
Telegrask

Release 0.3.8

samedamci

Aug 26, 2021

CONTENTS:

1	User’s Guide	3
1.1	Getting Started	3
1.2	Usage Basics	3
2	Main Framework	5
2.1	Handlers Decorators	5
2.2	Help Parser	6
3	Extensions	9

Welcome to Telegrask's documentation!

USER'S GUIDE

1.1 Getting Started

1.1.1 Installing

To install Telegrask run following command:

```
$ python3 -m pip install Telegrask
```

1.1.2 Project Initialization

You can manually create project structure or use built-in project initializer:

```
$ python3 -m telegrask init <PROJECT_NAME>
```

1.2 Usage Basics

Main goal of this framework is use handlers decorators and make bot creating more intuitive.

1.2.1 Simplest Example

Hello World bot example in Telegrask:

```
from telegrask import Telegrask

bot = Telegrask("TOKEN")

@bot.command("hello", help='display "Hello, World!"')
def hello_command(update, context):
    update.message.reply_text("Hello, World!")

if __name__ == "__main__":
    bot.run(debug=True)
```

and equivalent code in pure `python-telegram-bot` library:

```
from telegram.ext import Updater, CommandHandler
from telegram import ParseMode
import logging

logging.basicConfig(format="%(levelname)s - %(message)s", level=logging.DEBUG)
logger = logging.getLogger(__name__)

def hello_command(update, context):
    update.message.reply_text("Hello, World!")

def help_command(update, context):
    help_content = """*Available commands*

/hello
display "Hello, World!"

/help
display this message
"""
    update.message.reply_text(help_content, parse_mode=ParseMode.MARKDOWN)

def main():
    global updater
    updater = Updater("BOT_TOKEN")
    dispatcher = updater.dispatcher
    dispatcher.add_handler(CommandHandler("hello", hello_command))
    dispatcher.add_handler(CommandHandler(["help", "start"], help_command))
    updater.start_polling()
    updater.idle()

if __name__ == "__main__":
    main()
```


MAIN FRAMEWORK

2.1 Handlers Decorators

Each handler decorator is a child method of `Telegrask` class, so you need to have an object of this class first.

```
from telegram import Update
from telegram.ext import CallbackContext

bot = Telegrask("TOKEN")
```

2.1.1 @command

Handler for commands (messages starting with / prefix, e.g. /cmd).

```
@bot.command("command_name", help="command help message")
def callback_function(update: Update, context: CallbackContext):
    pass
```

Parameters:

- `commands`: `Union[str, list]` - command or list of commands which will use this function as callback
- `help`: `str` - description for command which will be included in automatically generated /help command message
- `allow_without_prefix`: `bool = False` - handle command also without / prefix, e.g. /cmd and cmd

2.1.2 @message

Handler for messages.

```
from telegram.ext import Filters

@bot.message(...)
def callback_function(update: Update, context: CallbackContext):
    pass
```

Parameters:

- `filters`: `Filters` - any combination of `Filters` from `python-telegram-bot` library

2.1.3 @message_regex

Handler for messages by regexes instead of filters.

```
@bot.message_regex(r"regex or just text")
def callback_function(update: Update, context: CallbackContext):
    pass
```

Parameters:

- `regex`: Pattern - regular expression to filter a message or just text

2.1.4 @inline_query

Handler for inline queries.

```
from telegrask import InlineQuery

@bot.inline_query
def inline(query: InlineQuery):
    pass
```

2.2 Help Parser

HelpParser is built-in Telegrask module which parses `/help` command automatically. To disable this feature you have to set `bot.config["HELP_MESSAGE"] = False` (where `bot` is instance of Telegrask class).

We recommend to use `bot.help` object which is default instance of HelpParser class.

2.2.1 @custom_help_command

This is another Telegrask decorator which is helpful when you want to create own help command.

```
from telegram import Update
from telegram.ext import CallbackContext

@bot.custom_help_command
def help(update: Update, context: CallbackContext, commands):
    # commands is a dict {"command_name": "command_description", ...}
    # reference for `bot.help.commands_descriptions`
    pass
```

2.2.2 Helpful Hacks

- `bot.help.header = "Custom help message header"`
- `bot.help.help_description = "custom /help command description"`
- Adding Custom Handlers
- Configuration
- Running

EXTENSIONS

- Chat
- Moderation
- UserURL