Депозитарий https://github.com/aldav99/analystTelegram

**Файл app.py.** Развернут в сервисе Render.com  
 Telegram Channel Analyzer FastAPI with Comments

Веб-сервис для анализа Telegram каналов с комментариями из группы обсуждений.  
Извлекает статистику сообщений и комментарии к постам за указанный период.

Этот код представляет собой асинхронное веб-приложение на FastAPI для анализа Telegram-каналов.

Основные функции:

- Интеграция с Telegram API через библиотеку Telethon для получения данных канала и сообщений.

- Поддержка анализа постов, включая просмотры, реакции, пересылки и комментарии из групп обсуждений.

- Валидация входных данных с помощью Pydantic (имя канала, даты, диапазон анализа до 365 дней).

- Асинхронная обработка запросов для эффективной работы с большим количеством сообщений.

- Логирование для отслеживания работы и диагностики ошибок.

- Управление сессиями Telegram через файл или переменную окружения.

Эндпоинты API:

- /: базовая проверка работоспособности сервиса.

- /health: проверка состояния Telegram-клиента.

- /analyze: выполнение анализа канала с возвратом структурированных данных.

- /status: информация о состоянии сервиса и Telegram-клиента.

Особенности:

- Обработка различных типов медиа в постах (фото, видео, документы и т.д.).

- Устойчивость к ошибкам Telegram API, включая FloodWaitError и приватные каналы.

- Конфигурация через переменные окружения для безопасного хранения креденциалов.

**Файл create\_session.py.** Запускается один раз для

 Создания сессии для дальнейшей работы с Telegram API (например, для ботов, парсинга сообщений, автоматизации).

 Получения строки сессии, которую можно использовать в других приложениях или хранить в переменных окружения для безопасного доступа

После получения строки сессии она сохраняется в переменной окружения при развертывании файла app.py Telegram Channel Analyzer в сервисе Render.com. Таким образом отпадает необходимость регулярного получения кода для входа в Telegram.

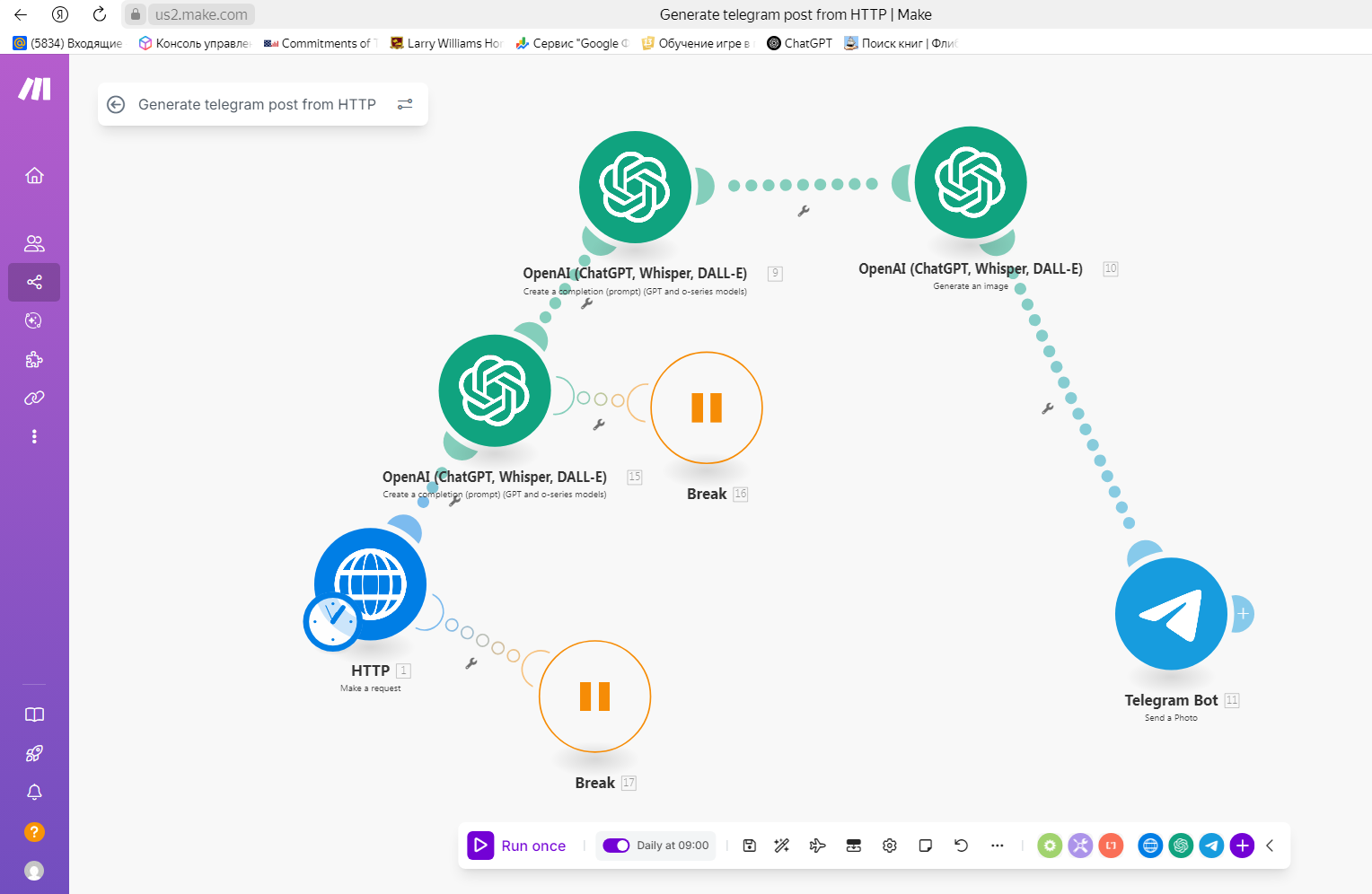
Депозитарий https://github.com/aldav99/bloggpt

**Файл app.py.** Развернут в сервисе Render.com

Это FastAPI-приложение, которое предоставляет API для генерации постов на основе последних новостей по заданной теме, используя внешний сервис Currents API. Созлает строку с последними новостями на заданную тему.

Интеграция с make.com

**Сценарий генерации поста**



По расписанию из модуля HTTP make a request запрос, в качестве входного параметра содержащий тему "Путешествия", направляется на **https://bloggpt-2-nx5h.onrender.com/generate-post** в Render.com для поиска последних новостей по теме.

Далее тема и последние новости передаютcя в модуль Open AI, где генерируется текст статьи при помощи промпта следующего содержания:  
" Напишите подробную статью на тему {{1.data.topic}} на русском языке, используя последние новости:{{1.data.recent\_news}}.

Статья должна быть:

1. Информативной и логичной

2. Содержать не менее 300 символов

3. Иметь четкую структуру с подзаголовками

4. Включать анализ текущих трендов

5. Иметь вступление, основную часть и заключение

6. Включать примеры из актуальных новостей

7. Каждый абзац должен быть не менее 2-3 предложений

8. Текст должен быть легким для восприятия и содержательным

9. Текст должен быть в формате plain text и не содержать управляющих символов.

10. Текст должен содержать не более 400 символов всего вместе с переводами строк, пробелами, знаками препинания.

В ответ выдай только текст статьи, ничего более."

" Дана статья для Telegram канала:{{15.result}}

Напиши мне промпт для генерации картинке по этой теме.

Следующий модуль Open AI генерирует описание картинки при помощи промпта

" Дана статья для Telegram канала:{{15.result}}

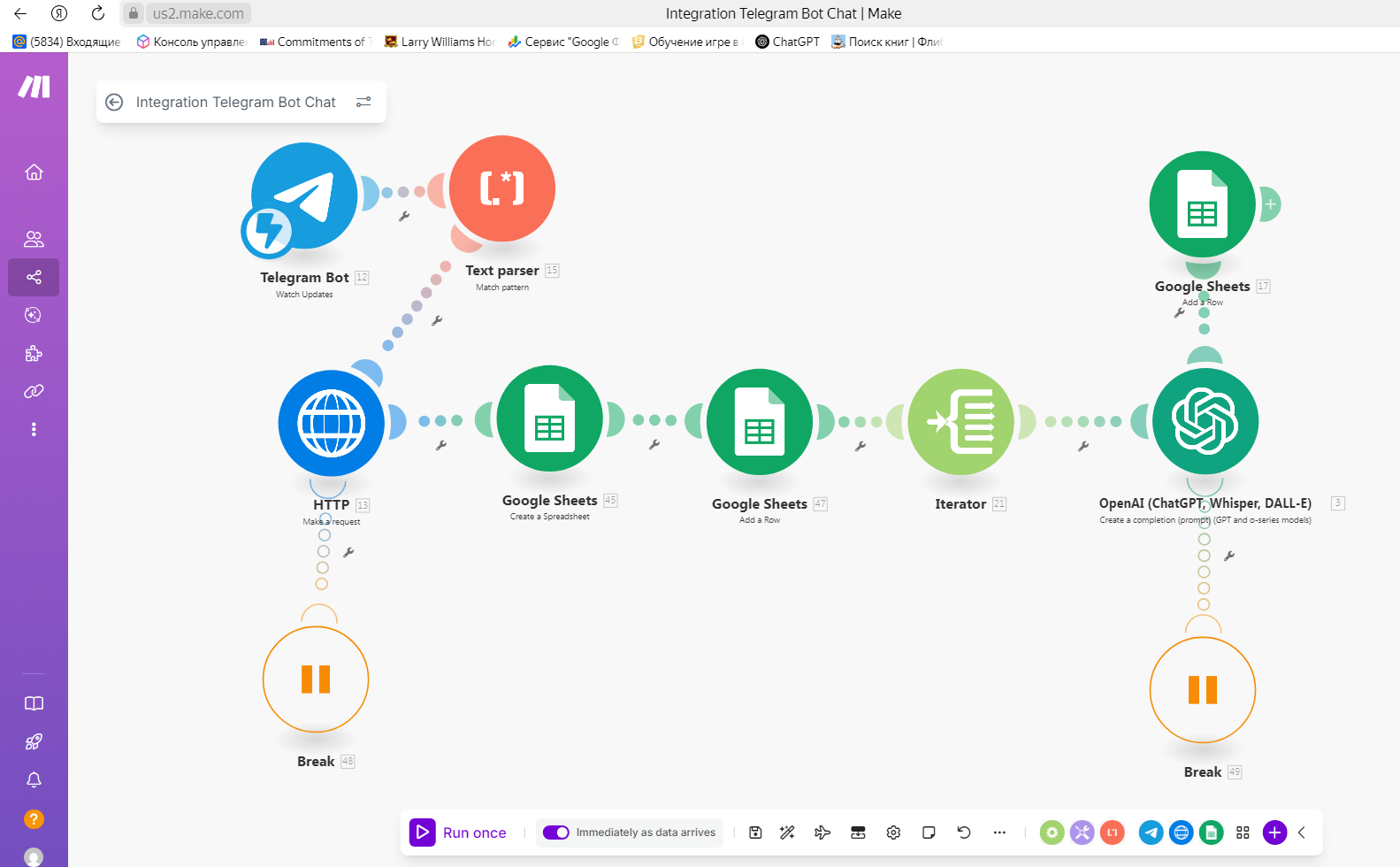
Напиши мне промпт для генерации картинке по этой теме.

Промпт на английиском языке. Ограничение 300 символов. В ответ выдай только промпт, ничего более. "

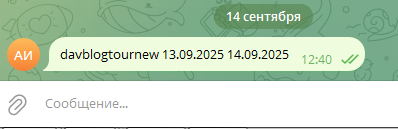
Следующий модуль Open AI, используя модель Dall-E 3, по описанию генерирует картинку, которая вместе с текстом статьи публикуется в Telegram канале при помощи модуля Telegram Bot Send a Photo.

Ошибки обрабатываются при помощи обработчика Break.

**Сценарий генерации статистики**



Модуль Telegram Bot Watch Updates отслеживает изменения в боте davAdmin. Информация из бота приходит в виде "Название канала, период отчета"



Далее в модуле Text Parser Match Pattern строка разбивается на значимые элементы - название канала, начальная дата отчета, конечная дата отчета и передается в модуль HTTP make a request на url https://analysttelegram.onrender.com/analyze в виде запроса POST с параметрами {"channel\_username": "{{15.`$1`}}", "start\_date": "{{15.`$2`}}", "end\_date": "{{15.`$3`}}"}, где код Telegram Channel Analyzer обрабатывает входящие данные.

Следующий модуль Google Sheets Create a Spreadsheet создает отчетную таблицу с именем в формате Telegram Report {{formatDate(now; "YYYY-MM-DD\_HH-mm-ss")}}, за ним модуль Google Sheets Add a Row добавляет строку заголовка столбцов.

Модуль Iterator в цикле передает по одному посты посты в модуль Open AI, который при помощи промпта

"Ты опытный аналитик контента, специализирующийся на анализе пользовательских реакций. Твоя задача — проанализировать эмоциональный окрас комментариев к посту в Telegram-канале и определить, является ли каждый из них позитивным или негативным по отношению к материалу.

Правила обработки:

1. Входные данные: Комментарии подаются в виде одной строки, где несколько комментариев объединены с разделителем `~~~~`.

2. Анализ: Ты должен проанализировать каждый комментарий отдельно.

3. Пустой ввод: Если на вход подается пустая строка, то и на выходе должна быть пустая строка.

4. Формат ответа:

\* Для каждого комментария дай краткий, односложный вердикт: "Понравилось" или "Не понравилось".

\* Не указывай авторов комментариев или их идентификаторы в ответах.

\* Ответ должен быть строкой, в которой исходный разделитель `~~~~` заменен на два символа перевода строки (`\n\n`).

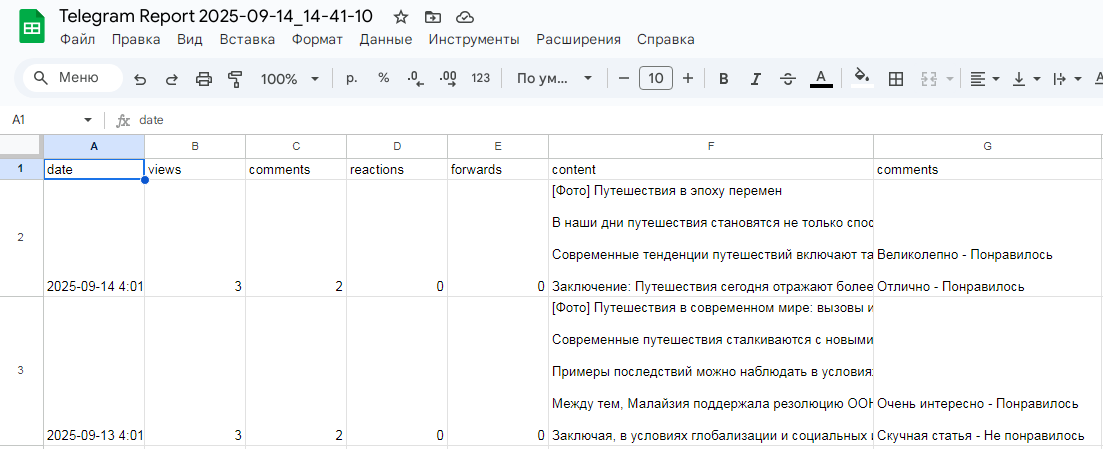
\* Рядом с каждым комментарием через тире и пробел (` - `) должен быть указан результат его оценки.

5. Важно: Ответы будут отправляться в Google Sheets, где не поддерживается HTML-разметка. Используй только чистый текст. Для визуального выделения в таблицах будет использовано условное форматирование на основе слов "Понравилось" и "Не понравилось".

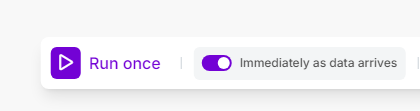
Теперь проанализируй предоставленные тебе комментарии, строго следуя формату ответа из примеров: {{21.post\_info.comments\_text}}"

Далее данные передаются в модуль Google Sheets Add a Row и сохраняются в Google Disk в виде

Ошибки обрабатываются при помощи обработчика Break.



Весь сценарий запускается по мере поступления данных из бота



Пример поста в Telegram-канале

Пояснение: разница между запланированным в make.com временем публикации (9:00) и временем публикации в Telegram-канале (7:00) определяется различием часовых поясов. Сервер make.com находится в Алматы ( Москва +2 часа)



Примечания:

Файл отчета https://docs.google.com/spreadsheets/d/13-7qwsX02slGrsJNhrQfFXMofUwL7tJzyhu1BqNEGIU/edit?usp=sharing

Репозитарии проекта:

https://github.com/aldav99/analystTelegram

https://github.com/aldav99/bloggpt