
wyvern

Release 0.2.0

sarthhh

Mar 15, 2023

CONTENTS

1	API Reference	1
1.1	api	1
1.2	events	7
1.3	models	8
1.4	colors	17
1.5	utils	21
2	Getting started	23
2.1	First basic bot	23
	Python Module Index	25
	Index	27

API REFERENCE

1.1 api

1.1.1 bot

The class to connect your application with the Discord API.

class `wyvern.GatewayBot`(*token*: *str*, *, *api_version*: *int* = 10, *intents*: *SupportsInt* | *Intents* = 3178233)

Bases: [*ImplementsEventDecos*](#)

The main bot class that interacts with the discord API through both REST and gateway paths.

Parameters

- **token** (*str*) – The bot token to use while execution.
- **api_version** (*int*) – Discord API version in usage, defaults to 10.
- **intents** (*SupportsInt* | *Intents*) – Library's [*Intents*](#) builder or any object that returns the intent value when passed to *int*

Example

```
import asyncio

import wyvern

bot = wyvern.GatewayBot(
    "BOT_TOKEN_HERE",
    intents=(
        wyvern.Intents.GUILD_MEMBERS
        | wyvern.Intents.GUILDS
        | wyvern.Intents.GUILD_MESSAGES
        | wyvern.Intents.DIRECT_MESSAGES
    ),
)

asyncio.run(bot.start())
```

event_handler: [*EventHandler*](#)

The [*EventHandler*](#) attached to the instance.

intents: *Intents*

Intents being used by the bot.

listener(*event*: type[Event], *, *max_trigger*: int | Undefined = <wyvern.utils.consts.Undefined object>) → Callable[[types.EventListenerCallbackT], *EventListener*]

Adds a listener to the bot's event handler

Parameters

- **event** (type[Event]) – Type of the event.
- **max_trigger** (int) – Maximum number of times this event should be triggered.

Returns

listener – The event listener that was constructed.

Return type

EventListener

rest: *RESTClientImpl*

The REST handler attached to the instance.

async start(*raise_exception*: bool = False) → None

This function is a *coroutine*.

Verifies the bot token and starts listening to the gateway.

Parameters

raise_exception (bool) – Set to true if exceptions should be raised at this endpoint.

1.1.2 event_decos

class wyvern.api.event_decos.ImplementsEventDecos

Interface for event decorators in *GatewayBot*, this class is purely for the bot class to inherit from.

Example

```
@bot.on_started()
async def started(event: wyvern.StartedEvent) -> None:
    bot.logger.info(f"Logged in as {event.user.tag}")
```

on_started(**kwargs: Any) → Callable[[types.EventListenerCallbackT], handler.EventListener]

Used to create a *StartedEvent* listener.

Returns

The listener that was created.

Return type

EventListener

on_starting(**kwargs: Any) → Callable[[types.EventListenerCallbackT], handler.EventListener]

Used to create a *StartingEvent* listener.

Returns

The listener that was created.

Return type

EventListener

1.1.3 event_handler

```
class wyvern.api.event_handler.EventHandler(*, bot: GatewayBot, listeners: dict[type[Event],
list[EventListener]] = {})
```

Bases: `object`

Class handling dispatches and containing of the event listeners. An instance of this class is bound to the `GatewayBot` to handle events.

add_listener(*lsnr*: `EventListener`) → `None`

Adds a listener to the container.

Parameters

lsnr (`EventListener`) – The listener to add.

listeners: `dict[type[Event], list[EventListener]]`

Mapping of `Event` types to list of `EventListener` s listening to the event.

```
class wyvern.api.event_handler.EventListener(*, type: type[Event], max_trigger: int | Undefined =
<wyvern.utils.consts.Undefined object>, callback:
types.EventListenerCallbackT, bot: GatewayBot |
Undefined = <wyvern.utils.consts.Undefined object>)
```

Bases: `object`

Represents an event listener, the callback of this class gets triggered whenever the event this listener is bound to gets triggered.

callback: `types.EventListenerCallbackT`

The callback for listener.

max_trigger: `int | Undefined`

Maximum number of times this event can get triggered.

type: `type[Event]`

Type of event this listener listens to.

```
wyvern.api.event_handler.listener(event: type[Event], *, max_trigger: int | Undefined =
<wyvern.utils.consts.Undefined object>) →
Callable[[types.EventListenerCallbackT], EventListener]
```

Used to create an `EventListener`.

Parameters

- **event** (`type[Event]`) – Class of the event this listener is bound to.
- **max_trigger** (`int`) – Maximum trigger limit of the event callback.

Returns

The listener that was created.

Return type

`EventListener`

1.1.4 intents

Intents are bitwise value that are passed to the Discord API while creating a websocket connection to let the API know what events the connection is requesting for.

You can read more about intents in the [Official documentation](#).

final class wyvern.**Intents**(*value: int*)

Intents constructor to provide to the GatewayBot class.

value

The final value of intents generated by the constructor.

Type

int

ALL = 3211259

All intents.

AUTO_MODERATION_CONFIGURATION = 1048576

Required for these gateway events:

- AUTO_MODERATION_RULE_CREATE
- AUTO_MODERATION_RULE_UPDATE
- AUTO_MODERATION_RULE_DELETE

AUTO_MODERATION_EXECUTION = 2097152

Required for these gateway events:

- AUTO_MODERATION_ACTION_EXECUTION

DIRECT_MESSAGES = 4096

Required for these gateway events:

- MESSAGE_CREATE
- MESSAGE_UPDATE
- MESSAGE_DELETE
- CHANNEL_PINS_UPDATE

DIRECT_MESSAGE_REACTIONS = 8192

Required for these gateway events:

- MESSAGE_REACTION_ADD
- MESSAGE_REACTION_REMOVE
- MESSAGE_REACTION_REMOVE_ALL
- MESSAGE_REACTION_REMOVE_EMOJI

DIRECT_MESSAGE_TYPING = 16384

Required for these gateway events:

- TYPING_START

GUILDS = 1

Required for these gateway events:

- GUILD_CREATE

- GUILD_UPDATE
- GUILD_DELETE
- GUILD_ROLE_CREATE
- GUILD_ROLE_UPDATE
- GUILD_ROLE_DELETE
- CHANNEL_CREATE
- CHANNEL_UPDATE
- CHANNEL_DELETE
- CHANNEL_PINS_UPDATE
- THREAD_CREATE
- THREAD_UPDATE
- THREAD_DELETE
- THREAD_LIST_SYNC
- THREAD_MEMBER_UPDATE
- THREAD_MEMBERS_UPDATE
- STAGE_INSTANCE_CREATE
- STAGE_INSTANCE_UPDATE
- STAGE_INSTANCE_DELETE

GUILD_BANS = 4

Required for these gateway events:

- GUILD_BAN_ADD
- GUILD_BAN_REMOVE

GUILD_EMOJIS = 8

Required for these gateway events:

- GUILD_EMOJIS_UPDATE
- GUILD_STICKERS_UPDATE

GUILD_INTEGRATIONS = 16

Required for these gateway events:

- GUILD_INTEGRATIONS_UPDATE
- INTEGRATION_CREATE
- INTEGRATION_UPDATE
- INTEGRATION_DELETE

GUILD_INVITES = 64

Required for these gateway events:

- INVITE_CREATE
- INVITE_DELETE

GUILD_MEMBERS = 2

Required for these gateway events:

- GUILD_MEMBER_ADD
- GUILD_MEMBER_UPDATE
- GUILD_MEMBER_REMOVE
- THREAD_MEMBERS_UPDATE

Warning: This is a privileged intent.

GUILD_MESSAGES = 512

Required for these gateway events:

- MESSAGE_CREATE
- MESSAGE_UPDATE
- MESSAGE_DELETE
- MESSAGE_DELETE_BULK

GUILD_MESSAGE_REACTIONS = 1024

Required for these gateway events:

- MESSAGE_REACTION_ADD
- MESSAGE_REACTION_REMOVE
- MESSAGE_REACTION_REMOVE_ALL
- MESSAGE_REACTION_REMOVE_EMOJI

GUILD_MESSAGE_TYPING = 2048

Required for these gateway events:

- TYPING_START

GUILD_PRESENCES = 256

Required for these gateway events:

- PRESENCE_UPDATE

Warning: This is a privileged intent.

GUILD_SCHEDULED_EVENTS = 65536

Required for these gateway events:

- GUILD_SCHEDULED_EVENT_CREATE
- GUILD_SCHEDULED_EVENT_UPDATE
- GUILD_SCHEDULED_EVENT_DELETE
- GUILD_SCHEDULED_EVENT_USER_ADD
- GUILD_SCHEDULED_EVENT_USER_REMOVE

GUILD_VOICE_STATES = 128

Required for these gateway events:

- VOICE_STATE_UPDATE

GUILD_WEBHOOKS = 32

Required for these gateway events:

- WEBHOOKS_UPDATE

MESSAGE_CONTENT = 32768

Required for guild message's content.

Warning: This is a privileged intent.

NONE = 0

No intents.

PRIVILEGED = 33026

All privileged intents.

UNPRIVILEGED = 3178233

All unprivileged intents.

1.1.5 rest_client

```
class wyvern.api.rest_client.RESTClientImpl(*, token: str, bot: GatewayBot, api_version: int,
                                            client_session: aiohttp.ClientSession | Undefined =
                                            <wyvern.utils.consts.Undefined object>)
```

Bases: RESTClient

Class handling operations related to REST/HTTP requests to the discord API. An instance of this class is binded with the *GatewayBot* instance.

async fetch_current_user() → *BotUser*

This function is a *coroutine*.

Gets the current bot user.

Returns

The bot user.

Return type

BotUser

1.2 events

Events from discord are recieved in form of raw payloads and are converted into Event classes internally by the library.

1.2.1 base

class wyvern.events.base.Event(*, bot: GatewayBot)

The base event class. All library (or custom) events are derived from this class.

bot: GatewayBot

The bot bound to the event.

lib_events

class wyvern.events.lib_events.StartedEvent(*, bot: GatewayBot, user: BotUser)

Bases: Event

The StartedEvent is dispatched when the GatewayBot.start method is called and the bot receives first Websocket response from API after token verification.

Note: This event gets triggered only once in the runtime.

bot: GatewayBot

The bot bound to the event.

user: BotUser

The bot user that the client is logged in as.

class wyvern.events.lib_events.StartingEvent(*, bot: GatewayBot)

Bases: Event

This event is dispatched when the bot is about to connect to the gateway.

Note: This event gets triggered only once in the runtime.

bot: GatewayBot

The bot bound to the event.

1.3 models

1.3.1 Base Classes

class wyvern.Snowflake

Represents a discord ID, this class inherits from int.

class wyvern.DiscordObject

Base class for all discord snowflake models. Models like User inherit from this.

property created_at: datetime

Datetime at which the snowflake was created.

Returns

Creation time.

Return type

datetime.datetime

id: *Snowflake*

ID of the object.

components**class** wyvern.models.components.**ActionRow**(*, components: list[wyvern.models.components.Component/])Bases: *Component***class** wyvern.models.components.**Button**(*, style: ButtonStyle, label: str | None = None, emoji: Any = None, custom_id: str | None = None, disabled: bool = False, url: str | None = None)Bases: *Component*

Represents a discord button. The properties mentioned below can be used to create a button.

custom_id: str | None

Custom id for the component.

disabled: bool

True if the component is disabled.

emoji: Any

Emoji embedded in the button.

label: str | None

Button's label.

style: ButtonStyle

Style of the button.

url: str | None

The URL this button points to, if any.

class wyvern.models.components.ButtonStyle(value)Bases: *IntEnum*

Enums for Button style.

BLURPLE = 1

Alias for PRIMARY

DANGER = 4

Red discord button.

GRAY = 2

Alias for SECONDARY

GREEN = 3

Alias for SUCCESS

GREY = 2

Alias for SECONDARY

LINK = 5

Button pointing to an URL

PRIMARY = 1

A primary blurple discord button.

RED = 4

Alias for DANGER

SECONDARY = 2

A secondary gray discord button.

SUCCESS = 3

Green discord button.

URL = 5

Alias for LINK

class wyvern.models.components.**Component**

Bases: [ABC](#)

Represents a discord component. Is the base class for other components.

class wyvern.models.components.**ComponentType**(*value*)

Bases: [IntEnum](#)

An enumeration.

class wyvern.models.components.**Modal**(**, title: str, custom_id: str, text_inputs: list[wyvern.models.components.TextInput]*)

Bases: [object](#)

Represents a discord modal form.

custom_id: str

Custom ID of the modal

text_inputs: list[wyvern.models.components.TextInput]

List of textinputs in the modal.

title: str

Title of the modal.

final class wyvern.models.components.**TextInput**(**, custom_id: str, label: str, style: TextInputStyle, min_length: int | None = None, max_length: int | None = None, required: bool = True, default_value: str | None = None, placeholder: str | None = None*)

Bases: [Component](#)

Represents a modal text-input.

custom_id: str

Custom id of the textinput.

default_value: str | None

Default value for this textinput.

label: str

Label of the textinput.

max_length: int | None

The minimum allowed length for textinput.

min_length: `int | None`

The maximum allowed length for textinput.

placeholder: `str | None`

The placeholder used, if any.

required: `bool`

Whether the field is required.

style: `TextInputStyle`

Textinput style.

class `wyvern.models.components.TextInputStyle(value)`

Bases: `IntEnum`

An enumeration.

PARAGRAPH = 2

For multi line text-inputs.

SHORT = 1

For single line text-inputs.

members

class `wyvern.models.members.GuildMember(*, id: Snowflake, username: str, discriminator: str, avatar_hash: str | None, is_bot: bool, is_system: bool, is_mfa_enabled: bool, banner_hash: str | None, accent_color: int | None, locale: str | None, flags_value: int | None, premium_type_value: int | None, public_flags_value: int | None, bot: GatewayBot, partial_user: Undefined | PartialUser = <wyvern.utils.consts.Undefined object>, raw: discord typings.GuildMemberData, user: User, guild_id: Snowflake, nickname: str | None, role_ids: list[Snowflake], joined_at: datetime.datetime, premium_since: datetime.datetime | None, deaf: bool | None, mute: bool | None, pending: bool, communication_disabled_until: datetime.datetime | None, guild_avatar_hash: str | None)`

Bases: `User`

accent_color: `int | None`

User's accent color.

avatar_hash: `str | None`

Hash of the user's avatar

banner_hash: `str | None`

User's banner hash.

bot: `GatewayBot`

The current bot application.

communication_disabled_until: `datetime.datetime | None`

Timeout for the member.

property created_at: `datetime.datetime`

returns: The datetime when this user was created. :rtype: datetime.datetime

deaf: `bool` | `None`
True if member is deafened.

discriminator: `str`
User's discriminator.

property display_name: `str`
returns: Nickname of the user if exists, else the username. :rtype: str

flags_value: `int` | `None`
Integer value for user flags

guild_avatar_hash: `str` | `None`
Avatar hash for the guild.

guild_id: `Snowflake`
ID of the guild this member belongs to.

id: `Snowflake`
ID of the user.

is_bot: `bool`
True if the user is a bot.

is_mfa_enabled: `bool`
True if the user has mfa enabled (can be false even if it's enabled)

is_system: `bool`
True if the user is a system user.

joined_at: `datetime.datetime`
Datetime on which the member joined the guild.

locale: `str` | `None`
User's locale.

mute: `bool` | `None`
True if member is muted.

nickname: `str` | `None`
Nickname of the member, if any.

partial_user: `Undefined` | `PartialUser`
The partial user this class was constructed from, if any.

pending: `bool`
True if member is in pending state.

premium_since: `datetime.datetime` | `None`
Datetime since member has been boosting the guild.

premium_type_value: `int` | `None`
Premium type integer.

public_flags_value: `int` | `None`
Integer value for user's public flags

raw: `discord_typings.GuildMemberData`
Raw payload.

role_ids: `list[Snowflake]`
List of role ids.

property tag: `str`
Return user tag (username#discriminator)

user: `User`
User of the member.

username: `str`
Username of the user.

messages

class `wyvern.models.messages.AllowedMentions`(*, *users: bool = False, roles: bool = False, everyone: bool = False, replied_user: bool = False*)

Bases: `object`

Allowed mentions in a message sent from bot.

Parameters

- **users** (*bool*) – Set to *True* to mention users.
- **roles** (*bool*) – Set to *True* to mention roles.
- **everyone** (*bool*) – Set to *True* to allow everyone/here mentions.
- **replied_user** (*bool*) – Weather the replied users should be pinged.

class `wyvern.models.messages.PartialMessage`(*, *id: Snowflake*)

Bases: `DiscordObject`

property created_at: `datetime`

Datetime at which the snowflake was created.

Returns

Creation time.

Return type

`datetime.datetime`

id: `Snowflake`

ID of the message.

users

class `wyvern.models.users.BotUser`(*, *raw: discord_typings.UserData, id: Snowflake, username: str, discriminator: str, avatar_hash: str | None, is_bot: bool, is_system: bool, is_mfa_enabled: bool, banner_hash: str | None, accent_color: int | None, locale: str | None, flags_value: int | None, premium_type_value: int | None, public_flags_value: int | None, bot: GatewayBot, partial_user: Undefined | PartialUser = <wyvern.utils.consts.Undefined object>*)

Bases: `User`

accent_color: `int | None`

User's accent color.

avatar_hash: `str | None`

Hash of the user's avatar

banner_hash: `str | None`

User's banner hash.

bot: `GatewayBot`

The current bot application.

property created_at: `datetime.datetime`

returns: The datetime when this user was created. :rtype: datetime.datetime

discriminator: `str`

User's discriminator.

flags_value: `int | None`

Integer value for user flags

id: `Snowflake`

ID of the user.

is_bot: `bool`

True if the user is a bot.

is_mfa_enabled: `bool`

True if the user has mfa enabled (can be false even if it's enabled)

is_system: `bool`

True if the user is a system user.

locale: `str | None`

User's locale.

partial_user: `Undefined | PartialUser`

The partial user this class was constructed from, if any.

premium_type_value: `int | None`

Premium type integer.

public_flags_value: `int | None`

Integer value for user's public flags

raw: `discord_typings.UserData`

The raw data from discord this object was constructed from.

property tag: `str`

Return user tag (username#discriminator)

username: `str`

Username of the user.

```
class wyvern.models.users.PartialUser(*, raw: discord_typings.UserData, id: Snowflake, username: str,
    discriminator: str, avatar_hash: str | None, is_bot: bool,
    is_system: bool, is_mfa_enabled: bool, banner_hash: str | None,
    accent_color: int | None, locale: str | None, flags_value: int | None,
    premium_type_value: int | None, public_flags_value: int | None)
```

Bases: *UserLike*

Object representing a user object.

accent_color: `int | None`

User's accent color.

avatar_hash: `str | None`

Hash of the user's avatar

banner_hash: `str | None`

User's banner hash.

property created_at: `datetime.datetime`

returns: The datetime when this user was created. :rtype: datetime.datetime

discriminator: `str`

User's discriminator.

flags_value: `int | None`

Integer value for user flags

id: *Snowflake*

ID of the user.

is_bot: `bool`

True if the user is a bot.

is_mfa_enabled: `bool`

True if the user has mfa enabled (can be false even if it's enabled)

is_system: `bool`

True if the user is a system user.

locale: `str | None`

User's locale.

premium_type_value: `int | None`

Premium type integer.

public_flags_value: `int | None`

Integer value for user's public flags

raw: `discord_typings.UserData`

The raw data from discord this object was constructed from.

property tag: `str`

Return user tag (username#discriminator)

username: `str`

Username of the user.

```
class wyvern.models.users.User(*, raw: discord_typings.UserData, id: Snowflake, username: str,
                               discriminator: str, avatar_hash: str | None, is_bot: bool, is_system: bool,
                               is_mfa_enabled: bool, banner_hash: str | None, accent_color: int | None,
                               locale: str | None, flags_value: int | None, premium_type_value: int | None,
                               public_flags_value: int | None, bot: GatewayBot, partial_user: Undefined |
                               PartialUser = <wyvern.utils.consts.Undefined object>)
```

Bases: *PartialUser*, *ImplementsMessage*

accent_color: *int* | *None*

User's accent color.

avatar_hash: *str* | *None*

Hash of the user's avatar

banner_hash: *str* | *None*

User's banner hash.

bot: *GatewayBot*

The current bot application.

property created_at: *datetime.datetime*

returns: The datetime when this user was created. :rtype: datetime.datetime

discriminator: *str*

User's discriminator.

flags_value: *int* | *None*

Integer value for user flags

id: *Snowflake*

ID of the user.

is_bot: *bool*

True if the user is a bot.

is_mfa_enabled: *bool*

True if the user has mfa enabled (can be false even if it's enabled)

is_system: *bool*

True if the user is a system user.

locale: *str* | *None*

User's locale.

partial_user: *Undefined* | *PartialUser*

The partial user this class was constructed from, if any.

premium_type_value: *int* | *None*

Premium type integer.

public_flags_value: *int* | *None*

Integer value for user's public flags

raw: *discord_typings.UserData*

The raw data from discord this object was constructed from.

property tag: *str*

Return user tag (username#discriminator)

username: *str*

Username of the user.

```
class wyvern.models.users.UserLike(*, id: Snowflake, username: str, discriminator: str)
```

Bases: *DiscordObject*

Object representing a user like entity (user, member, thread member, etc...).

```
property created_at: datetime.datetime
```

returns: The datetime when this user was created. :rtype: datetime.datetime

```
discriminator: str
```

Discriminator of the user.

```
id: Snowflake
```

ID of the particular user.

```
property tag: str
```

Return user tag (username#discriminator)

```
username: str
```

Username of the user.

1.4 colors

```
final class wyvern.colors.Color(value: int)
```

Class representing a color in the RGB color space. Alias name Colour exists for convenience.

value

The value of the color. This is a 24-bit integer, where the first 8 bits are the red value, the next 8 bits are the green value, and the last 8 bits are the blue value.

Type

int

```
classmethod aqua() → Color
```

Creates a Color object from the aqua color. This is #00ffff. (Aqua)

```
property b: int
```

The blue value of the color.

```
classmethod black() → Color
```

Creates a Color object from the black color. This is 0x000000. (Black)

```
classmethod blue() → Color
```

Creates a Color object from the blue color. This is 0x0000ff. (Blue)

```
classmethod brown() → Color
```

Creates a Color object from the brown color. This is #a52a2a. (Brown)

```
classmethod cyan() → Color
```

Creates a Color object from the cyan color. This is 0x00ffff. (Cyan)

```
classmethod default() → Color
```

Creates a Color object from the default color. This is 0x000000. (Black)

Examples

```
>>> Color.default()
Color(0)
```

classmethod `from_hex(hex_value: str) → Color`

Creates a Color object from a hex value.

Parameters

hex_value (*str*) – The hex value to use.

Returns

A Color object.

Return type

Color

Examples

```
>>> Color.from_hex('#ff0000')
Color(16776960)
>>> Color.from_hex('#00ff00')
Color(255)
>>> Color.from_hex('#0000ff')
Color(0)
```

classmethod `from_hsl(h: float, s: float, l: float) → Color`

Creates a Color object from HSL values.

Parameters

- **h** (*float*) – The hue value.
- **s** (*float*) – The saturation value.
- **l** (*float*) – The lightness value.

Returns

A Color object.

Return type

Color

Examples

```
>>> Color.from_hsl(0, 1, 0.5)
Color(16711680)
>>> Color.from_hsl(120, 1, 0.5)
Color(16711680)
>>> Color.from_hsl(240, 1, 0.5)
Color(16711680)
```

classmethod `from_hsv(h: float, s: float, v: float) → Color`

Creates a Color object from HSV values.

Parameters

- **h** (*float*) – The hue value.
- **s** (*float*) – The saturation value.
- **v** (*float*) – The value in HSV color space.

Returns

A Color object.

Return type

Color

Examples

```
>>> Color.from_hsv(0, 1, 1)
Color(16711680)
>>> Color.from_hsv(120, 1, 1)
Color(16711680)
>>> Color.from_hsv(240, 1, 1)
Color(16711680)
```

classmethod `from_random()` → *Color*

Creates a Color object from a random color. Randomly generates a color in the RGB color space.

Returns

A Color object.

Return type

Color

classmethod `from_rgb(r: int, g: int, b: int)` → *Color*

Creates a Color object from RGB values.

Parameters

- **r** (*int*) – The red value.
- **g** (*int*) – The green value.
- **b** (*int*) – The blue value.

Returns

A Color object.

Return type

Color

Examples

```
>>> Color.from_rgb(255, 0, 0)
Color(16711680)
>>> Color.from_rgb(0, 255, 0)
Color(65280)
>>> Color.from_rgb(0, 0, 255)
Color(255)
```

classmethod `from_string(string: str) → Color`

Creates a Color object from a string.

Parameters

string (*str*) – The string to use.

Returns

A Color object.

Return type

Color

Examples

```
>>> Color.from_string('rgb(255, 0, 0)')
Color(16711680)
>>> Color.from_string('hsl(0, 100%, 50%)')
Color(-80727249750)
>>> Color.from_string('hsv(0, 100%, 100%)')
Color(1022371500)
>>> Color.from_string('#ff0000')
Color(16776960)
```

property `g: int`

The green value of the color.

classmethod `gray() → Color`

Creates a Color object from the gray color. This is `0x808080`. (Gray)

classmethod `green() → Color`

Creates a Color object from the green color. This is `0x00ff00`. (Green)

classmethod `grey() → Color`

Creates a Color object from the grey color. This is `0x808080`. (Grey)

property `hex: str`

The hex value of the color.

property `hsl: tuple[float, float, float]`

The HSL values of the color.

property `hsv: tuple[float, float, float]`

The HSV values of the color.

classmethod `magenta() → Color`

Creates a Color object from the magenta color. This is `0xff00ff`. (Magenta)

classmethod `orange() → Color`

Creates a Color object from the orange color. This is `0xffa500`. (Orange)

classmethod `purple() → Color`

Creates a Color object from the purple color. This is `#800080`. (Purple)

property `r: int`

The red value of the color.

classmethod `red()` → *Color*

Creates a Color object from the red color. This is `0xff0000`. (Red)

property `rgb`: `tuple[int, int, int]`

The RGB values of the color.

classmethod `silver()` → *Color*

Creates a Color object from the silver color. This is `#c0c0c0`. (Silver)

classmethod `white()` → *Color*

Creates a Color object from the white color. This is `0xffffffff`. (White)

classmethod `yellow()` → *Color*

Creates a Color object from the yellow color. This is `0xffff00`. (Yellow)

`wyvern.colors.Colour`

An alias for the *Color* class.

1.5 utils

1.5.1 Constants

`wyvern.utils.consts.EMPTY` = `<wyvern.utils.consts.Empty object>`

Instance of Empty.

class `wyvern.utils.consts.Empty`

This class is used for variables in an embed constructor to represent empty values.

`wyvern.utils.consts.NULL` = `<wyvern.utils.consts.Null object>`

Instance of Null.

class `wyvern.utils.consts.Null`

Class to simulate difference between an argument provided as None v/s a *null* value to send to the discord API.
This class is used as a placeholder for arguments in REST methods.

`wyvern.utils.consts.UNDEFINED` = `<wyvern.utils.consts.Undefined object>`

Instance of Undefined.

class `wyvern.utils.consts.Undefined`

An instance of this class is used as a placeholder to represent attributes and variables which are not defined *yet*.

GETTING STARTED

2.1 First basic bot

2.1.1 Creating a bot instance

```
import asyncio
import os

import wyvern

bot = wyvern.GatewayBot(os.environ["TOKEN"], intents=wyvern.Intents.UNPRIVILEGED)
```

2.1.2 Listening to events

```
@bot.listener(wyvern.StartingEvent)
async def starting(event: wyvern.StartingEvent) -> None:
    bot.logger.info("Starting bot...")

# or using the implemented decorators

@bot.on_started()
async def started(event: wyvern.StartedEvent) -> None:
    bot.logger.info(f"Logged in as {event.user.tag}")
```

2.1.3 Running the bot

```
async def main():
    async with bot:
        await bot.start()

asyncio.run(main())
```


PYTHON MODULE INDEX

W

- `wyvern.api.event_decos`, 2
- `wyvern.api.event_handler`, 3
- `wyvern.api.rest_client`, 7
- `wyvern.colors`, 17
- `wyvern.events.base`, 8
- `wyvern.events.lib_events`, 8
- `wyvern.models.components`, 9
- `wyvern.models.members`, 11
- `wyvern.models.messages`, 13
- `wyvern.models.users`, 13
- `wyvern.utils.consts`, 21

A

accent_color (wyvern.models.members.GuildMember attribute), 11
 accent_color (wyvern.models.users.BotUser attribute), 13
 accent_color (wyvern.models.users.PartialUser attribute), 15
 accent_color (wyvern.models.users.User attribute), 16
 ActionRow (class in wyvern.models.components), 9
 add_listener() (wyvern.api.event_handler.EventHandler method), 3
 ALL (wyvern.Intent attribute), 4
 AllowedMentions (class in wyvern.models.messages), 13
 aqua() (wyvern.colors.Color class method), 17
 AUTO_MODERATION_CONFIGURATION (wyvern.Intent attribute), 4
 AUTO_MODERATION_EXECUTION (wyvern.Intent attribute), 4
 avatar_hash (wyvern.models.members.GuildMember attribute), 11
 avatar_hash (wyvern.models.users.BotUser attribute), 14
 avatar_hash (wyvern.models.users.PartialUser attribute), 15
 avatar_hash (wyvern.models.users.User attribute), 16

B

b (wyvern.colors.Color property), 17
 banner_hash (wyvern.models.members.GuildMember attribute), 11
 banner_hash (wyvern.models.users.BotUser attribute), 14
 banner_hash (wyvern.models.users.PartialUser attribute), 15
 banner_hash (wyvern.models.users.User attribute), 16
 black() (wyvern.colors.Color class method), 17
 blue() (wyvern.colors.Color class method), 17
 BLURPLE (wyvern.models.components.ButtonStyle attribute), 9
 bot (wyvern.events.base.Event attribute), 8
 bot (wyvern.events.lib_events.StartedEvent attribute), 8

bot (wyvern.events.lib_events.StartingEvent attribute), 8
 bot (wyvern.models.members.GuildMember attribute), 11
 bot (wyvern.models.users.BotUser attribute), 14
 bot (wyvern.models.users.User attribute), 16
 BotUser (class in wyvern.models.users), 13
 brown() (wyvern.colors.Color class method), 17
 Button (class in wyvern.models.components), 9
 ButtonStyle (class in wyvern.models.components), 9

C

callback (wyvern.api.event_handler.EventListener attribute), 3
 Color (class in wyvern.colors), 17
 Colour (in module wyvern.colors), 21
 communication_disabled_until (wyvern.models.members.GuildMember attribute), 11
 Component (class in wyvern.models.components), 10
 ComponentType (class in wyvern.models.components), 10
 created_at (wyvern.DiscordObject property), 8
 created_at (wyvern.models.members.GuildMember property), 11
 created_at (wyvern.models.messages.PartialMessage property), 13
 created_at (wyvern.models.users.BotUser property), 14
 created_at (wyvern.models.users.PartialUser property), 15
 created_at (wyvern.models.users.User property), 16
 created_at (wyvern.models.users.UserLike property), 17
 custom_id (wyvern.models.components.Button attribute), 9
 custom_id (wyvern.models.components.Modal attribute), 10
 custom_id (wyvern.models.components.TextInput attribute), 10
 cyan() (wyvern.colors.Color class method), 17

D

DANGER (wyvern.models.components.ButtonStyle attribute), 9

deaf (wyvern.models.members.GuildMember attribute), 11

default() (wyvern.colors.Color class method), 17

default_value (wyvern.models.components.TextInput attribute), 10

DIRECT_MESSAGE_REACTIONS (wyvern.Intent attribute), 4

DIRECT_MESSAGE_TYPING (wyvern.Intent attribute), 4

DIRECT_MESSAGES (wyvern.Intent attribute), 4

disabled (wyvern.models.components.Button attribute), 9

DiscordObject (class in wyvern), 8

discriminator (wyvern.models.members.GuildMember attribute), 12

discriminator (wyvern.models.users.BotUser attribute), 14

discriminator (wyvern.models.users.PartialUser attribute), 15

discriminator (wyvern.models.users.User attribute), 16

discriminator (wyvern.models.users.UserLike attribute), 17

display_name (wyvern.models.members.GuildMember property), 12

E

emoji (wyvern.models.components.Button attribute), 9

Empty (class in wyvern.utils.consts), 21

EMPTY (in module wyvern.utils.consts), 21

Event (class in wyvern.events.base), 8

event_handler (wyvern.GatewayBot attribute), 1

EventHandler (class in wyvern.api.event_handler), 3

EventListener (class in wyvern.api.event_handler), 3

F

fetch_current_user() (wyvern.api.rest_client.RESTClientImpl method), 7

flags_value (wyvern.models.members.GuildMember attribute), 12

flags_value (wyvern.models.users.BotUser attribute), 14

flags_value (wyvern.models.users.PartialUser attribute), 15

flags_value (wyvern.models.users.User attribute), 16

from_hex() (wyvern.colors.Color class method), 18

from_hsl() (wyvern.colors.Color class method), 18

from_hsv() (wyvern.colors.Color class method), 18

from_random() (wyvern.colors.Color class method), 19

from_rgb() (wyvern.colors.Color class method), 19

from_string() (wyvern.colors.Color class method), 19

G

g (wyvern.colors.Color property), 20

GatewayBot (class in wyvern), 1

GRAY (wyvern.models.components.ButtonStyle attribute), 9

gray() (wyvern.colors.Color class method), 20

GREEN (wyvern.models.components.ButtonStyle attribute), 9

green() (wyvern.colors.Color class method), 20

GREY (wyvern.models.components.ButtonStyle attribute), 9

grey() (wyvern.colors.Color class method), 20

guild_avatar_hash (wyvern.models.members.GuildMember attribute), 12

GUILD_BANS (wyvern.Intent attribute), 5

GUILD_EMOJIS (wyvern.Intent attribute), 5

guild_id (wyvern.models.members.GuildMember attribute), 12

GUILD_INTEGRATIONS (wyvern.Intent attribute), 5

GUILD_INVITES (wyvern.Intent attribute), 5

GUILD_MEMBERS (wyvern.Intent attribute), 5

GUILD_MESSAGE_REACTIONS (wyvern.Intent attribute), 6

GUILD_MESSAGE_TYPING (wyvern.Intent attribute), 6

GUILD_MESSAGES (wyvern.Intent attribute), 6

GUILD_PRESENCES (wyvern.Intent attribute), 6

GUILD_SCHEDULED_EVENTS (wyvern.Intent attribute), 6

GUILD_VOICE_STATES (wyvern.Intent attribute), 6

GUILD_WEBHOOKS (wyvern.Intent attribute), 7

GuildMember (class in wyvern.models.members), 11

GUILDS (wyvern.Intent attribute), 4

H

hex (wyvern.colors.Color property), 20

hsl (wyvern.colors.Color property), 20

hsv (wyvern.colors.Color property), 20

I

id (wyvern.DiscordObject attribute), 9

id (wyvern.models.members.GuildMember attribute), 12

id (wyvern.models.messages.PartialMessage attribute), 13

id (wyvern.models.users.BotUser attribute), 14

id (wyvern.models.users.PartialUser attribute), 15

id (wyvern.models.users.User attribute), 16

id (wyvern.models.users.UserLike attribute), 17

ImplementsEventDecos (class in wyvern.api.event_decos), 2

Intents (class in wyvern), 4

intents (wyvern.GatewayBot attribute), 1

is_bot (wyvern.models.members.GuildMember attribute), 12

[is_bot \(wyvern.models.users.BotUser attribute\)](#), 14
[is_bot \(wyvern.models.users.PartialUser attribute\)](#), 15
[is_bot \(wyvern.models.users.User attribute\)](#), 16
[is_mfa_enabled \(wyvern.models.members.GuildMember attribute\)](#), 12
[is_mfa_enabled \(wyvern.models.users.BotUser attribute\)](#), 14
[is_mfa_enabled \(wyvern.models.users.PartialUser attribute\)](#), 15
[is_mfa_enabled \(wyvern.models.users.User attribute\)](#), 16
[is_system \(wyvern.models.members.GuildMember attribute\)](#), 12
[is_system \(wyvern.models.users.BotUser attribute\)](#), 14
[is_system \(wyvern.models.users.PartialUser attribute\)](#), 15
[is_system \(wyvern.models.users.User attribute\)](#), 16

J

[joined_at \(wyvern.models.members.GuildMember attribute\)](#), 12

L

[label \(wyvern.models.components.Button attribute\)](#), 9
[label \(wyvern.models.components.TextInput attribute\)](#), 10
[LINK \(wyvern.models.components.ButtonStyle attribute\)](#), 9
[listener\(\) \(in module wyvern.api.event_handler\)](#), 3
[listener\(\) \(wyvern.GatewayBot method\)](#), 2
[listeners \(wyvern.api.event_handler.EventHandler attribute\)](#), 3
[locale \(wyvern.models.members.GuildMember attribute\)](#), 12
[locale \(wyvern.models.users.BotUser attribute\)](#), 14
[locale \(wyvern.models.users.PartialUser attribute\)](#), 15
[locale \(wyvern.models.users.User attribute\)](#), 16

M

[magenta\(\) \(wyvern.colors.Color class method\)](#), 20
[max_length \(wyvern.models.components.TextInput attribute\)](#), 10
[max_trigger \(wyvern.api.event_handler.EventListener attribute\)](#), 3
[MESSAGE_CONTENT \(wyvern.Intent attribute\)](#), 7
[min_length \(wyvern.models.components.TextInput attribute\)](#), 10
[Modal \(class in wyvern.models.components\)](#), 10
[module](#)

- [wyvern.api.event_decos](#), 2
- [wyvern.api.event_handler](#), 3
- [wyvern.api.rest_client](#), 7
- [wyvern.colors](#), 17
- [wyvern.events.base](#), 8

[wyvern.events.lib_events](#), 8
[wyvern.models.components](#), 9
[wyvern.models.members](#), 11
[wyvern.models.messages](#), 13
[wyvern.models.users](#), 13
[wyvern.utils.consts](#), 21
[mute \(wyvern.models.members.GuildMember attribute\)](#), 12

N

[nickname \(wyvern.models.members.GuildMember attribute\)](#), 12
[NONE \(wyvern.Intent attribute\)](#), 7
[Null \(class in wyvern.utils.consts\)](#), 21
[NULL \(in module wyvern.utils.consts\)](#), 21

O

[on_started\(\) \(wyvern.api.event_decos.ImplementsEventDecos method\)](#), 2
[on_starting\(\) \(wyvern.api.event_decos.ImplementsEventDecos method\)](#), 2
[orange\(\) \(wyvern.colors.Color class method\)](#), 20

P

[PARAGRAPH \(wyvern.models.components.TextInputStyle attribute\)](#), 11
[partial_user \(wyvern.models.members.GuildMember attribute\)](#), 12
[partial_user \(wyvern.models.users.BotUser attribute\)](#), 14
[partial_user \(wyvern.models.users.User attribute\)](#), 16
[PartialMessage \(class in wyvern.models.messages\)](#), 13
[PartialUser \(class in wyvern.models.users\)](#), 14
[pending \(wyvern.models.members.GuildMember attribute\)](#), 12
[placeholder \(wyvern.models.components.TextInput attribute\)](#), 11
[premium_since \(wyvern.models.members.GuildMember attribute\)](#), 12
[premium_type_value \(wyvern.models.members.GuildMember attribute\)](#), 12
[premium_type_value \(wyvern.models.users.BotUser attribute\)](#), 14
[premium_type_value \(wyvern.models.users.PartialUser attribute\)](#), 15
[premium_type_value \(wyvern.models.users.User attribute\)](#), 16
[PRIMARY \(wyvern.models.components.ButtonStyle attribute\)](#), 9
[PRIVILEGED \(wyvern.Intent attribute\)](#), 7
[public_flags_value \(wyvern.models.members.GuildMember attribute\)](#), 12
[public_flags_value \(wyvern.models.users.BotUser attribute\)](#), 14

`public_flags_value` (*wyvern.models.users.PartialUser attribute*), 15

`public_flags_value` (*wyvern.models.users.User attribute*), 16

`purple()` (*wyvern.colors.Color class method*), 20

R

`r` (*wyvern.colors.Color property*), 20

`raw` (*wyvern.models.members.GuildMember attribute*), 12

`raw` (*wyvern.models.users.BotUser attribute*), 14

`raw` (*wyvern.models.users.PartialUser attribute*), 15

`raw` (*wyvern.models.users.User attribute*), 16

`RED` (*wyvern.models.components.ButtonStyle attribute*), 10

`red()` (*wyvern.colors.Color class method*), 20

`required` (*wyvern.models.components.TextInput attribute*), 11

`rest` (*wyvern.GatewayBot attribute*), 2

`RESTClientImpl` (*class in wyvern.api.rest_client*), 7

`rgb` (*wyvern.colors.Color property*), 21

`role_ids` (*wyvern.models.members.GuildMember attribute*), 13

S

`SECONDARY` (*wyvern.models.components.ButtonStyle attribute*), 10

`SHORT` (*wyvern.models.components.TextInputStyle attribute*), 11

`silver()` (*wyvern.colors.Color class method*), 21

`Snowflake` (*class in wyvern*), 8

`start()` (*wyvern.GatewayBot method*), 2

`StartedEvent` (*class in wyvern.events.lib_events*), 8

`StartingEvent` (*class in wyvern.events.lib_events*), 8

`style` (*wyvern.models.components.Button attribute*), 9

`style` (*wyvern.models.components.TextInput attribute*), 11

`SUCCESS` (*wyvern.models.components.ButtonStyle attribute*), 10

T

`tag` (*wyvern.models.members.GuildMember property*), 13

`tag` (*wyvern.models.users.BotUser property*), 14

`tag` (*wyvern.models.users.PartialUser property*), 15

`tag` (*wyvern.models.users.User property*), 16

`tag` (*wyvern.models.users.UserLike property*), 17

`text_inputs` (*wyvern.models.components.Modal attribute*), 10

`TextInput` (*class in wyvern.models.components*), 10

`TextInputStyle` (*class in wyvern.models.components*), 11

`title` (*wyvern.models.components.Modal attribute*), 10

`type` (*wyvern.api.event_handler.EventListener attribute*), 3

U

`Undefined` (*class in wyvern.utils.consts*), 21

`UNDEFINED` (*in module wyvern.utils.consts*), 21

`UNPRIVILEGED` (*wyvern.Intent attribute*), 7

`url` (*wyvern.models.components.Button attribute*), 9

`URL` (*wyvern.models.components.ButtonStyle attribute*), 10

`User` (*class in wyvern.models.users*), 15

`user` (*wyvern.events.lib_events.StartedEvent attribute*), 8

`user` (*wyvern.models.members.GuildMember attribute*), 13

`UserLike` (*class in wyvern.models.users*), 16

`username` (*wyvern.models.members.GuildMember attribute*), 13

`username` (*wyvern.models.users.BotUser attribute*), 14

`username` (*wyvern.models.users.PartialUser attribute*), 15

`username` (*wyvern.models.users.User attribute*), 16

`username` (*wyvern.models.users.UserLike attribute*), 17

V

`value` (*wyvern.colors.Color attribute*), 17

`value` (*wyvern.Intent attribute*), 4

W

`white()` (*wyvern.colors.Color class method*), 21

`wyvern.api.event_decos`
module, 2

`wyvern.api.event_handler`
module, 3

`wyvern.api.rest_client`
module, 7

`wyvern.colors`
module, 17

`wyvern.events.base`
module, 8

`wyvern.events.lib_events`
module, 8

`wyvern.models.components`
module, 9

`wyvern.models.members`
module, 11

`wyvern.models.messages`
module, 13

`wyvern.models.users`
module, 13

`wyvern.utils.consts`
module, 21

Y

`yellow()` (*wyvern.colors.Color class method*), 21