# Telegram-бот «Изливание души»

Асинхронный Telegram-бот для сервиса психологической поддержки. Позволяет пользователям выговориться голосом и получить поддерживающий голосовой ответ от ИИ (GPT-4).

## Возможности

* 🎙 **Голосовое взаимодействие**: пользователи отправляют голосовые сообщения, получают голосовые ответы
* 🤖 **ИИ-поддержка**: использует GPT-4 для генерации эмпатичных ответов
* ⏱ **Контроль времени**: сессии ограничены 30 минутами, отдельные сообщения — 7 минутами
* 👥 **Административный контроль**: администраторы получают уведомления и могут управлять промптами
* 🔧 **Гибкие настройки**: системный промпт можно изменять в реальном времени

## Технологии

* **Python 3.10+**
* **python-telegram-bot v20.x** (асинхронная версия)
* **OpenAI API**: GPT-4, Whisper (STT), TTS (модель onyx)
* **pydub + ffmpeg** для обработки аудио
* **python-dotenv** для конфигурации

## Установка

1. **Клонируйте репозиторий:**

* git clone <repository-url>  
  cd GPTnitik

1. **Создайте и активируйте виртуальное окружение:**

* # Создание виртуального окружения  
  python -m venv venv  
    
  # Активация окружения  
  # Linux/macOS:  
  source venv/bin/activate  
    
  # Windows:  
  venv\Scripts\activate
* **Важно:** Всегда активируйте виртуальное окружение перед работой с проектом!

1. **Установите зависимости:**

* pip install -r requirements.txt

1. **Установите ffmpeg** (для обработки аудио):

* # Ubuntu/Debian  
  sudo apt update && sudo apt install ffmpeg  
    
  # macOS  
  brew install ffmpeg  
    
  # Windows  
  # Скачайте с https://ffmpeg.org/download.html

1. **Настройте конфигурацию:** Создайте файл .env в корне проекта:

* TELEGRAM\_TOKEN=your\_telegram\_bot\_token\_here  
  OPENAI\_API\_KEY=your\_openai\_api\_key\_here  
  ADMIN\_ID\_1=123456789  
  ADMIN\_ID\_2=987654321  
  DEBUG\_SEND\_VOICE=false

1. **Запустите бота:**

* # Убедитесь, что виртуальное окружение активировано  
  python bot.py

### Работа с виртуальным окружением

* **Активация окружения** (каждый раз при работе с проектом):
* # Linux/macOS:  
  source venv/bin/activate  
    
  # Windows:  
  venv\Scripts\activate
* **Деактивация окружения**:
* deactivate
* **Проверка активного окружения**:
* which python # Linux/macOS  
  where python # Windows
* Должен показать путь к Python внутри папки venv/

## Структура проекта

/project  
├── bot.py # основной бот, FSM  
├── stt.py # speech\_to\_text(file\_path)  
├── tts.py # text\_to\_speech(text, output\_path)  
├── gpt.py # get\_gpt\_response(text)  
├── admin.py # /prompt, /setprompt, /resetprompt  
├── utils.py # temp files, timer, send\_to\_admins  
├── config.py # load config, is\_admin(), read\_prompt()  
├── data/  
│ └── prompt.txt # текущий системный промпт  
├── temp/ # временные файлы (создаётся автоматически)  
├── requirements.txt # зависимости Python  
├── .env # конфигурация (создайте сами)  
└── README.md # этот файл

## Использование

### Для пользователей

1. Начните диалог командой /start
2. Введите имя или пропустите этот шаг
3. Отправьте голосовое сообщение (до 7 минут)
4. Получите голосовой ответ от бота
5. Продолжайте диалог или завершите сессию

### Для администраторов

Доступные команды: - /prompt — показать текущий системный промпт - /setprompt <новый промпт> — установить новый промпт - /resetprompt — сбросить промпт к значению по умолчанию - /stats — показать статистику бота - /cleanup — очистить старые временные файлы

## Настройки

### Переменные окружения (.env)

* TELEGRAM\_TOKEN — токен Telegram-бота
* OPENAI\_API\_KEY — ключ OpenAI API
* ADMIN\_ID\_1, ADMIN\_ID\_2 — ID администраторов
* DEBUG\_SEND\_VOICE — пересылать ли голосовые сообщения админам (true/false)

### Режимы отладки

**По умолчанию (DEBUG\_SEND\_VOICE=false):** - ❌ Голосовые сообщения пользователей НЕ пересылаются - ❌ Голосовые ответы бота НЕ пересылаются - ✅ Текст после STT пересылается: [STT] {текст} - ✅ Ответ GPT пересылается: [GPT] {ответ}

**В режиме отладки (DEBUG\_SEND\_VOICE=true):** - ✅ Все голосовые сообщения пересылаются администраторам - ✅ Все текстовые уведомления также отправляются

## Ограничения

* **Длительность голосового сообщения**: максимум 7 минут
* **Длительность сессии**: максимум 30 минут
* **Размер аудиофайла**: максимум 25MB (ограничение OpenAI)
* **Длина текста для TTS**: максимум 4096 символов

## Логирование

Бот ведет логи в файле session.log:

2025-04-05 15:30 | Иван | Сессия: 28 мин | Обменов: 4

## Безопасность

* Проверка прав администратора для всех админ-команд
* Валидация пользовательского ввода
* Обработка всех исключений с логированием
* Автоматическая очистка временных файлов
* Ограничения на размер и длительность файлов

## Устранение неполадок

### Ошибки аудио

* Убедитесь, что установлен ffmpeg
* Проверьте качество записи (говорите четко и громко)
* Проверьте длительность сообщения (не более 7 минут)

### Ошибки API

* Проверьте правильность токенов в .env
* Убедитесь, что у вас есть кредиты на OpenAI
* Проверьте лимиты запросов

### Ошибки бота

* Проверьте логи в session.log
* Убедитесь, что бот имеет права на отправку сообщений
* Проверьте ID администраторов в .env

## Лицензия

Этот проект создан для образовательных целей.

# 📁 Структура проекта “Изливание души”

## 🗂️ Основные файлы

### Исполняемые модули:

* **bot.py** - Основной модуль бота с логикой FSM и обработчиками
* **config.py** - Конфигурация, настройки и работа с промптом
* **admin.py** - Административные команды для управления ботом
* **gpt.py** - Модуль для работы с GPT-4 API
* **stt.py** - Модуль преобразования речи в текст (Speech-to-Text)
* **tts.py** - Модуль преобразования текста в речь (Text-to-Speech)
* **utils.py** - Вспомогательные функции и утилиты

### Конфигурация:

* **.env** - Переменные окружения (токены, ключи API, ID администраторов)
* **requirements.txt** - Зависимости Python
* **.gitignore** - Исключения для Git

### Документация:

* **README.md** - Основная документация проекта
* **ADMIN\_GUIDE.md** - Руководство для администраторов
* **PROJECT\_STRUCTURE.md** - Этот файл с описанием структуры

## 📂 Директории

### data/

* **prompt.txt** - Текущий системный промпт для GPT-4
* Создается автоматически при первом запуске

### temp/

* Временные аудиофайлы (.mp3, .ogg, .wav)
* Автоматически очищается от старых файлов
* Не включается в Git

### venv/

* Виртуальное окружение Python
* Не включается в Git

## 🚀 Запуск проекта

1. **Установка зависимостей:**

* pip install -r requirements.txt

1. **Настройка .env файла:**

* TELEGRAM\_TOKEN=your\_bot\_token  
  OPENAI\_API\_KEY=your\_openai\_key  
  ADMIN\_ID\_1=your\_telegram\_id  
  ADMIN\_ID\_2=second\_admin\_id  
  DEBUG\_SEND\_VOICE=false

1. **Запуск бота:**

* python bot.py

## 🔧 Архитектура

### Поток обработки сообщений:

1. **Пользователь** отправляет голосовое сообщение
2. **STT модуль** преобразует речь в текст
3. **GPT модуль** генерирует ответ на основе промпта
4. **TTS модуль** озвучивает ответ
5. **Бот** отправляет голосовой ответ пользователю

### Состояния FSM:

* AWAIT\_NAME - Ожидание ввода имени
* MAIN\_MENU - Главное меню (не используется после обновлений)
* RECORDING - Режим записи и обработки голосовых сообщений

### Административные функции:

* Изменение системного промпта
* Мониторинг статистики
* Очистка временных файлов
* Просмотр текущих настроек

## 📝 Логирование

* **session.log** - Подробные логи работы бота
* Уровень логирования: DEBUG
* Ротация логов не настроена (требует ручной очистки)

## 🔒 Безопасность

* Административные команды доступны только указанным пользователям
* Временные файлы автоматически удаляются
* Чувствительные данные (.env) исключены из Git
* Логи могут содержать пользовательские данные

## 📊 Мониторинг

* Проверка логов на ошибки
* Мониторинг размера temp/ директории
* Отслеживание использования API квот
* Статистика через команду /stats

# 🔧 Руководство администратора бота “Изливание души”

## 📋 Обзор

Этот бот предназначен для психологической поддержки пользователей через голосовое общение. Пользователи отправляют голосовые сообщения, которые обрабатываются через STT (Speech-to-Text), анализируются GPT-4, и ответ озвучивается через TTS (Text-to-Speech).

## 🔑 Административные команды

### /prompt

**Описание:** Показывает текущий системный промпт GPT-4  
**Использование:** /prompt  
**Пример ответа:**

📝 Текущий системный промпт:  
  
Ты — поддерживающий и доброжелательный собеседник...

### /setprompt <новый\_промпт>

**Описание:** Устанавливает новый системный промпт для GPT-4  
**Использование:** /setprompt Новый текст промпта здесь...  
**Ограничения:** - Минимум 10 символов - Максимум 2000 символов - Изменения применяются мгновенно для всех новых запросов

**Пример:**

/setprompt Ты психолог-консультант. Отвечай с эмпатией, задавай уточняющие вопросы. Ответы должны быть краткими и поддерживающими.

### /resetprompt

**Описание:** Сбрасывает промпт к значению по умолчанию  
**Использование:** /resetprompt  
**Промпт по умолчанию:** > “Ты — поддерживающий и доброжелательный собеседник. Пользователь хочет выговориться. Отвечай тёплым, спокойным тоном, с эмпатией. Не давай советов, если не просят. Отвечай коротко — 2–4 предложения.”

### /stats

**Описание:** Показывает статистику работы бота  
**Использование:** /stats  
**Информация включает:** - Настройки отладки - Количество администраторов - Статистику временных файлов - Длину текущего промпта

### /cleanup

**Описание:** Очищает старые временные файлы (старше 1 часа)  
**Использование:** /cleanup  
**Назначение:** Освобождает дисковое пространство от накопившихся аудиофайлов

## ⚙️ Настройки бота

### Переменные окружения (.env файл):

TELEGRAM\_TOKEN=your\_bot\_token\_here  
OPENAI\_API\_KEY=your\_openai\_api\_key\_here  
ADMIN\_ID\_1=your\_telegram\_user\_id  
ADMIN\_ID\_2=second\_admin\_telegram\_user\_id  
DEBUG\_SEND\_VOICE=false  
  
# Лимиты пользователей  
MAX\_MESSAGES\_PER\_SESSION=10  
SESSION\_DURATION\_MINUTES=30

### Лимиты системы:

* **Голосовое сообщение:** максимум 7 минут
* **Сессия пользователя:** настраивается через SESSION\_DURATION\_MINUTES (по умолчанию 30 минут)
* **Количество сообщений в сессии:** настраивается через MAX\_MESSAGES\_PER\_SESSION (по умолчанию 10)
* **Промпт:** 10-2000 символов
* **GPT ответ:** максимум 500 токенов

### Настройка лимитов:

* MAX\_MESSAGES\_PER\_SESSION - максимальное количество голосовых сообщений в одной сессии
* SESSION\_DURATION\_MINUTES - максимальная продолжительность сессии в минутах
* При достижении любого из лимитов сессия автоматически завершается

## 🎯 Рекомендации по настройке промпта

### Хорошие практики:

1. **Тон:** Указывайте желаемый тон (тёплый, поддерживающий, профессиональный)
2. **Длина ответов:** Ограничивайте длину (например, “2-4 предложения”)
3. **Роль:** Четко определяйте роль бота (психолог, друг, консультант)
4. **Границы:** Указывайте, что не должен делать бот

### Примеры промптов:

**Для поддержки:**

Ты — внимательный слушатель и друг. Пользователь делится переживаниями. Отвечай с пониманием и сочувствием. Не давай советов, если не просят. Задавай открытые вопросы, чтобы человек мог больше рассказать. Ответы 2-3 предложения.

**Для консультирования:**

Ты — психолог-консультант. Помогай пользователю разобраться в чувствах и ситуациях. Используй техники активного слушания. Задавай уточняющие вопросы. Предлагай мягкие инсайты. Избегай прямых советов. Ответы краткие и профессиональные.

**Для кризисных ситуаций:**

Ты — кризисный консультант. Пользователь может находиться в сложной ситуации. Отвечай спокойно и поддерживающе. Фокусируйся на безопасности и стабилизации. При упоминании суицидальных мыслей рекомендуй обратиться к специалистам. Ответы короткие и четкие.

## 🚨 Мониторинг и безопасность

### Что отслеживать:

* Логи ошибок в session.log
* Размер папки temp/ (временные файлы)
* Использование API квот OpenAI
* Необычная активность пользователей

### Признаки проблем:

* Частые ошибки TTS/STT
* Превышение лимитов API
* Большое количество временных файлов
* Жалобы пользователей на неподходящие ответы

## 📞 Поддержка пользователей

### Типичные проблемы:

1. **“Бот не понимает речь”** → Проверить качество записи, фоновый шум
2. **“Нет голосового ответа”** → Проверить TTS, квоты API
3. **“Неподходящие ответы”** → Настроить промпт
4. **“Сессия завершилась”** → Объяснить 30-минутный лимит

### Рекомендации пользователям:

* Говорить четко и без фонового шума
* Записывать сообщения до 7 минут
* Использовать наушники для лучшего качества
* Обращаться в тихом месте

## 🔄 Регулярное обслуживание

### Ежедневно:

* Проверить логи на ошибки
* Запустить /cleanup при необходимости

### Еженедельно:

* Проверить /stats для мониторинга использования
* Очистить старые логи при необходимости

### По мере необходимости:

* Обновить промпт на основе обратной связи
* Проверить квоты OpenAI API
* Обновить список администраторов

**Важно:** Все административные команды доступны только пользователям, указанным в ADMIN\_ID\_1 и ADMIN\_ID\_2 в файле .env. Для получения своего Telegram ID можно использовать бота @userinfobot.

# 🚫 Руководство по управлению лимитами пользователей

## Обзор

Система управления лимитами пользователей автоматически отслеживает использование бота и блокирует пользователей при превышении установленных лимитов. Заблокированные пользователи сохраняются в CSV файле для административного контроля.

## Лимиты пользователей

### Автоматические лимиты

* **Максимум сообщений**: 7 запросов за сессию (настраивается в MAX\_MESSAGES\_PER\_SESSION)
* **Максимальная длительность**: 30 минут за сессию (настраивается в SESSION\_DURATION\_MINUTES)

### Условия блокировки

Пользователь автоматически блокируется при превышении любого из лимитов: 1. Отправил 7 или более голосовых сообщений в одной сессии 2. Сессия длилась 30 или более минут

## CSV файл заблокированных пользователей

### Расположение

data/blocked\_users.csv

### Структура файла

| Поле | Описание |
| --- | --- |
| user\_id | ID пользователя в Telegram |
| username | Имя пользователя в Telegram (@username) |
| first\_name | Имя пользователя |
| blocked\_at | Дата и время блокировки (ISO формат) |
| reason | Причина блокировки |
| message\_count | Количество сообщений в сессии |
| session\_duration | Длительность сессии в минутах |

### Пример записи

user\_id,username,first\_name,blocked\_at,reason,message\_count,session\_duration  
123456789,john\_doe,John,2024-01-15T14:30:00.123456,Превышен лимит сообщений (7),7,25  
987654321,jane\_smith,Jane,2024-01-15T15:45:00.654321,Превышен лимит времени (30 мин),5,30

## Административные команды

### Управление лимитами

#### /limits - Просмотр текущих лимитов

Показывает текущие лимиты пользователей и команды для их изменения.

**Пример вывода:**

⚙️ Текущие лимиты пользователей:  
  
💬 Максимум сообщений за сессию: 7  
⏱️ Максимальная длительность сессии: 30 минут  
  
Для изменения используйте:  
• /setlimits СООБЩЕНИЯ МИНУТЫ  
• /resetlimits - сброс к значениям по умолчанию

#### /setlimits СООБЩЕНИЯ МИНУТЫ - Изменение лимитов

Устанавливает новые лимиты для пользователей.

**Использование:**

/setlimits 5 20  
/setlimits 10 45

**Ограничения:** - Количество сообщений: от 1 до 100 - Длительность сессии: от 1 до 1440 минут (24 часа)

**Результат:**

✅ Лимиты обновлены:  
💬 Максимум сообщений: 5  
⏱️ Длительность сессии: 20 минут

#### /resetlimits - Сброс лимитов

Сбрасывает лимиты к значениям по умолчанию из файла .env.

**Результат:**

✅ Лимиты сброшены к значениям по умолчанию:  
💬 Максимум сообщений: 10  
⏱️ Длительность сессии: 30 минут

### Управление заблокированными пользователями

#### /blocked - Просмотр заблокированных пользователей

Показывает список заблокированных пользователей с подробной информацией.

**Пример вывода:**

🚫 Заблокированные пользователи (2):  
  
• ID: 123456789 (John @john\_doe)  
 📅 15.01.2024 14:30  
 📝 Превышен лимит сообщений (7)  
 💬 7 сообщений, 25 мин  
  
• ID: 987654321 (Jane @jane\_smith)  
 📅 15.01.2024 15:45  
 📝 Превышен лимит времени (30 мин)  
 💬 5 сообщений, 30 мин

### /unblock USER\_ID - Разблокировка пользователя

Удаляет пользователя из списка заблокированных.

**Использование:**

/unblock 123456789

**Результат:**

✅ Пользователь 123456789 разблокирован.

### /block USER\_ID [причина] - Ручная блокировка

Добавляет пользователя в список заблокированных вручную.

**Использование:**

/block 123456789 Нарушение правил  
/block 987654321

**Результат:**

✅ Пользователь 123456789 заблокирован.  
Причина: Нарушение правил

### /cleanup\_blocks [дни] - Очистка старых блокировок

Удаляет блокировки старше указанного количества дней (по умолчанию 30).

**Использование:**

/cleanup\_blocks 30  
/cleanup\_blocks 7

**Результат:**

✅ Удалено 5 старых блокировок (старше 30 дней).

### /stats - Статистика (обновлена)

Команда статистики теперь включает информацию о заблокированных пользователях:

📊 Статистика бота:  
  
🔧 Настройки:  
• DEBUG\_SEND\_VOICE: false  
• Администраторов: 1  
  
⏱️ Лимиты сессий:  
• Максимум сообщений: 7  
• Длительность сессии: 30 минут  
  
🚫 Заблокированные пользователи:  
• Количество: 15  
  
📁 Временные файлы:  
• Количество: 3  
• Размер: 2.45 MB  
  
📝 Промпт:  
• Длина: 156 символов

## Поведение бота для заблокированных пользователей

### При попытке запуска

Когда заблокированный пользователь отправляет команду /start, он получает сообщение:

❌ Ваш период эксплуатации бота истёк.  
  
Вы превысили лимиты использования (7 запросов или 30 минут).  
Для восстановления доступа обратитесь к администратору.

### Логирование

Все попытки доступа заблокированных пользователей логируются:

[START] Пользователь 123456789 заблокирован, отказываем в доступе

## Автоматическое управление

### Проверка при завершении сессии

Система автоматически проверяет лимиты при завершении каждой сессии и блокирует пользователя при необходимости.

### Логирование блокировок

[END\_SESSION] Пользователь 123456789 заблокирован за превышение лимитов

### Создание файлов

* CSV файл создается автоматически при первой блокировке
* Директория data/ создается автоматически если не существует

## Безопасность и надежность

### Обработка ошибок

* Система продолжает работать даже при ошибках в модуле лимитов
* Все ошибки логируются для диагностики
* Некорректные записи в CSV игнорируются

### Целостность данных

* CSV файл перезаписывается при разблокировке для обеспечения целостности
* Поддерживается кодировка UTF-8 для корректного отображения имен
* Автоматическое создание заголовков при создании файла

### Производительность

* Список заблокированных пользователей загружается в память при старте
* Быстрая проверка блокировки через множество (set)
* Минимальное влияние на производительность бота

## Настройка лимитов

Лимиты настраиваются в файле .env:

# Максимальное количество сообщений за сессию  
MAX\_MESSAGES\_PER\_SESSION=7  
  
# Максимальная длительность сессии в минутах  
SESSION\_DURATION\_MINUTES=30

После изменения настроек необходимо перезапустить бота.

## Мониторинг и аналитика

### Рекомендации по мониторингу

1. Регулярно проверяйте список заблокированных пользователей командой /blocked
2. Анализируйте причины блокировок для корректировки лимитов
3. Используйте /cleanup\_blocks для очистки старых записей
4. Мониторьте логи на предмет попыток доступа заблокированных пользователей

### Экспорт данных

CSV файл можно легко импортировать в Excel, Google Sheets или другие инструменты для анализа: - Статистика по времени блокировок - Анализ причин блокировок - Выявление паттернов использования

## Восстановление доступа

Для восстановления доступа пользователя: 1. Найдите его ID в списке заблокированных: /blocked 2. Разблокируйте командой: /unblock USER\_ID 3. Пользователь сможет снова использовать бота

## Резервное копирование

Рекомендуется регулярно создавать резервные копии файла data/blocked\_users.csv для восстановления данных в случае необходимости.

# 🕛 Руководство по планировщику автоматической очистки

## Обзор

Система автоматической очистки заблокированных пользователей работает в фоновом режиме и выполняет полную очистку списка блокировок каждый день в **00:00 часов**.

## Как это работает

### 🔄 Автоматическая очистка в 00:00

1. **Планировщик запускается** вместе с ботом
2. **Вычисляет время** до следующего 00:00
3. **Ждет** до назначенного времени
4. **Выполняет очистку** всех блокировок
5. **Отправляет уведомление** администраторам
6. **Планирует** следующую очистку на завтра

### 📋 Что происходит при очистке

* ✅ Все заблокированные пользователи разблокируются
* 🗑️ CSV файл data/blocked\_users.csv очищается
* 📝 Создается новый пустой файл с заголовками
* 📊 Логируется количество разблокированных пользователей
* 📨 Администраторы получают уведомление

## Административные команды

### /clearblocks - Ручная очистка

/clearblocks

* Выполняет полную очистку всех блокировок немедленно
* Имитирует автоматическую очистку в 00:00
* Доступна только администраторам

**Пример ответа:**

🕛 Выполнена полная очистка блокировок  
  
✅ Разблокировано пользователей: 15  
📅 Дата: 15.12.2024 14:30  
  
Все пользователи снова могут пользоваться ботом.

### Другие команды управления блокировками

* /blocked - показать заблокированных пользователей
* /unblock USER\_ID - разблокировать конкретного пользователя
* /block USER\_ID [причина] - заблокировать пользователя
* /cleanup\_blocks [дни] - удалить старые блокировки

## Уведомления администраторам

### 🕛 Ежедневное уведомление в 00:00

🕛 Ежедневная очистка блокировок (00:00)  
  
✅ Разблокировано пользователей: 23  
📅 Дата: 15.12.2024  
  
Все пользователи снова могут пользоваться ботом.

## Логирование

### Записи в логах

2024-12-15 00:00:01 - scheduler - INFO - 🕛 Выполняется ежедневная очистка заблокированных пользователей (00:00)  
2024-12-15 00:00:01 - user\_limits - INFO - Выполнена полная очистка блокировок: удалено 23 записей  
2024-12-15 00:00:01 - scheduler - INFO - ✅ Ежедневная очистка завершена: разблокировано 23 пользователей

## Технические детails

### Файлы системы

* **scheduler.py** - модуль планировщика задач
* **user\_limits.py** - управление блокировками (добавлен метод clear\_all\_blocks())
* **data/blocked\_users.csv** - файл с заблокированными пользователями

### Интеграция с ботом

* Планировщик запускается в bot.py при старте бота
* Корректно останавливается при завершении работы бота
* Работает асинхронно, не блокирует основную работу бота

### Обработка ошибок

* Все ошибки логируются
* При ошибке планировщик продолжает работу
* Уведомления администраторам отправляются по возможности

## Настройка времени очистки

По умолчанию очистка происходит в **00:00**. Для изменения времени:

1. Отредактируйте файл scheduler.py
2. Измените строку:

* target\_time = time(0, 0) # 00:00
* На желаемое время, например:
* target\_time = time(3, 0) # 03:00

## Мониторинг

### Проверка работы планировщика

1. Запустите бота
2. Проверьте логи на наличие записи:

* Следующая очистка блокировок запланирована на: 2024-12-16 00:00:00

### Тестирование

* Используйте команду /clearblocks для тестирования функциональности
* Проверьте, что уведомления приходят администраторам
* Убедитесь, что файл data/blocked\_users.csv очищается

## Преимущества системы

✅ **Автоматизация** - не требует ручного вмешательства  
✅ **Надежность** - работает даже при перезапуске бота  
✅ **Прозрачность** - все действия логируются  
✅ **Контроль** - администраторы получают уведомления  
✅ **Гибкость** - можно выполнить очистку вручную

## Устранение неполадок

### Планировщик не запускается

* Проверьте логи на ошибки при запуске
* Убедитесь, что модуль scheduler.py импортируется корректно

### Очистка не выполняется

* Проверьте системное время сервера
* Убедитесь, что бот работает непрерывно
* Проверьте права доступа к файлу data/blocked\_users.csv

### Уведомления не приходят

* Проверьте настройки ADMIN\_IDS в .env
* Убедитесь, что у бота есть права отправлять сообщения администраторам

# 🎯 Руководство по управлению токенами GPT

## Обзор

Администраторы могут управлять лимитом токенов для ответов GPT-4. Токены определяют максимальную длину ответа, который может сгенерировать искусственный интеллект.

## Что такое токены?

**Токены** — это единицы измерения текста для GPT. Примерно: - 1 токен ≈ 0.75 слова на английском языке - 1 токен ≈ 0.5-1 слово на русском языке - 100 токенов ≈ 75 английских слов или 50-100 русских слов

## Административные команды

### /tokens - Просмотр настроек

/tokens

Показывает текущий лимит токенов и значение по умолчанию.

**Пример ответа:**

🔧 Настройки токенов GPT:  
  
🎯 Текущий лимит: 800 токенов  
📋 По умолчанию: 500 токенов  
  
ℹ️ Информация:  
• Токены определяют максимальную длину ответа GPT  
• Больше токенов = более длинные ответы  
• Меньше токенов = более короткие ответы  
  
Команды управления:  
• /settokens КОЛИЧЕСТВО - установить новый лимит  
• /resettokens - сброс к значению по умолчанию

### /settokens - Установка лимита

/settokens КОЛИЧЕСТВО

Устанавливает новый лимит токенов для ответов GPT.

**Параметры:** - КОЛИЧЕСТВО - число от 50 до 4000

**Примеры:**

/settokens 300 # Короткие ответы  
/settokens 800 # Средние ответы  
/settokens 1500 # Длинные ответы

**Пример ответа:**

✅ Лимит токенов обновлен:  
  
🎯 Новый лимит: 800 токенов  
  
Изменения вступят в силу для новых запросов к GPT.

### /resettokens - Сброс к умолчанию

/resettokens

Сбрасывает лимит токенов к значению по умолчанию из файла .env.

**Пример ответа:**

✅ Лимит токенов сброшен к значению по умолчанию:  
  
🎯 Текущий лимит: 500 токенов  
📋 Значение по умолчанию: 500 токенов  
  
Изменения вступят в силу для новых запросов к GPT.

## Рекомендуемые значения

### 🔹 **Короткие ответы (200-400 токенов)**

* **Применение:** Быстрые, лаконичные ответы
* **Подходит для:** Простых вопросов, базовой поддержки
* **Примерная длина:** 1-2 предложения

### 🔸 **Средние ответы (500-800 токенов)**

* **Применение:** Сбалансированные ответы (по умолчанию)
* **Подходит для:** Большинства случаев психологической поддержки
* **Примерная длина:** 2-4 предложения

### 🔶 **Длинные ответы (1000-1500 токенов)**

* **Применение:** Подробные, развернутые ответы
* **Подходит для:** Сложных эмоциональных ситуаций
* **Примерная длина:** 4-8 предложений

### 🔺 **Очень длинные ответы (2000+ токенов)**

* **Применение:** Максимально подробные ответы
* **Подходит для:** Особых случаев, требующих детального разбора
* **Примерная длина:** 8+ предложений

## Ограничения

* **Минимум:** 50 токенов
* **Максимум:** 4000 токенов
* **По умолчанию:** 500 токенов (настраивается в .env)

## Влияние на работу бота

### ✅ **Больше токенов:**

* Более подробные и развернутые ответы
* Лучше для сложных эмоциональных ситуаций
* Больше расход API OpenAI
* Дольше генерация ответа

### ⚡ **Меньше токенов:**

* Более краткие и сжатые ответы
* Быстрее генерация ответа
* Меньше расход API OpenAI
* Подходит для простых случаев

## Мониторинг в статистике

Текущий лимит токенов отображается в команде /stats:

📊 Статистика бота:  
  
🎯 Настройки GPT:  
• Лимит токенов: 800  
  
...

## Техническая информация

### Файлы системы

* **data/tokens.txt** - хранит текущий лимит токенов
* **.env** - содержит значение по умолчанию (MAX\_TOKENS=500)
* **config.py** - функции управления токенами
* **gpt.py** - использует лимит при запросах к GPT-4

### Применение изменений

* Изменения **вступают в силу немедленно** для новых запросов
* Текущие активные запросы используют старый лимит
* Перезапуск бота **не требуется**

### Персистентность

* Настройки сохраняются в файле data/tokens.txt
* При перезапуске бота настройки загружаются автоматически
* Если файл отсутствует, используется значение из .env

## Примеры использования

### Сценарий 1: Увеличение детализации

/tokens # Проверяем текущий лимит  
/settokens 1200 # Увеличиваем для более подробных ответов

### Сценарий 2: Экономия ресурсов

/settokens 300 # Уменьшаем для экономии API

### Сценарий 3: Возврат к стандарту

/resettokens # Сбрасываем к значению по умолчанию

## Мониторинг и оптимизация

### 📊 **Отслеживание эффективности:**

1. Используйте /stats для мониторинга
2. Анализируйте качество ответов пользователей
3. Корректируйте лимит в зависимости от потребностей

### 🎯 **Оптимальная настройка:**

* Начните с **500 токенов** (стандарт)
* Увеличьте до **800-1000** если нужны более подробные ответы
* Уменьшите до **300-400** для экономии ресурсов

### ⚠️ **Важные моменты:**

* Слишком низкий лимит может обрезать важную информацию
* Слишком высокий лимит увеличивает расходы на API
* Оптимальное значение зависит от специфики использования бота

## Устранение неполадок

### Команды не работают

* Убедитесь, что у вас есть права администратора
* Проверьте правильность синтаксиса команды

### Изменения не применяются

* Проверьте права доступа к файлу data/tokens.txt
* Убедитесь, что директория data/ существует

### Ошибки сохранения

* Проверьте свободное место на диске
* Убедитесь в корректности прав доступа к файловой системе

# 📘 Project Best Practices

## 1. Project Purpose

This is an asynchronous Telegram bot for psychological support service called “Изливание души” (Soul Outpouring). Users send voice messages to express their feelings and receive empathetic voice responses generated by GPT-4. The bot provides a safe space for emotional expression with time-limited sessions and administrative oversight.

## 2. Project Structure

* **bot.py** - Main bot module with FSM (Finite State Machine) conversation handler
* **stt.py** - Speech-to-text functionality using OpenAI Whisper
* **tts.py** - Text-to-speech functionality using OpenAI TTS (onyx model)
* **gpt.py** - GPT-4 API integration for generating empathetic responses
* **admin.py** - Administrative commands (/prompt, /setprompt, /resetprompt, /stats, /cleanup)
* **utils.py** - Utilities for temp files, session timers, and admin notifications
* **config.py** - Configuration management and environment variable loading
* **data/** - Contains system prompt file (prompt.txt)
* **temp/** - Temporary audio files (auto-created, auto-cleaned)

## 3. Test Strategy

* **Framework**: No formal testing framework currently implemented
* **Testing Philosophy**: Manual testing and production monitoring through logging
* **Recommendations**:
  + Add pytest for unit testing core functions
  + Mock OpenAI API calls for testing
  + Test conversation flow states and error handling
  + Add integration tests for audio processing pipeline

## 4. Code Style

* **Language**: Python 3.10+ with async/await patterns throughout
* **Naming Conventions**:
  + Functions: snake\_case (e.g., handle\_voice\_message, send\_to\_admins)
  + Classes: PascalCase (e.g., PsychologyBot, SessionTimer)
  + Constants: UPPER\_SNAKE\_CASE (e.g., MAX\_MESSAGES\_PER\_SESSION)
  + Files: snake\_case.py
* **Async Usage**: All bot handlers and API calls are async
* **Type Hints**: Used in function signatures, especially for return types
* **Docstrings**: Triple-quoted docstrings for modules and functions
* **Error Handling**: Comprehensive try-catch blocks with specific error messages for users
* **Logging**: Structured logging with different levels (INFO, ERROR, DEBUG)

## 5. Common Patterns

* **FSM States**: Uses ConversationHandler with states (AWAIT\_NAME, MAIN\_MENU, RECORDING)
* **Temporary File Management**: Create with create\_temp\_file(), cleanup with cleanup\_temp\_file()
* **Admin Notifications**: Use send\_to\_admins() for debugging and monitoring
* **Session Management**: SessionTimer class for tracking session duration
* **Configuration**: Environment variables loaded via python-dotenv
* **Error Propagation**: Catch API errors and convert to user-friendly messages
* **Resource Cleanup**: Always cleanup temp files in finally blocks

## 6. Do’s and Don’ts

### ✅ Do’s

* Always use async/await for bot handlers and API calls
* Cleanup temporary files in finally blocks
* Log important events with appropriate log levels
* Validate user input before processing
* Use type hints for function parameters and returns
* Handle API rate limits and quota errors gracefully
* Check admin permissions before executing admin commands
* Use environment variables for sensitive configuration

### ❌ Don’ts

* Don’t store sensitive data in code (use .env file)
* Don’t forget to cleanup temporary audio files
* Don’t expose internal errors to users (convert to friendly messages)
* Don’t hardcode file paths (use Path objects from pathlib)
* Don’t ignore session time limits and message count limits
* Don’t send voice messages to admins unless DEBUG\_SEND\_VOICE=true
* Don’t use blocking I/O operations in async functions

## 7. Tools & Dependencies

* **python-telegram-bot v20.x** - Async Telegram bot framework
* **OpenAI API** - GPT-4 for responses, Whisper for STT, TTS for voice synthesis
* **pydub + ffmpeg** - Audio file processing and format conversion
* **python-dotenv** - Environment variable management
* **pathlib** - Modern path handling
* **asyncio** - Asynchronous programming support

### Setup Requirements

1. Install ffmpeg system dependency for audio processing
2. Configure .env file with TELEGRAM\_TOKEN, OPENAI\_API\_KEY, and ADMIN\_IDs
3. Ensure temp/ and data/ directories exist (auto-created)

## 8. Other Notes

* **Session Limits**: 30-minute sessions, max 10 messages per session, 7-minute voice message limit
* **Audio Processing**: Voice messages are converted to text, processed by GPT-4, then converted back to speech
* **Admin Features**: Real-time prompt modification, statistics, cleanup commands
* **Logging**: All sessions logged to session.log with duration and message count
* **Debug Mode**: DEBUG\_SEND\_VOICE controls whether voice messages are forwarded to admins
* **Error Recovery**: Bot continues operation even if individual components fail
* **Russian Language**: All user-facing messages and prompts are in Russian
* **State Management**: Uses telegram-bot’s ConversationHandler for managing user conversation states
* **File Safety**: Temporary files have unique timestamps to avoid conflicts
* **API Limits**: Handles OpenAI rate limits and quota exceeded errors gracefully