



РУКОВОДСТВО АДМИНИСТРАТОРА

Версия 2025.3.0

СОДЕРЖАНИЕ

Аннотация	3
Термины и сокращения	4
1. Лицензии	5
2. Управление учётными записями	6
2.1 Категории учетных записей	6
2.2 Управление техническими учетными записями	6
2.3 Управление пользовательскими учётными записями	8
3. Управление подключениями к моделям	12
3.1 Создание нового подключения	12
3.2 Удаление подключения	13
4. Управление моделями	14
4.1 Регистрация новой модели	14
4.2 Удаление моделей	14
5. Управление областями знаний	16
5.1 Настройка и индексация области знаний	16
5.2 Тестирование области знаний	17
6. Управление шаблонами промптов	18
7. Мониторинг использования токенов	19
7.1 Настройка периода отображения данных	19
8. API AI BOX	20
8.1 Проверка состояния (Health)	20
8.2 Версия API (Version)	20
8.3 Срок действия лицензии (License)	21
8.4 Chain API	21
8.5 Copilot API	27
8.6 Conversation Tone API	30
8.7 History API	32
8.8 Config API	33
8.9 Data Cortex API	41
8.10 Prompt Templates API	51
8.11 LLM Request Raw API	53
8.12 Periods API	54
8.13 Time Ranges API	56
8.14 Dashboard API	57
9. Настройка коннекторов с внешними системами	59
9.1 Настройка коннектора с ITSM box	59
9.2 Настройка коннектора к внешней системе мониторинга на примере подключения к Prometheus в Grafana	59

АННОТАЦИЯ

Данный документ содержит описание настройки AI BOX Администратором.
Администратору системы доступны следующие функции:

- Получение информации о лицензии на AI BOX;
- управление учётными записями;
- управление подключениями к моделям;
- управление моделями;
- управление областями знаний;
- управление шаблонами промптов;
- мониторинг расходования токенов;
- описание доступных методов API AI BOX;
- настройка коннекторов с внешними системами.

ТЕРМИНЫ И СОКРАЩЕНИЯ

API – это набор правил и инструкций, который позволяет разным программам взаимодействовать друг с другом и обмениваться данными.

Embedding – векторное представление данных для поиска и анализа.

LLM – Large Language Model, большая языковая модель.

Индексация – это процесс, который включает в себя подготовку данных области знаний, разделение данных на фрагменты, преобразование текстовых фрагментов в векторы.

Область знаний – набор информации, которая является источником знаний для LLM при формировании ответов.

Провайдер – сервис, предоставляющий функциональность LLM или embedding.

Промпт – ввод, инструкция, предоставляемый модели для выполнения задачи.

Токен – минимальная единица текста, с которой работает LLM.

1. ЛИЦЕНЗИИ

В административной консоли AI BOX доступна возможность получить информацию, какая версия AI BOX используется, а также, до какого срока действует лицензия.

Для этого в интерфейсе административной консоли AI BOX требуется в правом верхнем углу нажать на шестеренку. Откроется модальное окно, в котором отображается:

- название учётной записи текущего пользователя;
- кнопка выхода из системы;
- версия AI BOX;
- срок действия лицензии.

2. УПРАВЛЕНИЕ УЧЁТНЫМИ ЗАПИСЯМИ

2.1 Категории учетных записей

Выделяются 2 категории учетных записей:

- пользовательские (персонифицированные) учетные записи, предназначены для предоставления доступа к пользовательскому интерфейсу AI BOX;
- технические учетные записи, предназначены для предоставления доступа к AI BOX на уровне программных интерфейсов (API).

2.2 Управление техническими учетными записями

Работа с техническими учетными записями осуществляется в разделе «Clients» административной консоли Keycloak. Администратор может:

- создавать и редактировать новые технические учетные записи;
- блокировать технические учетные записи;
- удалять технические учетные записи.

2.2.1 Создание технической учетной записи

Чтобы создать техническую учётную запись, необходимо выполнить следующие действия:

1. **Открыть** административную консоль Keycloak.
2. **Открыть** раздел **Clients**.
3. **Нажать Create client**. Откроется окно создания клиента на вкладке **General Settings**.
4. **Заполнить** параметры:
 - **Client type**: OpenID Connect
 - **Client ID**: уникальный идентификатор (например, `llm-client`)
5. **Нажать Next**.
6. **Открыть** вкладку **Capability config**.
7. **Выставить** параметры:
 - **Client authentication**: ON
 - **Authorization**: OFF

- **Authentication flow:**

- Standard flow — OFF
- Direct access grants — OFF
- Implicit flow — OFF
- Service accounts roles — ON
- OAuth 2.0 Device Authorization Grant — OFF
- OIDC CIBA Grant — OFF

8. Нажать **Next**.

9. Открыть вкладку **Login settings**.

10. Сохранить нового клиента.

Клиент создан.

В некоторых случаях требуется добавить в токен дополнительные параметры (claims), используемые внешними системами при интеграции. Чтобы добавить эти параметры, необходимо создать маппер для соответствующего клиента:

1. Открыть вкладку **Client scopes** клиента.
2. Найти запись [название_клиента]-dedicated.
3. Открыть найденную запись.
4. Перейти на вкладку **Mappers**.
5. Выбрать действие **Add mapper → By configuration**. Откроется окно **Configure a new mapper**.

6. Выбрать тип маппера **Hardcoded claim**.

Откроется окно добавления маппера.

7. Заполнить параметры маппера.

8. Сохранить маппер.

Новый маппер добавлен.

2.2.2 Блокировка технической учётной записи

Чтобы временно отключить техническую учётную запись (клиента) и заблокировать возможность её использования при интеграции, необходимо выполнить следующие действия:

1. Открыть административную консоль **Keycloak**.

2. Открыть раздел **Clients**.

3. В списке клиентов выбрать нужную техническую учётную запись. Откроется карточка выбранного клиента.

4. В правом верхнем углу карточки клиента переключить тумблер **Enabled** в положение **OFF** (тумблер станет серым).

5. Сохранить изменения.

Техническая учётная запись заблокирована.

Все запросы от этого клиента будут отклоняться до момента повторного включения (возврата тумблера Enabled в состояние ON).

2.2.3 Удаление технической учётной записи

Чтобы полностью удалить техническую учётную запись (клиента) из Keycloak, выполните следующие действия:

1. Открыть административную консоль **Keycloak**.

2. Открыть раздел **Clients**.

3. В списке клиентов выбрать нужную техническую учётную запись. Откроется карточка выбранного клиента.

4. В правом верхнем углу карточки выбрать действие **Actions** → **Delete**.

5. Подтвердить удаление в открывшемся окне подтверждения.

6. Техническая учётная запись удалена.

Все связанные с ней настройки (роли, мапперы, client secrets) также будут удалены без возможности восстановления.

2.3 Управление пользовательскими учётными записями

Работа с пользовательскими учётными записями выполняется в разделе «**Users**» административной консоли Keycloak.

Администратор может выполнять следующие действия:

- Создавать новых пользователей.
- Блокировать учетные записи.
- Удалять пользователей.
- Сбрасывать пароли.
- Завершить пользовательскую сессию.

2.3.1 Создание пользовательской учётной записи

Чтобы создать пользовательскую учётную запись, необходимо выполнить следующие действия:

1. Открыть административную консоль **Keycloak**.
 2. Открыть раздел **Users**.
 3. Нажать **Add user**. Откроется окно создания пользователя.
 4. Заполнить параметры:
 - **Username** — обязательное поле.
 - **Email** — опционально.
 - **Required user action** — опционально.
 5. Нажать **Create**. Пользователь создан.
 6. Для задания пароля:
 - Открыть вкладку **Credentials** → **Set password**.
 - Ввести новый пароль.
 - Настроить флаг **Temporary**:
 - **ON** — пользователь должен сменить пароль при первом входе.
 - **OFF** — пароль постоянный.
- Пароль задан.

2.3.2 Блокировка пользователя

Чтобы заблокировать пользовательскую учётную запись, необходимо выполнить следующие действия:

1. Открыть административную консоль **Keycloak**.
2. Открыть раздел **Users**.
3. Найти и выбрать нужного пользователя. Откроется карточка пользователя.
4. В правом верхнем углу переключить тумблер **Enabled** в положение **OFF**. Тумблер станет серым.
5. Учётная запись заблокирована. Пользователь не сможет авторизоваться в системе до повторного включения параметра **Enabled**.

2.3.3 Удаление пользователя

Чтобы удалить пользовательскую учётную запись, необходимо выполнить следующие действия:

1. Открыть административную консоль **Keycloak**.
2. Открыть раздел **Users**.
3. Найти и выбрать нужного пользователя. Откроется карточка пользователя.
4. В правом верхнем углу выбрать **Action** → **Delete**.
5. Подтвердить удаление.

Пользователь удалён.

2.3.4 Сброс пароля пользователя

Чтобы сбросить пароль пользователя, необходимо выполнить следующие действия:

1. Открыть административную консоль **Keycloak**.
2. Открыть раздел **Users**.
3. Найти и выбрать нужного пользователя. Откроется карточка пользователя.
4. Открыть вкладку **Credentials**.
5. У строки с **Type = Password** выбрать действие **Reset password**. Откроется окно

Reset password for.

6. Ввести новый пароль.
7. Настроить флаг **Temporary**:
 - **ON** — пользователь должен сменить пароль при первом входе.
 - **OFF** — пароль постоянный.
8. Нажать **Save**.

Пароль пользователя обновлён.

2.3.5 Завершение пользовательской сессии

Чтобы завершить активные сессии пользователя, необходимо выполнить следующие действия:

1. Открыть административную консоль **Keycloak**.
2. Открыть раздел **Users**.
3. Найти и выбрать нужного пользователя. Откроется карточка пользователя.

4. Открыть вкладку **Sessions**. Отобразится список активных сессий пользователя.
5. Нажать **Logout all sessions**.

Все активные сессии пользователя завершены.

3. УПРАВЛЕНИЕ ПОДКЛЮЧЕНИЯМИ К МОДЕЛЯМ

Подключение к моделям – это объект системы, который хранит параметры для безопасного взаимодействия с внешними сервисами, предоставляющими языковые модели (LLM) или модели эмбедингов.

В подключении хранятся следующие данные:

- учётные данные (например, API-ключ);
- версия API;
- дополнительные сервисные переменные, необходимые для авторизации и работы с сервисом.

Для взаимодействия с внешним сервисом используется понятие «Провайдер». Провайдер определяет способ подключения к конкретной платформе. Платформа может быть:

- публичным веб-приложением;
- внутренним API-шлюзом, развернутым в корпоративном контуре компании.

Работа с подключениями выполняется в разделе «Подключения» административной консоли AI BOX.

Администратор может выполнять следующие действия:

- создавать новые подключения;
- редактировать параметры существующих подключений;
- удалять неиспользуемые подключения/

3.1 Создание нового подключения

Чтобы создать новое подключение, требуется выполнить следующие действия:

1. Открыть административную консоль AI BOX.
2. Перейти в раздел «Подключения».
3. Нажать **Создать подключение**. Откроется окно создания нового подключения.
4. Выбрать **Провайдера**, соответствующего целевому сервису.
5. Заполнить параметры подключения
 - ключ доступа (API Key);
 - другие служебные параметры, требуемые провайдером для авторизации (например, базовый URL).

6. Нажать **Сохранить**.

После сохранения подключение станет доступным для выбора в настройках моделей.

3.2 Удаление подключения

Подключения, которые устарели или выведены из эксплуатации, необходимо удалять из системы, чтобы избежать ошибок при выполнении запросов.

Чтобы удалить подключение:

1. Открыть раздел **«Подключения»**.
2. Найти подключение, подлежащее удалению.
3. Выбрать действие **Удалить** и подтвердить операцию.

Подключение будет удалено из системы.

4. УПРАВЛЕНИЕ МОДЕЛЯМИ

Управление моделями выполняется в разделе «**Модели**» административной консоли AI BOX. В этом разделе администратор может выполнять следующие действия:

- создавать новые записи о моделях;
- редактировать параметры существующих моделей;
- удалять неиспользуемые модели.

4.1 Регистрация новой модели

Если провайдер предоставил новую модель, её необходимо вручную зарегистрировать в системе. Чтобы зарегистрировать модель, требуется выполнить следующие действия:

1. Открыть административную консоль AI BOX.
2. Перейти в раздел «**Модели**».
3. Нажать **Создать модель**. Откроется карточка регистрации модели.
4. Выбрать **тип модели**, определяющий её функциональное назначение:

- **LLM (Large Language Model)** – используется для генерации текстовых ответов в чат-боте. Модель получает промпт и фрагменты текста из области знаний и формирует связный ответ пользователю.

- **Embedding (Эмбединг)** – применяется для индексации области знаний. Модель преобразует текстовые фрагменты документов в числовые векторные представления (эмбединги), что позволяет системе быстро и точно находить релевантную информацию.

5. Указать параметры модели, включая:

- подключение (источник, через который выполняются запросы к модели);
- стоимость входящих и исходящих токенов.

6. Нажать **Сохранить**.

Модель будет добавлена в систему и станет доступна для использования.

4.2 Удаление моделей

Модели, которые устарели или выведены провайдером из эксплуатации, необходимо удалять из системы, чтобы избежать ошибок при выполнении запросов.

Чтобы удалить модель:

4. Открыть раздел **«Модели»**.
5. Найти модель, подлежащую удалению.
6. Выбрать действие **Удалить** и подтвердить операцию.

Модель будет удалена из системы.

5. УПРАВЛЕНИЕ ОБЛАСТЯМИ ЗНАНИЙ

Область знаний используется для определения набора статей базы знаний или документов, которые чат-бот может использовать для поиска релевантной информации. Эта функция позволяет чат-боту давать точные ответы на часто задаваемые вопросы, автоматизируя поддержку и повышая уровень обслуживания клиентов и сотрудников.

Процесс настройки области знаний состоит из следующих этапов:

1. Создание конфигурации.
2. Индексация
3. Тестирование работы.
4. Настройка доступа в каналах коммуникации.

5.1 Настройка и индексация области знаний

Шаг 1. Создание области знаний и загрузка данных

1. Открыть раздел **«Область знаний»** административной консоли AI BOX.
2. Нажать **Создать область знаний**. Откроется карточка создания области знаний.
3. Заполнить поля карточки:
 - **Название области знаний** – указать имя;
 - **Источник данных** – выбрать источник для загрузки данных.
4. Загрузить данные из выбранного источника.

Шаг 2. Индексация области знаний

1. Открыть карточку области знаний.
2. Заполнить параметры индексации:
 - **Максимальная длина фрагмента** – указать размер фрагмента в символах для оптимального поиска;
 - **Перекрытие фрагментов** – задать количество символов для сохранения контекста между фрагментами документа;
 - **Используемый эмбеддер** – выбрать модель для преобразования текстовых фрагментов в векторные представления.
3. Нажать **Запустить индексацию**.

Примечание. Процесс индексации может занять некоторое время в зависимости от объёма загруженных данных. Закрывать карточку области знаний можно сразу после запуска процесса; уведомление о завершении отобразится в интерфейсе продукта.

4. Проверить результат индексации в карточке области знаний.
5. При загрузке новых данных или изменении параметров индексации повторно выполнить индексацию аналогично первоначальной.

5.2 Тестирование области знаний

1. Открыть карточку области знаний.
2. Перейти в подраздел **Тестирование**. Отобразится эмулятор чат-бота и панель настройки тестирования.

5.2.1 Настройка параметров тестирования

1. В панели **Параметры тестирования** заполнить поля:
 - **LLM модель** – выбрать языковую модель для генерации ответа;
 - **Шаблон промпта** – выбрать существующий или создать новый промпт, который инструктирует LLM, как формировать ответ на основе полученных данных.
2. Нажать **Применить** для применения настроек к текущему диалогу тестирования.

5.2.2 Проведение тестирования

При проведении тестирования области знаний можно руководствоваться следующими рекомендациями:

1. Задавать вопросы в чате и анализировать ответы системы. Каждый ответ сопровождается ссылкой на файл-источник для проверки достоверности информации.
2. Использовать панель **Журнал исполнения** для анализа логов запросов к LLM и просмотренных текстовых фрагментов области знаний.
3. Тестировать область знаний на различных вопросах, экспериментируя с настройками LLM и промптов.
4. При необходимости вернуться к настройкам области знаний, выполнить переиндексацию и продолжить тестирование.
5. По окончании тестирования сохранить рабочую версию промпта.

6. УПРАВЛЕНИЕ ШАБЛОНАМИ ПРОМПТОВ

Шаблон промпта – это подробная инструкция, которая задаёт модели контекст, правила и ограничения, определяя содержание, тон и структуру её ответов.

Регистрация и редактирование шаблонов промптов выполняется в разделе **«Шаблоны промптов»** административной консоли AI BOX.

Кроме того, создать или изменить промпт можно непосредственно в процессе тестирования конкретной области знаний.

Примечание. Качество промпта напрямую влияет на точность и полезность ответов чат-бота. Рекомендуется создавать, редактировать и тестировать промпты совместно с конкретной областью знаний, для которой они предназначены.

7. МОНИТОРИНГ ИСПОЛЬЗОВАНИЯ ТОКЕНОВ

Для отслеживания динамики расхода токенов LLM предусмотрены дашборды в разделе «**Мониторинг**» административной консоли AI BOX.

На дашбордах отображаются следующие показатели за выбранный период (по умолчанию – текущий месяц):

- **Расход токенов LLM по моделям** — показывает общее потребление токенов для каждой языковой модели, позволяя оценить нагрузку на разные модели;
- **Стоимость запросов к LLM по моделям (руб.)** — отображает финансовые затраты, рассчитанные на основе потребленных токенов и тарифов провайдеров, что помогает контролировать бюджет;
- **Количество запросов к LLM по моделям** — считает общее число обращений к каждой языковой модели, позволяя анализировать активность использования моделей;
- **Расход токенов эмбединга** — показывает потребление токенов моделями векторизации, используемыми для индексации и поиска по областям знаний.

7.1 Настройка периода отображения данных

Для настройки периода отображения данных требуется выполнить действия:

1. Открыть раздел «**Мониторинг**».
2. На дашборде выбрать период отображения данных.
3. Изменить период для анализа динамики показателей (например, за прошлый месяц или за последнюю неделю).

После настройки периода на дашбордах отобразится статистика потребления токенов и стоимости запросов за выбранный интервал, что позволяет выявлять тенденции и контролировать использование AI-моделей.

8. API AI BOX

Для интеграции с внешними системами AI BOX предоставляет набор инструментов для обмена данными через API.

Доступ к API возможен только под действующей технической учетной записью.

8.1 Проверка состояния (Health)

- **Эндпоинт:** /health
- **Метод:** GET
- **Авторизация:** не требуется
- **Описание:** Позволяет проверить доступность сервиса.

Пример запроса:

```
GET /health
```

Пример ответа:

```
{  
  "status": "ok"  
}
```

Коды ответов:

Код	Описание
200	Успешно обработано

8.2 Версия API (Version)

- **Эндпоинт:** /version
- **Метод:** GET
- **Авторизация:** не требуется
- **Описание:** Позволяет получить информацию о текущей версии API.

Пример запроса:

```
GET /version
```

Пример ответа:

```
{  
  "version": "2025.4.0"  
}
```

```
}
```

Коды ответов:

Код	Описание
200	Успешно обработано

8.3 Срок действия лицензии (License)

- **Эндпоинт:** `/license`
- **Метод:** GET
- **Авторизация:** не требуется
- **Описание:** Позволяет получить информацию о лицензии системы.

Пример запроса:

```
GET /license
```

Пример ответа:

```
{  
  "expiration_date": "2025-10-23T08:36:21.871Z"  
}
```

Коды ответов:

Код	Описание
200	Успешно обработано

8.4 Chain API

Раздел **Chain** включает методы для взаимодействия с языковой моделью (LLM), включая выполнение инференса, выбор модели, классификацию и работу с историей сообщений.

8.4.1 Обработка запроса к LLM с использованием RAG

- **Эндпоинт:** `/chain/message`
- **Метод:** POST
- **Авторизация:** обязательна (HTTP Bearer)

- **Описание:** Выполняет обработку запроса к LLM с использованием базы знаний (RAG).

Параметры запроса:

Параметр	Тип	Обязательно	Описание
user_name	string	Да	Имя пользователя
client_id	string	Да	Идентификатор клиента
vectorstore_name	string	Нет	Имя хранилища векторных представлений
message	string	Да	Текст запроса
chat_id	string	Да	Идентификатор чата
llm_model_id	string	Да	Идентификатор LLM модели
embeddings_model_id	string	Да	Идентификатор модели эмбеддингов
process_element_id	string	Нет	Идентификатор элемента процесса
input_data	string	Нет	Входные данные
input_data_format_id	string	Нет	Формат входных данных

Пример запроса:

```
{
  "user_name": "string",
  "client_id": "string",
  "vectorstore_name": "",
  "message": "Пример запроса пользователя",
  "chat_id": "session_001",
  "llm_model_id": "model_01",
  "embeddings_model_id": "embed_01",
  "process_element_id": "",
  "input_data": "context",
  "input_data_format_id": "json"
}
```

Коды ответов:

Код	Описание
200	Успешно обработано
422	Ошибка валидации

8.4.2 Обработка запроса к LLM без RAG

- **Эндпоинт:** /chain/llm-request
- **Метод:** POST
- **Авторизация:** обязательна (HTTP Bearer)

- **Описание:** Выполняет обращение к LLM без использования базы знаний.

Параметры запроса: аналогичны /chain/message

Пример запроса:

```
{
  "user_name": "string",
  "client_id": "string",
  "vectorstore_name": "",
  "message": "Как работает модель без RAG?",
  "chat_id": "chat_123",
  "llm_model_id": "llama-8B",
  "embeddings_model_id": "embed-01",
  "process_element_id": ""
}
```

Коды ответов:

Код	Описание
200	Успешно обработано
422	Ошибка валидации

8.4.3 Обработка запроса к LLM с изображениями

- **Эндпоинт:** /chain/llm-request-with-images
- **Метод:** POST
- **Авторизация:** обязательна
- **Описание:** Выполняет обращение к LLM с возможностью передачи изображений.

Параметры:

Параметр	Тип	Обязательно	Описание
images	array<binary>	Да	Список изображений
message	string	Да	Текст запроса

Пример запроса:

```
{
  "message": "Определи, что изображено на фото.",
  "images": ["image1.jpg", "image2.jpg"]
}
```

Коды ответов:

Код	Описание
200	Успешно обработано
422	Ошибка валидации

8.4.4 Перезапуск сессии

- **Эндпоинт:** `/chain/restart`
- **Метод:** `POST`
- **Авторизация:** обязательна
- **Описание:** Перезапускает активную сессию для указанного пользователя и клиента.

Параметры запроса:

Параметр	Тип	Обязательно	Описание
<code>user_name</code>	<code>string</code>	Да	Имя пользователя
<code>client_id</code>	<code>string</code>	Да	Идентификатор клиента
<code>vectorstore_name</code>	<code>string</code>	Нет	Имя векторного хранилища

Пример запроса:

```
{
  "user_name": "test_user",
  "client_id": "client_001",
  "vectorstore_name": "default"
}
```

Коды ответов:

Код	Описание
200	Успешно обработано
422	Ошибка валидации

8.4.5 Выбор модели LLM

- **Эндпоинт:** `/chain/selectmodel`
- **Метод:** `POST`
- **Авторизация:** обязательна
- **Описание:** Устанавливает активную LLM модель для работы текущей сессии.

Параметры запроса:

Параметр	Тип	Обязательно	Описание
user_name	string	Да	Имя пользователя
client_id	string	Да	Идентификатор клиента
vectorstore_name	string	Нет	Имя хранилища
llm_model_id	string	Да	Идентификатор модели LLM

Пример запроса:

```
{
  "user_name": "test_user",
  "client_id": "client_01",
  "vectorstore_name": "",
  "llm_model_id": "meta-llama-3-8B"
}
```

Коды ответов:

Код	Описание
200	Успешно обработано
422	Ошибка валидации

8.4.6 Классификация текста

- **Эндпоинт:** /chain/classify
- **Метод:** POST
- **Авторизация:** обязательна
- **Описание:** Выполняет классификацию переданного текста по заданной модели.

Параметры запроса:

Параметр	Тип	Обязательно	Описание
classification_element_id	string (uuid4)	Да	Идентификатор элемента классификации
classification_text	string	Да	Текст для классификации
classification_examples	array<object>	Да	Примеры для обучения
classification_model_id	string	Да	Идентификатор модели классификации

Пример запроса:

```
{
```

```

"classification_element_id": "6b98b3f0-1111-4bcd-b8d2-7c5b12eac312",
"classification_text": "Клиент недоволен качеством обслуживания",
"classification_examples": [
  {"label": "Положительный", "text": "Все отлично"},
  {"label": "Негативный", "text": "Очень плохо"}
],
"classification_model_id": "model_23"
}

```

Коды ответов:

Код	Описание
200	Успешно обработано
422	Ошибка валидации

8.4.7 История сообщений

- **Эндпоинт:** /chain/message-history
- **Метод:** POST
- **Авторизация:** обязательна
- **Описание:** Возвращает историю сообщений в рамках указанной сессии.

Параметры запроса:

Параметр	Тип	Обязательно	Описание
session_key	string	Да	Ключ сессии
session_lifetime_min	integer	Да	Время жизни сессии (в минутах)

Пример запроса:

```

{
  "session_key": "session_001",
  "session_lifetime_min": 60
}

```

Коды ответов:

Код	Описание
200	Успешно обработано
422	Ошибка валидации

8.5 Copilot API

Раздел **Copilot** предназначен для работы с интеллектуальным ассистентом (вторым пилотом) – управлением сессиями, отправкой сообщений и получением истории взаимодействий.

8.5.1 Перезапуск дочерней сессии

- **Эндпоинт:** `/copilot/restart`
- **Метод:** `POST`
- **Авторизация:** обязательна (HTTP Bearer)
- **Описание:** Перезапускает дочернюю (подчинённую) сессию копилота.

Параметры запроса:

Параметр	Тип	Обязательно	Описание
<code>session_id</code>	string	Да	Идентификатор сессии
<code>user_name</code>	string	Да	Имя пользователя
<code>client_id</code>	string	Да	Идентификатор клиента

Пример запроса:

```
{
  "session_id": "child_session_01",
  "user_name": "test_user",
  "client_id": "client_123"
}
```

Коды ответов:

Код	Описание
200	Успешно обработано
422	Ошибка валидации

8.5.2 Получение истории сообщений (корневая сессия)

- **Эндпоинт:** `/copilot/copilot-message-history/{session_key}`
- **Метод:** `GET`
- **Авторизация:** обязательна
- **Описание:** Возвращает историю сообщений в рамках корневой сессии копилота.

Параметры запроса:

Параметр	Тип	Обязательно	Описание
session_key	string	Да	Ключ текущей сессии

Пример запроса:

GET /copilot/copilot-message-history/session_abc123

Пример ответа:

```
{
  "messages": [
    {
      "sent_on": "2025-10-23T08:40:20.220Z",
      "message": "string",
      "sent_by_user": true,
      "relative_sources": [
        {
          "name": "string",
          "url": "string",
          "source_type_id": "string"
        }
      ]
    }
  ]
}
```

8.5.3 Отправка сообщения в Copilot

- **Эндпоинт:** /copilot/copilot-message
- **Метод:** POST
- **Авторизация:** обязательна
- **Описание:** Выполняет обработку сообщения от пользователя в чате копилота.

Параметры запроса:

Параметр	Тип	Обязательно	Описание
user_name	string	Да	Имя пользователя
client_id	string	Да	Идентификатор клиента
vectorstore_name	string	Нет	Имя хранилища векторных данных
message	string	Да	Текст сообщения
chat_id	string	Да	Идентификатор чата
llm_model_id	string	Да	Идентификатор LLM модели
embeddings_model_id	string	Да	Идентификатор модели эмбеддингов
process_element_id	string	Нет	Идентификатор элемента процесса
skill_id	string	Нет	Идентификатор навыка

Параметр	Тип	Обязательно	Описание
context_data	string	Нет	Контекстные данные
is_rephrase_mode	boolean	Нет	Режим переформулировки

Пример запроса:

```
{
  "user_name": "test_user",
  "client_id": "client_01",
  "vectorstore_name": "",
  "message": "Создай отчет по активности за неделю",
  "chat_id": "copilot_001",
  "llm_model_id": "meta-llama-3-8B",
  "embeddings_model_id": "embed-01",
  "process_element_id": "",
  "skill_id": "report_gen",
  "context_data": "{\"project\":\"sales\"}",
  "is_rephrase_mode": false
}
```

8.5.4 Отправка нового сообщения в Copilot

- **Эндпоинт:** /copilot/copilot-new-message
- **Метод:** POST
- **Авторизация:** обязательна
- **Описание:** Отправляет новое сообщение в копилот сессию в расширенном формате, включающем структуру conversation_message.

Параметры запроса:

Параметр	Тип	Обязательно	Описание
session_id	string	□□	Идентификатор сессии
user_name	string	□□	Имя пользователя
client_id	string	□□	Идентификатор клиента
conversation_message	object	□□	Объект сообщения (CopilotMessageUpdate)

Пример запроса:

```
{
  "user_name": "test_user",
  "client_id": "client_01",
  "vectorstore_name": "",
  "message": "Создай презентацию по итогам квартала",
  "chat_id": "chat_01",
  "llm_model_id": "meta-llama-3-8B",
  "embeddings_model_id": "embed-01",
  "process_element_id": "",

```

```
"skill_id": "presentation",  
"context_data": "{}",  
"is_rephrase_mode": false  
}
```

8.5.5 Заккрытие дочерней сессии

- **Эндпоинт:** /copilot/close-child-session/{session_key}
- **Метод:** POST
- **Авторизация:** обязательна
- **Описание:** Закрывает дочернюю сессию копилота.

Параметры запроса:

Параметр	Тип	Обязательно	Описание
session_key	string	Да	Ключ дочерней сессии

Пример запроса:

POST /copilot/close-child-session/session_001

Пример ответа:

```
{  
  "result": true  
}
```

8.6 Conversation Tone API

Раздел **Conversation Tone** предназначен для анализа эмоционального и речевого тона переписки между пользователем и специалистом. Методы позволяют классифицировать сообщения по их настроению (тональности) с использованием моделей LLM.

8.6.1 Классификация тона пользователя

- **Эндпоинт:** /conversation-tone/classify-user-tone
- **Метод:** POST
- **Авторизация:** обязательна (HTTP Bearer)
- **Описание:** Выполняет классификацию сообщений пользователя по эмоциональному тону (например: позитивный, нейтральный, негативный).

Параметры запроса:

Параметр	Тип	Обязательно	Описание
llm_model_id	string	Да	Идентификатор используемой LLM модели
case_id	string	Да	Идентификатор обращения (кейса)
messages	array<string>	Да	Список сообщений пользователя

Пример запроса:

```
{
  "llm_model_id": "meta-llama-3-8B",
  "case_id": "case_1452",
  "messages": [
    "Здравствуйте, у меня не работает личный кабинет",
    "Вы вообще собираетесь что-то делать?"
  ]
}
```

Пример ответа:

```
{
  "result": {
    "tone": "Негативный",
    "confidence": 0.91
  }
}
```

Коды ответов:

Код	Описание
200	Успешно обработано
422	Ошибка валидации

8.6.2 Классификация тона специалиста

- **Эндпоинт:** /conversation-tone/classify-specialist-tone
- **Метод:** POST
- **Авторизация:** обязательна (HTTP Bearer)
- **Описание:** Определяет тон сообщений специалиста (например, вежливый, нейтральный, грубый).

Параметры запроса:

Параметр	Тип	Обязательно	Описание
llm_model_id	string	Да	Идентификатор модели LLM
case_id	string	Да	Идентификатор обращения

Параметр	Тип	Обязательно	Описание
messages	array<string>	Да	Список сообщений специалиста

Пример запроса:

```
{
  "llm_model_id": "meta-llama-3-8B",
  "case_id": "case_1452",
  "messages": [
    "Понимаю, сейчас попробую помочь вам с доступом.",
    "Проверьте, пожалуйста, еще раз. Ошибка должна исчезнуть."
  ]
}
```

Пример ответа:

```
{
  "result": {
    "tone": "Дружелюбный",
    "confidence": 0.94
  }
}
```

8.7 History API

Раздел **History** предназначен для работы с историей сообщений между пользователем и LLM. Методы позволяют синхронизировать сообщения и получать информацию о токенах и стоимости запросов.

8.7.1 Синхронизация истории сообщений

- **Эндпоинт:** /history/sync-message-history
- **Метод:** POST
- **Авторизация:** обязательна (HTTP Bearer)
- **Описание:** Синхронизирует историю сообщений между клиентом и сервером — возвращает список событий/записей истории, начиная с указанного идентификатора синхронизации.

Параметры запроса:

Параметр	Тип	Обязательно	Описание
last_sync_id	string	Нет	Идентификатор последней синхронизированной записи (пустая строка — синхронизировать с начала)

Пример запроса:


```
{
  "last_sync_id": ""
}
```

Пример ответа:

```
[
  {
    "id": "string",
    "date": "2019-08-24T14:15:22Z",
    "llm_model_name": "string",
    "llm_input_tokens": 0,
    "llm_input_price": 0,
    "llm_output_tokens": 0,
    "llm_output_price": 0,
    "embedding_tokens": 0,
    "embedding_price": 0
  }
]
```

Коды ответов:

Код	Описание
200	Успешно обработано
422	Ошибка валидации

8.8 Config API

Раздел **Config** предоставляет методы для получения информации о моделях, провайдерах, соединениях и типах моделей.

8.8.1 Получить все типы моделей

- **Эндпоинт:** /config/model-type
- **Метод:** GET
- **Авторизация:** обязательна (HTTP Bearer)
- **Описание:** Возвращает список всех типов моделей.

Пример ответа:

```
[
  {
    "id": "2b167be8-c92c-4a29-b3e4-f86b067ecd73",
    "name": "Embedding",
    "created_on": "2025-08-04T10:19:52.236000",
    "modified_on": "2025-08-04T10:19:52.236000",
    "created_by": "chatbot",
    "modified_by": "chatbot",
    "icon": "data:image/svg+xml;base64,"
  },
  {
    "id": "5811d7b0-c185-4a72-a7d2-4ac623c37684",
```

```

    "name": "LLM",
    "created_on": "2025-08-04T10:20:39.308000",
    "modified_on": "2025-08-04T14:17:13.322000",
    "created_by": "chatbot",
    "modified_by": "chatbot",
    "icon": "data:image/svg+xml;base64 "
  }
]

```

Коды ответов:

Код	Описание
200	Успешно обработано

8.8.2 Получить тип модели по ID

- **Эндпоинт:** /config/model-type/{model_type_id}
- **Метод:** GET
- **Авторизация:** обязательна
- **Описание:** Возвращает информацию о конкретном типе модели.

Параметры запроса:

Параметр	Тип	Обязательно	Описание
model_type_id	string	Да	Идентификатор типа модели

Пример ответа:

```

{
  "id": "2b167be8-c92c-4a29-b3e4-f86b067ecd73",
  "name": "Embedding",
  "created_on": "2025-08-04T10:19:52.236000",
  "modified_on": "2025-08-04T10:19:52.236000",
  "created_by": "chatbot",
  "modified_by": "chatbot",
  "icon": ""
}

```

Коды ответов:

Код	Описание
200	Успешно обработано
422	Ошибка валидации

8.8.3 Получить всех провайдеров

- **Эндпоинт:** /config/provider
- **Метод:** GET
- **Авторизация:** обязательна
- **Описание:** Возвращает список всех провайдеров.

Пример ответа:

```
[
  {
    "id": "7e21f4ce-0904-4ad8-ab9d-f4e5edb96b64",
    "name": "Yandex.Cloud",
    "created_on": "2025-08-04T10:22:51.985000",
    "modified_on": "2025-09-26T10:40:00",
    "created_by": "chatbot",
    "modified_by": "service-account-bpmsoft",
    "icon": "data:image/svg+xml;base64, ",
    "connection_config": {
      "api_key": {
        "type": "secure_string",
        "caption": "API ключ",
        "is_required": true
      },
      "catalog_id": {
        "type": "string",
        "caption": "Каталог",
        "is_required": true
      }
    }
  },
  {
    "id": "c98ae711-3f46-4531-b89c-ad24ec4ac8ec",
    "name": "OpenAI совместимый",
    "created_on": "2025-08-04T10:23:15.698000",
    "modified_on": "2025-09-25T10:30:00",
    "created_by": "chatbot",
    "modified_by": "chatbot",
    "icon": "data:image/svg+xml;base64 ",
    "connection_config": {
      "base_url": {
        "type": "url",
        "caption": "Базовый URL",
        "is_required": true
      },
      "api_key": {
        "type": "secure_string",
        "caption": "API ключ",
        "is_required": true
      }
    }
  }
]
```

Коды ответов:

Код	Описание
200	Успешно обработано
422	Ошибка валидации

8.8.4 Получить провайдера по ID

- **Эндпоинт:** /config/provider/{provider_id}
- **Метод:** GET
- **Авторизация:** обязательна
- **Описание:** Возвращает информацию о конкретном провайдере.

Параметры запроса:

Параметр	Тип	Обязательно	Описание
provider_id	string	Да	Идентификатор провайдера

Пример ответа:

```
{
  "id": "ebd50c4e-2c95-48eb-b2f6-caef061558d7",
  "name": "SberDevices",
  "created_on": "2025-08-04T10:23:49.672000",
  "modified_on": "2025-09-25T10:30:00",
  "created_by": "chatbot",
  "modified_by": "chatbot",
  "icon": "data:image/svg+xml;base64",
  "connection_config": {
    "api_version": {
      "type": "string",
      "caption": "Scope",
      "is_required": true
    },
    "api_key": {
      "type": "secure_string",
      "caption": "API ключ",
      "is_required": true
    }
  }
}
```

Коды ответов:

Код	Описание
200	Успешно обработано
422	Ошибка валидации

8.8.5 Получить все соединения

- **Эндпоинт:** /config/connection
- **Метод:** GET
- **Авторизация:** обязательна
- **Описание:** Возвращает список всех соединений.

Пример ответа:

```
[
  {
    "id": "18b2922a-1c1c-4b8b-922e-bfc833a525bf",
    "name": "SberDevices connection",
    "created_on": "2025-08-04T10:31:38.960000",
    "modified_on": "2025-09-29T11:00:00",
    "created_by": "chatbot",
    "modified_by": "admin-ui",
    "provider_id": "ebd50c4e-2c95-48eb-b2f6-caef061558d7",
    "config": {
      "api_version": "GIGACHAT_API_CORP"
    },
    "connection_config": {
      "api_version": {
        "type": "string",
        "caption": "Scope",
        "is_required": true
      },
      "api_key": {
        "type": "secure_string",
        "caption": "API ключ",
        "is_required": true
      }
    }
  },
  {
    "id": "7cd44794-1c58-40e2-8d64-aabe6d9b5f37",
    "name": "SelfHosted connection",
    "created_on": "2025-08-04T10:32:48.194000",
    "modified_on": "2025-09-29T11:00:00",
    "created_by": "chatbot",
    "modified_by": "chatbot",
    "provider_id": "fb25a5a5-7f7c-4009-a400-d5f60e56d9ae",
    "config": {
      "base_url": "http://localhost:8000",
      "client_id": ""
    },
    "connection_config": {
      "base_url": {
        "type": "url",
        "caption": "Базовый URL",
        "is_required": true
      },
      "client_id": {
        "type": "string",
        "caption": "Client ID",

```

```

        "is_required": true
    },
    "client_secret": {
        "type": "secure_string",
        "caption": "Client Secret",
        "is_required": true
    }
}
]

```

Коды ответов:

Код	Описание
200	Успешно обработано

8.8.6 Получить соединение по ID

- **Эндпоинт:** /config/connection/{connection_id}
- **Метод:** GET
- **Авторизация:** обязательна
- **Описание:** Возвращает информацию о конкретном соединении.

Параметры запроса:

Параметр	Тип	Обязательно	Описание
connection_id	string	Да	Идентификатор соединения

Пример ответа:

```

{
  "id": "18b2922a-1c1c-4b8b-922e-bfc833a525bf",
  "name": "SberDevices connection",
  "created_on": "2025-08-04T10:31:38.960000",
  "modified_on": "2025-09-29T11:00:00",
  "created_by": "chatbot",
  "modified_by": "admin-ui",
  "provider_id": "ebd50c4e-2c95-48eb-b2f6-caef061558d7",
  "config": {
    "api_version": "GIGACHAT_API_CORP"
  },
  "connection_config": {
    "api_version": {
      "type": "string",
      "caption": "Scope",
      "is_required": true
    },
    "api_key": {
      "type": "secure_string",
      "caption": "API ключ",
      "is_required": true
    }
  }
}

```

Коды ответов:

Код	Описание
200	Успешно обработано
422	Ошибка валидации

8.8.7 Получить все модели

- **Эндпоинт:** /config/model
- **Метод:** GET
- **Авторизация:** обязательна
- **Описание:** Возвращает список всех моделей.

Пример ответа:

```
[
  {
    "id": "f4725b4c-d912-4629-8b33-3753f85a0f7b",
    "name": "Gemini25Pro",
    "created_on": "2025-08-04T10:42:56.800000",
    "modified_on": "2025-08-04T10:42:56.800000",
    "created_by": "chatbot",
    "modified_by": "chatbot",
    "api_model_name": "just-ai/gemini/gemini-2.5-pro",
    "is_multimodal": true,
    "model_type_id": "5811d7b0-c185-4a72-a7d2-4ac623c37684",
    "connection_id": "e5236950-082d-49e7-a8d5-c612f8afdbe3",
    "cost_1000_incoming_tokens": 0.413,
    "cost_1000_outgoing_tokens": 2.475
  },
  {
    "id": "c4520b3b-e304-4421-a5c8-b9542f4c4e38",
    "name": "Llama8b",
    "created_on": "2025-08-04T10:45:04.816000",
    "modified_on": "2025-08-04T10:45:04.816000",
    "created_by": "chatbot",
    "modified_by": "chatbot",
    "api_model_name": "llama-lite",
    "is_multimodal": false,
    "model_type_id": "5811d7b0-c185-4a72-a7d2-4ac623c37684",
    "connection_id": "8fb477bc-0e61-4a24-8db9-579b9296003e",
    "cost_1000_incoming_tokens": 0.2,
    "cost_1000_outgoing_tokens": 0.2
  }
]
```

Коды ответов:

Код	Описание
200	Успешно обработано

8.8.8 Получить модель по ID

- **Эндпоинт:** /config/model/{model_id}
- **Метод:** GET
- **Авторизация:** обязательна
- **Описание:** Возвращает информацию о конкретной модели.

Параметры запроса:

Параметр	Тип	Обязательно	Описание
model_id	string	Да	Идентификатор модели

Пример ответа:

```
[
  {
    "id": "f4725b4c-d912-4629-8b33-3753f85a0f7b",
    "name": "Gemini25Pro",
    "created_on": "2025-08-04T10:42:56.800000",
    "modified_on": "2025-08-04T10:42:56.800000",
    "created_by": "chatbot",
    "modified_by": "chatbot",
    "api_model_name": "just-ai/gemini/gemini-2.5-pro",
    "is_multimodal": true,
    "model_type_id": "5811d7b0-c185-4a72-a7d2-4ac623c37684",
    "connection_id": "e5236950-082d-49e7-a8d5-c612f8afdbe3",
    "cost_1000_incoming_tokens": 0.413,
    "cost_1000_outgoing_tokens": 2.475
  }
]
```

Коды ответов:

Код	Описание
200	Успешно обработано
422	Ошибка валидации

8.8.9 Подписка на события по каналу

- **Эндпоинт:** /events/{channel}
- **Метод:** GET

- **Авторизация:** не требуется
- **Описание:** Позволяет получать события в реальном времени для указанного канала.

Параметры запроса:

Параметр	Тип	Обязательно	Описание
channel	string	Да	Название канала для подписки

Пример ответа:

```
{
  "type": "card_update",
  "data_cortex_id": "[data_cortex_setting_id]"
}
{"type": "indexing_error", "error_message": "[escaped_error_message]",
"data_cortex_id": "[data_cortex_setting_id ]"}
```

Коды ответов:

Код	Описание
200	Успешно обработано
422	Ошибка валидации

8.9 Data Cortex API

Раздел **Data Cortex** включает методы для получения и управления настройками областей знаний, документами, индексированием и источниками знаний.

8.9.1 Получение настроек Data Cortex по идентификатору

- **Эндпоинт:** /data-cortex/data-cortex-setting/{data_cortex_id}
- **Метод:** GET
- **Авторизация:** обязательна
- **Описание:** Возвращает настройки Data Cortex по идентификатору.

Параметры запроса:

Параметр	Тип	Обязательно	Описание
data_cortex_id	string	Да	Идентификатор Data Cortex

Пример ответа:

```
{
  "result": {
    "source_id": "string",
```

```
"max_segment_length": 0,
"segment_overlap": 0,
"search_context_terms": "string",
"embedding_model_id": "string",
"id": "",
"name": "string",
"indexing_status_id": "string",
"last_indexing_date": "2019-08-24T14:15:22Z",
"indexing_error": "string"
},
"error_message": "",
"is_success": true
}
```

Коды ответов:

Код	Описание
200	Успешно обработано
422	Ошибка валидации

8.9.2 Удаление настроек Data Cortex

- **Эндпоинт:** /data-cortex/data-cortex-setting/{data_cortex_id}
- **Метод:** DELETE
- **Авторизация:** обязательна
- **Описание:** Удаляет настройки Data Cortex по идентификатору.

Пример ответа:

```
{
  "result": true,
  "error_message": "",
  "is_success": true
}
```

Коды ответов:

Код	Описание
200	Успешно обработано
422	Ошибка валидации

8.9.3 Получение всех настроек Data Cortex

- **Эндпоинт:** /data-cortex/data-cortex-setting
- **Метод:** GET

- **Авторизация:** обязательна (HTTP Bearer)
- **Описание:** Возвращает список всех настроек Data Cortex.

Пример ответа:

```
{
  "result": [
    { ... },
    { ... }
  ],
  "error_message": "",
  "is_success": true
}
```

Коды ответов:

Код	Описание
200	Успешно обработано

8.9.4 Обновление Data Cortex

- **Эндпоинт:** /data-cortex/data-cortex-setting
- **Метод:** PUT
- **Авторизация:** обязательна
- **Описание:** Обновляет существующие настройки Data Cortex.

Параметры запроса:

Параметр	Тип	Обязательно	Описание
source_id	string	Да	Идентификатор источника
max_segment_length	integer	Да	Максимальная длина сегмента
segment_overlap	integer	Да	Перекрытие сегментов
search_context_terms	string	Да	Контекст поиска
embedding_model_id	string	Да	Идентификатор модели эмбеддингов
id	string	Да	Идентификатор Data Cortex
name	string	Да	Название области знаний

Пример запроса:

```
{
  "source_id": "string",
  "max_segment_length": 0,
  "segment_overlap": 0,
  "search_context_terms": "string",
}
```

```

    "embedding_model_id": "string",
    "id": "string",
    "name": "string"
  }

```

Пример ответа:

```

{
  "result": true,
  "error_message": "",
  "is_success": true
}

```

Коды ответов:

Код	Описание
200	Успешно обработано
422	Ошибка валидации

8.9.5 Добавление Data Cortex

- **Эндпоинт:** /data-cortex/data-cortex-setting
- **Метод:** POST
- **Авторизация:** обязательна
- **Описание:** Добавляет новую область знаний в Data Cortex.

Параметры запроса:

Параметр	Тип	Обязательно	Описание
source_id	string	Да	Идентификатор источника
max_segment_length	integer	Да	Максимальная длина сегмента
segment_overlap	integer	Да	Перекрытие сегментов
search_context_terms	string	Да	Контекст поиска
embedding_model_id	string	Да	Идентификатор модели эмбеддингов
id	string	Нет	Идентификатор Data Cortex (по умолчанию "")
name	string	Да	Название области знаний
indexing_status_id	string	Да	Статус индексации

Пример запроса:

```

{
  "source_id": "string",
  "max_segment_length": 0,

```

```

"segment_overlap": 0,
"search_context_terms": "string",
"embedding_model_id": "string",
"id": "",
"name": "string",
"indexing_status_id": "string"
}

```

Пример ответа:

```

{
  "result": "string",
  "error_message": "",
  "is_success": true
}

```

Коды ответов:

Код	Описание
200	Успешно обработано
422	Ошибка валидации

8.9.6 Запуск индексации Data Cortex

- **Эндпоинт:** /data-cortex/start-indexing/{data_cortex_id}
- **Метод:** POST
- **Авторизация:** обязательна
- **Описание:** Запускает процесс индексации для указанной области знаний.

Параметры запроса:

Параметр	Тип	Обязательно	Описание
data_cortex_id	string	Да	Идентификатор Data Cortex

Пример ответа:

```

{
  "result": true,
  "error_message": "",
  "is_success": true
}

```

Коды ответов:

Код	Описание
200	Успешно обработано

Код	Описание
422	Ошибка валидации

8.9.7 Получение всех типов источников Data Cortex

- **Эндпоинт:** /data-cortex/data-cortex-source-type
- **Метод:** GET
- **Авторизация:** обязательна (HTTP Bearer)
- **Описание:** Возвращает список всех типов источников Data Cortex.

Пример ответа:

```
{
  "result": [
    { ... },
    { ... }
  ],
  "error_message": "",
  "is_success": true
}
```

Коды ответов:

Код	Описание
200	Успешно обработано

8.9.8 Получение всех статусов индексации

- **Эндпоинт:** /data-cortex/indexing-status
- **Метод:** GET
- **Авторизация:** обязательна (HTTP Bearer)
- **Описание:** Возвращает список всех возможных статусов индексации Data Cortex.

Пример ответа:

```
{
  "result": [
    { ... },
    { ... }
  ],
  "error_message": "",
  "is_success": true
}
```

Коды ответов:

Код	Описание
200	Успешно обработано

8.9.9 Получение всех следующих действий по базе знаний

- **Эндпоинт:** /data-cortex/knowledge-base-next-indexing-action
- **Метод:** GET
- **Авторизация:** обязательна (HTTP Bearer)
- **Описание:** Возвращает список всех следующих действий по индексации базы знаний

Пример ответа:

```
{
  "result": [
    { ... },
    { ... }
  ],
  "error_message": "",
  "is_success": true
}
```

Коды ответов:

Код	Описание
200	Успешно обработано

8.9.10 Получение всех статусов индексации документов

- **Эндпоинт:** /data-cortex/document-indexing-status
- **Метод:** GET
- **Авторизация:** обязательна (HTTP Bearer)
- **Описание:** Возвращает список статусов индексации документов Data Cortex.

Пример ответа:

```
{
  "result": [
    { ... },
    { ... }
  ],
}
```

```
"error_message": "",  
"is_success": true  
}
```

Коды ответов:

Код	Описание
200	Успешно обработано

8.9.11 Получение документа Data Cortex по идентификатору

- **Эндпоинт:** /data-cortex/data-cortex-document/{data_cortex_document_id}
- **Метод:** GET
- **Авторизация:** обязательна
- **Описание:** Возвращает документ Data Cortex по идентификатору.

Параметры запроса:

Параметр	Тип	Обязательно	Описание
data_cortex_document_id	string	Да	Идентификатор документа

Пример ответа:

```
null
```

Коды ответов:

Код	Описание
200	Успешно обработано
422	Ошибка валидации

8.9.12 Удаление документа Data Cortex

- **Эндпоинт:** /data-cortex/data-cortex-document/{data_cortex_document_id}
- **Метод:** DELETE
- **Авторизация:** обязательна
- **Описание:** Удаляет документ Data Cortex по идентификатору.

Пример ответа:

```
{
```



```
"result": true,  
"error_message": "",  
"is_success": true  
}
```

Коды ответов:

Код	Описание
200	Успешно обработано
422	Ошибка валидации

8.9.13 Добавление документа в Data Cortex

- **Эндпоинт:** /data-cortex/data-cortex-document/{data_cortex_id}
- **Метод:** POST
- **Авторизация:** обязательна
- **Описание:** Добавляет документ в указанную область знаний.

Параметры запроса (multipart/form-data):

Параметр	Тип	Обязательно	Описание
file	binary	Да	Файл документа
file_mime_type	string	Да	MIME-тип файла

Пример ответа:

```
{  
  "result": { "id": "string" },  
  "error_message": "",  
  "is_success": true  
}
```

Коды ответов:

Код	Описание
200	Успешно обработано
422	Ошибка валидации

8.9.14 Получение списка документов по Data Cortex

- **Эндпоинт:** /data-cortex/get_file_documents/{data_cortex_id}
- **Метод:** GET

- **Авторизация:** обязательна
- **Описание:** Возвращает список документов для указанной области знаний.

Пример ответа:

```
{
  "result": [
    { ... },
    { ... }
  ],
  "error_message": "",
  "is_success": true
}
```

Коды ответов:

Код	Описание
200	Успешно обработано
422	Ошибка валидации

8.9.15 Получение индексированных статей базы знаний

- **Эндпоинт:**
`/data-cortex/get_indexed_knowledge_base_documents/{data_cortex_id}`
- **Метод:** GET
- **Авторизация:** обязательна
- **Описание:** Возвращает список индексированных статей базы знаний по идентификатору области знаний.

Пример ответа:

```
{
  "result": [
    { ... },
    { ... }
  ],
  "error_message": "",
  "is_success": true
}
```

Коды ответов:

Код	Описание
200	Успешно обработано
422	Ошибка валидации

8.9.16 Получение актуальных статей базы знаний

- **Эндпоинт:** /data-cortex/get_actual_knowledge_base_documents
- **Метод:** POST
- **Авторизация:** обязательна
- **Описание:** Возвращает актуальные статьи базы знаний на основе условий контекста поиска.

Параметры запроса:

Параметр	Тип	Обязательно	Описание
search_context_terms	string	Да	Термины поиска
data_cortex_id	string	Да	Идентификатор Data Cortex

Пример запроса:

```
{
  "search_context_terms": "string",
  "data_cortex_id": "string"
}
```

Пример ответа:

```
{
  "result": [
    { ... },
    { ... }
  ],
  "error_message": "",
  "is_success": true
}
```

Коды ответов:

Код	Описание
200	Успешно обработано
422	Ошибка валидации

8.10 Prompt Templates API

Раздел **Prompt Templates** включает методы для управления и получения информации о промптах, которые используются в различных сценариях LLM.

8.10.1 Получение списка всех промптов

- **Эндпоинт:** /prompt-templates
- **Метод:** GET
- **Авторизация:** обязательна (HTTP Bearer)
- **Описание:** Возвращает список всех промптов.

Пример ответа:

```
[
  {
    "id": "",
    "name": "string",
    "created_on": "2019-08-24T14:15:22Z",
    "modified_on": "2019-08-24T14:15:22Z",
    "created_by": "string",
    "modified_by": "string",
    "prompt": "string"
  }
]
```

Коды ответов:

Код	Описание
200	Успешно найдено
500	Ошибка обработки запроса

8.10.2 Получение промпта по идентификатору

- **Эндпоинт:** /prompt-templates/{prompt_id}
- **Метод:** GET
- **Авторизация:** обязательна (HTTP Bearer)
- **Описание:** Возвращает промпт с переданным идентификатором, если такой существует.

Параметры:

Параметр	Тип	Обязательно	Описание
prompt_id	string	Да	Идентификатор промпта

Пример ответа:

```
{
```

```

    "id": "123e4567-e89b-12d3-a456-426614174000",
    "name": "Example Prompt",
    "created_on": "2019-08-24T14:15:22Z",
    "modified_on": "2019-08-24T14:15:22Z",
    "created_by": "admin",
    "modified_by": "editor",
    "prompt": "Пример текста промпта"
  }

```

Коды ответов:

Код	Описание
200	Промпт найден
404	Промпт не найден
422	Ошибка валидации
500	Ошибка обработки запроса

8.11 LLM Request Raw API

Раздел **LLM Request Raw** включает методы для отправки запроса к LLM с указанием сессии, истории сообщений и системного промпта.

8.11.1 Отправка запроса к LLM

- **Эндпоинт:** `/llm-request-raw/request-llm`
- **Метод:** `POST`
- **Авторизация:** обязательна (HTTP Bearer)
- **Описание:** Возвращает ответ LLM с указанием токенов, стоимости, истории сообщений и других деталей.

Параметры запроса:

Параметр	Тип	Обязательно	Описание
<code>session_id</code>	string	Да	Уникальный идентификатор сессии
<code>data_cortex_id</code>	string	Да	Идентификатор данных корпорации
<code>initial_prompt</code>	string	Да	Системный (установочный) промпт для LLM
<code>llm_id</code>	string	Да	Идентификатор используемой LLM модели
<code>message</code>	string	Да	Сообщение (запрос) пользователя
<code>history</code>	array	Нет	История сообщений диалога

Пример запроса:

```
{
  "session_id": "94237eec-8c7c-4806-b4e8-50f3e22bfb59",
  "data_cortex_id": "8d97b6da-ed46-4698-a03b-4d2052aa3b97",
  "initial_prompt": "Hello, how can I assist you today? {context}",
  "llm_id": "c4246274-41df-4cca-b961-f2220c37c52c",
  "message": "I'm looking for help with creating an AI-based service",
  "history": [
    { "user_message": "Hi", "llm_message": "Hello!" },
    { "user_message": "I need guidance", "llm_message": "Sure, let's start" }
  ]
}
```

Пример ответа:

```
{
  "session_id": "94237eec-8c7c-4806-b4e8-50f3e22bfb59",
  "date": "2025-10-22T12:34:56Z",
  "user_message": "I'm looking for help with creating an AI-based service",
  "prompt": "Hello, how can I assist you today? {context}",
  "llm_message": "Вот подробный ответ модели...",
  "llm_model_name": "Meta LLaMA 3 8B",
  "llm_model_id": "c4246274-41df-4cca-b961-f2220c37c52c",
  "llm_input_tokens": 150,
  "llm_input_price": 0.003,
  "llm_output_tokens": 300,
  "llm_output_price": 0.006,
  "embedding_model_name": "Embedding Model X",
  "embedding_model_id": "embed_01",
  "embedding_tokens": 50,
  "embedding_price": 0.001,
  "embeddings": [
    { "vector": [0.12, 0.34, 0.56], "metadata": "..." }
  ],
  "relative_sources": [
    "Knowledge Base Article 1",
    "Knowledge Base Article 2"
  ]
}
```

Коды ответов:

Код	Описание
200	Успешно обработано
422	Ошибка валидации
500	Ошибка обработки запроса

8.12 Periods API

Раздел **Periods** включает методы для получения данных о периодах с указанием идентификатора или списка всех существующих периодов.

8.12.1 Получение периода по идентификатору

– **Эндпоинт:** `/periods/{period_id}`

- **Метод:** GET
- **Авторизация:** обязательна (HTTP Bearer)
- **Описание:** Возвращает период с переданным идентификатором, если такой существует.

Параметры запроса:

Параметр	Тип	Обязательно	Описание
period_id	string	Да	Идентификатор периода

Пример ответа:

```
{
  "id": "123e4567-e89b-12d3-a456-426614174000",
  "name": "Q1 2025",
  "created_on": "2019-08-24T14:15:22Z",
  "modified_on": "2019-08-24T14:15:22Z",
  "created_by": "admin",
  "modified_by": "editor"
}
```

Коды ответов:

Код	Описание
200	Период найден
404	Период не найден
422	Ошибка валидации
500	Ошибка обработки запроса

8.12.2 Получение списка всех периодов

- **Эндпоинт:** /periods
- **Метод:** GET
- **Авторизация:** обязательна (HTTP Bearer)
- **Описание:** Возвращает список всех периодов.

Пример ответа:

```
[
  {
    "id": "123e4567-e89b-12d3-a456-426614174000",
    "name": "Q1 2025",
    "created_on": "2019-08-24T14:15:22Z",
    "modified_on": "2019-08-24T14:15:22Z",
  }
]
```

```

    "created_by": "admin",
    "modified_by": "editor"
  },
  {
    "id": "223e4567-e89b-12d3-a456-426614174111",
    "name": "Q2 2025",
    "created_on": "2019-11-24T14:15:22Z",
    "modified_on": "2019-11-25T10:05:12Z",
    "created_by": "admin",
    "modified_by": "editor"
  }
]

```

Коды ответов:

Код	Описание
200	Успешно найдено
500	Ошибка обработки запроса

8.13 Time Ranges API

Раздел **Time Ranges** включает методы для получения данных о временных диапазонах с указанием идентификатора или списка всех существующих диапазонов.

8.13.1 Получение временного диапазона по идентификатору

- **Эндпоинт:** `/time-ranges/{time_range_id}`
- **Метод:** GET
- **Авторизация:** обязательна (HTTP Bearer)
- **Описание:** Возвращает временной диапазон с переданным идентификатором, если такой существует.

Параметры запроса:

Параметр	Тип	Обязательно	Описание
<code>time_range_id</code>	string	Да	Идентификатор временного диапазона

Пример ответа:

```

{
  "id": "123e4567-e89b-12d3-a456-426614174000",
  "name": "Morning Shift",
  "created_on": "2019-08-24T14:15:22Z",
  "modified_on": "2019-08-24T14:15:22Z",
  "created_by": "admin",
  "modified_by": "editor"
}

```


Коды ответов:

Код	Описание
200	Временной диапазон найден
404	Временной диапазон не найден
422	Ошибка валидации
500	Ошибка обработки запроса

8.13.2 Получение списка всех временных диапазонов

- **Эндпоинт:** /time-ranges
- **Метод:** GET
- **Авторизация:** обязательна (HTTP Bearer)
- **Описание:** Возвращает список всех временных диапазонов.

Пример ответа:

```
[
  {
    "id": "123e4567-e89b-12d3-a456-426614174000",
    "name": "Morning Shift",
    "created_on": "2019-08-24T14:15:22Z",
    "modified_on": "2019-08-24T14:15:22Z",
    "created_by": "admin",
    "modified_by": "editor"
  },
  {
    "id": "223e4567-e89b-12d3-a456-426614174111",
    "name": "Evening Shift",
    "created_on": "2019-08-24T15:00:00Z",
    "modified_on": "2019-08-24T16:00:00Z",
    "created_by": "admin",
    "modified_by": "editor"
  }
]
```

Коды ответов:

Код	Описание
200	Успешно найдено
500	Ошибка обработки запроса

8.14 Dashboard API

Раздел **Dashboard** включает метод для получения информации о токенах, стоимости и использовании моделей и эмбедингов.

8.14.1 Получение данных Dashboard

- **Эндпоинт:** /dashboard
- **Метод:** GET
- **Авторизация:** обязательна (HTTP Bearer)
- **Описание:** Возвращает статистику использования моделей и эмбеддингов, включая токены и стоимость.

Пример ответа:

```
{
  "model_token_usage": [
    { "model_name": "Meta LLaMA 3 8B", "tokens_used": 1500 },
    { "model_name": "GPT-4", "tokens_used": 3200 }
  ],
  "model_money_usage": [
    { "model_name": "Meta LLaMA 3 8B", "cost": 0.45 },
    { "model_name": "GPT-4", "cost": 1.28 }
  ],
  "embedding_token_usage": [
    { "embedding_model_name": "Embedding Model X", "tokens_used": 500 }
  ]
}
```

Коды ответов:

Код	Описание
200	Успешно обработано
422	Ошибка валидации
500	Ошибка обработки запроса

9. НАСТРОЙКА КОННЕКТОРОВ С ВНЕШНИМИ СИСТЕМАМИ

9.1 Настройка коннектора с ITSM box

Для настройки коннектора к ITSM box (используется при наполнении областей знаний) необходимо дополнить .env файл для бэка AI box значениями переменных окружения (Таблица 1). После чего переустановить контейнер.

Таблица 1. Переменные среды

Название	Описание	Пример
KEYCLOAK_CLIENT_ID	Client Id клиента из KeyCloak (AIbox->ITSMbox)	AIbox-ITSMbox
KEYCLOAK_CLIENT_SECRET	Client secret клиента из KeyCloak (AIbox->ITSMbox)	AIbox->ITSMbox

9.2 Настройка коннектора к внешней системе мониторинга на примере подключения к Prometheus в Grafana

До создания нового подключения требуется определить URL будущего подключения. Для Docker: `https://{SERVER_HOST}:{PROMETHEUS_PORT}` (заполнить значениям указанных переменных).

После потребуется создать новое подключение через административную Grafana. Последовательность действий:

1. Открыть административную консоль Grafana.
2. Перейти в раздел **Connections - Add new connection**.
3. В блоке **Data sources** выбрать требуемый источник данных (в данном примере – **Prometheus**). Откроется карточка источника данных.
4. Нажать **Add new data source**.
5. Заполнить параметры подключения:
 - **Connection**. Prometheus server URL: указать URL подключения, определенного ранее.
 - **Authentication**. В поле «Authentication method» Выбрать Basic authentication. Заполнить значениями:

- User: admin,
- Password: пароль в незашифрованном формате.

6. Нажать **Save & test** в конце страницы.

После сохранения подключение станет доступным для мониторинга.

9.2.1 Импорт дашбордов Prometheus

Для мониторинга развёрнутых контейнеров можно использовать следующие дашборды:

- Docker containers resources (дашборд потребления ресурсов развёрнутыми контейнерами);
- Services availability (дашборд мониторинга доступности развёрнутых сервисов);
- VM resources (дашборд потребления ресурсов VM, на которой развёрнуты контейнеры).

После настройки подключения к Prometheus и выбора соответствующих сборке дашбордов можно начать импорт. Последовательность шагов:

1. Перейти в **dashboards**. Нажать на кнопку **New - New dashboard**.
2. Нажать на кнопку **Import dashboard** и далее с помощью **Upload dashboard JSON file** загрузить необходимый json-Файл с дашбордом.
3. В поле **Prometheus data source** выбрать ранее созданное подключение к Prometheus.