keyboards.py

from aiogram.types import ReplyKeyboardMarkup, KeyboardButton, InlineKeyboardMarkup, InlineKeyboardButton

from aiogram.utils.keyboard import ReplyKeyboardBuilder, InlineKeyboardBuilder

main = ReplyKeyboardMarkup(keyboard=[

[KeyboardButton(text="Тестовая кнопка 1")],

[KeyboardButton(text="Тестовая кнопка 2"), KeyboardButton(text="Тестовая кнопка 3")]

], resize\_keyboard=True)

inline\_keyboard\_test = InlineKeyboardMarkup(inline\_keyboard=[

[InlineKeyboardButton(text="Каталог", callback\_data='catalog')],

[InlineKeyboardButton(text="Новости", callback\_data='news')],

[InlineKeyboardButton(text="Профиль", callback\_data='person')]

])

test = ["кнопка 1", "кнопка 2", "кнопка 3", "кнопка 4"]

async def test\_keyboard():

keyboard = InlineKeyboardBuilder()

for key in test:

keyboard.add(InlineKeyboardButton(text=key, url='https://www.youtube.com/watch?v=HfaIcB4Ogxk'))

return keyboard.adjust(2).as\_markup()

main.py

import asyncio

from aiogram import Bot, Dispatcher, F

from aiogram.filters import CommandStart, Command

from aiogram.types import Message, FSInputFile, CallbackQuery

import random

from gtts import gTTS

import os

from config import TOKEN

import keyboards as kb

bot = Bot(token=TOKEN)

dp = Dispatcher()

@dp.callback\_query(F.data == 'news')

async def news(callback: CallbackQuery):

await callback.answer("Новости подгружаются", show\_alert=True)

await callback.message.edit\_text('Вот свежие новости!', reply\_markup=await kb.test\_keyboard())

@dp.message(F.text == "Тестовая кнопка 1")

async def test\_button(message: Message):

await message.answer('Обработка нажатия на reply кнопку')

@dp.message(Command('help'))

async def help(message: Message):

await message.answer('Этот бот умеет выполнять команды: \n /start \n /help \n /minitraining')

@dp.message(CommandStart())

async def start(message: Message):

await message.answer(f'Приветики, {message.from\_user.first\_name}', reply\_markup=kb.inline\_keyboard\_test)

async def main():

await dp.start\_polling(bot)

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

asyncio.run(main())