



Geniiium Agent

Автономный самообучающийся ИИ-агент для бизнеса

Руководство пользователя и администратора

Версия документа 1.0 · 2026

Платформа: панель управления, мультиканальный шлюз, память и самообучение

Развёртывание: собственный сервер (self-hosted), Docker, доступ по HTTPS с cookie-авторизацией

Содержание

1. Введение	3
2. Преимущества и новизна технологии	4
3. Архитектура платформы	5
4. Быстрый старт (для пользователя)	6
5. Модели ИИ и провайдеры	7
6. Память и самообучение	8
7. Навыки (Skills)	10
8. Инструменты и наборы	11
9. Автоматизации и делегирование	12
10. Каналы связи	13
11. Интеграции	14
12. Панель управления	15
13. Безопасность	16
14. Развёртывание и эксплуатация	17
15. Справочник администратора	18
16. Частые вопросы	20

1. Введение

Geniiium Agent — программная платформа автономного искусственного интеллекта, которая по заданию пользователя самостоятельно планирует и выполняет многошаговые задачи. В отличие от обычного чат-бота «вопрос — ответ» или узкого помощника-копилота, Geniiium работает как настоящий агент: рассуждает, вызывает инструменты (терминал, файлы, интернет, внешние сервисы), проверяет результат и доводит задачу до конца, а накопленный опыт использует в следующих задачах.

Ключевое свойство платформы — она становится тем полезнее, чем дольше работает. Geniiium запоминает ваши предпочтения, проекты и окружение, обучается на собственном опыте, самостоятельно формирует повторно используемые «навыки» и выстраивает всё более точную модель пользователя от сессии к сессии. Всё это — без дообучения нейросетевой модели, программными средствами.

Платформа разворачивается в собственном контуре организации (self-hosted): на сервере компании, включая серверы архитектуры ARM64, в контейнере Docker. Управление — через русскоязычную веб-панель с доступом по защищённому URL (HTTPS, вход по логину/паролю, cookie-сессия). Данные, история и ключи доступа остаются на вашем сервере.

1.1. Для кого это руководство

- пользователей, которые работают с агентом через веб-панель и мессенджеры;
- администраторов, которые разворачивают, настраивают и сопровождают платформу.

Разделы 1–7 адресованы прежде всего пользователям, разделы 8–15 — администраторам. Технические идентификаторы (команды, переменные окружения, ключи конфигурации) приводятся в справочных разделах и в приложении.

1.2. Что умеет Geniiium — кратко

- **Агент.** Ведёт диалог и выполняет задачи с использованием больших языковых моделей любого поставщика.
- **Память.** Многоуровневая память и самообучение между сессиями (раздел 6).
- **Каналы.** Один процесс обслуживает Telegram, WhatsApp, Slack, Discord, e-mail и ещё 20+ платформ (раздел 10).
- **Навыки.** Библиотека компетенций, которая пополняется автоматически из опыта (раздел 7).
- **Автоматизации.** Задачи по расписанию с доставкой результата в любой канал (раздел 9).
- **Интеграции.** Подключение внешних инструментов по открытому протоколу MCP, веб-поиск, голос (раздел 11).
- **Панель.** Веб-панель с русским интерфейсом, темами и расширением плагинами (раздел 12).
- **Безопасность.** Семиуровневая модель защиты, изоляция исполнения, разграничение доступа (раздел 13).

2. Преимущества и новизна технологии

Geniium построен вокруг идеи замкнутого контура обучения (learning loop): агент не просто отвечает, а превращает каждый завершённый эпизод работы в долговременный актив — запись в памяти или новый навык. Именно это отличает платформу от диалоговых ассистентов и определяет её новизну.

2.1. Новизна

- **Замкнутый контур самообучения:** платформа программно реализует цикл «опыт → отбор → сохранение → повторное использование → уточнение», работающий поверх любой языковой модели и без её дообучения.
- **Курируемая память:** наряду с оперативной памятью агент по завершении работы сам решает, что достойно сохранения, и фиксирует это компактно и структурированно.
- **Автогенерация навыков:** успешно решённая задача обобщается в переиспользуемую процедуру; библиотека компетенций растёт сама, а фоновой «куратор» не даёт ей засоряться (раздел 7).
- **Поиск по прошлому:** полнотекстовый поиск по всей истории диалогов позволяет агенту вспоминать детали разговоров многомесячной давности мгновенно и без затрат на модель.
- **Мультиканальность из одного процесса:** один агент одновременно присутствует в веб-панели, мессенджерах и почте — с единой памятью и настройками.

2.2. Преимущества для организации

Преимущество	Что это даёт
Рост компетенций	Качество выполнения повторяющихся задач повышается со временем без переобучения моделей и без участия разработчика.
Суверенитет данных	Развертывание в контуре заказчика; история, память и ключи хранятся только на вашем сервере, телеметрия наружу не отправляется.
Независимость от поставщика	Единый интерфейс к моделям; смена поставщика или модели выполняется настройкой, без изменения кода и потери данных.
Мультиканальность	Один агент обслуживает мессенджеры, почту и веб-интерфейс одновременно.
Русский интерфейс	Панель, страница входа и документация — на русском языке.
Готовность к эксплуатации	Docker, поддержка ARM64, TLS-шифрование, разграничение доступа, журналирование, обновление из репозитория.

3. Архитектура платформы

Geniiium — модульная платформа. Компоненты работают совместно, но развиваются и настраиваются независимо.

- **Агентное ядро.** управляющий цикл: приём задачи, декомпозиция, вызовы инструментов, контроль результата, формирование ответа. Поддерживает подагентов для параллельных подзадач и программные вызовы инструментов из кода.
- **Шлюз коммуникаций.** единый фоновый процесс, обслуживающий все каналы связи (Telegram, WhatsApp, Slack, Discord, e-mail и др.), сессии, планировщик и голос.
- **Панель управления.** веб-приложение администратора: сессии, модели, ключи, навыки, каналы, планировщик, журналы, аналитика, встроенный терминал чата. Доступ по HTTPS с cookie-авторизацией.
- **Среды исполнения.** выполнение инструментов в изолированных окружениях: локально, в контейнере Docker, на удалённом узле по SSH, а также в бессерверных средах.
- **Планировщик.** регулярные задачи по cron-расписанию с доставкой результата в выбранный канал.
- **Подсистема расширений.** подключаемые модули интерфейса и серверной части, каталог навыков, клиент протокола MCP.
- **Слой моделей.** унифицированный интерфейс к поставщикам моделей; выбор основной и вспомогательных моделей.
- **Хранилище.** хранилище истории и состояния (SQLite с полнотекстовым индексом) и файлы памяти агента.

Важно. Каталог данных платформы (история, память, навыки, конфигурация, ключи) располагается на сервере и не затрагивается при обновлении образа — настройки и накопленный опыт сохраняются.

4. Быстрый старт (для пользователя)

4.1. Вход в панель

Откройте адрес панели, выданный администратором (вида `https://ваш-домен`). Появится страница входа Geniiium. Введите логин и пароль. После успешного входа устанавливается защищённая cookie-сессия (по умолчанию 12 часов с автоматическим продлением), и открывается панель управления на русском языке.

4.2. Подготовка к работе — три шага

- **Ключи.** на вкладке «Ключи» добавьте API-ключ выбранного поставщика моделей (например, OpenRouter). Ключ хранится только на вашем сервере.
- **Модель.** на вкладке «Модели» кнопкой «Сменить» выберите основную модель.
- **Чат.** перейдите на вкладку «Чат» и начните диалог. Агент выполняет задачи, использует инструменты и запоминает контекст.

Важно. Минимальная длина контекста модели — 64 000 токенов. Для локальных моделей задайте не меньше этого значения, иначе агент будет работать нестабильно.

4.3. Первый диалог

Сформулируйте задачу обычным языком — агент сам решит, какие инструменты применить. Признаки корректной работы: агент отвечает без ошибок, при необходимости обращается к инструментам (поиск, файлы, терминал) и поддерживает несколько ходов диалога подряд. История сохраняется автоматически: к прошлому разговору можно вернуться на вкладке «Сессии».

5. Модели ИИ и провайдеры

Geniium не привязан к одному поставщику. Платформа предоставляет единый интерфейс к десяткам провайдеров больших языковых моделей; переключение выполняется настройкой, без изменения кода. Различают два вида моделей: основная (ведёт диалог) и вспомогательные (обслуживают побочные задачи).

5.1. Поддерживаемые поставщики (выборка)

OpenRouter (маршрутизация к сотням моделей), OpenAI, Anthropic (Claude), Google Gemini / Vertex AI, AWS Bedrock, Azure AI Foundry, GitHub Copilot, xAI (Grok), DeepSeek, Mistral, Hugging Face, Alibaba (Qwen), MiniMax, Moonshot (Kimi), NVIDIA NIM, а также локальные и самостоятельно размещённые модели через OpenAI-совместимый интерфейс (Ollama, vLLM, SGLang, llama.cpp, LM Studio) и произвольные пользовательские эндпоинты. Ключ поставщика задаётся на вкладке «Ключи».

5.2. Вспомогательные модели

Побочные задачи можно (для экономии) выполнять на более дешёвой модели, отличной от основной. Отдельные слоты назначаются на вкладке «Модели»:

Слот	Назначение
Vision	Анализ изображений и скриншотов.
Web extract	Извлечение и резюмирование содержимого веб-страниц.
Compression	Сжатие длинной истории диалога при переполнении контекста.
Curator	Фоновой пересмотр библиотеки навыков (раздел 7).
Title / Approval / прочие	Генерация заголовков сессий, оценка запросов на подтверждение и др.

Значение «Авто» для слота означает «использовать основную модель». По умолчанию все вспомогательные задачи идут на основную модель.

5.3. Кэширование запросов

Для совместимых моделей семейства Claude (напрямую, через OpenRouter и др.) действует встроенное кросс-сессионное кэширование префикса промпта с интервалом до 1 часа. Оно работает автоматически и снижает стоимость и задержку при повторных обращениях; настройка не требуется.

6. Память и самообучение

Память — центральная подсистема Geniium и главный источник его новизны. Она устроена как несколько взаимодополняющих уровней и образует замкнутый контур накопления опыта, работающий программно, без дообучения нейросетевой модели.

6.1. Оперативная память: заметки агента и профиль пользователя

Ограниченная, курируемая память сохраняется между сессиями в двух файлах и в начале каждой сессии включается в системный промпт модели как зафиксированный снимок:

Хранилище	Назначение	Лимит
Заметки агента (MEMORY)	Факты об окружении, соглашения проекта, выявленные особенности и уроки, дневник выполненных задач.	≈2 200 симв. (~800 токенов)
Профиль пользователя (USER)	Имя, роль, часовой пояс, стиль общения, предпочтения и то, чего следует избегать.	≈1 375 симв. (~500 токенов)

Агент управляет памятью самостоятельно с помощью специального инструмента: добавляет, заменяет и удаляет записи. Отдельного действия «прочитать» нет — память и так присутствует в контексте с начала сессии. Замена и удаление выполняются по короткой уникальной подстроке, а не по полному тексту записи.

Важно. Строгие лимиты удерживают память компактной и всегда «в контексте». Память не уплотняется молча: если запись не помещается, инструмент возвращает ошибку, и агент сам консолидирует или удаляет устаревшее в том же ходе. Точные дубликаты отклоняются автоматически.

«Замороженный снимок»: содержимое памяти захватывается один раз в начале сессии и не меняется до её конца (это сохраняет кэш префикса модели). Изменения памяти по ходу сессии немедленно пишутся на диск, но в системном промпте отражаются со следующей сессии.

Что сохраняется, а что нет

Агент сохраняет проактивно, без напоминаний: устойчивые предпочтения пользователя, факты об окружении (ОС, инструменты, структура проекта), исправления и соглашения, сведения о завершённой работе, явные просьбы «запомни...». Пропускает: тривиальное и очевидное, легко находимое повторно, крупные «сырые» данные, разовые пути и эфемерный контекст сессии.

Управление записью памяти

По умолчанию агент сохраняет записи свободно. Если требуется предварительное подтверждение, включается режим апробации: тогда автоматические сохранения (в том числе из фонового самопересмотра) не попадают в профиль, пока вы не подтвердите их через раздел «ожидающие записи». Это ответ на ситуацию «агент сохранил обо мне неверное предположение».

6.2. Поиск по прошлым разговорам

Помимо оперативной памяти агент умеет искать по всей истории диалогов. Все сессии (и в панели, и в мессенджерах) хранятся в базе данных SQLite с полнотекстовым индексом (FTS5). Поиск возвращает реальные сообщения из базы — без резюмирования и усечения; агент может найти обсуждавшееся недели назад и «прокручивать» найденную сессию вперёд и назад.

Свойство	Оперативная память	Поиск по сессиям
Объём	≈1 300 токенов всего	Не ограничен (все сессии)
Скорость	Мгновенно (в промпте)	≈20 мс запрос, ≈1 мс прокрутка
Стоимость	Токены в каждом промпте	Бесплатно — без обращений к модели
Назначение	Ключевые факты всегда под рукой	«Обсуждали ли мы X на прошлой неделе?»
Ведение	Курируется агентом вручную	Автоматически — хранятся все сессии

6.3. Замкнутый контур самообучения

После завершения хода работы в фоне запускается пересмотр (background review): повторное «проигрывание» разговора, в ходе которого повторяющиеся исправления и устойчивые рабочие уроки превращаются в компактные записи памяти или в процедурные навыки. Так реализуется обучение с учётом согласия пользователя: режим апробации может ставить такие записи на подтверждение до того, как они повлияют на будущие сессии.

Пересмотр по умолчанию выполняется на основной модели (используя уже «тёплый» кэш промпта, поэтому дёшево), но при дорогой основной модели его можно перенести на более дешёвую — тогда в неё проигрывается компактная выжимка разговора. В панели по завершении хода выводится короткая отметка вида «Память обновлена», чтобы изменения были прозрачны; степень подробности этих уведомлений настраивается.

6.4. Куратор: сопровождение навыков

Навыки, созданные агентом в ходе самообучения (раздел 7), нуждаются в уходе, иначе библиотека засоряется узкими почти-дубликатами. Этим занимается фоновой «куратор» — обслуживающий проход, который:

- отслеживает частоту просмотра, применения и правок каждого навыка;
- переводит давно не используемые навыки по состояниям «активный → устаревший → архив» (по умолчанию через 30 и 90 дней бездействия);
- периодически (опционально) запускает пересмотр вспомогательной моделью, который предлагает объединять пересекающиеся навыки под «зонтичные» и устранять дрейф.

Важно. Куратор никогда не удаляет навыки безвозвратно — худший исход — перенос в архив, откуда навык восстанавливается. Перед каждым реальным проходом создаётся резервный снимок библиотеки; любой проход можно откатить одной командой. Установленные из каталога навыки куратор не трогает; отдельные навыки можно «закрепить», защитив от любых автопереходов.

6.5. Внешние провайдеры памяти

Для более глубокой персонализации помимо встроенной памяти подключаются внешние провайдеры (Honcho, OpenViking, Mem0, Hindsight, Holographic, RetainDB, ByteRover, Supermemory). Они работают рядом со встроенной памятью, не заменяя её, и добавляют графы знаний, семантический поиск, автоматическое извлечение фактов и кросс-сессионное моделирование пользователя. Подключение выполняется администратором.

7. Навыки (Skills)

Навык — это документ знаний, который агент подгружает по мере надобности (принцип прогрессивного раскрытия: сначала — краткий список навыков, затем — тело конкретного навыка, затем — его вложенные файлы). Это экономит контекст. Каждый установленный навык доступен и как команда в чате.

7.1. Использование и установка

- просмотр и поиск установленных навыков — в панели на вкладке «Навыки»;
- каталог (Hub) навыков открытого стандарта agentskills.io — установка в один клик;
- навык можно вызвать явно (как команду) или естественным языком — агент подберёт подходящий сам.

7.2. Самосоздание и жизненный цикл

Ключевая способность: когда агент успешно решает новую задачу, он может обобщить решение в новый навык. Такие навыки помечаются как «созданные агентом» и попадают под сопровождение куратора (раздел 6.4), проходя жизненный цикл «активный → устаревший → архив». Ваши собственные и установленные из каталога навыки куратор не трогает.

Важно. Навыки и память решают разные задачи. Память — это компактные факты, всегда присутствующие в контексте. Навык — это процедура (как что-то делать), подгружаемая по требованию. Вместе они дают и «что я знаю о тебе», и «как мне это сделать».

8. Инструменты и наборы

Инструменты — это функции, расширяющие возможности агента; они сгруппированы в логические наборы (toolsets), которые можно включать и отключать по каждой платформе. Всего доступно свыше 60 встроенных инструментов. Основные группы:

- **Веб.** поиск в интернете и извлечение содержимого страниц (несколько поисковых движков на выбор).
- **Терминал.** выполнение команд в изолированном окружении (локально, Docker, SSH и др.).
- **Файлы.** чтение, запись и точечное редактирование файлов; снимки рабочей директории и откат изменений.
- **Память.** управление заметками агента и профилем пользователя (раздел 6).
- **Делегирование.** запуск подагентов с изолированным контекстом для параллельных подзадач (раздел 9).
- **Исполнение кода.** агент пишет короткий сценарий, который вызывает инструменты платформы напрямую, сворачивая многошаговый процесс в один ход.
- **Браузер.** полноценная автоматизация браузера (облачные и локальные движки): навигация, заполнение форм, извлечение данных.
- **Мультимодальность.** генерация изображений, синтез и распознавание речи (голосовой режим).

Важно. Опасные действия проходят через подтверждение, а исполнение по умолчанию изолировано (раздел 13). Набор инструментов, доступных агенту, ограничивается администратором.

9. Автоматизации и делегирование

9.1. Задачи по расписанию (Cron)

Планировщик выполняет задачи автоматически по расписанию, заданному естественным языком или cron-выражением. Задача может подключать навыки, доставлять результат в любой канал связи и поддерживает операции паузы/возобновления/редактирования. Управление — в панели на вкладке «Cron». Примеры применения: утренние сводки, регулярные отчёты, мониторинг и напоминания.

Платформа также предлагает «заготовки» (blueprints) и подсказки автоматизаций: при установке некоторых навыков предлагается (с вашего согласия) зарегистрировать соответствующее регулярное задание.

9.2. Делегирование подагентам

Инструмент делегирования порождает дочерние экземпляры агента с изолированным контекстом, ограниченным набором инструментов и собственными терминальными сессиями. По умолчанию параллельно работает до трёх подагентов (настраивается), что ускоряет независимые направления работы. Это же лежит в основе исполнения кода и параллельных рабочих процессов.

10. Каналы связи

Мультиканальный шлюз — единый фоновый процесс, который подключает все платформы сразу и обслуживает сессии, планировщик и голос. Агент отвечает там, где вам удобно: в мессенджерах, почте и веб-чате — с общей памятью и настройками.

10.1. Безопасность доступа: «запрет по умолчанию»

Шлюз по умолчанию отклоняет всех пользователей, которых нет в списке разрешённых или которые не прошли сопряжение в личных сообщениях. Разрешённые пользователи задаются для каждой платформы. Сопряжение выполняется по одноразовому коду (срок действия — 1 час, с защитой от перебора). Это исключает несанкционированный доступ к агенту через мессенджер.

10.2. Поддерживаемые платформы

Более 20 платформ из одного шлюза. Для каждой требуется токен/учётные данные соответствующего сервиса; настройка выполняется администратором (мастер настройки шлюза) и хранится в конфигурации сервера.

Платформа	Что нужно для подключения
Telegram	Токен бота (@BotFather) и список разрешённых пользователей; опционально webhook-режим.
WhatsApp	Личный номер через мост (QR-привязка) или официальный WhatsApp Business Cloud API (токены Meta).
Slack	Токены бота и приложения (Socket Mode); генерация манифеста приложения.
Discord	Токен бота, включённые Privileged Intents, список пользователей/ролей; навыки как слэш-команды.
Электронная почта	Адрес, пароль приложения, серверы IMAP/SMTP; список разрешённых отправителей.
Signal	Локальный демон signal-cli (привязка как второе устройство) и номер аккаунта.
Microsoft Teams	Регистрация приложения Azure AD (client/secret/tenant).
Google Chat, Matrix, Mattermost	Учётные данные соответствующих сервисов (Pub/Sub, токены доступа).
SMS (Twilio), LINE, ntfy, IRC	Ключи API/токены сервисов; для входящих — публичный webhook.
DingTalk, Feishu/Lark, WeCom, Weixin, QQ	Идентификаторы и секреты приложений соответствующих экосистем.
Home Assistant	Долгоживущий токен и адрес — добавляет инструменты умного дома.
Веб-панель, API-сервер	Встроенный чат панели и OpenAI-совместимый HTTP-эндпоинт (раздел 11).

11. Интеграции

11.1. Протокол MCP (внешние инструменты)

Geniium — клиент открытого протокола Model Context Protocol (MCP). Через него агент подключается к любому MCP-серверу (по stdio или HTTP) и получает доступ к внешним инструментам — GitHub, базам данных, файловым системам, внутренним API — без написания собственных инструментов. Поддерживаются фильтрация инструментов по серверам и sampling. Серверы MCP добавляются в конфигурацию платформы.

11.2. Веб-поиск, голос, изображения

- **Веб-поиск:** несколько движков на выбор (Firecrawl, SearXNG, Brave, DuckDuckGo, Tavily, Exa, Parallel, xAI).
- **Голос:** синтез речи (Edge TTS бесплатно, ElevenLabs, OpenAI, MiniMax, Gemini и др.) и распознавание речи (локально или через облачные сервисы); голосовой режим в панели и мессенджерах.
- **Изображения:** генерация изображений из текста через FAL.ai (несколько моделей на выбор).

11.3. OpenAI-совместимый API-сервер

Платформу можно выставить как OpenAI-совместимый HTTP-эндпоинт и подключить к нему любой совместимый интерфейс (Open WebUI, LobeChat, LibreChat и др.). Сервер требует ключа-предъявителя (Bearer) на каждом развёртывании, включая локальное. Предоставляются эндпоинты чата, запусков (runs), задач (jobs), сессий и обнаружения навыков/наборов инструментов.

12. Панель управления

Веб-панель — основной интерфейс администрирования и работы. Русскоязычная, с фирменным оформлением Geniiium (тёмная и светлая темы). Доступ — по HTTPS с cookie-авторизацией. Разделы панели:

- Чат — диалог с агентом и встроенный терминал; Сессии — история с полнотекстовым поиском.
- Модели и Ключи — выбор моделей и хранение ключей поставщиков.
- Навыки, Инструменты (наборы), Плагины — управление возможностями агента.
- Каналы и Сопряжение — подключение мессенджеров и допуск пользователей.
- Срон — задачи по расписанию; Журналы и Аналитика — наблюдаемость.
- Документация — этот справочник; Конфигурация — параметры платформы.

12.1. Расширение панели

Панель расширяется тремя способами без правки исходного кода, что и используется для оформления Geniiium:

- **Темы:** YAML-описание палитры, шрифтов, радиусов и стилей (в том числе фирменная тема Geniiium).
- **UI-плагины:** подключаемые модули интерфейса (собственные вкладки, страницы, встраиваемые блоки).
- **Бэкенд-плагины:** серверные модули на FastAPI, добавляющие собственные HTTP-эндпоинты панели.

13. Безопасность

Geniiium рассчитан на одного владельца (single-tenant, персональный агент). Модель безопасности многоуровневая, а её основной принцип формулируется прямо: единственная граница безопасности против недоверенной языковой модели — это операционная система. Поэтому изоляция исполнения инструментов первична, а внутрипроцессные проверки — вспомогательны.

13.1. Семь уровней защиты

- **Авторизация:** «запрет по умолчанию», списки разрешённых пользователей, сопряжение по коду с защитой от перебора.
- **Подтверждения:** опасные команды требуют подтверждения; режимы «ручной / умный / выкл».
- **Изоляция:** исполнение в контейнере со сброшенными привилегиями (cap-drop, no-new-privileges, лимиты процессов и ресурсов) либо на выделенном узле по SSH.
- **Фильтрация учётных данных:** во внешние инструменты и MCP передаётся минимум переменных окружения; секреты редактируются в выводе.
- **Сканирование контекста:** проверка контекстных файлов проекта на попытки инъекций перед загрузкой.
- **Изоляция сессий:** сессии и данные разных каналов и пользователей разделены.
- **Санитаризация:** защита от обращения к внутренним адресам (SSRF: блокировка приватных/loopback/метаданных облака), чёрные списки сайтов, предзапусковый сканер команд.

13.2. Защищённый удалённый доступ

Панель на публичном интерфейсе включает шлюз аутентификации: без действительной сессионной cookie доступна только страница входа, а при отсутствии настроенного провайдера аутентификации сервер сознательно отказывается запускаться (fail-closed). Поддерживаются вход по логину/паролю (за TLS), а также внешние OIDC-провайдеры (Keycloak, Authentik и др.). Cookie-сессии — HttpOnly, Secure (по HTTPS), с ограниченным сроком и защитой от перебора пароля.

Внимание. Рекомендации по развёртыванию: работать не под root, ограничить права файлов с секретами, не выставлять шлюз и API в интернет без TLS и списков доступа, задавать список разрешённых пользователей для каждого канала, проверять сторонние навыки и плагины перед установкой.

14. Развёртывание и эксплуатация

14.1. Развёртывание в Docker

Рекомендуемый способ — контейнер Docker (образ мультиархитектурный, поддерживается ARM64). Каталог данных платформы монтируется с хоста, поэтому история, память, навыки, конфигурация и ключи сохраняются между пересборками образа. Доступ извне — через обратный прокси с TLS-терминацией; наружу открываются только порты 80/443.

14.2. Обновление

Обновление сервера в контейнерном развёртывании выполняется пересборкой образа из репозитория (получение новой версии кода и повторная сборка), после чего контейнеры перезапускаются. Пользовательские данные при этом не затрагиваются. Перед публикацией версии проводится smoke-проверка брендинга (страница входа, отсутствие посторонних упоминаний, работа авторизации).

14.3. Резервное копирование и профили

Поддерживаются экспорт и импорт полного состояния платформы, а также профили — независимые конфигурации (например, для разных ботов или окружений) со своими настройками и данными.

15. Справочник администратора

Важно. В этом разделе и в приложении технические идентификаторы (команды, переменные окружения, ключи конфигурации, пути) приведены в исходном виде и не выносятся в пользовательский интерфейс продукта. Их переименование нарушает работу системы.

15.1. Административный интерфейс командной строки

Внутри контейнера доступна утилита командной строки (далее — `geniium`). Основные группы команд:

Команда	Назначение
<code>geniium model</code>	Выбор основной и вспомогательных моделей (мастер настройки).
<code>geniium gateway</code>	Управление шлюзом каналов: <code>run</code> / <code>start</code> / <code>stop</code> / <code>status</code> / <code>setup</code> / <code>install</code> .
<code>geniium dashboard</code>	Запуск веб-панели; <code>serve</code> — <code>headless</code> -режим для удалённого доступа.
<code>geniium tools / skills</code>	Управление наборами инструментов и навыками (<code>browse/install/list/...</code>).
<code>geniium cron</code>	Задачи по расписанию: <code>list</code> / <code>create</code> / <code>edit</code> / <code>pause</code> / <code>resume</code> / <code>run</code> .
<code>geniium memory / curator</code>	Внешние провайдеры памяти; сопровождение библиотеки навыков.
<code>geniium mcp / plugins</code>	Подключение MCP-серверов и модулей расширения.
<code>geniium config</code>	Просмотр и изменение конфигурации: <code>show</code> / <code>edit</code> / <code>set</code> / <code>check</code> / <code>migrate</code> .
<code>geniium pairing</code>	Сопряжение пользователей мессенджеров: <code>approve</code> / <code>list</code> / <code>revoke</code> .
<code>geniium doctor / logs / backup</code>	Диагностика, журналы, резервное копирование.
<code>geniium update</code>	Обновление платформы (в контейнере — пересборкой образа).

Команда одноразового запуска (для сценариев) возвращает только итоговый текст ответа; в панели и мессенджерах доступен параллельный набор слэш-команд управления диалогом.

15.2. Частые команды в чате (слэш-команды)

Команда	Действие
<code>/new</code> (<code>/reset</code>)	Начать новую сессию (сбросить контекст).
<code>/model</code> [модель]	Сменить модель для текущего диалога.
<code>/personality</code> [имя]	Переключить стиль общения (личность).
<code>/retry</code> , <code>/undo</code>	Повторить последний ход / отменить последний.
<code>/compress</code>	Сжать историю при переполнении контекста.
<code>/status</code> , <code>/usage</code>	Состояние сессии; разбивка по токенам и стоимости.
<code>/memory</code> , <code>/skills</code>	Управление ожидающими записями памяти и навыков.
<code>/cron</code> , <code>/background</code>	Задачи по расписанию; запуск фоновой задачи.

Команда	Действие
/rollback	Откат изменений файлов к снимку (checkpoint).
/help	Список команд.

16. Частые вопросы

В. Где хранятся мои данные?

О. Только на вашем сервере: история, память, навыки и ключи поставщиков находятся в каталоге данных платформы. Телеметрия наружу не отправляется.

В. Чат не отвечает и показывает ошибку про поставщика модели.

О. Не задан ключ модели. Добавьте на вкладке «Ключи» ключ поставщика (например, OPENROUTER_API_KEY) и выберите модель на вкладке «Модели».

В. Можно ли работать полностью офлайн, на локальной модели?

О. Да. Подключите локальную модель (Ollama, vLLM, llama.cpp, LM Studio) через OpenAI-совместимый эндпоинт; задайте длину контекста не меньше 64 000 токенов.

В. Как сменить поставщика модели без потери данных?

О. Смена выполняется настройкой на вкладке «Модели»; история, память и навыки сохраняются — они не привязаны к конкретному поставщику.

В. Чем память отличается от навыков?

О. Память — это факты о вас и окружении, всегда присутствующие в контексте. Навык — это процедура «как сделать», подгружаемая по требованию (раздел 7).

В. Сколько пользователей может обслуживать агент?

О. Через шлюз агент обслуживает несколько пользователей и каналов одновременно; доступ каждого регулируется списками разрешённых и сопряжением.