

**Московский государственный технический
университет им. Н.Э. Баумана**

**Факультет «Информатика и системы управления»
Кафедра ИУ5 «Системы обработки информации и управления»**

Курс «Базовые компоненты интернет-технологий»

**Отчет по лабораторной работе №6
«Разработка простого бота для Telegram с использованием языка Python»**

Выполнил:

студент группы ИУ5-32

Вольвач Михаил

Подпись и дата: 29.12.2022

Проверил:

преподаватель каф. ИУ5

Гапанюк Ю. Е.

Подпись и дата: 29.12.2022

Москва, 2022

Задание

Разработайте простого бота для Telegram. Бот должен использовать функциональность создания кнопок.

Текст программы

Файл bot.py

```
import asyncio
import logging
import random
import sys

from typing import Any, Dict

import aiogram
from aiogram import Bot, Dispatcher, F
from aiogram.dispatcher.filters import Command
from aiogram.dispatcher.fsm.context import FSMContext
from aiogram.dispatcher.fsm.state import StatesGroup, State
from aiogram.dispatcher.router import Router
from aiogram.types import Message, ReplyKeyboardRemove, ReplyKeyboardMarkup,
KeyboardButton
from aiogram.utils.markdown import text, hide_link

from parser import resource_parser

logging.basicConfig(level=logging.INFO)
dp = Dispatcher()
form_router = Router()

class Form(StatesGroup):
    resource = State()

@form_router.message(Command(commands=["start"]))
async def form_start(message: Message, state: FSMContext):
    await state.set_state(Form.resource)
    await message.reply(
        "Hi there! What do you want to see?",
        reply_markup=ReplyKeyboardMarkup(
            keyboard=[
                [
                    KeyboardButton(text="Posts"),
                    KeyboardButton(text="Photos"),
                    KeyboardButton(text="/cancel")
                ]
            ],
            resize_keyboard=True
        )
    )
```

```

@form_router.message(Command(commands=["cancel"]))
@form_router.message(F.text.casefold() == "cancel")
async def form_cancel_handler(message: Message, state: FSMContext):
    current_state = await state.get_state()
    if current_state is None:
        return

    logging.info("Cancelling state %r", current_state)
    await state.clear()
    await message.answer(
        "Cancelled.",
        reply_markup=ReplyKeyboardRemove()
    )

@form_router.message(Form.resource)
async def process_posts(message: Message, state: FSMContext):
    await state.set_state(Form.resource)

    response_data = resource_parser(message.text)
    number = random.randrange(1, len(response_data))

    if message.text == "Posts":
        await message.answer(
            f"<b>{response_data[number]['title']}</b>\n{response_data[number]['body'
' ]}",
        )
    elif message.text == "Photos":
        await message.answer(
            f"<b>{response_data[number]['title']}</b>{hide_link(response_data[numbe
r]['url'])}"
        )

    await message.answer(
        "Do you want to go to the start?",
        reply_markup=ReplyKeyboardMarkup(
            keyboard=[
                [
                    KeyboardButton(text="/start")
                ]
            ],
            resize_keyboard=True
        )
    )

async def main():
    bot = Bot(token="5972437920:AAGcRIT6V0aJKLkn1MN3qqpVSVHvbQ8tqCE",
parse_mode="HTML")
    dp.include_router(form_router)

    await dp.start_polling(bot)

```

```
if __name__ == "__main__":
    asyncio.run(main())
```

Файл parser.py

```
import requests
```

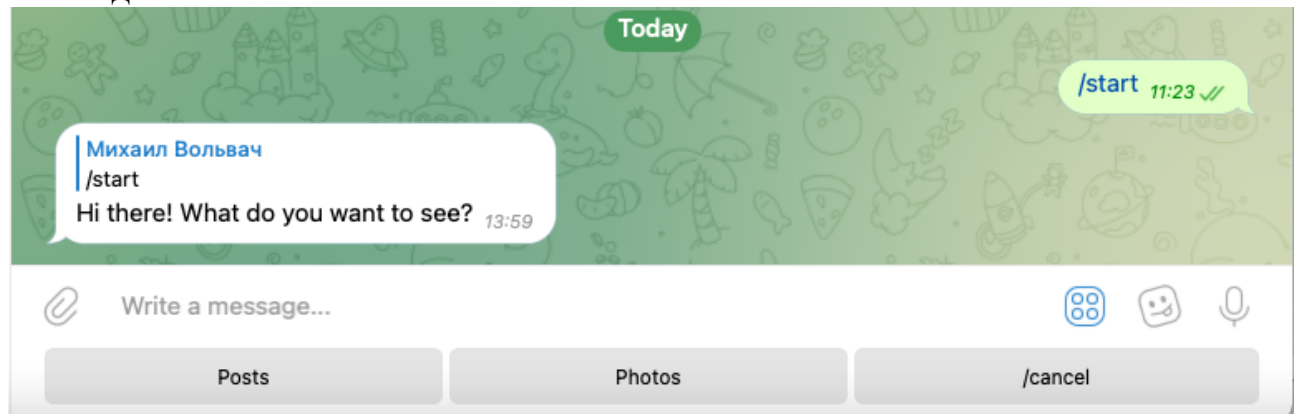
```
def resource_parser(contentType="Posts"):
    headers = {
        'User-Agent': 'Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/84.0.4147.89 Safari/537.36',
        'Accept':
'text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=
0.8,application/signed-exchange;v=b3;q=0.9',
        'Accept-Language': 'ru-UA,ru;q=0.9,en-US;q=0.8,en;q=0.7,ru-RU;q=0.6',
    }
    link = "https://jsonplaceholder.typicode.com/"

    if contentType == "Posts":
        link += "posts"
    elif contentType == "Comments":
        link += "comments"
    elif contentType == "Albums":
        link += "albums"
    elif contentType == "Photos":
        link += "photos"
    elif contentType == "To Do":
        link += "todos"
    elif contentType == "Users":
        link += "users"
    else:
        return ""

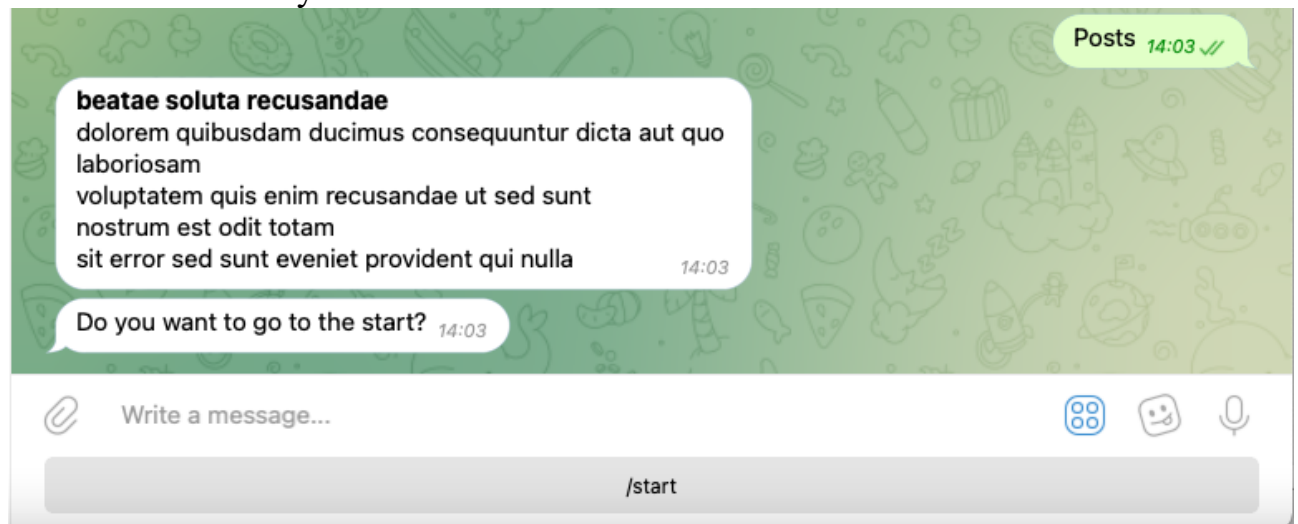
    data = requests.get(link).json()
    return data
```

Пример работы программы

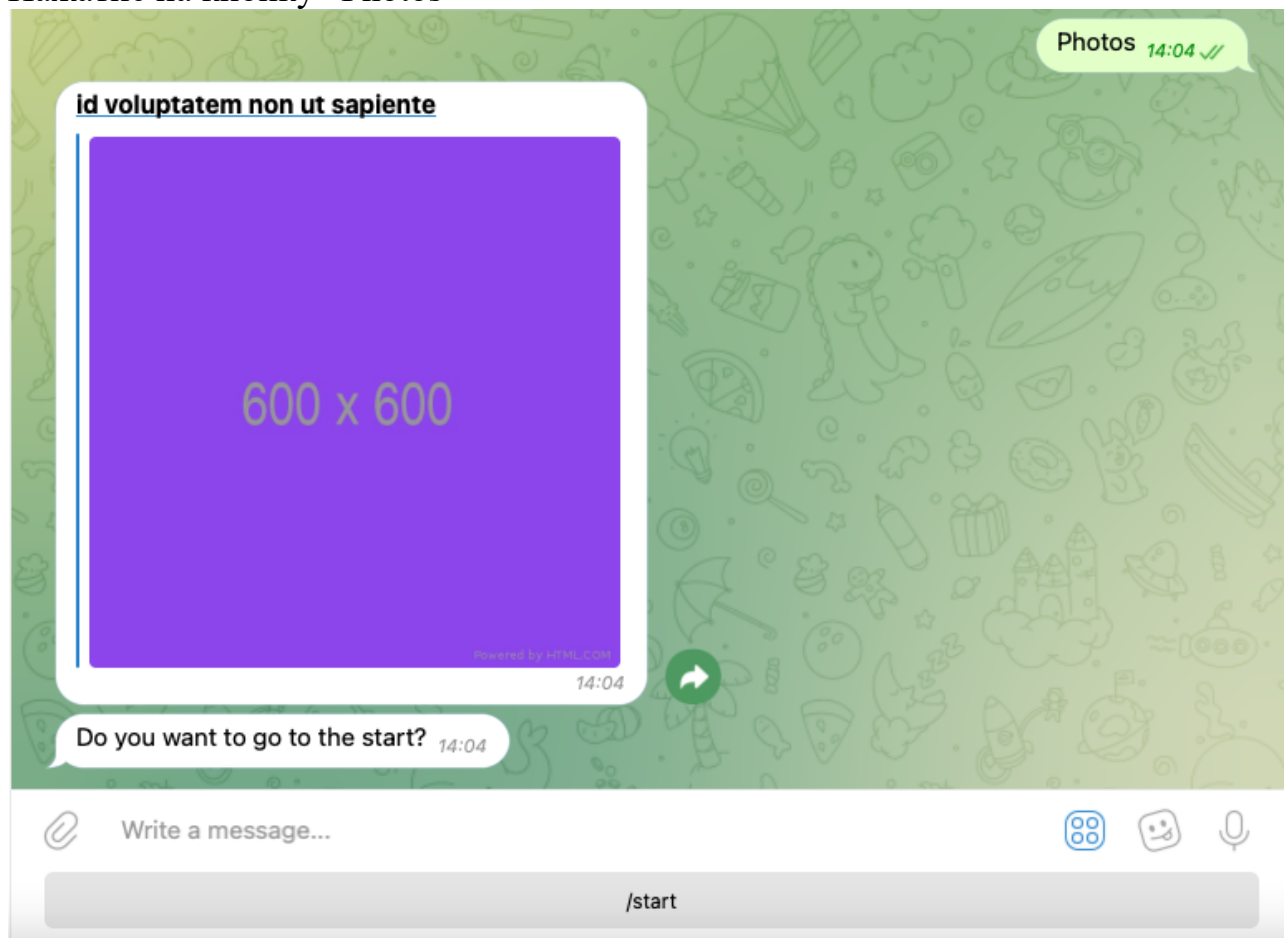
Команда `/start`



Нажатие на кнопку “Posts”



Нажатие на кнопку “Photos”



Команда [/cancel](#)

