

■ Telegram Insights Bot - Полный проект

■ Резюме

Вы получили **полностью готовый к развертыванию** Telegram-бот для управления инсайдами по макрорегионам и отраслям. Проект включает:

- ✓ Полный исходный код на Python (aiogram 3.x)
- ✓ Интеграция с облачной БД Supabase (PostgreSQL)
- ✓ Экспорт данных в Excel с форматированием
- ✓ Поддержка прикрепления файлов
- ✓ Динамический подсчет записей на кнопках
- ✓ Пошаговая форма через FSM
- ✓ Конфигурация для развертывания 24/7 на Render/Railway
- ✓ Полная документация на русском языке

■ Структура проекта

```
insights-bot/
|
└── └── Python модули (основная логика)
    ├── main.py                  # Основной файл бота
    ├── database.py              # Работа с БД
    ├── export_excel.py          # Экспорт в Excel
    └── config.py                # Конфигурация
|
└── └── Конфигурация
    ├── requirements.txt         # Зависимости
    ├── .env.example             # Пример конфига
    ├── Procfile                 # Для Render
    ├── runtime.txt               # Версия Python
    └── .gitignore                # Git исключения
|
└── └── Документация
    ├── README.md                # Полное руководство
    ├── INSTALL.md                # Быстрый старт
    ├── install.sh                # Скрипт (Linux/macOS)
    └── install.bat                # Скрипт (Windows)
```

1 Быстрый старт (5 минут)

1.1 Подготовка Supabase БД

1. supabase.com → Create project
2. SQL Editor → выполнить SQL (см. README.md)
3. Settings → API → скопировать SUPABASE_URL и SUPABASE_KEY

2. Создание Telegram бота

1. Telegram → @BotFather → /newbot
2. Следовать инструкциям
3. Получить BOT_TOKEN

3. Локальный запуск

Windows:

```
install.bat
```

macOS/Linux:

```
chmod +x install.sh
./install.sh
```

4. Заполнить .env файл

```
BOT_TOKEN=ваш_токен
SUPABASE_URL=https://ваш-проект.supabase.co
SUPABASE_KEY=ваш_ключ
WEBHOOK_URL=http://localhost:8000
```

5. Запустить бота

```
python main.py
```

Откройте Telegram и напишите боту /start

☁ Развертывание на Render (24/7)

Шаг 1: GitHub репозиторий

```
git init  
git add .  
git commit -m "Telegram bot"  
git push to GitHub
```

Шаг 2: Render.com

1. **New Web Service** → выбрать репозиторий
2. **Build Command**: pip install -r requirements.txt
3. **Start Command**: gunicorn --bind 0.0.0.0:\$PORT main:app --worker-class aiohttp.GunicornWebWorker
4. **Environment variables**:
 - BOT_TOKEN
 - SUPABASE_URL
 - SUPABASE_KEY
 - WEBHOOK_URL (будет указан после создания)

Шаг 3: Update WEBHOOK_URL

Скопировать URL приложения Render (вида <https://insights-bot-xxxx.onrender.com>) и обновить в Environment.

Готово! Бот работает 24/7! ✅

☰ Функциональность бота

☰ Пользователь видит:

Главное меню:

- + Создать новый инсайт
- ☰ Поиск и просмотр
- ☰ Экспорт в Excel

Процесс создания инсайта:

1. Введи тему
2. Введи описание
3. Выбери макрорегион (МСК, ЦФО, СЗФО, УФО, ЮФО, ПФО, СДФО, СНГ)

4. Выбери отрасль (Оборона, Промышленность, Торговля, Банки, Нефть и газ, Энергетика)
5. Прикрепи файл или пропусти
6. ✓ Инсайт создан!

Поиск:

1. Выбери регион (видно кол-во записей на кнопке)
2. Выбери отрасль (видно кол-во записей на кнопке)
3. Смотри результаты (можно листать)
4. Скачивай прикрепленные файлы

Экспорт:

1. Нажми "≡ Экспорт в Excel"
2. Получи файл со всеми инсайтами

☰ Основные компоненты

main.py (800+ строк)

- Обработчик всех команд Telegram
- FSM для пошагового ввода данных
- Inline клавиатуры с динамическим подсчетом
- Webhook для работы на хостинге

database.py (250+ строк)

- Функции для CRUD операций
- Подсчет записей по полям
- Фильтрация и поиск
- Асинхронная работа с Supabase

export_excel.py (200+ строк)

- Экспорт в Excel с форматированием
- Автоматическая подгонка ширины столбцов
- Стилизация заголовков
- Обработка больших объемов данных

□ Безопасность

- ✓ Переменные окружения не коммитятся (.gitignore)
- ✓ API ключи Supabase хранятся в переменных окружения
- ✓ Row Level Security доступна в Supabase
- ✓ Все данные хранятся в защищенной БД

□ Примеры использования

Добавить новый макрорегион

```
# main.py
MACRO_REGIONS = [..., "НОВЫЙ_РЕГИОН"]
```

Добавить новую отрасль

```
# main.py
INDUSTRIES = [..., "НОВАЯ_ОТРАСЛЬ"]
```

Добавить логирование в файл

```
# main.py - добавить в logging.basicConfig
handlers=[
    logging.FileHandler('bot.log'),
    logging.StreamHandler()
]
```

□ Масштабирование

Метрика	Бесплатный план	Платный план
Supabase БД	500 МБ	от 100 ГБ
Render часов	750/месяц	неограниченно
Пользователей	неограниченно	неограниченно
Хранение данных	1 год	неограниченно

□ Что дальше?

1. **Локальное тестирование** (5 мин): запустить install.bat/install.sh
2. **Развертывание на Render** (10 мин): GitHub → Render
3. **Мониторинг** (постоянно): проверять логи

4. **Расширение** (по необходимости): добавлять функции

?

Решение проблем

Бот не отвечает

- Проверить BOT_TOKEN у @BotFather
- Проверить логи на Render/Railway
- Убедиться, что webhook установлен

Ошибка БД

- Проверить SUPABASE_URL и SUPABASE_KEY
- Убедиться, что таблица создана
- Проверить, что ключ типа "anon"

Excel не скачивается

- Убедиться, что есть данные в БД
- Проверить права доступа к /tmp
- Увеличить timeout

□ Полезные ссылки

- [Aiogram 3.x документация](#)
- [Supabase документация](#)
- [Telegram Bot API](#)
- [Render документация](#)
- [Python 3.11](#)

★ Особенности реализации

- ✓ **Асинхронность:** полностью асинхронный код с async/await
- ✓ **FSM:** машина состояний для управления диалогами
- ✓ **Динамические кнопки:** подсчет записей в реальном времени
- ✓ **Файловая поддержка:** прикрепление и скачивание файлов
- ✓ **Excel экспорт:** с форматированием и стилизацией
- ✓ **Webhook:** работа 24/7 без polling
- ✓ **Error handling:** обработка ошибок и логирование
- ✓ **Docker-ready:** готово к контейнеризации

└ Поддержка

Все файлы хорошо задокументированы с комментариями на русском языке.

При вопросах — обратитесь к [README.md](#) и [INSTALL.md](#)

Спасибо за использование проекта! └

Версия: 1.0.0

Дата: Ноябрь 2025

Лицензия: Открытый исходный код