

AI Sputnik / Телеграм-бот с искусственным интеллектом

Его основная цель — служить онлайн-базой знаний компании и предоставлять быстрые ответы с ИИ на все вопросы по данным компании прямо в Telegram, не требует ИИ обучения или встраивания данных в модель ИИ, обновленные данные сразу используются для ответов.

Он также служит вашим универсальным помощником в области искусственного интеллекта для работы с личными данными (файлами), позволяя вам вести собственную ИИ базу знаний, к которой могут получить доступ другие пользователи/сотрудники.

The image shows a promotional landing page for AI Sputnik. At the top, the text "AI Sputnik" is displayed in large blue letters, followed by the subtitle "/ Telegram bot with artificial intelligence". Below this is a background image of a satellite in space, orbiting around the Earth. In the foreground, there is a screenshot of a mobile application window. The window has a dark theme and displays a message from the bot: "Hello! I am your AI Knowledge Base". Below this message is a button labeled "Ask me anything.". At the bottom of the application window is a text input field. To the left of the application window, there is some descriptive text: "Company AI Knowledge Base" followed by a bulleted list: "• Quick answers to company data" and "• No downloads, no manual updates". To the right of the application window, there is another section titled "Universal AI Assistant" with its own bulleted list: "• Personal data management", "• Shareable knowledge bases", and "• Employee collaboration". A small decorative star icon is located in the bottom right corner of the page.

AI Sputnik
/ Telegram bot with artificial intelligence

Hello! I am your AI Knowledge Base

Ask me anything.

Company AI Knowledge Base

- Quick answers to company data
- No downloads, no manual updates

Universal AI Assistant

- Personal data management
- Shareable knowledge bases
- Employee collaboration

Кодовая база

Хорошая ассоциация с обычной папкой в Windows.

Все файлы при отправки в папку автоматически конвертируются в текстовые с разметкой Markdown.

Можно применять различные агенты к этой папке:

Системные агенты (доступны всем):

- @rag - 🔎 RAG Интеллектуальный поиск по базе
- @filejob - 📁 Пакетная обработка файлов
- @zip - Распаковка целого архивов до 20Mb, конвертация документов и сохранение в кодовую базу

📁 Управление файлами:

/files - постраничный просмотр всех файлов

/search - поиск файлов по части имени

/download - скачивание файлов

/delete - удаление файлов



Кодовая база

📌 Команды:

/codebases - список ваших кодовых баз

/create_codebase - создать новую

/active - показать активную (*)

/switch - быстрое переключение

/make_public - сделать базу публичной

/hide_codebase - скрыть базу (необратимо)

/index_rag - проиндексировать активную базу для **RAG** (в большинстве случаев не требуется, только в исключительных случаях как например физическое копирование файлов вне телеграм и тп.)

(*) Единовременно может быть выбрана только одна активная кодовая база. Все файлы, отправленные в телеграм, будут распознаны как текст и сохранены в активную кодовую базу. Пользователю всегда будет предложен выбор: сохранить файл или перезаписать его, если он уже существует. Кроме того, все агенты будут использовать активную кодовую базу при вызове, если они разработаны для работы с кодовыми базами.

Публичная кодовая база

Доступна для работы агентов других пользователей, например @rag

 Управление доступом

Команды:

/public_list_users - список пользователей с доступом

/public_add_user <id> - добавить

/public_remove_user <id> - удалить

<id> - **Telegram ID**, можно увидеть используя команду **/settings**:

 **Telegram ID:** 1234567890

Кодовая база devnull



└ System null codebase

Команда: /select_cb_0

devnull — создается первым для каждого пользователя и имеет индекс 0. В эту базу данных ничего не может быть сохранено, а чтение из нее всегда возвращает пустой список файлов. Используется для агентов и передачи файлов, когда требуется только распознавание без сохранения файлов.

Агент @nethack

@nethack - работает с распределенной онлайн базой знаний компании.

Он умеет:

- Обрабатывать документы по ссылкам из сети
- Работать в простом режиме «вопрос/ответ»
- Управлять доступом к документам
Пользователи разбиваются на группы, для каждого документа можно указать какие группы имеют доступ.
- Автоматически синхронизировать данные для ИИ



Агент @chat

Возможности:

- Отправляйте текстовые сообщения для диалога
- Отправьте текстовый файл для добавления в контекст
- Загружайте несколько файлов для работы с ними
- Используйте /stop для завершения чата

Работа с файлами:

- Режим: объединение всех файлов
- Можно загрузить несколько файлов
- Все файлы будут объединены в один контекст
- Макс. размер: 195 КБ

Настройки истории:

- История сообщений: Включена (макс. 20 сообщений)
- При очистке истории удаляются и файлы

Команды управления:

- /clear_history - очистить историю и файлы
- /export_history - экспорт в файл
- /history_info - информация и статистика
- /stop - завершить чат и очистить контексты

The screenshot shows a Telegram chat window with the following details:

- Bot Information:** ИИ спутник bot
- Message:** Translate to English
- Time:** 4:00 PM
- Message:** Chat with @chat activated
- Description:** Now all your messages will be sent to this agent.
- Section:** Возможности:
- List:** Same as the first section (Text, File, Upload, Stop).
- Section:** Работа с файлами:
- List:** Merge mode: file merging, max size 195 KB.
- Section:** Настройки истории:
- List:** History messages: Enabled (max 20 messages), files are deleted during history clearing.
- Section:** Команды управления:
- List:** /clear_history - clear history and files, /export_history - export to file, /history_info - info and stats, /agent_stop - stop chat and clear contexts.
- Message:** напиши код игры змейка на Си с использованием raylib
- Message:** напиши код игры змейка на Си с использованием га...
- File:** chat_response_20250926_160255.txt (6.7 KB)
- Message:** Ответ слишком большой и сохранен в файл
- Time:** 4:02 PM

Агент @chat - Управление историей чата



Управление историей чата:

save [ID] - сохранить текущую историю

load [ID] - загрузить сохраненную историю

delete [ID] - удалить сохраненную историю

rename [старый] [новый] - переименовать историю

list - показать список сохраненных историй (до 50 чатов)

Примеры:

save project_v1 - сохранить как **project_v1**

load project_v1 - загрузить **project_v1**

rename project_v1 project_final



ID может содержать буквы, цифры, _ и -

(*) - Функция автоматического сохранения истории чата отсутствует; пользователи сами решают, когда сохранять историю чата. Это может быть полезно для ветвящихся историй, например, при проведении исследований, чтобы сохранять контрольные точки и только «успешные» ветви чата.



@macros Быстрые макро команды для файлов

Используйте после конвертации файла:

Обработка текста:

- @sum - Суммаризация текста
- @analyze - Детальный анализ содержимого
- @translate - Перевод на английский язык
- @translate_ru - Перевод на русский язык
- @keywords - Извлечение ключевых слов и тегов
- @questions - Генерация вопросов для проверки
- @outline - Создание структуры документа
- @improve - Улучшение и редактирование текста
- @explain - Объяснение простым языком
- @tasks - Извлечение задач и действий
- @entities - Извлечение сущностей



Обработка кода:

- @code - Анализ кода и документация
- @refactor - Рефакторинг кода
- @comment - Добавление комментариев к коду
- @tests - Генерация тестов для кода
- @security - Проверка безопасности кода
- @readme - Генерация README документации
- @api - Документация API
- @clean - Очистка и форматирование кода
- @extract - Извлечение функций/классов



@macros Быстрые макро команды для файлов



Возможности:

- Команды применяются к последнему загруженному файлу
- Можно добавить свой запрос после команды:

`@sum` с фокусом на технические детали

`@translate` на французский язык

- Результат сохраняется как новый файл
- К результату можно применить другую команду



Агент @zip

Что умеет:

 Автоматическая конвертация документов в текст:

- PDF → извлечение текста
- Word (.docx, .rtf, .odt) → markdown
- Excel (.xlsx, .xls) → таблицы в markdown
- PowerPoint (.pptx) → текст слайдов
- HTML → чистый текст

Как это работает:

1. Отправляете архив (.zip или .7z)
2. Агент распаковывает все файлы
3. Документы автоматически конвертируются в текст
4. Всё сохраняется в вашу кодовую базу

Особенности:

-  Сохранение русских букв в именах файлов
-  Все файлы в одной папке (плоская структура)
-  Имена приводятся к нижнему регистру
-  Пробелы заменяются на _

 Конвертация происходит локально, без использования AI и интернета.

Агент @rag (заточен на работу с русским языком)

Интеллектуальный поиск по кодовой базе с контекстными ответами

 Тип: Поиск по базе знаний

Использование:

@rag ваш вопрос - поиск ответа в базе

Примеры запросов:

- @rag как работает функция X?
- @rag найди примеры использования Y
- @rag объясни алгоритм Z

 Совет: Можете загрузить документ для анализа.

Просто отправьте текстовый файл после использования агента.

@rag config - скачать конфигурацию агента

Агент @filejob - Пакетная ИИ обработка файлов

Формат использования:

@filejob селектор_файлов<перенос_строки>
ИИ запрос для применения к каждому файлу

Примеры селекторов:

- * - все файлы
- .py** - все Python файлы
- 1,3,5** - файлы с номерами 1, 3 и 5
- 2-10** - файлы с 2 по 10
- test_*.js** - файлы по маске
- main.py, utils.py** - конкретные файлы

Пример:

@filejob *.py
Найди все функции и опиши что они делают

Работа с аудио файлами

Размер файла максимум для Телеграм ботов: **20.0Mb**

Автоматически при получении распознает в текст в формате MarkDown.

При отправке большого файла выдает подсказку как можно уменьшить размер файла используя **Audacity** (<https://www.audacityteam.org/>) или онлайн конвертеры (без установки) или мобильные приложения.

Настройки **Audacity** экспорта для речи/подкастов:

- Format: Ogg Vorbis Files (лучшее сжатие)
- Channels: Mono (уменьшает размер вдвое)
- Sample Rate: 22050 Hz или 44100 Hz
- Quality: 0-2 (для речи достаточно)

С такими настройками вы сможете обработать до 2 часов аудиозаписи и получить для вашей конференции полную транскрипцию в текст и например через макро команду @sum сразу получить суммаризацию всей записи. Либо используйте для больших файлов Агент @upload



Агент @upload - Загрузчик файлов

Использование:

@upload URL - скачать и обработать файл по публичной ссылке

Примеры:

@upload <https://example.com/document.pdf>

@upload <https://raw.githubusercontent.com/user/repo/main/file.py>

@upload https://drive.google.com/file/d/FILE_ID/view

@upload <https://disk.yandex.ru/d/XXXXXX>

Поддерживаемые типы:

- PDF, Word, Excel, PowerPoint
- Аудио файлы (транскрипция)
- Изображения (OCR)
- Текстовые файлы и код
- HTML, JSON и XML документы
- Архивы (zip, 7z) — обрабатываются автоматически

Особенности:

- Автоматическое определение типа файла
- Поддержка загрузки по публичным ссылкам (Google Drive, Яндекс.Диск, Dropbox, GitHub)
- Конвертация документов в Markdown
- Транскрипция аудио в текст
- OCR распознавание текста на изображениях
- Распаковка архивов и сохранение содержимого
- Сохранение файлов в активную кодовую базу
- Автоматическая индексация для RAG

Ограничения:

- Максимальный размер: 100 МБ
- Протоколы: HTTP, HTTPS
- Таймаут: 60 секунд

Основной функционал бот-рассылок пользователям

Типы рассылок

- **Одноразовая** — отправка в конкретную дату и время
- **Периодическая** — повторяющаяся рассылка:
 - Ежемесячно (в указанный день месяца)
 - Интервальная (через N секунд)

Учёт рабочих дней

- Автоматический пропуск выходных (суббота, воскресенье)
- Пропуск праздничных дней РФ (библиотека `holidays`)
- Перенос на ближайший рабочий день **в пределах того же месяца**:
 - Сначала попытка переноса вперёд
 - Если выходит за границы месяца — перенос назад
 - Направление переноса логируется

Группы получателей

- Создание групп с ID и названием
- Добавление пользователей:
 - Поступчно из списка активных
 - Массовый импорт через распознавание @username в тексте
- Рассылка идёт только активным пользователям (`active: true` из `users.json`)
- **User IDs не хранятся** в рассылке — только имена групп

Управление

- Создание через пошаговый диалог (тип → группы → текст → ссылка → дата/период)
- Просмотр с отображением переносов на рабочие дни
- Удаление через `/delete_<ID>`
- Статистика отправок

Выполнение

- Проверка каждые 60 секунд
- Логирование в `spam_execution.log`
- Учёт блокировок бота (`TelegramForbiddenError` → `active: false`)
- Задержка 3 сек между отправками

Планы на будущее развитие ИИ Спутник

- Добавление новых агентов по запросам компаний
Ближайший план ИИ обработка баз резюме используя интеграцию с <https://huntflow.ru/api> (сортировка/суммаризация/поиск и отбор резюме по запросам HR)
- Добавление и поддержка новых провайдеров знания для @nethack
- Разработка специализированных агентов для мозговых штурмов с возможностью задать все роли участников прописав каждый ИИ агент отдельно
- Интеграция с [teamwork.com](#) для быстрой постановки задач

Контакты

Петров Андрей Михайлович

Генеральный директор

ООО "Локальные Интеллектуальные Системы"

ОГРН: 1243900008626 / ИНН: 3900028438

Контакты

E-mail: biz@d00m4ace.com

TenChat: tenchat.ru/d00m4ace

WEB: ЛОКАЛЬНЫЕ-ИНТЕЛЛЕКТУАЛЬНЫЕ-СИСТЕМЫ.РФ

BLOG: d00m4ace.com