Кастомные таблицы

«После того, как бизнес-данные были централизованы и интегрированы, ценность базы данных больше, чем сумма ранее существовавших частей».
— Ларри Эллисон, основатель Oracle

Кастомные таблицы — удобное и практичное хранилище данных, которое вы можете использовать как составную часть решения, создаваемого на Metabot, так и как отдельный сервис, доступный по API из других приложений и платформ.

Кастомные таблицы от Metabot — это альтернатива (или дополнение) к таким продуктам как Google Sheets, AirTable и Notion, созданное специально для разработки приложений с бизнес-логикой.

Примеры использования:

- Таблица пользователей для регистрации/авторизации и личного кабинета в мессенджере;
- Таблицы заявок или заказов, заполняемая ботом, как замена формы;
- Таблицы, представляющие операционную модель вашего бизнеса (ERP);
- Таблицы для учёта выдачи материалов на мероприятии.

Используя кастомные таблицы в сочетании с диалоговыми сценариями, JavaScript и API конструктором вы сможете создать фронтенд и бэкэнд, которая заменят вам мобильное приложение и серверный бэкэнд.

Таблицы незаменимы при реализации глубоких интеграций мессенджеров в процессы предприятия, диалогового интерфейса к системам предприятия и диалогового подключенного опыта (CCX) — таблицы используются для хранения данных для синхронизации систем.

Обращаться к таблицам вы можете прямо в диалоговых сценариях вашего бота, через методы JS: вы сможете выполнять операции работы с таблицей, такие как, поиска записей, их редактирования, удаления.

Если вам необходимо обратиться к таблицам Metabot из внешних систем, для этого вы можете создать внутренние API эндпоинты.

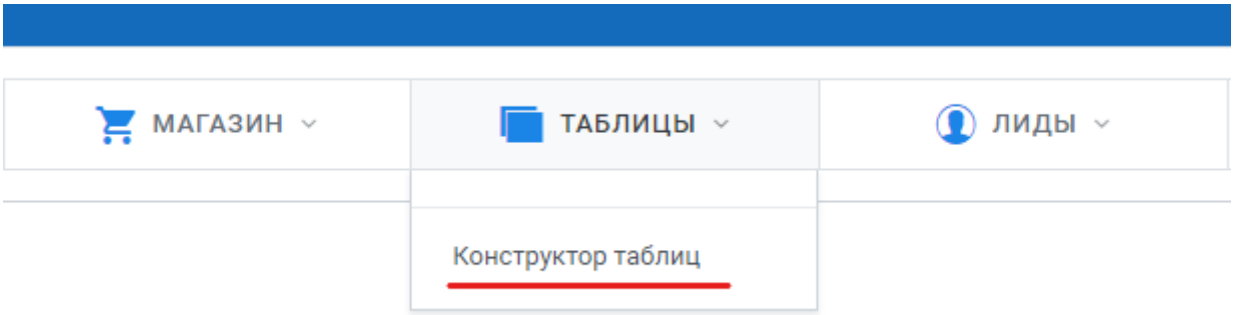
С помощью плагина вы сможете расширить методы работы с таблицами, например, сделав прямые обращение к базе данных и вызов SQL.

>

Работа с SQL доступна на выделенном сервере и в on-premise коробочной версии.

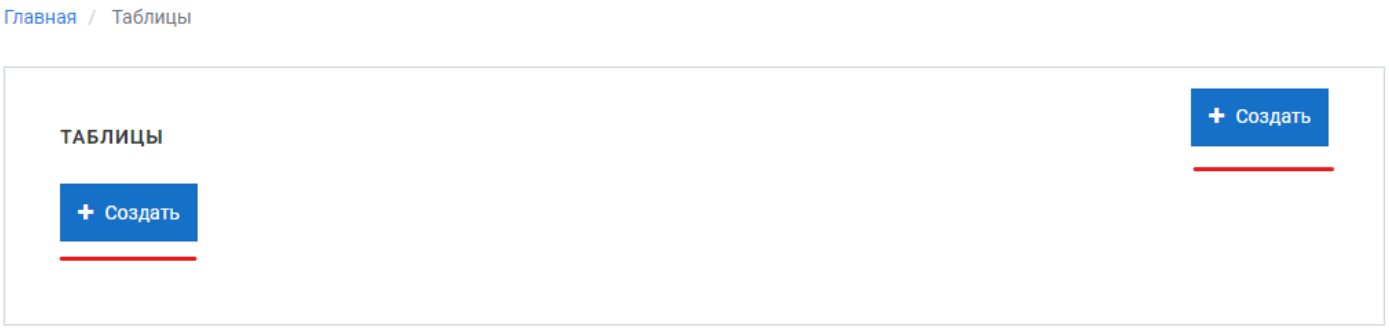
Конструктор таблиц

Конструктор таблиц находится в верхнем меню бота в разделе **Таблицы**:

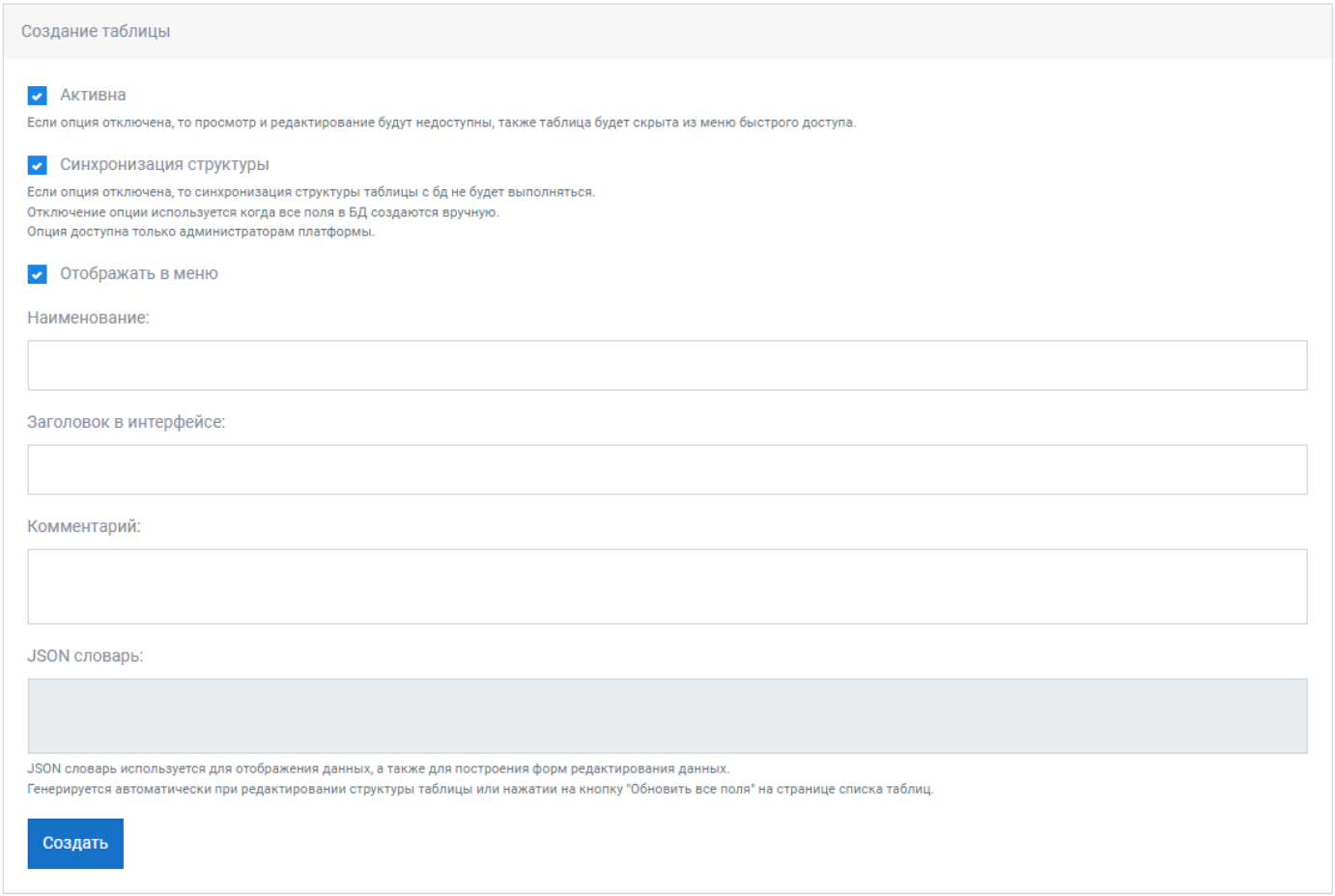


Он предназначен для работы с таблицами, например, для создания таблиц, их форматирования и удаления.

Для создания таблицы необходимо нажать кнопку **Создать**:



В открывшемся окне **Создание таблицы** можно заполнить информацию о создаваемой таблице:



- **Активна** — если опция отключена, то просмотр и редактирование будут недоступны, также таблица будет скрыта из меню быстрого доступа;
- **Синхронизация структуры** — если опция отключена, то синхронизация структуры таблицы с базой данных (БД) не будет выполняться. Отключение опции используется когда все поля в БД создаются вручную;

> Опция **Синхронизация структуры** доступна только администраторам платформы.

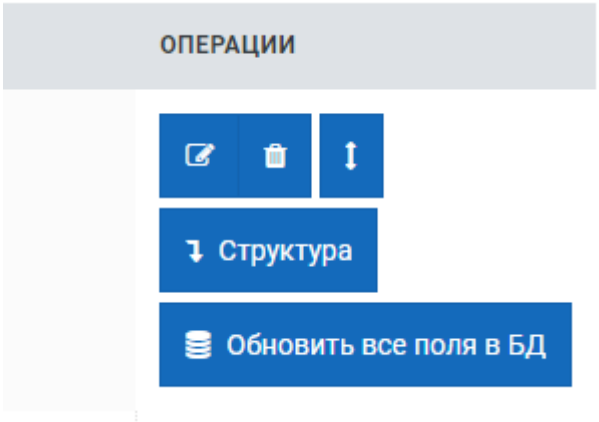
- **Отображать в меню** — если данная опция включена, то таблица будет отображаться в выпадающем списке раздела **Таблицы** верхнего меню бота;
- **Наименование** — название таблицы, которое будет использоваться при обращении к ней методами;

> В имени таблицы можно указывать только латиницу (только нижний регистр), цифры и подчеркивание.

- **Заголовок в интерфейсе** — название таблицы, которое будет отображаться в интерфейсе;
- **Комментарий** — пояснение к таблице;
- **JSON словарь** — JSON словарь используется для отображения данных, а также для построения форм редактирования данных. Генерируется автоматически при

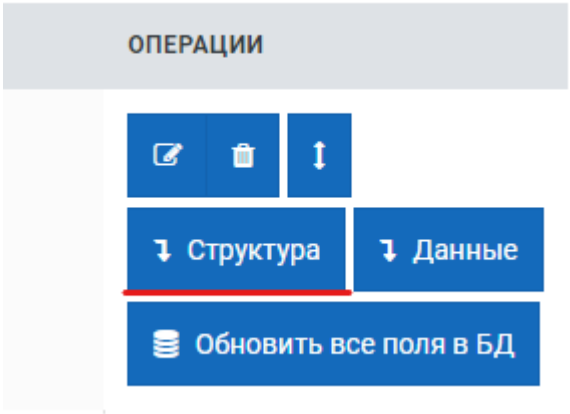
редактировании структуры таблицы или нажатии на кнопку "Обновить все поля" на странице списка таблиц.

После создания таблицы появляются функции ее редактирования, удаления, создания полей и их обновления в БД:



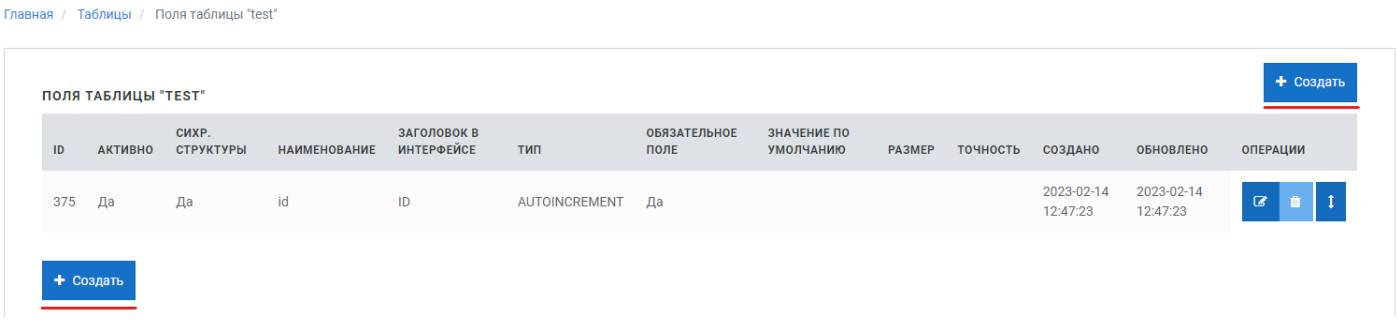
Конструктор полей

Чтобы заполнить таблицу данными, нужно для начала создать ей соответствующие поля. Это можно сделать при помощи операции **Структура:**

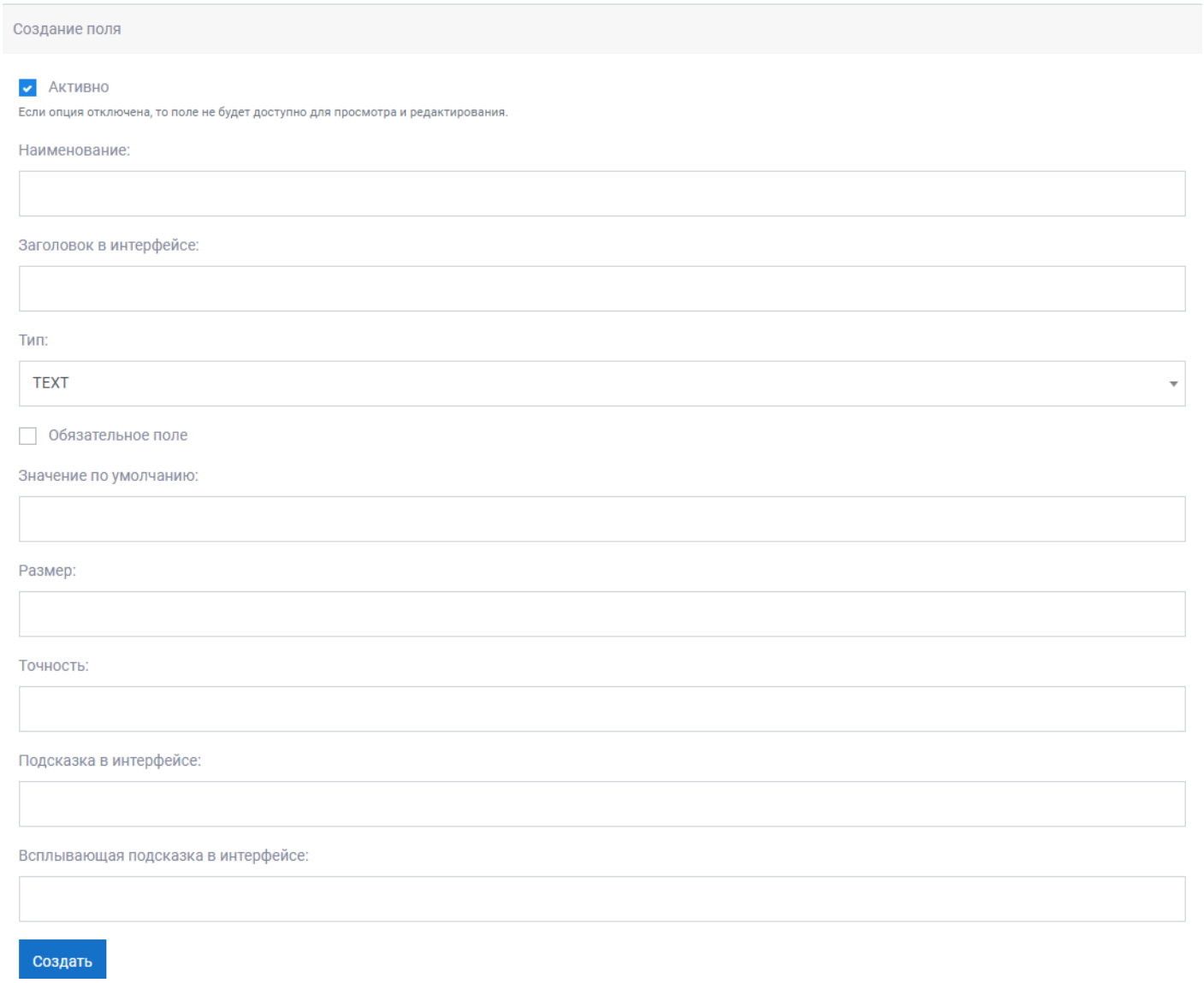


Первым полем любой таблицы всегда является ID строки. Это поле создается автоматически при создании таблицы и не подлежит удалению. По этому полю вы сможете идентифицировать отдельные строки таблицы, а также искать и редактировать записи через JS.

Остальные необходимые поля можно добавить щелкнув по кнопке **Создать:**



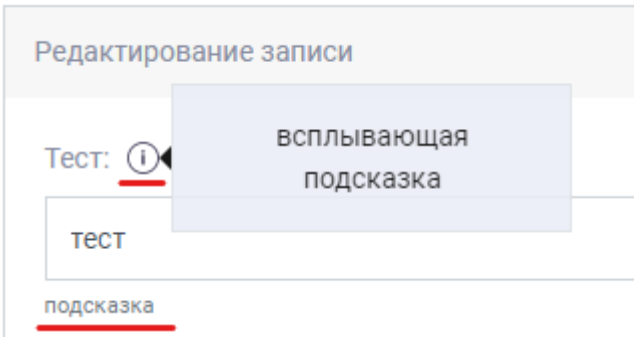
В открывшемся окне **Создание поля** необходимо заполнить информацию о создаваемом поле:



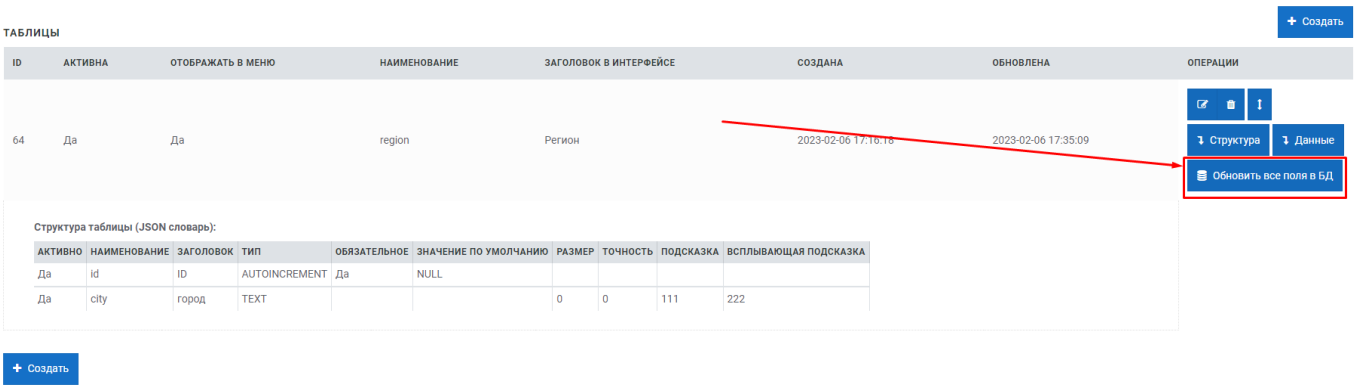
- **Активно** — если опция отключена, то поле не будет доступно для просмотра и редактирования;
- **Наименование** — название поля, которое будет использоваться при обращении к нему методами;

> В имени поля можно указывать только латиницу (только нижний регистр), цифры и подчеркивание.

- **Заголовок в интерфейсе** — название поля, которое будет отображаться в интерфейсе;
- **Тип** — тип данных поля;
  - **Text** — строковый тип данных;
  - **TextArea** — строковый тип данных с возможностью переноса строк;
  - **Int** — числа в диапазоне **от -2 147 483 648 до 2 147 483 647**;
  - **Big_Int** — числа в диапазоне **от -9 223 372 036 854 775 808 до 9 223 372 036 854 775 807**;
  - **Tiny_Int** — числа в диапазоне **от -32768 до 32767**;
  - **Decimal** — числа с плавающей точкой;
  - **DateTime** — дата в формате **дд.мм.гггг --:--:--** ;
  - **Date** — дата в формате **дд.мм.гггг** ;
  - **Vector** — динамический массив;
  - **Created_At** — дата и время создания (не редактируется);
  - **Updated_At** — дата и время обновления (не редактируется);
  - **Deleted_At** — дата и время удаления (не редактируется);
- **Обязательное поле** — если данная опция включена, то в поле не может быть задано значение NULL;
- **Значение по умолчанию** — предустановленное значение поля;
- **Размер** — максимальное количество символов в данном поле;
- **Точность** — количество символов в числе после точки;
- **Подсказка в интерфейсе** — подсказка отображаемая в интерфейса в режиме просмотра и редактирования данных;
- **Всплывающая подсказка в интерфейсе** — подсказка в интерфейсе, возникающая при наведении мышкой на знак вопроса рядом с полем.



После создания и настройки всех ячеек таблицы можно провести обновление таблицы в базе данных платформы. Для это необходимо в конструкторе таблиц нажать **Обновить все поля в БД**:



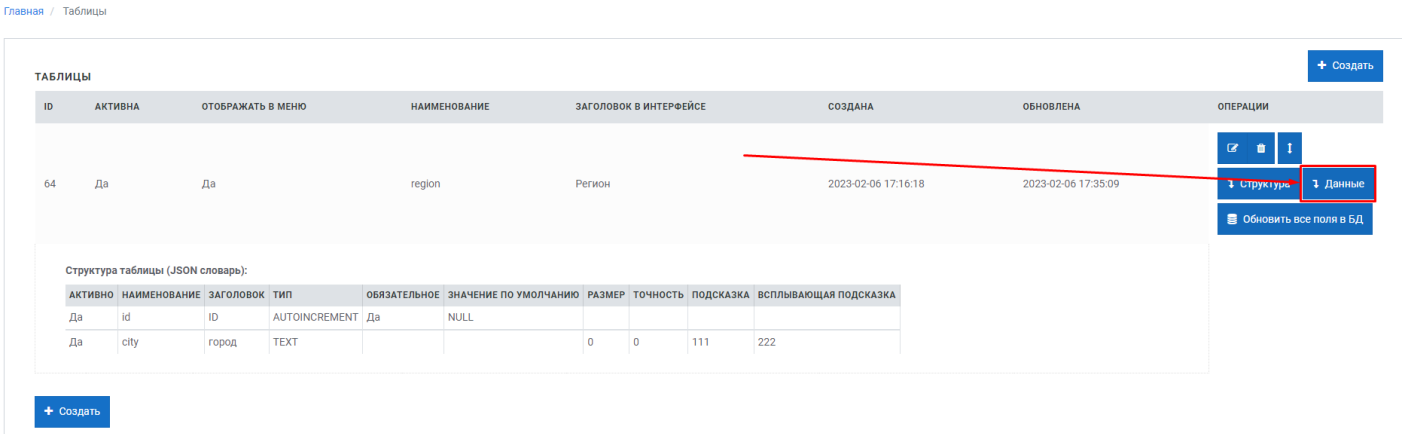
Это синхронизирует и обновит таблицу в базе данных, а так же сгенерирует ее **JSON словарь** (Структуру таблицы).

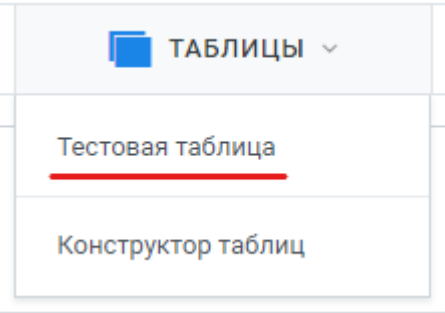
> На текущий момент, в качестве базы данных платформа Metabot поддерживает PostgreSQL.

> Формирование модели данных через текстовое описание мета-словаря на базе JSON в текущей версии не доступно.

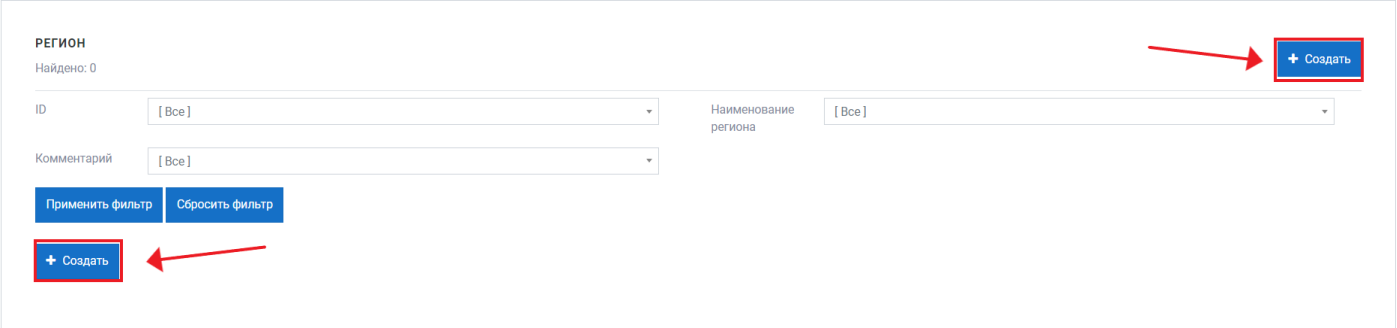
Просмотр и редактирование данных

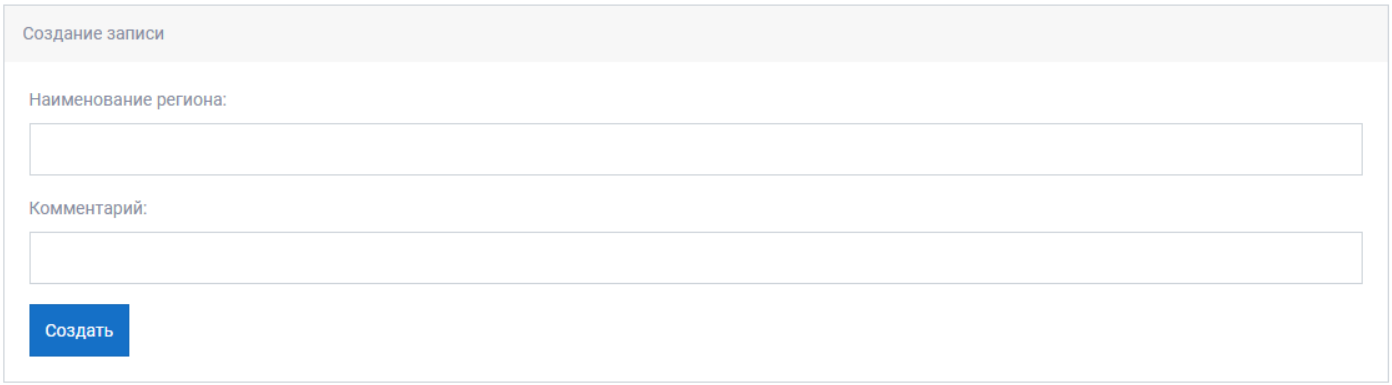
После создания структуры таблицы можно вручную заполнить ее данными. Для этого необходимо зайти в **Данные** в конструкторе таблиц либо, если вы включали опцию **Отображать в меню**, то нажать на название таблицы в выпадающем списке раздела **Таблицы** верхнего меню бота.





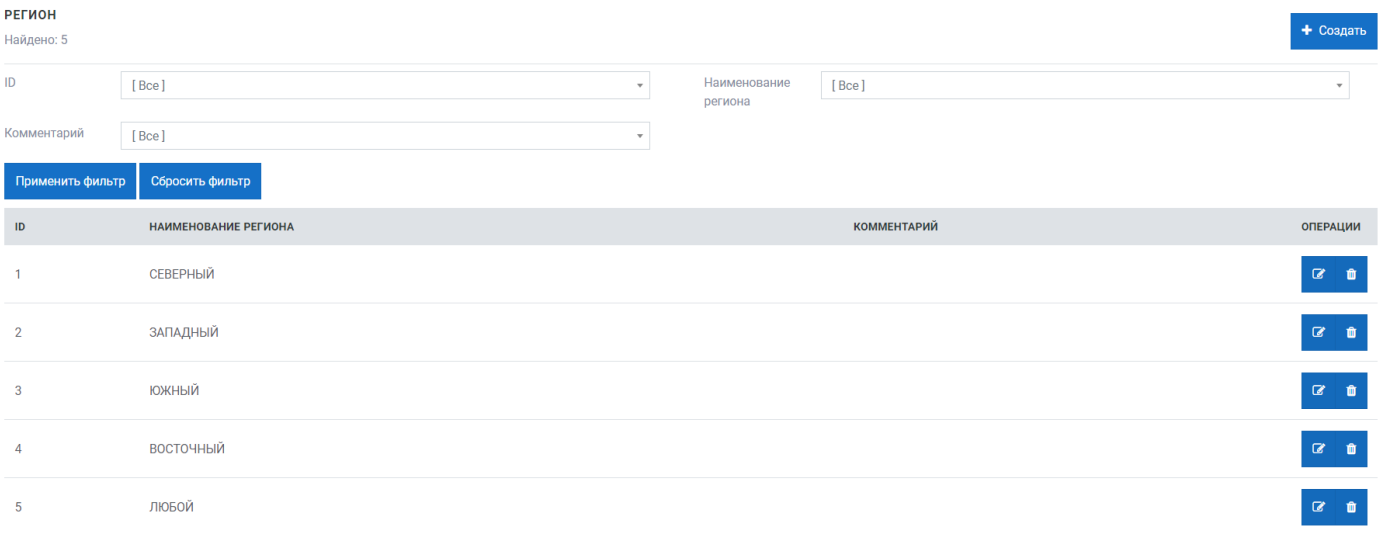
После этого нажмите на кнопку **Создать** и добавьте необходимые данные.





После создания запись появится в таблице. Записи так же можно редактировать и удалять.

По итогу у вас получится таблица, которая может выглядеть, например, так:

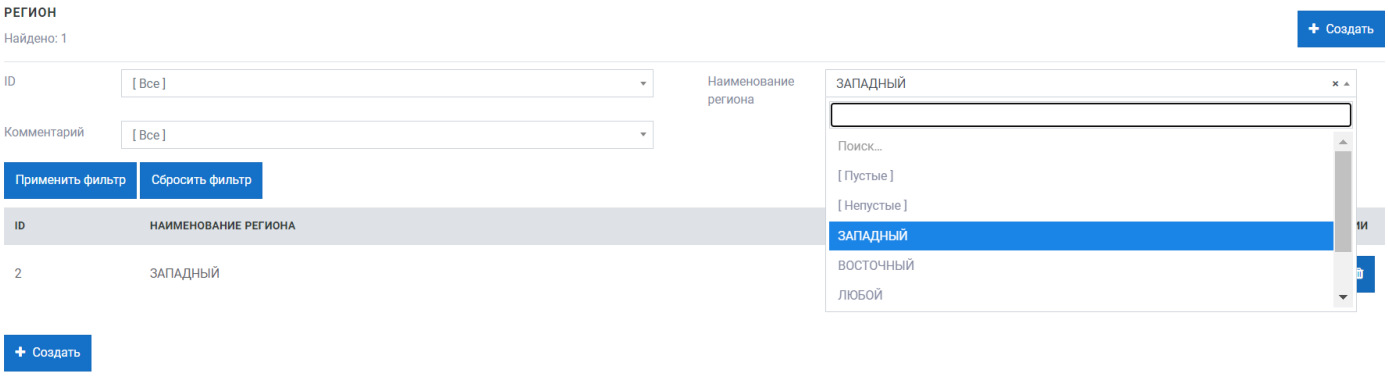


> Данная таблица**"Регионы"** будет использоваться для примеров использования методов в **Справочнике функций JS**.

Фильтры

Для ID и всех кастомных полей таблиц предусмотрены фильтры. В зависимости от созданных полей таблицы, будут автоматически созданы фильтры в виде выпадающих списков.

При задании фильтров и нажатии на кнопку **Применить фильтр** будут выведены соответствующие записи в таблице:



> С кастомными таблицами так же можно работать с помощью JS методов. Все методы и их описание вы можете посмотреть в **Справочнике по функциям JS**

Версия #9
Юрий Гарашко создал 2023-06-12 10:59:05 UTC
Ирина Петрова обновил 2025-11-11 07:00:50 UTC