dbots

Release 3.0.0

Snazzah

Nov 01, 2021

CONTENTS

1 Documentation Contents **3**

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

2 Links **23**

Index **25**

A poster/wrapper for Discord bot lists

CHAPTER

ONE

DOCUMENTATION CONTENTS

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.

property client_id: str
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.

async 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.

async 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.

property shard_count: int
The shard count of the poster.

property shard_id: int
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.

property has_token: bool
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.**BladeList**(*token=None*, ***options*)
Represents the BladeList service.

See also:

- BladeList Website
- BladeList API Documentation

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

Gets a bot listed on this service.

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

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

Gets the bot's reviews 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_server(*server_id: str*) → *dbots.http.HTTPResponse*
This function is a *coroutine* returning a HTTPResponse.

Gets the server listed on this service.

Parameters **server_id** (str) – The server'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 on this service.

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

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

Gets the user's servers 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.**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.**DBots**(*token=None*, ***options*)
Represents the DBots service.

See also:

- DBots Website
- DBots API Documentation

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

Gets the bot's audit logs.

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

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

Regenerates the bot API token.

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

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.**DiscordBotlistEU**(*token=None*, ***options*)
Represents the DiscordBotlistEU service.

See also:

- DiscordBotlistEU Website
- DiscordBotlistEU API Documentation

get_analytics() → *dbots.http.HTTPResponse*
This function is a *coroutine* returning a HTTPResponse.

Gets this bot's analytics.

get_bot() → *dbots.http.HTTPResponse*
This function is a *coroutine* returning a HTTPResponse.

Gets this bot's data.

get_votes() → *dbots.http.HTTPResponse*
This function is a *coroutine* returning a HTTPResponse.

Gets this bot's votes.

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.**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_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.

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

Gets the votes for this bot.

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

class dbots.**DiscordListSpace**(*token=None*, ***options*)
Represents the discordlist.space service.

See also:

- discordlist.space Website

- discordlist.space API Documentation

get_audit_log(*bot_id: str*, ***query*) → *dbots.http.HTTPResponse*
This function is a *coroutine* returning a HTTPResponse.

Gets the bot listing audit log.

Parameters

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

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_analytics(*bot_id: str*, ***query*) → *dbots.http.HTTPResponse*
This function is a *coroutine* returning a HTTPResponse.

Gets the analytics on a bot.

Parameters

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

get_bot_owners(*bot_id: str*, ***query*) → *dbots.http.HTTPResponse*
This function is a *coroutine* returning a HTTPResponse.

Gets the owners of the bot listing.

Parameters

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

get_bot_reviews(*bot_id: str*, ***query*) → *dbots.http.HTTPResponse*
This function is a *coroutine* returning a HTTPResponse.

Gets the reviews of a bot.

Parameters

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

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_languages(**query) → *dbots.http.HTTPResponse*
This function is a *coroutine* returning a HTTPResponse.

Gets all the available languages that bots or servers can set as their language.

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

get_statistics() → *dbots.http.HTTPResponse*
This function is a *coroutine* returning a HTTPResponse.

Gets the statistics of this service.

get_tags(**query) → *dbots.http.HTTPResponse*
This function is a *coroutine* returning a HTTPResponse.

Gets all available tags for use on bots or servers.

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

get_upvote_leaderboard(*bot_id: str*, **query) → *dbots.http.HTTPResponse*
This function is a *coroutine* returning a HTTPResponse.

Gets the top upvoters of this month.

Parameters

- **bot_id** (str) – The bot's ID.
- ****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_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_user_upvote(*bot_id: str*, *user_id: str*) → *dbots.http.HTTPResponse*
This function is a *coroutine* returning a HTTPResponse.

Checks if a specific user has upvoted the bot.

Parameters

- **bot_id** (str) – The bot's ID.
- **user_id** (str) – The user's ID.

get_widget_url(*bot_id: str*, *style: int = 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.**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.**DiscordServices**(*token=None*, ***options*)
Represents the DiscordServices service.

See also:

- DiscordServices Website
- DiscordServices API Documentation

post_commands(*bot_id: str*, *commands: list*) → *dbots.http.HTTPResponse*
This function is a *coroutine* returning a HTTPResponse.

Posts commands info to your bot page.

Parameters

- **bot_id** (`str`) – The bot's ID.
- **commands** (`list`) – The command info to post.

post_news(*bot_id: str*, *title: str*, *content: str*) → *dbots.http.HTTPResponse*
This function is a *coroutine* returning a HTTPResponse.

Posts news to your bot page.

Parameters

- **bot_id** (`str`) – The bot's ID.
- **title** (`str`) – The title of the post.
- **content** (`str`) – The content of the post.

class dbots.**DiscordsCom**(*token=None*, ***options*)
Represents the Discords.com service (formerly Bots For Discord).

See also:

- Discords.com Website
- Discords.com 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 a bot.

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

get_bot_votes_12h(*bot_id: str*) → *dbots.http.HTTPResponse*
This function is a *coroutine* returning a HTTPResponse.

Gets the list of people who voted a bot in the last 12 hours.

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.**Disforge**(*token=None*, ***options*)
Represents the Disforge service.

See also:

- Disforge Website
- Disforge API Documentation

get_homepage() → *dbots.http.HTTPResponse*
This function is a *coroutine* returning a HTTPResponse.

Retreives the data shown on the homepage.

get_stats() → *dbots.http.HTTPResponse*
This function is a *coroutine* returning a HTTPResponse.

Retreives statistics about Disforge.

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

See also:

- FatesList Website
- FatesList API Documentation

add_bot_command(*bot_id: str*, *command: dict*, ***query*) → *dbots.http.HTTPResponse*
This function is a *coroutine* returning a HTTPResponse.

Adds a bot promotion.

Parameters

- **bot_id** (str) – The bot's ID.
- **command** (dict) – The command payload.
- ****query** – The query string to append to the URL.

add_promotion(*bot_id: str*, *promotion: dict*) → *dbots.http.HTTPResponse*
This function is a *coroutine* returning a HTTPResponse.

Adds a bot promotion.

Parameters

- **bot_id** (str) – The bot's ID.
- **promotion** (dict) – The promotion payload.

delete_bot_command(*bot_id: str*, *command_id: str*) → *dbots.http.HTTPResponse*
This function is a *coroutine* returning a HTTPResponse.

Deletes a bot's command.

Parameters

- **bot_id** (str) – The bot's ID.
- **command_id** (str) – The command ID.

delete_promotion(*bot_id: str*, *promotion_id: str*) → *dbots.http.HTTPResponse*
This function is a *coroutine* returning a HTTPResponse.

Deletes a bot promotion.

Parameters

- **bot_id** (str) – The bot's ID.
- **promotion_id** (str) – The promotion ID.

edit_bot_command(*bot_id: str*, *command: dict*) → *dbots.http.HTTPResponse*
This function is a *coroutine* returning a HTTPResponse.

Edits a bot's command.

Parameters

- **bot_id** (str) – The bot's ID.

- **command** (dict) – The command payload.

edit_promotion(*bot_id: str*, *promotion: dict*) → *dbots.http.HTTPResponse*
This function is a *coroutine* returning a HTTPResponse.

Edits a bot promotion.

Parameters

- **bot_id** (str) – The bot's ID.
- **promotion** (dict) – The promotion payload.

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

Gets a bot from the API.

Parameters

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

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

Get a bot's commands.

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

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

Gets a bot promotion.

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

get_random_bot() → *dbots.http.HTTPResponse*
This function is a *coroutine* returning a HTTPResponse.

Gets a random bot.

get_votes(*bot_id: str*, *user_id: str*) → *dbots.http.HTTPResponse*
This function is a *coroutine* returning a HTTPResponse.

Gets the number of votes a user gave to a bot.

Parameters

- **bot_id** (str) – The bot's ID.
- **user_id** (str) – The user's ID.

get_votes_timestamped(*bot_id: str*, *user_id: str*) → *dbots.http.HTTPResponse*
This function is a *coroutine* returning a HTTPResponse.

Gets the number of votes a user gave to a bot with timestamps.

Parameters

- **bot_id** (str) – The bot's ID.
- **user_id** (str) – The user's ID.

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

Regenerates the API token.

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

class dbots.**InfinityBotList**(*token=None*, ***options*)
Represents the Infinity Bot List service.

See also:

- Infinity Bot List Website
- Infinity 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.**Listcord**(*token=None*, ***options*)
Represents the Listcord service.

See also:

- Listcord Website
- Listcord 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_pack(*pack_id: str*) → *dbots.http.HTTPResponse*
This function is a *coroutine* returning a HTTPResponse.

Gets a bot pack.

Parameters **pack_id** (str) – The pack's ID.

get_bot_packs() → *dbots.http.HTTPResponse*
This function is a *coroutine* returning a HTTPResponse.

Gets all botpacks.

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

Gets a bot's reviews.

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

user_voted(*bot_id: str*, *user_id: str*) → *dbots.http.HTTPResponse*
This function is a *coroutine* returning a HTTPResponse.

Gets whether a user has voted for a bot.

Parameters

- **bot_id** (str) – The bot's ID.
- **user_id** (str) – The user's ID.

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

See also:

- MotionBotlist Website
- MotionBotlist API Documentation

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

Gets a bot.

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 a bot's reviews.

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

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.

get_bots() → *dbots.http.HTTPResponse*
This function is a *coroutine* returning a HTTPResponse.

Lists every bot on this service.

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: Optional[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.**VoidBots**(*token=None*, ***options*)
Represents the Void Bots service.

See also:

- Void Bots Website
- Void 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_bot_analytics(*bot_id: str*) → *dbots.http.HTTPResponse*
This function is a *coroutine* returning a HTTPResponse.

Gets the bot's analytics on this service.

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

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

Gets the bot's reviews on this service.

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

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 voted for a 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

TWO

LINKS

- PyPi
- Libraries.io
- README.md
- genindex
- search

INDEX

Symbols

__version__ (*in module dbots*), 3

A

add_bot_command() (*dbots.FatesList method*), 15
add_promotion() (*dbots.FatesList method*), 15
AsyncLoop (*class in dbots*), 5

B

BASE_URL (*dbots.Service attribute*), 5
BladeList (*class in dbots*), 6
Blist (*class in dbots*), 6
body (*dbots.HTTPResponse attribute*), 5
BotsOnDiscord (*class in dbots*), 7

C

Carbon (*class in dbots*), 7
check_review() (*dbots.BotsOnDiscord method*), 7
client_id (*dbots.Poster property*), 4

D

DBots (*class in dbots*), 8
delete_bot_command() (*dbots.FatesList method*), 15
delete_promotion() (*dbots.FatesList method*), 15
DiscordBoats (*class in dbots*), 8
DiscordBotList (*class in dbots*), 9
DiscordBotlistEU (*class in dbots*), 9
DiscordBotsGG (*class in dbots*), 9
DiscordExtremeList (*class in dbots*), 10
DiscordLabs (*class in dbots*), 10
DiscordListology (*class in dbots*), 12
DiscordListSpace (*class in dbots*), 10
DiscordsCom (*class in dbots*), 13
DiscordServices (*class in dbots*), 13
Disforge (*class in dbots*), 14

E

edit_bot_command() (*dbots.FatesList method*), 15
edit_promotion() (*dbots.FatesList method*), 16

F

FatesList (*class in dbots*), 15

G

get() (*dbots.Service static method*), 5
get_analytics() (*dbots.DiscordBotlistEU method*), 9
get_audit() (*dbots.DBots method*), 8
get_audit_log() (*dbots.DiscordListSpace method*), 11
get_bot() (*dbots.BladeList method*), 6
get_bot() (*dbots.Blist method*), 6
get_bot() (*dbots.DiscordBoats method*), 8
get_bot() (*dbots.DiscordBotlistEU method*), 9
get_bot() (*dbots.DiscordBotsGG method*), 9
get_bot() (*dbots.DiscordExtremeList method*), 10
get_bot() (*dbots.DiscordLabs method*), 10
get_bot() (*dbots.DiscordListSpace method*), 11
get_bot() (*dbots.DiscordsCom method*), 14
get_bot() (*dbots.FatesList method*), 16
get_bot() (*dbots.InfinityBotList method*), 17
get_bot() (*dbots.Listcord method*), 17
get_bot() (*dbots.MotionBotlist method*), 18
get_bot() (*dbots.SpaceBotsList method*), 18
get_bot() (*dbots.TopCord method*), 18
get_bot() (*dbots.TopGG method*), 19
get_bot() (*dbots.VoidBots method*), 20
get_bot() (*dbots.WonderBotList method*), 20
get_bot() (*dbots.YABL method*), 21
get_bot_analytics() (*dbots.DiscordListSpace method*), 11
get_bot_analytics() (*dbots.VoidBots method*), 20
get_bot_commands() (*dbots.FatesList method*), 16
get_bot_owners() (*dbots.DiscordListSpace method*), 11
get_bot_pack() (*dbots.Listcord method*), 17
get_bot_packs() (*dbots.Listcord method*), 17
get_bot_promotion() (*dbots.FatesList method*), 16
get_bot_reviews() (*dbots.Blist method*), 6
get_bot_reviews() (*dbots.DiscordListSpace method*), 11
get_bot_reviews() (*dbots.Listcord method*), 17
get_bot_reviews() (*dbots.VoidBots method*), 20
get_bot_stats() (*dbots.DiscordListology method*), 13
get_bot_stats() (*dbots.TopGG method*), 19
get_bot_votes() (*dbots.Blist method*), 6
get_bot_votes() (*dbots.DiscordLabs method*), 10

get_bot_votes() (*dbots.DiscordListSpace method*), 11
get_bot_votes() (*dbots.DiscordsCom method*), 14
get_bot_votes() (*dbots.MotionBotlist method*), 18
get_bot_votes() (*dbots.TopGG method*), 19
get_bot_votes_12h() (*dbots.DiscordsCom method*), 14
get_bots() (*dbots.Carbon method*), 8
get_bots() (*dbots.DiscordBotsGG method*), 9
get_bots() (*dbots.DiscordListSpace method*), 11
get_bots() (*dbots.TopCord method*), 18
get_bots() (*dbots.TopGG method*), 19
get_bots() (*dbots.YABL method*), 21
get_bots_by_page() (*dbots.YABL method*), 21
get_guild_stats() (*dbots.DiscordListology method*), 13
get_homepage() (*dbots.Disforge method*), 14
get_key() (*dbots.Poster method*), 4
get_languages() (*dbots.DiscordListSpace method*), 11
get_random_bot() (*dbots.FatesList method*), 16
get_random_bots() (*dbots.YABL method*), 21
get_server() (*dbots.Blist method*), 6
get_statistics() (*dbots.DiscordExtremeList method*), 10
get_statistics() (*dbots.DiscordListSpace method*), 12
get_stats() (*dbots.Disforge method*), 14
get_tags() (*dbots.DiscordListSpace method*), 12
get_unverified_bots() (*dbots.YABL method*), 21
get_upvote_leaderboard() (*dbots.DiscordListSpace method*), 12
get_user() (*dbots.Blist method*), 7
get_user() (*dbots.DiscordBoats method*), 8
get_user() (*dbots.DiscordExtremeList method*), 10
get_user() (*dbots.DiscordListSpace method*), 12
get_user() (*dbots.DiscordsCom method*), 14
get_user() (*dbots.InfinityBotList method*), 17
get_user() (*dbots.TopGG method*), 19
get_user() (*dbots.WonderBotList method*), 20
get_user_bots() (*dbots.Blist method*), 7
get_user_bots() (*dbots.DiscordListSpace method*), 12
get_user_bots() (*dbots.DiscordsCom method*), 14
get_user_bots() (*dbots.YABL method*), 21
get_user_servers() (*dbots.Blist method*), 7
get_user_upvote() (*dbots.DiscordListSpace method*), 12
get_votes() (*dbots.DiscordBotlistEU method*), 9
get_votes() (*dbots.FatesList method*), 16
get_votes_timestamped() (*dbots.FatesList method*), 16
get_widget_url() (*dbots.Blist method*), 7
get_widget_url() (*dbots.BotsOnDiscord method*), 7
get_widget_url() (*dbots.DiscordBoats method*), 8
get_widget_url() (*dbots.DiscordListSpace method*), 12
get_widget_url() (*dbots.DiscordsCom method*), 14
get_widget_url() (*dbots.TopGG method*), 19

H

has_token (*dbots.Service property*), 5
http (*dbots.Service attribute*), 5
HTTPClient (*class in dbots*), 5
HTTPResponse (*class in dbots*), 5

I

InfinityBotList (*class in dbots*), 17
invalidate() (*dbots.YABL method*), 21

K

kill_loop() (*dbots.Poster method*), 4

L

Listcord (*class in dbots*), 17

M

manual_post() (*dbots.Poster method*), 4
method (*dbots.HTTPResponse attribute*), 6
MotionBotlist (*class in dbots*), 18

P

post() (*dbots.Poster method*), 4
post_commands() (*dbots.DiscordServices method*), 13
post_news() (*dbots.DiscordServices method*), 13
Poster (*class in dbots*), 3
Python Enhancement Proposals
    PEP 440, 3

R

raw (*dbots.HTTPResponse attribute*), 5
regen_token() (*dbots.DBots method*), 8
regenerate_token() (*dbots.FatesList method*), 16
remove_key() (*dbots.Poster method*), 4

S

Service (*class in dbots*), 5
set_key() (*dbots.Poster method*), 4
shard_count (*dbots.Poster property*), 4
shard_id (*dbots.Poster property*), 4
SpaceBotsList (*class in dbots*), 18
start_loop() (*dbots.Poster method*), 4
status (*dbots.HTTPResponse attribute*), 5

T

text (*dbots.HTTPResponse attribute*), 5
token (*dbots.Service attribute*), 5
TopCord (*class in dbots*), 18
TopGG (*class in dbots*), 18

U

url (*dbots.HTTPResponse attribute*), 6
user_voted() (*dbots.DiscordBoats method*), 9
user_voted() (*dbots.Listcord method*), 17
user_voted() (*dbots.TopGG method*), 19
user_voted() (*dbots.VoidBots method*), 20
user_voted_bot() (*dbots.DiscordListology method*), 13
user_voted_guild() (*dbots.DiscordListology method*), 13

V

version_info (*in module dbots*), 3
VoidBots (*class in dbots*), 19

W

WonderBotList (*class in dbots*), 20

Y

YABL (*class in dbots*), 20