Общество с ограниченной ответственностью
«СберМедИИ»
ИНН: 9731065465
Адрес электронной почты: contact@sbermed.ai

Документация, содержащая описание функциональных характеристик программного обеспечения и информацию, необходимую для установки и эксплуатации программного обеспечения

«Мультиагентная информационная система здравоохранения (МАИСЗ)»

Оглавление

Термины и сокращения....3
Назначение настоящего документа....4
Общее описание функциональных характеристик ПО....4
Уровень подготовки пользователей....4
Программные и аппаратные требования....5
Инструкция по установке и получению доступа к ПО....5
Инструкция по эксплуатации ПО....6
Завершение работы ПО....19
Аварийные ситуации....19

Термины и сокращения

Термин	Определение
МАИСЗ	Мультиагентная информационная система здравоохранения.
ПО	программное обеспечение «Мультиагентная информационная система здравоохранения».
ИСЗ	Информационная система здравоохранения.
МИС	Медицинская информационная система.
AI-агент	программный модуль, реализованный на базе технологий искусственного интеллекта, в том числе с использованием больших языковых моделей (LLM), выполняющий определенные функции по обработке запросов пользователя.
Виджет	встраиваемый элемент пользовательского интерфейса, обеспечивающий интеграцию ПО с МИС или внешними веб-интерфейсами.
LLM (Large Language Model)	большая языковая модель, тип нейросети, обученной на больших объемах текстовых данных и способной понимать, генерировать и преобразовывать естественный язык.
On-premise	модель поставки и развертывания ПО, при которой программное обеспечение устанавливается на серверном оборудовании заказчика.
API (Application Programming Interface)	программный интерфейс приложения, обеспечивающий взаимодействие между различными компонентами системы или внешними сервисами.
SSO (Single Sign-On)	система единой аутентификации, позволяющая пользователю использовать один набор учетных данных для доступа к нескольким независимым системам.
Docker-контейнер	форма распространения ПО, представляющая собой изолированную среду для запуска приложения со всеми его зависимостями.
Сессия диалога	непрерывный период взаимодействия пользователя с AI-агентом, объединенный единым контекстом общения.
Визуальный редактор для создания агентов	Настраиваемая платформа с открытым исходным кодом на основе Python для создания ИИ-приложений.
Узлы	Отдельные функциональные блоки (ввод/вывод чата, языковые модели, базы данных, инструменты агента).

Граф	Описание логики работы ИИ-агента и функционирования узлов ИИ-сервиса
API-ключ	Пароль, который программы используют для безопасного общения друг с другом.
MCP	Универсальный «переходник», через который ИИ легко и безопасно подключается к вашим личным файлам, базам данных и другим программам.

Назначение настоящего документа

Руководство предназначено для специалистов, которые настраивают и проверяют рабочие процессы МАИСЗ.

В руководстве описаны порядок получения доступа, работа с визуальным редактором и проверка рабочих процессов.

Общее описание функциональных характеристик ПО

«Мультиагентная информационная система здравоохранения (МАИСЗ) » - ПО, предназначенное для создания, настройки и запуска ИИ-агентов и ИИ-сервисов в здравоохранении.

Рабочий процесс задаётся графом компонентов. В визуальном редакторе пользователь размещает компоненты на холсте, настраивает параметры и соединяет выходные порты с входными. Соединения определяют порядок передачи данных между компонентами.

МАИСЗ позволяет создавать процессы с нуля и загружать готовые шаблоны в формате JSON. В граф можно включать узлы ввода и вывода, языковые модели и инструменты агентов. Модель выбирается из настроенных провайдеров, а внешние инструменты подключаются через MCP.

Рабочие процессы выполняются в песочнице или вызываются внешними приложениями через API. При проверке можно просмотреть сообщения сессии и трассировки запусков.

В поставку МАИСЗ входит:

- Базовая версия компонента «Ассистент врача» - подключает к рабочему процессу сервис, который отвечает на медицинские вопросы специалистов и при необходимости обращается к клиническим рекомендациям. В данной версии не предусмотрены поиск и сопоставление сведений о лекарственных препаратах по инструкциям и фармацевтическим справочникам.
- Компонент «GigaChat Med» - подключает языковую модель GigaChat к рабочему процессу. При использовании GigaChat через МАИСЗ можно отключить медицинский фильтр для обработки профессиональных медицинских запросов

Уровень подготовки пользователей

Пользователь ПО должен обладать следующей квалификацией:

- Уверенное пользование операционной системой ЭВМ и веб-браузером.

- Изучение пользователем эксплуатационной документации на ПО.
- Соблюдение трудовой этики в рамках профессиональной деятельности.

Программные и аппаратные требования

Сервер приложений	
Процессор	1x8 core
Оперативная память	1x 16Гб
Дисковое пространство	1x100Гб
Количество виртуальных машин	1
Операционная система	семейства Unix
Иное установленное вспомогательное ПО (не является частью данного ПО)	Docker, Docker Compose V2

Инструкция по установке и получению доступа к ПО

Способ предоставления доступа зависит от выбранной модели поставки:

• SaaS (облачный сервис) - для авторизации используется SSO или выданные учетные данные. Установка ПО на ЭВМ пользователя не требуется.

• On-premise (локальное развертывание) - предоставление доступа осуществляется путем передачи дистрибутива (docker-контейнера с ПО) по электронным каналам связи и активации лицензионного ключа (электронного ключа в формате текстовой строки). Инсталляция на сервер заказчика выполняется инженером техподдержки вручную с использованием инструкции по развертыванию. Деинсталляция осуществляется вручную путем удаления контейнера с ПО.

Инструкция по локальному развертыванию ПО

Очистка системы от существующих образов

Выполните команду для очистки системы от старых образов docker

```
sudo docker system prune -a
Y
```

Распаковка дистрибутива

Распакуйте архив с программным обеспечением и дождитесь окончания выполнения команды:

```
sudo unzip AGENT.zip
```

И перейдите в директорию проекта

```
cd AGENT
```

Первичный запуск (инициализация окружения)

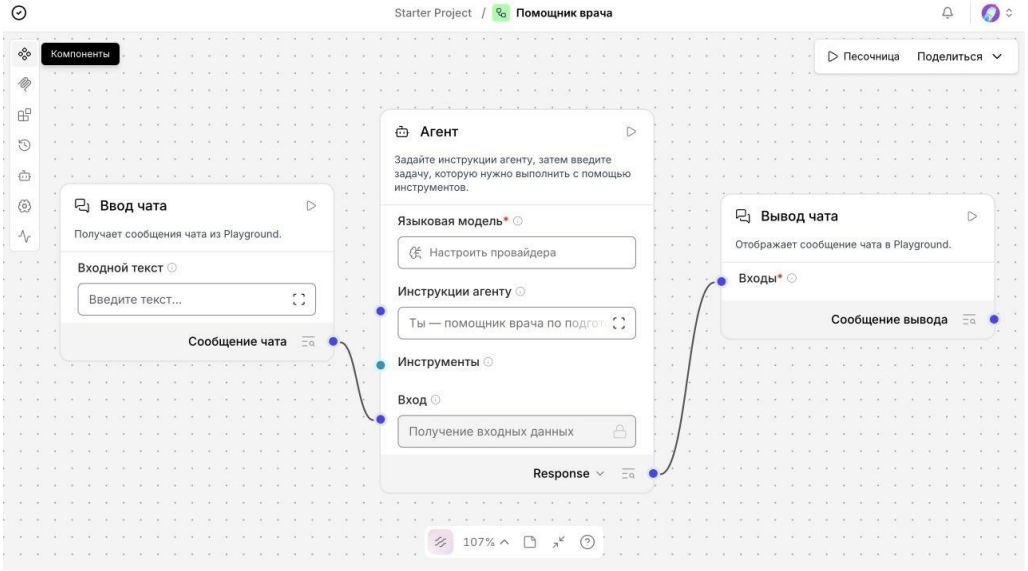
Запустите скрипт `start.sh` для загрузки Docker-образов и запуска контейнеров:

```
sudo chmod +x start.sh
sudo ./start.sh
```

Инструкция по эксплуатации ПО

Интерфейс редактора

Редактор открывается при выборе рабочего процесса. Холст занимает основную часть окна. На верхней панели отображаются проект и название процесса. Нажатие на название открывает свойства процесса.



Левая панель

- **Компоненты** - открывает каталог компонентов. Компоненты сгруппированы по назначению, их можно найти по названию и добавить на холст.
- **MCP** - открывает список серверов MCP. Здесь можно добавить сервер или перейти к управлению серверами.
- **Пакеты** - показывает компоненты, сгруппированные по интеграциям и наборам функций.
- **Версии** - открывает историю версий рабочего процесса и показывает текущую рабочую версию.
- **Агент** - открывает настройки публикации рабочего процесса как агента для вызова из внешних систем.
- **Память** - открывает раздел создания и просмотра баз памяти.
- **Трассировки** - показывает историю запусков процесса. Запуски можно сгруппировать по сессиям и отфильтровать по статусу или дате.

Правая часть верхней панели

- **Песочница** - открывает чат для проверки рабочего процесса. Здесь можно отправить сообщение и просмотреть ответ.
- **Поделиться** - открывает меню доступа к API, экспорта и встраивания чата. В меню есть ссылка на MCP-сервер проекта.

Нижняя панель холста

- **Встроенный ассистент открывает панель помощи в редакторе. Для его работы требуется настроенный провайдер модели.**
- **Управление масштабом холста** - показывает текущий масштаб. В меню можно увеличить или уменьшить масштаб, сбросить его до 100 % или вписать все компоненты в область просмотра.
- **Добавить стикер** - создаёт на холсте текстовую заметку.
- **Свернуть все** - переключает компактное и развёрнутое отображение компонентов на холсте.
- **Справка** - содержит ссылки на документацию и список горячих клавиш, а также переключатель умных направляющих для выравнивания компонентов.

Для просмотра схемы можно изменить масштаб и положение холста. Перетаскивание компонента за его заголовок меняет расположение компонента. Соединения при этом сохраняются.

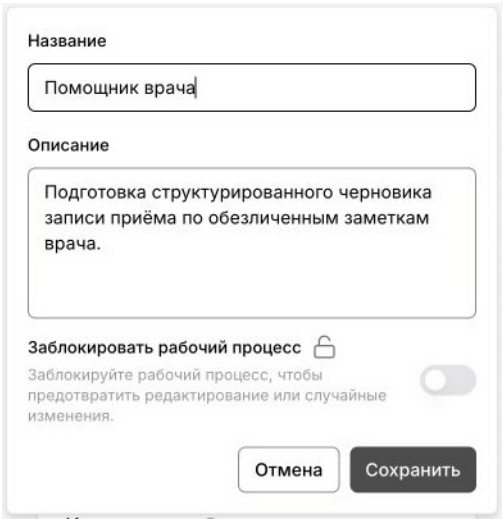
Рабочие процессы

Создание и открытие

Рабочий процесс открывается из списка процессов. Команда создания находится в том же списке. Если в системе ещё нет процессов, на стартовой странице отображается кнопка «Создать первый поток». В новом процессе может открыться окно выбора шаблона. После его закрытия доступен холст.

Свойства процесса

Для просмотра свойств нажмите название процесса в верхней части редактора. В открывшейся панели можно указать название и описание. Переключатель «Заблокировать рабочий процесс» ограничивает редактирование.

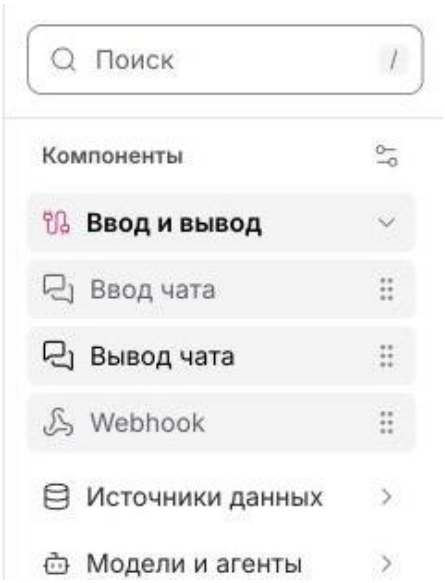


Изменения названия, описания и блокировки применяются кнопкой «Сохранить». Кнопка «Отмена» закрывает панель и отбрасывает изменения, внесённые в ней.

Работа с компонентами

Каталог компонентов

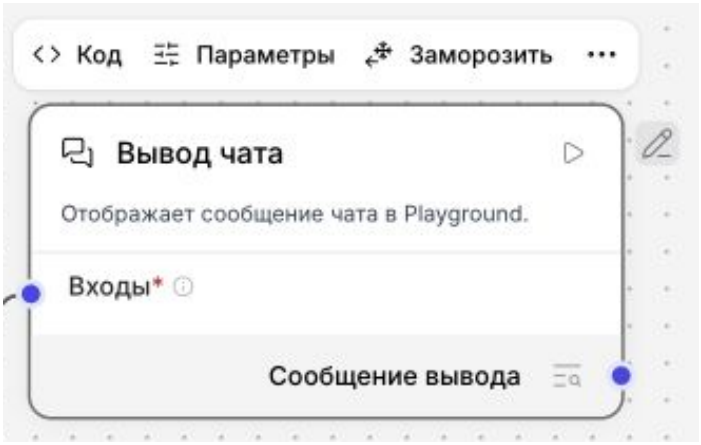
Каталог открывается слева по кнопке «Компоненты». Название компонента можно ввести в строку поиска. Компоненты «Ввод чата» и «Вывод чата» находятся в категории «Ввод и вывод», компонент «Агент» в категории «Модели и агенты».



Компонент добавляется на холст из каталога. Доступна кнопка добавления рядом с названием. Компонент также можно перетащить на холст. Для перемещения уже добавленного компонента используется его заголовок.

Компонент на холсте

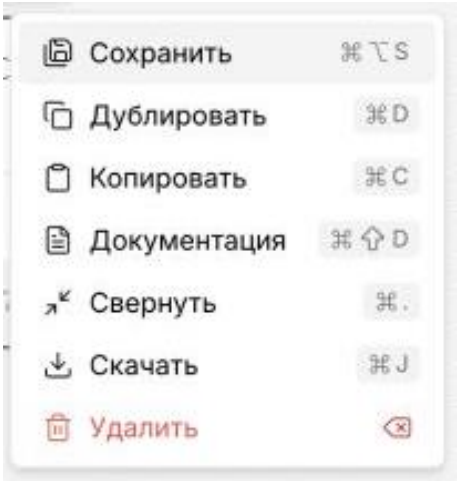
Компонент на холсте содержит название, описание, поля и порты. Обязательное поле отмечено звёздочкой. Чтобы увидеть панель действий, нажмите на компонент. Панель появляется над ним.



На рисунке панель действий расположена непосредственно над названием «Вывод чата». Кнопка с треугольником внутри компонента запускает его выполнение при наличии необходимых входных данных.

Меню действий

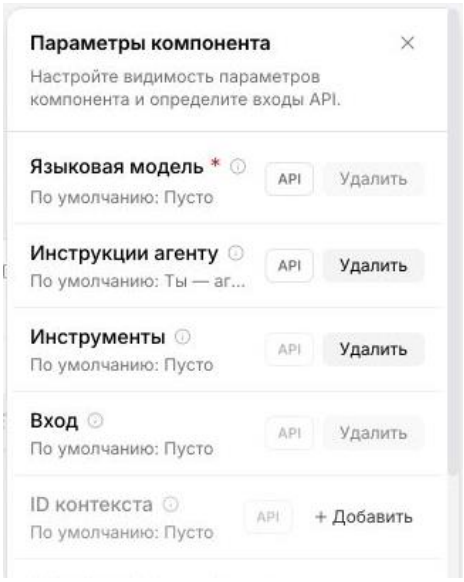
Кнопка «…» в правой части панели действий открывает меню компонента. Команда «Дублировать» добавляет копию на холст. Команда «Копировать» помещает компонент в буфер обмена. Команды «Свернуть» и «Развернуть» меняют объём сведений, видимых на холсте.



Команда «Удалить» исключает компонент из рабочего процесса. После удаления следует проверить соединения соседних компонентов. Команда «Скачать» позволяет получить файл компонента.

Панель параметров справа

Кнопка «Параметры» на панели действий открывает справа панель с заголовком «Параметры компонента». В ней настраивается, какие поля отображаются в компоненте. На рисунке показаны параметры агента.

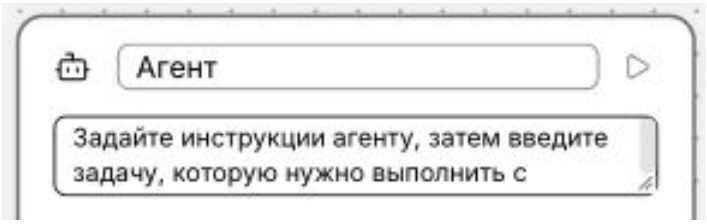


Команда «Добавить» выводит соответствующее поле в компоненте. Команда «Удалить» в строке параметра скрывает это поле. У скрытого поля может сохраняться значение по умолчанию. Значения открытых полей редактируются в самом компоненте.

Кнопка «API» в строке параметра управляет включением этого поля в автоматически сформированные примеры запросов. Доступность кнопки зависит от поля.

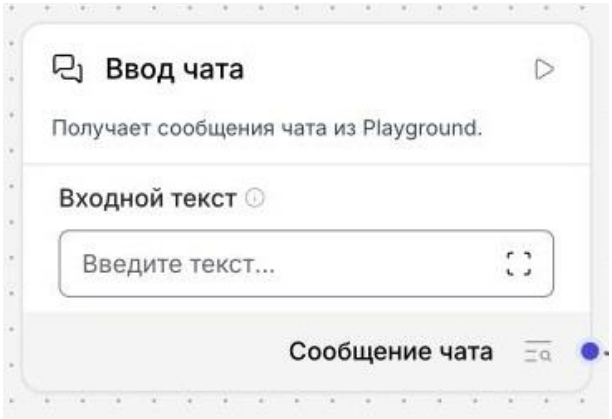
Название и описание компонента

Значок карандаша справа от выбранного компонента переводит его название и описание в режим редактирования. Поля появляются внутри компонента на холсте.



Компонент «Ввод чата»

Компонент «Ввод чата» принимает сообщение пользователя. В песочнице пользователь вводит сообщение в поле чата. При запуске через API текст передаётся в поле input_value.



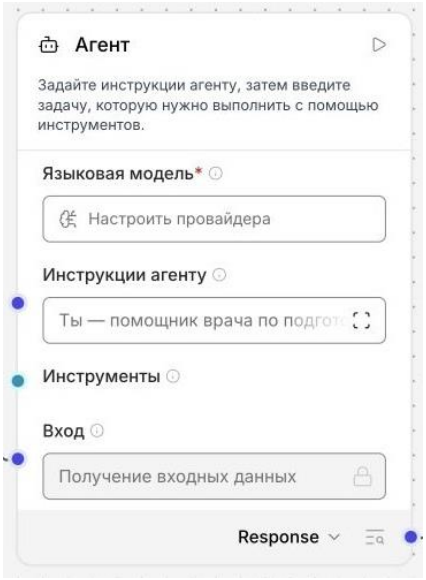
- **Входной текст** - Текст сообщения для обработки. Поле на холсте позволяет задать его вручную.

- **Сообщение чата** - Выходной порт. Через него принятое сообщение поступает в следующий компонент.
- **Сохранять сообщения** - Параметр сохранения входных сообщений в журнале сессии. По умолчанию включён.
- **Тип отправителя** - Категория отправителя входного сообщения.
- **Имя отправителя** - Имя, которое отображается у входного сообщения.
- **ID сессии** - Идентификатор сессии, к которой относится входное сообщение.

Дополнительные поля, в том числе «ID сессии» и «Сохранять сообщения», доступны в правой панели параметров. Для передачи сообщения на обработку выходной порт компонента должен быть соединён с подходящим входным портом.

Компонент «Агент»

Компонент «Агент» обрабатывает входной запрос языковой моделью. Результат зависит от выбранной модели, инструкций и подключённых инструментов.

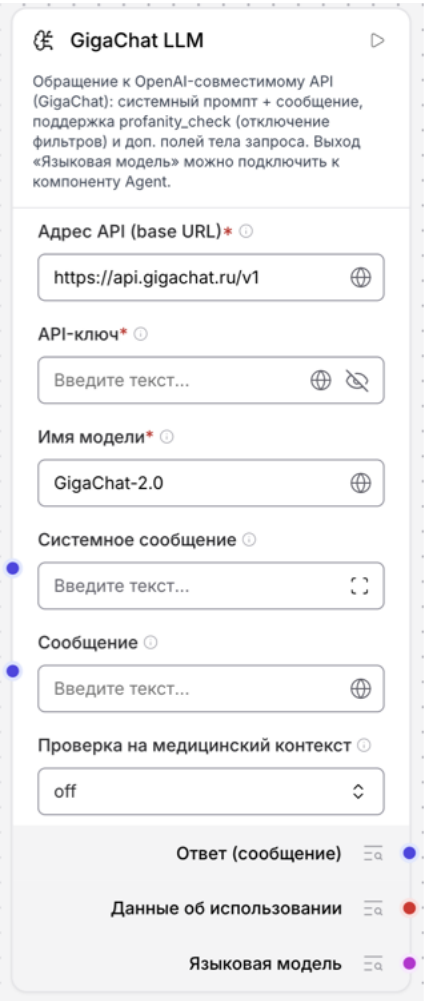


- **Языковая модель** - Обязательное поле. В нём задаётся модель, которая обрабатывает запрос.
- **Инструкции агенту** - Правила обработки запросов. В них задают назначение агента, требования к содержанию и формату ответа, а также ограничения.
- **Вход** - Поле для запроса, который должен обработать агент. Когда к порту подключён источник, в поле отображается «Получение входных данных».
- **Инструменты** - Входной порт для подключения инструментов, которые агент может вызывать при обработке запроса.
- **Response** - Выходной порт агента. Через него результат обработки передаётся подключённому компоненту.

Ввод запроса и задание инструкций выполняются в разных полях. Для постоянного правила, например требования вернуть ответ в заданной структуре, используется поле «Инструкции агенту». Текст отдельного запроса поступает в поле «Вход». Редактор инструкций открывается значком разворачивания справа в поле.

Компонент «GigaChat Med»

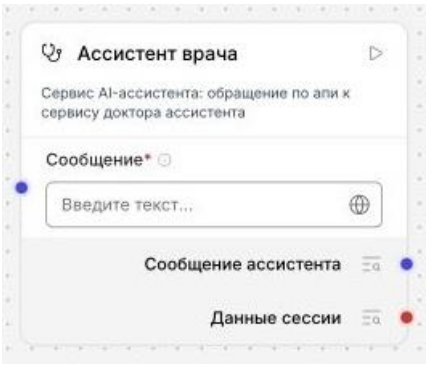
Компонент «GigaChat Med» обращается к модели GigaChat через OpenAI-совместимый API. Он формирует ответ на входное сообщение или передаёт настроенную модель компоненту «Агент».



- **Адрес API (base URL)** - Обязательное поле для адреса сервиса до /v1 включительно.
- **API-ключ** - Обязательное поле для ключа доступа к API GigaChat.
- **Имя модели** - Обязательное поле для имени модели в сервисе.
- **Системное сообщение** - Инструкция, которая задаёт правила формирования ответа.
- **Сообщение** - Текст запроса к модели. Его можно ввести вручную или передать через входной порт.
- **Проверка на медицинский контекст** - Управляет медицинским фильтром. Значение on включает проверку, off (по умолчанию) отключает, default использует настройку сервиса.
- **Ответ (сообщение)** - Выходной порт с ответом модели.
- **Данные об использовании** - Выходной порт с именем модели и статистикой использования из ответа API.
- **Языковая модель** - Выходной порт для подключения к одноимённому входу компонента «Агент».

Компонент «Ассистент врача»

Компонент «Ассистент врача» передаёт текстовые вопросы медицинских специалистов в сервис ассистента. Сервис учитывает сведения из запроса и при необходимости обращается к клиническим рекомендациям. В данной версии компонента не предусмотрены поиск и сопоставление сведений о лекарственных препаратах по инструкциям и фармацевтическим справочникам.



- **Сообщение** - Обязательное поле для вопроса. Введите вопрос вручную или подключите выход «Сообщение чата» компонента «Ввод чата».
- **Сообщение ассистента** - Выходной порт с ответом сервиса. Его можно соединить с портом «Входы» компонента «Вывод чата».
- **Данные сессии** - Выходной порт с ответом и идентификаторами сессии и запроса в формате JSON.

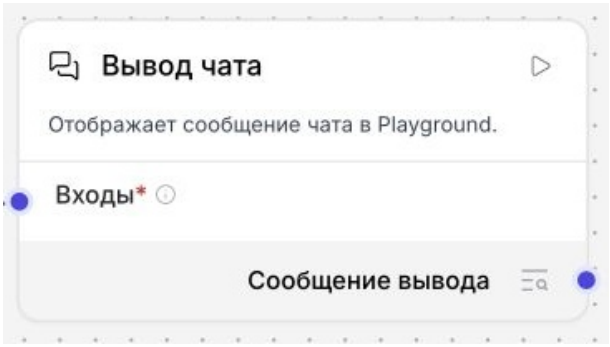
Дополнительные параметры

Чтобы открыть настройки, выберите компонент и нажмите «Параметры». Команда «Код» открывает редактор исходного кода компонента.

- **Ссылка на сервис** - Базовый адрес сервиса ассистента. Компонент сам добавляет к нему /v1.
- **Ссылка на кейклок** - Адрес эндпоинта /token в Keycloak, где компонент получает токен доступа.
- **Ключ идентификатора клиента** - Идентификатор клиента, зарегистрированного в Keycloak.
- **Секретный ключ клиента** - Секрет клиента Keycloak. Его можно выбрать из сохранённых переменных.
- **Grant Type** - Тип запроса токена. По умолчанию client_credentials.
- **Origin Header** - Значение HTTP-заголовка Origin. Укажите источник, разрешённый для токена.
- **Doctor User ID** - Идентификатор врача. Если поле пустое, используется идентификатор пользователя, от имени которого запущен процесс.
- **TypeApp** - Тип приложения. Доступны widget (по умолчанию) и extension. При выборе none поле TypeApp не передаётся.
- **Session ID (external)** - UUID сессии сервиса ассистента. Если поле пустое, компонент определяет сессию сервиса по сессии МАИС3.
- **Session ID** - Идентификатор сессии МАИС3. Заполняется при запуске, в том числе из поля session_id запроса к API.
- **Timeout, sec** - Тайм-аут HTTP-запроса в секундах. По умолчанию 300.

Компонент «Вывод чата»

Компонент «Вывод чата» формирует сообщение для отображения результата в песочнице. Он получает данные от компонента, который сформировал результат.



- **Входы** - Обязательный порт для данных, которые требуется показать в чате.
- **Сообщение вывода** - Выходной порт сформированного сообщения. Если вывод чата завершает процесс, этот порт можно оставить свободным.
- **Сохранять сообщения** - Параметр сохранения выходных сообщений в журнале сессии. По умолчанию включён.
- **Тип отправителя** - Категория отправителя результата.
- **Имя отправителя** - Имя, которое отображается у результата.
- **ID сессии** - Идентификатор сессии, к которой относится выходное сообщение.

Содержание результата определяется настройками компонента обработки. В компоненте вывода задаются параметры отображения и сохранения сообщения.

Соединения

Входные и выходные порты

Порты отображаются точками на границе компонента. Порт слева принимает данные, порт справа передаёт результат. Назначение порта определяется подписью рядом с ним.



На рисунке линия начинается у порта «Сообщение чата» и заканчивается у порта поля «Вход». Порт рядом с полем «Инструкции агенту» остаётся отдельным входом для передачи правил обработки.

Создание и проверка соединения

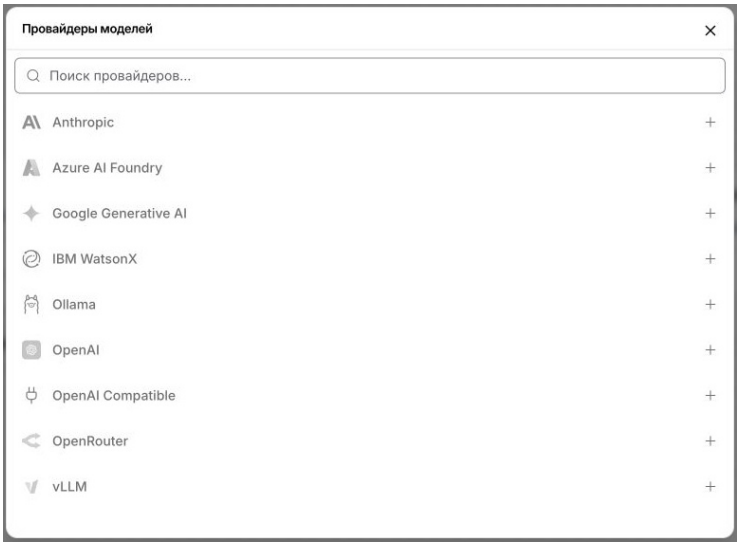
Соединение создаётся перетаскиванием от выходного порта источника к совместимому входному порту получателя. После соединения поле получателя может стать недоступным для ручного ввода. В таком случае данные поступают из подключённого компонента.

Для проверки пути данных проследите линию от выходного порта до входного. Если линии пересекаются, ориентируйтесь на их конечные точки. После удаления или замены

компонента проверьте, что входные порты следующих компонентов получают необходимые данные.

Языковые модели

Языковая модель задаётся в поле «Языковая модель» компонента «Агент». Доступный список зависит от настроенных провайдеров. Если источник не настроен, в поле отображается кнопка «Настроить провайдера».



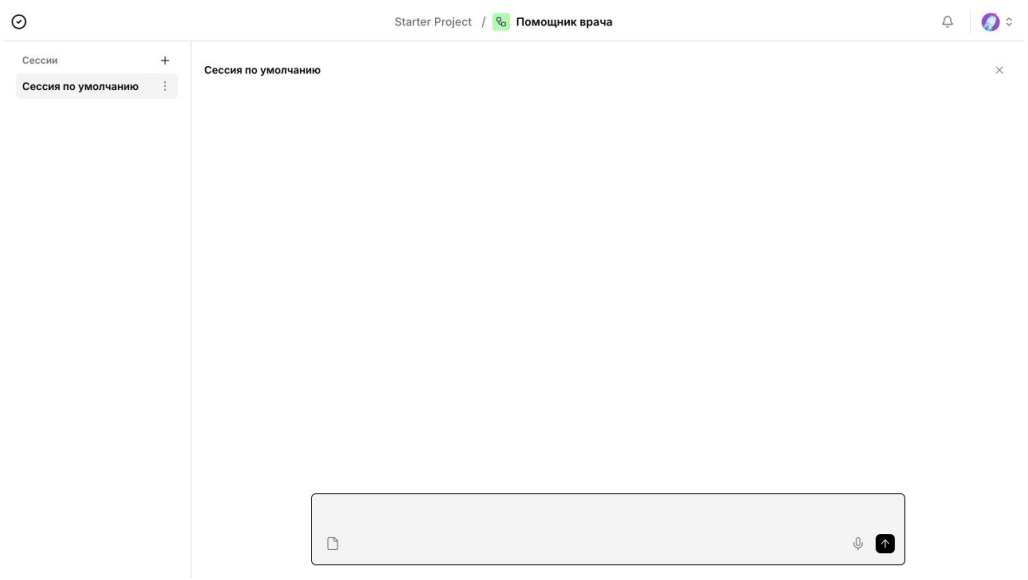
В окне можно найти провайдера по названию и открыть его настройки. Для сетевого сервиса могут потребоваться адрес и ключ доступа. Для локального источника используются настройки соответствующей среды. После настройки источника модель выбирается в компоненте агента.

Ключ провайдера используется для обращения к модели. Ключ МАИСЗ нужен вызывающей программе для доступа к API МАИСЗ. Доступность выбранной модели проверяется выполнением запроса в песочнице.

После смены модели следует повторить проверку с тем же запросом и сопоставить структуру и содержание результатов.

Проверка в песочнице

Песочница открывается кнопкой «Песочница» в правой верхней части редактора. Запрос вводится в нижнее поле чата. Кнопка со стрелкой вверх отправляет сообщение на выполнение. В области сообщений отображаются запрос пользователя и ответ процесса.

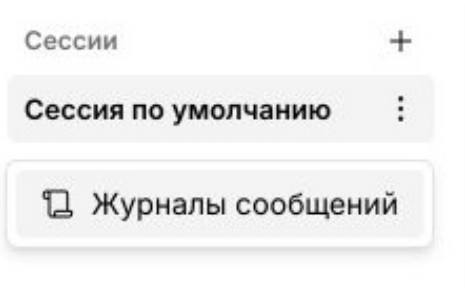


Для проверки процесса в режиме чата требуются компоненты приёма, обработки и вывода сообщения с действующими соединениями. Если обработку выполняет агент, ему должна быть доступна выбранная модель.

Перед проверкой выберите сессию. Чтобы выполнить запрос в новой сессии, откройте новый чат. После выполнения сопоставьте результат с исходным запросом и инструкциями агента. Для возврата к холсту нажмите значок закрытия в правой части окна песочницы.

Сессии и журналы сообщений

Выбор сессии
Список сессий расположен слева в песочнице. При выборе сессии открывается её история сообщений. Кнопка «Новый чат» с символом «+» создаёт новую сессию. В запросе к API идентификатор сессии задаётся в поле session_id.



Журнал сообщений
Чтобы открыть журнал, нажмите «…» рядом с названием сессии и выберите «Журналы сообщений». В окне «Логи сессии» отображаются сохранённые сообщения. По журналу можно проверить содержимое сохранённых входных и выходных сообщений.

Сохранение зависит от параметра «Сохранять сообщения» в компонентах ввода и вывода чата. При пустом журнале проверьте выбранную сессию, наличие отправленных сообщений и настройки сохранения.

Настройки памяти агента определяют, какие сообщения из истории сессии передаются модели при формировании ответа. При сравнении результатов после изменения настроек следует учитывать, в какой сессии выполнен запрос.

Диагностика

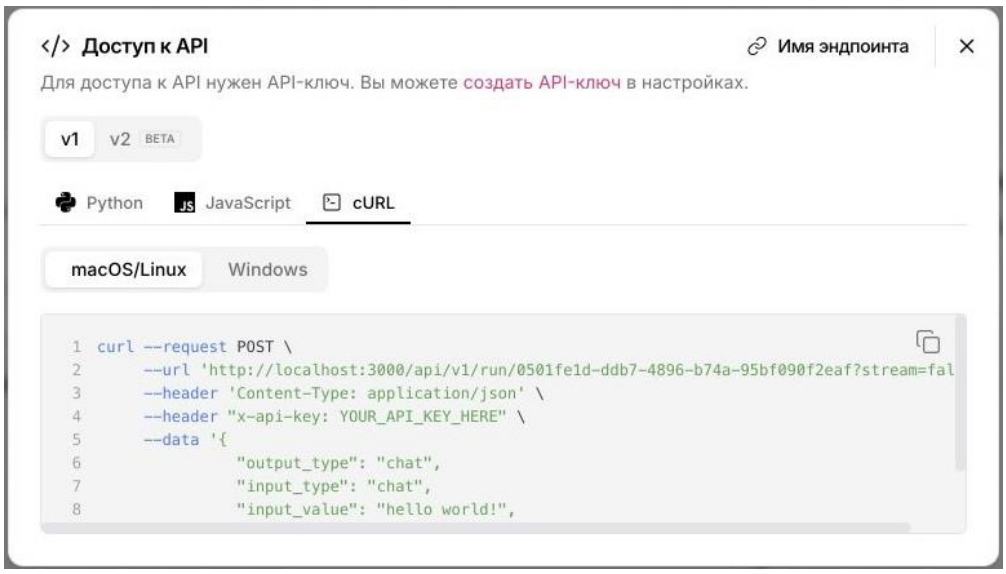
До запуска проверьте обязательные поля, доступность модели и путь передачи данных до компонента вывода. После запуска проверьте, что результат соответствует запросу и заданному формату.

- **В песочнице нет поля ввода** - Проверьте наличие компонента «Ввод чата» и его соединение с компонентом обработки.
- **Ответ не отображается** - Проверьте, завершилась ли обработка и поступают ли данные в порт «Входы» компонента «Вывод чата».
- **Ошибка обязательного параметра** - Проверьте поля со звёздочкой. Для агента обязательным является выбор языковой модели.
- **Ошибка обращения к модели** - Проверьте выбранную модель, параметры доступа к провайдеру и доступность сервиса.
- **Соединение не создаётся** - Проверьте, начинается ли оно у выходного порта и подходит ли тип данных входному порту.
- **Поле недоступно для ввода** - Проверьте, поступают ли данные в поле из подключённого источника.
- **Ответ не соответствует задаче** - Сопоставьте исходный запрос с инструкциями агента. После изменения настроек повторите запрос с теми же данными.

Для регистрации ошибки укажите название процесса, действие, входные данные и сообщение об ошибке. Для запроса через API также укажите код HTTP. Значения ключей в описание ошибки не включаются.

Доступ к API

Для запуска рабочего процесса из другой программы откройте в редакторе «Поделиться», затем выберите «Доступ к API». На вкладке «v1» отображаются адрес запроса, заголовок с ключом доступа и пример запроса.



Идентификатор процесса указан в адресе после /api/v1/run/. Адрес берётся из окна выбранного процесса. Для локального экземпляра он начинается с http://localhost:3000/. Для другой среды используется адрес соответствующего экземпляра МАИСЗ.

Создание ключа

Форма создания ключа открывается по ссылке «создать API-ключ» в окне доступа. Её также можно открыть кнопкой «Добавить» в разделе «API» настроек. В форме можно заполнить описание и выбрать дату истечения.

Кнопка «Сгенерировать API-ключ» создаёт ключ с указанными параметрами. При вызове процесса ключ передаётся в заголовке x-api-key.

Вызов через API использует настройки сохранённого процесса. Типы входа и выхода указываются в теле запроса. МАИСЗ связывает входное и выходное сообщения с сессией по её идентификатору.

Запрос через curl

Команда curl отправляет запрос методом POST. Пример приведён для macOS и Linux. Вариант для Windows доступен в окне «Доступ к API». Вместо YOUR_FLOW_ID указывается идентификатор процесса, вместо YOUR_API_KEY_HERE используется ключ МАИСЗ.

```
curl --request POST \
  --url 'http://localhost:3000/api/v1/run/YOUR_FLOW_ID?stream=false' \
  --header 'Content-Type: application/json' \
  --header 'x-api-key: YOUR_API_KEY_HERE' \
  --data '{
    "input_value": "Структурируй обезличенные заметки врача.",
    "input_type": "chat",
    "output_type": "chat",
    "session_id": "doctor-session-001"
  }'
```

- **input_value** - Текст сообщения для обработки.
- **input_type и output_type** - Типы входа и выхода. Значение chat используется для сообщений чата.
- **session_id** - Идентификатор сессии. Для продолжения сессии передайте прежнее значение, для новой сессии задайте другое.
- **stream=false** - Отключает потоковую передачу. Результат возвращается после завершения обработки.

Чтение ответа

При успешном вызове возвращается результат в формате JSON. В стандартном ответе v1 текст сообщения обычно находится по следующему пути.

```
outputs[0].outputs[0].results.message.data.text
```

Состав ответа зависит от выходных компонентов процесса. При кодах 401 и 403 проверьте, передан ли действующий ключ в заголовке x-api-key. При коде 404 проверьте адрес и идентификатор процесса. При кодах 400 и 422 проверьте тело запроса и описание ошибки в ответе.

Завершение работы ПО

При локальном размещении серверные контейнеры останавливает администратор.

Аварийные ситуации

Если ошибка повторяется после проверок из раздела «Диагностика», передайте описание ошибки в службу поддержки по адресу it@sbermed.ai.