

## Теоретические материалы к занятию 8

Давайте сегодня поговорим о ограничении доступа к боту для различных типов сообщений(личных, группы, канала). Для начала поговорим о сообщениях в группах, так как возможно вы захотите создать бота администратора чата. Мы рассмотрим типовые функции для данного бота, помимо данных функций бот может быть полезен для удаления сообщений.

Создадим типовой шаблон нашего бота

```
import logging

from aiogram import Bot, Dispatcher, executor, types

from config import BOT_TOKEN
from filters import AdminFilter

bot = Bot(token=BOT_TOKEN, parse_mode=types.ParseMode.HTML)
dp = Dispatcher(bot)
logging.basicConfig(level=logging.INFO)

if __name__ == "__main__":
    # Запуск бота
    executor.start_polling(dp, skip_updates=True)
```

Давайте посмотрим, как мы можем реализовать удаление сообщений ботом. Для удаления сообщения мы можем воспользоваться методом `delete_message()`, который принимает `chat_id` и `message_id`. Создадим хендлер.

```
@dp.message_handler()
async def del_m(message: types.Message):
    await bot.delete_message(message.chat.id, message.message_id)
```

Либо можно все это сделать проще

```
@dp.message_handler()
async def del_m(message: types.Message):
    await message.delete()
```

Для управления группами существует огромное количество методов, мы же повторюсь рассмотрим самые основные из них.

Создадим необходимые для примера команды по умолчанию

```

async def set_default_commands(dp: Dispatcher):
    await dp.bot.set_my_commands([
        types.BotCommand('set_photo', 'Установить фото в
группе'),
        types.BotCommand('set_title', 'Установить название
группы'),
        types.BotCommand('set_description', 'Установить описание
группы'),
        types.BotCommand('ro', 'Режим Read Only'),
        types.BotCommand('unro', 'Отключить RO'),
        types.BotCommand('ban', 'Забанить пользователя'),
        types.BotCommand('unban', 'Разбанить пользователя'),
    ])

```

После этого давайте приступим к рассмотрению возможностей нашего бота. Для начала давайте мы сделаем обработчик, который срабатывает при появлении нового пользователя.

```

@dps.message_handler(content_types=types.ContentType.NEW_CHAT_MEMBERS)
async def new_member(message: types.Message):
    await message.reply(f'Привет {message.new_chat_members[0].full_name}')

```

В данном случае наш бот будет здороваться с новым участником.

Рассмотрим теперь вариант если пользователь покинул нашу группу.

```

@dps.message_handler(content_types=types.ContentType.LEFT_CHAT_MEMBER)
async def banned_member(message: types.Message):
    await message.reply(
        f'{message.left_chat_member.full_name} был удален из чата
пользователем {message.from_user.full_name}')

```

В данном обработчике мы видим, что пользователь был удален из чата, давайте добавим проверку, что пользователь сам покинул наш чат.

```

@dps.message_handler(content_types=types.ContentType.LEFT_CHAT_MEMBER)
async def banned_member(message: types.Message):
    if message.left_chat_member.id == message.from_user.id:
        await
message.answer(f'{message.left_chat_member.get_mention(as_html=True)} вышел из чата')
    else:
        await message.reply(

```

```
f'{message.left_chat_member.full_name} был удален из чата пользователем {message.from_user.full_name}'
```

get\_mention - ссылка на нашего пользователя, т.е. если мы щелкнем по ней, то перейдем на пользователя.

Добавим проверку, что пользователь удалил наш бот

```
@dp.message_handler(content_types=types.ContentType.LEFT_CHAT_MEMBER)
async def banned_member(message: types.Message):
    if message.left_chat_member.id == message.from_user.id:
        await
    message.answer(f'{message.left_chat_member.get_mention(as_html=True)} вышел из чата')
    elif message.from_user.id == (await bot.me).id:
        return
    else:
        await message.reply(
            f'{message.left_chat_member.full_name} был удален из чата пользователем {message.from_user.get_mention(as_html=True)}')
```

Давайте теперь перейдем к рассмотрению команд чата. Для начала мы напишем обработчик, который будет устанавливать изображение нашей группы. В него также добавим проверку, является ли сообщение из группы и от администратора.

```
class AdminFilter(BoundFilter):
    async def check(self, message: types.Message):
        member = await
        message.chat.get_member(message.from_user.id)
        return member.is_chat_admin()

class IsGroup(BoundFilter):
    async def check(self, message: types.Message):
        return message.chat.type in (
            types.ChatType.GROUP,
            types.ChatType.SUPERGROUP,
        )

@dp.message_handler(IsGroup(), Command('set_photo',
    prefixes='!/'), AdminFilter())
async def set_new_photo(message: types.Message):
    source_message = message.reply_to_message
    photo = source_message.photo[-1]
```

```
photo = await photo.download(destination=io.BytesIO())
input_file = types.InputFile(path_or_bytesio=photo)
await message.chat.set_photo(photo=input_file)
```

Как мы видим из данного обработчика, мы получаем фото и сохраняем его в переменную, далее мы устанавливаем его с помощью команды `set_photo()`

Сделаем тоже самое и для установки заголовка

```
@dp.message_handler(IsGroup(), Command('set_title',
prefixes='!/'), AdminFilter())
async def set_new_title(message: types.Message):
    source_message = message.reply_to_message
    title = source_message.text
    await message.chat.set_title(title=title)
```

а также для описания нашей группы

```
@dp.message_handler(IsGroup(), Command('set_description',
prefixes='!/'), AdminFilter())
async def set_new_description(message: types.Message):
    source_message = message.reply_to_message
    description = source_message.text
    await message.chat.set_description(description=description)
```

Давайте добавим бан для какого либо пользователя

```
@dp.message_handler(IsGroup(), Command('ban', prefixes='!/'),
AdminFilter())
async def ban_user(message: types.Message):
    member = message.reply_to_message.from_user
    member_id = member.id
    chat_id = message.chat.id
    await message.chat.kick(user_id=member_id)
    await message.reply(f'Пользователь {member.full_name} был забанен')
```

Для бана пользователя используется команда `kick()`, которая принимает `id` пользователя.