Урок 26. Обработчики, фильтры и расширение функций Telegram-бота

9 класс · Урок 26 из 34 · 45 минут

Вся программа 9 класса

Обработчики в aiogram. На уроке для 9 класса разбираем фильтры и роутер, добавляем ответы Telegram-бота на разные сообщения и проверяем стикеры.

Содержание урока

- 01. Цель урока
- 02. Ход урока
- 03. Этап 1. Вступление: Не та ветка
- 04. Этап 2. Как сообщение находит ответ
- 05. Этап 3. Лаборатория обработчиков
- 06. Этап 4. Приёмка чужого бота
- 07. Этап 5. Главное и проверка понимания

01. Цель урока

Научиться разделять функции Telegram-бота по обработчикам и проверять, как он отвечает на разные сообщения.

02. Ход урока

Этап	Время
1. Вступление: Не та ветка	5 мин
2. Как сообщение находит ответ	11 мин
3. Лаборатория обработчиков	17 мин
4. Приёмка чужого бота	8 мин
5. Главное и проверка понимания	4 мин
Всего	45 мин

03. Этап 1. Вступление: Не та ветка 5 мин

> ГОВОРИМ УЧЕНИКАМ
>
> Бот школьного клуба умеет отвечать на любое сообщение: «Спасибо, я вас услышал». Ребята добавили отдельный ответ на «грустно», но в чате всё равно появляется старая фраза. Почему новый ответ может так и не дойти до собеседника?
>
> Сегодня разберём, как фильтры в aiogram помогают выбрать обработчики. Соберём несколько ответов и проверим, как бот ведёт себя на границе между командами, настроением и обычным разговором.

04. Этап 2. Как сообщение находит ответ 11 мин

> ГОВОРИМ УЧЕНИКАМ
>
> Общий обработчик подходит под любое сообщение. Если он стоит выше специальных, библиотека может выбрать его первым. Поэтому сначала удобно проверять конкретные команды и фразы, а общий ответ оставлять последним. Это объясняет, почему новый ответ мог не сработать.
>
> Диспетчер получает новое событие. Роутер объединяет обработчики на одну тему, например всё, что отвечает за разговор о настроении. Фильтр проверяет условие: начинается ли сообщение с команды или совпадает ли текст со словом «грустно». Обработчик выполняет действие после совпадения. Для выбора двух разных стикеров можно хранить соответствие «настроение → идентификатор» в словаре и использовать его в двух обработчиках.
>
> Вместе с текстом Telegram передаёт метаданные, то есть сведения о сообщении и его отправителе. Из поля `first_name` бот может узнать, как подписан профиль. Человек мог указать там псевдоним, поэтому лучше спросить, подходит ли это обращение. Для задания мы договоримся о трёх отдельных действиях: команда спрашивает об имени, затем бот предлагает написать настроение, а нужный ответ выбирается по тексту. Этот бот не хранит этап разговора, поэтому среагирует на «да» или «нет», даже если ещё не спрашивал имя.

Вход	Что проверяет фильтр	Какое действие выполняет бот
Команда /start	Получена команда /start	Спрашивает, подходит ли имя из профиля
«да» или «нет»	Текст совпал с «да» или «нет»	Задаёт вопрос о настроении
«весело» или «грустно»	Выбрана одна из двух фраз	Отправляет цветок и стикер по настроению
Любое другое сообщение	Ранее не подошло ни одно правило	Подсказывает варианты

05. Этап 3. Лаборатория обработчиков 17 мин

> ГОВОРИМ УЧЕНИКАМ
>
> Сейчас у вас есть две разные задачи. Сначала нужно понять, какое правило поймает сообщение, а затем проверить, отправил ли бот нужный ответ. Если ответ удивил, ищите первую развилку, где код пошёл по другому пути.

Задание. Соберите разговор для двух настроений

Откройте файл бота из урока 25. Сохраните `main` и точку входа. Вместо прежнего блока импортов, диспетчера и обработчика используйте фрагмент ниже. Это демонстрационный разговор, в котором бот реагирует на точные варианты и не хранит стадию диалога.

1. Выберите по цветку для каждого настроения и придумайте ответ, который ему подходит.
2. Запустите бота со своим учебным токеном. Чтобы подготовить стикеры, отправьте ему два обычных стикера. Бот вернёт их `file_id` — идентификаторы для повторной отправки. Скопируйте разные значения в словарь `STICKERS` и перезапустите программу.
3. В парах проведите по одному разговору с каждым настроением. Сохраните только команды и ответы без имён и секретных данных.

```
import asyncio
import os

from aiogram import Bot, Dispatcher, F, Router
from aiogram.filters import CommandStart
from aiogram.types import Message

dp = Dispatcher()
router = Router()
dp.include_router(router)

STICKERS = {
    "весело": "",
    "грустно": "",
}

@router.message(CommandStart())
async def ask_name(message: Message):
    if message.from_user:
        name = message.from_user.first_name
        await message.answer(f"К тебе можно обращаться «{name}»? Ответь
«да» или «нет».")
```

```
    else:
        await message.answer("Ответь одним словом: «весело» или
«грустно».")

@router.message(F.sticker)
async def show_sticker_id(message: Message):
    await message.answer(message.sticker.file_id)

@router.message(F.text.casefold().in_({"да", "нет"}))
async def ask_mood(message: Message):
    if message.text.casefold() == "нет":
        await message.answer("Хорошо, обойдёмся без имени.")
    await message.answer("Как настроение? Ответь: «весело» или
«грустно».")

@router.message(F.text.casefold() == "весело")
async def cheerful(message: Message):
    await message.answer("Держи подсолнух ")
    if STICKERS["весело"]:
        await message.answer_sticker(STICKERS["весело"])

@router.message(F.text.casefold() == "грустно")
async def sad(message: Message):
    await message.answer("Вот тебе тюльпан ")
    if STICKERS["грустно"]:
        await message.answer_sticker(STICKERS["грустно"])

@router.message()
async def other(message: Message):
    await message.answer("Напиши «весело», «грустно» или начни с /start.")
```

Результат: бот спрашивает, можно ли обращаться по имени из профиля, и выдаёт разные цветы и стикеры для двух настроений. Объясните, какие фильтры выбрали эти ответы.

Для учителя

Проверьте, что два `file_id` получены тем же ботом. Пустое значение оставляет текстовый ответ с цветком. Если у класса один токен, выполните живую проверку на одном процессе. `casefold()` приводит текст к форме для сравнения без учёта регистра. Обработчик `show_sticker_id` нужен для настройки. Когда идентификаторы внесены, его можно убрать. Спросите, чего не хватает для настоящего пошагового диалога: состояние должно храниться отдельно для каждого собеседника.

06. Этап 4. Приёмка чужого бота 8 мин

ГОВОРИМ УЧЕНИКАМ

Проверка на двух ожидаемых ответах покажет, что основные функции работают. Сложнее предугадать, что случится, когда человек начнёт с середины разговора или ответит своими словами. Для этого пригодятся пробы, которые автор не предусмотрел.

Задание. Найдите границу

Выберите два сообщения, которые могли бы отправить настоящие пользователи. Одно должно быть нестандартным ответом на вопрос, другое — сообщением другого типа или с непривычным оформлением. До проверки запишите, какой ответ считаете уместным. Обменяйтесь сообщениями с соседями и предложите одну правку, которая улучшит поведение, сохранив два рабочих случая.

Для учителя

При «Весело» фильтр с `casefold()` сработает, а при «мне грустно» сработает общий ответ. Если предложат использовать поиск подстроки, попросите проверить фразу «мне не грустно». Принять можно уточняющий ответ, отдельную кнопку или хранение состояния, если ученики объяснят, какую проблему решает правка.

07. Этап 5. Главное и проверка понимания 4 мин

ГОВОРИМ УЧЕНИКАМ

Когда бот отвечает не на ту фразу, сначала проверьте условия выбора. Часто достаточно понять, какой фильтр совпал раньше или почему точное совпадение не нашлось.

Главное за урок

- Роутер собирает правила, фильтр проверяет условие, обработчик выполняет действие.
- Имя из профиля позволяет предложить обращение, которое человек может подтвердить.
- Запасной ответ лучше проверять вместе со специальными ответами: от порядка зависит, какой из них сработает.

Итоговые вопросы

1. Почему запасной обработчик поставили последним?
2. Для чего хранить стикеры в словаре, если ответы на настроения уже разнесены по функциям?

Для учителя

1. Иначе он может первым подойти к сообщению и перехватить его. 2. По словарю проще увидеть, какой стикер соответствует настроению, и поменять его без поиска по всему коду.

Источники

- aiogram 3. Фильтры · Проверено 05.11.2026. Документация показывает выражения F для проверки свойств сообщения.
- aiogram 3. Отправка стикера · Проверено 05.11.2026. Стикер, уже известный Telegram, можно отправить по file_id.