

---

# **python-bale-bot**

***Release 2.4.7.2***

**Kian Ahmadian**

**Oct 22, 2023**



REFERENCE

- 1 What is Bale? 3
- 2 What is python-bale-bot? 5
- 3 Installing 7
  - 3.1 PyPi: . . . . . 7
  - 3.2 Git: . . . . . 7
- 4 Documentation 9
- 5 Contact to Developers 11
  - 5.1 bale package . . . . . 11
  - 5.2 Helpers . . . . . 39
  - 5.3 Change Log . . . . . 39
  - 5.4 Examples . . . . . 42
- Index 45



An API wrapper for Bale written in Python.



## WHAT IS BALE?

**The “Bale” is a messenger-platform for send and receive messages.** it's provides services for developers and they can send or receive messages through bots like normal users and These services are provided by [web services](#) (API).





## WHAT IS PYTHON-BALE-BOT?

The “python-bale-bot” is a Python language package optimized for developers to use web services provided by “Bale”.



## INSTALLING

You can install or update ``python-bale-bot`` via:

### 3.1 PyPi:

```
$ pip install python-bale-bot -U
```

### 3.2 Git:

```
$ git clone https://github.com/python-bale-bot/python-bale-bot
$ cd python-bale-bot
$ python setup.py install
```



## DOCUMENTATION

The `package documentation` is the technical reference for `python-bale-bot`. It contains descriptions of all available classes, modules, methods and arguments as well as the changelog.



## CONTACT TO DEVELOPERS

### 5.1 bale package

#### 5.1.1 Classes in this package

##### Bot

**class** `bale.Bot`(*token*, *\*\*kwargs*)

Bases: `object`

This object represents a Bale Bot.

##### Parameters

**token** (*str*) – Bot Token

**Attention:** When you create bot and run for first-step, use `bale.Bot.delete_webhook()` method in `on_before_ready` event.

**add\_event**(*event\_name*, *wrapper*)

Set wrapper or listener for an event.

##### Parameters

- **event\_name** (*str*) – Name of the event
- **wrapper** (*Callable*[]) – Function to add as wrapper for event

**async ban\_chat\_member**(*chat\_id*, *user\_id*)

Use this method to ban a user from a group, supergroup or a channel. In the case of supergroups and channels, the user will not be able to return to the group on their own using invite links, etc., unless unbanned first.

##### Parameters

- **chat\_id** (*str* | *int*) – Unique identifier for the target chat or username of the target channel (in the format @channelusername).
- **user\_id** (*int* | *str*) – Unique identifier of the target user.

##### Returns

On success, `True` is returned.

**Return type**`bool`**Raises**

- **NotFound** – Invalid Chat or User ID.
- **Forbidden** – You do not have permission to ban Chat Member.
- **APIError** – ban chat member Failed.

**async close()**

Close http Events and bot

**async delete\_message(chat\_id, message\_id)**

You can use this service to delete a message that you have already sent through the arm.

**Warning:****In Channel or Group:**

If it is a group or channel Manager, it can delete a message from (group or channel).

**In private message (PV):**

If the message was sent by a bot, it can be deleted with this method

**Parameters**

- **chat\_id**(`str` | `int`) – Unique identifier for the target chat or username of the target channel (in the format @channelusername).
- **message\_id**(`bale.Message`) – Unique identifier for the message to delete.

**Raises**

- **NotFound** – Invalid Message or Chat ID.
- **Forbidden** – You do not have permission to Delete Message.
- **APIError** – Delete Message Failed.

**async delete\_webhook()**

This service is used to remove the webhook set for the bot.

**Returns**

On success, True is returned.

**Return type**`bool`**Raises**

- **Forbidden** – You do not have permission to delete Webhook.
- **APIError** – Delete webhook Failed.

**async edit\_message(chat\_id, message\_id, text, \*, components=None)**

You can use this service to edit text messages that you have already sent through the arm.

**Parameters**

- **chat\_id**(`str` | `int`) – Unique identifier for the target chat or username of the target channel (in the format @channelusername).



- **message\_id** (*str* | *int*) – Unique identifier for the message to edit.
- **text** (*str*) – New text of the message, 1- 4096 characters after entities parsing.
- **components** (Optional[*bale.Components* | *bale.RemoveComponents*]) – An object for an inline keyboard.

**Raises**

- **NotFound** – Invalid Message or Chat ID.
- **Forbidden** – You do not have permission to Edit Message.
- **APIError** – Edit Message Failed.

**event**(*coro*)

Set wrapper or listener for selected event (the name of function).

---

**Hint:** The name of the function for which you write the decorator is considered the name of the event.

---

**async forward\_message**(*chat\_id*, *from\_chat\_id*, *message\_id*)

This service is used to send text messages.

**Parameters**

- **chat\_id** (*str* | *int*) – Unique identifier for the target chat or username of the target channel (in the format @channelusername).
- **from\_chat\_id** (*str* | *int*) – the chat where the original message was sent (or channel username in the format @channelusername).
- **message\_id** (*int* | *str*) – Message in the chat specified in *from\_chat\_id*.

**Returns**

The Message

**Return type**

*bale.Message*

**Raises**

- **NotFound** – Invalid Chat ID.
- **Forbidden** – You do not have permission to send Message to this chat.
- **APIError** – Forward Message Failed.

**async get\_bot**()

Get bot information

**Returns**

Bot User information.

**Return type**

*bale.User*

**Raises**

**APIError** – Get bot Failed.

**async get\_chat**(*chat\_id*)

Use this method to get up-to-date information about the chat (current name of the user for one-on-one conversations, current username of a user, group or channel, etc.).

**Parameters**

**chat\_id**(*int* / *str*) – Unique identifier for the target chat or username of the target channel (in the format @channelusername).

**Returns**

The chat or None if not found.

**Return type**

Optional[*bale.Chat*]

**Raises**

- **Forbidden** – You do not have permission to get Chat.
- **APIError** – Get chat Failed.

**async get\_chat\_administrators**(*chat\_id*)

Use this method to get a list of administrators in a chat.

**Parameters**

**chat\_id**(*str* | *int*) – Unique identifier for the target chat or username of the target channel (in the format @channelusername).

**Returns**

list of chat member.

**Return type**

List[*bale.ChatMember*]

**Raises**

- **NotFound** – Invalid Chat ID.
- **Forbidden** – You do not have permission to get Administrators of the Chat.
- **APIError** – get Administrators of the Chat from chat Failed.

**async get\_chat\_member**(*chat\_id*, *user\_id*)

Use this method to get information about a member of a chat. The method is only guaranteed to work for other users if the bot is an administrator in the chat.

**Parameters**

- **chat\_id**(*str* | *int*) – Unique identifier for the target chat or username of the target channel (in the format @channelusername).
- **user\_id** (Optional[*int* | *str*]) – Unique identifier of the target user.

**Returns**

The chat member of None if not found.

**Return type**

Optional[*bale.ChatMember*]

**Raises**

- **NotFound** – Invalid Chat or User ID.
- **Forbidden** – You do not have permission to get Chat Member.
- **APIError** – Get chat member Failed.

**async get\_chat\_members\_count**(*chat\_id*)

Use this method to get the number of members in a chat.

**Parameters**

**chat\_id** (`str` | `int`) – Unique identifier for the target chat or username of the target channel (in the format @channelusername).

**Raises**

- **NotFound** – Invalid Chat ID.
- **Forbidden** – You do not have permission to get Members count of the Chat.
- **APIError** – get Members count of the Chat Failed.

**Returns**

The members count of the chat

**Return type**

`int`

**async get\_file**(*file\_id*)

Use this method to get basic info about a file and prepare it for downloading. For the moment, bots can download files of up to 20 MB in size.

**Parameters**

**file\_id** (`str`) – Either the file identifier to get file information about.

**Returns**

The content of the file

**Return type**

`bytes`

**Raises**

- **NotFound** – Invalid file ID.
- **Forbidden** – You do not have permission to download File.
- **APIError** – download File Failed.

**async get\_user**(*user\_id*)

This Method almost like `bale.Bot.get_chat`, but this a filter that only get user.

**Parameters**

**user\_id** (`int`) – user id

**Returns**

The user or None if not found.

**Return type**

Optional[`bale.User`]

**Raises**

- **Forbidden** – You do not have permission to get User.
- **APIError** – Get user Failed.

**async invite\_user**(*chat\_id*, *user\_id*)

Invite user to the chat

**Parameters**

- **chat\_id** (`str` | `int`) – Unique identifier for the target chat or username of the target channel (in the format @channelusername).
- **user\_id** (`str` | `int`) – Unique identifier for the target user.

**Raises**

- **NotFound** – Invalid Chat or User ID.
- **Forbidden** – You do not have permission to Add user to Chat.
- **APIError** – Invite user Failed.

**is\_closed()**

**bool:** Connection Status

**async leave\_chat(chat\_id)**

Use this method for your bot to leave a group, channel.

**Parameters**

**chat\_id** (**str** | **int**) – Unique identifier for the target chat or username of the target channel (in the format @channelusername).

**Raises**

- **Forbidden** – You do not have permission to Leave from chat.
- **APIError** – Leave from chat Failed.

**listen(event\_name)**

Set wrapper or listener for selected event (custom function name).

**Parameters**

**event\_name** (**str**) – Name of the event to set.

**async on\_error(event\_name, error)**

a Event for get errors when exceptions

**run()**

Starting the bot, updater and HTTPClient.

**async send\_audio(chat\_id, audio, \*, caption=None, components=None, reply\_to\_message\_id=None)**

This service is used to send Audio.

**Parameters**

- **chat\_id** (**str** | **int**) – Unique identifier for the target chat or username of the target channel (in the format @channelusername).
- **audio** (*bale.InputFile*) – File to send. visit *bale.InputFile* to see more info.
- **caption** (Optional[**str**]) – Audio caption.
- **components** (Optional[*bale.Components* | *bale.RemoveComponents*]) – Message Components
- **reply\_to\_message\_id** (Optional[**str** | **int**]) – If the message is a reply, ID of the original message.

**Returns**

The Message.

**Return type**

*bale.Message*

**Raises**

- **NotFound** – Invalid Chat ID.
- **Forbidden** – You do not have permission to Send Audio to chat.

- **APIError** – Send Audio Failed.

**async send\_contact**(*chat\_id, contact*)

This service is used to send contact.

#### Parameters

- **chat\_id** (*str* | *int*) – Unique identifier for the target chat or username of the target channel (in the format @channelusername).
- **contact** (*bale.ContactMessage*) – The Contact.

#### Returns

The Message.

#### Return type

*bale.Message*

#### Raises

- **NotFound** – Invalid Chat ID.
- **Forbidden** – You do not have permission to send Contact Message to this chat.
- **APIError** – Send Contact Message Failed.

**async send\_document**(*chat\_id, document, \*, caption=None, components=None, reply\_to\_message\_id=None*)

This service is used to send document.

#### Parameters

- **chat\_id** (*str* | *int*) – Unique identifier for the target chat or username of the target channel (in the format @channelusername).
- **document** (*bale.InputFile*) – File to send. visit *bale.InputFile* to see more info.
- **caption** (Optional[*str*]) – Document caption.
- **components** (Optional[*bale.Components* | *bale.RemoveComponents*]) – Message Components
- **reply\_to\_message\_id** (Optional[*str* | *int*]) – If the message is a reply, ID of the original message.

#### Returns

The Message.

#### Return type

*bale.Message*

#### Raises

- **NotFound** – Invalid Chat ID.
- **Forbidden** – You do not have permission to send Document to this chat.
- **APIError** – Send Document Failed.

**async send\_invoice**(*chat\_id, title, description, provider\_token, prices, \*, payload=None, photo\_url=None, need\_name=False, need\_phone\_number=False, need\_email=False, need\_shipping\_address=False, is\_flexible=True*)

You can use this service to send money request messages.

---

**Important:** When paying the amount, a fee will be charged from the sender.

---

### Parameters

- **chat\_id** (*str* | *int*) – Unique identifier for the target chat or username of the target channel (in the format @channelusername).
- **title** (*str*) – Product name. 1- 32 characters.
- **description** (*str*) – Product description. 1- 255 characters.
- **provider\_token** (*str*) – You can use 3 methods to receive money: 1.Card number 2. Port number and acceptor number 3. Wallet number “Bale”
- **prices** (List[*bale.Price*]) – A list of prices.
- **payload** (Optional[*str*]) – Bot-defined invoice payload. This will not be displayed to the user, use for your internal processes.
- **photo\_url** (Optional[*str*]) – URL of the product photo for the invoice. Can be a photo of the goods or a marketing image for a service. People like it better when they see what they are paying for.
- **need\_name** (Optional[*bool*]) – Pass True, if you require the user’s full name to complete the order.
- **need\_phone\_number** (Optional[*bool*]) – Pass True, if you require the user’s phone number to complete the order.
- **need\_email** (Optional[*bool*]) – Pass True, if you require the user’s email to complete the order.
- **need\_shipping\_address** (Optional[*bool*]) – Pass True, if you require the user’s shipping address to complete the order.
- **is\_flexible** (Optional[*bool*]) – Pass True, if the final price depends on the shipping method.

### Return type

*bale.Message*

### Raises

- **NotFound** – Invalid Chat ID.
- **Forbidden** – You do not have permission to send Invoice to this chat.
- **APIError** – Send Invoice Failed.

**async send\_location**(*chat\_id*, *location*)

Use this method to send point on the map.

### Parameters

- **chat\_id** (*str* | *int*) – Unique identifier for the target chat or username of the target channel (in the format @channelusername).
- **location** (*bale.Location*) – The Location.

### Returns

The Message.

**Return type***bale.Message***Raises**

- **NotFound** – Invalid Chat ID.
- **Forbidden** – You do not have permission to send Location to this chat.
- **APIError** – Send Location Failed.

**async send\_message**(*chat\_id, text, \*, components=None, reply\_to\_message\_id=None*)

This service is used to send text messages.

**Parameters**

- **chat\_id** (*str* | *int*) – Unique identifier for the target chat or username of the target channel (in the format @channelusername).
- **text** (*str*) – Text of the message to be sent. Max 4096 characters after entities parsing.
- **components** (Optional[*bale.Components* | *bale.RemoveComponents*]) – Message Components
- **reply\_to\_message\_id** (Optional[*str* | *int*]) – If the message is a reply, ID of the original message.

**Returns**

The Message

**Return type***bale.Message***Raises**

- **NotFound** – Invalid Chat ID.
- **Forbidden** – You do not have permission to send Message to this chat.
- **APIError** – Send Message Failed.

**async send\_photo**(*chat\_id, photo, \*, caption=None, components=None, reply\_to\_message\_id=None*)

This service is used to send photo.

**Parameters**

- **chat\_id** (*str* | *int*) – Unique identifier for the target chat or username of the target channel (in the format @channelusername).
- **photo** (*bale.InputFile*) – File to send. visit *bale.InputFile* to see more info.
- **caption** (Optional[*str*]) – Photo caption.
- **components** (Optional[*bale.Components* | *bale.RemoveComponents*]) – Message Components
- **reply\_to\_message\_id** (Optional[*str* | *int*]) – If the message is a reply, ID of the original message.

**Returns**

The Message.

**Return type***bale.Message***Raises**

- **NotFound** – Invalid Chat ID.
- **Forbidden** – You do not have permission to Send Photo to chat.
- **APIError** – Send photo Failed.

**async send\_video**(*chat\_id*, *video*, \*, *caption=None*, *components=None*, *reply\_to\_message\_id=None*)

This service is used to send Video.

#### Parameters

- **chat\_id**(*str* | *int*) – Unique identifier for the target chat or username of the target channel (in the format @channelusername).
- **video** (*bale.InputFile*) – File to send. visit [bale.InputFile](#) to see more info.
- **caption** (Optional[*str*]) – Video caption.
- **components** (Optional[*bale.Components* | *bale.RemoveComponents*]) – Message Components
- **reply\_to\_message\_id** (Optional[*str* | *int*]) – If the message is a reply, ID of the original message.

#### Returns

The Message.

#### Return type

[bale.Message](#)

#### Raises

- **NotFound** – Invalid Chat ID.
- **Forbidden** – You do not have permission to Send Video to chat.
- **APIError** – Send Video Failed.

**async set\_webhook**(*url*)

Use this method to specify an url and receive incoming updates via an outgoing webhook. Whenever there is an update for the bot, Bale will send an HTTPS POST request to the specified url, containing An Update. In case of an unsuccessful request, Bale will give up after a reasonable amount of attempts.

#### Parameters

**url** (*str*) – HTTPS url to send updates to. Use an empty string to remove webhook integration.

#### Returns

On success, True is returned.

#### Return type

*bool*

**property user**

Represents the connected client. None if not logged in

#### Type

Optional[[bale.User](#)]

**wait\_for**(*event\_name*, \*, *check=None*, *timeout=None*)

Waits for an event to be dispatched.

This could be used to wait for a user to reply to a message, or send a photo, or to edit a message in a self-contained way. The timeout parameter is passed onto `asyncio.wait_for()`. By default, it does not timeout.



Note that this does propagate the `asyncio.TimeoutError` for you in case of timeout and is provided for ease of use. In case the event returns multiple arguments, a tuple containing those arguments is returned instead. This function returns the first event that meets the requirements.

#### Parameters

- **event\_name** (`str`) – Name of the event
- **check** (Optional[Callable[`...`, `bool`]]) – A predicate to check what to wait for. The arguments must meet the parameters of the event being waited for.
- **timeout** (Optional[`float`]) – The number of seconds to wait before timing out and raising `asyncio.TimeoutError`.

#### Raises

**`asyncio.TimeoutError`** – If a timeout is provided, and it was reached.

## Available Types

### Message

```
class bale.Message(message_id, date, text=None, caption=None, forward_from=None,
                    forward_from_chat=None, forward_from_message_id=None, from_user=None,
                    document=None, contact=None, location=None, chat=None, video=None, photos=None,
                    reply_to_message=None, invoice=None, audio=None, successful_payment=None,
                    bot=None, **options)
```

Bases: `object`

This object shows a message.

#### **message\_id**

Unique message identifier inside this chat.

#### Type

`str`

#### **from\_user**

Sender of the message; empty for messages sent to channels. For backward compatibility, this will contain a fake sender user in non-channel chats, if the message was sent on behalf of a chat.

#### Type

Optional[`bale.User`]

#### **chat**

Conversation the message belongs to.

#### Type

`bale.Chat`

#### **date**

Date the message was sent in Unix time.

#### Type

`datetime.datetime`

#### **text**

Message Content

#### Type

Optional[`str`]

**caption**

Caption for the animation, audio, document, photo, video or voice.

**Type**

Optional[[str](#)]

**forward\_from**

For forwarded messages, sender of the original message.

**Type**

Optional[[bale.User](#)]

**forward\_from\_chat**

For messages forwarded from channels or from anonymous administrators, information about the original sender chat.

**Type**

Optional[[bale.Chat](#)]

**reply\_to\_message**

For replies, the original message. Note that the Message object in this field will not contain further reply\_to\_message fields even if it itself is a reply.

**Type**

Optional[[bale.Message](#)]

**contact**

Message is a shared contact, information about the contact.

**Type**

Optional[[bale.ContactMessage](#)]

**location**

Message is a shared location, information about the location.

**Type**

Optional[[bale.Location](#)]

**document**

Message is a general file, information about the file.

**Type**

Optional[[bale.Document](#)]

**video**

Message is a video, information about the video.

**Type**

Optional[[bale.Video](#)]

**audio**

Message is an audio, information about the Audio.

**Type**

Optional[[bale.Audio](#)]

**new\_chat\_members**

New members that were added to the group or supergroup and information about them (the bot itself may be one of these members). This list is empty if the message does not contain new chat members.

**Type**Optional[List[[bale.User](#)]]**left\_chat\_member**

A member was removed from the group, information about them (this member may be the bot itself).

**Type**Optional[[bale.User](#)]**invoice**

Message is an invoice for a payment, information about the invoice.

**Type**Optional[[bale.Invoice](#)]**successful\_payment**

Message is a service message about a successful payment, information about the payment.

**Type**Optional[[bale.SuccessfulPayment](#)]**property attachment**

Represents the message attachment. None if the message don't have any attachments

**Type**Optional[[bale.File](#)]**property author**An alias for [from\\_user](#)**property chat\_id**

Represents the Unique identifier of Conversation the message belongs to.

**Type**[str](#) | [int](#)**property content**

Represents the message content. None if the message don't have text or caption

**Type**Optional[[str](#)]**async delete()**For the documentation of the arguments, please see [bale.Bot.delete\\_message\(\)](#).**async edit(text, \*, components=None)**For the documentation of the arguments, please see [bale.Bot.edit\\_message\(\)](#)**async forward(chat\_id)**For the documentation of the arguments, please see [bale.Bot.forward\\_message\(\)](#).**async reply(text, \*, components=None)**For the documentation of the arguments, please see [bale.Bot.send\\_message\(\)](#).**async reply\_audio(audio, \*, caption=None, components=None)**For the documentation of the arguments, please see [bale.Bot.send\\_audio\(\)](#).**async reply\_document(document, \*, caption=None, components=None)**For the documentation of the arguments, please see [bale.Bot.send\\_document\(\)](#).

**async reply\_photo**(*photo*, \*, *caption=None*, *components=None*)

For the documentation of the arguments, please see [`bale.Bot.send\_photo\(\)`](#).

**property reply\_to\_message\_id**

Represents the Unique identifier of Original message, if the message has been replied. None If the message is not replied

**Type**

Optional[[`str`](#)]

**async reply\_video**(*video*, \*, *caption=None*, *components=None*)

For the documentation of the arguments, please see [`bale.Bot.send\_video\(\)`](#).

## Chat

**class bale.Chat**(*chat\_id*, *type*, *title=None*, *username=None*, *first\_name=None*, *last\_name=None*, *pinned\_message=None*, *all\_members\_are\_administrators=None*, *invite\_link=None*, *bot=None*)

Bases: [`object`](#)

This object indicates a chat.

**chat\_id**

Unique identifier for this chat.

**Type**

[`str`](#)

**type**

Type of chat.

**Type**

[`str`](#)

**title**

Title, for channels and group chats.

**Type**

Optional[[`str`](#)]

**username**

Username, for private chats, supergroups and channels if available.

**Type**

Optional[[`str`](#)]

**first\_name**

First name of the other party in a private chat.

**Type**

Optional[[`str`](#)]

**last\_name**

Last name of the other party in a private chat.

**Type**

Optional[[`str`](#)]

**pinned\_message**

Pinned messages in chat. Defaults to None.

**Type**

Optional[[bale.Message](#)]

**invite\_link**

Primary invite link, for groups and channel. Returned only in [bale.Bot.get\\_chat\(\)](#).

**Type**

Optional[[str](#)]

**all\_members\_are\_administrators**

Does everyone have admin access?. Defaults to True. (for Group)

**Type**

[bool](#)

**async add\_user(*user*)**

For the documentation of the arguments, please see [bale.Bot.invite\\_user\(\)](#).

**async ban\_chat\_member(*user*)**

For the documentation of the arguments, please see [bale.Bot.ban\\_chat\\_member\(\)](#).

**async get\_chat\_administrators()**

For the documentation of the arguments, please see [bale.Bot.get\\_chat\\_administrators\(\)](#).

**async get\_chat\_member(*user*)**

For the documentation of the arguments, please see [bale.Bot.get\\_chat\\_member\(\)](#).

**async get\_chat\_members\_count()**

For the documentation of the arguments, please see [bale.Bot.get\\_chat\\_members\\_count\(\)](#).

**async leave()**

For the documentation of the method, please see [bale.Bot.leave\\_chat\(\)](#).

**property parsed\_type**

Represents the parsed type of chat.

**Type**

[bale.ChatType](#)

**async send(*text*, *components=None*)**

For the documentation of the arguments, please see [bale.Bot.send\\_message\(\)](#).

**async send\_audio(*audio*, \*, *caption=None*, *components=None*)**

For the documentation of the arguments, please see [bale.Bot.send\\_audio\(\)](#).

**async send\_contact(*contact*)**

For the documentation of the arguments, please see [bale.Bot.send\\_contact\(\)](#).

**async send\_document(*document*, \*, *caption=None*, *components=None*)**

For the documentation of the arguments, please see [bale.Bot.send\\_document\(\)](#).

**async send\_invoice(*title*, *description*, *provider\_token*, *prices*, \*, *payload=None*, *photo\_url=None*, *need\_name=False*, *need\_phone\_number=False*, *need\_email=False*, *need\_shipping\_address=False*, *is\_flexible=True*)**

For the documentation of the arguments, please see [bale.Bot.send\\_invoice\(\)](#)

**async send\_location**(*location*)

For the documentation of the arguments, please see [bale.Bot.send\\_location\(\)](#).

**async send\_photo**(*photo*, \*, *caption=None*, *components=None*)

For the documentation of the arguments, please see [bale.Bot.send\\_photo\(\)](#).

**async send\_video**(*video*, \*, *caption=None*, *components=None*)

For the documentation of the arguments, please see [bale.Bot.send\\_video\(\)](#).

## User

**class** `bale.User`(*user\_id*, *is\_bot*, *first\_name*, *last\_name=None*, *username=None*, *bot=None*)

Bases: `object`

This object represents a Bale user or bot.

**user\_id**

Unique identifier for this user or bot.

**Type**

`int`

**is\_bot**

True, if this user is a bot.

**Type**

`bool`

**first\_name**

User's or bot's first name.

**Type**

`str`

**last\_name**

User's or bot's last name.

**Type**

Optional[`str`]

**username**

User's or bot's username.

**Type**

Optional[`str`]

**property chat\_id**

Aliases for [user\\_id](#)

**property mention**

mention user with username.

**Type**

Optional[`str`]

**async send**(*text*, *components=None*)

For the documentation of the arguments, please see [bale.Bot.send\\_message\(\)](#).

**async send\_audio**(audio, \*, caption=None, components=None)

For the documentation of the arguments, please see [bale.Bot.send\\_audio\(\)](#).

**async send\_contact**(contact)

For the documentation of the arguments, please see [bale.Bot.send\\_contact\(\)](#).

**async send\_document**(document, \*, caption=None, components=None)

For the documentation of the arguments, please see [bale.Bot.send\\_document\(\)](#).

**async send\_invoice**(title, description, provider\_token, prices, \*, payload=None, photo\_url=None, need\_name=False, need\_phone\_number=False, need\_email=False, need\_shipping\_address=False, is\_flexible=True)

For the documentation of the arguments, please see [bale.Bot.send\\_invoice\(\)](#)

**async send\_location**(location)

For the documentation of the arguments, please see [bale.Bot.send\\_location\(\)](#).

**async send\_photo**(photo, \*, caption=None, components=None)

For the documentation of the arguments, please see [bale.Bot.send\\_photo\(\)](#).

**async send\_video**(video, \*, caption=None, components=None)

For the documentation of the arguments, please see [bale.Bot.send\\_video\(\)](#).

## Callback Query

**class bale.CallbackQuery**(callback\_id, data=None, message=None, inline\_message\_id=None, from\_user=None, bot=None)

Bases: `object`

This object represents an incoming callback query from a callback button in an inline keyboard.

**callback\_id**

Unique identifier for this query.

Type

`str`

**from\_user**

Sender.

Type

`bale.User`

**message**

Message with the callback button that originated the query. Note that message content and message date will not be available if the message is too old.

Type

`bale.Message`

**inline\_message\_id**

Identifier of the message sent via the bot in inline mode, that originated the query.

Type

`str`

**data**

Data associated with the callback button. Be aware that the message, which originated the query, can contain no callback buttons with this data.

**Type**

`str`

**property user**

Aliases for `from_user`

## Chat Member

**class** `bale.ChatMember(chat_id, status, user, permissions, bot)`

Bases: `object`

This object shows a member in chat

**user**

Information about the user.

**Type**

`bale.User`

**status**

The member's status in the chat.

**Type**

`str`

**permissions**

The member's permissions in the chat.

**Type**

`bale.Permissions`

**async ban()**

For the documentation of the arguments, please see `bale.Bot.ban_chat_member()`.

**property parsed\_status**

Represents the parsed member's status.

**Type**

`bale.ChatMemberStatus`

## Permissions

**class** `bale.Permissions(can_be_edited=False, can_change_info=False, can_post_messages=False, can_edit_messages=False, can_delete_messages=False, can_invite_users=False, can_restrict_members=False, can_pin_messages=False, can_promote_members=False, can_send_messages=False, can_send_media_messages=False)`

Bases: `object`

This object shows the permissions and permissions of an admin or a member in a group (or channel).

**Parameters**



- **can\_be\_edited** (*bool*) – Can you edit?. Defaults to False.
- **can\_change\_info** (*bool*) – Can you edit group information? Defaults to False.
- **can\_post\_messages** (*bool*) – Can he post a message?. Defaults to False.
- **can\_edit\_messages** (*bool*) – Can you edit your message? Defaults to False.
- **can\_delete\_messages** (*bool*) – Can it erase messages? Defaults to False.
- **can\_invite\_users** (*bool*) – Can it invite users to chat? Defaults to False.
- **can\_restrict\_members** (*bool*) – Defaults to False.
- **can\_pin\_messages** (*bool*) – Can you pin your message? Defaults to False.
- **can\_promote\_members** (*bool*) – Defaults to False.
- **can\_send\_messages** (*bool*) – Can he send a message?. Defaults to False.
- **can\_send\_media\_messages** (*bool*) – Can it attach a file with the message? Defaults to False.

**classmethod** **from\_dict**(*data*)

**Parameters**

**data** (*dict*) – Data

## Update

**class** **bale.Update**(*update\_id, type, callback\_query=None, message=None, edited\_message=None, bot=None*)

Bases: **object**

This object represents an incoming update.

**update\_id**

The update's unique identifier. Update identifiers start from a certain positive number and increase sequentially. This ID becomes especially handy if you're using Webhooks, since it allows you to ignore repeated updates or to restore the correct update sequence, should they get out of order. If there are no new updates for at least a week, then identifier of the next update will be chosen randomly instead of sequentially.

**Type**

*int*

**callback\_query**

New incoming callback query.

**Type**

Optional[*bale.CallbackQuery*]

**message**

New incoming message of any kind - text, photo, sticker, etc.

**Type**

Optional[*bale.Message*]

**edited\_message**

New version of a message that is known to the bot and was edited.

**Type**

Optional[*bale.Message*]

## UI

### Components

**class** `bale.Components`

Bases: `object`

This object shows a Component.

**add\_inline\_keyboard**(*inline\_keyboard*, *row=1*)

Use this method to add InlineKeyboard component.

**Parameters**

- **inline\_keyboard** (`bale.InlineKeyboard`) – The Component you want to add
- **row** (`int`) – The Component row number

**add\_menu\_keyboard**(*menu\_keyboard*, *row=1*)

Use this method to add MenuKeyboard component.

**Parameters**

- **menu\_keyboard** (`bale.MenuKeyboard`) – The Component you want to add
- **row** (`int`) – The Component row number

**property inline\_keyboards**

Represents the InlineKeyboards list.

**Type**

List[`bale.InlineKeyboard`]

**property menu\_keyboards**

Represents the MenuKeyboards list.

**Type**

List[`bale.MenuKeyboard`]

**remove\_inline\_keyboard**(*inline\_keyboard*, *row=1*)

Use this method to remove InlineKeyboard component.

**Parameters**

- **inline\_keyboard** (`bale.InlineKeyboard`) – The Component you want to remove
- **row** (`int`) – The Component row number

**remove\_menu\_keyboard**(*menu\_keyboard*, *row=1*)

Use this method to remove MenuKeyboard component.

**Parameters**

- **menu\_keyboard** (`bale.MenuKeyboard`) – The Component you want to remove
- **row** (`int`) – The Component row number

## Inline Keyboard

```
class bale.InlineKeyboard(text, *, callback_data=None, url=None, switch_inline_query=None,
                           switch_inline_query_current_chat=None)
```

Bases: `object`

This object shows an inline keyboard (within the message).

### **text**

Label text on the button.

#### **Type**

`str`

### **callback\_data**

If set, pressing the button will prompt the user to select one of their chats, open that chat and insert the bot's username and the specified inline query in the input field. Can be empty, in which case just the bot's username will be inserted. Defaults to None.

#### **Type**

Optional[`str`]

### **url**

HTTP url to be opened when the button is pressed. Defaults to None.

#### **Type**

Optional[`str`]

### **switch\_inline\_query**

If set, pressing the button will prompt the user to select one of their chats, open that chat and insert the bot's username and the specified inline query in the input field. Can be empty, in which case just the bot's username will be inserted. Defaults to None.

#### **Type**

Optional[`str`]

### **switch\_inline\_query\_current\_chat**

If set, pressing the button will insert the bot's username and the specified inline query in the current chat's input field. Can be empty, in which case only the bot's username will be inserted. Defaults to None.

#### **Type**

Optional[`str`]

## Menu Keyboard

```
class bale.MenuKeyboard(text, *, request_contact=False, request_location=False)
```

Bases: `object`

This object shows a Menu keyboard

### **text**

Keyboard Text.

#### **Type**

`str`

**request\_contact**

If True, the user's phone number will be sent as a contact when the button is pressed.

**Type**

Optional[bool]

**request\_location**

If True, the user's current location will be sent when the button is pressed. Available in private chats only.

**Type**

Optional[bool]

## Payments

### Invoice

**class** bale.Invoice(*title, description, start\_parameter, currency, total\_amount*)

Bases: object

This object shows Invoice

**title**

Product name.

**Type**

str

**description**

Product description.

**Type**

str

**start\_parameter**

Unique bot deep-linking parameter that can be used to generate this invoice.

**Type**

str

**currency**

Three-letter ISO 4217 currency code.

**Type**

str

**total\_amount**

Total price in the smallest units of the currency (integer, not float/double).

**Type**

int

## Price

**class** `bale.Price`(*label=None, amount=None*)

Bases: `object`

This object shows a Price

**label**

Label Price.

**Type**

Optional[`str`]

**amount**

Amount Price.

**Type**

Optional[`int`]

## Successful Payment

**class** `bale.SuccessfulPayment`(*currency, total\_amount, invoice\_payload=None, shipping\_option\_id=None*)

Bases: `object`

This object contains basic information about a successful payment.

**currency**

The currency in which the transaction was made.

**Type**

`str`

**total\_amount**

The total sum of the transaction amount.

**Type**

`int`

**invoice\_payload**

Bot specified invoice payload.

**Type**

Optional[`str`]

**shipping\_option\_id**

Identifier of the shipping option chosen by the user.

**Type**

Optional[`str`]

**property** `payload`

Aliases for `invoice_payload`

## Attachments

### Audio

**class** bale.**Audio**(*file\_id*, *duration=None*, *file\_size=None*, *bot=None*, *mime\_type=None*, *title=None*)

Bases: *File*

This object shows an Audio.

**file\_id**

Identifier for this file, which can be used to download or reuse the file.

**Type**

*str*

**duration**

Duration of the audio in seconds as defined by sender.

**Type**

*int*

**file\_size**

File size in bytes.

**Type**

*int*

**mime\_type**

MIME type of the file as defined by sender.

**Type**

Optional[*str*]

**title**

Title of the audio as defined by sender or by audio tags.

**Type**

Optional[*str*]

### Contact

**class** bale.**ContactMessage**(*phone\_number*, *first\_name=None*, *last\_name=None*)

Bases: *object*

This object shows a Message Contact.

**phone\_number**

**Type**

*int*

**first\_name**

**Type**

*str*

**last\_name**

**Type**

Optional[*str*]

## Document

**class** bale.Document(*file\_id*, *file\_name=None*, *mime\_type=None*, *file\_size=None*, *bot=None*)

Bases: [File](#)

This object shows a Document.

**file\_id**

Identifier for this file, which can be used to download or reuse the file.

**Type**

[str](#)

**file\_name**

Original filename as defined by sender.

**Type**

Optional[[str](#)]

**mime\_type**

MIME type of the file as defined by sender.

**Type**

Optional[[str](#)]

**file\_size**

File size in bytes.

**Type**

Optional[[int](#)]

## File

**class** bale.File(*file\_type*, *file\_id*, *file\_size*, *mime\_type*, *bot*, *\*\*kwargs*)

Bases: [object](#)

This object shows a Base File Class.

**file\_type**

Type of the file.

**Type**

[str](#)

**file\_id**

Identifier for this file, which can be used to download or reuse the file.

**Type**

[str](#)

**file\_size**

File size in bytes.

**Type**

Optional[[int](#)]

**mime\_type**

MIME type of the file as defined by sender.

**Type**

Optional[[str](#)]

**extra**

The rest of the file information.

**Type**

Optional[[dict](#)]

---

**Note:** You can get more information from the file with extra.

---

**property base\_file**

Represents the Base File Class of this file

**Type**

[bale.File](#)

**async get()**

For the documentation of the arguments, please see [bale.Bot.get\\_file\(\)](#).

**async save\_to\_memory(out)**

Download this file into memory. out needs to be supplied with a [io.BufferedIOBase](#), the file contents will be saved to that object using the [io.BufferedIOBase.write\(\)](#) method.

**Parameters**

**out** ([io.BinaryIO](#)) – A file-like object. Must be opened for writing in binary mode.

**to\_input\_file()**

Converts the file to a standard object for sending/uploading it. This object is require in the file sending methods.

**Returns**

The [bale.InputFile](#) Object for send.

**Return type**

[bale.InputFile](#)

**property type**

a Shortcut for use [bale.File.file\\_type](#)

**Type**

[str](#)

## Input File

**class** [bale.InputFile](#)(*file*, \*, *file\_name=None*)

Bases: [object](#)

This object shows a file ready to send/upload.

**Warning:** Just for upload file, you can use “file\_name” param.



**Parameters**

- **file** (`io.BufferedReader` | `str` | `bytes`) – Your File. Pass a `file_id` as String to send a file that exists on the Bale servers (recommended), pass an HTTP URL as a String for Bale to get a file from the Internet, or upload a new one.
- **file\_name** (Optional[`str`]) – Additional interface options. It is used only when uploading a file.

**Location**

**class** `bale.Location`(*longitude, latitude*)

Bases: `object`

This object shows an Location

**longitude**

Location longitude

**Type**

`int`

**latitude**

Location latitude

**Type**

`int`

**property link**

Export location link from Google map

**Type**

`str`

**Photo**

**class** `bale.Photo`(*file\_id, width, height, file\_size, bot*)

Bases: `File`

This object shows a Photo.

**file\_id**

Identifier for this file, which can be used to download or reuse the file.

**Type**

`str`

**width**

Photo width as defined by sender.

**Type**

`int`

**height**

Photo height as defined by sender.

**Type**

`str`

**file\_size**

File size in bytes.

**Type**

int

**Video**

**class** bale.Video(*file\_id, mime\_type, width, height, file\_size, duration, bot*)

Bases: *File*

This object shows a Video.

**file\_id**

Identifier for this file, which can be used to download or reuse the file.

**Type**

str

**width**

Video width as defined by sender.

**Type**

int

**file\_size**

File size in bytes.

**Type**

int

**height**

Video height as defined by sender.

**Type**

str

**duration**

Duration of the video in seconds as defined by sender.

**Type**

int

**mime\_type**

MIME type of file as defined by sender.

**Type**

str

## 5.2 Helpers

### 5.2.1 Deep Link

`helpers.create_deep_linked_url(payload)`

Creating a deep link for the bot.

#### Parameters

- **bot\_username** (`str`) – The username of bot.
- **payload** (`str`) – The Payload of deep link

## 5.3 Change Log

Project changes are shown on this page.

### 5.3.1 v2.4.6

#### New Features

- Components have been moved to *ui*
- Improve the process of using Components and Files
- Add new method `bale.Bot.send_audio()`
- Support from 400 status code Errors in `bale.HTTPClient`
- Full support from “Bale” rate limits
- Add `bale.Message.attachment`, `bale.Components.menu_keyboards`, `bale.Components.inline_keyboards`
- Update License to LGP
- Improve documentation
- Update `examples` directory

#### Bug Fixes

- Fix bug of stopping the Bot
- Fix the problem of `bale.Bot.send_invoice()` & `bale.Bot.send_video()` checkers

### 5.3.2 v2.4.5

#### New Features

- Improve documentation
- Add new methods (`bale.Bot.forward_message()`, `bale.Bot.ban_chat_member()`, `bale.Bot.send_video()`)
- Add new event `bale.Bot.on_edited_message()`

#### Bug Fixes

- Fix bugs in parse Updates
- Updater.\_\_lock bug
- `bale.Update.type` bugs
- Fix `bale.Chat.invite_link` bugs
- Improving the `bale.Message` class (`__eq__`, `__ne__`, `__repr__`)
- Add new type `channel` to `bale.ChatType`

### 5.3.3 v2.4.4

#### New Features

- Improve the `bale.RateLimit` object
- Add new `bale.Updater`
- Adding the method of `bale.Bot.send_location()` and `bale.Bot.send_contact()`
- Update Readme file

#### Bug Fixes

- Fix `http` error
- Fix `bale.EventType.BEFORE_READY` and `bale.on_before_ready()` bug
- Fix bot closing problem

### 5.3.4 v2.4.3

#### Bug Fixes

- Changes in some functions and commands
- Add `bale.Bot.download_file()` for Download files with `file_id`
- Update `LICENSE`
- Improve Code Quality

### 5.3.5 v2.4.2

#### Bug Fixes

- Changes in some functions and commands
- Improve Code Quality

### 5.3.6 v2.4.1

#### New Features

- Add updater param to `bale.Bot` for Custom-Updater
- Add `bale.Message.type` & support `bale.UpdateType` from it
- Update Readme file

#### Bug Fixes

- Improve Code Quality

### 5.3.7 v2.4.0

#### New Features

- New changes for better Connections
- Synchronization of Exceptions with document
- Add Support from local rate limits
- Add support from `bale.HTTPClient` errors

#### Advance

- Add a Response Parser for connections
- Add Type Checker to All functions
- Add new supporter class for Rate Limits
- Synchronization of methods and Improve Code in many Models ( `bale.User`, `bale.Chat`, `bale.Bot`, `bale.Message` )
- Add `bale.User.chat_id`
- Add `bale.error.RateLimited` Error
- Add `sleep_after_every_get_updates` param to `bale.Bot.run()` and `bale.Bot.start()`

## Bug Fixes

- Improve Code Quality

## 5.3.8 v2.3.2

### New Features

- Add new methods `bale.Bot.get_user()` and `bale.Bot.invite_to_chat()` function
- Support `bale.Chat.invite_to_chat()` from `bale.Bot.get_user()`
- Add `bale.Chat.mention` and `bale.Chat.link` property to `bale.Chat`
- Add `bale.CallbackQuery.user` property to `bale.CallbackQuery` . `bale.CallbackQuery.user` is a aliases for `bale.CallbackQuery.from_user`.
- Add `on_member_chat_join` and `on_member_chat_leave` events
- Add `bale.MemberRole.is_admin()` and `bale.MemberRole.is_owner()` function to `bale.MemberRole`
- Add `save` and `read` function to be `bale.Document`
- Add `get_file` function to `bale.HTTPClient`

## Bug Fixes

- Fixed some function in `bale.Bot`
- Fixed Bad Request error in `bale.Bot.get_chat()`
- Fixed `on_ready` event bug
- Fixed `bale.Bot.get_chat()` bugs

## 5.4 Examples

In this section, there are some robots that are written with python-bale-bot.

### 5.4.1 examples.basic

This robot will answer you with only some commands.

### 5.4.2 examples.inlinemarkup

This example sheds some light on inline keyboards, callback queries and message editing.

### 5.4.3 `examples.attachment`

A basic example of a bot that can send media

### 5.4.4 `examples.invoice`

A basic example of a bot that can accept payments.

### 5.4.5 `examples.conversation`

A common task for a bot is to ask information from the user.





## A

add\_event() (*bale.Bot* method), 11  
 add\_inline\_keyboard() (*bale.Components* method), 30  
 add\_menu\_keyboard() (*bale.Components* method), 30  
 add\_user() (*bale.Chat* method), 25  
 all\_members\_are\_administrators (*bale.Chat* attribute), 25  
 amount (*bale.Price* attribute), 33  
 attachment (*bale.Message* property), 23  
 audio (*bale.Message* attribute), 22  
 Audio (class in *bale*), 34  
 author (*bale.Message* property), 23

## B

ban() (*bale.ChatMember* method), 28  
 ban\_chat\_member() (*bale.Bot* method), 11  
 ban\_chat\_member() (*bale.Chat* method), 25  
 base\_file (*bale.File* property), 36  
 Bot (class in *bale*), 11

## C

callback\_data (*bale.InlineKeyboard* attribute), 31  
 callback\_id (*bale.CallbackQuery* attribute), 27  
 callback\_query (*bale.Update* attribute), 29  
 CallbackQuery (class in *bale*), 27  
 caption (*bale.Message* attribute), 22  
 chat (*bale.Message* attribute), 21  
 Chat (class in *bale*), 24  
 chat\_id (*bale.Chat* attribute), 24  
 chat\_id (*bale.Message* property), 23  
 chat\_id (*bale.User* property), 26  
 ChatMember (class in *bale*), 28  
 close() (*bale.Bot* method), 12  
 Components (class in *bale*), 30  
 contact (*bale.Message* attribute), 22  
 ContactMessage (class in *bale*), 34  
 content (*bale.Message* property), 23  
 create\_deep\_linked\_url() (*bale.helpers* method), 39  
 currency (*bale.Invoice* attribute), 32  
 currency (*bale.SuccessfulPayment* attribute), 33

## D

data (*bale.CallbackQuery* attribute), 27  
 date (*bale.Message* attribute), 21  
 delete() (*bale.Message* method), 23  
 delete\_message() (*bale.Bot* method), 12  
 delete\_webhook() (*bale.Bot* method), 12  
 description (*bale.Invoice* attribute), 32  
 document (*bale.Message* attribute), 22  
 Document (class in *bale*), 35  
 duration (*bale.Audio* attribute), 34  
 duration (*bale.Video* attribute), 38

## E

edit() (*bale.Message* method), 23  
 edit\_message() (*bale.Bot* method), 12  
 edited\_message (*bale.Update* attribute), 29  
 event() (*bale.Bot* method), 13  
 extra (*bale.File* attribute), 36

## F

File (class in *bale*), 35  
 file\_id (*bale.Audio* attribute), 34  
 file\_id (*bale.Document* attribute), 35  
 file\_id (*bale.File* attribute), 35  
 file\_id (*bale.Photo* attribute), 37  
 file\_id (*bale.Video* attribute), 38  
 file\_name (*bale.Document* attribute), 35  
 file\_size (*bale.Audio* attribute), 34  
 file\_size (*bale.Document* attribute), 35  
 file\_size (*bale.File* attribute), 35  
 file\_size (*bale.Photo* attribute), 37  
 file\_size (*bale.Video* attribute), 38  
 file\_type (*bale.File* attribute), 35  
 first\_name (*bale.Chat* attribute), 24  
 first\_name (*bale.ContactMessage* attribute), 34  
 first\_name (*bale.User* attribute), 26  
 forward() (*bale.Message* method), 23  
 forward\_from (*bale.Message* attribute), 22  
 forward\_from\_chat (*bale.Message* attribute), 22  
 forward\_message() (*bale.Bot* method), 13  
 from\_dict() (*bale.Permissions* class method), 29  
 from\_user (*bale.CallbackQuery* attribute), 27

`from_user` (*bale.Message* attribute), 21

## G

`get()` (*bale.File* method), 36  
`get_bot()` (*bale.Bot* method), 13  
`get_chat()` (*bale.Bot* method), 13  
`get_chat_administrators()` (*bale.Bot* method), 14  
`get_chat_administrators()` (*bale.Chat* method), 25  
`get_chat_member()` (*bale.Bot* method), 14  
`get_chat_member()` (*bale.Chat* method), 25  
`get_chat_members_count()` (*bale.Bot* method), 14  
`get_chat_members_count()` (*bale.Chat* method), 25  
`get_file()` (*bale.Bot* method), 15  
`get_user()` (*bale.Bot* method), 15

## H

`height` (*bale.Photo* attribute), 37  
`height` (*bale.Video* attribute), 38

## I

`inline_keyboards` (*bale.Components* property), 30  
`inline_message_id` (*bale.CallbackQuery* attribute), 27  
`InlineKeyboard` (class in *bale*), 31  
`InputFile` (class in *bale*), 36  
`invite_link` (*bale.Chat* attribute), 25  
`invite_user()` (*bale.Bot* method), 15  
`invoice` (*bale.Message* attribute), 23  
`Invoice` (class in *bale*), 32  
`invoice_payload` (*bale.SuccessfulPayment* attribute), 33  
`is_bot` (*bale.User* attribute), 26  
`is_closed()` (*bale.Bot* method), 16

## L

`label` (*bale.Price* attribute), 33  
`last_name` (*bale.Chat* attribute), 24  
`last_name` (*bale.ContactMessage* attribute), 34  
`last_name` (*bale.User* attribute), 26  
`latitude` (*bale.Location* attribute), 37  
`leave()` (*bale.Chat* method), 25  
`leave_chat()` (*bale.Bot* method), 16  
`left_chat_member` (*bale.Message* attribute), 23  
`link` (*bale.Location* property), 37  
`listen()` (*bale.Bot* method), 16  
`location` (*bale.Message* attribute), 22  
`Location` (class in *bale*), 37  
`longitude` (*bale.Location* attribute), 37

## M

`mention` (*bale.User* property), 26  
`menu_keyboards` (*bale.Components* property), 30  
`MenuKeyboard` (class in *bale*), 31  
`message` (*bale.CallbackQuery* attribute), 27

`message` (*bale.Update* attribute), 29  
`Message` (class in *bale*), 21  
`message_id` (*bale.Message* attribute), 21  
`mime_type` (*bale.Audio* attribute), 34  
`mime_type` (*bale.Document* attribute), 35  
`mime_type` (*bale.File* attribute), 35  
`mime_type` (*bale.Video* attribute), 38

## N

`new_chat_members` (*bale.Message* attribute), 22

## O

`on_error()` (*bale.Bot* method), 16

## P

`parsed_status` (*bale.ChatMember* property), 28  
`parsed_type` (*bale.Chat* property), 25  
`payload` (*bale.SuccessfulPayment* property), 33  
`permissions` (*bale.ChatMember* attribute), 28  
`Permissions` (class in *bale*), 28  
`phone_number` (*bale.ContactMessage* attribute), 34  
`Photo` (class in *bale*), 37  
`pinned_message` (*bale.Chat* attribute), 24  
`Price` (class in *bale*), 33

## R

`remove_inline_keyboard()` (*bale.Components* method), 30  
`remove_menu_keyboard()` (*bale.Components* method), 30  
`reply()` (*bale.Message* method), 23  
`reply_audio()` (*bale.Message* method), 23  
`reply_document()` (*bale.Message* method), 23  
`reply_photo()` (*bale.Message* method), 23  
`reply_to_message` (*bale.Message* attribute), 22  
`reply_to_message_id` (*bale.Message* property), 24  
`reply_video()` (*bale.Message* method), 24  
`request_contact` (*bale.MenuKeyboard* attribute), 31  
`request_location` (*bale.MenuKeyboard* attribute), 32  
`run()` (*bale.Bot* method), 16

## S

`save_to_memory()` (*bale.File* method), 36  
`send()` (*bale.Chat* method), 25  
`send()` (*bale.User* method), 26  
`send_audio()` (*bale.Bot* method), 16  
`send_audio()` (*bale.Chat* method), 25  
`send_audio()` (*bale.User* method), 26  
`send_contact()` (*bale.Bot* method), 17  
`send_contact()` (*bale.Chat* method), 25  
`send_contact()` (*bale.User* method), 27  
`send_document()` (*bale.Bot* method), 17  
`send_document()` (*bale.Chat* method), 25

[send\\_document\(\)](#) (*bale.User method*), 27  
[send\\_invoice\(\)](#) (*bale.Bot method*), 17  
[send\\_invoice\(\)](#) (*bale.Chat method*), 25  
[send\\_invoice\(\)](#) (*bale.User method*), 27  
[send\\_location\(\)](#) (*bale.Bot method*), 18  
[send\\_location\(\)](#) (*bale.Chat method*), 25  
[send\\_location\(\)](#) (*bale.User method*), 27  
[send\\_message\(\)](#) (*bale.Bot method*), 19  
[send\\_photo\(\)](#) (*bale.Bot method*), 19  
[send\\_photo\(\)](#) (*bale.Chat method*), 26  
[send\\_photo\(\)](#) (*bale.User method*), 27  
[send\\_video\(\)](#) (*bale.Bot method*), 20  
[send\\_video\(\)](#) (*bale.Chat method*), 26  
[send\\_video\(\)](#) (*bale.User method*), 27  
[set\\_webhook\(\)](#) (*bale.Bot method*), 20  
[shipping\\_option\\_id](#) (*bale.SuccessfulPayment attribute*), 33  
[start\\_parameter](#) (*bale.Invoice attribute*), 32  
[status](#) (*bale.ChatMember attribute*), 28  
[successful\\_payment](#) (*bale.Message attribute*), 23  
[SuccessfulPayment](#) (*class in bale*), 33  
[switch\\_inline\\_query](#) (*bale.InlineKeyboard attribute*), 31  
[switch\\_inline\\_query\\_current\\_chat](#) (*bale.InlineKeyboard attribute*), 31

## T

[text](#) (*bale.InlineKeyboard attribute*), 31  
[text](#) (*bale.MenuKeyboard attribute*), 31  
[text](#) (*bale.Message attribute*), 21  
[title](#) (*bale.Audio attribute*), 34  
[title](#) (*bale.Chat attribute*), 24  
[title](#) (*bale.Invoice attribute*), 32  
[to\\_input\\_file\(\)](#) (*bale.File method*), 36  
[total\\_amount](#) (*bale.Invoice attribute*), 32  
[total\\_amount](#) (*bale.SuccessfulPayment attribute*), 33  
[type](#) (*bale.Chat attribute*), 24  
[type](#) (*bale.File property*), 36

## U

[Update](#) (*class in bale*), 29  
[update\\_id](#) (*bale.Update attribute*), 29  
[url](#) (*bale.InlineKeyboard attribute*), 31  
[user](#) (*bale.Bot property*), 20  
[user](#) (*bale.CallbackQuery property*), 28  
[user](#) (*bale.ChatMember attribute*), 28  
[User](#) (*class in bale*), 26  
[user\\_id](#) (*bale.User attribute*), 26  
[username](#) (*bale.Chat attribute*), 24  
[username](#) (*bale.User attribute*), 26

## V

[video](#) (*bale.Message attribute*), 22  
[Video](#) (*class in bale*), 38

## W

[wait\\_for\(\)](#) (*bale.Bot method*), 20  
[width](#) (*bale.Photo attribute*), 37  
[width](#) (*bale.Video attribute*), 38