

Startup House Telegram Bot

Telegram-бот, предоставляющий информацию о стартапах, бизнесе и искусственном интеллекте. Бот использует API OpenAI для генерации ответов и имеет систему резервных ответов при недоступности API.

Функциональность

Общие команды

- `/start` - Запуск бота и приветственное сообщение
- `/help` - Справка по использованию бота

Административные команды

- `/stats` - Статистика использования (количество пользователей, сообщений, топ активных пользователей)
- `/reset_stats` - Сброс статистики (не реализовано в текущей версии)

Обработка сообщений

- Ответы на приветствия
- Распознавание ключевых слов по темам ИИ и бизнеса
- Генерация ответов с использованием OpenAI API
- Fallback-ответы при недоступности API

Интерфейс

- Основная клавиатура с кнопками по темам (определена, но не используется в текущей версии)
- Инлайн-клавиатура для выбора конкретных тем (определена, но не используется в текущей версии)

Примечание: Хотя в проекте определены клавиатуры в файлах `keyboards/reply.py` и `keyboards/inline.py`, они не интегрированы с обработчиками сообщений в текущей версии кода.

Технический стек

- Python 3.12
- aiogram 3.x (фреймворк для Telegram ботов)
- PostgreSQL (хранение логов и статистики)
- OpenAI API (генерация ответов)
- Docker (контейнеризация)

Структура проекта

```
startup-house-bot/
├── bot/ ..... # Основной модуль бота
│   ├── __init__.py ..... # Инициализация бота и диспетчера
├── config/ ..... # Конфигурация
│   ├── __init__.py ..... # Загрузка и структурирование конфигурации
├── filters/ ..... # Фильтры для обработчиков
│   ├── __init__.py
│   └── custom_filters.py # Фильтр IsAdmin и другие
├── handlers/ ..... # Обработчики сообщений
│   ├── __init__.py ..... # Регистрация всех обработчиков
│   ├── admin.py ..... # Обработчики команд администратора
│   ├── common.py ..... # Общие обработчики команд
│   └── user.py ..... # Обработчики пользовательских сообщений
├── keyboards/ ..... # Клавиатуры
│   ├── __init__.py
│   ├── inline.py ..... # Инлайн-клавиатуры
│   └── reply.py ..... # Reply-клавиатуры
├── middlewares/ ..... # Middleware для обработки сообщений
│   ├── __init__.py ..... # Настройка middleware
│   ├── logger.py ..... # Логирование в БД и файлы
│   └── throttling.py ..... # Защита от спама
├── services/ ..... # Сервисы для работы с внешними API и БД
│   ├── __init__.py
│   ├── openai_service.py # Взаимодействие с OpenAI API
│   └── stats_service.py # Получение статистики из БД
├── utils/ ..... # Утилиты
│   ├── __init__.py
│   ├── db_utils.py ..... # Утилиты для работы с БД
│   └── text_utils.py ..... # Обработка текста
├── .env.example ..... # Пример .env файла
├── .gitignore ..... # Файлы, исключаемые из Git
├── Dockerfile ..... # Конфигурация Docker
├── main.py ..... # Точка входа в приложение
├── requirements.txt ..... # Зависимости Python
├── scripts/ ..... # Скрипты для обслуживания
│   └── init_db.py ..... # Инициализация БД
└── logs/ ..... # Директория для логов (создается автоматически)
```

Установка и запуск

Предварительные требования

- Python 3.12+
- PostgreSQL

- Токен Telegram бота от @BotFather
- API ключ OpenAI (опционально)

Установка через Docker

1. Клонировать репозиторий:

```
git clone https://github.com/tsabyrovvv/bot_example.git
cd bot_example
```

2. Создайте файл `.env` на основе `.env.example`:

```
cp .env.example .env
```

3. Отредактируйте `.env`, указав необходимые параметры:

```
BOT_TOKEN=your_telegram_bot_token
ADMIN_IDS=123456789,987654321
DB_HOST=postgres
DB_PORT=5432
DB_NAME=startup_house_bot
DB_USER=postgres
DB_PASSWORD=your_password
OPENAI_API_KEY=your_openai_api_key
OPENAI_MODEL=gpt-3.5-turbo
```

4. Соберите и запустите контейнеры:

```
docker-compose up -d
```

Ручная установка

1. Клонировать репозиторий:

```
git clone https://github.com/username/startup-house-bot.git
cd startup-house-bot
```

2. Создайте виртуальное окружение и установите зависимости:

```
python -m venv venv
source venv/bin/activate # На Windows: venv\Scripts\activate
pip install -r requirements.txt
```

3. Создайте файл `.env` и настройте параметры (как в шаге 3 для Docker).

4. Инициализируйте базу данных:

```
python scripts/init_db.py
```

5. Запустите бота:

```
python main.py
```

Логирование и мониторинг

Бот имеет встроенную систему логирования:

- Логи сохраняются в директории `logs/` с ежедневной ротацией
- События и ошибки записываются в базу данных PostgreSQL
- Для аналитики используйте команду `/stats` (доступна только администраторам)

Настройка базы данных

База данных автоматически инициализируется при первом запуске. Скрипт создает:

- Таблицу `bot_logs` для сообщений и действий пользователей
- Таблицу `bot_errors` для логирования ошибок
- Индексы для оптимизации запросов

Дополнительная информация

Защита от спама

Бот использует механизм throttling, ограничивающий частоту сообщений от одного пользователя до 1 сообщения в секунду.

Fallback-ответы

При недоступности OpenAI API бот использует предустановленные ответы по популярным темам.

Администраторы

Для добавления администратора укажите его Telegram ID в переменной `ADMIN_IDS` (через запятую, если несколько).

Возможные улучшения

- Хранение контекста диалога для более персонализированного общения
- Расширение базы знаний для автономной работы
- Интеграция с внешними API для получения актуальной информации
- Добавление многоязычности
- Система модерации контента
- Веб-интерфейс для администраторов

Лицензия

MIT