# Текст программы

## bot.py

import asyncio  
  
from aiogram import Bot, Dispatcher  
from core.config import load\_config  
from core.router import router  
  
  
async def main():  
 config = load\_config()  
  
 bot = Bot(config.telegram.token, parse\_mode="HTML")  
 dp = Dispatcher()  
  
 dp.include\_router(router)  
  
 await dp.start\_polling(bot)  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 asyncio.run(main())  
import asyncio  
  
from aiogram import Bot, Dispatcher  
from core.config import load\_config  
from core.router import router  
  
  
async def main():  
 config = load\_config()  
  
 bot = Bot(config.telegram.token, parse\_mode="HTML")  
 dp = Dispatcher()  
  
 dp.include\_router(router)  
  
 await dp.start\_polling(bot)  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 asyncio.run(main())

## config.py

from dataclasses import dataclass  
from decouple import config  
import os  
  
  
@dataclass  
class Telegram:  
 token: str  
  
  
@dataclass  
class Config:  
 telegram: Telegram  
  
  
def load\_config():  
 return Config(  
 telegram=Telegram(  
 token=config("BOT\_TOKEN", cast=str),  
 ),  
 )  
from dataclasses import dataclass  
from decouple import config  
import os  
  
  
@dataclass  
class Telegram:  
 token: str  
  
  
@dataclass  
class Config:  
 telegram: Telegram  
  
  
def load\_config():  
 return Config(  
 telegram=Telegram(  
 token=config("BOT\_TOKEN", cast=str),  
 ),  
 )

## router.py

from aiogram.types import Message  
from aiogram import Router  
from aiogram import F  
from aiogram.filters.command import Command, CommandStart  
from aiogram.types import ReplyKeyboardMarkup, KeyboardButton  
from .cat\_fact\_service import CatFactService  
  
router = Router()  
catFactService = CatFactService()  
  
main\_kb = ReplyKeyboardMarkup(  
 keyboard=[[KeyboardButton(text="Give me a fact"), KeyboardButton(text="💥")]],  
 resize\_keyboard=True,  
 input\_field\_placeholder="Choose an option from the menu",  
)  
  
  
@router.message(CommandStart())  
async def start(message: Message):  
 await message.answer(  
 "Hewwo! pwease choose an option on your Keyboawd!!",  
 reply\_markup=main\_kb,  
 )  
  
  
@router.message(F.text == "Give me a fact")  
async def get\_cat\_fact(message: Message):  
 await message.answer(catFactService.get\_cat\_fact())  
  
  
@router.message(F.text == "💥")  
async def get\_cat\_sticker(message: Message):  
 await message.answer\_sticker(  
 "CAACAgIAAxkBAAELDqlljYOpmoIoaN7w36gBuOInF8EWIgACGhkAAje1KEj8SIqr6ZeTOTME"  
 )  
from aiogram.types import Message  
from aiogram import Router  
from aiogram import F  
from aiogram.filters.command import Command, CommandStart  
from aiogram.types import ReplyKeyboardMarkup, KeyboardButton  
from .cat\_fact\_service import CatFactService  
  
router = Router()  
catFactService = CatFactService()  
  
main\_kb = ReplyKeyboardMarkup(  
 keyboard=[[KeyboardButton(text="Give me a fact"), KeyboardButton(text="💥")]],  
 resize\_keyboard=True,  
 input\_field\_placeholder="Choose an option from the menu",  
)  
  
  
@router.message(CommandStart())  
async def start(message: Message):  
 await message.answer(  
 "Hewwo! pwease choose an option on your Keyboawd!!",  
 reply\_markup=main\_kb,  
 )  
  
  
@router.message(F.text == "Give me a fact")  
async def get\_cat\_fact(message: Message):  
 await message.answer(catFactService.get\_cat\_fact())  
  
  
@router.message(F.text == "💥")  
async def get\_cat\_sticker(message: Message):  
 await message.answer\_sticker(  
 "CAACAgIAAxkBAAELDqlljYOpmoIoaN7w36gBuOInF8EWIgACGhkAAje1KEj8SIqr6ZeTOTME"  
 )

## cat\_fact\_service.py

import requests  
  
  
class CatFactService:  
 CAT\_API\_URL = "https://catfact.ninja/fact"  
  
 @staticmethod  
 def get\_cat\_fact():  
 try:  
 response = requests.get(CatFactService.CAT\_API\_URL)  
 response.raise\_for\_status() # Raise an HTTPError for bad responses  
 cat\_fact\_data = response.json()  
 return cat\_fact\_data.get("fact")  
 except requests.RequestException as e:  
 print(f"Error fetching cat fact: {e}")  
 return None  
import requests  
  
  
class CatFactService:  
 CAT\_API\_URL = "https://catfact.ninja/fact"  
  
 @staticmethod  
 def get\_cat\_fact():  
 try:  
 response = requests.get(CatFactService.CAT\_API\_URL)  
 response.raise\_for\_status() # Raise an HTTPError for bad responses  
 cat\_fact\_data = response.json()  
 return cat\_fact\_data.get("fact")  
 except requests.RequestException as e:  
 print(f"Error fetching cat fact: {e}")  
 return None

# Результат выполнения программы

