Урок 25. Простой Telegram-бот на Python через API

9 класс · Урок 25 из 34 · 45 минут

Вся программа 9 класса

Telegram-бот на Python для 9 класса. Разбираем Bot API, токен и точку входа, запускаем бота через aiogram и asyncio и проверяем ответ на команду.

Содержание урока

- 01. Цель урока
- 02. Ход урока
- 03. Этап 1. Вступление: Бот замолчал
- 04. Этап 2. Чат, сервер и программа
- 05. Этап 3. Как выглядит первый ответ
- 06. Этап 4. Первый запуск в парах
- 07. Этап 5. Приёмка и поиск сбоя
- 08. Этап 6. Главное и проверка понимания

01. Цель урока

Понять, как Python-программа получает сообщения из Telegram, и запустить простого бота с одной командой.

02. Ход урока

Этап	Время
1. Вступление: Бот замолчал	5 мин
2. Чат, сервер и программа	8 мин
3. Как выглядит первый ответ	4 мин
4. Первый запуск в парах	17 мин
5. Приёмка и поиск сбоя	7 мин
6. Главное и проверка понимания	4 мин

Этап	Время
Всего	45 мин

03. Этап 1. Вступление: Бот замолчал 5 мин

ГОВОРИМ УЧЕНИКАМ

Ребята сделали бота для выбора темы фотопрогулки. Пока ноутбук был включён и программа работала, бот отвечал. После того как программу закрыли, чат остался на месте, но ответа больше не было. Какая часть перестала работать?

Сегодня создадим простого Telegram-бота на Python. Посмотрим, кто принимает сообщения и что должно быть запущено, чтобы на них пришёл ответ.

Для учителя

Запишите две версии класса. Прямой ответ прозвучит в объяснении следующего этапа.

04. Этап 2. Чат, сервер и программа 8 мин

ГОВОРИМ УЧЕНИКАМ

Чат хранится на стороне Telegram, поэтому его по-прежнему видно в приложении. В нашем примере отвечала отдельная программа на ноутбуке. Когда её закрыли, она перестала забирать новые сообщения и отправлять ответы. Если программа будет работать на сервере, школьный ноутбук можно выключить: сообщения будет обрабатывать сервер.

Telegram предоставляет Bot API — набор команд, по которым программа может запрашивать новые события и отправлять сообщения. Имя бота помогает человеку найти чат. Для обращения к Bot API нужен токен: секретный код, который даёт программе право действовать от имени бота. Токен выдаёт BotFather, служебный бот Telegram. Если токен попал в общий чат или в открытый код, его следует отозвать в BotFather и заменить.

Библиотека aiogram помогает Python работать с Bot API. Когда команда приходит, диспетчер передаёт событие функции, которая должна на него ответить. Такую функцию называют обработчиком. В этом уроке у нас одна команда и один обработчик. В конце файла есть точка входа: это место, с которого при запуске начинается работа нашей программы. В ней `asyncio.run` запускает функцию `main`. Когда программа ждёт ответ от Telegram, `await` позволяет выполнять другие задачи, пока ответ не пришёл.

Кто или что	Роль при запуске
BotFather	Регистрирует бота и выдаёт секретный токен

Кто или что	Роль при запуске
Telegram Bot API	Принимает запросы программы и передаёт события
aiogram и диспетчер	Получают события и передают их обработчику
asyncio и точка входа	Запускают асинхронную работу программы

05. Этап 3. Как выглядит первый ответ 4 мин

> ГОВОРИМ УЧЕНИКАМ
>
> Пока ответ пишется в программе, его легко проверить. Отправим команду один раз, посмотрим на чат и терминал, а затем изменим текст в коде и повторим запуск. Так мы увидим, в каком месте задаётся фраза.

Для учителя

Покажите один заранее подготовленный запуск. Попросите класс предсказать, изменится ли ответ после правки кода без перезапуска.

06. Этап 4. Первый запуск в парах 17 мин

> ГОВОРИМ УЧЕНИКАМ
>
> Мы начнём с команды `/start`. Получив её, программа отправит заранее написанный ответ. Для школьного клуба это уже полезно: одним сообщением можно подсказать участнику, с чего начать.

Задание. Запустите один полезный ответ

1. Выберите, кому адресован бот вашего клуба. Напишите ответ на `/start` из одной-двух фраз. Из него должно быть понятно, какое первое действие ждёт новичка.
2. Если aiogram ещё не установлен, на компьютере с Python 3.10 или новее выполните `python -m pip install 'aiogram>=3,<4'`. В Windows PowerShell команда установки такая же. Если Python в вашей системе запускается как `python3` или `py`, используйте это имя и при запуске файла.
3. Если вы работаете со своим разрешённым учебным аккаунтом, откройте официального `@BotFather`, отправьте `/newbot` и задайте имя и свободное имя пользователя, которое заканчивается на `bot`. Скопируйте токен только в свой терминал. В общем классе возьмите отдельный учебный токен у учителя.
4. Сохраните код как `bot.py`. Замените ответ внутри кавычек своим. В терминале на macOS или Linux задайте `export TELEGRAM_BOT_TOKEN='ВАШ_ТОКЕН'`. В Windows

PowerShell задайте `$env:TELEGRAM_BOT_TOKEN = 'ВАШ_ТОКЕН'`. В этом же терминале выполните `python bot.py`.

5. Откройте чат со своим ботом, отправьте `/start` и сравните полученный текст с ожидаемым. Поменяйтесь с напарником ролями: один проверяет чат, другой наблюдает за терминалом.

```
import asyncio
import os

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

dp = Dispatcher()

@dp.message(CommandStart())
async def greet(message: Message):
    await message.answer("Привет! Выбери тему для фотопрогулки и запиши её
в блокнот.")

async def main():
    token = os.environ["TELEGRAM_BOT_TOKEN"]
    async with Bot(token=token) as bot:
        await dp.start_polling(bot, close_bot_session=False)

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

Результат: подтверждённый ответ в учебном чате и объяснение, кто доставил команду и кто составил ответ. Сохраните только текст команды и ответ, без токена.

Для учителя

В классе достаточно одной разрешённой учительской учётной записи и одного демонстрационного запуска. Пары готовят ответы и прогнозы, а учитель по очереди показывает два варианта, перезапуская программу. При самостоятельной работе используйте разные токены на запущенных компьютерах: два процесса с одним токеном мешают друг другу получать события. При работе в кабинете используйте заранее подготовленное окружение, чтобы не тратить этап на скачивание. Если сеть недоступна, сопоставьте код с контрольным журналом из следующего этапа, а сетевой запуск перенесите до восстановления доступа.

07. Этап 5. Приёмка и поиск сбоя 7 мин

ГОВОРИМ УЧЕНИКАМ

Бот может молчать по разным причинам. Полезно проверить, дошла ли программа хотя бы до запуска, смогла ли она обратиться к Telegram и какой командой её проверили. Тогда одну ошибку можно отделить от другой.

Проверка. Три наблюдения

Контрольный журнал: `/start` → ответ, заданный в `greet`. Обычное «привет» → в этом варианте обработчик не вызывается.

Сопоставьте каждое наблюдение с первой проверкой, которую стоит сделать. У одной пары терминал закрыл работу с `KeyError: TELEGRAM_BOT_TOKEN`. Другая видит отказ в доступе со стороны Telegram. У третьей `/start` работает, а «привет» ответа не вызывает. Для одного случая предложите, как проверить исправление.

Для учителя

При `KeyError` переменная среды не задана в этом терминале. При отказе проверьте действительность токена, не выводя его в общий чат. Если работает `/start`, соединение установлено, но обычный текст не подходит обработчику `CommandStart()`. Остановить программу можно сочетанием Ctrl+C.

08. Этап 6. Главное и проверка понимания 4 мин

ГОВОРИМ УЧЕНИКАМ

Запуск бота легче проверять по участкам. Сначала запускается Python, затем программа связывается с Telegram, после этого команда попадает в обработчик.

Главное за урок

- Чат в Telegram и программа, которая отвечает, работают отдельно.
- Токен подтверждает право программы действовать от имени бота.
- Точка входа запускает приём событий, а обработчик определяет ответ на команду.

Итоговые вопросы

1. Почему бот не отвечает после остановки Python, хотя чат открывается?
2. Что сделать, если секретная строка уже попала в общий чат?

Для учителя

1. Telegram сохраняет чат, но обработчик больше не выполняется. 2. Отозвать токен через BotFather и задать новый в переменной среды.

Источники

- Telegram. Регистрация бота через BotFather · Проверено 05.11.2026. BotFather выдаёт токен после команды /newbot. Токен подтверждает право работать с Bot API от имени бота.
- aiogram 3. Получение событий через long polling · Проверено 05.11.2026. Документация показывает получение событий с помощью Dispatcher.start_polling.