import asyncio

import requests

from aiogram import Bot, Dispatcher

from aiogram.filters import Command

from aiogram.types import Message

from config import TOKEN, THE\_CAT\_API\_KEY

# Вставьте сюда ваш токен телеграм-бота и API-ключ для TheCatAPI

bot = Bot(token=TOKEN)

dp = Dispatcher()

# Функция для получения списка пород кошек

def get\_cat\_breeds():

url = "https://api.thecatapi.com/v1/breeds"

headers = {"x-api-key": THE\_CAT\_API\_KEY}

response = requests.get(url, headers=headers)

return response.json()

# Функция для получения картинки кошки по породе

def get\_cat\_image\_by\_breed(breed\_id):

url = f"https://api.thecatapi.com/v1/images/search?breed\_ids={breed\_id}"

headers = {"x-api-key": THE\_CAT\_API\_KEY}

response = requests.get(url, headers=headers)

data = response.json()

return data[0]['url']

# Функция для получения информации о породе кошек

def get\_breed\_info(breed\_name):

breeds = get\_cat\_breeds()

for breed in breeds:

if breed['name'].lower() == breed\_name.lower():

return breed

return None

@dp.message(Command("start"))

async def start\_command(message: Message):

await message.answer("Привет! Напиши мне название породы кошки, и я пришлю тебе её фото и описание.")

@dp.message()

async def send\_cat\_info(message: Message):

breed\_name = message.text

breed\_info = get\_breed\_info(breed\_name)

if breed\_info:

cat\_image\_url = get\_cat\_image\_by\_breed(breed\_info['id'])

info = (

f"Breed: {breed\_info['name']}\n"

f"Origin: {breed\_info['origin']}\n"

f"Description: {breed\_info['description']}\n"

f"Temperament: {breed\_info['temperament']}\n"

f"Life Span: {breed\_info['life\_span']} years"

)

await message.answer\_photo(photo=cat\_image\_url, caption=info)

else:

await message.answer("Порода не найдена. Попробуйте еще раз.")

async def main():

await dp.start\_polling(bot)

if \_\_name\_\_ == '\_\_main\_\_':

asyncio.run(main())