
python-bale-bot

Release 2.5.0

Kian Ahmadian

Mar 30, 2025

REFERENCE

1	Introduction	3
1.1	What is Bale?	3
1.2	What is python-bale-bot?	3
2	Installing	5
2.1	PyPi:	5
2.2	Git:	5
3	Quick Start	7
4	Documentation	9
5	Contact to Developers	11
5.1	Bale Package	11
5.2	Helpers	12
5.3	Event Reference	12
5.4	Examples	14
	Index	15



python-bale-bot

An API wrapper for Bale written in Python.

INTRODUCTION

1.1 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 \(API\)](#).

1.2 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:

2.1 PyPi:

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

2.2 Git:

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


QUICK START

To get started, learn how the library works through the library. In addition, there are examples in the “[Examples](#)” section of the library.

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

Available Types

Update

Message

Chat

Chat Member

Chat Photo

User

Callback Query

Sticker

Wait Context

Webhook info

UI

Inline Keyboard Markup

Menu Keyboard Markup

Reply Markup Item

Inline Keyboard Button

Menu Keyboard Button

Payments

Invoice

Labeled Price

Successful Payment

Attachments

Audio

Voice

Contact

Document

Base File

Input File

Location

Photo Size

Video

Animation

Handlers

Base Handler

Message Handler

Command Handler

Callback Query Handler

Edited Message Handler

5.2 Helpers

5.2.1 Deep Link

5.2.2 Find

5.3 Event Reference

The list of events that can be received by the bot.

An example of how to listen to events in different situations (for `on_message` event):

```
from bale import Bot, Message

bot = Bot("YOUR_TOKEN")

@bot.event
```

(continues on next page)

(continued from previous page)

```

async def on_message(message: Message):
    if message.content == '/start':
        return await message.reply("Hi, python-bale-bot!")

bot.run()

```

```

from bale import Bot, Message

class MyBot(Bot):
    async def on_message(self, message: Message):
        if message.content == '/start':
            return await message.reply("Hi, python-bale-bot!")

MyBot('YOUR_TOKEN').run()

```

5.3.1 Connection

`async on_before_ready()`

This event is called before the updater starts.

`async on_ready()`

When the updater starts working and the Bot information is placed in `bale.Bot.user`.

5.3.2 Updates

`async on_update(update)`

This event is called when an update is received from “Bale” servers.

Parameters

update (`bale.Update`) – update received.

5.3.3 Messages

`async on_message(message)`

This event is called when sending a message in a chat to which the bot is connected.

Parameters

message (`bale.Message`) – message sent.

`async on_message_edit(message)`

This event is called when the sent message is edited.

Parameters

message (`bale.Message`) – message edited.

5.3.4 CallbackQuery

`async on_callback(callback)`

This event is called when a callback query is created.

Parameters

callback (`bale.CallbackQuery`) – callback received.

5.3.5 Groups

`async on_member_chat_join(message, chat, user)`

This event is called when a user joins the chat.

Parameters

- **message** (`bale.Message`) – message sent.
- **chat** (`bale.Chat`) – the chat.
- **user** (`bale.User`) – the user.

`async on_member_chat_leave(message, chat, user)`

This event is called when a user leaves the chat.

Parameters

- **message** (`bale.Message`) – message sent.
- **chat** (`bale.Chat`) – the chat.
- **user** (`bale.User`) – the user.

5.3.6 Payments

`async on_successful_payment(payment)`

This event is called when a transaction is completed and its status is successful.

Parameters

- **successful_payment** (`bale.SuccessfulPayment`) – the payment.

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.

B

built-in function

- on_before_ready(), 13
- on_callback(), 13
- on_member_chat_join(), 14
- on_member_chat_leave(), 14
- on_message(), 13
- on_message_edit(), 13
- on_ready(), 13
- on_successful_payment(), 14
- on_update(), 13

O

on_before_ready()

- built-in function, 13

on_callback()

- built-in function, 13

on_member_chat_join()

- built-in function, 14

on_member_chat_leave()

- built-in function, 14

on_message()

- built-in function, 13

on_message_edit()

- built-in function, 13

on_ready()

- built-in function, 13

on_successful_payment()

- built-in function, 14

on_update()

- built-in function, 13