

УТВЕРЖДЕН
RU.1107746574308.069-01 34 01-ЛУ

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ
«Платформа разработки программного обеспечения БЗ»

Руководство оператора

RU.1107746574308.069-01 34 01

Листов 105

2023

АННОТАЦИЯ

Настоящий документ является руководством оператора (далее – Руководство) программного обеспечения «Платформа разработки программного обеспечения БЗ» RU.1107746574308.069-01 (далее – Платформа).

Руководство содержит общие сведения о Платформе, ее характеристиках, а также порядке выполнения различных операций при эксплуатации Платформы.

Руководство разработано с учетом требований ГОСТ 19.505-79 «Единая система программной документации. Руководство оператора. Требования к содержанию и оформлению».

СОДЕРЖАНИЕ

1. Общие сведения	5
1.1. Наименование	5
1.2. Назначение	5
1.3. Функции Платформы	5
2. Описание характеристик	7
2.1. Общее программное обеспечение, необходимое для работы Платформы.....	7
2.2. Состав Платформы	7
2.3. Технические средства, необходимые для работы Платформы.....	9
2.4. Уровень квалификации пользователя	10
3. Подготовка к работе	11
3.1. Общие сведения.....	11
3.2. Работа по администрированию Платформы.....	11
3.3. Порядок проверки работоспособности Платформы	11
4. Работа с Платформой.....	12
4.1. Общие сведения.....	12
4.2. Авторизация и ролевая модель пользователей.....	12
4.3. Работа с простыми формами	13
4.3.1. Настройка отображения полей в форме	14
4.3.2. Создание и первичная настройка типа представления	25
4.3.3. Создание и настройка формы списка	28
4.3.4. Создание и настройка детальной страницы	32
4.4. Использование формул в конструкторе	34
4.4.1. Общее описание и контексты данных	34
4.4.2. Доступные методы в формулах	38
4.4.3. Доступные методы в объекте FormсыMoment	41
4.4.4. Проверка работоспособности формул.....	48
4.5. Работа с бизнес-процессами со статусами.....	49
4.5.1. Создание бизнес-процесса со статусами	50
4.5.2. Настройка прав доступа	55
4.6. Работа с печатными формами	57
4.6.1. Создание печатной формы.....	57
4.6.2. Создание шаблонов печатных форм.....	61
4.6.3. Настройка подписания печатных форм.....	62
4.7. Работа с аналитическими панелями (дашбордами)	63
4.7.1. Создание аналитической панели (дашборда)	64
4.8. Работа с витриной данных.....	66
4.8.1. Создание витрины данных.....	66
4.9. Работа с аналитическими функциями	71
4.9.1. Добавление агрегации на форму элемента/списка	71
4.9.2. Добавление состояния на форму элемента/списка.....	73
4.9.3. Добавление состояния на карте.....	75

4.10. Настройка операций импорта/экспорта в Excel	75
4.11. Работа с публичными сайтами	80
4.11.1. Создание публичного сайта	81
4.12. Модуль «Онбординг»	84
4.12.1. Создание урока (подсказки) на HTML-странице	86
5. Решение проблем	93
5.1. Техническая поддержка.....	93
5.2. Резервное копирование и восстановление Платформы.....	93
5.2.1. Плановое резервное копирование Платформы.....	93
5.2.2. Внеплановое резервное копирование Платформы	94
5.2.3. Восстановление информации из резервных копий	94
5.3. Аварийные ситуации.....	94
Приложение А. Арифметические функции	96
Приложение Б. Статистические функции	98
Приложение В. Тригонометрические функции	99
Перечень сокращений.....	101
Термины и определения	102

1. ОБЩИЕ СВЕДЕНИЯ

1.1. Наименование

Полное наименование программы: Платформа разработки программного обеспечения БЗ RU.1107746574308.069-01.

Сокращенное наименование программы: Платформа БЗ. В рамках настоящего документа употребляется также термин «Платформа».

Обозначение программы: RU.1107746574308.069-01.

Платформа разработки программного обеспечения БЗ RU.1107746574308.069-01 – это российское программное обеспечение, организация-разработчик: Общество с ограниченной ответственностью «Большая Тройка» (ООО «Большая Тройка»).

Сайт организации-разработчика: <https://big3.ru>.

Организация-правообладатель: Общество с ограниченной ответственностью «Большая Тройка» (ООО «Большая Тройка»).

1.2. Назначение

Платформа представляет собой облачную платформу для создания и развития интернет-проектов, которая позволяет конструировать информационные системы и их мобильные версии.

1.3. Функции Платформы

Платформа обеспечивает выполнение следующих основных функций:

– создание страниц (форм) следующих типов:

- форма списка,
- форма элемента,
- форма экспорта,
- форма для операции,
- аналитическая панель,
- лендинг;

– настройка бизнес-процессов;

– выполнение операций с печатными формами документов:

- создание шаблонов в формате .DOCX,
- формирование и скачивание в формате .DOCX и/или .PDF,

- подписание электронно-цифровой подписью;
- визуализация и анализ информации о бизнес-процессах и их эффективности посредством аналитической панели;
- создание обучающих курсов для пользователей информационных систем;
- возможность интеграции Платформы с информационными системами, находящимися внутри (или вне) ИТ-инфраструктуры организации-пользователя, включая:
 - Open Source Routing Machine (OSRM – сервис маршрутизации),
 - Open Street Map (OSM – картографический сервис);
- обеспечение коммуникации пользователей:
 - уведомление о проводимых мероприятиях;
 - массовые рассылки;
- настройка и конфигурирование Платформы.

2. ОПИСАНИЕ ХАРАКТЕРИСТИК

2.1. Общее программное обеспечение, необходимое для работы Платформы

Программное обеспечение (ПО) Платформы включает в себя:

- специальное программное обеспечение (СПО): веб-приложение;
- общее программное обеспечение (ОПО), предназначенное для обеспечения работы СПО Платформы, в состав которого входят:
 - библиотеки стандартных методов разработки приложений: библиотеки стандартных методов разработки веб-приложения, библиотека Leaflet для отображения карт при взаимодействии с Open Street Map,
 - типовое ПО: Open Street Map, GPS\ГЛОНАСС,
 - операционная система: Microsoft Windows, Debian GNU/Linux,
 - система управления базами данных: PostgreSQL, ClickHouse, Redis,
 - сервис брокера сообщений: RabbitMQ,
 - сервис фоновых процедур: Celery,
 - сервис регламентных заданий: Cron,
 - сервис WebSocket,
 - сервис полнотекстового поиска: Manticore Search,
 - связующее ПО: Nginx;
 - промежуточное системное ПО: HAProxy, PgBouncer;
- ПО, установленное на рабочих станциях: Microsoft Edge, Google Chrome, антивирус.

2.2. Состав Платформы

Платформа реализована в виде веб-приложения. По функциональному признаку функции Платформы объединены в модули:

- модуль авторизации и ролевой модели пользователей – реализует процессы регистрации пользователей, а также настройки прав доступа их в Платформе;
- модуль конструктора страниц интерфейса – реализует функции создания страниц (форм);
- модуль настройки бизнес-процессов – реализует функции настройки бизнес-процессов двух типов: первый – это бизнес-процесс, основанный на смене состояний объектов, второй – это система настраиваемых шаблонов событий и реакция на эти события;
- модуль конструктора печатных форм – реализует функции выполнения операций с печатными формами документов;

- модуль генерации аналитических панелей – реализует функции визуализации и анализа информации о бизнес-процессах и их эффективности;
- модуль онбординга – реализует функцию создания обучающихся курсов для конечных пользователей информационных систем.

Платформа состоит из следующих компонентов:

- серверная часть («Backend»);
- клиентская часть («Frontend»).

Серверная часть функционирует под управлением ОС Debian GNU/Linux.

В качестве систем управления баз данных (далее – СУБД) применяются свободная объектно-реляционная СУБД PostgreSQL, колоночная аналитическая СУБД ClickHouse, резидентная СУБД Redis.

Также применяются обязательные к установке следующие сервисы:

- сервис брокера сообщений: RabbitMQ;
- сервис фоновых процедур: Celery;
- сервис регламентных заданий: Cron;
- сервис WebSocket;
- сервис полнотекстового поиска: Manticore Search.

Серверная часть реализует выполнение следующих основных функций:

- взаимодействие с периферийными устройствами (сбор данных, передача данных и управляющих команд);
- ведение внутренней базы данных;
- управление настройками Платформы;
- журналирование работы Платформы;
- предоставление программного интерфейса взаимодействия (API) для клиентской части.

Клиентская часть функционирует в рамках веб-браузера на ПЭВМ пользователя и реализует функции взаимодействия с пользователем с использованием графического пользовательского интерфейса.

Пользовательский интерфейс Платформы реализован по технологии «тонкий клиент» и обеспечивает возможность работы пользователя с Платформой с использованием распространенных веб-браузеров актуальных версий.

Клиентская часть реализует выполнение функций следующих модулей:

- модуль авторизации и ролевой модели пользователей;

- модуль конструктора страниц интерфейса;
- модуль настройки бизнес-процессов;
- модуль конструктора печатных форм;
- модуль генерации аналитических панелей;
- модуль онбординга.

2.3. Технические средства, необходимые для работы Платформы

Серверная часть Платформы реализована с использованием технологии виртуализации. Система виртуализации развернута на стороне организации-разработчика.

На базе виртуальных машин (ВМ) развернуты база данных и серверная часть веб-сервисов Платформы.

Необходимые характеристики к вычислительным мощностям виртуальных машин представлены в таблице 1.

Таблица 1 – Необходимые характеристики в ВМ

Наименование параметра, единицы измерения	Значение	
	ВМ Приложений	ВМ БД Master
Частота процессора, ГГц	не ниже 2,1	не ниже 2,1
Количество ядер (процессоров)	16	16
Количество оперативной памяти, ГБ	64	64
Объем диска, ГБ	60	400
Канал связи, Гбит/сек: для локальной сети для сети Интернет	не ниже 10 не ниже 1	не ниже 10 -

Необходимые характеристики дискового пространства ВМ представлены в таблице 2.

Таблица 2 – Необходимые характеристики дискового пространства ВМ

Дисковое пространство в разрезах:	Единицы измерения	Значение	
		ВМ Приложений	ВМ БД Master
ОС и системное ПО	ГБ	30	30
Дистрибутивы и исходные коды	ГБ	10	20
Логи	ГБ	20	300
Итого:	ГБ	60	350

Клиентская часть Платформы выполняется в рамках веб-браузера на ПЭВМ. ПЭВМ должна иметь характеристики не хуже:

- центральный процессор (CPU): Intel Core i3, 2 ГГц (или эквивалент);
- объем оперативной памяти (RAM): 4 ГБ;
- жесткий диск (HDD): 1x HDD – объем свободного пространства 10 ГБ;
- видеоадаптер: встроен в системную плату;
- сетевая плата: Ethernet 10 Мбит/с;
- дополнительное оборудование: монитор SVGA 1024x768, мышь, клавиатура.

В качестве хранилища резервных копий допустимо использовать следующие типы аппаратного обеспечения:

- съемные жесткие диски с USB-интерфейсом;
- NAS – сетевые системы хранения данных.

Для работы мобильного приложения на мобильном устройстве должно быть не менее 100 Мб свободного пространства постоянной памяти.

Приведенные выше требования к техническим средствам являются минимально допустимыми. Применение более производительных технических средств улучшает функциональные свойства Платформы.

2.4. Уровень квалификации пользователя

Для работы с Платформой пользователь должен обладать:

- знаниями соответствующей предметной области;
- знаниями функциональности Платформы согласно эксплуатационной документации;
- навыками работы на персональном компьютере под управлением ОС Microsoft Windows и Linux;
- навыками использования распространенных веб-браузеров актуальных общедоступных версий.

3. ПОДГОТОВКА К РАБОТЕ

3.1. Общие сведения

Работа пользователя в Платформе возможна, если для пользователя создана учетная запись. Создание учетных записей пользователей входит в обязанности Администратора. Создав учетную запись, Администратор сообщает пользователю ее регистрационное имя, а также пароль для доступа к Платформе.

Серверная часть СПО Платформы запускается автоматически при старте ОС на соответствующих технических средствах.

Клиентская часть СПО Платформы запускается при открытии пользователем Платформы веб-приложения, которое функционирует на клиентском рабочем месте в рамках веб-браузера, при обращении по соответствующему URL-адресу.

3.2. Работа по администрированию Платформы

Для начала работы пользователя в Платформе необходимо открыть веб-браузер, перейти по URL-адресу сервера, на котором развернут проект, реализованный на Платформе. На открывшейся странице входа ввести логин и пароль. Далее откроется страница панели администрирования. Предполагается, что пользователь с ролью «Администратор» был создан в Платформе в процессе ее развертывания на сервере.

3.3. Порядок проверки работоспособности Платформы

Для проверки работоспособности необходимо убедиться, что при первом входе в Платформу не возникает уведомлений об ошибках, корректно отображается экран с информацией о пользователе.

Дальнейшая работа с Платформой описана в разделе 4.

4. РАБОТА С ПЛАТФОРМОЙ

4.1. Общие сведения

Платформа представляет собой решение, на базе которого разрабатываются проекты в сфере экологии: для учета и мониторинга обращения с отходами, мониторинга состояния и загрязнения окружающей среды, атмосферного воздуха и т.п.

Платформа состоит из двух основных частей:

- личного кабинета пользователя – веб-интерфейса проекта, который создается на базе Платформы;
- админ-панели – конструктора Платформы, в котором производится настройка личного кабинета пользователя.

4.2. Авторизация и ролевая модель пользователей

Авторизация пользователей на Платформе доступна одним из способов:

- через ЕСИА (Единую систему аутентификации, используемую в России);
- через токен JWT;
- через логин-пароль.

В Платформе предусмотрена ролевая модель, основанная на группах «разрешений».

Разрешение – это запись в таблице разрешений, содержащая информацию о сущности, для которой определяется разрешение, тип действия (чтение, запись, создание, обновление), а также «отношение». Разрешение определяет, какие действия над какими объектами/типами представления/операциями/переходами и в каком состоянии может выполнять пользователь.

Отношение в контексте разрешения – это условие, сужающее выборку элементов, к которым предоставляется разрешение (например, «только в рамках своей организации»).

Права доступа пользователю на Платформе настраиваются через кабинеты, роли и разрешения.

Кабинет соответствует одному типу организации и определяет какие страницы будут доступны пользователям организаций данного типа. Также в рамках кабинетов можно настраивать доступ к пунктам меню конкретным ролям.

Роль – это группа разрешений, к которой привязывается профиль пользователя.

4.3. Работа с простыми формами

Конструктор позволяет создавать страницы, которые в контексте Платформы называются формами.

Формы создаются в разделе «Формы» административной панели. Формы могут быть следующих видов:

- форма списка,
- форма элемента,
- форма экспорта,
- форма для операции,
- аналитическая панель,
- лендинг.

Для создания формы предварительно должен быть создан тип представления, по которому будет строиться форма. При создании типа представления форма списка и форма элемента создадутся автоматически. Для дальнейшей работы с ними необходимо будет только их настроить.

Тип представления в контексте Платформы – это любой объект учета, с которым пользователю предстоит работать (документ, сотрудник, компания, товар и т.п.).

Таблицы, на которых строится тип представления, могут быть трех типов:

- модель `django`,
- модель `clickhouse`;
- представление.

Представление – это таблица только для чтения на основе введенного SQL-запроса.

На текущий момент часто используемый тип представления – это тип представления, основанный на модели `django`. Его особенность в том, что он описывает сущность, которая хранится в конкретной таблице, описанной в модели.

Чаще всего для одной таблицы создается один тип представления. Однако на Платформе предусмотрена возможность нескольких типов представлений. Чтобы создать для одной модели несколько типов представления, нужно чтобы у модели был атрибут, указывающий на таблицу типов представления. Без этого атрибута информационная система не сможет определить, к какому типу представления относится конкретная запись таблицы, если для этой таблицы их создано несколько.

В контексте Платформы тип представления отвечает за то, как данные будут храниться, форма отвечает за то, как данные будут представляться пользователю.

Форма списка может представлять собой таблицу с перечнем объектов, в которой можно настроить названия колонок и выводимую в них информацию из объектов. Также форма списка объектов может представлять собой географическую карту, на которой размещены точки объектов. Можно настроить, какие поля объектов отвечают за широту и долготу, а также за цвет и иконку маркера на карте.

Форма элемента (или детальная страница) – это обычно или форма для ввода данных, в которой можно настроить перечень полей для редактирования пользователем, или страница со структурированной информацией об объекте, которая представлена в виде нескольких колонок атрибутов объекта и их значений. Также в форме элемента могут быть размещены и сложные комплексные визуальные компоненты, например, визуальный редактор геометрии объекта.

4.3.1. Настройка отображения полей в форме

В таблице 3 представлены описание возможных полей при работе с Платформой.

Таблица 3 – Настройка свойств элемента

№ п/п	Свойство	Назначение	Условие отображения	Пример заполнения
1	Главное			
1.1	Заголовок	Наименование поля в форме. Поддерживаются формулы		Муниципалитет
1.2	Компонент вывода	Выбор из доступных компонентов отображения поля. Список компонентов зависит от типа атрибута		Выбор из списка
1.3	Обязательное	При выставлении данной отметки поле в форме будет обязательным для заполнения		Чекбокс
1.4	Отключено	При выставлении данной отметки поле в форме не будет отображаться		Чекбокс
1.5	Только для чтения	При выставлении данной отметки поле в форме нельзя будет редактировать		Чекбокс
1.6	Показывать только по выбору пользователя	При выставлении данной отметки поле в форме будет отображаться, если только пользователь в настройках отображения формы в Личном кабинете отметит данное поле		Чекбокс
1.7	Путь к данным	Поле, из которого будут подтягиваться значения		borough
1.8	Lookup	При выборе Lookup меняется способ фильтрации основного списка по данному полю	Доступно только для компонентов вывода в фильтре: однострочный текст, многострочный текст, расписание, поле ввода числа	Выбор из списка
1.9	Доступно для сортировки	При выставлении данной отметки		Чекбокс

№ п/п	Свойство	Назначение	Условие отображения	Пример заполнения
		становится доступна сортировка по полю в форме списка		
1.10	Является ссылкой для перехода из списка в детальную страницу	При выставлении данной отметки по полю в форме можно будет переходить в детальную страницу		Чекбокс
1.11	По умолчанию истина	При выставлении данной отметки в поле по умолчанию будет устанавливаться значение «Истина»	Доступно только для компонента «Чекбокс»	Чекбокс
1.12	Атрибут названия связанного элемента	Указывает, из какого атрибута брать название элемента для отображения в выпадающем списке (по умолчанию – name)	Доступно только для компонента «Выбор из внешней таблицы»	name
1.13	Базовая ссылка на API	Дает возможность переопределить ссылку, по которой выпадающий список будет запрашивать элементы для выбора. Обычно в уже существующую ссылку добавляют get-параметры фильтров или сортировки для управления выпадающим списком элементов	Доступно только для компонента «Выбор из внешней таблицы»	/api/wf__waste_site/?ordering=-address®ion=12 для табов Сортировка статуса в зависимости от проставленного «Порядка»: /api/wf__waste_site/?ordering=sort
1.14	Тип представления	Позволяет ограничить выборку элементов выпадающего списка каким-то типом представления (для случаев, когда по одной таблице доступно несколько типов представления). Например, физ. и юр. лица		Выбор из списка
1.15	Показывать кнопку редактирования объекта	Добавляет кнопку перехода к редактированию выбранного в выпадающем списке объекта (требуется выбор формы редактирования)	Доступно только для компонентов «Выбор из внешней таблицы», «Множественный выбор из внешней	Чекбокс

№ п/п	Свойство	Назначение	Условие отображения	Пример заполнения
			таблицы», «Файлы»	
1.16	Показывать кнопку создания объекта	Добавляет кнопку перехода к созданию нового объекта для выбора в выпадающем списке (требуется форма редактирования)	Доступно только для компонентов «Выбор из внешней таблицы», «Файлы»	Чекбокс
1.17	Разрешить выбор из расширенного списка	Показывает рядом с выпадающим списком иконку для перехода на отдельную страницу списка элементов для выбора	Доступно только для компонентов «Выбор из внешней таблицы», «Файлы»	Чекбокс
1.18	Поле широты А	Задаёт название атрибута, из которого будет подтягиваться значение широты для точки А на изображении маршрута	Доступно только для компонента «Маршрут на карте»	waste_source.lat
1.19	Поле долготы А	Задаёт название атрибута, из которого будет подтягиваться значение долготы для точки А на изображении маршрута	Доступно только для компонента «Маршрут на карте»	waste_source.lon
1.20	Поле широты В	Задаёт название атрибута, из которого будет подтягиваться значение широты для точки В на изображении маршрута	Доступно только для компонента «Маршрут на карте»	waste_site.lat
1.21	Поле долготы В	Задаёт название атрибута, из которого будет подтягиваться значение долготы для точки В на изображении маршрута	Доступно только для компонента «Маршрут на карте»	waste_site.lon
2	Карта			
2.1	Показывать выбранный элемент на карте	Отображает под выпадающим списком карту, на которой можно кликом выбирать элемент в дополнение к выбору из выпадающего списка	Доступно только для компонента «Выбор из внешней таблицы»	Чекбокс
3	Дополнительные параметры			
3.1	Query-параметр для фильтрации по тексту (по	Значение параметра, по которому будет производиться поиск элементов при	Доступно только для компонента «Выбор из	address__icontains

№ п/п	Свойство	Назначение	Условие отображения	Пример заполнения
	умолчанию «search»)	вводе текста в поисковую строку открытого выпадающего списка	внешней таблицы»	
3.2	Query-параметр для фильтрации по списку UID (по умолчанию «{primaryKey}__in»)		Доступно только для компонента «Выбор из внешней таблицы»	
3.3	Значение по умолчанию	Указанное значение свойства будет отображаться в форме по умолчанию	Доступно только для полей ввода (не выбор из внешней таблицы)	
3.4	Автодополнение: Количество символов для подгрузки списка	Минимальное количество введенных символов, которые нужно ввести, чтобы был отправлен запрос поиска подсказок	Доступно только для текстовых полей	
3.5	Автодополнение	Включает запрос подсказок, начиная с определенного количества символов ввода. Подсказки загружаются из выбранного источника	Доступно только для текстовых полей	
3.6	Маска	Выбор текстовой маски из доступных. Новые маски создаются в форме конструктора «Текстовые маски»	Доступно только для текстовых полей	Выбор из списка
3.7	Тема	Выбор варианта визуального отображения	Доступно только для компонентов, в которых доступны различные варианты отображения (например, чекбокс)	Выбор из списка
3.8	Точность	Год – в календаре будут доступны только года; Месяц – в календаре будут доступны месяцы и года;	Доступно только для компонентов вывода «Дата» и «Дата со временем»	Выбор из списка

№ п/п	Свойство	Назначение	Условие отображения	Пример заполнения
		<p>День – в календаре будут доступны дни, месяцы, года;</p> <p>Часы – рядом с поле даты появится дополнительное поле со временем с выбором только часов;</p> <p>Минуты – рядом с поле даты появится дополнительное поле со временем с выбором часов и минут;</p> <p>Секунды – рядом с поле даты появится дополнительное поле со временем с выбором часов, минут и секунд</p>		
3.9	Тема вывода значения (в режиме «Просмотр данных»)	Выбор отображения даты и времени в одну или в две строки в режиме просмотра. По умолчанию отображается в две строки (время переносится на вторую строку)	Доступно только для компонентов вывода «Дата» и «Дата со временем»	Выбор из списка
3.10	Скрывать заголовок поля	При выставлении данной отметки заголовок поля в форме отображаться не будет		Чекбокс
3.11	Жирный стиль (в режиме «Просмотр данных»)	При выставлении данной отметки поле в форме будет выделено жирным стилем		Чекбокс
3.12	Ссылка (в режиме «Просмотр данных»)	Если задано, то в режиме просмотра данных (обычно это просто текст) текст значения будет являться ссылкой на указанный здесь URL		
3.13	Шаблон вывода значения (в режиме «Просмотр данных»)	Позволяет изменить выводимый текст компонента в режиме «Только чтение»		<p>Элемент №{data.id}</p> <p>Для отображения : "null" \ "Нет""Да" , при data.free= null\True\False { data.free== null? null: (data.free?</p>

№ п/п	Свойство	Назначение	Условие отображения	Пример заполнения
				"Нет" : "Да")}
3.14	Максимальная длина текста (в режиме «Просмотр данных»)	Максимальное количество отображаемых в поле символов в режиме просмотра данных		
3.15	Обрезать текст до последнего пробела (в режиме «Просмотр данных»)	При выставлении данной отметки текст в поле отображается только до последнего пробела		Чекбокс
3.16	Светофор: Тема (в режиме «Просмотр данных»)	Вид отображения заполненной в поле информации в режиме «Только чтение» – это может быть просто текст или стилизованный элемент (плашка, прогресс-бар и т.д.)		Выбор из списка
3.17	Светофор: Цвет (в режиме «Просмотр данных»)	Цвет фона или точки или прогрессбара для выбранной темы отображения		Выбор из списка
3.18	Светофор: Иконка (в режиме «Просмотр данных»)	Иконка стилизации для некоторых тем в режиме «Светофор»		Выбор из списка
3.19	Светофор: Значение прогресса (в режиме «Просмотр данных»)	Значение прогресса на прогресс-баре от 0 до 100		
3.20	Светофор: Скрыть значение (в режиме «Просмотр данных»)	Используется, если нужно показать только точку светофора, а само текстовое значение не показывать		Чекбокс
3.21	Ширина столбца, относительно остальных (в форме списка)			Выбор из списка
3.22	Аннотация к полю	Текст будет доступен под знаком вопроса справа от названия поля		

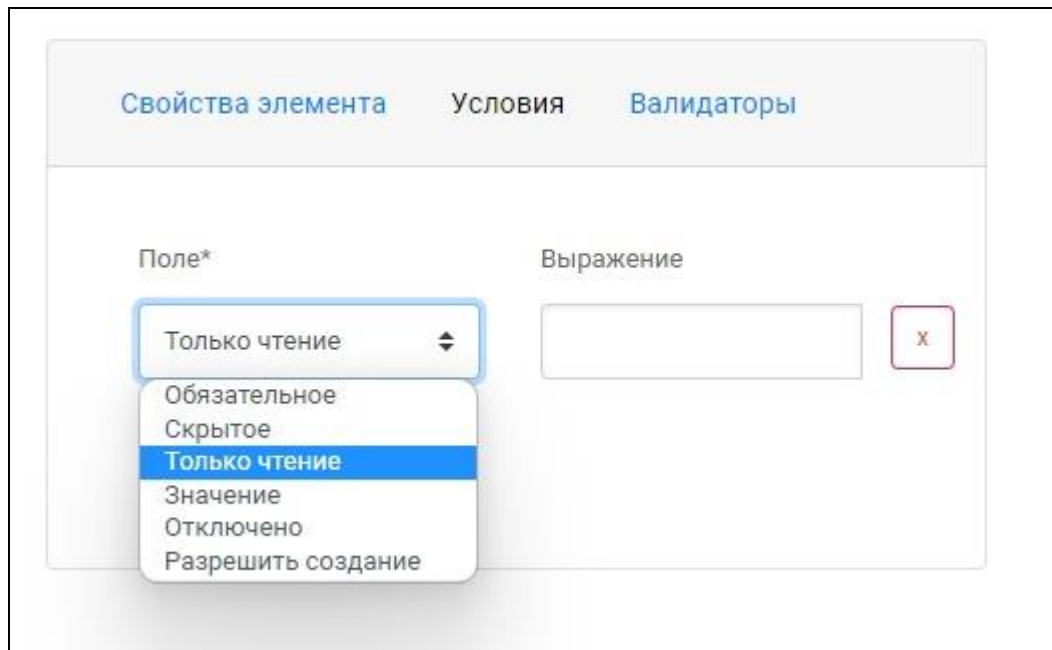
№ п/п	Свойство	Назначение	Условие отображения	Пример заполнения
3.23	Количество занимаемых колонок 1-12 (не в форме списка)	Задаёт относительную ширину блока внутри группы полей		
3.24	Количество колонок в форме, 1-12	Для группы с типом «Колонки» – все поля или группы полей внутри нее разместятся горизонтально		
3.25	Вид группы	Обычная, раскрываемая (свернутая), раскрываемая (развернутая), колонки		Выбор из списка
3.26	Цвет линии	Выбор цвета линии маршрута на карте	Доступно только для компонента «Маршрут на карте»	
3.27	Иконка точки А	Выбор иконки точки А, отображаемой на карте	Доступно только для компонента «Маршрут на карте»	
3.28	Цвет точки А	Выбор цвета точки А на карте	Доступно только для компонента «Маршрут на карте»	
3.29	Иконка точки В	Выбор иконки точки В, отображаемой на карте	Доступно только для компонента «Маршрут на карте»	
3.30	Цвет точки В	Выбор цвета точки В на карте	Доступно только для компонента «Маршрут на карте»	
3.31	Тип подложки карты		Доступно только для компонента «Маршрут на карте»	
3.32	Поиск	Свойство включает поиск по адресу на карте	Доступно только для компонента «Маршрут на карте»	

№ п/п	Свойство	Назначение	Условие отображения	Пример заполнения
3.33	API списка точек	Позволяет разместить в компоненте маршрута еще точки других объектов	Доступно только для компонента «Маршрут на карте»	
3.34	Разрешить переход на другие объекты	Разрешает переход по клику на точки объектов на карте (возможно только если объекты одного типа с текущим)	Доступно только для компонента «Маршрут на карте»	
3.35	Максимальный размер (МВ)	Максимально допустимый размер загружаемых файлов в Мб	Доступно только для компонента «Файлы»	100
3.36	Максимальное кол-во файлов	Максимально допустимое количество загружаемых файлов	Доступно только для компонента «Файлы»	10
3.37	Поддерживаемые форматы файлов (пример "image/*,application/pdf,.psd")	Перечисление поддерживаемых форматов через запятую	Доступно только для компонента «Файлы»	
4	Автокомплит			
4.1	Связанная форма списка	Выбирает форму списка элементов, которая укажет информационной системе, откуда загружать элементы для выпадающих подсказок		
4.2	Базовая ссылка на API	Позволяет переопределить базовую ссылку выбранной формы (добавить фильтры, сортировку и т.д.)		
4.3	Поле вывода значения	Поле, которое будет отображаться в каждой строке выпадающих подсказок		
4.4	Поле для фильтрации	Поле, по которому будет осуществляться поиск подсказок		
5	Роли и статусы			
5.1	Отображать только для статусов	Выбор статусов, при которых в форме будет отображаться поле		Выбор из списка

№ п/п	Свойство	Назначение	Условие отображения	Пример заполнения
5.2	Отображать только для ролей	Выбор ролей, для которых в форме будет отображаться поле		Выбор из списка
5.3	Разрешено редактирование для статусов	Выбор статусов, при которых будет доступно редактирование поля в форме		Выбор из списка
5.4	Разрешено редактирование для ролей	Выбор ролей, для которых будет доступно редактирование поля в форме		Выбор из списка
5.5	Разрешено создание для ролей	<p>В связанных списках регулирует наличие кнопки «Создать» во вкладке связанного списка.</p> <p>В выпадающих списках регулирует наличие кнопки «+» напротив выпадающего списка</p>	Только для связанных списков и выпадающих списков	Выбор из списка
5.6	Разрешено создание для статусов		Только для связанных списков	Выбор из списка
5.7	Только чтение для статусов	Выбор статусов, при которых поле будет доступно только для чтения		Выбор из списка
5.8	Только чтение для ролей	Выбор ролей, для которых поле будет доступно только для чтения		Выбор из списка

На Платформе предусмотрена возможность настройки условий и валидаторов.

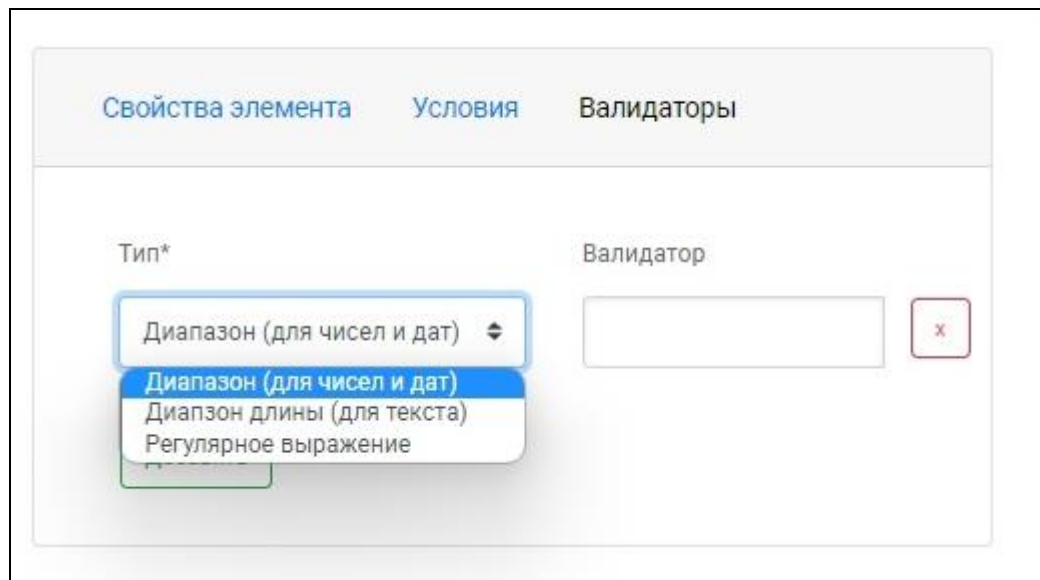
Условия позволяют формулам задавать критерии отображения полей (рис. 1).



Настройка условий отображения полей

Рис. 1

Валидаторы позволяют ввести дополнительные ограничения на ввод данных в поле. В качестве валидатора также можно использовать регулярные выражения (рис. 2).



Настройка валидаторов

Рис. 2

4.3.2. Создание и первичная настройка типа представления

Все формы на Платформе строятся на основе типов представления. Тип представления характеризует любую группу объектов в информационной системе. Тип представления привязывается к таблице базы данных, благодаря чему становится возможной дальнейшая работа с объектами из таблицы базы данных.

Для создания типа представления и его основной настройки необходимо выполнить следующие действия:

1. Перейдите в раздел конструктора «Документооборот → 1. Типы представлений».
2. Нажмите кнопку «Добавить тип представления».
3. Откроется форма создания типа представления. Заполните поля в форме:
 - в поле «Наименование» укажите удобное для пользователя наименование типа представления, которое отражает суть хранимых в типе представления данных;
 - в поле «Код» укажите уникальный код типа представления, используя буквы латинского алфавита. Код будет использоваться в клиентской и серверной частях конструктора для доступа к данным типа представления. Можно также не заполнять поле, тогда код сгенерируется автоматически;
 - в поле «Тип таблицы» выберите тип:
 - «Модель django» – будет создана сущность на основе таблицы в PostgreSQL,
 - «Представление» – будет создана таблица только для чтения на основе введенного SQL-запроса. По сути, это есть витрина данных,
 - «Модель clickhouse» – будет создана сущность на основе таблицы в Clickhouse. Данный тип следует выбирать, если в сущности планируется хранить данные, в которых не так важна точность (например, данные с датчиков);
 - в поле «Таблица» выберите таблицу БД (PostgreSQL или Clickhouse в зависимости от выбранного типа таблицы), в которой будут храниться данные этого типа представления. Если подходящей таблицы нет, то необходимо обратиться к разработчикам Платформы для того, чтобы создать ее;
 - если в выбранной таблице есть поле со статусом, то также обязательно необходимо заполнить поле «Статус по умолчанию» (рис. 3).

Изменить Тип представления

Загрязняющие вещества

Основные данные

Наименование:	<input type="text" value="Загрязняющие вещества"/>	Код:	<input type="text" value="pollutant"/>
Заголовок:	<input type="text"/>	Наименование (мн. число):	<input type="text" value="Загрязняющие вещества"/>
Тип таблицы:	<input type="text" value="Модель django"/>		
Таблица:	<input type="text" value="workflow Справочники и реестры"/> x + 👁 <small>По умолчанию - workflow.Docitem</small>		

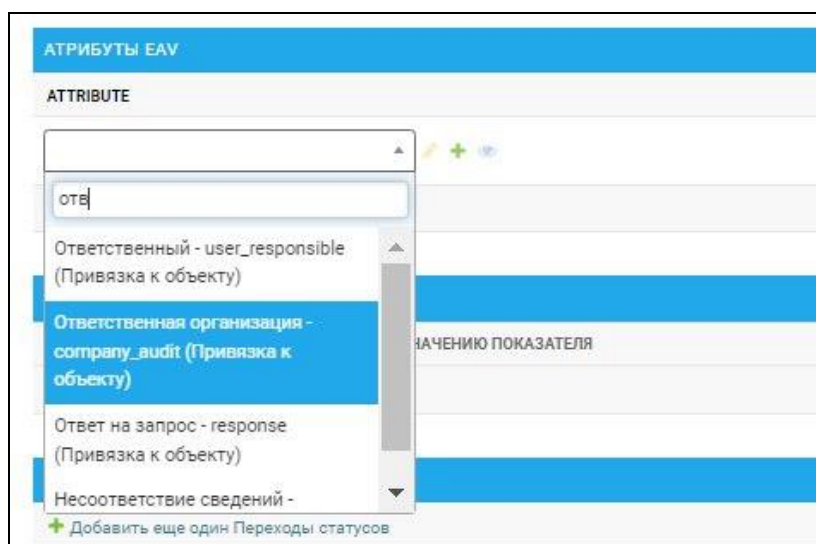
Параметры

- Активен
- Показывать печатные формы
- Разрешить обсуждение
- Хранить значения в отдельной таблице: ✖
Для изменения значений воспользуйтесь командой в списке
- Передавать данные на шину
- Получать данные с шины
- Запретить удаление
- Запретить изменение

Форма создания типа представления и его основные настройки

Рис. 3

4. При необходимости добавьте дополнительные атрибуты к типу представления в подразделе формы «Атрибуты EAV»: нажмите кнопку «Добавить еще один Связь doctype-attribute» и в появившемся поле можно выбрать атрибут из ранее созданных (рис. 4).



Добавление дополнительных атрибутов к типу представления

Рис. 4

Создать новый атрибут можно и с помощью кнопки «+» напротив соответствующего поля. После нажатия кнопки, откроется форма создания атрибута, со следующими полями:

- в поле «Тип данных» выберите тип. В зависимости от выбранного типа в форме станут доступны дополнительные поля;
- в поле «Required» поставьте отметку, если поле должно быть обязательным для заполнения;
- в поле «Символьный код» укажите уникальный код, используя буквы латинского алфавита;
- в поле «Название» укажите удобное для пользователя название атрибута;
- в поле «Описание» укажите краткое предназначение атрибута.

Для поля с типом «Привязка к объекту» дополнительно необходимо в поле «Related table» указать таблицу, к объектам которой будет привязан этот атрибут.

Для поля с типом «Выбор из списка» дополнительно необходимо в поле «Группа выбора из списка» выбрать группу из уже созданных или создать новую, нажав кнопку «+» и перечислив все значения для выпадающего списка (рис. 5).

Изменить Атрибут

Допустимое значение - max_value (Дробное число)

Тип данных:

Required

Символьный код:
Short unique attribute label

Название:
User-friendly attribute name

Описание:
Short description

РАЗРЕЗЫ АТТРИБУТОВ

РАЗРЕЗ	УДАЛИТЬ?
+ Добавить еще один Разрезы атрибутов	

Выбор группы из списка

Рис. 5

5. Сохраните изменения в форме создания типа представления.

4.3.3. Создание и настройка формы списка

В форме списка отображается список объектов выбранного типа представления.

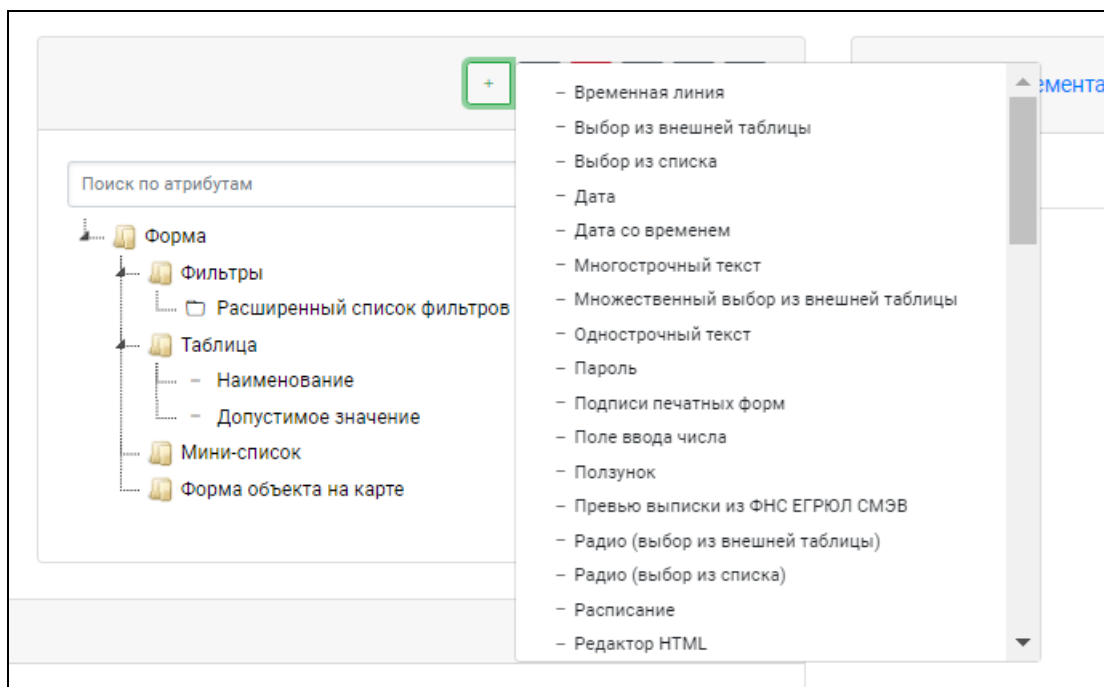
Форма списка создается автоматически после создания типа представления. Для того чтобы настроить форму списка, необходимо выполнить следующие действия:

1. Перейдите в раздел конструктора «Конструктор форм → 1. Формы». Найдите форму списка по созданному ранее типу представления.

2. В подразделе формы «Конструктор» в блоке «Структура данных» отображаются атрибуты типа представления, доступные для добавления на форму списка. В блоке посередине отражается древовидная структура формы. Чтобы вывести атрибут на форму в виде поля:

- в среднем блоке нажмите на папку в структуре, куда необходимо добавить атрибут.

При этом можно создавать новые группы и добавлять новые элементы (рис. 6);

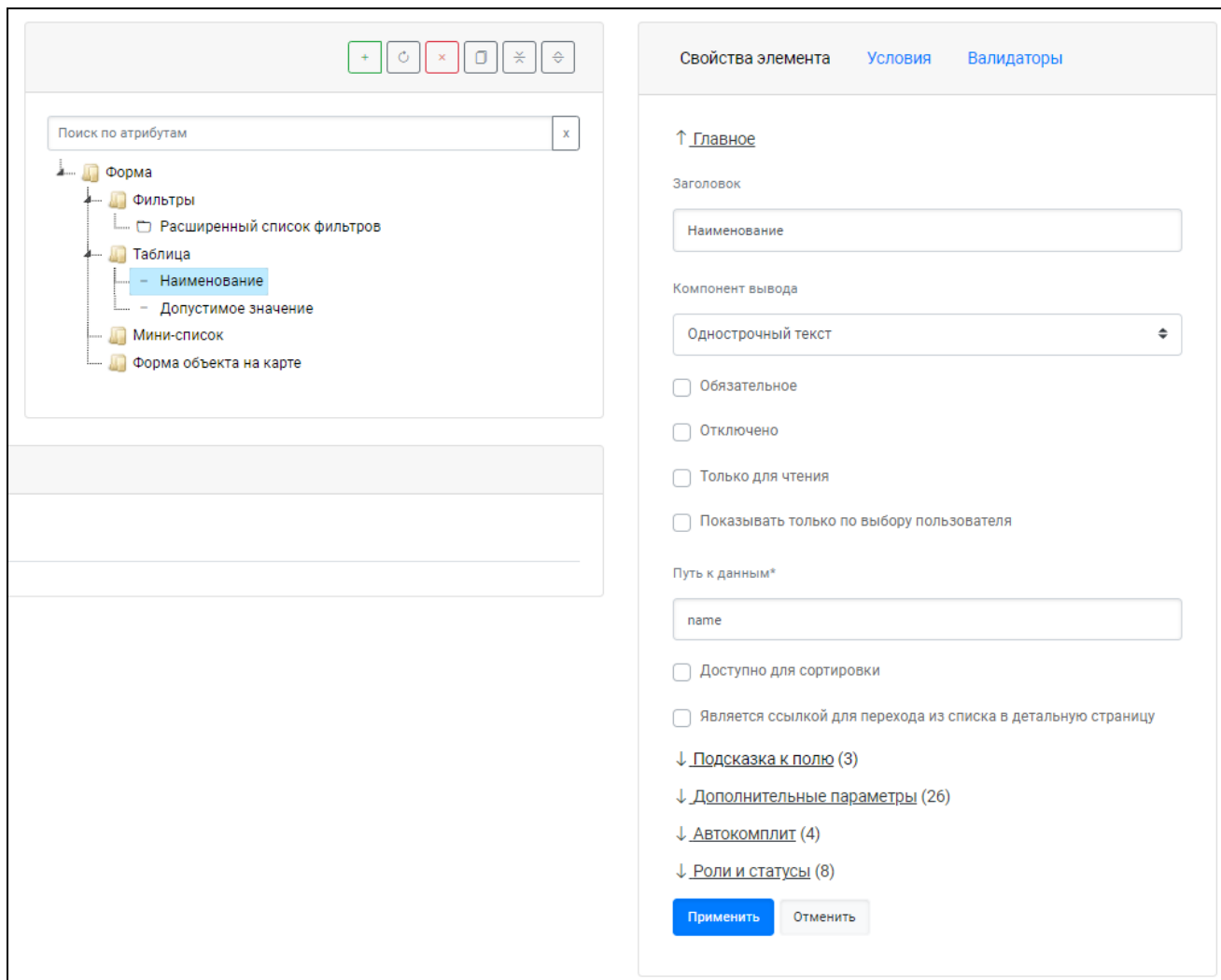


Форма создания новой группы и добавления нового элемента

Рис. 6

- в блоке «Структура данных» найдите атрибут, который необходимо добавить на форму списка. Двойным кликом по атрибуту добавьте его на форму. Атрибут отобразится в среднем блоке в папке, которая была выбрана до добавления. При этом атрибуты можно перемещать по структуре перетаскиванием.

3. Нажмите на добавленный атрибут и настройте его отображение (поле в форме) в третьем блоке конструктора формы (рис. 7). Описание настроек отображения полей приведено в п. 4.3.1 настоящего документа.



Настройка отображения добавленного атрибута

Рис. 7

4. Выведите форму в меню Личного кабинета. Для этого:
 - перейдите в раздел конструктора «Конструктор форм → 2. Главное меню»;
 - нажмите кнопку «Добавить пункт главного меню»;
 - заполните форму (рис. 8):

Изменить Пункт главного меню

Загрязняющие вещества

Наименование:

Код:

Активен

Изображение иконки: Файл не выбран

Код иконки:

Типы компаний:

- Оператор объекта обращения
- РОИВ
- Транспортировщик
- Отходообразователь
- Посетитель
- Оператор комплексной услуги
- Управляющая компания
- Орган местного самоуправления
- Тестовый объект

Удерживайте "Control" (или "Command" на Mac), чтобы выбрать несколько значений.

Форма списка:

Форма объекта:

Форма лендинга:

Родитель:

URL:

Если заполнен, ссылка главного меню будет вести на данных url

Изображение баннера: Файл не выбран

Пример заполнения формы пункта главного меню

Рис. 8

- в поле «Наименование» укажите удобное для пользователя название пункта меню, которое будет отображаться в интерфейсе,
- в поле «Код» укажите уникальный код пункта меню, используя буквы латинского алфавита,
- в поле «Активен» поставьте отметку, чтобы пункт меню отображался в интерфейсе,
- в поле «Код иконки», нажав кнопку «Q» можно выбрать иконку из уже доступных в Платформе или загрузить новую,

- в поле «Типы компаний» выберите все типы компаний, для которых будет доступен данный пункт меню,
- в поле «Форма списка» выберите форму списка, которая должна открываться при переходе по пункту меню,
- в поле «Родитель» выберите родительский пункт меню, если необходимо, чтобы создаваемый пункт меню был вложенным (не первого уровня),
- в поле «URL» при необходимости укажите адрес (можно как внутри информационной системы, так и внешний), на который будет осуществляться переход из пункта меню,
- в поле «Индекс сортировки» укажите число, от которого будет зависеть расположение пункта меню среди других пунктов. Чем меньше число, тем выше пункт в списке,
- в поле «Только для ролей» выберите все роли, для которых будет доступен данный пункт меню.

5. Форму списка теперь можно просматривать в личном кабинете (рис. 9). При нажатии на элемент списка будет открываться детальная страница, если она создана для данного типа представления.

Загрязняющие вещества

Список + Создать

Наименование

Введите

Применить Сбросить

Всего записей: 34

НАИМЕНОВАНИЕ	символьный код	пдк
Диоксид серы	Не заполнено	Атмосферный воздух 0.5 мг/м3
Формальдегид	Не заполнено	Атмосферный воздух 0.035 мг/м3
Толуол	Не заполнено	Атмосферный воздух 50.0 мг/м3

« < 1 2 3 4 > » 10

Просмотр формы списка через личный кабинет пользователя

Рис. 9

4.3.4. Создание и настройка детальной страницы

Детальная страница отражает данные по одному объекту выбранного типа представления.

Детальная страница также создается автоматически после создания типа представления и привязывается к форме списка, также созданной автоматически. Т.е. данная форма будет открываться при нажатии на какую-либо запись в форме списка.

Кроме этого, для одного типа представления можно создать несколько детальных страниц и привязать в форме списка нужную детальную страницу.

Для создания детальной страницы необходимо выполнить следующие действия:

1. Перейдите в раздел конструктора «Конструктор форм → 1. Формы».
2. Нажмите кнопку «Добавить форму». Заполните форму (рис. 10):

Изменить Форма

ИСТОРИЯ

Форма элемента "Вещество" для Загрязняющие вещества

Основные данные

Name: Вещество По умолчанию - Основная Код: veshestvo_object337 По умолчанию формируется как DocType.code-listform

Тип формы: Форма элемента

Активен: Да

Использовать по умолчанию

Тип представления: Загрязняющие вещества

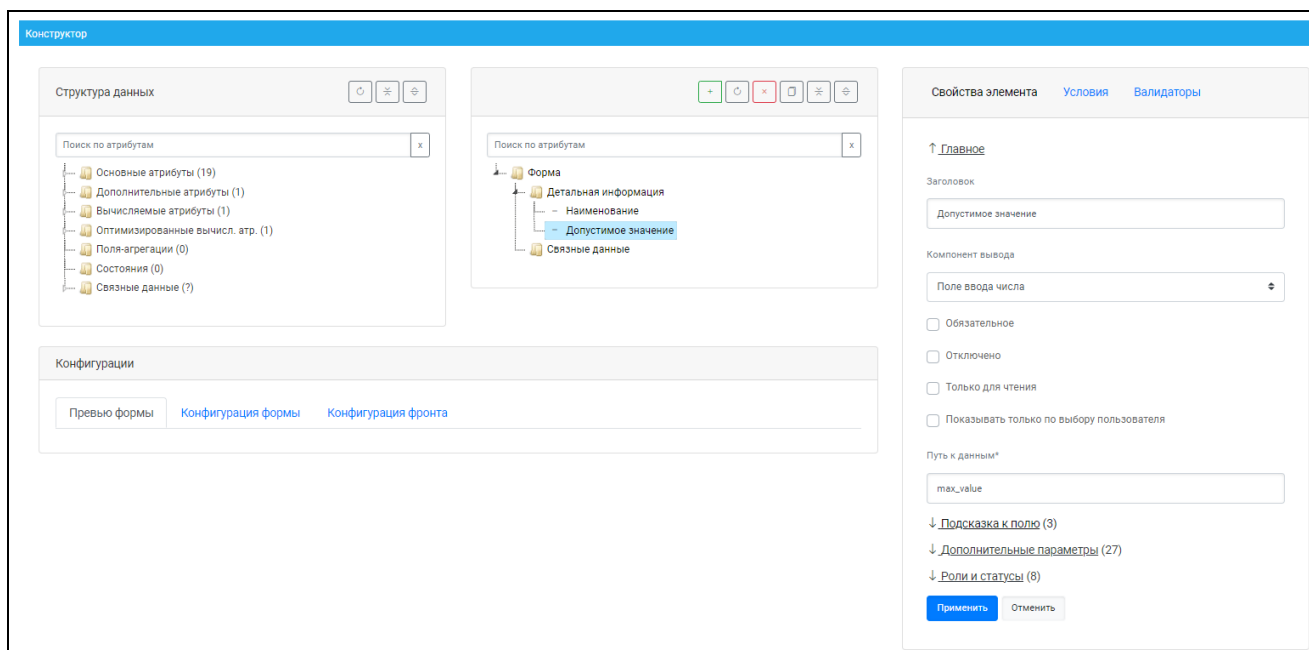
Заполнение формы при создании детальной страницы

Рис. 10

- в поле «Name» укажите удобное для пользователя название формы;
- в поле «Код» можно указать уникальный код формы или оставить значение, сформированное автоматически;
- в поле «Тип формы» выберите «Форма элемента»;
- в поле «Активен» оставьте выбранное по умолчанию значение «Да», если необходимо, чтобы форма отображалась в интерфейсе;
- в поле «Тип представления» выберите тип представления, для которого необходимо создать детальную страницу.

3. Сохраните изменения.

4. После сохранения формы станут доступны расширенные настройки формы. В разделе «Конструктор» добавьте и настройте необходимые поля в форме (рис. 11). Описание настроек отображения полей приведено в п. 4.3.1 настоящего документа.



Настройка полей в форме детальной страницы

Рис. 11

5. Снова сохраните изменения.

4.4. Использование формул в конструкторе

4.4.1. Общее описание и контексты данных

В конструкторе реализована возможность обработки данных формы с помощью формул. Например, можно вывести значение поля в зависимости от выполнения условий; вычислить значение поля, используя различные арифметические, статистические функции; можно скрыть поле, сделать его обязательным, изменить его внешний вид в зависимости от значений других полей и т.п.

За обработку формул в конструкторе отвечает фронтенд-модуль «Агент формул». Поскольку формулы выполняются на серверной части Платформы, то изменения, которые вносит пользователь, отражаются в информационной системе сразу.

Формулы могут быть добавлены в свойства и условия атрибутов.

Формулы оформляются в фигурных скобках с соблюдением синтаксиса библиотеки `math.js`.

В формулах можно обращаться к различным данным из форм и объектов (например, использовать значение другого поля формы или получить данные пользователя). Для обращения к таким данным используются контексты данных.

Контекст данных – объекты, в которых хранятся данные, к которым формулы имеют доступ. В разных типах форм доступны разные контексты данных. На текущий момент, в формах доступны следующие контексты:

Форма сущности (форма и поля формы):

data – текущая сущность;

user – объект текущего пользователя;

app – контекст для определения мобильной версии.

Форма перехода статуса сущности (поля формы):

data – текущая сущность;

user – объект текущего пользователя;

app – контекст для определения мобильной версии.

Форма списка (только форма):

data – родительская сущность списка, если есть (иначе контекст недоступен и при обращении к нему будет появляться ошибка);

user – объект текущего пользователя;

app – контекст для определения мобильной версии;

filter – форма фильтра;

filterSaved – сохраненные данные формы фильтра;

parentFilter – родительская форма фильтра на мультикарте, если есть (иначе контекст недоступен и при обращении к нему будет появляться ошибка);

item – сущность элемента списка (недоступен в формулах, но доступен в коде для того, чтобы получить зависимости от данных элемента списка).

Форма элемента списка (форма и поля формы):

data – форма текущего элемента списка;

user – объект текущего пользователя;

app – контекст для определения мобильной версии;

parent – родительская сущность списка, если есть (иначе контекст недоступен и при обращении к нему будет появляться ошибка);

filter – форма фильтра списка;

filterSaved – сохраненные данные формы фильтра;

parentFilter – родительская форма фильтра на мультикарте, если есть (иначе контекст недоступен и при обращении к нему будет появляться ошибка).

Форма фильтра списка (поля формы):

data – форма фильтра;

user – объект текущего пользователя;

app – контекст для определения мобильной версии;

filterSaved – сохраненные данные формы фильтра;

parent – родительская сущность списка, если есть (иначе контекст недоступен и при обращении к нему будет появляться ошибка);

parentFilter – родительская форма фильтра на мультикарте, если есть (иначе контекст недоступен и при обращении к нему будет появляться ошибка).

Форма родительского фильтра списка (поля формы):

data – форма фильтра;

user – объект текущего пользователя;

app – контекст для определения мобильной версии.

Форма операции (поля формы):

data – форма операции;

user – объект текущего пользователя;

app – контекст для определения мобильной версии.

В большинстве случаев одноименные контексты несут одну и ту же логику. Однако необходимо обращать внимание на тип формы, в котором создаются формулы, т.к. в некоторых случаях логика может отличаться:

data – в большинстве случаев означает контекст текущей формы, за исключением случаев, когда у формы есть родительская форма. В контекст data попадает модель данных, которая описывается ObjectModel формы;

user – во всех формах означает объект текущего пользователя. В контекст попадает объект пользователя с данными, которые получены после запроса на api/auth;

`app` – контекст `app` доступен во всех формах и полях форм (также, как контекст `user`);

`filter` – все, что попадает в контекст `data` в форме фильтра, является контекстом `filter` в остальных формах, где доступен данный контекст. В контекст попадают данные, которые введены в форму фильтра и при этом еще не применены. При вводе значения в поле фильтра агент формул сразу пересчитывает формулы, в которых используется данное поле;

`filterSaved` – в данный контекст попадают значения полей фильтра после применения. Если после применения фильтра будут внесены изменения в поля фильтра, но без повторного применения, то будет меняться только контекст `filter`;

`parent` – все, что попадает в контекст `data` в форме сущности, является контекстом `parent`, если сущность является родительской для сущности списка. В некоторых видах форм такие данные находятся в контексте `data`. Если у формы нет родительской сущности, то при обращении к данному контексту будет выводиться ошибка;

`parentFilter` – контекст доступен только для мультикарт (страниц, на которых находится несколько форм списков). Включает в себя фильтр, доступный для всех форм списков на мультикарте. Основные операторы, доступные в формулах.

На Платформе при создании формул можно использовать любые операторы из библиотеки `math.js`.

В таблице 4 представлены основные операторы, которые могут быть полезны при работе с Платформой.

Таблица 4 – Основные операторы данных

Оператор	Название	Пример использования
,	Разделитель параметров и элементов	<code>max(2, 1, 5)</code>
<code>a(b)</code>	Вызов метода	<code>str(1, 2) → "12"</code>
{ }	Создание объекта	<code>{a: 1, b: 2}</code>
[]	Создание массива	<code>[1, 2, 3]</code>
(a)	Группировка части выражения	<code>2 * (3 + 4) → 14</code>
<code>a.b</code>	Обращение к свойствам	<code>obj={a: 12}; obj.a → 12</code>
<code>a[b]</code>	Динамическое обращение к свойствам	<code>obj={a: 12}; x='a'; obj[x] → 12</code>
<code>a ? b : c</code>	Условное выражение, где: <code>a</code> – условие, <code>b</code> – значение, если условие истинно, <code>c</code> – если ложно	<code>15 > 100 ? 1 : -1 → -1</code>
<code>a and b</code>	Конъюнкция	<code>true and false → false</code>
<code>a or b</code>	Дизъюнкция	<code>true or false → true</code>

Оператор	Название	Пример использования
$a \text{ xor } b$	Строгая дизъюнкция	$\text{true xor true} \rightarrow \text{false}$
$\text{not } a$	Отрицание	$\text{not true} \rightarrow \text{false}$
$a + b$	Сложение	$4 + 5 \rightarrow 9$
$a - b$	Вычитание	$7 - 3 \rightarrow 4$
$a * b$	Умножение	$2 * 3 \rightarrow 6$
a / b	Деление	$6 / 2 \rightarrow 3$
$a ^ b$	Возведение в степень	$2 ^ 3 \rightarrow 8$
$a \% b$	Остаток	$8 \% 3 \rightarrow 2$
$a\%$	Процент	$8\% \rightarrow 0.08$
$+a$	Унарный плюс. Приведение к числу	$+" 17.56" \rightarrow 17.56$
$-a$	Унарный минус	$-" 17.56" \rightarrow -17.56$
$a == b$	Равенство	$2 == 4 - 2 \rightarrow \text{true}$
$a != b$	Неравенство	$2 != 3 \rightarrow \text{true}$
$a < b$	Меньше	$2 < 3 \rightarrow \text{true}$
$a > b$	Больше	$2 > 3 \rightarrow \text{false}$
$a <= b$	Меньше или равно	$4 <= 3 \rightarrow \text{false}$
$a >= b$	Больше или равно	$2 + 4 >= 6 \rightarrow \text{true}$

4.4.2. Доступные методы в формулах

На Платформе при создании формул используются большинство методов из библиотеки `math.js`. Кроме этого, имеются методы, реализованные непосредственно на Платформе.

В таблице 5 представлены основные методы в формулах, которые могут быть полезны при работе с Платформой.

Таблица 5 – Основные методы в формулах

Функция	Название	Параметры	Пример использования
<code>str(x, y, z...): string</code>	Воспринимает все переданные аргументы как строки и конкатенирует их в одну строку, которая возвращается в результате. Конвертирует переданные аргументы в строку, если это не строки	<code>x, y, z</code> – параметры в формате строк, чисел, обращений к объектам и т.п.	<code>str('Строка') // 'Строка'; str('Строка1', 'Строка2') // 'Строка1Строка2'; str(user.firstName, ' ', user.lastName) // 'Иван Иванов'</code>
<code>date(raw): FormcyMoment</code>	Создание объекта FormcyMoment	Необязательный параметр raw – значение в формате <code>date</code> , <code>datetime</code> или <code>local_moment</code> . Если ничего не указано, то будет использована текущая дата и время	
<code>has(object: object, prop: string): boolean</code>	Проверяет, что в объекте есть свойство. Вернет <code>true</code> , если в <code>object</code> есть свойство с названием, указанным в <code>prop</code> . Иначе вернет <code>false</code>		<code>has(user.roles, 'admin') // true/false, has(data.roles, 0) // вернет true, если есть хотя бы один элемент в списке data.roles</code>
<code>print(template, values): string</code>	Интерполирует значения <code>values</code> в строковый шаблон <code>template</code>	template: <code>string</code> – строка, содержащая плейхолдеры переменных; values: <code>Object/Array/Matrix</code> – объект или массив, содержащий переменные, которые будут заполнены в шаблоне	<code>print('За год было проведено \$num проверок', { num: 1235 }) // 'За год было проведено 1235 проверок'</code>
<code>boolean(x): boolean / Array / Matrix</code>	Создает логическое значение или преобразует строку или число в логическое значение	<code>x: string / number / boolean / Array / Matrix / null</code>	<code>boolean([1, 0, 1, 1]) // [true, false, true, true]</code>
<code>number(x): number / Array / Matrix</code>	Создает число или преобразует строку, логическое значение или единицу измерения в число	<code>x: string / number / BigNumber / Fraction / boolean / Array / Matrix / Unit / null</code>	<code>number([true, false, true, true]) // [1, 0, 1, 1]</code>
<code>distance([x1, y1], [x2, y2]):</code>	Вычисляет евклидово	<code>x: Array / Matrix / Object</code> – координаты первой	<code>distance({pointOneX: 0,</code>

Функция	Название	Параметры	Пример использования
Number / BigNumber	расстояние между двумя точками в N-мерных пространствах, расстояние между точкой и линией в 2-х и 3-х мерных пространствах, попарное расстояние между набором 2D- или 3D-точек	точки, y : Array / Matrix / Object – координаты второй точки	pointOneY: 0}, {pointTwoX: 10, pointTwoY: 10}) // 14.142135623730951
pickRandom(array):number / Array	Случайный выбор одного или нескольких значений из одномерного массива. Элементы массива выбираются случайным образом с равномерным или взвешенным распределением	array – одномерный массив	pickRandom([3, 6, 12, 2]) // возвращает одно из значений массива
random([min, max]): number / Array / Matrix	Используя равномерное распределение, возвращает случайное число, большее или равное min и меньшее max	min : number – минимальная граница случайного значения, включая; max : number – максимальная граница случайного значения, исключая	random(30, 40) // возвращает случайное число от 30 до 40
randomInt([min, max])	Используя равномерное распределение, возвращает случайное целое число, большее или равное min и меньшее max	min : number – минимальная граница случайного значения, включая; max : number – максимальная граница случайного значения, исключая	randomInt(30, 40) // возвращает случайное целое число от 30 до 40
Арифметические функции	см. Приложение А настоящего документа		
Статистические функции	см. Приложение Б настоящего документа		
Тригонометрические функции	см. Приложение В настоящего документа		

4.4.3. Доступные методы в объекте FormsyMoment

FormsyMoment – объект, который связан с какой-то конкретной датой, временем и таймзоной. Объект FormsyMoment имеет несколько полезных методов для манипуляции и форматирования даты и времени.

Можно использовать большинство методов из библиотеки moment.js. Также есть методы, реализованные непосредственно на Платформе.

В таблице 6 представлены основные методы в объекте FormsyMoment, которые могут быть полезны при работе с Платформой.

Таблица 6 – Основные методы в объекте FormcyMoment

Функция	Название	Параметры	Пример использования
format(string): string	Форматирование даты в удобочитаемый вид с учетом текущей локали для интернационализации	format – формат данных, доступный в Angular DatePipe. Это может быть настраиваемый формат или один из готовых пресетов. Готовые пресеты форматов учитывают интернационализацию с учетом текущей локали. По умолчанию «short», если не указано	date('2022-01-01T12:30Z').format('short') // '01.01.2022, 12:30'
toSrc('datetime' 'date'): string	Форматирование даты в формат конкретного типа данных	format – тип данных, который может использоваться для дат в моделях данных, т.е. в хранилище данных и для обмена между сервером и клиентской части Платформы. Поддерживаемые типы: «date», «datetime». По умолчанию «datetime», если не указано или указано неверно	date('2022-01-01T12:30Z').toSrc('date') // '2022-01-01'
add(number, string): FormcyMoment	Добавляет время к времени объекта FormcyMoment	number – количество, которое необходимо добавить; string – единица времени, которое необходимо добавить	add(7, 'days') // добавит 7 дней
subtract(number, string): FormcyMoment	Вычитает время из времени объекта FormcyMoment	number – количество, которое необходимо вычесть; string – единица времени, которое необходимо вычесть	subtract(7, 'days') // вычитет 7 дней
startOf(string): FormcyMoment	Устанавливает время объекта FormcyMoment на начало единицы времени	string – единица времени в строковом формате	startOf('year')// устанавливает дату на 1 января, 00:00 этого года
endOf(string): FormcyMoment	Устанавливает время объекта FormcyMoment на	string – единица времени в строковом формате	endOf('year')// устанавливает дату на 31 декабря,

Функция	Название	Параметры	Пример использования
	конец единицы времени		23:59:59.999 этого года
get(string)	Возвращает значение указанной единицы времени в объекте FormcyMoment	string – единица времени в строковом формате	date('2022-01-01T12:30Z').get('year')// возвращает число 2022
set(string, int): FormcyMoment	Устанавливает указанное значение единицы времени объекту FormcyMoment	string – единица времени в строковом формате; int – значение, которое нужно установить единице времени	set('year', 2013)// устанавливает в объекте FormcyMoment 2013 год
millisecond()	Задаёт или получает миллисекунды времени объекта FormcyMoment	Если передается параметр number , то устанавливается указанное число миллисекунд. Принимаются числа от 0 до 999. Если диапазон превышен, он будет увеличиваться до секунд. Если параметр не задан, то возвращает миллисекунды времени объекта FormcyMoment	millisecond(123)// устанавливает в объекте FormcyMoment значение миллисекунд = 123 date([2010, 1, 14, 15, 25, 50, 33]).millisecond()// возвращает число 33
second()	Задаёт или получает секунды времени объекта FormcyMoment	Если передается параметр number , то устанавливается указанное число секунд. Принимаются числа от 0 до 59. Если диапазон превышен, он будет увеличиваться до минут. Если параметр не задан, то возвращает секунды времени объекта FormcyMoment	second(42)// устанавливает в объекте FormcyMoment значение секунд = 42 date([2010, 1, 14, 15, 25, 50, 33]).second()// возвращает число 50
minute()	Задаёт или получает минуты времени объекта FormcyMoment	Если передается параметр number , то устанавливается указанное число минут. Принимаются числа от 0 до 59. Если диапазон превышен, он будет увеличиваться до часов. Если параметр не задан, то возвращает минуты	minute(42)// устанавливает в объекте FormcyMoment значение минут = 42 date([2010, 1, 14, 15, 25, 50, 33]).second()// возвращает число 25

Функция	Название	Параметры	Пример использования
		времени объекта FormcyMoment	
hour()	Задаёт или получает часы времени объекта FormcyMoment	<p>Если передается параметр number, то устанавливается указанное число часов. Принимаются числа от 0 до 23.</p> <p>Если диапазон превышен, он будет увеличиваться до дней.</p> <p>Если параметр не задан, то возвращает часы времени объекта FormcyMoment</p>	hour(2)// устанавливает в объекте FormcyMoment значение часа =2 date([2010, 1, 14, 15, 25, 50, 33]).second()// возвращает число 15
date()	Задаёт или получает день месяца	<p>Если передается параметр number, то в день месяца устанавливается указанное число. Принимаются числа от 1 до 31.</p> <p>Если диапазон превышен, он будет увеличиваться до месяцев.</p> <p>Если параметр не задан, то возвращает день месяца времени объекта FormcyMoment</p>	date(2)// устанавливает в объекте FormcyMoment значение числа месяца = 2 date([2010, 1, 14, 15, 25, 50, 33]).second()// возвращает число 14
dayOfWeek()	Получает или задает день недели времени объекта FormcyMoment по стандарту ISO, где 1 – понедельник, а 7 – воскресенье	<p>Если передается параметр number, то устанавливается соответствующий день недели. Принимаются числа от 1 до 7.</p> <p>Если параметр не задан, то возвращает день недели времени объекта FormcyMoment</p>	dayOfWeek(1) // устанавливает в объекте FormcyMoment значение дня недели = Понедельник. При этом дата автоматически устанавливается в соответствии с заданным днем недели (устанавливается дата этого дня недели на неделе времени объекта FormcyMoment). date([2010, 1, 14, 15, 25, 50, 33]).dayOfWeek()// возвращает число 7
dayOfYear()	Получает или задает день	Если передается параметр number , то	dayOfYear(13)// устанавливает

Функция	Название	Параметры	Пример использования
	года времени объекта FormcyMoment	устанавливается соответствующий день года. Принимаются числа от 1 до 366. Если параметр не задан, то возвращает день года времени объекта FormcyMoment	в объекте FormcyMoment дату 13 января date([2010, 1, 14, 15, 25, 50, 33]).dayOfYear() // возвращает число 45
weekOfYear()	Получает или задает неделю года времени объекта FormcyMoment по стандарту ISO	Если передается параметр number , то устанавливается соответствующая неделя года. Если параметр не задан, то возвращает неделю года времени объекта FormcyMoment	weekOfYear(1) // устанавливает в объекте FormcyMoment последнюю дату 1й недели года (7 января) date([2010, 1, 14, 15, 25, 50, 33]).weekOfYear() // возвращает число 6
month()	Получает или задает месяц времени объекта FormcyMoment	Если передается параметр number , то устанавливается соответствующий месяц. Принимаются числа от 0 до 11 (0 – январь, 11 – декабрь). Если параметр не задан, то возвращает месяц времени объекта FormcyMoment	month(10) // устанавливает в объекте FormcyMoment месяц ноябрь date([2010, 1, 14, 15, 25, 50, 33]).month() // возвращает число 1
quarter()	Получает или задает квартал времени объекта FormcyMoment	Если передается параметр number , то устанавливается соответствующий квартал. Принимаются числа от 1 до 4. Если параметр не задан, то возвращает квартал времени объекта FormcyMoment	quarter(2) // устанавливает в объекте FormcyMoment первый месяц 2го квартала (апрель) date([2010, 1, 14, 15, 25, 50, 33]).quarter() // возвращает число 1
year()	Получает или задает год времени объекта FormcyMoment	Если передается параметр number , то устанавливается соответствующий год. Принимаются числа от -270000 до 270000. Если параметр не задан, то возвращает год времени объекта FormcyMoment	year(2023) устанавливает в объекте FormcyMoment год = 2023 date([2010, 1, 14, 15, 25, 50, 33]).year() // возвращает число 2010
weeksInYear(): number	Получает количество недель года времени		date([2010, 1, 14, 15, 25, 50, 33]).weeksInYear() //

Функция	Название	Параметры	Пример использования
	объекта FormсуMoment по стандарту ISO		возвращает число 52
weekday()	Получает или задает день недели времени объекта FormсуMoment в соответствии с локалью	Если передается параметр number , то устанавливается соответствующий день недели в соответствии с локалью. Если параметр не задан, то возвращает день недели времени объекта FormсуMoment в соответствии с локалью	date([2022, 9, 7]).weekday()// возвращает число 5 date([2022, 9, 7]).weekday(-7) // устанавливает дату на 25 сентября 2022
day()	Получает или задает день недели времени объекта FormсуMoment. Этот метод можно использовать для установки дня недели, где воскресенье – 0, а суббота – 6.	Если передается параметр number , то устанавливается соответствующий день недели. Если параметр не задан, то возвращает день недели времени объекта FormсуMoment	date([2022, 9, 7]).day()// возвращает число 5 date([2022, 9, 7]).day(-7) // устанавливает дату на 25 сентября 2022
week()	Получает или задает неделю года времени объекта FormсуMoment в соответствии с локалью	Если передается параметр number , то устанавливается соответствующая неделя года в соответствии с локалью. Если параметр не задан, то возвращает неделю года времени объекта FormсуMoment в соответствии с локалью	date([2022, 9, 7]).week()// возвращает число 41 date([2022, 9, 7]).week(1) // устанавливает дату на 31 декабря 2021
toNow(): string	Возвращает время от времени объекта FormсуMoment до текущего момента		date([2007, 0, 29]).toNow() // возвращает строку 'in 16 years' date([2007, 0, 29]).toNow(true) // возвращает строку '16 years'
to()	Возвращает время от времени объекта FormсуMoment до указанного	Если параметр не задан, то используется текущая дата	date().to([2022, 6, 29]) // возвращает строку '2 month ago' date().to([2022, 11, 29]) // возвращает строку 'in 3 month' date([2007, 0, 29]).to()

Функция	Название	Параметры	Пример использования
			// возвращает строку 'in 16 years'
locale()	Получает или задает локаль (язык, на котором будут выводиться ответы методов)	По умолчанию Moment.js выдает ответы на английском языке. Для смены локали необходимо указать в параметре двухбуквенный код языка	date().locale() //возвращает строку 'en' date().locale('ru')// возвращает строку 'ru' и меняет локаль date().to([2022, 11, 29]) // возвращает строку 'через 3 месяца'
isLeapYear()	Возвращает true, если этот год високосный, false – если нет		date([2000]).isLeapYear() // true
localeData()	Возвращает словарь языка (короткие названия месяцев, дней недель и т.п.)		

4.4.4. Проверка работоспособности формул

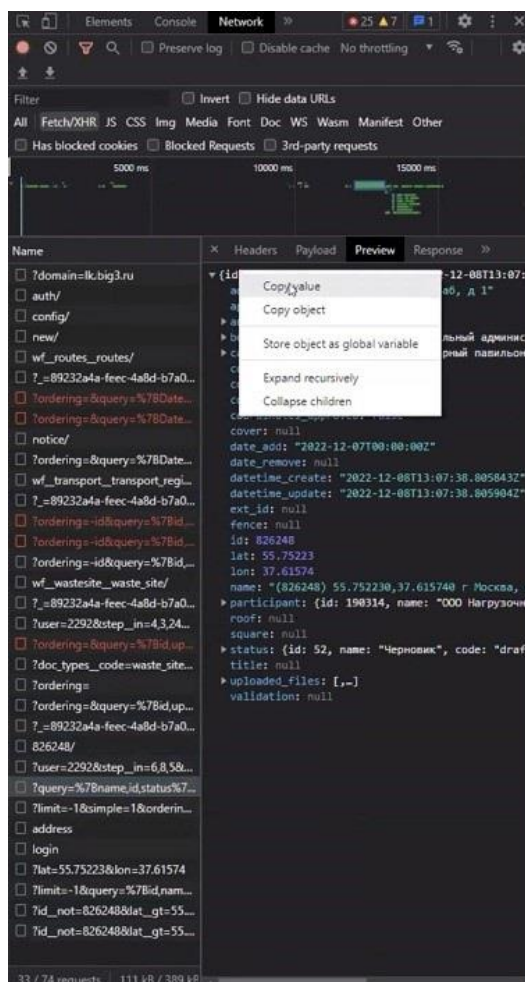
Для удобной проверки работоспособности формул реализована песочница формул. Песочница формул доступна только для пользователя с ролью «Администратор».

Песочница формул («Профиль → Dev: Инструменты») позволяет скопировать готовый результат из формы в контекст data и протестировать работу формул.

Для проверки работоспособности формул необходимо выполнить следующие действия:

1. Откройте любую детальную форму.
2. Откройте инструменты разработчика в браузере, перейдите на вкладку «Network».

Найдите запрос и скопируйте значение (рис. 12).



Инструменты разработчика в браузере. Вкладка «Network»

Рис. 12

2. В песочнице формул в поле «Данные» укажите контекст data (рис. 13)



Поле «Данные» песочницы формул

Рис. 13

3. Вставьте скопированное значение из формы.

Теперь можно проверять работу формул (рис. 14).



Пример проверки работы формул

Рис. 14

4.5. Работа с бизнес-процессами со статусами

На Платформе предусмотрена возможность настройки двух типов бизнес-процессов.

Первый тип бизнес-процессов – это «машина состояний» – бизнес-процесс, основанный на смене состояний объектов учета. Для каждой группы сущностей в информационной системе можно настроить перечень доступных состояний (статусов) и граф переходов между ними. После

этого пользователям информационной системы в интерфейсе становятся доступны кнопки переходов между статусами (при наличии соответствующих разрешений, разрешения настраиваются в разделе «Права на переходы»). С помощью этих кнопок пользователи меняют состояние объекта учета.

Администратор Платформы может задавать в административной панели правила, согласно которым страница отображения объекта учета будет меняться в зависимости от его состояния. В зависимости от состояния объекта разные роли пользователей могут иметь разные права на действия с объектом.

Второй тип бизнес-процессов – это система настраиваемых шаблонов событий и реакций на эти события. В админ-панели можно задать условия, при которых в базе данных будет возникать запись о событии (например, событие должно возникать при обновлении записи в реестре МНО с категорией «Контейнерная площадка»). Запись о событии будет содержать также данные объекта, с которым данное событие произошло. Далее в админ-панели можно задать перечень реакций на каждое такое событие, среди реакций может быть отправка письма по e-mail, создание новой записи в каком-либо реестре, операция, переход статуса или http-запрос. Таким образом можно задавать цепочки событий и реакций информационной системы на эти события, создавая дерево автоматизированных процессов.

Объекты в Платформе могут иметь разные статусы. Пользователи могут производить операции над объектами, при которых объект будет переходить из одного статуса в другой. При этом для различных статусов объекта можно настраивать доступность данного объекта для пользователей, перечень действий, которые можно производить с объектом, и т.п. (см. п. 4.3.1 настоящего документа)

Статусы неразрывно связаны с ролевой моделью и правами и разрешениями на Платформе (см. п. 4.3.2 настоящего документа)



4.5.1. Создание бизнес-процесса со статусами

Для создания бизнес-процесса со статусами для объектов типа представления необходимо выполнить следующие действия:

1. Перейдите в раздел конструктора «Документооборот → 1. Типы представлений».
2. Откройте тип представления, для которого необходимо настроить бизнес-процесс со статусами.
3. В подразделе формы «Параметры» в поле «Доступные статусы» добавьте в правый блок все статусы, которые должны быть у объектов данного типа представления. В блоке слева

отображаются все созданные в информационной системе статусы (в том числе для других типов представлений).

Если вы не нашли нужный статус, нажмите кнопку «+» напротив поля и заполните форму создания нового статуса (рис. 15):

Наименование:	<input type="text" value="Согласован"/>
Категория:	<input type="text" value="Общие"/>
Код:	<input type="text" value="soglasovan"/>
Цвет:	<input type="text" value="neutral"/> 
Иконка:	<input type="text" value="fas fa-asterisk"/> 
<input checked="" type="checkbox"/> Активен	
Порядок:	<input type="text" value="200"/>
<input type="checkbox"/> Считать объекты с этим статусом удаленными	
<input type="button" value="СОХРАНИТЬ"/>	

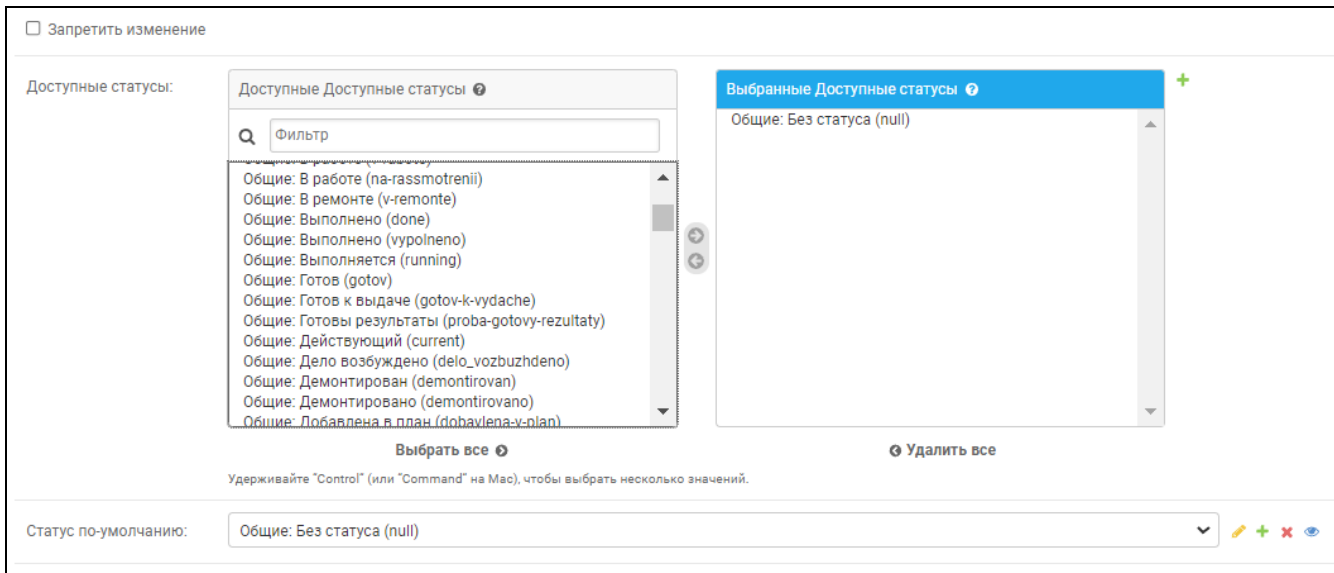
Пример заполнения формы создания нового статуса

Рис. 15

- в поле «Наименование» укажите удобное для пользователя название статуса;
- в поле «Категория» укажите удобное для аналитика название категории. Если такой категории еще нет, то она будет создана;
- в поле «Код» укажите уникальный код статуса или оставьте код, сформированный автоматически;
- в поле «Цвет» выберите цвет из доступных;
- в поле «Иконка» выберите иконку статуса из доступных;
- в поле «Активен» поставьте отметку, если необходимо, чтобы статус отображался в интерфейсе;
- в поле «Порядок» при необходимости укажите числовое значение, которое будет использоваться при сортировке доступных статусов. Но обратите внимание, что это значение не влияет напрямую на сортировку статусов во всех формах, а может использоваться, как параметр при задании условий отображения в конкретной форме;
- в поле «Считать объекты с этим статусом удаленными» можно поставить отметку, если требуется настройка скрытия элементов с таким статусом, не прибегая к коду. Например, статусы «Завершил действие» и «Удален» могут иметь такую отметку, а

аналитик может настроить дашборд таким образом, чтобы не учитывать объекты с такими статусами.

4. В форме типа представления в поле «Статус по умолчанию» можно также выбрать статус, в который будут переводиться все объекты типа представления при создании (рис. 16).



Выбор статуса в форме типа представления

Рис. 16

5. Создайте переходы статусов, т.е. действия, после запуска которых будет осуществляться смена статуса. Для этого в подразделе «Переходы» формы типа представления нажмите «Добавить еще один Переходы статусов» и заполните данные (рис. 17):

ПЕРЕХОДЫ

Переходы статусов: #1

Наименование:

Код:

Тип стиля:

Из статуса:

В статус:

Форма перехода:

Операция перед переходом:

Операция после перехода:

Создание на основании:

Заполнение атрибутов при переходе:

Формат - атрибут: значение, каждый на новой строке.
Для текстовых или числовых атрибутов требуется обрамление кавычками.
Для доступа к типу представления следует использовать "entity"
Примеры: attr_name1: "test"
attr_name2: True
attr_name3: entity.participant.name

Создание переходов статусов

Рис. 17

- в поле «Наименование» укажите наименование кнопки, которая будет отображаться в интерфейсе пользователю, и при нажатии на которую будет происходить смена статуса;
- в поле «Код» укажите уникальный код перехода или оставьте код, сформированный автоматически;
- в поле «Тип стиля» выберите цвет кнопки перехода статуса, которая будет отображаться в интерфейсе пользователю;

- в поле «Из статуса» выберите статус, из которого будет осуществляться переход в другой статус;
- в поле «В статус» выберите статус, в который будет осуществляться переход из другого статуса;
- в поле «Форма перехода» при необходимости выберите форму, которая будет открываться при переводе статуса (в таком случае нельзя будет изменить статус, пока пользователь не пройдет форму перехода);
- в поле «Операция перед переходом» при необходимости выберите операцию, которая будет выполняться при переходе, но до изменения статуса. Например, можно выбрать операцию валидации каких-либо данных;
- в поле «Операция после перехода» при необходимости выберите операцию, которая будет выполняться при переходе, но после изменения статуса. Например, можно выбрать операцию обновления каких-либо данных;
- в поле «Создание на основании» при необходимости выберите операцию, которая на основании одного объекта создаст другой. Например, при переходе объект можно удалить, но создать такой же объект в статусе «Черновик».

Повторите действия, чтобы создать столько переходов, сколько необходимо, и сохраните изменения.

б. Задайте права на переходы статусов. Для этого:

- перейдите в раздел конструктора «Документооборот → б. Права на переходы статусов»;
- в правом верхнем углу нажмите кнопку «Добавить право на переходы статусов»;
- в форме выберите созданный ранее переход, а также роль, которой необходимо разрешить перевод статуса. Дополнительно можно выбрать одно или несколько условий, при которых данной роли будет доступен выбранный переход статуса (рис. 18).

Добавить Право на переходы статусов

Переход: Заголовок 102: Общине: Черновик (d...)

Роль: admin

Условия: entity.name == 'Не определен'

Удерживайте "Control" (или "Command" на Mac), чтобы выбрать несколько значений.

Сохранить и добавить другой объект Сохранить и продолжить редактирование СОХРАНИТЬ

Задание прав на переходы статусов

Рис. 18

Если необходимо разрешить перевод статуса для нескольких ролей, необходимо создать новые права на каждую роль;

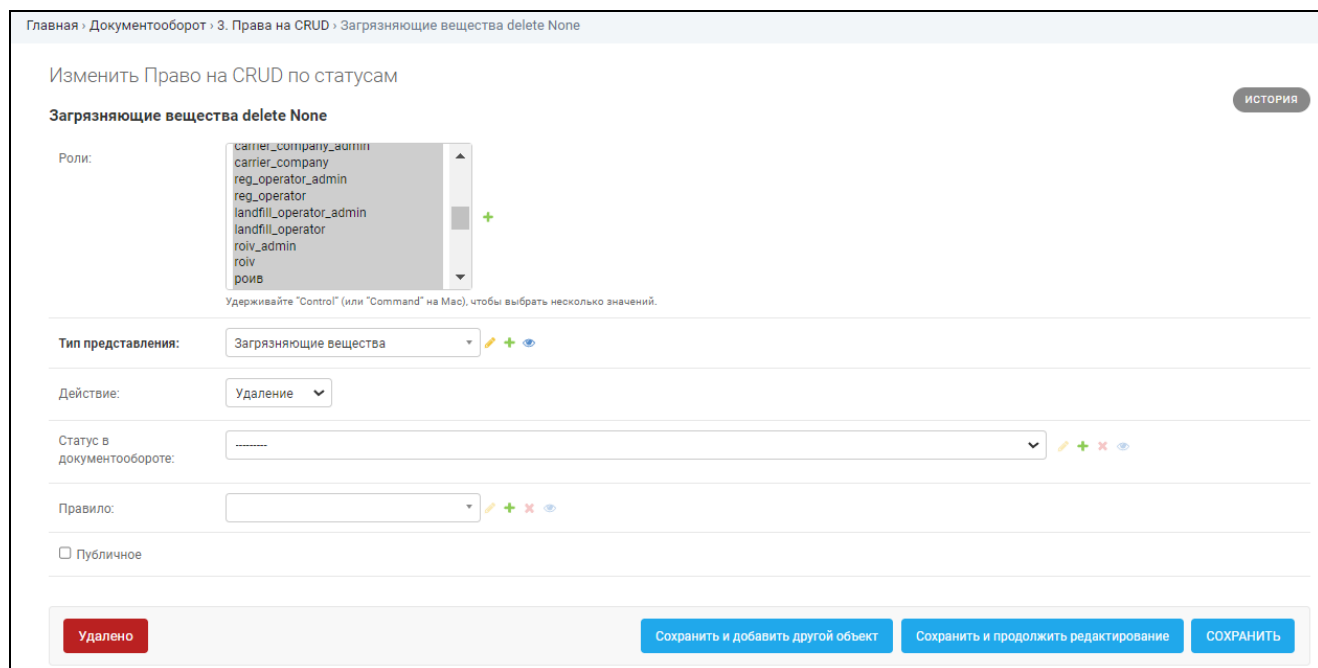
- сохраните изменения.

4.5.2. Настройка прав доступа

4.5.2.1. Создание прав на документы

Для создания прав на документы необходимо выполнить следующие действия:

1. Перейдите в раздел конструктора «Документооборот → 3. Права на CRUD».
2. Нажмите кнопку «Добавить правило» и заполните форму (рис. 19):

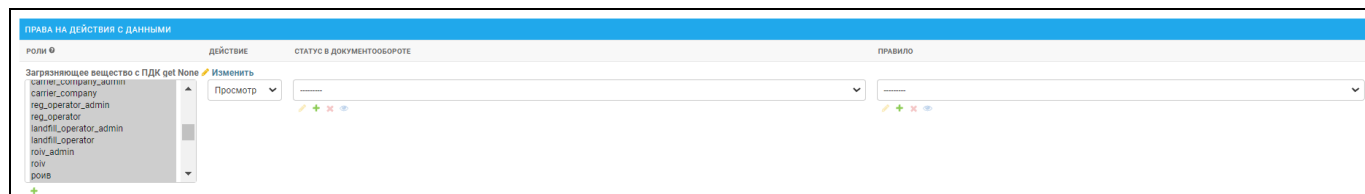


Создание прав на документы

Рис. 19

- в поле «Роли» выберите одну или несколько ролей, для которых необходимо создать право. Если создается публичное право, то роль можно не выбирать. Подробнее про публичные сайты см. п. 4.11 настоящего документа;
- в поле «Тип представления» выберите тип представления, для объектов которого создается право;
- в поле «Действие» выберите действие, для которого создается право;
- в поле «Статус в документообороте» выберите статус, при котором должно срабатывать право;
- в поле «Правило» выберите json-правило, которое будет применяться к праву;
- в поле «Публичное» поставьте отметку, если необходимо чтобы действие стало доступно на публичном сайте.

3. Права на документы можно также создать и редактировать прямо в форме типа представления в подразделе «Права на действия с данными» (рис. 20).



Создание и редактирование прав на документы в форме типа представления

Рис. 20

4.5.2.2. Создание прав на операции

Для создания права на операцию необходимо выполнить следующие действия:

1. Перейдите в раздел конструктора «Документооборот → 8. Права на операции».
2. Нажмите кнопку «Добавить право на операции» и заполните форму (рис. 21):

Главная » Документооборот » 8. Права на операции » RuleOperations object (526)

Изменить Право на операции ИСТОРИЯ

RuleOperations object (526)

Тип представления: ✎ + 👁

Роль: ✎ + 👁

Операция:

Статус в документообороте: ✎ + ✕ 👁

Правило:

Правила отображения для фронта:

Для операций с типом one, detail можно задать произвольную формулу отображения этой операции для фронта

Удалено Сохранить и добавить другой объект Сохранить и продолжить редактирование СОХРАНИТЬ

Создание прав на операции

Рис. 21

- в поле «Тип представления» выберите тип представления, для объектов которого создается право;
- в поле «Роль» выберите роль, для которой необходимо создать право;
- в поле «Операция» выберите операцию, для которой создается право;
- в поле «Статус в документообороте» выберите статус объекта, если необходимо, чтобы право применялось только в определенном статусе;
- в поле «Правило» выберите правило, которое будет применяться к праву, если это необходимо.

4.6. Работа с печатными формами

Печатная форма – документ, который позволяет вывести на печать данные из информационной системы по определенному шаблону.

4.6.1. Создание печатной формы

Для создания печатной формы необходимо выполнить следующие действия:

1. Перейдите в раздел конструктора «Документооборот → 9. Печатные формы». Нажмите кнопку «Добавить печатные формы».

2. Откроется форма создания печатной формы. Заполните поля в форме (рис. 22):

Изменить Печатные формы ИСТОРИЯ

Акт приемки работ (27)

Тип:

Код:

Название:

Файл: На данный момент: [doc_templates/act_28.01.2022_1.docx](#)
Изменить: Файл не выбран

Формат генерируемого файла:

Расширение файла:
json, xml, tex

Подписанты:

Удерживайте "Control" (или "Command" на Mac), чтобы выбрать несколько значений.

ИСПОЛЬЗОВАНИЕ В ТИПАХ ПРЕДСТАВЛЕНИЯ

ТИП ПРЕДСТАВЛЕНИЯ	ШАБЛОН ИМЕНИ ГЕНЕРИРУЕМОГО ФАЙЛА
DocTemplateDocTypeMembership object (14) <input type="text" value="Проверочные мероприятия"/>	<input type="text"/>
DocTemplateDocTypeMembership object (18) <input type="text" value="Акт проверки"/>	<input type="text"/>

Форма создания печатной формы

Рис. 22

- в поле «Код» укажите уникальный код печатной формы, используя буквы латинского алфавита;
- в поле «Название» укажите название печатной формы, которое будет отображаться в интерфейсе пользователю;
- в поле «Файл» выберите шаблон печатной формы. Подробнее про создание файла шаблона см. в разделе Создание шаблонов для печатных форм;

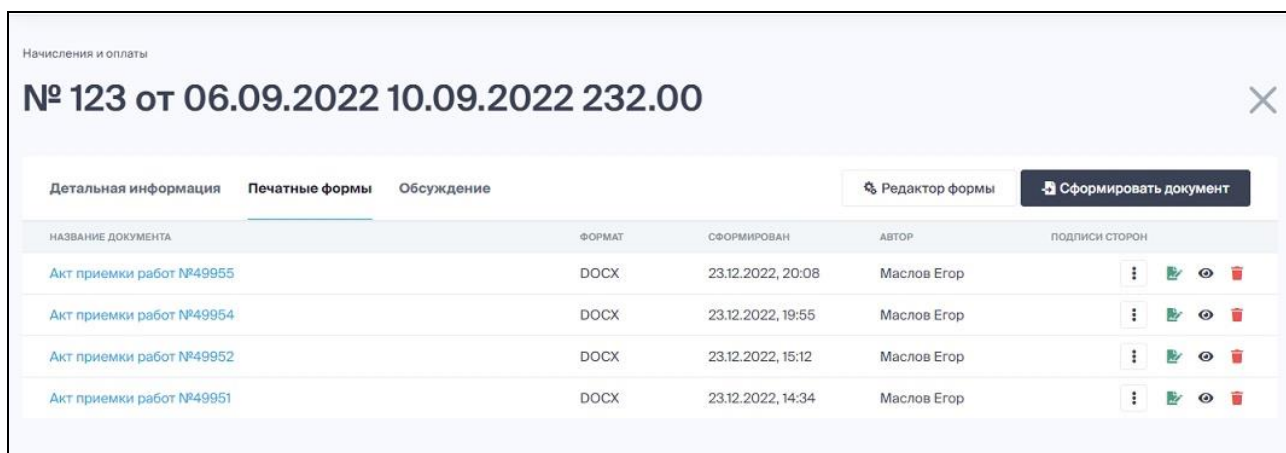
- в поле «Формат генерируемого файла» можно выбрать формат генерируемого файла: DOCX или текст. Для текстовой формы можно задать формат файла – JSON или XML;
- в поле «Подписанты» при необходимости можно отметить стороны подписания, которым будет доступно подписание печатной формы (см. п. 4.6.3 настоящего документа);
- в разделе формы «Использование в типах представления» в столбце «Doctype» выберите тип(ы) представления, для которых будет доступна данная печатная форма.

Чтобы печатная форма отображалась в формах, построенных по выбранным типам представления, перейдите из формы в редактирование типа представления и поставьте отметку «Показывать печатные формы».

В настройках печатной формы (или типа представления) в поле «Шаблон» можно указать шаблон, по которому будет сформировано имя генерируемого файла. Для обращения к полям шаблона используется `template`, для обращения к полям объекта используется `obj`. Например:

новый_файл_{obj.__str__()}_{obj.id}_{template.name}

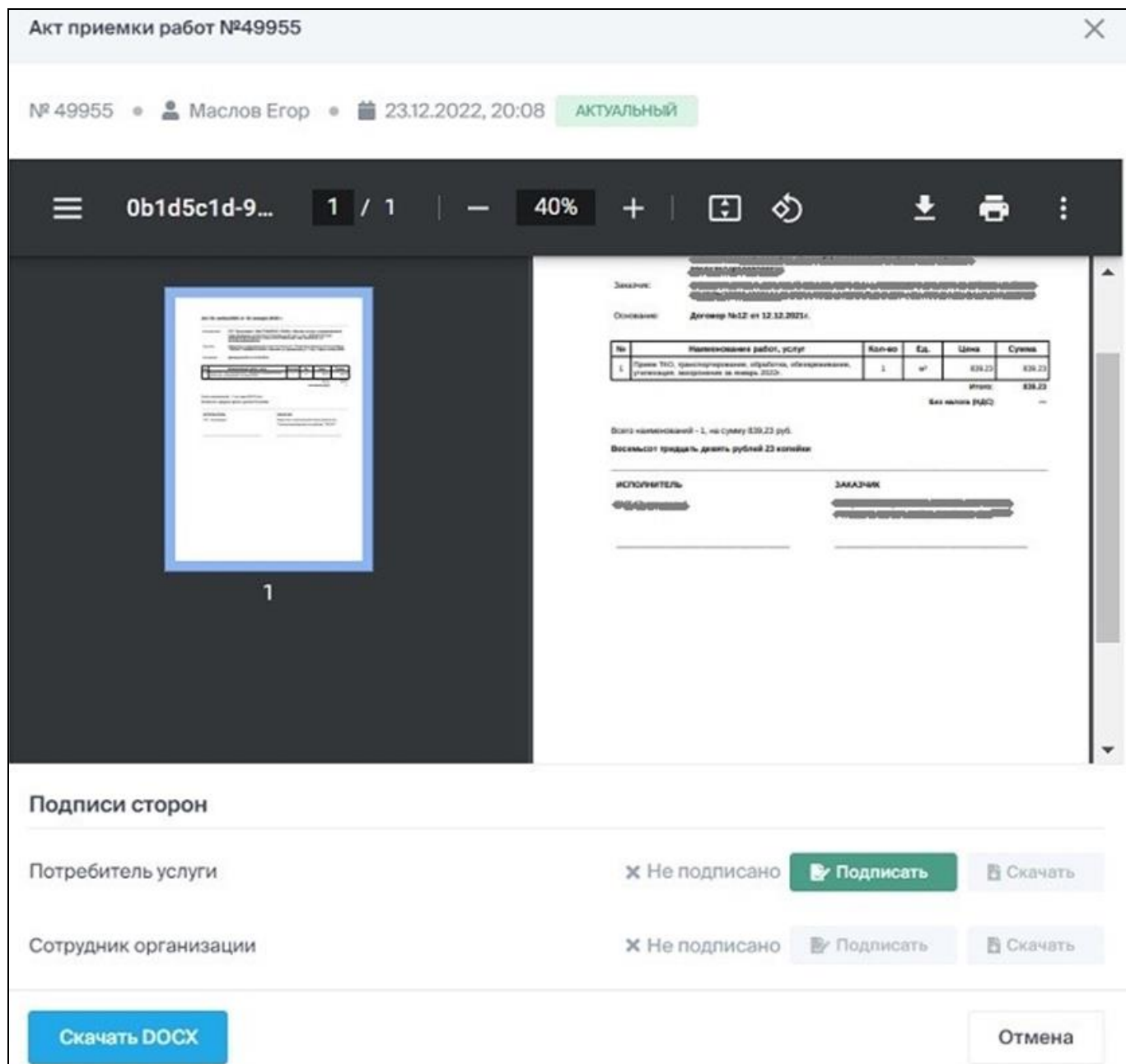
3. В форме элемента появится вкладка «Печатные формы». На вкладке списком будут отображаться распечатанные версии документа (рис. 23).



Отображение распечатанных версий документов

Рис. 23

4. Нажмите кнопку «Сформировать документ», чтобы сформировать новую версию. После формирования откроется окно предпросмотра документа с функциями подписания (рис. 24).



Окно предпросмотра документа с функциями подписания

Рис. 24

Если у пользователя, открывшего документ, недостаточно прав для подписания, кнопка подписания будет неактивна.

Документ можно скачать как отдельно, так и с ЭЦП, которые станут доступны после подписания документа всеми сторонами. Также, чтобы пользователю стала доступна кнопка «Скачать подпись» в форме подписания, необходимо добавить пользователю право на «Чтение» типа представления «Подпись».

4.6.2. Создание шаблонов печатных форм

Для создания печатной формы необходимо создать шаблон печатной формы, содержащий Jinja-разметку.

Для создания шаблона печатной формы необходимо:

1. Взять готовый отчет в формате .DOCX.
2. Текст, который нужно заменить значениями из БД, выделить и к выделению создать комментарий.
3. В комментарии написать запрос к объекту в формате Jinja2 (<https://jinja.palletsprojects.com/en/3.0.x/templates/>)

Пример добавления имени объекта: `{{ obj.name }}`

4. Для добавления элементов списка в таблицу:

- выделить фрагмент готовой таблицы (все столбцы и строки, значения которых должны заполниться из списка);
- создать для этого фрагмента комментарий с текстом типа: `{%tr for item in obj.test_results.all() %}`;
- для каждого элемента первой строки выделенного фрагмента таблицы создать комментарий типа: `{{ item.name }}`.

5. Вставка изображений в таблицу:

Для одиночных изображений: Если у объекта есть поле <имя_поля> типа FileField, то в шаблоне можно вставить данное изображение с тэгом `{{ obj.<имя_поля> | image_filter(40) }}`, где 40 – высота изображения в мм.

Для группы изображений: Возможность вставки изображений в шаблон есть у объектов, которые содержат множественное поле files (например many-to-many на таблицу с объектами, содержащими поле file). В этом случае в шаблон можно добавить таблицу с блоком:

```
{%tr for f in files.all() %}  
  
{{ f.file | image_filter(20) }}
```

6. Для документов, требующих подписания, можно добавить штампы подписи, и они будут вставляться, только если кто-то уже подписал документ.

Для добавления штампа необходимо указать в комментарии в месте отображения подписи функцию `{{ show_sign('oku_signer') }}`, где oku_signer – конкретный тип подписанта.

В штампе будут отображаться данные из сертификата, если есть соответствующая типу подписанта подпись.

7. Дополнительные теги шаблонов:

- `morph_gent` – представление слова в родительном падеже;
- `num_to_text_money` – число в текст в рублях;
- `format_datetime` – форматирование даты
 - формат: `obj.date | format_datetime(format, timezone)`. Например: `obj.date | format_datetime("%d.%m.%YГ.", 180)`,
где `format` – формат даты, не обязательное; `timezone` - таймзона в минутах, не обязательное;
- `format_date` – форматирование даты, только дата | `format_date(timezone)`;
- `format_time` – форматирование даты, только минуты | `format_date(timezone)`.

4.6.3. Настройка подписания печатных форм

У печатных форм при необходимости можно настроить подписание ЭЦП разными сторонами подписания.

Для настройки подписания печатной формы необходимо выполнить следующие действия:

1. Перейдите в раздел конструктора «Документооборот → 9.1 Стороны подписания». Нажмите кнопку «Добавить сторона подписания».
2. Откроется форма создания стороны подписания. Заполните поля в форме:
 - в поле «Название» введите название стороны, которое будет отображаться в интерфейсе пользователю;
 - в поле «Лейбл» введите уникальный код стороны подписания.
3. В созданной печатной форме выберите необходимые стороны подписания.
4. Перейдите в раздел конструктора «Документооборот → 8. Права на операции». Нажмите кнопку «Добавить право на операции».
5. Откроется форма создания права. Заполните поля в форме (рис. 25):

Изменить Право на операции

RuleOperations object (584) ИСТОРИЯ

Тип представления: Подписант конкретного документа

Роль: admin

Операция: Подписать (sign)

Сторона подписания: _____

Статус в документообороте: _____

Правило: все

Правила отображения для фронта: _____

Для операций с типом one, detail можно задать произвольную формулу отображения этой операции для фронта

Удалено Сохранить и добавить другой объект Сохранить и продолжить редактирование СОХРАНИТЬ

Форма создания права

Рис. 25

- в поле «Тип представления» выберите «Подписант конкретного документа». Для создания права на подписание необходимо выбирать только данный тип представления. Создание права на каждый отдельный документ не требуется;
- в поле «Роль» выберите роль, для которой будет доступно подписание документа под выбранной стороной подписания;
- в поле «Операция» выберите операцию «Подписать (sign)»;
- в поле «Сторона подписания» выберите сторону подписания.

При необходимости заполните дополнительные условия доступности операции подписания в полях «Статус в документообороте» и «Правило».

4.7. Работа с аналитическими панелями (дашбордами)

Аналитические панели (дашборды) – это тип формы, в котором основные данные по объектам типа представления наглядно отображаются с помощью различных диаграмм, графиков и др. компонентов. Аналитические панели отображают в конструкторе агрегированную информацию из реестров.

Аналитическая панель является дополнением к форме списка. На Платформе предусмотрена возможность отображение формы списка и аналитической панели как по отдельности, так и вместе.

Аналитическая панель представляет собой страницу, состоящую из следующих частей:

- область фильтрации – на панели доступны те же фильтры, что настроены в форме списка. При этом в расширенных фильтрах добавляются фильтры, которые генерируются на основе виджетов дашборда;
- поля аналитики – в них отображаются агрегированные числовые показатели из реестра (например, сумма, максимальное число, минимальное, количество и т.п.);
- виджеты дашборда – отображаются виджеты по разрезам аналитики.

4.7.1. Создание аналитической панели (дашборда)

Создание аналитических панелей доступно в разделе «Формы» админ-панели.

Для создания аналитической панели необходимо выполнить описанные ниже действия.

1. Перейти в редактирование формы списка.
2. В разделе «Аналитическая панель» создать новую форму, нажав на знак «+». В форме заполнить следующие поля:
 - в поле «Наименование» указать наименование дашборда;
 - в поле «Тип формы» выбрать «Аналитическая панель»;
 - в поле «Тип представления» необходимо выбрать соответствующий форме списка тип представления.
3. После создания новой формы необходимо в разделе «Аналитическая панель» выбрать созданную форму и поставить отметку «Показывать дашборд».
4. Далее необходимо настроить форму дашборда. Для этого в режиме редактирования формы реализованы три папки: «Поля аналитики», «Поля срезы данных», «Компоненты».

В папке «Поля аналитики» создаются поля, которые отображаются на странице дашборда в части полей аналитики.

В папке «Поля срезы данных» создаются свойства, в разрезе которых будут отображаться данные дашборда.

В папке «Компоненты» создаются компоненты, которые будут отображаться на странице дашборда в части виджетов. При создании компонента необходимо выбрать срез, поэтому компонент не может быть создан без среза.

В Платформе доступны следующие компоненты:

- карта регионов;
- динамика;
- круговая диаграмма;
- линейная диаграмма;
- рейтинг.

Коды регионов для работы карты статистики должны быть в ISO формате типа RU-CR

В компоненте «Динамика» на дашборде есть два таймлайна (шкалы времени):

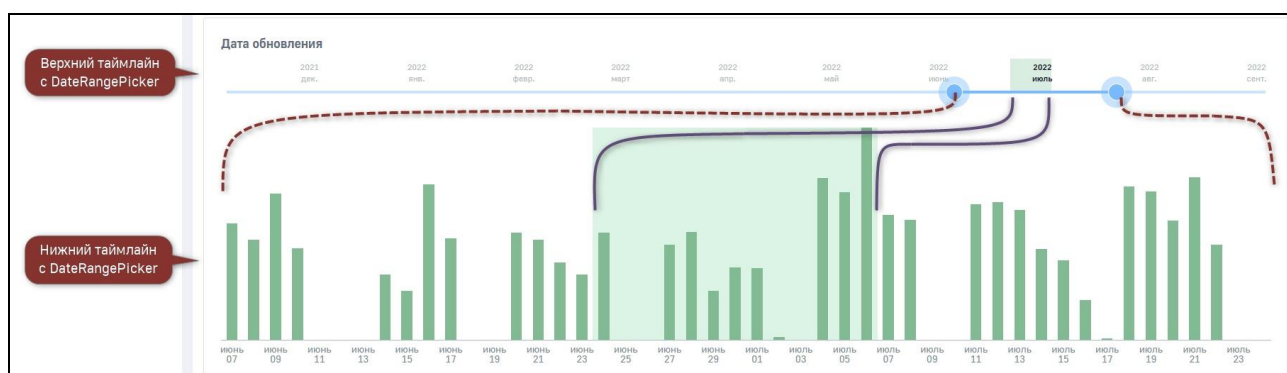
Верхний таймлайн показывает весь диапазон, за который есть данные с учетом фильтров. На этом таймлайне можно выбирать диапазон для отображения на нижнем таймлайне. Верхний таймлайн загружает значения с лимитом 3000 фреймов.

Нижний таймлайн с графиком показывает более детально диапазон, выбранный на верхнем таймлайне.

Компонент «Динамика», как и другие компоненты дашборда, учитывает все фильтры, кроме атрибута фильтра, по которому он сам фильтрует значения. Каждый компонент дашборда может добавлять значение для одного или нескольких атрибутов фильтра, но при загрузке данных для себя учитывает только «чужие» атрибуты фильтра.

Фильтры учитываются как в верхнем, так и в нижнем таймлайне. Если «чужие» фильтры поменялись, то верхний и нижний таймлайн загружают обновленные данные и перерисовываются.

При перерисовке верхнего таймлайна учитывается выбранный ранее диапазон для нижнего таймлайна, т.е. выбранный ранее диапазон не меняется, а новые границы верхнего таймлайна должны включать этот диапазон. Если выбранный диапазон не входит в новые границы верхнего таймлайна, то границы автоматически расширятся (рис. 26).



Аналитическая панель (дашборд)

Рис. 26

4.8. Работа с витриной данных

При необходимости отображения в одной форме данных из нескольких таблиц можно создавать витрины данных.

Витрина данных – это таблица только для чтения, которая включает в себя данные из нескольких таблиц и с которой можно работать как с обычной таблицей БД. Можно создать на основе витрины тип представления, построить дашборд и т.п.

Данные в витринах не обновляются в режиме реального времени. Витрины необходимо использовать тогда, когда не требуется 100% точность данных.

4.8.1. Создание витрины данных

Возможности Платформы позволяют пользователю создавать витрины данных одним из двух способов:

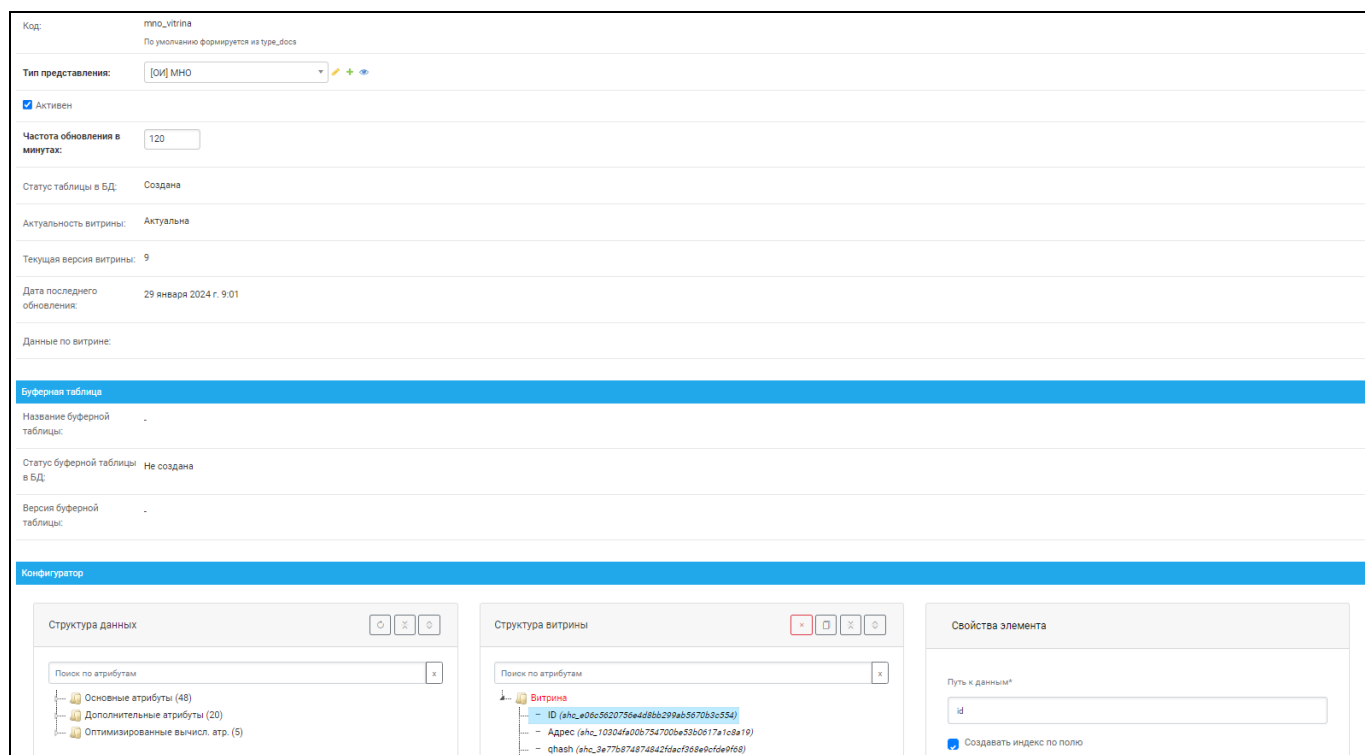
- упрощенный способ – позволяет создавать витрину в пользовательском интерактивном режиме, выбирая нужные поля в конструкторе;
- продвинутый способ – позволяет создать витрину на основе произвольного SQL-запроса.

4.8.1.1. Создание витрины данных упрощенным способом

Для создания витрины данных упрощенным способом необходимо выполнить следующие действия:

1. Перейдите в раздел конструктора «Конструктор витрин → 1. Витрины данных». Нажмите кнопку «Добавить витрина данных».
2. Откроется форма создания витрины. Заполните поля в форме:
 - в поле «Наименование» укажите наименование витрины, которое будет использоваться в конструкторе;
 - в поле «Код» укажите уникальный код витрины, используя буквы латинского алфавита. Можно оставить поле пустым, тогда код сформируется автоматически;
 - в поле «Тип представления» выберите тип представления, на основе данных которого необходимо создать витрину;
 - в поле «Активен» поставьте отметку, чтобы данные в витрине обновлялись;
 - в поле «Частота обновления в минутах» укажите с какой периодичностью необходимо обновлять данные в витрине.

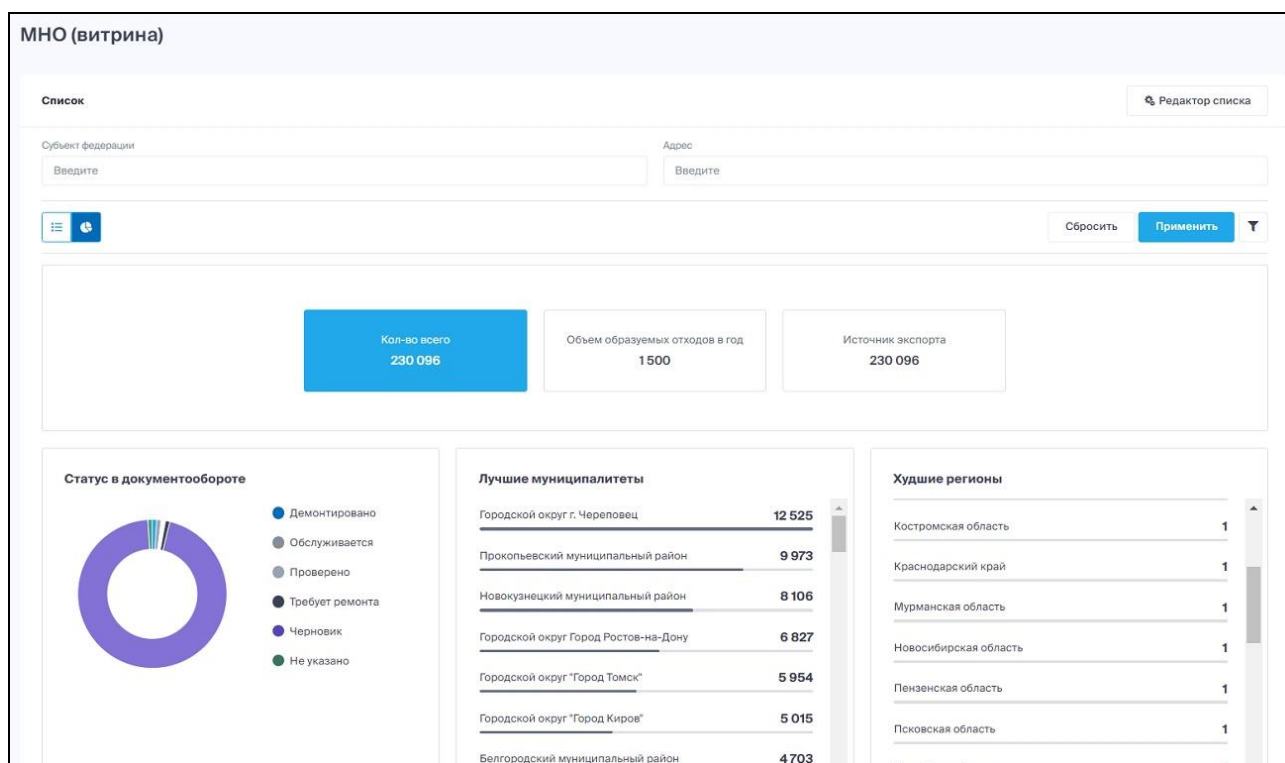
3. Сохраните форму. После сохранения станет доступным Конфигуратор витрины (рис. 27). В нем необходимо добавить поля на витрину как при создании обычной формы.



Конфигуратор витрины в форме создания витрины

Рис. 27

4. После сохранения конфигурации витрины нажмите кнопку «Обновить витрину». После обновления создастся таблица с выбранными полями, с которой можно работать как с обычной таблицей БД. Витрина также будет обновляться с указанной в форме редактирования витрины периодичностью. Пример дашборда, построенного на основе витрины данных, представлен на рис. 28.



Пример дашборда, построенного на основе витрины данных

Рис. 28

4.8.1.2. Создание витрины данных продвинутым способом

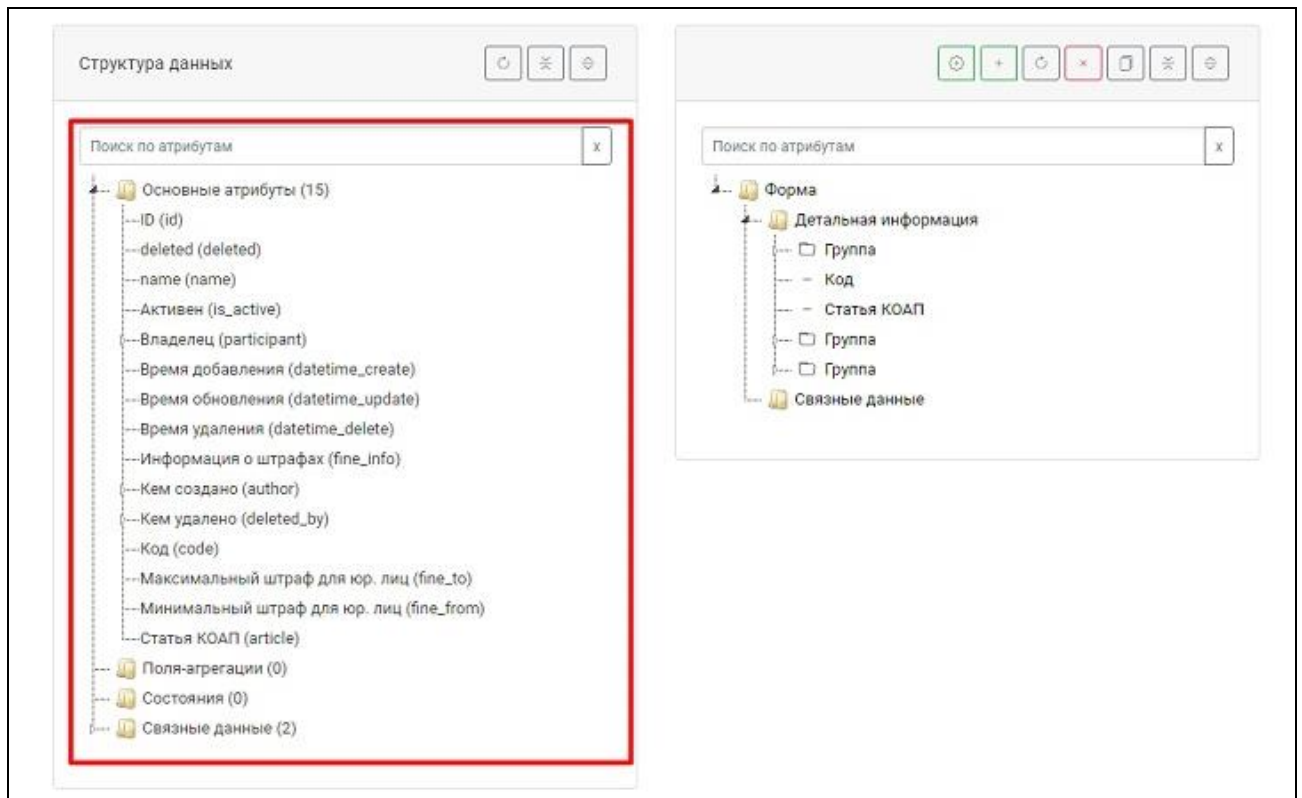
Для создания витрины данных продвинутым способом необходимо выполнить следующие действия:

1. Перейдите в раздел конструктора «Документооборот → 1. Типы представлений». Нажмите кнопку «Добавить тип представления».
2. В форме создания типа представления в поле «Тип таблицы» выберите «Представление».
3. В поле «SQL запрос» напишите запрос для создания витрины данных:
 - имена таблиц, из которых необходимо получить данные, можно узнать в коде модели или в конструкторе в формах типов представления.

Если в модели в классе Meta указана переменная `db_table`, то в запросе необходимо использовать это имя в качестве имени таблицы в БД.

Если в модели не задано специальное имя для таблицы, то по умолчанию оно формируется как `<app label>_<Имя класса модели>`;

- названия полей таблиц БД можно узнать из форм, построенных на типах представления, из которых необходимо получить данные (рис. 29).



Пример названия полей

Рис. 29

Пример SQL-запроса представлен на рис. 30.

Главная > Документооборот > 1. Типы представлений > Представление мно

Изменить Тип представления

Представление мно

Основные данные

Наименование: Код:

Заголовок: Наименование (мн. число):

Тип таблицы:

Sql запрос:

```
1 SELECT waste_site.id,  
2 waste_site.lon,  
3 waste_site.lat,  
4 waste_site.borough_id,  
5 waste_site.address  
6 FROM waste_site;
```

Параметры

Активен

Показывать печатные формы

Разрешить обсуждение

Хранить значения в отдельной таблице:

Для изменения значений воспользуйтесь командой в списке

Пример SQL-запроса представления

Рис. 30

4. После сохранения будет создана таблица, с которой можно работать, как с обычной таблицей БД. Также по умолчанию будут созданы две формы (рис. 31).

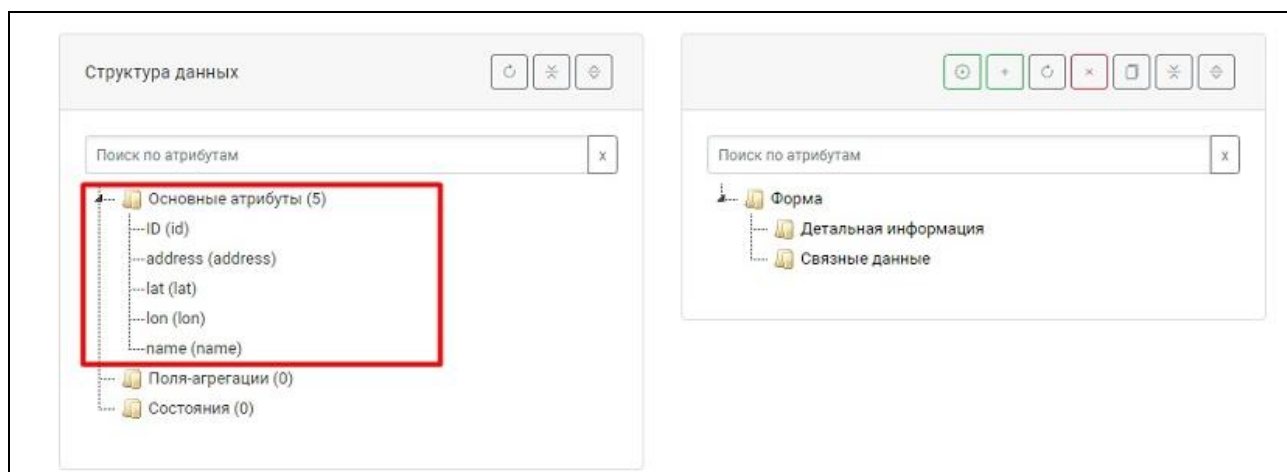
ИМЯ Ф	КОД Ф	АКТИВЕН	ИСПОЛЬЗОВАТЬ ПО УМОЛЧАНИЮ	ТИП ФОРМЫ	ДЕТАЛЬНАЯ ФОРМА	УДАЛИТЬ?
Форма списка "Представление мно" для Представление мно / Изменить Представление МНО	ws_view_list	Да	<input type="checkbox"/>	Форма списка	5965 Q Форма элемента "Представление мно" для Представление мно	<input type="checkbox"/>
Форма элемента "Представление мно" для Представление мно / Изменить Представление МНО	ws_view_object	Да	<input type="checkbox"/>	Форма элемента	Q	<input type="checkbox"/>

+ Добавить еще одну форму

Формы при создании витрин данных продвинутым способом

Рис. 31

В формах в папке «Основные атрибуты» будут доступны для добавления поля, указанные в SQL-запросе при создании типа представления (рис. 32).



Доступные для добавления поля, указанные в SQL-запросе при создании типа представления

Рис. 32

4.9. Работа с аналитическими функциями

Функции аналитики (поля-аналитики, поля-агрегации) – это агрегированные числовые показатели объектов.

Состояние – это значение атрибута объекта в определенный момент времени, хранимое во внешней таблице (таблице состояний), связанной с данным объектом.

4.9.1. Добавление агрегации на форму элемента/списка

Поля-агрегации – это агрегированные значения свойств связанной с объектом таблицы, например, суммарное количество книг автора.

Поля-агрегации добавляются в форме элемента или списка.

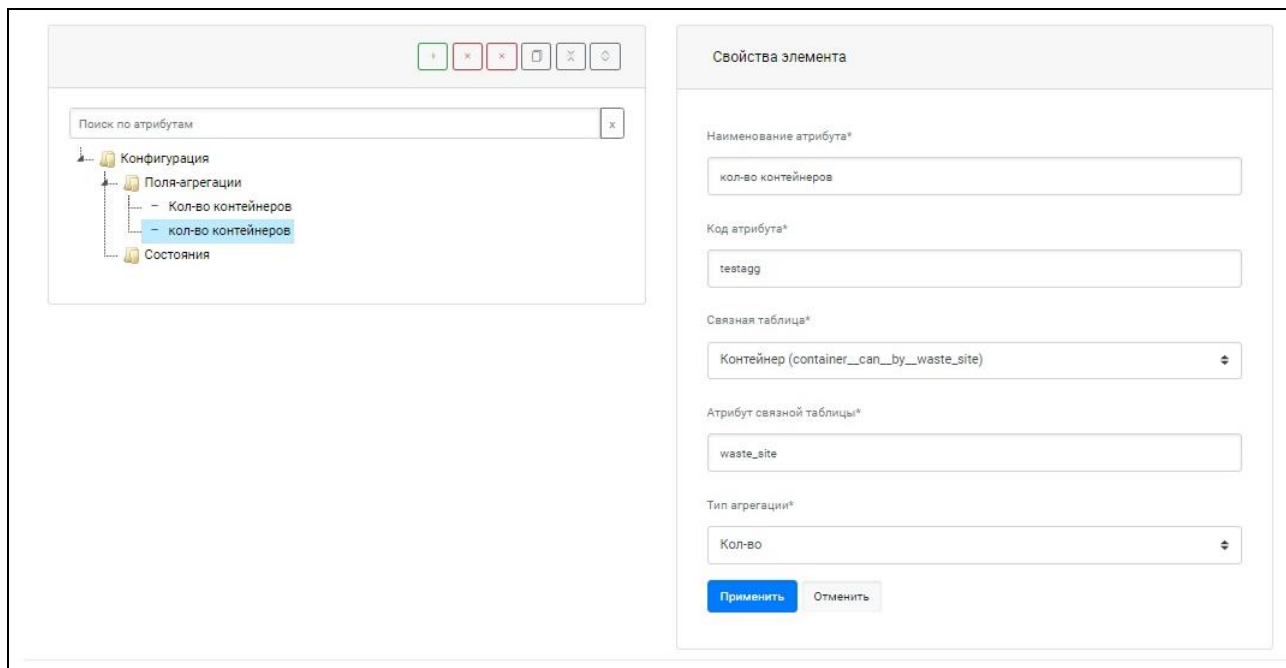
Запрос к полям агрегации идет отдельно от запроса основных свойств.

В случае с формой экспорта запрос агрегации делается оптимизированно для всей выборки.

Для того чтобы добавить поля-агрегации в форму элемента/списка, необходимо сначала создать поля в типе представления, а затем добавить их на форму. Необходимо выполнить следующие действия:

1. Перейдите в раздел конструктора «Документооборот → 1. Типы представлений».
2. Откройте на редактирование тип представления, в который необходимо добавить поля-агрегации.

3. В разделе формы «Конфигуратор» нажмите на папку «Поля-агрегации», затем нажмите кнопку «+», чтобы добавить новое поле-агрегацию. В столбце «Свойства элемента» заполните параметры поля (рис. 33):



Добавление нового поля-агрегации

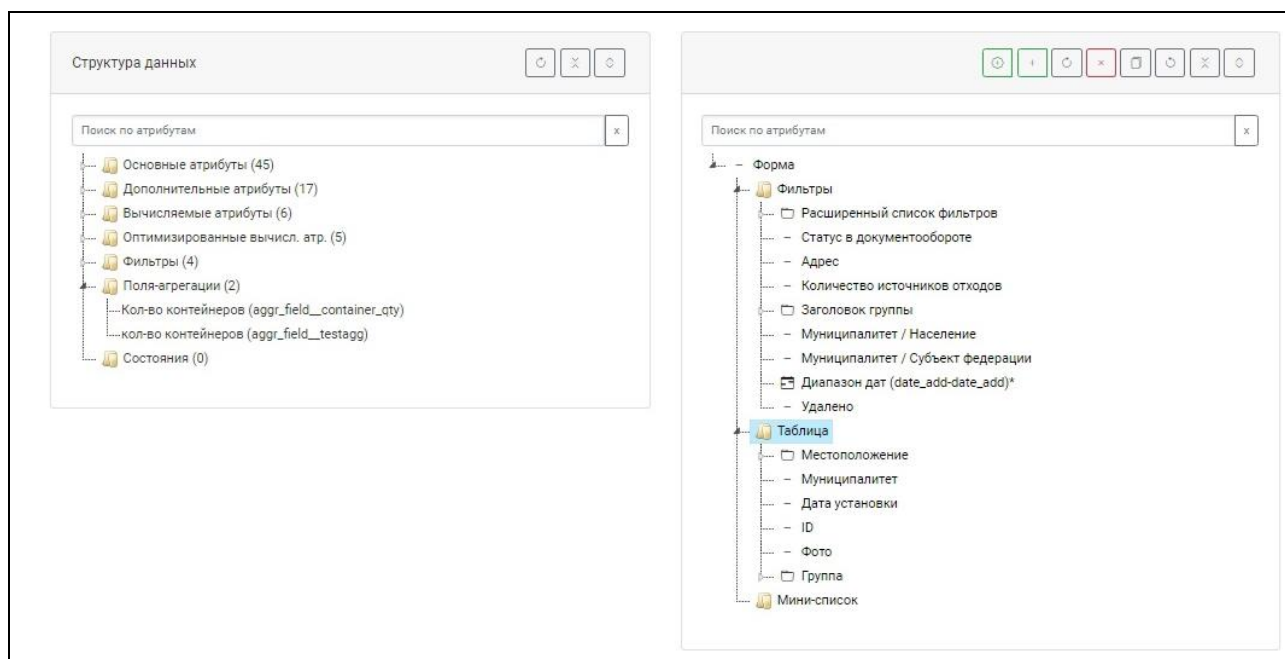
Рис. 33

- в поле «Наименование атрибута» введите название;
- в поле «Код атрибута» введите уникальный код, используя буквы латинского алфавита;
- в поле «Связная таблица» выберите связанную таблицу, из которой будут загружаться данные для поля;
- в поле «Атрибут связанной таблицы» начните вводить название или код атрибута. Доступные атрибуты связанной таблицы отобразятся списком, в котором выберите нужный. По этому атрибуту будет выполняться функция агрегации;
- в поле «Тип агрегации» выберите функцию агрегации из доступных.

Поле отображается в зависимости от типа связанного атрибута. Если тип атрибута - текст, то применяется агрегация по количеству. Если тип атрибута - число, то доступны различные функции, например, максимум, минимум, количество уникальных значений и т.п.

4. Сохраните тип представления.
5. Перейдите в раздел конструктора «Конструктор форм → 1. Формы».
6. Откройте форму элемента/списка, построенную на основе типа представления, в который были добавлены поля-агрегации.

7. В разделе формы «Конструктор» в столбце «Структура данных» в папке «Поля-агрегации» будут доступны созданные в типе представления поля-агрегации (рис. 34).



Доступные в типе представления поля-агрегации

Рис. 34

8. Для добавления поля на форму в среднем столбце выберите элемент структуры, куда необходимо добавить поле, и дважды кликните по полю-агрегации в столбце «Структура данных».

9. Сохраните изменения.

4.9.2. Добавление состояния на форму элемента/списка

Состояние – это значение свойства, установленное в связанной с объектом записи во внешней таблице.

Для того чтобы состояние можно было использовать, у связанной с объектом таблицы обязательно должны быть:

- внешний ключ на объект, которому добавляем состояние;
- поле даты-времени с автопроставлением (с названием timestamp или datetime_create);
- поле значения состояния.

При этом связанная таблица может быть как в PostgreSQL, так и в ClickHouse.

Состояния можно добавлять в форму как обычные свойства, при этом их значение будет зависеть от контекста времени (по умолчанию - текущее время), заданного для формы.

Контекст времени может быть задан в настройках формы явно (при помощи формул) или при помощи компонента «Таймлайн».

Состояния добавляются в форме по аналогии с полями агрегации.

В настройках поля указывается свойство связанного типа представления, отображающее показатель, поле связанной таблицы, отвечающее за дату-время (для формы списка это свойство должно быть указано в фильтре).

Запрос состояний происходит отдельно от основного запроса элемента и может содержать контекст времени (по умолчанию - текущее время).

Для форм списка запрос происходит по каждому состоянию, но сразу по группе объектов.

Для форм экспорта запросы делаются оптимизировано для всех объектов выгрузки.

Для того чтобы добавить состояния в форму элемента/списка, необходимо сначала создать состояния в типе представления, а затем добавить их на форму. Необходимо выполнить следующие действия:

1. Перейдите в раздел конструктора «Документооборот → 1. Типы представлений».
2. Откройте на редактирование тип представления, в который необходимо добавить состояния.
3. В разделе формы «Конфигуратор» нажмите на папку «Состояния», затем нажмите кнопку «+», чтобы добавить новое состояние. В столбце «Свойства элемента» заполните параметры состояния:
 - в поле «Наименование атрибута» введите название;
 - в поле «Код атрибута» введите уникальный код, используя буквы латинского алфавита;
 - в поле «Связная таблица» выберите связанную таблицу, из которой будут загружаться данные для состояния;
 - в поле «Атрибут связанной таблицы» начните вводить название или код атрибута. Доступные атрибуты связанной таблицы отобразятся списком, в котором выберите нужный. По этому атрибуту будет выполняться функция агрегации;
 - в поле «Тип агрегации» выберите функцию агрегации из доступных (последнее значение или первое);
 - в поле «Сортировка по» выберите атрибут, по которому будут отсортированы значения атрибута связанной таблицы;
 - в поле «Направление сортировки» выберите как будут сортироваться значения.
4. Сохраните тип представления.
5. Перейдите в раздел конструктора «Конструктор форм -> 1. Формы».

6. Откройте форму элемента/списка, построенную на основе типа представления, в который были добавлены состояния.

7. В разделе формы «Конструктор» в столбце «Структура данных» в папке «Состояния» будут доступны созданные в типе представления состояния.

8. Для добавления состояния на форму в среднем столбце выберите элемент структуры, куда необходимо добавить состояние, и дважды кликните по состоянию в столбце «Структура данных».

9. Сохраните изменения.

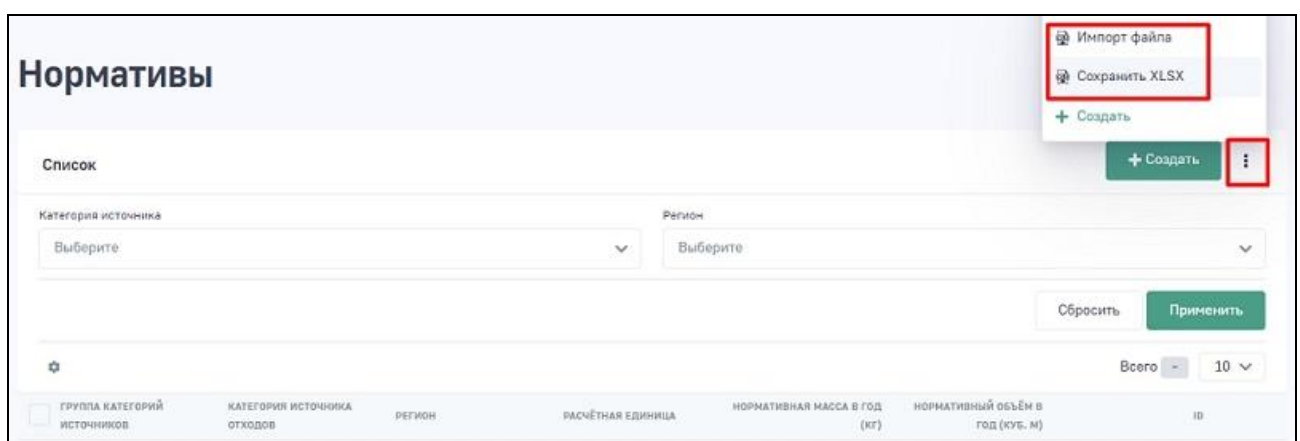
4.9.3. Добавление состояния на карте

Состояния на карте работают также, как и агрегации. Можно использовать в формулах для расцветки точек, в этом случае на большом зуме будет работать группировка по кластерам координат родительских объектов.

4.10. Настройка операций импорта/экспорта в Excel

В Платформе реализована возможность настройки операции импорта/экспорта данных в табличный формат. Операция экспорта позволяет выгружать данные из формы списка в формате .XLSX или .CSV. Операция импорта позволяет загружать данные в форму списка через файл формата .XLSX.

При добавлении операции импорта/экспорта на форму списка необходимо также создать форму экспорта/импорта (рис. 35).



Операции экспорта/импорта в Excel

Рис. 35

Доступность данных операций в форме настраивается в админ-панели в форме редактирования списка в разделе «Дополнительные настройки» (рис. 36).

Дополнительные настройки

Детальная форма: Форма элемента "Норматив отходаобразования" для Нормативы отходаобразования

Мини форма: -----

Не показывать список если есть календарь или карта

Мультикарта

Доступные операции:

- Загрузить для работы офлайн (download_for_offline_mode_all)
- Загрузить для работы офлайн (download_for_offline_mode_partial)
- Импорт файла (import_xlsx)
- Сохранить CSV (load_csv)
- Сохранить JSON (load_json)
- Сохранить XLSX (load_xlsx)
- Скачать печатные формы (print_archive)
- Удалить (remove_operation)
- Подписать печатные формы (signatures_for_print_forms)

Показывать примерное кол-во на первой странице списка

Внешний вид фильтра: Как поля ввода

Тип таблицы: Обычный

Форма редактирования списка в разделе «Дополнительные настройки»

Рис. 36

По умолчанию данные выгружаются/загружаются по формату детальной формы. При этом можно задать другой формат, выбрав в разделе «Настройки импорта/экспорта» форму экспорта (рис. 37).

Настройки импорта/экспорта

Форма экспорта: Форма экспорта "Основная форма" для Акт проверки

Форма экспорта JSON: -----

Настройка формы импорта/экспорта

Рис. 37

При этом формат для экспорта и импорта совпадает. Т.е. сейчас нельзя выбрать одну форму для импорта, а другую – для экспорта.

Также формой импорта/экспорта может быть только форма, привязанная к тому же типу представления, что и форма списка.

При выгрузке учитываются примененные в форме списка фильтры. Т.е. в выгрузку попадут только те данные, которые соответствуют выбранным фильтрам.

Для создания и настройки формы импорта/экспорта необходимо:

1. Перейти в админ-панель и создать новую форму.
2. В форме добавления:
 - укажите наименование формы;
 - выберите тип формы «Форма экспорта»;
 - выберите тип представления (такой же, как и в форме списка, для которой создается форма экспорта) (рис. 38).

Добавить Форма

Основные данные

Name: Основная Код: оспчппау
По умолчанию - Основная По умолчанию формируется как DocType.code-listform

Тип формы: Форма элемента

Активен: Да

Использовать по умолчанию

Тип представления:

СОХРАНИТЬ

Форма добавления форма импорта/экспорта

Рис. 38

3. Нажмите «Сохранить и продолжить редактирование».
4. Теперь можно настроить форму. Добавьте атрибуты, которые будут выгружаться/загружаться в форме импорта/экспорта (рис. 39).

Конструктор

Структура данных

Поиск по атрибутам

- Основные атрибуты (38)
- Дополнительные атрибуты (7)
- Вычисляемые атрибуты (9)

Конфигурации

Превью формы Конфигурация формы Конфигурация фронта

Свойства элемента Условия Валидаторы

↑ Главное

Название формы

Недоступна для редактирования

Только для чтения
Для таблиц на основе моделей кликаус редактирование недоступно

Показывать только по выбору пользователя

Описание

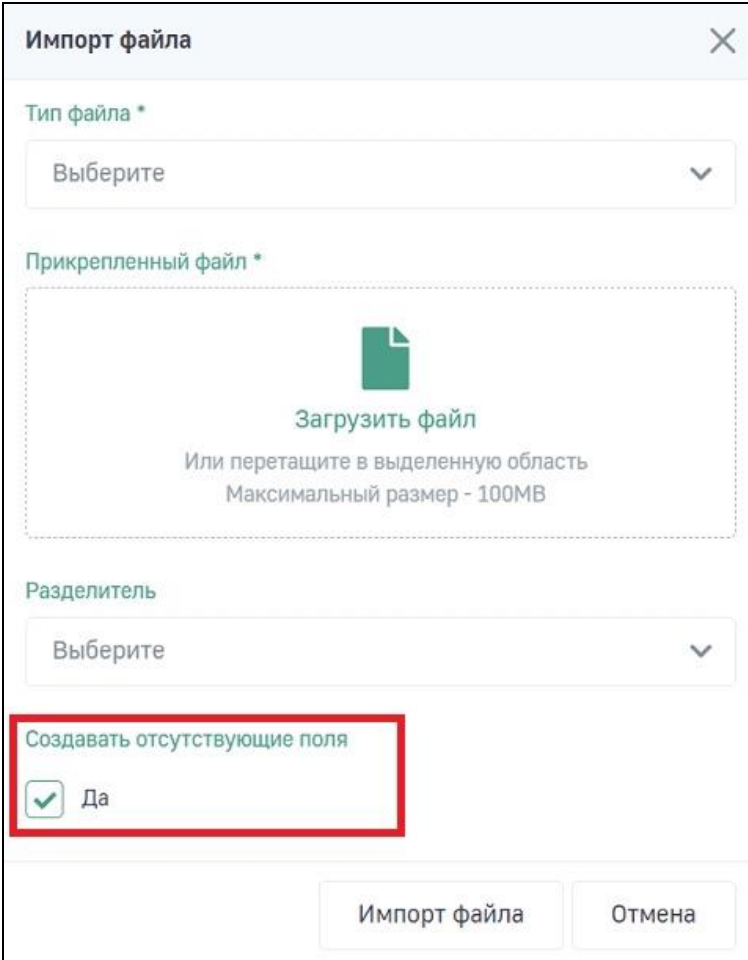
Настройка формы импорта/экспорта

Рис. 39

Дополнительные особенности импорта данных:

Импорт данных работает только для обычных таблиц. Если данные отображаются в витрине, то импорт работать не будет.

При импорте данных также можно поставить галочку «Создавать отсутствующие поля» (рис. 40). В таком случае, если в файле импорта будут присутствовать дополнительные колонки, то соответствующие атрибуты будут добавлены в разделе «Дополнительные атрибуты» формы списка.



Настройка дополнительных возможностей импорта данных

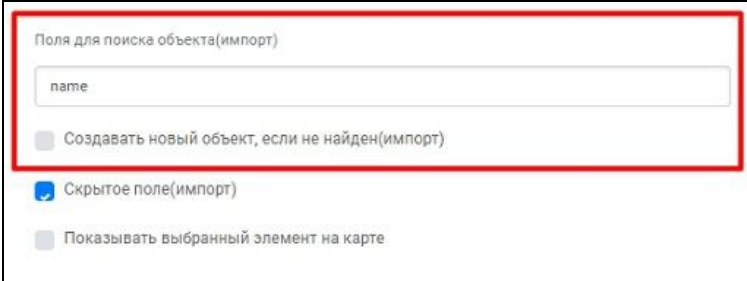
Рис. 40

Видимость полей во всплывающей форме импорта настраивается, как и во всех остальных видах форм.

Для загрузки связанных объектов в настройках формы экспорта/импорта можно указать, по каким полям осуществлять поиск и привязку объектов. По умолчанию поиск происходит по полю «name». Если при настройке формы экспорта/импорта в свойстве «Поля для поиска объекта (импорт)» указать через запятую другие поля, то сервис будет искать первый попавшийся по этому набору полей связанный объект и привязывать его к загружаемому элементу.

Например, есть контейнеры, у каждого контейнера есть МНО. Когда привязываем в импорте данные, необходимо указать в свойстве id МНО, но если это свойство не заполнено, то по умолчанию будет осуществляться поиск в столбце «МНО.name». А если это свойство заполнено, то по указанным в свойстве полям. Чаще всего для МНО это поля lat, lon. Если заполнить свойство таким образом, то в файле будет происходить поиск таких колонок, и по данным из них будет находиться МНО.

Также для импорта в настройках формы экспорта/импорта можно включить функцию «Создавать новый объект, если не найден(импорт)» (рис. 41). При этом если при загрузке данных не находится объект по указанному для поиска набору полей, то создается новый объект и привязывается к загружаемому элементу (все обязательные поля объекта при этом должны быть также заполнены в загружаемом файле XLSX).



The image shows a screenshot of a web form titled "Поля для поиска объекта(импорт)". It contains a text input field with the value "name". Below the input field are three checkboxes: "Создавать новый объект, если не найден(импорт)" (unchecked), "Скрытое поле(импорт)" (checked), and "Показывать выбранный элемент на карте" (unchecked).

Добавление функции «Создавать новый объект, если не найден(импорт)»

Рис. 41

Для связанных объектов также можно использовать свойство «Скрытое поле (импорт)». Для этого в форме должно быть добавлено родительское поле связанного объекта.

Например, в форме контейнеров должно быть добавлено поле МНО и необходимые дочерние поля, например, «МНО.lat», «МНО.lon», «МНО.name». Но иногда для отображения в файле экспорта родительское поле не нужно, тогда ставится отметка скрывания, и в выгружаемом файле колонки МНО не будет, но все дочерние поля останутся.

После выполнения импорта пользователю становится доступен для скачивания отчет об импорте. В данном отчете жирным выделяются успешно загруженные строки. Если при импорте строки возникла ошибка, то она указывается в столбце Errors, строка при этом не выделяется жирным (рис. 42).

	A	B	C	D	E	F	G	H	I
1	id	waste_sou	normative	normative	errors				
2		Ресторан	12	посетител	'NoneType' object has no attribute 'strftime'				
3	+	37 Спортивн	120.000						
4		35 Индивиду	1072.000						
5		34 Спортивн	1321.000						
6		33 Универса	1072.000						
7		32 Научно-и	1552.000						
8		31 Крематор	1331.000						
9		30 Кафе, рес	1552.000						
10		29 Детские д	341.000						
11		28 Бассейны	511.000						
12		27 Кафе, рес	123.000						
13		26 Предприя	1234.000						
14		25 Гаражи, п	1234.000						
15		24 Организа	1234.000						

Использование свойства «Скрытое поле(импорт)»

Рис. 42

ManyToMany поля:

- при экспорте по умолчанию возвращает список id в ячейке
- при импорте по умолчанию понимает список id, в настройках можно указать поле поиска, тогда можно передать строку в формате csv содержащие значения для поиска.

4.11. Работа с публичными сайтами

В Платформе реализована возможность создания публичных сайтов. Для доступа к публичному сайту не требуется авторизация, т.е. любой желающий может посмотреть информацию, опубликованную на таком сайте.

Для публичных сайтов на Платформе реализовано дополнительное кэширование, чтобы выдерживать бóльшую нагрузку, чем обычно бывает в частных кабинетах.

Публичные сайты создаются через публичные стенды клиентской части Платформы.

Публичный стенд – это стенд, на котором отключены все виды авторизации. Т.е. любой пользователь работает на этом стенде под гостевой неавторизованной виртуальной учетной записью.

С одним и тем же бэк-стендом могут работать несколько стендов клиентской части Платформы. Любой из фронт-стендов может быть «публичным».

На публичном стенде доступны только публичные кабинеты – это кабинеты, которые помечены в админ-панели как публичные (public=true). Непубличные кабинеты доступны только на непубличных стендах.

4.11.1. Создание публичного сайта

В админ-панели перейдите в раздел «Конструктор форм → 7. Настройки приложения». Добавьте новое приложение. Заполните данные по приложению и отметьте поле «Публичный стенд (без функционала авторизации)» (рис. 43).

Почта техподдержки:

Показывать техническую поддержку на внутренних страницах приложения

Показывать техническую поддержку на страницах авторизации

Dadata-key:

Публичный стенд (без функционала авторизации)

Версия:

Кастомный CSS:

Личный кабинет:
Страница организации:
Код формы:

Личный кабинет:
Страница организации:
Заголовок:

Вход через госуслуги

Текст для пустого списка организаций из ЕСИА:

Вы не добавлены в перечень сотрудников ни одной организации в ЕСИА.

Форма создания публичного сайта

Рис. 43

В админ-панели перейдите в раздел «Rules → Типы компаний». Создайте новый тип компании (кабинет) или измените существующий. Отметьте поле «Показывать без авторизации», чтобы сделать кабинет публичным (рис. 44).

Изменить Тип компании

Посетитель

Наименование:

Заголовок:

Символьный код:

Показывать без авторизации

Дерево меню:

```
{ "id": 63, "children": [ { "id": 15216, "children": [ ] }, { "id": 21, "children": [ ] }, { "id": 14418, "children": [ ] }, { "id": 11014, "children": [ ] }, { "id": 11091, "children": [ ] }, { "id": 11830, "children": [ ] }, { "id": 10759, "children": [ ] }, { "id": 10757, "children": [ ] }, { "id": 7, "children": [ ] }, { "id": 119, "children": [ ] }, { "id": 14407, "children": [ ] }, { "id": 191, "children": [ ] }, { "id": 137, "children": [ ] }, { "id": 729, "children": [ ] }, { "id": 87, "children": [ ] }, { "id": 212, "children": [ ] }, { "id": 211, "children": [ ] } ] }
```

Поле «Показывать без авторизации»

Рис. 44

В публичном кабинете должно быть настроено меню. Отдельно задавать публичность пунктов меню не требуется. Один и тот же пункт меню может быть и публичным, и приватным, в зависимости от того, к каким кабинетам он привязан (рис. 45).

Изменить Пункт главного меню

О компании

Наименование:

Код:

Активен

Изображение иконки: Файл не выбран

Код иконки:

Типы компаний:

- транспортировщик
- Кабинет
- Отходообразователь
- Посетитель
- Управляющая компания
- Орган местного самоуправления
- Региональный оператор
- Контролирующий орган
- Компания перевозчик

Удерживайте "Control" (или "Command" на Mac), чтобы выбрать несколько значений.

Форма списка:

Форма объекта:

Форма лендинга:

Родитель:

URL:
Если заполнен, ссылка главного меню будет вести на данных url

Изображение баннера: Файл не выбран

Заголовок под баннером:

Текст под заголовком:

Настройка меню публичного кабинета

Рис. 45

На страницах, на которые ведут пункты меню, должны быть правильно настроены права доступа к данным в разделе «Документооборот → Права на CRUD». Отметьте поле «Публичное», чтобы действие стало доступно на публичном сайте (рис. 46).

Изменить Право на CRUD по статусам

Адрес компании get None ИСТОРИЯ

Роли:

- carrier_company_admin
- carrier_company
- reg_operator_admin
- reg_operator
- landfill_operator_admin
- landfill_operator
- emitter_admin
- emitter
- guest

Удерживайте "Control" (или "Command" на Mac), чтобы выбрать несколько значений.

Тип представления: Адрес компании

Действие: Просмотр

Статус в документообороте: -----

Правило: -----

Публичное

Удалено | Сохранить и добавить другой объект | Сохранить и продолжить редактирование | СОХРАНИТЬ

Действия доступные на публичном сайте

Рис. 46

4.12. Модуль «Онбординг»

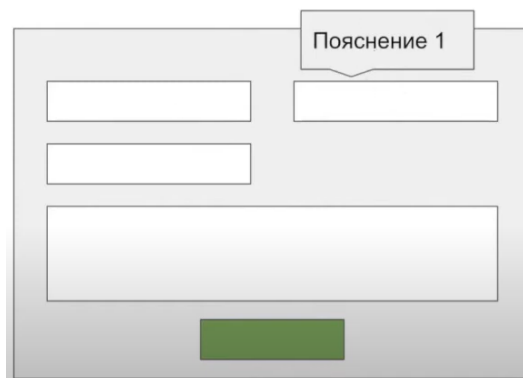
На Платформе реализован модуль «Онбординг», который предоставляет возможность обучения пользователя информационных систем непосредственно на страницах, которые он просматривает.

Модуль «Онбординг» позволяет создавать уроки (подсказки) на страницах информационной системы, которые появляются в виде всплывающих подсказок.

Администратор Платформы может привязывать любое количество уроков к страницам (формам) информационной системы и задавать права доступа к ним для пользователя.

Рассмотрим принцип работы процедуры онбординга.

Предположим, что у нас есть форма, которая содержит несколько полей. Процедура онбординга позволяет создать некий набор всплывающих пояснений, которые будут появляться либо в центре страницы, либо в той области, в которой находится объект пояснения (рис. 47).



Принцип работы процедуры онбординга

Рис. 47

Пояснения могут содержать текстовые элементы и/или графические элементы.

После того как пользователь перемещается от первого элемента к последнему, факт прохождения урока сохраняется в базу данных. Это делается для того, чтобы не показывать один и тот же урок пользователю дважды без необходимости.

Для того, чтобы прикреплять элементы онбординга (пояснения, подсказки) к определенным областям HTML-страниц необходимо понимать как можно эти элементы различать.

Различать элементы на странице можно при помощи CSS-селекторов.

CSS-селекторы – это некоторые текстовые записи, которые позволяют однозначно определить элемент или группу элементов на странице. При просмотре HTML-страниц могут быть отображены следующие теги:

```
<button class='testclass1 testclass2' id='test' rel='testrel' >Кнопка</button>
```

где `button` – название тега; `class`, `id`, `rel` – атрибуты тега; «Кнопка» – содержимое тега.

Для того, чтобы обратиться к конкретному элементу или его идентифицировать используются различные варианты селекторов: по названию тега и/или обращению по классу (классы могут быть разделены пробелами при перечислении).

В данном примере к конкретному элементу можно обратиться: по названию `button`, по первому классу `.testclass1`, по второму классу `.testclass2`, по идентификатору `#test` и по любому другому атрибуту. Для того, чтобы обратиться к конкретному атрибуту, необходимо в квадратных скобках передать название этого атрибута и его значение, т.е. `[rel='testrel']`.

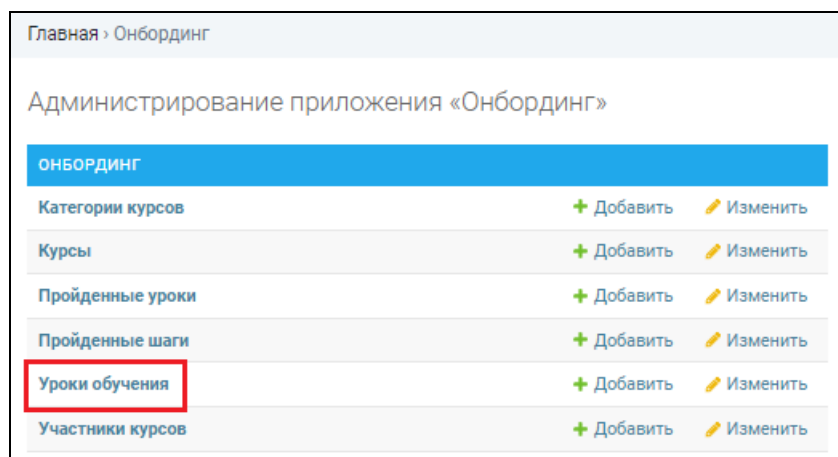
Примечание. Обращение по классу начинается с точки «.», обращение к идентификатору – с решетки «#», к любому другому атрибуту – с квадратных скобок «`[rel='testrel']`».

При этом только id однозначно идентифицирует конкретный элемент. Остальные атрибуты (название, классы) однозначно идентифицируют группу объектов на странице и поэтому применяются реже.

4.12.1. Создание урока (подсказки) на HTML-странице

Создадим подсказку для пользователя информационной системы при первичном его входе на детальную страницу. Для этого выполним следующие действия.

1. Зайдем в админ-панель Платформы и выберем раздел «Уроки обучения» в блоке «ОНБОРДИНГ» (рис. 48).



Раздел «Уроки обучения» в блоке «ОНБОРДИНГ»

Рис. 48

2. В разделе «Уроки обучения» с помощью кнопки «ДОБАВИТЬ УРОК ОБУЧЕНИЯ +» откроем форму добавления урока обучения (рис. 49).

Добавить Урок обучения

Название урока:

Форма, на которой показывать урок:

Только для ролей:

Урок доступен только пользователям перечисленных ролей, если ни одной роли не указано, то урок доступен любому пользователю. Удерживайте "Control" (или "Command" на Mac), чтобы выбрать несколько значений.

ШАГИ УРОКА

Шаг урока: #1

Порядковый номер:

Текст заголовка:

Текст описания:

Изображение: Файл не выбран

Селектор:

Используйте css-селекторы для того чтобы шаг урока показался в нужном месте экрана например, .classname, #some_id и т.д.

Шаг урока: #2

Порядковый номер:

Форма добавления урока обучения

Рис. 49

Форма добавления урока обучения состоит из следующих блоков со следующими полями:

– основные сведения:

- «Название урока» – поле ввода, в котором указывается название урока,
- «Форма, на которой показывать урок» – поле с выпадающим списком форм информационной системы, на которых будет отображаться урок,
- «Только для ролей» – поле с перечнем ролей пользователей, для которых будет доступен данный урок;

– шаги урока – блок с пошаговыми разделами урока (в каждом шаге урока поля повторяются):

- «Порядковый номер» порядковый номер урока,
- «Текст заголовка»,
- «Текст описания» – область для ввода текстового описания,

- «Изображение» – поле прикрепления файла с графическим изображением посредством нажатия кнопки «Выбрать файл»,
 - «Селектор» – поле для указания CSS-селектора;
- кнопка «+ Добавить еще один Шаг урока»;
- кнопка «Сохранить и добавить другой объект»;
- кнопка «Сохранить и продолжить редактирование»;
- кнопка «СОХРАНИТЬ».

3. После заполнения формы добавления урока, необходимо нажать кнопку «СОХРАНИТЬ» (рис. 50-рис.52).

The screenshot shows a form for creating a lesson. It has three main sections:

- Название урока:** A text input field containing "Создание пробы".
- Форма, на которой показывать урок:** A dropdown menu showing "Форма элемента 'Основная форма ...". To the right of the dropdown are three icons: a pencil, a plus sign, and a globe.
- Только для ролей:** A list of roles: "admin", "carrier_company_admin", "carrier_company", "reg_operator_admin", "reg_operator", "landfill_operator_admin", "landfill_operator", "emitter_admin", "emitter". To the right of the list is a plus sign icon.

Below the list of roles, there is a small note: "Урок доступен только пользователям перечисленных ролей, если ни одной роли не указано, то урок доступен любому пользователю. Удерживайте "Control" (или "Command" на Mac), чтобы выбрать несколько значений."

Пример заполнения формы добавления урока. Блок основных сведений

Рис. 50

The screenshot shows the "Шаги урока" (Lesson Steps) section of the form. It has a blue header with the text "ШАГИ УРОКА". Below the header, it shows "Шаг урока: <<Шаг урока: Связанные объекты, № 1>>" and a "Удалено" checkbox.

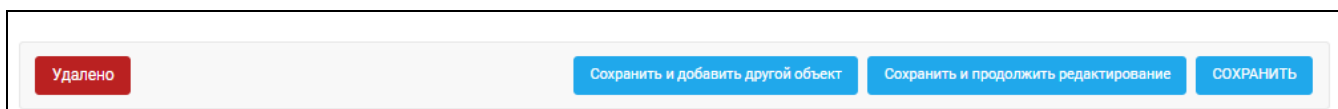
The form contains several fields:

- Порядковый номер:** A text input field containing "1".
- Текст заголовка:** A text input field containing "Связанные объекты".
- Текст описания:** A large text area containing "Ссылки на связанные объекты появятся после сохранения элемента". Below the text area is the label "допустим html" and a small icon.
- Изображение:** A section with the text "На данный момент: media/photo/444.png" and a "Очистить" checkbox. Below it is the text "Изменить:" followed by a "Выберите файл" button and "Файл не выбран".
- Селектор:** A text input field.

Below the "Селектор" field, there is a note: "Используйте css-селекторы для того чтобы шаг урока показался в нужном месте экрана например, .classname, #some_id и т.д."

Пример заполнения формы добавления урока. Блок «Шаг урока: #1»

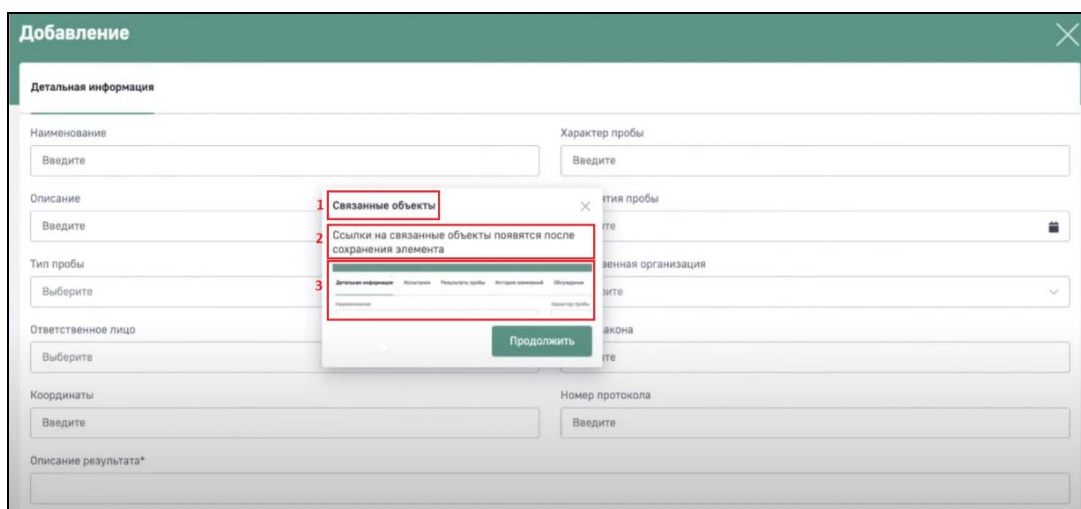
Рис. 51



Командные кнопки формы добавления урока

Рис. 52

В результате выполненных действий при первом входе на страницу информационной системы (в данном примере – это страница добавления пробы) будет отображаться созданный урок (подсказка) (рис. 53),



Результат создания урока (подсказки)

Рис. 53

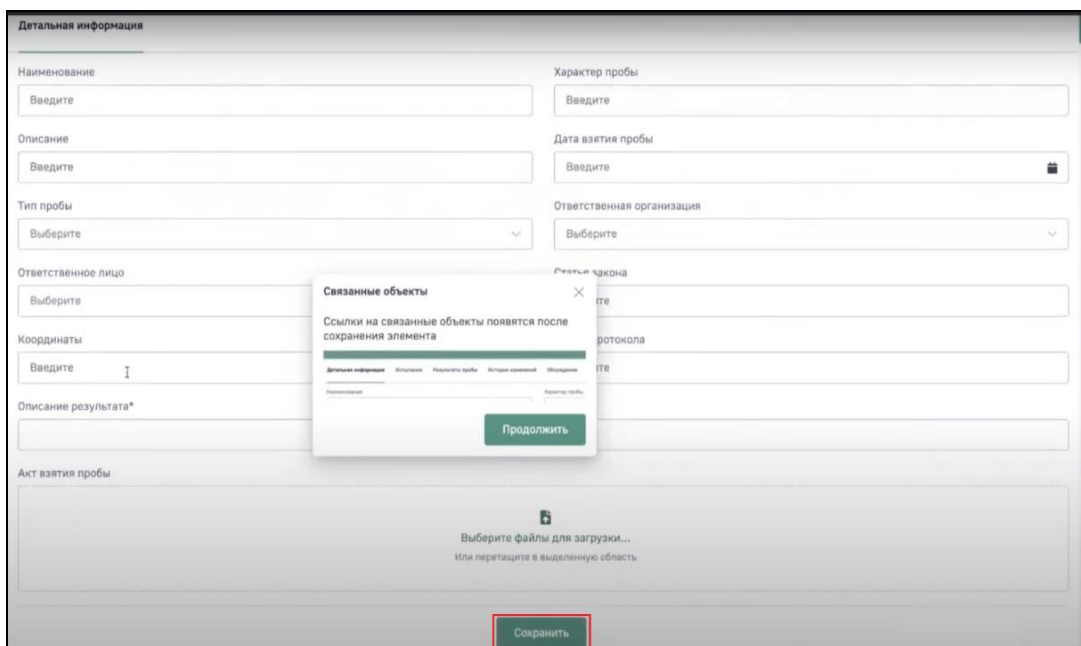
где область 1 – отображение поля «Текст заголовка» блока «Шаг урока: #1»;

область 2 – отображение поля «Текст описания» блока «Шаг урока: #1»;

область 3 – отображение файла с графическим изображением, которое было добавлено в поле «Изображение» блока «Шаг урока: #1».

Создадим подсказку для пользователя информационной системы при первичном его нажатии командной кнопки.

В рассмотренном выше примере, добавим второй шаг урока, благодаря которому будет появляться подсказка при нажатии в детальной странице кнопки «Сохранить» (рис. 54).

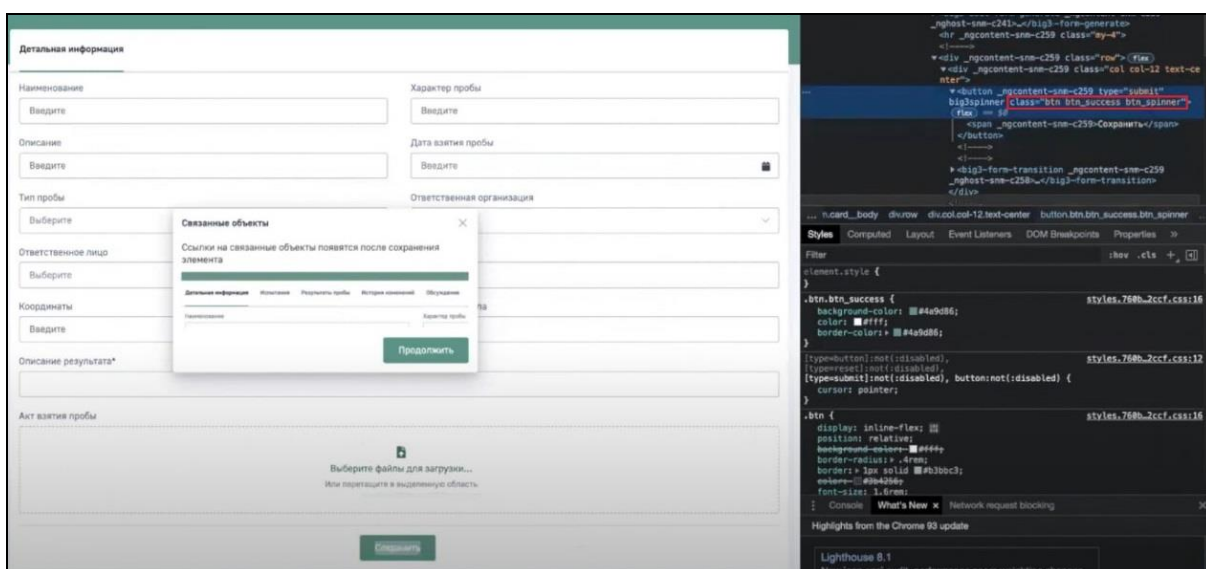


Кнопка «Сохранить» на детальной странице информационной системы

Рис. 54

Для этого нажмите правой клавише мыши кнопку «Сохранить» и из всплывающего меню выберите «Просмотреть код». В результате выполненного действия в правой части экрана будет отображаться код.

В рассматриваемом примере выбранный нами элемент (кнопка «Сохранить») определяется тремя классами: `btn`, `btn_success` и `btn_spinner` (рис. 55). Эти три класса будут служить идентификаторами во втором шаге урока.



Три класса выбранного элемента

Рис. 55

Заполним в форме добавления урока блок «Шаг урока: #2» (рис. 56), а затем нажмем кнопку «Сохранить».

Пример заполнения формы добавления урока. Блок «Шаг урока: #2»

Рис. 56

Примечание. При указании нескольких CSS-селекторов в поле «Селектор», классы перечисляются через точку, т.е. в рассматриваемом примере, это будет выглядеть следующим образом: btn.btn_success.btn_spinner.

В результате выполненных действий, после обновления детальной странице информационной системы будет отображаться первый шаг урока (см. рис. 53). Далее нажав кнопку «Продолжить», рядом с кнопкой «Сохранить» будет отображаться второй шаг урока (рис. 57).

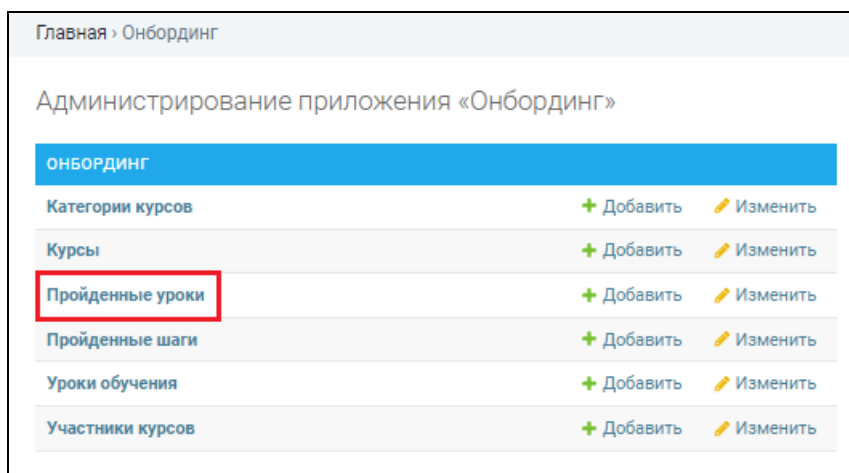
Результат создания второго шага урока (подсказки)

Рис. 57

В подсказке будут отображаться кнопки «Назад» и «Закончить». При выборе кнопки «Назад» пользователь перейдет обратно в первый шаг урока. При выборе кнопки «Закончить»

урок будет окончен. При этом прохождение урока пользователем сохранится в базе данных информационной системе.

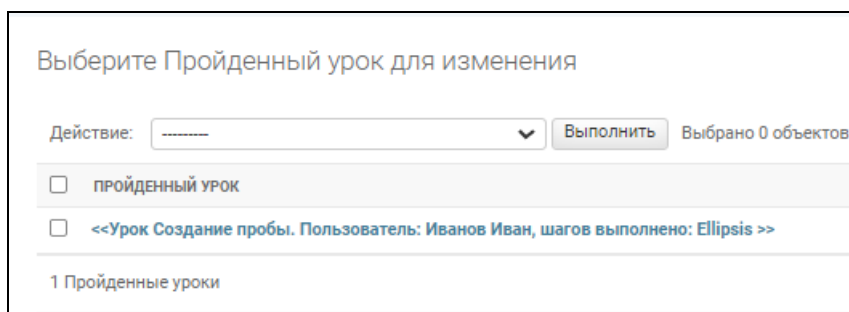
Для того, чтобы просмотреть информацию о прохождении урока необходимо зайти в админ-панель Платформы и выбрать раздел «Пройденные уроки» в блоке «ОНБОРДИНГ» (рис. 58).



Раздел «Пройденные уроки» в блоке «ОНБОРДИНГ»

Рис. 58

В разделе «Пройденные уроки» будут отображаться история пройденных пользователем уроков (рис. 59).



История пройденных пользователем уроков

Рис. 59

5. РЕШЕНИЕ ПРОБЛЕМ

5.1. Техническая поддержка

В случае возникновения проблем пользователь может обратиться в организацию-разработчика по следующим каналам связи:

- электронная почта platform@big3.ru.

Режим работы специалистов организации-разработчика по вопросам технической поддержки Платформы: пн.-пт. 9:00-18:00 (по московскому времени).

5.2. Резервное копирование и восстановление Платформы

Резервное копирование и восстановление Платформы осуществляется в соответствии с Федеральным законом от 27.07.2006 № 149-ФЗ «Об информации, информационных технологиях и о защите информации».

Резервное копирование и восстановление Платформы осуществляется в следующих целях:

- обеспечения бесперебойной работы Платформы;
- определения порядка резервного копирования Платформы;
- определения порядка восстановления Платформы при полной или частичной потере информации, вызванной сбоями или отказами аппаратного или программного обеспечения, ошибками пользователей, стихийными бедствиями или иными чрезвычайными обстоятельствами.

5.2.1. Плановое резервное копирование Платформы

Настройка параметров резервного копирования определяется Администратором самостоятельно. Запуск операций резервного копирования Платформы производится с помощью встроенных средств ОС Linux и СУБД PostgreSQL.

Резервное копирование осуществляется с периодичностью не реже, чем 1 (один) раз в сутки. Полное копирование Платформы должно применяться не реже, чем 1 (один) раз в неделю.

Контроль результатов выполнения процедур резервного копирования осуществляется Администратором в срок до 12:00 рабочего дня следующего за установленной датой выполнения этих процедур.

Контроль результатов резервного копирования производится путем просмотра журналов событий ОС и СПО. В случае обнаружения сбоя планового резервного копирования, Администратор должен выполнить процедуру внепланового резервного копирования Платформы.

5.2.2. Внеплановое резервное копирование Платформы

Внеплановое резервное копирование Платформы выполняется Администратором в следующих случаях:

- сбой планового резервного копирования Платформы;
- установка на сервер нового программного обеспечения, модификация установленного программного обеспечения;
- наличие необходимости создания копии Платформы перед выполнением работ.

Действия Администратора при внеплановом сохранении информационных ресурсов Платформы аналогичны действиям при плановом резервном копировании.

Если внеплановое резервное копирование производится по причине сбоя планового, то Администратор должен выявить и устранить причину возникновения сбоя, а именно:

- проверить функционирование аппаратной части сервера (дисковые массивы, оперативную память, сетевую карту). Устранить причины сбоя путем замены вышедших из строя компонентов;
- проверить функционирование ОС сервера. Устранить причины сбоя путем перенастройки, установки обновлений, переустановки ОС;
- проверить доступности сервера из локальной вычислительной сети;
- проверить функционирование сетевых сервисов ОС, сетевых настроек. Устранить причины сбоя путем переустановки и перенастройки сетевых сервисов.

5.2.3. Восстановление информации из резервных копий

Восстановление данных из резервных копий производится Администратором в случаях неработоспособности серверов или сервисов, либо на основании служебной записки ответственного за наполнение и работоспособность ПО, если содержимое информационных ресурсов повреждено или нарушена работоспособность Платформы.

Для восстановления данных из резервных копий Администратор использует встроенные средства ОС Linux и СУБД PostgreSQL.

5.3. Аварийные ситуации

Аварийные ситуации, при которых выполнение основных функций Платформы невозможно, и способы их устранения приведены в таблице 7.

Таблица 7 – Аварийные ситуации и способы их устранения

№ п/п	Аварийная ситуация	Признаки	Способ устранения
1	Отсутствие связи с сервером Платформы	Сервер Платформы выключен	Запустить на выполнение сервер Платформы
		Сервер Платформы не работает	Восстановить работоспособность сервера Платформы и запустить его на выполнение
2	Отсутствие сетевого соединения с сервером Платформы	<p>Отсутствует сетевое соединение с сервером Платформы.</p> <p>Отсутствует физическое соединение по сети между сервером и клиентом:</p> <ul style="list-style-type: none"> – выключено или неисправно активное сетевое оборудование; – нарушена целостность физического тракта сети (деформированы сетевые кабели, отсутствует контакт в коннекторах). 	<p>Восстановить соединение с сервером Платформы.</p> <p>Восстановить физическое соединение:</p> <ul style="list-style-type: none"> – отремонтировать и включить активное сетевое оборудование; – заменить неисправные сетевые кабели исправными, восстановить контакты на исправных кабелях
		Отсутствуют (изменены или удалены) корректные сетевые настройки для связи между сервером и клиентом	Восстановить системные настройки на сервере и клиенте, позволяющие устанавливать сетевые соединения между сервером и клиентом

ПРИЛОЖЕНИЕ А. АРИФМИТИЧЕСКИЕ ФУНКЦИИ

Функция	Описание
<code>math.abs(x)</code>	Вычисление абсолютного значения числа
<code>math.add(x, y)</code>	Добавьте два или более значений, $x + y$
<code>math.cbrt(x [, allRoots])</code>	Вычислите кубический корень из значения
<code>math.ceil(x)</code>	Округлите значение до плюс бесконечности, если x оно сложное, как действительная, так и мнимая части округляются до плюс бесконечности
<code>math.cube(x)</code>	Вычислите куб значения, $x * x * x$
<code>math.divide(x, y)</code>	Разделите два значения, x / y
<code>math.dotDivide(x, y)</code>	Разделите две матрицы поэлементно
<code>math.dotMultiply(x, y)</code>	Поэлементно умножьте две матрицы
<code>math.dotPow(x, y)</code>	Вычисляет мощность от x до y по элементам
<code>math.exp(x)</code>	Вычисление экспоненты значения
<code>math.expml(x)</code>	Вычислите значение вычитания 1 из экспоненциального значения
<code>math.fix(x)</code>	Округлите значение до нуля
<code>math.floor(x)</code>	Округлите значение до минус бесконечности
<code>math.gcd(a, b)</code>	Вычислите наибольший общий делитель для двух или более значений или массивов
<code>math.hypot(a, b, ...)</code>	Вычислите гипотенузу списка со значениями
<code>math.invmod(a, b)</code>	Вычислите (модульную) мультипликативную инверсию a по модулю b
<code>math.lcm(a, b)</code>	Вычислите наименьшее общее кратное для двух или более значений или массивов
<code>math.log(x [, base])</code>	Вычисление логарифма значения
<code>math.log10(x)</code>	Вычислите логарифм значения по 10 основаниям
<code>math.log1p(x)</code>	Вычислите логарифм a value+1
<code>math.log2(x)</code>	Вычислите 2-основание значения
<code>math.mod(x, y)</code>	Вычисляет модуль, остаток от целочисленного деления
<code>math.multiply(x, y)</code>	Умножьте два или более значений, $x * y$
<code>math.norm(x [, p])</code>	Вычислите норму числа, вектора или матрицы
<code>math.nthRoot(a)</code>	Вычислите n -й корень значения
<code>math.nthRoots(x)</code>	Вычислите n -е корни значения
<code>math.pow(x, y)</code>	Вычисляет степень от x до y , $x ^ y$
<code>math.round(x [, n])</code>	Округлите значение до ближайшего округленного значения
<code>math.sign(x)</code>	Вычислите знак значения
<code>math.sqrt(x)</code>	Вычислите квадратный корень из значения

Функция	Описание
math.square(x)	Вычислите квадрат значения, $x * x$
math.subtract(x, y)	Вычтите два значения, $x - y$
math.unaryMinus(x)	Переверните знак значения, примените операцию унарного минуса
math.unaryPlus(x)	Операция «Унарный плюс»
math.xgcd(a, b)	Вычислите расширенный наибольший общий делитель для двух значений

ПРИЛОЖЕНИЕ Б. СТАТИСТИЧЕСКИЕ ФУНКЦИИ

Функция	Описание
math.cumsum(a, b, c, ...)	Вычислите совокупную сумму матрицы или списка со значениями
math.mad(a, b, c, ...)	Вычислите среднее абсолютное отклонение матрицы или списка со значениями
math.max(a, b, c, ...)	Вычислите максимальное значение матрицы или списка со значениями
math.mean(a, b, c, ...)	Вычислите среднее значение матрицы или списка со значениями
math.median(a, b, c, ...)	Вычислите медиану матрицы или списка со значениями
math.min(a, b, c, ...)	Вычислите минимальное значение матрицы или списка значений
math.mode(a, b, c, ...)	Вычисляет режим набора чисел или списка со значениями (числами или символами)
math.prod(a, b, c, ...)	Вычислите произведение матрицы или списка со значениями
math.quantileSeq(A, prob[, отсортировано])	Вычислите квантиль вероятностного порядка матрицы или списка со значениями
math.std(a, b, c, ...)	Вычислите стандартное отклонение матрицы или списка со значениями
math.sum(a, b, c, ...)	Вычислите сумму матрицы или списка со значениями
math.variance(a, b, c, ...)	Вычислите дисперсию матрицы или списка со значениями

ПРИЛОЖЕНИЕ В. ТРИГОНОМЕТРИЧЕСКИЕ ФУНКЦИИ

Функция	Описание
<code>math.acos(x)</code>	Вычисление обратного косинуса значения
<code>math.acosh(x)</code>	Вычислите гиперболические дуги значения, определенного как $\operatorname{acosh}(x) = \ln(\sqrt{x^2 - 1} + x)$
<code>math.acot(x)</code>	Вычислите обратный котангенс значения, определенного как $\operatorname{acot}(x) = \operatorname{atan}(1/x)$
<code>math.acoth(x)</code>	Вычислите гиперболический арккотангенс значения, определенного как $\operatorname{acoth}(x) = \operatorname{atanh}(1/x) = (\ln((x+1)/x) + \ln(x/(x-1))) / 2$
<code>math.acsc(x)</code>	Вычислите обратный косекант значения, определенного как $\operatorname{acsc}(x) = \operatorname{asin}(1/x)$
<code>math.acsch(x)</code>	Вычислите гиперболическую арккосекансу значения, определенного как $\operatorname{acsch}(x) = \operatorname{asinh}(1/x) = \ln(1/x + \sqrt{1/x^2 + 1})$
<code>math.asec(x)</code>	Вычислите обратную секущую значения
<code>math.asech(x)</code>	Вычислите гиперболическую дуговую секущую значения, определенного как $\operatorname{asech}(x) = \operatorname{acosh}(1/x) = \ln(\sqrt{1/x^2 - 1} + 1/x)$
<code>math.asin(x)</code>	Вычислите обратный синус значения
<code>math.asinh(x)</code>	Вычислите гиперболическую дугообразную линию значения, определенного как $\operatorname{asinh}(x) = \ln(x + \sqrt{x^2 + 1})$
<code>math.atan(x)</code>	Вычислите обратную касательную к значению
<code>math.atan2(y, x)</code>	Вычислите обратную касательную функцию с двумя аргументами, y / x
<code>math.atanh(x)</code>	Вычислите гиперболический арктангенс значения, определенного как $\operatorname{atanh}(x) = \ln((1 + x)/(1 - x)) / 2$
<code>math.cos(x)</code>	Вычисление косинуса значения.
<code>math.cosh(x)</code>	Вычислите гиперболический косинус значения, определенного как $\operatorname{cosh}(x) = 1/2 * (\exp(x) + \exp(-x))$
<code>math.cot(x)</code>	Вычислите котангенс значения
<code>math.coth(x)</code>	Вычислите гиперболический котангенс значения, определенного как $\operatorname{coth}(x) = 1 / \operatorname{tanh}(x)$
<code>math.csc(x)</code>	Вычислите косекант значения, определенного как $\operatorname{csc}(x) = 1/\sin(x)$
<code>math.csch(x)</code>	Вычислите гиперболический косекант значения, определенного как $\operatorname{csch}(x) = 1 / \operatorname{sinh}(x)$

Функция	Описание
math.sec(x)	Вычислите секущую часть значения, определенного как $\sec(x) = 1/\cos(x)$
math.sech(x)	Вычислите гиперболическую секущую значения, определенного как $\operatorname{sech}(x) = 1 / \cosh(x)$
math.sin(x)	Вычислите синус значения
math.sinh(x)	Вычислите гиперболический синус значения, определенного как $\sinh(x) = 1/2 * (\exp(x) - \exp(-x))$
math.tan(x)	Вычислите тангенс значения
math.tanh(x)	Вычислите гиперболический тангенс значения, определенного как $\tanh(x) = (\exp(2 * x) - 1) / (\exp(2 * x) + 1)$

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ

АИС	– автоматизированная информационная система
ВМ	– виртуальная машина
ОПО	– общее программное обеспечение
ОС	– операционная система
ПО	– программное обеспечение
СПО	– специальное программное обеспечение
СУБД	– система управления базами данных
ЭЦП	– электронная (цифровая) подпись
API	– от англ. Application Programming Interface – программный интерфейс взаимодействия
DOCX	– формат файлов для хранения электронных документов пакетов офисных приложений, который ориентирован, в первую очередь, на отображение данных в виде текста
OSRM	– от англ. Open Source Routing Machine – сервис построения маршрутов
PDF	– от англ. Portable Document Format – Межплатформенный формат электронных документов
XLSX	– формат файлов для хранения электронных документов пакетов офисных приложений, который ориентирован, в первую очередь, на отображение данных в табличном виде
URL	– от англ. Uniform Resource Locator – система унифицированных адресов электронных ресурсов

ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ

Авторизация	–	процедура проверки прав пользователя на выполнение определенных действий на Платформе, результатом которой становится разрешение или отказ в осуществлении запрашиваемых операций, а также предоставление в соответствии с ролевой моделью пользователю возможностей, которые гарантированы ему Платформой
Аналитическая панель (дашборд)	–	тип формы, в котором основные данные по объектам типа представления наглядно отображаются с помощью различных диаграмм, графиков и др. компонентов
Вычисляемый атрибут только для чтения	–	метод, используемый для вывода в формах каких-либо вычисляемых по определенной логике данных, которые нельзя изменить
Группировка	–	объединение данных по какому-либо разрезу с использованием функций аналитики
Динамический фильтр	–	метод, используемый для фильтрации данных в форме по какой-либо сложной логике (а не просто фильтрации данных по выбранному значению поля/полей).
Дополнительный атрибут (EAV-атрибут)	–	атрибут, который добавляется не в коде модели, а в Конструкторе
Информационная система	–	система, созданная на базе Платформы
Конструктор (админ-панель)	–	административная панель django – одна из составляющих Платформы. В конструкторе производится настройка личного кабинета
Лендинг	–	тип формы, для которого скрыты шапка и вapper стилей карточки
Личный кабинет пользователя	–	веб-интерфейс проекта, который создается на базе Платформы
Оптимизированный вычисляемый атрибут	–	метод, объединяющий плюсы вычисляемых атрибутов только для чтения и динамических фильтров
Переходы статусов	–	запись в таблице «Переходы» (DocStatusTransition), определяющий возможность перевода статуса объекта из одного в другой
Пользователь	–	учетная запись пользователя в системе (уникальный логин и пароль для входа)

- Пункт главного меню – запись в таблице «Главное меню» (NavItemMain). Каждый пункт меню определяет страницу, доступную в одном из кабинетов пользователей. Пункт меню должен быть обязательно привязан к Типу организации и группе ролей пользователей
- Разбиение – еще один разрез внутри существующего разреза, который в интерфейсе разделяет колонку функции аналитики.
- Разрезы (срезы) – атрибуты из дашборда или типа представления, по которым разделяются данные. Разрез может быть скаляром (число, строка и т.п., например, госномер автомобиля) или объектом.
- Разрешения – записи в различных таблицах прав (права на CRUD, Права на операции, права на переходы и тд), определяющие право на определенное действие или объект в информационной системе
- Роль пользователя – группа разрешений. Один пользователь может обладать несколькими ролями
- Состояние – значение атрибута объекта в определенный момент времени, хранимое во внешней таблице (таблице состояний), связанной с данным объектом
- Статус – запись в таблице «Статусы» (DocStatus). Статусы являются основой бизнес-процесса «Машина состояний», который реализуется и дорабатывается в рамках Платформы. Если у таблицы какого-либо типа представления есть свойство status с внешним ключом на таблицу «Статусы», то на Платформе можно настроить граф переходов состояний данного типа объектов. От статуса объекта могут зависеть права на операции с ним различных групп пользователей и форма отображения данных о нем
- Тип представления – запись в таблице типов представления (DocType). Характеризует любую группу объектов в информационной системе
- Форма – конфигурация внешнего вида определенной страницы на Платформе, запись в таблице «Формы» (FormItem). Может быть разного типа
- Форма операции – тип формы, который применяется для операций, требующих до запуска указания дополнительных параметров
- Форма списка – тип формы, в котором отображаются основные данные по объектам типа представления, на котором построена форма
- Форма экспорта – тип формы, в котором задается формат экспорта и импорта данных через XLS в форме списка

- Форма элемента (детальная форма) – тип формы, в котором отображаются подробные данные по определенному объекту типа представления, на котором построена форма. Переход в форму элемента осуществляется из формы списка по клику на объект
- Функции аналитики (поля аналитики, поля агрегации) – агрегированные числовые показатели объектов. Например, суммарное кол-во контейнеров, максимальный объем и т.п.
- Шаблон печатной формы – запись в таблице «Печатные формы» (DocTemplate) и привязанный к ней DOCX-шаблон, содержащий jinja-разметку. Печатная форма является шаблоном вывода информации об объекте какого-либо типа представления
- CSS-селектор – некоторая текстовая запись, которая позволяет однозначно определить элемент или группу элементов на HTML-странице

