

Отчет о текущем состоянии проекта "TelePost Manager"

Дата: 07.05.2024

I. Обзор текущего состояния и реализованного функционала

Проект "TelePost Manager" представляет собой Telegram-бота, разработанного для автоматизации и упрощения процесса управления публикациями в Telegram-каналах. На данном этапе разработки основной акцент был сделан на создании надежной многопользовательской архитектуры и реализации ключевых функций, связанных с постингом и управлением контентом.

Ключевые достижения и реализованные возможности:

1. Многопользовательская система:

- **Изоляция данных пользователей:** Реализована полная изоляция данных для каждого пользователя бота. Каналы, шаблоны (личные) и история публикаций привязаны к уникальному `user_id` пользователя в Telegram. Это гарантирует, что пользователи видят и управляют только своей информацией.
- **Регистрация и идентификация:** При первом взаимодействии (команда `/start`) данные пользователя (ID, имя и т.д.) автоматически сохраняются/обновляются в базе данных (`bot_users`).
- **Система администрирования:**
 - Внедрен механизм "супер-администратора", чей ID задается через конфигурацию (`SUPER_ADMIN_ID`). Этот пользователь автоматически получает административные права в системе.
 - Разработан кастомный фильтр `isAdmin`, позволяющий ограничивать доступ к определенным функциям бота только для администраторов.
 - Администраторы имеют возможность создавать "общие" (системные) шаблоны, доступные всем пользователям.

2. Управление каналами:

- Пользователи могут подключать свои Telegram-каналы к боту путем пересылки сообщения из канала. В процессе добавления выполняется проверка наличия у бота необходимых административных прав в целевом канале (включая право на публикацию).
- Список подключенных каналов хранится в БД с привязкой к `user_id` и доступен пользователю для просмотра и управления (удаления).

- Все сообщения, касающиеся управления каналами, используют HTML-форматирование для безопасного отображения динамических данных (названия каналов и т.д.).

3. Управление шаблонами постов:

- Реализована двухуровневая система шаблонов:
 - **Общие шаблоны:** Создаются администратором бота (`user_id=0`) и доступны для использования всем пользователям.
 - **Личные шаблоны:** Создаются конкретным пользователем (`user_id=ID_пользователя`) и доступны только ему.
- Пользователи могут просматривать список доступных шаблонов (общие + свои личные), создавать новые личные шаблоны и удалять их.
- Шаблоны поддерживают текстовое содержимое, один прикрепленный медиафайл (фото/видео) и предопределенные переменные (например, `{дата}`, `{автор}`), которые автоматически подставляются при генерации поста.
- Администратор имеет специальный интерфейс (кнопка в меню шаблонов) для создания общих шаблонов.

4. Создание и публикация постов:

- Процесс создания поста реализован через пошаговый диалог (FSM).
- Поддерживается создание постов с использованием шаблонов (общих или личных) или полностью вручную.
- Реализована работа с текстовым контентом и медиафайлами (фото/видео), включая сохранение типа медиа (`media_type`) для корректной отправки.
- Перед публикацией пользователю отображается предпросмотр поста с использованием HTML-форматирования.
- Доступны опции немедленной и запланированной публикации (с использованием `APScheduler`).
- Каждый пост привязывается к `user_id` автора.

5. Контроль контента:

- Внедрена система фильтрации по списку запрещенных слов. Проверка выполняется при вводе текста для поста или при генерации текста из шаблона.
- При обнаружении запрещенных слов пользователь получает уведомление и должен исправить текст.

6. История публикаций:

- Ведется история всех опубликованных и запланированных постов, с привязкой к `user_id` автора.

- Пользователи могут просматривать *свою* историю публикаций с пагинацией. В истории отображается статус поста и ссылка на него в канале (для опубликованных).

7. Пользовательский интерфейс и архитектура:

- Реализована навигация с помощью стандартных команд (/start, /help, /cancel) и кнопочных меню (ReplyKeyboardMarkup и InlineKeyboardMarkup).
- Проведен рефакторинг кода: утилиты вынесены в bot_utils.py, инициализация основных компонентов централизована в loader.py.
- Приняты меры по улучшению обработки ошибок и внедрено базовое логирование событий.

II. Задачи для дальнейшей разработки и улучшения:

Несмотря на достигнутый прогресс, для полноценного завершения проекта и расширения его возможностей необходимо реализовать следующий функционал:

1. Доработка модуля истории публикаций:

- Обеспечить корректное отображение информации о канале в истории, даже если пользователь удалил его из своего списка в боте (рассмотреть возможность хранения channel_title в таблице posts или использовать LEFT JOIN с обработкой NULL).
- Реализовать для пользователей возможность отмены и, в перспективе, редактирования запланированных постов через интерфейс истории.

2. Расширение системы контроля контента:

- Предоставить администратору бота интерфейс для управления глобальным списком запрещенных слов (CRUD-операции).
- (Опционально) Рассмотреть возможность добавления пользовательских списков запрещенных слов для их каналов.

3. Управление запланированными публикациями:

- Реализовать функцию просмотра списка *своих* запланированных постов.
- Добавить возможность отмены *своего* запланированного поста (с удалением задачи из APScheduler и обновлением статуса в БД).

4. Развитие административных функций:

- Реализовать для администратора интерфейс для управления общими шаблонами (редактирование, удаление).
- (Опционально) Добавить сбор базовой статистики использования бота (количество пользователей, постов и т.д.).

5. Общие улучшения и полировка:

- Детализация обработки ошибок и предоставление более информативной обратной связи пользователю.
- Реализация функции редактирования уже созданных пользователем личных шаблонов.
- Усиление логирования для более эффективной отладки и мониторинга работы бота.

6. Тестирование и документация:

- Проведение всестороннего тестирования всех реализованных функций, включая сценарии с участием нескольких пользователей.
- Создание детальной документации по проекту (README.md), описывающей его функционал, команды и инструкции по развертыванию/использованию.

III. Приоритетные задачи на ближайший период:

1. Доработка отображения каналов в истории публикаций.
2. Реализация просмотра и отмены запланированных постов.
3. Внедрение административного интерфейса для управления списком запрещенных слов.
4. Продолжение интенсивного тестирования существующего функционала для выявления и исправления возможных недочетов.