Паспорт проекта: Spysky

Программа управления списками и напоминаниями в Telegram

Автор: Щелкунов Тимофей, Стройков Арсений

Общая информация

Этот проект — это Telegram-бот, реализованный на Python с использованием библиотеки aiogram, предназначенный для удобного управления списками, их совместного использования, установки напоминаний и организации работы с данными. Скрипт ориентирован на личное использование и автоматизацию ежедневных задач по ведению списков.

Цели проекта:

- Создавать, редактировать и удалять списки

- Добавлять и управлять элементами внутри списков

- Обеспечивать совместный доступ к спискам между пользователями

- Устанавливать напоминания по времени для конкретных списков

- Экспортировать списки в текстовые файлы для внешнего использования

- Обеспечивать удобный интерфейс через inline-кнопки и команды бота

- Автоматически обновлять структуру базы данных для поддержки новых функций

- Вести логирование событий и ошибок для диагностики и мониторинга

Используемые технологии и библиотеки

- asyncio — асинхронное выполнение задач

- aiogram — асинхронная работа с Telegram API

- sqlite3 — встроенная библиотека для хранения данных

- json — сериализация и десериализация списков элементов

- logging — ведение логов событий, ошибок и информационных сообщений

- datetime — управление датой и временем для напоминаний

- typing — типизация данных для повышения читаемости и надежности кода

Ключевые компоненты и структура проекта

1. \*\*Конфигурация и инициализация\*\*

- Токен бота (`API\_TOKEN`) для подключения к Telegram

- Путь к базе данных (`DB\_PATH`)

- Настройка логирования с уровнем INFO для отслеживания событий

2. \*\*База данных\*\*

- Таблица lists с полями:

- user\_id — ID пользователя, владеющего списком

- list\_name — название списка

- items — список элементов (в JSON-формате)

- shared\_with — список пользователей, имеющих доступ (в JSON)

- reminders — время напоминания (в строковом формате)

- archived — статус архивации (0 или 1)

- created\_at — дата создания

- Автоматическая проверка наличия колонок shared\_wit`, reminders, archived и добавление их при необходимости

3. \*\*Обработчики команд\*\*

- /start — стартовое сообщение и инициализация

- /help — список команд и инструкций

- /create\_list — создание нового списка (запрос названия)

- /add\_to\_list — добавление элемента в выбранный список

- /show\_lists — просмотр всех активных списков пользователя

- /delete\_list — удаление выбранного списка

- /delete\_item — удаление элемента из списка

- /search — поиск по содержимому всех списков

- /export — экспорт выбранного списка в текстовый файл

- /share — совместное использование списка с другим пользователем по ID

- /remind — установка времени напоминания по формату ЧЧ:ММ

- /archive — архивация списка

- /unarchive — восстановление из архива

4. \*\*Обработка сообщений\*\*

- Ввод названий списков и элементов через состояния пользователя

- Обработка поиска, установки напоминаний и совместного доступа

- Валидация данных (например, формат времени)

5. \*\*Инлайн-кнопки и меню\*\*

- Для выбора списков, элементов и подтверждения действий

- Навигация по спискам и операциям (удаление, экспорт, деление доступа, архивирование)

6. \*\*Фоновая задача\*\*

- Постоянный мониторинг напоминаний

- Каждую минуту проверяет текущие напоминания и отправляет уведомления пользователям, если пришло время

- После отправки напоминания очищает или обновляет соответствующее поле

7. \*\*Логирование и обработка ошибок\*\*

- Ведение логов для событий, ошибок и исключений

- Использование расширенного форматирования трассировки ошибок для удобства диагностики

Общий сценарий работы

Пользователь запускает бота, получает инструкцию и команды для начала работы. Он создает списки, добавляет элементы, делится ими или устанавливает напоминания. Вся работа осуществляется через команды и кликабельные inline-кнопки, что повышает удобство взаимодействия. В фоне бот автоматически отслеживает время напоминаний и уведомляет пользователя. Также реализована возможность экспорта списков для дальнейшего использования вне Telegram.

Особенности и расширяемость

- Скрипт легко расширяется добавлением новых команд и функций

- Возможна интеграция с внешними сервисами или API

- Структура базы данных допускает добавление новых полей и таблиц при необходимости

- Логика обработки состояний обеспечивает последовательное выполнение команд

- Переменные и функции структурированы для удобства поддержки и развития

Заключение

Данный проект — это полнофункциональный Telegram-бот для управления списками, напоминаниями и совместной работой. Он написан на Python с использованием библиотеки aiogram, использует SQLite для хранения данных и включает автоматическую поддержку структуры базы данных. В проекте реализованы основные операции по управлению списками, а также автоматизация уведомлений и удобный интерфейс через inline-кнопки. Проект может служить основой для более сложных решений или внедрений в командные системы.