

Создание бота модератора для мессенджера Telegram

Эрдман Александр Алексеевич

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

В данной статье рассмотрен процесс создания бота для модерации в мессенджере Telegram. Бот написан на языке программирования Python с использованием библиотеки aiogram. Результатом исследования является бот модератор, который отвечает за порядок во время диалога.

Ключевые слова: бот, telegram, python, aiogram

Creating a moderator chatbot for Telegram messenger

Erdman Alexander Alekseevich

Sholom-Aleichem Priamursky State University

Student

Abstract

This article discusses the process of creating a bot for moderation in the Telegram messenger. The bot is written in the Python programming language using the aiogram library. The result of the study is a moderator bot, which is responsible for the order during the dialogue.

Keywords: bot, telegram, python, aiogram

1 Введение

1.1 Актуальность

В настоящее время для мобильных устройств распространены различные мессенджеры для общения. В частности, таким мессенджером является Telegram. Зачастую в больших групповых чатах необходим модератор, который отвечает за фильтрацию сообщений в чате, соблюдением порядка и выдачей определённых санкций за несоблюдение правил чата. Для маленьких групп удобно иметь в виде модератора человека, но в больших группах – человек модератор зачастую не справляется с количеством пользователей. В таких случаях в чаты добавляют ботов модераторов, которые автоматизируют работу человека, позволяя тем самым эффективнее следить за соблюдением правил в чате. В связи с данной популярностью бота-модератора, создание ботов такого типа является актуальным. О процессе создания такого бота будет изложен в статье.

1.2 Обзор исследований

О.Ю. Косенко и Б.З. Заврумова провели анализ использования чат-ботов в целях маркетинговой коммуникации [1]. Е.И. Еськин и Т.С. Катермина реализовали чат-бота для любых коммерческих или информативных компаний [2]. С.Э. Исхамова рассмотрела аспекты образовательного процесса, которые можно автоматизировать или сделать их более эффективными с помощью внедрения чат-ботов [3]. О.А. Румянцева рассмотрела особенности применения искусственного интеллекта в образовательном процессе на примере чат-ботов [4]. А.И. Джангаров, Д.Ш. Калхиташвили и Д.М. Магамедова рассмотрели наиболее востребованные среды разработки и технологии создания чат-ботов различного назначения [5].

1.3 Цель исследования

Целью исследования является создание Telegram бота на языке программирования Python для соблюдения правил чата.

2 Материалы и методы

Бот разработан на языке программирования Python с использованием библиотеки aiogram. В качестве IDE используется PyCharm.

3 Результаты и обсуждения

Перед созданием необходимо установить библиотеку aiogram через команду `pip install` в PyCharm, а также получить токен бота через Telegram бота «BotFather». Для получения токена отправляется сообщение о создании бота, далее получается токен. Для хранения токена будет использоваться отдельный файл конфигурации в проекте `config.py`. Код функционала бота модератора содержится в файле `bot.py`. Бот будет находится в одной из групп. Для того, чтобы он мог читать и отвечать на сообщения в группах, нужно через BotFather изменить настройки приватности через команду `/setprivacy` с «Enabled» на «Disable». Первичная настройка завершена.

В файле `config.py` создаётся строка с токеном бота:

```
ТОКЕН = "имя_токена"
```

Далее в файле `bot.py` прописывается функционал бота для его проверки работоспособности (рис. 1).

```
import config
import logging

from aiogram import Bot, Dispatcher, executor, types

logging.basicConfig(level=logging.INFO)

bot = Bot(token=config.TOKEN)
dp = Dispatcher(bot)

@dp.message_handler()
async def echo(message: types.Message):
    await message.answer(message.text)

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

Рисунок 1. Первичный функционал бота

В данном файле прописываются импортируемые библиотеки и необходимые модули aiogram; указывается уровень логирования; получение токена ботом; в обращение обработчика вызывается декоратор `message_handler`, который отвечает за обработку сообщений. Обработка событий происходит с помощью метода long-polling, когда пользователи отправляют сообщения и только после этого сервер бота начинает работу.

После первого запуска бот работает в режиме «эхо», то есть дублирует все отправленные ему сообщения. После проверки работы, бот добавляется в группу, в которой боту выдаются администраторские права. У группы есть собственный ID. Данный ID заносится в файл с токеном бота.

Далее меняется функция `echo`, на функцию `filter_messages`, которая позволяет фильтровать сообщения с определёнными словами. Данная функция будет удалять те сообщения, в которых было написано оскорбительное или какое-либо нежелательное слово:

```
async def filter_messages(message: types.Message):
    if "нежелательное слово" in message.text:
        await message.delete()
```

Примером слова будет «нежелательное слово». При запуске бота и попытки отправить сообщение, в котором содержится «нежелательное слово», то бот будет удалять данное сообщение. Данный список «нежелательных слов» можно расширить, в том числе добавить «нежелательные» ссылки на какие-либо ресурсы.

Другой функцией бота является функция удаления надписи о том, что к группе был подключён новый пользователь:

```
@dp.message_handler(content_types=["new_chat_members"])
async def on_user_joined(message: types.Message):
    await message.delete()
```

В данной функции через декоратор идёт указание типа контента, а именно тип событие «присоединение нового пользователя». Далее производится удаление сообщения о присоединении.

Для более полной функциональности бота добавляется команда удаления пользователя из чата и запрет на присоединение (ban) за нарушение правил чата (рис.2). Для декоратора указываются аргументы `commands` (команды на которые реагирует бот) и `commands_prefix` (префиксы для команды), а также аргумент `is_admin=True`, который позволяет боту иметь права администратора. С помощью методов `delete_message` и `kick_chat_member` происходит удаление нежелательного сообщения пользователя и отключение пользователя от чата. Конструкция `if not message.reply_to_message` позволяет вызывать команду бана пользователя в ответ на его «плохое» сообщение. Для работы функции понадобится создать файл `filter.py`, где будут храниться фильтры функции для определения нарушения. В файле `bot.py` заранее подключается данный фильтр. В фильтре объявляется ключ для определения администратора бота, присвоение администрирования боту и функция `check` проверки сообщения (рис. 3).

```
@dp.message_handler(is_admin=True, commands=["ban"], commands_prefix="!/")
async def cmd_ban(message: types.Message):
    if not message.reply_to_message:
        await message.reply("Ответом на сообщении!")
        return

    await message.bot.delete_message(chat_id=config.GROUP_ID, message.message_id)
    await message.bot.kick_chat_member(chat_id=config.GROUP_ID, user_id=message.reply_to_message.from_user.id)
    await message.reply_to_message.reply("Пользователь забанен!")
```

Рисунок 2. Функция бана пользователей

```
from aiogram import types
from aiogram.dispatcher.filters import BoundFilter
class IsAdminFilter(BoundFilter):
    key = "is admin"
    def __init__(self, is_admin):
        self.is_admin = is_admin
    async def check(self, message: types.Message):
        member = await message.bot.get_chat_member(message.chat.id, message.from_user.id)
        return member.is_chat_admin()
```

Рисунок 3. Фильтр бота

После всех действий тестируется бот (рис. 4)

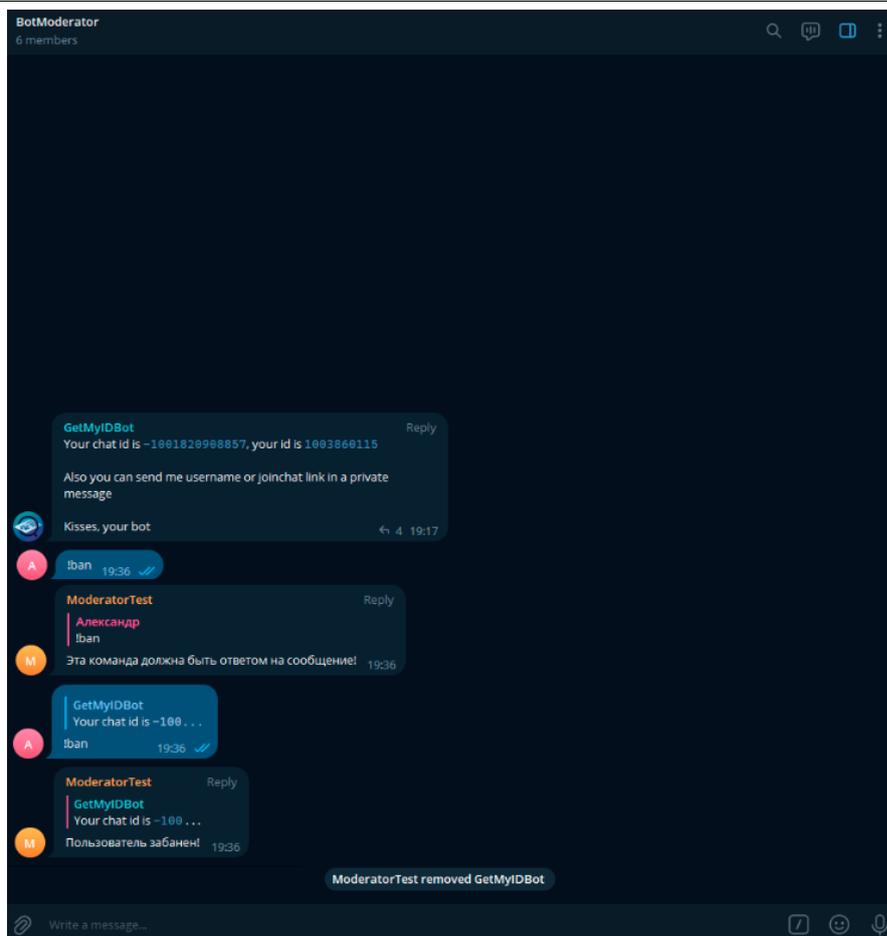


Рисунок 4. Результат работы бота модератора

Выводы

Таким образом был разработан бот модератор чата на Python с использованием библиотеки `iaogram` в мессенджере Telegram со следующими функциями: удаление участника чата, удаление сообщений, в которых есть запрещённые слова, а также удаление надписей о присоединении нового участника.

Библиографический список

1. Косенко О.Ю., Заврумова Б.З. Чат-бот как инструмент цифровой маркетинговой коммуникации (анализ использования чат-ботов в России и Австрии) // В сборнике: Инноватика: современные технологии модернизации общества. Материалы III Региональной научно-практической конференции с международным участием. 2021. С. 27-36.
2. Еськин Е.И., Катермина Т.С. Разработка чат-бота, его возможности и организация полноценного чата с оператором // Научный электронный журнал Меридиан. 2020. № 15 (49). С. 21-23.
3. Исхакова С.Э. Автоматизация процесса обучения с помощью технологии чат-бот // Вестник магистратуры. 2022. № 4-4 (127). С. 25-26.
4. Румянцева О.А. Применение чат-ботов в образовательном процессе //

Молодой ученый. 2022. № 47 (442). С. 424-427.

5. Жангаров А.И., Калхиташвили Д.Ш., Магамедова Д.М. Среды разработки и технологии создания чат-ботов // Тенденции развития науки и образования. 2021. № 80-2. С. 99-102.