MWS AI Agents Platform

Инструкция по эксплуатации после установки

Содержание

О ПЛАТФОРМЕ

MWS AI Agents Platform – это платформа для создания AI решений, которые автоматизируют обслуживание клиентов через коммуникационные каналы и поддерживают омниканальное взаимодействие. Особенность MWS AI Agents Platform – набор готовых инструментов, не требующих от пользователей опыта в машинном обучении или навыков программирования. Благодаря этому клиенты платформы могут самостоятельно управлять жизненным циклом ботов и AI-агентов, создавать и обслуживать ML-модели и AI-сервисы. В зависимости от варианта поставки платформа MWS AI Agents Platform предоставляет следующие возможности:

- создание сценария бота или агента в удобном no-code-конструкторе;
- обработка логики бота на высокопроизводительном и отказоустойчивом движке;
- подключение ботов к каналам взаимодействия с поверхностями;
- подключение и использование LLM к ботам;
- создание и обучение ML-моделей типов Классификатор и NER;
- разметка данных для формирования и совершенствования обучающих датасетов;
- создание RAG-сервисов для быстрого поиска ответов на вопросы в регулярно обновляемой базе знаний.

Основные понятия и термины

Агент
Автономная система, использующая LLM для сбора данных, анализа, принятия решений и выполнения задач различной сложности.

База знаний RAG
Набор документов, загруженных в систему, используемый RAG для выдачи релевантных ответов пользователю.

Блок активации (активационный блок)
Блок, содержащий условие активации сценария, к которому он относится. Условие активации проверяется относительно запроса пользователя.

Блок активации event
Блок, который запускает сценарий при наступлении определённого события, такого как начало диалога (init), пользовательское событие на поверхности, подключенной через канал HTTP, или отсутствие совпадения (no_match).

Блок активации intent
Блок, запускающий сценарий, если в сообщении пользователя распознан определённый интент.

Блок активации match
Блок, запускающий сценарий, если сообщение пользователя соответствует заданному регулярному выражению.

Блок реакции (реакционный блок)
Блок, содержащий логику, которая выполняется при активации сценария. Если сценарий содержит несколько блоков реакции, они выполняются последовательно.

Бот
Программа для имитации общения с пользователями через голосовые или текстовые интерфейсы, предназначенная для предоставления информации или выполнения задач.

Веб-клиент MWS AI Agents Platform
Пользовательский веб-интерфейс для работы с платформой через браузер. Позволяет создавать и адаптировать ботов под конкретные требования.

Датасет
Набор данных для обучения, тестирования и оценки ML-моделей.

Движок (agent-engine)
Сервис для обработки запросов пользователей с помощью ботов, созданных в no-code-конструкторе.

Индекс RAG
Хранилище фактологической информации для использования в RAG, содержащее векторную часть (embedding) и текстовое представление, позволяющее проводить быстрый поиск релевантных записей.

Интент
Намерение пользователя, отражённое в поисковом запросе.

Краулинг (crawling) Процесс автоматического сканирования, обхода и скачивания веб-страниц с помощью специальных сервисов – краулеров. Краулинг используется для сбора информации с веб-сайтов, анализа и индексации контента.

Классификатор (Classifier)
Модель машинного обучения, которая автоматически обучается и оптимизируется для решения задач классификации.

Модель машинного обучения (ML-модель)
Алгоритмическая конструкция для прогнозирования или классификации данных на основе обучения.

Паттерн
Формальное правило, описывающее ключевые слова и выражения для классификации запросов пользователя.

Проект

Совокупность сценариев разработанного бота или агента. Может содержать несколько версий.

Промпт
Подсказка для нейросети о том, что именно от неё требуется.

Рабочий процесс Langflow
Процесс в рамках фреймворка Langflow, интегрирующий несколько компонентов для обработки и генерации текста на основе пользовательских запросов и технологии RAG.

Узел (нода, node)
Вычислительная единица, выполняющая определенные действия. Состоит из блоков активации и реакции.

Сценарий
Логика работы навыка, направленная на достижение определённой цели в диалоге с клиентом. В no-code-конструкторе сценарий состоит из нод, связанных ребрами.

AI-сервис
Сервисы, использующие искусственный интеллект для обработки информации и решения задач в различных сферах.

AncSetFit (Anchored SetFit)

Расширение способа обучения ML-моделей SetFit, использующее «якорные» (anchor) примеры для каждого класса, что обеспечивает более стабильное обучение и лучшую генерализацию при экстремально малом количестве данных.

AutoML
Автоматизированный процесс создания и улучшения моделей машинного обучения.

Autoscaling (автомасштабирование)

Автоматическое изменение количества работающих экземпляров (инстансов) модели в зависимости от текущей нагрузки и потребления ресурсов.

Batching (батчинг)

Подход, при котором несколько отдельных запросов объединяются в одну группу (батч), чтобы выполнить их одновременно за одно обращение к модели.

Embedding (Эмбеддинг, вектор)
Векторное представление слова или фразы, полученное из моделей обработки естественного языка.

Fine-tuning

Дообучение предварительно обученной модели на новых данных путём обновления всех или части весов модели и/или добавления новых слоёв для адаптации к специфической задаче.

Keycloak
Стороннее решение для идентификации пользователей и контроля доступа.

Langflow
Фреймворк для создания агентов с помощью no-code конструктора.

Large language model (LLM)
Языковая модель, обученная на больших объёмах текста для выполнения задач NLP.

Named Entity Recognition (NER)
Модель машинного обучения для выделения и классификации именованных сущностей в тексте.

Natural Language Processing (NLP)
Область искусственного интеллекта, занимающаяся обработкой и анализом естественного языка.

Retrieval Augmented Generation (RAG)
Технология ML, объединяющая поиск информации и генерацию текста для создания ответов на запросы.

SetFit

Метод эффективного обучения ML-моделей на малом количестве данных, использующий обучение через сравнение примеров из разных классов и последующее обучение классификатора на полученных векторных представлениях (эмбеддингах) без дообучения всей модели.

Slug (слаг)
Уникальный ключ, который представляет собой имя сценария строчными буквами. Вместо пробелов в качестве разделителей используются знаки нижнего подчеркивания.

Webim
Омниканальная платформа для коммуникаций с клиентами. Подробнее см. https://webim.ru.

Проекты

Для создания и редактирования проектов используется специальный конструктор сценариев. Он представляет собой визуальный редактор с предопределенным набором блоков. Достаточно перетащить их в рабочую область и заполнить поля и связи. Проект представляет собой совокупность сценариев разработанного бота или агента. Может содержать несколько версий.

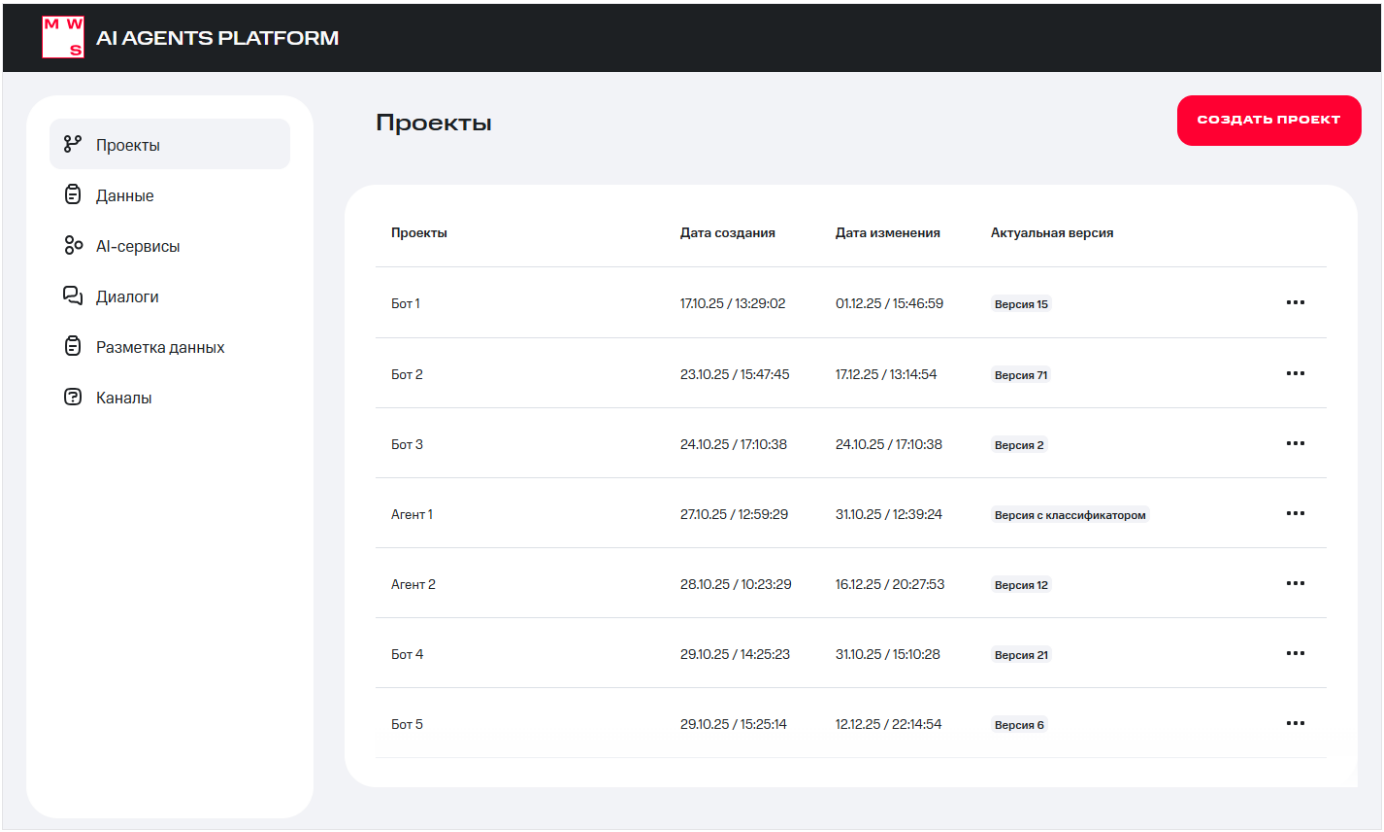
No-code-конструктор (конструктор сценариев) – это графический редактор в веб-клиенте MWS AI Agents Platform, который позволяет создавать, просматривать и редактировать диалоговые сценарии бота или агента без разработки кода.

Бот – это программа, предназначенная для имитации общения с пользователями с помощью голосовых или текстовых интерфейсов. Его основная задача – предоставить информацию или выполнить задачи, которые в обычной жизни решаются через общение с человеком. Проект бота представляет собой совокупность заранее написанных сценариев, которые активируются в зависимости от определенного действия пользователя.

Созданные боты работают на едином движке. Загрузка, сохранение и исполнение логики выполняются по определенным правилам. После создания бота его настройки формируются автоматически и их можно сохранить в файл формата JSON.

Агент – это обработчик запроса для передачи ответа на естественном языке. Агент является оберткой над ML-моделью с механизмом Function calling, позволяющим вызывать внешние инструменты для выполнения различных задач. Если для обработки запроса пользователя нужно использовать ML-модели, то создайте проект агента и подключите к нему внешние инструменты.

Созданные проекты отображаются в разделе **Проекты**. В нем можно посмотреть информацию о статусе, дате создания, изменения и актуальной версии:



Чтобы создать новый проект:

1. Создайте проект.
2. Создайте новую версию для сценариев.
3. Продумайте структуру сценариев.
4. Разработайте сценарий в конструкторе.

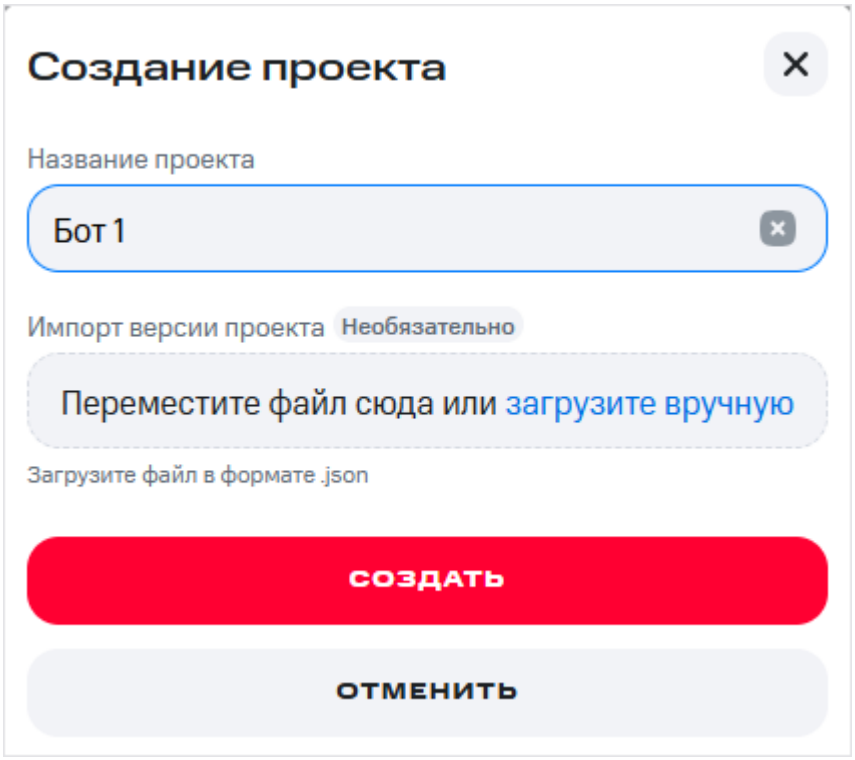
5. Привяжите классификатор для лучшего определения тематики диалога.

После этого подготовьте проект к использованию.

Создание проекта

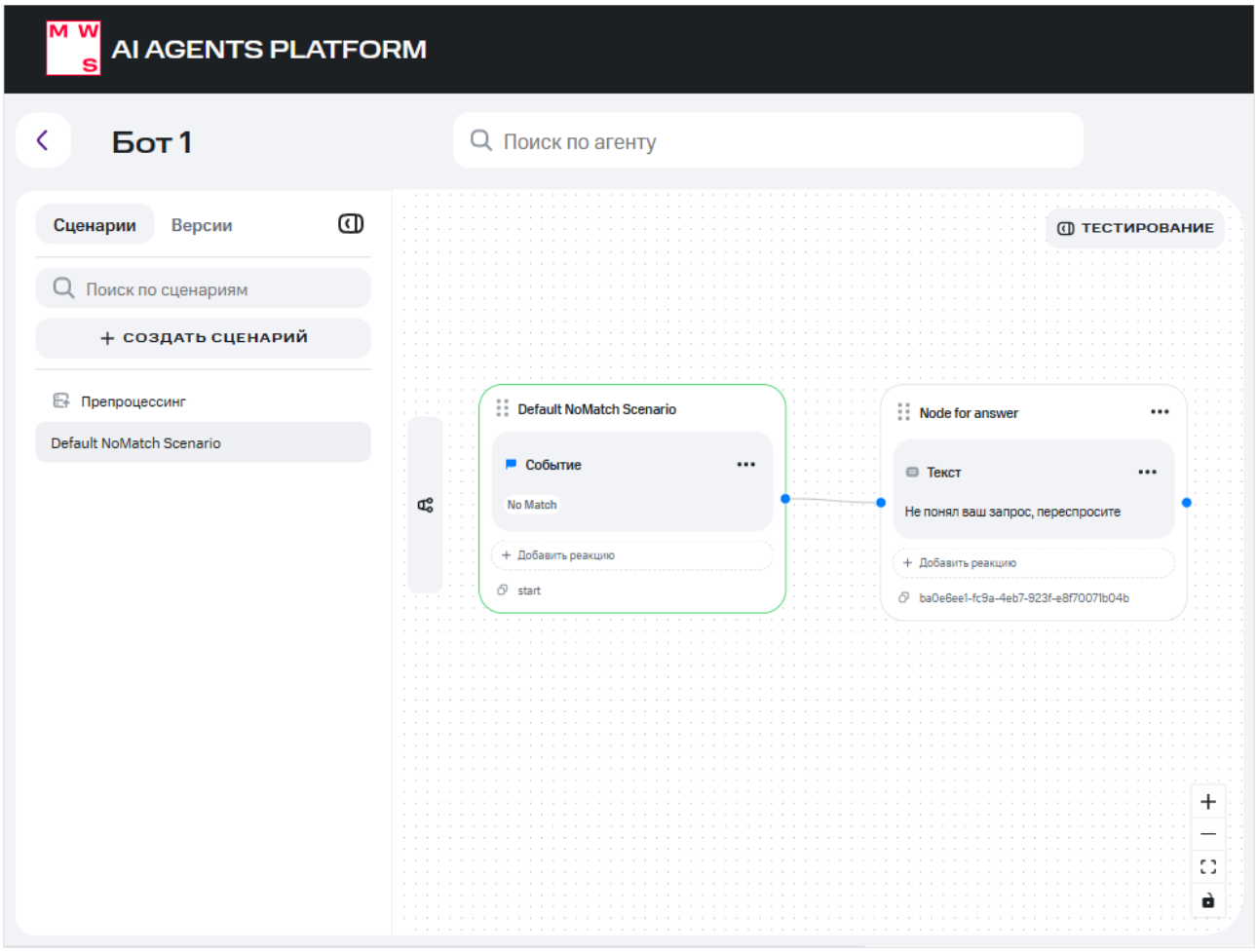
Чтобы создать проект:

1. На странице **Проекты** нажмите на кнопку **Создать проект**.
2. В открывшемся окне задайте название бота. Затем нажмите на кнопку **Создать**:



Если JSON-файл проекта ранее был выгружен, то на этом этапе вы можете импортировать их в систему. Для этого перетащите файл в область загрузки или выберите вручную через проводник. В результате в созданный проект добавятся сценарии, описанные в файле.

Открывается конструктор сценариев. Первый сценарий с именем «Default NoMatch Scenario» создается автоматически. Он содержит по одному активационному и реакционному блоку. При необходимости сценарий можно удалить. Также автоматически создается пустой сценарий препроцессинга. Его удаление и изменение имени невозможно.



3. Создайте новую версию, чтобы вносить изменения в сценарии.

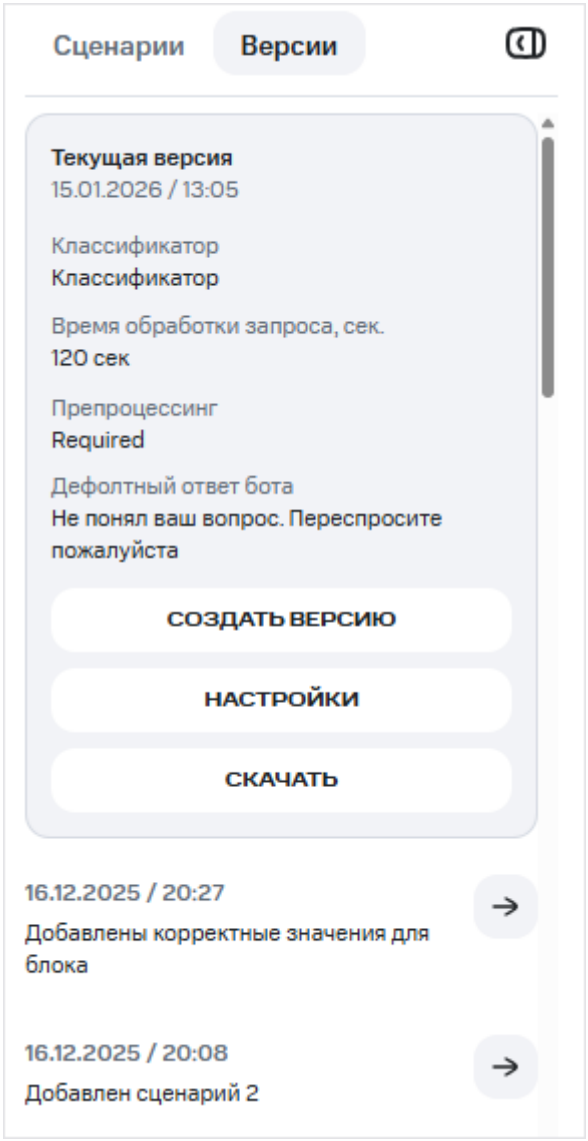
   Для изменения сценария должна быть создана новая версия. Если версия не создана, то изменения теряются.

Перед началом работы со сценариями продумайте их структуру. Затем наполните сценарий логикой.

Версии

Для проектов предусмотрено версионирование. Каждая новая версия проекта подразумевает его усовершенствование. Все изменения вносятся в текущую рабочую версию, по окончании редактирования ее нужно сохранить. После этого версию можно опубликовать, чтобы она стала доступна для выбора в канале.

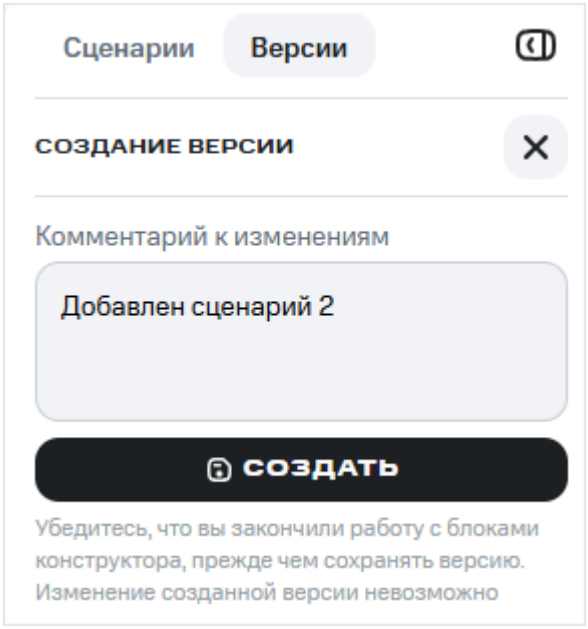
Сохраненная версия отображается в списке с указанным комментарием. Все новые изменения вносятся в текущую. Предыдущие версии открываются только на чтение.



Изменения в текущей версии сохраняются автоматически каждые 10 секунд бездействия, при переключении на другой сценарий или при выходе из проекта. Благодаря этому изменения не потеряются. При этом виджет тестирования запускается для последнего сохраненного состояния, поэтому рекомендуется предварительно нажать на кнопку **Сохранить изменения**, чтобы учесть последние доработки.

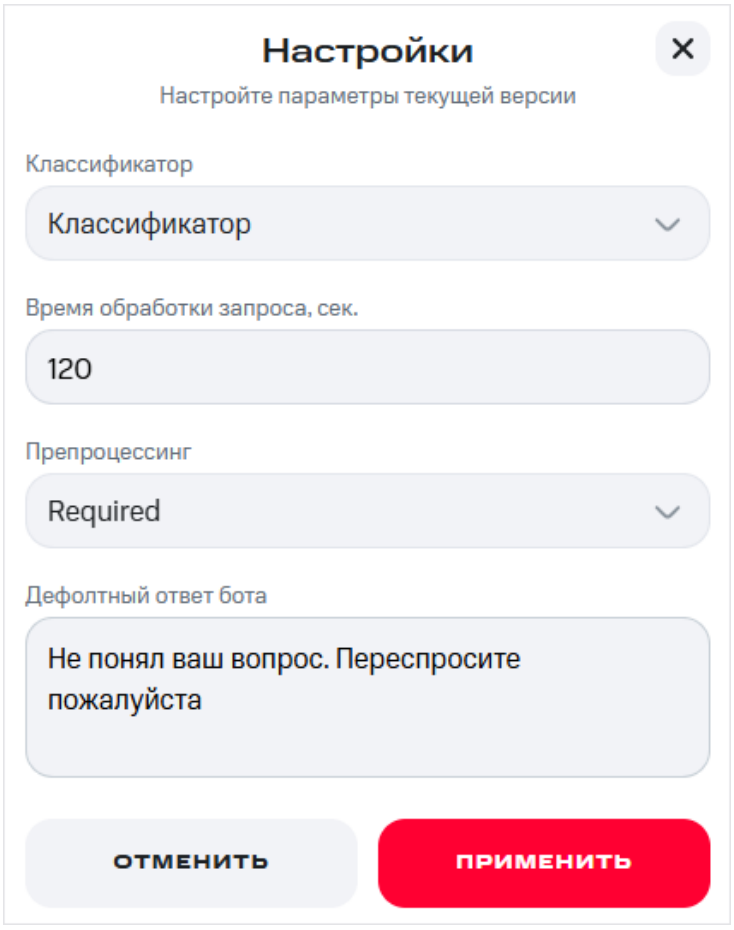
На панели **Версии** для текущей версии доступны кнопки:

- **Создать версию**. Нажмите на кнопку, чтобы создать копию текущей версии и сохранить в ней разработанные сценарии. Укажите комментарий к изменениям и нажмите на кнопку **Создать**.



  В результате создается версия и отображается в списке. Дальнейшая работа со сценариями продолжается в текущей версии. Чтобы сохранить внесенные изменения, снова создайте версию;

- **Настройки**. По кнопке открывается окно для изменения настроек версии:



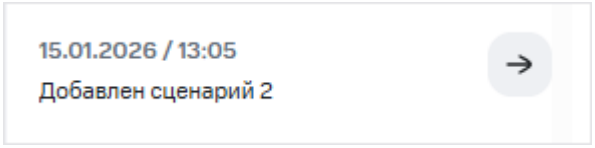
Классификатор. Имя сервиса классификатора.
Время обработки запроса, сек. Время обработки запроса в секундах. Значение по умолчанию – 5 секунд.

Время обработки запроса должно быть синхронизировано с тем, которое указано в вашем чат-сервисе.

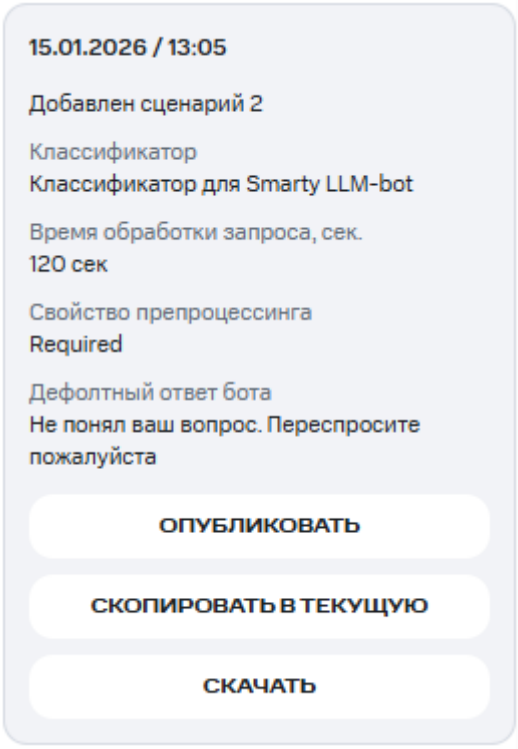
Препроцессинг. Возможные значения: Disabled – препроцессинг отключен, Required – выполнение препроцессинга обязательно.
Дефолтный ответ бота. Ответ от бота по умолчанию, если стейт не найден.

- **Скачать**. Нажмите на кнопку, чтобы сохранить настройки в формате JSON. При создании нового бота можно импортировать эти настройки.

Чтобы отобразить информацию об одной из предыдущих версий, нажмите на кнопку :



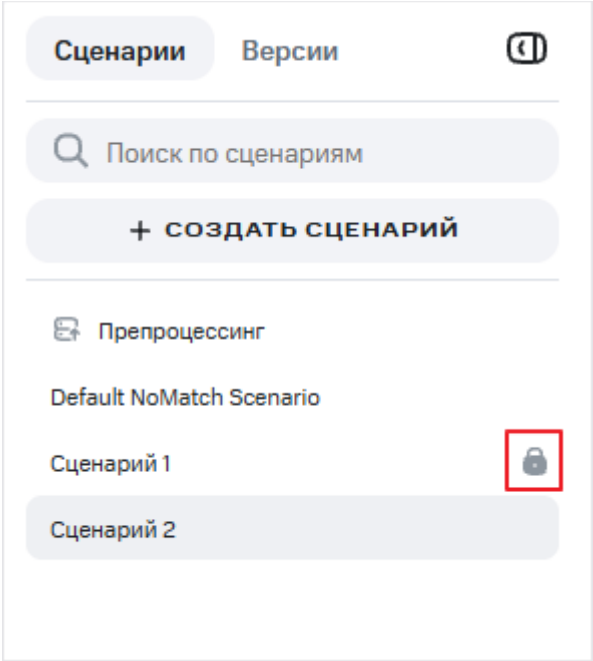
Для предыдущих версий доступны действия:



- **Опубликовать**. Нажмите на кнопку, чтобы опубликовать версию. В списке рядом с опубликованной версией отображается значок ;
- **Скопировать в текущую**. В результате выбранная версия будет скопирована в текущую;
- **Скачать**. Нажмите на кнопку, чтобы сохранить версию в файле формата JSON.

Совместное редактирование

Редактировать текущую версию могут сразу несколько пользователей. При изменении сценария на него устанавливается блокировка. Для других пользователей отображается соответствующий значок:



Это означает, что другие пользователи смогут его редактировать, когда:

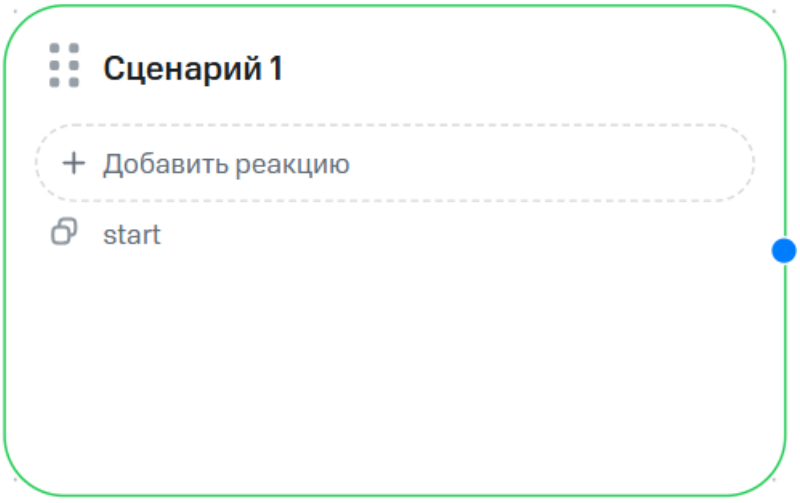
- текущий пользователь сохранит изменения;
- текущий пользователь перейдет к редактированию другого сценария;
- истечет время бездействия, по умолчанию 5 минут.

Кроме этого, для сохранения версии должны быть сняты все блокировки со сценариев.

Создание версии

Чтобы создать новую версию:

1. Создайте сценарий. Для этого на вкладке **Сценарии** нажмите на кнопку **Создать сценарий**. В результате создается новый сценарий с пустой стартовой нодой:



При создании и редактировании сценария на него устанавливается блокировка. Другие пользователи не могут редактировать его, пока время блокировки не истечет или она не будет снята, например при переходе к редактированию другого сценария.

Задайте имя для сценария. Добавьте в стартовую ноду нужные блоки активации. Наполните сценарий логикой. Для этого добавьте в него реакционные блоки и связи. Подробнее см. раздел «Разработка сценариев».

2. Заполните настройки версии. Для этого нажмите на кнопку **Настройки**. В открывшемся окне заполните поля: **Классификатор**, **Время обработки запроса, сек.**, **Препроцессинг** и **Дефолтный ответ бота**.

   Время обработки запроса должно быть синхронизировано с тем, которое указано в вашем чат-сервисе.

3. На панели **Версии** нажмите на кнопку **Создать версию**.

   Чтобы сохранить версию при совместном редактировании проекта, нужно дождаться снятия блокировок со всех сценариев.

4. Укажите комментарий к изменениям. Нажмите на кнопку **Создать**.

В результате создается копия текущей версии и сохраняется с указанным комментарием в списке версий. Чтобы она стала доступна в канале, ее нужно опубликовать. Все новые изменения вносятся в текущую версию. При необходимости вы можете вернуть состояние ранее сохраненной версии. Для этого перейдите в нее и нажмите на кнопку **Скопировать в текущую**. Сценарии и настройки этой версии будут скопированы в рабочую.

Разработка сценариев

Для определения логики бота используются сценарии. Они представляют собой последовательность связанных нод, которые, в свою очередь состоят из блоков. Внутри одной ноды блоки выполняются последовательно. Таким образом, сценарий определяет поведение и реакцию бота на входящие сообщения или события.

В качестве запускающего механизма для сценария используются *блоки активации*. Они определяют, когда бот должен активироваться и начать взаимодействие с пользователем. Для каждого блока активации создаются *блоки реакции* – они определяют, как бот должен реагировать на входящее сообщение.

Чтобы создать новый сценарий:

1. Продумайте структуру.
2. В конструкторе сценариев на вкладке **Сценарии** нажмите на кнопку **Создать сценарий**.
3. Кликом правой кнопки мыши по созданному сценарию вызовите контекстное меню. В нем выберите пункт **Переименовать**. Заполните название сценария и нажмите ENTER.
4. Наполните сценарий заранее разработанной логикой. Для этого добавьте нужные блоки и связи между ними.

   В реакционный блок можно добавлять сразу несколько блоков. В этом случае он является нодой и все блоки внутри него выполняются последовательно.

5. Создайте все необходимые сценарии аналогичным образом.
6. Привяжите классификатор. Классификатор позволяет группировать ответы пользователей на основе предварительно размеченных данных. Это означает, что можно задать фразу или набор фраз, которые послужат эталоном для сравнения с запросом пользователя. Эти запросы будут проверяться на семантическое соответствие заданным фразам. Если порог такого соответствия достаточно высок, то можно считать, что две реплики относятся к одному и тому же классу. Соответственно, чат-бот будет на них реагировать одинаково.

Структура сценариев

Перед созданием сценариев продумайте их логическую структуру. Она может представлять собой как линейный список, так и иерархию. Сценарии должны определять логику работы бота. Они описывают переходы бота из одного состояния в другое в зависимости от полученного ответа клиента. Важно продумать максимальное количество возможных вариантов запросов и ответов, а также описать переход в блок с типом No match, когда для запроса не удалось найти подходящее условие активации.

Чтобы создать логическую структуру:

1. Определите тематики запросов, которые можно сгруппировать, и используйте их при именовании сценариев.

Например, бот для обслуживания клиентов медицинской организации может записать клиента к врачу, зарегистрировать в программе лояльности или показать информацию о доступных услугах. В этом случае нужно создать три глобальных сценария: «Запись к врачу», «Регистрация в программе лояльности», «Справка по услугам».

2. Продумайте логику активации. Определите варианты ситуаций и реакции на них. Например, пользователь отправляет боту сообщение: «Привет». В этом случае:
   - блок активации бота изменяется, бот активируется для начала диалога;
   - блок реакции определяет, что бот должен ответить пользователю приветствием, например: «Привет! Как дела?»

После разработки структуры приступайте к созданию сценария в конструкторе.

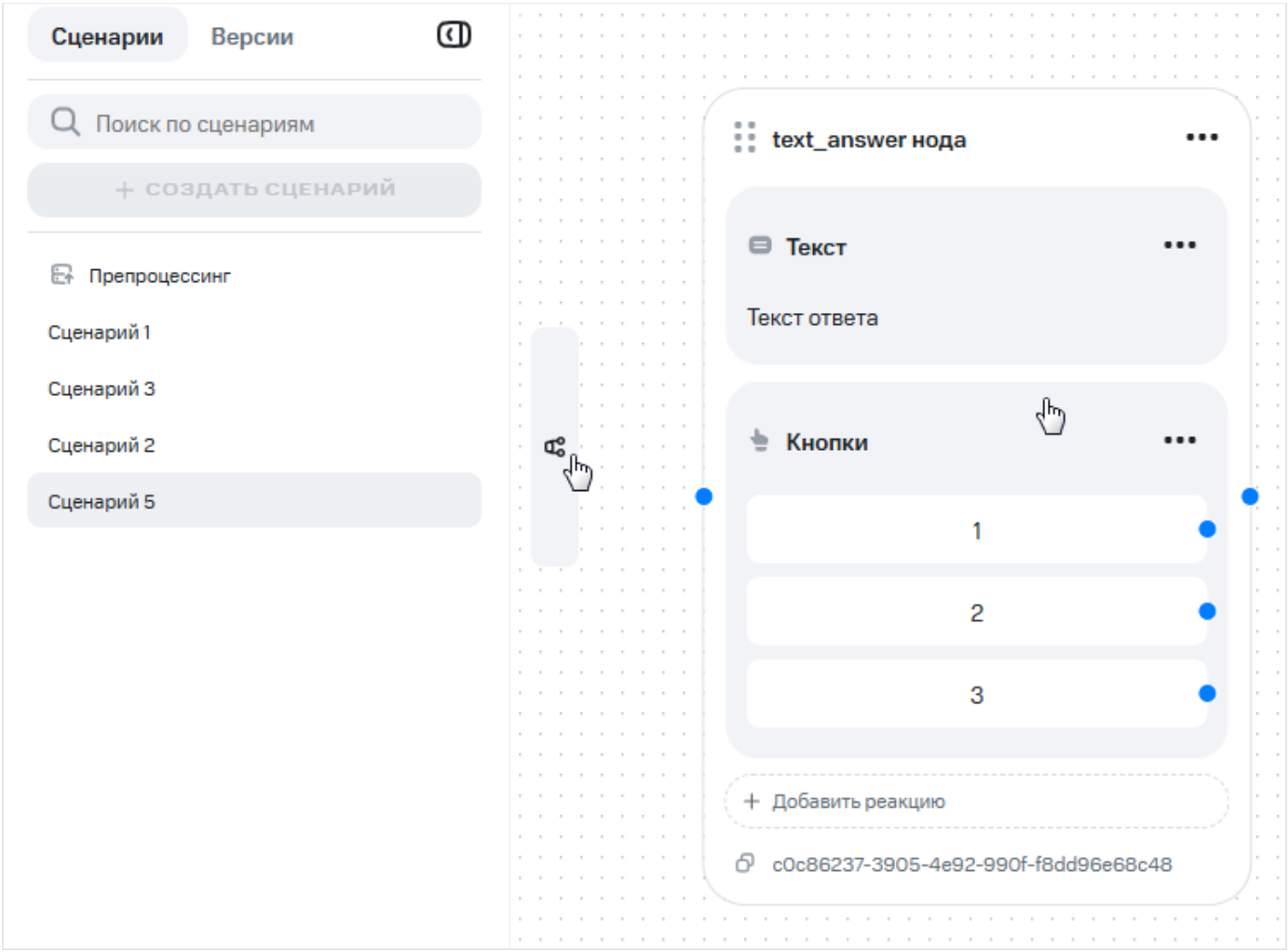
Конструктор сценариев

Сценарий представляет собой последовательность связанных нод, внутри которых располагаются блоки. В зависимости от действий пользователя срабатывает определенная реакция.

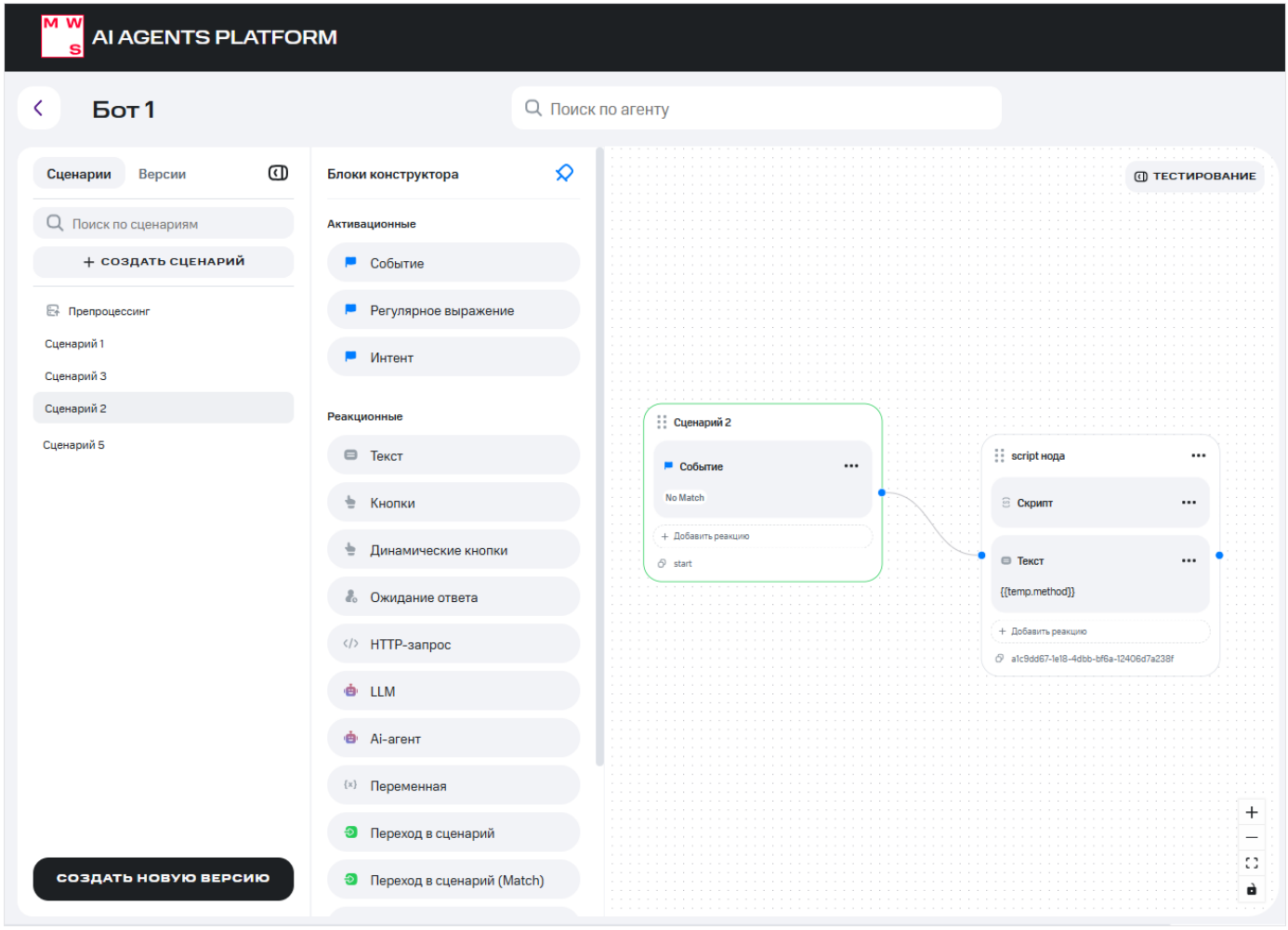
Ноды состоят из блоков – это элементы, которые описывают, как сработает определенный сценарий и какие действия при этом выполняются. Связи представлены в виде линий, показывающих переходы.

Для одного бота можно добавить несколько сценариев, каждый из которых может содержать вложенные. Список сценариев, блоки и настройки версии доступны в левой части конструктора. При необходимости можно настроить переход из одного сценария в другой.

Чтобы отобразить список доступных блоков, нажмите на кнопку :



В результате открывается панель **Блоки конструктора**. Для удобства ее можно закрепить по кнопке .



Рабочая область

Вся работа со сценариями ведется в рабочей области. Она представляет собой двумерное пространство, которое не ограничено в высоту и ширину.

Чтобы добавить элементы, перетащите их с панели компонентов в рабочую область (drag-and-drop). Изменить масштаб поля можно прокруткой колесика мыши.

Также в рабочей области доступен поиск по элементам сценария. Например, введите в строке поиска название блока, и фокус переместится на первый найденный блок с совпадением.

На рабочей области доступны действия:

Кнопка	Описание
+	Увеличить масштаб
—	Уменьшить масштаб
⛶	Изменить масштаб, чтобы сценарий поместился в зону видимости
🔒	Установить/снять блокировку на сценарий

Сценарии

Работа со сценариями ведется во вкладке **Сценарии** на панели в левой части экрана.

По умолчанию в новом проекте автоматически создаются сценарии:

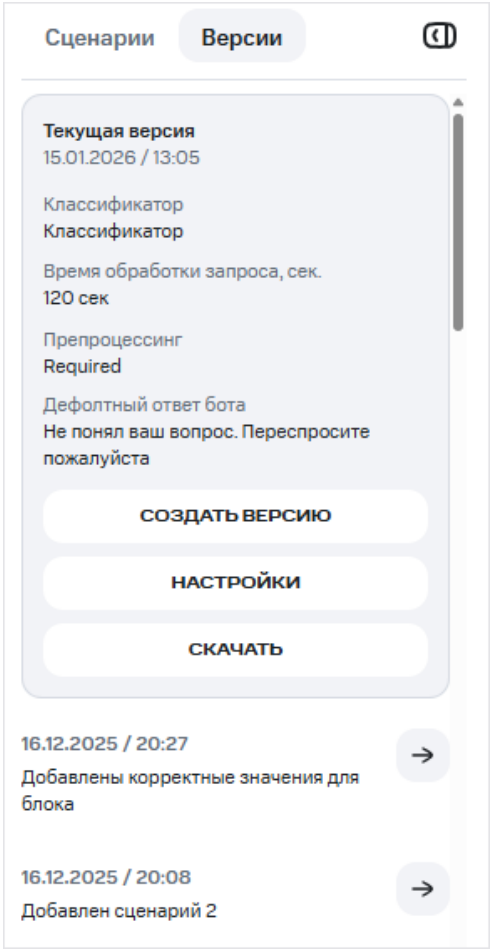
- Default NoMatch Scenario – для ситуаций, когда интент или регулярное выражение не определены;
- Препроцессинг – это технический сценарий, который выполняется сразу после поступления запроса от пользователя и до выбора блока активации.

Чтобы добавить новый сценарий, нажмите на кнопку **Создать сценарий**. По щелчку правой клавиши мыши на имя сценария доступно контекстное меню, из которого можно переименовать, удалить сценарий или скопировать его slug (слаг) – уникальный ключ, который представляет собой имя сценария строчными буквами. В качестве разделителя вместо пробела используется знак нижнего подчеркивания. Slug предназначен для задания логики перехода в блоке **Скрипт**.

При необходимости обратиться к сценарию рекомендуется использовать slug сценария вместо его идентификатора, так как он остается постоянным, а ИД может измениться, например при импорте сценария.

Версии

Для работы с версиями используется вкладка **Версии**. На ней отображаются все созданные версии и информация о них.



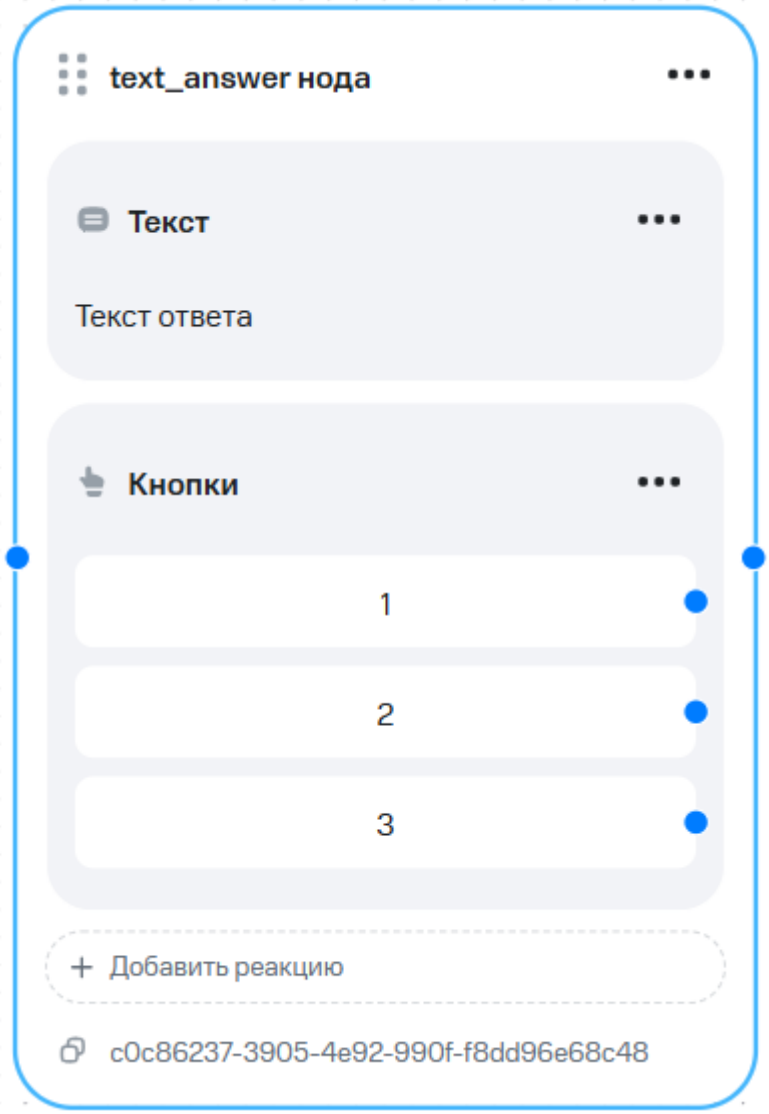
Подробнее см. раздел «Версии».

Блоки конструктора

Сценарии состоят из нод (стейтов). Нода представляет собой блок, внутри которого содержится набор блоков. Они выполняются последовательно в рамках одного стейта. В конструкторе

сценариев доступны активационные и реакционные блоки. Блоки активации определяют, по каким действиям пользователя бот переходит к конкретному сценарию. При попадании в этот сценарий выполняются блоки реакции.

Доступные для добавления блоки располагаются на панели **Блоки конструктора**. Чтобы добавить блок в нужную ноду, перетащите его с помощью левой клавиши мыши.



По кнопке ••• ноду можно переименовать или удалить.

В нижней части нод и блоков располагаются идентификаторы. При необходимости их можно скопировать. Для этого нажмите на кнопку . Например, при редактировании большого сценария запустите поиск по ИД, чтобы вернуться к заполнению блока.

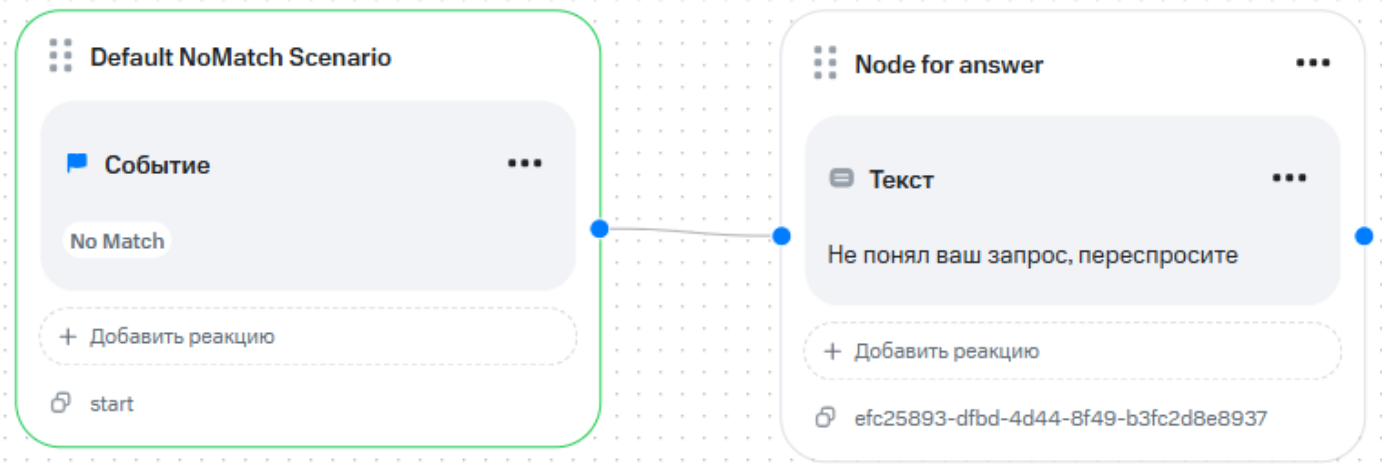
Для удобства используйте горячие клавиши:

- CTRL+C – копировать блок;
- CTRL+V – вставить скопированный блок;
- CTRL+Z – отменить действие;
- DELETE/BACKSPACE + выделенный блок/нода/связь – удалить выделенный элемент.

В сценарии рекомендуется использовать не более 200 блоков, так как большое количество блоков может повлиять на производительность.

Связи

Связи между нодами образуют последовательность выполнения сценария. Чтобы создать связь, нажмите на синюю метку на блоке и протяните курсор мыши к нужному блоку.



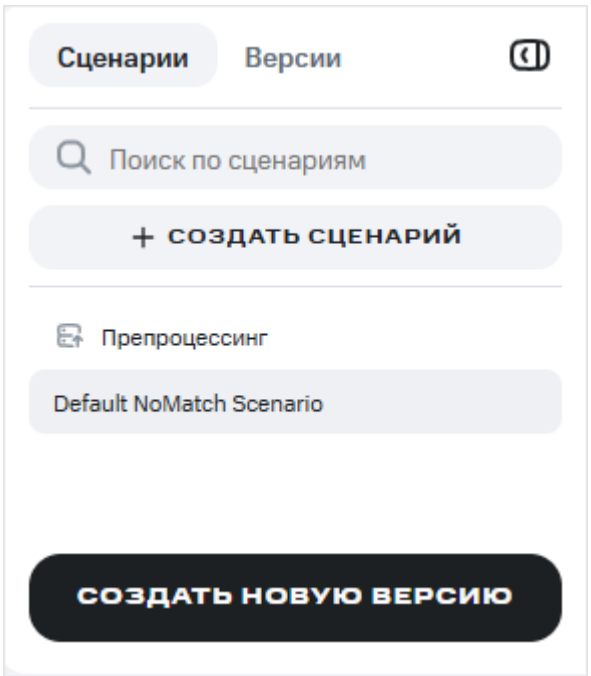
Чтобы выполнять блоки последовательно в рамках одной ноды, добавьте их друг за другом, перетащив из панели компонентов в область **Добавить реакцию**.

Компоненты сценария

При создании проекта автоматически создаются сценарии:

- Препроцессинг – это технический сценарий, который используется для обогащения контекста сессии. Позволяет повысить качество и скорость обработки запроса пользователя. Выполняется сразу после поступления запроса от пользователя и до выбора блока активации. По умолчанию не содержит блоков;
- Default NoMatch Scenario – сценарий по умолчанию. Содержит блок активации **Событие** с типом No Match и блок реакции **Текст**.

Остальные сценарии нужно создавать вручную.



При поступлении запроса от пользователя сначала выполняется препроцессинг, если в настройках версии указано его выполнение. После этого проверяются активационные блоки остальных сценариев. Как только срабатывает блок активации, выполняются реакционные блоки

соответствующего сценария. Подробнее о процессе активации сценариев с иерархией см. в разделе «Создание структуры».

На панели **Блоки конструктора** располагаются доступные активационные и реакционные блоки. Чтобы добавить блок, наведите курсор мыши на компонент и, удерживая левую клавишу мыши, перетащите его в рабочую область. После создания блок можно перемещать по рабочей области с помощью механизма drag-and-drop.

Состав доступных для добавления блоков зависит от типа сценария. Для сценария «Препроцессинг» набор блоков ограничен.

Для всех активационных и реакционных блоков можно заполнить поле **Тэги** по кнопке **Добавить тэги**. Примеры: zapis_yes, good_bot, success. Значения из этого поля используются для потребностей аналитики диалогов, например для расчета метрик. Кроме этого, при заполнении блоков можно указывать зарезервированные переменные.

Препроцессинг

До активации основных сценариев в боте может выполняться препроцессинг. Для одной версии бота допускается один такой сценарий. Выполняется после каждого запроса пользователя. Препроцессинг не может содержать блоки для взаимодействия с пользователем в чате и перевод на оператора. При этом в препроцессинге можно задать условие прямого перехода в конкретный сценарий бота, определить значения переменных, использовать интеграционные функции HTTP-вызова и т.д.

Чтобы сценарий препроцессинга выполнялся, в настройках версии в поле **Препроцессинг** должно быть установлено значение **Required**.
Если в препроцессинге используется блок **Переход в сценарий** и в нем установлен флажок **Вернуться после завершения**, то сценарий, к которому нужно перейти, также выполняется в рамках препроцессинга. Поэтому проверьте, что в нем используются только допустимые блоки: **HTTP-запрос**, **Переменная**, **Переход в сценарий**, **Переход в сценарий (Match)**, **Условие**, **Скрипт**. При попытке выполнить недопустимые блоки возникает ошибка.

Кроме этого, в сценарии препроцессинга задаются:

- вычисление кандидатов для активации по правилам;
- вычисление кандидатов для активации по интентам.

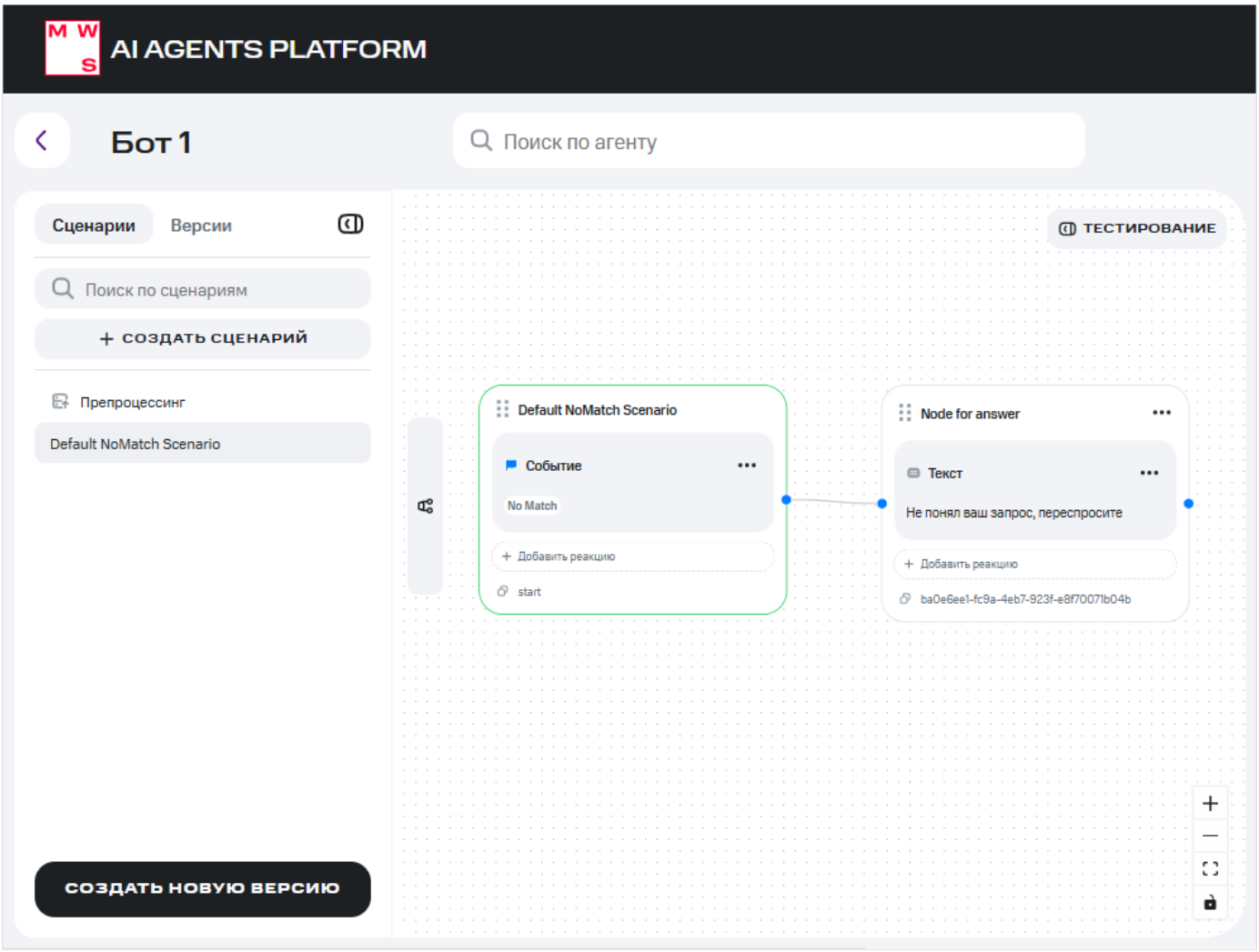
Подробнее об активации см. раздел «Блоки активации».

Информация о работе этого типа сценария записывается в историю диалогов аналогично другим сценариям.

Препроцессинг нельзя удалить. Если он не нужен для исполнения логики вашего бота, то оставьте его пустым.

Default NoMatch Scenario

При создании бота по умолчанию создается сценарий, в который бот попадает, если интент или регулярное выражение не определены:



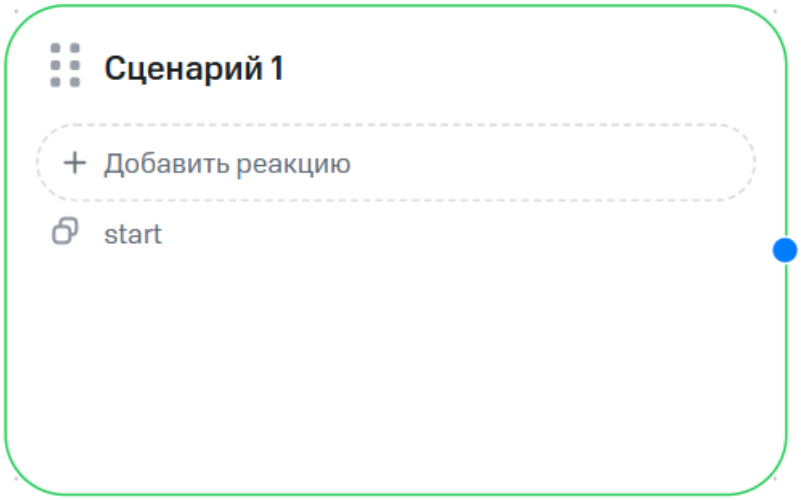
При необходимости сценарий можно удалить.

Блоки активации

Блоки активации определяют, по какому запросу пользователя бот переходит к выполнению конкретного сценария. Сценарий может выбираться по одному из условий активации:

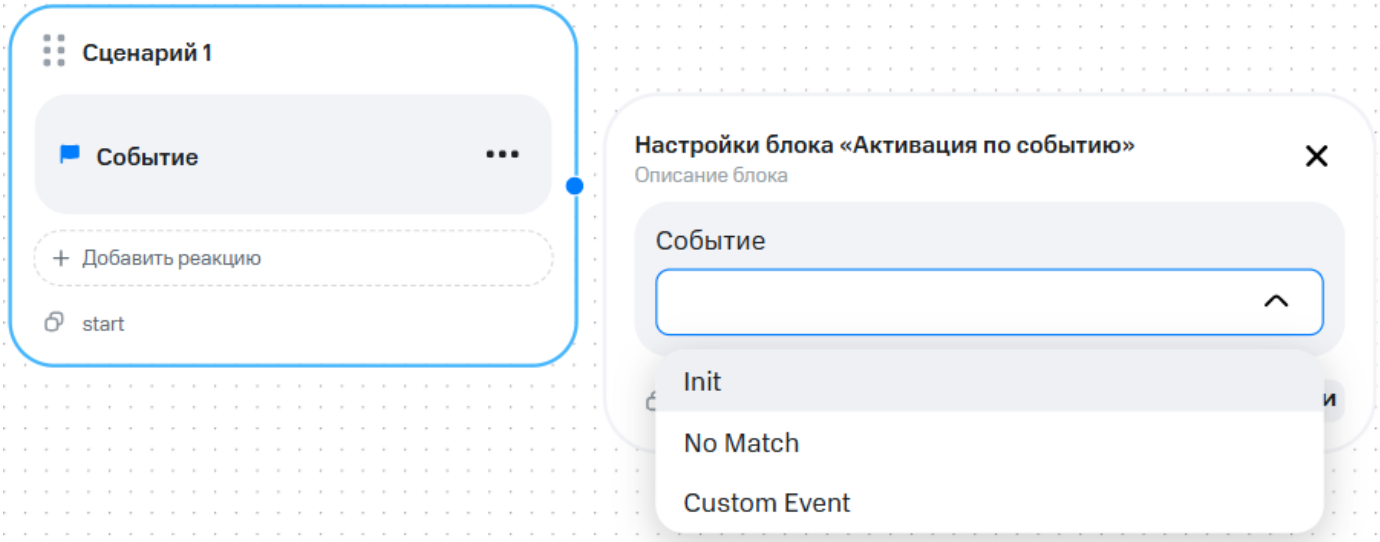
- событие (Event);
- интент (Intent);
- регулярное выражение (Match).

Блоки активации можно добавлять только в стартовую ноду:



Приоритет активации настраивается в параметре **PRIORITY_ACTIVATOR** файла values.yaml.

Событие



Активация по событию: по первому запросу пользователя, если не удалось определить сценарий для активации или по событию, пришедшему с поверхности.

Активация по пользовательскому событию поддерживается только в проектах, подключенных к поверхности через канал HTTP. Если используется другой тип канала, то активация по пользовательскому событию не выполнится.

В выпадающем списке выберите одно из возможных значений:

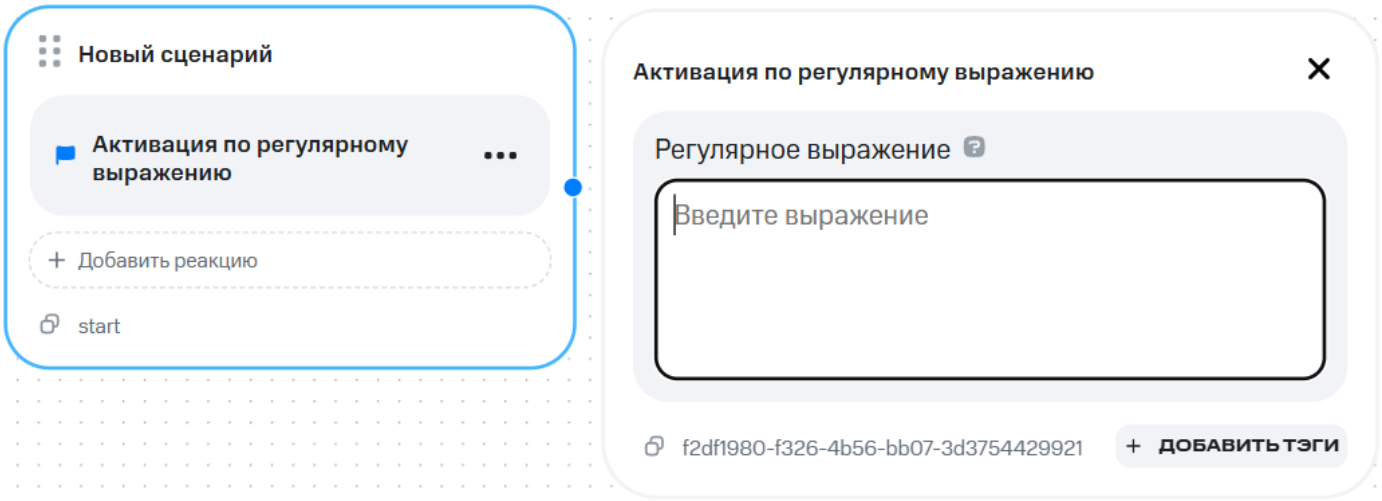
- **Init** – событие начала диалога. Сценарий активируется, если у пользователя не было открытой сессии до обработки текущего сообщения. Если в блоке активации по событию указано значение **Init**, то при первом же сообщении от пользователя выбирается сценарий, в котором находится этот блок;
- **No Match** – не удалось определить сценарий для активации;
- **Custom Event** – пользовательское событие. Тип предназначен для активации по событию, пришедшему с поверхности по протоколу HTTP. Примеры событий: открытие виджета с чатом на сайте, комментарий к посту в социальной сети, поступление входящего сообщения на почтовый адрес компании и т.д. Событие отслеживается на поверхности, его имя и другая информация поступает в запросе к HTTP-адаптеру.

Подробнее о структуре запроса см. раздел «Метод POST /api/{channelId} – получить входящее сообщение от http-клиента».

Чтобы протестировать активацию по пользовательскому событию, в тестовом виджете переопределите системную переменную **system.input**: задайте для нее тип и имя события, которые нужно проверить.
Пример настройки контекста:

```
{
  "system": {
    "surfaceMetadata": {},
    "input": {
       "type": "event",
       "name": "my_event",
       "data": {
          "data_1":"data_1_value"
       }
    }
  },
  "session": {},
  "request": {},
  "temp": {}
}
```

Регулярное выражение

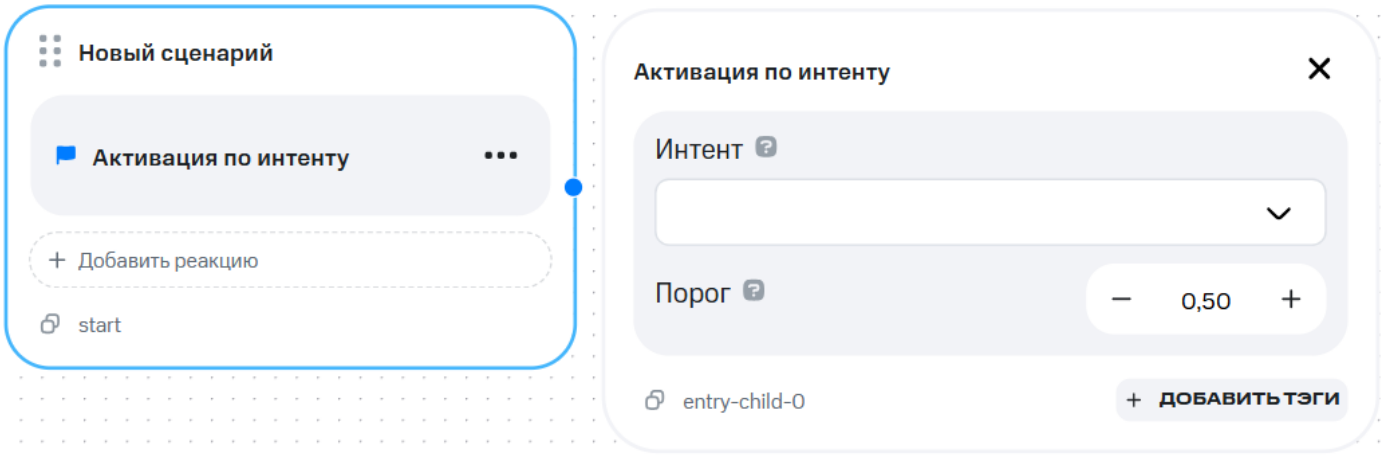


Активация по регулярному выражению. Для активации стейта происходит поиск соответствия запроса пользователя регулярному выражению. Укажите его в поле **Регулярное выражение**.

При написании регулярного выражения используйте стандартный синтаксис regexp.

Интент

Кандидаты-интенты для активации не вычисляются автоматически. Чтобы получить интенты для активации, в сценарии препроцессинга напишите скрипт для определения кандидатов-интентов.



Активация по интенту. Чтобы активировать стейт, выполняется запрос к классификатору для определения интента по запросу пользователя.

Имя классификатора указывается на вкладке с настройками версии в поле Классификатор. Интент должен соответствовать одному из существующих интентов в подключенном классификаторе.

Заполните поля:

- **Интент** – название интента;
- **Порог** – значение, при достижении которого совпадающий со значением в данном блоке интент, полученный от классификатора, рассматривается в качестве активатора сценария. Значение сравнивается с параметром **score**, которое также возвращает классификатор вместе с интентом. По умолчанию 0,5.

Чтобы активационный блок **Интент** сработал, предварительно выполните:

1. Убедитесь, что для версии бота включено использование препроцессинга.
2. В сценарий препроцессинга добавьте блок **Скрипт**.
3. В асинхронной функции handler(context: Context) с помощью предопределенной функции **predict_intent** получите наиболее вероятные интенты для активации запроса.

   Формат функции:

   ```
   context.nlu.predict_intent(message, top_n=1)
   ```

   Где:

   message – сообщение, для которого нужно определить наиболее подходящих кандидатов;
   top_n – количество наиболее вероятных кандидатов. Например, в ситуациях, когда одному интенту соответствует сразу несколько блоков активации в разных сценариях, можно указать top_n=1. Это обеспечит наличие только одного подходящего кандидата в списке наиболее вероятных.
   Полученные интенты сохраняются в виде массива в переменную **context.nlu.intents**. Элементы массива упорядочены по вероятности и по приоритету иерархии. В переменной **context.nlu.raw_intents** формируется массив из top_n интентов, состоящий из имен интентов и коэффициентов вероятности (score). Для обратной совместимости имя первого интента из массива **context.nlu.raw_intents** и его score сохраняются в переменные **context.system.sure_topic** и **context.system.topic_score** соответственно.

   В результате сработает активация по интенту. При необходимости переопределите кандидатов вручную.

   Пример скрипта:

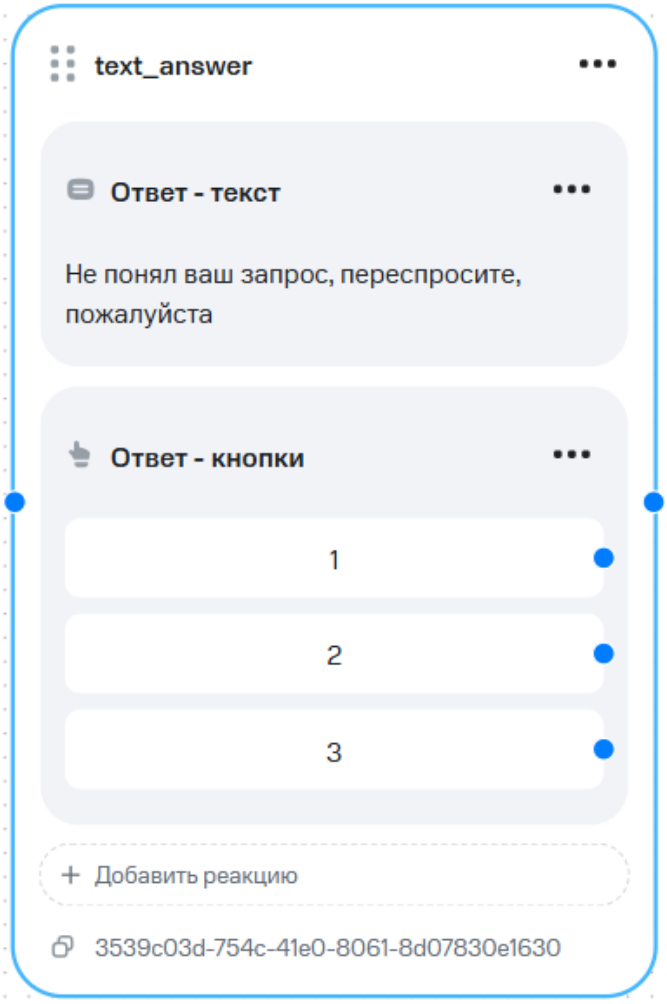
```
async def handler(context: Context) -> None:
    message = context.system.last_user_message.strip()
    # Получение top_n кандидатов для активации по интенту
    await context.nlu.predict_intent(message, top_n=1)

    # Ручное переопределение списка кандидатов для активации по регулярному
     выражению
    if message == "message":
        context.nlu.intents = []
        context.nlu.matches = [
            ActivationCandidate(
                scenario_id=1,
                edge=MatchEdge(value="scenario_1", target_node_id="c34dcad6-5343-4f06-
     a55d-bf68424b28b1"),
                score=1.0,
            )
        ]
```

Блоки реакции

После того, как срабатывает активационный блок в сценарии, выполняются блоки реакции. Например, бот может отправить пользователю текст, кнопки для выбора вариантов ответов или автоматически перейти к другому сценарию.

Для настройки логики можно использовать зарезервированные переменные, а также переменные, созданные в сценарии. При этом в качестве префикса указывайте для них области видимости.

Чтобы добавить блок в рабочую область, перенесите его с панели **Блоки конструктора**. При этом в один блок (стейт, ноду) можно добавить несколько реакционных блоков. Блоки будут выполняться последовательно.



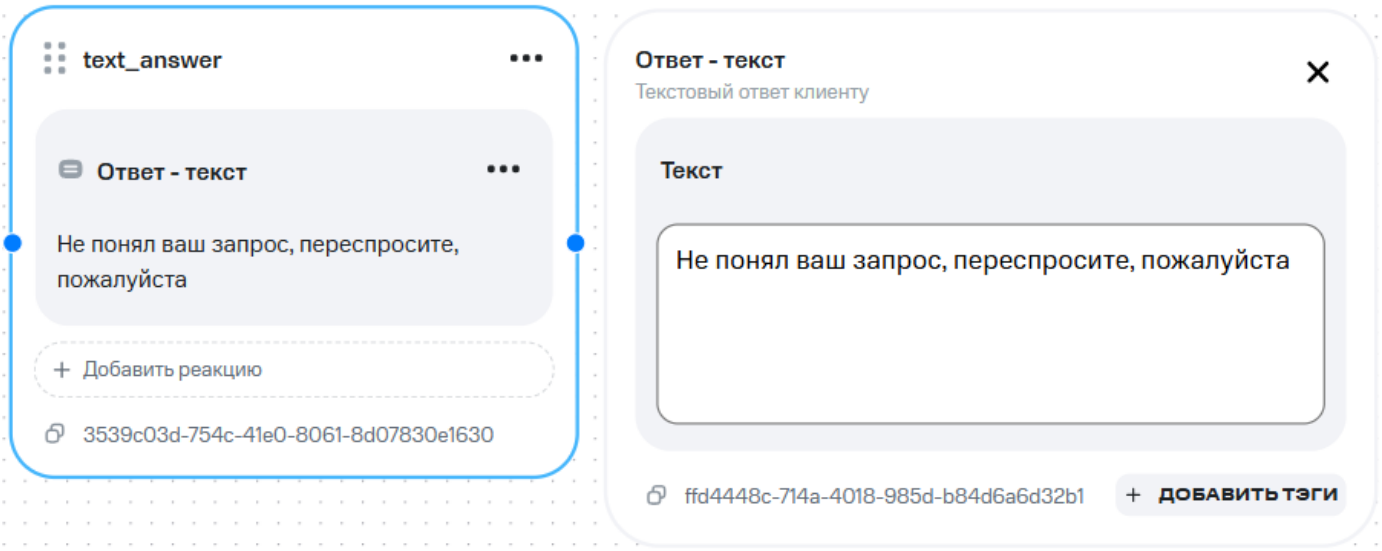
Для сценария препроцессинга набор блоков ограничен.

При заполнении блоков вы можете задействовать функцию **normalize**, которая выполняет лемматизацию слов – преобразует слова в их начальные, базовые формы (леммы).
Формат заполнения: {{normalize(Имя переменной)}}
Примеры лемматизации слов:

- «бежал», «бежит», «бегущий» – «бежать»;
- «стола», «столов», «столами» – «стол».

Например, в сообщении клиента передано местоположение – «к неврологу». Чтобы сформировать название кнопки, требуется наименование города в начальной форме – «невролог». Для этого в текстовом поле укажите переменную **last_user_message** с добавлением функции **normalize**: {{normalize(system.last_user_message)}}. Результатом работы блока будет кнопка с лемматизированным сообщением клиента – «невролог».
Учитывайте, что в начальные формы возвращаются все слова в сообщении.
При лемматизации регистр всех букв в тексте меняется на нижний.

Текст



Текстовый ответ клиенту.

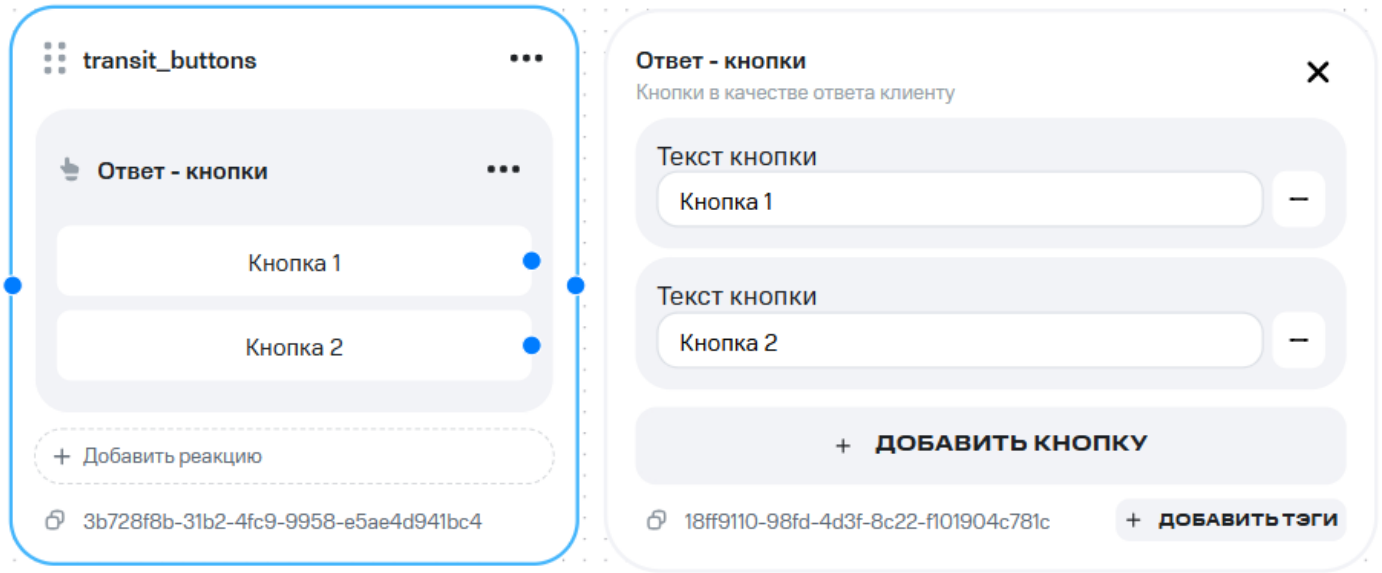
Заполните поле **Текст** – ответ, который нужно выводить в чате с пользователем. При необходимости в поле **Текст** можно использовать переменные, заключенные в двойные фигурные скобки **{{}}**.

Пример:

«Количество символов в вашем сообщении превышает {{max}}, оно составляет {{system.last_user_message_length}}. Попробуйте перефразировать»

В данном случае **max** и **last_user_message_length** – имена зарезервированных переменных или переменных, которые будут сохранены в контекст пользователя другими реакционными блоками в процессе диалога. При выполнении данного реакционного блока вместо названий переменных подставляются их значения.

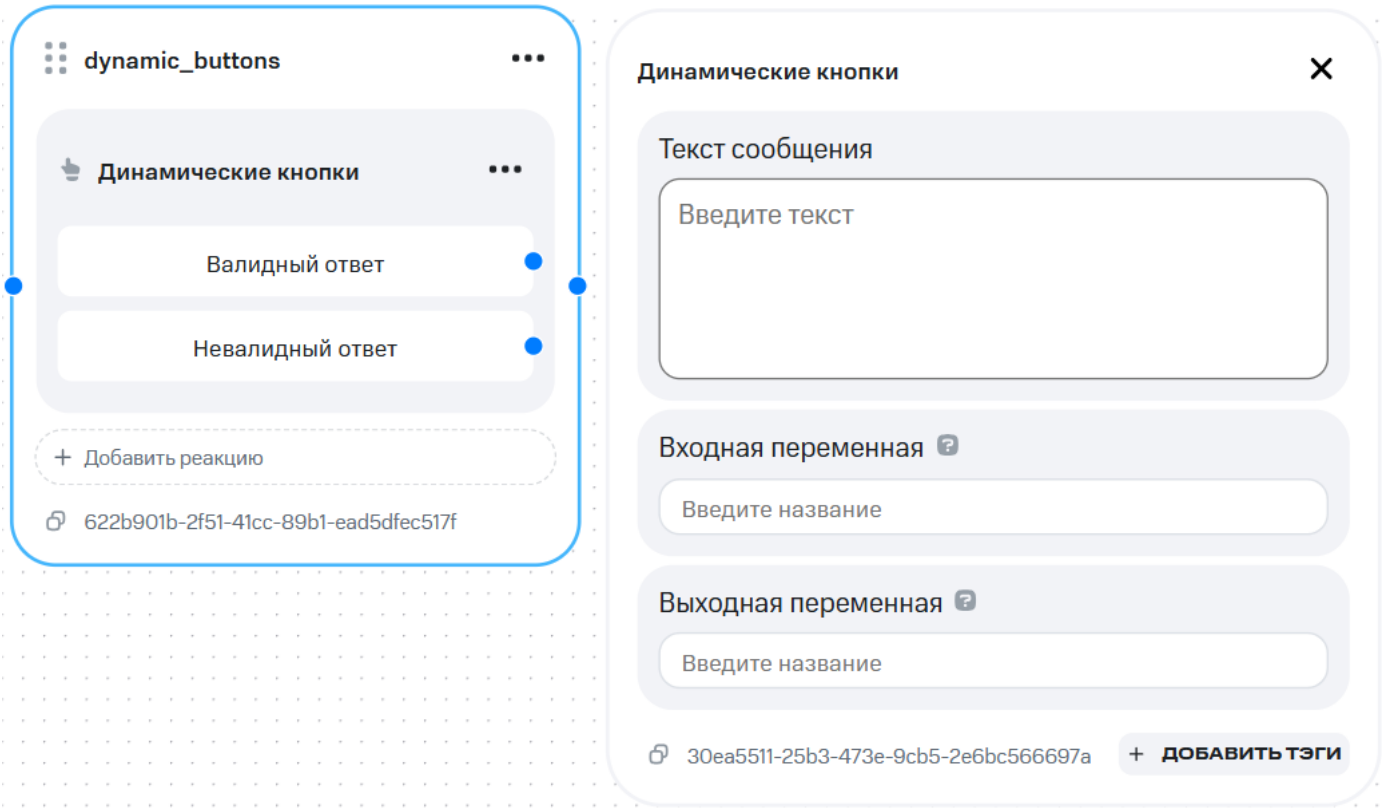
Кнопки



Кнопки в качестве ответа клиенту. Значения задаются в сценарии. В поле **Текст кнопки** укажите название кнопки, которое нужно отобразить в чате для пользователя. Нажмите **Добавить кнопку**, чтобы добавить необходимое количество кнопок.

Если нужны кнопки, которые формируются автоматически, воспользуйтесь блоком **Динамические кнопки**.

Динамические кнопки



Кнопки, которые автоматически формируются на основе полученных данных в ходе выполнения сценария.

При разработке сценария рекомендуется самостоятельно проверять количество кнопок, иначе их может сформироваться большое количество.

Кликните на блок и заполните поля:

- **Текст сообщения** – текст, который должен отображаться перед списком кнопок;
- **Входная переменная** – переменная, из которой нужно формировать список кнопок для отображения. Обязательное поле для заполнения.
  Формат значения переменной:

```
{
 "<Кнопка 1>": <Значение>,
 "<Кнопка 2>": <Значение>,
 ...
 "<Кнопка N>": <Значение>
}
```

Где:
Кнопка 1, **Кнопка 2**, **Кнопка N** – названия кнопки;
Значение – значение, которое сохраняется при выборе кнопки.

Пример заполнения входной переменной:

```
{
  "Клиника 1": 10,
  "Клиника 2": 12,
  "Другое": -1
}
```

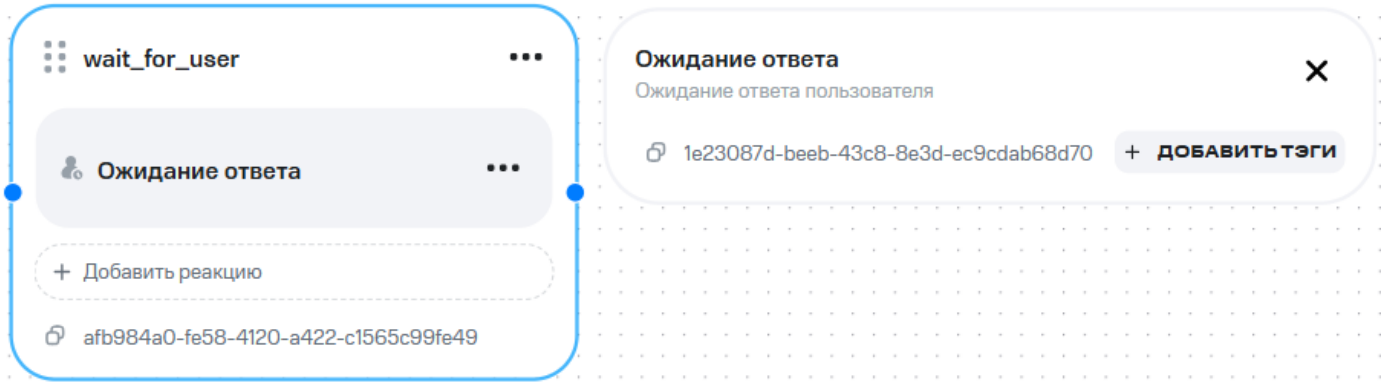
- **Выходная переменная** – переменная, в которую сохраняется значение выбранной пользователем кнопки. Например, если нажали на кнопку **Клиника 1**, то в выходную переменную сохраняется значение **10**. Указанную переменную можно использовать в следующих блоках сценария.

Добавьте связи в зависимости от ответа:

- **Валидный ответ** – связь с реакционными блоками, которые должны выполниться, если пользователь выберет одну из динамических кнопок;
- **Невалидный ответ** – связь с реакционными блоками, которые должны выполниться, если от пользователя придет отличное от названий динамических кнопок значение.

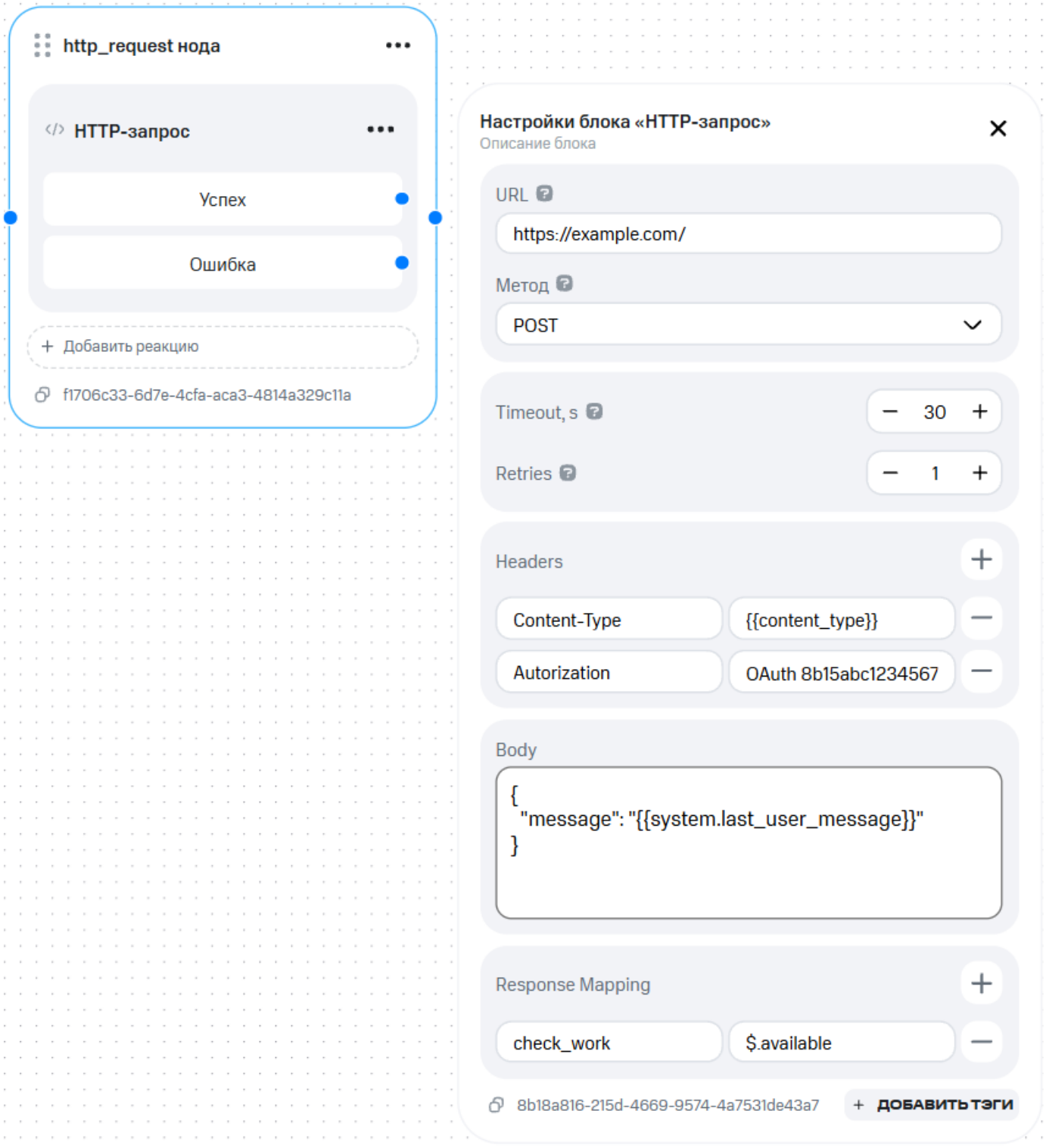
Если получен валидный или невалидный ответ, связь для которого не добавлена, то после блока **Скрипт** выполняется переход к следующему реакционному блоку.

Ожидание ответа



Блок ожидания ответа от пользователя. Используйте его, если нужно дождаться ответа, прежде чем переходить к следующим блокам сценария.

HTTP-запрос



Выполнение HTTP-запроса.

В параметрах блока укажите:

- **URL** – конечная точка запроса (эндпоинт);
- **Метод** – метод запроса, например, POST. Выберите HTTP-метод из выпадающего списка;
- **Таймаут, секунды** – максимальное время ожидания ответа на запрос в секундах. Если запрос завершается по тайм-ауту, то выполняется переход к стейту обработки невалидного ответа;

- **Retries** – количество попыток повторить неуспешный запрос. По умолчанию **0** – неуспешный запрос не повторяется. Чтобы повторить запрос один раз, укажите 1;
- **Headers** – заголовки запроса. Включает в себя название заголовка (key) и значение (value). По кнопкам + и — вы можете добавить или удалить строки для заполнения;
- **Body** – тело запроса. Должно соответствовать формату, указанному в поле **Body format**. Пример:

```
{
  "message": "{{system.last_user_message}}"
}
```

, где **last_user_message** – имя зарезервированной переменной или переменной, которая будет сохранена в контекст пользователя другим реакционным блоком в процессе диалога. При выполнении данного реакционного блока вместо имени переменной подставляется ее значение;

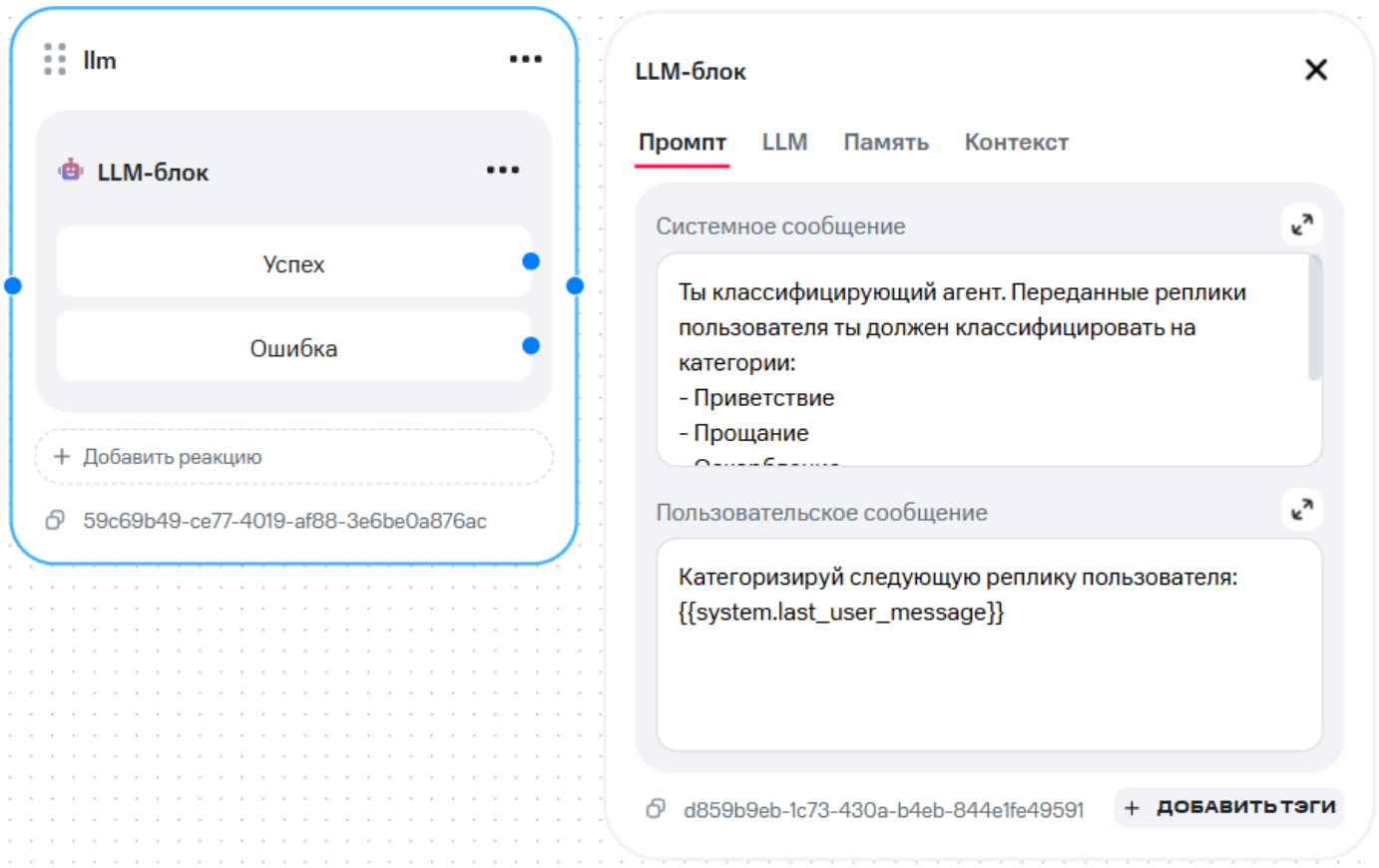
- **Response Mapping** – маппинг значений параметров, полученных в ответе на запрос, на переменные движка/контекста пользователя. Включает в себя название переменной (key) и описание пути для извлечения и записи значения (value). Переменные сохраняются в контекст текущего диалога. По кнопкам + и — вы можете добавить или удалить строки для заполнения. В значениях допускается использовать предикаты:
  - all – выбрать и записать все элементы массива;
  - keep_null – если в элементе массива в каком-либо ключе пришло значение null, то записать null в переменную, указанную в key;
  - all&keep_null – выполнение обоих предикатов.
    Доступны опции:
  - $response.<...> или $response.body.<...> – извлечение значений переменных из body ответа;
  - $response.headers.<...> – извлечение значений переменных из headers ответа;
  - $response.status_code – извлечение http-кода ответа.

В полях **URL**, **Header**, **Body** вы можете использовать переменные, заключенные в двойные фигурные скобки: **{{Имя переменной}}**.

Добавьте связи для блока:

- **Успех** – с блоком реакции, к которому нужно перейти, если запрос выполнен успешно;
- **Ошибка** – с блоком реакции, к которому нужно перейти в случае неудачи при выполнении запроса.

LLM



Блок позволяет решать задачи, связанные с пониманием естественного языка и генерации ответа. Примеры задач: перевод текста, генерация описания какого-либо объекта, ответ на конкретный вопрос клиента, генерация краткого содержания длинного текста. Результат работы большой языковой модели записывается в переменную, которую можно использовать в следующих реакционных блоках.

> Для решения более сложных задач с применением дополнительных инструментов используйте блок **AI-агент**.

Чтобы настроить отправку запроса в LLM:

1. Заполните промпт для модели:
   - **Системное сообщение** – инструкция для модели, в которой определяется роль, правила и цели модели, в том числе возможности и ограничения;
   - **Пользовательское сообщение** – контекст, в котором может содержаться входящее сообщение клиента или другие данные, которые модель должна обработать.

     > В промпте вы можете использовать переменные из текущего проекта, заключенные в двойные фигурные скобки: **{{Имя переменной}}**.
2. На вкладке **LLM** заполните настройки подключаемой модели:

Название параметра	**Описание**
URL	Ссылка на модель
Token	Токен доступа к модели. Необязательный параметр
Модель	Имя модели, к которой выполняется запрос. Поддерживаются OpenAI-совместимые API-модели
Timeout	Максимальное время ожидания ответа от модели за одну попытку. Указывается в секундах

Название параметра	Описание
Retries	Количество попыток получить ответ от модели, если при обращении к ней возникает ошибка
Max number of tokens	Максимальное количество токенов, сгенерированное LLM в ответ на запрос
Sampling temperature	Уровень случайности в ответах, генерируемых большой языковой моделью. Высокие температуры приводят к более разнообразным и креативным результатам, в то время как низкие температуры – к более консервативным и предсказуемым реакциям. Возможные значения: от 0.0 до 2.0
Top K	Количество наиболее вероятных токенов, которые модель учитывает при генерации текста. Чем ниже значение **Top K**, тем более предсказуемым и повторяющимся будет ответ модели. Возможные значения: от 1 до 10000
Top P	Пороговая вероятность включения токенов в набор кандидатов, используемый LLM для генерации выходных данных. Под токеном понимается минимальная единица текста, с которой способна работать языковая модель. Низкие значения параметра **Top P** приводят к более точным и основанным на фактах ответам от LLM, тогда как более высокие значения увеличивают случайность и разнообразие в сгенерированном ответе. Возможные значения: от 0.1 до 1.0
Frequency penalty	Штраф за повторение токенов в тексте в зависимости от частоты появления. Токены, которые встречаются в тексте чаще, с меньшей вероятностью будут использоваться ИИ снова. Параметр позволяет уменьшить частоту повторений. Возможные значения: от -2 до 2
Presence penalty	Фиксированный штраф за повторение токенов, независимо от частоты появления. Возможные значения: от -2 до 2

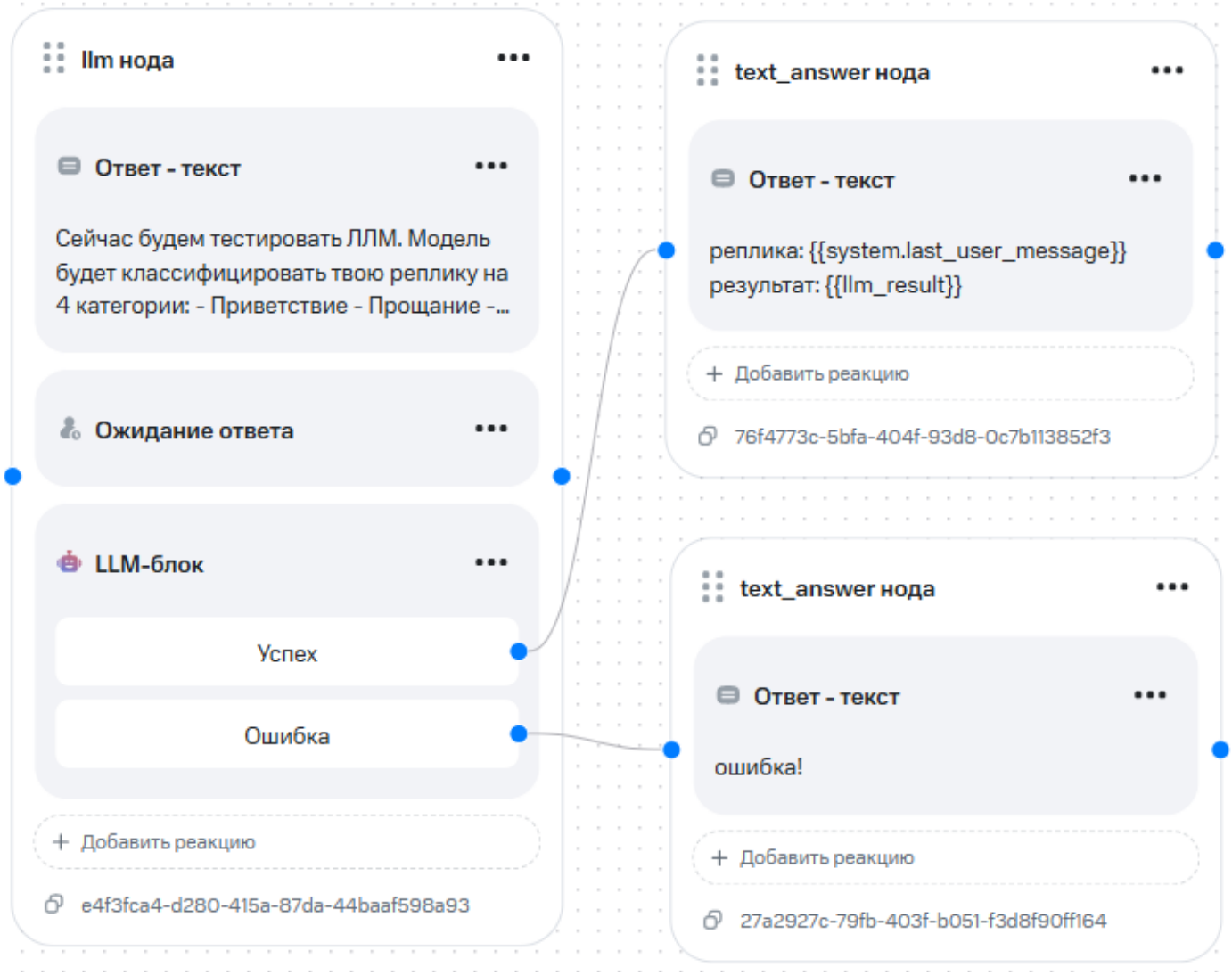
Если в ваш комплект поставки входит большая языковая модель Cotype Pro, то на вкладке **LLM** вы можете указать ее. Для этого заполните поля:

- **URL** – введите адрес до модели в формате: <Путь до модели>:8080/v1/, например, http://cotype.dev-mars:8000/v1. Адрес по умолчанию можно посмотреть в файле values,yaml сервиса rag-manager в параметре **LLM_URL**;
- **Модель** – имя используемой версии Cotype Pro, например cotype_pro_2_mars.

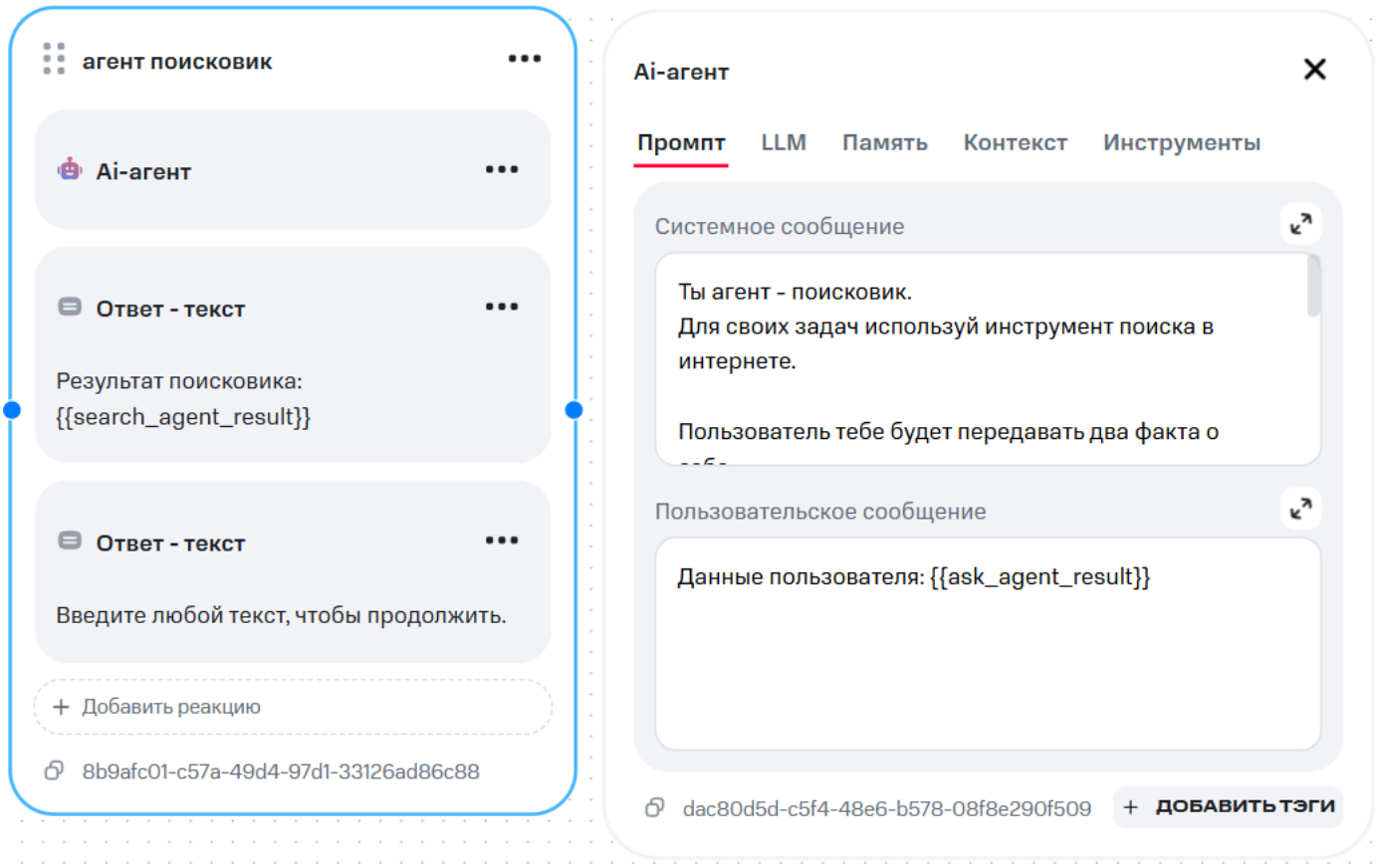
3. На вкладке **Память** настройте контекст для модели. Установите ползунок **Итерация взаимодействия** на том значении, которое обозначает количество последних итераций взаимодействия клиента и бота. Возможные значения: от 0 до 100.
4. На вкладке **Контекст** в поле **Переменная** укажите имя переменной. В эту переменную сохраняется результат работы LLM. Ее можно использовать в следующих блоках сценария. Например, добавьте блок **Текст** для отображения результата работы LLM в чате с пользователем.
5. Добавьте связи для блока:
    - **Успех** – переход в реакционный блок, если блок LLM выполнен успешно;

- **Ошибка** – переход в реакционный блок, если во время выполнения блока LLM возникла ошибка.

Пример создания блока LLM:



AI-агент



Блок для отправки запроса в большую языковую модель (LLM) и генерации ответа с подключением дополнительных инструментов. Инструменты – это вспомогательные ресурсы, которые позволяют агенту решать задачи за пределами возможностей LLM. Например, искать информацию в интернете, выполнять математические вычисления, сохранять информацию в базе данных и т.д.

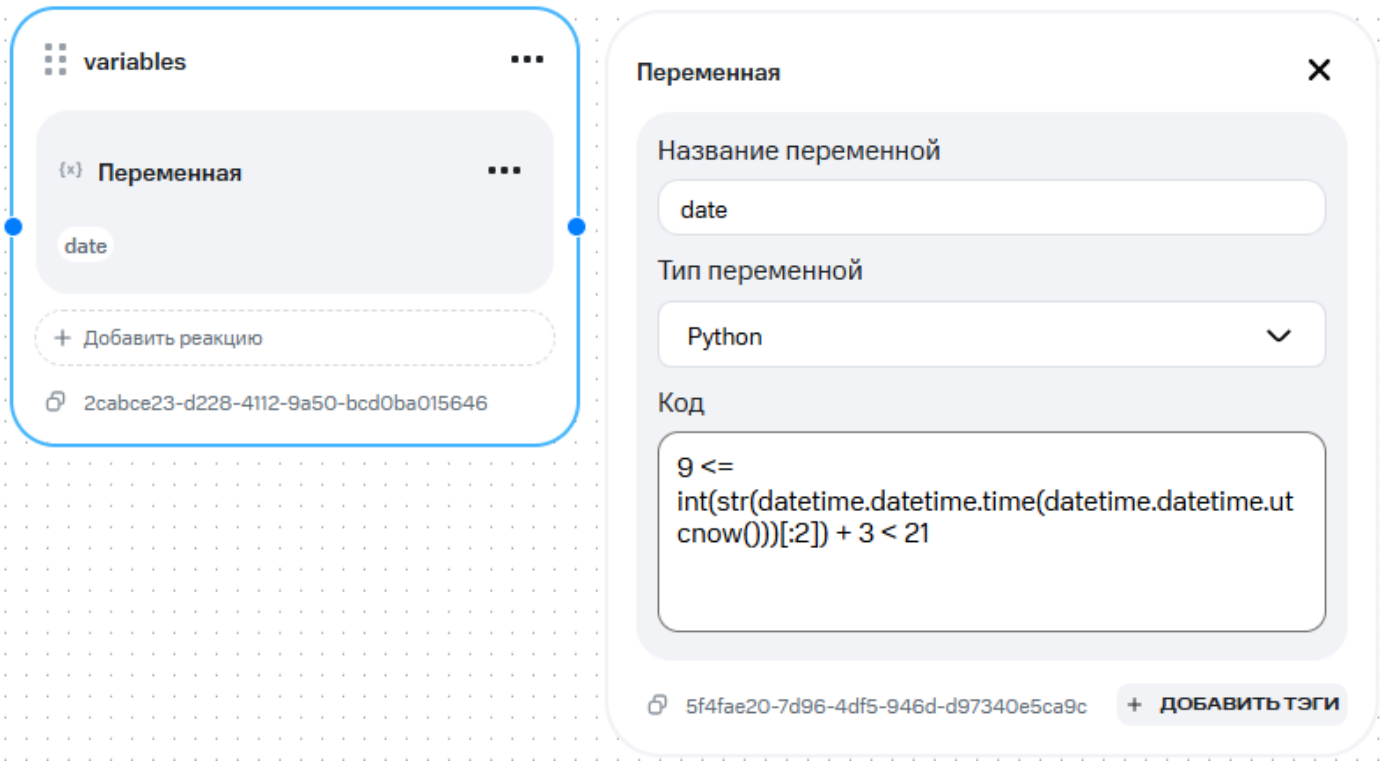
Если проект с блоком **AI-агент** подключен к каналу с типом HTTP, то сгенерированные агентом уточняющие вопросы в чате с пользователем приходят в стриминговом режиме. Это значит, что текст отправляется чанками, и задержка времени на формирование полного ответа не происходит. Для других типов каналов и в тестовом виджете стриминговый режим не поддерживается.

Если предполагается только работа с естественным языком, то вместо блока **AI-агент** используйте блок **LLM**.

1. Заполните поля на вкладках **Промпт**, **LLM**, **Память**, **Контекст**. Подробнее см. в описании блока **LLM**.
2. На вкладке **Инструменты** введите путь до **MCP-сервера**, который предоставляет инструменты, которые агент может использовать для получения ответа на запрос.

   Инструменты к агенту можно подключить только с помощью MCP-серверов. Поддерживается только транспортный протокол Streamable HTTP.

Переменная



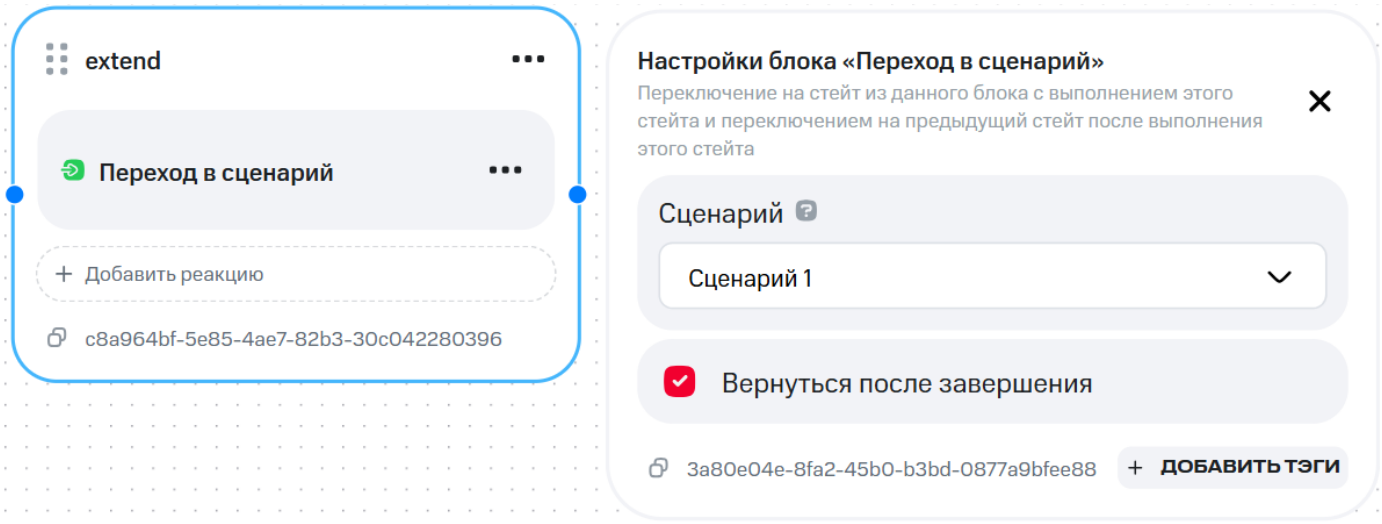
Создание и сохранение переменных в диалог пользователя.

Чтобы добавить переменную, нажмите на блок **Переменная** и в открывшемся окне заполните поля:

- **Название переменной**;
- **Тип переменной**. Выберите значение из списка. Доступные типы:
  - **Constant** – константа. Значение по умолчанию. В качестве значения переменной используется то, что указано в поле **Value**;
  - **Python** – код на языке Python. В этом случае значение поля **Value** будет отправляться на вычисление в интерпретатор кода;
  - **Regexp** – регулярное выражение. В качестве значения используется результат вычисления регулярного выражения;
  - **Regexp map** – массив регулярных выражений. В качестве значения переменной используется результат первого сработавшего регулярного выражения.

    Если тип переменной **Regexp** или **Regexp map** и после выполнения регулярных выражений значение переменной не определено, то в качестве него используется **null**. Это происходит, если запрос пользователя не подходит ни под одно регулярное выражение.
- **Код** – значение переменной. Может быть задано константой или кодом на языке Python. Пример значения:
  9 <= int(str(datetime.datetime.time(datetime.datetime.utcnow()))[:2]) + 3 < 21

Переход в сценарий

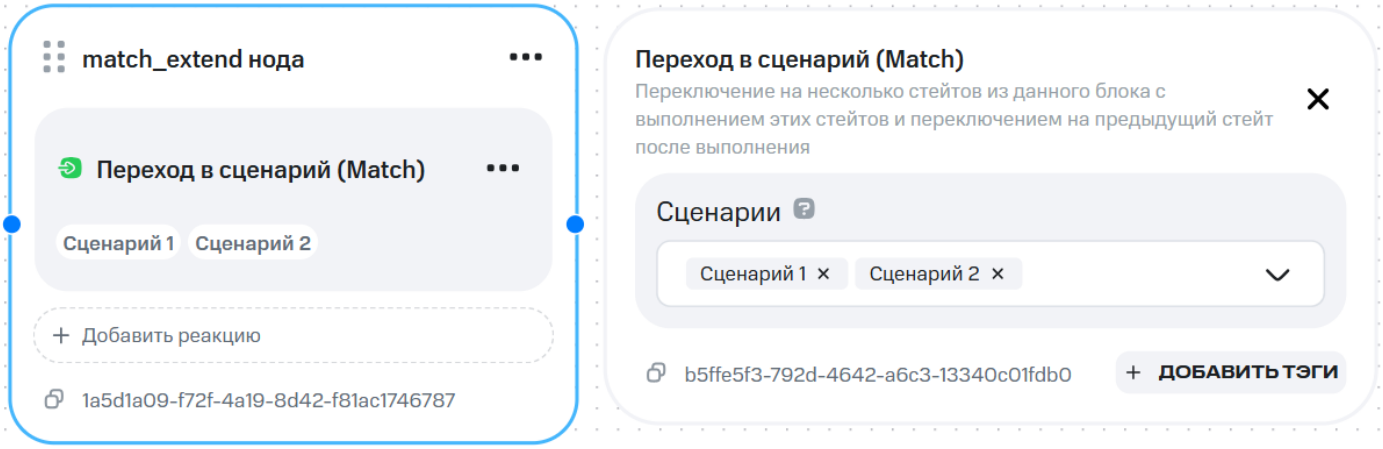


Вызов другого сценария из данного блока.
В выпадающем списке **Сценарий** отображается список всех сценариев бота. Выберите тот, к которому нужно перейти после выполнения текущего блока. Если по завершению выбранного сценария нужно выполнить переход к исходному, установите флажок **Вернуться после завершения**. В этом случае сценарий продолжится со следующего блока.

Если нужно перейти к конкретной ноде другого сценария, то используйте метод контекста **context.defer_jump_to** в блоке **Скрипт**.

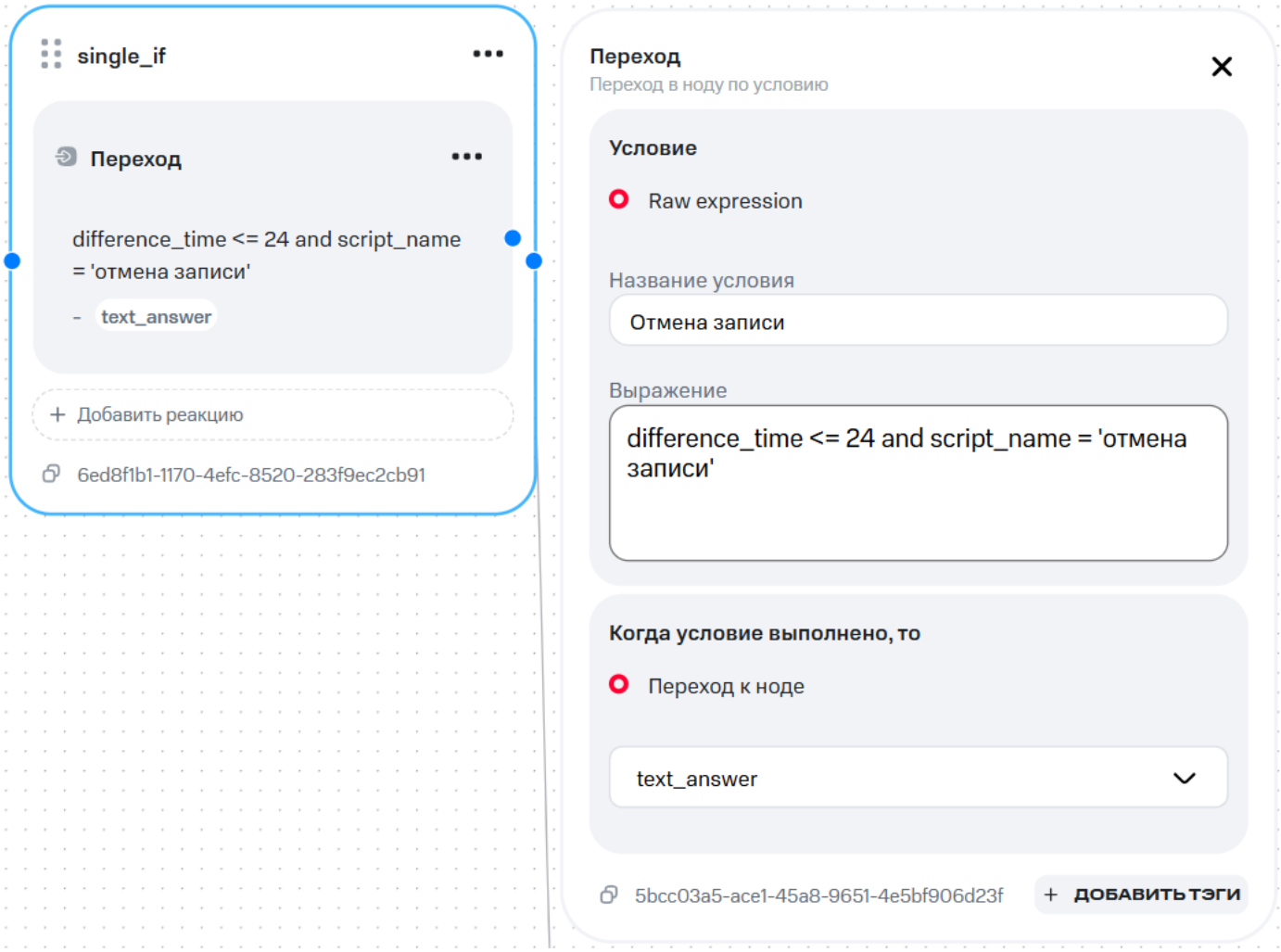
> Если блок **Переход в сценарий** добавлен в сценарий препроцессинга и в нем установлен флажок **Вернуться после завершения**, то сценарий, к которому нужно перейти, также выполняется в рамках препроцессинга. Проверьте, что в нем используются только допустимые блоки: **HTTP-запрос**, **Переменная**, **Переход в сценарий**, **Переход в сценарий (Match)**, **Условие**, **Скрипт**. При попытке выполнить недопустимые блоки возникает ошибка.

Переход в сценарий (Match)



Вызов сценария. Для выбора указывается несколько сценариев. По активационным блокам в них движок определяет, в какой сценарий нужно перейти после выполнения текущего блока. По завершению выбранного сценария выполнится переход к исходному сценарию, он продолжится со следующего блока.

Условие



Условие выбора стейта для перехода. Установите маркер **Raw expression**, чтобы переходить к выбранной ноде (стейту) при выполнении условия. Если маркер не установлен, то переход будет выполняться всегда.

Если выбрано **Raw expression**, то заполните поля:

- **Название условия**;
- **Выражение**. В поле укажите условие, которое нужно проверить. При заполнении можно использовать переменные. Например, если нужно проверять последнее сообщение пользователя, укажите переменную **last_user_message** – запрос пользователя или текст кнопки.

При заполнении выражения используйте:

- операторы:
  = – сравнение на равенство;
  != – сравнение на неравенство;
  ( ) – выделение конструкций;
  < – меньше;
  <= – меньше либо равно;
  == – сравнение на равенство;
  > – больше;
  >= – больше либо равно;
  + – сложение;
  - – вычитание;
  ***** – умножение;

/ – деление;
is – проверка совместимости результата выражения с указанным типом;
and – логическое «И»;
or – логическое «ИЛИ»;
not – логическое «НЕ»;
contains – проверка на наличие искомой подстроки в исходной строке;
not_contains – проверка на отсутствие искомой подстроки в исходной строке;
matches – проверка на наличие строки, представленной в виде регулярного выражения.

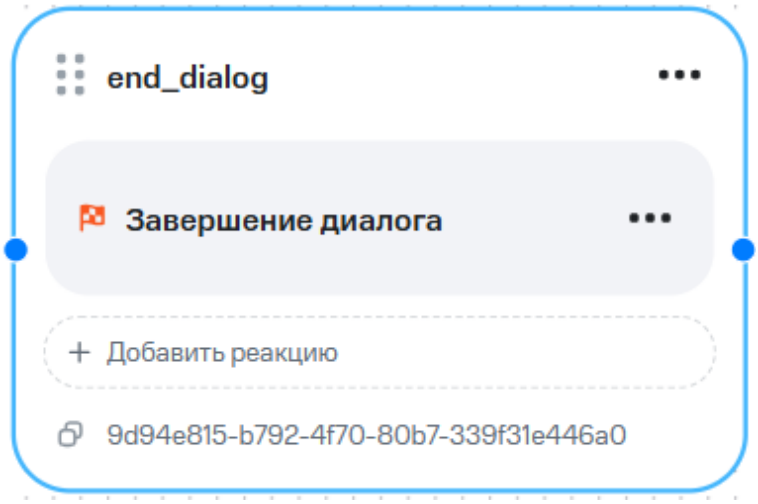
Пример 1. difference_time <= 24 and script_name = 'отмена записи'
Пример 2. system.united_user_messages matches 'парков(ка|ки|ок)|стоян(ка|ки|ок)|парковочн(ое|ые)'
Пример 3. (system.last_user_message contains 'Купить') or (system.last_user_message contains '1') or (system.last_user_message contains 'дай')

- константы:
  null – значение переменной до инициализации;
  true – «ИСТИНА»;
  false – «ЛОЖЬ»;
  undefined – отсутствие значения
- одинарные кавычки – для указания значений строк. Строка может содержать переменные, заключенные в **{{}}**. Например, "system.last_user_message = '{{another_var}}'";
- переменные, элементы массива, функцию map.
  Пример:
  sarr[1] == "bar"
  oarr[666].field1 == "foo"

В блоке **Когда условие выполнено, то** установите маркер **Переход к ноде** и укажите стейт (ноду), к которому нужно перейти, если условие истинно.

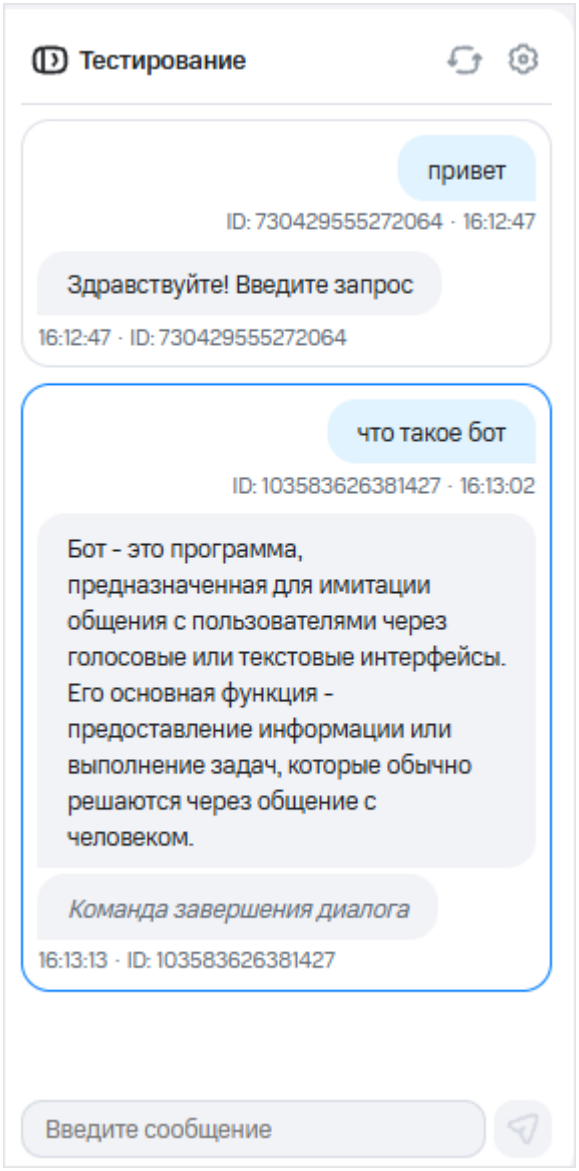
Для ситуаций, когда условие ложно или нужно задать другое условие, добавьте в стейт (ноду) дополнительный блок **Условие**.

Завершение диалога

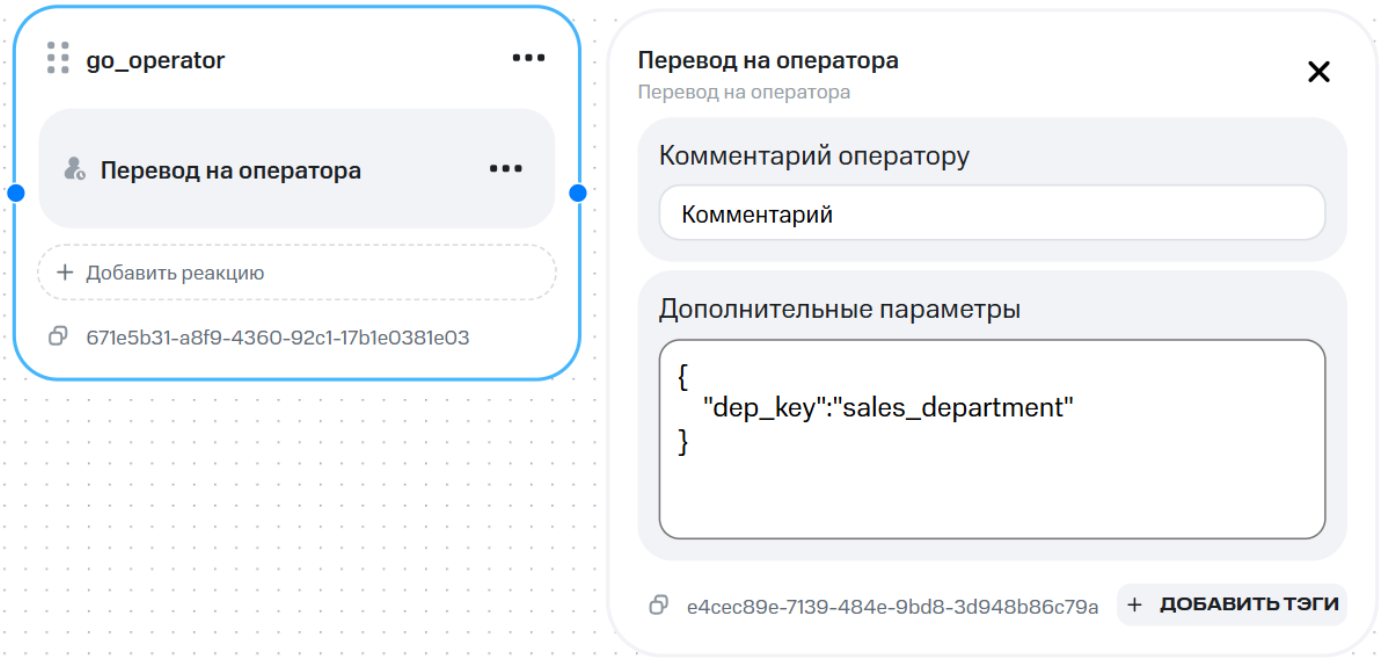


Завершение сценария. Добавьте блок и связь с ним, чтобы закончить диалог.

В тестовом виджете для этого блока отображается сообщение: «Команда завершения диалога»:



Перевод на оператора



Перевод диалога на оператора.

В поле **Дополнительные параметры** в формате JSON можно указать параметры:

- **operator_id** – идентификатор оператора, на которого нужно перевести диалог;
- **dep_key** – идентификатор отдела, в который нужно перевести диалог.

Если указаны оба параметра, то диалог переводится на оператора. В случае ошибки выполняется перевод на отдел. Если снова возникает ошибка, то диалог переводится в общую очередь.

Если параметры не указаны, то также осуществляется перевод в общую очередь.

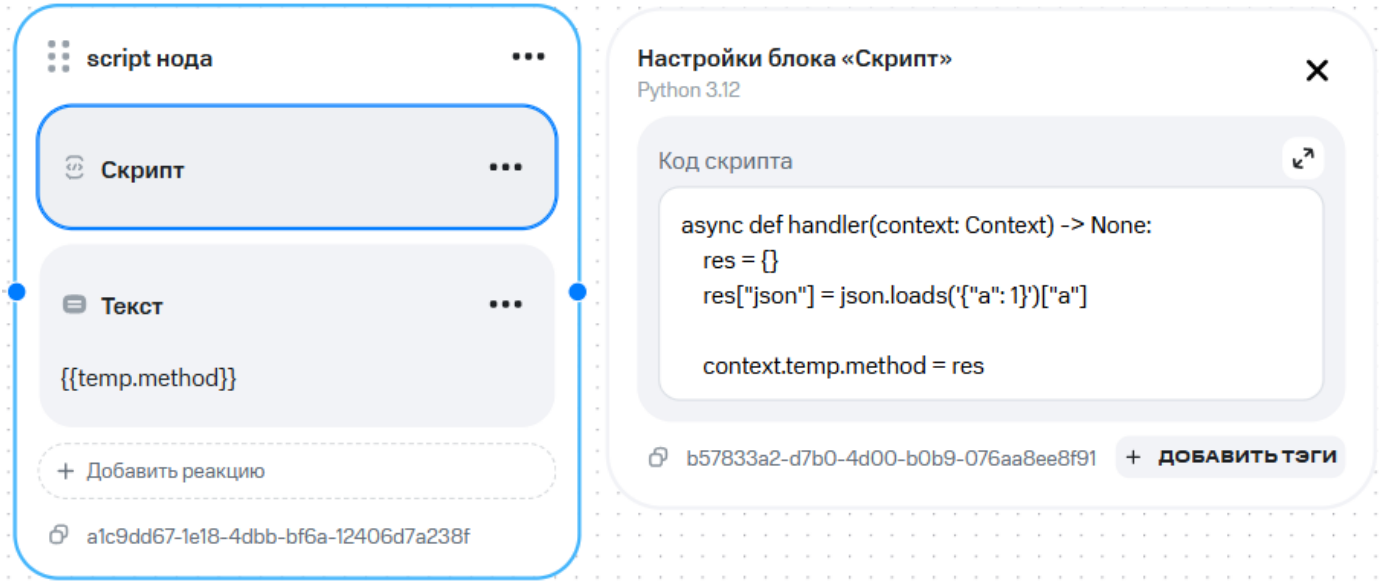
Пример значения:

```
{
   "dep_key":"sales_department"
}
```

В поле **Комментарий оператору** можно задать сообщение для оператора.

После блока **Перевод на оператора** сценарий завершается. Блоки, следующие за текущим, не выполняются.

Скрипт



Выполнение произвольного скрипта на языке Python. В скрипте должна быть определена асинхронная функция handler(context: Context), которая принимает контекст типа dict. В контексте содержатся переменные пользователя, а также доступны зарезервированные переменные движка и вызов предопределенных функций. Контекст, переданный в функцию, можно изменить. В результате заданные переменные становятся доступными в следующих блоках сценария.

При обращении к переменным указывайте области видимости в формате: **context.<scope>.<Имя переменной>**
Пример: context.system.last_user_message

Для удобства в код скрипта добавлен шаблон, который содержит обязательную функцию обработчика и вспомогательные функции. Заполните код обязательной функции, при необходимости скорректируйте или удалите вспомогательные.

В программном коде вы можете использовать функции следующих модулей: array, base64, binascii, bisect, calendar, cmath, collections, contextlib, contextvars, copy, csv, dataclasses, decimal, email, encodings, enum, fractions, functools, hashlib, heapq, hmac, html, ipaddress, itertools, json, keyword, math, numbers, pprint, quopri, re, reprlib, secrets, shlex, statistics, string, textwrap, tomllib, typing, unicodedata, uuid, zoneinfo, datetime. Они уже подключены, и дополнительно использовать оператор import не нужно.

Подробнее о функциях модулей см. статью The Python Standard Library в официальной документации Python.

Пример

В блоке **Скрипт** задан код:

```
async def handler(context: Context) -> None:
    res = {}
    res["json"] = json.loads('{"a": 1}')["a"]
    context.temp.method = res
```

В результате выполнения кода создается переменная **temp.method**, которая становится доступна в следующем реакционном блоке, например в блоке **Текст**.

Предопределенные функции

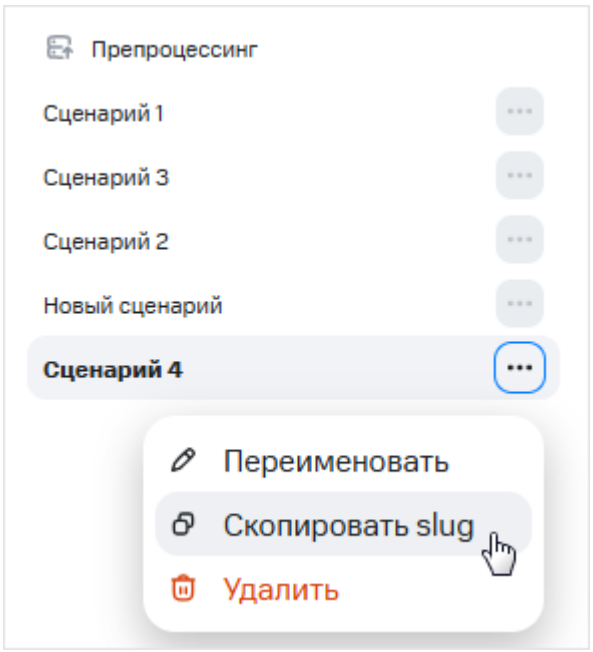
В блоке **Скрипт** вы можете использовать предопределенные функции:

- **defer_jump_to** – перейти к сценарию или ноде;
- **predict_intent** – получить наиболее вероятные интенты для запроса.

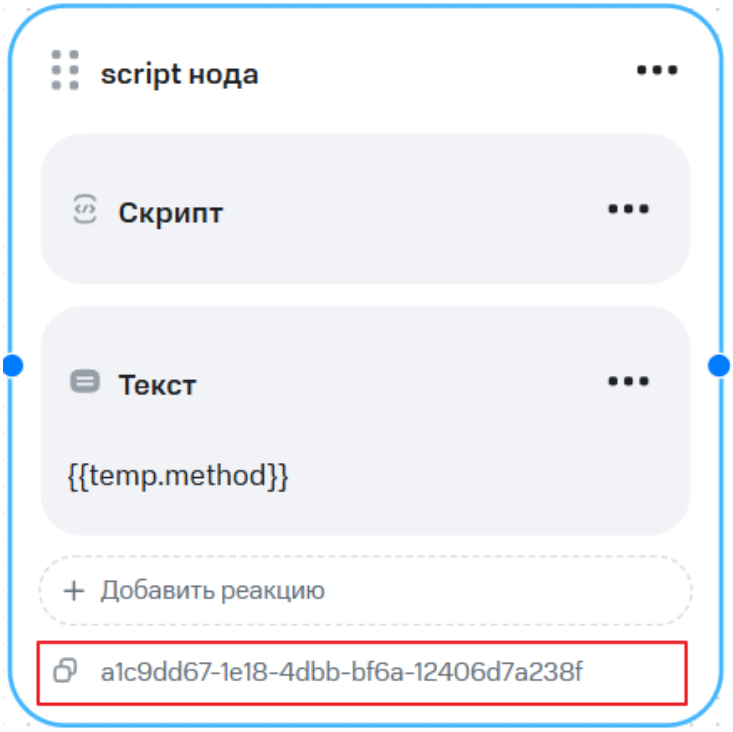
defer_jump_to – перейти к сценарию или ноде

Из блока **Скрипт** можно выполнить переход к другому сценарию. Для этого:

1. Скопируйте идентификатор сценария или ноды, к которым нужно перейти.
   ИД сценария:



   ИД ноды:



2. В сценарии, из которого нужно перейти, в блок **Скрипт** добавьте функцию defer_jump_to:

```
def handler(context: Context) -> None:
    context.defer_jump_to(
        scenario_slug="<scenario_slug>",
        node_id="<node_id>",
        need_return=<True/false>,
    )
```

Где:

scenario_slug – ранее скопированный slug сценария, к которому нужно перейти после выполнения скрипта. Если нужно перейти к конкретной ноде, то переменную не указывайте.

> Чтобы обратиться к сценарию, используйте его slug вместо идентификатора, так как слаг остается постоянным, а ИД может измениться, например при импорте сценария.

node_id – идентификатор ноды, к которой нужно перейти. Если нужно выполнить переход по slug, то переменную не указывайте;
need_return – признак того, что после выполнения сценария нужно вернуться в исходную ноду. По умолчанию **False** – возвращаться в исходную ноду не нужно. Если выполняется переход без возврата из сценария препроцессинга, то препроцессинг прерывается.

> Если функция defer_jump_to описана несколько раз, то в результате переход будет выполнен к сценарию или ноде, указанным в последней функции.

predict_intent – получить кандидатов для активации по интенту

Формат функции:

```
context.nlu.predict_intent(message, top_n=1)
```

Где:
message – сообщение, для которого нужно определить наиболее подходящих кандидатов;
top_n – количество наиболее вероятных кандидатов. Например, в ситуациях, когда одному интенту соответствует сразу несколько блоков активации в разных сценариях, можно указать top_n=1. Это обеспечит наличие только одного подходящего кандидата в списке наиболее вероятных.
Полученные интенты сохраняются в виде массива в переменную **context.nlu.intents**. Элементы массива упорядочены по вероятности и по приоритету иерархии. В переменной **context.nlu.raw_intents** формируется массив из top_n интентов, состоящий из имен интентов и коэффициентов вероятности (score). Для обратной совместимости имя первого интента из массива **context.nlu.raw_intents** и его score сохраняются в переменные **context.system.sure_topic** и **context.system.topic_score** соответственно.

Подробнее об использовании функции **predict_intent** см. в описании блока **Интент**, раздел «Блоки активации».

Области видимости и времени жизни переменных

Созданную пользователем или зарезервированную системой переменную можно определить в одну из областей видимости и времени жизни – scope (скоуп).

Если переменная используется в сценарии и префикс для нее не указан, система автоматически относит переменную в session-скоуп и присваивает префикс **session**. Поэтому при обращении к зарезервированным переменным указывайте префикс.
Если имя переменной совпадает с названием скоупа, система автоматически разрешает конфликт, чтобы избежать неоднозначности. Например, имя **session** преобразуется в **session.session**.
При использовании переменных в блоке Скрипт нужно дополнительно указать префикс **context**. Пример: **context.session.value**

Область видимости	Описание	Практическое использование	Пример использования
session	Может использоваться пользователем для записи. Время жизни – пока диалоговая сессия активна	Долгосрочное хранение данных. Бот «запоминает» контекст, делая взаимодействие естественным. Пользователю не нужно повторять информацию. Используется, например, для персистентных данных клиента: можно записать в переменную список симптомов из предыдущих сообщений пациента. При новом цикле «реплика-ответ» в рамках текущей сессии бот проанализирует список из переменной и предложит рекомендации. Снижает нагрузку на пользователя, при этом данные хранятся только в сессии, что важно для приватности.	session.value

Область видимости	Описание	Практическое использование	Пример использования
		Длительность сессии задается пользователем при создании канала, подробнее см. раздел «Публикация в канале»	
request	Может использоваться пользователем для записи. Время жизни – в пределах одной обработки запроса: от реплики пользователя до ответа бота	Для временных данных текущего запроса. Бот «забывает» данные из переменной после ответа на запрос клиента. Например, пациент описывает в сообщении симптомы, бот анализирует их и дает ответ, после чего данные стираются. Ненужные данные не накапливаются – это повышает производительность	request.value
temp	Может использоваться пользователем для записи. Время жизни ограничено рамками выполнения сценария с восстановлением после прерываний	Хранение данных для промежуточных шагов сценария. Устойчивы к прерываниям сценария. Если бот выполняет блоки сценария Ожидание ответа или Переход в сценарий, переменная не стирается. Создается snapshot (снимок состояния) и при возобновлении сценария, данные восстанавливаются в точности такими, какими были до прерывания. Например, при опросе пациента ботом симптомы сохраняются в переменную. Бот запрашивает у клиента уточнения. Если пациент не отвечает сразу, сценарий прерывается. После продолжения данные переменной восстанавливаются и бот продолжает, используя старые ответы из переменной. Для клиента экономится время – не нужно повторять ответы. Для системы эти же данные могут быть, например, интегрированы в отчет по окончании диалога. Полезно применять в ветвлениях графа, данные не теряются, даже если процесс приостанавливается	temp.value
system	Недоступен для записи данных. Время жизни переменных управляется системой	Скоуп системных переменных, предоставляемых платформой	system.last_user_message

Область видимости	Описание	Практическое использование	Пример использования
nlu	Создается автоматически, заполняется результатами работы классификатора или регулярных выражений. При необходимости список кандидатов можно изменить	Чтобы вычислить кандидатов для активации по правилам или интентам, нужно добавить программный код в сценарий препроцессинга. Подробнее о функциях см. в описании блока **Скрипт**, раздел «Скрипт». При необходимости в переменные можно добавлять нового кандидата. Для этого также необходимо добавить код в блоке **Скрипт**	nlu.matches
events	Создается автоматически при обработке входящего события	Заполняется информацией о кандидатах для активации по событию	events.events_candidates

Зарезервированные переменные

В платформе предусмотрены системные переменные, значения которых по умолчанию рассчитываются движком. Их можно использовать при наполнении сценария логикой. Например, при составлении выражения в блоке **Условие** укажите переменную **last_user_message**, чтобы проверить последнее сообщение от пользователя.

При обращении к предопределенным переменным указывайте область видимости. Например, для системных переменных префикс **system**. Формат: **<Префикс>.<Имя переменной>**. Например, **system.last_user_message**. Если префикс не указан, то переменной автоматически присваивается префикс по умолчанию – **session**.

system

Переменная	Тип	Описание
system.system_session_id	**STRING**	Идентификатор чата
system_message_id	**STRING** или **INT**	Идентификатор запроса пользователя
sure_topic	**STRING**	Последний определенный интент в диалоге пользователя с ботом. Заполняется автоматически при любом запросе
topic_score	**DOUBLE**	Вероятность **sure_topic**
last_error_message	**STRING**	Последнее сообщение об ошибке с локализацией места в сценарии, где она произошла
last_user_message	**STRING**	Последнее сообщение пользователя из диалога с ботом **Примечание**. Текст сообщения сохраняется в нижнем регистре

Переменная	Тип	Описание
united_user_messages	STRING	Конкатенация (объединение) всех сообщений пользователя в рамках диалога
last_user_message_length	INT	Количество символов в последнем сообщении пользователя из диалога
messages_number	INT	Номер сообщения пользователя в диалоге
number_of_words	INT	Количество слов в последнем сообщении пользователя из диалога
source_message	STRING	Текст сообщения **Примечание**. В тексте сообщения сохраняется исходный регистр
surface_metadata	JSON OBJECT	Метаданные поверхности. Содержит данные поверхности, которые могут быть использованы в сценарии. Например, system.surface_metadata.chat_id, system.surface_metadata.user_id, system.surface_metadata.fields
utc_now_dt	STRING	Текущие дата и время

response_additional_data	JSON OBJECT	Дополнительные данные для ответа. Значение задается вручную в блоке **Скрипт**.
		Пример:
		`def handler(context: Context) -> None:`
		`  context.system.response_additional_data = {"step": 1}`

channel_id	STRING	ИД канала

input	JSON OBJECT	Данные входящего сообщения.
		В зависимости от типа сообщения объект имеет вид:
		`{`
		`  "type": "message",`
		`  "original_text": "text"`
		`}`
		или
		`{`
		`  "type": "event",`
		`  "name": "some_event",`
		`  "data": {"data_1":"data_1_value"},`
		`}`
		Данные из переменной можно использовать для построения логики. Например, если пришло входящее сообщение с типом "event", то можно активировать сценарий по событию с именем "some_event". Подробнее

Переменная	Тип	Описание
		об активации по пользовательскому событию см. раздел «Событие»

nlu

Область видимости создается автоматически. Переменные заполняются результатами работы правил, классификатора или регулярных выражений.

Переменная	Тип	Описание
matches	ARRAY	Кандидаты для активации на основе регулярных выражений. Вычисляются движком автоматически до препроцессинга
intents	ARRAY	Кандидаты-интенты для активации. Не вычисляются движком автоматически. Чтобы вычислить их, нужно вызвать контекстную функцию predict_intent в блоке **Скрипт** сценария препроцессинга
raw_intents	ARRAY	Список интентов и их вероятности (score), сформированный в результате анализа сообщения. Интент и score первого элемента списка также сохраняется в переменные **system.sure_topic** и **system.topic_score** соответственно

При необходимости в переменных можно переопределять результаты работы правил или классификатора. Для этого в блоке **Скрипт** добавьте код, например:

```
def handler(context: Context) -> None:
    edge = IntentEdge(type="intent", value="new", threshold=0.5, target_node_id="node2")
    new_candidate = ActivationCandidate(scenario_id=2, edge=edge, score=0.8)
    context.nlu.intents.append(new_candidate)
```

events

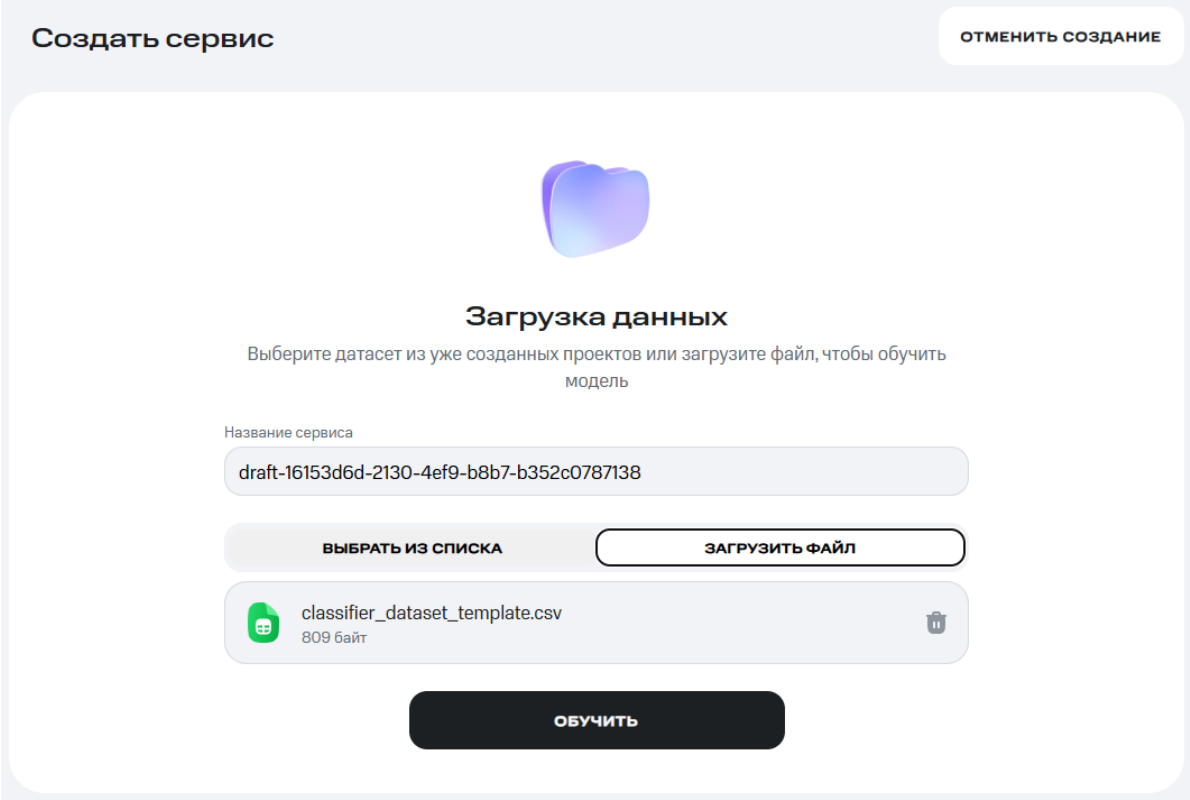
Переменная	Тип	Описание
candidates	ARRAY	Массив кандидатов для активации по событию

Привязка классификатора

Чтобы бот попадал в нужные сценарии на основе примеров, создайте сервис классификатора и привяжите его к сценарию. Для этого:

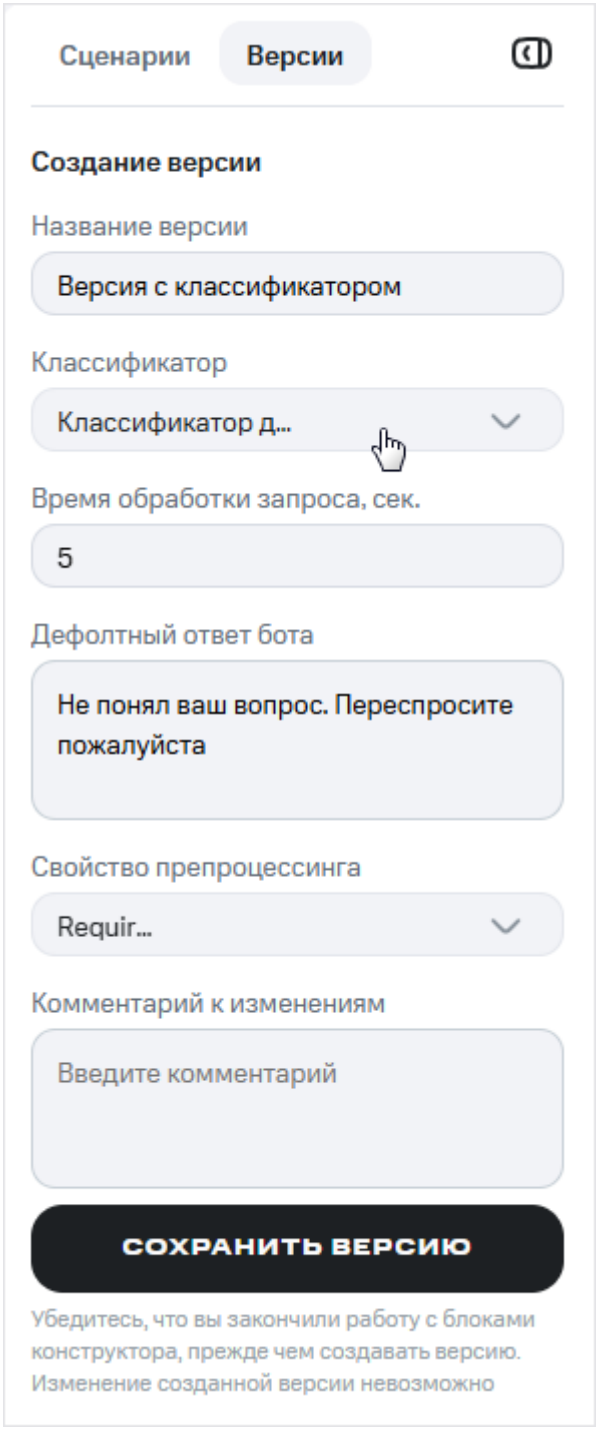
1. Создайте проект разметки данных и скачайте полученный датасет для обучения сервиса классификатора. Подробнее см. в разделе «Разметка данных».
2. В веб-клиенте перейдите на страницу **AI сервисы**.
3. Нажмите на кнопку **Создать сервис**.
4. В открывшемся окне выберите тип сервиса **Классификатор**.

5. В окне **Создание сервиса** измените название сервиса. По умолчанию в качестве названия сервиса используется 36-значный идентификатор. При необходимости его можно изменить позднее.



6. Загрузите сохраненный ранее датасет. Размер не должен превышать 100 МБ.
7. Нажмите на кнопку **Обучить**. В результате начинается процесс обучения модели. Он может занимать некоторое время. По окончании запуска готовый сервис переходит в статус **Запущен**.
8. На странице **Боты и агенты** перейдите в список ботов и откройте конструктор нужного бота.
9. Создайте для бота новую версию, чтобы изменить настройки. Для этого нажмите на кнопку **Создать новую версию**.

10. Задайте для версии название. В выпадающем списке **Классификатор** выберите ранее обученный классификатор:



Для работы классификатора в сценарий препроцессинга необходимо добавить скрипт для вычисления результатов классификации. Проверьте, что в поле **Свойство препроцессинга** указано значение **Required**. Подробнее см. описание блока **Интент** в разделе «Блоки активации».

11. Сохраните изменения по кнопке **Сохранить версию**.

Подробнее о сервисе классификатора см. раздел «AutoML-сервисы».

Удаление сценария

Не рекомендуется удалять сценарий целиком, так как на него могут вести ссылки из других сценариев. В этом случае появляется соответствующее предупреждение. Если нужно

перезаписать сценарий, рекомендуется очистить только его содержимое. Для этого удалите блоки и связи, затем заново наполните сценарий логикой.

Если сценарий не нужен, и их из других сценариев нет на него ссылок, то вы можете удалить его. Для этого выделите его на вкладке **Сценарии**, вызовите контекстное меню по кнопке и выберите пункт **Удалить**.

Улучшение и кастомизация бота

Бота можно обновлять и развивать, например, добавлять новые функции и исправлять ошибки. Все доработки выполняются в новой версии.

Для расширения возможностей бота в конструкторе доступны:

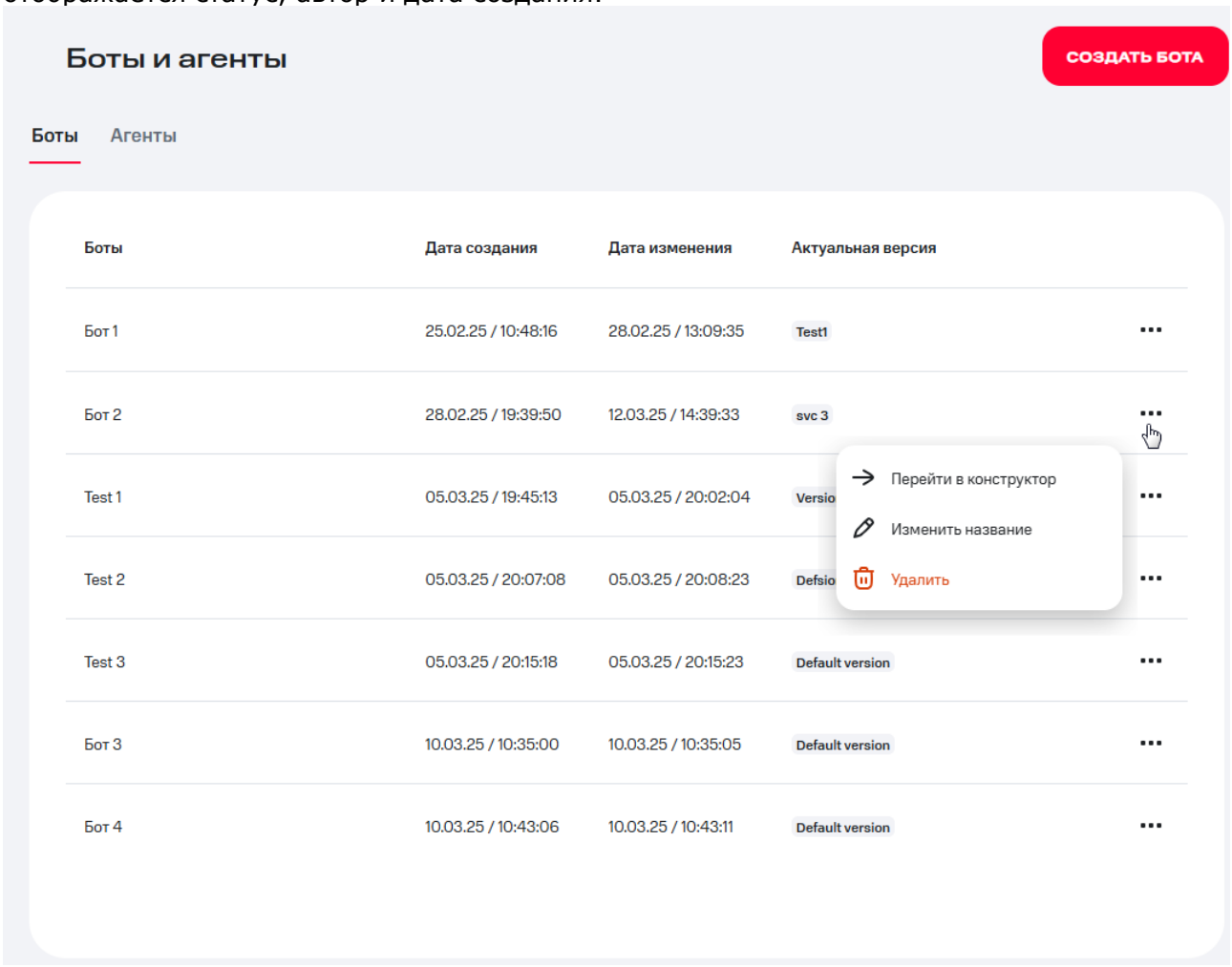
- обучение классификатора на основе размеченных диалогов. Обучите сервис классификатора, чтобы в дальнейшем бот выполнял запросы наиболее точно;
- использование RAG-сервисов для создания специальных типов сценариев. Это полезно, если нужно выделять информацию из документации или базы знаний и на основе нее формировать запрос к модели. В результате запрос пользователя обогащается дополнительными данными, что позволяет генерировать максимально точные и полезные ответы;
- использование сервиса NER для распознавания именованных сущностей в теле запроса;
- добавление управляющих элементов с произвольным кодом на JavaScript;
- использование управляющих элементов для HTTP-вызовов интеграций.

При необходимости вы можете открывать сценарии ботов на просмотр или редактирование. При этом для доработки версию нужно создать вручную.

Просмотр и редактирование

Со страницы «Боты» можно переходить в созданного бота и открывать его версии. Для этого:

1. Нажмите на кнопку → . Открывается список версий выбранного бота. По каждой версии отображается статус, автор и дата создания.



2. Вызовите контекстное меню по кнопке ••• и выберите пункт **Перейти в конструктор**.

   Если бот заблокирован другим пользователем, то в веб-клиенте появляется сообщение с информацией о пользователе, который работает со сценарием. Вы сможете начать редактирование после снятия блокировки.

В результате конструктор сценариев открывается на просмотр. Чтобы редактировать сценарий, предварительно установите на него блокировку. Для этого нажмите на кнопку **Начать работу**.

Подключение RAG-сервисов

Сервис RAG предоставляет возможность интегрировать в сценарии ботов технологию Retrieval-Augmented Generation. Перед обращением к языковой модели (LLM) навык выделяет необходимую информацию из базы знаний и на основе этой информации формирует запрос к модели. В результате запрос пользователя обогащается дополнительными данными, что позволяет генерировать максимально точные и полезные ответы. Подробнее о сервисах см. раздел «Сервисы RAG».

Сервисы RAG рекомендуется подключать в сценарии для простых запросов, которые не требуют сложной обработки. Это могут быть запросы формата «Вопрос-ответ». В таком случае информация ищется в заранее подготовленной документации – базе знаний.

Например, клиент обращается к боту медицинской организации с вопросом «Как подготовиться к УЗИ брюшной полости». Так как запрос подразумевает ответ, который есть в общедоступной документации, то для него можно использовать сервис RAG.

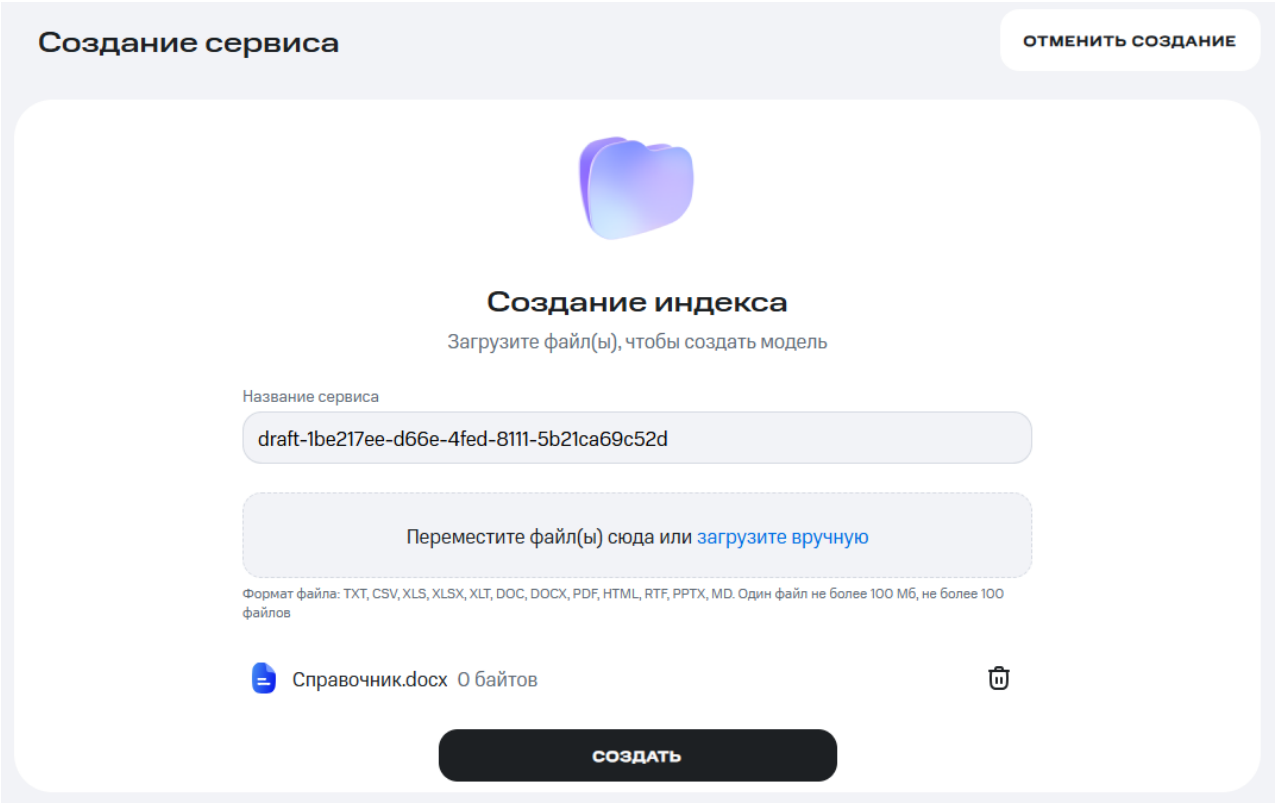
Чтобы использовать RAG в сценариях:

1. Создайте сервис.
2. Протестируйте работу модели.
3. Подключите сервис к боту. Для этого используйте интеграционный блок **HTTP-запрос**.

Создание сервиса RAG

1. Перейдите на страницу **AI-сервисы**. Нажмите на кнопку **Создать сервис**.
2. В открывшемся окне выберите тип сервиса **RAG**:

3. В окне **Создание сервиса** задайте название сервиса. По умолчанию в качестве названия сервиса используется 36-значный идентификатор. При необходимости изменить его можно позднее.



4. Загрузите файлы документации или базы знаний.

Поддерживаются файлы в форматах TXT, CSV, XLS, XLST, XLT, DOC, DOCX, PDF, HTML, RTF, PPTX, MD. Загрузить можно до 100 файлов, размер каждого – не более 100 МБ.

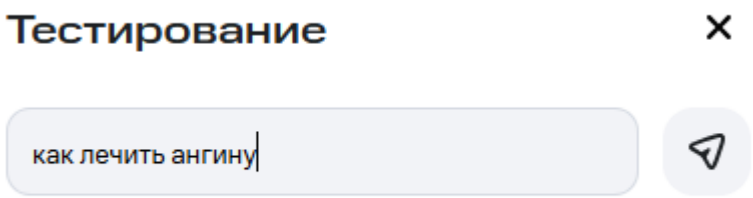
5. Нажмите на кнопку **Создать**. Создание индекса занимает некоторое время. Когда сервис готов, его статус в списке меняется на **Запущен**.

В процессе создания:

- проверяется лимит уже созданных сервисов. Если он превышен, то отображается сообщение об ошибке создания сервиса. Нажмите на кнопку **Вернуться к списку**, удалите один или несколько сервисов и попробуйте создать сервис заново;
- проверяется уникальность названия сервиса. Если в системе уже существует сервис с таким же названием, то отображается сообщение о том, что имя сервиса занято. Нажмите на кнопку **Вернуться к списку** и переименуйте сервис.

Тестирование сервиса RAG

1. Откройте карточку сервиса.
2. Нажмите на кнопку **Тестировать**. В результате открывается панель **Тестирование**.
3. Введите запрос для тестирования и нажмите на кнопку .

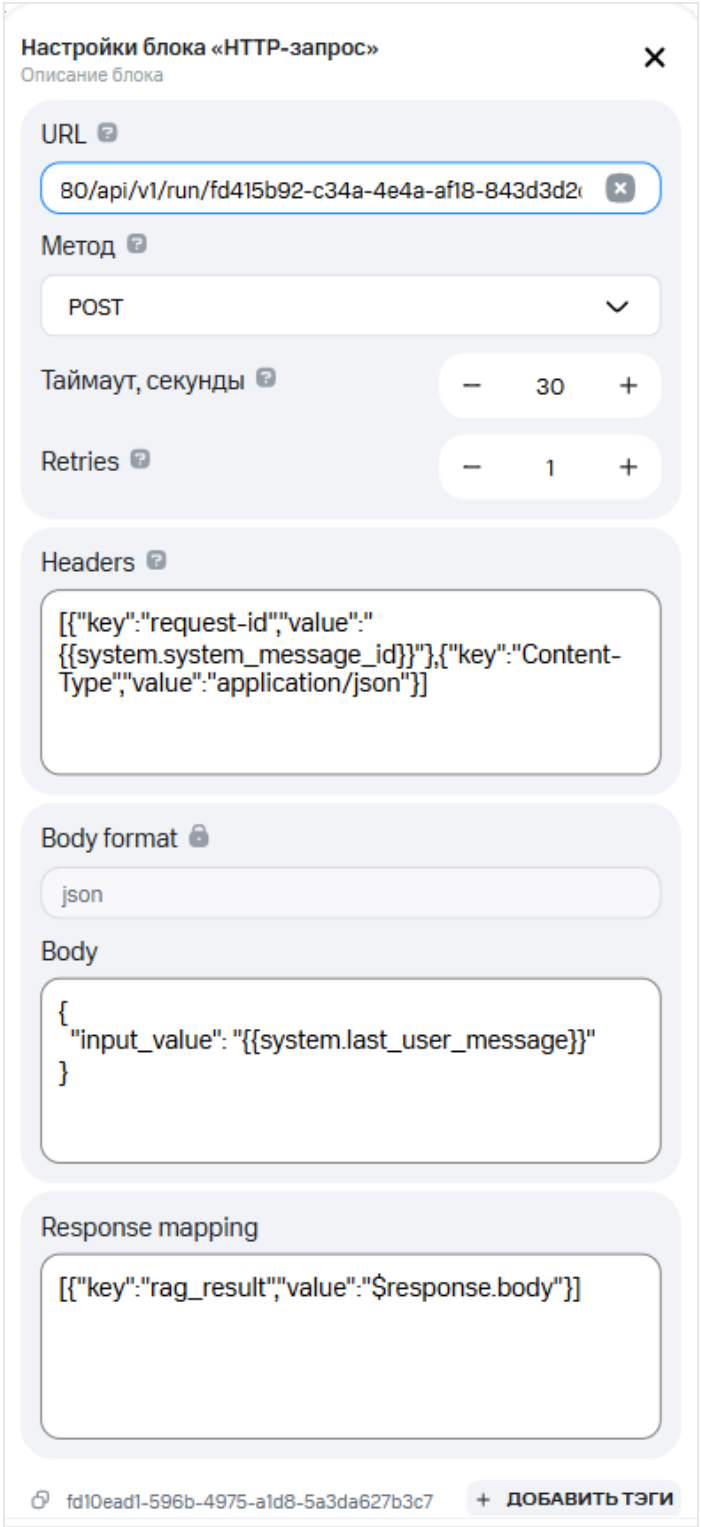


В результате модель подготавливает ответ и ссылки на документы, в которых этот ответ был найден.

Подключение сервиса к боту

1. Перейдите в конструктор сценариев бота.

2. Добавьте в сценарий блок **HTTP-запрос**. Подробнее о параметрах блока см. раздел реакции». Заполните поля блока, например:



В параметрах укажите:
URL. Адрес сервиса RAG. Перейдите в раздел AI-сервисы, откройте карточку нужного RAG-сервиса и скопируйте его адрес из поля **URL сервиса**.
Метод. Выберите значение **POST**.
Таймаут, секунды. При необходимости скорректируйте значение. Как правило, RAG обрабатывает запросы дольше других ML-сервисов, поэтому для него таймаут может быть

увеличен.
Headers. Укажите значение:

```
[{"key":"request-id","value":"{{system.system_message_id}}"},{"key":"Content-Type","value":"application/json"}]
```

Body. Укажите значение:

```
{"input_value": "{{system.last_user_message}}"}
```

Response mapping – маппинги. Подробнее о заполнении поля см. в описании блока HTTP-запрос в «Блоки реакции».

3. Наполните сценарий остальными блоками. Например, добавьте блок LLM, который будет заниматься обработкой полученного результата, и текстовый блок для вывода результата в чате.

Подключение сервиса NER

Платформа поддерживает AutoML-сервисы, обслуживающие типы моделей Классификатор и **NER**.

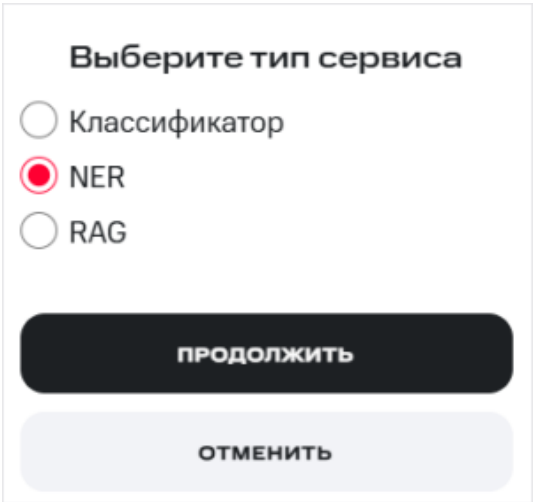
NER – это ML-модель, способная распознавать именованные сущности в тексте запроса. Например, NER-сервис можно подключить в сценарий бота медицинского помощника для извлечения и структурирования информации. Из предложения «Появилась сыпь на руках и ногах» с помощью обученного сервиса определяются сущности «симптом» и «локация».

Чтобы использовать NER в сценариях:

1. Создайте сервис.
2. Протестируйте работу модели.
3. Подключите сервис к боту. Для этого используйте интеграционный блок **HTTP-запрос**. Укажите в нем URL созданного сервиса, а также заполните другие параметры.

Создание и обучение сервиса NER

1. Перейдите на страницу **AI-сервисы**. Нажмите на кнопку **Создать сервис**.
2. В открывшемся окне выберите тип сервиса **NER**. Нажмите на кнопку **Продолжить**.



3. Укажите имя сервиса и выберите датасет из созданных проектов или загрузите файл. Если нужно добавьте, пользовательский словарь синонимов. Нажмите на кнопку **Обучить**, чтобы начать обучение.

Подробнее см. раздел «Создание AutoML-сервиса».

Тестирование сервиса NER

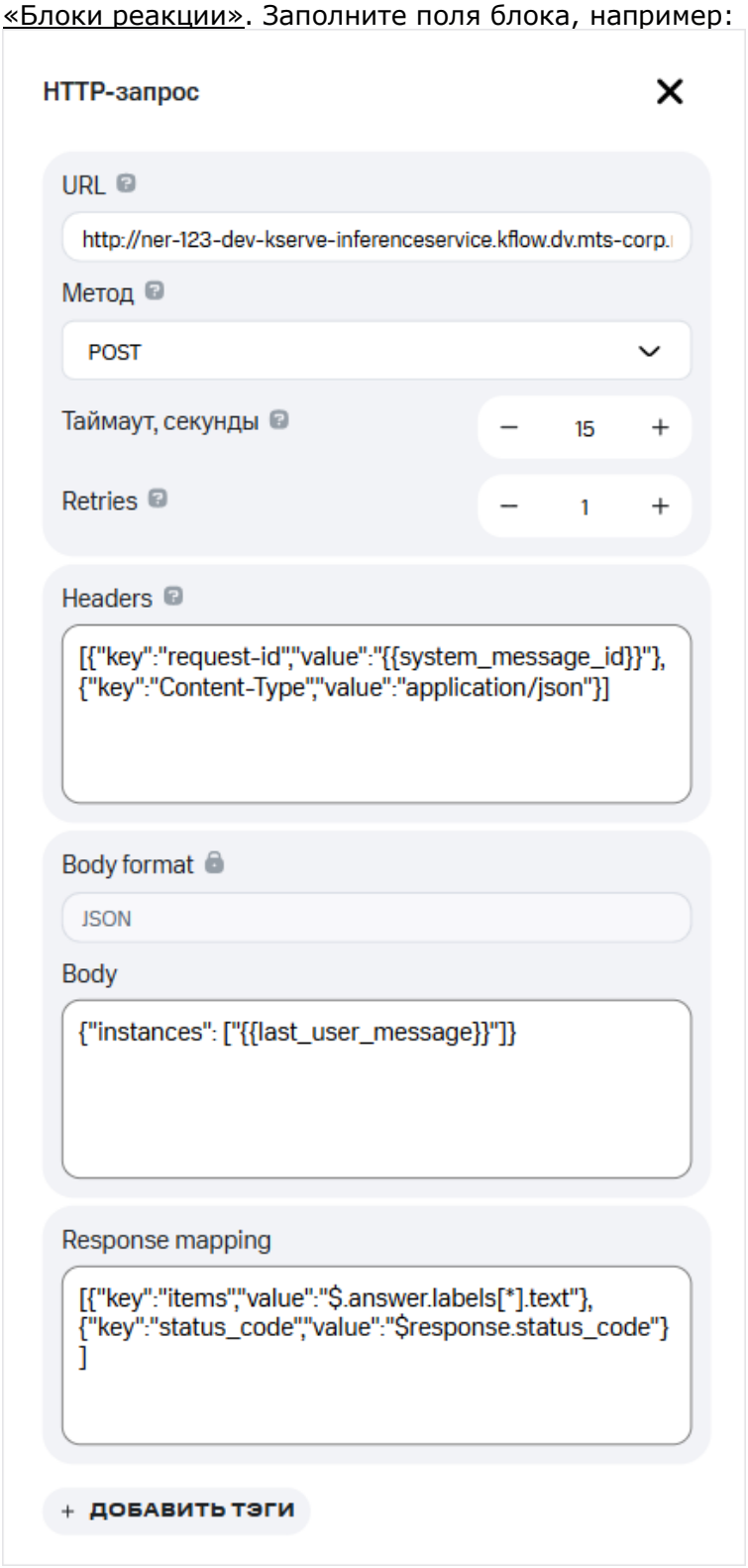
1. Откройте карточку сервиса.
2. Нажмите на кнопку **Тестировать**. В результате открывается панель **Тестирование**.
3. Введите запрос для тестирования и нажмите на кнопку  .

Подробнее см. раздел «Тестирование AutoML-сервиса».

Подключение сервиса к боту

1. Перейдите в конструктор сценариев бота.

2. Добавьте в сценарий блок **HTTP-запрос**. Подробнее о параметрах блока см. раздел «Блоки реакции». Заполните поля блока, например:



В параметрах укажите:
URL. Адрес сервиса NER. Скопируйте его из поля **URL модели** в карточке сервиса.
Метод. Выберите значение **POST**.
Headers. Укажите значение:

```
[{"key":"request-id","value":"{{system.system_message_id}}"},{"key":"Content-Type","value":"application/json"}]
```

Body. Укажите значение:

```
{"instances": ["{{system.last_user_message}}"]}
```

Response mapping – маппинги. Подробнее о заполнении поля см. в описании блока HTTP-запрос в «Блоки реакции».

3. Наполните сценарий остальными блоками.

Использование программного кода

Если функциональности no-code-конструктора недостаточно для закрытия потребностей организации, то можно встроить в сценарий бота произвольный программный код. В конструкторе поддерживается код на языке Python. Для этого в рабочую область добавьте блок **Скрипт**.

Запуск и использование

1. Протестируйте созданного бота, чтобы проверить логику работы и убедиться в отсутствии ошибок в написании сценария.
2. Подключите канал, по которому клиент может обратиться к боту и получить информацию.
3. Сделайте разметку истории диалогов по подключенному каналу, чтобы улучшить работу следующих запросов.

При необходимости ненужного бота можно удалить.

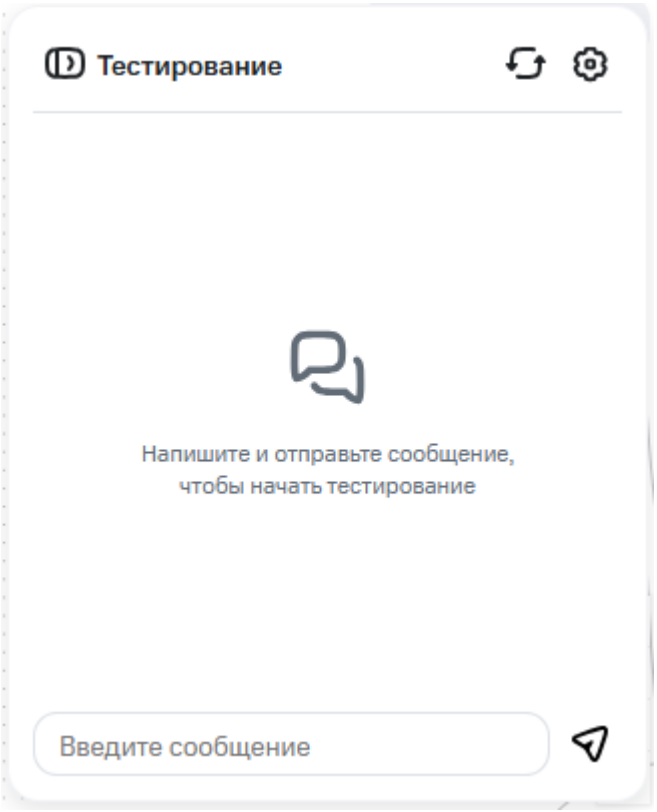
Тестирование

> Перед началом тестирования убедитесь, что версия сохранена.

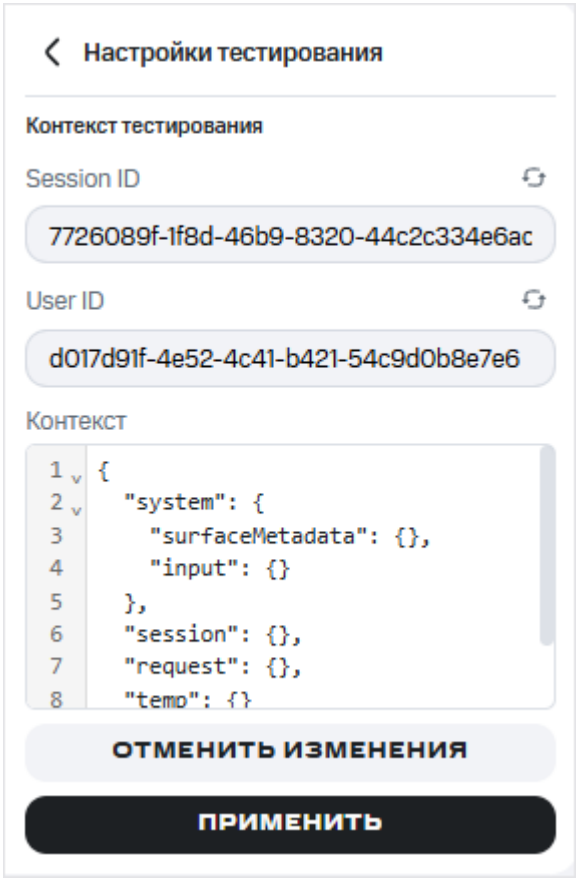
После разработки сценариев протестируйте бота. Для этого:

1. В конструкторе сценариев нажмите на кнопку **Тестирование**.

2. В открывшемся чате введите текстовый запрос и проверьте работу сценария.

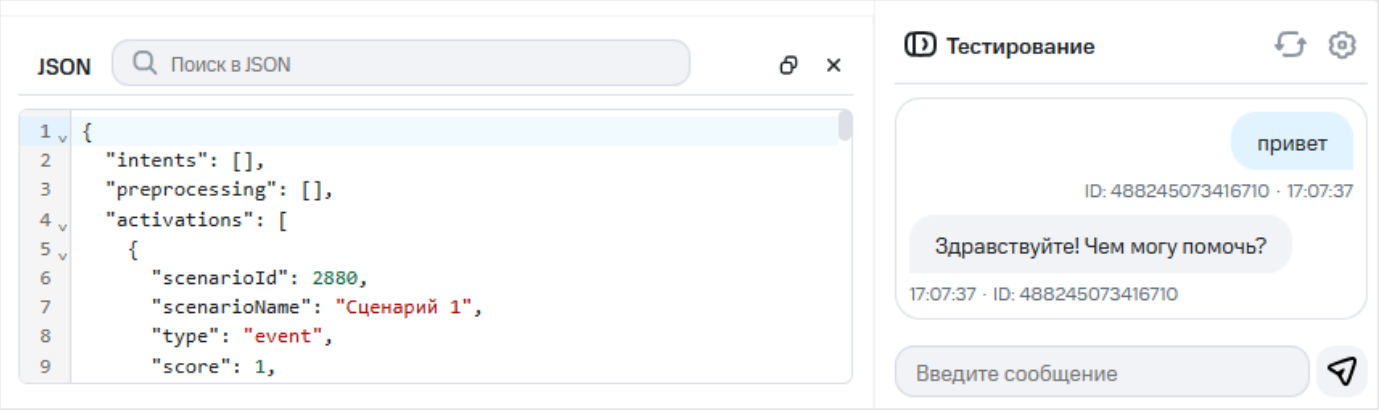


3. При необходимости настройте контекст тестирования. Для этого нажмите на кнопку и скорректируйте значения полей:



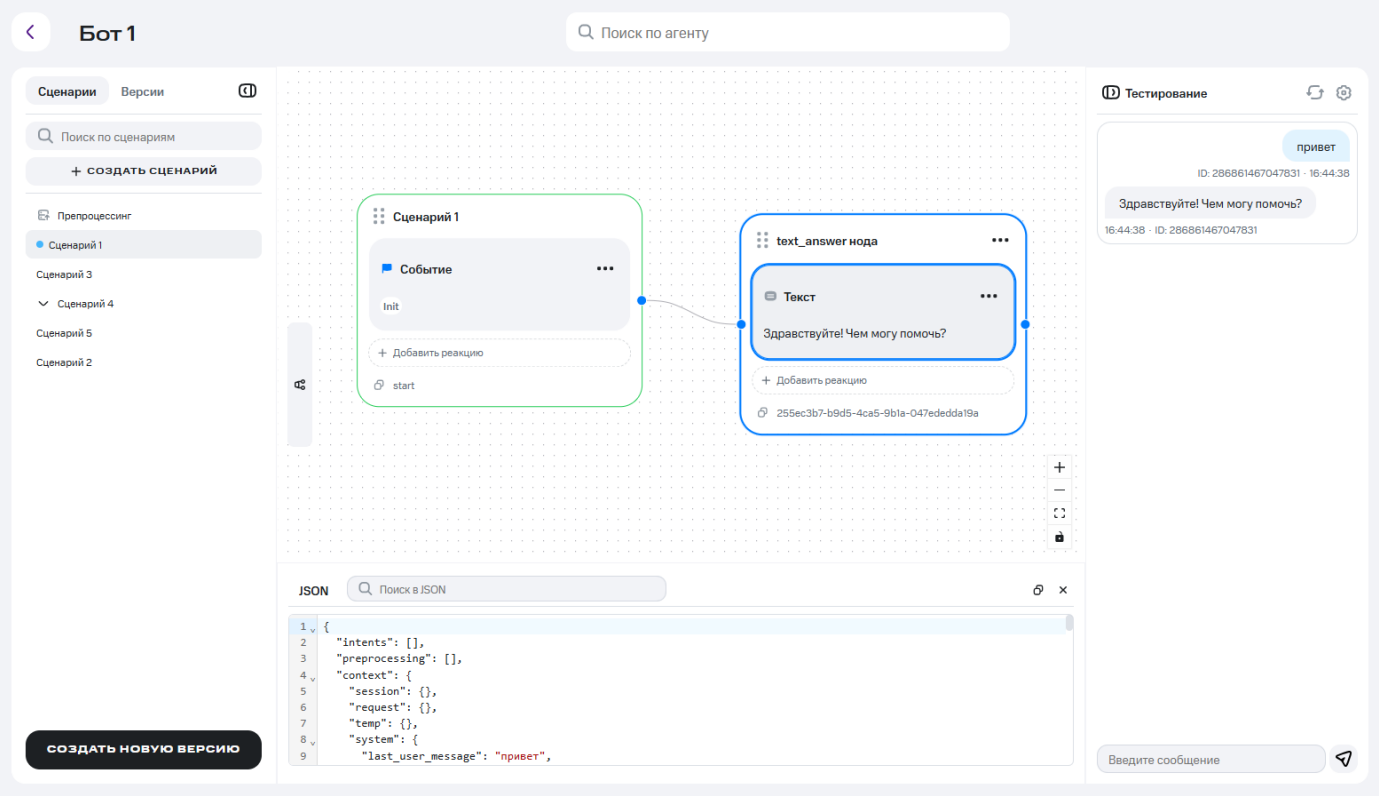
Для отладки вы можете посмотреть информацию о шаге диалога. После получения ответа от бота нажмите на шаг. Сценарий, который был выбран для прохождения, выделяется синим цветом. В

нижней части экрана отображается дебаг-панель – виджет с содержимым JSON-файла, который сформировался в результате выполнения сценария:



По сформированному JSON можно определить, сработавшие во время запроса ноды и блоки, значения переменных, а также другие технические параметры. Кроме этого, из дебаг-панели можно скопировать переменные для использования в блоках сценария.

Для наглядности подсвечиваются блоки, которые выполнились при обработке запроса:



В режиме отладки и тестирования редактирование сценариев недоступно.

Чтобы очистить диалог, нажмите на кнопку . Также диалог очищается при сворачивании тестового виджета.

Удаление проекта

Чтобы удалить неиспользуемый проект:

1. Перейдите на страницу **Проекты.**
2. По кнопке ••• вызовите контекстное меню нужного бота и выберите пункт **Удалить**.
3. В появившемся диалоговом окне подтвердите удаление.
4. В списке проектов проверьте, что удаленного проекта в нем нет.

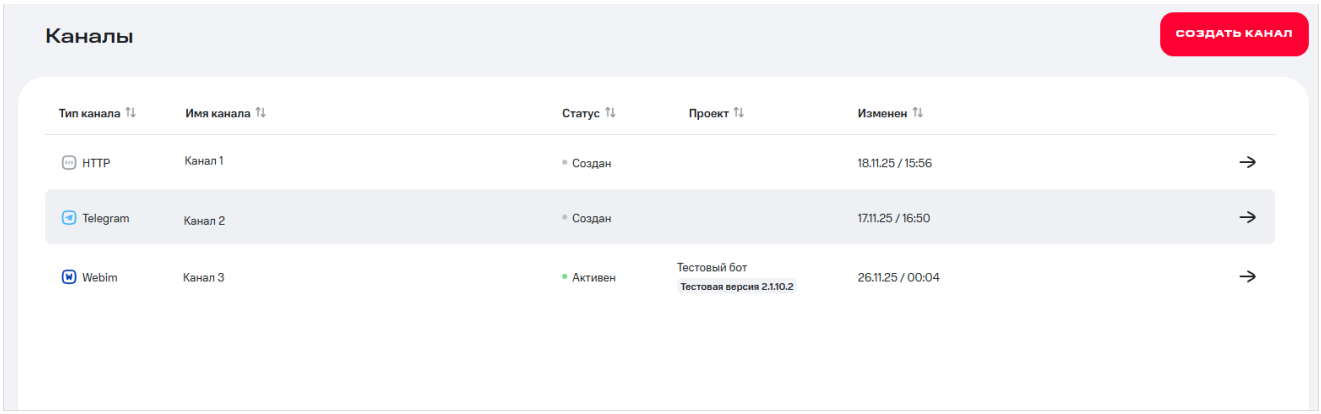
Каналы

Для того чтобы клиенты могли общаться с ботом, его нужно разместить в канале. Канал — это сущность, которая соединяет бота с пользователем, будь то мессенджер или чат на сайте. В одном канале работает только один бот, но бот может быть доступен в нескольких каналах одновременно.

Платформа поддерживает каналы для следующих типов поверхностей:

- Webim – это омниканальная система для общения с клиентами. Вы можете упростить и автоматизировать коммуникационные процессы в Webim, если интегрируете с ней сценарий вашего бота.
- Telegram – канал обеспечивает прием и отправку текстовых и звуковых сообщений, файлов, изображений, документов в мессенджере Telegram. Для интеграции используется Telegram Bot API;
- HTTP – поддерживает прием и отправку данных по HTTP -протоколу, например чаты на сайтах, виджеты, умные устройства.

Список каналов и основная информация о них, включая статус, отображается в разделе **Каналы**. Чтобы изменить настройки какого-либо канала из списка, перейдите в его карточку по кнопке  .

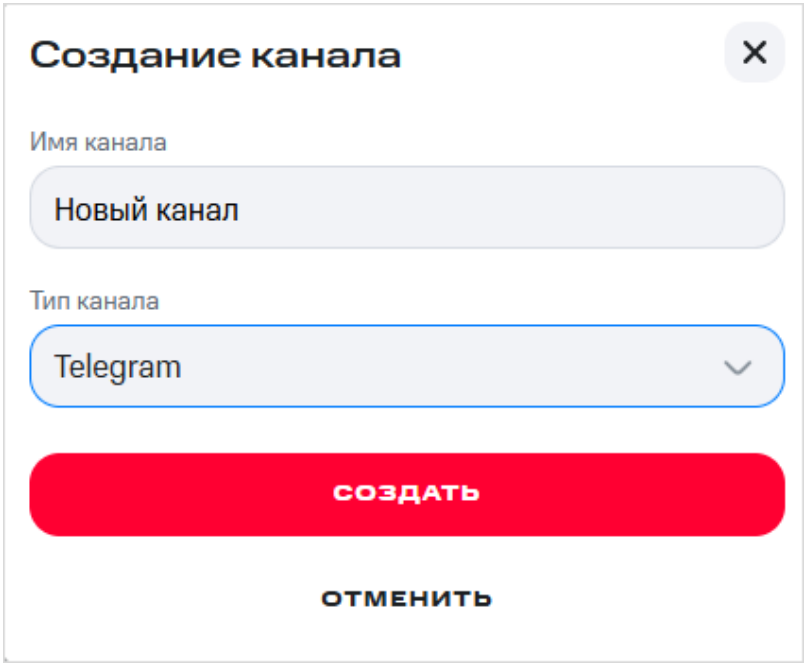


Создание канала

Чтобы создать новый канал для публикации бота:

1. Перейдите в раздел **Каналы**.

2. Нажмите на кнопку **Создать канал**.

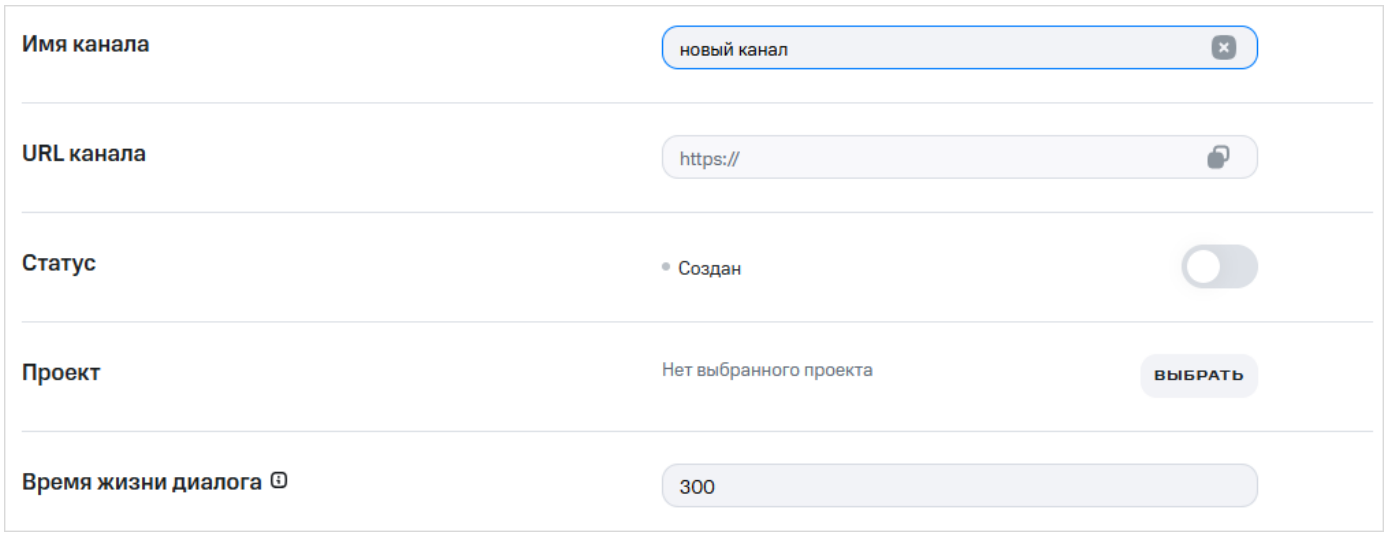


3. В окне **Создание канала** задайте имя и тип, затем нажмите на кнопку **Создать** – откроется карточка канала выбранного типа. Теперь нужно выполнить настройку.

   Тип созданного канала нельзя изменить.

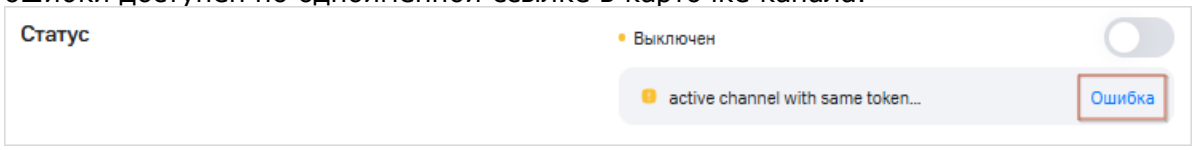
Настройка канала

Базовые настройки одинаковы для всех типов каналов.



- **Имя канала**. Уникальное название канала в системе.
- **URL канала.** Генерируемый системой адрес, по которому бот будет доступен после активации канала. Поле не редактируется.
- **Статус**. Кнопка включения/выключения канала. Статус канала меняется в следующем контексте:
  - **Создан** – стартовый статус канала при создании.
  - **Активируется** – канал в процессе запуска.

  - **Активен** – канал работает.
  - **Выключается** – канал в процессе выключения.
  - **Выключен** – канал выключен. Этот статус присваивается в том числе и при выключении канала из-за ошибки в работе или при активации канала (например, когда истек срок жизни токена или токен уже используется в другом канале). В этом случае отображается информация о причинах выключения. Полный текст ошибки доступен по одноименной ссылке в карточке канала.



- **Проект**. По кнопке **Выбрать** открывается окно для выбора бота и его версии для размещения в канале.

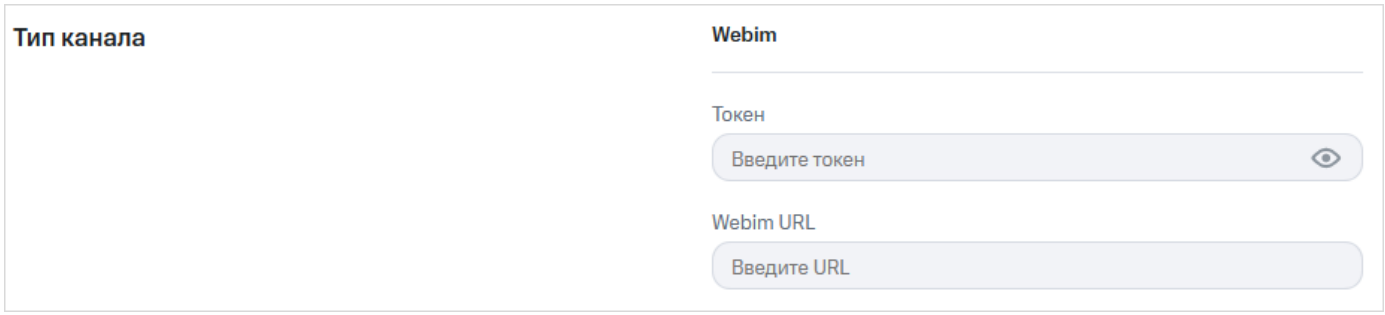
  Для выбора доступны только версии, которые опубликованы. Подробнее см. раздел «Версии».

- **Время жизни диалога**. Время существования сессии диалога с пользователем после последнего пользовательского запроса. По умолчанию – 300 секунд.

Чтобы создать работоспособный канал, установите все перечисленные базовые настройки. Затем в зависимости от типа канала заполните перечисленные ниже дополнительные поля в карточке.

Webim

Для канала Webim необходимо заполнить поля **Токен** и **Webim URL**.

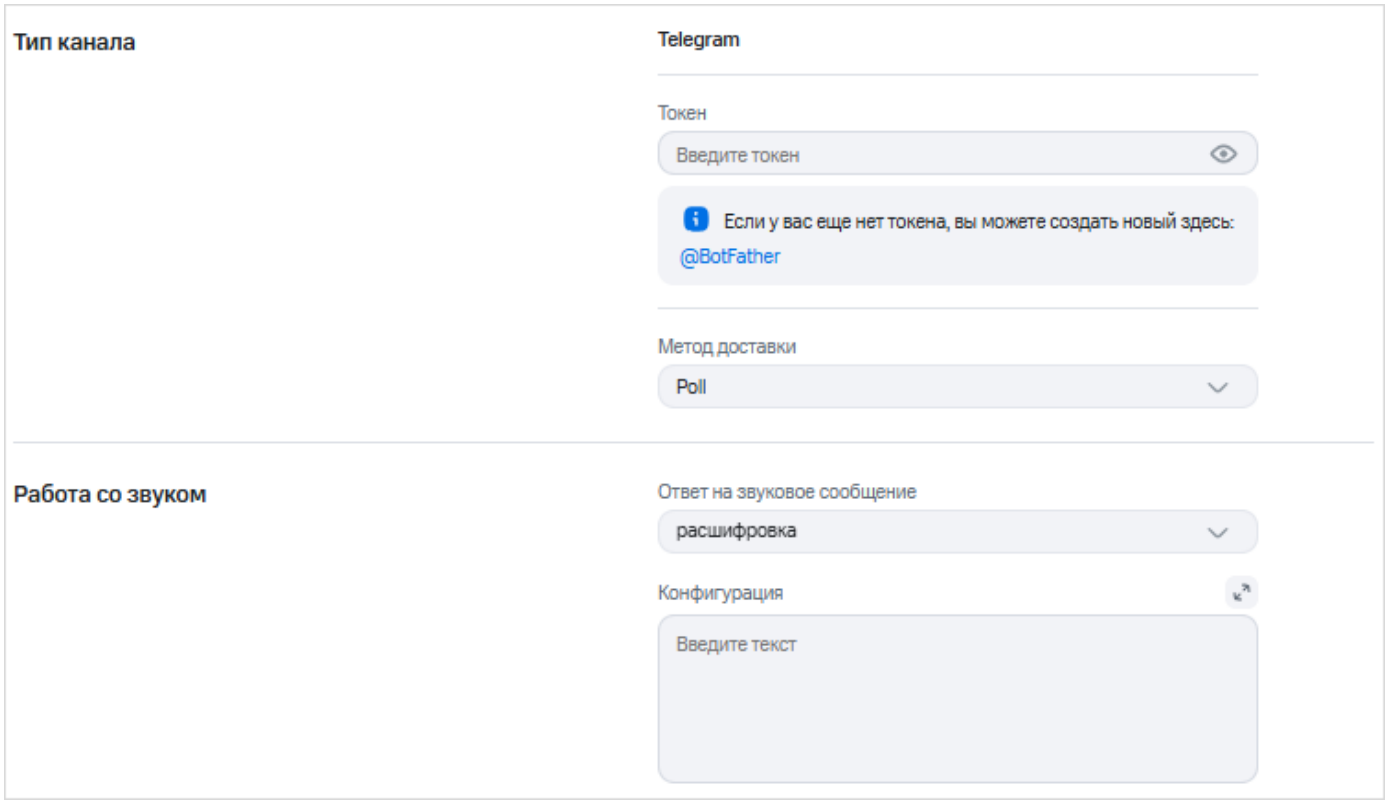


Telegram

Для Telegram укажите:

- Токен для регистрации канала в Telegram – его необходимо получить с помощью telegram-бота BotFather (ссылка указана в карточке канала).
- Метод доставки:
  - **POLL** – бот опрашивает telegram-сервер с определенным интервалом. Если есть новые сообщения, они возвращаются в ответе. Подходит для тестирования и небольших проектов с низким трафиком, где допустимы минимальные задержки в отклике.
  - **PUSH** – telegram-сервер сам отправляет, «пушит» новые сообщения в реальном времени на URL канала. Подходит для ботов с высоким трафиком, где важны мгновенные обновления и масштабируемость.
- Настройки передачи аудио в сообщениях:
  - Ответ на звуковое сообщение:
    - расшифровка;
    - звук;

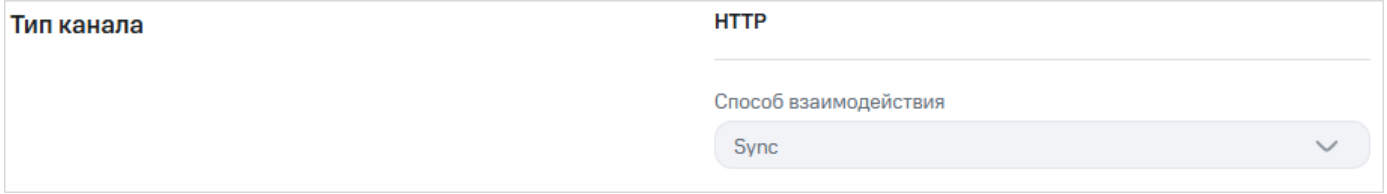
        - звук+расшифровка;
    - Конфигурация STT (Speech-to-Text) и TTS (Text-to-Speech) задается в одноименном поле при помощи JSON. Если оставить поле пустым будут использованы настройки, заданные по умолчанию при установке платформы. Подробнее см. раздел «Конфигурация аудио-сообщений в Telegram».



HTTP

Для HTTP-канала определите **Способ взаимодействия**:

- **Sync (Synchronous)** – классический «запрос-ответ» способ, где клиент – платформа отправляет запрос к поверхности (http-сервер) и блокируется (ждёт), пока не получит полный ответ. Подходит для тестовых или простых, низкотрафиковых ботов.
- **SSE (Server-Sent Events)** – способ для односторонних push-уведомлений от поверхности (http-сервера) к платформе в реальном времени. Подходит для систем, требующих оперативного, но не всегда полного ответа, который можно дослать в процессе взаимодействия. Например, в чатах с эффектом «печатания», когда бот отвечает, набирая текст по словам (например, Chat-GPT) или умных колонках, которые озвучивают ответ поэтапно.



После того, как поля настроек в карточке канала заполнены или в них внесены правки, станет активна кнопка **Сохранить изменения**. Нажмите, чтобы настройки вступили в силу.

Управление настройками канала

В процессе работы канала может понадобиться внести изменения в его настройки.

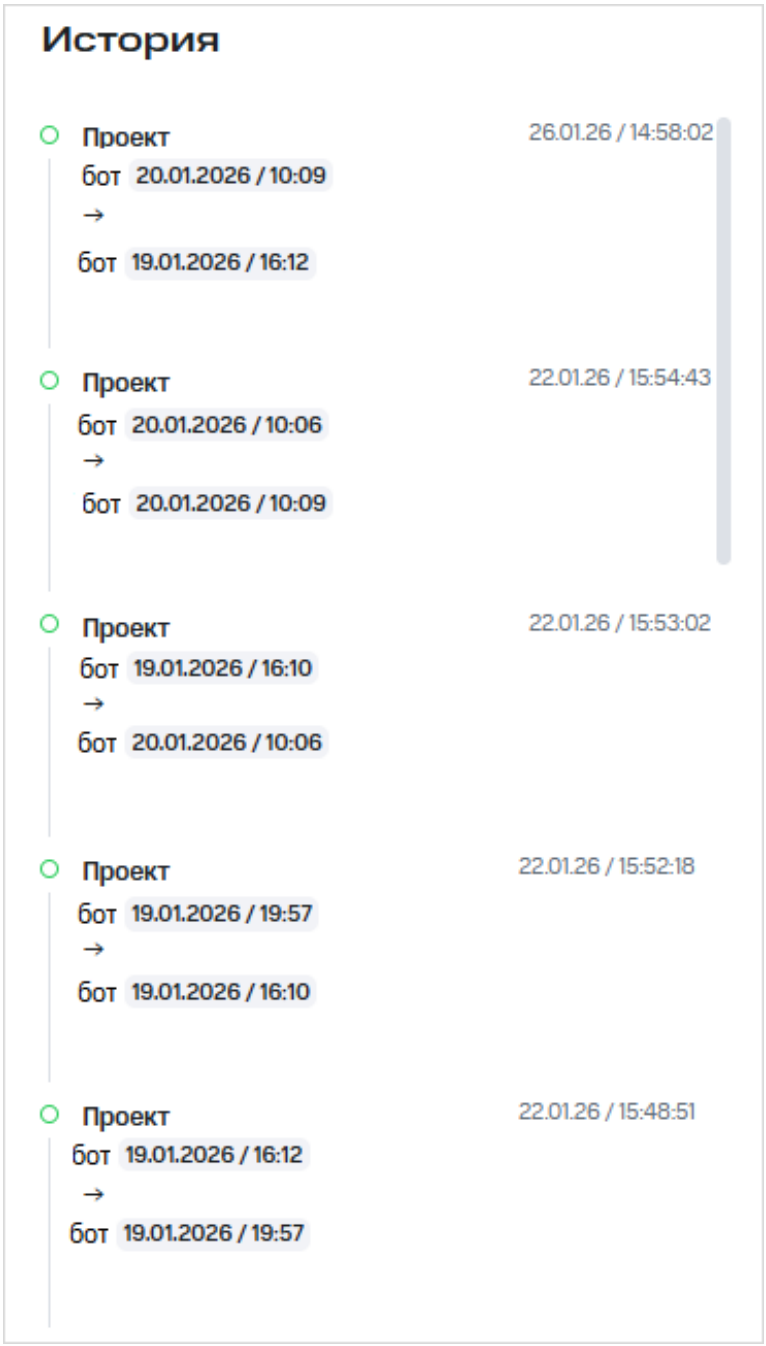
Часть настроек всегда доступна для редактирования (если канал в статусе **Активен**, его не нужно выключать):

- **Имя канала**.
- **Проект** – выбор нового бота, сценария.
- **Время жизни**.

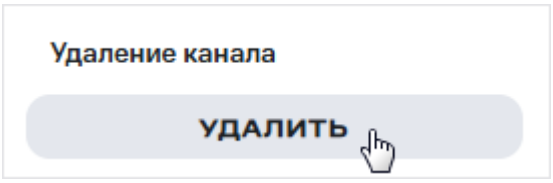
Чтобы изменить токен, метод доставки или способ взаимодействия, а также, чтобы удалить бота из канала, канал необходимо перевести в статус **Выключен**.

История канала

Все изменения канала отображаются в его карточке в разделе **История**:



При необходимости канал можно удалить. Для этого в карточке канала нажмите на кнопку **Удалить**.



Конфигурация аудиосообщений в Telegram

Настройки обработки аудиосообщений для канала Telegram задаются в поле **Конфигурация** при помощи JSON, содержащего конфигурацию механизмов STT (Speech-to-Text, распознавание речи) и TTS (Text-to-Speech, синтез речи). В качестве обработчика звука платформа использует продукт Audiogram.

> Ниже приведен пример JSON и описание базовых параметров. Более подробную информацию о настройке STT и TTS вы можете найти в официальной документации Audiogram https://mts.ai/ru/product/audiogram/audiogram-doc/.

Пример JSON с базовой конфигурацией:

```
{
  "STT": {
    "encoding": 1,
    "sample_rate_hertz": 8000,
    "language_code": "ru",
    "audio_channel_count": 1,
    "model": "e2e-v3",
    "va_config": {
      "usage": 1
    },
    "punctuation_config": {
      "enable": true
    },
    "denormalization_config": {
      "enable": true
    }
  },
  "TTS": {
    "language_code": "ru",
    "encoding": 1,
    "sample_rate_hertz": 8000,
    "voice_name": "gavrilov",
    "synthesize_options": {
     "model_type": "high_quality",
      "voice_style": 0
    }
  }
}
```

Параметры STT (Speech-to-Text)

Параметр	Описание	Возможные значения
encoding	Формат кодирования аудио	1 – LINEAR_PCM (PCM). WAV linear PCM аудиофайл с заголовком, содержащий целые знаковые 16-битные сэмплы в линейном распределении (PCM 16bit) и заданной частотой дискретизации в соответствии с полем sample_rate_hertz 3 – MULAW. WAV PCM аудиофайл с заголовком, содержащий 8-битные сэмплы в формате mu-law и заданной частотой дискретизации в соответствии с полем sample_rate_hertz 20 – ALAW. AV PCM аудиофайл с заголовком, содержащий 8-битные сэмплы в формате a-law и заданной частотой дискретизации в соответствии с полем sample_rate_hertz.
sample_rate_hertz	Частота дискретизации модели (в герцах). Если указана частота дискретизации отличная от значений,	Число, обычно 8000, 16000, 44100, 48000 Гц

Параметр	Описание	Возможные значения
	поддерживаемых моделью, то: • в случае распознавания речи (**ASR**) произойдет перекодирование частоты дискретизации на значение, поддерживаемое моделью (16000 Гц). • в случае синтеза речи (**TTS**) будет использована модель с ближайшей частотой дискретизации в большую сторону.	
language_code	Код языка аудио для распознавания. По умолчанию ru	Строка, например "ru", "en", "kk"
audio_channel_count	Количество аудиоканалов	Целое число, обычно 1 (моно) или 2 (стерео)
model	Модель распознавания речи	e2e-v3 (sample_rate = 16000 Гц)
va_config.usage	Выбор алгоритма обнаружения голоса	0 – стандартное обнаружение голоса) 1 – DO_NOT_PERFORM_VOICE_ACTIVITY (без обнаружения голоса) 2 – USE_DEP (использовать DEP алгоритм) 3 – USE_ENHANCED_VAD (улучшенное обнаружение голоса) 4 – USE_TARGET_SPEECH_VAD (целевое обнаружение речи)
punctuation_config.enable	Включение автоматической пунктуации	true – включить false – отключить
denormalization_config.enable	Включение денормализации текста	true – включить false – отключить

Параметры TTS (Text-to-Speech)

Параметр	Описание	Возможные значения
language_code	Код языка для синтеза речи	Строка, например "ru", "en"

Параметр	Описание	Возможные значения
encoding	Язык, используемый в аудиофайле	1 – LINEAR_PCM (PCM) 3 – MULAW 20 – ALAW
sample_rate_hertz	Частота дискретизации синтезированного аудио в герцах	Число, обычно 8000, 16000, 22050, 24000, 44100, 48000 Гц
voice_name	Имя голоса для синтеза речи	Строка с названием голоса, зависит от доступных голосов на сервере
synthesize_options.model_type	Тип модели синтеза	Строка, возможные значения зависят от сервера, например "high_quality", "standard"
synthesize_options.voice_style	Стиль голоса	0 – нейтральный 1 – радостный 2 – злой 3 – грустный 4 – удивленный 5 – разговорный

Рекомендации по настройкам

- Частота 8000 Гц оптимальна для голосовых сообщений в Telegram (баланс между качеством и размером файла).
- Модель "e2e-v3" – современная модель распознавания речи с использованием сквозных (end-to-end) нейронных сетей.
- Голос "gavrilov" – популярный русскоязычный мужской голос.
- При использовании голосовых сообщений рекомендуется установить encoding = 1 (LINEAR_PCM) для наилучшей совместимости.
- При необходимости улучшения качества распознавания можно включить дополнительные опции в va_config.

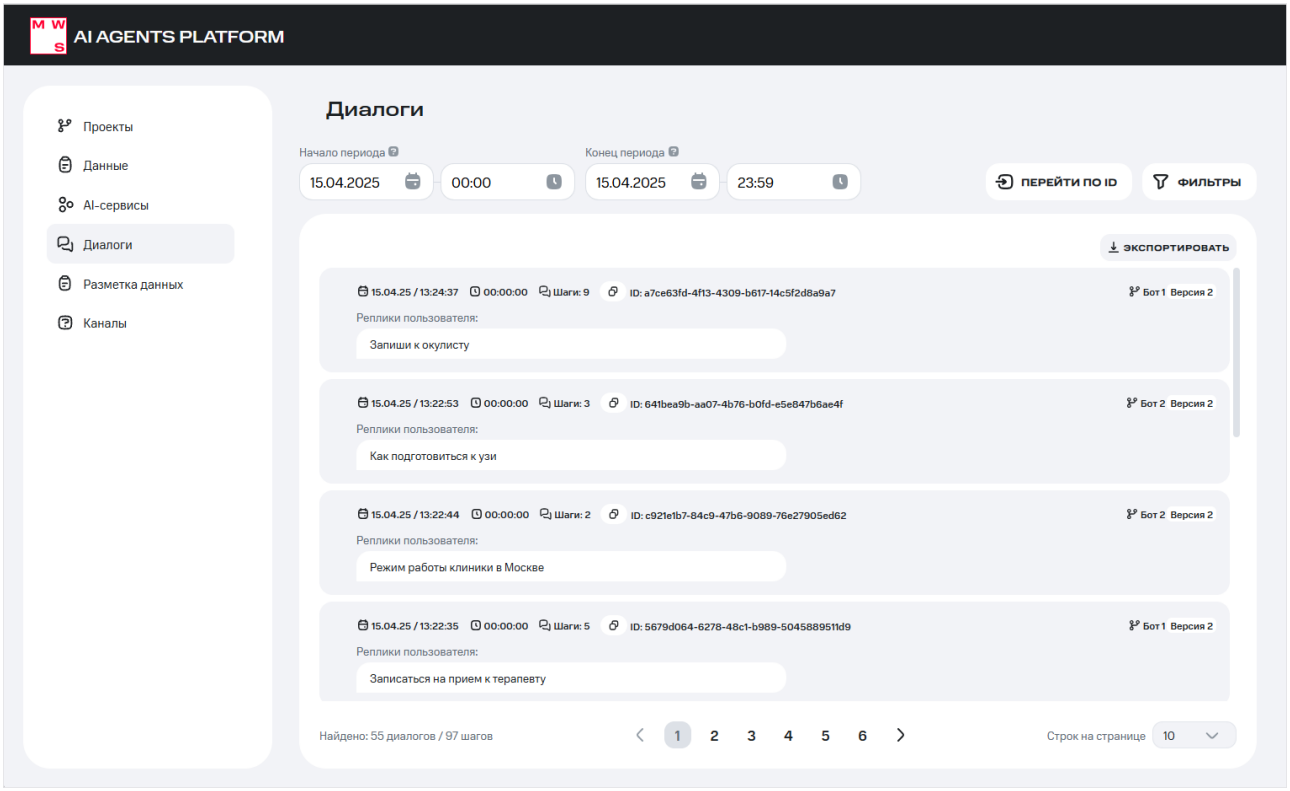
Диалоги

Чтобы в дальнейшем бот отвечал на вопросы более точно, рекомендуется регулярно размечать диалоги, которые уже велись с пользователями.

Для этого разметьте диалоги, затем добавьте файл с результатами к существующему проекту разметки.

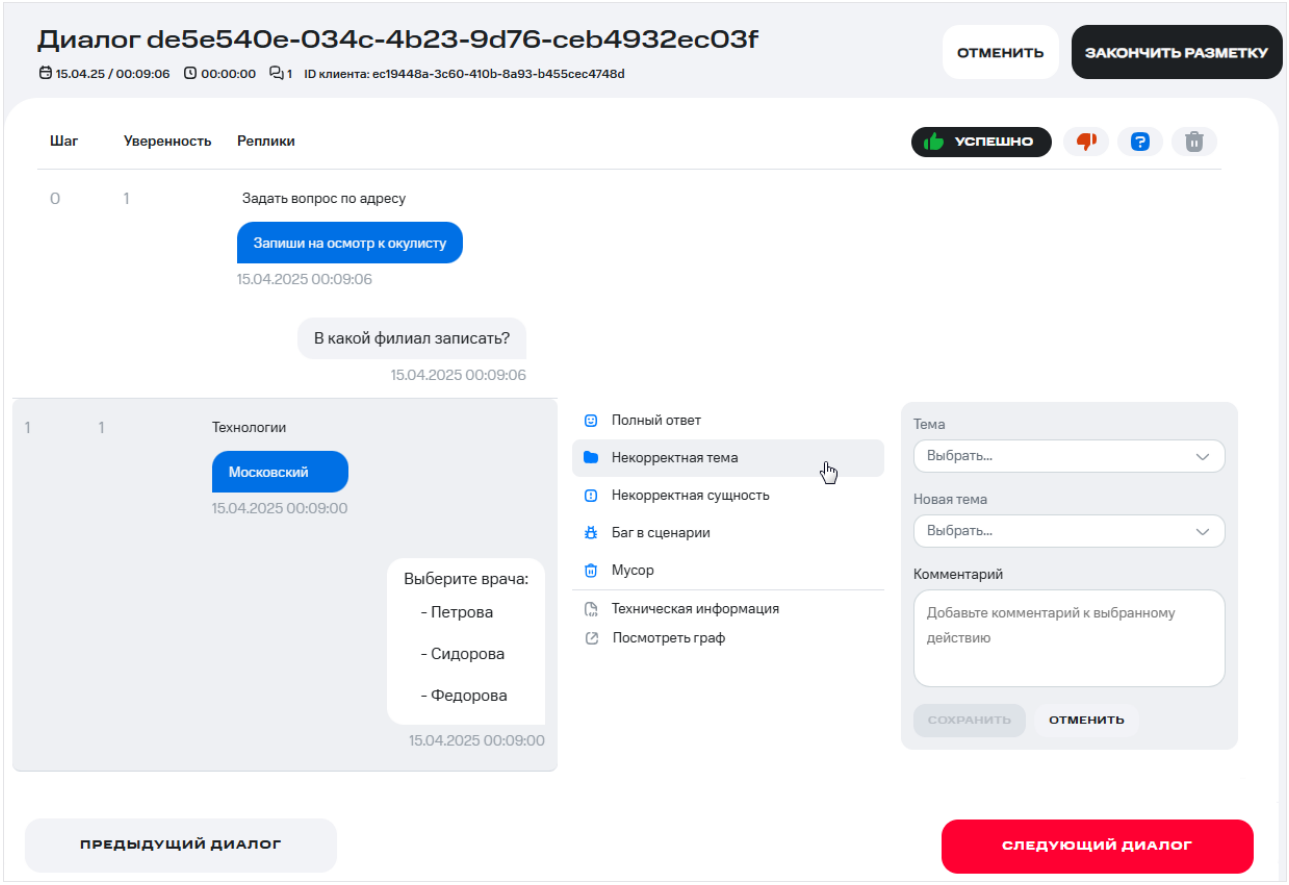
Разметка диалога

1. В веб-клиенте откройте страницу **Диалоги**.



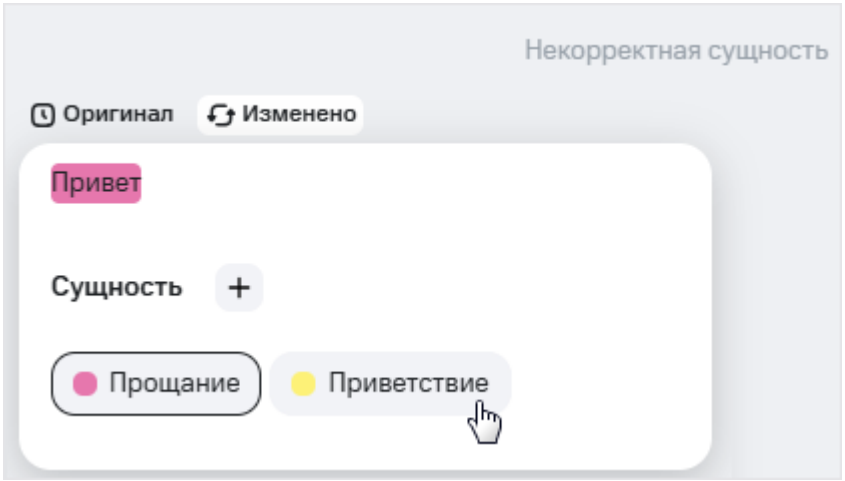
2. Найдите нужный диалог. Для удобного поиска вы можете указать временной промежуток, в который состоялся диалог, а также использовать кнопки **Перейти по ID** и **Фильтры**.
3. Откройте диалог. Для этого нажмите на строку с ним.
4. Перейдите в режим разметки по кнопке **Разметить**.

5. Выделите шаг и скорректируйте разметку. Под шагом понимается реплика пользователя и ответ бота на нее. Например, если неверно была определена тематика, то выделите шаг, отметьте его соответствующим маркером и выберите нужную тему.



Для разметки используются маркеры:

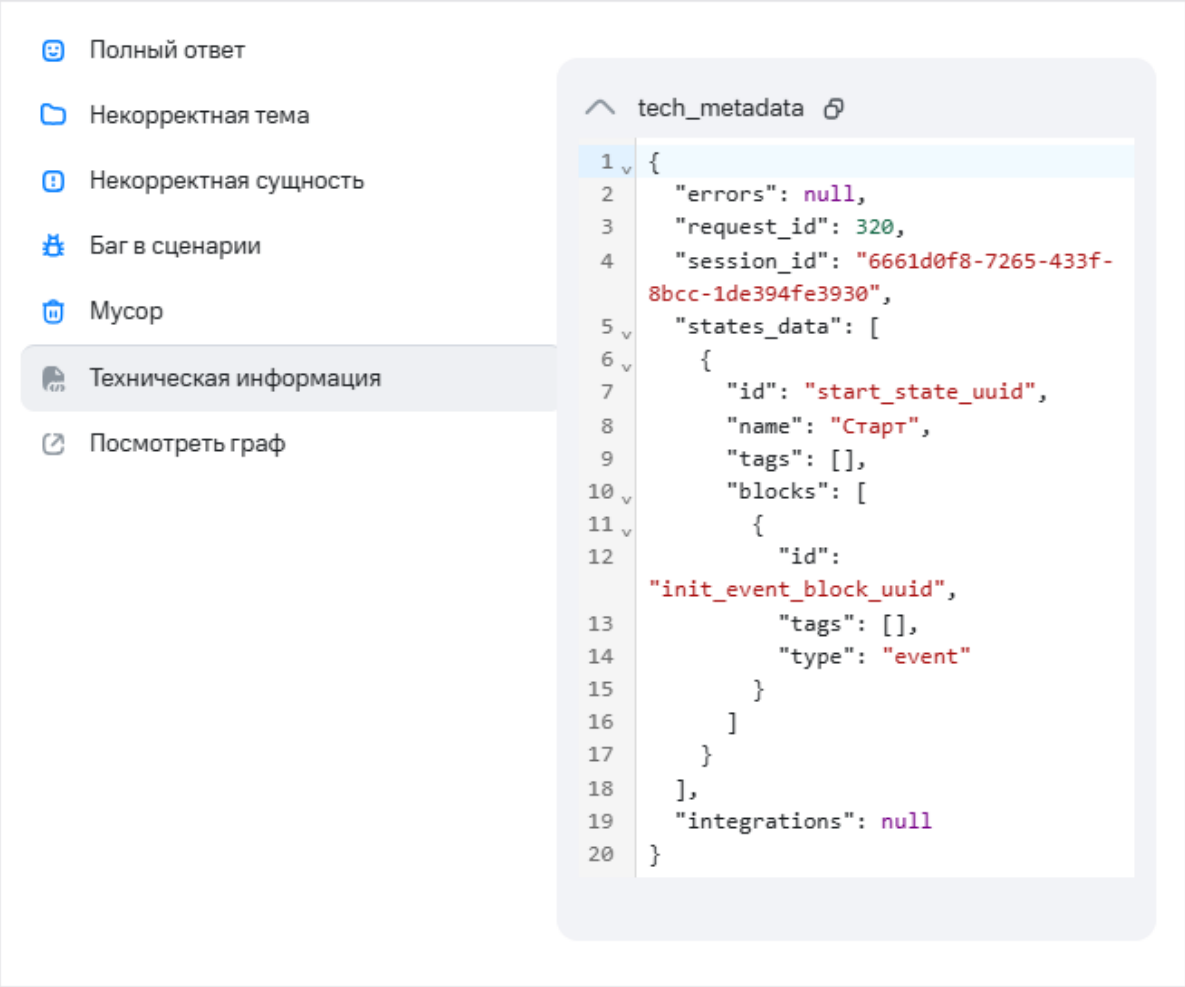
- **Полный ответ**. Выберите этот маркер, если ответ бота удовлетворяет запросу;
- **Некорректная тема**. В этом случае выберите новую тему и оставьте комментарий для следующей разметки;
- **Некорректная сущность**. Выделите слово или его часть и выберите маркер, если сущность не была определена. Если сущность была определена неверно, то предварительно удалите маркер. Для этого наведите на слово и нажмите на ✖ рядом во всплывающей подсказке. Если нужной сущности нет, добавьте ее по кнопке +.



- **Баг в сценарии**. Выберите маркер, если ответ бота не соответствует теме запроса;
- **Мусор**. Отметьте реплику бота, если она избыточна.

Для просмотра вспомогательной информации воспользуйтесь кнопками:

- **Техническая информация**. Выберите пункт, чтобы открыть метаданные диалога в формате JSON. Например, по ней можно определить идентификатор сессии, он указывается в параметре **session_id.** А в параметре **request_id** хранится идентификатор запроса по конкретному шагу диалога.



- **Посмотреть граф**. Нажмите на кнопку, чтобы перейти в сценарий, который был выбран для прохождения на данном шаге.

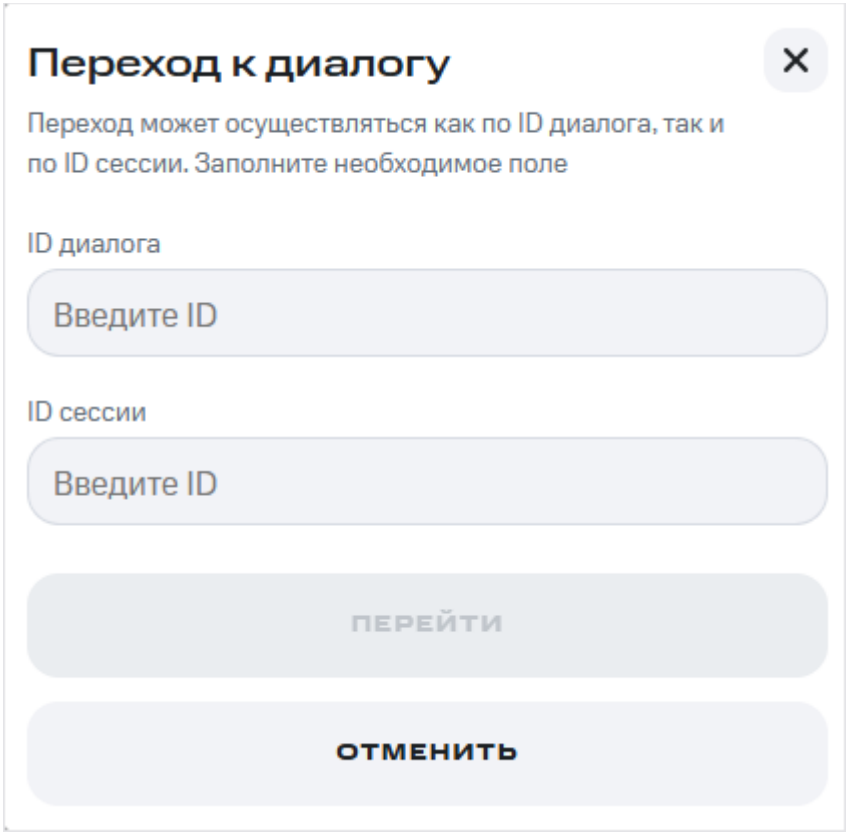
6. При необходимости оцените диалог в целом. Для этого выберите одну из оценок:
   - – успешный диалог;
   - – неудачный диалог;
   - – под вопросом;
   - – мусор.
7. Завершите работу с разметкой. Для этого нажмите на кнопку **Закончить разметку**.

Поиск диалога

Чтобы перейти к нужному диалогу, вы можете найти его по идентификатору или отфильтровать список по определенным критериям.

Переход по ИД

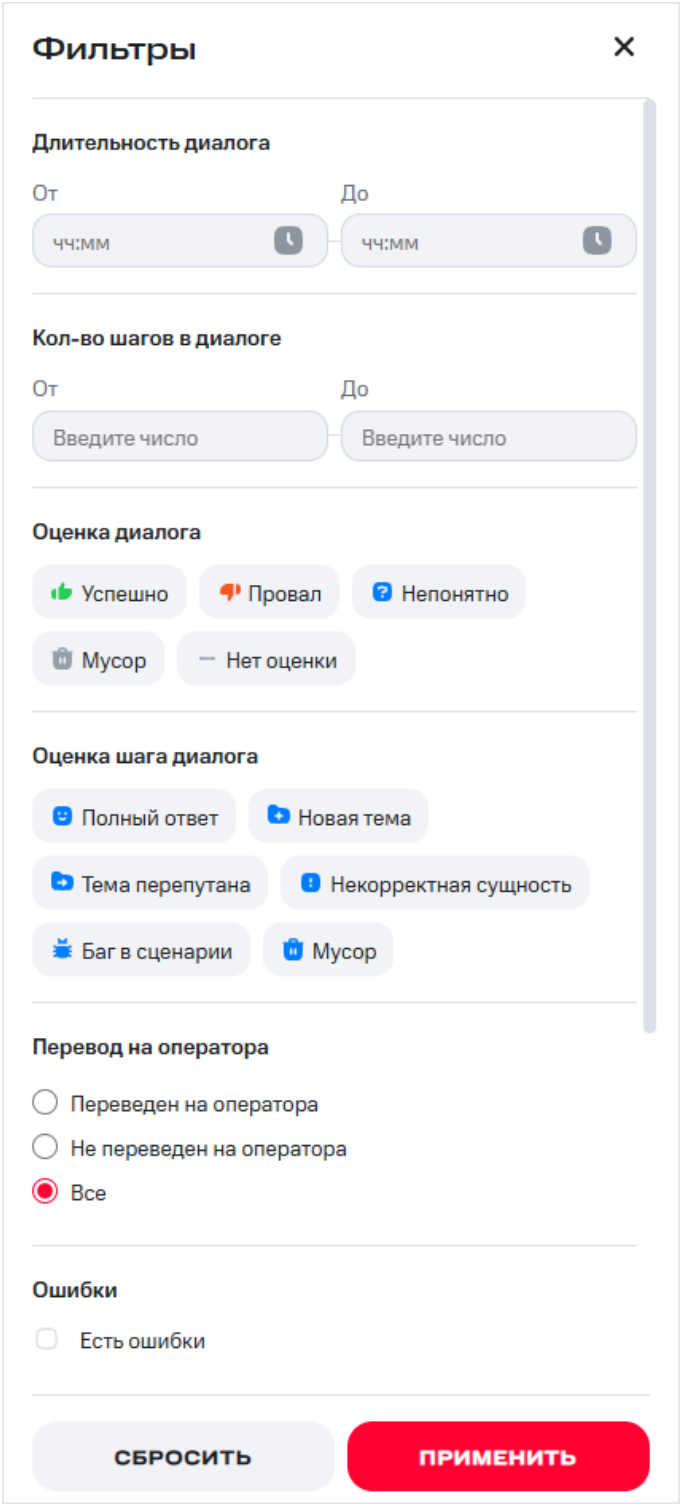
По кнопке **Перейти по ID** открывается модальное окно:



Укажите значение в одном из полей **ID диалога** или **ID сессии** и нажмите на кнопку **Перейти**.

Фильтрация диалогов

Вы можете отфильтровать диалоги по длительности, количеству шагов, оценке диалога и т.д. Для этого по кнопке **Фильтры** откройте панель фильтрации и укажите нужные критерии. Примените выбранные фильтры по кнопке **Применить**.



Обучение классификатора

После разметки экспортируйте размеченные диалоги и привяжите классификатор:

1. Перейдите на страницу **Диалоги**.
2. Чтобы сформировать файл для обучения классификатора, отфильтруйте диалоги. Для этого откройте панель фильтрации по кнопке **Фильтры**, установите нужные признаки и

нажмите на кнопку **Применить**. Например, отфильтруйте диалоги по признаку «Тема перепутана».

3. Нажмите на кнопку **Экспортировать**.
4. Перейдите на страницу **Разметка данных**.
5. Перейдите в созданный ранее проект разметки и загрузите в него сохраненный файл разметки. При загрузке выберите пункт **Сохранять разметку из файла**, чтобы перезаписать существующую разметку. Подробнее о добавлении данных см. в разделе «Добавление данных в проект». Скачайте получившийся датасет.
6. Создайте AI сервис и загрузите в него датасет.
7. Подключите сервис к боту. Для этого перейдите в конструктор нужного бота. Создайте новую версию и в ее настройках в поле **Классификатор** выберите ранее обученный классификатор.

   Если в классификаторе появились новые классы, привяжите их к сценариям.
8. Протестируйте бота и опубликуйте в канале.

AI-сервисы

AI-сервисы — это технологические решения, использующие искусственный интеллект для обработки информации и решения задач в самых разнообразных сферах. Они могут выполнять автоматизированный анализ данных, распознавать изображения и речь, выполнять автоматический перевод, предсказательное моделирование, систематизировать неструктурированные данные и многое другое. В платформе MWS AI Agents Platform сервисы включают автоматизированное обучение моделей на пользовательских данных, не требуя экспертизы в ML (Machine Learning).

Ключевые преимущества AI-сервисов платформы:

- **No-code подход**: создавайте и обучайте модели без программирования.
- **Масштабируемость:** автоскейлинг и батчинг для обработки пиковых нагрузок.
- **Интеграция:** с LLM (Large Language Models), базами знаний и ботами.
- **Гибкость:** управляйте жизненным циклом сервисов (создание, запуск, остановка, удаление).

Типы AI-сервисов платформы

Платформа поддерживает RAG и AutoML-сервисы, включая классификатор и NER. Каждый из этих сервисов оснащен встроенным обучение или индексацию, чтобы адаптироваться к вашим данным. Для обучения используются наборы данных – датасеты или данные, размеченные в соответствующем разделе платформы Разметка данных.

- **Сервис RAG –** сервис для интеллектуального поиска и генерации ответов с использованием технологии RAG (Retrieval-Augmented Generation). Автоматически индексирует большие массивы документов (базы знаний). При ответе на вопрос ищет в базе релевантную информацию, а затем использует найденные сведения для формирования запроса к LLM. В итоге сервис возвращает ответы LLM, подкрепленные ссылками на источники. Идеален для задач, требующих точности и обоснованности. Например, ответы на вопросы клиентов к корпоративной юридической службе.
- **Классификатор** (Classifier) – AutoML-сервис для классификации текстов по классам (например, "запись на прием" или "запрос информации"). Обучается на наборах данных, в которых особое внимание уделено равномерному распределению классов. Это позволяет модели точно определить ключевые намерения пользователя в запросе и направить в нужный сценарий бота.
- модель, распределяющая входные данные по определённым категориям, классам. Основная задача классификатора – выделить из текста запроса намерение пользователя, например, запись на приём, запрос консультации, уточнение диагноза, получение справочной информации.
- **NER** (Named Entity Recognition) – AutoML-сервис для распознавания в тексте именованных сущностей, таких как имена, даты, географические названия и термины. В процессе обучение к датасету NER можно добавить пользовательский словарь синонимов, что повышает точность анализа. Сервис извлекает структурированные данные из неструктурированных текстов, которые затем могут быть использованы в работе ботов.

Сравнение типов

Тип	Назначение	Формат данных	Обучение/Индексация	Ключевые опции и ограничения
RAG	Поиск и генерация ответов	Документы (PDF, DOCX и т.д., до 100 Мб)	Индексация (векторизация) баз знаний	• периодическая ручная актуализация баз знаний;

Тип	Назначение	Формат данных	Обучение/Индексация	Ключевые опции и ограничения
	с использование базы знаний			• ссылки на источники; • подходит для больших объемов данных
Classifier	Категоризация по классам	CSV (text, label; мин. 2 класса, 2 примера на класс)	Fine-tuning, SetFit, AncSetFit	• необходим баланс классов; • опциональный anc_label для малых датасетов; • мин. 81 примеров на класс для качества
NER	Распознавание сущностей	JSON (с entities; до 100 Мб)	Fine-tuning, SetFit с словарями синонимов (JSON)	• словарь загружается после датасета; • улучшает распознавание синонимов; • извлекает структурированную информацию

Как создать и использовать AI-сервис

AI-сервис создается в статусе **Черновик**. Для любого типа сервиса следуйте приведенным ниже шагам. Вы можете прервать процесс и вернуться позже — сервис сохранится.

1. Перейдите в раздел AI-сервисы и нажмите **Создать сервис**.
2. Выберите тип сервиса. Если передумали – нажмите **Отменить создание**.
3. Название генерируется автоматически как "draft-uuid", но лучше сразу ввести уникальное. Дубликаты названий подсвечиваются предупреждением «Данное имя уже занято».
4. Настройте параметры (см. разделы Настройка AutoML-сервиса и Карточка RAG-сервиса).
5. Загрузите данные (датасет или документы для базы знаний).
6. Запустите обучение/индексацию.
7. Протестируйте сервис в виджете.
8. Подключите к боту (см. разделы Подключение обученной модели к боту и Подключение бота к RAG-сервису).
9. Управляйте: запустите/остановите с помощью переключателя; удалите с подтверждением (необратимо, удаляются модель, датасет из S3 и записи в БД).

Статусы сервисов:

-
    - **Черновик**,
    - **Создаётся,**
    - **Запускается** (только Классификатор и NER),

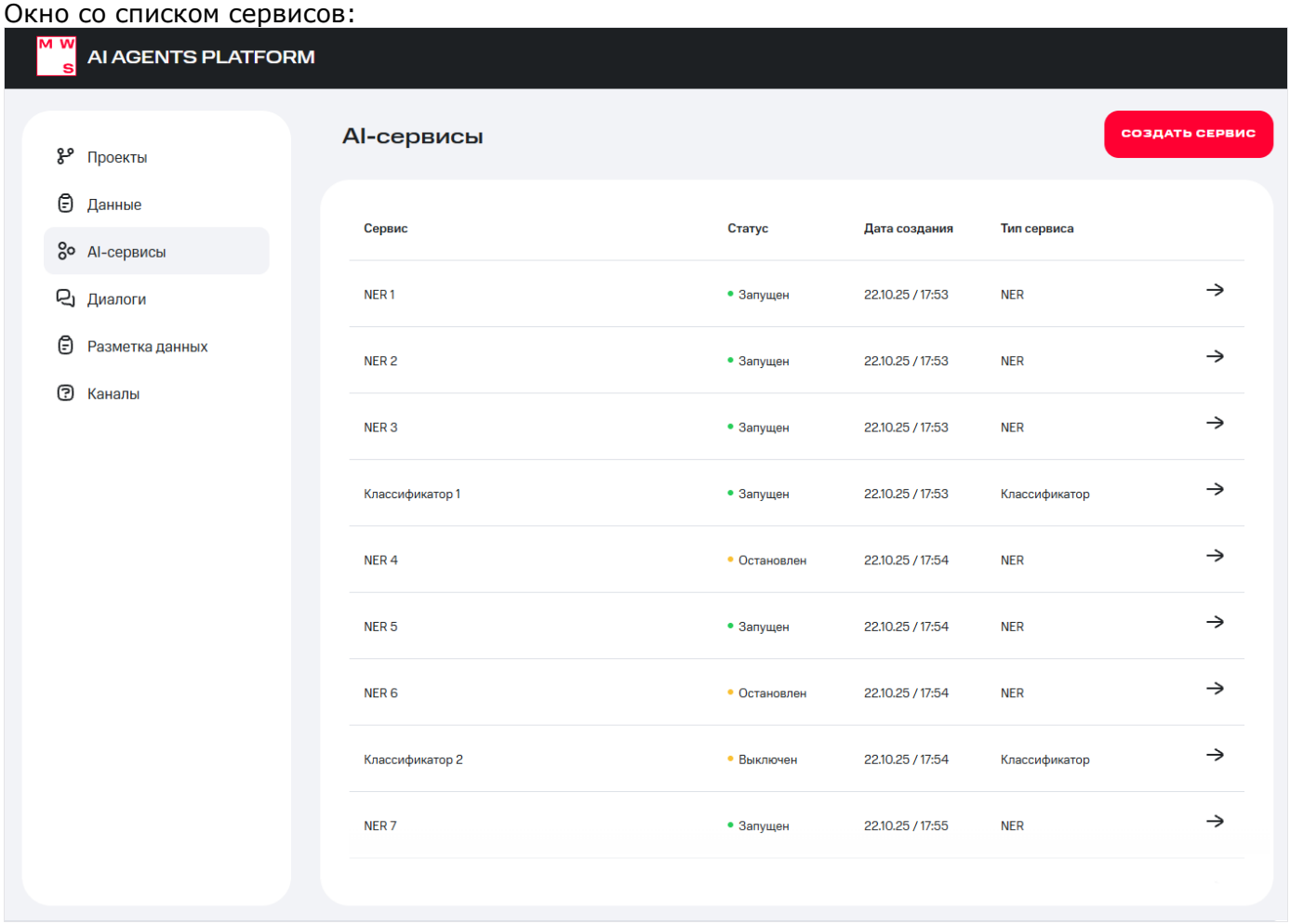
- **Запущен**,
- **Выключен**,
- **Удаляется**,
- **Ошибка.**

Пользовательский интерфейс для управления AI-сервисами

Элементы интерфейса:

Элемент	Описание	Различия элементов по типам сервиса и советы
Список сервисов	Таблица с именем, статусом, датой создания, типом сервисов, кнопка Создать сервис	• общий вид для всех AI-сервисов; • если таблица пуста, значит сервисы еще не созданы или все удалены; • кликните строку таблицы для перехода в карточку AI-сервиса
Создание сервиса	Здесь выполняется базовая настройка и загрузка данных сервиса	• для RAG – содержит поле для загрузки документов с сайта или с ПК; • для NER – поле для словаря; • скачайте шаблон датасета для образца; • проверьте форматирование датасета перед загрузкой; • отмените создание сервиса для удаления черновика
Карточка сервиса	Полная информация о сервисе на вкладках **Детали** и **Настройки**	Для RAG — просмотр индекса; для NER — словарь; редактируйте только в остановленном состоянии; сохраните изменения для применения
Тестирование	Виджет для ввода запроса к сервису и просмотра ответа в формате JSON	• Классификатор: label/score; • NER: labels (text, label, start/end, score); • RAG: ответы + источники; • доступно только для запущенных сервисов

Окно со списком сервисов:



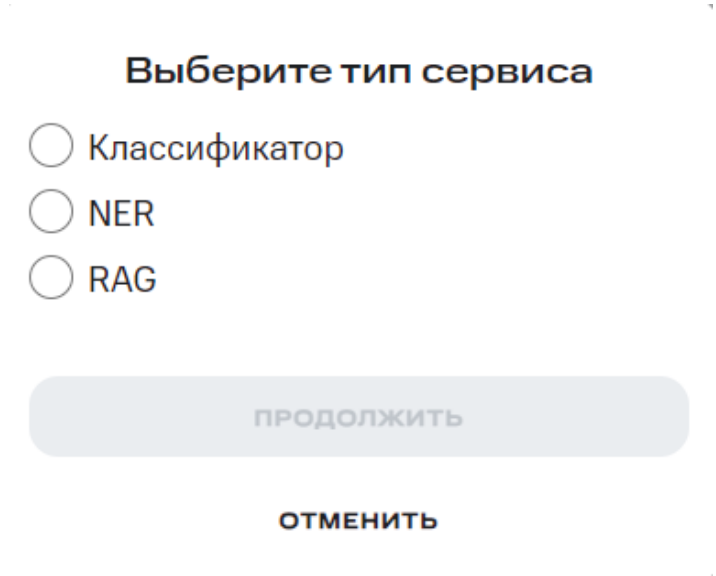
AutoML-сервисы

AutoML-сервисы автоматизируют разработку и эксплуатацию ML-моделей. Поскольку такие сервисы требуют значительных вычислительных ресурсов, рекомендуется использовать встроенные механизмы автоскейлинга и батчинга для их оптимизации (подробнее см. раздел «Настройка AutoML-сервиса»).

Общий процесс для Классификаторов и NER

Чтобы создать новый AutoML-сервис типа Классификатор и NER:

1. Перейдите в раздел **AI-сервисы** и нажмите кнопку **Создать сервис** – откроется окно **Выберите тип сервиса**.



2. Выберите тип AutoML-сервиса.
3. Нажмите – **Продолжить**.
4. Откроется окно **Создать сервис**. В поле **Название сервиса** будет автоматически сформировано название, состоящее из стартового статуса сервиса – draft (черновик) и его uuid – 36-символьного универсального уникального идентификатора. Измените стартовое название. Если сейчас в этом нет необходимости, вы сможете переименовать сервис в любой другой момент в Карточке сервиса.

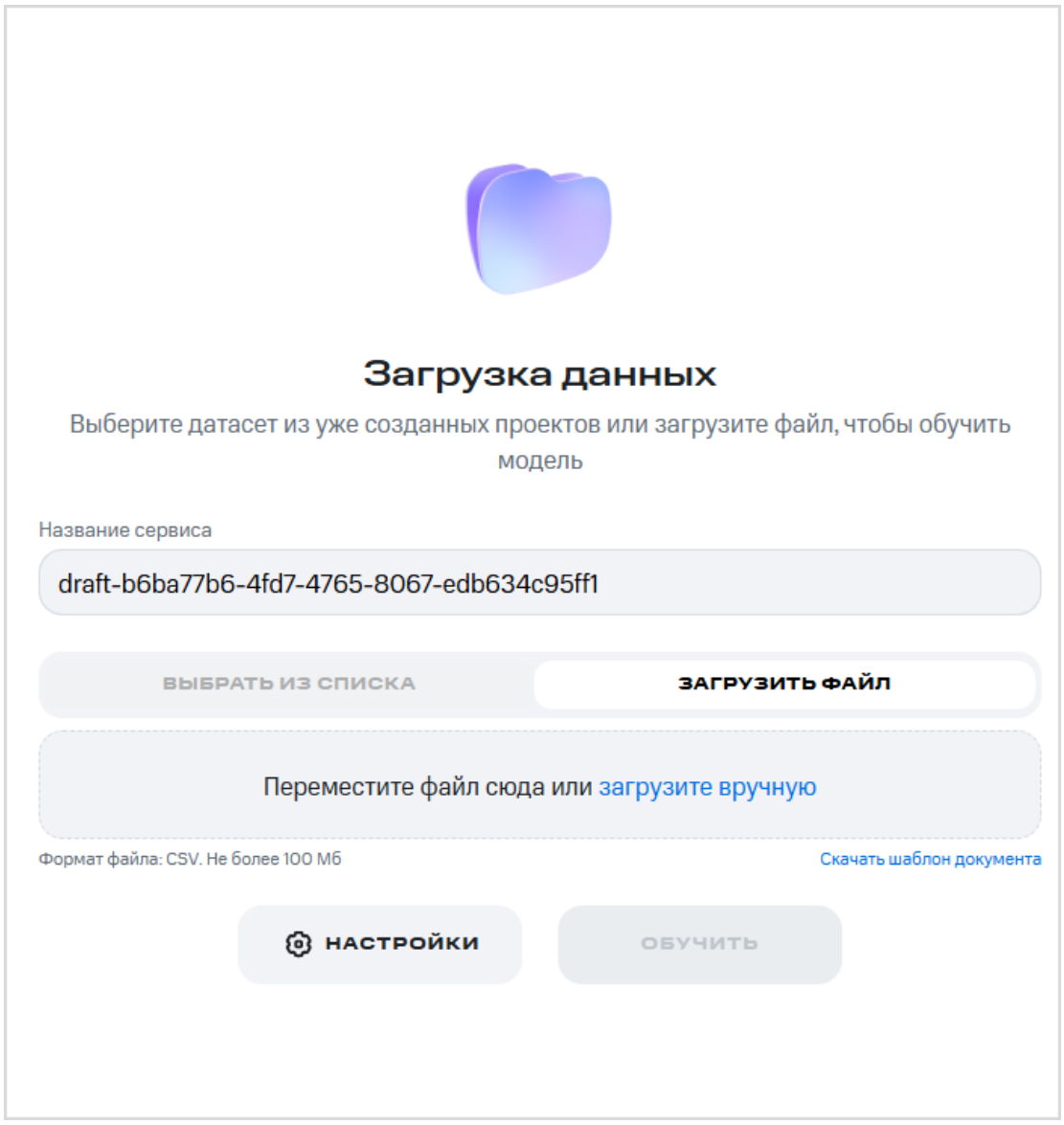
AutoML-сервис создан и готов к настройке и загрузке данных для обучения модели.

Различия между Классификатором и NER

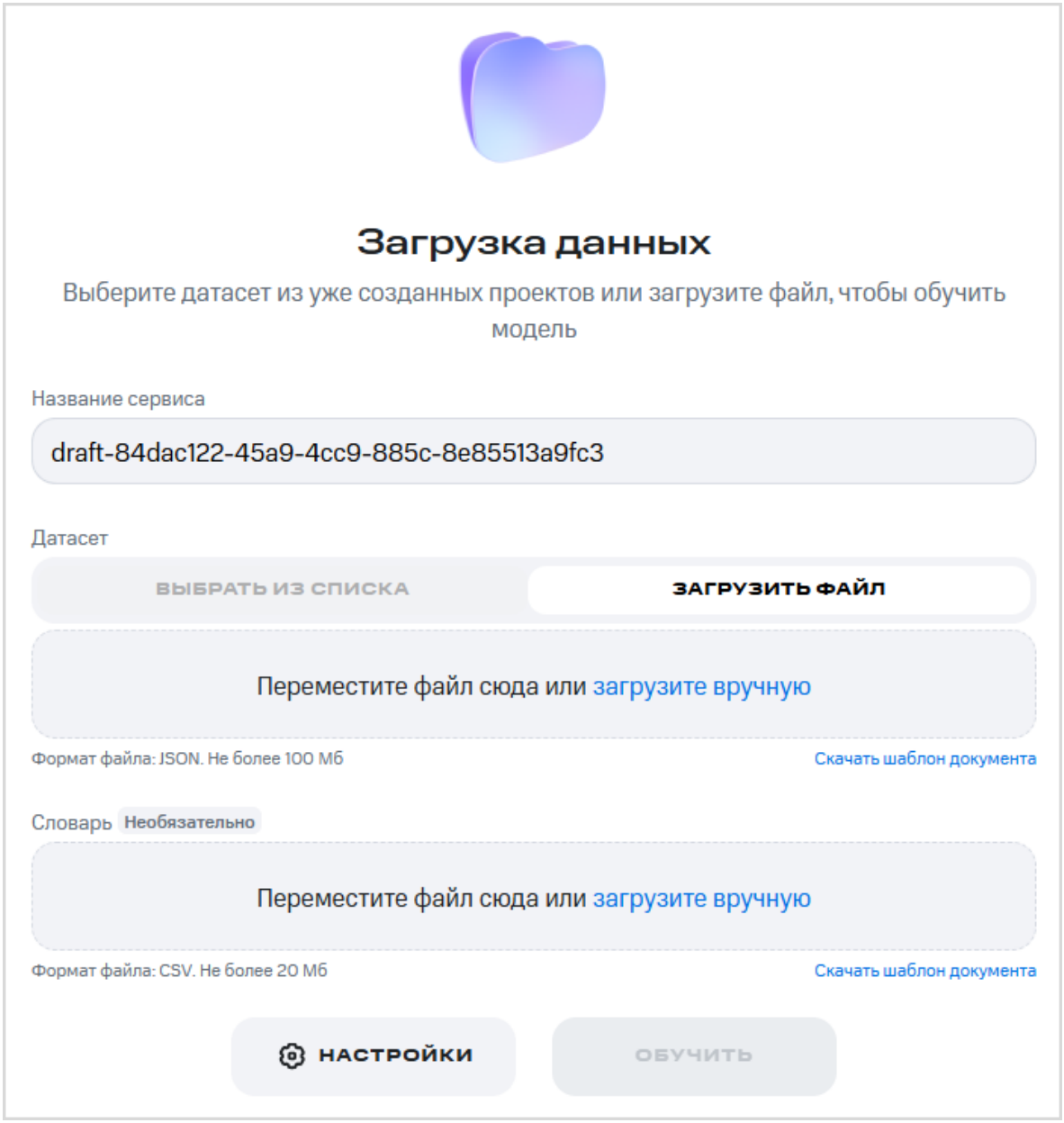
Классификатор и NER имеют схожий интерфейс, но отличаются по формату и опциям.

Аспект	**Классификатор**	**NER**
Цель модели	Распределение по классам	Распознавание сущностей
Формат датасета	CSV	JSON
Дополнительно данные для загрузки	Нет	Словарь синонимов CSV
В карточке	Классы	Сущности, просмотр словаря, F1-мера, confusion matrix
Тестирование	Top-N классов, score	Массив сущностей (text, label, score), top-n

Окно Создание сервиса (Классификатор)



Окно Создание сервиса (NER)



Настройка AutoML-сервиса

ML-модели требуют значительных вычислительных ресурсов. Если нагрузка пиковая, система может «задохнуться»: задержки вырастут, запросы начнут падать, серверы с развернутыми моделями могут выйти из строя.

Чтобы оптимизировать нагрузку, в MWS AI Agents Platform реализованы стратегии автоскейлинга и батчинга.

- **Автоскейлинг** – это автоматическое горизонтальное масштабирование, при котором система в зависимости от нагрузки динамически добавляет или удаляет реплик (копии) ML-моделей. Автоскейлинг отслеживает метрики ресурсов системы. При достижении метриками установленных значений автоматически запускаются дополнительные реплики моделей. Когда нагрузка падает, лишние реплики удаляются, чтобы не тратить ресурсы зря.
- **Батчинг** – это метод обработки данных, при котором несколько входящих запросов группируются в один "батч" (пакет) и обрабатываются моделью одновременно. Вместо

обработки запросов по одному, модель получает их пачкой, что оптимизирует использование аппаратных ресурсов. Батчинг увеличивает количество обработанных запросов в единицу времени. Запросы накапливаются в буфере до достижения заданного размера батча, длительности задержки или таймаута, чтобы не задерживать слишком долго.

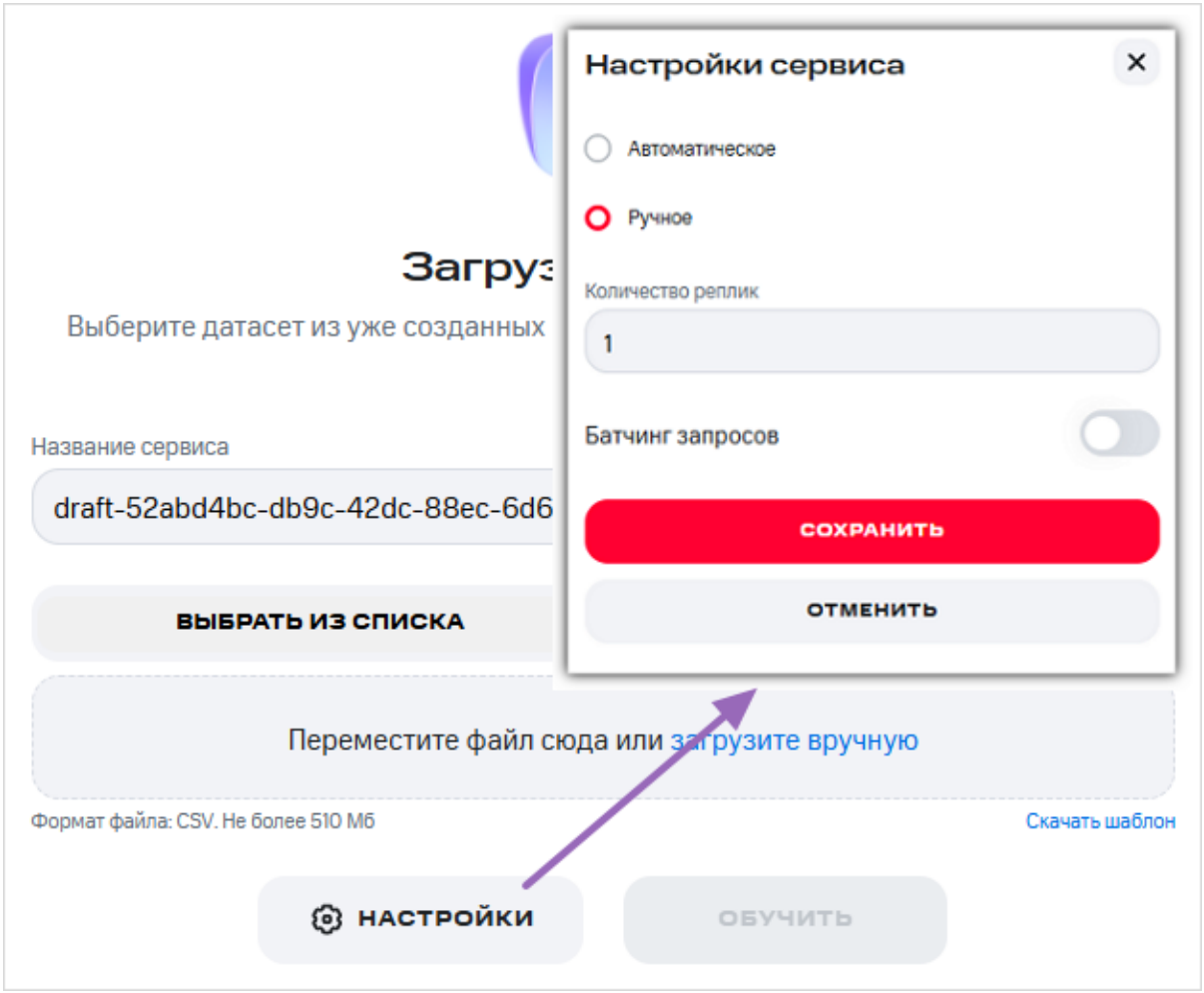
Если автоскейлинг и батчинг «работают» вместе, то при пиковом увеличении количества запросов батчинг группирует их в пакеты, а если при этом возрастет нагрузка на аппаратные ресурсы – автоскейлинг запустит дополнительные реплики модели.

Платформа поддерживает ручной режим скейлинга, когда количество инстансов модели задается при создании AutoML-сервиса и балансировка выполняется между ними без репликации.

По умолчанию новый AutoML-сервис создается со скейлингом, установленным в ручном режиме, и отключенным батчингом запросов.

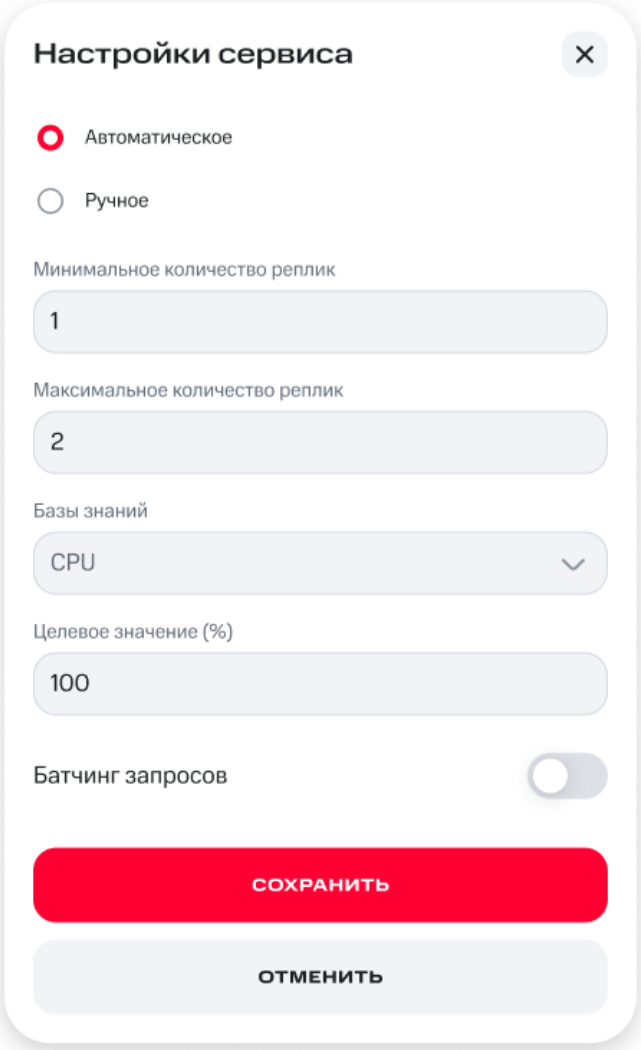
Чтобы изменить настройки:

1. На странице **Создать сервис**, нажмите кнопку **Настройка** – откроется диалоговое окно **Настройка сервиса**.

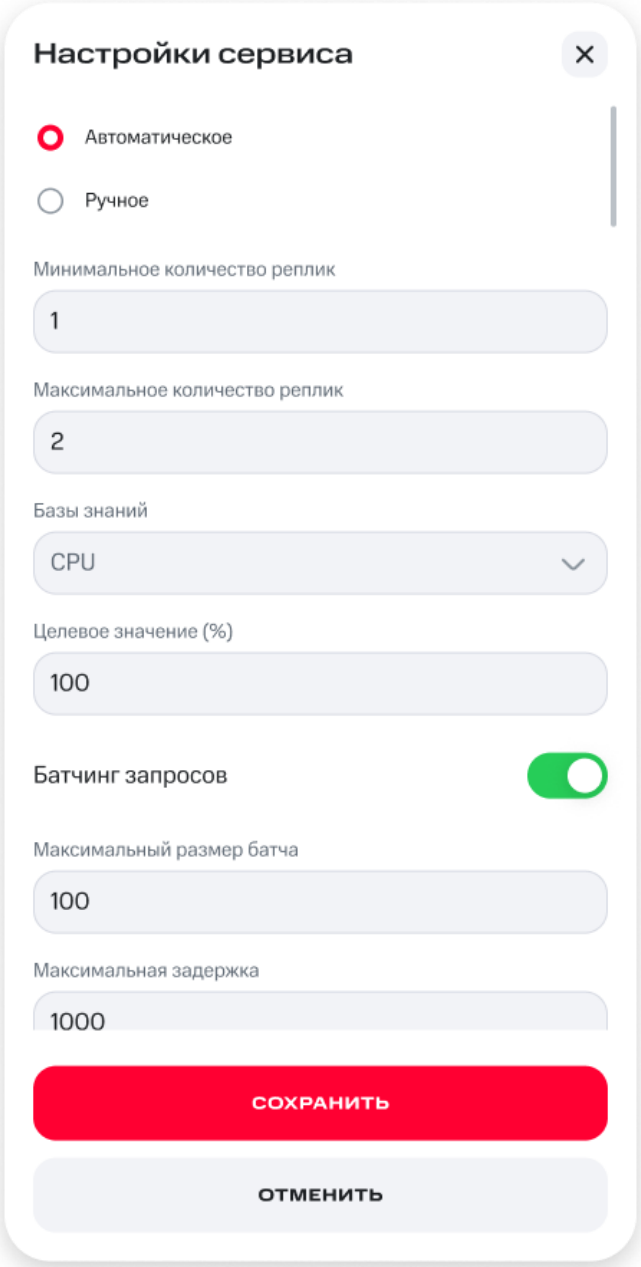


   Если скейлинг остается в ручном режиме, то установите необходимое количество реплик. Минимальное значение 1, верхний предел не ограничен. Но стоит учитывать возможности ваших ресурсов: большое число реплик может негативно повлиять на систему.

2. Если нужно включить автоскейлинг, поставьте отметку **Автоматическое** – в диалоговом окне появятся поля для его настройки.
   - Установите минимальное и максимальное количество реплик.
   - Выберите метрику значение которой будет отслеживаться – CPU или память.
   - Установите целевое значение нагрузки от 1 % до 100 %.



3. Если необходимо включить батчинг, переведите переключатель **Батчинг запросов** вправо – появятся поля для настройки.
    - Установите максимальный размер батча, т.е. максимальное количество запросов, которое может быть сгруппировано в батч для одновременной обработки моделью. Чем более высоконагруженной является ваша система, тем выше может быть значение этого параметра. При этом большой размер батча требует больше ресурсов памяти.
    - Установите значение времени максимальной задержки в миллисекундах. В течение этого времени система будет ожидать накопления запросов в батче, прежде чем отправить его на обработку. Если во время ожидания задержка достигнет установленного значения, то батч отправится в ML-модель даже если он не полный. Это предотвращает бесконечное ожидание при низкой нагрузке.
    - Установите таймаут обработки – максимальное время в секундах, отведенное на обработку одного батча или запроса моделью. Если длительность обработки превысит заданную, то запрос будет считается неудачным, а система может повторить или отклонить его. Чем сложнее задача в боте с ML-моделью (например, анализ длинного текста или генерация идей), тем больше времени нужно на обработку – устанавливайте длительный таймаут. Иначе бот может зависнуть или отклонить запрос, чтобы не тратить ресурсы зря. Для простых вопросов хватит 10 секунд, а для тяжёлых – до минуты.



- Для высокой нагрузки: большой размер батча + умеренная задержка = высокое количество обработанных запросов в единицу времени.
- Для низкой задержки ответов: маленький размер батча + маленькая задержка = быстрые ответы, но ниже эффективность.

4. Сохраните настройки.

Сервис настроен. Изменить значения параметров можно в соответствующем разделе Карточки AutoML-сервиса.

Загрузка данных для обучения ML-модели сервиса

Загрузить данные можно двумя способами:

- Выбрать существующий проект разметки данных (подробнее о разметке → Разметка данных).

- Самостоятельно создать и разметить датасет, затем загрузить его в сервис файлом со своего компьютера;

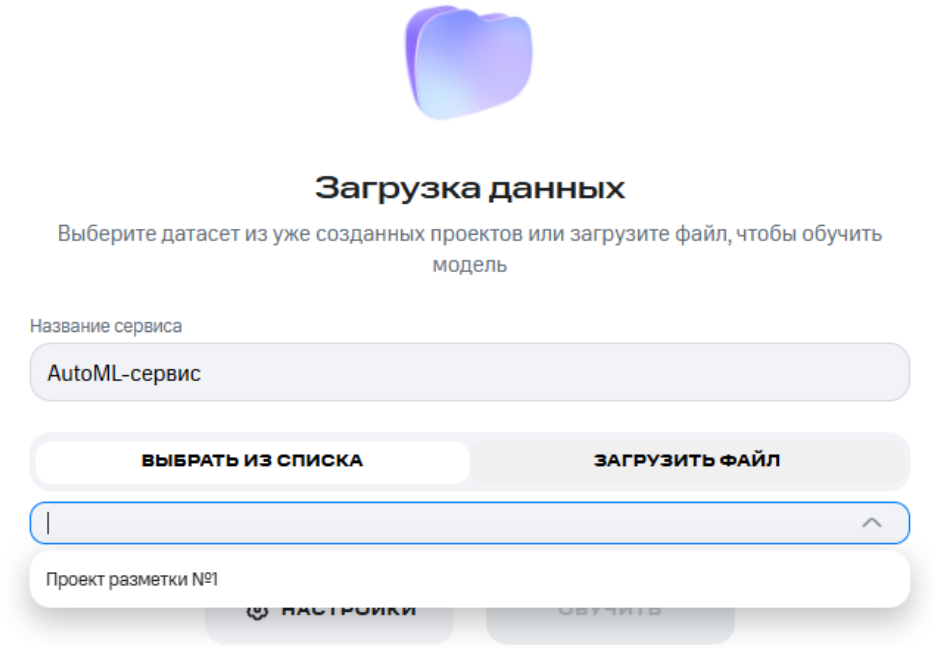
Выбор для обучения проекта разметки данных

Чтобы выбрать для обучения ML-модели готовый проект разметки:

1. Нажмите **Выбрать из списка** – появится поле выбора проекта разметки.

   Обратите внимание!
   Если в системе еще не существует ни одного проекта разметки для обучаемого сервиса, кнопка **Выбрать из списка** будет неактивна. Для ее активации необходимо создать хотя бы один проект разметки (подробнее см. здесь Создание проекта разметки данных).

2. Нажмите и выберите проект в выпадающем списке.



3. Активируется кнопка **Обучить**.

Данные загружены в сервис. Можно начинать обучение.

Загрузка файла датасета

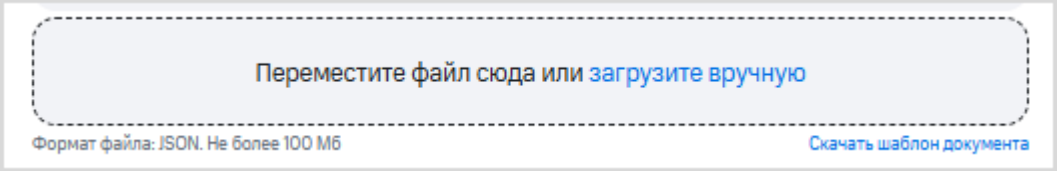
В качестве датасета для обучения моделей используются:

- Для **Классификатора** – csv-файл размером не более 100 Мб. Информация о создании качественных датасетов для классификатора представлены в разделе Рекомендации по созданию датасетов для обучения классификатора.
- Для **NER** – JSON. Не более 100 Мб.

Чтобы подготовить и загрузить файл датасета:

1. На странице **Создать сервис** нажмите **Скачать шаблон документа**. На ваш компьютер загрузится заранее заполненный и правильно размеченный датасет с тестовыми данными в формате, соответствующем типу сервиса, который вы создаете (csv или

json). Работайте прямо в скачанном шаблоне, затем сохраните его с нужным именем.



2. Готовый файл можно перетащить мышкой в область загрузки или щелкнуть в области загрузки и загрузить через проводник операционной системы. На экране вы увидите загружаемый файл и индикатор загрузки , который при успешном выполнении процесса изменится на иконку . Загрузку можно отменить, нажав на .

   В процессе загрузки датасета система проверяет его формат (расширение файла) и размер. Если файл не прошел проверку – вы увидите сообщение **Неправильный формат файла** или **Превышен максимальный размер файла**. Кнопка **Обучить** останется неактивной. Решение – смена формата на требуемый или уменьшение размера файла.

3. После того, как датасет успешно загрузится, кнопка **Обучить** станет активной. Если понадобится удалить датасет, нажмите .

   Обратите внимание!
   На этом шаге создания сервиса вы можете перейти в другой раздел веб-интерфейса платформы или выйти из системы. Создаваемый вами сервис сохранится как **черновик,** и вы сможете продолжить работу с ним позже. Файла датасета останется прикрепленным. Обратите внимание, если вы уже успели переименовать сервис при создании, то его название вернется к изначальному системному формату – draft– uuid

Файл датасета загружен. Можно начинать обучение.

Загрузка пользовательского словаря синонимов сущностей для NER

Словарь синонимов сущностей – это CSV-таблица, в которой для каждого типа определенной в датасете NER сущности (label) хранятся:

- Нормализованная форма (каноническое название);
- Варианты написания (синонимы для поиска и нормализации).

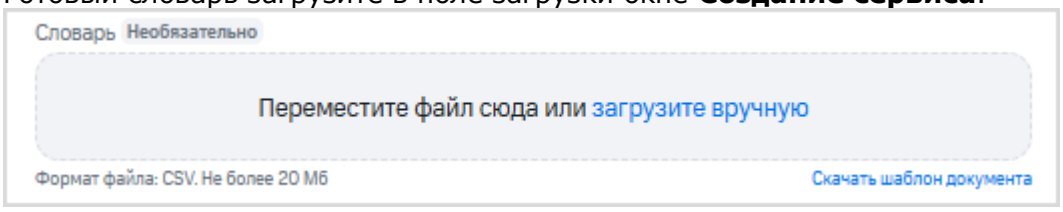
Загрузка словаря – это не обязательное действие. При этом словарь улучшает NER для домен-специфических терминов, например, таких как медицинские симптомы. Модель с интегрированным словарем учитывает синонимы в ответах, что может быть полезно для ботов с вариативным вводом (например, «Питер» вместо «Санкт-Петербург»).

Словарь является неотделимой частью сервиса и загружается только при создании. Удалить словарь из сервиса нельзя. Чтобы модель перестала использовать словарь в ответах, вам придется пересоздать сервис без словаря.

Чтобы создать и загрузить словарь:

1. Создайте NER-сервис. Загрузите в него датасет. Это обязательный шаг – без датасета на одном словаре обучить NER невозможно.
2. В окне **Создание сервиса** скачайте шаблон словаря.
3. Работайте со словарем прямо в скачанном файле. Заполните его в соответствии с форматом (см. ниже).

4. Готовый словарь загрузите в поле загрузки окне **Создание сервиса**.



5. Система проверит файл. И если выявит не соответствие правилам валидации – отобразит сообщение рядом с полем загрузки словаря.
6. Кнопка обучение станет активна.

NER-сервис готов к обучению с интеграцией пользовательского словаря синонимов.

Формат csv-файла словаря

```
label, canonical_name, aliases
исследование, МРТ головного мозга, магнитно-резонансная томография головного мозга\tМРТ ГМ
исследование,КТ брюшной полости,компьютерная томография брюшной полости\tКТ БП
лекарство,ацетилсалициловая кислота,аспирин\tАСК\tацетилсалициловая кислота
симптом,головная боль,цефалгия\tголовная боль\tболь в голове
```

Колонка CSV	Описание	Пример
label	Тип сущности из датасета NER	исследование, лекарство, симптом
canonical_name	Нормализованная (каноническая) форма	МРТ головного мозга, ацетилсалициловая кислота, головная боль
aliases	Варианты написания (синонимы через \t)	магнитно-резонансная томография головного мозга\tМРТ ГМ

Управление словарем

Действие	Как выполнить	Результат
Загрузка словаря	Только при создании сервиса	Выбрать CSV-файл и загрузить
Просмотр статуса словаря	AI-сервисы → Карточка сервиса → Детали сервиса	Отображается поле **Словарь**.
Скачивание словаря	AI-сервисы → Карточка сервиса → Детали сервиса	Скачать текущий словарь в CSV-формате
Удаление словаря	✕Невозможно	Удалить словарь нельзя. Только пересоздать сервис
Замена словаря	✕Невозможно	Заменить словарь нельзя. Только пересоздать сервис

Ограничения для словарей

Тип ограничения	Описание
Размер файла	Максимальный размер – 20 Мб
Формат	Только CSV

Тип ограничения	Описание
Типы сущностей	Только из датасета, на котором будет обучен NER
Количество словарей	1 словарь на сервис

Что точно НЕ получится выполнить

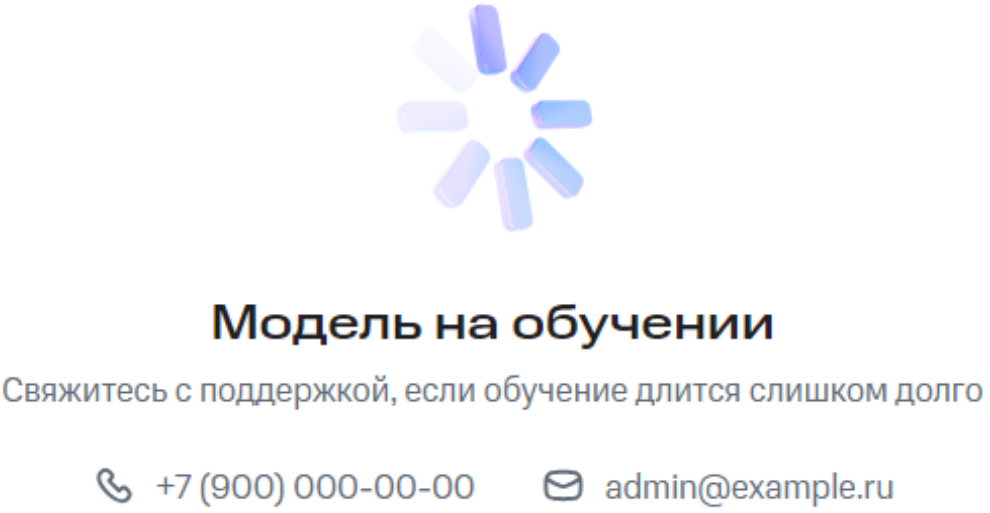
Сценарий	Результат
Загрузить словаря ПОСЛЕ создания сервиса	✗Не работает
Создать NER-сервиса только со словарем (без датасета)	✗Не работает
Удалить словарь из NER-сервиса	✗Не работает
Заменить словарь в обученном сервисе	✗Не работает
Использовать в словаре только сущности, которых нет в датасете	✗Не пройдет валидацию
Использовать в словаре некоторые сущности, которых нет в датасете	✗Строки словаря не попадут в ответ модели

Обучение ML-модели

Способы обучения ML-моделей
При загрузке датасета система автоматически выбирает оптимальный метод обучения. Выбор зависит от количества примеров в датасете и их распределения по классам. Система всегда стремится выбрать метод, который даст наилучший результат именно для ваших данных.
Вот какие методы используются:
Fine-tuning (Дообучение)
Мощная предобученная языковая модель адаптируется под конкретную задачу. К модели добавляется новый слой для категорий, представленных в датасете, после чего вся модель или её часть дообучается на этих данных. Fine-tuning выбирается в качестве способа обучения для датасетов с 81+ примеров на каждый класс и хорошим балансом данных. Обучение может занимать длительное время. При достаточном объёме данных достигается максимальная точность обученной модели.
SetFit
Модель учится различать тексты, сравнивая их между собой – какие похожи (из одного класса), а какие отличаются (из разных классов). После этого обучается компактный классификатор, который работает с векторными представлениями текстов. Способ обучения определяется для датасетов, содержащих от 8 до 80 примеров на класс. Обучение происходит быстрее, чем Fine-tuning. В итоге даже на малых данных получается хорошее качество модели.
AncSetFit (Anchored SetFit)
Усовершенствованная версия SetFit, которая автоматически находит в датасете наиболее типичные примеры для каждого класса (якоря) и использует их как эталоны при обучении. В нашей системе якоря указываются в столбце anclable на русском языке. Это помогает модели лучше понять суть каждой категории. Способ обучения задействуется, если в датасете от 2 до 5 классов. Лучшие результаты обучения на минимальном количестве данных.

Чтобы начать обучение модели, нажмите кнопку **Обучить**.

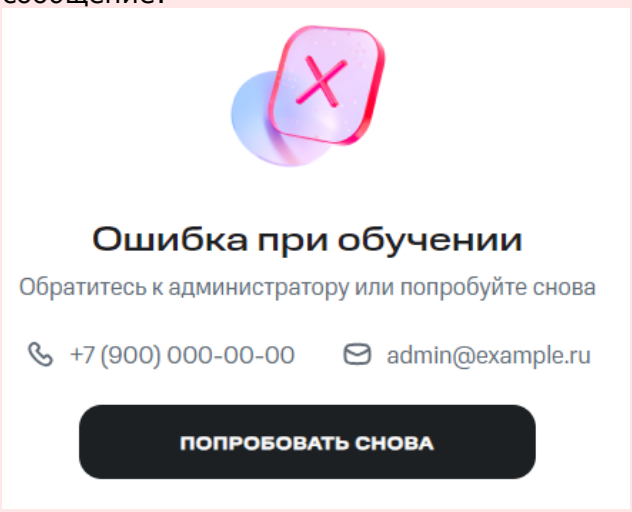
1. На экране появится сообщение **Модель на обучении**.



2. Обучение модели занимает некоторое время. Пока процесс идет, ничто не мешает вам продолжить работу в других разделах MWS AI Agents Platform. При этом вы можете периодически проверять статус вашего сервиса в разделе **AI-сервисы**. В ходе процесса система выполняет проверку качества датасета. Если эта проверка занимает длительное время вы можете увидеть сообщение **Проверка данных**. Но обычно проверка выполняется быстро и это сообщение не появляется.

Если в процессе обучения возникнет необходимость отменить создание сервиса, используйте кнопку **Отменить создание** – она активна в течение всего процесса обучения. Отмена создания подразумевает полное удаление сервиса.

Если при обучении возникает ошибка, на экране отображается вот такое сообщение:



На странице **Список сервисов** сервис, получивший ошибку в процессе обучения, приобретает статус **Ошибка**. Открыв его карточку, вы встретитесь с приведенным выше сообщением.
Что можно сделать для исправления ситуации:

- Нажать кнопку **Попробовать снова**.
- Если ситуация не изменилась – самостоятельно проверьте датасет на ошибки. Если ошибки найдены – нажмите кнопку **Отменить создание,** подтвердите удаление сервиса и попробуйте создать новый, используя исправленный файл датасета.

- Если ошибки не нашлось или ситуация повторяется и с новым датасетом – обращайтесь по указанным в сообщении контактам.

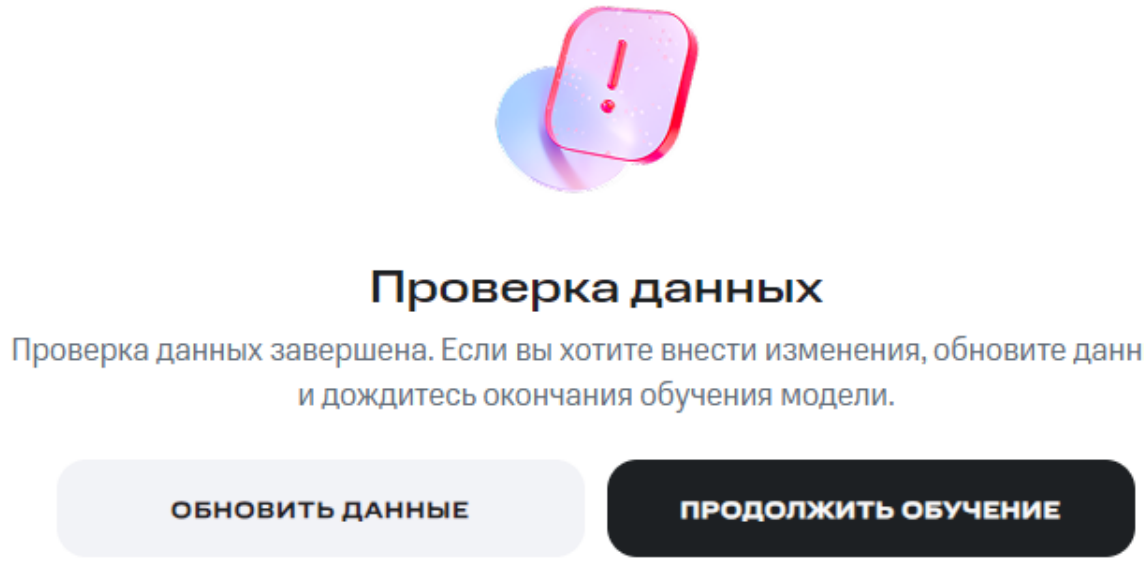
3. Когда обучение успешно завершится, откроется карточка сервиса и начнется процесс его запуска, т.е. развертывания готовой ML-модели в рабочей среде. Статус сервиса в списке изменится на **Запускается**.

Если во время запуска модели открыть карточку сервиса – в графе **Результаты обучения** будет отображаться положительный статус: **Обучение модели успешно завершено**.
Ход запуска модели показывает лоадер в графе **Статус**. Отключить сервис во время запуска нельзя, переключатель заблокирован.

Обучение модели успешно завершено.

Остановка обучения в ходе проверки данных для переразметки

Если по результатам проверки данных будет определено, что датасет недостаточно качественный, процесс обучения модели остановится.



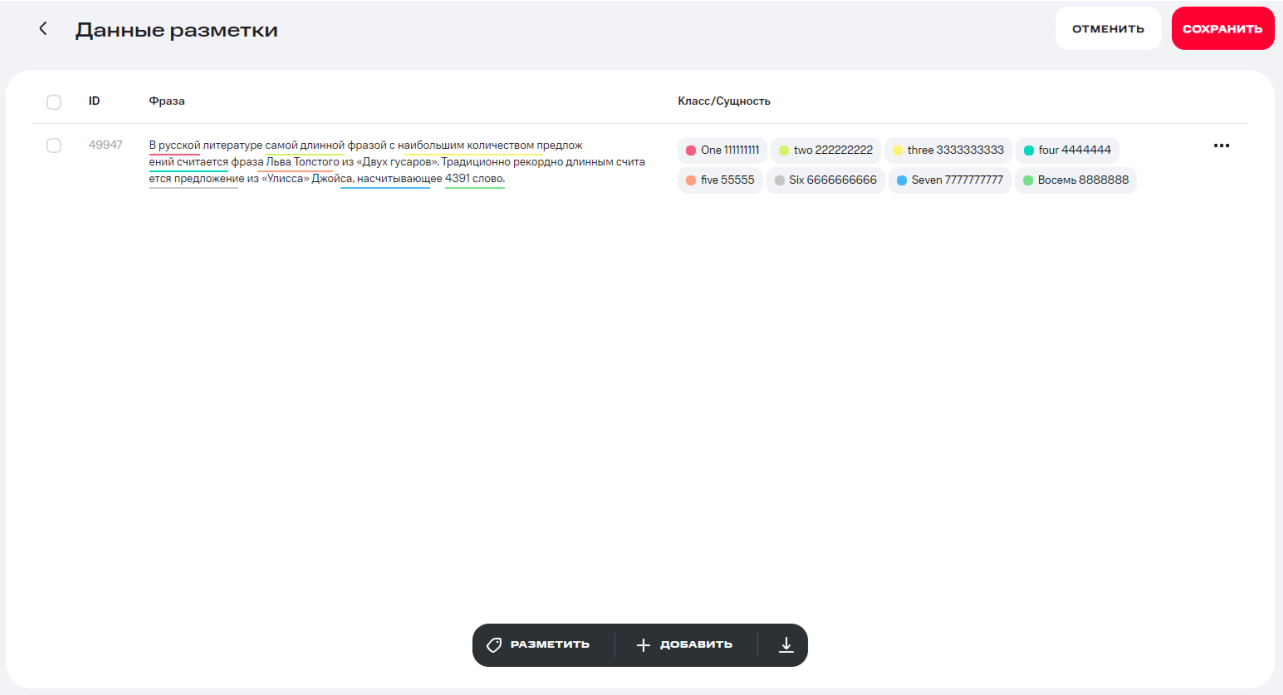
Смысл этой остановки в том, чтобы предоставить пользователю возможность обновить данные, т.е. переразметить датасет в сервисе Разметка (подробнее здесь Разметка данных). А затем продолжить обучение модели на обновленном датасете.
При этом остается возможность продолжить обучение и без переразметки, на текущем датасете. Для этого нажмите **Продолжить обучение**.

Чтобы обновить данные:

1. Нажмите **Обновить данные** – откроется страница **Данные разметки.** На странице в виде списка представлены те данные, которые в ходе проверки были определены как некачественные.

Чтобы пользователь мог работать именно с некачественными данными, под капотом системы выполняется следующий алгоритм.

- Из датасета, который был загружен в AutoML-сервис при его создании, удаляются валидные данные полученные по результатам проверки. Для оставшихся данных генерируются метки (лейблы).
- По шаблону создается новый проект разметки, основанный на данных и лейблах первого шага.
- Создается копия этого проекта и именно в ней работает пользователь.



2. Внесите правки. Подробнее о разметке данных см. в разделе Разметка данных.

Внимание!
Обратите внимание, что внесенные изменения нельзя отменить. Если вы ошиблись и нужно вернуться в первоначальное состояние вариант только один:

- Нажмите **Отменить** – вы окажетесь на экране **Проверка данных**.
- Нажмите **Обновить данные** – откроется страница **Данные разметки**.

3. Нажмите кнопку **Сохранить** – откроется страница **Данные разметки**.
4. Нажмите **Продолжить обучение**.

Обучение продолжится на исправленных данных.

Карточка AutoML-сервиса

Карточка сервиса содержит детальную информацию о сервисе и одновременно выполняет роль «пульта управления».

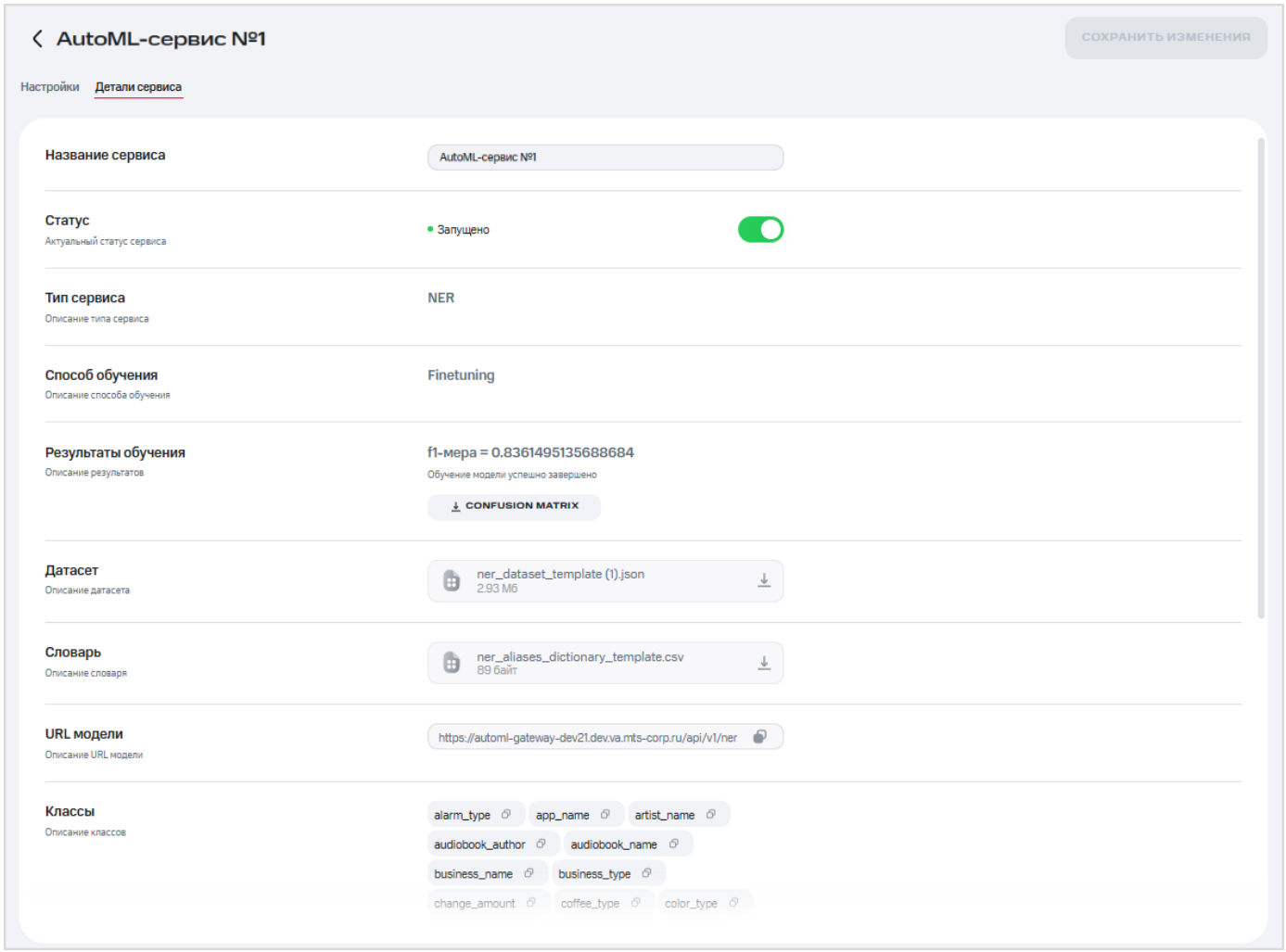
Все изменения, сделанные в карточке AutoML-сервиса, вступают в силу только после нажатия кнопки **Сохранить изменения**.

Вкладка Детали сервиса

На вкладке вы найдете:

- **Название сервиса** – редактируется.
- **Статус** – текущий статус сервиса и переключатель, запускающий или останавливающий работу сервиса.
- **Тип сервиса** – Классификатор или NER.
- **Способ обучения** – здесь наименование метода, которым была обучена модель. (подробнее здесь → Обучение ML-модели)
- **Результаты обучения** – данные о результатах обучения. Если сервис обучен по методу Fine-tuning, то отображаются f1-мера и confusion matrix.
- **Датасет** – указан файл датасета с возможностью скачать его. Если датасет был некачественным, дополнительно можно скачать отчет о его проверке в формате csv.

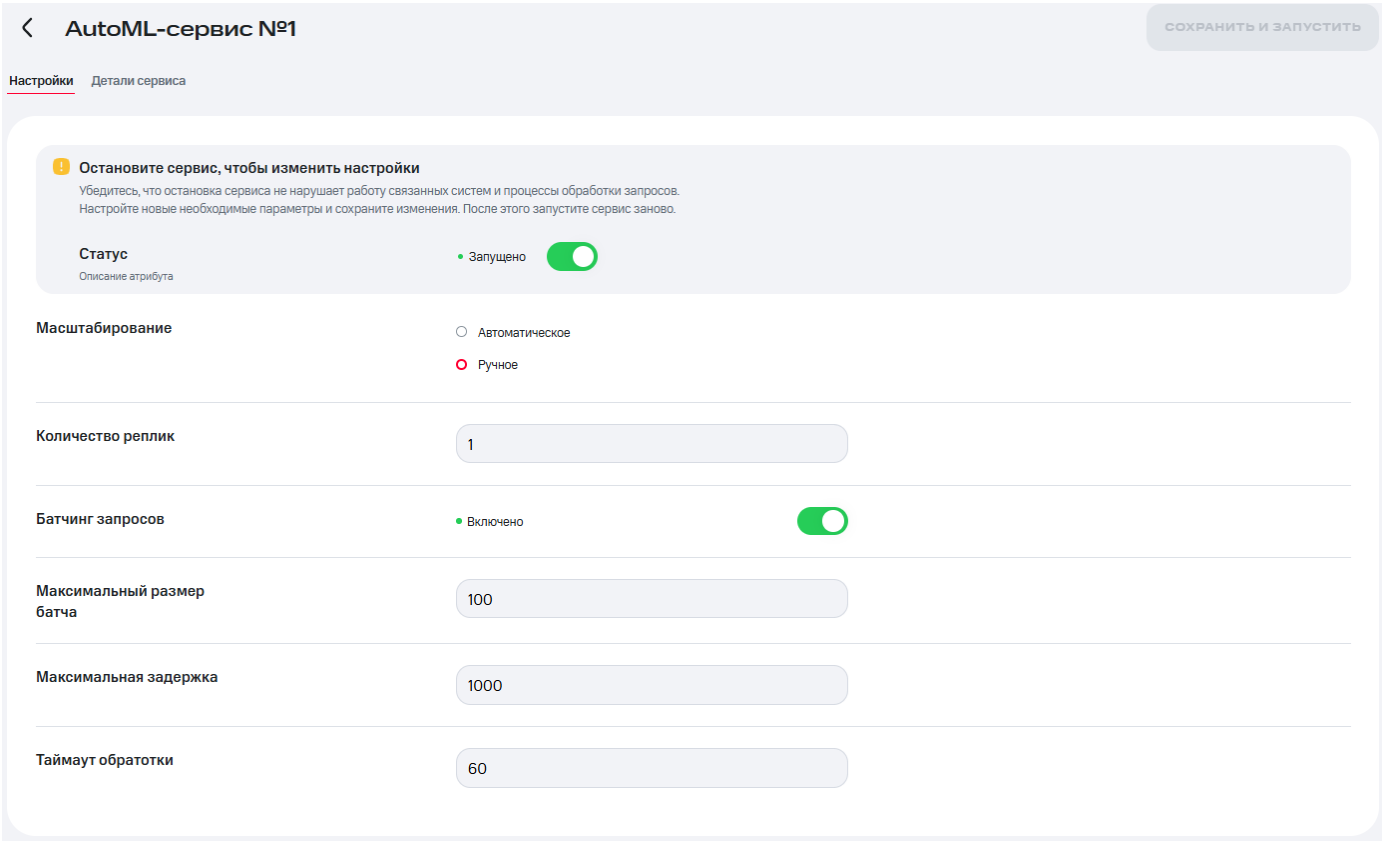
- **Словарь** (только для NER) **–** если словарь загружен – отображается название файла с возможностью скачать его. Если NER обучен без словаря – поле отсутствует в карточке.
- **URL сервиса** – адрес модели, который можно скопировать и использовать для взаимодействия с моделью как ботов, созданных в MWS AI Agents Platform (см. раздел Проекты), так и сторонних приложений.
- **Классы** – теги классов или сущностей (для NER), которые модель может определить после обучения. Тег можно копировать.



Вкладка Настройки

Здесь можно изменить параметры автоскейлинга и батчинга, заданные во время создания сервиса (подробнее здесь → Настройка AutoML-сервиса).

Обратите внимание! Редактировать параметры можно только у **остановленного** AutoML-сервиса. Пока сервис запущен, кнопка **Сохранить и запустить** – не активна.
Прежде чем остановить сервис, обязательно убедитесь, что его остановка не повлияет на работу связанных систем, так как модель сервиса может использоваться ботом.
После настройки сервиса нажмите **Сохранить и запустить**.

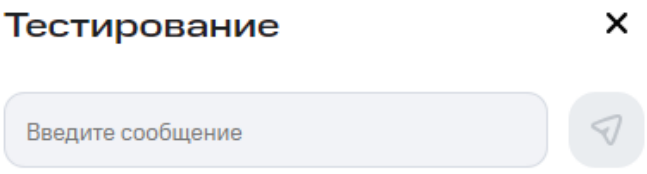


Тестирование AutoML-сервиса

Тестирование AutoML-сервиса необходимо для проверки качества ответов модели и правильности разметки данных, на которых она обучена.

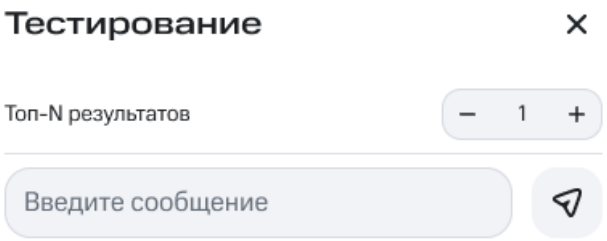
Чтобы протестировать AutoML-сервис:

1. В списке AI-сервисы найдите сервис, тестирование которого нужно провести. Войдите в Карточку сервиса.
2. Убедитесь, что сервис запущен. Если нет – запустите сервис переключателем , затем нажмите кнопку **Сохранить изменения**. Дождитесь пока сервис запустится.
3. Нажмите кнопку **Тестировать** – появится виджет **Тестирование.**



4. Можно воспользоваться функцией **Top-N**. Для классификатора с ее помощью задается количество гипотез в ответе, отсортированных в порядке убывания параметра **score** (величины, которая показывает насколько текст запроса близок к классу, который

определила модель)**.** Для NER, обученного с поддержкой пользовательского словаря, Top-N устанавливает в ответе модели количество синонимов из словаря, соответствующих найденным сущностям. Сортировка выдачи синонимов также выполняется в порядке убывания параметра **score**.



5. Введите в поле ваше сообщение и нажмите  .
6. В поле **Ответ модели** появится json с ответом.
   Для Классификатора в нем будут **label** – класс, который определила модель и **score**.

Для NER без словаря в json будет содержаться массив из сущностей, которые модель определила в вашем запросе. В каждом элементе массива: **text** – текст запроса, **label** – сущность, которую определила модель, **start/end** – положение первого и последнего символа сущности, а также **score**. Для NER с пользовательским словарем в ответе будет присутствовать дополнительный массив найденных синонимов **aliases**, содержащий информацию из CSV-колонок словаря.

Если включен батчинг, то к ответу добавиться **batchid** – идентификатор батча.

7. Вы можете задать столько вопросов, сколько необходимо для проверки сервиса. Изучите ответы модели и примите решение о качестве работы сервиса.
8. Чтобы завершить тест, закройте виджет **Тестирование** или выйдите из **Карточки сервиса**.

Тестирование AutoML-сервиса выполнено.

Подключение обученной модели к боту

Подключить Классификатор

Чтобы подключить Классификатор, обученный с помощью AutoML-сервиса, к боту:

1. На странице **Проекты** откройте конструктор нужного бота.
2. По кнопке **Версии** перейдите в настройки текущей версии. Выберите в выпадающем списке ваш AutoML-сервис, и настройте подключение, как описано в разделе Привязка классификатора.
3. Сохраните изменения кнопкой  .

Классификатор подключен к боту.

Подключить NER

Чтобы в процессе выполнения сценария бот обращался к модели NER:

1. Откройте карточку созданного сервиса типа NER и скопируйте содержание поля **URL**.
2. В разделе **Проекты** откройте конструктор нужного бота.
3. Перейдите в сценарий, в котором предполагается использовать выделение сущностей из запроса клиента. Вставьте скопированный **URL** в интеграционный блок HTTP-request и заполните параметры вызова, как описано в разделе Подключение сервиса NER.
4. Сохраните изменения кнопкой  .

При обработке запроса бот будет обращаться к NER.

Удаление AutoML-сервиса

Чтобы удалить AutoML-сервис:

1. Откройте Карточку сервиса и нажмите **Удалить**.
2. Подтвердите удаление – начнется процесс, в ходе которого будет остановлена и удалена модель, файл датасета из хранилища S3 и записи о сервисе в базе данных.

   Внимание!
   Удаление сервиса необратимо.
3. Карточка сервиса закроется. Вы переместитесь на страницу AI-сервисы. Сервис получит статус **Удаляется**. Снова открыть его карточку не получится. Через некоторое время сервис исчезнет из списка сервисов.

AutoML-сервис удален из системы.

Рекомендации по созданию датасетов для обучения классификатора

Используйте эту информацию при создании датасетов классификаторов.

Если обучаемый классификатор планируется использовать в продуктовом решении, то лучше, чтобы датасет для него содержал не менее 81 примера на класс. Кроме того, имеет значение, чтобы каждый класс в датасете был представлен примерно одинаковым количеством примеров.

Сбалансированный датасет – примерно равное количество:
"positive": 500 примеров,
"negative": 450 примеров,
"neutral": 520 примеров.

Несбалансированный датасет – сильный перекос:
"positive": 500 примеров,

"negative": 20 примеров,
"neutral": 200 примеров.

✅ При подготовке данных обращайте внимание на их качество.

Практические советы

1. Лучше меньше, но лучше. 20 качественных примеров дадут лучший результат, чем 200 сомнительных.
2. Проверяйте граничные случаи. Если сомневаетесь в категории – лучше исключите пример.
3. Используйте реальные данные. Примеры должны быть похожи на те тексты, которые модель увидит в работе.
4. Балансируйте специфичность. Нужны и простые типичные случаи, и более сложные варианты.

Перед добавлением примера проверьте:

- Понятно ли из текста, почему он относится к данной категории?
- Содержит ли текст конкретную информацию, а не общие слова?
- Относится ли текст только к одной категории?
- Встречаются ли подобные тексты в реальной работе?
- Правильно ли определена категория?

✅ Придерживаетесь следующих правил заполнения csv-файла.

- Обязательные столбцы:
  - **text** – текст для классификации на русском языке;
  - **label** – название класса, к которому принадлежит текст. Может быть любой строкой. Название класса должно совпадать с названием сценария бота, иначе при разметке истории диалогов будет отображаться некорректно.
- Опциональный столбец:
  - **anc_label** – якорь класса, содержит описание класса **только на русском языке**. Например – label: music, anc_label: музыка. Рекомендуется использовать anc_label, если хотя бы в одном из классов не более 5 примеров.
- Общие ограничения:
  - Минимальное количество классов – 2;
  - Минимальное количество примеров в классе – 2;
  - Разделитель – запятая;
  - Если в самом примере текста есть запятая, то весь пример нужно заключить в кавычки (" ").
    Пример: "можно ли построить семью с человеком, если он козерог сварщик и время рождения 14:03";
  - Если в примере есть кавычки (" "), их нужно экранировать, поставив рядом с ними дополнительные кавычки. Весь пример также нужно заключить в кавычки.
    Пример: """вкусно и точка"" доставляют сюда".

Образец правильного оформления csv-файла датасета с заполненным столбцом anc_label.

```
text,label,anc_label
закажи такси домой,taxi,такси
отвези меня к маме,taxi,такси
вызови машину на подсосненский 25/1,taxi,такси
хочу есть,food,еда
закажи пиццу,food,еда
"""вкусно и точка"" доставляют сюда",food,еда
когда меркурий будет ретроградным,astrology,астрология
рассчитай мою натальную карту,astrology,астрология
тельцы с водолеями совместимы,astrology,астрология
"можно ли построить семью с человеком, если он козерог сварщик и виталий время
рождения 14:03",astrology,астрология
```

Платформа поддерживает **OOD (Out-of-Domain)** – функцию, позволяющую модели понимать, что она столкнулась с запросом, который выходит за пределы ее обучения или специализации (то есть лежат «вне домена»). Вы можете самостоятельно определить, какие запросы будут считаться **«вне домена»** для обучаемой модели. При подготовке обучающего датасета таким данным необходимо присваивать метку **outOfScope.**

Для создания сбалансированного датасета, включающего **outOfScope,** следуйте правилам.

1. OOD не работает на низкоресурсных датасетах, когда самый минимально представленный класс содержит всего 2-5 примеров, поэтому – **чем больше примеров на минимальный класс, тем лучше**.
2. Если вы все же используете низкоресурсный датасет из пункта 1, требуется добавлять в примеры классов anc_label, при этом **НЕ ДОБАВЛЯЙТЕ ПРИМЕРЫ outOFScope! В этом режиме система с ними не работает.**
3. Если вы хотите получить лучшее качество определения примеров категории outOfScope, необходимо представить в ней примеры из следующих подкатегорий:
    - CloseOutOfScope – данные, которые близки к существующим классам по теме, но отличаются семантически.
    - MidOutOfScope– данные, которые относятся к той же общей области, что и известные модели классы, но имеют другую тематику (запросы к скиллам, которые не существуют в классификаторе)
    - FarOutOfScope – данные, которые совсем не относятся к области знаний модели.

Ниже приведен пример правильного оформления датасета с категорией outOfScope.

text;label
ответь в письме ясону что я не приду сегодня вечером;email_sendemail
alexa начни новое письмо;email_sendemail
olly отправь и-мейл мои встречи надо перенести;email_sendemail
ответить на электронное письмо роберта сегодня утром;email_sendemail
пожалуйста отправь на новый электронный адрес из списка;email_sendemail
пошли письмо моему брату и напомни годовщина свадьбы;email_sendemail
я бы хотел отправить ответ;email_sendemail
открой пожалуйста ответить на это письмо;email_sendemail
выключи свет в гараже;iot_hue_lightoff
выключи свет на кухне;iot_hue_lightoff
выключи свет;iot_hue_lightoff
выключи свет в детской спальне пожалуйста и затем измени свет в моей комнате на красный;iot_hue_lightoff
выключи свет;iot_hue_lightoff
выключи верхний свет;iot_hue_lightoff
выключи свет в ванной;iot_hue_lightoff
выключи свет;iot_hue_lightoff
какие текущие списки у меня есть;lists_query
мой праздничный список;lists_query
как называются созданные мной списки;lists_query
что в моём списке продуктов;lists_query
названия всех списков которые я веду;lists_query
убедитесь что хлеб есть в моем списке продуктов;lists_query
я хочу чтобы вы удалили пункт из списка;lists_remove
это делает меня удачливым парнем внутри очень очень;lists_remove
удали пункт списка;lists_remove
удалить фрукты из списка;lists_remove
сотри список домашних дел;lists_remove
удалить список магазинов;lists_remove
удалить песню дельтаплан из моей музыки;lists_remove
ты знаешь этот текст;music_query
какой список доступен для моей любимой музыки мити фомина;music_query
где я могу найти эту песню;music_query
показать мне музыку того артиста;music_query
что это мы слушаем;music_query
название группы;music_query
повтори эту песню когда она закончится;outOfScope #(пример из подкатегории closeOutOfScope)
пожалуйста поставь этот плейлист на перемешку;outOfScope #(пример из подкатегории closeOutOfScope)
закажи китайскую кухню из тан жен и говядину с брокколи;outOfScope #(пример из подкатегории midOutOfScope)
как выглядит моё расписание сегодня;outOfScope #(пример из подкатегории midOutOfScope)
* боли в суставах механического типа - в пр плечевом суставе, к/с справа * боли в суставах воспалительного типа - в пр плечевом суставе, к/с справа * боли в позвоночнике - в поясничном отделе - облегчение в покое, усиливаются при ходьбе, работе внаклон, периодически онемение правого бедра * ВАШ 75 см * скованность по утрам - 15 минут в суставах * «стартовые» боли - периодически в пр к/с, ПОП * ограничение движения в суставах - отведение в пр плечевом суставе, заведение правого плеча за спину * в суставах - нет;outOfScope #(пример из подкатегории farOutOfScope)
на до 180, головную боль , головокружение;outOfScope #(пример из подкатегории farOutOfScope)

Сервисы RAG

Технология Retrieval-Augmented Generation (RAG)

Сервисы RAG дополняют возможности платформы технологией Retrieval-Augmented Generation – генерацией ответа с дополненным поиском. Перед обращением к языковой модели RAG сервис сначала находит нужную информацию в своей базе знаний, а затем дополняет этой информацией свой запрос к LLM. В результате вопрос пользователя обогащается релевантными данными, что позволяет сервису отвечать максимально точно. Кроме того, сервис RAG подкрепляет ответ ссылками на документы, в которых данные были найдены. Выражаясь образно, можно сказать, что технология RAG расширяет сознание искусственного интеллекта.

База знаний

Вы можете загрузить в базу текстовые файлы, например нормативные документы или прайсы. Кроме того, платформа MWS AI Agents Platform позволяет сканировать и загружать в базу знаний данные со страниц сайтов. Для хранения файлов базы знаний используется S3-хранилище. В идеальном случае в базе знаний находится всегда свежая и релевантная информация.

Индексация базы знаний

Чтобы сервис RAG мог взаимодействовать с базой знаний, системе нужно перевести текстовые данные в числовой формат, понятный компьютеру. Для этого отдельный сервис платформы выполняет embedding: преобразует текст в вектор. Векторизованные документы помещаются в векторную базу данных. Процесс сохранения векторов называется индексация, а результат этого процесса – RAG-индекс.

Взаимодействие с сервисом

Когда сервис RAG получает вопрос, его текст также конвертируется в вектор, а затем сравнивается с RAG-индексом. Ближайшие подобные векторы RAG-индекса конвертируются обратно в текст, после чего добавляются в промпт клиентского запроса, который передается на обработку в LLM.

Чтобы получить готовый к работе RAG-сервис, выполните следующие действия.

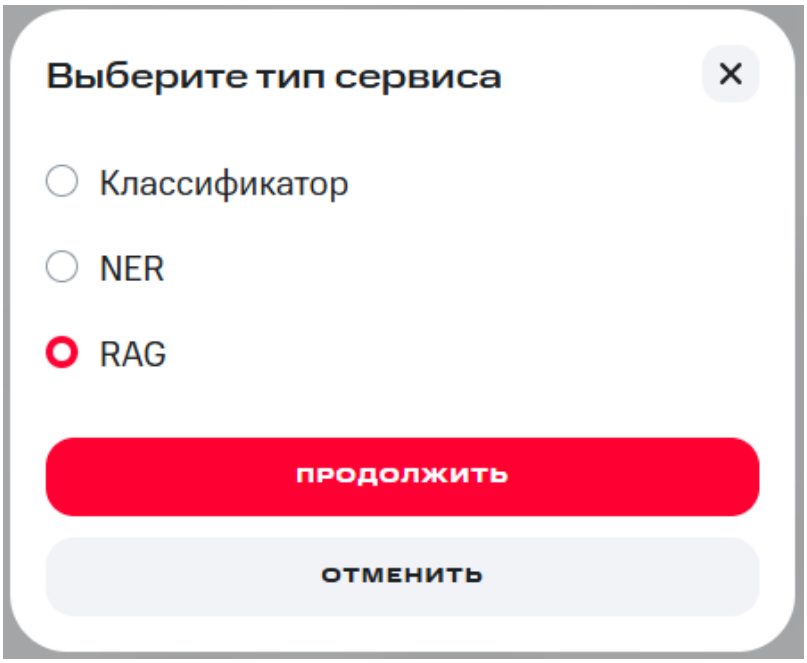
1. Создайте ваш сервис RAG в системе
2. Выберите тип загрузки документов в базу знаний сервиса:
   - загрузите файлы документов;
   - сканируйте данные сайта или нескольких сайтов.
3. Протестируйте работу сервиса и, если необходимо, выполните настройку сервиса.
4. Подключите сервис к навыку, боту или агенту.
5. Обновите, дополните или удалите документы из базы знаний.

Создание RAG-сервиса

Создание RAG-сервиса в системе

Чтобы создать RAG-сервис в системе:

1. Перейдите в раздел **AI-сервисы** и нажмите кнопку **Создать сервис** – откроется окно **Выберите тип сервиса**.



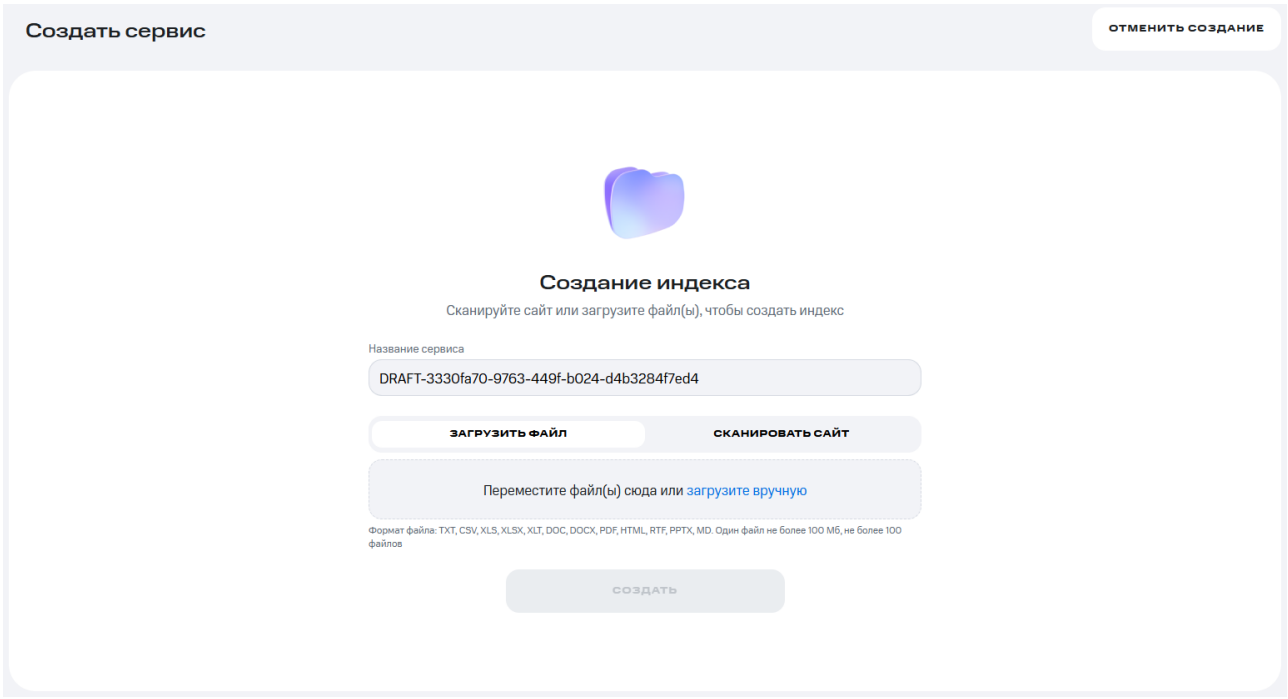
Выберите типа сервиса **RAG**. Если передумали создавать сервис – нажмите **Отменить** – откроется страница **Список сервисов**.

Если готовы продолжать, нажмите – **Продолжить**.

2. Откроется окно **Создать сервис**. Кнопка **Создать** неактивна.

В поле **Название сервиса** указан сгенерированный идентификатор, содержащий стартовый статус: **draft** (Черновик) и **uuid** (универсальный уникальный идентификатор: 36-символьный набор букв и цифр). Вы можете дать вашему сервису название сразу, а можете – в любой другой момент его жизненного цикла, даже когда сервис уже запущен и работает (подробнее см. Карточка RAG-сервиса).

> В названии сервиса должно быть от 4 до 50 символов. Кроме того, система не позволяет создавать сервисы с одинаковыми именами. В случае совпадения имён, система выдаст сообщение: «Это имя сервиса уже используется».



На этом этапе RAG-сервис уже создан. Запись о нем появилась в базе данных MWS AI Agents Platform. Теперь, если вам понадобится перейти в другой раздел платформы или выйти из системы, то по возвращении в раздел AI-сервисы, ваш RAG-сервис будет фигурировать в общем

списке в статусе **Черновик**. Обратите внимание, что название сервиса при этом вернется к первоначальному сгенерированному виду (**draf-uuid**).

Загрузка данных в базу знаний RAG-сервиса

Загрузка данных в базу знаний RAG-сервиса возможна двумя способами: загрузкой файлов или сканированием сайта с помощью краулера.

- **Загрузка файлами** подходит, если у вас есть набор готовых документов, которые имеют понятную структуру и требуют однократной или периодической загрузки из статичных источников.
- **Сканирование сайта** полезно, если исходные документы или информация находятся на веб-ресурсе, который регулярно обновляется или может содержать большое количество страниц. При этом сервис собирает информацию со страниц сайта автоматически и не требует предварительной загрузки файлов на сканируемый сайт.

Загрузка файлов

Чтобы загрузить файлы в базу знаний RAG сервиса:

1. На странице **Создание сервиса** перейдите на вкладку **Загрузить файлы**. Переместите нужные файлы в область загрузки или щелкните по области загрузки и загрузите данные классическим способом через проводник операционной системы.

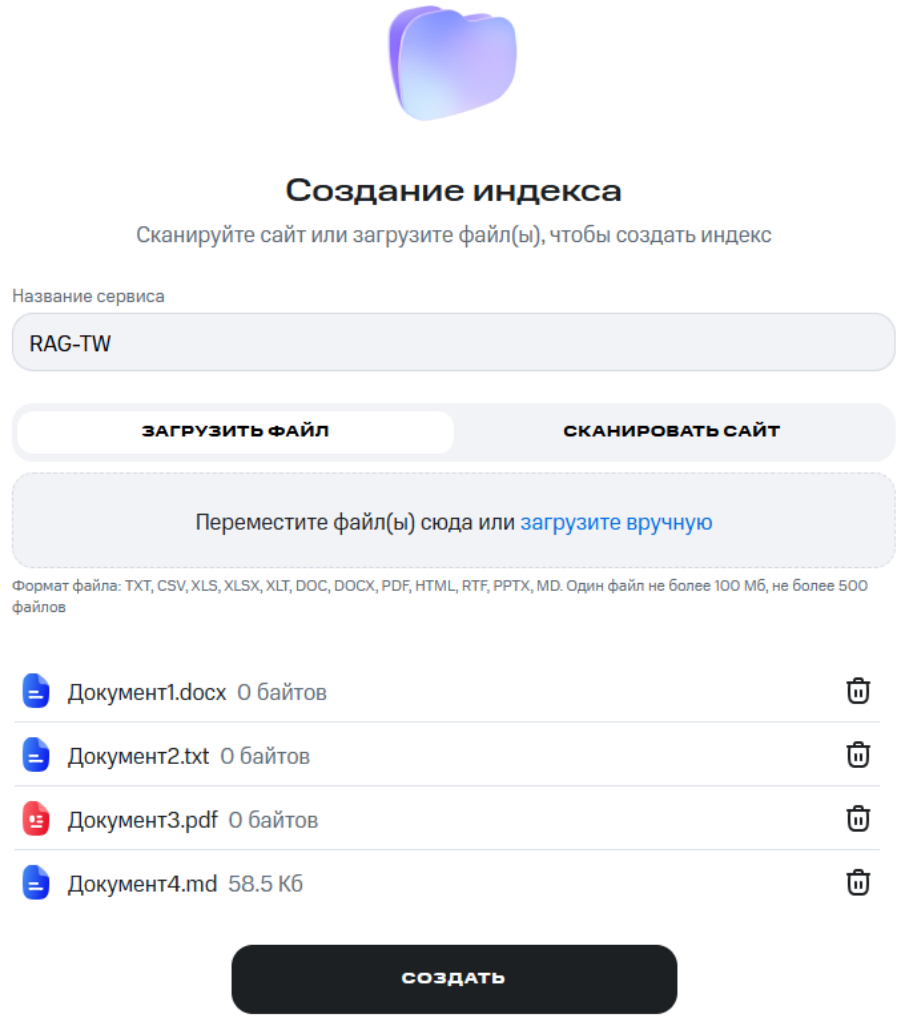
   Обратите внимание!

   - Формат файлов: TXT, CSV, XLS, XLSX, XLT, DOC, DOCX, PDF, HTML, RTF, PPTX, MD. Один файл не более 100 Мб, не более 100 файлов.
   - В процессе создания RAG-сервиса текст в файлах преобразуется в линейную структуру, поэтому не существует каких-либо требований к структуре загружаемых документов.
   - Изображения, встроенные в текст, не обрабатываются. Никакая текстовая информация из иллюстраций не может быть использована при формировании ответов пользователю RAG-сервиса.
   - Проверьте, что в файлах нет секретных сведений. Все загруженные материалы будут проиндексированы и доступны для использования. Ваши клиенты и сотрудники смогут получить к ним доступ, когда будут обращаться к базе знаний.

2. Начнется загрузка файлов.

   На экране вы увидите список загружаемых файлов и индикаторы их загрузки  , вы можете прервать загрузку, нажав на ✕. После того, как все файлы будут загружены, кнопка **Создать** станет активной.

   Если понадобится удалить файл после загрузки, нажмите  .



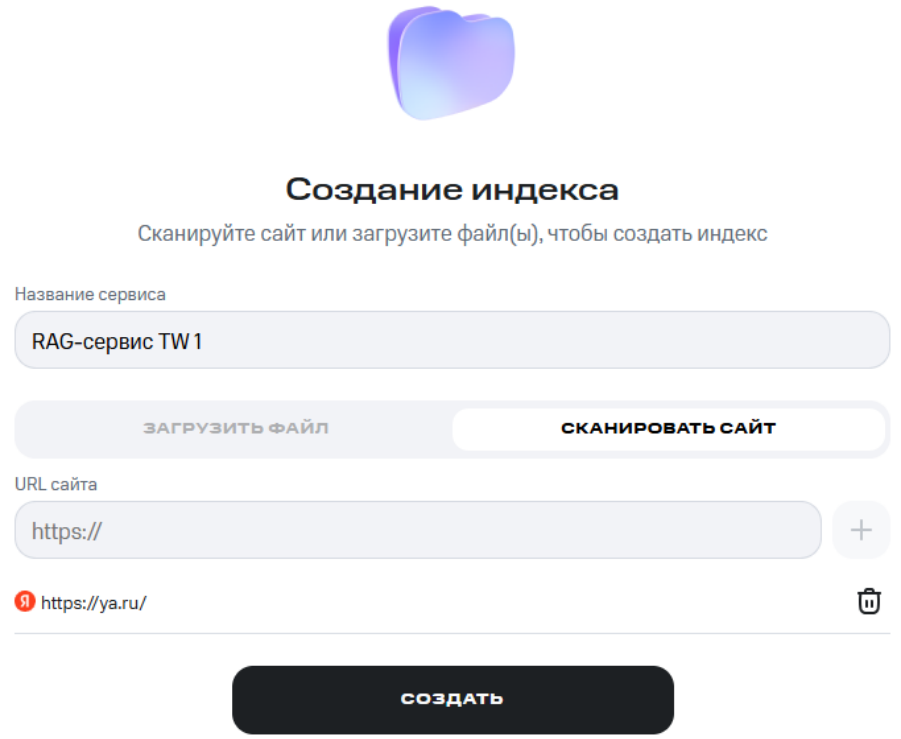
Если среди загружаемых документов случайно окажется файл неподдерживаемого формата, система укажет на него сообщением: **Неверный формат**. При этом кнопка **Создать** не блокируется. В процессе индексации документ-нарушитель игнорируется системой.
Если в ходе загрузки файла возникнут ошибки, система оповестит вас сообщением **Что-то пошло не так**. Выходом из подобных ситуаций может быть повторная загрузка документов, либо пересоздание сервиса. Если проблема не решается, следует обратиться в поддержку.

После того, как все файлы загружены, кнопка **Создать** станет активной. Вы можете переходить на этап создания индекса. При этом, если вам понадобится перейти в другой раздел платформы или выйти из системы, то при возвращении в раздел **AI-сервисы,** созданный вами RAG-сервис будет фигурировать в списке в статусе **Черновик**, и все загруженные в сервис файлы сохранятся. Обратите внимание, что название сервиса вернется к первоначальному сгенерированному виду (draf-uuid)

Сканирование сайта

Чтобы добавить в базу знаний данные с сайта:

1. На странице **Создание сервиса** перейдите на вкладку **Сканировать сайт**.



2. Введите URL сайта в соответствующее поле и нажмите **ENTER** или значок . Адрес сайта появится в списке. Можно внести несколько адресов.

   Платформа использует для сканирования рекурсивный краулинг — это процесс, при котором сервис-краулер скачивает страницу, извлекает из нее ссылки, скачивает страницы по этим ссылкам, снова извлекает ссылки и так далее, повторяя этот процесс до достижения заданной глубины. Значение глубины по умолчанию установлено на 10 уровней.

После того, как сайт добавлен, кнопка **Создать** станет активной. Вы можете переходить на этап создания индекса. При этом, если вам понадобится перейти в другой раздел платформы или выйти из системы, то при возвращении в раздел **AI-сервисы,** созданный вами RAG-сервис будет фигурировать в списке в статусе **Черновик** с первоначально сгенерированным наименованием (draf-uuid). Но обратите внимание, что внесенный сайт или список сайтов не сохранится. его придется внести снова.

Создание индекса

Чтобы создать индекс из загруженных в базу знаний файлов:

1. Нажмите кнопку **Создать** – на экране появится сообщение **Индекс создается**. В процессе индексации файлы базы знаний из текстового формата преобразуются в числовой, векторный формат и сохраняются в векторную базу данных. Процесс может занимать некоторое время (зависит от количества и размера файлов). Пока он продолжается, в **Списке сервисов** ваш сервис будет иметь статус **Создается**.

   - Если вы передумали создавать сервис, но процесс уже запущен – есть кнопка **Отменить создание**. Нажмите на нее, и сервис удалится. Но прежде, чем это произойдет, система попросит у вас подтвердить ваше намерение, **потому что удаление необратимо**.
   - Если вы настаиваете на удалении, нажмите кнопку **Удалить** – откроется окно **Список сервисов.**
   - Статус вашего сервиса изменится на **Удаляется**. Спустя некоторое время он безвозвратно исчезнет из списка вместе с базой знаний.

Если при создание индекса возникает ошибка, то на экране появляется такое сообщение – **Ошибка при создании индекса**.
Если вы уже вернулись в **Список сервисов**, то вы увидите, что создаваемый вами сервис имеет статус **Ошибка**. Открыв его карточку, вы опять же встретитесь с приведенным выше сообщением об ошибке.
Что можно сделать для исправления ситуации:

- В окне ошибки нажать кнопку **Попробовать снова**.
- Проверить загруженные документы. И если вы обнаружили, что файлы были некачественными (например пустыми), нажать кнопку **Изменить файлы** и загрузить новые.
- Если ситуация повторится и с новыми качественными файлами – обращайтесь по указанным в сообщении об ошибке контактам.

2. Когда индекс будет готов, его статус в списке изменится на **Запущен**.
3. По окончании индексации откроется Карточка сервиса на вкладке База знаний со списком документов в статусе **Индексирован**. Документы, загруженные с сайта в качестве префикса, будут иметь страницу, с которой они были скачаны в базу знаний RAG-сервиса (например, asciidoctor.org_docs_documentation.html).

База знаний проиндексирована, RAG-сервис запущен и готов к использованию.

Карточка RAG-сервиса

Карточка RAG-сервиса содержит детальную информацию и служит своего рода "пультом управления" базой знаний RAG-сервиса и его настройками.

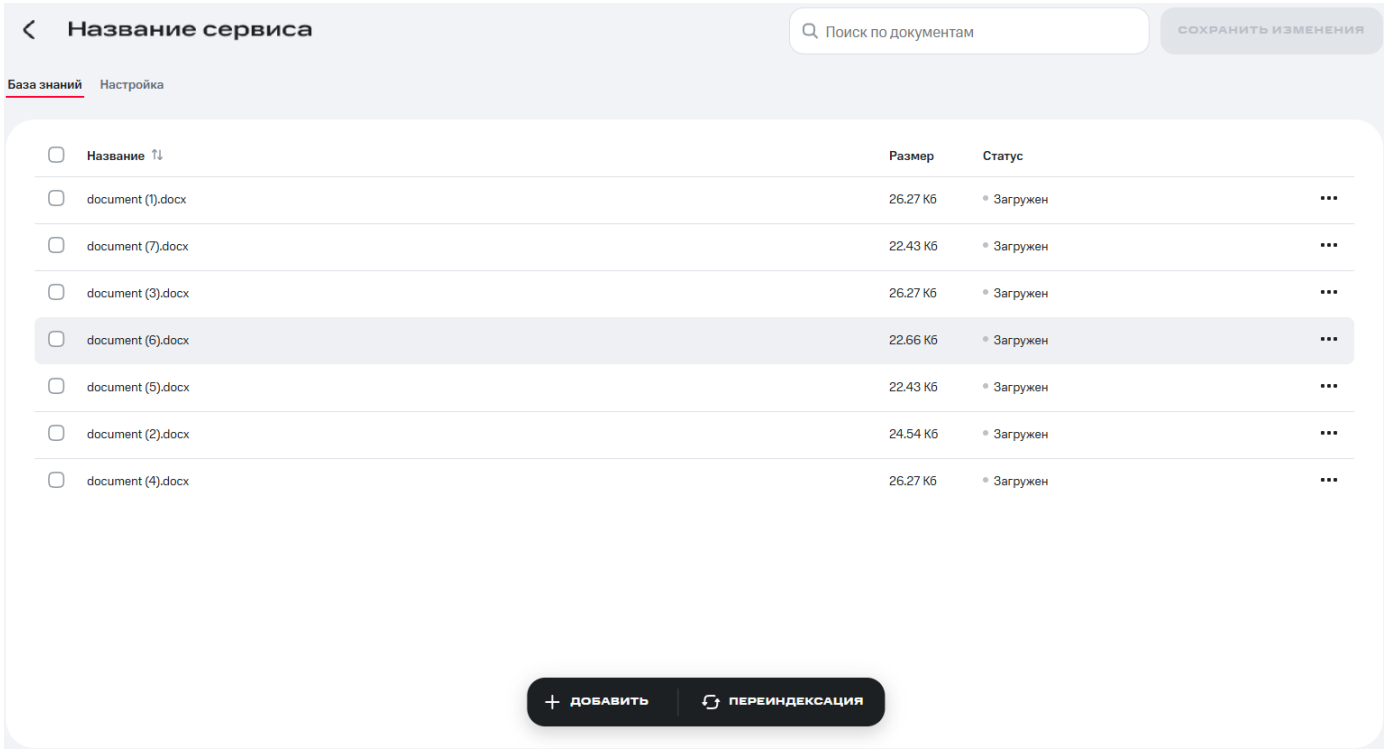
Вкладка База знаний

Во вкладке размещается список документов, загруженных в базу знаний RAG-сервиса. Список можно фильтровать по наименованию документа. Доступен автоматический поиск документов по базе.

Во вкладке вы можете:

- Отфильтровать список документов по названию.
- Выполнить поиск документа по списку.
- Получить информацию о размерах и статусах документов.
- Добавить, исключить или полностью удалить один, несколько или все документы из базы знаний.
- Выполнить переиндексацию базы знаний по результатам изменений списка документов.
- Скачать документы.

Управление Базой знаний описано в разделе **Работа с базой знаний RAG**.



Вкладка Настройка

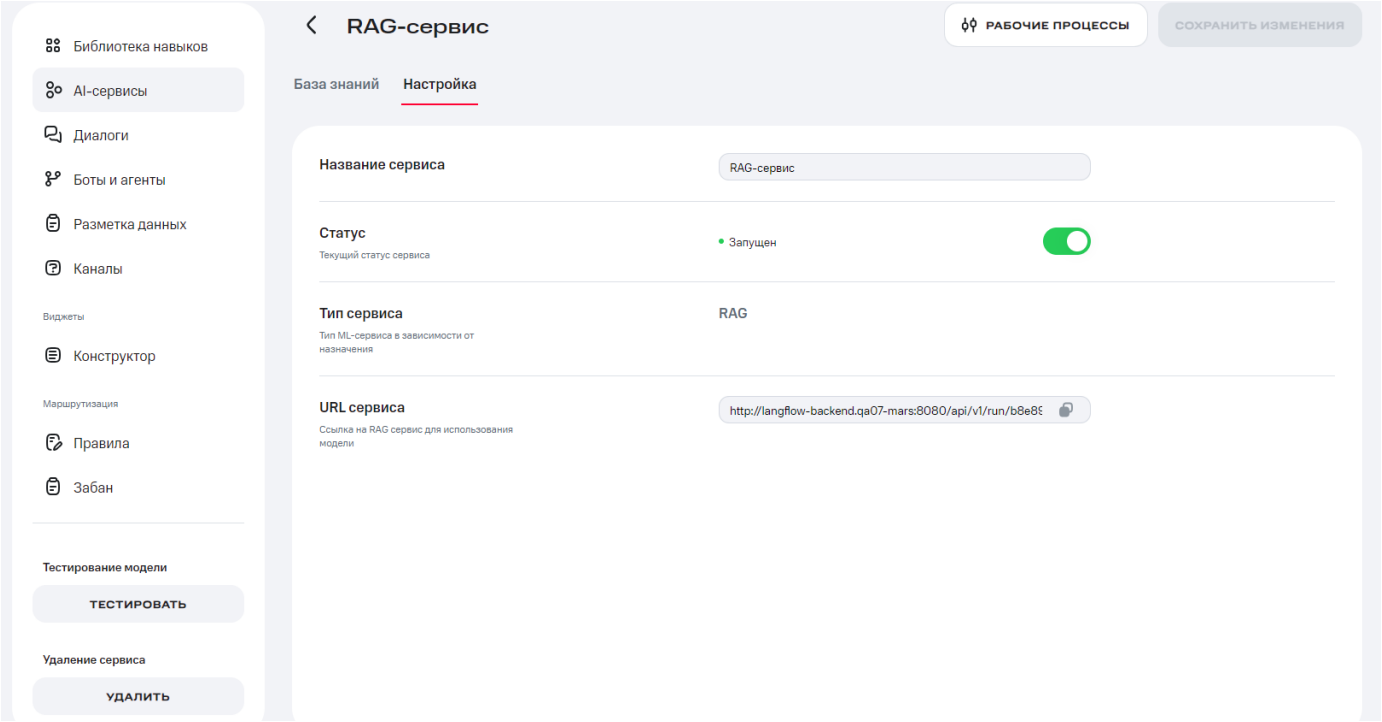
Вкладка содержит следующую информацию, кнопки и переключатели:

- **Название сервиса** – при необходимости редактируется;
- **Рабочие процессы** – кнопка перехода к интерфейсу настройки процессов langflow RAG-сервиса (подробнее см. Рабочие процессы langflow);
- **Сохранить изменения** – кнопка становится активной, если настройки сервиса изменены;
- **Статус** – текущий статус сервиса и переключатель, запускающий или останавливающий RAG-сервис;
- **Тип сервиса** – в случае с RAG тип всегда **RAG**;
- **URL сервиса** – конечная точка процесса Langflow, который обрабатывает логику RAG-сервиса (подробнее см. Подключение сервиса RAG);

Кроме того, при переходе во вкладку **Настройки** в нижней части основной панели инструментов отображаются кнопки:

- **Тестировать** – кнопка открывает виджет тестирования сервиса (подробнее см. Тестирование RAG-сервиса);
- **Удалить** – сервис и его база знаний полностью удаляются из системы (подробнее см. Удаление RAG-сервиса).

Все изменения во вкладке **Настройка** сохраняются только при нажатии кнопки **Сохранить изменения**. Кнопка активируется только после внесения изменений.

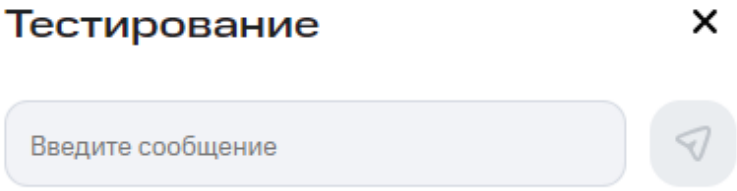


Тестирование RAG-сервиса

Тестирование RAG-сервиса необходимо для проверки качества ответов модели и достаточности базы знаний RAG.

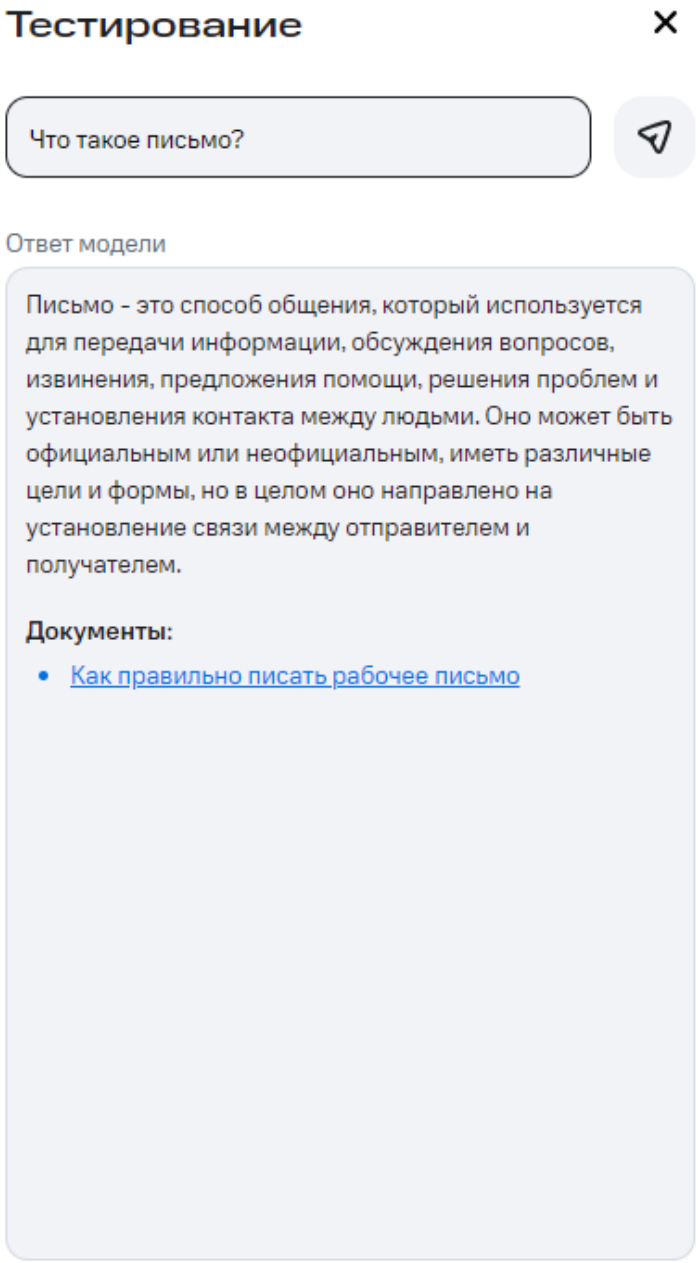
Чтобы протестировать RAG-сервис:

1. В списке найдите сервис, тестирование которого нужно провести. Войдите в Карточку сервиса и перейдите на вкладку **Настройка**.
2. Убедитесь, что сервис запущен. Если нет – запустите сервис переключателем , затем нажмите кнопку **Сохранить изменения**.
3. Теперь нажмите кнопку **Тестировать**. На экране появится виджет **Тестирование**.



4. Введите в поле ваш вопрос к модели и нажмите  .

5. В поле **Ответ модели** появится ответ и ссылка на документ базы данных.



6. Вы можете задать столько вопросов, сколько необходимо для проверки работы сервиса. Изучите ответы модели и примите решение о качестве работы RAG-сервиса.
7. Чтобы завершить тест, закройте виджет **Тестирование** или выйдите из **Карточки сервиса**.

Тестирование RAG-сервиса выполнено.

По результатам тестирования можно принять решение о добавлении либо исключении документов в базу знаний или об обновлении базы знаний. Кроме того, тест может указать на необходимость внести изменения в рабочие процессы RAG-сервиса (подробнее см. в разделе **Рабочие процессы langflow**).

Работа с базой знаний RAG

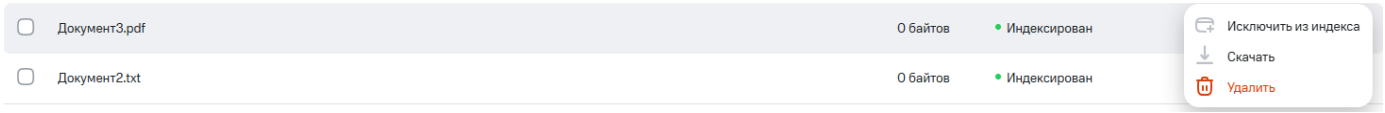
Для того чтобы RAG-сервис предоставлял всегда актуальные и полезные ответы, необходимо "следить за свежестью" его базы знаний: удалять или обновлять устаревшие версии документов, загружать новые.

Изменения в базе знаний учитываются в индексе RAG только после проведения переиндексации. До переиндексации RAG-сервис работает со старой версией базы.

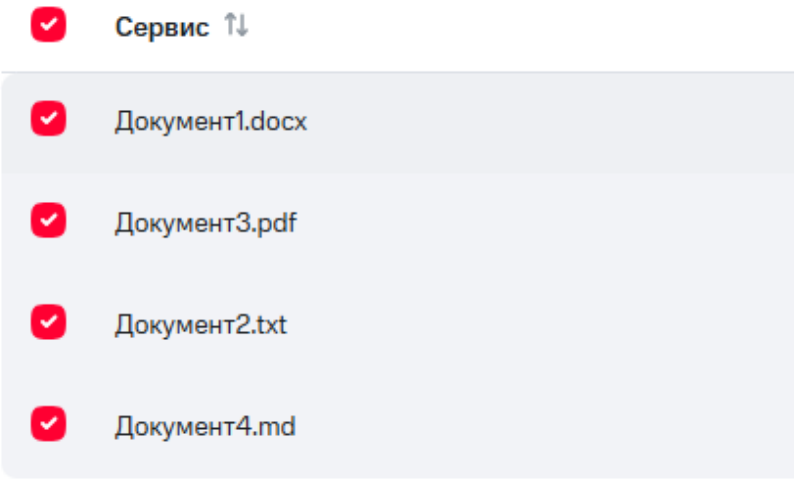
Интерфейс вкладки **База знаний** организован в виде списка, где вам доступны следующие операции:

- Автоматический поиск документов по названию;
- Сортировка документов по названию в алфавитном порядке;
- Добавление новых документов в базу знаний;
- Скачивание документов из базы знаний;
- Обновление существующих документов в базе знаний;
- Исключение документов из индекса;
- Удаление документов из базы знаний;
- Переиндексация базы знаний.

Вы можете работать с отдельным документом. Для этого нажмите на значок ⋯ в конце строки списка и выберите действие из выпадающего меню.



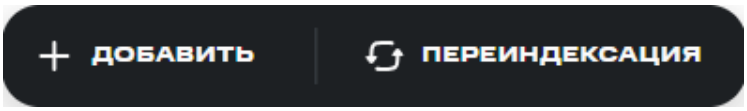
Вы можете работать с несколькими документами. Отметьте нужные файлы или выберите сразу все.



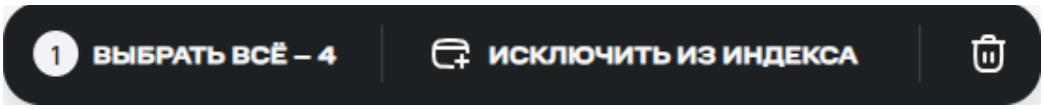
В нижней части экрана размещен тулбар с кнопками массовых операций. Нажмите нужную кнопку тулбара, чтобы выполнить операцию с выбранными файлами.

Тулбар меняет свою функциональность в зависимости от количества выбранных документов и их статусов.

Если ни один документ не выбран, на тулбаре будут только две кнопки **Добавить** и **Переиндексация**.



Если выбрано несколько документов, то в зависимости от их количества и статусов тулбар меняет количество и назначение кнопок: **Выбрать все/снять выделение**, **Исключить из индекса**, **Удалить файлы**.

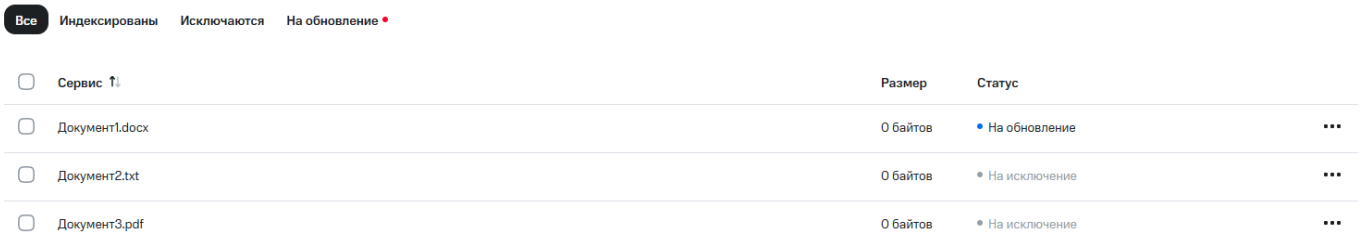


Статусы документов

Документы базы знаний RAG-сервиса имеют определенный жизненный цикл, этапы которого отражены в следующих статусах.

Статус	Описание
Загружен	Документ успешно загружен в базу знаний, но еще не добавлен в RAG-индекс. До переиндексации базы такой документ не используется RAG-сервисом
Индексирован	Документ успешно загружен в базу знаний и добавлен в RAG-индекс. Используется при генерации ответа RAG-сервисом
На исключение	Документ отмечен на исключение из RAG-индекса и будет исключен после переиндексации базы знаний. Продолжает использоваться при генерации ответа сервисом RAG
Исключен	Документ исключен из RAG-индекса после переиндексации. Не используется при генерации ответа RAG-сервисом
На обновление	В базу знаний загружена новая версия существующего документа. До переиндексации базы, RAG-сервис использует старую версия документа

На вкладке **База знаний** при наличии более одного элемента списка выполняется группировка документов по статусам. Для переключения между группами кликните наименование группы в шапке списка.



Индексация и переиндексация базы знаний

Первоначальная индексация базы знаний происходит на этапе создания RAG-сервиса (подробнее см. Создание сервиса RAG), поэтому все документы в базе "свежеиспеченного" сервиса всегда будут в стартовом статусе **Индексирован**. Если после тестирования сервиса пользователя все устраивает, то ничего с ним делать и не нужно. База знаний существует и работает.

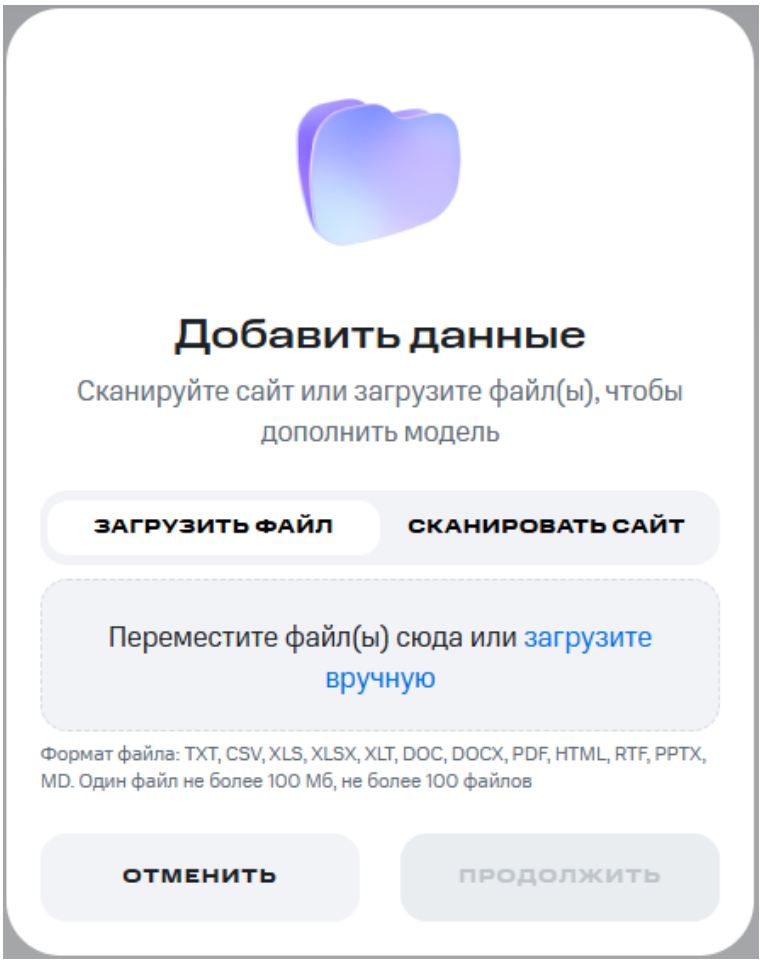
Но прелесть технологии RAG состоит именно в возможности оперативно и незаметно для клиента управлять базой. Вы вносите изменения: исключаете файлы из индекса, обновляете или удаляете, а, чтобы база знаний учла эти изменения, нажимаете кнопку **Переиндексация**. Пока переиндексация продолжается, RAG-сервис использует старые версии документов, чтобы работа агента или навыка, который использует RAG, не прерывалась. По окончании процесса обновленная база знаний включается в работу RAG-сервиса.

Внимание!
Процесс переиндексации затрагивает **ВСЮ** базу знаний. Нельзя сначала внести несколько изменений, а затем проиндексировать только некоторые из них. В переиндексацию попадут все загруженные, обновленные, исключаемые и удаленные файлы.
Процесс может занять много времени при большом объеме изменений. Рекомендуется проводить переиндексацию после внесения всех изменений.
Во время переиндексации список документов базы знаний блокируется для любых действий пользователя.

Добавление новых документов в базу знаний

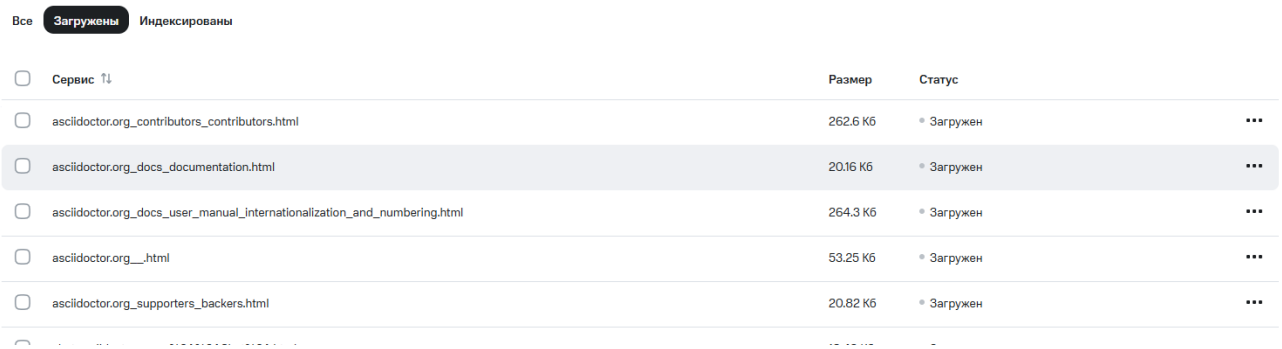
Чтобы добавить документы в базу знаний существующего RAG-сервиса:

1. Перейдите в **Карточку сервиса** и откройте вкладку **База знаний**. В нижней части вкладки нажмите кнопку **Добавить**.
2. В открывшейся форме **Добавить данные** выберите тип загрузки: **Загрузить файл** или **Сканировать сайт**.



3. Если выбрана загрузка файлов – переместите нужные файлы в область загрузки или щелкните по области загрузки и загрузите данные классическим способом через проводник операционной системы. Документы появятся в списке загрузок (см. Загрузка файлов).
4. Если выбрано сканирование сайта – на вкладке **Сканировать сайт** введите адрес в поле **URL** и нажмите **ENTER** или значок +. Сайт появится в списке сканирования (см. Сканирование сайта), затем установите глубину сканирования.
5. После того, как список документов или сайтов будет готов, нажмите **Продолжить**. Вы вновь окажетесь на вкладке **База знаний**. Список документов пополнился новыми файлами в статусе **Загружен**. В шапке появился соответствующий раздел **Загружены**.

   Если было выбрано сканирование сайтов, то в шапке списка вы увидите сообщение: **Идёт сканирование данных с сайта**. Список документов блокируется на время сканирования. После окончания процесса в списке появятся новые документы с сайта.



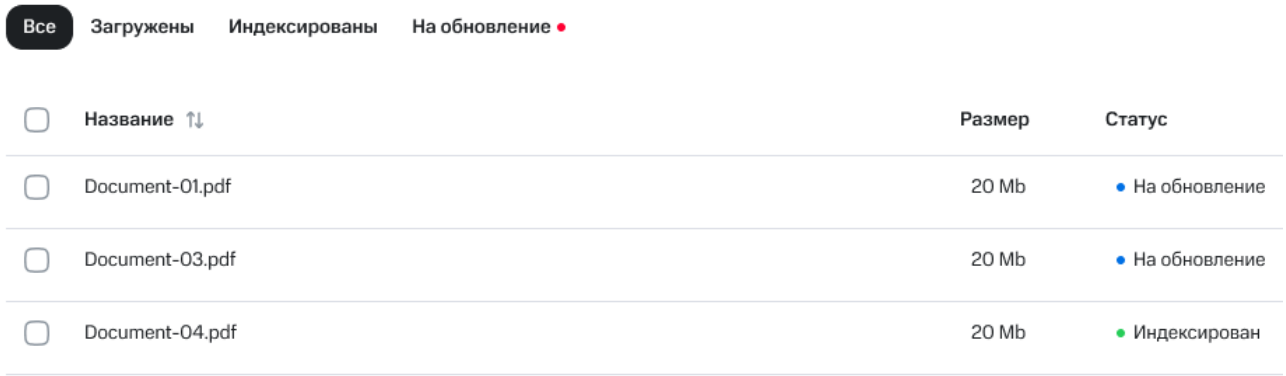
Документы добавлены в список в статусе **Загружен** и начнут использоваться в запросах к RAG-сервису после переиндексации базы знаний.

Обновление документов базы знаний

Если потребовалось изменить содержание загруженных или проиндексированных документов, не меняя их имени и формата, выполните следующие шаги:

1. Во вкладке **База знаний** нажмите кнопку **Добавить**.
2. В открывшейся форме **Добавить данные** загрузите новые версии обновляемых документов. Либо пересканируйте страницу сайта с обновленной информацией. Загруженные документы появятся в списке готовых к добавлению в базу знаний.

   Если вы загружаете файлы, то убедитесь, что имя загружаемого файла идентично обновляемому документу базы знаний.
   Вы также можете скачать документ из базы знаний, отредактировать его и с тем же именем загрузить в базу. Чтобы скачать документа нажмите значок ••• в его строке, а затем из выпадающего меню выберите **Скачать**.
3. После того, как список документов будет готов, нажмите **Продолжить**. Загруженные документы в списке получили статус **На обновление**. Одноименный раздел появится в шапке списка.



После переиндексации документы будут обновлены и получат статус **Индексирован**.

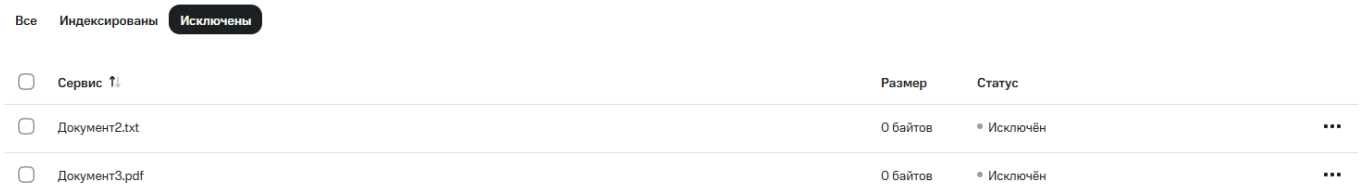
Исключение документов из базы знаний

Документы можно временно исключить из базы знаний. Такое действие может пригодиться, если возникло подозрение, что документ потерял актуальность. Например, если перечню филиалов организации понадобилась актуализация. После обновления перечень можно будет вернуть в базу знаний.

Чтобы пометить файл на исключение.

1. В строке документа, который необходимо исключить нажмите значок ••• в строке документа, а затем выберите в выпадающем меню команду **Исключить из индекса**. Либо отметьте несколько документов и нажмите аналогичную кнопку тулбара.
2. Документы переведен в статус **На исключение**. В шапке списка появится раздел Исключаются.

После переиндексации базы знаний документы получат статус **Исключен** и больше не будут использоваться сервисом RAG. В шапке списка появится раздел **Исключены**.



Подсказка:
Чтобы вернуть исключенный документ в индекс, выберите один или несколько документов, и нажмите кнопку **Индексировать** в выпадающем меню списка или на тулбаре. Документы перейдут в статус **Загружен**. И после переиндексации вернутся в индекс со статусом **Индексирован.**

Удаление документов из базы знаний

Удалить можно документ в любом статусе. Для этого выполните следующие шаги.

1. Чтобы удалить документ, выберите один или несколько файлов на вкладке и нажмите кнопку , в выпадающем меню для одного файла, либо в тулбаре, если удаляете несколько.
2. Подтвердите удаление

   Внимание!
   Удаление документа необратимо.
   RAG-сервис прекратит использовать удаленный документ после переиндексации базы знаний.

Документы удалены из списка файлов во вкладке и будут удалены из индекса RAG после переиндексации базы знаний.

Рабочие процессы RAG-сервисов в Langflow

Помимо базовых настроек и управления Базой знаний, RAG-сервис предлагает инструментарий тонкой настройки самого процесса RAG. Этот инструментарий базируется на фреймворке Langflow – no-code конструкторе AI-агентов и приложений.

Подробнее см в документации системы langflow https://docs.langflow.org/).

Рабочие процессы RAG – это сценарии основных флоу RAG, подготовленные с помощью фреймворка Langflow разработчиками MWS AI Agents Platform.

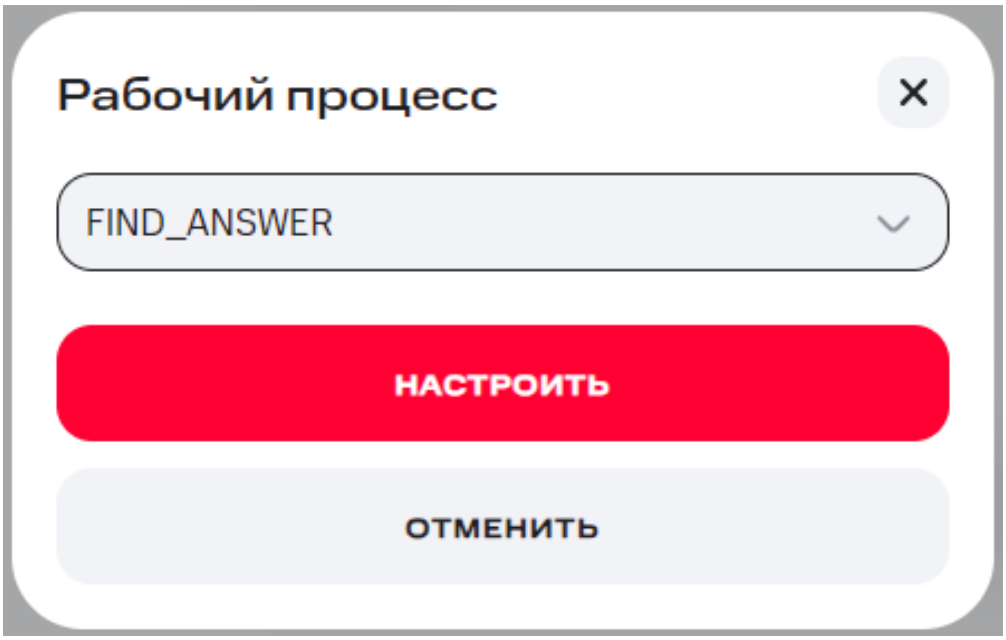
У каждого RAG-сервиса три рабочих процесса.

- Процесс **CREATE_INDEX** управляет созданием индекса RAG-сервиса.
- Процесс **DELETE_INDEX** отвечает за удаление индекса RAG-сервиса.
- Процесс **FIND_ANSWER** обеспечивает основную работу RAG-сервиса – поиск и возвращение ответа на вопрос пользователя.

Рабочие процессы привязаны к конкретному сервису RAG. Настройка процессов этого сервиса влияет только на его работу. Удалить процесс из системы, отвязать процесс от сервиса или привязать процесс одного сервиса к другому невозможно.

Чтобы перейти в интерфейс платформы Langflow и внести изменения в процессы:

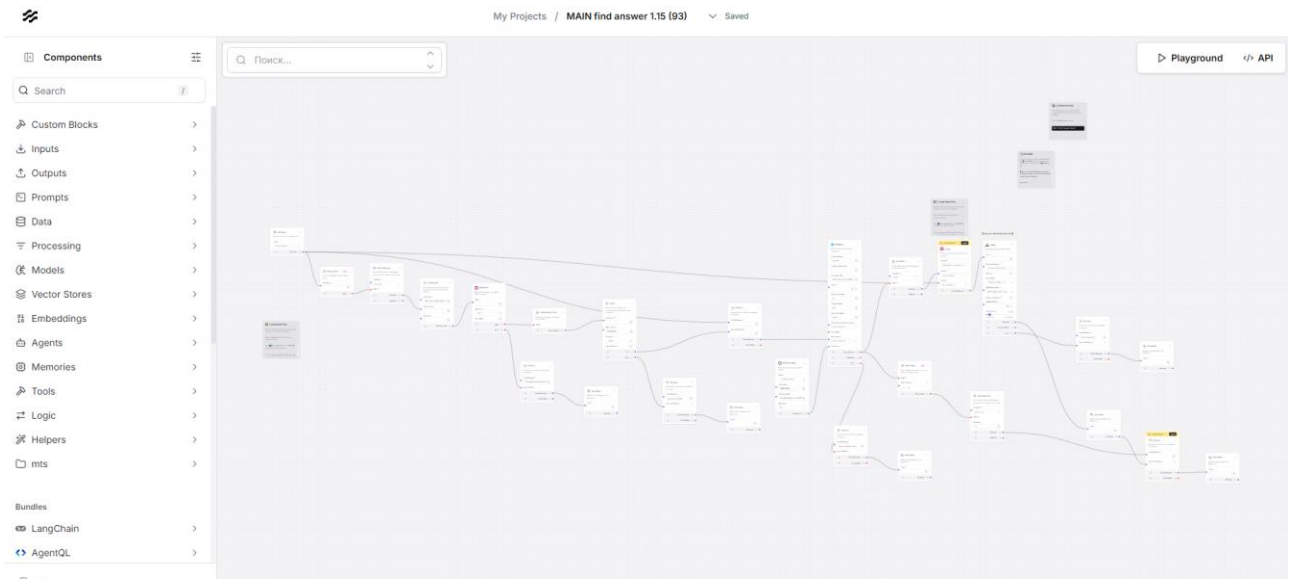
1. Откройте карточку RAG-сервиса на вкладке **Настройка** и нажмите кнопку **Рабочие процессы** – откроется окно выбора процесса.



2. В выпадающем списке выберите один из процессов и нажмите кнопку **Настроить** – откроется интерфейс Langflow, представляющий из себя конструктор сценариев. В рабочем поле конструктора отображен сценарий, по которому движется выбранный на предыдущем шаге процесс.

Интерфейс платформы Langflow позволяет редактировать рабочий процесс: изменить векторную базу данных, которую использует RAG или отредактировать системный и пользовательский промпты и так далее.

Перемещение по полю конструктора выполняется с зажатой левой кнопкой мыши, увеличить рабочее поле можно колесиком мыши. Чтобы перенести элементы с панели компонентов, воспользуйтесь механизмом drag-and-drop.
Также на рабочей области доступен поиск по компонентам сценария. Введите название или часть названия компонента в поле поиск и нажмите значок 🔍 . Количество элементов с совпадениями в названиях появится в поле поиска. Чтобы перемещаться между найденными компонентами используйте кнопки 5/16 .



3. В рабочих процессах RAG используются стандартные компоненты сценариев Langflow. Компоненты рассортированы в левой части интерфейса по смысловым группам и бандлам компаний разработчиков. В папке mts собраны компоненты, использующиеся в базовых сценариях МТС.

Вот основные компоненты, которые могут быть полезны для настройки RAG-сервисов.

- Чтобы заменить модель, выполняющую эмбеддинг, найдите подходящую в разделе **Embeddings**.
- Чтобы сервис использовал другую векторную базу данных, обратитесь в раздел **Vector stores**, выберите требуемый вариант.
- Чтобы изменить пользовательский промпт процесса, найдите на рабочем поле компонент **Prompt** и отредактируйте текст промпта в поле **Template**.
- Чтобы изменить системный промпт процесса, найдите на рабочем поле компонент **Model**, отвечающий за подключение LLM, (в базовом процессе модель развернута в блоке NVIDIA) и отредактируйте текст промпта в поле **System Message**.

> Подробная документация платформы Langflow находится здесь https://docs.langflow.org

4. Чтобы проверить изменения в процессе, нажмите кнопку ▷ Playground .
5. Чтобы сохранить новую версию процесса, нажмите **Завершить работу** и подтвердите сохранение – откроется список **AI-сервисы**.
6. Для выхода без сохранения используйте кнопку браузера **Назад** – вы вернетесь в карточку сервиса на вкладку **Настройка.**

Работа с платформой Langflow завершена. RAG-сервис будет использовать измененный процесс.

Подключение бота к RAG-сервису

Чтобы подключить RAG-сервис к боту:

1. Откройте карточку созданного сервиса RAG и из поля **URL** сервиса скопируйте его адрес.
2. В разделе **Боты и агенты** перейдите на вкладку **Боты** и откройте конструктор нужного бота.
3. Перейдите в сценарий, в котором предполагается использовать поиск ответа по документации. Вставьте скопированный **URL** в интеграционный блок HTTP-вызова.

> Подробнее о работе с конструктором ботов см. Разработка сценариев, а также в документации системы langflow https://docs.langflow.org/).

4. Сохраните изменения кнопкой  .

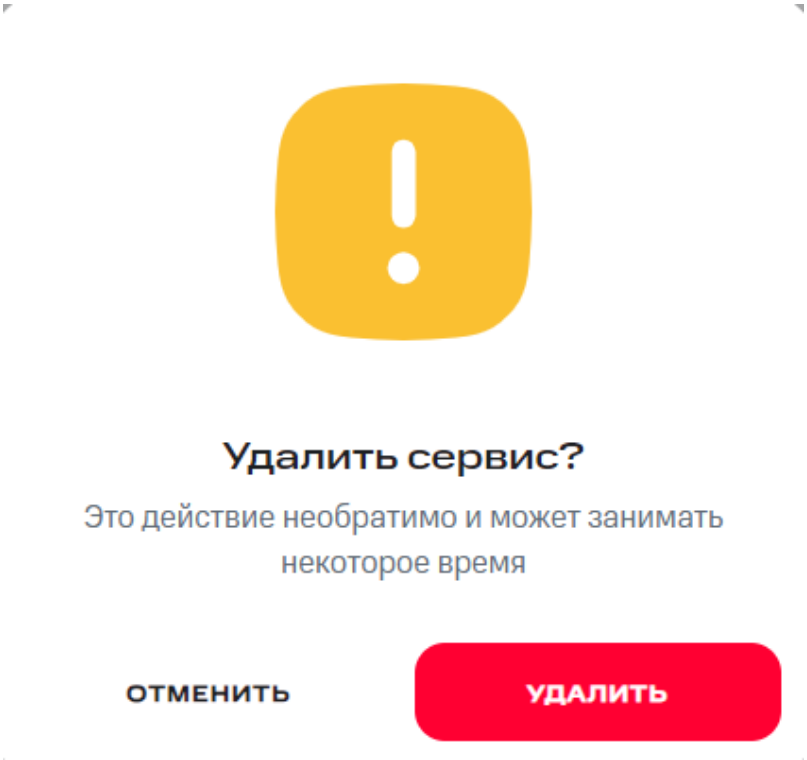
При обработке запроса бот будет искать ответ в Базе знаний RAG сервиса.

Удаление RAG-сервиса

Чтобы удалить RAG-сервис:

1. Откройте **Карточку сервиса** перейдите на вкладку **Настройка** и нажмите кнопку **Удалить.**
2. Подтвердите удаление – начнется процесс, входе которого остановится будет остановлена и удалена модель сервиса, все файлы базы знаний в хранилище S3, RAG-индекс и записи о сервисе в базе данных платформы.

   Внимание!
   Удаление сервиса необратимо.



3. Карточка сервиса закроется. Вы переместитесь на страницу AI-сервисы. Сервис получит статус **Удаляется**. Снова открыть его карточку не получится. Через некоторое время сервис исчезнет из списка сервисов.

RAG-сервис удален из системы.

РАЗМЕТКА ДАННЫХ

В разделе **Разметка данных** вы можете создавать обучающие датасеты для ML-моделей AutoML-сервисов.

В процессе создания датасета для обучения модели классификатора каждому текстовому фрагменту присваивается метка, определяющая класс, к которому относится фрагмент текста. Например, фразам "запиши меня к терапевту" и "как попасть к терапевту" можно присвоить класс – therapist.
Разметка данных для обучения модели NER – это выделение сущностей во фразе и присвоение каждой выделенной сущности собственной метки. Так во фразе "записаться к терапевту в Москве", можно определить сущность "в Москве" и назначить ей метку "city", а за сущностью "терапевт" закрепить метку therapist.

На главной странице раздела отображаются плитки проектов разметки и лаконичный инструментарий – кнопка **Создать проект**.

Каждая плитка проекта содержит краткую информацию:

- тип модели, которую можно обучить на размеченных данных;
- название проекта;
- количество данных к разметке и количество уже размеченных данных;
- дату и время последнего изменения проекта.

Кнопка ... открывает контекстное меню управления проектом, с помощью которого можно:

- редактировать проект;
- дублировать проект;
- скачать датасет проекта;
- удалить проект.

Чтобы открыть какой-либо проект для разметки или настройки, нажмите на его плитку – отобразится карточка проекта на вкладке **Данные разметки**.

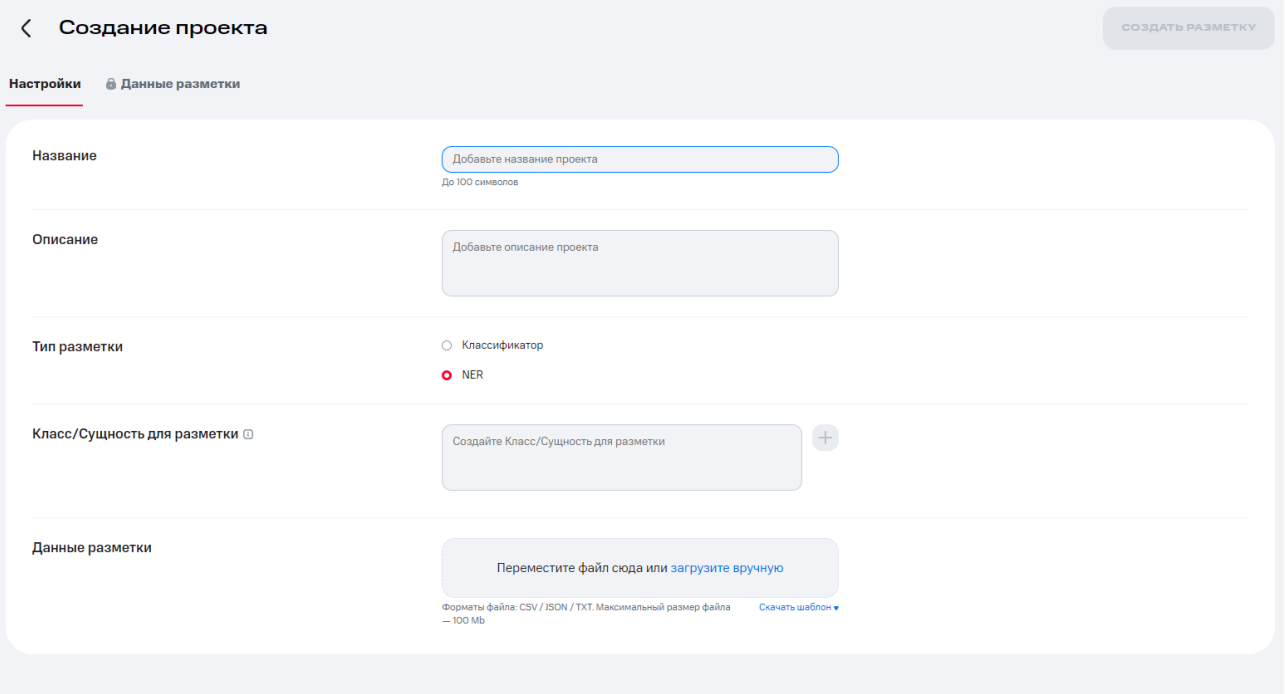
Обычный сценарий создания обучающего датасета в разделе **Разметка данных** таков:

1. В разделе **Разметка данных** создайте новый проект разметки.
2. Добавьте данные для разметки.
3. Разметьте данные.
4. Скачайте датасет и используйте его в обучении AI-сервисов.

Создание проекта разметки данных

Чтобы создать проект разметки данных:

1. На странице **Разметка данных** нажмите кнопку **Создать проект** – откроется окно **Создание проекта** на вкладке **Настройка**. Вкладка **Данные разметки** на этапе создания заблокирована, т.к. данных еще нет.



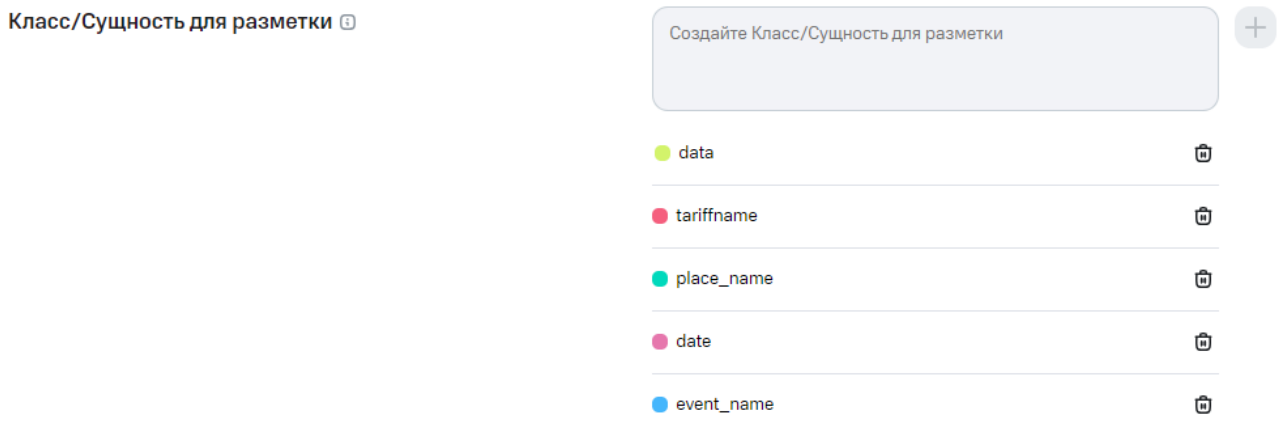
2. Введите название проекта. До 100 символов. **Это обязательный шаг.**
3. Добавьте описание. **Это не обязательный шаг.**
4. Выберите нужный **Тип разметки**.
5. Подготовьте набор меток (лейблов) классов/сущностей, которые будут присваиваться фрагментам текста в процессе разметки данных.

Для этого в поле **Класс/Сущность для разметки** через запятую (или с новой строки) введите названия меток. Добавляйте столько меток, сколько потребуется для работы с данными: количество не ограничено.

Затем нажмите кнопку  . Список меток появится на странице.

Чтобы отредактировать метку, кликните на ней и исправьте текст.

Удалить метку можно кнопкой  .



Этот шаг необязателен. Вы можете добавлять или удалять метки классов/сущностей в процессе разметки данных.

6. Вы можете добавить в проект заранее подготовленные по шаблонам данные разметки. Для этого перетащите файл разметки в поле **Данные разметки** или загрузите его через проводник операционной системы.

Форматы файла: CSV / JSON / TXT. Максимальный размер файла – 100 Мб.

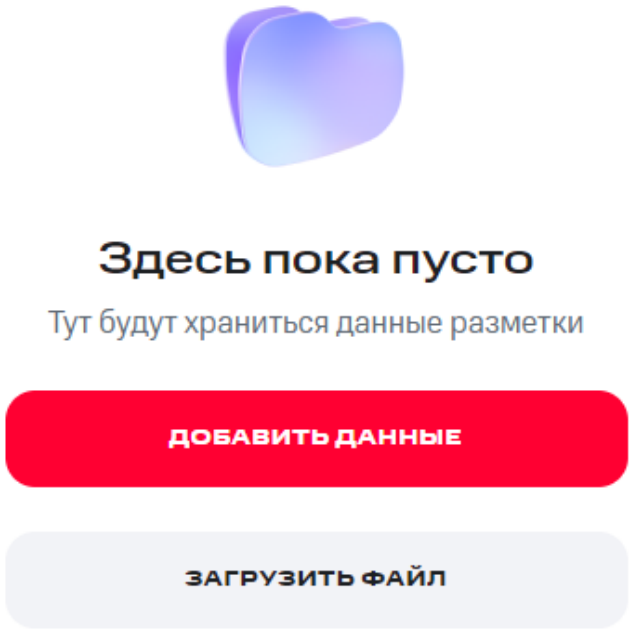
Чтобы получит шаблоны, нажмите **Скачать шаблон**, и выберите нужный из выпадающего списка.

> Этот шаг необязателен. Вы сможете добавить, отредактировать или удалить данные в процессе разметки данных. Подробнее об этом см. Добавление данных в проект.

7. Таким образом для создания проекта, вам достаточно ввести название и выбрать тип разметки. После этих действий кнопка **Создать разметку** становится активной.
8. Нажмите кнопку **Создать разметку** – откроется вкладка **Данные разметки**.

Если на этапе создания проекта вы не добавляли в него данных, то на вкладке **Данные разметки** вы увидите сообщение **Здесь пока пусто.**

Вы можете начать добавление данных сразу или вернуться к проекту позже – если выйти в основной раздел **Разметка данных**, вы увидите проект в списке.



Проект разметки создан и появился в списке.

Добавление данных в проект

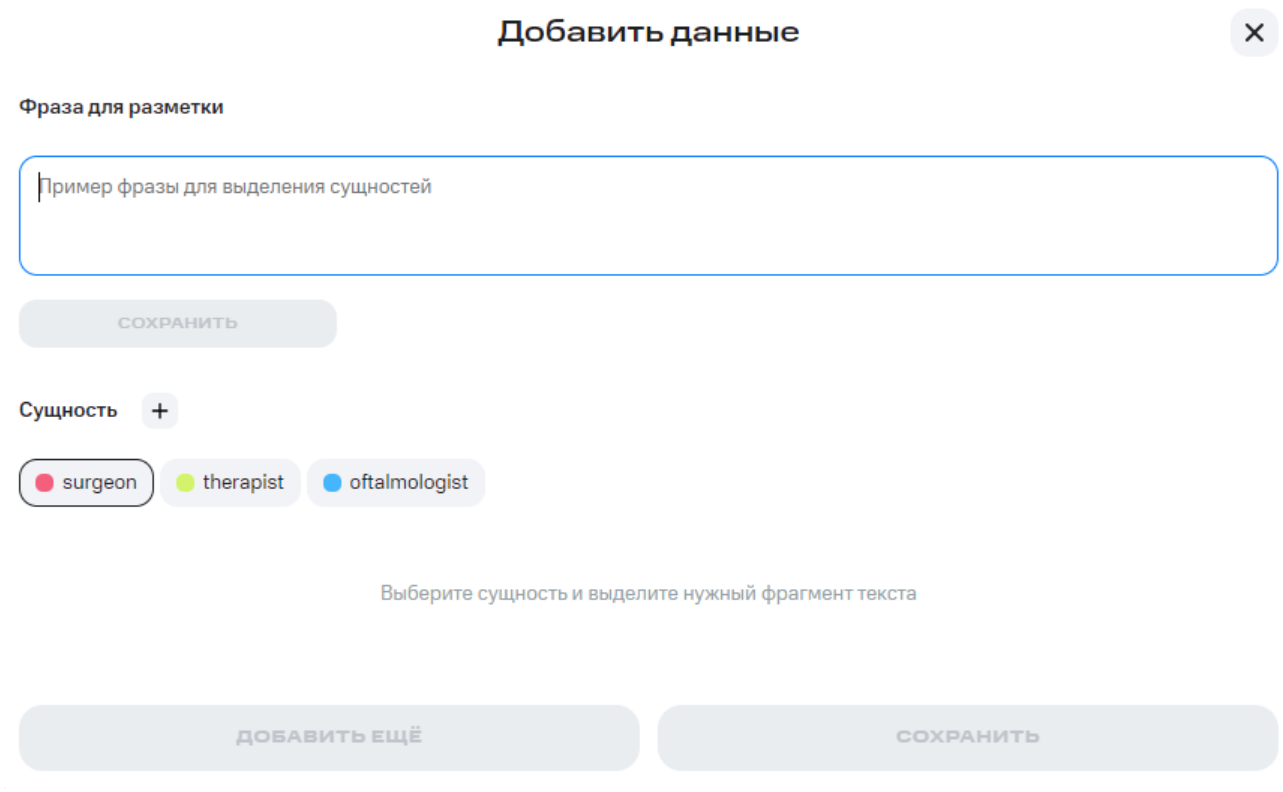
После создания проекта (см. Создание проекта разметки данных) вы можете выбрать способ добавления данных в проект: вручную либо загрузкой файла с датасетом.

> Оба способа имеют свои преимущества:
>
> - ручной способ позволяет редактировать и размечать загружаемые фразы в процессе добавления;
> - с помощью файла датасета можно загрузить значительные объемы данных.

Добавление данных вручную

Чтобы добавить данные:

1. Во вкладке **Данные проекта** нажмите кнопку **Добавить данные** – откроется одноименная форма.



Если на этапе создания вы добавили в проект метки сущностей/классов, то в форме будет отображен их список. Если нет – форма будет пустой.

Чтобы добавить метку, нажмите + введите в появившемся поле название и снова нажмите кнопку + . Новая метка появится в списке. Количество меток не ограничено.

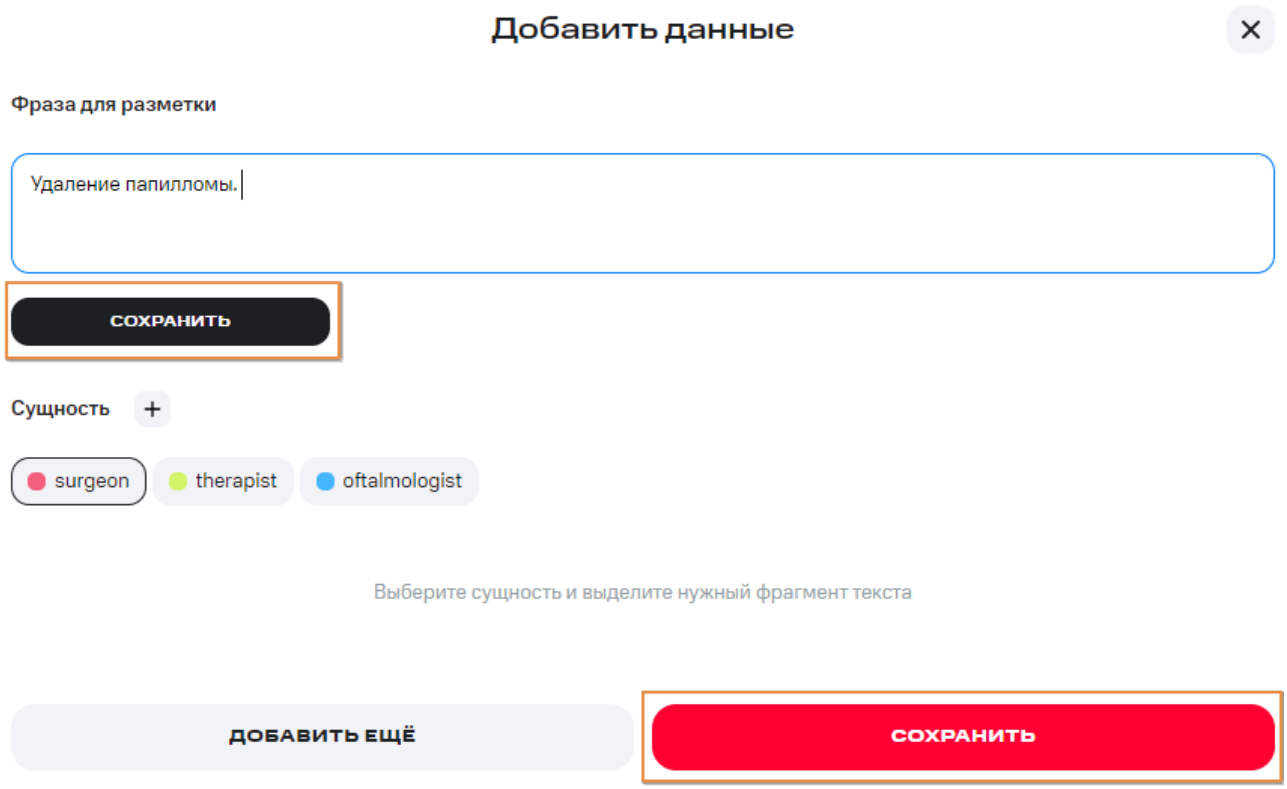
Система не допускает создания двух одинаковых меток и сообщит вам, если название метки уже используется.

Созданные метки можно редактировать и удалять на вкладке **Настройки**. Подробнее см. в разделе **Управление проектами.**

2. Введите фразу в соответствующее поле – станут активны черная и красная кнопки **Сохранить**.

У вас есть выбор.

- Чтобы начать разметку непосредственно в текущей форме – нажмите черную кнопку **Сохранить**, разметьте фразу, как описано на следующем шаге, затем нажмите кнопку **Добавить еще**. Форма очистится для ввода новой фразы. Размеченная фраза появится в списке на вкладке **Данные разметки**.
- Чтобы фраза сразу попала в список вкладки **Данные разметки**, нажмите красную кнопку **Сохранить** – форма закроется, фраза появится в списке.

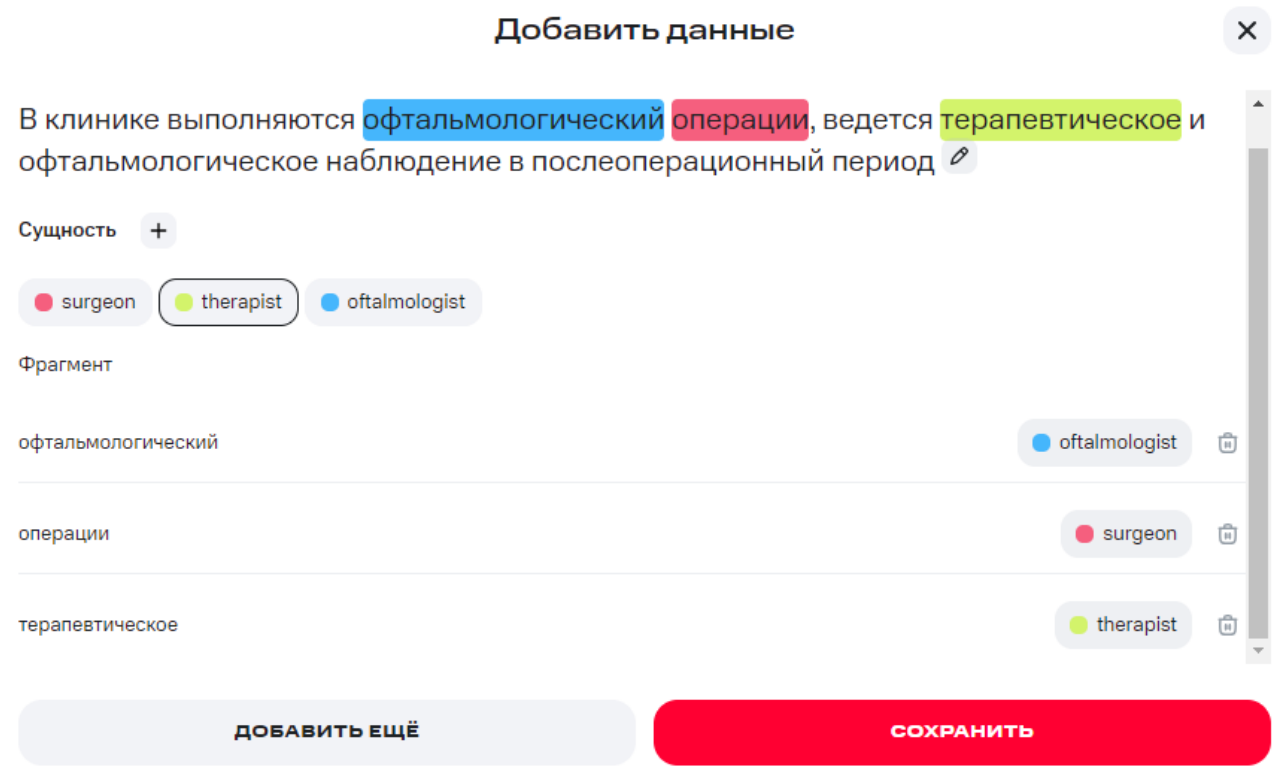


При добавлении фразы система проверяет ее уникальность. Если фраза уже существует в списке **Данные разметки**, она будет перезаписана, чтобы не произошло дублирование.
Вы можете добавить неограниченное количество фраз. Чтобы отредактировать сохраненную фразу, наведите на нее курсор – появится значок . Нажмите на него и приступайте к правке.

3. Приступить к разметке данных можно непосредственно в форме **Добавить данные** в процессе добавления фраз в проект.

Разметка фразы проекта NER.

Выберите в списке метку, а затем выделите мышкой сущность во фразе, которой следует эту метку присвоить. Фрагмент подсветится цветом, идентичным цвету метки. Размеченные фрагменты будут отображаться в виде списка. Вы можете добавлять и присваивать неограниченное количество меток. Удалить размеченный фрагмент фразы можно значком .



Разметка фразы проекта Классификатор. Класс присваивается всей фразе. Поэтому вам достаточно кликнуть по нужной метке. Метка будет присвоена фразе целиком.

4. Чтобы закончить разметку, нажмите кнопку **Сохранить** – форма закроется. Размеченная фраза появится в списке на вкладке **Данные разметки**. Если вы хотите продолжить добавлять и размечать фразы в текущей экранной форме, нажмите кнопку **Добавить еще.** Форма очистится для ввода новой фразы. Готовая фраза отобразится в списке на вкладке **Данные разметки**.

   При добавлении размеченной фразы система проверяет уникальность ее разметки. Если фраза уже существует в списке **Данные разметки**. Она будет перезаписана, чтобы не произошло дублирование. Если вы присвоили фразе-дубликату новые лейблы, они перезапишутся поверх старых. Система оповестит вас об этом сообщением: **Найден дубликат. Разметка перезаписана.**

   Количество фраз, которые можно создать и разметить через форму **Добавить данные**, не ограничено.

Данные добавлены в проект.

Загрузка файла с данными разметки

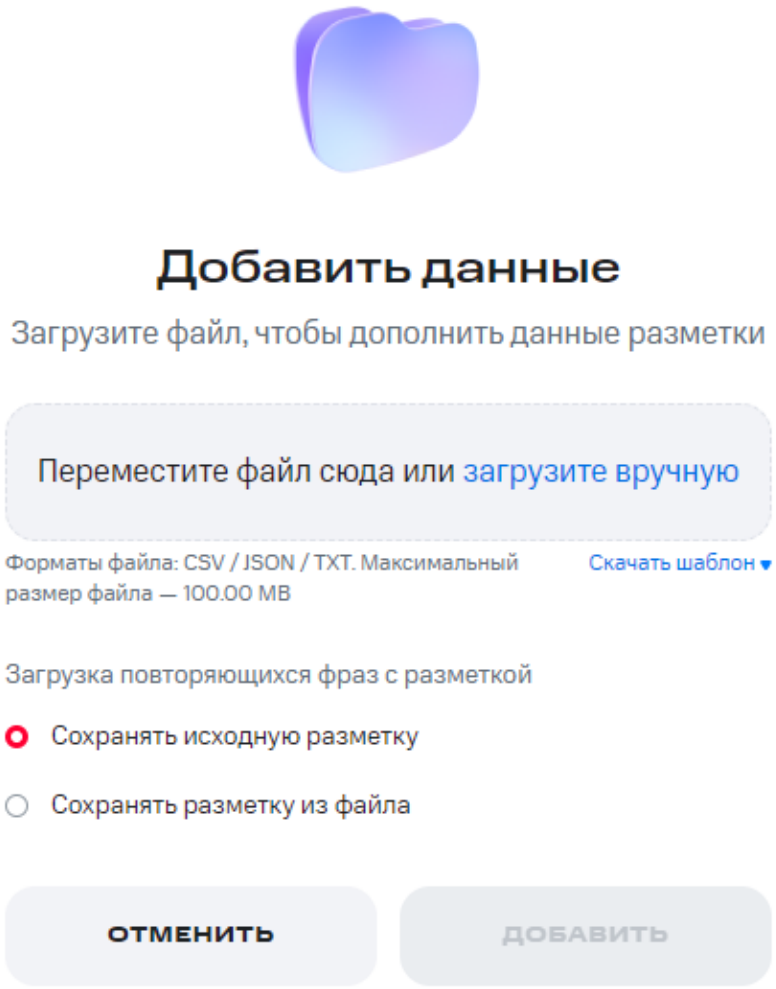
Для загрузки файла:

1. Нажмите кнопку **Загрузить файл** – откроется форма загрузки.

   Файлы должны отвечать следующим требованиям:

   - размер не более 100 MB;
   - формат файла для классификатора – CSV;
   - формат файла для NER – JSON;
   - формат для загрузки неразмеченного текста в проект любого типа – TXT.

   Написать файл разметки можно по шаблону. Чтобы получить шаблон, нажмите **Скачать шаблон** и выберите нужный вам тип из выпадающего списка.



2. Загрузите подготовленный файл с данными разметки: перетащите его в поле загрузки или воспользуйтесь стандартным проводником.

Если вы ошибочно загрузили CSV для NER или JSON для классификатора – то в проект загрузятся только фразы, разметка будет проигнорирована системой.

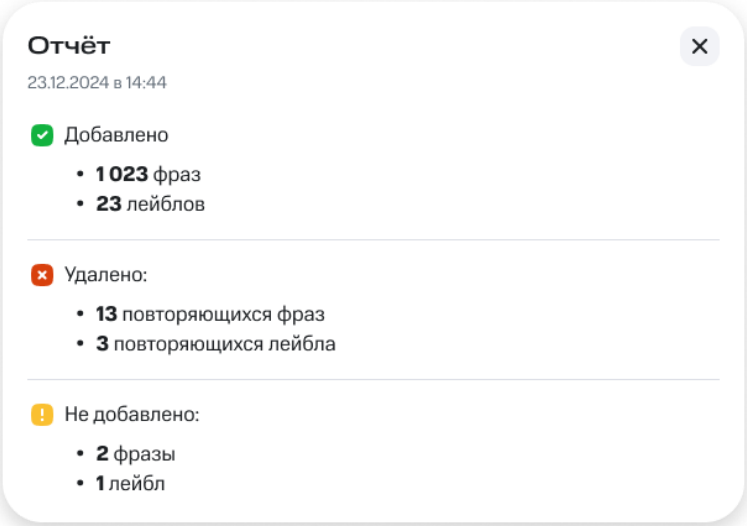
На форме загрузки есть опции **Сохранять исходную разметку/Сохранять разметку из файла**. Если файл загружается в проект, в котором уже есть данные, эти опции позволяют сохранить существующую разметку проекта или заменить ее разметкой из файла.
В процессе загрузки новых данных в дополнение к существующим, система отслеживает повторяющиеся фразы и лейблы. Сравнивается список фраз и меток, загружаемых из файла с теми, что уже есть в системе.
Вот такими могут быть результат сравнения.

- Если у фразы с меткой в системе обнаружен дубль с меткой, то фраза не загружается. Метка загружается без привязки к фразе, но при условии, что ее еще нет в системе.
- Если у фразы с меткой в системе обнаружен дубль без метки, то фраза не загружается. Дублю в системе присваивается метка фразы из файла данных. Метка может быть загружена в систему при условии, что ее еще нет в системе.
- Неразмеченные фразы загружаются, если у них нет дублей в системе.
- Если в файле данных обнаружена метка без фразы, она будет загружена, если у нее в системе нет дубля.

Если вы загружаете файл в пустой проект, выбор опции не имеет значения – загрузятся все без исключения данные и лейблы из файла.

О загрузке данных в существующий список проекта разметки см. в разделе **Управление проектами**.

3. Чтобы проверить статистику загрузки, нажмите **Посмотреть отчет**.



Файл с данными загружен.

Разметка данных в проекте

Чтобы разметить данные в проекте:

1. Откройте проект и перейдите на вкладку **Данные разметки**.

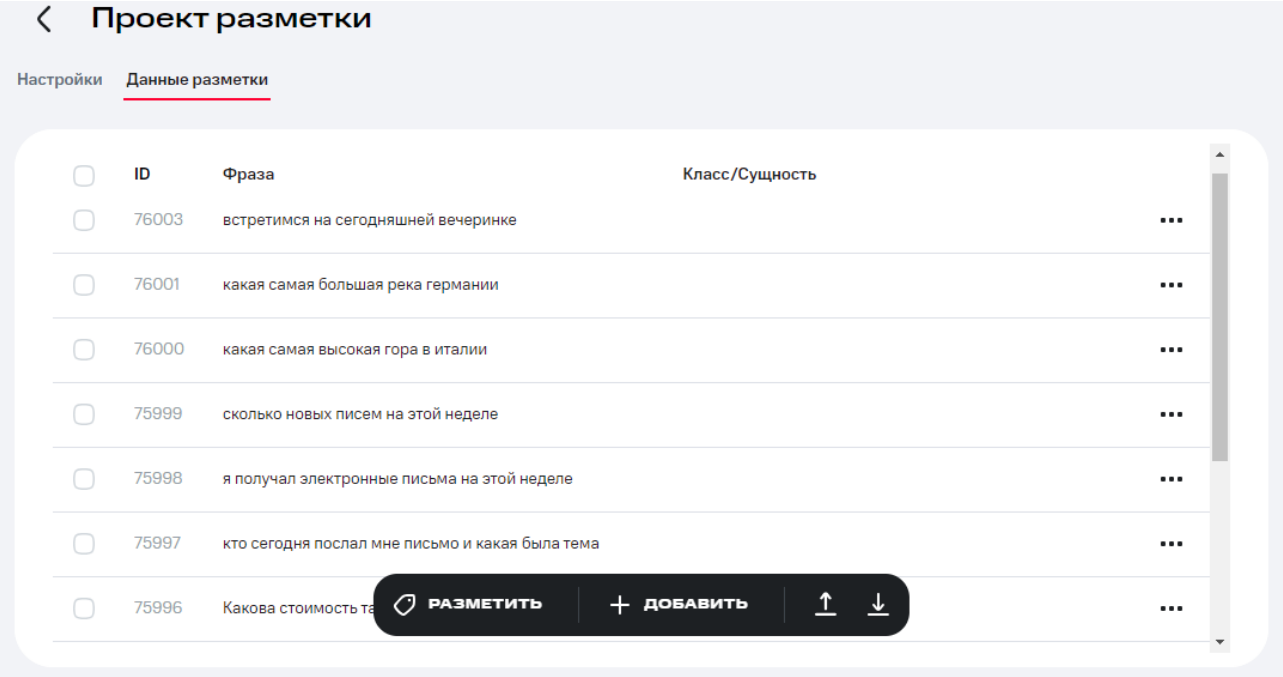
Вы увидите список:

- **id** – уникальный номер фразы в рамках проекта разметки;
- **Фраза** – собственно размечаемая фраза;
- **Класс/сущность** – метки присвоенные фразе или ее сэмплам.

Если фразы не размечены, столбец Класс/сущность будет пустым.

Управлять списком в процессе разметки можно с помощью тулбара внизу страницы и кнопки вызова контекстного меню ⋯.

Можно отметить одну, несколько или сразу все фразы для массовых операций удаления и выгрузки выбранных фраз в датасет.



2. Чтобы разметить любую фразу из списка, вызовите ее контекстное меню и в нем выберите пункт **Разметить**. Либо щелкните кнопку разметить на тулбаре: в этом случае разметка начнется с первой фразы в списке.

Откроется форма **Разметка**.

3. Разметьте фразу в соответствии с заданием на разметку.

Если размечается проект **NER** – выберите в списке метку, а затем выделите мышкой фрагмент фразы, которому эту метку следует присвоить. Фрагмент подсветится цветом, идентичным цвету метки. Размеченные фрагменты будут отображаться внутри формы в

виде списка. Вы можете добавлять и присваивать неограниченное количество меток. Удалить размеченный фрагмент фразы можно значком  .

Если размечаете **Классификатор** выберите метку класса, который необходимо присвоить фразе.

Если в проекте еще нет меток, или возникла необходимость создать новые, нажмите  и введите в появившемся поле наименование метки, затем снова нажмите кнопку  . Новая метка появится в списке. Количество меток не ограничено.

Чтобы отредактировать фразу, наведите на нее курсор – появится значок  . Нажмите его и приступайте к правке.

4. Чтобы перейти к следующей или предыдущей фразе, нажмите соответствующую кнопку внизу формы.
5. Повторите шаги 3 и 4 для всех фраз в списке.

Разметка завершена.

Для проектов **Классификатор** в столбце **Класс/сущность** появились присвоенные фразам метки.

Для проектов **NER**, помимо заполненного столбца **Класс/сущность**, размеченные фрагменты фраз обозначаются подчеркиванием в цветах присвоенных меток.

Управление проектами

В процессе работы с проектами разметки вы можете редактировать, добавлять или удалять как сами проекты, так и содержащиеся в них данные разметки.

Основные настройки проекта

На вкладке **Настройки** вы можете:

- Изменить название проекта;
- Добавить или отредактировать описание проекта;
- Добавить или удалить новые метки классов/сущностей (подробнее см. Создать проект разметки данных)

Добавить данные в проект

Чтобы добавить в существующий проект новые данные:

Добавить данные в проект разметки можно на любом этапе работы с проектом.

1. Откройте проект и перейдите на вкладку **Данные разметки**.
2. В тулбаре нажмите кнопку **Добавить данные** – откроется одноименная экранная форма.
3. Добавьте данные как описано в разделе Добавление данных в проект.

Данные добавлены.

Загрузка данных разметки из файла

Вы можете дополнить существующий список данных новым данными из файла разметки.

Чтобы загрузить данные:

1. Откройте проект и перейдите на вкладку **Данные разметки**.
2. В тулбаре нажмите кнопку  – откроется форма **Загрузка данных.**
3. Добавьте данные, как описано в разделе Загрузка файла с данными разметки.

   При импорте данных система проверяет наличие дубликатов фраз в проекте и файле данных. Дубли не загружаются.
   На форме загрузки можно выбрать, сохранять для дублирующихся фраз текущую разметку проекта или заменить её на разметку из файла.
   Для этого используйте опцию **Сохранять исходную разметку** или **Сохранять разметку** из файла соответственно.

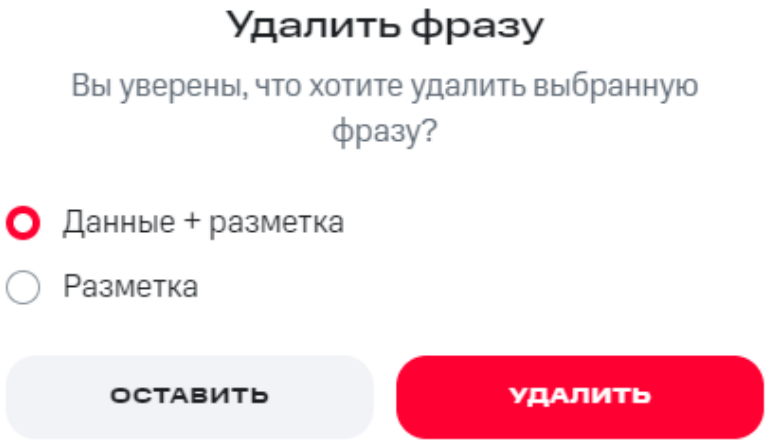
Данные из файла загружены.

Удаление данных разметки

Для удаления данных:

1. Откройте проект и перейдите на вкладку **Данные разметки**.
2. Чтобы удалить одну фразу, вызовите контекстное меню и нажмите кнопку **Удалить**. Для массового удаления выберите несколько фраз в списке или сразу весь список. Нажмите значок  .

   Откроется диалоговое окно:



3. Выберите способ удаления данных: **Данные+разметка** – удаляются фразы вместе с разметкой, т.е. все данные проекта. Список **Данные разметки** станет пустым.
   Разметка – удаляется привязка меток к фразам, сами фразы и метки останутся в проекте раздельно.
4. **Нажмите Удалить.**

В результате в зависимости от выбранного способа будут удалены фразы или только их разметка.

Дублирование проекта

Быстрый способ создать проект разметки – продублировать существующий.

Для этого:

1. Вызовите контекстное меню проекта в списке проектов и нажмите **Дублировать**.
2. В диалоговом окне выберите способ дублирования:
   Данные – проект будет скопирован без разметки, только фразы и список меток – вы сможете переразметить данные.
   Данные + разметка – в этом случае проект будет дублирован полностью.

3. Пустой проект дублируется по умолчанию, без диалогового окна. Если в нем были созданы метки – они также будут скопированы.

Проект дублирован. Новый проект появился в списке под тем же название с префиксом **Копия N** (где N – порядковый номер проекта в системе).

Скачивание датасета

Чтобы скачать результат разметки данных – датасет:

1. Откройте проект и перейдите на вкладку **Данные разметки**.
2. Отметьте фразу или выборку фраз, которые должны быть включены в датасет. Затем нажмите 

   Чтобы скачать полный датасет, выберите все фразы или не выбирайте ни одной и нажмите кнопку  .

Файл с датасетом соответствующего типа скачан в папку **Загрузки**.

Удаление проекта разметки

Чтобы удалить проект разметки, вызовите его контекстное меню и нажмите **Удалить**.

Подтвердите удаление.

Все данные разметки, связанные с проектом, будут удалены.