

Telegram-бот для автоматизации внутренних процессов

Технологии: Python, aiogram, FSM, SQLite

Формат: CLI-бот с доступом по Telegram ID и ролевой системой

1. Общая информация о проекте

Цель проекта — создать надёжный и понятный Telegram-бот для автоматизации внутренних процессов. Основной задачей является предоставление закрытого доступа к функциям через систему ролей (гость, участник, админ), управление которыми осуществляется вручную через Telegram ID.

Бот будет использоваться в рамках закрытого сообщества (например, ассоциация, организация, клуб), и должен гарантировать:

- безопасность и приватность;
- простую навигацию;
- поддержку многопользовательского режима с разными уровнями доступа;
- лёгкую масштабируемость.

Реализация включает серверную часть на Python, асинхронную архитектуру с FSM-механизмом (Finite State Machine) и хранение пользовательских данных в SQLite-базе.

2. Функциональные требования

2.1 Команды и взаимодействие с пользователем:

- `/start` — приветствие, проверка Telegram ID, определение роли, переход в соответствующее меню.
- `/myid` — вывод Telegram ID пользователя (для передачи администратору).
- `/setrole <user_id> <role>` — (только для администратора) назначение роли пользователю.
- `/help` — краткая справка по командам и функциям (опционально).

2.2 Состояния и меню (FSM):

- Для участников — интерактивное меню (например, «Проверить статус», «Подать заявку», «Обратная связь»).
- Для гостей — сообщение с ограничением доступа и инструкцией, как получить роль.
- Для администраторов — скрытое админ-меню с доступом к изменению ролей.

2.3 Уровни доступа:

- `guest` — доступ ограничен, только информация о Telegram ID и просьба связаться с админом.
- `member` — доступ к основному функционалу FSM.
- `admin` — доступ к управлению ролями, журналу действий и будущим админ-функциям.

2.4 Хранение данных:

- Таблица `users` с полями: `user_id`, `role`, `created_at`, `updated_at`.
- Возможность расширить таблицу (например, добавить имя, email и логи активности).

3. Архитектура и реализация

3.1 Стек технологий:

- Язык: Python 3.10+
- Библиотека: `aiogram` (второе поколение Telegram API, поддерживает FSM)
- База данных: SQLite (встроенная, не требует отдельного сервера)
- Хостинг: локальный сервер / VPS / cloud (например, Yandex Cloud, Hetzner)
- Хранение токена: `.env` файл или секреты на сервере

3.2 Основные модули:

- `bot.py` — инициализация, маршруты команд, запуск FSM
- `states.py` — определения состояний (FSM)
- `db.py` — работа с SQLite: создание базы, добавление и обновление пользователей
- `admin.py` — команды администратора

- `utils.py` — функции-помощники (например, генерация меню)

3.3 Логика доступа:

Каждая команда обёрнута в проверку по роли пользователя. FSM запускается только если роль — `member`. Все команды `admin` доступны только ID, указанному как `ADMIN_ID` в конфиге.

4. Безопасность, надёжность и ограничения

4.1 Проверка идентичности:

В системе отсутствует авторизация по паролю или телефону — идентификатором служит Telegram ID. Поэтому:

- любые команды, кроме `/start` и `/myid`, требуют предварительного назначения роли админом.
- Telegram ID можно получить, отправив команду `/myid` и передав его вручную администратору.

4.2 Защита от ошибок:

- Все команды имеют `try/catch` обработку ошибок.
- База данных проверяется на наличие записи перед выполнением операций обновления.
- Действия администратора логируются (в будущем — лог-файл или вывод в Telegram).

4.3 Ограничения:

- Пользователь может быть только в одной роли.
- Пока отсутствует полноценный интерфейс для админа (только команды).
- SQLite подходит для проектов до ~1000 пользователей, далее рекомендуется PostgreSQL или Supabase.

5. Масштабирование и перспективы развития

5.1 Возможные доработки:

- Мультиязычная поддержка (английский, французский, армянский).

- Веб-интерфейс для администратора: просмотр базы, ролей, логов.
- Интеграция с внешними API (например, платёжные системы, CRM, email-уведомления).
- Поддержка голосовых команд и кнопок inline-меню.
- Возможность добавления категорий заявок или подачи форм с подтверждением.
- Ведение отчётности (например, выгрузка в Excel/CSV).
- Настройка времени действия роли (временные участники).

5.2 Интеграция с другими системами:

- Привязка к корпоративному сайту или внутренней CRM.
- Экспорт пользовательской активности в Google Sheets или BI-панели.
- Получение уведомлений по email или в Slack.

5.3 Автоматизация:

- Таймеры: автоматическая смена статуса, напоминания.
- Проверка активности участников (кто не заходил N дней).
- Интеграция с Telegram Web Apps (красивые формы поверх бота).