
BB8 Documentation

Release 0.0.1

Jan Ferko

Sep 18, 2019

Contents

1	Commands	3
2	Instalation	5
3	API Reference	7
3.1	Api Reference	7
4	Developers	11
	Python Module Index	13
	Index	15

Welcome to BB8's documentation. BB8 is [Discord](#) bot, that provides useful tools for fans of The Star Wars Destiny game. It's open source and written in Python. This documentation is split into multiple parts. We recommend that you get started with [Commands](#) to get familiar with different commands that BB8 supports and then head to [Installation](#) to find out how to add BB8 to your channel.

If you'd like to dive into BB8 internals, we have [API Reference](#) section and if you'd like to contribute to BB8 check out [Developers](#) section.

CHAPTER 1

Commands

CHAPTER 2

Instalation

If you're looking for information on specific function, class or method, this part of documentation is for you.

3.1 Api Reference

3.1.1 Bot

This module provides BB8 bot implementation.

class `bb8.bot.BB8` (*extensions*, *search*, ***options*)
Bot implementation class.

It wires all dependencies required by extensions and loads different cogs.

Parameters

- **extensions** (*list*) – the list of extensions module names, that are loaded to bot.
- **search** (*obj*) – the card search class
- **command_prefix** (*str*) – the prefix of all commands.
- **description** (*str*) – the bot description

on_ready (**_*)
The callback for logging that commands are ready for use.

3.1.2 Cogs

This module provides bot commands and event listeners.

class `bb8.cogs.SWCardSearch` (*bot*)
This class provides command for searching Star Wars Destiny cards.

It also tries to parse card mentions from regular messages in channel and send results back to enhance conversation.

Cards must be mentioned in format `[[term]]` to be recognized by bot.

Parameters `bot` (*obj*) – the bot instance where command should be registered.

on_message (*message*)

The callback that tries to parse card mentions from all messages sent to channel.

Callback ignores messages from sent by bot and after it is done with its processing, it forwards it to other registered commands.

Parameters `message` (*obj*) – the message received from Discord channel.

3.1.3 Embeds

Provides different Discord embed renderings.

class `bb8.embed.CardDetail` (*card*, *emojis*)

Renders embed with card details.

footer_line ()

Constructs footer line containing illustrator, set membership and position.

Example:

Tony Fotti • Awakenings # 64

Returns the footer line describing card set and illustrator

Return type str

render ()

Renders card detail as Discord Embed.

Returns the rendered embed

text_line ()

Constructs card text by substituting icons with emojis and adding dice sides iconography.

Example:

special

- Deal 2 Damage to each of an opponent's characters.

Discard this upgrade from play.

Returns the card text and dice description

Return type str

type_line ()

Construct card's type containing faction, affiliation and rarity as well as cost.

Example:

Character: Villain - Rogue - Rare • Points: 8/11 • Health: 8

class `bb8.embed.CardEmbed` (*card*)

Provides base for all card embeds.

Parameters `card` – the card which embed should be rendered for.

```

colour()
    Gets faction colour of card.

class bb8.embed.CardImage(card)
    Renders embed with full-size image.

render()
    Renders card image as Discord Embed.

```

3.1.4 Search

This module provides Star Wars Destiny cards fuzzy search.

Example: This example shows simple fuzzy search:

```

$ search = Search()
$ search.find_card("captain phasma")

```

```

class bb8.search.Search(db_client=None)
    Searches for Star Wars Destiny cards based on their label.

    Parameters db_client (obj) – The client to get cards from datastore. (default: SWDestinyDB-Client)

find_card(text)
    Finds card with label closest to given text. If no card was found, it returns None. Score (0-100) is assigned to each potential and only matches with score > 50 are considered.

    Parameters text (string) – the partial label of card.

    Returns the card closest to given text

    Return type dict

```

3.1.5 SWDestinyDB

SWDestinyDB handling module.

This module provides utilities for communication with SWDestinyDB API. At the moment, it only supports public endpoints.

Example: A short usage example of getting card by its key:

```

$ client = SWDestinyClient()
$ client.get_card("01001")

```

```

class bb8.swdestinydb.SWDestinyDBClient(base_url=None, format='json', session=None)
    API client for StarWars Destiny DB.

```

Supports only publicly available endpoints. See [official docs](#) for more details.

Parameters

- **base_url** (*str*) – The url where api resources can be found. (default: <https://swdestinydb.com/api/public>)
- **format** (*str*) – The expected output format. (default: json)
- **session** (*obj*) – The HTTP session to communicate with API (default: requests.Session())

get_card (*key*)

Gets card with given key.

See [card docs](#) for more details.

Parameters **key** (*str*) – The card identifier

Returns The Card with given key.

Return type dict

get_cards (*set_code=None*)

Gets all cards.

If *set_code* is provided, it only returns cards from given set. See [cards docs](#) for more details.

Parameters **set_code** (*str*) – The code of set cards should be from, e.g 'AW'. (default: None)

Returns The list of found cards

Return type list(dict)

get_decklist (*key*)

Gets decklist by its identifier.

See [decklist docs](#) for more details.

Parameters **key** (*str*) – The decklist identifier

Returns The decklist with given key

get_formats ()

Gets all available competitive formats.

See [format docs](#) for more details.

Returns The list of available competitive formats

get_sets ()

Gets all available sets.

See [sets docs](#) for more details.

Returns The list of available card sets

exception `bb8.swdestinydb.UnsupportedFormat`

The exception thrown if requested format is not supported by client.

CHAPTER 4

Developers

b

`bb8.bot`, 7
`bb8.cogs`, 7
`bb8.embed`, 8
`bb8.search`, 9
`bb8.swdestinydb`, 9

B

BB8 (class in bb8.bot), 7
bb8.bot (module), 7
bb8.cogs (module), 7
bb8.embed (module), 8
bb8.search (module), 9
bb8.swdestinydb (module), 9

C

CardDetail (class in bb8.embed), 8
CardEmbed (class in bb8.embed), 8
CardImage (class in bb8.embed), 9
colour() (bb8.embed.CardEmbed method), 9

F

find_card() (bb8.search.Search method), 9
footer_line() (bb8.embed.CardDetail method), 8

G

get_card() (bb8.swdestinydb.SWDestinyDBClient method), 9
get_cards() (bb8.swdestinydb.SWDestinyDBClient method), 10
get_decklist() (bb8.swdestinydb.SWDestinyDBClient method), 10
get_formats() (bb8.swdestinydb.SWDestinyDBClient method), 10
get_sets() (bb8.swdestinydb.SWDestinyDBClient method), 10

O

on_message() (bb8.cogs.SWCardSearch method), 8
on_ready() (bb8.bot.BB8 method), 7

R

render() (bb8.embed.CardDetail method), 8
render() (bb8.embed.CardImage method), 9

S

Search (class in bb8.search), 9
SWCardSearch (class in bb8.cogs), 7
SWDestinyDBClient (class in bb8.swdestinydb), 9

T

text_line() (bb8.embed.CardDetail method), 8
type_line() (bb8.embed.CardDetail method), 8

U

UnsupportedFormat, 10