



ЗАДАНИЕ 4. ДОПИСЫВАЕМ КОД САМОСТОЯТЕЛЬНО

Вы, наверное, заметили, что два обработчика исполняют по два декоратора. Сделайте ещё два обработчика, чтобы на каждое настроение бот присылал вполне определённый стикер из словаря.

БОТ С ИСКУССТВЕННЫМ ИНТЕЛЛЕКТОМ

Мы научились по-разному отвечать на сообщение пользователя в зависимости от его содержания, но для этого пришлось писать разные обработчики и пытаться предусмотреть, что может сообщить нам пользователь.

Можно поступить по-другому: предоставить пользователю возможность общаться с искусственным интеллектом через наше приложение. Для этого нужно подключить его к нашему боту.

Рассмотрим пример подключения YandexGPT, подключение других языковых моделей аналогично.

В документации API YandexGPT подробно рассказано, как это сделать. У вас должен быть платёжный аккаунт в Yandex Cloud («Яндекс.Облако»), даже если вы собираетесь использовать только бесплатные сервисы. Если такого аккаунта нет, его несложно завести.

Теперь нам нужно получить токен — секретный API-ключ и ID каталога с ролью user. Чтобы получить токен для использования API YandexGPT, необходимо выполнить следующие шаги:

1. В личном кабинете Yandex Cloud получить идентификатор каталога, по умолчанию создаётся каталог *default*.
2. Зайти в этот каталог, перейти в раздел «Сервисные аккаунты» и нажать «Создать сервисный аккаунт». Необходимо задать имя, описание и выбрать роль *ai.languageModels.user*.
3. Зайти внутрь созданного сервисного аккаунта и нажать «Создать API-ключ».
4. Ввести описание, чтобы было понятно, где используется ключ.
5. Будет выведен ключ. Важно сохранить секретный ключ перед закрытием этого окна, иначе его не получится увидеть снова.

Следует помнить, что использование токена должно быть безопасным и соответствовать правилам конфиденциальности. Это значит, что секретные данные нельзя размещать в открытых источниках, в том числе в коде бота. Поэтому сохраним ключ и ID каталога в файл с токеном телеграм-бота.

Создадим простого бота, который при старте предлагает пользователю написать запрос к искусственному интеллекту, затем переадресует его YandexGPT и возвращает ответ в виде сообщения пользователю. Как сделать основную часть бота, вы уже знаете, осталось написать часть, отвечающую за общение с искусственным интеллектом.

Напишем функцию `yandex_gpt(message)`, принимающую сообщение пользователя и возвращающую ответ искусственного интеллекта. Обратите внимание, что теперь мы читаем из файла *token.txt* все секретные ключи.

Теперь наш бот выглядит следующим образом:

```
import asyncio
import requests
from aiogram import Bot, Dispatcher
from aiogram.filters import Command
from aiogram.types import Message

with open('token.txt') as file:
    API_TOKEN = file.readline().rstrip()
    FOLDER_ID = file.readline().rstrip()
    YANDEX_GPT_KEY = file.readline().rstrip()

URL = 'https://llm.api.cloud.yandex.net/foundationModels/v1/completion'
dp = Dispatcher()
# В коде ссылку необходимо набирать в одну строку.
```

```
def yandex_gpt(message):
```

```
    '''
```

```
    Функция для обращения к ИИ и получения ответа
```

```
    :param message: текст вопроса
```

```
    :return: ответ в формате json
```

```
    '''
```

```
    response = requests.post(
```

```
        URL,
```

```
        headers={
```

```
            'Authorization': f'Api-Key {YANDEX_GPT_KEY}',
```

```
            'x-folder-id': FOLDER_ID
```

```
        },
```

```
        json={
```

```
            'modelUri':
```

```
            f'gpt://{FOLDER_ID}/yandexgpt/latest',
```

```
            'completionOptions': {
```

```
                'stream': False,
```

```
                'temperature': 0.6
```

```
            },
```

```
            'messages': [
```

```
                {
```

```
                    'role': 'system',
```

```
                    'text': 'Задавай вопрос. Отвечу'
```

```
                },
```

```
                {
```

```
                    'role': 'user',
```

```
                    'text': message
```

```
                }
```

```
            ]
```

```
        }
```

```
    )
```

```
    return response.json()
```

```
async def main():
```

```
    bot = Bot(token=API_TOKEN)
```

```
    await dp.start_polling(bot)
```

```
@dp.message(Command('start'))
```

```
async def cmd_start(msg: Message):
```

```
    await msg.answer(f'Привет, {msg.from_user.first_name}!')
```

```
    f'Задай текстовый вопрос, я постараюсь ответить.'
```



```

gdp.message()
async def ai_messaging(msg: Message):
    try:
        # пока бот думает, пусть делает вид, что печатает
        await msg.bot.send_chat_action(msg.chat.id,
            'typing')
        # если у сообщения есть текст
        if msg.text:
            # получим ответ на свой вопрос у ИИ
            text = yandex_gpt(msg.text)
            answer = text['result']['alternatives'][0]
                ['message']['text']
            await msg.answer(answer)
        else:
            await msg.answer('Пожалуйста, задай свой '
                'текстовый вопрос!')
    except TypeError:
        await msg.answer('Что-то пошло не так...')

if __name__ == '__main__':
    asyncio.run(main())

```



Воспроизведите код и проверьте работу бота.

Рис. 79. Наш бот отвечает на вопрос с помощью искусственного интеллекта