# spectrum.py Documentation

*Release 0.01*

treefroog

May 24, 2021

# Contents:

You can join the Spectrum API community at This Discord Server

CHAPTER 1

# API Reference

The following section outlines the API of spectrum.py.

## 1.1 Data Classes

**Warning:** Most of the objects have __slots__ defined. This allows the API to save space in RAM, but it makes it so that it is impossible to have dynamic attributes to the data classes. I am sorry. There is one object, *Object*, the does not have __slots__ defined. This is because it is made specifically so that dynamic attributes can be applied.

More information about `__slots__` can be found in the official python documentation.

### 1.1.1 Object

**class** spectrum.**Object**(*id*)

Represents a generic Spectrum object

Supported Operations:

| Operation | Description |
|---|---|
| x == y | Checks if two objects are equal. |
| x != y | Checks if two objects are not equal. |

This class is the base class of most objects, since most have an ID number.

**id** [int] The ID of the object

### 1.1.2 Message

**class** spectrum.**Message**(***kwargs*)

Represents a Spectrum message object

Supported Operations:

| Operation | Description |
|---|---|
| x == y | Checks if two messages are equal. |
| x != y | Checks if two messages are not equal. |
| str(x) | Returns the message's content. TODO |

**id** [int] The ID of the Message

**time_created** [`datetime.datetime`] The time that the message was created at

**time_modified** [`datetime.datetime`] The time that the message was modified. NOTE: May be the same as the time that it was created. This just means it has not been modified. Maybe changed in the future.

**lobby** [*`Lobby`*] The lobby that the message was sent in. I know the general plan of how to execute this, but since a lobby object isn't sent with the message (which makes sense) I will have to pull it from the cache when the message is sent. I have it in my head of how it will work. # TODO

**author** [*`Member`*] The author of the message. While this is a Member object, it is called author because it make more sense in context.

**content** [*`Content`*] The content object in the message.

**media_id** [str] This is in the format of embed:<embed ID here>. It shows the embed ID if the message has an embed. Can be `None` if there are no embeds

**highlight_role** [int or NotImplemented] The role that the member used for that message. It may return an int if the role was not in the clients cache, or it may return a `Role` object if it was in the clients cache.

**reactions** [NotImplemented] The reactions to the message. TODO

### 1.1.3 Community

**class** `spectrum.`**`Community`**`(`***kwargs*`)`

Represents a Spectrum Community object

Supported Operations:

| Operation | Description |
|---|---|
| x == y | Checks if two communities are equal. |
| x != y | Checks if two communities are not equal. |
| str(x) | Returns the community's name. |

**id** [int] The ID of the community

**type** [str] The type of community. Can be one of the following: (to my knowledge)

- Public
- Private

**slug** [str] The shorthand abbreviation of the community

**name** [str] The name of the community

**avatar** [str] This is a URL for the avatar of the community

**banner** [str] This is a URL for the banner of the community

**lobbies :** An iterator of *`Lobby`* objects that the community has.

**roles** [NotImplemented] A list of the `Role` that the community has. #TODO

### 1.1.4 Lobby

**class** `spectrum.`**`Lobby`**`(`***kwargs*`)`

Represents a Spectrum lobby object

Supported Operations:

| Operation | Description |
|---|---|
| x == y | Checks if two lobbies are equal. |
| x != y | Checks if two lobbies are not equal. |
| str(x) | Returns the lobbies name. |

**id** [int] The ID of the lobby

**name** [str] The name of the lobby

**description** [str] The description of the lobby

**color** [str] The color of the lobby in hex (May make a color object in the future) #TODO

**online_members_count** [int] The number of members online

**key** [str] The API subscription key used to connect to the lobby (May make a key object in the future) #TODO

**permissions** [NotImplemented] Not sure if this is the special permission that the client has in this lobby, or something else. #TODO

**community** [*`Community`*] The ID of the community that this lobby is a part of

### 1.1.5 Member

**class** `spectrum.`**`Member`**`(`***kwargs*`)`

Represents a Spectrum Member Object

Supported Operations:

| Operation | Description |
|---|---|
| x == y | Checks if two members are equal. |
| x != y | Checks if two members are not equal. |
| str(x) | Returns the member's name. |

**id** [int] The ID of the Member

**name** [str] The name chosen to be displayed publicly everywhere by the member

**handle** [str] The unique identifying name that the member chose

**avatar** [str] URL to the member's avatar

**presence** [*`Presence`*] The presence of the member.

**roles** [NotImplemented] The roles held by the member. TODO

### 1.1.6 Presence

**class** spectrum.**Presence**(***kwargs*)

Represents a Spectrum Presence object

Supported Operations:

| Operation | Description |
|---|---|
| str(x) | Returns the presence's status |

**status** [str] Returns the status of the member. Can be one of the following:

- away
- online
- do_not_disturb
- invisible

**info** [str/None] Returns a string of what the member is doing. If the member is playing Star Citizen it can be one of two things:

- "Playing Star Citizen" if they are currently in game
- "In Menus" if they have the game open, but are in in game

**since** [`datetime.datetime`] Return a datetime object from when this member's presence has changed

### 1.1.7 Content

**class** spectrum.**Content**(***kwargs*)

Represents a Spectrum content object

Supported Operations:

| Operation | Description |
|---|---|
| str(x) | Returns the content's raw_content. |

**lines** [int] The number of blocks that makes up the Content object

**blocks** [list] A `list` of *`Block`* objects that makes up the Content object. Each block represents a line in the message.

**raw_content** [str] The raw content string of the message. Every block (line) ends with a ' \n' before the next block (line) starts.

### 1.1.8 Emoji

**class** spectrum.**Emoji**(***kwargs*)

Represents a Spectrum Emoji object

Supported Operations:

| Operation | Description |
|---|---|
| str(x) | Returns the emoji's name. |

**mutability** [str] Displays the mutability of the emoji. I do not know what this means in context, or when they are ever mutable. I have an open prize of 3 free original tapirs to anyone who can solve this.

**emoji** [str] Displays the name in the form :<name>: just taken from the raw JSON. I do not know if I will ever do able to code and decode from Unicode. There is a large block of Javascript in the Javascript that runs Spectrum that converts between str and an object, but I do not think I can convert between Unicode and str.

**offset** [int] The offset of where the emoji resides within the block.

**length** [int] The length of string that the emoji takes up from the offset in the raw block content.

### 1.1.9 Link

**class** spectrum.**Link**(***kwargs*)

Represents a Spectrum Link object

Supported Operations:

| Operation | Description |
|---|---|
| str(x) | Returns the link's url. |

**mutability** [str] Displays the mutability of the link. I do not know what this means in context, or when they are ever mutable. I have an open prize of 3 free original tapirs to anyone who can solve this.

**url** [str] The url that the link object represents.

**offset** [int] The offset of where the link resides within the block.

**length** [int] The length of string that the link takes up from the offset in the raw block content.

### 1.1.10 Mention

**class** spectrum.**Mention**(*member=None*, ***kwargs*)

Represents a Spectrum mention object

Supported Operations:

| Operation | Description |
|---|---|
| str(x) | Returns the mention's user_id, or the member's name |

**mutability** [str] Displays the mutability of the mention. I do not know what this means in context, or when they are ever mutable. I have an open prize of 3 free original tapirs to anyone who can solve this.

**user_id** [int] Displays the id of the user mentioned

**member** [*Member* or None] This attribute represents the member that the mention mentions if they are in the client's cache. If not it defaults to None

**offset** [int] The offset of where the mention resides within the block.

**length** [int] The length of string that the mention takes up from the offset in the raw block content.

### 1.1.11 Style_range

**class** spectrum.**Style_range**(*style=None*, *offset=None*, *length=None*)

Represents a Spectrum inlineStyleRange object

Supported Operations:

| Operation | Description |
|---|---|
| str(x) | Returns the style_range's style. |

**style** [str] The style that this style_range represents

**offset** [int] The offset of the style represented by the style_range from the beginning of the block

**length** [int] The length of the style represented by the style_range starting from the offset

### 1.1.12 Block

**class** spectrum.**Block**(***kwargs*)

Represents a Spectrum block object

Supported Operations:

| Operation | Description |
|---|---|
| x == y | Checks if two blocks are equal. |
| x != y | Checks if two blocks are not equal. |
| str(x) | Returns the block's text. |

This object represents each individual line in a Spectrum message. For example if you press Shift+Enter to create a new line, this would be a new block in the message. Also each code block wrapper in ``` would be its own block.

**id** [str] The ID of the block.

> **Warning:** Most of the ID attributes for objects in Spectrum are an `int`, but this ID is a `str`. I do not know why

**text** [str] The raw text that makes up the block

**type** [str] Represents the type of text in the whole block Can be one of the following:

- unstyled
- code-block
- ???

**depth** [int] I do not know what this is. But it is here

**style_ranges :** A `list` of *`Style_range`* that are in the block

**entities :** A `list` of either *`Emoji`*, *`Mention`*, or *`Link`* that are in the block's content.

# CHAPTER 2

## Indices and tables

- genindex
- modindex
- search

# Index