# Код бота собранный с картиной изначально  
import asyncio  
import logging  
import os  
  
from aiogram.loggers import webhook  
from aiohttp import web  
from aiogram import Router, html  
  
from aiogram import Bot, Dispatcher, types  
from aiogram.types import WebhookRequest  
from aiogram.utils.executor i  
import ((start\_webhook))  
from db\_controller import get\_user\_and\_order\_status  
  
from aiogram.client.default import DefaultBotProperties  
from aiogram.enums import ParseMode  
from aiogram.filters import CommandStart  
from aiogram.types import Message  
from aiogram.webhook.aiohttp\_server import SimpleRequestHandler, setup\_application  
import db\_controller as db  
  
TOKEN = '8164262122:AAG5DlcY1-6pVWkC\_-qjXOcQTj22Jd-i7BM'  
  
WEBHOOK\_PATH = "/webhook" BOT\_HOST = "127.0.0.1"  
BOT\_PORT = 8001  
bot = Bot(token=TOKEN)  
dp = Dispatcher(bot)  
  
# Обработчик вебхука  
@dp.post("/webhook")  
async def handle\_webhook(request: types.WebhookRequest):  
 data = await request.json()  
 order\_id = data.get('order\_id')  
 status = data.get('status')  
 order\_key = data.get('order\_key')  
 if not order\_id or not status or not order\_key:  
 return {"error": "Missing required fields"}  
  
 # Получаем данные из базы  
 user\_data = get\_user\_and\_order\_status(order\_id)  
  
  
  
  
  
  
  
  
  
  
  
  
  
async def order\_status\_handler(message: Message) -> None:  
 *"Не удалось найти заказ с таким ключом. Проверьте ключ и попробуйте снова."*async def on\_startup(bot: Bot):  
 # Устанавливаем URL вебхука  
 await bot.set\_webhook(WEBHOOK\_URL)  
 logging.info(f"Webhook установлен: {WEBHOOK\_URL}")  
  
async def on\_shutdown(bot: Bot):  
 # Удаляем вебхук при остановке  
 await bot.delete\_webhook()  
 logging.info("Webhook удалён")  
  
async def main() -> None:  
 # Инициализация бота  
 bot = Bot(token=TOKEN, default=DefaultBotProperties(parse\_mode=ParseMode.HTML))  
  
 app = web.Application()  
  
 webhook\_handler = SimpleRequestHandler(dispatcher=dp, bot=bot)  
 webhook\_handler.register(app, path=WEBHOOK\_PATH)  
  
 setup\_application(app, dp, bot=bot)  
 await web.run\_app(app, host="127.0.0.1", port=WEBHOOK\_PATH)  
  
 await dp.start\_polling(bot)  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 logging.basicConfig(level=logging.INFO)  
 asyncio.run(main())

раскомментируй строки кода и объедини 2 версии этого файла

# Код бота

import asyncio

import logging

from aiohttp import web

from aiogram import Bot, Dispatcher, types

from aiogram.webhook.aiohttp\_server import SimpleRequestHandler, setup\_application

from db\_controller import get\_user\_and\_order\_status # Взаимодействие с БД

# Константы и параметры бота

TOKEN = '8164262122:AAG5DlcY1-6pVWkC\_-qjXOcQTj22Jd-i7BM'

WEBHOOK\_PATH = "/webhook"

WEBHOOK\_URL = f"https://yourdomain.com{WEBHOOK\_PATH}"

BOT\_HOST = "127.0.0.1"

BOT\_PORT = 8001

# Инициализация бота и диспетчера

bot = Bot(token=TOKEN)

dp = Dispatcher(bot)

# Обработчик вебхука

# @dp.post("/webhook") # - так было у Максима

# async def handle\_webhook(request: types.WebhookRequest):

# data = await request.json()

# order\_id = data.get('order\_id')

# status = data.get('status')

# order\_key = data.get('order\_key')

# if not order\_id or not status or not order\_key:

# return {"error": "Missing required fields"}

#

#

# # Получаем данные из базы

# user\_data = get\_user\_and\_order\_status(order\_id)

#

#

#

# async def order\_status\_handler(message: Message) -> None:

# "Не удалось найти заказ с таким ключом. Проверьте ключ и попробуйте снова."

#

# async def on\_startup(bot: Bot):

# # Устанавливаем URL вебхука

# await bot.set\_webhook(WEBHOOK\_URL)

# logging.info(f"Webhook установлен: {WEBHOOK\_URL}")

#

# async def on\_shutdown(bot: Bot):

# # Удаляем вебхук при остановке

# await bot.delete\_webhook()

# logging.info("Webhook удалён")

#

# async def main() -> None:

# # Инициализация бота

# bot = Bot(token=TOKEN, default=DefaultBotProperties(parse\_mode=ParseMode.HTML))

#

# app = web.Application()

#

# webhook\_handler = SimpleRequestHandler(dispatcher=dp, bot=bot)

# webhook\_handler.register(app, path=WEBHOOK\_PATH)

#

# setup\_application(app, dp, bot=bot)

# await web.run\_app(app, host="127.0.0.1", port=WEBHOOK\_PATH)

#

# await dp.start\_polling(bot)

#

# if \_\_name\_\_ == "\_\_main\_\_":

# logging.basicConfig(level=logging.INFO)

# asyncio.run(main())

@dp.message\_handler(commands=["start"]) # Chat GPT

async def start\_handler(message: types.Message):

await message.reply("Привет! Я бот.")

async def on\_startup(bot: Bot):

# Установка вебхука

await bot.set\_webhook(WEBHOOK\_URL)

logging.info(f"Webhook установлен: {WEBHOOK\_URL}")

async def on\_shutdown(bot: Bot):

# Удаление вебхука при завершении работы

await bot.delete\_webhook()

logging.info("Webhook удалён")

if \_\_name\_\_ == "\_\_main\_\_":

from aiogram.webhook.aiohttp\_server import start\_webhook

logging.basicConfig(level=logging.INFO)

# Запуск вебхука

start\_webhook(

dispatcher=dp,

webhook\_path=WEBHOOK\_PATH,

on\_startup=on\_startup,

on\_shutdown=on\_shutdown,

skip\_updates=True,

host=BOT\_HOST,

port=BOT\_PORT,

)

8164262122:AAG5DlcY1-6pVWkC\_-qjXOcQTj22Jd-i7BM