
aiogram*opframework*

Release 0.0.1

Apr 05, 2022

Contents:

1	Installation	1
2	Quick-Start	3
3	Views documentation	5
3.1	Message Views	5
3.2	Other update types' views	9
3.3	Custom views	9
4	Settings	11
5	Additional Filters	13
5.1	filter_execute decorator for methods	13
6	Utils	15
6.1	aiogram_oop_framework.utils.get_help	15
6.2	Commands	16
7	Indices and tables	17
	Python Module Index	19
	Index	21

CHAPTER 1

Installation

install with:

```
pip install aiogram_oop_framework
```

uninstall with:

```
pip uninstall aiogram_oop_framework
```


CHAPTER 2

Quick-Start

1) Create project

```
python -m aiogram_oop_framework startproject mybot
```

aiogram_oop_framework will create directory with your project.

Create projects in root directory, otherwise you may get in some bugs (but if you are ready to fix it, than it is even better for this framework's future xD)

2) Create your views in mybot.views directory You can create views anywhere inside mybot (but not outside), but preferably in mybot.views or it's subdirectories Your view must be inherited from any View class, for example: [MessageView](#) and have classmethod .execute() which is used to run code. And it also may have additional class-methods with any name, with filters (see [additional-filters](#) for more info about filters) Code example:

```
class Example(MessageView):

    @classmethod
    async def execute(cls, m: types.Message, state: FSMContext = None, **kwargs):
        await m.answer('Hello, World!')
```

3) If you had created views not in mybot.views (if you created views in it's subdirectories, this is also for you): In settings.py change project structure by writing struc.include(<path to directory containing your views> with dots as separators) before pr.structure = struc For example:

```
pr: Project = Project(PROJECT_NAME, PATH)
struc: ProjectStructure = ProjectStructure(pr)
struc.include('views')
struc.include('views.messages.user_commands')
struc.include('views.callbacks')
pr.structure = struc

PROJECT: Project = pr
```

4) Write your settings in settings.py (see detailed explanation of all settings in [Settings](#))

5) You can now start polling with `python -m mybot.manage start-polling` or in code:

```
from aiogram import executor

from mybot.manage import initialize_project

if __name__ == '__main__':
    dp, bot = initialize_project()
    executor.start_polling(dp)
```

OR

```
from aiogram import executor, Dispatcher, Bot

from mybot.manage import initialize_project
from mybot.settings import TELEGRAM_BOT_TOKEN

if __name__ == '__main__':
    bot: Bot = Bot(TELEGRAM_BOT_TOKEN)
    dp: Dispatcher = Dispatcher(bot)
    initialize_project(dp, bot)
    executor.start_polling(dp)
```


3.1 Message Views

class `aiogram_oop_framework.views.message.MessageView`

View for updates of type “message”

custom_filters

Custom filters (for ex.: `[lambda m: m.reply_to_message]`).

Type list

commands

List of commands (for ex.: `['start', 'help']`), defaults to None

Type list

regexp

Regexp string for matching message text/caption (for ex.: `r'^hello$'`), defaults to None

Type str

content_types

List of `content_type`'s of message (for ex.: `['text', 'photo']`), defaults to None

Type list

state

Function, which returns a State object of aiogram or “*”, defaults to `lambda: None`

Type Callable

run_task

Run callback in task (no wait results), defaults to None

Type bool

register_kwargs

Kwargs, which you would add in `@dp.message_handler` in fresh aiogram, defaults to None

Type dict

index

in which order to register the view, defaults to None, may be negative to add at the end like in a list. If None that it's put in empty indexes left after others. Example explaining every detail hopefully: [0, None, 2, 3, None, None, -7, -2, -1]

Type int

auto_register

set to False if you don't want register the view automatically, ignored, if AUTO_REGISTER_VIEWS in settings.py is set to False, defaults to True

Type bool

help_text

help's text to return by [aiogram_oop_framework.utils.get_help](#), also user in utils.Commands to get description for command if command_description is empty or None

Type str

short_description

command's description for every command in commands attribute to call set_my_commands in initialize_projects, if no view has command_description, then it is not called, also user in utils.Commands to get description for command

Type str

set_my_commands

“first” -> register in set_my_commands the first command from commands; “all” -> register in set_my_commands the first command from commands; list, set, tuple of strings -> register the specified commands; None -> don't register this view in set_my_commands

Type set, list, tuple, str

Make sure you don't want to use a more high-level view like [aiogram_oop_framework.views.custom_views.command.CommandView](#) or [aiogram_oop_framework.views.content_types_views.text.TextView](#) instead.

You may found more info about custom_filters, commands, regexp, content_types, state and run_task attributes in aiogram's docs on Dispatcher.message_handler or you may not, depends on aiogram's docs.

classmethod register (dp: *aiogram.dispatcher.dispatcher.Dispatcher*)

method to register the handler, normally called automatically in manage.initialize_project()

Parameters dp (*Dispatcher*) – in which dispatcher to register the handler

class aiogram_oop_framework.views.edited_message_view.**EditedMessageView**

classmethod register (dp: *aiogram.dispatcher.dispatcher.Dispatcher*)

method to register the handler, normally called automatically in manage.initialize_project()

Parameters dp (*Dispatcher*) – in which dispatcher to register the handler

class aiogram_oop_framework.views.channel_post_view.**ChannelPostView**

classmethod register (dp: *aiogram.dispatcher.dispatcher.Dispatcher*)

method to register the handler, normally called automatically in manage.initialize_project()

Parameters dp (*Dispatcher*) – in which dispatcher to register the handler

```
class aiogram_oopframework.views.edited_channel_post_view.EditedChannelPostView
```

```
classmethod register (dp: aiogram.dispatcher.dispatcher.Dispatcher)
```

method to register the handler, normally called automatically in manage.initialize_project()

Parameters dp (*Dispatcher*) – in which dispatcher to register the handler

3.1.1 Views by content_types

```
class aiogram_oopframework.views.content_types_views.animation.AnimationView
```

May be same as *aiogram_oopframework.views.message.MessageView*, or *aiogram_oopframework.views.edited_message_view.EditedMessageView*, or *aiogram_oopframework.views.channel_post_view.ChannelPostView*, or *aiogram_oopframework.views.edited_channel_post_view.EditedChannelPostView*, dependently on update_type field, with difference that content_types field defaults to [ContentType.ANIMATION], and it has one more field:

update_type

Update type, must be one of: 'message', 'edited_message', 'channel_post', 'edited_channel_post', defaults to 'message's

Type str

```
class aiogram_oopframework.views.content_types_views.audio.AudioView
```

Same as *aiogram_oopframework.views.content_types_views.animation.AnimationView*, but content_types defaults to [ContentType.AUDIO]

```
class aiogram_oopframework.views.content_types_views.connected_website.ConnectedWebsiteView
```

Same as *aiogram_oopframework.views.content_types_views.animation.AnimationView*, but content_types defaults to [ContentType.CONNECTED_WEBSITE]

```
class aiogram_oopframework.views.content_types_views.contact.ContactView
```

Same as *aiogram_oopframework.views.content_types_views.animation.AnimationView*, but content_types defaults to [ContentType.CONTACT]

```
class aiogram_oopframework.views.content_types_views.delete_chat_photo.DeleteChatPhotoView
```

Same as *aiogram_oopframework.views.content_types_views.animation.AnimationView*, but content_types defaults to [ContentType.DELETE_CHAT_PHOTO]

```
class aiogram_oopframework.views.content_types_views.dice.DiceView
```

Same as *aiogram_oopframework.views.content_types_views.animation.AnimationView*, but content_types defaults to [ContentType.DICE]

```
class aiogram_oopframework.views.content_types_views.document.DocumentView
```

Same as *aiogram_oopframework.views.content_types_views.animation.AnimationView*, but content_types defaults to [ContentType.DOCUMENT]

```
class aiogram_oopframework.views.content_types_views.game.GameView
```

Same as *aiogram_oopframework.views.content_types_views.animation.AnimationView*, but content_types defaults to [ContentType.GAME]

```
class aiogram_oopframework.views.content_types_views.group_chat_created.GroupChatCreatedView
```

Same as *aiogram_oopframework.views.content_types_views.animation.AnimationView*, but content_types defaults to [ContentType.GROUP_CHAT_CREATED]

```
class aiogram_oopframework.views.content_types_views.invoice.InvoiceView
```

Same as *aiogram_oopframework.views.content_types_views.animation.AnimationView*, but content_types defaults to [ContentType.INVOICE]

```
class aiogram_oop_framework.views.content_types_views.left_chat_member.LeftChatMemberView
    Same      as      aiogram_oop_framework.views.content_types_views.animation.
                    AnimationView, but content_types defaults to [ContentType.LEFT_CHAT_MEMBER]

class aiogram_oop_framework.views.content_types_views.location.LocationView
    Same      as      aiogram_oop_framework.views.content_types_views.animation.
                    AnimationView, but content_types defaults to [ContentType.LOCATION]

class aiogram_oop_framework.views.content_types_views.migrate_from_chat_id.MigrateFromChatIdView
    Same      as      aiogram_oop_framework.views.content_types_views.animation.
                    AnimationView, but content_types defaults to [ContentType.MIGRATE_FROM_CHAT_ID]

class aiogram_oop_framework.views.content_types_views.migrate_to_chat_id.MigrateToChatIdView
    Same      as      aiogram_oop_framework.views.content_types_views.animation.
                    AnimationView, but content_types defaults to [ContentType.MIGRATE_TO_CHAT_ID]

class aiogram_oop_framework.views.content_types_views.new_chat_members.NewChatMembersView
    Same      as      aiogram_oop_framework.views.content_types_views.animation.
                    AnimationView, but content_types defaults to [ContentType.NEW_CHAT_MEMBERS]

class aiogram_oop_framework.views.content_types_views.new_chat_photo.NewChatPhotoView
    Same      as      aiogram_oop_framework.views.content_types_views.animation.
                    AnimationView, but content_types defaults to [ContentType.NEW_CHAT_PHOTO]

class aiogram_oop_framework.views.content_types_views.new_chat_title.NewChatTitleView
    Same      as      aiogram_oop_framework.views.content_types_views.animation.
                    AnimationView, but content_types defaults to [ContentType.NEW_CHAT_TITLE]

class aiogram_oop_framework.views.content_types_views.passport_data.PassportDataView
    Same      as      aiogram_oop_framework.views.content_types_views.animation.
                    AnimationView, but content_types defaults to [ContentType.PASSPORT_DATA]

class aiogram_oop_framework.views.content_types_views.photo.PhotoView
    Same      as      aiogram_oop_framework.views.content_types_views.animation.
                    AnimationView, but content_types defaults to [ContentType.PHOTO]

class aiogram_oop_framework.views.content_types_views.pinned_message.PinnedMessageView
    Same      as      aiogram_oop_framework.views.content_types_views.animation.
                    AnimationView, but content_types defaults to [ContentType.PINNED_MESSAGE]

class aiogram_oop_framework.views.content_types_views.poll.PollMessageView
    Same      as      aiogram_oop_framework.views.content_types_views.animation.
                    AnimationView, but content_types defaults to [ContentType.POLL]

class aiogram_oop_framework.views.content_types_views.sticker.StickerView
    Same      as      aiogram_oop_framework.views.content_types_views.animation.
                    AnimationView, but content_types defaults to [ContentType.STICKER]

class aiogram_oop_framework.views.content_types_views.successful_payment.SuccessfulPaymentView
    Same      as      aiogram_oop_framework.views.content_types_views.animation.
                    AnimationView, but content_types defaults to [ContentType.SUCCESSFUL_PAYMENT]

class aiogram_oop_framework.views.content_types_views.text.TextView
    Same      as      aiogram_oop_framework.views.content_types_views.animation.
                    AnimationView, but content_types defaults to [ContentType.TEXT]

class aiogram_oop_framework.views.content_types_views.venue.VenueView
    Same      as      aiogram_oop_framework.views.content_types_views.animation.
                    AnimationView, but content_types defaults to [ContentType.VENUE]
```

```

class aiogram_oopframework.views.content_types_views.video.VideoView
    Same as aiogram_oopframework.views.content_types_views.animation.AnimationView, but content_types defaults to [ContentType.VIDEO]

class aiogram_oopframework.views.content_types_views.video_note.VideoNoteView
    Same as aiogram_oopframework.views.content_types_views.animation.AnimationView, but content_types defaults to [ContentType.VIDEO_NOTE]

class aiogram_oopframework.views.content_types_views.voice.VoiceView
    Same as aiogram_oopframework.views.content_types_views.animation.AnimationView, but content_types defaults to [ContentType.VOICE]

class aiogram_oopframework.views.content_types_views.any_content_type_message.AnyContentTypeMessageView
    Same as aiogram_oopframework.views.content_types_views.animation.AnimationView, but content_types defaults to ContentType.all()

```

3.2 Other update types' views

```

class aiogram_oopframework.views.inline_query.InlineQueryView
    View for updates of type "message"

```

custom_filters

Custom filters (for ex.: [lambda m: m.reply_to_message]).

Type list

state

Function, which returns a State object of aiogram or "*"

Type Callable

run_task

Run callback in task (no wait results)

Type bool

register_kwargs

Kwargs, which you would add in @dp.message_handler in fresh aiogram

Type dict

You may found more info about attributes in aiogram's docs on Dispatcher.message_handler or you may not, depends on aiogram's docs.

```

class aiogram_oopframework.views.callback_query.CallbackQueryView

```

```

class aiogram_oopframework.views.poll.PollView

```

```

class aiogram_oopframework.views.poll_answer.PollAnswerView

```

```

class aiogram_oopframework.views.shipping_query.ShippingQueryView

```

3.3 Custom views

```

class aiogram_oopframework.views.custom_views.command.CommandView

```

May be same as *aiogram_oopframework.views.message.MessageView*, or
aiogram_oopframework.views.edited_message_view.EditedMessageView,
or *aiogram_oopframework.views.channel_post_view.ChannelPostView*,

or `aiogram_oop_framework.views.edited_channel_post_view.EditedChannelPostView`, dependently on `update_type` field, with difference that `content_types` field defaults to `[ContentType.TEXT]`, and it has some differences, look down:

commands

commands list (not necessary, default command is auto-generated basing on class's name)

Type list

update_type

Update type, must be one of: 'message', 'edited_message', 'channel_post', 'edited_channel_post', defaults to 'message'

Type str

append_commands

if True -> appends default auto-generated command to commands; ignored (just uses default auto-generated command) if `bool(commands)` is False

Type bool

default_command_case

default auto-generated command's case (snake_case/lowercase)

Type str

while registering looks at your view's name and creates a default command with it (Start -> ['start'])

classmethod register (*dp: aiogram.dispatcher.dispatcher.Dispatcher*)

CHAPTER 4

Settings

Your project settings should be in `projects_name/settings.py`

PATH: path to folder in which project folder is **PROJECT_NAME:** name of the folder of the project

PROJECT: `aiogram_oop_framework.core.project.Project` instance, you don't need to change it, only it's `structure` field (see in quick-start)

AUTO_REGISTER_VIEWS: does the framework need to register your views automatically or you will do it yourself

TELEGRAM_BOT_TOKEN: token of your bot

MIDDLEWARES: list of your middlewares (from `aiogram`), must be classes, not instances

BOT_STORAGE: storage of the bot, you may look closer about that in `aiogram`'s docs, or you may not, depends in `aiogram`'s docs

MEMORY_STORAGE: DEPRECATED IN 0.2.dev3, will be removed, old name of **BOT_STORAGE**

PARSE_MODE: default parse mode for your bot

DEFAULT_COMMANDS_CASE: (str or `aiogram_oop_framework.utils.CommandCaseHelper` attribute (which is still str)) specify the default `aiogram_oop_framework.views.CommandView`'s auto-generated command's case, `CommandCaseHelper.snake_case` by default

5.1 filter_execute decorator for methods

You can add to your views additional filters for calling different methods dependently on conditions. `filter_execute` accepts all `aiogram` filters with `content_types` set to `ContentType.all()` by default

Example:

```

from aiogram import types
from aiogram.dispatcher import FSMContext
from aiogram.types.chat_member import ChatMemberStatus

from aiogram_oop_framework.views import MessageView
from aiogram_oop_framework.filters import filter_execute

class Example(MessageView):

    @classmethod
    async def execute(cls, m: types.Message, state: FSMContext = None, **kwargs):
        """this will be executed if chat type is not private"""
        await m.answer('undef')

    @classmethod
    @filter_execute(chat_member_status=ChatMemberStatus.ADMINISTRATOR)
    async def execute_for_admins(cls, m: types.Message, state: FSMContext = None,
    → **kwargs):
        """execute if sender is administrator"""
        await m.answer('gg ADMIN brr')

```

5.1.1 aiogram_oop_framework builtin filters list:

chat_member_status status of from_user member (it calls `bot.get_chat_member` to find out the status!)

chat_type type of chat where update occurred

entities if message.entities/message.caption_entities/poll.explanation_entities contains entities with specified types

poll_type type of the Poll

dice_emoji dice's emoji

func

a callable, which returns boolean value

DEPRECATED in 0.2.dev4, because of starting using aiogram filters, so you can simply put the func in args (ex.: filter_execute(foo))

While func may only be a function or lambda, other filter types accept much more values:

- str
- aiogram.util.helper.Item (for ex.: ChatType.PRIVATE)
- list of the above
- a function, which accepts a TelegramObject as parameter and returns any of the above

Note!

filter_execute works with execute_in_<chat_type> with no problem, if it's given chat_type in filter_execute, then chat_type from method's name is omitted

6.1 aiogram_oop_framework.utils.get_help

Use `get_help(command)` to get help for command

It looks for all registered handlers with the specified command in handler's filters and returns `.get_help()` return (which is `(View.help_text or View.__doc__)` by default) of the first found `View`, if the handler is method of a `BaseView` subclass, else it returns `__doc__` argument of the function

So, you can write help's text in `View`'s docs, in `View`'s `help_text` attribute, in `View`'s `get_text()` method, or if not using views (what the point using the framework then however?) - in function docs.

6.1.1 Usage examples:

```
class Start(CommandView):
    """start doc"""
    @classmethod
    async def execute(cls, m: Message, state: FSMContext = None, **kwargs):
        await m.answer('hi')

@dp.message_handler(commands=['cancel'])
async def foo(m: Message, state: FSMContext = None, **kwargs):
    """cancel doc"""
    await m.answer('canceled')

class Help(CommandView):
    """your description for you and other developers"""
    help_text = "help help for users"
    @classmethod
    async def execute(cls, m: Message, state: FSMContext = None, **kwargs):
        await m.answer('...\n\nWrite /help &lt;command> for more info')
```

(continues on next page)

(continued from previous page)

```
@classmethod
@filter_execute(lambda m: m.get_args(), update_type='message')
async def execute_with_args(cls, m: Message, state: FSMContext = None, **kwargs):
    help_ = get_help(m.get_args())
    if help_:
        await m.answer(help_)
    else:
        await m.answer("Sry, help for command isn't found")
```

6.2 Commands

Docs not ready yet...

CHAPTER 7

Indices and tables

- `genindex`
- `modindex`
- `search`

Python Module Index

```
a      aioogram_oop_framework.views.content_types_views.new_chat_member, 8
aioogram_oop_framework.views.callback_query, 9
aioogram_oop_framework.views.channel_post_view, 6
aioogram_oop_framework.views.content_types_views.animation, 7
aioogram_oop_framework.views.content_types_views.any_content_type_message, 9
aioogram_oop_framework.views.content_types_views.audio, 7
aioogram_oop_framework.views.content_types_views.connected_website, 7
aioogram_oop_framework.views.content_types_views.contact, 7
aioogram_oop_framework.views.content_types_views.delete_chat_photo, 7
aioogram_oop_framework.views.content_types_views.dice, 7
aioogram_oop_framework.views.content_types_views.document, 7
aioogram_oop_framework.views.content_types_views.game, 7
aioogram_oop_framework.views.content_types_views.group_chat_created, 7
aioogram_oop_framework.views.content_types_views.invoice, 7
aioogram_oop_framework.views.content_types_views.left_chat_member, 7
aioogram_oop_framework.views.content_types_views.location, 8
aioogram_oop_framework.views.content_types_views.migrate_from_chat_id, 8
aioogram_oop_framework.views.content_types_views.migrate_to_chat_id, 8
aioogram_oop_framework.views.content_types_views.new_chat_members, 8
aioogram_oop_framework.views.content_types_views.new_chat_photo, 8
```


A

```

A
aiogram_oop_framework.views.callback_query (module), 8
aiogram_oop_framework.views.channel_post_view (module), 6
aiogram_oop_framework.views.content_types_views.animation (module), 7
aiogram_oop_framework.views.content_types_views.any_content_type_message (module), 9
aiogram_oop_framework.views.content_types_views.audio (module), 7
aiogram_oop_framework.views.content_types_views.connected_website (module), 7
aiogram_oop_framework.views.content_types_views.contact (module), 7
aiogram_oop_framework.views.content_types_views.delete_chat_photo (module), 7
aiogram_oop_framework.views.content_types_views.dice (module), 7
aiogram_oop_framework.views.content_types_views.document (module), 7
aiogram_oop_framework.views.content_types_views.game (module), 7
aiogram_oop_framework.views.content_types_views.group_chat_created (module), 7
aiogram_oop_framework.views.content_types_views.inline (module), 7
aiogram_oop_framework.views.content_types_views.location (module), 6
aiogram_oop_framework.views.content_types_views.migrate_from_chat_id (module), 8
aiogram_oop_framework.views.content_types_views.migrate_to_chat_id (module), 8
aiogram_oop_framework.views.content_types_views.new_chat_members (module), 8
aiogram_oop_framework.views.content_types_views.new_chat_photo (module), 8
aiogram_oop_framework.views.content_types_views.poll (module), 9
aiogram_oop_framework.views.content_types_views.poll_answer (module), 9
aiogram_oop_framework.views.content_types_views.shipping_query (module), 9

```

AnimationView (class in **E**
 aiogram_oopframework.views.content_types_views.animation),
 EditedChannelPostView (class in
 aiogram_oopframework.views.edited_channel_post_view),
 AnyContentTypeMessageView (class in
 aiogram_oopframework.views.content_types_views.any_content_type_message),
 EditedMessageView (class in
 aiogram_oopframework.views.edited_message_view),
 append_commands (aiogram_oopframework.views.custom_views.command.CommandView
 attribute), 10
 AudioView (class in **G**
 aiogram_oopframework.views.content_types_views.audio),
 GameView (class in aiogram_oopframework.views.content_types_views.g),
 auto_register (aiogram_oopframework.views.message.MessageView
 attribute), 6
 GroupChatCreatedView (class in
 aiogram_oopframework.views.content_types_views.group_chat_7
C
 CallbackQueryView (class in **H**
 aiogram_oopframework.views.callback_query), help_text (aiogram_oopframework.views.message.MessageView
 attribute), 6
 ChannelPostView (class in
 aiogram_oopframework.views.channel_post_view),
 index (aiogram_oopframework.views.message.MessageView
 attribute), 6
 commands (aiogram_oopframework.views.custom_views.command.CommandView
 attribute), 10
 InlineQueryView (class in
 aiogram_oopframework.views.inline_query),
 commands (aiogram_oopframework.views.message.MessageView
 attribute), 5
 CommandView (class in InvoiceView (class in
 aiogram_oopframework.views.custom_views.command), aiogram_oopframework.views.content_types_views.invoice),
 ConnectedWebsiteView (class in
 aiogram_oopframework.views.content_types_views.connected_website),
 LeftChatMemberView (class in
 aiogram_oopframework.views.content_types_views.left_chat_m7
 ContactView (class in
 aiogram_oopframework.views.content_types_views.contact),
 LocationView (class in
 aiogram_oopframework.views.content_types_views.location),
 content_types (aiogram_oopframework.views.message.MessageView
 attribute), 5
 custom_filters (aiogram_oopframework.views.inline_query.InlineQueryView
 attribute), 9
M
 MessageView (class in
 aiogram_oopframework.views.message),
 MigrateFromChatIdView (class in
 aiogram_oopframework.views.content_types_views.migrate_from8
 default_command_case (aiogram_oopframework.views.custom_views.command.CommandView
 attribute), 10
 MigrateToChatIdView (class in
 aiogram_oopframework.views.content_types_views.migrate_to_8
 DeleteChatPhotoView (class in
 aiogram_oopframework.views.content_types_views.delete_chat_photo),
 DiceView (class in aiogram_oopframework.views.content_types_views.dice),
 NewChatMembersView (class in
 aiogram_oopframework.views.content_types_views.new_chat_m8
 DocumentView (class in
 aiogram_oopframework.views.content_types_views.document),

NewChatPhotoView (class in short_description
aiogram_oopframework.views.content_types_views.new_chat_photo), 6

NewChatTitleView (class in state (aiogram_oopframework.views.inline_query.InlineQueryView
aiogram_oopframework.views.content_types_views.new_chat_title), 9

P

PassportDataView (class in aiogram_oopframework.views.content_types_views.sticker),
aiogram_oopframework.views.content_types_views.passport_data), 8

PhotoView (class in aiogram_oopframework.views.content_types_views.successful_payment),
aiogram_oopframework.views.content_types_views.photo), 8

PinnedMessageView (class in T
aiogram_oopframework.views.content_types_views.pinned_message), 8

PollAnswerView (class in U
aiogram_oopframework.views.poll_answer), 9

PollMessageView (class in update_type (aiogram_oopframework.views.content_types_views.animation_poll),
aiogram_oopframework.views.content_types_views.poll), 8

PollView (class in aiogram_oopframework.views.poll), 9

R

regex (aiogram_oopframework.views.message.MessageView
attribute), 5

register () (aiogram_oopframework.views.channel_post_view.ChannelPostView
class method), 6

register () (aiogram_oopframework.views.custom_views.command.CommandView (class in
class method), 10

register () (aiogram_oopframework.views.edited_channel_post_view.EditedChannelPostView
class method), 7

register () (aiogram_oopframework.views.edited_message_view.EditedMessageView
class method), 6

register () (aiogram_oopframework.views.message.MessageView
class method), 6

register_kwargs (aiogram_oopframework.views.inline_query.InlineQueryView
attribute), 9

register_kwargs (aiogram_oopframework.views.message.MessageView
attribute), 5

run_task (aiogram_oopframework.views.inline_query.InlineQueryView
attribute), 9

run_task (aiogram_oopframework.views.message.MessageView
attribute), 5

S

set_my_commands (aiogram_oopframework.views.message.MessageView
attribute), 6

ShippingQueryView (class in
aiogram_oopframework.views.shipping_query), 9