









Справка по Telegram боту - Управление строительными отчетами

Назначение бота

Система управления строительными отчетами с workflow-процессами одобрения и контроля строительных работ.





Роли пользователей

Основные роли:





-  **Owner (Владелец)** - полный доступ ко всем функциям
 -  **Админ** - управление пользователями и справочниками
 -  **Менеджер** (уровень 1/2) - надзор за дисциплинами
 -  **Супервайзер** - создание отчетов о работах
 -  **Мастер** - подтверждение отчетов
 -  **КИОК** - финальное согласование отчетов
 -  **Бригадир** - подача табелей
 -  **ПТО** - технический надзор
-

РАБОТАЮЩИЕ ФУНКЦИИ

Авторизация и пользователи

-  **Регистрация новых пользователей** через /start
-  **Система одобрения заявок** админами
-  **Управление ролями** (добавление, редактирование, удаление)
-  **Мультиязычность** (русский, английский, узбекский)

Отчеты и workflow

-  **Создание отчетов** супервайзерами
-  **Workflow одобрения:** Супервайзер → Мастер → КИОК
-  **Статусы отчетов:** черновик, ожидание, одобрен, отклонен
-  **Комментарии при отклонении**

Табели бригадиров

-  **Подача ежедневных табелей** с разбивкой по ролям

- ☒ **Контроль подачи табелей** по датам
- ☒ **Система предупреждений** о неподанных табелях



Аналитика и отчетность

- ☒ **Дашборды по дисциплинам** с метриками
- ☒ **HR-отчеты** по персоналу на дату
- ☒ **Аналитика проблемных бригад**
- ☒ **Экспорт данных в Excel**



Справочники

- ☒ **Дисциплины** - отрасли строительства
- ☒ **Корпуса** - объекты строительства
- ☒ **Виды работ** - типы строительных работ
- ☒ **UPSERT логика** - умное обновление данных



Импорт/Экспорт данных

- ☒ **Импорт справочников** из Excel (3 листа)
- ☒ **Полное восстановление БД** из Excel
- ☒ **ID контроль**: указанные ID используются, пустые - автогенерация
- ☒ **Отключение внешних ключей** при восстановлении
- ☒ **Экспорт шаблонов** для редактирования
- ☒ **Резервное копирование БД**



Администрирование

- ☒ **Управление пользователями** с пагинацией
- ☒ **Сброс табелей** бригадиров
- ☒ **Управление данными БД**
- ☒ **Полная резервная копия** системы






ЧАСТИЧНО РАБОТАЮЩИЕ ФУНКЦИИ



Workflow процессы

- ☐ **Подтверждение мастером** - базовая логика есть, нужно тестирование
- ☐ **КИОК согласование** - может потребовать доработки
- ☐ **Уведомления между ролями** - планировщик работает, нужна проверка

ConversationHandler'ы

-  **Создание отчетов** - основная логика есть, возможны баги состояний
 -  **Подача табелей** - работает, но может нуждаться в доработке
 -  **Авторизация** - переписана, требует тестирования
-

ТРЕБУЕТ ДОРАБОТКИ

Известные проблемы

- **Состояния ConversationHandler** - возможны конфликты между handlers
- **Расширенная аналитика** - некоторые отчеты могут быть неполными
- **Обработка ошибок** - нужно больше проверок edge cases
- **Производительность** - при больших объемах данных

Планируемые улучшения

- **Веб-интерфейс** для админов
 - **API для интеграции** с внешними системами
 - **Более детальная аналитика** и дашборды
 - **Система бекапов** по расписанию
 - **Мобильные уведомления**
-

Структура проекта

```

📁 bot/
├── 📁 handlers/      # Обработчики команд
├── 📁 conversations/ # Диалоговые процессы
├── 📁 middleware/     # Безопасность и состояния

📁 services/          # Бизнес-логика
├── import_service.py  # Импорт/экспорт данных
├── workflow_service.py # Процессы одобрения
├── analytics_service.py # Аналитика и отчеты

📁 database/          # Работа с БД
├── connection.py      # Подключения (sync/async)
├── queries.py         # SQL запросы
├── migrations.py      # Миграции схемы

📁 utils/             # Утилиты
├── localization.py    # Мультиязычность
├── constants.py       # Константы
├── chat_utils.py      # Телеграм утилиты
```

Запуск и конфигурация

Переменные окружения (.env):

```
bash

# Telegram
TOKEN=ваш_токен_бота
OWNER_ID=ваш_telegram_id

# База данных
DATABASE_URL=postgresql://user:pass@host:port/dbname

# Группы
REPORTS_GROUP_URL=ссылка_на_группу_отчетов
```

Запуск:

```
bash







python main.py
```

Миграции:

```
bash
```

Статистика проекта

Готовность функций:

-  **Авторизация:** 90%
-  **Справочники:** 95%
-  **Импорт/Экспорт:** 95%
-  **Workflow:** 75%
-  **Аналитика:** 80%
-  **Администрирование:** 90%

Общая готовность: ~85%

Установки для нового чата

Контекст проекта:

ПРОЕКТ: Telegram бот для управления строительными отчетами
СТАТУС: 85% готов, основные функции работают
АРХИТЕКТУРА: PostgreSQL + Telegram Bot + Workflow система
ПОСЛЕДНЕЕ: Завершили импорт/экспорт с ID контролем





Приоритеты работы:

1. **Тестирование workflow** - проверить создание и одобрение отчетов
2. **Отладка ConversationHandler** - устранить конфликты состояний
3. **Производительность** - оптимизация при больших данных
4. **UX улучшения** - более понятные сообщения и меню

Стиль работы:

- Сначала уточняю детали ОДНИМ сообщением
- Жду подтверждения перед кодом
- Один готовый рабочий вариант (без вариантов)
- Краткие комментарии, фокус на решении
- Формат: План → Структура → Diff → Тест

Критические файлы:

- `services/import_service.py` - универсальная UPSERT логика 
- `services/workflow_service.py` - процессы одобрения 
- `bot/handlers/` - все обработчики команд 
- `database/migrations.py` - схема БД 

ГОТОВ ПРОДОЛЖАТЬ РАЗРАБОТКУ! 