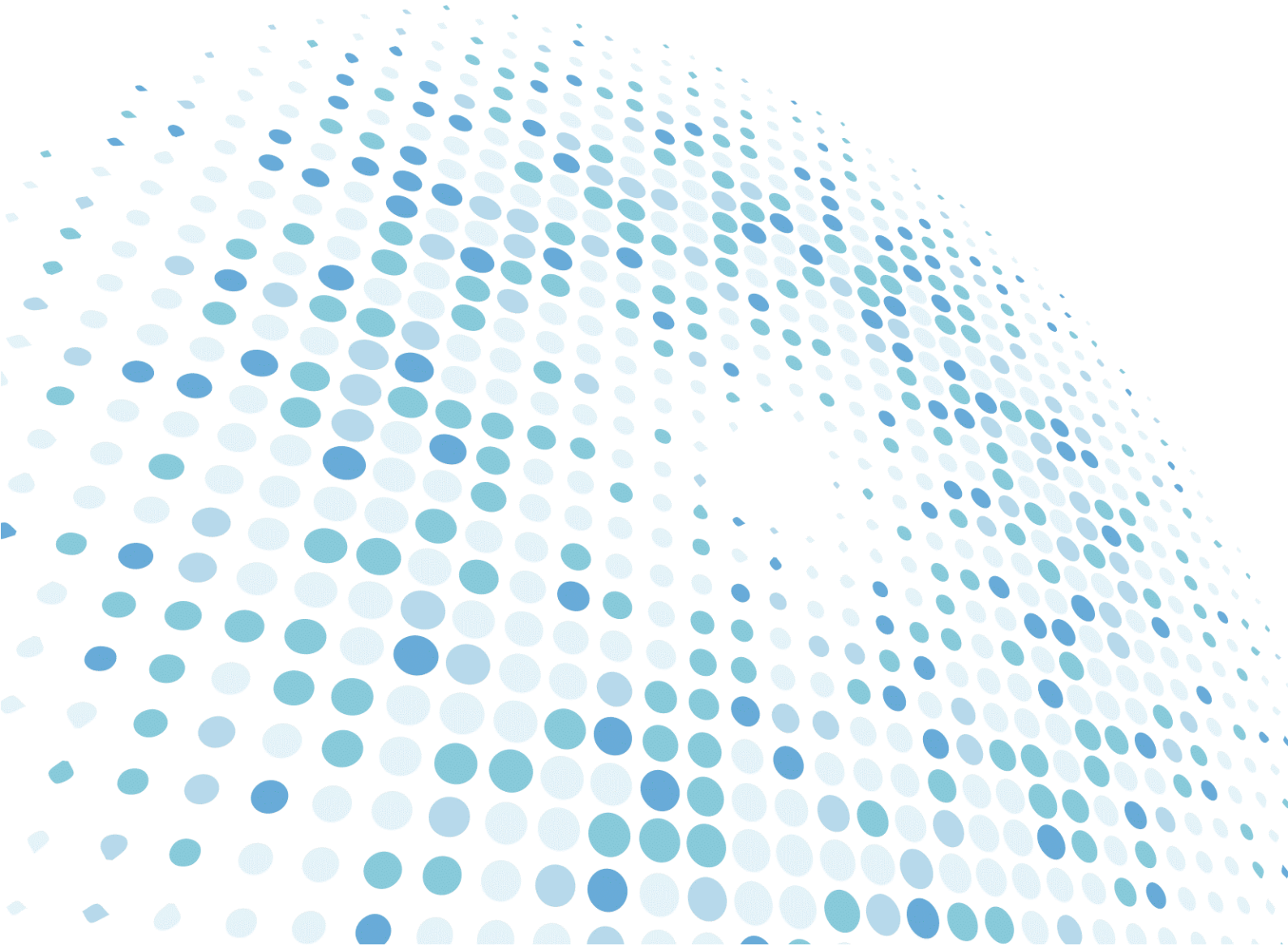
Цифровая платформа
«Управление в пространстве» (ЦП УВП)

версия 0.43

РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ



Содержание

1 Перечень терминов, сокращений и обозначений 9
2 Назначение и основные возможности 10
2.1 Ключевые особенности 10
2.2 Возможности платформы 10
2.2.1 Режимы работы 11
3 Подготовка к работе 12
3.1 Запуск системы 12
4 Пользовательское приложение 14
4.1 Пресеты 14
4.1.1 Переключение между пресетами 14
4.1.2 Редактирование и просмотр пресета 14
4.1.3 Автообновление пресета 15
4.1.4 Отображение пресета модальным окном 16
4.1.5 Экспорт пресета 16
4.1.6 Печать визуализаторов пресета 17
4.1.7 Особенности работы визуализаторов, которые меняются после событий 17
4.1.8 Описание работы пресета (в части backend) 18
4.1.9 Открытие пресета (в части backend) 18
4.1.10 Избранные задачи 19
4.2 Визуализаторы 20
4.2.1 Меню визуализатора 21
4.2.2 Особенности размещения визуализаторов на пресете 23
4.2.3 Состояния визуализаторов 24
4.2.4 Экспорт визуализаторов 24
4.2.5 Быстрое добавление визуализатора 24
4.2.6 Добавление визуализаторов вместе с привязкой к данным 26
4.2.7 Настройки блока визуализатора 30
4.2.8 Настройки визуализатора 31
4.2.9 Размещение визуализатора 39
4.2.10 Сообщения об ошибках в визуализаторах 44
4.2.11 Сохраненные визуализаторы и управление состояниями 45
4.2.12 Экспорт визуализаторов 51
4.3 Режимы работы 53
4.3.1 Режим редактирования пресета 53
4.3.2 Режим автоматического обновления визуализаторов 54
4.3.3 Режим embedded – встраивание без меню 54
4.3.4 Режим открытия приложения без системного меню 54

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

4.4 Сохранение данных из пользовательского приложения 54
4.4.1 Редактирование данных 55
4.4.2 Сохранение изменений 55
4.4.3 Отмена изменений 56
4.4.4 Расчет графа после сохранения 57
4.4.5 Сохранение данных при использовании воркеров 57
4.5 Блоки 57
4.5.1 Санитайзер HTML 57
4.6 Стандартные сочетания клавиш 59
4.7 Поиск 60
4.7.1 Задачи 60
4.7.2 Пресеты 61
4.7.3 Блоки 61
4.7.4 Визуализаторы 62
4.8 Экспериментальные функции 64
4.8.1 Включение отображения экспериментальные функции 64
4.8.2 Индикация экспериментальных функций 64
4.9 Особенности работы приложения с настройкой браузера «Экономия памяти» 65
4.10 Ошибки приложения 65
4.10.1 Ошибка со статусом HTTP 403 65
4.10.2 Ошибка со статусом HTTP 404 65
4.10.3 Ошибка со статусом HTTP 500 65
4.10.4 Ошибка со статусом HTTP 503 65
4.10.5 Разрыв соединения 66
5 Навигатор задач 67
5.1 Задачи 67
5.1.1 3D-Карты 67
5.1.2 Доступ 67
5.1.3 Граф 67
5.1.4 Импорт/экспорт задачи 67
5.1.5 НСИ и Источник данных 69
5.1.6 Управление доступом к задачам 71
5.2 Папки 75
5.3 Теги 78
5.3.1 Добавление тега 79
5.3.2 Изменение тега 80
5.3.3 Удаление тега 81
5.4 Выбор нескольких тегов 81

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

5.4.1 Тег «Мои задачи» 82
5.5 Окно «О приложении» 83
6 Библиотека блоков 84
6.1 Визуализаторы (библиотека) 84
6.1.1 Диаграммы 84
6.1.2 Таблицы 130
6.1.3 Карта 170
6.1.4 Элементы управления 287
6.1.5 Шаблонизатор Jinja 380
6.1.6 Контейнеры 381
6.1.7 Изображения 395
6.2 Импорт и экспорт данных 401
6.2.1 Блок «Соединение с PostgreSQL DB» 401
6.2.2 Блок «Соединение с Microsoft SQL DB» 406
6.2.3 Блок «Соединение с Oracle DB» 408
6.2.4 Блок «Соединение с SAP BI» 411
6.2.5 Блок «Соединение с Apache Kafka» 413
6.2.6 Блок «Соединение SOAP» 417
6.2.7 Блок «Соединение с Mongo» 419
6.2.8 Блок «Соединение с ClickHouse» 422
6.2.9 Блок «SQL – выражение» 425
6.2.10 Блок «SQL – запрос для рядов» 428
6.2.11 Блок «SQL – запрос-справочник» 449
6.2.12 Блок «SAP – запрос» 470
6.2.13 Блок «Генерация временного ряда» 472
6.2.14 Блок «Расширение атрибутов временного ряда» 473
6.2.15 Блок «Сохранение данных в БД» 475
6.2.16 Блок «Справочник» 475
6.2.17 Блок «Показатели» 480
6.2.18 Блок «JSON-Справочник» 482
6.2.19 Блок «JSON-Показатели» 483
6.2.20 Блок «Импорт файла» 486
6.2.21 Блок «SOAP-запрос» 494
6.2.22 Блок «Apache Kafka Consumer» 497
6.2.23 Блок «Apache Kafka Producer» 498
6.2.24 Блок «MongoDB – запрос» 499
6.2.25 Блок «Конструктор запросов» 502
6.3 Источники данных и НСИ 505

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.3.1 Блок «&Источник данных» 505
6.3.2 Блок «&НСИ» 507
6.3.3 Блок «&НСИ – Выход» 507
6.4 Оформление графа 507
6.4.1 Блок «Граф» 507
6.4.2 Блок «Задача» 508
6.4.3 Блок «Надпись» 508
6.4.4 Блок «Прямоугольник» 509
6.5 Преобразование данных 511
6.5.1 Блок «Выражение» 511
6.5.2 Блок «Конвертор объектов в ряды» 533
6.5.3 Блок «Коннектор временных рядов» 534
6.5.4 Блок «Обрезка рядов» 536
6.5.5 Блок «Правила» 537
6.5.6 Блок «Преобразование рядов» 540
6.5.7 Блок «Срез» 545
6.5.8 Блок «Формирование кросс-таблицы» 548
6.6 Статистические методы 551
6.6.1 Блок «ARIMA» 551
6.6.2 Описание работы блока 554
6.6.3 Блок «K-Means» 561
6.6.4 Блок «Автоподбор линейной регрессии» 568
6.6.5 Блок «Адаптивный МНК (AOLS)» 572
6.6.6 Блок «Бинарная регрессия» 579
6.6.7 Блок «Импорт статистических характеристик из БД» 584
6.6.8 Блок «Корреляционная матрица» 585
6.6.9 Блок «Линейная регрессия» 588
6.6.10 Блок «МНК с ограничениями» 593
6.6.11 Блок «Методика АМ. Задача оптимизации» 597
6.6.12 Блок «Методика АМ» 601
6.6.13 Блок «Методы автоподбора факторов» 604
6.6.14 Блок «Нелинейная регрессия» 610
6.6.15 Блок «Тренд» 614
6.6.16 Блок «Экспоненциальное сглаживание» 623
6.6.17 Блок «Эластичная сеть» 627
6.7 Управление порядком расчета 631
6.7.1 Блок «Входные данные» 631
6.7.2 Блок «Выходные данные» 633

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.7.3 Блок «Генератор события» 635
6.7.4 Блок «Обработчик события» 636
6.7.5 Блок «Диспетчер» 637
6.7.6 Блок «Итератор» 641
6.7.7 Блок «Выход итератора» 651
6.7.8 Блок «Условие» 655
6.7.9 Блок «Пресет» 658
6.8 Форматирование 660
6.8.1 Блок «Стили таблицы» 660
6.8.2 Блок «Форматирование по условию» 668
6.8.3 Блок «Шкала форматирования» 686
6.9 Элементы машинного обучения 691
6.9.1 Блок «Python» 691
6.9.2 Локальная разработка. Пакет vm-local-developer 706
6.9.3 Модуль vmResource 712
6.9.4 Описание пользовательских настроек 717
6.10 Формы 743
6.10.1 Блок «Кнопка» 743
6.10.2 Блок «Строка ввода» 744
6.10.3 Блок «Вопрос» 747
7 Граф 750
7.1 Руководство по работе с графом 750
7.1.1 Добавление блоков графа 751
7.1.2 Добавление связей между блоками 754
7.1.3 Связи между блоками 758
7.2 Работа с блоками 760
7.2.1 Структура блока 760
7.2.2 Оформление блока 762
7.2.3 Групповые операции 776
7.2.4 История изменений блоков 777
7.2.5 Поддерживаемые типы входов и выходов в блоке 780
7.2.6 Предпросмотр результатов настройки блока 786
7.2.7 Переменные входов/выходов 787
7.3 Работа со связями 792
7.3.1 Обработка данных в связях между блоками 792
7.4 Расчет графа 794
7.4.1 Описание процесса расчета графа в режиме «Отладки графа» 796
7.4.2 Описание процесса расчета графа, запущенного из пресета 799

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

7.4.3 События в блоках 800
7.4.4 Механизм расчета 801
7.5 Режим отладки 802
7.6 Редактирование и отладка блока Python в браузере 807
7.6.1 Редактирование кода 807
7.6.2 Отладка кода 808
7.7 Совместная работа 813
7.7.1 Отображение пользователей в графе при совместной работе 813
7.7.2 Отображение совместной работы с блоками 813
7.7.3 Отмена и восстановление изменений, вносимых пользователем (undo/redo) 816
7.8 Мастера 817
7.8.1 Мастер настройки JSON справочника 817
7.8.2 Мастер настройки OLAP 825
7.8.3 Мастер настройки импорта 828
7.8.4 Мастер настройки показателей 834
7.8.5 Мастер настройки JSON показателей 843
7.8.6 Мастер настройки расширения атрибутов 850
7.8.7 Мастер настройки справочника 855
7.8.8 Мастер данных блока 862
7.8.9 Мастер анализа временного ряда для блока «Арима» 865
7.8.10 Мастер автоподбора максимальных/специфических порядков для блока «Арима» 867
7.8.11 Мастер настройки блока «Выходные данные – НСИ» 869
7.9 Отображение визуализатора в блоке 872
7.10 Оформление графа (блоки, стрелки и т.д.) 874
7.10.1 Визуальные настройки блоков на графе 874
7.11 Фигуры 876
7.11.1 Миникарта (минограф) 879
7.11.2 Масштабирование графа 882
7.11.3 Избранные блоки 883
7.12 Настройки задачи 887
7.12.1 Наименование 887
7.12.2 Расширенные возможности режима отладки 887
7.12.3 Задача запрещена для расчета 888
7.12.4 Использовать механизм оповещений 888
7.12.5 Отображение ошибок в режиме просмотра пресета 888
7.12.6 Обновить граф до последней версии 888
7.13 Лог расчета 889

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

7.14 Undo/Redo 889
7.14.1 Особенность работы 890
8 Мобильное приложение 891
8.1 Описание 891
8.2 Функционал 891
8.3 Особенности 892
8.4 Ограничения 892

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

1 Перечень терминов, сокращений и обозначений

Сокращение	Определение
БД	База данных
ЛКМ	Левая кнопка мыши
НСИ	Нормативно-справочная информация
ПКМ	Правая кнопка мыши
ЦП УВП, Платформа	Цифровая платформа «Управление в пространстве»
API	анг. Application Programming Interface – программный интерфейс приложения
JSON	англ. JavaScript Object Notation – текстовый формат обмена данными, основанный на JavaScript
GUID	анг. Globally Unique Identifier – статистически уникальный 128-битный идентификатор
LDAP	анг. Lightweight Directory Access Protocol – протокол доступа к службе каталогов
OLAP	анг. On-Line Analytical Processing – технология обработки данных, заключающаяся в подготовке суммарной информации на основе больших массивов данных, структурированных по многомерному принципу

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

2 Назначение и основные возможности

Цифровая платформа «Управление в пространстве» – платформа, автоматизирующая процесс загрузки, анализа и интерпретации данных на основе методов математического моделирования.

Российская аналитическая low-code BI-платформа «Управление в пространстве» со встроенным искусственным интеллектом и геоаналитикой, которые используются для поддержки принятия управленческих решений. Используется программное обеспечение с открытым исходным кодом (не используются права третьих лиц, одним из основных преимуществ является свобода использования, модифицирования программы), полностью собственная разработка компании БИТ.

Включена в Единый реестр российских программ для ЭВМ и БД под рег. номером 7516. Может быть использована для импортозамещения таких продуктов как Power BI, Tableau, Qlik Sense, SPSS Statistics, ArcGIS Business Analyst.

BI-платформа «Управление в пространстве» в сравнении с другими BI-платформами на российском рынке имеет расширенные функциональные возможности в части решения задач 3D-картографического мониторинга, построения цифровых двойников, моделей и расчетов в пространстве, динамического моделирования, сценарного и целевого прогнозирования и формирования рекомендаций, low-code и визуального конструирования аналитических решений.

2.1 Ключевые особенности

- Платформа реализует комплексную аналитическую поддержку принятия решений с использованием технологий продвинутой аналитики на основе математического моделирования, геоаналитики, искусственного интеллекта и больших данных.
- Платформа обеспечивает визуальное конструирование аналитических приложений.

2.2 Возможности платформы

- Базовая аналитика:
    - сбор, загрузка и преобразование данных;
    - визуализация данных с использованием электронной картографии, таблиц и диаграмм, аналитических панелей.
- Продвинутая аналитика:
    - геоаналитика (продвинутая обработка пространственных данных, 3D-картография, геоинформационный мониторинг, пространственный анализ и пространственное моделирование);
    - математическое моделирование с использованием математической статистики и методов машинного обучения (ML);
    - сценарное многовариантное прогнозирование, оптимизационное и целевое управление;
    - предписывающая аналитика на основе генерации рекомендаций на естественном языке с использованием продукционного подхода.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

2.2.1 Режимы работы

Цифровая платформа поддерживает два режима работы:

- конструкторский;
- пользовательский.

Конструкторский режим (Граф) – режим, который позволяет размещать в приложении блоки, связи, произвольные элементы (текст, изображение, видео), слайды и совершать операции над ними (копирование, удаление, перемещение).

Конструкторский режим предназначен для решения задач:

- конструирования бизнес-логики приложения для последующего использования в Пользовательском режиме преднастроенных вариантов отображения информации;
- загрузки первичных данных средствами импорта.

Пользовательский режим (Пользовательское приложение) – режим при котором осуществляется визуализация данных в виде:

- картографической информации;
- дашбордов – Dashboard, аналитическая (контрольная, приборная) панель – общепринятое средство визуализации данных в современных BI-решениях.

Пользовательский режим предназначен для решения функциональных задач:

- мониторинг и анализ данных;
- прогнозирование и моделирование.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

3 Подготовка к работе

3.1 Запуск системы

Для начала работы с платформой необходимо запустить браузер и в адресной строке указать веб-ресурс Системы.

После этого будет открыта страница авторизации (Рисунок 1). Для входа в личный кабинет сотруднику необходимо ввести свой логин и пароль, после чего нажать на кнопку «Войти», откроется окно выбора задачи (Рисунок 2).

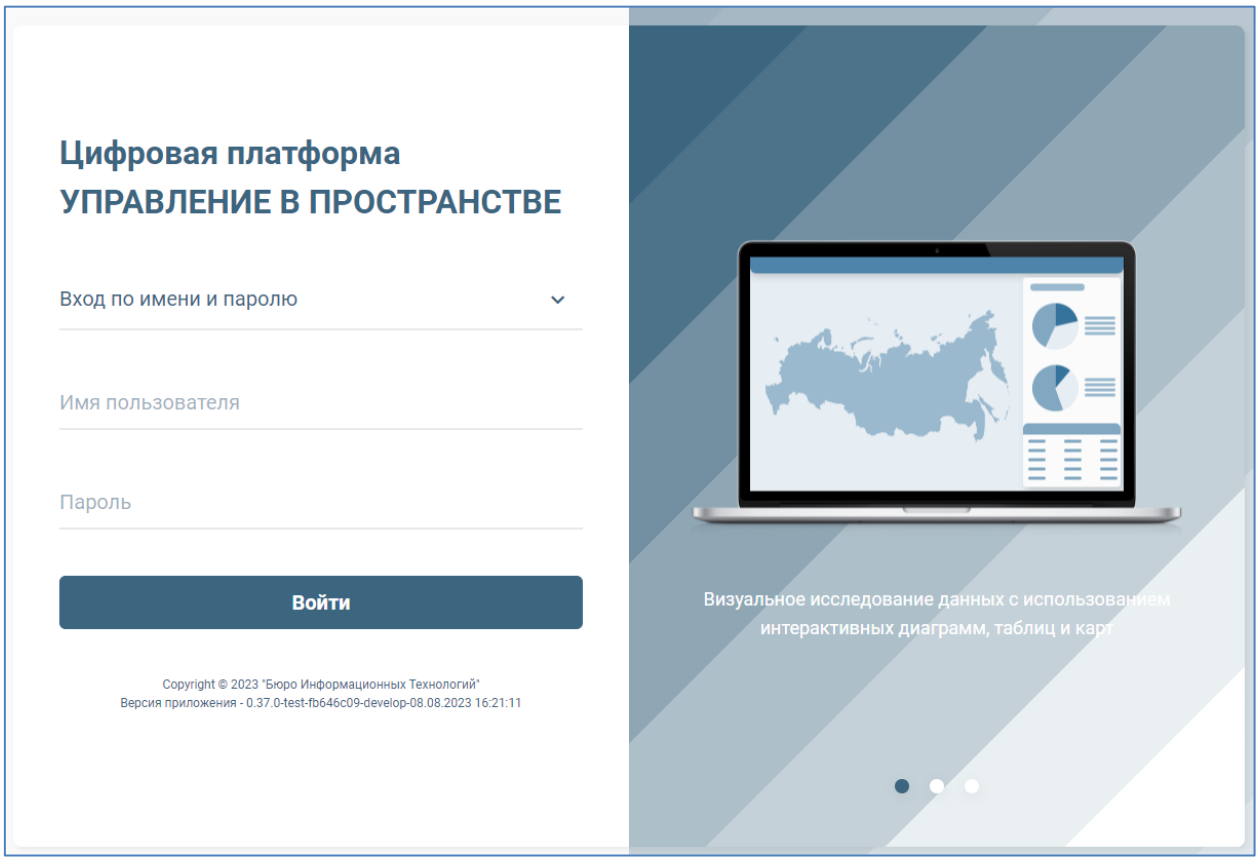


Рисунок 1 – Страница авторизации

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

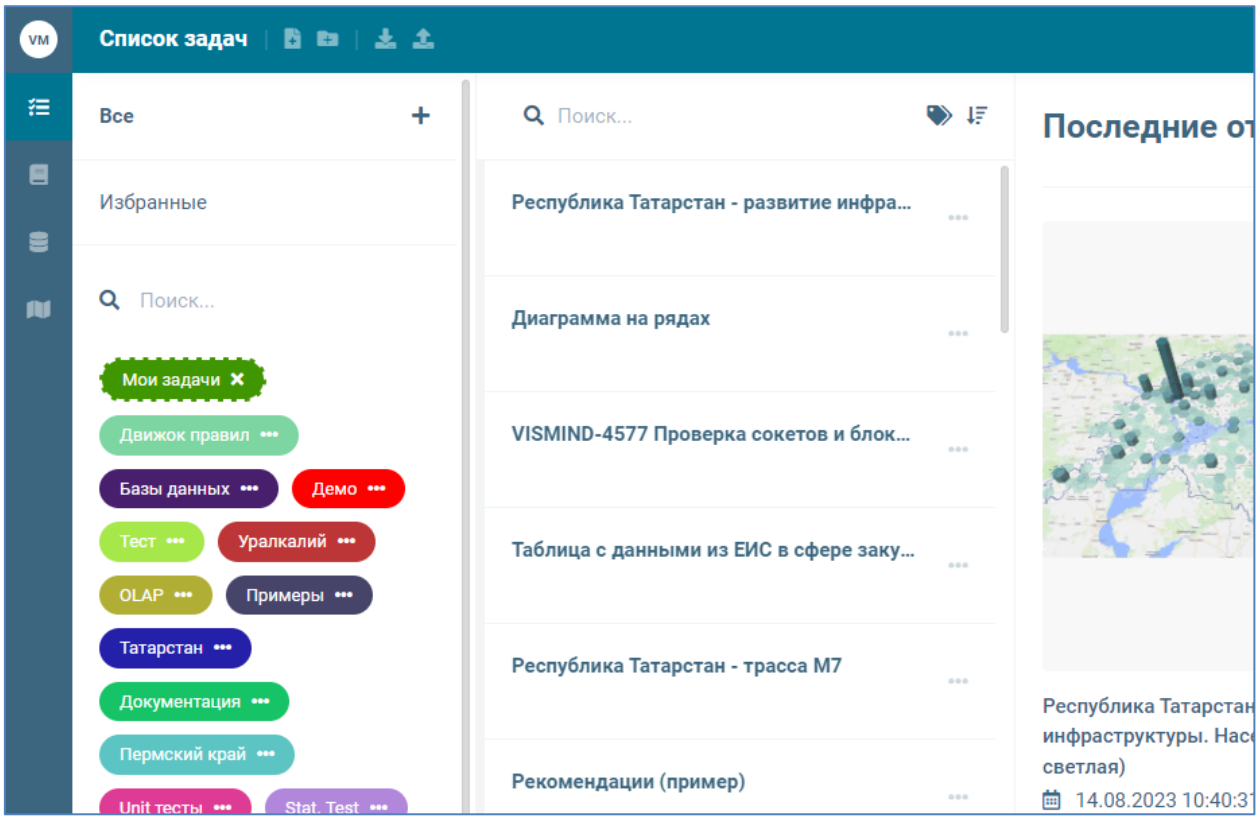


Рисунок 2 – Страница выбора задачи

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

4 Пользовательское приложение

4.1 Пресеты

Пресет – страница проекта (задачи), на которой размещены визуализаторы.

4.1.1 Переключение между пресетами

Существует два способа переключения между пресетами:

1. Через меню «Пресеты» в верхней панели инструментов пресета.

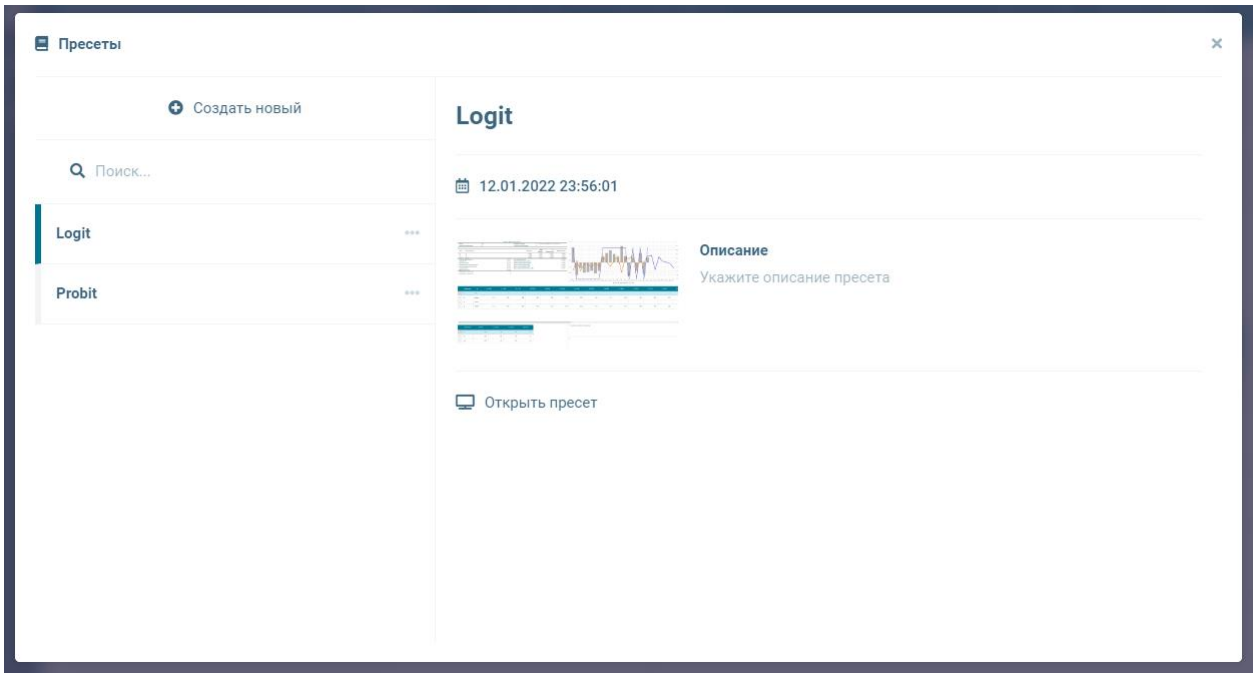


Рисунок 3 – Пример переключения между пресетами через меню

2. По нажатию кнопки. Кнопки, соответствующие пресетам, появляются при добавлении на граф блока «Меню».

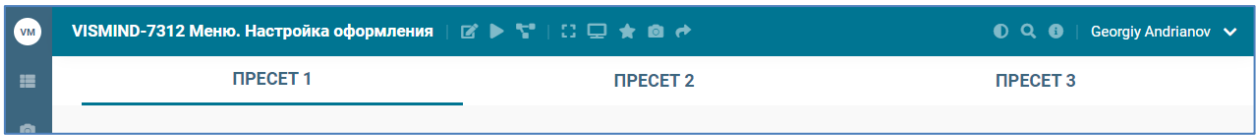


Рисунок 4 – Пример переключения между пресетами через кнопку

4.1.2 Редактирование и просмотр пресета

У пресета есть 2 режима работы:

1. Редактирование пресета. Данный режим позволяет:
    - просматривать библиотеку визуализаторов и «Сохраненные состояния";
    - создавать новые визуализаторы;
    - размещать визуализаторы на пресете;
    - настраивать пресет и визуализаторы;
    - сохранять пресет.
2. Просмотр пресета. Скрывает функционал редактирования для комфортной работы пользователя.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

4.1.2.1 Отображение визуализаторов на пресете

Для того чтобы визуализаторы корректно отображались на пресете при различных разрешениях экрана, а также при изменении размеров окна браузера, используется блок «Сетка».

Помимо блока «Сетка» для обеспечения необходимого отображения визуализаторов есть и другие блоки-контейнеры, такие как:

- Вкладки;
- Выпадающее меню;
- Аккордеон;
- Выдвигающиеся панели;
- Шторка.

Если использовать блоки-контейнеры невозможно по какой-либо причине, то можно настроить у визуализатора привязку к границам экрана. Для этого:

1. Открыть меню визуализатора.
2. Открыть «Настройки внешнего вида».

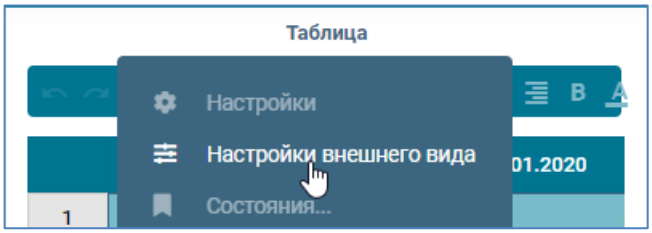


Рисунок 5 – Пример пункта меню «Настройки внешнего вида»

3. Настроить привязку в разделе «Привязка к границам экрана».

Рисунок 6 – Пример раздела «Привязка к границам экрана»

4.1.3 Автообновление пресета

В настройках пресета существует возможность задать автообновление пресета.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

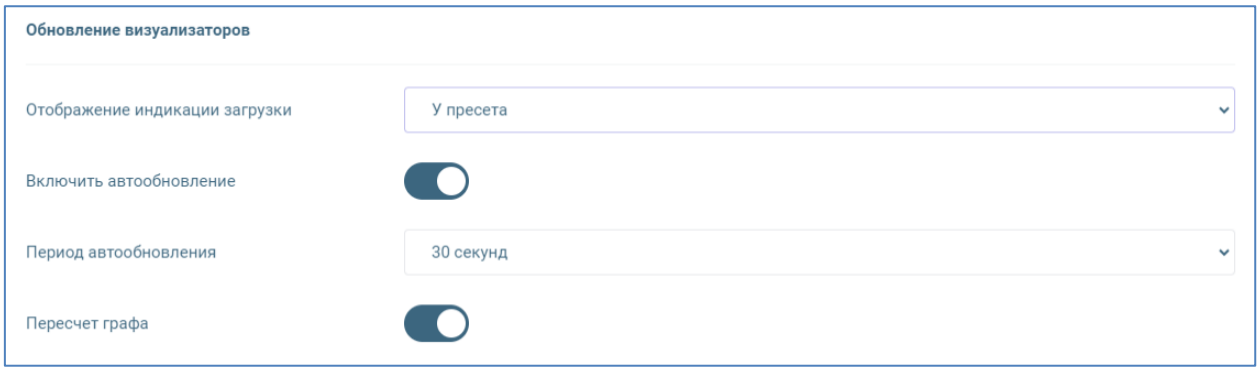


Рисунок 7 – Пример настройки автообновления пресета

4.1.4 Отображение пресета модальным окном

В настройках пресета есть возможность отображать пресет в качестве модального окна в других пресетах (с использованием блока "Пресет"):

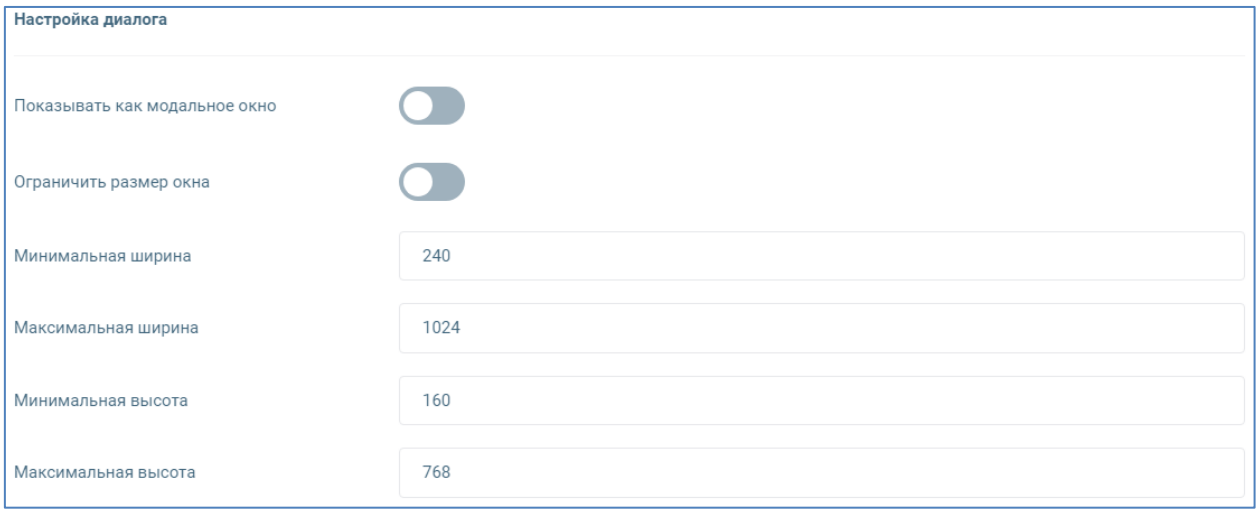


Рисунок 8 – Пример настройки модального отображения пресета

В случае, если один или несколько пресетов на графе будут помечены для отображения как модальное окно, в графе появляется блок "Выходные данные - Задача" с предустановленными входами.

Входы будут группироваться по пресетам, т.е. для каждого пресета, отображаемого в качестве модального окна, будут заданы отдельные выходы.

Изменение входов блока недоступно. Удаление блока произойдет автоматически, если ни один пресет не будет помечен как модальное окно.

4.1.5 Экспорт пресета

Можно сделать снимок пресета, нажав на кнопку в верхней панели инструментов или клавишу F8.

После чего он станет доступен для скачивания в меню «Снимки экрана» слева.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Рисунок 9 – Пример различных типов визуализаторов

4.1.6 Печать визуализаторов пресета

У любого визуализатора (кроме элементов управления, как, например, «Кнопка») есть возможность отправить его на печать.

Для этого нужно открыть меню визуализатора и выбрать пункт «Печать».

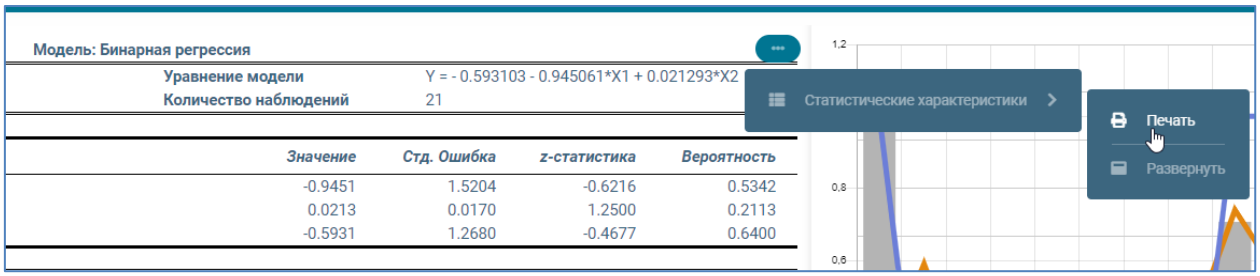


Рисунок 10 – Пример пункта меню «Печать»

4.1.7 Особенности работы визуализаторов, которые меняются после событий

Визуализаторы могут меняться в зависимости от действий пользователя, например, от изменения отметки в выпадающем списке блока «Панель OLAP».

В приложении реализована независимая работа с пресетом, т.е.:

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

- с одним пресетом может работать несколько пользователей, при этом каждый пользователь видит свой вариант пресета, соответствующий его действиям на пресете

4.1.8 Описание работы пресета (в части backend)

При открытии в приложении пресета, запускается его расчет.

Во время расчета:

1. Пользовательские события (изменения отметки в элементах управления) сбрасываются до дефолтных (т.е. визуализаторы отображают то, что было рассчитано при полном расчете графа).
2. Если есть блок «События пресета» (см. п. 6.1.4.19.6), то рассчитывается ветка графа, начиная с данного блока.
3. Если пресет открывается с входными параметрами (в строке запроса есть параметр «param»), то значения из атрибута param передаются в выход блока «События пресета» (п. 6.1.4.19.6). Зависимые блоки рассчитываются с этими параметрами.
    - Например: http://test.int.bittechno.ru:8888/analytics?task=fcae0b86-30b3-462a-bfea-ec12618cd2bd&preset=7d3155e9-8ba4-491a-af4b-429154ef9744&mode=edit&param={"region":[2]}

4.1.9 Открытие пресета (в части backend)

Основные этапы при открытии пресета:

- получение информации по пресету:
    - какие визуализаторы есть на пресете;
    - где они расположены;
- запуск расчета пресета:
    - очищаются события пользователя с прошлого посещения пресета;
    - если переданы параметры, то пресет рассчитывается с этими параметрами;
- получение визуализаторов (на всех визуализаторах индикация загрузки):
    - либо приходят данные визуализатора;
    - либо информация о необходимости ожидания расчета событий других визуализаторов;
- вызов событий визуализаторов, если требуется;
- получение оставшихся визуализаторов, которые ожидали события, если такие были.

4.1.9.1 Короткая ссылка

Короткая ссылка пресета – ссылка на пресет, которая не требует аутентификации пользователя

Пример ссылки на пресет: http://test.int.bittechno.ru:8888/analytics?task=b5adb15b-cf8d-4c46-83c9-b9ac609679b0&preset=28f546e7-9820-4cd6-9ea6-c041df32dff4

ее аналог с короткой ссылкой: http://test.int.bittechno.ru:8888/x?r=t590j

Для получения ссылки необходимо:

1. Нажать на кнопку «Поделиться ссылкой» в верхнем меню пресета.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Рисунок 11 – Пример меню пресета

2. Включить флаг «Опубликовать пресет».
3. Скопировать ссылку.

Рисунок 12 – Пример ссылки для публикации пресета

Можно запросить короткую ссылку для пресета по API: /preset/<preset_id>/short (GET)

При удалении пресета, короткая ссылка удаляется автоматически.

4.1.10 Избранные задачи

В пресете есть возможность перейти на пресет другой задачи, если она является избранной

Для этого:

1. Нажать кнопку «Избранные задачи» в панели инструментов сверху.
2. В появившемся меню нажать на задачу, на которую нужно перейти. Будет открыт первый пресет этой задачи.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

4.2 Визуализаторы

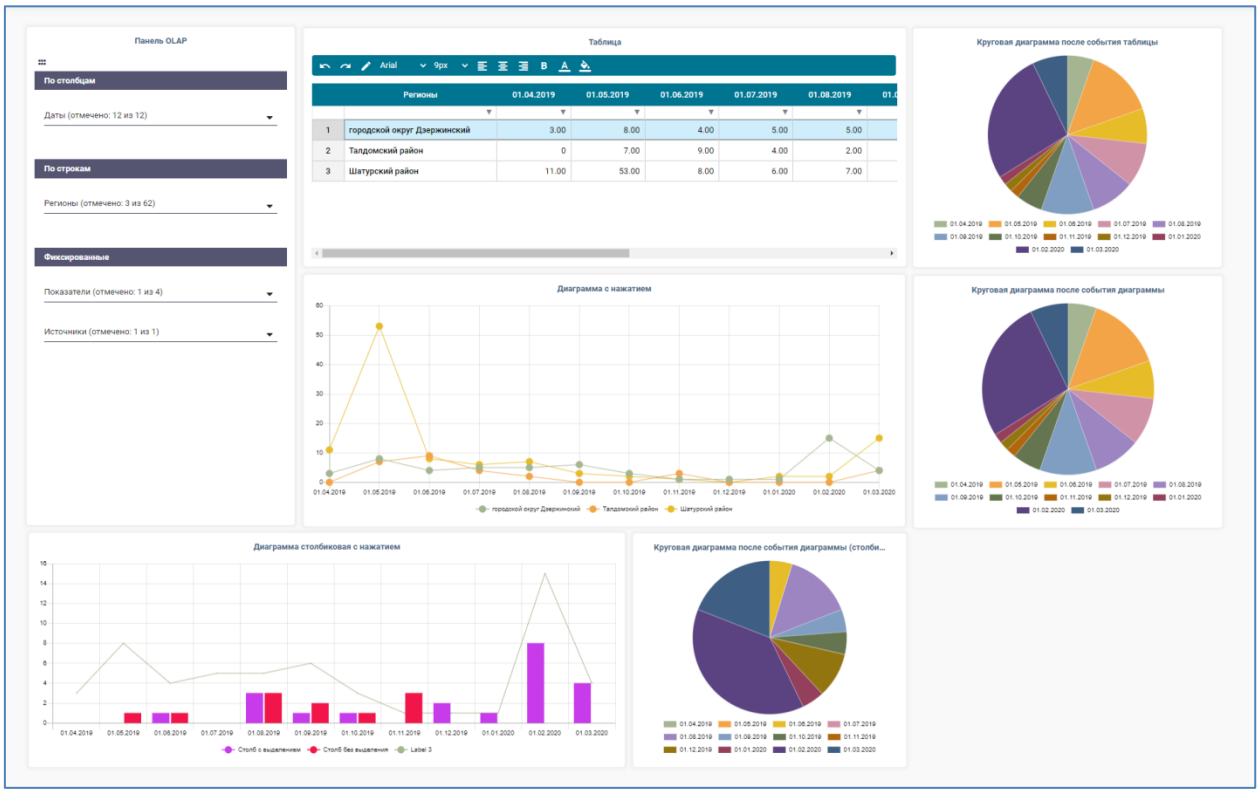
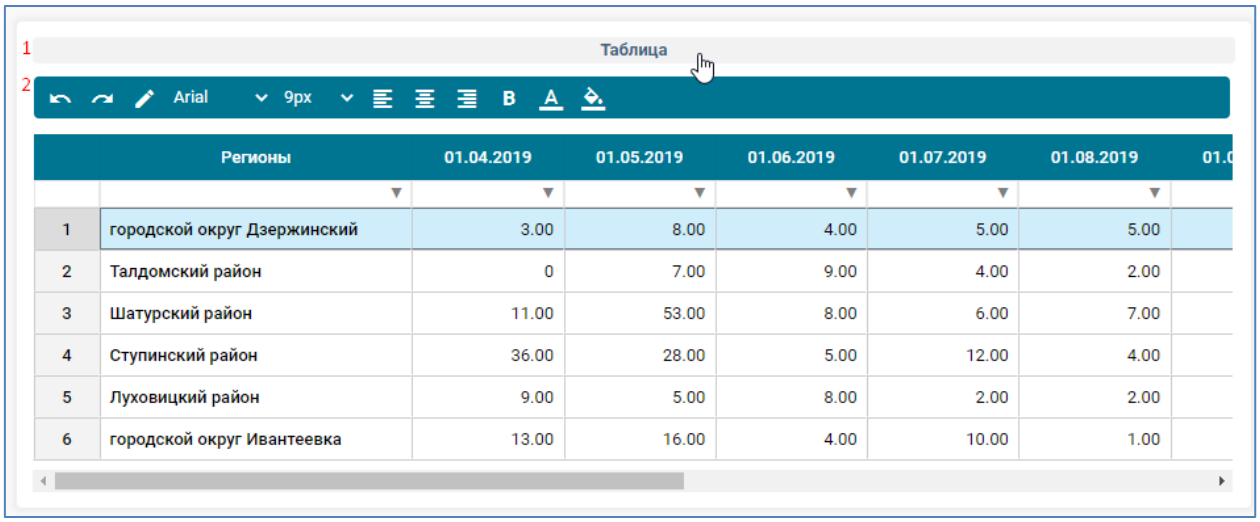


Рисунок 13 – Пример различных типов визуализаторов

Визуализаторы – элементы, которые служат для графического представления данных. Чтобы создать визуализатор, необходимо сначала создать блок на графе, это можно сделать двумя способами: непосредственно на графе (см. п. 7.2), либо с помощью меню «Быстрое добавление визуализатора» в пресете (через «Быстрое добавление визуализатора» возможно добавить только блоки без связей). После того как блок был добавлен на граф, он должен быть рассчитан.

Алгоритм размещения визуализатора на пресете приведен в п. 4.2.9. На одном пресете нельзя разместить несколько раз один и тот же визуализатор, при этом, он может быть размещен на нескольких пресетах.

Пример визуализатора приведен на рисунке.



	Регионы	01.04.2019	01.05.2019	01.06.2019	01.07.2019	01.08.2019	01.0
1	городской округ Дзержинский	3.00	8.00	4.00	5.00	5.00	
2	Талдомский район	0	7.00	9.00	4.00	2.00	
3	Шатурский район	11.00	53.00	8.00	6.00	7.00	
4	Ступинский район	36.00	28.00	5.00	12.00	4.00	
5	Луховицкий район	9.00	5.00	8.00	2.00	2.00	
6	городской округ Ивантеевка	13.00	16.00	4.00	10.00	1.00	

Рисунок 14 – Пример визуализатора

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

1. Заголовок визуализатора. Помимо текста заголовка, в нем также может быть размещена произвольная ссылка (указывается в настройках блока на графе в поле «Ссылка по клику на заголовок визуализатора»).

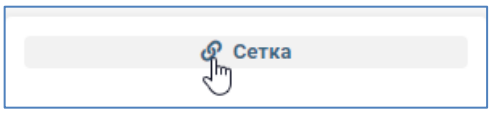


Рисунок 15 – Пример ссылки в заголовке визуализатора

2. Визуализатор.

4.2.1 Меню визуализатора

Способ открытия меню визуализатора зависит от того, включено или нет отображение заголовка визуализатора:

- Если отображение заголовка визуализатора включено, необходимо нажать правой кнопкой мыши (далее – ПКМ) по области заголовка.
- Если отображение заголовка визуализатора выключено, необходимо навести курсор на визуализатор, затем нажать левой кнопкой мыши (далее – ЛКМ) по значку во всплывающем меню.

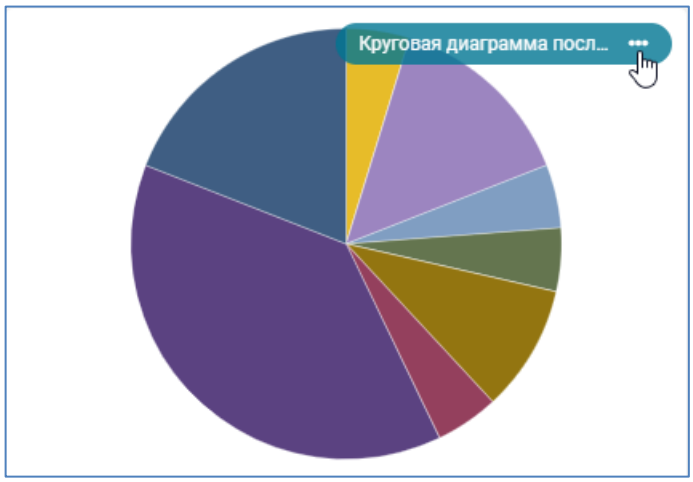


Рисунок 16 – Пример способа открытия меню визуализатора

Пункты меню визуализатора зависят от режима пресета и от типа визуализатора.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

4.2.1.1 Меню визуализатора в случае вложенности

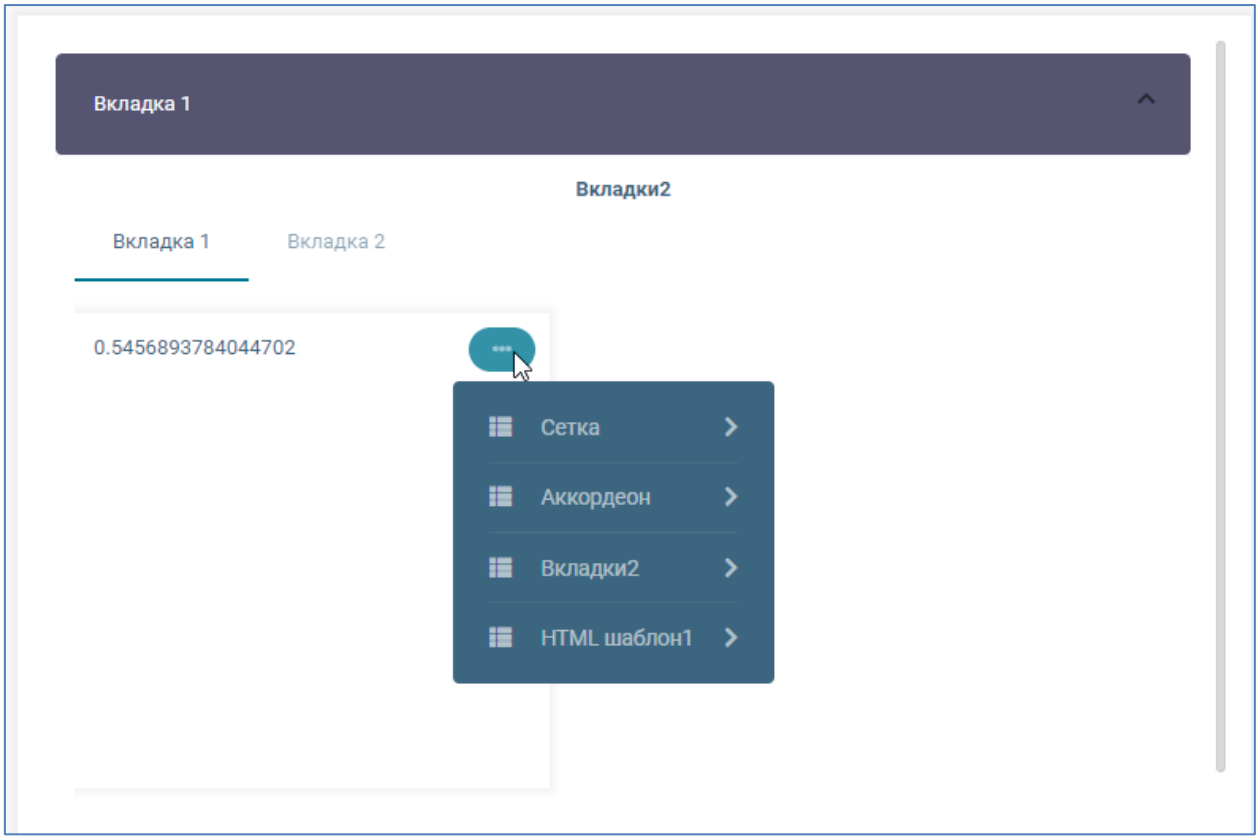


Рисунок 17 – Пример меню визуализатора в случае вложенности

При размещении визуализатора внутри визуализаторов-контейнеров: «Сетка», «Вкладки», «Аккордеон», открыв меню текущего визуализатора, пользователю доступны меню данного визуализатора и всех визуализаторов вверх по иерархии.

4.2.1.2 Общие настройки в режиме просмотра пресета

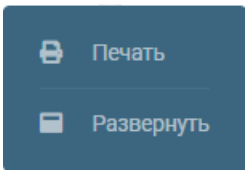


Рисунок 18 – Пример меню в режиме просмотра пресета

1. **Печать –** открывает в новой вкладке предпросмотр визуализатора для печати.
2. **Развернуть –** открывает визуализатор в отдельном окне на весь экран.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

4.2.1.3 Общие настройки в режиме редактирования пресета

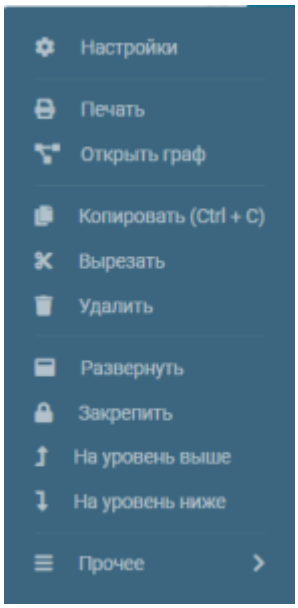


Рисунок 19 – Пример меню в режиме редактирования пресета

1. **Настройки –** открывает настройки внешнего вида визуализатора (см. п. 4.2.7).
2. **Печать –** открывает в новой вкладке предпросмотр визуализатора для печати.
3. **Открыть граф –** открывает в новой вкладке граф. Блок, которому соответствует данный визуализатор, будет расположен по центру рабочей области и также выделен.
4. **Копировать (Ctrl + C) –** копирует визуализатор. Используется для вставки на другой пресет.
5. **Вырезать –** вырезает визуализатор.
6. **Удалить –** удаляет визуализатор.
7. **Развернуть –** разворачивает визуализатор на весь экран. Одновременно в пресете на весь экран может быть развернут только один визуализатор.
8. **Закрепить –** запрещает перемещать визуализатор в пресете.
9. **На уровень выше –** увеличивает z-индекс визуализатора. Позволяет разместить визуализатор над другими визуализаторами.
10. **На уровень ниже –** уменьшает z-индекс визуализатора. Позволяет разместить визуализатор под другими визуализаторами.
11. **Прочее > Скопировать GUID –** копирует идентификатор визуализатора. Используется для формирования ссылок на визуализатор и для отладки.

Специфические настройки визуализаторов указаны в описании соответствующих блоков.

4.2.2 Особенности размещения визуализаторов на пресете

Способы размещения:

1. Стандартный. Визуализатор размещен непосредственно на пресете.
   - Размер и положение визуализатора регулирует пользователь вручную.
   - В настройках внешнего вида визуализатора в разделе «Привязка к границам экрана» может быть указана зависимость расположения визуализатора относительно границ окна браузера.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

- С помощью пунктов меню визуализатора «На уровень выше», «На уровень ниже» пользователь может располагать визуализаторы друг над другом.

2. Внутри визуализаторов-контейнеров. Визуализатор размещен внутри блоков:
   - «Сетка» (см. п. 6.1.6.5);
   - «Вкладки» (см. п. 6.1.6.2);
   - «Аккордеон» (см. п. 6.1.6.1).

   Поведение визуализаторов описано в документации к данным блокам.

4.2.3 Состояния визуализаторов

Для части визуализаторов есть возможность создать для них «Состояние». Для таких визуализаторов в меню отображается раздел «Состояния».

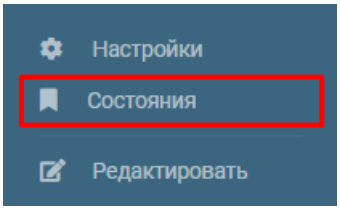


Рисунок 20 – Пример раздела меню «Состояния»

Подробное описание механизма работы приведено в п. 4.2.11 «Сохраненные визуализаторы и управление состояниями».

4.2.4 Экспорт визуализаторов

Для визуализаторов блока «Таблица» (п. 6.1.2.3) существует возможность экспорта в файлы формата xlsx и ods.

4.2.5 Быстрое добавление визуализатора

Некоторые визуализаторы можно добавить непосредственно в пресете, используя функцию быстрого добавления визуализатора.

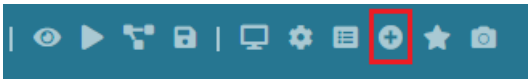


Рисунок 21 – Пример функции быстрого добавления визуализатора

Список визуализаторов с возможностью быстрого добавления:

- Блок «Меню» (см. п. 6.1.4.7);
- Блок «HTML-статика» (см. п. 6.1.4.1);
- Блок «Сетка» (см. п. 6.1.6.5);
- Блок «Вкладки» (см. п. 6.1.6.2);
- Блок «Аккордеон» (см. п. 6.1.6.1);
- Блок «Панель размещения и фильтрации» (см. п. 6.1.4.10);
- Блок «Таблица» (см. п. 6.1.2.3);
- Блок «Диаграмма» (см. п. 6.1.1.2).

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

1. На первом шаге необходимо выбрать визуализатор для размещения (для примера будет использован блок «HTML статика»):

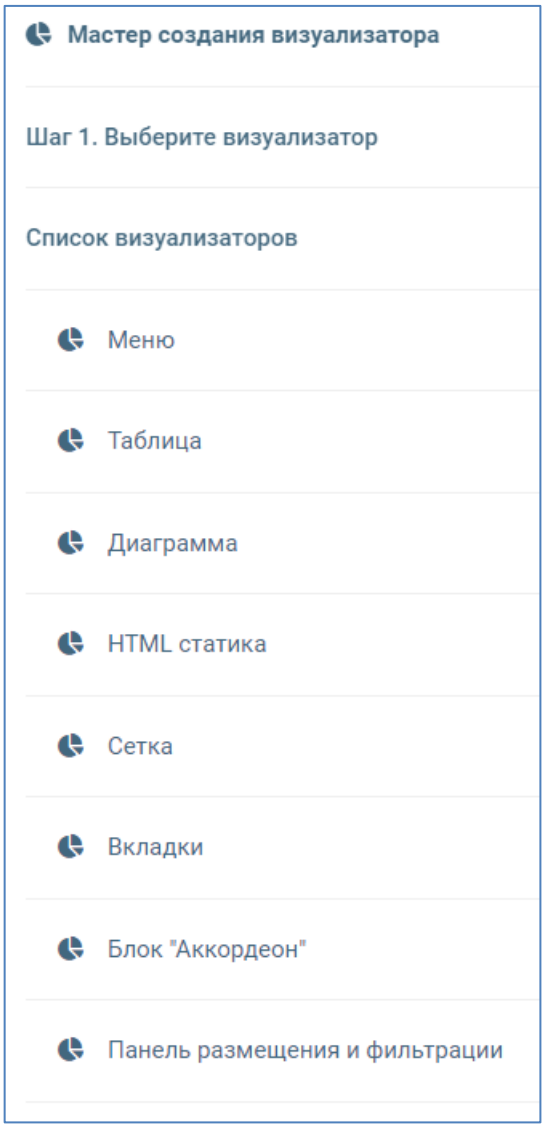


Рисунок 22 – Пример мастера создания визуализатора (шаг 1)

Блоки «Панель размещения и фильтрации», «Таблица» и «Диаграмма» имеют более сложный механизм размещения (см. п. 4.2.6).

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

2. На втором шаге необходимо заполнить настройки блока:

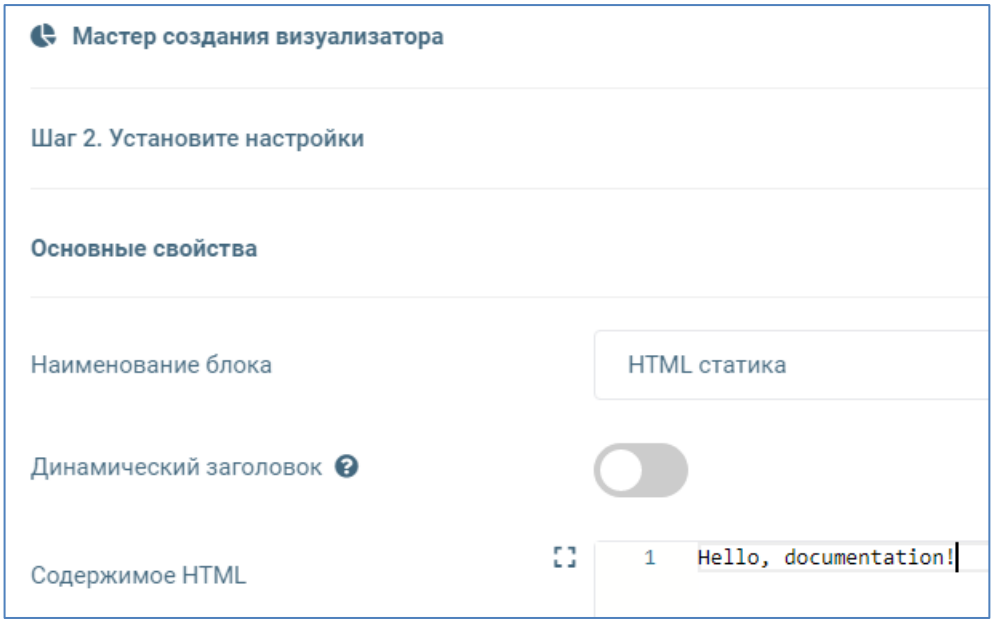


Рисунок 23 – Пример мастера создания визуализатора (шаг 2)

3. После чего блок появится в списке визуализаторов:

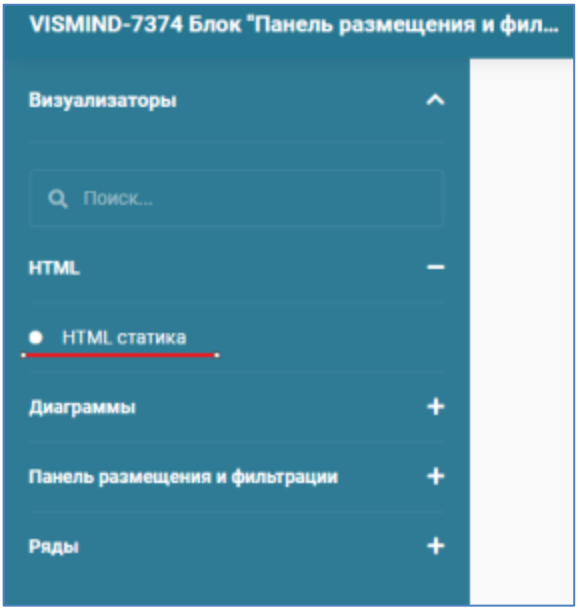


Рисунок 24 – Пример отображения добавленного блока в списке визуализаторов

4.2.6 Добавление визуализаторов вместе с привязкой к данным

Блоки «Панель размещения и фильтрации», «Таблица» и «Диаграмма» имеют более сложный мастер размещения, т.к. помимо самого визуализатора так же добавляется и источник данных (см. п. 5.1.5), из которого и будут браться данные для отображения в визуализаторе.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

4.2.6.1 Блок «Панель размещения и фильтрации»

В случае добавления блока «Панель размещения и фильтрации» следующим шагом после выбора визуализатора будет:

1. Выбор источника данных из всех, доступных в приложении.

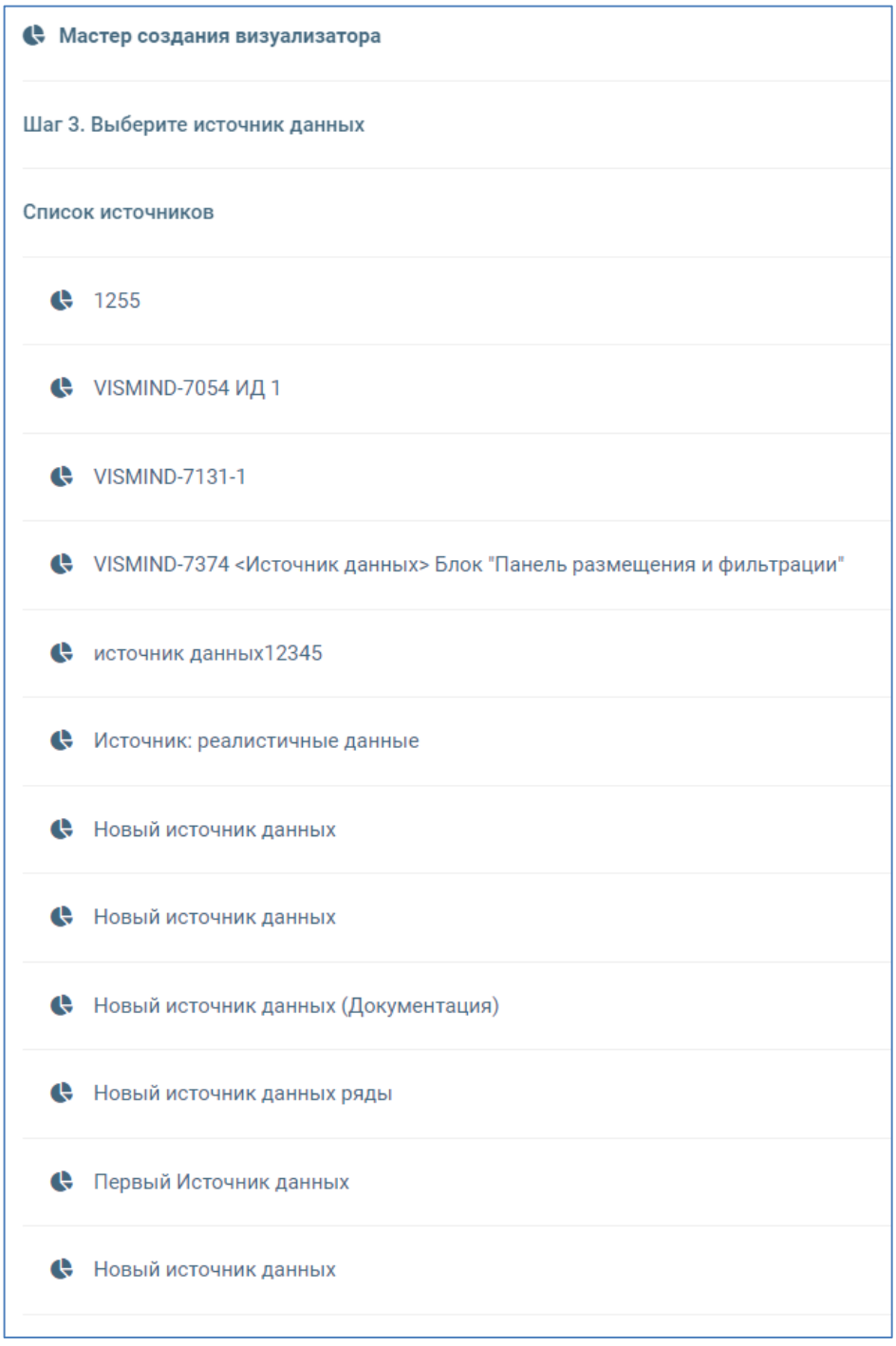


Рисунок 25 – Пример мастера создания визуализатора (шаг 3) для добавления блока «Панель размещения и фильтрации»

2. Следующий шаг, как и в случае с другими визуализаторами – настройка блока.

Для корректного создания связей со справочниками в параметрах привязки (см. п. 6.1.4.10) необходимо указывать поле **dict** с идентификатором нужного справочника.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

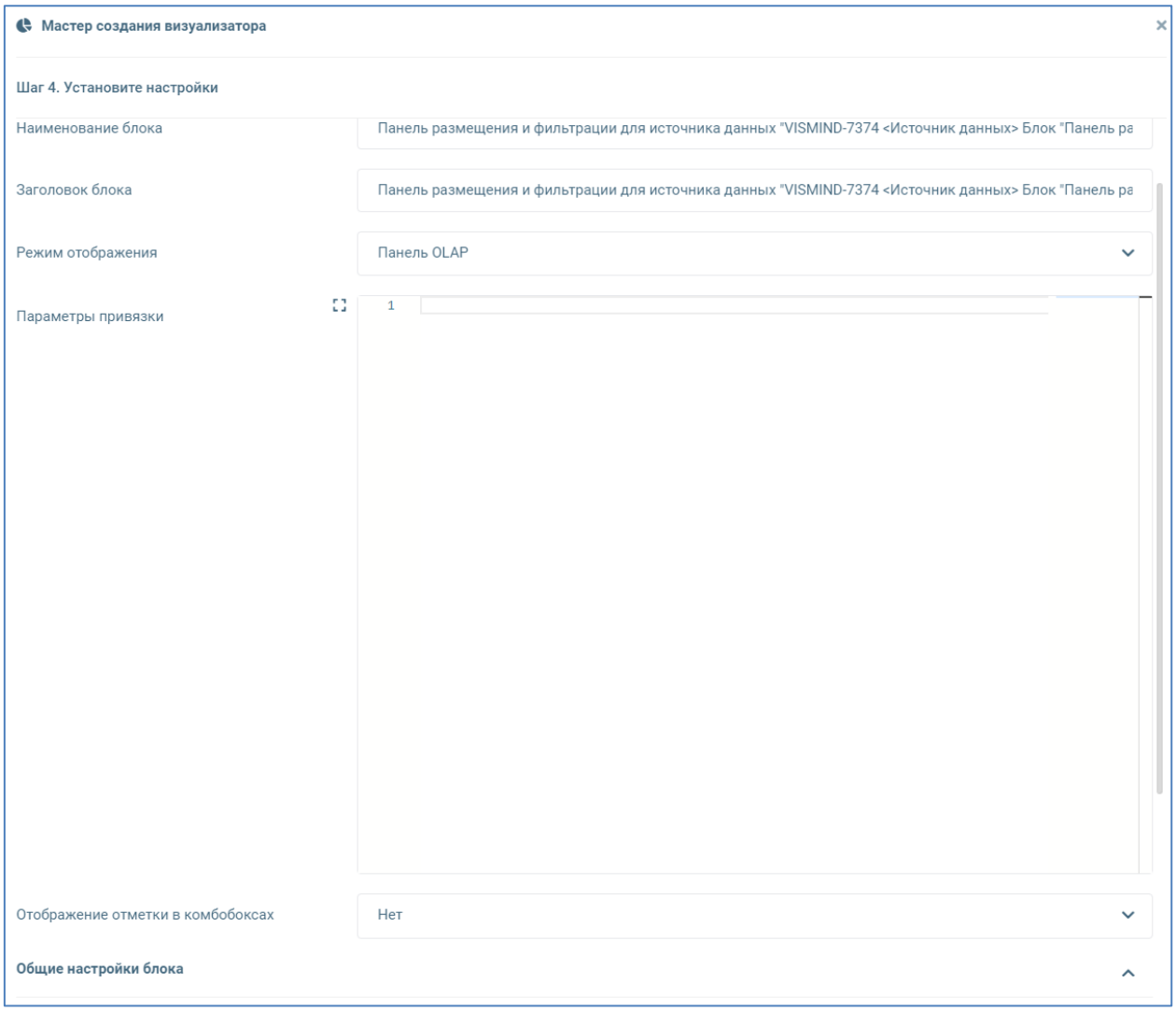


Рисунок 26 – Пример мастера создания визуализатора (шаг 4)

3. Последним шагом является сопоставление входов источника данных (левый столбец) и справочников панели фильтрации (правый столбец, выпадающий список).

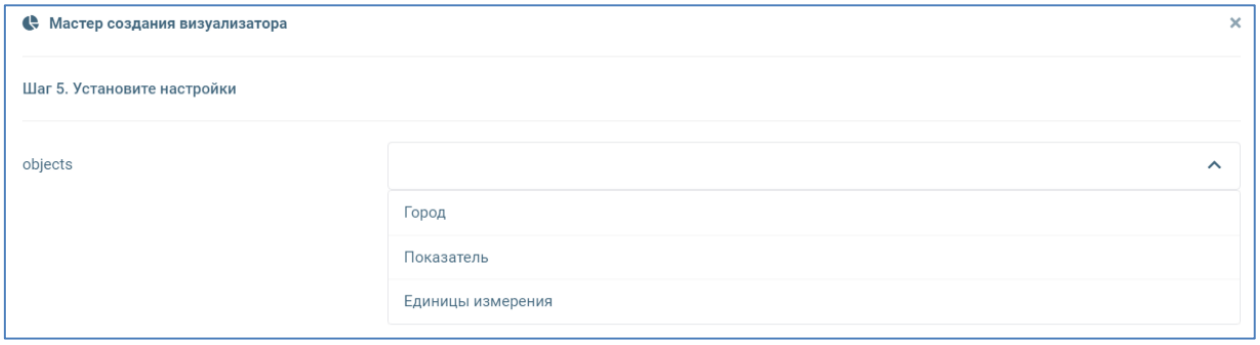


Рисунок 27 – Пример мастера создания визуализатора (шаг 5)

4.2.6.2 Блоки «Таблица» или «Диаграмма»

В случае добавления блоков «Таблица» или «Диаграмма» первым шагом будет выбор источника данных. Однако, в отличие от шагов при добавлении блока «Панель размещения и фильтрации», в мастере можно выбрать как и источник данных (тогда первые шаги будут аналогичны при добавлении панели), так и уже созданный блок-источник данных (с привязанной панелью размещения и фильтрации).

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

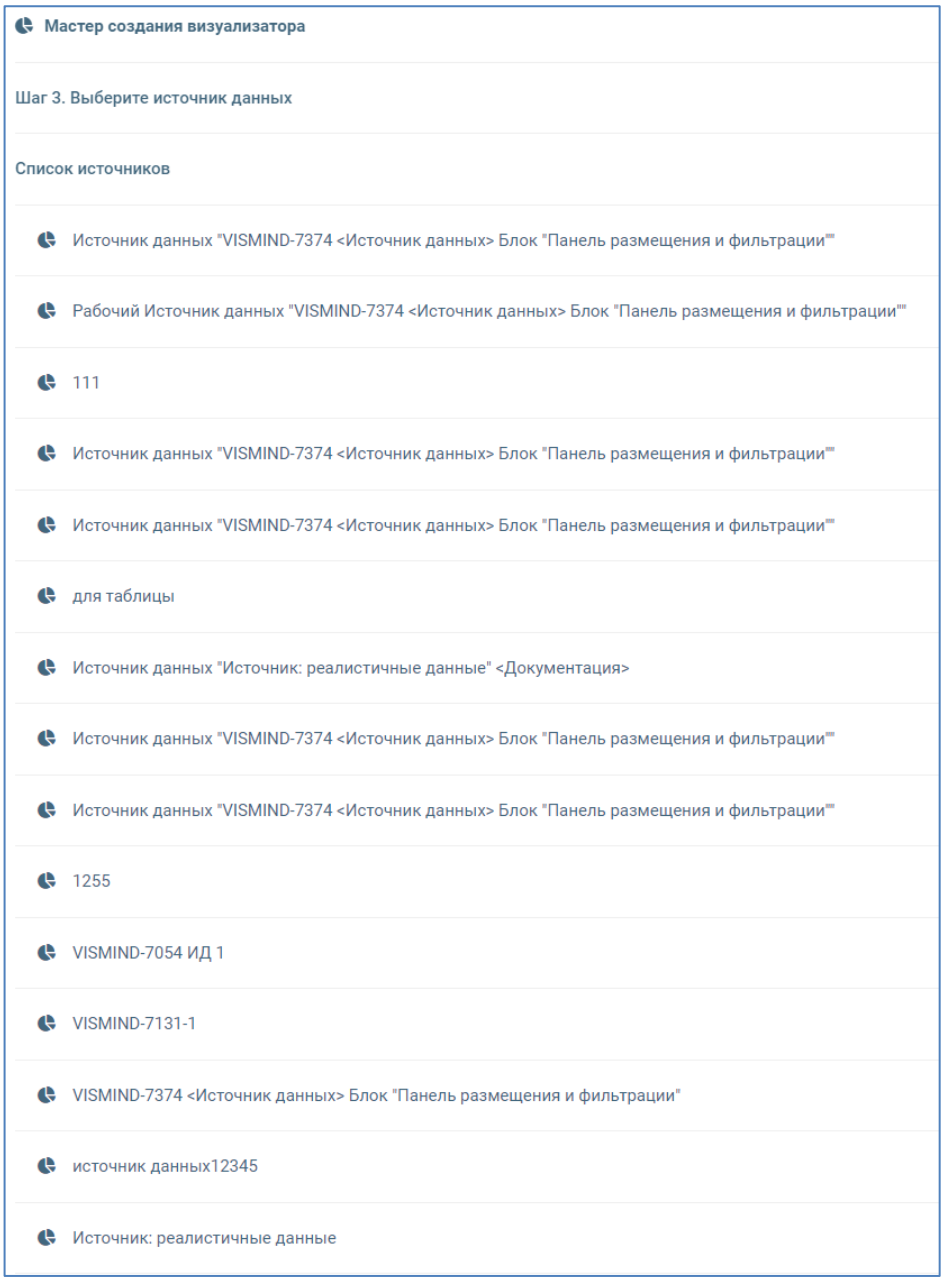


Рисунок 28 – Пример мастера создания визуализатора (шаг 3) для добавления блоков «Таблица» или «Диаграмма»

После создания или выбора источника данных появляется последний шаг, в котором необходимо настроить вид данных будущего визуализатора (аналогично блоку OLAP).

Для корректного отображения визуализатора после завершения работы мастера рекомендуется использовать блок «Панель размещения и фильтрации" (см. п. 6.1.4.10) в режиме отображения «Панель OLAP" (см. п. 6.1.4.8).

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

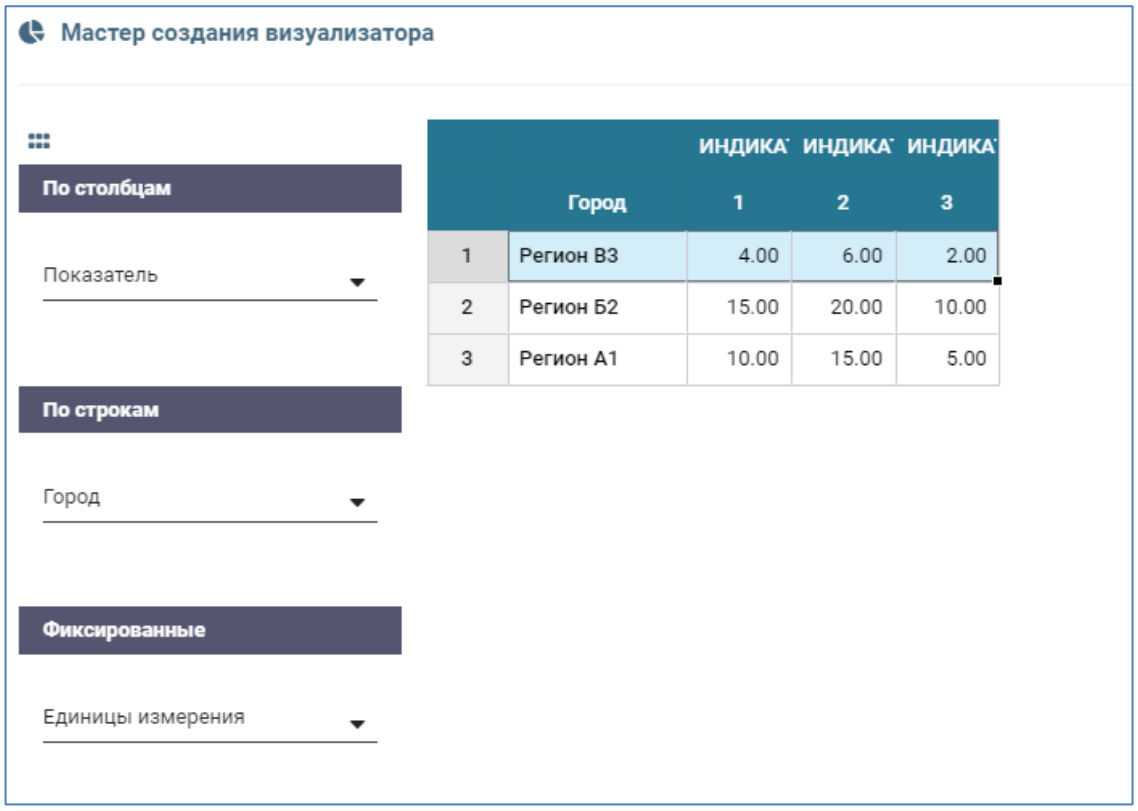


Рисунок 29 – Пример мастера создания визуализатора в режиме отображения «Панель OLAP»

4.2.7 Настройки блока визуализатора

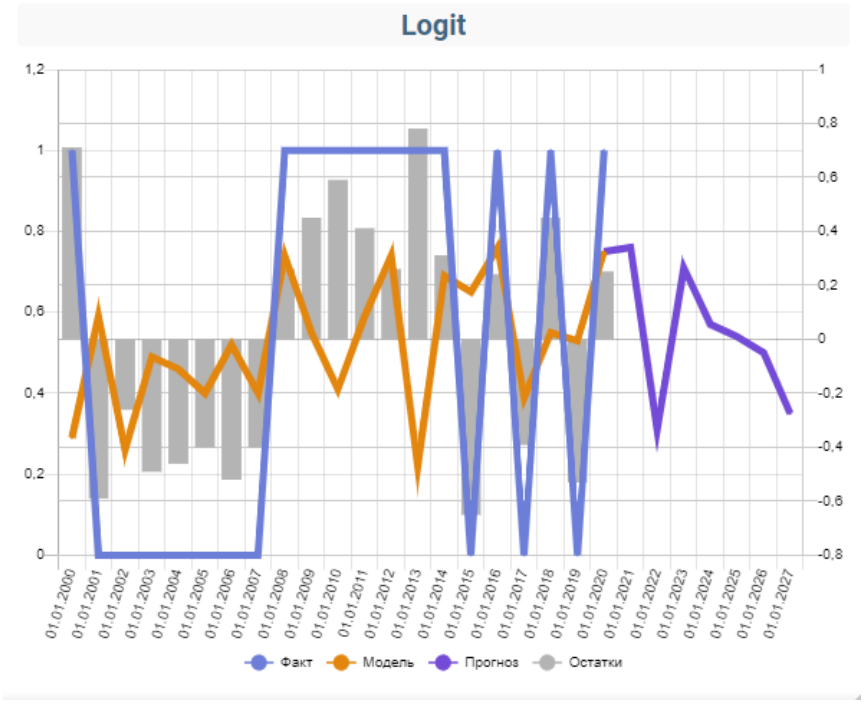
Через меню «Настройки» пользователь может открыть настройки блока визуализатора без необходимости заходить в граф. Функционал доступен только в режиме редактирования.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

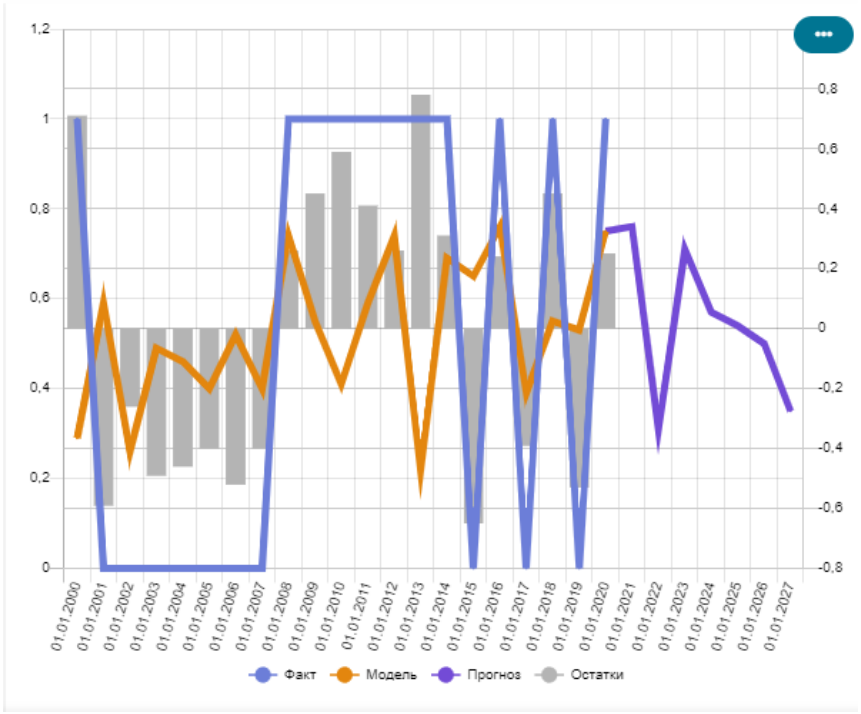
4.2.8 Настройки визуализатора

4.2.8.1 Вызов настроек визуализатора

Способ вызова окна настроек визуализатора на пресете зависит от настройки отображения заголовка.

Настройка заголовка	Описание способа вызова настроек	Пример
Отображать заголовок: **Активно**	В этом случае, для вызова настроек требуется навести курсор на визуализатор и нажать ПКМ по заголовку визуализатора	На примере визуализатора «Диаграмма» 

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Настройка заголовка	Описание способа вызова настроек	Пример
Отображать заголовок: **Не активно**	В этом случае, для вызова настроек требуется навести курсор на визуализатор и нажать ЛКМ по появившейся специфической кнопке	На примере визуализатора «Диаграмма» 

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

4.2.8.2 Настройки блока визуализатора

Через меню «Настройки» пользователь может открыть настройки блока визуализатора без необходимости заходить в граф. Функционал доступен только в режиме редактирования.

4.2.8.3 Окно настроек визуализатора

После вызова появляется окно настроек визуализатора. По умолчанию, раскрывается первая группа настроек «Общие», все остальные свёрнуты.

Пример настройки визуализатора «Таблица» приведен на рисунке (Рисунок 30).

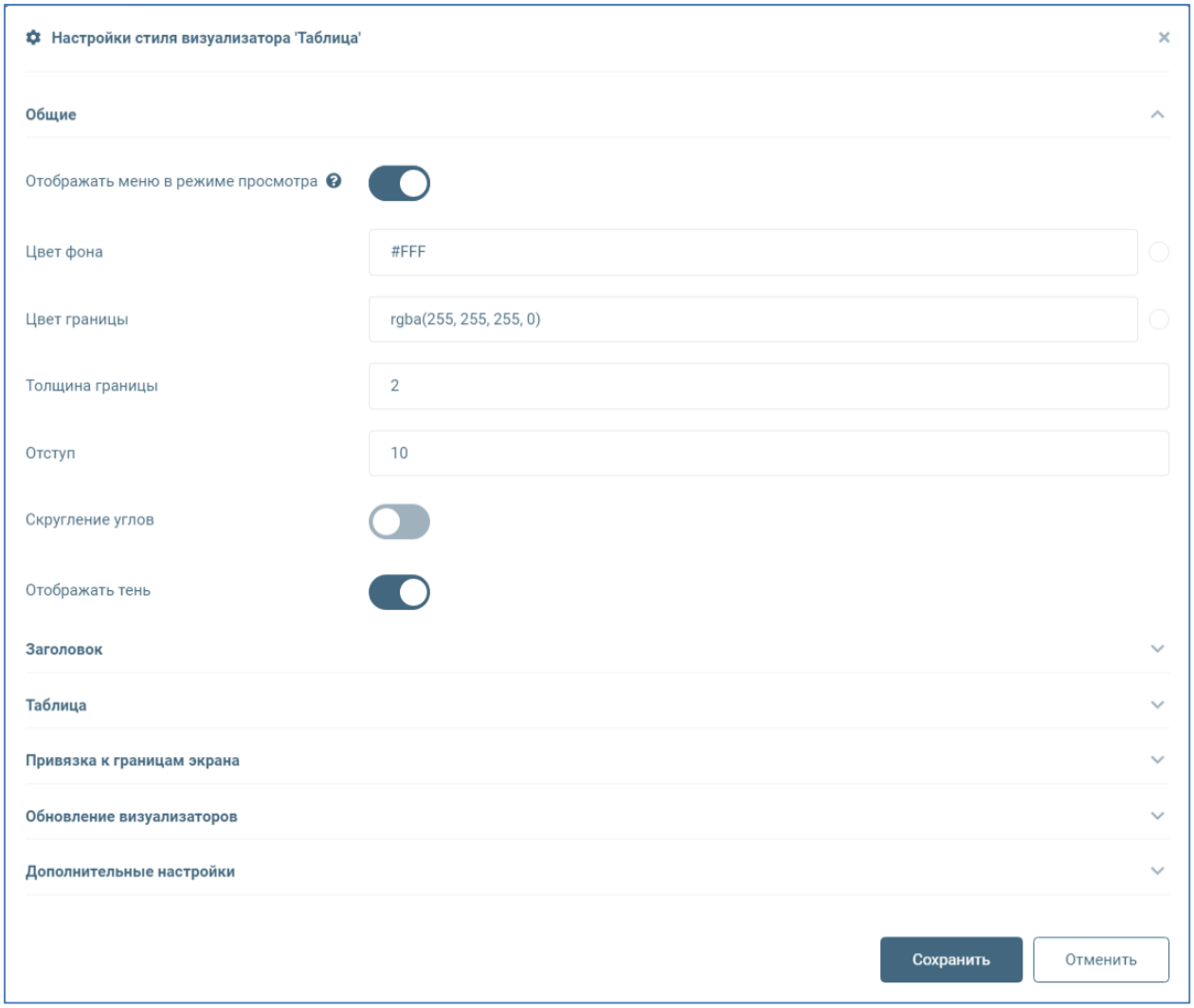


Рисунок 30 – Настройки визуализатора «Таблица» (группа «Общие»)

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

4.2.8.4 Общие настройки визуализатора

Название параметра	Тип параметра	Обязательный	Описание	Пример
Группа «Общие»				
Отображать меню в режиме просмотра	Флаг	Да	Задает возможность отключения системного меню для визуализаторов в режиме просмотра пресета	
Размер текста	Целое число	Нет	Только для визуализаторов, содержащих текст. Задает размер текста в визуализаторе. Значение по умолчанию: **13**	
Начертание	Выпадающий список	Нет	Только для визуализаторов, содержащих текст. Задает начертание текста в визуализаторе. Значение по умолчанию: **Обычный**	
Цвет текста	Цвет	Нет	Только для визуализаторов, содержащих текст. Задает цвет текста в визуализаторе. Значение по умолчанию: **#3c667f**	
Цвет фона	Цвет	Нет	Задает цвет фона визуализатора. Значение по умолчанию: **#FFFFFF**	
Цвет границы	Цвет	Нет	Задает цвет границы визуализатора. Значение по умолчанию: **rgba(255, 255, 255, 0)**	
Толщина границы	Вещественное число	Нет	Задает толщину границы визуализатора. Значение по умолчанию: **2**	
Отступ	Целое число	Нет	Задает отступ визуализатора от границы. Значение по умолчанию: **100**	
Скругление углов	Флаг	Нет	Включает и отключает углов окна визуализатора. Значение по умолчанию: **Не активно**	
Отображать тень	Флаг	Нет	Включает и отключает «тень» у окна визуализатора. Значение по умолчанию: **Активно**	
Группа «Заголовок»				
Важно! Если визуализатор является Компонентом, у него отсутствует раздел «Заголовок»				
Отображать заголовок	Флаг	Нет	Включает и отключает отображение заголовка окна визуализатора. Значение по умолчанию: **Активно**	
Размер текста заголовка	Целое число	Нет	Задает размер текста заголовка. Значение по умолчанию: **13**	
Цвет текста заголовка	Цвет	Нет	Задает цвет заголовка. Значение по умолчанию: **#3c667f**	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
Тип заголовка визуализатора	Выпадающий список	Нет	Определяет тип заголовка визуализатора. Возможные значения: • **Задать вручную** – возможность задать заголовок визуализатора сразу на пресете. • **Заголовок из графа** – формирование заголовка осуществляется из графа (используется поле «**Заголовок визуализатора**»). Значение по умолчанию: **Заголовок из графа**	
Заголовок визуализатора	Строка	Нет	Текст заголовка визуализатора. Поле появляется при выборе значения «**Задать вручную**» в настройке «**Тип заголовка визуализатора**». Значение по умолчанию отсутствует	
Выравнивание	Выпадающий список	Да	Задает выравнивание заголовка «По левому краю» «По центру» – выбран по умолчанию «По правому краю»	
Группа «Привязка к границам экрана»				
Левая граница	Чекбокс со слайдером	Нет	Регулирует привязку окна визуализатора к левой границе окна. По умолчанию: **Не активно**	
Правая граница	Чекбокс со слайдером	Нет	Регулирует привязку окна визуализатора к правой границе окна. По умолчанию: **Не активно**	
Верхняя граница	Чекбокс со слайдером	Нет	Регулирует привязку окна визуализатора к верхней границе окна. По умолчанию: **Не активно**	
Нижняя граница	Чекбокс со слайдером	Нет	Регулирует привязку окна визуализатора к нижней границе окна. По умолчанию: **Не активно**	
Группа «Обновление визуализатора»				
Отображение индикации загрузки	Флаг	Нет	Включает и отключает отображение индикатора загрузки визуализатора при его обновлении. Значение по умолчанию: **Активно**	
Включить автообновление	Флаг	Нет	Включает и отключает автообновление визуализатора. Значение по умолчанию: **Не активно**	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
Период автообновления	Выпадающий список	Нет	Задает период обновления визуализатора. Значение по умолчанию: **Не активно**	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

4.2.8.5 Обновление визуализаторов на пресете

Функция позволяет автоматически обновлять визуализаторы с определенной периодичностью. Подразумевается включение обновления всех визуализаторов на пресете, а также можно включить обновления у отдельных визуализаторов.

Если опция задана в общих настройках пресета – для каждого визуализатора она неактивна с указанием, что она задана в общих настройках.
Если была активна опция автообновления у визуализатора, визуально она остается активной, но отключается с пояснением что «Автообновление задано в общих настройках» (Рисунок 31).

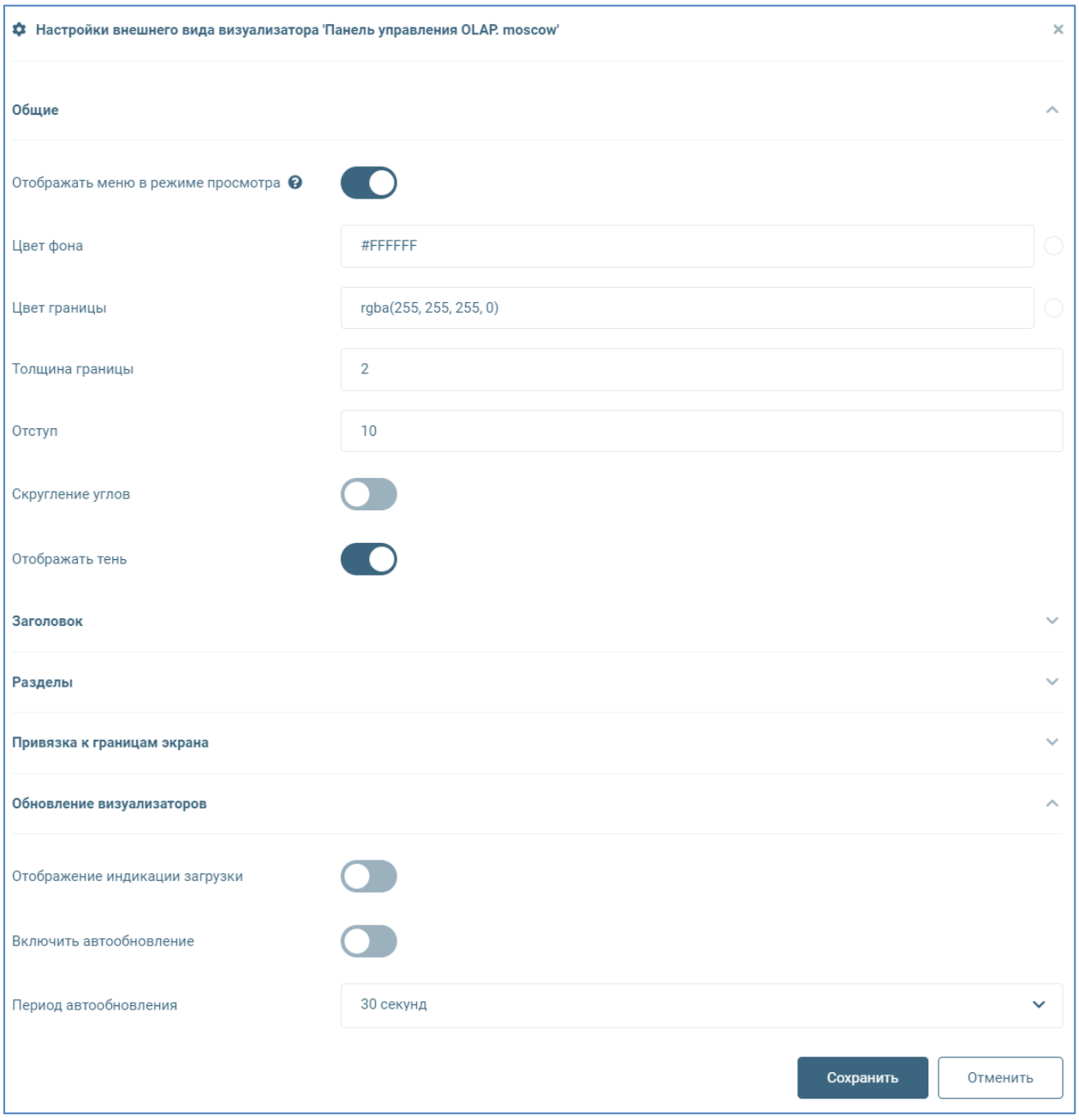


Рисунок 31 – Пример настройки визуализатора (группа «Обновление визуализаторов»)

При установке таймаута автообновления конкретный визуализатор или все визуализаторы одновременно должны обновляться с указанной периодичностью.

Это не должно быть переоткрытие пресета, т.к. оно вызывает перерасчет графа на бэкенде.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

4.2.8.6 Дополнительные настройки

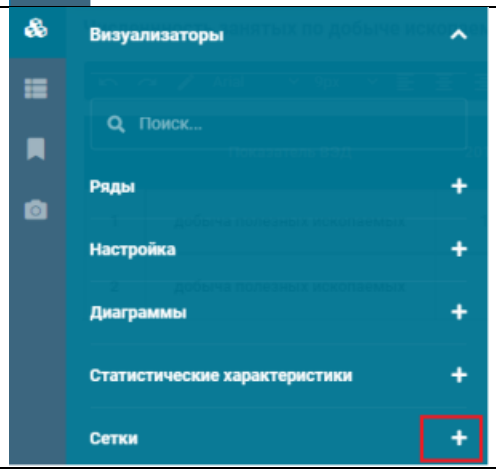
Некоторые визуализаторы имеют дополнительную секцию настроек, с дополнительными настройками. Например, блок «Таблица» (см. п. 6.1.2.3).

Эти настройки позволяют в режиме редактирования настроить некоторые аспекты отображения визуализаторов, при наличии, и сохранить их как часть стиля визуализатора при сохранении пресета.

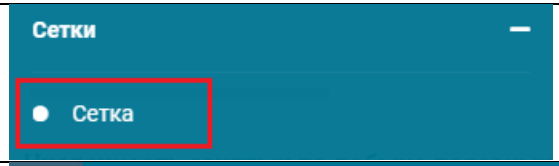
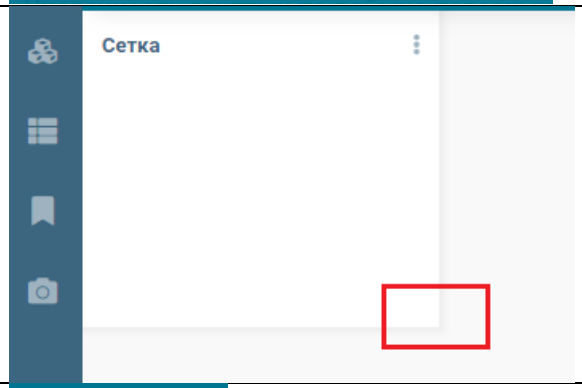
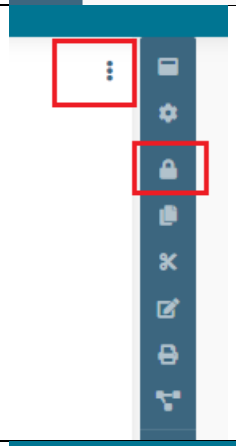
© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

4.2.9 Размещение визуализатора

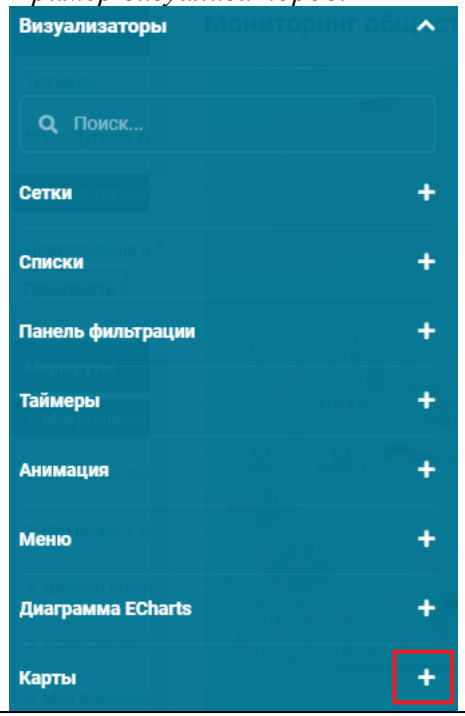
4.2.9.1 Создание пресета из готовых блоков пользовательского приложения

№	Действие	Скриншот
1.	Открыть раздел «Редактирование пресета» на верхней панели управления	
2.	Открыть панель «Визуализаторы» в левом боковом меню	
3.	Выбрать раздел «Сетки», раскрыть его, нажав на значок «+» (*В различных задачах количество и качество разделов визуализации будут различны в зависимости от функционала задачи)*	

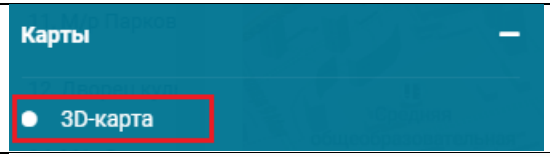
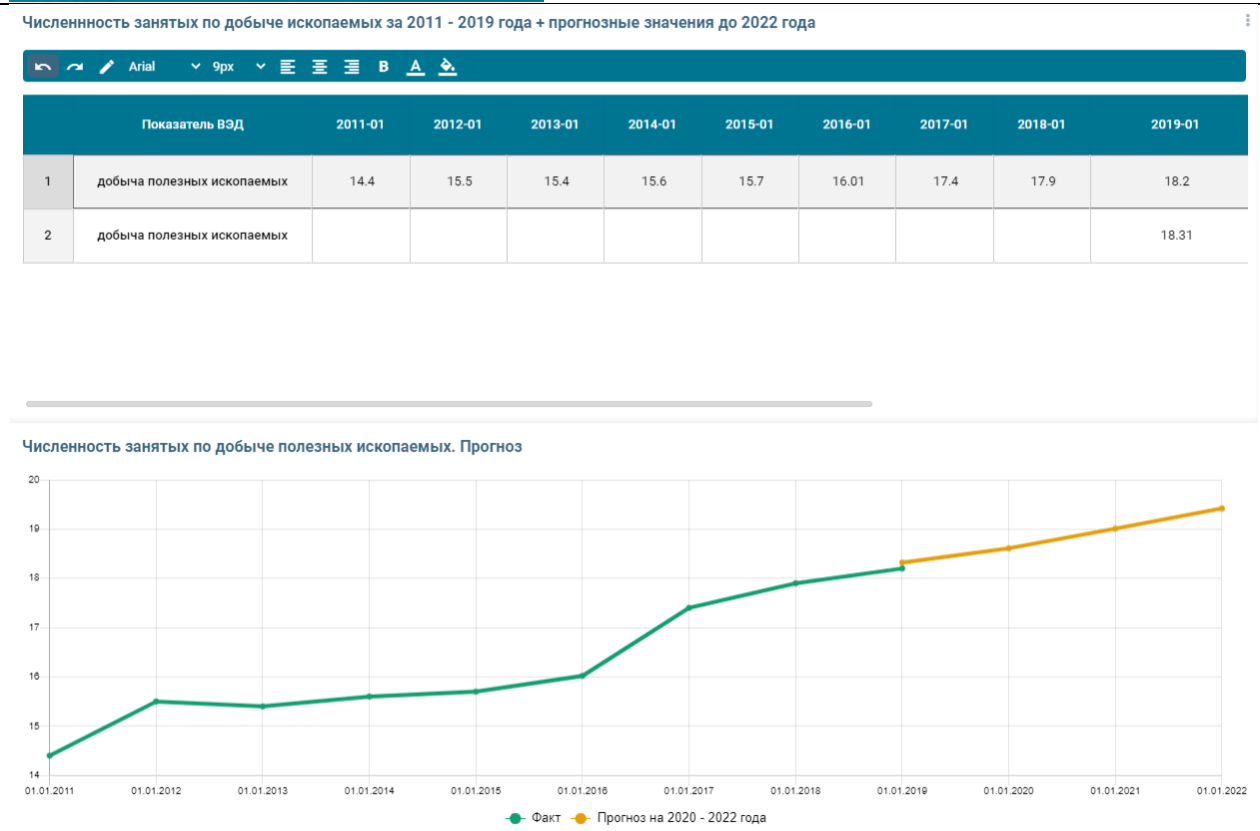
© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

№	Действие	Скриншот
4.	Нажать левой кнопкой мыши на визуализатор «Сетка» и, зажимая кнопку, перетащить на рабочую область справа	
5.	Растянуть «Сетку» за края визуализатора на всю рабочую область	
6.	Нажать на значок ⋮ в правом верхнем углу «Сетки» и закрепить ее, нажав на значок «замок» в выпадающем меню	
7.	Нажать «Сохранить пресет» на верхней панели управления	

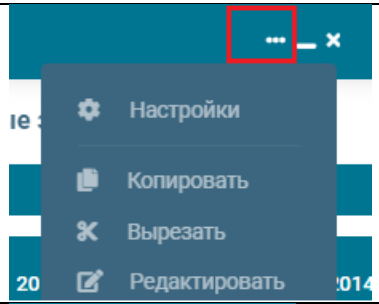
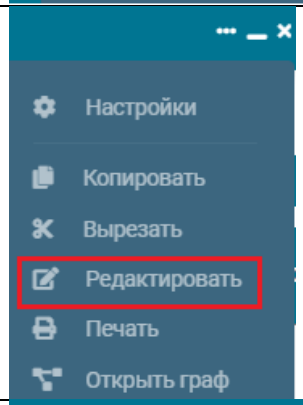
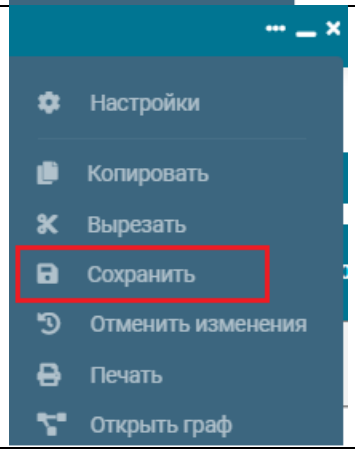
© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

№	Действие	Скриншот
8.	Открыть панель «Визуализаторы» в боковом меню	
9.	Выбрать необходимый блок визуализации из списка Раскрыть нужный блок визуализации, нажав на значок «+» (*В различных задачах количество и качество разделов визуализации будут различны в зависимости от функционала задачи)*	*Пример визуализаторов:* 

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

№	Действие	Скриншот
10.	В раскрытом блоке найти нужный визуализатор и, зажав левую кнопку мыши, перетащить его на рабочую область справа	
11.	Повторить пункты 8 – 10 необходимое количество раз. Визуализаторы автоматически распределятся по сетке, занимая одинаковое положение в пространстве	
12.	После добавления визуализаторов необходимо сохранить пресет, нажав на значок «Сохранить пресет» на верхней панели управления	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

№	Действие	Скриншот
13.	Для изменения расположения визуализаторов необходимо навести курсором на правый верхний угол сетки и открыть выпадающее меню, нажав на значок •••	
14.	Перейти в раздел редактирования сетки, нажав соответствующую кнопку «Редактировать» в выпадающем меню. Изменить положения визуализаторов: перетащить визуализатор, зажав левую кнопку мыши	
15.	Сохранить изменения, нажав соответствующую кнопку в выпадающем меню сетки	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

4.2.9.1.1 Сетка и якоря

4.2.9.1.1.1 Якоря

Якоря позволяют привязывать визуализаторы к границам пресета.

В форме настройки стиля визуализатора предусмотрен чекбокс (флаг) для включения привязки к каждой границе, а также настройка интенсивности привязки границам.

При установке значения на 100% происходит жесткая привязка к грани.

При установке значения менее 100% происходит перемещение по принципу работы **win forms** – компонент смещается только на X% от грани.

Пример меню якорей приведен на рисунке (Рисунок 32).

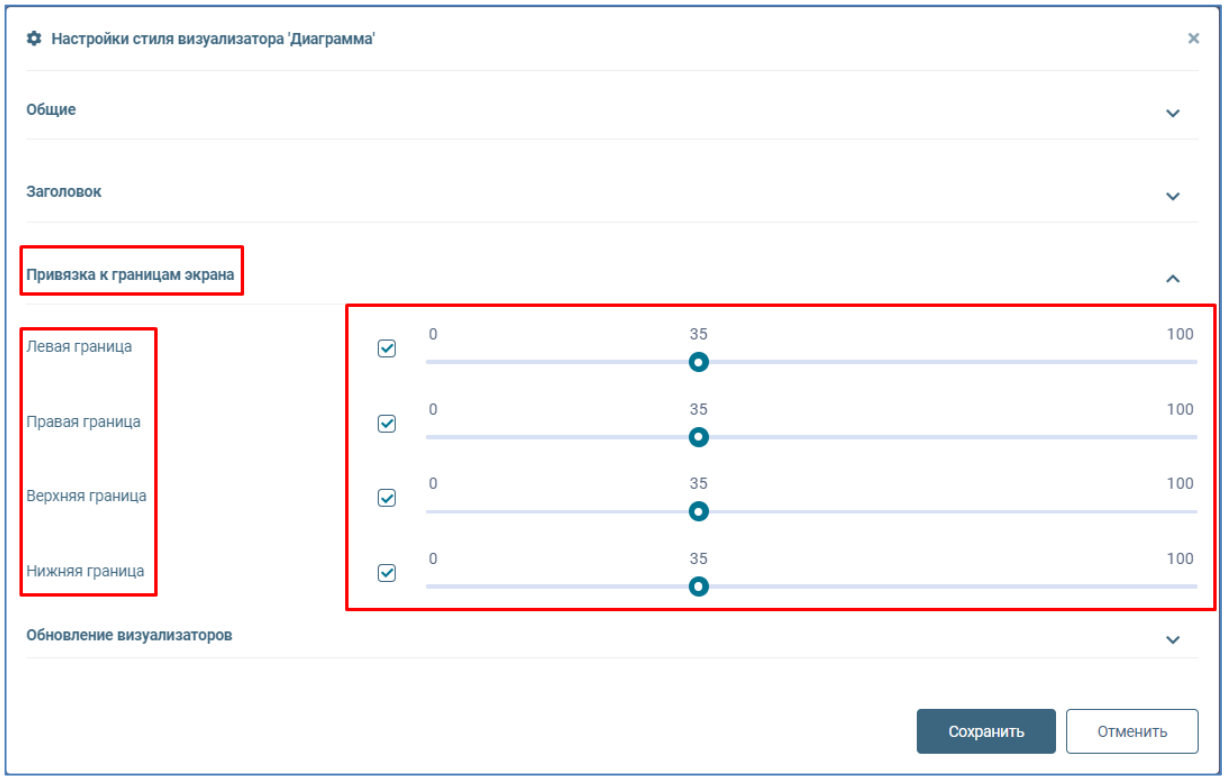


Рисунок 32 – Пример настройки привязки к границам экрана

4.2.10 Сообщения об ошибках в визуализаторах

В некоторых случаях визуализаторы не могут быть отображены, при этом визуализатор будет содержать предупреждающий индикатор и сообщение о причине, по которой не отображается.

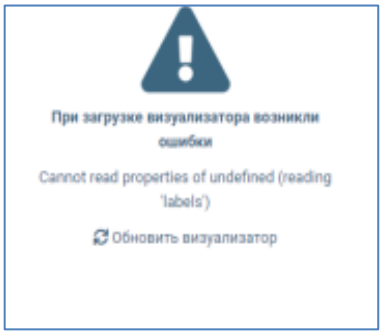


Рисунок 33 – Пример индикатора об ошибке

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Причины, по которым визуализаторы не отображаются

- Блок на графе рассчитан с ошибкой:

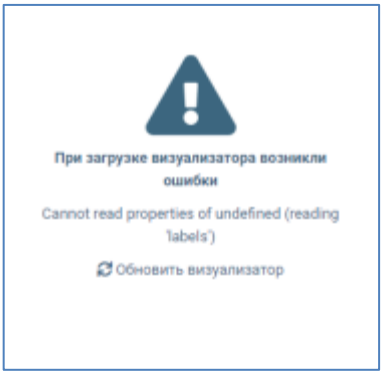


Рисунок 34 – Пример индикатора об ошибке расчета графа

- Блок на графе не был рассчитан или ему на вход не пришли данные. Например: если во время расчета блоков для визуализатора, произошла ошибка, то блок визуализатора не посчитается и выведется пустым.

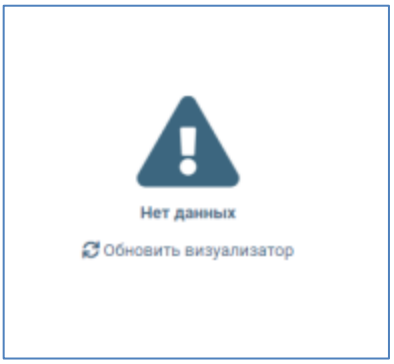


Рисунок 35 – Пример индикатора об ошибке с данными

Взаимодействие пользователя с визуализатором

Пользователь может обновить визуализатор, нажав на кнопку «Обновить визуализатор":

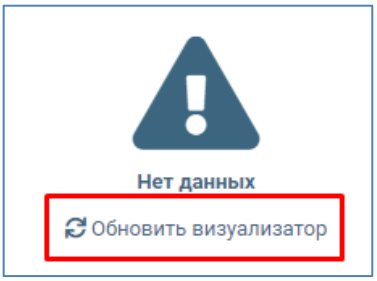


Рисунок 36 – Пример кнопки «Обновить визуализатор»

4.2.11 Сохраненные визуализаторы и управление состояниями

Для визуализатора можно сохранить его состояние. При этом оно становится доступно в отдельной панели приложения «Сохраненные состояния», и его можно переиспользовать в других пресетах.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

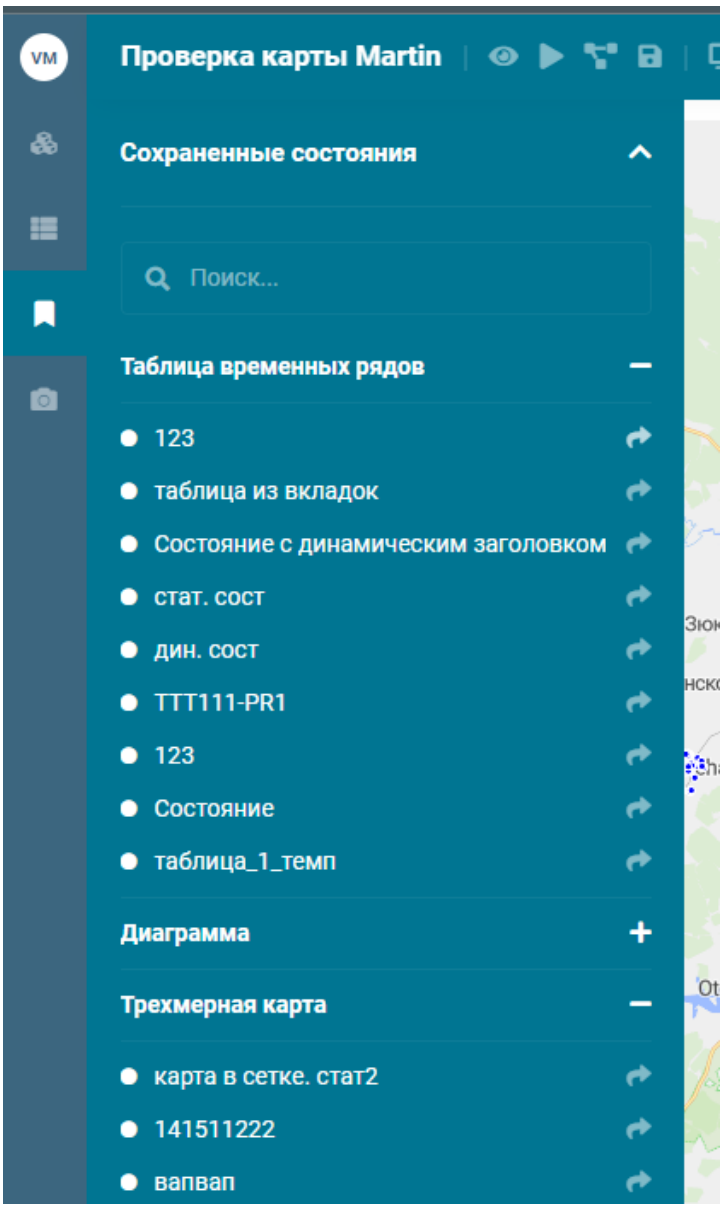


Рисунок 37 – Пример панели приложения «Сохраненные состояния»

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

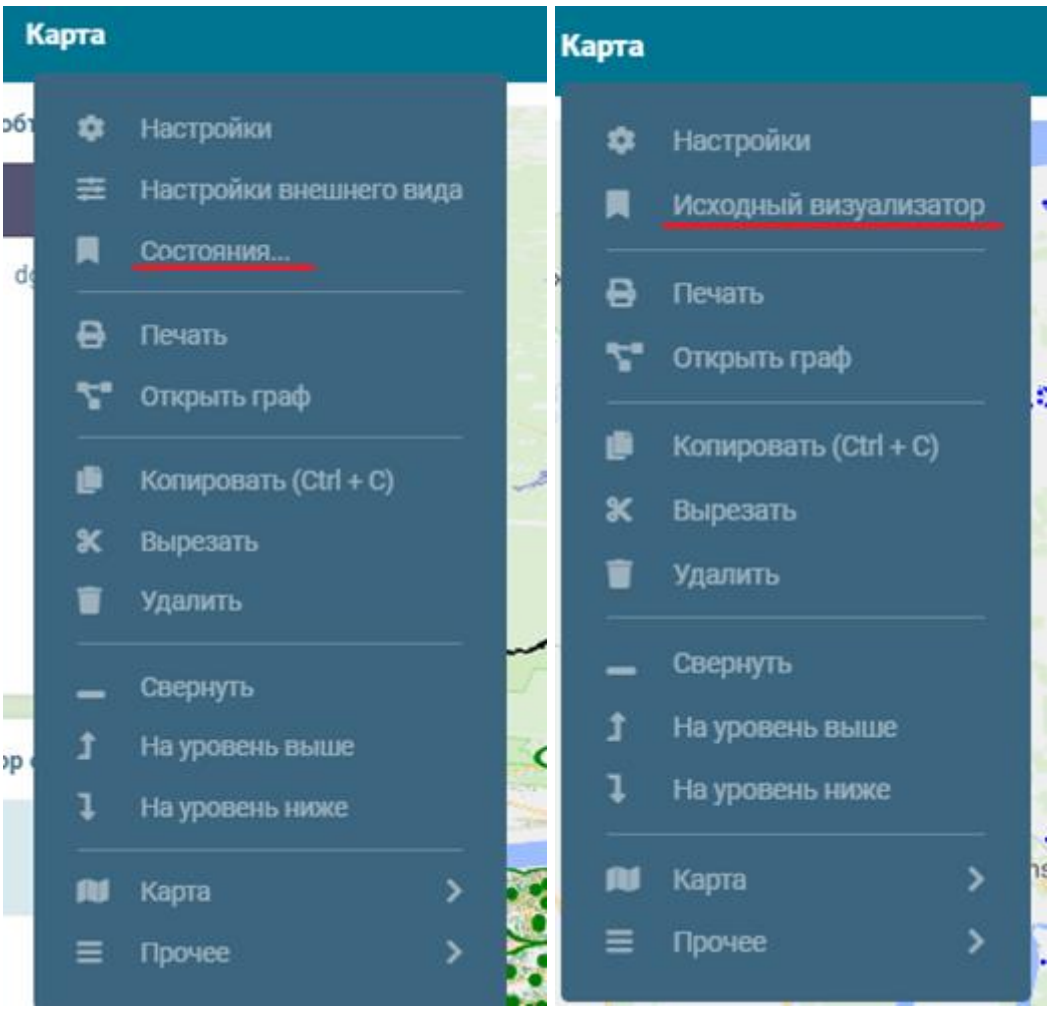


Рисунок 38 – Пример перехода на исходный визуализатор

Для состояния можно перейти на оригинальный визуализатор, нажав на иконку стрелки в списке или, если визуализатор добавлен на рабочую область – через пункт меню «Исходный визуализатор». При этом у визуализатора на пресете, если он является состоянием другого, вместо управления состояниями будет переход на оригинальный визуализатор.

Для управления состояниями у визуализатора предназначено специальное меню «Состояния».

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

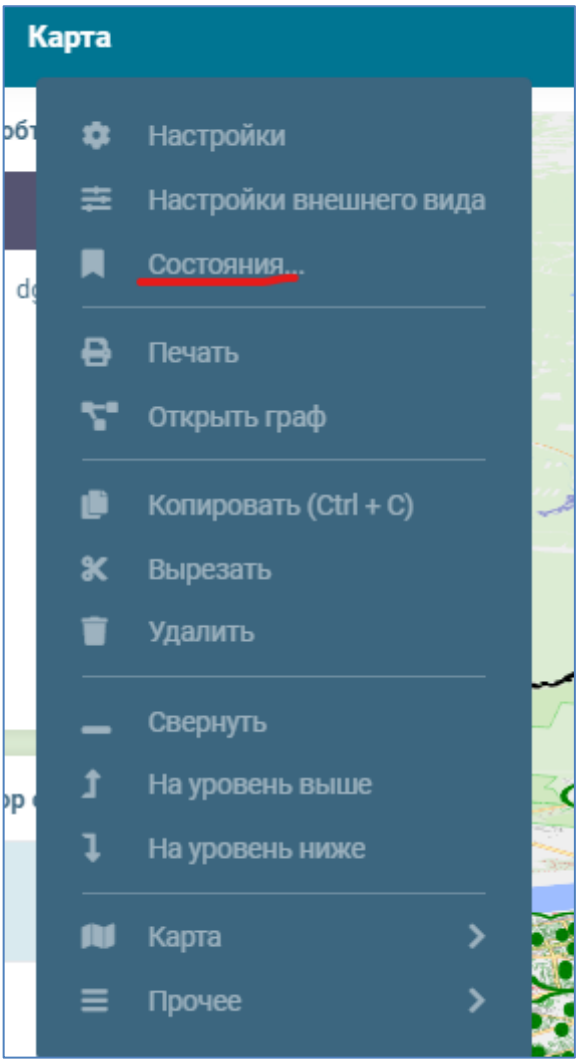


Рисунок 39 – Пример перехода в меню «Состояния визуализатора»

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

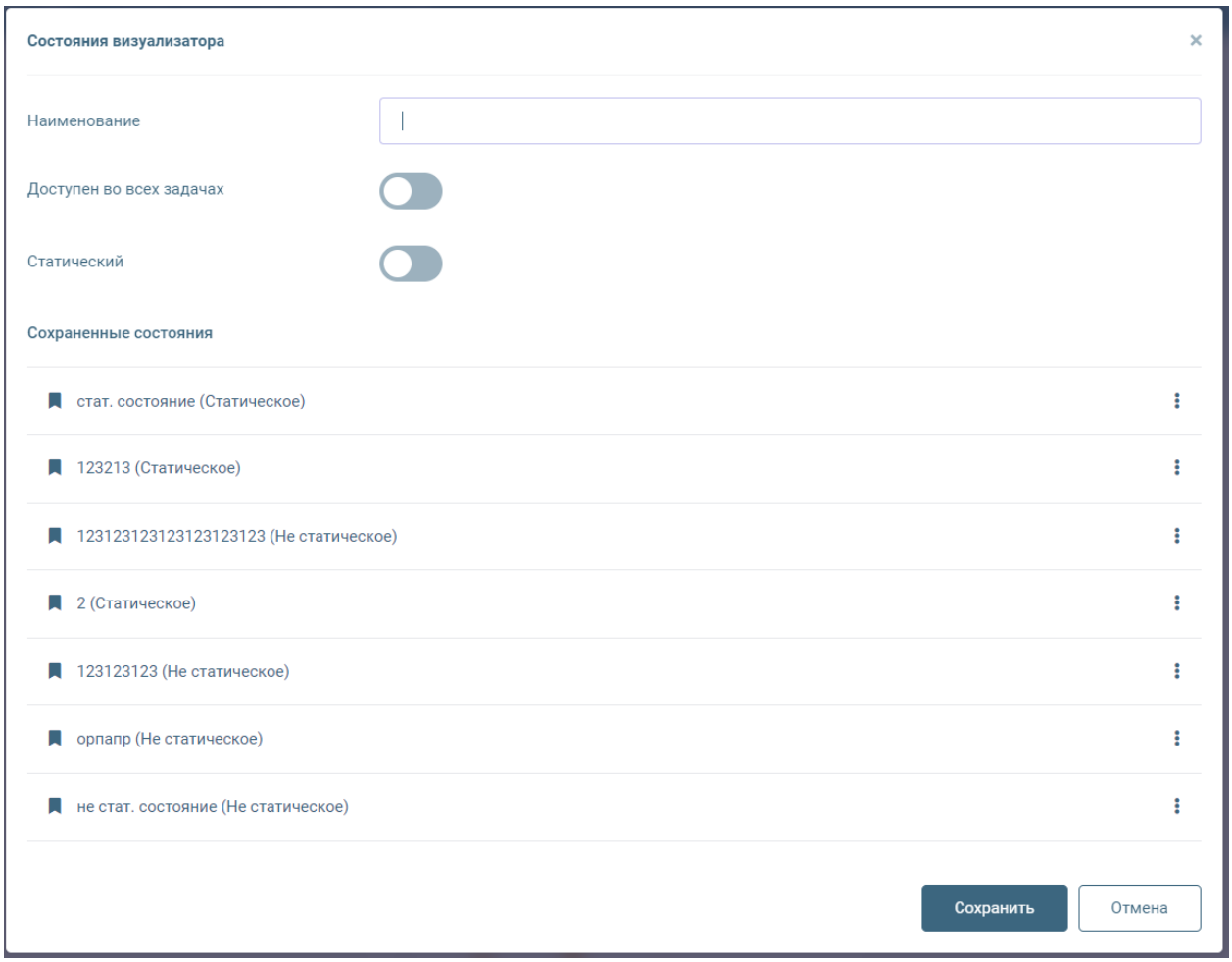


Рисунок 40 – Пример окна «Состояние визуализатора»

В окне «Состояние визуализатора» отображаются все созданные состояния, с возможностью удаления или ограничения области видимости только текущей задачей. Для добавления нового состояния необходимо указать его наименование и тип, после чего выбрать будет ли он доступен во всех задачах или только в текущей, а потом нажать на кнопку «Сохранить».

Для удаления визуализатора необходимо открыть меню по значку ••• и нажать кнопку «Удалить». Будет выведено диалоговое окно для подтверждения действия.

4.2.11.1 Типы состояний

Состояния бывают 2-х типов:

1. Статичное состояние – «снимок» визуализатора в момент создания. При изменении оригинального визуализатора такое состояние не будет изменяться.
2. Динамическое состояние – при изменении оригинального визуализатора будет изменено и это состояние.

Схема возможного представления визуализаторов и их состояний представлена на рисунке.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

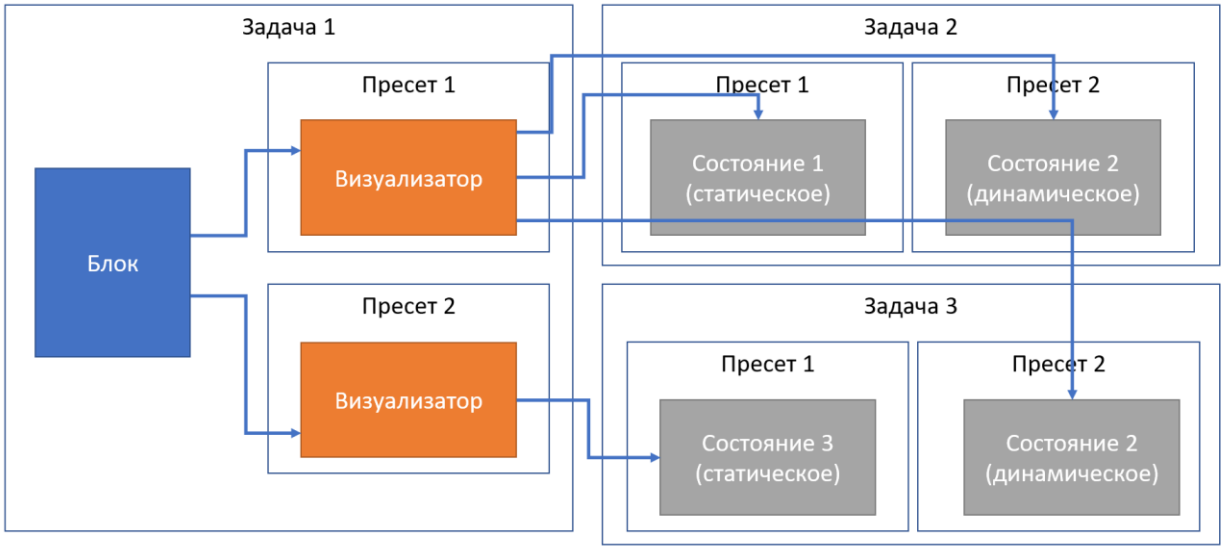


Рисунок 41 – Пример схемы возможного представления визуализаторов и их состояний

4.2.11.2 Управление стилем сохраненных визуализаторов

Для сохраненных визуализаторов доступны все настройки, применяемые для простых визуализаторов.

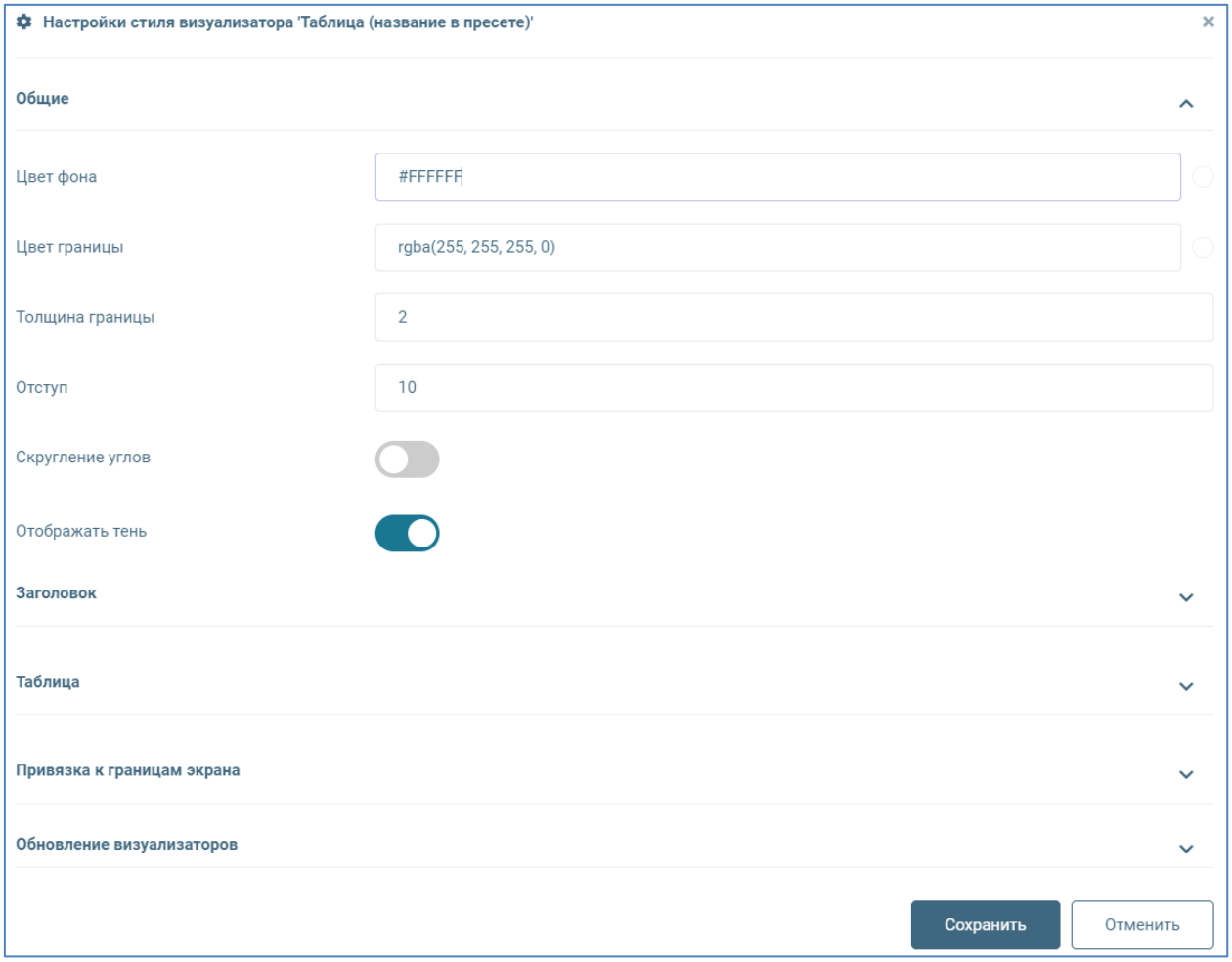


Рисунок 42 – Пример настройки стиля визуализатора

При редактировании настроек сохраненного визуализатора, все настройки сохраняются только для данного визуализатора в конкретном пресете.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

При удалении и повторном добавлении визуализатора все сохраненные настройки будут применены повторно.

4.2.11.3 Технические детали

Схема хранения состояний и их стилей приведена на рисунке.

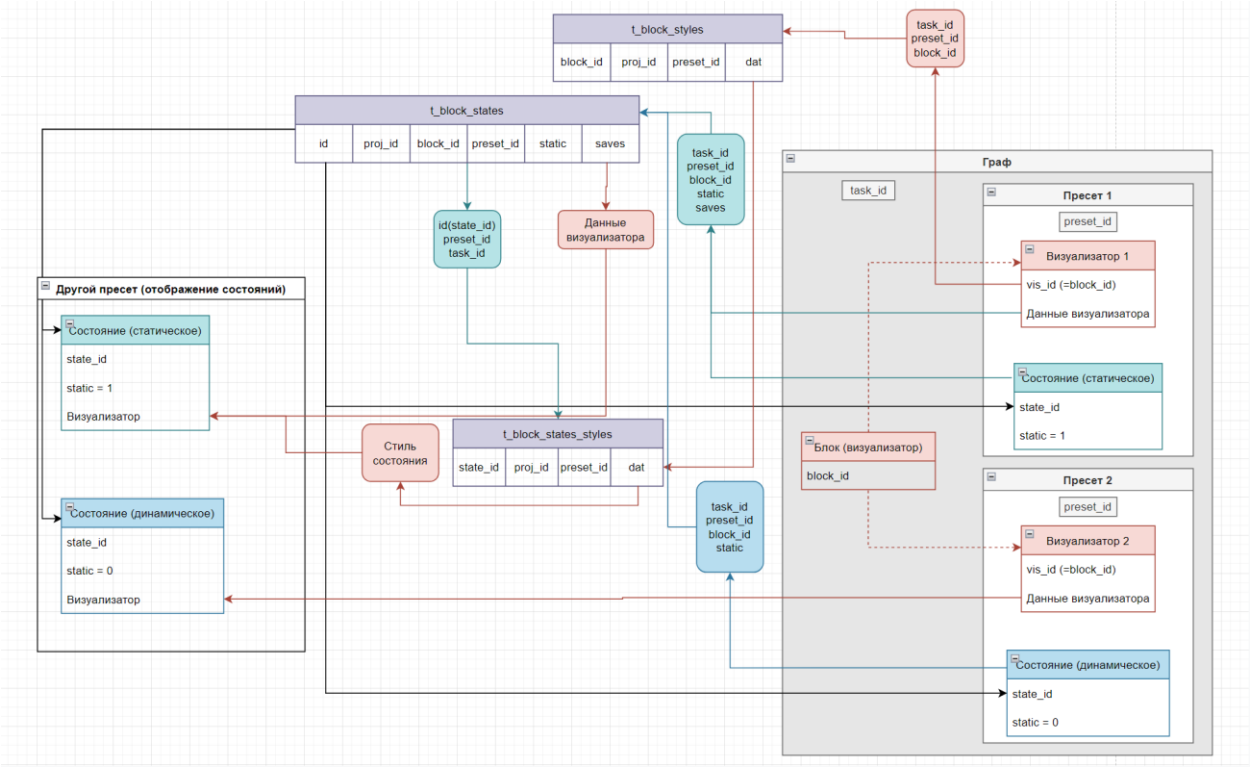


Рисунок 43 – Схема хранения состояний и их стилей

4.2.12 Экспорт визуализаторов

Приложение позволяет осуществлять экспорт любых визуализаторов в форматы PDF, PPTX, XLSX.

4.2.12.1 Процесс экспорта

1. Открыть пресет.
2. В верхней панели инструментов нажать на кнопку «Сделать снимок экрана» (или F8 на клавиатуре).
3. В панели инструментов слева открыть меню «Снимки экрана».
4. Нажать на кнопку экспорта в желаемый формат.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

4.2.12.2 Меню «Снимки экрана»

Рисунок 44 – Пример меню «Снимки экрана»

1. Список всех снимков задачи.
2. Экспорт в XLSX.
3. Экспорт в PPTX.
4. Экспорт в PDF.
5. Удалить все снимки.
6. Удалить данный снимок.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

4.3 Режимы работы

4.3.1 Режим редактирования пресета

4.3.1.1 Панель «Визуализаторы»

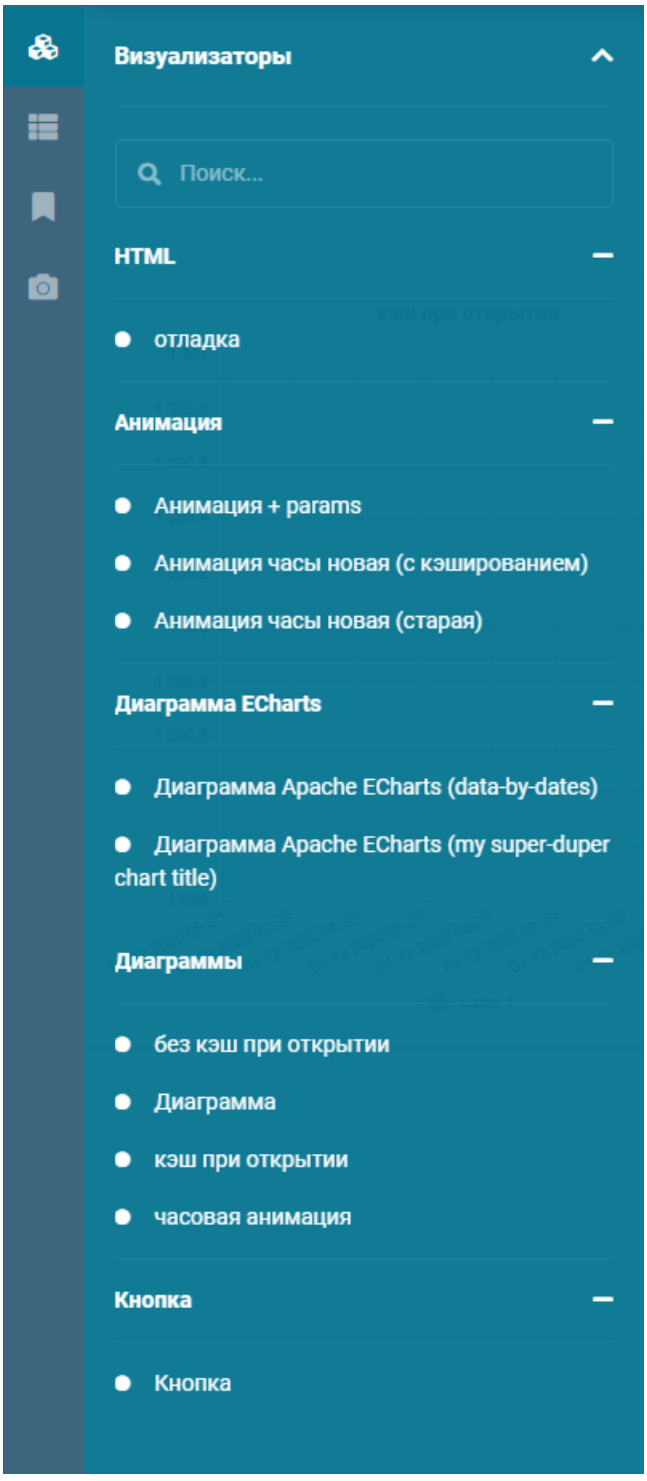


Рисунок 45 – Пример панели «Визуализаторы»

Панель содержит список визуализаторов, которые могут быть размещены на рабочей области в рамках текущего пресета.

Список визуализаторов сгруппирован по типу. Список групп отсортирован по алфавиту. Визуализаторы в рамках каждой группы также отсортированы по алфавиту.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Присутствует возможность отфильтровать список визуализаторов по их названию с помощью поля «Поиск» в верхней части панели.

4.3.2 Режим автоматического обновления визуализаторов

Режим автоматического обновления визуализаторов включается в настройках графа: Настройки задачи (см. п. 7.12) при помощи флага «Использовать механизм оповещений».

Если настройка включена:

1. Если на графе рассчитывается визуализатор, который находится не в событийной ветке, то после расчета данный визуализатор обновится на всех пресетах автоматически.
2. Если на графе рассчитывается визуализатор, который вызывает событие, или в событийной ветке на пресете отображается индикатор, что данные обновлены и пользователю нужно обновить страницу для их отображения: .

Важно! Индикатор не отображается у пользователя на странице, с которой было вызвано событие.

4.3.3 Режим embedded – встраивание без меню

При открытии пресета с параметром в строке адреса **embedded=1** пресет будет отображаться без бокового и верхнего меню. В остальном этот режим аналогичен режиму просмотра. В том числе доступно и разворачивание визуализаторов на весь экран.

4.3.4 Режим открытия приложения без системного меню

При необходимости для определенных пользователей может быть задан режим открытия приложения без отображения системного бокового и верхнего меню. Поведение приложения в этом режиме аналогично режиму встраивания, но не зависит от параметров в строке адреса. Необходимость скрытия или отображения системного меню задается при настройке ролей.

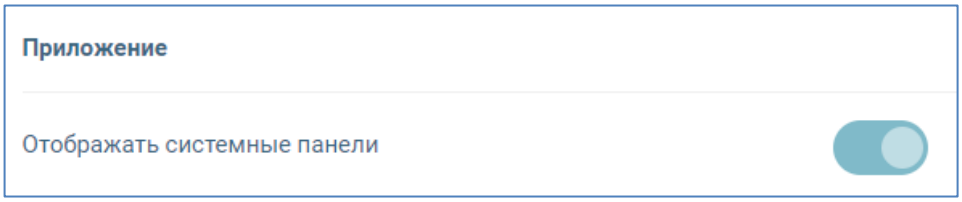


Рисунок 46 – Пример настройки скрытия или отображения системного меню

4.4 Сохранение данных из пользовательского приложения

Сохранение данных из пользовательского приложения (writeback) – механизм, позволяющий при изменении данных в приложении (к примеру, в блоке «Таблица» или «Диаграмма»), изменить данные и в таблицах, которые являются источниками данных. При этом, данные в таблицах будут изменены напрямую в БД, с использованием той учетной записи, с помощью которой они были получены в блоках извлечения данных.

Для того чтобы сохранение было произведено, необходимо обеспечить в данных наличие определенных атрибутов. Блоки типа «SQL – запрос для рядов» (см. п. 6.2.10) и «SQL – запрос-справочник» (см. п. 6.2.11) позволяют извлекать из источника данные без указания привязок и структуры, но такие данные не будут поддерживаться механизмом сохранения изменений.

Для этого необходимо:

- Для блоков «SQL – запрос для рядов» и «SQL – запрос-справочник»:

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

- Прописать атрибуты справочника и наименование таблицы, в которую будут сохраняться данные.

  Важно! Необходимо указать все первичные ключи, по которым происходит выборка данных из таблицы.

- Для блоков, имеющих мастер настройки справочника (см. п. 7.8.7) или мастер настройки JSON справочника (см. п. 7.8.1):
  - Указать все атрибуты справочника и наименование таблицы (если оно доступно).

4.4.1 Редактирование данных

Для редактирования данных в приложении, необходимо щелкнуть правой кнопкой мыши по названию визуализатора и выбрать пункт «Редактировать данные».

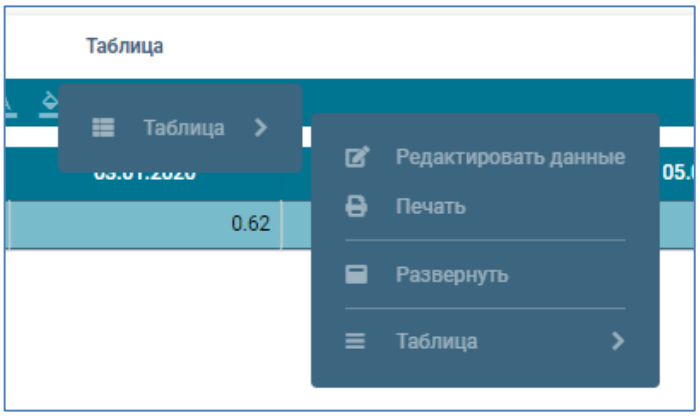


Рисунок 47 – Пример пункта «Редактировать данные»

В случае блока «Таблица» редактирование так же доступно при нажатии на значок «Карандаш» в верхнем меню.

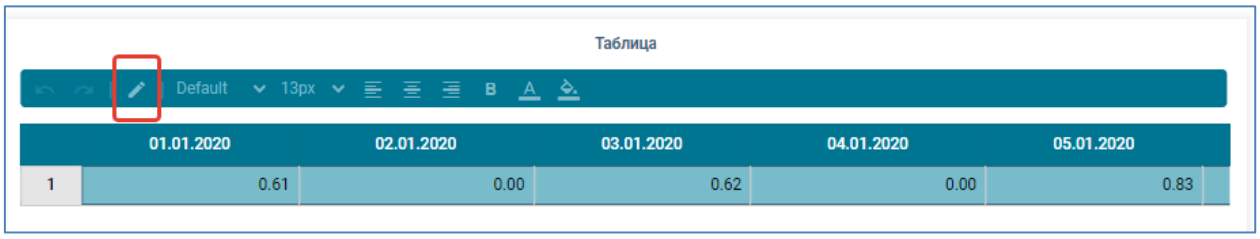


Рисунок 48 – Пример кнопки «Карандаш»

Переход в режим редактирования позволит вводить новые значения в поля таблицы, или изменять значения «рисованием» на диаграмме блока «Диаграмма».

4.4.2 Сохранение изменений

После завершения редактирования, аналогично, щелчком правой кнопки мыши по названию, открыть меню и выбрать пункт «Сохранить».

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

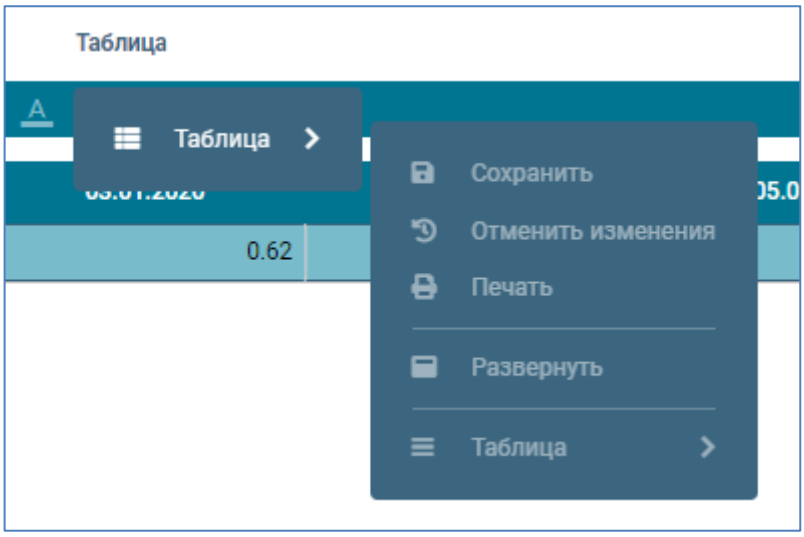


Рисунок 49 – Пример пункта «Сохранить»

В случае блока «Таблица» можно так же выбрать пункт «Сохранить» в верхнем меню.

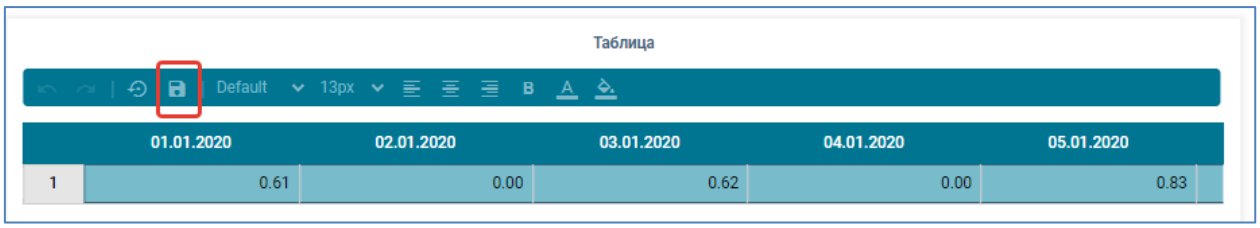


Рисунок 50 – Пример кнопки «Сохранить» в верхнем меню

После этого Системой будет сформирован SQL запрос с измененными данными на основе атрибутов и наименования таблицы (кроме JSON справочника), и данные в таблице будут обновлены.

4.4.3 Отмена изменений

Для отмены сделанных изменений в меню доступен пункт «Отменить изменения». При этом визуализатор выйдет из режима редактирования и все внесенные значений будут возвращены к своим изначальным состояниям.

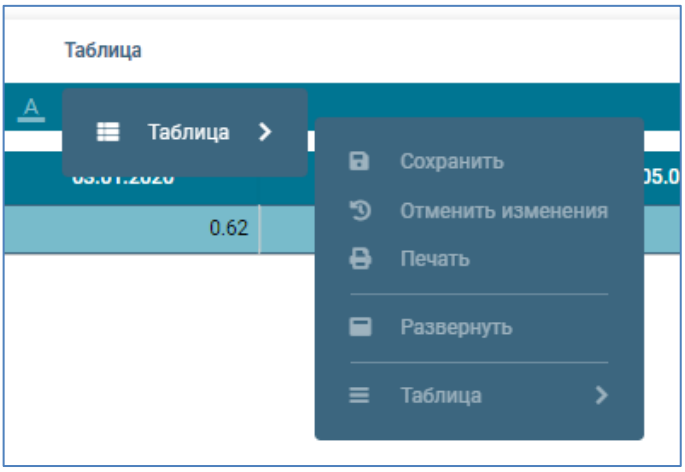


Рисунок 51 – Пример пункта «Отменить изменения»

В случае визуализатора «Таблица» на панели доступна кнопка «Отменить изменения».

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

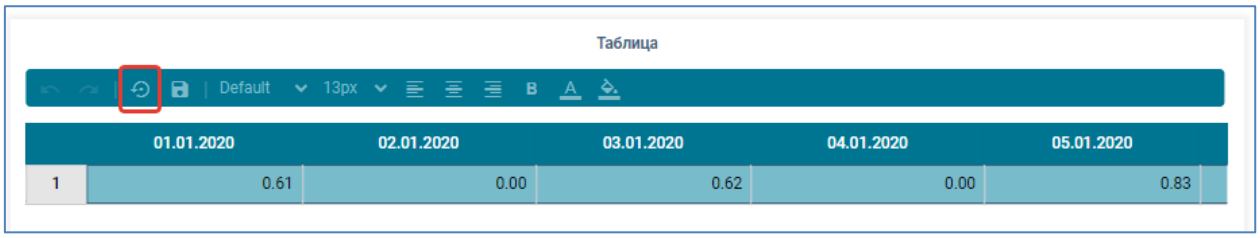


Рисунок 52 – Пример кнопки «Отменить изменения»

4.4.4 Расчет графа после сохранения

После выполнения сохранения на графе производится расчет всех блоков во всех цепочках, стоящих **после** блока, в котором получаются данные. Таким образом, после сохранения данных будут пересчитаны все блока **и события**, которые есть в графе в цепочках после блока получения данных. При этом будут пересчитаны так же и события, и все визуализаторы.

После этого произойдет обновление визуализаторов на пресете, если они попали в эти пересчитываемые цепочки блоков.

4.4.5 Сохранение данных при использовании воркеров

Если в приложение включен режим работы с воркерами (т.е. само приложение работает как координатор, а все расчеты отравляются на воркеры) – при сохранении данных приложение отправляет сохранение на воркер, где и происходит сохранение данных и перерасчет веток графа.

4.5 Блоки

4.5.1 Санитайзер HTML

Санитайзер позволяет определить потенциально вредоносный код HTML. При проверке кода используется принцип позитивной модели безопасности: «что не разрешено – то запрещено». Допустимые действия в HTML описаны ниже:

4.5.1.1 Допустимые элементы

- article, aside;
- figure;
- section;
- summary;
- Заголовки: h1, h2, h3, h4, h5, h6;
- Hgroup;
- br, div, hr, p, span, wbr;
- Строковые элементы для работы с текстом: abbr, acronym, cite, code, dfn, em, figcaption, mark, s, samp, strong, sub, sup, var;
- Блочные и строковые элементы, не влияющие на семантику документа, но определяющие стиль: b, i, pre, small, strike, tt, u;
- rp, rt, ruby.

Помимо тегов выше допустимы **списки, таблицы и изображения.**

4.5.1.2 Работа с URL

- Допустимы стандартные схемы: http, https, mailto.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

- Допустимы относительные ссылки.
- При описании ссылок атрибут rel="nofollow" не требуется.

4.5.1.3 Работа с атрибутами

Для всех элементов допустимы атрибуты:

- dir (допустимы значения, удовлетворяющие регулярному выражению `(?i)^(rtl|ltr)$`);
- id (допустимы значения, удовлетворяющие регулярному выражению `[a-zA-Z0-9\:\-_\.]+`);
- title (значения, удовлетворяющие `^[\p{L}\p{N}\s\-_',\[\]!\./\\\(\)]*$`);
- align;
- style;
- color;
- controls;
- height;
- id;
- src;
- target;
- title;
- type;
- width.

Для элемента **details** доступен атрибут **open** со значениями, удовлетворяющими регулярному выражению `(?i)^(|open)$`.

Для элемента **blockquote** доступен атрибут **cite**.

Для элемента **map** доступен атрибут **name** со значениями, удовлетворяющими регулярному выражению `^([\p{L}\p{N}_-]+)$`.

Для элемента **area**:

- Доступен атрибут **alt** со значениями, удовлетворяющими регулярному выражению `^[\p{L}\p{N}\s\-_',\[\]!\./\\\(\)]*$`;
- Доступен атрибут **coords** со значениями, удовлетворяющими регулярному выражению `^([0-9]+,)+[0-9]+$`;
- Доступен атрибут **rel**;
- Доступен атрибут **shape** со значениями, удовлетворяющими регулярному выражению `(?i)^(default|circle|rect|poly)$`.

Для элемента **img** доступен атрибут **usemap** со значениями, удовлетворяющими регулярному выражению `(?i)^#[\p{L}\p{N}_-]+$`.

Для элементов **bdi** и **bdo** доступен атрибут **dir** со значениями, удовлетворяющими регулярному выражению `(?i)^(rtl|ltr)$`.

Для элементов **del**, **ins**:

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

- Доступен атрибут **cite** со значениями, удовлетворяющими регулярному выражению `^[\p{L}\p{N}\s\-_',\[\]!\./\\\(\)]*$`;
- Доступен атрибут **datetime** со значениями, удовлетворяющими стандарту ISO8601 (`^[0-9]{4}(-[0-9]{2}(-[0-9]{2}([ T][0-9]{2}(:[0-9]{2}){1,2}(.[0-9]{1,6})` + `?Z?([\+-][0-9]{2}:[0-9]{2})?)?)?)?$`).

Для элемента **meter** доступны атрибуты **value**, **min, max**, **low**, **high, optimum** со значениями, удовлетворяющими регулярному выражению `^[-+]?[0-9]*\.?[0-9]+([eE][-+]?[0-9]+)?$`.

Для элемента **progress** доступны атрибуты **value**, **max** со значениями, удовлетворяющими регулярному выражению `^[-+]?[0-9]*\.?[0-9]+([eE][-+]?[0-9]+)?$`.

4.6 Стандартные сочетания клавиш

4.6.1.1 Описание

Стандартные сочетания клавиш («hotkey») через нажатие определённых клавиш (сочетания клавиш) позволяют упростить и ускорить работу на графе и пресете приложения, путём более быстрого вызова необходимых, часто используемых, функций.

4.6.1.2 Перечень сочетаний клавиш

Граф	
Сочетание клавиш	**Эффект**
Ctrl+Enter	Подтверждения ввода и закрытия модального окна с клавиатуры
Ctrl+H	Повторить действие (Redo)
Ctrl+S	Сохранение страницы браузера в HTML формате
Ctrl+Z	Отменить действие (Undo)
Esc	Отмена ввод и закрытие модального окна
F1	Вызов информации «О приложении»
F2	Включение/отключение режима отладки графа
F3	Запуск расчёта задачи
F4	Перейти в пользовательское приложение
F5	Перезагрузка страницы приложения
F9	Открыть окна поиска по приложени

Ctrl+Enter	Подтверждения ввода и закрытия модального окна с клавиатуры
Ctrl+S	Сохранение страницы браузера в HTML формате
Esc	Отмена ввод и закрытие модального окна
F1	Вызов информации «О приложении»
F2	Включение режима редактирования пресета
F3	Запуск расчёта задачи
F4	Перейти на граф задачи
F5	Перезагрузка страницы приложения
F7	Откртие окна настроек пресета
F8	Создание снимка экрана
F9	Открыть окна поиска по приложени

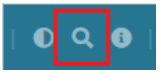
Приложение (Режим редактирования)	
Сочетание клавиш	**Эффект**
Ctrl+Enter	Подтверждения ввода и закрытия модального окна с клавиатуры
Ctrl+S	Сохранение страницы браузера в HTML формате
Esc	Отмена ввод и закрытие модального окна

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Приложение (Режим редактирования)	
Сочетание клавиш	**Эффект**
F1	Вызов информации «О приложении»
F2	Выключение режима редактирования пресета
F3	Запуск расчёта задачи
F4	Перейти на граф задачи
F5	Перезагрузка страницы приложения
F6	Сохранение пресета
F7	Откртие окна настроек пресета
F8	Создание снимка экрана
F9	Открыть окна поиска по приложени

4.7 Поиск

Функция поиска позволяет найти задачи, пресеты, блоки и визуализаторы по их наименованию, а так же нескольким другим параметрам.

Для включения поиска необходимо нажать иконку с лупой в правом верхнем углу главной страницы приложения  или нажать клавишу **F9**.

После включения откроется окно поиска.

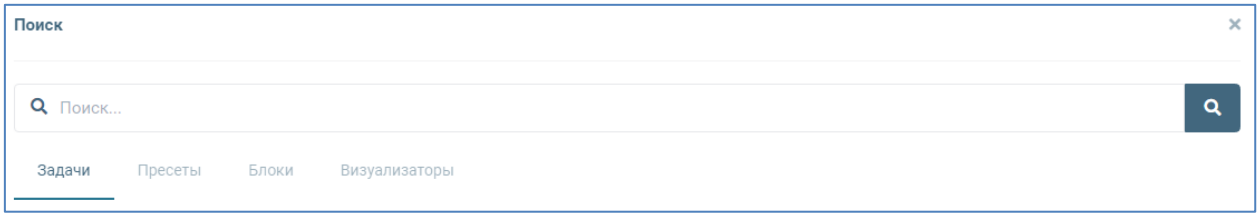


Рисунок 53 – Пример окна «Поиск»

4.7.1 Задачи

Поиск по задачам производится по наименованию задачи.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

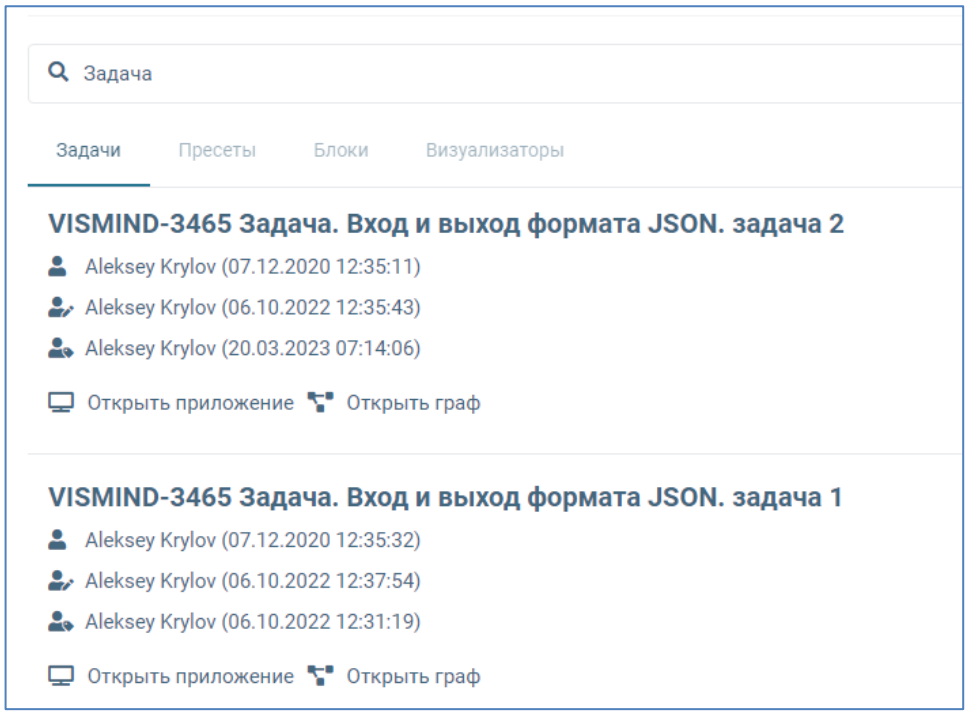


Рисунок 54 – Пример поиска по задачам

4.7.2 Пресеты

Поиск по пресетам производится по наименованию пресета.

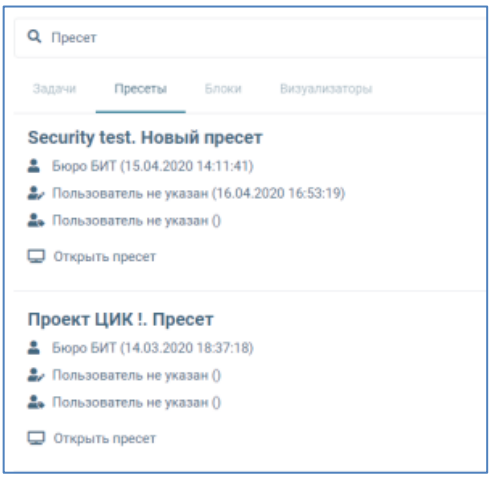


Рисунок 55 – Пример поиска по пресетам

4.7.3 Блоки

Поиск по блокам может осуществляться как по наименованию блока, так и по содержимому блока.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

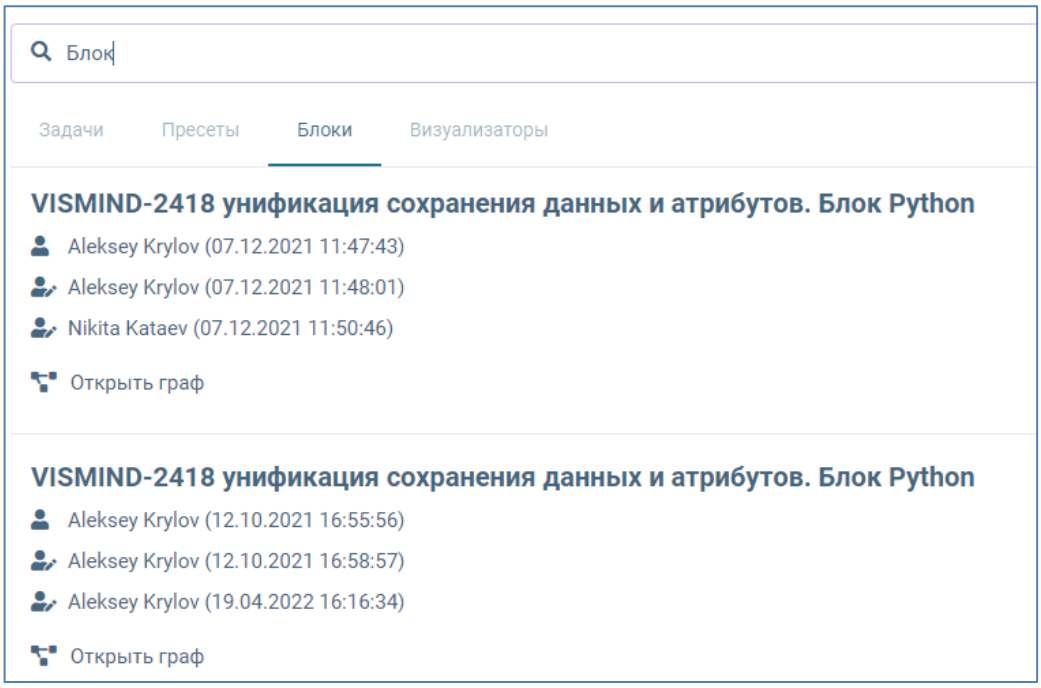


Рисунок 56 – Пример поиска по блокам

Во втором случае, поисковый запрос включает в себя поисковые конструкции, разделенные точкой с запятой:

<Параметр1>: значение1; <Параметр2>: значение2;...

Для поиска все параметры считаются объединенными конструкцией «И».

Параметры делятся на две категории: общие параметры блоков и специфические параметры каждого блока.

4.7.4 Визуализаторы

Поиск по визуализаторам производится по наименованию визуализатора.

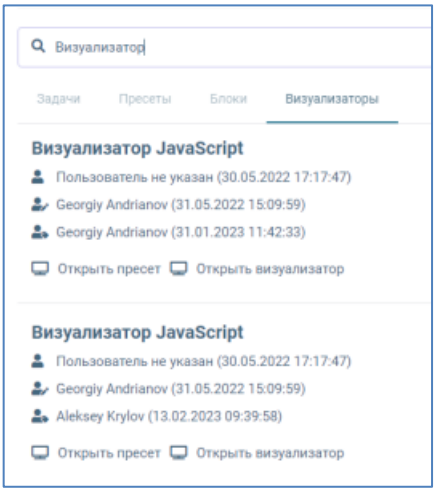


Рисунок 57 – Пример поиска по визуализаторам

4.7.4.1 Общие параметры блоков

- Наименование блока – наименование блока. Аналогично простому поиску по блокам.
- Тип блока – наименование типа блока. После введения параметра, будет доступен выпадающий список со всеми типами блоков.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

- Версия блока – версия блока, целое число. Обратите внимание, что для поиска требуется строгое равенство.
- Пропуск расчета блока – флаг (Истина/Ложь).

4.7.4.2 Специфические параметры блоков

Уникальные параметры, характерные для определенного блока. Без указания типа блока – будут доступны все параметры всех блоков.

В текущей реализации доступен поиск только текстовых параметров блока.

Для поиска параметров конкретных блоков необходимо указать тип блока (Рисунок 58).

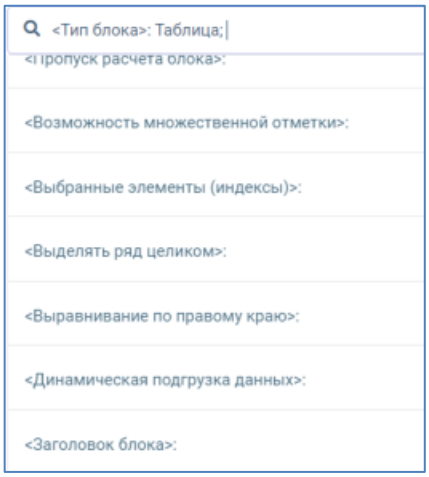


Рисунок 58 – Пример параметров конкретных блоков

В таком случае, поиск будет сделан только для конкретного типа блоков (в примере поиск всех блоков «OLAP») (Рисунок 59).

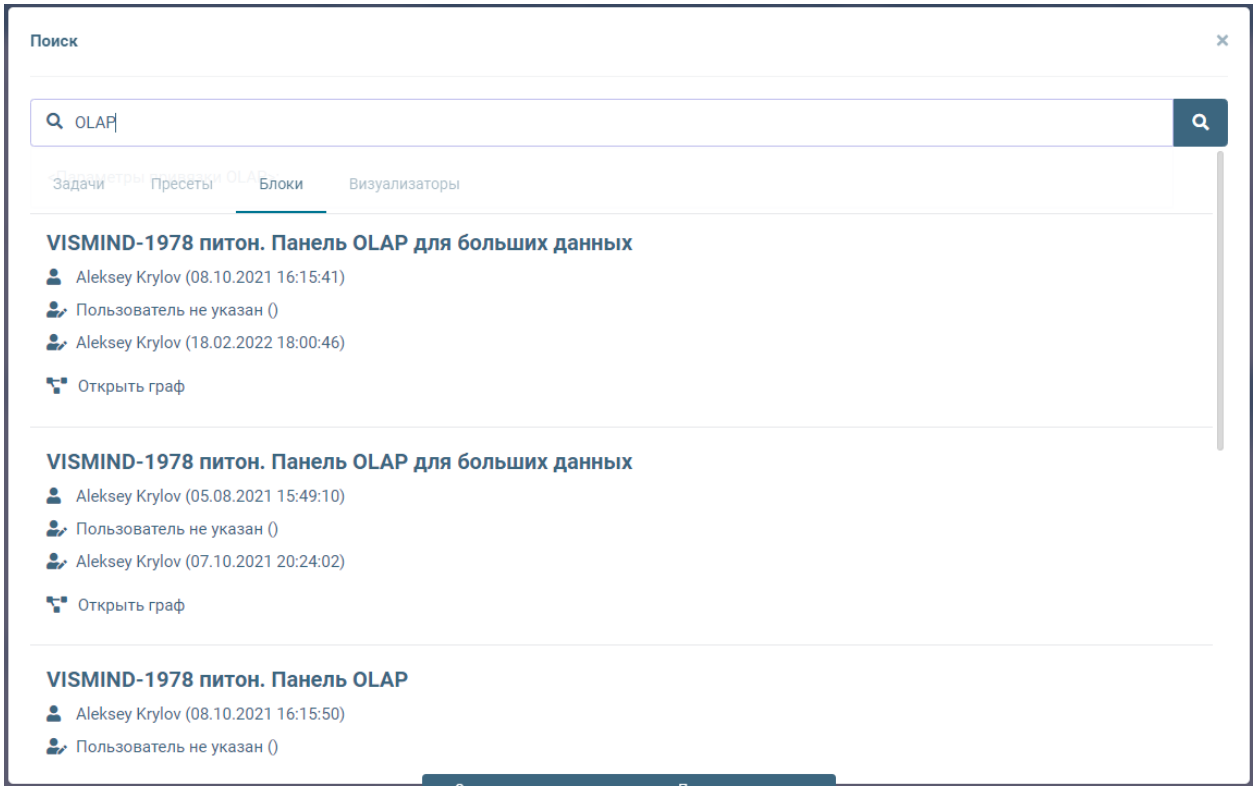


Рисунок 59 – Пример поиска для типа блоков «OLAP»

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

4.8 Экспериментальные функции

4.8.1 Включение отображения экспериментальные функции

Чтобы пользователь мог увидеть экспериментальные функции, необходимо в настройках пользователя включить флаг отображения экспериментальных функций.

Переключатель экспериментальных функций доступен только в случае, если у пользователя в настройках установлен флаг «Использовать экспериментальные функции».

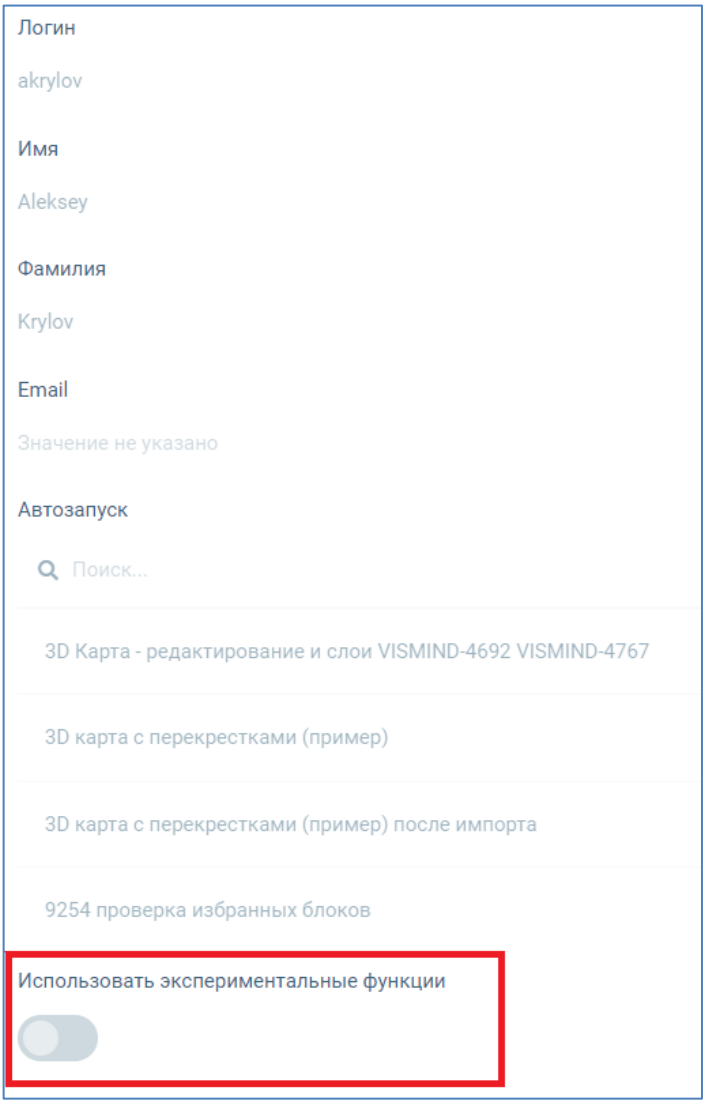


Рисунок 60 – Пример отображения флага «Использовать экспериментальные функции»

4.8.2 Индикация экспериментальных функций

Экспериментальные функции отображаются со значком .

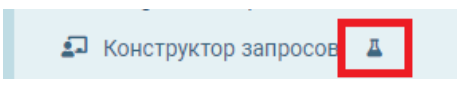


Рисунок 61 – Пример индикации экспериментальной функции

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

4.9 Особенности работы приложения с настройкой браузера «Экономия памяти»

Для корректной работы приложения настройка браузера «Экономия памяти» должна быть выключена или адрес приложения должен быть добавлен в список «Сайты, которые всегда активны».

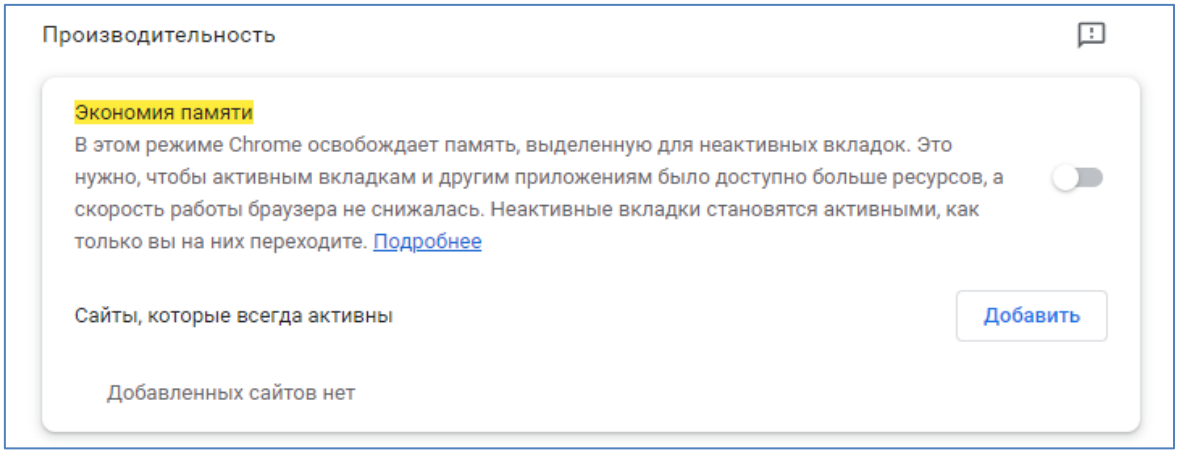


Рисунок 62 – Пример настройки браузера «Экономия памяти»

Если данная настройка включена

С включенной настройкой приложение не может сделать **logout** для неактивного пользователя. В таком случае, например, может быть заблокирован блок на графе, и другие пользователи не будут иметь возможности работать с данным блоком

4.10 Ошибки приложения

4.10.1 Ошибка со статусом HTTP 403

Код ответа на статус ошибки HTTP 403 **Forbidden** указывает, что сервер понял запрос, но отказывается его авторизовать.

В случае с приложением, данная ошибка может появиться при выполнении операции, к которой у пользователя нет доступа (к примеру, попытка зайти в задачу).

4.10.2 Ошибка со статусом HTTP 404

Код ответа на ошибку HTTP 404 **Not Found** указывает, что сервер не может найти запрошенный ресурс.

В случае с приложением, данная ошибка может появиться при попытке зайти в несуществующую задачу.

4.10.3 Ошибка со статусом HTTP 500

Код ответа сервера 500 **Internal Server Error** указывает на то, что сервер столкнулся с неожиданной ошибкой, которая помешала ему выполнить запрос.

4.10.4 Ошибка со статусом HTTP 503

HTTP 503 **Service Unavailable** код состояния сервера, указывающий на то, что сервер не готов обработать данный запрос.

Часто причиной этого оказывается закрытие сервера для технических работ или его перегрузка.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

4.10.5 Разрыв соединения

В случае если соединение с сервером потеряно или его невозможно установить, пользователь получит уведомление о переподключении.

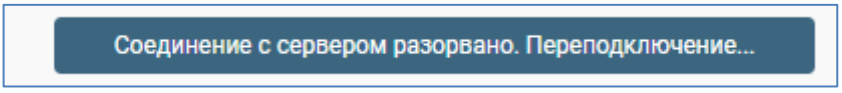


Рисунок 63 – Пример сообщения о разрыве соединения

Попытка повторного подключения к серверу происходит раз в несколько секунд.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

5 Навигатор задач

5.1 Задачи

5.1.1 3D-Карты

3D-Карты представляют собой структуры, аналогичные задачам, НСИ и источникам данных, используемые для более гибкой работы с блоком «3D-карта» (см. п. 6.1.3.1).

Структура аналогична задачам (поиск, наличие тегов), однако в этих структурах не создаются пресеты по умолчанию.

Иерархия папок сквозная, т.е. одни и те же папки применимы как для хранения задач, так и для хранения 3D-Карты.

5.1.2 Доступ

Доступ к структуре 3D-Карты осуществляется через выбор в левом верхнем углу главной страницы приложения.

Рисунок 64 – Пример выбора раздела меню «3D-Карты»

5.1.3 Граф

Граф полностью аналогичен графу задачи.

5.1.4 Импорт/экспорт задачи

Приложение поддерживает перенос задач между различными окружениями платформы.

5.1.4.1 Экспорт задачи

Для экспорта задачи необходимо:

1. Перейти в раздел приложения «Список задач». Выбрать экспортируемую задачу из общего списка, нажав на неё.
2. Нажать кнопку «Экспорт» (пиктограмма) в меню сверху, либо зажать комбинацию клавиш **Ctrl+Alt+E**.

Результатом успешного экспорта является файл с расширением json, загруженный на устройство пользователя.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

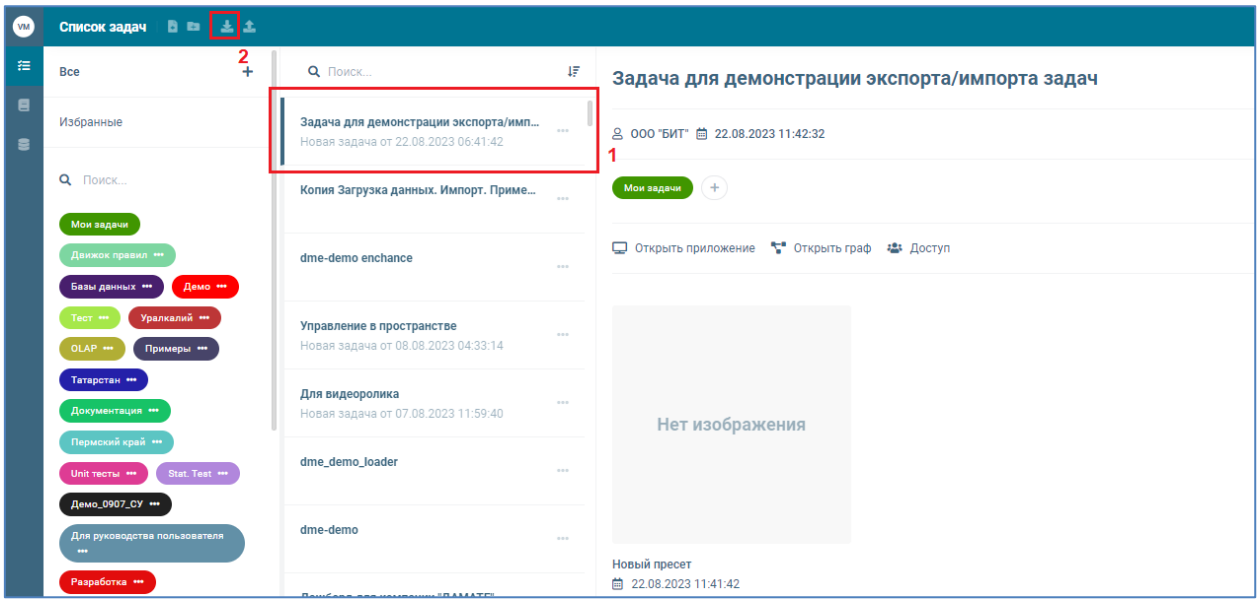


Рисунок 65 – Пример меню приложения (кнопка «Экспорт»)

5.1.4.2 Формат файла экспорта

Файл имеет расширение «.json». Имя файла определяется именем экспортируемой задачи.

5.1.4.3 Импорт задачи

Для импорта задачи необходимо:

1. Перейти в раздел приложения «Список задач».
2. В верхнем меню приложения нажать кнопку «Импорт» (пиктограмма), либо зажать комбинацию клавиш **Ctrl+Alt+I**.

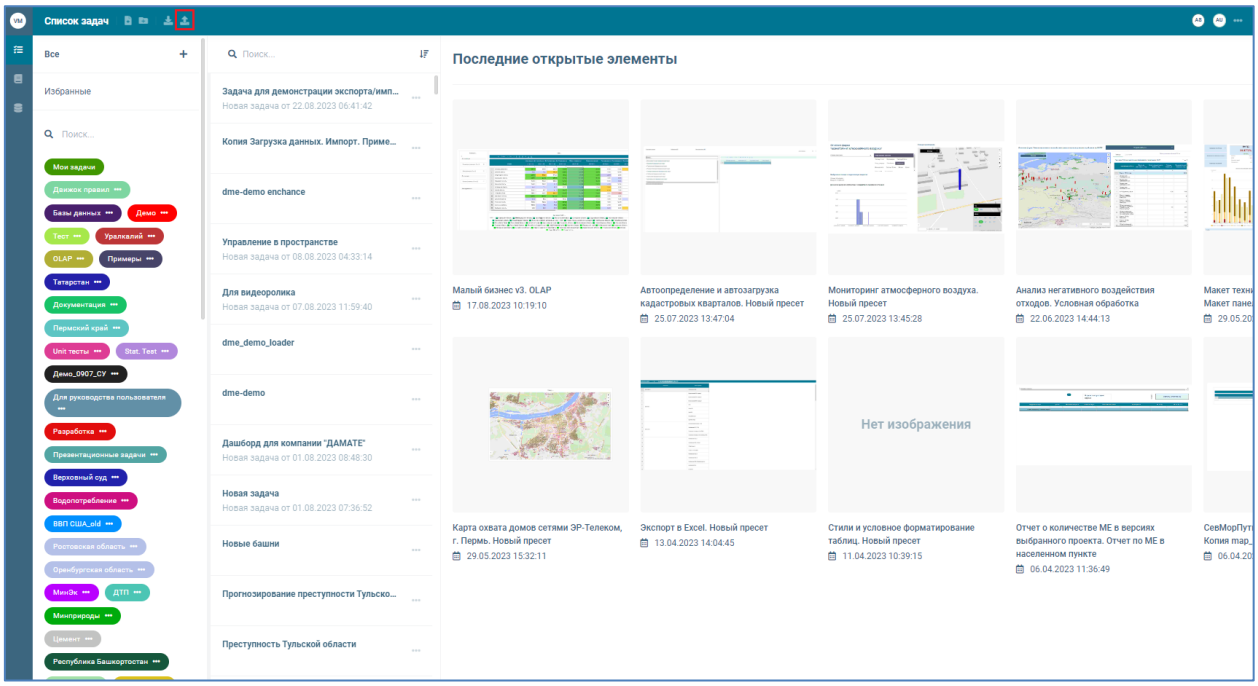


Рисунок 66 – Пример меню приложения (кнопка «Импорт»)

3. В открывшемся диспетчере файлов задать путь до загружаемого файла экспорта.
4. После выбора файла экспорта откроется системное окно «Загрузка файла». В нём отражается ход импорта задачи.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

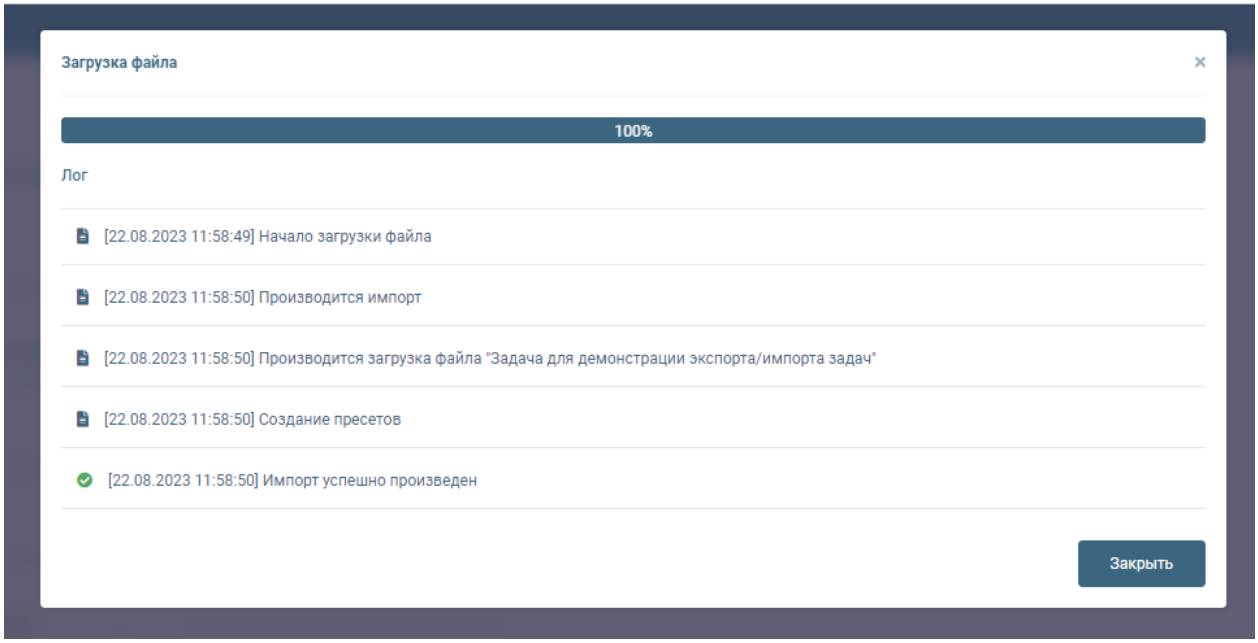


Рисунок 67 – Пример окна «Загрузка файла»

Результатом успешного импорта задачи станет новая задача, идентичная экспортируемой задаче

5.1.4.4 Примечания

Экспорт соединений с БД

При экспорте задачи не сохраняются данные соединений с БД. Ограничение введено в целях защиты подключений к БД от несанкционированного доступа.

Обратная совместимость между версиями платформы

Механизм импорта/экспорта не имеет обратной совместимости при импорте задачи в окружение с более старой версией платформы. Как правило, при переносе задач не возникает конфликтов импорта, но отсутствие обратной совместимости не гарантирует корректного импорта между разными версиями платформы.

Версионность блоков

При переносе задачи необходимо учитывать различия не только версий платформы, но и версии экспортируемых блоков. Если в целевом окружении отсутствует поддержка требуемых версий блоков, при расчёте графа возникнут ошибки.

5.1.5 НСИ и Источник данных

НСИ и Источники данных представляют собой структуры, аналогичные задачам, используемые для моделирования и расчета объектов (НСИ) или рядов данных (Источник данных), с последующим использованием этих структур в других задачах.

Структура НСИ и Источников аналогична задачам (поиск, наличие тегов), однако в этих структурах не создаются пресеты по умолчанию.

Иерархия папок сквозная, т.е. одни и те же папки применимы как для хранения задач, так и для хранения НСИ/Источников данных.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

5.1.5.1 Доступ

Доступ к структурам осуществляется через выбор нужной структуры в левом верхнем углу главной страницы приложения.

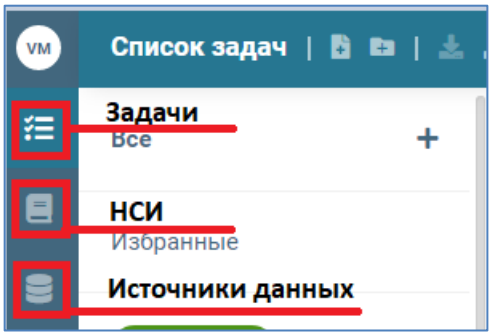


Рисунок 68 – Пример выбора структуры

5.1.5.2 Граф

Граф НСИ/Источников данных аналогичен графу задачи, за исключением наличия блока выхода на графе.

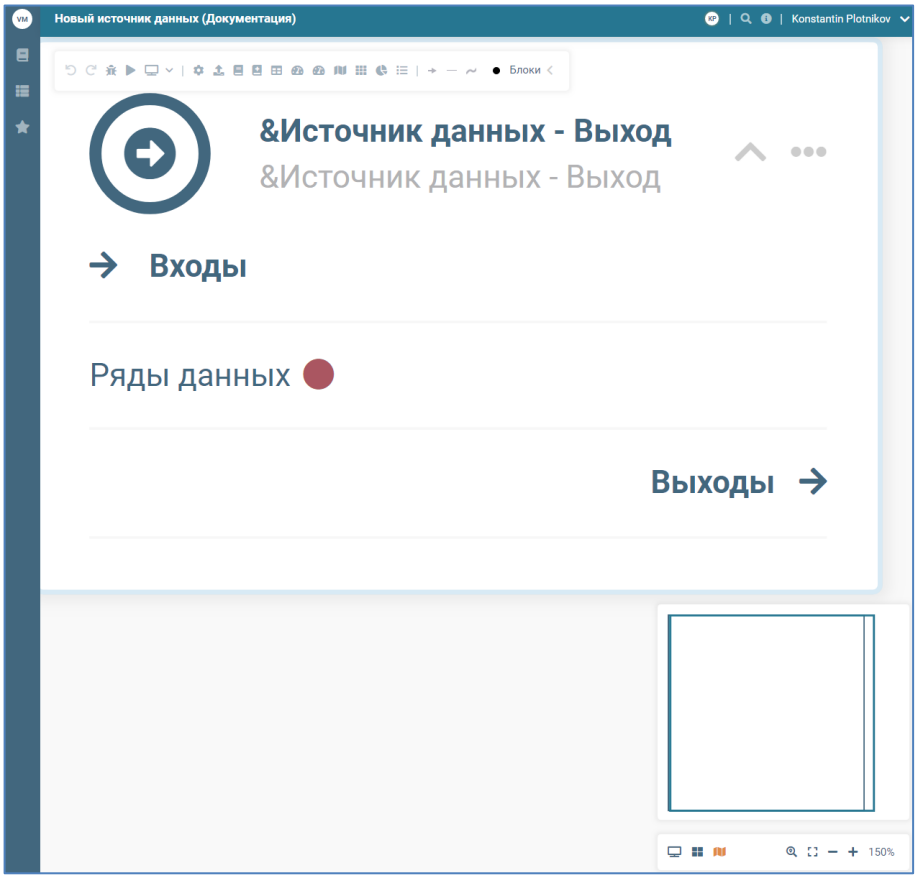


Рисунок 69 – Пример блока выхода на графе

Только что созданный Источник данных (Рисунок 69) содержит на графе блок выхода.

Для НСИ это будет блок «&НСИ – Выход», а для источника данных «&Источник данных – Выход» соответственно.

Эти блоки невозможно удалить, так же нельзя создать дополнительный выход.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Любые данные, переданные на вход этих блоков, будут переданы на выход блока «&НСИ» (если работа велась в структуре НСИ), или же на выход блока «&Источник данных» (если структура – Источник данных), при выборе нужной структуре в данных блоках.

5.1.5.3 Параметризация

Для осуществления параметризации структур применяются блоки-входы.

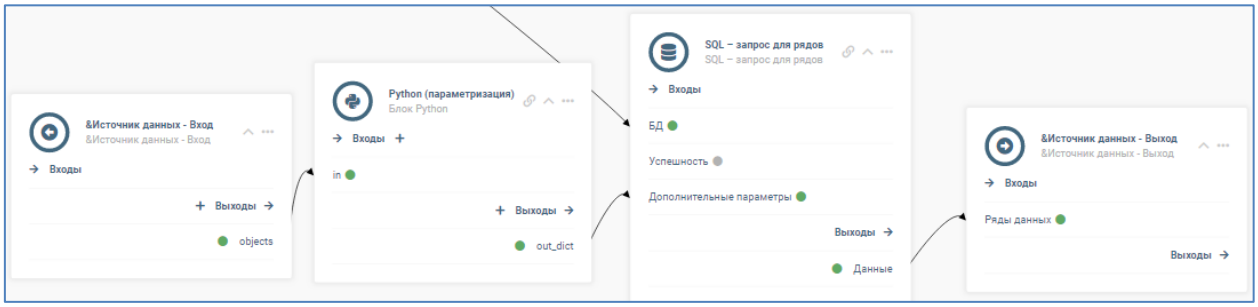


Рисунок 70 – Пример цепочки блоков с параметризацией для источника данных

По умолчанию входа нет на графе, его можно добавить из библиотеки (**Управление порядком расчета -> &НСИ – Вход/&Источник данных – Вход**).

Допустимо наличие только одного входа для структуры.

По умолчанию у входов отсутствуют выходы. Для создания выхода необходимо нажать на знак + в разделе выходов блока. Подробное создание выходов приведено в п. 6.7.1 Блок «Входные данные».

Выход блока «Вход» связан со входом блока «&НСИ»/«&Источник данных» (соответственно для каждой структуры), при выборе нужной структуре в блоках.

При изменении выходов блока «Вход», будут автоматически изменены входы блоков «&НСИ»/«&Источник данных», у которых была выбрана необходимая структура.

5.1.6 Управление доступом к задачам

В приложении поддерживается возможность настройки ограничения доступа к определённым задачам и пресетам. Механизм работает по принципу избирательного управления доступом (англ. DAC – Discretionary access control).

5.1.6.1 Ограничение доступа к задаче

Выдача доступа

При ограничении общего доступа к задаче доступ назначается на определённые пресеты, относящиеся к задаче. При этом выдача доступа к пресету подразумевает доступ и к графу соответствующей задачи.

По умолчанию все задачи находятся в общем доступе. Для ограничения доступа к задаче необходимо:

1. Найти необходимую задачу в общем списке задач и нажать на неё.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

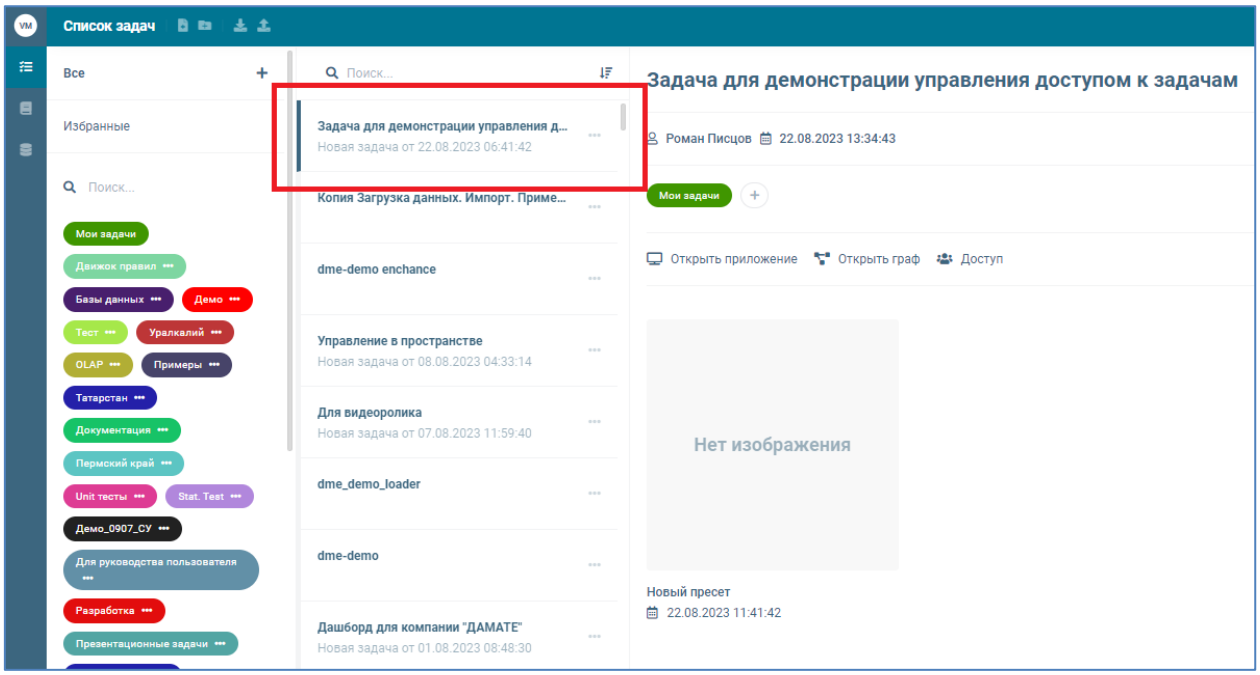


Рисунок 71 – Пример выбора задачи

2. На странице задачи нажать кнопку «Доступ».

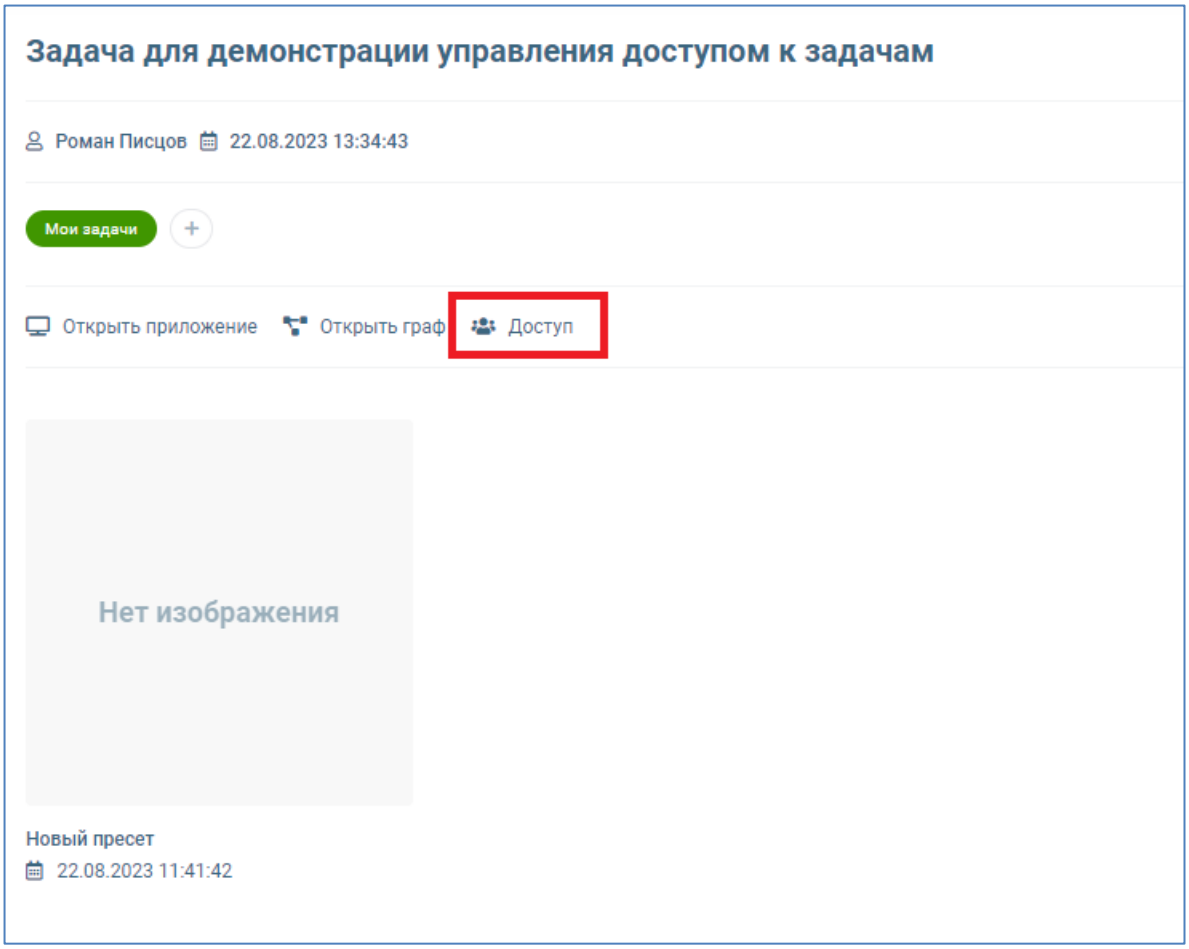


Рисунок 72 – Пример кнопки «Доступ»

3. В появившемся окне отключить настройку «Общий доступ».

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Рисунок 73 – Пример настройки «Доступ»

4. После отключения настройки задача не будет доступна пользователям, не имеющих полномочий администратора.
5. Для выдачи доступа необходимо нажать кнопку «Выдать доступ».

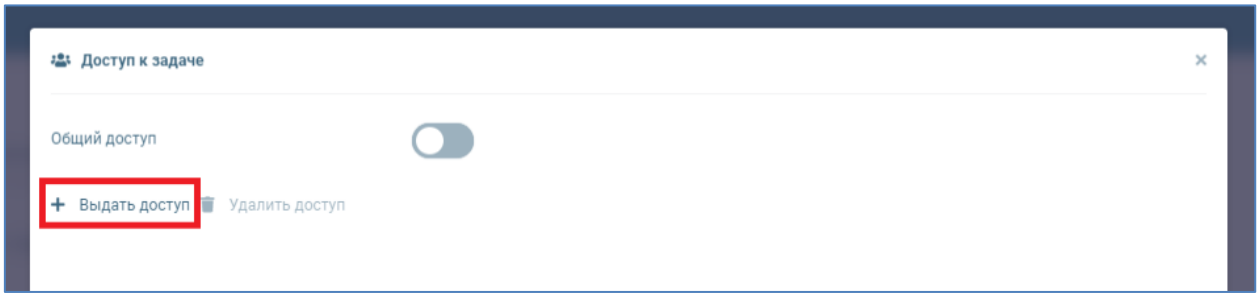


Рисунок 74 – Пример кнопки «Выдать доступ»

6. В появившемся меню откроется список всех групп в приложении а также пользователей, входящих в эти группы. Чтобы увидеть полный список пользователей группы, необходимо развернуть его нажатием на кнопку + рядом с именем групп.

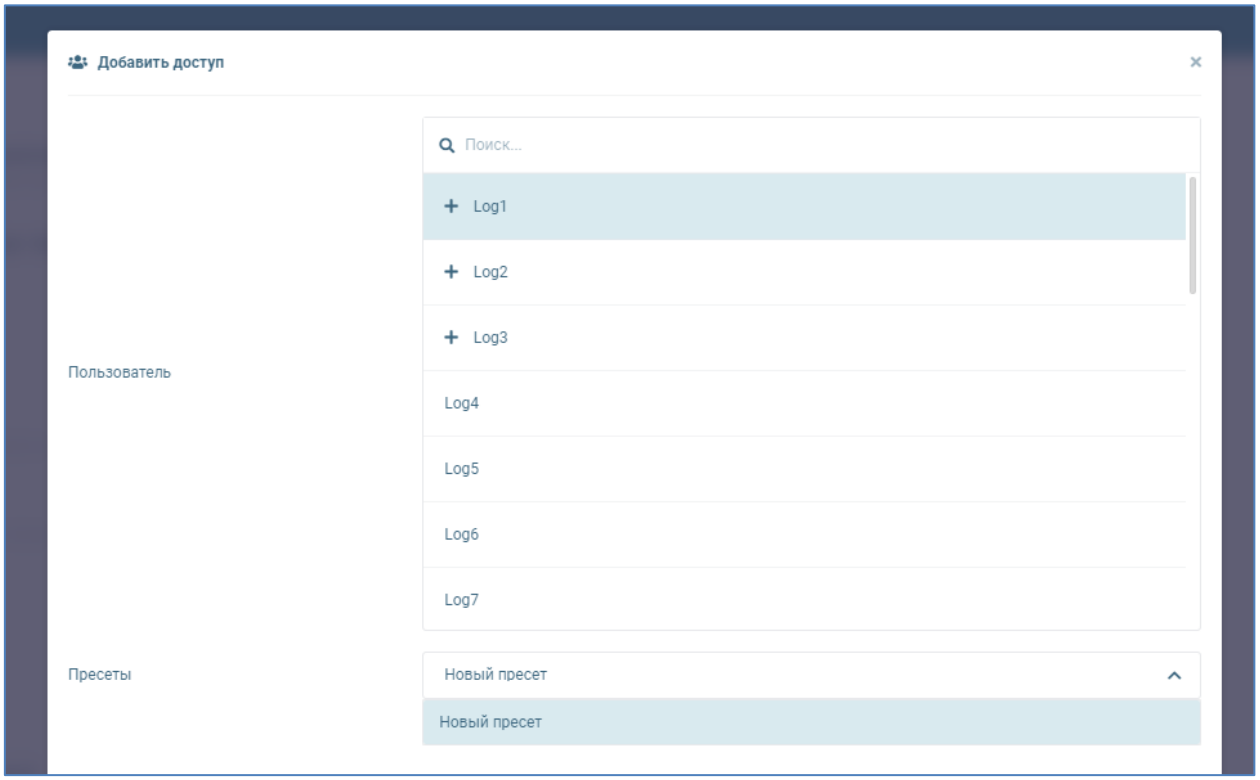


Рисунок 75 – Пример списка всех групп

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

На изображении ниже представлен развёрнутый список пользователей в группе «Log1».

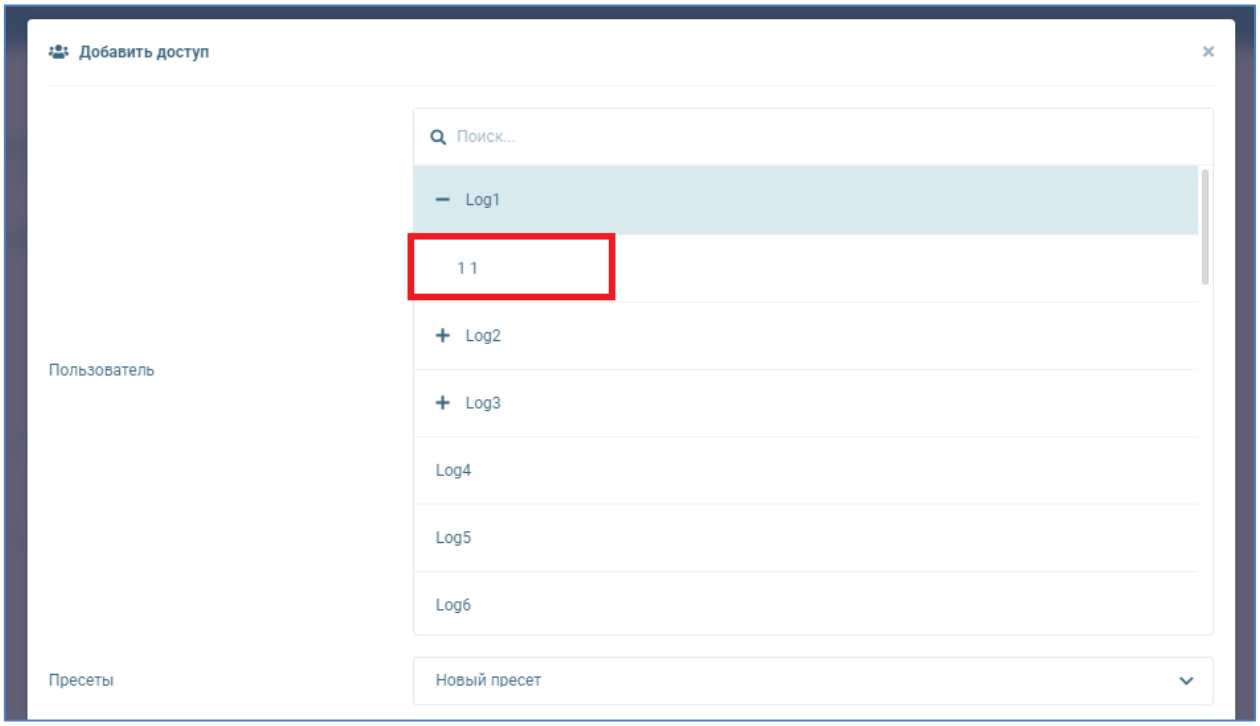


Рисунок 76 – Пример пользователей в группе «Log1»

7. Чтобы выдать доступ к группе или пользователю, требуется нажать на имя необходимой группы или пользователя, а также выбрать пресет, доступ к которому выдаётся.
8. Нажать кнопку «Сохранить».

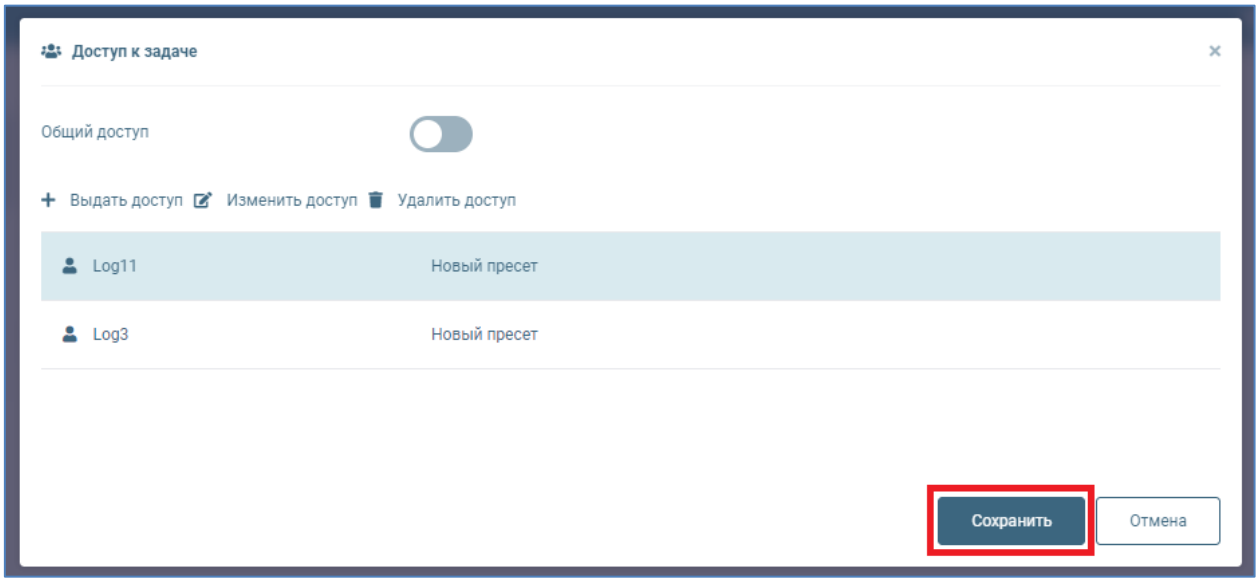


Рисунок 77 – Кнопка «Сохранить»

Результатом выдачи доступа к пресету будет добавленная запись в окне «Доступ к задаче» с указанием группы/пользователям, получившего доступ, и имени пресета.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

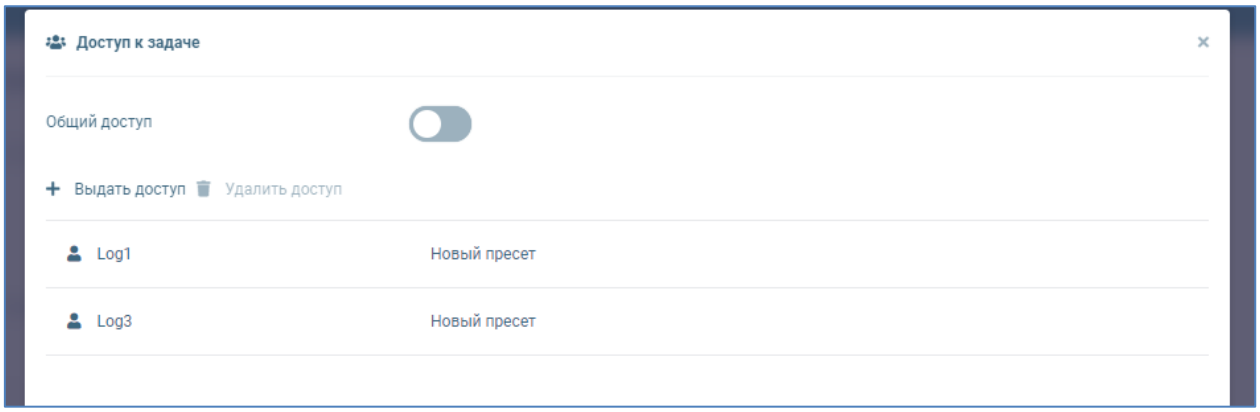


Рисунок 78 – Пример окна «Доступ к задаче»

5.1.6.2 Удаление выданного доступа к задаче

Чтобы отозвать ранее выданный доступ группы/пользователя к пресету, в окне «Доступ к задаче» нужно выбрать соответствующую строку и нажать кнопку «Удалить доступ».

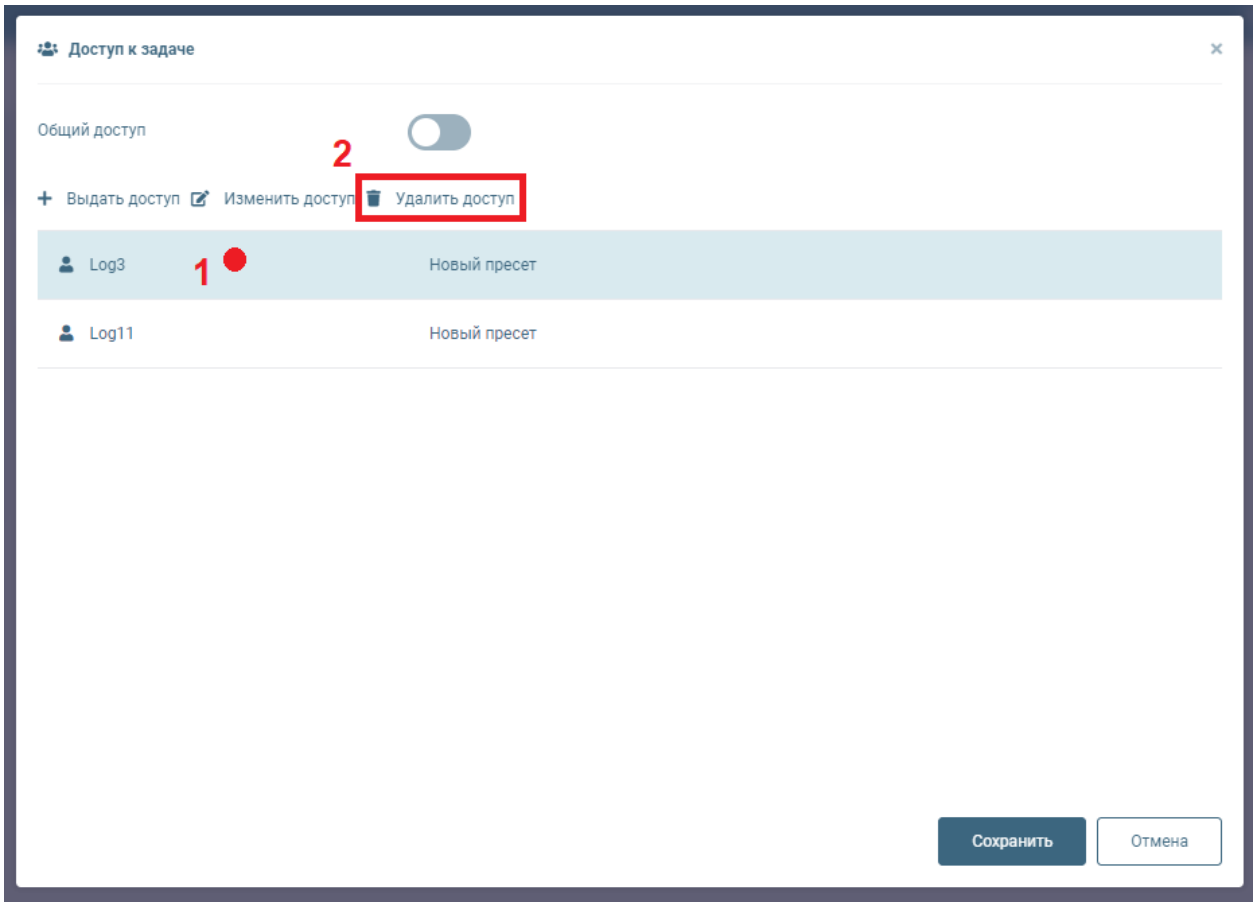


Рисунок 79 – Пример удаления доступа

5.2 Папки

Данный элемент служит для упорядочивания списка задач, созданных в приложении.

Дополнительно структура папок позволяет отделить прикладные задачи друг от друга и, внутри самой прикладной задачи, отделить необходимые элементы друг от друга.

После авторизации, если у пользователя не установлена задача для автостарта, он попадает на страницу «**Список задач».**

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Слева, выше раздела с тегами, на этой странице отображаются доступные пользователю папки с задачами.

По умолчанию перечень папок свёрнут.

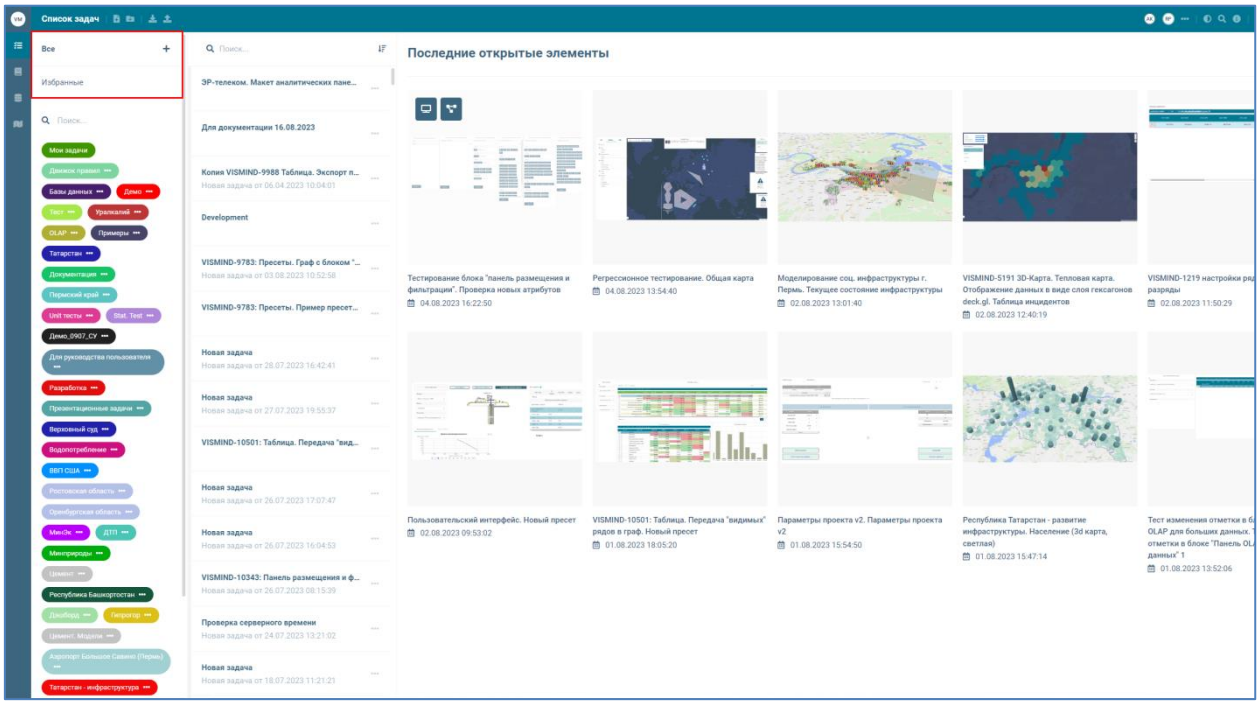


Рисунок 80 – Пример на страницы «**Список задач»**

Для папок допустима вложенность.

Папки со вложенными папками отображаются в списке значком + в свёрнутом состоянии и значком − в развёрнутом состоянии.

Для сворачивания и разворачивания списка вложенных папок необходимо нажать ЛКМ по вышеуказанным значкам.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

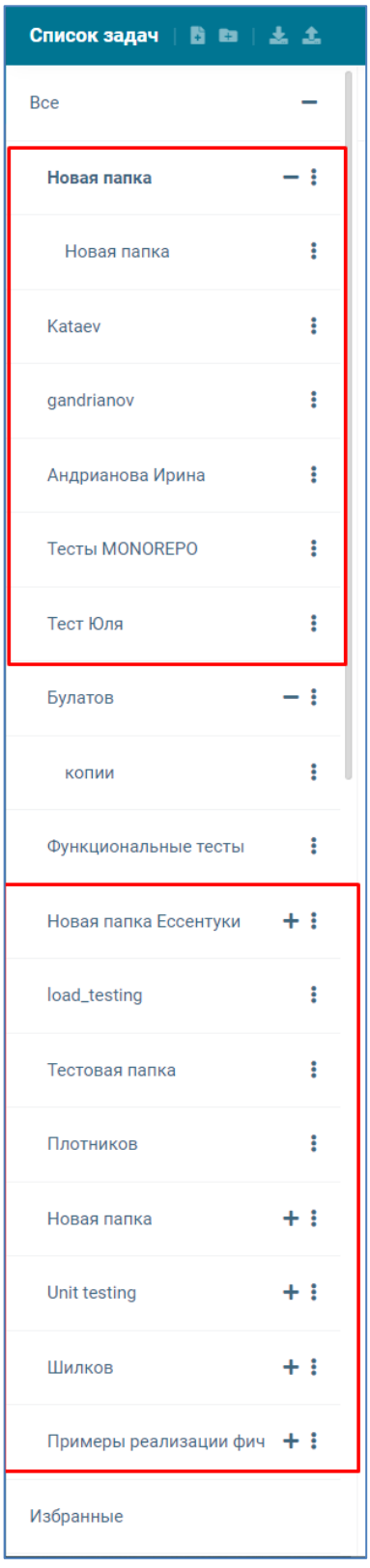


Рисунок 81 – Пример списка вложенных папок

Для создания папки необходимо выбрать куда будет создана папка, кликнув по ней ЛКМ.

Это может быть как ранее созданная папка, так и папка «**Все».**

После необходимо нажать ЛКМ по кнопке «**Создать папку».**

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

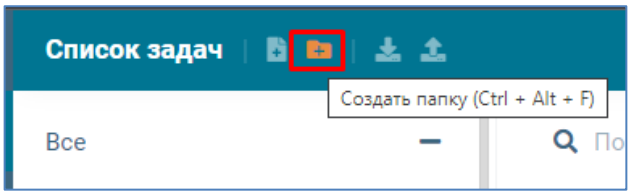


Рисунок 82 – Пример кнопки «**Создать папку**»

Для удалению папки необходимо нажать по кнопке ⋮ у паки подлежащей удалению. В появившемся окне выбрать «**Удалить».**

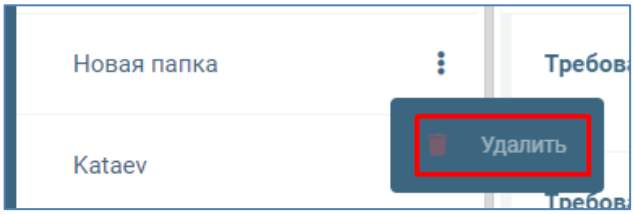


Рисунок 83 – Пример кнопки «**Удалить**»

Важно!

- Для удаления папки со вложенностями, в первую очередь необходимо удалить вложенные папки, затем удалить основную папку. В противном случае, будет выведена ошибка и папка не будет удалена.

Для переименования папки необходимо дважды нажать ЛКМ по названию папки, выделить старое название и ввести новое.

После необходимо нажать ЛКМ по значку ✓ для сохранения изменений.

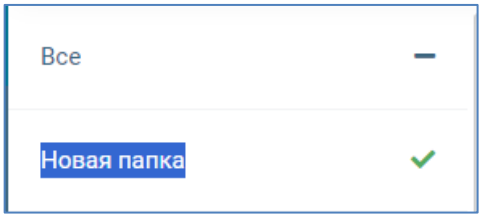


Рисунок 84 – Пример отображения папки

5.3 Теги

В приложении доступно добавление тегов для задач. Теги позволяют группировать задачи по смыслу.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

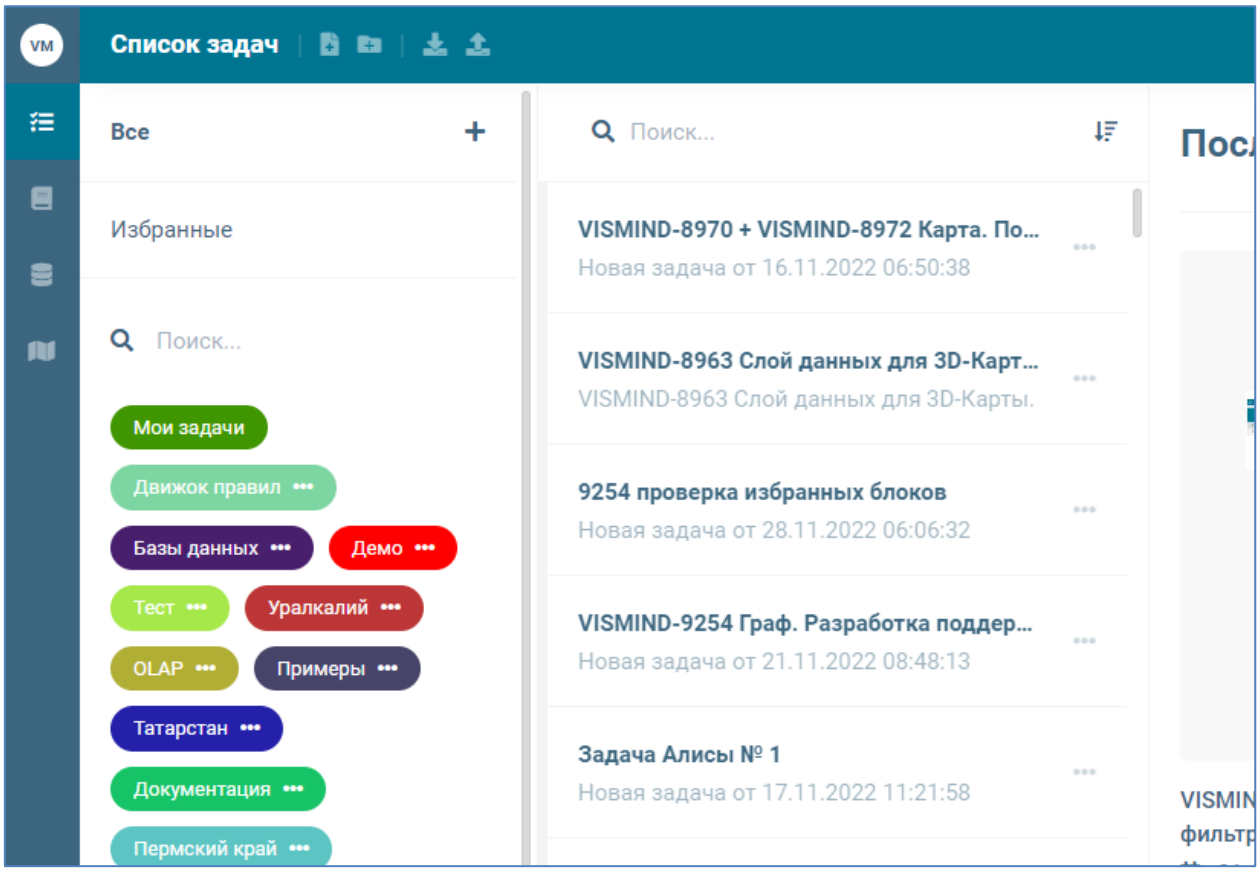


Рисунок 85 – Пример тегов для задач

5.3.1 Добавление тега

Для добавления тега необходимо нажать ЛКМ по кнопке .

Кнопка отображается последней в списке тегов, смещается при создании нового тега.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

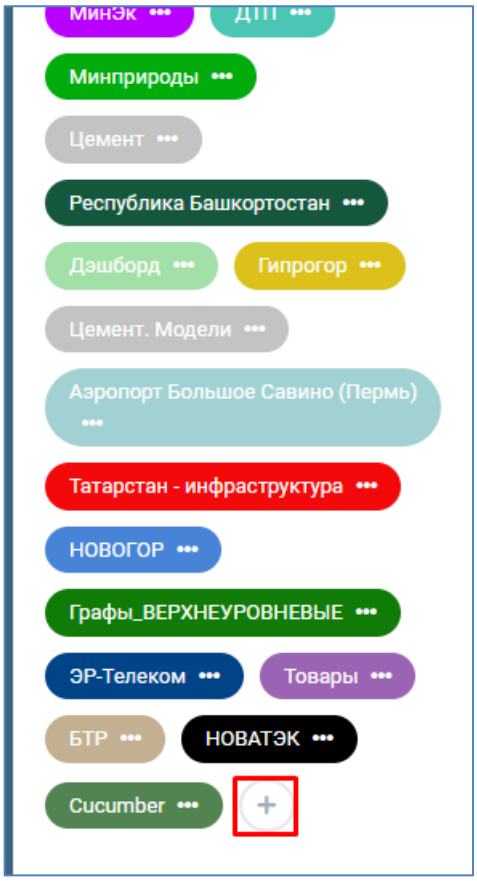


Рисунок 86 – Пример кнопки для добавления тега

В появившемся окне необходимо ввести наименование, выбрать цвет тега, затем нажать ЛКМ по кнопке «**Сохранить**»

.

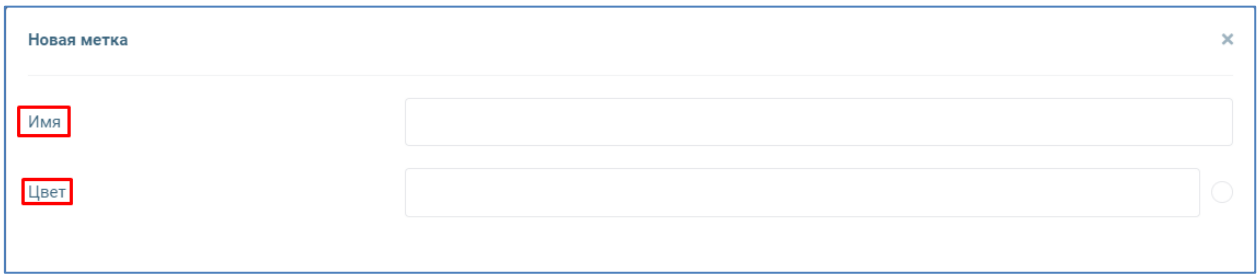


Рисунок 87 – Пример окна «Новая метка»

Созданный тег отображается последним в списке.

5.3.2 Изменение тега

Для редактирования существующего тега необходимо нажать ЛКМ по кнопке существующего тега, в появившемся окне выбрать «**Изменить».**

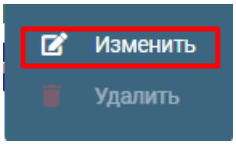


Рисунок 88 – Пример кнопки «**Изменить**»

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

В появившемся окне внести необходимые правки в наименование и цвет, затем нажать ЛКМ по кнопке «**Сохранить**»

.

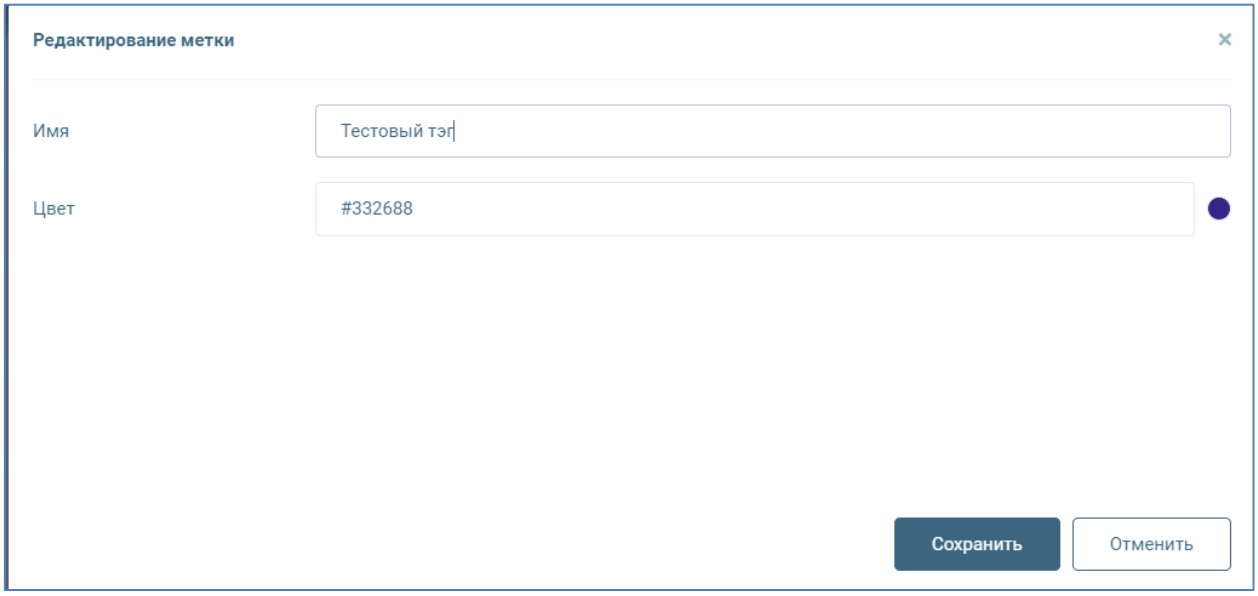


Рисунок 89 – Пример окна «Редактирование метки»

5.3.3 Удаление тега

Для редактирования существующего тега необходимо нажать ЛКМ по кнопке существующего тега, в появившемся окне выбрать «Удалить».

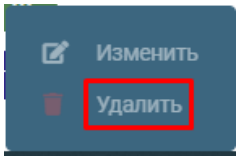


Рисунок 90 – Пример кнопки «Удалить»

В появившемся окне нажать ЛКМ по кнопке «Удалить».

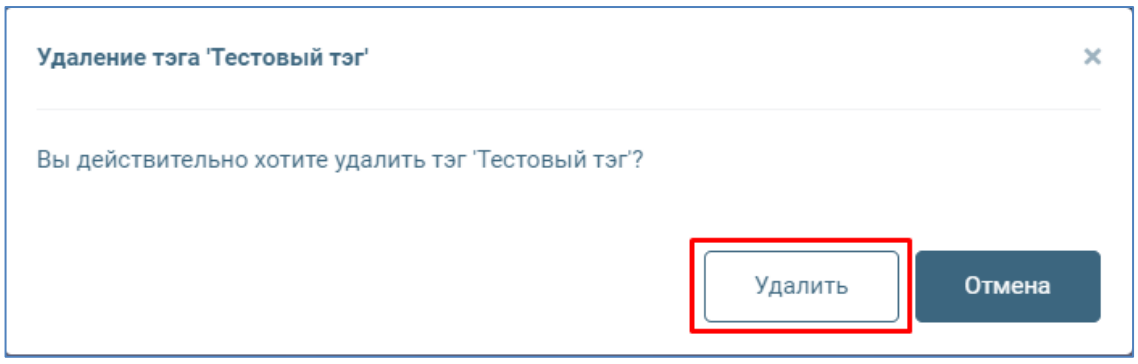


Рисунок 91 – Пример удаления тега

5.4 Выбор нескольких тегов

Для одновременного выбора нескольких тегов необходимо нажать на них левой клавишей мыши при зажатой клавише **Ctrl** на клавиатуре. При этом выбранные теги будут выведены в раздел рядом со строкой поиска, где доступно их отключение, и подсвечены в списке тегов.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

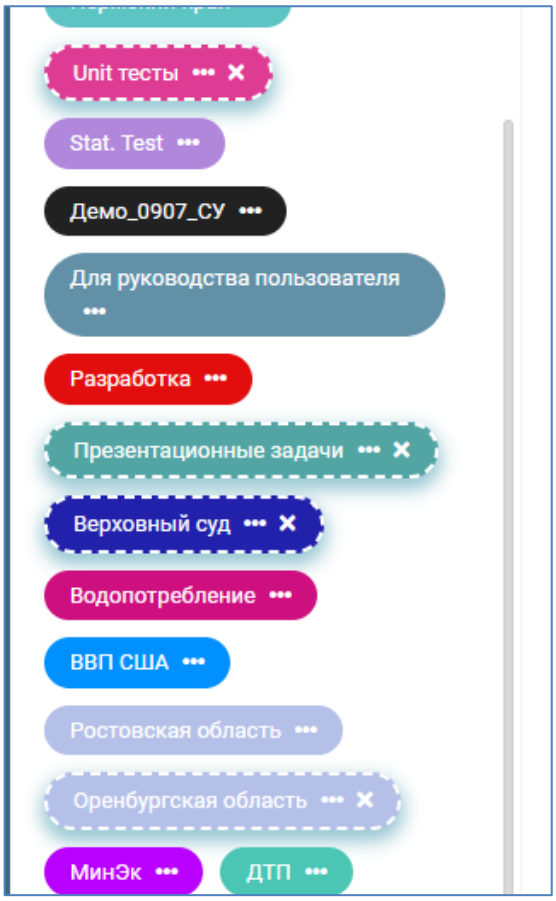


Рисунок 92 – Пример выбора нескольких тегов

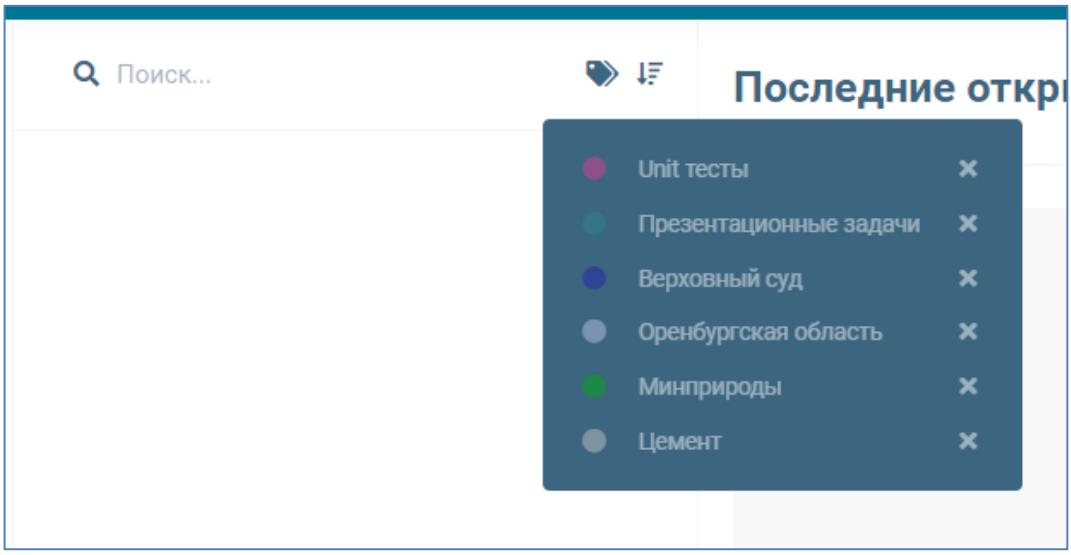


Рисунок 93 – Пример отображния тегов

5.4.1 Тег «Мои задачи»

Данный тег считается системным. Отображается первым в списке тегов.

У пользователей нет возможности изменить или удалить этот тег.

Тегом «Мои задачи» отмечаются задачи, которые были созданы или скопированы конкретным пользователем.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

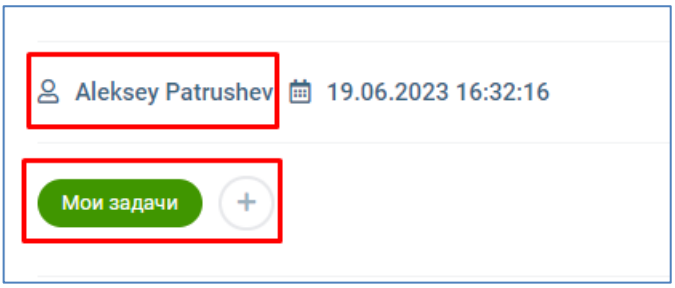


Рисунок 94 – Пример тега «Мои задачи»

5.5 Окно «О приложении»

Для вызова окна «О приложении» необходимо нажать на значок в правом верхнем углу.

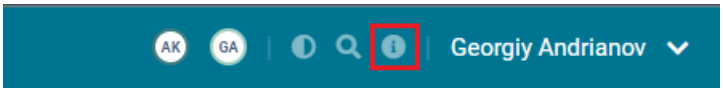


Рисунок 95 – Пример значка «О приложении»

В окне «О приложении» отображаются

1. Наименование – наименование приложения. Может быть изменено при сборке приложения.
2. Версия – версия клиента.
3. Версия API – версия сервера.
4. Идентификатор окружения – идентификатор окружения, используется, например, для управления воркерами. Может быть изменен в настройках сервера.
5. Версия репозитория – версия базы данных приложения (репозитория).
6. Разработчик – информация о разработчике приложения.
7. Адрес формы обратной связи.
8. Сведения о регистрации ЦП УВП.
9. Поддерживаемые браузеры.
10. Перечень используемых компонентов – перечень библиотек и сервисов, использованных при разработке.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6 Библиотека блоков

6.1 Визуализаторы (библиотека)

6.1.1 Диаграммы

6.1.1.1 Блок «Визуализатор JavaScript»

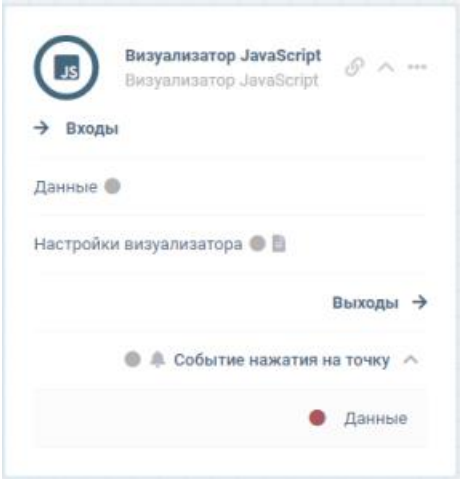


Рисунок 96 – Пример блока «Визуализатор JavaScript»

6.1.1.1.1 Назначение

Блок позволяет выводить в приложении визуализатор на основании написанного JavaScript кода.

6.1.1.1.2 Настройки блока

Название параметра	Тип параметра	Обязательный	Описание	Пример
Компонент	Выпадающий список	Да	Меню с выбором компонента, на котором будет происходить отрисовка визуализатора.	Возможные варианты: div canvas
Код для выполнения	JavaScript	Да	Описание визуализатора, а так же всех связанных с ним событий, на языке JavaScript.	
Скрытый	Флаг	Нет	Если флаг включен - визуализатор не будет отображаться на пресете в режиме просмотра.	
Выполнять в момент открытия пресета	Флаг	Нет	Параметр доступен, если включен флаг "Скрытый" Если флаг включен - код блока выполнится в	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

			момент открытия пресета.	

6.1.1.1.3 Группа «Общие настройки блока»

Название параметра	Тип параметра	Обязательный	Описание	Пример
Наименование блока	Строка	Нет	Наименование блока, которое отображается на графе. В панели размещенных блоков возможен по нему поиск	
Заголовок блока	Строка	Нет	Наименование визуализатора, которое будет отображаться в пользовательском приложении, в заголовке блока. Возможно форматирование с использованием JINJA-шаблонов. Поле доступно только для блоков-визуализаторов	
Ссылка на документацию	Строка	Нет	Поле со ссылкой на документацию. Можно изменить на произвольную пользовательскую ссылку. При клике на иконку у названия блока будет осуществлен на эту ссылку. При замене на пользовательскую ссылку – ссылка на документацию будет перезаписана	
Описание блока	Строка	Нет	Поле для пользовательского описания блока, можно указать произвольный текст. Поле является сворачиваемым и по умолчанию свёрнуто.	
Ссылка по клику на заголовок визуализатора	Строка	Нет	Поле с произвольной пользовательской ссылкой, переход на нее осуществляется при клике ЛКМ на заголовок визуализатора. Может быть как абсолютной, так и относительной	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
			Поле доступно только для блоков-визуализаторов	
Тип открытия ссылки	Выпадающий список	Нет	Регулирует поведение для параметра «Ссылка по клику на заголовок визуализатора»: • В новом окне – ссылка открывается в новой вкладке браузера; • В этом же окне – ссылка открывается в текущей вкладке браузера. Поле доступно только для блоков-визуализаторов	
Пропуск расчета блока	Флаг	Нет	Пропускать или нет блок при расчете. При включенном флаге во время расчета графа блок будет пропущен. При этом его все еще можно посчитать через расчет блока и цепочки. Блоки, расположенные после текущего, будут рассчитаны Для блоков типа «Соединение с БД» поле недоступно	
Кэширование результатов расчета блока	Флаг	Нет	Сохранять или нет выходы блока при расчете графа. Если флаг включен – результаты расчета блока будут использоваться даже в случае если блок выключен из расчета. При выключенном флаге – блоки, идущие за текущим, не получат на вход данных в случае если блок пропускается во время расчета.	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
			Для блоков типа «Соединение с БД» поле недоступно	
Пропуск события по умолчанию	Флаг	Нет	Выключает расчет блока с событием по умолчанию, при открытии его визуализатора в пользовательском приложении. Поле доступно только для блоков-визуализаторов у которых есть событие по умолчанию	

6.1.1.1.4 Входы

Название входа	Тип	Обязательный	Описание
Данные	Ряд данных	Нет	Массив данных с типом «Ряд данных» или «Объект»
Настройки визуализатора	Строка	Нет	Описание визуализатора, обычно строка в формате JSON, при передаче в блок переопределяет значение «Настройки визуализатора» из параметров

Кроме указанных выходов, так же доступно создание динамических входов.

Для создания входа необходимо нажать на знак + в разделе входов блока. Для каждого из входов задается следующий перечень полей:

Название входа	Тип	Обязательный	Описание
Идентификатор	Строка	Да	Идентификатор входа – используется внутри блока, для получения доступа к данным входа. Идентификаторы **data** и **options** зарезервированы для уже существующих входов
Имя	Строка	Да	Имя входа для отображения в списке входов блока
Описание	Строка	Нет	Дополнительная информация с описанием входа
Тип	Выпадающий список	Да	Тип данных, которые будут приходить на вход. Нужен для построения корректных связей с другими блоками. По умолчанию установлен тип "Объект".

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название входа	Тип	Обязательный	Описание
Обязательный	Флаг	Да	Обязательность входа. Если вход обязательный и на него не пришло данных – блок не будет рассчитан с ошибкой
Массив	Флаг	Да	Является ли вход массивом данных

6.1.1.1.5 Выходы

В блоке нет выходов.

6.1.1.1.6 События блока

Название выхода	Тип	Описание
Событие нажатия на точку	JSON	Используется для динамического обновления связанных визуализаторов, при выборе конкретной точки на визуализаторе

6.1.1.1.7 Описание работы блока

Блок используется для построения сложных визуализаторов, в том числе с анимацией, 3D и т.п. Настройки визуализатора могут быть заданы, как в настройках блока, так и переданы на вход блока. Настройки с входа блока имеют приоритет (будут использованы именно они).

6.1.1.1.8 Использования шаблонизатора Jinja

Для применения шаблона необходимо понимать структуру данных, которые пришли на вход.

Первым параметром необходимо указывать идентификатор входа с данными.

Идентификатором для входа «**Данные**» является ключ **input**.

Вход «**Настройки визуализатора**» недоступен в качестве данных для шаблона.

6.1.1.1.9 Описание работы визуализатора

Блок отображает сформированный в соответствии с входными данными визуализатор.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

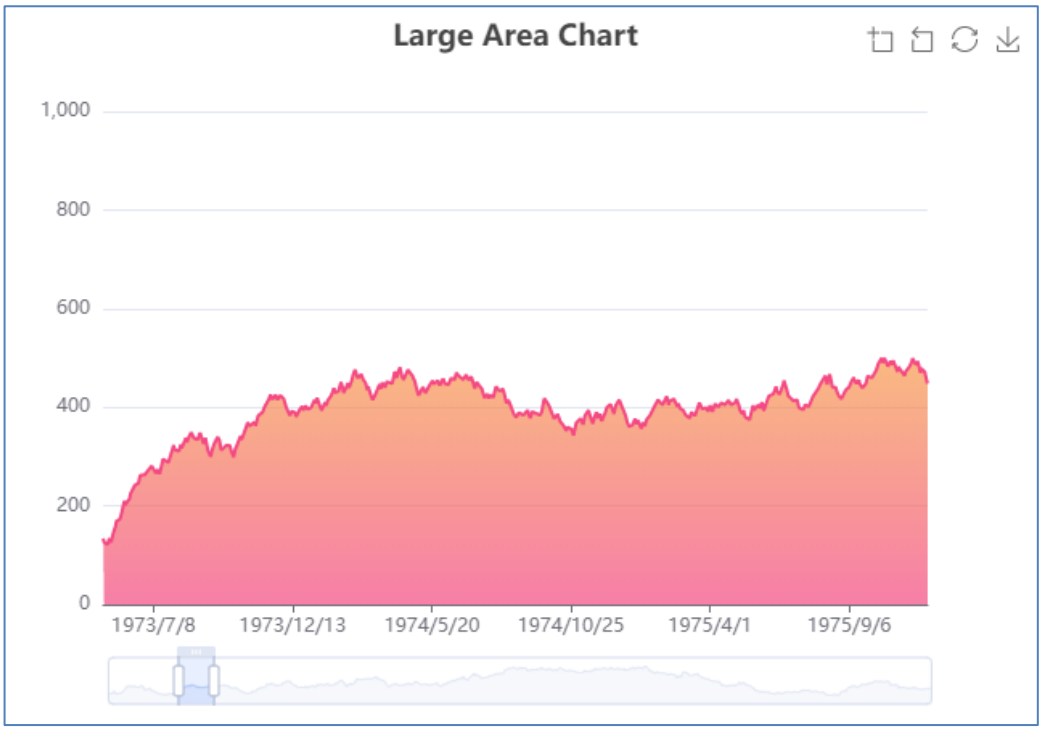


Рисунок 97 – Внешний вид блока «Визуализатор JavaScript»

6.1.1.1.9.1 Событие нажатия на точку

Если в настройках блока указать событие нажатия на точку, то в момент нажатия на выход блока уйдет информация, заданная в настройках (содержание информации определяет пользователь, работающий с блоком).

6.1.1.1.10 Проверка безопасности прикладного кода

При изменении кода внутри блока JS выполняется его проверка на соответствие ограничениям безопасности. В блоке проверяются строковые значения, которые могут содержать потенциально вредоносный код HTML. Проверка выполняется санитайзером HTML (см. п. 4.5.1).

6.1.1.1.11 Настройки визуализатора

Визуализатор не имеет специфических настроек. Общий список приведен в п. 4.2.1.3 «Общие настройки в режиме редактирования пресета».

6.1.1.1.12 Дополнительные ссылки

Документация Jinja: https://jinja.palletsprojects.com/en/3.0.x/

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.1.1.2 Блок «Диаграмма»

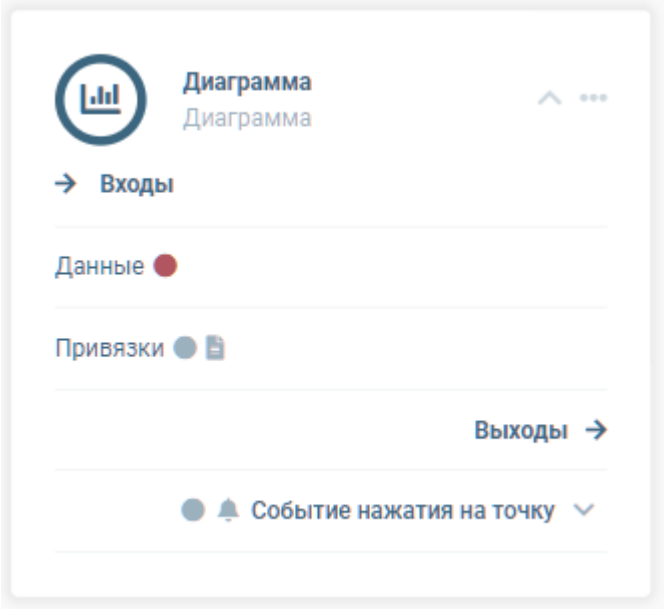


Рисунок 98 – Пример блока «Диаграмма»

6.1.1.2.1 Назначение

Блок позволяет формировать визуализатор диаграммы на основе массива рядов, приходящих из другого блока.

6.1.1.2.2 Особенность

Если диаграмма стоит после события, например изменения отметки в таблице, и после этой диаграммы так же есть блоки, которые должны считаться по событию нажатия – в таком случае диаграмма, после обновления по отметке в таблице, будет пересчитана с тем рядом, который был выбран до нажатия в таблице. Если такого ряда после обновления визуализатора нет – будет эмулировано нажатие на первый ряд. Если рядов нет вообще – события не будет.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.1.1.2.3 Настройки блока

Название параметра	Тип параметра		Описание	Пример
Группа "Специфические настройки блока"				
Тип диаграммы	Выпадающий список	Да	Выбор типа диаграммы. Возможные варианты: • Линейная • Столбчатая • С областями • Смешаная • Круговая • Радар • Горизонтальная • Кольцевая • Каскадная По умолчанию используется "Линейная".	
Накопление	Флаг	Нет	Отображать диаграмму с накоплением.	
Сортировка	Выпадающий список	Да	Выбор способа сортировки рядов данных. Возможные варианты: • Без сортировки • По дате/алфавиту (убывание) • По дате/алфавиту (возрастание) • По значению (убывание) • По значению (возрастание) По умолчанию используется "**По дате/алфавиту (возрастание)**".	
Количество знаков после запятой	Целое число	Да	Настройка количества знаков после запятой у данных, которые отобразятся на диаграмме	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра		Описание	Пример
Масштабирование данных	Флаг	Нет	Включить масштабирование данных	
Количество отображаемых точек	Целое число	Нет	Отображается, если включена настройка "**Масштабирование данных**". Количество отображаемых точек при включенном масштабировании данных.	
Группа "Легенда"				
Отображать легенду	Флаг	Нет	Включение или отключение отображения легенды.	
Размещение легенды	Выпадающий список	Нет	Выбор расположения легенды на диаграмме. Возможные варианты: • Вверху • Слева • Внизу • Справа По умолчанию используется "**Внизу**".	
Размер шрифта	Целое число	Нет	Размер шрифта легенды диаграммы.	
Группа "Оси данных"				
Маска оси времени	Строка	Нет	Настройка отображения даты в диаграмме, используется только для временных рядов. Для использования маски, пришедшей вместе с настройками временных рядов, необходимо оставить поле пустым	Исходная дата: 20 декабря 2015 года Маска: %Y-%m-%d - указывается в параметре Отображаемая дата: 2015-12-20 - в таком виде дата отобразится на диаграмме

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра		Описание	Пример
			Маска используется только для отображения временной шкалы в нужном для пользователя формате. Для корректной работы совместно с блоком "Анимация", дата должна быть указана в формате "%d.%m.%Y" Дата указывается в формате Python.	
Общие настройки стилей рядов				
Настройки, указанные ниже, применяются для всех рядов на диаграмме				
Вид линии	Выпадающий список	Нет	Выбор вида линии на диаграмме. Возможные варианты: • Прямая - линия не скругляется в точках • Сглаженная - линия скругляется в точках По умолчанию используется "**Прямая**".	
Вид раскраски	Выпадающий список	Нет	Выбор вида раскраски ряда диаграммы. Возможные варинаты: • Отсутствует • Цветовой градиент • Градиент прозрачности По умолчанию используется "**Отсутствует**".	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра		Описание	Пример
Начальный цвет	Настройка цвета	Нет	Отображается, если выбран вид раскраски "**Цветовой градиент**". Выбор начального цвета для градиента.	
Конечный цвет	Настройка цвета	Нет	Отображается, если выбран вид раскраски "**Цветовой градиент**". Выбор конечного цвета для градиента.	
Цвет	Настройка цвета	Нет	Отображается, если выбран вид раскраски "**Градиент прозрачности**". Выбор цвета для градиента.	
Каскадная диаграмма - начальный цвет	Настройка цвета	Нет	Отображается, если выбран тип диаграммы "**Каскадная**". Начальный цвет каскадной диаграммы.	
Каскадная диаграмма - убывающий цвет	Настройка цвета	Нет	Отображается, если выбран тип диаграммы "**Каскадная**". Убывающий цвет каскадной диаграммы.	
Каскадная диаграмма - растущий цвет	Настройка цвета	Нет	Отображается, если выбран тип диаграммы "**Каскадная**". Растущий цвет каскадной диаграммы.	
Выводить линию	Флаг	Нет	Отображать или нет линию на диаграмме	
Тип линии или границы	Выпадающий список	Нет	Выбор типа отображения линии на диаграмме. Возможные варианты: • Сплошная линия • Штриховая линия • Линия из точек • Штрихпунктирная линия • Штрихпунктирная линия с двумя точками	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра		Описание	Пример
			По умолчанию используется "**Сплошная линия**".	
Толщина линии, пт	Целое число	Нет	Толщина линии на диаграмме. Минимальное значение - 1, максимальное - 10.	
Тип маркера	Выпадающий список	Нет	Тип маркера на диаграмме Возможные варианты: • Круг • Треугольник • Квадрат • Квадрат с загругленными углами • Ромб • Крестик • Перекрестие • Звездочка • Линия • Черта По умолчанию используется "**Круг**".	
Размер маркера	Целое число	Нет	Размер маркера на диаграмме	
Показывать подписи данных	Флаг	Нет	Показывать или нет подписи данных	
Подписи данных - положение	Выпадающий список	Нет	Местоположение подписей данных. Возможные значения: • В центре • Слева • Сверху • Справа	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра		Описание	Пример
			• Снизу По умолчанию используется "**В центре**".	
Подписи данных - размер шрифта	Целое число	Нет	Размер шрифта подписей данных	
Подписи данных - цвет шрифта	Настройка цвета	Нет	Цвет шрифта подписей данных	
Группа "Цвета точек для круговой диаграммы"				
В группе задаются цвета точек для круговой и кольцевой диаграммы. В случае отсутствия (или если количество точек будет больше, чем выбрано цветов) - будут использованы цвета по умолчанию.				
Название точки	Строка	Нет	Название точки с выбраным цветом. Используется только в настройках блока.	
Цвет	Настройка цвета	Нет	Выбор цвета для точки.	
Группа "Ряды данных"				
В группе задаются настройки для отдельных рядов данных. Номер настройки определяет номер ряда, к которому будут применены данные настройки. Для рядов, у которых не указаны настройки, будут применены "**Общие настройки стилей рядов**"				
Название ряда	Строка	Нет	Название ряда данных. Для использования названия из привязок (см. ниже) оставить значение данного атрибута пустым.	
Отображать ряд	Флаг	Нет	Отобразить или скрыть ряд в настройках диаграммы. Для передачи как часть данных (в условном форматировании) нужно задать атрибут style.showRow	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра		Описание	Пример
			Так как условное форматирование задается для точки ряда, а отображение ряда - общее, возможно задание одного форматирования для одной точки. Если будет передано несколько значений, использоваться будет последнее.	
Тип ряда	Выпадающий список	Нет	Тип ряда данных. Используется только для диаграммы с типом "Смешанная". Возможные значения: • По умолчанию • Линия • Столбец По умолчанию используется "**По умолчанию**".	
Отображение выделения	Флаг	Нет	Отображается, если выбран тип ряда "**Столбец**" Выделение столбца при клике на нем.	
Горизонтальное смешение	Флаг	Нет	Отображается, если выбран тип ряда "**Столбец**" Отображение столбцов со смещением.	
Вид линии	Выпадающий список	Нет	Выбор вида линии на диаграмме. Возможные варианты: • Прямая - линия не скругляется в точках • Сглаженная - линия скругляется в точках	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра		Описание	Пример
			По умолчанию используется "**Прямая**".	
Цвет	Настройка цвета	Нет	Цвет конкретного ряда данных.	
На дополнительную ось	Флаг	Нет	Отображение ряда данных на дополнительной оси.	
Ряды данных. Выводить линию	Флаг	Нет	Отображение линии ряда для линейной диаграммы.	
Тип линии или границы	Выпадающий список	Нет	Выбор типа отображения ряда данных на диаграмме. Возможные варианты: • По умолчанию - использовать "**Общие настройки стилей рядов**" • Сплошная линия • Штриховая линия • Линия из точек • Штрихпунктирная линия • Штрихпунктирная линия с двумя точками По умолчанию используется "**По умолчанию**".	
Толщина линии, пт	Целое число	Нет	Толщина линии на диаграмме. Минимальное значение - 1, максимальное - 10.	
Тип маркера	Выпадающий список	Нет	Тип маркера на диаграмме Возможные варианты: • По умолчанию - использовать "**Общие настройки стилей рядов**" • Круг • Треугольник	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра		Описание	Пример
			• Квадрат • Квадрат с загругленными углами • Ромб • Крестик • Перекрестие • Звездочка • Линия • Черта По умолчанию используется "**По умолчанию**".	
Размер маркера	Целое число	Нет	Размер маркера на диаграмме	
Показывать подписи данных	Флаг	Нет	Показывать или нет подписи данных	
Подписи данных - положение	Выпадающий список	Нет	Местоположение подписей данных. Возможные значения: • По умолчанию - использовать "**Общие настройки стилей рядов**" • В центре • Слева • Сверху • Справа • Снизу По умолчанию используется "**По умолчанию**".	
Подписи данных - размер шрифта	Целое число	Нет	Размер шрифта подписей данных	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра		Описание	Пример
Подписи данных - цвет шрифта	Настройка цвета	Нет	Цвет шрифта подписей данных	
Группа "Коридоры"				
Название коридора	Строка	Нет	Название коридора. Используется только в настройках блока.	
Индекс ряда 1	Целое число	Нет	Индекс ряда для первой границы коридора. Индексы рядов начинаются с **1**	
Индекс ряда 2	Целое число	Нет	Индекс ряда для второй границы коридора Индексы рядов начинаются с **1**	
Цвет	Настройка цвета	Нет	Цвет коридора	
Группа "Привязки"				
Настройка привязки наименований рядов и точек данных, к отдельным атрибутам ряда. Возможно задавать привязки как в рукописной форме (через поле ввода JSON), так и через выбор элементов из выпадающего списка. Переключение осуществляется с помощью кнопки Настройки не синхронизированы, т.е. настройки из рукописной формы не будут применены в выпадающем списке, так и наоборот. Для корректного отображения настроек в выпадающих списках необходимо, чтобы на вход блока приходили данные из расчитанного блока.				
Наименование ряда данных	Выпадающий список	Нет	Ключ атрибута, отвечающего за наименование ряда. Можно выбрать несколько значений. В таком случае, они будут отображаться через разделитель " - ".	
Наименование точек данных	Выпадающий список	Нет	Ключ атрибута, отвечающий за наименование точек. Для временных рядов, в качестве точек используется дата, формат которой определяется маской времени (см. выше).	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра		Описание	Пример
			В случае, если поле не задано, то по умолчанию проставляются ключи точек.	

Описание привязок	JSON	Нет	Используются для отображения корректных наименований точек и рядов с данными.
			
			Возможные параметры:
			
			• **name** - Ключ атрибута, отвечающего за наименование ряда.
			  • Возможно указать, как строковое значение, так и массив из нескольких строк. В таком случае, они будут отображаться через разделитель " - ".
			• **label** - Ключ атрибута, отвечающий за наименование точек. Строковое значение.
			  • Для временных рядов, в качестве точек используется дата, формат которой определяется маской времени (см. выше).
			В случае, если поле не задано, то по умолчанию проставляются ключи точек.
				Пример с единичным наименованием точек
				`{ "label": "region_name", "name": "date" }`
				Пример со множественным наименованием точек
				`{ "label": "region_name", "name": ["date", "pok1"] }`
				Допустимый вариант единичного наименования точек
				`{ "label": "region_name", "name": ["date"] }`

Группа "Ось X"				

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра		Описание	Пример
Начертание	Выпадающий список	Нет	Начертание шрифта надписей. Возможные варианты: • Обычный • Курсив • Полужирный По умолчанию используется "**Обычный**".	
Перенос по словам	Флаг	Нет	Включить перенос надписей по словам	
Поворот надписи	Выпадающий список	Нет	Настройки поворота надписей. Возможные варианты: • Автоматически (по умолчанию) • Отображать горизонтально По умолчанию используется "**Автоматически (по умолчанию)**".	
Подгруппа "Сетка"				
Отображать линии	Флаг	Нет	Отображать линии сетки	
Цвет линии	Настройки цвета	Нет	Цвет линии сетки	
Толщина линии	Вещественное число	Нет	Толщина линии сетки	
Тип штриха	Выпадающий список	Нет	Выбор типа штриха линии. Возможные варианты: Сплошная линия Штриховая линия Линия из точек Штрихпунктирная линия Штрихпунктирная линия с двумя точками	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра		Описание	Пример
			По умолчанию используется "**Сплошная линия**".	
Подгруппа "Деления"				
Количество	Целое число	Нет	Количество делений. При отметке 0 - деления не указываются.	
Пропускать подписи оси	Флаг	Нет	В случае включения - оси диаграмм выводятся подряд.	
Отступ	Целое число	Нет	Отступ делений	
Группа "Ось Y"				
Размер шрифта	Целое число	Нет	Размер шрифта надписей на оси.	
Подгруппа "Сетка"				
Отображать линии	Флаг	Нет	Отображать линии сетки	
Цвет линии	Настройки цвета	Нет	Цвет линии сетки	
Толщина линии	Вещественное число	Нет	Толщина линии сетки	
Тип штриха	Выпадающий список	Нет	Выбор типа штриха линии. Возможные варианты: Сплошная линия Штриховая линия Линия из точек Штрихпунктирная линия Штрихпунктирная линия с двумя точками По умолчанию используется "**Сплошная линия**".	
Подгруппа "Деления"				
Количество	Целое число	Нет	Количество делений. При отметке 0 - деления не указываются.	
Пропускать подписи оси	Флаг	Нет	В случае включения - оси диаграмм выводятся подряд.	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра		Описание	Пример
Отступ	Целое число	Нет	Отступ делений	

6.1.1.2.4 Группа «Общие настройки блока»

Аналогично приведенным в п. 6.1.1.1.3.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.1.1.2.5 Входы

Название входа	Тип	Обязательный	Описание
Данные	Ряд данных	Да	Массив данных с типом "Ряд данных"
Привязки	Строка	Нет	Привязки которые приходят на вход блока используются вместо параметра настроек "Привязки". Имеют наивысший приоритет среди всех привязток. Для динамической перестройки, к примеру, когда таблица является частью ОЛАПа, статичные привязки могут привести к некорректному отображению данных. Поэтому, в таких случаях, в таблицу нужно передавать и привязки. Формат привязок аналогичен описанию из настройки "**Описание привязок**".

6.1.1.2.6 Выходы

Блок не имеет конкретных выходов.

6.1.1.2.7 События

Название выхода	Тип	Описание
События нажатия на точку – Временные ряды	Ряд данных	Данные ряда, на который было осуществено нажатие.

6.1.1.2.8 Настройки визуализатора

Визуализатор не имеет специфических настроек. Общий список приведен в п. 4.2.1.3 «Общие настройки в режиме редактирования пресета».

6.1.1.2.9 Описание работы визуализатора

Чтобы отредактировать данные необходимо нажать кнопку «Редактировать данные» в локальном меню.

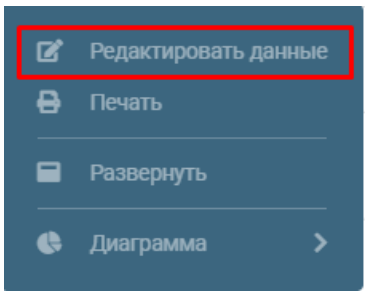


Рисунок 99 – Пример кнопки «Редактировать данные»

Особенность работы диаграммы, стоящей после события

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Если диаграмма стоит после события, например изменение отметки в таблице, и после этой диаграммы так же есть блоки, которые должны считаться по событию нажатия - в таком случае диаграмма, после обновления по отметке в таблице, будет пересчитана с тем рядом который был выбран до нажатия в таблице. Если такого ряда после обновления визуализатора нет - будет эмулировано нажатие на первый ряд. Если рядов нет вообще - события не будет.

6.1.1.3 Блок «Диаграмма Apache ECharts»

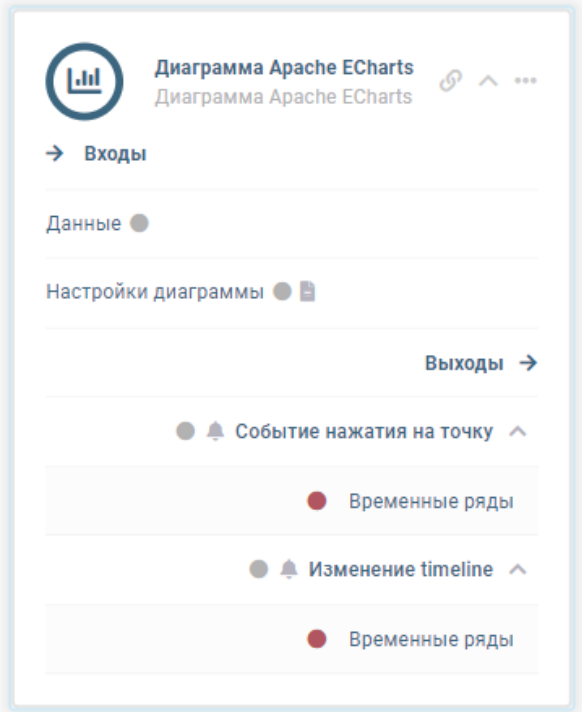


Рисунок 100 – Пример блока «Диаграмма Apache ECharts»

6.1.1.3.1 Назначение

Блок позволяет выводить в приложении диаграмму, использующую возможности Apache ECharts, используется версия 5.1.2 библиотеки.

6.1.1.3.2 Настройки блока

Название параметра	Тип параметра	Обязательный	Описание	Пример
Настройки диаграммы	JSON	Да	Настройки диаграммы в формате JSON, при наличии данных на входе «Настройки диаграммы» игнорируется	`{ "xAxis": { "type": "category", "data": ["Mon", "Tue", "Wed", "Thu", "Fri", "Sat", "Sun"] }, "yAxis": { "type": "value" }, "series": [{ "data": [150, 230, 224, 218, 135, 147, 260], "type": "line" }] }`

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.1.1.3.3 Группа «Общие настройки блока»

Аналогично приведенным в п. 6.1.1.1.3.

6.1.1.3.4 Входы

Название входа	Тип	Обязательный	Описание
Данные	Ряд данных	Нет	Массив данных с типом «Ряд данных» или «Объект»
Настройки диаграммы	Строка	Нет	Описание диаграммы, обычно строка в формате JSON, при передаче в блок переопределяет значение «Настройки диаграммы» из параметров

6.1.1.3.5 Выходы

Блок не имеет конкретных выходов.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.1.1.3.6 События блока

Название выхода	Тип	Описание

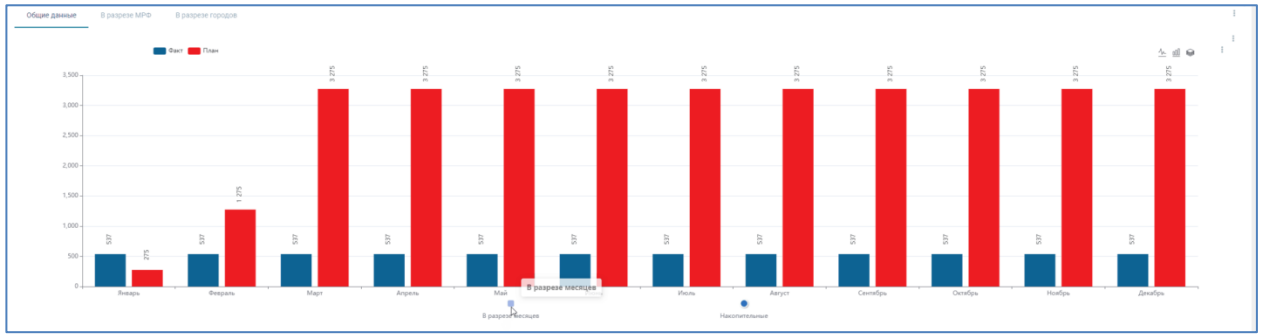
Событие нажатия на точку	Ряд данных	Используется для динамического обновления связанных визуализаторов, при выборе конкретной точки на диаграмме.
		Для корректной работы события необходимо выполнить некоторые дополнительные условия:
		1. Выход считается только если есть соединение с входом «Данные».
		2. Для работы события необходимо в настройках диаграммы (входе или в блоке) описать шаблон Jinja2.
		3. При обработке события данные на выходе будут объектами с входа (они будут подставлены в шаблон).
		4. Порядок рядов важен, т.к. при вызове события будет передан индекс ряда, на котором произошло нажатие.
		Пример шаблона настроек
		`option = {`
		`  title: {`
		`  text: 'sometext'`
		`  },`
		`  tooltip: {`
		`  trigger: 'axis'`
		`  },`
		`  legend: {`
		`  data: data: {{input[0]["some-attr"]}}`
		`  },`
		`  grid: {`
		`  left: '3%',`
		`  right: '4%',`
		`  bottom: '3%',`
		`  containLabel: true`
		`  },`
		`  toolbox: {`
		`  feature: {`
		`  saveAsImage: {}`
		`  }`
		`  },`
		`  xAxis: {`
		`  type: 'category',`
		`  boundaryGap: false,`

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название выхода	Тип	Описание

```
        data: {{input[0]["keys"]}}
    },
    yAxis: {
        type: 'value'
    },
    series: [
        {
            name: 'sometext1',
            type: 'line',
            stack: 'stack',
            data: {{input[0]["vl"]}} // вместо [120, 132, 101, 134, 90, 230, 210]
        },
        {
            name: 'sometext2',
            type: 'line',
            stack: 'stack',
            data: {{input[1]["vl"]}} // вместо data: [220, 182, 191, 234, 290, 330, 310]
        },
        {
            name: 'sometext3',
            type: 'line',
            stack: 'stack',
            data: {{input[2]["vl"]}} // вместо data: [150, 232, 201, 154, 190, 330, 410]
        },
        {
            name: 'sometext4',
            type: 'line',
            stack: 'stack',
            data: {{input[3]["vl"]}} // вместо data: [320, 332, 301, 334, 390, 330, 320]
        },
        {
            name: 'sometext5',
            type: 'line',
            stack: 'stack',
            data: {{input[4]["vl"]}} // вместо data: [820, 932, 901, 934, 1290, 1330, 1320]
        }
    ]
```


© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

<table>
<tr><th>Название выхода</th><th>Тип</th><th>Описание</th></tr>
<tr><td></td><td></td><td>};</td></tr>
<tr><td>Изменение timeline</td><td>JSON</td><td>Используется для динамического обновления связанных визуализаторов в процессе анимации диаграммы ECharts.
 
Для корректной работы события необходимо определенным образом подготовить данные. В частности, входные данные должны быть в виде линейного списка:
<pre>option = {
  ...
  timeline: {
    ...
  }
  options: [
    { series: ... }, // будет передано в событие при нажатии на 1 элемент таймлайна
    { series: ... }, // будет передано в событие при нажатии на 2 элемент таймлайна
    ...
  ]
}</pre>Обратите внимание – данные возвращаются в формате диаграммы, и для дальнейшего использования требуется преобразовать их в ряды таблицы с помощью блока Python.
Подробнее см. примеры:
https://echarts.apache.org/en/option.html#timeline
https://echarts.apache.org/en/api.html#events.timelinechanged</td></tr>
</table>

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.1.1.3.6.1 Описание работы блока

Блок используется для построения сложной визуализации диаграмм, в том числе с анимацией, 3D и т. п. Настройки диаграммы могут быть заданы в настройках блока или переданы на вход, настройки, полученные со входа блока, имеют приоритет. Блок может использовать как переданные ряды с данными и настройки диаграммы, так и одни настройки диаграммы.

В первом случае, в настройках диаграммы будет использоваться шаблонизатор Jinja, пример настроек с использованием шаблонизатора:

```
{
    "xAxis": {
        "type": "category",
        "data": "{{ to_list(input[0]['keys']) }}"
    },
    "yAxis": {
        "type": "value"
    },
    "series": [{
        "data": "{{ to_list(input[0]['vl']) }}",
        "type": "line",
        "options": "{{ to_options(input[0]) }}"
    },
    {
        "data": "{{ to_list(input[1]['vl']) }}",
        "type": "line",
        "options": "{{ to_options(input[1]) }}"
    }]
}
```

Во втором случае все значения для отображения на диаграмме должны присутствовать в настройках диаграммы, для этого может использоваться, например, отдельный блок Python (см. п. 6.9.1), выход которого направляется на вход «Настройки диаграммы».

6.1.1.3.6.2 Порядок генерации кода

В том случае, если в настройках диаграммы используется шаблонизатор Jinja, то в первую очередь будет происходить замена шаблонов на данные, а уже потом – валидация JSON. Таким образом, подобный код (см. ниже) будет не валиден при его написании, но после подстановки шаблонов станет абсолютно корректным:

```
{
    "xAxis": {
        "type": "category",
        "data": [{% for i in input[0]["keys"] %} "{{ i }}" {% if not loop.last
%},{% endif %}{% endfor %}] <=== Заменится на массив строк: ["key1", "key2",...,
"keyN"]
    },
    "yAxis": {
        "type": "value"
    },
    "series": [{
        "data": {{ input[0].vl }}, <=== Заменится на массив значений:
[<value1>, <value2>,..., <valueN>]
        "type": "line"
    }]
}
```


© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.1.1.3.7 Описание работы визуализатора

Визуализатор отображает сформированную в соответствии с входными данными диаграмму.

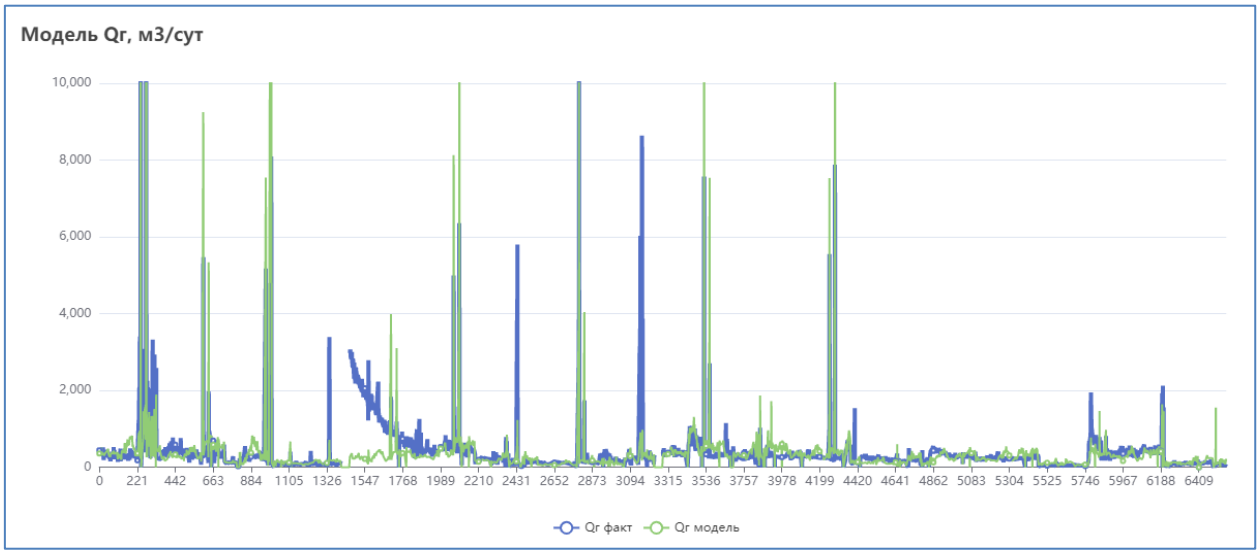


Рисунок 101 – Внешний вид визуализатора «Диаграмма Apache ECharts»

6.1.1.3.7.1 Событие нажатия на точку

Событие позволяет провести расчет отдельной цепочки блоков, в том числе других визуализаторов, например, с целью расшифровки выбранных на диаграмме данных. Для работы события при формировании визуализатора «Диаграмма Apache ECharts» необходимо передать в данные в разделе **series** дополнительный раздел **options**, содержащий информацию о временном ряде, соответствующему ряду данных диаграммы. Подробнее см. п. 6.1.1.3.7.1 «Событие нажатия на точку».

6.1.1.3.7.2 Изменение timeline

При нажатии на timeline (если он добавлен на диаграмму) позволяет запустить событие с данными на указанный элемент таймлайна. При событии из диаграммы будут переданы ряды данных, соответствующие нажатому элементу таймлайна (видимые в данный момент на диаграмме). Подробнее см. п. 6.1.1.3.7.2 «Изменение timeline».

6.1.1.3.8 Настройки визуализатора

Визуализатор не имеет специфических настроек. Общий список приведен в п. 4.2.1.3 «Общие настройки в режиме редактирования пресета».

6.1.1.3.9 Дополнительные ссылки

Сайт проекта Apache ECharts: https://echarts.apache.org/en/index.html

Документация шаблонизатора Jinja2: https://jinja.palletsprojects.com/en/2.11.x/

Настройки диаграммы: Официальная документация Apache ECharts

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.1.1.4 Блок «Диаграмма Ганта»

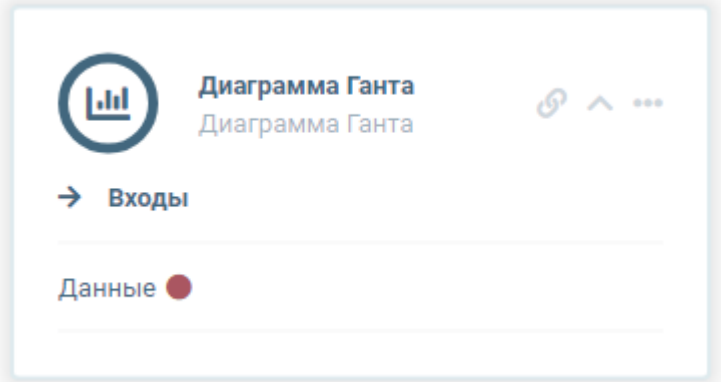


Рисунок 102 – Пример блока «Диаграмма Ганта»

6.1.1.4.1 Назначение

Блок позволяет формировать визуализатор диаграммы Ганта на основе массива объектов, приходящих из другого блока.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.1.1.4.2 Настройки блока

Название параметра	Тип параметра	Описание	Пример
Наименование блока	Строка	Наименование блока	
Заголовок диаграммы	Строка	Наименование блока, которое отобразится в приложении	
Специфические настройки блока			

Параметры привязки	Строка	Настройка соответствий полей диаграммы Ганта с атрибутами объекта
		настройка формируется из:
		обязательных полей ("pID", "pName", "pStart", "pEnd", "pClass", "pComp", "pParent", "pOpen", "pRes")
		необязательных
		зарезервированных ("pLink", "pMile", "pGroup", "pDepend", "pNotes")
		пользовательских (все, что не перечислено п предыдущих пунктах)
		
		"pID" - уникальный ключ (число)
		"pName" - название (строка)
		"pStart" - дата начала (дата)
		"pEnd" - дата окончания (дата)
		"pClass" - цвет ряда (ggroupblack, gtaskblue, gtaskred, gtaskgreen, gtaskyellow, gtaskpurple, gtaskpink) (строка из перечисленных вариантов)
		"pComp" - процент (число)
		"pParent" - id родителя (число)
		"pOpen" - раскрытая\закрытая ветка (число 1\0) по умолчанию раскрыта(1)
		"pRes" - ресурс (строка)
		"pLink" - ссылка (строка)
		"pMile" - если одна дата начала и окончания то ставить в 1 (число)
		"pGroup" - группы (для родителей 1 для всех остальных 0) (число)
		"pDepend" - стрелочка на другой ряд (id ряда куда рисовать стрелку) (число)
		"pNotes" - описание (строка)
		
		Формат такой: "поле в диаграмме": "поле атрибута из "pkey" или "attrs" объекта"
			Пример параметров привязки:
			`{`
			`"pID": "id",`
			`"pName": "name",`
			`"pStart": "ddstart",`
			`"pEnd": "ddend",`
			`"pClass": "eclass",`
			`"pComp": "comp",`
			`"pParent": "parent",`
			`"pGroup": "egroup",`
			`"pRes": "res",`
			`"Актер": "actor"`
			`}`

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Описание	Пример
		Если поле не системное: "как будет называться в диаграмме": "поле атрибута из "pkey" или "attrs" объекта"	
Выходные дни	Флаг	Отображать выходные дни или нет	
Начальная дата	Дата	С какой даты будет отображаться график Формат даты DD.MM.YYYY По умолчанию пустой	
Последняя дата	Дата	По какую дату будет отображаться график Формат даты DD.MM.YYYY По умолчанию пустой	
Динамика отображения дат	Выпадающий список	Варианты отображений дат на диаграмме Ганта	
Выводить часовую динамику	Флаг	Отображение часовой динамики. **Важно**: отображение часовой динамики может привести к существенному замедлению работы визуализатора	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.1.1.4.3 Входы

Название входа	Тип	Обязательный	Описание
Данные	Ряд данных	Да	Массив данных с типом «Ряд данных» или «Объект»

6.1.1.4.4 Выходы

Блок не имеет конкретных выходов.

6.1.1.4.5 Описание работы визуализатора

Помимо стандартных, визуализатор позволяет задать следующие настройки:

Название параметра	Тип	Обязательный	Описание
Прозрачность при наведении на период	Вещественное число	Нет	Изменение прозрачности подсказки при наведении на период.

6.1.1.5 Блок «Спарклайн»

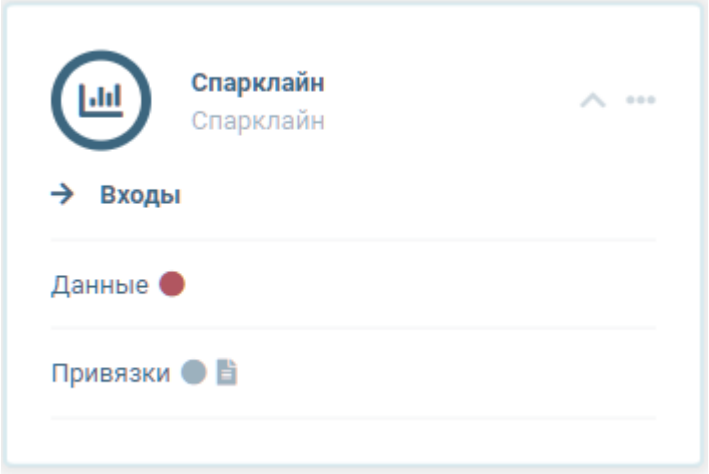


Рисунок 103 – Пример блока «Спарклайн»

6.1.1.5.1 Назначение

Блок позволяет формировать визуализатор «спарклайн» на основе массива рядов, приходящих из другого блока.

Данный блок похож на блок «Диаграмма» (п. 6.1.1.2), но с несколько меньшим количеством настроек.

6.1.1.5.2 Настройки блока

Название параметра	Тип параметра	Обязательный	Описание	Пример
Группа "Специфические настройки блока"				
Тип диаграммы	Выпадающий список	Да	Выбор типа диаграммы. • Линейная • Столбчатая • С областями • Смешаная • Круговая	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
			• Радар • Горизонтальная • Кольцевая • Каскадная По умолчанию используется «Линейная»	
Группа "Ряды данных"				
В группе задаются настройки для отдельных рядов данных. Номер настройки определяет номер ряда, к которому будут применены данные настройки.				
Название ряда	Строка	Нет	Название ряда данных. Для использования названия из привязок (см. ниже) оставить значение данного атрибута пустым.	
Тип ряда	Выпадающий список	Нет	Тип ряда данных. Используется только для диаграммы с типом "Смешанная". Возможные значения: По умолчанию Линия Столбец По умолчанию используется "По умолчанию".	
Цвет	Настройка цвета	Нет	Цвет конкретного ряда данных.	
Группа "Привязки"				
Настройка привязки наименований рядов и точек данных, к отдельным атрибутам ряда. Возможно задавать привязки как в рукописной форме (через поле ввода JSON), так и через выбор элементов из выпадающего списка. Переключение осуществляется с помощью кнопки Настройки не синхронизированы, т.е. настройки из рукописной формы не будут применены в выпадающем списке, так и наоборот. Для корректного отображения настроек в выпадающих списках необходимо, чтобы на вход блока приходили данные из рассчитанного блока.				
Наименование ряда данных	Выпадающий список	Нет	Ключ атрибута, отвечающего за наименование ряда. Можно выбрать несколько значений. В таком случае, они	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
			будут отображаться через разделитель " - ".	
Наименование точек данных	Выпадающий список	Нет	Ключ атрибута, отвечающий за наименование точек. Для временных рядов, в качестве точек используется дата, формат которой определяется маской времени (см. выше). В случае, если поле не задано, то по умолчанию проставляются ключи точек.	

Описание привязок	JSON	Нет	Используются для отображения корректных наименований точек и рядов с данными.
			
			Возможные параметры:
			
			name - Ключ атрибута, отвечающего за наименование ряда. Возможно указать, как строковое значение, так и массив из нескольких строк. В таком случае, они будут отображаться через разделитель " - ".
			label - Ключ атрибута, отвечающий за наименование точек. Строковое значение. Для временных рядов, в качестве точек используется дата, формат которой определяется параметром
				Пример с единичным наименованием точек
				`{ "label": "region_name", "name": "date" }`
				Пример со множественным наименованием точек
				`{ "label": "region_name", "name": ["date", "pok1"] }`
				Допустимый вариант единичного наименования точек
				`{ "label": "region_name",`

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
			временных рядов в самом ряде. В случае, если поле не задано, то по умолчанию проставляются ключи точек.	"name": ["date"] }

6.1.1.5.3 Группа «Общие настройки блока»

Аналогично приведенным в п. 6.1.1.1.3.

6.1.1.5.4 Входы

Название входа	Тип	Обязательный	Описание
Данные	Ряд данных	Да	Массив данных с типом "Ряд данных"
Привязки	Строка	Нет	Привязки которые приходят на вход блока используются вместо параметра настроек "Привязки". Имеют наивысший приоритет среди всех привязок. Для динамической перестройки, к примеру, когда таблица является частью ОЛАПа, статичные привязки могут привести к некорректному отображению данных. Поэтому, в таких случаях, в таблицу нужно передавать и привязки. Формат привязок аналогичен описанию из настройки "Описание привязок".

6.1.1.5.5 Выходы

Блок не имеет конкретных выходов.

6.1.1.5.6 События

В блоке нет событий.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.1.1.5.7 Настройки визуализатора

Визуализатор не имеет специфических настроек. Общий список приведен в п. 4.2.1.3 «Общие настройки в режиме редактирования пресета».

6.1.1.6 Блок «Спидометр/Кольцо»

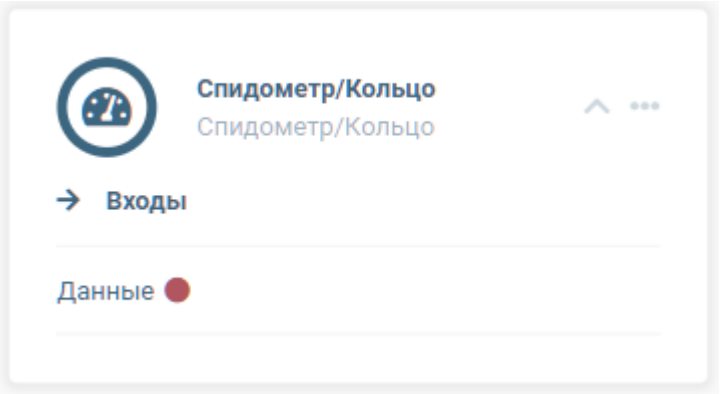


Рисунок 104 – Пример блока «Спидометр/Кольцо»

6.1.1.6.1 Назначение

Блок позволяет формировать визуализатор "Спидометр" или визуализатор "Кольцо" на основе массива рядов или объектов. Так как для блока требуется только одно значение, то в случае ряда берется последний его элемент.

6.1.1.6.2 Настройки блока

Название параметра	Тип параметра	Обязательный	Описание	Пример
Группа "Специфические настройки блока"				
Тип диаграммы	Выпадающий список	Да	Тип диаграммы для отображения. Возможные варианты значения: • Спидометр - для отображения визуализатора "Спидометр" • Кольцо - для отображения визуализатора "Кольцо" По умолчанию используется "**Спидометр**".	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
Количество знаков после запятой	Целое число	Нет	Только для диаграммы с типом "**Спидометр**". Настройка количества знаков после запятой у данных, которые отобразятся на диаграмме	
Группа "Настройка диапазонов"				
Наименование диапазона	Строка	Нет	Наименование диапазона. Используется только в настройках блока.	
Значение диапазона	Целое число	Да	Значение диапазона	
Цвет диапазона	Настройка цвета	Да	Цвет диапазона	
Группа "Настройки связи"				
Управляемый блок	Флажок	Нет	При включении элемента - диаграмма будет зависеть от управляемого блока	
Наименование	Строка	Нет	Ключ управляемого блока	

6.1.1.6.3 Группа «Общие настройки блока»

Аналогично приведенным в п. 6.1.1.1.3.

6.1.1.6.4 Входы

Название входа	Тип	Обязательный	Описание
Данные	Ряд данных	Да	Массив данных с типом "Ряд данных"

6.1.1.6.5 Выходы

В блоке нет выходов.

6.1.1.6.6 События

В блоке нет событий.

6.1.1.6.7 Настройки визуализатора

Визуализатор не имеет специфических настроек. Общий список приведен в п. 4.2.1.3 «Общие настройки в режиме редактирования пресета».

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.1.1.7 Блок «Тепловая карта»

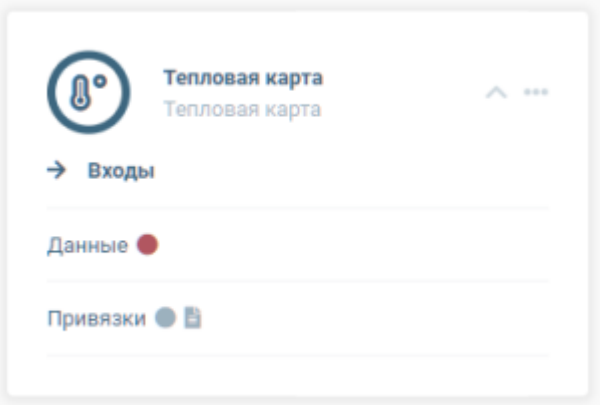


Рисунок 105 – Пример блока «Тепловая карта»

6.1.1.7.1 Назначение

Блок позволяет формировать визуализатор «Тепловая карта» на основе массива рядов, , приходящих из другого блока.

6.1.1.7.2 Настройки блока

Название параметра	Тип параметра	Обязательный	Описание	Пример
Наименование диапазона	Строка	Нет	Наименование диапазона. Используется только в настройках блока.	
Значение диапазона	Целое число	Да	Значение диапазона	
Цвет диапазона	Настройка цвета	Да	Цвет диапазона	
Отображать значения	Флажок	Нет	Отображать или нет значения данных на визуализаторе	
Количество знаков после точки	Целое число	Нет	Настройка количества знаков после точки у данных, которые отобразятся на диаграмме	
Маска оси времени	Строка	Нет	Настройка отображения даты в диаграмме, используется только для временных рядов. Для использования маски, пришедшей вместе с настройками временных рядов, необходимо оставить поле пустым\ Маска используется только для	Исходная дата: 20 декабря 2015 года Маска: %Y-%m-%d - указывается в параметре Отображаемая дата: 2015-12-20 - в таком виде дата отобразится на диаграмме

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
			отображения временной шкалы в нужном для пользователя формате. Для корректной работы совместно с блоком "Анимация", дата должна быть указана в формате "%d.%m.%Y" Дата указывается в формате Python.	
Привязки	JSON		Настройка привязки наименований рядов и точек данных, к отдельным атрибутам ряда. Возможно задавать привязки как в рукописной форме (через поле ввода JSON), так и через выбор элементов из выпадающего списка. Переключение осуществляется с помощью кнопки Настройки не синхронизированы, т.е. настройки из рукописной формы не будут применены в выпадающем списке, так и наоборот. Для корректного отображения настроек в выпадающих списках необходимо, чтобы на вход блока приходили данные из расчитанного блока.	
Наименование ряда данных	Выпадающий список	Нет	Ключ атрибута, отвечающего за наименование ряда. Можно выбрать несколько значений. В таком случае, они	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
			будут отображаться через разделитель " - ".	
Наименование точек данных	Выпадающий список	Нет	Ключ атрибута, отвечающий за наименование точек. Для временных рядов, в качестве точек используется дата, формат которой определяется маской времени (см. выше). В случае, если поле не задано, то по умолчанию проставляются ключи точек.	

Описание привязок	JSON	Нет	Используются для отображения корректных наименований точек и рядов с данными.
			
			Возможные параметры:
			
			name - Ключ атрибута, отвечающего за наименование ряда. Возможно указать, как строковое значение, так и массив из нескольких строк. В таком случае, они будут отображаться через разделитель " - ".
			label - Ключ атрибута, отвечающий за наименование точек. Строковое значение.
			Для временных рядов, в качестве точек используется дата, формат которой определяется маской времени (см. выше).
			
			В случае, если поле не задано, то по
				Пример с единичным наименованием точек
				`{ "label": "region_name", "name": "date" }`
				Пример со множественным наименованием точек
				`{ "label": "region_name", "name": ["date", "pok1"] }`
				Допустимый вариант единичного наименования точек Collapse source
				`{ "label": "region_name", "name": ["date"] }`

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
			умолчанию проставляются ключи точек.	

6.1.1.7.3 Группа «Общие настройки блока»

Аналогично приведенным в п. 6.1.1.1.3.

6.1.1.7.4 Входы

Название входа	Тип	Обязательный	Описание
Данные	Ряд данных	Да	Массив данных с типом "Ряд данных"
Привязки	Строка	Нет	Привязки которые приходят на вход блока используются вместо параметра настроек "Привязки". Имеют наивысший приоритет среди всех привязток. Для динамической перестройки, к примеру, когда таблица является частью ОЛАПа, статичные привязки могут привести к некорректному отображению данных. Поэтому, в таких случаях, в таблицу нужно передавать и привязки. Формат привязок аналогичен описанию из настройки "Описание привязок".

6.1.1.7.5 Выходы

В блоке нет выходов.

6.1.1.7.6 События

В блоке нет событий.

6.1.1.7.7 Настройки визуализатора

Визуализатор не имеет специфических настроек. Общий список приведен в п. 4.2.1.3 «Общие настройки в режиме редактирования пресета».

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.1.1.8 Блок «Точечная диаграмма»

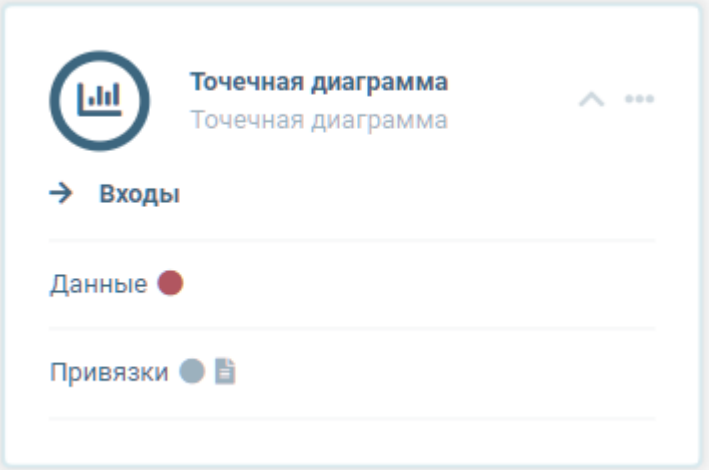


Рисунок 106 – Пример блока «Точечная диаграмма»

6.1.1.8.1 Назначение

Блок позволяет формировать визуализатор диаграммы на основе массива рядов. Каждый ряд представляет собой одну точку. Координаты (x, y) берутся из последних двух точек ряда соответственно.

6.1.1.8.2 Настройки блока

Название параметра	Тип параметра	Обязательный	Описание	Пример
Группа "Специфические настройки блока"				
Количество знаков после запятой	Целое число	Нет	Настройка количества знаков после запятой у данных, которые отобразятся на диаграмме	
Диаметр точки	Целое число	Нет	Настройка диаметра точки на диаграмме	
Группа "Привязки"				
Настройка привязки наименований, цветов групп, а также наименование точек и осей, к отдельным атрибутам ряда. Возможно задавать привязки как в рукописной форме (через поле ввода JSON), так и через выбор элементов из выпадающего списка. Переключение осуществляется с помощью кнопки Настройки не синхронизированы, т.е. настройки из рукописной формы не будут применены в выпадающем списке, так и наоборот. Для корректного отображения настроек в выпадающих списках необходимо, чтобы на вход блока приходили данные из расчитанного блока.				
Атрибут для группировки	Выпадающий список	Да	Ключ атрибута, отвечающего группировку значений. Обязательное поле.	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
Наименование групп	Выпадающий список	Нет	Ключ атрибута, отвечающий за наименование группы. В случае отсутствия, в качестве наименования будет использоваться поле "Атрибут для группировки".	
Цвет группы	Выпадающий список	Нет	Ключ атрибута, отвечающий за цвет точек одной группы. В случае, если поле не задано, цвета групп будут установлены по умолчанию.	
Наименование точки	Выпадающий список	Нет	Ключ атрибута, отвечающий за наименование точки.	
Наименование осей	Выпадающий список	Нет	Ключ атрибута, отвечающий за наименование осей. Для наименования будет использован первый найденый атрибут в рядах.	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
Описание привязок	JSON	Нет	Используются для отображения корректных наименований точек и рядов с данными. Возможные параметры: • **group_field_key** - Ключ атрибута, отвечающего группировку значений. • Обязательное поле • **group_name_key** - Ключ атрибута, отвечающий за наименование группы. • **color_group_key** - Ключ атрибута, отвечающий за цвет точек одной группы • **point_name_key** - Ключ атрибута, отвечающий за наименование точки. **labels_key** - Ключ атрибута, отвечающий за наименование осей.	`{ "group_field_key": "group_id", "group_name_key": "group_name", "color_group_key": "color_group", "point_name_key": "name", "labels_key": "labels" }`

6.1.1.8.3 Группа «Общие настройки блока»

Аналогично приведенным в п. 6.1.1.1.3.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.1.1.8.4 Входы

Название входа	Тип	Обязательный	Описание
Данные	Ряд данных	Да	Массив данных с типом "Ряд данных"
Привязки	Строка	Нет	Привязки которые приходят на вход блока используются вместо параметра настроек "Привязки". Имеют наивысший приоритет среди всех привязок. Для динамической перестройки, к примеру, когда таблица является частью ОЛАПа, статичные привязки могут привести к некорректному отображению данных. Поэтому, в таких случаях, в таблицу нужно передавать и привязки. Формат привязок аналогичен описанию из настройки "Описание привязок".

6.1.1.8.5 Выходы

В блоке нет выходов.

6.1.1.8.6 События

В блоке нет событий.

6.1.1.8.7 Настройки визуализатора

Визуализатор не имеет специфических настроек. Общий список приведен в п. 4.2.1.3 «Общие настройки в режиме редактирования пресета».

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.1.2 Таблицы

6.1.2.1 Блок «Гиперссылки»

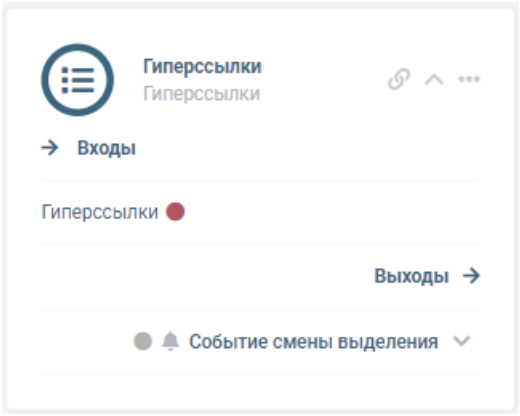


Рисунок 107 – Пример блока «Гиперссылки»

6.1.2.1.1 Назначение

Блок позволяет задавать список гиперссылок с использованием шаблонизатора Jinja2 в зависимости от входных данных.

6.1.2.1.2 Настройки блока

Название параметра	Тип параметра	Обязательный	Описание	Пример
Перетаскивать данные из списка гиперссылок	Флаг	Нет	Устарел	
Шаблон	Строка	Нет	Шаблон гиперссылки. Настройки шаблона указываются для одного элемента (не для массива)	
Событие по умолчанию	Выпадающий список	Да	Поведение при вызове события по умолчанию • Отображать первую строку • Отображать выбранный элемент	
Выбранные элементы (индексы)	Целое число	Нет	Индекс элемента для события по умолчанию. Параметр применяется, если настройка «**Событие по умолчанию**» принимает значение «**Отображать выбранный элемент**»	

6.1.2.1.3 Группа «Общие настройки блока»

Аналогично приведенным в п. 6.1.1.1.3.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.1.2.1.4 Входы

Название входа	Тип	Обязательный	Описание
Гиперссылки	Объект	Да	Массив данных с типом «Ряд данных» или «Объект»

В том случае когда не задан шаблон, в атрибутах объекта должны присутствовать следующие поля для отображения гиперссылок:

```
{
 "url": <Гиперссылка. Обязательное поле!>
 "name": <Наименование гиперссылки. Необязательное поле.>
 "img": <Иконка гиперссылки. Необязательное поле.>
}
```

6.1.2.1.5 Выходы

В блоке нет выходов.

6.1.2.1.6 События блока

Название выхода	Тип	Описание
Событие смены выделения	Объект	Используется для динамического обновления визуализаторов, при изменении отметки в блоке «Гиперссылки»

6.1.2.1.7 Описание работы блока

Блок принимает на вход массив с объектами или рядами данных. Каждый элемент массива обрабатывается шаблоном, указанным в соответствующем поле настроек блока с использованием шаблонизатора Jinja2. Шаблон может быть встроен в произвольную html верстку:

```
<div style="line-height:12px">
<p style="color:red"><strong>{{ input.attrs.name }}</strong></p>
<p>{{ input.attrs.address }}</p>
<p><em>{{ input.attrs.phone }}</em></p>
</div>
```

6.1.2.1.8 Описание работы визуализатора

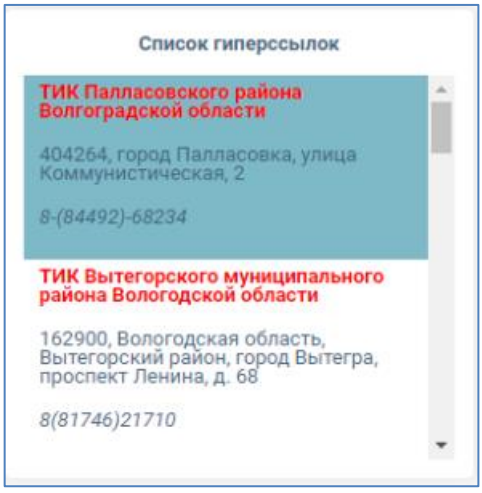


Рисунок 108 – Внешний вид визуализатора «Гиперссылки»

- Чтобы выбрать элемент в списке гиперссылок необходимо нажать на него ЛКМ.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

- Одновременно может быть выбран только один элемент.
- По умолчанию выбран элемент в соответствии с настройками блока на графе («Событие по умолчанию» и «Выбранные элементы (индексы)»).

6.1.2.1.9 Настройки визуализатора

Визуализатор не имеет специфических настроек. Общий список настроек приведен в п. 4.2.8 «Настройки визуализатора».

6.1.2.2 Блок «Итоги таблицы»

6.1.2.2.1 Назначение

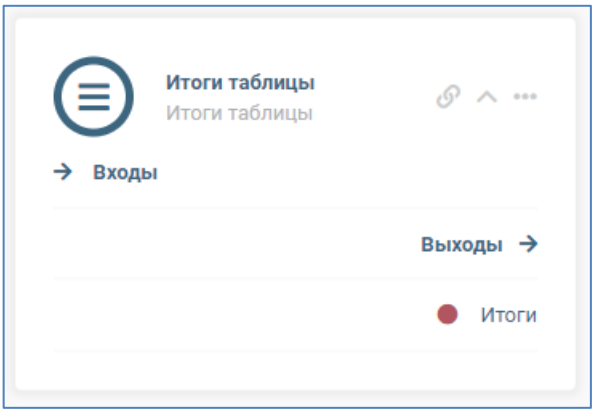


Рисунок 109 – Пример настройки блока «Итоги таблицы»

Блок позволяет рассчитывать различные типы итогов по строкам и столбцам в таблице.

6.1.2.2.2 Настройки блока

Название параметра	Тип параметра	Обязательный	Описание	Пример
Раздел настроек «Основные свойства»				
Итоги по строкам	Раздел настроек	Нет	Задает количество вариантов итогов по строкам	
Название итога	Строка	Нет	Задает название итога строк	
Использовать по умолчанию	Флаг	Да	Задает использование итога строк по умолчанию	
Тип итога	Выпадающий список	Нет	Позволяет выбрать тип итога по строкам	
Выводить пользовательское наименование	Флаг	Да	Задает использование итога пользовательского наименование итога по умолчанию	
Итоги по столбцам	Раздел настроек	Нет	Задает количество вариантов итогов по столбцам	
Название итога	Строка	Нет	Задает название итога столбцов	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
Использовать по умолчанию	Флаг	Да	Задает использование итога столбцов по умолчанию	
Тип итога	Выпадающий список	Нет	Позволяет выбрать тип итога по столбцам	
Выводить пользовательское наименование	Флаг	Да	Задает использование итога пользовательского наименование итога по умолчанию	

6.1.2.2.3 Группа «Общие настройки блока»

Аналогично приведенным в п. 6.1.1.1.3.

6.1.2.2.4 Входы

В блоке нет входов.

6.1.2.2.5 Выходы

Название выхода	Тип	Описание
Итоги	JSON	Используется для вывода различных типов итогов строк и столбцов.

6.1.2.2.6 События блока

В блоке нет событий.

6.1.2.2.7 Описание работы блока

Блок взаимодействует с блоком «Таблица» (п. 6.1.2.3).

Блок рассчитывает различные типы итогов по строкам и столбцам.

Так же доступна пользовательская настройка наименований итогов строк и столбцов.

Пример настройки блока «Итоги таблицы» приведены на рисунке (Рисунок 110).

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

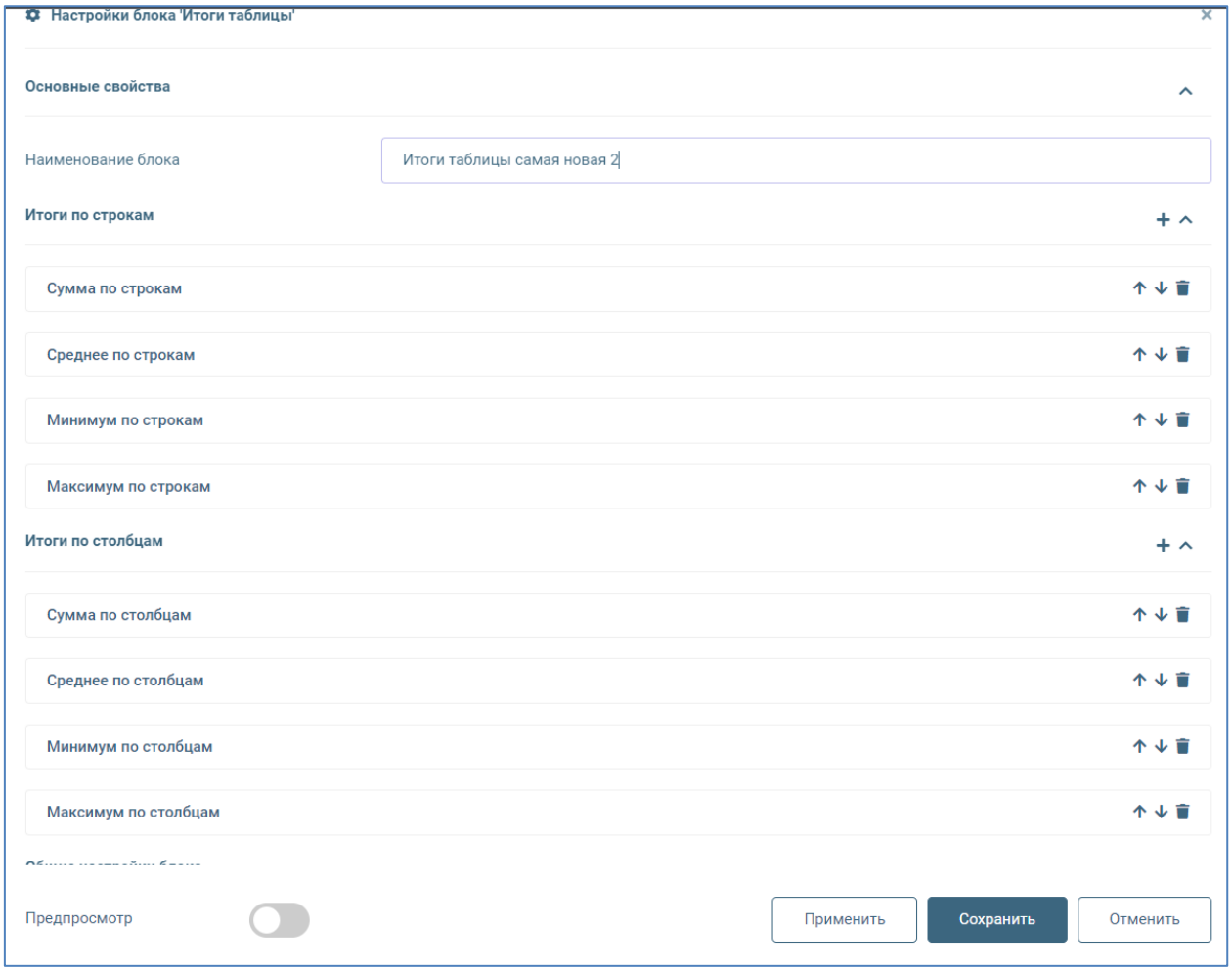


Рисунок 110 – Пример настройки блока «Итоги таблицы»

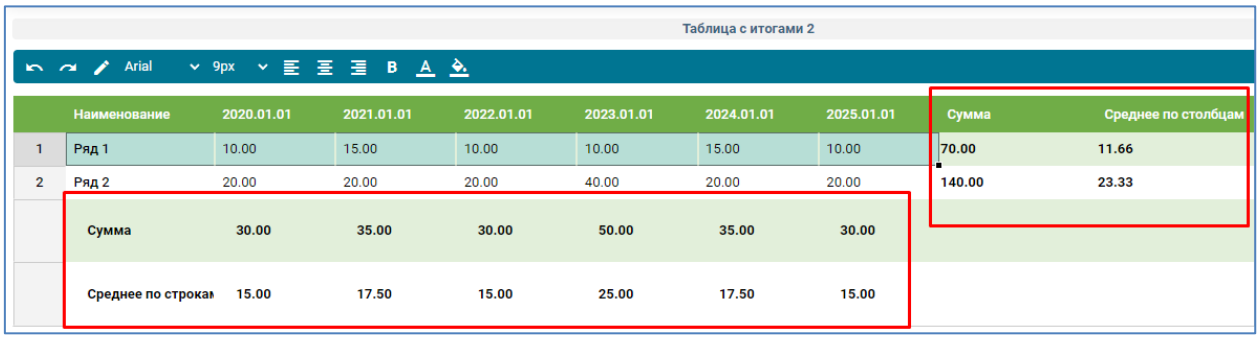
6.1.2.2.8 Описание работы визуализатора

Пример визуализатора «Итоги таблицы» с пользовательскими настройками приведен на рисунке (Рисунок 111).

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Рисунок 111 – Пример настройки блока «Итоги таблицы»

Пример отображения результатов работы блока «Итоги таблицы» на визуализаторе «Таблица» приведен на рисунке (Рисунок 112).



	Наименование	2020.01.01	2021.01.01	2022.01.01	2023.01.01	2024.01.01	2025.01.01	Сумма	Среднее по столбцам
1	Ряд 1	10.00	15.00	10.00	10.00	15.00	10.00	70.00	11.66
2	Ряд 2	20.00	20.00	20.00	40.00	20.00	20.00	140.00	23.33
	Сумма	30.00	35.00	30.00	50.00	35.00	30.00		
	Среднее по строка	15.00	17.50	15.00	25.00	17.50	15.00		

Рисунок 112 – Пример отображения результатов блока «Итоги таблицы»

6.1.2.2.9 Настройки визуализатора

Визуализатор не имеет специфических настроек. Общий список настроек приведен в п. 4.2.8 «Настройки визуализатора».

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.1.2.3 Блок «Таблица»

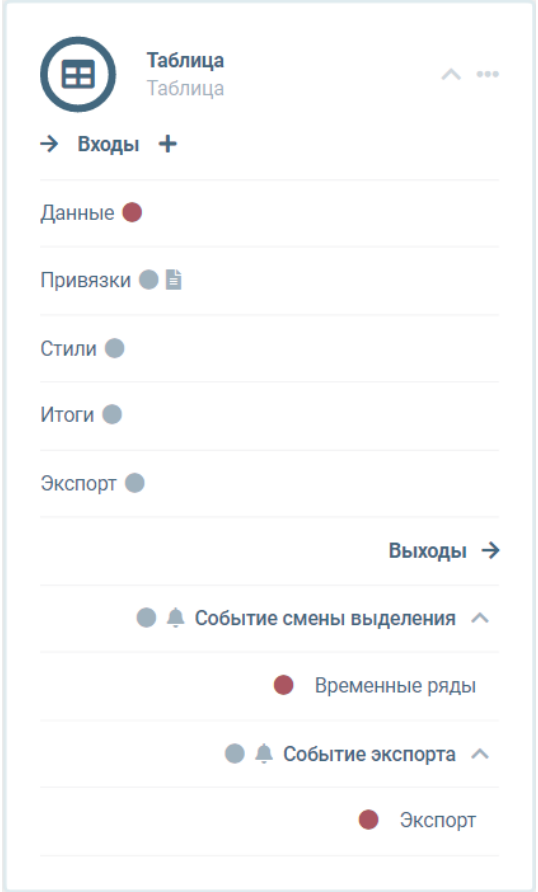


Рисунок 113 – Пример блока «Таблица»

6.1.2.3.1 Назначение

Блок позволяет формировать визуализатор таблицы из массива рядов данных или объектов.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.1.2.3.2 Настройки блока

Название параметра	Тип параметра	Обязательный	Описание	Пример
Привязки	JSON	Да	Параметры, на основе которых происходит связь заголовка и атрибута таблицы с рядами данных. Доступные параметры: • **label** – ключ поля, значения которого будут использоваться в качестве заголовков таблицы. Поле не должно являться первичным ключом. Применяется для простых (не временных) рядов. В случае с временными рядами, в качестве заголовков будут использоваться даты. • **labelMapping** – словарь связи ключей заголовков с ключами атрибутов, отвечающими за наименование. Используется для форматирования наименований в сложной шапке (когда в шапке больше одного ключа). Может использоваться как замена параметра **label**. В случае наличия обоих параметров (**label** и **labelMapping**) приоритет форматирования будет у поля **labelMapping**. **Формат привязки** {"<идентификатор поля с заголовком>": "<идентификатор поля с наименованием заголовка>"} • **label_name** – массив наименований справочников, которые в текущий момент находятся в столбцах (т.е. при транспонировании они будут в строках), не обязательное. • **attrs** – массив элементов, которые необходимо вывести в качестве атрибутов таблицы: • **key** – ключ элемента из атрибутов (**attrs**), значения которого пойдут в атрибуты для рядов; Для применения настроек ко всем атрибутам сразу используется технический ключ **_all.** Обратите внимание, что атрибуты с ключом **_all** не будут отображаться, т.к. данный ключ зарезервирован системой.	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
			В текущей версии приложения поддерживается только настройка условного форматирования (**format_conditions**): • **name_key** – ключ, отвечающий за значения этого атрибута (если он есть), не обязательное (не используется в блока с версии 0.38); • **label** – наименование столбца; • **type** – формат столбца с атрибутами (string|list|float|checkbox|control|int), в случае отсутствия используется string; • **editable** – возможность редактирования (true|false), не обязательное; • **visible** – видимость атрибута (true|false), не обязательное; • **fixed** – фиксирование столбца с атрибутов, столбцы фиксируются по порядку (true|false), не обязательное; Применяется только для таблицы, состоящей из объектов, а не рядов. • **style-default** – стиль ячейки по умолчанию (словарь), не обязательное; • **width** – ширина столбца с атрибутами (целое число), не обязательное; • **sort_field** – ключ атрибута, который нужно использовать в качестве сортировки (при сортировки по атрибуту будут использоваться значения, находящиеся в указанном атрибуте); • **format_conditions** – настройки условного форматирования для атрибутов. Настройки будут применяться для всего столбца с атрибутами. Настройки форматирования из привязок имеют средний приоритет – они будут изменять стили, применяющиеся для всего блока (к примеру стили из Блока «Стили таблицы» (п. 6.8.1)). Но условное форматирование единичной ячейки будет изменять стили условного форматирования из привязок.	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
			Настройка **width** имеет более высокий приоритет и будет перезаписывать аналогичное поле из настройки **format_conditions.** **Настройка иерархии по дополнительному справочнику** Для отображения иерархии необходимо, чтобы настройка "**Отображать иерархию**" была включена. Для отображения иерархии в таблице, можно использовать дополнительный иерархический справочник. Необходимо добавить два параметра в атрибут, по которому должна формироваться иерархия: **hierarchy_dict** - идентификатор входа со справочником, из которого необходимо брать иерархию для отображения. **hierarchy_key** - ключ в справочнике (из параметра **hierarchy_dict**), отвечающий за иерархию. Этот ключ будет сопоставляться с атрибутом, в котором указаны параметры иерархии (поля **hierarchy_dict** и **hierarchy_key**) • **control** – настройки управления атрибутами. Доступен функционал выпадающего списка, аналогично блоку «Панель размещения и фильтрации»: В текущей версии блока (14) поддерживается только выпадающий список «комбобокс» с единичной отметкой. • **input_key** – Идентификатор входа справочника. • **key** – Идентификатор поля в справочнике, который использовать как ключ. Если не задано берется поле по умолчанию – ***id***. • **name** – Идентификатор поля в справочнике, который использовать как наименование. Если не задано берется поле по умолчанию – ***name***. • **default** – Ключи элементов, которые необходимо отмечать по умолчанию. Если поле не задано – отмечается первый элемент при	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример

			отмеченной единичной отметке или все элементы при включенной мультиотметке.
			• **multiselect** – Целое число, 1 или 0. Включение множественной отметки у справочника (возможность отобразить от нуля до всех элементов). Если не задано – множественная отметка выключена.
			• **singleselect** – Целое число, 1 или 0. Включение единичной отметки у справочника (наличие минимум одного элемента в отметке). Если не задано – множественная отметка выключена.
			Отметки multiselect и singleselect могут быть включены вместе.
			• **combobox** – Целое число, 1 или 0. Отображение справочника в виде выпадающего списка. По умолчанию выпадающий список выключен.
			• **hierarchy** – Целое число, 1 или 0. Отображение данных в виде иерархии. Без включенного флага будут отображаться только данные на первом уровне. При включении иерархии автоматически включается выпадающий список. Отдельно флаг прописывать не нужно. По умолчанию – выключено.
			• **filter** – Массив словарей. Отображение элементов данного справочника зависит от другого. Допустимые поля:
			  • **from** – Идентификатор справочника, от которого зависит данный справочник.
			  • **foreignAttr** – Идентификатор поля в справочнике, от которого зависит данный справочник, по которому будут фильтроваться элементы.
			  • **localAttr** – Идентификатор поля в данном справочнике, по которому будет идти поиск соответствия с «**foreignAttr**».
			  • **addFields** – Массив строк. Состоит из идентификаторов полей в справочнике, по которым идет фильтрация (все значения «**localAttr**»).	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример

			• **date_range** – Целое число, 1 или 0. Использование справочника для выбора диапазона дат.
			• **date_range_type** – Тип диапазона дат. Возможные значения:
			  • 1 – год,
			  • 2 – полугодие,
			  • 3 – квартал,
			  • 4 – месяц,
			  • 5 – день
			• **date_range_start** – Дата – начало диапазона в справочнике.
			• **date_range_finish** – Дата – конец диапазона в справочнике
			• **data** – массив элементов для настройки области с данными:
			  • **key** – ключ (или массив ключей) (fkey) ряда данных.
			Для применения настроек ко всем элементам сразу используется технический ключ **_all.**
			  • **format_conditions** – настройки условного форматирования для данных. Подробнее см. условное форматирование таблицы (п. 6.8.2). Настройки будут применяться для всего столбца с данными целиком.
			Настройки форматирования из привязок имеют средний приоритет – они будут изменять стили, применяющиеся для всего блока (к примеру стили из Блока «Стили таблицы» (п. 6.8.1)). Но условное форматирование единичной ячейки будет изменять стили условного форматирования из привязок	

Количество знаков после запятой	Целое число	Да	Настройка количества знаков после запятой у данных, которые отобразятся в таблице	
Разделитель групп разрядов	Строка	Нет	Настройка разделителя групп разрядов. Может использоваться «,» или «» Устаревшие настройки С версии 24 и выше используется параметр "Разделитель разрядов"	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
Использовать системные разделители	Флаг	Нет	Включить использование системных разделителей разрядов и целой и дробной части чисел. По умолчанию - включено.	
Разделитель целой и дробной части	Строка	Нет	Отображается только если настройка **"Использовать системные разделители"** отключена. Набор символов, которые будут использоваться для разделения целой и дробной части числа. По умолчанию используется символ "**.**" (точка) Набор символов не должен совпадать с настройкой "Разделитель разрядов"	
Разделитель разрядов	Строка	Нет	Отображается только если настройка **"Использовать системные разделители"** отключена. Набор символов, которые будут использоваться для разделения разрядов числа. По умолчанию используется пробел. Набор символов не должен совпадать с настройкой "Разделитель целой и дробной части" До версии 24 параметр был доступен всегда и назывался "Разделитель групп разрядов"	
Нумерация строк	Флаг	Нет	Включить нумерацию строк Устарел и отключен в новых блоках (версия 6 и выше)	
Поиск и фильтрация	Флаг	Нет	Включить поиск по наименованию	
Возможность множественной отметки	Флаг	Нет	Включить возможность множественной отметки	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
Выделить ряд целиком	Флаг	Нет	Включить выделение ряда целиком (а не только выделенный диапазон)	
Подсветка ячеек для события смены выделения	Флаг	Нет	Включить подсветку ячеек при их выделении. Параметр доступен, если настройка «**Выделить ряд целиком»** отключена.	
Скрывать данные в таблице	Флаг	Нет	Скрыть данные таблицы (оставить только заголовки)	
Скрыть номера строк	Флаг	Нет	Скрыть номера строк таблицы	
Переносить текст	Флаг	Нет	Включить автоматический перенос текста	
Перетаскивать данные из таблицы	Флаг	Нет	Устарел и отключен в новых блоках (версия 7 и выше)	
Выравнивание по правому краю	Флаг	Нет	При открытии таблицы горизонтальная полоса прокрутки будет в крайнем правом положении	
Динамическая подгрузка данных	Флаг	Нет	Устарел	
Отображать иерархию	Выпадающий список	Нет	Отображение иерархических справочников. Возможные варианты значения: • Нет – иерархия отсутствует. • Отступы – иерархия представлена в виде отступов в названиях строк. • Экспандеры – иерархия представлена в виде сворачивающихся строк. Для передачи уровня иерархии используется технический параметр **_hierarchy_level** (целое число, от «1». «0» – иерархия выключена). Параметр приходит автоматически после блоков OLAP с включенной иерархией. Можно задавать параметр вручную (должен являться атрибутом), но следует учитывать, что параметр может быть перезаписан (к примеру, в блоке OLAP), т.к. является системным	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
Отступ иерархии	Целое число	Нет	Количество пробелов для отступов иерархии. Параметр доступен, если настройка «**Отображать иерархию»** включена	
Событие по умолчанию	Выпадающий список	Да	Поведение таблицы при вызове события по умолчанию	
Выбранные элементы (индексы)	JSON	Нет	Список индексов элементов для события по умолчанию. Параметр применяется, если настройка «**Событие по умолчанию»** принимает значение «**Отображать выбранные элементы».** Если значение индекса больше, чем количество элементов в выходе, индекс пропускается	[0, 1, 5, 10, 42]
Отображение страниц	Флаг	Нет	Включить постраничное отображение таблицы	
Объединение ячеек	Флаг	Нет	Включить объединение ячеек с одинаковыми данными (для заголовков)	

Отображать меню	Выпадающий список	Нет	Поведение отображения меню таблицы (панель инструментов)
			• Не отображать:
			  • нечего не отображать.
			• Отображать все:
			  • выводить Undo/Redo;
			  • выводить кнопку редактирования/отмены редактирования;
			  • выводить меню стилей.
			• Редактирование:
			  • выводить Undo/Redo;
			  • выводить кнопку редактирования/отмены редактирования.
			• Стили:
			  • выводить Undo/Redo;
			  • выводить только меню стилей	

Транспонирование данных	Флаг	Нет	Отобразить таблицу в транспонированной форме	
Сортировка столбцов данных	Флаг	Нет	Включить возможность сортировки данных	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
Использовать цвета сценариев	Флаг	Нет	Устарел	
Передавать информацию в видимых рядах	Флаг	Нет	При включении флага появляется новый событийный выход Видимые ряды, в который будут передаваться все ряды, видимые пользователем в блоке	
Сохранять выделенные элементы при изменении таблицы	Флаг	Нет	При включении блок будет пытаться сохранить отметку при любом изменении таблицы. Если отметку сохранить не получилось - будет получена отметка, указанная в параметре "Событие по умолчанию"	
Отображать строку с номерами столбцов	Флаг	Нет	При включении в таблице у колонок отобразятся их номера под шапкой. Флаг должен быть включен для возможности сортировки и изменения ширины столбцов, если в последней строчке шапки есть объединенные ячейки	
Группа «Настройки связи»				
Управляемый блок	Флаг	Нет	Устарел	
Наименование	Строка	Нет	Устарел	
Очистка дополнительных рядов при расчете задачи	Флаг	Нет	Устарел и отключен в новых блоках (версия 7 и выше)	
Группа «Экспорт»				
При экспорте удалять HTML	Флаг	Нет	Если установлен в «Да», то удаляется весь HTML и SVG контент ячеек. Текстовые данные при этом сохраняются.	
Выводить заголовок при экспорте	Флаг	Нет	При включенном параметре в итоговой таблице будет добавлена строка-заголовок. **Заголовок по умолчанию** – название блока таблицы. Заголовок может быть настроен с помощью поля «**Текст заголовка»**	
Текст заголовка	Строка	Нет	Тестовое поле для описания заголовка файла. Поле отображается лишь при включенном параметре «**Выводить заголовок при**	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
			экспорте». Поле поддерживает форматирование Jinja. В качестве значения по умолчанию берётся название блока. Пример: текста заголовка ниже будет обработан как «<название блока>. Отчет за месяц». {{input}}. Отчет за месяц	
Имя файла	Строка	Нет	Имя экспортируемого файла. Имя по умолчанию – название блока	
Удалять скрытые столбцы	Флаг	Нет	Флаг, определяющий будут удалены скрытые столбцы или нет. Настройка скрытия столбцов возможна в описании привязок	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.1.2.3.3 Группа «Общие настройки блока»

Аналогично приведенным в п. 6.1.1.1.3.

6.1.2.3.4 Входы

Название входа	Тип	Обязательный	Описание
Данные	Ряд данных	Да	Массив данных с типом «Ряд данных» или «Объект»
Привязки	Строка	Нет	Привязки которые приходят на вход блока используются вместо параметра настроек «Привязки». Для динамической перестройки, к примеру, когда таблица является частью OLAP, статичные привязки могут привести к некорректному отображению данных. Поэтому, в таких случаях, в таблицу нужно передавать и привязки
Стили	JSON	Нет	Настройки стилей для таблицы. Формируются с помощью блока «Стили таблицы» (п. 6.8.1)
Итоги	JSON	Нет	Настройки итогов для таблицы. Формируются с помощью блока «Итоги таблицы» (п. 6.1.2.2)
Экспорт	JSON	Нет	Настройки для экспорта таблицы. В случае появления данных у этого входа будет произведен экспорт таблицы и в событии экспорта будет указан путь до экспортируемой таблицы. Формат параметров экспорта `{"format": <format>}` где format – формат экспорта: • xlsx – для экспорта в формат Excel • ods – для экспорта в формат ODS

6.1.2.3.4.1 Динамические входы

Для подключения выпадающих списков в атрибутах (см. настройку «Привязки») необходимо добавить входы для справочников.

Каждый новый справочник добавляется в отдельный вход. Параметры «Обязательный» и «Массив» должны быть включены. Параметр «Идентификатор» необходимо указать в поле **input_key** в параметрах привязки.

Для создания входа необходимо нажать на знак в разделе входов блока. Для каждого из входов задается следующий перечень полей

Название входа	Тип	Обязательный	Описание
Идентификатор	Строка	Да	Идентификатор входа – используется внутри блока, для получения доступа к данным входа
Имя	Строка	Да	Имя входа для отображения в списке входов блока

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название входа	Тип	Обязательный	Описание
Описание	Строка	Нет	Дополнительная информация с описанием входа
Тип	Выпадающий список	Да	Тип данных, которые будут приходить на вход. Нужен для построения корректных связей с другими блоками. По умолчанию установлен тип "Объект".
Обязательный	Флаг	Да	Обязательность входа. Если вход обязательный и на него не пришло данных – блок не будет рассчитан с ошибкой
Массив	Флаг	Да	Является ли вход массивом данных

6.1.2.3.5 Выходы

Название выхода	Тип	Описание
Добавленные данные	Ряд данных	Устарел и отключен в новых блоках (версия 7 и выше)

6.1.2.3.6 События блока

Название выхода	Тип	Описание
Событие смены выделения	Ряд данных	Используется для динамического обновления визуализаторов, при изменении отметки в таблице
Событие экспорта	JSON	Используется в момент экспорта таблицы (см. вход «**Экспорт**»). Для скачивания файла необходимо использовать блок «Скачать файл» (см. п. 6.1.4.14)
Видимые ряды	Ряд данных	Используется для передачи всех видимых пользователю рядов. Появляется при включенном параметре "Передавать информацию в видимых рядах"

6.1.2.3.7 Описание работы блока

Блок принимает на вход массив с объектами или рядами данных, после чего проводит преобразования:

- Блок ищет пересечение всех значений поля **fkey** (параметров привязки для рядов), формируя ключи заголовка, после чего подставляет значения ряда, соответствующие значениям ключей из заголовков.

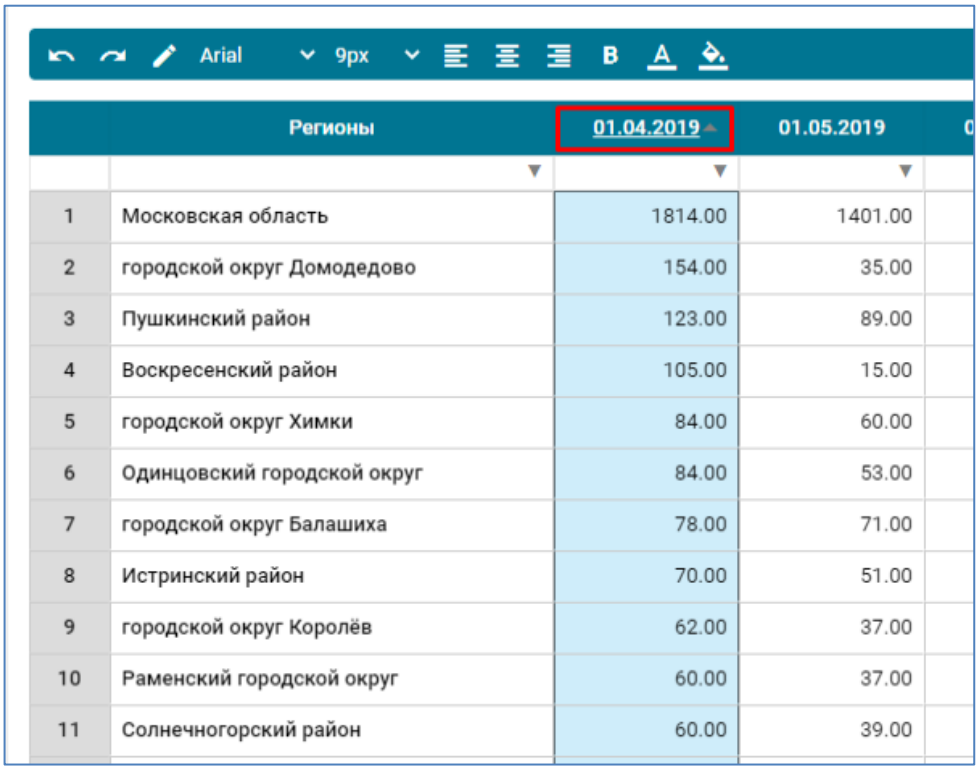
  Только если на вход таблицы пришли ряды данных, для объектов формируются только атрибуты.

- Один ряд данных на входе соответствует одной строке в таблице.
- В конце, если необходимо – применяет привязки **label** или **labelMapping** для формирования наименований заголовков.
- Если заданы атрибуты таблицы (**attrs**), то формируются дополнительные столбцы в начале таблицы.
- Общие настройки стилей (к примеру, указанные в привязках с ключом «**_all**») сохраняются в общие стили таблицы.
- Стили, применяемые для конкретных рядов, сохраняются в выбранном ряду.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.1.2.3.8 Описание работы визуализатора

- Чтобы отсортировать значения в столбце необходимо сделать двойной клик ЛКМ по заголовку столбца.



	Регионы	01.04.2019	01.05.2019
1	Московская область	1814.00	1401.00
2	городской округ Домодедово	154.00	35.00
3	Пушкинский район	123.00	89.00
4	Воскресенский район	105.00	15.00
5	городской округ Химки	84.00	60.00
6	Одинцовский городской округ	84.00	53.00
7	городской округ Балашиха	78.00	71.00
8	Истринский район	70.00	51.00
9	городской округ Королёв	62.00	37.00
10	Раменский городской округ	60.00	37.00
11	Солнечногорский район	60.00	39.00

Рисунок 114 – Пример сортировки значений в столбце

- Чтобы произвести фильтрацию в столбце необходимо дважды щелкнуть ЛКМ по полю с фильтром, затем выбрать значения, по которым будет происходить фильтрация, и щелкнуть ЛКМ в любое место, кроме окна фильтрации.

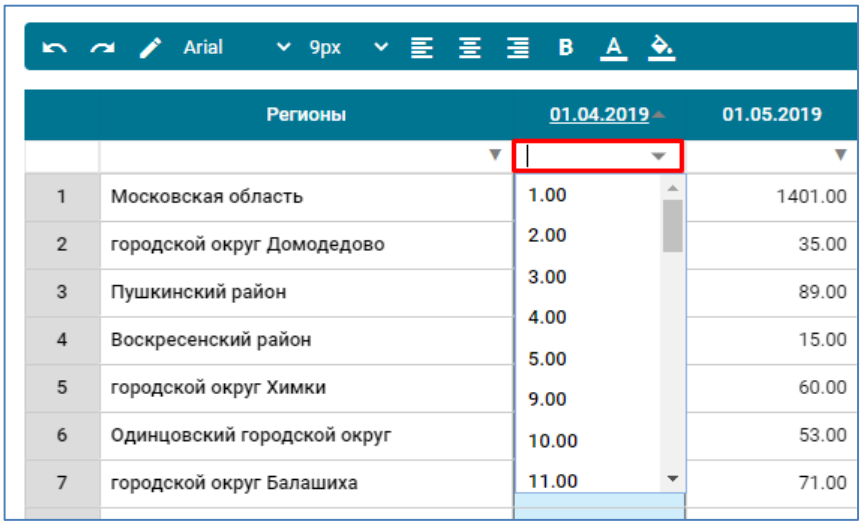


Рисунок 115 – Пример фильтрации в столбце

- В фильтре можно открыть контекстное меню, кликнув ПКМ, чтобы выбрать все элементы или снять отметку со всех элементов

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

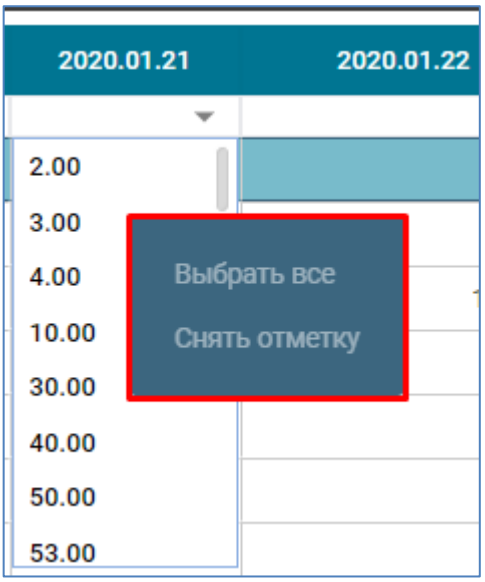


Рисунок 116 – Пример контекстного меню фильтра

- Чтобы произвести поиск по таблице необходимо ввести искомое слово или фразу в поле «Поиск».

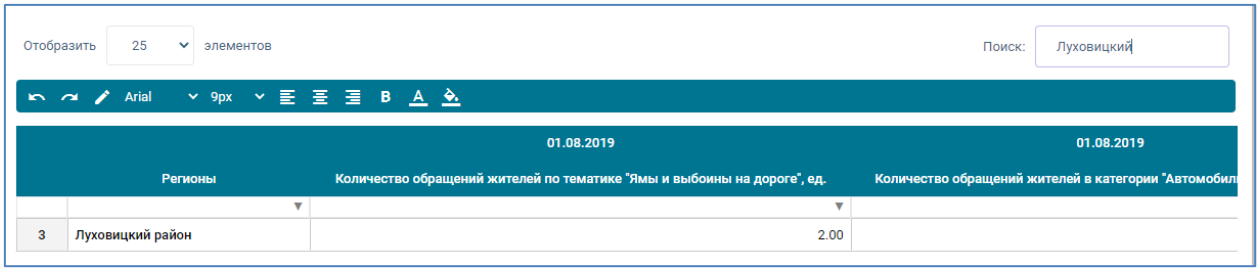


Рисунок 117 – Пример поиска в таблице

- Чтобы отредактировать данные необходимо нажать кнопку «Редактировать данные» на панели инструментов или в локальном меню.

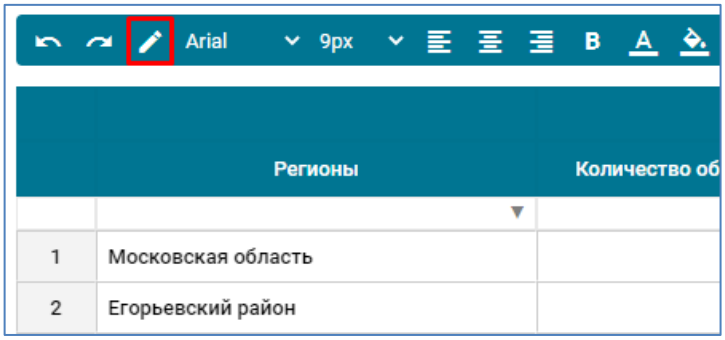


Рисунок 118 – Пример кнопки «Редактировать данные» на панели инструментов.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

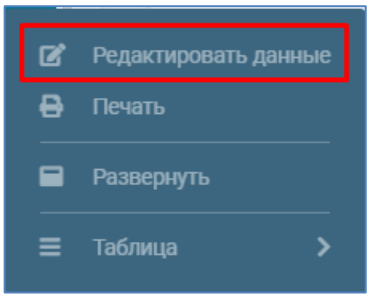


Рисунок 119 – Пример кнопки «Редактировать данные» в локальном меню

- Чтобы добавить новую строку с данными необходимо, находясь в режиме редактирования таблицы, нажать на кнопку "Вставить строку" в панели инструментов.

Важно! Для добавления новых строк, необходимо, чтобы в атрибутах таблицы были указаны все первичные ключи ряда. В противном случае сохранение вызовет ошибку.

Рисунок 120 – Пример кнопки «Вставить строку» на панели инструментов

- Для возможности добавления атрибутов таблицы через выпадающий список, необходимо настроить атрибут в блоке "SQL-запрос для рядов".

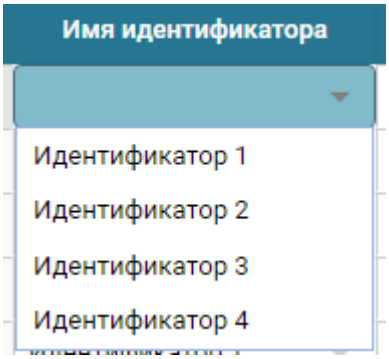


Рисунок 121 – Пример выпадающего меню

- Чтобы удалить строку с данными, необходимо, находясь в режиме редактирования таблицы, выделить нужную строку и нажать на кнопку "Удалить строку" в панели инструментов.

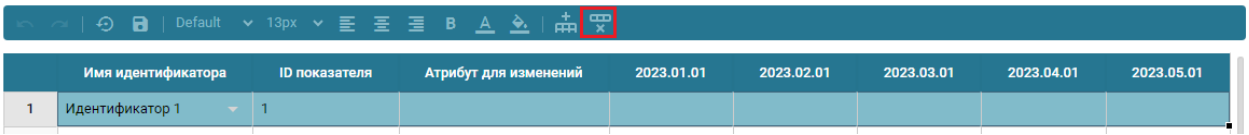


Рисунок 122 – Пример кнопки «Удалить строку» в панели инструментов

- Чтобы сохранить измененные данные необходимо, находясь в режиме редактирования таблицы, нажать кнопку «Сохранить» на панели инструментов или в локальном меню.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

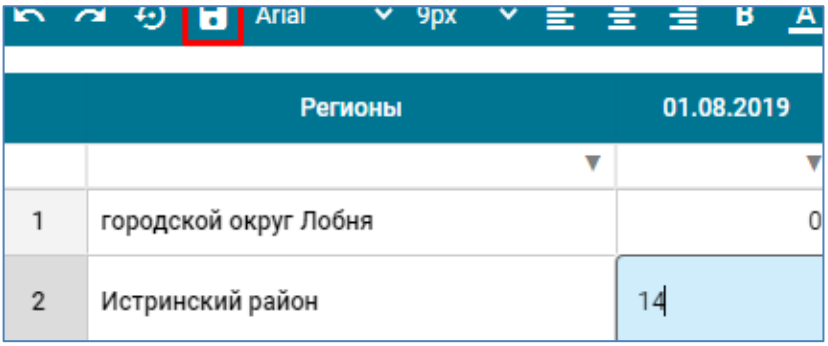


Рисунок 123 – Пример кнопки «Сохранить» на панели инструментов

- Чтобы выйти из режима редактирования таблицы без сохранения необходимо нажать кнопку «Отменить изменения» на панели инструментов или в локальном меню.

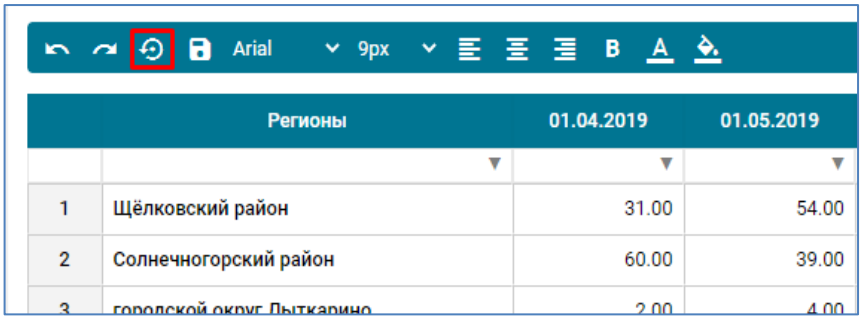


Рисунок 124 – Пример кнопки «Отменить изменения» на панели инструментов

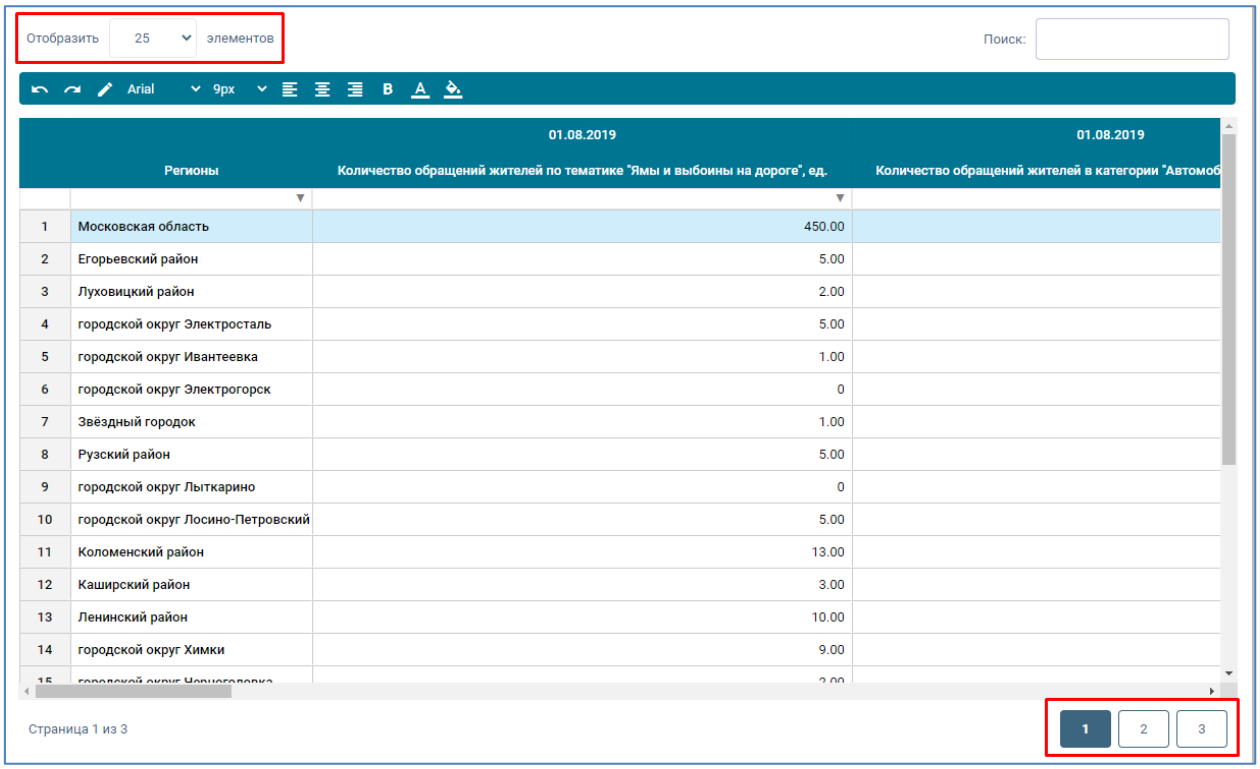
Рисунок 125 – Пример кнопки «Отменить изменения» в локальном меню

- В зависимости от параметра блока «Отображать меню» в таблице может отображаться или быть скрыта панель инструментов.

Рисунок 126 – Пример панели инструментов

- С включенным в блоке параметром «Отображение страниц» содержимое в таблице может отображаться постранично.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)



	Регионы	01.08.2019 Количество обращений жителей по тематике "Ямы и выбоины на дороге", ед.	01.08.2019 Количество обращений жителей в категории "Автомоб
1	Московская область	450.00	
2	Егорьевский район	5.00	
3	Луховицкий район	2.00	
4	городской округ Электросталь	5.00	
5	городской округ Ивантеевка	1.00	
6	городской округ Электрогорск	0	
7	Звёздный городок	1.00	
8	Рузский район	5.00	
9	городской округ Лыткарино	0	
10	городской округ Лосино-Петровский	5.00	
11	Коломенский район	13.00	
12	Каширский район	3.00	
13	Ленинский район	10.00	
14	городской округ Химки	9.00	

Рисунок 127 – Пример постраничного отображения

С постраничным отображением существенно улучшается скорость работы таблицы при больших объемах данных.

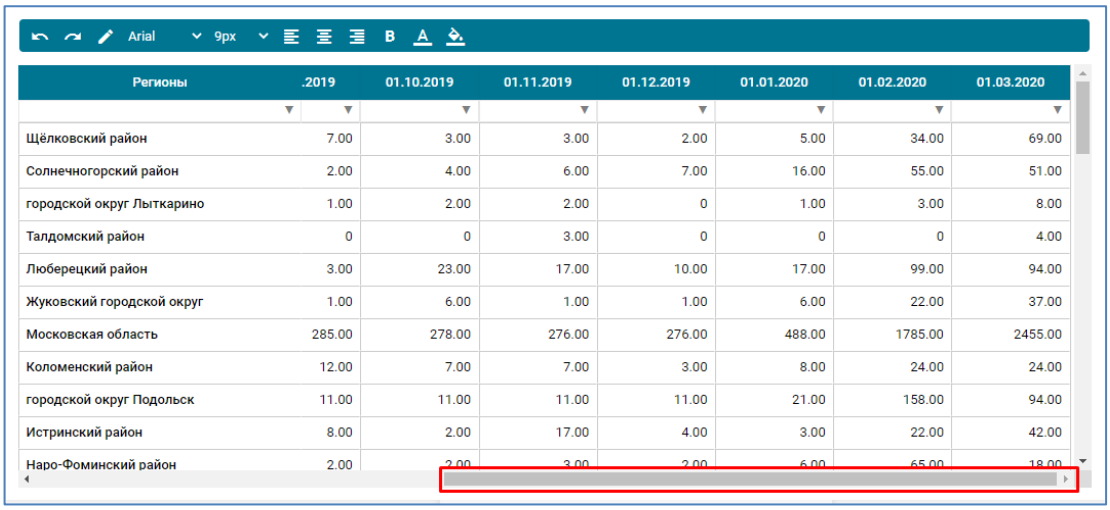
К таблице могут применяться стили при помощи дополнительного блока «Стили таблицы» (см. п. 6.8.1).

	Показатели	2005 data_Perm_region_csv Пермский край	2006 data_Perm_region_csv Пермский край	2007 data_Perm_region_csv Пермский край	2008 data_Perm_region_csv Пермский край
1	Валовой региональный продукт на душу населения, рублей Y	119654.00	141864.80	178096.80	227719.00
2	Оборот общественного питания, млн. Руб X8	6903.00	7292.00	10945.00	18401.00
3	Оборот розничной торговли ,млн. руб. X7	130236.00	178031.00	220407.00	278260.00
4	Объем инвестиций (в основной капитал) по территории за счет всех исто	56800.00	75519.00	122480.00	152363.00
5	Объем отгруженной продукции (работ, услуг) X3	180884.46	188481.61	198094.17	201461.78
6	Объем платных услуг, млн руб X4	39333.00	46168.00	56287.00	68629.00
7	Объем продукции сельского хозяйства, млн. руб. X5	18127.00	19010.00	20238.00	26972.00
8	Объем работ, выполненных по виду деятельности "Строительство" X2	23976.30	26671.90	38895.20	60280.40
9	Среднемесячная заработная плата (в среднем за год), руб X1	7748.90	9516.20	11856.00	14774.10

Рисунок 128 – Пример блока «Стили таблицы»

- При открытии таблицы горизонтальная полоса прокрутки может быть в крайнем правом положении, если в блоке включен параметр «Выравнивание по правому краю».

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)



Регионы	.2019	01.10.2019	01.11.2019	01.12.2019	01.01.2020	01.02.2020	01.03.2020
Щёлковский район	7.00	3.00	3.00	2.00	5.00	34.00	69.00
Солнечногорский район	2.00	4.00	6.00	7.00	16.00	55.00	51.00
городской округ Лыткарино	1.00	2.00	2.00	0	1.00	3.00	8.00
Талдомский район	0	0	3.00	0	0	0	4.00
Люберецкий район	3.00	23.00	17.00	10.00	17.00	99.00	94.00
Жуковский городской округ	1.00	6.00	1.00	1.00	6.00	22.00	37.00
Московская область	285.00	278.00	276.00	276.00	488.00	1785.00	2455.00
Коломенский район	12.00	7.00	7.00	3.00	8.00	24.00	24.00
городской округ Подольск	11.00	11.00	11.00	11.00	21.00	158.00	94.00
Истринский район	8.00	2.00	17.00	4.00	3.00	22.00	42.00
Наро-Фоминский район	2.00	2.00	3.00	2.00	6.00	65.00	18.00

Рисунок 129 – Пример горизонтальной полосы прокрутки в крайнем правом положении

В таблице может производится расчет итогов при помощи дополнительного блока «Итоги таблицы».

	Наименование	2020.01.01	2021.01.01	2022.01.01	2023.01.01	2024.01.01	2025.01.01	Сумма	Среднее по столбцам	Минимум по столбцам	Максимум по столбцам
1	Ряд 1	10.00	15.00	10.00	10.00	15.00	10.00	70.00	11.66	10.00	15.00
2	Ряд 2	20.00	20.00	20.00	40.00	20.00	20.00	140.00	23.33	20.00	40.00
	Сумма	30.00	35.00	30.00	50.00	35.00	30.00				
	Среднее по строкам	15.00	17.50	15.00	25.00	17.50	15.00				
	Минимум по строкам	10.00	15.00	10.00	10.00	15.00	10.00				
	Максимум по строкам	20.00	20.00	20.00	40.00	20.00	20.00				

Рисунок 130 – Пример отображения блока «Итоги таблицы»

- По умолчанию таблица является кросс-таблицей. Также таблица может быть плоской, если на вход ей приходят данные с типом «Объект» или в настройках блока включен параметр «Скрывать данные в таблице».
- Если на вход таблицы приходят данные с типом «Объект» существует возможность зафиксировать произвольное количество столбцов при горизонтальной прокрутке (значение «fixed» в параметрах привязок блока в графе).
- С включенным параметром «Объединение ячеек» в таблице будут объединяться ячейки с одинаковыми значениями в шапке и боковике.

	Сценарные показатели	Вариант	Отчет 2020	Отчет 2021
1	Валовой внутренний продукт, млрд. рублей	Базовый	106967.45	119426.34
2		Консервативный	106967.45	119234.77
3	Цены на нефть Urals (мировые), долл. / барр.	Базовый	55.00	50.00
4		Консервативный	40.00	40.00
5	Курс доллара среднегодовой, рублей за доллар США	Базовый	71.94	73.27
6		Консервативный	71.94	73.70

Рисунок 131 – Пример объединения ячеек

- К данным таблицы может применяться условное форматирование при помощи блоков, таких как: Блок «Форматирование по условию» (п. 6.8.2), «Форматирование

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

по условию (расширенное)» (п. **Error! Reference source not found.**) и «Python» (п. 6.9.1). Возможны настройки локали, цветов, выравнивания и т.д.

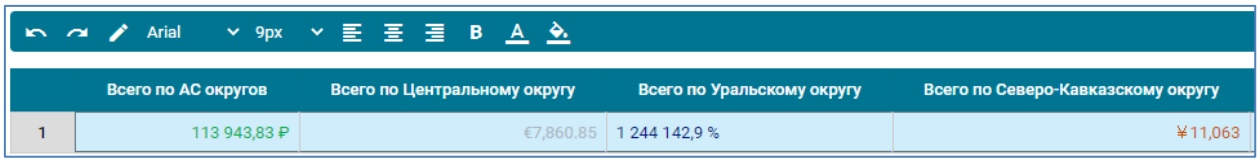


Рисунок 132 – Пример условного форматирования

6.1.2.3.9 Настройки визуализатора

Визуализатор не имеет специфических настроек. Общий список настроек приведен в п. 4.2.8 «Настройки визуализатора».

Помимо стандартных, визуализатор позволяет задать следующие настройки.

Название параметра	Тип параметра	Обязательный	Описание	Пример
Отображать шапку	Флаг	Да	Указывает отображать или нет в таблице шапку	
Фиксированная ширина колонок	Целое число	Да	Задать фиксированную ширину колонок в таблице	
Фиксировать ширину колонок	Флаг	Да	Указывает можно или нет изменять ширину колонок в таблице	
Сохранение настроек	Флаг	Да	Сохранять или нет внесенные пользователем изменения (шапка, ширина колонок, высота строк)	

6.1.2.3.10 Дополнительные настройки

Дополнительными настройками визуализатора является возможность сохранения, при сохранении пресета:

1. Шрифт текста в ячейках.
2. Размер текста в ячейках.
3. Выравнивание текста в ячейках.
4. Начертание текста в ячейках.
5. Цвет текста в ячейках.
6. Заливка фона ячеек.
7. Ширина колонок, установленная в режиме редактирования.
8. Автофильтр.
9. Разреженность в отображении рядов таблицы.

6.1.2.3.11 Экспорт таблиц

Настройки. Для экспорта возможно задание настроек. Подробное описание настроек находится в разделе Группа «Экспорт».

Форматирование. Поддерживается:

- Объединение ячеек;

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

- Удаление скрытых колонок при экспорте;
- Группировка строк по иерархии;
- Ширина колонок;
- Разделитель целой и дробной части;
- Разделитель разрядов.

Также при экспорте учитываются настройки стилей ячеек:

- Настройки шрифтов;
- Выравнивание текста по горизонтали;
- Цвет ячеек;
- Настройки границ ячеек.

Если пользователь при взаимодействии с визуализатором изменяет стили ячеек или меняет ширину колонок, сортировку колонок и фильтрацию колонок, то эти изменения будут учтены при экспорте.

Типы данных. Помимо стилей ячеек, возможна поддержка типов данных, определённых в колонках таблицы. Для этого в настройках таблицы для колонок необходимо явно задать тип данных, которые будут в ней храниться. Если в таблице будут содержаться ячейки, не соответствующие заданному типу колонки, приложение попытается привести их к заданному типу и в случае неудачи сохранит данные ячейки в исходном виде.

HTML. Поддерживается перекодирование спецсимволов HTML, если спецсимволы встречаются в исходных данных. Возможно удаление HTML-контента с сохранением исходного текста.

Форматы. Доступны три формата выходных файлов – Microsoft Excel (XLSX), OpenDocument Spreadsheet (ODS) и CSV. Для форматов XLSX и ODS одинаково поддерживаются все настройки экспорта. Форматы XLSX и ODS поддерживают настройки форматирования ячеек. Отличия форматов описаны ниже:

- В формате ODS и XLSX разделитель целой и дробной части будет преобразован в соответствии с настройками локали на компьютере пользователя (для российской локали - это запятая ",");
- В формате CSV поддерживается объединение ячеек, форматирование значений, разделители разрядов;
- В формате CSV разделитель целой и дробной части не преобразуется, т.о. контроль за корректным форматом чисел осуществляет пользователь
- В формате CSV нет поддержки группировки строк при экспорте иерархических таблиц.

6.1.2.4 Блок «Табличная область данных»

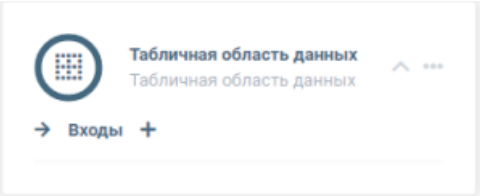


Рисунок 133 – Пример блока «Табличная область данных»

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.1.2.4.1 Назначение

Блок позволяет сформировать таблицу из нескольких срезов с данными.

6.1.2.4.2 Настройки блока

Название параметра	Тип параметра	Обязательный	Описание	Пример
Привязки	JSON	Да	Массив с параметрами привязок срезов к их расположению в визуализаторе. Настройки параметров: • **id** – идентификатор входа. • **row** – номер «строки» для размещения среза. Номера могут идти не по порядку, в таком случае пропущенные номера будут игнорироваться. • **col** – номер «столбца» для размещения среза. Номера могут идти не по порядку. Номера «столбцов» должны быть уникальны только в рамках одной «строки». В данном случае «строки» и «столбцы» – это не строки и столбцы одной таблицы, а условное расположение одного среза данных в общем массиве срезов. (В одном срезе может быть больше одной строки/столбца с данными)	`[ {"id": "1", "row": 2, "col": 1}, {"id": "2", "row": 2, "col": 2}, {"id": "3", "row": 3, "col": 1}, {"id": "4", "row": 3, "col": 2}, {"id": "5", "row": 4, "col": 1}, {"id": "6", "row": 4, "col": 1} ]`

6.1.2.4.3 Группа «Общие настройки блока»

Аналогично приведенным в п. 6.1.1.1.3.

6.1.2.4.4 Входы

Каждый новый срез добавляется в отдельный вход. Параметры «Обязательный» и «Массив» должны быть включены. Параметр «Идентификатор» необходимо указать в поле **id** в настройках «Привязки».

Для создания входа необходимо нажать на значок + в разделе входов блока. Для каждого из входов задается следующий перечень полей.

Название входа	Тип	Обязательный	Описание
Идентификатор	Строка	Да	Идентификатор входа – используется внутри блока, для получения доступа к данным входа

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название входа	Тип	Обязательный	Описание
Имя	Строка	Да	Имя входа для отображения в списке входов блока
Описание	Строка	Нет	Дополнительная информация с описанием входа
Тип	Выпадающий список	Да	Тип данных, которые будут приходить на вход. Нужен для построения корректных связей с другими блоками. По умолчанию установлен тип "JSON".
Обязательный	Флаг	Да	Обязательность входа. Если вход обязательный и на него не пришло данных – блок не будет рассчитан с ошибкой

6.1.2.4.5 Выходы

У блока нет выходов.

6.1.2.4.6 События блока

У блока нет событий.

6.1.2.5 Описание работы блока

Блок формирует визуализатор таблицы на основе привязок и всех срезов, поступивших на вход.

Порядок формирования визуализатора:

1. Срезы группируются по строкам в зависимости от привязок.
2. По каждой строке идет формирование общего среза, в зависимости от привязок по столбцам.
3. Строка выравнивается по высоте – ищется срез с максимальным количеством строк, и остальные срезы подгоняются к нему по высоте.
    a. Если стока является первой в общем списке строк – в ней ищутся настройки шапки, и, если они есть, добавляются к визуализатору.
4. После формирования всех строк, вся таблица выравнивается по ширине – ищется строка с максимальным количеством столбцов и остальные строки подгоняются к ней по ширине.
    a. Если есть параметр **columns** (описание заголовка столбца) – то он так же подгоняется по ширине (добавляются пустые заголовки справа, если данных больше чем заголовков).
    b. Если есть параметр **colWidths** (ширина столбца) – то он так же подгонятеся по ширине (на недостающие значения справа проставляется число 120).

6.1.2.5.1 Настройки визуализатора

Визуализатор не имеет специфических настроек. Общий список настроек приведен в п. 4.2.8 «Настройки визуализатора».

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.1.2.6 Блок «Табличный срез»

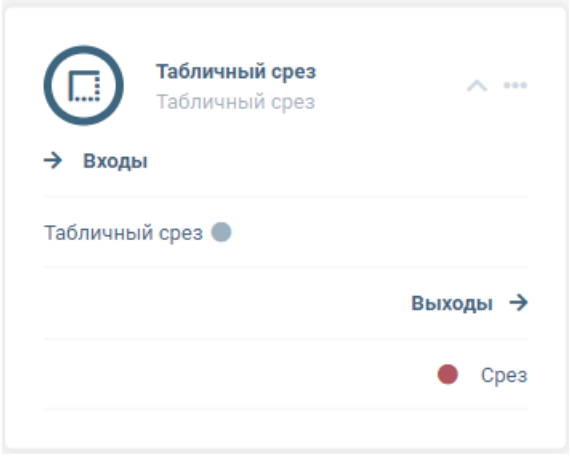


Рисунок 134 – Пример блока «Табличный срез»

6.1.2.6.1 Назначение

Блок позволяет формировать данные для отображения в табличной области данных

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.1.2.6.2 Настройки блока

Название параметра	Тип параметра	Обязательный	Описание	Пример

Табличный срез	JSON	Нет	Содержит данные и их стили в формате JExcel, позволяющем отображать и редактировать их в мастере работы с табличным срезом.	**Табличный срез**
				{
				"data": [
				{
				"0": "номер",
				"1": "показатели",
				"2": "",
				"3": "страна",
				"4": "",
				"5": "",
				"6": ""
				},
				{
				"0": "",
				"1": "",
				"2": "",
				"3": "МО 1",
				"4": "",
				"5": "МО 2",
				"6": ""
				},
				{
				"0": "",
				"1": "отрасль 1",
				"2": "отрасль 2",
				"3": "город 1",
				"4": "город 2",
				"5": "город 3",
				"6": "город 4"
				}
				],
				"style": {
				"A1": "text-align: center; overflow: hidden; background-color:

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
				rgb(255, 243, 224); writing-mode: vertical-lr; font-weight: bold; font-size: 18px;", "B1": "text-align: center; overflow: hidden; background-color: rgb(224, 242, 241);", "C1": "text-align: center; overflow: hidden; background-color: rgb(224, 242, 241);", "D1": "text-align: center; overflow: hidden; background-color: rgb(227, 242, 253);", "E1": "text-align: center; overflow: hidden; background-color: rgb(227, 242, 253);", "F1": "text-align: center; overflow: hidden; background-color: rgb(227, 242, 253);", "G1": "text-align: center; overflow: hidden; background-color: rgb(227, 242, 253);", "A2": "text-align: center; overflow: hidden; background-color: rgb(255, 243, 224); writing-mode: vertical-lr; font-weight: bold; font-size: 18px;", "B2": "text-align: center; overflow: hidden; background-color: rgb(224, 242, 241);", "C2": "text-align: center; overflow: hidden; background-color: rgb(224, 242, 241);", "D2": "text-align: center; overflow: hidden; background-color: rgb(227, 242, 253);", "E2": "text-align: center; overflow: hidden; background-color: rgb(227, 242, 253);",

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
				"F2": "text-align: center; overflow: hidden; background-color: rgb(227, 242, 253);", "G2": "text-align: center; overflow: hidden; background-color: rgb(227, 242, 253);", "A3": "text-align: center; overflow: hidden; background-color: rgb(255, 243, 224); writing-mode: vertical-lr; font-weight: bold; font-size: 18px;", "B3": "text-align: right; overflow: hidden; background-color: rgb(224, 242, 241);", "C3": "text-align: right; background-color: rgb(224, 242, 241); overflow: hidden;", "D3": "text-align: left; overflow: hidden; background-color: rgb(227, 242, 253);", "E3": "text-align: left; overflow: hidden; background-color: rgb(227, 242, 253);", "F3": "text-align: left; overflow: hidden; background-color: rgb(227, 242, 253);", "G3": "text-align: left; background-color: rgb(227, 242, 253);" }, "mergeCells": { "A1": [1, 3], "B1": [2, 2],

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
				"D1": [4, 1], "D2": [2, 1], "F2": [2, 1] }, "comments": {}, "colWidths": ["120", "120", "120", "120", "120", "120", "120"], "columns": [{ "type": "text", "name": "0", "source": [], "options": [], "editor": null, "allowEmpty": false, "title": "", "width": "120", "align": "left" }, { "type": "text",

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
				"name": "1", "source": [], "options": [], "editor": null, "allowEmpty": false, "title": "", "width": "120", "align": "left" }, { "type": "text", "name": "2", "source": [], "options": [], "editor": null, "allowEmpty": false, "title": "", "width": "120", "align": "left" }, { "type": "text", "name": "3", "source": [], "options": [], "editor": null, "allowEmpty": false, "title": "", "width": "120", "align": "left" }, { "type": "text", "name": "4", "source": [], "options": [], "editor": null,

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример

				"allowEmpty": false,
				"title": "",
				"width": "120",
				"align": "left"
				},
				{
				"type": "text",
				"name": "5",
				"source": [],
				"options": [],
				"editor": null,
				"allowEmpty": false,
				"title": "",
				"width": "120",
				"align": "left"
				},
				{
				"type": "text",
				"name": "6",
				"source": [],
				"options": [],
				"editor": null,
				"allowEmpty": false,
				"title": "",
				"width": "120",
				"align": "left"
				}
				]
				}

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.1.2.6.3 Группа «Общие настройки блока»

Аналогично приведенным в п. 6.1.1.1.3.

6.1.2.6.4 Входы

Название входа	Тип	Обязательный	Описание
Табличный срез	JSON	Нет	Содержит данные и их стили в формате, позволяющем отображать и редактировать их в мастере работы с табличным срезом При наличии данных на этом входе, настройка «Табличный срез» не будет учитываться во время формирования блока

6.1.2.6.5 Выходы

Название выхода	Тип	Описание
Срез	JSON	Данные таблицы, представленные в специальном формате для отображения в табличной области данных

6.1.2.6.6 События блока

В блоке нет событий.

6.1.2.6.7 Описание работы блока

Блок преобразует ряды или объекты, поступившие на вход в табличные данные, по аналогии с блоком «Таблица» (см. п. 6.1.2.3). Однако на выходе представляет визуализатор, а выдает специализированный табличный JSON (похожий на формат JExcel, но с некоторыми изменениями для возможности формирования таблиц сложной структуры).

Блок «Табличная область данных» (см. п. 6.1.2.4) принимает этот формат данных и преобразует его в визуализатор.

Пример выходных данных блока:

```
{
    "type": "sliceHeader",
    "data": [
        [
            "COL 1",
            "COL 1.2",
            ""
        ],
        [
            "",
            "COL 2.2",
            "COL 2.3"
        ]
    ],
    "style": [
        [
            "text-align: center; overflow: hidden; font-weight: bold;",
            "text-align: center; overflow: hidden; font-weight: bold;",
            "text-align: center; overflow: hidden; font-weight: bold;"
        ],
        [
            "text-align: center; overflow: hidden; font-weight: bold;",
```


© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

```
                "text-align: center; overflow: hidden; font-weight: bold;",
                "text-align: center; font-weight: bold;"
            ]
        ],
        "mergeCells": [
            [
                [
                    1,
                    2
                ],
                [
                    2,
                    1
                ]
            ]
        ],
        "comments": [],
        "colWidths": [
            "120",
            "120",
            "120"
        ]
}
```


- **type** – тип данных. Для всех данных используется тип silceHeader;
- **data** – массив с данными по строкам (массив массивов);
- **style** – массив массивов со стилями каждой ячейки;
- **mergeCells** – массив массивов с индексами ячеек, которые должны быть объеденены;
- **comments** – массив с комментариями;
- **colWidths** – массив с данными о ширине каждого столбца.

6.1.2.6.8 Мастер блока

Просматривать и редактировать табличный срез можно с помощью мастера работы с табличным срезом.

Чтобы открыть мастер необходимо в локальном меню блока выбрать соответствующий пункт (Рисунок 135).

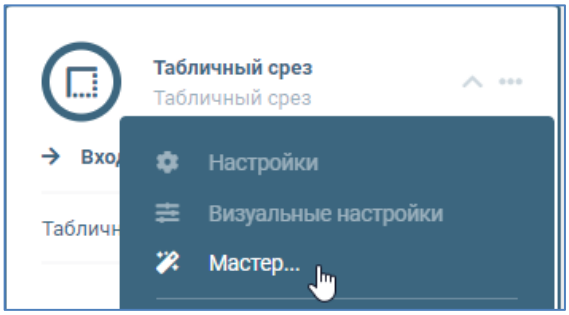


Рисунок 135 – Пример раздела меню «Мастер блока»

После чего открывается окно мастера (Рисунок 136).

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

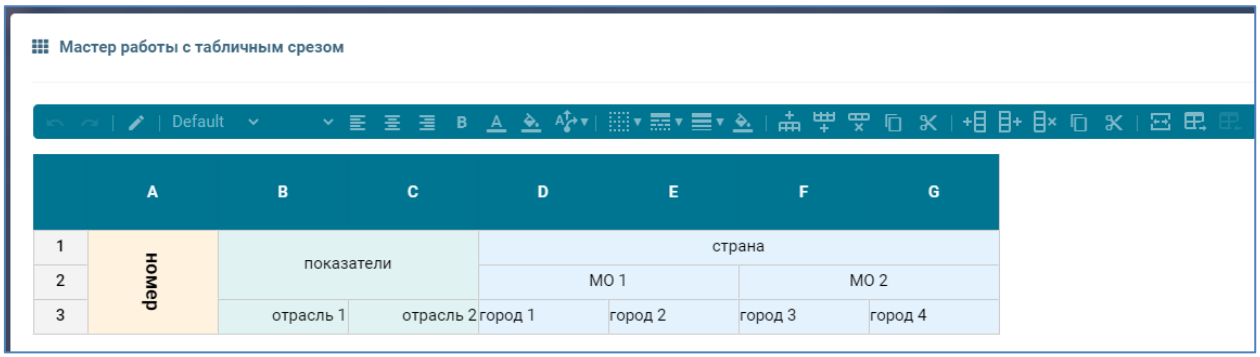


Рисунок 136 – Пример блока «Мастер работы с табличным срезом»

Панель инструментов мастера работы с табличным срезом включает:

	Отменить изменение	
	Вернуть отмененное изменение	
	Редактировать	
	Откатить	
	Сохранить	
	Шрифт	
	Размер шрифта	
	Выровнять по левому краю	
	Выровнять по центру	
	Выровнять по правому краю	
	Полужирный текст	
	Цвет текста	
	Цвет заливки текста	
	Направление текста	Позволяет выбрать между горизонтальным (слева направо) и вертикальным (сверху вниз) направлением текста
	Границы	
	Стиль границ	
	Толщина границ	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

	Цвет границ	
	Вставить строки выше	
	Вставить строки ниже	
	Удалить строки	
	Копировать строки	
	Вырезать строки	
	Вставить столбцы слева	
	Вставить столбцы справа	
	Удалить столбцы	
	Копировать столбцы	
	Вырезать столбцы	
	Разъединить объединенные ячейки	
	Присоединить следующий столбец к боковику	
	Убрать последний столбец боковика	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.1.3 Карта

6.1.3.1 Блок «3D-карта»

6.1.3.1.1 Назначение

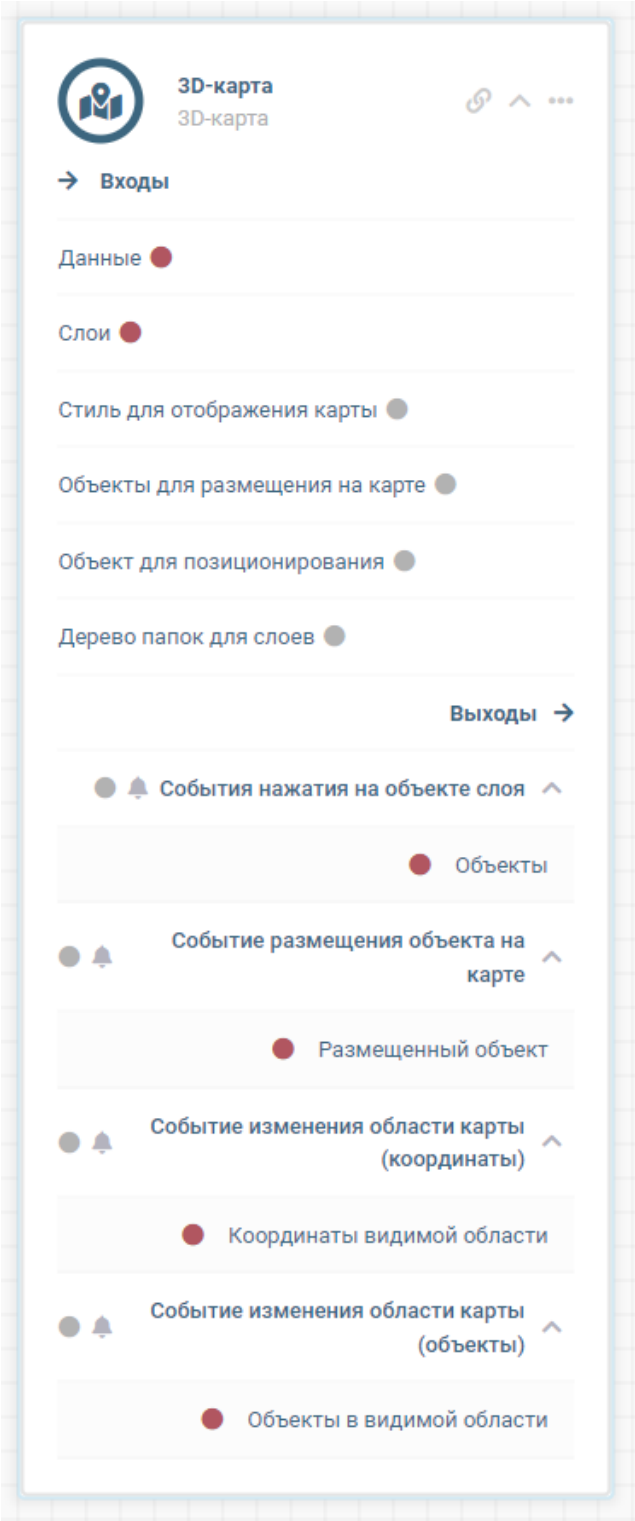


Рисунок 137 – Пример блока «3D-карта»

Блок 3D-Карта позволяет визуализировать информацию на картографической подложке.

Данные в блок поступают на входы, блок обрабатывает их и подготавливает визуализатор. Блок содержит множество вариантов настроек и обладает визуализатором с достаточно широким функционалом. Основной концепт работы карты заключается в том, что каждый

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

слой имеет свой стиль, в том числе и слой подложки – каждый элемент подложки, будь то озеро, лес, дорога или номер дома прописан в стиле. Этот стиль является основным элементом карты, т.к. определяет, откуда брать данные для отрисовки картографической подложки и как их отображать. Стили карты пишутся на основе MapBox стилей (Документация по стилям MapBox Layer Style). В приложении заранее загружены несколько стилей карты со свободной лицензией.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.1.3.1.2 Настройки блока

Название параметра	Тип параметра	Обязательный	Описание	Пример
Настройки стиля карты	Выпадающий список	Да	Выбирает откуда брать стиль для карты, из списка «Глобальные стили карты» или задавать вручную. По умолчанию задается ручной стиль со светлой подложкой	
Стиль	JSON	Да	Доступен, если в настройках стилей выбран вариант «Задать параметры вручную». Определяет стиль карты. Важно! Если на вход блока «Стиль отображения карты» был передан стиль, эта настройка будет игнорироваться	
Выбор стиля	Список	Да	Доступен, если в настройках стилей выбран вариант «Глобальные стили». Определяет стиль карты из списка глобальных стилей в приложении (см. п. 7.2.3 «Глобальные стили карты» Руководство администратора)	
Центр карты, широта	Вещественное число	Нет	Начальное положение карты, широта (latitude)	0
Центр карты, долгота	Вещественное число	Нет	Начальное положение карты, долгота (logitude)	0
Начальный масштаб	Целое число	Нет	Начальный масштаб карты	2

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
Разрешить масштабирование	Флаг		Разрешить или нет изменение масштаба карты	
Минимальный масштаб	Целое число	Нет	Минимальный масштаб, который будет доступен	0
Максимальный масштаб	Целое число	Нет	Максимальный масштаб, который будет доступен	18
Угол наклона камеры	Целое число	Да	Угол наклона камеры карты, который будет выставлен по умолчанию	0
Азимут	Целое число	Нет	Азимут камеры, который будет выставлен по умолчанию	0
Сглаживание	Флаг	Да	Использовать или нет сглаживание на карте (antialiasing), по умолчанию включено	
Включить 3D	Флаг	Да	Включить или отключить возможность отображения в 3D. При отключении карта будет выглядеть как 2D и настройки камеры не будут применяться	
Пользовательские иконки	Строка	Нет	Адрес откуда карте брать иконка для точечного слоя pointsLayer (https://docs.mapbox.com/mapbox-gl-js/style-spec/layers/#symbol)	/static/map/custom-icons/city/icons.json
Редактирование объектов	Флаг	Нет	Доступность редактирования слоев карты. **Важно!** Если в описании слоя есть флаг **editable=true** эта настройка будет игнорироваться	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
Отображать карточку при переходе на объект	Флаг	Нет	Включить отображение карточки при переходе на объект. По умолчанию – включено	
Отображать отсутствие карточки	Флаг	Нет	Включить отображение пустой карточки (отображение карточки «Нет данных»). По умолчанию – включено	
Режим отображения карточки	Выпадающий список	Нет	Настройки режимов отображения карточки. Возможные значения: • **Карточки отображаются** – при нажатии на объект с карточкой, последняя отобразится (по умолчанию). • **Карточки не отображаются** – при нажатии на объект с карточкой, она не откроется	
Раздел «Настройки карточки»				
Цвет карточки	Да	Да	Задает цвет фона карточек объектов	rgba(255, 255, 255, 1)
Цвет границы карточки	Цвет	Да	Задает цвет границы карточек объектов	rgba(255, 255, 255, 1)

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
Тип границы карточки	Выпадающий список	Да	Тип границы для карточки объекта: • Сплошная линия • Штриховая линия • Линия из точек • Двойная линия • Скрытая линия • Отсутствует	Сплошная линия
Толщина границы карточки	Целое число	Да	Задает толщину границ карточек объектов	1
Цвет текста	Цвет	Да	Задает цвет текста карточек объектов (если он не переопределен стилем в html внутри)	rgba(0, 0, 0, 1)
Вращение камеры	Флаг	Да	Запускать или нет эффект вращения камеры при отсутствии действий с картой	
Отображать координаты курсора	Флаг	Да	Отображать или нет координаты курсора на карте	
Расположение	Выпадающий список	Нет	Доступен при включении «Отображать координаты курсора». Позволяет выбрать угол карты, в котором отображать координаты: • Справа внизу • Справа вверху • Слева вверху • Слева внизу	По умолчанию «Справа внизу»

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
Единицы	Выпадающий список	Нет	Доступен при включении «Отображать координаты курсора». Позволяет выбрать единицы отображения: • Десятичные градусы (DD) • Градусы, минуты, секунды (DMS)	По умолчанию «Десятичные градусы (DD)»
Отображать текущий масштаб	Флаг	Да	Отображать текущий масштаб на карте	
Расположение	Выпадающий список	Нет	Доступен при включении "Отображать текущий масштаб". Позволяет выбрать угол карты, в котором отображать координаты: • Справа внизу • Справа вверху • Слева вверху • Слева внизу	По умолчанию "Справа внизу"
Вид	Выпадающий список	Нет	Доступен при включении "Отображать текущий масштаб". Позволяет выбрать текст для описания масштаба: • Фактор: текст вида "7.8" • Пропорция: текст вида "1:100 000"	По умолчанию "Фактор"
Отображать инструменты	Флаг	Да	Отображать или нет инструменты измерения на карте	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
Поиск на карте	Флаг	Да	Отображать или нет поиск на карте	
Исключить из поиска	Строка	Нет	Доступен при включении «Поиск на карте». Атрибуты слоев, которые требуется исключить из поиска. Ввод атрибутов через разделитель «;» (точка с запятой), без пробелов (наименование регистрозависимое): **Attr1;Attr2;...;AttrN**	popup;attr1;region
Скрывать не привязанные слои	Флаг	Да	В случае, если дерево слоев задается через вход «Дерево папок для слоев» включение флага скроет с карты все слои, которые отсутствуют в иерархии	
Отображать индикатор загрузки визуализатора	Флаг	Нет	Включение индикатора загрузки визуализатора. По умолчанию – индикатор включен	
Раздел "Геокодирование"				

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример

Геокодер	Выпадающий список	Нет	Выбор геокодера для расширеного поиска объектов. Возможные варианты:
			• Без геокодирования
			• Геокодер Dadata
			  ○ Подробнее см. https://dadata.ru/
			• Геокодер Яндекс (подробнее https://yandex.ru/maps-api/products/geocoder-api)
			• Встроенный геокодер
			  ○ В версии 0.38 функционал находится в разработке	

API-ключ DaData	Шифрованная строка	Нет	Только для геокодера Dadata. API ключ для подключения геокодера.	
Секретный ключ DaData	Шифрованная строка	Нет	Только для геокодера Dadata. Секретный ключ для подключения геокодера.	
API-ключ Яндекс	Шифрованная строка	Нет	Только для геокодера Яндекс. API-ключ для подключения геокодера Яндекс	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
Ввод пароля	Выпадающий список	Нет	Поле появляется, если выбраны геокодеры Dadata или Яндекс. Настройка применяется только к выбранному геокодеру. Выбор способа ввода пароля. Возможные варианты: • Вручную • Глобальные пароли - выбор пароля из списка глобальных паролей	
Проксирование запроса	Флаг	Нет	*Только для геокодера Яндекс.* Флаг для передачи запросов к геокодеру Яндекс через сервис посредник	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример

Настройки проксирования запроса	JSON	Нет	*Только для геокодера Яндекс при включённом проксировании запросов*
			Доступна подстановка глобальных паролей для аутентификации на прокси-сервере. Подробнее в 11.17 Руководства Администратора «Управление глобальными паролями».
			Настройки для проксирования. Обязательные параметры - "headers" и "host". В разделе "headers" описываются HTTP-заголовки, которые будут добавлены ко всем уходящим на прокси запросам. В разделе "host" указывается адрес прокси-сервера.
				{
				"headers": {"X-Gravitee-Api-Key": {"$pass": "gravitee-api-key"}},
				"host": "http://192.168.88.240:8082/yageocoder"
				}
				Здесь gravitee-api-key - ключ глобального пароля. Им можно управлять через раздел "Пароли" в меню администрирования

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.1.3.1.3 Группа «Общие настройки блока»

Аналогично приведенным в п. 6.1.1.1.3.

6.1.3.1.4 Входы

Название входа	Тип	Обязательный	Описание
Данные	JSON	Да	Данные для слоев, с идентификаторами. Вход общий – все выходы с данными блоков «Слой данных для 3D-карты» (п. 6.1.3.5) входят в этот.
Слои	JSON	Да	Список слоев для отображения на карте, с идентификаторами и настройками. Общий – все выходы с описанием слоев блоков «Слой данных для 3D карты» (п. 6.1.3.5) входят в этот.
Данные карточки	Строка	Нет	Данные карточки выбранного объекта при включенных динамических карточках
Стиль для отображения карты	JSON	Нет	Стиль для отображения подложки карты. Заменяет тот, который указывается в настройках блока. Может динамически генерироваться в другом блоке, перед картой
Объекты для размещения на карте	Объекты	Нет	Объекты, которые будут сформированы в список объектов для размещения на карте (см. п. 6.1.3.1.8.5 «Размещение заранее подготовленных объектов на карте»).
Объекты для позиционирования	Объекты	Нет	Массив объектов, на которых необходимо позиционировать карту при открытии или обновлении во время события (см. п. 6.1.3.1.8.5 «Размещение заранее подготовленных объектов на карте»)
Дерево папок для слоев	Объекты	Нет	Список разделов для отображения иерархии в дереве слоев (см. п. 6.1.3.1.8.4 «Иерархический список слоев»).

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название входа	Тип	Обязательный	Описание

Видимость слоев	JSON	Нет	Список слоев с указанием флага видимости для них. Формат данных:
			`{`
			`"layer_id_1": {"visible": true},`
			`"layer_id_2": {"visible": false}`
			`}`
			В случае некорректного идентификатора слоя (идентификатор должен совпадать с ID слоя, переданным во входе «**Данные**») и/или некорректного флага видимости (словарь отличный от "visible": true/false), будет выдано предупреждение и данная строка не попадет в итоговый визуализатор

6.1.3.1.5 Выходы

В блоке нет выходов.

6.1.3.1.6 События блока

Название выхода	Тип	Описание
Событие нажатия на объекте слоя	Объекты	Происходит при нажатии левой кнопкой мыши на карте. Если под курсором находятся объекты со слоев – они будут выведены в этот событийный выход
Событие размещения объекта на карте	Объекты	Происходит при размещении объекта на карте (см. п. 6.1.3.1.8.5 «Размещение заранее подготовленных объектов на карте»)
Событие изменения области карты (координаты)	JSON	Происходит при изменении масштаба карты. На него выдается левая верхняя и правая нижняя координаты текущего viewport карты (bbox)
Событие изменения области карты (объекты)	Объекты	Происходит при изменении масштаба карты. На него выдаются объекты, которые в данный момент находятся во viewport карты
Изменение видимости слоев	Объекты	Происходит при изменении видимости слоев карты. Возвращаются все слои и объекты карты с указанием видим он или нет. Без координат и атрибутов

6.1.3.1.7 Описание работы блока

Блок собирает информацию о слоях со своих входов и готовит визуализатор 3D-карта. Для оформления подложки, а также для определения источника данных для нее используется специальный стиль. Стиль основан на основе MapBox стилей (Документация по стилям MapBox Layer Style).

В качестве источника данных для стилей могут выступать векторные и растровые источники картографических тайлов.

Список источников, которые были проверены на работоспособность:

- Maptiler;
- OSM;
- Bing;

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

- Geoserver;
- Yandex Tiles API.

Важно! Для корректной работы карты в ней должен быть хотя бы один слой, т.к. входы являются обязательными.

6.1.3.1.7.1 Порядок слоев на входах

Для входа Слои важен порядок, в котором они привязываются ко входу или к дочерним входам. Именно в этом порядке слои будут отрисованы на карте. т.е. если слой непрозрачный и расположен выше (связь заведена во вход последней) – этот слой будет отрисован последним и будет перекрывать остальные слои.

При этом дочерние входы, как и в других случаях, позволяют управлять своим порядком.

Особенность

Благодаря гибкому механизму стилей можно создавать слои без данных – в этом случае в слое должен быть указан идентификатор источника из другого слоя. Таким образом можно на одних и тех же данных строить разные слои. Например, полигоны для отображения фигурой с заливкой и их границы можно использовать один набор данных и два слоя **fill** и **line**. В этом случае один их них может не содержать данных, что позволит избежать дублирования данных и переиспользовать их.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Пример с блоком, у которого есть дочерние входы для входа «Слои»:

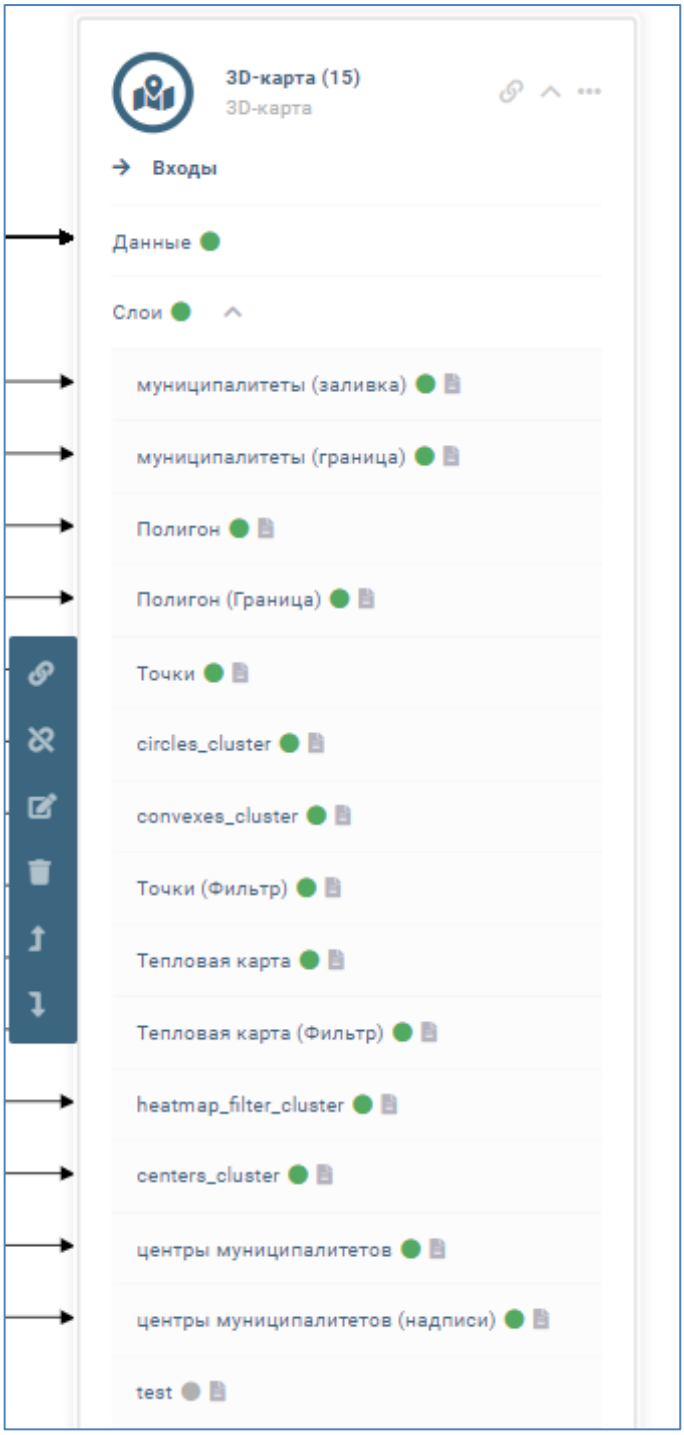


Рисунок 138 – Пример блока «3D-карта»

6.1.3.1.7.2 Расчет блока

Во время расчета блока проводится множество проверок на целостность данных для избегания проблем при отрисовке визуализатора.

- В первую очередь проверяется что все данные имеют описание своих слоев.
- Проверяется что объект для позиционирования есть в списке слоев.
- Проверяется корректность настроек.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

- Проверяется какие слои могут быть редактируемы и имеют возможность сохранения. Эти слои должны быть из блоков с поддержкой сохранения данных, например, справочник.

6.1.3.1.7.3 Кэширование

Для обеспечения скорости работы данные для блока карта кэшируются в БД и при работе с событиями блок получает из кэша оригинальный объект для обеспечения сохранения или срабатывания события.

6.1.3.1.7.4 Позиционирование на объекте при расчете блока

При расчете блока, если на вход **Объекты для позиционирования** был подан объект или массив объектов – карта будет позиционироваться на нем. При этом, это может быть как объект при открытии приложения, так и объект, поступивший по цепочке событий. В этом случае визуализатор переместит камеру на переданный во вход объект.

Если на вход пришел массив объектов, то блок попытается вычислить видимую область, на которую будет произведено позиционирование.

Если этот вход не пустой – осуществляется проверка, что указанные объекты есть в списке объектов на каком-либо слое. Если объект не найден – выдается предупреждение и объект убирается из массива для позиционирования. В том случае если после расчета массив оказался пустым – позиционирования не происходит.

Важно! Объект должен быть на видимом слое, иначе позиционирование будет произведено на «пустое» место на карте.

6.1.3.1.8 Описание работы визуализатора

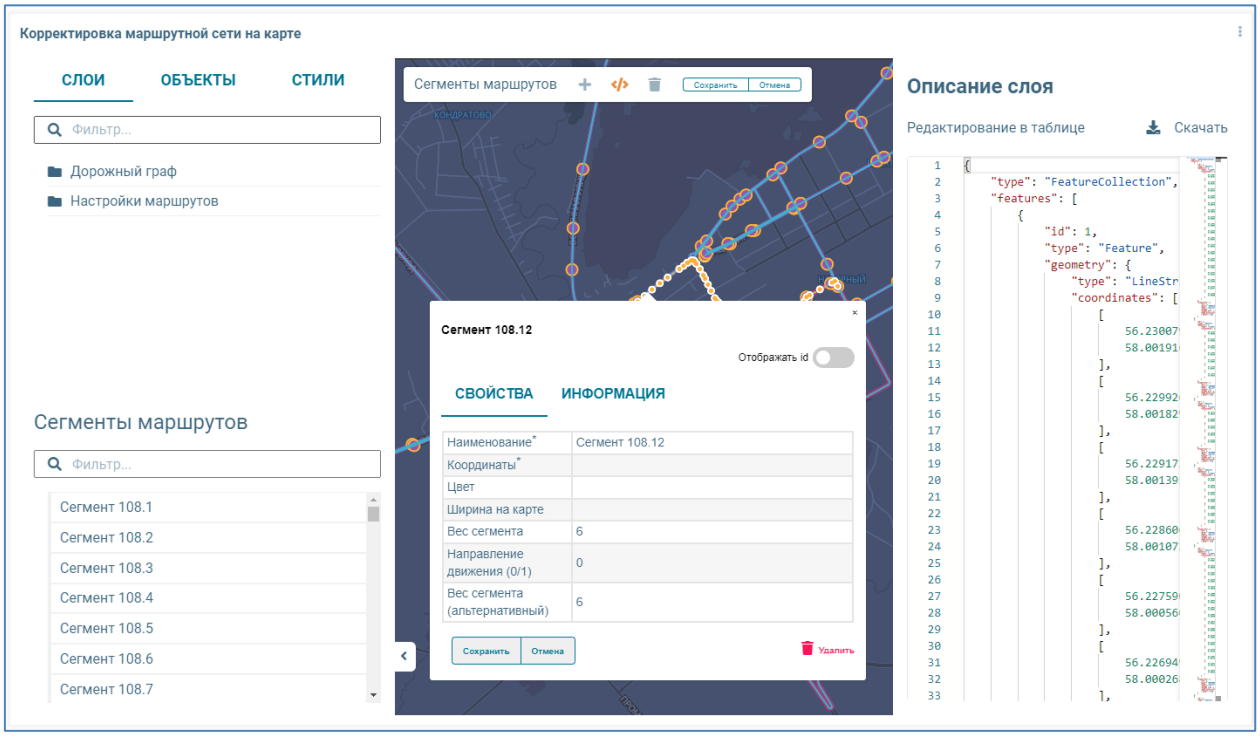


Рисунок 139 – Внешний вид визуализатора «3D-карта»

6.1.3.1.8.1 Описание визуализатора

Визуализатор «3D-карта» разработан на основе библиотеки maplibre.gl (MIT, https://github.com/maplibre/maplibre-gl-js). Используя стили, он выводит слои на карту и позволяет пользователю взаимодействовать с ними. В карте используется проекция Меркатора.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Сам визуализатор состоит из 3 основных частей:

1. Карта.
2. Панель слоев/объектов для размещения/стилей подложек.
3. Панель с данными слоя (доступна только в режиме редактирования слоя).

Важно! При сохранении состояния карты, в случае выбора статического состояния, будет сохранена позиция камеры, которая в текущий момент установлена на карте. Эта настройка переопределит начальное положение, которое было задано в блоке.

6.1.3.1.8.2 Инструменты карты

На карте есть панель инструментов, которая позволяет вызвать тот или иной инструмент и управлять картой.

Рисунок 140 – Пример панели инструментов карты

1. Поиск на карте – открывает раздел поиска.
2. Увеличить/уменьшить масштаб карты.
3. Текущее положение камеры карты относительно севера. При клике возвращает к положению «север-сверху».
4. Экспорт текущей карты в png.
5. Инструменты измерения на карте.

6.1.3.1.8.3 Список слоев

Слои, поступающие на карту, отображаются в левой панели согласно своим настройкам видимости и группировками. Пример панели слоев приведен на рисунке (Рисунок 141).

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

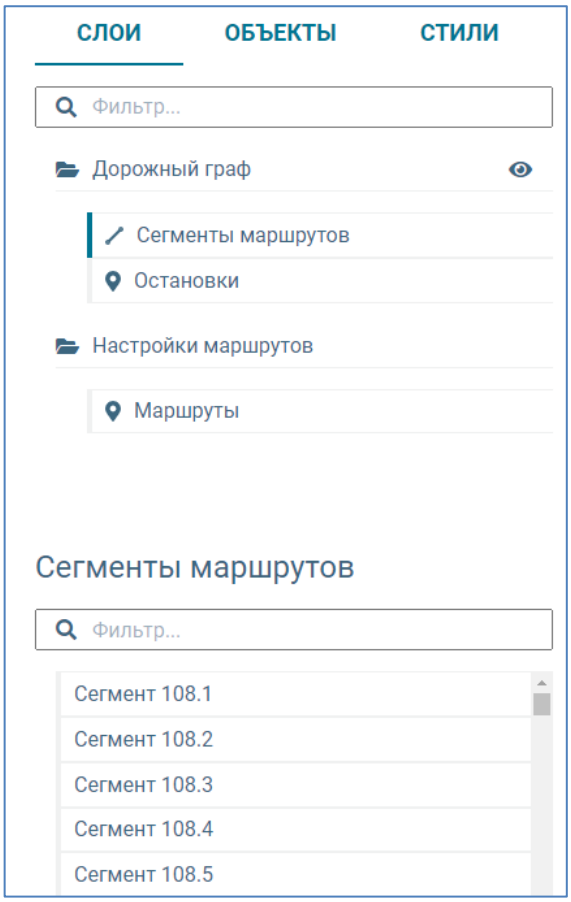


Рисунок 141 – Пример панели слоев

- При наведении курсора на слой или группу появляются дополнительные элементы управления:

  – скрыть или отобразить слой или групп;

  – позиционирование на слое (для группы кнопка недоступна).

- Доступен поиск по наименованию слоя или группы.

В дереве слоев, если не применяется иерархия, доступно меню по нажатию на правую клавишу мыши:

- Показать все слои – включить отображение всех слоев.
- Скрыть все слои – выключить отображение всех слоев.
- Показать по умолчанию – вернуть состояние видимости к дефолтному, которое задано при открытии карты.

При выборе слоя можно просмотреть список его объектов.

Важно! Просмотр объектов недоступен для слоев **3D-объекты, Маршруты, Дуги, 3D-модели, Гексагональная тепловая карта, Внешний источник данных (геосервер), Геосервер Martin.**

Список объектов представляет собой линейный список, где каждый элемент – это объект слоя. При просмотре слоя доступно позиционирование на объекте, его отображение или скрытие, и, если перейти в режим редактирования – удаление.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Важно! При формировании списка имена для объектов берутся из поля **name** оригинального ряда, если оно было передано в слой. Иначе вместо имени будет отображаться уникальный GUID объекта.

6.1.3.1.8.4 Иерархический список слоев

В случае если на вход блока «Дерево папок для слоев» приходит список объектов, отражающих иерархию слоев, поведение панели изменяется. На вход должны прийти данные из блока «Справочник» с включенным флагом «Элементы имеют иерархию» в котором задается иерархия и который содержит описание папок для отображения в дереве.

Важно! Порядок элементов будет зависеть от поля Порядок (ord) в справочнике с иерархией

Пример входных данных для иерархии папок:

```
[
    {
        "pkey": {
            "id": 1
        },
        "_params": {
            "ts": [],
            "lazy": [],
            "parent_field": "parent",
            "key_field": "id"
        },
        "attrs": {
            "name": "город Пермь",
            "ord": 1,
            "parent": null
        },
        "child": [
            {
                "pkey": {
                    "id": 2
                },
                "_params": {
                    "ts": [],
                    "lazy": [],
                    "parent_field": "parent",
                    "key_field": "id"
                },
                "attrs": {
                    "name": "Школы",
                    "ord": 1,
                    "parent": 1
                },
                "child": [],
                "_source": ""
            },
            {
                "pkey": {
                    "id": 3
                },
                "_params": {
                    "ts": [],
                    "lazy": [],
```


© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

```
                "parent_field": "parent",
                "key_field": "id"
            },
            "attrs": {
                "name": "Больницы",
                "ord": 2,
                "parent": 1
            },
            "child": [],
            "_source": ""
        }
    ],
    "_source": "c5022c4a-e763-48b9-9b08-e746065f18e7"
},
{
    "pkey": {
        "id": 4
    },
    "_params": {
        "ts": [],
        "lazy": [],
        "parent_field": "parent",
        "key_field": "id"
    },
    "attrs": {
        "name": "Кунгур",
        "ord": 2,
        "parent": null
    },
    "child": [
        {
            "pkey": {
                "id": 5
            },
            "_params": {
                "ts": [],
                "lazy": [],
                "parent_field": "parent",
                "key_field": "id"
            },
            "attrs": {
                "name": "Школы",
                "ord": 1,
                "parent": 4
            },
            "child": [],
            "_source": ""
        },
        {
            "pkey": {
                "id": 6
            },
            "_params": {
                "ts": [],
                "lazy": [],
```


© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

```
                    "parent_field": "parent",
                    "key_field": "id"
                },
                "attrs": {
                    "name": "Больницы",
                    "ord": 2,
                    "parent": 4
                },
                "child": [],
                "_source": ""
            },
            {
                "pkey": {
                    "id": 7
                },
                "_params": {
                    "ts": [],
                    "lazy": [],
                    "parent_field": "parent",
                    "key_field": "id"
                },
                "attrs": {
                    "name": "Пещеры",
                    "ord": 3,
                    "parent": 4
                },
                "child": [],
                "_source": ""
            }
        ],
        "_source": "c5022c4a-e763-48b9-9b08-e746065f18e7"
    }
]
```


В этом случае, слои, у которых есть атрибут **groupName**, будут расположены в соответствующих элементах иерархии. Если **groupName** не задан – элемент будет размещен в папку «Прочие слои» на первом уровне.

Скрытие непривязанных слоев

Папка "Прочие слои" будет отображаться в том случае, если флаг "Скрывать не привязанные слои" выключен.

Разделение папок и слоев

Папка не может являться одновременно и слоем. Если обнаружена привязка папки и слоя, то будет выдано предупреждение и слой не будет отображаться в визуализаторе

Дублирование слоев

Если один и тот же слой привязан к нескольким объектам, то будет отображено только первое найденное вхождение слоя.

Расширенные возможности привязки - для продвинутых пользователей

В блоке "Слой данных для 3D карты" в разделе "Описание слоя" можно задать более точную привязку слоя к иерархическому справочнику.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Формат записи:

```
"dictMapping": {"key": "mapping", "value": 10}
```

key - атрибут или первичный ключ справочника, для которого осуществляется привязка.

value - значение атрибута.

т.е. в примере, слой будет привязан к элементу, у которого значение атрибута mapping равно 10.

В случае, если в описании слоя атрибут dictMapping не найден (осуществляется привязка по умолчанию), то формально ее можно записать следующим образом:

```
"dictMapping": {"key": "id", "value": <layer_id>}
```

где <layer_id> - идентификатор слоя.

Так же в случае отображения иерархии, будет изменено контекстное меню дерева слоев:

- Показать все слои – включить отображение всех слоев.
- Скрыть все слои – выключить отображение всех слоев.
- Показать подчиненные слои – показать все слои в текущей папке и ее вложенных папках.
- Скрыть подчиненные слои – скрыть все слои в текущей папке и ее подпапках.
- Показать слои уровня – показать все слои всех папок текущего уровня иерархии.
- Скрыть слои уровня – скрыть все слои всех папок текущего уровня иерархии.
- Показать по умолчанию – вернуть состояние видимости к дефолтному, которое задано при открытии карты.

Работа со списком объектов не меняется.

Управление z-индексами слоев

В списке объектов, приходящих на вход "Дерево папок для слоев" можно передать технический атрибут z_index, который будет отвечать за порядок слоя.

Индекс должен приходить в формате целочисленного числа. Любой другой тип индекса будет вызывать ошибку при расчете блока.

Для папок индексы не проверяются и не учитываются.

Индексы должны идти по порядку с шагом 1. Если порядок не был соблюден, то во время расчета, блок попробует преобразовать индексы в нужный порядок (при большом количестве слоев может занять некоторое время).

Дублирование индексов допускается, однако возможна некорректная работа карты (при обнаружении дубликатов будет выведено предупреждение).

Обратная совместимость со старым типом иерархии

В блоке доступна возможность использовать старый тип иерархии, основанный на поле groupName.

Включение режима обратной совместимости

Для перехода в режим обратной совместимости необходимо, чтобы один или несколько слоев карты имели атрибут groupName и не имели атрибута dictMapping. В таком случае,

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

иерархия будет работать в режиме обратной совместимости, о чем будет выведена информация при расчете блока.

Важно: z-индексы не доступны в режиме обратной совместимости.

При работе в режиме обратной совместимости все элементы из входа "Дерево папок для слоев" являются папками, а слои добавляются отдельно, в зависимости от пути, указанного в поле groupName.

Поле groupName состоит из набора наименований, разделенных "|". Наименования берутся из поля attrs.name справочников, пришедших на вход "Дерево папок для слоев"

Пример поля:

```
"groupName": "Пермь|Школы"
```

Поиск папки будет просиходить вначале по первому наименованию (Пермь), после чего в его дочерних элементах начнется поиск элемента с наименованием Школы.

Если такой элемент будет найден, то слой будет записан в его дочерние элементы.

6.1.3.1.8.5 Размещение заранее подготовленных объектов на карте

В левой панели карты есть раздел «Объекты», где можно задать список объектов, которые пользователь может размещать на карте, наполняя тот или иной слой.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

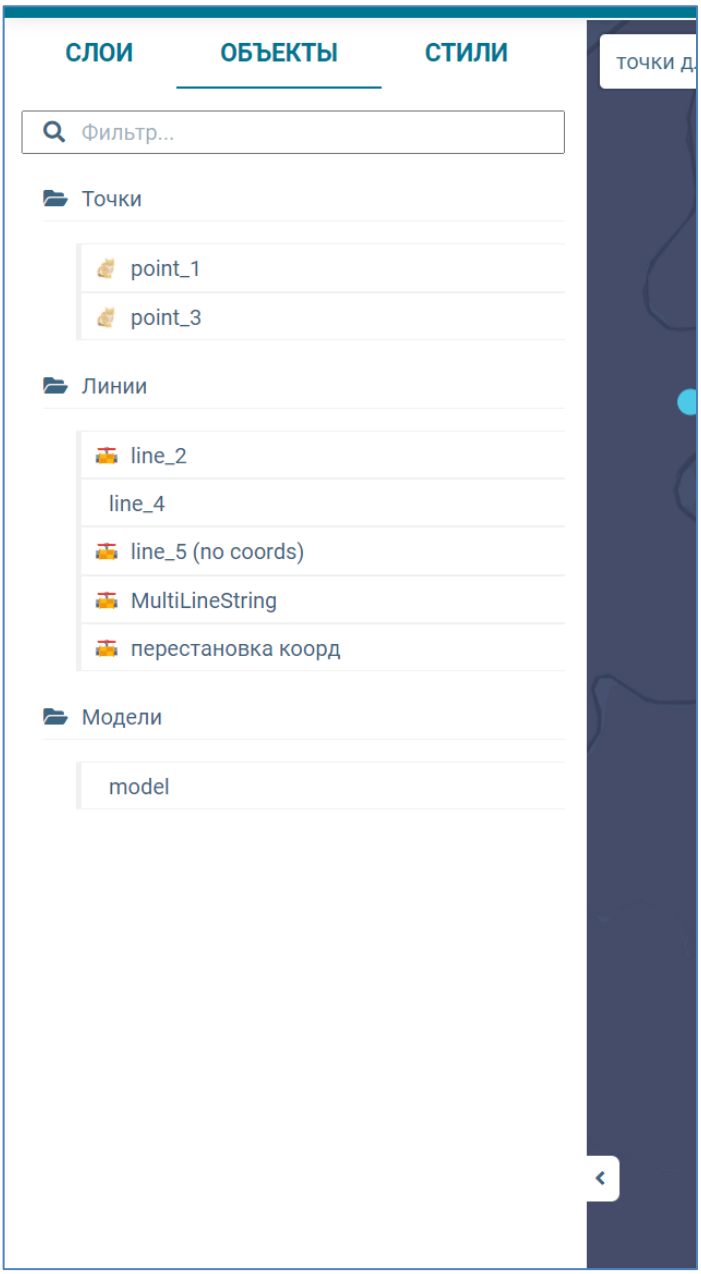


Рисунок 142 – Пример раздела «Объекты»

Для того чтобы добавить в визуализатор объекты для размещения, необходимо во вход блока «Объекты для размещения на карте» подключить справочник с описанием этих объектов. Для корректной работы в справочнике должны быть обязательные атрибуты:

1. **coords** – координаты. Может быть заполнен или оставлен пустым. При размещении объекта, если это точка, в это поле объекта будут прописаны координаты. Если объект линия или полигон – координаты в этом поле будут использованы для размещения объекта той же формы, которая задана в координатах, но в месте клика пользователя.
2. **source –** обязательное, поле в котором указывается **идентификатор** слоя, на который необходимо разместить объект.
3. **geometry –** обязательное, тип геометрии размещаемого объекта
    a. Point – точечный объект;
    b. LineString – линия;

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

c. Polygon – полигон.

4. **groupName** – имя группы для отображения иконок в списке. Если оставить пустым – иконки будут сгруппированы в группе «Неотсортированные».
5. **icon** – имя иконки для отображения объекта. Для корректного отображения иконки в блоке карта должно быть заполнено поле «Пользовательские иконки».

Механизм размещения объектов предполагает обязательное наличие поля **source.** В этом поле указывается идентификатор слоя, в который будут сохранены добавленные объекты. При размещении объекта, перетаскивая его с помощью Drag&Drop с панели на карту, будет включено редактирование того слоя, который указан в **source.** Затем можно будет работать с ним как с любым другим слоем в режиме редактирования.

Важно! Для корректной работы, слой, на который размещаются объекты, должен иметь возможность сохранения.

Если у размещаемого объекта указаны координаты, то после перетаскивания объекта на карте он будет иметь геометрию, соответствующую порядку указанных точек и расстоянию между ними, отстроенную относительно точки размещения объекта на карте.

Если же координаты размещаемого объекта не указаны, то при попытке размещения объекта на карте будет включен режим редактирования слоя, указанного в source, и точка размещения станет первой точкой, относительно которой пользователю будет дана возможность построить геометрию добавляемого на карту объекта.

6.1.3.1.8.6 Подложки карты

В левой панели карты есть раздел «Стили» в котором можно выбрать для текущей карты стиль отображения, заданный в списке «Глобальные стили карты».

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

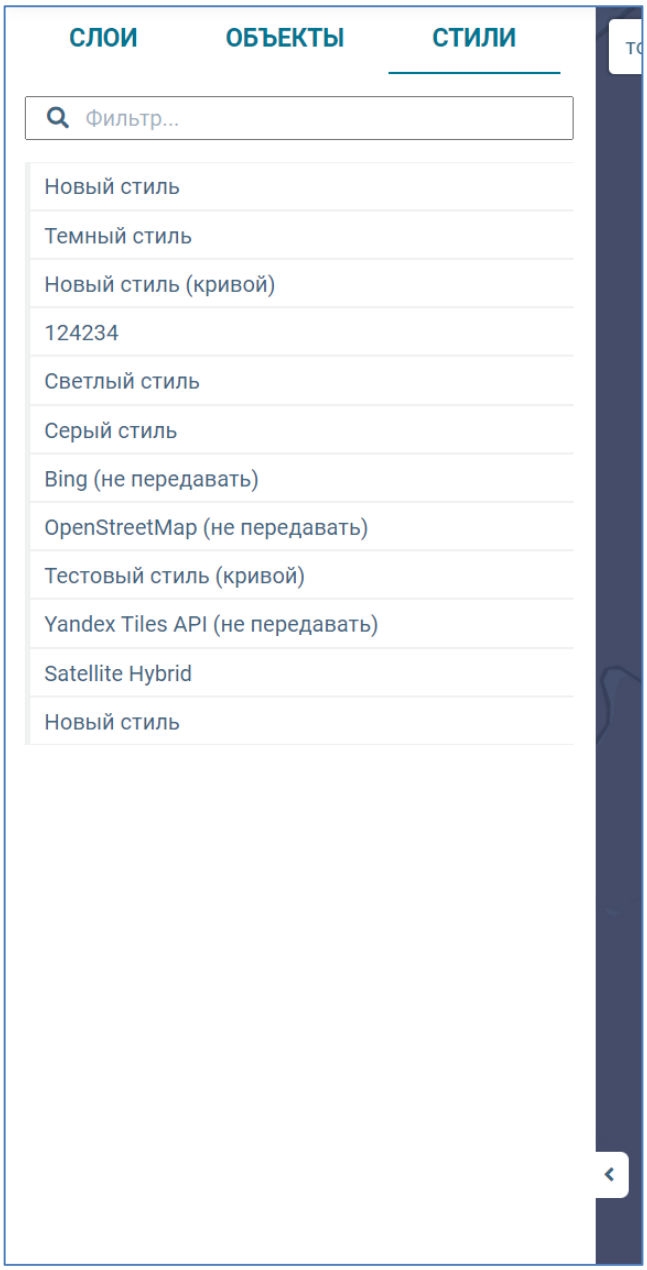


Рисунок 143 – Пример раздела «Стили»

Важно!

Выбранный стиль применяется только в рамках текущего визуализатора. Выбранный стиль не сохраняется в визуализаторе.

Для выбора другого стиля необходимо сделать это в блоке, выбрав стиль из списка Глобальных стилей.

Если в блоке выбран глобальный стиль – он будет выбран по умолчанию при открытии панели.

Если в блоке стиль задан вручную – на панели стилей он будет отображаться первым в списке – «Стиль по умолчанию».

Для обеспечения работы с внешним источником данных для слоя может понадобится передать в адресе источника какие-либо заголовки или параметры входа. Для этого предусмотрены следующие дополнительные параметры:

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

host - строка, с которой начинается url, необходима для идентификации источника в списке источников для получения оттуда набора заголовков или параметров входа. В случае нескольких источников, имеющих host, значение этого параметра в разных источниках должно отличаться друг от друга и не должны являться подстрокой друг для друга, иначе источник может быть найден неверно и соответственно в запрос будут переданы заголовки другого источника.

headers - json-объект, содержащий пары ключ - значение, для передачи в качестве заголовков запроса

credentials - дополнительное поле для передачи параметров входа(если необходимо)

Для растровых слоев или других, где необходимо передавать перечень адресов для получения данных, параметры немного видоизменяются:

- hosts - массив строк, являющихся началом для адресов, передаваемых для получения данных
- headers - массив json-объектов для передачи в качестве заголовков, где индекс в массиве заголовков для передачи соответствует индексу хоста, по которому эту заголовки требуется передавать
- credentials - массив параметров входа, также соответствующих хостам по индексу в массиве

Данный механизм так же применяется в Блок "Слой данных для 3D карты".

6.1.3.1.8.7 Редактирование объектов слоя

Для перехода в режим редактирования слоя необходимо нажать на кнопку «Редактировать» или перетащить объект для размещения на карту, что приведет к включению редактирование слоя, на котором он размещается.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

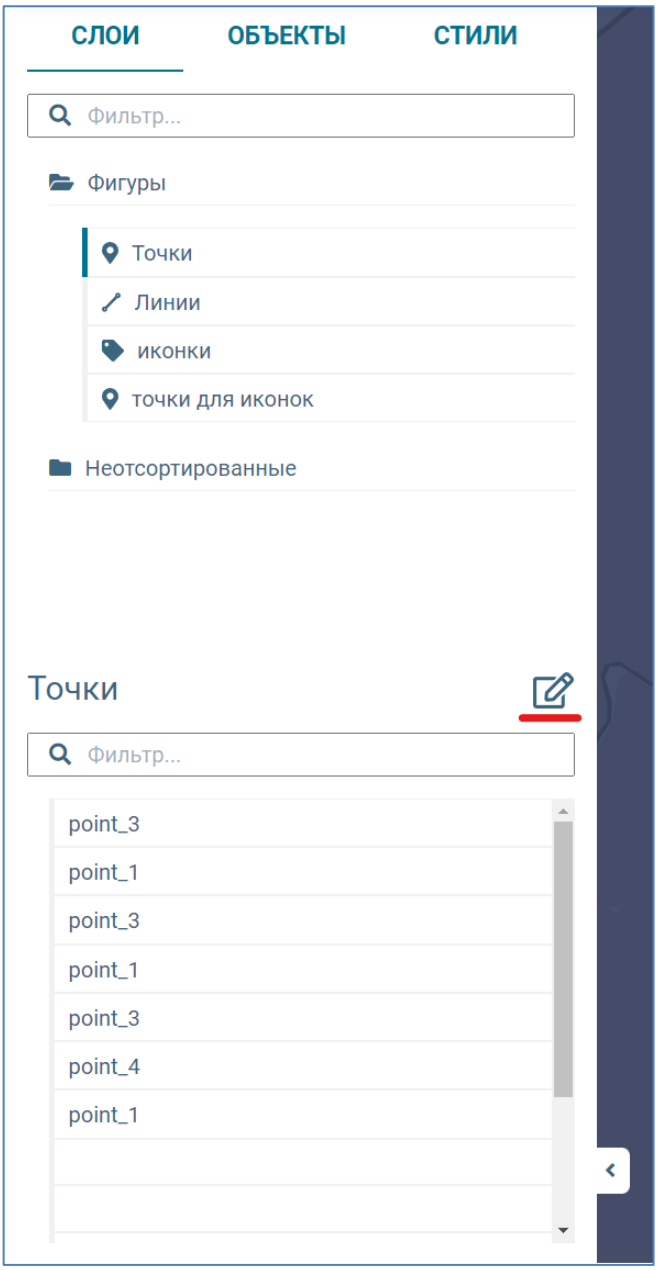


Рисунок 144 – Пример кнопки «Редактировать» объектов слоя

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

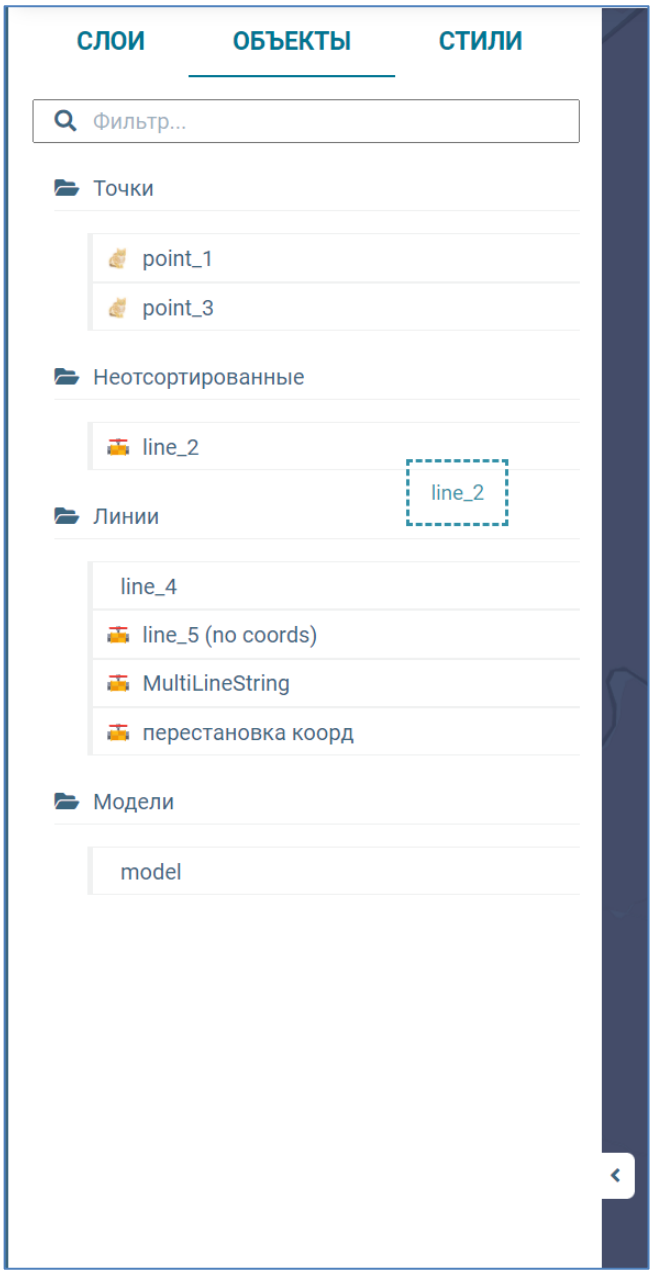


Рисунок 145 – Пример режима редактирования слоя

Важно!

Слой должен иметь возможность сохранения, для сохранения результатов редактирования. Так же редактирование не должно быть запрещено с помощью флага **editable** в описании слоя.

Редактирование происходит для каждого объекта по отдельности, таким образом, несколько пользователей могут редактировать один и тот же слой.

Однако, если один из пользователей удалит объект редактирование, то будет выдано предупреждение, и данные по этому объекту не будут сохранены, т.к. он уже был удален.

Для перевода объекта, который уже расположен на слое, в режим редактирования необходимо нажать на него левой клавишей мыши в режиме редактирования слоя.

В режиме редактирования слоя появляется дополнительная панель.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Рисунок 146 – Пример дополнительной панели в режиме редактирования

Добавить новый объект – добавление нового объекта на карту.

1. Добавление точки – достаточно нажать на карту и в месте нажатия появится точка.
2. Линия – необходимо нарисовать линию, размещая ее с первой до последней точки. Для завершения рисования необходимо второй раз нажать на последнюю точку. Для добавления точки на ребро необходимо нажать на центр ребра (небольшая оранжевая точка) – это добавит новую точку.
3. Полигон – необходимо нарисовать полигон. Для завершения рисования необходимо нажать на первую точку замкнув полигон. В нем должно быть минимум три вершины. Для добавления точки на ребро необходимо нажать на центр ребра (небольшая оранжевая точка) – это добавит новую точку.

Рисунок 147 – Пример добавления нового объекта на карту

Редактировать GeoJSON слоя – редактирование слоя в виде нативного geojson. При нажатии показывает дополнительную панель, в которой можно увидеть geosjon слоя целиком со всеми объектами.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

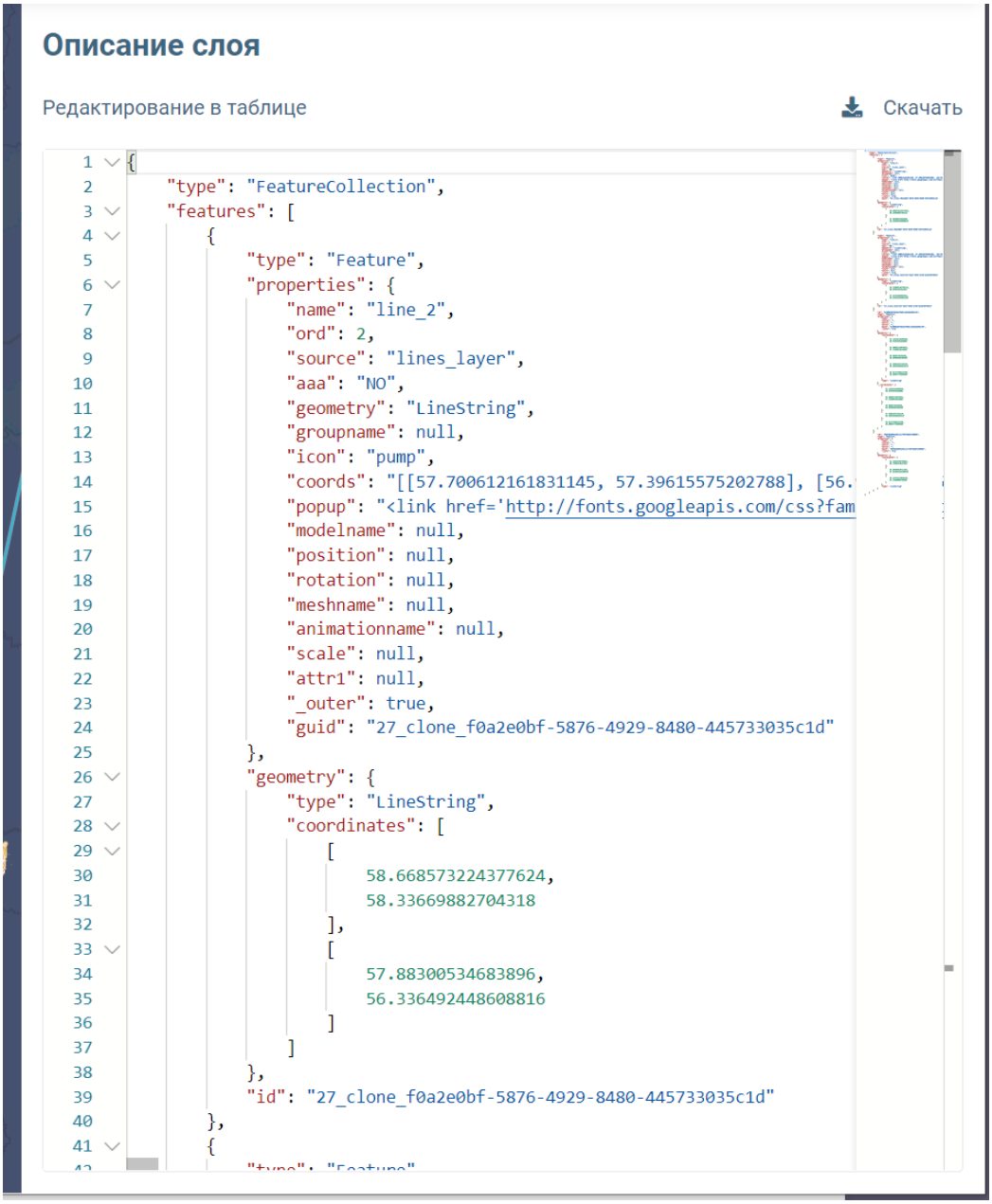


Рисунок 148 – Пример редактирования слоя в geojson

При нажатии на кнопку «**Скачать**» произойдет скачивание geosjon слоя.

Обратите внимание:

Редактирование слоя в виде нативного geojson перезапишет все изменения, сохраненные другими пользователями.

При попытке включить редактирование, так же будет выдано предупреждение:

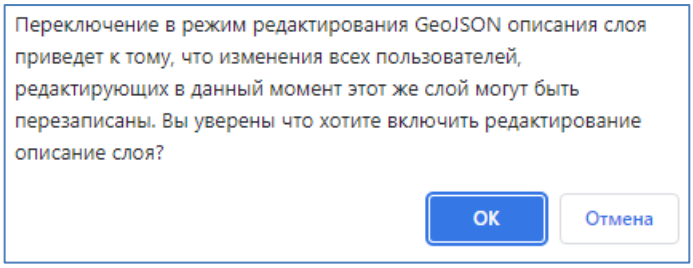


Рисунок 149 – Пример предупреждения

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.1.3.1.8.8 Редактирование объектов в виде таблицы

При нажатии «Редактирование в таблице» будет открыто новое окно, в котором все объекты редактируемого слоя представляются в виде таблицы и пользователь может менять их поля.

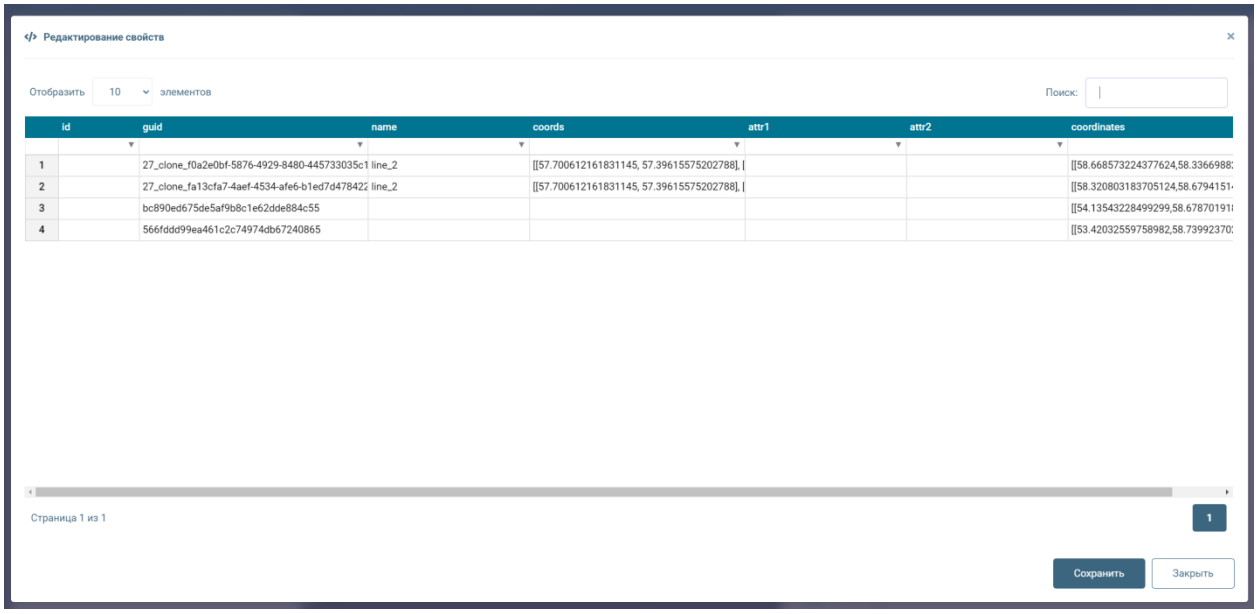


Рисунок 150 – Пример окна «Редактирование свойств»

6.1.3.1.8.9 Карточки объектов

При нажатии на объект, если у него задана карточка (атрибут popup) – она будет открыта. При этом, если на месте нажатия расположены несколько объектов – пользователю сможет переключаться между ними.

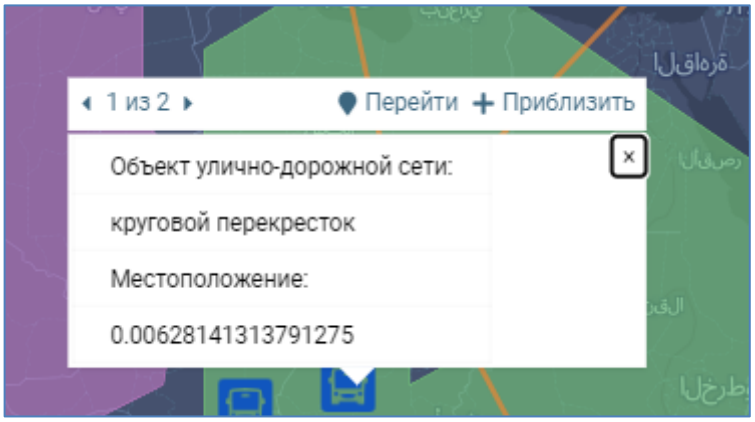


Рисунок 151 – Пример карточки объектов

Карточка поддерживает позиционирование на объекте и приближение к нему.

Важно!

При маленьком зуме одна карточка может открываться для большого количества объектов. При приближении объект может перестать находиться под карточкой, так как приближение изменяет уровень зума карты.

При переключении между объектами карта будет позиционироваться на выбранном.

В режиме редактирования карточка меняется и представляет собой набор атрибутов объекта, которые можно изменять.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

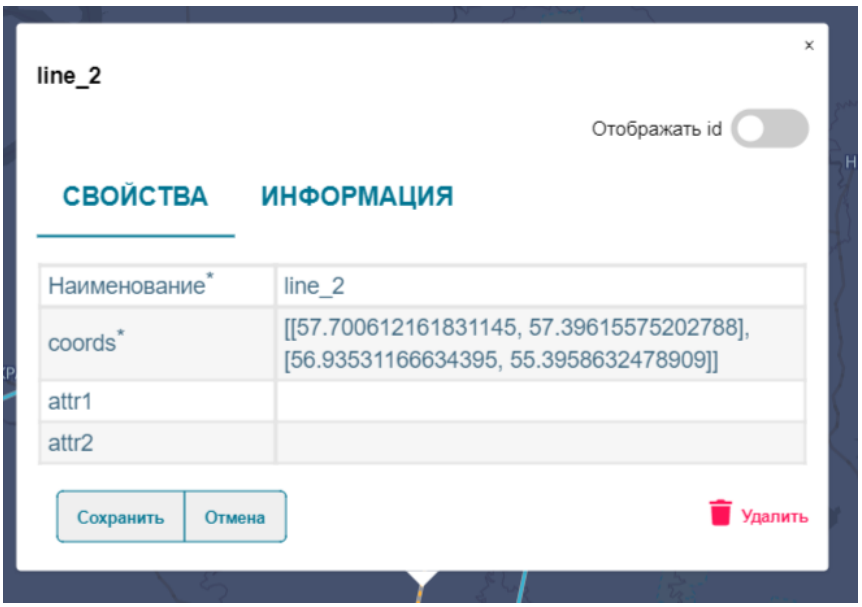


Рисунок 152 – Пример режима редактирования карточки

При этом, в карточке доступен раздел с информацией, в которой можно увидеть пространственные характеристики объекта – его размеры и координаты.

6.1.3.1.8.10 Анимация

Если у объекта есть атрибут **animate=true** и он был передан на карту – изменение положения объекта будет произведено с анимацией движения.

Если анимация работает медленно в браузере Google Chrome флаг "GPU rasterization" в chrome://flags должен быть включен.

6.1.3.1.8.11 Отображение координат курсора

При включении настройки «Отображать координаты курсора» и выборе позиции на карту будут выведены координаты курсора мыши.

Рисунок 153 – Пример отображения координат курсора мыши

6.1.3.1.8.12 Поиск

При нажатии на кнопку поиска, если настройка включена в блоке, открывается строка поиска. При вводе информации и нажатии на клавишу **Enter** будет произведен поиск по всем объектам всех слоев, при этом, он будет проводиться не только по наименованиям, но и по всем атрибутам, кроме тех, что исключены из поиска настройкой «Исключить из поиска». В случае использования геокодера поиск будет происходить с подстановкой адресов через обращение к геокодеру. Переход на карте будет осуществляться по координатам, переданным геокодером.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

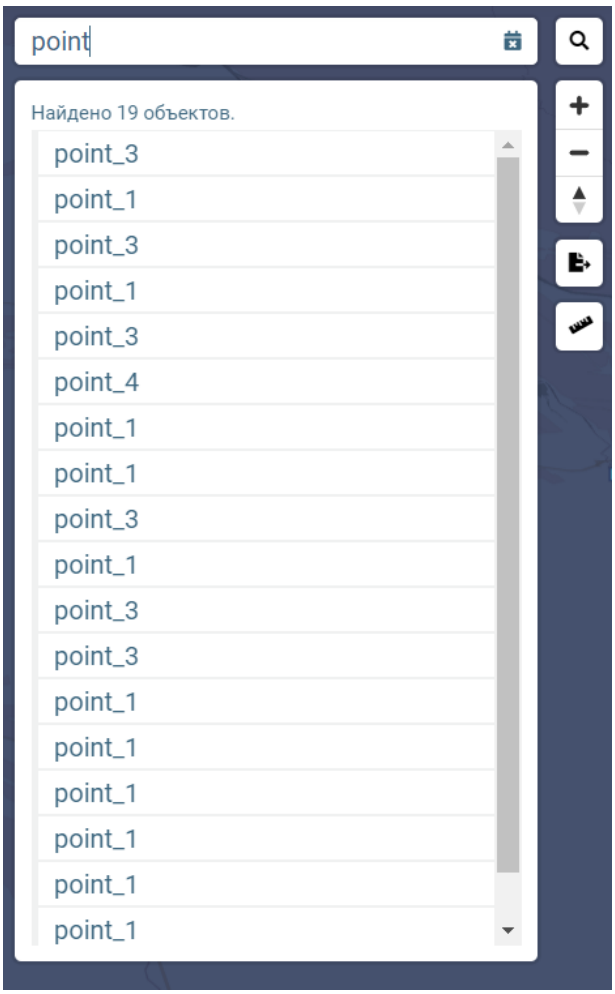


Рисунок 154 – Пример поиска по атрибутам

Особенность работы Яндекс.Геокодера

Яндекс.Геокодер возвращает схожие результаты и при поисковых подсказках, и при полноценном поиске. У API сервиса один метод под поиск и подсказки. Результаты там и там будут одинаковыми.

Особенности работы поиска

При поиске на карте учитываются только объекты в пределах области видимости карты

При выборе встроенного геокодера для поиска поддерживается поиск с учетом опечато

Особенность работы геокодера DaData

При поиске на карте зона видимости учитывается только при области видимости с радиусом до 100км. При увеличении масштаба поиск не ограничивается текущей областью видимости. Ограничение связано с особенностями API сервиса DaData

6.1.3.1.8.13 Инструменты измерения на карте

При нажатии на кнопку «Панель измерений», которая доступна, если включена соответствующая настройка, открывается «Панель измерений». Доступны 3 варианта.

Линейный объект

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

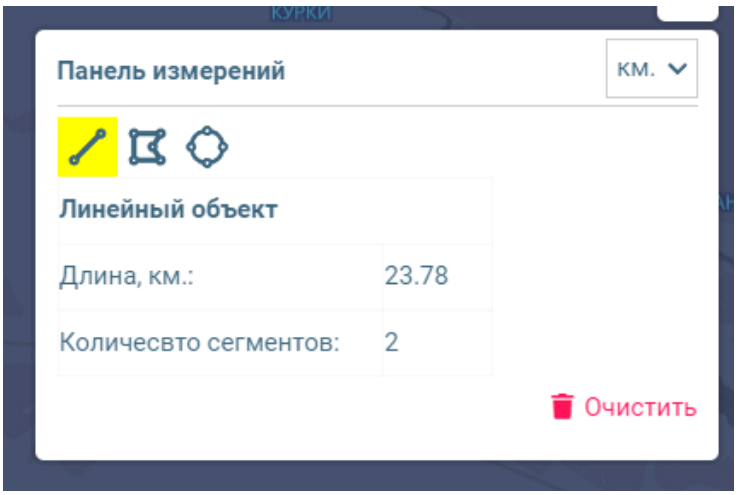


Рисунок 155 – Пример окна «Панель измерений» линейного объекта

1. Единицы измерения: километры и метры.
2. При рисовании измеряется длина каждого сегмента на карте.

Рисунок 156 – Пример отображения длины каждого сегмента на карте

3. В панели измеряется общая длина в единицах измерения.
4. В панели измеряется количество сегментов линии.

Площадной объект

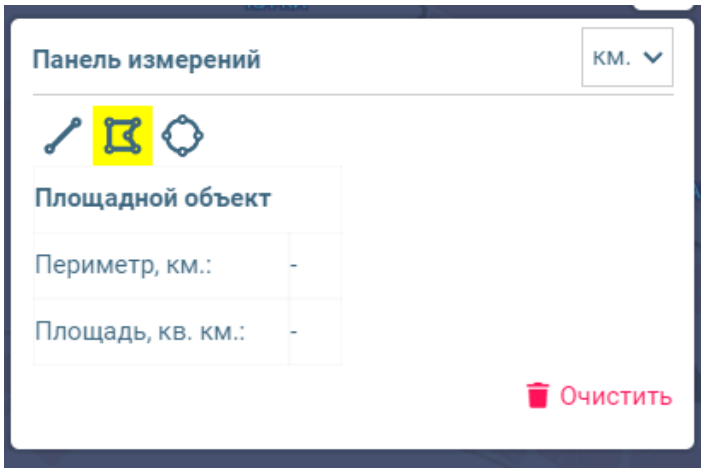


Рисунок 157 – Пример окна «Панель измерений» площадного объекта

1. Единицы измерения: километры и метры (т.к. объект площадной – длина линейная, площадь в квадратных единицах).
2. При рисовании измеряется длина каждого сегмента периметра объекта и его площадь.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Рисунок 158 – Пример отображения периметра объекта и его площади

3. Для завершения рисования надо нажать клавишу **Enter.**
4. В панели измеряется периметр объекта и его площадь.

Радиус окружности

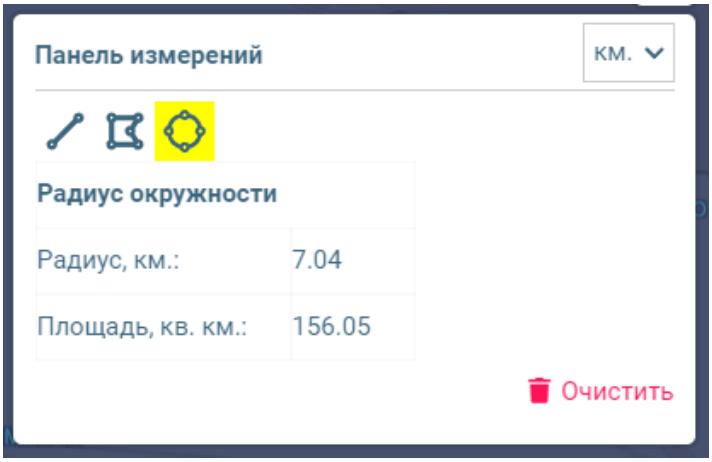


Рисунок 159 – Пример окна «Панель измерений» радиуса окружности

1. Единицы измерения: километры и метры.
2. При рисовании необходимо нажать один раз и «растянуть» окружность до нужного радиуса. При этом второе нажатие завершит рисование.
3. На карте и в панели измеряется длина радиуса и площадь окружности.

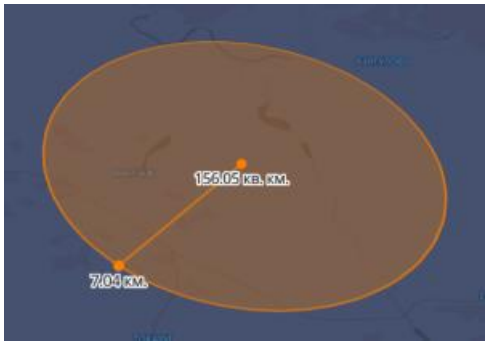


Рисунок 160 – Пример отображения радиуса окружности

Важно! При рисовании окружности нужно учитывать искажения проекции карты.

6.1.3.1.8.14 Геосервер

Работа со слоями типа «Внешний источник данных (геосервер)» и «Геосервер Martin» отличается от других слоев.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

1. Слои получаются в виде векторных данных при запросе их картой с сервера.
2. Слои не редактируемые и не содержат списка объектов.
3. Слои не поддерживают поиск
4. Слои показывают карточки, если в данных объекта есть поле **popup.**
5. При нажатии на такой слой в выход блока «Событие нажатия на объекте слоя» передаются все объекты со слоя геосервера, на которые «попало» нажатие.

6.1.3.1.9 Печать

Для корректной печати в настройках необходимо включить параметр "Фон".

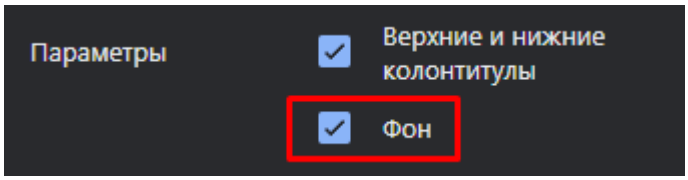


Рисунок 161 – Пример отображения радиуса окружности

6.1.3.1.10 Настройки визуализатора

Общий список настроек приведен в п. 4.2.8 «Настройки визуализатора».

Дополнительными настройками является:

1. Возможность сохранения текущего положения карты при сохранении пресета.

6.1.3.1.11 Дополнительные ссылки

Документация по стилям MapBox Layer Style

Документация по слоям библиотеки deck.gl

Визуальный редактор стилей Maputnik

6.1.3.1.12 Работа с 3D-картой и ее слоями

Работа с картой включает в себя редактирование настроек блоков графа «3D-карта», «Слой данный для 3D карты», возможность управления стилями и геоподложкой карты, возможность осуществлять поиск на карте и работу с объектами на карте.

Для **редактирования блоков создания карты (слоя карты)** необходимо:

1. Открыть настройки блока «Слой данных для 3D карты».
2. При необходимости изменить параметры настройки типа слоя и типа геометрии слоя карты.
3. При необходимости изменить параметры настройки в части описания слоя в формате json и описание атрибутов для передачи на слой в формате json.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

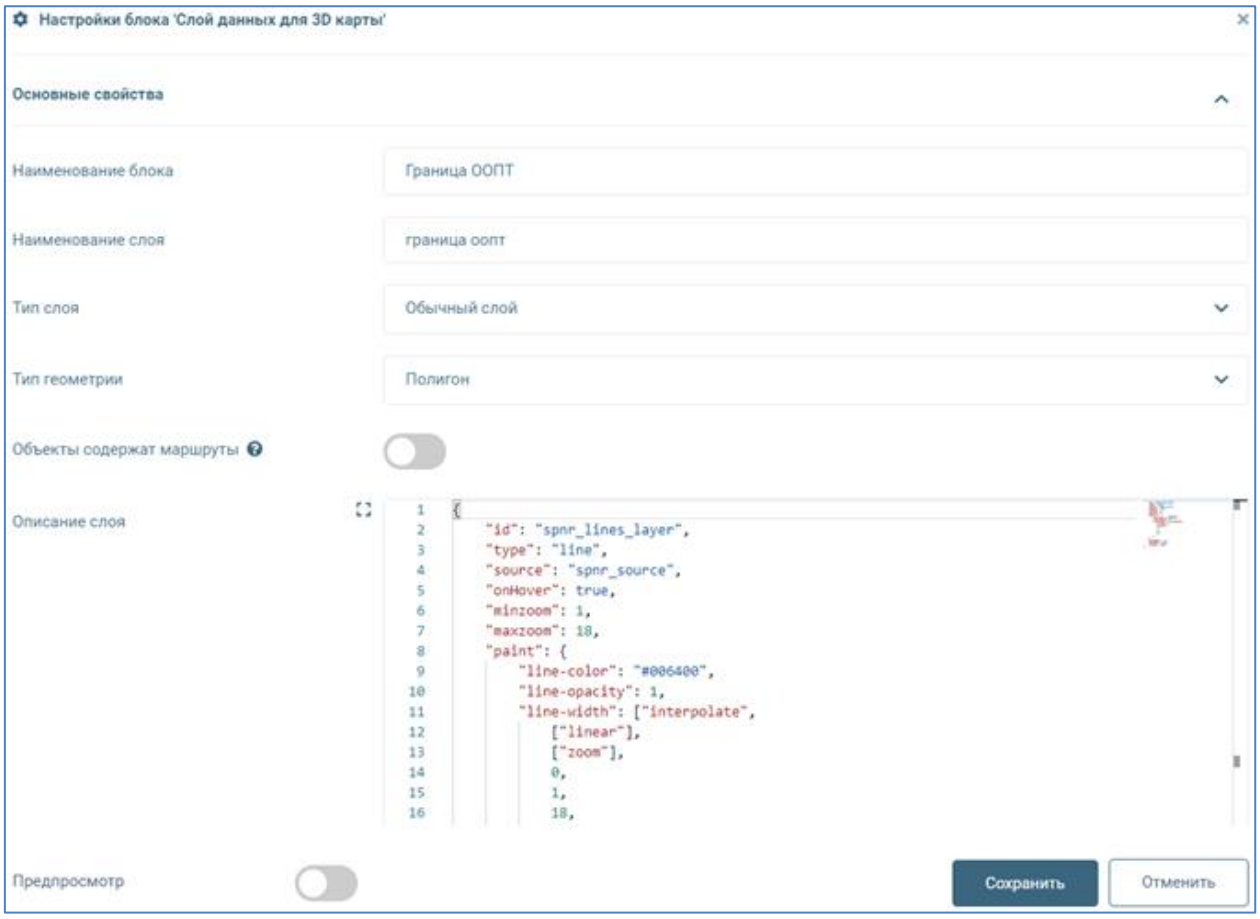


Рисунок 162 – Пример настройки блока «Слой данных для 3D карты»

Для **редактирования настроек блока тепловой карты** необходимо:

1. Открыть настройки блока «Слой данных для 3D карты». Если в описании слоя указан тип «heatmap» – будут открыты настройки тепловой карты.
2. При необходимости изменить параметры настройки в части описания слоя в формате json и описание атрибутов для передачи на слой в формате json.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

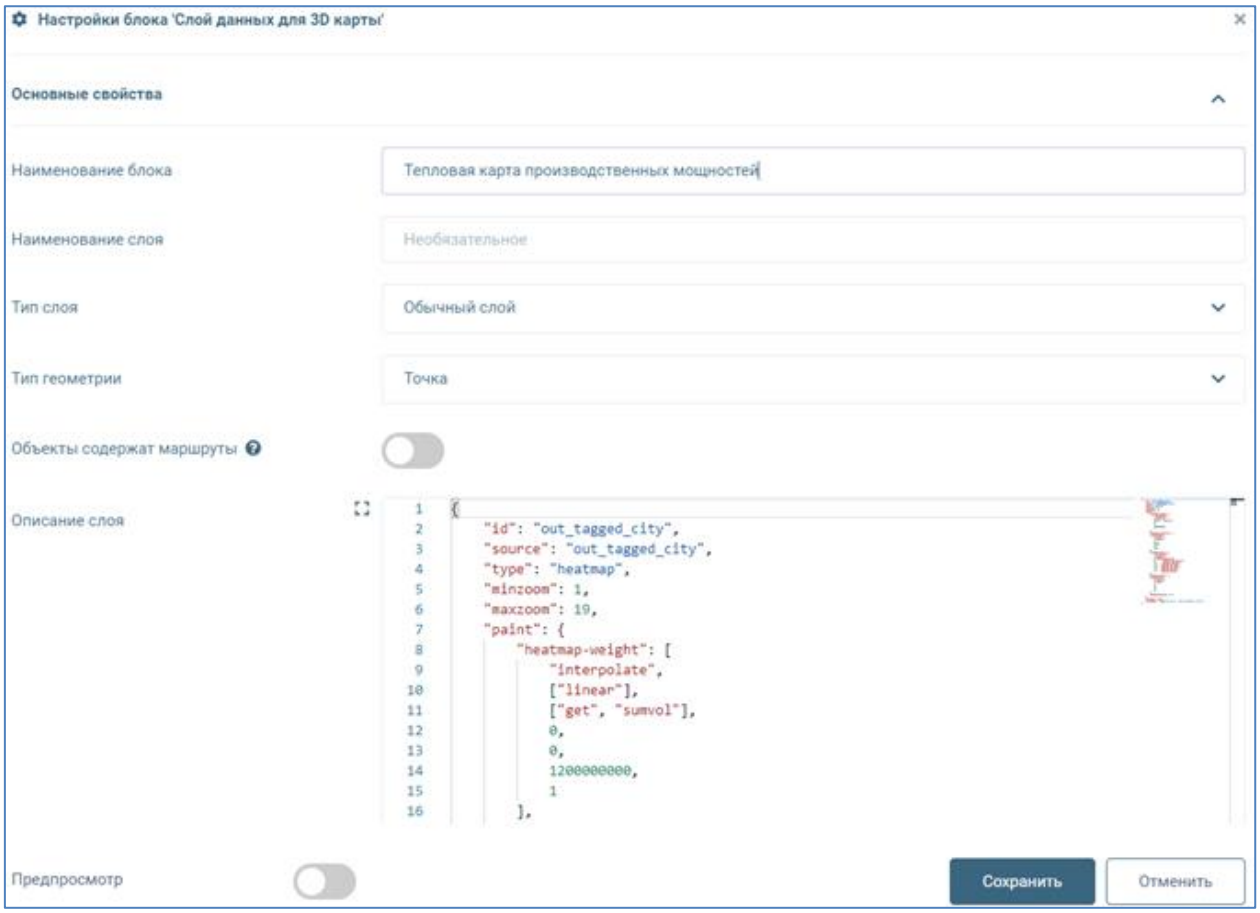


Рисунок 163 – Пример настройки блока «Слой данных для 3D карты»

3. Для предпросмотра слоя карты – тепловая карта в режиме работы с графом включить «Предпросмотр» в настройках блока «3D-карта».

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

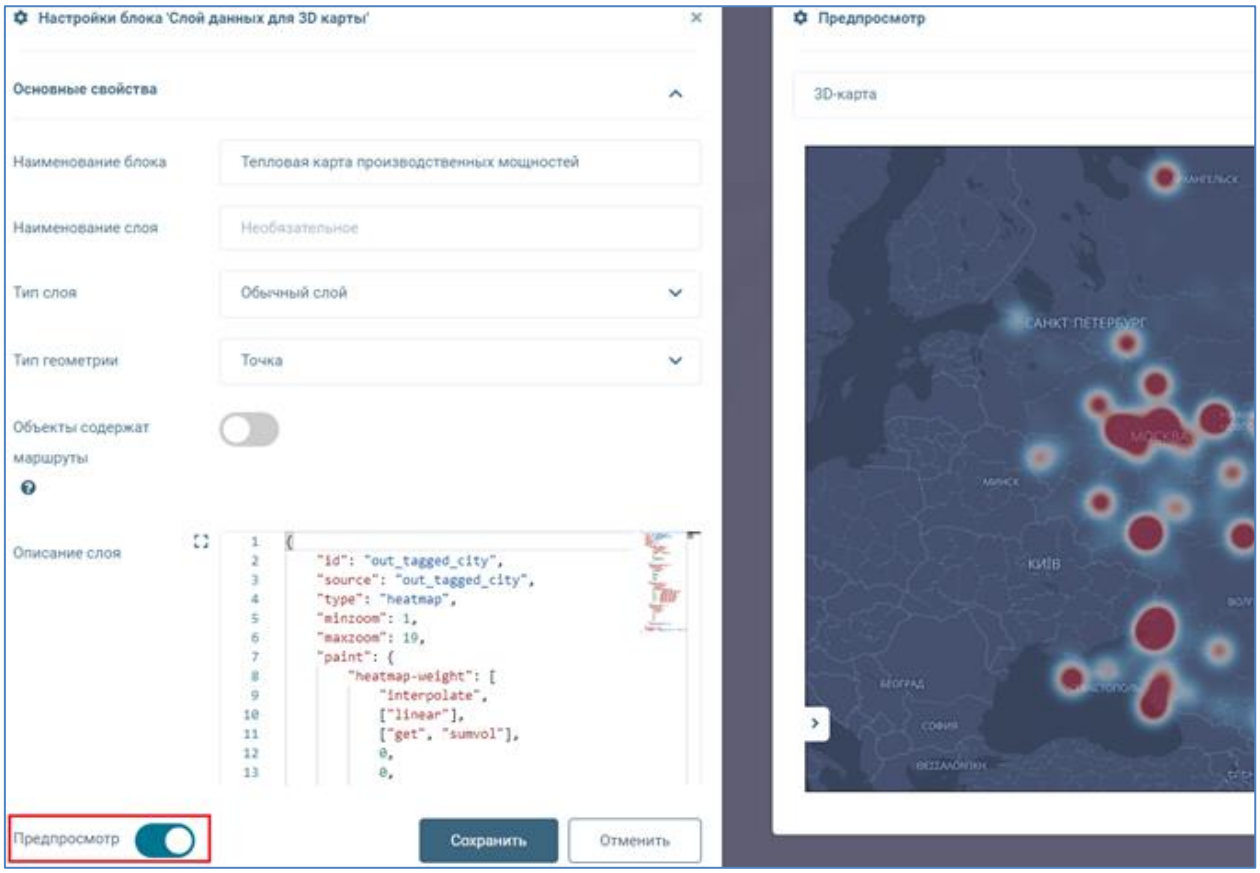


Рисунок 164 – Пример предпросмотра слоя карты – тепловая карта

4. Открыть блок «3D-карта».
5. При необходимости изменить параметры настройки стиля карты в формате json, указание центра карты, указание начального масштаба, разрешение масштабирования карты (минимальный, максимальный масштабы, угол наклона камеры, азимут), сглаживание, настройка карточек объектов

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

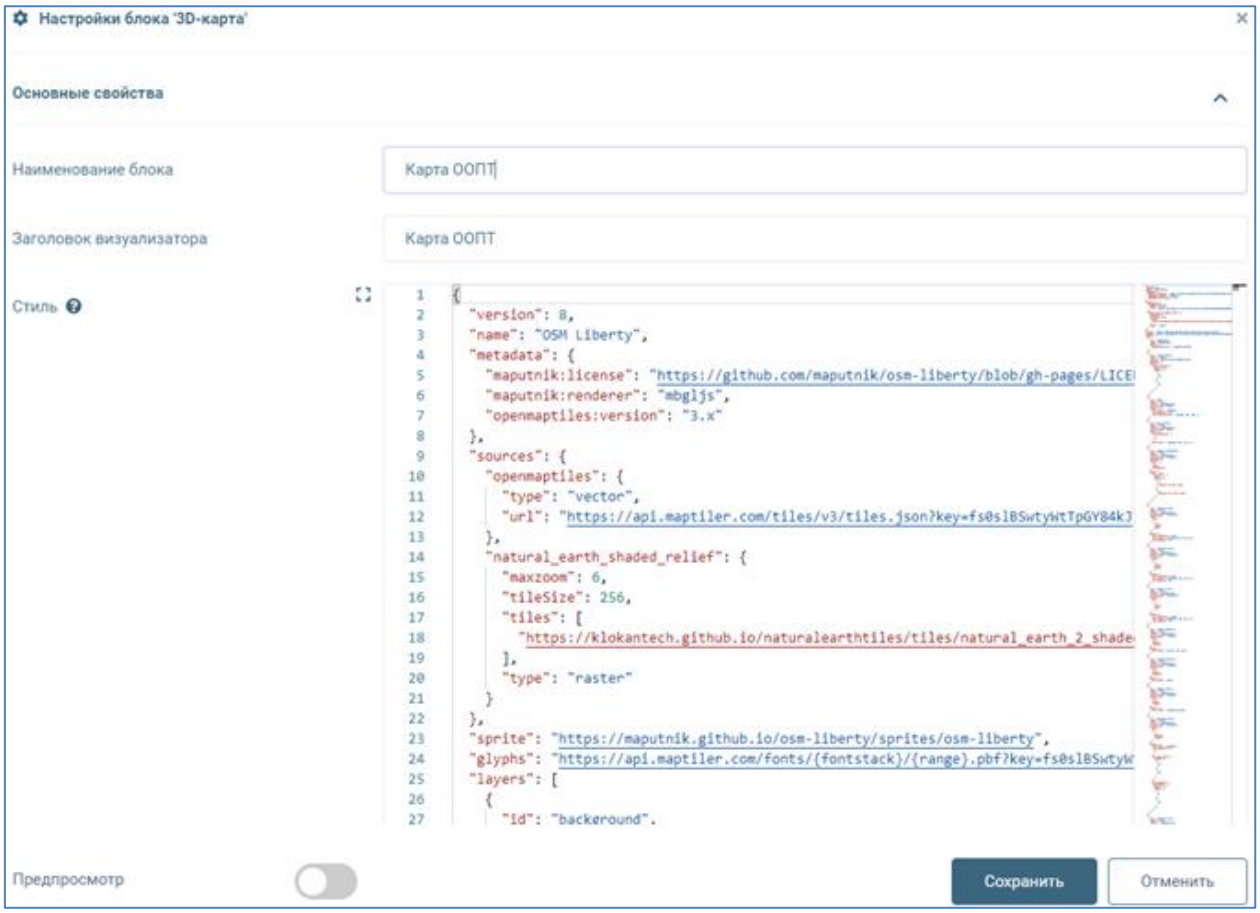


Рисунок 165 – Пример настройки стиля карты в формате json

6. Для предпросмотра карты (слоя карты) в режиме работы с графом включить «Предпросмотр» в настройках блока «3D-карта».

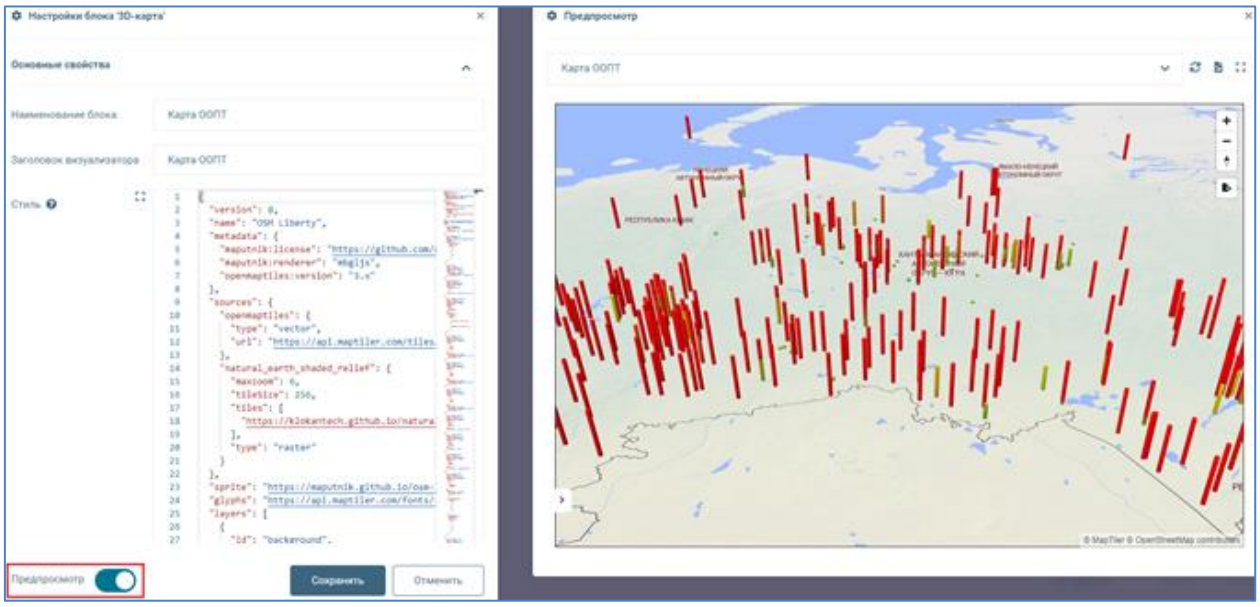


Рисунок 166 – Пример предпросмотра слоя карты

При работе с картой присутствует возможность добавлять и изменять стили подложки карты. Для **добавления нового стиля карты** необходимо:

1. Перейти в выпадающее меню в правом верхнем углу рабочей области, нажав на кнопку, соответствующую имени Пользователя.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

2. Перейти в раздел «Стили карты», нажав на соответствующую кнопку.

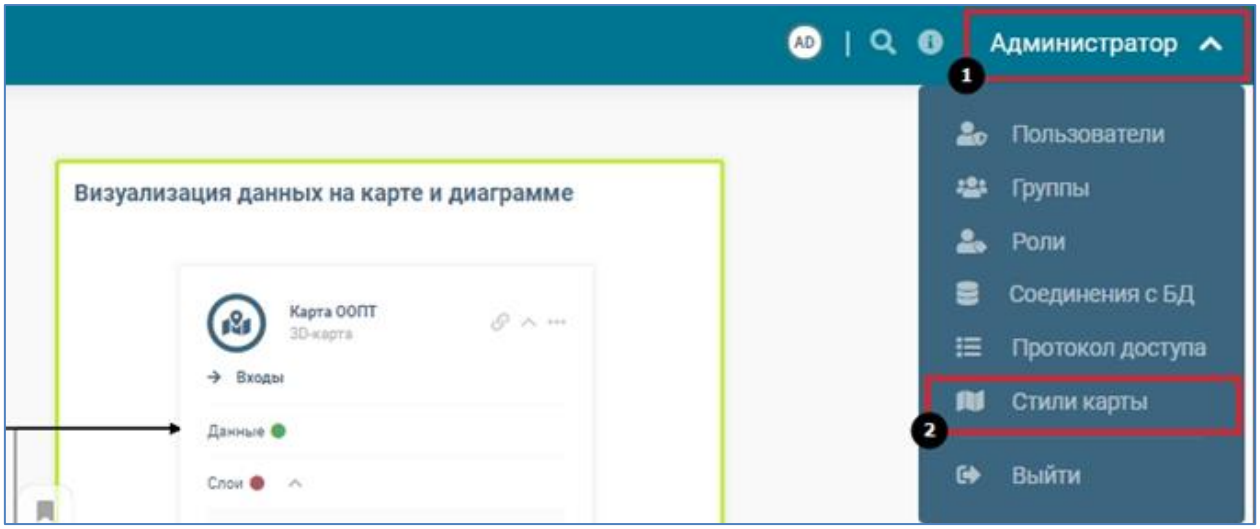


Рисунок 167 – Пример раздела «Стили карты»

3. Создать новый стиль карты, нажав на кнопку + в меню раздела «Стили карты».

Рисунок 168 – Пример добавления нового стиля карты

4. Добавить имя и описание нового стиля карты, изменить параметры настройки стиля карты в формате json и сохранить изменения.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

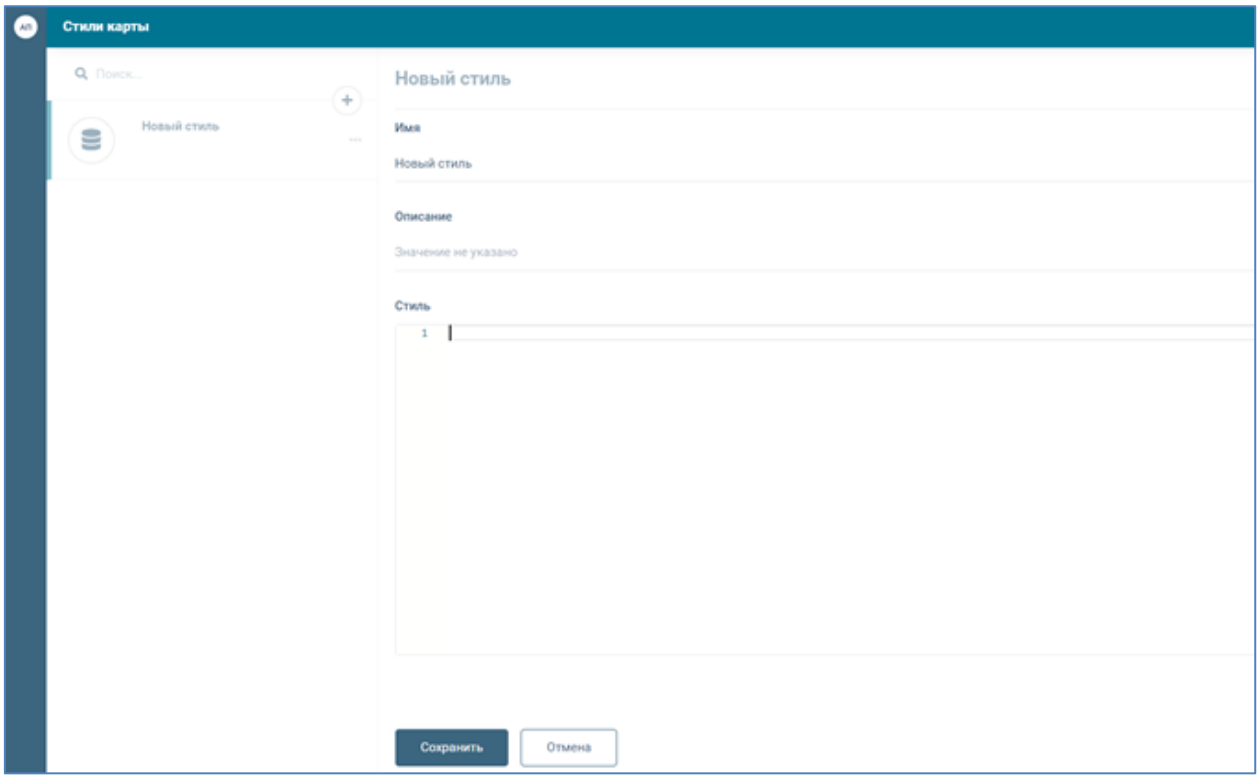


Рисунок 169 – Пример настроек стиля карты

После добавления и сохранения всех настроек новый стиль карты появится в списке стилей карты и отобразится информационное сообщение об успешном создании нового элемента в правом верхнем углу рабочей обрасти.

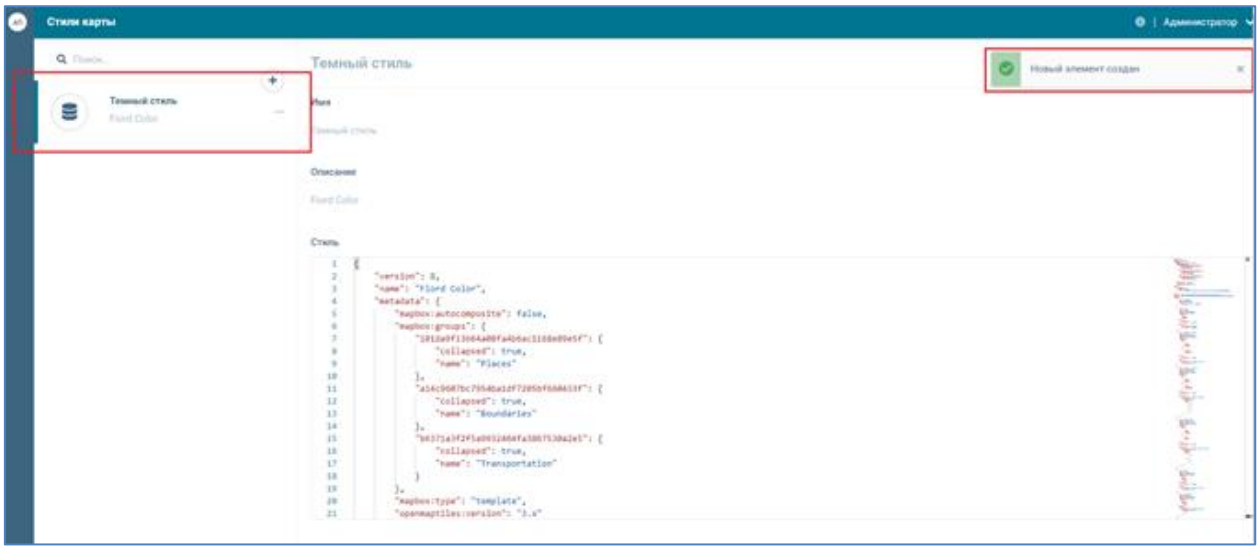


Рисунок 170 – Пример создания стиля карты

7. После успешного добавления нового стиля карты в настройках блока «3D-карта»:
   - Изменить настройки стиля карты на «Глобальные стили».
   - Отключить флаг "Использовать стиль по умолчанию".
   - Выбрать свой стиль из списка.
   - Сохранить изменения.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

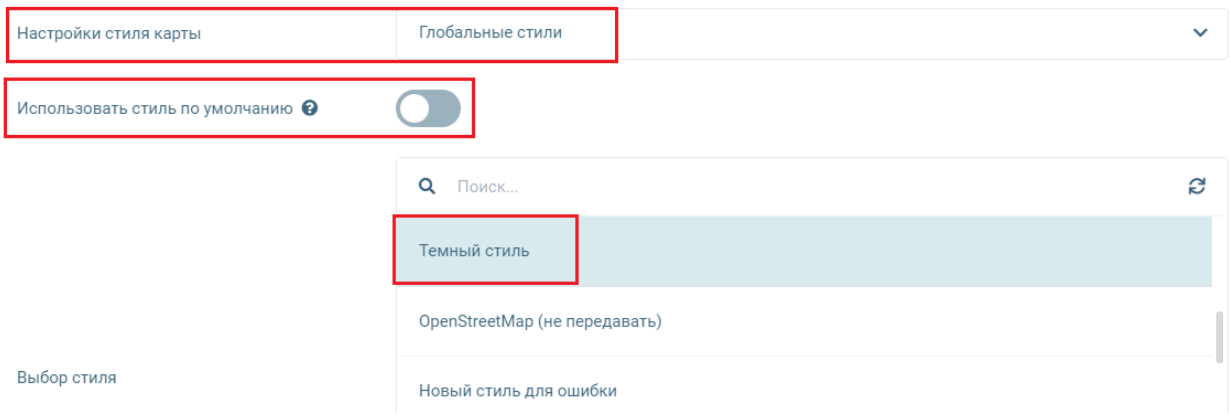


Рисунок 171 – Пример настройки стиля карты на «Глобальные стили»

Для **настройки поиска по карте** необходимо:

1. Открыть настройки блока «3D-карта».
2. Перейти в настройки карточки и включить «Поиск по карте», сохранить изменения.

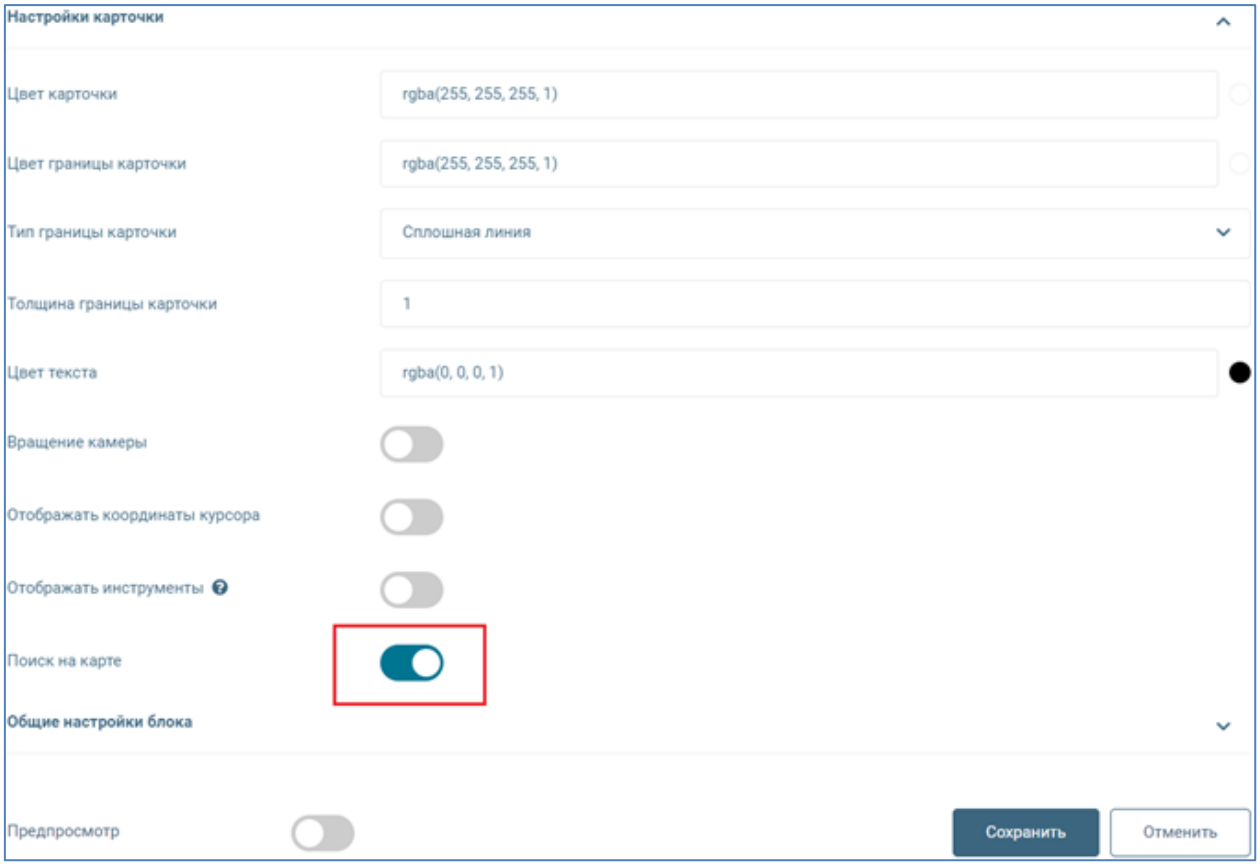


Рисунок 172 – Флаг «Поиск на карте»

После добавления поиска по карте в настройках блока «3D-карта» в пресете появляется возможность осуществления поиска по географическим координатам, названиям, географическим объектам, адресам, событиям, внесенным в БД.

Для **поиска объекта на карте** необходимо:

1. Открыть поле для ввода координат или наименования объекта, нажав на соответствующую кнопку в верхнем правом углу визуализатора карта.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

2. Ввести данные, по которым необходимо осуществить поиск на карте. Координаты объекта необходимо вводить через запятую, без пробела в соответствии со стандартом geojson.
3. Нажать на клавиатуре кнопку «Enter», после чего объект будет найден.

Рисунок 173 – Поиск по координатам объекта

К работе с объектами на карте относятся:

- Просмотр информации об объекте на карте.
- Добавление объектов на карте.
- Удаление объектов на карте.
- Измерение объектов на карте.
- Углубление в данные (Drill Down).

Для **работы с объектами на карте** необходимо перейти в режим пресета. Работа с объектами на карте возможна в режиме просмотра пресета.
Для отображения детальной информации объектов на карте необходимо нажать на выбранный объект. Если информация об объекте присутствует, на карте появится информационное окно.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

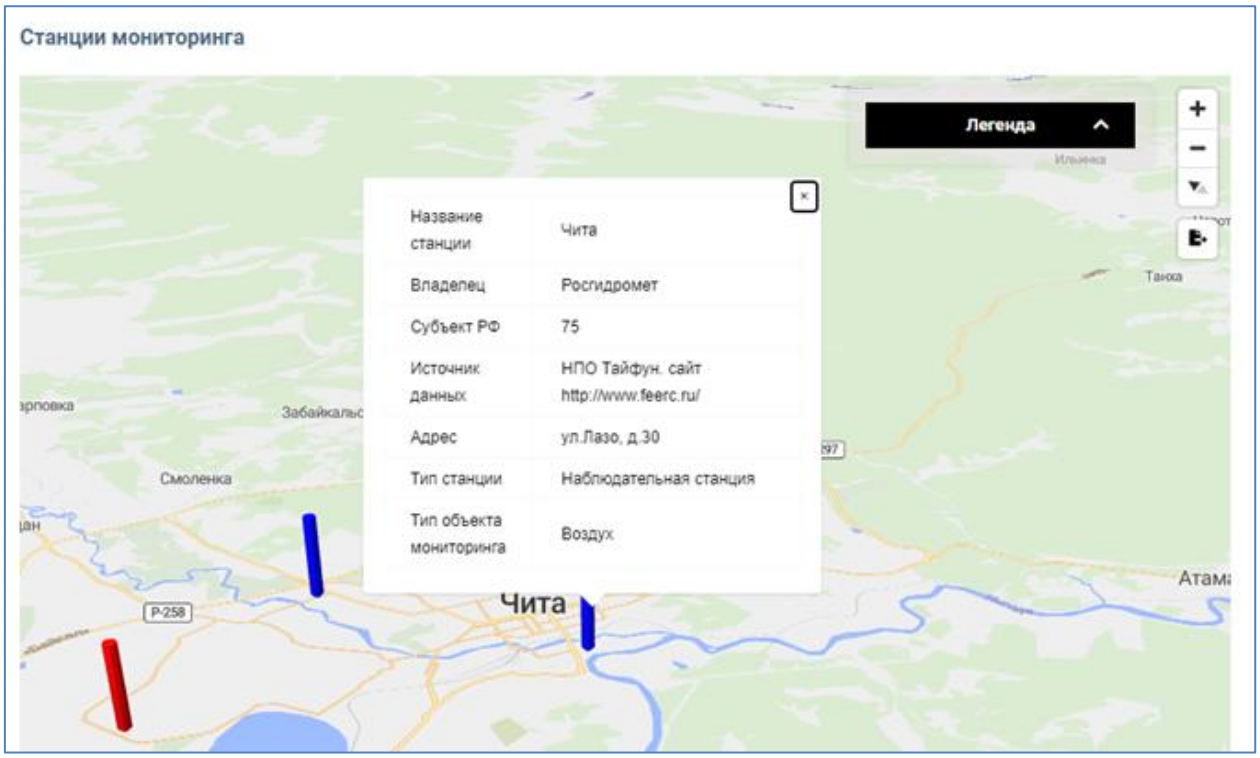


Рисунок 174 – Окно с данными объекта

Для добавления объектов на карте необходимо выполнить следующие действия:

1. Открыть боковое меню карты в режиме пресета

Рисунок 175 – Пример бокового меню карты в режиме пресета

2. Перейти в раздел «Объекты» бокового меню.
3. Выбрать объект из списка объектов и перетащить его на карту.
4. При необходимости отредактировать положение добавленного объекта.
5. Сохранить добавленный объект, нажав на соответствующую кнопку в верхней части карты.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

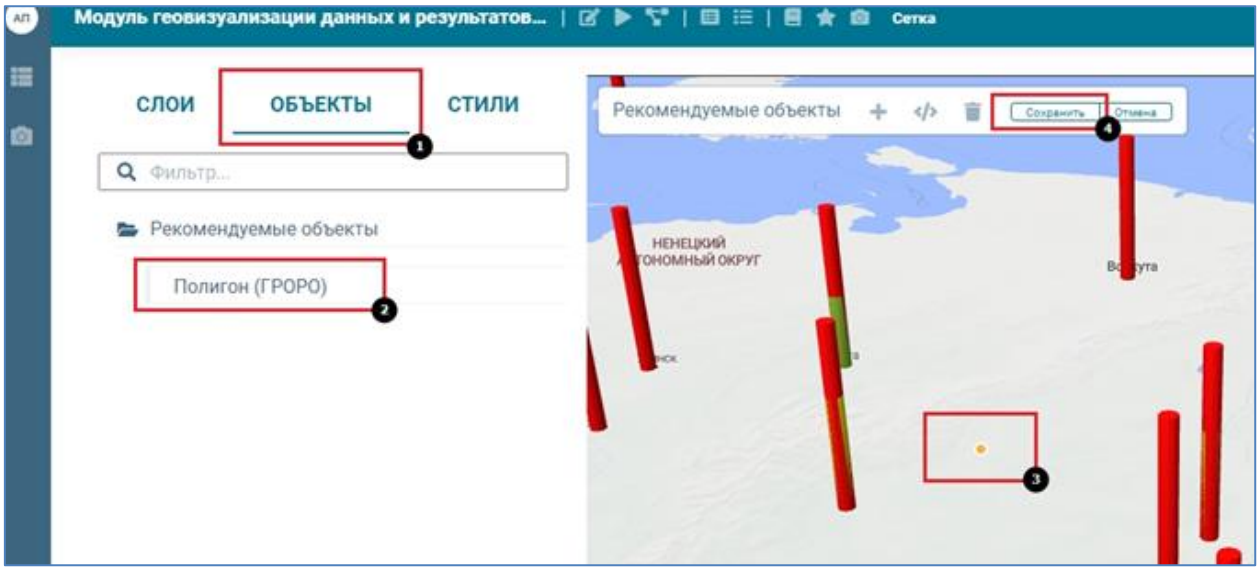


Рисунок 176 – Пример раздела «Объекты» бокового меню

После успешного добавления объект появится на карте.

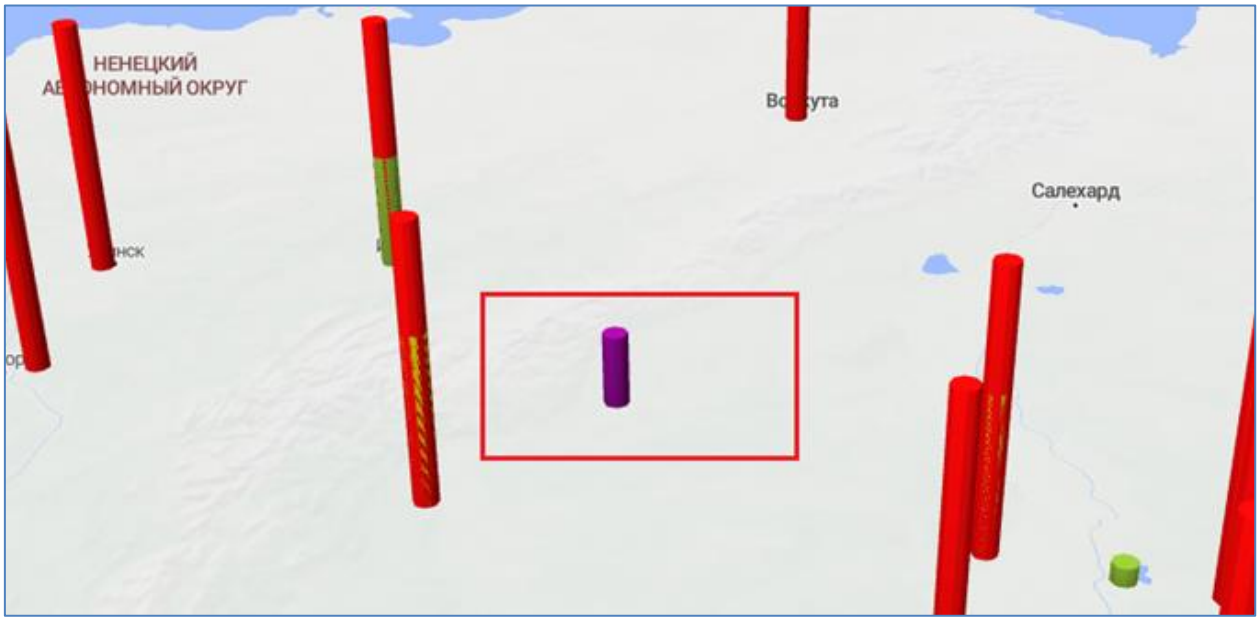


Рисунок 177 – Пример отображения добавленного объекта

Удалять с карты можно только добавленные объекты. Объекты, которые были заданы в блоках графа, удалить с карты нельзя. Для **удаления объекта с карты** необходимо выбрать следующие действия:

1. Перейти в раздел «Объекты» бокового меню.
2. Выбрать наименование объекта из списка объектов.
3. Выбрать объект для удаления объект на карте, нажать на него.
4. Нажать на кнопку удаления в верхней части карты.
5. Сохранить удаление объекта, нажав на соответствующую кнопку в верхней части карты.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

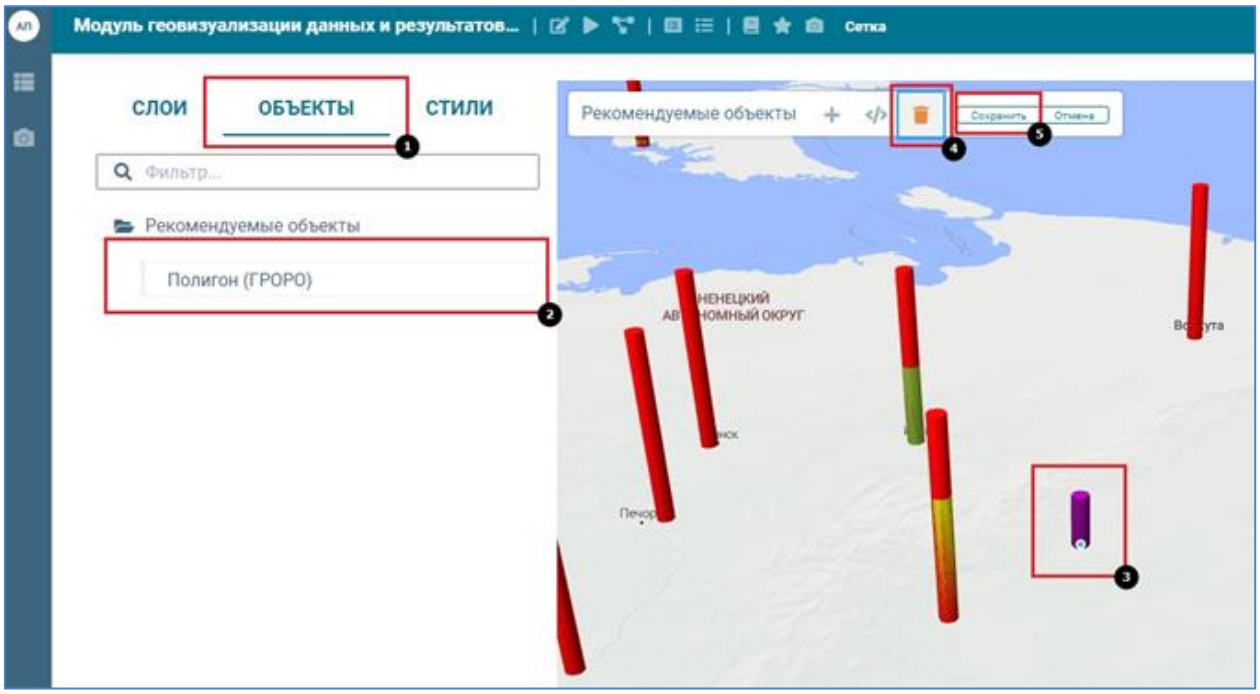


Рисунок 178 – Пример панели инструментов

Для того чтобы в режиме пресете на карте стали доступны **измерения на карте**, необходимо:

1. В режиме работы с графом перейти в настройки блока «3D-карта».
2. В настройках карточки включить метку «Отображать инструменты» и сохранить изменения.

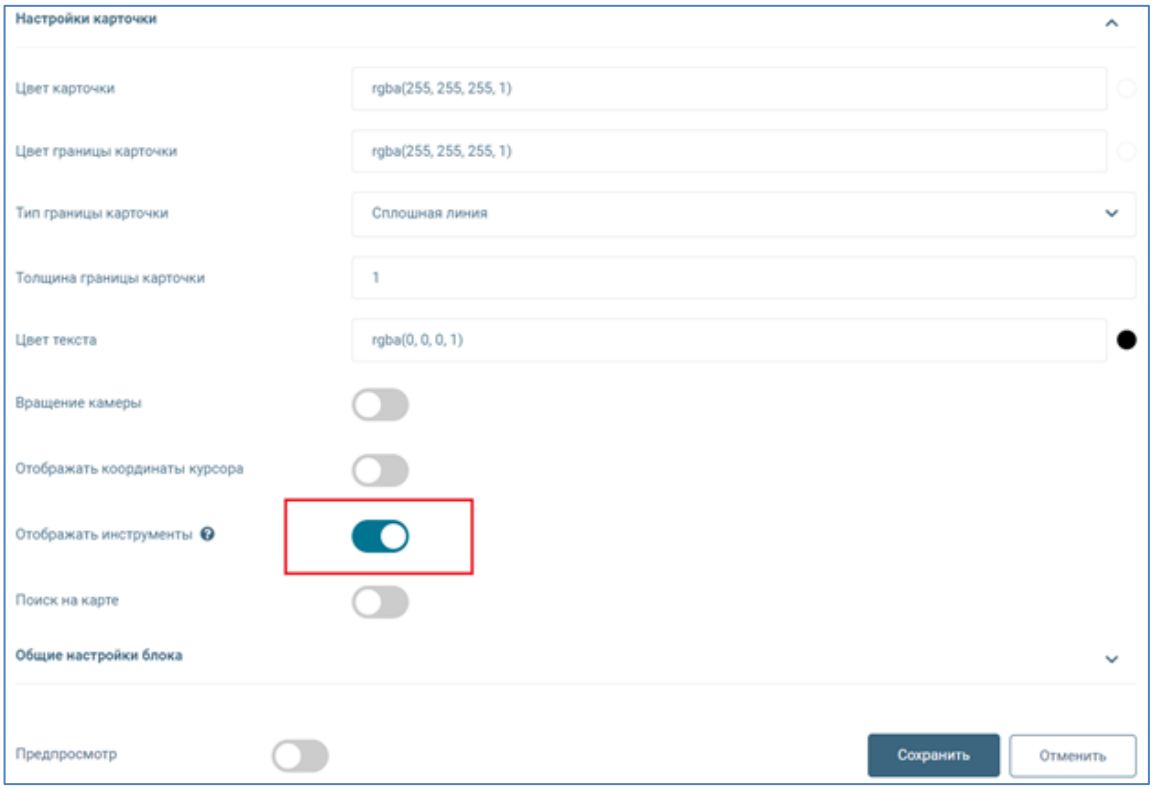


Рисунок 179 – Окно «Настройки карточки»

После успешного сохранения настроек измерения на карте в режиме пресета будут доступны инструменты для измерений на карте.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Для проведения **измерений на карте** необходимо:

1. Перейти в режим измерений, нажав на соответствующую кнопку в правой части визуализатора карта. Откроется панель измерений.
2. Выбрать инструмент для проведения измерений, нажав на соответствующую ему кнопку.

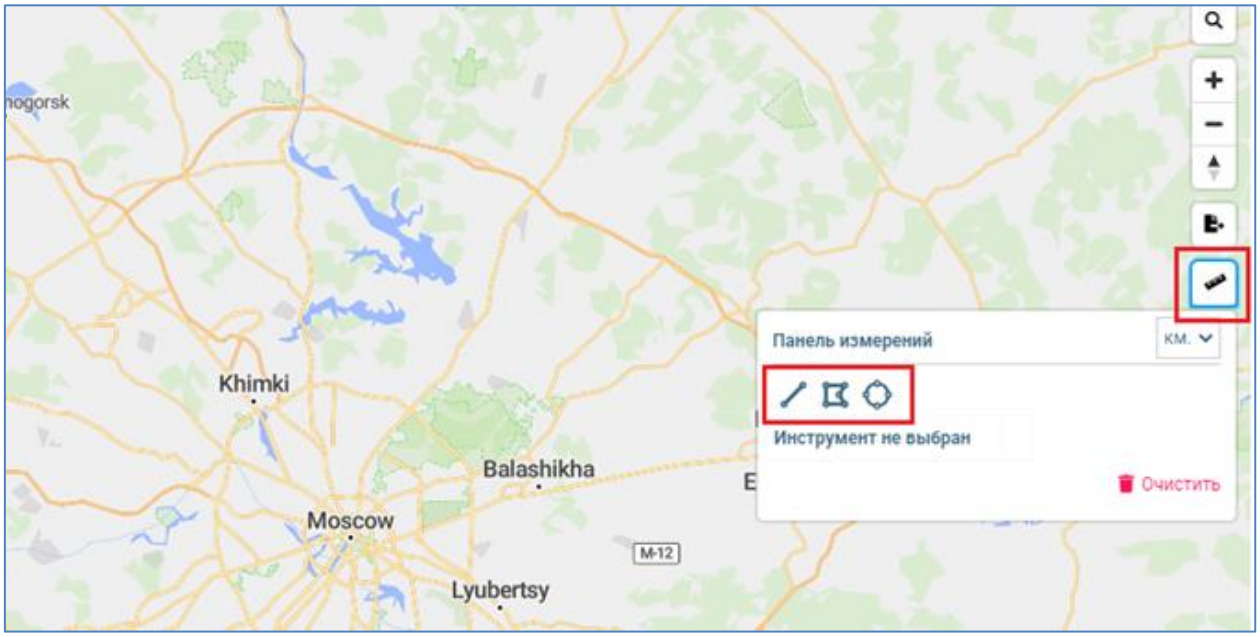


Рисунок 180 – Пример панели измерений на карте

3. Провести измерения, нажав на необходимую область карты, и растянуть инструмент измерения в необходимом объеме.
4. Для окончания рисования объекта нажать клавишу «Enter».

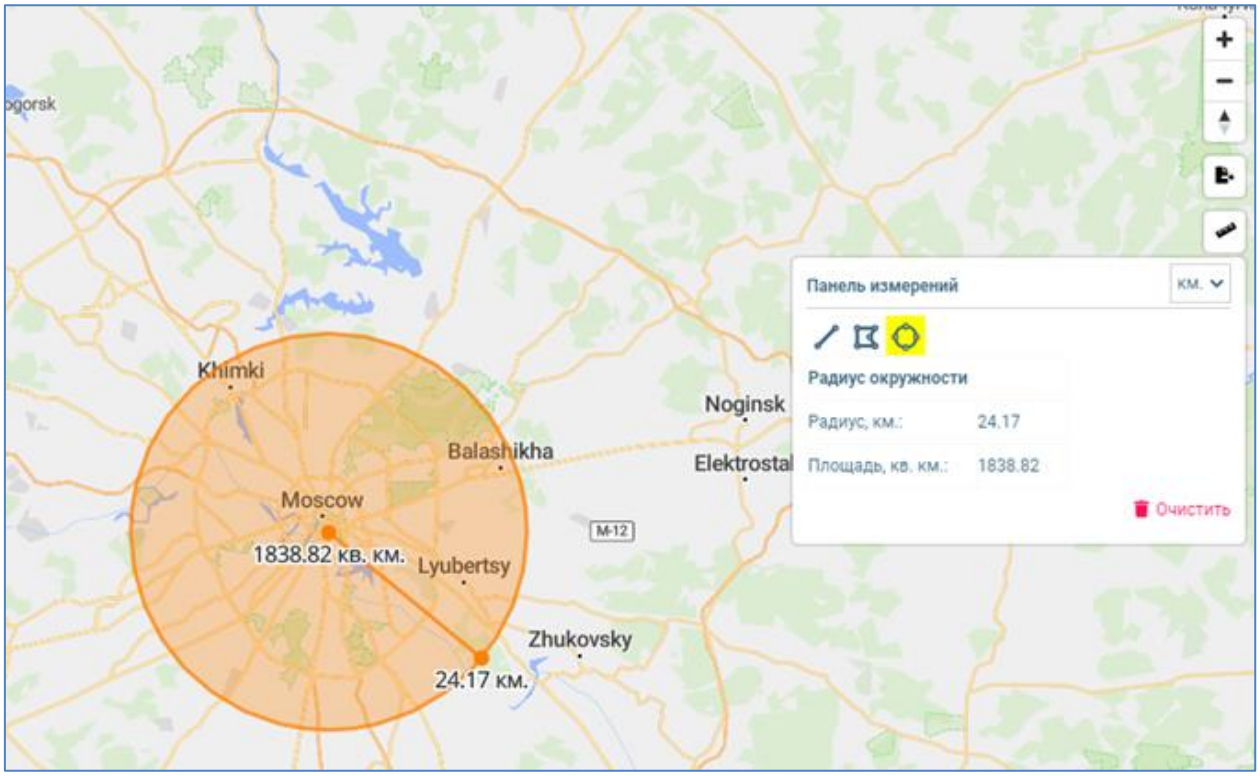


Рисунок 181 – Пример отображения карты

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.1.3.2 Блок «Генерация полигональной сетки»

6.1.3.2.1 Назначение

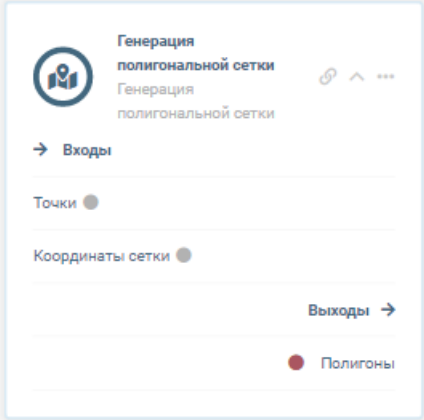


Рисунок 182 – Блок «Генерация полигональной сетки»

Блок позволяет сгенерировать сетку (слой, состоящий из одинаковых объектов) для отображения в блоке «3D-карта» (п. 6.1.3.1). Выходные данные должны быть обработаны в блоке «Слой данных для 3D карты» (п. 6.1.3.5).

Название параметра	Тип параметра	Обязательный	Описание	Пример
Наименование слоя	Строка	Нет	Название слоя для отображения в дереве слоев. Может отсутствовать или задаваться в описании слоя, см. детальное описание каждого слоя	
Тип геометрии	Выпадающий список	Да	Типы геометрии для генерации сетки. Возможные значения: • Шестиугольники (правильные) • Ромбы • Прямоугольники	
Сторона правильного шестиугольника, м	Целое число	Да	Доступен, если выбран тип геометрии: **Шестиугольники (правильные).** Задает сторону правильного шестиугольника в метрах	
Ширина ромба, м	Целое число	Нет	Доступен, если выбран тип геометрии: **Ромбы** Задает ширину ромба в метрах.	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
			Не обязательный параметр. При значении параметра равным нулю, будет сгенерирован правильный ромб (ширина равна высоте)	
Высота ромба, м	Целое число	Да	Доступен, если выбран тип геометрии: **Ромбы.** Задает высоту ромба в метрах	
Ширина прямоугольника, м	Целое число	Нет	Доступен, если выбран тип геометрии: **Прямоугольники.** Задает ширину прямоугольника в метрах. Не обязательный параметр. При значении параметра равным нулю, будет сгенерирован квадрат (ширина равна высоте)	
Высота прямоугольника, м	Целое число	Да	Доступен, если выбран тип геометрии: **Прямоугольники.** Задает высоту прямоугольника в метрах	
Покрытие сеткой	Выпадающий список	Да	Способ создания сетки. Возможные варианты: • По экстенту – задаются верхний левый и правый нижние координаты сетки. Генерация нескольких полигонов по экстенту невозможна. Требуется указать только две точки для одного полигона • По полигону – задаются все координаты полигона	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
Только пересечение с полигоном	Флаг	Да	Оставить только ту часть сетки, которая пересекается с полигоном. Не применяется при способе покрытия **По экстенту**	
Координаты сетки	JSON	Нет	Указываются координаты полигона/точки экстента для генерации сетки. Если на вход «Координаты сетки» пришли значения, то эти параметры игнорируются (координаты берутся из входа блока)	
Генерировать идентификаторы	Флаг	Нет	Необходимо ли сгенерировать идентификаторы для полигонов сетки. Идентификатором является целое число, начиная от нуля. Хранится в первичном ключе (**pkey**) объекта с идентификатором **id**	
Формирование связей полигонов	Флаг	Нет	Доступен, если включена настройка **Генерировать идентификаторы**. Необходимо ли сформировать массив со связями полигонов. При включении появляется отдельный выход «**Связи полигонов**»	
Генерировать наименования	Флаг	Нет	Необходимо ли сгенерировать наименования для полигонов сетки.	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
			Наименованием является строка, формата <ТИП><ЧИСЛО>, где: • ТИП – тип геометрии; • ЧИСЛО – уникальный идентификатор полигона, начиная с «1». Хранится в атрибутах (**attrs**) объекта с идентификатором **name**	
Атрибуты для агрегации	JSON	Нет	Формат агрегации атрибутов объектов, поступивших на вход «**Точки**». Массив словарей формата {**name**: <Имя атрибута>, **method**: <Метод, см. ниже>} Методы агрегации: • **sum** – сумма; • **avg** – среднее; • **concat** – соединение строковых элементов. Необходимо указать разделитель (параметр 'splitter'), при его отсутствии строки склеиваются без пробела	`[ {"name": "attr1", "method": "sum"}, {"name": "attr2", "method": "avg"}, {"name": "attr3", "method": "concat", "splitter": " "}, {"method": "concat", "splitter": " - "}, {"name": "attr5", "splitter": " - "}, {"name": "name", "method": "concat", "splitter": " - "} ]`

6.1.3.2.2 Группа «Общие настройки блока»

Аналогично приведенным в п. 6.1.1.1.3.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.1.3.2.3 Входы

Название входа	Тип	Обязательный	Описание
Точки	Объекты	Нет	Массив объектов с точками. Обязательным является наличие координат (поле **coords**) в атрибутах (**attrs**) объекта
Координаты сетки	Объекты	Нет	Массив объектов с полигонами. Обязательным является наличие координат (поле **coords**) в атрибутах (**attrs**) объекта

6.1.3.2.4 Выходы

Название выхода	Тип	Описание
Полигон	Объекты	Массив объектов с координатами полигонов сетки
Связи полигонов	JSON	Выход доступен если включена настройка **Формирование связей полигонов**. Словарь с идентификатором каждого полигона и массивом идентификаторов всех полигонов, соприкасающихся с ним. Пример выхода `{ "1": [2, 3, 4, 5], "2": [1, 3, 4], "3": [1, 2], "4": [1, 2], "5": [1] }` Из-за особенности формата JSON ключи словарей являются строками, хотя идентификаторы всех полигонов – целое число

6.1.3.2.5 События блока

В блоке нет событий.

6.1.3.2.6 Описание работы блока

Блок позволяет сгенерировать один из трех вариантов сетки: шестиугольники, ромбы или прямоугольники.

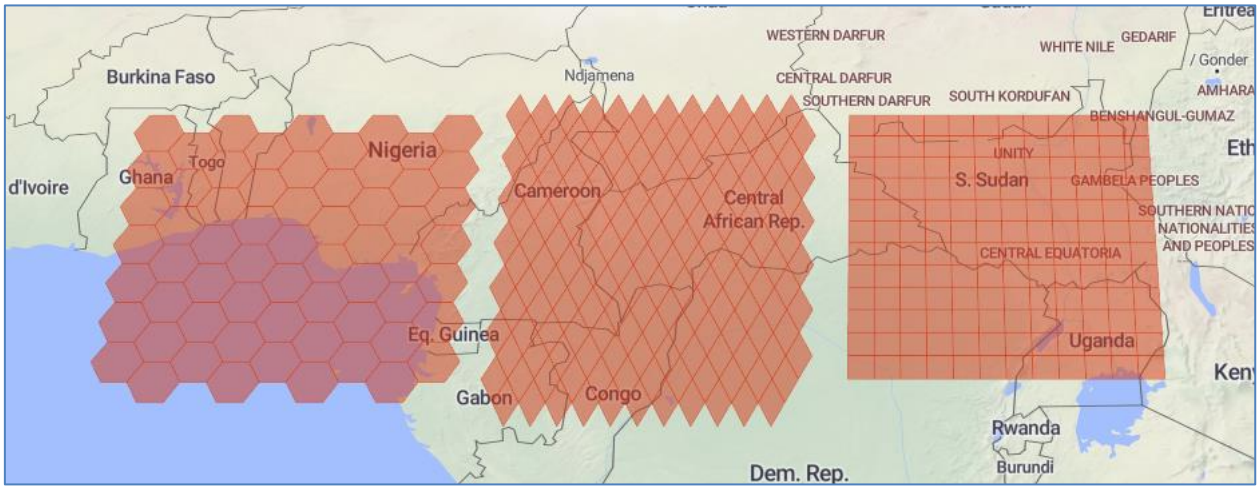


Рисунок 183 – Пример трех вариантов сетки

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

При генерации **по экстенту** блок создает полигон с крайними точками, указанными в настройках, после чего заполняет весь полигон фигурами с указанным размером.

При генерации **по полигону** блок объединяет все полигоны в один мультиполигон, после чего ищет его крайние точки, создает полигон и заполняет его фигурами. При необходимости (параметр **Только пересечение с полигоном**) удаляются лишние полигоны.

Если на вход «**Точки**» пришли объекты, то блок сопоставляет координаты точек с координатами полигонов сетки, и если в один полигон попало несколько точек, то они будут агрегированы (если задан параметр **Атрибуты для агрегации**).

6.1.3.2.7 Описание работы визуализатора

Блок не является визуализатором.

6.1.3.2.8 Настройки визуализатора

Блок не является визуализатором.

6.1.3.3 Блок «Конструктор графа»

6.1.3.3.1 Назначение

Блок позволяет объединить массив объектов в структуру «Граф».

6.1.3.3.2 Настройки блока

Название параметра	Тип параметра	Обязательный	Описание
Способ построения графа	Выпадающий список	Да	Выбор способа построения графа. Доступны четыре способа: 1. На основе географических точек и линий Для построения необходимо два массива объектов – Ребра и Вершины. В первичных ключах каждого объекта обязательно наличие поле **id**, содержащего уникальный идентификатор ребра или вершины. Все объекты без идентификатора не будут учитываться в построении графа. В атрибутах каждого объекта обязательно наличие параметра **coords**, содержащего координаты (линии для ребер или точки для вершин). Все объекты без координат не будут учитываться в построении графа. Для координат начала и конца каждой грани ищется вершина, попадающая в радиус формирования (см. параметр «*Радиус вершины для формирования графа*»). Первая найденная вершина будет соотнесена с началом/концом грани.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание
			Все ребра без вершин игнорируются Все вершины без ребер добавляются в граф (при этом граф автоматически становится несвязным) Граф проверяется на связность. 2. На основе связей (без географических привязок) Для построения необходимо два массива объектов – Ребра и Вершины. В первичных ключах каждого объекта обязательно наличие поле **id**, содержащего уникальный идентификатор ребра или вершины. Все объекты без идентификатора не будут учитываться в построении графа. В атрибутах каждого объекта-ребра обязательно наличие параметров **source_node_id** и **destination_node_id**, отвечающих за идентификаторы начальной и конечной вершины соответственно. Все ребра без одного или обоих параметров не будут учитываться в построении графа. Граф проверяется на связность
Радиус вершины для формирования графа	Целое число	Да	Доступно, если способ построения графа на основе географических точек и линий. Определяет радиус вокруг вершины, в котором будет происходить поиск начала или конца ребра, связанного с этой вершиной
Граф должен быть связным	Флажок	Нет	При включенном значении блок не будет продолжать расчет, если граф не связный

6.1.3.3.3 Входы

- **Ребра**

Обязательный вход. Массив объектов.

- **Вершины**

Не обязательный вход (необходим, если выбран способ построения «*На основе географических точек и линий»* или «*На основе связей (без географических привязок)»)*. Массив объектов.

6.1.3.3.4 Выходы

- **Граф**

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Структура из двух массивов объектов, связанных между собой.

Структура объекта «Граф»

```
{
 "id": <Уникальный идентификатор, генерируется при создании графа.>,
 "_source: <Блок-родитель графа>,
 "nodes": [<Массив объектов-вершин>],
 "edges": [<Массив объектов-граней>],
 "isMathGraph": <Флаг, показывающий то, что граф математический (без координат). Не обязательный параметр.>
}
```

6.1.3.4 Блок «Построение маршрутов объектов»

6.1.3.4.1 Назначение

Блок позволяет проложить маршрут по графу для одного или нескольких объектов.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.1.3.4.2 Настройки блока

Название параметра	Тип параметра	Обязательный	Описание	Пример
Раздел «Начало маршрута»				
Тип начальной точки	Выпадающий список	Да	Выбор типа начальной точки для объекта	
Выбор координат	Выпадающий список	Нет	Выбор варианта ввода координат	
Координаты начальной точки	Строка	Нет	Ввод координат в формате [<долгота>,<широта>]	[59.81, 44.72]
Атрибут начальной точки	Строка	Нет	Ввод идентификатора атрибута объекта, отвечающий за координату начала	
Выбор вершины	Выпадающий список	Нет	Выбор варианта ввода вершины. Если в атрибутах объекта есть поля **coords** или **start**, то в маршрут будет добавлена еще одна точка – от координат объекта до первой точки	
Идентификатор начальной вершины	Строка	Нет	Ввод идентификатора вершины	
Атрибут начальной вершины	Строка	Нет	Ввод идентификатора атрибута объекта, отвечающий за идентификатор вершины	
Раздел «Окончание маршрута»				
Тип конечной точки	Выпадающий список	Да	Выбор типа конечной точки для объекта	
Выбор координат	Выпадающий список	Нет	Выбор варианта ввода координат	
Координаты конечной точки	Строка	Нет	Ввод координат в формате [<долгота>,<широта>]	[59.81, 44.72]
Атрибут конечной точки	Строка	Нет	Ввод идентификатора атрибута объекта, отвечающий за координату окончания	
Выбор вершины	Выпадающий список	Нет	Выбор варианта ввода вершины. Если в атрибутах объекта есть поле **end**, то в маршрут будет добавлена еще одна точка – от координат объекта до первой точки	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
Идентификатор конечной вершины	Строка	Нет	Ввод идентификатора вершины	
Атрибут конечной вершины	Строка	Нет	Ввод идентификатора атрибута объекта, отвечающий за идентификатор вершины	
Раздел «Настройки маршрута»				
Учитывать веса ребер	Флажок	Да	Если включен – ищется маршрут с минимальным суммарным весом ребер. Обязательно наличия поле **weight** в атрибутах объектов-ребер (в графе)	
Учитывать направления ребер	Флажок	Да	Если включен – маршрут строится с учетом направления ребер (прямое/обратное). Обязательно наличия поле **edge_direction** в атрибутах объектов-ребер (в графе). Значения **edge_direction**: 0 – прямое – от первой точки координат до последней; 1 – обратное – от последней точки до первой	
Определить скорость	Выпадающий список	Да	Способ выбора скорости	
Скорость	Целое число	Нет	Ввод скорости всех объектов (км/ч)	
Атрибут, отвечающий за скорость	Строка	Нет	Ввод идентификатора атрибута объекта, скорость объекта (км/ч)	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.1.3.4.3 Входы

- **Объекты**

Обязательный вход. Массив объектов.

- **Граф**

Обязательный вход. Единичный объект типа «Граф».

6.1.3.4.4 Выходы

- **Объекты**

Массив объектов, содержащих в себе параметры маршрута:

Маршрут объекта

```
"route": {
 "graph": {
  "guid": <Идентификатор графа, на основании которого был построен маршрут>,
  "nodes": [<Список идентификаторов вершин, по которым прошел маршрут>],
  "edges": [<Список идентификаторов ребер, по которым прошел маршрут>]
 },
 "path": [<Массив с координатами пути>],
 "timestamps": [<Список временных меток маршрута (начало – всегда 0)>]
}
```

6.1.3.5 Блок «Слой данных для 3D-карты»

6.1.3.5.1 Назначение

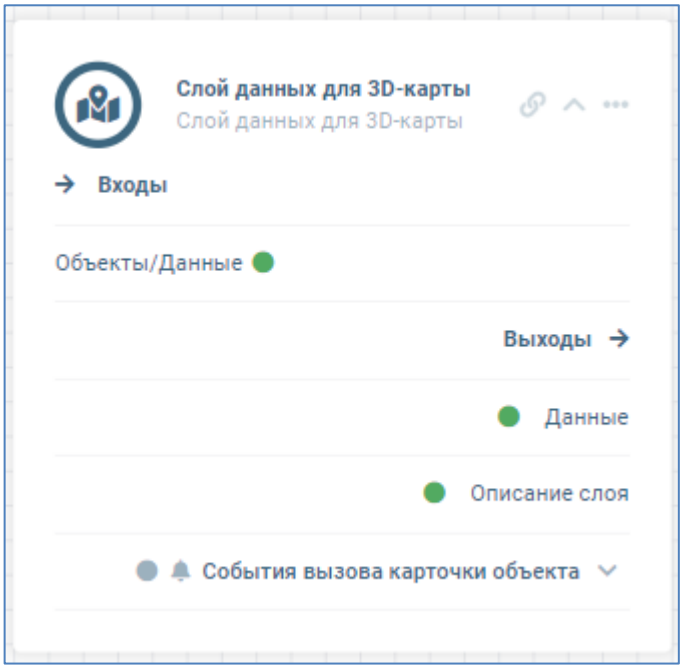


Рисунок 184 – Блок «Слой данных для 3D-карты»

Блок позволяет создать слой для отображения на 3D-карте (см. п. 6.1.3.1). Данные с входа преобразуются в geojson формат, который вместе с метаданными слоя может быть передан в Блок «3D-карта» (см. п. 6.1.3.1). Блок поддерживает разные типы слоев и обладает гибкой возможностью по их настройке.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
Наименование слоя	Строка	Нет	Название слоя для отображения в дереве слоев. Может отсутствовать или задаваться в описании слоя, см. детальное описание каждого слоя	
Тип слоя	Выпадающий список	Да	Выбор типа слоя карты. Доступные типы слоев: 1. **Обычный слой** – стандартный слой для карты. Доступен выбор типа геометрии объектов. 2. **Слой корреспонденций** – Расширенный «Обычный слой», позволяющий отображать связи между площадными объектами Несмотря на то, что для слоя доступен выбор геометрии, отображение линий корреспонденции возможно только для полигона. 3. **Слой маршрутов** – позволяет отображать анимацию линий (маршрутов) на карте 4. **Слой 3d-моделей –** позволяет отображать 3D-модели на карте. 5. **Географический граф –** слой, позволяющий отображать граф (вершины и ребра) на карте. 6. **Гексагональная тепловая карта** – позволяет отобразить на карте гексагональную сетку и раскрасить ее на основе значений по принципу тепловой карты. 7. **Внешний источник данных (геосервер)** – позволяет отобразить векторный слой данных, источником которых является геосервер (например Martin). 8. **Геосервер Martin** – специальный слой	См. раздел «Описание работы блока» для детального описания полей и их значений у каждого слоя. По умолчанию выбран **Обычный слой**

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
			для отображения функциональных источников данных (задающихся с помощью функций sql на сервере PostGIS) из геосервера Martin. 9. **Диаграмма (столбики)** – Расширенный «Обычный слой», позволяющий отобразить объекты в виде столбчатой диаграммы. 10. **Диаграмма (подписи)** – Расширенный «Обычный слой», позволяющий отобразить объекты в виде подписей	
Видимость слоя зависит от уровня зума	Флаг	Нет	Включение настроек минимального и максимального зума, при которых слой будет отображаться на карте.	
Минимальный зум	Вещественное число	Нет	Если включена настройка "Видимость слоя зависит от уровня зума". Минимальный зум, при котором слой будет отображаться на карте.	
Максимальный зум	Вещественное число	Нет	Если включена настройка "Видимость слоя зависит от уровня зума". Максимальный зум, при котором слой будет отображаться на карте.	
Динамические карточки	Флаг	Нет	Включение динамической карточки.	
Раздел «Настройки связи»				
Ограничения Раздел «Настройки связи» применяется только в случае, когда выбран тип слоя «Геосервер Martin» и имеет смысл только для связи с блоком «Анимация»				
Управляемый блок	Флаг	Да	Определяет, будет ли блок зависеть (пересчитываться) при событии в управляющем блоке	
Наименование	Строка	Нет	Идентификатор, задаваемый пользователем, для указания связи с управляющем блоке	

6.1.3.5.2 Группа «Общие настройки блока»

Аналогично приведенным в п. 6.1.1.1.3.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.1.3.5.3 Входы

Название входа	Тип	Обязательный	Описание
Объекты/Данные	JSON	Нет	Массив данных с типом «Ряд данных» или «Объект». См. подробное описание форматов данных для каждого слоя. Так же можно на вход передать сформированный ответ из блока "Кластеризация данных для 3D-Карты"

6.1.3.5.4 Выходы

Название выхода	Тип	Описание
Данные	JSON	Объекты слоя для размещения на карте в виде Json (объекты представляются в виде geojson)
Описание слоя	JSON	Описание слоя (стиль + метаданные)

6.1.3.5.5 События блока

Название выхода	Тип	Описание
Событие вызова карточки объекта	JSON	Параметры объекта, для которого была открыта карточка. Подробнее см. «Карточки объектов» → «Динамические карточки объектов» (п. 6.1.3.5.6.4.1)

6.1.3.5.6 Описание работы блока

Блок позволяет сформировать структуру данных, которую можно разместить на карте. Т.к. карта является достаточно специфическим визуализатором с большим количеством внутренних механизмов работы – данные для нее готовятся отдельным блоком (а для некоторых слоев – набором блоков). На выходе данные преобразуются в geojson согласно его спецификации (https://gis-lab.info/docs/geojson_ru.html) и набора метаданных слоя. В метаданных основными элементами является стиль отрисовки слоя картой.

Карта использует библиотеку maplibre.gl для работы, и может обрабатывать слои векторных или растровых источников согласно спецификации стилей MapBox Layer Style (Документация по стилям MapBox Layer Style).

Ниже представлены описания для каждого типа слоев с указанием их набора параметров.

6.1.3.5.6.1 Передача атрибутов на слой и обязательные атрибуты

Во входных данных может быть очень много разной атрибутики, которую нет необходимости передавать на карту, т.к. она будет только увеличивать объем информации, которая уходит на клиента. Для этого у некоторых слоев есть настройка «Атрибуты для передачи на слой». В ней перечисляются в виде массива идентификаторы всех атрибутов, которые должны быть переданы на слой и будут на нем доступны (в режиме редактирования или для других целей).

При этом, есть несколько важных моментов, связанных с именованием атрибутов.

В объекте обязательно должен быть атрибут **coords** для преобразования его в координаты объекта. В нем должны быть координат в виде пары [широта, долгота] или массива пар – структура аналогичная спецификации goejson. Передавать **coords** в списке атрибутов для слоя не нужно.

Атрибут **name** будет использоваться в качестве имени объекта. Если его нет – имя объекта будет равно уникальному GUID. Передавать **name** в списке атрибутов для слоя не нужно.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Атрибут **popup** необходимо передать в списке атрибутов. Если он будет в объекте – карта покажет его содержимое как карточку объекта, на базе HTML верстки. Можно использовать стили или генерировать эту карточку в других блоках.

При расчете блока координаты проверяются на соответствие стандарту и корректность геометрии, атрибуты на наличие – после чего объект конвертируется в goejson и передается на выход.

6.1.3.5.6.2 Кэширование

Блок Слой 3Д карты позволяет работать с объектами на карте, не передавая их в карту целиком. Это достигается за счет использования механизмов кэширования в Redis исходных данных и связки их с передаваемыми на карту через уникальный GUID. При редактировании данных их кэша вытягивается оригинальный объект, после чего происходит его изменение и сохранение в источник данных (справочник, запрос и т.д.)

6.1.3.5.6.3 Управление при позиционировании на объект

Для возможности управления поведением при позиционировании на объекте необходимо настроить атрибут **flyTo**.

Возможные параметры для настройки описаны в документации mapbox:

- https://docs.mapbox.com/mapbox-gl-js/api/map/?size=n_10_n#map#flyto
- https://docs.mapbox.com/mapbox-gl-js/api/properties/#cameraoptions
- https://docs.mapbox.com/mapbox-gl-js/api/properties/#animationoptions

При расчете блока не происходит валидации настроек атрибута.

6.1.3.5.6.4 Карточки объектов

При наличии у объекта атрибута **popup** в списке атрибутов в момент расчета данные атрибута заменяются флагом **true** (исходные данные сохраняются в системной БД для получения в момент щелчка по объекту).

При этом, в данных объектов будет отображен системный флаг **lazyPopup**: **true**.

6.1.3.5.6.4.1 Динамические карточки объектов

В отличии от простых карточек динамические карточки формируются в графе с использованием данных объекта.

Для включения динамической карточки необходимо в описании слоя указать атрибут **dynamicPopup**: **1**.

При этом, в данных объектов будет отображен системный флаг **dynamicPopup**: **true**.

В момент открытия динамической карточки, на событийный выход **Данные карточки** приходит информация об объекте: атрибут **popup** и ключ объекта.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Рисунок 185 – Пример формирования динамической карточки в графе

Для отображения будет использован HTML, сформированный на выходе блока «HTML-шаблон» (п. 6.1.4.2).

Цепочка блоков должна быть соединена со входом «Данные карточки».

Предпоследним блоком (до блока карты) в цепочке должен быть блок «HTML-шаблон».

Любые визуализаторы в этой цепочке в текущей реализации обновляться при расчете не будут, если они размещаются на пресете, данный механизм только для всплывающих подсказок.

Если в цепочке будет стоять несколько блоков "HTML-шаблон", то будет выдано предупреждение.

Данные для карточки, в таком случае, придут из первого рассчитанного блока

Данные, переданные на событийный выход:

```
{
    "key": <Ключ объекта>,
    "dat": <Данные атрибута popup>,
    "source": <Идентификатор блока слоя карты>
}
```

6.1.3.5.6.4.2 Динамические карточки объектов для слоя Martin

При отображении карточки для гео-сервера Martin в событийном выходе придет идентификатор карточки (поле **properties.id** из объекта на геосервере).

Данные, переданные на событийный выход:

```
{
    "key": {
        "id": <Идентификатор объекта (properties.id)>
    },
    "source": <Идентификатор блока слоя карты>
}
```

Остальная логика работы не меняется, т.е. для отображения будет использован HTML, из блока "HTML-шаблон".

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.1.3.5.6.5 Редактирование

Редактирование доступно для слоя **Обычный слой** в случае:

1. Данные были переданы из блока с возможностью их сохранения обратно в источник, например Справочник.
2. В описании слоя стоит флаг **editable: true**.
3. Слой не является 3D – редактировать можно только 2D-слои.

6.1.3.5.6.6 Полезные атрибуты в описании слоя, с которыми может работать карта

id	идентификатор слоя
name	наименование
type	тип геометрии (mapbox style)
editable	можно ли редактировать слой
hidden	скрывать слой по умолчанию
groupName	имя группы в которую будет добавлен слой в дереве слоев на карте

Для продвинутых пользователей

После этого блока geojson с данными может быть изменен до попадания в карту как необходимо прикладному разработчику. Однако, сохранение в этом случае может перестать работать.

6.1.3.5.7 Обычный слой

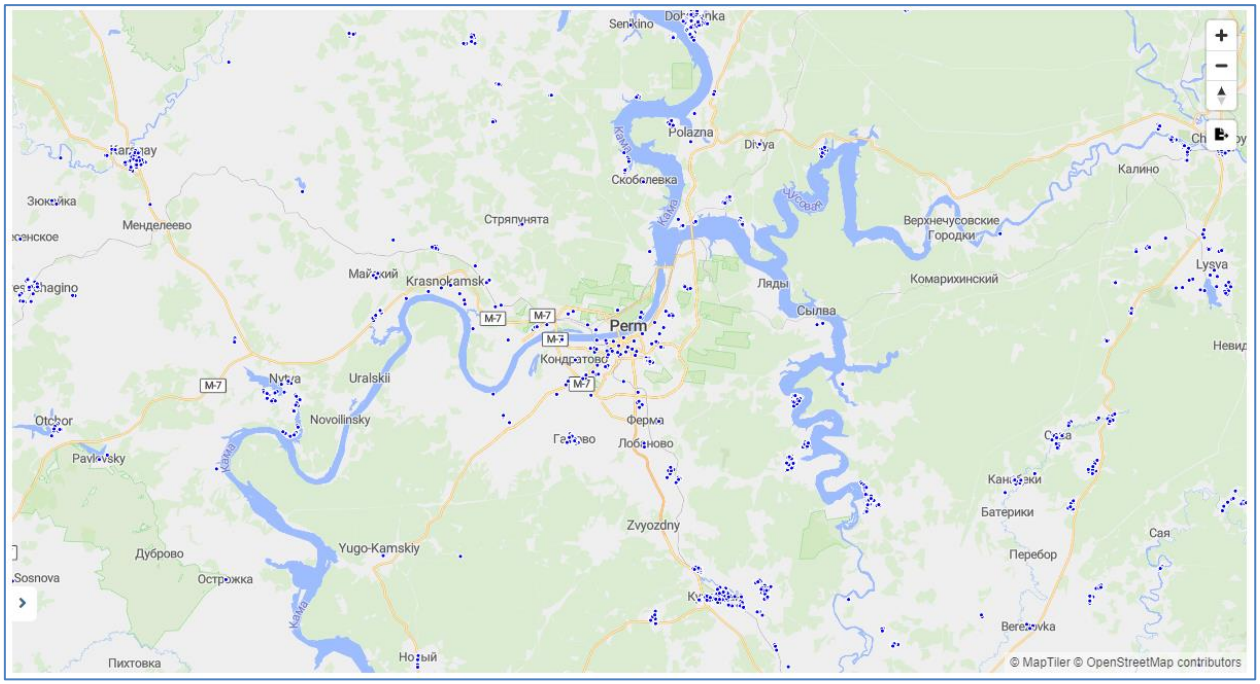


Рисунок 186 – Пример обычного слоя

Назначение: Для отображения на карте объектов заданной геометрии с задаваемым пользователем стилем.

Описание: стандартный слой для карты. Доступен выбор типа геометрии объектов. При расчете блока с этим типом слоя будет подготовлен geojson и стиль для отрисовки слоя на карте.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Параметры слоя:

Название параметра	Тип параметра	Обязательный	Описание	Пример
Тип геометрии	Выпадающий список	Да	Определяет какой тип геометрии будет использоваться в слое. От него зависит и стиль для слоя.	
Объекты содержат маршруты	Флаг	Да	Задает поведение слоя. Если в исходных данных присутствует атрибут **routes** с точками для построения маршрута – будет добавлен раздел настроек для слоя с маршрутами и слой будет отрисован как дополнительный к создаваемому (поверх него)	

Описание слоя	JSON	Да	Описание слоя согласно стилю Mapbox для отображения на карте. Дополнительные поля, которые должны быть указаны
			1. Id – уникальный идентификатор слоя, так же служит для связи с другими механизмами карты, например иерархией слоев.
			2. Source – уникальный идентификатор, который будет назначен geojson с данными для карты, который будет получен из блока.
			3. type – тип геометрии слоя согласно спецификации стилей: Документация по стилям MapBox Layer Style
				Пример настроек
				`{`
				`'id': 'circle',`
				`'source': 'my-awsome-source',`
				`'type': 'circle',`
				`'paint': {`
				`'circle-radius': ['get', 'radius'],`
				`'circle-color': '#B42222',`
				`'circle-stroke-color': 'white',`
				`'circle-stroke-opacity': 0.9,`
				`'circle-stroke-width': 2,`
				`'circle-opacity': 0.4,`
				`'circle-pitch-alignment': 'map',`
				`'circle-pitch-scale': 'map',`
				`}`
				`}`

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

<table>
<tr>
<td>Описание слоя маршрутов</td>
<td>JSON</td>
<td>Нет (допустим пустой стиль)</td>
<td>Описание слоя аналогично обычному слою
В случае если есть необходимость отобразить маршрут движения объекта по карте – можно в данных задать специальный атрибут route в котором определенным образом должны быть указаны координаты точек движения объекта для отображения на карте. Из раздела route берется подраздел path.
В случае отображения слоя маршрутов блок сгенерирует 2 слоя и 2 набора метаданных.
Этот вариант можно совместить со слоем тип «Слой маршрутов» для отрисовки и анимированного объекта и его маршрута.
Пример входных данных с маршрутами:
<pre>{
        "pkey": {
            "id": 1
        },
        "_params": {
            "ts": [],
            "lazy": [],
            "key_field": "id",
            "name_field": "name"
        },
        "attrs": {
            "name": "OBJECT 1",
            "ord": 1,
            "coords": "[1,1]",
            "end_coord": "[21, 3]",
            "end_node": 17,
            "my_coords": "[1,3]",
            "start_node": 13,
            "route": {
                "graph": {
                    "guid": "16d38a35-502a-493c-a038-ba5cdaa8ae3e",
                    "nodes": [
                        "13",</pre></td>
<td>Пример настроек
<pre>{
    "id": "edge_layer_trip",
    "source": "edge_layer_trip_source",
    "type": "line",
    "paint": {
        "line-color": "#73fc03",
        "line-opacity": 0.8,
        "line-width": 15
    },
    "layout": {
        "line-cap": "butt",
        "line-join": "round"
    },
    "groupName": "Граф"
}</pre></td>
</tr>
</table>

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
			"17"], "edges": ["15"] }, "path": [[1, 1], [10, 2], [13, 4], [20, 2]], "timestamps": [0, 402.62403509459534, 562.8340301192493, 886.2129248722995] } }, "child": [], "_source": "371f222d-2cc9-4409-b6e4-18ba7cc39f32" },	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
Атрибуты для передачи на слой	JSON	Нет (допустим пустой список)	Список атрибутов, которые будут переданы на слой карты	`["popup", "name"]`

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.1.3.5.7.1 Подсветка объектов

Для того чтобы на карте объект подсвечивался при наведении курсора или при нажатии – можно использовать флаги onHover = true и onClick = true. Для самой подсветки необходимо в описании слоя задавать стиль отображения для наведения и клика по объекту. Описание стилей соответствует описанию в документации стилей MapBox.

Пример onHover и onClick:

```
{
    "id": "squares_aero_fill_layer",
    "source": "squares_aero_source",
    "type": "fill",
    "onHover": true,
    "onClick": true,
    "paint": {
        "fill-color": [
          "case",
          ["boolean", ["feature-state", "hover"], false],
          "#fbb03b",
          ["boolean", ["feature-state", "clicked"], false],
          "#fbb03b",
          "#F6FBF5"
        ],
        "fill-outline-color": [
          "case",
          ["boolean", ["feature-state", "hover"], false],
          "#fbb03b",
          ["boolean", ["feature-state", "clicked"], false],
          "#ff0000",
          "#FFFF00"
        ]
    }
}
```

Дополнительные возможности:

Если слою указать тип **heatmap** и при этом отобразить точки – карта перейдет в режим отображения тепловой карты с группировкой точек. Подробнее: https://docs.mapbox.com/mapbox-gl-js/style-spec/layers/#heatmap

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.1.3.5.8 Слой корреспонденций

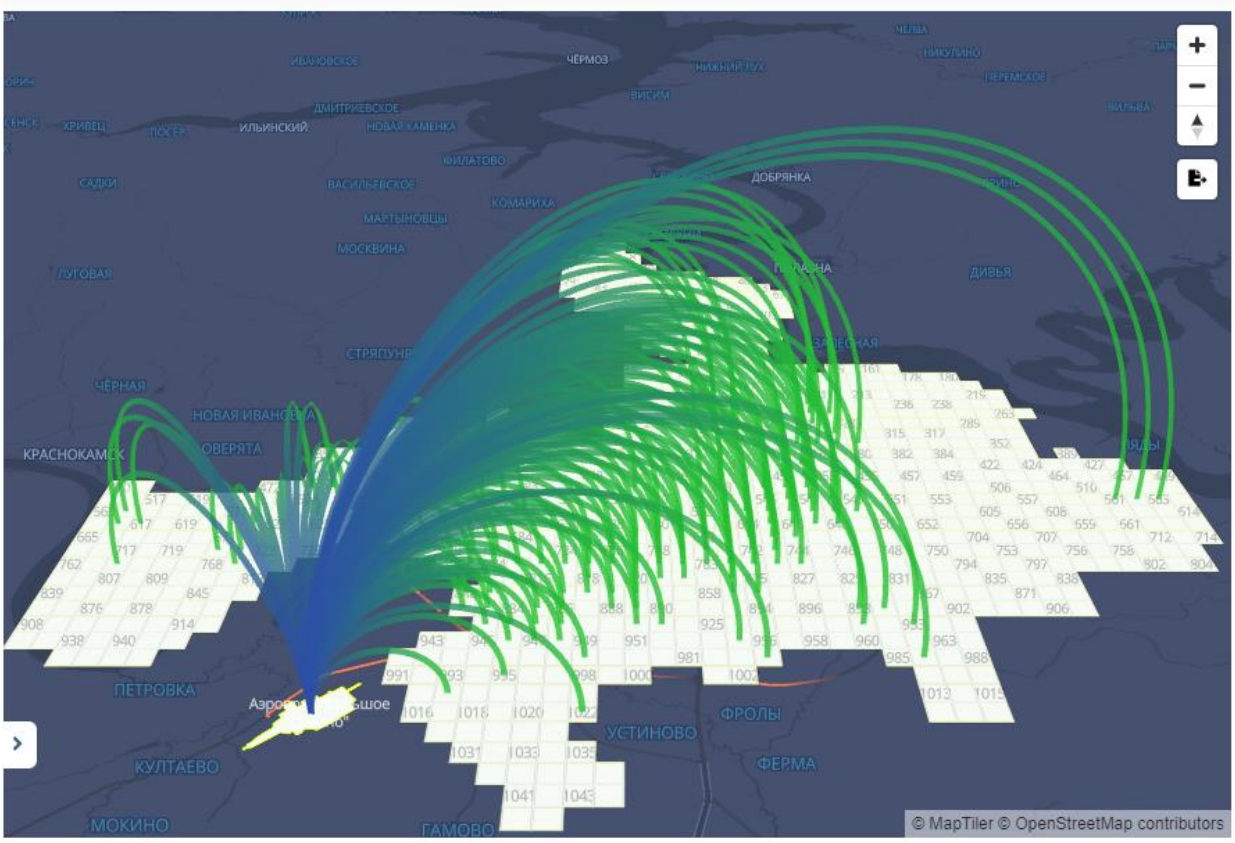


Рисунок 187 – Пример слоя корреспонденций

Назначение: для отображения связей между объектами пользовательского слоя, который является аналогичным «Обычному слою», но дополнительно содержит информацию о связях между объектами.

Описание: позволяет отображать на карте связи между объектами в виде дуг, идущих от одного полигона до другого. Для работы слоя используется библиотека **deck.gl**, слой которой рисуется поверх карты.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Параметры слоя:

Название параметра	Тип параметра	Обязательный	Описание	Пример
Тип геометрии	Выпадающий список	Да	Определяет какой тип геометрии будет использоваться в слое. От него зависит и стиль для слоя. В данном слое для отображения связей необходимо указывать базовый слой, из которого будут выходить дуги. Обычно это полигоны – регионы или какой либо вариант сетки (квадраты, гексагоны)	
Описание слоя	JSON	Да	Описание слоя согласно стилю Mapbox для отображения на карте. Дополнительные поля, которые должны быть указаны: 1. Id – уникальный идентификатор слоя, так же служит для связи с другими механизмами карты, например иерархией слоев. 2. Source – уникальный идентификатор, который будет назначен geojson с данными для карты, который будет получен из блока. 3. type – тип геометрии слоя согласно спецификации стилей (Документация по стилям MapBox Layer Style)	**Пример настроек** `{ "id": "squares_aero_fill_layer", "source": "squares_aero_source", "type": "fill", "onHover": true, "onClick": true, "paint": { "fill-color": ["case", ["boolean", ["feature-state", "hover"], false], "#fbb03b", ["boolean", ["feature-state", "clicked"], false], "#fbb03b", "#F6FBF5"], "fill-outline-color": ["case",`

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
				["boolean", ["feature-state", "hover"], false], "#fbb03b", ["boolean", ["feature-state", "clicked"], false], "#ff0000", "#FFFF00"] } }
Атрибуты для передачи на слой	JSON	Нет (допустим пустой список)	Список атрибутов, которые будут переданы на слой карты	["id", "name"]
Отображать все связи	Флаг	Нет	При включении - будут отображаться все связи для выбранного объекта. Поле **defaultId** при этой настройке можно не указывать.	
Точность расчета центра полигонов	Вещественное число	Да	Для поиска центра полигона (координаты из которой должны выходить дуги) используется метод polylabel. Эта настройка задает погрешность метода, чем меньше значение, тем точнее будет определен центр и тем дольше будет считаться блок	0.01
Настройки отображения корреспонденций	JSON	Да	Описание слоя ArcLayer (https://deck.gl/docs/api-reference/layers/arc-layer) для отрисовки связей. Слой задает оформление для дуг. Обязательные поля, которые должны быть указаны, но не входят в описание слоя в документации: • id – уникальный идентификатор слоя;	**Пример настроек** { "id": "arc", "autoHighlight": true, "highlightColor": [249,138,44], "showTooltip": false, "defaultId": 0, "getSourceColor": [44, 79, 167],

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример

			• defaultId – идентификатор объекта, дуги из которого будут отображаться при открытии карты;
			• name – наименование слоя для отображения в списке слоев.
			Внимание!
			Если указать некорректный defaultId – связи не будут отображаться.
			Структура данных:
			Для корректного отображения в данных, поступающих в блок, должны быть специальные атрибуты **flows** и атрибут **id**
			Структура атрибута **flows: «{\"231\":1080,\"248\":1520}"**
			Представляет собой пары – идентификатор полигона, в который должна идти связь, и значение (это значение будет выведено при нажатии на связь).
			Пример входных данных со связями:
			`{`
			`"pkey": {},`
			`"_params": {`
			`"lazy": []`
			`},`
			`"attrs": {`
			`"id": 176,`
			`"name": "176",`
			`"flows": "{\"0\":10}",`
			`"coords": "[[[56.443641, 58.146598], [56.460773, 58.146598], [56.460938, 58.155591], [56.443804, 58.155591], [56.443641, 58.146598]]]",`
			`"popup": "<b>Квадрат 176</b>"`
				`"getTargetColor": [35, 196, 46],`
				`"getWidth": 5,`
				`"opacity": 0.6,`
				`"name": ""`
				`}`

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
			}, "child": [], "_source": "6dea435d-183d-457a-a03c-b24107efff24" }	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.1.3.5.8.1 Особенности работы слоя:

- Слой позволяет настраивать градиент от начала до конца дуги, ее размеры.
- Слой **не поддерживает** задание цвета и размера дуги в зависимости от значений атрибутов объекта.
- В подсказке к дугам выводится наименование стартового полигона и конечного полигона.
- Чем больше расстояние между стартовым и конечным полигоном, тем выше будет дуга.
- По слою самих дуг недоступен поиск.

6.1.3.5.9 Слой маршрутов

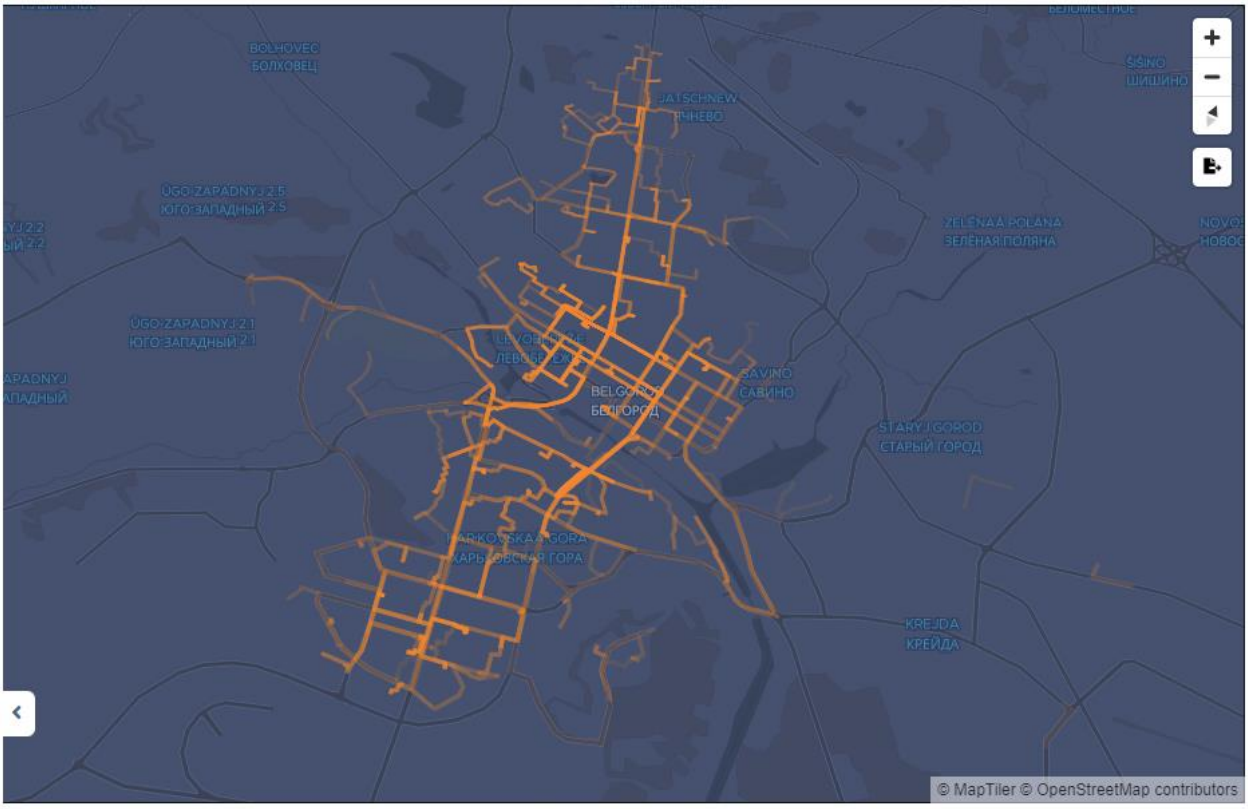


Рисунок 188 – Пример слоя маршрутов

Назначение: позволяет визуализировать карту маршрутов или загруженности дорог с анимацией движения по ним точек.

Описание: позволяет отображать анимацию линий (маршрутов) на карте. Анимация выглядит как перемещение по кривой точки, на определенной высоте с «остаточным» следом за ней. Слой базируется на слое tripsLayer библиотеки deck.gl (https://deck.gl/docs/api-reference/geo-layers/trips-layer) с добавлением анимации движения.

Блок работает в связке с блоком «Построение маршрутов объектов» (см. п. 6.1.3.4), но если на вход подать готовые данные – слой будет нарисован и без блока, генерирующего маршруты.

При использовании анимации в слое она зацикливается и прокручивается по кругу.

Слой не содержит как таковых geojson объектов, что отличает его от других.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Важно!

Слой требует специфически подготовленных данных. Для быстрой генерации используется Блок «Построение маршрутов объектов» (п. 6.1.3.4), но можно подготовить и вручную на уровне БД или Блок «Python» (п. 6.9.1).

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Параметры слоя:

Название параметра	Тип параметра	Обязательный	Описание	Пример
Объекты содержат маршруты	Флаг	Да	Определяет содержат ли объекты маршруты (атрибут **route**) или отображение будет только для точек. Обычно для этого слоя флаг включают	

Описание слоя	JSON	Да	Описание слоя TripsLayer (https://deck.gl/docs/api-reference/geo-layers/trips-layer) для отрисовки маршрутов. Слой задает оформление для движения точек и их следов.
			Обязательные поля, которые должны быть указаны, но не входят в описание слоя в документации
			• id – уникальный идентификатор слоя;
			• sourceId – уникальный идентификатор набора данных, которые будут переданы на карту для этого слоя;
			• animate – флаг анимировать или нет слой;
			• startTime – с какого фрейма стартовать анимацию. См. структуру данных, атрибут **route.timestamps;**
			• loopLength – сколько фреймов будет в цикле анимации;
			• speed – скорость анимации, во фреймах.
			Структура данных
			Обычно, под каждую карту и каждый набор данных эти параметры приходится подбирать вручную.
			Для корректного отображения маршрута в объекте используется специальный атрибут **route**. Внутри этого атрибута содержится два важных раздела (если он сгенерирован с помощью графа маршрутов, то три):
			• path – набор координат LineString формата по которому будет двигаться точка;
				Пример описания слоя
				`{`
				`id: 'trips',`
				`sourceId: 'data_for_routes',`
				`getColor: [255,255,0],`
				`opacity: 1,`
				`widthMinPixels: 3,`
				`rounded: true,`
				`trailLength: 600,`
				`shadowEnabled: true,`
				`animate: true, <- не является частью настроек слоя в карте, но нужна для определения надо ли анимировать слой`
				`startTime: 4000, <- не является частью настроек слоя в карте, но нужна для определения с какой миллисекунды начинать анимацию`
				`loopLength: 10000, <- не является частью настроек слоя в карте, длина цикла анимации`
				`speed: 5 <- не является частью настроек слоя в карте, скорость анимации`
				`}`

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример

			• timestamps – миллисекунды, когда точка должна оказаться в следующем узле своего маршрута. За счет этого можно обеспечивать ускорение точки на одних участках маршрута и замедление на других.
			Важно!
			Количество timestamps элементов должно совпадать с количеством узлов маршрута.
			Пример входных данных с маршрутами:
			{
			"pkey": {
			"id": 1
			},
			"_params": {
			"ts": [],
			"lazy": [],
			"key_field": "id",
			"name_field": "name"
			},
			"attrs": {
			"name": "OBJECT 1",
			"ord": 1,
			"coords": "[1,1]",
			"end_coord": "[21, 3]",
			"end_node": 17,
			"my_coords": "[1,3]",
			"start_node": 13,
			"route": {
			"graph": {
			"guid": "16d38a35-502a-493c-a038-ba5cdaa8ae3e",
			"nodes": [
			"13",
			"17"
			],	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
			"edges": ["15"] }, "path": [[1, 1], [10, 2], [13, 4], [20, 2]], "timestamps": [0, 402.62403509459534, 562.8340301192493, 886.2129248722995] } }, "child": [], "_source": "371f222d-2cc9-4409-b6e4-18ba7cc39f32" }	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.1.3.5.9.1 Особенности работы слоя:

- Слой на карте не поддерживает какого-либо интерактивного взаимодействия.
- Скорость анимации будет зависеть от мощности клиента, т.к. этот слой для отрисовки активно использует GPU.
- Слой поддерживает достаточно большое количество одновременных точек (тысячи одновременно анимируемых точек).
- По слою недоступен поиск.

6.1.3.5.10 Слой 3D-моделей

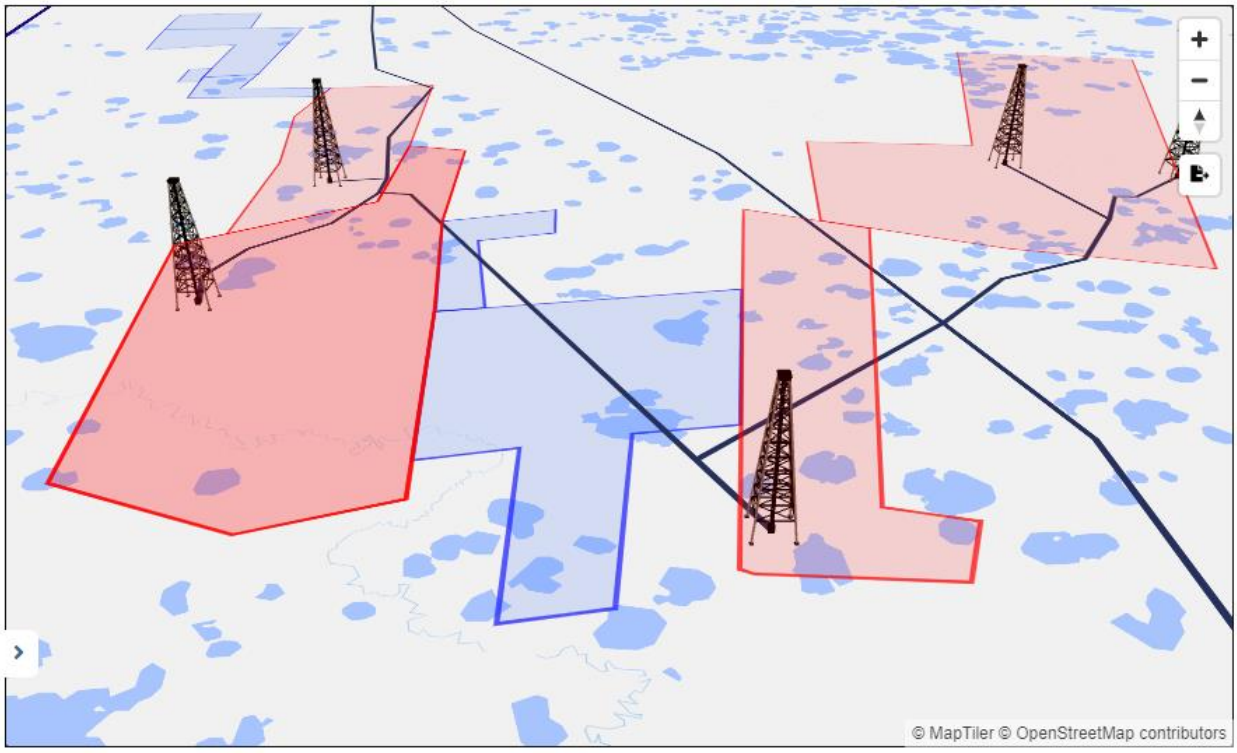


Рисунок 189 – Слой 3D-моделей

Назначение: размещение 3D-моделей на слой, для реализации более наглядного функционала на карте.

Описание: слой позволяет разместить на карте 3D-модели в формате .gltf или .glb. Модели при этом должны быть размещены на внешнем ресурсе, откуда клиент сможет их скачать (карта). Слой базируется на основе библиотеки three.js и ее интеграции с картой (https://docs.mapbox.com/mapbox-gl-js/example/add-3d-model/).

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Параметры слоя:

Название параметра	Тип параметра	Обязательный	Описание	Пример

Описание слоя	JSON	Да	В поле задается описание используемых 3d-моделей. Обязательно наличие трех параметров:
			• список моделей слоя ("models", массив);
			• идентификатора слоя ("id", строка);
			• идентификатора источника данных ("sourceId", строка).
			В разделе models описывается какие модели необходимо будет добавить на карту. Указывается полный список, который надо размещать на слое. Каждый элемент списка – это одна модель с полями:
			• ext – расширение файла модели;
			• name – наименование (идентификатор) модели. По нему осуществляется связка модели и объекта в данных;
			• path – путь для скачивания модели. Может быть относительным и абсолютным. Модель должна быть размещена на доступном для клиента ресурсе (в локальной сети или сети Интернет)
				Пример настроек
				`{`
				`"id": "my_awsome_3d_layer",`
				`"sourceId": "my_awsome_3d_layer_source",`
				`"models": [`
				`{`
				`"ext": ".glb",`
				`"name": "oil_tower",`
				`"path": "static/3d-models/oil_tower.gltf"`
				`},`
				`{`
				`"ext": ".glb",`
				`"name": "oil_tower2",`
				`"path": "static/3d-models/oil_tower2.gltf"`
				`}`
				`],`
				`"groupName": "3D Модели"`
				`}`

Привязка слоя 3D объектов	JSON	Да	В поле задается привязка атрибутов объекта с полями моделей.
			Формат записи: <Наименование поля в модели (можно не изменять значения по умолчанию)>:<Наименование поля у объекта>
			Эти поля должны приходить в начальных данных и затем происходит их связка с моделью на слое. Каждому объекту может соответствовать одна модель, однако одна модель может быть использована для разных объектов.
				Пример настройки
				`{`
				`"modelName": "modelName",`
				`"position": "coords",`
				`"rotation": "rotation",`
				`"scale": "scale",`
				`"altitude": "altitude",`
				`"meshName": "meshName",`
				`"animationName": "animationName"`
				`}`

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример

			Поля привязки:
			• "modelName" – наименование модели, строка, обязательное;
			• "position" – координаты расположения объекта, обязательное;
			• rotation" – поворот модели относительно ее оси;
			• "scale" – масштаб модели, относительно ее начального размера;
			• "altitude" – угол наклона модели;
			• "meshName" – необходимо для анимации, должно быть внутри 3d-модели;
			• "animationName" – имя анимации модели, которую нужно запустить.
			Ниже приведена структура объекта из блока Справочник с полями, которые нужны для размещения модели.
			Пример входных данных:
			{
			"pkey": {
			"id": 6
			},
			"_params": {
			"ts": [],
			"lazy": [],
			"key_field": "id"
			},
			"attrs": {
			"name": "Вышка 1",
			"ord": 6,
			"modelName: "Tower", // обязательное поле
			"position": [148.9819, -35.39847], // обязательное поле	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
			"altitude": 2.5, // опциональное поле, по умолчанию = 0, наклон модели "rotation": [3.1415, 0, 0], // опциональное поле, поворот модели относительно начального положения "scale": 5, // опциональное поле, размер модели относительно начального "meshName": "mesh_0", // опциональное поле, необходимо для анимации, должно быть внутри 3d-модели "animationName": "Run" // опциональное поле, необходимо для анимации, должно быть внутри 3d-модели }, "child": [], "_source": "e4af2dfa-6dc8-49d2-afcb-e0812cefbbe4" }	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.1.3.5.10.1 Особенности работы слоя:

- Так как модель не имеет привязки к размерам в реальном мире, необходимо использовать преобразования поворота и изменения размера для отображения модели в нужной точке и нужном масштабе.
- Модели должны быть размещены на доступном пользователе ресурсы. В карту передаются только описания откуда их брать и как размещать.
- Модели могут иметь текстуры.
- Модели могут быть анимированными.
- По слою недоступен поиск.

6.1.3.5.11 Географический граф

Рисунок 190 – Пример географического слоя

Назначение: отображение графа на карте, например графа транспортной сети или авиаперевозок.

Описание: Слой позволяет построить на карте граф из вершин и ребер, и отобразить его в виде двух слоев – слоя точек (вершин графа) и линий (ребра графа). Обычно, используется в связи с блоком «Конструктор графа» (п. 6.1.3.3). При этом, пользователь может самостоятельно подготовить данные для отображения.

Ребра графа могут быть как прямой линией, так и набором координат – например для дорог.

6.1.3.5.11.1 Структура объектов

Для отображения графа объекты на вход должны быть в определенной структуре.

На вход подается json внутри которого содержится два массива **nodes** и **edges**, вершины и ребра.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

```
{
  id: guid,
  nodes: [node:{}, node:{}]
 edges:[edge:{}, edge:{}],
}
```

Каждая вершина или ребро – это объект во внутренней структуре платформы, которые могут браться из блоков источников данных, но затем должны быть собраны через блок «Python» (п. 6.9.1).

Пример вершины (node)

```
{
 "pkey": {
  "id": 1  # уникальный идентификатор вершины
 },
 "_params": {
  "ts": [],
  "lazy": [],
  "key_field": "id",
  "name_field": "name"
 },
 "attrs": {
  "name": "Node 1", # наименование вершины, может быть выведено на слое
  "ord": 1,
  "coords": [  # координаты вершины
   0,
   0
  ]
 },
 "child": [],
 "_source": "0b362bb0-fa89-40ba-81d2-ae8003eb34e7"
}
```

Пример ребра (edge)

```
 {
 "pkey": {
  "id": 1  "id": 1  # уникальный идентификатор ребра
 },
 "_params": {
  "ts": [],
  "lazy": [],
  "key_field": "id",
  "name_field": "name"
 },
 "attrs": {
  "name": "Edge 1", # наименование ребра, может быть выведено на слое
  "ord": 1,
  "coords": [   # координаты ребра, может быть как линией из 2х точек, так и
сложной геометрии
   [
    0,
    0
   ],
   [
    1,
    1
   ],
```


© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

```
    [
     5,
     0
    ]
   ],
   "edge_direction": 0,        # направление движения по ребру
   "weight": 2,                # вес ребра
   "popup": "Edge 1",          # всплывающее окно при клике на граф
   "source_node_id": 1,        # идентификатор вершины из которой выходит ребро
   "destination_node_id": 2    # идентификатор вершины в которую входит граф
  }
}
```


© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Параметры слоя:

Название параметра	Тип параметра	Обязательный	Описание	Пример

Описание слоя (вершины)	JSON	Да	Стиль для отображения слоя с вершинами. Структура такая же, как у Обычного слоя, стиль согласно MapBox Style	**Пример настроек**
				`{`
				`"id": "graph_node_layer",`
				`"source": "graph_node_layer_source",`
				`"type": "circle",`
				`"paint": {`
				`"circle-radius": 20,`
				`"circle-color": "#FFFFFF",`
				`"circle-opacity": 0.9`
				`},`
				`"groupName": "Граф"`
				`}`

Описание слоя (ребра)	JSON	Да	Стиль для отображения слоя с ребрами. Структура такая же, как у Обычного слоя, стиль согласно MapBox Style	**Пример настроек**
				`{`
				`"id": "graph_edge_layer",`
				`"source": "graph_edge_layer_source",`
				`"type": "line",`
				`"onHover": true,`
				`"paint": {`
				`"line-width": 5,`
				`"line-color": [`
				`"case",`
				`["boolean", ["feature-state", "hover"], false],`
				`"#7137A1",`
				`"#fbb03b"`
				`],`
				`"line-opacity": [`
				`"case",`
				`["boolean", ["feature-state", "hover"], false],`
				`0.9,`
				`0.6`
				`]`

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример

				<pre>},
				"layout": {
				"line-cap": "butt",
				"line-join": "round"
				},
				"groupName": "Граф"
				}</pre>

С учетом геометрии	Флаг	Да	Отображение ребер графа с учетом их географических координат. По умолчанию включен, при выключении ребра будут рисоваться только по координатам вершин (прямая линия)	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.1.3.5.11.2 Особенности работы слоя:

Граф может быть построен пользователем или с помощью блока «Конструктор графа» (п. 6.1.3.3).

На выходе блока генерируется два слоя – точки и линии.

Оба слоя являются аналогичными «Обычному слою».

6.1.3.5.12 Гексагональная тепловая карта

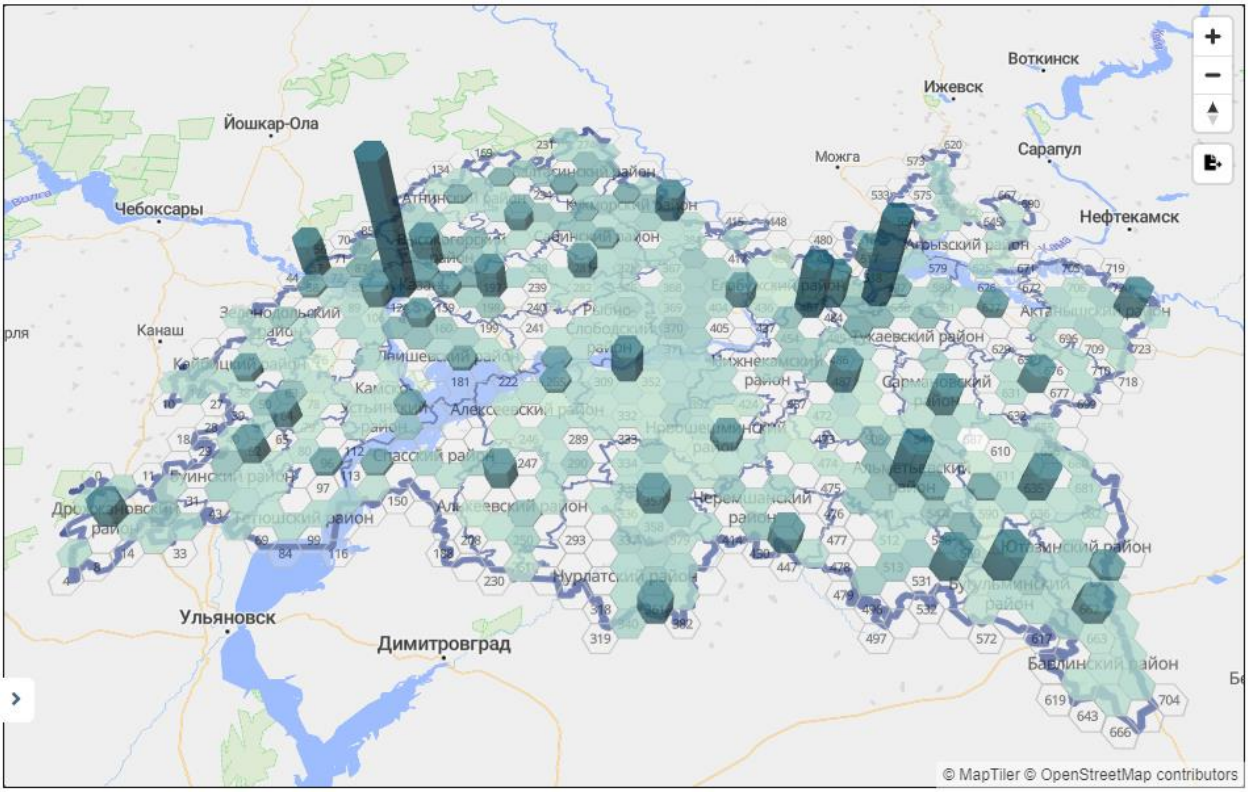


Рисунок 191 – Пример гексагональной 3D-тепловой карты

Назначение: размещение на карте данных по принципе гексагональной 3D-тепловой карты.

Описание: слой позволяет разместить на карте гексагональную сетку с 3D-столбиками. Слой базируется на слое hexagonLayer библиотеки deck.gl (https://deck.gl/examples/hexagon-layer/). Объекты передаются на слой в виде обычного объекта с координатами, затем блок превращает их в точки для отображения слоя.

Пример входных данных:

```
{
        "pkey": {},
        "_params": {
            "lazy": []
        },
        "attrs": {
            "id": "29500001",
            "address": "Пермь, Хабаровская, 171",
            "coords": "[56.08923912,58.04399496]"
        },
        "child": [],
        "_source": "21e00983-8bc8-4f65-8933-44e4b207ba6d"
}
```


© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Так же слой поддерживает добавления на карту элементов управления.

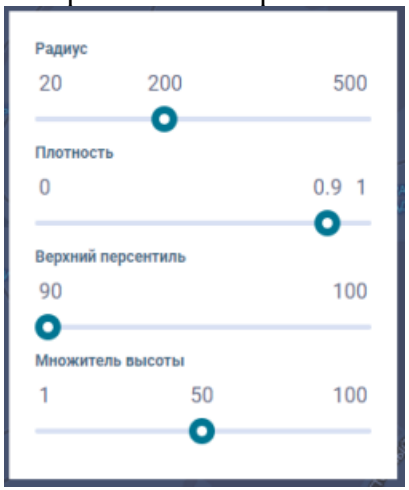
© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Параметры слоя:

Название параметра	Тип параметра	Обязательный	Описание	Пример

Описание слоя (гексагональная тепловая карта)	JSON	Да	Описание слоя HexagonLayer (https://deck.gl/examples/hexagon-layer/) для отрисовки гексагонов. Слой задает оформление для гексагонов и их поведение.
			Обязательные поля, которые должны быть указаны, но не входят в описание слоя в документации deck.gl
			• id – уникальный идентификатор слоя;
			• sourceId – уникальный идентификатор набора данных, которые будут переданы на карту для этого слоя;
			• groupName – имя группы в которую добавить слой для отображения его в списке слоев на карте.
				Пример описания слоя
				{
				"id": "hex_heatmap",
				"sourceId": "data_for_hex_heatmap",
				"radius": 200,
				"coverage": 0.9,
				"upperPercentile": 90,
				"colorRange": [
				[1, 152, 189],
				[73, 227, 206],
				[216, 254, 181],
				[254, 237, 177],
				[254, 173, 84],
				[209, 55, 78]
				],
				"elevationRange": [0, 100],
				"elevationScale": 50,
				"extruded": true,
				"opacity": 0.7,
				"pickable": true,
				"autoHighlight": true,
				"highlightColor": [253, 141, 60, 50],
				"transitions": {
				"elevationScale": {
				"duration": 1000
				},
				"coverage": {
				"duration": 1000
				}
				}
				}

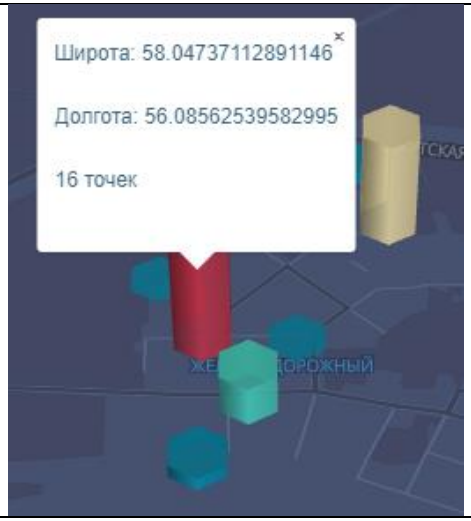
© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
Отображать элементы управления	Флаг	Да	Отображать или нет на карте элементы управления слоем. Далее идет список настроек для отображения на карте.  Вывод настроек подходит для подбора значений на карте наглядно	
Раздел настроек «Радиус»				
Минимальное значение, м	Целое число	Нет	Минимальное значение радиуса гексагона, задается в метрах	20
Максимальное значение, м	Целое число	Нет	Максимальное значение радиуса гексагона, задается в метрах	500
Наименование	Строка	Нет	Наименование раздела элемента управления	Радиус
Раздел настроек «Плотность»				
Минимальное значение	Вещественное число	Нет	Минимальное значение для плотности расположения гексагонов, задается в процентах	0
Максимальное значение	Вещественное число	Нет	Максимальное значение для плотности расположения, задается в процентах	1

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
Шаг	Вещественное число	Нет	Шаг с которым будет изменяться элемент управления	0.1
Наименование	Строка	Нет	Наименование раздела элемента управления	Плотность
Раздел настроек «Верхний персентиль»				
Минимальное значение	Целое число	Нет	Минимальное значение для верхнего персентиля, задается в процентах	90
Максимальное значение	Целое число	Нет	Максимальное значение для верхнего персентиля, задается в процентах	100
Шаг	Целое число	Нет	Шаг с которым будет изменяться элемент управления	1
Наименование	Строка	Нет	Наименование раздела элемента управления	Верхний персентиль
Раздел настроек «Верхний персентиль»				
Минимальное значение	Целое число	Нет	Минимальное значение для высоты полигонов, задается в процентах	1
Максимальное значение	Целое число	Нет	Максимальное значение для высоты полигонов, задается в процентах	100
Шаг	Целое число	Нет	Шаг с которым будет изменяться элемент управления	1
Наименование	Строка	Нет	Наименование раздела элемента управления	Множитель высоты
Выводить попап	Флаг	Да	Флаг отображать или нет всплывающее окно при нажатии на гексагон	
Текст попапа	Строка	Нет	По умолчанию во всплывающем окне выводится широта и долгота центра, а так же количество точек внутри. С помощью этого поля можно задать текст после количества точек.	точек

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
				

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.1.3.5.12.1 Особенности работы слоя:

- Все настройки элементов управления доступны на карте.
- Слой достаточно требователен к ресурсам компьютера.
- Может отображать достаточно большое количество точек.
- При высоте в «0» может выступать как гексагональная сетка.
- По слою недоступен поиск.

6.1.3.5.13 Внешний источник данных (геосервер)

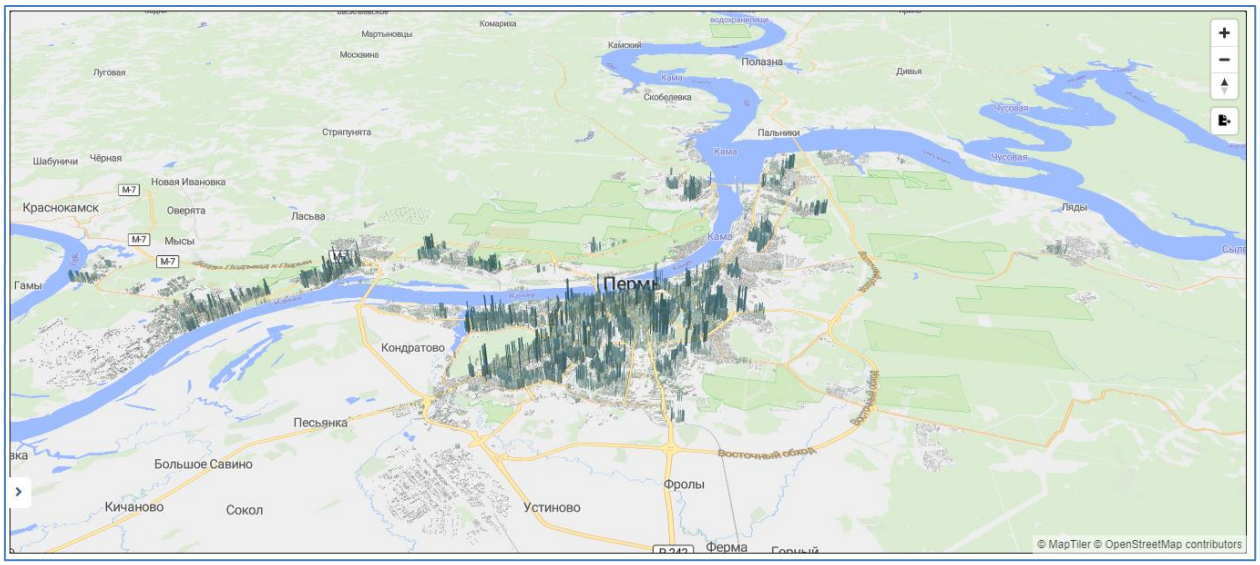


Рисунок 192 – Внешний источник данных (геосервер)

Слои с геосервера

При подключении нового источника на геосервере необходимо вызвать индексацию слоев геосервера, обратившись на его ендпоинт /index.json (для функциональных слоев – /rpc/index,json), например http://geoserver/index.json. Это сработает, если геосервер запущен в режиме автоматического подключения новых источников.

Важно!

Не подходит для использования с анимацией, т.к. анимация будет менять исходные данные слоя в БД! **Только для обновления слоя по анимации.**

Описание: специальной слой для упрощенного подключения векторных слоев с геосервера (например Геосервер Martin).

Назначение: Возможность подключения внешних слоев, через которые на карту можно грузить миллионы объектов как часть подложки. При этом этот слой будет отображаться в списке и корректно позволит показывать/скрывать себя.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Параметры слоя:

Название параметра	Тип параметра	Обязательный	Описание	Пример

Описание слоя	JSON	Да	Описание слоя, аналогично «Обычному слою»	**Пример настроек**
				`{`
				`"id": "buildings_region_layer",`
				`"source": "buildings_region",`
				`"source-layer": "test.mp_buildings_test",`
				`"type": "fill-extrusion",`
				`"paint": {`
				`"fill-extrusion-color": [`
				`"interpolate",`
				`["linear"],`
				`["get", "height"],`
				`50,`
				`"#F6FBF5",`
				`150,`
				`"#D4EBCF",`
				`300,`
				`"#A3CFC4",`
				`500,`
				`"#3E7C82",`
				`1000,`
				`"#003948"`
				`],`
				`"fill-extrusion-height": ["get", "height"],`
				`"fill-extrusion-opacity": 0.75`
				`},`
				`"groupName": "Фигуры",`
				`"onHover": true`
				`}`

Источник данных	JSON	Да	В этой настройке указывается откуда брать слой, его тип и обновлять ли его в случае применения к карте Блок «Анимация» (п. 6.1.4.4):	**Пример настроек**
				`{`
				`"type": "vector",`
				`"url": "http://stage.int.bittechno.ru:3000/test.mp_buildings_test.json",`
				`"animate": true,`

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример

			• type – тип слоя. Поддерживается только **vector**;
			• url – адрес с описанием откуда брать слой;
			• animate – обновлять ли слой на каждый такт анимации;
			• promoteId – атрибут для использования в качестве идентификатора (можно задавать при невозможности на стороне сервера передавать идентификатор объекта), для объектов без уникальных идентификатов невозможно использование состояний OnHover и OnClick
			
			Для обеспечения работы с внешним источником данных для слоя может понадобится передать в запросе на указанный адрес какие-либо заголовки или параметры входа. Для этого
				"promoteId": "id"
				}
				
				или
				
				{
				"type": "vector",
				"url":
				"http://stage.int.bittechno.ru:3000/test.mp_buildings_test.json",
				"host": ""http://stage.int.bittechno.ru:3000/",
				"headers": {
				"Parameter1": 123,
				"Parameter2": "Some string"
				}
				}
				
				или
				
				{
				"tileSize": 256,
				"tiles": [
				
				"http://192.168.88.240:8082/osm/{z}/{x}/{y}.png",
				
				"http://192.168.88.240:8082/osm_2/{z}/{x}/{y}.png"
				],
				"type": "raster",
				"headers": [
				{
				"X-Gravitee-Api-Key": "First host key"
				},
				{
				"X-Gravitee-Api-Key": "Second host key"
				}
				],
				"hosts": [

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример

			предусмотрены следующие дополнительные параметры:
			
			• host - строка, с которой начинается url, необходима для идентификации источника в списке источников для получения оттуда набора заголовков или параметров входа. В случае нескольких источников, имеющих host, значение этого параметра в разных источниках должно отличаться друг от друга и не должны являться подстрокой друг для друга, иначе источник может быть найден неверно и соответственно в запрос будут переданы заголовки другого источника.
			• headers - json-объект, содержащий пары ключ - значение, для
				<pre> "http://192.168.88.240:8082/osm/",
				        "http://192.168.88.240:8082/osm_2/"
				    ]
				}</pre>

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
			передачи в качестве заголовков запроса • credentials - дополнительное поле для передачи параметров входа(если необходимо) Для растровых слоев или других, где необходимо передавать перечень адресов для получения данных, параметры немного видоизменяются: • hosts - массив строк, являющихся началом для адресов, передаваемых для получения данных • headers - массив json-объектов для передачи в качестве заголовков, где индекс в массиве заголовков для передачи соответствует индексу хоста, по которому эту	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
			заголовки требуется передавать • credentials - массив параметров входа, также соответствующих хостам по индексу в массиве	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.1.3.5.13.1 Особенности работы слоя:

- По слою недоступен поиск.
- Слой должен быть сформирован как векторный или растровый и доступен для получения с сервера в локальной сети или сети Интернет.
- Для тестирования использовался геосервер Martin (https://github.com/urbica/martin).

6.1.3.5.13.2 Относительная адресация слоя:

Для слоя доступна относительная адресация. Для ее корректного функционирования необходимо указывать адрес /tileserver/имя_слоя.json (без кэша) или /tilecache/имя_слоя.json (для использования кэширования).

6.1.3.5.14 Геосервер Martin (Геосервер Martin)

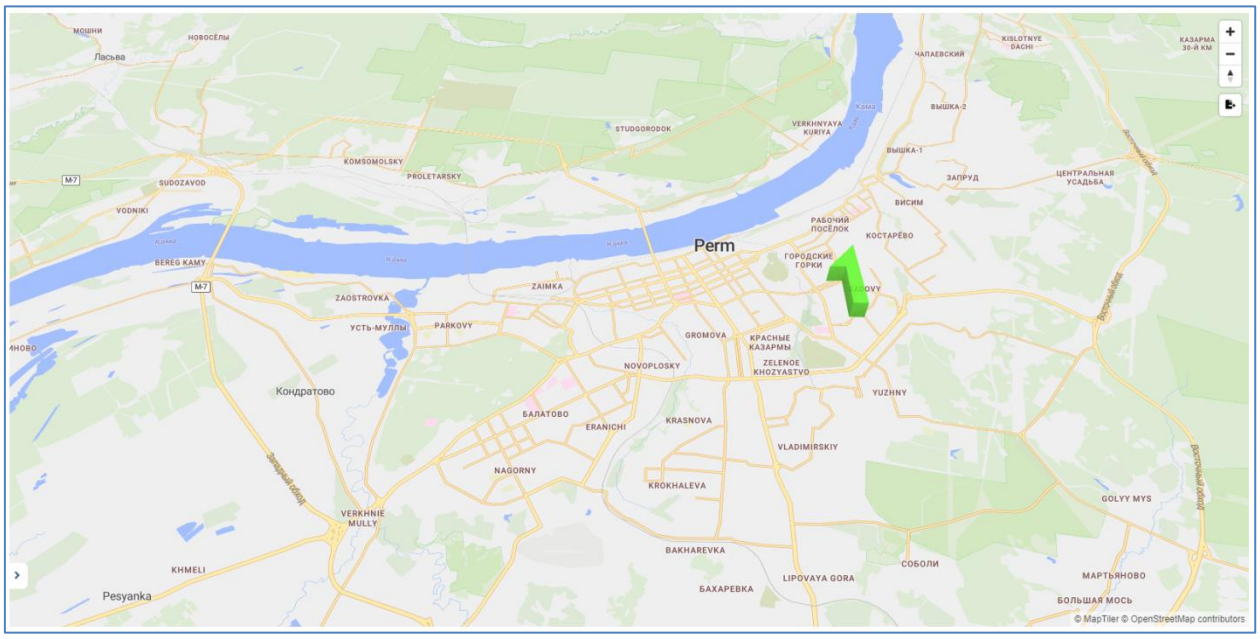


Рисунок 193 – Геосервер Martin (Геосервер Martin)

Слои с геосервера

При подключении нового источника на геосервере необходимо вызвать индексацию слоев геосервера, обратившись на его ендпоинт /index.json (для функциональных слоев – /rpc/index,json), например http://geoserver/rpc/index.json. Это сработает, если геосервер запущен в режиме автоматического подключения новых источников.

Важно!

Подходит для использования с анимацией

Описание: Специальной слой для упрощенного подключения векторных слоев с геосервера Martin.

Назначение: Возможность подключения внешних слоев, через которые на карту можно грузить миллионы объектов как часть подложки. При этом этот слой будет отображаться в списке и корректно позволит показывать/скрывать себя. В отличии от слоя «Внешний источник данных», этот слой работает только с геосервером Martin и позволяет динамически обновлять карту.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Параметры слоя:

Название параметра	Тип параметра	Обязательный	Описание	Пример

Описание слоя	JSON	Да	Описание слоя, аналогично «Обычному слою»	**Пример настроек**
				{
				"id": "mp_buildings_fill",
				"source": "buildings_region",
				"source-layer": "public.function_source",
				"type": "fill-extrusion",
				"minzoom": 8,
				"paint": {
				"fill-extrusion-color": [
				"interpolate",
				["linear"],
				["get", "height"],
				40,
				"#00FF00",
				2400,
				"#FFFF00",
				3600,
				"#FF0000"
				],
				"fill-extrusion-height": [
				"get",
				"height"
				],
				"fill-extrusion-base": 0,
				"fill-extrusion-opacity": 0.8
				},
				"groupName": "Население Пермского края",
				"editable": false,
				"visible": true
				}

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Источник данных	JSON	Да	В этой настройке указывается откуда брать слой и его тип.
			• type – тип слоя. Поддерживается только **vector**;
			• url – адрес с описанием откуда брать слой;
			• promoteId – атрибут для использования в качестве идентификатора (можно задавать при невозможности на стороне сервера передавать идентификатор объекта), для объектов без уникальных идентификатов невозможно использование состояний OnHover и OnClick
			Для обеспечения работы с внешним источником данных для слоя может понадобится передать в запросе на указанный адрес какие-либо заголовки или параметры входа. Для этого
				Пример настроек
				`{`
				`"type": "vector",`
				`"url": "http://dev.int.bittechno.ru:3000/rpc/test.animate_buildings_test_function.json",`
				`"promoteId": "id"`
				`}`

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
			предусмотрены следующие дополнительные параметры: • host - строка, с которой начинается url, необходима для идентификации источника в списке источников для получения оттуда набора заголовков или параметров входа. В случае нескольких источников, имеющих host, значение этого параметра в разных источниках должно отличаться друг от друга и не должны являться подстрокой друг для друга, иначе источник может быть найден неверно и соответственно в запрос будут переданы заголовки другого источника.	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
			• headers - json-объект, содержащий пары ключ - значение, для передачи в качестве заголовков запроса • credentials - дополнительное поле для передачи параметров входа(если необходимо)	
Динамический слой	Флаг	Да	Позволяет динамически обновлять данные карты, к примеру с помощью подключения к слоую карты блока «Анимация» (см. раздел «Особенности работы слоя»)	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.1.3.5.14.1 Особенности работы слоя:

- По слою недоступен поиск.
- Слой должен быть сформирован с использованием геосервера Martin (https://github.com/urbica/martin).
- Если включен флаг «Динамический слой» и на вход блока приходят динамические данные (к примеру, из блока «Анимация» (п. 6.1.4.4)), то слой будет изменяться каждый раз при получении новых данных:

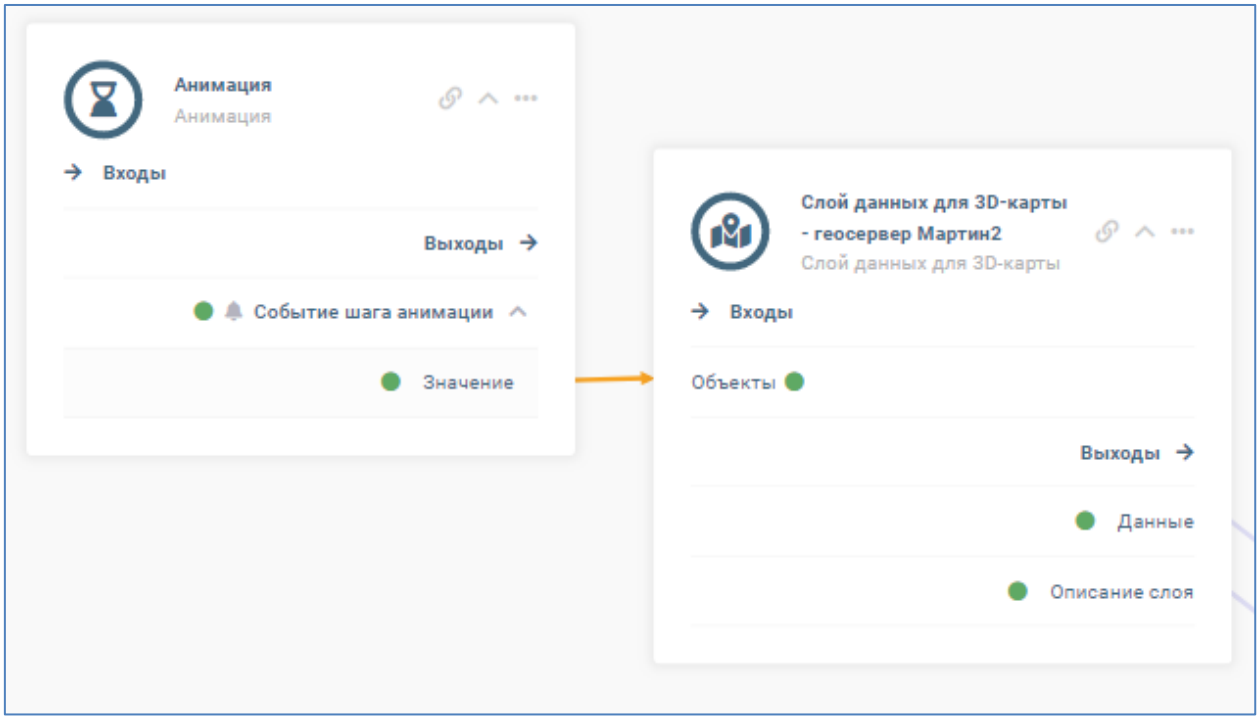


Рисунок 194 – Пример графа поступления динамических данных

- В качестве значения для обновления будет использован первый справочник, обнаруженный на входе блока.
- В массиве объектов этого справочника будет взят атрибут label первого элемента.
- В выходе блока «Данные» будет добавлен атрибут animation со значением, полученным выше.

Выход «Данные» с динамическим параметром

```
{
    "id": "buildings_region",
    "type": "externalDynamic",
    "data": {
        "type": "vector",
        "url":
"http://dev.int.bittechno.ru:3000/rpc/test.animate_buildings_test_function.js
on"
    },
    "animation": "2020",
    "_blockid": "ca209d32-c4ba-439e-9718-ea4f17c4744f"
}
```

Не рекомендуется подавать на вход более одного справочника, т.к. данные второго и всех последующих справочников не будут обработаны.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.1.3.5.14.2 Относительная адресация слоя

Для слоя доступна относительная адресация. Для ее корректного функционирования необходимо указывать адрес /tileserver/имя_слоя.json (без кэша) или /tilecache/имя_слоя.json (для использования кэширования).

6.1.3.5.15 Диаграмма (столбики)

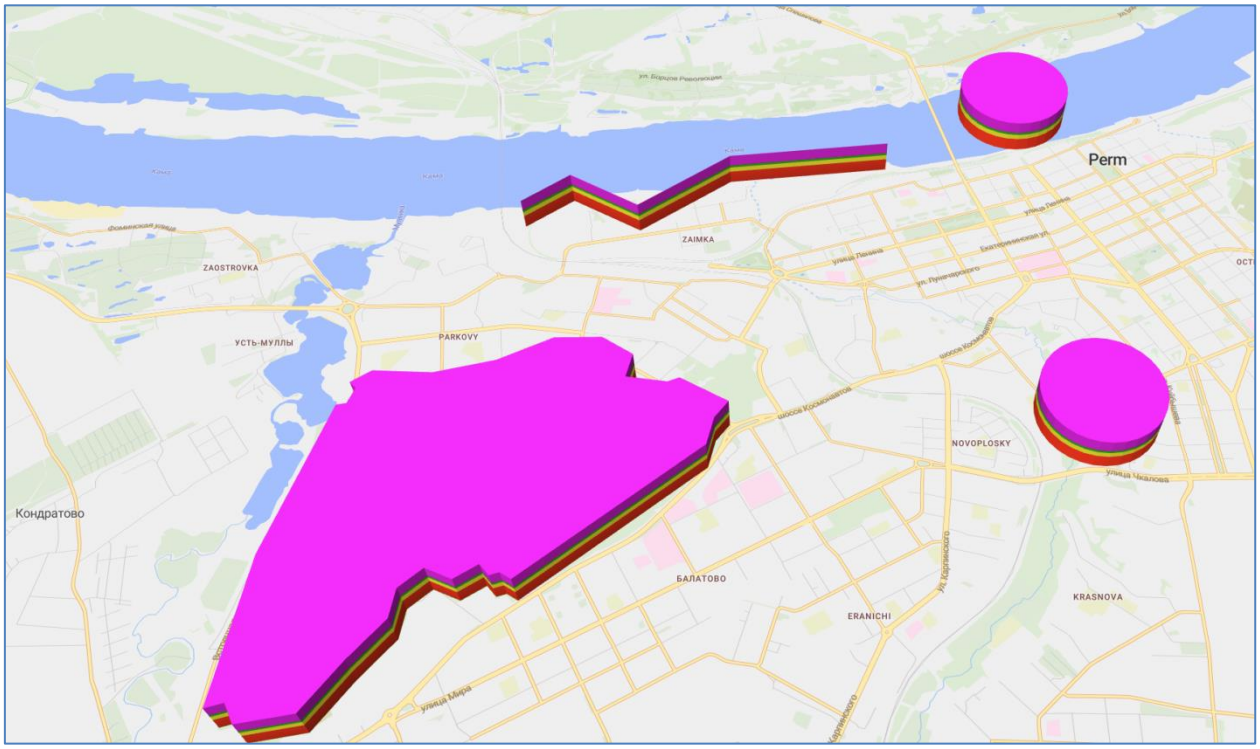


Рисунок 195 – Пример отображения объектов в виде столбиков диаграммы

Назначение: Для отображения на карте объектов заданной геометрии в виде столбиков диаграммы

Описание: позволяет отобразить объекты в виде столбиковой диаграммы. Доступен выбор типа геометрии объектов. При расчете блока с этим типом слоя будет подготовлен geojson и стиль для отрисовки слоя на карте.

Каждый цвет в столбике – это отдельный объект (координаты у объектов в одном столбике должны совпадать).

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Параметры слоя:

Название параметра	Тип параметра	Обязательный	Описание	Пример
Тип геометрии	Выпадающий список	Да	Определяет какой тип геометрии будет использоваться в слое. От него зависит и стиль для слоя	

Описание слоя	JSON	Да	Описание слоя согласно стилю Mapbox для отображения на карте. Дополнительные поля, которые должны быть указаны:
			1. Id – уникальный идентификатор слоя, так же служит для связи с другими механизмами карты, например иерархией слоев.
			2. Source – уникальный идентификатор, который будет назначен geojson с данными для карты, который будет получен из блока.
				Пример настроек
				`{`
				`"id": "chart_poly_layer_1",`
				`"source": "chart_poly_layer_1",`
				`"type": "fill-extrusion",`
				`"onHover": true,`
				`"onClick": false,`
				`"minzoom": 7,`
				`"paint": {`
				`"fill-extrusion-color": ["get", "color"],`
				`"fill-extrusion-height": [`
				`"interpolate",`
				`["linear"],`
				`["zoom"],`
				`7, ["*", ["get", "height"], 10],`
				`14, ["get", "height"]`
				`],`
				`"fill-extrusion-base": [`
				`"interpolate",`
				`["linear"],`
				`["zoom"],`
				`7, ["*", ["get", "base"], 10],`
				`14, ["get", "base"]`
				`],`

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример

				<pre>"fill-extrusion-vertical-gradient": true,
				"fill-extrusion-opacity": 1
				},
				"groupName": "Диаграмма",
				"visible": true
				}</pre>

Радиус	Целое число	Да	Для точек: радиус (в метрах) окружности столбика диаграмм. Для линий: толщина (в метрах) линии	
Атрибуты для передачи на слой	JSON	Нет (допустим пустой список)	Список атрибутов, которые будут переданы на слой карты	<pre>["color", "height", "base"]</pre>

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.1.3.5.15.1 Форматирование столбиков

Для корректного форматирования столбиков, у объекта необходимо передать три атрибута на слой:

- **color** – цвет столбца;
- **base** – начальная высота (целое не отрицательное число);
- **height** – максимальная высота (начиная от 0).

Включает в себя значение base: если у объекта base: 20 и необходимо сделать объект высотой 30, то значение height должно быть равным 50.

6.1.3.5.16 Диаграмма (подписи)

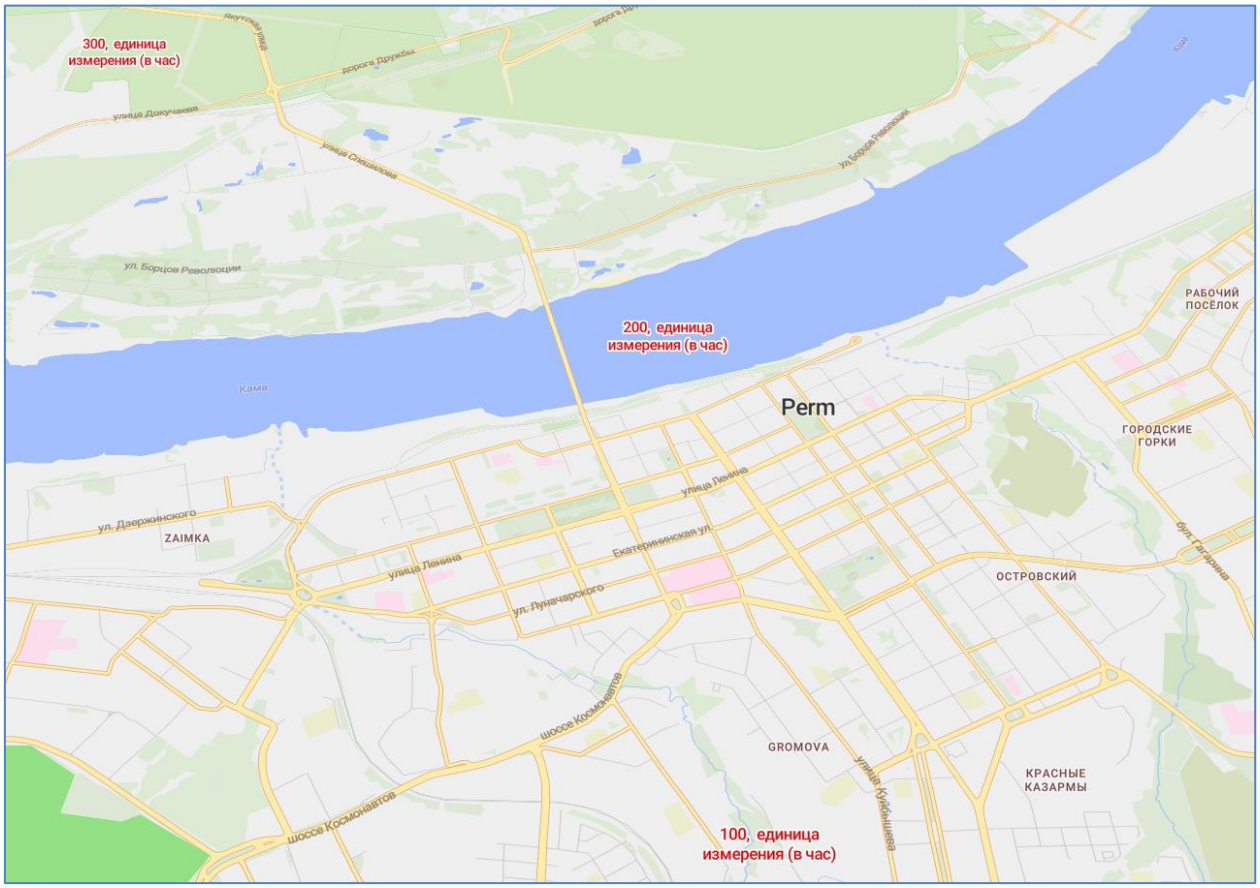


Рисунок 196 – Пример отображения объектов заданной геометрии в виде подписей диаграммы

Назначение: Для отображения на карте объектов заданной геометрии в виде подписей диаграммы.

Описание: позволяет отобразить объекты в виде подписей для диаграммы с конкретным значением для каждого объекта. Доступен выбор типа геометрии объектов. При расчете блока с этим типом слоя будет подготовлен geojson и стиль для отрисовки слоя на карте.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Параметры слоя:

Название параметра	Тип параметра	Обязательный	Описание	Пример
Тип геометрии	Выпадающий список	Да	Определяет какой тип геометрии будет использоваться в слое. От него зависит и стиль для слоя	
Описание слоя	JSON	Да	Описание слоя согласно стилю Mapbox для отображения на карте. Дополнительные поля, которые должны быть указаны 1. Id – уникальный идентификатор слоя, так же служит для связи с другими механизмами карты, например иерархией слоев. 2. Source – уникальный идентификатор, который будет назначен geojson с данными для карты, который будет получен из блока	**Пример настроек** `{ "id": "chart_point_layer_2", "source": "chart_point_layer_2", "type": "symbol", "layout": { "symbol-placement": "point", "text-font": ["Roboto Medium"], "text-field": ["concat", ["get", "value"], " ", единица измерения (в час)"], "text-size": ["interpolate", ["linear"], ["zoom"], 7, 12, 14, 16], "text-allow-overlap": false }, "paint": { "text-translate": [0, 12], "text-color": "#e00000", "text-halo-color": "white", "text-halo-width": 2 },`

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
				"groupName": "Диаграмма", "visible": true }
Атрибуты для передачи на слой	JSON	Нет (допустим пустой список)	Список атрибутов, которые будут переданы на слой карты	["value"]

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.1.3.5.16.1 Форматирование подписей

Для корректного форматирования подписей, у объекта необходимо передать на слой атрибут **value**.

6.1.3.5.17 Описание работы визуализатора

Блок не является визуализатором, однако в режиме отладки графа доступен просмотр карты с этим слоем.

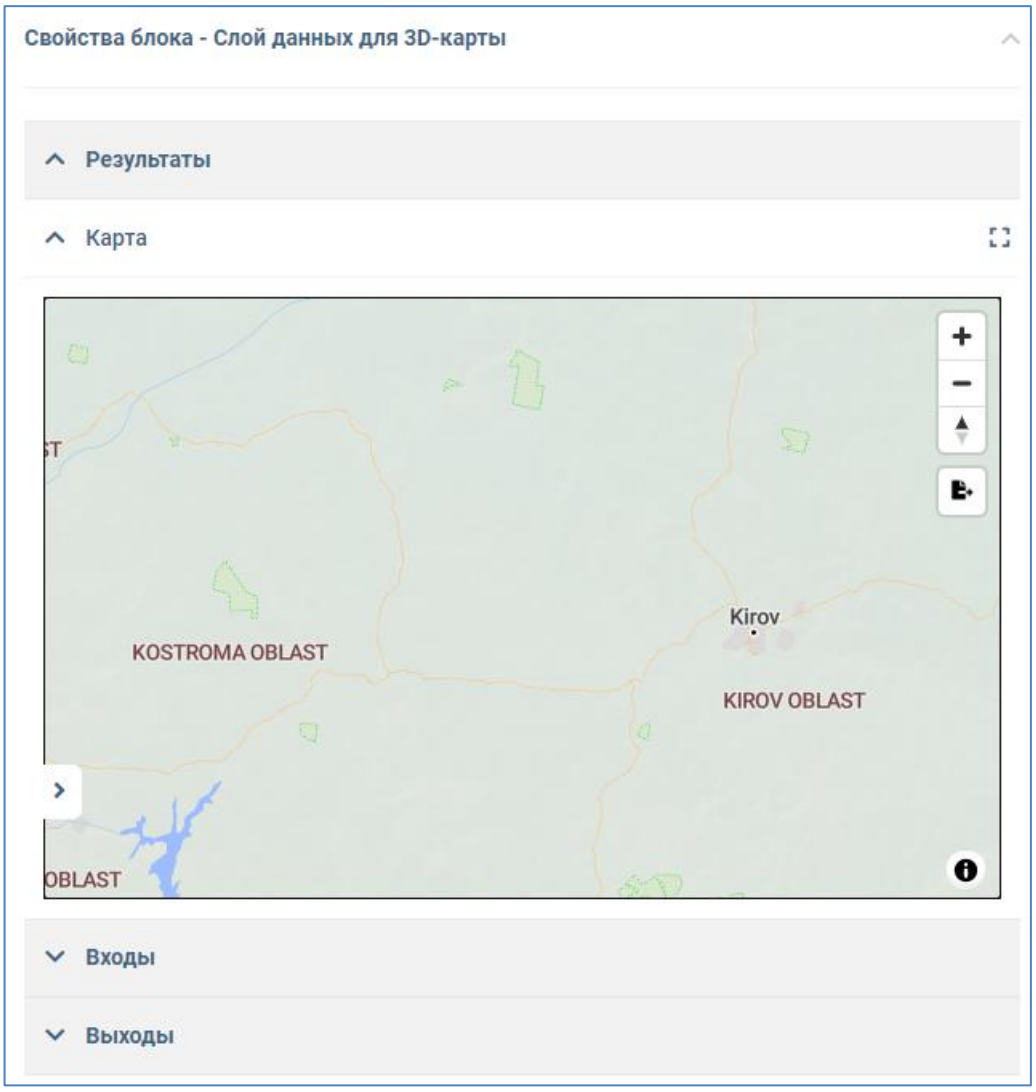


Рисунок 197 – Внешний вид визуализатора «Свойства блока – Слой данных для 3D-моделей»

6.1.3.5.17.1 Настройки визуализатора

Блок не является визуализатором

6.1.3.5.17.2 Дополнительные ссылки

Документация по стилям MapBox Layer Style

Документация по слоям библиотеки deck.gl

Визуальный редактор стилей Maputnik

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.1.3.6 Блок «Кластеризация данных для 3D-Карты»

6.1.3.6.1 Назначение

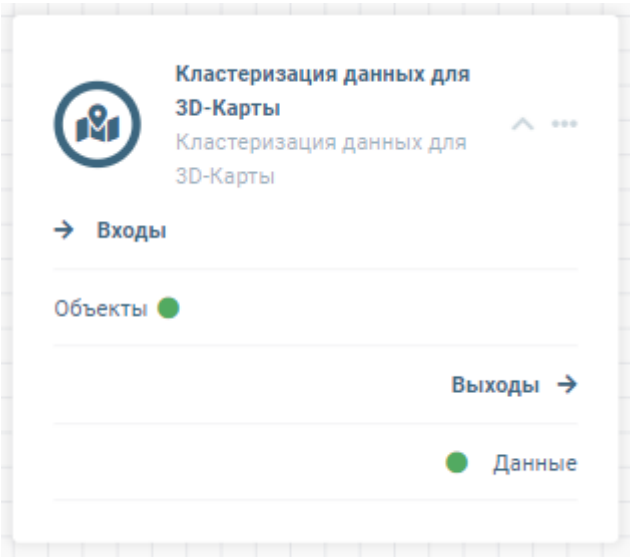


Рисунок 198 – Блок «Кластеризация данных для 3D-Карты»

Блок позволяет создать кластерный слой для отображения на 3д карте (Блок "3D-карта"). Данные со входа преобразуются в geojson формат, который быть передан в Блок "3D-карта"

Название параметра	Тип параметра	Обязательный	Описание	Пример
Наименование слоя	Строка	Нет	Название слоя для отображения в дереве слоев. Может отсутствовать или задаваться в описании слоя, см. детальное описание каждого слоя	

Описание слоя	JSON	Да	Описание слоя согласно стилю Mapbox для отображения на карте. Дополнительные поля, которые должны быть указаны.
			Id - уникальный идентификатор слоя, так же служит для связи с другими механизмами
				{
				"id": "squares_aero_fill_layer",
				"source": "squares_aero_source",
				"type": "geojson",
				"cluster": true,
				"clusterMaxZoom": 14,
				"clusterRadius": 50,
				}

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
			карты, например иерархией слоев. Source - уникальный идентификатор, который будет назначен geojson с данными для карты, который будет получен из блока. • type - тип геометрии слоя согласно спецификации стилей.	

6.1.3.6.2 Группа «Общие настройки блока»

Аналогично приведенным в п. 6.1.1.1.3.

6.1.3.6.3 Входы

Название входа	Тип	Обязательный	Описание
Объекты	Объекты	Нет	Массив данных с типом "Ряд данных" или "Объект"

Входы задаются пользователем при создании блока. Для создания входа необходимо нажать на знак «+» в разделе входов блока.

6.1.3.6.4 Выходы

Название выхода	Тип	Описание
Данные	JSON	Описаны слоя для размещения на карте в виде json (объекты представляются в виде geojson)

6.1.3.6.5 События блока

В блоке нет событий.

6.1.3.6.6 Описание работы блока

Блок позволяет сформировать структуру данных, которую можно разместить на карте. Т.к. карта является достаточно специфическим визуализатором с большим количеством внутренних механизмов работы - данные для нее готовятся отдельным блоком (а для некоторых слоев - набором блоков). На выходе данные преобразуются в geojson согласно его спецификации и набора метаданных слоя. В метаданных основными элементами является стиль отрисовки слоя картой.

Для того, чтобы на карте объекты объединялись в группы (кластеры) необходимо настроить "Описание слоя".

Блок работает так же как и блок "Слой данных для 3D карты".

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.1.3.6.7 Описание работы визуализатора

Блок не является визуализатором.

6.1.3.6.8 Настройки визуализатора

Блок не является визуализатором.

6.1.4 Элементы управления

6.1.4.1 Блок «HTML-статика»

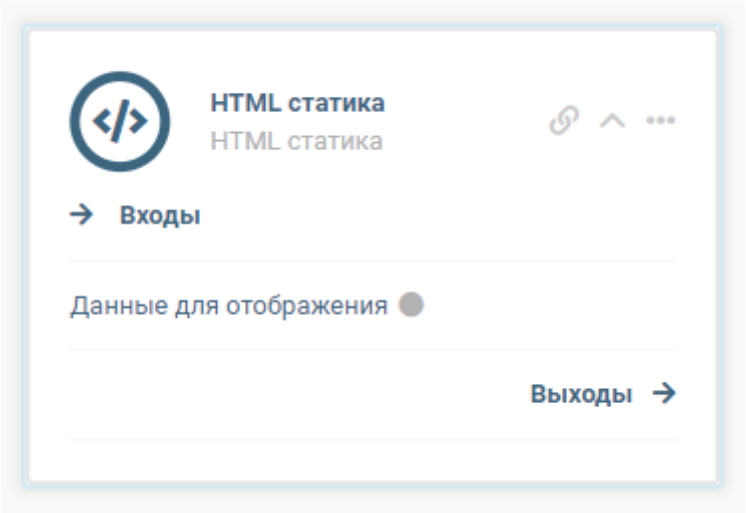


Рисунок 199 – Пример блока «HTML-статика»

6.1.4.1.1 Назначение

Блок позволяет формировать статичный визуализатор HTML. Внутри визуализатора пользователь может размещать произвольную html верстку с использованием css стилей.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.1.4.1.2 Настройки блока

Название параметра	Тип параметра	Обязательный	Описание	Пример
Содержимое HTML	Строка (HTML)	Нет	Код html который будет отображен в визуализаторе. Может включать в себя стили css	Пример HTML кода <style> .my_h1 {color:red;} .my_p {color:blue;} </style> <h1 class="my_h1">A heading</h1> A paragraph. Диаграмма 3
Разрешать копировать текст	Флаг	Да	Определяет разрешать ли копировать содержимое в визуализаторе при отображении пользователю	
Строгая валидация HTML	Флаг	Нет	Включает строгую валидацию HTML – в случае ошибки валидации блок выдаст ошибку. По умолчанию – будет выдано предупреждение	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.1.4.1.3 Группа «Общие настройки блока»

Аналогично приведенным в п. 6.1.1.1.3.

6.1.4.1.4 Входы

Название входа	Тип	Обязательный	Описание
Данные для отображения	Строка (HTML)	Нет	HTML код в виде строки, который необходимо отобразить в визуализаторе. При передаче данных на этот вход содержимое поля настроек «Содержимое HTML» игнорируется при расчете. Не может содержать шаблоны Jinja. Для шаблонов используется Блок «HTML-шаблон» (п. 6.1.4.2)

6.1.4.1.5 Выходы

Название выхода	Тип	Обязательный	Описание
HTML (Строка)	Строка	Нет	HTML код в виде строки, аналогичный тому, что будет отображен в визуализаторе. Если на вход пришел массив с несколькими HTML, то в выходе будет отображен только первый из них

6.1.4.1.6 События блока

В блоке нет событий.

6.1.4.1.7 Описание работы блока

Блок «HTML-статика» предназначен для размещения в пользовательском приложении произвольной html верстки. Верстка может быть задана статичной прямо в блоке, а может быть передана на вход блока «Данные для отображения». В случае, если верстка передана через вход – содержимое настройки блока будет игнорироваться, даже если заполнено. В html коде можно использовать стили, но разработчику необходимо следить за тем, какие стили к каким тегам применяются. Например, если переопределить стиль границы всех таблиц, указав border-style – будут изменены все таблицы на странице, т.к. браузер будет воспринимать этот тег как часть общего стиля страницы.

После расчета блока пользовательский html в визуализатор передается в неизменном виде. Поддерживаются стандартные конструкции, а так же вставка media содержимого.

За счет входа, на который можно передать статичный html код, полученный, например, в блоке Python – можно реализовывать генерацию содержимого «на лету» – через события в других блоках или при открытии страницы пользователем.

Дополнительно, в html коде можно указывать гиперссылки, при нажатии на которые будет осуществлен переход на выбранный визуализатор. Для этого надо указать специальный тег **goto** и идентификатор (GUID) визуализатора. Идентификатор можно скопировать через меню в режиме редактирования. Это имеет смысл для случаев, когда целевой визуализатор расположен на другой вкладке в Панели закладок или визуализаторе Аккордеон (см. п. 6.1.6.1) и не виден пользователю.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Пример ссылки на визуализатор:

```
<ul>
    <a goto="0722f50b-ea85-440c-8337-e9a399e96d21">Диаграмма 1</a>
</ul>
<ul>
    <a goto="02496f8b-baf1-4ec0-8be9-45314dd843fe">Диаграмма 2</a>
</ul>
<ul>
    <a goto="b65e11ba-fa0a-450e-8e43-a5d1e34183a3">Диаграмма 3</a>
</ul>
```

Валидация HTML

В блоке расположен валидатор HTML и санитайзер, которые в случае некорректного HTML на входе выдаст сответствующие ошибки (код ошибки 141 для валидатора и код ошибки 152 для санитайзера).

Однако, в связи со сложной структурой HTML, далеко не все ошибки могут быть обнаружены. К примеру, встроенный валидатор не проверяет наличие закрывающих тегов. Для более полной и корректной проверки рекомендуется использовать сторонние сервисы, к примеру validator.w3.org.

Подробнее о санитайзере приведено в п. 4.5.1.

6.1.4.1.8 Описание работы визуализатора

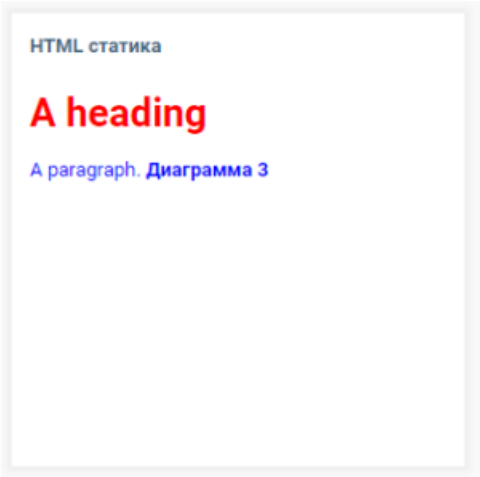


Рисунок 200 – Внешний вид визуализатора «HTML-статика»

Визуализатор представляет собой контейнер для отображения пользовательского html.

Визуализатор доступен в режиме отладки графа.

Визуализатор поддерживает выделение содержимого мышью.

6.1.4.1.9 Настройки визуализатора

Визуализатор не имеет специфических настроек. Общий список настроек приведен в п. 4.2.8 «Настройки визуализатора».

6.1.4.1.10 Дополнительно

Блок не поддерживает шаблонизацию.

6.1.4.1.11 Дополнительные ссылки

Онлайн валидатор **HTML:** validator.w3.org

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.1.4.2 Блок «HTML-шаблон»

6.1.4.2.1 Назначение

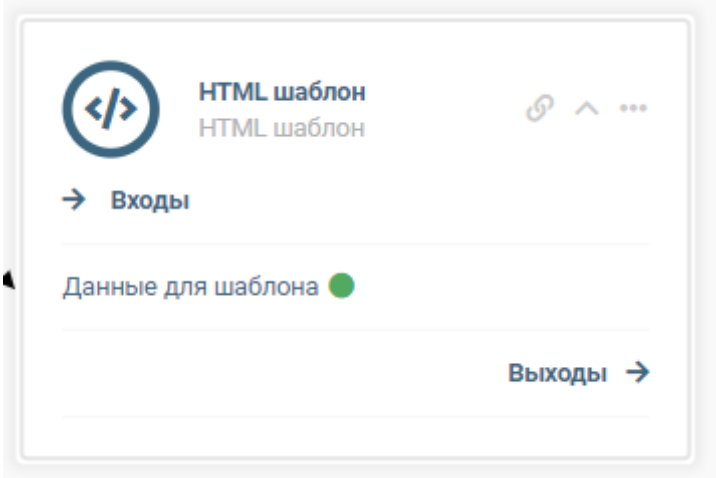


Рисунок 201 – Пример блока «HTML-шаблон»

Блок позволяет формировать динамически измененяемый визуализатор HTML. Блок содержит тот же функционал что и Блок «HTML-статика», но позволяет изменять свое содержимое с помощью шаблонизатора **Jinja2**.

6.1.4.2.2 Настройки блока

Название параметра	Тип параметра	Обязательный	Описание	Пример

Шаблон	Строка (HTML)	Нет	Код html который будет отображен в визуализаторе. Может включать в себя стили css и Jinja шаблоны	HTML с Jinja шаблоном
				`<title>Jinja шаблон</title>`
				`<ul>`
				`{% for k in input.vl%}`
				`<li>{{ k }}</li>`
				`{% endfor %}`
				`</ul>`

Разрешать копировать текст	Флаг	Да	Определяет разрешать ли копировать содержимое в визуализаторе при отображении пользователю	
Строгая валидация HTML	Флаг	Нет	Включает строгую валидацию HTML – в случае ошибки валидации блок выдаст ошибку. По умолчанию – будет выдано предупреждение	

6.1.4.2.3 Группа «Общие настройки блока»

Аналогично приведенным в п. 6.1.1.1.3.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.1.4.2.4 Входы

Название входа	Тип	Обязательный	Описание
Данные для шаблона	JSON	Нет	Данные, которые необходимы для размещения в html коде с помощью шаблонизатора Jinja, см. Описание работы блока

6.1.4.2.5 Выходы

Название выхода	Тип	Обязательный	Описание
HTML (Строка)	Строка	Нет	HTML код в виде строки, аналогичный тому, что будет отображен в визуализаторе. Если код был с использованием Jinja шаблона, то на выходе код будет уже с примененным шаблоном

6.1.4.2.6 События блока

В блоке нет событий.

6.1.4.2.7 Описание работы блока

Возможности блока «HTML-шаблон» полностью аналогичны блоку «HTML-статика» (п. 6.1.4.1), но за счет использования шаблонизатора Jinja блок может динамически менять свое содержимое. Шаблонизатор позволяет взять данные, которые пришли на вход «Данные для шаблона» и разместить их в коде html.

Шаблон Jinja является довольно распространённым способом генерации статических HTML страниц, например с его помощью работает Django. Для применения шаблона необходимо понимать структуру данных, которые пришли на вход. Данные всегда json к полям которого можно обратиться используя шаблон.

Важно!

Корневой элемент всегда должен называться **input**

Ниже представлен пример работы шаблонизатора приведен ниже на рисунке.

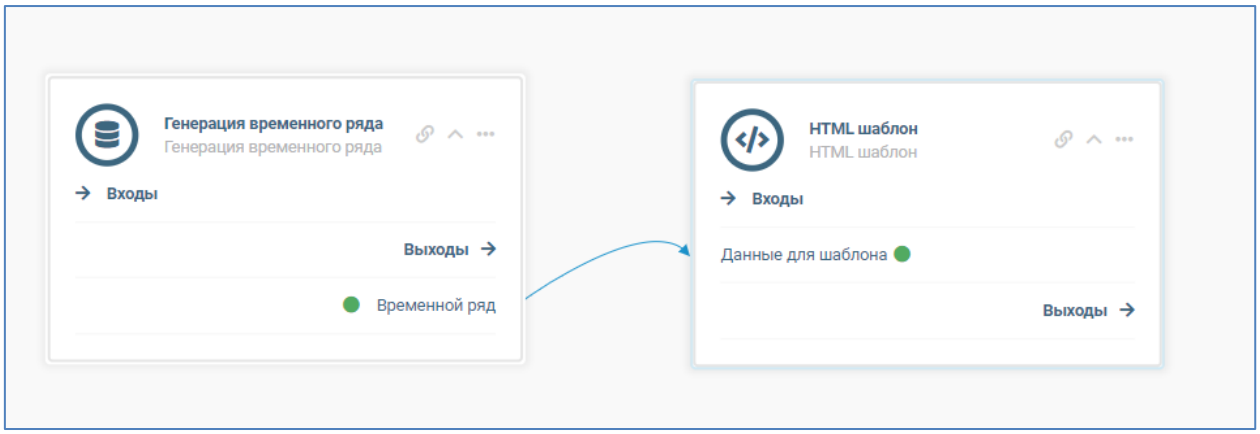


Рисунок 202 – Пример работы шаблонизатора

1. В графе генерируется временной ряд (Блок «Генерация временного ряда» п. 6.2.13).
2. После этого в шаблоне формируется список из всех значений ряда. При этом, на момент создания графа, неизвестно количество элементов – поэтому используем шаблон.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

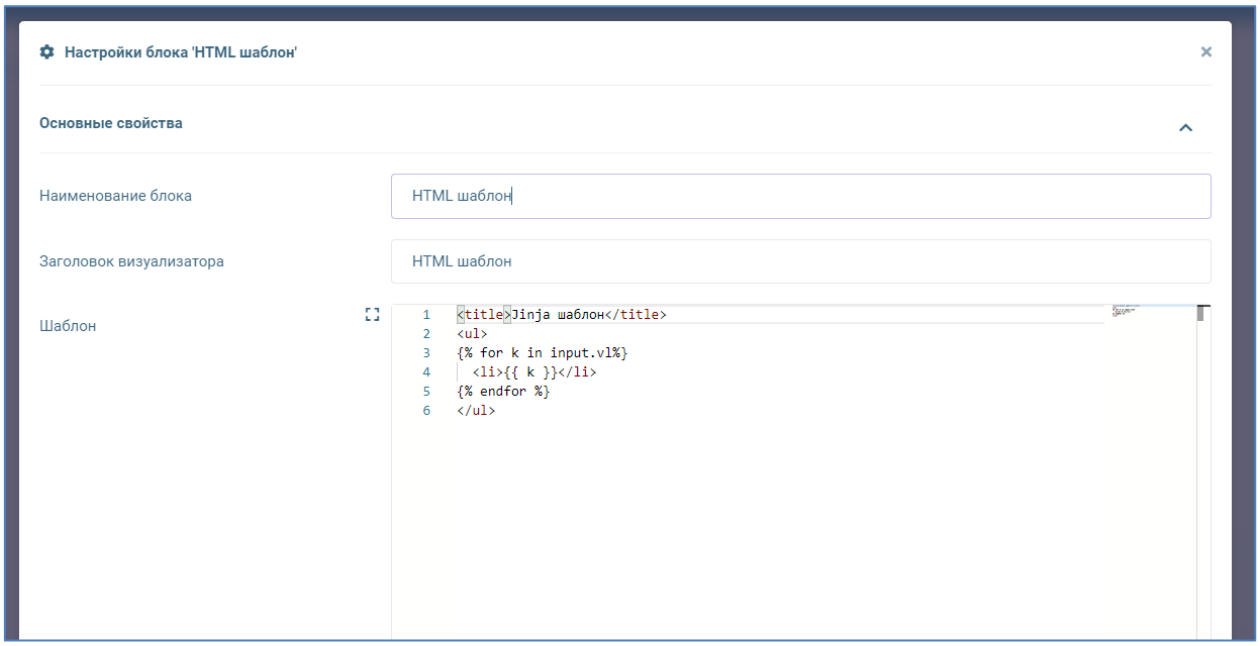


Рисунок 203 – Настройки блока «HTML-шаблон»

HTML с Jinja шаблоном

```
<title>Jinja шаблон</title>
<ul>
{% for k in input.vl%}
  <li>{{ k }}</li>
{% endfor %}
</ul>
```

В шаблоне из входа, который **всегда должен назваться input**, берется массив vl значений ряда. Структуру входных данных можно посмотреть в режиме отладки графа, на входе «Данные для шаблона» блока.

После пересчета графа в пользовательском приложении будет отображен сформированный html код в визуализаторе. Из всех значений ряда, поступивших на вход, будет сформирован линейный список.

Механизм шаблонов позволяет достаточно гибко управлять содержимым html шаблона, т.к. поддерживает основные конструкции: if, for, while

Валидация HTML

В блоке расположен валидатор HTML и санитайзер, которые в случае некорректного HTML на входе выдаст сответствующие ошибки (код ошибки 141 для валидатора и код ошибки 152 для санитайзера).

Однако, в связи со сложной структурой HTML, далеко не все ошибки могут быть обнаружены. К примеру, встроенный валидатор не проверяет наличие закрывающих тегов. Для более полной и корректной проверки рекомендуется использовать сторонние сервисы, к примеру validator.w3.org.

Подробнее о санитайзере приведено в п. 4.5.1.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.1.4.2.8 Описание работы визуализатора

Рисунок 204 – Пример HTML-шаблона

Визуализатор представляет собой контейнер для отображения пользовательского html с использованием шаблонов Jinja.

Визуализатор доступен в режиме отладки графа.

Визуализатор поддерживает выделение содержимого мышью.

6.1.4.2.9 Настройки визуализатора

Визуализатор не имеет специфических настроек. Общий список настроек приведен в п. 4.2.8 «Настройки визуализатора».

6.1.4.2.10 Дополнительно

При работе с данными из других блоков необходимо знать внутреннюю структуру приходящих на вход данных для корректного отображения с использованием шаблона

6.1.4.2.11 Дополнительные ссылки

Документация **Jinja:** https://jinja.palletsprojects.com/en/3.0.x/

Онлайн валидатор **HTML:** validator.w3.org

6.1.4.3 Блок «Telegram»

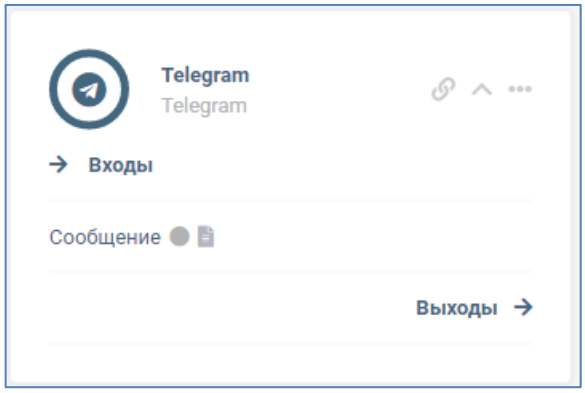


Рисунок 205 – Пример блока «Telegram»

6.1.4.3.1 Назначение

Блок позволяет формировать и отправлять сообщения в Telegram.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.1.4.3.2 Настройки блока

Название параметра	Тип параметра	Обязательный	Описание	Пример
API Token	Шифрованная строка	Да	Токен бота, через которой будет осуществляться транслирование сообщений	
Идентификатор чата	Строка	Да	Идентификатор чата или идентификатор пользователя, которому необходимо отправлять сообщения	
Разметка	Выпадающий список	Да	Стиль разметки сообщения. Возможные варианты: Текст – простой текст без разметки. Markdown – разметка markdown HTML – разметка HTML При выборе стиля разметки изменяется поле для ввода сообщения. Информация при этом не переносится и остается в предыдущем поле ввода	
Сообщенеие (Текст)	Строка	Нет	Сообщение для отправки в Telegram. Без разметки	
Сообщение (Markdown)	Строка (markdown)	Нет	Сообщение для отправки в Telegram. Разметка markdown	
Сообщение (HTML)	Строка (HTML)	Нет	Сообщение для отправки в Telegram. Разметка HTML	
Отключить уведомления о сообщениях	Флажок	Нет	При включеном параметре, сообщение придет в чат без звукового уведомления	

6.1.4.3.3 Группа «Общие настройки блока»

Аналогично приведенным в п. 6.1.1.1.3.

6.1.4.3.4 Входы

Название входа	Тип	Обязательный	Описание
Сообщение	Строка	Нет	Текст сообщения для транслирования Вход имеет больший приоритет, чем настройки блока. При наличии данных на входе, текст будет браться из него. Настройки параметра «**Разметка**» будут применяться к значению на входе

6.1.4.3.5 Выходы

В блоке нет выходов.

6.1.4.3.6 События блока

В блоке нет событий.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.1.4.3.7 Описание работы блока

Для работы с блоком необходимо зарегистрировать нового бота с помощью BotFather. После чего указать полученный токен в поле **API Token**.

При каждом расчете блока, бот будет транслировать сообщения в чат с указанным идентификатором.

6.1.4.3.8 Дополнительные ссылки

https://www.markdownguide.org/basic-syntax/

https://ru.botostore.com/c/myidbot/ – узнать идентификатор чата или пользователя в Telegram.

6.1.4.4 Блок «Анимация»

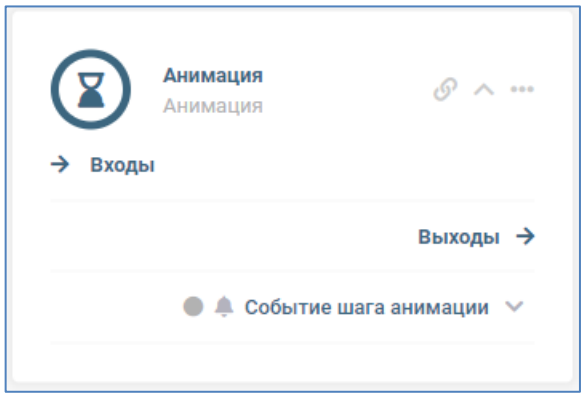


Рисунок 206 – Пример блока «Анимация»

6.1.4.4.1 Назначение

Блок позволяет формировать визуализатор анимации, с помощью которого могут анимироваться другие визуализаторы.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.1.4.4.2 Настройки блока

Название параметра	Тип параметра	Обязательный	Описание	Пример
Вид отображения	Выпадающий список	Да	Устарел и не отображается в версии блока 3 и выше	
Тип делений анимации	Выпадающий список	Да	Определяет какие точки будут в анимации, а также список настроек блока. • По датам. Точки анимации могут быть заполнены блоком автоматически, если указаны необходимые параметры (Дата начала, Дата окончания, Динамика) • По значениям. Точки анимации могут быть любым строковым значением, все точки должны быть указаны в параметрах (Значения для анимации)	
Дата начала	Строка	Да	Параметр доступен, если выбран тип делений анимации «По датам». Дата начала отображения анимации. Задается датой в формате, указанном в маске. Формат дат должен быть одинаковым во всех настройках блока	01.01.2010
Дата окончания	Строка	Да	Параметр доступен, если выбран тип делений анимации «По датам». Дата окончания отображения анимации. Задается датой в формате, указанном в маске	01.04.2021

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
Динамика	Выпадающий список	Да	Параметр доступен, если выбран тип делений анимации «По датам». Динамика анимации. От этого параметра зависит количество точек в анимации. Например, если выбрана годовая динамика, то расстояние между точками анимации будет равно одному году • Годовая • Месячная • Дневная • Часовая • Минутная • Секундная	
Интервал	Целое число	Нет	Параметр доступен, если выбран тип делений анимации «По датам». Количество точек в указанной динамике, которые будут объединены в один шаг. Например, если выбрана годовая динамика и интервал = 3, то расстояние между точками анимации будет равно трем годам	
Маска	Строка	Нет	Параметр доступен, если выбран тип делений анимации «По датам». Формат даты	
Даты анимации	Строка	Нет	Параметр доступен, если выбран тип делений анимации «По датам» Точные даты, по которым будет происходить анимация. Указываются в формате, указанном в маске, с разделителем в виде точки с запятой (;), без пробелов	01.01.2015;01.01.2016;01.01.2019

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
Даты для отображения	Строка	Нет	Параметр доступен, если выбран тип делений анимации «По датам» Даты, которые будут отображаться в визуализаторе. Указываются в формате, указанном в маске, с разделителем в виде точки с запятой (;), без пробелов	01.01.2015;01.01.2016;01.01.2019
Значения для анимации	Строка	Нет	Параметр доступен, если выбран тип делений анимации «По значениям» Значения, по которым будет происходить анимация. Указываются с разделителем в виде точки с запятой (;), без пробелов	Луховицкий район; Талдомский район; Шатурский район; городской округ Дзержинский
Значения для отображения	Строка	Нет	Параметр доступен, если выбран тип делений анимации «По значениям» Значения, которые будут отображаться в визуализаторе. Указываются с разделителем в виде точки с запятой (;), без пробелов	Луховицкий район; Талдомский район; Шатурский район; городской округ Дзержинский
Отображать промежуточные шаги анимации	Флаг	Нет	Отображает точки, по которым происходит анимация, но для которых не отображаются подписи	
Задержка шага анимации (мс)	Целое число	Нет	Время, которое будет затрачено на анимацию одного шага, указывается в миллисекундах	
Рассчитывать и кэшировать результат	Флаг	Нет	Все шаги анимации будут рассчитаны и сохранены при расчете графа	
Автоматическая анимация при открытии	Флаг	Нет	При открытии пресета анимация включается автоматически	
Зацикливание анимации	Флаг	Нет	После анимации последней точки, анимация начнется заново с первой точки	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
Скрыть точки анимации в визуализаторе	Флаг	Нет	В визуализаторе отображаются только кнопки. Точки и прогрессбар не отображаются	
Управляющий блок	Флаг	Нет	При включении анимация сможет управлять другими блоками без связей на графе	
Наименование	Строка	Нет	Идентификатор, по которому происходит привязка анимации и управляемых блоков (одинаковые значения должны быть в данном параметре в анимации и в управляемых визуализаторах)	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.1.4.4.3 Группа «Общие настройки блока»

Аналогично приведенным в п. 6.1.1.1.3.

6.1.4.4.4 Входы

В блоке нет входов.

6.1.4.4.5 Выходы

В блоке нет выходов.

6.1.4.4.6 События блока

Название выхода	Тип	Описание
Событие шага анимации	Объект	Используется для динамического обновления визуализаторов, при изменении отметки в блоке «Анимация»

6.1.4.4.7 Описание работы блока

С помощью блока «Анимация» другие визуализаторы могут анимироваться двумя способами:

- Через механизм событий: текущая отметка в анимации используется как параметр в других блоках, из-за чего меняются данные в визуализаторах. При переключении на следующий шаг анимации происходит расчет цепочки графа после события анимации, и визуализаторы запрашиваются с актуальными для данного шага данными.
- С помощью параметра «Управляющий блок» в настройках блока: в таком случае в визуализаторе изначально есть все необходимые для анимации данные (актуально только для блока «Диаграмма»).

6.1.4.4.8 Описание работы визуализатора

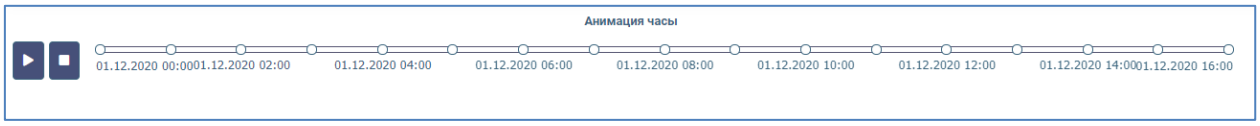


Рисунок 207 – Пример способа анимации через механизм событий

- Чтобы запустить анимацию необходимо нажать кнопку Play.
- Чтобы поставить анимацию на паузу необходимо нажать кнопку Pause.
- Чтобы остановить анимацию необходимо нажать кнопку Stop.
- Существует возможность вручную переключиться на любую точку вручную.

6.1.4.4.9 Настройки визуализатора

Общий список настроек приведен в п. 4.2.8 «Настройки визуализатора».

Помимо стандартных, визуализатор позволяет задать следующие настройки.

Название параметра	Тип параметра	Обязательный	Описание	Пример
Размер шрифта текста кнопок	Целое число	Нет	Размер кнопок	
Цвет текста кнопок	Цвет	Нет	Цвет иконок play, stop, pause внутри кнопок	
Цвет фона кнопок	Цвет	Нет	Цвет фона кнопок	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
Скругление углов кнопок	Целое число	Нет	Скругление кнопок в px	

6.1.4.5 Блок «Видео»

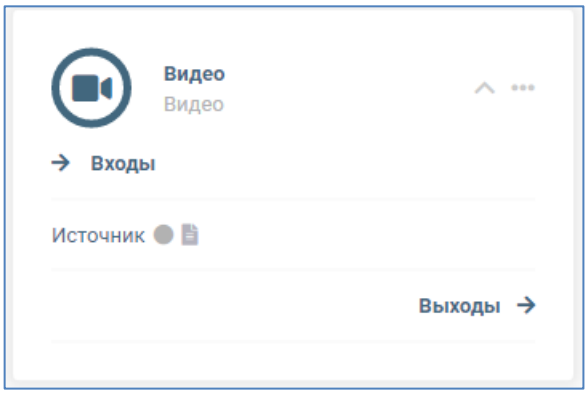


Рисунок 208 – Пример блока «Видео»

6.1.4.5.1 Назначение

Блок используется для воспроизведения потокового видео.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.1.4.5.2 Настройки блока

Название параметра	Тип параметра	Обязательный	Описание	Пример
Раздел настроек «Основные свойства»				

Источник	JSON	Нет	Описание источника видеосигнала и его тип
			Важно!
			На текущий момент поддерживается **только** формат **hls.**
			Формат **rtsp** не поддерживается
				`{`
				`"src": "https://vs.domru.ru/translation?id=562636189&guid=7efd02321cfd51fc7da7&mode=hls",`
				`"type": "application/x-mpegURL"`
				`}`

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.1.4.5.3 Группа «Общие настройки блока»

Аналогично приведенным в п. 6.1.1.1.3.

6.1.4.5.4 Входы

Название входа	Тип	Обязательный	Описание
Источник	Строка (JSON)	Нет	Строковое описание источника видеосигнала и его тип. Вход аналогичен полю «Источник» из настроек блока Приоритет данных будет за входом блока, т.е., при наличии заполненного поля «Источник» и данных на этом входе, в визуализатор будет передана информация из входа блока

6.1.4.5.5 Выходы

В блоке нет выходов.

6.1.4.5.6 События блока

В блоке нет событий.

6.1.4.5.7 Описание работы блока

Блок «Видео» предназначен для отображения потокового видео в пресете из источника, заданного в блоке (или переданного на вход блока).

6.1.4.5.8 Описание работы визуализатора

Рисунок 209 – Внешний вид визуализатора «Видео»

Визуализатор отображает видеопоток, который передается из настроек блока.

6.1.4.5.9 Настройки визуализатора

Визуализатор не имеет специфических настроек. Общий список настроек приведен в п. 4.2.8 «Настройки визуализатора».

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.1.4.6 Блок «Загрузить файл»

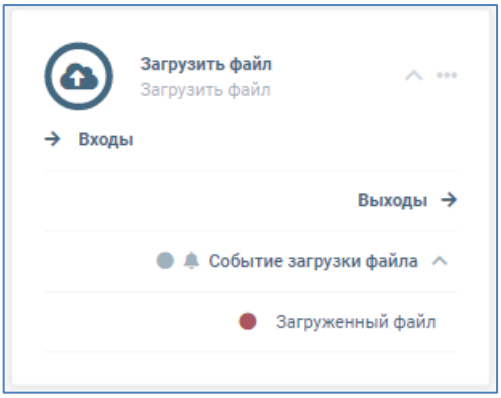


Рисунок 210 – Пример Блока «Загрузить файл»

6.1.4.6.1 Назначение

Блок позволяет отобразить визуализатор, с помощью которого можно произвести загрузку файла на сервер.

6.1.4.6.2 Настройки блока

Название параметра	Тип параметра	Обязательный	Описание	Пример
Расширение файла	Строка	Нет	Перечисление расширений файлов, которые можно загружать. Формат – через запятую (,), без пробелов. Чтобы выбрать все файлы для загрузки, поле необходимо оставить пустым	json, csv, txt

6.1.4.6.3 Группа «Общие настройки блока»

Аналогично приведенным в п. 6.1.1.1.3.

6.1.4.6.4 Входы

У блока нет входов.

6.1.4.6.5 Выходы

У блока нет выходов.

6.1.4.6.6 События блока

Название выхода	Тип	Описание
Событие загрузки файла	JSON	JSON, содержащий в себе полный путь до загруженного файла. Файл должен располагаться в той же задаче, что и сам блок. **Формат выхода** `{"fileNames": ["<путь до файла>"]}`

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.1.4.6.7 Описание работы блока

6.1.4.6.8 Описание работы визуализатора

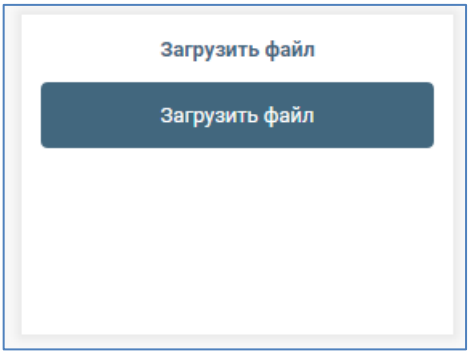


Рисунок 211 – Внешний вид визуализатора «Загрузить файл»

При нажатии на кнопку «Загрузить файл» появляется окно с выбором файла, после чего происходит загрузка файла.

Об успешности загрузки будет свидетельствовать сообщение в визуализаторе «Файл успешно загружен». В этом случае на событийный выход блока уйдет информация по местонахождению файла на сервере.

6.1.4.6.9 Настройки визуализатора

Общий список настроек приведен в п. 4.2.8 «Настройки визуализатора».

Помимо стандартных настроек, визуализатор позволяет задать следующие настройки:

Название параметра	Тип параметра	Обязательный	Описание
Размер шрифта текста кнопок	Целое число	Нет	Задает размер текста в кнопке
Цвет текста кнопок	Цвет	Нет	Задает цвет текста в кнопке
Цвет фона кнопок	Цвет	Нет	Задает цвет фона в кнопке
Цвет текста активной кнопки	Цвет	Нет	Задает цвет текста активной (нажатой) кнопки
Цвет фона активной кнопки	Цвет	Нет	Задает цвет фона активной (нажатой) кнопки
Скругление углов кнопок	Целое число	Нет	Задает скругление углов кнопки

6.1.4.7 Блок «Меню»

6.1.4.7.1 Назначение

Блок позволяет настраивать переключение пресетов в приложении без необходимости задействовать меню пресета.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

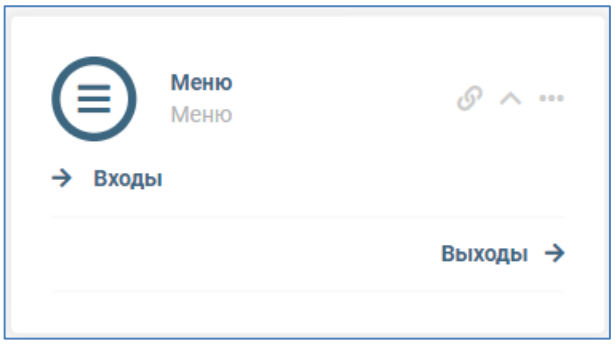


Рисунок 212 – Пример Блока «Меню»

6.1.4.7.2 Настройки блока

Название параметра	Тип параметра	Обязательный	Описание	Пример
Вид отображения	Выпадающий список	Да	Вид отображения меню в приложении. По умолчанию: **Комбобокс**	
Настройки стиля меню	Выпадающий список	Да	Выбор какие настройки будут применяться. По умолчанию: **Задать параметры вручную**	
Начертание	Выпадающий список	Да	Задаёт настройки шрифта названия пресетов. По умолчанию: **Полужирный**	
Регистр	Выпадающий список	Да	Задаёт регистр названия пресета. По умолчанию: **ВСЕ ПРОПИСНЫЕ**	
Размер текста	Целое число	Нет	Задает размер шрифта названия пресета. По умолчанию: **18**	
Цвет текста	Цвет	Нет	Задает цвет шрифта названия пресета. По умолчанию: **#3c667f**	
Растягивать по ширине	Флаг	Да	Растянуть или нет блок по ширине экрана. По умолчанию: **Активно**	
Представления для отображения	Список	Нет	Выбор нужных пресетов из списка всех доступных пресетов в задаче	

6.1.4.7.3 Группа «Общие настройки блока»

Аналогично приведенным в п. 6.1.1.1.3.

6.1.4.7.4 Входы

В блоке нет входов.

6.1.4.7.5 Выходы

В блоке нет выходов.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.1.4.7.6 События блока

В блоке нет событий.

6.1.4.7.7 Описание работы блока

Блок размещается на графе в **единственном** экземпляре. При попытке создать ещё один блок «Меню» (п. 6.1.4.7) из библиотек возникнет ошибка.

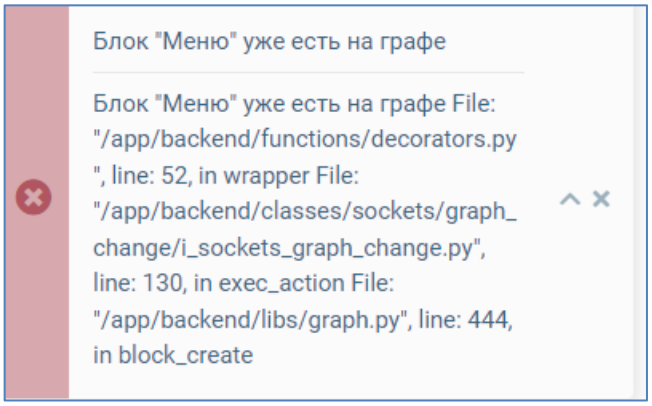


Рисунок 213 – Пример ошибки

Визуальные настройки устанавливаются в блоке на графе. Настройки прописываются вручную, либо могут использоваться глобальные стили.

Глобальные стили находятся в меню администрирования.

Меню глобальных стилей блока «Меню».

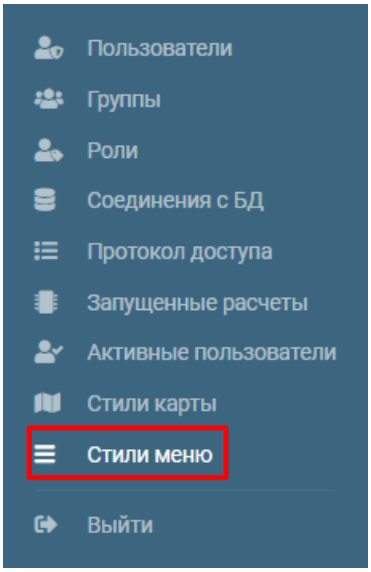


Рисунок 214 – Меню глобальных стилей блока «Меню»

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Настройки блока на графе приведены ниже на рисунке.

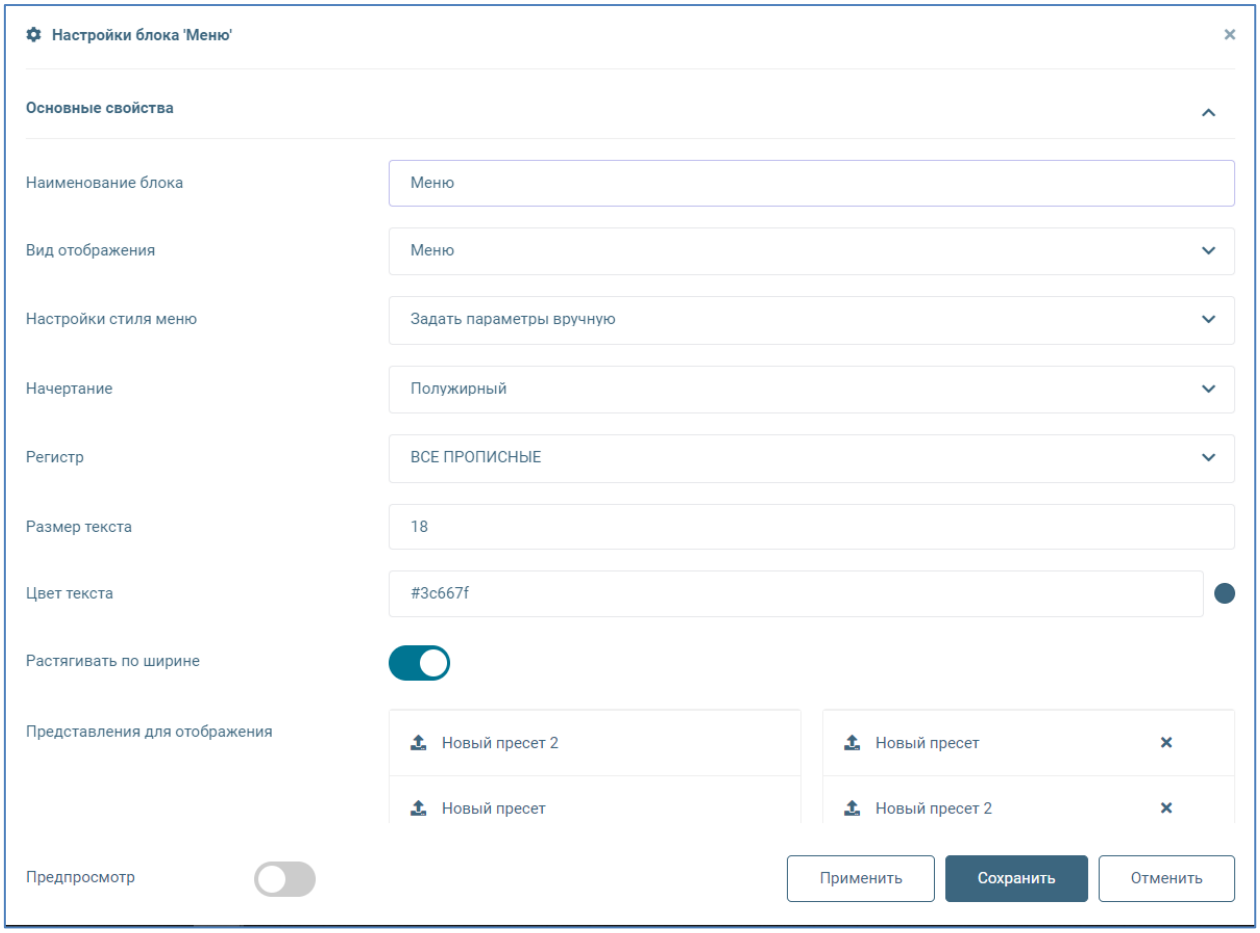


Рисунок 215 – Настройки блока «Меню» на графе

Настройка «Представление для отображения» позволяет выбрать необходимые для отображения пресеты и скрыть ненужные.

Настройка осуществляется перетаскиванием между двумя колонками.

Для перетаскивания необходимо навести курсор на пресет, зажать ЛКМ и перетащить нужный пресет в нужную колонку.

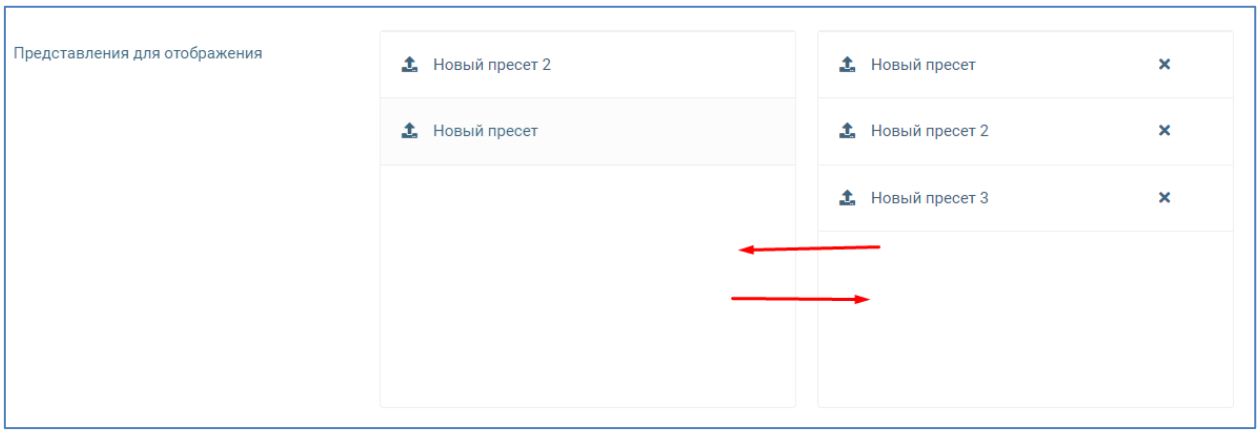


Рисунок 216 – Настройка «Представление для отображения»

Отображаться будут пресеты из правого столбца, скрываться пресеты из левого столбца.

Настройка «**Вид отображения**» Комбобокс никак не отображается на пресете.

При помощи этой настойки в меню пресета скрываются не нужные пресеты.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Пример работы настройки «**Комбобокс**» приведен ниже на рисунке.

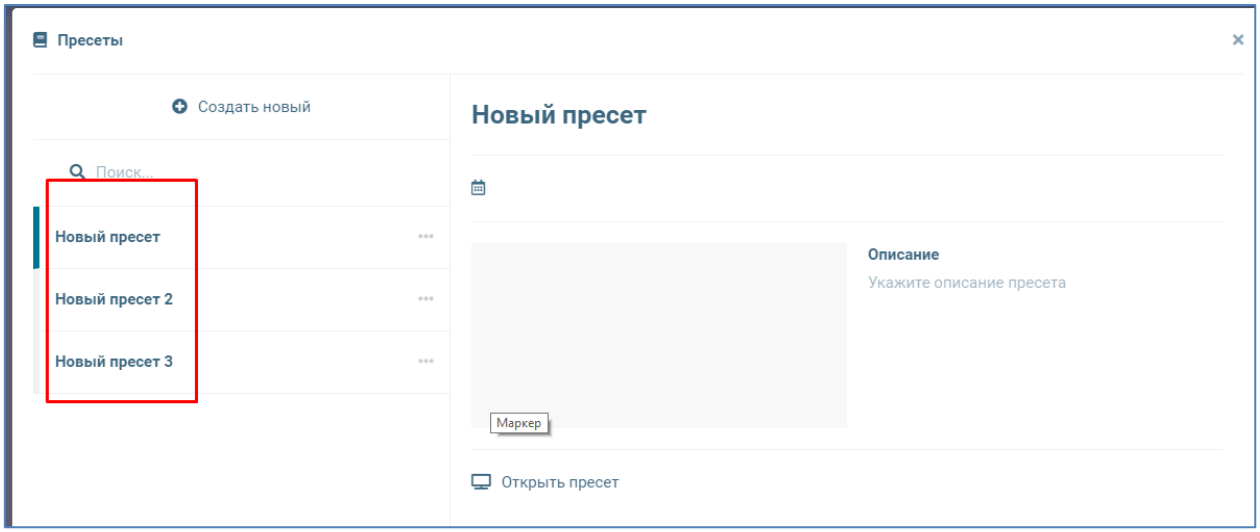


Рисунок 217 – Пример работы настройки «**Комбобокс**»

Пример отображения блока «Меню» на пресете приведен ниже на рисунке.

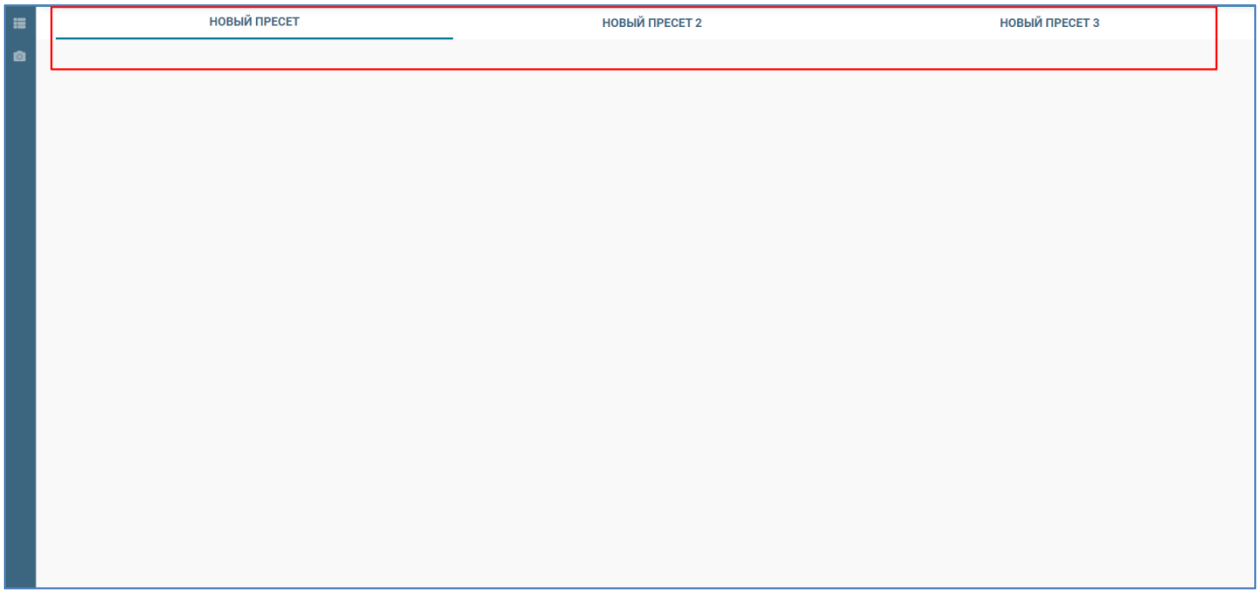


Рисунок 218 – Пример отображения блока «Меню» на пресете

Для переключения между пресетами необходимо нажать по названию пресета.

6.1.4.7.8 Настройки визуализатора

Настройки визуализатора не доступны для блока «Меню» (п. 6.1.4.7).

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.1.4.8 Блок «Панель OLAP»

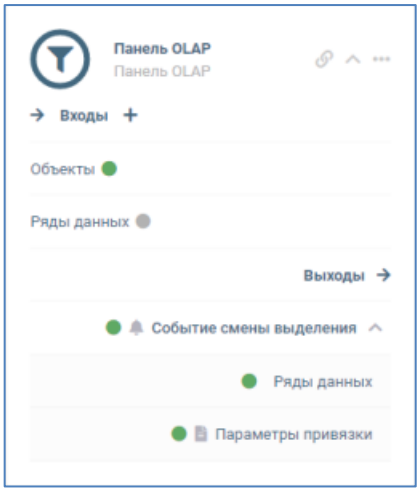


Рисунок 219 – Пример блока «Панель OLAP»

6.1.4.8.1 Назначение

Блок позволяет формировать визуализатор и ряды данных на основании привязок и данных, поступивших на вход. Пользователь может управлять отметкой, при этом зависимые визуализаторы будут обновлены с актуальными данными.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.1.4.8.2 Настройки блока

Название параметра	Тип параметра	Обязательный	Описание	Пример

Параметры привязки	JSON	Да	Параметры сопоставления входного массива данных, для формирования визуализатора и выходных данных.
			Возможные значения настроек:
			• **name** – Текстовое наименование справочника.
			• **key** – Первичный ключ, отвечающий за формирование справочника.
			• **name_key** – Идентификатор атрибута, отвечающий за наименование элемента.
			• **hierarchy** – Флаг наличия иерархии в выбранном справочнике.
			• **rows** – Флаг для отображения справочника по строкам. Не использовать вместе с флагом **columns**.
			• **columns** – Флаг для отображения справочника по столбцам. Не использовать вместе с флагом **rows**.
			Если отмечены обе настройки (**rows** и **columns**), то приоритет будет у флага **columns**.
			Если обе настройки не отмечены, то справочник уйдет в фиксированные значения.
			lazy – Флаг, отвечающий за «ленивую загрузку» данных справочника
			input_id - Идентификатор входа справочника. Требуется для иерархических справочников.
			dict_key - Идентификатор справочника, который будет использоваться как ключ. Требуется для иерархических справочников.
			dict_name - Идентификатор справочника, который будет использоваться как наименование. Требуется для иерархических справочников.
				Параметры привязки
				`[`
				`  {`
				`  "name": "Регионы",`
				`  "key": "region_id",`
				`  "name_key": "region_name",`
				`  "rows": "True"`
				`  },`
				`  {`
				`  "name": "Даты",`
				`  "key": "data",`
				`  "columns": "True"`
				`  },`
				`  {`
				`  "name": "Показатели",`
				`  "key": "pok"`
				`  },`
				`  {`
				`  "name": "Источник данных",`
				`  "key": "source"`
				`  }`
				`]`

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.1.4.8.3 Группа «Общие настройки блока»

Аналогично приведенным в п. 6.1.1.1.3.

6.1.4.8.4 Входы

Название входа	Тип	Обязательный	Описание
Объекты	Объект	Нет	Массив с объектами
Ряды данных	Ряд данных	Нет	Массив с рядами данных

Если будут задействованы оба входа, то приоритет (т.е. те данные, которые будет использовать блок) будет у входа «Объекты».

6.1.4.8.5 Динамические входы

Для подключения выпадающих списков в атрибутах (см. настройку "Параметры привязки") необходимо добавить входы для справочников.

Каждый новый справочник добавляется в отдельный вход. Параметры "Обязательный" и "Массив" должны быть включены. Параметр "Идентификатор" необходимо указать в поле input_id в параметрах привязки.

Для создания входа необходимо нажать на знак «+» в разделе входов блока. Для каждого из входов задается следующий перечень полей:

Название входа	Тип	Обязательный	Описание
Идентификатор	Строка	Да	Идентификатор входа - используется внутри блока, для получения доступа к данным входа
Имя	Строка	Да	Имя входа для отображения в списке входов блока
Описание	Строка	Нет	Дополнительная информация с описанием входа
Тип	Выпадающий список	Да	Тип данных, которые будут приходить на вход. Нужен для построения корректных связей с другими блоками. По умолчанию установлен тип "Объект".
Обязательный	Флаг	Да	Обязательность входа. Если вход обязательный и на него не пришло данных - блок не будет рассчитан с ошибкой
Массив	Флаг	Да	Является ли вход массивом данных
Обязательный	Флаг	Да	Обязательность входа. Если вход обязательный и на него не пришло данных - блок не будет рассчитан с ошибкой

6.1.4.8.6 Выходы

В блоке нет выходов.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.1.4.8.7 События блока

Название выхода	Тип	Описание
Событие смены выделения		
Ряды данных	Ряд данных	Массив отформатированных рядов данных Используется для динамического обновления визуализаторов, при изменении отметки в таблице.
Параметры привязки	Строка	Динамические параметры привязки для корректного отображения блоков-визуализаторов. Для корректного отображения визуализаторов, необходимо чтобы этот параметр был присоединен ко всем блокам-визуализатором (у которых есть аналогичный вход), находящимся после этого блока.

6.1.4.8.8 Описание работы блока

Блок формирует визуализатор, данные и параметры отображения визуализаторов согласно настройкам, заданным в поле «**параметры привязки**».

В момент работы, при изменении отметки в визуализаторе, автоматически пересчитываются ряды и все блоки, стоящие после данного блока.

6.1.4.8.9 Описание работы визуализатора

Все справочники, описанные в настройках в разделе «Параметры привязки», попадают в визуализатор, в одну из трех секций: по столбцам, по строкам или же фиксированные.

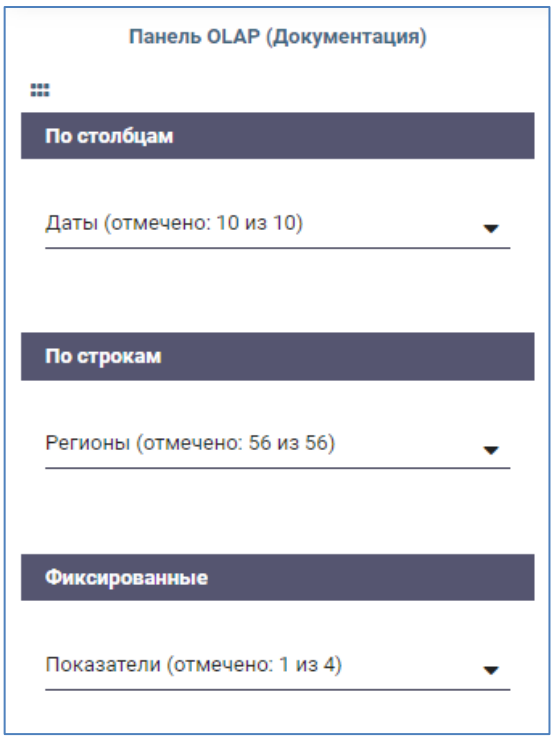


Рисунок 220 – Внешний вид визуализатора «Панель OLAP»

В самом визуализаторе можно перетаскивать справочники между секциями с помощью механизма drag-n-drop (необходимо навести курсор на справочник, зажать левую кнопку мыши и перетащить справочник в нужную секцию).

Ряды данных будут отображаться, только если в разделе столбцов и в разделе строк находятся какие-либо справочники.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Для того чтобы отметить или снять отметку с элементов, необходимо нажать на справочник левой кнопкой мыши, и отметить (или наоборот, снять отметку) с нужных элементов.

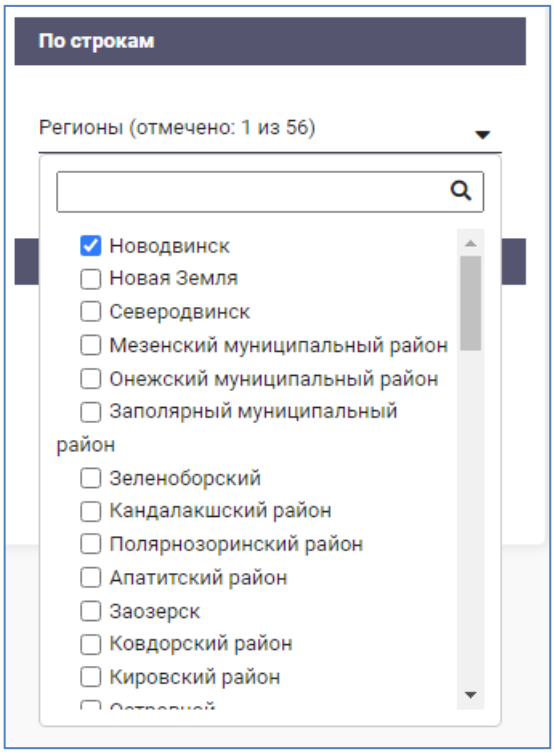


Рисунок 221 – Пример отметки элементов

Множественная отметка доступна только для столбцов и строк. Отметка у справочников в фиксированном разделе может содержать только один элемент.

Чтобы одновременно отметить или снять отметку для нескольких элементов, есть возможность открыть контекстное меню, нажав правой кнопкой мыши на любой элемент справочника.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

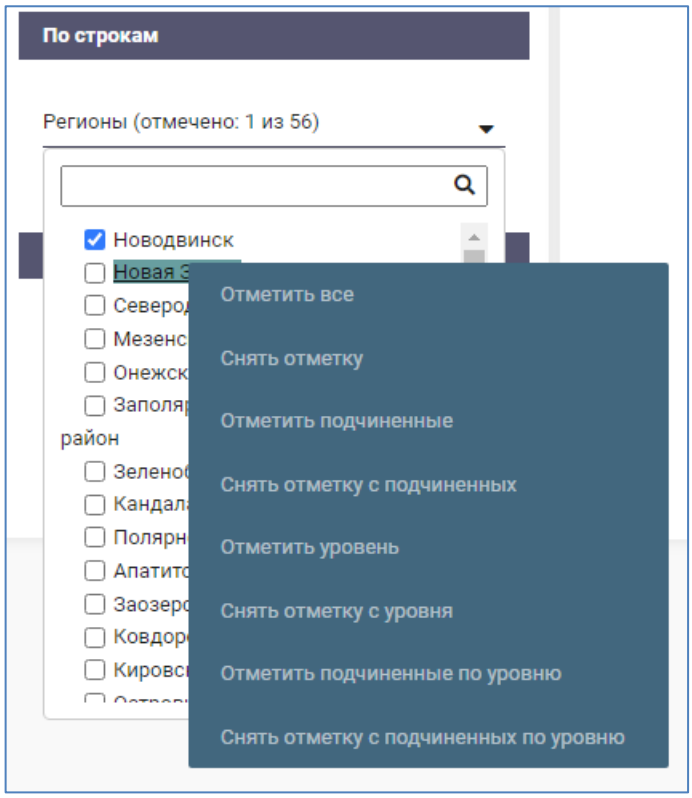


Рисунок 222 – Пример контекстного меню

Контекстное меню содержит следующие пункты:

1. Отметить все.
2. Снять отметку.
3. Отметить подчиненные.
4. Снять отметку с подчиненных.
5. Отметить уровень.
6. Снять отметку с уровня.
7. Отметить подчиненные по уровню.
8. Снять отметку с подчиненных по уровню.

После каждой смены отметки элементов и/или перемещения справочников, происходит событие смены отметки, изменяющее ряды данных и перерисовывающее все визуализаторы, стоящие после блока OLAP.

Для более детальной работы с панелью OLAP см. п. 6.1.4.8.11.

6.1.4.8.10 Настройки визуализатора

Общий список настроек приведен в п. 4.2.8 «Настройки визуализатора».

Помимо стандартных, визуализатор позволяет задать следующие настройки:

Название параметра	Тип параметра	Обязательный	Описание	Пример
Размер шрифта заголовка раздела	Целое число	Да	Задает размер заголовка раздела	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
Цвет текста заголовка раздела	Цвет	Да	Задает цвет текста для заголовка секции	rgb(255, 255, 255)
Цвет фона заголовка раздела	Цвет	Да	Задает цвет фона для заголовка раздела	rgb(85, 84, 113)

6.1.4.8.11 Работа с панелью OLAP

Блок «Панель OLAP» позволяет в режиме реального времени изменять структуру представленных отчетных данных: возможность расширенной фильтрации данных, сравнения нескольких показателей, изменение шапки таблицы, открытие или скрытие нескольких показателей.

Для редактирования блока «Панель OLAP» в графе необходимо выполнить следующие действия:

- Открыть настройки блока «Панель OLAP».
- При необходимости изменить параметры привязок данных таблицы, включить или выключить предпросмотр данных.

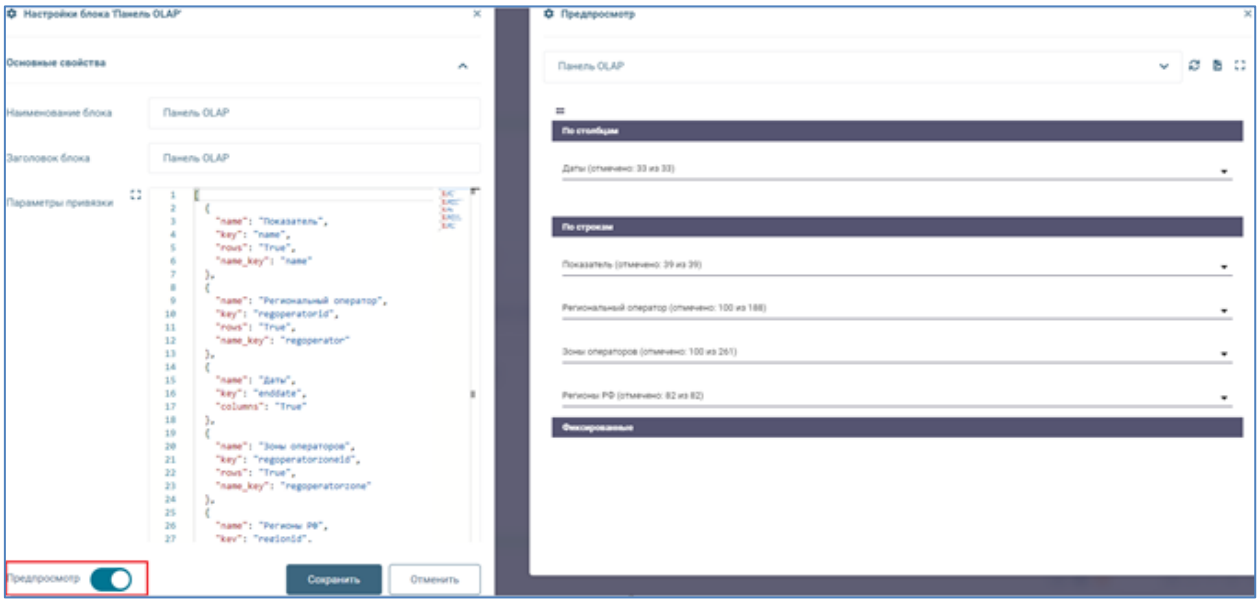


Рисунок 223 – Пример настройки блока «Панель OLAP»

Редактирование отчетной формы, содержащей панель OLAP:

- Нажать на кнопку «Настройка гибкой отчетности» в левой верхней части пресета. При нажатии на кнопку откроется фильтр для таблицы, разделенный по столбцам и строкам, который соответствует блоку «Панель OLAP».
- При необходимости изменить настройки фильтрации, например порядок показателей в столбцах и строках.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

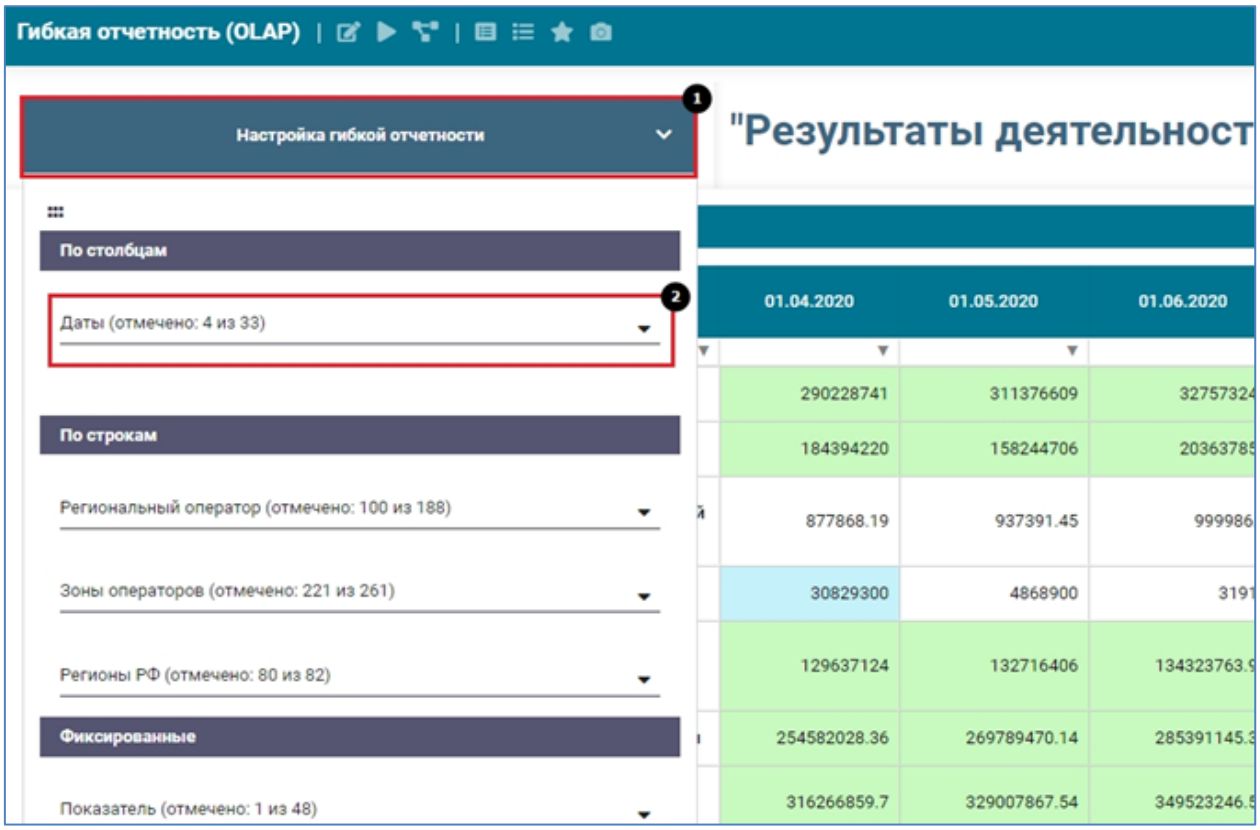


Рисунок 224 – Настройка гибкой отчетности

Для того чтобы скрыть панель фильтрации, вновь нажать на кнопку «Настройка гибкой отчетности». Выбранный фильтр применится автоматически, результат фильтрации будет отображен на всех связанных с панелью OLAP визуализаторами (таблица, диаграмма).

При работе с панелью OLAP присутствует **возможность добавления строк и колонок с вычисляемыми значениями** суммы, среднего, максимального или минимального значений.

Для того чтобы добавить строку или столбец с вычисляемыми значениями, необходимо:

- Открыть граф, содержащий блок «Панель OLAP».
- Открыть настройки блока «Итоги таблицы».

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

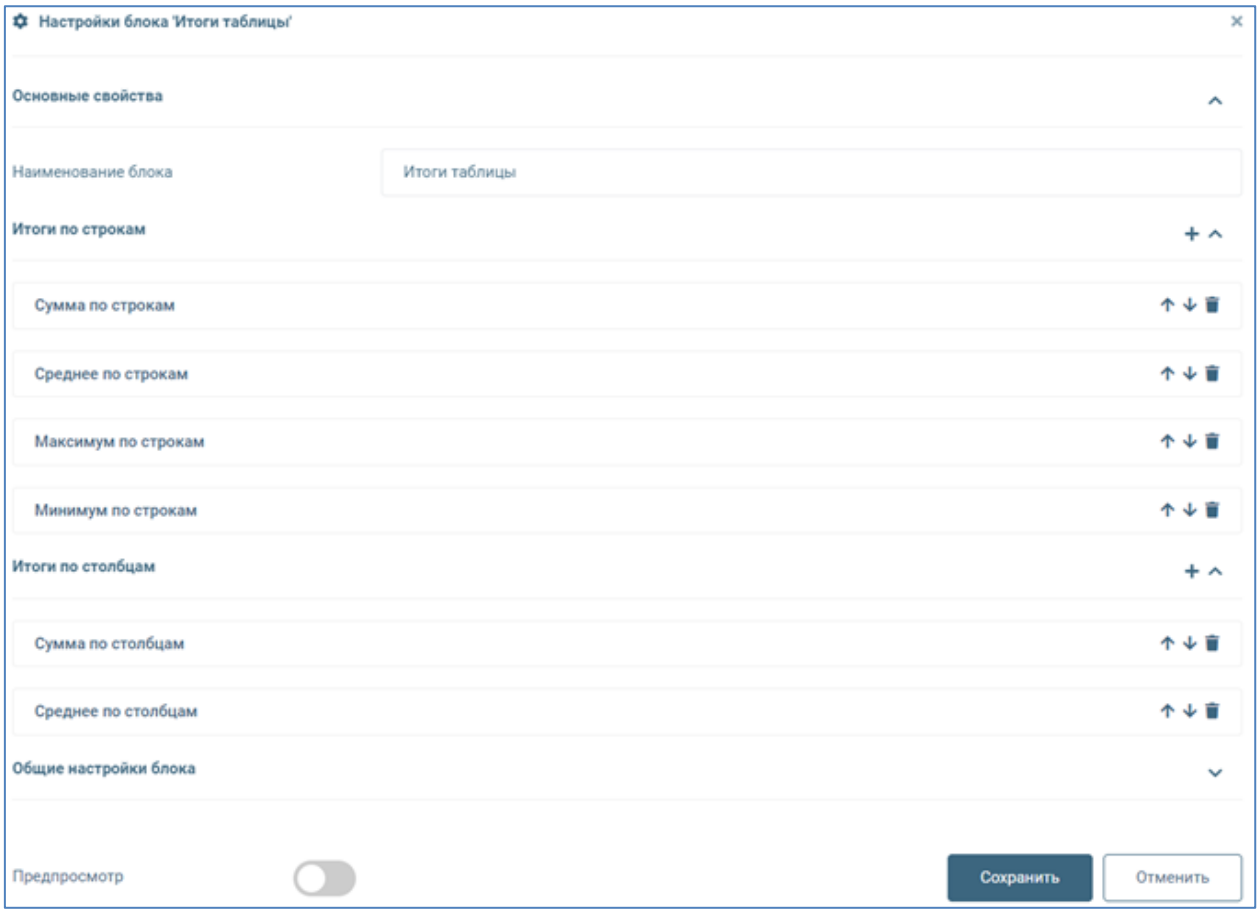


Рисунок 225 – Пример настройки блока «Итоги таблицы»

- В настройках блока при необходимости внести изменения в разделы «Итоги по строкам» и «Итоги по столбцам».
    - Добавить новую строку или колонку с вычисляемыми значениями, нажав на кнопку + в соответствующих настройках блока.
    - Прописать название итоговой строки (столбца), ее (его) тип.
    - Сохранить изменения блока, нажав на соответствующую кнопку.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Рисунок 226 – Пример настройки блока «Итоги по столбцам»

- Открыть выпадающее меню блока, нажав на кнопку ••• в правой части блока.
- Рассчитать изменённый блок и следующие за ним блоки, нажав на соответствующую кнопку в меню блока.

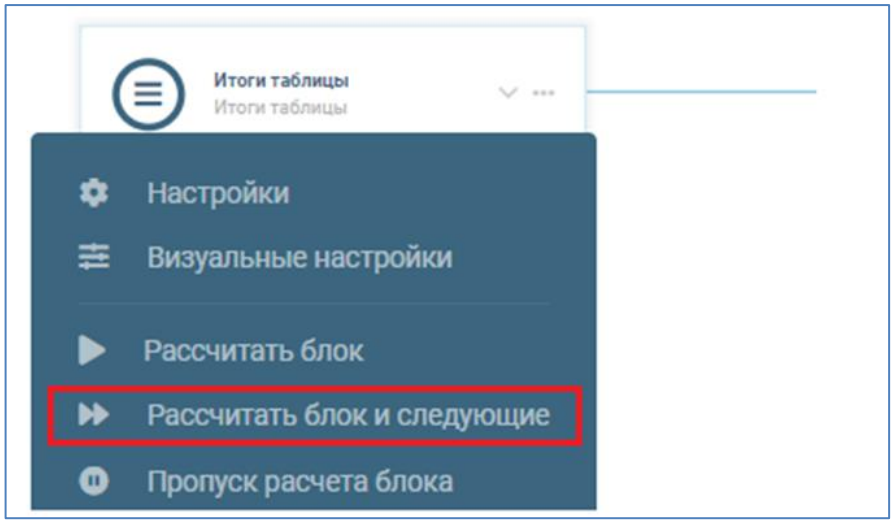


Рисунок 227 – Пункт пункта меню «Рассчитать блок и следующие»

- Открыть соответствующий графу пресет.
- Открыть выпадающее меню «Итоги таблицы».
- Включить отображение необходимых строк или столбцов с вычисляемыми значениями.
- Нажать на кнопку «Применить» в выпадающем меню.
- Скрыть выпадающее меню «Итоги таблицы».

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

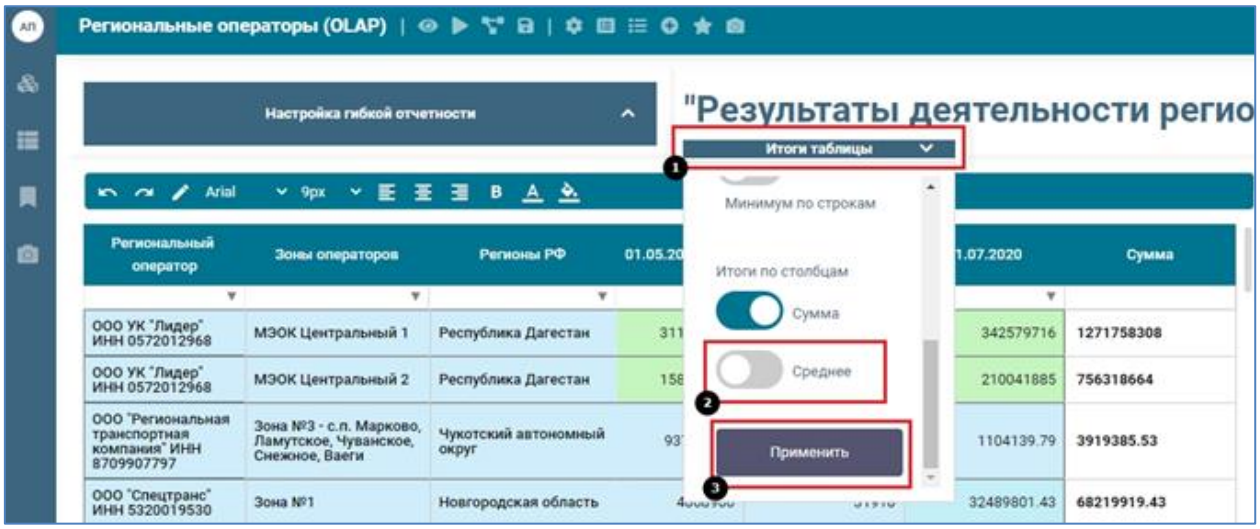
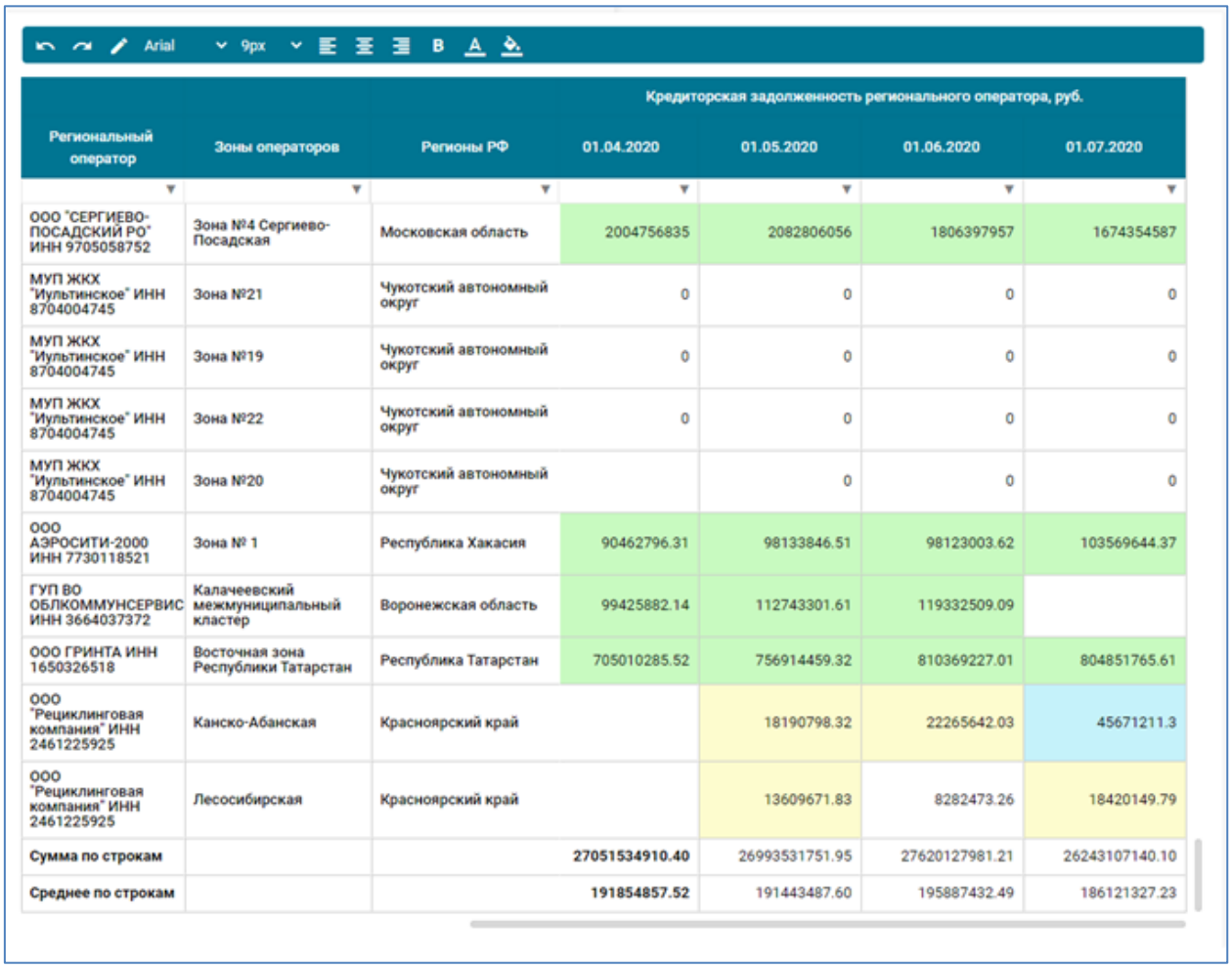


Рисунок 228 – Пример меню «Итоги таблицы»

Пример добавления суммы по строкам и среднего по строкам выглядит следующим образом:



Региональный оператор	Зоны операторов	Регионы РФ	Кредиторская задолженность регионального оператора, руб. 01.04.2020	01.05.2020	01.06.2020	01.07.2020
ООО "СЕРГИЕВО-ПОСАДСКИЙ РО" ИНН 9705058752	Зона №4 Сергиево-Посадская	Московская область	2004756835	2082806056	1806397957	1674354587
МУП ЖКХ "Иультинское" ИНН 8704004745	Зона №21	Чукотский автономный округ	0	0	0	0
МУП ЖКХ "Иультинское" ИНН 8704004745	Зона №19	Чукотский автономный округ	0	0	0	0
МУП ЖКХ "Иультинское" ИНН 8704004745	Зона №22	Чукотский автономный округ	0	0	0	0
МУП ЖКХ "Иультинское" ИНН 8704004745	Зона №20	Чукотский автономный округ		0	0	0
ООО АЭРОСИТИ-2000 ИНН 7730118521	Зона № 1	Республика Хакасия	90462796.31	98133846.51	98123003.62	103569644.37
ГУП ВО ОБЛКОММУНСЕРВИС ИНН 3664037372	Калачеевский межмуниципальный кластер	Воронежская область	99425882.14	112743301.61	119332509.09	
ООО ГРИНТА ИНН 1650326518	Восточная зона Республики Татарстан	Республика Татарстан	705010285.52	756914459.32	810369227.01	804851765.61
ООО "Рециклинговая компания" ИНН 2461225925	Канско-Абанская	Красноярский край		18190798.32	22265642.03	45671211.3
ООО "Рециклинговая компания" ИНН 2461225925	Лесосибирская	Красноярский край		13609671.83	8282473.26	18420149.79
Сумма по строкам			**27051534910.40**	26993531751.95	27620127981.21	26243107140.10
Среднее по строкам			**191854857.52**	191443487.60	195887432.49	186121327.23

Рисунок 229 – Пример добавления суммы по строкам и среднего по строкам

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.1.4.9 Блок «Панель OLAP для больших данных»

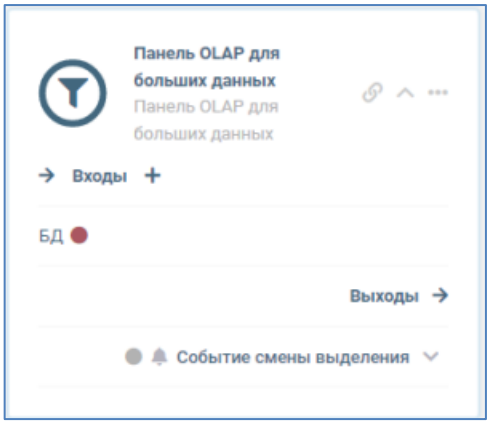


Рисунок 230 – Блок «Панель OLAP для больших данных»

6.1.4.9.1 Назначение

Блок позволяет формировать визуализатор и ряды данных на основании привязок и данных, поступивших на вход. Пользователь может управлять отметкой, при этом зависимые визуализаторы будут обновлены с актуальными данными.

В отличии от блока «Панель OLAP» (п. 6.1.4.8), данный визуализатор работает напрямую с базой данных, что увеличивает быстродействие и позволяет работать с большим количеством точек.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.1.4.9.2 Настройки блока

Название параметра	Тип параметра	Обязательный	Описание
Наименование таблицы	Строка	Да	Идентификатор таблицы в БД, из которой извлекаются данные. В версии блока 3 и ниже наименования атрибутов автоматически подставляются только из блока «Справочник». Для отображения наименования атрибутов из блока «SQL-запрос справочник» необходимо наличие наименований в таблице с данными
Параметры привязки таблицы	JSON	Да	Описание полей из таблицы, которые необходимы в расчетах. Поля для привязки: • **pkey** – массив полей, заданных как первичный ключ. Необходимы для привязки таблицы с данными к справочникам. • **vl** – поле, в котором хранятся значения. • **attrs** – массив с дополнительными атрибутами, необходимыми в расчетах
Параметры привязки OLAP	JSON	Да	Привязка справочников Параметры сопоставления входного массива данных, для формирования визуализатора и выходных данных. Количество описаний должно совпадать с количеством справочников на входе. Возможные значения настроек: • **input_key** – Идентификатор входа справочника • **input_name** – Наименование справочника в визуализаторе. Если не задано, используется значение по умолчанию: «Справочник N», где N – порядковый номер справочника. • **dict_key** – Идентификатор поля в справочнике, который использовать как ключ. Если не задано берется поле по умолчанию – **id**. Поле должно находиться в первичных ключах справочника. • **dict_name** – Идентификатор поля в справочнике, который использовать как наименование. Если не задано берется поле по умолчанию – name. Поле должно находиться в атрибутах справочника.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание
			• **table_key** – Идентификатор поля в таблице, с которым необходимо связывать ключ справочника. Если не задано используется значение из поля **dict_key**. • **table_name** – Идентификатор поля в таблице, из которого необходимо брать наименование для ряда. Поле используется только для отображения наименования из блока «SQL-запрос справочник». Если используется блок «Справочник», добавлять данное поле не рекомендуется. • **hierarchy** – Флаг наличия иерархии в выбранном справочнике. • **rows** – Флаг для отображения справочника по строкам. Не использовать вместе с флагом **columns**. • **columns** – Флаг для отображения справочника по столбцам. Не использовать вместе с флагом **rows**. Если отмечены обе настройки (**rows** и **columns**), то приоритет будет у флага **columns**. Если обе настройки не отмечены, то справочник уйдет в фиксированные значения. • **lazy** – Флаг, отвечающий за «ленивую загрузку» данных справочника
Максимальное количество отображаемых ячеек данных	Целое число	Да	Максимальное число ячеек с данными. По умолчанию – 10000

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.1.4.9.3 Группа «Общие настройки блока»

Аналогично приведенным в п. 6.1.1.1.3.

6.1.4.9.4 Входы

Название входа	Тип	Обязательный	Описание
БД	Соединение с БД	Да	Соединение с базой данных, из которой необходимо извлекать данные. Поддерживаются разные диалекты баз, а именно: • PostgreSQL • MSSQL • Oracle

Каждый новый справочник добавляется в отдельный вход. Параметры «Обязательный» и «Массив» должны быть включены. Параметр «Идентификатор» необходимо указать в поле input_key в параметрах привязки

Для создания входа необходимо нажать на знак + в разделе входов блока. Для каждого из входов задается следующий перечень полей.

Название входа	Тип	Обязательный	Описание
Идентификатор	Строка	Да	Идентификатор входа – используется внутри блока, для получения доступа к данным входа
Имя	Строка	Да	Имя входа для отображения в списке входов блока
Описание	Строка	Нет	Дополнительная информация с описанием входа
Тип	Выпадающий список	Да	Тип данных, которые будут приходить на вход. Нужен для построения корректных связей с другими блоками. По умолчанию установлен тип "Объект".
Обязательный	Флаг	Да	Обязательность входа. Если вход обязательный и на него не пришло данных – блок не будет рассчитан с ошибкой
Массив	Флаг	Да	Является ли вход массивом данных

6.1.4.9.5 Выходы

В блоке нет выходов.

6.1.4.9.6 События блока

Название выхода	Тип	Описание
Событие смены выделения		

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название выхода	Тип	Описание
Ряды данных	Ряд данных	Массив отформатированных рядов данных Используется для динамического обновления визуализаторов, при изменении отметки в таблице
Параметры привязки	Строка	Динамические параметры привязки для корректного отображения блоков-визуализаторов. Для корректного отображения визуализаторов, необходимо чтобы этот параметр был присоединен ко всем блокам-визуализатором (у которых есть аналогичный вход), находящимся после этого блока

6.1.4.9.7 Описание работы блока

Блок извлекает данные из таблицы, указанной в настройках, согласно привязкам, заданным в поле «**параметры привязки таблицы**».

После чего формируется визуализатор и параметры отображения визуализаторов согласно настройкам, заданным в поле «**параметры привязки OLAP**».

В момент работы, при изменении отметки в визуализаторе, автоматически пересчитываются ряды и все блоки, стоящие после данного блока.

6.1.4.9.8 Описание работы визуализатора

Все справочники, описанные в настройках в разделе «Параметры привязки», попадают в визуализатор, в одну из трех секций: по столбцам, по строкам или же фиксированные.

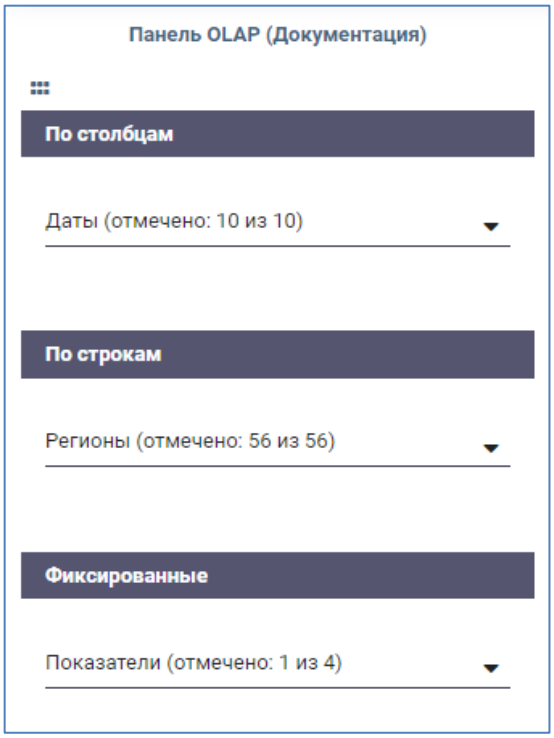


Рисунок 231 – Пример визуализатора «Панель OLAP (Документация)»

В самом визуализаторе можно перетаскивать справочники между секциями с помощью механизма drag-n-drop (необходимо навести курсор на справочник, зажать левую кнопку мыши и перетащить справочник в нужную секцию).

Ряды данных будут отображаться только если в разделе столбцов и в разделе строк находятся какие-либо справочники.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Для того чтобы отметить или снять отметку с элементов, необходимо нажать на справочник левой кнопкой мыши, и отметить (или наоборот, снять отметку) с нужных элементов.

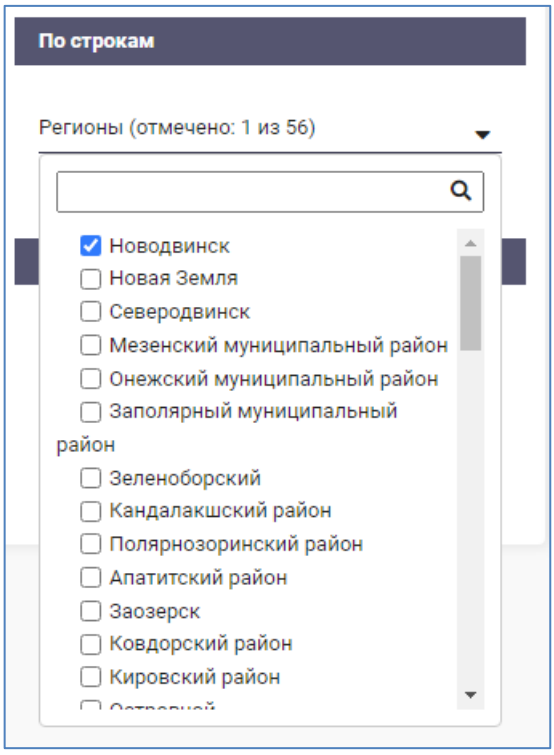


Рисунок 232 – Отметка элементов

Множественная отметка доступна только для столбцов и строк. Отметка у справочников в фиксированном разделе может содержать только один элемент.

Чтобы одновременно отметить или снять отметку для нескольких элементов, есть возможность открыть контекстное меню, нажав правой кнопкой мыши на любой элемент справочника.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

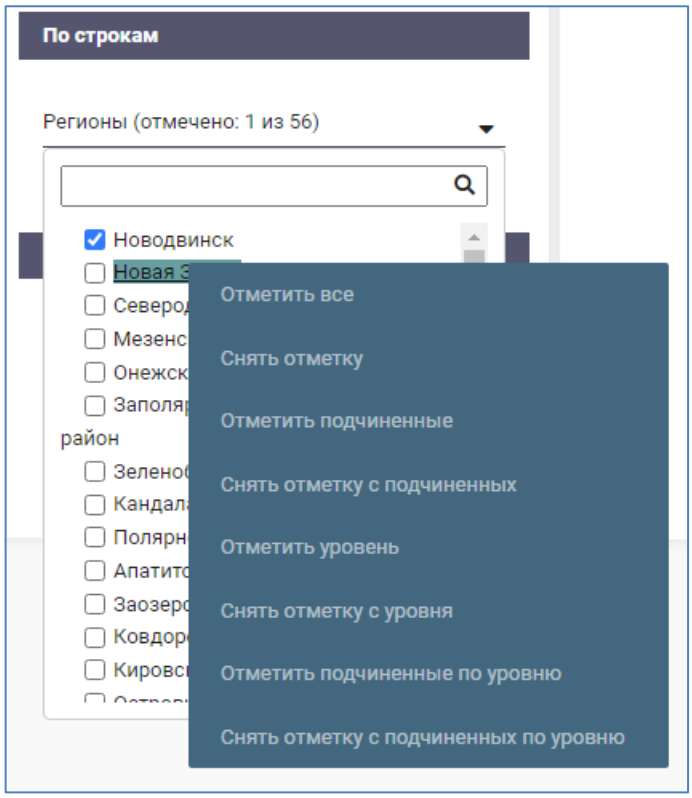


Рисунок 233 – Контекстное меню

Контекстное меню содержит следующие пункты:

1. Отметить все.
2. Снять отметку.
3. Отметить подчиненные.
4. Снять отметку с подчиненных.
5. Отметить уровень.
6. Снять отметку с уровня.
7. Отметить подчиненные по уровню.
8. Снять отметку с подчиненных по уровню.

После каждой смены отметки элементов и/или перемещения справочников, происходит событие смены отметки, изменяющее ряды данных и перерисовывающее все визуализаторы, стоящие после блока OLAP.

Описание работы с панелью OLAP приведено в п. 6.1.4.8.11.

6.1.4.9.9 Настройки визуализатора

Общий список настроек приведен в п. 4.2.8 «Настройки визуализатора».

Помимо стандартных, визуализатор позволяет задать следующие настройки.

Название параметра	Тип параметра	Обязательный	Описание	Пример
Размер шрифта заголовка раздела	Целое число	Да	Задает размер заголовка раздела	
Цвет текста заголовка раздела	Цвет	Да	Задает цвет текста для заголовка секции	rgb(255, 255, 255)

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
Цвет фона заголовка раздела	Цвет	Да	Задает цвет фона для заголовка раздела	rgb(85, 84, 113)

6.1.4.10 Блок «Панель размещения и фильтрации»

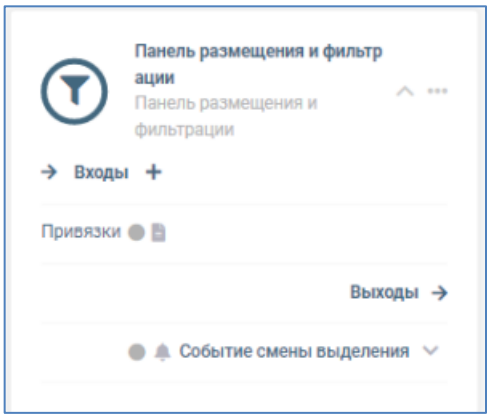


Рисунок 234 – Пример блока «Панель размещения и фильтрации»

6.1.4.10.1 Назначение

Блок позволяет формировать визуализатор для отображения отметок у выбранных справочников. Блок сочетает в себе функционал блока «Панель фильтрации» (п. 6.1.4.11) и блока «Панель OLAP» (п. 6.1.4.8).

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.1.4.10.2 Настройки блока

Название параметра	Тип параметра	Обязательный	Описание	Пример
Режим отображения	Выпадающий список	Да	Выбор режима отображения визуализатора. От выбранного режима так же зависят настройки параметров привязки и возможность работы с другими блоками. Возможные режимы отображения: • Панель фильтрации: • При работе в этом режиме функционал полностью аналогичен блоку «Панель фильтрации»; • Панель OLAP. При работе в этом режиме, визуализатор аналогичен блоку «Панель OLAP», однако сам блок при этом не обрабатывает данные. Подробнее ниже в разделе «**Работа в режиме панели OLAP**»	

Параметры привязки	JSON	Да	Параметры привязки справочников.
			Возможные настройки зависят от выбранного режима отображения:
			• Панель фильтрации
			  • При работе в этом режиме настройки во многом схожи с настройками «Параметры привязки» из блока «Панель фильтрации».
			  • **input_key** – Идентификатор входа справочника.
			  • **input_name** – Наименование справочника в визуализаторе. Если поле не задано, по умолчанию используется «Справочник N», где N – порядковый номер справочника в описании.
				Режим «Панель фильтрации», пример 1:
				`[`
				`  {`
				`  "input_key": "region",`
				`  "input_name": "Город",`
				`  "default": 1,`
				`  "emptyselect": 1`
				`  },`
				`  {`
				`  "input_key": "indicator",`
				`  "input_name": "Показатель",`

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример

			• **key** – Идентификатор поля в справочнике, который использовать как ключ. Если не задано берется поле по умолчанию – ***id***.
			• **name** – Идентификатор поля в справочнике, который использовать как наименование. Если не задано берется поле по умолчанию – ***name***.
			• **default** – Ключи элементов, которые необходимо отмечать по умолчанию. Если поле не задано – отмечается первый элемент при отмеченной единичной отметке или все элементы при включенной мультиотметке.
			• **multiselect** – Целое число, «1» или «0». Включение множественной отметки у справочника (возможность отобразить от нуля до всех элементов). Если не задано – множественная отметка выключена.
			• **singleselect** – Целое число, «1» или «0». Включение единичной отметки у справочника (наличие минимум одного элемента в отметке). Если не задано – множественная отметка выключена.
			Отметки multiselect и singleselect могут быть включены вместе.
			Устарело. Используйте флаг **emptyselect**
			• **emptyselect** – Целое число, «1» или «0». Возможность пустой отметки у справочника (нет отмеченных элементов). По умолчанию – отключено.
			• **combobox** – Целое число, «1» или «0». Отображение справочника в виде
				"default": [1,2],
				"multiselect": 1,
				"combobox": 1
				},
				{
				"input_key": "units",
				"input_name": "Единицы измерения",
				"default": [],
				"multiselect": 1,
				"hierarchy": 0
				}
				]
				Режим «Панель фильтрации», пример 2:
				[
				{
				"input_key": "period",
				"input_name": "Период",
				"key": "id",
				"name": "name",
				"default": 1,
				"multiselect": 1,
				"emptyselect": 1
				},
				{
				"input_key": "year",

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример

			выпадающего списка. По умолчанию выпадающий список выключен.
			• **collapsable** – Целое число, «1» или «0». Возможность сворачивания группы элементов.
			Работает, если combobox = 0
			• **collapsed** - Целое число, «1» или «0». Возможность отобразить группу уже в свернутом состоянии.
			Работает, если collapsable= 1
			• **limit**- Целое число. Количество отображаемых элементов. По умолчанию – отключено (не используется).
			Работает, если combobox = 0
			• **hierarchy** – Целое число, «1» или «0». Отображение данных в виде иерархии. Без включенного флага будут отображаться только данные на первом уровне. При включении иерархии автоматически включается выпадающий список. Отедльно флаг прописывать не нужно. По умолчанию – отключено.
			• **align** – Целое число, «1» или «0». Выравнивание элементов справочника по левому краю. По умолчанию – отключено.
			Работает, если combobox = 0
			• **afterCloseCombobox** – Целое число, «1» или «0». При включении – событие изменения отметки будет выполнено только после закрытия выпадающего списка.
			Работает, если combobox = 1
				<pre>"key": "id",
				"name": "name",
				"default": 1,
				"combobox": 1,
				"collapsable": 1,
				"collapsed":0,
				"limit": 5
				},
				{
				"input_key": "hier",
				"input_name": "Иерархия",
				"default": [3],
				"multiselect": 1,
				"hierarchy": 1,
				"align": 1
				},
				{
				"input_key": "reg",
				"input_name": "Регионы",
				"key": "id",
				"name": "name",
				"multiselect": 1,
				"filter": [
				{
				"from": "year",
				"foreignAttr": "id",
				"localAttr": "region_years"
				}

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример

			• **filter** – Массив словарей. Отображение элементов данного справочника зависит от другого. Допустимые поля:
			  • **from** – Идентификатор справочника, от которого зависит данный справочник.
			  • **foreignAttr** – Идентификатор поля в справочнике, от которого зависит данный справочник, по которому будут фильтроваться элементы
			  • **localAttr** – Идентификатор поля в данном справочнике, по которому будет идти поиск соответсвия с "**foreignAttr**"
			• **addFields** - Массив строк. Состоит из идентификаторов полей в справочнике, по которым идет фильтрация (все значения "**localAttr**") (Обязательно при использовании filter, для всего, кроме временных рядов)
			• **showHeader** – Целое число, «1» или «0». Отключение отображения заголовка раздела. По умолчанию – включено.
			• **canBeEmpty** – Целое число, «1» или «0». При включении раздел будет корректно отображен даже при отсутствии в нем данных.
			• **addFields** – Массив строк. Состоит из идентификаторов полей в справочнике, по которым идет фильтрация (все значения «**localAttr**»)
			• **date_range** – Целое число, «1» или «0». Использование справочника для выбора диапазона дат.
				<pre>],
				  "addFields": [
				   "region_years"
				  ]
				 },
				    {
				  "input_key": "range",
				  "input_name": "Диапазон дат",
				
				"multiselect": 0,
				        "date_range": 1,
				
				"date_range_type": 5,
				
				"date_range_start": "17.05.2019",
				
				"date_range_finish": "01.01.2025",
				        "default": ["01.01.2020", "01.04.2022"]
				 },
				    {
				        "input_key": "string_edit",
				        "input_name": "Строка",
				        "inputLabel": 1,
				
				"inputLabelType": "string",</pre>

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример

			• **date_range_type** – Тип диапазона дат. Возможные значения:
			  • 1 – год;
			  • 2 – полугодие;
			  • 3 – квартал;
			  • 4 – месяц;
			  • 5 – день;
			• **date_range_start** – Дата – начало диапазона в справочнике.
			• **date_range_finish** – Дата – конец диапазона в справочнике
			• **inputLabel** – Целое число, «1» или «0». Вывод в качестве элемента управления строки ввода.
			• **inputLabelType** – Тип данных для строки ввода, возможные значения:
			  • string – строка для ввода;
			  • text – текстовое поле для ввода;
			  • email – электронная почта;
			  • date – дата;
			  • time – время;
			  • phone – телефон;
			  • int – целое число;
			  • float – вещественное число;
			  • color – цвет.
			• **inputLabelHint –** Текстовое поле для подсказки. Не обязательное поле.
			• Панель OLAP
			  • В этом режиме, настройки схожи с настройками «Параметры привязки» блока
				<pre>"inputLabelHint": "Текст подсказки"
				    }
				]</pre>Режим «Панель OLAP»:<pre>[
				    {
				        "input_key": "region",
				        "input_name": "Город",
				        "default": 1,
				        "axis": "row",
				        "name_key": "name"
				
				    },
				    {
				        "input_key": "indicator",
				        "input_name": "Показатель",
				        "default": [1,2],
				        "axis": "col",
				        "name_key": "indicator_name"
				    },
				    {
				        "input_key": "units",
				        "input_name": "Единицы измерения",</pre>

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
			«Панель OLAP» (п. 6.1.4.8), Возможные значения настроек: • **input_key** – Идентификатор входа справочника. Также по этому полю определяется первичный ключ ряда, отвечающий за справочник. • **input_name** – Наименование справочника в визуализаторе. Если поле не задано, по умолчанию используется «Справочник N", где N – порядковый номер справочника в описании. • **name_key** – Идентификатор атрибута, отвечающий за наименование элемента. В отличии от остальных настроек, данный параметр требует указания атрибута из ряда, который будет преобразован с использованием данной панели. • **axis** – Формат расположения справочника. Возможные значения: • **row** – по строкам; • **col** – по столбцам; • **fix** – фиксированный. • **hierarchy** – Флаг наличия иерархии в выбранном справочнике. • **default** – Массив ключей элементов, которые необходимо отмечать по умолчанию. Другие атрибуты, доступные в панели фильтрации, не применяются при данном режиме отображения. Их наличе в параметрах вызовет предупреждение при	`"default": [1], "hierarchy": 0, "name_key": "units_name", "axis": "fix" } ]`

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
			расчете блока. Сами параметры будут игнорироваться. Кроме вышеуказанных параметров, для обоих режимов так же доступен следующий параметр: • **dict** – Идентификатор справочника, для с которым необходимо связать текущий вход блока. С указанным параметром блок во время сохранения автоматически создаст связь между выходом блока с указанным идентификатором и входом панели фильтрации. В качестве идентификатора входа используется **input_key** указанного справочника. Обратите внимание, что параметр **dict** нужно указать для всех привязок, т.к. все входы без указания этого параметра будут удалены	
Отображать кнопку «Применить"	Флаг	Нет	Настройка доступна при выборе «**Панель фильтрации**» в режиме отображения. С включенным флагом отображается кнопка «Применить». Событие смены выделения отправляется после нажатия данной кнопки С выключенным флагом не отображается кнопка «Применить». Событие смены выделения отправляется сразу после изменения отметки	
Отображать всплывающую кнопку «Применить"	Флаг	Нет	Настройка доступна при включении настройки **Отображать кнопку «Применить»**. Кнопка «Применить» всплывает при изменении отметки в панели	
Отображение отметки в комбобоксах	Выпадающий список	Нет	Формат отображения выбранных элементов в выпадающих списках. Используется либо в режиме отображения **OLAP**, либо в случае, когда включен параметр **combobox**.	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
			Возможные режимы отображения: • **Нет** – отметка не отображается. • **Отметка** – отображается список отмеченных элементов. • **Количество** – отображается количество отмеченных элементов в виде надписи ***(n из m)***	
Открытие/закрытие выпадающих списков при наведении курсора мыши	Флаг	Нет	Флаг для возможности открытия и закрытия выпадающих списков не по щелчку мыши, а при наведении курсора	
Фильтры пользователя	Флаг	Нет	Позволяет включить фильтры отметок для пользователей.	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.1.4.10.3 Группа «Общие настройки блока»

Аналогично приведенным в п. 6.1.1.1.3.

6.1.4.10.4 Входы

Каждый новый справочник добавляется в отдельный вход. Параметры «Обязательный» и «Массив» должны быть включены. Параметр «Идентификатор» необходимо указать в поле **input_key** в параметрах привязки.

Для создания входа необходимо нажать на знак + в разделе входов блока. Для каждого из входов задается следующий перечень полей.

Название входа	Тип	Обязательный	Описание
Идентификатор	Строка	Да	Идентификатор входа – используется внутри блока, для получения доступа к данным входа
Имя	Строка	Да	Имя входа для отображения в списке входов блока
Описание	Строка	Нет	Дополнительная информация с описанием входа
Тип	Выпадающий список	Да	Тип данных, которые будут приходить на вход. Нужен для построения корректных связей с другими блоками. По умолчанию установлен тип "Объект".
Обязательный	Флаг	Да	Обязательность входа. Если вход обязательный и на него не пришло данных – блок не будет рассчитан с ошибкой
Массив	Флаг	Да	Является ли вход массивом данных

6.1.4.10.5 Выходы

В блоке нет выходов.

6.1.4.10.6 События блока

Название выхода	Тип	Описание
Событие смены выделения		

Объекты	Объекты	Используется для динамического обновления визуализаторов, при изменении отметки в фильтре. Для возможности корректного определения справочника, чьи значения необходимо изменить, в атрибуты каждого объекта добавляется поле "**_section**" со значением из поля **input_key** в привязках.
		
		`{`
		`  "pkey": {`
		`  "id": 1`
		`  },`
		`  "attrs": {`

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название выхода	Тип	Описание

		"ord": 1,
		"name": "U 1",
		"_section": "units" <- Добавленное поле
		},
		"_params": {},
		"_source": "f208267c-a2e0-4df5-b8f7-63313da7b454"
		}
		В случае отображения строки ввода (включен параметр **inputLabel**), в качестве результата возвращается объект без ключей, с двумя атрибутами: value - результат ввода и type - тип данных (поле **inputLabelType**)
		
		{
		"pkey": {},
		"attrs": {
		"value": "ВВеденный текс",
		"type": <inputLabelType>,
		"_section": "units" <- Добавленное поле
		},
		"_params": {}
		}

Параметры расположения визуализаторов (привязки)	Строка	Передается только при выборе «**Панель OLAP**» (п. 6.1.4.8) в режиме отображения. Строковое описание настройки «**Параметры привязки**». Используется блоком «&Источник данных» (п. 6.3.1) для формирования рядов данных

6.1.4.10.7 Описание работы блока

1. Блок формирует визуализатор согласно настройкам. В визуализатор попадают все элементы справочников, которые приходят на вход блока (при открытии в пресете может происходить фильтрация, в таком случае может отображаться только часть элементов). Визуализатор также содержит информацию о том, какие элементы должны быть отмечены по умолчанию.
2. Блок формирует событие смены отметки:
    a. При расчете графа в событии передаются объекты согласно отметке по умолчанию (значение «default» в параметрах привязок блока в графе);
    b. При вызове события в пресете передаются объекты согласно текущей пользовательской отметке.

6.1.4.10.7.1 Работа панели в режиме OLAP

При выборе «Панель OLAP» в режиме отображения, блок создает привязки для формирования рядов данных с учетом расположения справочников на визуализаторе (по столбцам, строкам или фиксированные). Однако, у самого блока нет входа для получения изначальных рядов данных, как нет и функционала для их формирования. Поэтому, в таком режиме, блок должен работать в связке с другим блоком «&Источник данных» (п. 6.3.1).

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

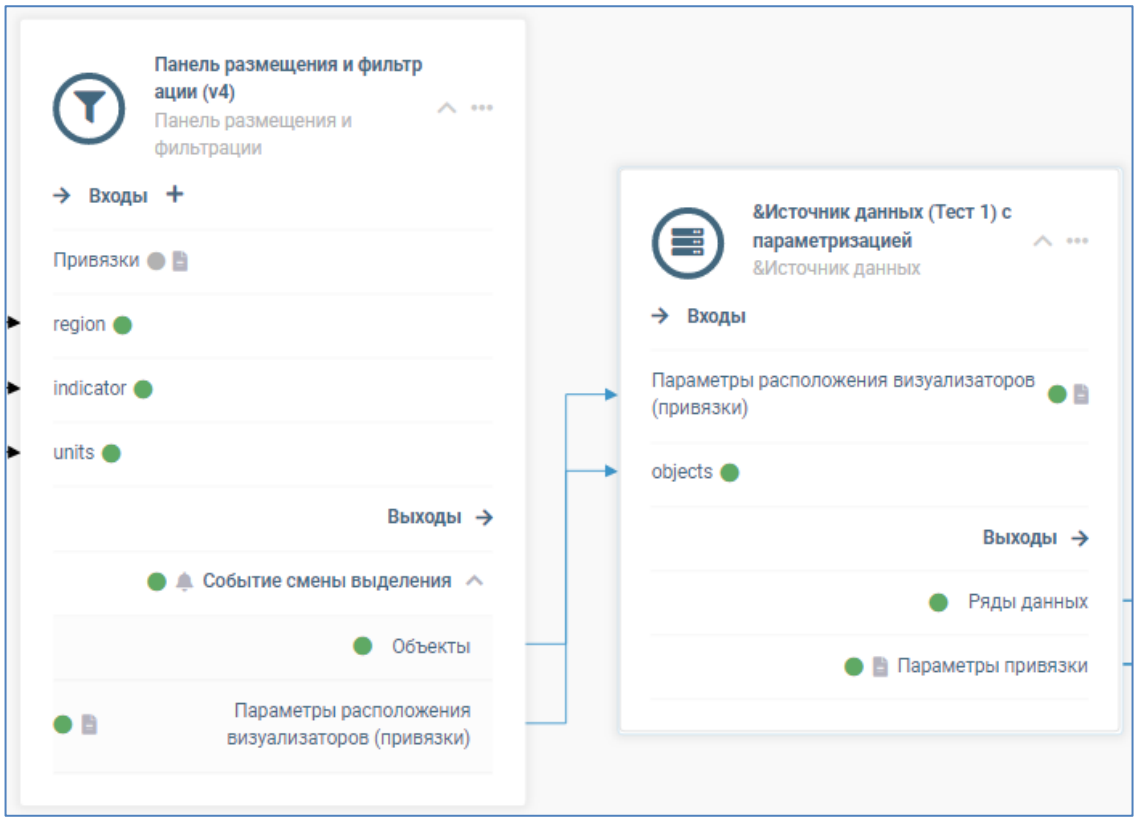


Рисунок 235 – Режим отображения блока «Панель OLAP» в связке с блоком «&Источник данных»

Блок «Панель фильтрации и размещения» создает параметры привязки, а блок «&Источник данных» формирует ряды на основе этих привязок.

Для формирования данных блок «&Источник данных» проверяет наличие данных на своем входе «**Параметры расположения визуализаторов (привязки)**».

Это поле может быть передано не только через панель фильтрации и размещения, но и через другие блоки, к примеру быть сгенерированным в блоке «Python» (п. 6.9.1).

В мастере настройки OLAP (п. 7.8.2) этих двух блоков определяется системой как блок OLAP с сохранением всего функционала.

Между выходом «**Параметры расположения визуализаторов (привязки)**» блока фильтрации и аналогичным входом блока OLAP не должно быть других блоков.

Несмотря на то, что в мастере настройки OLAP доступны так же и блоки, с выбранным режимом работы «Фильтр», не рекомендуется работать с ними в этом режиме, т.к. после завершения работы мастера не гарантируется корректная работа блока (в режиме фильтра не создается выход с привязками, который обеспечивает изменение структуры ряда).

6.1.4.10.8 Описание работы визуализатора

В зависимости от выбранного режима отображения, блок может использовать два различных визуализатора: панель фильтрации (п. 6.1.4.11) или панель OLAP (п. 6.1.4.8).

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

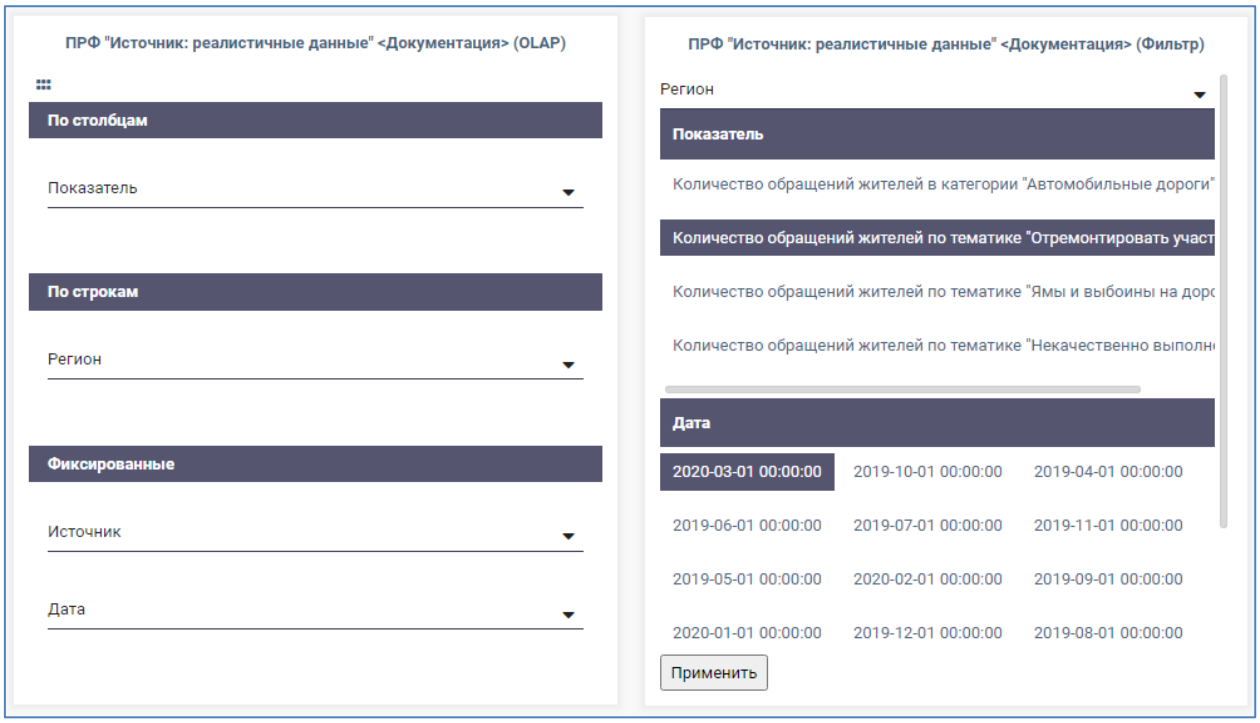


Рисунок 236 – Внешний вид визуализатора «Панель размещения и фильтрации»

Вне зависимости от выбранного режима работы, событийный выход «**Объекты**» будет возвращать массив с выделенными объектами.

Обратите внимание, что порядок объектов в массиве может меняться (к примеру, от настроенного порядка справочников в визуализаторе OLAP).

Однако, выход «**Параметры расположения визуализаторов**» будет использоваться только в режиме отображения OLAP. В режиме фильтра никакие данные не будут приходить на него.

6.1.4.10.9 Контекстное меню

В зависимости от значения настройки **combobox**, при открытии контекстного меню справочника (чтобы открыть меню, надо кликнуть ПКМ на любой из элементов справочника) оно имеет соответствующий перечень элементов.

Если combobox = 1, т.е. справочник имеет вид выпадающего списка, то меню выглядит следующим образом:

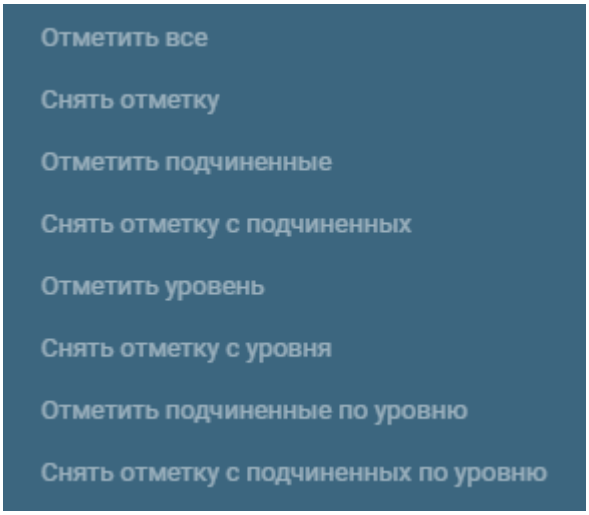


Рисунок 237 – Внешний вид контекстного меню выпадающего списка

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

- Отметить все - отмечает все элементы справочника
- Снять отметку - снимает отметку со всех элементов справочника
- Отметить подчиненные (для случая иерархического справочника) - отмечает все подчиненные элементы
- Снять отметку с подчиненных (для случая иерархического справочника) - снимает отметку со всех подчиненных элементов
- Отметить уровень (для случая иерархического справочника) - отмечает элементы на одном уровне иерархии
- Снять отметку с уровня (для случая иерархического справочника) - снимает отметку с элементов на одном уровне иерархии
- Отметить подчиненные по уровню (для случая иерархического справочника) - отмечает элементы на уровне иерархии на 1 ниже
- Снять отметку с подчиненных по уровню (для случая иерархического справочника) - снимает отметку с элементов на уровне иерархии на 1 ниже

Если combobox = 0, т.е. справочник имеет стандартный вид, то меню выглядит следующим образом:

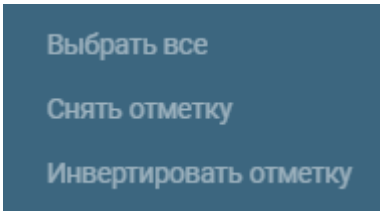


Рисунок 238 – Внешний вид стандартного контекстного меню

- Выбрать все - отмечает все элементы справочника
- Снять отметку - снимает отметку со всех элементов справочника
- Инвертировать отметку - отмечает невыделенные элементы и снимает отметку с выделенных элементов

6.1.4.10.10 Диапазон дат

Справочники могут поддерживать отметку диапазона дат (значение "date_range" в параметрах привязок блока в графе). Границы, внутри которых можно отметить диапазон, задаются в параметрах привязок блока в графе ("date_range_start","date_range_finish"). Динамика, которая будет использоваться, задается в параметрах привязок блока в графе ("date_range_type"). При включенном значении "date_range": 1.

Диапазон строится исходя из настроек параметров (date_range_type, date_range_start, date_range_finish). Если "date_range": 0, то справочником его не построить. Значение "input_key" может быть произвольным, но уникальным. Для диапазона дат можно задавать любое значение, независящее от входов.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

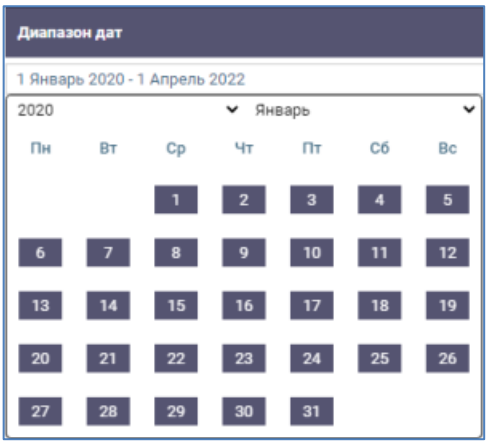


Рисунок 239 – Справочник «Диапазон дат»

Нажатие на дату первый раз выбирает первую дату диапазона, второй – вторую. При нажатии на одну дату – диапазон из одной даты. Следующий клик снова выбирает первую дату и сбрасывает предыдущий диапазон.

При выборе одной даты в заголовок будет выведена одна дата, а не диапазон.

При изменении гранулярности, например: "месяцы" -> "дни", выбранный период сохраняется, т.е. например "Январь 2022"-"Февраль 2022" -> "1 января 2022"-"28 февраля 2022". При выборе периода на границах диапазона (date_range_start и date_range_finish, указанные в привязках) изменение гранулярности будет учитывать эти границы.

6.1.4.10.11 Пользовательские фильтры

Фильтры сохраняются для каждого отдельного пользователя, т.е. на одном блоке у двух разных пользователей могут быть разные фильтры.

При включении настройки "Фильтры пользователя" в визуализаторе появляется выпадающий список с сохраненными пользовательскими фильтрами:

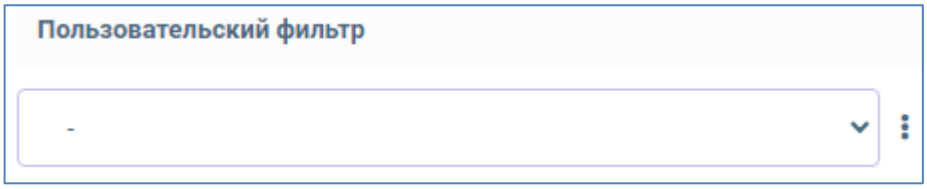


Рисунок 240 – Пример пользовательского фильтра

Для редактирования/удаления фильтра необходимо нажать на кнопку с тремя точками ( ), расположенную слева от выпадающего списка.

Возможные настройки:

- Сохранить фильтр - сохраняет отметку для пользователя.
- Удалить фильтр (только при выбранной отметке) - удаляет отметку.

При любом изменении отметки (ручном или в результате расчета) в комбобоксе фильтр сбрасывается, то есть выводится элемент "-"

6.1.4.10.12 Настройки визуализатора

Специфические настройки:

1. Разделы:

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

a. Размер шрифта заголовка секции
b. Цвет текста заголовка секции
c. Цвет фона заголовка раздела

2. Кнопки:
   a. Размер шрифта текста кнопок
   b. Цвет текста кнопок
   c. Цвет фона кнопок
   d. Цвет текста активной кнопки
   e. Цвет фона активной кнопки
   f. Скругление углов кнопок

Рисунок 241 – Настройки визуализатора (раздел «Разделы»)

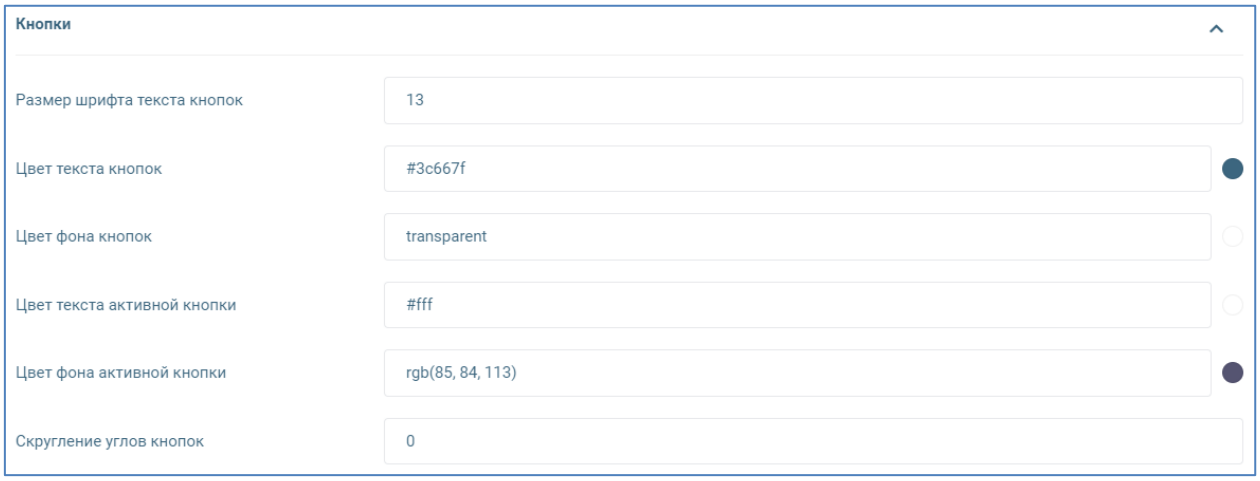


Рисунок 242 – Настройки визуализатора (раздел «Кнопки»)

Общий список настроек приведен в п. 4.2.8 «Настройки визуализатора».

Помимо стандартных, визуализатор позволяет задать следующие настройки в разделе «Кнопки».

Название параметра	Тип параметра	Обязательность	Описание	Пример
Размер шрифта текста кнопок	Целое число	Да	Задает размер заголовка секции аккордеона	
Цвет текста кнопок	Цвет	Да	Задает цвет текста для заголовка секции аккордеона	rgb(255, 255, 255)

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательность	Описание	Пример
Цвет фона кнопок	Цвет	Да	Задает цвет фона для заголовка секции аккордеона	rgba(214, 33, 33, 0.87)
Цвет текста активной кнопки	Цвет	Да	Цвет текста ля активной кнопки	#fff
Цвет фона активной кнопки	Цвет	Да	Цвет фона активной кнопки	rgb(85, 84, 113)
Скругление углов кнопок	Целое число	Да	Размер скругления углов	0

6.1.4.11 Блок «Панель фильтрации»

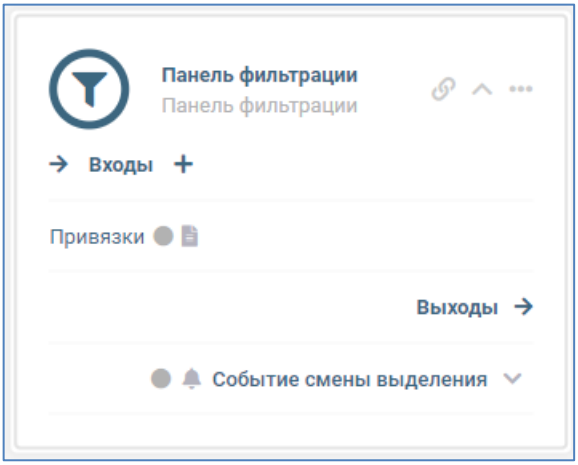


Рисунок 243 – Пример блока «Панель фильтраци»

6.1.4.11.1 Назначение

Блок позволяет формировать визуализатор для отображения отметок у выбранных справочников. Пользователь может управлять отметкой, при этом зависимые визуализаторы будут обновлены с актуальными данными.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.1.4.11.2 Настройки блока

Название параметра	Тип параметра	Обязательный	Описание	Пример
Параметры привязки	JSON	Да	Параметры привязки справочников Параметры привязки [<Массив описаний. Количество описаний должно совпадать с количеством справочников на входе.> { "input_key": Идентификатор входа справочника "input_name": Наименование справочника в визуализаторе. Если не задано, используется значение по умолчанию: "Справочник N", где N - порядковый номер справочника "key": Идентификатор поля в справочнике, который использовать как ключ. Если не задано берется поле по умолчанию - id "name": Идентификатор поля в справочнике, который использовать как наименование. Если не задано берется поле по умолчанию - name "default": Ключи элементов, которые необходимо отмечать по умолчанию. Если поле не задано - отмечается первый элемент (или все элементы при включенной мультиотметке) "multiselect": Целое число, 1 или 0. Включение множественной отметки у справочника. Если не задано - множественная отметка выключена "combobox": Целое число, 1 или 0. Отображение справочника в виде выпадающего списка. По умолчанию - выпадающий список выключен "hierarchy": Целое число, 1 или 0. Отображение данных в виде иерархии. Без включенного флага будут отображаться только данные на первом уровне. При включении иерархии автоматически включается выпадающий список. Отедльно флаг прописывать не нужно. По умолчанию - выключено. "filter": Массив словарей. Отображение элементов данного справочника зависит от другого	Параметры привязки [{ "input_key": "period", "input_name": "Период", "key": "id", "name": "name", "default": 1, "multiselect": 1 }, { "input_key": "year", "key": "id", "name": "name", "default": 1, "combobox": 1 }, { "input_key": "hier", "input_name": "Иерархия", "default": [3], "multiselect": 1, "hierarchy": 1 }, { "input_key": "reg", "input_name": "Регионы", "key": "id",

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример

			"from": Идентификатор справочника, от которого зависит данный справочник
			"foreignAttr": Идентификатор поля в справочнике, от которого зависит данный справочник, по которому будут фильтроваться элементы
			"localAttr": Идентификатор поля в данном справочнике, по которому будет идти поиск соответсвия с "foreignAttr"
			"addFields": Массив строк. Состоит из идентификаторов полей в справочнике, по которым идет фильтрация (все значения "localAttr")
			"date_range": Целое число, 1 или 0. Использование справочника для выбора диапазона дат
			"date_range_type": Возможные значения: 1 - год, 2 - полугодие, 3 - квартал, 4 - месяц, 5 - день
			"date_range_start": Дата - начало диапазона в справочнике
			"date_range_finish": Дата - конец диапазона в справочнике
			}
			]
				"name": "name",
				"multiselect": 1,
				"filter": [
				{
				"from": "year",
				"foreignAttr": "id",
				"localAttr": "region_years"
				}
				],
				"addFields": [
				"region_years"
				]
				},
				{
				"input_key": "range",
				"input_name": "Диапазон дат",
				"multiselect": 0,
				"date_range": 1,
				"date_range_type": 5,
				"date_range_start": "17.05.2019",
				"date_range_finish": "01.01.2025",
				"default": ["01.01.2020", "01.04.2022"]
				}
				]

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
Отображать кнопку «Применить"	Флаг	Нет	С включенным флагом отображается кнопка «Применить». Событие смены выделения отправляется после нажатия данной кнопки. С выключенным флагом не отображается кнопка «Применить». Событие смены выделения отправляется сразу после изменения отметки	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.1.4.11.3 Группа «Общие настройки блока»

Аналогично приведенным в п. 6.1.1.1.3.

6.1.4.11.4 Входы

Каждый новый справочник добавляется в отдельный вход. Параметры «Обязательный" и «Массив" должны быть включены. Параметр «Идентификатор» необходимо указать в поле **input_key** в параметрах привязки.

Для создания входа необходимо нажать на знак + в разделе входов блока. Для каждого из входов задается следующий перечень полей.

Название входа	Тип	Обязательный	Описание
Идентификатор	Строка	Да	Идентификатор входа – используется внутри блока, для получения доступа к данным входа
Имя	Строка	Да	Имя входа для отображения в списке входов блока
Описание	Строка	Нет	Дополнительная информация с описанием входа
Тип	Выпадающий список	Да	Тип данных, которые будут приходить на вход. Нужен для построения корректных связей с другими блоками. По умолчанию установлен тип "Объект".
Обязательный	Флаг	Да	Обязательность входа. Если вход обязательный и на него не пришло данных – блок не будет рассчитан с ошибкой
Массив	Флаг	Да	Является ли вход массивом данных

6.1.4.11.5 Выходы

В блоке нет выходов.

6.1.4.11.6 События блока

Название выхода	Тип	Описание
Событие смены выделения	Объекты	Используется для динамического обновления визуализаторов, при изменении отметки в фильтре

6.1.4.11.7 Описание работы блока

1. Блок формирует визуализатор согласно настройкам. В визуализатор попадают все элементы справочников, которые приходят на вход блока (при открытии в пресете может происходить фильтрация, в таком случае может отображаться только часть элементов). Визуализатор также содержит информацию о том, какие элементы должны быть отмечены по умолчанию.
2. Блок формирует событие смены выделения:

- При расчете графа в событии передаются объекты согласно отметке по умолчанию (значение «default» в параметрах привязок блока в графе).

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

- При вызове события в пресете передаются объекты согласно текущей пользовательской отметке. Диапазоны дат передаются с помощью «selected_range_start», «selected_range_finish».

6.1.4.11.8 Описание работы визуализатора

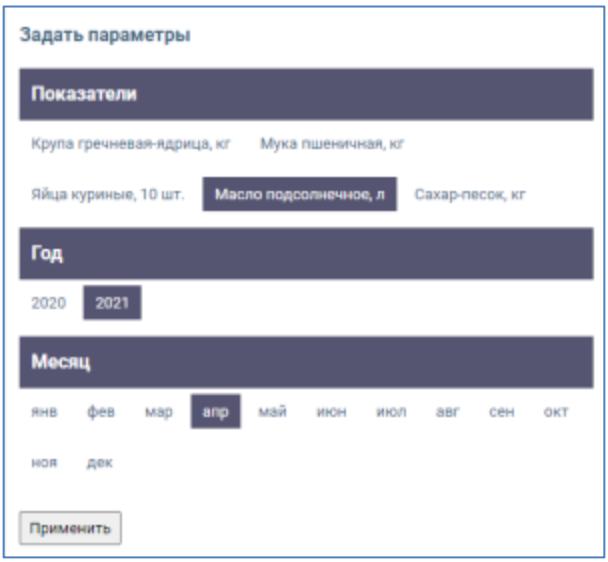


Рисунок 244 – Внешний вид визуализатора «Панель фильтрации»

1. Справочники могут отображаться в виде списка или выпадающего списка (значение «combobox» в параметрах привязок блока в графе).
2. Справочники в виде выпадающего списка могут иметь иерархию (значение «hierarchy» в параметрах привязок блока в графе).
3. При открытии пресета в панели фильтрации устанавливается отметка по умолчанию (значение «default» в параметрах привязок блока в графе).
4. В справочнике может быть отмечено один или несколько элементов (значение «multiselect» в параметрах привязок блока в графе).
5. Список отображаемых элементов одного справочника может зависеть от отметки в другом (значение «filter» в параметрах привязок блока в графе).
6. Чтобы вызвать событие смены выделения, нужно отметить желаемые элементы и нажать «Применить» (Событие смены выделения отправляется сразу после изменения отметки, если выключен параметр «Отображать кнопку «Применить»).
7. Чтобы одновременно отметить или снять отметку для нескольких элементов, есть возможность открыть контекстное меню с помощью нажатия ПКМ и выбрать опцию, в соответствии с которой будет произведена отметка. Если справочник отображается в виде списка, то контекстное меню содержит следующие пункты:
    - Выбрать все;
    - Снять отметку;
    - Инвертировать отметку.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

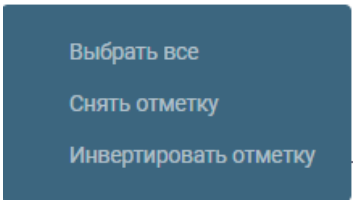


Рисунок 245 – Контекстное меню

Если справочник отображается в виде выпадающего списка, то контекстное меню содержит следующие пункты:

1. Отметить все.
2. Снять отметку.
3. Отметить подчиненные.
4. Снять отметку с подчиненных.
5. Отметить уровень.
6. Снять отметку с уровня.
7. Отметить подчиненные по уровню.
8. Снять отметку с подчиненных по уровню.

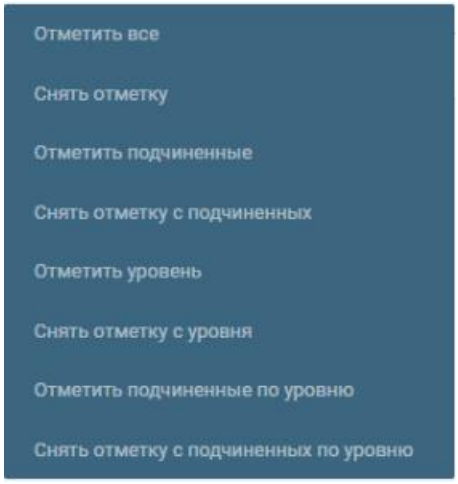


Рисунок 246 – Контекстное меню

- Справочники могут поддерживать отметку диапазона дат (значение «date_range» в параметрах привязок блока в графе). Границы, внутри которых можно отметить диапазон, задаются в параметрах привязок блока в графе ("date_range_start","date_range_finish»). Динамика, которая будет использоваться, задается в параметрах привязок блока в графе ("date_range_type»).

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

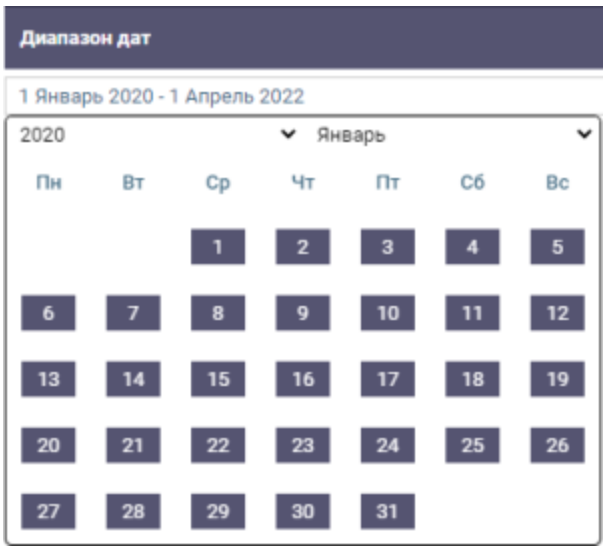


Рисунок 247 – Визуализатор блока «Панель фильтрации»

Нажатие на дату первый раз выбирает первую дату диапазона, второй – вторую. При нажатии на одну дату – диапазон из одной даты. Следующий клик снова выбирает первую дату и сбрасывает предыдущий диапазон.

При выборе одной даты в заголовок будет выведена одна дата, а не диапазон.

- При изменении гранулярности, например "месяцы" -> "дни", выбранный период сохраняется, т.е. например "Январь 2022"-"Февраль 2022" -> "1 января 2022"-"28 февраля 2022". При выборе периода на границах диапазона (date_range_start и date_range_finish, указанные в привязках) изменение гранулярности будет учитывать эти границы.

6.1.4.11.9 Настройки визуализатора

Специфические настройки:

1. Разделы:
    a. Размер шрифта заголовка секции
    b. Цвет текста заголовка секции
    c. Цвет фона заголовка раздела
2. Кнопки:
    a. Размер шрифта текста кнопок
    b. Цвет текста кнопок
    c. Цвет фона кнопок
    d. Цвет текста активной кнопки
    e. Цвет фона активной кнопки
    f. Скругление углов кнопок

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Рисунок 248 – Настройки визуализатора (раздел «Разделы»)

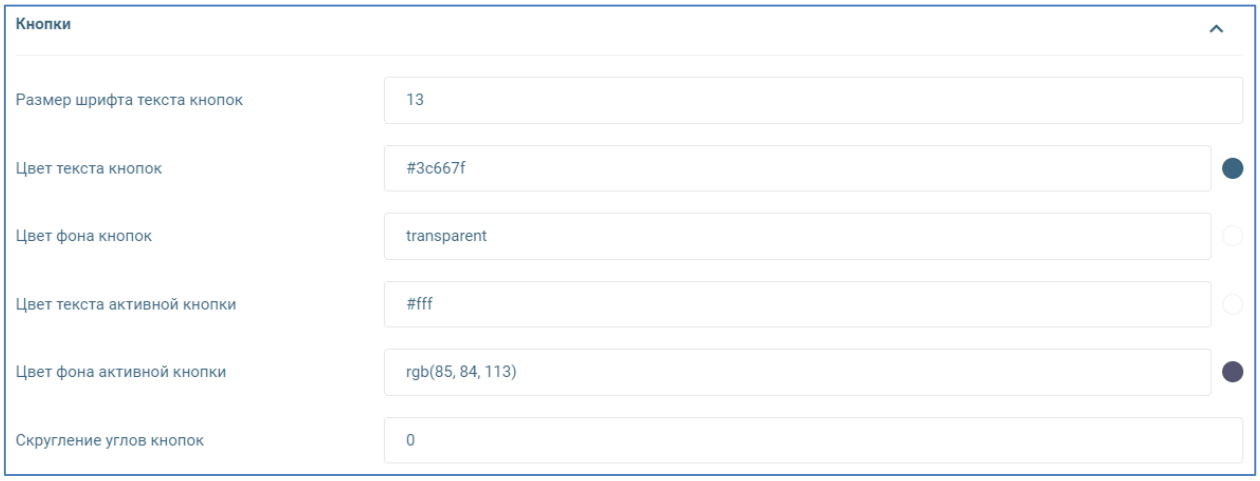


Рисунок 249 – Настройки визуализатора (раздел «Кнопки»)

Общий список приведен в описании настроек визуализатора (см. п. 6.1.4.14.9).

Помимо стандартных, визуализатор позволяет задать следующие настройки в разделе «Кнопки».

Название параметра	**Тип параметра**	**Обязательный**	**Описание**	**Пример**
Размер шрифта текста кнопок	Целое число	Да	Задает размер заголовка секции аккордеона	
Цвет текста кнопок	Цвет	Да	Задает цвет текста для заголовка секции аккордеона	rgb(255, 255, 255)
Цвет фона кнопок	Цвет	Да	Задает цвет фона для заголовка секции аккордеона	rgba(214, 33, 33, 0.87)
Цвет текста активной кнопки	Цвет	Да	Цвет текста ля активной кнопки	#fff
Цвет фона активной кнопки	Цвет	Да	Цвет фона активной кнопки	rgb(85, 84, 113)
Скругление углов кнопок	Целое число	Да	Размер скругления углов	0

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.1.4.12 Блок «Периоды идентификации и прогнозирования»

6.1.4.12.1 Назначение

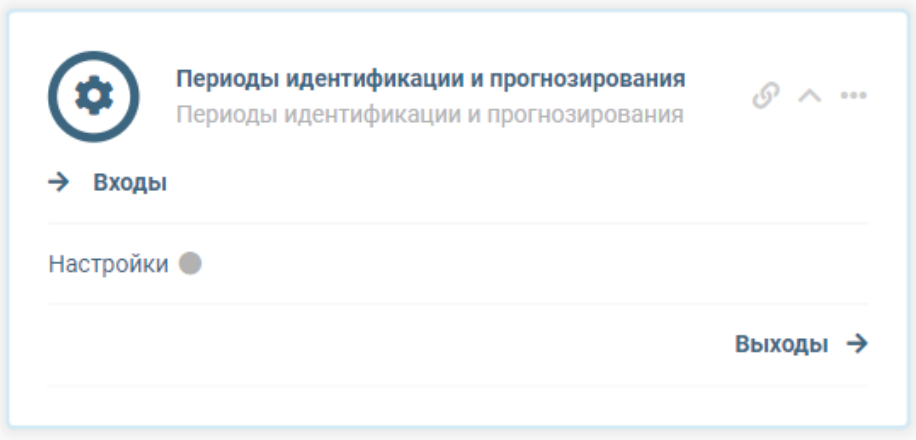


Рисунок 250 – Пример блока «Периоды идентификации и прогнозирования»

Блок предназначен для установки настроек параметров задачи работы с датами, которые можно передать в статистические блоки.

Блок «Периоды идентификации и прогнозирования» может быть размещен на графе не более одного раза.

6.1.4.12.2 Настройки блока

Название параметра	Тип параметра	Обязательный	Описание	Пример
Основные свойства				
Наименование блока	Строка (string)	Нет	Наименование блока, которое отображается на графе. В панели размещенных блоков возможен по нему поиск	
Заголовок блока	Строка (string)	Нет	Наименование, которое будет отображаться в пользовательском приложении, в заголовке блока	
Период моделирования				
Начальная дата	Дата и время (datatime)	Нет	«Начальная дата» должна быть меньше «Последней даты»	
Последняя дата	Дата и время (datatime)	Нет		
Период прогнозирования				
Начальная дата	Дата и время (datatime)	Нет	«Начальная дата» должна быть меньше «Последней даты»	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
Последняя дата	Дата и время (Datatime)	Нет		

6.1.4.12.3 Группа «Общие настройки блока»

Аналогично приведенным в п. 6.1.1.1.3.

6.1.4.12.4 Входы

Название входа	Тип	Обязательный	Описание
Настройки	Объект (object)	Нет	Данные, которые установятся в настройки блока

6.1.4.12.5 Выходы

В блоке нет выходов.

6.1.4.12.6 События блока

В блоке нет событий.

6.1.4.12.7 Описание работы блока

Блок содержит настройки к которым обращаются статистические блоки. Чтобы подключить эти настройки в стат. блоках, необходимо:

1. Зайти в настройку статистического блока.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

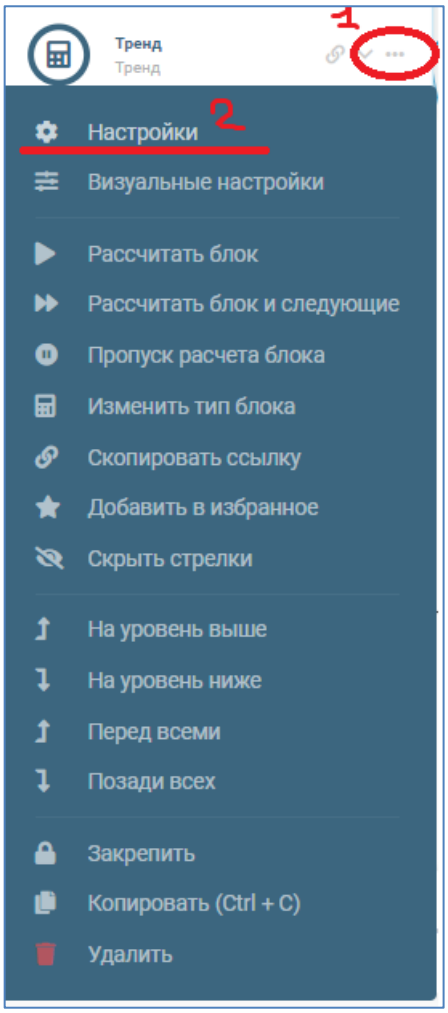


Рисунок 251 – Выбор раздела меню «Настройки»

2. Выбрать в блоке «Структурный тип данных» в параметре значение «Спецификация» значение «Даты».

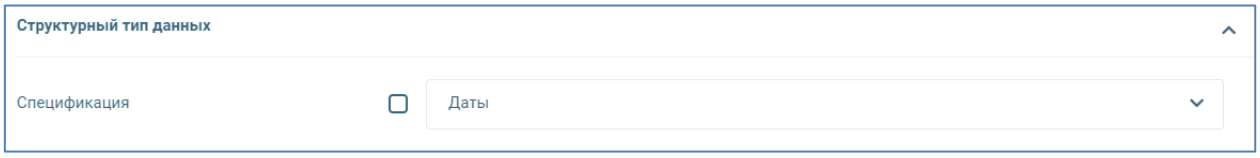


Рисунок 252 – Блок «Структурный тип данных»

3. Поставить флаг «Использовать параметр задачи», у тех параметров, которые будут брать настройки из блока «Периоды идентификации и прогнозирования».

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

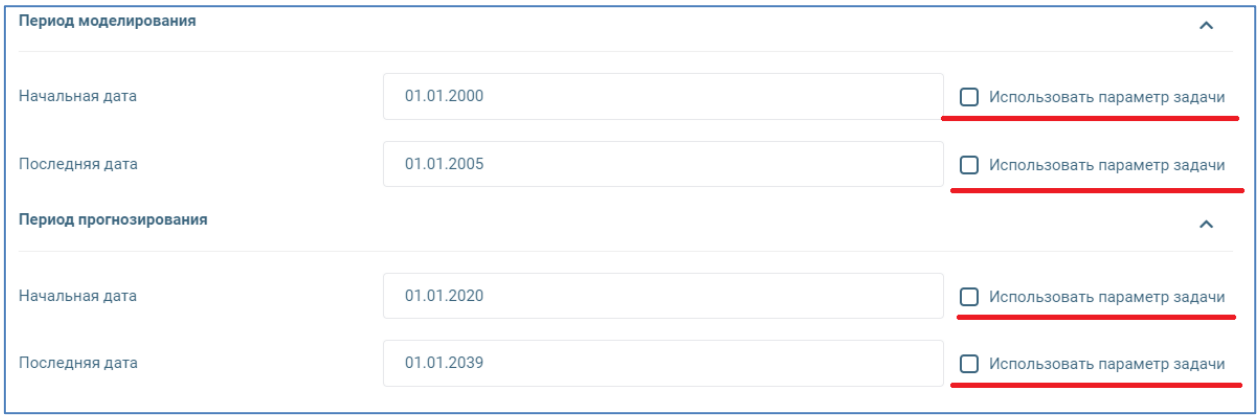


Рисунок 253 – Блок «Периоды идентификации и прогнозирования»

4. Сохранить настройки.

Так же значения можно задавать не только в настройках блока «Периоды идентификации и прогнозирования», но и через его входные данные. Для этого надо на вход блока передать объект с параметрами:

- modeling_date_start – дата старта периода моделирования;
- modeling_date_end – дата окончания периода моделирования;
- forecast_date_start – дата старта периода прогнозирования;
- forecast_date_end – дата окончания периода прогнозирования.

Пример для входа «Настройки»:

```
[
    {
        "pkey": {},
        "_params": {
            "lazy": []
        },
        "attrs": {
            "modeling_date_start": "01.01.2000 00:00:00",
            "modeling_date_end": "01.01.2024 10:00",
            "forecast_date_start": "01.01.2025 11",
            "forecast_date_end": "01.01.2029"
        },
        "child": [],
        "_source": "39cc26d4-f40d-4c7f-b728-9ccc13d687cc"
    }
]
```

В этом случае значения будут установлены для всей задачи в момент расчета этого блока. Таким образом можно динамически менять их при срабатывании какого-либо события.

Так же к настройкам блока «Периоды идентификации и прогнозирования» можно обратиться в jinja шаблоне (в тех блоках, которые ее поддерживают).

Пример обращения к настройкам блока в **Jinja**.

Наименование параметра	Пример использования в запросах и параметрах блоков
Параметры	
Маска даты	{{ global.parameters.mask }}

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Период	{{ global.parameters.frequency }}
Период моделирования	
Начальная дата	{{ global.modeling_period.modeling_date_start }}
Последняя дата	{{ global.modeling_period.modeling_date_end }}
Период прогнозирования	
Начальная дата	{{ global.forecast_period.forecast_date_start }}
Последняя дата	{{ global.forecast_period.forecast_date_end }}

6.1.4.13 Блок «Письмо (email)»

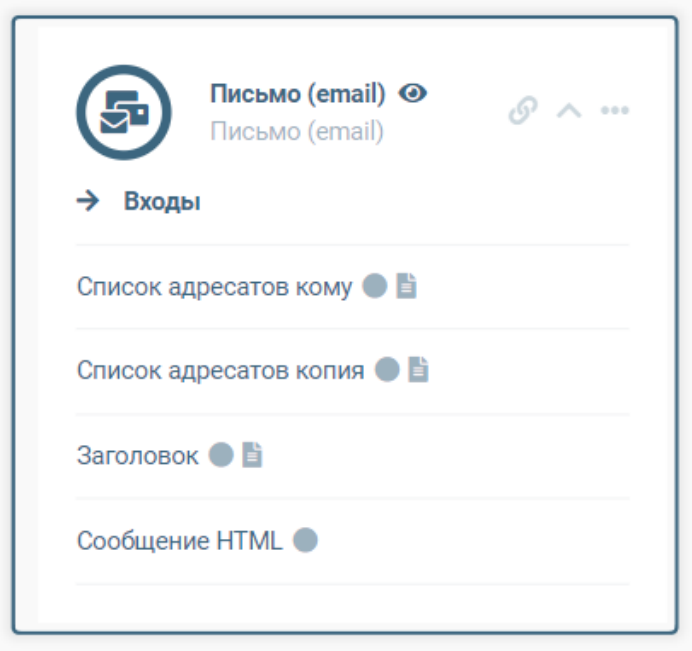


Рисунок 254 – Блок «Письмо (email)»

6.1.4.13.1 Назначение

Блок позволяет формировать и отправлять сообщения на e-mail адресатов.

6.1.4.13.2 Настройки блока

Название параметра	Тип параметра	Обязательный	Описание	Пример
Адрес SMTP сервера	Строка	Да	Адрес сервера отправки сообщений	
Порт	Строка	Да	Порт сервера отправки сообщения	
Логин	Строка	Да	Имя учетной записи, которая будет отправлять сообщения	
Пароль	Шифрованная строка	Да	Пароль учетной записи, которая будет отправлять сообщения	
Настройки сообщения				

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
От	Строка	Нет	Адрес отправителя. Если поле не задано – используется адрес, который указан в поле Логин.	
Список адресатов кому	Строка	Нет	Список адресатов для получения письма, основной. Несколько адресов необходимо разделить точкой с запятой ';'. Если на вход «Список адресов кому» пришли значения – данная настройка игнорируется	
Список адресатов копия	Флажок	Нет	Список адресатов для получения письма, копия. Несколько адресов необходимо разделить точкой с запятой ';'. Если на вход «Список адресов копия» пришли значения – данная настройка игнорируется	
Заголовок	Строка	Нет	Заголовок сообщения e-mail. Если на вход «Заголовок» пришли значения – данная настройка игнорируется	
Сообщение HTML	Строка HTML	Нет	Тело письма в формате HTML. Если на вход «Сообщение HTML» пришло значение – данная настройка игнорируется	

6.1.4.13.3 Группа «Общие настройки блока»

Аналогично приведенным в п. 6.1.1.1.3.

6.1.4.13.4 Входы

Название входа	Тип	Обязательный	Описание
Список адресов кому	Строка	Нет	Список адресов для отправки, основной. Список передается через точку с запятой ';'. Если на вход пришло значение, одноименная настройка внутри блока игнорируется
Список адресов копия	Строка	Нет	Список адресов для отправки, копия. Список передается через точку с запятой ';'. Если на вход пришло значение, одноименная настройка внутри блока игнорируется

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название входа	Тип	Обязательный	Описание
Заголовок	Строка	Нет	Заголовок письма. Если на вход пришло значение, одноименная настройка внутри блока игнорируется
Сообщения HTML	Строка HTML	Нет	Сообщение для отправки в письме. Если на вход пришло значение, одноименная настройка внутри блока игнорируется

6.1.4.13.5 Выходы

В блоке нет выходов.

6.1.4.13.6 События блока

В блоке нет событий.

6.1.4.13.7 Описание работы блока

Блока отправляет на переданные или указанные в настройках адреса письмо с содержимым из настройки «Сообщение HTML». При этом все содержимое письма можно генерировать автоматически, например в блоках Python.

6.1.4.14 Блок «Скачать файл»

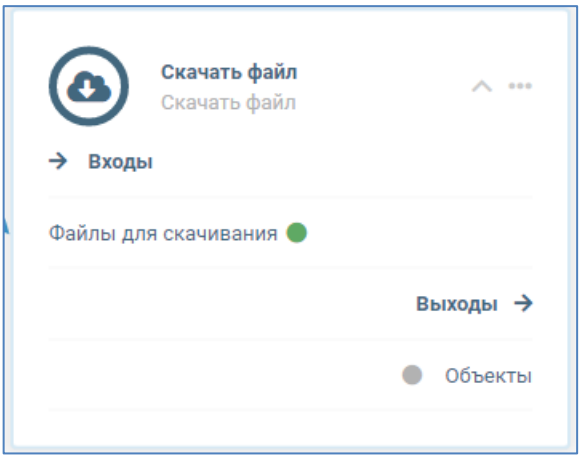


Рисунок 255 – Пример Блока «Скачать файл»

6.1.4.14.1 Назначение

Блок позволяет выполнить скачивание одного или нескольких файлов с сервера. Возможно, как отображение блока в качестве визуализатора, так и загрузка файлов без визуализатора.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.1.4.14.2 Настройки блока

Название параметра	Тип параметра	Обязательный	Описание	Пример
Вид отображения визуализатора	Выпадающий список	Да	Выбор режима отображения визуализатора. Возможные режимы отображения: • Ссылки • Визуализатор отображается в пресете. • Скрытый Визуализатор отображается только при редактировании пресета	

6.1.4.14.3 Группа «Общие настройки блока»

Аналогично приведенным в п. 6.1.1.1.3.

6.1.4.14.4 Входы

Название входа	Тип	Обязательный	Описание
Файлы для скачивания	JSON	Да	JSON со ссылками файлов на сервере, которые необходимо скачать. Файл должен находится внутри той же задачи, в какой размещен и сам блок. Формат JSON: Пример входа с файлами для скачивания: {"fileNames": ["doc/test_file_1--2021-10-18--05-09-53.docx",..., "doc/test_file_n--2021-10-19--05-09-53.docx"]}

6.1.4.14.5 Выходы

Название выхода	Тип	Обязательный	Описание
Объекты	Объекты	Нет	Массив объектов с информацией о файлах, доступных для скачивания

6.1.4.14.6 События блока

У блока нет событий.

6.1.4.14.7 Описание работы блока

1. Блок считывает все ссылки на файлы, которые пришли ему на вход в разделе **fileNames**.
2. Проверяется наличие файла на сервере. В случае отсутствия файла будет выдано предупреждение.
3. Блок формирует массив объектов с именами файла (переменная **name** в атрибутах) и ссылкой на файл (переменная **path**).

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

4. Блок формирует визуализатор, даже если выбран вид отображения «Скрытый».

6.1.4.14.8 Описание работы визуализатора

Можно разместить визуализатор в режиме редактирования пресета. При этом, для отображения будет доступен и визуализатор с видом отображения «Скрытый».

Если визуализатор не разместить, скачивание файлов будет недоступно.

- Два размещенных визуализатора блока. При этом у левого выбран вид отображения «Скрытый». Но в режиме редактирования пресета видны оба блока:

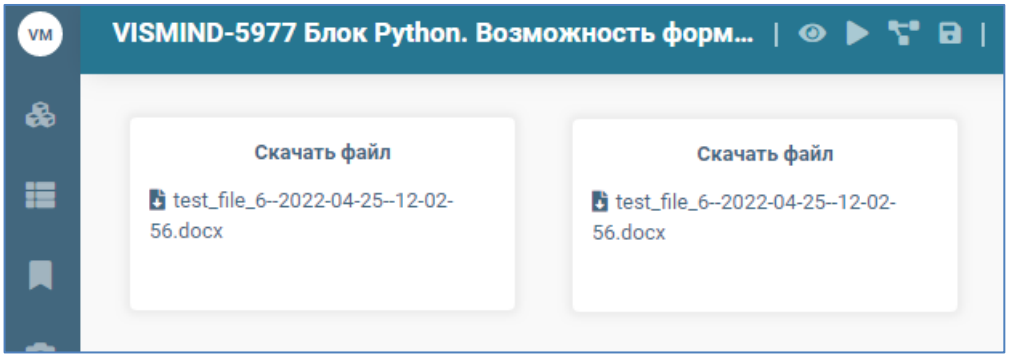


Рисунок 256 – Визуализатор в режиме редактирования пресета

- Однако, если перейти в режим просмотра пресета, то видимым остается только один блок. При этом, скачивание файлов доступно у обоих блоков:

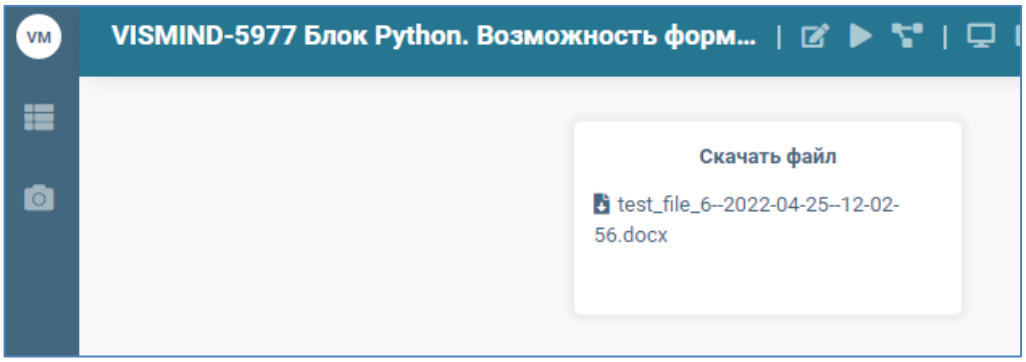


Рисунок 257 – Визуализатор в режиме просмотра пресета

Скачивание файлов из скрытого визуализатора происходит в момент расчета события, вызвать которое можно, к примеру, используя Блок «Кнопка» (п. 6.10.1).

Для скачивания файла в режиме отображения «Ссылки», необходимо нажать на значок скачивания, расположенный слева от названия файла.

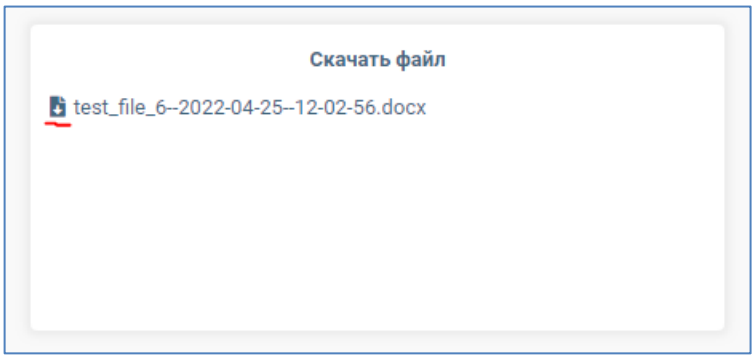


Рисунок 258 – Пример отображения значка скачивания

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.1.4.14.9 Настройки визуализатора

Общий список настроек приведен в п. 4.2.8 «Настройки визуализатора».

Помимо стандартных, визуализатор позволяет задать следующие настройки.

Размер шрифта текста кнопок	Целое число	Нет	Задает размер текста в кнопке
Цвет текста кнопок	Цвет	Нет	Задает цвет текста в кнопке
Цвет фона кнопок	Цвет	Нет	Задает цвет фона в кнопке
Цвет текста активной кнопки	Цвет	Нет	Задает цвет текста активной (нажатой) кнопки
Цвет фона активной кнопки	Цвет	Нет	Задает цвет фона активной (нажатой) кнопки
Скругление углов кнопок	Целое число	Нет	Задает скругление углов кнопки

6.1.4.15 Блок «Слайдер»

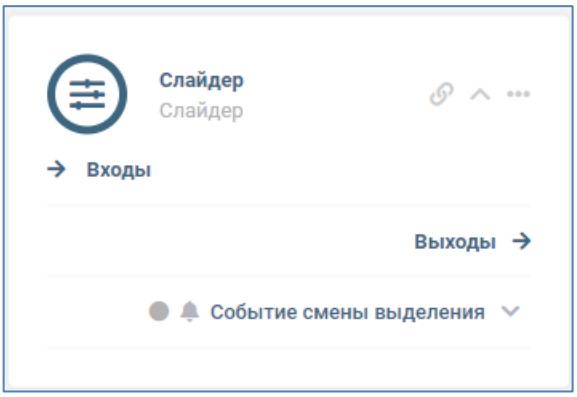


Рисунок 259 – Пример блока «Слайдер»

6.1.4.15.1 Назначение

Блок позволяет формировать слайдер, способный переключать значения в заданном диапазоне с заданным шагом.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.1.4.15.2 Настройки блока

Название параметра	Тип параметра	Обязательный	Описание	Пример
Группа «Основные свойства»				
Минимальное значение	Вещественное число	Да	Минимальное значение слайдера	
Максимальное значение	Вещественное число	Нет	Максимальное значение слайдера	
Стартовое значение	Вещественное число	Да	Стартовое значение слайдера	
Шаг слайдера	Вещественное число	Да	Величина на которую изменяется значение при перемещении бегунка на слайдере	
Группа « Диапазон"				
Стартовое значение верхней границы диапазона	Вещественное число	Нет	При указании значения в этом поле блок будет отображать диапазон значений (подробнее см. ниже). Без указания этого параметра, остальные настройки не будут использоваться. Для отключения диапазонов требуется удалить все значения из параметра и оставить пустую строку	
Минимальный размер диапазона	Вещественное число	Нет	Минимальный размер диапазона. Применяется только для диапазонов	
Максимальный размер диапазона	Вещественное число	Нет	Максимальный размер диапазона. Применяется только для диапазонов	
Без изменения порядка границ диапазона	Флаг	Да	Если выключен, то при изменении границы диапазона возможно передвинуть ползунок так, что нижняя граница станет верхней и наоборот. Применяется только для диапазонов	
Сдвиг диапазона	Флаг	Да	Если включен, то диапазон будет сдвигаться при изменении границы диапазона с сохранением минимального и максимального размеров диапазона	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
Группа «Деления"				
Отображать деления	Флаг	Да	Отображать деления на визуализаторе	
Отображать значения делений	Флаг	Да	Отображать значения делений на визуализаторе	
Шаг деления	Целое число	Нет	Шаг деления	
Список делений	Строка	Нет	Перечисление делений строкой, без пробелов с разделителем «,» (запятая)	5, 10, 15, 20, 25, 30, 35
Список шагов	JSON	Нет	Перечисление шагов массивом, с параметрами: • **value** – значение шага, число, обязательное; • **legend** – описание шага, строка, не обязательное.	[{"value": 0}, {"value": 10, «legend": «Плохо"}, {"value": 40}, {"value": 50, «legend": «Средне"}, {"value": 60}, {"value": 70, «legend": «Хорошо"}, {"value": 80}, {"value": 90, «legend": «Отлично"}, {"value": 100}]
Префикс значения деления	Строка	Нет		
Постфикс значения деления	Строка	Нет		
Группа «Метки"				
Скрывать метки границ	Флаг	Нет	Скрыть отображение меток начального и конечного значений	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
Скрывать метки указателя/диапазона	Флаг	Нет	Скрыть отображение меток текущего значения/диапазона	
Префикс метки текущего значения/границы диапазона	Строка	Нет	Допустимо использование тегов HTML	Значение:
Постфикс текущего значения/границы диапазона	Строка	Нет	Допустимо использование тегов HTML	руб.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.1.4.15.3 Группа «Общие настройки блока»

Аналогично приведенным в п. 6.1.1.1.3.

6.1.4.15.4 Входы

В блоке нет входов.

6.1.4.15.5 Выходы

В блоке нет выходов.

6.1.4.15.6 События

Название выхода	Тип	Описание
Событие смены выделения		
Ряды данных	Вещественное число	Событие передает отмеченное значение из элементов (вещественное число) на выход блока. При установке параметра «**Стартовое значение верхней границы диапазона**» (работа в режиме диапазонов) передает нижнюю границу диапазона.
Значение верхней границы диапазона	Вещественное число	Вход появляется только при установке значений в параметр «**Стартовое значение верхней границы диапазона**». Передает верхнюю границу диапазона

6.1.4.15.7 Описание работы блока

Блок может работать в режиме отметки диапазона. Для этого нужно указать любое число в параметре «**Стартовое значение верхней границы диапазона**». После чего у блока появляется дополнительный событийный выход:

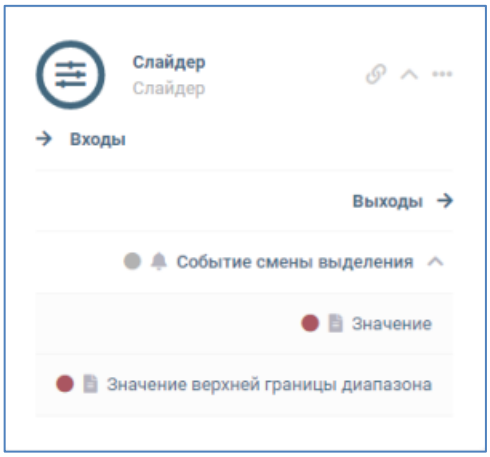


Рисунок 260 – Пример блока «Слайдер»

При удалении значения в параметре диапазона, выход (а так же все связи, исходящие из данного выхода) будут удалены.

6.1.4.15.8 Описание работы визуализатора

Блок используется для динамического изменения значения какого-либо показателя.

Изменение положения бегунка визуализатора приводит к пересчёту соединённых блоков и отображению новых результатов визуализаторов на пресете.

Так же установкой минимального значения в блоке «Слайдер», настраивается состояние визуализаторов при открытии пресета.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Для использования слайдера необходимо навести на бегунок курсор, зажать ЛКМ и перемещать мышь в горизонтальной плоскости.

6.1.4.15.9 Визуализатор «Слайдер»

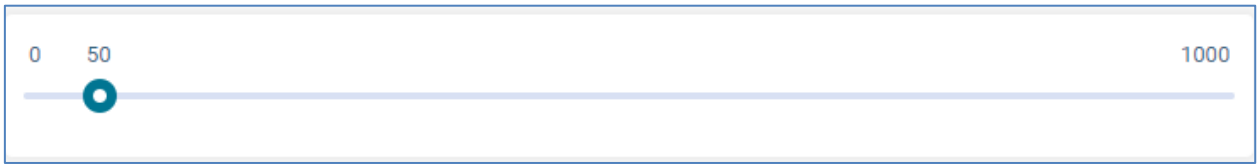


Рисунок 261 – Внешний вид визуализатора «Слайдер»

6.1.4.15.9.1 Настройки визуализатора

Общий список приведен в п. 4.2.1.3 «Общие настройки в режиме редактирования пресета».

6.1.4.16 Блок «События пресета»

6.1.4.16.1 Назначение

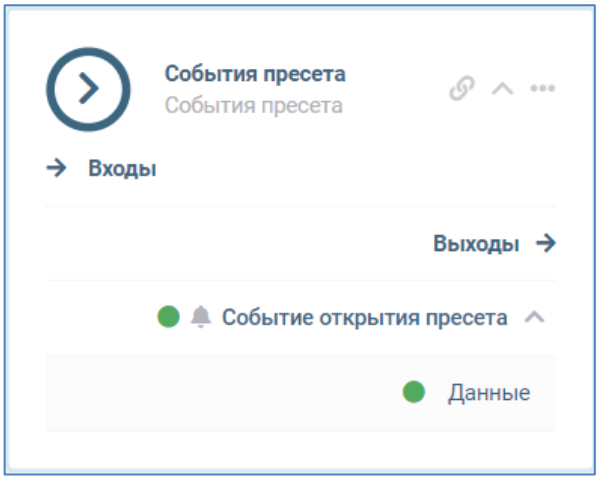


Рисунок 262 – Пример Блока «События пресета»

Запускает событие расчет ветки блоков за ним при открытии пресета

6.1.4.16.2 Настройки блока

Наименование	Тип данных	Примечание
Специфические настройки блока		
Метод расчета	Выпадающий список	Поле поддерживает единичную отметку. По умолчанию выбран: «Хотя бы один визуализатор с пресета в расчете». Содержит следующие элементы: • Хотя бы один визуализатор с пресета в расчете. Всегда при открытии пресета
Способ расчета	Выпадающий список	Поле поддерживает единичную отметку. Способ расчета можно выбрать только для метода «Хотя бы один визуализатор с пресета в расчете". По умолчанию выбран: «Рассчитывать блоки после события, относящиеся к открываемому пресету". Содержит следующие элементы: • Рассчитывать блоки после события, относящиеся к открываемому пресету. Рассчитывать все блоки после события

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Наименование	Тип данных	Примечание
Данные	JSON	Дефолтные данные с которыми будет рассчитываться задача. По умолчанию {}

6.1.4.16.3 Группа «Общие настройки блока»

Аналогично приведенным в п. 6.1.1.1.3.

6.1.4.16.4 Входы

В блоке нет входов.

6.1.4.16.5 Выходы

Наименование	Тип данных	Примечание
Событие открытия пресета. Данные	JSON	Выводит данные, которые были переданы в параметры при открытии пресета. Если переданных данных нет, то передаются данные, которые находятся в настройках блока в поле «Данные"

6.1.4.16.6 События блока

Название выхода	Тип	Описание
OnPresetOpen	JSON	Событие срабатывает при открытии пресета

6.1.4.16.7 Описание работы блока

Проверяются условия для его выполнения. Если хотя бы одно из условий выполнены, то запускает событие блока «OnPresetOpen».

Проверяемые условия для запуска блока на расчет:

1. Есть ли визуализаторы в открываемом пресете в ветке расчета (в списке блоков, которые будут рассчитываться после блока «События пресета»).
2. Если метод расчета выбран «Всегда при открытии пресета».

Если выбран метод «Хотя бы один визуализатор с пресета в расчете», то можно выбрать способ расчета:

1. Рассчитывать блоки после события, относящиеся к открываемому пресету.
2. Рассчитывать все блоки после события.

Если выбран первый вариант, то будут рассчитываться только те блоки в событии, которые нужны для расчета визуализаторов на пресете.

Если выбран второй вариант, то будут рассчитываться все блоки в событии, если есть ходя бы один блок из пресета в событии.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.1.4.17 Блок «Статистические характеристики»

6.1.4.17.1 Назначение

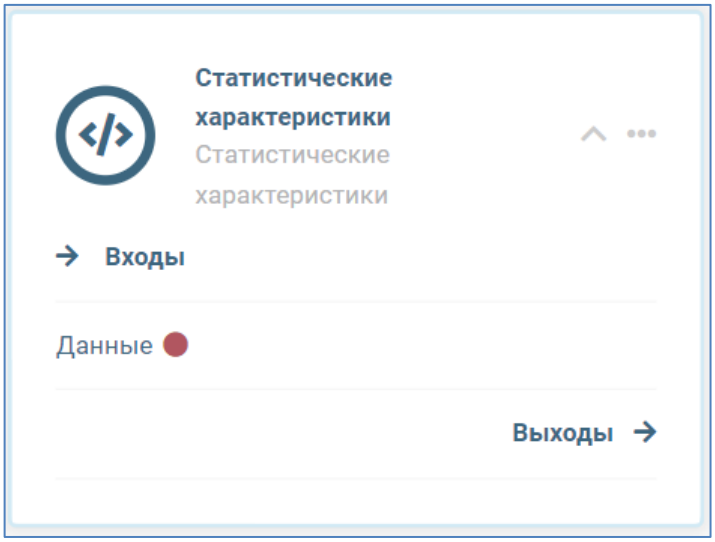


Рисунок 263 – Пример блока «Статистические характеристики»

Блок предназначен для визуализации общих статистических характеристик моделей.

6.1.4.17.2 Настройки блока

Наименование	Тип данных	Примечание
Основные свойства		
Количество знаков в дробной части	Целое число	Число знаков после запятой, до которого будет округлены десятичные дроби.

6.1.4.17.3 Группа «Общие настройки блока»

Аналогично приведенным в п. 6.1.1.1.3.

6.1.4.17.4 Входы блока

Название входа	Тип	Описание

Данные	Структура (json)	Обязательная переменная. Структура содержит json формата:
		Пример на основе стат. характеристик из блока «Арима»
		{
		"model": {
		"model_name": "Линейная регрессия",
		"method": "OLS",
		"row_name": "y",
		"count_point": 25,
		"model": "Y = 1.994313*X1"
		},
		"param": [
		{
		"name": "x1",
		"param": 1.9943130486061926,
		"bse": 0.3103413185409698,
		"tv": 6.426192483753699,
		"pv": 0.00000120627983569209 37,

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название входа	Тип	Описание

		"alias": "X1"
		}
		],
		"stat": {
		"R-квадрат": -0.8476741462936912,
		"Среднее ЗП": 22484.08,
		"Нормированный R-квадрат": -0.8476741462936912,
		"Стд. Отклон ЗП": 11435.469415084513,
		"Стандартная ошибка регрессии": 15544.13958948188,
		"Асимметрия": 0.41722776828814523,
		"Сумма квадратов отклонений": 5798886613.85515,
		"Эксцесс": 0.6785635679086428,
		"Log правдоподобия": -276.24916268105284,
		"Статистика Жака-Бера (JB)": 1.2049630814747212,
		"Инф. критерий Акаика (AIC)": 554.4983253621057,
		"Вероятность (JB)": null,
		"Инф. критерий Байеса (BIC)": 562.1549528367103,
		"F статистика": null,
		"Инф. критерий Ханна-Куини (HQ)": 554.8363897138798,
		"Вероятность (F)": null,
		"Статистика Дарбина-Уотсона (DW)": 1.2865268577470172
		}
		}

6.1.4.17.5 Выходы блока

Блок не содержит выходов.

6.1.4.17.6 События блока

В блоке нет событий.

6.1.4.17.7 Теоретическое описание

6.1.4.17.7.1 Общие характеристики моделей:

Среднее зависимой переменной. Если T – количество точек, на котором была рассчитана модель, то статистика рассчитывается по формуле:

$$\overline{y} = \frac{\sum_{i=1}^{T} y_i}{T}$$

Стандартное отклонение зависимой переменной. Статистика рассчитывается по формуле:

$$\sigma_y = \sqrt{\frac{\sum_{i=1}^{T}(y_i - \overline{y})^2}{T-1}}$$

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Сумма квадратов отклонений: Если r_i – остатки модели, то статистика рассчитывается по формуле:

$$SSR = \sum_{i=1}^{T} r_i$$

Стандартная ошибка регрессии: Если k – количество переменных в модели, то статистика рассчитывается по формуле:

$$S = \sqrt{\frac{SSR}{T-k}}$$

R-квадрат. Статистика рассчитывается по формуле:

$$R^2 = 1 - \frac{SSR}{\sum_{i=1}^{T}(y_i - \overline{y})^2}$$

Нормированный R-квадрат. Статистика рассчитывается по формуле:

$$adjR^2 = 1 - (1 - R^2)\left(\frac{k-1}{T-k}\right)$$

Статистика Дарбина – Уотсона. Статистика рассчитывается по формуле:

$$DW = \frac{\sum_{i=1}^{T}(r_t - r_{t-1})^2}{SSR}$$

F статистика. Статистика имеет распределение Фишера со (k-1), (T-k) степенями свободы и рассчитывается по формуле:

$$F = \frac{R^2/(k-1)}{((1-R^2)/(T-k))}$$

Log правдоподобия. Статистика рассчитывается по формуле:

$$logl = -\frac{T}{2}\left(1 + \ln 2\pi + \ln\left(\frac{SSR}{T}\right)\right)$$

Информационный критерий Акаика. Статистика рассчитывается по формуле:

$$AIC = 2k - 2logl$$

Информационный критерий Байеса. Статистика рассчитывается по формуле:

$$BIC = 3\ln T - 2logl$$

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Информационный критерий Ханна-Куини. Статистика рассчитывается по формуле:

$$HQ = 2k\ln(\ln T) - 2logl$$

Асимметрия или момент 3 порядка. Статистика рассчитывается по формуле:

$$A = \frac{m_3}{\sigma_r^3}$$

Эксцесс или момент 4 порядка. Статистика рассчитывается по формуле:

$$E = \frac{m_4}{\sigma_r^4} - 3$$

Статистика Жака – Бера. Статистика имеет распределение Хи квадрат с (k -1) степенью свободы и рассчитывается по формуле:

$$JB = \frac{T}{6}\left(A^2 + \frac{E^2}{4}\right)$$

6.1.4.17.7.2 Частные характеристики моделей:

R-квадрат МакФадена. Если logl_0 – нулевой log правдоподобия, то статистика рассчитывается по формуле:

$$R_{mf} = 1 - \frac{logl}{logl_0}$$

Статистика LR. Статистика имеет распределение Хи квадрат с (k -1) степенью свободы и рассчитывается по формуле:

$$LR = -2(logl_0 - logl)$$

Средний log правдоподобия. Статистика рассчитывается по формуле:

$$logl_{maen} = \frac{logl}{T}$$

6.1.4.17.8 Ограничения метода

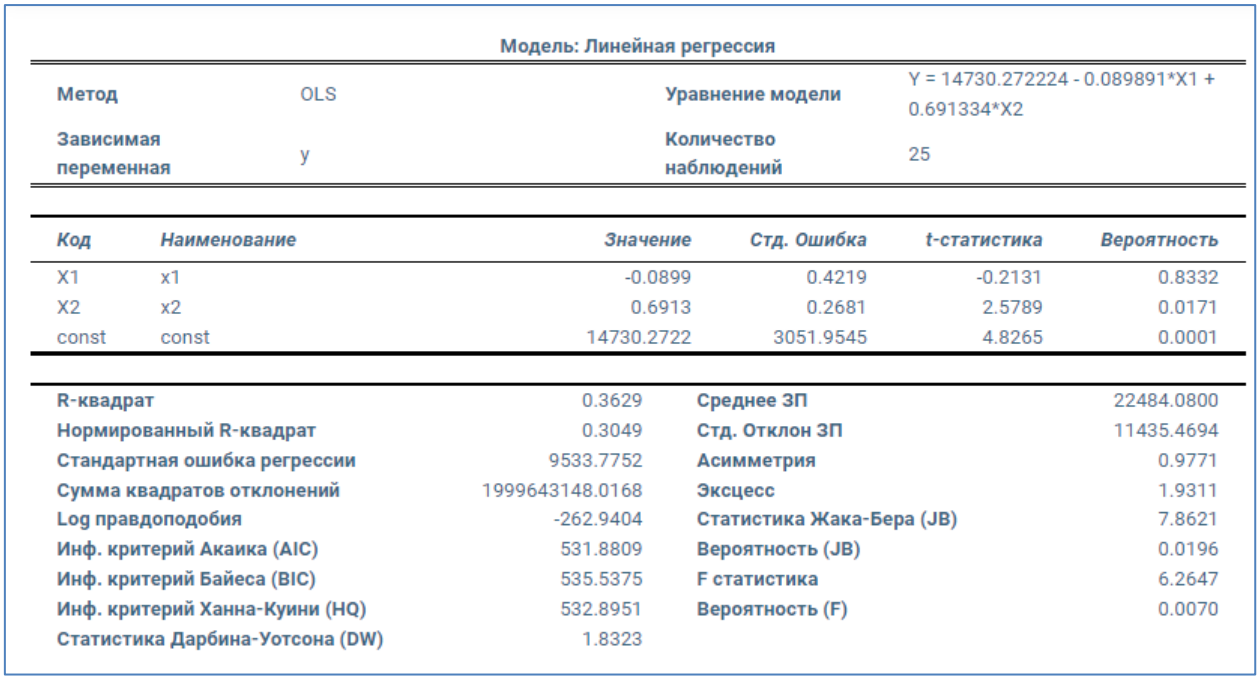
- Блок может визуализировать расчета характеристик только одной модели.
- Каждая модели имеет свой набор характеристик.

6.1.4.17.9 Описание работы блока

Подставляет входные данные блока в jinja шаблон и возвращает результат в виде html разметки.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.1.4.17.10 Описание работы визуализатора



Модель: Линейная регрессия

Метод	OLS	**Уравнение модели**	Y = 14730.272224 - 0.089891*X1 + 0.691334*X2
Зависимая переменная	y	**Количество наблюдений**	25

Код	*Наименование*	*Значение*	*Стд. Ошибка*	*t-статистика*	*Вероятность*
X1	x1	-0.0899	0.4219	-0.2131	0.8332
X2	x2	0.6913	0.2681	2.5789	0.0171
const	const	14730.2722	3051.9545	4.8265	0.0001

R-квадрат	0.3629	**Среднее ЗП**	22484.0800
Нормированный R-квадрат	0.3049	**Стд. Отклон ЗП**	11435.4694
Стандартная ошибка регрессии	9533.7752	**Асимметрия**	0.9771
Сумма квадратов отклонений	1999643148.0168	**Эксцесс**	1.9311
Log правдоподобия	-262.9404	**Статистика Жака-Бера (JB)**	7.8621
Инф. критерий Акаика (AIC)	531.8809	**Вероятность (JB)**	0.0196
Инф. критерий Байеса (BIC)	535.5375	**F статистика**	6.2647
Инф. критерий Ханна-Куини (HQ)	532.8951	**Вероятность (F)**	0.0070
Статистика Дарбина-Уотсона (DW)	1.8323		

Рисунок 264 – Внешний вид визуализатора

Предназначен для визуализации общих статистических характеристик моделей.

6.1.4.17.11 Настройки визуализатора

Визуализатор не имеет специфических настроек. Общий список приведен в п. 4.2.1.3 «Общие настройки в режиме редактирования пресета».

6.1.4.17.12 Дополнительные ссылки

Коэффициент детерминации R2

Критерий Дарбина — Уотсона

Информационные критерии

Критерий Фишера

Функция правдоподобия

Тест Харке — Бера

Тест отношения правдоподобия

6.1.4.18 Блок «Фильтр»

6.1.4.18.1 Назначение

Блок позволяет формировать фильтр, способный фильтровать другие блоки в приложении

6.1.4.18.2 Настройки блока

Название параметра	Тип параметра	Обязательный	Описание	Пример
Название блока	Строка	Нет	Наименование блока	
Заголовок визуализатора	Строка	Нет	Наименование блока, которое отобразится в приложении	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
Ориентация	Выпадающий список	Да	Расположение блока в приложении	
Фильтры	Строка	Нет		
Управляющий блок	Флажок	Нет	При включении элемента – анимация сможет управлять другими блоками	
Наименование	Строка	Нет	Ключ блока – его следует указывать в поле «наименование» управляемых блоков.	
Ссылка на документацию	Строка	Нет		
Описание блока	Строка	Нет		
Пропуск расчета блока	Флажок	Нет	Рассчитывать или нет данный блок	

6.1.4.18.3 Входы

- **Справочники**

Массив объектов, на основании которого будут строиться справочники фильтра.

Для корректной работы фильтра необходимо, чтобы в атрибуте каждого объекта присутствовали следующие поля:

Поля для фильтра

```
{
 "dict_id": <Идентификатор справочника>
 "dict_name": <Наименование справочника>
 "pivot": <Не обязательное поле>
 "id": <Идентификатор элемента>
 "name": <Наименование элемента>
}
```

6.1.4.18.4 Выходы

Блок не имеет конкретных выходов.

6.1.4.19 Блок «Форматированный текст»

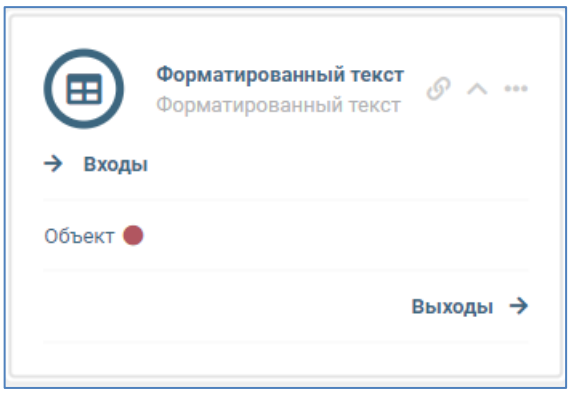


Рисунок 265 – Пример Блока «Форматированный текст»

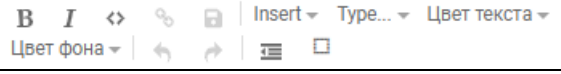
© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.1.4.19.1 Назначение

Блок позволяет формировать визуализатор для ввода, редактирования и сохранения произвольного текста.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.1.4.19.2 Настройки блока

Название параметра	Тип параметра	Обязательный	Описание	Пример
Параметры привязки	Строка	Нет	Идентификатор атрибута справочника, в который будет происходить сохранение текста	
Отображать панель редактирования	Флаг	Нет	С включенным флагом отображается панель инструментов	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.1.4.19.3 Группа «Общие настройки блока»

Аналогично приведенным в п. 6.1.1.1.3.

6.1.4.19.4 Входы

Название входа	Тип	Обязательный	Описание
Объект	Объект	Да	Данные должны содержать атрибут справочника, в который будет происходить сохранение текста. Этот атрибут указывается в настройках «Параметры привязки" Данные, которые поступают на вход, должны быть из блока с поддержкой сохранения атрибутов. Такими блоками могут быть блок «Справочник» (см. п. 6.2.16) или блок «SQL – запрос-справочник» (см. п. 6.2.11) (в случае блока «SQL – запрос-справочник», в нем должны быть настроены атрибуты для сохранения и указана таблица для сохранения). На вход может прийти один объект или массив, однако в случае массива сохранение будет происходить в первый объект

6.1.4.19.5 Выходы

В блоке нет выходов.

6.1.4.19.6 События блока

В блоке нет событий.

6.1.4.19.7 Описание работы блока

Блок формирует визуализатор, текст для него содержится в виде строки с разметкой и стилями.

Блок «Форматированный текст» не сохраняет текст сам, для этого нужен блок, который является источником данных. Таким блоком может быть, например, блок «Справочник» (см. п. 6.2.16) или блок «SQL – запрос-справочник» (см. п. 6.2.11) (в случае блока SQL – запрос-справочник, в нем должны быть настроены атрибуты для сохранения и указана таблица для сохранения).

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.1.4.19.8 Описание работы визуализатора

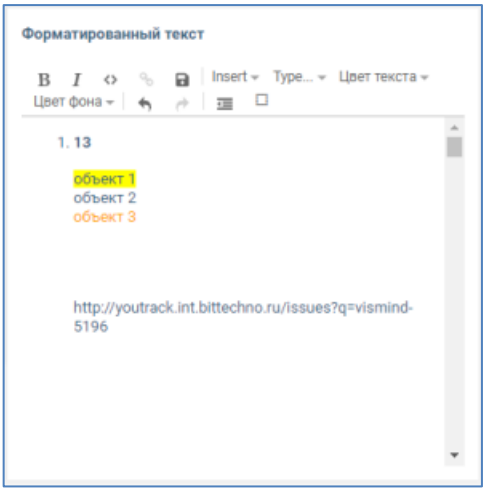


Рисунок 266 – Внешний вид визуализатора «Форматированный текст"

1. Чтобы ввести текст необходимо поставить курсор внутри визуализатора.
2. Чтобы сохранить изменения необходимо нажать кнопку «Сохранить» в панели инструментов.

6.1.4.19.9 Настройки визуализатора

Визуализатор не имеет специфических настроек. Общий список приведен п. 4.2.8 «Настройки визуализатора».

6.1.4.20 Блок «Лог вычисления правил»

6.1.4.20.1 Назначение

Блок устарел и будет удален в следующих версиях платформы.

Блок позволяет вывести лог расчета продукционных правил из блока «Вычисление правил».

На текущий момент этот блок можно соединить через вход только с выходом блока «Вычисление правил».

6.1.4.20.2 Настройки блока

Название параметра	Тип параметра	Обязательный	Описание	Пример
Название блока	Строка	Нет	Наименование блока	
Заголовок блока	Строка	Нет	Заголовок, который будет у визуализатора в пользовательском приложении	
Ссылка на документацию	Строка	Нет	Ссылка на эту страницу	
Описание блока	Строка	Нет	base64 строка для изображения, которое может содержать	
Пропуск расчета блока	Флаг (true/false)	Да	Пропуск расчета блока при расчете графа (по умолчанию выключен)	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.1.4.20.3 Входы

- **Список рекомендаций**

На вход подаются рекомендации из блока «Вычисление правил». Формат данных – json.

Т.к. блок является прямым продолжением блока «Вычисление правил», то подавать на вход другие данные, кроме как из блока «Вычисление правил» нельзя.

6.1.4.20.4 Выходы

Блок не имеет конкретных выходов.

6.1.5 Шаблонизатор Jinja

Шаблонизатор Jinja является довольно распространённым способом генерации статических HTML-страниц, например, с его помощью работает Django.

Это текстовый шаблонизатор, поэтому он может быть использован для создания любого вида разметки, а также исходного кода.

В приложении шаблонизатор используется, к примеру, в блоке «HTML-шаблон», а также для формирования динамических заголовков визуализаторов.

6.1.5.1 Особенности работы шаблонизатора

- Шаблонизатор всегда работает с «чистыми» (не измененными) данными, которые пришли на вход блока.
- Для блоков с единичным входом (к примеру Блок «HTML-шаблон») идентификатор корневого элемента называется **input**.
- Для блоков с динамическими входами (к примеру Блок «Панель OLAP») идентификатор корневого элемента называется так же, как и идентификатор входа, из которого необходимо получить данные.
- Для блоков, у которых есть и статические и динамические входы (Блок «Визуализатор JavaScript») идентификаторами являются, как и **input** (для корневого элемента), так и идентификаторы добавленных входов.
- Входы, которые могут перезаписывать настройки блока (к примеру вход «Привязки» блока «Таблица») недоступны для работы с шаблонизатором.

6.1.5.1.1 Дополнительные функции

Кроме стандартных функций шаблонизатора, в приложение добавлены некоторые дополнительные функции:

- **match(<pattern>, <string>)** – возвращает первое совпадение шаблона в строке. Аналог re.match из Python.
- **search(<pattern>, <string>)** – возвращает совпадение шаблона в любом месте строки. Аналог re.search из Python.
- **fullmatch(<pattern>, <string>)** – возвращает объект, если строка полностью совпадает с шаблоном. Аналог re.fullmatch из Python.
- **pretty(<row>)** – преобразует строку в json с форматированием и отступами.
- **linearize(<row>)** – преобразует строку в json (без форматирования).
- **to_list(<context>)** – преобразует данные в массив элементов, если это возможно. В противном случае – выдаст ошибку.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

- **to_dict(<context>) –** преобразует данные в словарь, если это возможно. В противном случае – выдаст ошибку.
- **to_options(<row>, <to_string>)** – приводит ряд с данными к формату данных, использующихся в визуализаторах (таблицах и диаграммах).
  - **to_string** – флаг форматирования данных в строковый тип.
- **now(<mask>)** - получение текущего времени сервера
  - **mask** - маска даты для отображения. Не обязательное.
- **utcnow(<mask>)** - получение текущего времени сервера в формате UTC
  - **mask** - маска даты для отображения. Аналогично now()
- **timezone()** - получение часового пояса. Отображение приходит в виде строки формата (+/-)HHMM, к примеру +0500 (часовой пояс Екатеринбурга)

6.1.5.2 Дополнительные ссылки

Документация **Jinja**: https://jinja.palletsprojects.com/en/3.0.x/

6.1.6 Контейнеры

6.1.6.1 Блок «Аккордеон»

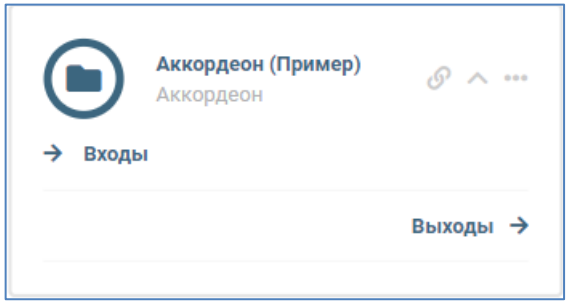


Рисунок 267 – Пример блока «Аккордеон»

6.1.6.1.1 Назначение

Блок используется для размещения визуализаторов, а также их группировки по секциям.

6.1.6.1.2 Настройки блока

Название параметра	Тип параметра	Обязательный	Описание	Пример
Раздел настроек «Секции»				
Название секции	Строка	Нет	Задает наименование секции визуализаторов.	

6.1.6.1.3 Группа «Общие настройки блока»

Аналогично приведенным в п. 6.1.1.1.3.

6.1.6.1.4 Входы

В блоке нет входов.

6.1.6.1.5 Выходы

В блоке нет выходов.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.1.6.1.6 События блока

В блоке нет событий.

6.1.6.1.7 Описание работы блока

Блок «Аккордеон» предназначен для размещения на вкладках различных, сгруппированных по какому-либо принципу, визуализаторов. Это позволяет быстро переключаться между ними при этом экономя пространство в окне приложения. Так же для упорядочивания визуализаторов на вкладках допустимо размещать визуализаторы «Сетка» (см. п. 6.1.6.5), «Вкладки» (см. п. 6.1.6.2) и т.п.

6.1.6.1.8 Описание работы визуализатора

Внешний вид визуализатора «Аккордеон» на пресете.

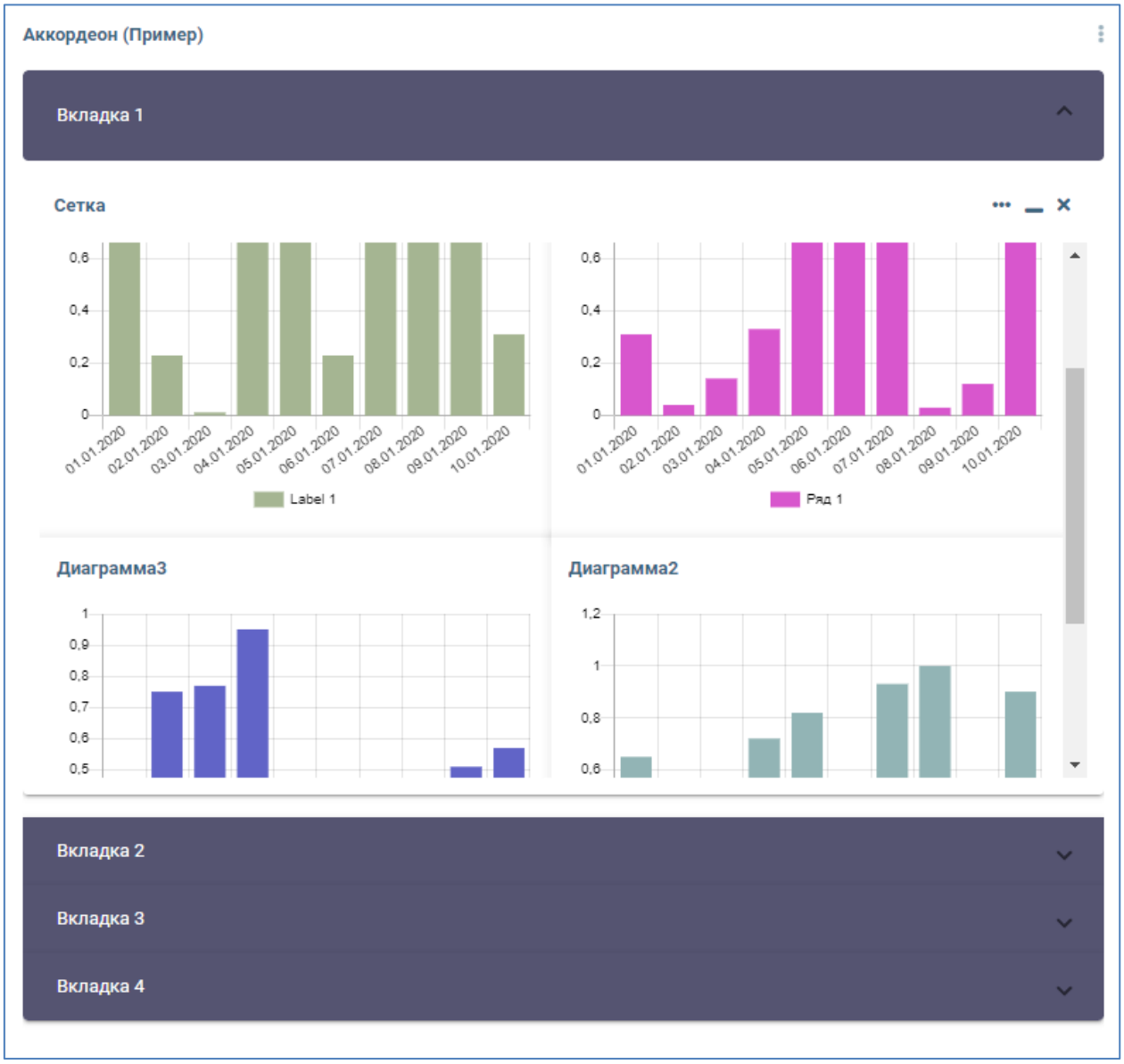


Рисунок 268 – Пример блока «Аккордеон»

Для переключения необходимо нажать ЛКМ по названию вкладки.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Рисунок 269 – Пример блока «Аккордеон»

6.1.6.1.9 Настройки визуализатора

Общий список приведен в п. 4.2.8 «Настройки визуализатора».

Помимо стандартных настроек визуализатор позволяет редактировать следующие настройки:

- закрытые секции;
- открытые секции.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

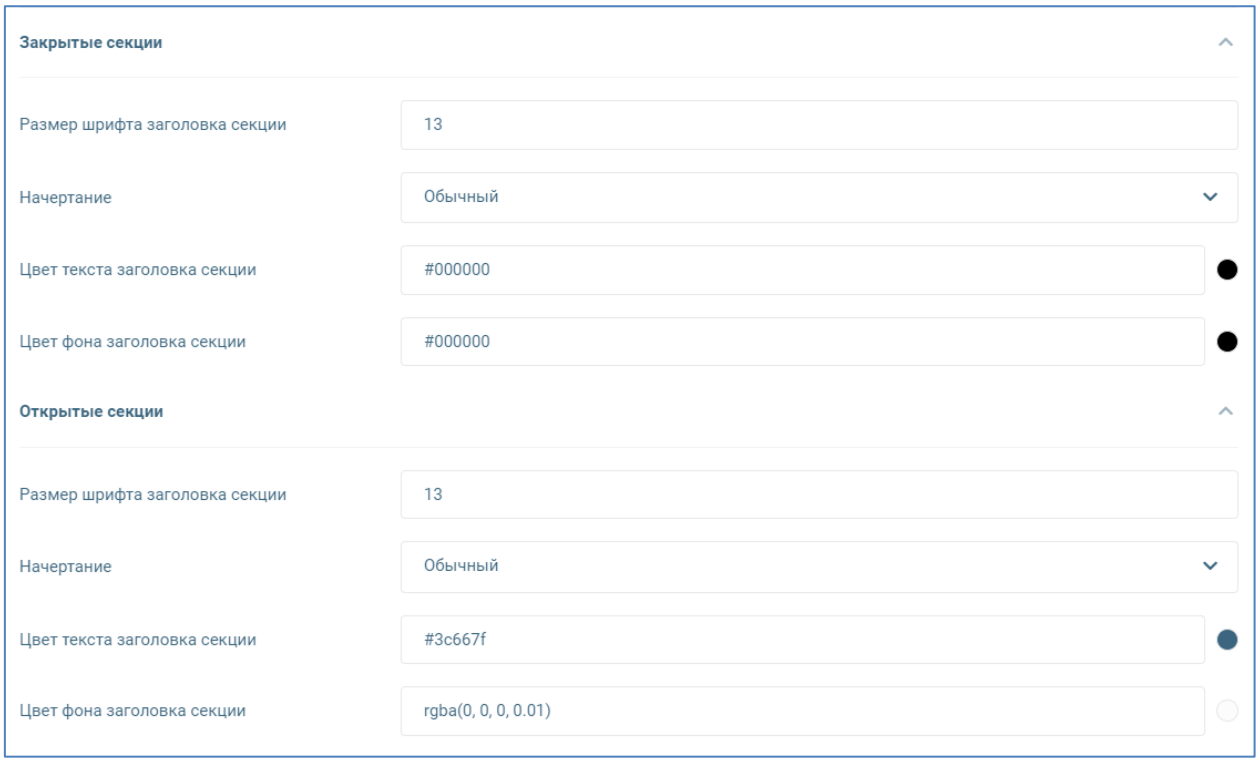


Рисунок 270 – Пример настроек визуализатора

Закрытые и открытые секции содержат одинаковые настройки:

Название параметра	Тип параметра	Обязательный	Описание	Пример
Размер шрифта заголовка секции	Целое число	Да	Задает размер заголовка секции аккордеона	13
Цвет текста заголовка секции	Цвет	Да	Задает цвет текста для заголовка секции аккордеона	rgb(255, 255, 255)
Цвет фона заголовка секции	Цвет	Да	Задает цвет фона для заголовка секции аккордеона	rgb(85, 84, 113)

6.1.6.1.10 Дополнительно

Через Блок «HTML-статика» можно указывать гиперссылки, при нажатии на которые будет осуществлен переход на выбранный визуализатор. Для этого надо указать специальный тег **goto** и идентификатор (GUID) визуализатора. Идентификатор можно скопировать через меню в режиме редактирования. Это имеет смысл для случаев, когда целевой визуализатор расположен на другой вкладке визуализаторе Аккордеон (см. п. 6.1.6.1) и не виден пользователю.

Пример ссылки на визуализатор

```
<ul>
    <a goto="0722f50b-ea85-440c-8337-e9a399e96d21">Диаграмма 1</a>
</ul>
<ul>
    <a goto="02496f8b-baf1-4ec0-8be9-45314dd843fe">Диаграмма 2</a>
</ul>
<ul>
    <a goto="b65e11ba-fa0a-450e-8e43-a5d1e34183a3">Диаграмма 3</a>
```


© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

```
</ul>
```

6.1.6.2 Блок «Вкладки»

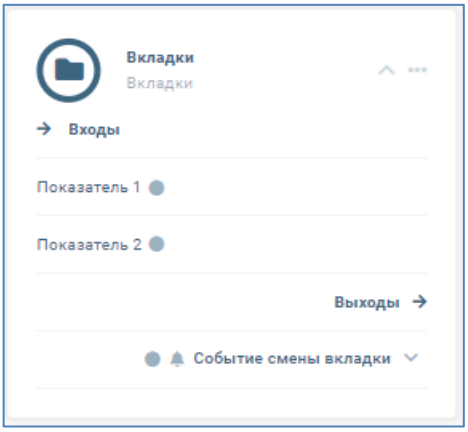


Рисунок 271 – Пример блока «Вкладки»

6.1.6.2.1 Назначение

Блок позволяет формировать вкладки с возможностью разместить визуализаторы на них.

6.1.6.2.2 Настройки блока

Название параметра	Тип параметра	Обязательное	Описание	Пример
Группа «Основные свойства»				
Ориентация	Выпадающий список	Да	Ориентация блока в приложении. Устарело, отключено с 3 версии блока	
Параметры отображения	Раздел настроек	Нет	Задает названия вкладок	
Название вкладки	Строка	Нет	Поле в котором задаётся имя вкладки непосредственно. По умолчанию выводится название "Вкладка N", где N - порядковый номер вкладки.	

6.1.6.2.3 Группа «Общие настройки блока»

Аналогично приведенным в п. 6.1.1.1.3.

6.1.6.2.4 Входы

Количество входов регулируется количеством вкладок заданных в настройках блока.

Названия входов зависят от названий вкладок, установленных в настройке "Параметры отображения" → "Название вкладки".

Название входа	Описание	Обязательное
Вкладка 1	Ряд данных	Нет
Вкладка 2	Ряд данных	Нет

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.1.6.2.5 Выходы

В блоке нет выходов.

6.1.6.2.6 События блока

Название выхода	Тип	Описание
Событие смены вкладки	Объект	Событие происходит при переключении между вкладками. На выходы передаются те значения, которые были на входах. Количество дочерних выходов события регулируется количеством вкладок, заданных в настройках блока. Названия событийных выходов аналогичны названиям входов и зависят от названий вкладок, установленных в настройке "Параметры отображения" → "Название вкладки".

6.1.6.2.7 Описание работы блока

Блок «Вкладки» предназначен для размещения на вкладках различных, сгруппированных по какому-либо принципу, визуализаторов.

Это позволяет быстро переключаться между ними при этом экономя пространство в окне приложения.

Так же для упорядочивания визуализаторов на вкладках допустимо размещать визуализаторы «Сетка» (см. п. 6.1.6.5), «Аккордеон» (см. п. 6.1.6.1) и т. п.

Внешний вид визуализатора приведен ниже на рисунке.

Рисунок 272 – Пример визуализатора «Вкладки»

Для переключения необходимо нажать по названию вкладок на визуализаторе. Если вкладки не помещаются по ширине – будет доступна прокрутка по горизонтали.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.1.6.2.8 Особенности работы

При переключении между вкладками визуализаторы, расположенные на других вкладках, невидимых пользователю, не обновляются. Вместо этого, обновление происходит в тот момент, когда пользователь начинает их видеть, причем **только в случае, если они должны быть обновлены.**

6.1.6.2.9 Настройки визуализатора

Визуализатор не имеет специфических настроек. Общий список приведен в п. 4.2.8 «Настройки визуализатора».

6.1.6.3 Блок «Выдвигающиеся панели»

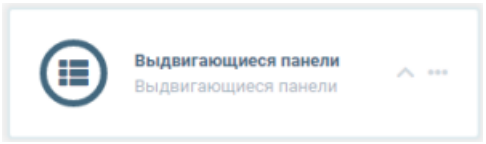


Рисунок 273 – Пример блока «Выдвигающиеся панели»

6.1.6.3.1 Назначение

Блок используется для размещения визуализаторов в специальных панелях, выдвигающихся с разных сторон экрана.

6.1.6.3.2 Настройки блока

Название параметра	Тип параметра	Обязательный	Описание	Пример
Левая панель	Флаг	Нет	Включает отображение левой выдвигающейся панели	
Правая панель	Флаг	Нет	Включает отображение правой выдвигающейся панели	
Верхняя панель	Флаг	Нет	Включает отображение верхней выдвигающейся панели	
Нижняя панель	Флаг	Нет	Включает отображение нижней выдвигающейся панели	

6.1.6.3.3 Группа «Общие настройки блока»

Аналогично приведенным в п. 6.1.1.1.3.

6.1.6.3.4 Входы

В блоке нет входов.

6.1.6.3.5 Выходы

В блоке нет выходов.

6.1.6.3.6 События блока

В блоке нет событий.

6.1.6.3.7 Описание работы блока

Блок предназначен для размещения визуализаторов на специальных выдвигающихся панелях.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

На одном пресете визуализатор «Выдвигающиеся панели» может быть размещен только один раз.

6.1.6.3.8 Описание работы визуализатора

После размещения блока, в пресете появляются стрелки, указывающие на наличие панелей (стрелки будут отображаться с тех сторон, для которых была включена выдвигающаяся панель): .

6.1.6.3.9 Настройки визуализатора

Общий список приведен в п. 4.2.8 «Настройки визуализатора».

6.1.6.4 Блок «Выпадающее меню»

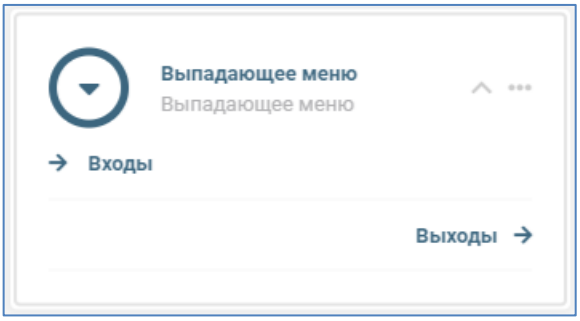


Рисунок 274 – Пример блока «Выпадающее меню»

6.1.6.4.1 Назначение

Блок позволяет формировать визуализатор, представляющий выпадающее меню. Внутри визуализатора пользователь может размещать другие визуализаторы.

6.1.6.4.2 Настройки блока

Название параметра	Тип параметра	Обязательное	Описание	Пример
Вид отображения	Выпадающий список	Нет	Вид отображения меню: Меню (под визуализатором «Выпадающее меню» открывается окно, его ширина как у выпадающего меню, высоту можно настроить произвольно и сохранить в пресет). Всплывающее окно (содержимое визуализатора «Выпадающее меню» открывается в окне на весь экран). Размещать визуализаторы внутри визуализатора. «Выпадающее меню» можно, если выбран вид отображения «Меню»	

6.1.6.4.3 Группа «Общие настройки блока»

Аналогично приведенным в п. 6.1.1.1.3.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.1.6.4.4 Входы

В блоке нет входов.

6.1.6.4.5 Выходы

В блоке нет выходов.

6.1.6.4.6 События блока

В блоке нет событий.

6.1.6.4.7 Описание работы блока

Блок «Выпадающее меню» не хранит в себе данные о визуализаторах, которые в нем размещаются. Результатом расчета блока являются настройки отображения выпадающего меню, сохраненные в блоке.

6.1.6.4.8 Описание работы визуализатора

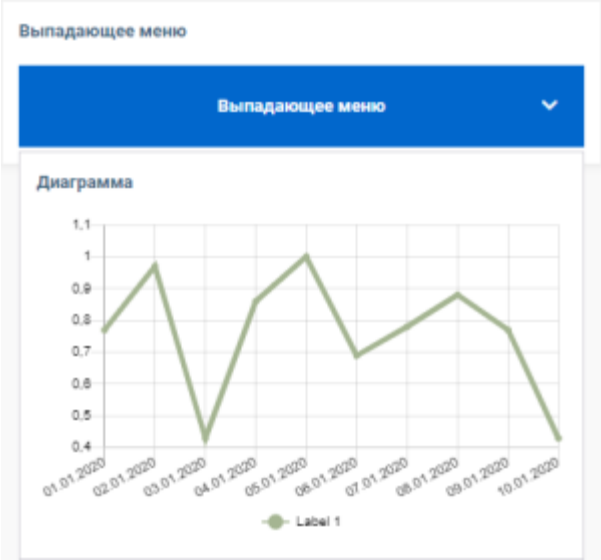


Рисунок 275 – Пример отображения выпадающего меню

Добавление визуализатора в выпадающее меню:

- Чтобы иметь возможность размещать визуализаторы в выпадающем меню, у данного блока в графе должен быть настроен вид отображения «Меню» (см. п. 6.1.4.7).
- Чтобы добавить новый визуализатор в выпадающее меню, необходимо раскрыть меню, затем перетащить визуализатор из библиотеки визуализаторов в открывшееся окно. Визуализатор максимизируется внутри области размещения.
- Чтобы добавить несколько визуализаторов в выпадающее меню, сначала нужно добавить в него сетку, или вкладки, или аккордеон.

Изменение размеров выпадающего меню:

- У области меню можно изменять размеры, для того, чтобы при открытии можно было разместить визуализатор больший, чем кнопка. При этом размеры будут сохраняться вместе с пресетом. При изменении, границы выпадающей области не могут быть менее границ кнопки.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Рисунок 276 – Пример расположения выпадающего меню

6.1.6.4.9 Настройки визуализатора

Специфические настройки:

- Размер текста меню;
- Цвет текста меню;
- Цвет фона меню;
- Цвет фона выпадающего меню.

Помимо стандартных, визуализатор позволяет задать следующие настройки.

Размер текста меню	Целое число	Да	Задает размер текста меню	
Цвет текста меню	Цвет	Да	Задает цвет текста для меню	rgb(255, 255, 255)
Цвет фона меню	Цвет	Да	Задает цвет фона меню	#0167CC
Цвет фона выпадающего меню	Цвет	Да	Задает цвет фона выпадающей части меню	#fff

Рисунок 277 – Пример дополнительных настроек

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Общий список приведен в п. 4.2.8 «Настройки визуализатора».

6.1.6.5 Блок «Сетка»

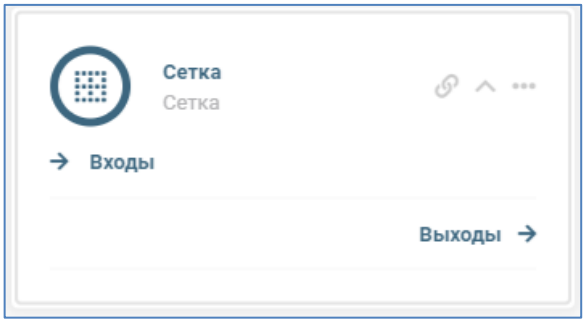


Рисунок 278 – Пример блока «Сетка»

6.1.6.5.1 Назначение

Блок позволяет размещать визуализаторы на пресете в сетке. Визуализаторы, размещенные в сетке, не накладываются друг на друга и автоматически подстраиваются под размеры сетки.

По умолчанию у визуализатора отсутствует какой-либо заголовок (см. ниже в разделе «Общие настройки блоков»).

6.1.6.5.2 Настройки блока

Название параметра	Тип параметра	Обязательный	Описание	Пример
Группа «Основные свойства»				
Величина отступа ячейки, пт	Целое число	Да	Отступ между ячейками сетки в пикселях	
Отображать сетку	Выпадающий список	Нет	Отображение границ ячеек сетки: • Не отображать • При размещении (при перемещении визуализаторов внутри сетки в режиме редактирования сетки) • Всегда	
Способ отображения	Выпадающий список	Нет	Наличие полос прокрутки: • Без полос прокрутки • Вертикальная полоса прокрутки (высота визуализаторов не будет уменьшаться при уменьшении высоты сетки) • Горизонтальная полоса прокрутки (ширина визуализаторов не будет уменьшаться при уменьшении ширины сетки)	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
Адаптивная верстка	Флаг	Нет	С включенным флагом в сетке будет уменьшаться количество столбцов в зависимости от ширины экрана	
Число колонок (min)	Целое число	Нет	Минимальное количество колонок в сетке. По умолчанию – 1, максимальное количество – 1000	
Число колонок (max)	Целое число	Нет	Минимальное количество колонок в сетке. По умолчанию – 100, максимальное количество – 1000	
Число строк (min)	Целое число	Нет	Максимальное количество строк в сетке. По умолчанию – 1, максимальное количество – 1000	
Число строк (max)	Целое число	Нет	Максимальное количество строк в сетке. По умолчанию – 100, максимальное количество – 1000	

6.1.6.5.3 Группа «Общие настройки блока»

Аналогично приведенным в п. 6.1.1.1.3.

6.1.6.5.4 Входы

В блоке нет входов.

6.1.6.5.5 Выходы

В блоке нет выходов.

6.1.6.5.6 События блока

В блоке нет событий.

6.1.6.5.7 Описание работы блока

Блок «Сетка» не хранит в себе данные о визуализаторах, которые в нем размещаются. Результатом расчета блока являются настройки отображения сетки, сохраненные в блоке

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.1.6.5.8 Описание работы визуализатора

Рисунок 279 – Пример визуализатора

Добавление визуализатора в сетку:

- Чтобы добавить новый визуализатор в сетку, необходимо перетащить его из библиотеки визуализаторов на сетку. По умолчанию визуализаторы располагаются в сетке, формируя таблицу n×n: сначала добавляется строка и заполняется визуализаторами, затем столбец и т.д.
- Чтобы добавить в сетку уже размещенный визуализатор, необходимо вырезать его, используя пункт контекстного меню визуализатора «Вырезать» или сочетанием клавиш **Ctrl+X**. Затем вставить его, для этого можно открыть контекстное меню сетки, кликнув ПКМ в пустую область внутри сетки, и выбрать пункт «Вставить» или нажать **Ctrl+V**.

Редактирование сетки:

- Чтобы изменить положение или размер визуализаторов, необходимо включить редактирование сетки в ее контекстном меню
- Чтобы добавить новые строки или столбцы в сетку, необходимо размещенный в сетке визуализатор перетаскивать вниз или вправо соответственно, двигая его к границе сетки

6.1.6.5.9 Настройки визуализатора

Визуализатор не имеет специфических настроек. Общий список приведен в п. 4.2.8 «Настройки визуализатора».

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.1.6.6 Блок «Шторка»

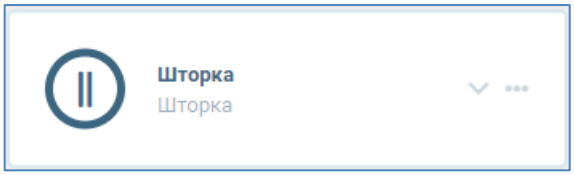


Рисунок 280 – Пример блока «Шторка»

6.1.6.6.1 Назначение

Блок позволяет синхронизировать действия пользователя с двумя визуализаторами «3D-карта» в пресете.

6.1.6.6.2 Настройки блока

В блоке нет специфических настроек.

6.1.6.6.3 Группа «Общие настройки блока»

Аналогично приведенным в п. 6.1.1.1.3.

6.1.6.6.4 Входы

В блоке нет входов.

6.1.6.6.5 Выходы

В блоке нет выходов.

6.1.6.6.6 События блока

В блоке нет событий.

6.1.6.6.7 Описание работы блока

Вся логика работы осуществляется в рамках визуализатора.

6.1.6.6.8 Описание работы визуализатора

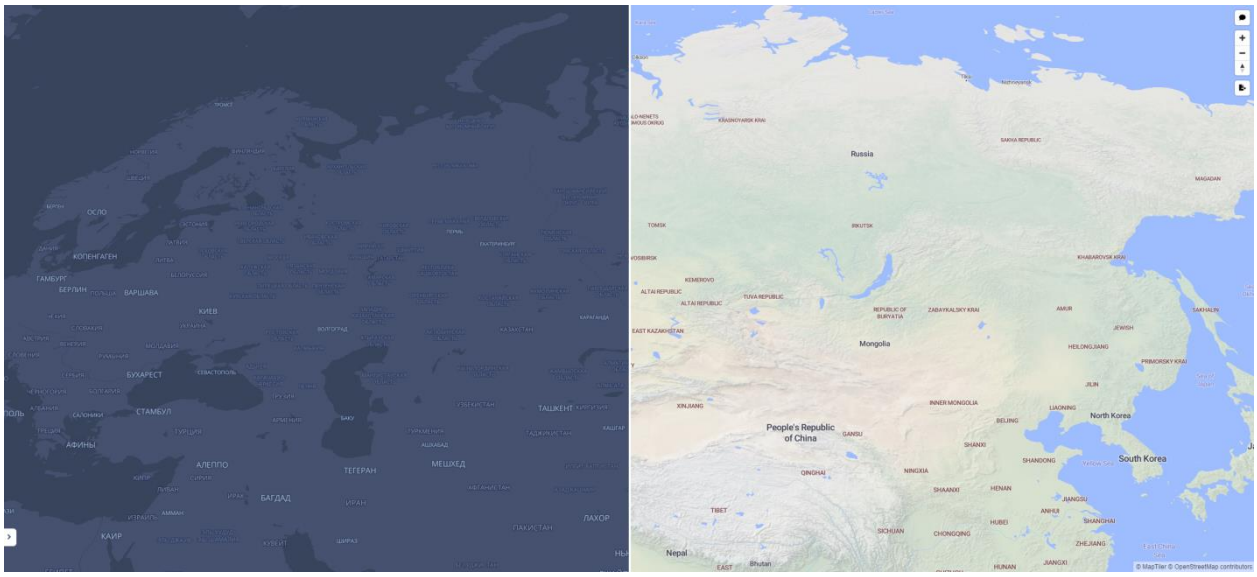


Рисунок 281 – Пример визуализатора

- Для корректной работы блока «Шторка» внутри необходимо разместить 2 блока «3D-карта».
- Карта, которая добавлена первой, отображается слева. Карта, которая добавлена второй, отображается справа

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

- Линию, которая разделяет 2 карты, можно перемещать, зажав ЛКМ.
- Между картами в блоке «Шторка» синхронизируются следующие действия пользователя:
    - изменение масштаба карты;
    - изменение положения карты;
    - изменение угла наклона камеры в карте.
- При открытии пресета начальные настройки масштаба, положения и угла наклона камеры берутся из карты, расположенной слева

6.1.6.6.9 Настройки визуализатора

Общий список настроек доступен в п. 4.2.1.3 «Общие настройки в режиме редактирования пресета».

6.1.7 Изображения

6.1.7.1 Блок «Изображение»

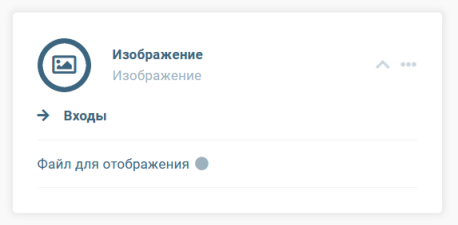


Рисунок 282 – Пример блока «Визуализатор Изображение»

6.1.7.1.1 Назначение

Визуализатор "Изображение" реализует минимальный функционал файлового менеджера и позволяет отображать изображения, которые находятся в директориях задачи.

6.1.7.1.2 Настройки блока

Название параметра	Тип параметра	Обязательный	Описание	Пример
Группа "Специфические настройки блока"				
Изображение	Файловый менеджер	Изображение	Файловый менеджер	

6.1.7.1.3 Группа «Общие настройки блока»

Название параметра	Тип параметра	Обязательный	Описание	Пример
Наименование блока	Строка	Нет	Наименование блока, которое отображается на графе. В панели размещенных блоков возможен по нему поиск	
Заголовок блока	Строка	Нет	Наименование визуализатора, которое будет отображаться в пользовательском приложении, в заголовке блока. Возможно форматирование с	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
			использованием JINJA-шаблонов. Поле доступно только для блоков-визуализаторов	
Ссылка на документацию	Строка	Нет	Поле со ссылкой на документацию. Можно изменить на произвольную пользовательскую ссылку. При клике на иконку у названия блока будет осуществлен на эту ссылку. При замене на пользовательскую ссылку – ссылка на документацию будет перезаписана	
Описание блока	Строка	Нет	Поле для пользовательского описания блока, можно указать произвольный текст. Поле является сворачиваемым и по умолчанию свёрнуто.	
Ссылка по клику на заголовок визуализатора	Строка	Нет	Поле с произвольной пользовательской ссылкой, переход на нее осуществляется при клике ЛКМ на заголовок визуализатора. Может быть как абсолютной, так и относительной Поле доступно только для блоков-визуализаторов	
Тип открытия ссылки	Выпадающий список	Нет	Регулирует поведение для параметра «Ссылка по клику на заголовок визуализатора»: • В новом окне – ссылка открывается в новой вкладке браузера; • В этом же окне – ссылка открывается в текущей вкладке браузера.	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
			Поле доступно только для блоков-визуализаторов	
Пропуск расчета блока	Флаг	Нет	Пропускать или нет блок при расчете. При включенном флаге во время расчета графа блок будет пропущен. При этом его все еще можно посчитать через расчет блока и цепочки. Блоки, расположенные после текущего, будут рассчитаны Для блоков типа «Соединение с БД» поле недоступно	
Кэширование результатов расчета блока	Флаг	Нет	Сохранять или нет выходы блока при расчете графа. Если флаг включен – результаты расчета блока будут использоваться даже в случае если блок выключен из расчета. При выключенном флаге – блоки, идущие за текущим, не получат на вход данных в случае если блок пропускается во время расчета. Для блоков типа «Соединение с БД» поле недоступно	
Пропуск события по умолчанию	Флаг	Нет	Выключает расчет блока с событием по умолчанию, при открытии его визуализатора в пользовательском приложении. Поле доступно только для блоков-визуализаторов у которых есть событие по умолчанию	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.1.7.1.4 Входы

Название входа	Тип	Обязательный	Описание

Файл для отображения	Массив	Нет	Массив данных, хранящий путь к изображению.
			
			Пример входных данных:
			
			{
			"fileNames": [
			"test/test.jpg"
			]
			}

Кроме указанных входов, так же доступно создание динамических входов.

Для создания входа необходимо нажать на знак + в разделе входов блока. Для каждого из входов задается следующий перечень полей:

Название входа	Тип	Обязательный	Описание
Идентификатор	Строка	Да	Идентификатор входа – используется внутри блока, для получения доступа к данным входа. Идентификаторы **data** и **options** зарезервированы для уже существующих входов
Имя	Строка	Да	Имя входа для отображения в списке входов блока
Описание	Строка	Нет	Дополнительная информация с описанием входа
Тип	Выпадающий список	Да	Тип данных, которые будут приходить на вход. Нужен для построения корректных связей с другими блоками. По умолчанию установлен тип "Объект".
Обязательный	Флаг	Да	Обязательность входа. Если вход обязательный и на него не пришло данных – блок не будет рассчитан с ошибкой
Массив	Флаг	Да	Является ли вход массивом данных

6.1.7.1.5 Выходы

В блоке нет постоянных выходов. Доступно создание динамических входов.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Для создания входа необходимо нажать на знак в разделе входов блока. Для каждого из входов задается следующий перечень полей:

Название входа	Тип	Обязательный	Описание
Идентификатор	Строка	Да	Идентификатор входа – используется внутри блока, для получения доступа к данным входа. Идентификаторы **data** и **options** зарезервированы для уже существующих входов
Имя	Строка	Да	Имя входа для отображения в списке входов блока
Описание	Строка	Нет	Дополнительная информация с описанием входа
Тип	Выпадающий список	Да	Тип данных, которые будут приходить на вход. Нужен для построения корректных связей с другими блоками. По умолчанию установлен тип "Объект".
Обязательный	Флаг	Да	Обязательность входа. Если вход обязательный и на него не пришло данных – блок не будет рассчитан с ошибкой
Массив	Флаг	Да	Является ли вход массивом данных

6.1.7.1.6 События блока

В блоке нет событий

6.1.7.1.7 Описание работы блока

Наивысший приоритет, по умолчанию, имеет вход блока, если его нет, то путь к изображению берется из настроек блока.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

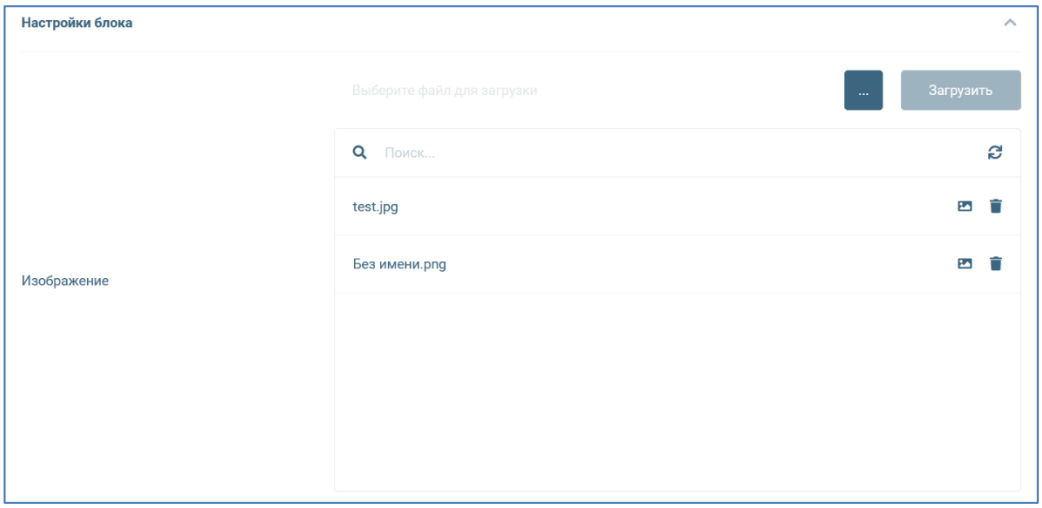


Рисунок 283 – Внешний вид блока настроек блока «Изображение»

Для настройки "Изображение" блока, реализован следующий функционал:

- Просмотр списка изображений
- Просмотр превью изображения
- Загрузка изображения
- Удаление изображения

6.1.7.1.8 Описание работы визуализатора

Визуализатор позволяет отображать на пресете загруженные в задачу изображения.

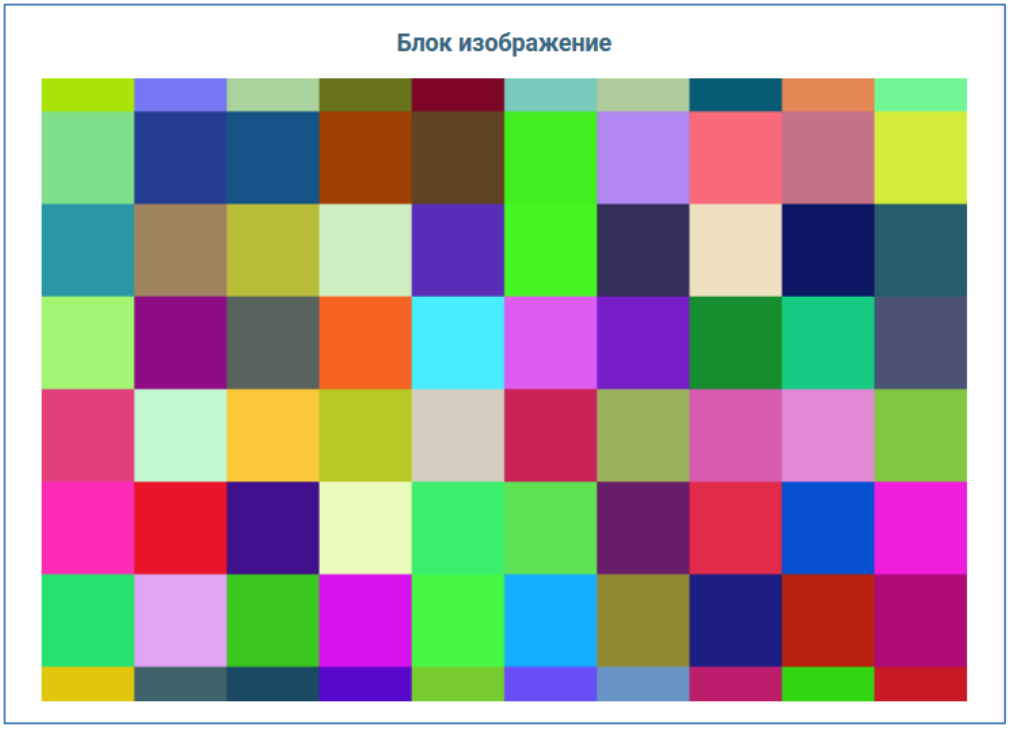


Рисунок 284 – Внешний вид блока «Визуализатор Изображения»

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.1.7.1.9 Настройки визуализатора

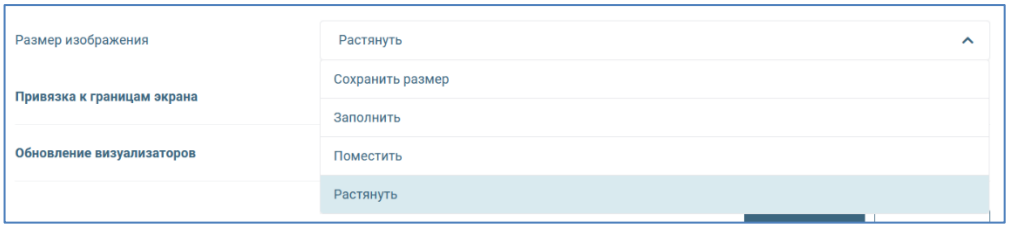


Рисунок 285 – Внешний вид настроек визуализатора

Название параметра	Тип параметра	Обязательный	Описание	Пример
Группа «Изображение»				
Размер изображения	Выпадающий список	Да	Выпадающий список с возможностью выбора визуального поведения блока.	Сохранить размер Заполнить Поместить Растянуть

6.2 Импорт и экспорт данных

6.2.1 Блок «Соединение с PostgreSQL DB»

6.2.1.1 Назначение

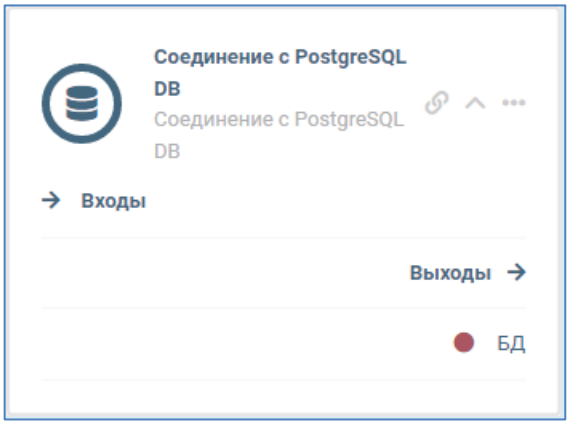


Рисунок 286 – Пример «Соединение с PostgreSQL DB»

Блок позволяет настроить соединение с базой данных PostgreSQL.

6.2.1.2 Настройки блока

Название параметра	Тип параметра	Обязательный	Описание	Пример
Группа «Основные свойства»				
Тип соединения	Выпадающий список	Да	Выбор способа соединения с базой данных. Доступно два варианта:	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
			• **Глобальное соединение** – выбор преднастроенного соединения с базой данных; • **Задать параметры соединения вручную** – ручная настройка соединения	
Выбор соединения с БД	Дерево	Нет	Доступно при выборе типа соединения «Глобальное соединение». Выбор соединения с базой данных из преднастроенного списка	
Адрес сервера	Строка	Нет	Доступно при выборе типа соединения «Задать параметры соединения вручную». Адрес сервера с базой данных	
Порт	Целое число	Нет	Доступно при выборе типа соединения «Задать параметры соединения вручную». Порт сервера с базой данных	
База данных	Строка	Нет	Доступно при выборе типа соединения «Задать параметры соединения вручную» Наименование базы данных	
Пользователь	Строка	Нет	Доступно при выборе типа соединения «Задать параметры соединения вручную». Логин пользователя	
Пароль	Шифрованная строка	Нет	Доступно при выборе типа соединения «Задать параметры соединения вручную». Пароль пользователя	

6.2.1.3 Группа «Общие настройки блока»

Аналогично приведенным в п. 6.1.1.1.3.

6.2.1.4 Входы

В блоке нет входов.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.2.1.5 Выходы

Название выхода	Тип	Описание
БД	Соединение с БД	Уникальный идентификатор соединения с базой данных. При каждом расчете графа генерируется заново.

6.2.1.6 События блока

В блоке нет событий.

6.2.1.7 Описание работы блока

В момент расчета графа блок открывает соединение с базой данных по заданным параметрам и передает уникальный идентификатор открытого соединения.

По этому идентификатору все блоки могут взаимодействовать с указанной в блоке базой данных.

В конце расчета соединение закрывается, и при следующем расчете будет сгенерирован новый идентификатор.

6.2.1.7.1 Проверка соединения

Для проверки корректности соединения с базой данных, после ввода всех настроек можно нажать на кнопку «**Проверка соединения**»:

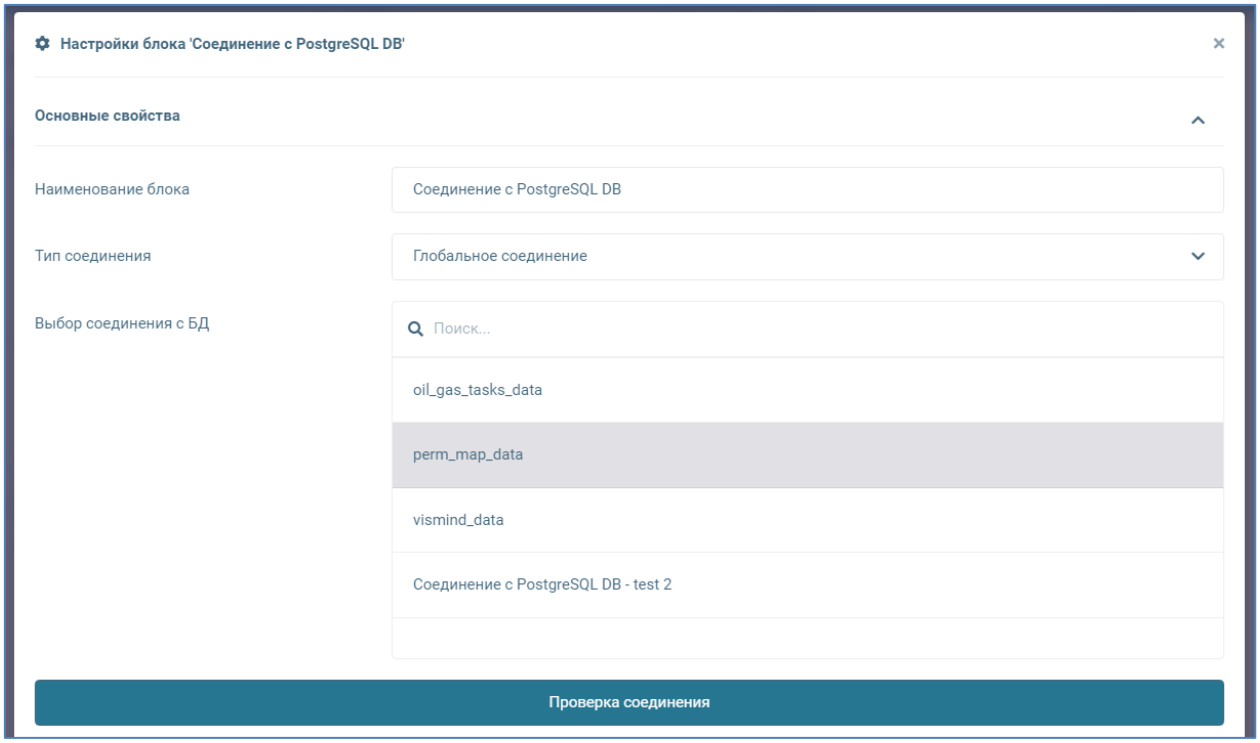


Рисунок 287 – Пример «Соединение с PostgreSQL DB»

После чего будет выдано сообщение об успешном или не успешном соединении с базой данных.

6.2.1.8 Дополнительные ссылки

https://www.postgresql.org/

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.2.1.9 Руководство «Соединение с БД»

6.2.1.9.1 Назначение

Блок «Соединение с БД» – это объект графа, содержащий основные настройки для подключения к существующей базе данных, созданной на сервере СУБД. Рассматривается PostgreSQL.

Подключение к базе данных используется различными объектами графа для получения доступа к данным хранящимся в БД.

6.2.1.9.2 Размещение

Для создания подключения к БД необходимо разместить блок «Соединение с БД» на граф, для этого необходимо:

- найти блок с помощью строки поиска в библиотеке блоков;
- методом Drag&Drop разместить блок на графе.

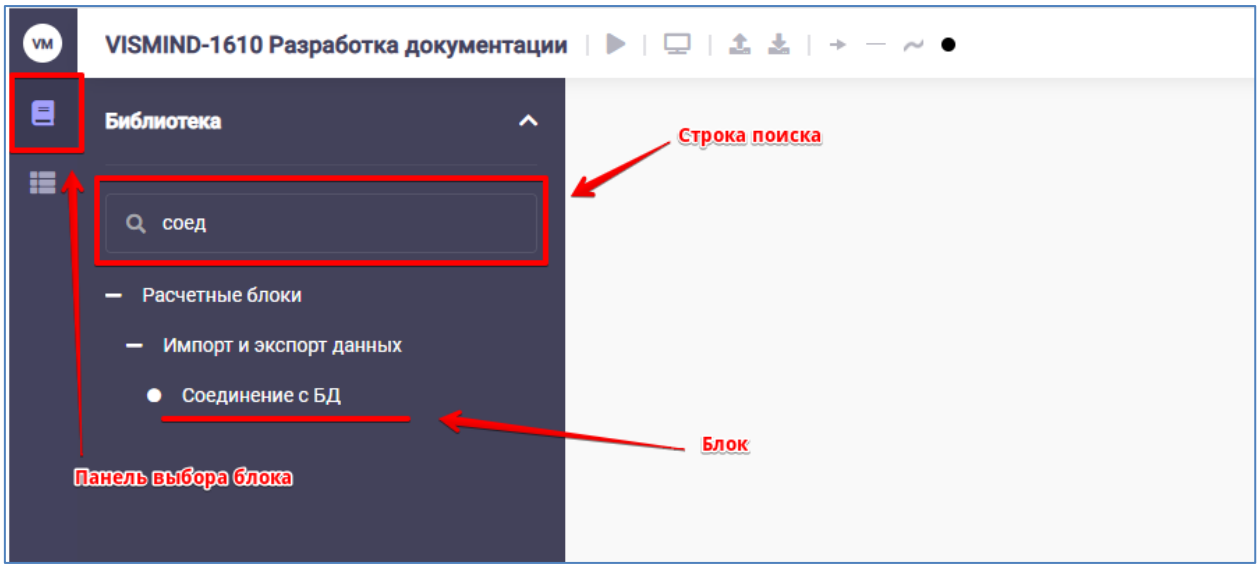


Рисунок 288 – Пример «Соединение с БД»

6.2.1.9.3 Настройка соединения с БД

Для задания настроек необходимо вызвать контекстное меню у размещенного блока, для этого необходимо:

1. Подвести курсор мыши к правому верхнему углу блока и нажать ЛКМ.
2. Выбрать «Настройки».

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

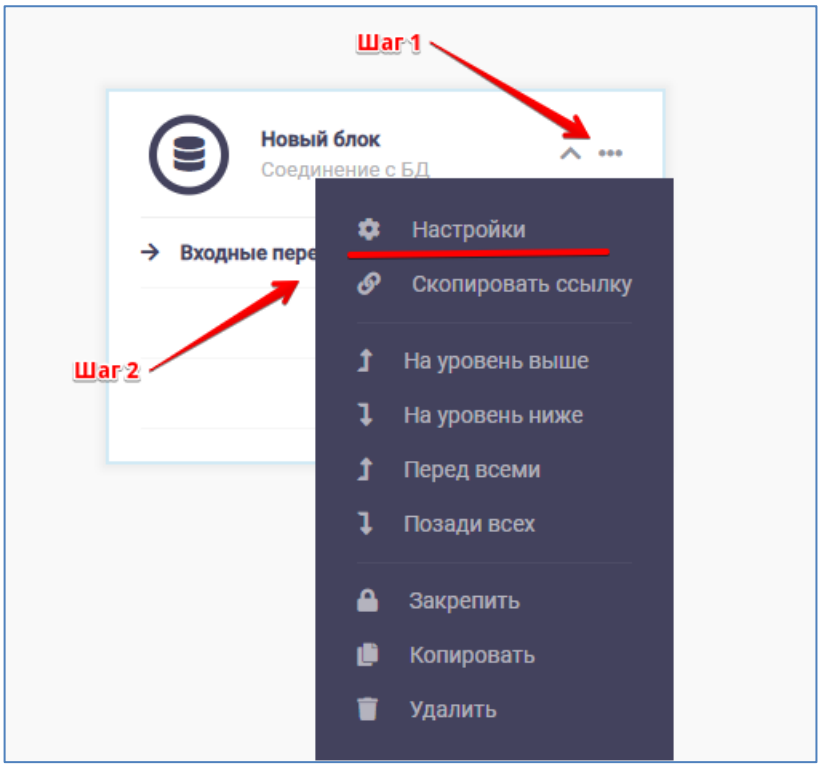


Рисунок 289 – Выбор раздела меню «Настройка»

3. Задать настройки соединения с БД.

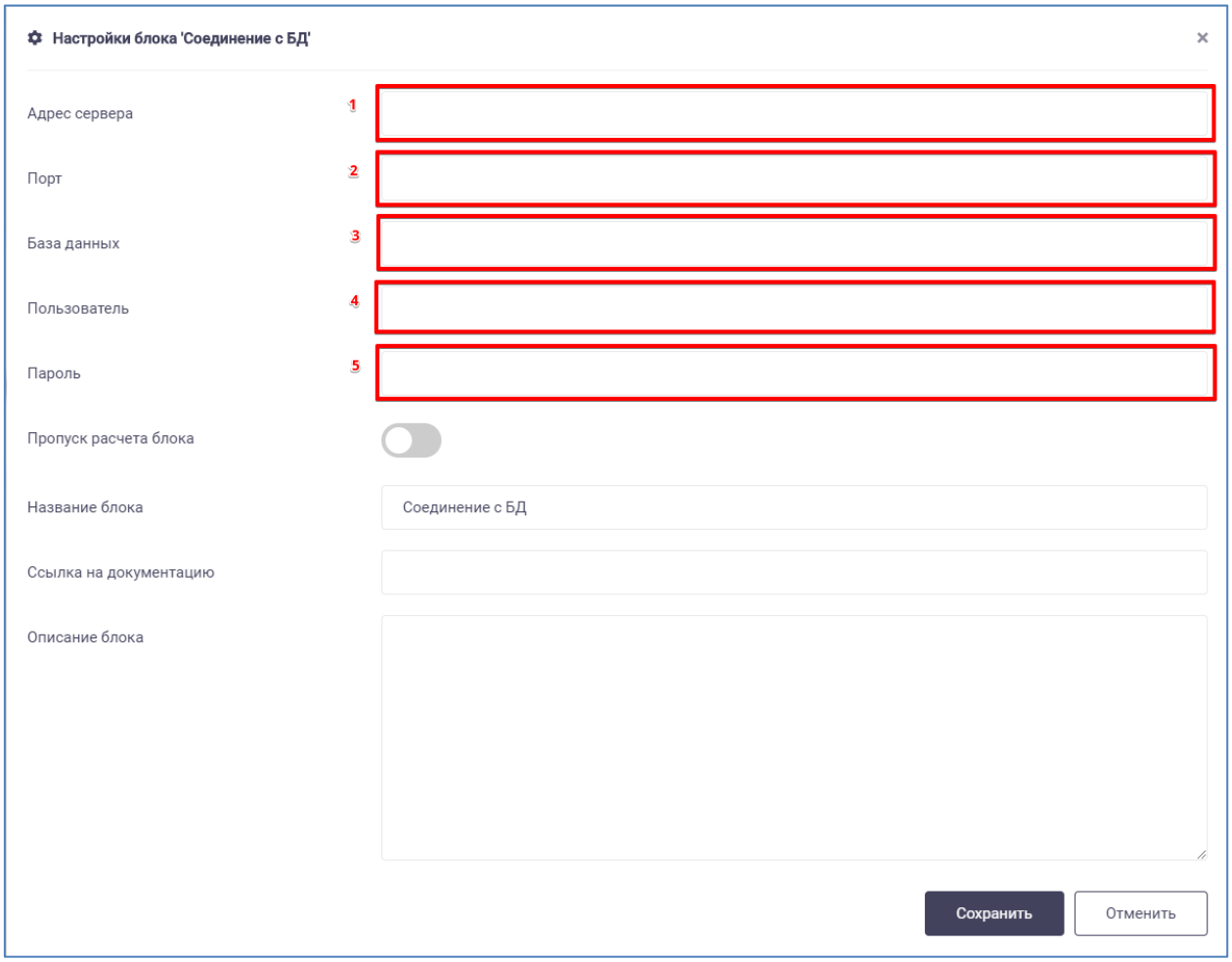


Рисунок 290 – Настройка блока «Соединение с БД»

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.2.2 Блок «Соединение с Microsoft SQL DB»

6.2.2.1 Назначение

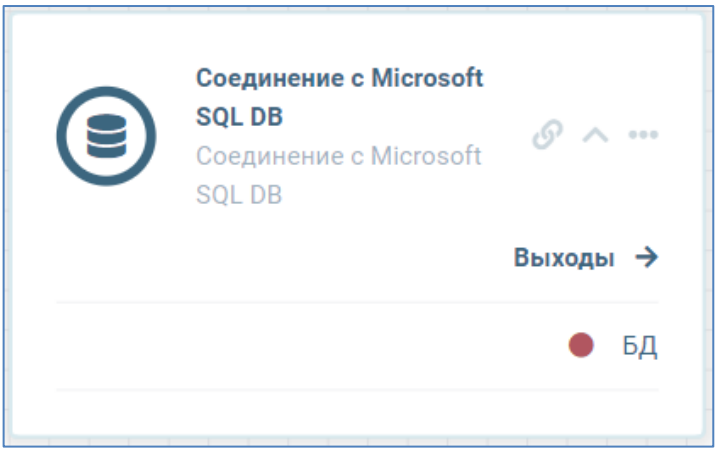


Рисунок 291 – Пример блока «Соединение с Microsoft SQL DB»

Блок позволяет настроить соединение с базой данных Microsoft SQL.

6.2.2.2 Настройки блока

Название параметра	Тип параметра	Обязательный	Описание	Пример
Группа «Основные свойства»				
Тип соединения	Выпадающий список	Да	Выбор способа соединения с базой данных. Доступно два варианта: • **Глобальное соединение** – выбор преднастроенного соединения с базой данных; • **Задать параметры соединения вручную** – ручная настройка соединения	
Выбор соединения с БД	Дерево	Нет	Доступно при выборе типа соединения «Глобальное соединение». Выбор соединения с базой данных из преднастроенного списка	
Адрес сервера	Строка	Нет	Доступно при выборе типа соединения «Задать параметры соединения вручную». Адрес сервера с базой данных	
Порт	Целое число	Нет	Доступно при выборе типа соединения «Задать параметры соединения вручную».	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
			Порт сервера с базой данных	
База данных	Строка	Нет	Доступно при выборе типа соединения «Задать параметры соединения вручную». Наименование базы данных	
Пользователь	Строка	Нет	Доступно при выборе типа соединения «Задать параметры соединения вручную». Логин пользователя	
Пароль	Шифрованная строка	Нет	Доступно при выборе типа соединения «Задать параметры соединения вручную». Пароль пользователя	

6.2.2.3 Группа «Общие настройки блока»

Аналогично приведенным в п. 6.1.1.1.3.

6.2.2.4 Входы

В блоке нет входов.

6.2.2.5 Выходы

Название выхода	Тип	Описание
БД	Соединение с БД	Уникальный идентификатор соединения с базой данных. При каждом расчете графа генерируется заново

6.2.2.6 События блока

В блоке нет событий.

6.2.2.7 Описание работы блока

В момент расчета графа блок открывает соединение с базой данных по заданным параметрам и передает уникальный идентификатор открытого соединения.

По этому идентификатору все блоки могут взаимодействовать с указанной в блоке базой данных. В конце расчета соединение закрывается, и при следующем расчете будет сгенерирован новый идентификатор.

6.2.2.7.1 Проверка соединения

Для проверки корректности соединения с базой данных, после ввода всех настроек можно нажать на кнопку «**Проверка соединения**»:

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

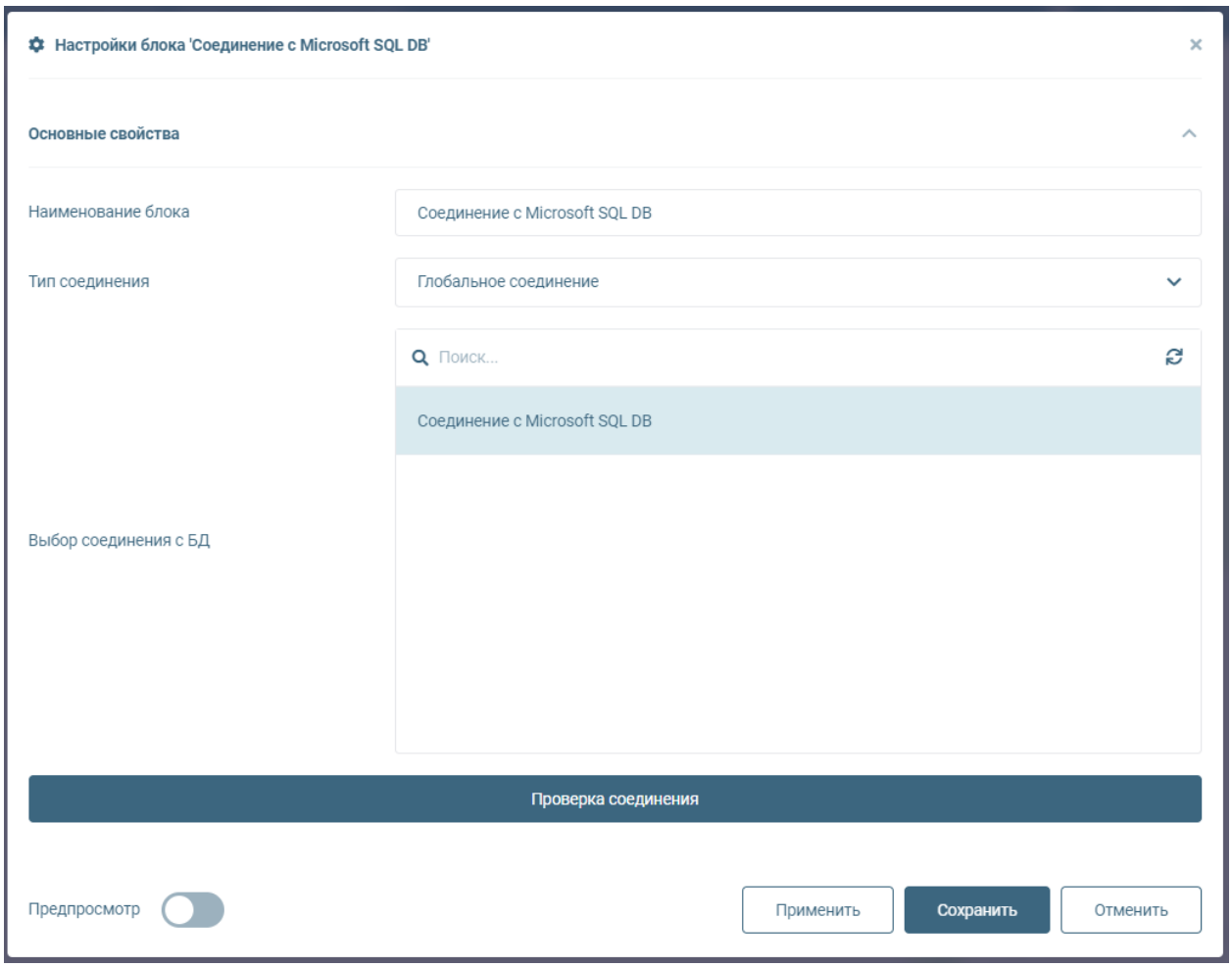


Рисунок 292 – Настройка блока «Соединение с Microsoft SQL DB»

После чего будет выдано сообщение об успешном или не успешном соединении с базой данных.

6.2.3 Блок «Соединение с Oracle DB»

6.2.3.1 Назначение

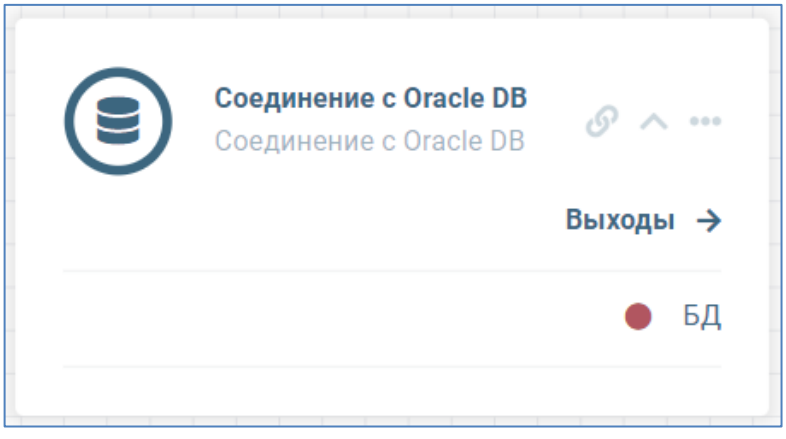


Рисунок 293 – Пример блока «Соединение с Oracle DB»

Блок позволяет настроить соединение с базой данных Oracle.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.2.3.2 Настройки блока

Название параметра	Тип параметра	Обязательный	Описание	Пример
Группа «Основные свойства»				
Тип соединения	Выпадающий список	Да	Выбор способа соединения с базой данных. Доступно два варианта: • **Глобальное соединение** – выбор преднастроенного соединения с базой данных; • **Задать параметры соединения вручную** – ручная настройка соединения	
Выбор соединения с БД	Дерево	Нет	Доступно при выборе типа соединения «Глобальное соединение». Выбор соединения с базой данных из преднастроенного списка	
Адрес сервера	Строка	Нет	Доступно при выборе типа соединения «Задать параметры соединения вручную». Адрес сервера с базой данных	
Порт	Целое число	Нет	Доступно при выборе типа соединения «Задать параметры соединения вручную». Порт сервера с базой данных	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
SID	Строка	Нет	Доступно при выборе типа соединения «Задать параметры соединения вручную». SID для доступа к базе данных. Каждый экземпляр базы данных идентифицируется с помощью SID (SystemIDentifier – системный идентификатор). SID состоит из алфавитно-цифровых символов, хранится в переменной среды ORACLE_SID и используется утилитами и сетевыми компонентами для доступа к базе данных	
Пользователь	Строка	Нет	Доступно при выборе типа соединения «Задать параметры соединения вручную». Логин пользователя	
Пароль	Шифрованная строка	Нет	Доступно при выборе типа соединения «Задать параметры соединения вручную». Пароль пользователя	

6.2.3.3 Группа «Общие настройки блока»

Аналогично приведенным в п. 6.1.1.1.3.

6.2.3.4 Входы

В блоке нет входов.

6.2.3.5 Выходы

Название выхода	Тип	Описание
БД	Соединение с БД	Уникальный идентификатор соединения с базой данных. При каждом расчете графа генерируется заново.

6.2.3.6 События блока

В блоке нет событий.

6.2.3.7 Описание работы блока

В момент расчета графа блок открывает соединение с базой данных по заданным параметрам и передает уникальный идентификатор открытого соединения.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

По этому идентификатору все блоки могут взаимодействовать с указанной в блоке базой данных. В конце расчета соединение закрывается, и при следующем расчете будет сгенерирован новый идентификатор.

6.2.3.7.1 Проверка соединения

Для проверки корректности соединения с базой данных, после ввода всех настроек можно нажать на кнопку «**Проверка соединения**»:

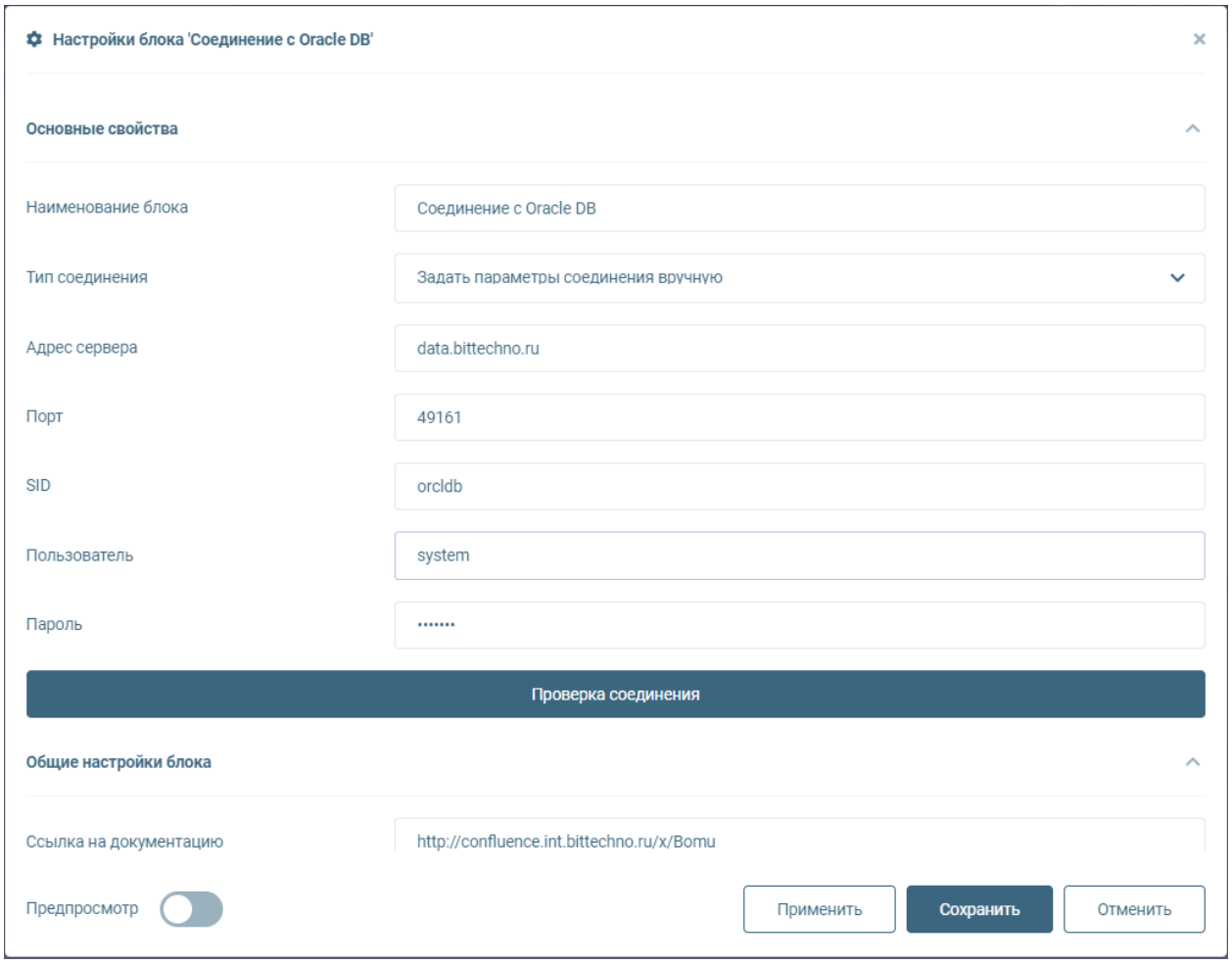


Рисунок 294 – Пример соединения с Oracle DB

После чего будет выдано сообщение об успешном или не успешном соединении с базой данных.

6.2.4 Блок «Соединение с SAP BI»

6.2.4.1 Назначение

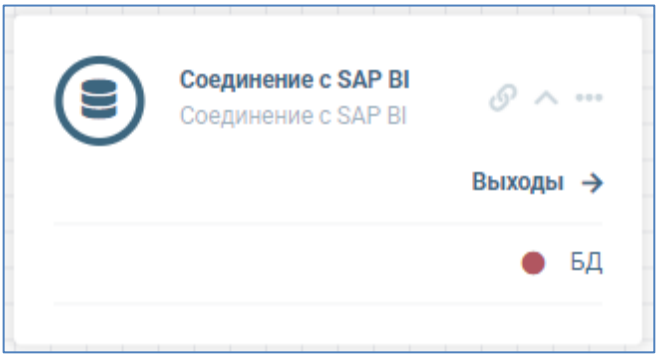


Рисунок 295 – Пример блока «Соединение с SAP BI»

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Блок позволяет настроить соединение с SAP BI для работы по XMLA протоколу обмена данными.

6.2.4.2 Настройки блока

Название параметра	Тип параметра	Обязательный	Описание	Пример
Группа «Основные свойства»				
URL	Строка	Да	Адрес для соединения с сервером SAP BI	
Пользователь	Строка	Да	Логин пользователя	
Пароль	Шифрованная строка	Да	Пароль пользователя	

6.2.4.3 Группа «Общие настройки блока»

Аналогично приведенным в п. 6.1.1.1.3.

6.2.4.4 Входы

В блоке нет входов.

6.2.4.5 Выходы

Название выхода	Тип	Описание
БД	Соединение с БД	Уникальный идентификатор соединения с базой данных. При каждом расчете графа генерируется заново.

6.2.4.6 События блока

В блоке нет событий.

6.2.4.7 Описание работы блока

В момент расчета графа блок открывает соединение с SAP BI по заданным параметрам и передает уникальный идентификатор открытого соединения. По этому идентификатору блоки типа «SAP – запрос» могут взаимодействовать с указанным в блоке инстансом SAP BI. В конце расчета соединение закрывается, и при следующем расчете будет сгенерирован новый идентификатор.

6.2.4.7.1 Проверка соединения

Для проверки корректности соединения, после ввода всех настроек можно нажать на кнопку «**Проверка соединения**":

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

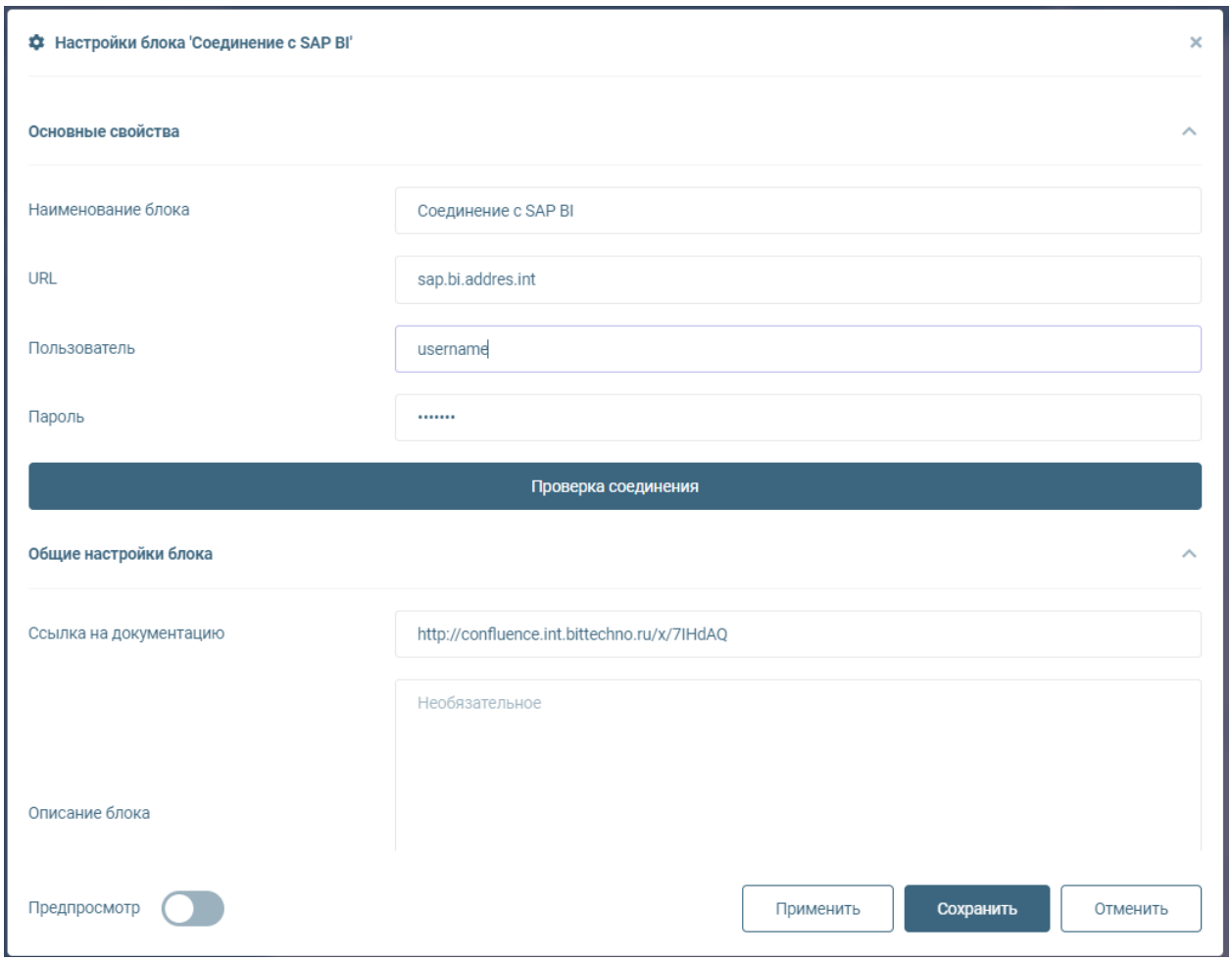


Рисунок 296 – Пример соединения с SAP BI

После чего будет выдано сообщение об успешном или не успешном соединении с SAP BI.

6.2.5 Блок «Соединение с Apache Kafka»

6.2.5.1 Назначение

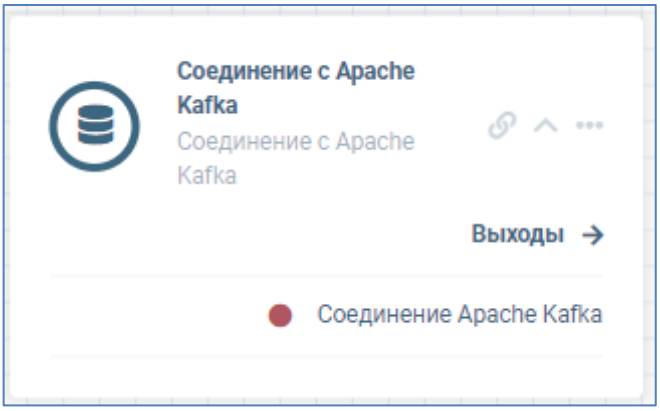


Рисунок 297 – Пример блока «Соединение с Apache Kafka»

Блок позволяет настроить соединение с очередью сообщений Apache Kafka. В блоке доступно два варианта подключения:

1. На чтение – режим Чтение (Consumer)
2. На запись - режим Запись (Producer). В версии 0.37 и ранее режим есть в настройках, но не поддерживается полностью ЦП УвП.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Блок может работать только в связке с блоком "Apache Kafka Consumer" для получения данных и "Apache Kafka Producer" для записи данных.

.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.2.5.2 Настройки блока

Название параметра	Тип параметра	Обязательный	Описание	Пример
Группа "Специфические настройки блока"				
Тип соединения	Перечисление	Да		
Адреса серверов	Строка	Нет	Адрес сервера очереди сообщений. Настройка: bootstrap.serversbootstrap.servers	
Имя группы	Строка	Нет	Идентификатор группы от имени которой читать или писать сообщения. Настройка: group.id	
Тип начала получения сообщений	Перечисление	Да	**Только для типа Consumer.** Способ чтения сообщений из очереди. Настройка: auto.offset.reset 1. Получить неотправленные сообщения (earliest) 2. Только новые (latest)	
Тип авторизации	Перечисление	Нет	Какой тип авторизации будет использоваться при подключении к Kafka. Настройка: security.protocol	
Пользователь	Строка	Да	Логин пользователя. Настройка: sasl.username. Только для типов авторизации SASL_PLAINTEXT, SASL_SSL	
Пароль	Шифрованная строка	Да	Пароль пользователя. Настройка: sasl.password. Только для типов авторизации SASL_PLAINTEXT, SASL_SSL	
Тип подключения клиента	Перечисление	Нет	Тип подключения пользователя. Настройка: sasl.mechanism. Только для типов авторизации SASL_PLAINTEXT, SASL_SSL	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Путь до файла сертификата ca-cert.pem	Строка	Нет	Путь с названием файла сертификата ca.pem. Сертификат должен быть на сервере. По умолчанию возьмется сертификат указанный в настройках приложения. Настройка: ssl.ca.location. Только для типов авторизации SSL, SASL_SSL, SSL с сертификатом пользователя	
Путь до файла сертификата client-key.pem	Строка	Нет	Путь с названием файла сертификата пользователя client.pem. Сертификат должен быть на сервере. По умолчанию возьмется сертификат указанный в настройках приложения. Настройка:ssl.key.location. Только для типа авторизации SSL с сертификатом пользователя	
Путь до файла сертификата client-certificate.pem	Строка	Нет	Путь с названием файла сертификата пользователя cerf.pem. Сертификат должен быть на сервере. По умолчанию возьмется сертификат указанный в настройках приложения. Настройка: ssl.certificate.location. Только для типа авторизации SSL с сертификатом пользователя	
Проверка имени сервера	Флаг	Нет	Флаг отвечающий, нужно ли проверять сертификат. Настройка: enable.ssl.certificate.verification. Только для типов авторизации SSL, SASL_SSL, SSL с сертификатом пользователя	
Дополнительные параметры	json	Нет	При использовании пользователя/пароля нужно указать параметры security.protocol и sasl.mechanism, иначе по умолчанию будут SASL_SSL и PLAIN	`{ "enable.auto.commit": false, "security.protocol": "SASL_PLAINTEXT", "sasl.mechanism": "PLAIN" }`

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.2.5.3 Группа «Общие настройки блока»

Аналогично приведенным в п. 6.1.1.1.3.

6.2.5.4 Входы

В блоке нет входов.

6.2.5.5 Выходы

Название выхода	Тип	Описание
Соединение Apache Kafka	Соединение с Apache Kafka	Уникальный идентификатор соединения с очередью Apache Kafka. При каждом расчете графа генерируется заново.

6.2.5.6 События блока

В блоке нет событий.

6.2.5.7 Описание работы блока

В момент расчета графа блок создает описание подключения к очереди Apache Kafka, по которому блок «Apache Kafka Topic» сможет подключиться к очереди и считать из нее сообщения.

6.2.6 Блок «Соединение SOAP»

6.2.6.1 Назначение

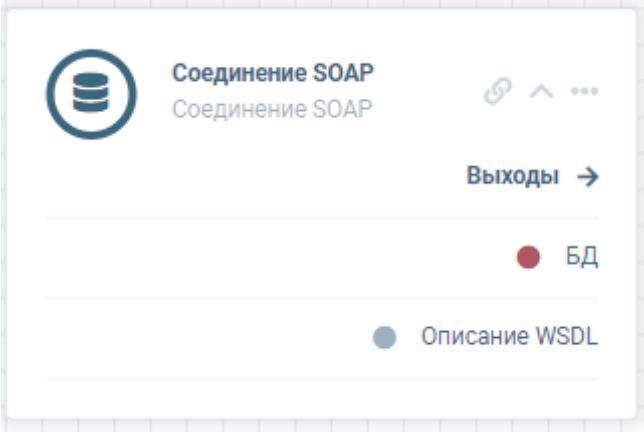


Рисунок 298 – Пример блока «Соединение SOAP»

Блок позволяет настроить соединение с SAP BI для работы по XMLA протоколу обмена данными.

6.2.6.2 Настройки блока

Название параметра	Тип параметра	Обязательный	Описание	Пример
Группа «Основные свойства»				
Адрес	Строка	Да	Адрес для соединения с SOAP сервисом	
Пользователь	Строка	Да	Логин пользователя	
Пароль	Шифрованная строка	Да	Пароль пользователя	

6.2.6.3 Группа «Общие настройки блока»

Аналогично приведенным в п. 6.1.1.1.3.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.2.6.4 Входы

В блоке нет входов.

6.2.6.5 Выходы

Название выхода	Тип	Описание
БД	SOAP connect	Настройки подключение к SOAP сервису
Описание WSDL	XML	WSDL схема, возвращаемая сервисом

6.2.6.6 События блока

В блоке нет событий.

6.2.6.7 Описание работы блока

В момент расчета графа блок открывает соединение с SOAP сервисом по заданным параметрам и передает далее информацию о соединении. По этой информации блоки типа «SOAP – запрос» могут взаимодействовать с указанным в блоке сервисом.

6.2.6.7.1 Проверка соединения

Для проверки корректности соединения, после ввода всех настроек можно нажать на кнопку «**Проверка соединения**»:

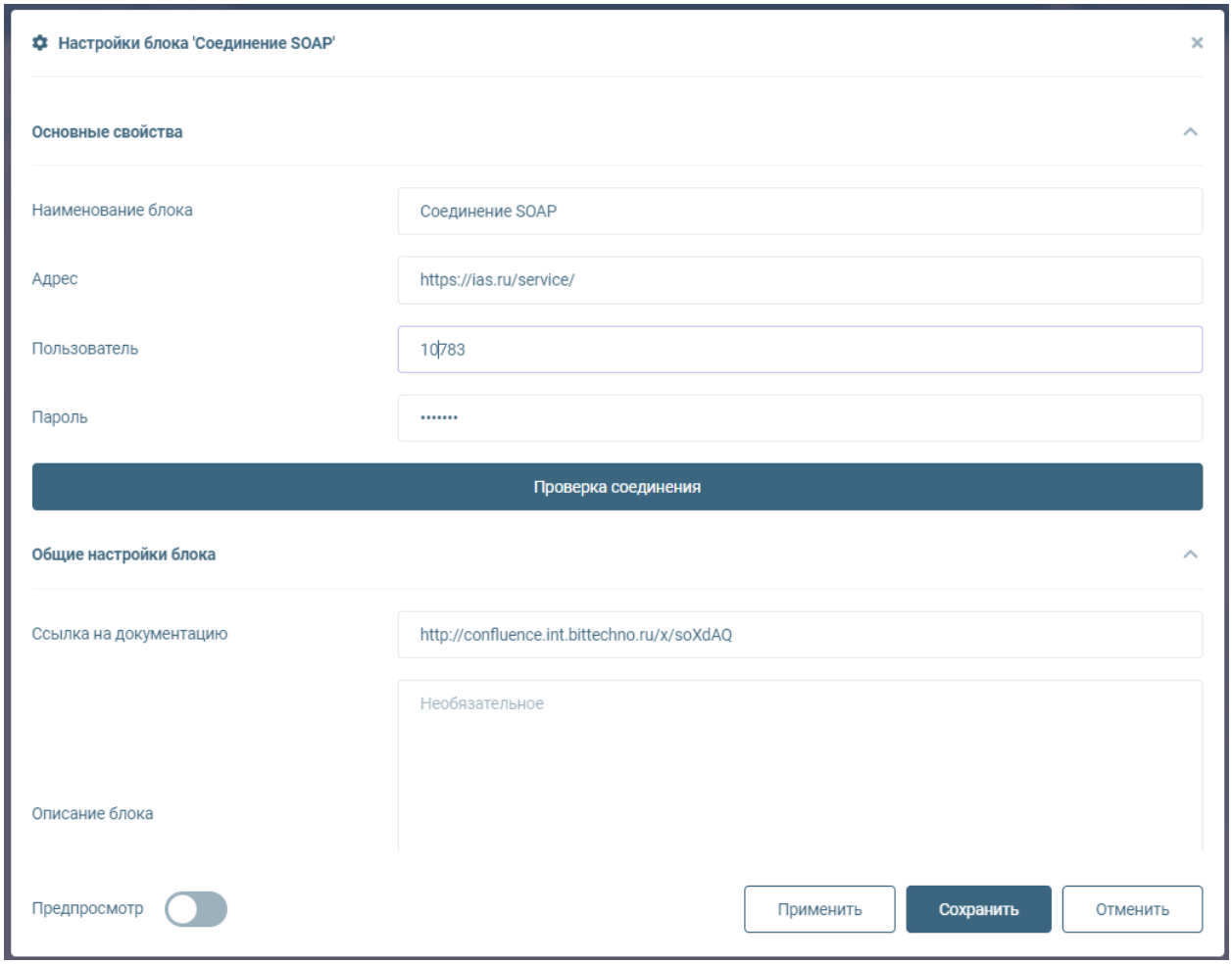


Рисунок 299 – Пример соединения SOAP

После чего будет выдано сообщение об успешном или не успешном соединении с SOAP сервисом.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.2.7 Блок «Соединение с Mongo»

6.2.7.1 Назначение

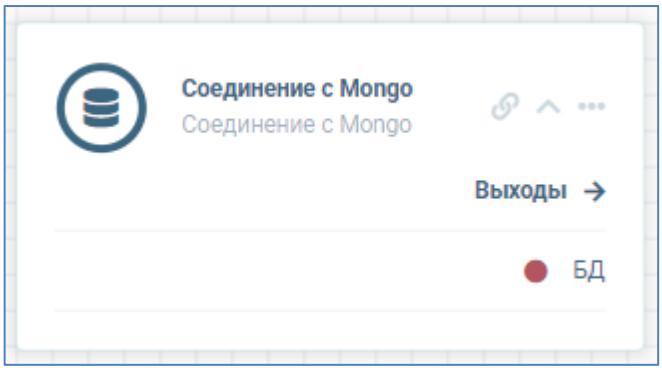


Рисунок 300 – Пример блока «Соединение с Mongo»

Блок позволяет настроить соединение с базой данных MongoDB.

6.2.7.2 Настройки блока

Название параметра	Тип параметра	Обязательный	Описание	Пример
Группа «Основные свойства»				
Тип соединения	Выпадающий список	Да	Выбор способа соединения с базой данных. Доступно два варианта: • **Глобальное соединение** – выбор преднастроенного соединения с базой данных; • **Задать параметры соединения вручную** – ручная настройка соединения	
Выбор соединения с БД	Дерево	Нет	Доступно при выборе типа соединения «Глобальное соединение». Выбор соединения с базой данных из преднастроенного списка	
Адрес сервера	Строка	Нет	Доступно при выборе типа соединения «Задать параметры соединения вручную». Адрес сервера с базой данных	
Порт	Целое число	Нет	Доступно при выборе типа соединения «Задать	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
			параметры соединения вручную». Порт сервера с базой данных	
База данных	Строка	Нет	Доступно при выборе типа соединения «Задать параметры соединения вручную». Идентификатор БД на сервере Mongo	
Аутентификация	Выпадающий список	Да	Доступно при выборе типа соединения «Задать параметры соединения вручную». Выбор типа соединения с БД – с использованием имени пользователя и пароля, или без них	
Пользователь	Строка	Нет	Доступно при выборе типа соединения «Задать параметры соединения вручную» и аутентификации «Пользователь/Пароль» Логин пользователя	
Пароль	Шифрованная строка	Нет	Доступно при выборе типа соединения «Задать параметры соединения вручную» и аутентификации «Пользователь/Пароль» Пароль пользователя	

6.2.7.3 Группа «Общие настройки блока»

Аналогично приведенным в п. 6.1.1.1.3.

6.2.7.4 Входы

В блоке нет входов.

6.2.7.5 Выходы

Название выхода	Тип	Описание
БД	Соединение с БД	Уникальный идентификатор соединения с базой данных. При каждом расчете графа генерируется заново

6.2.7.6 События блока

В блоке нет событий.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.2.7.7 Описание работы блока

В момент расчета графа блок открывает соединение с базой данных по заданным параметрам и передает уникальный идентификатор открытого соединения. По этому идентификатору все блоки могут взаимодействовать с указанной в блоке базой данных. В конце расчета соединение закрывается, и при следующем расчете будет сгенерирован новый идентификатор.

6.2.7.7.1 Проверка соединения

Для проверки корректности соединения с базой данных, после ввода всех настроек можно нажать на кнопку «**Проверка соединения**»:

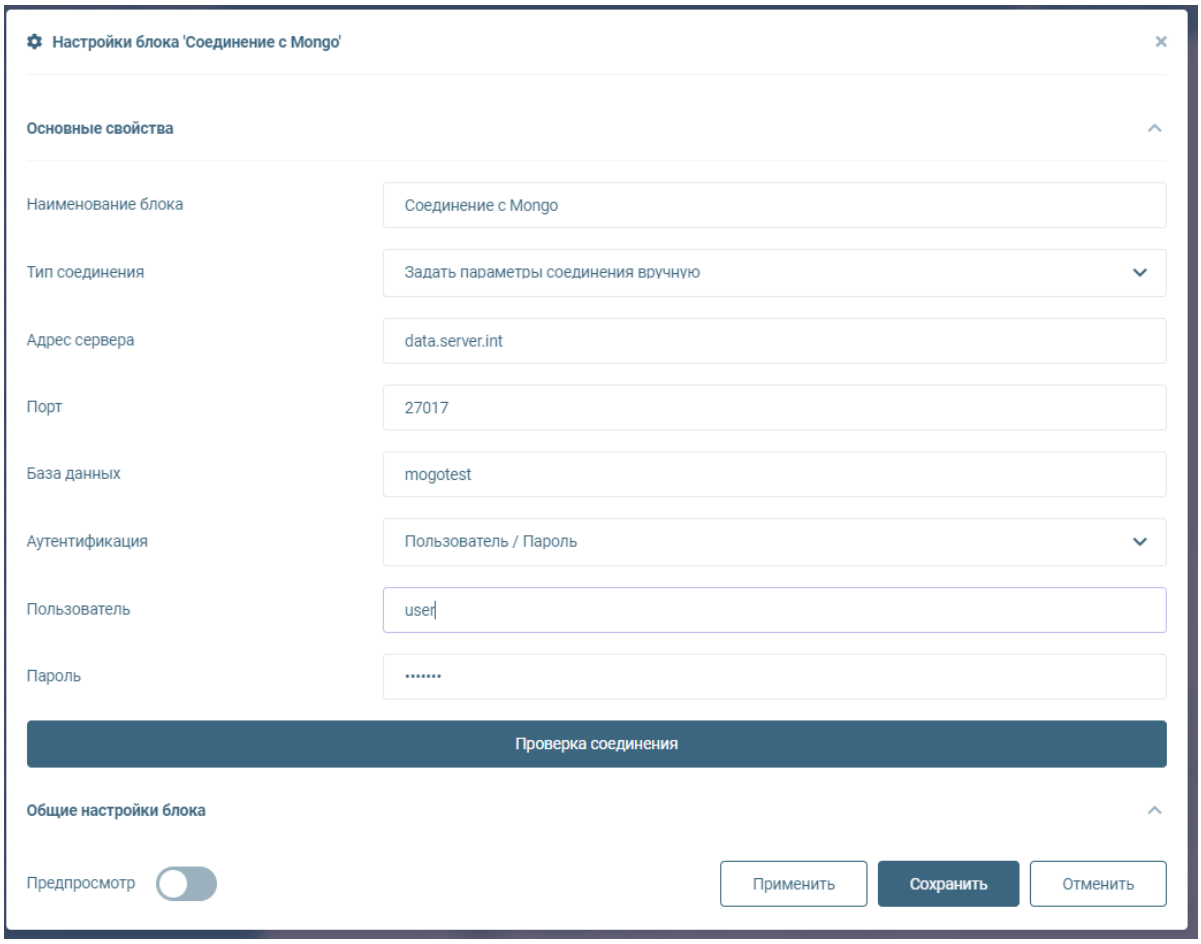


Рисунок 301 – Пример соединения с Mongo

После чего будет выдано сообщение об успешном или не успешном соединении с базой данных.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.2.8 Блок «Соединение с ClickHouse»

6.2.8.1 Назначение

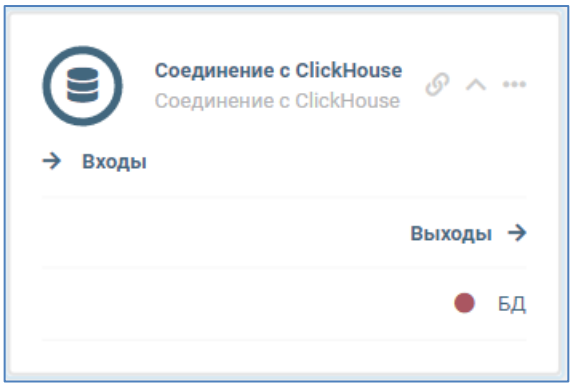


Рисунок 302 – Пример блока «Соединение с ClickHouse»

Блок позволяет настроить соединение с базой данных ClickHouse.

6.2.8.2 Настройки блока

Название параметра	Тип параметра	Обязательный	Описание	Пример
Группа «Основные свойства»				
Тип соединения	Выпадающий список	Да	Выбор способа соединения с базой данных. Доступно два варианта: • **Глобальное соединение** – выбор преднастроенного соединения с базой данных; • **Задать параметры соединения вручную** – ручная настройка соединения	
Выбор соединения с БД	Дерево	Нет	Доступно при выборе типа соединения «Глобальное соединение». Выбор соединения с базой данных из преднастроенного списка.	
Адрес сервера	Строка	Нет	Доступно при выборе типа соединения «Задать параметры соединения вручную».	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
			Адрес сервера с базой данных	
Порт	Целое число	Нет	Доступно при выборе типа соединения «Задать параметры соединения вручную». Порт сервера с базой данных	
База данных	Строка	Нет	Доступно при выборе типа соединения «Задать параметры соединения вручную». Наименование базы данных	
Пользователь	Строка	Нет	Доступно при выборе типа соединения «Задать параметры соединения вручную». Логин пользователя	
Пароль	Шифрованная строка	Нет	Доступно при выборе типа соединения «Задать параметры соединения вручную». Пароль пользователя	
Дополнительные параметры	JSON	Нет	Доступно при выборе типа соединения «Задать параметры соединения вручную». Дополнительные параметры для подключения к базе данных	Например: чтобы включить поддержку работы с датафреймом, надо задать такие настройки: `{"use_numpy" : true}`

6.2.8.3 Группа «Общие настройки блока»

Аналогично приведенным в п. 6.1.1.1.3.

6.2.8.4 Входы

В блоке нет входов.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.2.8.5 Выходы

Название выхода	Тип	Описание
БД	Соединение с БД	Уникальный идентификатор соединения с базой данных. При каждом расчете графа генерируется заново.

6.2.8.6 События блока

В блоке нет событий.

6.2.8.7 Описание работы блока

В момент расчета графа блок открывает соединение с базой данных по заданным параметрам и передает уникальный идентификатор открытого соединения.

По этому идентификатору все блоки могут взаимодействовать с указанной в блоке базой данных.

В конце расчета соединение закрывается, и при следующем расчете будет сгенерирован новый идентификатор.

6.2.8.7.1 Проверка соединения

Для проверки корректности соединения с базой данных, после ввода всех настроек можно нажать на кнопку «**Проверка соединения**».

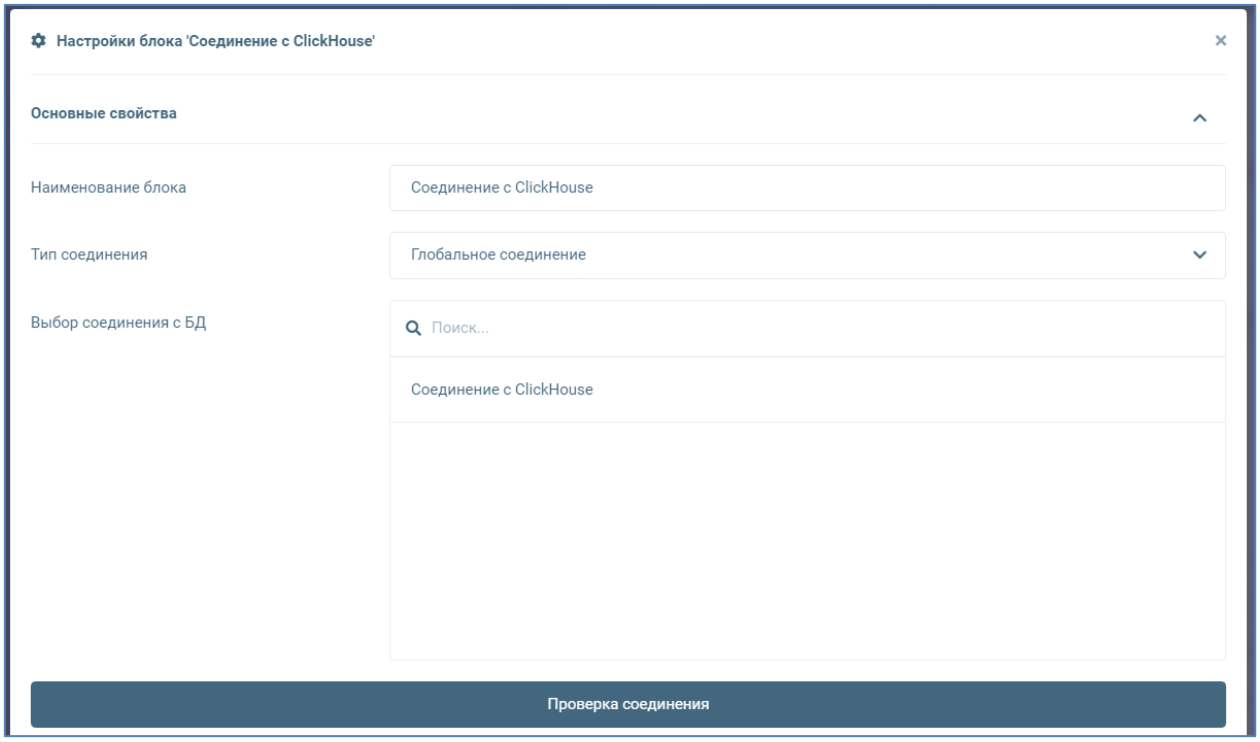


Рисунок 303 – Пример соединения с ClickHouse

После чего будет выдано сообщение об успешном или не успешном соединении с базой данных.

6.2.8.8 Дополнительные ссылки

https://clickhouse.com/docs/ru/

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.2.9 Блок «SQL – выражение»

6.2.9.1 Назначение

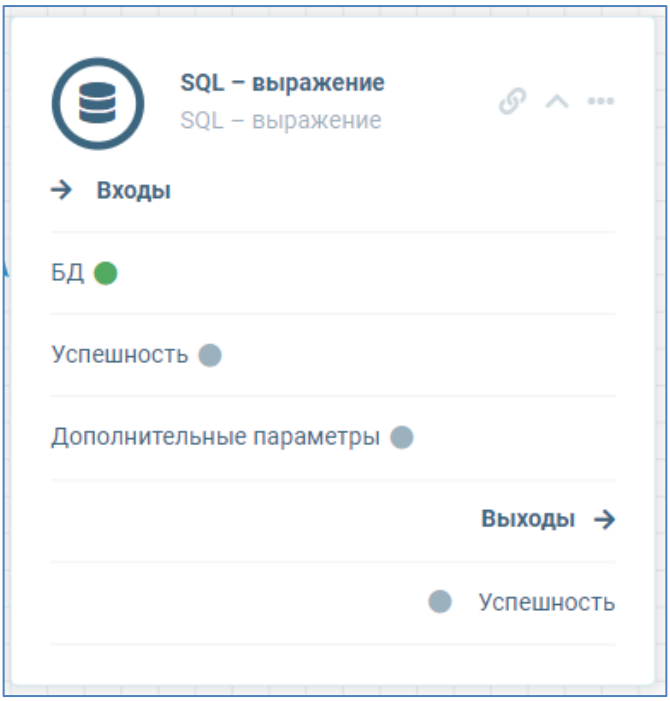


Рисунок 304 – Пример блока «SQL – выражение»

Блок позволяет выполнить на сервере БД какую-либо SQL команду.

6.2.9.2 Настройки блока

Название параметра	Тип параметра	Обязательный	Описание	Пример
Группа «Основные свойства»				
Запрос SQL	Строка (SQL запрос)	Да	SQL-выражение для выполнения на сервере. Допустимы запросы, не возвращающие какие-либо данные. Возможна подстановка запроса с JINJA-шаблоном. Данные для шаблона применяются из входа «Дополнительные параметры»	**Пример** `UPDATE table1 SET id=1 WHERE name='one';`

6.2.9.3 Группа «Общие настройки блока»

Аналогично приведенным в п. 6.1.1.1.3.

6.2.9.4 Входы

Название входа	Тип	Обязательный	Описание
БД	Соединение с БД	Да	Соединение с базой данных, из которой необходимо извлекать данные. Поддерживаются разные диалекты баз, а именно:

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название входа	Тип	Обязательный	Описание
			• PostgreSQL • MSSQL • Oracle • ClickHouse
Успешность	Флаг	Нет	Вход для запуска расчета блока в случае если нет входов (на него можно подать выход Успешность другого блока)
Дополнительные параметры	JSON	Нет	Массив объектов или временных рядов, атрибуты которых могут быть использованы для параметризации запроса

6.2.9.5 Выходы

Название выхода	Тип	Обязательный	Описание
Успешность	Флаг	Нет	Выход для продолжения расчетов в случае успешного выполнения блока. Этот выход можно подать вход Успешность других блоков.

6.2.9.6 События блока

В блоке нет событий.

6.2.9.7 Описание работы блока

Блок выполняет SQL запрос или выражение(при необходимости – формируя его из JINJA-шаблона) в присоединенной к нему базе данных, после чего выдает на выход «Успешность» сигнал об успешном выполнении.

Важно!

1. Для разных баз данных должен применяться свой диалект SQL.
2. При последовательном исполнении нескольких запросов в одном блоке данные запросы выполняются **одной транзакцией**.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.2.9.8 Описание работы визуализатора

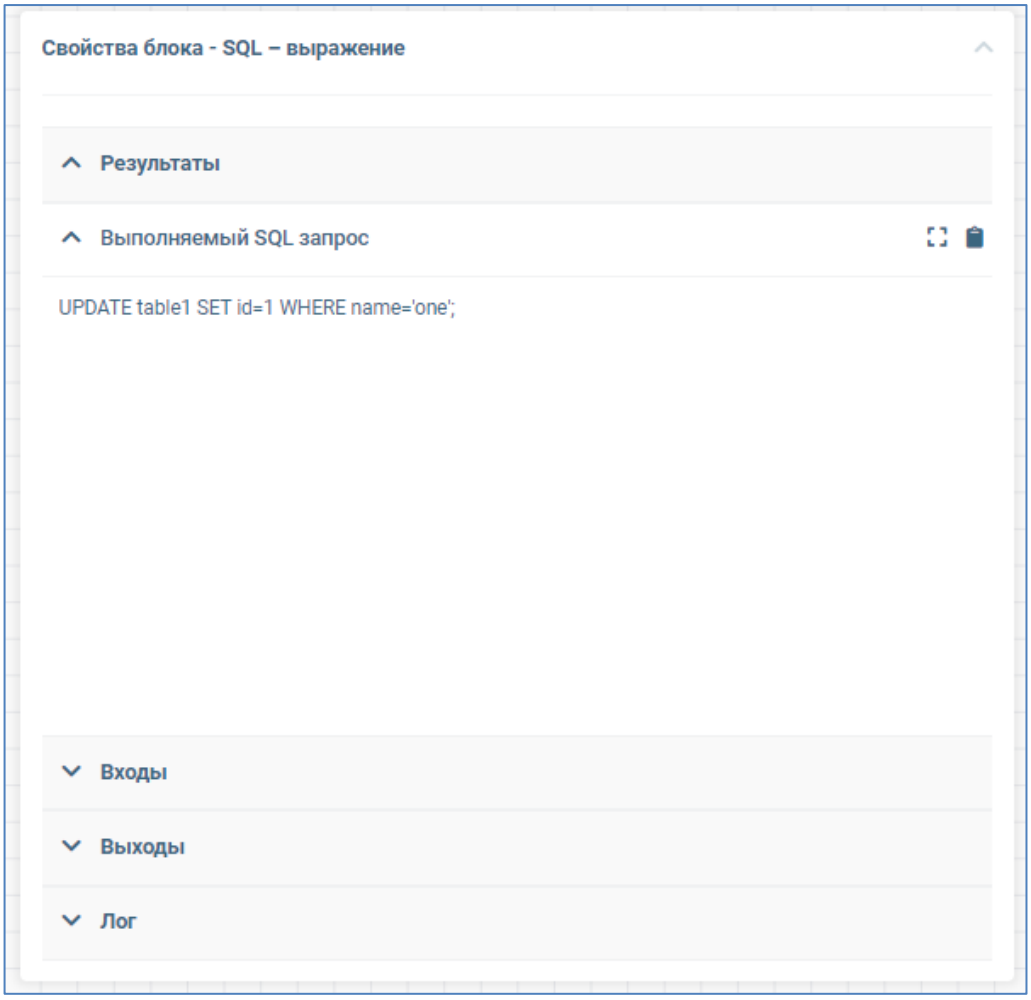


Рисунок 305 – Свойства блока «SQL – выражение»

При использовании режима отладки (см. п. 7.5), для блока доступен системный визуализатор с полным текстом SQL выражения, с примененным JINJA-шаблоном, если такой использовался.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.2.10 Блок «SQL – запрос для рядов»

6.2.10.1 Назначение

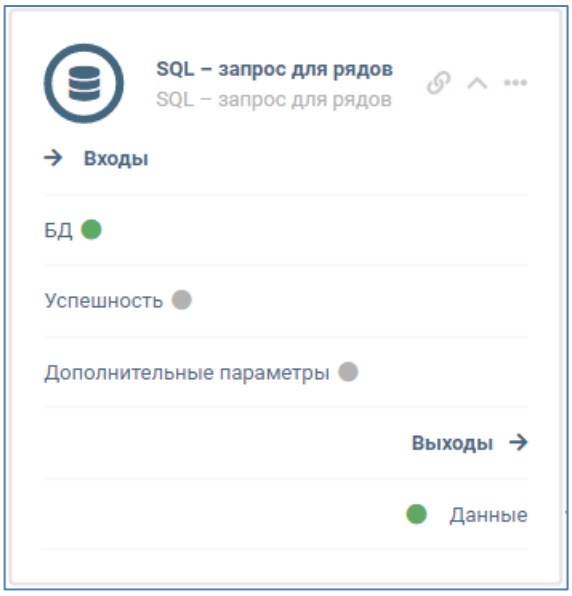


Рисунок 306 – Пример блока «SQL – запрос для рядов»

Блок позволяет получать массив рядов из БД по SQL-запросу. Так же доступен write-back для значений ряда.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.2.10.2 Настройки блока

Название параметра	Тип параметра	Обязательный	Описание	Пример
Группа «Основные свойства»				
Запрос SQL	Строка (SQL запрос)	Да	SQL-запрос для извлечения данных. Допустимы только запросы, возвращающие какие-либо данные. Возможна подстановка запроса с JINJA-шаблоном. Данные для шаблона применяются из входа «Дополнительные параметры»	Приме SELECT * FROM some_table

Параметры привязки	JSON	Да	Параметры, на основе которых формируются объекты. Без указания привязки объекты не будут иметь первичных ключей.
			• **pkey** – массив ключей, на основании которого будет сформирован первичный ключ каждого объекта;
			• **vl** – единичное поле, которое будет отвечать за значение объекта;
			• **params** – расширенные параметры объекта.
			Параметр необязателен для выполнения SQL-блока, если получаемый ряд не будет далее передан в один из блоков статистических моделей или в блоки «Выражение» и «Правила».
			• **ts** – массив с настройками формата даты:
			  • **key** – ключ поля, в котором находится значение даты;
			  • **mask** – маска даты. Используется для приведения значений даты в строковый вид при сериализации в JSON.
			Доступные форматы для маски
			`%a Weekday as locale’s abbreviated name. Mon`
			`%A Weekday as locale’s full name. Monday`
			`%w Weekday as a decimal number, where 0 is Sunday and 6 is Saturday. 1`
				Код привязок
				`{`
				`"pkey": [`
				`"id",`
				`"region_id",`
				`"date"`
				`],`
				`"vl": "value:,`
				`"params": {`
				`"ts": [`
				`{`
				`"key": "date",`
				`"mask": "%Y.%m.%d",`
				`"frequency": "D"`
				`}`
				`],`
				`"lazy": ["lazy_vl_1"]`
				`}`
				}

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
			%d Day of the month as a zero-padded decimal number. 30 %-d Day of the month as a decimal number. (Platform specific) 30 %b Month as locale’s abbreviated name. Sep %B Month as locale’s full name. September %m Month as a zero-padded decimal number. 09 %-m Month as a decimal number. (Platform specific) 9 %y Year without century as a zero-padded decimal number. 13 %Y Year with century as a decimal number. 2013 %H Hour (24-hour clock) as a zero-padded decimal number. 07 %-H Hour (24-hour clock) as a decimal number. (Platform specific) 7 %I Hour (12-hour clock) as a zero-padded decimal number. 07 %-I Hour (12-hour clock) as a decimal number. (Platform specific) 7 %p Locale’s equivalent of either AM or PM. AM %M Minute as a zero-padded decimal number. 06 %-M Minute as a decimal number. (Platform specific) 6 %S Second as a zero-padded decimal number. 05 %-S Second as a decimal number. (Platform specific) 5 %f Microsecond as a decimal number, zero-padded on the left. 000000 %z UTC offset in the form +HHMM or -HHMM (empty string if the object is naive). %Z Time zone name (empty string if the object is naive). %j Day of the year as a zero-padded decimal number. 273 %-j Day of the year as a decimal number. (Platform specific) 273	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
			%U Week number of the year (Sunday as the first day of the week) as a zero padded decimal number. All days in a new year preceding the first Sunday are considered to be in week 0. 39 %W Week number of the year (Monday as the first day of the week) as a decimal number. All days in a new year preceding the first Monday are considered to be in week 0. 39 %c Locale’s appropriate date and time representation. Mon Sep 30 07:06:05 2013 %x Locale’s appropriate date representation. 09/30/13 %X Locale’s appropriate time representation. 07:06:05 %% A literal '%' character. % • **frequency** – период; • **lazy** – массив ключей, для которых необходимо произвести «ленивую загрузку» – сохранить в системную таблицу, а вместо значения прописать уникальный идентификатор поля. Не все блоки могут преобразовать «ленивую загрузку» обратно в значение. • **vl_type** – техническое поле, автоматически заполняется при наличии заполненной настройки атрибутов (см. ниже). Поле означает тип значения для отображения в таблице. возможные значения vl_type: 2 – строка 4 – вещественный	

Параметры привязки для рядов	JSON	Да	Параметры, на основании которых из объектов будет собран ряд данных.
			• **fkey** – массив первичных ключей, на основании которых будет собран ряд.
			Все ключи так же необходимо указать в настройке **pkey** из меню «Параметры привязки»
				Код привязок
				{
				«fkey": ["date"]
				}

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
Запрос обновления данных	Строка	Нет	Устаревшее поле, использовалось в блоках до версии 1. На текущий момент функционал отключен. Шаблон запроса для обновления атрибутов объектов. Пример шаблона обновления данных `INSERT INTO <имя таблицы> ({0}) VALUES ({1}) ON CONFLICT ({2}) DO UPDATE SET {3} = EXCLUDED.{3}` **Значения в фигурных скобках необходимо оставить в неизменном виде.** Требуется только указать имя таблицы	
Запрос удаления данных	Строка	Нет	Устаревшее поле, использовалось в блоках до версии 1. На текущий момент функционал отключен. Шаблон запроса для удаления атрибутов объектов Пример шаблона обновления данных `DELETE FROM <имя таблицы> WHERE {0}` **Значения в фигурных скобках необходимо оставить в неизменном виде.** Требуется только указать имя таблицы	
Наименование таблицы для сохранения данных	Строка	Нет	Наименование таблицы, в которую будут сохраняться данные при настроенном сохранении объектов (write-back)	
Раздел настроек «Настройка атрибутов»				
Наименования поля	Строка	Нет	Текстовое описание поля	
Идентификатор	Строка	Да	Идентификатор поля, как он указан в запросе	
Тип	Выпадающий список	Да	Тип поля. Для поля с типом «дата» обязательно наличие настройки формата даты в меню «Параметры привязки»	
Элемент с пустым значением	Флаг	Нет	Имеет ли элемент пустые значения.	
Входит в первичный ключ	Флаг	Нет	Входит ли поле в первичный ключ (настройка **pkey** из меню «Параметры привязки»).	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
			Обязательно указывать все поля, входящие в первичный ключ	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.2.10.3 Группа «Общие настройки блока»

Аналогично приведенным в п. 6.1.1.1.3.

6.2.10.4 Входы

Название входа	Тип	Обязательный	Описание
БД	Соединение с БД	Да	Соединение с базой данных, из которой необходимо извлекать данные. Поддерживаются разные диалекты баз, а именно: • PostgreSQL • MSSQL • Oracle • ClickHouse
Успешность	Флаг	Нет	Вход для запуска расчета блока в случае если нет входов (на него можно подать выход Успешность другого блока).
Дополнительные параметры	JSON	Нет	Массив объектов или временных рядов, атрибуты которых могут быть использованы для параметризации запроса.

6.2.10.5 Выходы

Название выхода	Тип	Описание

Данные	Ряд данных	Массив рядов данных
		Выход из блока
		{
		«pkey": [
		<Набор первичных ключей>
		],
		«vl": [
		<Массив значений ряда>
		],
		«attrs": [
		<Набор атрибутов – полей, не попавших ни в первичный ключ, ни в значение.>
		],
		«_params": {
		<Опциональные параметры объекта>,
		«vl": <Ключ поля, отвечающего за значение>,
		«fkey": [
		<Ключ поля, по которому собран ряд>
		]
		},
		«_source": <Идентификатор блока-источника, поле заполняется калькулятором автоматически>,
		}

6.2.10.6 События блока

В блоке нет событий.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.2.10.7 Описание работы блока

Блок выполняет SQL запрос (при необходимости – формируя его из JINJA-шаблона) из выбранной базы данных, после чего преобразует каждую полученную строку в объект по шаблону, заданному в поле «Параметры привязки».

После этого происходит поиск всех уникальных ключей **fkey** из поля «Параметры привязки для рядов». Количество уникальных значений будет являться длиной ряда.

Если количество объектов с уникальным первичным ключом больше, чем длина массива, значит первичный ключ не обеспечивает уникальности записей и ряд сформирован не будет (не рассчитается блок), и расчетчик отобразит ошибку:

Ошибка при недостаточности первичного ключа

"Обнаружено дублирование уникальных ключей! Первичный ключ недостаточен."

Все не уникальные атрибуты ряда будут «схлопнуты» из массива в единичный элемент. Остальные, уникальные атрибуты, будут переданы в качестве массива длиной равной длине ряда.

Результат работы блока – массив рядов.

В том случае, если заданы поля для сохранения данных (write-back), то при изменении данных в визуализаторе, стоящим после данного блока, происходит формирование запроса на добавление и/или обновление данных с последующей записей в выбранную таблицу (поле «Наименование таблицы для сохранения данных»).

В отличии от блока «SQL – запрос-справочник» (п. 6.2.11), для рядов доступно обновление только значений ряда (поле **vl** из меню «Параметры привязки»).

Особенности работы механизма wrtie-back

Для корректной работы сохранения данных в таблице или диаграмме требуется:

- наличие поля date_level (целочисленное) в таблице в базе данных;
- все поля, являющиеся не пустыми (стоит флаг not null) должны входить в первичный ключ.

Для разных баз данных должен применяться свой диалект SQL.

6.2.10.8 Динамические выходы блока

Динамические выходы – дочерний выход блока, отфильтрованный по выбранным первичным ключам ряда (ключи привязки для рядов не используются).

Для настройки необходимо навести курсор на выход «**Данные**» и нажать на выпадающий элемент меню.

Настройку динамических выходов необходимо производить на уже рассчитанном блоке.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

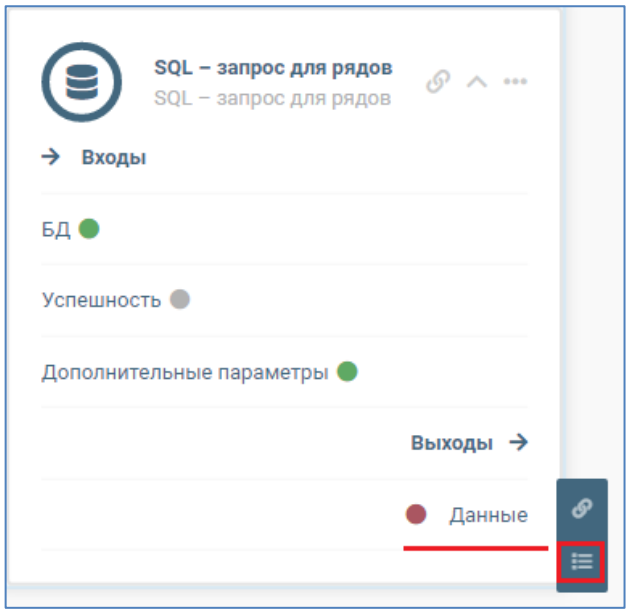


Рисунок 307 – Пример настройки выходов блока

После нажатия открывается окно с выбором переменных, представленное на рисунке ниже:

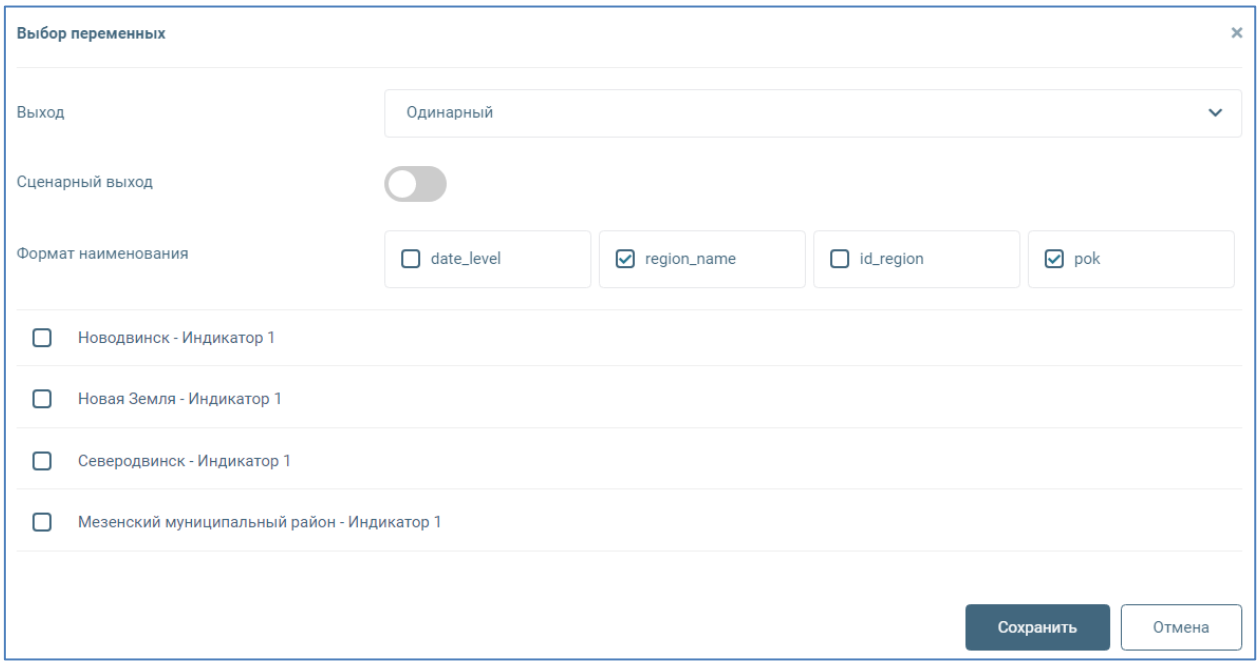


Рисунок 308 – Пример выбора переменных

- **Выход** – формат выхода данных. Выпадающее меню с двумя значениями:
  - **Одинарный** – каждый выбранный элемент создаст отдельный выход блока, содержащий один ряд данных.
  - **Массив** – все выбранные элементы объединяются в один массив (массив рядов). Для массива так же можно задать наименование выхода.
- **Формат наименования** – выбор первичных ключей и/или атрибутов объекта, которые будут отвечать за формирование наименования выхода.
  - Путем перетаскивания ключей (*drag-n-drop)* можно изменять порядок элементов.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

После выбора одной или нескольких переменных и нажатия на кнопку «Сохранить», у выбранного блока создадутся дополнительные выходы.

За одно редактирование переменных возможно создание нескольких одинарных выходов или одного массива (см. рисунок ниже).

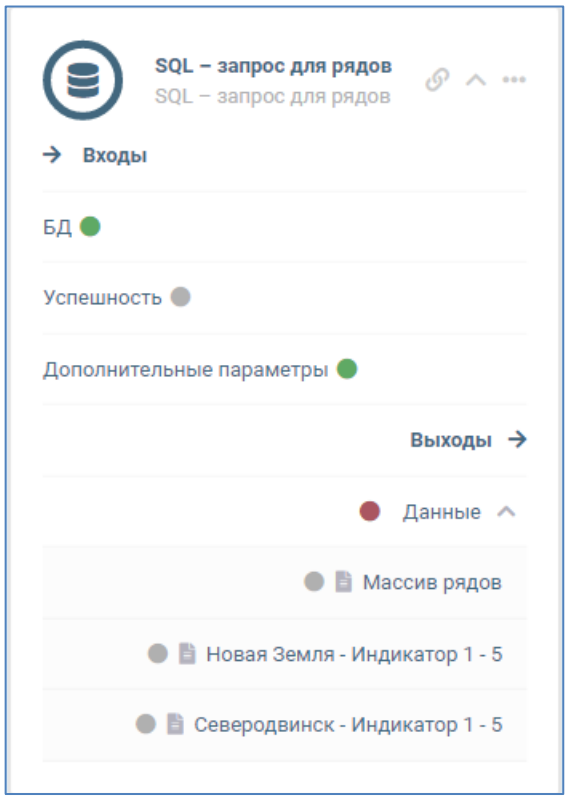


Рисунок 309 – Пример создания дополнительных выходов

*Блок с тремя добавленными выходами: одним **массивом** и двумя **одинарными** выходами.*

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

После создания выхода доступно так же его **редактирование** (иконка с карандашом) или **удаление** (иконка с мусорной корзиной):

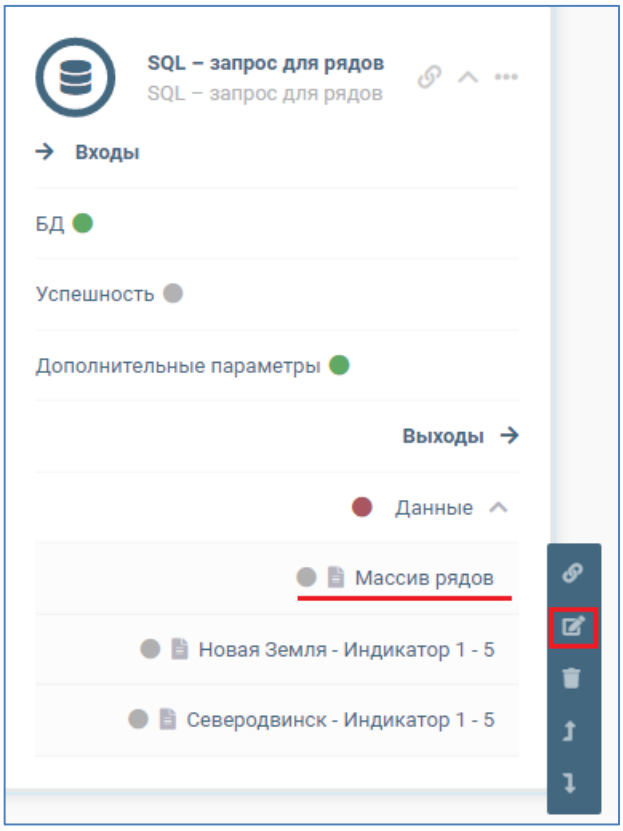


Рисунок 310 – Пример иконки редактирования

При выборе редактирования выхода открывается меню «**Выбор переменных»**, в котором можно изменить параметры выхода.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.2.10.9 Описание работы визуализатора

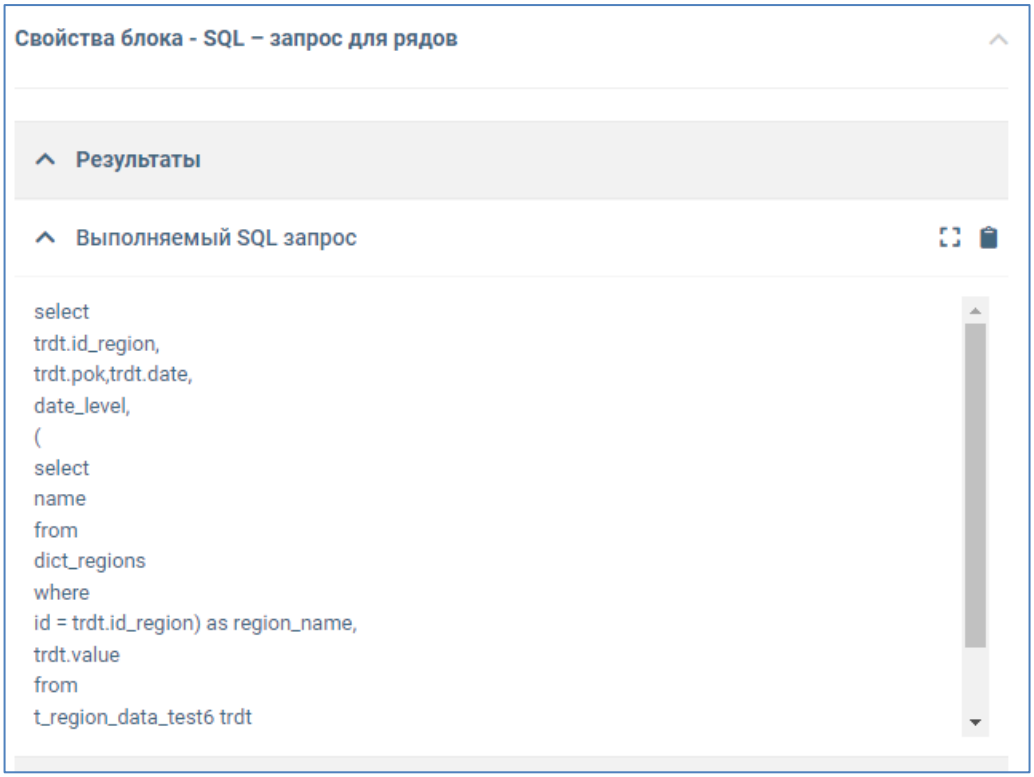


Рисунок 311 – Свойства блока «SQL – запрос для рядов»

При использовании режима отладки (см. п. 7.5), для блока доступен системный визуализатор с полным текстом SQL-запроса, с примененным JINJA-шаблоном, если такой использовался.

6.2.10.10 Блок «МастерSQL – запрос для рядов»

6.2.10.10.1 Назначение

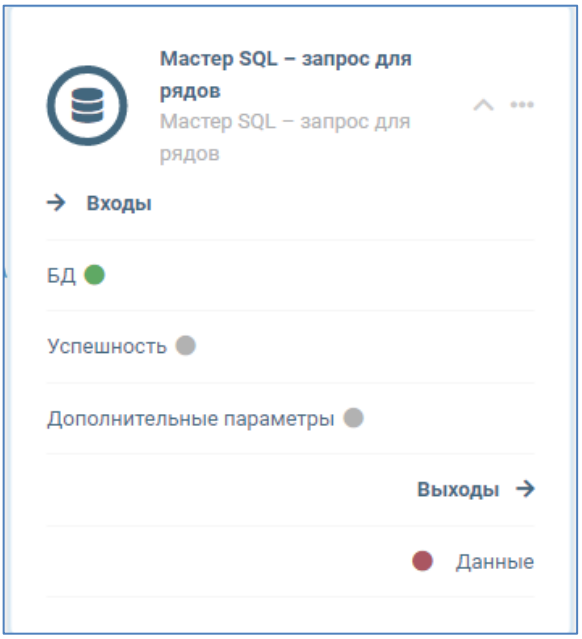


Рисунок 312 – Пример блока «МастерSQL – запрос для рядов»

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Блок позволяет получать массив рядов из БД по SQL-запросу. В отличии от блока «SQL – запрос для рядов» (п. 6.2.10), формирование запроса реализовано с помощью мастера.

Так же доступен write-back для значений ряда.

6.2.10.10.2 Настройки блока

Название параметра	Тип параметра	Обязательный	Описание	Пример
Атрибуты	Мастер	Да	Мастер заполнения атрибутов блока. Подробнее см. раздел Описание работы блока – Мастер «Атрибуты»	
SQL-запрос	Мастер	Да	Мастер по формированию запроса. Подробнее см. раздел Описание работы блока – Мастер «SQL-запрос»	

6.2.10.10.3 Группа «Общие настройки блока»

Аналогично приведенным в п. 6.1.1.1.3.

6.2.10.10.4 Входы

Название входа	Тип	Обязательный	Описание
БД	Соединение с БД	Да	Соединение с базой данных, из которой необходимо извлекать данные. Поддерживаются разные диалекты баз, а именно: • PostgreSQL • MSSQL • Oracle • ClickHouse
Успешность	Флаг	Нет	Вход для запуска расчета блока в случае если нет входов (на него можно подать выход Успешность другого блока).
Дополнительные параметры	JSON	Нет	Массив объектов или временных рядов, атрибуты которых могут быть использованы для параметризации запроса.

6.2.10.10.5 Выходы

Название выхода	Тип	Описание
Данные	Ряд данных	Массив рядов данных **Выход из блока** { "pkey": [<Набор первичных ключей>], "vl": [<Массив значений ряда>], "attrs": [

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название выхода	Тип	Описание

		<Набор атрибутов – полей, не попавших ни в первичный ключ, ни в значение.>
		],
		"_params": {
		<Опциональные параметры объекта>,
		"vl": <Ключ поля, отвечающего за значение>,
		"fkey": [
		<Ключ поля, по которому собран ряд>
		]
		},
		"_source": <Идентификатор блока-источника, поле заполняется калькулятором автоматически>,
		}

6.2.10.10.6 События блока

В блоке нет событий.

6.2.10.10.7 Описание работы блока

6.2.10.10.7.1 Работа с мастером

Для формирования запроса и выбора необходимых полей, требуется заполнить два мастера: **Атрибуты** и **SQL-запрос**.

Для каждого мастера доступно два режима работы: **ручной режим** (чаще всего – строковое поле для ввода) и **выбор данных**. Переключение режимов работы мастеров осуществляется нажатием на кнопку .

Обратите внимание, данные заведенные в разных режимах не пересекаются.

6.2.10.10.7.2 Мастер «Атрибуты»

Рисунок 313 – Мастер «Атрибуты» (ручной режим)

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Мастер – ручной режим

Название параметра	Тип параметра	Обязательный	Описание	Пример

Параметры привязки	JSON	Нет	Поле, совмещающее в себе поля «Параметры привязки» и «Параметры привязки для рядов» блока «SQL – запрос для рядов». Параметры, на основе которых формируются объекты. Без указания привязки объекты не будут иметь первичных ключей.
			• **pkey** – массив ключей, на основании которого будет сформирован первичный ключ каждого объекта
			• **vl** – единичное поле, которое будет отвечать за значение объекта
			• **params** – расширенные параметры объекта:
			  • **ts** – массив с настройками формата даты:
			  • **key** – ключ поля, в котором находится значение даты
			  • **mask** – маска даты
			Доступные форматы для маски
			%a Weekday as locale’s abbreviated name. Mon
			%A Weekday as locale’s full name. Monday
			%w Weekday as a decimal number, where 0 is Sunday and 6 is Saturday. 1
			%d Day of the month as a zero-padded decimal number. 30
			%-d Day of the month as a decimal number. (Platform specific) 30
			%b Month as locale’s abbreviated name. Sep
			%B Month as locale’s full name. September
			%m Month as a zero-padded decimal number. 09
			%-m Month as a decimal number. (Platform specific) 9
			%y Year without century as a zero-padded decimal number. 13
			%Y Year with century as a decimal number. 2013
			%H Hour (24-hour clock) as a zero-padded decimal number. 07
			%-H Hour (24-hour clock) as a decimal number. (Platform specific) 7
				Код привязок
				{
				"pkey": [
				"id_region",
				"pok",
				"date"
				],
				"vl": "value",
				"params": {
				"ts": [
				{
				"key": "date",
				"mask": "%Y.%m.%d",
				"frequency": "D"
				}
				]
				},
				"fkey": ["date"]
				}

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
			%I Hour (12-hour clock) as a zero-padded decimal number. 07 %-I Hour (12-hour clock) as a decimal number. (Platform specific) 7 %p Locale’s equivalent of either AM or PM. AM %M Minute as a zero-padded decimal number. 06 %-M Minute as a decimal number. (Platform specific) 6 %S Second as a zero-padded decimal number. 05 %-S Second as a decimal number. (Platform specific) 5 %f Microsecond as a decimal number, zero-padded on the left. 000000 %z UTC offset in the form +HHMM or -HHMM (empty string if the object is naive). %Z Time zone name (empty string if the object is naive). %j Day of the year as a zero-padded decimal number. 273 %-j Day of the year as a decimal number. (Platform specific) 273 %U Week number of the year (Sunday as the first day of the week) as a zero padded decimal number. All days in a new year preceding the first Sunday are considered to be in week 0. 39 %W Week number of the year (Monday as the first day of the week) as a decimal number. All days in a new year preceding the first Monday are considered to be in week 0. 39 %c Locale’s appropriate date and time representation. Mon Sep 30 07:06:05 2013 %x Locale’s appropriate date representation. 09/30/13 %X Locale’s appropriate time representation. 07:06:05 %% A literal '%' character. % • **frequency** – период • **lazy** – массив ключей, для которых необходимо произвести «ленивую загрузку» – сохранить в системную таблицу, а вместо значения прописать уникальный идентификатор поля.	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
			Не все блоки могут преобразовать «ленивую загрузку» обратно в значение. • **vl_type** – техническое поле, автоматически заполняется при наличии заполненной настройки атрибутов (см. ниже). Поле означает тип значения для отображения в таблице. На текущий момент может принимать только одно значение, равное 4 (если поле в таблицу не пришло, то все равно проставляется значение 4). Рекомендуется не использовать данное поле. • **fkey** – массив первичных ключей, на основании которых будет собран ряд. Все ключи так же необходимо указать в настройке **pkey**	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Рисунок 314 – Мастер «Атрибуты» (выбор данных)

Мастер – выбор данных

Название параметра	Тип параметра	Обязательный	Описание	Пример
Наименования поля	Строка	Нет	Текстовое описание поля	
Идентификатор	Строка	Да	Идентификатор поля, как он указан в запросе	
Тип	Выпадающий список	Да	Тип поля. Для поля с типом «дата» обязательно наличие настройки формата даты в меню «Параметры привязки»	
Входит в первичный ключ	Флаг	Нет	Входит ли поле в первичный ключ (настройка **pkey**). Обязательно указывать все поля, входящие в первичный ключ	
Ключ для формирования рядов	Флаг	Нет	Используется ли поле для формирования ряда (настройка **fkey**)	
Поле является значением	Флаг	Нет	Считать ли поле – значением для элемента ряда (настройка **vl**)	
Элемент с пустым значением	Флаг	Нет	Имеет ли элемент пустые значения	
Ленивая загрузка поля	Флаг	Нет	Используется ли поле для ленивой загрузки (настройка **lazy**)	

6.2.10.10.7.3 Мастер «SQL-запрос»

Рисунок 315 – Мастер «SQL-запрос» (ручной режим)

Мастер – ручной режим

Название параметра	Тип параметра	Обязательный	Описание	Пример
Запрос SQL	Строка (SQL запрос)	Да	SQL-запрос для извлечения данных. Допустимы только запросы, возвращающие какие-либо данные.	Пример `SELECT * FROM some_table`

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
			Возможна подстановка запроса с JINJA-шаблоном. Данные для шаблона применяются из входа «Дополнительные параметры»	
Таблица для сохранения данных	Строка	Нет	Наименование таблицы, в которую будут сохраняться данные при настроенном сохранении объектов (write-back)	

Рисунок 316 – Мастер «SQL-запрос» (выбор данных)

Мастер – выбор данных

Название параметра	Тип параметра	Обязательный	Описание	Пример
Таблица	Строка	Да	Наименование таблицы, из которой будет извлекаться информация, и в которую будут сохраняться данные при настроенном сохранении объектов (write-back)	
Столбцы для отображения	Список с обновлением	Да	Список столбцов, доступных для выбора в таблице. Формируется на основе данных, заведенных в разделе «**Атрибуты**». Для обновления информации необходимо нажать на знак со стрелочками в правой части поля для поиска	
Фильтр	Мастер	Да	Мастер для формирования запросов к таблице. Описание см. ниже	

6.2.10.10.7.4Мастер «Фильтр»

Рисунок 317 – Мастер «Фильтр» (ручной режим)

Мастер – ручной режим

Название параметра	Тип параметра	Обязательный	Описание	Пример
Текст запроса	Строка (SQL запрос)	Да	Текст запроса для выбора данных.	Пример

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
			Может включать как поля, выбранные в меню «**Столбцы для отображения**», так и другие, не указанные поля. Обратите внимание, пишутся только условия, без указания слова WHERE	`A = 10AND B = 'test'`

Рисунок 318 – Мастер «Фильтр» (выбор данных)

Мастер – выбор данных

В отличии от других мастеров, здесь еще доступен переключать И/ИЛИ, отвечающий за то, как условия запроса будут пересекаться друг с другом (AND/OR). Переключатель применяется сразу для всех условий (либо все И, либо все ИЛИ).

Внутри мастера можно добавить одно или несколько условий – используя кнопку, расположенную в правом углу:

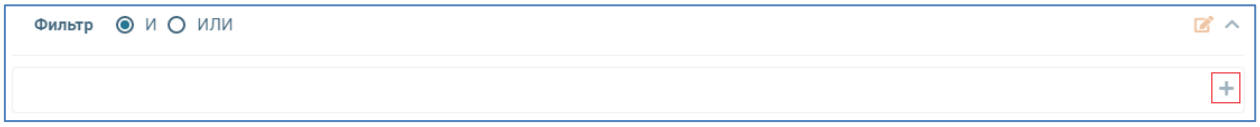


Рисунок 319 – Мастер «Фильтр»

Каждый элемент содержит в себе мастер.

6.2.10.10.7.5Мастер «Фильтр» (второй уровень)

Рисунок 320 – Мастер «Фильтр» (второй уровень)

Мастер – ручной режим

Название параметра	Тип параметра	Обязательный	Описание	Пример
Текст условия	Строка (SQL запрос)	Да	Текст одного условия для выбора данных. Может включать как поля, выбранные в меню «**Столбцы для отображения**», так и другие, не указанные поля.	Приме `B = 'HelloWord`

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Рисунок 321 – Мастер «Фильтр» (второй уровень)

Мастер – выбор данных

В данном мастере так же доступен переключатель И/ИЛИ, отвечающий за выбор агрегации условий запроса (AND/OR).

Название параметра	Тип параметра	Обязательный	Описание	Пример
Ключ	Строка	Да	Ключ поля для условия. Может включать как поля, выбранные в меню «**Столбцы для отображения**», так и другие, не указанные поля	
Условие	Выпадающий список	Да	Условие	
Значение	Строка	Да	Значение для выполнения условия	

6.2.10.10.7.6Работа блока

В начале блок собирает из условий один SQL запрос. Если в каких-то частях запроса был указан JINJA-шаблон, то на этом этапе происходит подстановка данных.

После чего происходит расчет запроса, полностью аналогичный блоку «SQL – запрос для рядов» (см. п. 6.2.10).

6.2.10.10.7.7Динамические выходы блока

Аналогично блоку «SQL – запрос для рядов» (см. п. 6.2.10).

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.2.10.10.8 Описание работы визуализатора

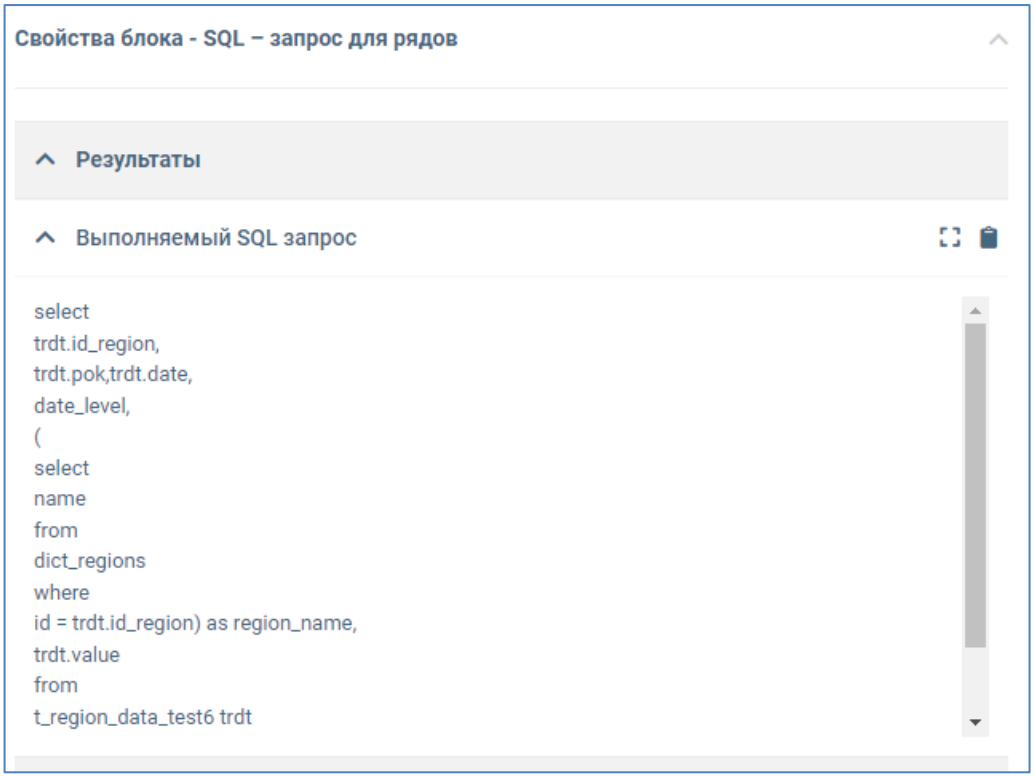


Рисунок 322 – Пример визуализатора «МастерSQL – запрос для рядов»

При использовании режима отладки (см. п. 7.5), для блока доступен системный визуализатор с полным текстом SQL запроса, с примененным JINJA-шаблоном, если такой использовался.

6.2.11 Блок «SQL – запрос-справочник»

6.2.11.1 Назначение

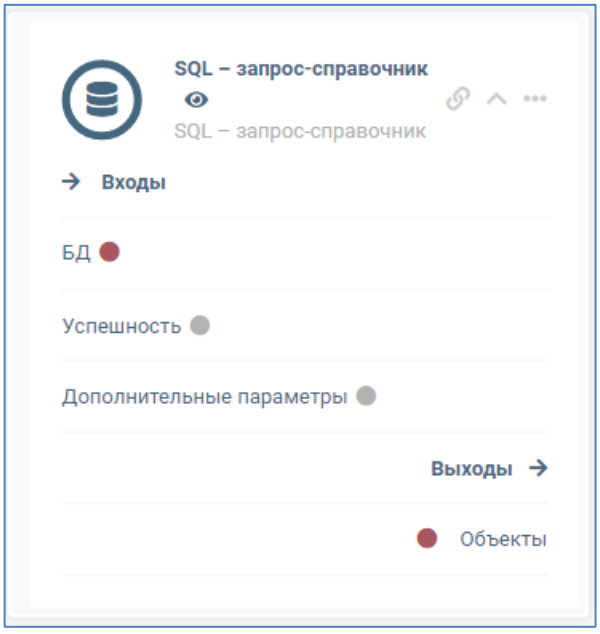


Рисунок 323 – Пример блока «SQL – запрос-справочник»

Блок позволяет получать массив объектов из БД по SQL-запросу. Так же доступен write-back атрибутов объекта.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.2.11.2 Настройки блока

Название параметра	Тип параметра	Обязательный	Описание	Пример
Группа «Основные свойства»				
Запрос SQL	Строка (SQL запрос)	Да	SQL-запрос для извлечения данных. Допустимы только запросы, возвращающие какие-либо данные. Возможна подстановка запроса с JINJA-шаблоном. Данные для шаблона применяются из входа «Дополнительные параметры»	Пример SELECT * FROM some_table

Параметры привязки	JSON	Нет	Параметры, на основе которых формируются объекты. Без указания привязки объекты не будут иметь первичных ключей.
			• **pkey** – массив ключей, на основании которого будет сформирован первичный ключ каждого объекта;
			• **vl** – единичное поле, которое будет отвечать за значение объекта;
			• **params** – расширенные параметры объекта.
			Параметр необязателен для выполнения SQL-блока, если получаемый ряд не будет далее передан в один из блоков статистических моделей или в блоки «Выражение» и «Правила».
			• **ts – массив с настройками формата даты:**
			  • **key** – ключ поля, в котором находится значение даты;
			  • **mask** – маска даты. Используется для приведения значений даты в строковый вид при сериализации в JSON.
			Доступные форматы для маски
			`%a Weekday as locale’s abbreviated name. Mon`
			`%A Weekday as locale’s full name. Monday`
			`%w Weekday as a decimal number, where 0 is Sunday and 6 is Saturday. 1`
			`%d Day of the month as a zero-padded decimal number. 30`
				Код привязок
				`{`
				` "pkey": [`
				`  "id",`
				`"region_id"`
				` ],`
				` "vl": "value:,`
				` "params": {`
				`  "ts": [`
				`  {`
				`  "key":`
				`"date",`
				`  "mask":`
				`"%Y.%m.%d",`
				`"frequency":`
				`"D"`
				`  }`
				`  ],`
				`  "lazy":`
				`["lazy_vl_1"]`
				` }`
				`}`

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
			%-d Day of the month as a decimal number. (Platform specific) 30 %b Month as locale’s abbreviated name. Sep %B Month as locale’s full name. September %m Month as a zero-padded decimal number. 09 %-m Month as a decimal number. (Platform specific) 9 %y Year without century as a zero-padded decimal number. 13 %Y Year with century as a decimal number. 2013 %H Hour (24-hour clock) as a zero-padded decimal number. 07 %-H Hour (24-hour clock) as a decimal number. (Platform specific) 7 %I Hour (12-hour clock) as a zero-padded decimal number. 07 %-I Hour (12-hour clock) as a decimal number. (Platform specific) 7 %p Locale’s equivalent of either AM or PM. AM %M Minute as a zero-padded decimal number. 06 %-M Minute as a decimal number. (Platform specific) 6 %S Second as a zero-padded decimal number. 05 %-S Second as a decimal number. (Platform specific) 5 %f Microsecond as a decimal number, zero-padded on the left. 000000 %z UTC offset in the form +HHMM or -HHMM (empty string if the the object is naive). %Z Time zone name (empty string if the object is naive). %j Day of the year as a zero-padded decimal number. 273 %-j Day of the year as a decimal number. (Platform specific) 273 %U Week number of the year (Sunday as the first day of the week) as a zero padded decimal number. All days in a new year preceding the first Sunday are considered to be in week 0. 39	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
			%W Week number of the year (Monday as the first day of the week) as a decimal number. All days in a new year preceding the first Monday are considered to be in week 0. 39 %c Locale’s appropriate date and time representation. Mon Sep 30 07:06:05 2013 %x Locale’s appropriate date representation. 09/30/13 %X Locale’s appropriate time representation. 07:06:05 %% A literal '%' character. % • **frequency** – период; • **lazy** – массив ключей, для которых необходимо произвести «ленивую загрузку» – сохранить в системную таблицу, а вместо значения прописать уникальный идентификатор поля. Не все блоки могут преобразовать «ленивую загрузку» обратно в значение. • **vl_type** – техническое поле, автоматически заполняется при наличии заполненной настройки атрибутов (см. ниже). Поле означает тип значения для отображения в таблице. Возможные значения vl_type 2 – строка 4 – вещественный	
Запрос обновления данных	Строка	Нет	Устаревшее поле, использовалось в блоках до версии 1. На текущий момент функционал отключен. Шаблон запроса для обновления атрибутов объектов. Пример шаблона обновления данных INSERT INTO <имя таблицы> ({0}) VALUES ({1}) ON CONFLICT ({2}) DO UPDATE SET {3} = EXCLUDED.{3} **Значения в фигурных скобках необходимо оставить в неизменном виде.** Требуется только указать имя таблицы	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
Запрос удаления данных	Строка	Нет	Устаревшее поле, использовалось в блоках до версии 1. На текущий момент функционал отключен Шаблон запроса для удаления атрибутов объектов Пример шаблона обновления данных `DELETE FROM <имя таблицы> WHERE {0}` **Значения в фигурных скобках необходимо оставить в неизменном виде.** Требуется только указать имя таблицы	
Поля, доступные для изменения	Строка	Нет	Устаревшее поле, использовалось в блоках до версии 1. На текущий момент функционал отключен. Список ключей атрибутов (без пробелов, через запятую), которые можно изменять во время write-back. Для этих полей будут применены запросы на обновление/удаление, указанные выше	
Наименование таблицы для сохранения данных	Строка	Нет	Наименование таблицы, в которую будут сохраняться данные при настроенном сохранении объектов (write-back)	
Раздел настроек «Настройка атрибутов"				
Наименования поля	Строка	Нет	Текстовое описание поля	
Идентификатор	Строка	Да	Идентификатор поля, как он указан в запросе	
Тип	Выпадающий список	Да	Тип поля. Для поля с типом «дата» обязательно наличие настройки формата даты в меню «Параметры привязки"	
Поле является наименованием	Флаг	Нет	Считать ли данное поле наименованием данных справочника.	
Элемент с пустым значением	Флаг	Нет	Имеет ли элемент пустые значения	
Уникальный ключ для иерархии	Флаг	Нет	Является ли поле ключом, по которому будет отстраиваться иерархический справочник. Допустимо наличие только одного поля с включенным флагом	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
Родитель	Флаг	Нет	Является ли поле родительским элементом, по которому будет отстраиваться иерархический справочник Допустимо наличие только одного поля с включенным флагом.	
Ленивая загрузка поля	Флаг	Нет	Использовать ли «ленивую загрузку» для данного поля. Аналогично настройке **lazy** из настройки «Параметры привязки»	
Входит в первичный ключ	Флаг	Нет	Входит ли поле в первичный ключ (настройка **pkey** из меню «Параметры привязки»). Обязательно указывать все поля, входящие в первичный ключ	
Write back	Флаг	Нет	Возможность изменения значения поля при использовании механизма write-back (изменение и сохранение данных)	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.2.11.3 Группа «Общие настройки блока»

Аналогично приведенным в п. 6.1.1.1.3.

6.2.11.4 Входы

Название входа	Тип	Обязательный	Описание
БД	Соединение с БД	Да	Соединение с базой данных, из которой необходимо извлекать данные. Поддерживаются разные диалекты баз, а именно: • PostgreSQL • MSSQL • Oracle • ClickHouse
Успешность	Флаг	Нет	Вход для запуска расчета блока в случае если нет входов (на него можно подать выход Успешность другого блока)
Дополнительные параметры	Ряд данных	Нет	Массив объектов или временных рядов, атрибуты которых могут быть использованы для параметризации запроса

6.2.11.5 Выходы

Название выхода	Тип	Описание

Объекты	Объект	Массив объектов с возможностью добавления динамических выходов (описано ниже в разделе «Динамические выходы блока»)
		Формат объекта
		{
		«pkey": [
		<Набор первичных ключей>
		],
		«vl": <Значение объекта>,
		«attrs": {
		<Набор атрибутов – полей, не попавших ни в первичный ключ, ни в значение>
		},
		«_params": {
		<Опциональные параметры объекта>,
		«vl": <Ключ поля, отвечающего за значение>
		},
		«_source": <Идентификатор блока-источника, поле заполняется калькулятором автоматически>,
		}

6.2.11.6 События блока

В блоке нет событий.

6.2.11.7 Описание работы блока

Блок выполняет SQL-запрос (при необходимости – формируя его из JINJA-шаблона) из выбранной базы данных, после чего преобразует каждую полученную строку в объект по шаблону, заданному в поле «Параметры привязки».

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Результат работы блока – массив объектов.

В том случае, если заданы поля для сохранения данных (write-back), то при изменении данных в визуализаторе, стоящим после данного блока, происходит формирование запроса на добавление и/или обновление данных с последующей записей в выбранную таблицу (поле «Наименование таблицы для сохранения данных»).

Для разных баз данных должен применяться свой диалект SQL.

6.2.11.8 Динамические выходы блока

Динамические выходы – дочерний выход блока, отфильтрованный по выбранным первичным ключам объекта.

Для настройки необходимо навести курсор на выход «**Объекты**» и нажать на выпадающий элемент меню.

Настройку динамических выходов необходимо производить на уже рассчитанном блоке.

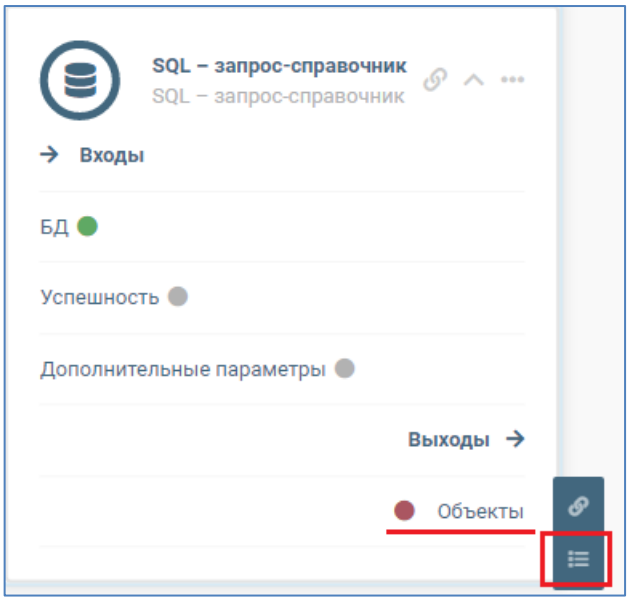


Рисунок 324 – Пример блока «SQL – запрос-справочник» (динамические выходы)

После нажатия открывается окно с выбором переменных.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

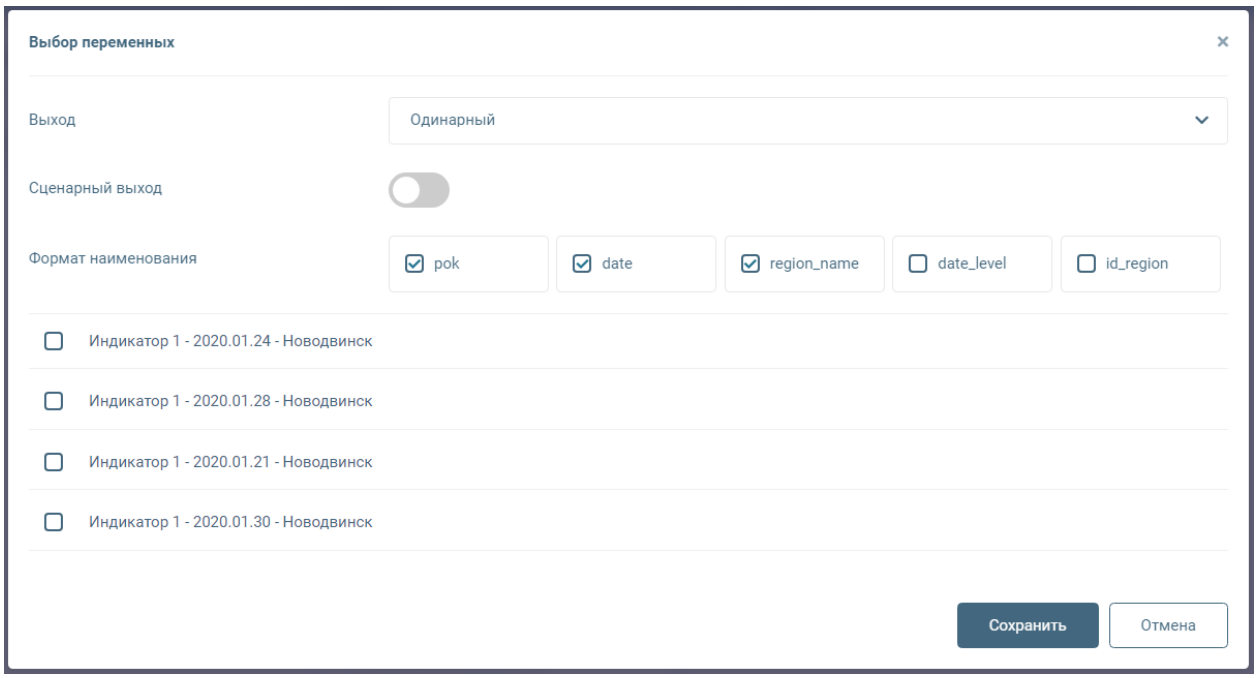


Рисунок 325 – Окно «Выбор переменных»

- **Выход** – формат выхода данных. Выпадающее меню с двумя значениями:
  - **Одинарный** – каждый выбранный элемент создаст отдельный выход блока;
  - **Массив** – все выбранные элементы объединяются в один массив. Для массива так же можно задать наименование выхода.
- **Формат наименования** – выбор первичных ключей и/или атрибутов объекта, которые будут отвечать за формирование наименования выхода.
  - Путем перетаскивания ключей (*drag-n-drop)* можно изменять порядок элементов.

После выбора одной или нескольких переменных и нажатия на кнопку «Сохранить», у выбранного блока создадутся дополнительные выходы.

За одно редактирование переменных возможно создание нескольких одинарных выходов или одного массива.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

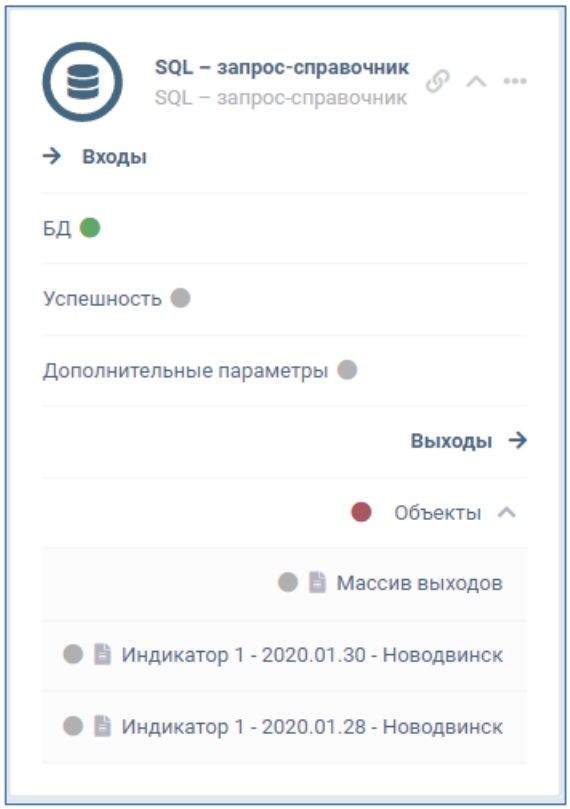


Рисунок 326 – Пример блока «SQL – запрос-справочник»

*Блок с тремя добавленными выходами: одним **массивом** и двумя **одинарными** выходами.*

После создания выхода, доступно так же его **редактирование** (иконка с карандашом) или **удаление** (иконка с мусорной корзиной):

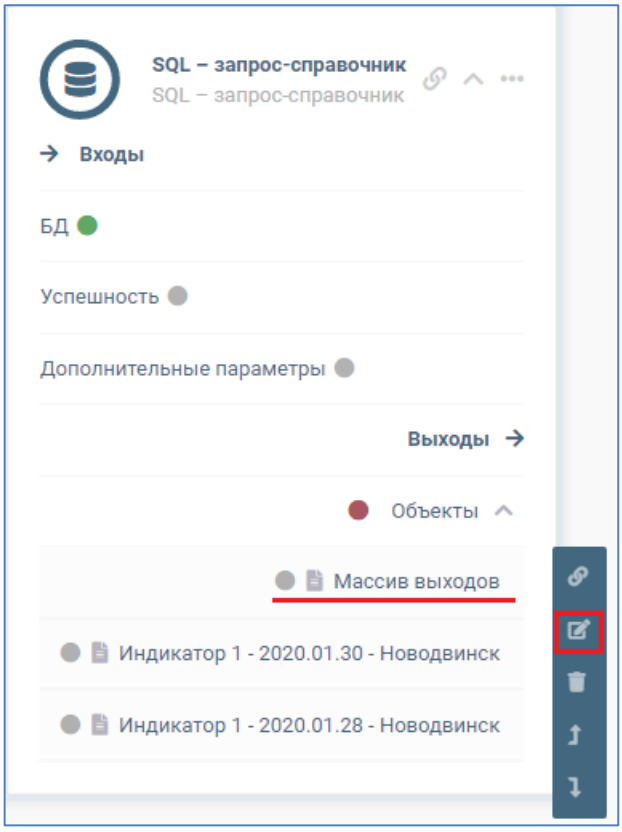


Рисунок 327 – Пример блока «SQL – запрос-справочник»

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

При выборе редактирования выхода открывается меню «**Выбор переменных**», в котором можно изменить параметры выхода.

6.2.11.9 Описание работы визуализатора

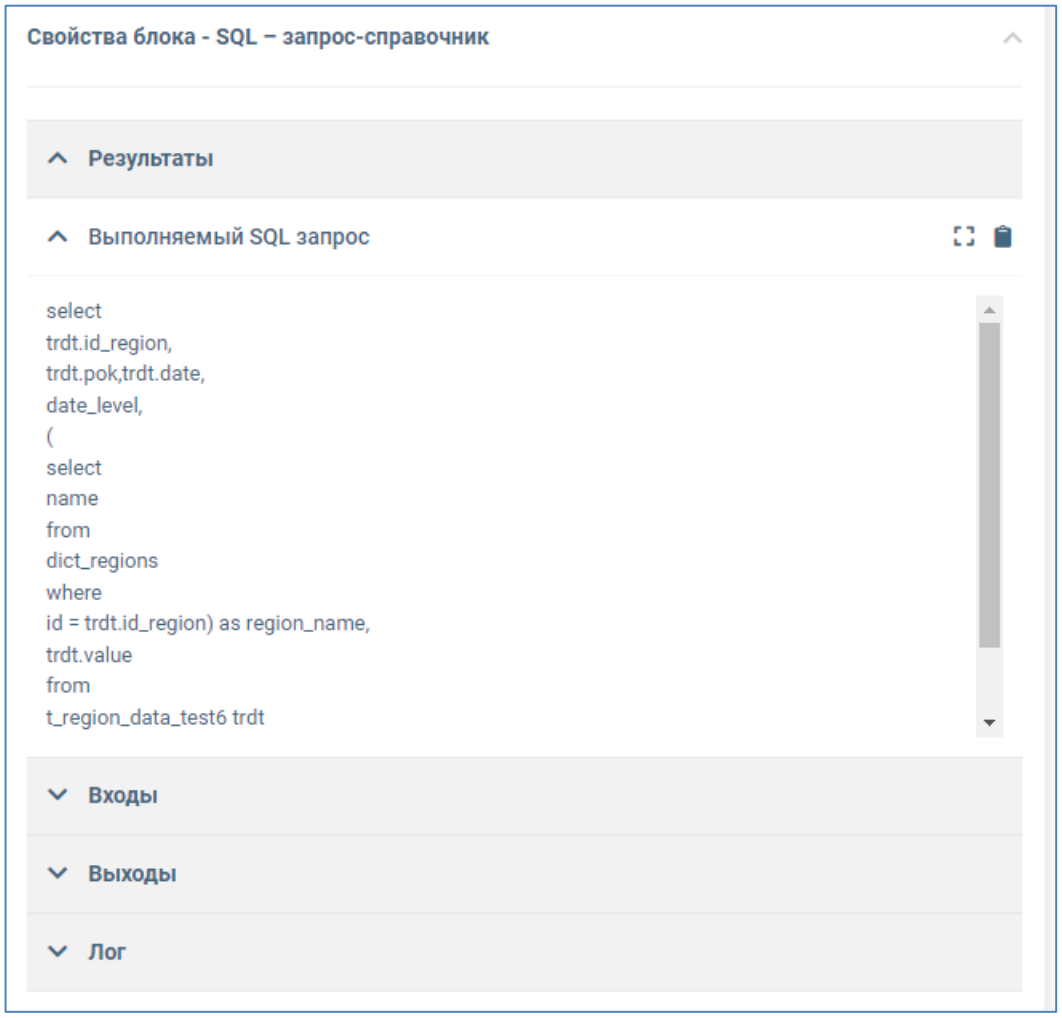


Рисунок 328 – Свойства блока «SQL – запрос-справочник»

При использовании режима отладки (см. п. 7.5), для блока доступен системный визуализатор с полным текстом SQL запроса, с примененным JINJA-шаблоном, если такой использовался.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.2.11.10 Блок «Мастер SQL – справочник»

6.2.11.10.1 Назначение

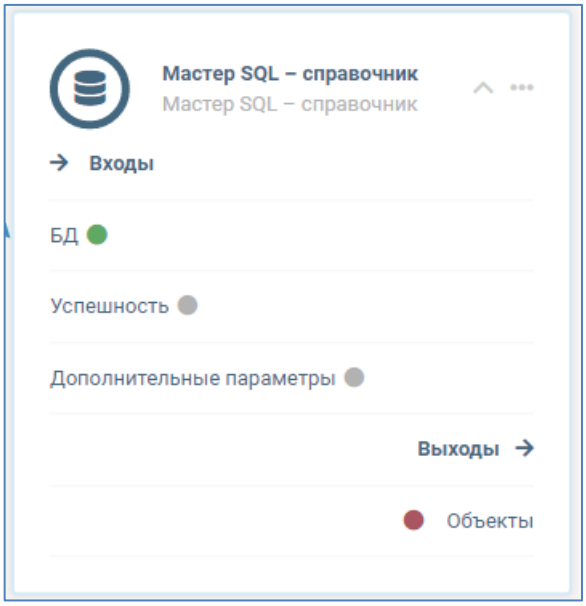


Рисунок 329 – Пример блока «Мастер SQL – справочник»

Блок позволяет получать массив объектов из БД по SQL-запросу. В отличии от блока «SQL – запрос-справочник» (см. п. 6.2.11), формирование запроса реализовано с помощью мастера.

Так же доступен write-back для значений ряда.

6.2.11.10.2 Настройки блока

Название параметра	Тип параметра	Обязательный	Описание	Пример
Атрибуты	Мастер	Да	Мастер заполнения атрибутов блока. Подробнее см. раздел Описание работы блока – Мастер «Атрибуты»	
SQL-запрос	Мастер	Да	Мастер по формированию запроса. Подробнее см. раздел Описание работы блока – Мастер «SQL-запрос»	

6.2.11.10.3 Группа «Общие настройки блока»

Аналогично приведенным в п. 6.1.1.1.3.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.2.11.10.4 Входы

Название входа	Тип	Обязательный	Описание
БД	Соединение с БД	Да	Соединение с базой данных, из которой необходимо извлекать данные. Поддерживаются разные диалекты баз, а именно: • PostgreSQL • MSSQL • Oracle • ClickHouse
Успешность	Флаг	Нет	Вход для запуска расчета блока в случае если нет входов (на него можно подать выход Успешность другого блока).
Дополнительные параметры	JSON	Нет	Массив объектов или временных рядов, атрибуты которых могут быть использованы для параметризации запроса.

6.2.11.10.5 Выходы

Название выхода	Тип	Описание

Объекты	Объект	Массив объектов с возможностью добавления динамических выходов (описано ниже в разделе «**Динамические выходы блока**»)
		Формат объекта
		{
		"pkey": [
		<Набор первичных ключей>
		],
		"vl": <Значение объекта>,
		"attrs": {
		<Набор атрибутов - полей, не попавших ни в первичный ключ, ни в значение>
		},
		"_params": {
		<Опциональные параметры объекта>,
		"vl": <Ключ поля, отвечающего за значение>
		},
		"_source": <Идентификатор блока-источника, поле заполняется калькулятором автоматически>,
		}

6.2.11.10.6 События блока

В блоке нет событий.

6.2.11.10.7 Описание работы блока

6.2.11.10.7.1 Работа с мастером

Для формирования запроса и выбора необходимых полей, требуется заполнить два мастера: **Атрибуты** и **SQL-запрос**.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Для каждого мастера доступно два режима работы: **ручной режим** (чаще всего – строковое поле для ввода) и **выбор данных**. Переключение режимов работы мастеров осуществляется нажатием на кнопку .

Обратите внимание, данные заведенные в разных режимах не пересекаются.

6.2.11.10.7.1.1 Мастер «Атрибуты»

Рисунок 330 – Мастер «Атрибуты» (ручной режим)

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Мастер – ручной режим

Название параметра	Тип параметра	Обязательный	Описание	Пример

Параметры привязки	JSON	Нет	Поле, совмещающее в себе поля «Параметры привязки» блока «SQL – запрос-справочник» (см. п. 6.2.11).
			Параметры, на основе которых формируются объекты. Без указания привязки объекты не будут иметь первичных ключей.
			• **pkey** – массив ключей, на основании которого будет сформирован первичный ключ каждого объекта;
			• **vl** – единичное поле, которое будет отвечать за значение объекта;
			• **params** – расширенные параметры объекта:
			  • **ts** – массив с настройками формата даты:
			  • **key** – ключ поля, в котором находится значение даты
			  • **mask** – маска даты.
			Доступные форматы для маски
			%a Weekday as locale’s abbreviated name. Mon
			%A Weekday as locale’s full name. Monday
			%w Weekday as a decimal number, where 0 is Sunday and 6 is Saturday. 1
			%d Day of the month as a zero-padded decimal number. 30
			%-d Day of the month as a decimal number. (Platform specific) 30
			%b Month as locale’s abbreviated name. Sep
			%B Month as locale’s full name. September
			%m Month as a zero-padded decimal number. 09
			%-m Month as a decimal number. (Platform specific) 9
			%y Year without century as a zero-padded decimal number. 13
			%Y Year with century as a decimal number. 2013
			%H Hour (24-hour clock) as a zero-padded decimal number. 07
			%-H Hour (24-hour clock) as a decimal number. (Platform specific) 7
			%I Hour (12-hour clock) as a zero-padded decimal number. 07
			%-I Hour (12-hour clock) as a decimal number. (Platform specific) 7
			%p Locale’s equivalent of either AM or PM. AM
			%M Minute as a zero-padded decimal number. 06
			%-M Minute as a decimal number. (Platform specific) 6
				Код привязок
				{
				"pkey": [
				"id_region",
				"pok",
				"date"
				],
				"vl": "value",
				"params": {
				"ts": [
				{
				"key": "date",
				"mask": "%Y.%m.%d",
				"frequency": "D"
				}
				]
				}
				}

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
			%S Second as a zero-padded decimal number. 05 %-S Second as a decimal number. (Platform specific) 5 %f Microsecond as a decimal number, zero-padded on the left. 000000 %z UTC offset in the form +HHMM or -HHMM (empty string if the the object is naive). %Z Time zone name (empty string if the object is naive). %j Day of the year as a zero-padded decimal number. 273 %-j Day of the year as a decimal number. (Platform specific) 273 %U Week number of the year (Sunday as the first day of the week) as a zero padded decimal number. All days in a new year preceding the first Sunday are considered to be in week 0. 39 %W Week number of the year (Monday as the first day of the week) as a decimal number. All days in a new year preceding the first Monday are considered to be in week 0. 39 %c Locale’s appropriate date and time representation. Mon Sep 30 07:06:05 2013 %x Locale’s appropriate date representation. 09/30/13 %X Locale’s appropriate time representation. 07:06:05 %% A literal '%' character. % • **frequency** – период; • **lazy** – массив ключей, для которых необходимо произвести «ленивую загрузку» – сохранить в системную таблицу, а вместо значения прописать уникальный идентификатор поля. Не все блоки могут преобразовать «ленивую загрузку» обратно в значение. • **vl_type** – техническое поле, автоматически заполняется при наличии заполненной настройки атрибутов (см. ниже). Поле означает тип значения для отображения в таблице. На текущий момент может принимать только одно значение, равное 4 (если поле в таблицу не пришло, то все равно проставляется значение 4). Рекомендуется не использовать данное поле.	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Рисунок 331 – Мастер «Атрибуты» (выбор данных)

Мастер – выбор данных

Название параметра	Тип параметра	Обязательный	Описание	Пример
Наименования поля	Строка	Нет	Текстовое описание поля	
Идентификатор	Строка	Да	Идентификатор поля, как он указан в запросе	
Тип	Выпадающий список	Да	Тип поля. Для поля с типом «дата» обязательно наличие настройки формата даты в меню «Параметры привязки»	
Поле является наименованием	Флаг	Нет	Считать ли данное поле наименованием данных справочника	
Элемент с пустым значением	Флаг	Нет	Имеет ли элемент пустые значения	
Уникальный ключ для иерархии	Флаг	Нет	Является ли поле ключом, по которому будет отстраиваться иерархический справочник. Допустимо наличие только одного поля с включенным флагом	Уникальный ключ для иерархии
Родитель	Флаг	Нет	Является ли поле родительским элементом, по которому будет отстраиваться иерархический справочник. Допустимо наличие только одного поля с включенным флагом	Родитель
Ленивая загрузка поля	Флаг	Нет	Используется ли поле для ленивой загрузки (настройка **lazy**)	Ленивая загрузка поля
Входит в первичный ключ	Флаг	Нет	Входит ли поле в первичный ключ (настройка **pkey**). Обязательно указывать все поля, входящие в первичный ключ	Входит в первичный ключ
Write back	Флаг	Нет	Возможность изменения значения поля при использовании механизма write-back (изменение и сохранение данных)	Write back

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.2.11.10.7.1.2 Мастер «SQL-запрос»

Рисунок 332 – Мастер «SQL-запрос» (ручной режим)

Мастер – ручной режим

Название параметра	Тип параметра	Обязательный	Описание	Пример
Запрос SQL	Строка (SQL запрос)	Да	SQL-запрос для извлечения данных. Допустимы только запросы, возвращающие какие-либо данные. Возможна подстановка запроса с JINJA-шаблоном. Данные для шаблона применяются из входа «Дополнительные параметры»	Пример `SELECT * FROM some_table`
Таблица для сохранения данных	Строка	Нет	Наименование таблицы, в которую будут сохраняться данные при настроенном сохранении объектов (write-back)	

Рисунок 333 – Мастер «SQL-запрос» (выбор данных)

Мастер – выбор данных

Название параметра	Тип параметра	Обязательный	Описание	Пример
Таблица	Строка	Да	Наименование таблицы, из которой будет извлекаться информация, и в которую будут сохраняться данные при настроенном сохранении объектов (write-back)	
Столбцы для отображения	Список с обновлением	Да	Список столбцов, доступных для выбора в таблице. Формируется на основе данных, заведенных в разделе «**Атрибуты**». Для обновления информации необходимо нажать на знак со стрелочками в правой части поля для поиска	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
Фильтр	Мастер	Да	Мастер для формирования запросов к таблице. Описание см. ниже	

6.2.11.10.7.1.3 Мастер «Фильтр»

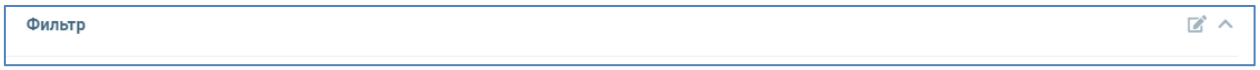


Рисунок 334 – Мастер «Фильтр» (ручной режим)

Мастер – ручной режим

Название параметра	Тип параметра	Обязательный	Описание	Пример
Текст запроса	Строка (SQL запрос)	Да	Текст запроса для выбора данных. Может включать как поля, выбранные в меню «**Столбцы для отображения**», так и другие, не указанные поля. Обратите внимание, пишутся только условия, без указания слова WHERE	Пример `A = 10AND B = 'test'`

Рисунок 335 – Мастер «Фильтр» (выбор данных)

Мастер – выбор данных

В отличии от других мастеров, здесь еще доступен переключать И/ИЛИ, отвечающий за то, как условия запроса будут пересекаться друг с другом (AND/OR). Переключатель применяется сразу для всех условий (либо все И, либо все ИЛИ).

Внутри мастера можно добавить одно или несколько условий – используя кнопку, расположенную в правом углу:

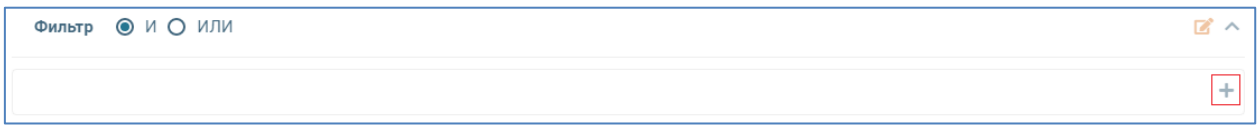


Рисунок 336 – Кнопка для добавления условий в мастере «Фильтр»

Каждый элемент содержит в себе мастер.

6.2.11.10.7.1.4 Мастер «Фильтр» (второй уровень)

Рисунок 337 – Мастер «Фильтр» (второй уровень) (ручной режим)

Мастер – ручной режим

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
Текст условия	Строка (SQL запрос)	Да	Текст одного условия для выбора данных. Может включать как поля, выбранные в меню «**Столбцы для отображения**», так и другие, не указанные поля	Пример `B = 'HelloWord`

Рисунок 338 – Мастер «Фильтр» (второй уровень) (выбор данных)

Мастер – выбор данных

В данном мастере так же доступен переключатель И/ИЛИ, отвечающий за выбор агрегации условий запроса (AND/OR).

Название параметра	Тип параметра	Обязательный	Описание	Пример
Ключ	Строка	Да	Ключ поля для условия. Может включать как поля, выбранные в меню «**Столбцы для отображения»**, так и другие, не указанные поля	
Условие	Выпадающий список	Да	Условие	
Значение	Строка	Да	Значение для выполнения условия	

6.2.11.10.7.2 Работа блока

В начале блок собирает из условий один SQL запрос. Если в каких-то частях запроса был указан JINJA-шаблон, то на этом этапе происходит подстановка данных.

После чего происходит расчет запроса, полностью аналогичный блоку «SQL – запрос-справочник» (п. 6.2.11.10).

6.2.11.10.7.3 Динамические выходы блока

Аналогично блоку «SQL – запрос-справочник» (см. п. 6.2.11).

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.2.11.10.8 Описание работы визуализатора

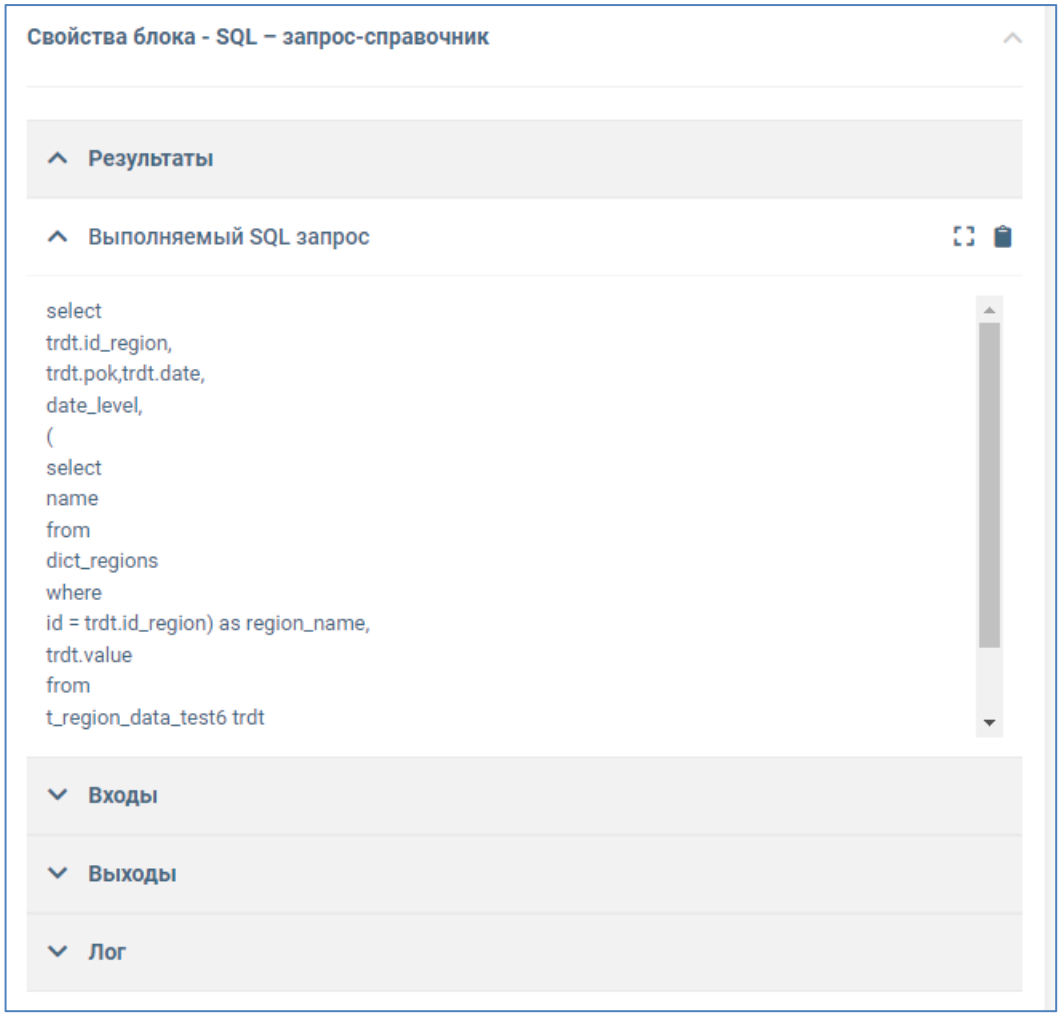


Рисунок 339 – Внешний вид визуализатора «Мастер SQL – справочник»

При использовании режима отладки (см. п. 7.5), для блока доступен системный визуализатор с полным текстом SQL запроса, с примененным JINJA-шаблоном, если такой использовался.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.2.12 Блок «SAP – запрос»

6.2.12.1 Назначение

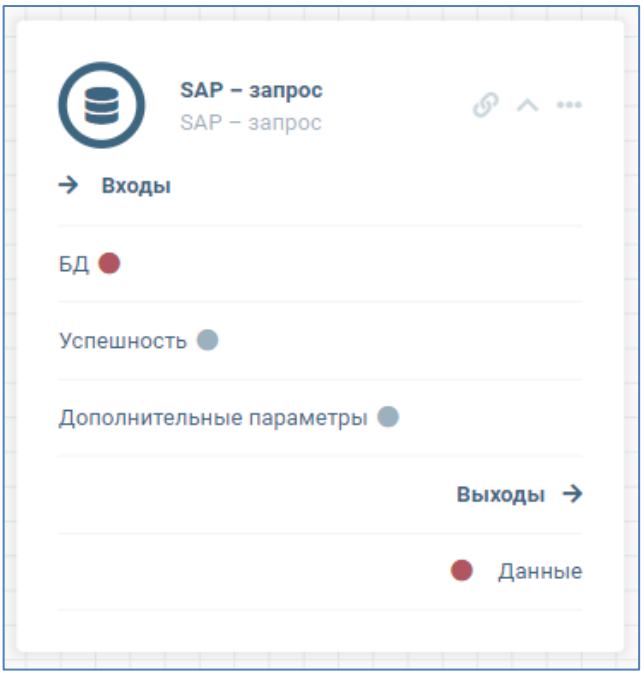


Рисунок 340 – Мастер «SAP – запрос»

Блок позволяет получать данные из SAP сервиса.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.2.12.2 Настройки блока

Название параметра	Тип параметра	Обязательный	Описание	Пример
Группа «Основные свойства»				
SAP ссылка	Строка	Да	Ссылка для подключения к SAP	http://sapserver/sap/bw/xml/soap/xmla/Execute

Запрос SAP	Строка	Да	Текст SAP запроса для получения данных. Может содержать jinja шаблон для подстановки данных со входа «Дополнительные параметры»	**Пример запроса:**
				`select`
				`  non empty {`
				`crossjoin(crossjoin(crossjoin(crossjoin([IND].[LEVEL01].MEMBERS,[IND].[LEVEL01].MEMBERS),[IND].[LEVEL01].MEMBERS),[IND].[LEVEL01].MEMBERS),[IND].[LEVEL01].MEMBERS) } on columns,`
				`  non empty { [Measures].[IND] } on rows`
				`from [$IND]`
				`where (`
				`  {{ inputs[1]['attrs']['member_unique_name'] }},`
				`  [IND].[01],`
				`  {{ inputs[0]['pkey']['member_unique_name'] }},`
				`  [IND].[#],`
				`  [IND].[RAS],`
				`  [IND].[ZPL_FR002]`
				`)`

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.2.12.3 Группа «Общие настройки блока»

Аналогично приведенным в п. 6.1.1.1.3.

6.2.12.4 Входы

Название входа	Тип	Обязательный	Описание
БД	Соединение с БД	Да	Соединение с базой данных MongoDB
Успешность	Флаг	Нет	Вход для запуска расчета блока в случае если нет входов (на него можно подать выход Успешность другого блока)
Дополнительные параметры	Ряд данных	Нет	Массив объектов или временных рядов, атрибуты которых могут быть использованы для параметризации запроса

6.2.12.5 Выходы

Название выхода	Тип	Описание
Данные	XML строка	XML ответ сервера SAP

6.2.12.6 События блока

В блоке нет событий.

6.2.12.7 Описание работы блока

Блок выполняет введенный пользователем запрос к SAP серверу, после чего возвращает на выходе полученный от сервиса ответ в виде XML структуры.

6.2.13 Блок «Генерация временного ряда»

6.2.13.1 Назначение

Блок позволяет сгенерировать временной ряд нужной длины со случайным значением каждой точки.

6.2.13.2 Настройки блока

Название параметра	Тип параметра	Описание	Пример
Название блока	Строка	Наименование блока	
Начальная дата	Строка	Начальная дата временного ряда в формате ДД.ММ.ГГГГ	20.05.2020
Динамика	Строка	Динамика временного ряда, может быть годовая (Y), месячная (M) или дневная (D)	1D
Количество точек	Целое число	Количество точек в ряду	
Минимальное значение	Целое число	Минимальное значение точки	
Максимальное значение	Целое число	Максимальное значение точки	
Ссылка на документацию	Строка		

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Описание	Пример
Описание блока	Строка		
Пропуск расчета блока	Флажок	Рассчитывать или нет данный блок	

6.2.13.3 Входы

У блока отсутствуют входы.

6.2.13.4 Выходы

- **Временной ряд**

Массив рядов, состоящий из одного временного ряда стандартизированного формата.

Выход из блока

```
[
 {
  "pkey": {},
  "vl": [
   <Массив сгенерированных значений ряда>
  ],
  "attrs": {},
  "keys": [
   <Даты ряда, начиная с даты, указанной в поле «Начальная дата»>
  ]
  "_params": {
   "ts": [
    "key": "dt",
    "mask": "%d.%m.%Y",
    "frequency": <Период, указанный в поле «Динамика»>
   ],
   "vl": "value",
   "key": ["dt"]
  },
  "_source": "",
 }
]
```

6.2.14 Блок «Расширение атрибутов временного ряда»

6.2.14.1 Назначение

Блок позволяет добавить дополнительные атрибуты в ряд и поддерживает их write-back.

6.2.14.2 Настройки блока

Название блока	Строка	Наименование блока	Пример
Идентификатор таблицы	Строка	Наименование таблицы, в которой находятся дополнительные атрибуты	
Первичный ключ таблицы	Строка	Перечисление всех полей в таблице, относящихся к первичным ключам, через запятую, без пробелов. Ключи должны входить в pkey	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название блока	Строка	Наименование блока	Пример
		ряда (за исключением элемента, который определяет привязку для рядов (поле fkey). В ряду может быть больше ключей, чем требуется для дополнительных атрибутов. В таком случае нужно указывать только те ключи, которые есть в таблице	

Параметры привязки атрибутов	JSON	Массив элементов определенного формата:
		Json биндинг атрибутов ряда
		[
		{
		"name": <Ключ дополнительного атрибута>,
		"type": <Тип атрибута, по аналогии с типами атрибутов таблиц>,
		"value": <Значение по умолчанию (в том случае, если в таблице нет записи по этому атрибуту)>
		"items": <Только для атрибута типа «Список». Ключ поля в котором хранится массив с элементами выпадающего списка. Формат массива: [{"id": <ключ>, "name": "<наименование>"}]>
		}
		]
			`[ {"name": "new_attr_str", "type":"string", "value": "qqq"}, {"name": "new_attr_bool", "type": "boolean", "value": 0}, {"name": "new_attr_list", "type": "list", "value": 1, "items": "attr_items"} ]`

Ссылка на документацию	Строка		
Описание блока	Строка		
Пропуск расчета блока	Флажок	Рассчитывать или нет данный блок	

6.2.14.3 Входы

- **БД**

Соединение с базой данных, из которой необходимо извлекать данные.

- **Данные**

Массив рядов, к которым необходимо добавить атрибуты. (Весь ряд должен иметь ключи, удовлетворяющие полю «Первичный ключ таблицы». Атрибуты добавятся к каждому из элементов массива)

- **Успешность**

Вход для запуска расчета блока в случае если нет входов (на него можно подать выход Успешность другого блока)

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.2.14.4 Выходы

- **Данные**

Выход из блока – массив рядов стандартизированного формата. Формат рядов аналогичен формату выхода «Ряды данных» из блока «Конвертор объектов в ряды» (п. 6.5.2).

6.2.15 Блок «Сохранение данных в БД»

6.2.15.1 Назначение

Блок позволяет сохранять ряды данных в отдельную таблицу.

Таблица для сохранения должна быть создана заранее. Необходимо, чтобы в таблице были столбцы для всех первичных ключей (**pkey**) и ключ, по которому строится ряд (**fkey**). Поле, в котором находится значение необходимо назвать **vl**.

6.2.15.2 Настройки блока

Название параметра	Тип параметра	Описание
Название блока	Строка	Наименование блока
Идентификатор таблицы	Строка	Ключ таблицы, в которую необходимо сохранять данные
Ссылка на документацию	Строка	
Описание блока	Строка	
Пропуск расчета блока	Флажок	Рассчитывать или нет данный блок

6.2.15.3 Входы

- **БД**

Соединение с базой данных, из которой необходимо извлекать данные.

- **Данные**

Массив рядов, к которым необходимо добавить атрибуты. (Весь ряд должен иметь ключи, удовлетворяющие полю «Первичный ключ таблицы». Атрибуты добавятся к каждому из элементов массива).

6.2.15.4 Выходы

- **Успешность записи**

Флаг, обозначающий успешность сохранения данных.

6.2.16 Блок «Справочник»

6.2.16.1 Назначение

Блок создает таблицу в базе данных с выбранными полями и позволяет получать массив объектов (запрос генерируется автоматически). Так же доступен write-back полей.

Эта страница посвящена описанию настроек блока, для описаний мастера блока см. «Мастер настройки справочника» (п. 7.8.7) и «Мастер данных блока» (п. 7.8.8).

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.2.16.2 Настройки блока

Название параметра	Тип параметра	Обязательный	Описание	Пример
Группа «Основные свойства»				
Привязка дополнительных параметров	JSON	Нет	Привязки формируют раздел условий при расчете блока. Необходимы для параметризации запроса на извлечение данных из справочника. Описание привязок [{"<Идентификатор поля в таблице>": «{{ <Jinja шаблон с выбранным полем> }}"}, <Единичный элемент> {"<Идентификатор поля в таблице>": «{{ <Jinja шаблон с выбранным массивом полей>|join(', ') }}» <Массив>] Для массива требуется, чтобы в параметре была структура со значениями через запятую (пример: 0,1,2,3). Используемый в примере вариант: https://jinja.palletsprojects.com/en/2.11.x/templates/#join где: **Пример макроса** <массив с данными>|join(', ', attribute='<атрибуты элементов массива, которые необходимо добавить в массив, опционально>')	**Пример макроса** [{"id": «{{ input[0].pkey.id }}"}, {"region_id": «{{ input|join(', ', attribute='pkey.region_id') }}"]

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.2.16.3 Группа «Общие настройки блока»

Аналогично приведенным в п. 6.1.1.1.3.

6.2.16.4 Входы

Название входа	Тип	Обязательный	Описание
БД	Соединение с БД	Да	Соединение с базой данных, из которой необходимо извлекать данные. Поддерживаются разные диалекты баз, а именно: • PostgreSQL • MSSQL • Oracle • ClickHouse не поддерживается данным типом блока
Дополнительные параметры	Ряд данных	Нет	Массив объектов или временных рядов, атрибуты которых могут быть использованы для параметризации запроса по извлечению данных (при расчете блока).

6.2.16.5 Выходы

Название выхода	Тип	Описание

Объекты	Объект	Массив объектов с возможностью добавления динамических выходов (описано ниже в разделе «**Динамические выходы блока**»).
		Формат объекта
		{
		«pkey": [
		<Набор первичных ключей>
		],
		«vl": <Значение объекта>,
		«attrs": {
		<Набор атрибутов – полей, не попавших ни в первичный ключ, ни в значение>
		},
		«_params": {
		<Опциональные параметры объекта>,
		«vl": <Ключ поля, отвечающего за значение>
		},
		«_source": <Идентификатор блока-источника, поле заполняется калькулятором автоматически>,
		}

6.2.16.6 События блока

В блоке нет событий

6.2.16.7 Описание работы блока

Блок формирует, а затем выполняет SQL-запрос (при необходимости – добавляя в него JINJA-шаблон из настройки «Привязка дополнительных параметров») из указанной таблице в базе данных, после чего преобразует каждую полученную строку в объект, используя в качестве шаблона описание полей, заданных в мастере.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Результат работы блока – массив объектов.

В том случае, если для одного или нескольких атрибутов включена настройка сохранения данных данных (write-back), то при изменении данных в визуализаторе, стоящим после данного блока, происходит формирование запроса на добавление и/или обновление данных с последующим сохранением в таблицу.

6.2.16.7.1 Динамические выходы блока

Динамические выходы – дочерний выход блока, отфильтрованный по выбранным первичным ключам объекта.

Для настройки необходимо навести курсор на выход «**Объекты**» и нажать на выпадающий элемент меню:

Настройку динамических выходов необходимо производить на уже рассчитанном блоке.

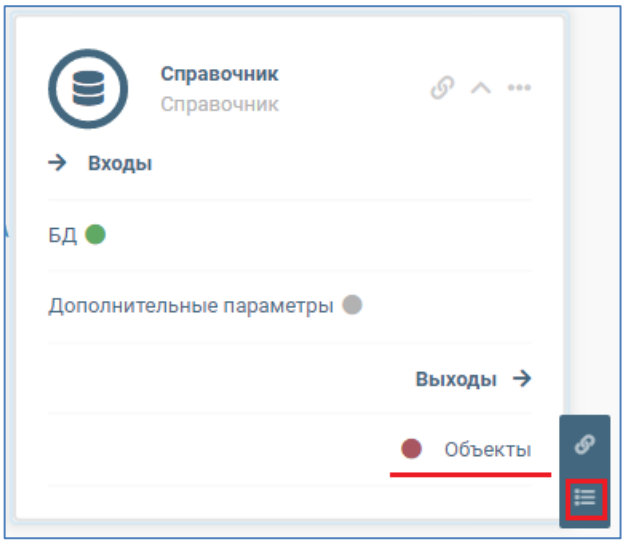


Рисунок 341 – Блок «Справочник» (динамические выходы)

После нажатия открывается окно с выбором переменных.

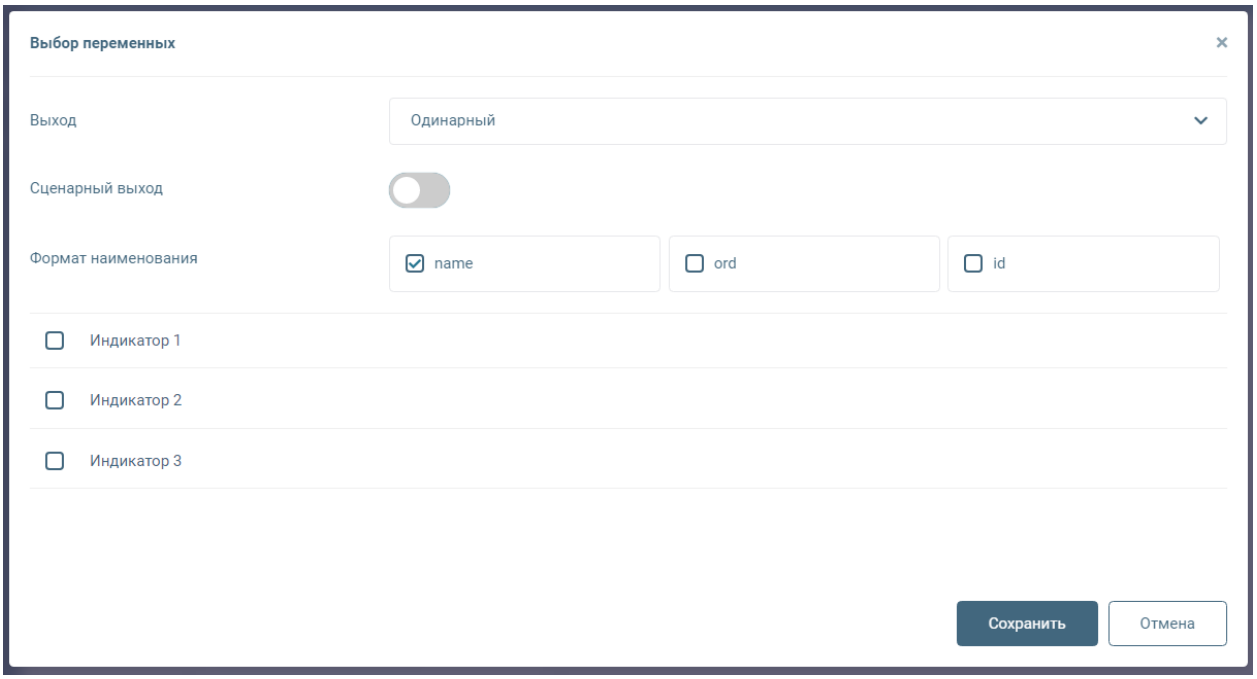


Рисунок 342 – Окно «Выбор переменных»

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

- **Выход** – формат выхода данных. Выпадающее меню с двумя значениями:
  - **Одинарный** – каждый выбранный элемент создаст отдельный выход блока;
  - **Массив –** все выбранные элементы объединятся в один массив. Для массива так же можно задать наименование выхода.
- **Формат наименования** – выбор первичных ключей и/или атрибутов объекта, которые будут отвечать за формирование наименования выхода.
  - Путем перетаскивания ключей (*drag-n-drop)* можно изменять порядок элементов.

После выбора одной или нескольких переменных и нажатия на кнопку «Сохранить», у выбранного блока создадутся дополнительные выходы.

За одно редактирование переменных возможно создание нескольких одинарных выходов или одного массива.

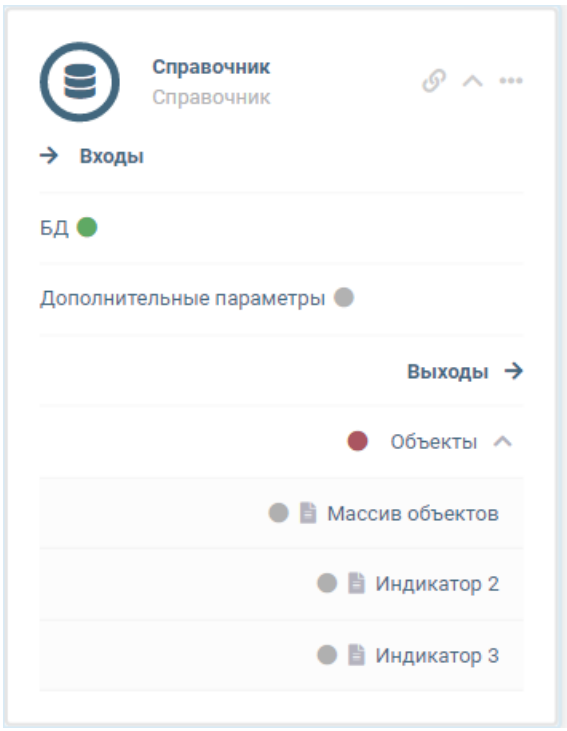


Рисунок 343 – Блок «Справочник» с тремя добавленными выходами: одним **массивом** и двумя **одинарными** выходами

После создания выхода, доступно так же его **редактирование** (иконка с карандашом) или **удаление** (иконка с мусорной корзиной).

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

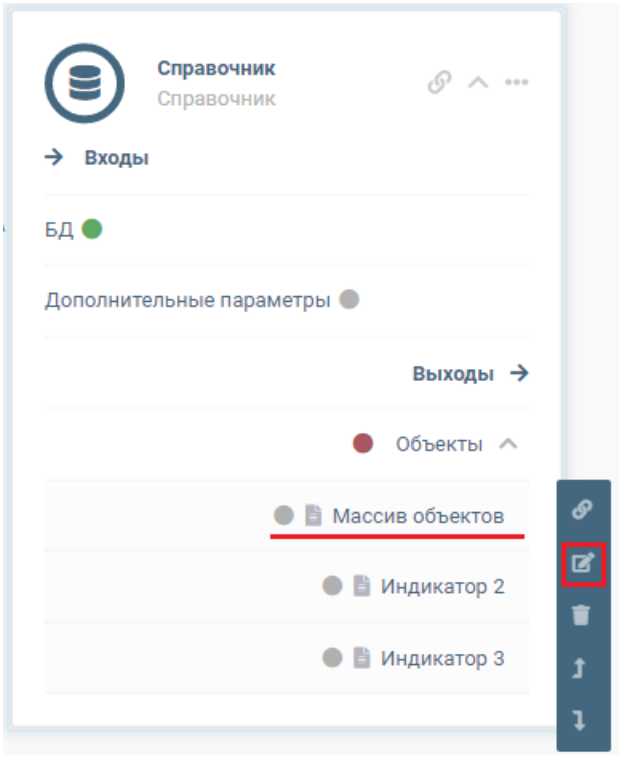


Рисунок 344 – Блок «Справочник» (кнопка «Редактировать»)

При выборе редактирования выхода открывается меню «**Выбор переменных»**, в котором можно изменить параметры выхода.

6.2.17Блок «Показатели»

6.2.17.1 Назначение

Блок создает таблицу в базе данных с выбранными полями и позволяет получать массив рядов (запрос и привязки генерируется автоматически). Так же доступен write-back поля со значением.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.2.17.2 Настройки блока

Название параметра	Тип параметра	Обязательный	Описание	Пример
Название блока	Строка	Нет	Наименование блока	

Привязка дополнительных параметров	JSON	Нет	Привязки формируют раздел условий при расчете блока. Необходимы для параметризации запроса на извлечение данных из справочника.
			Описание привязок
			`[`
			`{"<Идентификатор поля в таблице>": «{{ <Jinja шаблон с выбранным полем> }}"}, <Единичный элемент>`
			`{"<Идентификатор поля в таблице>": «{{ <Jinja шаблон с выбранным массивом полей>|join(', ') }}" <Массив>`
			`]`
			Для массива требуется, чтобы в параметре была структура со значениями через запятую (пример: 0,1,2,3).
			Используемый в примере вариант: https://jinja.palletsprojects.com/en/2.11.x/templates/#join где:
			Пример макроса
			<массив с данными>|join(', ', attribute='<атрибуты элементов массива, которые необходимо добавить в массив, опционально>')
				Пример макроса
				`[`
				`{"id": «{{ input[0].pkey.id }}"},`
				`{"region_id": «{{ input|join(', ', attribute='pkey.region_id') }}"`
				`]`

Ссылка на документацию	Строка	Нет		
Описание блока	Строка	Нет		
Пропуск расчета блока	Флажок	Нет	Рассчитывать или нет данный блок	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.2.17.3 Входы

- **БД**

Соединение с базой данных, из в которой хранится таблица и данные справочника.

- **Дополнительные параметры**

Массив объектов или временных рядов, атрибуты которых могут быть использованы для параметризации запроса. Не обязательный вход.

6.2.17.4 Выходы

- **Данные**

Выход из блока – массив рядов стандартизированного формата. Формат рядов аналогичен формату выхода «Ряды данных» из блока «Конвертор объектов в ряды" (п. 6.5.2).

6.2.18 Блок «JSON-Справочник»

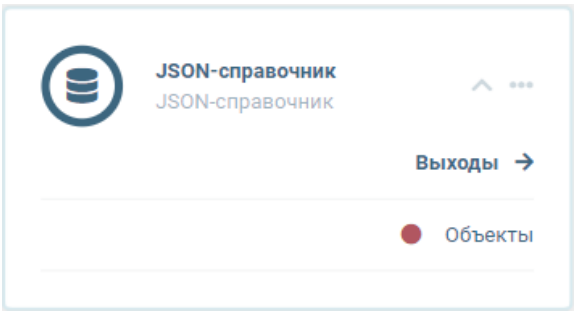


Рисунок 345 – Пример блока «JSON-Справочник»

6.2.18.1 Назначение

Блок позволяет создавать, изменять, удалять, получать объекты и их атрибуты

Использовать для прототипирования и отладки. Не использовать в реальных прикладных проектах, т.к. данные хранятся в системном репозитории

6.2.18.2 Настройки блока

В блоке нет специфических настроек.

6.2.18.3 Группа «Общие настройки блока»

Аналогично приведенным в п. 6.1.1.1.3.

6.2.18.4 Входы

В блоке нет входов.

6.2.18.5 Выходы

Название выхода	Тип	Описание

Объекты	Объект	Массив объектов с возможностью добавления динамических выходов (описано ниже в разделе «Динамические выходы блока»).
		Формат объекта
		`[`
		`{`
		`«pkey": {`
		`«id": 1`
		`},`
		`«attrs": {`

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название выхода	Тип	Описание
		«ord": 1, «name": «1" }, «_params": {}, «_source": «063e1ce9-bb60-4e15-a196-e42906970def" }]

6.2.18.6 Динамические выходы блока

Динамические выходы – дочерний выход блока, отфильтрованный по выбранным первичным ключам объекта.

Для настройки необходимо навести курсор на выход «**Объекты**» и нажать на выпадающий элемент меню.

Настройку динамических выходов необходимо производить на уже рассчитанном блоке.

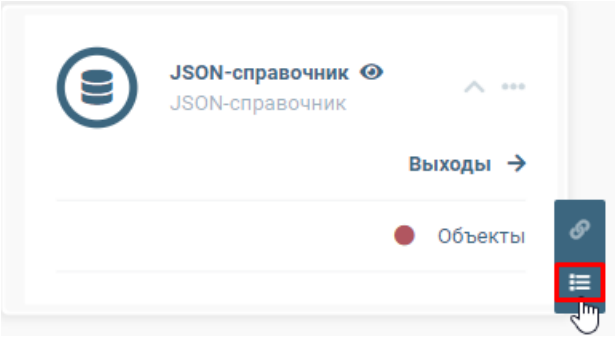


Рисунок 346 – Пример динамических выходов

6.2.18.7 События блока

В блоке нет событий.

6.2.18.8 Описание работы блока

Работа с объектами и их атрибутами осуществляется в рамках мастера настройки JSON справочника (см. п. 7.8.1).

6.2.19Блок «JSON-Показатели»

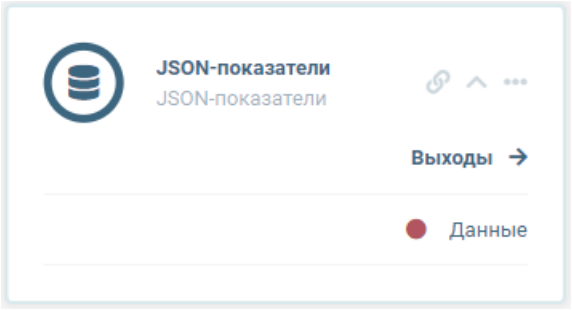


Рисунок 347 – Пример блока «JSON-Показатели»

6.2.19.1 Назначение

Блок позволяет создавать, изменять, удалять, получать ряды данных

Использовать для прототипирования и отладки. Не использовать в реальных прикладных проектах, т.к. данные хранятся в системном репозитории.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.2.19.2 Настройки блока

В блоке нет специфических настроек.

6.2.19.3 Группа «Общие настройки блока»

Аналогично приведенным в п. 6.1.1.1.3.

6.2.19.4 Входы

В блоке нет входов.

6.2.19.5 Выходы

Название выхода	Тип	Описание

Данные	Ряд данных	Массив объектов с возможностью добавления динамических выходов (описано ниже в разделе «Динамические выходы блока»).
		Формат объекта
		[
		{
		"pkey": {
		"date_level": 1
		},
		"keys": [
		"01.01.2020",
		"01.01.2021",
		"01.01.2022",
		"01.01.2023",
		"01.01.2024"
		],
		"vl": [
		777,
		151,
		1,
		5484,
		55
		],
		"attrs": {
		"region_name": "Пермь"
		},
		"_source": "ccd3f418-50a3-4e71-9e67-d04403a7c1c7",
		"_params": {
		"ts": [
		{
		"key": "date",
		"mask": "%d.%m.%Y",
		"frequency": "D"
		}
		"vl": "value",
		"vl_type": 4,
		"lazy": [],
		"fkey": [
		"date"
		]
		}

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название выхода	Тип	Описание
		}, { "pkey": { "date_level": 2 }, "keys": ["01.01.2022"], "vl": [1], "attrs": { "region_name": "Пермь" }, "_source": "ccd3f418-50a3-4e71-9e67-d04403a7c1c7", "_params": { «ts": [{ "key": "date", "mask": «%d.%m.%Y", "frequency": "D" }], "vl": "value", "vl_type": 4, "lazy": [], "fkey": ["date"] } }]

6.2.19.6 Динамические выходы блока

Динамические выходы – дочерний выход блока, отфильтрованный по выбранным первичным ключам объекта.

Для настройки необходимо навести курсор на выход «**Данные**» и нажать на выпадающий элемент меню.

Настройку динамических выходов необходимо производить на уже рассчитанном блоке.

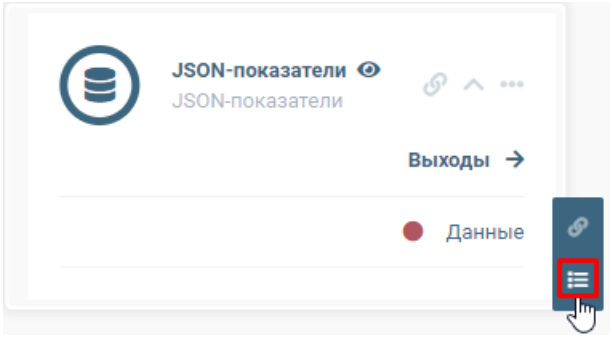


Рисунок 348 – Пример динамических выходов

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.2.19.7 События блока

В блоке нет событий.

6.2.19.8 Описание работы блока

Работа с рядами данных осуществляется в рамках мастера настройки JSON показателей (см. п. 7.8.5).

6.2.20 Блок «Импорт файла»

6.2.20.1 Назначение

Блок позволяет загружать файлы формата CSV в базу данных.

6.2.20.2 Настройки блока

Название параметра	Тип параметра	Описание	Пример
CSV файл	Строка	Ссылка на файл	
Кодировка	Выпадающий список	Варианты допустимых кодировок файла	
Кросс-таблица	Флажок		
Разделитель	Строка	Формат разделителя в CSV файле	
Маска	Маска	Маска формата даты	DD.MM.YYYY
Динамика	Строка	Динамика даты	
Начальный ряд	Целое число	Начальный ряд данных в файле	
Начальный столбец	Целое число	Начальный столбец данных в файле	
Наименование таблицы	Строка	Наименование таблицы, в которую будут сохраняться данные	
Показатели	Строка	Наименования поля в таблице, в котором будут сохранены показатели	
Столбец	Целое число	Номер колонки с показателями	
Столбец	Целое число	Номер колонки с регионами	
Первичный ключ	Флажок	Создать или нет первичный ключ для таблицы. В текущей реализации всегда должно быть включено!	
Очищать приемник перед загрузкой	Флажок	Очистить или нет таблицу от старых данных перед загрузкой файла	
Источник данных	Строка	Описание данных, которые будут загружены	
Отображать мастер в приложении	Флажок	При установке флажка – мастер импорта этого блока будет доступен и в приложении	
Название блока	Строка	Наименование блока	
Ссылка на документацию	Строка		
Описание блока	Строка		

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Описание	Пример
Пропуск расчета блока	Флажок	Рассчитывать или нет данный блок	
Отображать процесс выполнения	Флажок	Выводить или нет лог по данному блоку	

6.2.20.3 Входы

- **БД**

Соединение с базой данных, в которую необходимо сохранять данные.

6.2.20.4 Выходы

- **Справочник**

Массив объектов, который будет выполнять роль справочника по регионам.

Элементы справочника обязательно должен содержать поле «**id**» – идентификатор, либо в первичных ключах (**pkey**), либо в атрибутах (**attrs**). Кроме того, в атрибутах (**attrs**) справочника должно находится поле «**name**» – наименование элементов.

Пример справочника:

```
[
 {
  "pkey": {
   "id": 1
  },
  "attrs": {
   "name": "Регион 1"
  }
 },
 {
  "pkey": {
   "id": 2
  },
  "attrs": {
   "name": "Регион 2"
  }
 },
 {
  "pkey": {
   "id": 3
  },
  "attrs": {
   "name": "Регион 3"
  }
 },
]
```

6.2.20.5 Руководство «Мастер импорта данных»

6.2.20.5.1 Назначение

Импорт данных предназначен для загрузки данных из файла формата CSV в базу данных для дальнейшего использования средствами платформы ЦП УвП.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Успешным результатом выполнения мастера импорта данных является:

- создание таблицы в БД с наименованием указанным при выполнении импорта;
- наличие у созданной таблицы первичного ключа (в случае если установлен признак «Создать первичный ключ таблицы»), состоящего из следующего набора полей: pok, region_id, data, date_level, datasource.
- наличие загруженных данных в таблице;
- создание в графе блока «SQL – запрос-справочник» (в случае если установлен признак «Создать блок SQL запроса»).

6.2.20.5.2 Выбор приемника

Для успешного использования мастер импорта данных необходимо чтобы на графе располагались блоки:

- «Соединение с БД» (см. п. 6.2.1.9)**;**
- «Sql – запрос-справочник» (см. п. 6.2.11) – отвечающий за данные справочника территорий**.**

6.2.20.5.3 Создание нового импорта данных

Запустите мастер импорта следующим образом:

- выполните команду «Мастер настройки импорта»;
- на шаге 1 мастера загрузки данных выберите «Создать новый объект данных», нажмите «Далее»;
- на шаге 2 мастера загрузки данных:
    1. выберите путь до файла с данными в формате CSV по кнопке ;
    2. укажите параметр «Кросс-таблица» и выберите из выпадающего списка «Разделитель»;
    3. нажмите кнопку «Загрузить» или в области «Предпросмотр данных» в случае если не осуществляется загрузка файла на сервер нажмите кнопку «Обновить» ;
    4. на основании области «Предпросмотра данных» укажите маску, динамику календаря в которой представлены данные, используя пронумерованные колонки в области «Предпросмотра данных» укажите параметры «Начальная точка с данными кросс-таблицы»;
    5. занесите сведения об источнике предоставляемых данных в специальное поле «Источник данных»;
    6. нажмите «Далее».
- на шаге 3 мастера загрузки данных необходимо:
    1. Занести наименование таблицы которая будет создана в БД, указать признак для вновь создаваемой таблицы «Создать первичный ключ таблицы». Создание первичного ключа будет гарантировать уникальность каждой записи с данными в таблице.
    2. Указать привязку к столбцу раздела «Предпросмотр данных», в которой располагаются названия показателей, указать наименование поля в таблице и номер столбца.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

3. Указать привязку к столбцу файла в котором располагается наименование территории и выбрать из выпадающего справочника блок «SQL – запрос-справочник», отвечающий за справочник территорий в котором у элементов обязательно присутствуют атрибуты «id» и «name».
4. Выбрать из выпадающего списка существующее «Соединение с БД».
5. Установить отметку напротив полей мастера «Очищать приемник перед загрузкой», «Загрузить данные по окончании работы мастера», «Создать блок SQL запроса».
6. Нажать «Завершить».

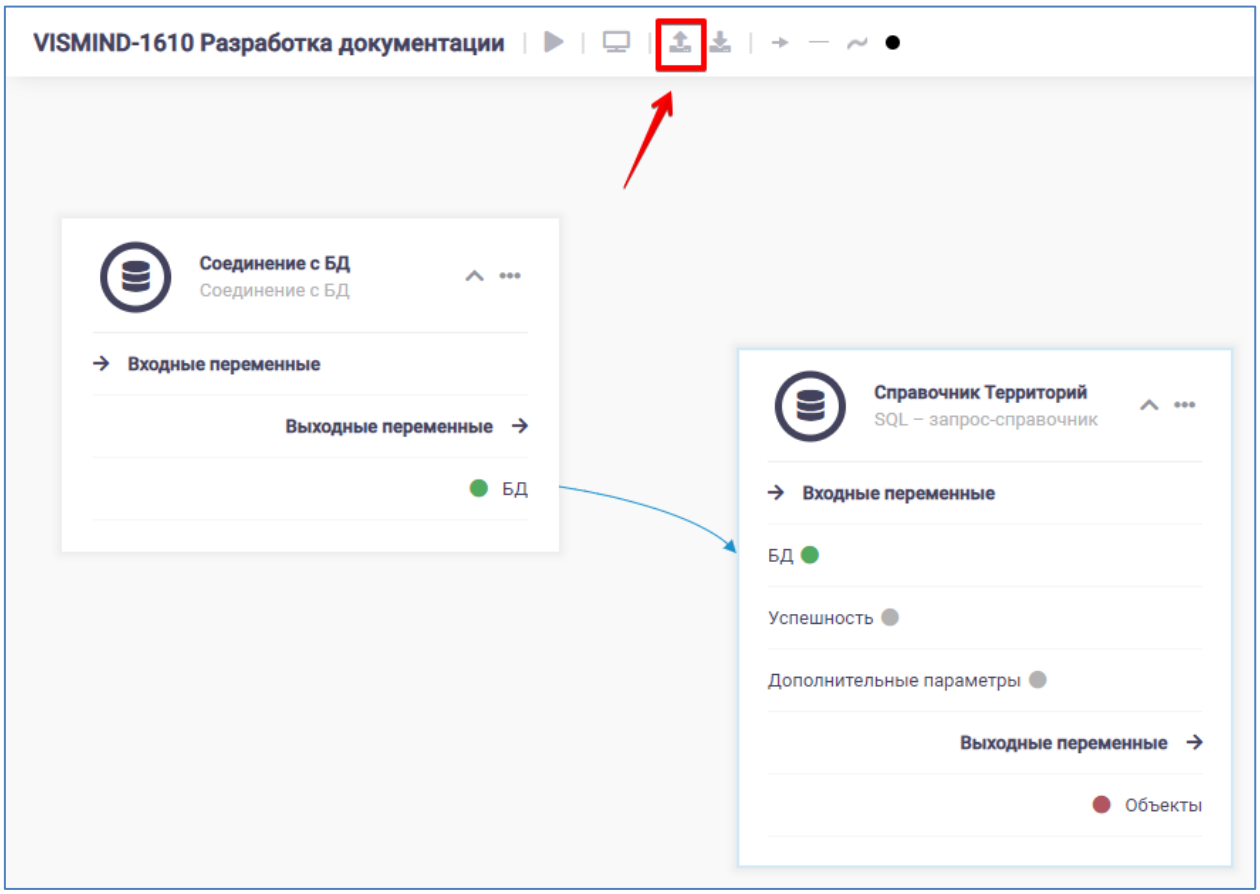


Рисунок 349 – Пример блока «Импорт данных»

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.2.20.6 Мастер настройки импорта

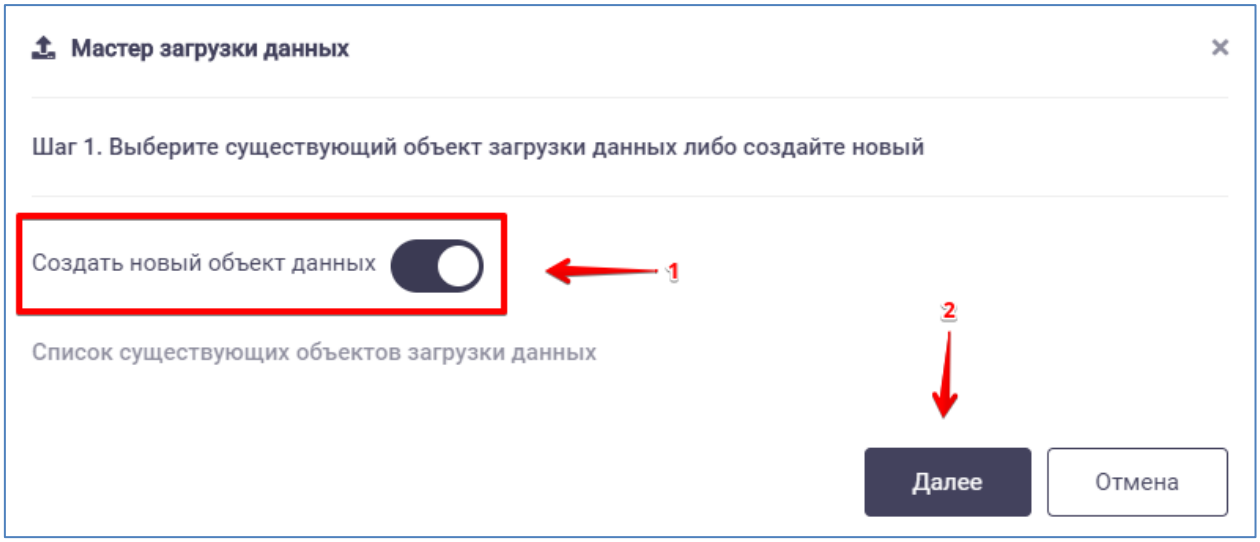


Рисунок 350 – Мастер загрузки данных (шаг 1)

1. Создание нового объекта данных.

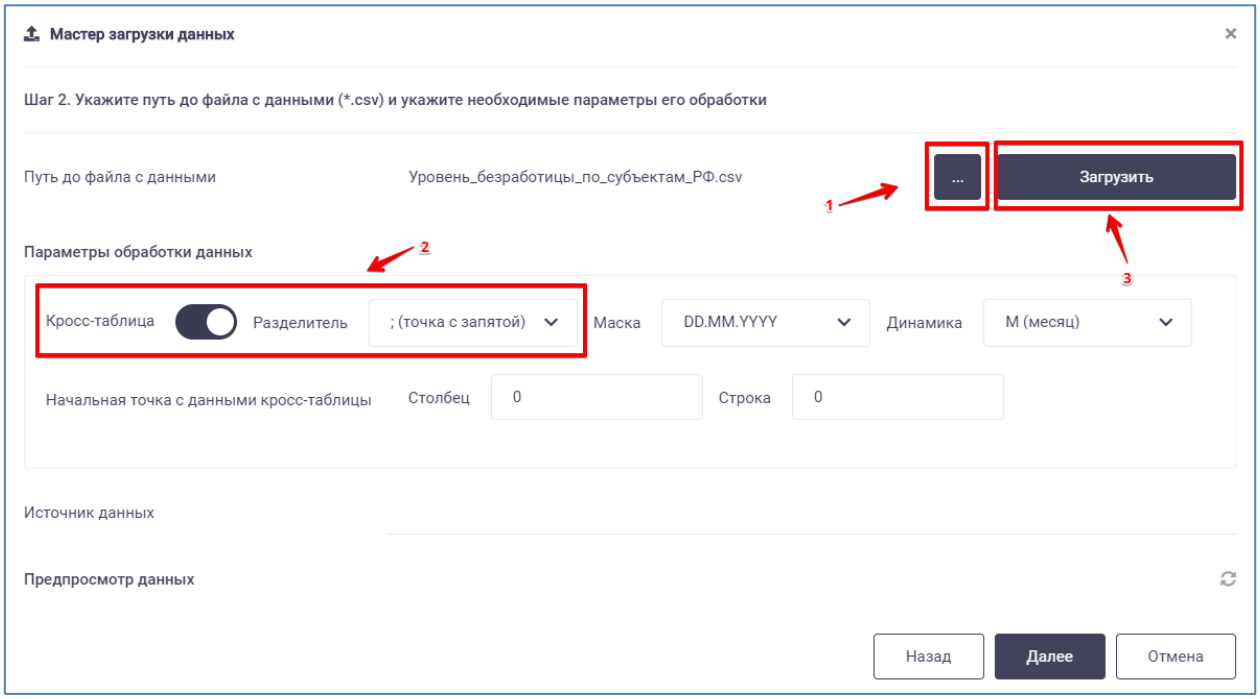


Рисунок 351 – Мастер загрузки данных (шаг 2)

2. Загрузка файла на сервер для дальнейшей обработки.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

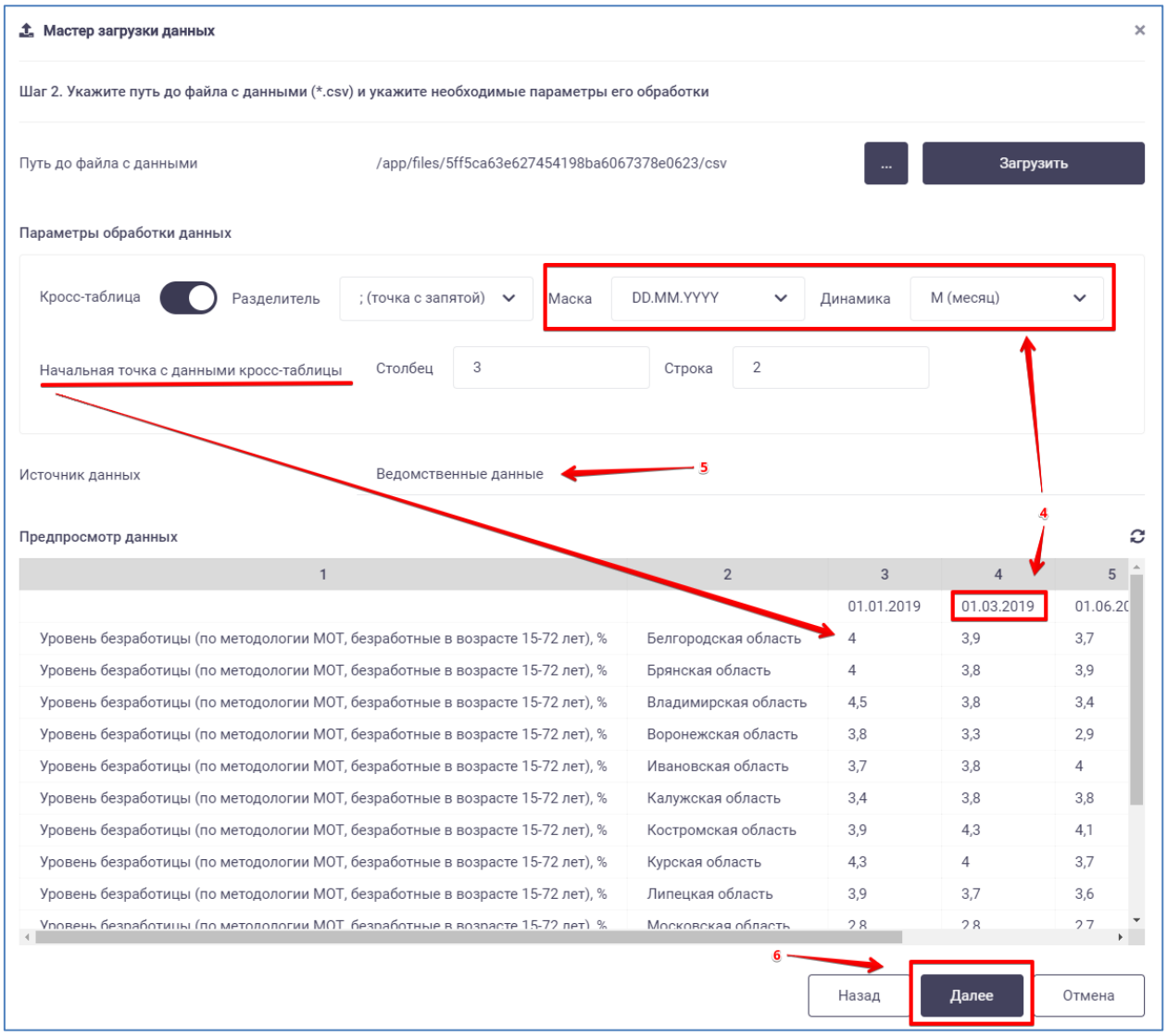


Рисунок 352 – Мастер загрузки данных (шаг 2)

3. Предпросмотр данных.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

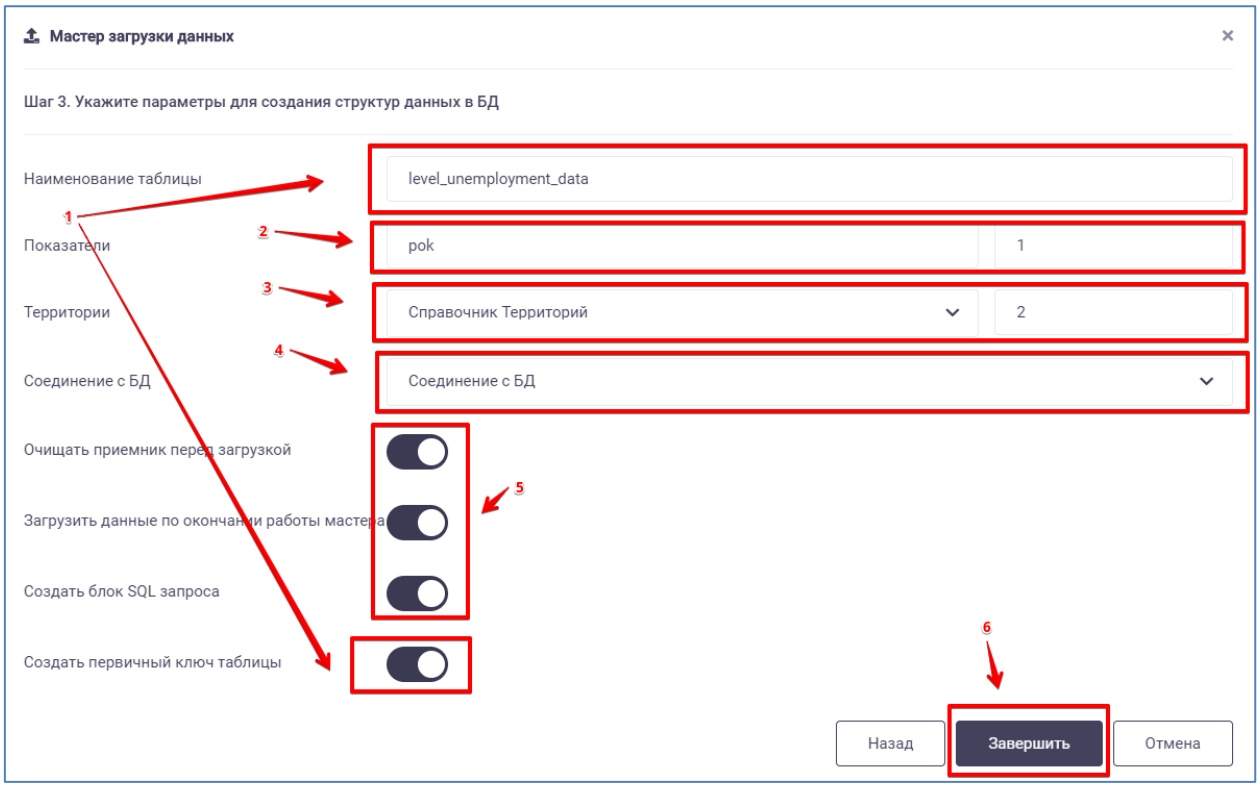


Рисунок 353 – Мастер загрузки данных (шаг 3)

4. Завершение мастера загрузки данных.

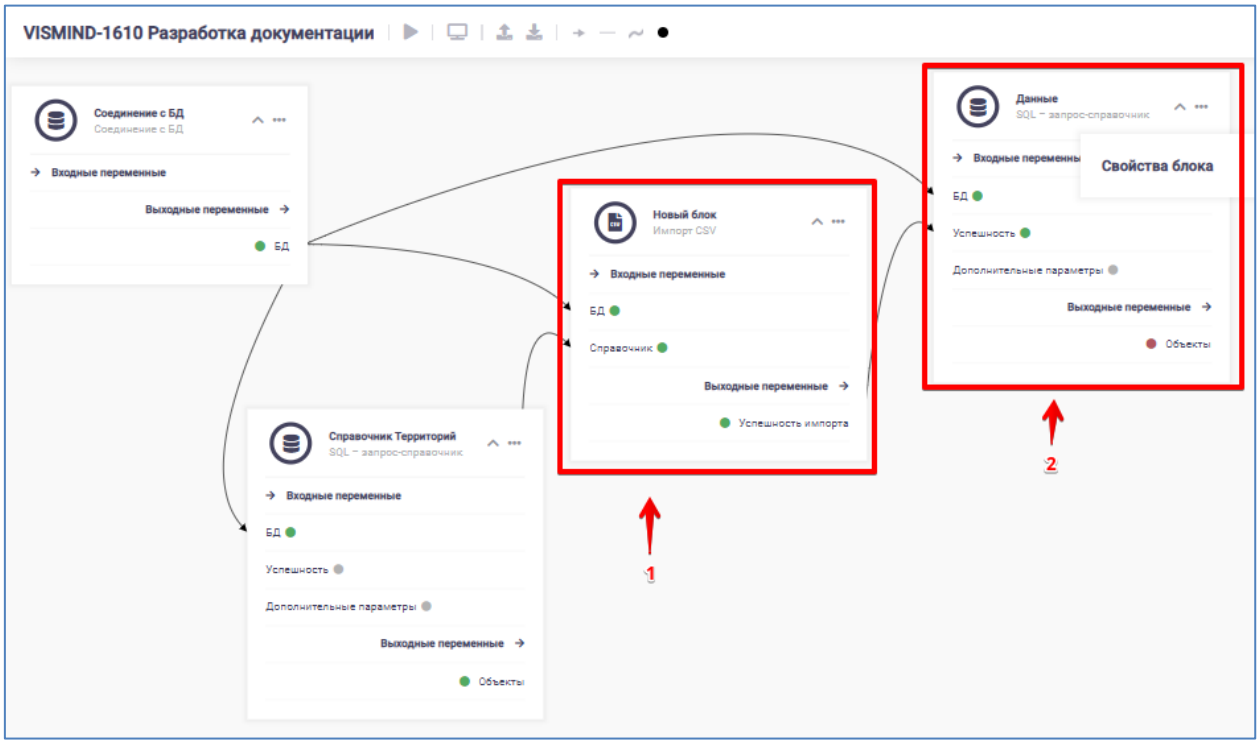


Рисунок 354 – Завершение мастера загрузки данных

Результатом работы мастера загрузки данных, в графе два новых блока:

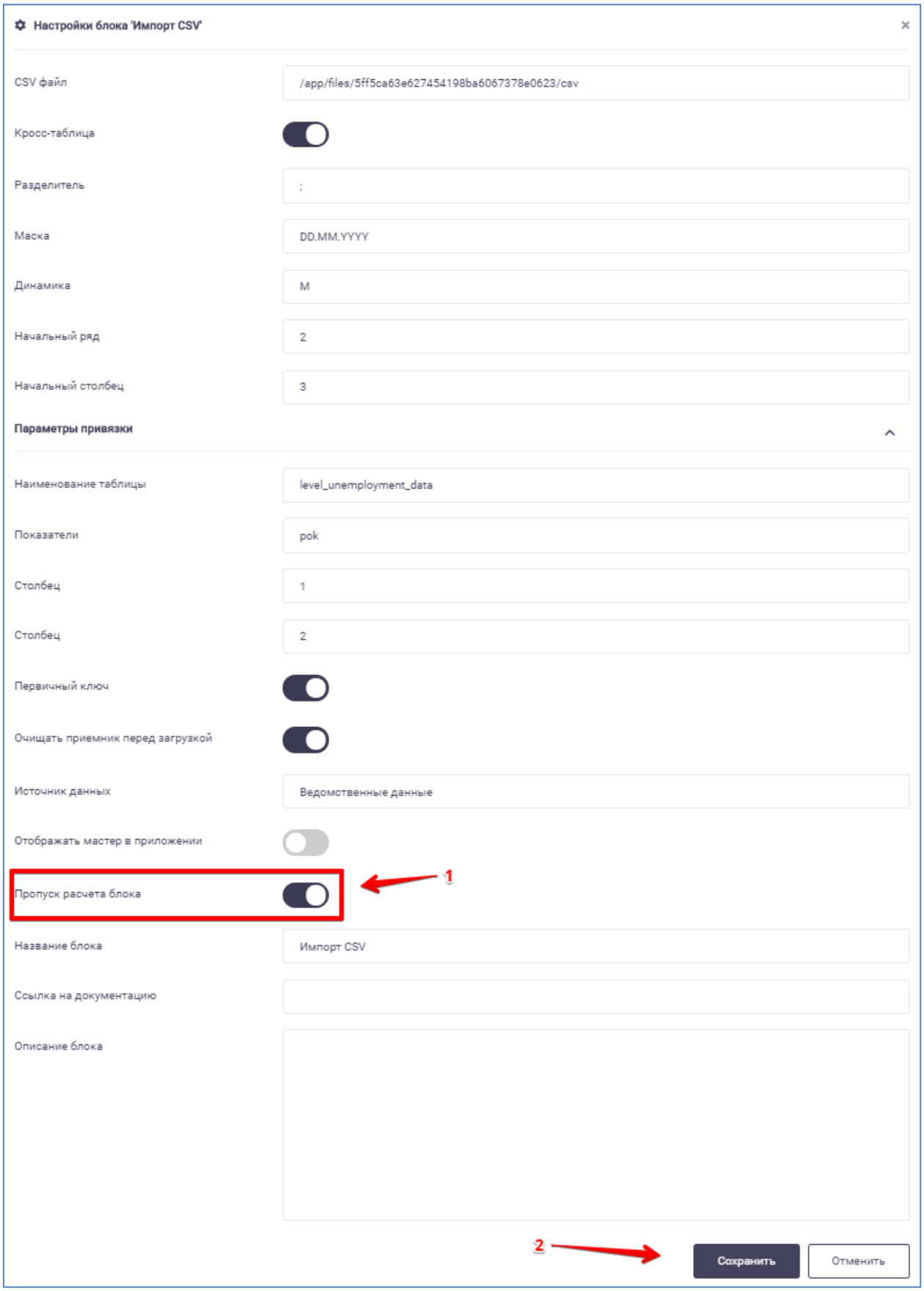
- Блок « «Импорт CSV»;
- Блок «Sql – запрос-справочник» настроенный на таблицу с данными которую указали на шаге 3 мастера загрузки данных.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.2.20.6.1 4.Завершающий этап успешного импорта данных

После завершения мастера загрузки данных необходимо рассчитать граф, после расчета графа перейти в настройки блока «Импорт CSV» и активировать опцию «Пропуск расчета блока».

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)



6.2.21 Блок «SOAP-запрос»

6.2.21.1 Назначение

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

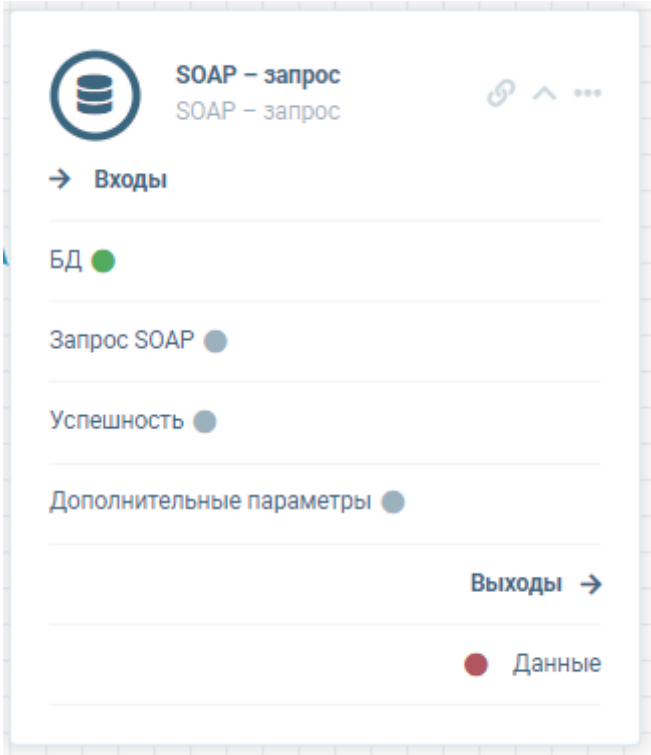


Рисунок 355 – Пример блока «SOAP-запрос»

Блок позволяет получать данные из SOAP сервиса.

6.2.21.2 Настройки блока

Название параметра	Тип параметра	Обязательный	Описание	Пример
Группа «Основные свойства»				
SOAP Action	Строка	Да	Ссылка для отправки запроса к SOAP сервису, по которой будет выполнено действие и получен результат	
Запрос SOAP	XML	Да	Текст SOAP запроса для получения данных. Если на вход блока Запрос SOAP переданы значения, этот параметр	Пример запроса: <?xml version="1.0" encoding="UTF-8"?> <SOAP-ENV:Envelope> <SOAP-ENV:Body> <export_indicatorResponse> <export_indicatorResult> <start_after> asdasdasd </start_after>

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
			игнорируется	</export_indicatorResult> </export_indicatorResponse> </SOAP-ENV:Body> </SOAP-ENV:Envelope>

6.2.21.3 Группа «Общие настройки блока»

Аналогично приведенным в п. 6.1.1.1.3.

6.2.21.4 Входы

БД	Соединение с БД	Да	Соединение с базой данных MongoDB	БД
Запрос SOAP	XML строка	Нет	SOAP запрос, который будет выполнен. Может быть получен во время расчета графа. При передаче его на этот вход, параметр Запрос SOAP внутри блока игнорируется	Запрос SOAP
Успешность	Флаг	Нет	Вход для запуска расчета блока в случае если нет входов (на него можно подать выход Успешность другого блока)	Успешность
Дополнительные параметры	Ряд данных	Нет	Массив объектов или временных рядов, атрибуты которых могут быть использованы для параметризации запроса	Дополнительные параметры

6.2.21.5 Выходы

Название выхода	Тип	Описание
Данные	XML строка	Строка – ответ сервиса

6.2.21.6 События блока

В блоке нет событий.

6.2.21.7 Описание работы блока

Блок выполняет XML запрос к SOAP сервису, после чего возвращает на выходе полученный от сервиса ответ. Ответ, в виде XML строки, необходимо разбирать с помощью блока Python.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.2.22 Блок «Apache Kafka Consumer»

6.2.22.1 Назначение

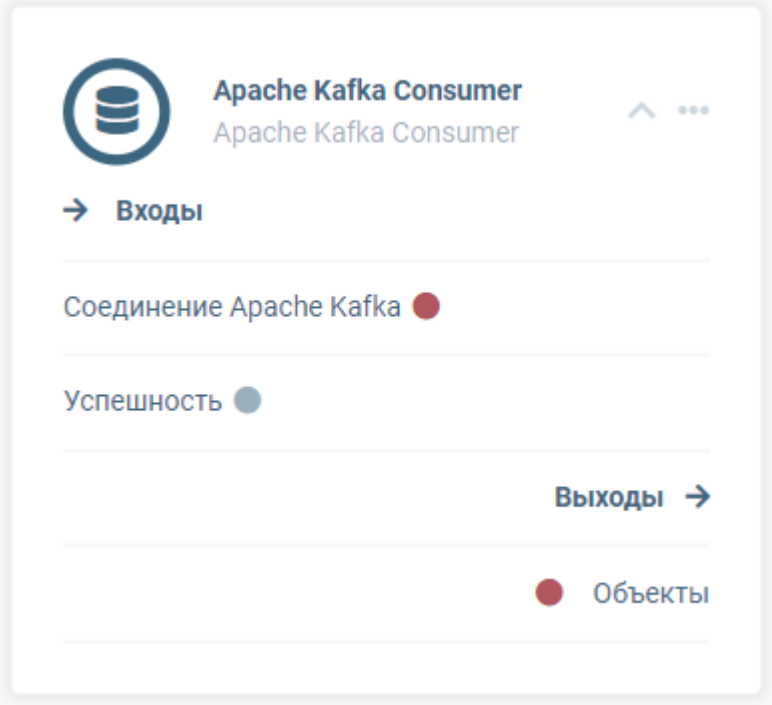


Рисунок 356 – Пример блока «Apache Kafka Consumer»

Блок позволяет получать массив сообщений из Apache Kafka.

6.2.22.2 Настройки блока

Название параметра	Тип параметра	Обязательный	Описание	Пример
Группа "Специфические настройки блока"				
Имя топика	Строка	Да	Топик чтения очереди	
Количество сообщений	Целое	Нет	Количество извлекаемых из очереди сообщений за один расчет блока. Если не задано будут извлечены все сообщения.	
Время ожидания сообщений	Целое	Нет	Время ожидания сообщений (в секундах), если на момент исполнения блока в топике их меньше указанного. Если не задано, будет ожидать поступление нужного количества сообщений.	

6.2.22.3 Группа «Общие настройки блока»

Аналогично приведенным в п. 6.1.1.1.3.

6.2.22.4 Входы

Название входа	Тип	Обязательный	Описание
Соединение Apache Kafka	Соединение с Apache Kafka	Да	Соединение с очередью Apache Kafka

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название входа	Тип	Обязательный	Описание
Успешность	Флаг	Нет	Вход для запуска расчета блока в случае если нет входов (на него можно подать выход Успешность другого блока)

6.2.22.5 Выходы

Название выхода	Тип	Описание
Объекты	JSON	Перечень сообщений, полученных от очереди. Необходимо обрабатывать с помощью блока Python

6.2.22.6 События блока

В блоке нет событий.

6.2.22.7 Описание работы блока

Блок представляет собой Consumer сообщений из очереди Apache Kafka. В параметрах блока указывается какой топик очереди читать, сколько сообщений вычитывать за раз и сколько времени ожидать, если сообщений меньше требуемого. Результатом работы блока является перечень сообщений из очереди. При этом, приложение не разбирает и не меняет их, так как в общем случае структура может быть любой. Обычно, разобрать их можно с помощью блока Python.

Соединение Apache Kafka должно быть передано с типом соединения "Чтение".

6.2.23 Блок «Apache Kafka Producer»

6.2.23.1 Назначение

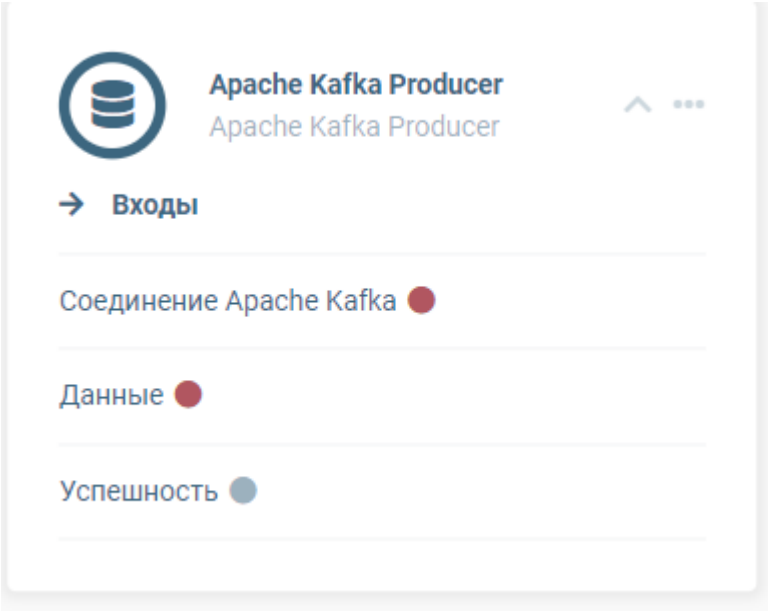


Рисунок 357 – Пример блока «Apache Kafka Producer»

Блок позволяет записать массив сообщений из Apache Kafka.

6.2.23.2 Настройки блока

Название параметра	Тип параметра	Обязательный	Описание	Пример
Группа "Специфические настройки блока"				

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
Имя топика	Строка	Да	Топик записи очереди	

6.2.23.3 Группа «Общие настройки блока»

Аналогично приведенным в п. 6.1.1.1.3.

6.2.23.4 Входы

Название входа	Тип	Обязательный	Описание
Соединение Apache Kafka	Соединение с Apache Kafka	Да	Соединение с очередью Apache Kafka
Данные	Список	Да	Данные для записи в Apache Kafka
Успешность	Флаг	Нет	Вход для запуска расчета блока в случае, если нет входов (на него можно подать выход Успешность другого блока)

6.2.23.5 Выходы

В блоке нет выходов.

6.2.23.6 События блока

В блоке нет событий.

6.2.23.7 Описание работы блока

Блок представляет собой Producer сообщений очереди Apache Kafka. В параметрах блока указывается в какой топик очереди будут сохраняться сообщения.

Соединение Apache Kafka должно быть передано с типом соединения "Запись".

6.2.24 Блок «MongoDB – запрос»

6.2.24.1 Назначение

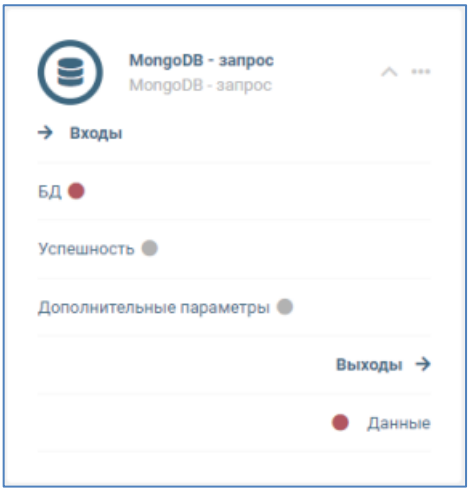


Рисунок 358 – Пример блока «MongoDB – запрос»

Блок позволяет получать массив рядов из MongoDB по запросу.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.2.24.2 Настройки блока

Название параметра	Тип параметра	Обязательный	Описание	Пример
Группа «Основные свойства»				
Коллекция	Строка	Да	Название коллекции	Пример: incidents_geo
Фильтр	JSON	Да	Строка запроса. Может быть параметризована из входа «Дополнительные параметры»	Пример фильтра: `{ "$and":[{"properties.area": "Иркутск"},{"properties.deviceId":897826}] }`
Видимость полей	Флаг	Нет	Флаг, отвечающий за видимость отображения полей, которые вернет запрос. Если флаг включен, то вернет только те поля, которые перечислены. Если флаг выключен, вернет все поля, кроме перечисленных. По умолчанию флаг включен	
Перечисление полей	Строка	Нет	Перечисление полей которые вернет запрос. Если поля не указаны, вернет все поля. Поля перечисляются через точку с запятой («;»)	Пример строки: _id;properties.eventName; name; start_timestamp; end_timestamp; properties.deviceType
Сортировка	Строка	Нет	Строка сортировки данных. Передаются в формате название поля: тип сортировка («1» – по возрастанию (asc), «-1» – по убыванию (desc)). Сортировки перечисляются через точку с запятой («;»)	Пример строки: _id:1;properties.eventName:-1
Количество выводимых записей	Целое число	Нет	Количество выводимых записей	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
Количество пропускаемых записей	Целое число	Нет	Количество пропускаемых записей	

Параметры привязки	JSON	Нет	Параметры, на основе которых формируются объекты. Без указания привязки объекты не будут иметь первичных ключей.
			• **pkey** – массив ключей, на основании которого будет сформирован первичный ключ каждого объекта;
			• **attrs** – массив ключей, на основании которого будет сформированы атрибуты объекта. Если параметр не указан, то атрибуты попадут все поля которые не указаны в первичном ключе, но есть в запросе данных;
			• **params** – расширенные параметры объекта:
			• **ts** – массив с настройками формата даты:
			  • **key** – ключ поля, в котором находится значение даты;
			  • **mask** – маска даты **Доступные форматы для маски** Развернуть исходный код;
			  • **frequency** – период
				Пример привязок:
				{
				"pkey":[
				"properties.eventName"
				],
				"attrs": [
				"properties.deviceType"
				],
				"params": {
				"ts": [{
				"mask": "%Y.%m.%d",
				"frequency": "Y"
				}]
				}
				}

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.2.24.3 Группа «Общие настройки блока»

Аналогично приведенным в п. 6.1.1.1.3.

6.2.24.4 Входы

Название входа	Тип	Обязательный	Описание
БД	Соединение с БД	Да	Соединение с базой данных MongoDB
Успешность	Флаг	Нет	Вход для запуска расчета блока в случае если нет входов (на него можно подать выход Успешность другого блока)
Дополнительные параметры	Ряд данных	Нет	Массив объектов или временных рядов, атрибуты которых могут быть использованы для параметризации запроса

6.2.24.5 Выходы

Название выхода	Тип	Описание
Данные	Объект	Массив объектов

6.2.24.6 События блока

В блоке нет событий.

6.2.24.7 Описание работы блока

Блок выполняет запрос из базы данных MongoDB, после чего преобразует каждую полученную строку в объект по шаблону, заданному в поле «Параметры привязки».

Результат работы блока – массив объектов.

6.2.25Блок «Конструктор запросов»

6.2.25.1 Назначение

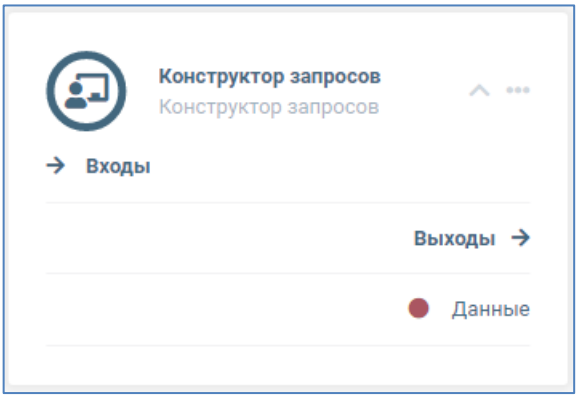


Рисунок 359 – Блок «Конструктор запросов»

Блок позволяет сконструировать запрос для извлечения данных из выбранного источника данных.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.2.25.2 Настройки блока

Название параметра	Тип параметра	Обязательный	Описание	Пример
Группа «Основные свойства»				
Источник данных	Дерево	Да	Выбор необходимого источника данных из доступных в приложении	
Столбцы для отображения	Дерево	Да	Выбор первичных ключей и атрибутов ряда, сформированного в источнике, для отображения в результатах выполнения блока. Для отображения элементов необходимо выбрать источник данных и нажать на кнопку «Обновить» . Источник данных, при этом, должен быть с рассчитанными данными	
Условия				
Фильтр	Дерево	Да	Выбор входа в источнике данных Блок «Входные данные» (п. 6.7.1) для установки параметров запроса. Для отображения элементов необходимо выбрать источник данных и нажать на кнопку «Обновить»	
Выбор элемента	Выпадающий список	Да	Выбор формата настройки элемента. Возможные значения: **Задать вручную** – ручной ввод необходимого значения. **Выбрать из справочника** – выбор значений из привязанного справочника. Подробнее см привязка справочников к выходу блока	
Операция	Выпадающий список	Да	Доступные операции для элементов	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
Значение	Строка/Дерево	Да	Значение элемента. Формат ввода зависит от настройки «Выбор элемента» • При «Задать вручную» – строка. • При «Выбрать из справочника» – дерево значений. Для отображения элементов необходимо заполнить поля «**источник данных**» и «**фильтр**» и нажать на кнопку «Обновить»	

6.2.25.3 Группа «Общие настройки блока»

Аналогично приведенным в п. 6.1.1.1.3.

6.2.25.4 Входы

В блоке нет входов.

6.2.25.5 Выходы

Название выхода	Тип	Описание

Данные	Ряд данных	Массив рядов данных
		Выход из блока
		{
		"pkey": [
		<Набор первичных ключей>
		],
		"vl": [
		<Массив значений ряда>
		],
		"attrs": [
		<Набор атрибутов – полей, не попавших ни в первичный ключ, ни в значение.>
		],
		"_params": {
		<Опциональные параметры объекта>,
		"vl": <Ключ поля, отвечающего за значение>,
		"fkey": [
		<Ключ поля, по которому собран ряд>
		]
		},
		"_source": <Идентификатор блока-источника, поле заполняется калькулятором автоматически>,
		}

6.2.25.6 События блока

В блоке нет событий.

6.2.25.7 Описание работы блока

Блок преобразует все условия во входные данные, после чего запускает расчет источника данных.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

После успешного расчета, у всех полученных рядов удаляются первичные ключи и/или атрибуты, не указанные в настройке «Столбцы для отображения».

6.2.25.7.1 Настройки источника данных

Для корректной работы блока необходимы специфические настройки источника данных:

- Все элементы, заданные вручную, приходят в источник данных в качестве строки формата <Операция>::<Значение>, где :: (двойное двоеточие) – константа для разделения операции и значения.
- Все элементы, выбранные из справочника приходят как массив объектов этого справочника.

6.2.25.8 Описание работы визуализатора

Блок не является визуализатором.

6.3 Источники данных и НСИ

6.3.1 Блок «&Источник данных»

6.3.1.1 Назначение

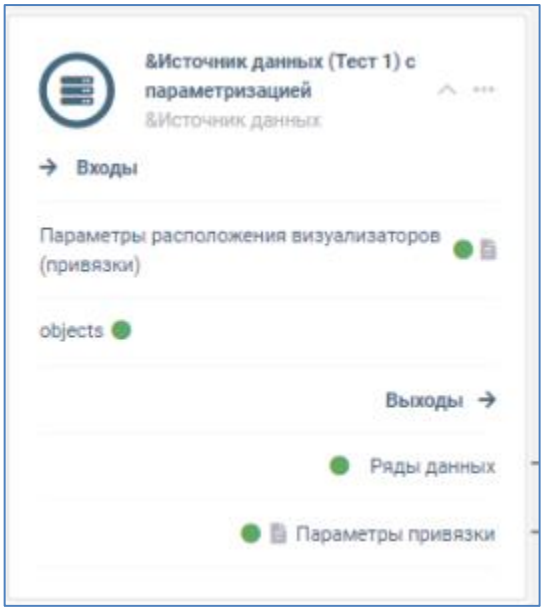


Рисунок 360 – Пример блока «&Источник данных»

Блок позволяет добавить выбранный источник данных в граф и использовать его данные в расчетах.

6.3.1.2 Настройки блока

Название параметра	Тип параметра	Обязательный	Описание	Пример
Группа «Основные свойства»				
Список источников данных	Дерево	Да	Выбор необходимого источника данных из доступных в приложении	

6.3.1.3 Группа «Общие настройки блока»

Аналогично приведенным в п. 6.1.1.1.3.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.3.1.4 Входы

Название входа	Тип	Описание
Параметры расположения визуализаторов (привязки)	Строка	Параметры привязки для дополнительной параметризации объектов, Применяется после расчета источника данных, на полученных данных. Представляют собой строковое описание настройки «**Параметры привязки**» блока «Панель OLAP» (п. 6.1.4.8) или блока «Панель размещения и фильтрации» (п. 6.1.4.10) (в режиме отображения «Панель OLAP»)

Количество входов может динамически менять, в зависимости от выбранного источника данных. Подробнее описание приведено в п. 6.7.1 Блок «Входные данные».

6.3.1.5 Выходы

Название выхода	Тип	Описание
Ряды данных	Ряд данных	Массив рядов данных
Параметры привязки	Строка	Динамические параметры привязки для корректного отображения блоков-визуализаторов

6.3.1.6 События блока

В блоке нет событий.

6.3.1.7 Описание работы блока

В момент расчета графа блок запускает расчет источника данных (с входными параметрами, если они заданы), после чего передает рассчитанное значение на выход блока.

6.3.1.7.1 Форматирование выходных данных

Блок умеет форматировать выходные данные по аналогии с блоком «Панель OLAP» (п. 6.1.4.8).

При наличии данных на входе «**Параметры расположения визуализаторов (привязки)**», после расчета источника данных и получения рядов, блок автоматически применяет параметры привязки для формирования новых рядов данных.

Кроме этого, так же передадутся данные в выход «**Параметры привязки**», для формирования корректных визуализаторов.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.3.2 Блок «&НСИ»

6.3.2.1 Назначение

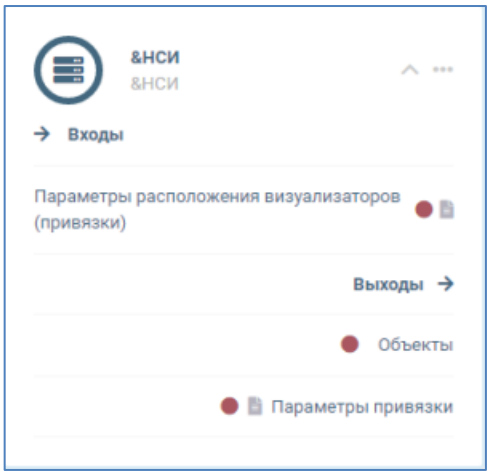


Рисунок 361 – Пример блока «&НСИ»

Блок позволяет добавить выбранный НСИ в граф и использовать его данные в расчетах.

6.3.2.2 Настройки блока

Название параметра	Тип параметра	Обязательный
Группа «Основные свойства»		
Список справочников	Дерево	Да

6.3.2.3 Группа «Общие настройки блока»

Аналогично приведенным в п. 6.1.1.1.3.

6.3.2.4 Входы

В блоке нет входов.

6.3.2.5 Выходы

Название выхода	Тип	Описание
Объекты	Объект	Массив объектов.

6.3.2.6 События блока

В блоке нет событий.

6.3.2.7 Описание работы блока

В момент расчета графа блок запускает расчет НСИ (с входными параметрами, если они заданы), после чего передает рассчитанное значение на выход блока.

6.3.3 Блок «&НСИ – Выход»

См. Блок «Выходные данные» (п. 6.7.2).

6.4 Оформление графа

Оформление графа.

6.4.1 Блок «Граф»

6.4.1.1 Назначение

Блок позволяет выводить граф на главную панель приложения.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.4.1.2 Настройки блока

Название параметра	Тип параметра	Обязательное	Описание	Пример
Название блока	Строка	Нет	Наименование блока	
Гиперссылка на граф	Строка	Да	Ссылка на граф	
Ссылка на документацию	Строка	Нет		
Описание блока	Строка	Нет		
Пропуск расчета блока	Флажок	Нет	Рассчитывать или нет данный блок	

6.4.1.3 Входы

Блок не имеет конкретных входов.

6.4.1.4 Выходы

Блок не имеет конкретных выходов.

6.4.2 Блок «Задача»

6.4.2.1 Назначение

Блок позволяет добавить пресеты с выбранной задачи на текущую.

Для корректной работы необходимо, чтобы в задаче присутствовал Блок «Меню» (п. 6.1.4.7).

6.4.2.2 Настройки блока

Название параметра	Тип параметра	Обязательный	Описание	Пример
Название блока	Строка	Нет	Наименование блока	
Задача	Выделение из списка	ДА	Выбор задачи, из которой необходимо показать пресеты	
Ссылка на документацию	Строка	Нет		
Описание блока	Строка	Нет		
Пропуск расчета блока	Флажок	Нет	Рассчитывать или нет данный блок	

6.4.2.3 Входы

Блок не имеет конкретных входов.

6.4.2.4 Выходы

Блок не имеет конкретных выходов.

6.4.3 Блок «Надпись»

6.4.3.1 Назначение

Блок позволяет формировать надпись на графе.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.4.3.2 Настройки блока

Название параметра	Тип параметра	Обязательное	Описание	Пример
Текст	Строка	Нет	Текст надписи	
Оформление	Выпадающий список	Нет	Варианты оформления надписи	
Прозрачность	Целое число	Нет	Прозрачность надписи	
Цвет	Выбор цвета	Нет		
Цвет заливки	Выбор цвета	Нет		
Тень	Флажок	Нет		

6.4.3.3 Входы

Блок не имеет конкретных входов.

6.4.3.4 Выходы

Блок не имеет конкретных выходов.

6.4.4 Блок «Прямоугольник»

6.4.4.1 Назначение

Рисунок 362 – Пример блока «Прямоугольник»

Блок позволяет формировать и настраивать прямоугольник на графе.

6.4.4.2 Настройки блока

Название параметра	Тип параметра	Обязательное	Описание	Пример
Раздел «Заголовок»				
Наименование	Строка	Нет	Наименование блока.	
Размер шрифта	Целое число	Нет	Позволяет настраивать шрифта заголовка. *По умолчанию – 20.*	
Раздел «Границы»				
Отображение	Флаг	Нет	Позволяет включать и отключать отображение границы.	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательное	Описание	Пример
Цвет	Цвет	Нет	Позволяет настраивать цвет границ блока. *По умолчанию – #000000*	
Толщина	Целое число	Нет	Позволяет настраивать толщину границ. *По умолчанию – 5.*	
Стиль	Выпадающий список	Нет	Позволяет настраивать стиль границ. Возможные варианты: Сплошная линия *(по умолчанию)* Пунктир	
Раздел «Отображение»				
Цвет	Цвет	Нет	Позволяет настраивать цвет заливки прямоугольника. *По умолчанию – rgba(255, 255, 255, 0)*	
Прозрачность	Целое число	Нет	Позволяет настраивать прозрачность заливку прямоугольника. *По умолчанию – 0*	
Тень	Флаг	Нет	Позволяет включать и отключать отображение «тени» на графе. *По умолчанию – активно*	

6.4.4.3 Входы

В блоке нет конкретных входов.

6.4.4.4 Выходы

В блоке нет конкретных выходов.

6.4.4.5 События блока

В блоке нет событий.

6.4.4.6 Описание работы блока

Позволяет создавать и настраивать прямоугольные фигуры на графе. Подобные фигуры используются для сортировки блоков на графе по различным признакам, что в дальнейшем упрощает навигацию по большим графам.

Блок создаётся через библиотеку блоков графа, но для взаимодействия с ним необходимо перейти на слой фигур.

Для этого в меню управления графом необходимо нажать кнопку «Блоки».

Рисунок 363 – Кнопка «Блоки»

В появившемся выпадающем списке выбрать «Фигуры».

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

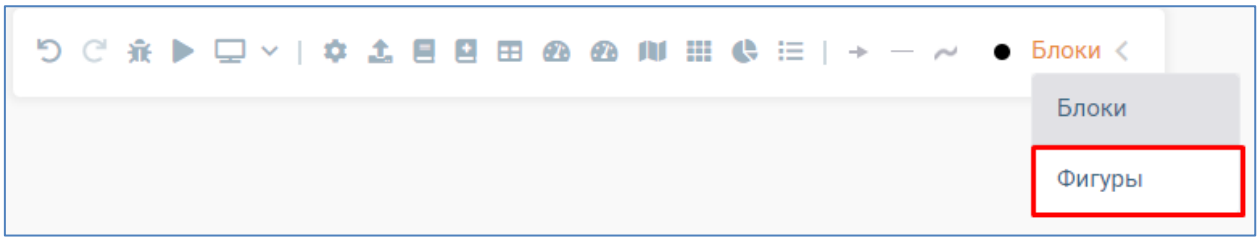


Рисунок 364 – Выбор значения «Фигуры»

6.4.4.7 Настройки визуализатора

Нет.

6.5 Преобразование данных

6.5.1 Блок «Выражение»

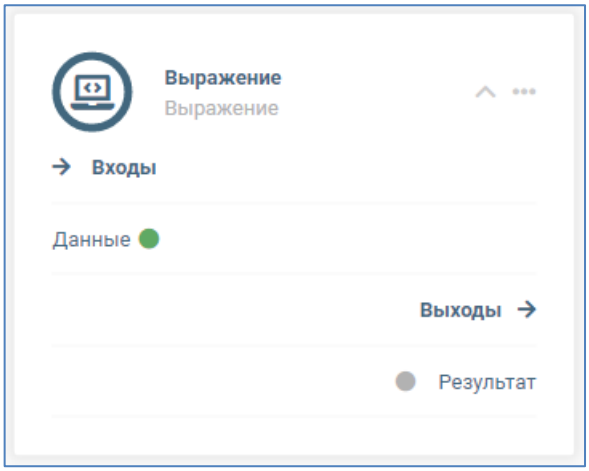


Рисунок 365 – Пример блока «Выражение»

6.5.1.1 Назначение

Блок предназначен для преобразования объектов или рядов с помощью различных выражений, описываемых в блоке.

6.5.1.2 Настройки блока

Название параметра	Тип параметра	Обязательный	Описание	Пример
Выражение	Ввод выражения	Да	Поле ввода кода выражения	
Количество знаков после запятой	Целое число	Нет	Настройка количества знаков после запятой для вещественного результата выражения	

6.5.1.3 Группа «Общие настройки блока»

Аналогично приведенным в п. 6.1.1.1.3.

6.5.1.4 Входы

Для создания входа необходимо нажать на знак + в разделе входов блока. Для каждого из входов задается следующий перечень полей.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Идентификатор	Строка	Да	Идентификатор входа	Идентификатор
Имя	Строка	Да	Имя входа для отображения в списке входов блока В отличии от других блоков, используется для получения доступа к данным входа	**Имя**
Описание	Строка	Нет	Дополнительная информация с описанием входа	**Описание**
Тип	Выпадающий список	Да	Тип данных, которые будут приходить на вход. Нужен для построения корректных связей с другими блоками. По умолчанию установлен тип "JSON".	**Тип**
Обязательный	Флаг	Да	Обязательность входа. Если вход обязательный и на него не пришло данных – блок не будет рассчитан с ошибкой	**Обязательный**
Массив	Флаг	Да	Является ли вход массивом данных	**Массив**

6.5.1.5 Выходы

Название выхода	Тип	Описание
Результат	JSON	Результат выражения

6.5.1.6 События блока

В блоке нет событий.

6.5.1.7 Описание работы блока

Блок считывает код, написанный в поле **Выражение**, после чего последовательно совершает операции.

Каждая операция выражения возвращает какой-либо результат. Результат последней операции будет направлен в выход блока.

6.5.1.8 Базовые операции

Для блока «Выражения» доступны следующие базовые операции над объектами:

- **Сложение**
- **Вычитание**
- **Умножение**
- **Деление**
    - Простое деление:
        - <Выражение 1> [/] <Выражение 2>
    - Получение остатка от деления:
        - <Выражение 1> [%] <Выражение 2>
    - Получение целой части от деления:

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

- <Выражение 1> [//] <Выражение 2>

- Возведение в степень
  - <Выражение> [** | ^] <Число>
- **Фактор** – отрицательное значение объекта.
  - [–]<Выражение>
- **Сравнение**
  - <Выражение 1> [< | <= | > | >= | == | !=] <Выражение 2>
- **Инверсия** – логическое «не»
  - [not] <Выражение 1>
- **Конъюнкция** – логическое «и»
  - <Выражение 1> [and]<Выражение 2>
- **Дизъюнкция** – логическое «или»
  - <Выражение 1> [or] <Выражение 2>

6.5.1.8.1 Грамматика операций

Операнды	Описание	Пример
Сложение		
<Выражение 1> **[+]** <Выражение 2>		
Целое число + Целое число = Целое число		2 + 2 = 4
Целое число + Число с плавающей точкой = Число с плавающей точкой		2 + 2.7 = 4.7
Целое число + Строка = Строка	Целое число превращается в строку и происходит суммирование строк	2 + 'String' = '2String'
Целое число + Логический тип = Целое число	Логический тип превращается в целое число (0 – False, 1 – True), после чего происходит сложение	2 + True = 3
Целое число + Дата = Дата	К дате добавляется <Целое число> дней	2 + 01.01.2022 = 03.01.2022
Целое число + Пустой элемент = Целое число	Пустой элемент приравнивается к нулю и происходит сложение	2 + Null = 2
Число с плавающей точкой + Число с плавающей точкой = Число с плавающей точкой		2.7 + 3.3 = 6.0
Число с плавающей точкой + Строка = Строка	Число с плавающей точкой превращается в строку и происходит суммирование строк	'str' + 0.1415 = 'str0.1415'
Число с плавающей точкой + Логический тип = Число с плавающей точкой	Логический тип превращается в целое число (0 – False, 1 – True), после чего происходит сложение	4.2 + True = 5.2
Число с плавающей точкой + Дата = Дата	К дате добавляется округленное до ближайшего целого <Число с плавающей точкой> дней	10.7 + 01.01.2022 = 12.01.2022

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Операнды	Описание	Пример
Число с плавающей точкой + Пустой элемент = Число с плавающей точкой	Пустой элемент приравнивается к нулю и происходит сложение	5.5 + Null = 5.5
Строка + Строка = Строка		'foo' + 'bar' = 'foobar'
Строка + Логический тип = Строка	Логический тип переводится в строку (True в 'True', False в 'False') и происходит сложение строк.	'not' + True = 'notTrue'
Строка + Дата = Строка	Дата преобразуется в строку и происходит сложение строк	01.01.2001 + 'Year' = '01.01.2001Year'
Строка + Пустой элемент = Строка	Пустой элемент приравнивается к пустой строке и происходит сложение строк	'Text' + Null = 'Text'
Логический тип + Логический тип = Целое число	Логический тип превращается в целое число (0 – False, 1 – True), после чего происходит сложение двух целых чисел.	True + True = 2
Логический тип + Дата = Дата	Логический тип превращается в целое число (0 – False, 1 – True), после чего происходит сложение целого числа и даты	True + 31.12.2021 = 01.01.2022
Логический тип + Пустой элемент = Логический тип	Пустой элемент игнорируется и возвращается значение логического типа	False + Null = False
Дата + Дата = Ошибка!	Сложение двух дат вызывает ошибку	
Дата + Пустой элемент = Дата	Пустой элемент приравнивается к нулю и происходит сложение даты и целого числа	Null + 31.12.2021 = 31.12.2021
Вычитание		
<Выражение 1> **[-]** <Выражение 2>		
Целое число – Целое число = Целое число		2 – 1 = 1
Целое число – Число с плавающей точкой = Число с плавающей точкой		3.14 – 3 = 0.14
Целое число – Строка = Ошибка!	Вычитание строки вызывает ошибку	
Целое число – Логический тип = Целое число	Логический тип превращается в целое число (0 – False, 1 – True), после чего происходит вычитание	2 – True = 1
Целое число – Дата = Ошибка!	Вычитать число можно только из даты	02.01.2020 – 1 = 01.01.2020
Дата – Целое число = Дата	Из даты вычитается <Целое число> дней	
Целое число – Пустой элемент = Целое число	Пустой элемент приравнивается к нулю и происходит вычитание	2 – Null = 2

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Операнды	Описание	Пример
Число с плавающей точкой – Число с плавающей точкой = Число с плавающей точкой		3.1415 – 2.1 = 1.015
Число с плавающей точкой – Строка = Строка	Вычитание строки вызывает ошибку	
Число с плавающей точкой – Логический тип = Число с плавающей точкой	Логический тип превращается в целое число (0 – False, 1 – True), после чего происходит вычитание	2.7 – True= 1.7
Число с плавающей точкой – Дата = Ошибка!	Вычитать число можно только из даты	02.02.2020 – 2.7 = 31.01.2020
Дата – Число с плавающей точкой = Дата	Из даты вычитается округленное до ближайшего целого <Число с плавающей точкой> дней	
Число с плавающей точкой – Пустой элемент = Число с плавающей точкой	Пустой элемент приравнивается к нулю и происходит вычитание	2.7 – Null = 2.7
Строка – Строка = Ошибка!	Вычитание строки вызывает ошибку.	
Строка – Логический тип = Ошибка!		
Строка – Дата = Ошибка!		
Строка – Пустой элемент = Ошибка!		
Логический тип – Логический тип = Целое число	Логический тип превращается в целое число (0 – False, 1 – True), после чего происходит вычитание двух целых чисел.	False – True = -1
Логический тип – Дата = Ошибка!	Вычитать число можно только из даты	01.01.2022 – False = 01.01.2022
Дата – Логический тип – Дата	Логический тип превращается в целое число (0 – False, 1 – True), после чего происходит вычитание целого числа из даты	
Логический тип – Пустой элемент = Ошибка!	Вычитание пустого элемента вызывает ошибку	
Дата – Дата = Ошибка!	Вычитание даты вызывает ошибку.	
Дата – Пустой элемент = Ошибка!	Вычитание пустого элемента вызывает ошибку	
Умножение		
<Выражение 1> **[*]** <Выражение 2>		
Целое число * Целое число = Целое число		6 * 5 = 30
Целое число * Число с плавающей точкой = Число с плавающей точкой		2 * 2.2 = 4.4
Целое число * Строка = Строка	Происходит дублирование строки <Целое число> раз	5 * 'ta' = 'tatatatata'

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Операнды	Описание	Пример
Целое число * Логический тип = Целое число	Логический тип превращается в целое число (0 – False, 1 – True), после чего происходит умножение	10 * False = 0
Целое число * Дата = Ошибка!	Умножение даты вызывает ошибку	
Целое число * Пустой элемент = Целое число	При умножении на пустой элемент всегда получится ноль.	5 * Null = 0
Число с плавающей точкой * Число с плавающей точкой = Число с плавающей точкой		2.1 * 2.5 = 5.25
Число с плавающей точкой * Строка = Строка	Происходит дублирование строки <Число с плавающей точкой>, округленное до ближайшего целого раз	2.4 * 'foo' = 'foofoo'
Число с плавающей точкой * Логический тип = Число с плавающей точкой	Логический тип превращается в целое число (0 – False, 1 – True), после чего происходит умножение	3.14 * True = 3.14
Число с плавающей точкой * Дата = Ошибка!	Умножение даты вызывает ошибку	
Число с плавающей точкой * Пустой элемент = Целое число	При умножении на пустой элемент всегда получится ноль	3.14 * Null = 0
Строка * Строка = Ошибка!	Перемножение двух строк вызывает ошибку	
Строка * Логический тип = Строка	Логический тип превращается в целое число (0 – False, 1 – True), после чего происходит дублирование строки (итогом будет либо пустая строка, либо та же самая строка без изменений)	'foo' * False = ''
Строка * Дата = Ошибка!	Умножение даты вызывает ошибку	
Строка * Пустой элемент = Целое число	При умножении на пустой элемент всегда получится ноль	'foo' * Null = 0
Логический тип * Логический тип = Целое число	Логический тип превращается в целое число (0 – False, 1 – True), после чего происходит умножение	True * False = 0
Логический тип * Дата = Дата	Логический тип работает как «маска» в логическом умножении. Итогом будет либо 01.01.0001 (минимальная дата для языка Python), либо исходная дата	01.01.2020 * True = 01.01.2020
Логический тип * Пустой элемент = Логический тип	При умножении на пустой элемент всегда получится ноль	True * Null = 0
Дата * Дата = Ошибка!	Перемножение дат вызывает ошибку	
Дата * Пустой элемент = Целое число	При умножении на пустой элемент всегда получится ноль	01.01.2020 * Null = 0
Деление		
Простое деление:		
<Выражение 1> **[/]** <Выражение 2>		

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Операнды	Описание	Пример
Получение остатка от деления:		
<Выражение 1> **[%]** <Выражение 2>		
Получение целой части от деления:		
<Выражение 1> **[//]** <Выражение 2>		
Целое число / Целое число = Число с плавающей точкой		4 / 2 = 2.0
Целое число / Число с плавающей точкой = Число с плавающей точкой		5 / 2.5 = 2.0
Целое число / Строка = Ошибка!	Деление строки вызывает ошибку	
Целое число / Логический тип = Целое число	Логический тип превращается в целое число (0 – False, 1 – True), после чего происходит деление Если производить деление на False, то это повлечет за собой ошибку деления на ноль	1 / True = 1
Целое число / Дата = Ошибка!	Деление даты вызывает ошибку	
Целое число / Пустой элемент = Пустой элемент	При делении на пустой элемент всегда образуется пустой элемент	10 / Null = Null
Пустой элемент / Целое число = Целое число	При делении пустого элемента всегда образуется ноль	Null / 10 = 0
Число с плавающей точкой / Число с плавающей точкой = Число с плавающей точкой		12.1 / 2.5 = 4.84
Число с плавающей точкой / Строка = Ошибка!	Деление строки вызывает ошибку	
Число с плавающей точкой / Логический тип = Целое число	Логический тип превращается в целое число (0 – False, 1 – True), после чего происходит деление Если производить деление на False, то это повлечет за собой ошибку деления на ноль	
Число с плавающей точкой / Дата = Ошибка!	Деление даты вызывает ошибку	
Число с плавающей точкой / Пустой элемент = Пустой элемент	При делении на пустой элемент всегда образуется пустой элемент	3.14 / Null = Null
Пустой элемент / Число с плавающей точкой = Целое число	При делении пустого элемента всегда образуется ноль	Null / 3.14 = 0
Строка / Строка = Ошибка! Строка / Логический тип = Ошибка! Строка / Дата = Ошибка!	Деление строки вызывает ошибку	
Строка / Пустой элемент = Пустой элемент	При делении на пустой элемент всегда образуется пустой элемент	'foo' / Null = Null

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Операнды	Описание	Пример
Пустой элемент / Строка = Целое число	При делении пустого элемента всегда образуется ноль	Null / 'foo' = 0
Логический тип / Логический тип = Целое число	Логический тип превращается в целое число (0 – False, 1 – True), после чего происходит деление. Если производить деление на False, то это повлечет за собой ошибку деления на ноль	True / True = 1
Логический тип / Дата = Ошибка!	Деление даты вызывает ошибку	
Логический тип / Пустой элемент = Пустой элемент	При делении на пустой элемент всегда образуется пустой элемент	True / Null = Null
Пустой элемент / Логический тип = Целое число	При делении пустого элемента всегда образуется ноль (даже в случае деления на False)	Null / True = 0
Дата / Дата = Ошибка!	Деление даты вызывает ошибку	
Дата / Пустой элемент = Пустой элемент	При делении на пустой элемент всегда образуется пустой элемент	01.01.2020 / Null = Null
Пустой элемент / Дата = Целое число	При делении пустого элемента всегда образуется ноль.	Null / 01.01.2020 = 0
Возведение в степень		
<Выражение> **[** | ^]** <Число>	Возведение в степень представляет собой перемножение <Выражение> <Число> раз, таким образом, исходя из правил выше, и <Выражение> и <Число> могут являться только следующими объектами: • Целое число • Число с плавающей точкой • Логический тип	5 ** 2 = 125
Фактор		
[–] <Выражение>	Операция представляет собой умножение <Выражение> на целое число -1	-True = -1
Сравнение		
<Выражение 1> **[< | <= | > | >= | == | !=]** <Выражение 2> --> Логический тип	Не зависимо от операндов, результатом сравнения всегда будет логический тип	1 < 2 = True
Инверсия		
[not | не] <Выражение 1> --> Логический тип	Не зависимо от операндов, результатом инверсии всегда будет логический тип Любые выражения, отличные от False, 0 или Null являются истиной, и следовательно, результатом будет False (для всех исключений результатом будет True)	not 1 = False

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Операнды	Описание	Пример
Конъюнкция		
<Выражение 1> **[and | и]** <Выражение 2> --> Логический тип	Не зависимо от операндов, результатом конъюнкции всегда будет логический тип	'foo' and 2 = True
	Любые выражения, отличные от False, 0 или Null являются истиной	
	В связи с особенностью реализации, руский вариант конъюнкции (И) обязательно должен разделяться пробелами от операндов	
Дизъюнкция		
<Выражение 1> **[or | или]** <Выражение 2> --> Логический тип	Не зависимо от операндов, результатом дизъюнкции всегда будет логический тип	False or 0 = False
	Любые выражения, отличные от False, 0 или Null являются истиной	
Присваивание		
<Выражение 1> **[=]** <Выражение 2>	Присваивание значения, полученного в <Выражение 2> в <Выражение 1>	
	Доступно только для типов <Объект> и <Ряд>	

6.5.1.9 Конструкции

6.5.1.9.1 Вход

Вход используется для получения доступа к данным извне блока.

Получение входа по идентификатору

```
<[Вход | Input]>{[id | идентификатор] = <идентификатор>}
```

- `Идентификатор - идентификатор входа.`

Несмотря на то, что можно создавать входы с одинаковыми наименованиями, не рекомендуется этого делать, во избежание некорректного расчета.

Сокращенная нотация: получение входа по его наименованию или идентификатору

```
<Идентификатор/Наименование>
```

Поиск будет осуществляться как по идентификатору, так и по наименованию.

Если имеется два входа с одинаковыми наименованиями/идентификаторами, то будет взят первый попавшийся.

Данные во входе приходят в тех настройках, которые были указаны при создании входа. Таким образом, если указать во входе, что он является массивом, то и в блок попадет массив, даже если внутри этого массива содержится один элемент.

Актуально и обратное - если в данных приходит массив, а в настройках входа массив не указан, то блок не будет считать входные данные массивом.

6.5.1.9.1.1 Получение значений входа

Для объектов и рядов есть возможность получить значение выбранного элемента.

Параметры получения значения задаются в фигурных скобках (**{}**) после определения входа.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Если была использована не сокращенная нотация входа (с фигурными скобками), то для получения значений необходимо добавить после предыдущих скобок:

```
<[Вход | Input]>{[id | идентификатор] = <идентификатор>}{<параметры>}
```

В описании примеров ниже, первые фигурные скобки будут опущены для упрощения записи

6.5.1.9.1.1.1 Формат записи:

```
'<ВХОД>'{<ключ> : <значение>, <ключ> : <значение>,...}
```

- **ключ** – наименование атрибута или ключа объекта/ряда;
- **значение** – значение выбранного ключа.

6.5.1.9.1.1.2 Особенности реализации:

- Вход должен быть массивом (т.к. происходит выборка элементов из массива).
- Доступно только для объектов и рядов; для всех остальных типов данных конструкция «фигурные скобки» у входа будет вызывать ошибку.
- Если параметры недостаточны (подходит несколько элементов для выбора) – блок выдаст ошибку.

6.5.1.9.1.1.3 Формат выхода:

- Для объектов – берется значение атрибута **vl**, если его нет, то в качестве значения берется **пустой элемент**.
- Для рядов – словарь формата {<ключ>: <значение>}.
    - Для получение конкретного значения ряда в параметрах требуется указать ключ ряда (ключ поля fkey) и значение отдельного элемента, в качестве параметров <ключ>=<значение> соответственно.

6.5.1.9.1.2 Получение целых объектов и рядов

Если вход является массивом, то для получения объектов или рядов можно использовать следующие Методы:

- Для выбора нескольких элементов из входа:
    - **.[where | где]**(<Переменная> => <Условие>) → <Массив>
- Для выбора единичного элемента из входа:
    - **.[first | первый]**(<Переменная> => <Условие>) → [<Объект> | <Ряд>]

Обратите внимание, что при получении элементов из входа, в **Условие** можно использовать не только атрибуты, но и первичные ключи, к примеру:

```
[{"pkey": {"id": 1},...}, {"pkey": {"id": 2},...}].first(i => i['id'] == 2)
```

Извлечет из входа объект с первичным ключем id = 2.

Кроме того, т.к. вход является массивом, то можно получить конкретный элемент массива по его индексу:

- `<Массив>[<индекс>]`

6.5.1.10 Константы

6.5.1.10.1 Вход

Получение глобальных констант. Синтаксис аналогичен получению входов.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Получение константы по идентификатору

```
<[Константа | Const]>{[id | идентификатор] = <идентификатор_константы>}
```

Идентификатор_константы - идентификатор константы.

Получение константы по наименованию

```
<[Константа | Const]>{[name | имя] = <наименование_константы>}
```

Наименование_константы - строковое наименование константы.

Сокращенная нотация: получение константы по его наименованию или идентификатору

```
<Идентификатор_константы/Наименование_константы>
```

6.5.1.10.2 Условия

```
[ if | если ] <Условие> [ then | тогда ] <Правда> (Не обязательно [ else | иначе ] <Ложь>)
```

Условие представляет собой выбор одного из двух элементов: при выполнении условия и при его не выполнении.

Условие, **Правда** и **Ложь** могут быть выражениями, которые будут рассчитаны при необходимости (условие – будет расчитано всегда, правда и ложь – в зависимости от выполенния условия).

В **Условие** используются базовые операции (см. п. 6.5.1.8), возвращающий логический тип данных.

Часть ***[else | иначе]*** допустимо не писать, однако, если условие не выполнено, то вернется **Пустой элемент**.

Актуально для блока «Правила» (см. п. 6.5.5).

В элементы **Правда** и **Ложь**, кроме одного выражения, так же можно добавить функцию msg(). Несмотря на то, что эта фукниция возвращает пустой элемент, здесь он будет проигнорирован.

Формально, в элементах **Правда** и **Ложь** допустимо несколько выражений, однако в качетстве результата выполнения вернется только первое не пустое выражение.

6.5.1.10.3 VAR

```
[ var | имя ] <Переменная> = ( <Выражение> )
var items = ({1: 'a', 2: 'b', 3: 'c'}.values())
```

- `items → ['a', 'b', 'c']`
- `items[0] → 'a'`

Конструкция VAR позволяет записывать какое-угодно выражение в **переменную**, после чего эту переменную можно использовать в других частях выражения.

Конструкция VAR заменила собой устаревшею конструкцию WITH

VAR всегда указывается в самом начале выражения и вычисляется перед расчетом выражения.

Можно использовать несколько конструкций VAR, при этом нижние могут включать в себя переменные из верхних:

```
var elements = ([1, 2, 3, 4, 5])
var number = (100 - 77)
var numberInElement = (elements.contains(number))
```


© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

```
numberInElement -> Ложь
```

6.5.1.10.4 FOR

```
[ for | для ] <Переменная> [ in | в ] <Массив> ( <Выражение> )
for element in [1, 2, 3] (element ** 2) → [1, 4, 9]
```

Цикл FOR позволяет производить вычисления над каждым элементом из массива.

Результатом цикла будет массив выражений, выполненных внутри цикла.

Можно располагать несколько циклов друг в друге:

```
for element in [1, 2, 3] (
 for element2 in [10, 20] (
  element * element2
 )
)

 -> [[10, 20], [20, 40], [30, 60]]
```

6.5.1.10.4.1 Циклы и правила

Актуально для блока «Правила» (см. п. 6.5.5).

Можно располагать правила внутри циклов с разделением через **;** (точку с запятой):

```
[ for | для ] <Переменная> [ in | в ] <Массив> (
 [ if | если ] <Условие> [ then | тогда ] <Правда>;
 [ if | если ] <Условие> [ then | тогда ] <Правда>;
 [ if | если ] <Условие> [ then | тогда ] <Правда>
)
```

Если расчет стоит до первого сработовшего правила, то как только правило сработает в цикле, цикл прерывается и расчет завершается.

6.5.1.11 Методы

Методы – функции, которые вызываются для каких-либо объектов выражения через разделитель «.» (точка).

Обобщенное функцию можно представить как:

```
.<имя>( (Не обязательно <параметры>) )
```

где:

- **имя** – наименование метода;

  Наименование регистрозависимо.

- **параметры** – параметры для выполнения метода. Некоторые методы не требуют дополнительных параметров. Но скобки «()» после наименования метода – обязательны.

Можно размещать несколько методов друг за другом «паравозиком», к примеру:

```
<value>.select().where().orderBy()
```

6.5.1.11.1 Лямбда функции

Лямбда функции (или просто «лямбда») – функция, расположенные в параметрах метода. Используется проведения выражения для каждого из элементов объекта, для которого вызывается метод.

Обобщенное представление выглядит так:

```
<переменная> => <выражение>
```


© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

где:

- **переменная** – идентификатор единичного элемента.
  - Допустимые символы: Aa-Zz, 0-9 и _ (нижнее подчеркивание);
  - Переменная не может начинаться с числа.
- **выражение** – какое-либо выражение, которое будет выполняться для каждого элемента. Чаще всего содержит в себе **переменную**.

6.5.1.11.2 Список методов

- **.[select | выбрать]**(<Переменная> => <Условие> ? <Правда> : <Ложь>) → <Массив>

```
[1, 2, 3, 4].select(i => i < 2 ? -i : i) → [-1, -2, 3, 4]
```

Условие отбора элементов из массива объектов

В **Условие** используются базовые операции (см. п. 6.5.1.8), возвращающий Логический тип данных.

Метод работает только с **Массивами**

- **.[where | где]**(<Переменная> => <Условие>) → <Массив>

```
[1, 2, 3, 4].where(i => i < 2) → [3, 4]
```

Выбор нескольких элементов из массива по условию.

В **Условие** используются базовые операции (см. п. 6.5.1.8), возвращающий Логический тип данных.

Метод работает только с **Массивами**

- **.[first | первый]**(<Переменная> => <Условие>) → <Значение>

```
[1, 2, 3, 4].first(i => i < 2) → 3
```

Выбор первого элемента из массива по условию.

Логически, представляет собой краткую форму записи выражения:

```
.where(<Переменная> => <Условие>)[0]
```

В **Условие** используются базовые операции (см. п. 6.5.1.8), возвращающий Логический тип данных.

Метод работает только с **Массивами**

- **.[orderBy | упорядочитьПо]**([asc | desc], (Не обязательно [key])) → <Массив>

```
[[1, 2, 3, 4].orderBy(desc) → [4, 3, 2, 1]
```

Сортировка элементов массива.

key – флаг, используется, если необходимо отсортировать значения по ключам.

Работает только со **Словарями** или **Рядами**

asc | desc – направление сортировки:

asc – сортировка по возрастанию.

desc – сортировка по убыванию.

- **.[contains | содержит]**(<Элемент>) → <Логический тип>

```
{1: 'a', 2: 'b', 3: 'c'}.containsByKey(1) → Ложь
```

Проверка вхождения **элемента** в значение.

Метод работает только с **Массивами**

- **.[containsByKey | содержитКлюч]**(<Элемент>) → <Логический тип>

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

```
{1: 'a', 2: 'b', 3: 'c'}.containsByKey(1) → Истина
```

Проверка вхождения **элемента** в ключи значений.

Работает только со **Словарями** или **Рядами**

- **.[cumSum | накопительныйИтог]**() → <Массив>

```
[1, 2, 3, 4].cumSum() → [1, 3, 6, 10]
```

Накопительный итог ряда.

Метод работает **Массивами** и **Рядами**

- **.[movAvg | скользящееСреднее]**(<Целое число>) → <Массив>

```
[1, 2, 3, 4].movAvg(3) → [Null, Null, 2, 3]
```

Простое скользящее среднее с указанием сглаживающего интервала.

Интервал – целое положительное число.

Метод работает **Массивами** и **Рядами**

- **.[keys | ключи]**() → <Массив>

```
{1: 'a', 2: 'b', 3: 'c'}.keys() → [1, 2, 3]
```

Возвращает массив ключей.

Для **Объекта** – массив ключей атрибутов.

Для **Ряда** – массив ключей значений.

Метод работает только со **Словарями**, **Объектами** или **Рядами**

- **.[values | значения]**() → <Массив>

```
{1: 'a', 2: 'b', 3: 'c'}.values() → ['a', 'b', 'c']
```

Возвращает массив значений.

Для **Объекта** – массив значений атрибутов.

Для **Ряда** – массив значений.

Метод работает только со **Словарями**, **Объектами** или **Рядами:**

- **.[data | данные]**() → <Словарь>

```
{"pkey": {}, "keys": ['a', 'b', 'c'], "1": [0.1, 0.2, 0.3], "attrs": {},
"_params": {}.data() → {'a': 0.1, 'b': 0.2, 'c': 0.3]}
```

Возвращает словарь с данными ряда.

Метод работает только с **Рядами:**

- **.[min | мин]**() → <Значение>

```
[1, 2, 3, 4].min() → 1
```

Возвращает минимальное значение. Метод аналогичен функции min().

Метод работает только со **Словарями**, **Массивами** или **Рядами:**

- **.[max | макс]**() → <Значение>

```
[1, 2, 3, 4].max() → 4
```

Возвращает максимальное значение. Метод аналогичен функции max().

Метод работает только со **Словарями**, **Массивами** или **Рядами:**

- **.[sum | сумма]**() → <Значение>

```
[1, 2, 3, 4].sum() → 10
```

Возвращает сумму ряда. Метод аналогичен функции sum().

Метод работает только со **Словарями**, **Массивами** или **Рядами:**

- **.[avg | среднее]**() → <Значение>

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

```
[1, 2, 3, 4].avg() → 2.5
```

Возвращает среднее значение. Метод аналогичен функции avg().

Метод работает только со **Словарями**, **Массивами** или **Рядами:**

- **.[median | медиана]**() → <Значение>

```
[1, 2, 2, 4].median() → 2
```

Возвращает медианное значение. Метод аналогичен функции median().

Метод работает только со **Словарями**, **Массивами** или **Рядами:**

- **.[addAttr| добавитьАтрибут]**(<Переменная> => {<ключ>: <значение>,... }) → [<Объект> | <Ряд>]

```
{"pkey": {"id": 1}, "attrs": {"ord": 10, "name": "OBJ"}}.addAttr(i =>
{'doubleOrd': i['ord'] * 2}) → {"pkey": {"id": 1}, "attrs": {"ord": 10,
"name": "OBJ","doubleOrd": 20}}
```

Добавление одного или нескольких атрибутов в объект или ряд.

Переменная позволяет производить операции, специфические для каждого отдельного ряда. Если указать атрибут без переменной, то для всех элементов добавится атрибут с одним и тем же значением.

Ключ является ключем нового атрибута.

Значение является значением этого атрибута. Может содержать в себе переменную.

Результатом метода является **Объект** или **Ряд** (или **Массив** с **Объектами**/**Рядами**) с измененными атрибутами.

Метод работает только с **Объектами** или **Рядами.** Или с **Массивами**, содержащими в себе **Объекты** или **Ряды** (к примеру, вход).

6.5.1.12 Типы данных

6.5.1.12.1 Базовые типы

Название объекта	Формат записи	Описание	Пример
Целое число	<Число>		42
Число с плавающей точкой	<Число>.<Число>		3.1415
Строка	'<Выражение>'		'Я-Строка'
Логический тип	[True | False | Истина | Ложь]		True
Дата	<ДД>.<ММ>.<ГГГГ>	Дата в формате день.месяц.год	01.02.2022
Пустой элемент	[Null | None | Пусто]		None
Комментарий	# <Любое количество элементов до конца строки> или /* <Любое количество элементов> */		# я – однострочный комментарий /* а я – многострочный комментарий */

6.5.1.12.1.1 Операции над базовыми типами

Для базовых типов доступны базовые операции (см. п. 6.5.1.8) блока, за исключением присваивания.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.5.1.12.2 Массив

```
[ <Выражение1>,...,<ВыражениеN> ]
[4, 8.16, '23,' 4+2]
```

Массив представляет собой набор произвольных объектов выражения.

6.5.1.12.2.1 Динамическое создание массива

```
[ <Выражение> [ for | для ] <Переменная> [ in | в ] <Данные> (Не обязательно [ if | если ] <Условие>) ]
[ i + 1 for i in [-1, 1, -2, 2, -3, 3] if i < 0 ] → [0, -1, -2]
```

- **Выражение** и **Условие** могут содержать в себе **Переменную**.
- **Данные** могут быть любым итерируемым типом данных.
- В **Условие** используются базовые операции (см. п. 6.5.1.8), возвращающий Логический тип данных.

6.5.1.12.2.2 Операции над массивами

Для массивов доступны базовые операции (см. п. 6.5.1.8) блока, за исключением присваивания. при условии, что вторым операндом является базовый объект. В таком случае, для каждого элемента массива будет проведена своя операция:

```
[1, 2, 3, 4, 5] - 5 → [-4, -3, -2, -1, 0]
```

Если при выполнении операции, какой-либо элемент выдаст ошибку, то вся операция будет завершена с ошибкой.

Операции над двумя массивами приведут к ошибке.

6.5.1.12.2.3 Получение элемента массива

```
<Массив>[<индекс>]
```

Получение элемента массива происходит по его индексу:

```
[10, 20, 30, 40][0] → 10
```

- Индекс элемента начинается с нуля.
- Индекс является целым числом
- В качестве индекса может использоваться выражение, только при условии, что результатом этого выражения будет целое число:
    - `[10, 20, 30, 40][1+1] → 30`

6.5.1.12.3 Словарь

```
{ <Ключ1>: <Выражение1>,..., <КлючN>: <ВыражениеN> }
{'f': 0, 'b': 'a' + 'r'}
```

Словарь представляет собой неупорядоченный набор произвольных объектов выражения с доступом по ключу.

- **ключом** может быть любой базовый тип выражения;
- **значением** может быть как любой тип выражения, так и какая-либо операция.

6.5.1.12.3.1 Динамическое создание словаря

```
{ <ВыражениеКлюч> : <ВыражениеЗначение> [ for | для ] <Переменная> [ in | в ] <Данные> (Не обязательно [ if | если ] <Условие>) }
{ i: 0 for i in ['a', 'b', 'c'] } → {'a': 0, 'b': 0, 'c': 0}
```

- **ВыражениеКлюч** и **ВыражениеЗначение** и **Условие** могут содержать в себе **Переменную**.
- **Данные** могут быть любым итерируемым типом данных.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

- В **Условие** используются базовые операции (см. п. 6.5.1.8), возвращающий Логический тип данных.

6.5.1.12.3.2 Операции над словарями

```
{'a': 1, 'b': 2, 'c': 3} * 100 → {'a': 100, 'b': 200, 'c': 300}
```

Для словарей доступны базовые операции блока, за исключением присваивания. при условии, что вторым операндом является базовый объект. В таком случае, для каждого значения словаря будет проведена своя операция.

Если при выполнении операции, какой-либо элемент выдаст ошибку, то вся операция будет завершена с ошибкой.

Операции над двумя словарями приведут к ошибке.

6.5.1.12.3.3 Получение элемента словаря

```
<Словарь>[<ключ>]
```

Получение элемента словаря происходит по его ключу:

```
{'a': 100, 'b': 200, 'c': 300}['b'] → 200
```

- В качестве ключа может использоваться выражение:

```
{'aa': 100, 'bb': 200, 'cc': 300}['c' + 'c'] → 300
```

6.5.1.12.4 Объект

Объектом является структура данных формата

Краткое описание объекта

```
{
 "pkey": {<key>: <value>},
 "attrs": {<key>: <value>}
}
```

Представлена краткая форма объекта, указаны только поля, важные для выражения.

где:

- **pkey** – словарь первичных ключей, определяющих объект;
- **attrs** – словарь атрибутов, не обязательных параметрах объекта.

Объекты нельзя создать внутри выражения. Их можно получить только через вход.

6.5.1.12.4.1 Получение значения объекта

См. Вход -> Получение значений входа (см. п. 6.5.1.9.1.1).

Значение объекта хранится в атрибуте **vl**. Если такого атрибута нет, то в качестве значения будет получен **пустой элемент**.

6.5.1.12.4.2 Получение значений атрибутов объекта

```
<Объект>[<ключАтрибута>]
```

Получение значения атрибута объекта происходит по его ключу.

```
{"pkey": {"id": 1}, "attrs": {"ord": 10, "name": "OBJ"}}['name'] → 'OBJ'
```

Получение ключей объекта невозможно.

6.5.1.12.4.3 Изменение значений объекта

Изменение значений объекта производится с помощью операции *присваивания*.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.5.1.12.4.4 Операции над объектами

Для объектов доступны все базовые операции блока, причем они могут выполнятся как и между объектом и базовым объектом (в таком случае логика работы аналогична операциям над словарем), так и между двумя объектами.

Если результатом операции между двумя объектами является новый объект, то первичные ключи для получившегося объекта берется от первого операнда.

Операнды	Описание
Сложение, Вычитание, Умножение, Деление	
Объект + Объект = Объект	Происходит объединение объектов в один, после чего выполняется операция для каждых двух элементов с одинаковыми ключами. Если у одного из объектов нет атрибута, который присутствует в другом ряду, то в качестве значения используется **Пустой элемент**
Сравнение	
Объект + Объект = Объект (все значения имеют **Логический тип**)	Логика аналогична представленному выше, но все получившиеся значения будут иметь **Логический тип**
Инверсия, Конъюнкция, Дизъюнкция	
Объект + Объект = Логический тип	Результатом всегда будет логический тип.

6.5.1.12.4.5 Присваивание

6.5.1.12.4.5.1 Присваивание значения объекта

```
'ВходОбъекты'{<ключ> = <значениеКлюча>} = <значение>
```

Значение объекта сохраняется в атрибут **vl**. Если в исходном объекте этого атрибута не было, то он будет добавлен в момент операции присваивания.

6.5.1.12.4.5.2 Присваивание атрибутов объекта

```
<Объект>[<ключАтрибута>] = <значение>
```

В отличии от присваивания значения, присваивание атрибута можно сделать только при наличии самого атрибута в объекте.

6.5.1.12.4.5.3 Присваивание ключей объекта

Изменять значения ключей объекта нельзя.

6.5.1.12.5 Ряд данных

Рядом является структура данных формата.

Краткое описание ряда:

```
{
 "pkey": {<key>: <value>},
 "keys": [<key1>, <key2>,... <keyN>],
 "vl": [<value1>, <value2>,... <valueN>],
 "attrs": {<key>: <value>},
 "_params": {"fkey": [<labelKey>]}
}
```

Представлена краткая форма ряда, указаны только поля, важные для выражения.

где:

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

- **pkey** – словарь первичных ключей, определяющих ряд;
- **keys** – массив ключей ряда (совпадающий по размерности с массивом значений);
- **vl** – массив значений ряда (совпадающий по размерности с массивом ключей);
- **attrs** – словарь атрибутов, не обязательных параметрах ряда;
- **_params.fkey** – список ключей, формирующий ряд (ключи ряда).

Ряды нельзя создать внутри выражения. Их можно получить только через вход.

6.5.1.12.5.1 Получение значения ряда

См. Вход -> Получение значений входа

6.5.1.12.5.2 Получение значений атрибутов ряда

```
<Ряд>[<ключАтрибута>]
```

Получение значения атрибута ряда происходит по его ключу.

```
{"pkey": {"id": 1}, "keys": [1, 2, 3], мvl": [0.1, 0.2, 0.3], "attrs": {"ord": 25, "name": "ROW"}, "_params": {"fkey": ["num"]}}['name'] → 'ROW'
```

Получение ключей ряда невозможно.

6.5.1.12.5.3 Изменение значений ряда

Изменение значений ряда производится с помощью операции *присваивания*.

6.5.1.12.5.4 Операции над рядами

Для рядов доступны все базовые операции блока, причем они могут выполнятся как и между рядом и базовым объектом (в таком случае логика работы аналогична операциям над массивом), так и между двумя рядами.

Если результатом операции между двумя рядами является новый ряд, то первичные ключи для получившегося ряда берется от первого операнда.

Операнды	Описание
Сложение, Вычитание, Умножение, Деление	
Ряд + Ряд = Ряд	Происходит объединение рядов в один ряд (при этом так же происходит и сортировка получившегося ряда по ключам), после чего выполняется операция для каждых двух элементов ряда с одинаковыми ключами. Если у одного из рядов нет значения для конкретного ключа, то в качестве значения используется **Пустой элемент**
Сравнение	
Ряд + Ряд = Ряд (все значения имеют **Логический тип**)	Логика аналогична представленному выше, но все получившиеся значения будут иметь **Логический тип**
Инверсия, Конъюнкция, Дизъюнкция	
Ряд + Ряд = Логический тип	Результатом всегда будет логический тип

6.5.1.12.5.5 Присваивание

6.5.1.12.5.5.1 Присваивание значения ряда

6.5.1.12.5.5.1.1 Присваивание одного значения ряда

```
'ВходРяды'{<ключ> = <значениеКлюча>, <ключРяда> = <значениеКлючаРяда>} = <значение>
```

Происходит присваивание значения одному элементу из массива значений ряда.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.5.1.12.5.5.2 Присваивание всего значения ряда

```
'ВходРяды'{<ключ> = <значениеКлюча>} = {<ключ>: <значение>}
```

Происходит изменение всех значений и ключей ряда.

В качестве значения должен выступать массив. Тогда ключи массива станут новыми ключами ряда, а значения – значениями ряда соответственно.

6.5.1.12.5.5.3 Присваивание атрибутов ряда

```
<Ряд>[<ключАтрибута>] = <значение>
```

6.5.1.12.5.5.4 Присваивание ключей ряда

Изменять значения ключей ряда нельзя.

6.5.1.12.6 Функции

Функция – подпрограмма в выражении, выполняющая какую-либо операцию и возвращающая результат.

Обобщенно функцию можно представить так:

```
<имя>(<параметры>)
```

где:

- **имя** – наименование функции;

  Наименование функции регистрозависимо.

- **параметры** – параметры для выполнения функции.

6.5.1.12.6.1 Встроенные

Встроенные функции – функции, взятые напрямую из Python. Ниже приведены некоторые из функций.

Об остальных функциях можно узнать из официальной документации: https://docs.python.org/3.7/library/functions.html

Однако, в блоке заблокированы некоторые из функций, которые невозможно корректно выполнить в коде блока. К примеру, заблокирована функция **__import__**

```
all(<Массив>) → <Логический тип>
all([Истина, Истина, Ложь, Истина]) → Ложь
```

Возвращает <Истина> если все элементы массива являются Истиной.

Функция работает только с **Массивами**

- ```
  any(<Массив>) → <Логический тип>
  any([Истина, Истина, Ложь, Истина]) → Истина
  ```

Возвращает <Истина> если любой элемент массива является Истиной.

Функция работает только с Массивами

- ```
  len(<Массив>) → <Целое число>
  len([Истина, Истина, Ложь, Истина]) → 4
  ```

Возвращает количество элементов массива.

Функция работает только с **Массивами**

6.5.1.12.6.2 Математические

Список функций: https://docs.python.org/3.7/library/math.html

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Математические функции – функции, взятые напрямую из Python, использующиеся в математических расчетах. Ниже приведены некоторые из функций.

Об остальных функциях можно узнать из официальной документации: https://docs.python.org/3.7/library/math.html

Все математические функции работают с **Целыми** или **Вещественными** числами.

- **exp**([<Целое число> | <Вещественное число>]) → <Вещественное число>

Функция расчитывает экспоненту в указанной степени – e^x

- **log**([<Целое число> | <Вещественное число>], (Не обязательное [<Целое число> | <Вещественное число>])) → <Вещественное число>

Функция расчитывает логарифм первого числа по основанию второго числа. Если второй параметр не указан, вычисляется натуральный логарифм.

- **cos**([<Целое число> | <Вещественное число>]) → <Вещественное число>

Функция расчитывает косинус числа. В радианах.

- **acos**([<Целое число> | <Вещественное число>]) → <Вещественное число>

Функция расчитывает арккосинус числа. В радианах.

Кроме того, добавлены следующие функции, работающие с **Массивами** или **Рядами**:

- **[diff | АБС]**([<Массив> | <Ряд>]) → <Массив>

Функция расчитывает абсолютный прирост ряда.

- **[ratio | РОСТ]**([<Массив> | <Ряд>]) → <Массив>

Функция расчитывает темп роста ряда.

- **[pch | ПРСТ]**([<Массив> | <Ряд>]) → <Массив>

Функция расчитывает темп прироста ряда.

- **[ln | ЛОГ]**([<Массив> | <Ряд>]) → <Массив>

Функция осуществляет логарифмирование ряда.

- **[dlog | ЛОГПРСТ]**([<Массив> | <Ряд>]) → <Массив>

Функция расчитывает логарифмический прирост ряда.

- **[norm | НОРМ]**([<Массив> | <Ряд>]) → <Массив>

Функция осуществляет нормализацию ряда.

- **[std | СТД]**([<Массив> | <Ряд>]) → <Массив>

Функция осуществляет стандартизацию ряда.

- **[pchYtD | ГодПРСТ]**([<Ряд>]) → <Массив>

Функция расчитывает годовой темп прирост временного ряда.

Функция работает только с временными рядами, т.е. ключ ряда должен быть датой.

Период должен быть не меньше месяца.

6.5.1.12.6.3 Статистические

- **[min | МИН]**([<Массив> | <Ряд> | <Словарь> | <Объект>])

Функция расчитывает минимальное значение. Формат результата зависит от формата элементов внутри массива.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Для **Объекта** – высчитывается минимальное значение среди атрибутов.

- **[minByKey | МИНПоКлючу]**([<Ряд> | <Объект>])

Функция расчитывает минимальное значение ключей ряда. Формат результата зависит от формата элементов внутри массива.

Для **Объекта** – высчитывается минимальное значение среди ключей атрибутов.

- **[max | МАКС]**([<Массив> | <Ряд> | <Словарь> | <Объект>])

Функция расчитывает максимальное значение. Формат результата зависит от формата элементов внутри массива.

Для **Объекта** – высчитывается максимальное значение среди атрибутов.

- **[maxByKey | МАКСПоКлючу]**([<Ряд> | <Объект>])

Функция расчитывает максимальное значение ключей ряда. Формат результата зависит от формата элементов внутри массива.

Для **Объекта** – высчитывается максимальное значение среди ключей атрибутов.

- **[avg | СРД]**([<Массив> | <Ряд> | <Словарь> | <Объект>])

Функция расчитывает среднее арифметическое значение. Формат результата зависит от формата элементов внутри массива.

Для **Объекта** – высчитывается среднее арифметическое среди атрибутов.

- **[avgByKey | СРДПоКлючу]**([<Ряд> | <Объект>])

Функция расчитывает среднее арифметическое значение ключей ряда. Формат результата зависит от формата элементов внутри массива.

Для **Объекта** – высчитывается среднее арифметическое среди ключей атрибутов.

- **[median | МЕДИАНА]**([<Массив> | <Ряд> | <Словарь> | <Объект>])

Функция расчитывает медианное значение. Формат результата зависит от формата элементов внутри массива.

Для **Объекта** – высчитывается медиана среди атрибутов.

- **[medianByKey | МЕДИАНАПоКлючу]**([<Ряд> | <Объект>])

Функция расчитывает медианное значение ключей ряда. Формат результата зависит от формата элементов внутри массива.

Для **Объекта** – высчитывается медиана среди ключей атрибутов.

6.5.1.12.6.4 Разное

Функции, не вошедшие в предыдущие пункты.

- **[msg | сообщить]**(<Выражение>) → <Пустой элемент>

```
msg('Пример выражения')
```

Вычисляет **Выражение** внутри себя и передает его на выход «Рекомендации». Актуально для блока «Правила» (см. п. 6.5.5).

Для целостности выражений, в качестве результата вычисления функция возвращает **Пустой элемент**.

(Любая операция с этой функцией будет считаться операцией с пустым элементом).

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Получение константы из списка Глобальных констант в приложении.

6.5.2 Блок «Конвертор объектов в ряды»

6.5.2.1 Назначение

Блок позволяет преобразовать массив объектов в массив временных рядов.

При преобразовании объектов в ряды, происходит выборка элементов по первичному ключу, без учета поля **fkey**. В этот же момент находятся все значения поля **fkey** – длинна этого массива и будет длиной ряда.

Если количество объектов с уникальным первичным ключем больше, чем длина массива, значит первичный ключ не обеспечивает уникальности записей и ряд сформирован не будет (не расчитается блок), и расчетчик отобразит ошибку:

Ошибка блока

```
"Обнаружено дублирование уникальных ключей! Первичный ключ недостаточен."
```

Все не уникальные атрибуты ряда будут «схлопнуты» из массива в единичный элемент. Остальные, уникальные атрибуты, будут переданы в качестве массива длиной равной длине ряда.

6.5.2.2 Настройки блока

Название параметра	Тип параметра	Описание	Пример
Название блока	Строка	Наименование блока	

Параметры привязки	JSON	Для построения ряда требуется указать одно поле из первичных ключей, по которому и будет собран ряд.
		Если это поле так же прописано в параметрах, как поле даты/времени (заполнен раздел params.ts) то получившийся ряд будет временным. Иначе – простой ряд
		Код привязок
		{
		"fkey": [
		<Поле из pkey, по которому строится ряд. Наличие квадратных скобок обязательно (массив из одного элемента)>
		]
		"vl": <Идентификатор атрибута, отвечающий за значение>
		}	

Ссылка на документацию	Строка		
Описание блока	Строка		
Пропуск расчета блока	Флажок	Рассчитывать или нет данный блок	

6.5.2.3 Входы

- **Объекты**

Массив объектов см. Блок «SQL – запрос-справочник» (п. 6.2.11).

6.5.2.4 Выходы

- **Ряды данных**

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Выход из блока – массив рядов стандартизированного формата.

Выход из блока

```
{
 "pkey": [
  <Набор первичных ключей>
 ],
 "vl": [
  <Массив значений ряда>
 ],
 "attrs": [
  <Набор атрибутов – полей, не попавших ни в первичный ключ, ни в значение.>
 ],
 "_params": {
  <Опциональные параметры объекта>,
  "vl": <Ключ поля, отвечающего за значение>,
  "key": [
   <Ключ поля, по которому собран ряд>
  ]
 },
 "_source": <Идентификатор блока-источника, поле заполняется калькулятором автоматически>,
}
```

6.5.3 Блок «Коннектор временных рядов»

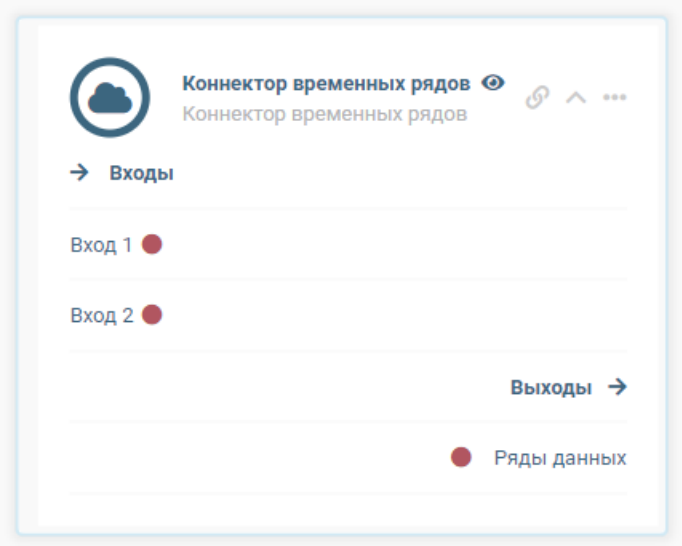


Рисунок 366 – Пример блока «Коннектор временных рядов»

6.5.3.1 Назначение

Блок позволяет собирать наборы временных рядов из отдельных временных рядов.

6.5.3.2 Настройки блока

Название параметра	Тип параметра	Обязательный	Описание	Пример
Количество входов	Целое число	Да	Задается необходимое количество рядов данных.	2

6.5.3.3 Группа «Общие настройки блока»

Аналогично приведенным в п. 6.1.1.1.3.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.5.3.4 Входы

Название входа	Тип	Обязательный	Описание
Вход 1	Ряд данных	Да	
Вход 2	Ряд данных	Да	
Вход 3	Ряд данных	Нет	Если в настройках поставить количество входов больше 2

6.5.3.5 Выходы

Название выхода	Тип	Обязательный	Описание
Ряды данных	Ряд данных	Да	

6.5.3.6 События блока

В блоке нет событий.

6.5.3.7 Описание работы блока

Блок объединяет в один массив рядов все ряды, которые были переданы на его входы, для дальнейшего его использования.

Количество входов блока определяется в его настройках в поле «Количество входов» (минимальное количество входов два и эти два входа обязательны, остальные входы будут не обязательными). После того как пользователь сохраняет настройки блока, у блока сразу обновляется список входов блока. На каждый вход может передаваться один ряд или же массив рядов. После выполнения блока в результате будет массив рядов (если на входы переданы пустые данные, то результатом блока будет пустой массив). Механизм работы блока: в выходной массив добавляет данные из входов.

Для продвинутых пользователей

Блок можно использовать для соединения данных, поступающих с блоков с событиями. При расчете, если в блок входит событийная связь – блок не будет рассчитан, т.к. событие не произошло (в случае, если нет события по умолчанию) – это может привести к ситуации, когда граф не считается в режиме отладки, а только при действиях пользователя. В этом случае можно поставить этот блок сразу после события – результат будет сохранен на его выходах и будет применяться при расчете графа, как если бы блок считался обычными способом.

6.5.3.8 Описание работы визуализатора

Нет.

6.5.3.9 Дополнительно

Нет.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.5.4 Блок «Обрезка рядов»

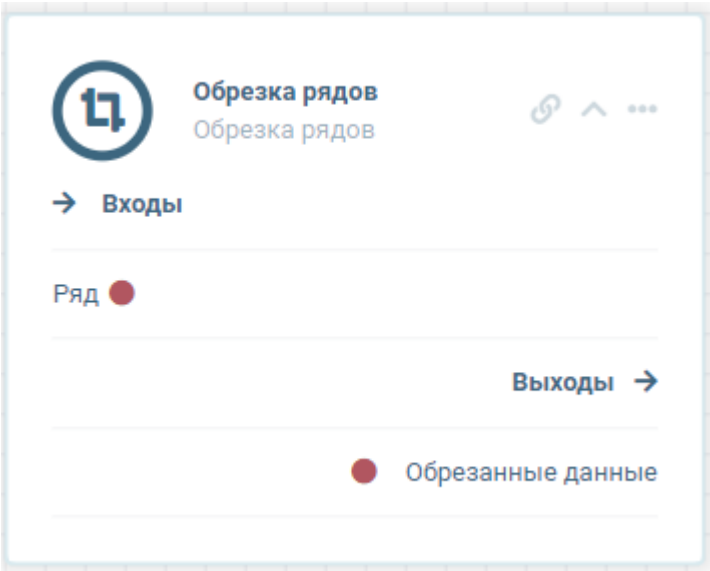


Рисунок 367 – Пример блока «Обрезка рядов»

6.5.4.1 Назначение

Блок позволяет корректировать размер входных временных рядов.

6.5.4.2 Настройки блока

Название параметра	Тип параметра	Описание	Пример
Группа «Основные свойства»			
Наименование блока	Строка	По умолчанию «Обрезка рядов»	
Группа «Структурный тип данных»			
Спецификация	Выпадающий список	Поле поддерживает единичную отметку. Определяет каким образом будут заданы периоды моделирования и прогнозирования. по умолчанию выбран «Индексы». Содержит следующие элементы: • Индексы • Даты	
Группа «Период моделирования»			
Первая точка / Начальная дата	Целое число / Datatime		
Последняя точка / Последняя дата	Целое число / Datatime		
Группа «Общие настройки блока»			
Поле таблицы с настройками	Строка		
Ссылка на документацию	Строка		
Описание блока	Строка		
Пропуск расчета блока	Флажок		

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.5.4.3 Входы

Название входа	Тип	Обязательный
Ряд	Временной ряд	Да

6.5.4.4 Выходы

Обрезанные данные	Временной ряд

6.5.4.5 Дополнительно

Описание особенностей:

- Блок работает только с временными рядами.

Описание ограничений:

- Значение поля «Последняя точка» должно быть больше поля «Первая точка».
- Значение поля «Последняя дата» должно быть больше поля «Начальная дата».

6.5.5 Блок «Правила»

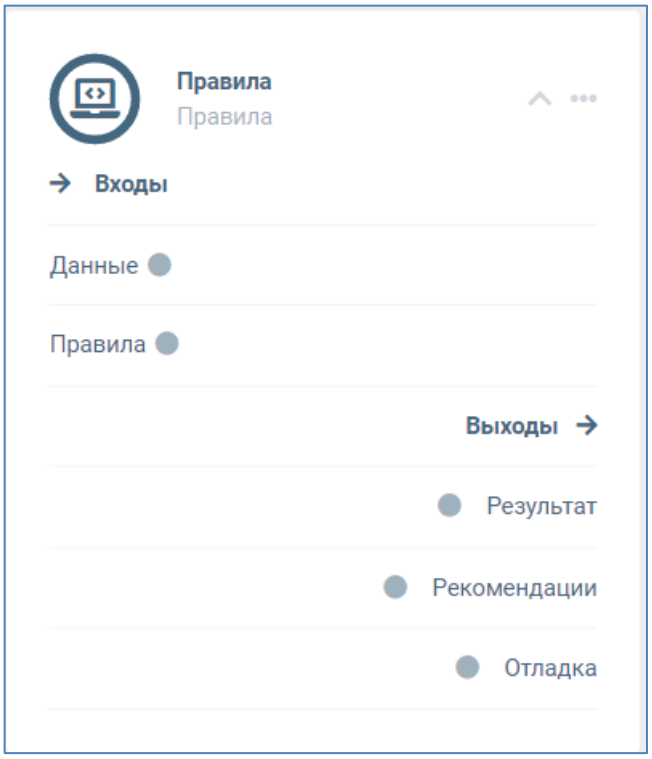


Рисунок 368 – Пример блока «Правила»

6.5.5.1 Назначение

Блок предназначен для выполнения правил, написанных с использованием выражений.

6.5.5.2 Настройки блока

Название параметра	Тип параметра	Обязательный	Описание	Пример
Правила	Ввод выражения	Да	Поле ввода кода правил. Правила разделяются между собой с помощью **;** (точки с запятой)	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
Количество знаков после запятой	Целое число	Нет	Настройка количества знаков после запятой для вещественного результата выражения	
Расчет	Выпадающий список	Да	Методика расчета правил. Возможные значения: • **До первого сработавшего правила** – расчет завершается на первом сработавшем правиле. • **Все правила** – один проход по правилам, от первого до последнего. **Пока срабатывает хотя бы одно правило** – расчет завершается, когда ни одно правило не выолняется, или не будет достигнут предел итераций	
Максимум циклов в итерации	Целое число	Нет	Доступно, только если в параметре «Расчет» установлено «Пока не сработает хотя бы одно правило». Количество итераций, после которых расчет правил прерывается	

6.5.5.3 Группа «Общие настройки блока»

Аналогично приведенным в п. 6.1.1.1.3.

6.5.5.4 Входы

Название входа	Тип	Описание

Правила	JSON	Массив, содержащий в себе правила.
		Пример массива
		`[`
		` {"rule": "<Правило 1>"},`
		` {"rule": "<Правило 2>"}`
		`]`
		Если на данном входе есть значения, то они будут применяться вместо правил из настройки «Правила»

Кроме того, доступно создание пользовательских входов.

Для создания входа необходимо нажать на знак + в разделе входов блока. Для каждого из входов задается следующий перечень полей:

Название входа	Тип	Обязательный	Описание
Идентификатор	Строка	Да	Идентификатор входа

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название входа	Тип	Обязательный	Описание
Имя	Строка	Да	Имя входа для отображения в списке входов блока В отличии от других блоков, используется для получения доступа к данным входа
Описание	Строка	Нет	Дополнительная информация с описанием входа
Тип	Выпадающий список	Да	Тип данных, которые будут приходить на вход. Нужен для построения корректных связей с другими блоками. По умолчанию установлен тип "JSON".
Обязательный	Флаг	Да	Обязательность входа. Если вход обязательный и на него не пришло данных – блок не будет рассчитан с ошибкой
Массив	Флаг	Да	Является ли вход массивом данных

6.5.5.5 Выходы

Название выхода	Тип	Описание
Результат	Ряды	Ряды данных, преобразованные после правил. Тип выхода зависит от типа входа. Если на вход были поданы объекты, то выход будет аналогичным
Рекомендации	JSON	Массив текстовых рекомендаций Добавляться будут только уникальные рекомендации (если текст есть в массиве рекомендаций, он будет проигнорирован). Для добавления текста в рекомендации используется функция msg() (см. Функции → Функции → Разное)

6.5.5.6 События блока

В блоке нет событий.

6.5.5.7 Описание работы блока

Блок считывает код, написанный в поле **Правила**, после чего последовательно выполняет все обнаруженные правила.

В отличии от блока «Выражение» (см. п. 6.5.1), корректная запись блока «Правила» должна включать в себя одно или несколько условий (if...else) разделенных ; (точка с запятой) и опционально, несколько конструкций with:

```
[ with | пусть ] <Переменная> [ as | как ] ( <Выражение> )
[ with | пусть ] <Переменная> [ as | как ] ( <Выражение> )

[ if | если ] <Условие> [ then | тогда ] <Правда>;
[ if | если ] <Условие> [ then | тогда ] <Правда>;
[ if | если ] <Условие> [ then | тогда ] <Правда> (Не обязательно [ else | иначе
] <Ложь>);
```

Правило сработает, если условие пойдет по пути <Правда> (см. Конструкции → Условия).

Все правила без условий if, никогда не будут считаться выполненными.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Условия, написанные в конструкции with не являются правилами.

Результатом работы блока (для выхода «Результат») является одномерный массив всех входов, задействованных в блоке.

6.5.5.8 Использование шаблонизатора

В блоке присутствует шаблонизатор Jinja2, связанный со входом «Данные».

Таким образом, выражения могут быть дополнены шаблонами. Шаблоны будут преобразованы в значения до выполнения правил, что предотвращает ошибки, связанные с некорректным синтаксисом.

6.5.5.9 Дополнительные ссылки

https://jinja.palletsprojects.com/en/3.0.x/

6.5.6 Блок «Преобразование рядов»

6.5.6.1 Назначение

Блок позволяет преобразовывать временные ряды для их дальнейшего анализа.

6.5.6.2 Настройки блока

Название параметра	Тип параметра	Описание	Пример	Название параметра
Группа «Основные свойства»				
Наименование блока	Строка	По умолчанию «Преобразование рядов»		Наименование блока
Группа «Специфические настройки блока»				
Отбор	Флажок	По умолчанию True		Отбор
Тип	Выпадающий список	Отвечает за то, к чему относится данный ряд (к фактору или целевой переменной). Поле поддерживает единичную отметку. По умолчанию выбран «Фактор (X)». Содержит следующие элементы: • Целевая переменная (Y). • Фактор (X)		Тип

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Описание	Пример	Название параметра
Преобразование	Выпадающий список	Поле поддерживает единичную отметку. По умолчанию выбран «Без преобразования». Содержит следующие элементы: • Без преобразования • Абсолютный прирост • Темп роста • Темп прироста • Логарифмирование ряда • Логарифмический прирост • Нормализация • Стандартизация • Годовой темп прироста		Преобразование
Тип сдвига	Выпадающий список	Поле поддерживает единичную отметку. По умолчанию выбран «Без сдвига». Содержит следующие элементы: • Без сдвига • Сдвиг влево • Сдвиг вправо		Тип сдвига
Лаг	Целое число	По умолчанию 0		Лаг
Группа «Общие настройки блока»				
Поле таблицы с настройками	Строка			Поле таблицы с настройками
Ссылка на документацию	Строка			Ссылка на документацию
Описание блока	Строка			Описание блока
Пропуск расчета блока	Флажок			Пропуск расчета блока

6.5.6.3 Входы

Название входа	Тип	Описание
Ряды	Временной ряд	Да

6.5.6.4 Выходы

Название выхода	Тип	Описание
Y	Временной ряд	Заполняется если в выпадающем списке Тип выбран элемент Целевая переменная (Y)
X	Временной ряд	Заполняется если в выпадающем списке Тип выбран элемент Фактор (X)

6.5.6.5 Дополнительно

Особенности блока:

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

- Блок работает только с временными рядами.

Ограничения:

- Если поле «Лаг» равно «0», то сдвига не произойдет.
- Значение поля «Лаг» должно быть больше или равно «0».
- Преобразование «Годовой темп прироста» работает только для годовой, квартальной и месячной динамики.

Описание блока в t_library


```
{
 "name": "Преобразование рядов",
 "desc": "Преобразование рядов",
 "version": "1.0.0",
 "package": "vmstatistic.transform",
 "module": "compute",
 "visualizer": 1,
 "changeshowinapp": 1,
 "settings": [
  {
   "name": "Отбор",
   "id": "_active",
   "desc": "",
   "type": "bool",
   "mandatory": 0,
   "efault": 1,
   "changeshowinapp": 1
  },
  {
   "name": "Тип",
   "id": "type",
   "desc": "",
   "type": "enum",
   "ultiselect": 0,
   "efault": 2,
   "mandatory": 0,
   "changeshowinapp": 1,
   "enum": [
    {
     "val": 1,
     "name": "Целевая переменная (Y)",
     "desc": ""
    },
    {
     "val": 2,
     "name": "Фактор (X)",
     "desc": ""
    }
   ]
  },
  {
   "name": "Преобразование",
   "id": "transform",
   "desc": "",
```


© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

```
 "type": "enum",
 "multiselect": 0,
 "default": 1,
 "mandatory": 0,
 "changeshowinapp": 1,
 "enum": [
  {
   "val": 0,
   "desc": "",
   "name": "Без преобразования"
  },
  {
   "val": 1,
   "desc": "",
   "name": "Абсолютный прирост"
  },
  {
   "val": 2,
   "desc": "",
   "name": "Темп роста"
  },
  {
   "val": 3,
   "desc": "",
   "name": "Темп прироста"
  },
  {
   "val": 6,
   "desc": "",
   "name": "Логарифмирование ряда"
  },
  {
   "val": 7,
   "desc": "",
   "name": "Логарифмический прирост"
  },
  {
   "val": 8,
   "desc": "",
   "name": "Нормализация"
  },
  {
   "val": 9,
   "desc": "",
   "name": "Стандартизация"
  },
  {
   "val": 10,
   "desc": "",
   "name": "Годовой темп прироста"
  }
 ]
},
{
 "id": "lag_y",
```


© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

```
   "desc": "",
   "name": "Настройки сдвига",
   "type": "json",
   "mandatory": 0,
   "elements": [
    {
     "id": "type_y",
     "desc": "",
     "enum": [
      {
       "val": 0,
       "desc": "",
       "name": "Без сдвига"
      },
      {
       "val": 4,
       "desc": "",
       "name": "Сдвиг влево"
      },
      {
       "val": 5,
       "desc": "",
       "name": "Сдвиг вправо"
      }
     ],
     "name": "Тип",
     "type": "enum",
     "default": 4,
     "mandatory": 0,
     "multiselect": 0,
     "changeshowinapp": 1
    },
    {
     "id": "lag",
     "val": 0,
     "desc": "",
     "name": "Лаг",
     "type": "int",
     "default": 0,
     "mandatory": 0,
     "changeshowinapp": 1
    }
   ]
  }
 ],
 "inputs": [
  {
   "name": "Ряды",
   "id": "Y",
   "type": "series",
   "andatory": 1,
   "desc": "",
   "array": 1
  }
 ],
```


© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

```
 "outputs": [
  {
   "name": "Y",
   "id": "Y_n",
   "desc": "",
   "type": "series",
   "mandatory": 1,
   "array": 0
  },
  {
   "name": "X",
   "id": "X",
   "desc": "",
   "type": "series",
   "mandatory": 1,
   "array": 1
  }
 ],
 "style": {
  "icon": "fas fa-calculator"
 }
}
```

6.5.7 Блок «Срез»

6.5.7.1 Назначение

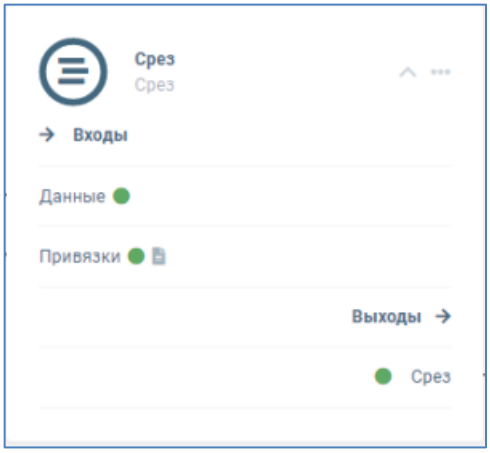


Рисунок 369 – Пример блока «Срез»

Блок позволяет сформировать табличные данные для отображения в табличной области данных.

6.5.7.2 Настройки блока

Название параметра	Тип параметра	Обязательный	Описание	Пример
Привязки	JSON	Да	Параметры связи заголовка и атрибутов таблицы с входными данными. Полностью аналогично параметру «Привязки» блока «Таблица» (п. 6.1.2.3)	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
Количество знаков после запятой	Целое число	Да	Настройка количества знаков после запятой у данных, которые отобразятся в таблице	
Разделитель групп разрядов	Строка	Нет	Настройка разделителя групп разрядов. Может использоваться «,» или «»	

6.5.7.3 Группа «Общие настройки блока»

Аналогично приведенным в п. 6.1.1.1.3.

6.5.7.4 Входы

Название входа	Тип	Обязательный	Описание
Данные	Ряд данных	Да	Массив данных с типом «Ряд данных» или «Объект»
Привязки	Строка	Нет	Связи заголовка и атрибутов с данными, которые приходят на вход блока используются вместо параметра настроек «Привязки». Полностью аналогично параметру «Привязки» блока «Таблица» (п. 6.1.2.3). При наличии данных на этом входе, настройка «Привязки» не будет учитываться во время формирования таблицы

6.5.7.5 Выходы

Название выхода	Тип	Описание
Срез	JSON	Данные таблицы, представленные в специальном формате для отображения в табличной области данных

6.5.7.6 События блока

В блоке нет событий.

6.5.7.7 Описание работы блока

Блок преобразует ряды или объекты, поступившие на вход в табличные данные, по аналогии с блоком «Таблица» (п. 6.1.2.3). Однако на выходе представляет визуализатор, а выдает специализированный табличный JSON (похожий на формат JExcel, но с некоторыми изменениями для возможности формирования таблиц сложной структуры).

Блок «Табличная область данных» (п. 6.1.2.4) принимает этот формат данных и преобразует его в визуализатор.

Пример выходных данных блока

```
{
    "type": "sliceData",
    "data": [
        [
            1
        ]
    ],
```


© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

```
    "columns": [
        {
            "type": "text",
            "title": "1 элемент"
        }
    ],
    "meta": [
        [
            {
                "id": "a938f6ff-29b2-4be2-a785-c3f64e208fed",
                "attrs": {
                    "letters_name": "A",
                    "count_name": "1 элемент"
                },
                "fields": [
                    {
                        "name": "letters",
                        "vl": 1
                    }
                ],
                "fieldset": {
                    "fields": [
                        {
                            "name": "count",
                            "vl": 1
                        }
                    ]
                }
            }
        ]
    ]
}
```


- **type** – тип данных. Для всех данных используется тип silceData;
- **data** – массив с данными по строкам (массив массивов);
- **columns** – массив с описанием всех заголовков таблицы:
  - **type** – всегда text;
  - **title** – описание заголовка;
- **meta** – массив массивов с метаданными каждой отдельной ячейки таблицы:
  - **id** – идентификатор блока «Срез», который обработал данную ячейку;
  - **attrs** – массив с атрибутами конкректного значения;
  - **fields** – массив с первичными ключами (**pkey**) для конкретной ячейки таблицы:
    - **name** – наименование ключа;
    - **vl** – значение ключа;
  - **fieldstet** – массив с ключами формирования ряда (**fkey**) для конкретной ячейки таблицы:
    - **name** – наименование ключа;
    - **vl** – значение ключа.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.5.8 Блок «Формирование кросс-таблицы»

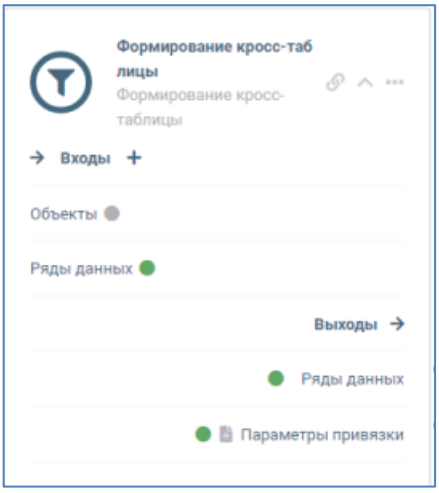


Рисунок 370 – Пример блока «Формирование кросс-таблицы»

6.5.8.1 Назначение

Блок позволяет преобразовать ряды данных к нужному виду, без использования визуализатора.

По логике работы блок аналогичен блоку «Панель OLAP» (см. п. 6.1.4.8), однако данные редактируются один раз, в момент расчета блока, и дополнительный визуализатор не создается.

6.5.8.2 Настройки блока

Название параметра	Тип параметра	Обязательный	Описание	Пример
Параметры привязки	JSON	Да	Параметры сопоставления входного массива данных, для выходных данных. Возможные значения настроек: • **name** – текстовое наименование справочника; • **key** – первичный ключ, отвечающий за формирование справочника; • **name_key** - идентификатор атрибута, отвечающий за наименование элемента; • **hierarchy** – флаг наличия иерархии в выбранном справочнике;	Параметры привязки [{ "name": "Регионы", "key": "region_id", "name_key": "region_name", "rows": "True" }, { "name": "Даты", "key": "data", "columns": "True" }, { "name": "Показатели", "key": "pok" }, {

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример

			• **rows** – флаг для отображения справочника по строкам. Не использовать вместе с флагом **columns**;
			• **columns** – флаг для отображения справочника по столбцам. Не использовать вместе с флагом **rows**;
			• **lazy** – флаг, отвечающий за «ленивую загрузку» данных справочника
				<pre>"name": "Источник данных", "key": "source" }
				]</pre>

6.5.8.3 Группа «Общие настройки блока»

Аналогично приведенным в п. 6.1.1.1.3.

6.5.8.4 Входы

Название входа	Тип	Обязательный	Описание
Объекты	Объект	Нет	Массив с объектами
Ряды данных	Ряд данных	Нет	Массив с рядами данных

Если будут задействованы оба входа, то приоритет (т.е. те данные, которые будет использовать блок) будет у входа «Объекты».

6.5.8.5 Динамические входы

Для подключения выпадающих списков в атрибутах (см. настройку «Привязки») необходимо добавить входы для справочников.

Каждый новый справочник добавляется в отдельный вход. Параметры «Обязательный» и «Массив» должны быть включены. Параметр «Идентификатор» необходимо указать в поле **input_key** в параметрах привязки.

Для создания входа необходимо нажать на знак + в разделе входов блока. Для каждого из входов задается следующий перечень полей

Название входа	Тип	Обязательный	Описание
Идентификатор	Строка	Да	Идентификатор входа – используется внутри блока, для получения доступа к данным входа
Имя	Строка	Да	Имя входа для отображения в списке входов блока
Описание	Строка	Нет	Дополнительная информация с описанием входа

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название входа	Тип	Обязательный	Описание
Тип	Выпадающий список	Да	Тип данных, которые будут приходить на вход. Нужен для построения корректных связей с другими блоками. По умолчанию установлен тип "Объект".
Обязательный	Флаг	Да	Обязательность входа. Если вход обязательный и на него не пришло данных – блок не будет рассчитан с ошибкой
Массив	Флаг	Да	Является ли вход массивом данных

6.5.8.6 Выходы

Название выхода	Тип	Описание
Событие смены выделения		
Ряды данных	Ряд данных	Массив отформатированных рядов данных Используется для динамического обновления визуализаторов, при изменении отметки в таблице
Параметры привязки	Строка	Динамические параметры привязки для корректного отображения блоков-визуализаторов. Для корректного отображения визуализаторов, необходимо чтобы этот параметр был присоединен ко всем блокам-визуализатором (у которых есть аналогичный вход), находящимся после этого блока

6.5.8.7 События блока

У блока нет событий.

6.5.8.8 Описание работы блока

Блок формирует данные и параметры отображения визуализаторов согласно настройкам, заданным в поле «**Параметры привязки**».

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.6 Статистические методы

6.6.1 Блок «ARIMA»

6.6.1.1 Назначение

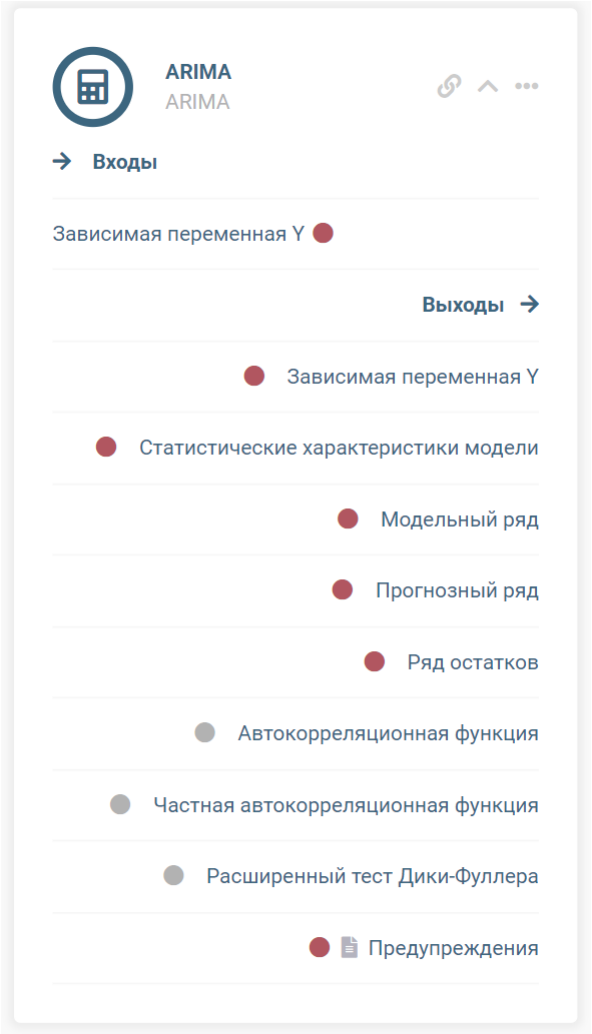


Рисунок 371 – Пример блока «ARIMA»

Блок позволяет рассчитать безфакторную модель ARIMA.

6.6.1.2 Системное описание

6.6.1.3 Специфические пункты меню

Специфических пунктов меню нет. Полное описание меню приведено в описании блока «Меню» (см. п. 6.1.4.7).

6.6.1.4 Настройки блока

Наименование	Тип данных	Примечание
Основные свойства		
Наименование блока	Строка	Наименование, которое будет иметь блок в графе
Структурный тип данных		
Спецификация	Выпадающий список	Поле поддерживает единичную отметку. Определяет каким образом будут заданы периоды моделирования и прогнозирования.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Наименование	Тип данных	Примечание
		по умолчанию выбран «Индексы». Содержит следующие элементы: • Индексы; • Даты
Период моделирования		
Первая точка/Начальная дата	Целое число / Datatime	Первая точка/Начальная дата должна быть меньше Последней точки/Последней даты
Последняя точка/Последняя дата	Целое число / Datatime	
Период прогнозирования		
Первая точка/Начальная дата	Целое число / Datatime	Первая точка/Начальная дата должна быть меньше Последней точки/Последней даты
Последняя точка/Последняя дата	Целое число / Datatime	
Автоопределение ретро	Флаг	
Специфические настройки блока		
Порядок дифференцирования	Целое число	
Порядки ARIMA	Выпадающий список	Поле поддерживает единичную отметку. По умолчанию выбран «Максимальные порядки ARIMA». Содержит следующие элементы: • Максимальные порядки ARIMA; • Специфические порядки ARIMA
Максимальный/Специфический порядок авторегрессии	Целое число / Строка	
Максимальный/Специфический порядок скользящего среднего	Целое число / Строка	
Компонент тренда	Выпадающий список	Поле поддерживает единичную отметку. По умолчанию выбран «Без тренда». Содержит следующие элементы: • Без тренда; • Константа; • Временной ряд. Предполагается псевдо временной ряд из натуральных чисел; • Константа и временной ряд
Методика оценки	Выпадающий список	Поле поддерживает единичную отметку. По умолчанию выбран «Pv.statsmodels». Содержит следующие элементы: • Pv.statsmodels. Расчет будет производиться в среде Python; • R.forecast. Расчет будет производиться в среде R.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Наименование	Тип данных	Примечание
Обработка пропусков		
Обработка пропусков	Выпадающий список	• По умолчанию выбран «Удаление пропусков». Поле поддерживает единичную отметку; • Удаление пропусков; • Средним по выборке; • Заданным значением; • Максимальным значением; • Минимальным значением; • Линейная интерполяция; • Квадратичная интерполяция; • Кубическая интерполяция
Значение	Вещественное число	Значение появляется только при выборе метода «Заданным значением»
Вывод значений	Выпадающий список	По умолчанию выбран «Выводить исходные ряды». Поле поддерживает единичную отметку. Содержит следующие элементы: • Выводить исходные ряды. Будут выведены ряды, в том виде, в котором они были введены в метод; • Выводить обработанные ряды. Будут выведены ряды с заполненными или удаленными точками
Общие настройки блока		
Поле таблицы с настройками	Строка	Устаревший параметр
Ссылка на документацию	Строка	Ссылка, добавленная в это поле будет доступна из графа при нажатии на символ 🔗 в поле блока.
Описание блока	Строка	Текст может содержать краткое описание блока
Пропуск расчета блока	Флаг	В случае если флаг включен – блок будет исключен из цепочки расчета и на графу будет отображен серым цветом
Кэширование результатов расчета блока	Флаг	Флаг предназначен для блоков карты
Предпросмотр	Флаг	Если блок связан с какими-то визуализаторами, то при включенном предпросмотре расчет отобразиться в специальном окне, которое будет содержать присоединенные визуализаторы. Механизм позволяет

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Наименование	Тип данных	Примечание
		корректировать настройки блока до основного расчета

6.6.1.4.1 Входы блока

Название входа	Тип	Описание
Зависимая переменная Y	Временной ряд	Обязательная переменная. Атрибуты входного ряда должны содержать поле ts_name

6.6.1.4.2 Выходы блока

Название выхода	Тип	Описание
Зависимая переменная Y	Временной ряд	Ряд имеет model_name «Факт». Ряд можно вывести в таблицу или на диаграмму
Статистические характеристики модели	Структура (html)	Структура содержит общие характеристики модели. (см. Блок «Статистические характеристики» п. 6.1.4.17)
Уравнение зависимости с оцененными характеристики	Строка	Содержит уравнение модели. Можно вывести в блок «html-шаблон"
Коэффициенты модели и из статистические характеристики	Таблица	Можно вывести в блок «Таблица» (п. 6.1.2.3)
Модельный ряд	Временной ряд	Ряд имеет model_name «Модель». Ряд можно вывести в таблицу или на диаграмму
Прогнозный ряд	Временной ряд	Ряд имеет model_name «Прогноз». Ряд можно вывести в таблицу или на диаграмму
Ряд остатков	Временной ряд	Ряд имеет model_name «Остатки». Ряд можно вывести в таблицу или на диаграмму
Автокорреляционная функция	Массив	Ряд значений автокорреляционной функции для лагов
Частная автокорреляционная функция	Массив	Ряд значений частной автокорреляционной функции для лагов
Расширенный тест Дики-Фуллера	Строка	Содержит статистику и вероятность метода. Можно вывести в блок «html-шаблон»
Предупреждения	Строка	Можно вывести в блок «html-шаблон»
Зависимая переменная Y	Временной ряд	Ряд имеет model_name «Факт». Ряд можно вывести в таблицу или на диаграмму

6.6.1.5 События блока

В блоке нет событий.

6.6.2 Описание работы блока

6.6.2.1 Общая теория блока

Общее представление интегрированной модели авторегрессии – скользящего среднего:

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

$$\triangle^d X_t = c + \sum_{i=1}^{p} a_i \triangle^d X_{t-i} + \sum_{j=1}^{q} b_j \varepsilon_{t-j} + \varepsilon_t$$

где ε_t – стационарный временной ряд;

c, a_i, b_j – параметры модели.

$\triangle^d$ – оператор разности временного ряда порядка d (последовательное взятие d раз разностей первого порядка — сначала от временного ряда, затем от полученных разностей первого порядка, затем от второго порядка и т. д.).

Метод предназначен для прогнозирования временных рядов. Прогнозирование производиться на основании методологии Бокса-Дженкинса – первую очередь оценивается стационарность ряда. Различными тестами выявляются наличие единичных корней (в нашем случае тестом Дики-Фуллера) и порядок интегрированности временного ряда (обычно ограничиваются первым или вторым порядком). Далее при необходимости (если порядок интегрированности больше нуля) ряд преобразуется взятием разности соответствующего порядка и уже для преобразованной модели строится некоторая ARMA-модель.

Для определения лагов авторегрессии и скользящего среднего используется автокорреляционная и частная автокорреляционная функция.

6.6.2.1.1 Ограничения метода

Описание особенностей:

- Блок работает только с временным рядом (в ряде должно быть поле ts_name, в атрибутах или же в первичном ключе).
- Если ряд не нужно дифференцировать необходимо в поле «Порядок дифференцирования» задать «0».
- При задании максимальных порядков ARIMA будут оцениваться все лаги от 1 до заданного числа. Например если задано 4, то оценятся 1, 2, 3 и 4 лаги.
- При задании специфических порядков ARIMA будут оцениваться только заданные лаги.
- Специфические порядки задаются через «,» или «-» если необходимо задать лаговый период. Например, если задано 1-3, 5,7, то оценятся лаги 1, 2, 3, 5, 7.
- В Python в моделях содержащих авторегрессионную составляющую const это коэффициент при константном тренде, а не константа модели. Для того чтобы рассчитать константу модели, необходимо воспользоваться формулой:

$$b_0 = \frac{const}{\left(1 - \sum_{i=1}^{p} b_{AR_i}\right)}$$

- Значения коэффициентов могут расходится с другими пакетам. Разница, вероятно, связана с ошибкой округления и точностью сходимости оптимизационных методов.
- Автокорреляционная и частная автокорреляционная функции рассчитываются только для оригинального ряда.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

- Текст Дики Фуллера на единичный корни рассчитывается только для оригинального ряда.

Описание ограничений:

- Последняя точка моделирования должна быть больше первой.
- Последняя точка прогнозирования должна быть больше первой.
- Ряд должен быть не константным.
- Максимальный порядок AR из рассчитываемых должен быть меньше количества точек в ряде.
- Максимальный порядок MA из рассчитываемых должен быть в 3 раза меньше количества точек в ряде.

6.6.2.1.2 Возможность анализа ряда

Мастер анализа временного ряда для блока «Арима» (см. п. 6.6.1).

6.6.2.1.3 Возможность автоподбора максимальных/специфических порядков

Описание работы анализа временного ряда для блока «Арима» приведено в п. 6.6.1.

6.6.2.1.4 Дополнительные ссылки

Модель ARIMA

Теоретическая информация об используемых функциях: ARIMA

6.6.2.2 ARIMA – моделирование временных рядов

Теоретическая справка

ARIMA модели – модель авторегрессии с интегрированием и скользящим средним. Обычно обозначаются так: ARIMA(p, d, q), где коэффициент p – порядок авторегрессии; d – количество дифференцирований исходного временного ряда до достижения его стационарности; q – порядок части скользящего среднего.

Коэффициент p показывает зависимость значения нынешнего значения от прошлых значений p-периодов назад. Коэффициент q показывает зависимость значения нынешнего периода от ошибок предсказания предыдущих q периодов.

Данная модель позволяет составить прогноз, зависящий от текущего и прошлого значения зависящей переменной и от текущих и прошлых значений величины случайного возмущения.

Примечание. Модель ARIMA является суммой модели авторегрессии AR(p) и скользящего среднего MA(q).

Реализация блока средствами платформы ЦП УвП

Добавить блок «ARIMA» на рабочую область создания графа можно двумя способами:

- Найти блок «ARIMA» в библиотеке в модуле «Статистические методы».
- Найти блок «ARIMA» через поиск в библиотеке.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

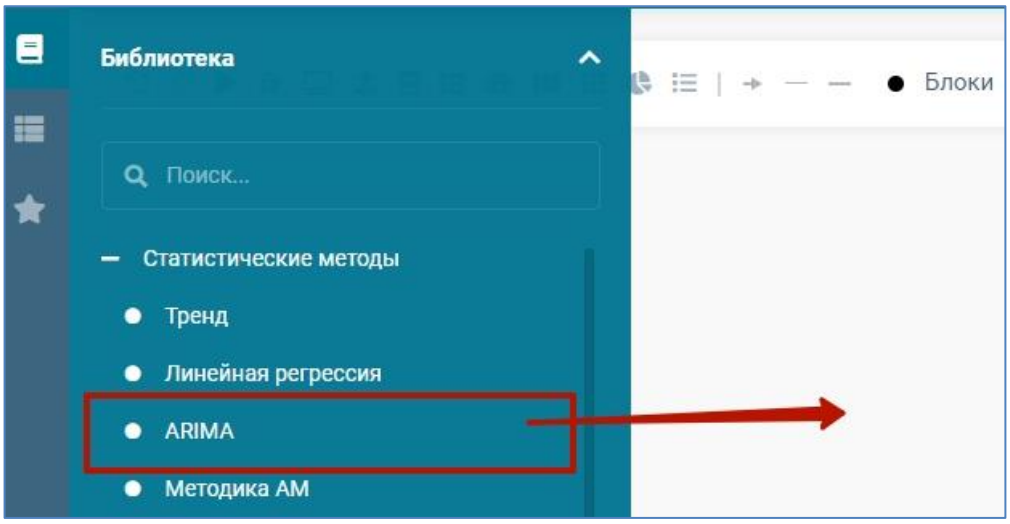


Рисунок 372 – Пример добавления блока «ARIMA»

Зажать блок ЛКМ и перетащить на рабочую область

Начальный вид блока «ARIMA»:

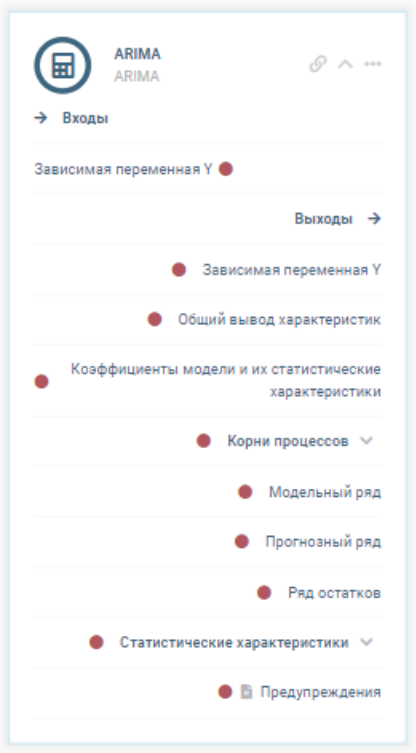


Рисунок 373 – Начальный вид блока «ARIMA»

Для дальнейшей работа необходимо передать во входной параметр блока «Зависимая переменная Y» временной ряд.

При добавлении корректной связи цветовой индикатор входа блока окрасится зеленым цветом.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

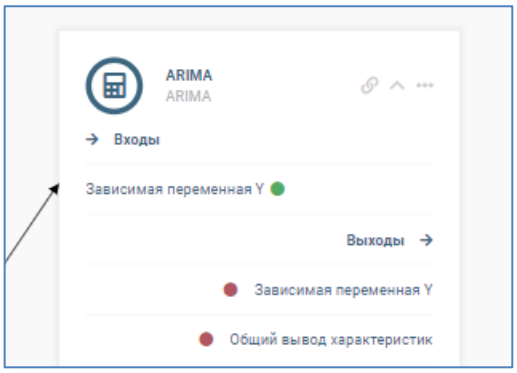


Рисунок 374 – Пример цветового индикатора входа блока

Для открытия настроек блока необходимо два раза щелкнуть по нему ЛКМ. Имеется возможность проскроллировать раскрытый блок для просмотра и настройки всех параметров модели.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Открытый блок настроек приведен на рисунке:

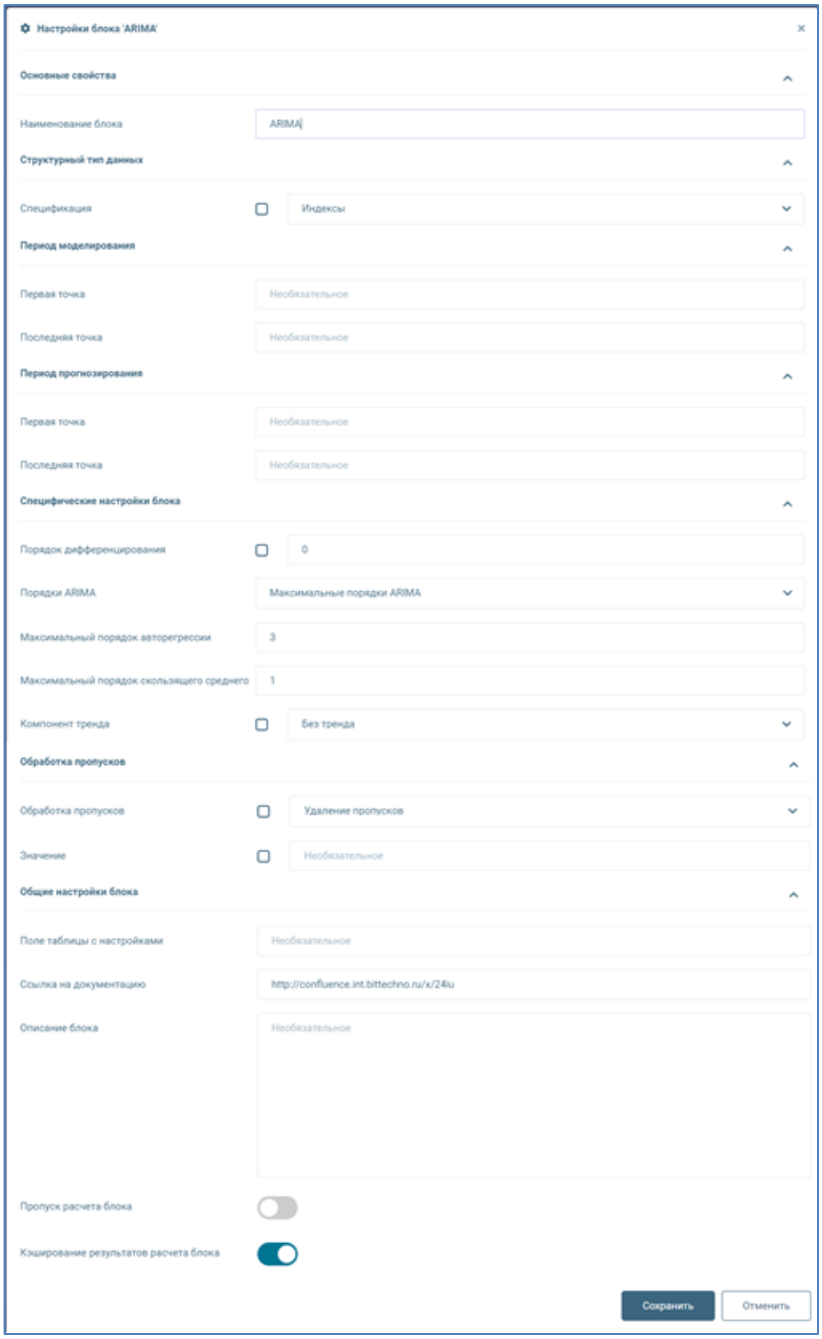


Рисунок 375 – Пример настроек блока «ARIMA»

Внутри настроек блока «ARIMA» обязательно задать структурный тип данных, период моделирования, период прогнозирования, задать специфические настройки блока и обработку пропусков.

Структурный тип данных

Спецификация:

- Индексы (данные за промежутки времени, даты которых неизвестны);
- Даты (данные за конкретные промежутки времени, известны даты).

Период моделирования – период, за который присутствуют данные (фомат записи даты: dd.mm.yyyy).

Период прогнозирования – период, на который необходим прогноз.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Примечание. В какой динамике были данные за период моделирования (по месяцам, по кварталам, по годам), в такой динамике будут и прогнозные значения (по месяцам, по кварталам, по годам соответственно).

Специфические настройки блока

1.Порядок дифференцирования:

Если временной ряд стационарен – устанавливаем порядок «0»;

Если временной ряд не стационарен – методом перебора определяем порядок дифференцирования (сколько раз необходимо взять разности ряда, чтобы ряд стал стационарным).

Примечание. Обычно порядок дифференцирования 1 или 2.

2. Порядки ARIMA:

- Максимальные порядки: модель будет построена от 1-го до указанного порядка.
- Специфические порядки: модель будет построена по указанным порядкам.

Пример. Модель с **максимальными** порядками ARIMA(3,0,2) будет выглядеть следующим образом: AR(1)+AR(2)+AR(3)+MA(1)+MA(2).
Модель со **специфическими** порядками ARIMA(3,0,2) будет выглядеть следующим образом: AR(3)+MA(2).

3. Компонент тренда:

- Без тренда (к модели ничего не добавляется).
- Константа (к модели добавляется число, некая случайная величина).
- Временной ряд (к модели добавляется линейный тренд вида a*x[t], где коэффициенты оцениваются внутри модели).
- Константа и временной ряд (к модели добавляется и случайная величина, и линейный тренд).

После настройки всех параметров модели необходимо нажать на кнопку «Сохранить».

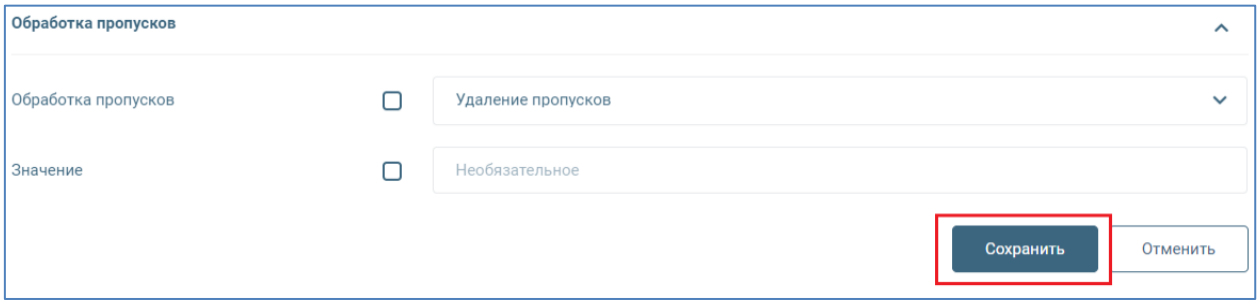


Рисунок 376 – Пример кнопки «Сохранить»

Для расчета модели необходимо сначала нажать на значок ••• для открытия контекстного меню в правом верхнем углу блока, затем нажать кнопку «Рассчитать блок».

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

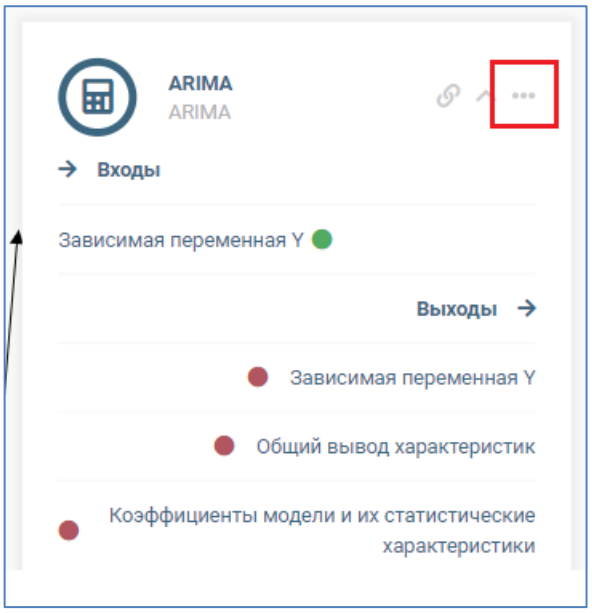


Рисунок 377 – Пример кнопки открытия контекстного меню

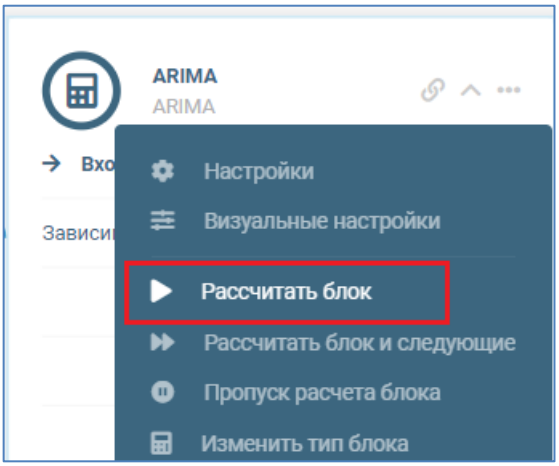


Рисунок 378 – Пример кнопки «Рассчитать блок»

6.6.3 Блок «K-Means»

6.6.3.1 Назначение

Блок позволяет кластеризовать объекты данных методом k-means.

6.6.3.2 Настройки блока

Название параметра	Тип параметра	Описание	Пример
Группа «Основные свойства»			
Наименование блока	Строка	По умолчанию «K-Means»	
Группа «Специфические настройки блока»			
Количество кластеров	Целое число	По умолчанию 8	
Максимальное число итераций	Целое число	По умолчанию 300	
Количество раз вычисления алгоритма с разными центрами	Целое число	По умолчанию 10	
Группа «Общие настройки блока»			
Ссылка на документацию	Строка		

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Описание	Пример
Описание блока	Строка		
Пропуск расчета блока	Флажок		

6.6.3.3 Входы

Название входа	Тип	Описание
Набор обучающих объектов	Временной ряд	Да
Набор тестовых объектов	Временной ряд	Нет

6.6.3.4 Выходы

Название выхода	Тип	Описание
Распределение объектов обучающей выборки	Ряд	
Сумма квадратов расстояний	Структура (html)	
Распределение тестовой выборки	Ряд	
Центры кластеров	Ряд	

6.6.3.5 Дополнительно

Описание особенностей:

- Формат обоих входов:
    - Тип данных series.
    - Ряд должен быть описан хотя бы одной координатой (значением).
    - В ряде должно быть поле ts_name, в атрибутах или же в первичном ключе.

Пример входных данных:

```
 [
    {
        "pkey": {
            "id_region": 146
        },
        "attrs": {
            "ts_name": "Республика Татарстан",
            "id_pok": [
                5,
                4,
                2,
                1,
                3
            ]
        },
        "keys": [
            "Ввод в действие жилых домов  в % к соответстующему периоду предыдущего года",
            "Ввод в действие жилых домов, кв.м.",
            мВеличина прожиточного минимума в % к предыдущему году",
            "Внешнеторговый оборот в % к предыдущему году",
            "Производство скота и птицы на убой в живом весе в % к соответствующему периоду предыдущего года"
        ],
```


© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

```
        "vl": [
            100.0,
            150.0,
            99.0,
            104.0,
            75.0
        ],
        "_source": "aa61ce77-e180-4b7f-a224-f2f7112b7d23",
        "_params": {
            "vl": "value",
            "fkey": [
                "name_ind"
            ]
        }
    },
    {
        "pkey": {
            "id_region": 68
        },
        "attrs": {
            "ts_name": "Республика Башкортостан",
            "id_pok": [
                5,
                4,
                2,
                1,
                3
            ]
        },
        "keys": [
            "Ввод в действие жилых домов  в % к соответстующему периоду
предыдущего года",
            "Ввод в действие жилых домов, кв.м.",
            "Величина прожиточного минимума в % к предыдущему году",
            "Внешнеторговый оборот в % к предыдущему году",
            "Производство скота и птицы на убой в живом весе в % к
соответствующему периоду предыдущего года"
        ],
        "vl": [
            38.86, 40.57, 40.03, 38.42, 42.88
        ],
        "_source": "aa61ce77-e180-4b7f-a224-f2f7112b7d23",
        "_params": {
            "vl": "value",
            "fkey": [
                "name_ind"
            ]
        }
    },
    {
        "pkey": {
            "id_region": 1461
        },
        "attrs": {
            "ts_name": "Республика Татарстан (тест)",
```


© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

```
            "id_pok": [
                5,
                4,
                2,
                1,
                3
            ]
        },
        "keys": [
            "Ввод в действие жилых домов  в % к соответстующему периоду
предыдущего года",
            "Ввод в действие жилых домов, кв.м.",
            "Величина прожиточного минимума в % к предыдущему году",
            "Внешнеторговый оборот в % к предыдущему году",
            "Производство скота и птицы на убой в живом весе в % к
соответствующему периоду предыдущего года"
        ],
        "vl": [
            170.09, 171.15, 188.72, 186.85, 207.12
        ],
        "_source": "aa61ce77-e180-4b7f-a224-f2f7112b7d23",
        "_params": {
            "vl": "value",
            "fkey": [
                "name_ind"
            ]
        }
    },
    {
        "pkey": {
            "id_region": 681
        },
       "attrs": {
            "ts_name": "Республика Башкортостан (тест)",
            "id_pok": [
                5,
                4,
                2,
                1,
                3
            ]
        },
        "keys": [
            "Ввод в действие жилых домов  в % к соответстующему периоду
предыдущего года",
            "Ввод в действие жилых домов, кв.м.",
            "Величина прожиточного минимума в % к предыдущему году",
            "Внешнеторговый оборот в % к предыдущему году",
            "Производство скота и птицы на убой в живом весе в % к
соответствующему периоду предыдущего года"
        ],
        "vl": [
            98.29, 94.96, 98.34, 108.23, 121.39
        ],
        "_source": "aa61ce77-e180-4b7f-a224-f2f7112b7d23",
```


© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

```
        "_params": {
            "vl": "value",
            "fkey": [
"name_ind"
            ]
        }
    }
]
```

Ограничения блока;

- Количество кластеров должно быть меньше количества рядов, переданных во вход «Набор обучающих объектов».
- Ряд объектов «Набор тестовых объектов» должен иметь такую же структуру как и ряд объектов «Набор обучающих объектов».

Пример блока запроса для формирования данных для блока «K-Means»:

- Запрос SQL:

```
select * from (

select 'Region 1' as ts_name, 'Indicator 1' as name_ind, 1 as id_region, 1 as
id_pok, round((random()*100)::numeric, 2)::float as value union

select 'Region 1' as ts_name, 'Indicator 2' as name_ind, 1 as id_region, 2 as
id_pok, round((random()*100)::numeric, 2)::float as value union

select 'Region 2' as ts_name, 'Indicator 1' as name_ind, 2 as id_region, 1 as
id_pok, round((random()*100)::numeric, 2)::float as value union

select 'Region 2' as ts_name, 'Indicator 2' as name_ind, 2 as id_region, 2 as
id_pok, round((random()*100)::numeric, 2)::float as value union

select 'Region 3' as ts_name, 'Indicator 1' as name_ind, 3 as id_region, 1 as
id_pok, round((random()*100)::numeric, 2)::float as value union

select 'Region 3' as ts_name, 'Indicator 2' as name_ind, 3 as id_region, 2 as
id_pok, round((random()*100)::numeric, 2)::float as value union

select 'Region 4' as ts_name, 'Indicator 1' as name_ind, 4 as id_region, 1 as
id_pok, round((random()*100)::numeric, 2)::float as value union

select 'Region 4' as ts_name, 'Indicator 2' as name_ind, 4 as id_region, 2 as
id_pok, round((random()*100)::numeric, 2)::float as value union

select 'Region 5' as ts_name, 'Indicator 1' as name_ind, 5 as id_region, 1 as
id_pok, round((random()*100)::numeric, 2)::float as value union

select 'Region 5' as ts_name, 'Indicator 2' as name_ind, 5 as id_region, 2 as
id_pok, round((random()*100)::numeric, 2)::float as value union

select 'Region 6' as ts_name, 'Indicator 1' as name_ind, 6 as id_region, 1 as
id_pok, round((random()*100)::numeric, 2)::float as value union

select 'Region 6' as ts_name, 'Indicator 2' as name_ind, 6 as id_region, 2 as
id_pok, round((random()*100)::numeric, 2)::float as value union

select 'Region 7' as ts_name, 'Indicator 1' as name_ind, 7 as id_region, 1 as
id_pok, round((random()*100)::numeric, 2)::float as value union

select 'Region 7' as ts_name, 'Indicator 2' as name_ind, 7 as id_region, 2 as
id_pok, round((random()*100)::numeric, 2)::float as value union

select 'Region 8' as ts_name, 'Indicator 1' as name_ind, 8 as id_region, 1 as
id_pok, round((random()*100)::numeric, 2)::float as value union

select 'Region 8' as ts_name, 'Indicator 2' as name_ind, 8 as id_region, 2 as
id_pok, round((random()*100)::numeric, 2)::float as value union
```


© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

```
select 'Region 9' as ts_name, 'Indicator 1' as name_ind, 9 as id_region, 1 as
id_pok, round((random()*100)::numeric, 2)::float as value union

select 'Region 9' as ts_name, 'Indicator 2' as name_ind, 9 as id_region, 2 as
id_pok, round((random()*100)::numeric, 2)::float as value

) t

order by 1,2
```

- Параметры привязки запроса:

```
{
    "pkey": [
        "id_region", "name_ind"
    ],
    "vl": «value"
}
```

- Параметры привязки для рядов:

```
{
    "fkey": [
        "name_ind"
    ]
}
```

- Распределение объектов выводятся в виде таблиц. Сам распределение кластера вынесено в атрибуты, и чтобы его увидеть в таблице нужно сделать привязку:

```
{
    "label": "",
    "name": "ts_name",
    "attrs": [
        {
            "key": "group",
            "label": "Группа",
            "type": "string",
            "editable": true,
            "visible": true,
            "style-default": {
                "stroke-color": "rgb(0,128,0,0.8)",
                "stroke-width": 6,
                "fill-color": "rgb(0,128,0,0.5)"
            },
            "style-conditions": [
                {
                    "condition": "Группа 1",
                    "style": {
                        "stroke-color": "rgb(200,35,44,1)",
                        "stroke-width": 6,
                        "fill-color": "rgb(200,35,44,0.7)"
                    }
                },
                {
                    "condition": "Группа 2",
                    "style": {
                        "stroke-color": "rgb(203,165,36,1)",
                        "stroke-width": 6,
                        "fill-color": "rgb(203,165,36,0.7)"
                    }
```


© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

```
                },
                {
                    "condition": "Группа 3",
                    "style": {
                        "stroke-color": "rgb(49,41,141,1)",
                        "stroke-width": 6,
                        "fill-color": "rgb(49,41,141,0.7)"
                    }
                },
                {
                    "condition": "Группа 4",
                    "style": {
                        "stroke-color": "rgb(42,166,29,1)",
                        "stroke-width": 6,
                        "fill-color": "rgb(42,166,29,0.7)"
                    }
                },
                {
                    "condition": "Группа 5",
                    "style": {
                        "stroke-color": "rgb(140,9,15,1)",
                        "troke-width": 6,
                        "fill-color": "rgb(140,9,15,0.7)"
                    }
                },
                {
                    "condition": "Группа 6",
                    "style": {
                        "stroke-color": "rgb(143,112,9,1)",
                        "stroke-width": 6,
                        "fill-color": "rgb(143,112,9,0.7)"
                    }
                },
                {
                    "condition": "Группа 7",
                    "style": {
                        "stroke-color": "rgb(25,19,99,1)",
                        "stroke-width": 6,
                        "fill-color": "rgb(25,19,99,0.7)"
                    }
                },
                {
                    "condition": "Группа 8",
                    "style": {
                        "stroke-color": "rgb(18,117,7,1)",
                        "stroke-width": 6,
                        "fill-color": "rgb(18,117,7,0.7)"
                    }
                }
            ]
        }
    ]
}
```


© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.6.3.6 Дополнительные ссылки

Общая теория

6.6.4 Блок «Автоподбор линейной регрессии»

6.6.4.1 Назначение

Блок позволяет подобрать лучшую модель линейной регрессии на заданных независимых переменных.

6.6.4.2 Настройки блока

Название параметра	Тип параметра	Описание	Пример
Группа «Основные свойства»			
Наименование блока	Строка	По умолчанию «Линейная регрессия»	
Группа «Специфические настройки блока»			
Критерий отбора	Выпадающий список	Поле поддерживает единичную отметку. По умолчанию выбран «Информационный критерий Акаика». Содержит следующие элементы: • Информационный критерий Акаика. • Информационный критерий Байеса.	
Включать ли в подбираемые модели константу	Флажок		
Ограничения на положительность/отрицательность коэффициентов	Строка	По умолчанию пуст. Ограничения можно задавать входным рядом или строкой. Приоритет у входного ряда	
Количество знаков после запятой	Целое число	По умолчанию 6	
Группа «Общие настройки блока»			
Ссылка на документацию	Строка		
Описание блока	Строка		
Пропуск расчета блока	Флажок		
Останавливать расчет на блоке	Флажок		
Отображать процесс выполнения	Флажок		

6.6.4.3 Входы

Название входа	Тип	Обязательный	Описание
Зависимая переменная Y	Временной ряд	Да	Ряд для которого необходимо подобрать лучшую модель
Независимые переменные X. Факт	Набор временных рядов	Да	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название входа	Тип	Обязательный	Описание
Независимые переменные X. Прогноз	Набор временных рядов	Нет	

Ограничения коэффициентов	Временной ряд	Нет	На вход подается тип series: параметры привязки запроса
			{
			"pkey": [
			"ts_name"
			],
			"vl": "value",
			"params": {
			}
			}
			параметры привязки для рядов
			{
			"fkey": [
			"ts_name"
			]
			}

6.6.4.4 Выходы

Название выхода	Тип	Описание
Зависимая переменная Y	Временной ряд	Временной ряд для которого была рассчитана модель тренда. Если в ряде были пропуски – то пропуски восстановлены или заполнены в зависимости от выбранного метода обработки пропусков
Уравнение модели	Строка (json)	Содержит уравнение лучшей модели
Модель регрессии	Строка (json)	Содержит уравнение лучшей модели с рассчитанными коэффициентами
Массив временных рядов	Набор временных рядов	
Список уравнений модели	Строка (json)	все уравнения (а так же их коэффициент, и отдельный список X) которые посчитала модель
Массив прогнозных рядов	Набор временных рядов	Период прогнозирования для выбранных факторов
Коэффициенты значимости факторов	Временной ряд.	Процент значимости каждой переменной
Предупреждение	Строка (json)	Сообщения метода в случае каких-то проблем с расчетом

6.6.4.5 Дополнительно

Описание особенностей:

- Ограничения на положительность и отрицательность коэффициентов в строке задаются через запятую, знаками «+» или «-». Если ограничения на коэффициент нет, то между запятыми должно быть пусто. Например, если задано 5 независимых

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

переменных и третья из них должна быть отрицательной, а первая положительной – то ограничения будут выглядеть следующим образом: +,,-,,

- Метод может выбирать модель только из **30 независимых переменных**.
- В ряде должно быть поле ts_name, в атрибутах или же в первичном ключе.

Описание ограничений:

- Набор независимых переменных **может содержать не более 30 рядов**.
- Если ограничения заданы – то они должны быть заданы для каждой независимой переменной.

Примечание

В R базовый расчет логарифма функции максимального правдоподобия производится с ошибкой (степени свободы содержат лишнюю единицу), следовательно все информационный критерии рассчитаны не корректно. В Python и во всех остальных методах критерии рассчитаны верно.

Описание блока в t_library

```
{
 "name": "Автоподбор линейной регрессии",
 "id": "4cf68542-6337-463a-ae3f-b7527e0d24eb",
 "desc": "Автоподбор факторов для линейной регрессии",
 "version": "1.0.0",
 "package": "vmstatistic.linear_regression",
 "module": "auto_factor",
 "visualizer": 0,
 "settings": [
  {
   "name": "Критерий отбора",
   "id": "crit",
   "desc": "",
   "type": "enum",
   "multiselect": 0,
   "default": 1,
   "mandatory": 1,
   "enum": [
    {
     "val": 1,
     "name": "Информационный критерий Акаика",
     "desc": ""
    },
    {
     "val": 2,
     "name": "Информационный критерий Байеса",
     "desc": ""
    },
    {
     "val": 3,
     "name": "Скорректированный информационный критерий Акаик",
     "desc": ""
    }
   ]
  },
```


© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

```
  {
   "name": "Включать ли в подбираемые модели константу",
   "id": "intercept",
   "type": "bool",
   "desc": "",
   "mandatory": 1,
   "default": 1
  },
  {
   "name": "Ограничения на положительность/отрицательность коэффициентов",
   "id": "limitation",
   "type": "string",
   "desc": "",
   ммandatory": 0
  }
 ],
 "inputs": [
  {
   "name": "Зависимая переменная Y",
   "id": "Y",
   "type": "series",
   "mandatory": 1,
   "desc": "Входной временной ряд",
   "array": 0
  },
  {
   "name": "Ряды независимых переменных X",
   "id": "X",
   "type": "series",
   "mandatory": 1,
   "desc": "Входной временной ряд",
   "array": 1
  }
 ],
 "outputs": [
  {
   "name": "Уравнение модели",
   "id": "equation",
   "desc": "",
   "type": "string",
   "mandatory": 1,
   "array": 0
  },
  {
   "name": "Модель регрессии",
   "id": "formula",
   "desc": "",
   "type": "string",
   "mandatory": 1,
   "array": 0
  },
  {
   "name": "Массив временных рядов",
   "id": "fittedvalues",
   "desc": "",
```


© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

```
   "type": "series",
   "mandatory": 1,
   "array": 0
  }
 ],
 "style": {
  "icon": "fas fa-calculator"
 }
}
```

6.6.4.6 Дополнительные ссылки

Основополагающая функция

6.6.5 Блок «Адаптивный МНК (AOLS)»

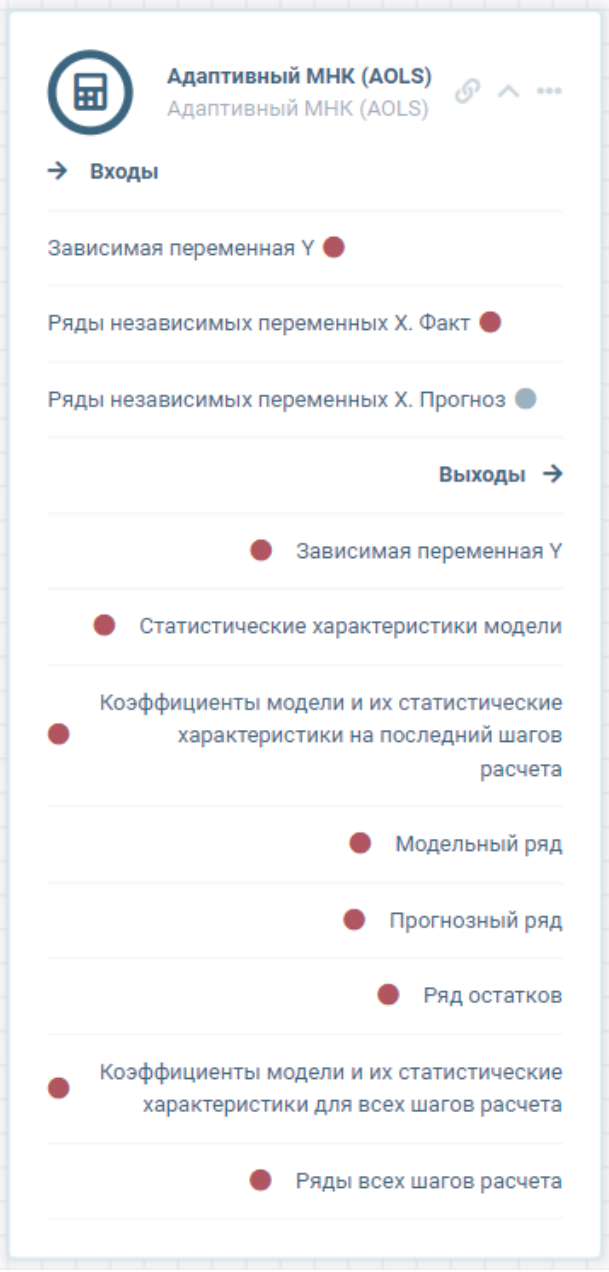


Рисунок 379 – Пример блока «Адаптивный МНК (AOLS)»

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.6.5.1 Назначение

Блок позволяет рассчитать адаптивный МНК. Адаптивный МНК является усложненным методом рекурсивного МНК, способного тонко чувствовать тенденции анализируемого ряда.

6.6.5.2 Настройки блока

Название параметра	Тип параметра	Описание	Пример
Группа «Основные свойства»			
Наименование блока	Строка	По умолчанию «Адаптивный МНК (AOLS)»	
Группа «Структурный тип данных»			
Спецификация	Выпадающий список	Поле поддерживает единичную отметку. Определяет каким образом будут заданы периоды моделирования и прогнозирования. по умолчанию выбран «Индексы». Содержит следующие элементы: • Индексы • Даты	
Группа «Период моделирования»			
Первая точка / Начальная дата	Целое число / Datatime		
Последняя точка / Последняя дата	Целое число / Datatime		
Группа «Период прогнозирования»			
Первая точка / Начальная дата	Целое число / Datatime		
Последняя точка / Последняя дата	Целое число / Datatime		
Автоопределение ретро	Флажок	Определяет может ли строится прогноз в пересечении с модельным рядом (false) или только после модельного ряда (true)	
Группа «Специфические настройки блока»			
Количество шагов	Целое число	Определяет количество шагов которое будет рассчитано.	
Количество наблюдений, добавляемых на каждом шаге	Строка	Целочисленный массив	
Параметр alpha	Строка	Вещественный массив	
Параметр beta	Строка	Вещественный массив	
Параметр gamma	Строка	Вещественный массив	
Группа «Общие настройки блока»			
Поле таблицы с настройками	Строка		
Ссылка на документацию	Строка		
Описание блока	Строка		

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Описание	Пример
Пропуск расчета блока	Флажок		

6.6.5.3 Входы

Название входа	Тип	Обязательный	Описание
Зависимая переменная Y	Временной ряд	Да	Ряд для которого необходимо рассчитать модель
Независимые переменные X. Факт	Набор временных рядов	Да	
Независимые переменные X. Прогноз	Набор временных рядов	Нет	

6.6.5.4 Выходы

Название выхода	Тип	Описание
Зависимая переменная Y	Временной ряд	Временной ряд для которого была рассчитана модель тренда. Если в ряде были пропуски – то пропуски восстановлены или заполнены в зависимости от выбранного метода обработки пропусков
Коэффициенты модели и из статистические характеристики	Таблица	Содержит значения всех рассчитанных коэффициентов модели, их стандартных ошибок, t-статистик и вероятностей t-статистики для последнего шага расчета.
Модельный ряд	Временной ряд	
Прогнозный ряд	Временной ряд	
Ряд остатков	Временной ряд	
Коэффициенты модели и из статистические характеристики всех шагов расчета	Строка (html)	Содержит значения всех рассчитанных коэффициентов модели, их стандартных ошибок, t-статистик и вероятностей t-статистики для всех шагов расчета
Ряды всех шагов расчета	Таблица	
Статистические характеристики	Строка (html)	Содержит описательные статистики модели для всех шагов расчета
Предупреждение	Строка (html)	Сообщения метода в случае каких-то проблем с расчетом

6.6.5.5 Дополнительно

Описание особенностей:

- Модель всегда содержит константу.
- Блок работает только с временными рядами.
- В ряде должно быть поле ts_name, в атрибутах или же в первичном ключе.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Описание ограничений:

- Наборы «Ряды независимых переменных X. Факт» и «Ряды независимых переменных X. Прогноз» должны иметь одинаковое количество независимых переменных.
- Значение поля «Последняя точка прогноза» должна быть меньше значения поля «Первая точка прогноза».
- Длинна массивов в полях «Параметр alpha», «Параметр beta», «Параметр gamma» и «Количество наблюдений, добавляемых на каждом шаге» должна быть ровна значению в поле «Количество шагов моделирования».
- Параметр alpha должен удовлетворять условию: $0 < \alpha \leq 1$.
- Параметры beta и gamma должны удовлетворять условию: $\gamma + \beta \leq 1$.
- Сумма элементов массива «Количество наблюдений, добавляемых на каждом шаге» должна быть меньше значения поля «Последняя точка расчета».

Пример шаблона для статистических характеристик

```
{% for step in input %}
<div>Шаг {{ step.step }}</div>
<div>
<table>
<tr>
<td>r2</td>
<td>{{"%.2f"|format(step.statistics.r2)}}</td>
<td>ssr</td>
<td>{{"%.2f"|format(step.statistics.ssr)}}</td>
<td>f_stat</td>
<td>{{"%.2f"|format(step.statistics.f_stat)}}</td>
<td>logl</td>
<td>{{"%.2f"|format(step.statistics.logl)}}</td>
</tr>
<tr>
<td>adjr2</td>
<td>{{"%.2f"|format(step.statistics.adjr2)}}</td>
<td>aic</td>
<td>{{"%.2f"|format(step.statistics.aic)}}</td>
<td>fp_val</td>
<td>{{"%.2f"|format(step.statistics.fp_val)}}</td>
<td></td>
<td></td>
</tr>
</table>
</div>
</br>
{% endfor %}
```

Пример шаблона для коэффициентов модели

```
{% for step in input %}
<div>Шаг {{ step.step }}</div>
<div>
  <table>
    <tr>
      <th>Наименование</th>
```


© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

```
      <th>Параметры модели</th>
      <th>Стандартная ошибка параметров модели</th>
      <th>t-статистика Стьюдента для параметров модели</th>
      <th>Вероятности для t-статистики</th>
    </tr>
    {% for item in step.coefficient %}
        <tr>
    <td>{{ item.name }}</td>
    <td>{{"%.2f"|format(item.B)}}</td>
    <td>{{"%.2f"|format(item.st_err)}}</td>
    <td>{{"%.2f"|format(item.t_value)}}</td>
    <td>{{"%.2f"|format(item.p_value)}}</td>
  </tr>
    {% endfor %}
  </table>
</div>
</br>
{% endfor %}
```

Описание блока в t_library

```
{
 "name": "Адаптивный МНК (AOLS)",

 "desc": "Адаптивный МНК (AOLS)",
 "package": "vmstatistic.aols",
 "module": "compute",
 "version": "1.0.0",
 "visualizer": 0,
 "changeshowinapp": 1,
 "settings": [
  {
   "name": "Первая точка моделирования",
   "id": "first_point",
   "desc": "",
   "type": "int",
   "mandatory": 1,
   "default": 0
  },
  {
   "name": "Последняя точка моделирования",
   "id": "last_point",
   "desc": "",
   "type": "int",
   "mandatory": 1,
   "default": 0
  },
  {
   "name": "Количество точек прогноза",
   "id": "forecast_count_point",
   "desc": "",
   "type": "int",
   "mandatory": 1,
   "default": 0
  },
  {
   "name": "Количество шагов моделирования",
```


© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

```
   "id": "step_count",
   "desc": "",
   "type": "int",
   "mandatory": 1,
   "default": 0
  },
  {
   "name": "Количество наблюдений, добавляемых на каждом шаге",
   "id": "observations",
   "desc": "",
   "type": "string",
   "mandatory": 1,
   "default": 0
  },
  {
   "name": "Параметр alpha",
   "id": "alpha",
   "desc": "",
   "type": "string",
   "mandatory": 1,
   "default": 0
  },
  {
   "name": "Параметр beta",
   "id": "beta",
   "desc": "",
   "type": "string",
   "mandatory": 1,
   "default": 0
  },
  {
   "name": "Параметр gamma",
   "id": "gamma",
   "desc": "",
   "type": "string",
   "mandatory": 1,
   "default": 0
  }
 ],
 "inputs": [
  {
   "name": "Зависимая переменная Y",
   "id": "Y",
   "type": "series",
   "mandatory": 1,
   "desc": "",
   "array": 0
  },
  {
   "name": "Ряды независимых переменных X. Факт",
   "id": "x_fact",
   "type": "series",
   "mandatory": 1,
   "desc": "",
   "array": 1
```


© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

```
  },
  {
   "name": "Ряды независимых переменных X. Прогноз",
   "id": "x_forecast",
   "type": "series",
   "mandatory": 1,
   "desc": "",
   "array": 1
  }
 ],
 "outputs": [
  {
   "name": "Модельный ряд",
   "id": "fittedvalues",
   "desc": "",
   "type": "series",
   "mandatory": 1,
   "array": 0
  },
  {
   "name": "Прогнозный ряд",
   "id": "predict",
   "desc": "",
   "type": "series",
   "mandatory": 1,
   "array": 0
  },
  {
   "name": "Ряд остатков",
   "id": "resid",
   "desc": "",
   "type": "series",
   "mandatory": 1,
   "array": 0
  },
  {
   "name": "Статистические характеристики",
   "id": "statistics",
   "desc": "",
   "type": "json",
   "mandatory": 1,
   "array": 0
  },
  {
   "name": "Коэффициенты модели и их статистические характеристики",
   "id": "stat_coefficient",
   "desc": "",
   "type": "json",
   "mandatory": 1,
   "array": 0
  },
  {
   "name": "Ряды всех шагов расчета",
   "id": "result_table",
   "desc": "",
```


© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

```
    "type": "series",
    "mandatory": 1,
    "array": 1
   }
  ],
  "style": {
   "icon": "fas fa-calculator"
  }
}
```

6.6.6 Блок «Бинарная регрессия»

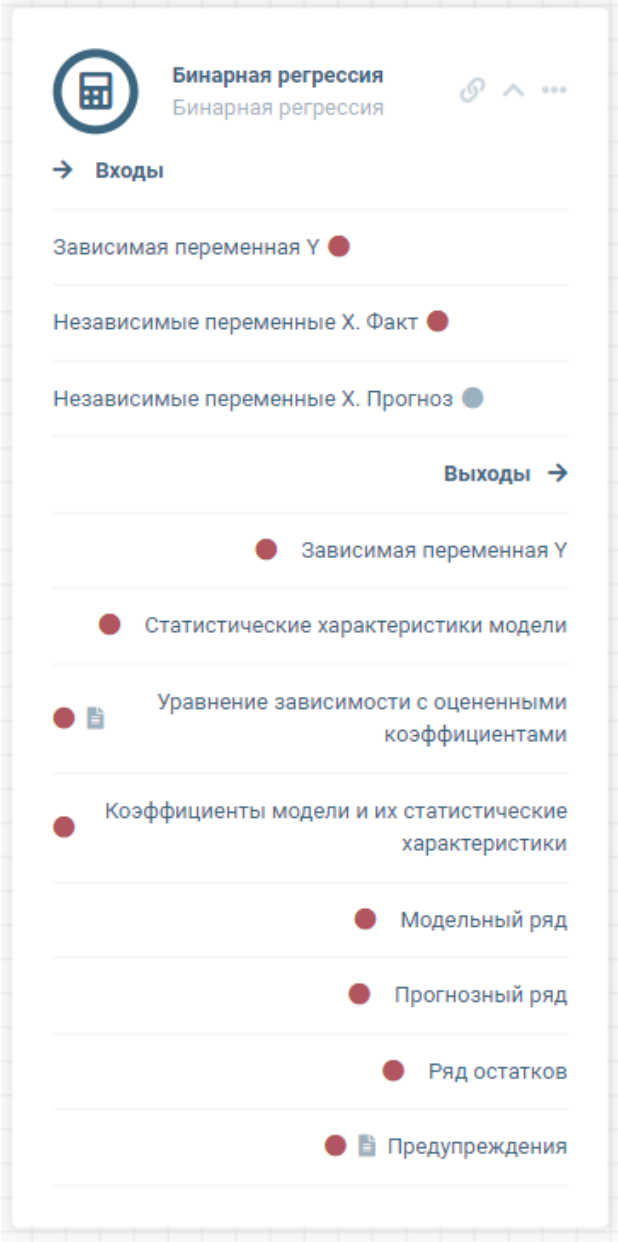


Рисунок 380 – Пример блока «Бинарная регрессия»

6.6.6.1 Назначение

Блок предназначен для расчет логистической регрессии (Logit и Probit). Логистическая регрессия – это статистическая модель, используемая для прогнозирования вероятности возникновения некоторого события путём его сравнения с логистической кривой. Эта регрессия выдаёт ответ в виде вероятности бинарного события (1 или 0).

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.6.6.2 Системное описание

6.6.6.2.1 Специфические пункты меню

Специфических пунктов меню нет. Полное описание меню приведено в описании блока «Меню» (см. п. 6.1.4.7).

6.6.6.2.2 Настройки блока

Наименование	Тип данных	Примечание
Основные свойства		
Наименование блока	Строка	Наименование, которое будет иметь блок в графе
Структурный тип данных		
Спецификация	Выпадающий список	Поле поддерживает единичную отметку. Определяет каким образом будут заданы периоды моделирования и прогнозирования. по умолчанию выбран «Индексы». Содержит следующие элементы: • Индексы • Даты
Период моделирования		
Первая точка/Начальная дата	Целое число / Datatime	Первая точка/Начальная дата должна быть меньше Последней точки/Последней даты
Последняя точка/Последняя дата	Целое число / Datatime	
Период прогнозирования		
Первая точка/Начальная дата	Целое число / Datatime	Первая точка/Начальная дата должна быть меньше Последней точки/Последней даты
Последняя точка/Последняя дата	Целое число / Datatime	
Автоопределение ретро	Флаг	
Специфические настройки блока		
С константой	Флаг	Определяет будет ли оцениваться константа в модели
Тип модели	Выпадающий список	Поле поддерживает единичную отметку. По умолчанию выбрана Logit. Содержит следующие элементы: • Logit • Probit
Обработка пропусков		
Обработка пропусков	Выпадающий список	По умолчанию выбран «Удаление пропусков». Поле поддерживает единичную отметку. Содержит следующие элементы: • Удаление пропусков • Без пропусков

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Наименование	Тип данных	Примечание
Вывод значений	Выпадающий список	По умолчанию выбран «Выводить исходные ряды». Поле поддерживает единичную отметку. Содержит следующие элементы: • Выводить исходные ряды. Будут выведены ряды, в том виде, в котором они были введены в метод • Выводить обработанные ряды. Будут выведены ряды с заполненными или удаленными точками
Общие настройки блока		
Ссылка на документацию	Строка	Ссылка, добавленная в это поле будет доступна из графа при нажатии на символ 🔗 в поле блока
Описание блока	Строка	Текст может содержать краткое описание блока
Пропуск расчета блока	Флаг	В случае если флаг включен – блок будет исключен из цепочки расчета и на графу будет отображен серым цветом
Кэширование результатов расчета блока	Флаг	Флаг предназначен для блоков карты
Предпросмотр	Флаг	Если блок связан с какими-то визуализаторами, то при включенном предпросмотре расчет отобразиться в специальном окне, которое будет содержать присоединенные визуализаторы. Механизм позволяет корректировать настройки блока до основного расчета

6.6.6.3 Входы блока

Название входа	Тип	Описание
Зависимая переменная Y	Временной ряд	Обязательная переменная. Атрибуты входного ряда должны содержать поле **ts_name**
Независимые переменные X. Факт	Временной ряд	Обязательная переменная. Атрибуты входного ряда должны содержать поле **ts_name**. Факт и прогноз одного ряда должны иметь одинаковый **ts_name**
Независимые переменные X. Прогноз	Временной ряд	Необязательная переменная. Атрибуты входного ряда должны содержать поле **ts_name**. Факт и прогноз одного ряда должны иметь одинаковый **ts_name**

6.6.6.4 Выходы блока

Название выхода	Тип	Описание
Зависимая переменная Y	Временной ряд	Ряд имеет model_name «Факт». Ряд можно вывести в таблицу или на диаграмму
Статистические характеристики модели	Структура (html)	Структура содержит общие характеристики модели (см. Блок «Статистические характеристики» п. 6.1.4.17)

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название выхода	Тип	Описание
Уравнение зависимости с оцененными характеристики	Строка	Содержит уравнение модели. Можно вывести в блок «html-шаблон»
Коэффициенты модели и их статистические характеристики	Таблица	Можно вывести в блок «Таблица» (п. 6.1.2.3)
Модельный ряд	Временной ряд	Ряд имеет model_name «Модель». Ряд можно вывести в таблицу или на диаграмму
Прогнозный ряд	Временной ряд	Ряд имеет model_name «Прогноз». Ряд можно вывести в таблицу или на диаграмму
Ряд остатков	Временной ряд	Ряд имеет model_name «Остатки». Ряд можно вывести в таблицу или на диаграмму
Предупреждения	Строка	Можно вывести в блок «Html-шаблон»

6.6.6.5 Теоретическое описание

6.6.6.5.1 Общая теория блока

Logit модель

Логистическая регрессия является одним из статистических методов классификации с использованием линейного дискриминанта Фишера. В отличие от обычной регрессии, в методе логистической регрессии не производится предсказание значения числовой переменной исходя из выборки исходных значений. Вместо этого, значением функции является вероятность того, что данное исходное значение принадлежит к определенному классу. Для простоты, давайте предположим, что у нас есть только два класса и вероятность, которую мы будем определять, P_+ вероятности того, что некоторое значение принадлежит классу «+». И конечно $P_- = 1 - P_+$. Таким образом, результат логистической регрессии всегда находится в интервале [0, 1].
Другими словами делается предположение о том, что вероятность наступления события $y = 1$ равна:

$$\mathbb{P}\{y = 1 \mid x\} = f(z),$$

где $z = \theta^T x = \theta_0 + \theta_1 x_1 + \ldots + \theta_n x_n$, x и θ – векторы-столбцы значений независимых переменных $1, x_1, \ldots, x_n$ и параметров (коэффициентов регрессии) – вещественных чисел $\theta_0, \ldots, \theta_n$, соответственно, а $f(z)$ – так называемая *логистическая функция* (иногда также называемая логит-функцией):

$$f(z) = \frac{1}{1 + e^{-z}}.$$

Probit модель

Пробит-модель является частным случаем модели бинарного выбора в которой используется нормальное распределение. А именно, пусть зависимая переменная Y является бинарной, то есть может принимать только два значения, которые для упрощения предполагаются равными 1 и 0. Например, Y может означать

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

наличие/отсутствие каких либо условий, успех или провал чего-либо, ответ да/нет в опросе и т. д. Пусть также имеется вектор регрессоров (факторов) X, которые оказывают влияние на Y. В пробит-модели предполагается, что вероятность того, что $Y = 1$ определяется нормальным распределением, таким образом пробит-модель имеет вид:

$$p(x) = P(Y = 1 \mid X = x) = \Phi(x^T b)$$

где Φ – интегральная функция распределения (*CDF*) стандартного нормального распределения, b – неизвестные параметры, которые требуется оценить.
Использование именно стандартного нормального распределения не ограничивает общности модели, так как возможное ненулевое среднее учтено в константе, которая обязательно присутствует в числе факторов, а возможная неединичная дисперсия учитывается за счет соответствующего нормирования всех коэффициентов b.

6.6.6.5.2 Ограничения метода

1. Зависимый ряд должен быть бинарным.
2. В ряде должно быть поле ts_name, в атрибутах или же в первичном ключе.
3. Обработка пропусков в зависимой переменной может быть только через удаление пропусков, метод заполнения должны быть исключены.
4. Структура массивов независимых рядов для моделирования и прогнозирования должна совпадать.
5. Иногда выходит ошибка Perfect separation detected, results not available, в таких случаях необходимо попросить пользователя подкорректировать период моделирования или зависимую переменную.
6. Ряды должны иметь минимум k+1/k+2(константа false/true) точек.

6.6.6.5.3 Дополнительные ссылки

Общая информация: Логистическая регрессия

Теоретическая информация об используемых функциях: Discrete Choice Models

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.6.7 Блок «Импорт статистических характеристик из БД»

6.6.7.1 Назначение

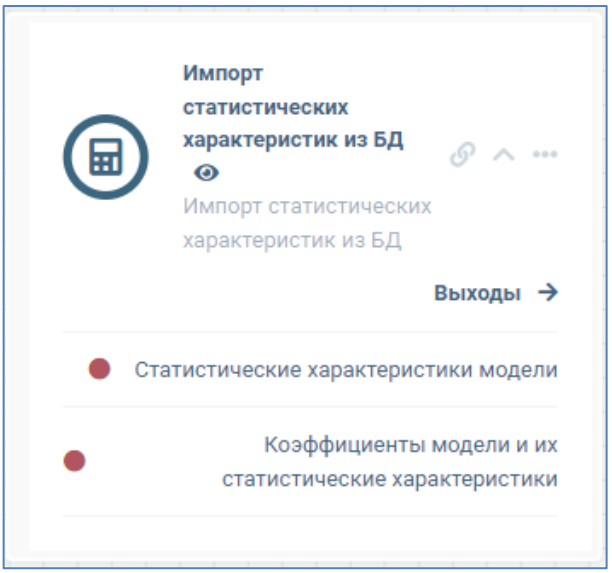


Рисунок 381 – Пример блока «Импорт статистических характеристик из БД»

Блок позволяет получить выходы статистического блока по его идентификатору для отображения в визуализаторе Блок «Статистические характеристики» (п. 6.1.4.17) или таблице.

6.6.7.2 Настройки блока

Название параметра	Тип параметра	Описание	Пример
Группа «Специфические настройки блока»			
id блока	Строка	ID блока, статистики которого необходимо получить	5fc20852-0d90-42d5-8ef7-0b0788bd079e

6.6.7.3 Группа «Общие настройки блока»

Аналогично приведенным в п. 6.1.1.1.3.

6.6.7.4 Входы

В блоке нет входов.

6.6.7.5 Выходы

Название выхода	Тип	Описание
Статистические характеристики модели	Структура (html)	Структура содержит общие характеристики модели (см. Блок «Статистические характеристики» п. 6.1.4.17)
Коэффициенты модели и из статистические характеристики	Объекты	Объекты, которые можно вывести в блок «Таблица» или обработать в других блоках

6.6.7.6 События блока

В блоке нет событий.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.6.7.7 Описание работы блока

Блок выводит рассчитанные статистические данные блока, указанного в настройках. При этом возвращается html структура, оформленная в виде таблицы – для быстрого отображения, или объекты платформы в виде, пригодном для отображения в таблице или дальнейшей обработки на графе.

Если блок, выходы которого необходимо получить, не рассчитан – расчет этого блока вернет ошибку.

6.6.8 Блок «Корреляционная матрица»

6.6.8.1 Назначение

Блок позволяет рассчитать корреляционную матрицу для набора рядов.

6.6.8.2 Настройки блока

Название параметра	Тип параметра	Описание	Пример
Группа «Основные свойства»			
Наименование блока	Строка	По умолчанию «Корреляционная матрица»	
Группа «Обработка пропусков»			
Обработка пропусков	Выпадающий список	• По умолчанию выбран «Удаление пропусков». Поле поддерживает единичную отметку • Удаление пропусков • Средним по выборке • Заданным значением • Максимальным значением • Минимальным значением • Линейная интерполяция • Квадратичная интерполяция • Кубическая интерполяция	
Значение	Вещественное число	Значение для метода «Заданным значением»	
Группа «Общие настройки блока»			
Поле таблицы с настройками	Строка		
Ссылка на документацию	Строка		
Описание блока	Строка		
Пропуск расчета блока	Флажок		

6.6.8.3 Входы

Название входа	Тип	Обязательный
Ряды	Набор временных рядов	Да

6.6.8.4 Выходы

Название выхода	Тип	Описание
Корреляционная матрица	Таблица	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.6.8.5 Дополнительно

Описание особенностей:

- Метод работает только с набором временных рядов.
- В ряде должно быть поле ts_name, в атрибутах или же в первичном ключе.

Описание ограничений:

- Набор должен состоять хотя бы из 2 временных рядов.

Описание блока в t_library


```
{
 "name": "K-Means",
 "id": "8ab13db6-39a4-49f6-acdd-79b6de8dd5e6",
 "desc": "K-Means",
 "version": "1.0.0",
 "package": "vmstatistic.k_means",
 "module": "compute",
 "visualizer": 0,
 "settings": [
  {
   "id": "n_cluster",
   "desc": "",
   "name": "Количество кластеров",
   "type": "int",
   "default": 8,
   "mandatory": 1
  },
  {
   "id": "max_iter",
   "desc": "",
   "name": "Максимальное число итераций",
   "type": "int",
   "default": 300,
   "mandatory": 1
  },
  {
   "id": "n_init",
   "desc": "",
   "name": "Количество раз вычисления алгоритма с разными центрами",
   "type": "int",
   "default": 10,
   "mandatory": 1
  }
 ],
 "inputs": [
  {
   "name": "Набор обучающих объектов",
   "id": "training_object",
   "type": "series",
   "mandatory": 1,
   "desc": "",
   "array": 1
  },
```


© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

```
  {
   "name": "Набор тестовых объектов",
   "id": "testing_object",
   "type": "series",
   "": 0,
   "desc": "",
   "array": 1
  }
 ],
 "outputs": [
  {
   "name": "Распределение объектов обучающей выборки",
   "id": "table",
   "desc": "",
   "type": "series",
   "mandatory": 1,
   "array": 1
  },
  {
   "name": "Сумма квадратов расстояний",
   "id": "inertia",
   "desc": "",
   "type": "float",
   "mandatory": 1,
   "array": 1
  },
  {
   "name": "Распределение объектов тестовой выборки",
   "id": "predict",
   "desc": "",
   "type": "series",
   "mandatory": 0,
   "array": 1
  }
 ],
 "style": {
  "icon": "fas fa-calculator"
 }
}
```

6.6.8.6 Дополнительные ссылки

Общая теория

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.6.9 Блок «Линейная регрессия»

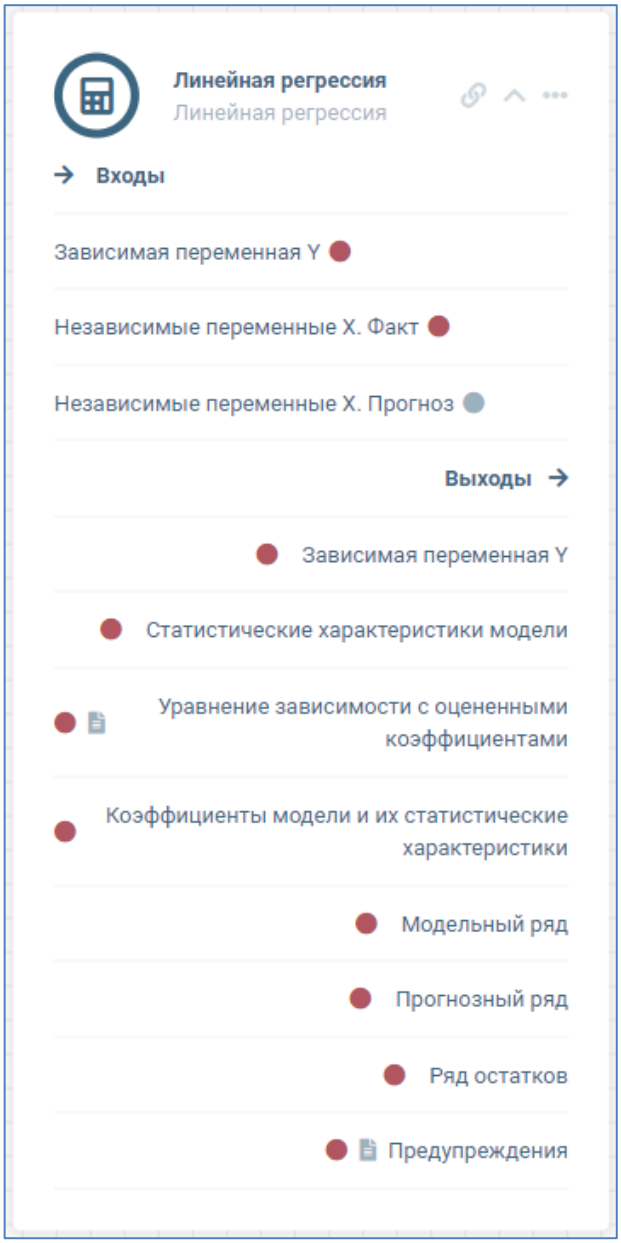


Рисунок 382 – Пример блока «Линейная регрессия»

6.6.9.1 Назначение

Блок предназначен для расчета множественной линейной модели методами OLS и GLS.

6.6.9.2 Системное описание

6.6.9.2.1 Специфические пункты меню

Наименование	Описание
Мастер автоподбора...	Вызывает меню «Мастер автоподбора факторов линейной регрессии». Мастер разработан для подбора лучшей модели регрессии из предложенный методу на вход, факторов

Полное описание меню приведено в описании блока «Меню» (см. п. 6.1.4.7).

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.6.9.2.2 Настройки блока

Наименование	Тип данных	Примечание
Основные свойства		
Наименование блока	Строка	Наименование, которое будет иметь блок в графе
Структурный тип данных		
Спецификация	Выпадающий список	Поле поддерживает единичную отметку. Определяет каким образом будут заданы периоды моделирования и прогнозирования. по умолчанию выбран «Индексы». Содержит следующие элементы: • Индексы • Даты
Период моделирования		
Первая точка/Начальная дата	Целое число / Datatime	Первая точка/Начальная дата должна быть меньше Последней точки/Последней даты
Последняя точка/Последняя дата	Целое число / Datatime	
Период прогнозирования		
Первая точка/Начальная дата	Целое число / Datatime	Первая точка/Начальная дата должна быть меньше Последней точки/Последней даты
Последняя точка/Последняя дата	Целое число / Datatime	
Автоопределение ретро	Флаг	
Специфические настройки блока		
С константой	Флаг	Определяет будет ли оцениваться константа в модели
Вид модели	Выпадающий список	Поле поддерживает единичную отметку. По умолчанию выбрана OLS. Содержит следующие элементы: • Линейная регрессия OLS • Линейная регрессия GLS
Семейство распределений	Выпадающий список	Поле поддерживает единичную отметку. Активен только если для расчета выбрана «Линейная регрессия GLS». По умолчанию выбрано нормальное распределение. Содержит следующие элементы: • Биноминальное распределение • Распределение Пуасона • Отрицательное биномиальное распределение • Нормальное распределение. По умолчанию. • Гамма распределение • Обратное нормальное распределение • Распределение Твидди

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Наименование	Тип данных	Примечание
Автоподбор		
Учет автоподбора	Флаг	Автоматически включается, если был использован «Мастер автоподбора факторов линейной регрессии» (см. раздел «Специфические пункты меню»). Если выключить флаг будет рассчитана обычная линейная регрессия
Список рядов, отобранных автоподбором	Строка	Результат работы мастера автоподбора факторов линейной регрессии
Обработка пропусков		
Обработка пропусков	Выпадающий список	По умолчанию выбран «Удаление пропусков». Поле поддерживает единичную отметку. • Удаление пропусков • Средним по выборке • Заданным значением • Максимальным значением • Минимальным значением • Линейная интерполяция • Квадратичная интерполяция • Кубическая интерполяция
Значение	Вещественное число	Значение появляется только при выборе метода «Заданным значением»
Вывод значений	Выпадающий список	По умолчанию выбран «Выводить исходные ряды». Поле поддерживает единичную отметку. Содержит следующие элементы: • Выводить исходные ряды. Будут выведены ряды, в том виде, в котором они были введены в метод. • Выводить обработанные ряды. Будут выведены ряды с заполненными или удаленными точками
Общие настройки блока		
Поле таблицы с настройками	Строка	Устаревший параметр
Ссылка на документацию	Строка	Ссылка, добавленная в это поле будет доступна из графа при нажатии на символ 🔗 в поле блока
Описание блока	Строка	Текст может содержать краткое описание блока
Пропуск расчета блока	Флаг	В случае если флаг включен – блок будет исключен из цепочки расчета и на графу будет отображен серым цветом
Кэширование результатов расчета блока	Флаг	Флаг предназначен для блоков карты.
Предпросмотр	Флаг	Если блок связан с какими-то визуализаторами, то при включенном предпросмотре расчет отобразиться в специальном окне, которое будет содержать присоединенные визуализаторы. Механизм позволяет корректировать настройки блока до основного расчета

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.6.9.2.3 Входы блока

Наименование входа	Тип данных	Описание
Зависимая переменная Y	Временной ряд	Обязательная переменная. Атрибуты входного ряда должны содержать поле ts_name
Независимые переменные X. Факт	Временной ряд	Обязательная переменная. Атрибуты входного ряда должны содержать поле ts_name. Факт и прогноз одного ряда должны иметь одинаковый ts_name
Независимые переменные X. Прогноз	Временной ряд	Необязательная переменная. Атрибуты входного ряда должны содержать поле ts_name. Факт и прогноз одного ряда должны иметь одинаковый ts_name

6.6.9.2.4 Выходы блока

Наименование выхода	Тип данных	Описание
Зависимая переменная Y	Временной ряд	Ряд имеет model_name «Факт». Ряд можно вывести в таблицу или на диаграмму
Статистические характеристики модели	Структура (html)	Структура содержит общие характеристики модели (см. Блок «Статистические характеристики» п. 6.1.4.17)
Уравнение зависимости с оцененными характеристики	Строка	Содержит уравнение модели. Можно вывести в блок «Html-шаблон»
Коэффициенты модели и из статистические характеристики	Таблица	Можно вывести в блок «Таблица»
Модельный ряд	Временной ряд	Ряд имеет model_name «Модель». Ряд можно вывести в таблицу или на диаграмму
Прогнозный ряд	Временной ряд	Ряд имеет model_name «Прогноз». Ряд можно вывести в таблицу или на диаграмму
Ряд остатков	Временной ряд	Ряд имеет model_name «Остатки». Ряд можно вывести в таблицу или на диаграмму
Предупреждения	Строка	Можно вывести в блок «Html-шаблон»

6.6.9.3 Теоретическое описание

6.6.9.3.1 Общая теория блока

В блоке представлено два метода оценки линейных моделей:

Метод наименьших квадратов (OLS)

Метод предназначен для решения различных задач, основанный на минимизации суммы квадратов отклонений некоторых функций от искомых переменных.

Другими словами пусть $\boldsymbol{x}$ – набор $\boldsymbol{n}$ неизвестных переменных (параметров).

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

$f_i(x),\ i = 1, \ldots, m,\ m > n$ – совокупность функций от этого набора переменных.

Задача заключается в подборе таких значений $\boldsymbol{x}$, чтобы значения этих функций были максимально близки к некоторым значениям y_i. По существу речь идет о «решении» переопределенной системы уравнений $f_i(x) = y_i,\ i = 1, \ldots, m$ в указанном смысле максимальной близости левой и правой частей системы. Суть МНК заключается в выборе в качестве «меры близости» суммы квадратов отклонений левых и правых частей $|f_i(x) - y_i|$. Таким образом, сущность МНК может быть выражена следующим образом:

$$\sum_i e_i^2 = \sum_i (y_i - f_i(x))^2 \to \min_x$$

В случае, если система уравнений имеет решение, то есть наименьшее значение суммы квадратов будет равно нулю, то для таких задач, лучше использовать «методы поиска точных решений», например решение может быть найдено аналитически или различными численными методами оптимизации. Метод наименьших квадратов лучше использовать, если система переопределена, то есть, говоря нестрого, количество независимых уравнений больше количества искомых переменных, то система не имеет точного решения. Метод позволяет найти некоторый «оптимальный» вектор $\boldsymbol{x}$ в смысле максимальной близости векторов $\boldsymbol{y}$ и $\boldsymbol{f(x)}$ или максимальной близости вектора отклонений $\boldsymbol{e}$ к нулю.

Обобщённый метод наименьших квадратов (GLS)

Обобщённый метод наименьших квадратов сводится к минимизации «обобщённой суммы квадратов» остатков регрессии – $e^T W e$, где $\boldsymbol{e}$ – вектор остатков, $\boldsymbol{W}$ – симметрическая положительно определенная весовая матрица. Обычный МНК является частным случаем обобщённого, когда весовая матрица пропорциональна единичной. Метод поддерживает несколько вариантов распределений остатков, для этого в методе предусмотрен параметр «Семейство распределений». В зависимости от выбранного семейства можно решить следующие модели:

- Биноминальное распределение – логистическая регрессия.
- Распределение Пуассона – регрессия Пуассона.
- Нормальное распределение – линейная регрессия.
- Гамма распределение – Гамма-регрессия.

6.6.9.3.2 Ограничения метода

- Блок работает только с временными рядами.
- Метод не может рассчитать без факторную модель типа y=const.
- В ряде должно быть поле ts_name, в атрибутах или же в первичном ключе.
- Для оценки параметров ряды должны иметь k+1 точку наблюдений, где k - количество независимых переменных X в наборе.
- Наборы «Ряды независимых переменных X. Факт» и «Ряды независимых переменных X. Прогноз» должны иметь одинаковое количество независимых переменных.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.6.9.3.3 Дополнительные ссылки

Общая информация о линейной регрессии: Обобщённый метод наименьших квадратов

Регрессия Пуассона: Poisson regression

Описание функции в пакете statmodels: OLS, GLS

6.6.10 Блок «МНК с ограничениями»

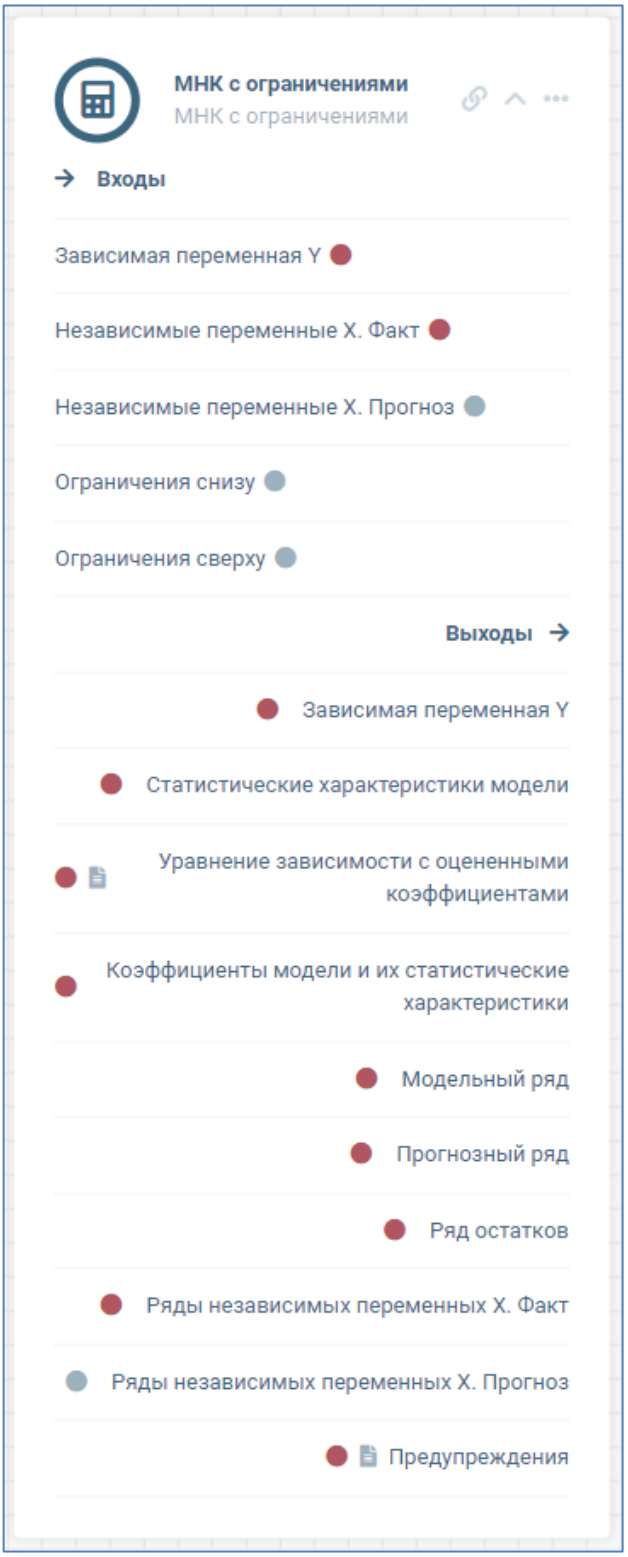


Рисунок 383 – Пример блока «МНК с ограничениями»

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.6.10.1 Назначение

Блок позволяет рассчитать линейную модель при условии ограничений на коэффициенты модели.

6.6.10.2 Настройки блока

Название параметра	Тип параметра	Описание	Пример
Группа «Основные свойства»			
Наименование блока	Строка	По умолчанию «МНК с ограничениями»	
Группа «Структурный тип данных»			
Спецификация	Выпадающий список	Поле поддерживает единичную отметку. Определяет каким образом будут заданы периоды моделирования и прогнозирования. по умолчанию выбран «Индексы». Содержит следующие элементы: • Индексы • Даты	
Группа «Период моделирования»			
Первая точка / Начальная дата	Целое число / Datatime		
Последняя точка / Последняя дата	Целое число / Datatime		
Группа «Период прогнозирования»			
Первая точка / Начальная дата	Целое число / Datatime		
Последняя точка / Последняя дата	Целое число / Datatime		
Автоопределение ретро	Флажок	Определяет может ли строится прогноз в пересечении с модельным рядом (false) или только после модельного ряда (true)	
Группа «Специфические настройки блока»			
С константой	Флажок	Определяет будет ли оцениваться константа в модели.	
Группа «Обработка пропусков»			
Обработка пропусков	Выпадающий список	По умолчанию выбран «Удаление пропусков». Поле поддерживает единичную отметку. • Удаление пропусков • Средним по выборке • Заданным значением • Максимальным значением • Минимальным значением • Линейная интерполяция • Квадратичная интерполяция. • Кубическая интерполяция	
Значение	Вещественное число	Значение для метода «Заданным значением"	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Описание	Пример
Группа «Общие настройки блока»			
Поле таблицы с настройками	Строка		
Ссылка на документацию	Строка		
Описание блока	Строка		
Пропуск расчета блока	Флажок		

6.6.10.3 Входы

Название входа	Тип	Обязательный	Описание
Зависимая переменная Y	Временной ряд	Да	Ряд для которого необходимо рассчитать модель
Независимые переменные X. Факт	Набор временных рядов	Да	
Независимые переменные X. Прогноз	Набор временных рядов	Нет	
Ограничение снизу	Ряд		
Ограничение сверху	Ряд		

6.6.10.4 Выходы

Название выхода	Тип	Описание
Зависимая переменная Y	Временной ряд	Временной ряд для которого была рассчитана модель тренда. Если в ряде были пропуски – то пропуски восстановлены или заполнены в зависимости от выбранного метода обработки пропусков
Общий вывод характеристик	Структура (html)	Структура в виде таблицы, которая содержит коэффициенты модели и их статистические характеристики, а также все описательные характеристики модели
Уравнение зависимости с оцененными коэффициентами	Строка	Формула модели
Коэффициенты модели и из статистические характеристики	Таблица	Содержит значения всех рассчитанных коэффициентов модели, их стандартных ошибок, t-статистик и вероятностей t-статистики
Модельный ряд	Временной ряд	
Прогнозный ряд	Временной ряд	
Ряд остатков	Временной ряд	
Статистические характеристики		
Коэффициент детерминации R^2	Строка (html)	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название выхода	Тип	Описание
Скорректированные коэффициент детерминации R^2	Строка (html)	
Сумма квадратов остатков	Строка (html)	
Информационный критерий Акаика	Строка (html)	
Информационный критерий Байеса	Строка (html)	
F-статистика Фишера	Строка (html)	
Вероятность для F-статистики	Строка (html)	
Предупреждение	Строка (html)	Сообщения метода в случае каких-то проблем с расчетом

6.6.10.5 Дополнительно

- Описание особенностей:
  - Блок работает только с временными рядами.
  - В ряде должно быть поле ts_name, в атрибутах или же в первичном ключе.
- Описание ограничений:
  - Для оценки параметров ряды должны иметь k+1 точку наблюдений, где k - количество независимых переменных X в наборе.
  - Наборы «Ряды независимых переменных X. Факт» и «Ряды независимых переменных X. Прогноз» должны иметь одинаковое количество независимых переменных.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.6.11 Блок «Методика АМ. Задача оптимизации»

6.6.11.1 Назначение

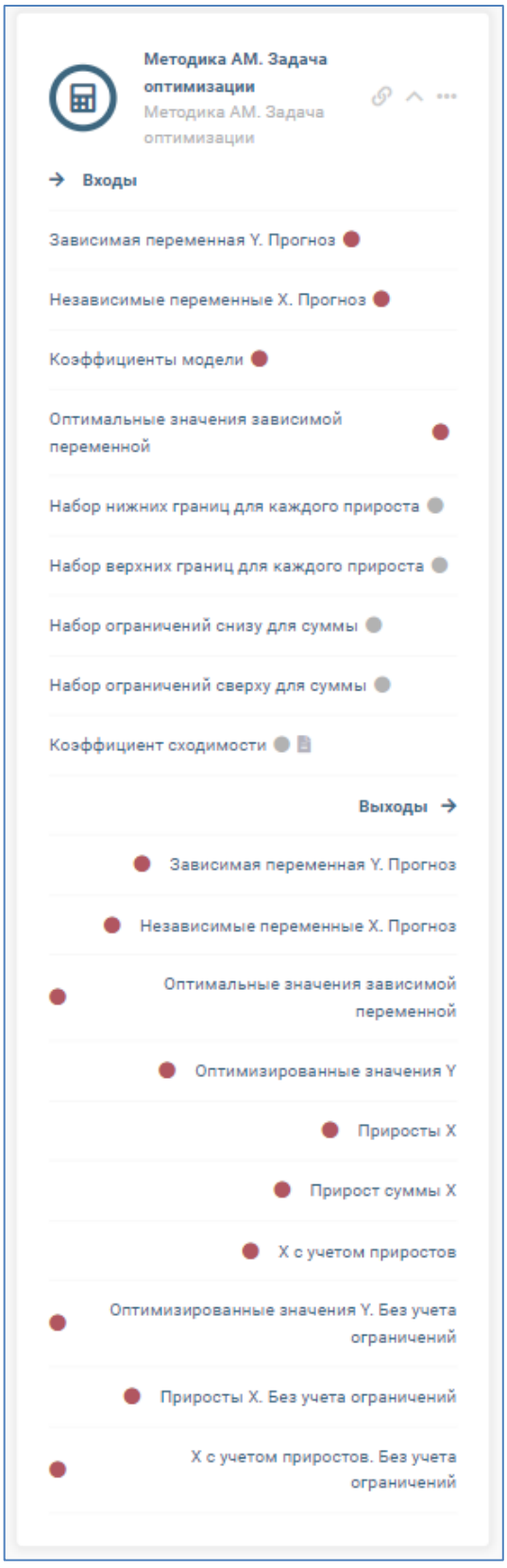


Рисунок 384 – Пример блока «Методика АМ. Задача оптимизации»

Блок позволяет рассчитать целевую задачу с использованием методики АМ посредством оптимизационного алгоритма **trust-constr**.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.6.11.2 Настройки блока

Название параметра	Тип параметра	Описание	Пример
Группа «Основные свойства»			
Наименование блока	Строка	По умолчанию «Методика АМ. Задача оптимизации»	
Группа «Специфические настройки блока»			
Коэффициент сходимости	Вещественное число		
Группа «Общие настройки блока»			
Ссылка на документацию	Строка		
Описание блока	Строка		
Пропуск расчета блока	Флажок		
Отображать процесс выполнения	Флажок		

6.6.11.3 Входы

Название входа	Тип	Обязательный	Описание
Зависимая переменная Y. Прогноз	Временной ряд	Да	• Для корректной работы метода должен быть выходом блока «Линейная регрессия». • Длина ряда должна быть равна длине входных данных «Независимые ряды X (Прогноз)»
Независимые ряды X. Прогноз	Набор временных рядов	Да	• На вход должен прийти массив рядов прогнозных X, на которых был произведен расчет линейной регрессии • В атрибуте 'ts_name' должен быть идентификатор ряда (должны соответствовать «ts_name» коэффициентам модели)
Коэффициенты модели	Ряд	Да	На вход должен передаваться выход «Статистические характеристики модели» блока «Линейная регрессия»
Оптимальные значения зависимой переменной	Временной ряд	Да	должен содержать на одно наблюдение больше, чем «Прогнозный ряд»

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название входа	Тип	Обязательный	Описание
Набор нижних границ для каждого прироста	Набор временных рядов	Нет	• На вход ожидается массив рядов • В атрибуте 'ts_name' должен быть идентификатор ряда (должны соответствовать «ts_name» независимые переменные X) к которому будет применено ограничение
Набор верхних границ для каждого прироста	Набор временных рядов	Нет	• На вход ожидается массив рядов • В атрибуте 'ts_name' должен быть идентификатор ряда (должны соответствовать «ts_name» независимые переменные X) к которому будет применено ограничение
Набор ограничений для суммы снизу	Набор временных рядов	Нет	Количество рядов в наборе должно соответствовать количеству заданных суммарных ограничений
Набор ограничений для суммы сверху	Набор временных рядов	Нет	Количество рядов в наборе должно соответствовать количеству заданных суммарных ограничений
Коэффициент сходимости	Вещественное число	Нет	К этому входу можно подключить слайдер. Если слайдера нет, берется значение установленное в настройках блока

6.6.11.4 Выходы

Название выхода	Тип	Описание
Зависимая переменная Y. Прогноз	Временной ряд	Возвращает данные которые пришли во вход «Зависимая переменная Y. Прогноз»
Независимые переменные X. Прогноз	Набор временных рядов	Возвращает данные которые пришли во вход «Независимые переменные X. Прогноз»
Оптимальные значения зависимой переменной	Временной ряд	Возвращает данные которые пришли во вход «Оптимальные значения зависимой переменной»
Оптимизированные значения Y	Временной ряд	
Приросты X	Набор временных рядов	
Прирост суммы X	Временной ряд	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название выхода	Тип	Описание
X с учетом приростов	Набор временных рядов	Независимые переменные X. Прогноз + Приросты X
Оптимизированные значения Y. Без учета ограничений	Временной ряд	
Приросты X. Без учета ограничений	Набор временных рядов	
X с учетом приростов. Без учета ограничений	Набор временных рядов	Независимые переменные X. Прогноз + Приросты X. Без учета ограничений

6.6.11.5 События блока

В блоке нет событий.

6.6.11.6 Дополнительно

Описание особенностей:

- Базовая задача может быть рассчитана без ограничений.
- Ограничения снизу можно задать только для конкретных независимых переменных. (для этого во вход «Ограничения снизу для каждого» надо передать ряды с конкретными идентификаторами ряда (x1, x2, x3 ...), по которым будет ограничение).
- В ряде должно быть поле ts_name, в атрибутах или же в первичном ключе.

Описание ограничений:

- Набор временных рядов «Независимые ряды X (Прогноз)» должен совпадать количеством переданных коэффициентов.
- Размер рядов ограничений должен совпадать в размеров прогнозного ряда.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.6.12 Блок «Методика АМ»

6.6.12.1 Назначение

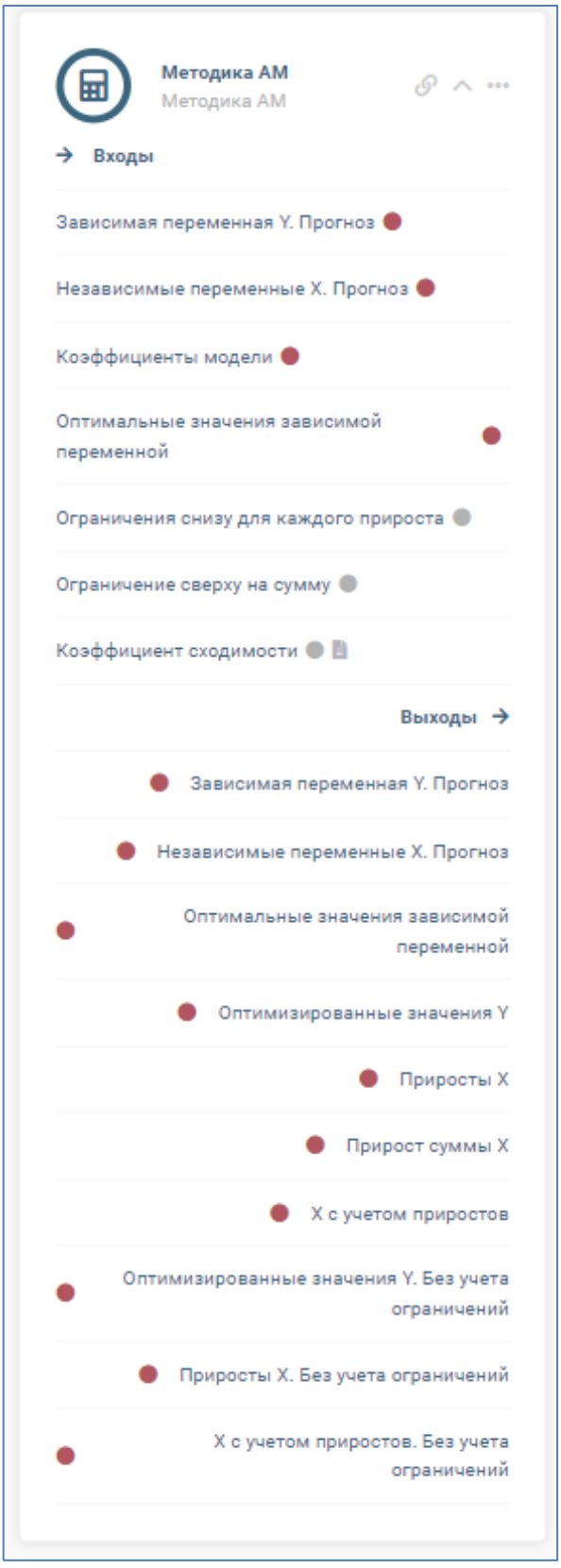


Рисунок 385 – Пример блока «Методика АМ»

Блок позволяет рассчитать целевую задачу с использованием методики АМ.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.6.12.2 Настройки блока

Название параметра	Тип параметра	Описание	Пример
Группа «Основные свойства»			
Наименование блока	Строка	По умолчанию «Методика АМ»	
Группа «Специфические настройки блока»			
Коэффициент сходимости	Вещественное число		
Группа «Общие настройки блока»			
Ссылка на документацию	Строка		
Описание блока	Строка		
Пропуск расчета блока	Флажок		
Отображать процесс выполнения	Флажок		

6.6.12.3 Входы

Название входа	Тип	Обязательный	Описание
Зависимая переменная Y. Прогноз	Временной ряд	Да	• Для корректной работы метода должен быть выходом «Прогнозный ряд» блока «Линейная регрессия». • Длина ряда должна быть равна длине входных данных «Независимые ряды X (Прогноз)»
Независимые переменные X. Прогноз	Набор временных рядов	Да	• На вход должен прийти массив рядов прогнозных X, на которых был произведен расчет линейной регрессии • В атрибуте 'ts_name' должен быть идентификатор ряда (должны соответствовать «ts_name» коэффициентам модели)
Коэффициенты модели	Ряд	Да	На вход должен передаваться выход «Статистические характеристики модели» блока «Линейная регрессия»
Оптимальные значения зависимой переменной	Временной ряд	Да	Должен содержать на одно наблюдение больше, чем «Зависимая переменная Y. Прогноз»
Ограничения снизу для каждого прироста	Временной ряд	Нет	• На вход ожидается массив рядов • В атрибуте 'ts_name' должен быть идентификатор ряда (должны соответствовать «ts_name» независимые переменные X) к которому будет применено ограничение

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название входа	Тип	Обязательный	Описание
Ограничения сверху на сумму	Временной ряд	Нет	
Коэффициент сходимости	Вещественное число	Нет	К этому входу можно подключить слайдер. Если слайдера нет, берется значение установленное в настройках блока

6.6.12.4 Выходы

Название выхода	Тип	Описание
Зависимая переменная Y. Прогноз	Временной ряд	Возвращает данные которые пришли во вход «Зависимая переменная Y. Прогноз»
Независимые переменные X. Прогноз	Набор временных рядов	Возвращает данные которые пришли во вход «Независимые переменные X. Прогноз»
Оптимальные значения зависимой переменной	Временной ряд	Возвращает данные которые пришли во вход «Оптимальные значения зависимой переменной»
Оптимизированные значения Y	Временной ряд	
Приросты X	Набор временных рядов	
Прирост суммы X	Временной ряд	
X с учетом приростов	Набор временных рядов	Независимые переменные X. Прогноз + Приросты X
Оптимизированные значения Y. Без учета ограничений	Временной ряд	
Приросты X. Без учета ограничений	Набор временных рядов	
X с учетом приростов. Без учета ограничений	Набор временных рядов	Независимые переменные X. Прогноз + Приросты X. Без учета ограничений

6.6.12.5 События блока

В блоке нет событий.

6.6.12.6 Дополнительно

Описание особенностей:

- Базовая задача может быть рассчитана без ограничений.
- Ограничения снизу можно задать только для конкретных независимых переменных. (для этого во вход «Ограничения снизу для каждого» надо передать ряды с конкретными идентификаторами ряда (x1, x2, x3 ...), по которым будет ограничение).
- В ряде должно быть поле ts_name, в атрибутах или же в первичном ключе.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Описание ограничений:

- Набор временных рядов «Независимые ряды X (Прогноз)» должен совпадать количеством переданных коэффициентов.
- Размер рядов ограничений должен совпадать в размеров прогнозного ряда.

6.6.13 Блок «Методы автоподбора факторов»

6.6.13.1 Назначение

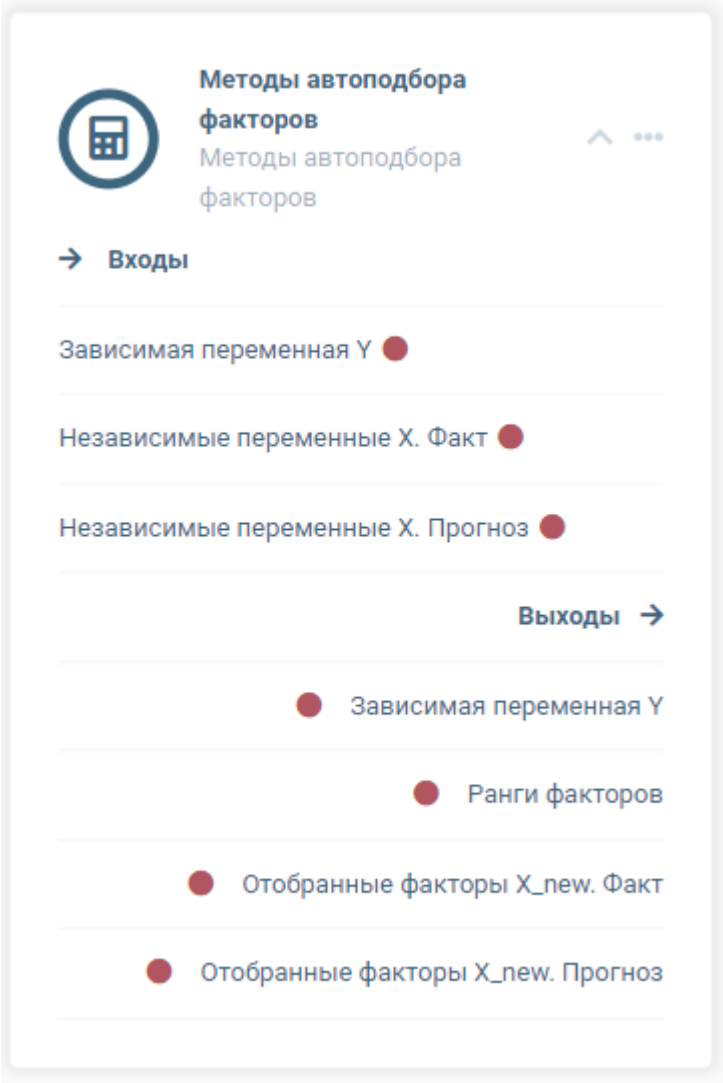


Рисунок 386 – Пример блока «Методы автоподбора факторов»

Блок предназначен для автоматического отбора факторов методами: рекурсивное исключение признаков с перекрестной проверкой (RFECV) и одномерный автоподбор факторов с настраиваемой стратегией (GUS).

Методы автоподбора можно использовать для выбора признаков (уменьшения размерности наборов выборок), для улучшения показателей точности моделей или для повышения производительности на наборах данных очень большой размерности.

6.6.13.2 Настройки блока

Название параметра	Тип параметра	Описание	Пример
Группа «Основные свойства»			

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Описание	Пример
Наименование блока	Строка	По умолчанию «Методы автоподбора факторов»	
Группа «Автоподбор факторов»			
Тип автоподбора	Выпадающий список	По умолчанию выбран "Рекурсивное исключение признаков (RFECV)". Поле поддерживает единичную отметку. Содержит следующие элементы: • Рекурсивное исключение признаков (RFECV). • Одномерный автоподбор факторов (GUS).	
Группа «Модель» *Раздел виден в случае, если в пункте* ***Тип автоподбора*** *выбран пункт Рекурсивное исключение признаков (RFECV)*			
Модель	Выпадающий список	По умолчанию выбран "Линейная регрессия". Поле поддерживает единичную отметку. Содержит следующие элементы: • Линейная регрессия. • Логистическая регрессия.	
С константой	Флаг		
Начальное состояние системы	Целое число	По умолчанию 123	
Группа «Специфические настройки блока» Раздел виден в случае, если в пункте ***Тип автоподбора*** выбран пункт Рекурсивное исключение признаков (RFECV)			
Минимальное количество отбираемых факторов	Целое число	По умолчанию 1	
Метод отбора	Выпадающий список	Если в настройке Модель выбран пункт "Линейная регрессия", то список выглядит следующим образом: По умолчанию выбран "Автоматический". Поле поддерживает единичную отметку. Содержит следующие элементы: • Автоматический. • Коэффициент детерминации R2. • Средняя абсолютная ошибка MAE. • Квадрат средней ошибка MSE. • Корень из квадрата средней ошибки RMSE. Если в настройке Модель выбран пункт "Логистическая регрессия", то список выглядит следующим образом:	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Описание	Пример
		По умолчанию выбран "Автоматический". Поле поддерживает единичную отметку. Содержит следующие элементы: • Автоматический. • Точность Accuracy. • Точность Precision. • Отзыв Recall. • Площадь под Roc кривой RocAUC.	
Группа «Настройки кросс-валидации» *Раздел виден в случае, если в пункте **Тип автоподбора** выбран пункт Рекурсивное исключение признаков (RFECV).*			
Тип кросс-валидации	Выпадающий список	По умолчанию выбран "Обычная". Поле поддерживает единичную отметку. Содержит следующие элементы: • Линейная регрессия. • Логистическая регрессия.	
Количество отрезков nfold	Целое число	По умолчанию 3	
Группа «Функция автоподбора» *Раздел виден в случае, если в пункте **Тип автоподбора** выбран пункт Одномерный автоподбор факторов (GUS).*			
Функция отбора данных	Выпадающий список	По умолчанию выбран "Регрессия. Коэффициент Пирсона". Поле поддерживает единичную отметку. Содержит следующие элементы: • Регрессия. Коэффициент Пирсона. • Регрессия. F-тест. • Классификация. Хи-квадрат. • Классификация. F-тест.	
Группа «Настройки стратегии R» *Раздел виден в случае, если в пункте **Тип автоподбора** выбран пункт Одномерный автоподбор факторов (GUS) и если в пункте **Функция автоподбора** выбраны пункты Регрессия. Коэффициент Пирсона или Классификация. Хи-квадрат.*			
Стратегия отбора данных	Выпадающий список	По умолчанию выбран "Процент лучших переменных". Поле поддерживает единичную отметку. Содержит следующие элементы: • Процент лучших переменных. • K лучших переменных.	
Процент	Целое число	Процент отбираемых факторов. По умолчанию 50. Виден, если в пункте ***Стратегия отбора данных*** выбран пункт ***Процент лучших переменных***.	
Параметр k	Целое число	Количество k отбираемых факторов. По умолчанию 5. Виден, если в пункте	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Описание	Пример
		Стратегия отбора данных выбран пункт ***К лучших переменных***.	
Группа «Настройки стратегии F» *Раздел виден в случае, если в пункте* ***Тип автоподбора*** *выбран пункт Одномерный автоподбор факторов (GUS) и если в пункте* ***Функция автоподбора*** *выбраны пункты Регрессия. F-тест или Классификация. F-тест.*			
Стратегия отбора данных	Выпадающий список	По умолчанию выбран "Процент лучших переменных". Поле поддерживает единичную отметку. Содержит следующие элементы: • Процент лучших переменных. • K лучших переменных. • Тест Fpr. • Процедура Беньямини-Хохберга. • Коэффициент семейных ошибок (FWE).	
Процент	Целое число	Процент отбираемых факторов. По умолчанию 50. Виден, если в пункте Стратегия отбора данных выбран пункт Процент лучших переменных.	
Параметр k	Целое число	Количество k отбираемых факторов. По умолчанию 5. Виден, если в пункте ***Стратегия отбора данных*** выбран пункт ***К лучших переменных***.	
Параметр alpha	Вещественное число	Уровень значимости критерия alpha. По умолчанию 0.5. Виден, если в пункте ***Стратегия отбора данных*** выбраны пункты: • Тест Fpr. • Процедура Беньямини-Хохберга. • Коэффициент семейных ошибок (FWE).	
Группа «Обработка пропусков»			
Обработка пропусков	Выпадающий список	По умолчанию выбран "Удаление пропусков". Поле поддерживает единичную отметку. Содержит следующие элементы: • Удаление пропусков. • Без пропусков.	
Вывод значений	Выпадающий список	По умолчанию выбран "Выводить исходные ряды". Поле поддерживает единичную отметку. Содержит следующие элементы: • Выводить исходные ряды. Будут выведены ряды, в том виде, в котором они были введены в метод	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Описание	Пример
		• Выводить обработанные ряды. Будут выведены ряды с заполненными или удаленными точками.	
Группа «Общие настройки блока»			
Ссылка на документацию	Строка		
Описание блока	Строка		
Пропуск расчета блока	Флажок		
Отображать процесс выполнения	Флажок		

6.6.13.3 Входы

Название входа	Тип	Обязательный	Описание
Зависимая переменная Y. Прогноз	Временной ряд	Да	Обязательная переменная. Атрибуты входного ряда должны содержать поле ts_name.
Независимые переменные X. Факт	Временной ряд	Да	Обязательная переменная. Атрибуты входного ряда должны содержать поле ts_name. Факт и прогноз одного ряда должны иметь одинаковый ts_name.
Независимые переменные X. Прогноз	Временной ряд	Нет	Необязательная переменная. Атрибуты входного ряда должны содержать поле ts_name. Факт и прогноз одного ряда должны иметь одинаковый ts_name.

6.6.13.4 Выходы

Название выхода	Тип	Описание
Зависимая переменная Y. Прогноз	Временной ряд	
Ранги факторов	Ряд данных	
Отобранные факторы X_new. Факт	Временной ряд	
Отобранные факторы X_new. Прогноз	Временной ряд	
Приросты X	Набор временных рядов	

6.6.13.5 События блока

В блоке нет событий.

6.6.13.6 Дополнительно

Метод RFECV

Цель рекурсивного исключения признаков (RFE) состоит в том, чтобы выбрать признаки путем рекурсивного рассмотрения все меньших и меньших наборов признаков. Метод работает с учетом внешних оценщиков, который присваивает веса признакам (например, коэффициентам линейной модели.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Оценщик обучается на начальном наборе факторов, важность каждого фактора рассчитывается автоматически. Из текущего набора факторов удаляются наименее не значимые факторы. Эта процедура рекурсивно повторяется для сокращенного набора до тех пор, пока в итоге не будет достигнуто желаемое количество факторов.

Метод GUS

Одномерный выбор факторов работает путем выбора лучших рядов на основе одномерных статистических тестов. Его можно рассматривать как этап предварительной обработки перед моделированием.

Для регрессионных моделей выбор происходит на основании r коэффициента Пирсона или F-статистики.

Для классификационных моделей на основании теста Хи квадрат (только для положительных данных) и F-теста ANOVA.

В качестве настраиваемых стратегий можно выбрать следующие:

- удаляет все переменные из набора кроме переменных с наивысшим рейтингом.
- удаляет все функции, кроме указанного пользователем процента с наивысшим рейтингом.
- выбор происходит на основании теста Fpr или теста на ложноположительные результаты. Фильтр отсеет все ранги ниже заданного.
- выбор происходит на основании процедуры Беньямини-Хохберга. Здесь параметр – это верхняя граница ожидаемого уровня ложных обнаружений.
- выбор происходит на основании коэффициента семейных ошибок (FWER).

Ограничения метода

- Размерность входного массива независимых переменных X должна содержать хотя бы 3 ряда.
- Минимальное количество объектов, которые необходимо выбрать должно принадлежать промежутку [1,k-1], где k - количество независимых факторов X.
- Параметр k должен принадлежать промежутку [1,k-1], где k - количество независимых факторов X.
- Параметр кросс-валидации n-fold принадлежать промежутку [1,n], где n - количество наблюдений на периоде моделирования.
- Методы классификации подходят только для бинарной классификации
- Метод Хи-квадрат может работать только с положительными независимыми рядами.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.6.14 Блок «Нелинейная регрессия»

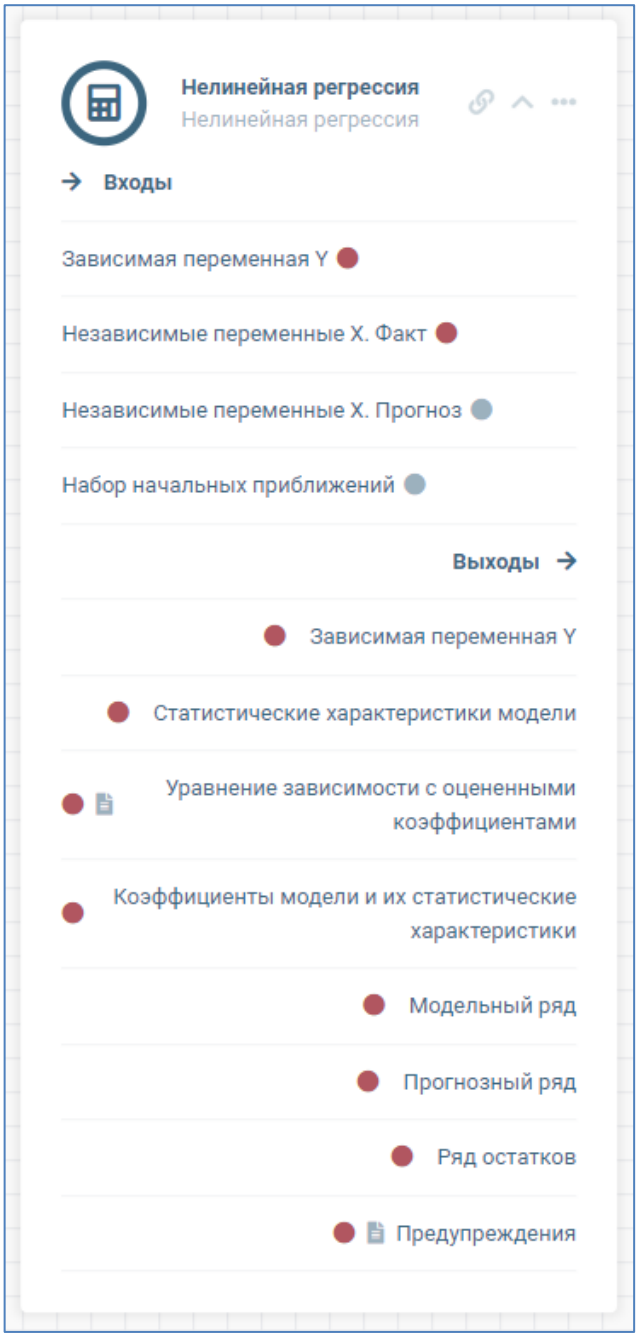


Рисунок 387 – Пример блока «Нелинейная регрессия»

6.6.14.1 Назначение

Блок позволяет рассчитать класс нелинейных регрессионных моделей. Решение получается путем оптимизации суммы квадратов отклонений.

6.6.14.2 Настройки блока

Название параметра	Тип параметра	Описание	Пример
Группа «Основные свойства»			
Наименование блока	Строка	По умолчанию «Нелинейная регрессия»	
Группа «Структурный тип данных»			

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Описание	Пример
Спецификация	Выпадающий список	Поле поддерживает единичную отметку. Определяет каким образом будут заданы периоды моделирования и прогнозирования. по умолчанию выбран «Индексы». Содержит следующие элементы: • Индексы • Даты	
Группа «Период моделирования»			
Первая точка / Начальная дата	Целое число / Datatime		
Последняя точка / Последняя дата	Целое число / Datatime		
Группа «Период прогнозирования»			
Первая точка / Начальная дата	Целое число / Datatime		
Последняя точка / Последняя дата	Целое число / Datatime		
Автоопределение ретро	Флажок	Определяет может ли строится прогноз в пересечении с модельным рядом (false) или только после модельного ряда (true)	
Группа «Специфические настройки блока»			
Формула	Строка	Спецификация рассчитываемой модели. формульное представление. Правила задания формул смотри в разделе «Дополнительно»	
Группа «Обработка пропусков»			
Обработка пропусков	Выпадающий список	По умолчанию выбран «Удаление пропусков». Поле поддерживает единичную отметку. • Удаление пропусков • Средним по выборке • Заданным значением • Максимальным значением • Минимальным значением • Линейная интерполяция • Квадратичная интерполяция • Кубическая интерполяция	
Значение	Вещественное число	Значение для метода «Заданным значением»	
Группа «Общие настройки блока»			
Поле таблицы с настройками	Строка		
Ссылка на документацию	Строка		
Описание блока	Строка		

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Описание	Пример
Пропуск расчета блока	Флажок		

6.6.14.3 Входы

Название входа	Тип	Обязательный	Описание
Зависимая переменная Y	Временной ряд	Да	Ряд для которого необходимо рассчитать модель
Независимые переменные X. Факт	Набор временных рядов	Да	
Независимые переменные X. Прогноз	Набор временных рядов	Нет	
Набор начальных приближений	Ряд	Нет	

6.6.14.4 Выходы

Название выхода	Тип	Описание
Зависимая переменная Y	Временной ряд	Временной ряд для которого была рассчитана модель тренда. Если в ряде были пропуски – то пропуски восстановлены или заполнены в зависимости от выбранного метода обработки пропусков
Общий вывод характеристик	Структура (html)	Структура в виде таблицы, которая содержит коэффициенты модели и их статистические характеристики, а также все описательные характеристики модели
Уравнение зависимости с оцененными коэффициентами	Строка	Формула модели
Коэффициенты модели и из статистические характеристики	Таблица	Содержит значения всех рассчитанных коэффициентов модели, их стандартных ошибок, t-статистик и вероятностей t-статистики
Модельный ряд	Временной ряд	
Прогнозный ряд	Временной ряд	
Ряд остатков	Временной ряд.	
Статистические характеристики		
Коэффициент детерминации R^2	Строка (html)	
Скорректированные коэффициент детерминации R^2	Строка (html)	
Сумма квадратов остатков	Строка (html)	
Информационный критерий Акаика	Строка (html)	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название выхода	Тип	Описание
Логарифмическая функция максимального правдоподобия	Строка (html)	
F-статистика Фишера	Строка (html)	
Вероятность для F-статистики	Строка (html)	
Предупреждение	Строка (html)	Сообщения метода в случае каких-то проблем с расчетом

6.6.14.5 Дополнительно

Описание особенностей:

- Блок работает только с временным рядом.
- В ряде должно быть поле ts_name, в атрибутах или же в первичном ключе.
- Статистики коэффициентов не рассчитываются в случае, если количество коэффициентов не совпадает с количеством заданных рядов (без учета константы).
- Последняя точка моделирования должна быть больше первой.
- Последняя точка прогнозирования должна быть больше первой.
- Ряд должен быть не константным.
- Если начальные приближения заданы – то они должны быть заданы для всех коэффициентов модели.

Правила задания формулы для расчета:

- формула может содержать следующий математические операции:
  - Сумма +
  - Вычитание -
  - Умножение *
  - Деление /
  - Возведение в степень **
  - Операции в скобочках ()
- Формула поддерживает следующие функции:
  - синус sin(x)
  - косинус cos(x)
  - арксинус arcsin(x)
  - арккосинус arccos(x)
  - арктангенс arctg(x)
  - арккотангенс arcctg(x)
  - экспонента exp(x)
  - логарифм log(x)
  - натуральный логарифм log1p (x)
  - десятичный логарифм log10(x)

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

- модуль abs(x)
- Пи pi

- Независимый ряд в формуле должен быть представлен в виде [ts_name] (Пример формулы: [X1]**c3 * sin(c1/[X2]) * c2 * exp([X4]/[X3]) + c0).
- Константа всегда обозначается как «0».

6.6.14.6 Дополнительные ссылки

Общая теория

6.6.15Блок «Тренд»

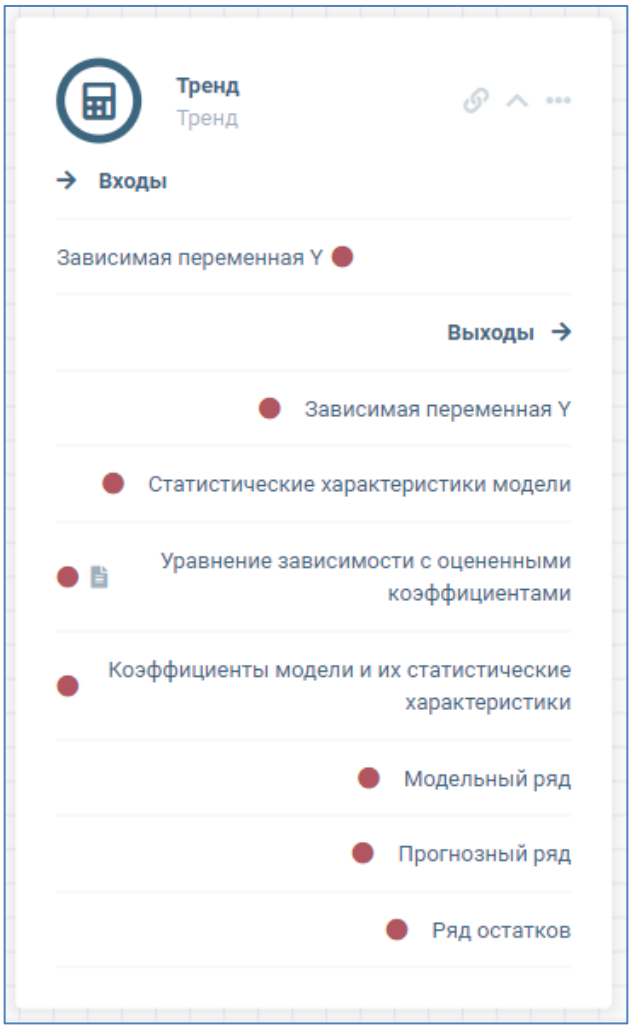


Рисунок 388 – Пример блока «Тренд»

6.6.15.1 Назначение

Тренд – неслучайная медленно меняющаяся составляющая временного ряда, на которую могут накладываться случайные колебания или сезонные эффекты. Метод позволяет представить временной ряд в виде суммы детерминированной функции времени $F(t, a)$ (a – вектор неизвестных параметров), зависящей от небольшого числа неизвестных параметров и случайной компоненты $err(t)$. По итогам работы методы на основании выбранного пользователем критерия будет выбрана лучшая функциональная зависимость описывающая ряд.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.6.15.2 Настройки блока

Название параметра	Тип параметра	Описание	Пример
Группа «Основные свойства»			
Наименование блока	Строка	По умолчанию «Тренд»	
Группа «Структурный тип данных»			
Спецификация	Выпадающий список	Поле поддерживает единичную отметку. Определяет каким образом будут заданы периоды моделирования и прогнозирования. по умолчанию выбран «Индексы». Содержит следующие элементы: • Индексы • Даты	
Группа «Период моделирования»			
Первая точка / Начальная дата	Целое число / Datatime		
Последняя точка / Последняя дата	Целое число / Datatime		
Группа «Период прогнозирования»			
Первая точка / Начальная дата	Целое число / Datatime		
Последняя точка / Последняя дата	Целое число / Datatime		
Группа «Специфические настройки блока»			
Функциональные зависимости	Выпадающий список	Поле поддерживает мульти отметку. По умолчанию выбрала линейная зависимость. Необходимо выбрать перечень функциональных зависимостей из которых будет производится подбор. Доступны следующие функциональные зависимости: • Линейная • Квадратичная • Кубическая • Четвертой степени • Составная • Уравнение роста • Логарифмическая • Гиперболическая • Экспонентная • Обратная • Степенная • Логистическая • Квадратный корень • Логарифмическая-параболическая	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Описание	Пример
Критерий отбора лучшей зависимости	Выпадающий список	По умолчанию выбран критерий «Коэффициент детерминации». Поле поддерживает единичную отметку. • Коэффициент детерминации • Скорректированный коэффициент детерминации • Сумма квадратов остатков	
Группа «Обработка пропусков"			
Обработка пропусков	Выпадающий список	По умолчанию выбран «Удаление пропусков». Поле поддерживает единичную отметку. • Удаление пропусков • Средним по выборке • Заданным значением • Максимальным значением • Минимальным значением • Линейная интерполяция • Квадратичная интерполяция • Кубическая интерполяция	
Значение	Вещественное число	Значение для метода «Заданным значением»	
Группа «Общие настройки блока»			
Поле таблицы с настройками	Строка		
Ссылка на документацию	Строка		
Описание блока	Строка		
Пропуск расчета блока	Флажок		

6.6.15.3 Входы

Название входа	Тип	Обязательный	Описание
Зависимая переменная Y	Временной ряд	Да	Ряд для которого необходимо рассчитать модель тренда

6.6.15.4 Выходы

Название выхода	Тип	Описание
Зависимая переменная Y	Временной ряд	Временной ряд для которого была рассчитана модель тренда. Если в ряде были пропуски – то пропуски восстановлены или заполнены в зависимости от выбранного метода обработки пропусков
Общий вывод характеристик	Структура (html)	Структура в виде таблицы, которая содержит коэффициенты модели и их статистические характеристики, а также все описательные характеристики модели

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название выхода	Тип	Описание
Уравнение зависимости с оцененными коэффициентами	Строка	Формула модели
Коэффициенты модели и из статистические характеристики	Таблица	Содержит значения всех рассчитанных коэффициентов модели, их стандартных ошибок, t-статистик и вероятностей t-статистики
Модельный ряд	Временной ряд	
Прогнозный ряд	Временной ряд	
Ряд остатков	Временной ряд	
Статистические характеристики		
Коэффициент детерминации R^2	Строка (html)	
Скорректированные коэффициент детерминации R^2	Строка (html)	
Сумма квадратов остатков	Строка (html)	
Информационный критерий Акаика	Строка (html)	
Информационный критерий Байеса	Строка (html)	
F-статистика Фишера	Строка (html)	
Вероятность для F-статистики	Строка (html)	

6.6.15.5 Дополнительно

В ряде должно быть поле **ts_name**, в атрибутах или же в первичном ключе.

Описание блока в t_library:

```
{
"name": "Тренд",
"id": "3786a440-dfc3-459c-ba43-5dd2470d8b95",
"desc": "Тренд и подбор функциональной зависимости",
"version": "1.0.0",
"package": "vmstatistic.trend",
"module": "compute",
"visualizer": 0,
"changeshowinapp": 1,
"settings": [
 {
  "name": "Начальная точка",
  "id": "n",
  "type": "int",
  "desc": "Точка, с которой начинается сгенерированный ряд",
  "mandatory": 1,
  "min": 1,
  "default": 1,
  "changeshowinapp": 1
 },
 {
  "name": "Функциональные зависимости",
```


© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

```
"id": "funcform",
"desc": "",
"type": "enum",
"multiselect": 1,
"default": 1,
"mandatory": 1,
"changeshowinapp": 1,
"enum": [
 {
  "val": 1,
  "name": "Линейная",
  "desc": ""
 },
 {
  "val": 2,
  "name": "Квадратичная",
  "desc": ""
 },
 {
  "val": 3,
  "name": "Кубическая",
  "desc": ""
 },
 {
  "val": 4,
  "name": "Четвёртой степени",
  "desc": ""
 },
 {
  "val": 5,
  "name": "Составная",
  "desc": ""
 },
 {
  "val": 6,
  "name": "Уравнение роста",
  "desc": ""
 },
 {
  "val": 7,
  "name": "Логарифмическая",
  "desc": ""
 },
 {
  "val": 8,
  "name": "Гиперболическая",
  "desc": ""
 },
 {
  "val": 9,
  "name": "Экспонентная",
  "desc": ""
 },
 {
  "val": 10,
```


© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

```
    "name": "Обратная",
    "desc": ""
   },
   {
    "val": 11,
    "name": "Степенная",
    "desc": ""
   },
   {
    "val": 12,
    "name": "Логистическая",
    "desc": ""
   },
   {
    "val": 13,
    "name": "Квадратный корень",
    "desc": ""
   },
   {
    "val": 14,
    "name": "Логарифмическая-параболическая",
    "desc": ""
   }
  ]
 },
 {
  "name": "Критерий отбора лучшей зависимости",
  "id": "crit",
  "desc": "",
  "type": "enum",
  "default": 1,
  "mandatory": 1,
  "multiselect": 0,
  "changeshowinapp": 1,
  "enum": [
   {
    "val": 1,
    "name": "Коэффициент детерминации",
    "desc": ""
   },
   {
    "val": 2,
    "name": "Скорректированный коэффициент детерминации",
    "desc": ""
   },
   {
    "val": 3,
    "name": "Сумма квадратов остатков",
    "desc": ""
   }
  ]
 },
 {
  "name": "Период прогнозирования",
  "id": "ForecastPeriod",
```


© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

```
   "type": "json",
   "desc": "",
   "mandatory": 1,
   "elements": [
    {
     "name": "Первая точка прогноза ",
     "id": "FstForecastPoint",
     "desc": "",
     "type": "int",
     "mandatory": 1,
     "min": 1,
     "default": 1,
     "changeshowinapp": 1
    },
    {
     "name": "Последняя точка прогноза",
     "id": "LstForecastPoint",
     "desc": "",
     "type": "int",
     "mandatory": 1,
     "min": 1,
     "default": 10,
     "changeshowinapp": 1
    }
   ]
  }
 ],
 "inputs": [
  {
   "name": "Показатель",
   "id": "Y",
   "type": "series",
   "mandatory": 1,
   "desc": "Входной временной ряд",
   "array": 1
  }
 ],
 "outputs": [
  {
   "name": "Уравнение зависимости с оцененными коэффициентами",
   "id": "equation",
   "desc": "",
   "type": "string",
   "mandatory": 1,
   "array": 0
  },
  {
   "name": "Коэффициенты модели и их статистические характеристики",
   "id": "par",
   "desc": "",
   "type": "series",
   "mandatory": 1,
   "array": 1
  },
  {
```


© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

```
"name": "Статистические характеристики",
"id": "statsistical_characteristics",
"desc": "",
"type": "json",
"mandatory": 1,
"array": 0,
"elements": [
 {
  "name": "Коэффициент R^2",
  "id": "rsquared",
  "desc": "",
  "type": "float",
  "array": 0,
  "mandatory": 1
 },
 {
  "name": "Скорректированный R^2",
  "id": "rsquared_adj",
  "desc": "",
  "type": "float",
  "array": 0,
  "mandatory": 1
 },
 {
  "name": "Информационный критерий Акаика",
  "id": "aic",
  "desc": "",
  "type": "float",
  "array": 0,
  "mandatory": 1
 },
 {
  "name": "Информационный критерий Байеса",
  "id": "bic",
  "desc": "",
  "type": "float",
  "array": 0,
  "mandatory": 1
 },
 {
  "name": "Сумма квадратов остатков модели",
  "id": "ssr",
  "desc": "",
  "type": "float",
  "array": 0,
  "mandatory": 1
 },
 {
  "name": "F – статистика модели ",
  "id": "fvalue",
  "desc": "",
  "type": "float",
  "array": 0,
  "mandatory": 1
 },
```


© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

```
    {
     "name": "Значение вероятности для F-статистики",
     "id": "f_pvalue",
     "desc": "",
     "type": "float",
     "array": 0,
     "mandatory": 1
    }
   ]
  },
  {
   "name": "Модельный ряд",
   "id": "fittedvalues",
   "desc": "",
   "type": "series",
   "mandatory": 1,
   "array": 0
  },
  {
   "name": "Прогнозный ряд",
   "id": "predict",
   "desc": "",
   "type": "series",
   "mandatory": 1,
   "array": 0
  },
  {
   "name": "Ряд остатков",
   "id": "resid",
   "desc": "",
   "type": "series",
   "mandatory": 1,
   "array": 0
  }
 ],
 "style": {
  "icon": "fas fa-calculator"
 }
}
```

6.6.15.6 Дополнительные ссылки

Общая теория

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.6.16 Блок «Экспоненциальное сглаживание»

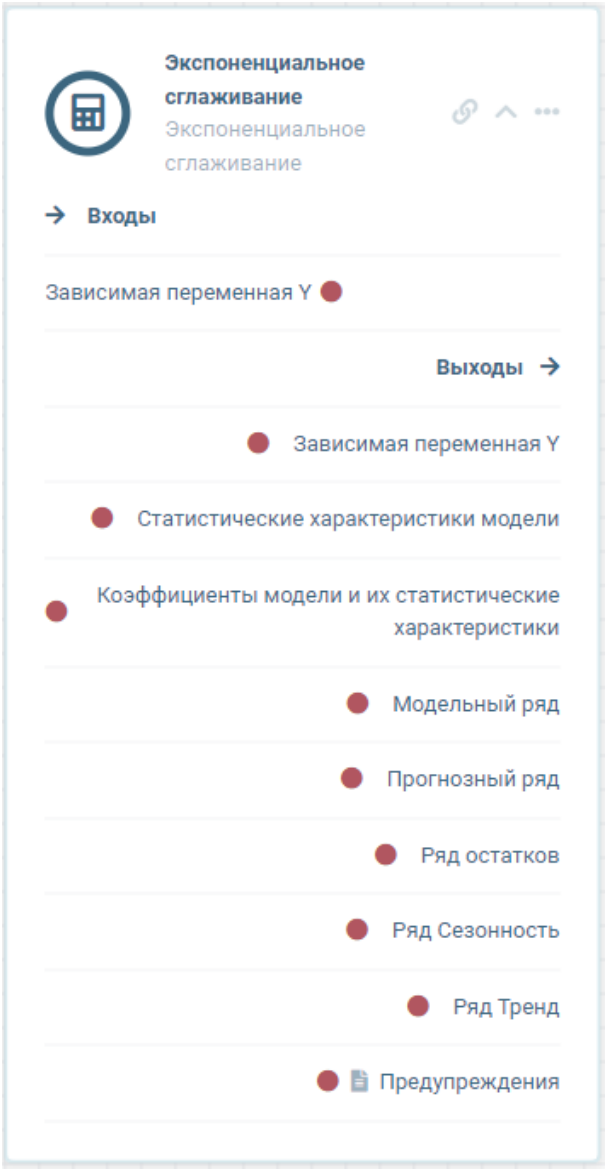


Рисунок 389 – Пример блока «Экспоненциальное сглаживание»

6.6.16.1 Назначение

Метод экспоненциальное сглаживание предназначен для выявления и анализа тенденций временного ряда с помощью его выравнивания и сглаживания.

6.6.16.2 Настройки блока

Название параметра	Тип параметра	Обязательный	Описание	Пример
Группа «Основные свойства»				
Наименование блока	Строка	+	По умолчанию «Экспоненциальное сглаживание"	
Группа «Структурный тип данных»				
Спецификация	Выпадающий список	+	Поле поддерживает единичную отметку. Определяет каким образом будут заданы	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
			периоды моделирования и прогнозирования. по умолчанию выбран «Индексы». Содержит следующие элементы: • Индексы • Даты	
Группа «Период моделирования»				
Первая точка / Начальная дата	Целое число / Datatime	+		1/01.01.2000
Последняя точка / Последняя дата	Целое число / Datatime	+		10/01.01.2010
Группа «Период прогнозирования»				
Первая точка / Начальная дата	Целое число / Datatime			11/01.01.2010
Последняя точка / Последняя дата	Целое число / Datatime			21/01.01.2020
Группа «Специфические настройки блока»				
Компонент тренда	Выпадающий список	-	• По умолчанию выбран «Аддитивный тренд». Поле поддерживает единичную отметку. • Аддитивный тренд. • Мультипликативный тренд. • Компонента тренда отсутствует	
Демпфированный тренд	Флажок	-	Рассчитывается только если компонента тренда задана	
Сезонная компонента	Выпадающий список	-	• По умолчанию выбран «Аддитивная сезонность». Поле поддерживает единичную отметку. • Аддитивная сезонность. • Мультипликативная сезонность. • Сезонная компонента отсутствует	
Параметр простого экспоненциального сглаживания	Вещественное число	-	Параметр альфа	0.8
Тренд параметр метода Хольта	Вещественное число	-	Параметр бета	0.2

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
Сезонный параметр метода Хольта-Винтерса	Вещественное число	-	Параметр гамма	0.5
Параметр демпфированного тренда	Вещественное число	-	Параметр фи	0.3
Период сезонности	Целое число	-	Количество точек ряда, образующих сезон	4 (при годовой динамике сезон будет равен 4 годам)
Оптимизация	Флажок	-		
Преобразование Бокса-Кокса	Флажок	-		
Группа «Обработка пропусков»				
Обработка пропусков	Выпадающий список	-	• По умолчанию выбран «Удаление пропусков». Поле поддерживает единичную отметку. • Удаление пропусков. • Средним по выборке. • Заданным значением. • Максимальным значением. • Минимальным значением. • Линейная интерполяция. • Квадратичная интерполяция. • Кубическая интерполяция.	
Значение	Вещественное число	-	Значение для метода «Заданным значением"	
Группа «Общие настройки блока»				
Ссылка на документацию	Строка			
Описание блока	Строка			
Пропуск расчета блока	Флажок			

6.6.16.3 Входы

Название входа	Тип	Обязательный	Описание
Зависимая переменная Y	Временной ряд	Да	Ряд для которого необходимо рассчитать экспоненциальное сглаживание

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.6.16.4 Выходы

Название выхода	Тип	Описание
Зависимая переменная Y	Временной ряд	Временной ряд для которого было рассчитано экспоненциальное сглаживание. Если в ряде были пропуски – то пропуски восстановлены или заполнены в зависимости от выбранного метода обработки пропусков
Общий вывод характеристик	Статистические характеристики	Структура выводиться в специализированный блок «Статистические характеристики» (см. п. 6.1.4.17), содержит общею информацию о оцененных параметрах и статистиках метода
Коэффициенты модели и из статистические характеристики	Таблица	Содержит значения всех рассчитанных коэффициентов модели
Модельный ряд	Временной ряд	
Прогнозный ряд	Временной ряд	
Ряд остатков	Временной ряд	
Ряд Сезонность	Временной ряд	Равен нулю, если не задана сезонная компонента
Ряд тренд	Временной ряд	Равен нулю, если не задана компонента тренда
Статистические характеристики		
Информационный критерий Акаика	Строка (html)	
Информационный критерий Байеса	Строка (html)	
Скорректированный информационный критерий Акаика	Строка (html)	
Сумма квадратов остатков	Строка (html)	
Коэффициент Бокса-Кокса	Строка (html)	
Предупреждение	Строка (html)	Сообщения метода в случае каких-то проблем с расчетом

6.6.16.5 Дополнительно

- Ряд должен иметь хотя бы 8 точек.
- Если компонента тренда не задана, но задано что тренд демпфированный – выходит ошибка: «Компонента тренда не задана».
- Рекомендуется все параметры задавать в диапазоне от 0 до 1, при иных значениях расчет также будет произведен, но не всегда результат будет адекватным.
- В ряде должно быть поле ts_name, в атрибутах или же в первичном ключе.

6.6.16.6 Дополнительные ссылки

Общая теория

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.6.17Блок «Эластичная сеть»

6.6.17.1 Назначение

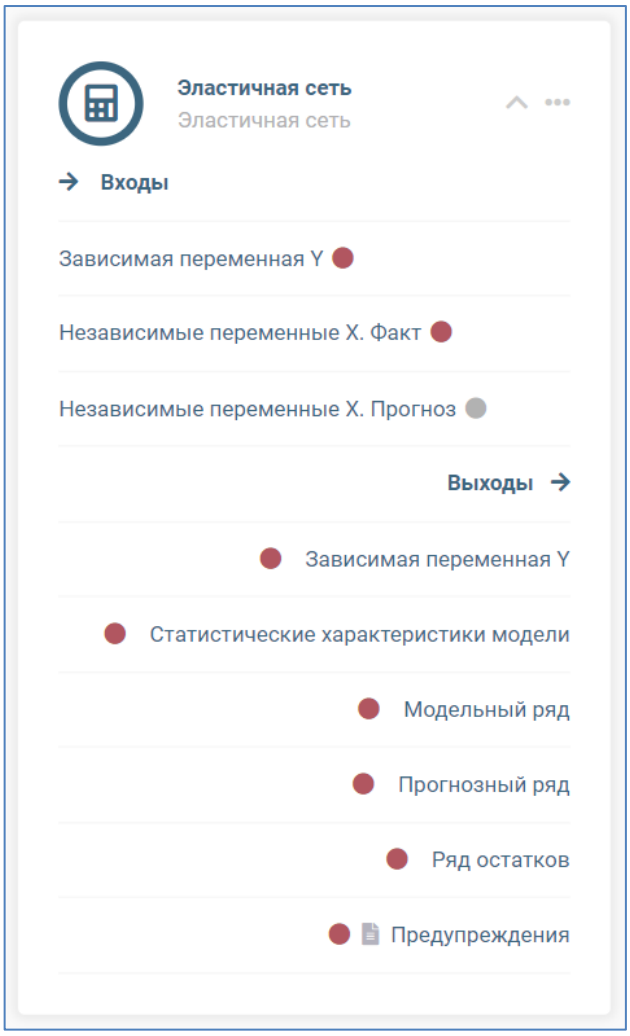


Рисунок 390 – Пример блока «Эластичная сеть»

Блок предназначен для расчета регрессии Elastic Net (эластическая сеть). Elastic-Net Regression представляет собой метод регуляризованной регрессии, сочитающий в себе Lasso (лассо L1) и Ridge (гребня) регрессии.

6.6.17.2 Настройки блока

Наименование	Тип данных	Примечание
Основные свойства		
Наименование блока	Строка	Наименование, которое будет иметь блок в графе
Структурный тип данных		
Спецификация	Выпадающий список	Поле поддерживает единичную отметку. Определяет каким образом будут заданы периоды моделирования и прогнозирования. по умолчанию выбран «Индексы». Содержит следующие элементы: • Индексы • Даты
Период моделирования		

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Наименование	Тип данных	Примечание
Первая точка/Начальная дата	Целое число / Datatime	Первая точка/Начальная дата должна быть меньше Последней точки/Последней даты
Последняя точка/Последняя дата	Целое число / Datatime	
Период прогнозирования		
Первая точка/Начальная дата	Целое число / Datatime	Первая точка/Начальная дата должна быть меньше Последней точки/Последней даты
Последняя точка/Последняя дата	Целое число / Datatime	
Специфические настройки блока		
С константой	Флаг	Определяет будет ли оцениваться константа в модели.
Семейство распределения остатков модели	Выпадающий список	Поле поддерживает единичную отметку. По умолчанию выбрано «Нормальное распределение». Содержит следующие элементы: • Нормальное распределение
Тип регрессии	Выпадающий список	Поле поддерживает единичную отметку. По умолчанию выбрано «Регрессия эластичная сеть». Содержит следующие элементы: • Регрессия гребня; • Регрессия Лассо; • Регрессия эластичная сеть. • Подбор лучшей модели
Настройки кросс-валидации		
Количество отрезков nfold	Целое число	По умолчанию задано 10
Обработка пропусков		
Обработка пропусков	Выпадающий список	• По умолчанию выбран «Удаление пропусков». Поле поддерживает единичную отметку; • Удаление пропусков; • Средним по выборке; • Заданным значением; • Максимальным значением; • Минимальным значением; • Линейная интерполяция; • Квадратичная интерполяция; • Кубическая интерполяция
Значение	Вещественное число	Значение появляется только при выборе метода «Заданным значением"
Вывод значений	Выпадающий список	По умолчанию выбран «Выводить исходные ряды». Поле поддерживает единичную отметку. Содержит следующие элементы:

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Наименование	Тип данных	Примечание
		• Выводить исходные ряды. Будут выведены ряды, в том виде, в котором они были введены в метод. • Выводить обработанные ряды. Будут выведены ряды с заполненными или удаленными точками

6.6.17.3 Группа «Общие настройки блока»

Аналогично приведенным в п. 6.1.1.1.3.

6.6.17.4 Входы блока

Название входа	Тип	Описание
Зависимая переменная Y	Временной ряд	Обязательная переменная. Атрибуты входного ряда должны содержать поле ts_name
Независимые переменные X. Факт	Временной ряд	Обязательная переменная. Атрибуты входного ряда должны содержать поле ts_name. Факт и прогноз одного ряда должны иметь одинаковый ts_name
Независимые переменные X. Прогноз	Временной ряд	Необязательная переменная. Атрибуты входного ряда должны содержать поле ts_name. Факт и прогноз одного ряда должны иметь одинаковый ts_name

6.6.17.5 Выходы блока

Название выхода	Тип	Описание
Зависимая переменная Y	Временной ряд	Ряд имеет model_name «Факт». Ряд можно вывести в таблицу или на диаграмму
Статистические характеристики модели	Структура (html)	Структура содержит общие характеристики модели (см. Блок «Статистические характеристики» п. 6.1.4.17)
Модельный ряд	Временной ряд	Ряд имеет model_name «Модель». Ряд можно вывести в таблицу или на диаграмму
Прогнозный ряд	Временной ряд	Ряд имеет model_name «Прогноз». Ряд можно вывести в таблицу или на диаграмму
Ряд остатков	Временной ряд	Ряд имеет model_name «Остатки». Ряд можно вывести в таблицу или на диаграмму.
Предупреждения	Строка	Можно вывести в блок «Html-шаблон"

6.6.17.6 События блока

В блоке нет событий

6.6.17.7 Теоретическое описание

6.6.17.7.1 Общая теория блока

Гребневая регрессия (ridge regression). Регуляризатор L2

Гребневая регрессия – это один из методов понижения размерности. Часто его применяют для борьбы с переизбыточностью данных, когда независимые переменные коррелируют друг с другом (т.е. имеет место мультиколлинеарность). Применение гребневой регрессии нередко оправдывают тем, что это практический приём, с помощью которого при желании можно получить меньшее значение среднего квадрата ошибки.

При оценке модели гребневой регрессии в общем случае вычисляется следующая задача:

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

$$\sum (y - \beta X)^2 + \lambda \beta^2 \rightarrow \min$$

,

где y – зависимая переменная, X – множество независимых переменных, β – коэффициента модели, λ – коэффициент штрафа.

Регрессия Лассо (lasso regressoin) Регуляризатор L1

Метод регрессии лассо похож на гребневую регрессию, но использует другое ограничение на коэффициенты. Основное различие лассо- и ридж-регрессии заключается в том, что первая может приводить к обращению некоторых независимых переменных в ноль, тогда как вторая уменьшает их до значений, близких к нулю.

При оценке модели Лассо регрессии в общем случае вычисляется следующая задача:

$$\sum (y - \beta X)^2 + \lambda |\beta| \rightarrow \min$$

Регрессия Эластичная сеть (elastic net regression)

Эластичная сеть – это модель регрессии с двумя регуляризаторами L1 и L2.

В общем случае при оценке модели решается следующая задача:

$$\sum (y - \beta X)^2 + \lambda_1 |\beta| + \lambda_2 \beta^2 \rightarrow \min$$

где λ_1 – коэффициент штрафа лассо регрессии, λ_2 – коэффициент штрафа гребневой регрессии.

В нашей реализации существует метод автоподбора лучшей модели эластической сети (сочетания λ_1 и λ_2). Для этого решается общее уравнение эластической сети модифицируется следующим образом:

$$\sum (y - \beta X)^2 + \lambda (\alpha |\beta| + (1 - \alpha) \beta^2)$$

здесь λ – общий штраф компонент уравнение, α – коэффициент вовлеченности компонент уравнения, который может принимать значения из промежутка [0;1].

При расчете подбирается наилучшая степень вовлеченности каждой компоненты, модель для которой сумма квадратов остатков будет минимальна – считается лучшей.

6.6.17.7.2 Ограничения метода

- Блок работает только с временными рядами.
- В ряде должно быть поле ts_name, в атрибутах или же в первичном ключе.
- Наборы «Ряды независимых переменных X. Факт» и «Ряды независимых переменных X. Прогноз» должны иметь одинаковое количество независимых переменных.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.7 Управление порядком расчета

6.7.1 Блок «Входные данные»

6.7.1.1 Назначение

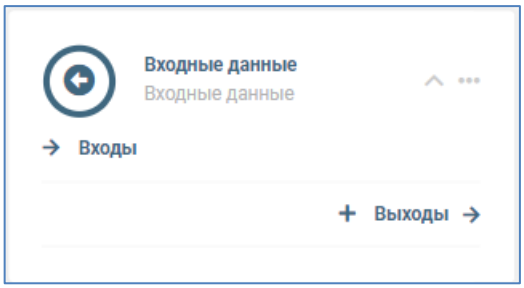


Рисунок 391 –Пример блока «Входные данные»

Технический блок, используемый для передачи данных из блоков «&НСИ» (п. 6.3.2) и «&Источник данных» (п. 6.3.1) в НСИ и источник данных соответственно.

Допустимо наличие только одного такого блока на графе.

6.7.1.2 Настройки блока

Название параметра	Тип параметра	Обязательный	Описание	Пример
Группа «Основные свойства»				
Входные данные	JSON	Нет	Массив идентификаторов входов с данными, которые будут переданы в выход блока при локальном расчете графа	

6.7.1.2.1 Динамические входы

Для создания входа необходимо нажать на знак «+» в разделе входов блока. Для каждого из входов задается следующий перечень полей:

Название входа	Тип	Обязательный	Описание
Идентификатор	Строка	Да	Идентификатор входа - используется внутри блока, для получения доступа к данным входа
Имя	Строка	Да	Имя входа для отображения в списке входов блока
Описание	Строка	Нет	Дополнительная информация с описанием входа
Тип	Выпадающий список	Да	Тип данных, которые будут приходить на вход. Нужен для построения корректных связей с другими блоками. По умолчанию установлен тип "Объект".
Обязательный	Флаг	Да	Обязательность входа. Если вход обязательный и на него не пришло данных - блок не будет рассчитан с ошибкой

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название входа	Тип	Обязательный	Описание
Массив	Флаг	Да	Является ли вход массивом данных
Обязательный	Флаг	Да	Обязательность входа. Если вход обязательный и на него не пришло данных - блок не будет рассчитан с ошибкой

6.7.1.3 Группа «Общие настройки блока»

Аналогично приведенным в п. 6.1.1.1.3.

6.7.1.4 Входы

У блока нет входов.

6.7.1.5 Выходы

Выходы задаются пользователем при создании блока. Для создания выхода необходимо нажать на знак + в разделе выходов блока. Для каждого из выходов задается следующий перечень полей:

Название выхода	Тип	Описание	Название выхода
Идентификатор	Строка	Да	Идентификатор выхода – используется внутри блока, для формирования выхода
Имя	Строка	Да	Имя выхода для отображения в списке выходов блока
Описание	Строка	Нет	Дополнительная информация с описанием выхода
Тип	Выпадающий список	Да	Тип данных, которые будут приходить на выход. Нужен для построения корректных связей с другими блоками
Обязательный	Флаг	Да	Обязательность выхода. Эту настройку можно проверить во время выполнения блока
Массив	Флаг	Да	Является ли выход массивом данных. Служит так же для проверки внутри блока

После создания нового выхода, его обновления или удаления будет автоматически изменен вход блоков «&НСИ» (п. 6.3.2) или «&Источник данных» (п. 6.3.1), у которых в параметрах была выбрана соответствующая структура.

6.7.1.6 Настройка привязок значений к конкретному выходу

Для возможности выбора необходимых значений из конкретного справочника (реализовано в блоке «Конструктор запросов» (см. п. 6.2.25)), реализован механизм привязки НСИ к любому из выходов блока.

Для этого в настройке «Описание» необходимо указать следующие параметры через разделитель «**;**» (точка с запятой):

- **dict** – флаг того, что включен выбор значений. Необходимо поставить 1 (один);
- **guid** – идентификатор НСИ;

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

- **id** – наименование первичного ключа, отвечающего за ключ элемента (по умолчанию id), не обязательное;
- **name –** наименование первичного ключа, отвечающего за ключ элемента (по умолчанию name), не обязательное.

Пример заполненных настроек:

```
dict: 1;
guid: 81f6129e-b527-4865-a6ae-3bbf07b5d612;
id: reg_id;
name: reg_name;
```

6.7.1.7 События блока

В блоке нет событий.

6.7.1.8 Описание работы блока

Блок позволяет получить входные данные для параметризации расчета НСИ или источника данных.

В момент локального расчета в качетсве входных данных берутся значения настройки «**Входные данные**».

При расчете через блоки «&НСИ» (п. 6.3.2) или «&Источник данных» (п. 6.3.1), данные на выход блока «Входные данные» будут взяты из входа блоков НСИ или Источник данных.

6.7.2 Блок «Выходные данные»

6.7.2.1 Назначение

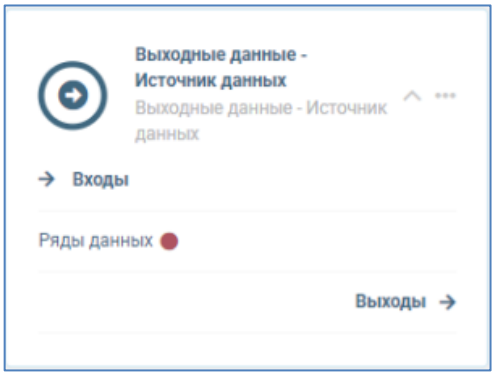


Рисунок 392 – Пример блока «Выходные данные»

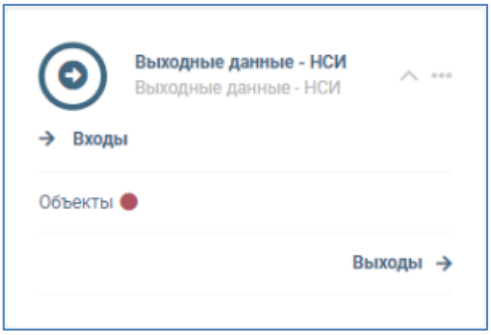


Рисунок 393 – Пример блока «Выходные данные – НСИ»

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

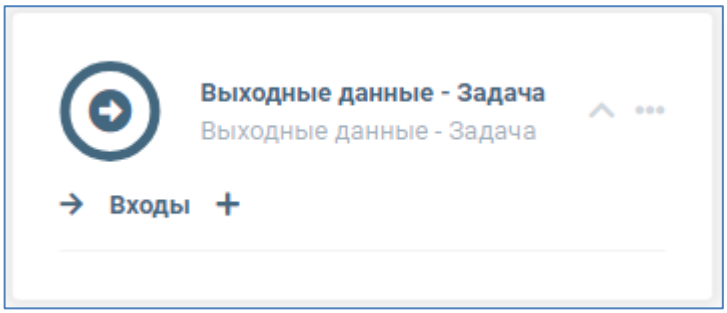


Рисунок 394 – Пример блока «Выходные данные – Задача»

Группа технических блоков, схожих по структуре, используемая для передачи данных из НСИ и источника данных в «&НСИ» (п. 6.3.2) или «&Источник данных» (п. 6.3.1) соответственно.

Автоматически создаются при создании НСИ или источника данных.

Не допускается их удаление или создание дубликатов.

6.7.2.2 Настройки блока

У блока нет специфических настроек.

6.7.2.3 Группа «Общие настройки блока»

Аналогично приведенным в п. 6.1.1.1.3.

6.7.2.4 Входы

Название входа	Тип	Описание
Объекты	Объект	Для блока «Выходные данные – НСИ" Вход с объектами, которые придут на выход блока «&НСИ» (п. 6.3.2)
Ряды данных	Ряд данных	Для блока «Выходные данные – Источник данных" Вход с рядами данных, которые придут на выход блока «&Источник данных» (п. 6.3.1)

6.7.2.5 Выходы

У блока нет выходов.

6.7.2.6 События блока

В блоке нет событий.

6.7.2.7 Описание работы блока

В момент локального расчета НСИ или источника данных блок не выполняет никаких функций. Однако, при расчете блоков «&НСИ» (п. 6.3.2) или «&Источник данных» (п. 6.3.1), только данные, переданные на вход блока «Выходные данные» будут переданы на выход блока, запустившего расчет задачи.

Иными словами, только с данными, переданными на выход блока «Выходные данные» можно продолжать работу вне НСИ или источника данных.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.7.3 Блок «Генератор события»

6.7.3.1 Назначение

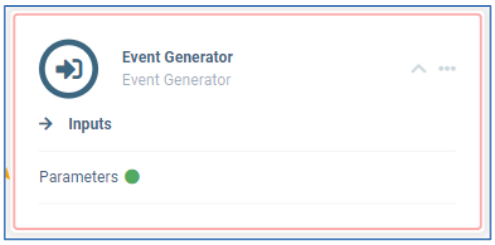


Рисунок 395 – Блок " Генератор события"

Блок для запуска расчета блоков "Обработчик события".

6.7.3.2 Настройки блока

Название параметра	Тип параметра	Обязательный	Описание	Пример
Номера событий	Раздел настроек	Да	Перечисления номеров событий, которые будут запускаться при расчете данного блока	
Номер события	Строка	Да	Номер события. Данный номер указывается такой же, который установлен в настройках блока "Обработчик события"	1

6.7.3.3 Группа «Общие настройки блока»

Аналогично приведенным в п. 6.1.1.1.3.

6.7.3.4 Входы

Название входа	Тип	Обязательный	Описание
Параметры	JSON	Да	Данные, с которыми будет рассчитываться связанные блоки "Обработчики событий"

6.7.3.5 Выходы

У блока нет выходов.

6.7.3.6 События блока

В блоке нет событий.

6.7.3.7 Описание работы блока

Блок вызывает на расчет цепочки графа, начиная с блоков "Обработчик событий". Список вызываемых блоков "Обработчик событий" задается в настройках блока.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.7.4 Блок «Обработчик события»

6.7.4.1 Назначение

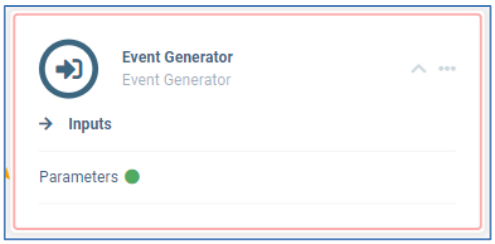


Рисунок 396 – Блок "Обработчик события"

Рассчитывает цепочку блоков. Вызывается блоком "Генератор события"

6.7.4.2 Настройки блока

Название параметра	Тип параметра	Обязательный	Описание	Пример
Номер события	Строка	Да	Номер события, который указывается в блоке "Генератор события"	1

6.7.4.3 Группа «Общие настройки блока»

Аналогично приведенным в п. 6.1.1.1.3.

6.7.4.4 Входы

У блока нет входов.

6.7.4.5 Выходы

Название входа	Тип	Обязательный	Описание
Параметры	JSON	Да	Параметры JSON Данные, которые пришли на вход блока "Генератор события"

6.7.4.6 События блока

В блоке нет событий.

6.7.4.7 Описание работы блока

Блок для расчета цепочек блоков. Блок вызывается блоков "Генератор события"

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.7.5 Блок «Диспетчер»

6.7.5.1 Назначение

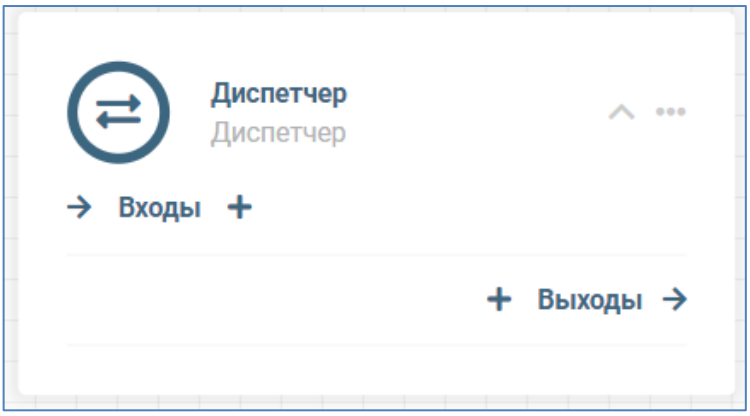


Рисунок 397 – Пример блока «Диспетчер»

Блок для управления данными по условию, которое задается в настройках блока.

6.7.5.2 Настройки блока

Название параметра	Тип параметра	Обязательный	Описание	Пример

Конфигурация	JSON	Да	Настройки, по которым будет рассчитываться блок. Обязательные атрибуты transitions и required. По умолчанию: `{"transitions": [], «required": []}`	**Пример:**
				`{`
				`"transitions": [`
				`{`
				`"in":"inp1",`
				`"out":"out1"`
				`},`
				`{`
				`"in":"inp2",`
				`"out":"out2",`
				`"cache": false`
				`},`
				`{`
				`"in":"inp3",`
				`"out":"out3",`
				`"cache": false`
				`}`
				`],`
				`"required": [`
				`["inp1", "inp2"],`
				`["inp3"]`
				`]`
				`}`

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
Останавливать расчет	Флаг	Нет	Надо ли рассчитывать блоки за блоком "Диспетчер", если не выполнились условия настройки "Конфигурация"	
Кэшировать визуализаторы			Надо ли очищать визуализаторы после ошибки блока или остановки на нем	

6.7.5.3 Группа «Общие настройки блока»

Аналогично приведенным в п. 6.1.1.1.3.

6.7.5.4 Входы

Входы задаются пользователем при создании блока. Для создания входа необходимо нажать на знак + в разделе входов блока. Для каждого из входов задается следующий перечень полей:

Название входа	Тип	Обязательный	Описание
Идентификатор	Строка	Да	Идентификатор входа – используется внутри блока, для получения доступа к данным входа
Имя	Строка	Да	Имя входа для отображения в списке входов блока
Описание	Строка	Нет	Дополнительная информация с описанием входа
Тип	Выпадающий список	Да	Тип данных, которые будут приходить на вход. Нужен для построения корректных связей с другими блоками. По умолчанию установлен тип "Объект".
Обязательный	Флаг	Да	Обязательность входа. Если вход обязательный и на него не пришло данных – блок не будет рассчитан с ошибкой
Массив	Флаг	Да	Является ли вход массивом данных

6.7.5.5 Выходы

Выходы задаются пользователем при создании блока. Для создания выхода необходимо нажать на знак + в разделе выходов блока. Для каждого из выходов задается следующий перечень полей:

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название выхода	Тип	Обязательный	Описание
Идентификатор	Строка	Да	Идентификатор выхода – используется внутри блока, для формирования выхода
Имя	Строка	Да	Имя выхода для отображения в списке выходов блока
Описание	Строка	Нет	Дополнительная информация с описанием выхода
Тип	Выпадающий список	Да	Тип данных, которые будут приходить на выход. Нужен для построения корректных связей с другими блоками
Обязательный	Флаг	Да	Обязательность выхода. Эту настройку можно проверить во время выполнения блока
Массив	Флаг	Да	Является ли выход массивом данных. Служит так же для проверки внутри блока

6.7.5.6 События блока

В блоке нет событий.

6.7.5.7 Описание работы блока

Блок предназначен для управления данными и расчетом на графе. Блок полученные данные проверяет на наличие переданных данных в блок и по условию выводит их в выходы блока. Условия задаются в настройках блока в поле «Конфигурация».
1. Поле transitions:
Данное поле отвечает за перенаправление данных из входов в выходы блока.

Поле transitions состоит из списка объектов. Объект состоит из:

- in - обязательного параметра id входа
- out - необязательных id выхода
- cache - использование кэширования данных. Возможные значения параметра: true/false. Данный параметр отвечает за то, какие данные будут браться при расчете блока. т.е. если блок, который передает данные в блок "Диспетчер" не участвует в расчете, то данные по этому блоку берутся с прошлого расчета. Если же кэширование выключено, то данные из такого блока не передаются в блок "Диспетчер".
    - По умолчанию кэширование данных включено (true).
- not_null - дополнительный параметр проверки на пустые данные на вход. Поле указывает на то, что если флаг включен, то при проверке required так же будет проводиться проверка поля на то, что он не пустой. Возможные значения параметра: true/false. Пустыми данными являются: None, пустой список ([]), пустой справочник ({}).
    - По умолчанию учет пустых значений выключен (false).

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Пример объекта: {"in":"id входа", "out":"id выхода", "cache": использование кэширования данных}, т.е. данные с **id** входа будут перенаправлены на **id** выход блока (если указанных **id** нет в блоке, блок выдаст об этом ошибку).

Использование кэширования данных бывает двух значений true/false. По умолчанию кэширование данных включено (true). Данный параметр отвечает за то, какие данные будут браться при расчете блока. т.е. если блок, который передает данные в блок «Диспетчер» не участвует в расчете, то данные по этому блоку берутся с прошлого расчета. Если же кэширование выключено, то данные из такого блока не передаются в блок «Диспетчер».

2. Поле required:

Данное поле отвечает за обязательность входов.

Поле **required** состоит из списка списков id входов.

Пример списка **id** входов: ["inp1", "inp2"], т.е. чтобы блок рассчитался на вход **inp1** и **inp2** должны прийти данные.

Если в поле **required** есть списки **id** входов, то обязательность входов проверяется по каждым спискам **id** входов отдельно, и успешностью расчета блока считается то, что хотя бы один из списков выполнился

Например:

```
  required": [
    ["inp1", «inp2"],
    ["inp3"]
]
```

Значит блок рассчитается в случаях:

1 – когда есть данные на входах "inp1", "inp2"

Или

2 – когда есть данные на входе "inp3"

Если в поле required не указаны списки id входов, то все входы считаются необязательными и блок будет рассчитываться без проверок на обязательность входов.

Если указан флаг "Останавливать расчет", то при расчете когда на вход не переданы данные или результатом блока является пустое значение, то расчет после этого блока не рассчитывает блоки и не вызывается ошибка у блока "Диспетчер". Иначе при данных условиях блок "Диспетчер" выполнится с ошибкой.

Если указан флаг "Кэшировать визуализаторы", то визуализаторы после блока "Диспетчер" если они не посчитались, у них остаются данные с прошлого расчета. Иначе, если во время расчета визуализатор не посчитался, таким визуализаторам ставится ошибка расчета.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.7.6 Блок «Итератор»

6.7.6.1 Назначение

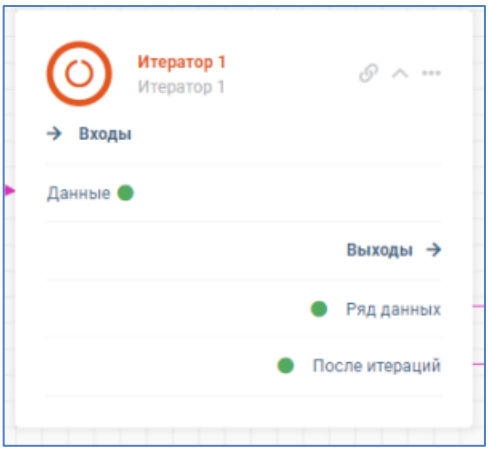


Рисунок 398 – Пример блока «Итератор»

Системный блок для управления расчетом (от латинского iteratio – «повторение»).

Блоки, которые находятся после выхода «Ряд данных», рассчитываются N раз (N зависит от длины списка данных, поступивших на вход «Данные»).

Блоки, которые находятся после выхода «После итераций», рассчитываются после N расчетов (после первого пункта).

6.7.6.2 Настройки блока

Название параметра	Тип параметра	Обязательный	Описание	Пример
Событие по умолчанию	Выпадающий список	Да	Данные данного шага будут запомнены во время расчета (с этими данными будут рассчитываться блоки, которые стоят после бока «Итератор», если их рассчитывать вне итератора). По умолчанию «Первый элемент»	

6.7.6.3 Группа «Общие настройки блока»

Аналогично приведенным в п. 6.1.1.1.3.

6.7.6.4 Входы

Название входа	Тип	Обязательный	Описание
Данные	Список	Да	Массив данных, по которому будет проходить цикл

6.7.6.5 Выходы

Название выхода	Тип	Описание
Ряд данных	Любой	Любая структура данных, которая была передана в элементах списка на вход блока

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

После итераций	Список	Список данных, который был собран блоком «Выход итератора» (см. п. 6.7.7)

6.7.6.6 События блока

В блоке нет событий.

6.7.6.7 Описание работы блока

Блок «Итератор» – это аналог цикла for, т.е. ветка графа, которая стоит после выхода «Ряд данных» выполняется N раз последовательно, где N – количество элементов списка данных, поступивших на вход «Данные».
После того как выполнятся все расчеты цикла, рассчитывается ветка графа, которая идет после выхода «После итераций».

Так же можно создать блок «Выход итератора». И на его входы передать результаты блоков, рассчитанных в цикле. Итоговый результат блока «Выход итератора» передается в выход «После итераций» блока «Итератор».

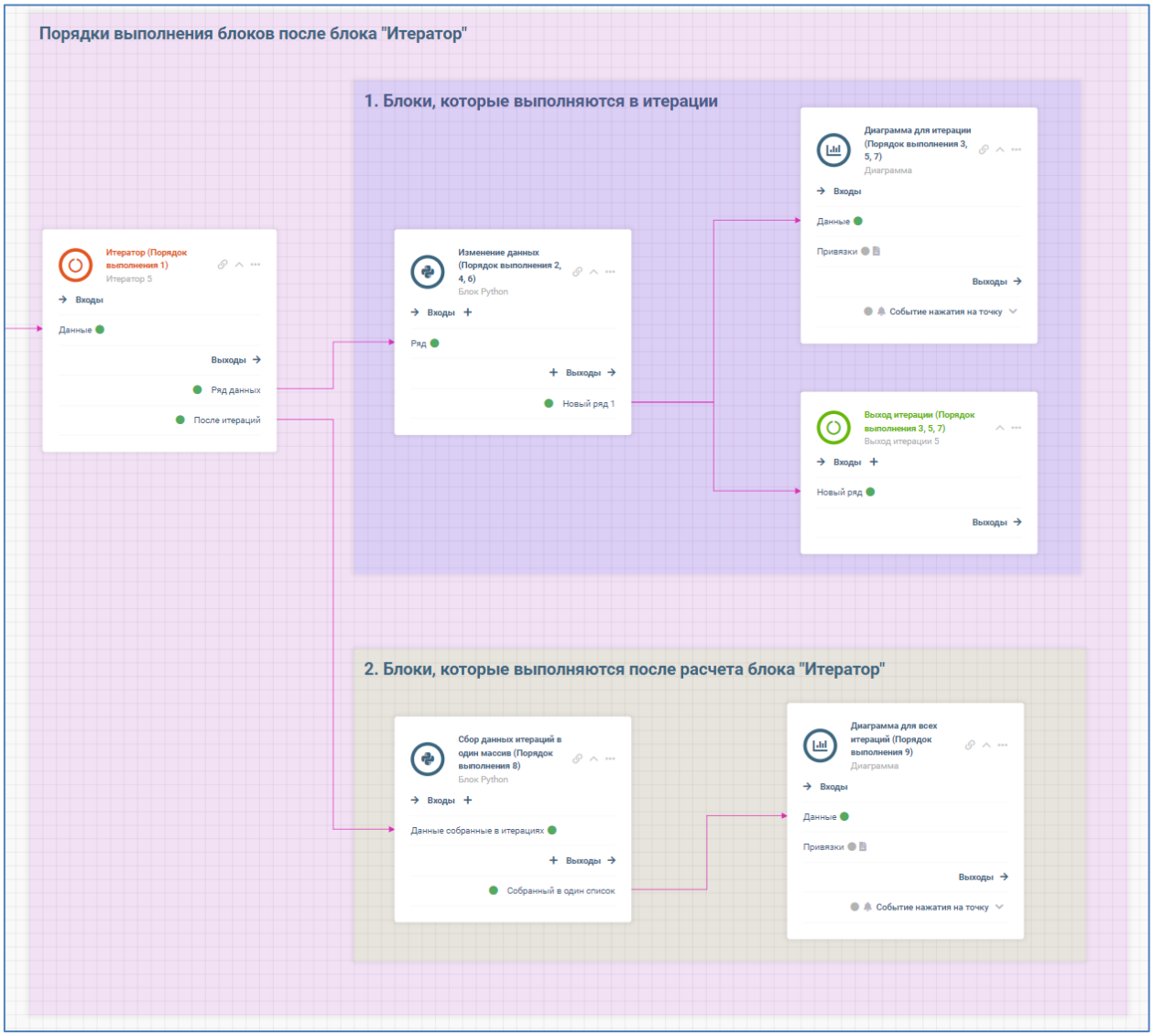


Рисунок 399 – Общий вид выполнения блоков после блока «Итератор»

1. Имеется массив из трех элементов типа данных «Ряд данных».

Пример данных:

```
[
```


© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

```
{
    "pkey": {},
    "attrs": {},
    "keys": [
        "01.01.2020",
        "02.01.2020",
        "03.01.2020",
        "04.01.2020",
        "05.01.2020",
        "06.01.2020",
        "07.01.2020",
        "08.01.2020",
        "09.01.2020",
        "10.01.2020"
    ],
    "vl": [
        0.3395592201661258,
        0.42789568809739664,
        0.2875478100139131,
        0.9992851882625527,
        0.502726781995488,
        0.03593920993304297,
        0.2124864158718153,
        0.9984123694858961,
        0.4632556469406398,
        0.9900078631434184
    ],
    "_source": "",
    "_params": {
        "ts": [
            {
                "key": "dt",
                "mask": "%d.%m.%Y",
                "frequency": "1D"
            }
        ],
        "vl": "value",
        "fkey": [
            "dt"
        ]
    }
},
{
    "pkey": {},
    "attrs": {},
    "keys": [
        "01.01.2020",
        "02.01.2020",
        "03.01.2020",
        "04.01.2020",
        "05.01.2020",
        "06.01.2020",
        "07.01.2020",
        "08.01.2020",
        "09.01.2020",
```


© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

```
            "10.01.2020"
        ],
        "vl": [
            0.48316072834559454,
            0.7159707276891684,
            0.001203040638855105,
            0.07097422089610954,
            0.11803521078925196,
            0.9868314719947913,
            0.017578828460893033,
            0.6495373736828195,
            0.00011687728450099985,
            0.28753142689724875
        ],
        "_source": "",
        "_params": {
            "ts": [
                {
                    "key": "dt",
                    "mask": "%d.%m.%Y",
                    "frequency": "1D"
                }
            ],
            "vl": "value",
            "fkey": [
                "dt"
            ]
        }
    },
    {
        "pkey": {},
        "attrs": {},
        "keys": [
            "01.01.2020",
            "02.01.2020",
            "03.01.2020",
            "04.01.2020",
            "05.01.2020",
            "06.01.2020",
            "07.01.2020",
            "08.01.2020",
            "09.01.2020",
            "10.01.2020"
        ],
        "vl": [
            0.046603716741338536,
            0.5777166449818106,
            0.8732493272498602,
            0.00753236956427512,
            0.8677426528302279,
            0.916651047337572,
            0.01420644651572065,
            0.21358290652916564,
            0.8082504968619894,
            0.15574500679550182
```


© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

```
        ],
        "_source": "",
        "_params": {
            "ts": [
                {
                    "key": "dt",
                    "mask": "%d.%m.%Y",
                    "frequency": "1D"
                }
            ],
            "vl": "value",
            "fkey": [
                "dt"
            ]
        }
    }
]
```

2. Эти данные передаются на вход «Данные» блока «Итератор».

3. Блок «Итератор» берет первый элемент списка, и передает его в выход «Ряд данных». (Начинается расчет 1 итерации).

Данные для 1 итерации:

```
{
    "pkey": {},
    "attrs": {},
    "keys": [
        "01.01.2020",
        "02.01.2020",
        "03.01.2020",
        "04.01.2020",
        "05.01.2020",
        "06.01.2020",
        "07.01.2020",
        "08.01.2020",
        "09.01.2020",
        "10.01.2020"
    ],
    "vl": [
        0.3395592201661258,
        0.42789568809739667,
        0.2875478100139131,
        0.9992851882625528,
        0.502726781995488,
        0.03593920993304297,
        0.2124864158718153,
        0.9984123694858961,
        0.4632556469406398,
        0.9900078631434184
    ],
    "_source": "",
    "_params": {
        "ts": [
            {
                "key": "dt",
```


© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

```
                "mask": "%d.%m.%Y",
                "frequency": "1D"
            }
        ],
        "vl": "value",
        "fkey": [
            "dt"
        ]
    }
}
```

4. Рассчитывается блок «Изменение данных».

5. Рассчитываются блоки «Диаграмма для итерации» и «Выход итерации» (Окончание расчета 1 итерации).

6. После окончания первой итерации, блок «Итератор» берет следующий элемент списка и передает его в выход «Ряд данных». (Начинается расчет 2 итерации).

Данные для 2 итерации:

```
{
    "pkey": {},
    "attrs": {},
    "keys": [
        "01.01.2020",
        "02.01.2020",
        "03.01.2020",
        "04.01.2020",
        "05.01.2020",
        "06.01.2020",
        "07.01.2020",
        "08.01.2020",
        "09.01.2020",
        "10.01.2020"
    ],
    "vl": [
        0.48316072834559456,
        0.7159707276891684,
        0.001203040638855105,
        0.07097422089610954,
        0.11803521078925196,
        0.9868314719947913,
        0.017578828460893035,
        0.6495373736828195,
        0.00011687728450099985,
        0.28753142689724878
    ],
    "_source": "",
    "_params": {
        "ts": [
            {
                "key": "dt",
                "mask": "%d.%m.%Y",
                "frequency": "1D"
            }
        ],
        "vl": "value",
```


© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

```
        "fkey": [
            "dt"
        ]
    }
}
```

7. Рассчитывается блок «Изменение данных».

8. Рассчитываются блоки «Диаграмма для итерации» и «Выход итерации» (Окончание расчета 2 итерации)

9. После окончания второй итерации, блок «Итератор» берет последний элемент списка и снова передает его в выход «Ряд данных». (Начинается расчет 3 итерации).

Данные для 3 итерации:

```
{
    "pkey": {},
    "attrs": {},
    "keys": [
        "01.01.2020",
        "02.01.2020",
        "03.01.2020",
        "04.01.2020",
        "05.01.2020",
        "06.01.2020",
        "07.01.2020",
        "08.01.2020",
        "09.01.2020",
        "10.01.2020"
    ],
    "vl": [
        0.046603716741338539,
        0.5777166449818106,
        0.8732493272498602,
        0.00753236956427512,
        0.8677426528302279,
        0.916651047337572,
        0.01420644651572065,
        0.21358290652916565,
        0.8082504968619894,
        0.15574500679550183
    ],
    "_source": "",
    "_params": {
        "ts": [
            {
                "key": "dt",
                "mask": "%d.%m.%Y",
                "frequency": "1D"
            }
        ],
        "vl": "value",
        "fkey": [
            "dt"
        ]
    }
}
```


© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

10. Рассчитывается блок «Изменение данных».

11. Рассчитываются блоки «Диаграмма для итерации» и «Выход итерации» (Окончание расчета 3 итерации)

12. После окончания расчета 3 итерации, блок «Итератор» передает рассчитанные данные из блока «Выход итерации» в выход «После итераций"

Данные собранные блоком «После итератора».


```
[
    {
        "inp": {
            "pkey": {},
            "attrs": {},
            "keys": [
                "01.01.2020",
                "02.01.2020",
                "03.01.2020",
                "04.01.2020",
                "05.01.2020",
                "06.01.2020",
                "07.01.2020",
                "08.01.2020",
                "09.01.2020",
                "10.01.2020"
            ],
            "vl": [
                0.6791184403322515,
                0.8557913761947933,
                0.8626434300417394,
                7.994281506100422,
                0,
                0.2515744695313008,
                0.6374592476154459,
                5.990474216915376,
                1.8530225877625592,
                5.94004717886051
            ],
            "_source": "",
            "_params": {
                "ts": [
                    {
                        "key": "dt",
                        "mask": "%d.%m.%Y",
                        "frequency": "1D"
                    }
                ],
                "vl": "value",
                "fkey": [
                    "dt"
                ]
            }
        }
    },
    {
        "inp": {
```


© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

```
            "pkey": {},
            "attrs": {},
            "keys": [
                "01.01.2020",
                "02.01.2020",
                "03.01.2020",
                "04.01.2020",
                "05.01.2020",
                "06.01.2020",
                "07.01.2020",
                "08.01.2020",
                "09.01.2020",
                "10.01.2020"
            ],
            "vl": [
                1.9326429133823781,
                3.579853638445842,
                0.001203040638855105,
                0.14194844179221908,
                0.11803521078925196,
                1.9736629439895825,
                0.17578828460893034,
                1.9486121210484586,
                0.00035063185350299957,
                0.5750628537944975
            ],
            "_source": "",
            "_params": {
                "ts": [
                    {
                        "key": "dt",
                        "mask": "%d.%m.%Y",
                        "frequency": "1D"
                    }
                ],
                "vl": "value",
                "fkey": [
                    "dt"
                ]
            }
        }
    },
    {
        "inp": {
            "pkey": {},
            "attrs": {},
            "keys": [
                "01.01.2020",
                "02.01.2020",
                "03.01.2020",
                "04.01.2020",
                "05.01.2020",
                "06.01.2020",
                "07.01.2020",
                "08.01.2020",
```


© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

```
                "09.01.2020",
                "10.01.2020"
            ],
            "vl": [
                0.46603716741338536,
                4.621733159854485,
                8.732493272498601,
                0.06025895651420096,
                7.809683875472051,
                1.833302094675144,
                0.12785801864148585,
                1.0679145326458281,
                1.6165009937239787,
                0
            ],
            "_source": "",
            "_params": {
                "ts": [
                    {
                        "key": "dt",
                        "mask": "%d.%m.%Y",
                        "frequency": "1D"
                    }
                ],
                "vl": "value",
                "fkey": [
                    "dt"
                ]
            }
        }
    }
]
```


13. Рассчитывается блок «Сбор данных итераций в один массив».

14. Рассчитывается блок «Диаграмма для всех итераций».

В режиме отладки можно посмотреть результаты блока (которые находятся после выхода «Ряд данных») на каждую итерацию:

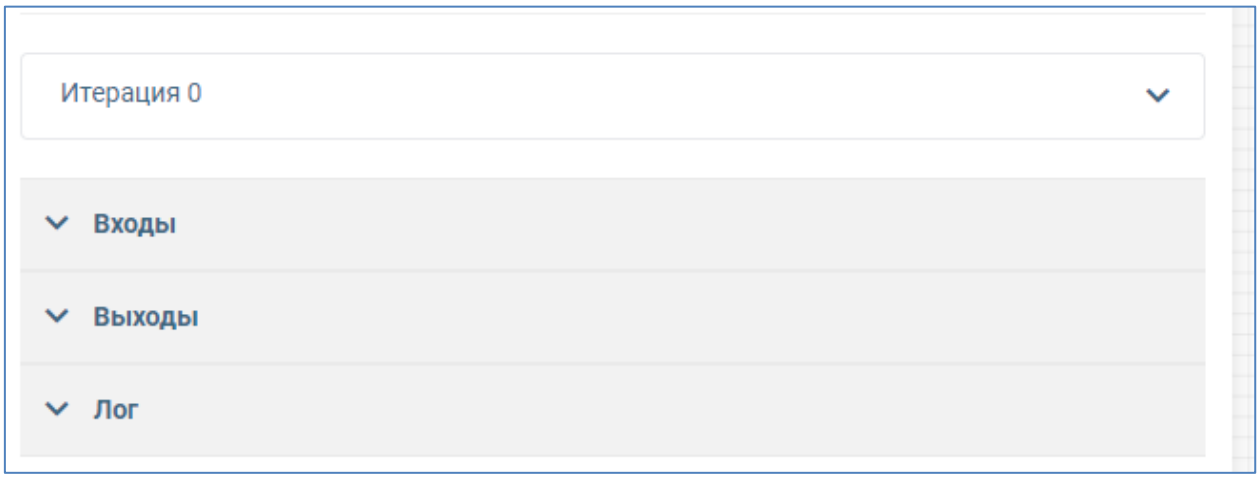


Рисунок 400 – Свойство блока «Итератор». Просмотр результатов расчета

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.7.7 Блок «Выход итератора»

В блоке нет выходов.

6.7.7.1 Назначение

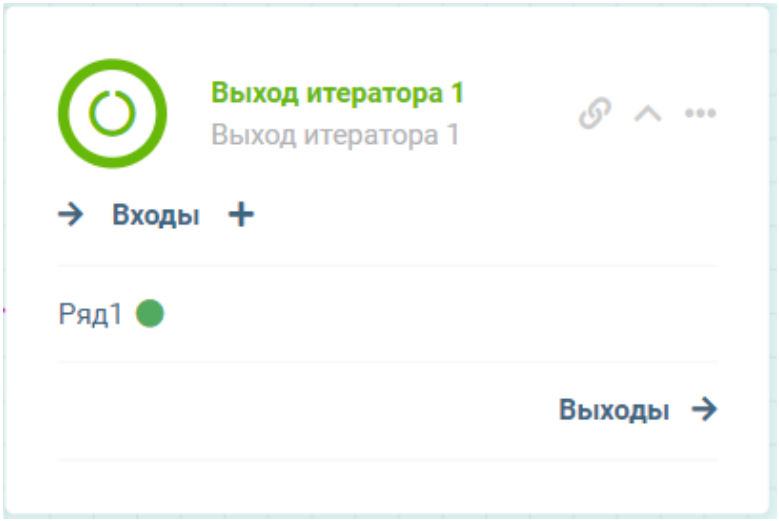


Рисунок 401 – Пример блока «Выход итератора»

Собирает значения, переданные в блок на каждой итерации, и по окончанию расчета всех итераций блока «Итератор» передает их на выход «Выход итератора» блока «Итератор».

6.7.7.2 Настройки блока

6.7.7.3 Группа «Общие настройки блока»

Аналогично приведенным в п. 6.1.1.1.3.

6.7.7.4 Входы

Входы задаются пользователем при создании блока. Для создания входа необходимо нажать на знак + в разделе входов блока. Для каждого из входов задается следующий перечень полей:

Название входа	Тип	Обязательный	Описание
Идентификатор	Строка	Да	Идентификатор входа – используется внутри блока, для получения доступа к данным входа
Имя	Строка	Да	Имя входа для отображения в списке входов блока
Описание	Строка	Нет	Дополнительная информация с описанием входа
Тип	Выпадающий список	Да	Тип данных, которые будут приходить на вход. Нужен для построения корректных связей с другими блоками. По умолчанию установлен тип "Объект".
Обязательный	Флаг	Да	Обязательность входа. Если вход обязательный и на него не пришло

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название входа	Тип	Обязательный	Описание
			данных – блок не будет рассчитан с ошибкой
Массив	Флаг	Да	Является ли вход массивом данных

6.7.7.5 Выходы

В блоке нет выходов.

6.7.7.6 События блока

В блоке нет событий.

6.7.7.7 Описание работы блока

Для каждого блока «Итератор» создается свой блок «Выход итератора».

id блока «Итератор» и блока «Выход итератора» должны совпадать, иначе на выход «После итератора» блока «Итератор» придет пустое значение.

Так же блок «Выход итератора» должен находиться после блока «Итератор» в итерируемой части.

Рисунок 402 – Пример последовательности расположения блока «Выход итератора»

Чтобы его создать на графе, необходимо у блока «Итератор» в меню блока нажать кнопку «Создать блок для выхода».

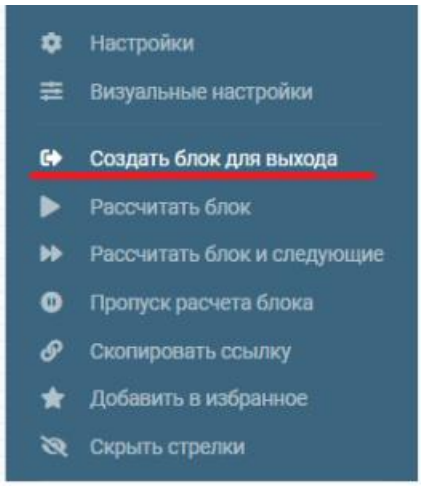
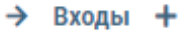


Рисунок 403 – Пример раздела меню «Создать блок для выхода»

После того как создастся блок, у него надо создать входы (Нажатием на + возле «Входы»)



Созданные входы надо соединить с выходами блоков, по которым нужно запомнить данные на каждую итерацию блока «Итератор».

После расчета всех итераций блока «Итератор», данные которые были переданы в блок «Выход итератора», можно получить из выхода «Выход итератора» блока «Итератор».

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Формат данных будет список из объектов ({«id входа»: «его значения», … }), а позиция в списке – это номер итерации.

Например, блок «Итератор» выполнялся 2 раза (на вход ему передано 2 ряда). В итерации обрабатывается блоком «Python», а затем обработанный ряд отправляется в блок «Диаграмму» и в блок «Выход итератора» (у блока создан вход «Новый ряд» с id «inp»).

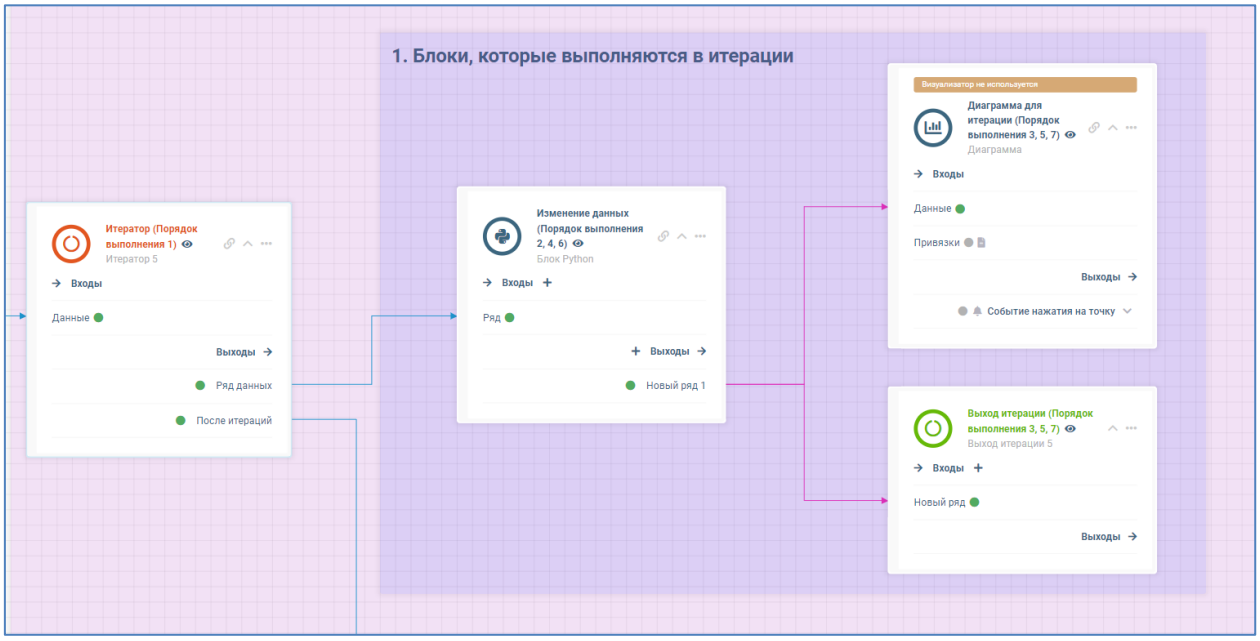


Рисунок 404 – Пример блоков, которые выполняются в итерации

Пример выхода «Выход итератора» у блока «Итератор»:


```
[
    {
        "inp": {
            "pkey": {},
            "attrs": {},
            "keys": [
                "01.01.2020",
                "02.01.2020",
                "03.01.2020",
                "04.01.2020",
                "05.01.2020",
                "06.01.2020",
                "07.01.2020",
                "08.01.2020",
                "09.01.2020",
                "10.01.2020"
            ],
            "vl": [
                0.6791184403322515,
                0.8557913761947933,
                0.8626434300417394,
                7.994281506100422,
                0,
                0.2515744695313008,
                0.6374592476154459,
                5.990474216915376,
                1.8530225877625592,
```


© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

```
            5.94004717886051
        ],
        "_source": "",
        "_params": {
            "ts": [
                {
                    "key": "dt",
                    "mask": "%d.%m.%Y",
                    "frequency": "1D"
                }
            ],
            "vl": "value",
            "fkey": [
                "dt"
            ]
        }
    }
},
{
    "inp": {
        "pkey": {},
        "attrs": {},
        "keys": [
            "01.01.2020",
            "02.01.2020",
            "03.01.2020",
            "04.01.2020",
            "05.01.2020",
            "06.01.2020",
            "07.01.2020",
            "08.01.2020",
            "09.01.2020",
            "10.01.2020"
        ],
        "vl": [
            1.9326429133823781,
            3.579853638445842,
            0.001203040638855105,
            0.14194844179221908,
            0.11803521078925196,
            1.9736629439895825,
            0.17578828460893034,
            1.9486121210484586,
            0.00035063185350299957,
            0.5750628537944975
        ],
        "_source": "",
        "_params": {
            "ts": [
                {
                    "key": "dt",
                    "mask": "%d.%m.%Y",
                    "frequency": "1D"
                }
            ],
```


© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

```
                "vl": "value",
                "fkey": [
                    "dt"
                ]
            }
        }
    }
]
```

6.7.8 Блок «Условие»

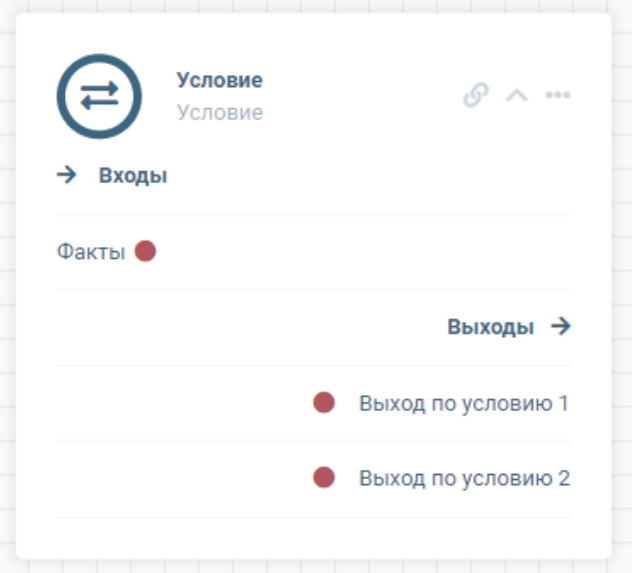


Рисунок 405 – Пример блока «Условие»

6.7.8.1 Назначение

Блок позволяет на основе правил определить выход для рядов, разделив их аналогично оператору IF .. THEN .. ELSE. Блок всегда оперирует массивом

```
Пример работы на псевдокоде
IF
   input.attrs["model-type"] == "ARIMA"
THEN
   return output2

IF
   input.attrs["model-type"] == "Trend"
THEN
   return output1
```

6.7.8.2 Настройки блока

Название параметра	Тип параметра	Обязательный	Описание	
Ссылка на документацию	Строка	Нет	Ссылка на эту страницу	
Описание блока	Строка	Нет	base64 строка для изображения,	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	
			которое может содержать	
Пропуск расчета блока	Флаг (true/false)	Да	Пропуск расчета блока при расчете графа (по умолчанию выключен)	

Правила форматирования	Строка (json)	Да	Правила, по которым производится проверка условия.
			В правилах могут быть условия по значениям атрибутов ряда (раздел properties)
			properties – поиск будет по значениям атрибутов ряда
			Ряд считается подходящим для правила только в случае **выполнения всех условий правила**.
			Внутри условия могут быть заданы сравнения
			equals
			greater
			less
			greater_or_equal
			less_or_equal
			not_equal
			При срабатывании правила ряды, которые подходят под него, будут перенаправлены на соответствующий выход. Для этого в разделе правила outputs необходимо указать куда подавать ряды, соответствующие правилу.
			
			"outputs": [
				<pre>{
				"rules": [
				{
				"rule": {
				"id": 1,
				"name": "Rule1",
				"description": "Rule 1 description",
				"conditions": [
				{
				"id": 1,
				"type": "CONDITION_TYPE_OBJECT",
				"alias": "object1",
				"properties": {
				"equals": {
				"model_type": "ARIMA"
				}
				}
				}
				],
				"result": {
				"outputs": [
				{
				"id": 1,
				"output": "input->output1"
				}
				]
				}
				}
				},</pre>

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	

			<pre>{
			«id": 2,
			«output": «input->output2"
			}</pre>
				<pre>{
				"rule": {
				"id": 2,
				"name": "Rule2",
				"description": "Rule 2 description",
				"conditions": [
				{
				"id": 1,
				"type": "CONDITION_TYPE_OBJECT",
				"alias": "object1",
				"properties": {
				"equals": {
				"model_type": "TREND"
				}
				}
				}
				],
				"result": {
				"outputs": [
				{
				"id": 2,
				"output": "input->output2"
				}
				]
				}
				}
				}
				]
				}</pre>

6.7.8.3 Входы

- **Факты**

Набор временных рядов, по которым блок будет производить расчет условия.

6.7.8.4 Выходы

- **Выход по условию 1**

Набор временных рядов, которые удовлетворяют правилу для первого выхода.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

- **Выход по условию 2**

Набор временных рядов, которые удовлетворяют правилу для второго выхода.

6.7.9 Блок «Пресет»

6.7.9.1 Назначение

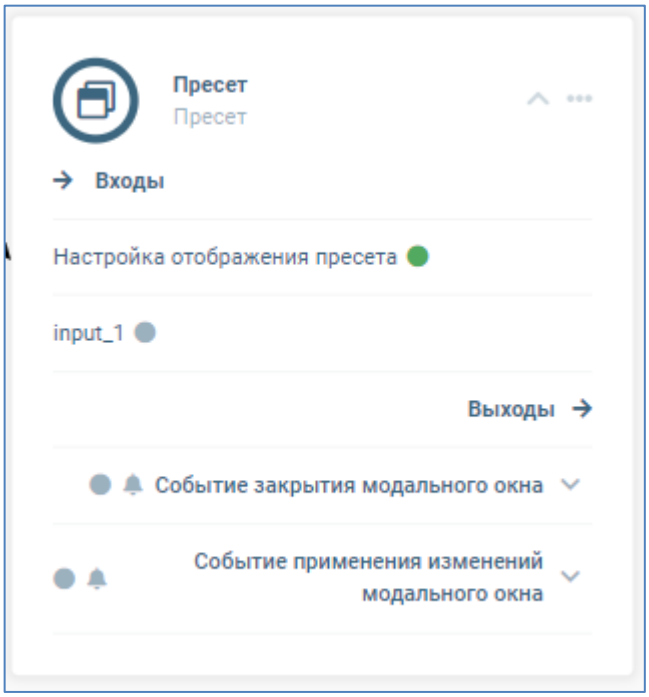


Рисунок 406 – Пример блока «Пресет»

Блок позволяет отобразить выбранный пресет в качестве визуализатора..

6.7.9.2 Настройки блока

Название параметра	Тип параметра	Обязательный	Описание	Пример
Заголовок блока	Строка	Нет	Наименование блока, которое отобразится в приложении	
Задача	Выпадающий список	Да	Выбор задачи, из которой будет выбран пресет для отображения	
Пресет	Выпадающий список	Да	Выбор пресета для отображения. Возможность выбора появляется при заданной задаче	

6.7.9.3 Группа «Общие настройки блока»

Аналогично приведенным в п. 6.1.1.1.3.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.7.9.4 Входы

Название входа	Тип параметра	Обязательный	Описание	Пример

Настройки отображения пресета	JSON	Да	Параметры для отображения блока на пресете.
			• visible 0/1 - скрытие/отображения визуализатора в текущем пресете.
			По умолчанию параметр visible равен нулю.
				{
				"visible":
				0
				}

Кроме этого, если в выбранной задаче присутствует блок "Входные данные", то все выходы блока будут добавлены как входы в блок "Пресет".

6.7.9.5 Выходы

Если в выбранной задаче присутствует блок "Выходные данные", то все входы блока будут добавлены как выходы в блок "Пресет".

6.7.9.6 События блока

При включенной настройке модальных окон в пресете, будут добавлены два события. Эти выходы предустановлены и изменить их нельзя.

Название выхода	Тип	Описание
Событие закрытия модального окна	JSON	Данные, пришедшие в граф при закрытии окна, без применения изменений.
Событие применения изменений модального окна	JSON	Изменения, которые переданы в граф

6.7.9.7 Описание работы блока

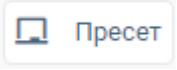
Блок проверяет наличие выбранной задачи/пресета и передает эти данные в визуализатор.

Если пресет был отмечен как модальное окно, то эта информация так же передается в визуализатор.

6.7.9.8 Описание работы визуализатора

Блок используется отображения одного пресета внутри блока в другом пресете.

Если выбранный пресет был отмечен как модальное окно, то оно будет показано только при передаче во вход параметра visible = 1.

Без этого, визуалиратор будет представлять собой технический блок, видимый только при редактировании блока: 

При отображении модального окна будет виден визуализатор с двумя кнопками:

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Рисунок 407 – Пример визуализатора

Ок - кнопка передает выбранные значения в выход "Применить измененеия" и закрывает окно.

Закрыть - кнопка закрывает окно и передает значения в выход "Закрытие окна".

6.7.9.9 Настройки визуализатора

Визуализатор не имеет специфических настроек. Общий список приведен в п. 4.2.8 «Настройки визуализатора».

6.8 Форматирование

6.8.1 Блок «Стили таблицы»

6.8.1.1 Назначение

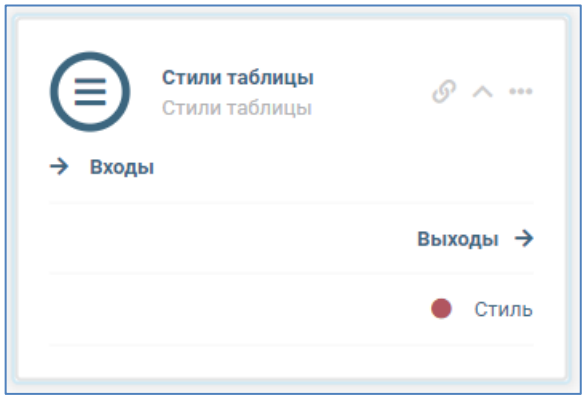


Рисунок 408 – Пример блока «Стили таблицы»

Блок позволяет задать конкретные стили для таблицы.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.8.1.2 Настройки блока

Название параметра	Тип параметра	Обязательный	Описание	Пример
Отображать в визуализаторе	Флаг	Да	Позволяет включать и отключать отображение блока на пресете. По умолчанию – не активно	
Раздел «Настройки стилей»				
Название стиля	Строка	Нет		
Использовать по умолчанию	Флаг	Да	Использовать выбранный стиль в качестве элемента по умолчанию. По умолчанию – не активно	
Раздел «Шапка»				
Отображать шапку	Флаг	Да	Позволяет включать и отключать отображение шапки таблицы. По умолчанию – активно	
Шапка – Отображать боковик	Флаг	Да	Позволяет включать и отключать отображение боковика шапки таблицы. По умолчанию – не активно	
Шапка – Боковик – Цвет обводки	Цвет	Нет	Позволяет задать цвет обводки боковика шапки таблицы	
Шапка – Боковик – Толщина обводки	Вещественное число	Нет	Позволяет задать толщину обводки боковика шапки таблицы. По умолчанию установлено значение – 1	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
Шапка – Боковик – Цвет заливки	Цвет	Нет	Позволяет задать цвет заливки ячеек боковика шапки таблицы	
Шапка – Боковик – Выравнивание текста	Выпадающий список	Да	Позволяет задать выравнивание текста в ячейке боковика шапки таблицы. По умолчанию – По левому краю	
Шапка – Боковик – Цвет текста	Цвет	Нет	Позволяет задать цвет текста в ячейках боковика шапки таблицы	
Шапка – Данные – Цвет обводки	Цвет	Нет	Позволяет задать цвет обводки данных в шапке таблице	
Шапка – Данные – Толщина обводки	Вещественное число	Нет	Позволяет задать толщину обводки данных в шапке таблицы. По умолчанию установлено значение – 1	
Шапка – Данные – Цвет заливки	Цвет	Нет	Позволяет задать цвет заливки ячеек данных в шапке таблицы	
Шапка – Данные – Выравнивание текста	Выпадающий список	Да	Позволяет задать выравнивание текста в ячейке данных в шапке таблице. По умолчанию – По левому краю	
Шапка – Данные – Цвет текста	Цвет	Нет	Позволяет задать цвет текста в ячейках данных шапки таблицы	
Раздел «Строки»				

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
Строки – Боковик – Отображать боковик	Флаг	Да	Позволяет включать и отключать отображение боковика строк таблицы. По умолчанию – активно	
Строки – Боковик – Цвет обводки	Цвет	Нет	Позволяет задать цвет обводки боковика в строках таблицы	
Строки – Боковик – Толщина обводки	Вещественное число	Нет	Позволяет задать толщину обводки боковика в строках таблицы. По умолчанию установлено значение – 1	
Строки – Боковик – Цвет заливки	Цвет	Нет	Позволяет задать цвет заливки ячеек боковика строк таблицы	
Строки – Боковик – Выравнивание текста	Выпадающий список	Да	Позволяет задать выравнивание текста в ячейке боковика строк таблицы. По умолчанию – По левому краю	
Строки – Боковик – Цвет текста	Цвет	Нет	Позволяет задать цвет текста в ячейках боковика строк таблицы	
Отображать чередующиеся строки	Флаг	Да	Позволяет включать и отключать отображение цветом четных и нечетные строки в таблице. По умолчанию – активно	
Нечетные Строки – Цвет обводки	Цвет	Нет	Позволяет задать цвет обводки ячеек нечетных строк таблицы	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
Нечетные Строки – Толщина обводки	Вещественное число	Да	Позволяет задать толщину обводки ячеек нечетных строк таблицы. По умолчанию установлено значение – 1	
Нечетные Строки – Цвет заливки	Цвет	Нет	Позволяет задать цвет заливки ячеек нечетных строк таблицы	
Нечетные Строки – Выравнивание текста	Выпадающий список	Да	Позволяет задать выравнивание текста ячеек нечетных строк таблицы. По умолчанию – По левому краю	
Нечетные Строки – Цвет текста	Цвет	Нет	Позволяет задать цвет текста в ячейках нечетных строка таблицы.	
Четные Строки – Цвет обводки	Цвет	Нет	Позволяет задать цвет обводки ячеек четных строк таблицы	
Четные Строки – Толщина обводки	Вещественное число	Да	Позволяет задать толщину обводки ячеек четных строк таблицы. По умолчанию установлено значение – 1	
Четные Строки – Цвет заливки	Цвет	Нет	Позволяет задать цвет заливки ячеек четных строк таблицы	
Четные Строки – Выравнивание текста	Выпадающий список	Да	Позволяет задать выравнивание текста ячеек четных строк таблицы. По умолчанию – По левому краю	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
Четные Строки – Цвет текста	Цвет	Нет	Позволяет задать цвет текста в ячейках четных строка таблицы	

6.8.1.3 Группа «Общие настройки блока»

Аналогично приведенным в п. 6.1.1.1.3.

6.8.1.4 Входы

В блоке нет входов.

6.8.1.5 Выходы

- **Стиль**

Выбранный стиль для таблицы. Соединяется с отдельным входом блока «Таблица» (п. 6.1.2.3).

- **Событие смены стиля таблицы**

Динамически изменяет стиль на выбранный. Соединяется с отдельным входом блока «Таблица» (п. 6.1.2.3).

6.8.1.6 События блока

При включенной активной настройке «Отображать визуализатор» в списке визуализаторов приложения появляется его визуализатор, который можно разместить на странице пользовательского приложения.

6.8.1.7 Описание работы блока

Блок «Стили таблицы» позволяет использовать задавать и передавать стили таблиц для дальнейшего отображения в пользовательском приложении. В настройках блока присутствуют как преднастроенные стили, так и стили которые может создать и настроить пользователь самостоятельно. Настройки разбиты на глобальные разделы «Шапка» и «Строки». Так же при включении настройки «Отображать в визуализаторе» появляется возможность разместить визуализатор в пользовательском приложении и динамически изменять стили таблиц в приложении.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Блок с открытыми настройками стиля

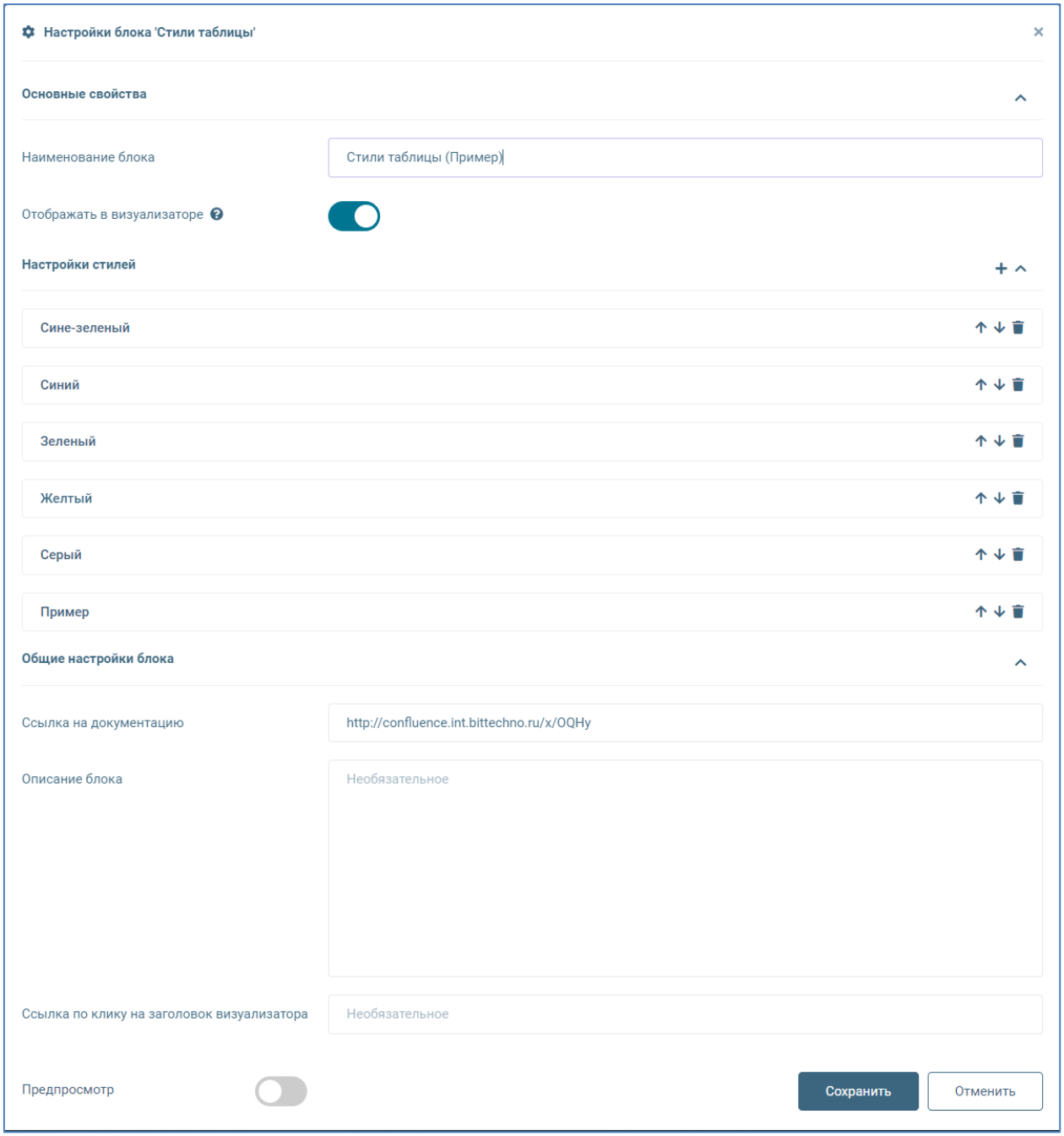


Рисунок 409 – Настройки блока «Стили таблицы»

6.8.1.8 Описание работы визуализатора

Внешний вид визуализатора на пресете.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

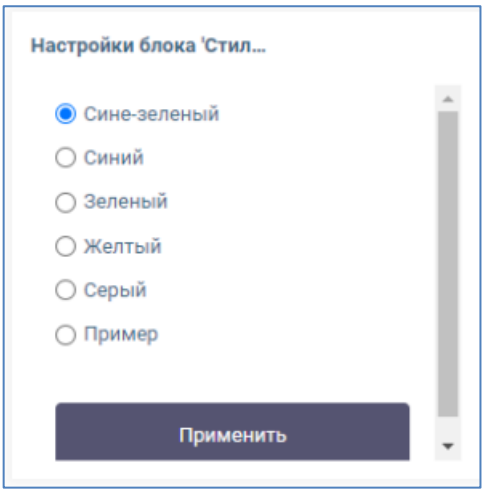


Рисунок 410 – Пример визуализатора

Для изменения стиль необходимо выбрать стиль из списка, нажав на него ЛКМ, а затем нажать кнопку «Применить».

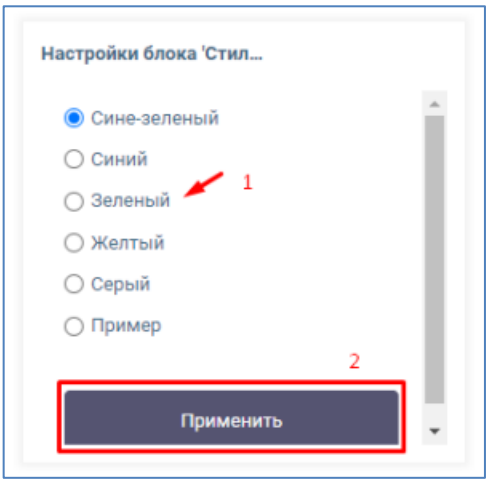


Рисунок 411 – Пример выбора стиля

6.8.1.9 Настройки визуализатора

Визуализатор не имеет специфических настроек. Общий список приведен в п. 4.2.8 «Настройки визуализатора».

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.8.2 Блок «Форматирование по условию»

6.8.2.1 Назначение

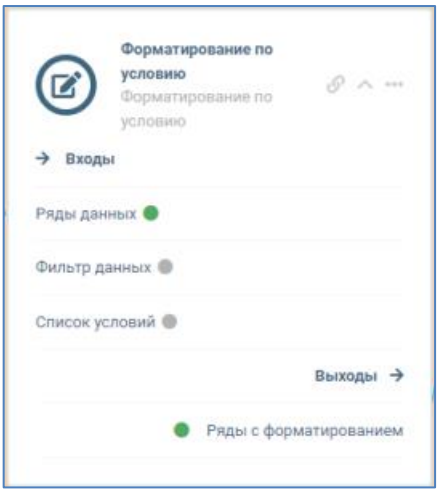


Рисунок 412 – Пример блока «Форматирование по условию»

Блок позволяет по значениям данных или атрибутов рядов осуществлять условное форматирование.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.8.2.2 Настройки блока

Название параметра	Тип параметра	Обязательный	Описание	Пример
Диапазоны	Раздел настроек	Да	Описание правила применения условного форматирования.	
Название диапазона	Строка	Нет	Название диапазона	Диапазон 1
Область применения	Выпадающий список	Да	К чему будет применяться фильтр: по значениям, по атрибутам, по ключам(датам) или смешанный вариант	
Идентификатор	Строка	Нет	Заполняется при выборе области применения по **атрибутам**. Задается id атрибута	ts_name
Тип диапазона	Выпадающий список	Да	Какой диапазон будет применятся в условном форматировании: всегда, между, не между, равно, не равно, больше, больше или равно, меньше, меньше или равно	
Значение (дата)	Дата и время	Нет	Заполняется при выборе области применения по **ключам(датам)**. Для всех типов диапазона, кроме между и не между	
Значение 1 (дата)	Дата и время	Нет	Заполняется при выборе области применения по **ключам(датам)**. Для типа диапазона между и не между	
Значение 2 (дата)	Дата и время	Нет	Заполняется при выборе области применения по **ключам(датам)**. Для типа диапазона между и не между	
Значение (строка)	Строка	Нет	Заполняется при выборе области применения по **атрибутам**.	
Значение (число)	Вещественное число	Нет	Заполняется при выборе области применения по **значениям**. Для всех типов диапазона, кроме между и не между	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
Значение 1	Вещественное число	Нет	Заполняется при выборе области применения по **значениям**. Для типа диапазона между и не между	
Значение 2	Вещественное число	Нет	Заполняется при выборе области применения по **значениям**. Для типа диапазона между и не между	

Значение	JSON	Нет	Заполняется при выборе области применения **смешанного** варианта. Описание формирование json. см ниже	**Пример:**
				`[`
				`{`
				`"area": "Value",`
				`"type": "Less",`
				`"value": 0`
				`},`
				`{`
				`"area": "Attribute",`
				`"id": "ts_name",`
				`"type": "Equal",`
				`"value": "x8"`
				`},`
				`{`
				`"area": "KeysData",`
				`"type": "Between",`
				`"value": "01.11.2020",`
				`"value2": "01.11.2021"`
				`}`
				`]`

Цвет заливки	Цвет	Нет	Цвет заливки	
Цвет границы	Цвет	Нет	Цвет границы	
Толщина границы	Целое число	Нет	Толщина границы	
Стиль границы	Выпадающий список	Нет	Стиль границы	
Текст (подписи) – размер шрифта	Целое число	Нет	Размер шрифта текста подписи	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
Текст (подписи) – цвет текста	Цвет	Нет	Цвет текста подписи	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.8.2.3 Группа «Общие настройки блока»

Аналогично приведенным в п. 6.1.1.1.3.

6.8.2.4 Входы

Название входа	Тип	Обязательный	Описание
Ряды данных	Ряд данных	Да	Данные, по которым будет осуществляется условное форматирование
Фильтр данных	Ряд данных	Нет	Фильтр для входных данных перед применением правил форматирования
Список условий	Ряд данных	Нет	Список описаний правил применения условного форматирования. Если в этот вход пришли описания правил, то **правила описанные с блоке игнорируются**. Структура описаний правил описана ниже в разделе «Список условий со входа блока»

6.8.2.5 Выходы

Название выхода	Тип	Описание
Ряды с форматированием	Ряд данных	Данные с форматированием

6.8.2.6 События блока

В блоке нет событий.

6.8.2.7 Описание работы блока

В блок приходят данные. Собираются правила для условного форматирования из настроек блока (или из входа «Список условий»). И в те данные, которые удовлетворяют правилам, добавляется атрибут «format_conditions» в котором будет содержаться условное форматирование.

Пример данных (10 рядов), на которых ниже будут строиться примеры, приведен на рисунке (Рисунок 413).

	ID	2020-01-01	2020-02-01	2020-03-01	2020-04-01	2020-05-01	2020-06-01	2020-07-01	2020-08-01	2020-09-01	2020-10-01	2020-11-01	2020-12-01	2021-01-01
1	x1	28.79	31.60	162.05	23.81	75.13	51.62	48.24	26.78	96.23	46.76	62.77	18.85	45.63
2	x2	16.00	11.72	136.03	9.57	30.92	21.15	17.35	13.75	46.51	15.19	29.91	16.46	22.71
3	x3	2.64	2.51	4.06	2.33	3.16	2.90	3.03	2.55	4.15	2.57	3.33	2.43	3.04
4	x4	0.39	0.53	0.03	0.61	0.31	0.24	0.25	0.42	0.07	0.33	0.24	0.52	0.22
5	x5	0.19	0.37	0.01	0.39	0.18	0.13	0.15	0.21	0.03	0.19	0.12	0.29	0.12
6	x6	27.70	26.27	116.74	27.21	-5.52	50.87	8.57	29.63	110.35	24.75	-33.90	25.12	-18.39
7	x7	22.01	23.82	-64.85	-6.46	28.55	43.18	38.09	20.45	85.04	43.87	57.58	-6.54	-20.64
8	x8	-8.20	-9.95	-45.21	-7.15	21.83	9.81	10.58	-2.92	43.18	-6.68	23.77	-16.93	-14.31
9	x9	40.09	6.36	76.96	10.44	4.64	3.61	4.79	0.81	4.85	7.87	9.91	11.46	0.38
10	x10	81.88	205.26	256.78	116.73	356.46	238.09	219.08	237.13	248.22	102.25	323.09	162.84	178.25

Рисунок 413 – Пример данных (10 рядов)

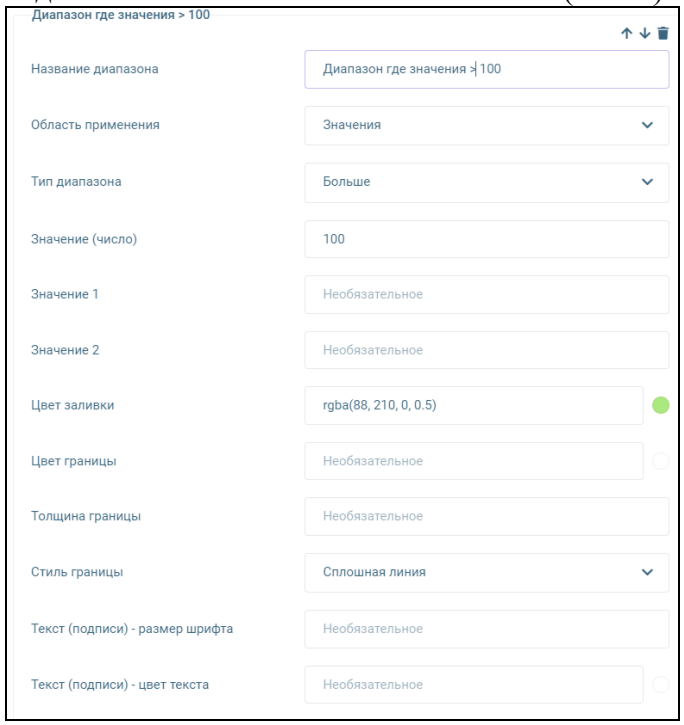
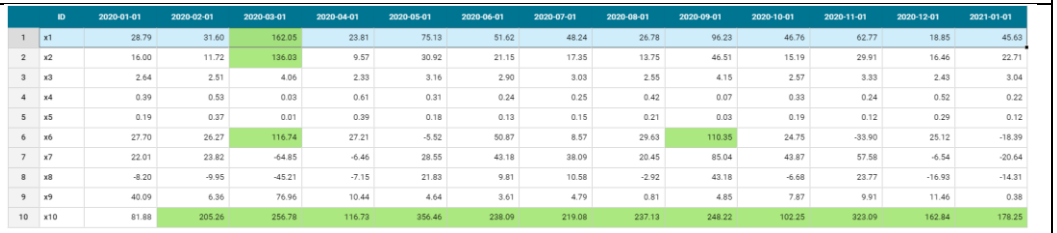
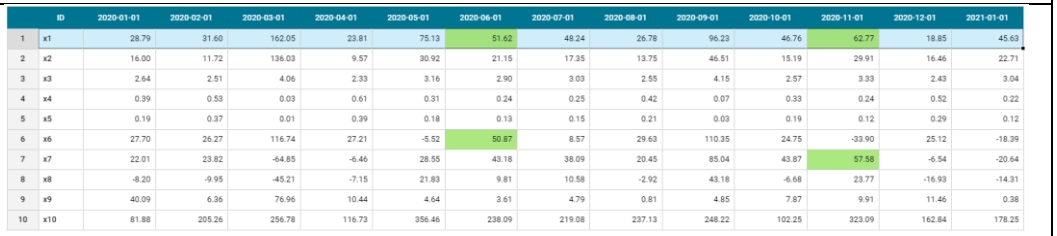
Описание работы областей применения («Область применения» в настройках блока):

1) **По значениям**

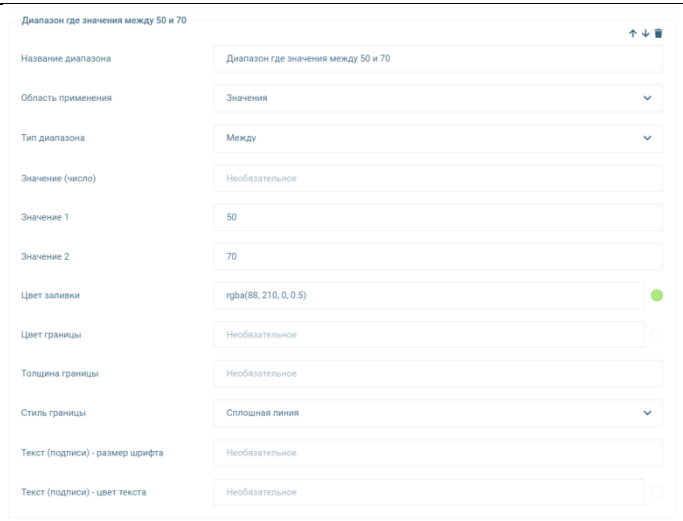
При данном варианте будут сравниваться каждые значения в каждом ряду с заданным условием. И если условие совпадает, раскрашивается конкретное значение

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Например:

<table>
<tr><th>Условие</th><th>Настройки блока</th><th>Результат</th></tr>
<tr>
<td>Покрасить данные, которые имеют значение больше 100</td>
<td>Выбирается «Область применения» – «Значения».
Выбирается «Тип диапазона» – «Больше».
Задается значение 100 в поле «Значение (число)».
 
</td>
<td>  </td>
</tr>
<tr>
<td>Покрасить данные, которые имеют значения</td>
<td>Выбирается «Область применения» – «Значения».
Выбирается «Тип диапазона» – «Между».
Задается значение 50 в поле «Значение 1».
Задается значение 70 в поле «Значение 2».</td>
<td>  </td>
</tr>
</table>

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Условие	Настройки блока	Результат
между 50 и 70.		

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

2) По атрибутам

При данном варианте будут сравниваться атрибут в каждом ряду с заданным условием. И если атрибут совпадает, раскрашивается весь ряд.

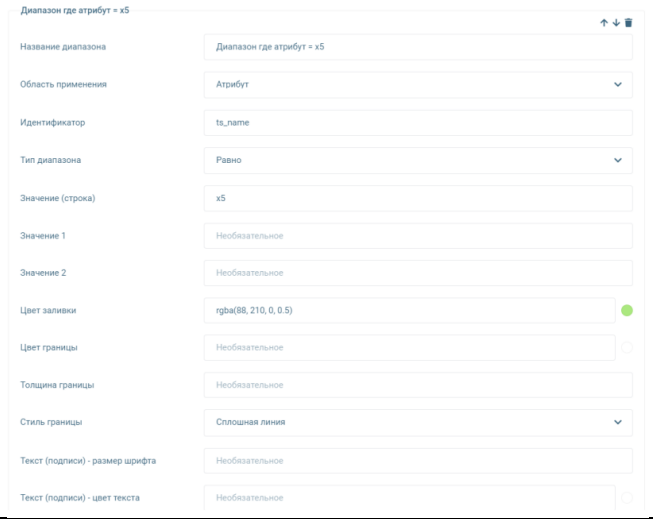
При выборе атрибута, надо задать его идентификатор.

К строковым атрибутам может применяться только "Тип диапазона": всегда, равно или не равно.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Например:

Условие	Настройки блока	Результат

Покрасить ряд, у которого атрибут «ts_name» равен «x5».
Пример описания ряда:
0.1215
],
"attrs": {
"ts_name": "x5",
"model_name": "x5"
},
"_source": "ed98f4bb-c3f7-4f58-9739-c433e4c99a07",
"_params": {
"ts": [
{
	Выбирается «Область применения» – «Атрибут».
	Задается «Идентификатор» равный «ts_name».
	Выбирается «Тип диапазона» – «Равно».
	Задается значение x5 в поле «Значение (строка)»
	 	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

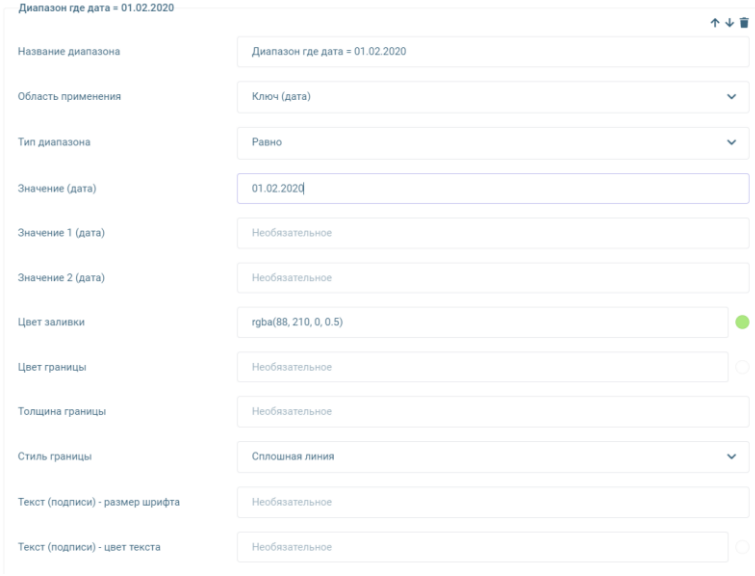
3) **По ключам (датам)**

При данном варианте будут сравниваться ключи (только ключи состоящие из дат) в каждом ряду с заданным условием. И если ключ совпадает, раскрашивается значение на данную дату.

Дата в полях «Значение (дата)», «Значение 1 (дата)», «Значение 2 (дата)» задается в формате День.Месяц.Год Час:Минута:Секунда.

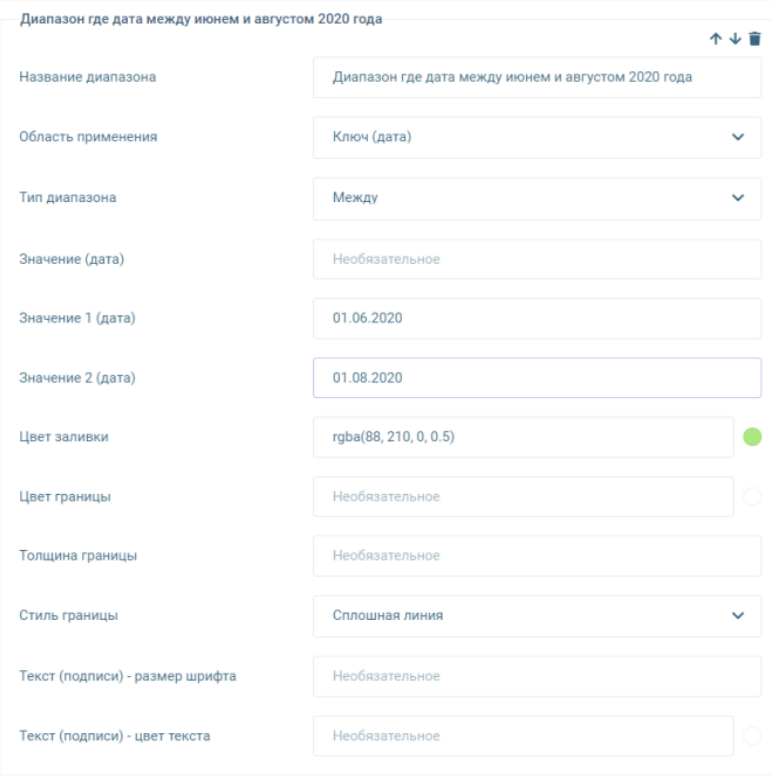
© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Например:

Условие	Настройки блока	Результат
Покрасить данные за февраль 2020 года	Выбирается «Область применения» – «Ключ (дата)». Выбирается «Тип диапазона» – «Равно». Задается значение 01.02.2020 в поле «Значение (дата)». 	

	ID	2020-01-01	2020-02-01	2020-03-01	2020-04-01	2020-05-01	2020-06-01	2020-07-01	2020-08-01	2020-09-01	2020-10-01	2020-11-01	2020-12-01
1	x1	28.79	31.60	162.05	23.81	75.13	51.62	48.24	26.78	96.23	46.76	62.77	18.85
2	x2	16.00	11.72	136.03	9.57	30.92	21.15	17.35	13.75	46.51	15.19	29.91	16.46
3	x3	2.64	2.51	4.06	2.33	3.16	2.90	3.03	2.55	4.15	2.57	3.33	2.43
4	x4	0.39	0.53	0.03	0.61	0.31	0.24	0.25	0.42	0.07	0.33	0.24	0.52
5	x5	0.19	0.37	0.01	0.39	0.18	0.13	0.15	0.21	0.03	0.19	0.12	0.29
6	x6	27.70	26.27	116.74	27.21	-5.52	50.87	8.57	29.63	110.35	24.75	-33.90	25.12
7	x7	22.01	23.82	-64.85	-6.46	28.55	43.18	38.09	20.45	85.04	43.87	57.58	-6.54
8	x8	-8.20	-9.95	-45.21	-7.15	21.83	9.81	10.58	-2.92	43.18	-6.68	23.77	-16.93
9	x9	40.09	6.36	76.96	10.44	4.64	3.61	4.79	0.81	4.85	7.87	9.91	11.46
10	x10	81.88	205.26	256.78	116.73	356.46	238.09	219.08	237.13	248.22	102.25	323.09	162.84

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

<table>
<tr><th>Условие</th><th>Настройки блока</th><th>Результат</th></tr>
<tr>
<td>Покрасить данные в промежуток между июнем и августом 2020 года</td>
<td>Выбирается «Область применения» – «Ключ (дата)».
Выбирается «Тип диапазона» – «Между».
Задается значение 01.06.2020 в поле «Значение 1 (дата)».
Задается значение 01.08.2020 в поле «Значение 1 (дата)».
 
</td>
<td>
<table>
<tr><th></th><th>ID</th><th>2020-01-01</th><th>2020-02-01</th><th>2020-03-01</th><th>2020-04-01</th><th>2020-05-01</th><th>2020-06-01</th><th>2020-07-01</th><th>2020-08-01</th><th>2020-09-01</th><th>2020-10-01</th></tr>
<tr><td>1</td><td>x1</td><td>28.79</td><td>31.60</td><td>162.05</td><td>23.81</td><td>75.13</td><td>51.62</td><td>48.24</td><td>26.78</td><td>96.23</td><td>46.76</td></tr>
<tr><td>2</td><td>x2</td><td>16.00</td><td>11.72</td><td>136.03</td><td>9.57</td><td>30.92</td><td>21.15</td><td>17.35</td><td>13.75</td><td>46.51</td><td>15.79</td></tr>
<tr><td>3</td><td>x3</td><td>2.64</td><td>2.51</td><td>4.06</td><td>2.33</td><td>3.16</td><td>2.90</td><td>3.03</td><td>2.55</td><td>4.15</td><td>2.67</td></tr>
<tr><td>4</td><td>x4</td><td>0.39</td><td>0.53</td><td>0.03</td><td>0.61</td><td>0.31</td><td>0.24</td><td>0.25</td><td>0.42</td><td>0.07</td><td>0.23</td></tr>
<tr><td>5</td><td>x5</td><td>0.19</td><td>0.37</td><td>0.01</td><td>0.39</td><td>0.18</td><td>0.13</td><td>0.15</td><td>0.21</td><td>0.03</td><td>0.19</td></tr>
<tr><td>6</td><td>x6</td><td>27.70</td><td>26.27</td><td>116.74</td><td>27.21</td><td>-5.52</td><td>50.87</td><td>8.57</td><td>29.63</td><td>110.35</td><td>24.75</td></tr>
<tr><td>7</td><td>x7</td><td>22.01</td><td>23.82</td><td>-64.85</td><td>-6.46</td><td>28.55</td><td>43.18</td><td>38.09</td><td>20.45</td><td>85.04</td><td>43.67</td></tr>
<tr><td>8</td><td>x8</td><td>-8.20</td><td>-9.95</td><td>-45.21</td><td>-7.15</td><td>21.83</td><td>9.81</td><td>10.58</td><td>-2.92</td><td>43.18</td><td>-6.48</td></tr>
<tr><td>9</td><td>x9</td><td>40.09</td><td>6.36</td><td>76.96</td><td>10.44</td><td>4.64</td><td>3.61</td><td>4.79</td><td>0.81</td><td>4.85</td><td>7.87</td></tr>
<tr><td>10</td><td>x10</td><td>81.88</td><td>205.26</td><td>256.78</td><td>116.73</td><td>356.46</td><td>238.09</td><td>219.08</td><td>237.13</td><td>248.22</td><td>102.25</td></tr>
</table>
</td>
</tr>
</table>

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

4) **Смешанный вариант**

При данном варианте будут сравниваться все ранее описанные способы задания условия одновременно. И если условие совпадет, раскрашивается значение

При смешанном диапазоне задается json. Json представляет из себя массив справочников.

Атрибуты справочника:

- area – Область применения:
  - Value – по значениям;
  - Attribute – по атрибуту;
  - KeysData – по ключу (дате);
- id – Идентификатор атрибута. Заполняется если выбрана область применения по атрибутам;
- type – Тип диапазона:
  - Always – всегда;
  - Between – между;
  - NotBetween – не между;
  - Equal – равен;
  - NotEqual – не равен;
  - Greater – больше;
  - Less – меньше;
  - GreaterOrEqual – больше или равно;
  - LessOrEqual – меньше или равно;
- value – Сравниваемое значение;
- value2 – Второе значение для промежутков.

По области применения может быть только одно условие. Для области по атрибутам одно условие на идентификатор.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Например:

<table>
<tr><th>Условие</th><th>Настройки блока</th><th>Результат</th></tr>
<tr>
<td>Покрасить данные за февраль 2020 года, где значения в промежутке между 10 и 100</td>
<td>Выбирается «Область применения» – «Смешанная»
В поле «Значение» задается
<pre>[
    {
        "area": "Value",
        "type": "Between",
        "value": 10,
        "value2": 100
    },
    {
        "area": "KeysData",
        "type": "Equal",
        "value": "01.02.2020"
    }
]</pre></td>
<td>
<table>
<tr><th></th><th>ID</th><th>2020-01-01</th><th>2020-02-01</th><th>2020-03-01</th><th>2020-04-01</th><th>2020-05-01</th><th>2020-06-01</th><th>2020-07-01</th><th>2020-08-01</th><th>2020-09-01</th><th>2020-10-01</th><th>2020-11-01</th><th>2020-12-01</th><th>2021-01-01</th></tr>
<tr><td>1</td><td>x1</td><td>28.79</td><td>31.60</td><td>162.05</td><td>23.81</td><td>75.13</td><td>51.62</td><td>48.24</td><td>26.78</td><td>96.23</td><td>46.76</td><td>62.77</td><td>18.85</td><td>45.63</td></tr>
<tr><td>2</td><td>x2</td><td>16.00</td><td>11.72</td><td>136.03</td><td>9.57</td><td>30.92</td><td>21.15</td><td>17.35</td><td>13.75</td><td>46.51</td><td>15.19</td><td>29.91</td><td>16.46</td><td>22.71</td></tr>
<tr><td>3</td><td>x3</td><td>2.64</td><td>2.51</td><td>4.06</td><td>2.33</td><td>3.16</td><td>2.90</td><td>3.03</td><td>2.55</td><td>4.15</td><td>2.57</td><td>3.33</td><td>2.43</td><td>3.04</td></tr>
<tr><td>4</td><td>x4</td><td>0.39</td><td>0.53</td><td>0.03</td><td>0.61</td><td>0.31</td><td>0.24</td><td>0.25</td><td>0.42</td><td>0.07</td><td>0.33</td><td>0.24</td><td>0.52</td><td>0.22</td></tr>
<tr><td>5</td><td>x5</td><td>0.19</td><td>0.37</td><td>0.01</td><td>0.39</td><td>0.18</td><td>0.13</td><td>0.15</td><td>0.21</td><td>0.03</td><td>0.19</td><td>0.12</td><td>0.29</td><td>0.12</td></tr>
<tr><td>6</td><td>x6</td><td>27.70</td><td>26.27</td><td>116.74</td><td>27.21</td><td>-5.52</td><td>50.87</td><td>8.57</td><td>29.63</td><td>110.35</td><td>24.75</td><td>-33.90</td><td>25.12</td><td>-18.39</td></tr>
<tr><td>7</td><td>x7</td><td>22.01</td><td>23.82</td><td>-64.85</td><td>-6.46</td><td>28.55</td><td>43.18</td><td>38.09</td><td>20.45</td><td>85.04</td><td>43.87</td><td>57.58</td><td>-6.54</td><td>-20.64</td></tr>
<tr><td>8</td><td>x8</td><td>-8.20</td><td>-9.95</td><td>-45.21</td><td>-7.15</td><td>21.83</td><td>9.81</td><td>10.58</td><td>-2.92</td><td>43.18</td><td>-6.68</td><td>23.77</td><td>-16.93</td><td>-14.31</td></tr>
<tr><td>9</td><td>x9</td><td>40.09</td><td>6.36</td><td>76.96</td><td>10.44</td><td>4.64</td><td>3.61</td><td>4.79</td><td>0.81</td><td>4.85</td><td>7.87</td><td>9.91</td><td>11.46</td><td>0.38</td></tr>
<tr><td>10</td><td>x10</td><td>81.88</td><td>205.26</td><td>256.78</td><td>116.73</td><td>356.46</td><td>238.09</td><td>219.08</td><td>237.13</td><td>248.22</td><td>102.25</td><td>323.09</td><td>162.84</td><td>178.25</td></tr>
</table>
</td>
</tr>
<tr>
<td>Покрасить данные с ноября 2020 года по ноябрь 2021, где значения меньше 0 и атрибут ts_name = x8</td>
<td>Выбирается «Область применения» – «Смешанная»
В поле «Значение» задается
<pre>[
    {
        "area": "Value",</pre></td>
<td>
<table>
<tr><th></th><th>ID</th><th>2020-01-01</th><th>2020-02-01</th><th>2020-03-01</th><th>2020-04-01</th><th>2020-05-01</th><th>2020-06-01</th><th>2020-07-01</th><th>2020-08-01</th><th>2020-09-01</th><th>2020-10-01</th><th>2020-11-01</th><th>2020-12-01</th><th>2021-01-01</th></tr>
<tr><td>1</td><td>x1</td><td>28.79</td><td>31.60</td><td>162.05</td><td>23.81</td><td>75.13</td><td>51.62</td><td>48.24</td><td>26.78</td><td>96.23</td><td>46.76</td><td>62.77</td><td>18.85</td><td>45.63</td></tr>
<tr><td>2</td><td>x2</td><td>16.00</td><td>11.72</td><td>136.03</td><td>9.57</td><td>30.92</td><td>21.15</td><td>17.35</td><td>13.75</td><td>46.51</td><td>15.19</td><td>29.91</td><td>16.46</td><td>22.71</td></tr>
<tr><td>3</td><td>x3</td><td>2.64</td><td>2.51</td><td>4.06</td><td>2.33</td><td>3.16</td><td>2.90</td><td>3.03</td><td>2.55</td><td>4.15</td><td>2.57</td><td>3.33</td><td>2.43</td><td>3.04</td></tr>
<tr><td>4</td><td>x4</td><td>0.39</td><td>0.53</td><td>0.03</td><td>0.61</td><td>0.31</td><td>0.24</td><td>0.25</td><td>0.42</td><td>0.07</td><td>0.33</td><td>0.24</td><td>0.52</td><td>0.22</td></tr>
<tr><td>5</td><td>x5</td><td>0.19</td><td>0.37</td><td>0.01</td><td>0.39</td><td>0.18</td><td>0.13</td><td>0.15</td><td>0.21</td><td>0.03</td><td>0.19</td><td>0.12</td><td>0.29</td><td>0.12</td></tr>
<tr><td>6</td><td>x6</td><td>27.70</td><td>26.27</td><td>116.74</td><td>27.21</td><td>-5.52</td><td>50.87</td><td>8.57</td><td>29.63</td><td>110.35</td><td>24.75</td><td>-33.90</td><td>25.12</td><td>-18.39</td></tr>
<tr><td>7</td><td>x7</td><td>22.01</td><td>23.82</td><td>-64.85</td><td>-6.46</td><td>28.55</td><td>43.18</td><td>38.09</td><td>20.45</td><td>85.04</td><td>43.87</td><td>57.58</td><td>-6.54</td><td>-20.64</td></tr>
<tr><td>8</td><td>x8</td><td>-8.20</td><td>-9.95</td><td>-45.21</td><td>-7.15</td><td>21.83</td><td>9.81</td><td>10.58</td><td>-2.92</td><td>43.18</td><td>-6.68</td><td>23.77</td><td>-16.93</td><td>-14.31</td></tr>
<tr><td>9</td><td>x9</td><td>40.09</td><td>6.36</td><td>76.96</td><td>10.44</td><td>4.64</td><td>3.61</td><td>4.79</td><td>0.81</td><td>4.85</td><td>7.87</td><td>9.91</td><td>11.46</td><td>0.38</td></tr>
<tr><td>10</td><td>x10</td><td>81.88</td><td>205.26</td><td>256.78</td><td>116.73</td><td>356.46</td><td>238.09</td><td>219.08</td><td>237.13</td><td>248.22</td><td>102.25</td><td>323.09</td><td>162.84</td><td>178.25</td></tr>
</table>
</td>
</tr>
</table>

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Условие	Настройки блока	Результат
	"type": "Less", "value": 0 }, { "area": "Attribute", "id": "ts_name", "type": "Equal", "value": "x8" }, { "area": "KeysData", "type": "Between", "value": "01.11.2020", "value2": "01.11.2021" }]	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.8.2.8 Список условий со входа блока

Описание формата входа «Список условий».

На вход передается список объектов.

Описание объекта (все ключи соответствуют полям из настройки блока):

- name – Название диапазона;
- app_area – Область применения:
  - 0 – Значения;
  - 1 – Атрибут;
  - 2 – Ключ (дата);
  - 3 – Смешанная;
- id_attribute – Идентификатор;
- type – Тип диапазона:
  - 0 – Всегда;
  - 1 – Между;
  - 2 – Не между;
  - 3 – Равно;
  - 4 – Не равно;
  - 5 – Больше;
  - 6 – Меньше;
  - 7 – Больше или равно;
  - 8 – Меньше или равно;
- val_str – Значение (строка). Заполняется при выборе области применения по **атрибутам**;
- val0 – Значение (число). Заполняется при выборе области применения по **значениям**. Для всех типов диапазона, кроме между и не между;
- val1 – Значение 1. Заполняется при выборе области применения по **значениям**. Для типа диапазона между и не между;
- val2 – Значение 2. Заполняется при выборе области применения по **значениям**. Для типа диапазона между и не между;
- val_js – Значение. Заполняется при выборе области применения **смешанного** варианта;
- val_dt_0 – Значение (дата). Заполняется при выборе области применения по **ключам(датам)**. Для всех типов диапазона, кроме между и не между;
- val_dt_1 – Значение 1 (дата). Заполняется при выборе области применения по **ключам(датам)**. Для типа диапазона между и не между;
- val_dt_2 – Значение 2 (дата). Заполняется при выборе области применения по **ключам(датам)**. Для типа диапазона между и не между;
- fillcolor – Цвет заливки. Задается цветом;
- bordercolor – Цвет границы. Задается цветом;

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

- borderweight – Толщина границы;
- borderstyle – Стиль границы:
  - 0 – Сплошная линия;
  - 1 – Пунктир;
- fontsize – Текст (подписи) – размер шрифта;
- fontcolor – Текст (подписи) – цвет текста. Задается цветом.

Цвета можно задавать так:

- #6f97f3;
- rgba(245, 36, 36, 0.5).

Пример со всеми полями:


```
[
    {
        "name": "Диапазон 1",
        "type": 3,
        "val0": "",
        "val1": "",
        "val2": "",
        "val_js": "[]",
        "val_str": "",
        "app_area": 2,
        "fontsize": "",
        "val_dt_0": "01.02.2020",
        "val_dt_1": "",
        "val_dt_2": "",
        "fillcolor": "rgba(245, 36, 36, 0.5)",
        "fontcolor": "",
        "bordercolor": "",
        "borderstyle": 0,
        "borderweight": "",
        "id_attribute": ""
    },
    {
        "name": "Диапазон 2",
        "type": 3,
        "val0": "",
        "val1": "",
        "val2": "",
        "val_js": "[]",
        "val_str": "x5",
        "app_area": 1,
        "fontsize": "",
        "val_dt_0": "",
        "val_dt_1": "",
        "val_dt_2": "",
        "fillcolor": "rgba(255, 192, 37, 0.5)",
        "fontcolor": "",
        "bordercolor": "",
        "borderstyle": 0,
        "borderweight": "",
        "id_attribute": "ts_name"
```


© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

```
    },
    {
        "name": "Диапазон 3",
        "type": 1,
        "val0": "",
        "val1": 4.0,
        "val2": 5.0,
        "val_js": "[]",
        "val_str": "",
        "app_area": 0,
        "fontsize": "",
        "val_dt_0": "",
        "val_dt_1": "",
        "val_dt_2": "",
        "fillcolor": "rgba(95, 195, 1, 0.5)",
        "fontcolor": "",
        "bordercolor": "",
        "borderstyle": 0,
        "borderweight": "",
        "id_attribute": ""
    },
    {
        "name": "Диапазон 4",
        "type": 3,
        "val0": "",
        "val1": "",
        "val2": "",
        "val_js": "[\r\n    {\r\n        \"area\": \"Value\",\r\n   \"type\":
\"Less\",\r\n          \"value\": 0\r\n     },\r\n     {\r\n          \"area\":
\"Attribute\",\r\n      \"id\": \"ts_name\",\r\n      \"type\": \"Equal\",\r\n
\"value\": \"x8\"\r\n      },\r\n      {\r\n       \"area\": \"KeysData\",\r\n
\"type\":  \"Between\",\r\n                    \"value\":  \"01.11.2020\",\r\n
\"value2\": \"01.11.2021\"\r\n    }\r\n]",
        "val_str": "",
        "app_area": 3,
        "fontsize": "",
        "val_dt_0": "",
        "val_dt_1": "",
        "val_dt_2": "",
        "fillcolor": "rgba(0, 232, 229, 0.5)",
        "fontcolor": "",
        "bordercolor": "",
        "borderstyle": 0,
        "borderweight": "",
        "id_attribute": ""
    }
]
```


Пример с короткой записью:


```
[
    {
        "type": 3,
        "app_area": 2,
        "val_dt_0": "01.02.2020",
        "fillcolor": "rgba(245, 36, 36, 0.5)",
    },
```


© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

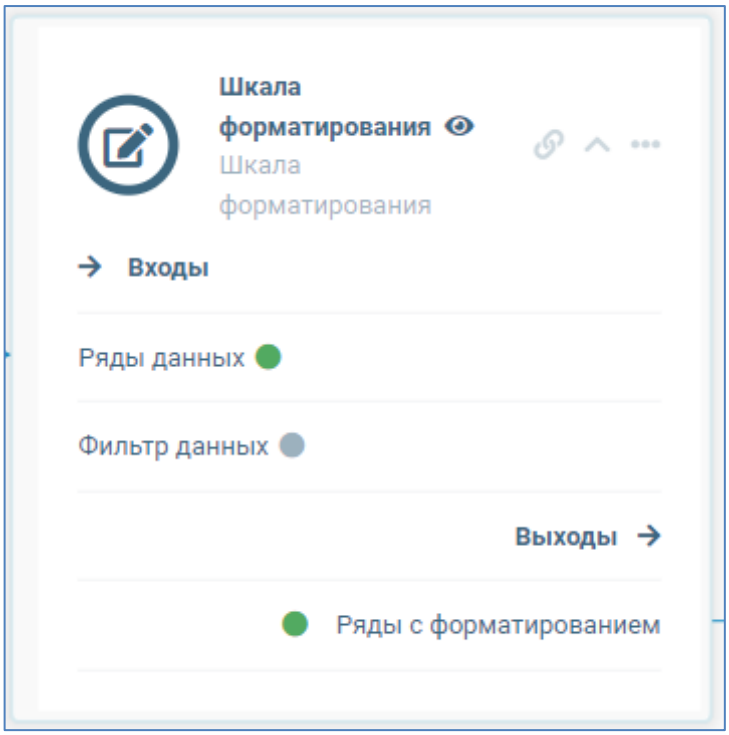
```
    {
        "type": 3,
        "val_str": "x5",
        "app_area": 1,
        "fillcolor": "rgba(255, 192, 37, 0.5)",
        "id_attribute": "ts_name",
    },
    {
        "type": 1,
        "val1": 4.0,
        "val2": 5.0,
        "app_area": 0,
        "fillcolor": "rgba(95, 195, 1, 0.5)",
    },
    {
        "type": 3,
        "val_js": "[\r\n    {\r\n        \"area\": \"Value\",\r\n    \"type\":
\"Less\",\r\n          \"value\": 0\r\n     },\r\n     {\r\n          \"area\":
\"Attribute\",\r\n       \"id\": \"ts_name\",\r\n       \"type\": \"Equal\",\r\n
\"value\": \"x8\"\r\n     },\r\n     {\r\n        \"area\": \"KeysData\",\r\n
\"type\":  \"Between\",\r\n                    \"value\":  \"01.11.2020\",\r\n
\"value2\": \"01.11.2021\"\r\n    }\r\n]",
        "app_area": 3,
        "fillcolor": "rgba(0, 232, 229, 0.5)",
    }
]
```


6.8.2.9 Дополнительно

Описание применяемых стилей приведено в п. **Error! Reference source not found.** «Блок «Форматирование по условию (расширенное)».

6.8.3 Блок «Шкала форматирования»

6.8.3.1 Назначение



© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Рисунок 414 – Пример блока «Шкала форматирования»

Блок для форматирования рядов по условию

6.8.3.2 Настройки блока

Название параметра	Тип параметра	Обязательный	Описание	Пример
Считать проценты	Флаг	Нет	Если флаг установлен, блок сначала рассчитает процент для каждого значения, где наибольше значение в ряде будет 100%, а наименьшее – 0%. Затем применит форматирование к значениям процентов.	
Диапазоны				
Диапазон больше (X > Значение)				
Тип диапазона	Выпадающий список	Да	Диапазон сравнения со значением (Больше или Больше и равно)	Больше
Значение	Вещественное число	Да	Значение условия, по которому будет формирование ряда	150.5
Цвет заливки	Цвет	Нет	Цвет заливки при выполнении условия	#aae2e8
Цвет границы	Цвет	Нет	Цвет границы при выполнении условия	#ff7e7e
Толщина границы	Целое число	Нет	Толщина границы при выполнении условия. Минимальное значение 0	1
Стиль границы	Выпадающий список	Нет	Стиль границы при выполнении условия (Сплошная или Пунктир)	Сплошная
Текст (подписи) – размер шрифта	Целое число	Нет	Текст (подписи) – размер шрифта при выполнении условия. Минимальное значение 1	12

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
Текст (подписи) – цвет текста	Цвет	Нет	Текст (подписи) – цвет текста при выполнении условия	#69e411
Раздел настроек *Здесь можно добавить промежутки диапазонов*				
Название диапазона	Строка	Да	Наименование диапазона	Между 90 и 150.5
Тип диапазона	Выпадающий список	Да	Диапазон сравнения со значением (Больше или Больше и равно)	Больше и равно
Значение	Вещественное число	Да	Значение условия, по которому будет формирование ряда	90
Цвет заливки	Цвет	Нет	Цвет заливки при выполнении условия	#ef6767
Цвет границы	Цвет	Нет	Цвет границы при выполнении условия	
Толщина границы	Целое число	Нет	Толщина границы при выполнении условия. Минимальное значение 0	2
Стиль границы	Выпадающий список	Нет	Стиль границы при выполнении условия (Сплошная или Пунктир)	Пунктир
Текст (подписи) – размер шрифта	Целое число	Нет	Текст (подписи) – размер шрифта при выполнении условия. Минимальное значение 1	
Текст (подписи) – цвет текста	Цвет	Нет	Текст (подписи) – цвет текста при выполнении условия	
Диапазон меньше (X < Значение)				
Цвет заливки	Цвет	Нет	Цвет заливки при выполнении условия	
Цвет границы	Цвет	Нет	Цвет границы при выполнении условия	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
Толщина границы	Целое число	Нет	Толщина границы при выполнении условия. Минимальное значение 0	5
Стиль границы	Выпадающий список	Нет	Стиль границы при выполнении условия (Сплошная или Пунктир)	Сплошная
Текст (подписи) – размер шрифта	Целое число	Нет	Текст (подписи) – размер шрифта при выполнении условия. Минимальное значение 1	
Текст (подписи) – цвет текста	Цвет	Нет	Текст (подписи) – цвет текста при выполнении условия	

6.8.3.3 Группа «Общие настройки блока»

Аналогично приведенным в п. 6.1.1.1.3.

6.8.3.4 Входы

Название входа	Тип	Обязательный	Описание
Ряды данных	Ряд данных	Да	Ряды данных, для которых нужно применить форматирование
Фильтр данных	Ряд данных	Нет	Фильтр для входных данных перед применением правил форматирования

6.8.3.5 Выходы

Название выхода	Тип	Описание
Ряды с форматированием	Ряд данных	Ряди данных с примененным форматированием

6.8.3.6 События блока

В блоке нет событий.

6.8.3.7 Описание работы блока

Блок предназначен для раскраски данных по заданным в настройках условиям. В данные, которые приходят в блок, добавляется специальный раздел с оформлением, который позволит визуализатору Таблица произвести раскраску значений. В настройках «Диапазоны» выбранные диапазоны значений не должны пересекаться. Для каждого диапазона задается свое оформление.

Вход «Фильтр данных» отвечает за дополнительные условия для форматирования данных. Форматирование будет происходить только для тех рядов, которые есть и во входных данных и в фильтре (**при совпадении** имени ряда).

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Например, данные для форматирования:

	ts_name	types	2000-01-01	2001-01-01	2002-01-01	2003-01-01	2004-01-01	2005-01-01	200
1	y	fact	1.00	2.00	3.00	75.00	36.00	72.00	

Рисунок 415 – Данные для форматирования

Заданные диапазоны для применения форматирования приведены на рисунке ниже (Рисунок 416).

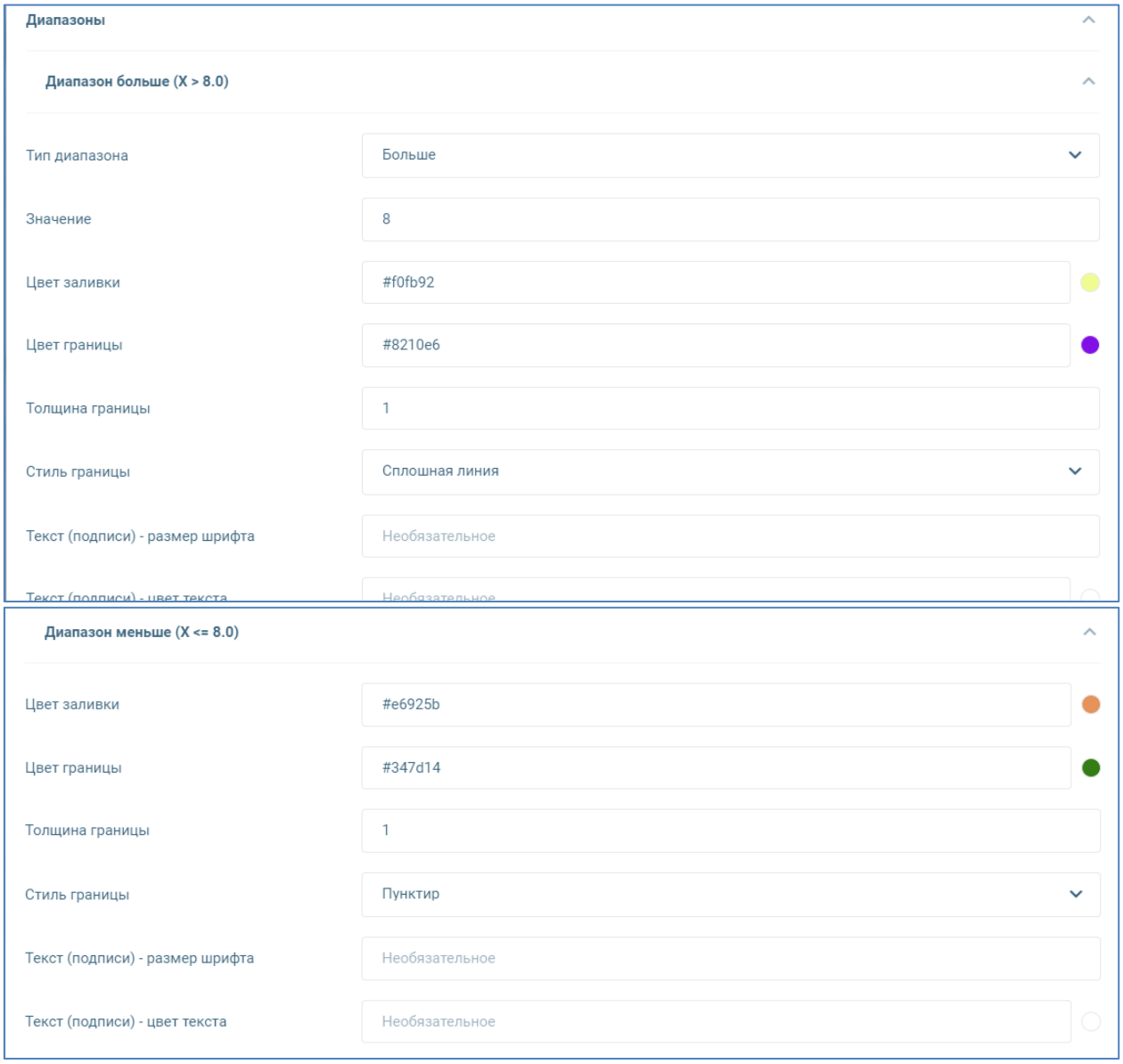


Рисунок 416 – Заданные диапазоны для форматирования

Результат после применения форматирования:

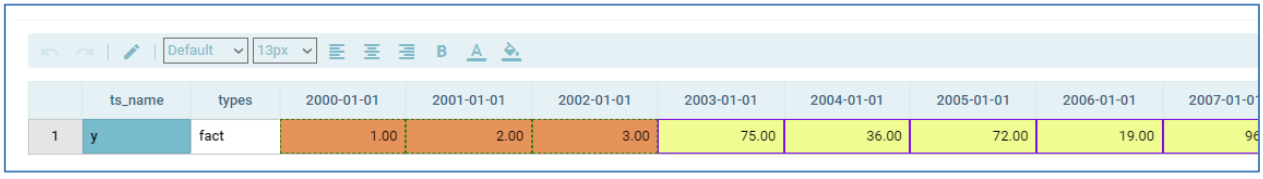


Рисунок 417 – Пример результата форматирования

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.9 Элементы машинного обучения

6.9.1 Блок «Python»

6.9.1.1 Назначение

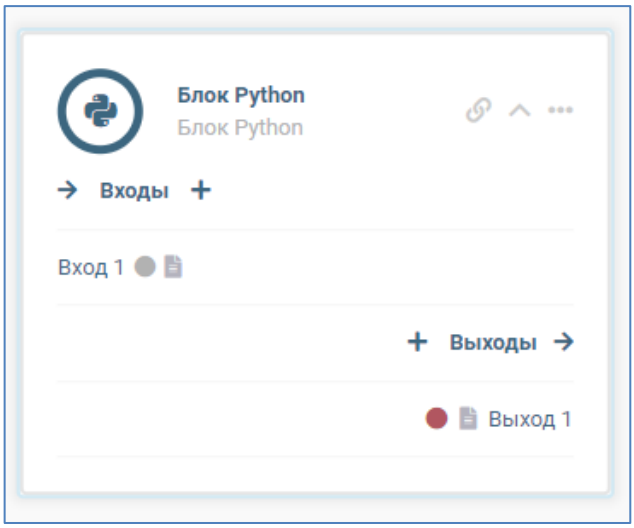


Рисунок 418 – Пример блока «Python»

Блок позволяет **небезопасно** выполнять произвольный Python код, написанный пользователем. Внутри этого кода можно гибко преобразовывать данные для последующих расчетов, а так же работать с БД и контекстом расчета графа (см. раздел Описание работы блока).

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.9.1.2 Настройки блока

Название параметра	Тип параметра	Обязательный	Описание	Пример
Выполнение кода	Выпадающий список	Нет	Место, где будет выполнятся код блока. По умолчанию: «Локально (на сервере)». Содержит следующие элементы: • Локально (на сервере) • На удаленном сервере. Подробное описание приведено в разделе «Песочница и удаленный запуск» документа «Руководство администратора»	

Код для выполнения	Строка (Python)	Нет	Код Python, который будет выполнен во время расчета блока	`# получение данных с 2 входов разными способами`
				`# изменение и передача на выход их объединения`
				`input_1 = execution_context.inputs[0].value`
				`input_2 = execution_context.get_input_by_id('in2').value`
				`input_2_id = execution_context.get_input_by_id('in2').id`
				
				`for elem in input_1:`
				`  elem['attrs']['indicator_variable'] = 'id выхода 2 = {0}'.format(input_2_id)`
				
				`execution_context.add_output('out1', input_1 + input_2)`

Мастер настроек	Флаг	Нет	Включает функцию прикладных настроек у блока.	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Обязательный	Описание	Пример
Описание настроек	JSON	Нет	Описание прикладных настроек у блока (см. п. 6.9.4)	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.9.1.3 Группа «Общие настройки блока»

Аналогично приведенным в п. 6.1.1.1.3.

6.9.1.4 Входы

Входы задаются пользователем при создании блока. Для создания входа необходимо нажать на знак + в разделе входов блока. Для каждого из входов задается следующий перечень полей:

Название входа	Тип	Обязательный	Описание
Идентификатор	Строка	Да	Идентификатор входа – используется внутри блока, для получения доступа к данным входа
Имя	Строка	Да	Имя входа для отображения в списке входов блока
Описание	Строка	Нет	Дополнительная информация с описанием входа
Тип	Выпадающий список	Да	Тип данных, которые будут приходить на вход. Нужен для построения корректных связей с другими блоками. По умолчанию установлен тип "Объект".
Обязательный	Флаг	Да	Обязательность входа. Если вход обязательный и на него не пришло данных – блок не будет рассчитан с ошибкой
Массив	Флаг	Да	Является ли вход массивом данных

6.9.1.5 Выходы

Выходы задаются пользователем при создании блока. Для создания выхода необходимо нажать на знак + в разделе выходов блока. Для каждого из выходов задается следующий перечень полей:

Название выхода	Тип	Описание	Название выхода
Идентификатор	Строка	Да	Идентификатор выхода – используется внутри блока, для формирования выхода
Имя	Строка	Да	Имя выхода для отображения в списке выходов блока
Описание	Строка	Нет	Дополнительная информация с описанием выхода

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название выхода	Тип	Описание	Название выхода
Тип	Выпадающий список	Да	Тип данных, которые будут приходить на выход. Нужен для построения корректных связей с другими блоками
Обязательный	Флаг	Да	Обязательность выхода. Эту настройку можно проверить во время выполнения блока
Массив	Флаг	Да	Является ли выход массивом данных. Служит так же для проверки внутри блока

6.9.1.6 Особенности работы с настройками блока

При работе с настройкам блока Python есть несколько отличий.

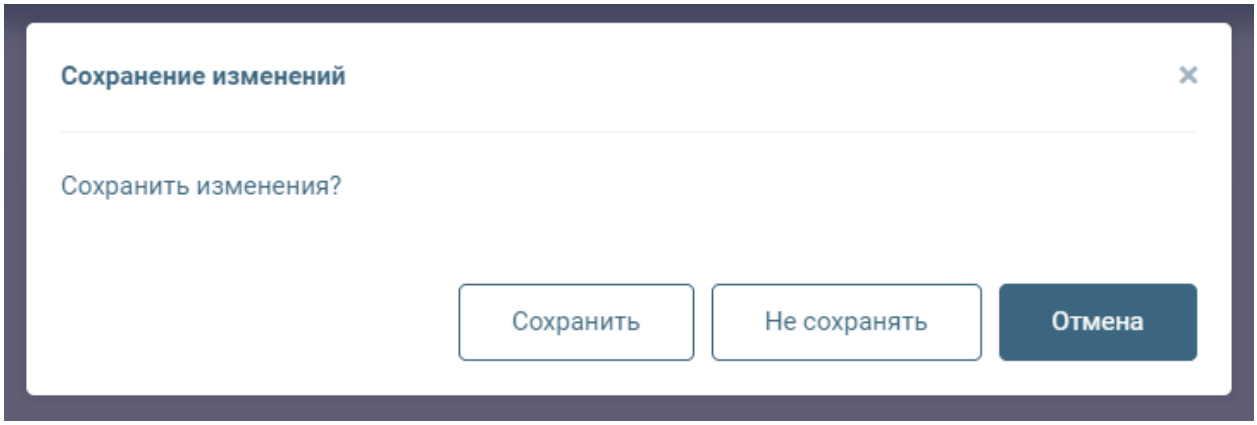


Рисунок 419 – Пример настроек блока «Python»

1. При попытке закрытия редактора кода выводится предупреждение, которое позволяет не потерять измененный код при закрытии окна.
2. В меню блока есть пункт для быстрого вызова редактора кода:

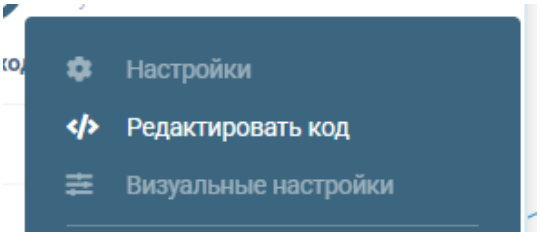
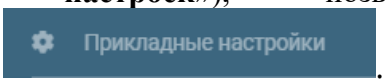


Рисунок 420 – Пример пункта «Редактировать код»

3. При двойном нажатии левой клавишей мыши на блок сразу открывается окно редактирования кода. Описание механизма работы приведено в п. 7.2.6 «Предпросмотр результатов настройки блока».
4. Для открытия настроек, как это происходит в других блоках 294px, необходимо вызвать пункт «Настройки» в меню блока.
5. Блок, не являясь визуализатором, поддерживает возможность предпросмотра своего результата выполнения

.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6. В меню блока доступен вызов прикладных настроек (если включен параметр «**Мастер настроек**»), позволяющий изменять прикладные настройки: 

.
7. В заголовке окна с кодом, развернутого на весь экран, отображается заголовок блока, который в данный момент редактирует пользователь.

6.9.1.7 События блока

В блоке нет событий.

6.9.1.8 Описание работы блока

Блок позволяет выполнять произвольный Python код, который пользователь загружает в него.

Если выбрано выполнение кода на:

- «Локально (на сервере)», то блок будет выполнятся **небезопасно**, потому что пользователю доступно как ядро системы, так и файловая система хоста приложения.
- «На удаленном сервере», то блок будет выполнятся **безопасно**, потому что пользователю недоступно как ядро системы, так и файловая система хоста приложения. На данный момент есть ограничения по использованию данного способа.

При использовании песочницы в коде **не обрабатывается** execution_context, кроме входов и выходов и прогресса расчета.

Пример того что обрабатывается:

- execution_context.get_input_by_id("id_input»).value
- execution_context.add_output("id_output", value)

Внутри блока пользователю доступен редактор Python с подсветкой синтаксиса и автодополнением ключевых конструкций для обращения к контексту выполнения.

Важно понимать как работает указание областей видимости **locals()** и **globals()**. Пользователю в коде доступно для использования только **locals()**, т.е. переменные и функции, только того контекста, который он сформировал сам. В коде не указывается ограничений на подключаемые библиотеки – единственное – они должны быть установлены на серверной части в момент запуска приложения.

Пользователь может подключить через import любую библиотеку из списка подключенных, а так же системные библиотеки. Потенциально – это может вызывать проблемы безопасности, так что блок Python **НЕЛЬЗЯ** делать доступным, если система находится в общем доступе.

Реализация механизма без песочницы, код будет выполнятся в потоке выполнения графа, вместе с основным.

6.9.1.8.1 Принципы выполнения кода и конструкция sleep()

Важная информация.

Важным моментом является то, что при выполнении кода Python блок забирает на себя процессорное время и если код выполняется вычислительно «тяжелый» (heavy CPU bound operation) – это можно привести к блокировке того процесса, в котором идет выполнение. По умолчанию, ядро системы выполняется на нескольких workers – эта настройка задается в Gunicorn сервере, при старте приложения (см. п. 11.5 «Настройки платформы» в

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

документе «Руководство администратора»). В случае тяжелых вычислений, запущенных несколько раз, может возникнуть ситуация, когда на обработку сообщений пользователей не останется процессорного времени, что приведет к таймаутам ответа сервера и неработоспособности приложения.

Для избегания таких проблем в коде Python блока можно использовать конструкцию time.sleep(x), которая позволит остановить выполнение кода и передать управление для обработки запросов пользователей к приложению. Использовать ее рекомендуется, когда есть циклы или есть возможность приостанавливать выполнение. В любом из этих случаев ответственность за код лежит на прикладном разработчике.

6.9.1.8.2 Принципы обработки ошибок в f-string выражениях

Важная информация.

В связи с особенностями обработки **f-string** в Python, необходимо учитывать, что при возникновении ошибки внутри такой строки, например синтаксической, Python выдаст позицию внутри этой строки. Из-за особенностей работы внутри **exec**, который выполняет пользовательский код нельзя определить в какой библиотеке произошла ошибка и понять что это **fstring**. Поэтому в случае ошибки будет видна только строка, но она может не совпадать с той, что в коде.

По возможности не используйте fstring конструкции внутри блока Python.

Пример ошибочного кода

В блоке код можно развернуть на весь экран с помощью кнопки .

В блоке нет возможности отладки или паузы выполнения. Для работы с этим функционалом см. раздел "Работа с внешней библиотекой локальной разработки"

```
in1 = execution_context.inputs[0].value
null = None

out = f"""function randomIntFromInterval(min, max) { // min and max included   # ошибка будет в этой строке, в отладке будет указано, что это строка 1, хотя в файле это строка 4
return Math.floor(Math.random() * (max - min + 1) + min)
}

const rndInt = randomIntFromInterval(100, 200)
var option = {
xAxis: {
type: 'category',
data: ['Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat', 'Sun']
},
yAxis: {
type: 'value'
},
series: [
{
data: [150, 230, 224, 218, 135, 147, rndInt],
type: 'line'
}
]
};

var dom = this.Element.nativeElement;
```


© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

```
dom.style.width = '100%';
dom.style.height = '100%';
var myChart = window['echarts'].init(dom);
myChart.on('rendered', (params) => {
this.IsLoaded = true;
});
var app = {};

if (option && typeof option === 'object') {
myChart.setOption(option);
}

this.resize = myChart.resize;

window.addEventListener('resize', myChart.resize);

myChart.on('click', (params) => {
this.OnGoToVisualizerEventHandler('fd347e74-6cb4-4e16-b29a-eb2128026308');
console.log(params);
if (params.componentType === 'series') {
const opts = myChart.getOption();
const series = opts.series;
const datasets = opts.datasets;
const formData = {
series: [],
index: []
};
const si = params.seriesIndex;
const di = params.dataIndex;
const sr = series[si];
const serie = { values: sr.data, options: sr.options };
formData.series.push(serie);
formData.index.push(di);
var data = {
id: this.Visualizer.ID,
data: formData
};
this.OnPointClickEventHandler(data);
}
});

// window.setTimeout(() => { myChart.resize; }, 300);'''
'''else:
out = {
"xAxis": {
"type": "category",
"data": ["Mon", "Tue", "Wed", "Thu", "Fri", "Sat", "Sun"]
},
"yAxis": {
"type": "value"
},
"series": [{
"data": [null],
"type": "line"
}]
```


© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

```
}"""

execution_context.output_connectors[0].value = outt
```

В блоке код можно развернуть на весь экран с помощью копки .

В блоке нет возможности отладки или паузы выполнения. Для работы с этим функционалом см. раздел «Работа с внешней библиотекой локальной разработки».

Пример кода:

```
input_1 = execution_context.inputs[0].value
input_2 = execution_context.get_input_by_id('in2').value
input_2_id = execution_context.get_input_by_id('in2').id

for elem in input_1:
    elem['attrs']['new_awsome_attr'] = 'id выхода 2 = {0}'.format(input_2_id)

execution_context.add_output('out1', input_1 + input_2)
```

Пример структуры входных данных для типа *Ряд данных.*

```
{
 "_params": {
  "fkey": [
   "date"
  ],
  "lazy": [],
  "ts": [
   {
    "frequency": "d",
    "key": "date",
    "mask": "%d.%m.%Y"
   }
  ],
  "vl": "vl"
 },
 "_source": "aaa479fd-db4e-42b3-a901-86fac8789edc",
 "attrs": {
  "arima": null,
  "model_type": "TREND",
  "trend": null
 },
 "keys": [
  "01.01.2020",
  "01.02.2020",
  "01.03.2020",
 ],
 "pkey": {
  "date_level": 7,
  "name": "X2"
 },
 "vl": [
  100,
  141.435621101643,
  159.417019107874,
 ]
}
```


© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.9.1.8.3 Основные концепты работы блока

- Импорты возможны только тех библиотек, которые подключены к расчетчику на этапе сборки или являются системными.
- Блок может работать с любыми типами входов или выходов.
- Работа со всеми типами входов происходит в их нативном формате через `execution_context.inputs[index].value`.
- Контроль корректности форматов не производится.
- При работе со входами, если передается массив – он будет доступен в коде как массив.
- При работе необходимо учитывать, что если один из входов не передан (пустой, или не рассчитан), то доступа к нему внутри кода не будет – расчетчик не передает его как вход.

6.9.1.8.4 Доступ ко входам и execution_context

Основным элементом для доступа к работе с входами/выходами и функциями ядра является переменная execution_context.

Для доступа к входам и выходам используются следующие конструкции:

```
Входы
 input_value                =                execution_context.inputs[0].value
 input_value          =          execution_context.get_input_by_id('id').value

Выходы
 execution_context.add_output('out1', 'some output value')
```

Выходы в коде всегда создаются динамически.

Для использования этой переменной можно вызывать автодополнение (CTRL+Пробел) с основными конструкциями.

6.9.1.8.5 Устаревшие варианты использования

Начиная с версии 2 конструкции ниже для описания входов не используются.

Входы:

Задаются динамически, через настройки. Пример:

```
[{
 "id": "input1",
 "type": "common",
 "mandatory": 1
},
{
 "id": "input2",
 "type": "common",
 "mandatory": 1
}]
```

Выходы:

Задаются динамически, через настройки. Пример:

```
[{
 "id": "outut1",
 "type": "common",
```


© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

```
 "mandatory": 1
}]
```

Следующие конструкции работают, но не рекомендуются к использованию:

```
execution_context.output_connectors[0].value = []
execution_context.output_connectors[0].value.set_value([])
inputs1 = execution_context.input_connectors[0].value
inputs2 = execution_context.input_connectors[1].value.get_value()
execution_context.output_connectors[0].value.set_value(execution_context.inpu
t_connectors[0].value.get_value())
execution_context.output_connectors[0].value.set_value(inputs1.get_value()+in
puts2.get_value())
```

!! В будущем поддержка этих конструкций будет прекращена.

6.9.1.8.6 Лог расчета и прогресс

В блоке есть возможность вывода пользовательских сообщений в лог и указания процента расчета блока.

6.9.1.8.6.1 Лог расчета

В лог выводятся сообщения пользователя в виде обычной строки, форматирование доступно обычными средствами Python.

Команды для работы с логом:

```
execution_context.log.info('Пример выдачи в лог записи с важностью info')

execution_context.log.warning('Пример выдачи в лог записи с важностью
warning – будет помечено желтым')


# записи с ошибкой могут быть для информации и могут останавливать расчет
принудительно

# для этого надо передать флаг stop

# при этом расчет будет остановлен на этой строке, в лог пойдет ошибка, весь
расчет будет помечен как ошибочный

execution_context.log.error('Пример error и блок не ошибочный, выполнение
продолжается')

execution_context.log.error('Пример error и блок не ошибочный, выполнение
продолжается', stop=True)
```

6.9.1.8.6.2 Прогресс расчета

Задавать прогресс расчета не обязательно, но если надо четко выделить шаги расчета по процентам можно воспользоваться возможностью задания процентов. При полном расчете графа прирост процентов расчета будет отображаться с учетом расчета блока. При расчете только блока будет произведен расчет корректного процента.

Проценты задаются с 0 до 100. При указании других значений будет взята ближайшая граница.

Команды для работы с прогрессом расчета блока

```
# инициализация прогресса, необязательна, но можно начать выполнение не с 0
процентов

execution_context.progress.initialize(0)
```


© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

```
# увеличение на N процентов
execution_context.progress.increase(10)
# установка конкретного процента
execution_context.progress.value = 30
# увеличение на 1%, до 100%
execution_context.progress.increase()
```

Так же через прогресс можно отслеживать прервал ли пользователь выполнение блока.

Приложение не может самостоятельно прервать расчет блока Python. Чтобы прервать расчет необходимо в коде указывать проверки на прерывание пользователем.

Для этого необходимо проверить в коде что есть прерывание. После этого можно корректно завершить выполнение своего кода, например, обработать транзакцию.

```
calc_stopped = execution_context.progress.is_stopped()
if calc_stopped:
    # выполняем вызов остановки и завершаем выполнение своего кода корректно
    # после того, как пользователь нажал на остановку – лога он не увидит, так что выводить сообщения тут бессмысленно
    # тут надо выполнить корректное завершение расчета, информация в лог попадает при вызове остановки
    # остановка вызовет ошибку расчета ErrorCodes.StoppedByUser и положит ее в лог пользователя и системы
    # эта ошибка так же вызовет остановку всего расчета, в т.ч. расчета графа – но блок свой расчет завершит
    execution_context.progress.do_stop()
```

6.9.1.8.7 Информация о контексте выполнения

В процессе выполнения блока возможно получение дополнительной информации о расчете и пользователе, который его запустил:

- Получение информации о пользователе (словарь)

```
user = execution_context.user_info
>> user: {
      "id": "105a9b07-fb45-4506-b519-fae19088df9b",
      "login": "kplotnikov",
      "fname": "Konstantin",
      "lname": "Plotnikov",
      "patronym": "",
      "email": "",
      "groups": ["Разработчики"]
  }
```

- Получение информации о задаче (словарь)

```
task = execution_context.task_info
>> task: {
      "id": "d533506a-8e81-4f67-8369-bf574096085b",
      "name": "VISMIND-6146 Блок Python. Текущий пользователь",
      "desc": "Новая задача от 05.10.2021 11:44:30",
      "created": "05.10.2021 11:44:30",
      "updated": "05.10.2021 11:47:11",
      "author": "Konstantin Plotnikov"
  }
```


© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

- Получение идентификатора расчета (строка, uuid)

```
calc_id = execution_context.calculation_id
>> calc_id: "26fb78c5-39b9-4bdf-b928-07a788a0cd8f"
```

- Получение информации о версии блока (целое число)

```
version = execution_context.block_version
>> version: 11
```

6.9.1.8.8 Работа с файловой системой

Этот функционал позволяет работать с файловой системой приложения.

Важно! Использовать **очень** аккуратно.

При необходимости генерации файла можно воспользоваться функционалом, который предоставляет библиотека docx.

Возможно формировать таким образом только docx файлы.

vmResource.create_file_name

Устарело. Для сохранения файлов необходимо использовать execution_context.save_bytes.

Для формирования файла используйте следующий пример (использует внутреннюю библиотеку vmResource).

```
import json
from calculator.system import vmResource
from docx import Document
from docx.shared import Inches

document = Document()
document.add_heading('Document Title', 0)
p = document.add_paragraph('A plain paragraph having some ')
p.add_run('bold').bold = True
p.add_run(' and some ')
p.add_run('italic.').italic = True

f = vmResource.create_file_name(name='test_file_6', ext='docx')
document.save(f)

result = json.dumps({'fileNames': [f]})

execution_context.add_output('input', result)
```

execution_context.save_bytes

```
execution_context.save_bytes(
    data: BinaryIO,
    name: str,
    ext: str,
    add_timestamp: bool = True,
    timezone: Optional[int] = None
) -> str
```

Метод позволяет сохранить файл в бинарном виде на файловый сервер. В ответе придет путь до объекта (внутри задачи).

Все файлы загружаются в папку /doc.

Параметры:

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

- **data** - данные файла.
- **name** - наименование файла, без расширения.
    - Если добавить в наименование файла разделитель "/" (обратный слеш), то можно указать папку, в которой нужно будет сохранить файл (корневой каталог /doc при этом останется), к примеру:
        - file- будет сохранен по пути /doc/file.ext
        - folder/file- будет сохранен по пути /doc/folder/file.ext
- **ext** - расширение файла.
- **add**_timestamp - флаг добавления временной отметки к имени файла.
    - Временная отметка будет отображена в формате --%Y-%m-%d--%H-%M-%S, к примеру someFileName--2024-02-01--13-29-29.docx
- **timezone** - позволяет настроить часовой пояс для временной отметки файла (если она включена). Целое число от -12 до 14. Если не указано для временной отметки будет использовано время сервера.

```
import io
import json
from docx import Document

# Создаем и заполняем docx документ
document = Document()
document.add_heading('Document Title', 0)
p = document.add_paragraph('A plain paragraph having some ')
p.add_run('bold').bold = True
p.add_run(' and some ')
p.add_run('italic.').italic = True

# Создаем экземпляр BytesIO
file_stream = io.BytesIO()
# Сохраняем документ в байты
document.save(file_stream)

# Сохраняем документ в хранилище
f = execution_context.save_bytes(
    data=file_stream,
    name='test_file_6',
    ext='docx',
    timezone=5
)
```

6.9.1.8.9 Работа с библиотекой vmResource

Для обращения к функциям ядра используется библиотека vmResource (см. п. 6.9.3).

6.9.1.8.10 Работа с внешней библиотекой локальной разработки

См. раздел Работа с внешней библиотекой локальной разработки vm-local-developer (см. п. 6.9.2).

6.9.1.8.11 Список пакетов, подключенных и доступных в блоке

Список пакетов приведен в приложении 1 «Внешние зависимости и библиотеки» в документе «Руководство администратора».

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.9.1.8.12 Удаленный запуск

Блок можно рассчитывать в удаленной, изолированной «песочнице» внутри docker контейнера. Подробное описание приведено в разделе «Песочница и удаленный запуск» в документе «Руководство администратора».

6.9.1.8.13 Пример блока

```
import time

val1 = execution_context.get_input_by_id('in1').value
execution_context.progress.initialize(0)

# info не помечает блок как ошибочный
execution_context.log.info('Пример info – Расчет блока запущен')
time.sleep(5)
execution_context.progress.increase(10)
execution_context.log.info('Пример info – Шаг 0, прогресс = 10')
time.sleep(5)
execution_context.progress.value = 30
execution_context.log.info('Пример info – Шаг 1, прогресс = 30')
time.sleep(5)
execution_context.progress.increase()
execution_context.log.info('Пример info – Шаг 2, прогресс + 1')

# warning не помечает блок как ошибочный
execution_context.log.warning('Пример warning – На входе {0}
рядов'.format(len(val1)))
time.sleep(5)
# таймаут на 20 секунд, пользователь может вызывать остановку расчета
execution_context.log.info('Пауза на 20 секунд – Можете вызывать остановку
расчета')
time.sleep(20)
execution_context.log.info('Проверяем был ли расчет блока остановлен')
calc_stopped = execution_context.progress.is_stopped()
if calc_stopped:
    # выполняем вызов остановки и завершаем выполнение своего кода корректно
    # после того, как пользователь нажал на остановку – лога он не увидит,
    # так что выводить сообщения тут бессмысленно
    # тут надо выполнить корректное завершение расчета, информация в лог
попадает при вызове остановки
    # остановка вызовет ошибку расчета ErrorCodes.StoppedByUser и положит ее в
лог
    execution_context.progress.do_stop()

execution_context.log.info('Остановки не было')
# выводим результат
res = val1
execution_context.add_output('out1', res)
execution_context.progress.value = 90
```

6.9.1.8.14 Работа с глобальными переменными

Важно!

Будьте осторожны с переопределением **глобальных переменных** библиотек (таких как например nympy.inf), т.к. это повлияет на весь проект.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Например с помощью библиотеки pandas можно изменить переменную **nympy.inf в nan.**

Пример, как делать нельзя:

```
import pandas as pd
import numpy as np

pd.set_option('use_inf_as_na', True)

df = pd.DataFrame([[1, 2], [2, np.nan], [3, np.inf]], columns=['col1', 'col2'])
df = df.dropna(subset=['col1', 'col2'], how='all')
```

Вместо него нужно сделать так:

Пример, который можно использовать:

```
import pandas as pd
import numpy as np

df = pd.DataFrame([[1, 2], [2, np.nan]], columns=['col1', 'col2'])
with pd.option_context('mode.use_inf_as_na', True):
    df = df.dropna(subset=['col1', 'col2'], how='all')
```

6.9.1.8.15 Работа с конструкциями управления

Важно!

Будьте осторожны с использованием конструкций управления выполнением, т.к. расчет блока происходит в контексте выполнения приложения (в т.ч. расчет событий).

Например, с помощью функции **exit()** будет завершено выполнение процесса целиком, после чего воркер, на котором происходило выполнение отключится.

6.9.1.9 Описание работы визуализатора

Блок не является визуализатором.

6.9.1.10 Настройки визуализатора

Блок не является визуализатором.

6.9.1.11 Дополнительно

Блок является одним из самых гибких инструментов для разработки, но необходимо помнить о том, что он не должен быть доступен всем пользователям из-за особенностей своей работы.

Описание пользовательских настроек (см. п. 6.9.4).

6.9.1.12 Дополнительные ссылки

Основы языка: https://www.w3schools.com/python/

StyleGuide: https://google.github.io/styleguide/pyguide.html

6.9.2 Локальная разработка. Пакет vm-local-developer

Версия пакета 0.1.2

Для разработки блоков Python можно использовать специальный пакет vm-local-developer, позволяющий локально работать с кодом блока Python, организуя рабочие пространства в стенды.

При работе код можно получить из блока Python в приложении и передать обратно. При этом доступна и локальная отладка кода.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Важно! vmResource

При локальной разработке с использованием vmResource доступны только БД: ClickHouse, MsSQL, Oracle, Postgres. Подробнее см. п. 6.9.3 «Модуль vmResource».

6.9.2.1 Установка Virtualenv

Virtualenv (виртуальная среда/виртуальное окружение) – это инструмент Python для управления зависимостями и изоляции проектов. Он позволяет устанавливать пакеты pip (сторонние библиотеки) локально в изолированном каталоге для конкретного проекта, а не глобально (этих пакетов не будет в других проектах Python).

Больше информации по виртуальным окружениям:

https://python-scripts.com/virtualenv

https://blog.sedicomm.com/2021/06/29/chto-takoe-virtualenv-v-python-i-kak-ego-ispolzovat/

Установка виртуального окружения в PyCharm

1. Создайте новый проект (File -> New Project...).
2. В настройках отметьте пункт **New environment using** (обычно отмечается по умолчанию):

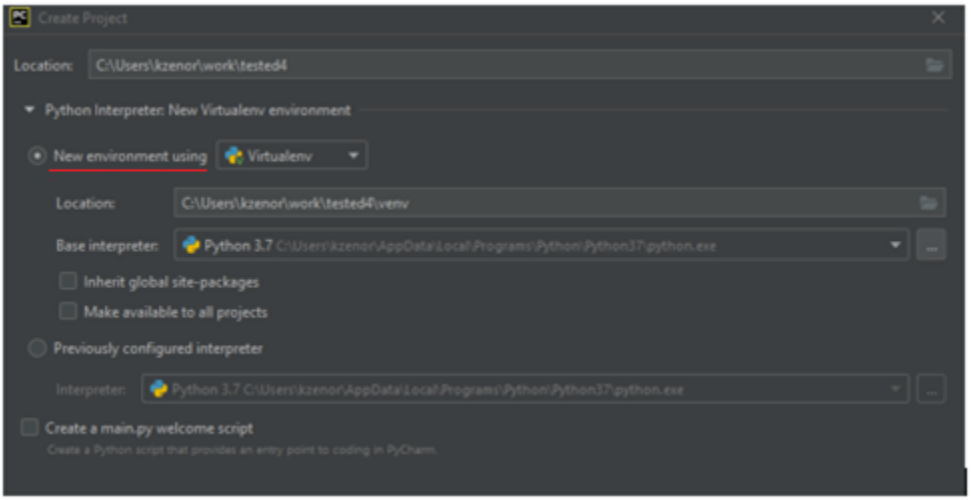


Рисунок 421 – Настройка флага «**New environment using**»

3. Создайте новый проект и дождитесь окончания его конфигурации.

6.9.2.1.1 Установка пакетов

1. Откройте терминал в PyCharm (в самом низу экрана):

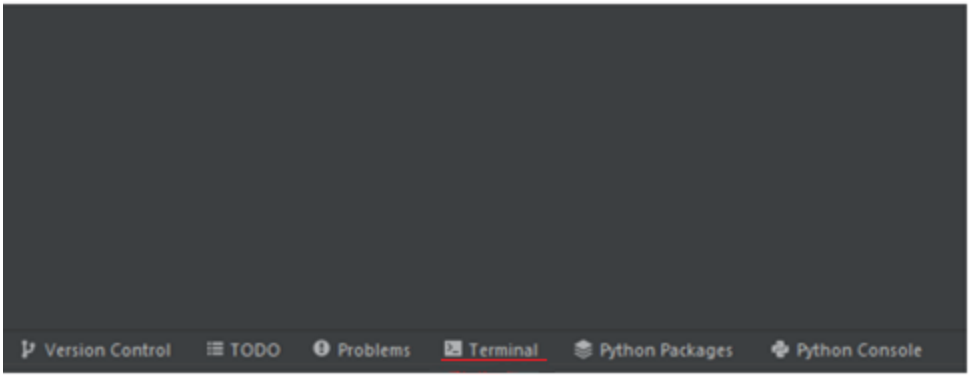


Рисунок 422 – Пример терминала в PyCharm

Важно! Проверьте, что открылась командная строка с виртуальным окружением:

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

```
(venv) C:\Users\kzenor\work\tested4>python -m vm_local_developer.vmld
usage: vmld.py [-h] -c CONFIG -p PATH [-n] [-s] [-e] [-u] [-d]
vmld.py: error: the following arguments are required: -c/--config, -p/--path
```

Рисунок 423 – Пример командной строки

2. В противном случае (актуально для новых версий) необходимо изменить настройки терминала.

   В терминале нажимаем значок ⌄.

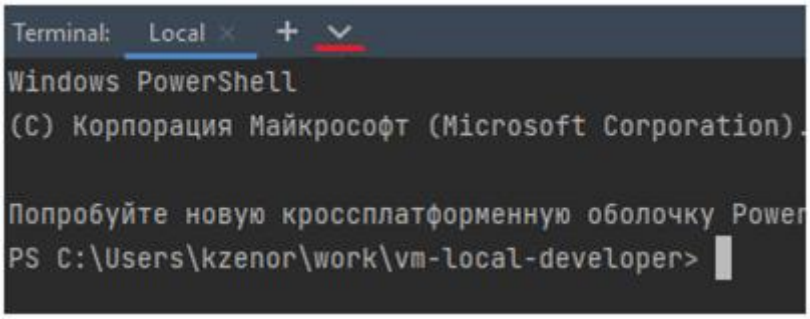


Рисунок 424 – Пример настройки терминала

3. Открываем настройки (**settings**) и в поле **Shell path** выбираем **...cmd.exe**:

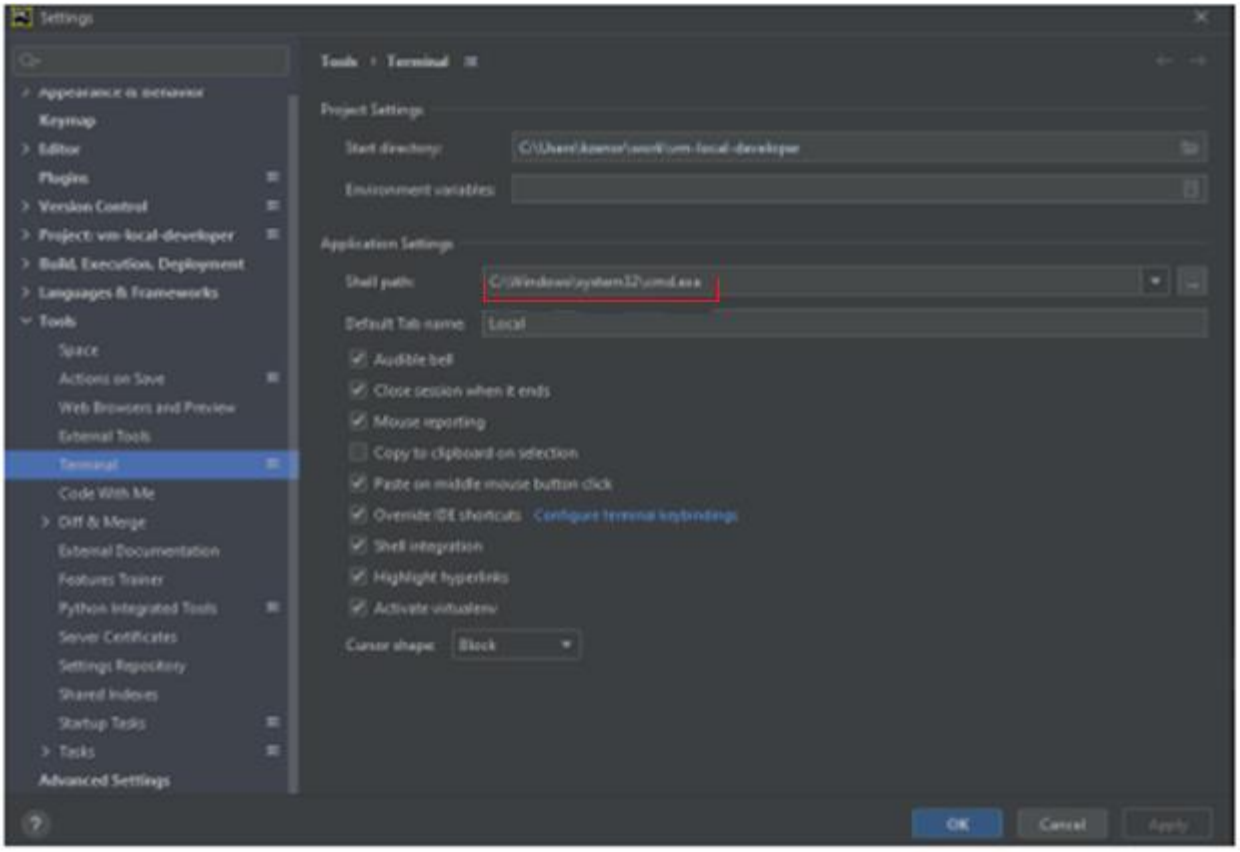


Рисунок 425 – Пример выбора файла

4. Нажимаем **Apply**, затем **OK** и переоткрываем терминал (закрываем все окна и открываем назад).

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

- После чего можно устанавливать все необходимые пакеты (либо через терминал, либо **File -> Settings -> Project -> Python Interpreter**; для отладчика Python блоков не используется, там только через терминал).

Рисунок 426 – Пример установки

- После установки всех пакетов можно запускать vm_local_developer для настройки стенда.

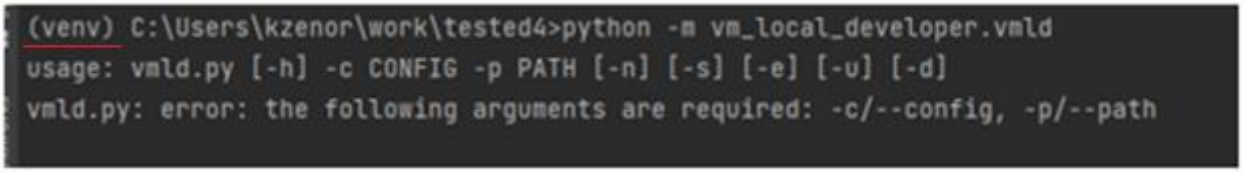


Рисунок 427 – Пример запуска vm_local_developer

6.9.2.2 Установка

Необходимо скачать файл .whl (`bit/vm-local-developer/-/tree/develop/dist`) и выполнить команду: `pip install <path_to_package>.whl`

После обновления версии модуля необходимо пересобрать стенд заново.

Важно! Не поддерживаются стенды, созданные до версии 0.1.0. После обновления с версии ниже 0.1.0 рекомендуется выполнить полную очистку метаданных: `vmld -clear`

6.9.2.2.1 Использование

В командной строке выполнить скрипт `vmld <params>`

Параметры скрипта:

- `-p <путь до папки, в которой будет создан (или уже находится) стенд разработки>`
    - Если параметр не указан, то в качестве пути будет использоваться каталог запуска скрипта, к примеру:

```
> C:\work\test vmld
--> Путь до файла: «C:\work\test"
> C:\work\test vmld -p D:\dist\other
--> Путь до файла: «D:\dist\other"
```

- `-c <путь или имя файла с конфигурацией>`
    - Можно указать полный путь до файла (`C:\work\test\block_config.json`), если файл с конфигурацией лежит отдельно от приложения.
    - Можно указать наименование файла (`block_config.json`), тогда в папке со стендом (см `-p`) будет произведен поиск файла по наименованию.
    - Если параметр не указан, будет произведен поиск файла по умолчанию (`config.json`) в папке со стендом.
- `-b <идентификатор блока>`

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

    - Идентификатор блока из задачи. Если параметр указан, то аналогичный параметр, если он есть в файле конфигурации, будет игнорироваться.
- `-t <токен пользователя>`
    - Токен авторизации пользователя. Если параметр указан, имя и логин, если они есть в файле конфигурации, будут игнорироваться.
- `-json <массив JSON без пробелов>`
    - Передача остальных параметров конфигурации, без указания отдельного файла.
    - По приоритету стоит выше файла конфигурации, но ниже параметров `-b` и `-t`
    - Формат передачи – массив JSON:
        - `{"task":"<task_guid>","user":"<user_id>"}`
    - Ограничение реализации: **запрещено указывать пробелы**, иначе параметры некорректно распознаются.
- `-clear` Полная очистка приложения: удаление всех стендов, сохраненных в метаданных, и очистка файла с метаданными. Требуется подтверждение операции.
- `-n` Флаг создания нового стенда, используется в качестве флага по умолчанию, если не заданы какие-либо другие флаги.
- `-u` Флаг обновления стенда (будет взята информация из блока в задаче). Требует подтверждения.
- `-d` Флаг удаления стенда и очистки метаданных. Требует подтверждения.
- `-s` Флаг сохранения стенда. Код из стенда будет сохранен в блок.
- `-e` Флаг запуска блока на расчет в приложении:
    - При вызове с флагом `-s` будет рассчитан новый код (вначале произойдет обновление блока кодом со стенда, а потом начнется расчет блока).
    - При одиночном вызове будет рассчитан текущий код в блоке.
- `-o` Флаг локального расчета стенда и сохранения в блок только выходов:
    - Код будет рассчитан на текущей рабочей машине, а обратно в блок будут сохранены только результаты расчета (выходы блока).

6.9.2.2.2 Стенд и метаданные

Стендам задается наименование по шаблону `python_block_<id блока>.py`, где `<id блока>` – идентификатор блока из приложения. При работе с уже созданным стендом поиск нужного стенда происходит по идентификатору блока. Если он не задан, а по указанному пути (или путь по умолчанию, если он не задан) находится единственный файл, то он будет использован в качестве указанного стенда. Если стендов несколько, то будет задан вопрос с просьбой указать необходимый стенд.

Метаданные сохраняются при создании нового стенда, а во всех остальных случаях подтягиваются автоматически, исходя из пути до стенда. (Это сделано для того, чтобы можно было создавать несколько стендов с одним и тем же блоком, но в разных папках). Исходя из этого, для корректной работы требуется не менять путь до файла и не изменять его название.

Примеры:

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

- Создание нового стенда:
    - С указанием пути до файла с конфигурацией:

      ```
      > C:\work\test vmld -c <C:\path\my_beautiful_config.json>
      ```

    - Без указания пути до файла с конфигурацией (будет осуществлен поиск файла config.json):

      ```
      > C:\work\test vmld -p <C:\path\>
      ```

- Запуск стенда на расчет с сохранением:

  ```
  > C:\work\test vmld -p <C:\path\> -b <block-uuid> -s -e
  ```

Файл конфигурации может быть любого расширения (.cfg .txt), но предпочтителен формат .json (т.к. внутри лежит JSON-массив).

Описание файла конфигурации:

```
{
  "block_id": <Идентификатор блока, строка>,
  "task_id": <Идентификатор задачи, строка>,
  "host": <Адрес приложения, строка, обязательное>,
  "port": <Порт приложения, целое число>,
  "user": <Логин пользователя, строка>,
  "password": <Пароль пользователя, строка>,
  "token": <Токен пользователя, строка>
}
```

6.9.2.2.3 Особенности реализации

- При каждой инициализации CDevTools выполняются запросы к приложению (получение данных/обновление блока/расчет блока).
- Реализован аналог класса CPythonBlockExecutionContext из backend, но с некоторыми изменениями:
    - В обычном приложении все параметры приходят во время расчета из калькулятора, здесь не выполняется расчет для получения данных (да и самого калькулятора нет). Вместо этого, все данные берутся запросами из других мест приложения (там где это возможно).
    - CProgressWrapper (методы `progress` и `log`) реализован на принтах, т.к. отдельно подключить прогрессбар не представляется возможным. Однако, все методы будут работать внутри приложения, во время расчета.
    - Данные по задачам и пользователю (`task_info` и `user_info`) извлекаются запросами на получение задачи и получение информации о пользователе соответственно.
    - Информация по входам и выходам блока извлекается запросом для отладки графа.
- Ограничение при работе с vmResource:
    - Каждое обращение к vmResource (к примеру `vmResource.get_sql_execute`) отправляет запрос на бекенд, там он выполняется и возвращается уже готовый ответ.
    - Для корректного определения блока соединения с БД параметр соединения, получаемый из входа блока, изменен на идентификатор блока соединения с БД.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

- На работу кода это никак не повлияло, но нельзя передавать этот параметр на выход блока, т.к. могут произойти сбои при расчете графа.

- Не работает метод `create_file_name` из-за специфики хранения файлов на сервере.

6.9.2.2.4 Для опытных пользователей

Можно использовать отдельно класс CDevTools (`from vm_local_developer import CDevTools`) для получения контекста. Параметры — все данные, которые передаются в файл конфигурации (user, password, и остальные).

В таком случае не будет поддержки на сохранение и расчет блока, но весь execution_context будет доступен.

```
from vm_local_developer import CDevTools

execution_context, vmResource = CDevTools(token=token,
 "user": username,
 "password": password,
    host='stage.int.bittechno.ru',
    port=80,
    task_id='f5776667-d26d-49a6-96a5-21551cd9d182',
    block_id='21ac1c56-c65c-461f-89b0-660cd3a09ed2'
).all_tools
```

Использовать можно либо связку пользователь-пароль, либо токен.

6.9.3 Модуль vmResource

Модуль позволяет работать с соединением с базой данных внутри блока Python.

6.9.3.1 Доступные методы

execute(id_connection, *args, **kwargs)

Позволяет выполнить запрос используя открытое соединение выбранной базы данных.

Параметры:

- id_connection (Строка) – id открытого соединения;
- sql_text (Строка) – запрос на получение данных;
- sql_params (Список или Справочник) – параметры запроса;
- дополнительные флаги:
    - dict_return – возвращение тела запроса в виде словаря (True|False);
    - только для MsSQL, Oracle, Postgres:
        - show_description – вывод описание полей (True|False);
        - one_row_return – вывод только одной строки запроса (True|False);
        - null_return_value – значение, возвращаемое при пустом запросе;
        - final_commit – коммит после завершения запроса (True|False).
            - Флаг **final_commit** отключен в версии 0.39. Выполнение транзакций с его помощью невозможно.

Поддерживает: ClickHouse, MsSQL, Oracle, Postgres.

Не поддерживает: Kafka, Mongo, SAP, SOAP.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Пример:

```
from calculator.system import vmResource

id_connect =  execution_context.inputs[0].value
sql = 'SELECT * FROM t_table WHERE name = %(name)s and model_name = %(model_name)s'
keys = {"name": "Пример", "model_name": "Модель"}

data = vmResource.execute(id_connect, sql, keys, final_commit=True
```

package_insert(id_connection, *args, **kwargs)

Позволяет выполнить пакетное сохранение данных, передаваемых в виде массива значений. Используется открытое соединение выбранной базы данных.

Параметры:

- id_connection (Строка) – id открытого соединения;
- sql_text (Строка) – запрос на вставку данных;
- sql_params (Список) – данные;
- дополнительные флаги:
    - final_commit – коммит после завершения запроса (True|False);
        - Флаг **final_commit** отключен в версии 0.39. Выполнение транзакций с его помощью невозможно
    - template – форма, по которым будут вставляться данные в БД.

Поддерживает: MsSQL, Oracle, Postgres.

Не поддерживает: ClickHouse, Kafka, Mongo, SAP, SOAP.

Пример:

```
from calculator.system import vmResource

id_connect =  execution_context.inputs[0].value

sql = """
    INSERT INTO t_table (
        column1, column2, vl
    )
    VALUES %s
"""

rows = [['Название 1', 'Атрибут 1', 10], ['Название 1', 'Атрибут 2', 100], ['Название 2', 'Атрибут 1', 1]]

vmResource.package_insert(id_connect,      sql_text=sql,      sql_params=rows, final_commit=True)
```

Пример вставки, с данными в виде списка:

```
from calculator.system import vmResource

id_connect =  execution_context.inputs[0].value

sql = """
    INSERT INTO t_table (
```


© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

```
        column1, column2, vl
    )
    VALUES %s
"""

# формат данных, по которым будут вставляться данные в бд
column_formats = '(%s::varchar, %s, %s::int)'

rows = [['Название 1', 'Атрибут 1', 10], ['Название 1', 'Атрибут 2', 100],
['Название 2', 'Атрибут 1', 1]]

vmResource.package_insert(id_connect,      sql_text=sql,      sql_params=rows,
template=column_formats, final_commit=True)
```

Пример вставки, с данными в виде справочника:

```
from calculator.system import vmResource

id_connect =  execution_context.inputs[0].value

sql = """
    INSERT INTO t_table (
        column1, column2, vl
    )
    VALUES %s
"""

# формат данных, по которым будут вставляться данные в бд
column_formats    =    "(%(CITY_ID)s,    NULLIF(%(TYPE_NAME)s,    ''):varchar,
%(TYPE_ID)s:int)"

rows = [{"CITY_ID": "Название 1", "TYPE_NAME": "Атрибут 1", "TYPE_ID": 10},
{"CITY_ID":  "Название  1",  "TYPE_NAME":  "Атрибут  2",  "TYPE_ID":  100},
{"CITY_ID": "Название 2", "TYPE_NAME": "Атрибут 1", "TYPE_ID": 1}]

vmResource.package_insert(id_connect,      sql_text=sql,      sql_params=rows,
template=column_formats, final_commit=True)
```

get_clickhouse(id_connection)

Получает класс для работы с ClickHouse.

Методы:

- execute(query, **kwargs) – выполнение запроса, где query – текст запроса.

Дополнительные параметры можно посмотреть в документации ClickHouse

Пример вызова метода:

```
from calculator.system import vmResource
conn_id = execution_context.inputs[0].value
click_house = vmResource.get_clickhouse(conn_id)

sql = 'select * from default.ora_vw_aff_channels'
res = click_house.execute(query=sql)

execution_context.log.info(res)
```


© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

- execute_iter(query, **kwargs) – создание итератора для получение данных, где query – текст запроса

Дополнительные параметры можно посмотреть в документации ClickHouse

Пример вызова метода:

```
from calculator.system import vmResource
conn_id = execution_context.inputs[0].value
click_house = vmResource.get_clickhouse(conn_id)

sql = 'select * from default.ora_vw_aff_channels'
settings = {'max_block_size': 100}
click_house.execute_iter(query=sql, settings=settings)

for row in click_house:
    execution_context.log.info(row)
```

- execute_with_progress(query, **kwargs) – запуск сложного запроса, с возможностью его остановить, где query – текст запроса.

Дополнительные параметры можно посмотреть в документации ClickHouse

После того как запрос выполнился надо вызвать get_result().

Пример вызова метода:

```
from calculator.system import vmResource
conn_id = execution_context.inputs[0].value
click_house = vmResource.get_clickhouse(conn_id)

sql = 'select * from default.ora_vw_aff_channels'
click_house.execute_with_progress(query=sql)
for num_rows, total_rows in click_house:
    execution_context.log.info(num_rows, total_rows)

execution_context.log.info(click_house.get_result())
```

- query_dataframe(query, **kwargs) – выполнение запроса, а результат в виде DataFrame, где query – текст запроса.

Дополнительные параметры можно посмотреть в документации ClickHouse

Условие для выполнения метода:

Работает если в блоке соединения с ClickHouse, в дополнительных параметрах установлен флаг use_numpy. Пример параметра; {"use_numpy": true}

Пример вызова метода:

```
import pandas as pd
import numpy as np
from calculator.system import vmResource

conn_id = execution_context.inputs[0].value
click_house = vmResource.get_clickhouse(conn_id)

sql = 'SELECT * FROM tutorial.test'
res = click_house.query_dataframe(query=sql)
```


© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

```
execution_context.log.info(str(res))
```

- insert_dataframe(query, dataframe, **kwargs) – вставка данных, где данные в виде DataFrame, где:
  - query – текст запроса;
  - dataframe – данные в формате DataFrame.

Дополнительные параметры можно посмотреть в документации ClickHouse

Условие для выполнения метода:

Работает если в блоке соединения с ClickHouse в дополнительных параметрах установлен флаг use_numpy. Пример параметра; {"use_numpy": true}.

Пример вызова метода:

```
import pandas as pd
import numpy as np
from calculator.system import vmResource

conn_id = execution_context.inputs[0].value
click_house = vmResource.get_clickhouse(conn_id)

df = pd.DataFrame({
    'a': [1, None, None],
    'b': [1.0, None, np.nan],
    'c': ['a', None, np.nan],
}, dtype=object)

sql = 'INSERT INTO tutorial.test VALUES'
click_house.insert_dataframe(sql, df)
```

Из-за специфики, данный метод не работает при локальной разработке (см. п. 6.9.2).

create_file_name(name, ext)

Формирует уникальное имя файла (переданное имя файла + текущее время) и возвращает полный путь до этого файла на сервере.

Параметры:

- **name** (Строка) – наименование файла;
- **ext** (Строка) – расширение файла;

Из-за специфики формирования пути на сервере, данный метод не работает при локальной разработке.

Пример:

```
from calculator.system import vmResource
f = vmResource.create_file_name(name='test_file_6', ext='docx')
```

6.9.3.2 Обработки исключений

В связи с тем, что на бэке происходит обработка ошибок, то через обычный except Exception не получится получить текст ошибки, т.к. он будет считан в CException.

Для получения текста ошибки нужно воспользоваться классом CException и получить атрибут message.

Пример обработки ошибки:

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

```
from calculator.system import vmResource
from core.errors import CException

id_connect =  execution_context.inputs[0].value
sql = 'SELECT * FROM t_table WHERE name = %(name)s and model_name =
%(model_name)s'
keys = {"name": "Пример", "model_name": "Модель"}

try:
    data = vmResource.execute(id_connect, sql, keys, final_commit=True)
except CException as e:
    execution_context.log.error(e.message)
```

6.9.4 Описание пользовательских настроек

Пользовательские настройки представляют собой массив с описанием настроек, которые будут отображены при вызове соответствующего мастера в блоке «Python» (см. п. 6.9.1).

Пример настроек (блок «Диаграмма»):

```
[
 {
  "name": "Заголовок блока",
  "id": "title",
  "type": "string",
  "desc": "",
  "hint": "Возможно формирование заголовка с помощью шаблона Jinja",
  "mandatory": 0,
  "default": "Диаграмма"
 },
 {
  "name": "Тип диаграммы",
  "id": "type",
  "type": "enum",
  "desc": "",
  "mandatory": 0,
  "default": 1,
  "multiselect": 0,
  "enum": [
   {
    "val": 1,
    "name": "Линейная",
    "desc": "Линейная диаграмма"
   },
   {
    "val": 2,
    "name": "Столбчатая",
    "desc": "Столбчатая диаграмма"
   },
   {
    "val": 3,
    "name": "С областями",
    "desc": "Диаграмма с областями"
   },
   {
    "val": 4,
```


© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

```
    "name": "Смешанная",
    "desc": "Смешанная линейная или столбчатая диаграмма"
   },
   {
    "val": 5,
    "name": "Круговая",
    "desc": "Круговая диаграмма"
   },
   {
    "val": 6,
    "name": "Радар",
    "desc": "Диаграмма радар или паутинная"
   },
   {
    "val": 7,
    "name": "Горизонтальная",
    "desc": "Горизонтальная столбчатая диаграмма"
   },
   {
    "val": 8,
    "name": "Кольцевая",
    "desc": "Кольцевая диаграмма"
   },
   {
    "val": 9,
    "name": "Каскадная",
    "desc": "Каскадная диаграмма"
   }
  ]
 },
 {
  "name": "Накопление",
  "id": "stacked",
  "type": "bool",
  "desc": "",
  "mandatory": 0,
  "default": 0
 },
 {
  "name": "Сортировка",
  "id": "sort",
  "type": "enum",
  "desc": "",
  "mandatory": 0,
  "default": 3,
  "multiselect": 0,
  "enum": [
   {
    "val": 1,
    "name": "Без сортировки",
    "desc": ""
   },
   {
    "val": 2,
    "name": "По дате/алфавиту (убывание)",
```


© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

```
    "desc": ""
   },
   {
    "val": 3,
    "name": "По дате/алфавиту (возрастание)",
    "desc": ""
   },
   {
    "val": 4,
    "name": "По значению (убывание)",
    "desc": "Смешанная линейная или столбчатая диаграмма"
   },
   {
    "val": 5,
    "name": "По значению (возрастание)",
    "desc": ""
   }
  ]
 },
 {
  "name": "Количество знаков после запятой",
  "id": "round",
  "type": "int",
  "desc": "Настройка количества знаков после запятой",
  "mandatory": 0,
  "default": 2,
  "min": 0,
  "max": 15
 },
 {
  "name": "Легенда",
  "id": "legend",
  "type": "json",
  "desc": "Параметры настройки легенды диаграммы",
  "mandatory": 0,
  "elements": [
   {
    "name": "Отображать легенду",
    "id": "display",
    "type": "bool",
    "desc": "",
    "mandatory": 0,
    "default": 1
   },
   {
    "name": "Размещение легенды",
    "id": "position",
    "type": "enum",
    "desc": "",
    "mandatory": 0,
    "default": 3,
    "multiselect": 0,
    "enum": [
     {
      "val": 1,
```


© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

```
      "name": "Вверху",
      "desc": ""
     },
     {
      "val": 2,
      "name": "Слева",
      "desc": ""
     },
     {
      "val": 3,
      "name": "Внизу",
      "desc": ""
     },
     {
      "val": 4,
      "name": "Справа",
      "desc": ""
     }
    ]
   },
   {
    "name": "Размер шрифта",
    "id": "legendFontSize",
    "type": "int",
    "desc": "",
    "mandatory": 0,
    "default": 10
   }
  ]
 },
 {
  "name": "Оси данных",
  "id": "axes",
  "type": "json",
  "desc": "Параметры настройки осей диаграммы",
  "mandatory": 0,
  "elements": [
   {
    "name": "Маска оси времени",
    "id": "dtMask",
    "type": "string",
    "desc": "Маска для формирования подписей на временной оси",
    "mandatory": 0,
    "default": "%d.%m.%Y",
    "taskSetting": "parameters.mask"
   }
  ]
 },
 {
  "name": "Общие настройки стилей рядов",
  "id": "common_style",
  "type": "json",
  "desc": "Общие настройки стилей рядов диаграммы",
  "mandatory": 0,
  "elements": [
```


© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

```
{
 "name": "Вид линии",
 "id": "LineView",
 "type": "enum",
 "desc": "",
 "mandatory": 0,
 "default": "line",
 "multiselect": 0,
 "enum": [
  {
   "val": "line",
   "name": "Прямая",
   "desc": "Линейная диаграмма"
  },
  {
   "val": "spline",
   "name": "Сглаженная",
   "desc": "Столбчатая диаграмма"
  }
 ]
},
{
 "name": "Вид раскраски",
 "id": "paint_types",
 "type": "enum",
 "desc": "",
 "mandatory": 0,
 "multiselect": 0,
 "default": 0,
 "enum": [
  {
   "val": 0,
   "name": "Отсутствует",
   "desc": ""
  },
  {
   "val": 1,
   "name": "Цветовой градиент",
   "desc": ""
  },
  {
   "val": 2,
   "name": "Градиент прозрачности",
   "desc": ""
  }
 ]
},
{
 "name": "Начальный цвет",
 "id": "paint_color_start",
 "type": "color",
 "desc": "",
 "mandatory": 0,
 "default": "#001DFF"
},
```


© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

```
{
 "name": "Конечный цвет",
 "id": "paint_color_end",
 "type": "color",
 "desc": "",
 "mandatory": 0,
 "default": "#00FF00"
},
{
 "name": "Цвет",
 "id": "paint_color",
 "type": "color",
 "desc": "",
 "mandatory": 0,
 "default": "#001DFF"
},
{
 "name": "Каскадная диаграмма - начальный цвет",
 "id": "waterfallStartColor",
 "type": "color",
 "desc": "",
 "mandatory": 0,
 "default": "#c3c3c3"
},
{
 "name": "Каскадная диаграмма - убывающий цвет",
 "id": "waterfallDownColor",
 "type": "color",
 "desc": "",
 "mandatory": 0,
 "default": "#c76879"
},
{
 "name": "Каскадная диаграмма - растущий цвет",
 "id": "waterfallUpColor",
 "type": "color",
 "desc": "",
 "mandatory": 0,
 "default": "#89c69c"
},
{
 "name": "Выводить линию",
 "id": "showLine",
 "type": "bool",
 "desc": "Выводить линию для линейной диаграммы",
 "mandatory": 0,
 "default": 1
},
{
 "name": "Тип линии или границы",
 "id": "linetype",
 "type": "enum",
 "desc": "",
 "mandatory": 0,
 "multiselect": 0,
```


© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

```
 "default": "solid",
 "enum": [
  {
   "val": "solid",
   "name": "Сплошная линия",
   "desc": ""
  },
  {
   "val": "dashed",
   "name": "Штриховая линия",
   "desc": ""
  },
  {
   "val": "dotted",
   "name": "Линия из точек",
   "desc": ""
  },
  {
   "val": "dot-dash",
   "name": "Штрихпунктирная линия",
   "desc": ""
  },
  {
   "val": "dot-dot-dash",
   "name": "Штрихпунктирная линия с двумя точками",
   "desc": ""
  }
 ]
},
{
 "name": "Толщина линии, пт",
 "id": "thickness",
 "type": "int",
 "desc": "",
 "mandatory": 0,
 "default": 1,
 "min": 1,
 "max": 10
},
{
 "name": "Тип маркера",
 "id": "pointstype",
 "type": "enum",
 "desc": "",
 "mandatory": 0,
 "multiselect": 0,
 "default": "circle",
 "enum": [
  {
   "val": "circle",
   "name": "Круг",
   "desc": ""
  },
  {
   "val": "triangle",
```


© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

```
    "name": "Треугольник",
    "desc": ""
   },
   {
    "val": "rect",
    "name": "Квадрат",
    "desc": ""
   },
   {
    "val": "rectRounded",
    "name": "Квадрат с закругленными углами",
    "desc": ""
   },
   {
    "val": "rectRot",
    "name": "Ромб",
    "desc": ""
   },
   {
    "val": "cross",
    "name": "Крестик",
    "desc": ""
   },
   {
    "val": "crossRot",
    "name": "Перекрестие",
    "desc": ""
   },
   {
    "val": "star",
    "name": "Звездочка",
    "desc": ""
   },
   {
    "val": "line",
    "name": "Линия",
    "desc": ""
   },
   {
    "val": "dash",
    "name": "Черта",
    "desc": ""
   }
  ]
 },
 {
  "name": "Размер маркера",
  "id": "pointsize",
  "type": "int",
  "desc": "",
  "mandatory": 0,
  "default": 1
 },
 {
  "name": "Показывать подписи данных",
```


© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

```
 "id": "showValues",
 "type": "bool",
 "desc": "",
 "mandatory": 0,
 "default": 0
},
{
 "name": "Подписи данных - положение",
 "id": "captionplace",
 "type": "enum",
 "desc": "",
 "mandatory": 0,
 "multiselect": 0,
 "default": "center",
 "enum": [
  {
   "val": "center",
   "name": "В центре",
   "desc": ""
  },
  {
   "val": "left",
   "name": "Слева",
   "desc": ""
  },
  {
   "val": "upper",
   "name": "Сверху",
   "desc": ""
  },
  {
   "val": "right",
   "name": "Справа",
   "desc": ""
  },
  {
   "val": "bottom",
   "name": "Снизу",
   "desc": ""
  }
 ]
},
{
 "name": "Подписи данных - размер шрифта",
 "id": "captionsfontsize",
 "type": "int",
 "desc": "",
 "mandatory": 0,
 "default": 10
},
{
 "name": "Подписи данных - цвет шрифта",
 "id": "captionsfontcolor",
 "type": "color",
 "desc": "",
```


© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

```
    "mandatory": 0,
    "default": "#000000"
   }
  ]
 },
 {
  "name": "Цвета точек для круговой диаграммы",
  "id": "pointsColor",
  "type": "repeater",
  "desc": "Цвета точек для круговой и кольцевой диаграммы",
  "mandatory": 0,
  "repeating": [
   {
    "name": "Название точки",
    "id": "name",
    "type": "string",
    "desc": "",
    "mandatory": 0,
    "default": "Точка {NUM}"
   },
   {
    "name": "Цвет",
    "id": "color",
    "type": "color",
    "desc": "Цвет точки",
    "mandatory": 0,
    "default": "#A5B592"
   }
  ]
 },
 {
  "name": "Ряды данных",
  "id": "rows",
  "type": "repeater",
  "desc": "Параметры настройки отображения рядов данных",
  "mandatory": 0,
  "repeating": [
   {
    "name": "Название ряда",
    "id": "name",
    "type": "string",
    "desc": "",
    "mandatory": 0,
    "default": "Ряд {NUM}",
    "hint": "Для использования названия ряда из привязок оставить значение
данного атрибута пустым"
   },
   {
    "name": "Отображать ряд",
    "id": "showRow",
    "type": "bool",
    "desc": "Отображать ряд",
    "mandatory": 0,
    "default": 1,
    "hint": "Возможность скрыть или показать ряд на диаграмме"
```


© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

```
},
{
 "name": "Тип ряда",
 "id": "type",
 "type": "enum",
 "desc": "Тип ряда задается только для смешанных диаграмм",
 "mandatory": 0,
 "default": "default",
 "enum": [
  {
   "val": "default",
   "name": "По умолчанию",
   "desc": ""
  },
  {
   "val": "line",
   "name": "Линия",
   "desc": ""
  },
  {
   "val": "bar",
   "name": "Столбец",
   "desc": ""
  }
 ]
},
{
 "name": "Отображение выделения",
 "id": "showClickBar",
 "type": "bool",
 "desc": "Отображение выделения",
 "mandatory": 0,
 "default": 0,
 "hint": "Выделение столбца, при клике на нем"
},
{
 "name": "Горизонтальное смещение",
 "id": "horizontalPlaced",
 "type": "int",
 "desc": "Горизонтальное смещение",
 "mandatory": 0,
 "default": 0,
 "hint": "Отображение столбцов со смещением"
},
{
 "name": "Вид линии",
 "id": "LineView",
 "type": "enum",
 "desc": "",
 "mandatory": 0,
 "default": "default",
 "multiselect": 0,
 "enum": [
  {
   "val": "default",
```


© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

```
    "name": "По умолчанию",
    "desc": ""
   },
   {
    "val": "line",
    "name": "Прямая",
    "desc": "Линейная диаграмма"
   },
   {
    "val": "spline",
    "name": "Сглаженная",
    "desc": "Столбчатая диаграмма"
   }
  ]
},
{
 "name": "Цвет",
 "id": "color",
 "type": "color",
 "desc": "Цвет заливки ряда",
 "mandatory": 0,
 "default": "#A5B592"
},
{
 "name": "На дополнительную ось",
 "id": "additionalAxes",
 "type": "bool",
 "desc": "Отобразить значения ряда на дополнительной оси",
 "mandatory": 0,
 "default": 0
},
{
 "name": "Выводить линию",
 "id": "showLine",
 "type": "bool",
 "desc": "Выводить линию для линейной диаграммы",
 "mandatory": 0,
 "default": 1
},
{
 "name": "Тип линии или границы",
 "id": "linetype",
 "type": "enum",
 "desc": "",
 "mandatory": 0,
 "multiselect": 0,
 "default": "default",
 "enum": [
  {
   "val": "default",
   "name": "По умолчанию",
   "desc": ""
  },
  {
   "val": "solid",
```


© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

```
    "name": "Сплошная линия",
    "desc": ""
   },
   {
    "val": "dashed",
    "name": "Штриховая линия",
    "desc": ""
   },
   {
    "val": "dotted",
    "name": "Линия из точек",
    "desc": ""
   },
   {
    "val": "dot-dash",
    "name": "Штрихпунктирная линия",
    "desc": ""
   },
   {
    "val": "dot-dot-dash",
    "name": "Штрихпунктирная линия с двумя точками",
    "desc": ""
   }
  ]
 },
 {
  "name": "Толщина линии, пт",
  "id": "thickness",
  "type": "int",
  "desc": "",
  "mandatory": 0,
  "default": 1,
  "min": 1,
  "max": 10
 },
 {
  "name": "Тип маркера",
  "id": "pointstype",
  "type": "enum",
  "desc": "",
  "mandatory": 0,
  "multiselect": 0,
  "default": "default",
  "enum": [
   {
    "val": "default",
    "name": "По умолчанию",
    "desc": ""
   },
   {
    "val": "circle",
    "name": "Круг",
    "desc": ""
   },
   {
```


© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

```
    "val": "triangle",
    "name": "Треугольник",
    "desc": ""
   },
   {
    "val": "rect",
    "name": "Квадрат",
    "desc": ""
   },
   {
    "val": "rectRounded",
    "name": "Квадрат с закругленными углами",
    "desc": ""
   },
   {
    "val": "rectRot",
    "name": "Ромб",
    "desc": ""
   },
   {
    "val": "cross",
    "name": "Крестик",
    "desc": ""
   },
   {
    "val": "crossRot",
    "name": "Перекрестие",
    "desc": ""
   },
   {
    "val": "star",
    "name": "Звездочка",
    "desc": ""
   },
   {
    "val": "line",
    "name": "Линия",
    "desc": ""
   },
   {
    "val": "dash",
    "name": "Черта",
    "desc": ""
   }
  ]
 },
 {
  "name": "Размер маркера",
  "id": "pointsize",
  "type": "int",
  "desc": "",
  "mandatory": 0,
  "default": 1
 },
 {
```


© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

```
 "name": "Показывать подписи данных",
 "id": "showValues",
 "type": "bool",
 "desc": "",
 "mandatory": 0,
 "default": 0
},
{
 "name": "Подписи данных - положение",
 "id": "captionplace",
 "type": "enum",
 "desc": "",
 "mandatory": 0,
 "multiselect": 0,
 "default": "default",
 "enum": [
  {
   "val": "default",
   "name": "По умолчанию",
   "desc": ""
  },
  {
   "val": "center",
   "name": "В центре",
   "desc": ""
  },
  {
   "val": "left",
   "name": "Слева",
   "desc": ""
  },
  {
   "val": "upper",
   "name": "Сверху",
   "desc": ""
  },
  {
   "val": "right",
   "name": "Справа",
   "desc": ""
  },
  {
   "val": "bottom",
   "name": "Снизу",
   "desc": ""
  }
 ]
},
{
 "name": "Подписи данных - размер шрифта",
 "id": "captionsfontsize",
 "type": "int",
 "desc": "",
 "mandatory": 0,
 "default": 10
```


© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

```
  },
  {
   "name": "Подписи данных - цвет шрифта",
   "id": "captionsfontcolor",
   "type": "color",
   "desc": "",
   "mandatory": 0,
   "default": "#000000"
  }
 ]
},
{
 "name": "Коридоры",
 "id": "corridors",
 "type": "repeater",
 "desc": "Параметры настройки отображения коридоров для данных",
 "mandatory": 0,
 "repeating": [
  {
   "name": "Название коридора",
   "id": "name",
   "type": "string",
   "desc": "",
   "mandatory": 0,
   "default": "Коридор {NUM}"
  },
  {
   "name": "Индекс ряда 1",
   "id": "row_first",
   "type": "int",
   "desc": "Индекс ряда (начиная с 1)",
   "mandatory": 0,
   "default": 1
  },
  {
   "name": "Индекс ряда 2",
   "id": "row_second",
   "type": "int",
   "desc": "Индекс ряда (начиная с 1)",
   "mandatory": 0,
   "default": 2
  },
  {
   "name": "Цвет",
   "id": "color",
   "type": "color",
   "desc": "Цвет заливки коридора",
   "mandatory": 0,
   "default": "#db1d59"
  }
 ]
},
{
 "name": "Привязки",
 "id": "binding",
```


© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

```
 "type": "json_str",
 "desc": "Привязки атрибутов объектов",
 "mandatory": 0
},
{
 "name": "Ось X",
 "id": "xAxes",
 "type": "json",
 "desc": "Параметры настройки оси X",
 "mandatory": 0,
 "elements": [
  {
   "name": "Сетка",
   "id": "grid",
   "type": "json",
   "desc": "",
   "mandatory": 0,
   "elements": [
    {
     "name": "Отображать линии",
     "id": "showGrid",
     "type": "bool",
     "desc": "",
     "mandatory": 0,
     "default": 1
    },
    {
     "name": "Цвет линии",
     "id": "gridLineColor",
     "type": "color",
     "desc": "",
     "mandatory": 0,
     "default": "#000000"
    },
    {
     "name": "Толщина линии",
     "id": "gridLineWidth",
     "desc": "",
     "type": "float",
     "mandatory": 0,
     "default": 1.0
    },
    {
     "name": "Тип штриха",
     "id": "gridLineBorderDash",
     "type": "enum",
     "desc": "",
     "mandatory": 0,
     "default": "solid",
     "multiselect": 0,
     "enum": [
      {
       "val": "solid",
       "name": "Сплошная линия",
       "desc": ""
```


© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

```
      },
      {
       "val": "dashed",
       "name": "Штриховая линия",
       "desc": ""
      },
      {
       "val": "dotted",
       "name": "Линия из точек",
       "desc": ""
      },
      {
       "val": "dot-dash",
       "name": "Штрихпунктирная линия",
       "desc": ""
      },
      {
       "val": "dot-dot-dash",
       "name": "Штрихпунктирная линия с двумя точками",
       "desc": ""
      }
     ]
    }
   ]
},
{
 "name": "Деления",
 "id": "ticks",
 "type": "json",
 "desc": "",
 "mandatory": 0,
 "elements": [
  {
   "name": "Количество (0 - не учитывать)",
   "id": "maxTicksLimit",
   "type": "int",
   "desc": "",
   "mandatory": 0,
   "default": 0
  },
  {
   "name": "Пропускать подписи оси",
   "id": "autoSkip",
   "type": "bool",
   "desc": "",
   "mandatory": 0,
   "default": 0
  },
  {
   "name": "Отступ",
   "id": "autoSkipPadding",
   "type": "int",
   "desc": "",
   "mandatory": 0,
   "default": 0
```


© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

```
  }
 ]
},
{
 "name": "Размер шрифта",
 "id": "axesFontSize",
 "type": "int",
 "desc": "",
 "mandatory": 0,
 "default": 10
},
{
 "name": "Начертание",
 "id": "axesFontStyle",
 "type": "enum",
 "desc": "",
 "mandatory": 0,
 "multiselect": 0,
 "default": "normal",
 "enum": [
  {
   "val": "normal",
   "name": "Обычный",
   "desc": ""
  },
  {
   "val": "italic",
   "name": "Курсив",
   "desc": ""
  },
  {
   "val": "bold",
   "name": "Полужирный",
   "desc": ""
  }
 ]
},
{
 "name": "Перенос по словам",
 "id": "axesWordWrap",
 "type": "bool",
 "desc": "",
 "mandatory": 0,
 "default": 0
},
{
 "name": "Поворот надписи",
 "id": "axesRotation",
 "type": "enum",
 "desc": "",
 "mandatory": 0,
 "multiselect": 0,
 "default": "auto",
 "enum": [
  {
```


© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

```
      "val": "auto",
      "name": "Автоматически (по умолчанию)",
      "desc": ""
     },
     {
      "val": "horizontal",
      "name": "Отображать горизонтально",
      "desc": ""
     }
    ]
   }
  ]
 },
 {
  "name": "Ось Y",
  "id": "yAxes",
  "type": "json",
  "desc": "Параметры настройки оси Y",
  "mandatory": 0,
  "elements": [
   {
    "name": "Сетка",
    "id": "grid",
    "type": "json",
    "desc": "",
    "mandatory": 0,
    "elements": [
     {
      "name": "Отображать линии",
      "id": "showGrid",
      "type": "bool",
      "desc": "",
      "mandatory": 0,
      "default": 1
     },
     {
      "name": "Цвет линии",
      "id": "gridLineColor",
      "type": "color",
      "desc": "",
      "mandatory": 0,
      "default": "#000000"
     },
     {
      "name": "Толщина линии",
      "id": "gridLineWidth",
      "desc": "",
      "type": "float",
      "mandatory": 0,
      "default": 1.0
     },
     {
      "name": "Тип штриха",
      "id": "gridLineBorderDash",
      "type": "enum",
```


© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

```
    "desc": "",
    "mandatory": 0,
    "default": "solid",
    "multiselect": 0,
    "enum": [
     {
      "val": "solid",
      "name": "Сплошная линия",
      "desc": ""
     },
     {
      "val": "dashed",
      "name": "Штриховая линия",
      "desc": ""
     },
     {
      "val": "dotted",
      "name": "Линия из точек",
      "desc": ""
     },
     {
      "val": "dot-dash",
      "name": "Штрихпунктирная линия",
      "desc": ""
     },
     {
      "val": "dot-dot-dash",
      "name": "Штрихпунктирная линия с двумя точками",
      "desc": ""
     }
    ]
   }
  ]
 },
 {
  "name": "Деления",
  "id": "ticks",
  "type": "json",
  "desc": "",
  "mandatory": 0,
  "elements": [
   {
    "name": "Количество (0 - не учитывать)",
    "id": "maxTicksLimit",
    "type": "int",
    "desc": "",
    "mandatory": 0,
    "default": 0
   },
   {
    "name": "Пропускать подписи оси",
    "id": "autoSkip",
    "type": "bool",
    "desc": "",
    "mandatory": 0,
```


© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

```
      "default": 0
     },
     {
      "name": "Отступ",
      "id": "autoSkipPadding",
      "type": "int",
      "desc": "",
      "mandatory": 0,
      "default": 0
     }
    ]
   },
   {
    "name": "Размер шрифта",
    "id": "axesFontSize",
    "type": "int",
    "desc": "",
    "mandatory": 0,
    "default": 10
   }
  ]
 },
 {
  "name": "Масштабирование данных",
  "id": "dataScaling",
  "type": "bool",
  "desc": "",
  "mandatory": 0,
  "default": 0
 },
 {
  "name": "Количество отображаемых точек",
  "id": "axisLength",
  "type": "int",
  "desc":  "Количество отображаемых точек при включенном масштабировании
данных",
  "mandatory": 0,
  "default": 200
 }
]
```


6.9.4.1 Обязательные поля настроек

- **name** – наименование поля;
- **id** – идентификатор поля;
- **type** – тип поля, подробнее см. п. 6.9.4.2;
- **desc** – описание поля (не используется);
- **hint** – текстовая подсказка, отображаемая слева от наименования ( ). Поддерживает коды HTML;
- **mandatory** – флаг обязательности поля;
- **default** – значение поля по умолчанию.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.9.4.2 Доступные типы полей

Наименование	Идентификатор	Описание
Строка	string	Поле для ввода текстовых значений
Шифрованная строка	string_encoded	Поле для ввода зашифрованной строк
Текст	text	Поле для ввода многострочных текстовых значений
JSON	json_str	Поле ввода кода JSON
HTML	html	Поле ввода кода HTML
SQL	sql	Поле ввода кода SQL
JavaScript	javascript	Поле ввода кода JavaScript
Python	python	Поле ввода кода Python
Выражение	vismind	Поле ввода кода для выражений.
Целое число	int	Поле ввода целого числа. Доступны дополнительные параметры: • **min** – минимальное значение; • **max** – максимальное значение
Вещественное число	float	Поле ввода вещественного числа. Доступны дополнительные параметры: • **min** – минимальное значение; • **max** – максимальное значение
Флаг (Логический тип)	bool	Флаг с переключением
Дата	date	Поле ввода даты
Дата и время	datetime	Поле ввода даты и времени
Цвет	color	Поле ввода (или выбора) цвета

Линейный список	list	Выбор элементов из линейного списка.
		Рекомендуется использовать тип «**Перечисление**».
		Обязательные параметры:
		• **list** – массив с описанием элементов:
		  • **id** – идентификатор элемента;
		  • **ord** – порядок расположения элемента:
		• **name** – наименование элемента.

Иерархический список	tree	Выбор элементов из иерархического списка.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Наименование	Идентификатор	Описание
		Поле отключено в приложении

Иерархический список с загрузкой данных	lazy_tree	Выбор элементов из иерархического списка.
		Список элементов извлекается при помощи запроса к API приложения.
		Обязательные параметры:
		• **api** – адрес запроса.
		Дополнительные параметры:
		• **apiMethod** – метод API. Доступные значения: «**GET**», «**POST**». По умолчанию (если поле не указано) используется метод GET;
		• **apiParameters** – массив параметров, переданных в параметре запроса:
		  • **id** – идентификатор параметра;
		  • **value** – значение параметров.
		  • Для использования параметров, задающихся в настройках, эти параметры задаются в фигурных скобках **{}**.
		  • Для вызова параметров, расположенных в корневом элементе, требуется просто указать идентификатор параметра: **{param_id}**
		  • Для вызова параметров, расположенных в структуре (см. ниже), требуется указать перечень параметров, начиная с корневого, через точку(**.**): **{folder.folder2.param_id}**
		  • Технические параметры:
		  • **{TASK_ID}** – идентификатор задачи;
		  • **{LIB_ID}** – идентификатор типа блока.
		• **apiBody** – массив параметров, переданных в теле запроса. Настройки аналогичны **apiParameters**

Кнопка с вызовом API	button	Кнопка для выполнения метода API. Обязательные параметры: • **api** – адрес запроса

Перечисление	enum	Выбор элементов из линейного списка.
		Обязательные параметры:
		• **enum** – массив с описанием элементов:
		  • **val** – идентификатор элемента;
		  • **name** – наименование элемента:

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Наименование	Идентификатор	Описание
		• **desc** – описание элемента (не используется)
Переключатель	radiobutton	Переключатель между несколькими вариантами. Обязательные параметры: • **elements** – массив с описанием элементов: • **val** – идентификатор элемента; • **name** – наименование элемента
Структура	json	Структура, содержащая в себе другие настройки. Обязательные параметры: • **elements** – массив с описанием настроек. Параметр **default** не применяется для структуры. В отличии от остальных настроек, при получении структуры через блок «Python», в качестве ответа вернется словарь вида **{<идентификатор настройки>: <значение настройки>}**
Структура – массив	repeater	Структура, содержащая в себе массив с другими настройками. Обязательные параметры: • **repeating –** массив с описанием настроек. Параметр **default** не применяется для структуры. В структуре доступен технический параметр **{NUM}**, возвращающий номер элемента в массиве (начиная с 1). Если в структуре есть элемент с ключом **name**, то он будет отображаться в качестве наименования элемента структуры. В отличии от остальных настроек, при получении структуры через блок «Python», в качестве ответа вернется массив вида **[{<идентификатор настройки>: <значение настройки>},...]**
Контейнер с блоками	buttonContainer	Структура, содержащая в себе два массива с настройками, между которыми можно переключаться. Обязательные параметры: • **enabled** – массив с описанием настроек; • **disabled** – массив с описанием настроек. Дополнительные параметры: • **additionControl** – словарь с описанием настройки, которая располагается рядом с переключателем настроек. Поддерживается только настройка типа «**Переключатель**»

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Наименование	Идентификатор	Описание
		Параметр **default** может принимать значения: 0 – отображена вкладка **disabled;** 1 – отображена вкладка **enabled.** При получении настройки через блок «Python», в качестве ответа вернется словарь с тремя параметрами: • **val** – флаг с указанием того, какая вкладка активна (0/1); • **enabled** – описание настроек вкладки (аналогично «Структуре»); • **disabled** – описание настроек вкладки (аналогично «Структуре»)

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.10 Формы

6.10.1 Блок «Кнопка»

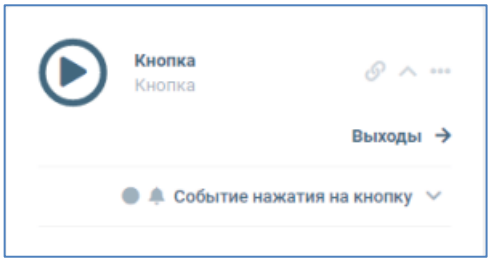


Рисунок 428 – Пример блока «Кнопка»

6.10.1.1 Назначение

Блок позволяет отобразить кнопку на графе.

Визуализатор блока является **Компонентом**

6.10.1.2 Настройки блока

Название параметра	Тип параметра	Обязательный	Описание	Пример
Наименование блока	Строка	Нет	Наименование блока	
Текст на кнопке	Строка	Да	Текст, написанный на кнопке	
Иконка кнопки	Строка	Нет	Ссылка на файл с инонкой для кнопки	
Данные при нажатии кнопки	JSON	Нет	JSON с описанием параметров, которые будут переданы в событийном выходе «Успешность». При пустом значении в выход будет передано значение True.	

6.10.1.3 Группа «Общие настройки блока»

Аналогично приведенным в п. 6.1.1.1.3.

6.10.1.4 Входы

В блоке нет входов.

6.10.1.5 Выходы

В блоке нет выходов.

6.10.1.6 События блока

Название выхода	Тип	Описание
Событие нажатия на кнопку		
Успешность	JSON	Флаг, обозначающий то, что кнопка была нажата. Передает данные, указаные в параметре «**Данные при нажатии кнопки**»

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.10.1.7 Описание работы блока

Блок формирует визуализатор в виде кнопки с текстом, указанным в настройке «**Текст на кнопке**».

6.10.1.8 Описание работы визуализатора

Визуализатор представляет собой кнопку с текстом:

.

После нажатия на кнопку происходит вызов события и передача данных.

6.10.1.9 Настройки визуализатора

Общий список приведен в п. 4.2.1.3 «Общие настройки в режиме редактирования пресета». У визуализатора отсутствует группа настроек «**Заголовок**», т.к. визуализатор является формой ввода.

Помимо стандартных, визуализатор позволяет задать следующие настройки:

Название параметра	Тип параметра	Обязательность	Описание
Размер шрифта текста кнопок	Целое число	Нет	Задает размер текста в кнопке
Цвет текста кнопок	Цвет	Нет	Задает цвет текста в кнопке
Цвет фона кнопок	Цвет	Нет	Задает цвет фона в кнопке
Цвет текста активной кнопки	Цвет	Нет	Задает цвет текста активной (нажатой) кнопки
Цвет фона активной кнопки	Цвет	Нет	Задает цвет фона активной (нажатой) кнопки
Скругление углов кнопок	Целое число	Нет	Задает скругление углов кнопки

6.10.2 Блок «Строка ввода»

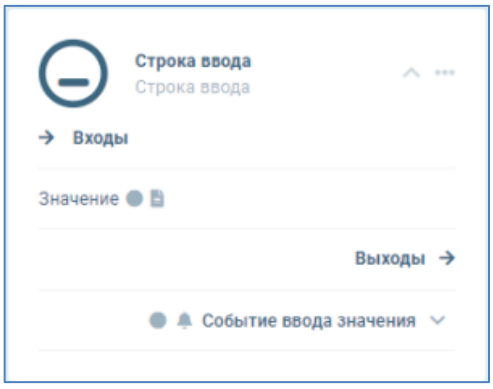


Рисунок 429 – Пример блока «Строка ввода»

6.10.2.1 Назначение

Блок позволяет формировать визуализатор для ввода значений из приложения.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.10.2.2 Настройки блока

Название параметра	Тип параметра	Обязательный	Описание	Пример
Тип значения	Выпадающий список	Да	Тип возможных форматов заполнения блока: • Строка • Текст • Электронная почта • Дата • Время • Телефон • Целое число • Дробное число • Цвет	
Пример заполнения	Строка	Нет	Пример заполнения пока поле не будет заполнено	
Текст заголовка	Строка	Нет	Текст заголовка у строки ввода	Текст заголовка
Обязательное поле	Флаг	Нет	При включении – визуализатор будет подсвечиваться красным, пока поле не будет заполнено	
Запускать событие по нажатию на клавишу Enter	Флаг	Нет		Запускать событие по нажатию на клавишу Enter. Не работает если "Тип значения": Текст, т.к. в этом случае на клавишу Enter происходит перенос каретки.

6.10.2.3 Группа «Общие настройки блока»

Аналогично приведенным в п. 6.1.1.1.3.

6.10.2.4 Входы

Название входа	Тип	Обязательный	Описание
Значение	Строка	Нет	Значение, которое будет использоваться в качестве подсказки

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.10.2.5 Выходы

В блоке нет выходов

6.10.2.6 События блока

Название выхода	Тип	Описание
Событие ввода значения		
Введенное значение	Строка	Введеное пользователем значение в строковом формате.

6.10.2.7 Описание работы блока

Блок считывает все настройки, а так же значение, поступившее на вход, после чего собирает визуализатор.

Никаких проверок на корректность данных для настройки «Пример» и для входа не производится.

6.10.2.8 Описание работы визуализатора

Визуализатор представляет собой поле для ввода с наименованием, подсказкой (знак вопроса справа от названия), если она есть, а так же с примером заполнения (если эта настройка задана).

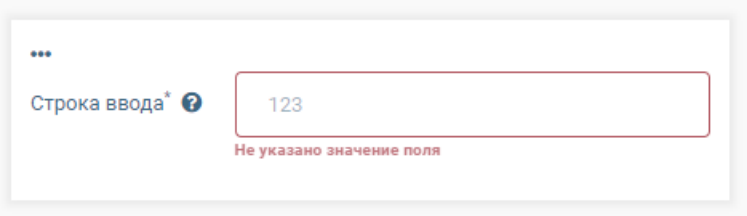


Рисунок 430 – Пример блока «Строка ввода»

В случае включения флага «Обязательное поле», строка ввода без значения будет подсвечена красным.

6.10.2.9 Настройки визуализатора

Общий список приведен в п. 4.2.1.3 «Общие настройки в режиме редактирования пресета». У визуализатора отсутствует группа настроек «Заголовок», т.к. визуализатор является формой ввода.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

6.10.3 Блок «Вопрос»

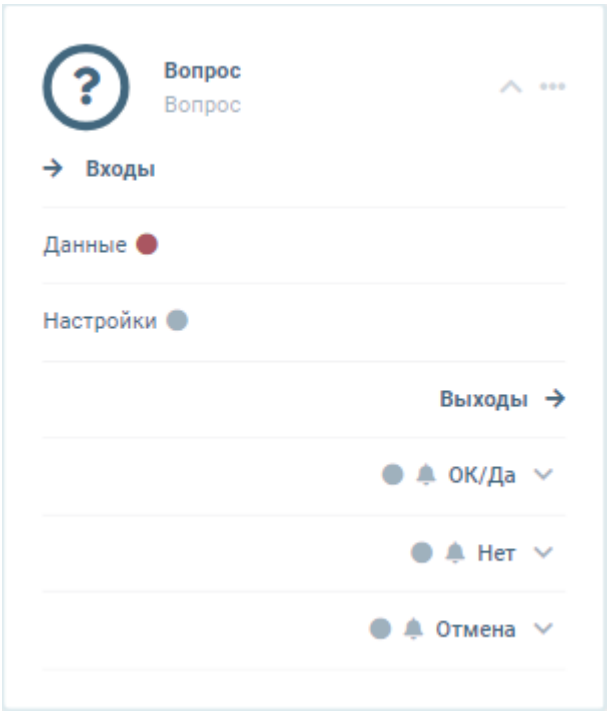


Рисунок 431 – Пример блока «Вопрос»

6.10.3.1 Назначение

Блок позволяет формировать интерактивный визуализатор для ответа на введенный вопрос.

6.10.3.2 Настройки блока

Название параметра	Тип параметра	Обязательный	Описание	Пример
Текст сообщения	Строка	Да	Отображаемый текст вопроса.	
Отображаемые кнопки	Выпадающий список	Да	Кнопки, доступные для отображения. Возможные варианты: • ОК • ОК/Отмена • Да/Нет • Да/Нет/Отмена	

6.10.3.3 Группа «Общие настройки блока»

Аналогично приведенным в п. 6.1.1.1.3.

6.10.3.4 Входы

Название входа	Тип	Обязательный	Описание
Данные	JSON	Да	Данные, которые передадутся дальше по графу, при нажатии кнопки ОК/Да
Настройки	JSON	Да	Настройки визуализатора в формате JSON. Имеют приоритет над настройками блока.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название входа	Тип	Обязательный	Описание
			Возможные поля настроек: • text - отвечает за текст сообщения. Переопределяет настройку "Текст сообщения". • buttons - отображаемые кнопки. Переопределяет настройку "Отображаемые кнопки". Доступные настройки кнопок: Список кнопок жестко определен и может включать один из следующих параметров: • ok - ОК • okCancel - ОК/Отмена • yesNo - Да/Нет • yesNoCancel - Да/Нет/Отмена

6.10.3.5 Выходы

В блоке нет выходов.

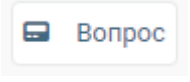
6.10.3.6 События блока

Название выхода	Тип	Описание
ОК/Да		
Данные	JSON	Данные, которые будут переданы при нажатии на кнопку "ОК" или "Да"
Нет		
Данные	JSON	**Важно!** В версии 0.39 из этого события никаких данных не выходит, только пустой массив. Данные, которые будут переданы при нажатии на кнопку "Нет".
Отмена		
Данные	JSON	**Важно!** В версии 0.39 из этого события никаких данных не выходит, только пустой массив. Данные, которые будут переданы при нажатии на кнопку "Отмена".

6.10.3.7 Описание работы блока

Блок считывает настройки и собирает визуализатор с нужным количеством кнопок.

6.10.3.8 Описание работы визуализатора

При размещении визуализатора в режиме редактирования будет отображен технический визуализатор: 

.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

При отображении пресета в режиме просмотра, визуализатор не будет отображаться до тех пор, пока не произойдет его обновление (к примеру, после события). В таком случае, будет отображено всплывающее окно с введенным текстом и выбранными кнопками:

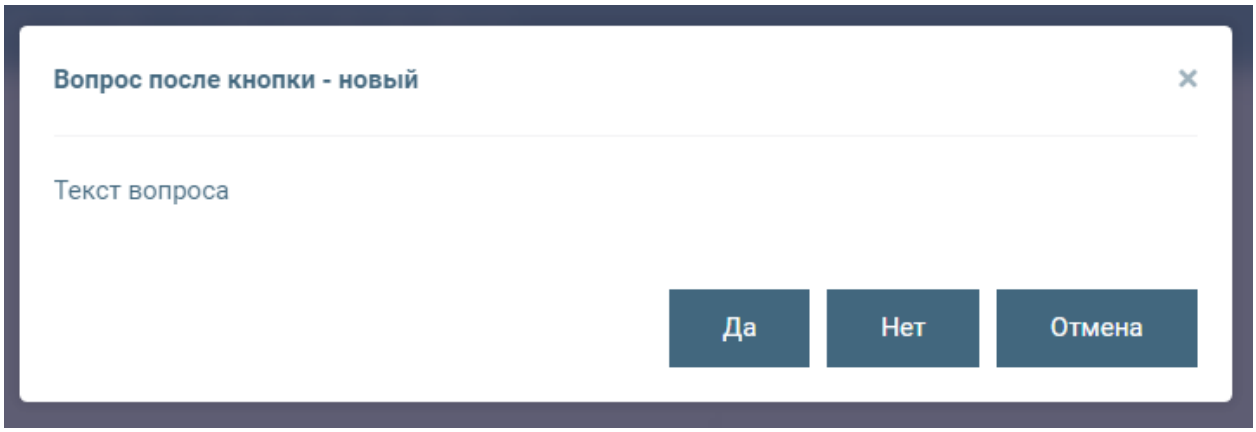


Рисунок 432 – Отображение диалогового окна с вопросом пользователю

После нажатия на одну из кнопок, визуализатор закроется и будет выполнена ветка, стоящая за выходом, относящимся к нажатой кнопке.

Закрытие окна (крестик в правом верхнем углу) приравнено к нажатию кнопки "Отмена"

Визуализатор без события

Если визуализатор стоит без события (т.е. рассчитывается вместе с основным графом), то всплывающее окно с вопросом будет отображено сразу при открытии пресета

6.10.3.9 Настройки визуализатора

Общий список приведен в п. 4.2.1.3 «Общие настройки в режиме редактирования пресета».

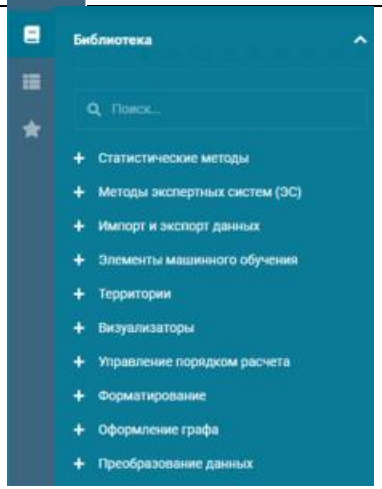
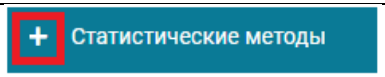
© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

7 Граф

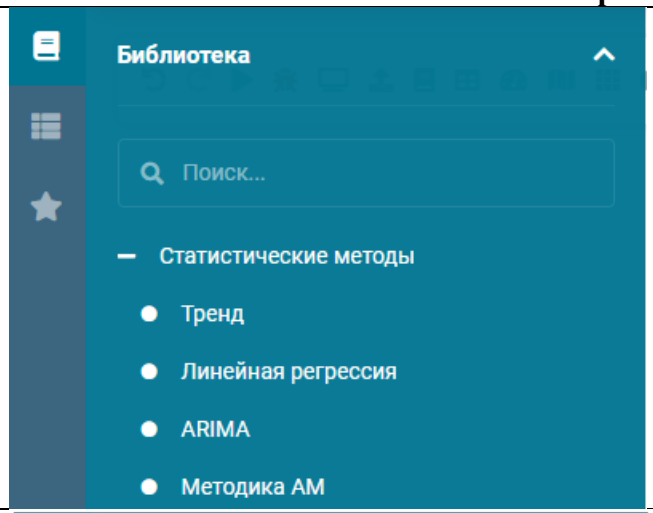
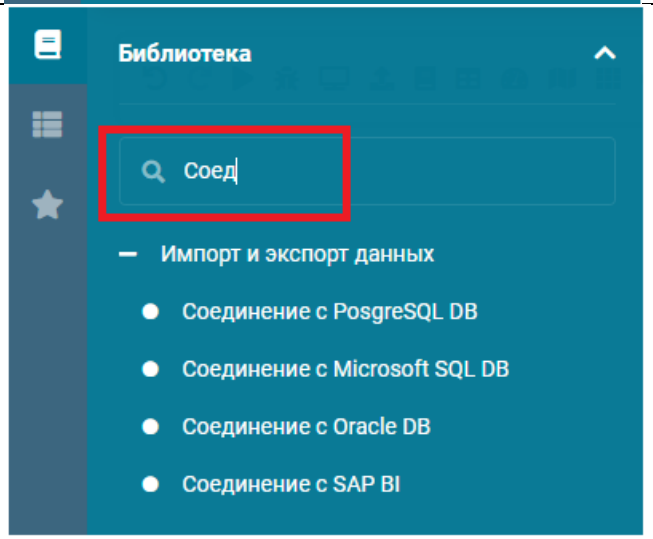
7.1 Руководство по работе с графом

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

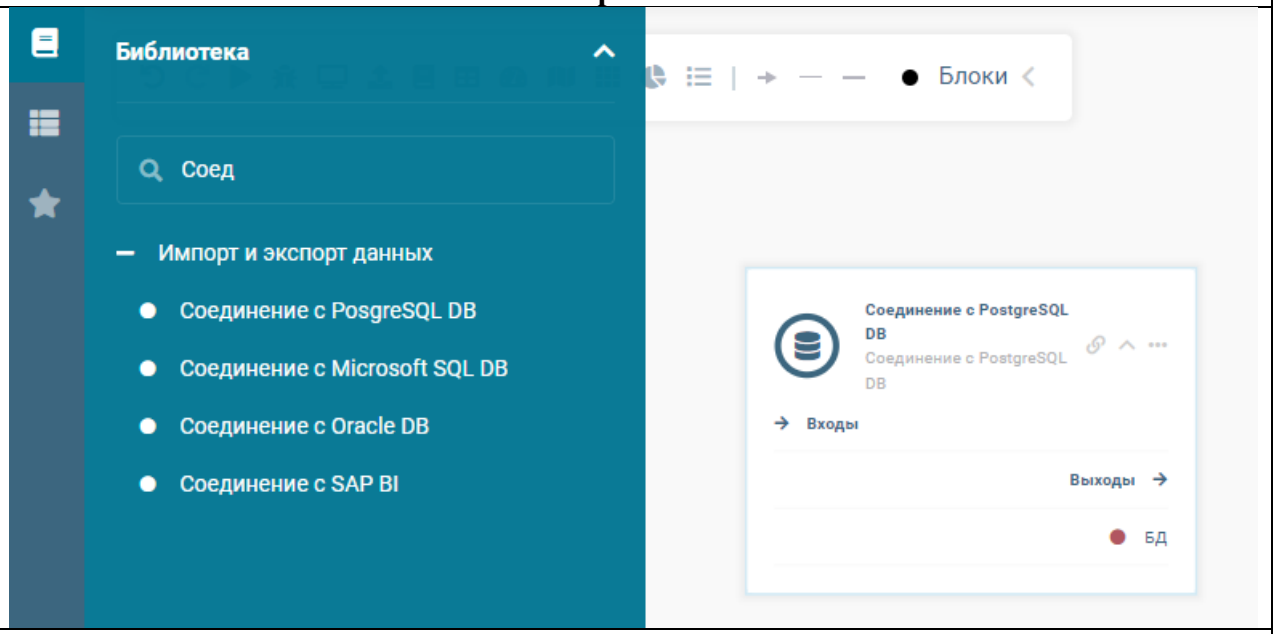
7.1.1 Добавление блоков графа

№	Действие	Скриншот
1	Открыть раздел «Библиотека» в левом боковом меню, нажав на него ЛКМ	
2	Найти в библиотеке нужный модуль для создания графа Библиотека имеет 10 основных модулей для создания графа: • Статистические методы • Методы экспертных систем (ЭС) • Импорт и экспорт данных • Элементы машинного обучения • Территории • Визуализаторы • Управление порядком расчета • Форматирование • Оформление графа • Преобразование данных Каждый модуль библиотеки разделяется на разное количество блоков, из которых и составляется граф	
3	Раскрыть модуль для просмотра блоков, нажав на значок «+»	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

№	Действие	Скриншот
4	Выбрать нужный блок раскрытого модуля	
5	**ИЛИ** вместо пунктов 2 – 4 Воспользоваться строкой «Поиск» для поиска необходимого блока	

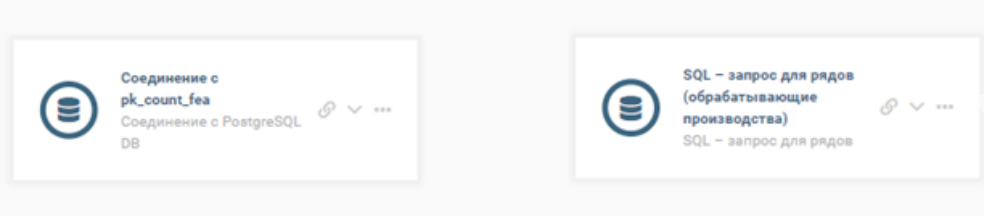
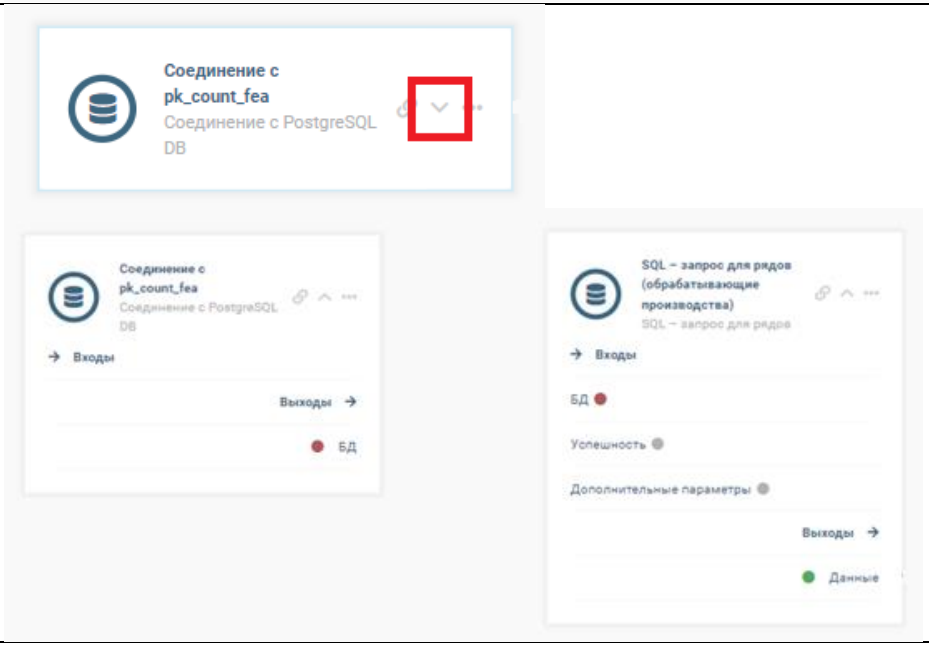
© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

№	Действие	Скриншот
		
6	Перетащить блок на рабочую область справа, зажав его ЛКМ	

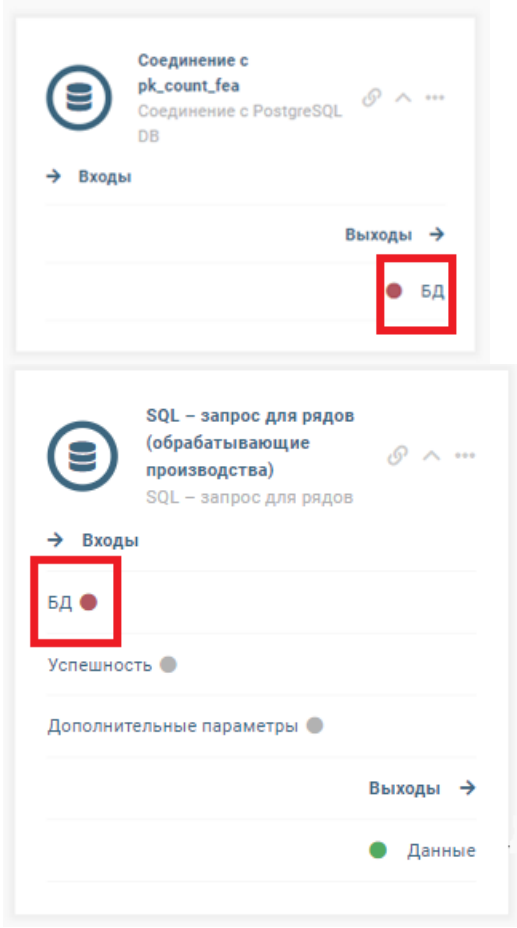
© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

7.1.2 Добавление связей между блоками

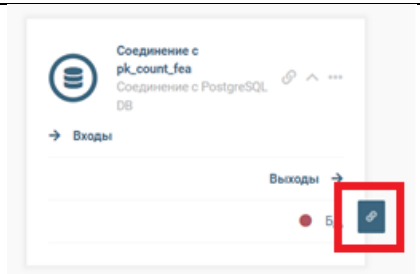
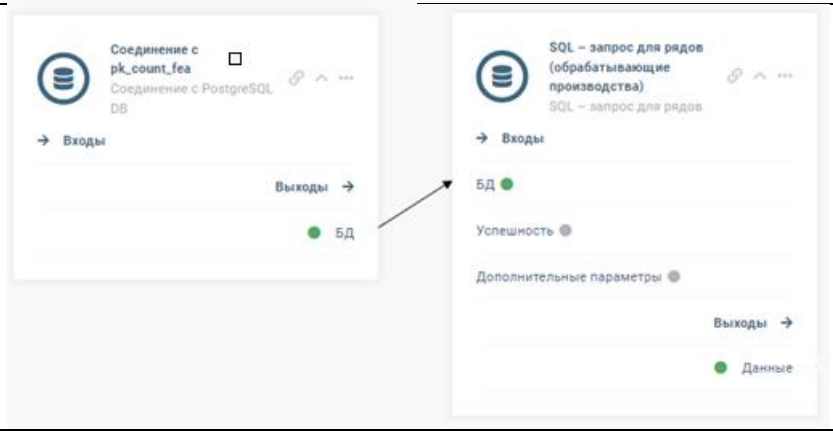
После добавления хотя бы двух блоков необходимо установить между ними связь

№	Действие	Скриншот
1	Выбрать два блока, между которыми будет установлена связь	
2	Раскрыть каждый блок, нажав на значок , расположенный справа на каждом блоке *(Пример двух раскрытых блоков)*	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

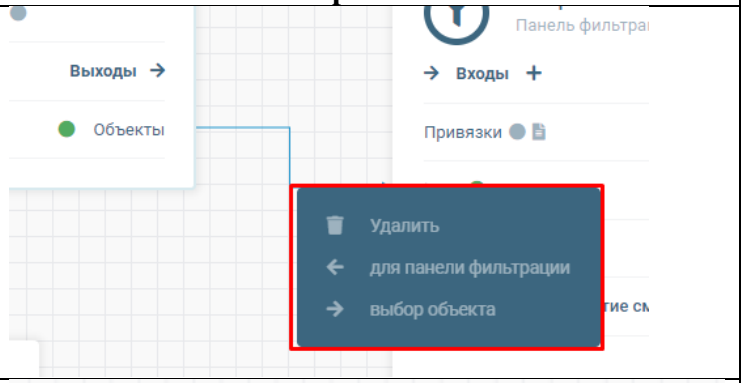
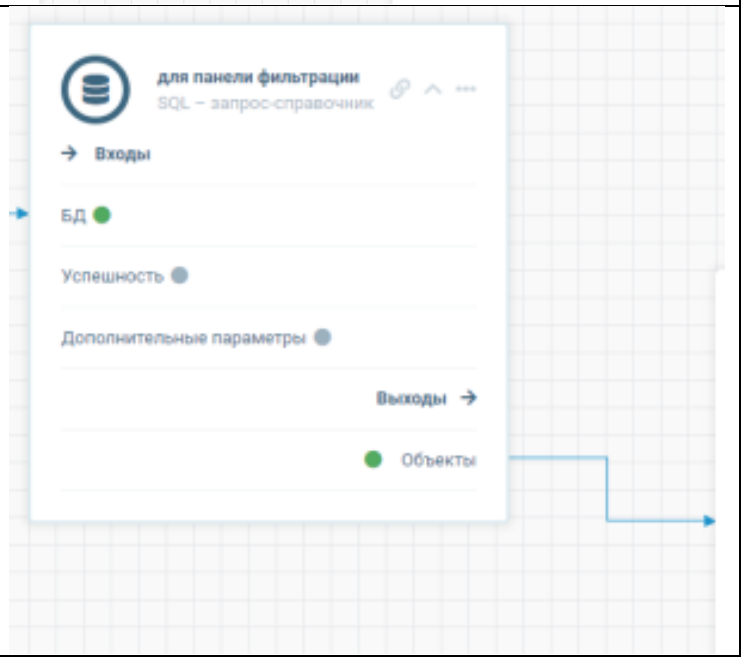
№	Действие	Скриншот
3	У одного блока выбрать **выходной параметр**, который нужно передать в другой блок У второго блока в свою очередь найти **входной параметр**, куда будут переданы данные	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

№	Действие	Скриншот
4	Навести мышкой на нужный **выходной параметр** первого блока *(Появится кнопка для создания связи с другим блоком)*	
5	Нажать на кнопку создания связи ЛКМ и протянуть связь до **входного параметра** второго блока Установить связь, нажав ЛКМ на **входной параметр** второго блока *(Если связь успешно установлена, то индикаторы **выходного и входного параметров** соответствующих блоков окрасятся в зеленый цвет)*	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

7.1.2.1 Переход к блоку через связь

№	Действие	Скриншот
1	Нажать ЛКМ по связи	
2	Нажать в меню ЛКМ по наименованию блока, к которому хотим перейти: • наименование со стрелкой соответствует блоку, из которого выходит связь • наименование со стрелкой соответствует блоку, в который входит связь	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

7.1.2.2 Пример графа после добавления всех блоков и связей

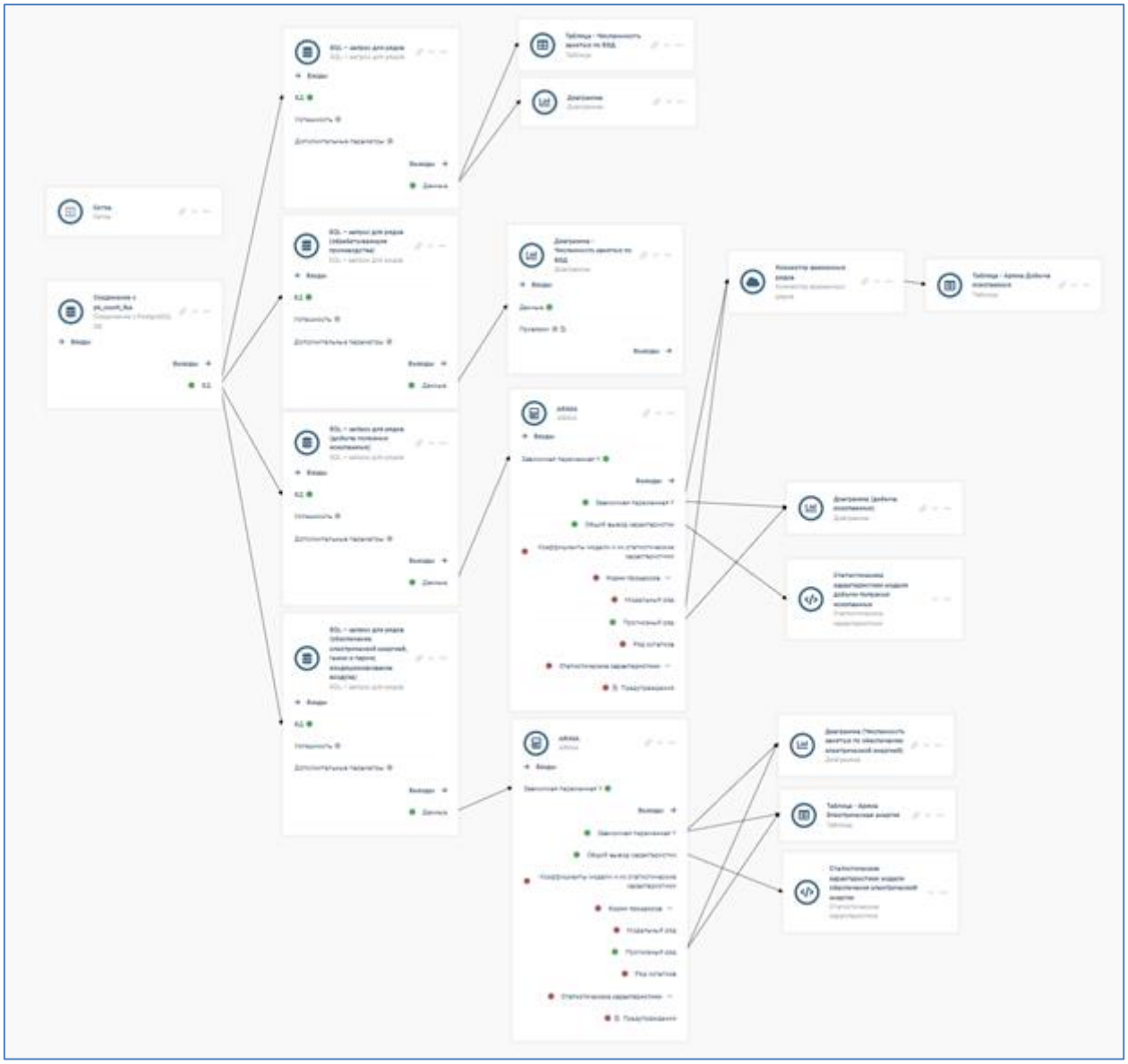


Рисунок 433 – Пример графа после добавления всех блоков и связей

7.1.3 Связи между блоками

На графе доступна возможность настройки внешнего вида связей между блоками.

Элементы настройки находятя в меню графа.

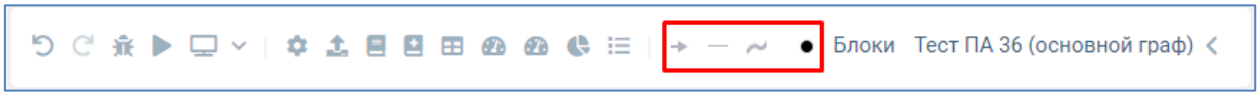


Рисунок 434 – Пример элементов настройки в меню

Чтобы их вызвать необходимо нажать по соответствующей кнопке.

Описание настроек связей (стрелок)

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Описание	Пример
Стиль стрелок	Выпадающий список	Задает стиль окончания связей соединяющих блоки в графе.	
Толщина линий	Выпадающий список	Задает толщину связей соединяющих блоки в графе.	
Стиль линий	Выпадающий список	Задает стиль линий связей соединяющих блоки в графе.	
Цвет линий	Цвет	Задает цвет линий связей соединяющих блоки в графе.	

Пример графа после добавления всех блоков и связей:

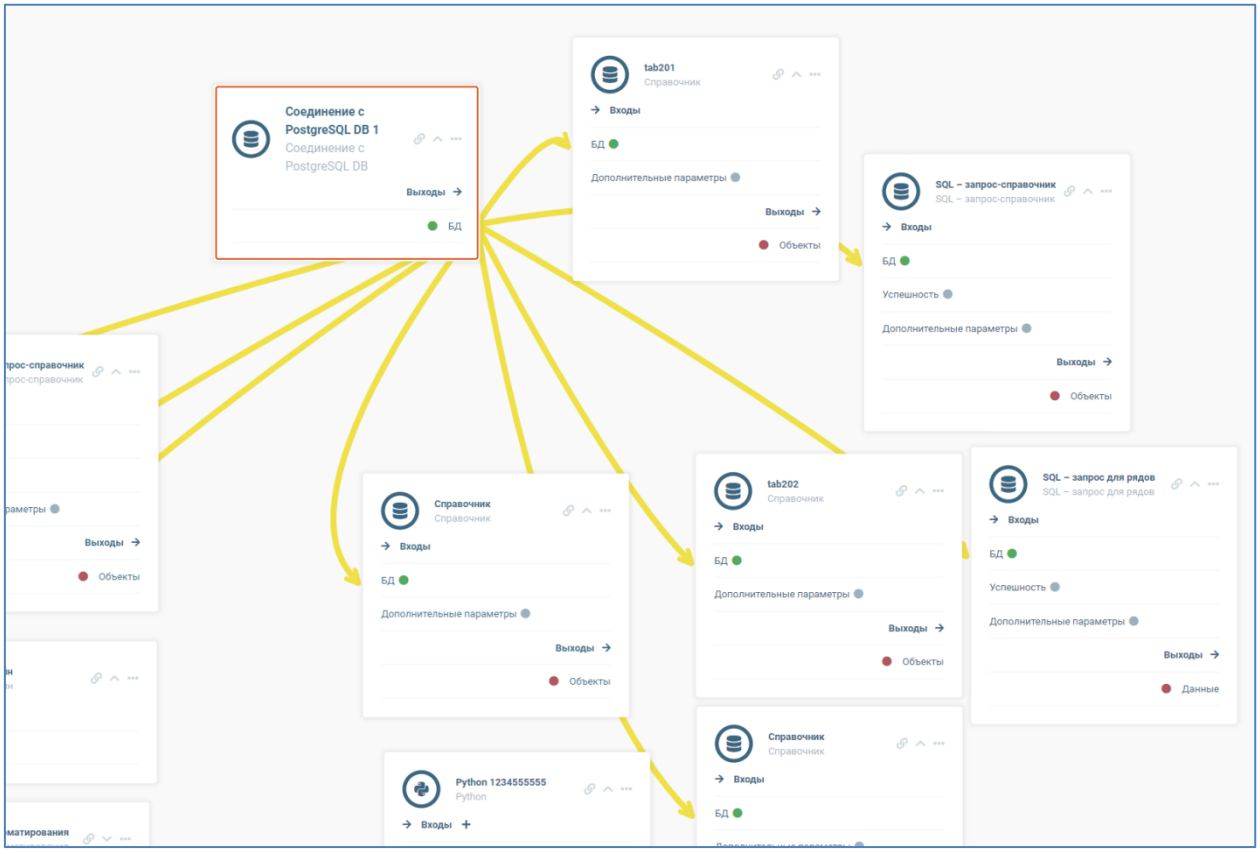


Рисунок 435 – Пример графа после добавления всех блоков и связей

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

7.2 Работа с блоками

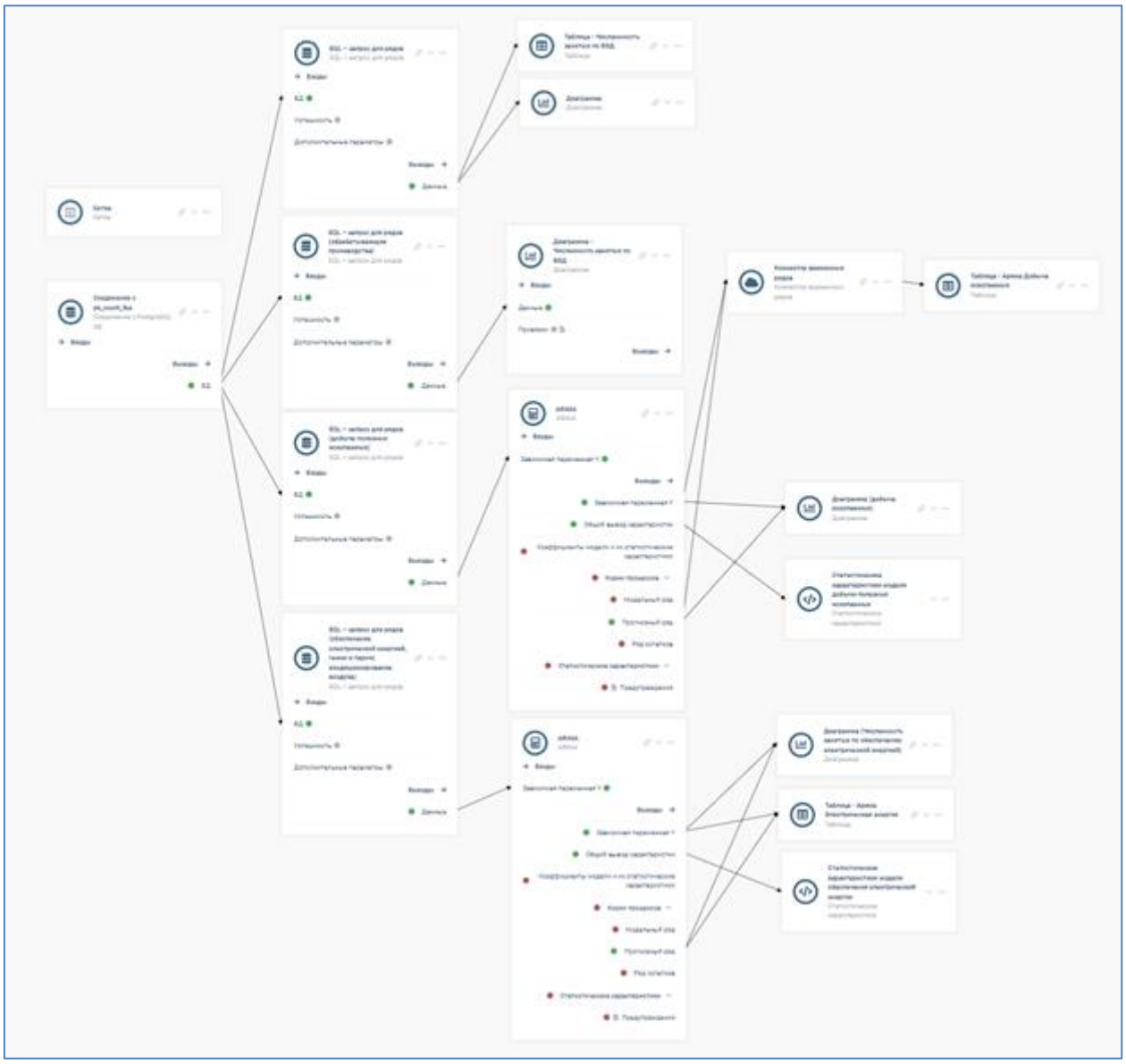


Рисунок 436 – Пример графа со связанными блоками

Блоки – элементы, из которых собирается граф и логика работы задачи. С помощью блоков можно выстраивать цепочки получения и преобразования данных, готовить визуализаторы и задавать логику взаимодействия с пользователем.

Блоки располагаются в библиотеке блоков. Пользователь можно перетащить блок на граф с помощью механизма Drag&Drop. Библиотека блоков подробно описана в разделе «Библиотека блоков» (см. п. 6).

При нажатии на блок он будет выбран – это будет видно по измененной границе блока.

7.2.1 Структура блока

Каждый блок имеет основные сегменты, которые видимы пользователю.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

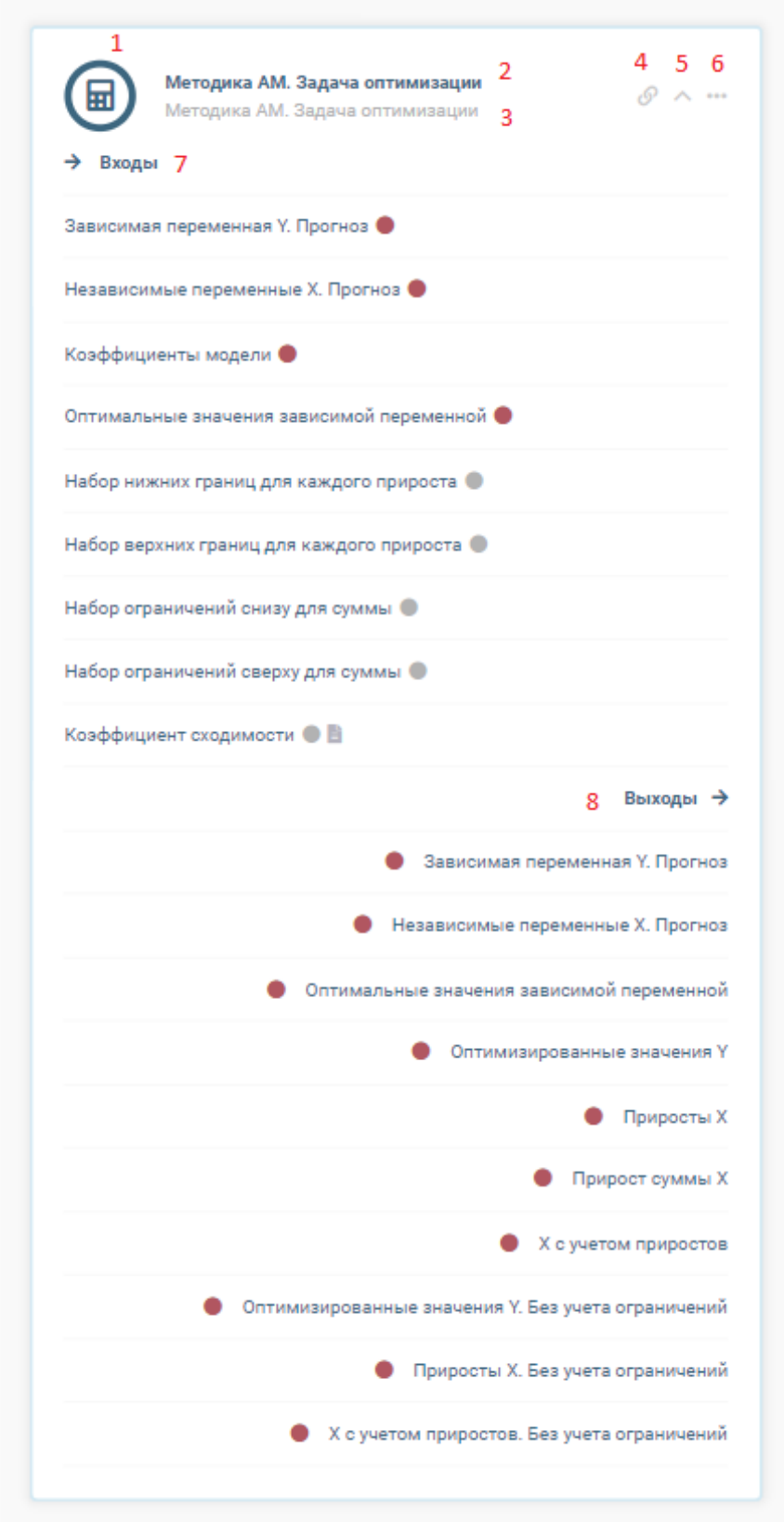
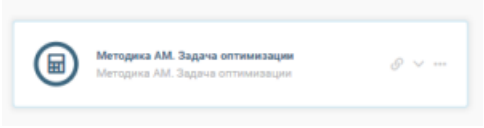


Рисунок 437 – Структура блока

1. Иконка блока – позволят быстро визуально определить тип блока.
2. Наименование блока – может быть изменено в настройках блока, по умолчанию является типом блока и совпадает с названием в библиотеке.
3. Тип блока – тип блока, неизменяемое.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

4. Ссылка – для добавленных блоков – ссылка на документацию, но может быть переопределена пользователем в настройках блока. При нажатии открывает ссылку в новом окне.
5. Иконка свернуть/развернуть блок – позволяет свернуть или развернуть блок, скрыв список входов и выходов. 

.
6. Вызов меню блока – открывает меню блока при нажатии ЛКМ.
7. Список входов блока, см. п. 7.2.2.3.2 «Входы, выходы и связи».
8. Список выходов блока, см. п. 7.2.2.3.2 «Входы, выходы и связи».

7.2.2 Оформление блока

В меню блоков доступна настройка оформления каждого блока индивидуально. Подробнее см. п. 7.2.2 «Оформление графа» (блоки, стрелки и т.д.).

7.2.2.1 Меню блока

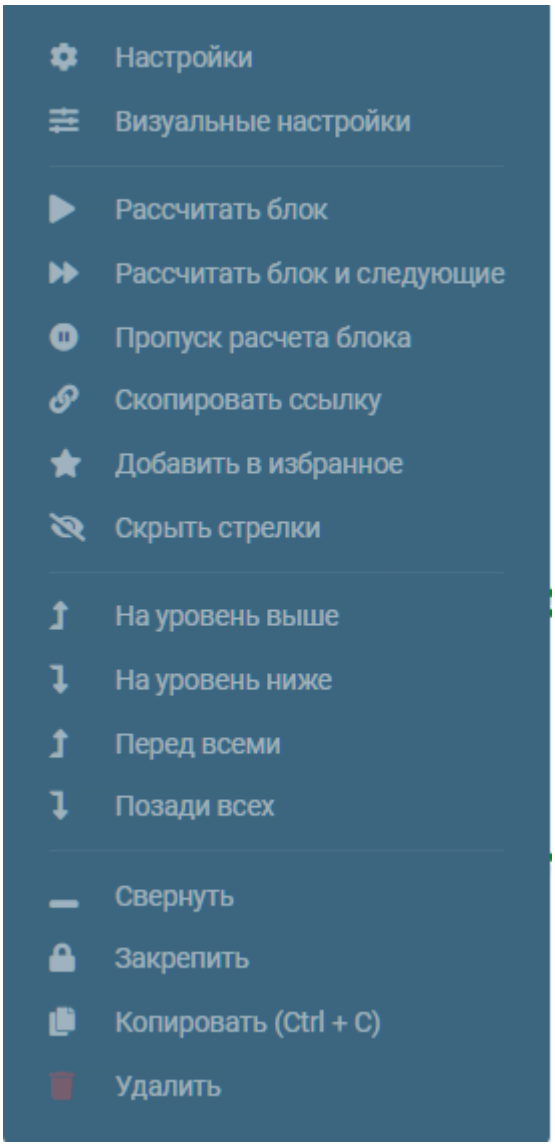


Рисунок 438 – Меню блока

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

7.2.2.2 Настройки

Открывает мастер настроек блока «Настройки блока» (см. п. 7.2.2.3).

Визуальные настройки

Открывает мастер визуальных настроек блока (см. п. 7.2.2).

7.2.2.2.1.1 Рассчитать блок

Запускает блок на расчет. В зависимости от того, включена или нет отладка пользователю будет выведен лог расчета в отдельном окне. Подробнее приведено в п. 7.3 «Расчет графа».

7.2.2.2.1.2 Рассчитать блок и следующие

Запускает на расчет блок и все блоки, идущие за ним по цепочке связей. В зависимости от того, включена или нет отладка пользователю, будет выведен лог расчета в отдельном окне. Подробнее приведено в п. 7.3 «Расчет графа».

7.2.2.2.1.3 Пропуск расчета блока

Выключить блок из расчета цепочек и графа целиком. Блок не будет рассчитываться при полном расчете графа.

Важно!

При выключении блока из расчета он будет пропущен, но блоки, стоящие за ним, будут рассчитываться.

При этом блок по прежнему можно считать через пункт меню «**Рассчитать блок**» и «**Рассчитать блок и следующие».**

При пропуске расчета блока он становится полупрозрачным. Настройка пропуска расчета так же доступна в настройках блока.

7.2.2.2.1.4 Скопировать ссылку

Скопировать в буфер обмена прямую ссылку на блок.

7.2.2.2.1.5 Добавить в избранное

Добавить блок в список избранных блоков. Можно задать тип блока, его наименование и выбрать раздел «Избранного», куда провести сохранение. Подробнее приведено в «Избранные блоки» (см. п. 7.11.3).

7.2.2.2.1.6 Скрыть стрелки

Скрывает с графа стелки входящие и выходящие из блока. При этом стрелки **не удаляются**.

При скрытии стрелок пункт меню изменяется на «Показать стрелки».

7.2.2.2.1.7 На уровень выше/На уровень ниже

Переместить блок на один уровень выше/ниже на графе.

7.2.2.2.1.8 Перед всеми/Позади всех

Переместить блок на передний/задний план в графе

7.2.2.2.1.9 Закрепить

При закреплении блока пользователь не сможет менять его местоположение на графе. После этого пункт меню меняется на «Открепить**».**

7.2.2.2.1.10 Копировать

Копирование блока. После этого можно вставить блок на граф, нажав правую кнопку на любом пустом месте графа и выбрав пункт «Вставить» или «Вставить со связями».

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Важно!

При вставке со связями будут так же скопированы все связи блока, при расчете он будет работать как полная копия исходного блока.

7.2.2.2.1.11 Удалить

Удаление блока с графа

При удалении пользователю будет задан вопрос для подтверждения в зависимости от типа блока.

1. Если это «простой» блок, у которого не зависимых от него структур, например таблиц – в вопросе будет подтверждение удаления.
2. Если блок является, например, Справочником, т.е. содержит под собой какие то структуры данных в БД – при подтверждении можно будет удалить и блок и его связанные таблицы.
3. При удалении нескольких блоков – подтверждения аналогичны.

Важно!

При удалении блоков с таблицами будет задан дополнительный вопрос с указанием необратимости операции

7.2.2.2.2 Контекстное меню

При нажатии на блок ПКМ будет выведено контекстное меню. Оно позволяет осуществлять быстрый доступ к части функций меню блока.

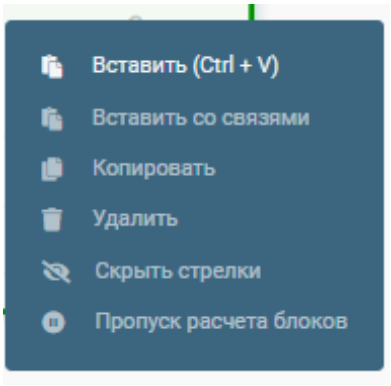


Рисунок 439 – Контекстное меню блока

7.2.2.3 Настройки блока

Каждый блок имеет несколько разделов настроек, доступных по двойному щелчку мышью на заголовок блока или через меню «Настройки».

В блоках есть раздел с группой «Общие настройки блоков» (см. п. 6.1.1.1.3) и специализированные настройки, описанные в документации по блоку.

При открытии настроек пользователь видит мастер настройки блока.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

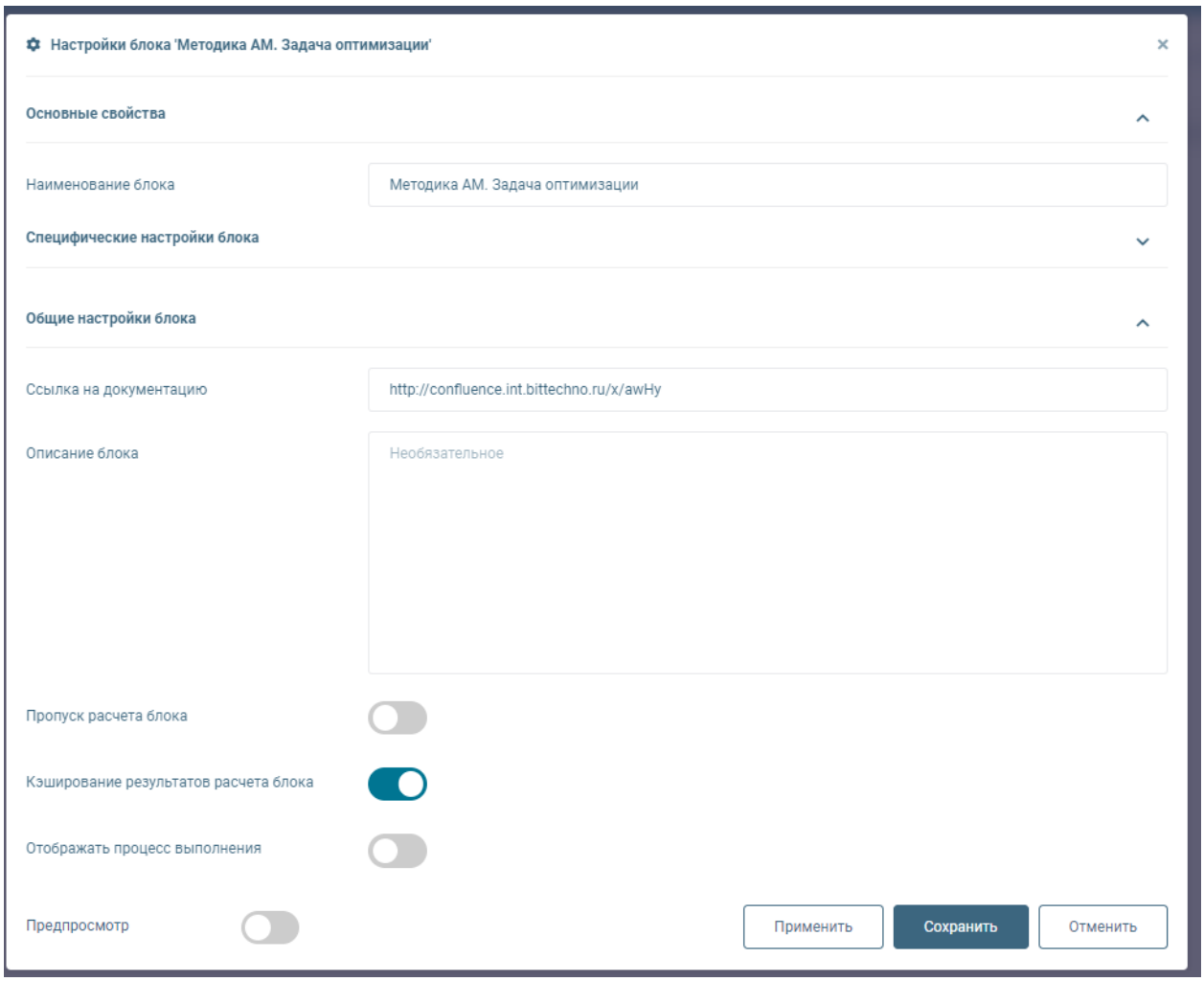


Рисунок 440 – Пример мастера настройки блока

При нажатии на «Применить» – настройки блока сохраняются, но мастер остается открытым.

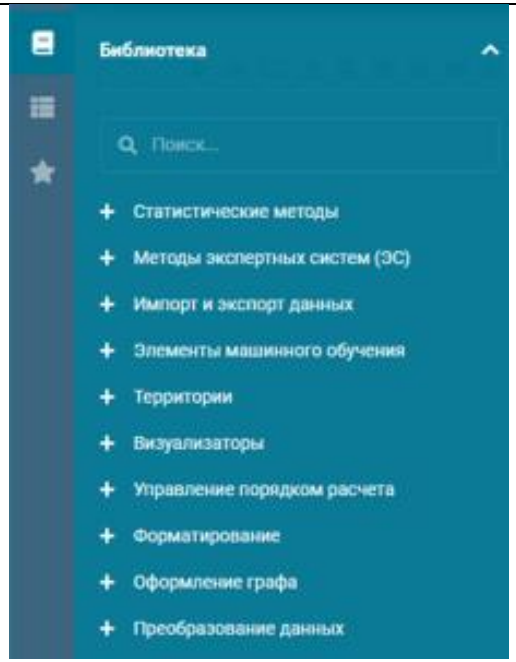
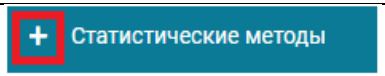
При нажатии на «Сохранить» – настройки блока сохраняются, мастер закрывается.

При нажатии на «Отменить» – мастер закрывается без сохранения измененных настроек.

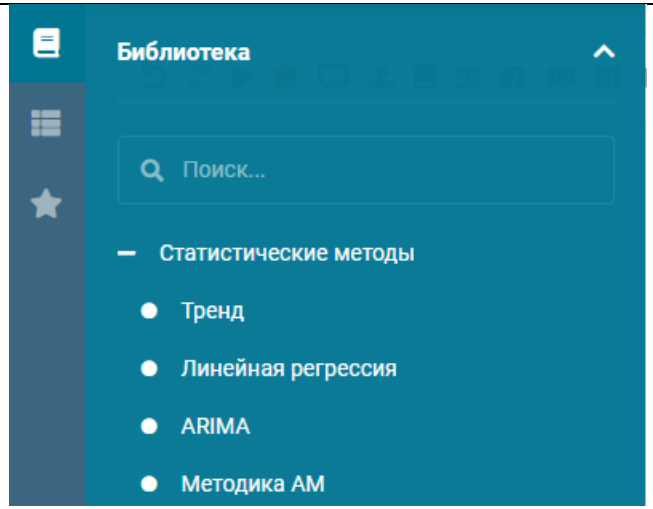
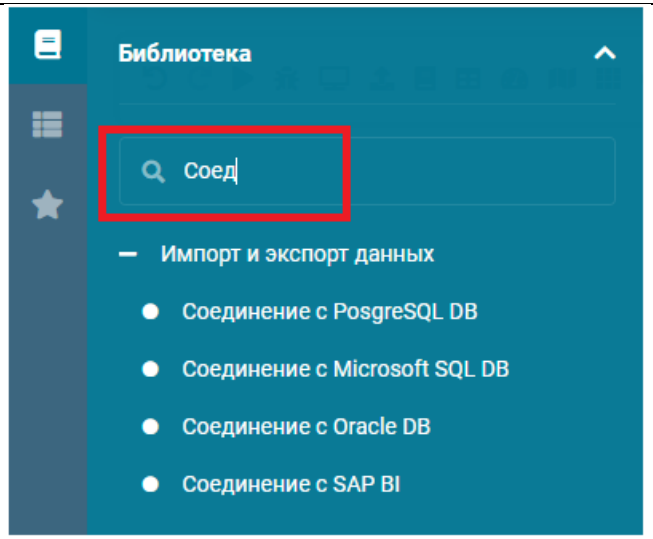
7.2.2.3.1 Добавление блоков на граф

Для добавления блока на граф достаточно «перетащить» его из библиотеки блоков с помощью Drag&Drop меахнизма.

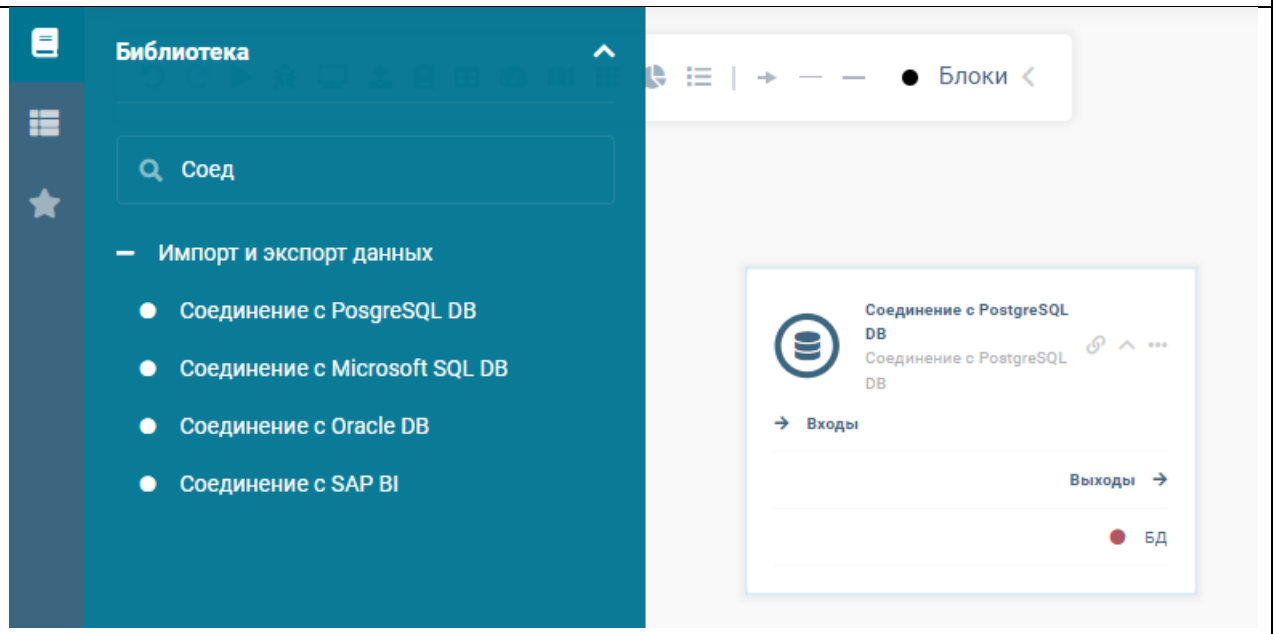
© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

№	Действие	Пример
1	Открыть раздел «Библиотека» в левом боковом меню, нажав на него ЛКМ	
2	Найти в библиотеке нужный модуль для создания графа Библиотека имеет 10 основных модулей для создания графа: • Статистические методы • Методы экспертных систем (ЭС) • Импорт и экспорт данных • Элементы машинного обучения • Территории • Визуализаторы • Управление порядком расчета • Форматирование • Оформление графа • Преобразование данных Каждый модуль библиотеки разделяется на разное количество блоков, из которых и составляется граф	
3	Раскрыть модуль для просмотра блоков, нажав на значок «+»	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

№	Действие	Пример
4	Выбрать нужный блок раскрытого модуля	
5	**ИЛИ** вместо пунктов 2 – 4 Воспользоваться строкой «Поиск» для поиска необходимого блока	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

№	Действие	Пример
6	Перетащить блок на рабочую область справа, зажав его ЛКМ	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

7.2.2.3.2 Входы, выходы и связи

Входы блока расположены в левой части блока. Выходы – в правой. Напротив каждого входа или выхода.

Блоки можно соединять друг с другом для выстраивания логики работы задачи. Каждый вход имеет свой тип и при попытке соединить выход и вход разных типов пользователь получит ошибку.

Если у блока нет входов и их нельзя создать, то раздел «Входы» не отображается.

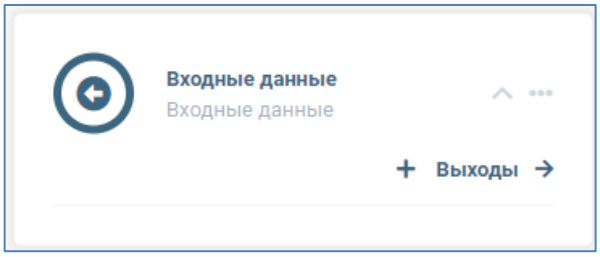


Рисунок 441 – Пример отсутствия раздела «Входы»

Если у блока нет выходов и их нельзя создать, то раздел «Выходы» не отображается

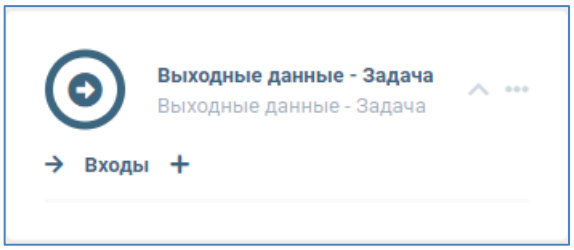
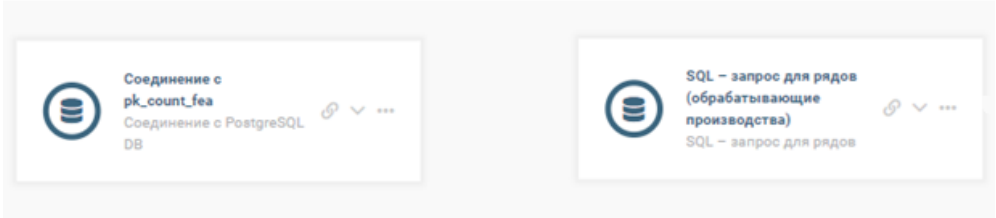
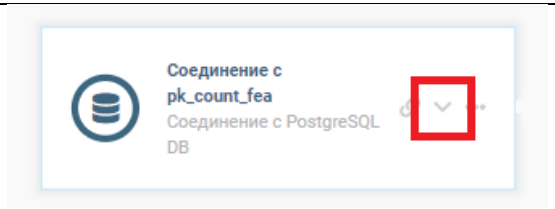
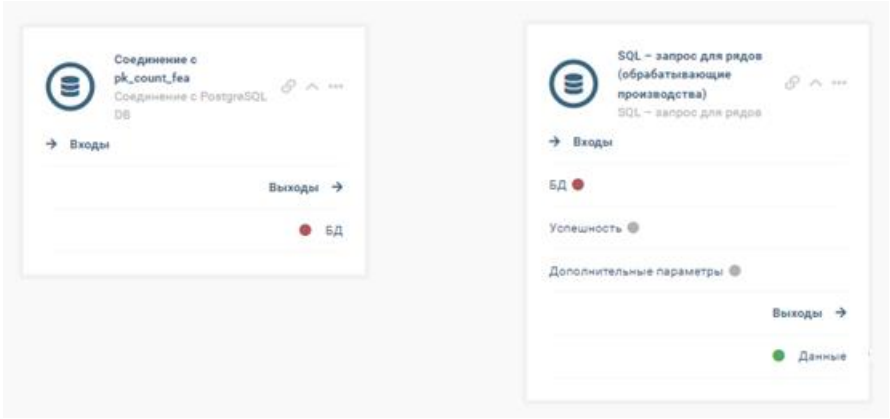


Рисунок 442 – Пример отсутствия раздела «Выходы»

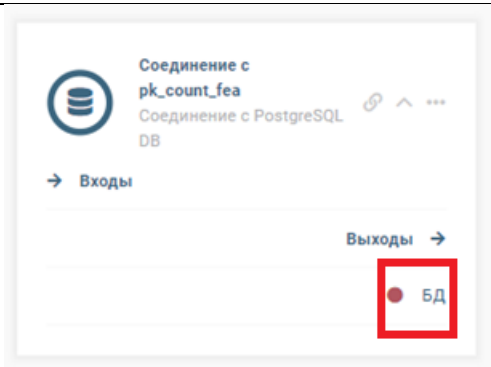
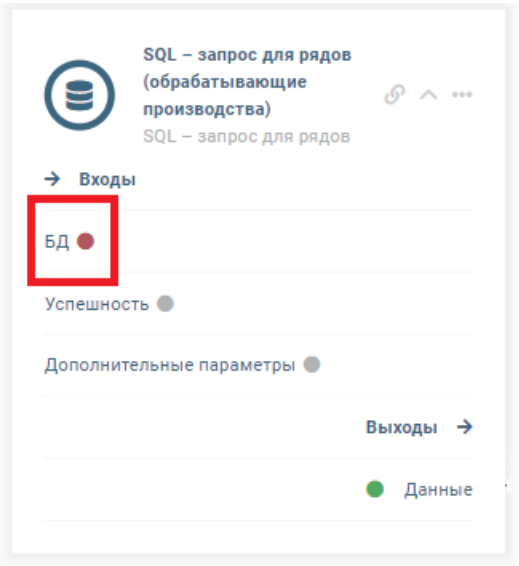
© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

7.2.2.3.3 Добавление связей между блоками

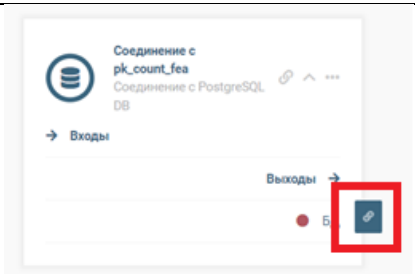
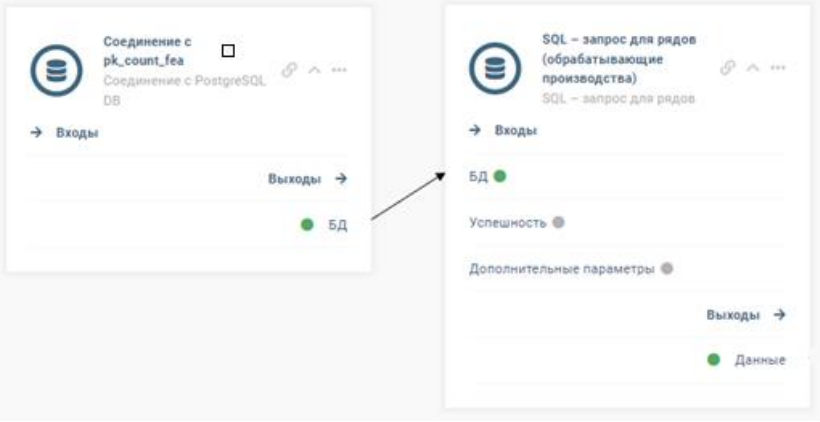
После добавления хотя бы двух блоков необходимо установить между ними связь.

1	Выбрать два блока, между которыми будет установлена связь	
2	Раскрыть каждый блок, нажав на значок , расположенный справа на каждом блоке *(Пример двух раскрытых блоков)*	 

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

3	У одного блока выбрать **выходной параметр**, который нужно передать в другой блок У второго блока в свою очередь найти **входной параметр**, куда будут переданы данные	 

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

4	Навести мышкой на нужный **выходной параметр** первого блока *(Появится кнопка для создания связи с другим блоком)*	
5	Нажать на кнопку создания связи ЛКМ и протянуть связь до **входного параметра** второго блока. Установить связь, нажав ЛКМ на **входной параметр** второго блока *(Если связь успешно установлена, то индикаторы **выходного и входного параметров** соответствующих блоков окрасятся в зеленый цвет)*	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

7.2.2.3.4 Добавление входов и выходов

Некоторые блоки позволяют управлять входами и выходами, их количеством и названием. В таких блоках, рядом с входами и выходами, есть иконка добавления нового входа/выхода. Например, блок «Python» (подробно п. 6.9.1).

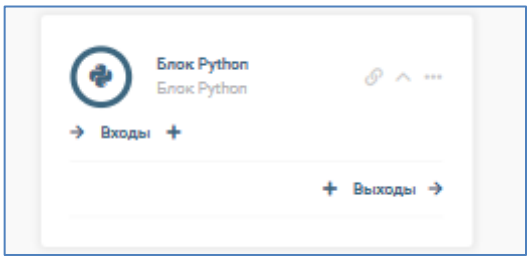


Рисунок 443 – Блок «Python»

При нажатии на значок «+» откроется окно добавления нового входа/выхода.

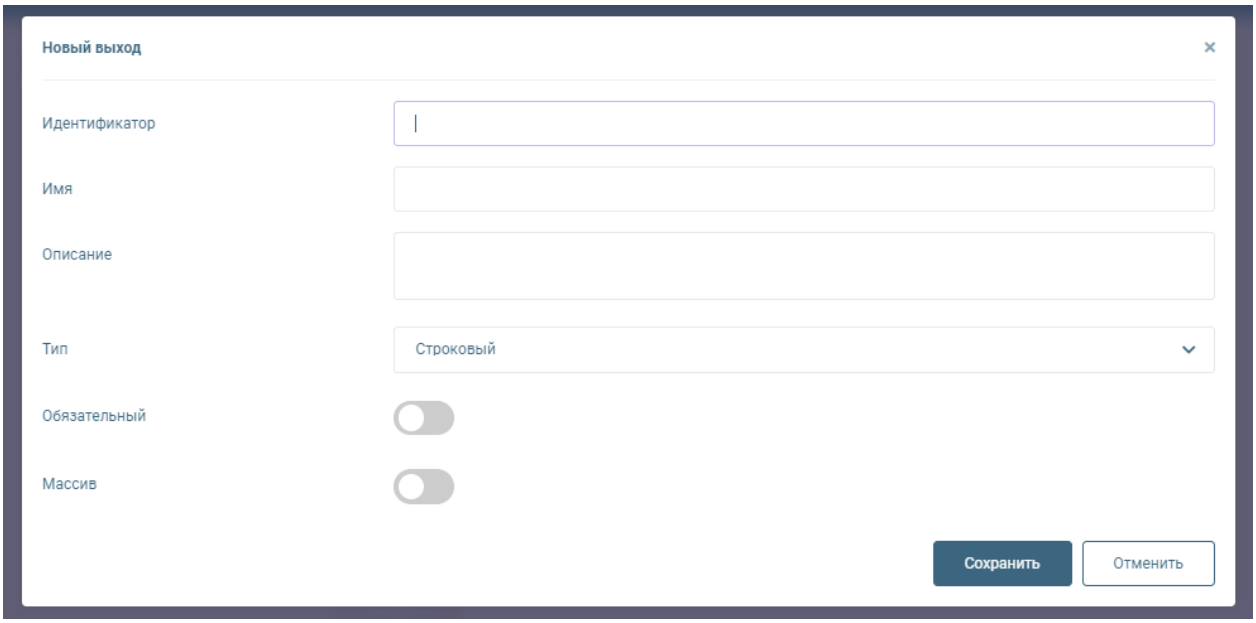


Рисунок 444 – Окно добавления нового входа/выхода

Пользователь можно задать:

- Идентификатор – уникальный идентификатор входа/выхода.

  Важно! После создания идентификатор изменять нельзя.
- Имя входа.
- Описание входа.
- Выбрать Тип входа – для соединения с входами/выходами разных типов.
- Флаг Обязательный – указывает на обязательность входа/выхода для расчета.
- Флаг Массив – указывает на то, будет ли приходить во вход массив.

7.2.2.4 Режимы работы блоков

При включении режима отладки графа внешний вид блоков будет изменен – добавится индикация результатов расчета блока. Подробнее приведено в п. 7.5.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

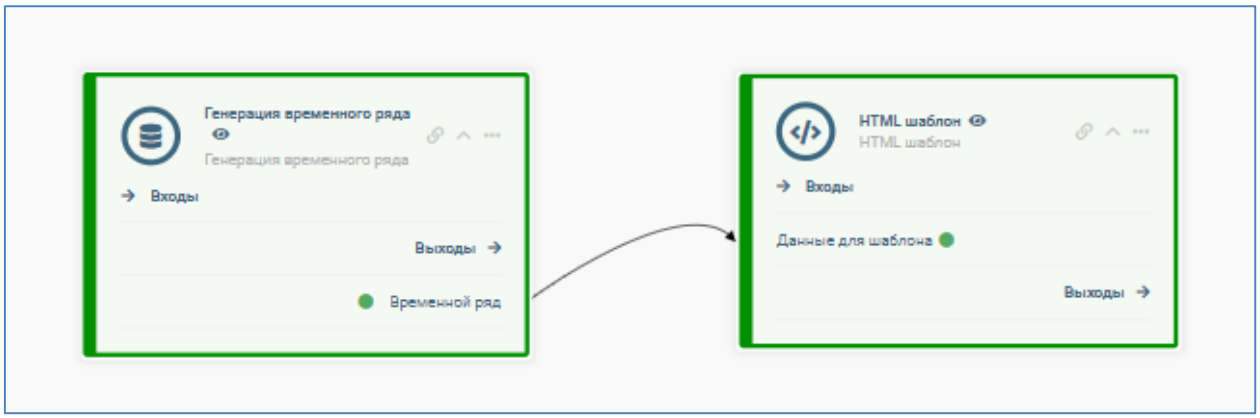


Рисунок 445 – Пример режима отладки графа

7.2.2.5 Выбор нескольких блоков и операции с ними

Для выбора нескольких блоков необходимо нажимать на них ЛКМ с **зажатой** клавишей CTRL. Вокруг выбранных блоков будет отображаться граница выделенного блока. После этого блоки можно передвигать по графу группой. При нажатии на любой из блоков ПКМ откроется контекстное меню, аналогичное меню для одного блока.

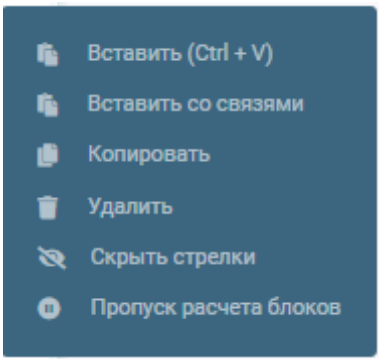


Рисунок 446 – Контекстное меню

7.2.2.6 Блокировка блока другим пользователем

В графе могут работать одновременно несколько пользователей. Для избегания ситуации, когда два разных пользователя изменяют настройки одного блока, во время изменения настроек он блокируется для всех пользователей, с индикацией того пользователя, который заблокировал блок.

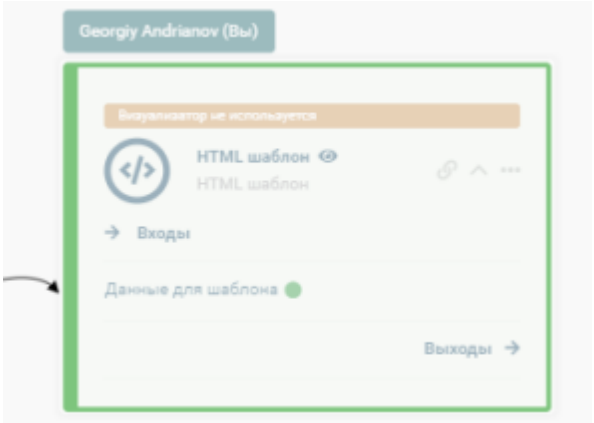


Рисунок 447 – Пример блокировки блока

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

7.2.2.7 Индикации блоков

7.2.2.8 Неиспользуемые блоки визуализаторы

Если блок визуализатор добавлен на граф, но при этом не был добавлен ни на один пересет, т.е. не используется визуализатор, в режиме отладки у блока будет выведена индикация об этом.

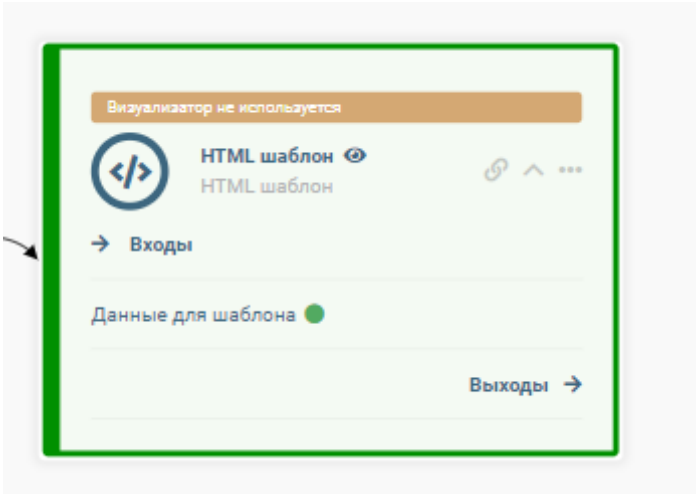


Рисунок 448 – Индикация о неиспользовании блока

7.2.2.9 Устаревшие блоки

Помимо режима отладки блоки могут иметь дополнительную индикацию о том, что блок является устаревшим и будет вскоре выведен из эксплуатации.

Пример индикации в библиотеке.

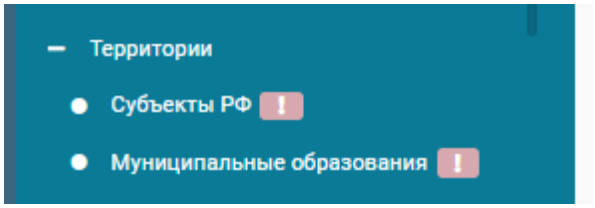


Рисунок 449 – Пример индикации в библиотеке

Пример индикации в блоке на графе (отображается только в режиме отладки):

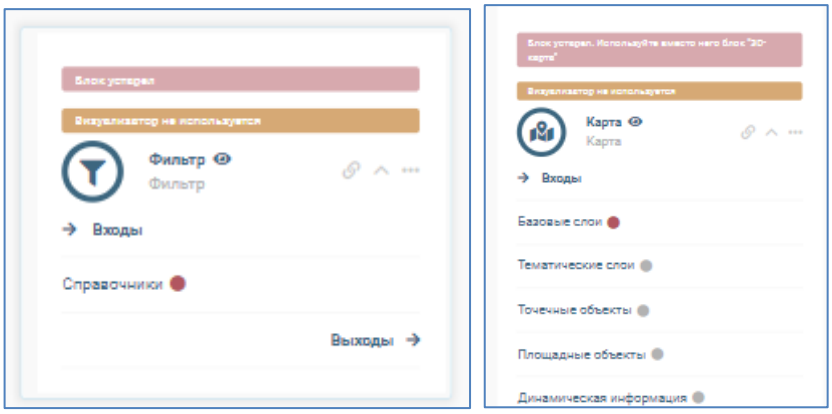


Рисунок 450 – Пример индикации в блоке на графе

Важно! Устаревшие блоки по-прежнему будут рассчитываться, однако их поддержка и проверка не осуществляется. Поэтому рекомендуется заменять такие блоки на альтернативные.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

7.2.3 Групповые операции

Граф поддерживает работу с несколькими блоками одновременно

Выбрать несколько блоков можно следующим образом:

- зажать кнопку Shift и с зажатой ЛКМ рамкой выделить несколько блоков;
- зажать кнопку Ctrl и, кликая ЛКМ по блокам, выделять их по одному.

Пользователь может:

- перетащить выбранные блоки, зажав ЛКМ;
- открыть меню группы блоков, нажав ПКМ.

7.2.3.1 Меню группы блоков

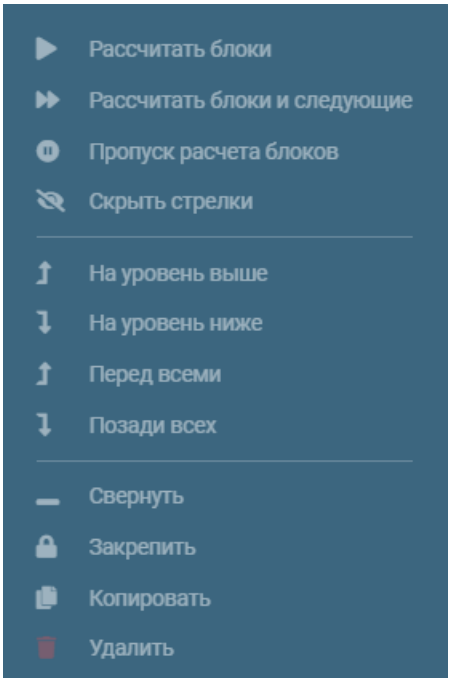


Рисунок 451 – Меню группы блоков

- Рассчитать блоки. Запускает выбранные блоки на расчет.
- Рассчитать блоки и следующие. Запускает на расчет выбранные блоки и все блоки, идущие за ними по цепочке связей.
- Пропуск расчета блоков/Рассчитывать блоки. Выключить/включить блоки из расчета цепочек и графа целиком.
- Скрыть стрелки/Отобразить стрелки. Скрывает/отображает в графе стрелки (связи) входящие и выходящие из блоков.
- На уровень выше. Переместить блоки на один уровень выше в графе.
- На уровень ниже. Переместить блоки на один уровень ниже в графе.
- Перед всеми. Переместить блоки на передний план в графе.
- Позади всех. Переместить блоки на задний план в графе.
- Свернуть/Развернуть. Сворачивает/разворачивает входы и выходы блоков.
- Закрепить/Открепить. Не позволяет/позволяет перемещать блоки в графе.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

- Копировать. После этого можно вставить блоки на граф, нажав правую кнопку на любом пустом месте графа и выбрав пункт «Вставить» или «Вставить» со связями.
- Удалить. Удаление блоков с графа.

7.2.4 История изменений блоков

На графе есть возможность просматривать и сравнивать историю изменений блоков.

Для этого в меню графа "Настройки задачи" необходимо включить флаг "Хранение истории изменения настроек". Изменения, которые были произведены при выключенном флаге, не будут фиксироваться.

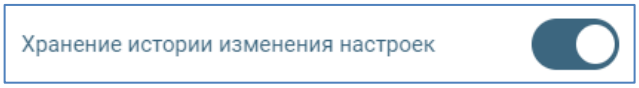


Рисунок 452 – Настройка «Хранение истории изменения настроек»

Когда флаг "Хранение истории изменения настроек" включен, у блоков в локальном меню появляются 2 дополнительных пункта:

- История
- Сравнить

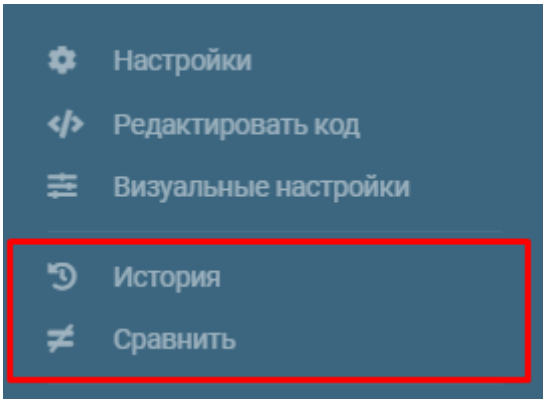


Рисунок 453 – Пункты истории в меню блоков

При открытии пункта история можно выбрать версию блока, которая сохранена ранее и посмотреть изменения, произведенные в ней.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

7.2.4.1 История

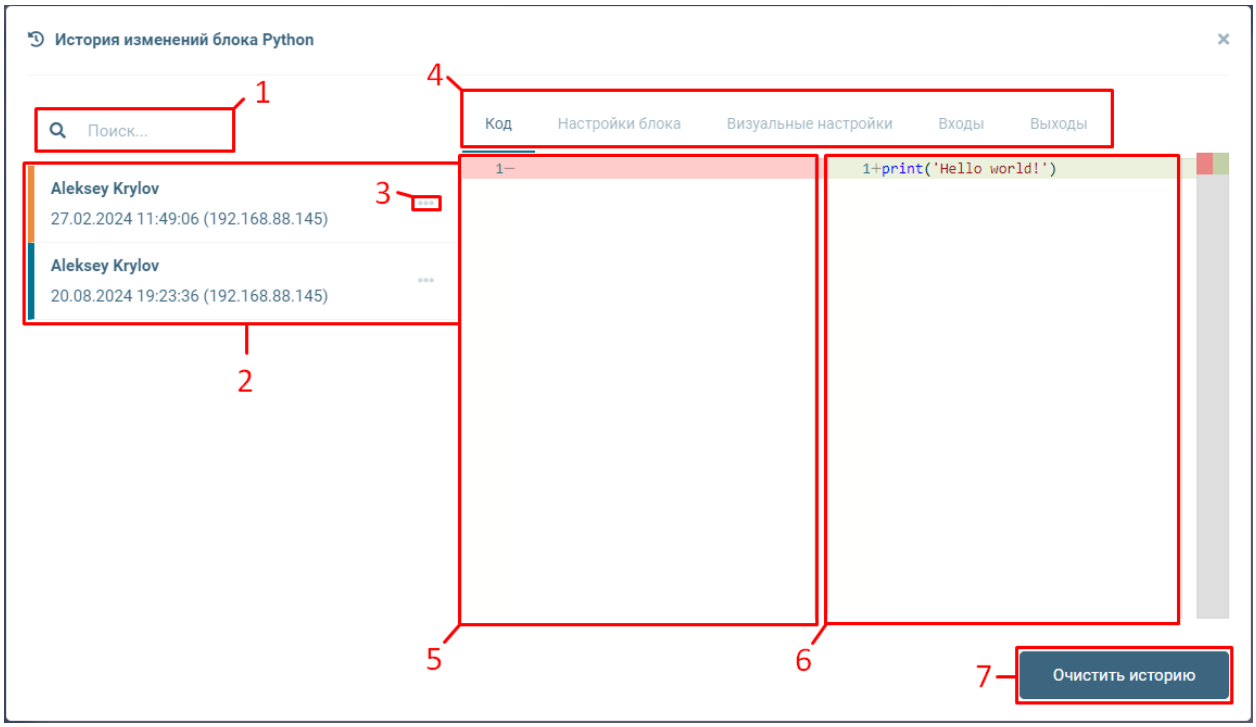


Рисунок 454 – Пример окна «История изменений блока Python»

1. Поле ввода для осуществления поиска по автору изменений
2. Список версий. Позволяет выбрать версию. Отсортирован по дате внесения изменений в порядке возрастания
3. Кнопка открытия контекстного меню версии. В контекстном меню располагается кнопка "Удалить", позволяет удалить версию из истории, при этом изменения, которые были сделаны в рамках нее сохраняются.
4. Меню, в котором представлены варианты изменений блока
    a. **Код**. Изменения, которые были произведены в рамках меню блока "Редактировать код". Актуально для блоков, у которых в меню есть пункт "Редактировать код"
    b. **Настройки блока**. Изменения, которые были произведены в рамках меню блока "Настройки"
    c. **Визуальные настройки**. Изменения, которые были произведены в рамках меню блока "Визуальные настройки"
    d. **Входы**. Изменения по добавлению/редактированию/удалению входов блока. Изменение данных, поступающих на вход из других блоков, не учитываются
    e. **Выходы**. Изменения по добавлению/редактированию/удалению выходов блока. Изменение данных, выходящих из блока, не учитываются
5. Окно, в котором отображается предыдущая версия блока
6. Окно, в котором отображается выбранная версия блока
7. Кнопка "Очистить историю". Удаляет все версии блока, при этом изменения, которые были сделаны в рамках этих версий, сохраняются

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

7.2.4.2 Сравнить

Для корректного отображения изменений слева следует выбирать более раннюю версию относительно версии справа.

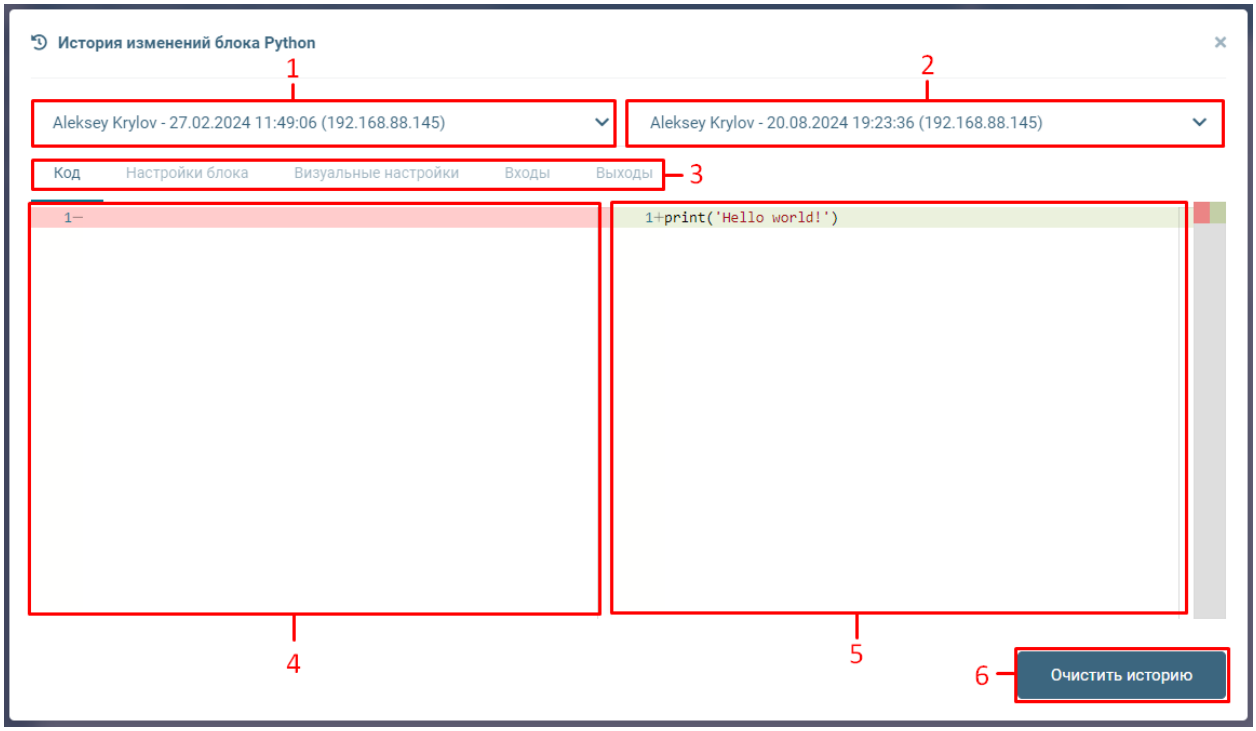


Рисунок 455 – Пример окна сравнения «История изменений блока Python»

1. Выпадающий список версий блока слева (выбранная версия должна быть более ранняя, чем справа)
2. Выпадающий список версий блока справа (выбранная версия должна быть более поздняя, чем слева)
3. Меню, в котором представлены варианты изменений блока
    a. Код. Изменения, которые были произведены в рамках меню блока "Редактировать код". Актуально для блоков, у которых в меню есть пункт "Редактировать код"
    b. Настройки блока. Изменения, которые были произведены в рамках меню блока "Настройки"
    c. Визуальные настройки. Изменения, которые были произведены в рамках меню блока "Визуальные настройки"
    d. Входы. Изменения по добавлению/редактированию/удалению входов блока. Изменение данных, поступающих на вход из других блоков, не учитываются
    e. Выходы. Изменения по добавлению/редактированию/удалению выходов блока. Изменение данных, выходящих из блока, не учитываются
4. Окно, в котором отображается выбранная версия блока слева
5. Окно, в котором отображается выбранная версия блока справа
6. Кнопка "Очистить историю". Удаляет все версии блока, при этом изменения, которые были сделаны в рамках этих версий, сохраняются

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

7.2.5 Поддерживаемые типы входов и выходов в блоке

Типы входов и выходов унифицированы. Для примера будут разобраны только выходы блоков (если такие блоки есть, если нет, будет использован блок Python). Для примера входов будет использован один блок Python.

Корректность данных проверяется только для входных данных в момент расчета блока.

7.2.5.1 Целочисленный тип

Передает целое число.

Пример блока «Python» (подробно см. п. 6.9.1).

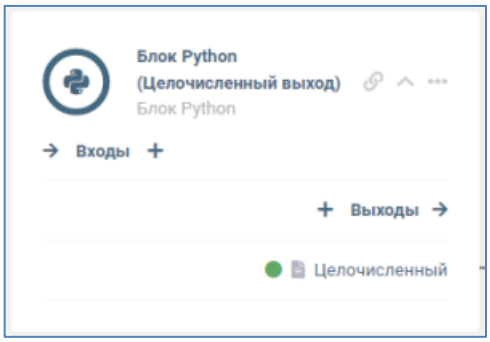


Рисунок 456 – Пример блока «Python»

7.2.5.2 Вещественный тип

Передает вещественное число. Количество знаков после запятой не ограничено.

Пример блока «Слайдер» (подробно см. п. 6.1.4.15).

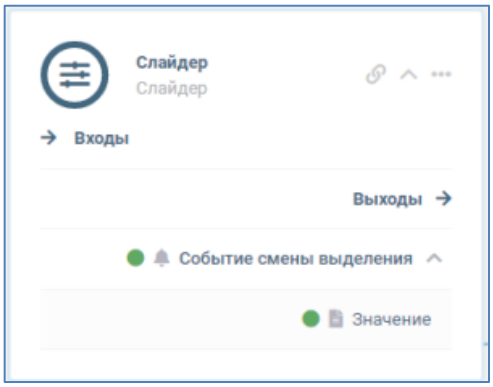


Рисунок 457 – Пример блока «Слайдер»

7.2.5.3 Строковый тип

Передает значения в строковом виде. Также в нем возможно передать JSON в строковом виде.

Пример Блока «Панель OLAP» (см. п. 6.1.4.8).

Все входы «Привязки» или «Параметры привязки» у таких блоков как «Таблица» (п. 6.1.2.3) или «Диаграмма» (п. 6.1.1.2) имеют строковый тип.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

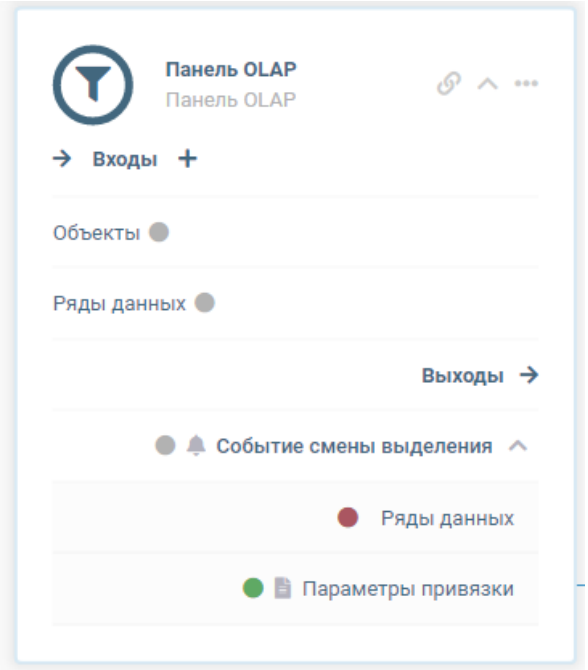


Рисунок 458 – Пример блока «Панель OLAP»

7.2.5.4 Дата и время

Передает значения даты с временной меткой.

Пример Блока «Python» (п. 6.9.1).

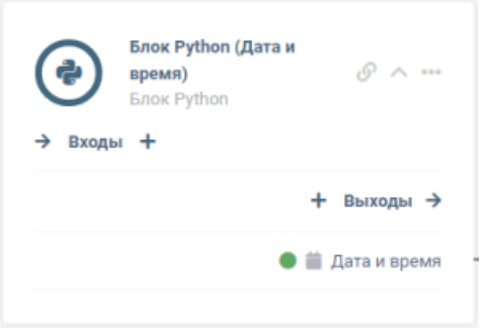


Рисунок 459 – Пример блока «Python (Дата и время)»

7.2.5.5 Дата

Передает значения даты без учета времени.

Пример Блока «Python» (п. 6.9.1).

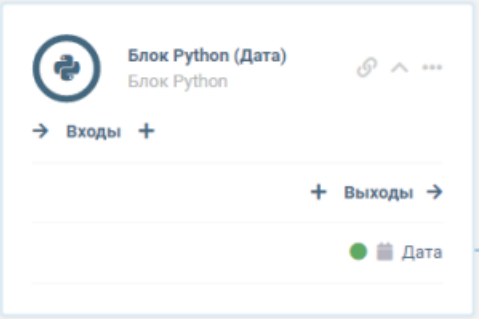


Рисунок 460 – Пример блока «Python (Дата)»

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

7.2.5.6 Логический

Передает логическое значение (ПРАВДА/ЛОЖЬ, 0/1).

Пример Блока «Кнопка» (п. 6.10.1).

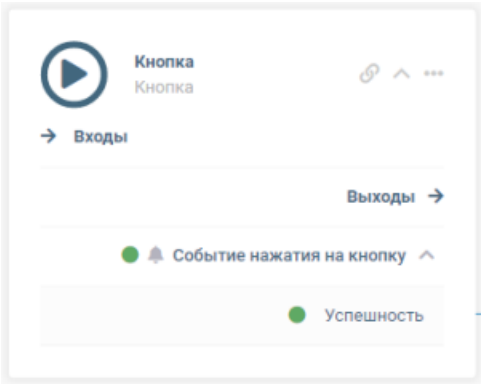


Рисунок 461 – Пример блока «Кнопка»

7.2.5.7 Ряд данных

Передает сформированный ряд с данными.

Формат ряда данных:

```
{
    "pkey": [
        <Набор первичных ключей>
    ],
    "vl": [
        <Массив значений ряда>
    ],
    "attrs": [
        <Набор атрибутов - полей, не попавших ни в первичный ключ, ни в
значение.>
    ],
    "_params": {
        <Опциональные параметры объекта>,
        "vl": <Ключ поля, отвечающего за значение>,
        "fkey": [
            <Ключ поля, по которому собран ряд>
        ]
    },
    "_source": <Идентификатор блока-источника, поле заполняется калькулятором
автоматически>,
}
```


© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Пример блока «Показатели» (см. п. 6.2.17):

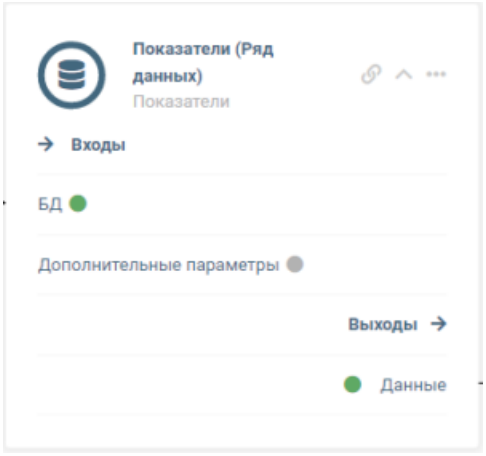


Рисунок 462 – Пример блока «Показатели (Ряд данных)»

7.2.5.8 Объект

Передает сформированный объект.

Формат объекта

```
{
    "pkey": [
        <Набор первичных ключей>
    ],
    "vl": <Значение объекта>,
    "attrs": {
        <Набор атрибутов - полей, не попавших ни в первичный ключ, ни в значение>
    },
    "_params": {
        <Опциональные параметры объекта>,
        "vl": <Ключ поля, отвечающего за значение>
    },
    "_source": <Идентификатор блока-источника, поле заполняется калькулятором автоматически>,
}
```

Пример Блока «Справочник» (см. п. 6.2.16):

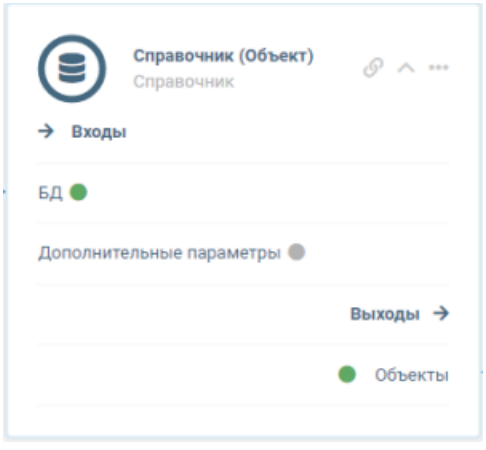


Рисунок 463 – Пример блока «Справочник (Объект)»

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

7.2.5.9 JSON (Структура)

Передает JSON объект. В отличии от передачи через строковый тип данных, здесь не нужно приводить JSON к строке для передачи.

В отличии от остальных типов, на вход «JSON» можно подавать любые типы значений, без возникновения ошибок.

Этот механизм используется для формирования шаблона в блоке «HTML-шаблон» (п. 6.1.4.2), или для формирования запроса через вход «Дополнительные параметры» у SQL-блоков (к примеру, у блока «SQL – запрос для рядов» (см. п. 6.2.10)).

Пример Блока «Слой данных для 3D-карты» (п. 6.1.3.5) (оба выхода имееют тип «JSON»):

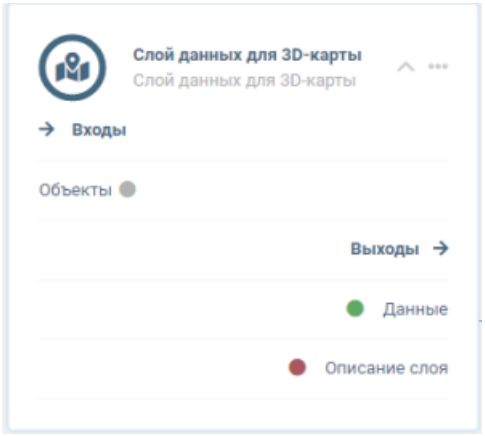


Рисунок 464 – Пример блока «Слой данных для 3D-карты»

7.2.5.10 HTML

Передает строку в формате HTML.

Несмотря на свое название, блок «HTML-шаблон» (см. п. 6.1.4.2) имеет вход типа «Структура» (см. выше), в отличии от блока «HTML-статика» (см. п. 6.1.4.1), у которого тип входа именно HTML.

Пример Блока «Python» (см. п. 6.9.1).

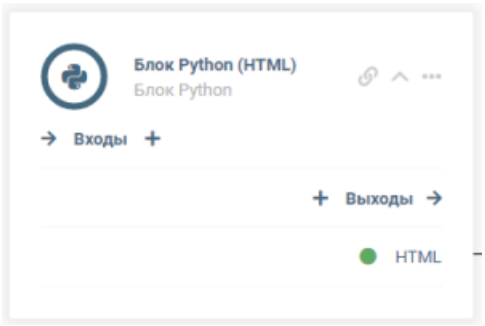


Рисунок 465 – Пример блока «Python (HTML)»

7.2.5.11 XML

Передает строку в формате XML.

Пример Блока «Соединение SOAP» (см. п. 6.2.6):

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

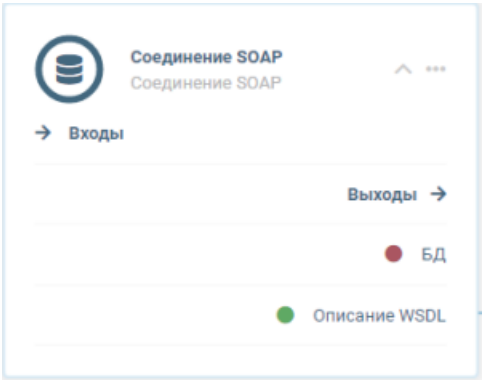


Рисунок 466 – Пример блока «Соединение SOAP»

7.2.5.12 Соединение с БД

В данном разделе собраны все варианты соединений с базой данных.

Они все имеют одинаковую логику работы:

- внутри передается уникальный идентификатор соединения;
- в момент расчета графа, по этому идентификатору можно получить открытое соединение с базой данных;
- после завершения расчета соединение закрывается и уникальный идентификатор становится не валидным;
- в момент начала следующего расчета (а именно, в момент расчета блока «...соединение с БД») открывается новое соединение и генерируется новый идентификатор.

Пример блока «Соединение с PostgreSQL DB» (см. п. 6.2.1).

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

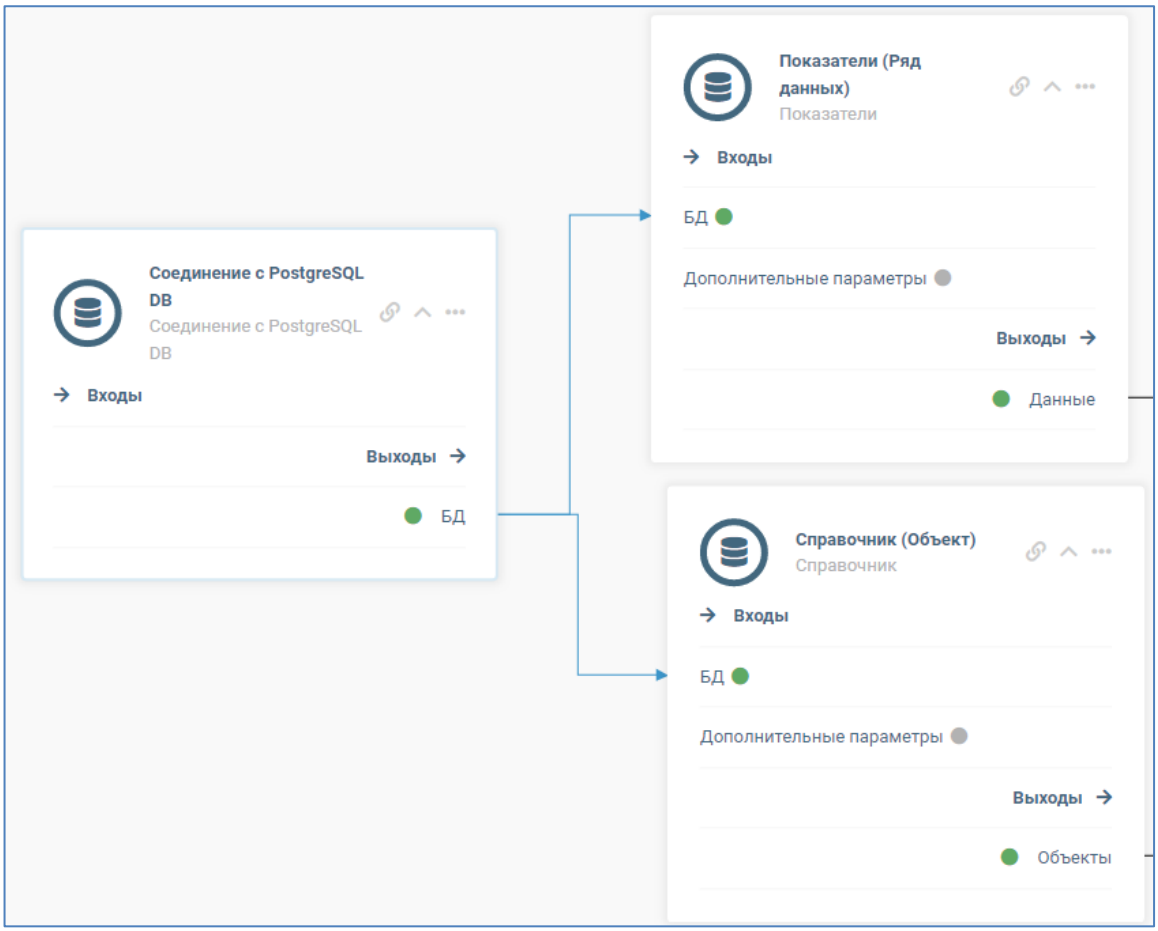


Рисунок 467 – Пример блока «Соединение с PostgreSQL DB»

7.2.6 Предпросмотр результатов настройки блока

Предпросмотр – запуск на расчет цепочки блоков до выбранного блока с настройками блока, из которого производится предпросмотр. Расчеты этой цепочки не сохраняются, а только выведут либо визуализатор, либо данные результата блока

Предпросмотр вызывается из настроек блока.

1. Открыть настройки блока.

2. Выбрать нужные настройки (с этими настройками будет произведен расчет предпросмотра).

3. Нажать на переключатель предпросмотра:

4. Откроется окно предпросмотра.

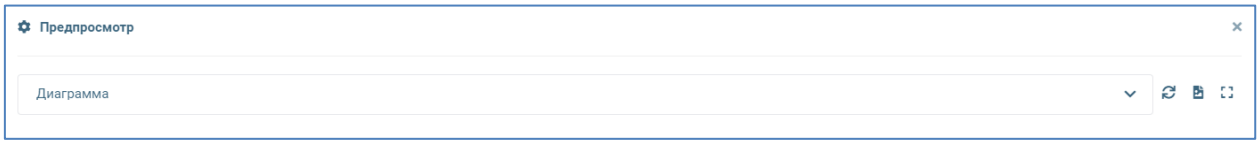


Рисунок 468 – Окно предпросмотра

5. В выпадающем списке выбрать нужный блок или визуализатор.

5.1 Если блок находится после блока «Итератор», то можно выбрать в выпадающем списке номер итерации.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

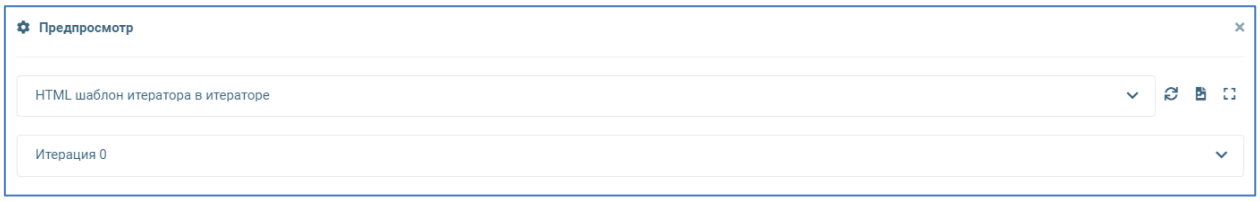


Рисунок 469 – Пример выбора номера итерации

6. Запустится перерасчет для выбранного блока или визуализатора.

7. Отобразятся данные выхода блока или данные визуализатора.

Если надо пересчитать блока в предпросмотре – можно нажать кнопку «Обновить» .

Если результат надо сохранить – можно нажать кнопку «Выгрузить во внешний файл» .

Так же можно посмотреть результат в полном окне – можно нажать кнопку «Открыть в увеличенном размере» .

7.2.7 Переменные входов/выходов

Переменные входов/выходов - это дочерние входы или выходы блока, которые можно добавить при редактировании блока.

7.2.7.1 Переменные входов

Для создания переменных необходимо навести курсор на нужный вход и во всплывающем меню выбрать пункт "Редактирование переменных":

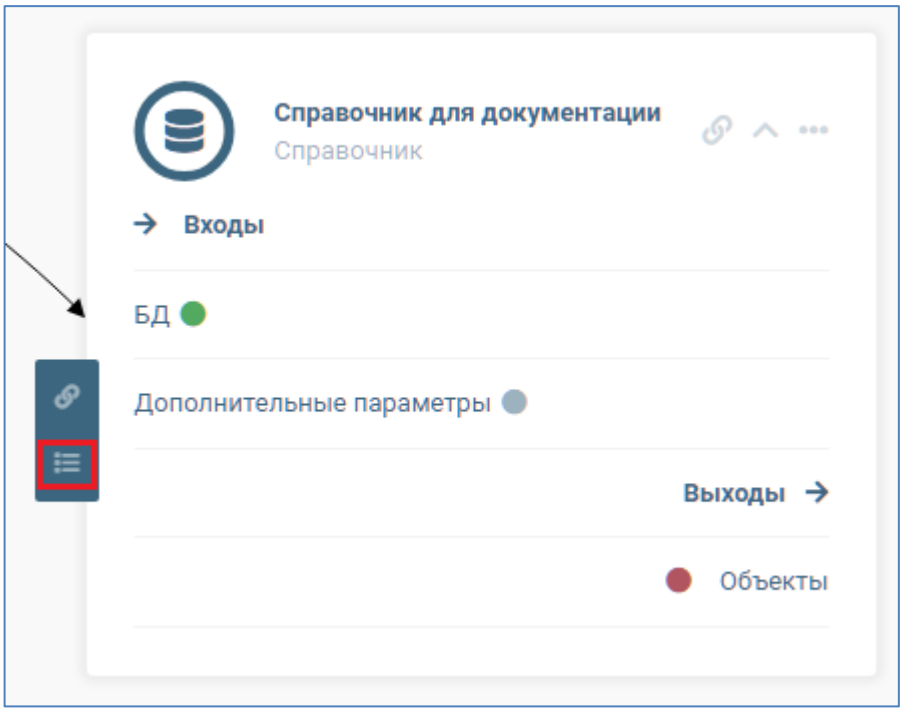


Рисунок 470 – Пример кнопки «Редактирование переменных»

После нажатия откроется окно редактирования переменных входа:

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

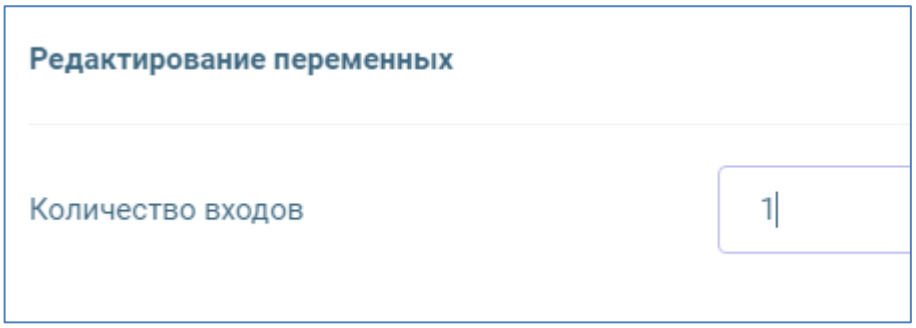


Рисунок 471 – Пример задания количества входов

В данном примере создано два переменных входа блока.

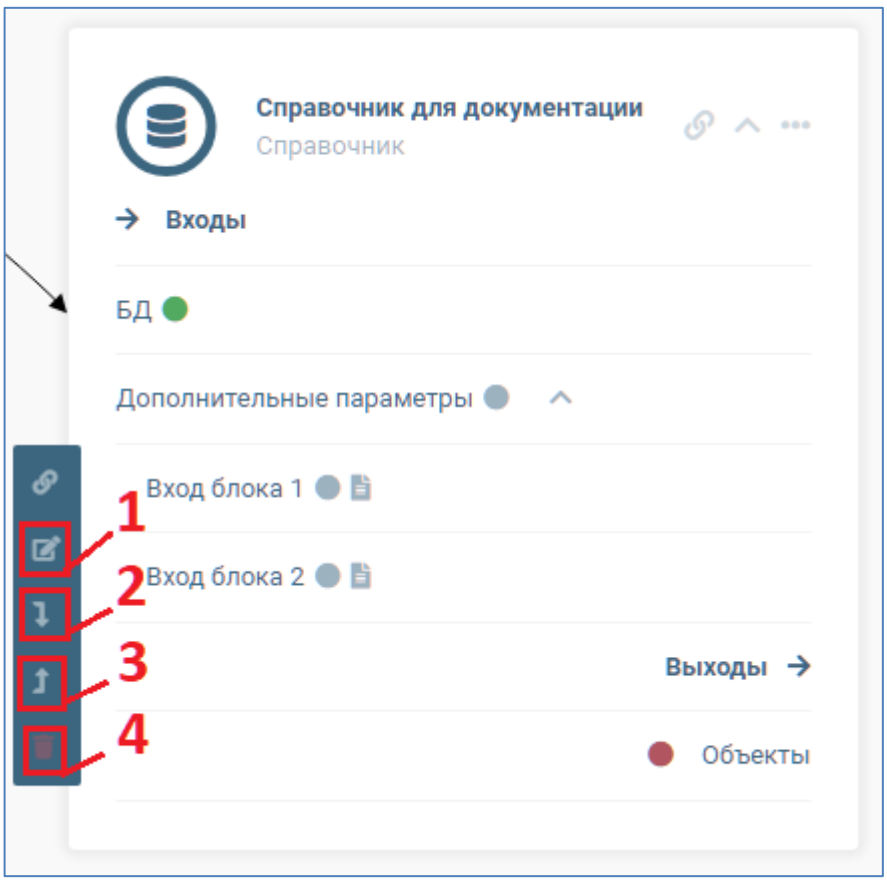
При повторном открытии пункта "Редактирование переменных" можно изменить число добавленных входов.

Если новое число больше текущего - добавятся дополнительные входы.

Если новое число меньше - лишние входы будут удалены.

Редактирование переменных входов

При наведении курсора на добавленную переменную появляется всплывающее меню со следующими параметрами:



© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Рисунок 472 – Пример блока с несколькими переменными входов

Редактирование входа. Можно изменить наименование входа:

Переместить вход ниже. Изменить порядок входов.

Переместить вход выше. Изменить порядок входов.

Удалить вход.

Переменные выходов

Переменные выходов - дочерний выход блока, отфильтрованный по выбранным первичным ключам объекта.

Для настройки необходимо навести курсор на выход блока и во всплывающем меню выбрать пункт "Добавить переменную":

Настройку переменных выходов необходимо производить на уже рассчитанном блоке.

Если блок был расчитан с ошибкой, то настройка переменных будет недоступна.

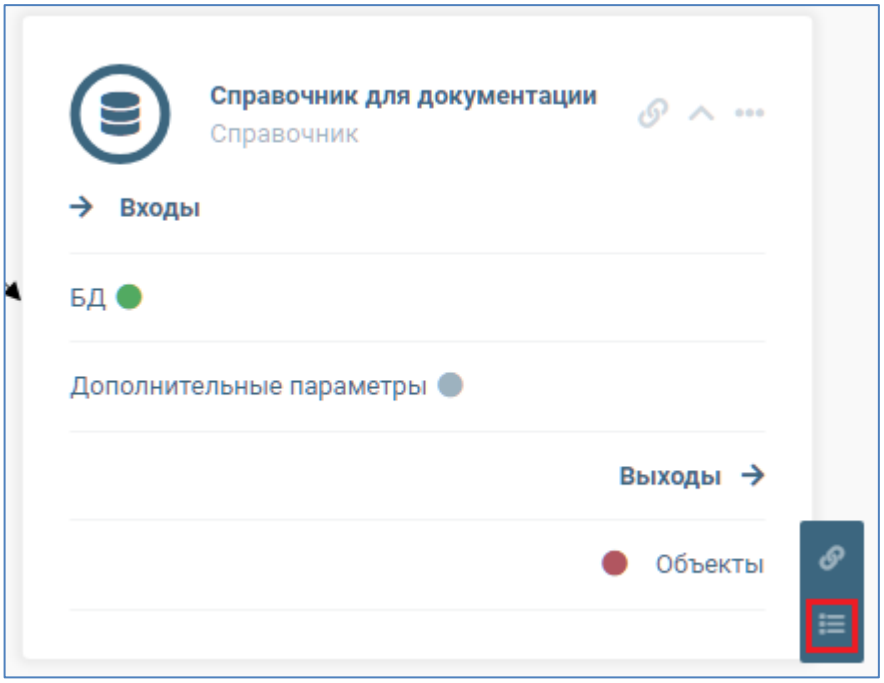


Рисунок 473 – Пример кнопки «Рассчитать задачу»

После нажатия открывается окно с добавлением переменных выхода:

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

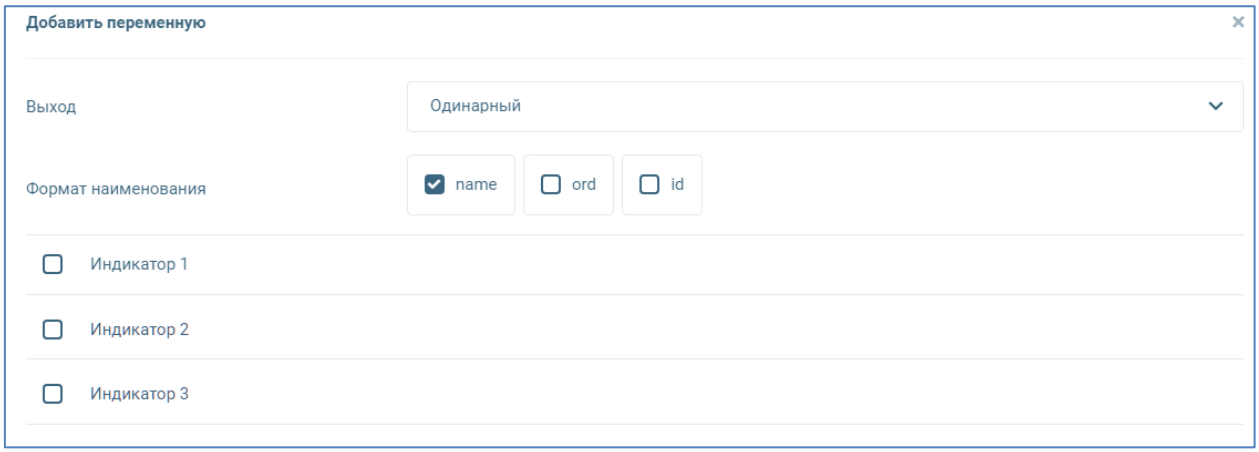


Рисунок 474 – Пример добавления переменной входа

Выход - формат выхода данных. Выпадающее меню с двумя значениями:

Одинарный - каждый выбранный элемент создаст отдельный выход блока;

Массив - все выбранные элементы объединятся в один массив. Для массива так же можно задать наименование выхода.

Формат наименования - выбор первичных ключей и/или атрибутов объекта, которые будут отвечать за формирование наименования выхода.

Путем перетаскивания ключей (drag-n-drop) можно изменять порядок элементов.

После выбора одной или нескольких переменных и нажатия на кнопку "Сохранить", у выбранного блока создадутся дополнительные выходы:

За одно редактирование переменных возможно создание нескольких одинарных выходов или одного массива.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

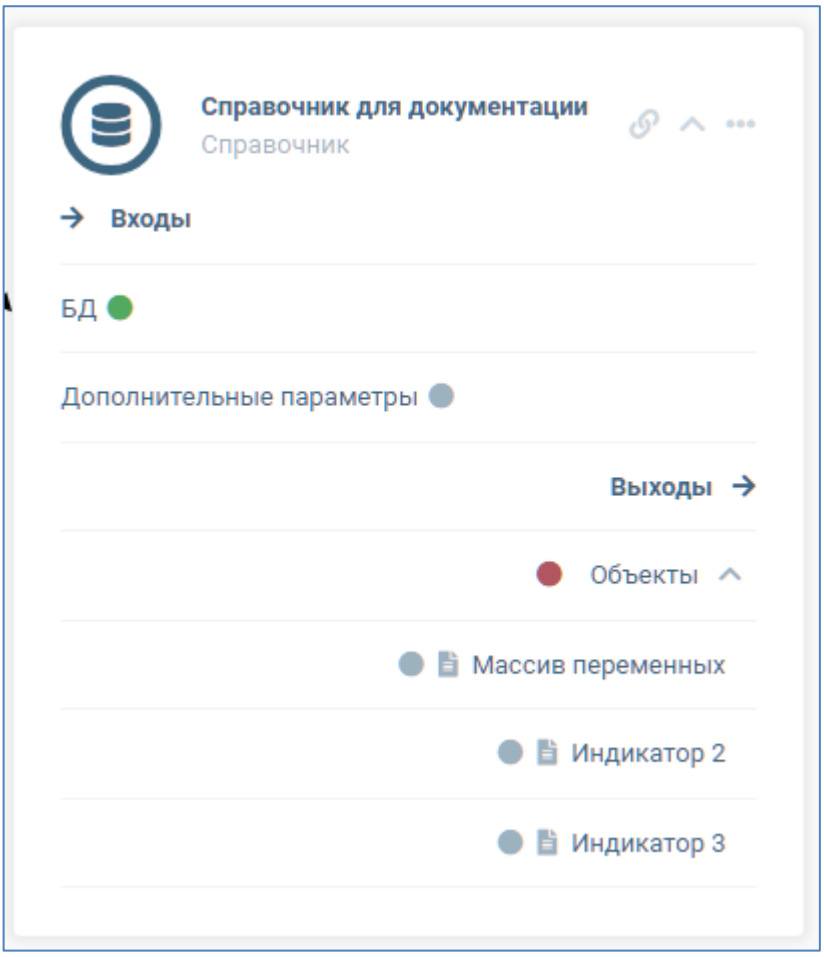


Рисунок 475 – Пример блока с тремя добавленными выходами: одним массивом и двумя одинарными выходами

Редактирование переменных выхода

При наведении курсора на добавленную переменную появляется всплывающее окно с параметрами, аналогичными параметрам при редактировании входа:

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

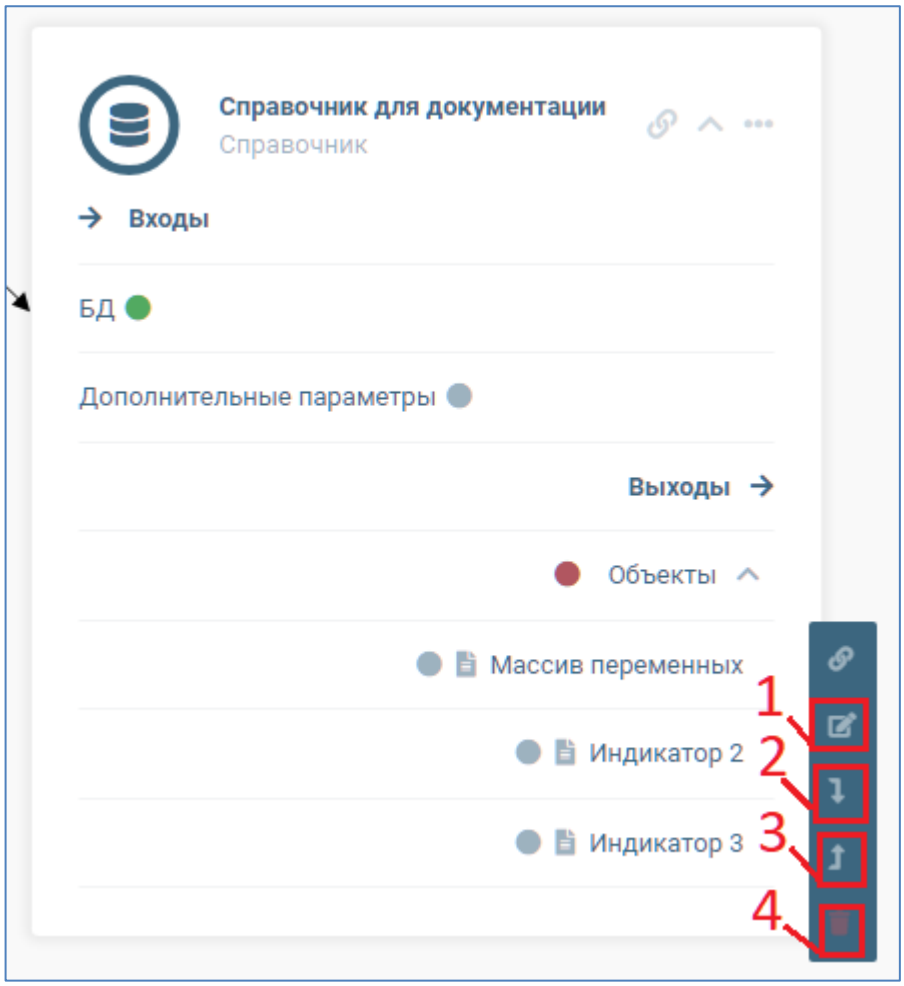


Рисунок 476 – Пример редактирования выхода

. Можно изменить наименование выхода (для массива) и формат наименования элементов:

В случае, если часть атрибутов, использовавшихся в отображении наименования, будет недоступна, формат наименования сбросится к первому доступному атрибуту.

Если блок расчитан с ошибкой, то редактирование будет недоступно.

Переместить выход ниже. Изменить порядок выходов.

Переместить выход выше. Изменить порядок выходов.

Удалить выход.

7.3 Работа со связями

7.3.1 Обработка данных в связях между блоками

Обработчик данных позволяет модифицировать данные непосредственно при передаче их между блоками.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

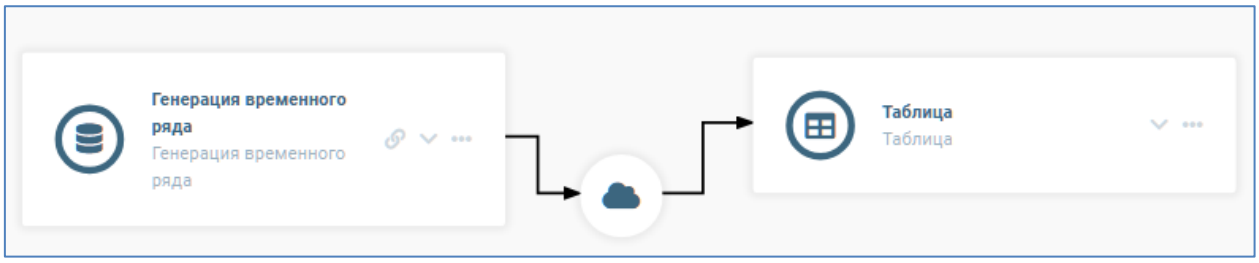


Рисунок 477 – Пример обработчика на связи между блоками

7.3.1.1 Создание

Для создания обработчика необходимо нажать левой кнопкой мыши на любую связь и выбрать пункт "Создать обработчик":

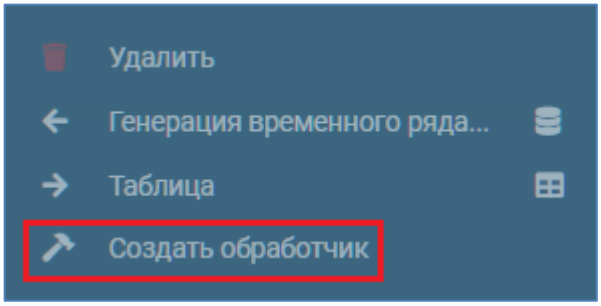


Рисунок 478 – Пример кнопки «Создать обработчик»

После этого будет создан обработчик данных:

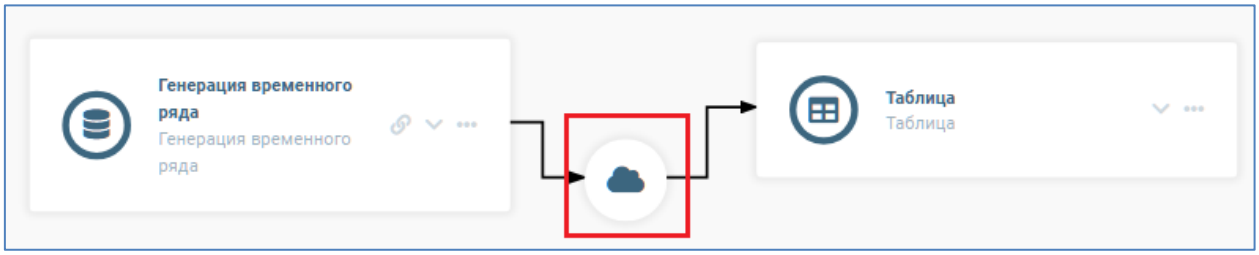


Рисунок 479 – Пример добавления обработчика на связь

Вход (наименование, идентификатор и тип) будет взят у блока, из которого исходит связь (в примере из блока "Генерация временного ряда").

Выход будет взят у блока, к которому идет связь (на рисунке - блок "Таблица).

7.3.1.2 Работа с обработчиком

Для работы с обработчиком можно дважды щелкнуть по нему левой кнопкой мыши или один раз нажать правой кнопкой и выбрать пункт "Редактировать код":

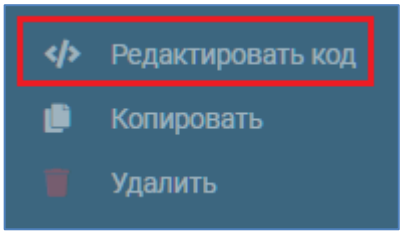


Рисунок 480 – Пример кнопки «Редактировать код»

Редактор обработчика по фунционалу аналогичен блоку "Выражение". См. раздел 6.5.1.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Для получения входных данных с помощью конструкции "Вход" необходимо указывать наименование входа, из которого исходит связь.

7.3.1.3 Удаление обработчика

Для удаления обработчика необходимо нажать левой кнопкой мыши на связь и выбрать пункт "Удалить обработчик":

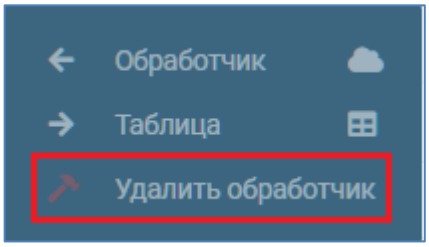


Рисунок 481 – Пример кнопки «Удалить обработчик»

Связь между блоков, при этом, будет сохранена.

Альтернативный способ удаления - нажать по нему правой кнопкой мыши и выбрать пункт "Удалить":

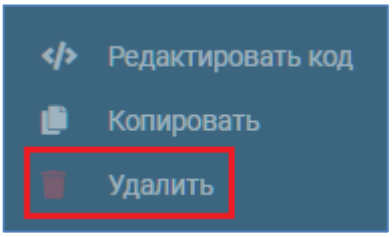


Рисунок 482 – Пример кнопки «Удалить» при щелчке на связь

В этом случае будет удален не только обработчик, но и связь между блоками.

7.4 Расчет графа

Расчет графа – это процесс последовательного исполнения всех цепочек блоков, находящихся на графе.

Расчет цепочек блоков – последовательное выполнение каждого находящегося в цепочке блока.

Цепочка блоков – это один или несколько блоков, последовательно соединённых друг с другом.

Расчет блока – получение результата, в зависимости от входных данных и настройки блока.

Запуск расчета графа

Запуск расчета графа осуществляется через меню управления графом. Для этого необходимо перейти в граф задачи и нажать кнопку «Расчет графа».

Рисунок 483 – Пример кнопки «Расчет графа»

Так же граф может быть запущен на пересчет из пресета нажатием кнопки «Рассчитать задачу» в меню пресета.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

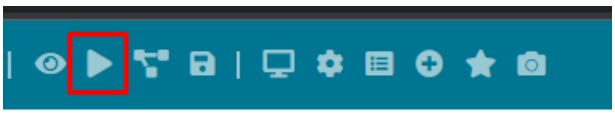


Рисунок 484 – Пример кнопки «Рассчитать задачу»

Помимо запуска всего графа на расчет, возможно запустить расчет одного блока или цепочки блоков.

Для этого в локальном меню блока необходимо выбрать соответствующую кнопку.

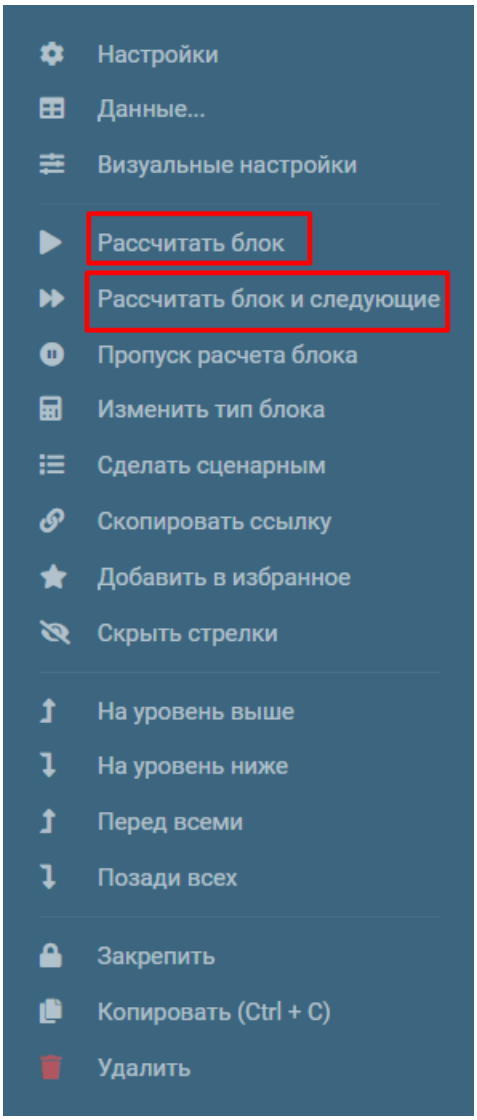


Рисунок 485 – Пример пунктов меню «Расчет блока» и «Рассчитать блок и следующие»

Частичный расчёт графа вызываются событиями на пресете (см. п. 7.4.3).

Описание процесса расчета графа

Расчет графа состоит из последовательного расчета всех блоков на графе. Блоки могут образовывать цепочки произвольной длины.

В связи с тем, что для расчета блоков требуются результаты предыдущих блоков, процесс протекает последовательно.

После запуска расчета блока появляется окно лога, отображающее течение процесса расчета.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

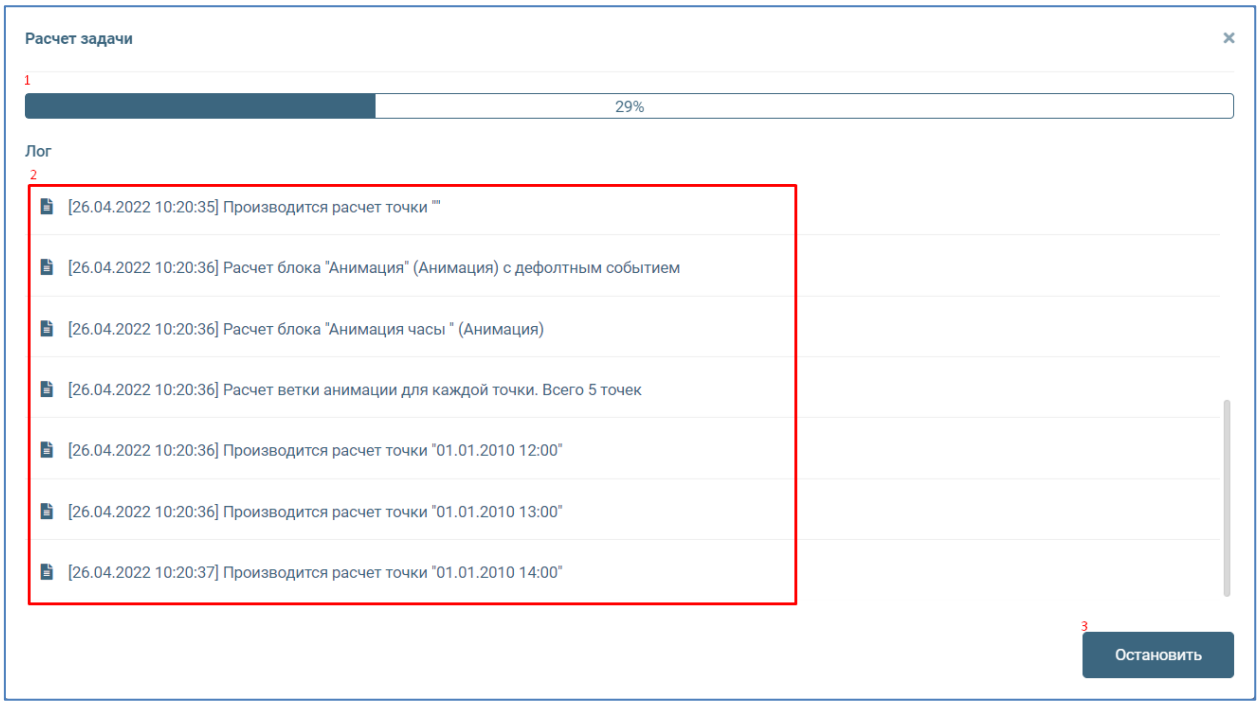


Рисунок 486 – Пример окна лога

Основные элементы окна лога:

1. Прогрессбар отображающий общий процесс выполнения расчета графа;
2. Лог расчета отображающий время старта расчета элементов графа, какие блоки графа были рассчитаны, а также отображает информацию об успешности расчета блока и графа в целом. Если произошла ошибка, к этому блоку можно перейти, кликнув ЛКМ по строке с ошибкой;
3. Кнопка, позволяющая остановить расчет графа, либо закрыть окно лога, если процесс завершён.

7.4.1 Описание процесса расчета графа в режиме «Отладки графа»

Расчет графа запущенный в режиме «Отладки графа» визуально отличается от описанного выше.

Внешний вид меню графа изменяется при включении режима «Отладки графа».

Рисунок 487 – Меню графа в режиме «Отладки графа»

Запуск осуществляется аналогично обычному расчету графа.

При запуске расчета, в этом случае, не появляется окно лога автоматически.

Так же на графе блоки подкрашиваются цветом, в зависимости от их состояния.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

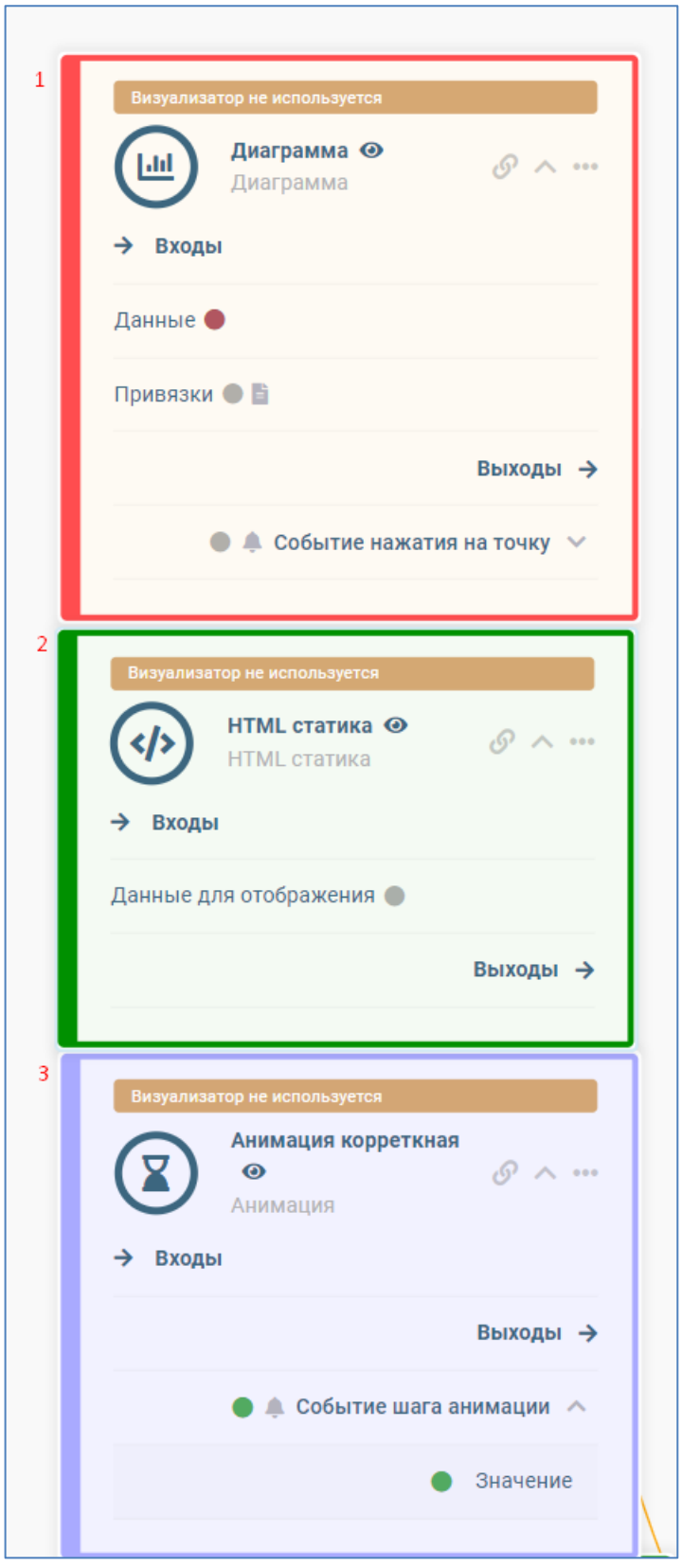


Рисунок 488 – Пример отображения блоков в зависимости от их состояния

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

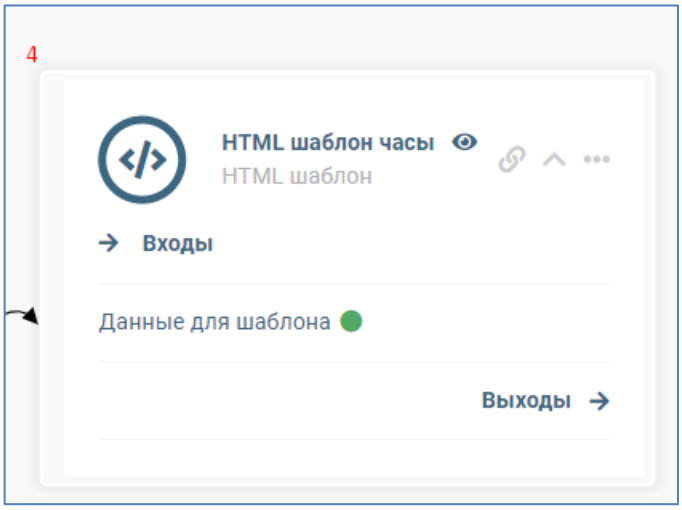


Рисунок 489 – Состояние блока «Блок не рассчитывался»

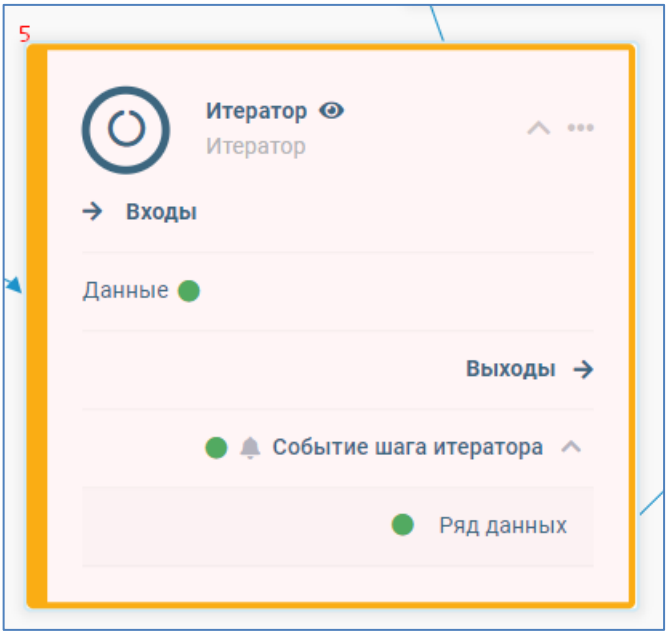


Рисунок 490 – Состояние блока «Рассчитан с предупреждением»

Цветовая индикация блоков при расчете на графе.

1. Блок рассчитан с ошибкой;
2. Блок рассчитан без ошибок;
3. Блок рассчитывается;
4. Блок еще не рассчитывался;
5. Блок рассчитан с предупреждением.

Во время расчета можно вызвать лог, кликнув по кнопке «Лог расчёта» в меню графа.

Рисунок 491 – Пример кнопки «Лог расчёта» в меню графа

При расчете графа в режиме отладки окно лога может быть закрыто без остановки расчета.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Для этого необходимо нажать по крестику в левом верхнем углу окна лога.

Рисунок 492 – Пример закрытия окна «Лог расчёта» в меню графа

7.4.2 Описание процесса расчета графа, запущенного из пресета

При запуске расчета графа из режима редактирования пресета, появляется стандартное окно лога.

Если расчет запущен из режима просмотра пресета, лог расчета будет свёрнут.

Для того чтобы развернуть лог, необходимо нажать на значок .

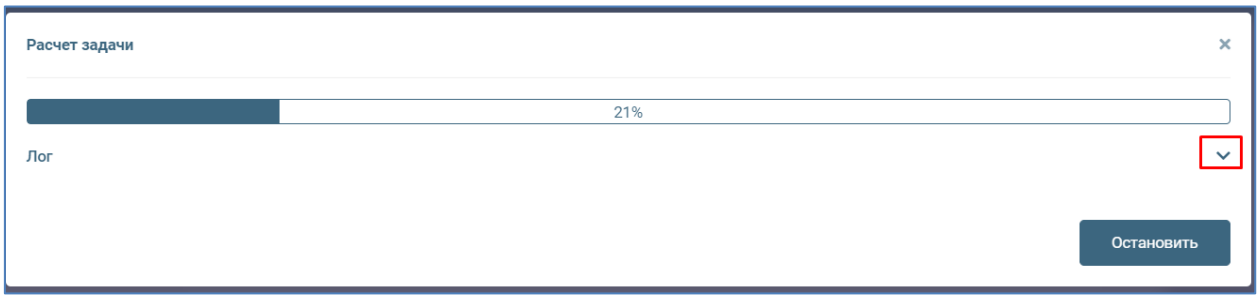


Рисунок 493 – Пример открытия лога

7.4.2.1 Остановка расчета графа

Расчет графа может быть остановлен путём нажатия кнопки «Остановить» в окне лога расчета. При остановке расчета графа, будет окончен расчет блока, который считался на момент нажатия кнопки.

В логе отображается остановка процесса пользователем.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

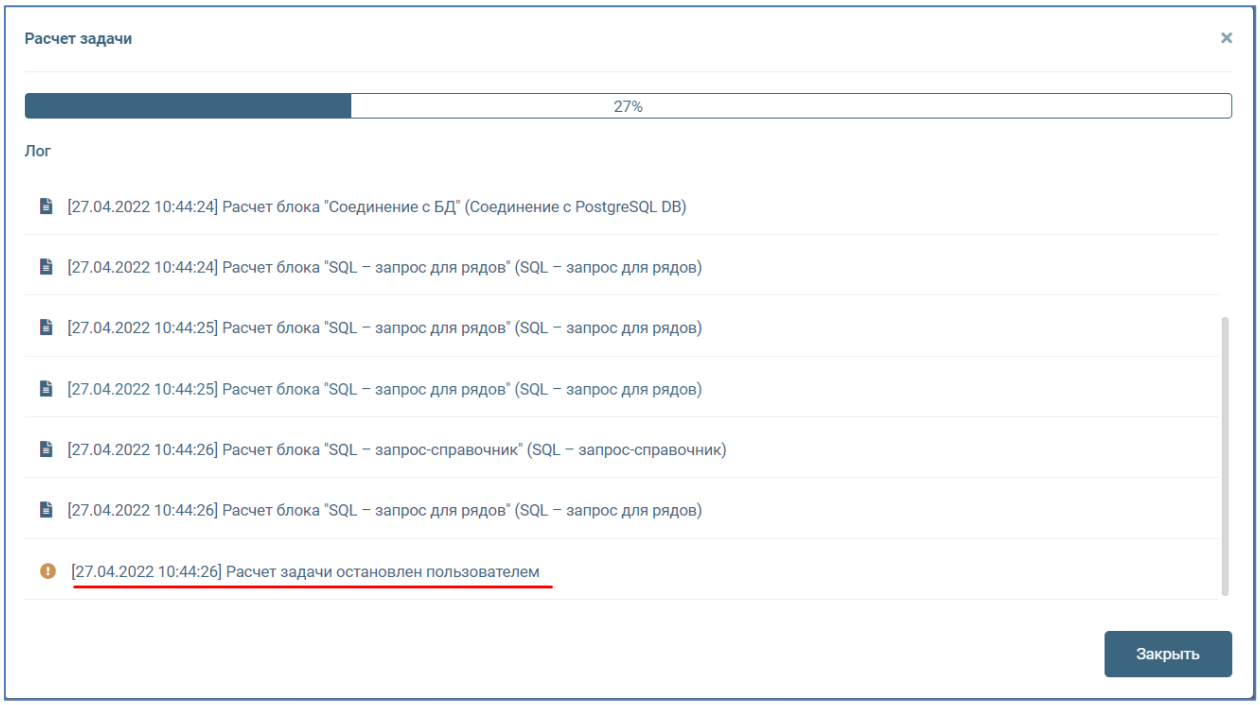


Рисунок 494 – Пример просмотра лога

7.4.2.2 Ограничения на размеры блоков

1. Объем данных всех выходов блока должна не превышать 150 Mb. В случае превышения блоком этого размера, блок рассчитан не будет и выведется сообщение об ошибке в логе.
2. Рекомендуемы объем для визуализаторов – 20 Mb. Если визуализатор превышает этот размер, в логе будет предупреждение об этом.

7.4.3 События в блоках

Есть 2 вида событий:

- вызываемое событие
- дефолтное событие

7.4.3.1 Вызываемое событие

Данное событие вызывается пользователем.

Данные для такого расчета выбираются пользователем и передаются на расчет события.

7.4.3.2 Дефолтное событие

Данное событие вызывается системой автоматически при расчете графа.

Данные, с которыми рассчитывается дефолтное событие, задаются в настройках блока у которого есть событийные выходы.

7.4.3.3 Особенности

Во время сбора списка визуализаторов (при открытии пресета) делается проверка, находится ли визуализатор в ветке по событию у которого стоит флаг «Не получать визуализаторы при открытии пресета». Если такие визуализаторы есть, они помечаются флагом «ожидания при открытии» и передаются на пресет.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

7.4.4 Механизм расчета

7.4.4.1 Терминология

Дефолтный расчет - расчет, при котором результаты не привязаны к пользователю, сохраняются в t_calc_data. Результаты такого расчета будем называть дефолтными

Событийный расчет - расчет, при котором результаты привязаны к пользователю, сохраняются в t_calc_data_event. Результаты такого расчета будем называть событийными

Событие - выход блока-визуализатора, в наименовании которого есть слово "Событие" и стрелка, которая выходит из него, оранжевого цвета (по умолчанию)

Цепочка блоков - несколько блоков, которые связаны между собой

Механизм writeback - расчет цепочки, который производит поиск источника данных и расчет от этого источника до зависимых визуализаторов

7.4.4.2 Принципы механизма расчета

- Результаты расчета всегда сохраняются как дефолтные при полном расчете графа
- Если перед блоком нет событий (непосредственно перед или где-либо в цепочке перед ним), то такой блок считается как дефолтный при расчете блока
- Если перед блоком есть события (непосредственно перед или где-либо в цепочке перед ним), то такой блок считается как событийный при расчете блока
- Если перед цепочкой блоков (непосредственно перед или где-либо в цепочке перед ней) нет событий и внутри этой цепочки так же нет событий, то такая цепочка считается как дефолтная при расчете "Рассчитать блок и следующие"
- Если перед цепочкой блоков (непосредственно перед или где-либо в цепочке перед ней) есть события и/или внутри этой цепочки есть события, то такая цепочка считается как событийная при расчете "Рассчитать блок и следующие"
- При открытии пресета (обновлении страницы пресета) происходит очистка событийных результатов расчета
- Действия пользователя на пресете, которые вызывают расчет, сохраняются как событийные (кроме сохранения данных - механизма writeback)
- Действия пользователя на пресете, которые используют механизм writeback, сохраняются как дефолтные (т.к. меняют данные в БД) и как событийные (чтобы текущая отметка пользователя осталась на пресете)

7.4.4.3 Отладка событий

Режим отладки графа позволяет просматривать результаты блоков, рассчитанных в т.ч. на событийных данных

Пример простого алгоритма отладки:

Открыть пресет с визуализатором, в котором есть событие

- Вызвать событие в визуализаторе (например, изменить отметку в таблице)
- Перейти в граф
- Включить режим отладки
- Выделить данный визуализатор

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

- Открыть выход события в панели свойств блока - выход будет соответствовать событию, вызванному в пресете

7.5 Режим отладки

Отладка – этап разработки, на котором обнаруживают, локализуют и устраняют ошибки. Для корректной работы режима отладки необходимо установить в настройках задачи режим «Разработка графа».

Рисунок 495 – Меню графа

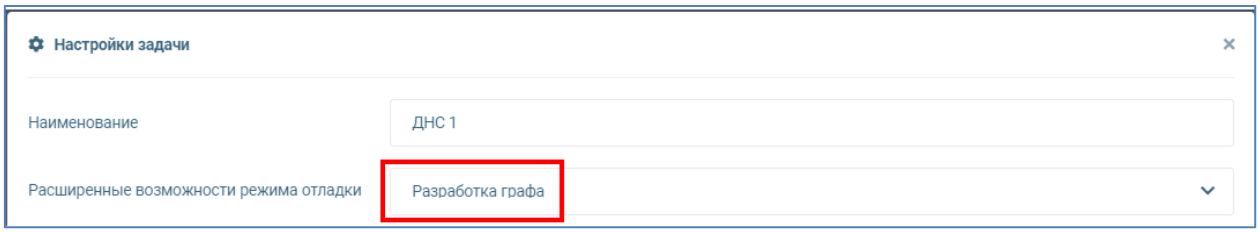
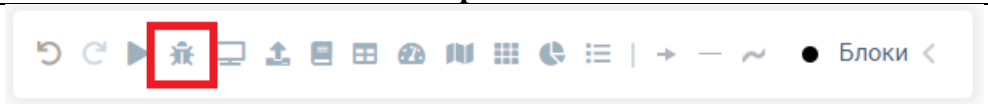
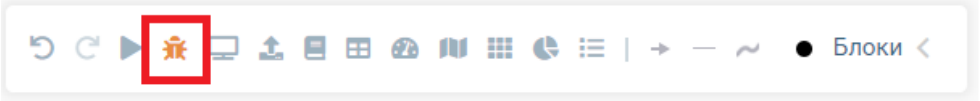
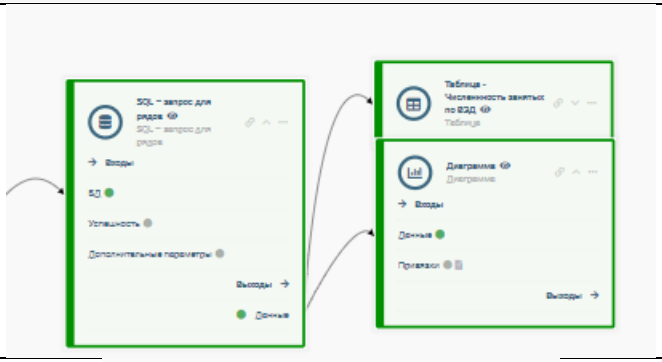
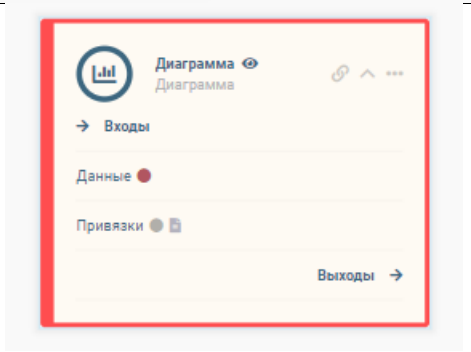
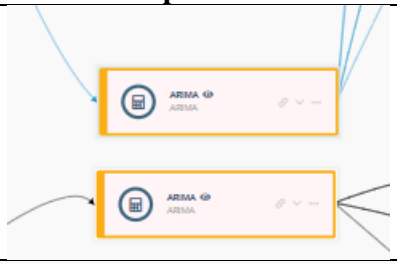
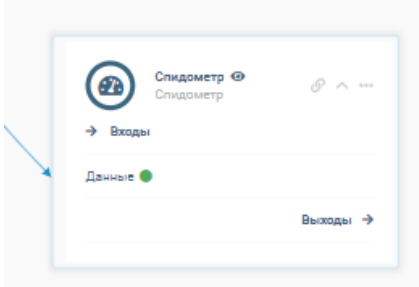


Рисунок 496 –Установка режима «Разработка графа»

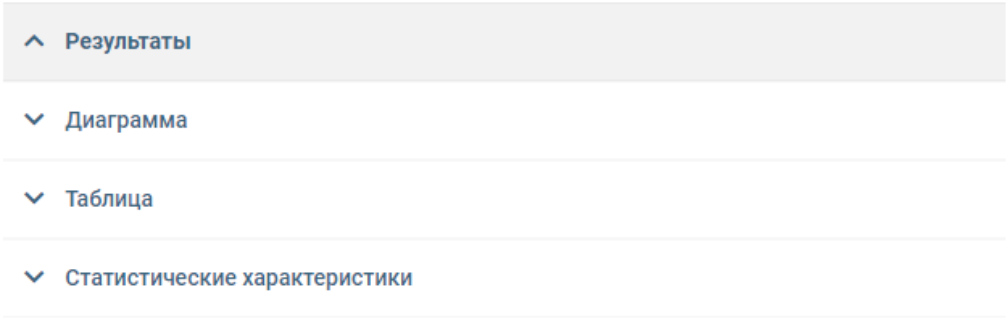
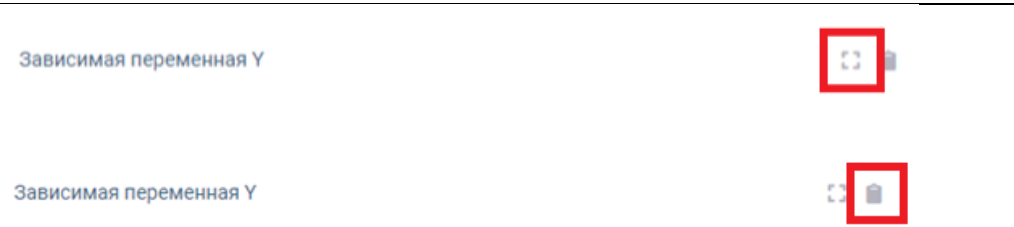
© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

№	Действие	Скриншот
1	Чтобы перейти в раздел отладки, необходимо нажать на значок «жучка» в верхней части рабочей области	
2	После нажатия на кнопку сам «жучок» и все блоки графа изменят цвет	
2.1	**Зеленый** цвет блока говорит о том, что блок рассчитан верно и не имеет ошибок	
2.2	**Красный** – наличие ошибки при последнем расчете блока. Вероятно, в описании блока или в данных есть ошибки. Блок не может быть рассчитан *Возможные причины ошибки:* *не протянута связь в обязательный вход блока (при этом вход будет выделен красным кружком)* *отсутствуют обязательные входные данные (связь протянута, но нет данных на выходе блока-источника)* *входные данные некорректны* *ошибки в настройках блока* *для блоков Python синтаксическая ошибка в коде блока или ошибка при выполнении кода*	

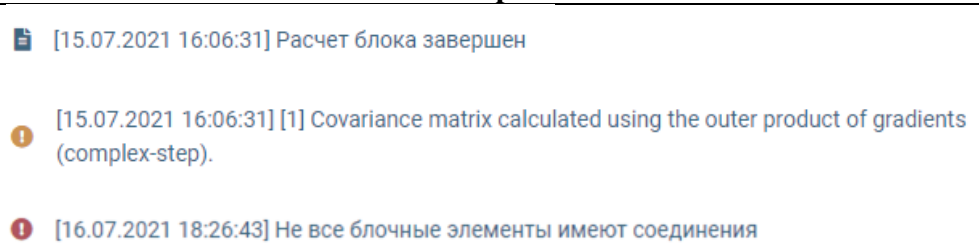
© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

№	Действие	Скриншот
2.3	**Желтый** – есть некритические замечания к описанию блока, но сам блок будет рассчитан верно *(Например, используем блок Arima для построения модели авторегрессии, но не строим прогноз, то есть используем не все возможности данного блока)*	
2.4	**Серый –** блок не был рассчитан после создания или внесения изменений в настройки. Так же серыми блоками становятся блоки, которые идут после блока, в котором изменяли настройки (или изменяли данные в мастере «Данные»)	
3.	В режиме отладки есть возможность просмотреть свойства блока для этого необходимо нажать ЛКМ на нужный блок. В левом верхнем углу рабочей области отроется меню «Свойства блока». Для того чтобы просмотреть каждое свойство отдельно, необходимо нажать ЛКМ на название подблока (Результаты, Входы, Выходы, Лог).	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

№	Действие	Скриншот
3.1	В разделе «Результаты» можно увидеть результаты выполненного блока Для **блоков моделирования** результаты будут представлены в блоках - «*Диаграмма*»: линейный график фактических, модельных и прогнозных значений, остатки модели - «*Таблица*»: содержит фактические, модельные и прогнозные значения, остатки модели - «*Статистические характеристики*»: подробное описание параметров модели и статистики Для **блоков получения данных** будет выведен фактически выполняемый запрос к данным (sql-запрос, скрипт после применения шаблонизатора) Для **блоков визуализаторов** отображаются результаты выполнения блока, т.е. непосредственно визуализатор (график, диаграмма, таблица) В режиме отладки для отображаемых визуализаторов отключены события (в отличие от отображения на пресетах)	
3.2	В разделах «Входы» и «Выходы» представлена информация о входных данных в выбранный блок и выходных данных из него *(У каждого блока свои входные и выходные значения)* Для того чтобы раскрыть входные или выходные значения, необходимо нажать ЛКМ на значок «Открыть в окне» с правой стороны подблока. Для того чтобы скопировать входные или выходные значения, необходимо нажать ЛКМ на значок «Копировать» с правой стороны подблока.	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

№	Действие	Скриншот
3.3	В разделе «Лог» есть возможность увидеть результаты компиляции и ошибки блока Если расчет блока завершен верно, то будет выведено сообщение с датой и временем расчета блока Если при расчете блока были найдены ошибки, то будет выведено сообщение с датой, временем расчета блока и текстом ошибки. Причем желтый цвет ошибки – блок будет рассчитан до конца; красный – блок не будет рассчитан.	
3.4	В разделе «Производительность» есть возможность увидеть время выполнения частей блока, такие как: • Подготовка входных данных для расчета блока • Расчет самого блока • Сохранение в бд результатов расчета блока • Итоговое время выполнения Время отображается в миллисекундах	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

7.6 Редактирование и отладка блока Python в браузере

Внутри блока Python пользователю доступен редактор кода с подсветкой синтаксиса и автодополнением ключевых конструкций для обращения к контексту выполнения. Чтобы открыть редактор кода блока, необходимо кликнуть по блоку на графе.

В редакторе предусмотрено редактирование нескольких блоков графа одновременно. В списке слева отображены все блоки Python на текущем графе. Кроме блоков, доступных на графе, возможно редактирование пользовательских скриптов, доступных на графе. Подробнее в разделе 7.5 Скрипты «Руководства Администратора».

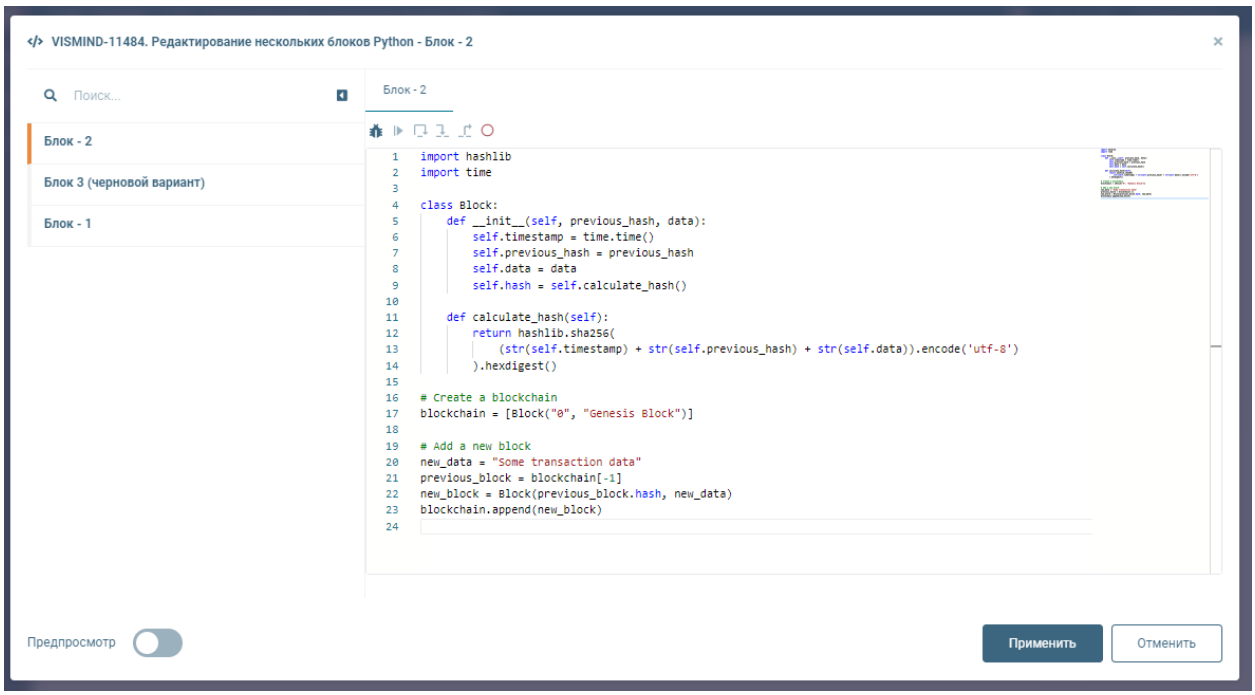


Рисунок 497 – Пример редактора кода в браузере

7.6.1 Редактирование кода

7.6.1.1 Сохранение изменений

Если в коде есть несохраненные изменения, вкладка с кодом будет помечена звездочкой.

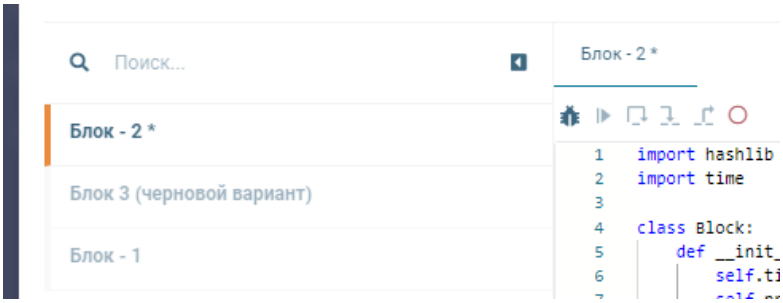


Рисунок 498 – Пример редактора с несохраненными изменениями

Чтобы сохранить изменения, необходимо нажать кнопку "Применить" в правом нижнем углу. При попытке закрытия редактора кода выводится предупреждение, которое позволяет не потерять измененный код при закрытии окна.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

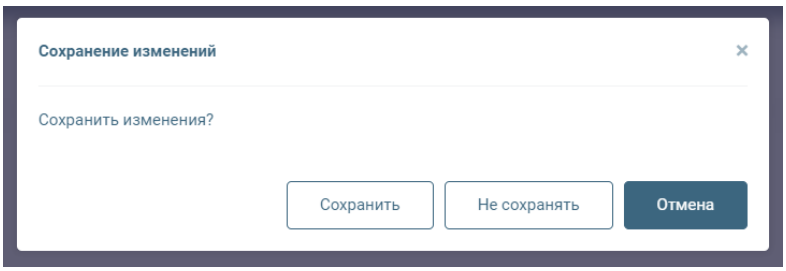


Рисунок 499 – Диалог сохранения изменений

7.6.1.2 Черновики. Хранение несохраненных изменений кода

В блоке предусмотрено сохранение пользовательских изменений в виде черновиков. Черновики сохраняются автоматически в процессе редактирования кода пользователем. Черновики индивидуальны для каждой учетной записи пользователя.

7.6.1.3 Удаление черновиков при изменении блока другим пользователем

Если при повторном открытии блока для продолжения редактирования произойдет ситуация, что за это время кто-то другой уже отредактировал этот же код, черновики, созданные до изменения блоки, будут удалены.

Чтобы удалить черновик, необходимо нажать кнопку «Обновить» в верхнем левом углу.

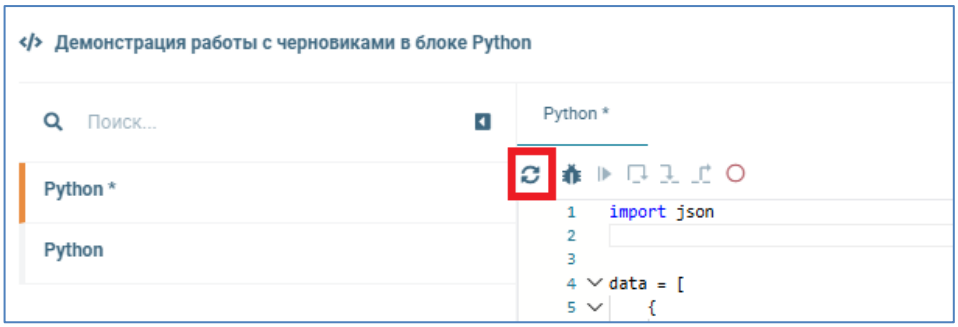


Рисунок 500 – Удаление черновика

7.6.1.4 Редактирование блоков несколькими пользователями

Когда пользователь редактирует код выбранного блока, доступ к нему блокируется для остальных пользователей, находящихся на графе. Блоки, которые недоступны для редактирования в текущий момент, выделены в общем списке.

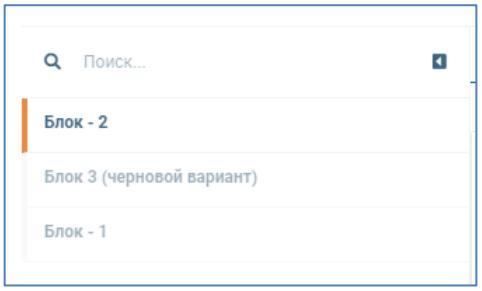


Рисунок 501 – Пример списка блоков, доступных для редактирования

Чтобы разблокировать блок для остальных, пользователю необходимо закрыть вкладку с кодом блока, либо выйти из режима редактирования.

7.6.2 Отладка кода

7.6.2.1 Назначение отладчика

Отладчик позволяет пошагово выполнять пользовательский код в блоке Python. Отладчик может пригодиться, когда код из блока Python проблематично переносить в среду

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

разработки на локальном компьютере: например, когда из кода выполняются обращения к входам блока, либо когда в коде выполняются обращения к базам данных или веб-сервисам, которые недоступны напрямую с пользовательского компьютера

Отладчик предоставляет базовый функционал:

- Точки останова;
- Построчное перемещение по коду;
- Просмотр переменных;
- Просмотр стека вызовов и перемещение по нему.

Доступны дополнительные функции: условные точки останова и выполнение произвольных однострочных выражений.

Рисунок 502 – Отладка кода в браузере

7.6.2.2 Базовые термины

Сессия отладки - разовый запуск кода на отладку

- Для одного блока Python доступен единовременный запуск лишь одной сессии отладки.
- Сессии имеют ограничение времени на бездействие пользователя. Сессия существует 24 часа с момента последнего действия пользователя, а затем удаляется.

Точка останова - указание отладчику остановиться на конкретной строке кода. При переходе на точку останова отладчик переходит в режим ожидания команд пользователя.

Шаг с заходом / шаг с выходом / шаг с обходом - способы перемещения по коду.

Стек вызовов - цепочка последовательных вызовов функции в коде пользователя.

7.6.2.3 Управление отладкой

7.6.2.3.1 Запуск отладчика

Чтобы запустить отладчик, нажмите кнопку "Начало отладки".

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

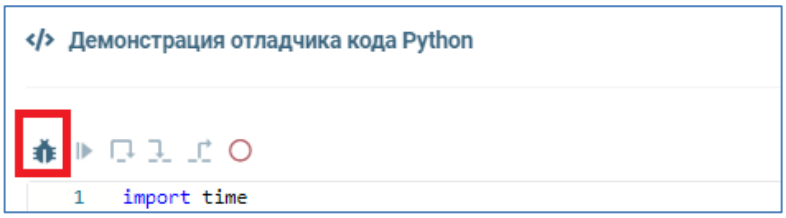


Рисунок 503 – Кнопка запуска отладки

После запуска приложение выведет сообщение запуске отладки.

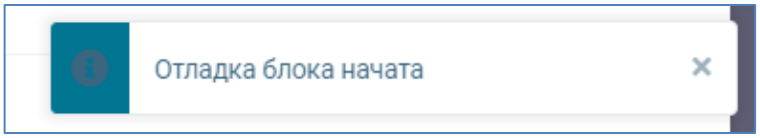


Рисунок 504 – Сообщение о начале отладки блока

При запуске отладки приложение инициализирует отладчик, и начинается выполнение кода до ближайшей точки останова. Когда отладчик дойдёт до точки останова и перейдёт в режим ожидания команд, подсветится соответствующая строка кода.

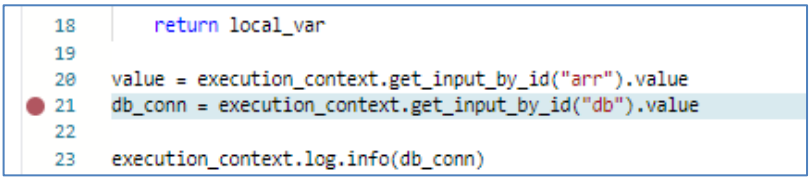


Рисунок 505 – Пример точки останова

7.6.2.3.2 Точки останова

Чтобы установить точку останова, кликните по номеру выбранной строки.

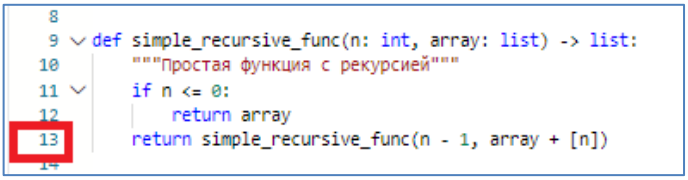


Рисунок 506 – Установка точки останова

Чтобы удалить точку останова, повторно кликните по номеру выбранной строки.

Редактирование точек останова доступно и при запущенной сессии отладки, но только на точках останова. При выполнении кода редактирование невозможно.

Отладчик не учитывает точки останова в пустых строках кода.

7.6.2.3.3 Отключение точек останова

Точки останова отключаются нажатием кнопки "Выключить точки останова".

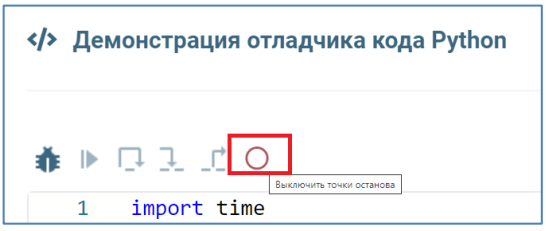


Рисунок 507 – Кнопка отключения точек останова

7.6.2.3.4 Условные точки останова

Кроме стандартных точек останова, поддерживаются условные точки останова. В них задаётся логическое выражение на языке Python. Отладчик останавливается в условной

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

точке, если логическое выражение истинно. Код не будет остановлен, если выражение не истинно, либо в нем содержится синтаксическая ошибка или исключение. Чтобы задать условную точку останова, нужно кликнуть правой кнопкой мыши по номеру выбранной строки. Откроется окно для задания условного выражения.

Рисунок 508 – Задание условия для точки останова

7.6.2.3.5 Перемещение по коду

Находясь в точке останова, отладчик поддерживает следующие способы перемещения по коду:

- Шаг с заходом выполняет очередную инструкцию, а затем переходит в режим ожидания. Если шаг с заходом выполняется на вызове функции, то отладчик переходит в функцию и останавливается в её начале.
- Шаг с обходом аналогичен шагу с заходом, но в отличие от него не останавливается при переходе в новую функцию. Вместо этого отладчик код внутри функции и переходит на строку, следующую за точкой останова.
- Шаг с выходом выполняет весь оставшийся код функции и затем возвращает управление разработчику.

При выполнении шага с заходом отладчик не переходит в код импортируемых модулей.

7.6.2.3.6 Перемещение по коду

В отладчике доступен редактор выражений. Он позволяет выполнить произвольное однострочное выражение на языке Python, находясь на точке останова. При написании выражения можно обращаться к переменным, доступным в коде на текущем шаге отладки.

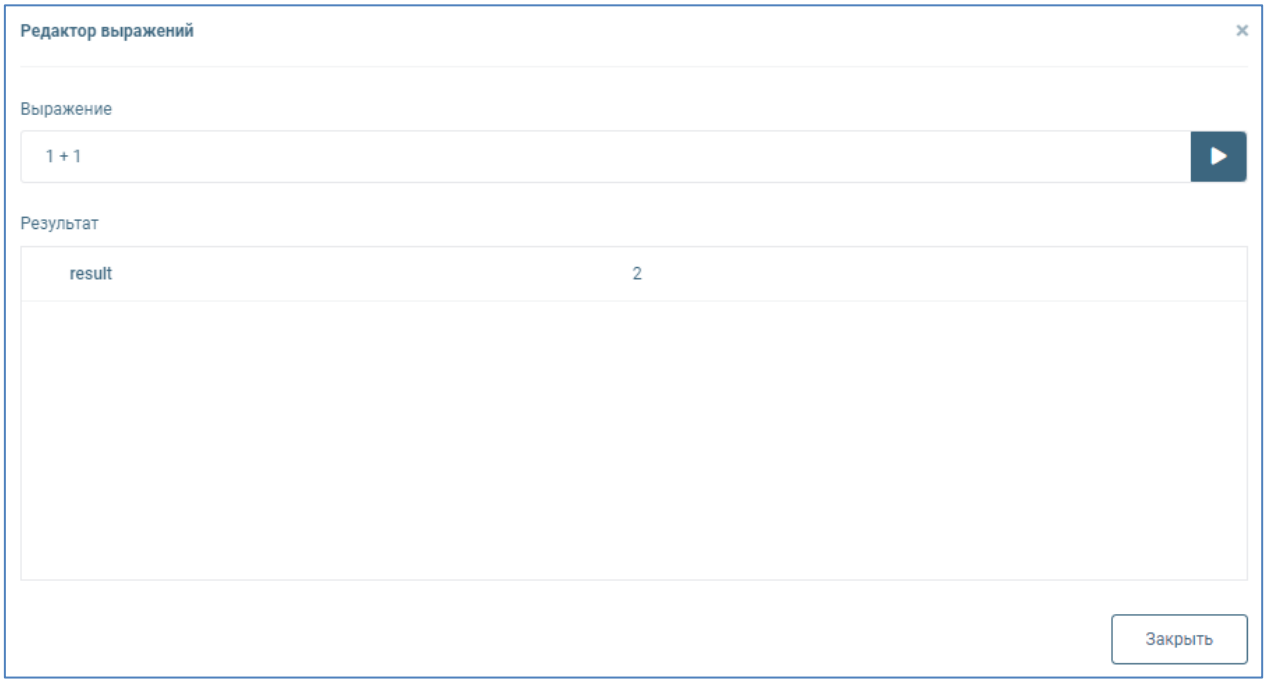


Рисунок 509 – Окно редактора выражений

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

7.6.2.4 Просмотр данных в точке останова

7.6.2.4.1 Переменные

На точках останова доступен список переменных на текущем шаге отладки. Если переменная является составной и имеет атрибуты, то отладчик позволяет просматривать их.

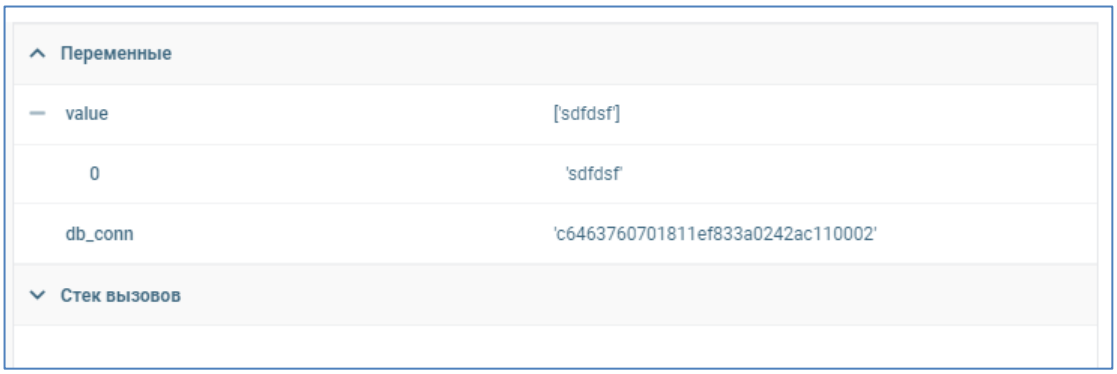


Рисунок 510 – Просмотр текущего состояния переменных

7.6.2.4.2 Стек вызовов

Доступен просмотр стека вызовов и перемещение по нему. При перемещении по стеку в списке переменных отображаются переменные, соответствующие выбранному вызову стека.

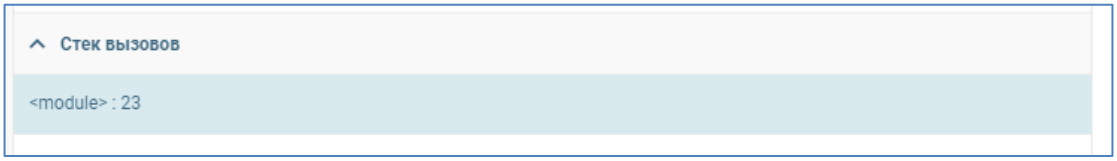


Рисунок 511 – Просмотр стека вызовов

7.6.2.4.3 Работа с исключениями

При возникновении ошибки код остановится на строке, в которой она возникла. В списке переменных при этом появится системная переменная **__exception__**, в которой содержится описание ошибки.

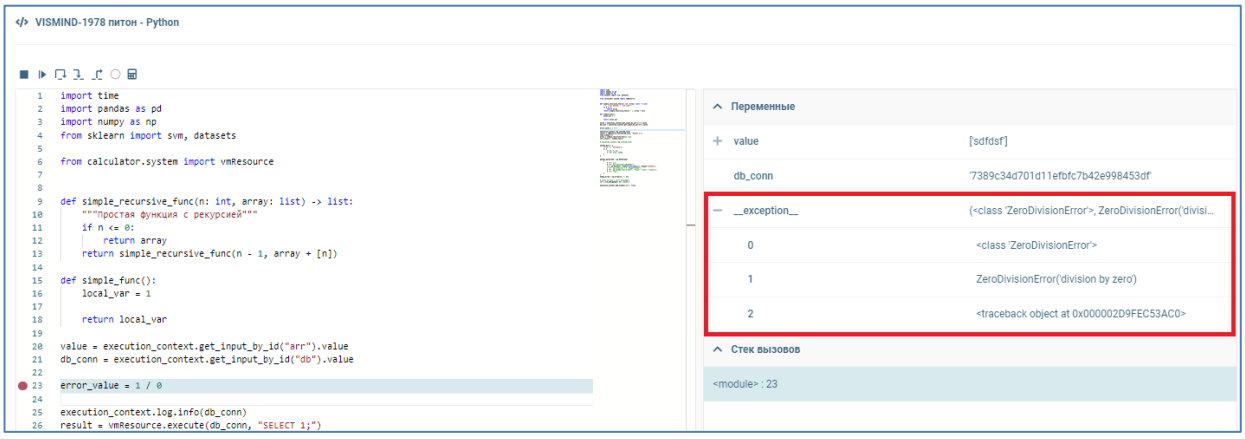


Рисунок 512 – Обработка исключений

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

7.7 Совместная работа

При работе на одном графе нескольким пользователям доступна совместная работа с блоками и фигурами на графе.

Под совместной работой понимается, возможность взаимодействовать со свободными блоками на графе, а также отображение пользователям, присутствующим на графе, действий других пользователей.

На текущий момент, отображаются следующие действия:

- Манипуляции с блоками (создание, удаление, перемещение).
- Редактирование настроек разных блоков.
- Создание связей между блоками.
- Статус «не рассчитан» для блока в котором изменили настройки.
- Отмена и восстановление изменений, вносимых пользователем (undo/redo).

7.7.1 Отображение пользователей в графе при совместной работе

Все пользователи, находящиеся в одной задаче на текущий момент, отображаются в правом верхнем углу.

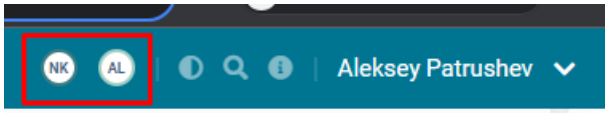


Рисунок 513 – Отображение пользователей в графе при совместной работе

Если пользователей больше двух, они отображаются в сокращённом виде.

Рисунок 514 – Отображение пользователей в графе при совместной работе

При нажатии ЛКМ по раскрывается окно со всеми пользователями, открывшими одну задачу.

Рисунок 515 – Окно со списком пользователей

При наведении курсора на иконку пользователя отображается полное имя пользователя.

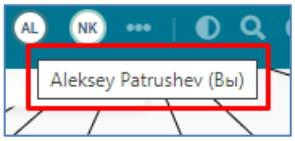


Рисунок 516 – Отображение полного имени пользователя

7.7.2 Отображение совместной работы с блоками

Создание и удаление блоков отображается для всех пользователей на графе.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

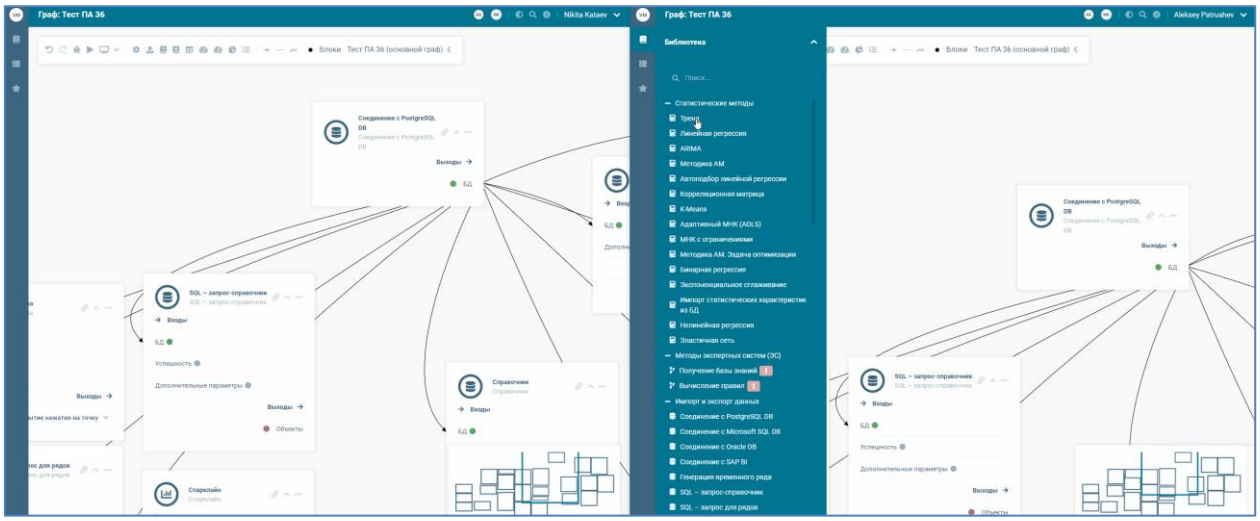


Рисунок 517 – Отображение создания и удаления блоков в графе при совместной работе

Перемещение блока отображается для остальных пользователей на графе.

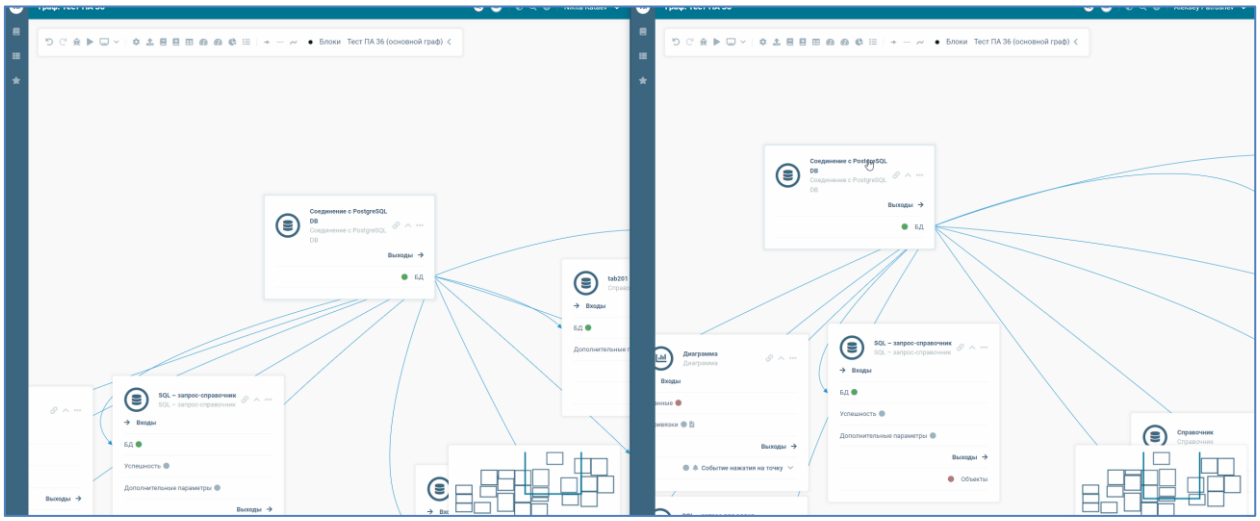


Рисунок 518 – Отображение перемещения блока в графе при совместной работе

Создание и удаление связей между блоками отображается для остальных пользователей на графе.

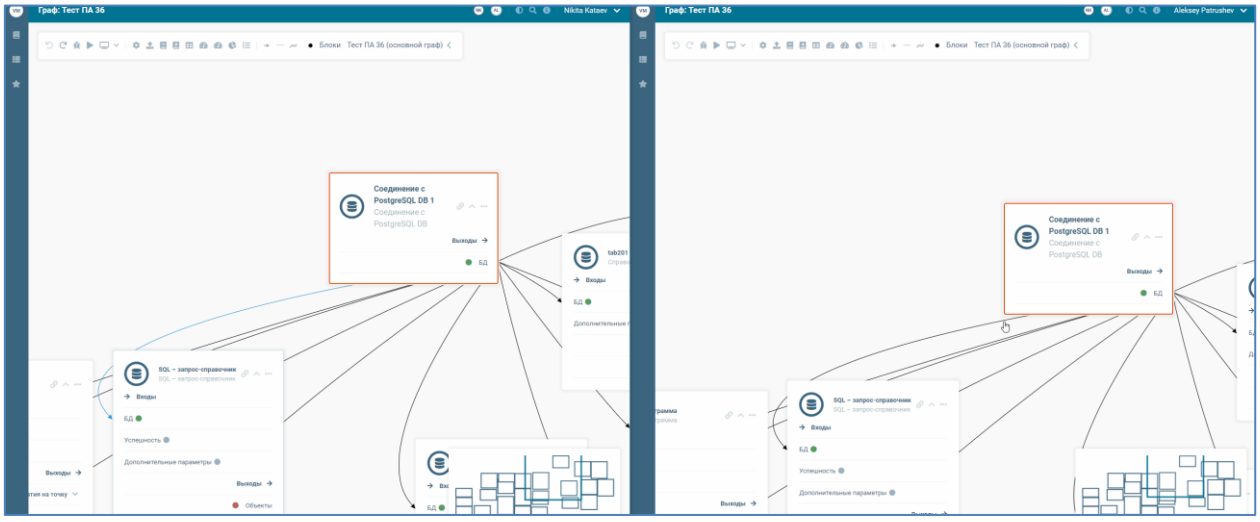


Рисунок 519 – Отображение создания и удаления связей между блоками в графе при совместной работе

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Визуальные изменения блока отображаются для остальных пользователей на графе.

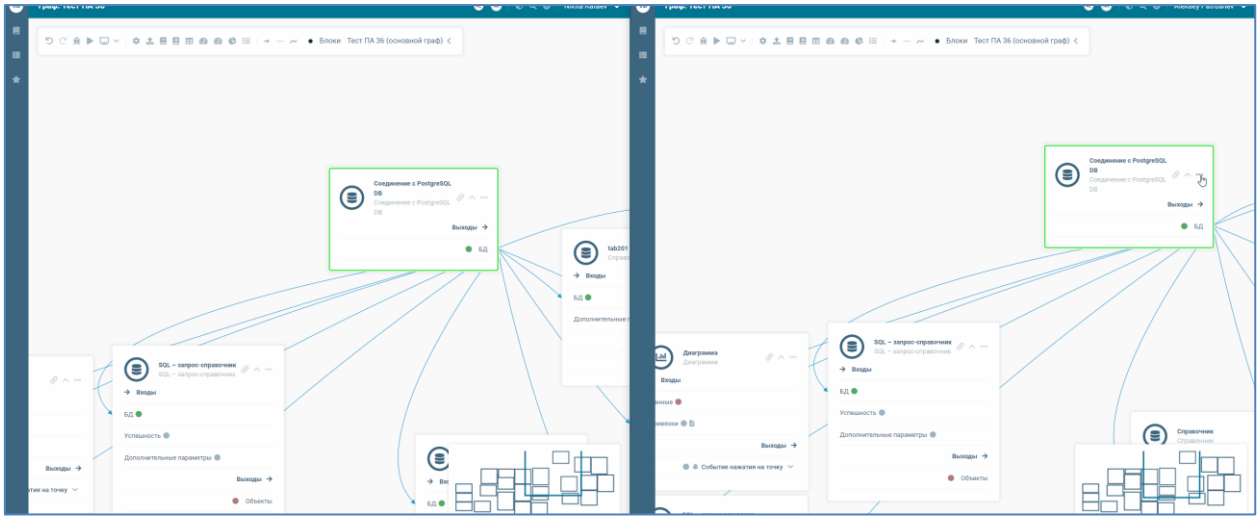


Рисунок 520 – Отображение визуальных изменений блока в графе при совместной работе

При редактировании настроек блока пользователем, редактируемый блок становится недоступным ко взаимодействию, окрашивается в серый цвет.

Также над заблокированным блоком отображается пользователь, который редактирует блок.

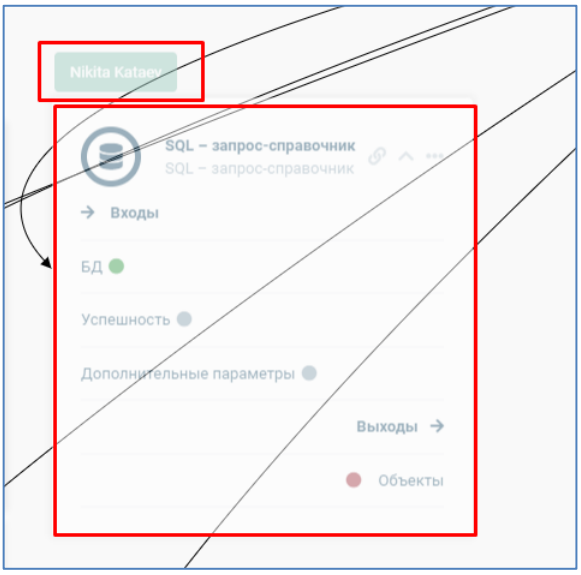


Рисунок 521 – Отображение пользователя при совместной работе

При нажатии кнопки «**Сохранить**» в настройках блока, он отображается на графе белым цветом, сообщая, что его статус не рассчитан.

Это отображается для всех пользователей на графе. Остальные состояния блоков отображаются только для пользователей, запустивших расчёт.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

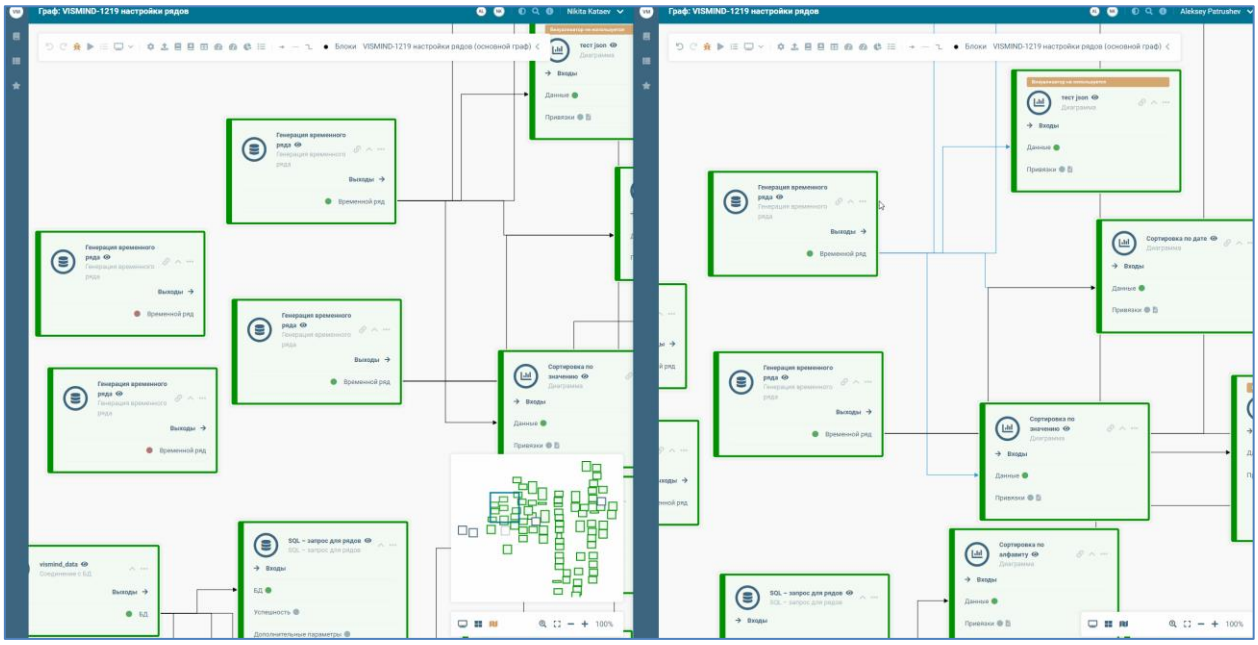


Рисунок 522 – Отображение статусов при совместной работе

7.7.3 Отмена и восстановление изменений, вносимых пользователем (undo/redo)

При работе на графе у пользователя есть возможность отменить совершенное изменение и восстановить отменённое изменение.

Механизм применяется только для изменений касающихся блоков на графе.

Если в ходе работы с графом, например, была создана таблица в БД, при отмене изменений на графе исчезнет созданный блок, но таблица сохранится в БД.

Сохраняются изменения конкретного пользователя, а не изменения всех пользователей присутствующих на графе.

Данные изменения отображаются для всех пользователей на графе.

Для отмены изменений, совершённых на графе, необходимо нажать ЛКМ по кнопке «**Отменить действие**» в меню графа, либо нажать на клавиатуре «**Ctrl + Z**».

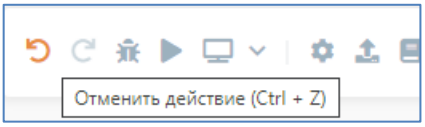


Рисунок 523 – Отображение кнопки «Отменить действие»

Для восстановления изменений, совершённых на графе необходимо нажать ЛКМ по кнопке «**Повторить действие**» в меню графа, либо нажать на клавиатуре «**Ctrl + Y**».

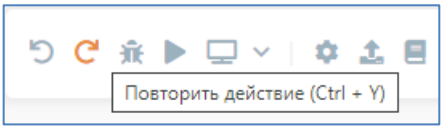


Рисунок 524 – Отображение кнопки «Повторить действие»

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

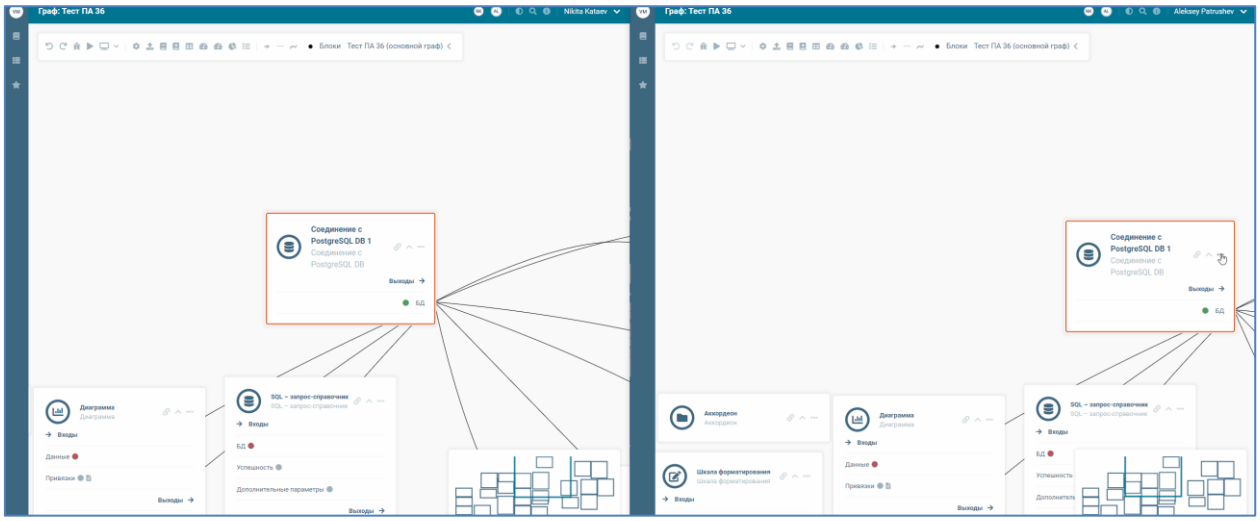


Рисунок 525 – Отображение изменений при совместной работе

Важно!

- У механизма есть ограничения в 20 изменений, после превышения этого количества удаляется самое старое изменение и в очередь добавляется новое изменение.

Важно!

- При редактировании одного блока несколькими пользователями, удаляется очередь изменений старых пользователей и остается очередь пользователя, вносившего изменения последним.

7.8 Мастера

7.8.1 Мастер настройки JSON справочника

7.8.1.1 Назначение

Мастер настройки JSON справочника позволяет создавать, изменять и удалять данные блока «JSON-Справочник» (п. 6.2.18).

7.8.1.2 Открытие мастера

Открыть мастер можно несколькими методами:

1. Нажать на кнопку «Мастер настройки JSON справочника» в панели инструментов графа.

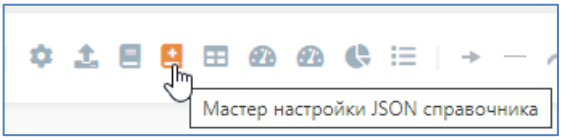


Рисунок 526 – Пример открытия мастера на панели инструментов

2. В контекстном меню блока «JSON-Справочник» выбрать пункт «Мастер...».

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

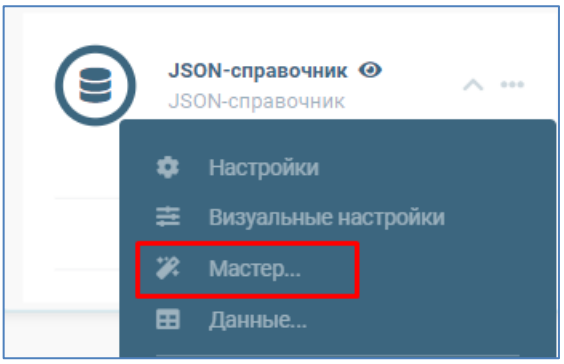


Рисунок 527 – Пример открытия мастера в настройках блока

3. Добавить блок «JSON-Справочник» на граф из библиотеки блоков через механизм drag-n-drop.

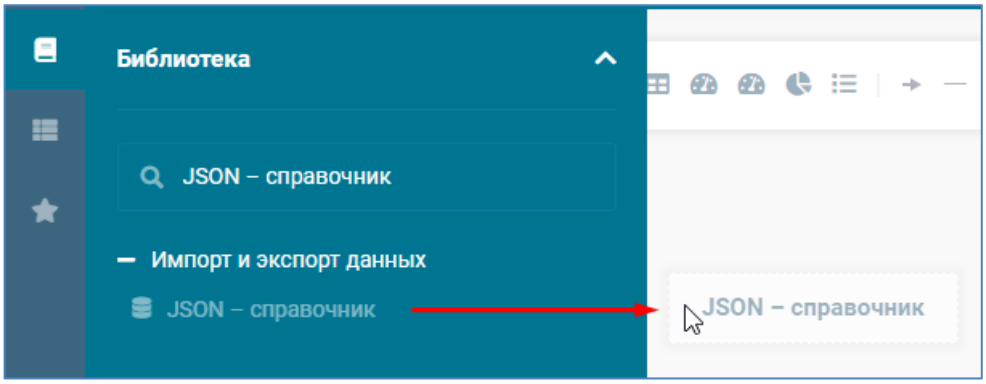


Рисунок 528 – Пример добавления блока

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

7.8.1.3 Окно работы мастера

Шаг 1

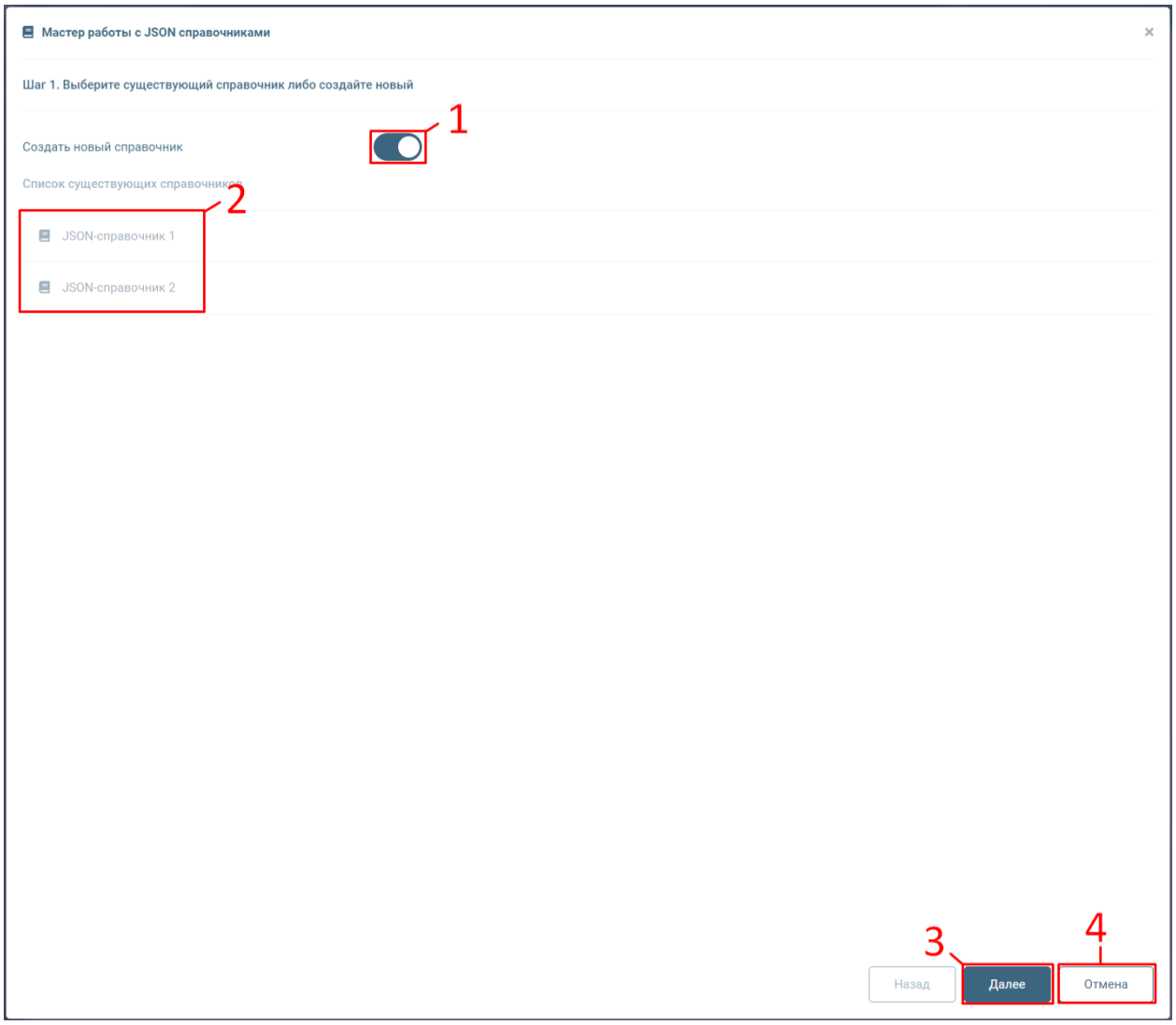


Рисунок 529 – Пример работы мастера (шаг 1)

1. Переключатель «Создать новый справочник».
    a. Если включен, то будет создан новый блок «JSON-Справочник».
    b. Если выключен, то можно выбрать один из существующих блоков «JSON-Справочник» для редактирования.
2. Список существующих на графе блоков «JSON-Справочник».
3. Кнопка «Далее». Осуществляет переход на следующий шаг мастера.
4. Кнопка «Отмена». Осуществляет выход из мастера настройки JSON справочника.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Шаг 2

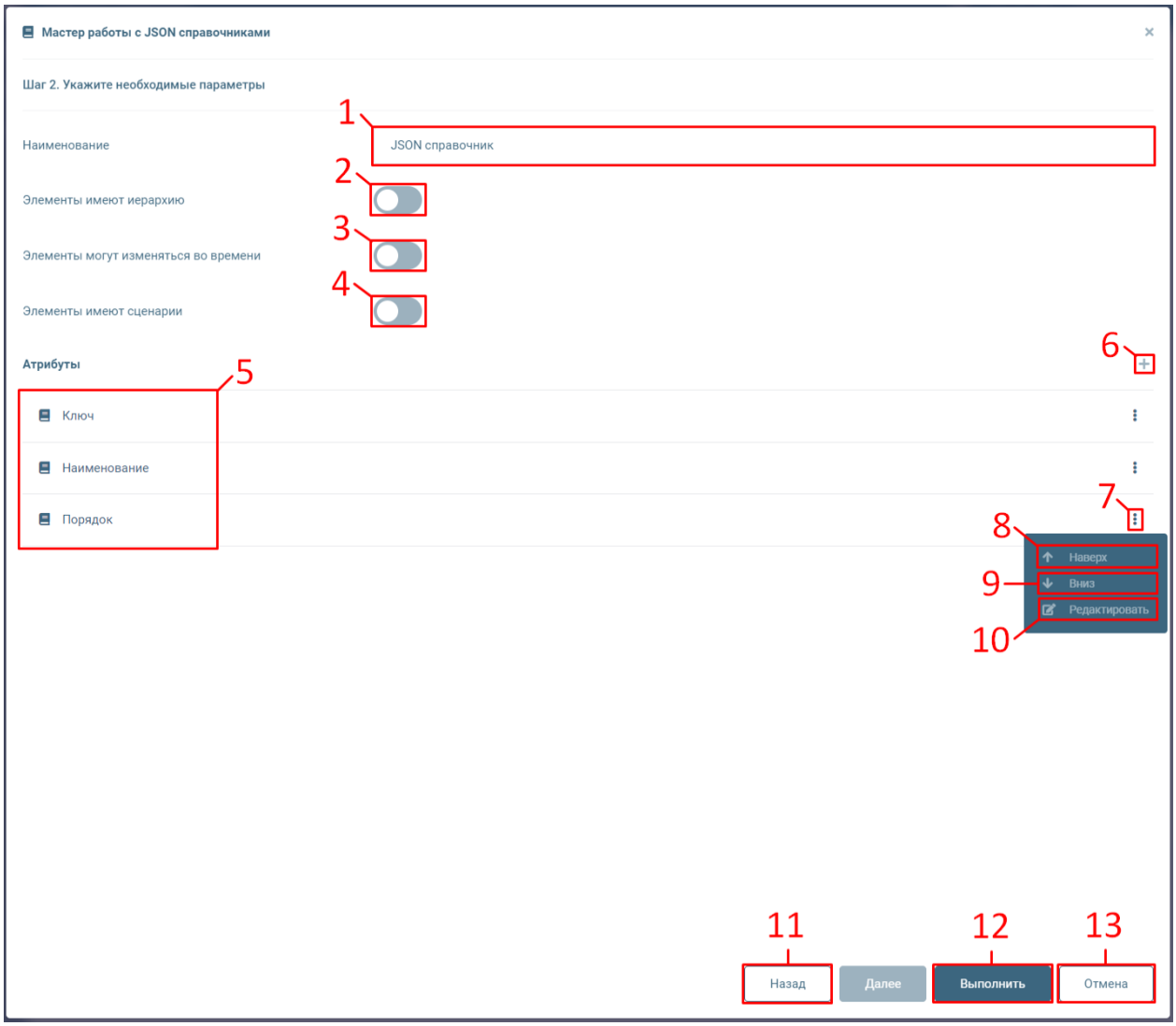


Рисунок 530 – Пример работы мастера (шаг 2)

1. Поле ввода наименования блока.
2. Переключатель «Элементы имеют иерархию». При включении автоматически добавляется атрибут «Родитель».

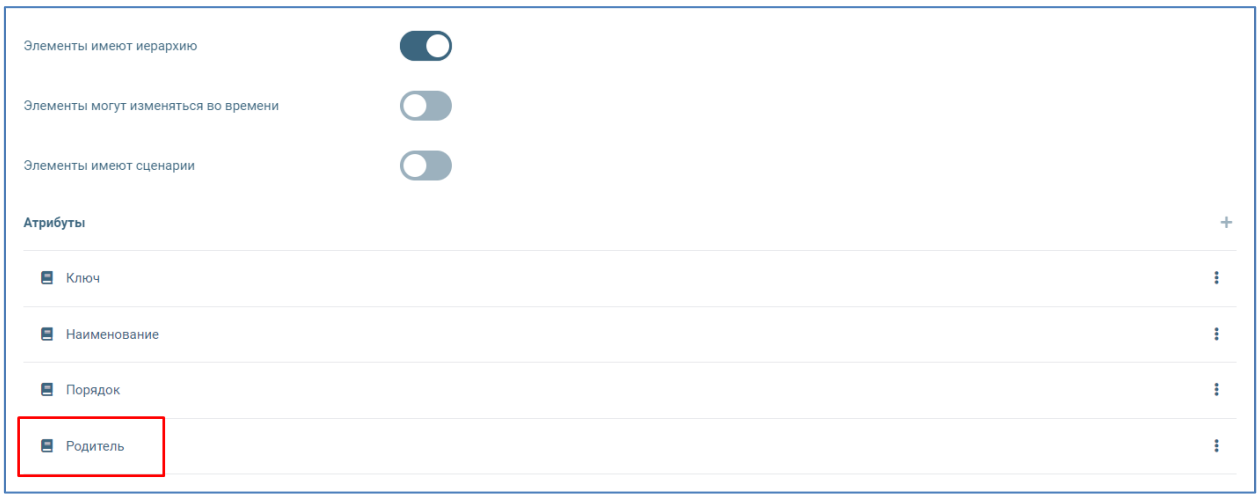


Рисунок 531 – Пример настройки элементов с иерархией

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

3. Переключатель «Элементы могут изменяться во времени». При включении автоматически добавляются атрибуты: «Версия», «Дата начала», «Дата окончания».

Рисунок 532 – Пример настройки элементов с версионностью

4. Список созданных атрибутов.
5. Кнопка добавления нового атрибута.
6. Кнопка открытия контекстного меню атрибута.
7. Кнопка «Наверх». Изменяет порядок атрибута: сдвигает на 1 позицию наверх.
8. Кнопка «Вниз». Изменяет порядок атрибута: сдвигает на 1 позицию вниз.
9. Кнопка «Редактировать». Открывает окно настройки атрибута (подробнее в соответствующем разделе ниже).
10. Кнопка «Назад». Возвращает на предыдущий шаг мастера.
11. Кнопка «Выполнить». Осуществляет сохранение справочника с текущими настройками.
12. Кнопка «Отмена». Осуществляет выход из мастера настройки JSON справочника.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

7.8.1.4 Окно «Настройки атрибута»

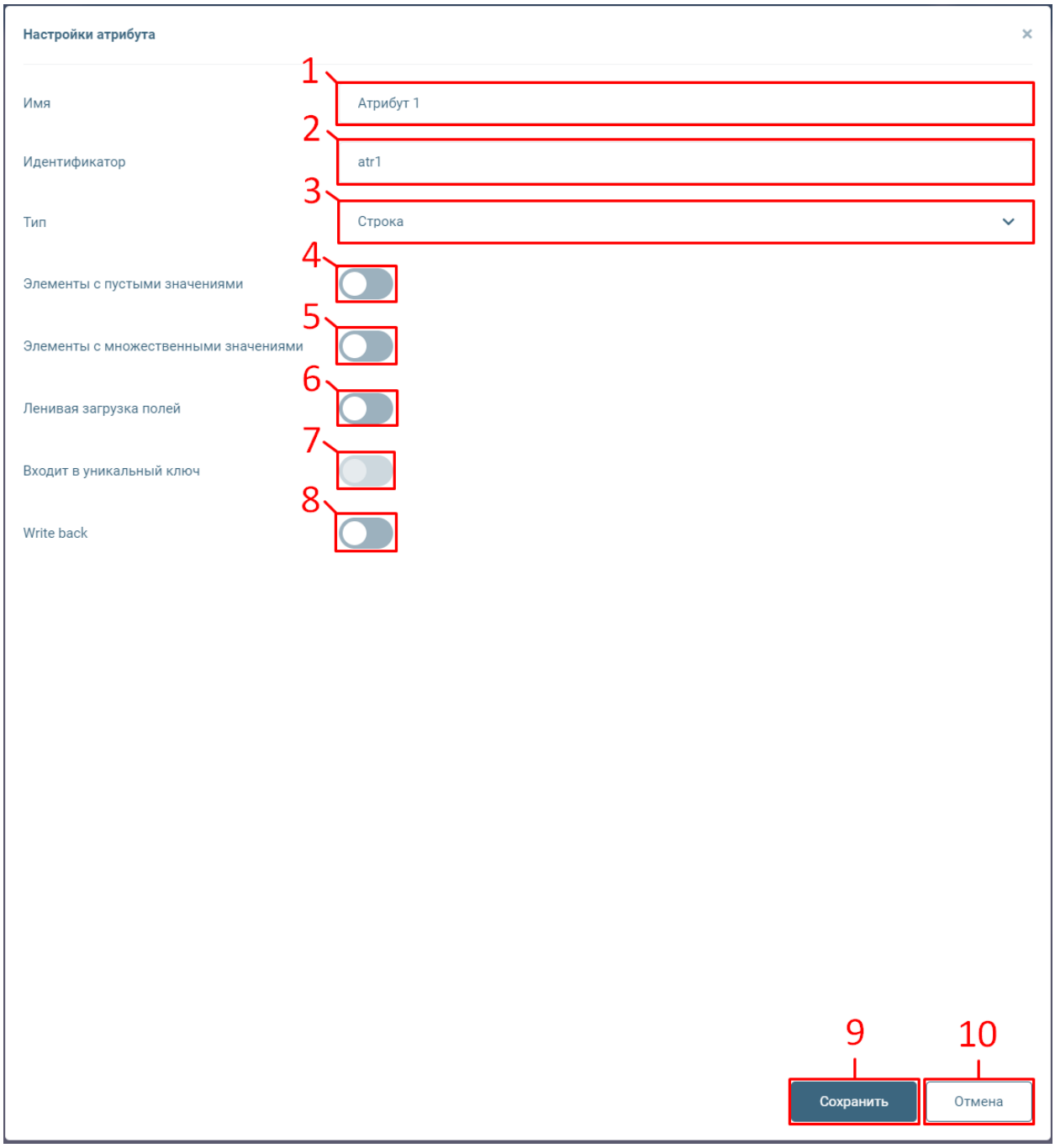


Рисунок 533 – Пример работы мастера (настройки атрибута)

1. Поле ввода наименования атрибута.
2. Поле ввода идентификатора атрибута (допускается ввод только с использованием латиницы).
3. Выпадающий список с выбором типа атрибута. Доступные типы:
    a. Строка.
    b. Дата. При выборе данного типа появляются дополнительные параметры: «Значение по умолчанию», «Маска», «Динамика».

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

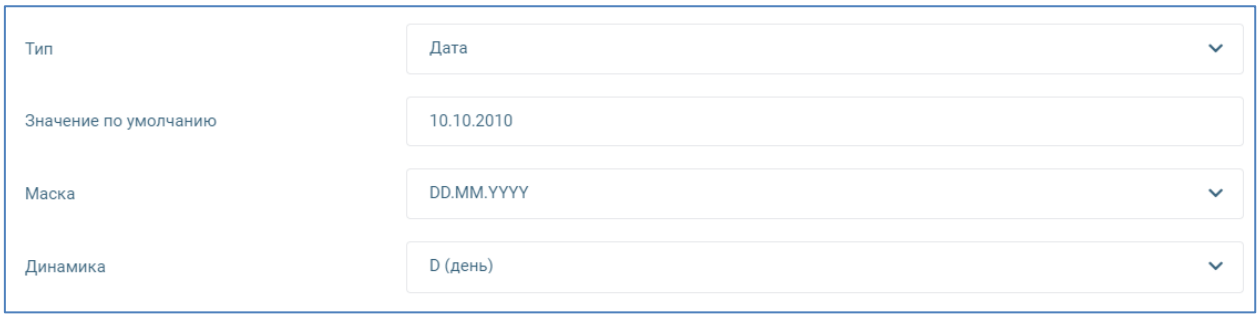


Рисунок 534 – Пример настройки атрибута с типом «Дата»

c. Целочисленный.

d. Вещественный.

e. Координаты.

f. JSON.

g. Справочник. При выборе данного типа появляется дополнительный параметр «Справочник», в котором из выпадающего списка можно выбрать другой справочник графа.

Рисунок 535 – Пример настройки атрибута с типом «Справочник»

4. Переключатель «Элементы с пустыми значениями»:
   a. Если включен, то значения данного атрибута могут быть пустыми.
   b. Если выключен, то значения данного атрибута не могут быть пустыми.
5. Переключатель «Элементы с множественными значениями». Не поддерживается в блоке «JSON-Справочник».
6. Переключатель «Ленивая загрузка полей». Не поддерживается в блоке «JSON-Справочник».
7. Переключатель «Входит в уникальный ключ». Не поддерживается в блоке «JSON-Справочник».
8. Переключатель «Write back». Не поддерживается в блоке «JSON-Справочник».
9. Кнопка «Сохранить». Сохраняет атрибут с текущими настройками.
10. Кнопка «Отмена». Закрывает окно «Настройки атрибута» без сохранения изменений.

7.8.1.5 Редактирование данных

Чтобы открыть меню редактирования данных, нужно в контекстном меню блока «JSON-Справочник» выбрать пункт «Данные...».

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

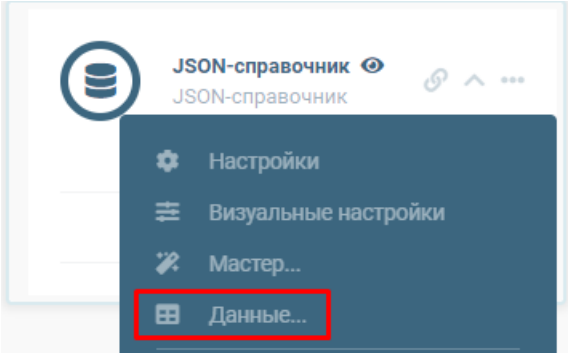


Рисунок 536 – Пример выбора меню редактирования данных

7.8.1.6 Окно «Данные»

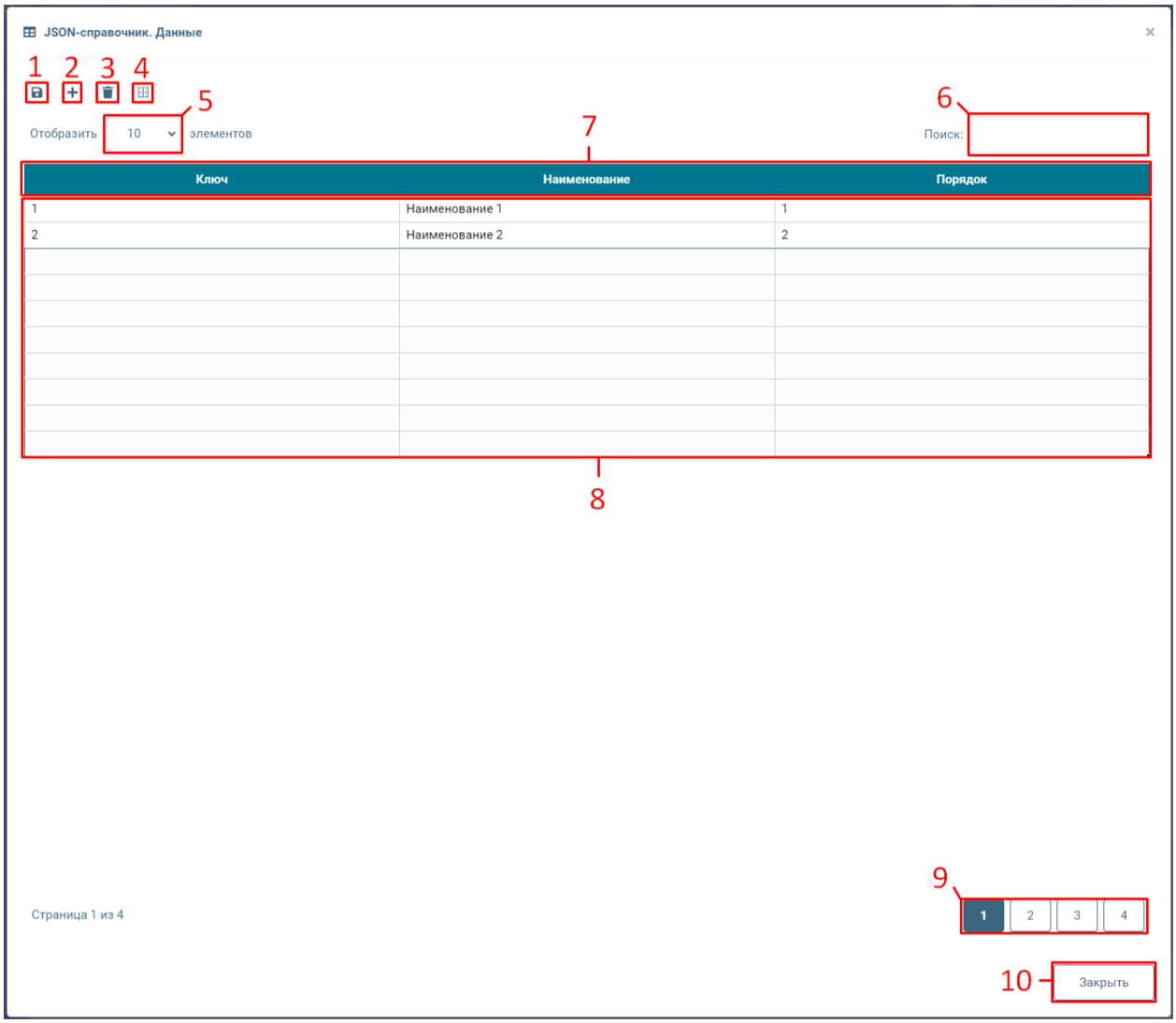


Рисунок 537 – Пример окна для редактирования данных

1. Кнопка «Сохранить». Активна, если данные были изменены. Сохраняет изменения данных.
2. Кнопка «Добавить». Создает новую пустую строку в таблице.
3. Кнопка «Удалить». Удаляет выбранный диапазон строк. После удаления необходимо нажать кнопку «Сохранить», чтобы изменения вступили в силу.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

4. Кнопка «Отобразить данные в виде кросс-таблицы». Не поддерживается в блоке «JSON-Справочник».
5. Выпадающий список, который управляет количеством отображаемых строк в таблице на одной странице. Возможно отображать по:
    a. 10 строк;
    b. 25 строк;
    c. 50 строк;
    d. 100 строк.
6. Поле поиска. Осуществляет поиск по всей таблице с любыми совпадениями в значениях.
7. Заголовок таблицы. Содержит список атрибутов JSON-Справочника.
8. Строки таблицы:
    a. В строках осуществляется редактирование значений.
    b. Возможны как ввод с клавиатуры, так и вставка из буфера обмена.
    c. Вставлять из буфера обмена можно как одно значение в ячейку, так и диапазон ячеек.
    d. Атрибуты «Ключ» и «Порядок» необязательно заполнять: при сохранении значения для этих атрибутов будут заполнены автоматически, если они пустые.
9. Страницы таблицы. Для удобства просмотра и улучшения производительности данные справочника отображаются постранично. Между страницами можно переключаться, кликнув ЛКМ на соответствующий номер.
10. Кнопка «Закрыть». Закрывает окно «Данные». Если при нажатии произведенные изменения в данных не были сохранены, то появится диалоговое окно с подтверждением сохранения.

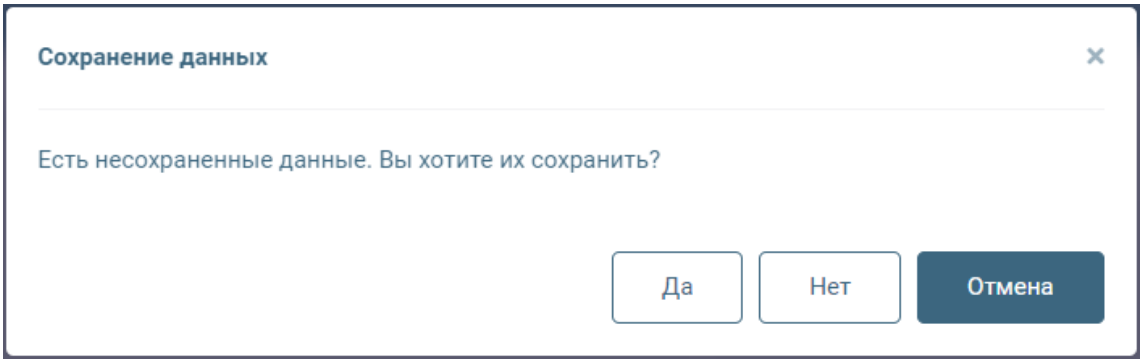


Рисунок 538 – Пример диалогового окна для сохранения данных

7.8.2 Мастер настройки OLAP

7.8.2.1 Назначение

Мастер настройки OLAP позволяет изменять и сохранять отметку в элементах управления (справочниках) блока «Панель OLAP». Сохраненная отметка будет использована при расчете блока/графа, а также отображаться при открытии пресета.

Настроенная отметка актуальна для всех пользователей (т.е. не привязана к пользователю, который вносит изменения).

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

7.8.2.2 Открытие мастера

Открыть мастер можно несколькими методами:

1. Нажать на кнопку «Мастер настройки OLAP» в панели инструментов графа

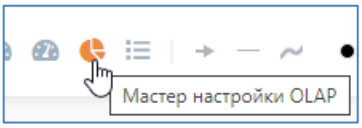


Рисунок 539 – Пример открытия мастера на панели инструментов

2. В контекстном меню блока «Панель OLAP» выбрать пункт «Мастер...».

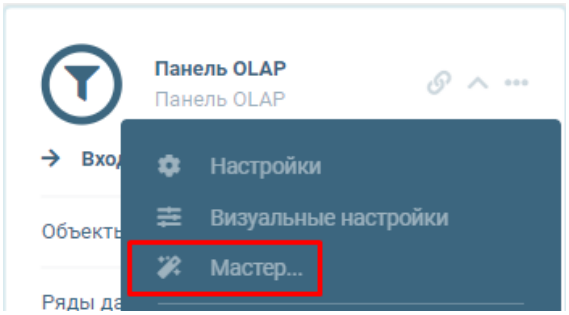


Рисунок 540 – Пример открытия мастера в настройках блока

7.8.2.3 Окно работы мастера

Шаг 1

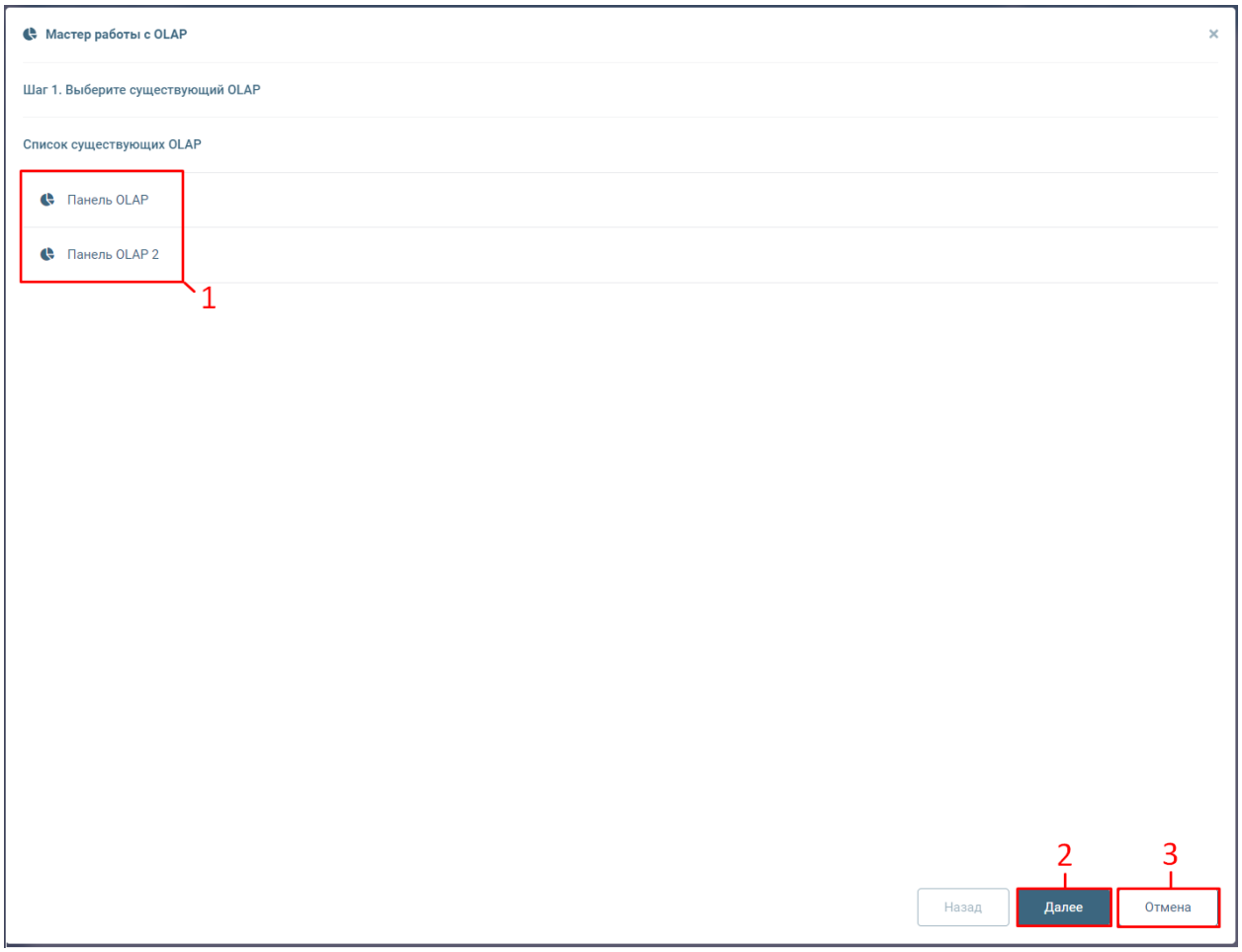


Рисунок 541 – Пример работы мастера (шаг 1)

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

1. Список существующих на графе блоков «Панель OLAP».
2. Кнопка «Далее». Осуществляет переход на следующий шаг мастера.
3. Кнопка «Отмена». Осуществляет выход из мастера настройки OLAP.

Шаг 2

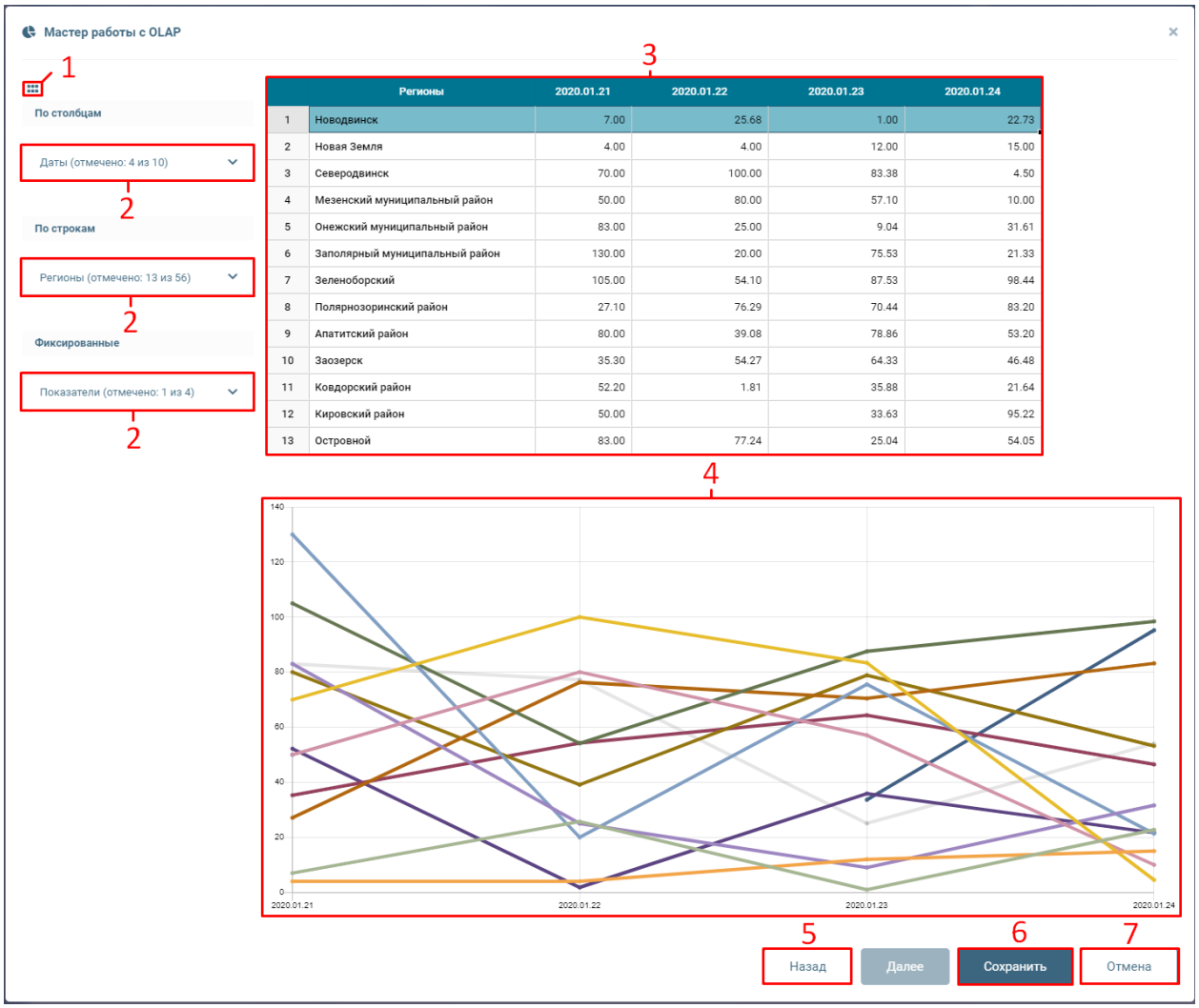


Рисунок 542 – Пример работы мастера (шаг 2)

1. Кнопка «Сменить расположение». Меняет местами справочники, расположенные по столбцам, со справочниками, расположенными по строкам.
2. Выпадающие списки – измерения OLAP, которые предварительно должны быть настроены в блоке «Панель OLAP». Позволяют пользователю отмечать интересующие его элементы.
3. Таблица. Отображает данные, приходящие на вход блока «Панель OLAP» согласно настроенной отметке в выпадающих списках. Аналогично будет выглядеть блок «Таблица», если расположить его после блока «Панель OLAP».
4. Диаграмма. Отображает данные, приходящие на вход блока «Панель OLAP» согласно настроенной отметке в выпадающих списках. Аналогично будет выглядеть блок «Диаграмма», если расположить его после блока «Панель OLAP».
5. Кнопка «Назад». Возвращает на предыдущий шаг мастера.
6. Кнопка «Сохранить». Осуществляет сохранение настроенной отметки. При успешном сохранении отображается окно.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

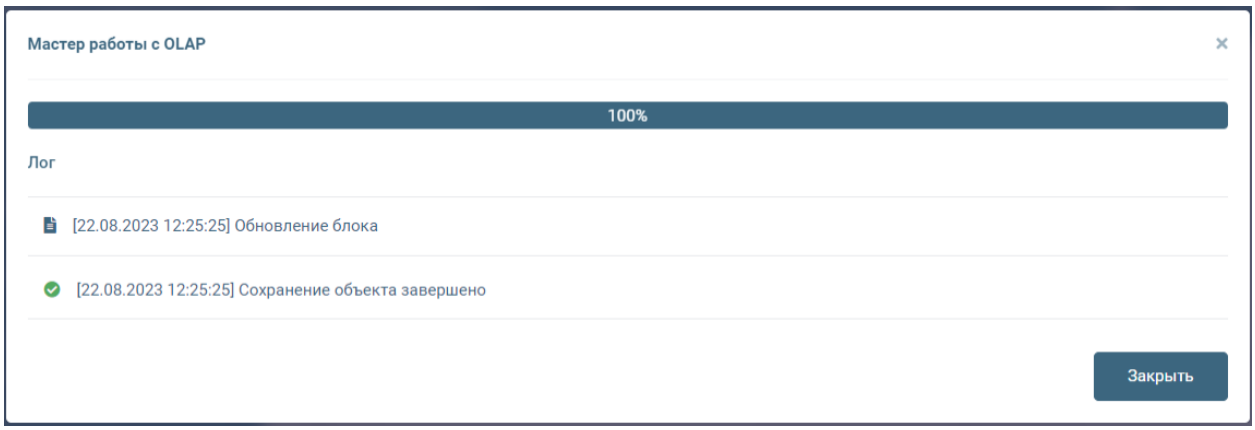


Рисунок 543 – Пример завершения работы мастера

Кнопка «Закрыть» завершает работу мастера.

7. Кнопка «Отмена». Осуществляет выход из мастера настройки OLAP.

7.8.3 Мастер настройки импорта

7.8.3.1 Назначение

Мастер настройки импорта позволяет импортировать в БД файлы формата csv, xlsx, xml и json

7.8.3.2 Открытие мастера

Открыть мастер можно несколькими методами:

1. Нажать на кнопку «Мастер настройки импорта» в панели инструментов графа.

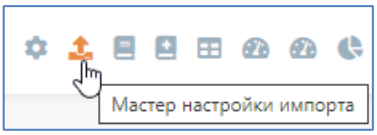


Рисунок 544 – Пример открытия мастера на панели инструментов

2. В контекстном меню блока «Импорт файла» выбрать пункт «Мастер...».

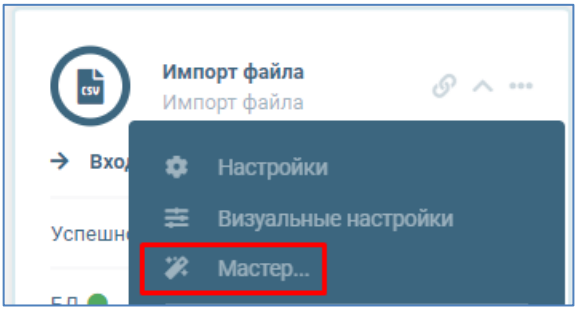


Рисунок 545 – Пример открытия мастера в настройках блока

3. Добавить блок «Импорт файла» на граф из библиотеки блоков через механизм drag-n-drop.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

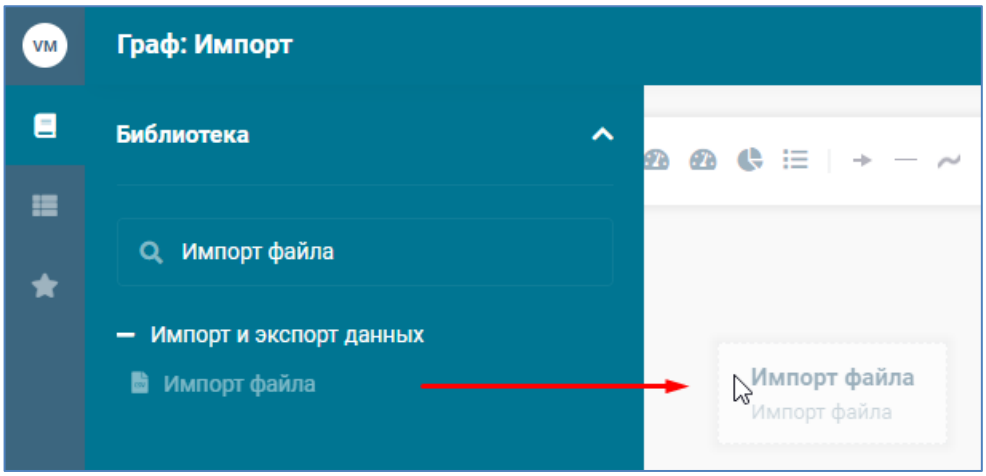


Рисунок 546 – Пример добавления блока

7.8.3.3 Окно работы мастера

Шаг 1

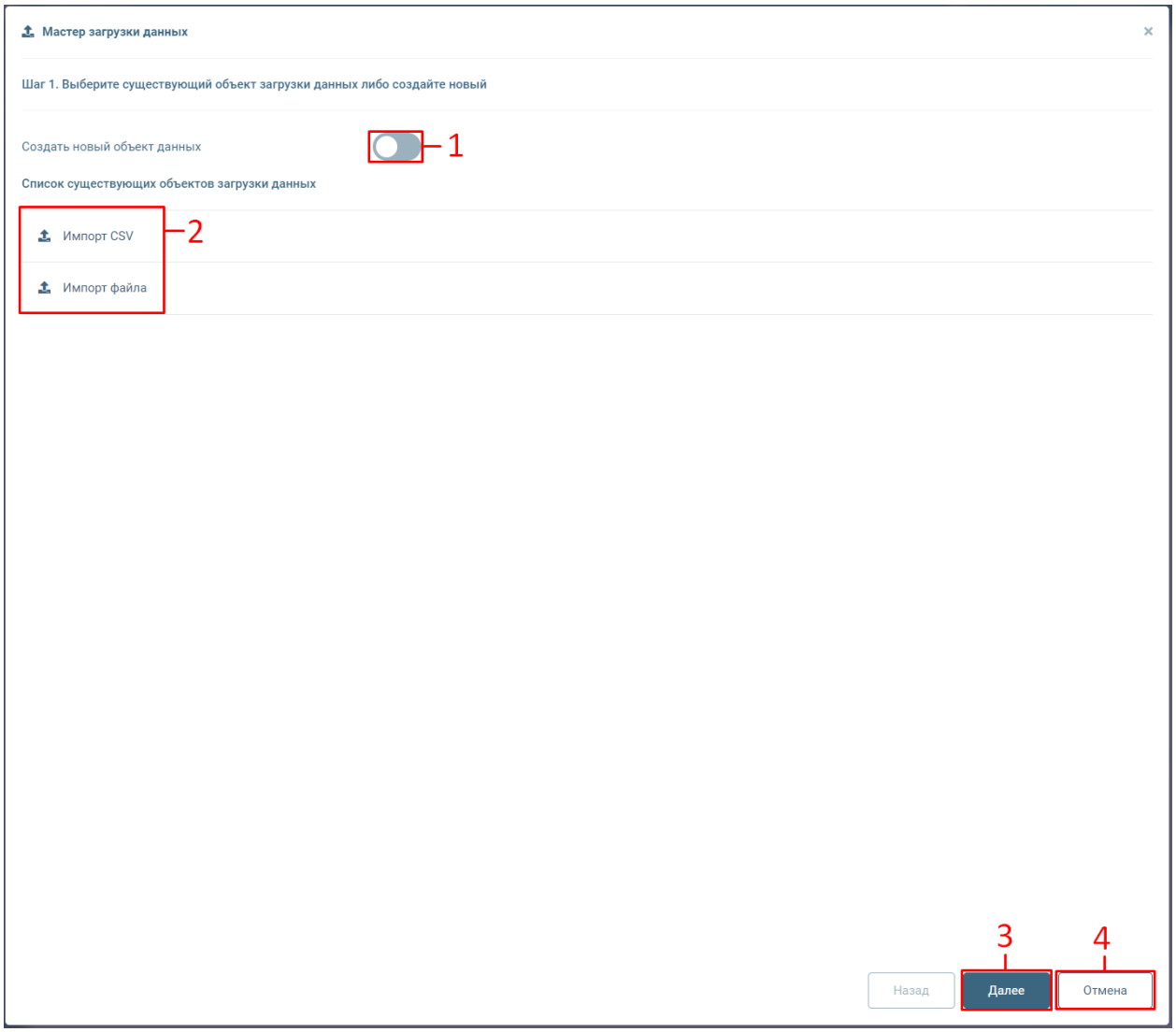


Рисунок 547 – Пример работы мастера (шаг 1)

1. Переключатель «Создать новый объект данных»:
   - Если включен, то будет создан новый блок «Импорт файла».

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

- Если выключен, то можно выбрать один из существующих блоков «Импорт файла».

2. Список существующих на графе блоков «Импорт файла».
3. Кнопка «Далее». Осуществляет переход на следующий шаг мастера.
4. Кнопка «Отмена». Осуществляет выход из мастера настройки импорта.

Шаг 2

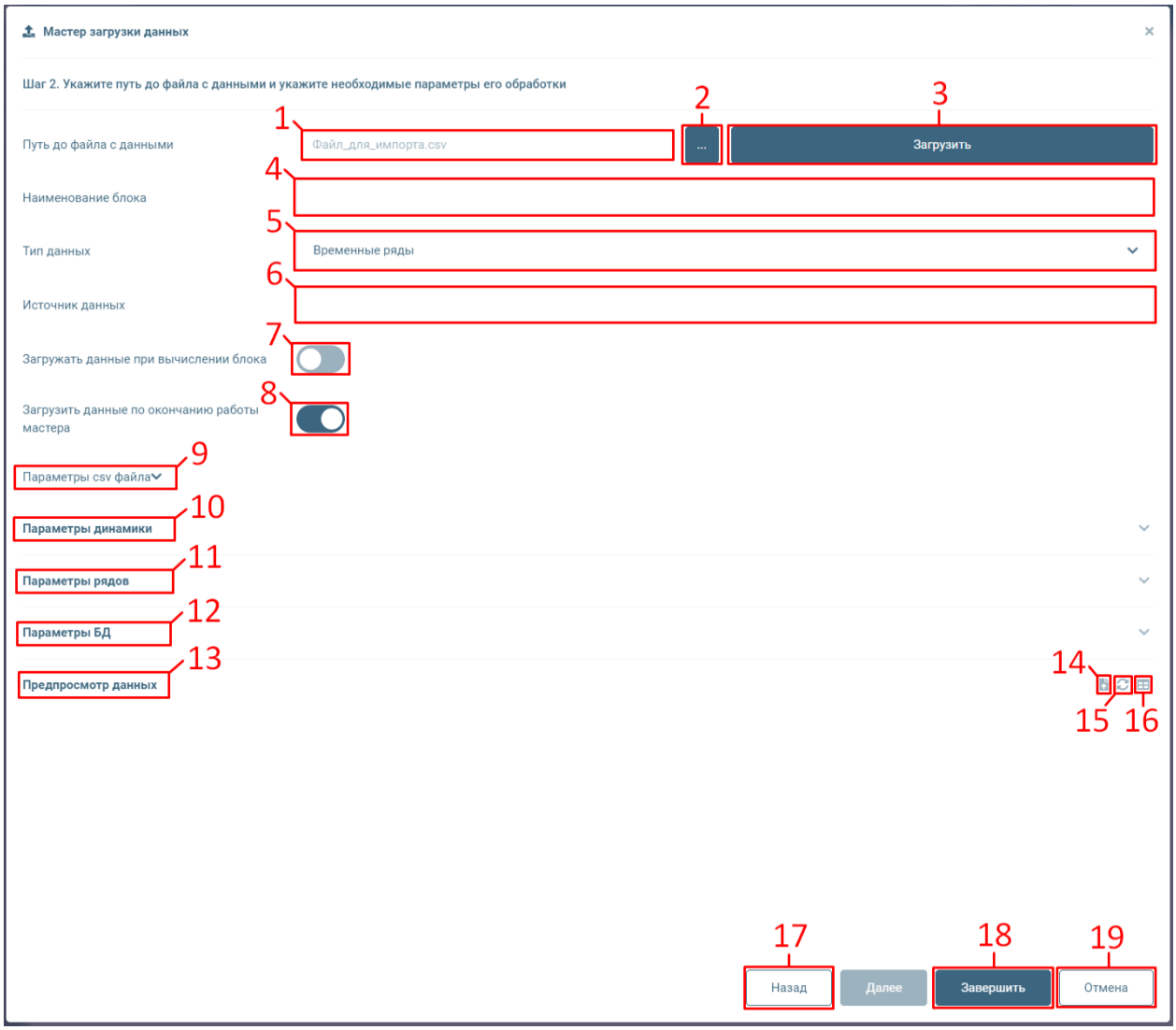


Рисунок 548 – Пример работы мастера (шаг 2)

1. Поле, в котором отображается имя файла.
2. Кнопка выбора файла. Открывает интерфейс выбора файла с компьютера пользователя.
3. Кнопка «Загрузить». Становится доступна после выбора файла. Загружает выбранный файл на сервер.

   Импорт в БД не происходит после нажатия кнопки «Загрузить», загружается сам файл для его последующей обработки и затем импорта.
4. Поле ввода наименования блока.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

5. Выпадающий список «Тип данных». В зависимости от выбранного типа меняются настройки в мастере. Доступные типы:
   - Временные ряды;
   - Справочник;
   - Кросс-таблица.
6. Поле ввода источника данных. При создании таблицы в БД, в поле datasource будет использовано значение из данной настройки.
7. Переключатель «Загружать данные при вычислении блока». Если включен, импорт файла будет происходить при каждом расчете блока.
8. Переключатель «Загрузить данные по окончанию работы мастера». Если включен, импорт файла произойдет после нажатия кнопки «Завершить».
9. Группа настроек «Параметры csv файла».

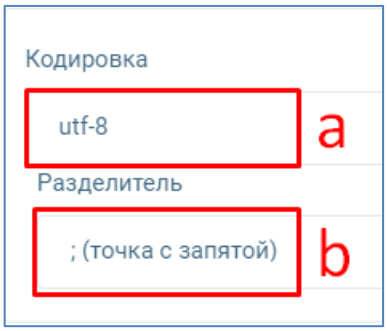


Рисунок 549 – Пример настройки параметров кодировки

   a. Выпадающий список «Кодировка». Доступные значения:
      - utf-8;
      - cp866;
      - cp1251;
      - maccyrillic;
      - koi8_r.
   b. Выпадающий список «Разделитель». Доступные значения:
      - ; (точка с запятой);
      - : (двоеточие);
      - , (запятая).
10. Группа настроек «Параметры динамики».

Рисунок 550 – Пример настройки параметры динамики

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

a. Выпадающий список «Маска». Доступные значения:
   - YYYY;
   - MM.YYYY;
   - QQ.YYYY;
   - DD.MM.YYYY;
   - DD.MM.YYYY hh:mm;
   - DD.MM.YYYY hh:mm:ss.

b. Выпадающий список «Динамика». Доступные значения:
   - Y (год);
   - M (месяц);
   - D (день).

11. Группа настроек «Параметры рядов».

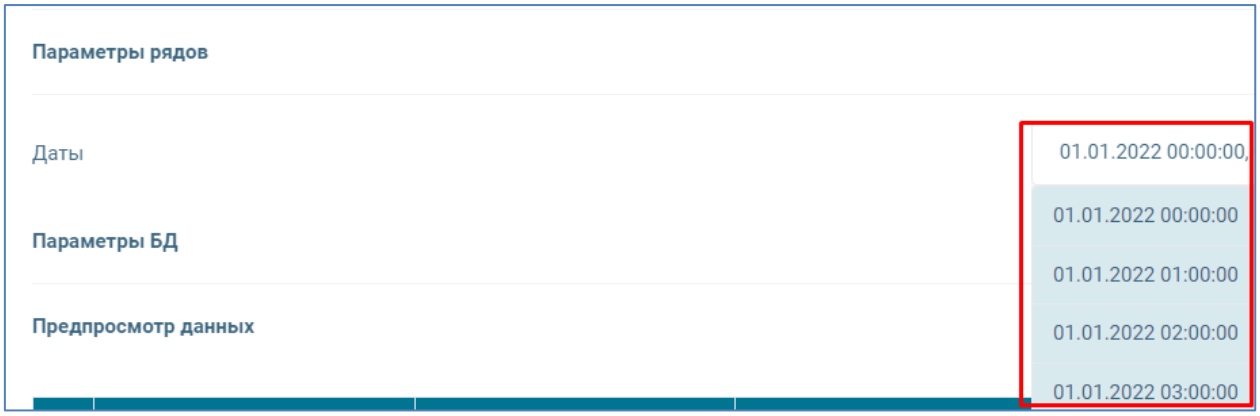


Рисунок 551 – Пример настройки параметров ряда

Выпадающий список «Даты» позволяет выбрать даты из файла, по которым будет происходить импорт (по умолчанию выбраны все даты)

12. Группа настроек «Параметры БД».

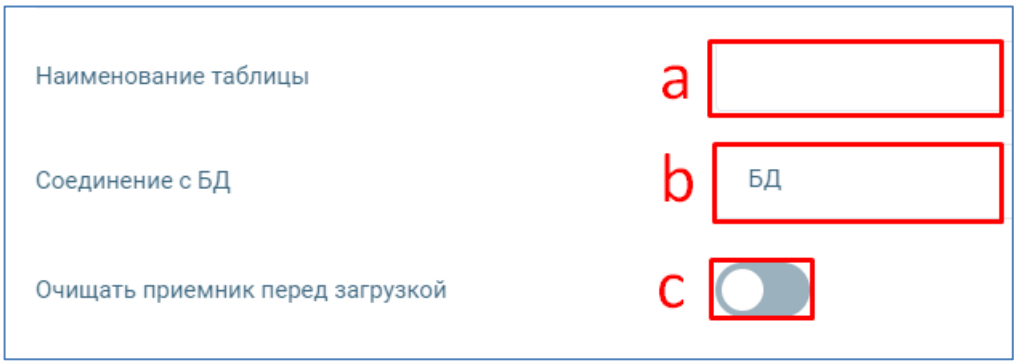


Рисунок 552 – Пример настройки параметров БД

a. Поле ввода «Наименование таблицы». С данным наименованием будет создана таблица в БД. Если такая таблица есть, то ее данные будут обновлены при импорте

b. Выпадающий список «Соединение с БД». Содержит перечень блоков графа, которые производят подключение к БД. Файл будет импортирован в ту БД, которая указана в выбранном соединении

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

c. Переключатель «Очищать приемник перед загрузкой». Если включен, то перед импортом данные в указанной таблице будут удаляться, если данная таблица существует

13. Предпросмотр данных. Отображает загруженный файл в виде таблицы.

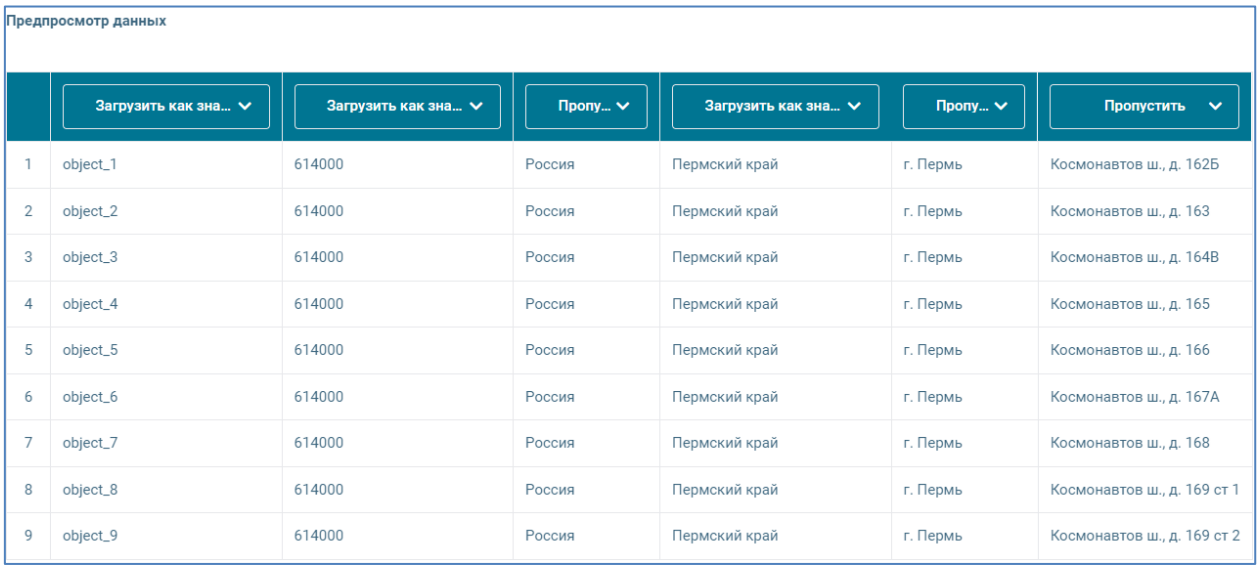


Рисунок 553 – Пример предпросмотра данных

В столбцах есть возможность выбрать, как их загружать:

a. Пропустить. Не производить загрузку.

b. Загрузить как значения. Загрузить значения с типом varchar.

c. Загрузить как дату. Загрузить значения с типом timestamp. При этом будут использованы настройки раздела «Параметры динамики».

d. Загрузить как атрибут справочника. Если в графе есть блоки «Справочник» или «SQL – запрос-справочник», то при выборе поля справочника, в котором содержатся наименования элементов, значения в нём будут сопоставлены со значениями в данном столбце файла. При импорте, в таблицу БД будут импортированы id элементов из справочника, имена которых были найдены в файле. Такие значения в таблице предпросмотра будут иметь зеленую заливку.

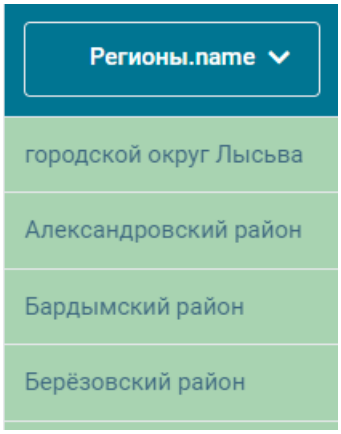


Рисунок 554 – Пример импорта данных из файла

14. Кнопка «Предварительная загрузка». Имитирует импорт файла. Служит для проверки корректности произведенных настроек. Импорт в БД при этом не происходит.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

15. Кнопка «Обновить». Обновляет таблицу предпросмотра. Для случаев, когда файл был изменен.
16. Кнопка «Загрузить больше данных». По умолчанию в таблице предпросмотра отображается 10 строк, если пользователь хочет увидеть больше строк, то может воспользоваться данной кнопкой.
17. Кнопка «Назад». Возвращает на предыдущий шаг мастера.
18. Кнопка «Завершить». Производит импорт файла в БД.

В процессе загрузки пользователь видит лог загрузки. При достижении 100% прогресса будет отображен результат импорта.

Рисунок 555 – Пример завершения работы мастера

Работа мастера на этом завершена. После нажатия кнопки «Закрыть» пользователь будет возвращен в граф.

19. Кнопка «Отмена». Осуществляет выход из мастера настройки импорта.

7.8.4 Мастер настройки показателей

7.8.4.1 Назначение

Мастер настройки показателей позволяет создавать, изменять и удалять данные блока «Показатели» (см. п. 6.2.17).

7.8.4.2 Открытие мастера

Открыть мастер можно несколькими методами:

1. Нажать на кнопку «Мастер настройки показателей» в панели инструментов графа.

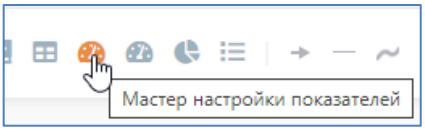


Рисунок 556 – Пример открытия мастера на панели инструментов

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

2. В контекстном меню блока «Показатели» выбрать пункт «Мастер...».

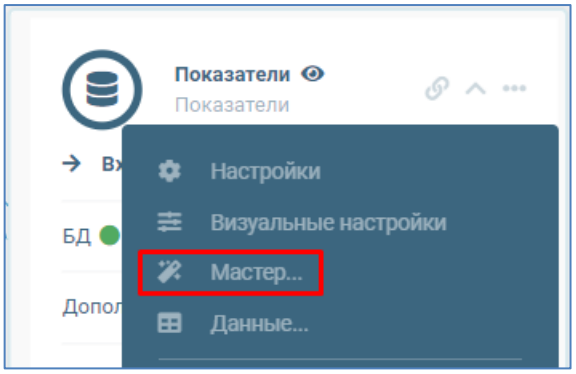


Рисунок 557 – Пример открытия мастера в настройках блока

3. Добавить блок «Показатели» на граф из библиотеки блоков через механизм drag-n-drop.

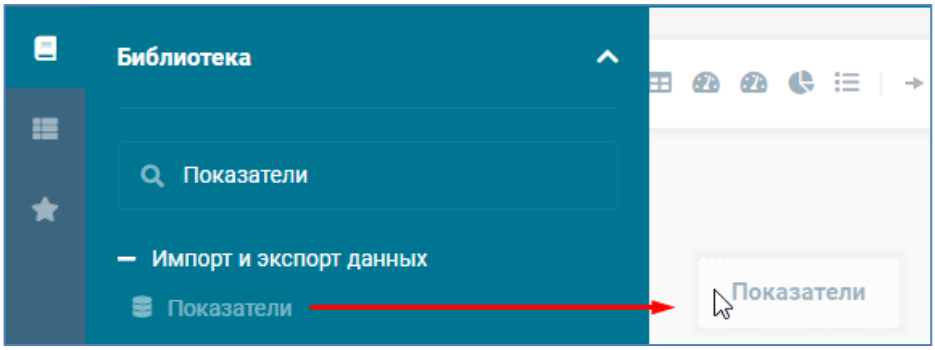


Рисунок 558 – Пример добавления блока

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

7.8.4.3 Окно работы мастера

Шаг 1

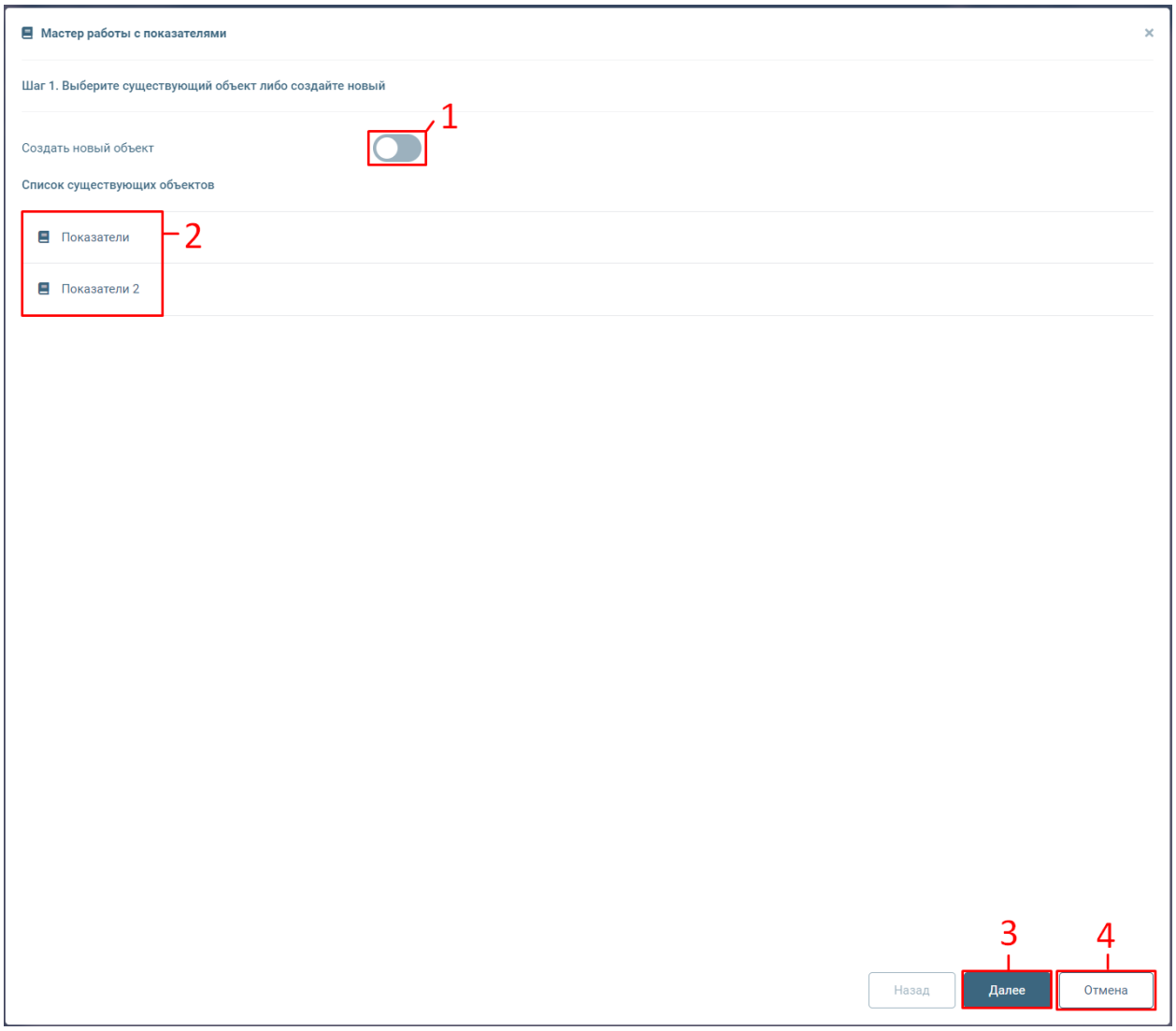


Рисунок 559 – Пример работы мастера (шаг 1)

1. Переключатель «Создать новый объект».
   - Если включен, то будет создан новый блок «Показатели».
   - Если выключен, то можно выбрать один из существующих блоков «Показатели» для редактирования.
2. Список существующих на графе блоков «Показатели».
3. Кнопка «Далее». Осуществляет переход на следующий шаг мастера.
4. Кнопка «Отмена». Осуществляет выход из мастера настройки показателей.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Шаг 2

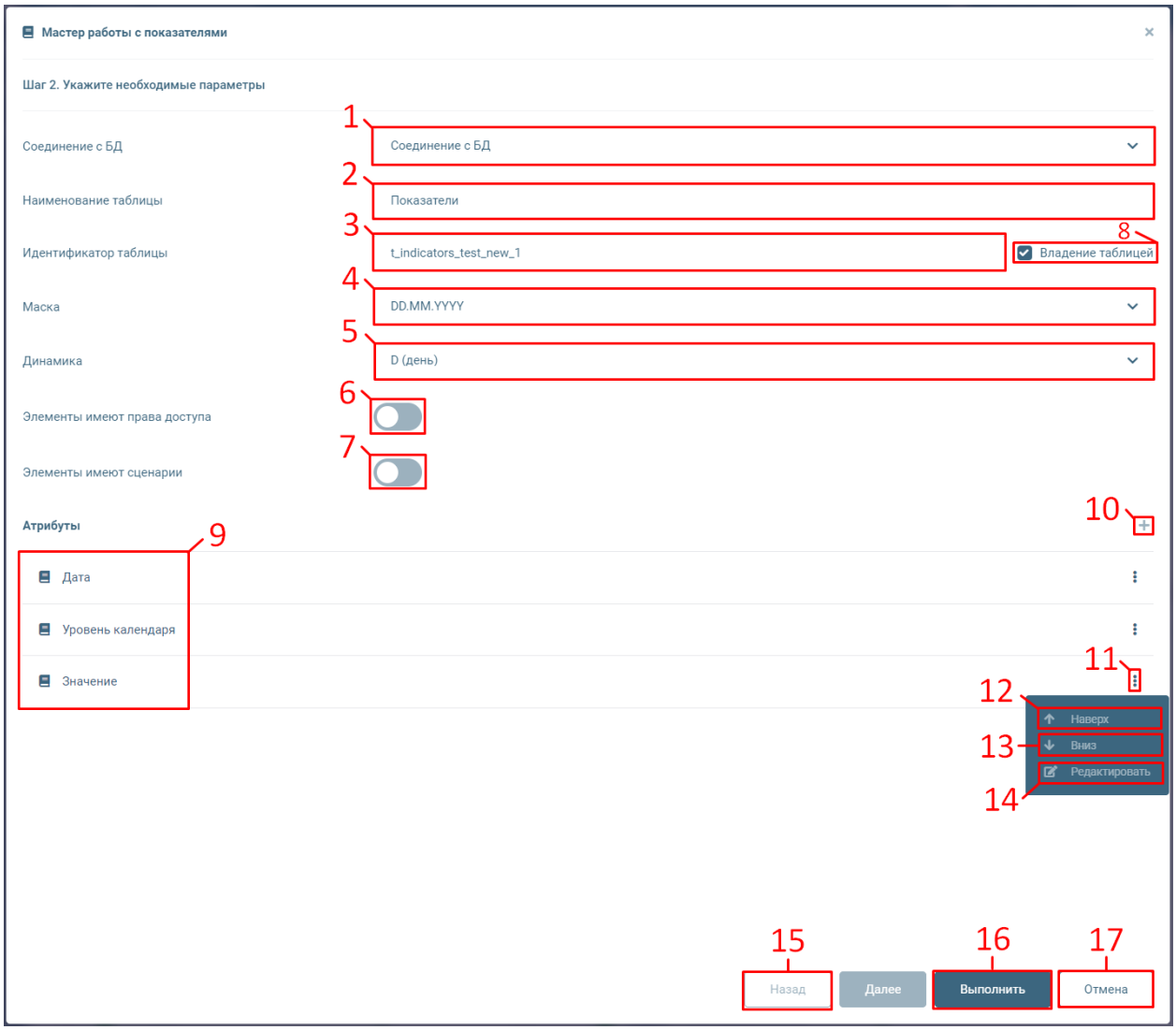


Рисунок 560 – Пример работы мастера (шаг 2)

1. Выпадающий список «Соединение с БД». Содержит перечень блоков графа, которые производят подключение к БД. Таблица будет создана в той БД, которая указана в выбранном соединении.
2. Поле ввода наименования блока.
3. Поле «Идентификатор таблицы». С данным наименованием будет создана таблица в БД.
4. Выпадающий список «Маска». Определяет формат значений атрибута «Дата». Доступные маски:
    - YYYY;
    - MM.YYYY;
    - QQ.YYYY;
    - DD.MM.YYYY;
    - DD.MM.YYYY hh:mm;
    - DD.MM.YYYY hh:mm:ss.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

5. Выпадающий список «Динамика». Определяет значение «Уровень календаря». Доступные варианты:
    - Y (год);
    - M (месяц);
    - D (день).
6. Переключатель «Элементы имеют права доступа». При включении автоматически добавляется атрибут «Права доступа».

Рисунок 561 – Пример настройки прав доступа элементов

7. Флаг «Владение таблицей». Автоматически включен при создании новой таблицы. Если таблица уже существует, то флаг выключен и не может быть включен. «Владение таблицей» по отношению к одной таблице может быть только у одного блока «Показатели».
8. Список созданных атрибутов.
9. Кнопка добавления нового атрибута.
10. Кнопка открытия контекстного меню атрибута.
11. Кнопка «Наверх». Изменяет порядок атрибута: сдвигает на 1 позицию наверх.
12. Кнопка «Вниз». Изменяет порядок атрибута: сдвигает на 1 позицию вниз.
13. Кнопка «Редактировать». Открывает окно настройки атрибута (подробнее в соответствующем разделе ниже).
14. Кнопка «Назад». Возвращает на предыдущий шаг мастера.
15. Кнопка «Выполнить». Осуществляет сохранение блока «Показатели» с текущими настройками.
16. Кнопка «Отмена». Осуществляет выход из мастера настройки показателей.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

7.8.4.4 Окно «Настройки атрибута»

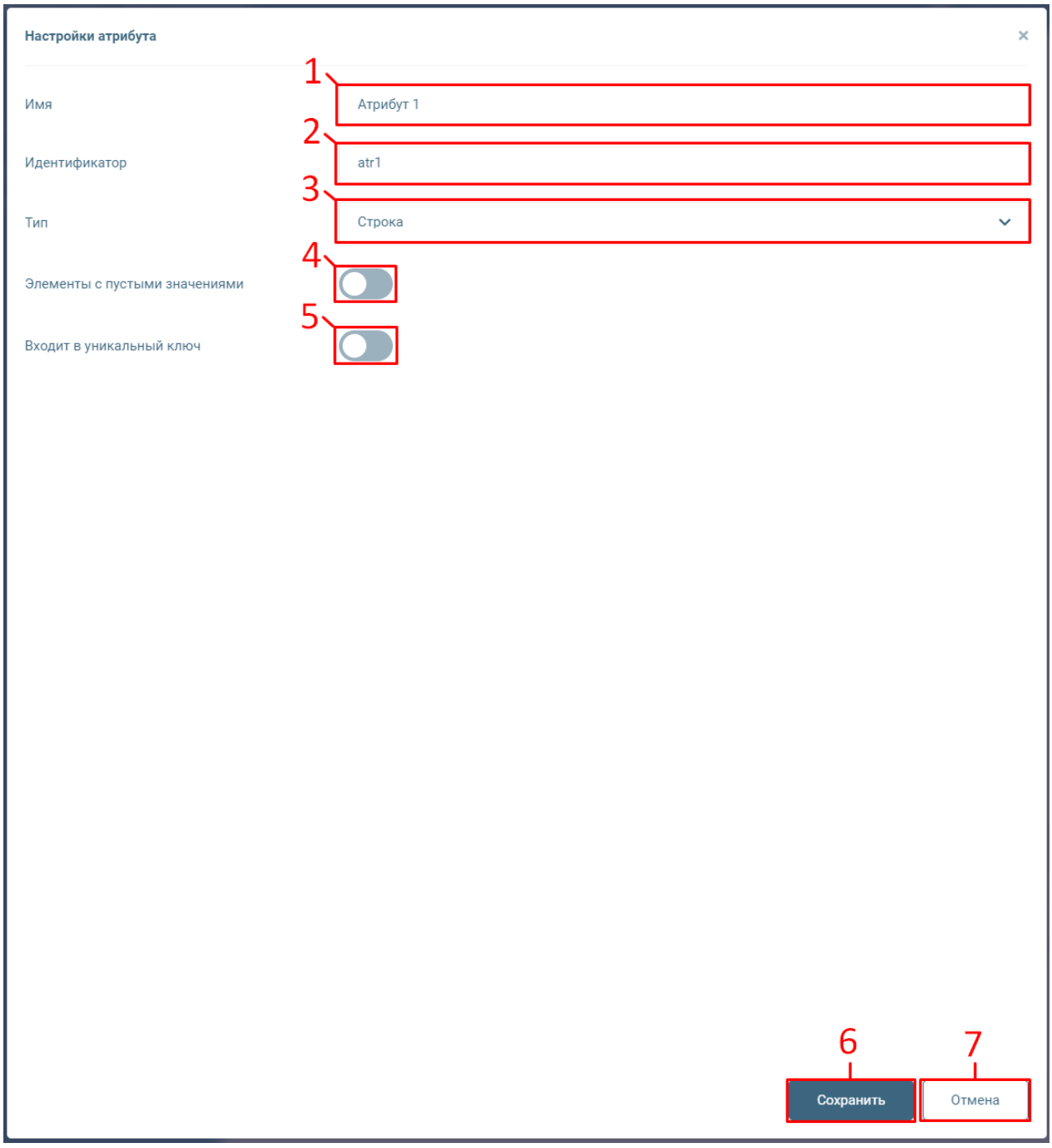


Рисунок 562 – Пример работы мастера (настройки атрибута)

1. Поле ввода наименования атрибута.
2. Поле ввода идентификатора атрибута (допускается ввод только с использованием латиницы).
3. Выпадающий список с выбором типа атрибута. Доступные типы:
   - Строка;
   - Дата. При выборе данного типа появляются дополнительные параметры: «Значение по умолчанию», «Маска», «Динамика».

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

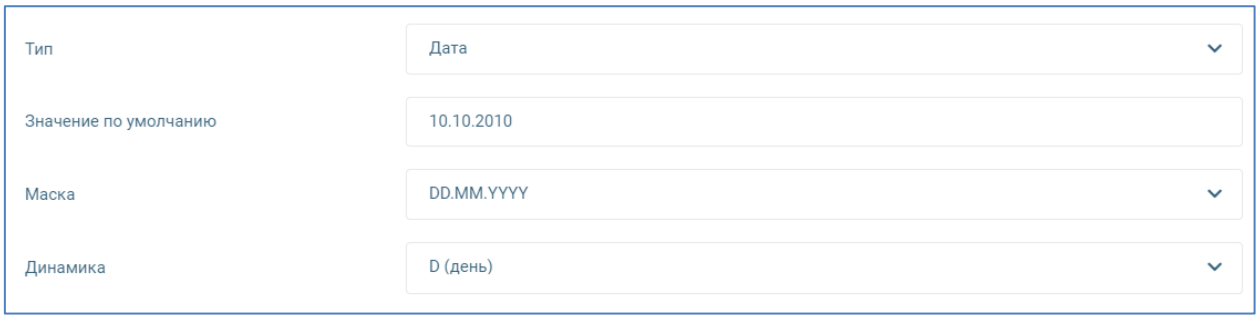


Рисунок 563 – Пример настройки атрибута с типом «Дата»

- Целочисленный;
- Вещественный;
- Координаты;
- JSON;
- Справочник. При выборе данного типа появляется дополнительный параметр «Справочник», в котором из выпадающего списка можно выбрать другой справочник графа.

Рисунок 564 – Пример настройки атрибута с типом «Справочник»

4. Переключатель «Элементы с пустыми значениями».
   - Если включен, то значения данного атрибута могут быть пустыми
   - Если выключен, то значения данного атрибута не могут быть пустыми
5. Переключатель «Входит в уникальный ключ». При включении атрибут добавляется в уникальный ключ.
6. Кнопка «Сохранить». Сохраняет атрибут с текущими настройками.
7. Кнопка «Отмена». Закрывает окно «Настройки атрибута» без сохранения изменений.

7.8.4.5 Редактирование данных

Чтобы открыть меню редактирования данных, нужно в контекстном меню блока «Показатели» выбрать пункт «Данные...».

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

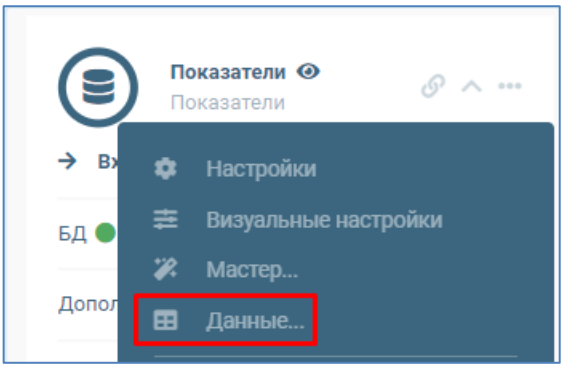


Рисунок 565 – Пример открытия мастера в настройках блока

7.8.4.6 Окно «Данные»

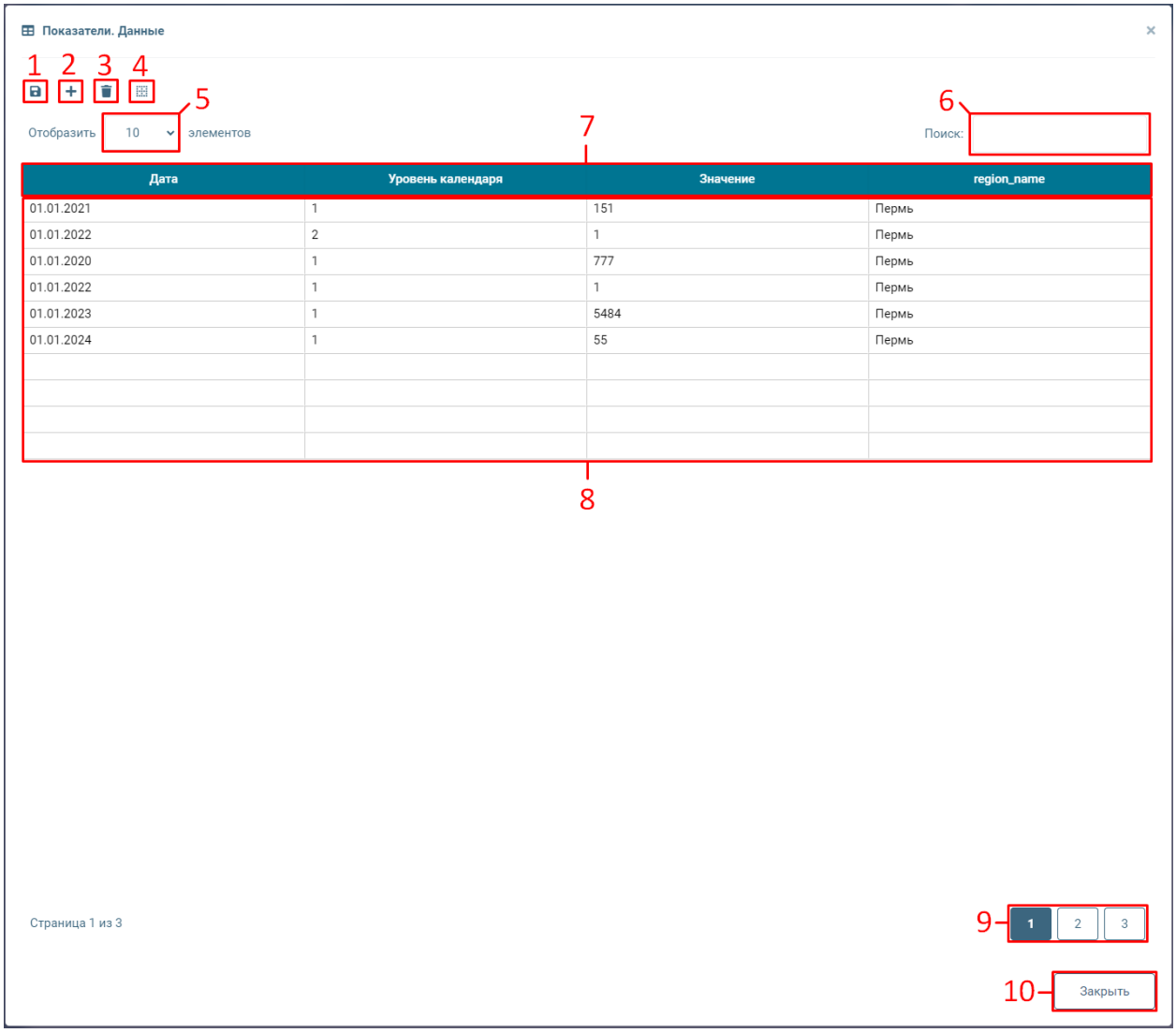


Рисунок 566 – Пример работы мастера (Данные)

1. Кнопка «Сохранить». Активна, если данные были изменены. Сохраняет изменения данных.
2. Кнопка «Добавить». Создает новую пустую строку в таблице.
3. Кнопка «Удалить». Удаляет выбранный диапазон строк. После удаления необходимо нажать кнопку «Сохранить», чтобы изменения вступили в силу.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

4. Кнопка «Отобразить данные в виде кросс-таблицы». При включении таблица перестраивается: значения атрибута «Дата» переходят в заголовки столбцов.

Уровень календаря	region_name	01.01.2020	01.01.2021	01.01.2022	01.01.2023	01.01.2024
1	Пермь	777	151	1	5484	55
2	Пермь			1		

Рисунок 567 – Пример работы мастера

5. Выпадающий список, который управляет количеством отображаемых строк в таблице на одной странице. Возможно отображать по:
   - 10 строк;
   - 25 строк;
   - 50 строк;
   - 100 строк;
6. Поле поиска. Осуществляет поиск по всей таблице с любыми совпадениями в значениях.
7. Заголовок таблицы:
   - Если кнопка «Отобразить данные в виде кросс-таблицы» выключена: в заголовке таблицы содержится список атрибутов показателей
   - Если кнопка «Отобразить данные в виде кросс-таблицы» включена: значения атрибута «Дата» переходят в заголовки столбцов
8. Строки таблицы:
   - В строках осуществляется редактирование значений.
   - Возможны как ввод с клавиатуры, так и вставка из буфера обмена.
   - Вставлять из буфера обмена можно как одно значение в ячейку, так и диапазон ячеек.
9. Страницы таблицы. Для удобства просмотра и улучшения производительности данные справочника отображаются постранично. Между страницами можно переключаться, кликнув ЛКМ на соответствующий номер.
10. Кнопка «Закрыть». Закрывает окно «Данные». Если при нажатии произведенные изменения в данных не были сохранены, то появится диалоговое окно с подтверждением сохранения.

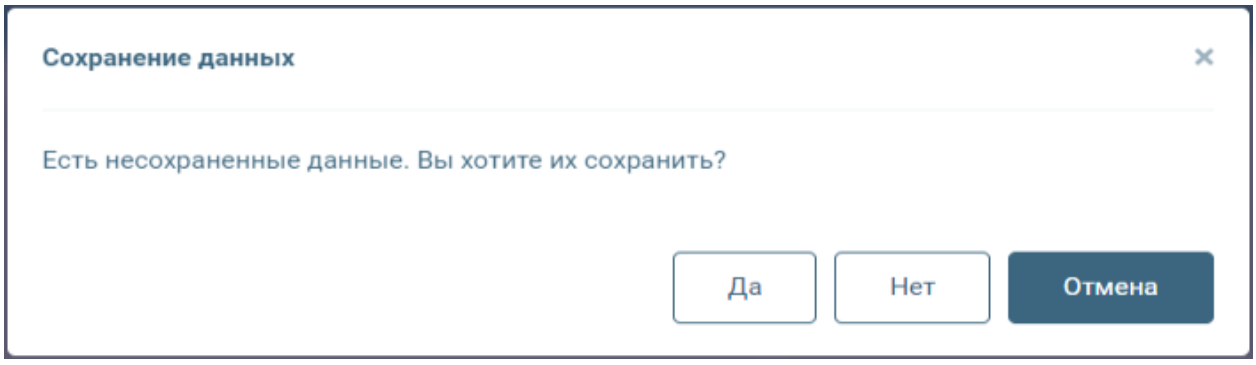


Рисунок 568 – Пример завершения работы мастера

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

7.8.5 Мастер настройки JSON показателей

7.8.5.1 Назначение

Мастер настройки JSON справочника позволяет создавать, изменять и удалять данные блока «JSON-показатели» (п. 6.2.19).

7.8.5.2 Открытие мастера

Открыть мастер можно несколькими методами:

1. Нажать на кнопку «Мастер настройки JSON показателей» в панели инструментов графа.

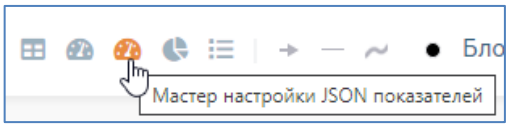


Рисунок 569 – Пример открытия мастера на панели инструментов

2. В контекстном меню блока «JSON-показатели» выбрать пункт «Мастер...».

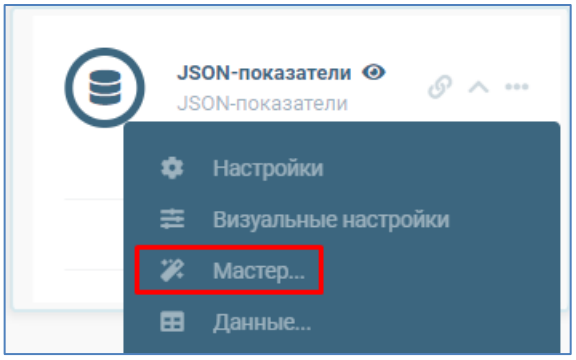


Рисунок 570 – Пример открытия мастера в настройках блока

3. Добавить блок «JSON-показатели» на граф из библиотеки блоков через механизм drag-n-drop.

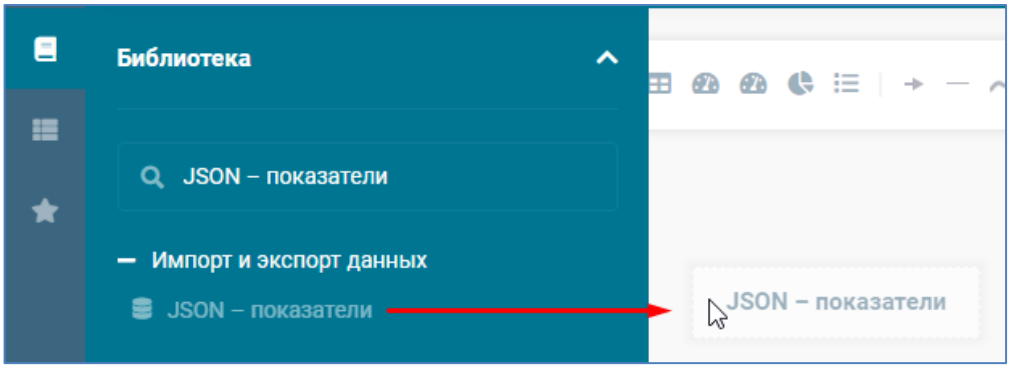


Рисунок 571 – Пример добавления блока

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

7.8.5.3 Окно работы мастера

Шаг 1

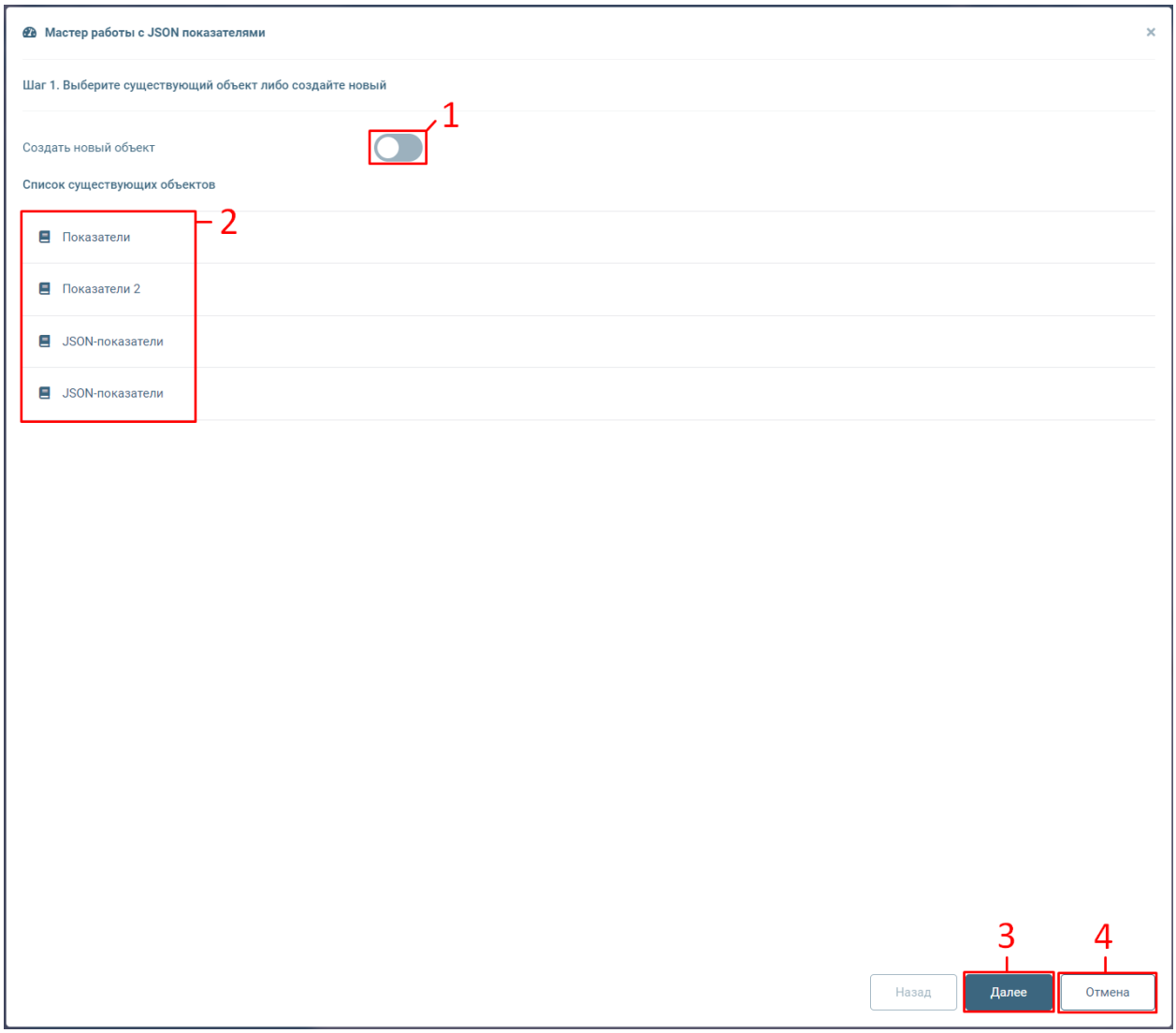


Рисунок 572 – Пример работы мастера (шаг 1)

1. Переключатель «Создать новый объект»:
    a. Если включен, то будет создан новый блок «JSON-показатели».
    b. Если выключен, то можно выбрать один из существующих блоков «JSON-показатели» для редактирования.
2. Список существующих на графе блоков «JSON-показатели».
3. Кнопка «Далее» осуществляет переход на следующий шаг мастера.
4. Кнопка «Отмена» осуществляет выход из мастера настройки JSON показатели.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Шаг 2

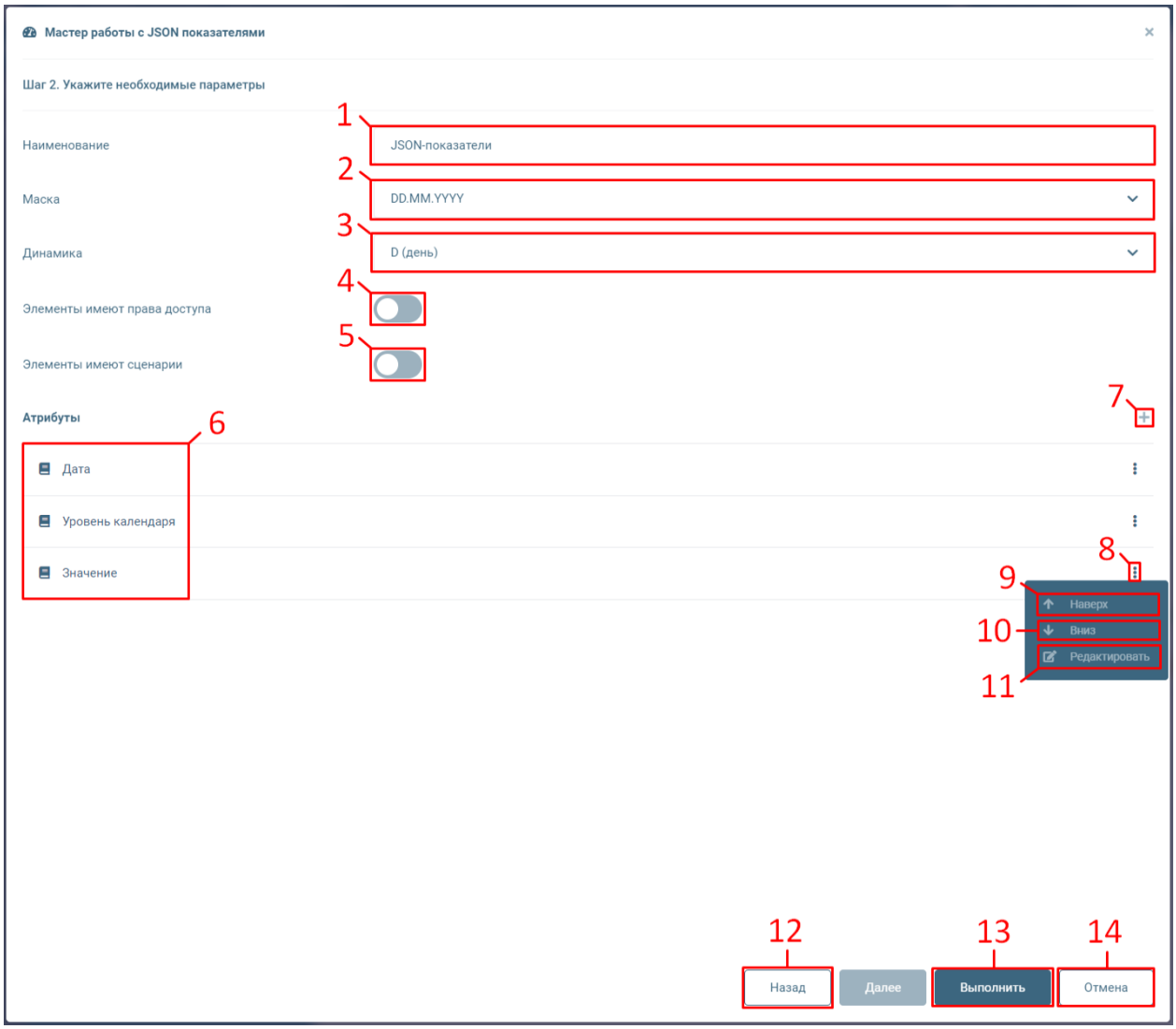


Рисунок 573 – Пример работы мастера (шаг 2)

1. Поле ввода наименования блока.
2. Выпадающий список «Маска».
3. Выпадающий список «Динамика».
4. Переключатель «Элементы имеют права доступа». При включении автоматически добавляется атрибут «Права доступа».

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Рисунок 574 – Пример настройки прав доступа элементов

5. Список созданных атрибутов.
6. Кнопка добавления нового атрибута.
7. Кнопка открытия контекстного меню атрибута.
8. Кнопка «Наверх». Изменяет порядок атрибута: сдвигает на 1 позицию наверх.
9. Кнопка «Вниз». Изменяет порядок атрибута: сдвигает на 1 позицию вниз.
10. Кнопка «Редактировать». Открывает окно настройки атрибута (подробнее в соответствующем разделе ниже)
11. Кнопка «Назад». Возвращает на предыдущий шаг мастера
12. Кнопка «Выполнить». Осуществляет сохранение блока «JSON-показатели» с текущими настройками.
13. Кнопка «Отмена». Осуществляет выход из мастера настройки JSON справочника.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

7.8.5.4 Окно «Настройки атрибута»

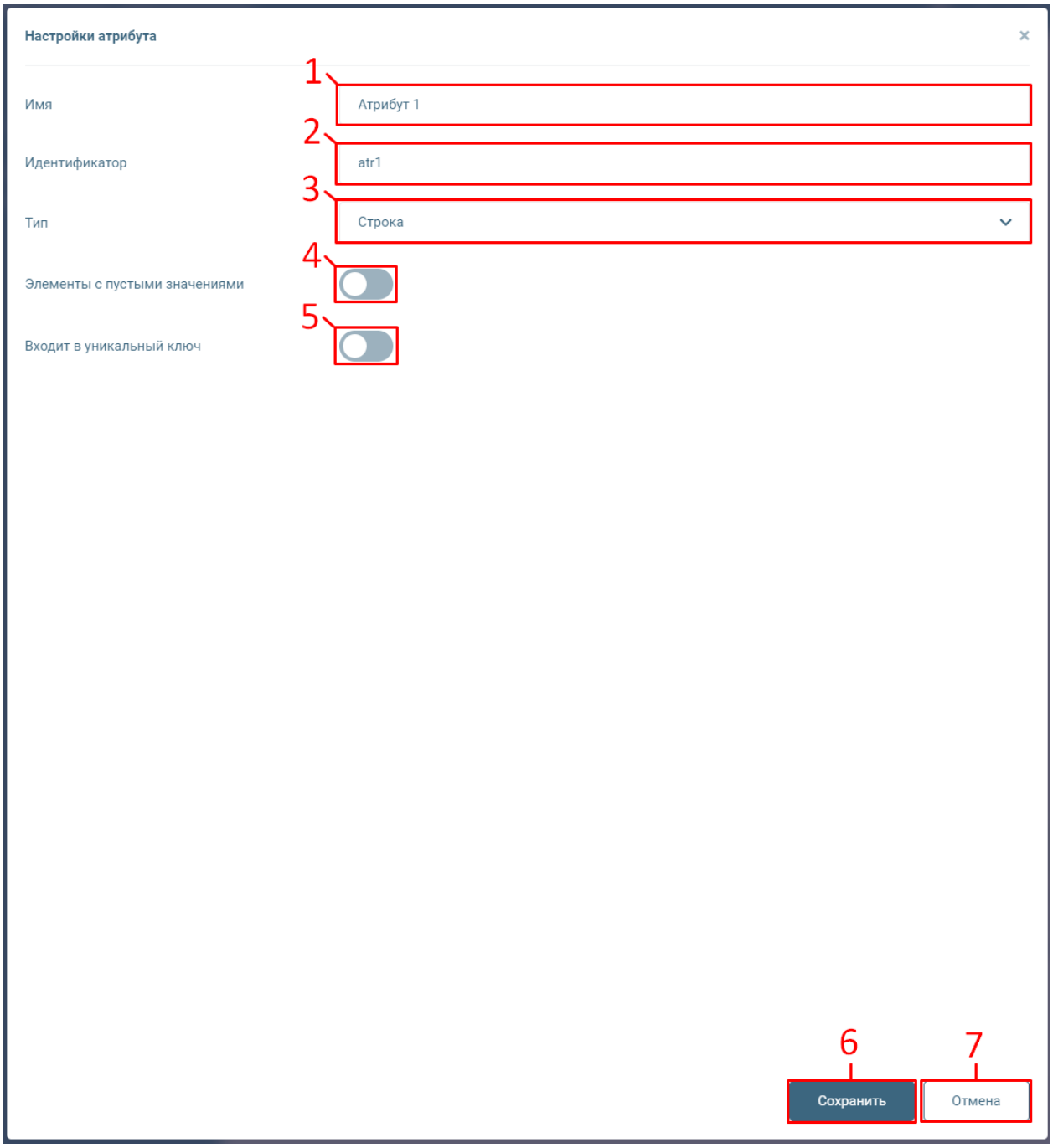


Рисунок 575 – Пример настройки атрибута

1. Поле ввода наименования атрибута.
2. Поле ввода идентификатора атрибута (допускается ввод только с использованием латиницы).
3. Выпадающий список с выбором типа атрибута. Доступные типы:
    a. Строка.
    b. Дата. При выборе данного типа появляются дополнительные параметры: «Значение по умолчанию», «Маска», «Динамика».

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

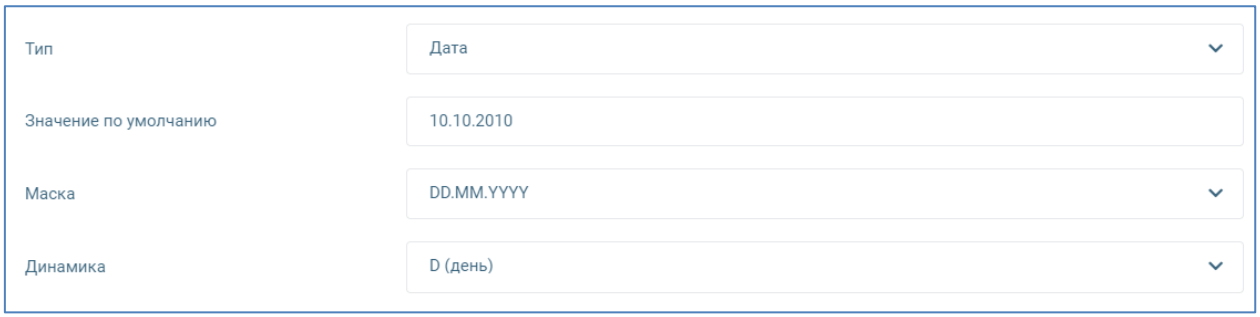


Рисунок 576 – Пример настройки атрибута с типом «Дата»

c. Целочисленный.

d. Вещественный.

e. Координаты.

f. JSON.

g. Справочник. При выборе данного типа появляется дополнительный параметр «Справочник», в котором из выпадающего списка можно выбрать другой справочник графа.

Рисунок 577 – Пример настройки атрибута с типом «Справочник»

4. Переключатель «Элементы с пустыми значениями»
   a. Если включен, то значения данного атрибута могут быть пустыми.
   b. Если выключен, то значения данного атрибута не могут быть пустыми.
5. Переключатель «Входит в уникальный ключ». При включении атрибут добавляется в уникальный ключ.
6. Кнопка «Сохранить». Сохраняет атрибут с текущими настройками
7. Кнопка «Отмена». Закрывает окно «Настройки атрибута» без сохранения изменений

7.8.5.5 Редактирование данных

Чтобы открыть меню редактирования данных, нужно в контекстном меню блока «JSON-показатели» выбрать пункт «Данные...».

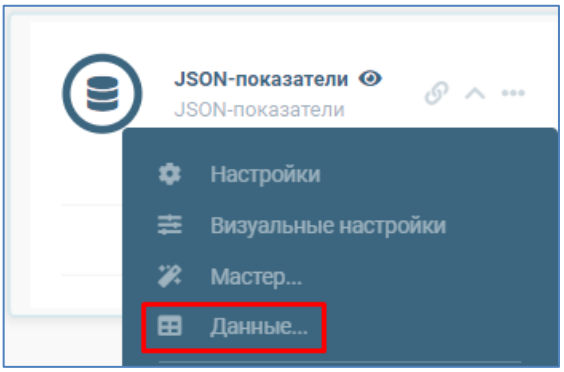


Рисунок 578 – Пример открытия мастера в настройках блока

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

7.8.5.6 Окно «Данные»

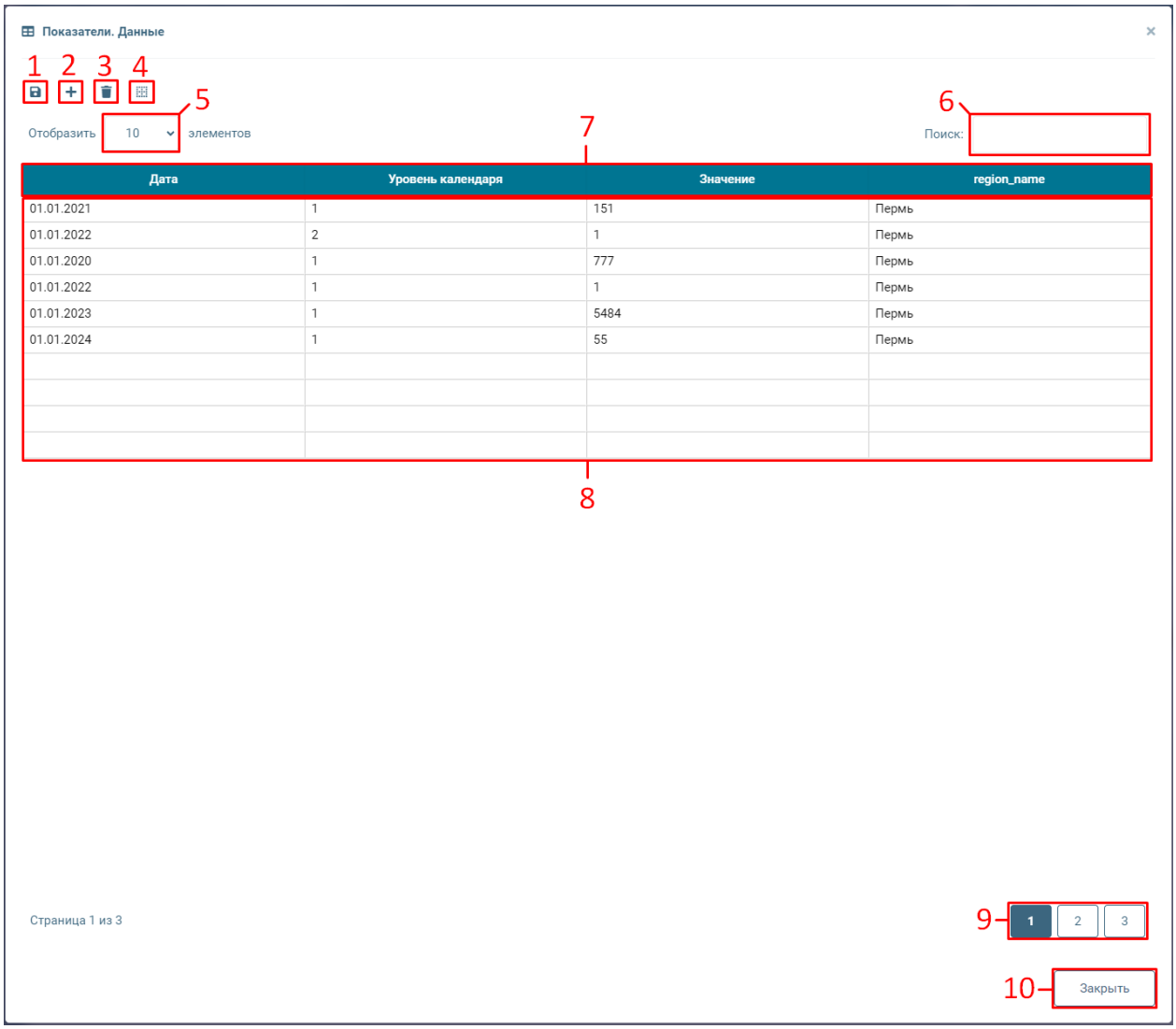


Рисунок 579 – Пример окна для редактирования данных

1. Кнопка «Сохранить». Активна, если данные были изменены. Сохраняет изменения данных.
2. Кнопка «Добавить». Создает новую пустую строку в таблице.
3. Кнопка «Удалить». Удаляет выбранный диапазон строк. После удаления необходимо нажать кнопку «Сохранить», чтобы изменения вступили в силу.
4. Кнопка «Отобразить данные в виде кросс-таблицы». При включении таблица перестраивается: значения атрибута «Дата» переходят в заголовки столбцов.

Уровень календаря	region_name	01.01.2020	01.01.2021	01.01.2022	01.01.2023	01.01.2024
1	Пермь	777	151	1	5484	55
2	Пермь			1		

Рисунок 580 – Пример предпросмотра данных

5. Выпадающий список, который управляет количеством отображаемых строк в таблице на одной странице. Возможно отображать по:
    a. 10 строк;
    b. 25 строк;

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

c. 50 строк;

d. 100 строк.

6. Поле поиска. Осуществляет поиск по всей таблице с любыми совпадениями в значениях.
7. Заголовок таблицы:
    a. Если кнопка «Отобразить данные в виде кросс-таблицы» выключена: в заголовке таблицы содержится список атрибутов JSON-показателей.
    b. Если кнопка «Отобразить данные в виде кросс-таблицы» включена: значения атрибута «Дата» переходят в заголовки столбцов.
8. Строки таблицы:
    a. В строках осуществляется редактирование значений.
    b. Возможны как ввод с клавиатуры, так и вставка из буфера обмена.
    c. Вставлять из буфера обмена можно как одно значение в ячейку, так и диапазон ячеек.
9. Страницы таблицы. Для удобства просмотра и улучшения производительности данные справочника отображаются постранично. Между страницами можно переключаться, кликнув ЛКМ на соответствующий номер.
10. Кнопка «Закрыть». Закрывает окно «Данные». Если при нажатии произведенные изменения в данных не были сохранены, то появится диалоговое окно с подтверждением сохранения.

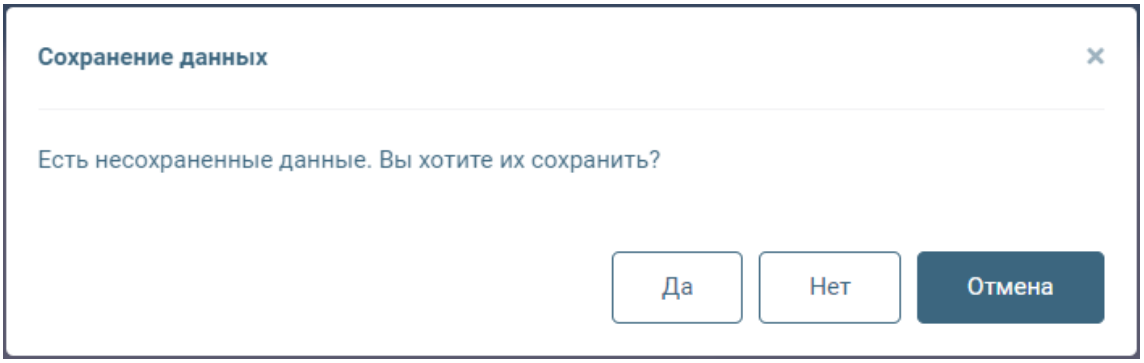


Рисунок 581 – Пример завершения работы мастера

7.8.6 Мастер настройки расширения атрибутов

7.8.6.1 Назначение

Мастер настройки расширения атрибутов позволяет выбрать блок «Показатели» (см. п. 6.2.17) или «Справочник» (см. п. 6.2.16), к которому будет добавляться атрибут, и указать какой атрибут необходимо добавить, а также из какой таблицы БД.

7.8.6.2 Открытие мастера

Открыть мастер можно несколькими методами:

1. Нажать на кнопку «Мастер настройки расширения атрибутов» в панели инструментов графа.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

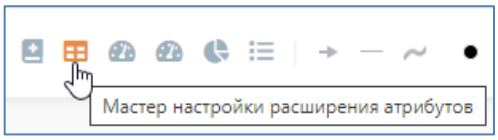


Рисунок 582 – Пример открытия мастера на панели инструментов

2. В контекстном меню блока «Расширение атрибутов временных рядов» выбрать пункт «Мастер...».

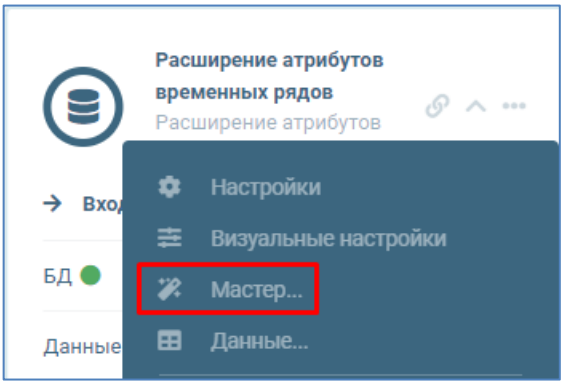


Рисунок 583 – Пример открытия мастера в настройках блока

3. Добавить блок «Расширение атрибутов временных рядов» на граф из библиотеки блоков через механизм drag-n-drop.

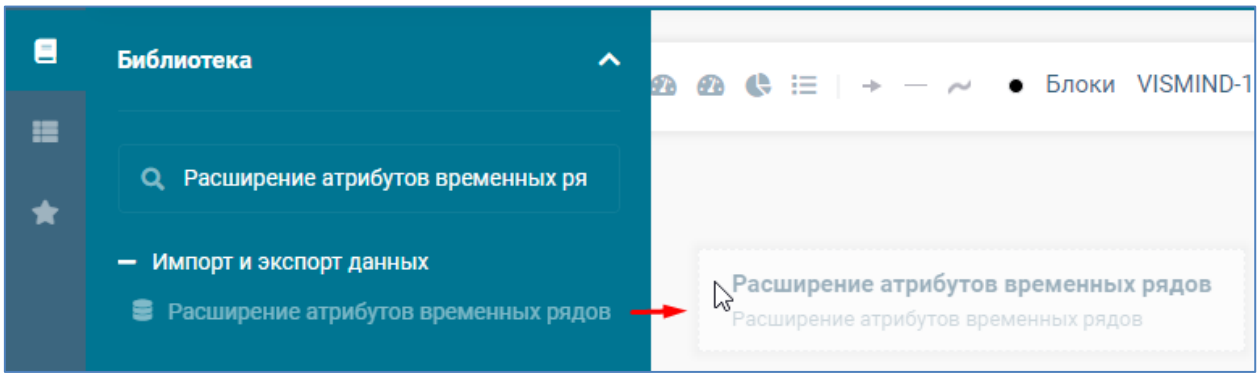


Рисунок 584 – Пример добавления блока

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

7.8.6.3 Окно работы мастера

Шаг 1

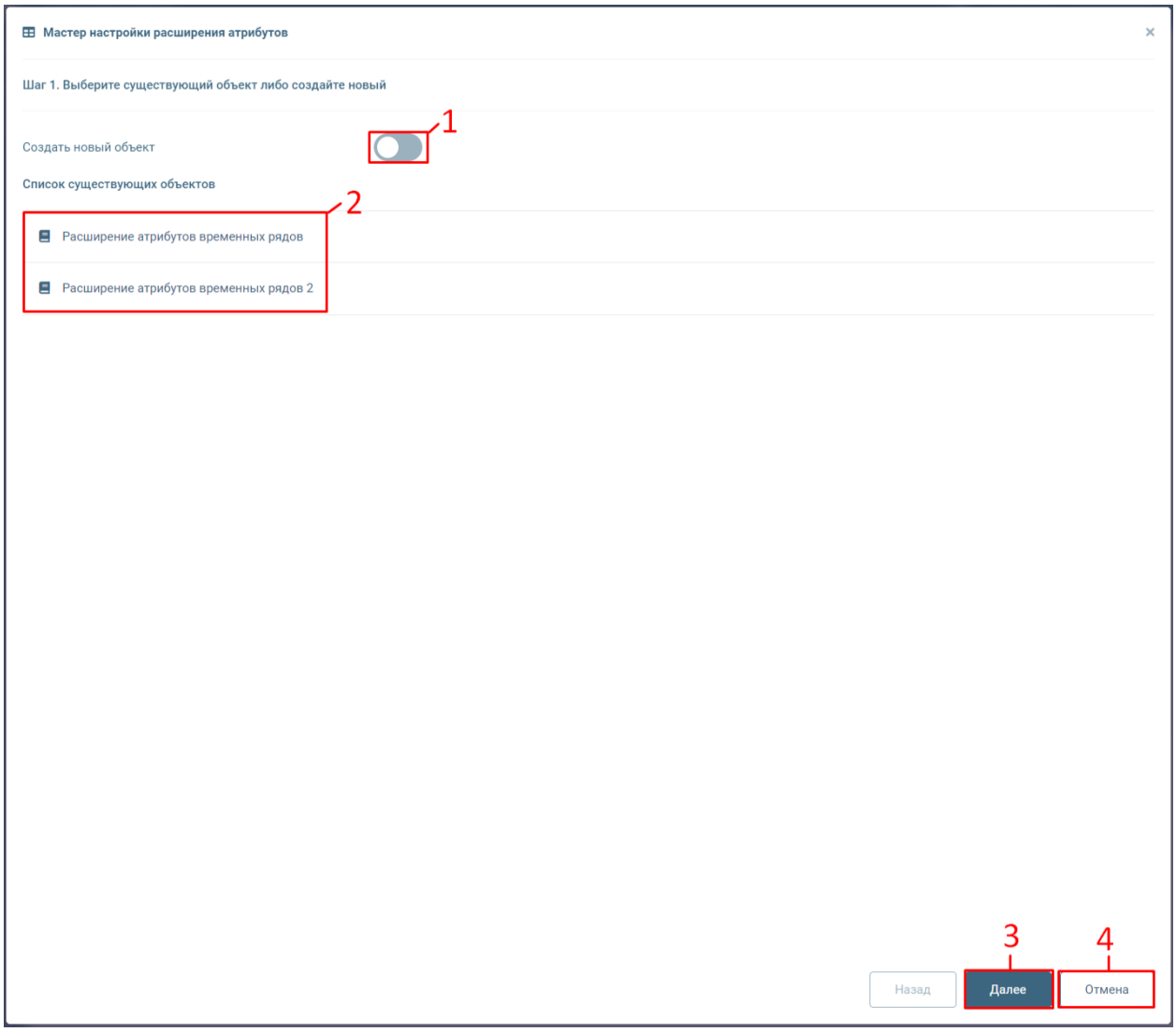


Рисунок 585 – Пример работы мастера (шаг 1)

1. Переключатель «Создать новый объект».
   - Если включен, то будет создан новый блок «Расширение атрибутов временных рядов».
   - Если выключен, то можно выбрать один из существующих блоков «Расширение атрибутов временных рядов» для редактирования.
2. Список существующих на графе блоков «Расширение атрибутов временных рядов».
3. Кнопка «Далее». Осуществляет переход на следующий шаг мастера.
4. Кнопка «Отмена». Осуществляет выход из мастера настройки расширения атрибутов.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Шаг 2

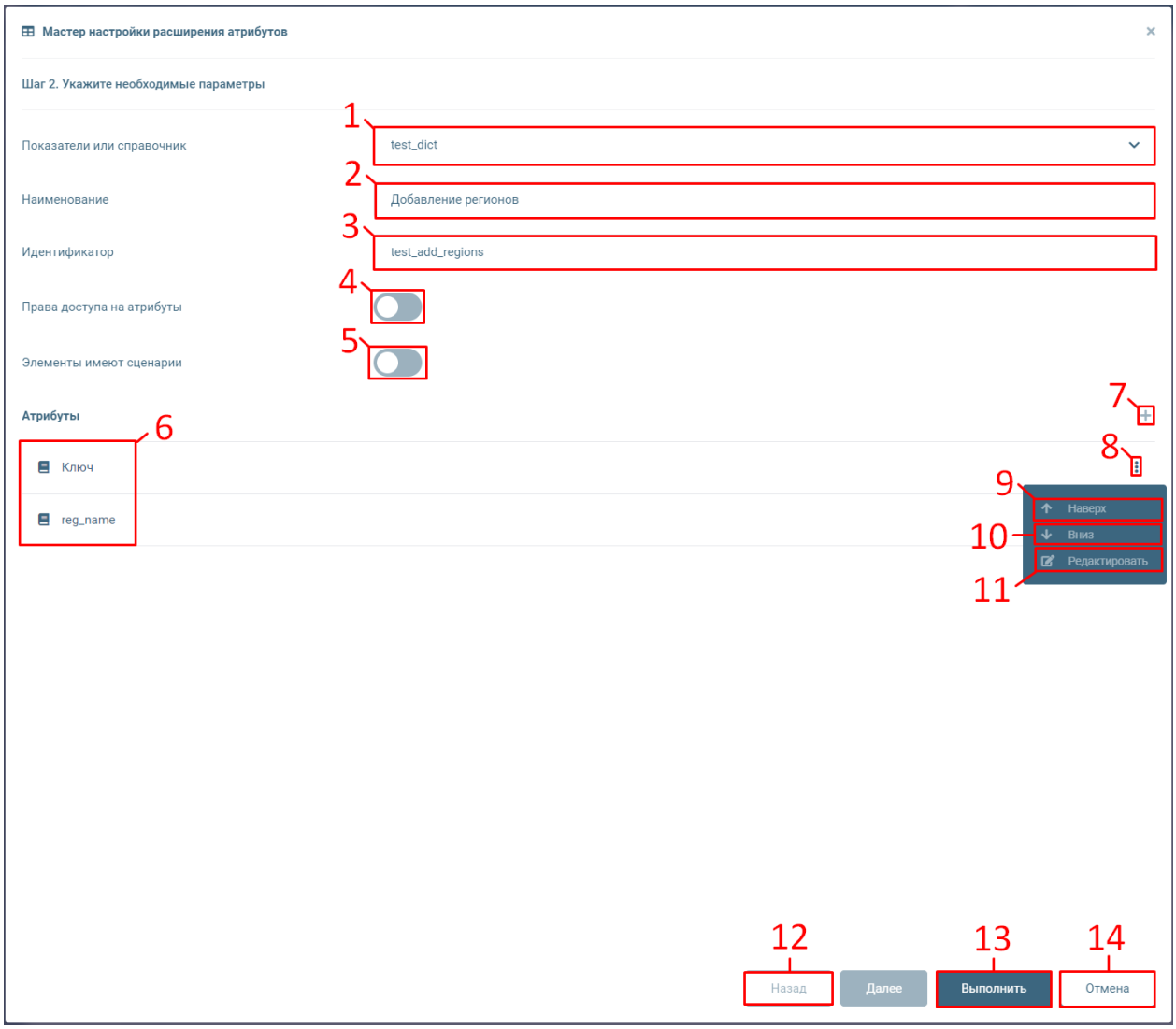


Рисунок 586 – Пример работы мастера (шаг 2)

1. Выпадающий список «Показатели или справочник». Содержит список блоков «Показатели» и «Справочник» текущего графа.
2. Поле ввода наименования блока.
3. Поле ввода «Идентификатор». В данное поле необходимо ввести имя таблицы в БД, атрибут которой хотим добавить.
4. Переключатель «Элементы имеют права доступа». Не поддерживается в блоке «Расширение атрибутов временных рядов».
5. Список созданных атрибутов.
6. Кнопка добавления нового атрибута. Добавлять необходимо атрибут, который содержится в таблице БД, указанной в поле «Идентификатор».
7. Кнопка открытия контекстного меню атрибута.
8. Кнопка «Наверх». Изменяет порядок атрибута: сдвигает на 1 позицию наверх.
9. Кнопка «Вниз». Изменяет порядок атрибута: сдвигает на 1 позицию вниз.
10. Кнопка «Редактировать». Открывает окно настройки атрибута (подробнее в соответствующем разделе ниже).

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

11. Кнопка «Назад». Возвращает на предыдущий шаг мастера.
12. Кнопка «Выполнить». Осуществляет сохранение блока «Расширение атрибутов временных рядов» с текущими настройками.
13. Кнопка «Отмена». Осуществляет выход из мастера настройки настройки расширения атрибутов.

7.8.6.4 Окно «Настройки атрибута»

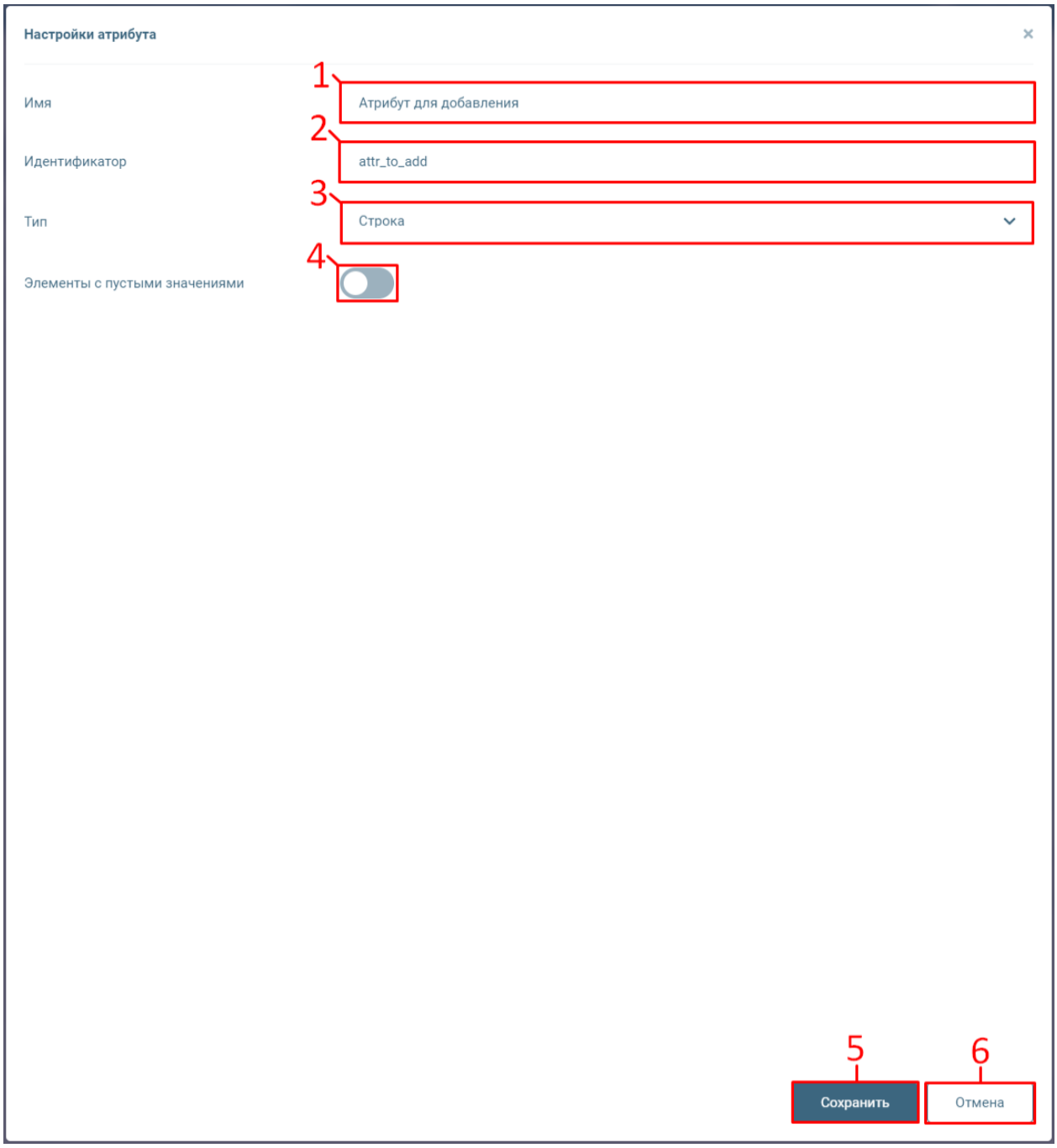


Рисунок 587 – Пример работы мастера (настройка атрибута)

1. Поле ввода наименования атрибута.
2. Поле ввода идентификатора атрибута (допускается ввод только с использованием латиницы).

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

3. Выпадающий список с выбором типа атрибута. Доступные типы:
    - Строка.
    - Дата. При выборе данного типа появляются дополнительные параметры: «Значение по умолчанию», «Маска», «Динамика».

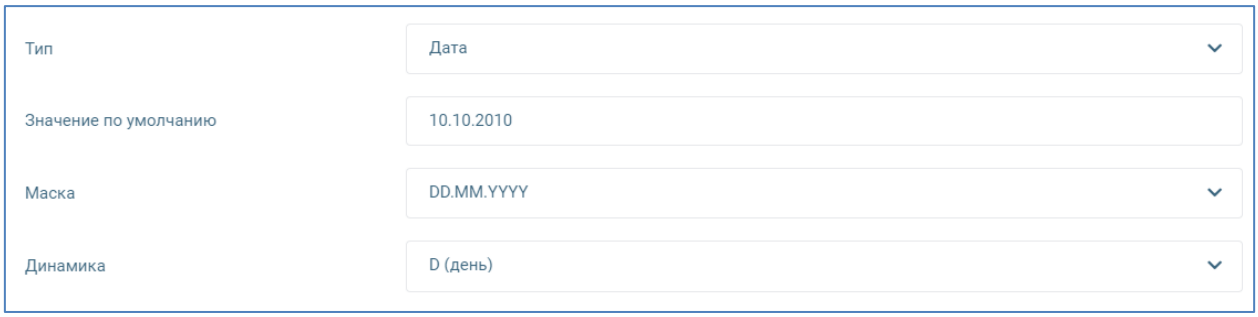


Рисунок 588 – Пример настройки атрибута с типом «Дата»

    - Целочисленный.
    - Вещественный.
    - Координаты.
    - JSON.
    - Справочник. Неактуально для блока «Расширение атрибутов временных рядов».
4. Переключатель «Элементы с пустыми значениями».
    - Если включен, то значения данного атрибута могут быть пустыми.
    - Если выключен, то значения данного атрибута не могут быть пустыми.
5. Кнопка «Сохранить». Сохраняет атрибут с текущими настройками.
6. Кнопка «Отмена». Закрывает окно «Настройки атрибута» без сохранения изменений.

7.8.7 Мастер настройки справочника

Мастер настройки справочника позволяет создать или обновить данные блока «Справочник» (см. п. 6.2.16).

7.8.7.1 Открытие мастера

Открыть мастер можно несколькими методами:

1. Открытие мастера через верхнее меню:
    a. На верхней панели управления щелкнуть по иконке справочника.

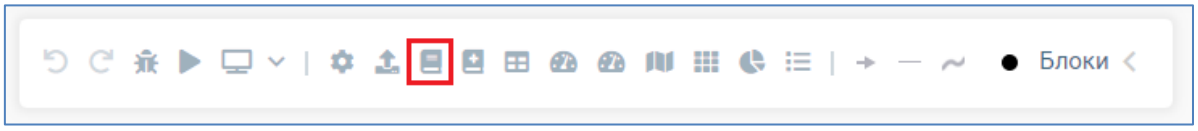


Рисунок 589 – Пример открытия мастера на панели инструментов

    b. В открывшимся меню выбрать справочник для редактирования, либо переключить флаг «**Создать новый справочни**к» для создания нового справочника, после чего нажать на кнопку «**Далее**».

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

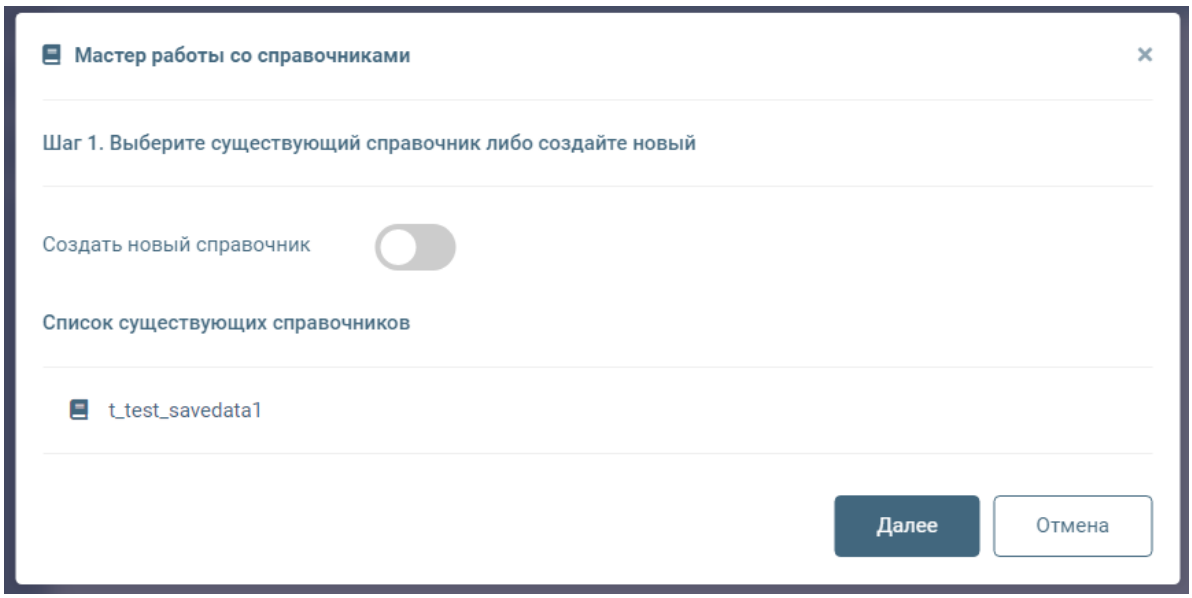


Рисунок 590 – Пример работы мастера (шаг 1)

2. Открытие мастера через настройки блока:
    a. Открыть настройки блока и выбрать там пункт «**Мастер**».

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

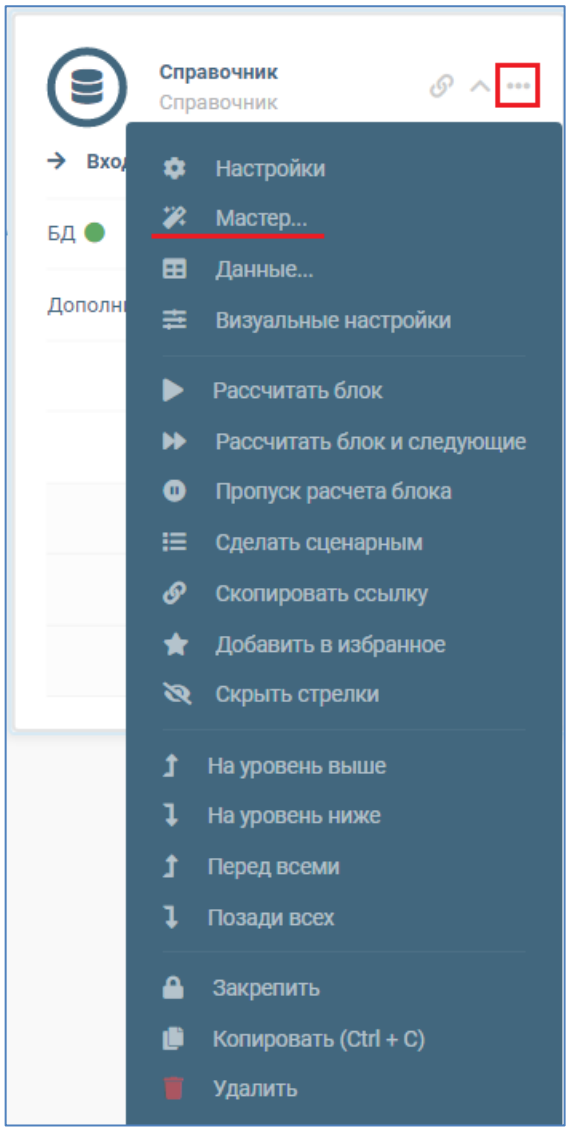


Рисунок 591 – Пример открытия мастера через настройки блока

3. Перетаскивание нового блока на граф (drag-n-drop):
    a. В **библиотеке** блоков найти блок «Справочник»;
    b. Зажать левую кнопку мыши и перетащить блок на граф:

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

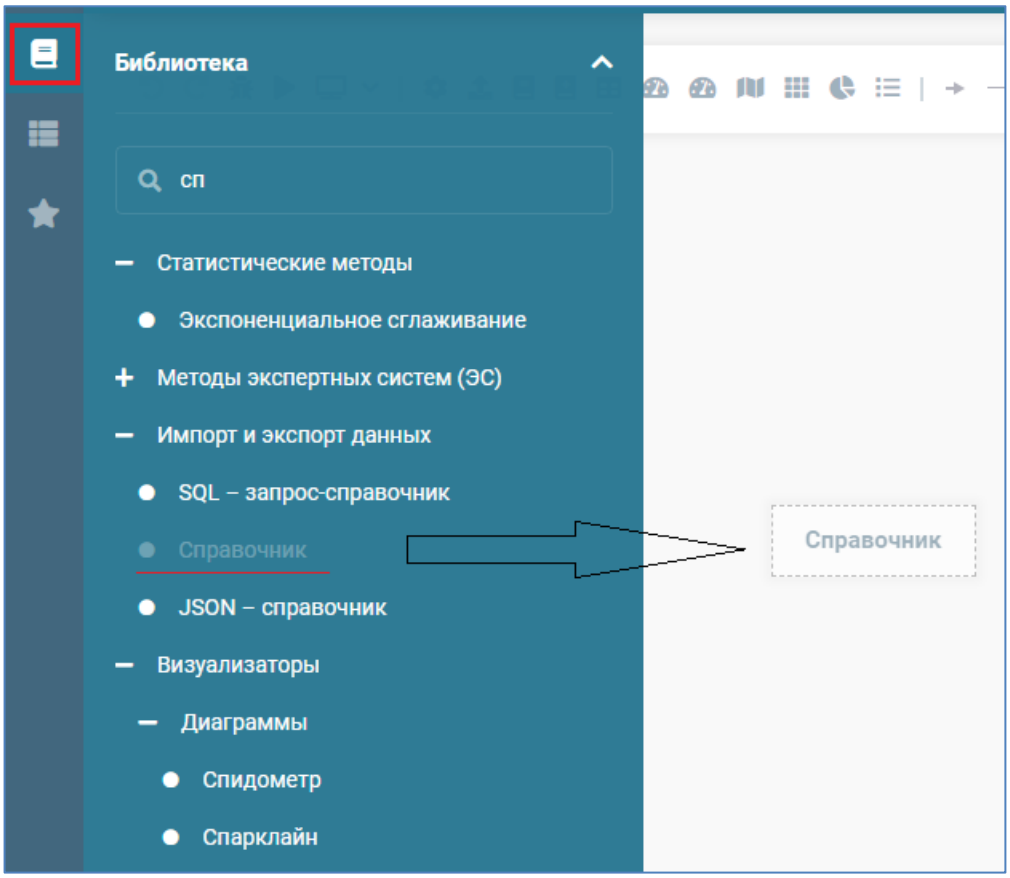


Рисунок 592 – Пример добавления блока

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

7.8.7.2 Окно работы мастера

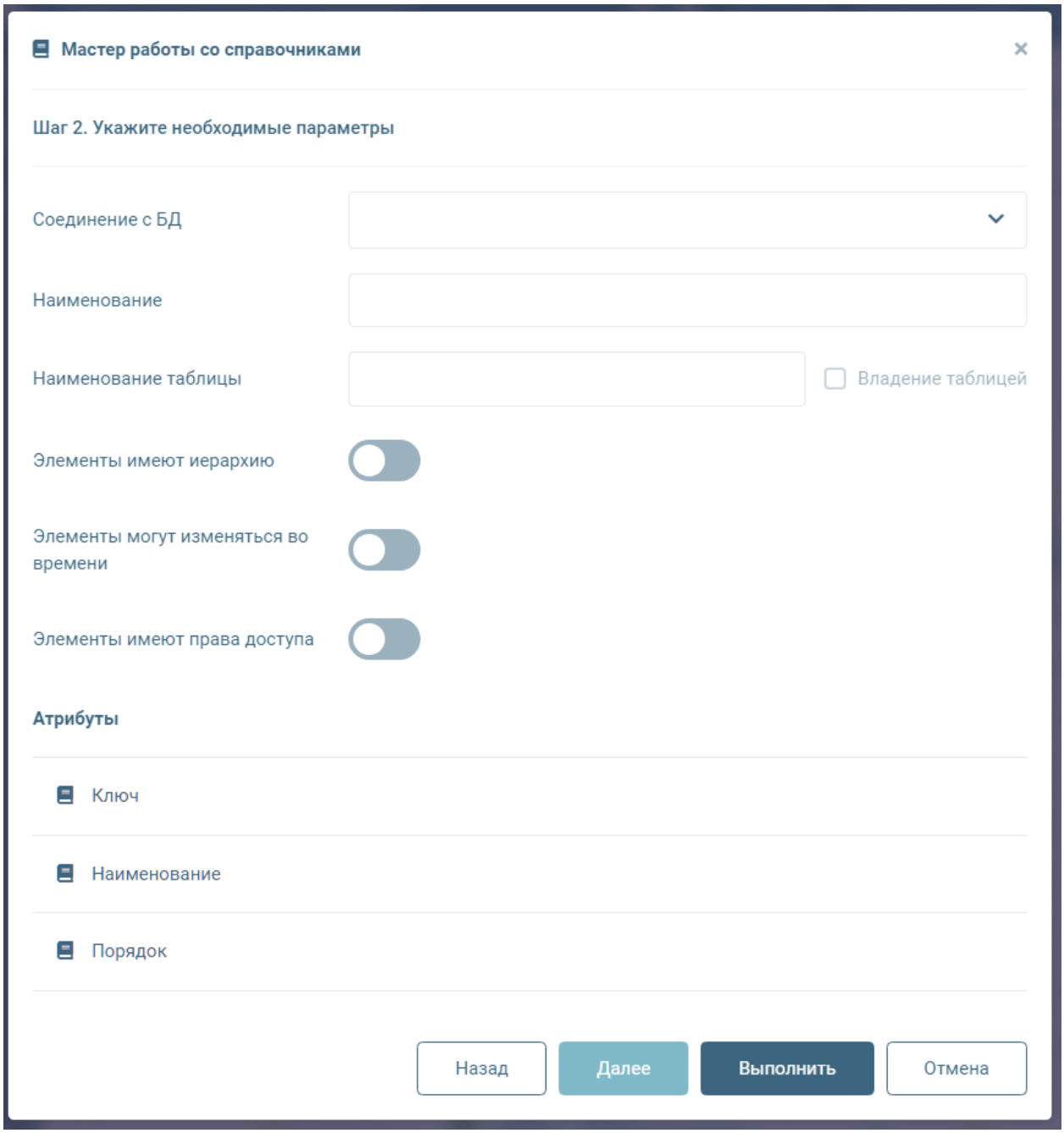


Рисунок 593 – Пример работы мастера (шаг 2)

- **Соединение с БД** – выбор базы данных из блоков-соединений, которые есть на графе.
- **Наименование** – наименование блока.
- **Наименование таблицы** – наименование таблицы, которая будет создана в базе данных.
    - **Владение таблицей** – флаг, позволяющий редактировать таблицу: добавлять, обновлять или удалять столбцы в таблице.
        - По умолчанию включен для всех новых таблиц (если таблица с таким наименованием не найдена в базе данных).

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

- У всех скопированных справочников данный флаг отключен, пока существует справочник родитель. Если исходный справочник был удален, то появляется возможность включить данный флаг.

Для корректной работы с атрибутами справочника рекомендуется всегда создавать справочник на новой таблице.

А для извлечения данных из уже существующей таблицы, воспользоваться блоком «SQL – запрос-справочник» (п. 6.2.11).

- **Элементы имеют иерархию** – флаг, добавляющий в таблицу системные атрибуты для работы с иерархическим справочником.
- **Элементы могут изменяться во времени** – флаг, добавляющий в таблицу системные атрибуты для возможности работы с версионностью данных.

Функционал не используется в текущей версии приложения.

- **Элементы имеют права доступа** – флаг, добавляющий в таблицу системные атрибуты для разделения прав доступа к таблице.

Функционал не используется в текущей версии приложения.

- **Атрибуты** – список всех атрибутов справочника. Совпадает с полями таблицы в базе данных.
    - Редактирование доступно при включенном флаге «**Владение таблицей**».
    - По умолчанию, в каждом справочнике присутствуют три системных атрибута:

Не рекомендуется изменять или удалять системные атрибуты. Без них корректная работа блока будет невозможна.

- **Ключ**, поле в таблице **ID** – уникальный идентификатор записи.
- **Наименование**, поле в таблице **NAME** – наименование элемента.
- **Порядок**, поле в таблице **ORD** – порядок сортировки элементов.

За порядок сортировки данных всегда отвечает это поле. Возможности изменения способа сортировки, на текущий момент, нет.

7.8.7.3 Редактирование атрибутов блока

При редактировании атрибутов блока так же редактируются поля в таблице в базе данных.

- Для добавления нового атрибута необходимо нажать значок «+», справа от надписи «**Атрибуты**»:

Рисунок 594 – Пример добавления нового атрибута

- Для редактирования атрибута необходимо нажать на три вертикальные точки, справа от наименования атрибута:

Рисунок 595 – Пример редактирования атрибута

После нажатия на добавление или редактирование атрибутов, откроется окно с настройками атрибута.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

В данном примере открыто окно с добавлением атрибута, при редактировании атрибута часть полей будет заполнена значениями.

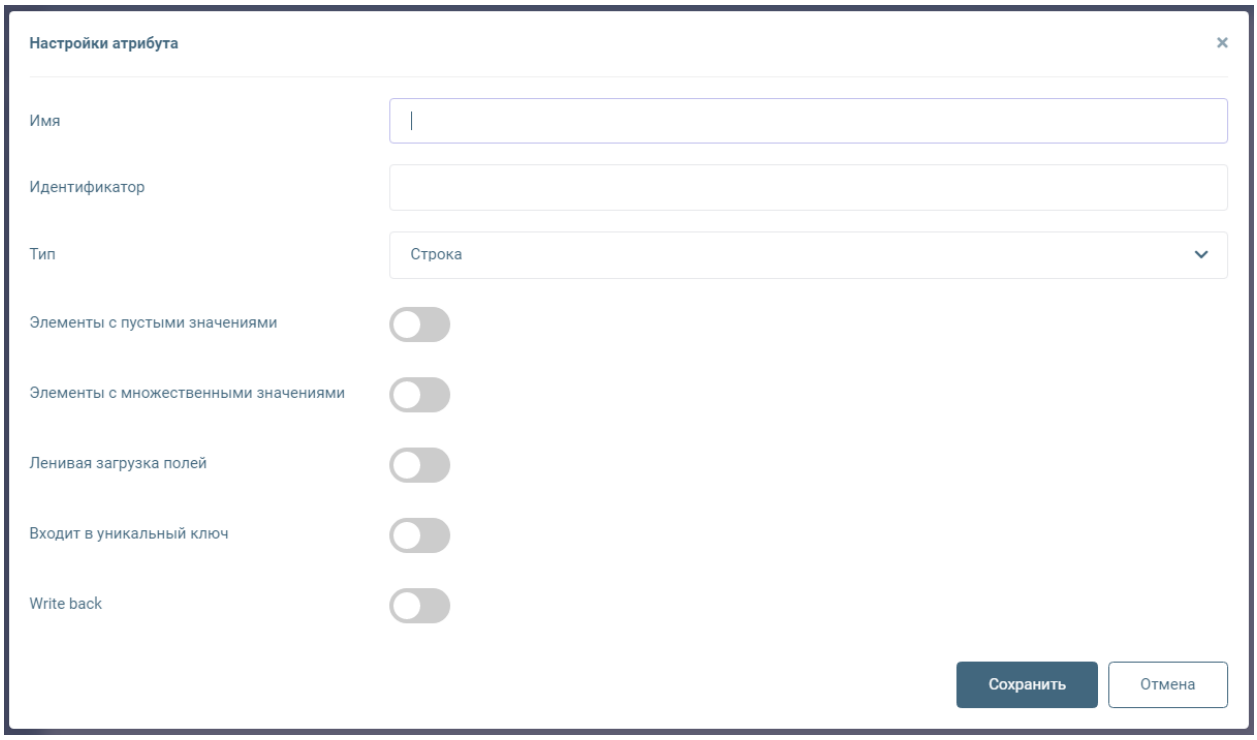


Рисунок 596 – Пример работы мастера (настройки атрибута)

- **Имя** – наименование атрибута, которое будет использоваться для наименования столбца в Мастере данных блока (см. п. 7.8.8).
- **Идентификатор** – ключ атрибута, по которому он будет доступен в таблице в базе данных.

Формат наименования может различаться, в зависимости от выбранной базы данных, но в общем случае рекомендуется использовать латинские буквы (a-Z), цифры (0-9) и знак подчеркивания (_).

- **Тип** – тип атрибута, возможные значения:
  - **Строка** – строковый тип данных;
  - **Дата** – значение даны. При выборе этого типа появляются дополнительные настройки:
    - **Значение по умолчанию** – значение даты по умолчанию;
    - **Маска** – маска даты для отображения в данных;
    - **Динамика** – динамика даты;
  - **Целочисленный** – целое число;
  - **Вещественный** – вещественное число;
  - **Координаты** – строковый тип данных, заточенный для распознавания координат;
  - **JSON** – строковый тип данных, в котором хранятся значения в формате JSON;
  - **Справочник** – возможность связывания одного справочника с другими. В качестве ключа используется поле ID.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

- **Элементы с пустыми значениями** – возможность сохранять атрибут без введённых значений.
- **Элементы с множественными значениями** – возможность записывать несколько значений в один атрибут.

Функционал не используется в текущей версии приложения.

- **Ленивая загрузка полей** – флаг, позволяющий при расчете блока произвести «ленивую загрузку» – сохранить в системную таблицу, а вместо значения прописать уникальный идентификатор поля.

Не все блоки могут преобразовать «ленивую загрузку» обратно в значение.

- **Входит в уникальный ключ** – флаг, отвечающий за вхождение поля в уникальный ключ таблицы (и в первичных ключ объектов).
- **Write-back** – возможность изменения значения поля при использовании механизма write-back (изменение и сохранение данных).

7.8.8 Мастер данных блока

Мастер данных блока позволяет просматривать и изменять данные без использования дополнительных визуализаторов.

Мастер доступен для разных типов блоков, однако его функционал (за некоторыми исключениями, указанными ниже) работает одинаково.

Для открытия мастера данных в настройках блока необходимо выбрать пункт «**Данные**»:

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

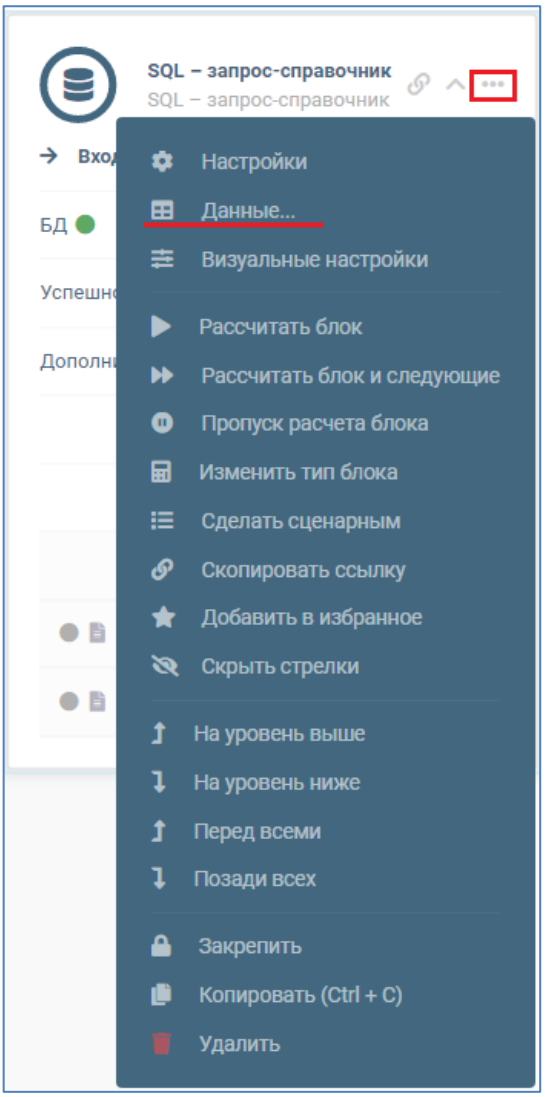


Рисунок 597 – Пример открытия мастера данных в настройках блока

После чего откроется мастер данных блока.

Максимальное количество просматриваемых записей – не более 1000 строк.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

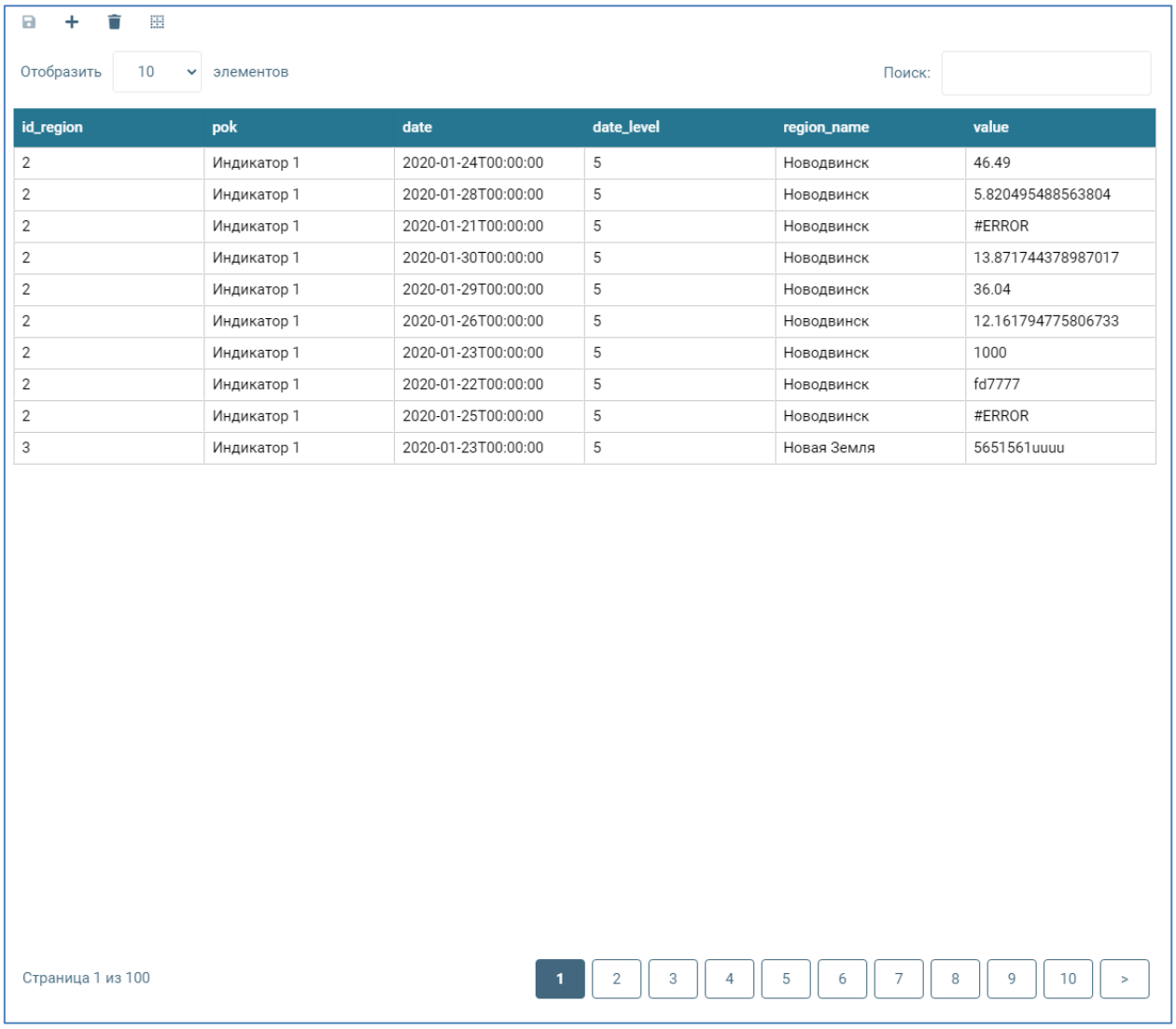


Рисунок 598 – Пример работы мастера

Настройка «**Отобразить**» позволяет выбрать количество элементов, отображаемых на одной странице. Возможные значения: 10, 25, 50 и 100 элементов.

Строка «**Поиск**» позволяет искать значения по всем атрибутам элементов.

Верхнее меню позволяет сохранять измененные данные, добавлять или удалять записи, а также просматривать данные в виде кросс-таблицы:

Рисунок 599 – Пример верхнего меню

1. **Сохранение** позволяет сохранить измененные данные.

Для SQL-блоков сохранение данных возможно только при указании таблицы для сохранения данных и всех используемых атрибутов (используется механизм write-back).

Для блока «Справочник» не обязательно указывать поля **Ключ** и **Порядок**. При их отсутствии они заполнятся автоматически.

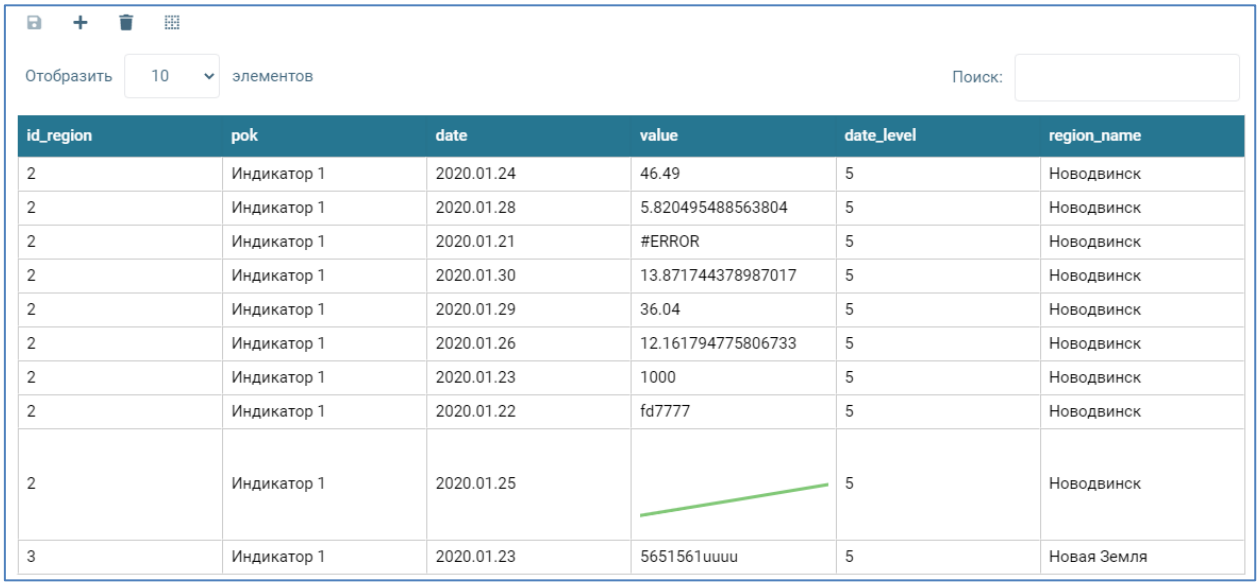
2. **Добавление** данных позволяет добавить пустую строку в конец таблицы.
3. **Удаление** данных позволяет удалить выбранную строку.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

4. **Отображение** данных в виде кросс-таблицы позволяет изменить отображение данных, вынеся все ключи рядов в шапку таблицы.

Функция доступна только для блоков «SQL-запрос для рядов», «Показатели» и «JSON-Показатели».

Отображение данных без включенной настройки:

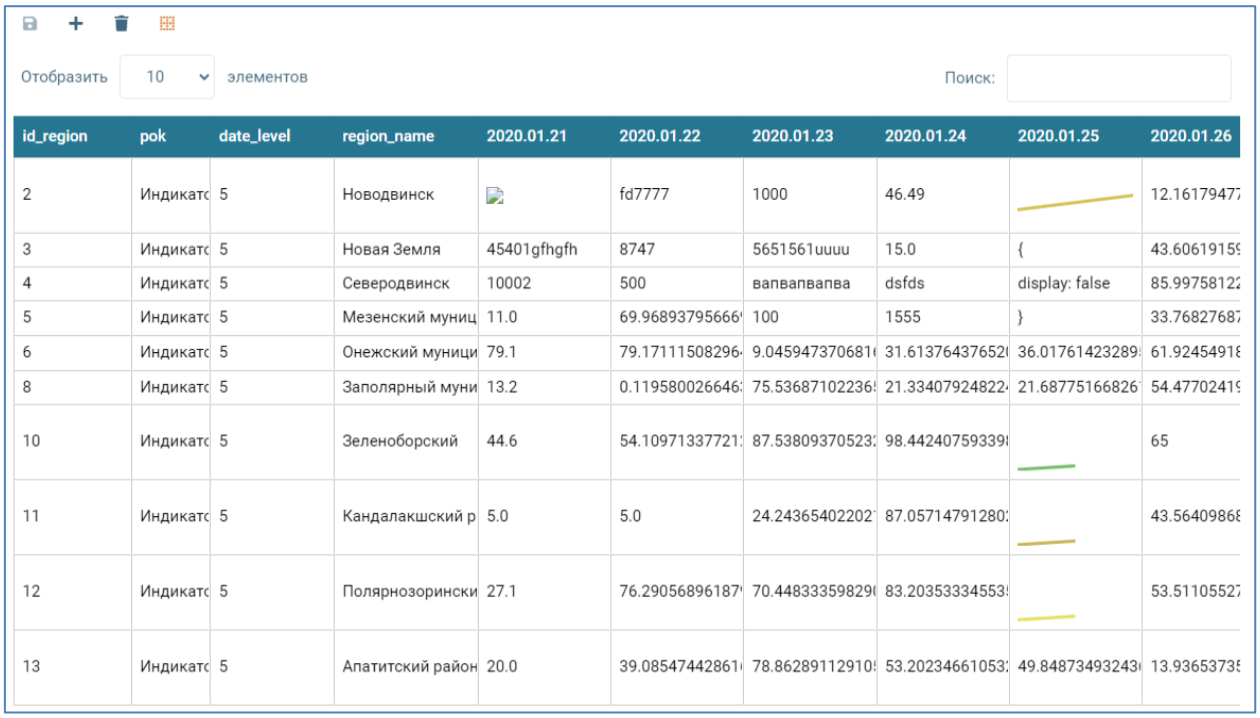


Отобразить 10 элементов Поиск:

id_region	pok	date	value	date_level	region_name
2	Индикатор 1	2020.01.24	46.49	5	Новодвинск
2	Индикатор 1	2020.01.28	5.820495488563804	5	Новодвинск
2	Индикатор 1	2020.01.21	#ERROR	5	Новодвинск
2	Индикатор 1	2020.01.30	13.871744378987017	5	Новодвинск
2	Индикатор 1	2020.01.29	36.04	5	Новодвинск
2	Индикатор 1	2020.01.26	12.161794775806733	5	Новодвинск
2	Индикатор 1	2020.01.23	1000	5	Новодвинск
2	Индикатор 1	2020.01.22	fd7777	5	Новодвинск
2	Индикатор 1	2020.01.25		5	Новодвинск
3	Индикатор 1	2020.01.23	5651561uuuu	5	Новая Земля

Рисунок 600 – Пример отображения данных без включенной настройки

Отображение данных с включенной настройкой (поле **date** являлось ключом ряда, поэтому его значения были вынесены в шапку таблицы, под которыми в строках расположились значения поля **value**):



Отобразить 10 элементов Поиск:

id_region	pok	date_level	region_name	2020.01.21	2020.01.22	2020.01.23	2020.01.24	2020.01.25	2020.01.26
2	Индикатс	5	Новодвинск		fd7777	1000	46.49		12.16179477
3	Индикатс	5	Новая Земля	45401gfhgfh	8747	5651561uuuu	15.0	{	43.60619159
4	Индикатс	5	Северодвинск	10002	500	вапвапвапва	dsfds	display: false	85.99758122
5	Индикатс	5	Мезенский муниц	11.0	69.96893795666	100	1555	}	33.76827687
6	Индикатс	5	Онежский муници	79.1	79.17111508296	9.0459473706816	31.613764376520	36.01761423289	61.92454918
8	Индикатс	5	Заполярный муни	13.2	0.1195800266463	75.53687102236	21.33407924822	21.68775166826	54.47702419
10	Индикатс	5	Зеленоборский	44.6	54.10971337721	87.53809370523	98.44240759398		65
11	Индикатс	5	Кандалакшский р	5.0	5.0	24.24365402202	87.05714791280		43.56409868
12	Индикатс	5	Полярнозоринский	27.1	76.29056896187	70.44833359829	83.20353334553		53.51105527
13	Индикатс	5	Апатитский район	20.0	39.08547442861	78.86289112910	53.20234661053	49.84873493243	13.93653735

Рисунок 601 – Пример отображения данных с включенной настройкой

7.8.9 Мастер анализа временного ряда для блока «Арима»

Мастер анализа временного ряда позволяет произвести анализ ряда, который входит в блок «Арима».

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Анализ включает в себя:

- таблицу с расширенным тестом Дики-Фуллера;
- диаграммой автокорреляционной функции;
- диаграммой частной автокорреляционной функции.

Для открытия мастера в настройках блока «Арима» необходимо выбрать пункт «Анализ временного ряда»:

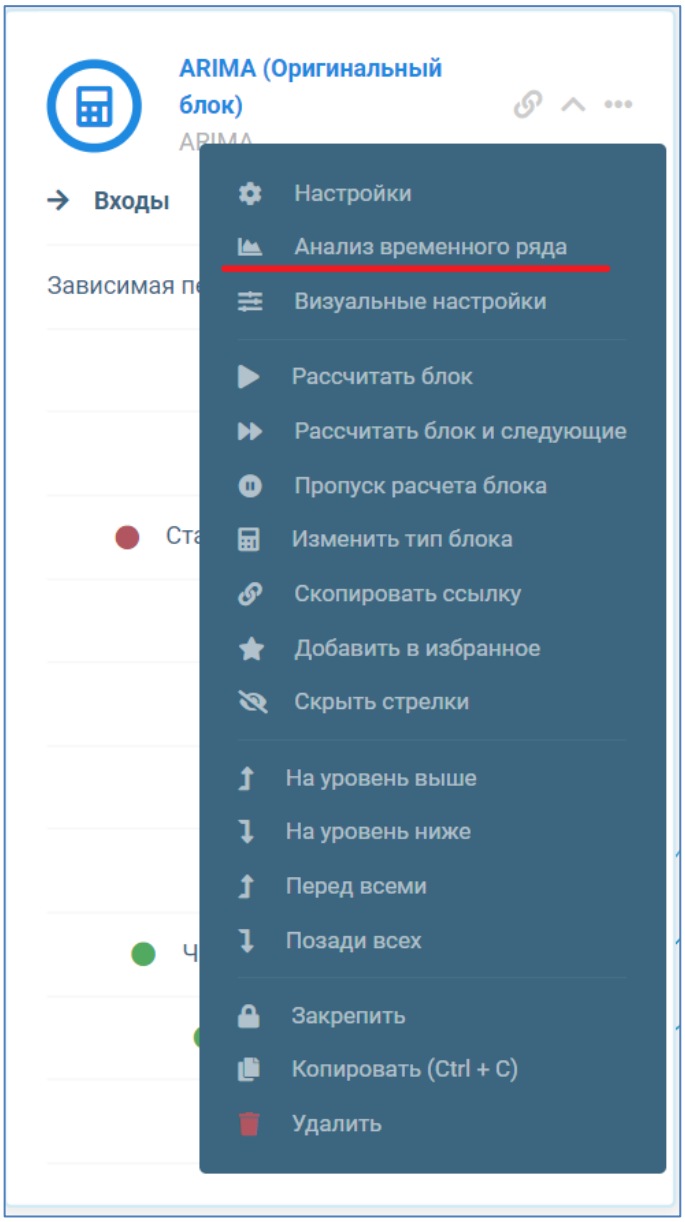


Рисунок 602 – Пример открытия мастера в настройках блока

После чего откроется мастер анализа временного ряда:

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Рисунок 603 – Пример мастера анализа временного ряда

Форма с мастером открывается со значением «Порядок дифференцирования» взятым из настроек блока «Арима» (по умолчанию значение равно «0»).

Ограничения поля «Порядок дифференцирования»:

- Значение «Порядок дифференцирования» должен быть в рамках от «0» до «5» включительно.
- При изменении параметра «Порядок дифференцирования» надо нажать кнопку «Применить», чтобы информация обновилась на форме мастера.

По окончанию анализа ряда можно:

- либо сохранить параметр «Порядок дифференцирования» нажав кнопку «Сохранить» в настройки блока «Арима» (параметр «Порядок дифференцирования» сохранится в настройках блока «Арима»).
- либо отменить действия нажав кнопку «Отмена» (параметр «Порядок дифференцирования» не сохранится в настройках блока «Арима»).

7.8.10 Мастер автоподбора максимальных/специфических порядков для блока «Арима»

Мастер позволяет произвести подбор максимального и специфического порядка для лучшей модели Аримы.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Автоподбор перебирает все возможные варианты порядков, из заданных пользователем значений. Результатом автоподбора является те порядки, при которых значение **aic** минимальное.

Мастер рассчитывает по методике оценки: Py.statsmodels

Для открытия мастера в настройках блока «Арима» необходимо выбрать пункт «Мастер автоподбора...»:

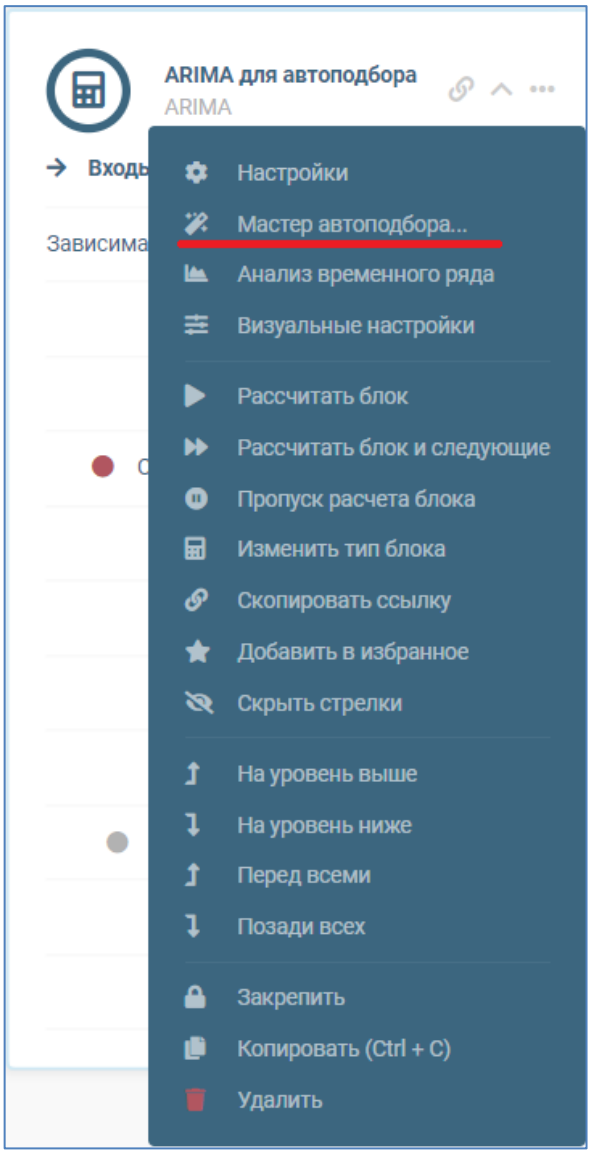


Рисунок 604 – Пример открытия мастера в настройках блока

После чего откроется мастер автоподбора максимальных/специфических порядков:

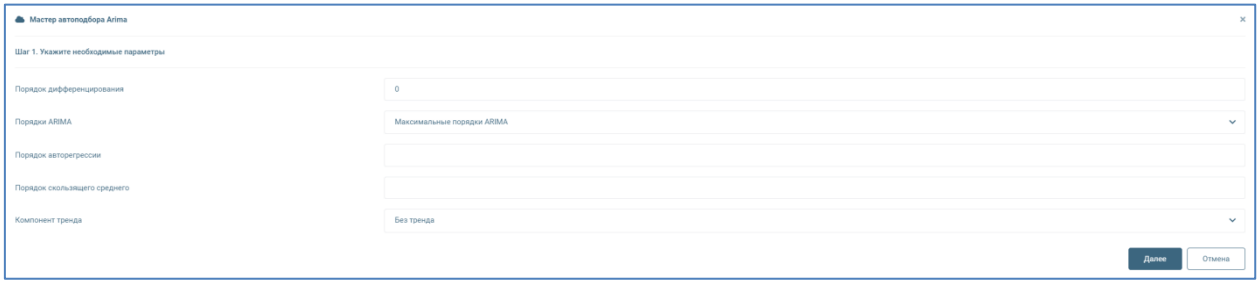


Рисунок 605 – Пример работы мастера (шаг 1)

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

На данной форме задаются параметры, с которыми будет рассчитываться автоподбор.

Если подбираются максимальные порядки ARIMA, то порядки авторегресси и скользящего среднего

Порядки авторегресии и скользящего среднего можно задать несколькими способами:

- через запятую. Например: 1,2,3,5,7;
- через дефис. Например: 1-5;
- через дефис и через запятую. Например: 1, 3-5, 7.

После того как задали параметры, нажимаем кнопку «Далее».

На следующей форме будет отображаться лог прогресса расчета автоподбора.

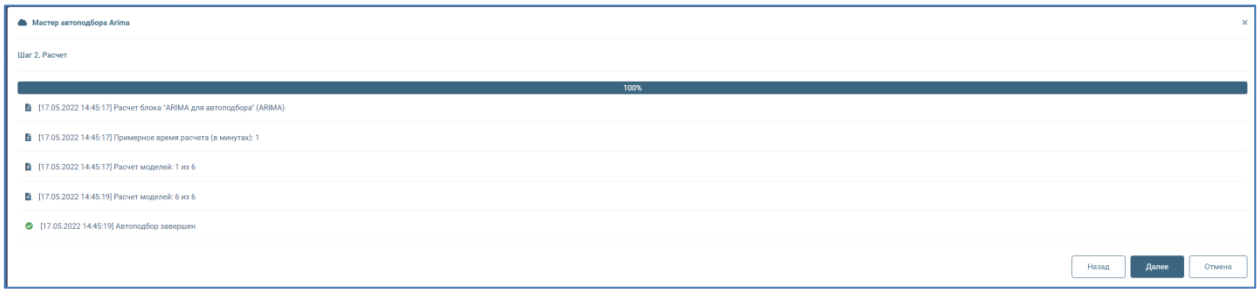


Рисунок 606 – Пример работы мастера (шаг 1)

После того как рассчитается автоподбор, нажимаем кнопку «Далее».

На данной форме выведутся первые 100 лучших вариантов автоподбора.

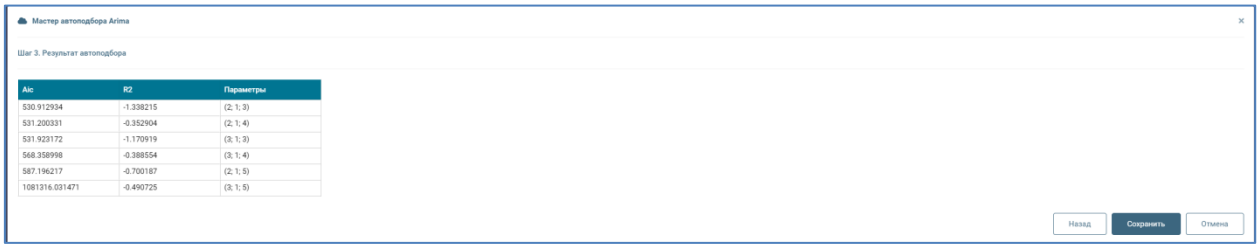


Рисунок 607 – Пример работы мастера (шаг 3)

Если нажать на кнопку «Сохранить», то лучший вариант (первый в списке), установится в настройки блока.

Значения из поля «Параметры» установятся в:

- первая – в порядок авторегресии;
- последняя – в порядок скользящего среднего.

При нажатии на кнопку «Отмена», настройки блока не изменятся, т.е. рассчитанные порядки не сохранятся.

7.8.11 Мастер настройки блока «Выходные данные – НСИ»

Мастер позволяет настроить формат выходных объектов из НСИ.

Экспериментальная функция. В расчетах не участвует.

7.8.11.1 Открытие мастера

Для открытия мастера необходимо перейти в настройки блока и выбрать там пункт «**Мастер**»:

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

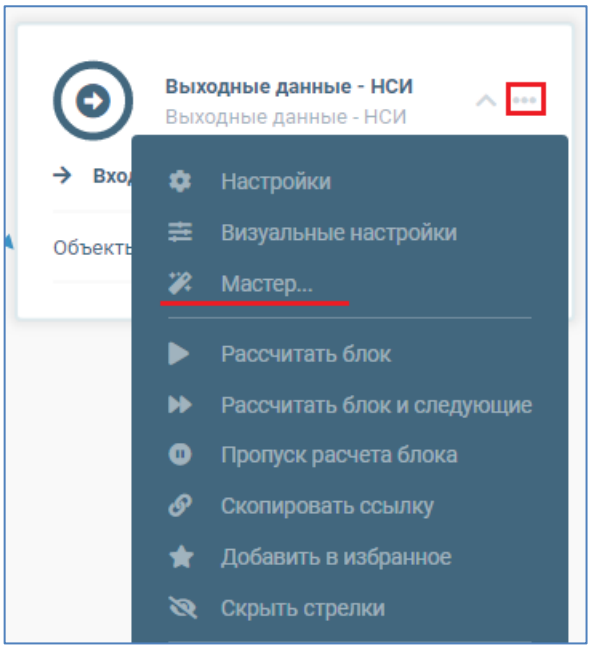


Рисунок 608 – Пример открытия мастера в настройках блока

7.8.11.2 Окно работы мастера

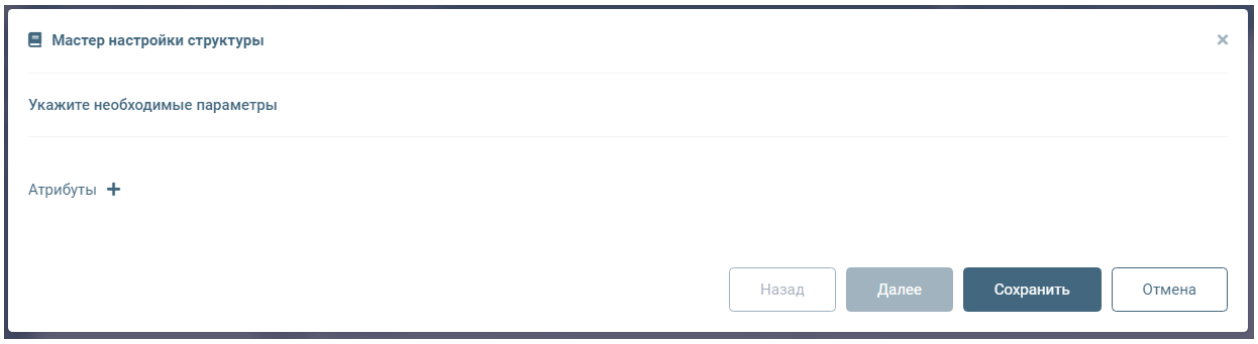


Рисунок 609 – Пример работы мастера (настройка структуры)

- **Атрибуты** – список добавленных атрибутов мастера.
  - В отличии от мастеров других блоков, к примеру Мастер настройки справочника (см. п. 7.8.7), в мастере НСИ атрибутов по умолчанию нет, т.к. все элементы могут быть изменены пользователем.

7.8.11.3 Редактирование атрибутов блока

- Для добавления нового атрибута необходимо нажать на значок «+», справа от надписи «**Атрибуты**»:

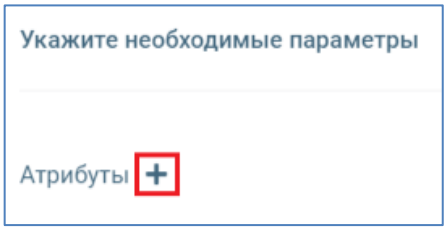


Рисунок 610 – Пример работы мастера (добавление атрибутов)

- Для редактирования атрибута необходимо нажать на знак ⋮ и выбрать пункт «**Редактировать**»:

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

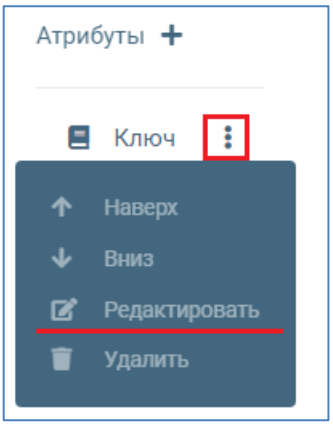


Рисунок 611 – Пример работы мастера (редактирование атрибутов)

После нажатия на добавление или редактирование атрибутов, откроется окно с настройками атрибута.

В данном примере открыто окно с добавлением атрибута, при редактировании атрибута часть полей будет заполнена значениями:

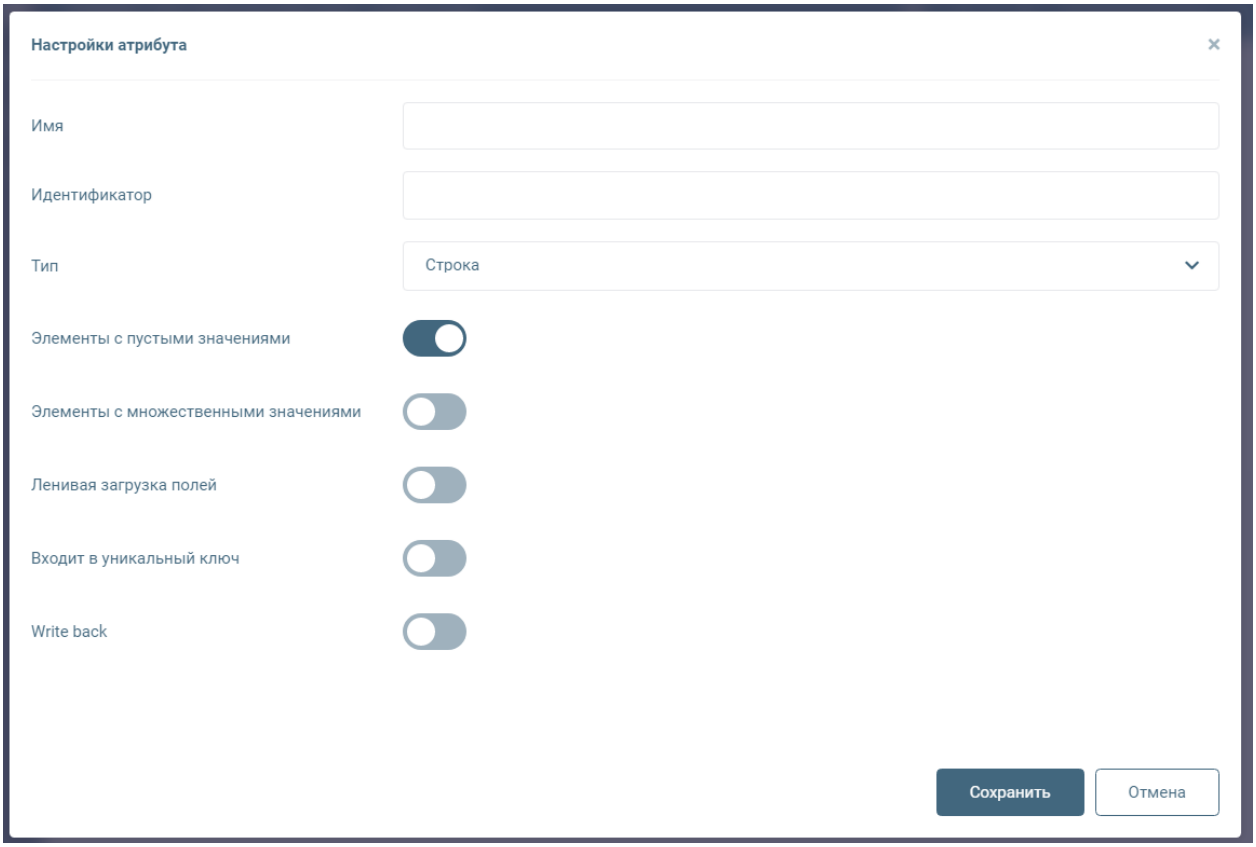


Рисунок 612 – Пример добавления атрибута

- **Имя** – наименование атрибута, которое будет использоваться для наименования столбца в мастере данных блока (см. п. 7.8.8).
- **Идентификатор** – ключ атрибута, по которому он будет доступен в таблице в базе данных.

Формат наименования может различаться, в зависимости от выбранной базы данных, но в общем случае рекомендуется использовать латинские буквы (a-Z), цифры (0-9) и знак подчеркивания (_).

- **Тип** – тип атрибута, возможные значения:

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

- **Строка** – строковый тип данных;
- **Дата** – значение даны. При выборе этого типа появляются дополнительные настройки:
  - **Значение по умолчанию** – значение даты по умолчанию;
  - **Маска** – маска даты для отображения в данных;
  - **Динамика** – динамика даты;
- **Целочисленный** – целое число;
- **Вещественный** – вещественное число;
- **Координаты** – строковый тип данных, заточенный для распознавания координат;
- **JSON** – строковый тип данных, в котором хранятся значения в формате JSON;
- **Справочник** – возможность связывания одного справочника с другими. В качестве ключа используется поле ID.

- **Элементы с пустыми значениями** – возможность сохранять атрибут без введённых значений;
- **Элементы с множественными значениями** – возможность записывать несколько значений в один атрибут.

Функционал не используется в текущей версии приложения:

- **Ленивая загрузка полей** – флаг, позволяющий при расчете блока произвести «ленивую загрузку» – сохранить в системную таблицу, а вместо значения прописать уникальный идентификатор поля;

Не все блоки могут преобразовать «ленивую загрузку» обратно в значение.

- **Входит в уникальный ключ** – флаг, отвечающий за вхождение поля в уникальный ключ таблицы (и в первичных ключ объектов);
- **Write-back** – возможность изменения значения поля при использовании механизма write-back (изменение и сохранение данных).

7.9 Отображение визуализатора в блоке

Визуализаторы – элементы, которые служат для графического представления данных.

Данный функционал позволяет отобразить и просматривать визуализаторы в рамках их же блоков на графе, без необходимости переходить на пресет.

Соответственно эта возможность функционирует только с блоками – визуализаторами.

Для включения отображения визуализатора в блоке на графе необходимо открыть меню блока и нажать по значку ⋯ у блока.

В появившемся меню выбрать «**Визуальные настройки»**.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

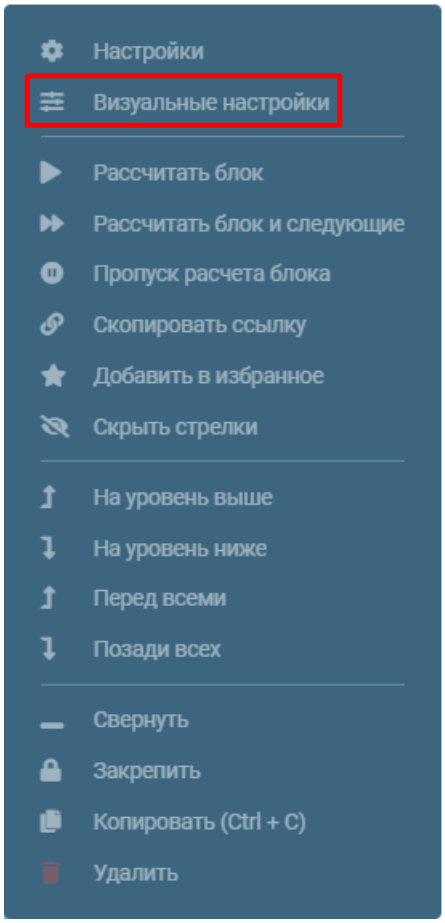


Рисунок 613 – Пример открытия мастера в настройках блока

В появившемся окне активировать флаг «**Отображать как визуализатор**».

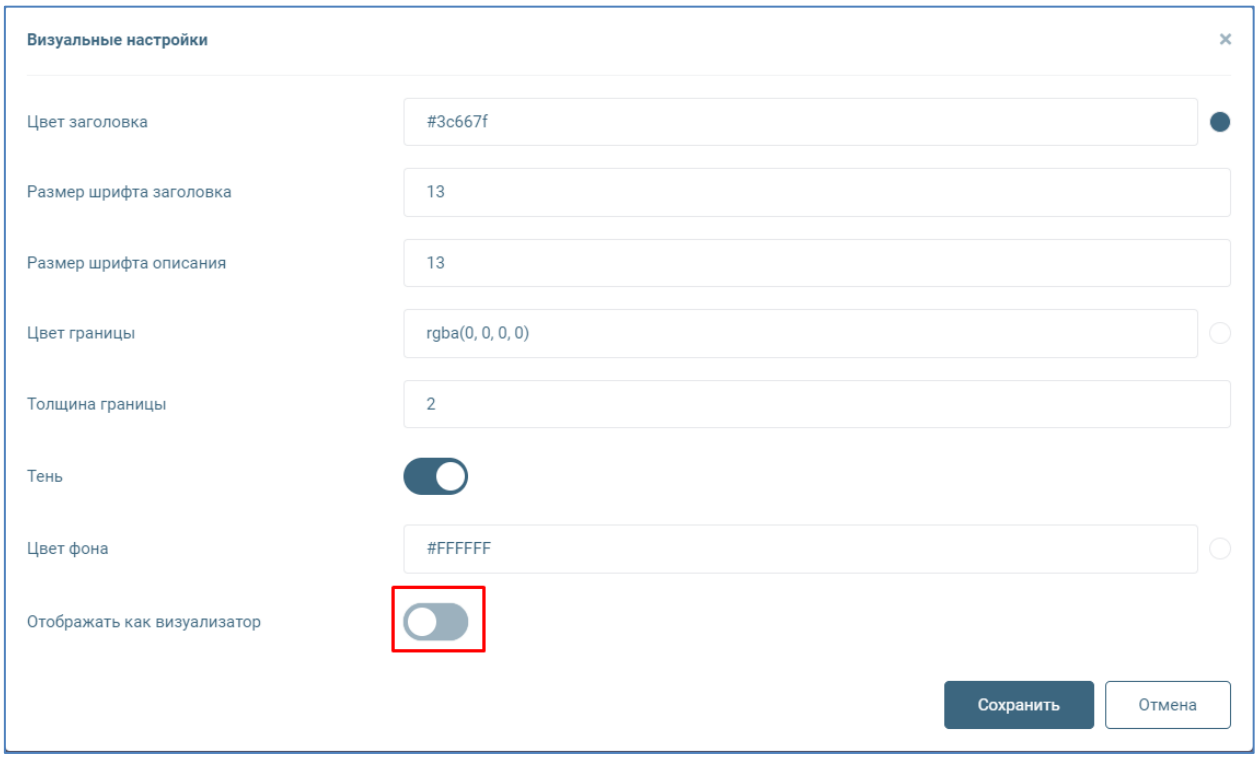
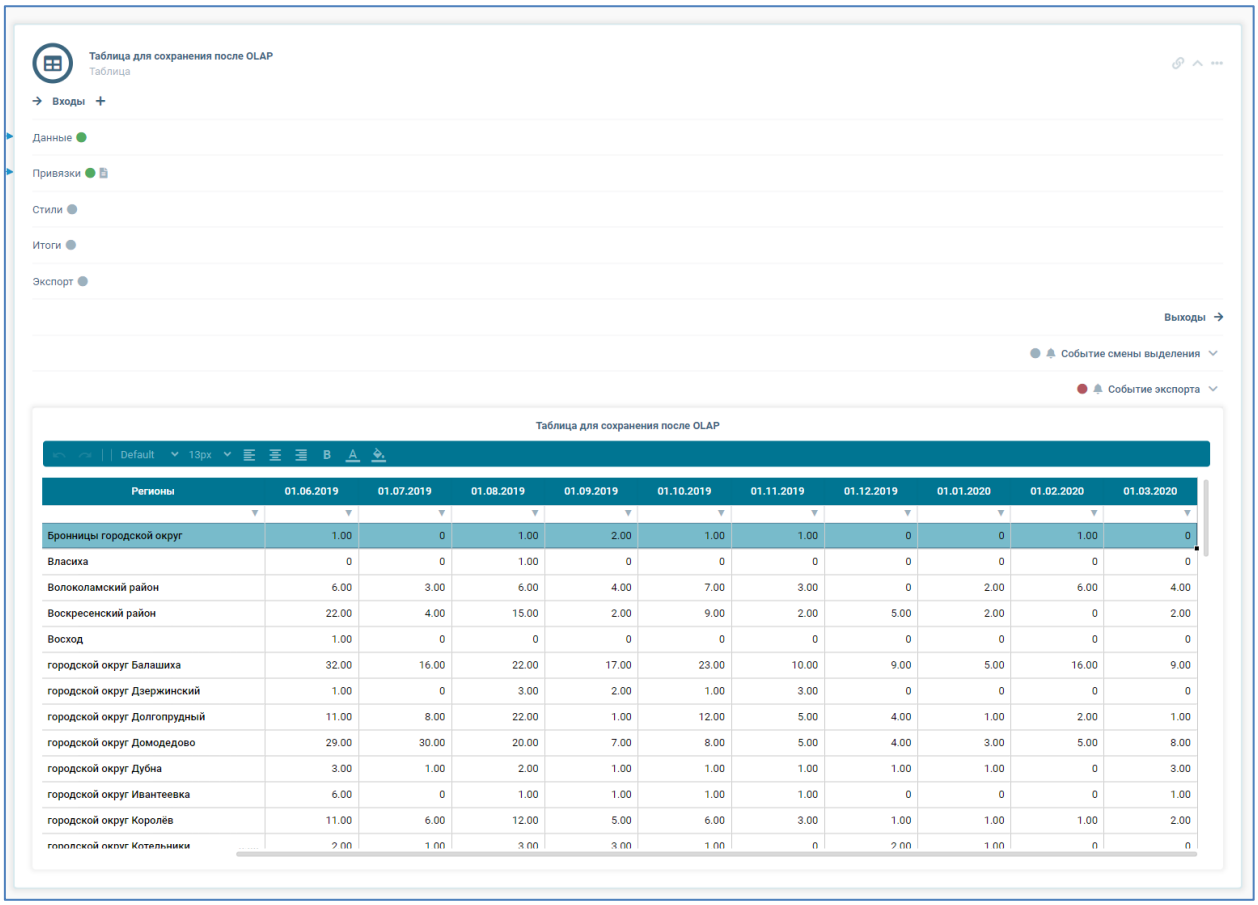


Рисунок 614 – Пример работы мастера (визуальные настройки)

Нажать «**Сохранить**».

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)



Регионы	01.06.2019	01.07.2019	01.08.2019	01.09.2019	01.10.2019	01.11.2019	01.12.2019	01.01.2020	01.02.2020	01.03.2020
Бронницы городской округ	1.00	0	1.00	2.00	1.00	1.00	0	0	1.00	0
Власиха	0	0	1.00	0	0	0	0	0	0	0
Волоколамский район	6.00	3.00	6.00	4.00	7.00	3.00	0	2.00	6.00	4.00
Воскресенский район	22.00	4.00	15.00	2.00	9.00	2.00	5.00	2.00	0	2.00
Восход	1.00	0	0	0	0	0	0	0	0	0
городской округ Балашиха	32.00	16.00	22.00	17.00	23.00	10.00	9.00	5.00	16.00	9.00
городской округ Дзержинский	1.00	0	3.00	2.00	1.00	3.00	0	0	0	0
городской округ Долгопрудный	11.00	8.00	22.00	1.00	12.00	5.00	4.00	1.00	2.00	1.00
городской округ Домодедово	29.00	30.00	20.00	7.00	8.00	5.00	4.00	3.00	5.00	8.00
городской округ Дубна	3.00	1.00	2.00	1.00	1.00	1.00	1.00	1.00	0	3.00
городской округ Ивантеевка	6.00	0	1.00	1.00	1.00	1.00	0	0	0	1.00
городской округ Королёв	11.00	6.00	12.00	5.00	6.00	3.00	1.00	1.00	1.00	2.00
городской округ Котельники	2.00	1.00	3.00	3.00	1.00	0	2.00	1.00	0	0

Рисунок 615 – Пример настроенного блока

7.10 Оформление графа (блоки, стрелки и т.д.)

С целью визуального разграничения объектов на графе используются настройки внешнего вида объектов и фигуры.

7.10.1 Визуальные настройки блоков на графе

Для визуальной настройки блока необходимо нажать по значку ••• соответствующего блока.

В появившемся меню выбрать «**Визуальные настройки**».

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

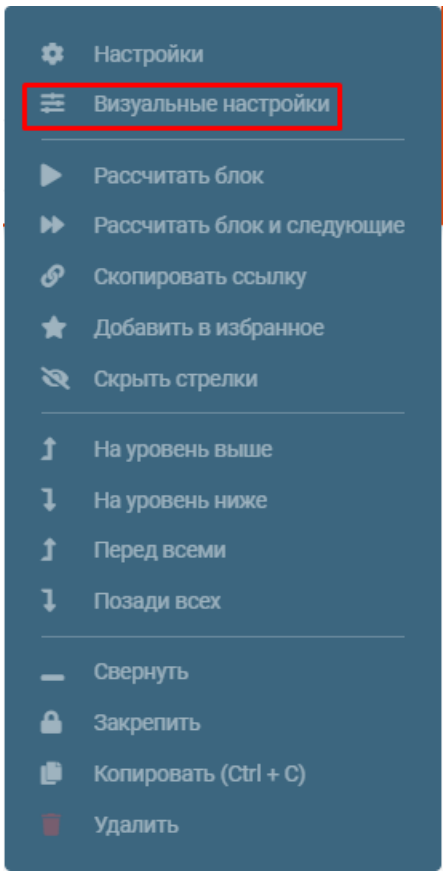


Рисунок 616 – Пример меню блока

Окно визуальных настроек блока:

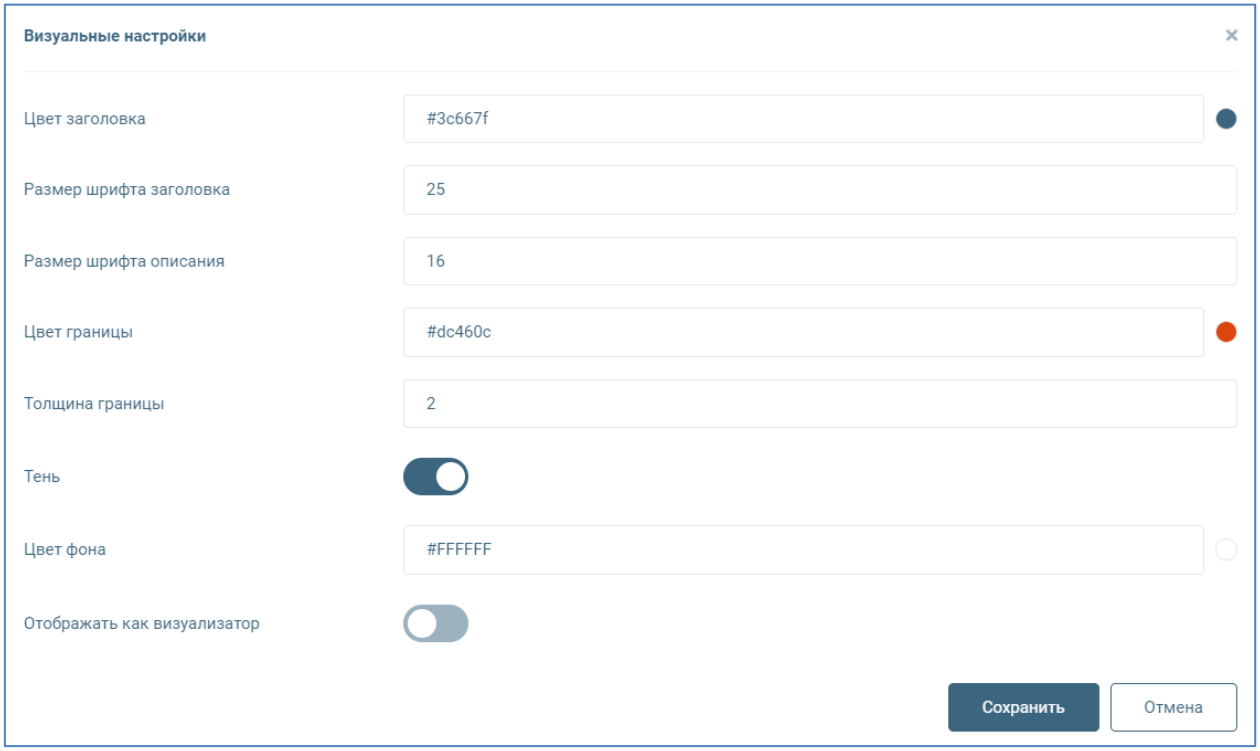


Рисунок 617 – Окно визуальных настроек блока

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Описание визуальных настроек блока

Название параметра	Тип параметра	Описание	Пример
Группа «Визуальные настройки»			
Цвет заголовка	Цвет	Задает цвет заголовка блока	
Размер шрифта заголовка	Целое число	Задает размер шрифта заголовка блока	
Размер шрифта описания	Целое число	Задает размер шрифта описания блока	
Цвет границы	Цвет	Задает цвет границы блока	
Толщина границы	Целое число	Задает толщину границы блока	
Тень	Флаг	Включает отображение тени блока на графе	
Цвет фона	Цвет	Задает цвет фона заголовка блока	
Отображать как визуализатор	Флаг	Включает отображение визуализатора на блоке	

Пример визуальных настроек блока:

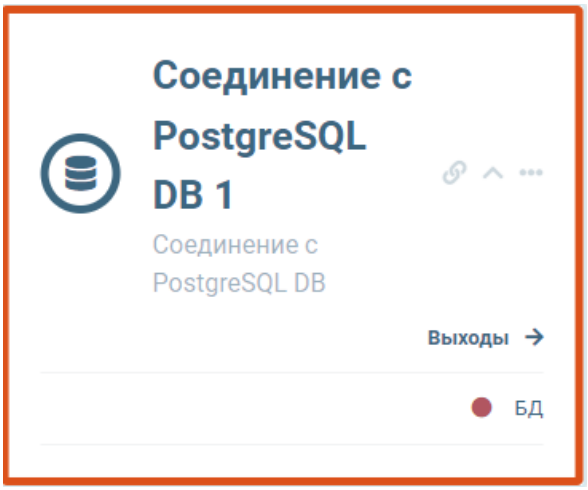


Рисунок 618 – Пример визуальных настроек блока

7.11 Фигуры

На графе существует возможность создавать специальный вид блоков, геометрические фигуры и задавать им визуальные настройки.

Эти объекты служат для визуального разграничения групп блоков по какому – либо признаку.

Создаются эти блоки из библиотеки, раздел «**Оформление графа**».

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

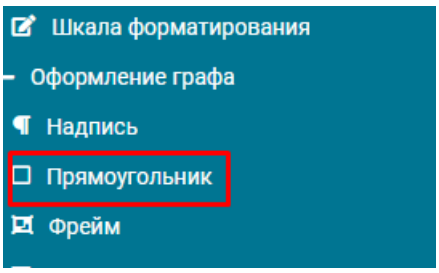


Рисунок 619 – Пример блока фигур

Блоки фигур располагаются на отдельном слое графа, где с ними можно взаимодействовать и настраивать.

Для перехода на этот слой необходимо в меню графа нажать по «**Блоки**».

Рисунок 620 – Пример меню графа

В появившемся окне выбрать «**Фигуры**».

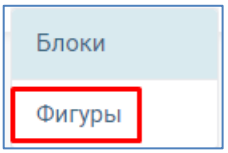


Рисунок 621 – Пример выбора «**Фигуры**»

Для перехода к настройкам блока «**Прямоугольник**» необходимо кликнуть по ••• .

В появившемся меню выбрать «**Настройки**».

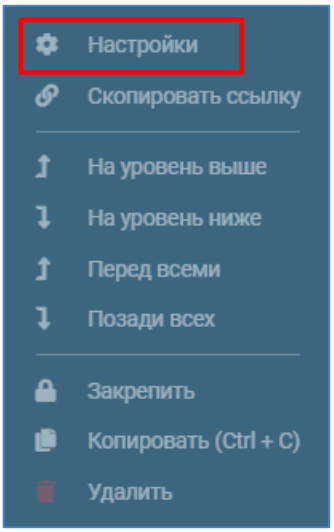


Рисунок 622 – Пример меню настроек

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

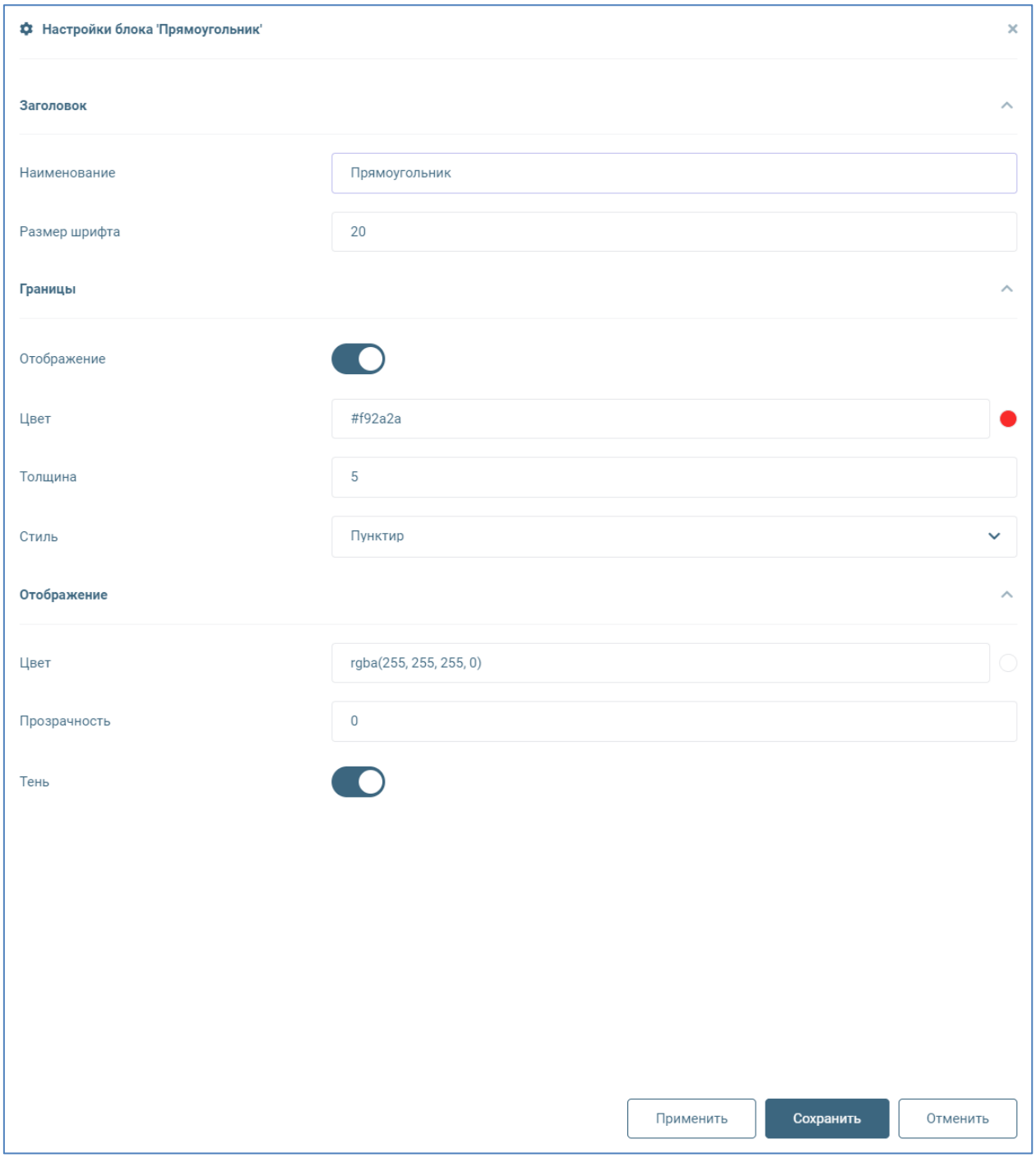


Рисунок 623 – Пример настройки блока «Прямоугольник»

Описание настроек

Название параметра	Тип параметра	Описание	Пример
Группа «Заголовок»			
Наименование	Строка	Задает наименование блока	
Размер шрифт	Целое число	Задает размер шрифта заголовка блока	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Тип параметра	Описание	Пример
Группа «Границы»			
Отображение	Флаг	Включает/отключает отображение границ	
Цвет	Цвет	Задает цвет границы	
Толщина	Целое число	Задает толщину границы	
Стиль	Выпадающий список	Задает стиль отображения границы	
Группа «Отображение»			
Цвет	Цвет	Задает цвет заливки	
Цвет фона	Целое число	Задает opacity для заливки	
Тень	Флаг	Включает отображение тени на графе	

Пример блока «Прямоугольник».

Рисунок 624 – Пример блока «Прямоугольник»

7.11.1Миникарта (миниграф)

Внешний вид приведен на рисунке.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

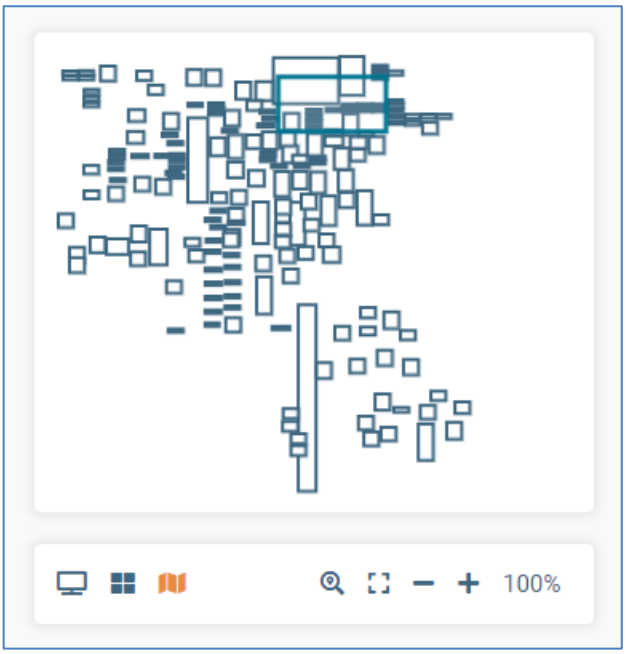


Рисунок 625 – Пример внешнего вида миниграфа

Это объект, расположенный в правом нижнем углу окна графа, схематично отображает количество блоков, а также позволяет переместиться по клику в любую область графа.

Непосредственно в окне миниграфа отображаются все блоки графа.

Рисунок 626 – Пример внешнего вида миниграфа

Там же отображается видимая пользователем на текущий момент часть графа (вьюпорт).

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

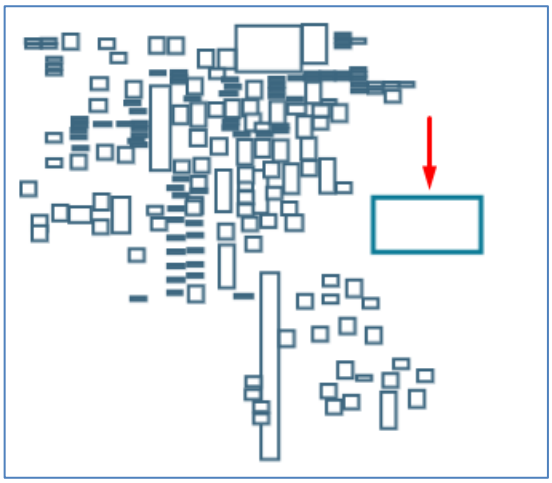


Рисунок 627 – Пример части графа (вьюпорт)

При включенном «Режиме отладки» цветом на миникарте отображается статус блоков.

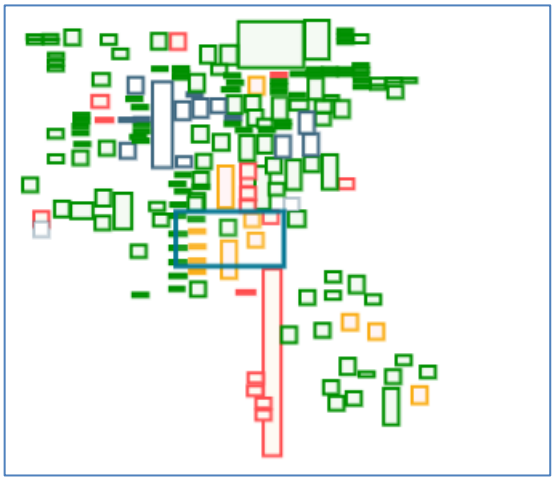


Рисунок 628 – Пример миниграфа (режим отладки)

Меню миниграфа.

Рисунок 629 – Пример меню миниграфа

Описание меню миниграфа:

Название параметра	Описание	Пример
Полноэкранный режим	Включает/отключает полноэкранный режим отображения графа	
Сетка	Включает/отключает отображение сетки на графе	
Карта графа	Включает/отключает миникарты (миниграфа). По умолчанию – включено	
Сохранить состояние	Задает место зум и координаты открытия графа по умолчанию	

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Название параметра	Описание	Пример
Показать граф полностью	Включает/отключает полноэкранный режим отображения графа	
Отдалить	Уменьшает масштаб графа	
Приблизить	Увеличивает масштаб графа	
Сбросить масштаб	Сбрасывает масштаб к дефолтному. По умолчанию – 100%	100%

Размер вьюпорта на миниграфе изменяется в зависимости от зума на графе.

Для перехода к нужной части графа можно нажать по ней в требуемом месте.

Активные кнопки меню миникарты изменяют цвет на оранжевый.

7.11.2 Масштабирование графа

Это процесс увеличения или уменьшения вьюпорта пользователя.

Масштабирование графа происходит при помощи:

- Клавиш в меню миникарты.

Рисунок 630 – Пример кнопок для масштабирования графа

- Зажав клавишу «**Ctrl**» + вращение колесо мыши.

При уменьшении масштаба перестает отображаться сетка, а блоки на графе отображаются в свёрнутом виде.

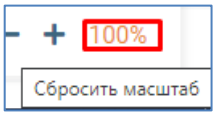


Рисунок 631 – Пример кнопок для масштабирования графа

Клик по кнопке «**Сбросить масштаб**» возвращает масштаб графа по умолчанию.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Рисунок 632 – Пример масштабирования графа

7.11.3 Избранные блоки

Функциональность позволяет создать пользовательскую библиотеку блоков, использованных ранее.

При создании блока из данного списка, будет создана копия блока со всеми настройками, на момент добавления в избранное.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

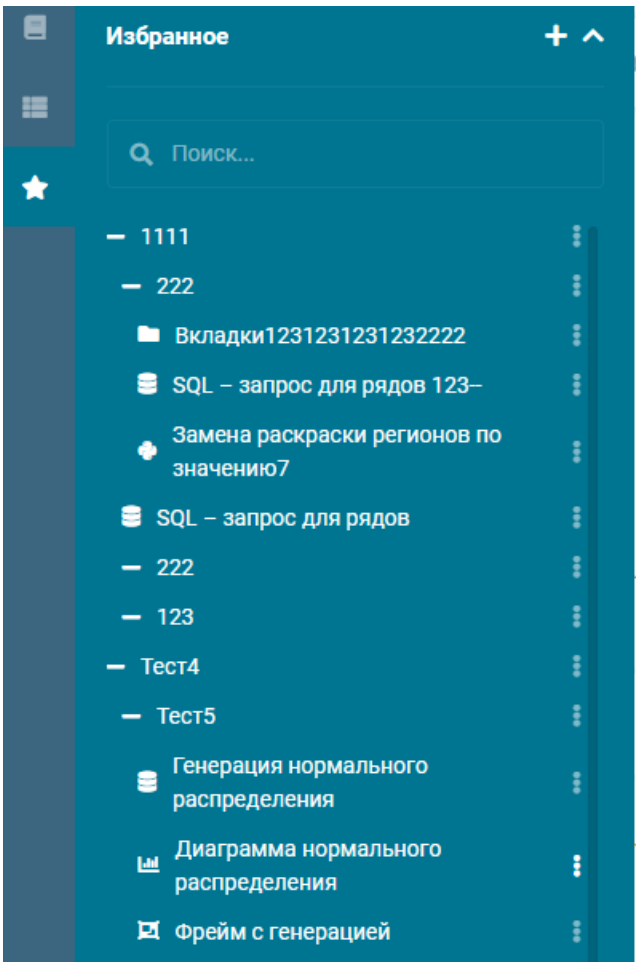


Рисунок 633 – Пример избранных блоков

Для открытия списка «Избранное» нужно нажать по соответствующей кнопке на странице графа.

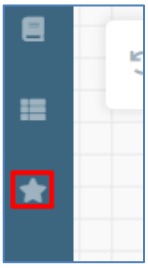


Рисунок 634 – Пример кнопки «Избранное»

Из появившегося списка создается блок путём перетаскивания из списка на граф.

В списке избранных блоков есть возможность создать папки.

Для этого необходимо нажать по значку создать папку.

В появившемся окне необходимо заполнить поля и нажать «**Сохранить».**

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

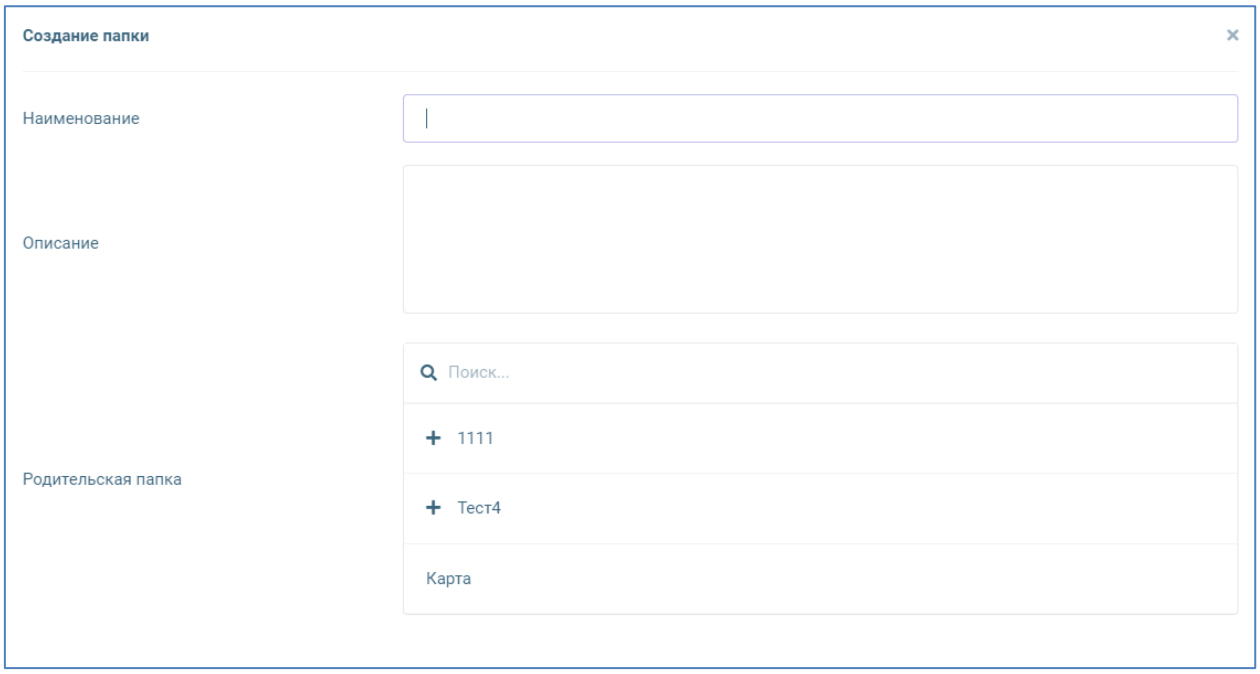


Рисунок 635 – Пример создания папки

Описание полей:

Название параметра	Описание	Пример
Наименование	Наименование папки отображаемое в списке «Избранное»	
Описание	Пользовательское описание создаваемой папки	
Родительская папка	Позволяет выбрать родительскую папку, при создании вложенных папок	

Для добавление блока в список «**Избранное»** необходимо нажать по значку ••• на добавляемом блоке.

В появившемся меню выбрать «**Добавить в избранное».**

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

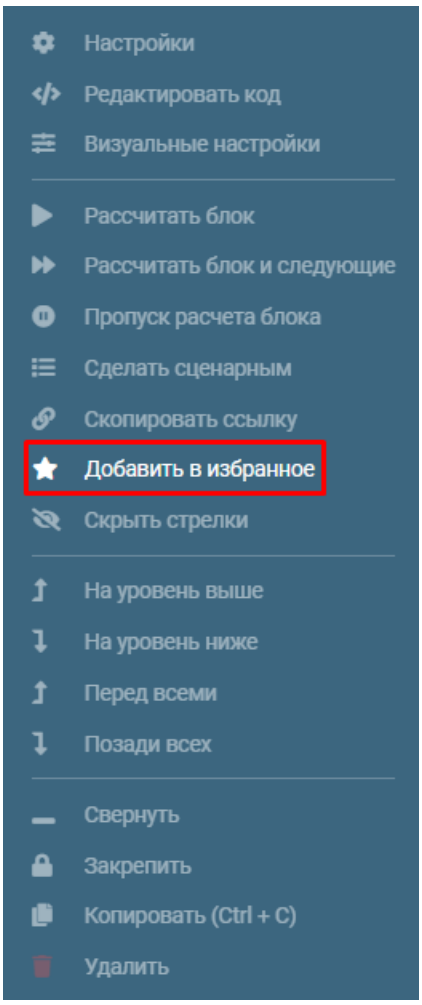


Рисунок 636 – Пример кнопки «**Добавить в избранное**»

В появившемся окне необходимо заполнить поля и нажать «**Сохранить».**

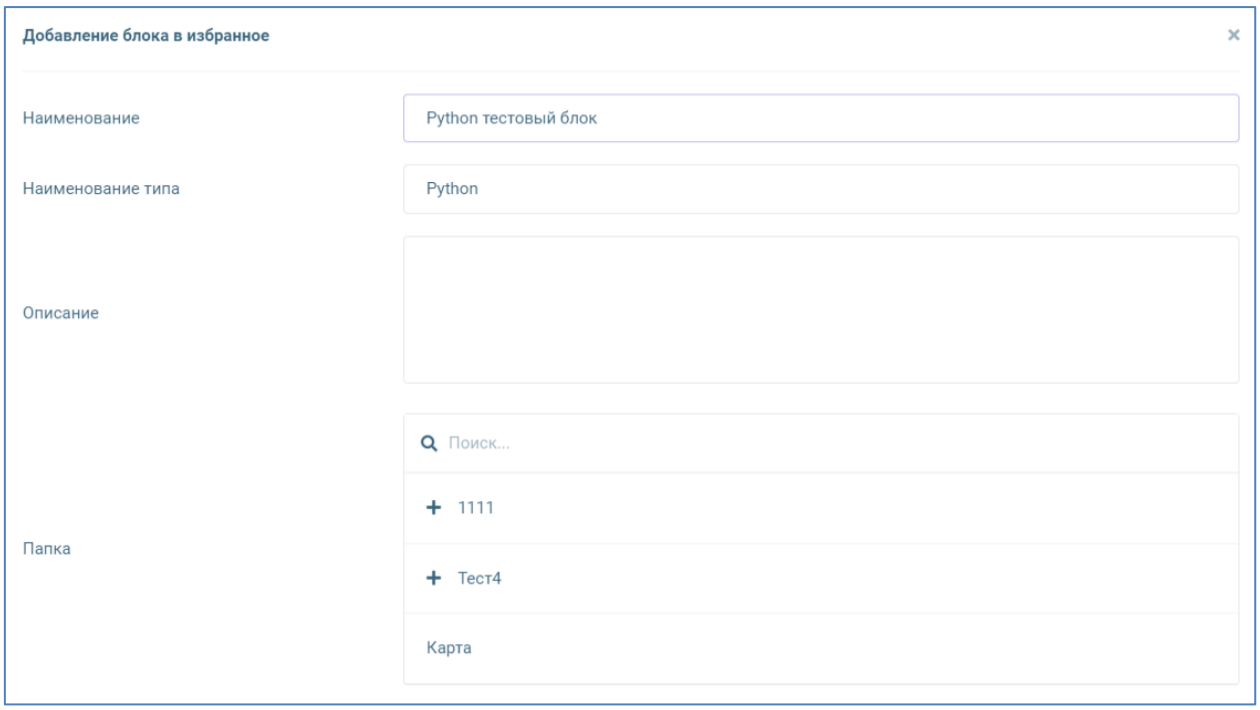


Рисунок 637 – Пример окна «**Добавление в избранное**»

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

Описание полей:

Название параметра	Описание	Пример
Наименование	Наименование блока отображаемое в списке Избранное	
Наименование типа	Тип сохраняемого блока	
Описание	Пользовательское описание добавляемого блока	
Папка	Позволяет выбрать родительскую папку, при добавлении блока в Избранное	

Важно! Список «Избранное» является общим для всех пользователей.

7.12 Настройки задачи

На графе можно настраивать параметры задачи.

При нажатии на значок «Шестеренка» откроется окно настроек.

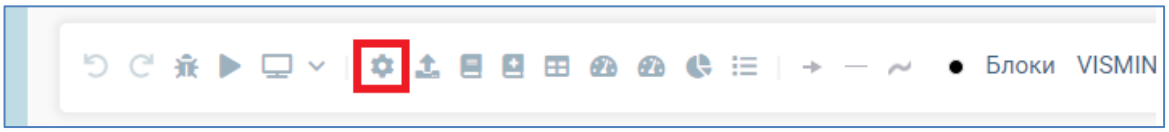


Рисунок 638 – Пример кнопки «Шестеренка»

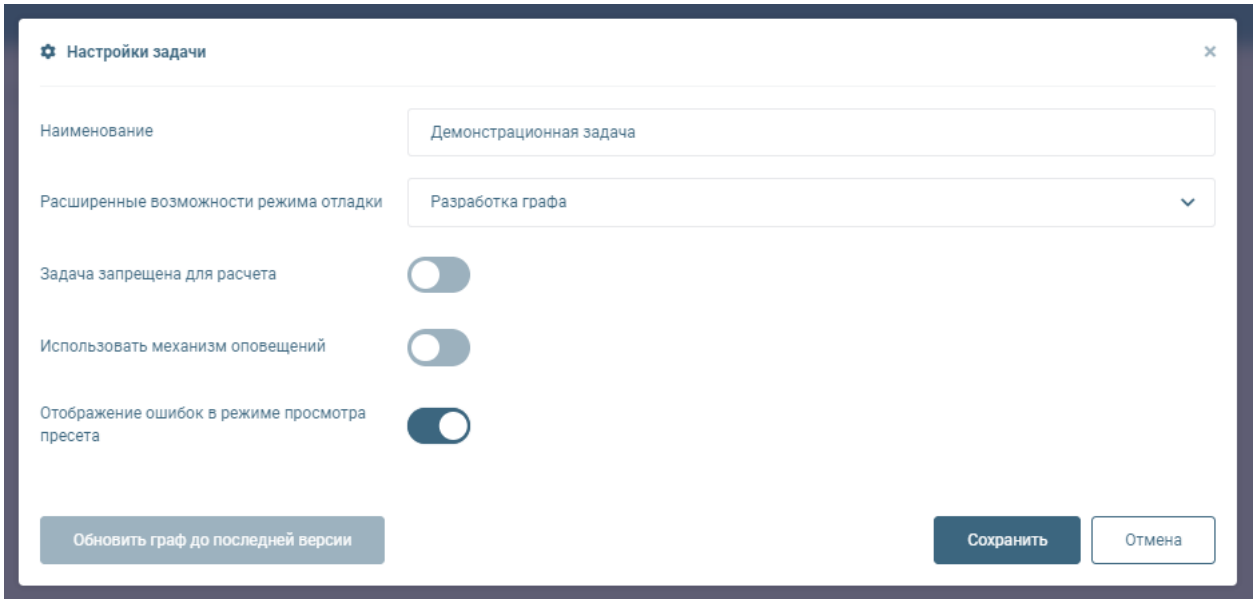


Рисунок 639 – Пример окна «Настройки задачи»

7.12.1 Наименование

Данный пункт отображает название задачи.

Здесь можно переименовать задачу из графа.

7.12.2 Расширенные возможности режима отладки

Данный пункт отвечает за то, как будет рассчитываться граф.

1) Разработка графа – в данном режиме можно просматривать в режиме отладки результаты каждого блока.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

2) Эксплуатация графа – в данном режиме отключается отображение в режиме отладки результатов блока. Это делается для ускорения расчета графа.

По умолчанию: Разработка графа

7.12.3 Задача запрещена для расчета

Флаг, отвечающий можно ли рассчитывать задачу или нет.

Данный флаг включается автоматически, если задачу не удается рассчитать.

По умолчанию: флаг выключен

7.12.4 Использовать механизм оповещений

Флаг, отвечающий надо ли уведомлять на пресетах об изменении данных в визуализаторах (например, после расчета графа каким-либо пользователем).

По умолчанию: флаг включен

7.12.5 Отображение ошибок в режиме просмотра пресета

Флаг, контролирующий вывод ошибок при взаимодействии с пресетами графа.

По умолчанию: флаг включен

7.12.6 Обновить граф до последней версии

Кнопка, при нажатии на которую можно обновить блоки на графе до последней версии.

Блок считается устаревшим в том случае, если при обновлении платформы до новой версии блок так же был обновлен или изменен (в библиотеке). Процесс обновления попытается сопоставить старые поля блока и новые.

Важно

Если изменения в блоке были критическими и большая его структура была сильно изменена – блок может перестать работать, так как часть его настроек не перенесется в новую версию.

Во время обновления происходит перебор всех блоков на графе, и платформа автоматически обновляет блоки до актуального состояния. В логе выполнения обновления будет содержаться информация о том, какие блоки были обновлены и до каких версий.

Кнопка «Обновить граф до последней версии» доступна только если на графе есть устаревшие блоки.

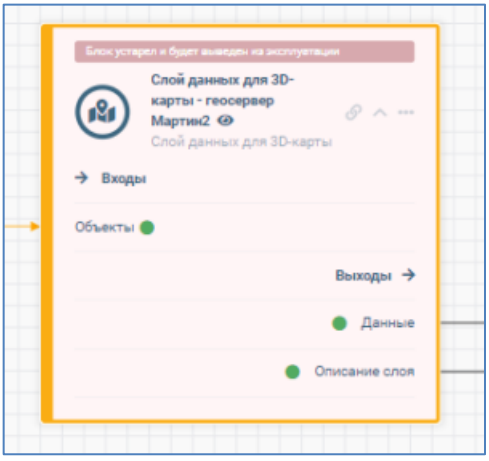


Рисунок 640 – Пример устаревшего блока

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

7.13Лог расчета

Во время расчета графа отображается лог его расчета.

В лог выводится информация:

- значок статус сообщения (информация, предупреждение, ошибка);
- время сообщения;
- действие (расчет блока);
- название блока;
- тип блока.

На сообщения со статусом «Предупреждение» или «Ошибка» можно нажать мышкой и перейти к блоку.

По окончанию расчета, в конце лога, может быть сообщение о количестве блоков с устаревшей версией (блоки, версия которых на графе отличается от версии в библиотеке).

7.14Undo/Redo

Во время работы с графом, запоминаются действия пользователя (до 20 действий) для того чтобы пользователь мог отменить или восстановить свои действия.

Запоминаемые действия:

- создание/удаление блока;
- создание/удаление входов/выходов блока;
- создание/удаление связи между блоками;
- вставка блока;
- трансформация блока;
- изменение стиля блока;
- сохранение настроек блока;
- изменение позиции блока;
- пропуск расчета блоков;
- скрытие/отображение связи между блоками;
- изменение позиции группы блоков;
- вставки/удаления группы блоков;
- пропуск расчета группы блоков;
- скрытие/отображение связи между группами блоков.

Чтобы отменить действие пользователя, надо нажать на стрелку, показывающею против часовой стрелки:

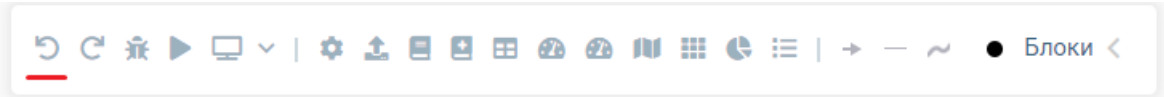


Рисунок 641 – Пример кнопки для отмены действия

Чтобы вернуть действие пользователя, надо нажать на стрелку, показывающею по часовой стрелке:

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

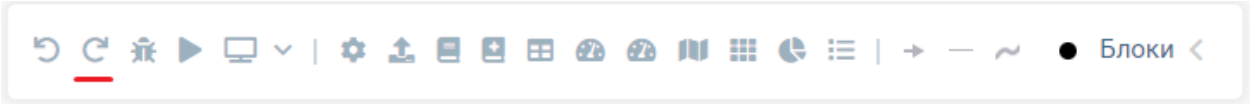


Рисунок 642 – Пример кнопки для повторения действия

7.14.1 Особенность работы

Если пользователь А производит действия с блоком, с которым работал пользователь Б, то пользователь Б не сможет отменить действия по этому блоку и последующие действия, которые были после этого блока.

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

8 Мобильное приложение

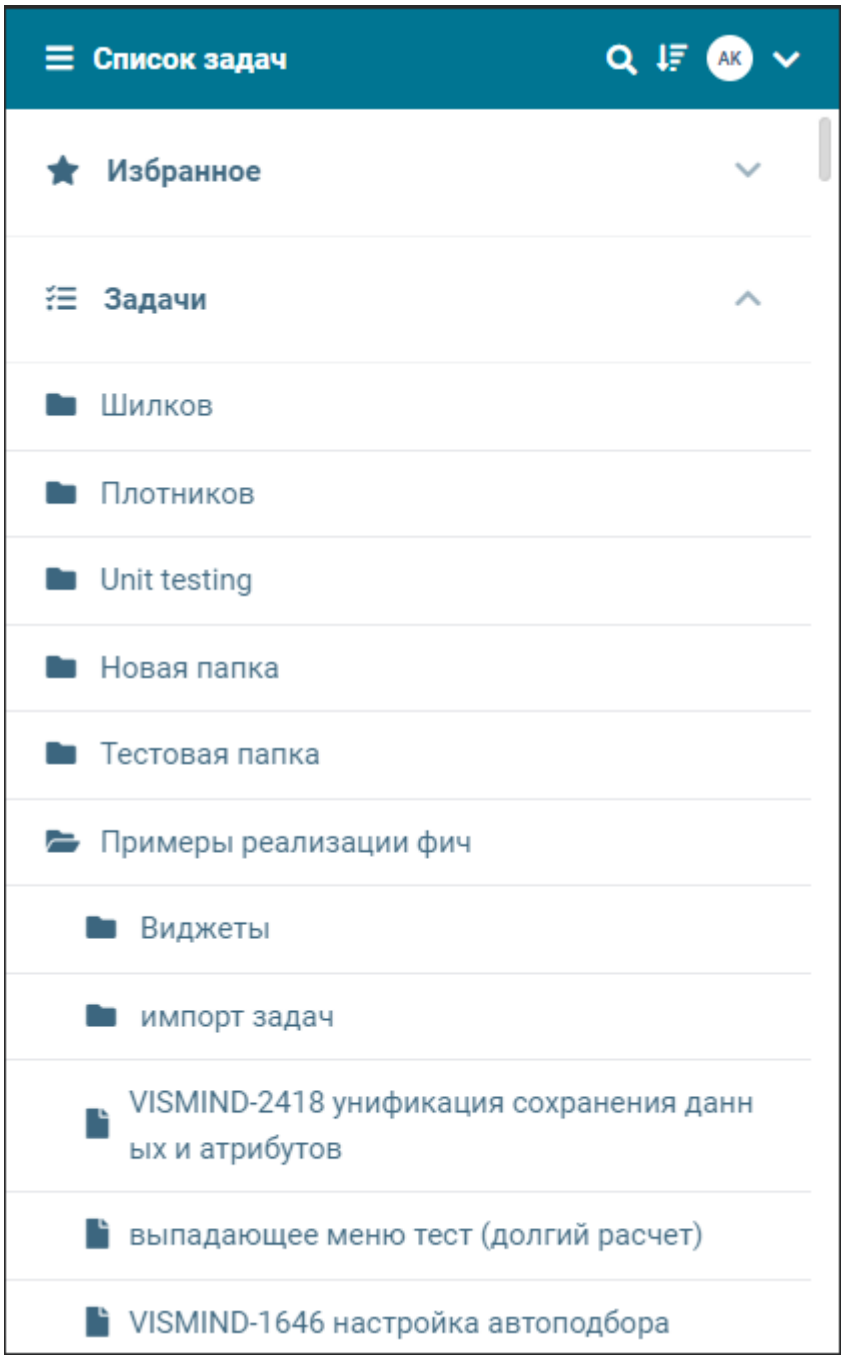


Рисунок 643 – Пример окна мобильного приложения

8.1 Описание

Мобильное приложение ЦП УвП — это отдельная мобильная версия приложения ЦП УвП. Открывается в браузере по ссылке. Не требует установки.

8.2 Функционал

В мобильной версии доступны:

- аутентификация/выход
- просмотр списка задач
- поиск и сортировка задач

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

- открытие и просмотр пресетов

8.3 Особенности

- в пресетах не отображаются визуализаторы-контейнеры
- в пресетах визуализаторы расположены на всю ширину экрана, друг под другом. Для их просмотра осуществляется прокрутка экрана вниз/вверх

8.4 Ограничения

Функционал, использующий библиотеку WebGL (в частности блок "Слой данных для 3D карты" типа "Слой корреспонденций", блок "Визуализатор JavaScript" при использовании 3D), не поддерживается на большинстве мобильных устройств

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)

История изменений

№ п/п	Версия	Дата	Внесенные изменения
1.	1.0	31.01.2025	Руководство пользователя ЦП УВП (версия платформы 0.43)

© ООО «Бюро Информационных Технологий» (ООО «БИТ»)