# Документация, содержащая информацию, необходимую для установки программного обеспечения Smart Platform

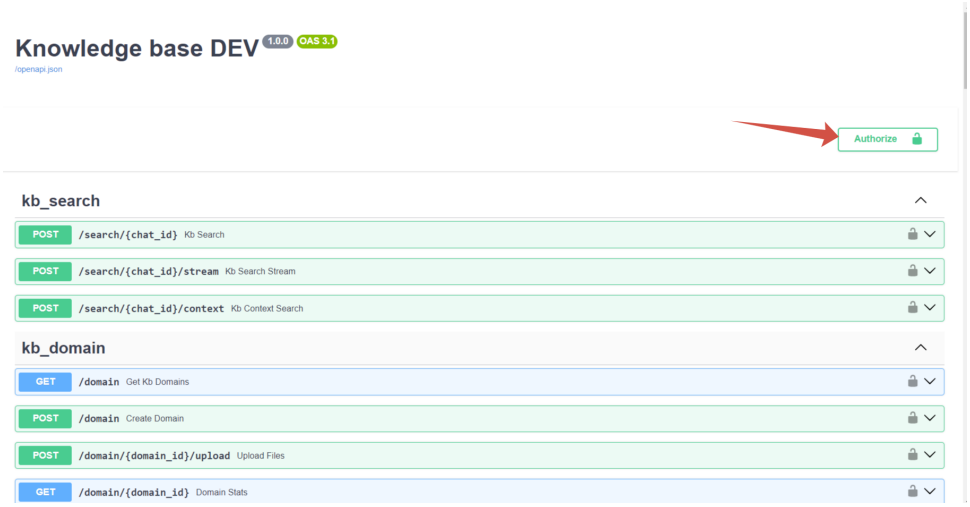
**ООО «Ресерч дата лаб»,**
**Москва, 2024**

# Авторизация и получение токена

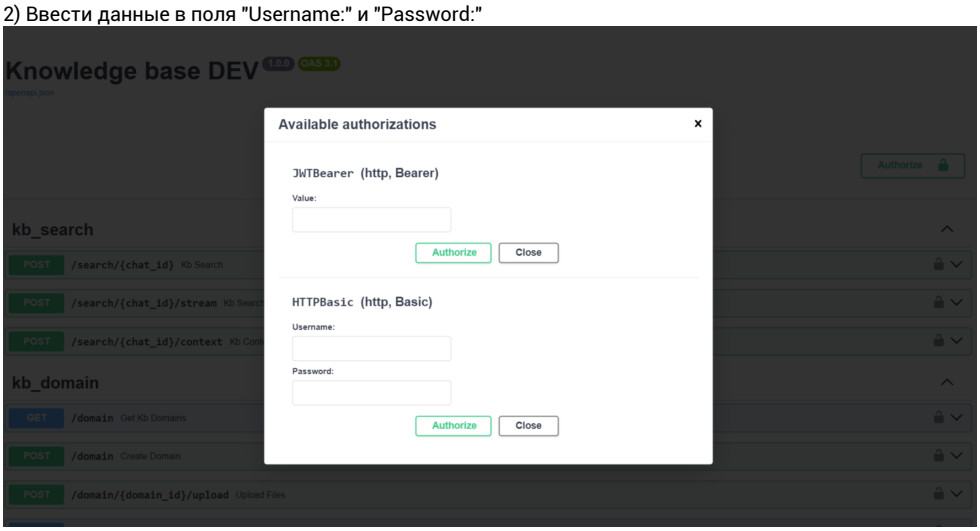
Для авторизации в API вам нужно использовать логин и пароль, которые были получены вами перед началом работы с API.

Эти логин и пароль вам нужно указать также, как это указано на скриншотах:

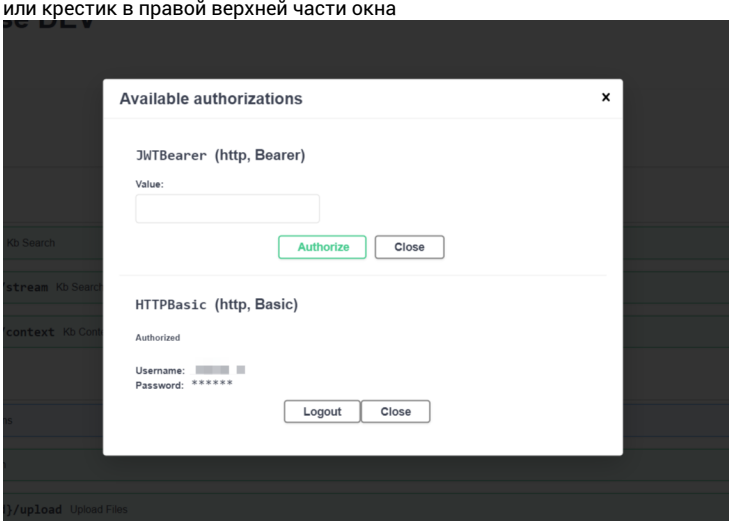
1) Нажимаем кнопку "Authorize"




2) Ввести данные в поля "Username:" и "Password:"

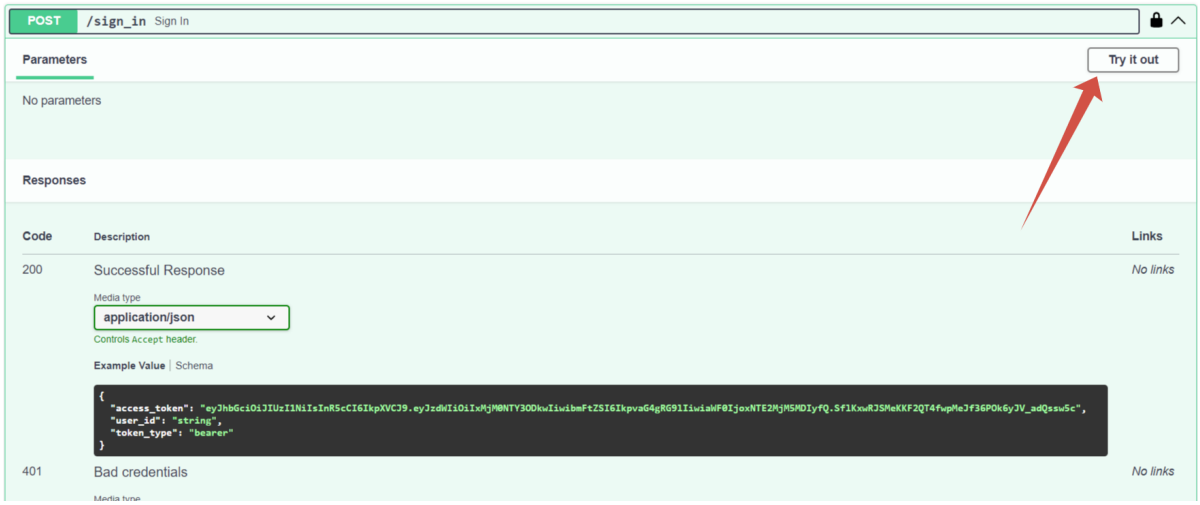


3) Нажать на кнопку Authorize находящуюся чуть ниже поля "Password:", чтобы получилось как на скриншоте, после чего можно нажать "Close" или крестик в правой верхней части окна

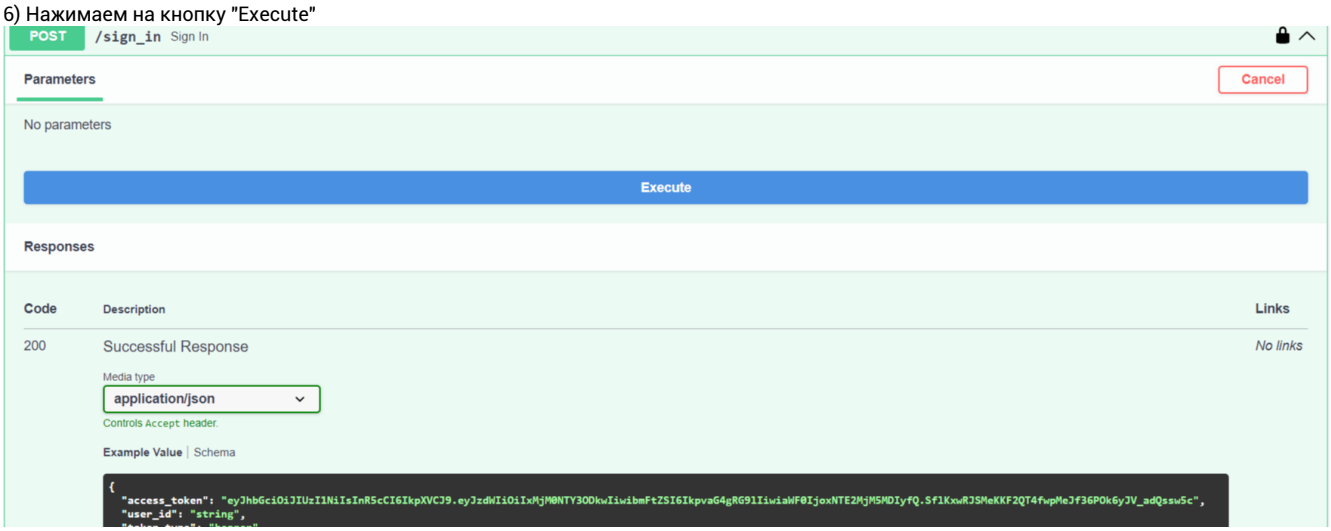



4) Листаем до пункта User и находим метод Sign in

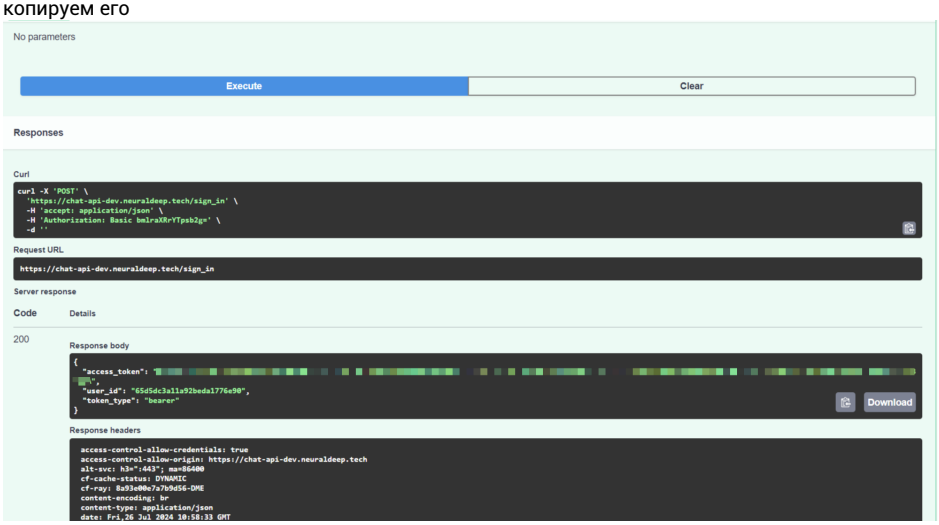

5) Открываем данный метод и нажимаем кнопку "Try it out"



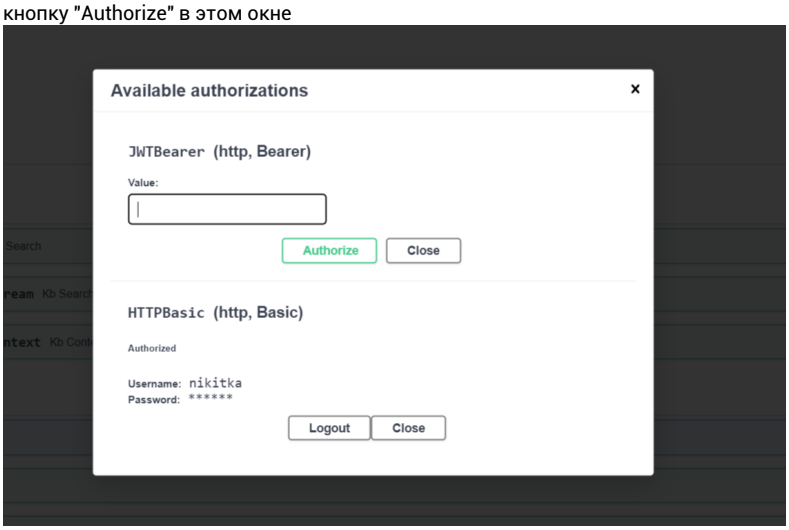
6) Нажимаем на кнопку "Execute"



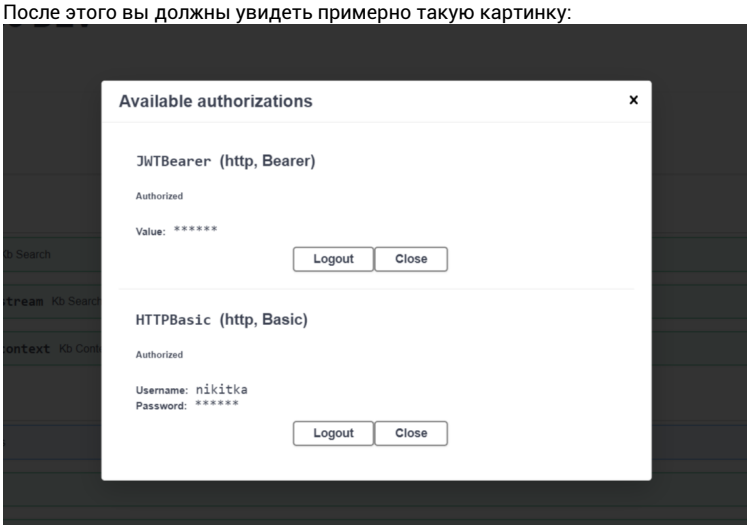

7) Пролистываем метод чуть ниже и в окне "Response body", находим поле "access_token" и в нём находится наш токен для авторизации, копируем его




8) Возвращаемся в начало и вновь нажимаем кнопку "Authorize", вводим скопированный токен авторизации в поле "Value" и нажимаем кнопку "Authorize" в этом окне



После этого вы должны увидеть примерно такую картинку:



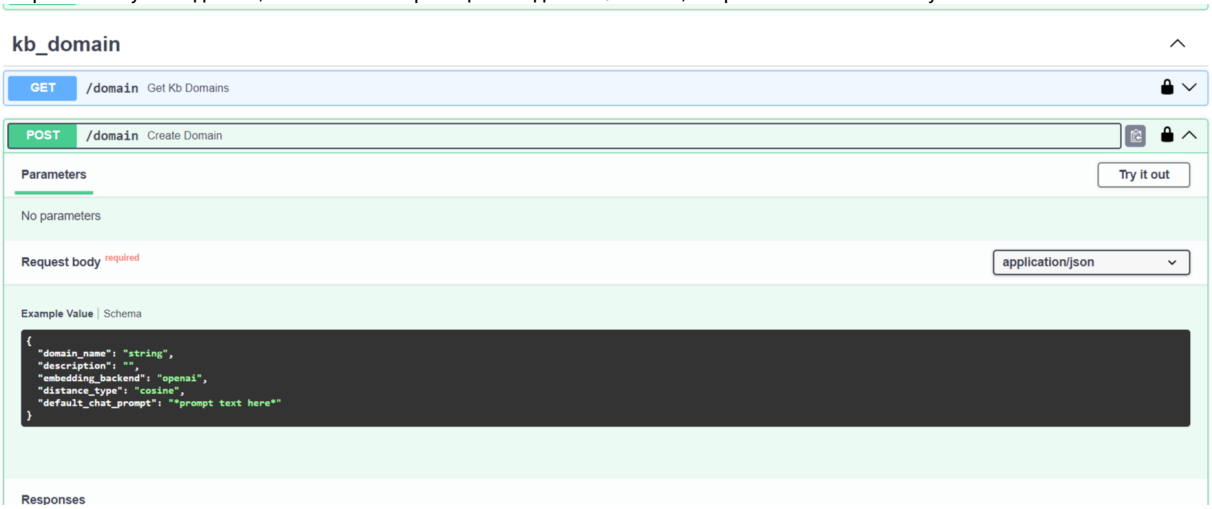

Это значит, что вы внесли все нужные для авторизации данные и теперь можете использовать API

# Как создать домен и добавить на него данные

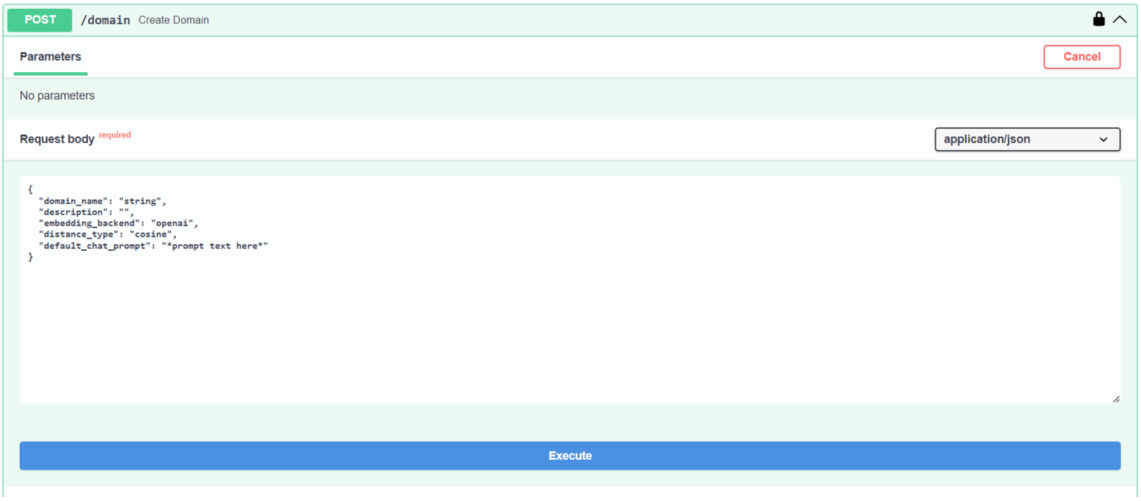
Для создания домена, вам нужно авторизоваться, используя аккаунт, на котором вы хотите создать домен.

После авторизации следует действовать по этим шагам

1) Первое что нужно сделать, это найти на странице метод POST /domain, открыть его и нажать "Try it out"

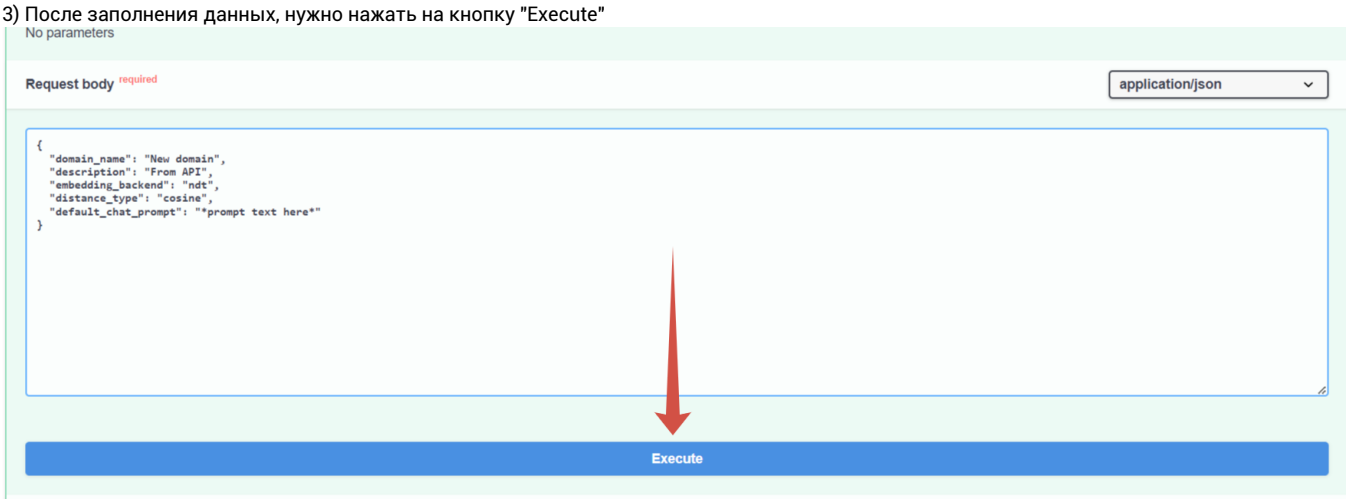



2) Следующий шаг, заполнить данные в текстовом поле

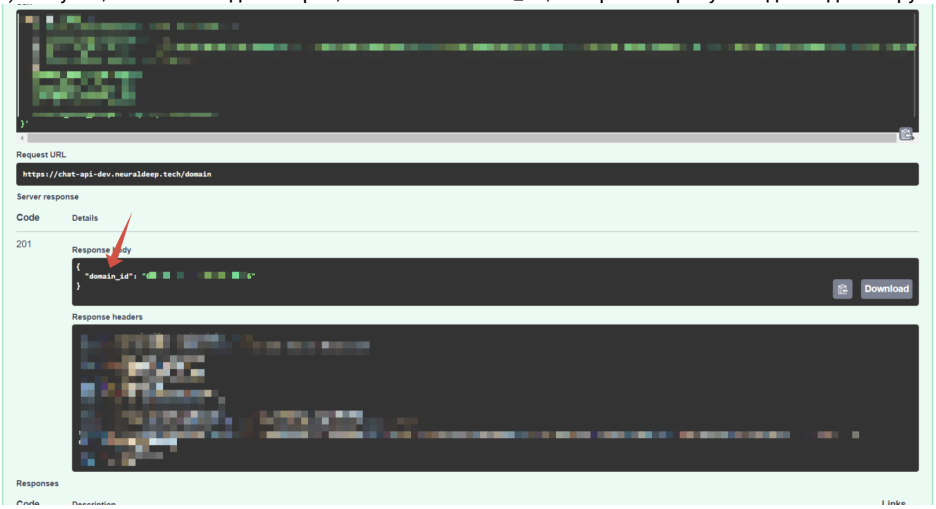



Подробнее про эти поля:
{
"domain_name": "string", - Название вашего нового домена
"description": "", - Описание для нового домена (необязательно)
"embedding_backend": "ndt", - Выбор системы эмбедингов, рекомендуется использовать значение ndt
"distance_type": "cosine", - Выбор системы векторного поиска, рекомендуется использовать значение cosine
"default_chat_prompt": "*prompt text here*" - Базовый промт для домена (необязательно)
}

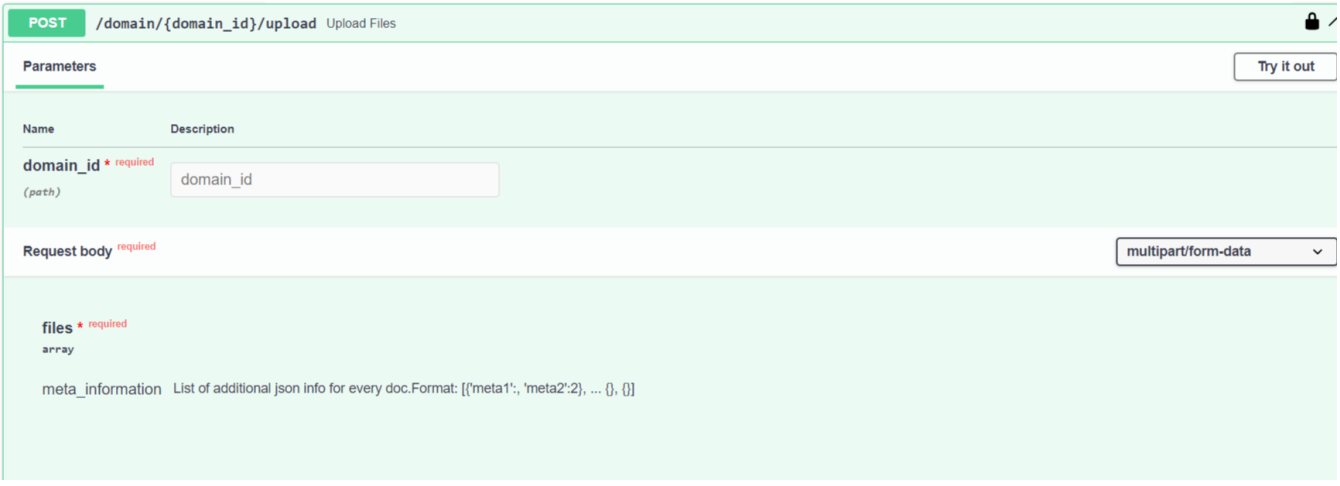
3) После заполнения данных, нужно нажать на кнопку "Execute"



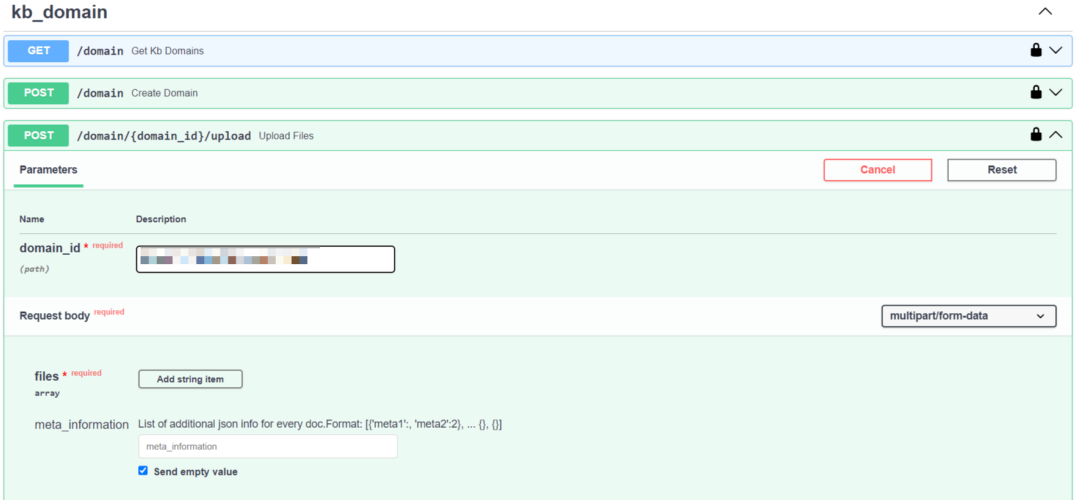

4) В случае, если всё создано верно, появится "domain_id", который потребуется далее для загрузки данных



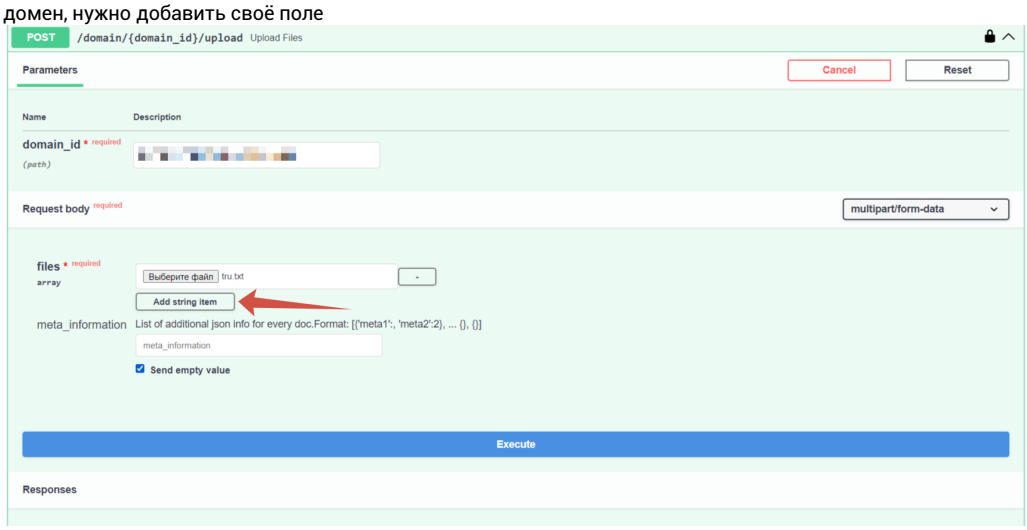

5) После этого, сохранив себе "domain_id" можно скрыть данный метод и открыть метод POST /domain{domain_id}/upload необходимый для загрузки данных в домен, нажимаем "Try it out"



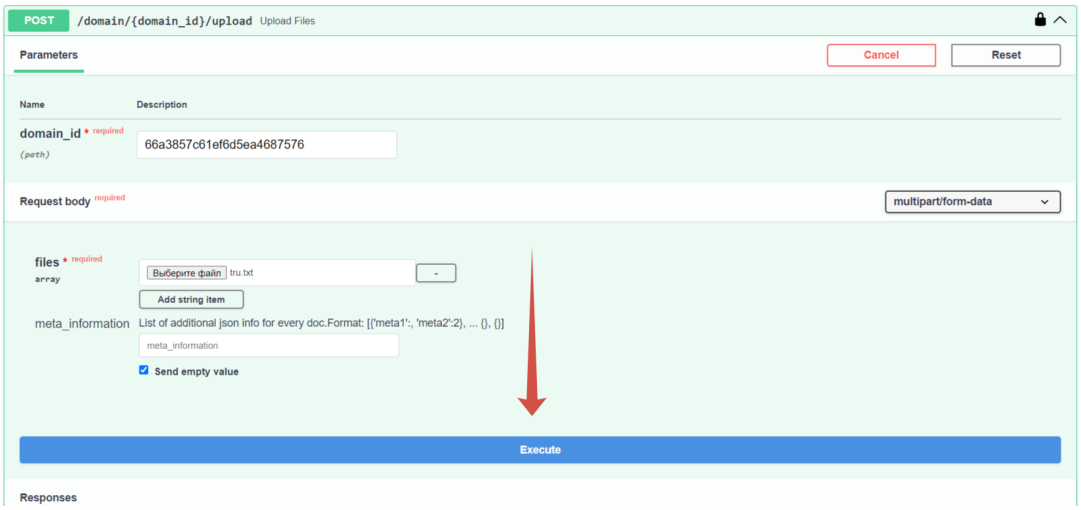
6) Вводим "domain_id" из прошлого метода в поле "domain_id"



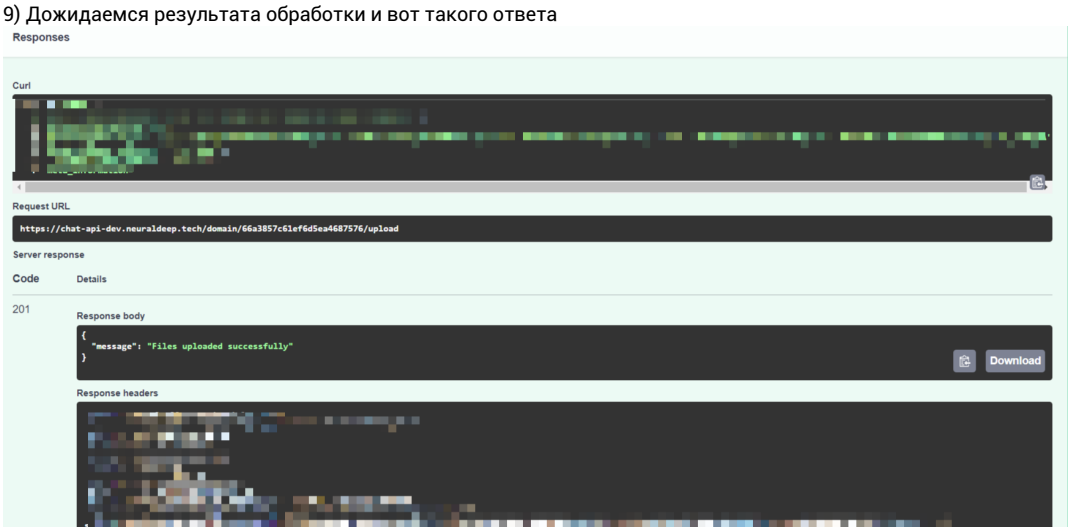

7) Используя кнопку Add string item в строке files добавляем поле для вноса документов, на каждый документ, который нужно добавить в домен, нужно добавить своё поле




8) После добавления всех документов, нажимаем "Execute"



9) Дожидаемся результата обработки и вот такого ответа



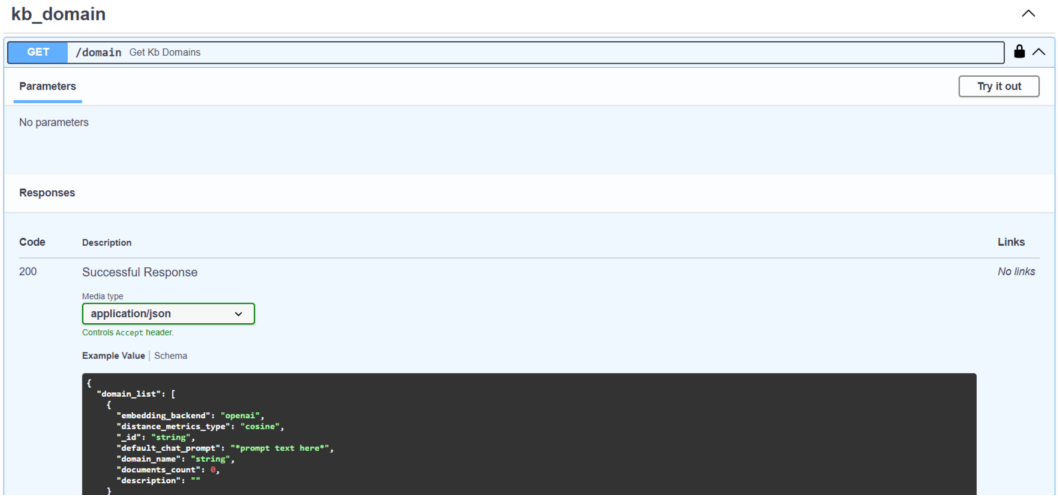

Всё, домен успешно создан и данные в него загружены.

# Как узнать domain_id/Как создать чат

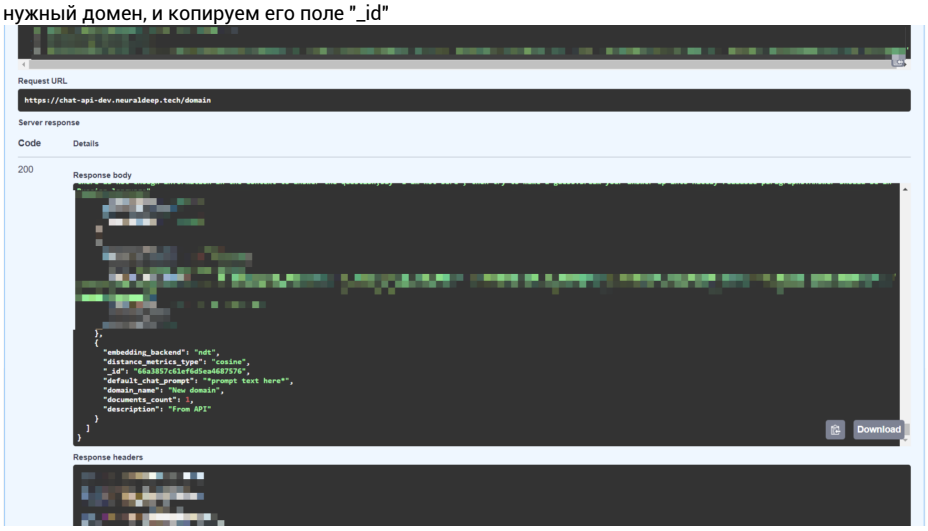
Для создания чата, вам нужно авторизоваться, используя аккаунт, на котором вы хотите создать чат, а также получить domain_id необходимого домена, к которому будет добавлен чат. Если domain_id уже вам известен, переходите к 3 шагу.

После авторизации следует действовать по этим шагам:

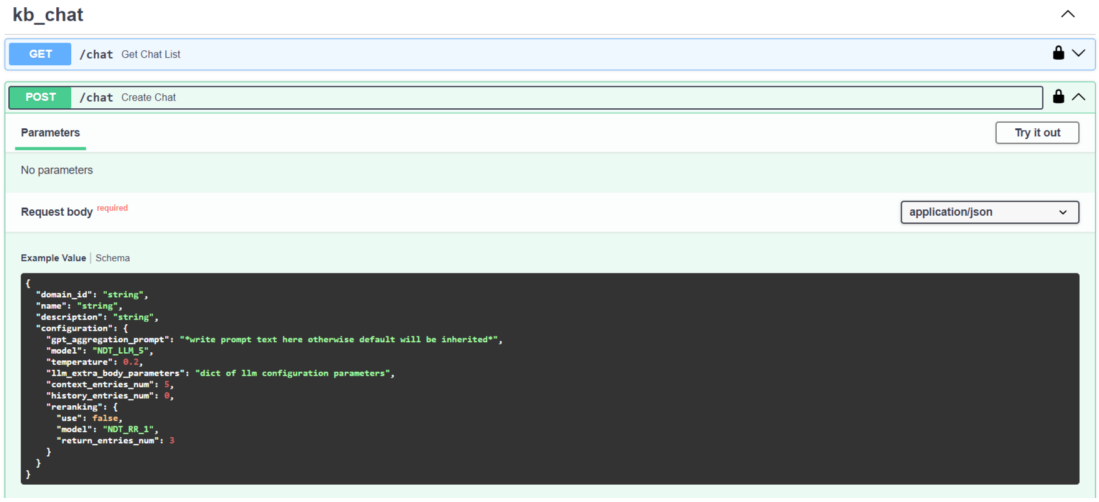
1) Для получения domain_id лучше всего использовать метод GET /domain



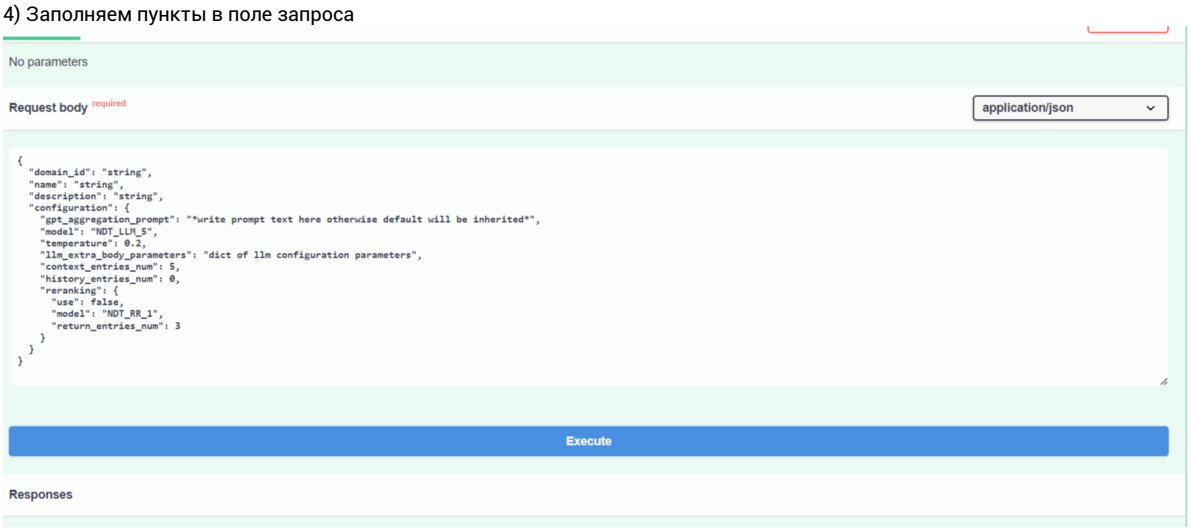

2) После нажатия "Try it out" нажимаем "Execute" и пролистываем вниз до окна ответа от API, с списком доступных доменов, где находим нужный домен, и копируем его поле "_id"



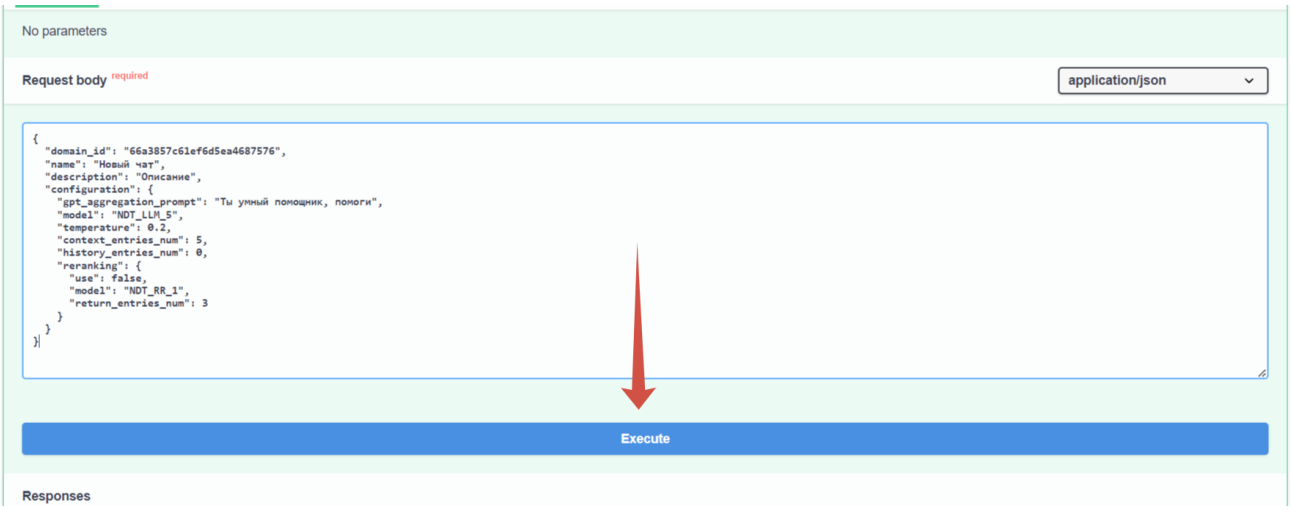
3) Используя полученный domain_id находим метод POST /chat и нажимаем кнопку "Try it out"



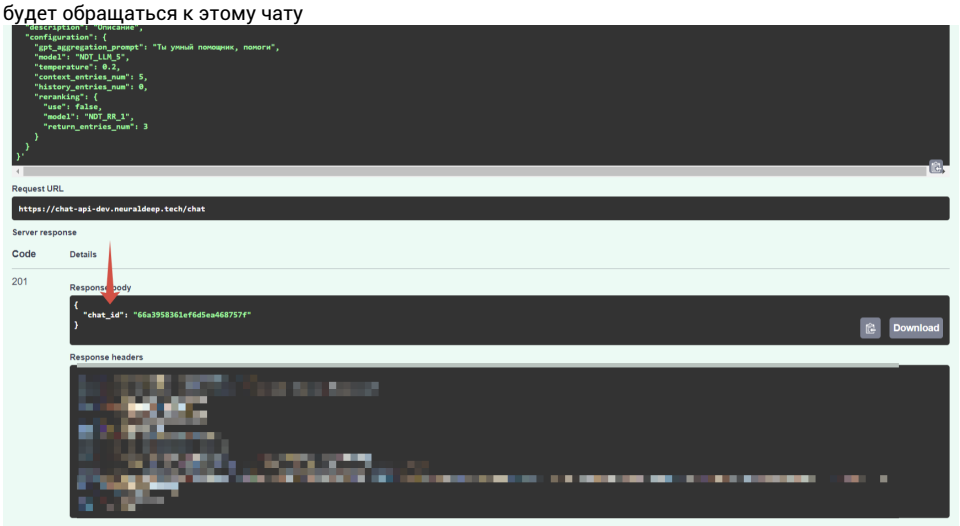

4) Заполняем пункты в поле запроса




Подробнее о пунктах при создании домена:
{
"domain_id": "string", - Тут нужно указать domain_id полученный ранее, для того, чтобы чат был привязан именно к этому домену
"name": "string", - Указываем название чата
"description": "string", - Указываем описание чата (необязательно)
"configuration": {
"gpt_aggregation_prompt": "*write prompt text here otherwise default will be inherited*", - Указываем "aggregation_prompt", инструкцию, которой будет следовать модель в данном чате
"model": "NDT_LLM_5", - Указываем модель, которая будет использоваться в этом чате по базе (изменять необязательно)
"temperature": 0.2, - Указываем базовую температуру модели (потом можно без труда менять, поэтому можно не изменять)
"llm_extra_body_parameters": "dict of llm configuration parameters", - Поле для дополнительных настроек модели (необязательно, если не используется, нужно удалить строчку)
"context_entries_num": 5, - Поле с значением контекста, устанавливает, сколько чанков из домена будет использоваться для каждого ответа (0 - общение с моделью без информации из домена, 3 - среднее значение точности, 5 - базовое значение, которое даёт хорошую точность ответов)
"history_entries_num": 0, - Поле с значением истории, устанавливает, сколько прошлых сообщений будет помнить модель (лучше устанавливать чётное значение, так как модель будет помнить и запрос к ней и свой ответ, например при установке значения на 6 модель будет помнить 3 прошлых запроса пользователя и 3 прошлых ответа)
"reranking": {
"use": false, - Использовать ли реранкер
"model": "NDT_RR_1", - Модель реранкера
"return_entries_num": 3 - Количество данных, которые возвращает реранкер
}
}
}
5) После указания всех данных в необходимых полях нажимаем кнопку "Execute" (на данном примере кстати видно, что "llm_extra_body_parameters": "dict of llm configuration parameters" я убрал из за ненадобности




6) После всех вышесказанных действий, получаем chat_id, который является идентификатором данного чата и по которому потом можно будет обращаться к этому чату



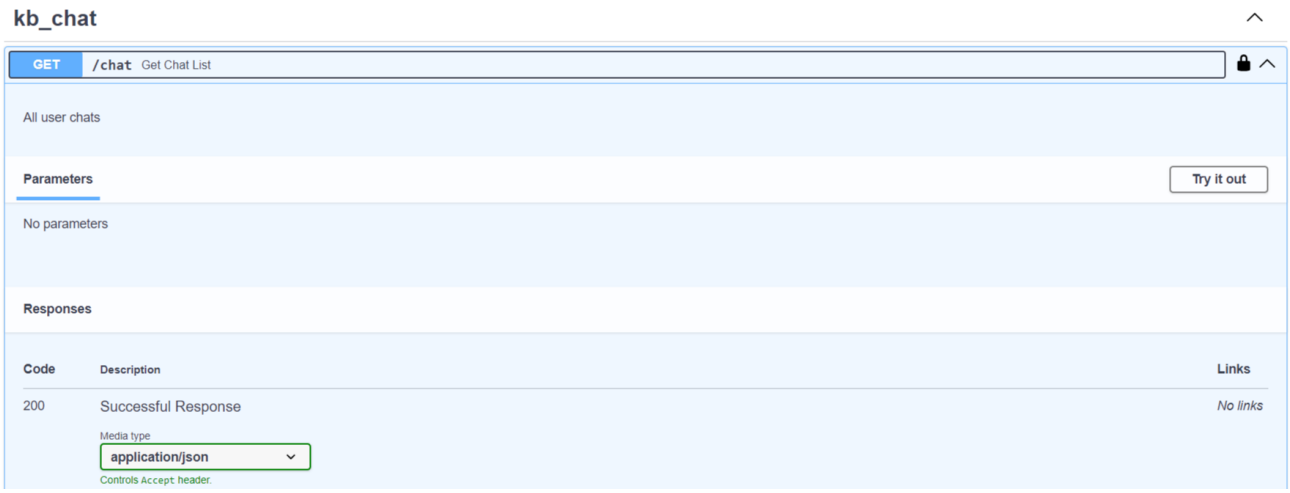

Таким образом мы создаём чат через API.

# Как узнать chat_id/Как задать вопрос к чату через API

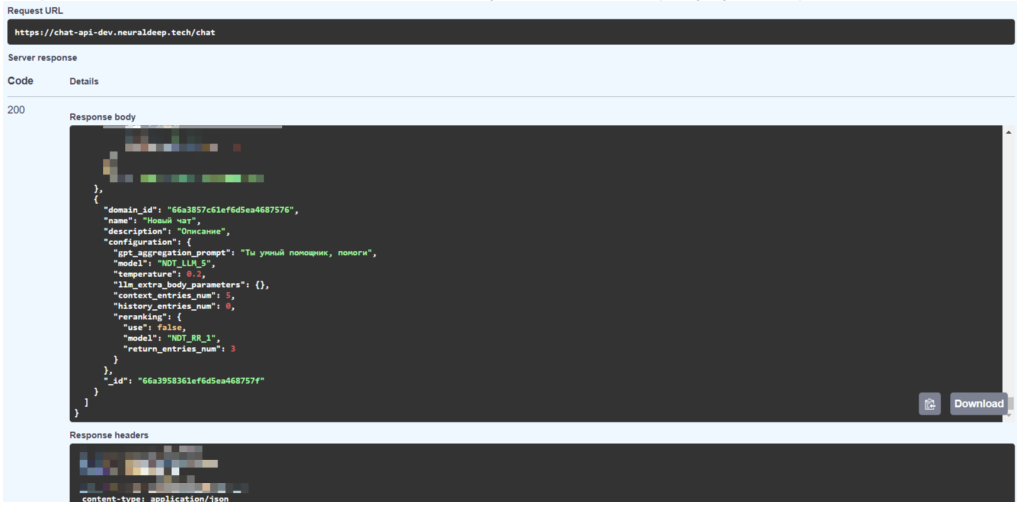
Для того чтобы задать вопрос через API, вам нужно авторизоваться, используя аккаунт, на котором находится домен и чат, через которые вы хотите задать вопрос, а также получить chat_id необходимого чата, к которому будет задан вопрос. Если chat_id уже вам известен, переходите к 3 шагу.

Для обращения к модели вам нужно действовать по этим шагам:

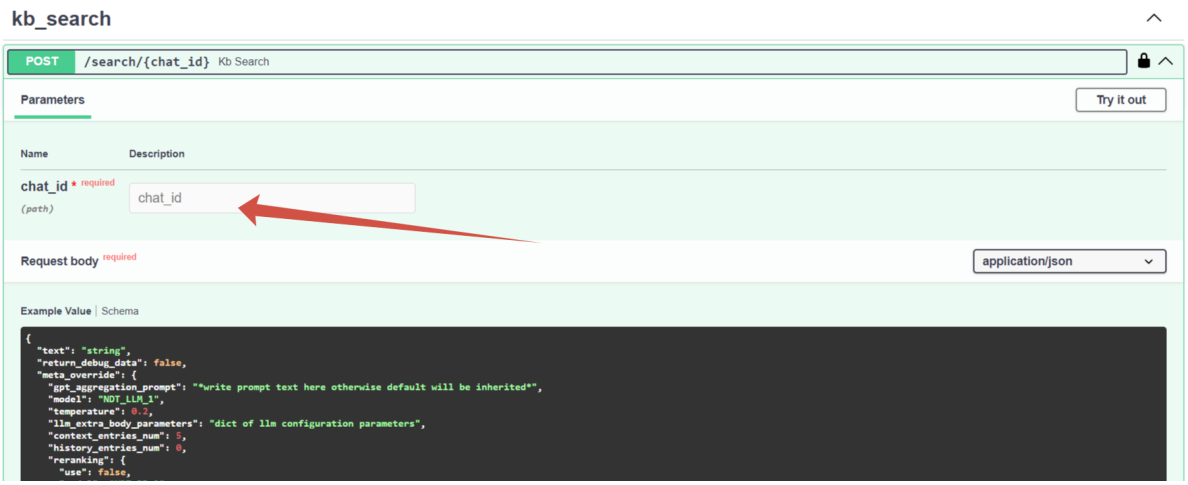
1) Для получения chat_id лучше всего использовать метод GET /chat, сначала открываем его, потом нажимаем "Try it out" и "Execute"



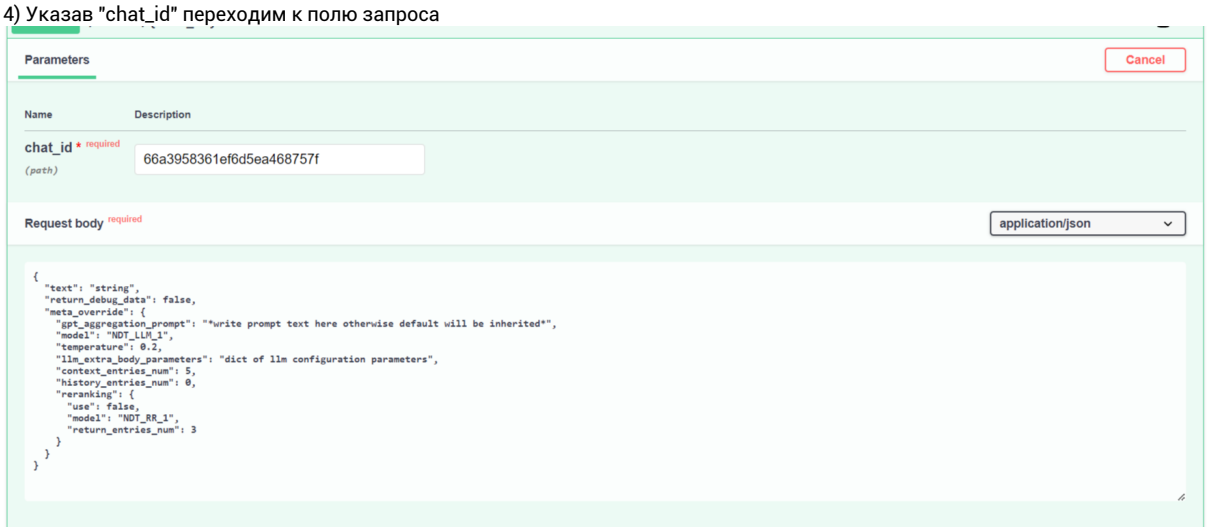

2) Пролистываем ниже и в ответе от API находим нужный чат, к которому будем обращаться, после чего копируем из него поле "_id"



3) Находим метод POST /search/{chat_id} нажимаем "Try it out" и указываем "chat_id"

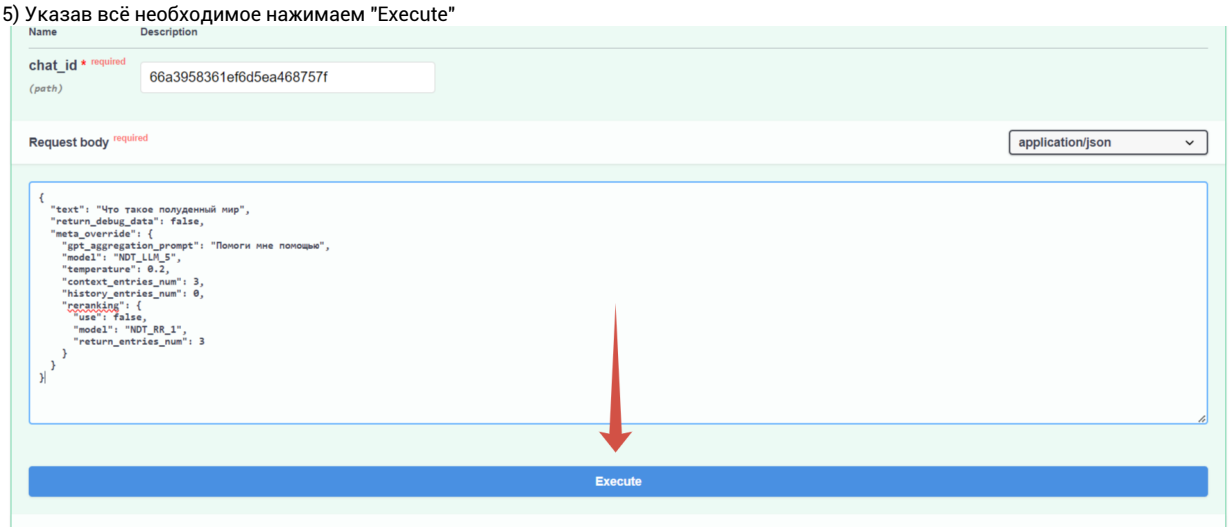



4) Указав "chat_id" переходим к полю запроса

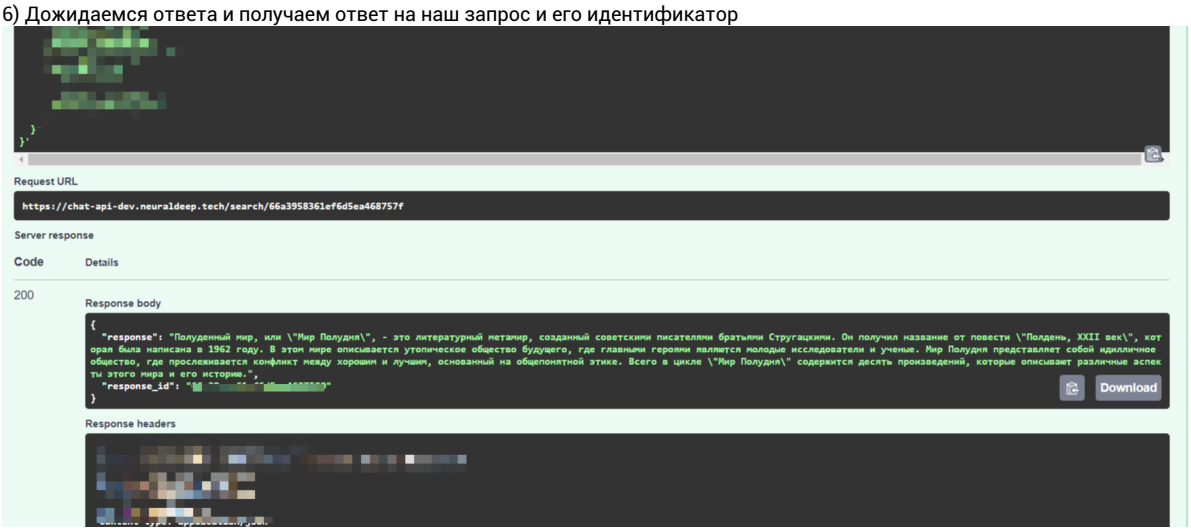



Подробнее про эти атрибуты запроса:
{
"text": "string", - Текст вашего запроса
"return_debug_data": false, - Возвращать ли debug_data, в которых хранится дебаг информация (контекст из базы данных, из истории) false - не возвращать true - возращать
"meta_override": {
"gpt_aggregation_prompt": "*write prompt text here otherwise default will be inherited*", - "aggregation_prompt" (Промт, инструкция для модели)
"model": "NDT_LLM_1", - Модель LLM, к которой будут отправляться запросы
"temperature": 0.2, - Температура, с которой будут отправляться запросы
"llm_extra_body_parameters": "dict of llm configuration parameters", - Поле для дополнительных настроек модели (необязательно, если не используется, нужно удалить строчку)
"context_entries_num": 5, - Поле с значением контекста, устанавливает, сколько чанков из домена будет использоваться для каждого ответа (0 - общение с моделью без информации из домена, 3 - среднее значение точности, 5 - базовое значение, которое даёт хорошую точность ответов)
"history_entries_num": 0, - Поле с значением истории, устанавливает, сколько прошлых сообщений будет помнить модель (лучше устанавливать чётное значение, так как модель будет помнить и запрос к ней и свой ответ, например при установке значения на 6 модель будет помнить 3 прошлых запроса пользователя и 3 прошлых ответа)
"reranking": {
"use": false, - Использовать ли реранкер
"model": "NDT_RR_1", - Модель реранкера
"return_entries_num": 3 - Количество данных, которые возвращает реранкер
}
}
}

5) Указав всё необходимое нажимаем "Execute"




6) Дожидаемся ответа и получаем ответ на наш запрос и его идентификатор




Запрос совершён, информация получена.

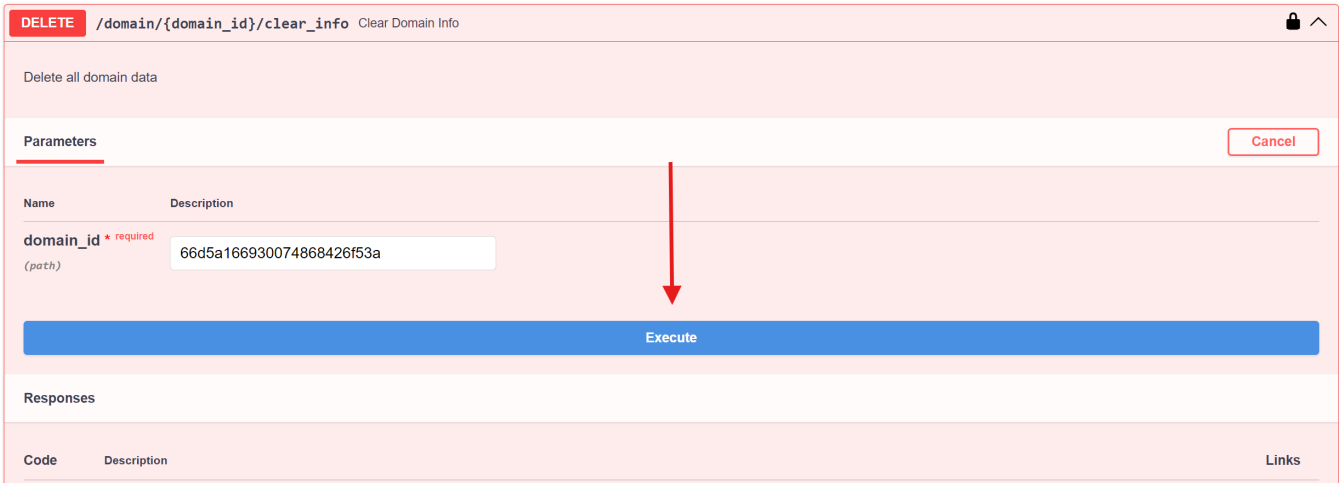
# Удаление данных домена

Для выполнения данного действия, вам потребуется заранее узнать domain_id домена, который вы хотите отчистить от старых данных.

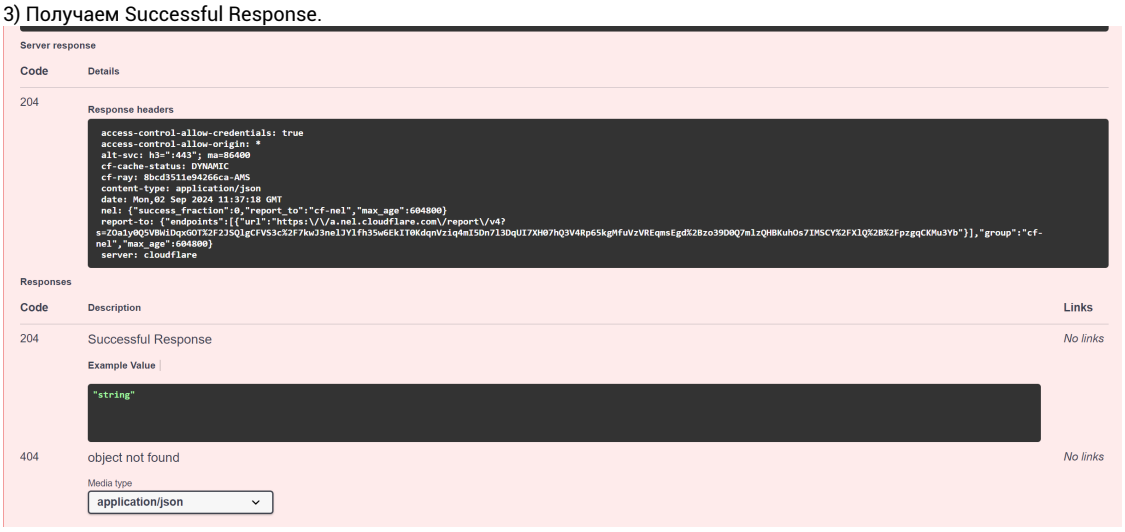
Внимание! При отчистке домена, удаляются все документы, которые были в нём и восстановлению внутри системы они не подлежат, не забывайте сохранять данные, чтоб не потерять важную информацию.

1) Находим функцию DELETE /domain/{domain_id}/clear_info и нажимаем Try it out.


2) Вставляем domain_id и нажимаем Execute.




3) Получаем Successful Response.



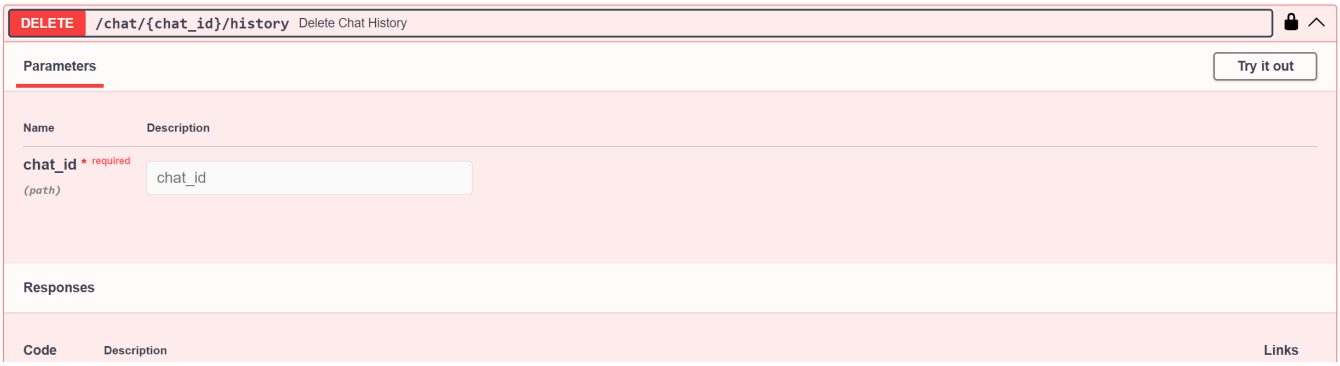

Данные успешно удалены из базы знаний.

# Отчистка истории чата

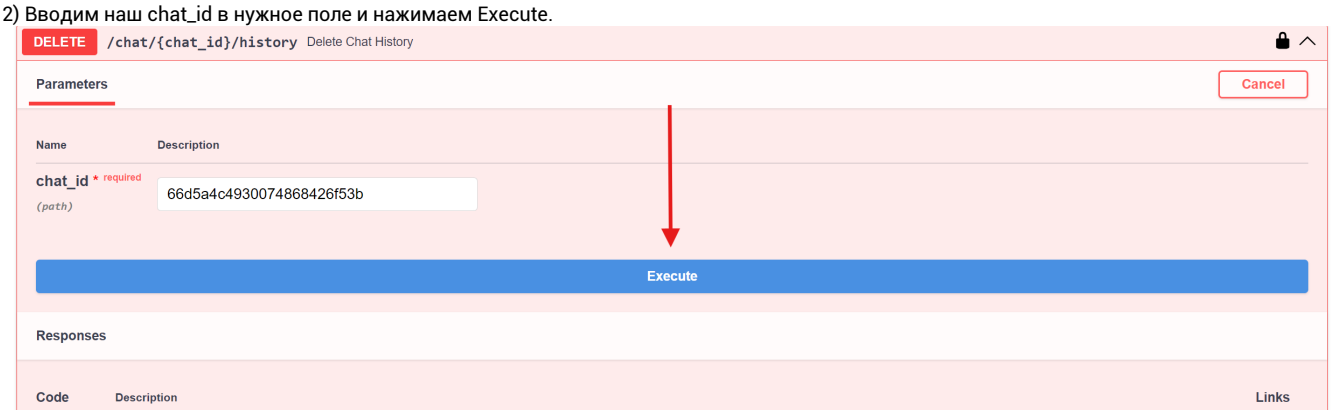
Перед началом нужно знать chat_id чата, историю которого вы хотите отчистить.

Внимание! При отчистке истории, удаляются все сообщения, которые были в нём и восстановлению они не подлежат, не забывайте сохранять данные, чтоб не потерять важную информацию.

1) Находим функцию DELETE /chat/{chat_id}/history и нажимаем Try it out.




2) Вводим наш chat_id в нужное поле и нажимаем Execute.




3) Получаем Successful Response.


История чата успешно отчищена.