\_widget\_url** (*bot\_id: str, small\_widget: str = None, \*\*query*) → str  
Gets the widget URL for this bot.

**Parameters**

- **bot\_id** (*str*) – The bot's ID.
- **small\_widget** (Optional[*str*]) – The sub-path name to turn the widget into a badge. (i.e. owner)
- **\*\*query** – The query string to append to the URL.

**user\_voted** (*bot\_id: str, user\_id: str*) → dbots.http.HTTPResponse  
This function is a *coroutine* returning a HTTPResponse.

Gets the list of people who voted this bot on this service.

**Parameters**

- **bot\_id** (*str*) – The bot's ID.
- **user\_id** (*str*) – The user's ID.

**class** dbots.**WonderBotList** (*token=None, \*\*options*)  
Represents the Wonder Bot List service.

**See also:**

- [Wonder Bot List Website](#)
- [Wonder Bot List API Documentation](#)

**get\_bot** (*bot\_id: str*) → dbots.http.HTTPResponse  
This function is a *coroutine* returning a HTTPResponse.

Gets the bot listed on this service.

**Parameters** **bot\_id** (*str*) – The bot's ID.

**get\_user** (*user\_id: str*) → dbots.http.HTTPResponse  
 This function is a *coroutine* returning a HTTPResponse.

Gets the user listed on this service.

**Parameters** **user\_id** (*str*) – The user’s ID.

**class** dbots.**YABL** (*token=None, \*\*options*)  
 Represents the YABL service.

**See also:**

- [YABL Website](#)
- [YABL API Documentation](#)

**get\_bot** (*bot\_id: str*) → dbots.http.HTTPResponse  
 This function is a *coroutine* returning a HTTPResponse.

Gets the bot listed on this service.

**Parameters** **bot\_id** (*str*) – The bot’s ID.

**get\_bots** () → dbots.http.HTTPResponse  
 This function is a *coroutine* returning a HTTPResponse.

Gets a list of bots on this service.

**get\_bots\_by\_page** (*\*\*query*) → dbots.http.HTTPResponse  
 This function is a *coroutine* returning a HTTPResponse.

Gets a page of bots on this service.

**Parameters** **\*\*query** – The query string to append to the URL.

**get\_random\_bots** () → dbots.http.HTTPResponse  
 This function is a *coroutine* returning a HTTPResponse.

Gets 20 random bots from this service.

**get\_unverified\_bots** () → dbots.http.HTTPResponse  
 This function is a *coroutine* returning a HTTPResponse.

Gets a list of unverified bots on this service.

**get\_user\_bots** (*user\_id: str*) → dbots.http.HTTPResponse  
 This function is a *coroutine* returning a HTTPResponse.

Gets the user’s bots listed for this service.

**Parameters** **user\_id** (*str*) – The user’s ID.

**invalidate** (*bot\_id: str*) → dbots.http.HTTPResponse  
 This function is a *coroutine* returning a HTTPResponse.

Invalidates the token being used in the request.





## CHAPTER 2

---

### Links

---

- [PyPi](#)
- [Libraries.io](#)
- [README.md](#)
- [genindex](#)
- [search](#)



## Symbols

`__version__` (in module *dbots*), 3

### A

`add_bot()` (*dbots.DBLista* method), 10

*Arcane* (class in *dbots*), 6

*AsyncLoop* (class in *dbots*), 5

### B

`BASE_URL` (*dbots.Service* attribute), 5

*Blist* (class in *dbots*), 6

`body` (*dbots.HTTPResponse* attribute), 5

*BotListSpace* (class in *dbots*), 7

*BotsDataBase* (class in *dbots*), 8

*BotsForDiscord* (class in *dbots*), 8

*BotsOnDiscord* (class in *dbots*), 9

### C

*Carbon* (class in *dbots*), 9

`check_review()` (*dbots.BotsOnDiscord* method), 9

`client_id` (*dbots.Poster* attribute), 4

### D

*DBLista* (class in *dbots*), 10

*DiscordAppsDev* (class in *dbots*), 11

*DiscordBoats* (class in *dbots*), 12

*DiscordBotList* (class in *dbots*), 13

*DiscordBotsCo* (class in *dbots*), 11

*DiscordBotsGG* (class in *dbots*), 11

*DiscordBotWorld* (class in *dbots*), 13

*DiscordExtremeList* (class in *dbots*), 13

*DiscordLabs* (class in *dbots*), 14

*DiscordListology* (class in *dbots*), 14

### G

`get()` (*dbots.Service* static method), 5

`get_apps()` (*dbots.DiscordAppsDev* method), 12

`get_bot()` (*dbots.Blist* method), 6

`get_bot()` (*dbots.BotListSpace* method), 7

`get_bot()` (*dbots.BotsDataBase* method), 8

`get_bot()` (*dbots.BotsForDiscord* method), 8

`get_bot()` (*dbots.DBLista* method), 10

`get_bot()` (*dbots.DiscordAppsDev* method), 12

`get_bot()` (*dbots.DiscordBoats* method), 12

`get_bot()` (*dbots.DiscordBotsCo* method), 11

`get_bot()` (*dbots.DiscordBotsGG* method), 11

`get_bot()` (*dbots.DiscordBotWorld* method), 13

`get_bot()` (*dbots.DiscordExtremeList* method), 14

`get_bot()` (*dbots.DiscordLabs* method), 14

`get_bot()` (*dbots.ListMyBots* method), 16

`get_bot()` (*dbots.MythicalBots* method), 16

`get_bot()` (*dbots.SpaceBotsList* method), 17

`get_bot()` (*dbots.TopCord* method), 17

`get_bot()` (*dbots.TopGG* method), 17

`get_bot()` (*dbots.WonderBotList* method), 18

`get_bot()` (*dbots.YABL* method), 19

`get_bot_favorites()` (*dbots.LBots* method), 15

`get_bot_likes()` (*dbots.DiscordBotWorld* method), 13

`get_bot_likes()` (*dbots.DiscordExtremeList* method), 14

`get_bot_stats()` (*dbots.DiscordBotWorld* method), 13

`get_bot_stats()` (*dbots.DiscordExtremeList* method), 14

`get_bot_stats()` (*dbots.DiscordListology* method), 14

`get_bot_stats()` (*dbots.TopGG* method), 17

`get_bot_uptime()` (*dbots.BotListSpace* method), 7

`get_bot_votes()` (*dbots.Blist* method), 6

`get_bot_votes()` (*dbots.BotListSpace* method), 7

`get_bot_votes()` (*dbots.BotsDataBase* method), 8

`get_bot_votes()` (*dbots.BotsForDiscord* method), 8

`get_bot_votes()` (*dbots.TopGG* method), 18

`get_bots()` (*dbots.BotListSpace* method), 7

`get_bots()` (*dbots.Carbon* method), 9

`get_bots()` (*dbots.DBLista* method), 10

`get_bots()` (*dbots.DiscordAppsDev* method), 12

`get_bots()` (*dbots.DiscordBotsGG* method), 11

- get\_bots() (*dbots.DiscordBotWorld* method), 13
  - get\_bots() (*dbots.TopGG* method), 18
  - get\_bots() (*dbots.YABL* method), 19
  - get\_bots\_by\_page() (*dbots.YABL* method), 19
  - get\_guild\_stats() (*dbots.DiscordListology* method), 15
  - get\_key() (*dbots.Poster* method), 4
  - get\_panel\_guild\_settings() (*dbots.LBots* method), 15
  - get\_random\_bots() (*dbots.YABL* method), 19
  - get\_rejected\_bots() (*dbots.DBLista* method), 10
  - get\_rpc\_apps() (*dbots.DiscordAppsDev* method), 12
  - get\_statistics() (*dbots.BotListSpace* method), 7
  - get\_statistics() (*dbots.DiscordExtremeList* method), 14
  - get\_status\_widget\_url() (*dbots.ListMyBots* method), 16
  - get\_unverified\_bots() (*dbots.DBLista* method), 10
  - get\_unverified\_bots() (*dbots.YABL* method), 19
  - get\_user() (*dbots.Blist* method), 6
  - get\_user() (*dbots.BotListSpace* method), 7
  - get\_user() (*dbots.BotsDataBase* method), 8
  - get\_user() (*dbots.BotsForDiscord* method), 9
  - get\_user() (*dbots.DiscordBoats* method), 12
  - get\_user() (*dbots.DiscordBotWorld* method), 13
  - get\_user() (*dbots.DiscordExtremeList* method), 14
  - get\_user() (*dbots.MythicalBots* method), 17
  - get\_user() (*dbots.TopGG* method), 18
  - get\_user() (*dbots.WonderBotList* method), 18
  - get\_user\_bots() (*dbots.BotListSpace* method), 7
  - get\_user\_bots() (*dbots.BotsForDiscord* method), 9
  - get\_user\_bots() (*dbots.ListMyBots* method), 16
  - get\_user\_bots() (*dbots.YABL* method), 19
  - get\_widget\_url() (*dbots.Blist* method), 7
  - get\_widget\_url() (*dbots.BotListSpace* method), 8
  - get\_widget\_url() (*dbots.BotsForDiscord* method), 9
  - get\_widget\_url() (*dbots.BotsOnDiscord* method), 9
  - get\_widget\_url() (*dbots.DiscordBoats* method), 12
  - get\_widget\_url() (*dbots.ListMyBots* method), 16
  - get\_widget\_url() (*dbots.MythicalBots* method), 17
  - get\_widget\_url() (*dbots.TopGG* method), 18
- H**
- has\_token (*dbots.Service* attribute), 5
  - http (*dbots.Service* attribute), 5
  - HTTPClient (*class in dbots*), 5
  - HTTPResponse (*class in dbots*), 5
- I**
- invalidate() (*dbots.LBots* method), 15
  - invalidate() (*dbots.YABL* method), 19
- K**
- kill\_loop() (*dbots.Poster* method), 4
- L**
- LBots (*class in dbots*), 15
  - ListMyBots (*class in dbots*), 16
- M**
- manual\_post() (*dbots.Poster* method), 4
  - method (*dbots.HTTPResponse* attribute), 6
  - MythicalBots (*class in dbots*), 16
- P**
- post() (*dbots.Poster* method), 4
  - Poster (*class in dbots*), 3
  - Python Enhancement Proposals  
PEP 440, 3
- R**
- rate\_bot() (*dbots.DBLista* method), 10
  - raw (*dbots.HTTPResponse* attribute), 6
  - remove\_bot() (*dbots.DBLista* method), 10
  - remove\_key() (*dbots.Poster* method), 4
  - remove\_rating() (*dbots.DBLista* method), 10
- S**
- search() (*dbots.DBLista* method), 11
  - Service (*class in dbots*), 5
  - set\_key() (*dbots.Poster* method), 4
  - shard\_count (*dbots.Poster* attribute), 5
  - shard\_id (*dbots.Poster* attribute), 5
  - SpaceBotsList (*class in dbots*), 17
  - start\_loop() (*dbots.Poster* method), 5
  - status (*dbots.HTTPResponse* attribute), 6
- T**
- text (*dbots.HTTPResponse* attribute), 6
  - token (*dbots.Service* attribute), 5
  - TopCord (*class in dbots*), 17
  - TopGG (*class in dbots*), 17
- U**
- update\_bot() (*dbots.DBLista* method), 11
  - update\_bot() (*dbots.DiscordAppsDev* method), 12
  - update\_panel\_guilds() (*dbots.LBots* method), 15
  - url (*dbots.HTTPResponse* attribute), 6
  - user\_favorited() (*dbots.LBots* method), 16
  - user\_voted() (*dbots.DiscordBoats* method), 12
  - user\_voted() (*dbots.TopGG* method), 18
  - user\_voted\_bot() (*dbots.DiscordListology* method), 15

`user_voted_guild()` (*dbots.DiscordListology*  
*method*), 15

## V

`version_info` (*in module dbots*), 3

## W

`WonderBotList` (*class in dbots*), 18

## Y

`YABL` (*class in dbots*), 19