

Comindware Business Application Platform

Полное руководство

© 2009-2019 Comindware. Все права защищены.

Программные продукты Comindware защищены одним или несколькими патентами. Другие патенты находятся на стадии рассмотрения.

Comindware, Comindware Project, Connectstep, Comindware Advantage, Comindware Task Management, Comindware Team Network, Comindware Process и Comindware Tracker являются зарегистрированными товарными знаками компании Comindware.

Все остальные упоминаемые товарные знаки являются зарегистрированными товарными знаками соответствующих владельцев.

Распространение данного руководства или измененных версий данного руководства, в том числе существенно измененных, запрещено без явного разрешения владельца авторских прав.

Распространение настоящих или переработанных материалов в виде печатного издания (книги) равно в коммерческих и некоммерческих целях запрещено без предварительного письменного разрешения их владельца.

ДОКУМЕНТАЦИЯ ПОСТАВЛЯЕТСЯ «КАК ЕСТЬ». ВСЕ ЯВНО ВЫРАЖЕННЫЕ ИЛИ ПОДРАЗУМЕВАЕМЫЕ ОБЯЗАТЕЛЬСТВА, ПОДТВЕРЖДЕНИЯ ИЛИ ГАРАНТИИ, В ТОМ ЧИСЛЕ СВЯЗАННЫЕ С ТОВАРНОСТЬЮ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ИЛИ ПРИГОДНОСТЬЮ ЕГО ДЛЯ ИСПОЛЬЗОВАНИЯ В ОПРЕДЕЛЕННЫХ ЦЕЛЯХ ИЛИ ГАРАНТИИ ОТСУТСТВИЯ НАРУШЕНИЙ ПРАВ ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ, НЕ ПРИЗНАЮТСЯ, ЗА ИСКЛЮЧЕНИЕМ СЛУЧАЯ, КОГДА ТАКОЙ ОТКАЗ ОТ ПРИЗНАНИЯ СЧИТАЕТСЯ НЕДЕЙСТВИТЕЛЬНЫМ ПО ЗАКОНУ.

Дата публикации: 17.07.2019

Оглавление

Добро пожаловать!	6
1. Приступая к работе	7
1.1 Кейс «Согласование договора»	7
1.1.1 Настройте модель данных решения	7
1.1.2 Настройте процесс согласования договора	12
2. Управление данными	18
2.1 Введение в управление данными	19
2.2 Создание шаблона записи	19
2.3 Настройка атрибутов	22
2.3.1 Добавление атрибутов	22
2.3.2 Выражения	31
2.3.3 Связи между шаблонами записей	32
2.3.4 Редактирование атрибутов	33
2.3.5 Архивирование и разархивирование атрибутов	34
2.4 Настройка форм	36
2.4.1 Добавление формы	36
2.4.2 Настройка формы	40
2.4.3 Удаление формы	83
2.4.4 Правила для формы	85
2.5 Операции	95
2.5.1 Добавление операции	96
2.5.2 Настройка операции	97
2.5.3 Удаление операции	107
2.6 Панель инструментов	108
2.6.1 Добавление панели инструментов	108
2.6.2 Настройка панели инструментов	110
2.6.3 Удаление панели инструментов	115
2.7 Списки	116
2.7.1 Добавление списка	117
2.7.2 Настройка списка в конструкторе списка	120
2.7.3 Удаление списка	126
2.7.4 Копирование списка	126
2.7.5 Работа со списком в пользовательском интерфейсе	127

2.8	Права доступа	133
2.8.1	Добавление роли	133
2.8.2	Настройка прав доступа в пользовательском интерфейсе	133
2.9	Шаблоны экспорта	135
2.9.1	Простой шаблон	135
2.9.2	Шаблон с использованием C#	140
2.10	Импорт данных записей	144
2.11	Страницы	147
3.	Управление процессами и кейсами	153
3.1	Введение в управление процессами	154
3.2	Создание шаблона процесса	154
3.3	Элементы процессной схемы	157
3.3.1	Задачи	158
3.3.2	События	164
3.3.3	Подпроцессы	176
3.3.4	Шлюзы	180
3.3.5	Другие элементы	183
3.4	Настройка бизнес-правил	184
3.5	Проверка и публикация схемы процесса	186
3.5.1	Типовые ошибки диаграммы процесса	187
3.6	Управление версиями схемы процесса	191
3.7	Настройка формы процесса	191
3.8	Межпроцессное взаимодействие	193
3.8.1	Отправка сообщения	193
3.8.2	Получение сообщения	196
3.9	Создание процесса	198
3.10	Настройка списков записей	201
3.11	Введение в управление кейсами	204
3.12	Создание и настройка шаблона кейса	204
3.13	Настройка доступа к процессам и кейсам	205
4.	Взаимодействие с внешними ресурсами	207
4.1	Основные сведения о взаимодействии с внешними ресурсами	208
4.2	Канал электронной почты	208
4.3	Веб-канал для приема запросов	212
4.4	Путь передачи электронной почты	214

4.5	Путь передачи веб-запросов	218
4.6	Типы сообщений	221
5.	Язык выражений Comindware Business Application Platform	223
5.1	Введение в язык выражений Comindware Business Application Platform	224
5.2	Основные сведения	225
5.2.1	Объекты	226
5.2.2	Идентификаторы и системные имена	226
5.2.3	Типы данных	227
5.2.4	Функции	227
5.2.5	Литералы	254
5.2.6	Операторы	255
5.2.7	Вызов связанных данных	258
5.3	Примеры использования	259
5.3.1	Задание вычисляемого атрибута	259
5.3.2	Задание фильтра	260
5.3.3	Настройка правил	261
5.3.4	Задание условия	263
5.3.5	Назначение исполнителя задачи	264
6.	Интеграция через API	266
6.1	Введение в Web API	267
6.2	RESTful Web Api (публичные методы)	267
6.3	System Core Api (системные методы)	269
6.3.1	Настройка связи «многие ко многим» в коллекции	270
6.4	Solution Api (пользовательские методы)	273
7.	Системное администрирование	276
7.1	Глобальные настройки	277
7.2	Глобальные функции	279
7.2.1	Введение в глобальные функции	280
7.2.2	Сигнатура глобальной функции	280
7.2.3	Определение глобальной функции	286
7.3	Переменные	288

Добро пожаловать!

Comindware Business Application Platform — это программный продукт, предназначенный для автоматизации управления бизнес-процессами (BPM-платформа) нового поколения. Благодаря уникальным разработкам компании **Comindware®**, платформа достигла качественно нового уровня удобства использования, надежности, эффективности и функциональности.

Бизнес-процессы, построенные в Comindware Business Application Platform, легко и просто масштабируются без каких-либо ограничений. Являясь универсальным инструментом, Comindware Business Application Platform обеспечивает автоматизацию, управление и оптимизацию бизнес-процессов любого уровня сложности и взаимосвязанности.

Возможности:

Полная поддержка стандарта управления бизнес-процессами BPMN 2.0.

Полностью настраиваемое управление процессами и задачами.

Полная автоматизация и мониторинг процессов.

Управление бизнес-данными любой степени сложности с помощью гибко настраиваемых записей и динамических форм ввода и отображения данных.

Управление неформализованными бизнес-процессами с использованием кейсов.

Комбинирование "на лету" данных, задач, кейсов и процессов.

Интеграции любой сложности через открытые Web API или через Enterprise Service Bus.

Интуитивный интерфейс. Настраиваемая бизнес-аналитика и построение печатных форм в формате MS Word и MS Excel.

Простая установка и поддержка продукта.

Настраиваемая рабочая среда. Системное администрирование в полном объеме и без усилий.

Легкий, но эффективный язык программирования, встроенный в платформу. Возможность кастомизации пользовательского интерфейса с использованием JavaScript.

1. Приступая к работе

Выполняя простые, но жизненные задачи из этой главы, вы узнаете, как работать с данными и процессами. Предложенный кейс поможет вам приобрести навыки, достаточные для управления данными и процессами любой степени сложности.

1.1 Кейс «Согласование договора»

Предположим, что вы представитель компании по осуществлению грузоперевозок. С помощью данного кейса в платформе Comindware Business Application Platform вы сможете легко настроить процесс согласования договора с контрагентами.

В этой главе:



[Данные](#) ⁽⁷⁾

Настройте модель данных.



[Процессы](#) ⁽¹²⁾

Настройте процесс согласования договора.

См. также:

[Управление данными](#) ⁽²⁾

[Управление процессами и кейсами](#) ⁽³⁾

[Взаимодействие с внешними ресурсами](#) ⁽⁴⁾

[Язык выражений Comindware Business Application Platform](#) ⁽⁵⁾

[Интеграция через API](#) ⁽⁶⁾

[Системное администрирование](#) ⁽⁷⁾

1.1.1 Настройте модель данных решения

В первую очередь необходимо настроить модель данных сущностей: Контрагенты и связанных с ними - Договоры. Необходимо, чтобы каждый Договор имел ссылку на Контрагента и Статус согласования, в Договоре должны присутствовать данные доставки. Согласованные и Несогласованные Договоры должны отображаться в разных списках.

Для того чтобы подготовить модель данных необходимо выполнить ряд простых действий:

1. Перейдите в раздел **Шаблоны**: для этого выберите на боковой панели «Конфигурации» -> «Настройки». Отобразится раздел **Администрирование** со списком доступных разделов платформы.

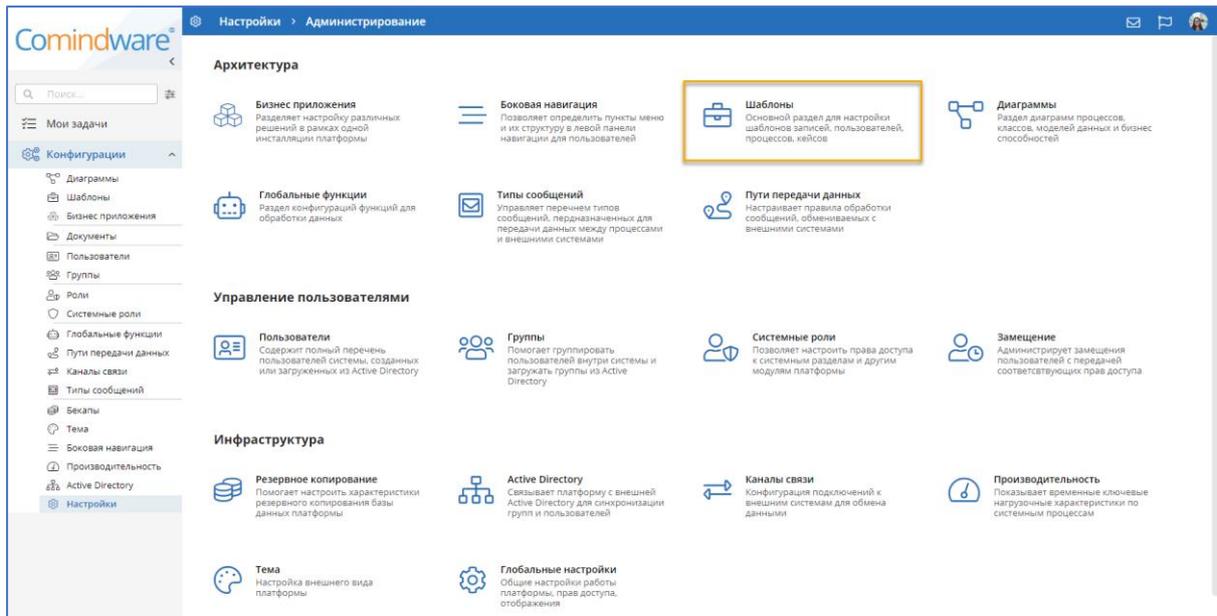


Рис. 1 Раздел «Администрирование» со списком разделов платформы

- Нажмите «+Добавить» и создайте по очереди 4 шаблона записи ⁽¹⁹⁾: «Контрагент», «Статус», «Адреса доставки» и «Договор».

Имя шаблона	ИД	Системное имя	Описание	Бизнес приложение	Тип шаблона	Статус	Дата создания	Создатель	Связи
Адреса доставки	oa.2	Adresadostavki		System business app	Запись	Активен	06.03.2019 17:01	Polina	
Договоры	oa.1	Dogovory		System business app	Запись	Активен	06.03.2019 16:53	Polina	
Контрагент	oa.4	Kontragent		System business app	Запись	Активен	06.03.2019 17:02	Polina	
Согласование договора	pa.1	Soglasovaniedogovora		System business app	Процесс	Активен	06.03.2019 17:02	Polina	
Статус	oa.3	Status		System business app	Запись	Активен	06.03.2019 17:02	Polina	

Рис. 2 Перечень шаблонов

Новый шаблон

Имя*

Системное имя*

Тип шаблона: Шаблон записи

Описание:

Отмена Создать

Рис. 3 Создание нового шаблона записи

3. Откройте шаблон двойным кликом и перейдите в раздел **Атрибуты**.

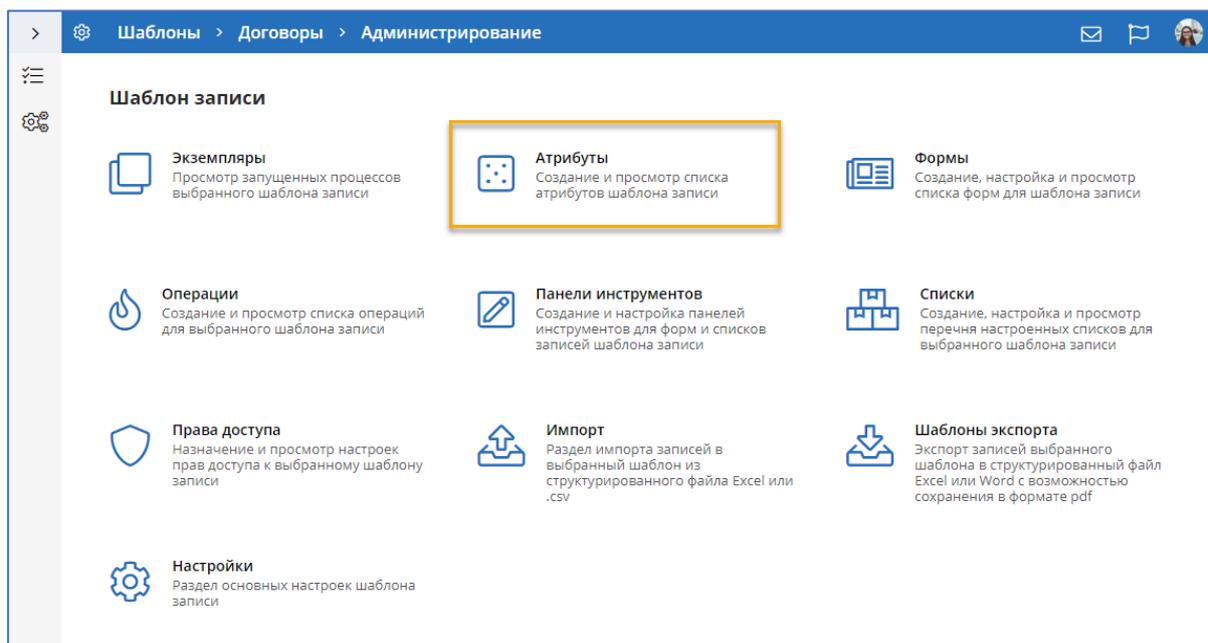


Рис. 4 Панель управления шаблоном записи

4. Создайте необходимые атрибуты (см. п.2.3) для шаблона записи «Договоры»:

- «Номер», тип данных - «Текстовый»;
- «Наименование», тип данных – «Текстовый». Пометьте атрибут отображаемым;
- «Дата», тип данных, тип данных - «Дата\Время»;
- «Статус, тип данных», тип данных - «ссылка», целевой шаблон – «Статус»;
- «Адрес доставки», тип данных - «ссылка», целевой шаблон – «Адреса доставки».

На рисунке ниже представлен пример создания атрибутов для шаблона «Договоры». Атрибуты, начинающиеся с «_», а также атрибут «id», являются системными и создаются автоматически.

Ид	Системное имя	Имя	Тип данных	Формат отображения	Вычисляемый	Отслеживание изменений	Доступно для поиска	Отображаемый атрибут	В архиве
ор.8	Adresadostavki	Адреса доставки	Коллекция	Не предусмотрен					
ор.5	Adresdostavki	Адрес доставки	Ссылка	Показывать как ссылка					
ор.3	Data	Дата	Дата / Время	4 сент. 1986 г.					
ор.6	Kontragent	Контрагент	Ссылка	Показывать как ссылка					
ор.1	Naimenovanie	Наименование	Текст	Обычный текст				✓	
ор.2	Nomer	Номер	Текст	Обычный текст					
ор.4	Status	Статус	Ссылка	Показывать как ссылка					

Рис. 5 Перечень атрибутов сущности «Договоры»

Атрибуты типа «Ссылка» позволяют установить связь между шаблонами записей (п.2.3.3).

Атрибут «Наименование» помечен «отображаемым», чтобы указать, какие данные будут отображаться в выпадающих списках при выборе значений из справочника (в данном случае – наименования).

5. Создайте необходимые атрибуты для шаблона записи «Адреса доставки»:
 - «Адрес», тип данных «Текстовый»;
 - «Получатель», тип данных», тип данных - «ссылка», целевой шаблон – «Контрагент». Отметьте флаг «Сделать связь взаимной» и задайте имя коллекции «Получатели».
6. Создайте необходимые атрибуты для шаблона записи «Статус»:
 - «Название», тип данных «Текстовый».

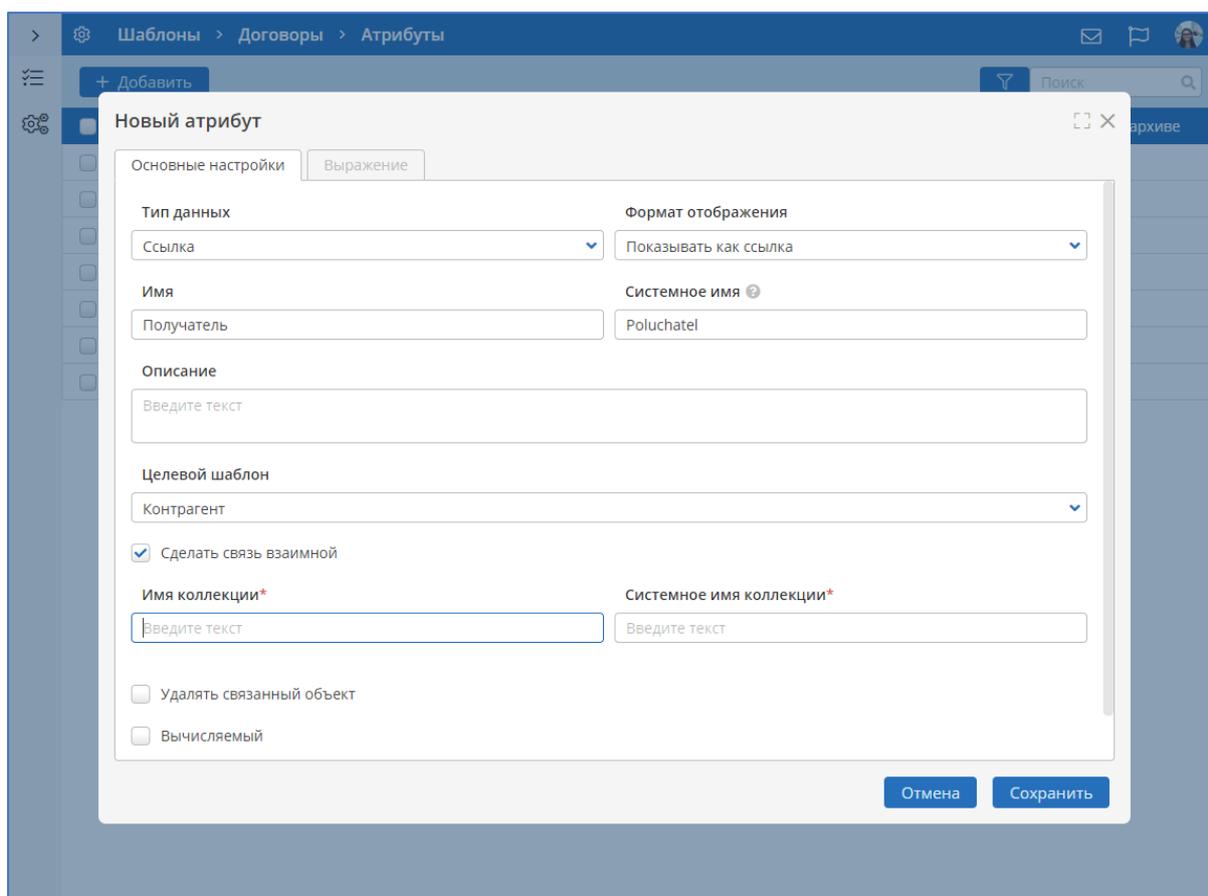


Рис. 6 Создание нового атрибута типа ссылка

7. Перейдите в раздел «**Формы**»: нажмите  на панели инструментов и выберите «Формы».

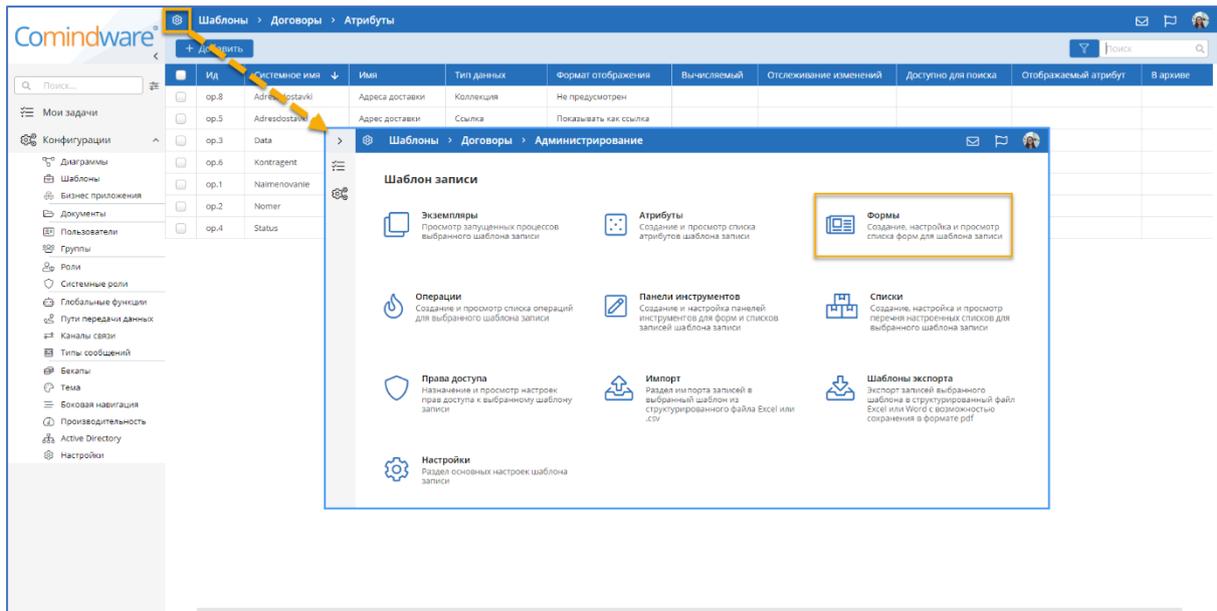


Рис. 7 Переход в раздел «Формы» из раздела «Атрибуты»

8. Создайте формы шаблонов записей.³⁶

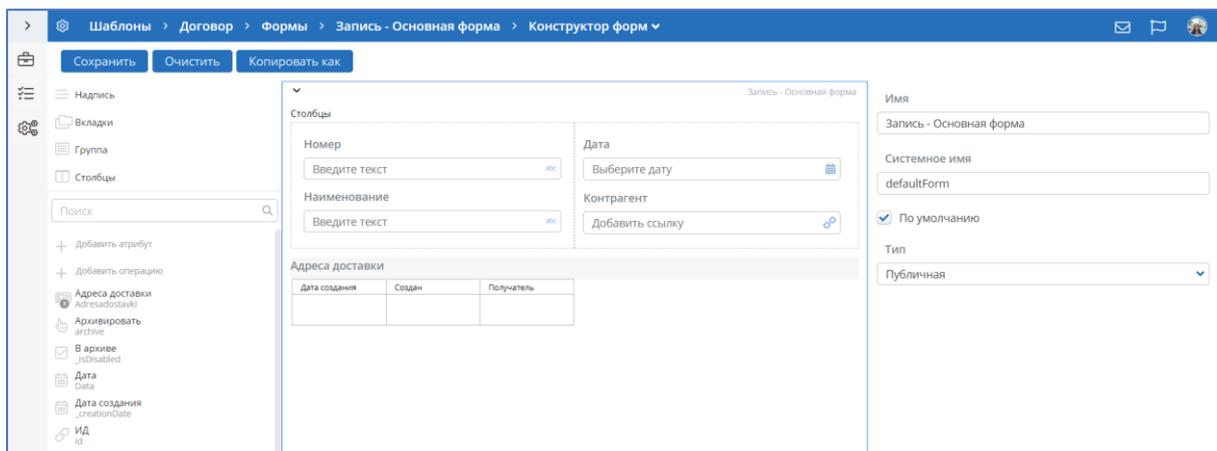


Рис. 8 Конструктор форм

9. Перейдите в раздел «**Списки**» (подробная настройка списков описана в п. 2.7). Настройте списки записей в шаблоне записей **Договоры**, например: «подписаны», «на рассмотрении» и «согласованы». Списки разделяются по статусу и настраиваются в поле системного фильтра с помощью языка запросов RDF N3 или непосредственно на странице.

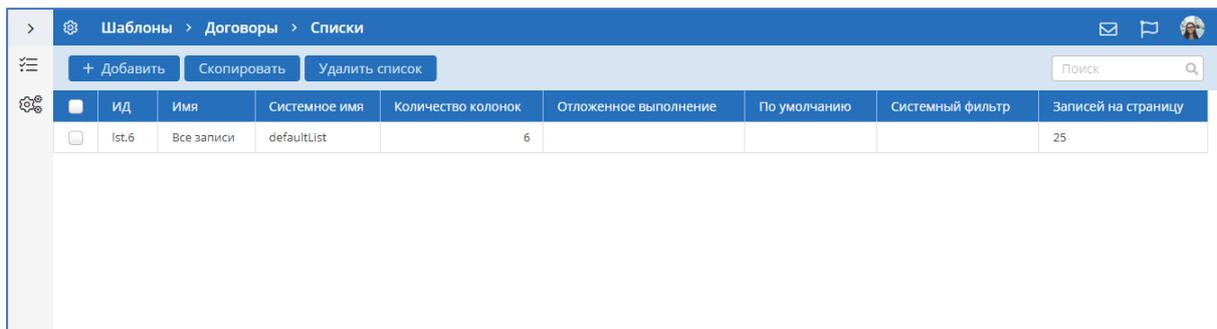


Рис. 9 Пользовательские списки

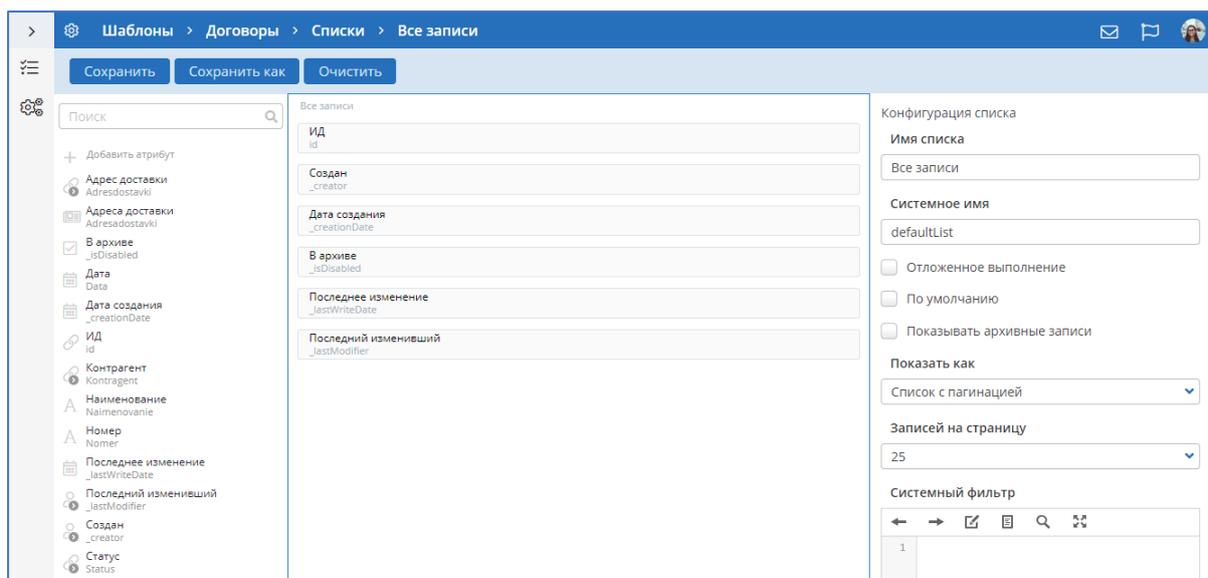


Рис. 10 Конфигурация списка

1.1.2 Настройте процесс согласования договора

После настройки модели данных следующим этапом работ необходимо настроить процесс согласования договора. Для настройки процесса согласования необходимо выполнить следующий набор простых действий:

1. На боковой панели навигации откройте раздел «Конфигурации» -> «Шаблоны».

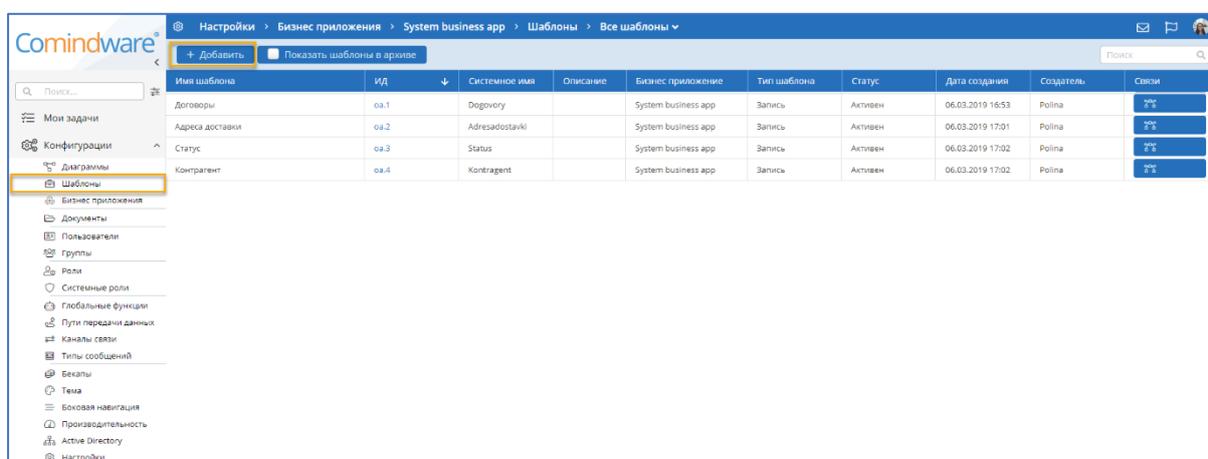


Рис. 11 Переход в раздел «Шаблоны» через боковую панель навигации

2. Нажмите «+Добавить» и создайте шаблон процесса (подробную настройку процесса см. в п. 3.2)

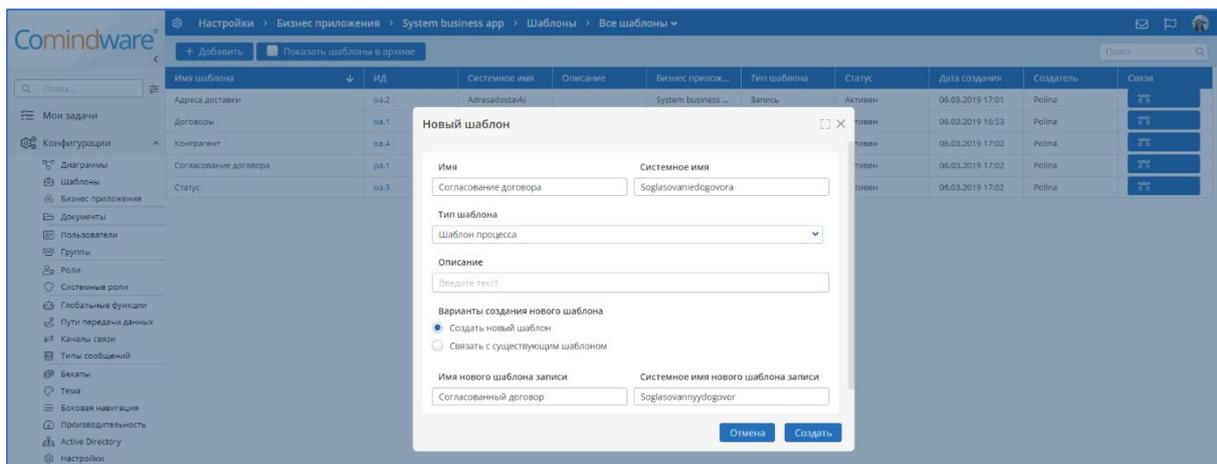


Рис. 12 Создание нового шаблона процесса

Существует 2 способа создать шаблон процесса: с привязкой к существующему шаблону записи и с созданием нового шаблона записи. Мы создаем с созданием нового шаблона записи.

3. Откройте созданный шаблон. Перейдите в раздел «Схема»

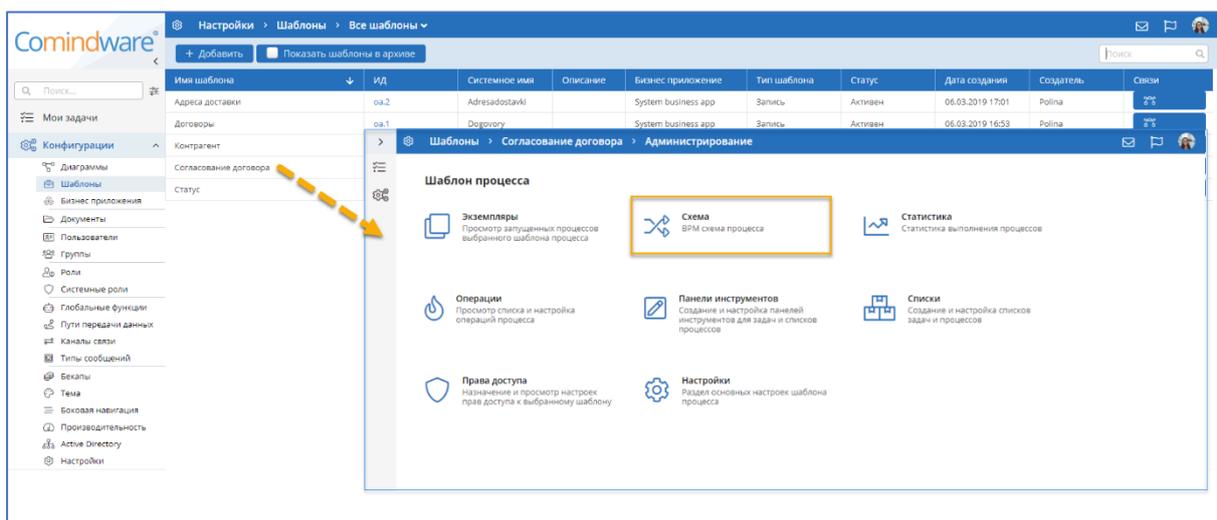


Рис. 13 Переход в раздел «Схема» выбранного шаблона процесса

4. Создайте схему процесса с помощью перемещения нужных элементов с панели элементов на рабочую область (подробное описание создания схемы процесса см. в главе 3).

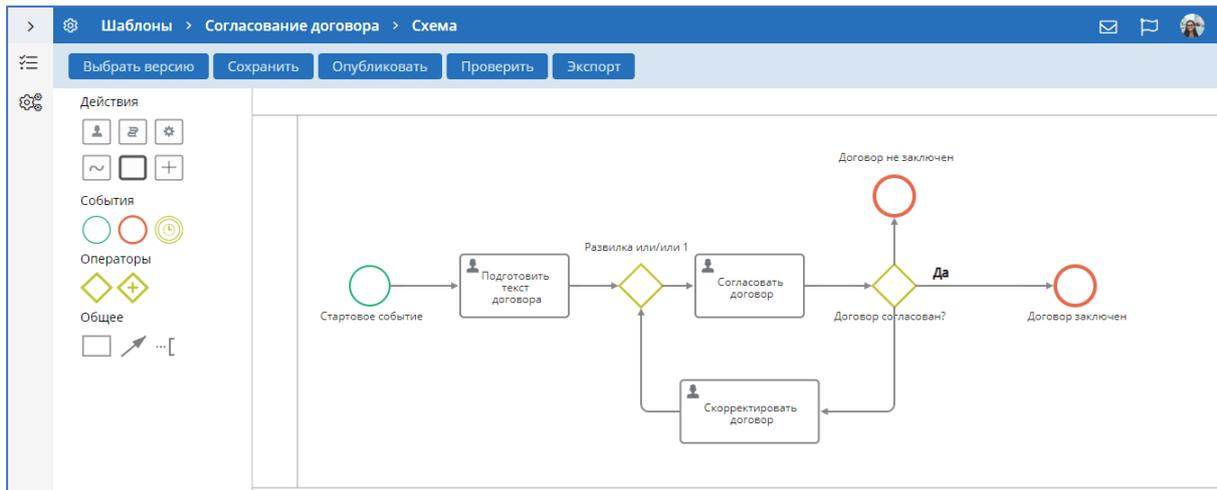


Рис. 14 Схема процесса

5. В настройках пользовательских задач выберите исполнителя и наименование задач. Для этого кликните по пользовательской задаче и выберите «Настройки». Во вкладке «Дополнительные свойства» выберите исполнителя.

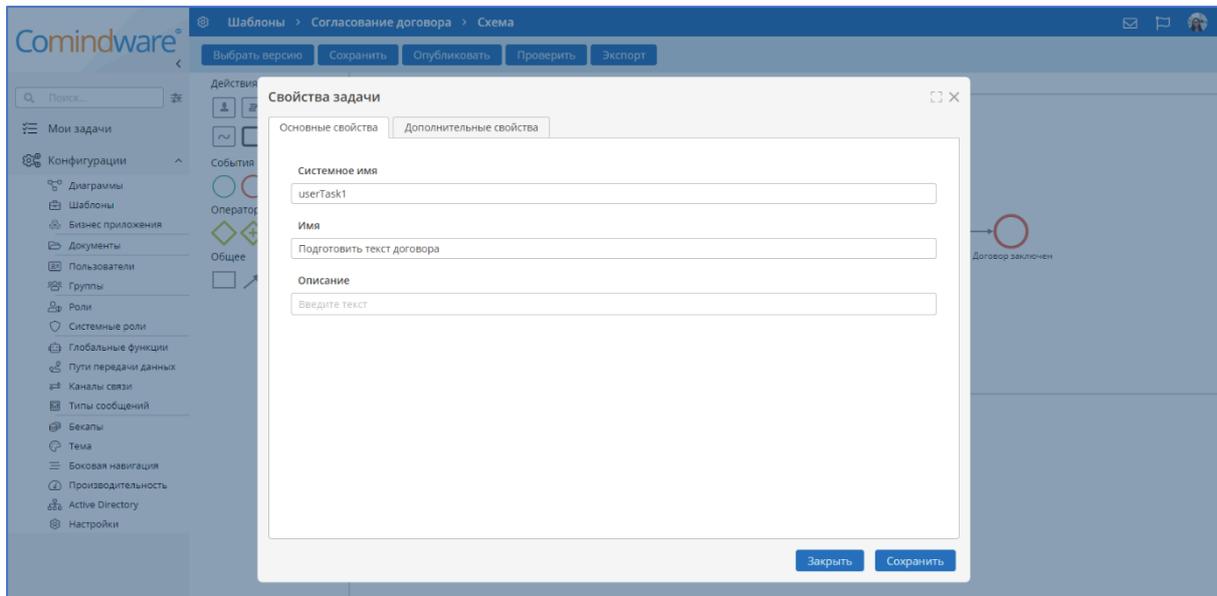


Рис. 15 Настройка пользовательской задачи (наименование)

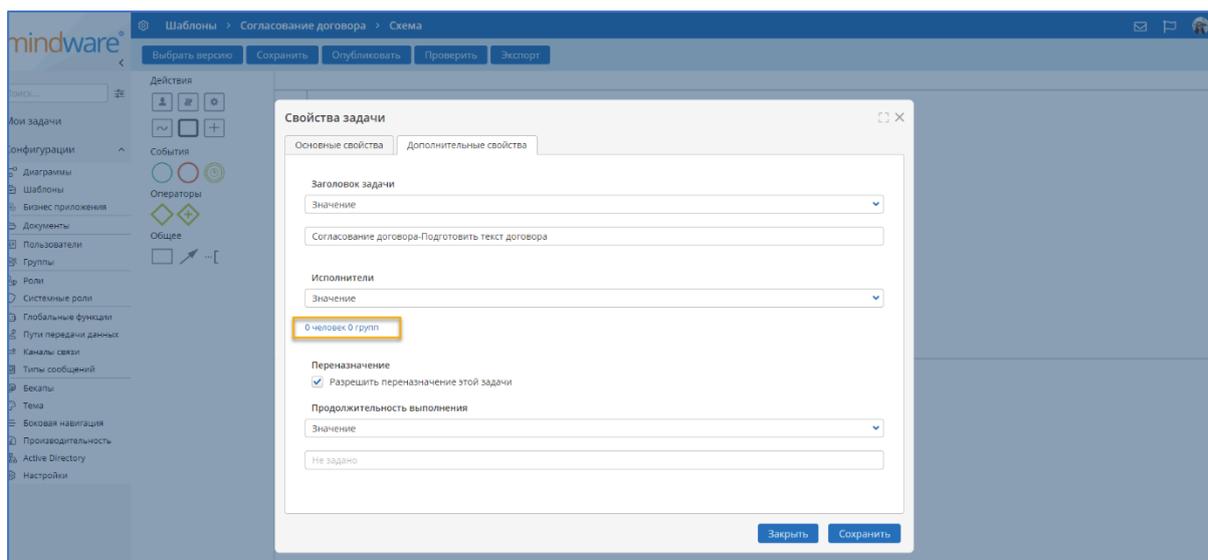


Рис. 16 Настройка пользовательской задачи (назначение исполнителя)

6. Создайте форму задачи.

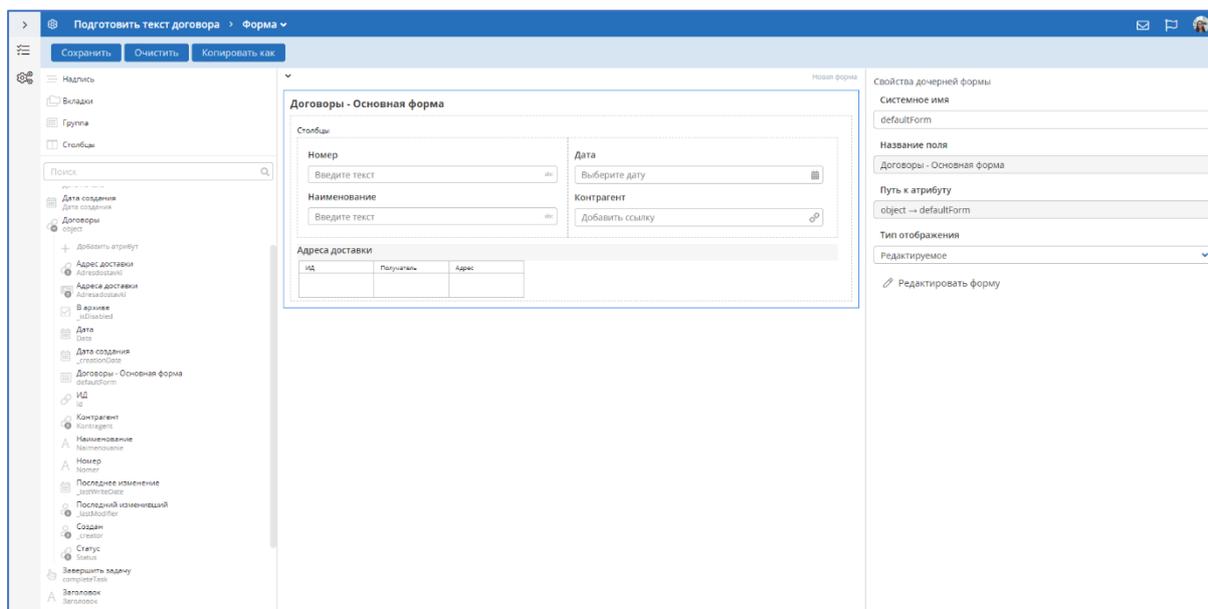


Рис. 17 Конструктор форм пользовательской задачи

Форму можно скопировать с одной задачи на другую путем нажатия на кнопку «Копировать как».

7. Настройте правила для данных. Создайте правило изменения статуса после окончания выполнения задачи. Для этого создайте правила для данных задач на выходе. Например, создайте правило на выходе развилки после задачи «Согласование договора»:

- если заявка согласована, то статус процесса меняется на «Согласован»,
- если заявка отправлена на исправление, то статус процесса меняется на «На рассмотрении».

Flow2 – системное имя потока «Договор согласован».

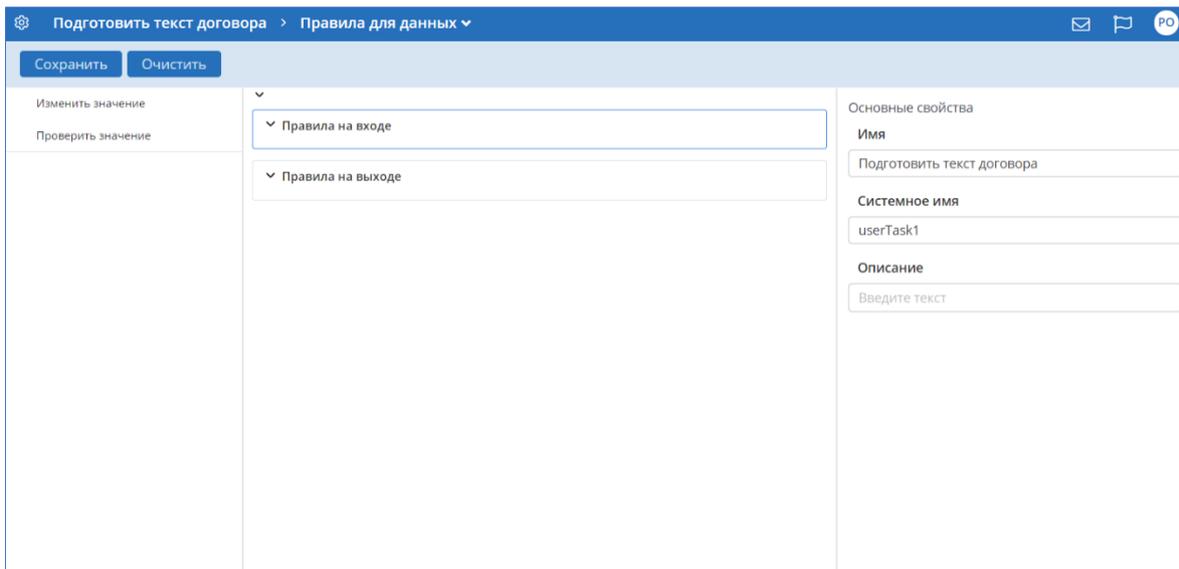


Рис. 18 Создание правил для данных

Проверьте схему и при отсутствии ошибок опубликуйте.

На форме шаблона записи ³⁶ создайте операцию «Регистрация договора» ⁸⁷. Для этого перейдите в раздел «Формы» выбранного шаблона записи и на конструкторе форм нажмите кнопку «Добавить операцию» (см. п. 2).

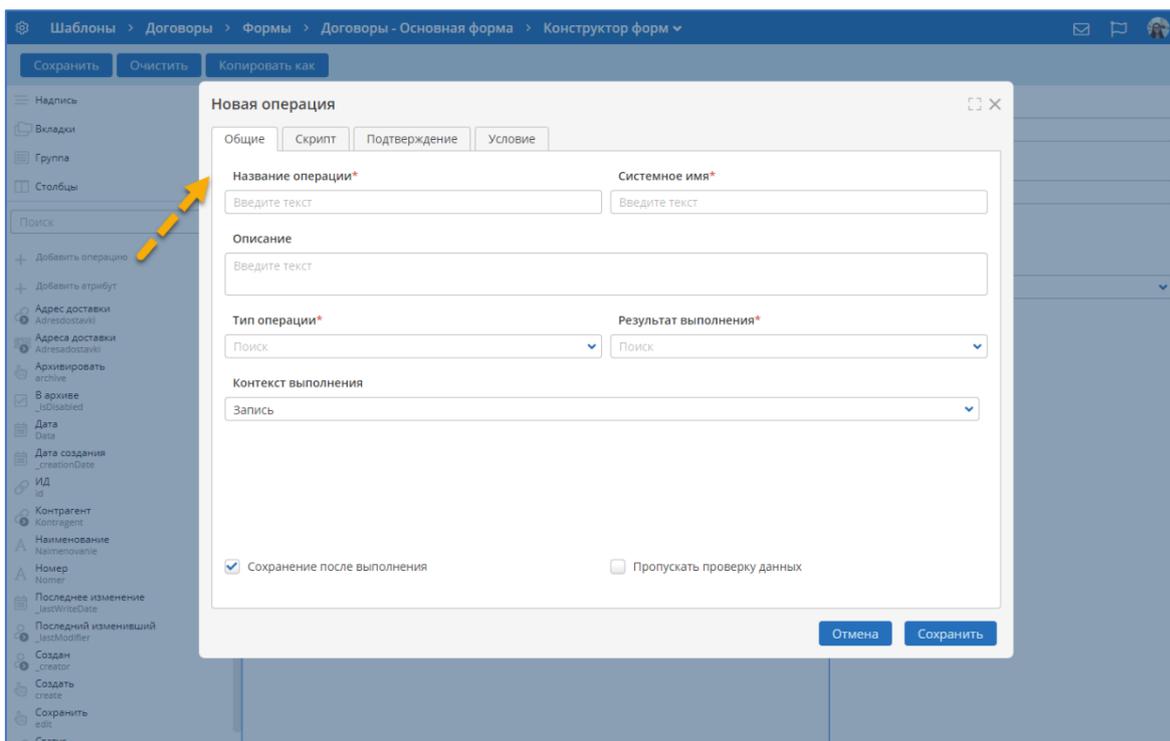


Рис. 19 Создание операции

Запустите созданный процесс. Пользователю, на которого назначена задача, приходит уведомление о назначении задачи. После исполнения своей задачи, пользователь завершает задачу и управление процессом передается следующему шагу процесса.

Согласование договора-Подготовить текст договора > Карточка задачи

Сохранить Завершить задачу

Номер	Дата
1	06.07.2018
Наименование	Контрагент
договор на поставку услуг	Введите имя...

Адреса доставки

+ ✕ Поиск

--	--	--	--

Текст договора

📎

- 2018-04-30.pdf

Рис. 20 Задача пользователя

2. Управление данными

В этой главе:



[2.1 Введение в управление данными](#)

Основы управления данными: запись и шаблон записи.



[2.3 Настройка атрибутов](#)

Присвоение шаблону записи набора атрибутов.



[2.8 Права доступа](#)

Настройка доступа пользователей к шаблонам записи и записям.



[2.6 Панель инструментов](#)

Настройка отображения операций на формах и/или списках.



[2.9 Шаблоны экспорта](#)

Общая инструкция о том, как создать и настроить шаблон документа.



[2.2 Создание шаблона записи](#)

Общая инструкция о том, как создать и настроить шаблон записи.



[2.4 Настройка форм](#)

Общая инструкция о том, как создать и настроить шаблон записи.



[0.9. Если выражение возвращает «false», то правило не запускает выполнение действий, добавленных в этот раздел](#)



[2.7 Списки](#)

Создание и настройка списков записей.



[2.10 Импорт данных записей](#)

Информация о том, как импортировать и экспортировать данные записей.

2.1 Введение в управление данными

Comindware Business Application Platform предоставляет инструменты для сбора, анализа и управления бизнес-данными любой степени сложности.

Платформа позволяет осуществлять моделирование данных в виде **Шаблонов записей** и хранить их в виде **Записей**.

- **Шаблон записи** представляет собой сущность, содержащую набор инструментов для настройки метаданных шаблона записи. Все данные, вводимые пользователями в платформу, привязываются к определенному шаблону записи.
- **Запись** представляет собой набор связанных данных.

Следующий наглядный пример поможет вам разобраться, как работает данная схема:

Параметр	Описание
Шаблон записи	«Пассажиры рейсов»
Атрибуты шаблона записи	«ФИО», «Номер паспорта», «Номер билета», «Рейс»
Работа с записями	Сотрудник авиакомпании вводит значения атрибутов шаблона записи в предустановленной форме для каждого пассажира во время регистрации на рейс
Данные на выходе	Список пассажиров рейсов с их именами, номерами паспортов и билетов.

2.2 Создание шаблона записи

Для того чтобы создать шаблон записи необходимо:

1. Перейти в раздел **Шаблоны**. Для этого кликните по иконке пользователя в правом верхнем углу и выберите пункт **Глобальные настройки**, затем перейдите в раздел **Шаблоны**.

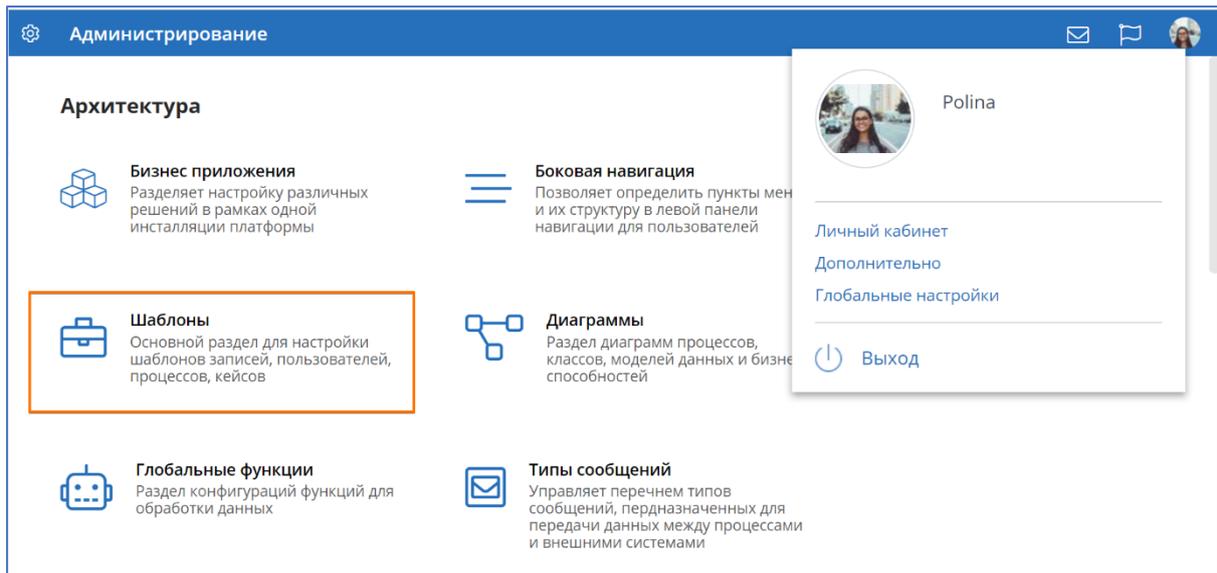


Рис. 21 Переход к списку шаблонов

Имя шаблона	ИД	Системное имя	Описание	Бизнес приложение	Тип шаблона	Статус	Дата создания	Создатель	Связи
ТестКоллекции	oa.14	TestKollektcii		System business app	Запись	Активен	20.03.2019 16:...	Polina	
Коллекции	oa.11	Kollektcii		System business app	Запись	Активен	19.03.2019 10:...	Polina	
Контрагент	oa.4	Kontragent123		System business app	Запись	Активен	06.03.2019 17:...	Polina	
Статус	oa.3	Status		System business app	Запись	Активен	06.03.2019 17:...	Polina	
Адреса доставки	oa.2	Adresadostavki		System business app	Запись	Активен	06.03.2019 17:...	Polina	
Договоры	oa.1	Dogovory		System business app	Запись	Активен	06.03.2019 16:...	Polina	
Согласование договора	pa.1	SoglasovanieDogovora		System business app	Процесс	Активен	06.03.2019 17:...	Polina	

Рис. 22 Перечень шаблонов записей

2. Нажать на кнопку «+Добавить».

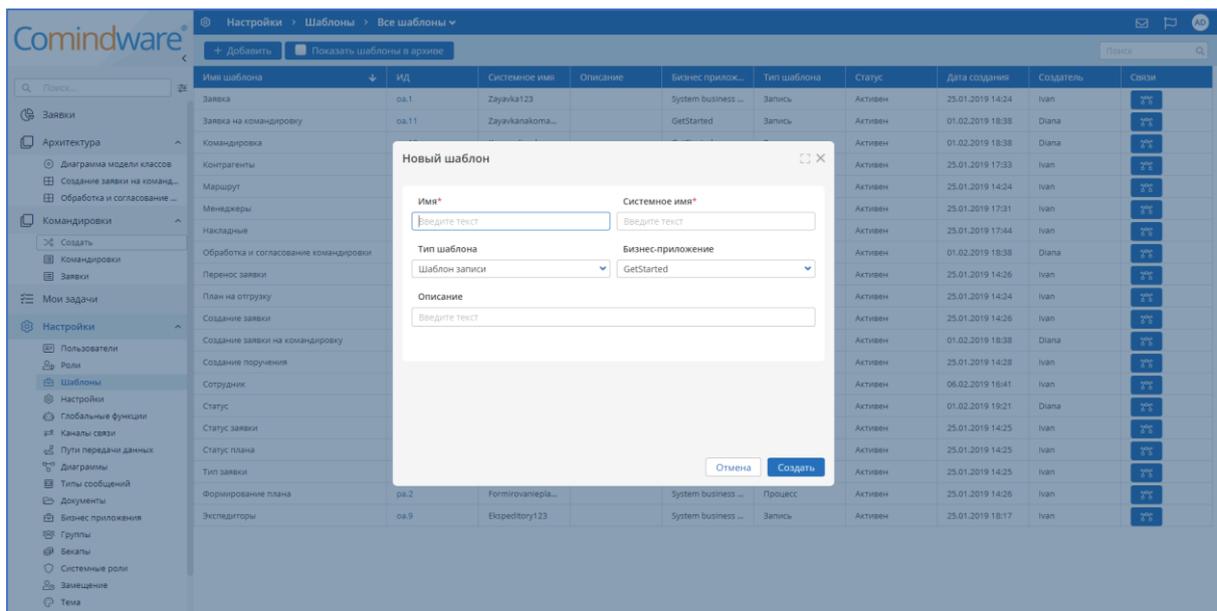


Рис. 23 Создание нового шаблона записи

При настройке шаблона записи доступны следующие разделы (Рис. 24):

- **Экземпляры** – переход в пользовательский интерфейс работы с записями (Рис. 156).
- **Атрибуты** – настройка атрибутов шаблона записи (2.3).
- **Операции** – настройка операций или пользовательских функций по обработке данных (0.7).
- **Формы** – настройка форм ввода и редактирования данных (2.4).
- **Панели инструментов** – настройка панелей отображения операций (2.6).
- **Списки** – настройка списков отображения записей (2.7).
- **Права доступа** – назначение и просмотр настроек прав доступа к выбранному шаблону (2.8).
- **Шаблоны экспорта** – настройка шаблонов печатных форм и отчетов (2.9).
- **Импорт** – позволяет осуществить первичную загрузку данных и файлов (2.10).
- **Настройки** – настройка основных параметров шаблона записи (Рис. 25).

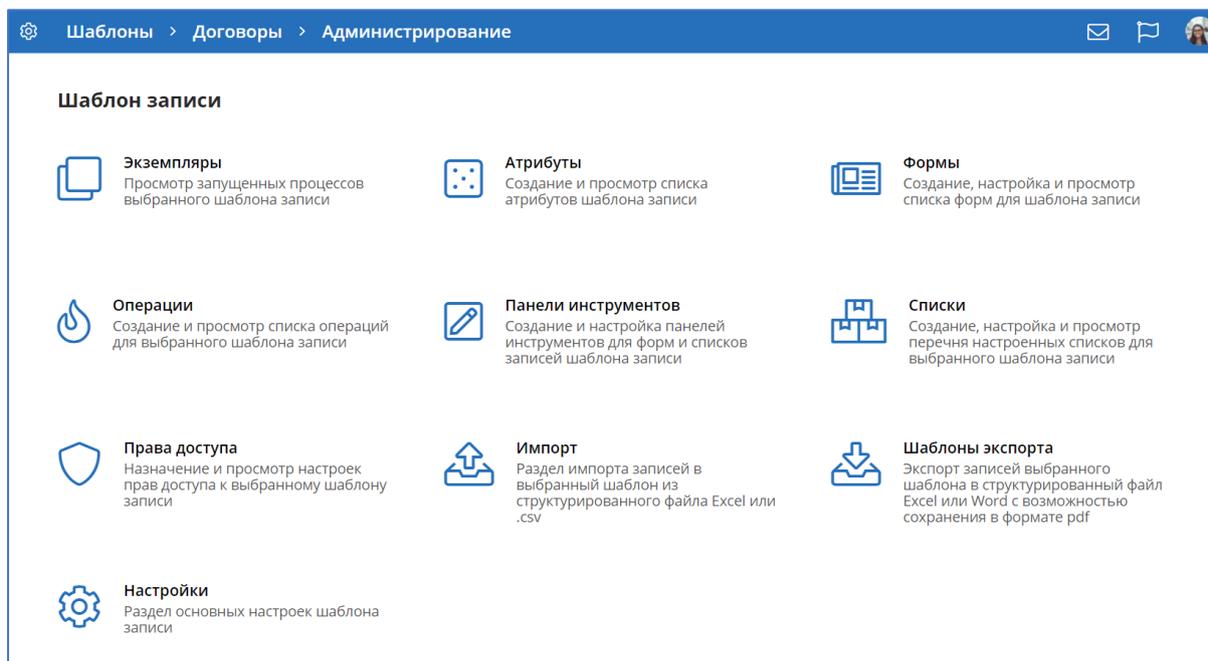


Рис. 24 Перечень разделов настройки шаблона записи

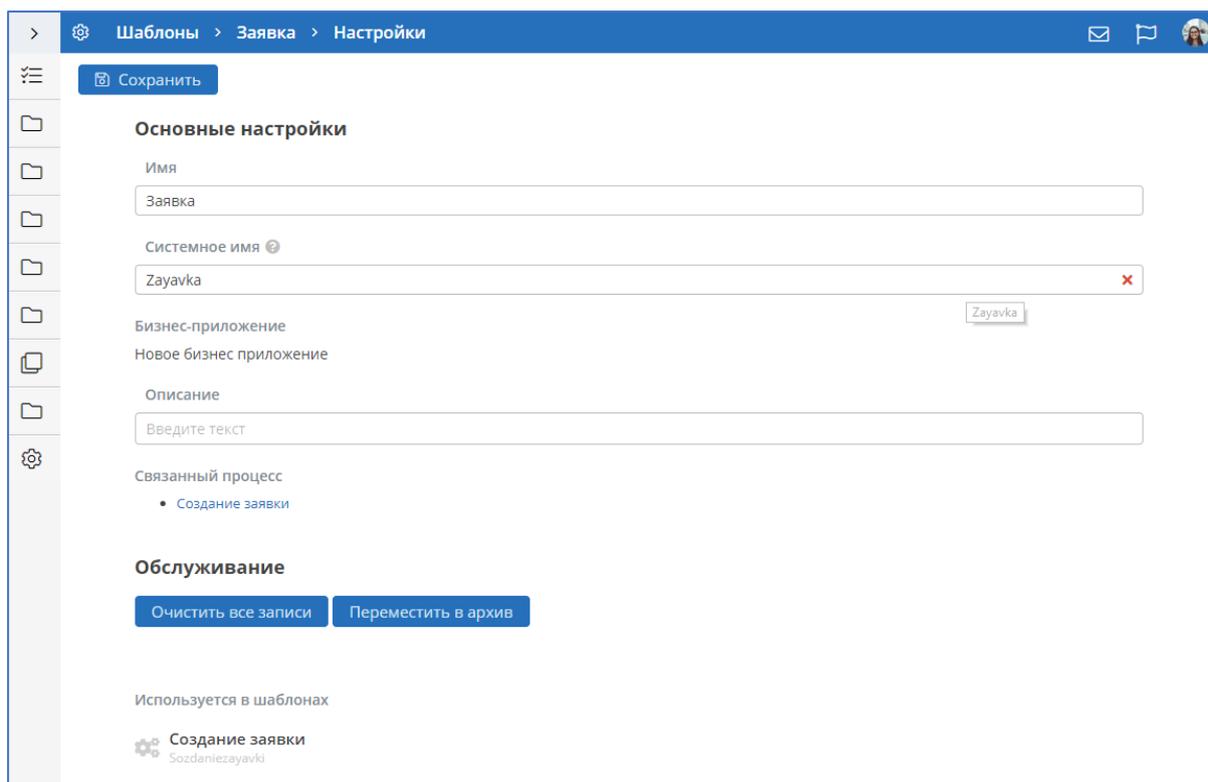


Рис. 25 Основные настройки шаблона записи

2.3 Настройка атрибутов

2.3.1 Добавление атрибутов

Для настройки атрибутов шаблона записи необходимо:

1. Перейти в раздел **Атрибуты** выбранного шаблона записи. Для этого кликните по иконке пользователя в правом верхнем углу, выберите «Глобальные настройки», перейдите в раздел **Шаблоны**, выберите нужный шаблон и перейдите в раздел **Атрибуты** (Рис. 21).

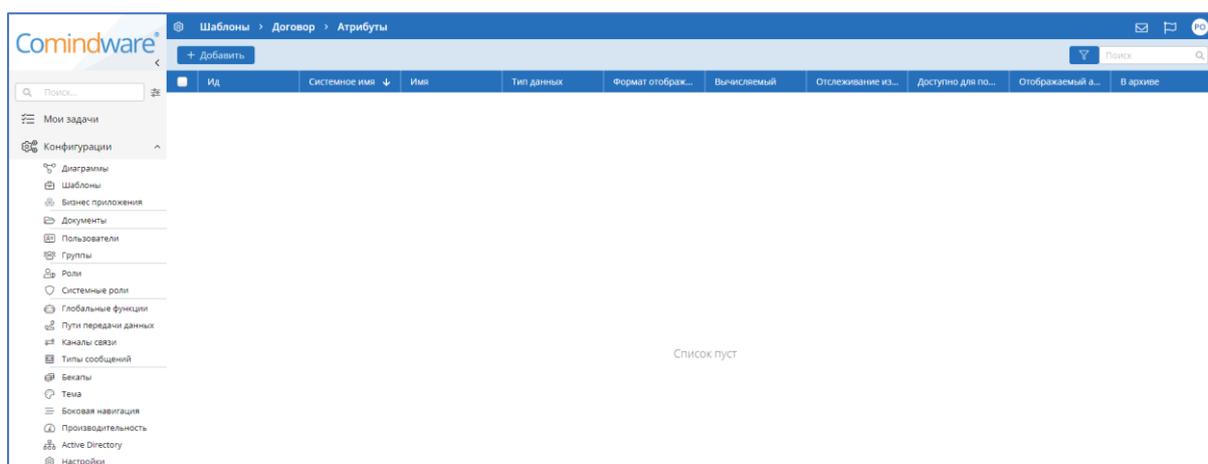


Рис. 26 Раздел **Атрибуты** выбранного шаблона записи

В списке атрибутов будут отображаться создаваемые атрибуты шаблона записи.

Новый шаблон записи по умолчанию содержит следующие системные атрибуты:

- ИД (id) – автоматически заполняется уникальный идентификатор записи.
- В архиве (_isDisabled) – флаг нахождения записи в архиве. По умолчанию при создании записи заполняется значение false.
- Создан (_creator) – автоматически заполняется пользователь, создавший запись.
- Дата создания (_creationDate) – автоматически заполняется дата и время создания записи.
- Последний изменивший (_lastModifier) – автоматически заполняется пользователь, последний изменивший любые данные записи.
- Последнее изменение (_lastWriteDate) – автоматически заполняется дата и время последнего изменения любых данных записи.

Настройки системных атрибутов не подлежат изменению пользователем. Изначально системные атрибуты не отображаются в списке атрибутов. Для их отображения нужно настроить видимость, нажав на кнопку в правом верхнем углу рядом с полем Поиска (Рис. 27). Там же можно настроить видимость архивных атрибутов (Архивирование и разархивирование атрибутов³⁴).

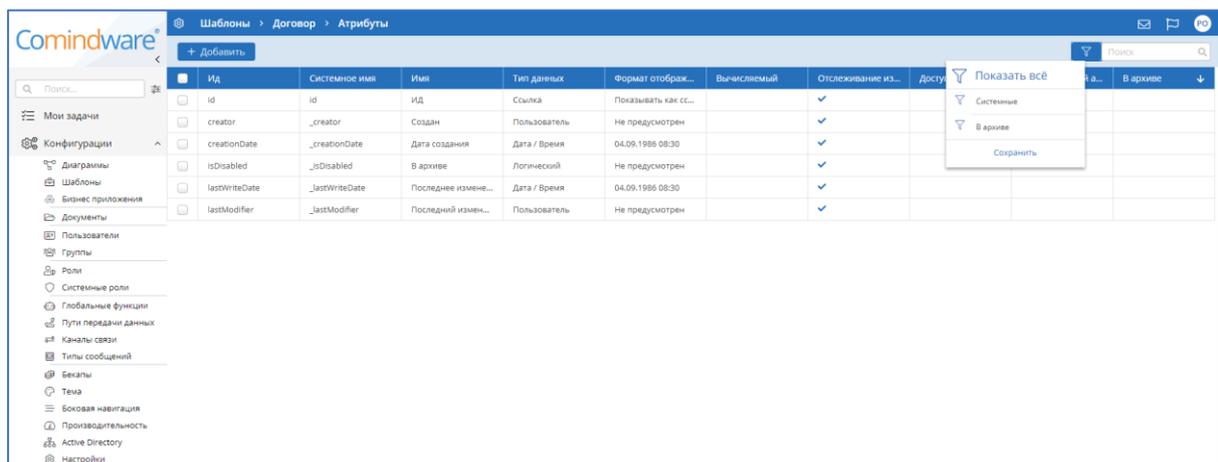


Рис. 27 Кнопка для настройки отображения системных и архивных атрибутов.

2. Для добавления нового атрибута необходимо нажать на кнопку «+Добавить».
3. В открывшейся форме «Новый атрибут» присутствуют две вкладки «Основные настройки» и «Выражение».

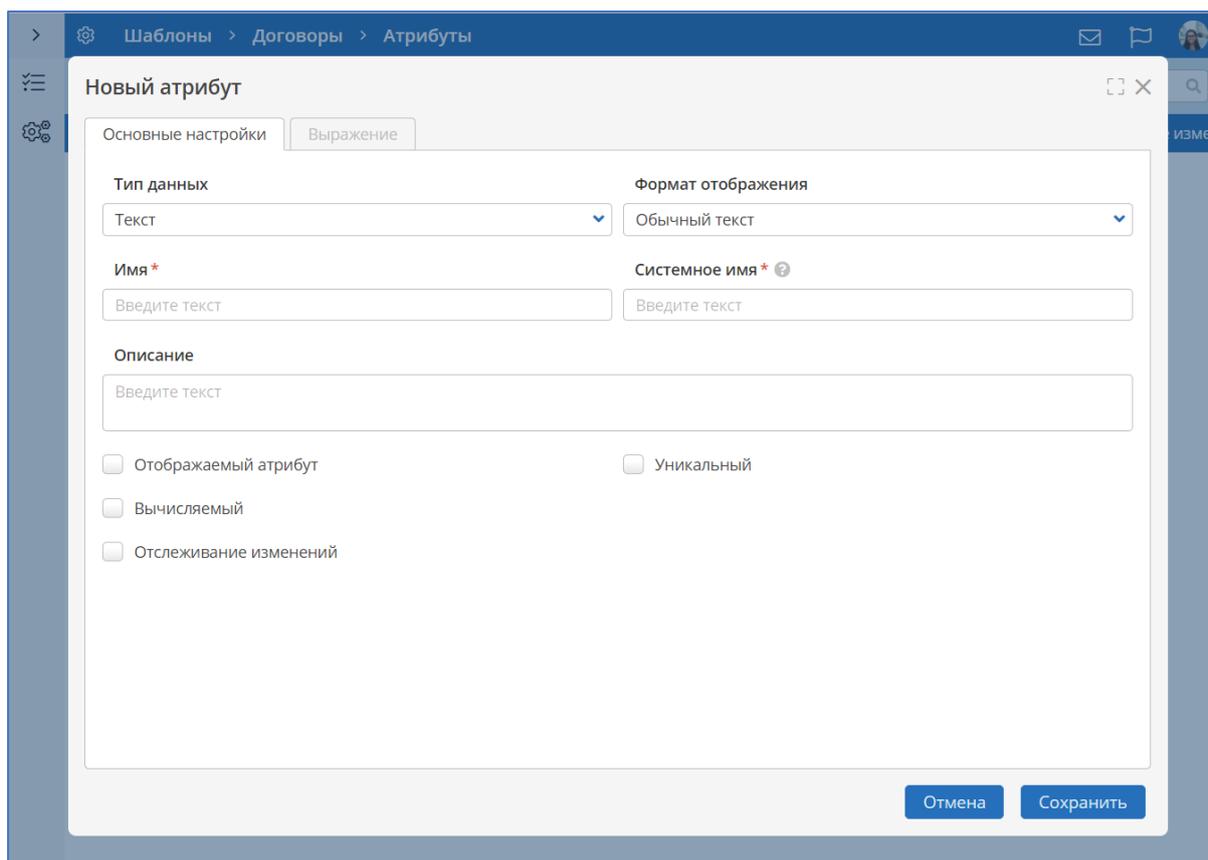


Рис. 28 Создание нового атрибута - вкладка Основные настройки.

Вкладка «Основные настройки» содержит настройки параметров атрибута. Вкладка «Выражение» становится доступна после того, как будет отмечен флаг «Вычисляемый». На вкладке «Выражение» вводится формула автоматического расчета для вычисляемого атрибута. Данные вычисляемого атрибута не сохраняются в базе данных.

Платформа поддерживает работу с атрибутами, имеющими следующие типы данных:

- Текст – атрибут может содержать любые текстовые данные. Для типа данных **Текст** доступна следующая детализация:
 - Формат отображения (необязательное поле): **Обычный текст** – сохраненные данные в атрибуте отображаются в платформе без форматирования, **HTML текст** – сохраненные данные в атрибуте отображаются в виде текста с форматированием.

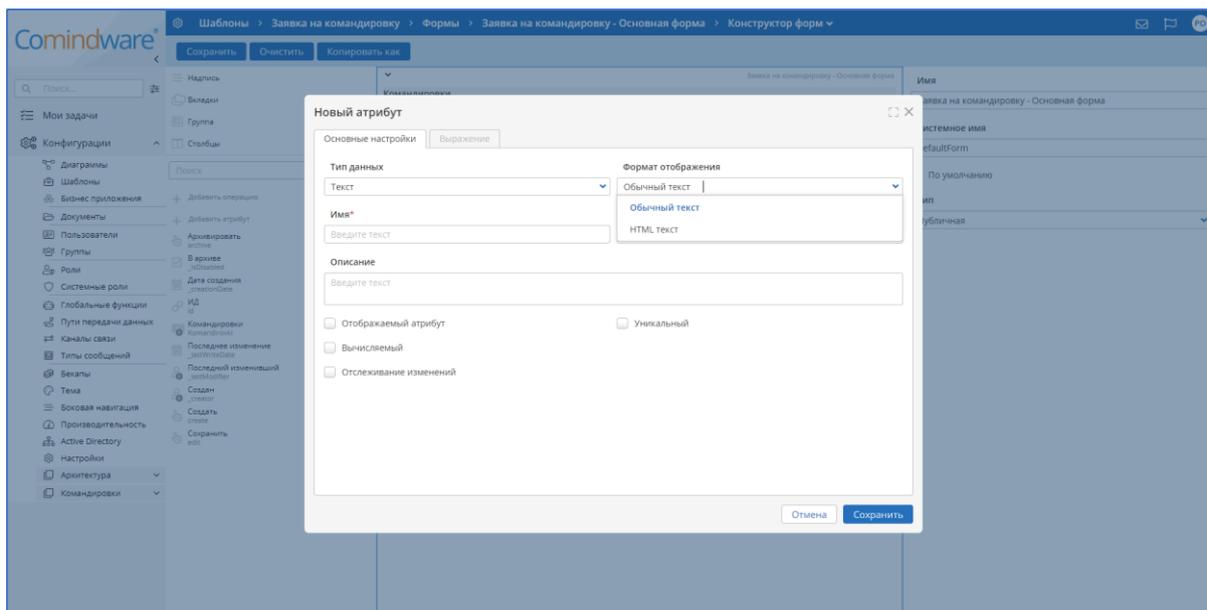


Рис. 29 Создание текстового атрибута

- Число – атрибут содержит любые числовые значения. Для типа данных **Число** доступна следующая детализация:
 - Формат отображения (необязательное поле): **Целое** – сохраненные данные в атрибуте отображаются в платформе в виде целого числа, **Десятичное** – сохраненные данные в атрибуте отображаются в виде числа с плавающей запятой, **Валюта** – сохраненное число в атрибуте отображается в виде числа с разделителем разрядов и двумя знаками после запятой.

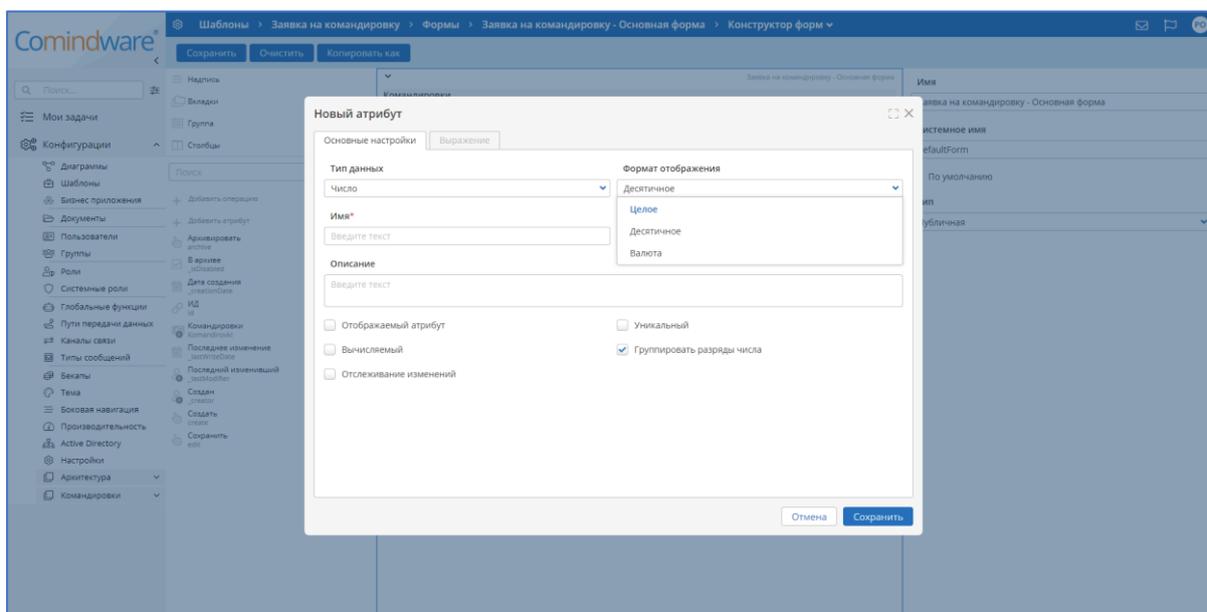


Рис. 30 Создание атрибута типа число

- Длительность – атрибут содержит данные о длительности времени. Для типа данных **Число** доступна следующая детализация:
 - Формат отображения (необязательное поле): **Дней**, **Часов**, **Минут**, **Секунд** – сохраненные данные в атрибуте отображаются в платформе в следующем

виде «0 д 0 ч 0 м 0 с», **Часов, Минут, Секунд** – сохраненные данные в атрибуте отображаются в платформе в следующем виде «0 ч 0 м 0 с», **Часов, Минут** – сохраненные данные в атрибуте отображаются в платформе в следующем виде «0 ч 0 м».

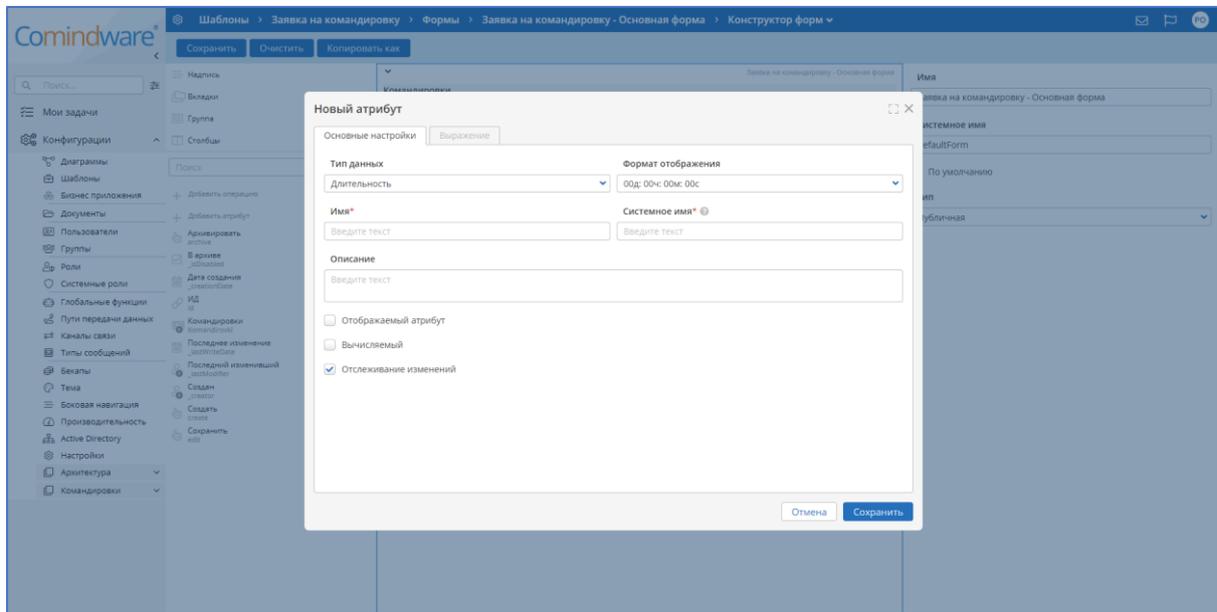


Рис. 31 Создание атрибута типа длительность

- **Дата / время** – атрибут содержит данные о дате и времени. Для типа данных **Дата / время** доступна следующая детализация:
 - **Формат отображения** (необязательное поле): указывается один из форматов отображения сохраненных данных в платформе. Форматы отображения изображены в виде примеров отображаемых дат.

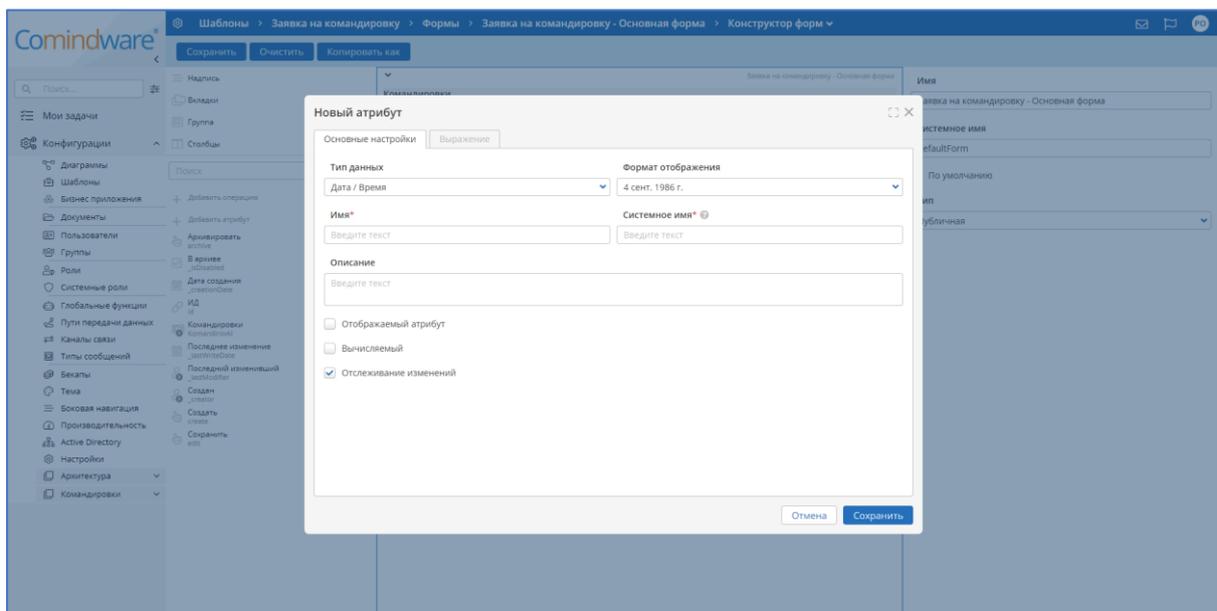


Рис. 32 Создание атрибута типа дата / время

- **Логический** – атрибут-флаг содержит значения true или false.

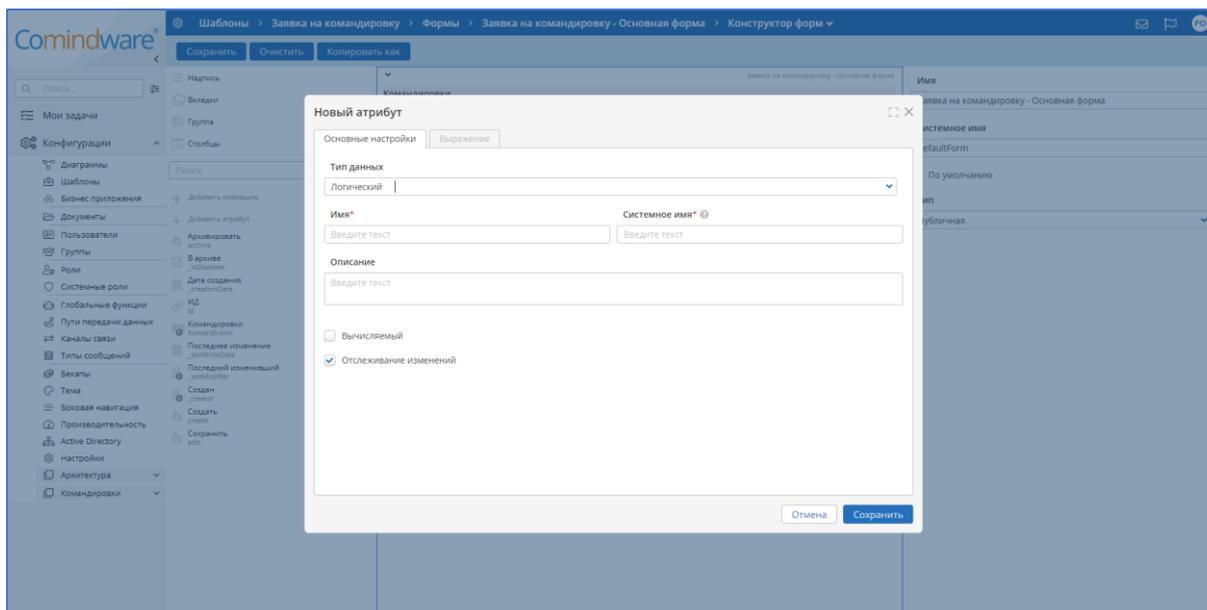


Рис. 33 Создание атрибута типа логический

- Документ – атрибут, позволяющий в качестве данных прикладывать и сохранять файлы пользователя. Для типа данных **Документ** доступна следующая детализация:
 - Фильтр расширений файлов (необязательное поле): ограничивает перечень расширений, которые возможно будет сохранить в атрибуте. Если не выбрано ни одного расширения, то к атрибуту можно будет прикладывать файлы любого расширения.

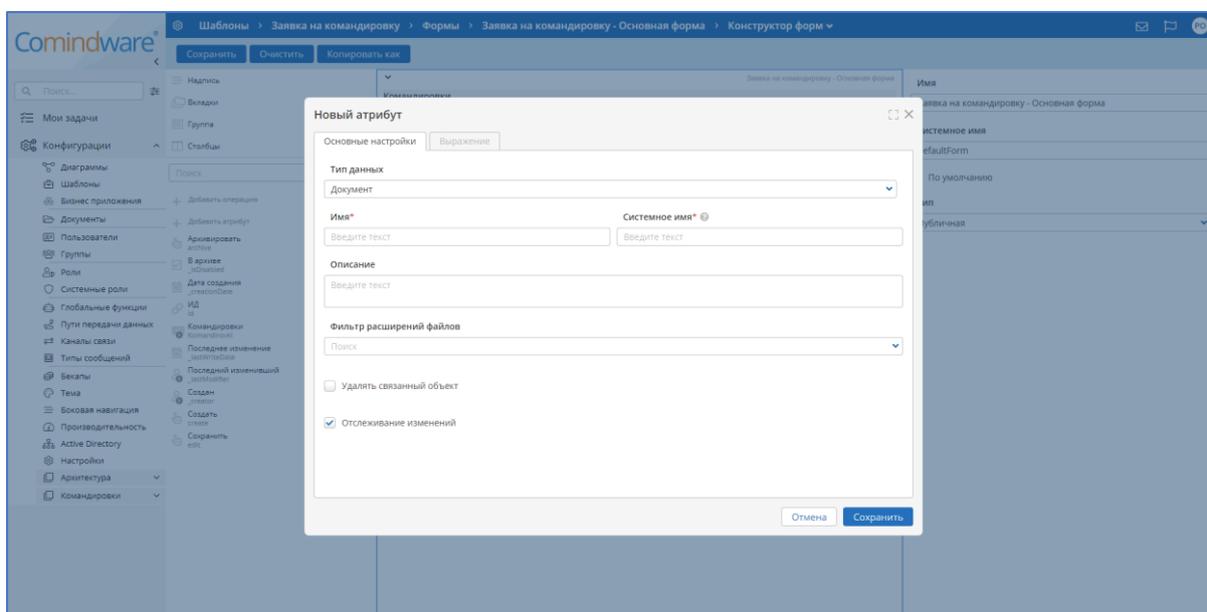


Рис. 34 Создание атрибута типа документ

- Пользователь – атрибут содержит данные о выбранном пользователе Системы. Атрибут хранит идентификатор записи пользователя в качестве данных. Отображаемый атрибут настраивается в шаблоне записи, на который устанавливается ссылка. Для типа данных **Пользователь** доступна следующая детализация:

- Целевой шаблон (необязательное поле): указывается один из шаблонов пользователей, которые созданы в системе. Необязательное поле. Если шаблон не указан, то атрибут будет ссылаться на системный справочник Пользователи (описано в документе «Руководство системного администратора»). При этом в качестве отображаемого значения в системном справочнике Пользователи используется поле (атрибут) Имя (fullName).

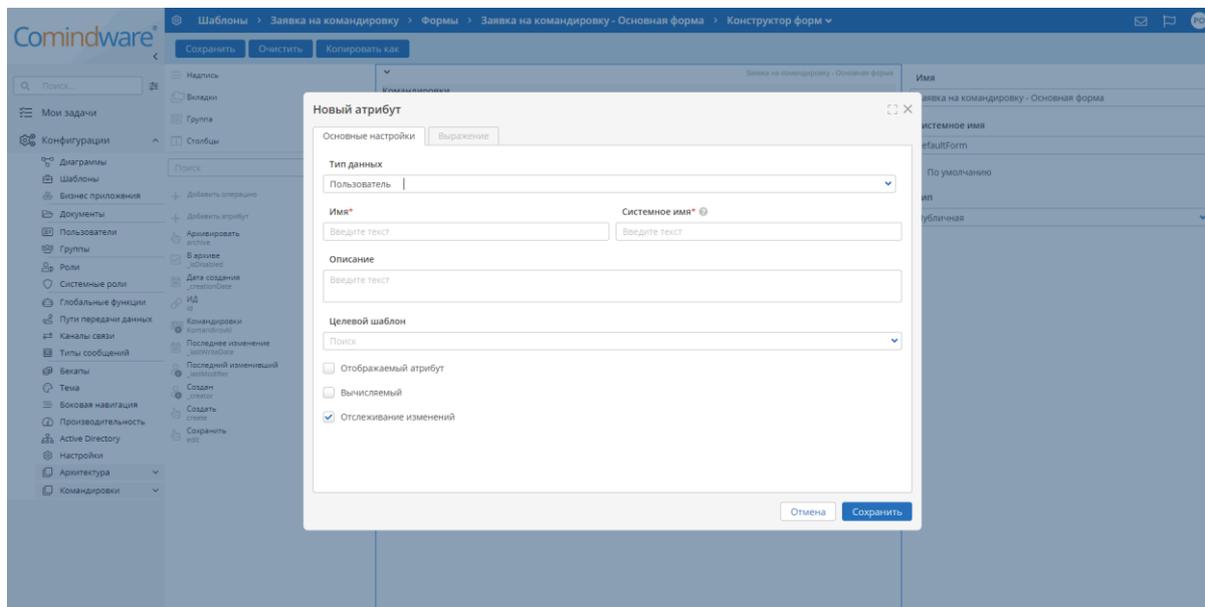


Рис. 35 Создание атрибута типа пользователь

- Ссылка – атрибут позволяет хранить ссылку на запись в другом шаблоне записи. Атрибут всегда содержит в качестве сохраняемого значения идентификатор записи (значение атрибута id), на которую он ссылается. Отображаемое значение настраивается в шаблоне записи¹⁹, на которую устанавливается ссылка. Для типа данных **Ссылка** доступна следующая детализация:
 - Формат отображения: **Показывать как ссылка** – данные отображаются в виде гиперссылки с возможностью перейти на форму редактирования выбранной записи, **Показывать как текст** – данные отображаются в виде текстовой строки в выпадающем списке и не позволяют перейти к связанной записи.
 - Целевой шаблон (обязательное поле): указывается шаблон записи, с которым будет связан настраиваемый атрибут-ссылка.
 - Сделать связь взаимной: флаг, который позволяет создать атрибут типа коллекция⁽³³⁾ в выбранном целевом шаблоне. Это необходимо в том случае, если в записи целевого шаблона необходимо будет отображать набор связанных с записью элементов текущего настраиваемого шаблона записи. Например, необходимо настроить сущность договор и атрибут-ссылку Контрагент на целевой шаблон Контрагенты. Для этого следует сделать атрибут-ссылку взаимной и в сущности Контрагент настроить атрибут-коллекцию, в котором будет содержаться набор элементов, ссылающихся на запись. Таким образом, на форме редактирования договора имеется простая

возможность без дополнительных действий отобразить перечень договоров выбранного контрагента.

- Имя коллекции (обязательное поле): указывается наименование создаваемого атрибута-коллекции⁽³³⁾ в целевом шаблоне. Доступно в случае, если выбран флаг «**Сделать связь взаимной**».
- Системное имя коллекции (обязательное поле): указывается системное имя создаваемого атрибута-коллекции⁽³³⁾ в целевом шаблоне. Доступно в случае, если выбран флаг «**Сделать связь взаимной**».

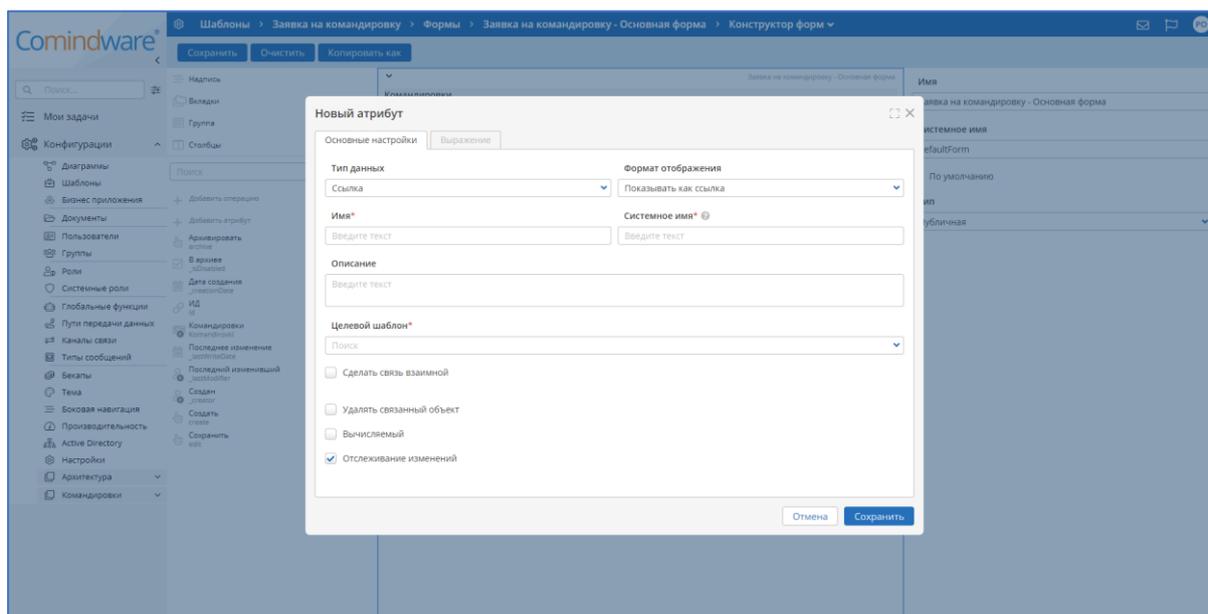


Рис. 36 Создание атрибута типа ссылка

- Коллекция – атрибут позволяет хранить, отображать и управлять набором связанных записей, созданных в другом шаблоне. Атрибут хранит набор ссылок на связанную запись в другом шаблоне записи. Атрибут может быть пустым, содержать один или несколько элементов. Ссылки хранятся в виде уникальных идентификаторов записей. Отображаемые значения определяются настройкой связанного шаблона записи⁽³²⁾. Для типа данных **Коллекция** доступна следующая детализация:
 - Целевой шаблон (обязательное поле): указывается шаблон записи, с которым будет связан настраиваемый атрибут-коллекция.
 - Использовать существующую ссылку: флаг, позволяющий указать возможность выбора существующего атрибута ссылки или создать новый атрибут. Флаг доступен пользователю в том случае, если в выбранном целевом шаблоне уже существует хотя бы один созданный атрибут типа ссылка на текущий настраиваемый шаблон. Для примера, если необходимо на шаблоне записи Контрагенты добавить атрибут типа коллекция со ссылкой на шаблон записи Договоры для отображения перечня договоров, существующих с контрагентом, то флаг «**Использовать существующую ссылку**» будет доступен, если в шаблоне записи Договоры уже существует атрибут типа ссылка Контрагент, который будет ссылаться на текущий шаблон записи Договоры. Если атрибут создан не будет, то шаблон пользователю будет доступно лишь создание нового атрибута типа ссылка при создании атрибута-коллекции на шаблоне записи Контрагенты.

- Имя для новой ссылки (обязательное поле): указывается наименование создаваемого атрибута-ссылки ⁽³²⁾ в целевом шаблоне. Доступно в случае, если флаг «Использовать существующую ссылку» отключен.
- Системное имя коллекции (обязательное поле): указывается системное имя создаваемого атрибута-ссылки в целевом шаблоне. Доступно в случае, если флаг «Использовать существующую ссылку» ⁽³²⁾ отключен.
- Существующая ссылка: выбирается среди атрибутов типа ссылка целевого шаблона, которые ссылаются на текущий настраиваемый шаблон записи. Это необходимо в случае, если в целевом шаблоне записи настроено несколько атрибутов ссылок на один шаблон записи. Например, для сущности Накладная необходимо указать несколько контрагентов: Плательщик, Грузополучатель, Грузоотправитель и т.д., которые являются одной или несколькими записями справочника Контрагенты.

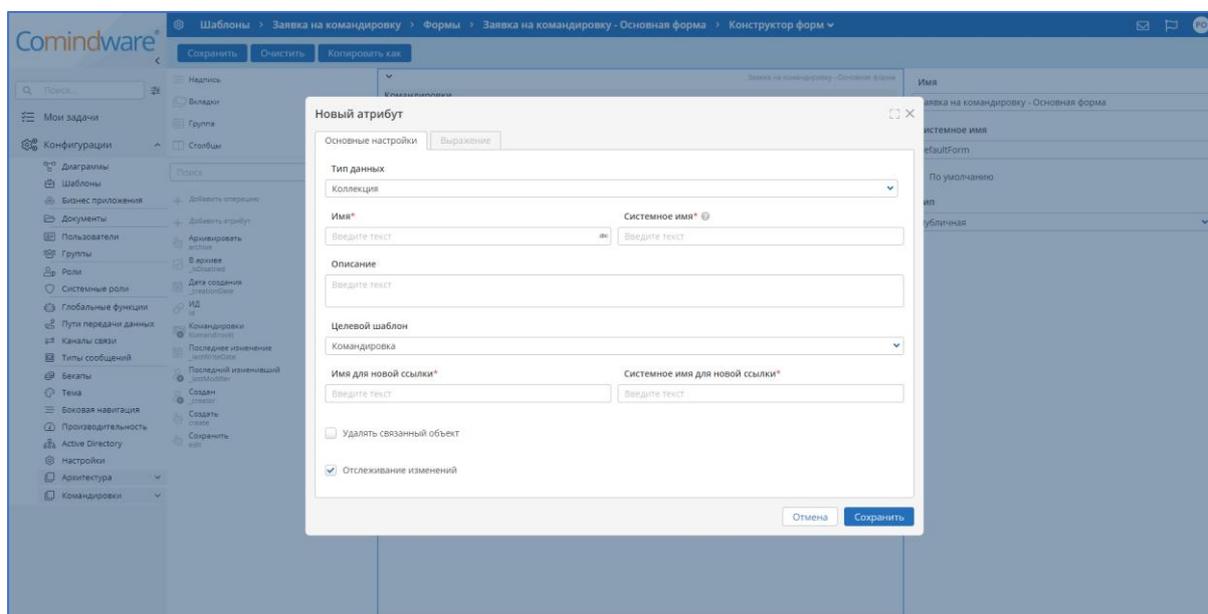


Рис. 37 Создание атрибута типа коллекция

Все создаваемые атрибуты имеют следующие общие атрибуты для заполнения:

- Имя (обязательное поле) – указывается наименование атрибута.
- Системное имя (обязательное поле) – указывается системное имя (синоним) атрибута для дальнейших ссылок в запросах и скриптах. Системное имя должно быть уникально в рамках настройки одного шаблона записи. Системное имя атрибута сериализуется в графовую модель данных на уровне базы данных и может использоваться для доступа к данным при написании запросов как внутри платформу, так и с помощью REST API.
- Описание (необязательное поле) – заполняется дополнительное текстовое описание назначения атрибута в случае необходимости. Рекомендуется заполнять и планируется использовать для генерации автодокументации в платформе.

В следующей таблице приведено описание доступных при настройке атрибута флагов.

Опция	Действие
Отображаемый атрибут	Атрибут будет по умолчанию отображаться в ссылках на шаблон записи. Если в качестве отображаемого не выбран ни один атрибут, в качестве такового используется формат «#id записи».
Уникальный	Значение атрибута будет проверяться на уникальность в границах текущего шаблона записи при сохранении записи.
Отслеживание изменений для атрибута	Все изменения значения атрибутов сохраняются в журнале истории изменений в платформе. Для просмотра истории изменений атрибута нужно нажать на кнопку настроек на форме и выбрать вкладку «История» в выпадающем списке.
Вычисляемый	Значение атрибута будет вычисляться на основании заданного выражения. Вычисляемый атрибут недоступен для изменения пользователем и не хранится в базе данных. Его значение вычисляется в момент отображения на интерфейсе пользователя.
Удалять связанный объект	Показывает действие, выполняемое при удалении ссылки на связанный объект. Если флаг указан, то объект будет удаляться. Если флаг не установлен, то при удалении ссылки будет рваться связь.

Следующая таблица показывает доступность указания флага в зависимости от типа данных атрибута.

Тип данных атрибута	Отображаемый атрибут	Уникальный	Отслеживание изменений для атрибута	Удалять связанный объект	Вычисляемый
Текст	+	+	+	-	+
Число	+	+	+	-	+
Длительность	+	-	+	-	+
Дата / время	+	-	+	-	+
Логический	-	-	+	-	+
Документ	-	-	+	+	-
Пользователь	+	-	+	-	+
Ссылка	-	-	+	+	+
Коллекция	-	-	+	+	-

2.3.2 Выражения

При отмеченном флаге **Вычисляемый** на форме редактирования атрибута становится активной вкладка **Выражение**, в которой возможно указать программное выражение с помощью языка выражений платформы⁽²²³⁾ или RDF N3 (подробно о языке RDF N3 описано в руководстве разработчика платформы).

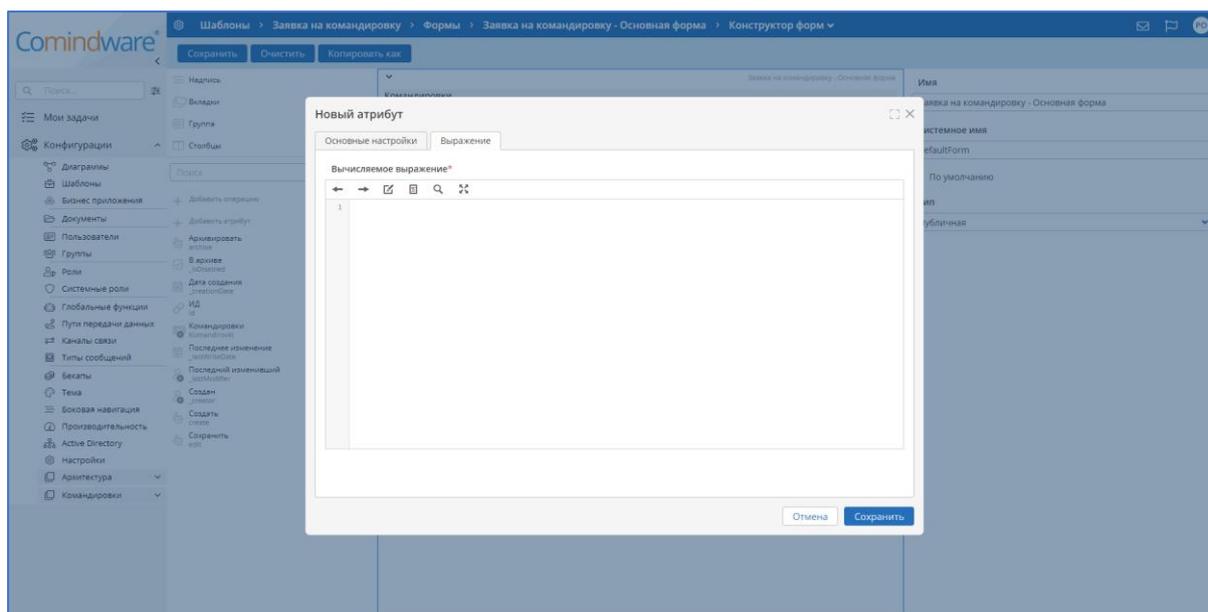


Рис. 38 Вкладка «Выражение»

Выражение должно возвращать результат в соответствии с типом данных атрибута. В случае, если пользователь снял флаг «Вычисляемый» с атрибута и сохранил изменения. При повторном его включении сохраненное ранее выражение на одноименной вкладке не восстанавливается.

2.3.3 Связи между шаблонами записей

Связь между шаблонами записей необходимо для описания логической связи между сущностями. Платформа поддерживает связи следующих типов:

Тип связи	Описание	Пример
Ссылка	Ссылка на запись, созданную в другом шаблоне.	Запрос в тех. Поддержку имеет ссылку на карточку отправившего его клиента.
Коллекция	Список ссылок на записи, созданные в другом шаблоне.	В карточке клиента есть перечень ссылок на отправленные этим клиентом запросы.

Ссылка

Данный вид связи относится к типу N:1. Это означает, что несколько элементов настраиваемого шаблона записи могут ссылаться на одну запись в целевом шаблоне записи. В одной записи может быть указана ссылка только на один элемент в связанном шаблоне записи.

В качестве примеров можно привести следующие:

- Перемещение товара. Для перемещения создается специальный документ, который имеет минимум две ссылки на одну сущность Склады: Склад откуда и Склад куда. Кроме этого, может иметь отдельную сущность Номенклатуры перемещений, в которой есть минимум две ссылки: Документ перемещение (на документ Перемещение) и Товар (на справочник Товары).

- Назначение сотрудника на работу. При этом создается документ Назначение, которые имеет следующие атрибуты-ссылки: Сотрудник (на справочник Сотрудники), Штатная позиция (на справочник Штатные позиции). В свою очередь штатная позиция – это набор ссылок, часть из которых на подразделение и должность.
- Заказ покупателя. Заказ покупателя при размещении ссылается на справочник покупателей. Кроме этого, в заказе присутствует сущность Детализация заказа, которая ссылается на заказ и на справочник номенклатуры.

Коллекция

Данный вид связи относится к типу 1:N. Это означает, что к одной записи может быть связано несколько уникальных записей связанного шаблона записи. Так как одна запись не может ссылаться на несколько записей связанного шаблона с использованием атрибута-ссылки, это значит, что в коллекции в разных записях не может присутствовать одна и та же запись.

- **Ограничение.** Атрибут типа коллекция всегда существует во взаимосвязи с атрибутом типа ссылка. В коллекции могут присутствовать только записи, которые ссылаются на текущую запись (в котором настроен атрибут-коллекция) с помощью атрибута-ссылки.

Коллекция – это вырожденный элемент для отображения обратной связи при наличии атрибута ссылки на текущую сущность.

Коллекция, представленная в виде древовидной структуры, описана в п. 2.4.2. Настройка коллекции со связью N-N описана в п. 6.3.1.

2.3.4 Редактирование атрибутов

Чтобы отредактировать атрибут, в списке атрибутов шаблона записи необходимо выбрать атрибут двойным нажатием на левую кнопку мыши. При этом открывается форма изменения атрибута.

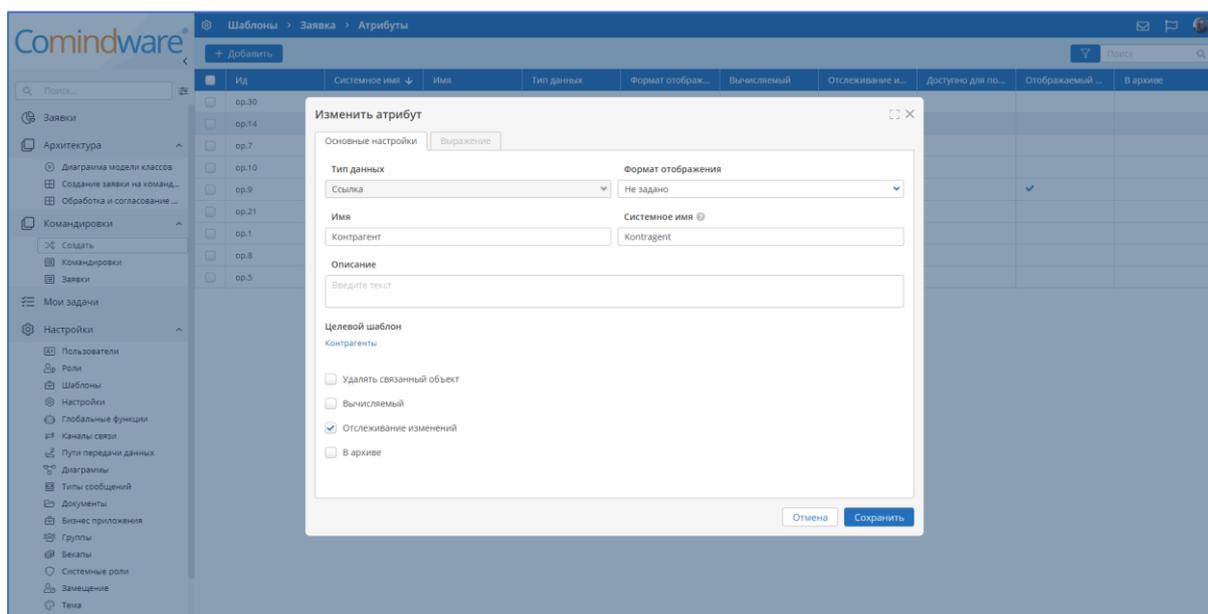


Рис. 39 Редактирование атрибута

На форме изменения атрибута возможно отредактировать характеристики атрибута кроме следующих:

- Тип данных.
- Целевой шаблон, если актуален для выбранного типа данных атрибута.

Крайне не рекомендуется изменять системное имя атрибута на существующем решении, так как системное имя атрибута уже может фигурировать в каких-либо скриптах платформы или во внешней интеграции.

2.3.5 Архивирование и разархивирование атрибутов

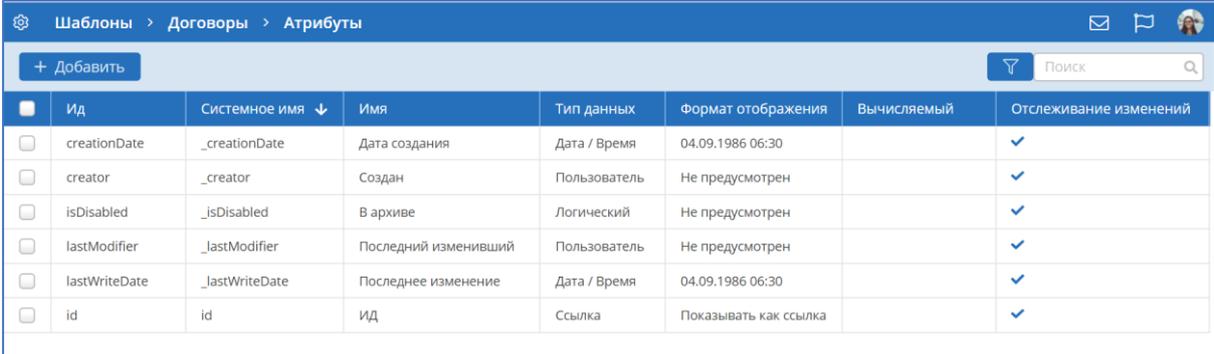
Платформа не поддерживает удаление атрибутов, так как это может нарушить целостность данных. Для этих целей введено понятие архивирования («мягкого удаления») с возможностью восстановления.

Архивные атрибуты и данные, введенные на форме редактирования пользователями, сохраняются в системе. Архивирование атрибута не влияет на настроенные формы и списки, в которых данный атрибут участвует. Пользователи продолжают работать со списками и формами в рабочем режиме и вводить данные. Значения, введенные в поля, связанные с архивными атрибутами, будут сохраняться.

В конструкторе форм и списков архивные атрибуты будут отсутствовать в списке атрибутов, доступных для добавления на форму или на список, однако уже добавленные архивные атрибуты будут присутствовать на форме и будут выделены красным цветом. Их можно только удалить с формы или убрать со списка. Если после изменения формы или списка архивный атрибут остался на форме или в списке, сохранение формы или списка будет невозможно.

Для того, чтобы архивировать атрибут необходимо:

1. Перейти в раздел **Атрибуты** выбранного шаблона записи. Для этого кликните по иконке пользователя в правом верхнем углу, выберите «Глобальные настройки», перейдите в раздел **Шаблоны**, выберите нужный шаблон и перейдите в раздел **Атрибуты** (Рис. 21).



<input type="checkbox"/>	Ид	Системное имя ↓	Имя	Тип данных	Формат отображения	Вычисляемый	Отслеживание изменений
<input type="checkbox"/>	creationDate	_creationDate	Дата создания	Дата / Время	04.09.1986 06:30		✓
<input type="checkbox"/>	creator	_creator	Создан	Пользователь	Не предусмотрен		✓
<input type="checkbox"/>	isDisabled	_isDisabled	В архиве	Логический	Не предусмотрен		✓
<input type="checkbox"/>	lastModifier	_lastModifier	Последний изменивший	Пользователь	Не предусмотрен		✓
<input type="checkbox"/>	lastWriteDate	_lastWriteDate	Последнее изменение	Дата / Время	04.09.1986 06:30		✓
<input type="checkbox"/>	id	id	ИД	Ссылка	Показывать как ссылка		✓

Рис. 40 Перечень атрибутов шаблона записи

2. Отметить флаги у атрибутов, которые требуется архивировать (первая колонка списка). После этого появится кнопка «Архивировать».

<input type="checkbox"/>	Ид	Системное имя	Имя	Тип данных	Формат отображ...	Вычисляемый	Отслеживание из...	Доступно для по...	Отображаемый а...	В архиве
<input type="checkbox"/>	creationDate	_creationDate	Дата создания	Дата / Время	04.09.1986 08:30		✓			
<input type="checkbox"/>	creator	_creator	Создан	Пользователь	Не предусмотрен		✓			
<input type="checkbox"/>	isDisabled	_isDisabled	В архиве	Логический	Не предусмотрен		✓			
<input type="checkbox"/>	lastModifier	_lastModifier	Последний измен...	Пользователь	Не предусмотрен		✓			
<input type="checkbox"/>	lastWriteDate	_lastWriteDate	Последнее измене...	Дата / Время	04.09.1986 08:30		✓			
<input type="checkbox"/>	id	id	ИД	Ссылка	Показывать как сс...		✓			
<input checked="" type="checkbox"/>	op.125	Tekstdogovora	Текст договора	Документ	Не предусмотрен					

Рис. 41 Отмеченные атрибуты и кнопка «Архивировать»

3. Нажать на кнопку «Архивировать» в панели инструментов списка атрибутов.

Чтобы настроить отображение архивных атрибутов, необходимо нажать на кнопку в панели инструментов списка атрибутов, выбрать «В архиве» и сохранить. При этом архивные атрибуты будут отмечены флагом в колонке «В архиве».

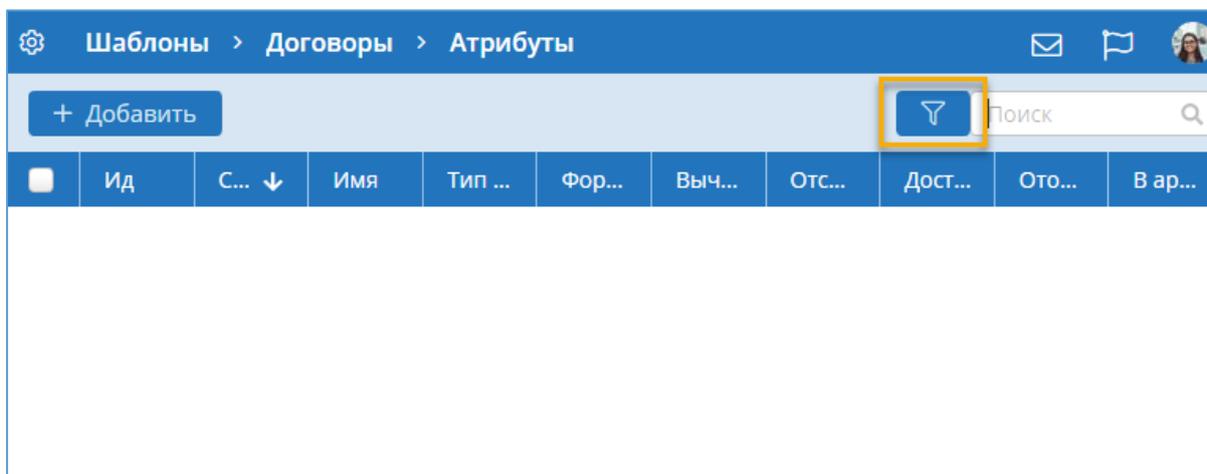


Рис. 42 Расположение кнопки настройки отображения системных и архивных атрибутов

Для того, чтобы восстановить атрибут из архива необходимо:

1. Перейти в раздел **Атрибуты** с отображением архивных атрибутов.
2. Отметить флаги у атрибутов, которые требуется восстановить из архива (первая колонка списка).

<input type="checkbox"/>	Ид	Системное имя	Имя	Тип данных	Формат отображ...	Вычисляемый	Отслеживание из...	Доступно для по...	Отображаемый а...	В архиве
<input type="checkbox"/>	creationDate	_creationDate	Дата создания	Дата / Время	04.09.1986 08:30		✓			
<input type="checkbox"/>	creator	_creator	Создан	Пользователь	Не предусмотрен		✓			
<input type="checkbox"/>	isDisabled	_isDisabled	В архиве	Логический	Не предусмотрен		✓			
<input type="checkbox"/>	lastModifier	_lastModifier	Последний измен...	Пользователь	Не предусмотрен		✓			
<input type="checkbox"/>	lastWriteDate	_lastWriteDate	Последнее измене...	Дата / Время	04.09.1986 08:30		✓			
<input type="checkbox"/>	id	id	ИД	Ссылка	Показывать как сс...		✓			
<input checked="" type="checkbox"/>	op.125	Tekstdogovora	Текст договора	Документ	Не предусмотрен					✓

Рис. 43 Отмеченные атрибуты и кнопка «Разархивировать»

3. Нажать на кнопку «Разархивировать» в панели инструментов списка атрибутов.

У восстановленных из архива атрибутов пропадает флаг в колонке «В архиве».

2.4 Настройка форм

Форма редактирования – это настроенное визуальное представление, предназначенное для ввода и редактирования данных одной записи и связанных с ней записей.

Настройка формы производится в визуальном редакторе форм (Конструктор форм) и позволяет мышью сформировать произвольный и удобный дизайн для конечного пользователя.

2.4.1 Добавление формы

Для перехода к настройке форм:

1. Кликнуть по иконке пользователя в правом верхнем углу и выбрать пункт «Глобальные настройки».
2. Перейдите в раздел **Шаблоны** и выберите нужный шаблон.

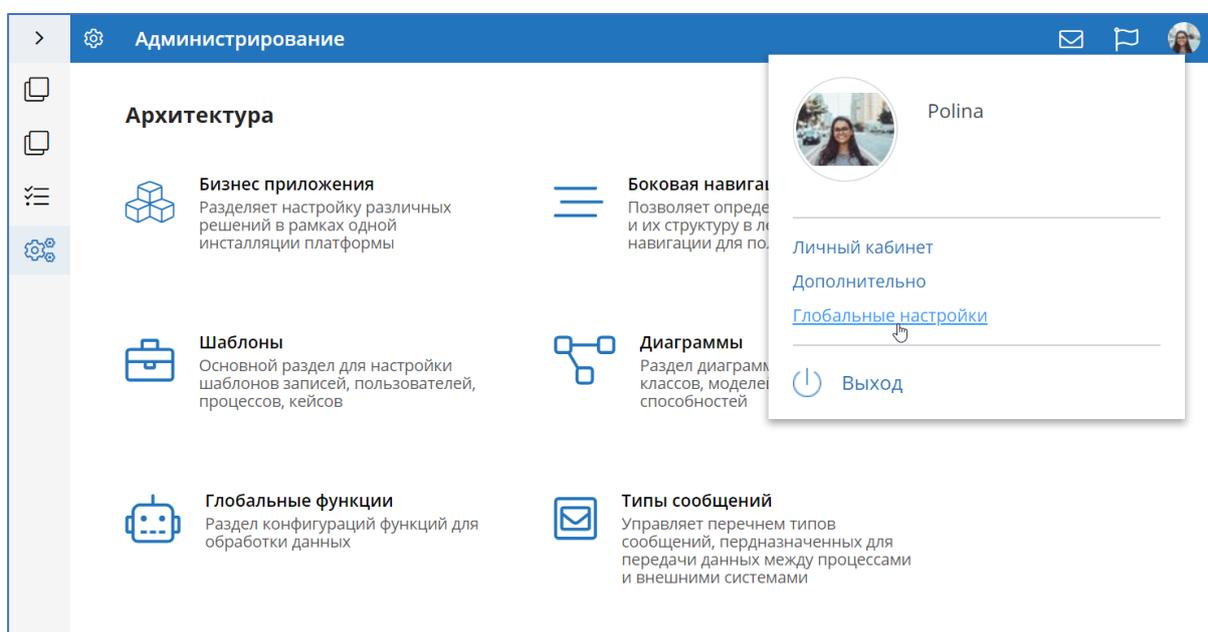


Рис. 44 Переход в раздел Шаблоны

3. Перейдите в раздел **Формы**.

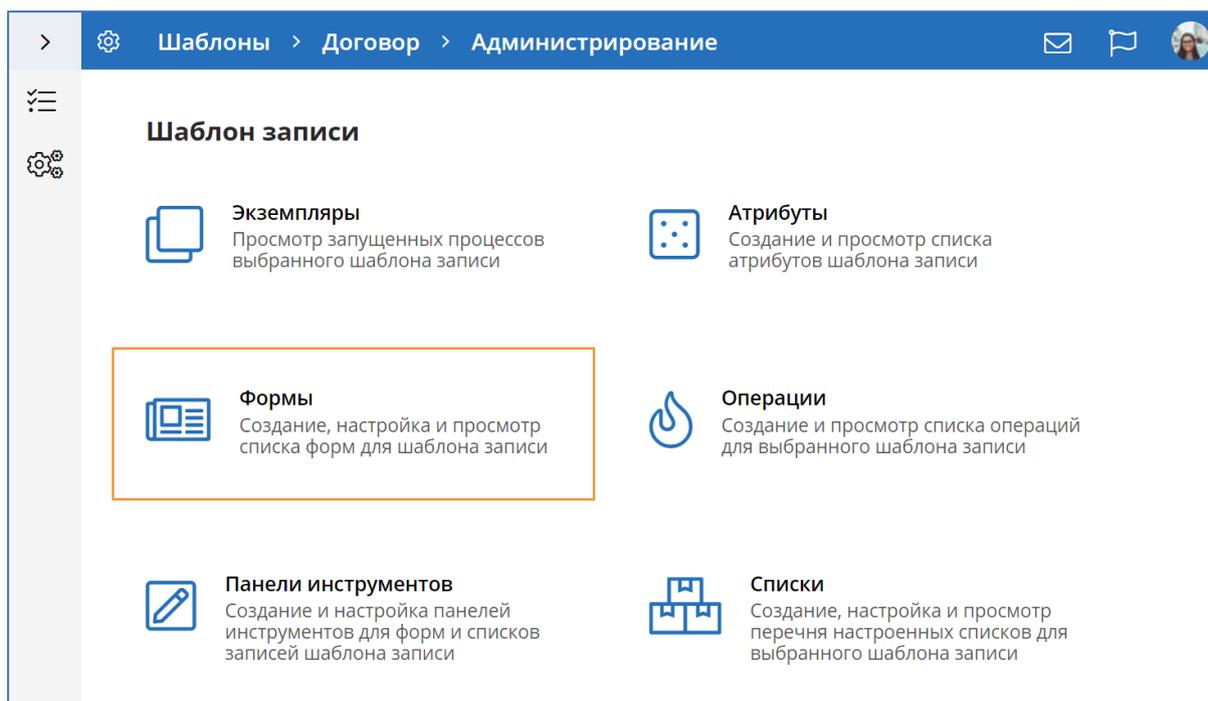


Рис. 45 Переход в раздел **Формы**

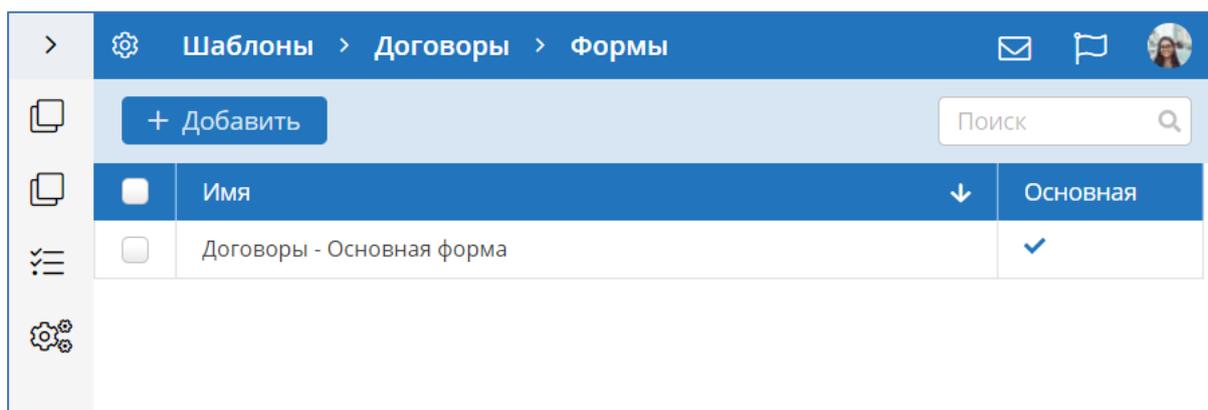


Рис. 46 Список форм шаблона записи

Открывается список настроенных форм шаблона записи. В новом шаблоне записи по умолчанию присутствует форма, которая носит наименование от наименования шаблона записи с постфиксом «Основная форма».

- Для добавления новой формы редактирования нажмите на кнопку «Добавить».

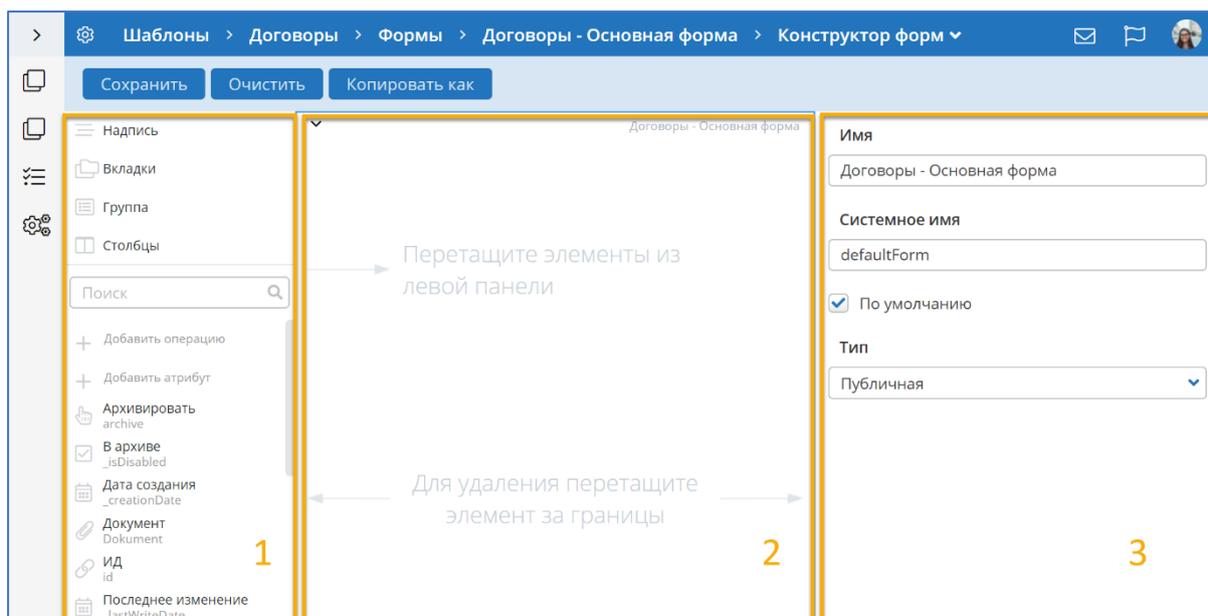


Рис. 47 Внешний вид конструктора форм

На пользовательском интерфейсе открывается конструктор формы редактирования шаблона записи.

Конструктор форм разбит на 3 области:

1. Панель элементов – отображается перечень доступных элементов для добавления в рабочую область конструктора.
2. Рабочая область – основная визуальная область конструктора форм, где производится наполнение и конструирование внешнего вида формы редактирования.
3. Панель настроек – содержит набор характеристик выбранного элемента в рабочей области. Для выбора доступны сама форма, а также любой элемент, помещаемый в рабочую область. По умолчанию, когда нет ни одного элемента выбрана и, следовательно, отображаются характеристики формы.

Подробное описание конструктора форм приведено в разделе «Настройка формы» (2.4.2).

5. После настройки формы сохраните, нажав на кнопку «Сохранить».

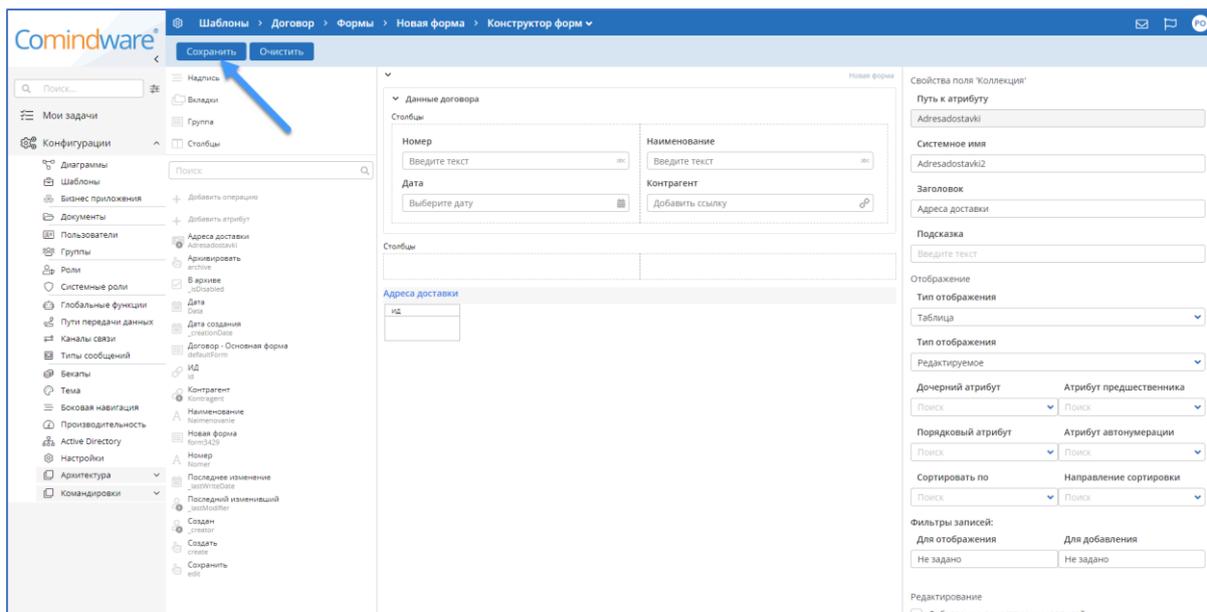


Рис. 48 Расположение кнопки «Сохранить»

6. Для копирования формы необходимо нажать на кнопку «Копировать как».

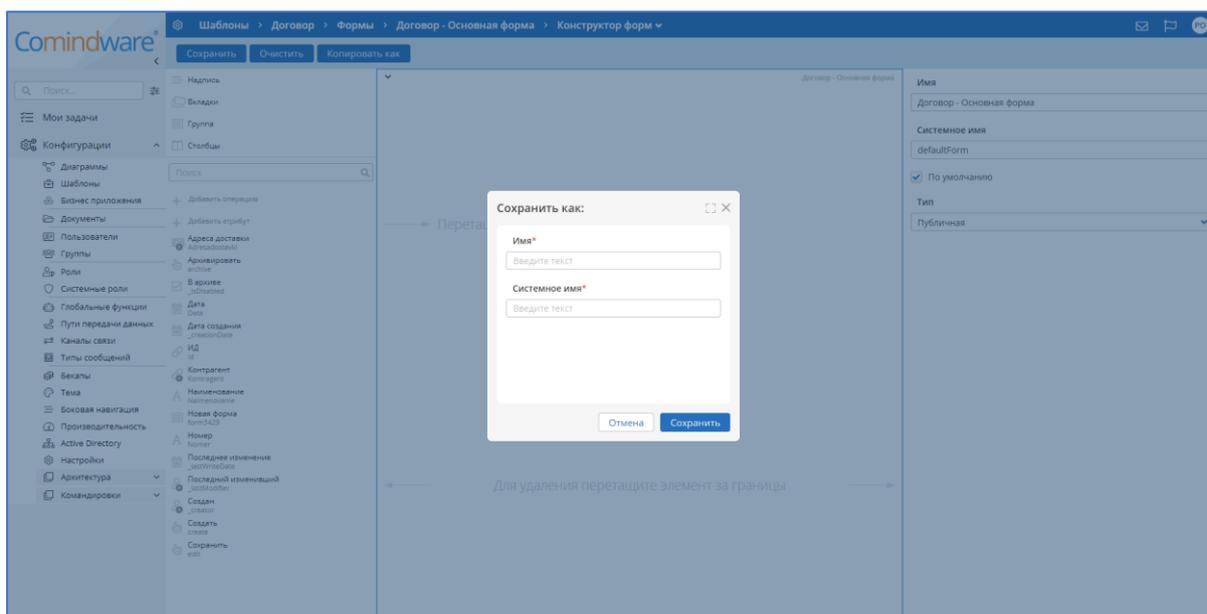


Рис. 49 Форма ввода реквизитов новой формы редактирования

На открывшемся окне необходимо заполнить наименование и системное имя новой формы.

7. Для сохранения новой формы редактирования нажмите на кнопку «Сохранить». Для отмены сохранения новой формы, необходимо нажать на кнопку «Отмена».

- **Внимание.** При копировании формы, копируются также все правила формы (2.4.4), которые с ней связаны.

2.4.2 Настройка формы

Основная работа по настройке будущей формы выполняется в конструкторе форм. Разработка формы производится путем вынесения из панели элементов на рабочую область необходимых элементов управления.

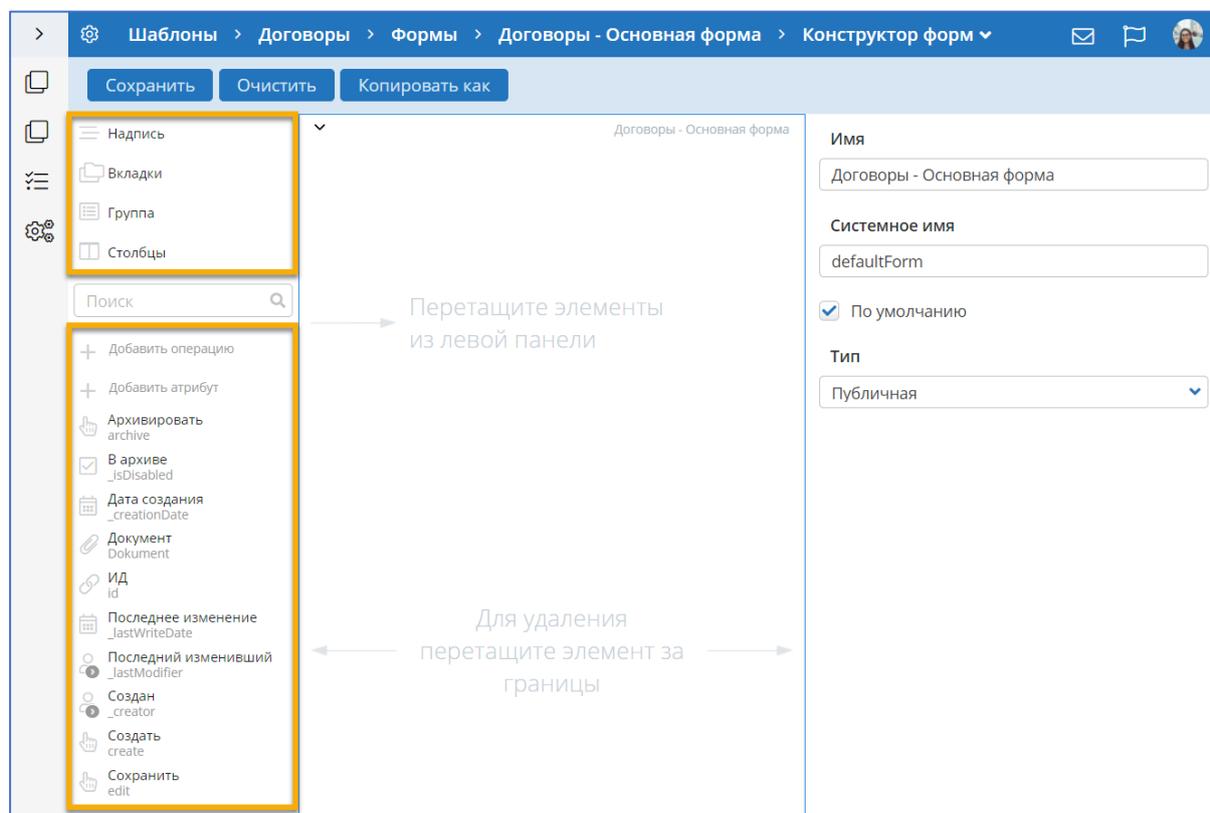


Рис. 50 Фиксированные и динамические элементы панели элементов конструктора форм

Для формы могут быть определены следующие характеристики:

- Имя – наименование формы, которое будет отображаться во всех разделах платформы, где необходимо осуществлять выбор настроенных форм редактирования.
- Системное имя – уникальное текстовое системное имя элемента в рамках шаблона записи. Может содержать только буквы латинского алфавита, цифры и знак «_».
- По умолчанию – флаг, который говорит о том, что указанная форма редактирования открывается по умолчанию при переходе к форме редактирования объекта.
- Тип – указывает тип формы, который может принимать следующие значения:
 - Публичная – форма будет доступна всем пользователям (которые имеют права доступа на форму) для выбора при редактировании записи.
 - Внутренняя – форма будет доступна всем пользователям (которые имеют права доступа на форму) только по прямой открытой ссылке или при прямом указании на операции или коллекции.

Панель элементов содержит следующие элементы:

- Фиксированные элементы – это набор визуальных элементов, которые не зависят от шаблона записи, на котором настраивается форма.

- **Динамические элементы** – это набор атрибутов, операций, форм, настроенных на текущем шаблоне записи, позволяющий вынести данные и кнопки на форму редактирования.

Добавление элементов на форму редактирования производится путем «перетаскивания» элементов с панели элементов конструктора форм в рабочую область и определения индивидуальных характеристик визуальных элементов с помощью панели настройки.

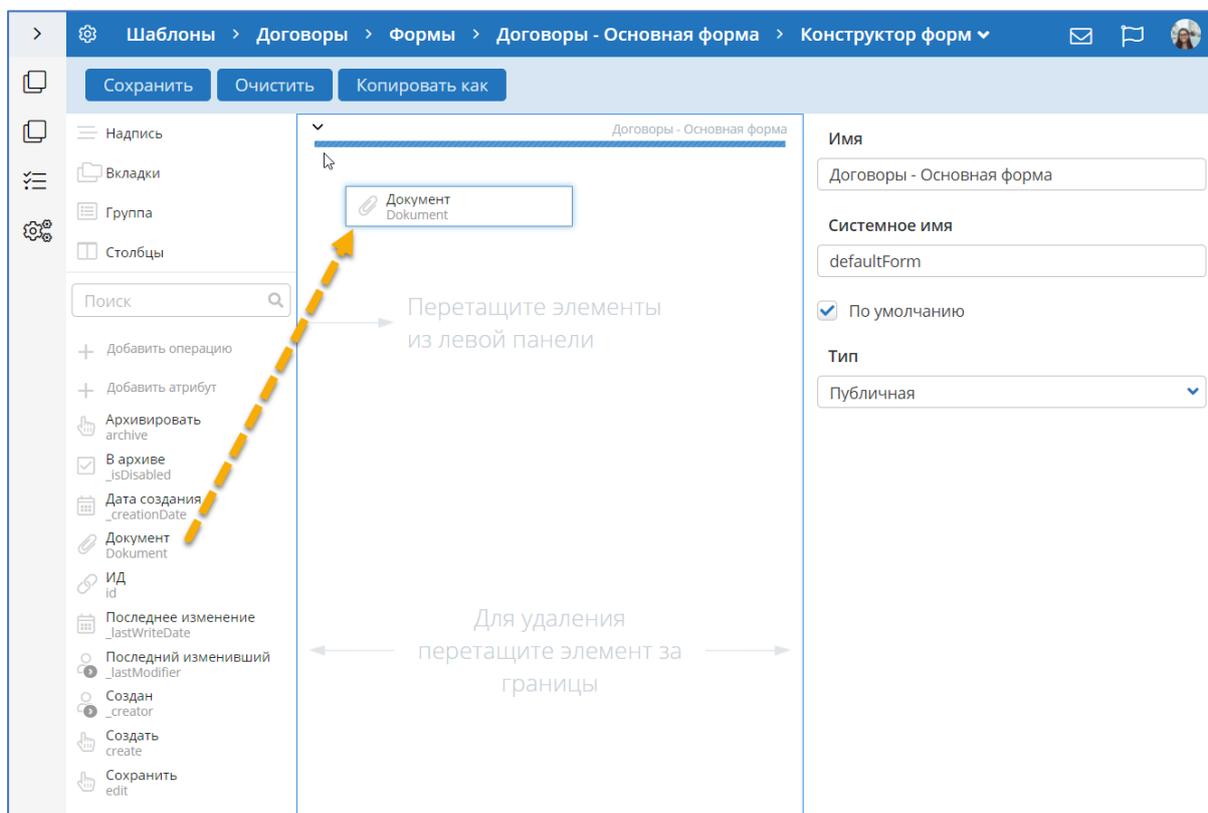


Рис. 51 Механизм добавления визуальных элементов на форму

Для добавления визуального элемента на форму нужно выбрать элемент на левой панели и, удерживая левую кнопку мыши, перетащить элемент на рабочее поле. Области, доступные для переноса элемента на рабочее поле, подсвечиваются синими зонами. Чтобы завершить перенос, необходимо отпустить удерживаемую левую кнопку мыши элемент в доступной синей зоне (при этом она окрасится в темно-синий цвет). Таким образом визуальный элемент будет размещен в нужном месте рабочей области, который затем будет отображаться на форме редактирования (2.4).

Чтобы изменить положение уже добавленного визуального элемента в рабочей области, необходимо аналогичным образом осуществить перенос элемента в нужное место.

Для просмотра характеристик визуальных элементов, необходимо выбрать нужный элемент на рабочей области конструктора форм. Выбранный элемент выделяется цветом и в панели настройки появляется перечень доступных характеристик, которые доступны проектировщику.

Некоторые элементы, такие как выбранный столбец визуального элемента «Столбцы», не содержат никаких характеристик. Также имеется возможность посмотреть характеристики текущей формы редактирования, нажав левую кнопку мыши в любом свободном от добавленных элементов месте рабочей области.

Динамические визуальные элементы на панели элементов конструктора форм содержат строку поиска по наименованию и системному имени атрибута. Поиск по умолчанию производится по верхнему и следующему уровню связанных шаблонов записей. Для поиска по более глубоким уровням, необходимо раскрыть уровни вручную.

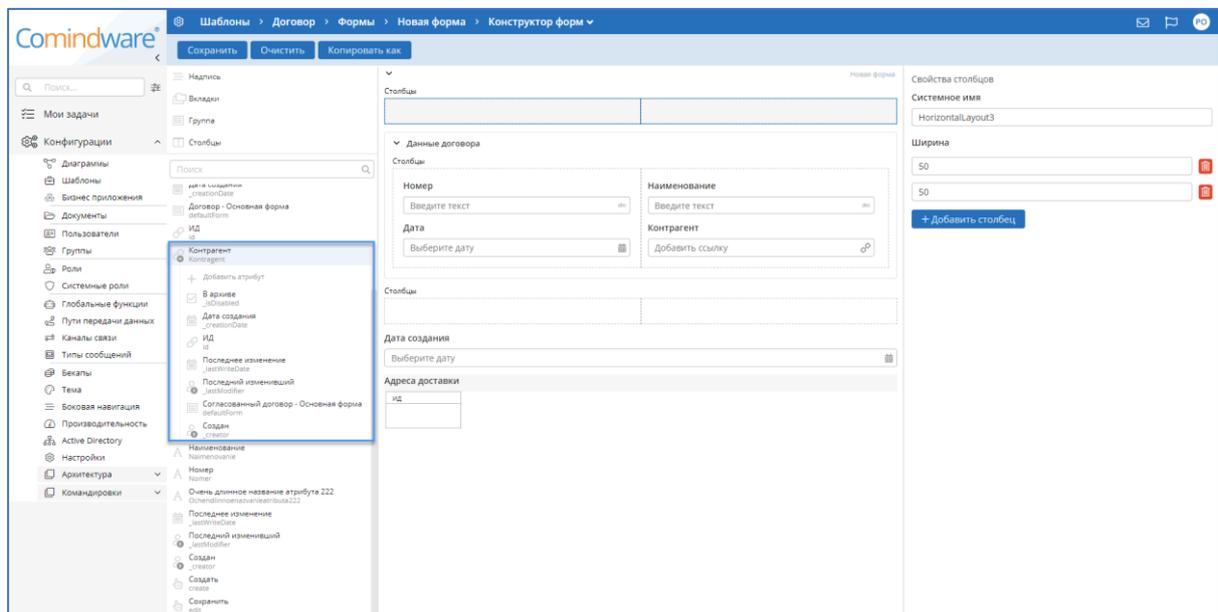


Рис. 52 Список элементов, связанного шаблона записи, доступных для помещения на форму

Ниже представлено описание фиксированных визуальных элементов панели элементов конструктора форм.

Надпись

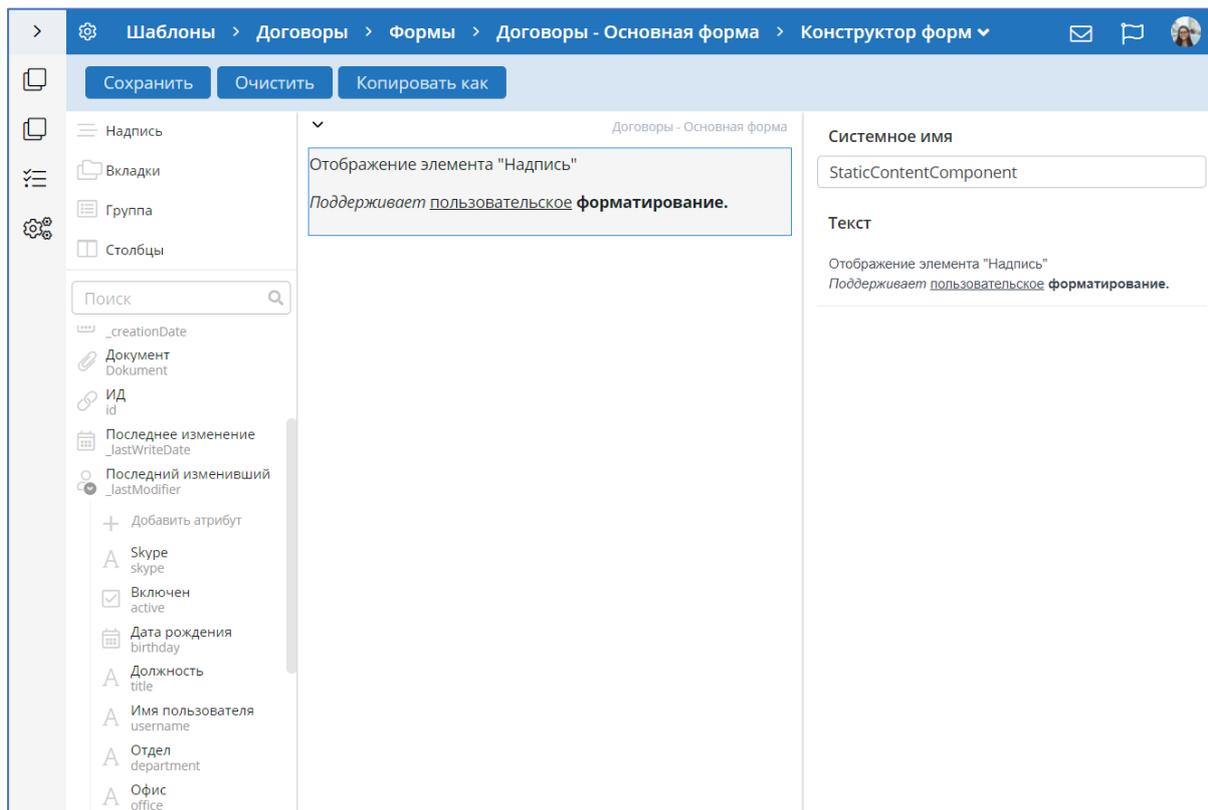


Рис. 53 Элемент «Надпись»

Предназначена для отображения статичного текста на пользовательской форме редактирования.

Элемент «Надпись» имеет следующие характеристики:

- Системное имя – уникальное текстовое системное имя элемента в рамках настраиваемой формы редактирования, используемое для настройки динамических правил на формы редактирования. Может содержать только буквы латинского алфавита, цифры и знак «_».
- Текст – предназначен для ввода и форматирования текстовой информации, которая будет отображаться в поле на интерфейсе пользователя.

На форме редактирования в пользовательском интерфейсе элемент «Надпись» выглядит следующим образом.

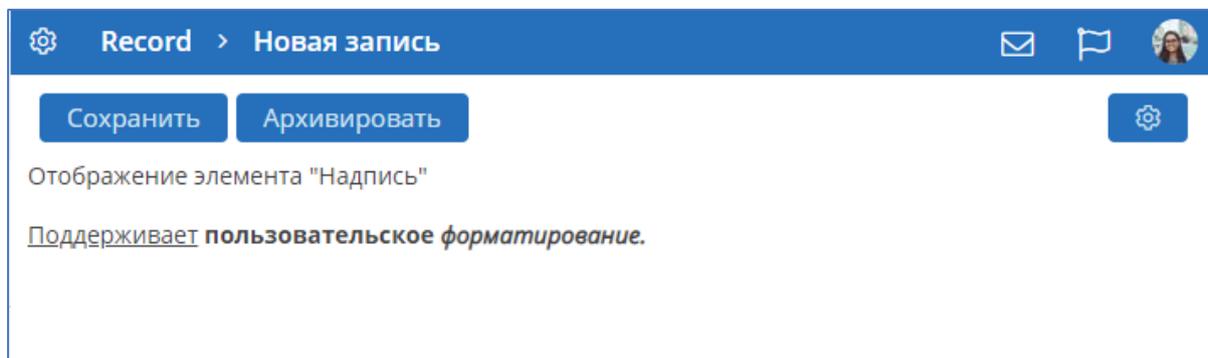


Рис. 54 Элемент «Надпись» на пользовательском интерфейсе

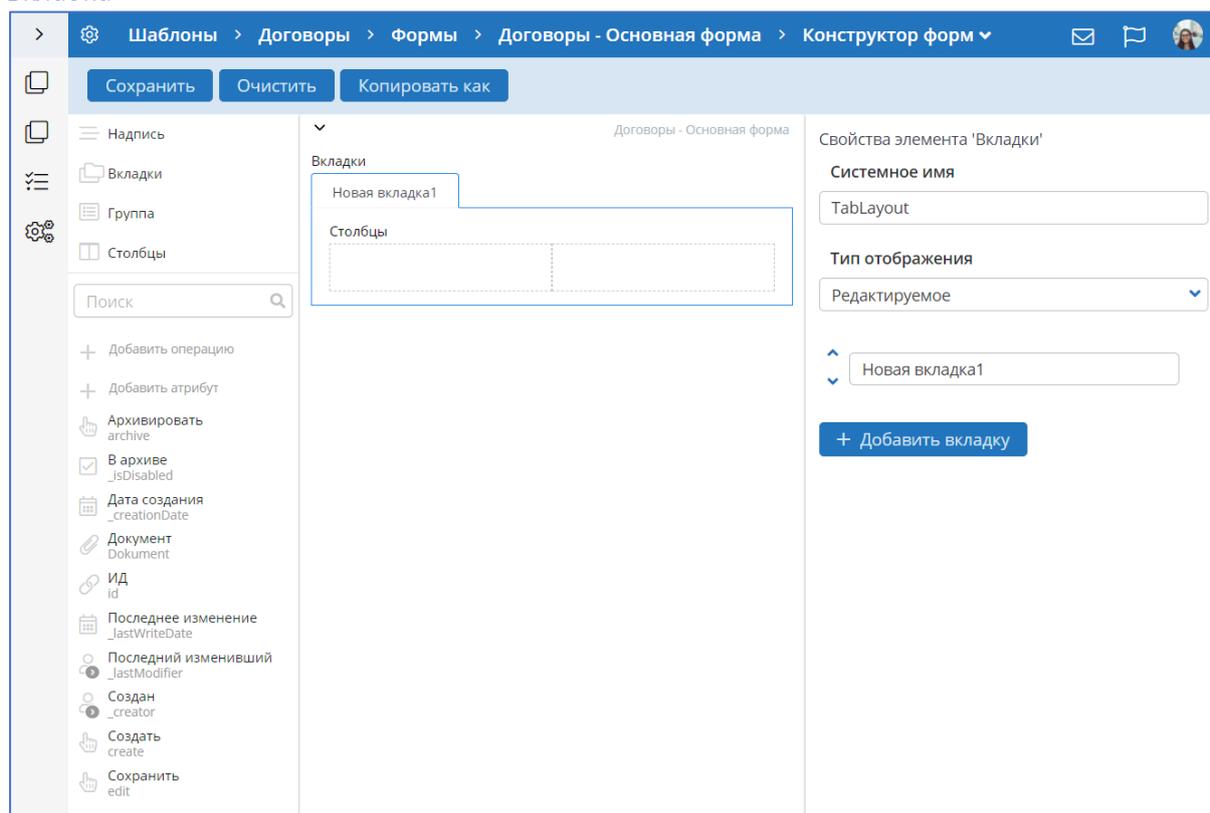


Рис. 55 Элемент «Вкладки»

Элемент-контейнер предназначен для создания элемента интерфейса пользователя, позволяющего хранить информацию в различных вкладках.

Элемент «Вкладки» имеет следующие характеристики:

- Системное имя – уникальное текстовое системное имя элемента в рамках настраиваемой формы редактирования, используемое для настройки динамических правил на формы редактирования. Может содержать только буквы латинского алфавита, цифры и знак «_».
- Тип отображения – тип доступа: «Редалируемое», «Только для чтения», «Скрытое».
- Кнопка Добавить вкладку – позволяет добавлять необходимое количество вкладок.
- Название – для каждой вкладки определяется наименование, которое будет отображаться на интерфейсе пользователя.
- Кнопки изменения порядка вкладок – позволяют менять порядок расположения вкладок на интерфейсе пользователя.
- Кнопки удаления вкладки – отображается в виде красной кнопки с изображением корзины при наведении курсора мыши на описание вкладки в панели настроек.

При добавлении вкладок на рабочую область конструктора форм, вкладки создаются с вложенным элементом «Столбцы», который возможно удалить.

Каждая вкладка имеет собственное системное имя и, следовательно, для каждой вкладки (как и для элемента в целом) имеется возможность настроить отдельное динамическое правило ⁽⁸⁷⁾ его отображения.

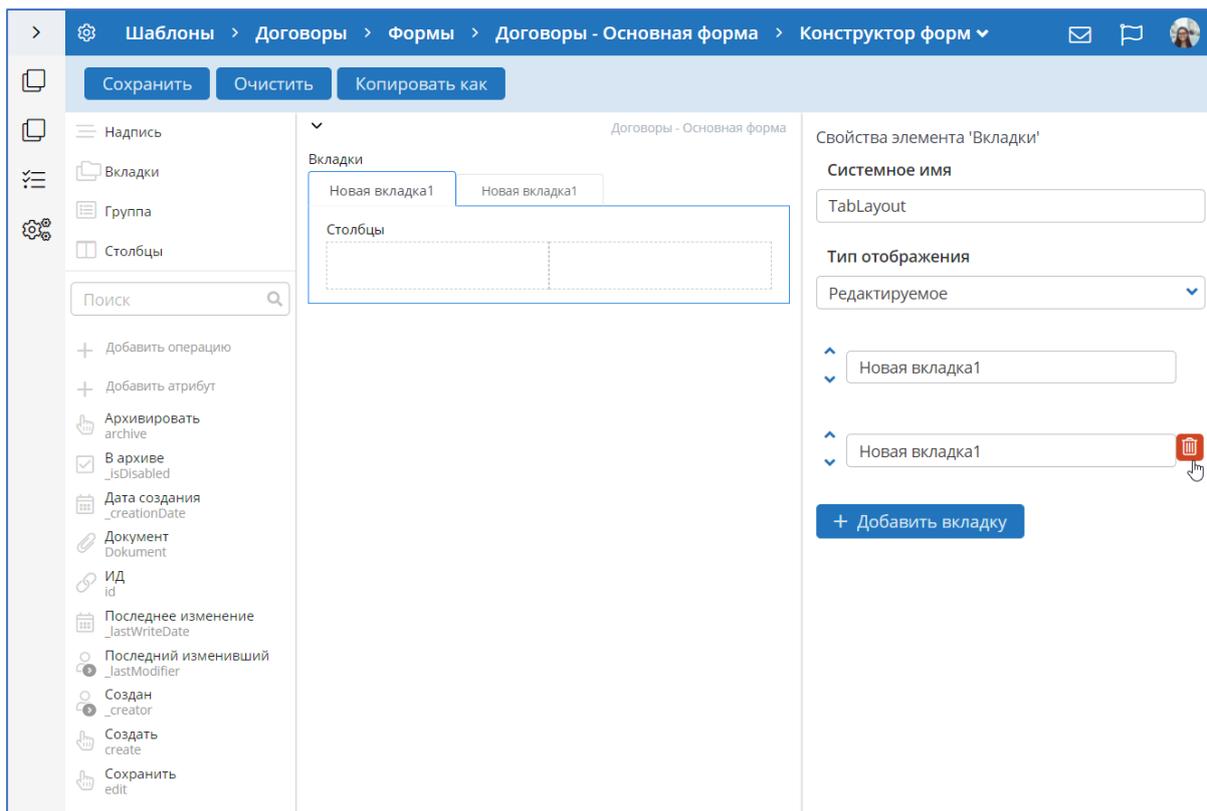


Рис. 56 Панель настроек элемента «Вкладки»

С помощью панели настроек элемента «Вкладки» имеется возможность добавлять новые вкладки, менять их порядок и удалять.

При удалении вкладки, если в ней находились какие-то вложенные визуальные элементы, платформа выдаст предупреждение с возможностью подтверждения действия или отмены.

При попытке удаления всего элемента «Вкладки», если хоть в одной вкладке присутствуют вложенные элементы, то платформа выдает аналогичное предупреждение.

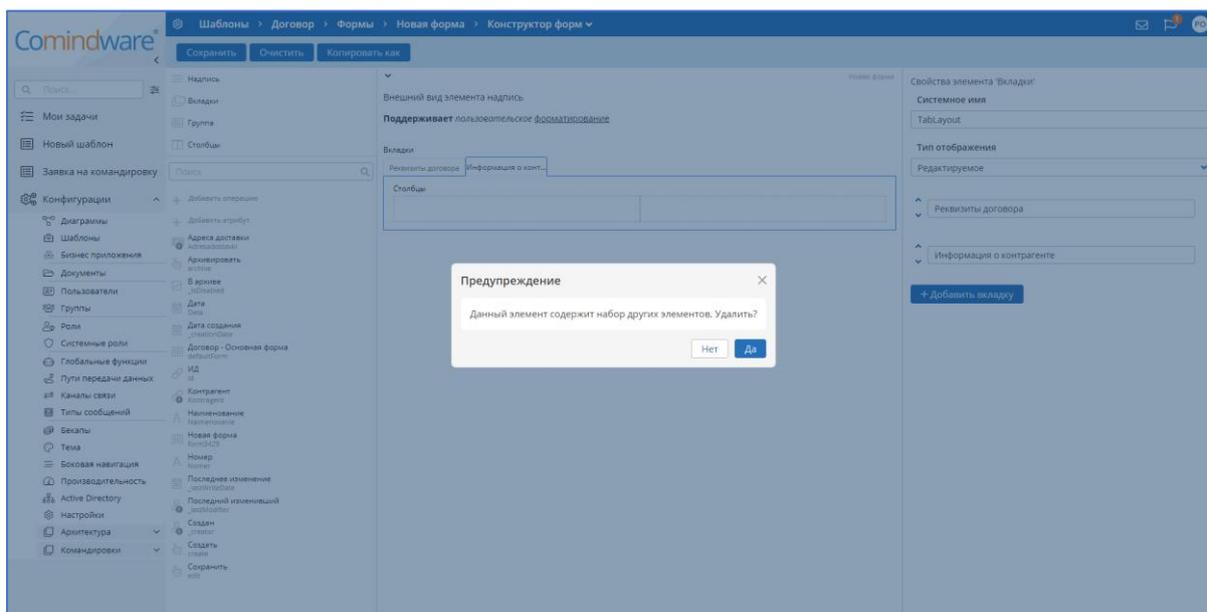


Рис. 57 Окно подтверждения удаления столбца

На форме редактирования в пользовательском интерфейсе элемент «**Вкладки**» выглядит следующим образом.

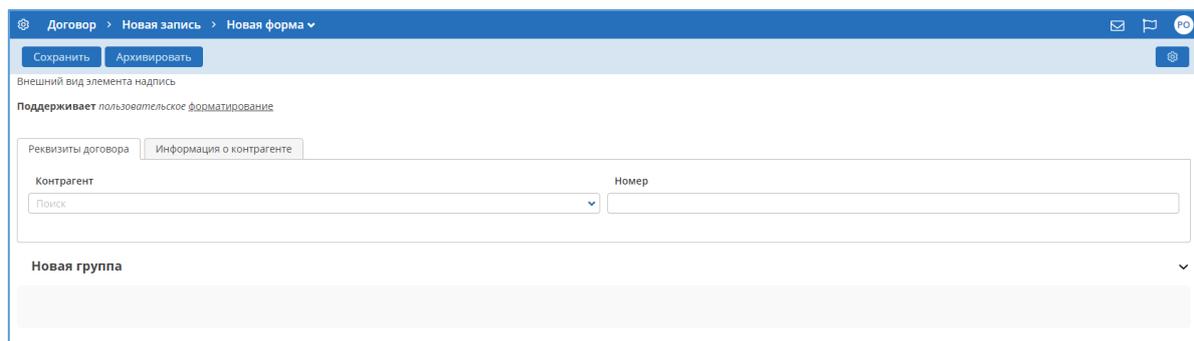


Рис. 58 Элемент «Вкладки» на пользовательском интерфейсе

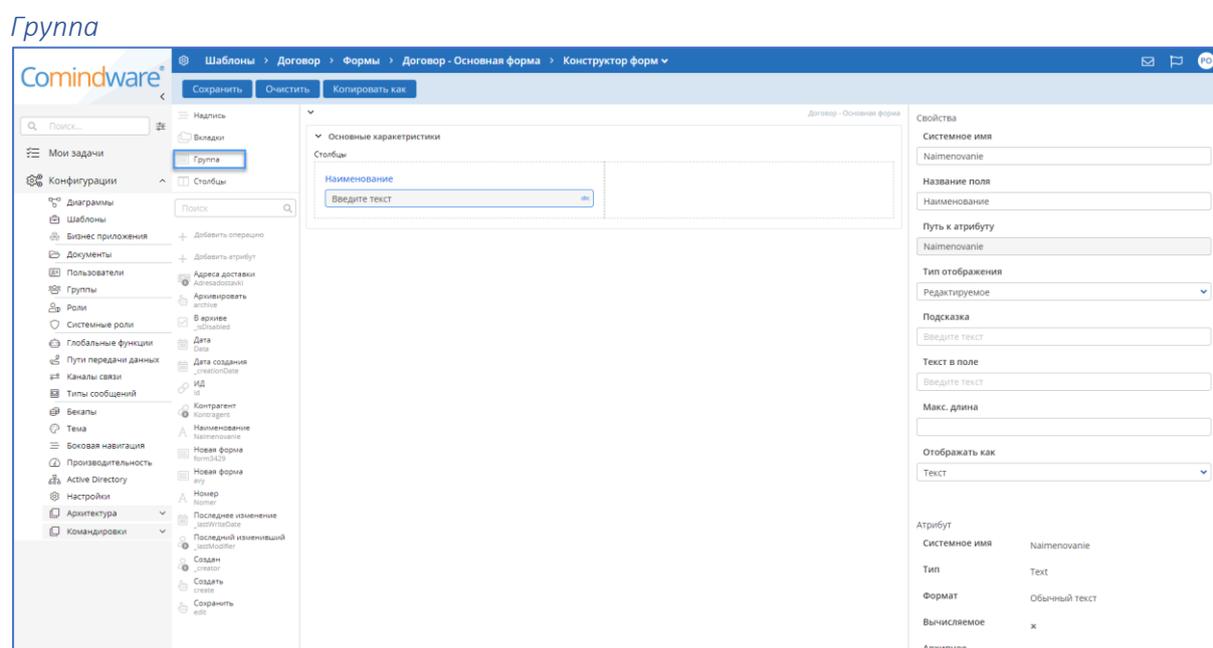


Рис. 59 Элемент «Группа»

Предназначена создания визуальной подписанной группы элементов с целью выделения требуемых элементов формы редактирования.

Элемент «**Группа**» имеет следующие характеристики:

- Системное имя – уникальное текстовое системное имя элемента в рамках настраиваемой формы редактирования, используемое для настройки динамических правил на формы редактирования. Может содержать только буквы латинского алфавита, цифры и знак «_».
- Имя – определяется текстовое наименование группы, которое будет отображено на интерфейсе пользователя на форме редактирования записи.

При добавлении группы на рабочую область конструктора форм, группа создается с вложенным элементом «**Столбцы**», который возможно удалить.

При удалении группы, если в ней находились какие-то вложенные визуальные элементы, платформа выдаст предупреждение с возможностью подтверждения действия или отмены.

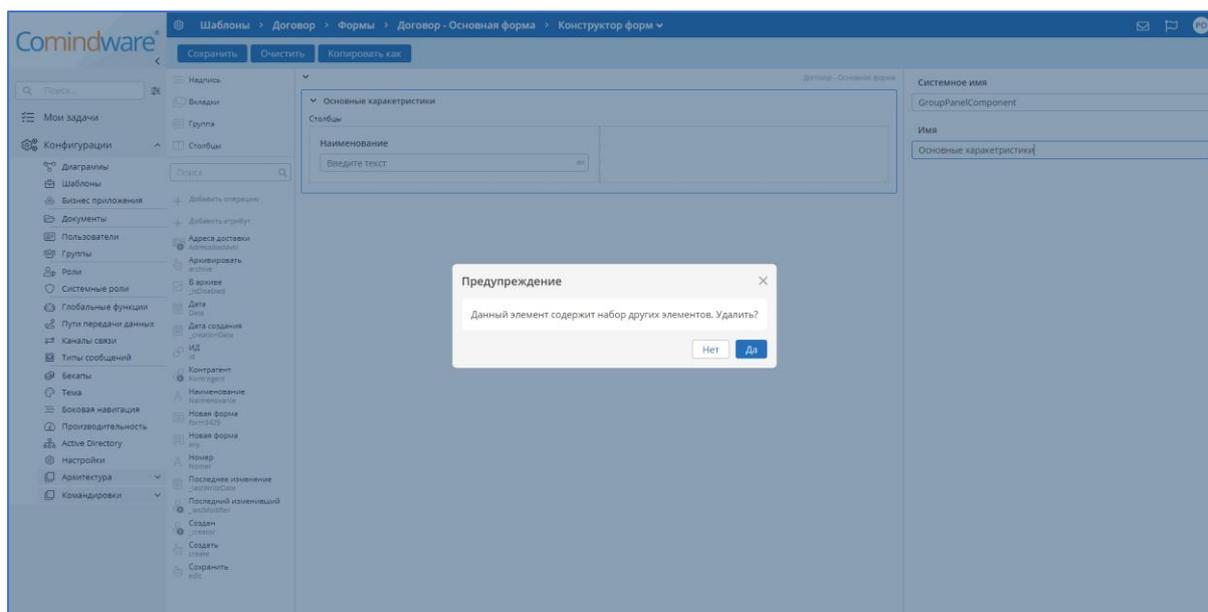


Рис. 60 Окно подтверждения удаления столбца

На форме редактирования в пользовательском интерфейсе элемент «Группа» выглядит следующим образом.

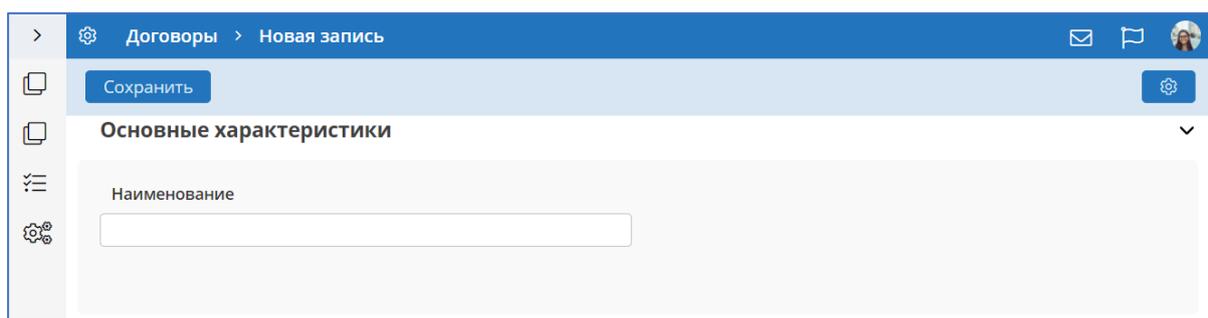


Рис. 61 Элемент «Группа» на пользовательском интерфейсе

Группа имеет свойство сворачиваться и разворачиваться, тем самым предоставляя возможность скрывать и отображать элементы группы. Сворачивать и разворачивать можно с помощью щелчка левой кнопкой мыши на заголовок группы.

Нужно иметь в виду, что в свернутых и скрытых группах не отображаются незаполненные обязательные атрибуты.

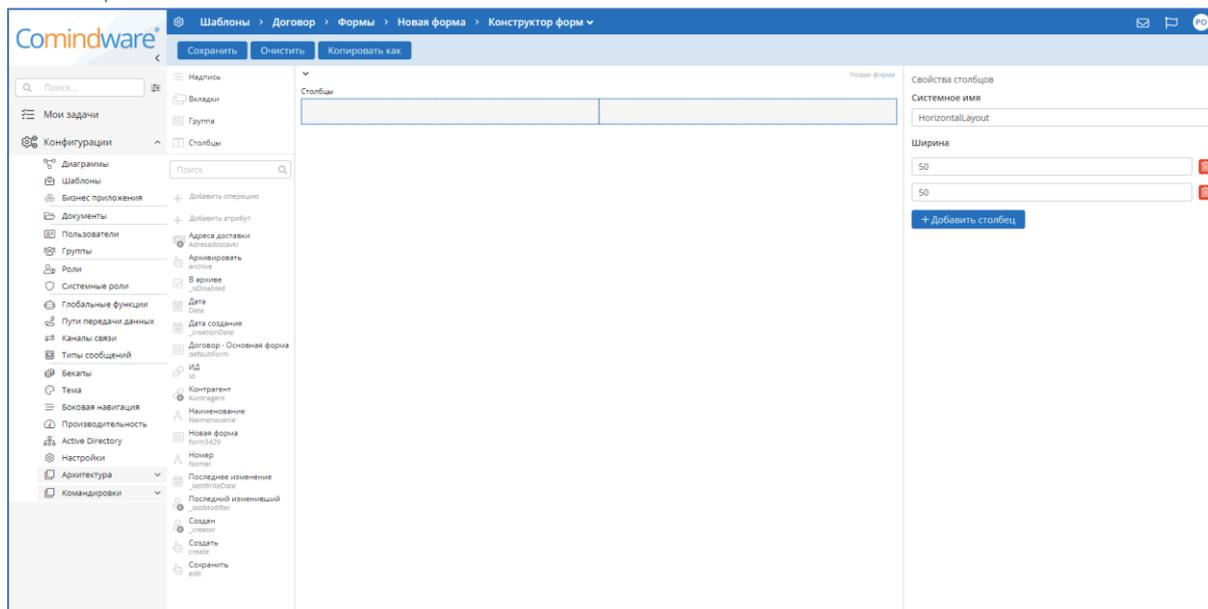


Рис. 62 Элемент «Столбцы»

Визуальный элемент-контейнер, позволяющий отображать другие элементы формы, выстроенные в виде колонок заданного размера и в заданном количестве.

Элемент «Столбцы» имеет следующие характеристики:

- Системное имя – уникальное текстовое системное имя элемента в рамках настраиваемой формы редактирования, используемое для настройки динамических правил на формы редактирования. Может содержать только буквы латинского алфавита, цифры и знак «_».
- Ширина – по умолчанию задана ширина каждого столбца «50».

С помощью кнопки «Добавить столбец» можно добавлять столбцы. В характеристике «Ширина» можно определять размер колонки в процентах. Столбцы отображаются в том порядке, в котором они создаются.

Кнопка в виде корзинки удаляет существующий столбец. Также столбец можно удалить путем выделения его в рабочей области конструктора форм и перетаскиванием мышью в область панели элементов конструктора форм. При удалении колонки, если в ней находились какие-то вложенные визуальные элементы, платформа выдаст предупреждение с возможностью подтверждения действия или отмены.

При попытке удаления всего элемента, если хоть в одной колонке присутствуют вложенные элементы, то платформа выдает аналогичное предупреждение.

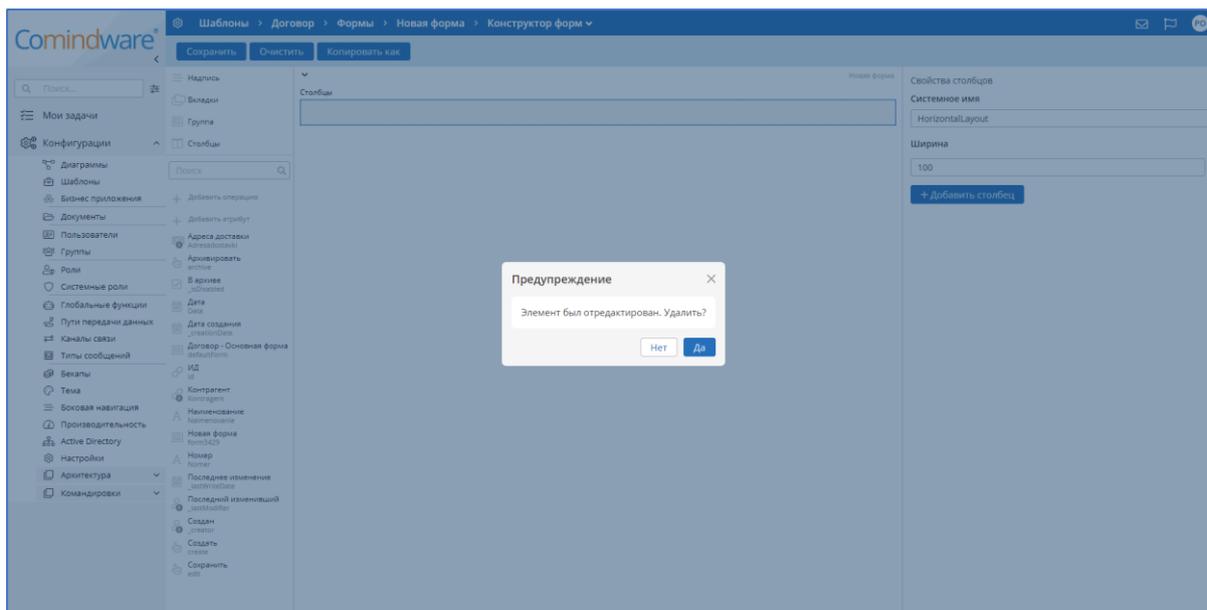


Рис. 63 Окно подтверждения удаления столбца

При попытке удаления элемента Столбцы, если хоть в одной колонке присутствуют вложенные элементы, платформа выдает аналогичное предупреждение.

Подпись «Столбцы» в рабочей области конструктора форм не отображается на пользовательском интерфейсе формы редактирования.

На форме редактирования в пользовательском интерфейсе элемент Столбцы выглядит следующим образом:

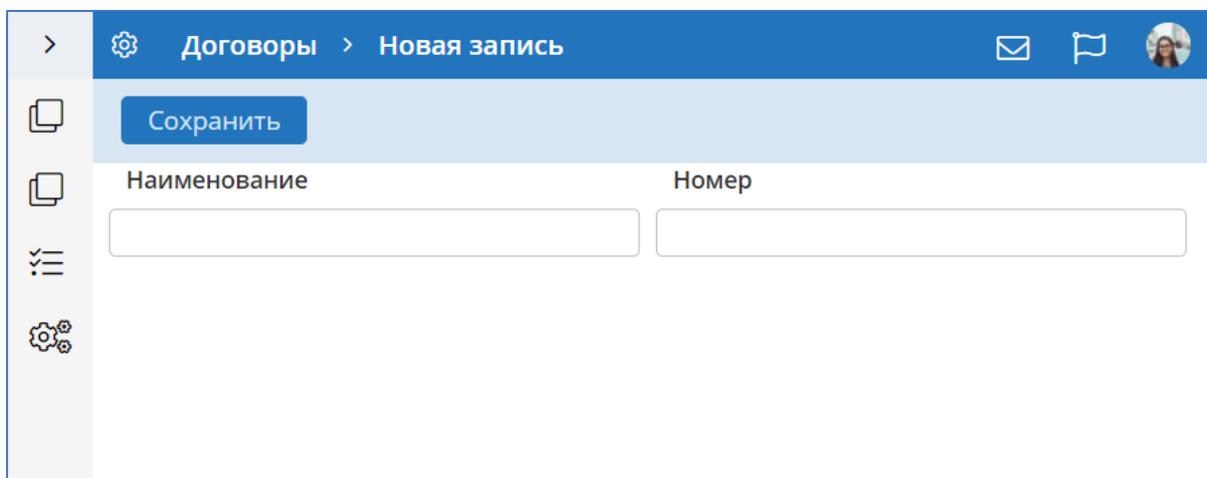
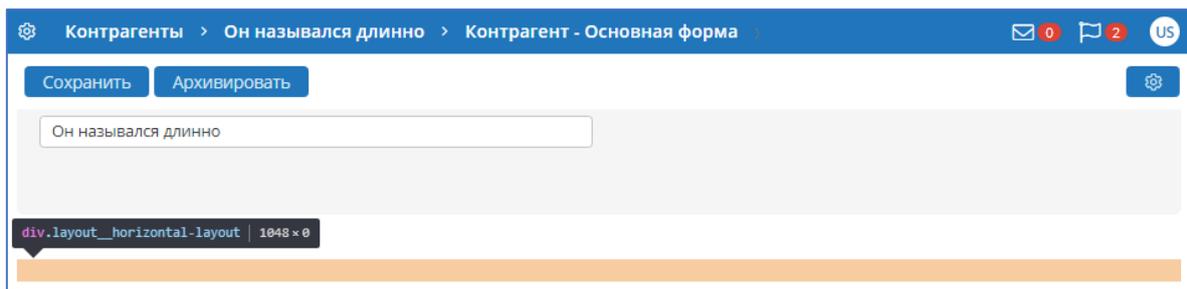


Рис. 64 Отображение столбцов на форме

Если элемент «Столбцы» оказывается незаполненным на конструкторе форм, то на форме редактирования он будет занимать свободное пространство в виде одной строки.



Динамические элементы представлены следующим набором элементов:

- Атрибут – позволяет вынести на форму поле, тип ввода данных которого определяется типом добавляемого на форму атрибута.
- Операция – позволяет вынести на форму кнопку, при нажатии на которую будет выполняться операция, добавляемая на форму.
- Форма редактирования – позволяет вынести на форму редактирования настроенную вложенную форму редактирования. Таким образом обеспечивается повторное использование настроенных форм редактирования в качестве стандартных блоков.

Ниже приведено детальное описание динамических элементов формы редактирования.

Атрибут

Подробное описание типов атрибутов и их назначения приведено в разделе «Настройка атрибутов» (2.3).

Панель элементов (Рис. 50) содержит набор системных и пользовательских атрибутов текущего шаблона записи.

Атрибуты, добавленные на рабочее поля конструктора формы, имеют свое визуальное представление на форме в зависимости от типа (текстовый, числовой, дата и т.д).

На панели настроек элемента отображается информация о выбранном атрибуте.

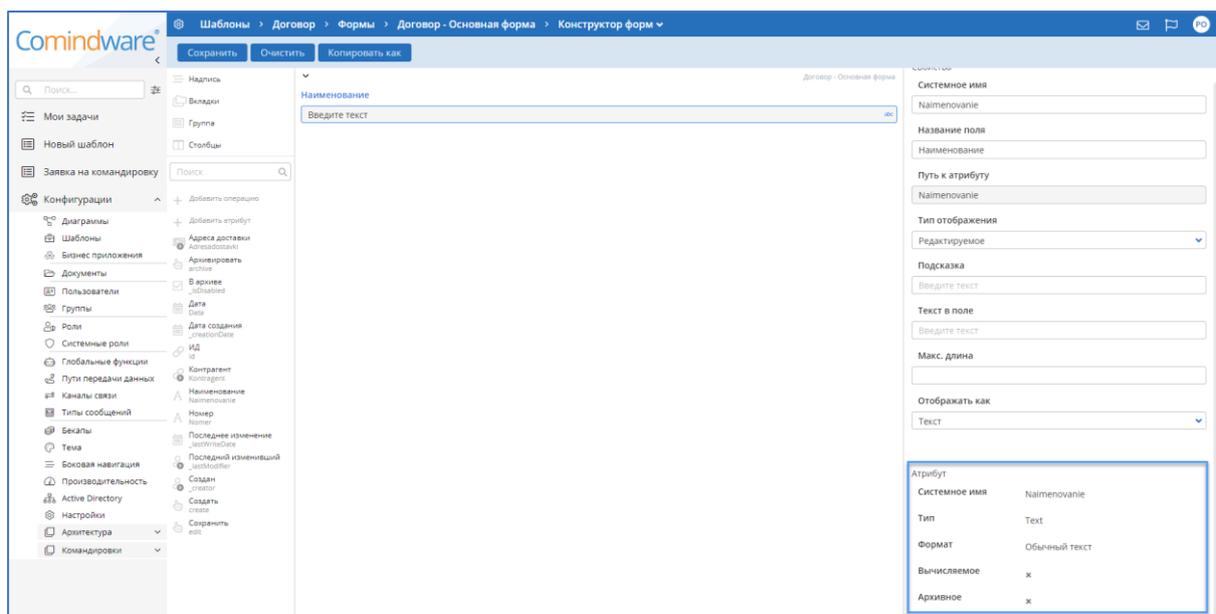


Рис. 65 Информация об атрибуте на панели настройки элемента

Рассмотрим каждый тип атрибута в отдельности.

Текст

Внешний вид в конструкторе форм.

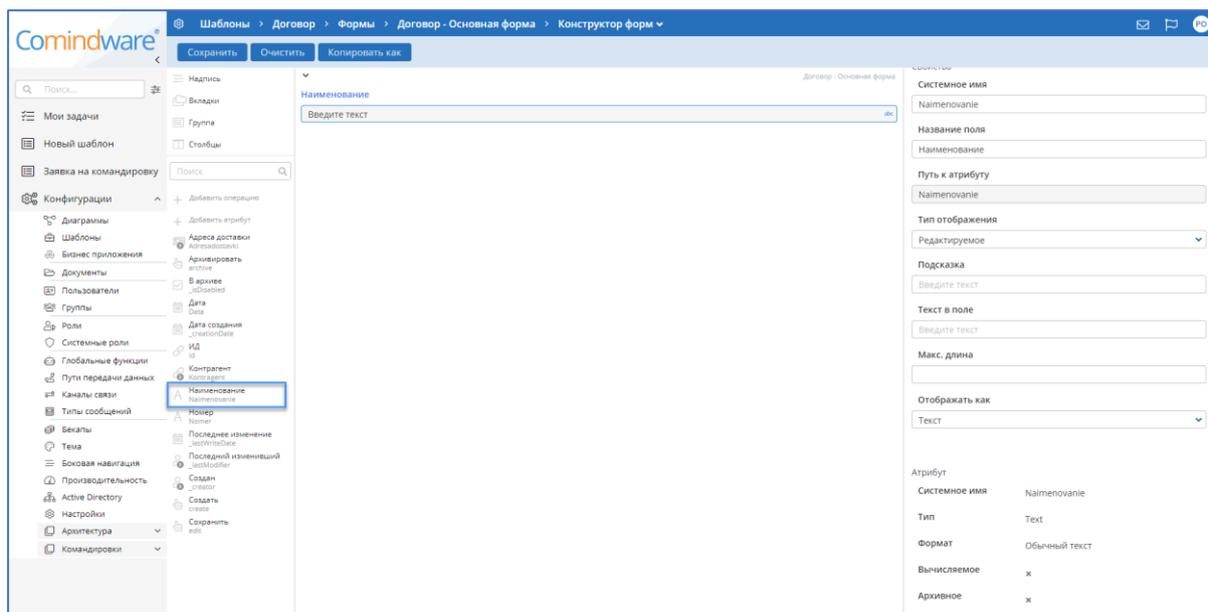


Рис. 66 Внешний вид атрибута типа «Текст» в конструкторе форм

Панель настроек атрибута типа Текст имеет следующие настройки:

- Системное имя – уникальное текстовое системное имя элемента в рамках настраиваемой формы редактирования, используемое для настройки динамических правил на формы редактирования. Может содержать только буквы латинского алфавита, цифры и знак «_».
- Название поля – отображаемая на интерфейсе пользователя подпись к полю.
- Путь к атрибуту – показывает полный путь к атрибуту.
 - Если на форму вынесен атрибут текущего шаблона записи, тогда в текущей характеристике отображается системное имя атрибута.
 - Если атрибут выбран из связанной сущности, то характеристика покажет полный путь до атрибута, например, вынесение описания получателя товара по выбранному адресу поставки.

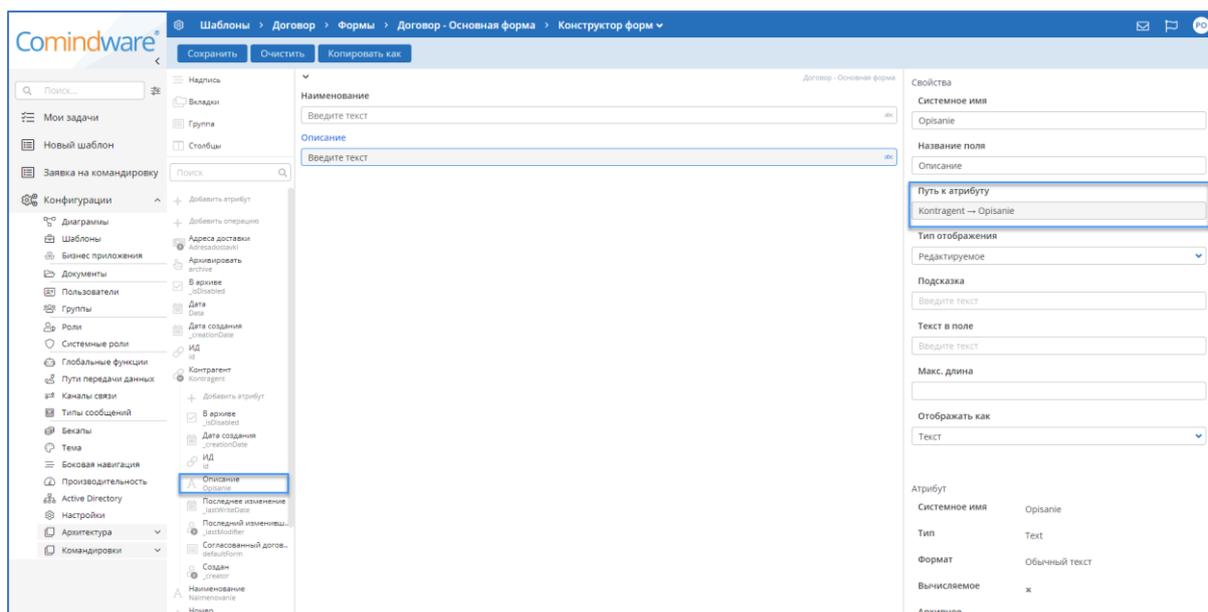


Рис. 67 Внешний вид сложного пути к атрибуту

- Тип отображения – показывает, каким образом данное поле должно быть доступно пользователю на форме редактирования. Имеет 4 состояния:
 - Обязательное – поле доступно для редактирования и обязательно для заполнения пользователем. Неактуально для вычисляемых и системных атрибутов.
 - Редактируемое – поле доступно для редактирования. Неактуально для вычисляемых и системных атрибутов.
 - Только для чтения – поле недоступно для редактирования и необязательно.
 - Скрытое – поле скрыто для пользователя на форме редактирования.
- Подсказка – текстовая информация, которая отображается пользователю при щелчке на знак вопроса, расположенный рядом с подписью поля, в котором определена подсказка.
- Текст в поле – текстовая информация, которая отображается пользователю в поле полупрозрачным цветом, если поле не заполнено.
- Макс. длина – указывается максимальная длина символов, которые пользователь сможет ввести в данное поле. Значение «0» означает, что поле не ограничено по длине. При установке максимальной длины, текущие данные открытой на редактирование записи на форме не будут обрезаны. Пользователю при этом будет доступно редактирование поля без возможности добавлять символы, если длина уже существующей текстовой строки окажется больше ограничения.

Данная характеристика доступна для атрибута типа «Текст» с видом отображения «Обычный текст».

- Отображать как – показывает, каким образом отображать текстовое поле.
 - Текст – отображается простое текстовое поле без возможности изменения высоты поля.

- Область текста – отображается как текстовый блок с фиксированной или динамической высотой. Если в блок с фиксированной высотой сохраняемый текст не помещается по высоте, то в блоке появляется вертикальная полоса прокрутки.

Данная характеристика доступна для атрибута типа «Текст» с видом отображения «Обычный текст». Для атрибута с видом отображения «HTML текст» характеристика «Отображать как» принимает значение «Область текста».

- Высота текста в строках – указывается высота текстового блока в строковых значениях. Доступны следующие значения:
 - Авто – динамическое значение высоты текстового блока в зависимости от содержимого.
 - 5, 10, 20 – фиксированное значение высоты текстового блока в строках.

Данная характеристика доступна для атрибута типа «Текст» с видом отображения «HTML текст» (Рис. 29), а также с видом отображения «Обычный текст» и значением «Область текста» характеристики поля «Отображать как».

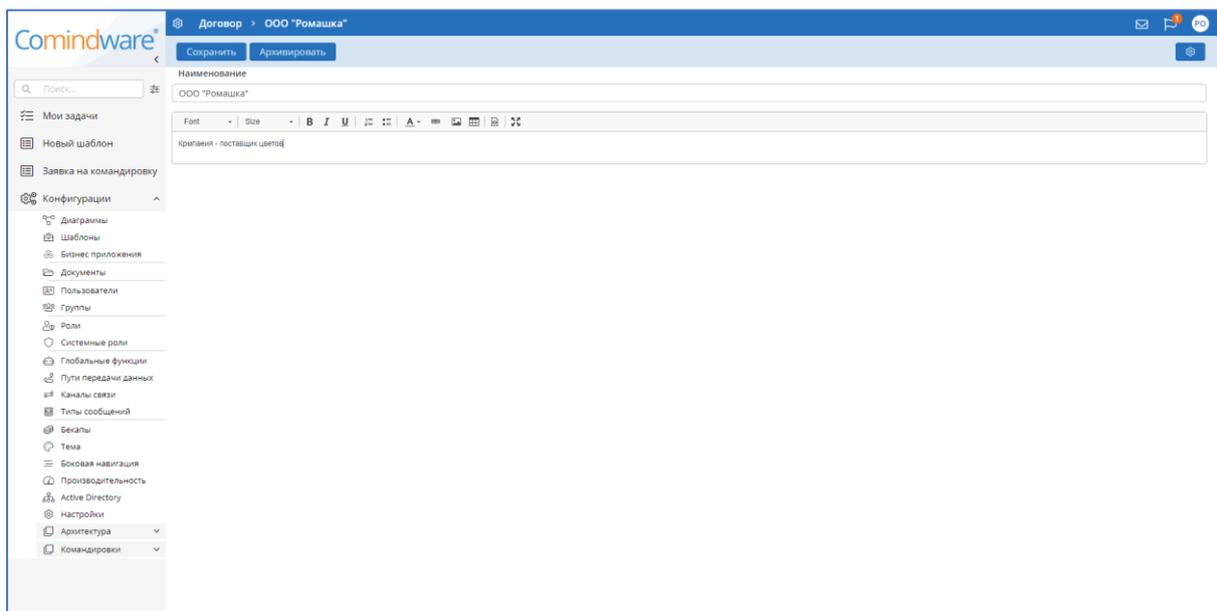


Рис. 68 Внешний вид поля атрибута типа «Текст» на форме редактирования в фокусе

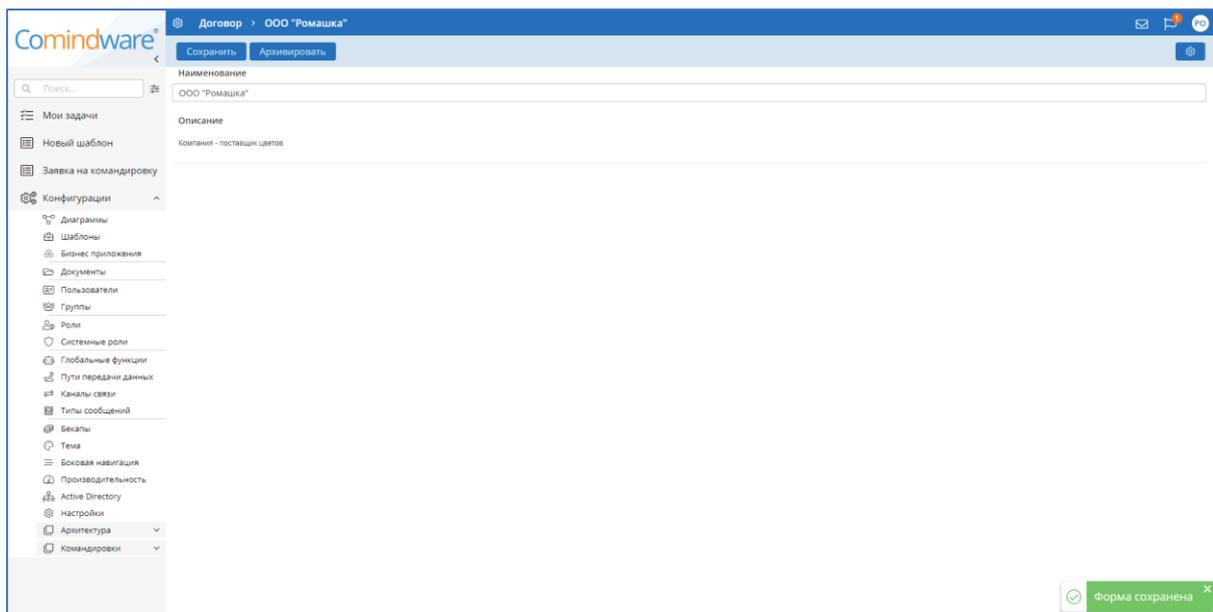


Рис. 69 Внешний вид поля атрибута типа «Текст» на форме редактирования вне фокуса

Редактируемое поле с отображением в виде текстового блока по умолчанию на форме отображается в виде статичного форматированного (или неформатированного) текста. При щелчке левой кнопкой мыши в блок данных (пустое место или текст) поле получает фокус и становится активным элементом редактирования.

Поле с атрибутом, имеющим формат отображения «HTML текст», имеет активные элементы управления форматированием текста.

Число

Внешний вид в конструкторе форм.

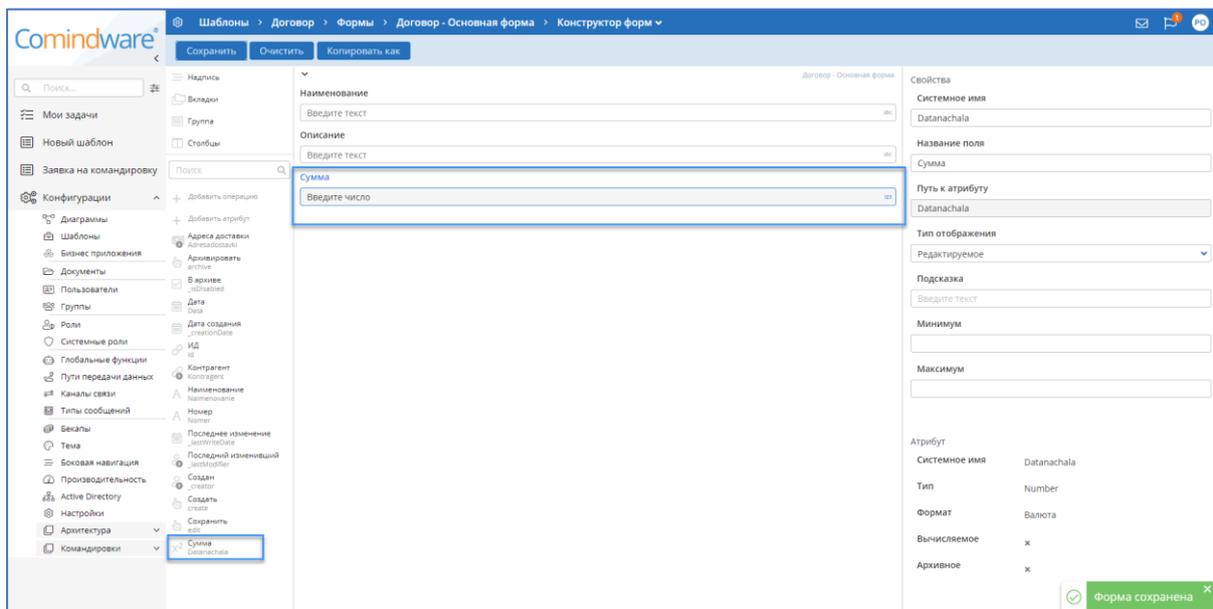


Рис. 70 Внешний вид атрибута типа «Число» на конструкторе форм

Панель настроек атрибута типа «Число» имеет следующие настройки:

- Системное имя – уникальное текстовое системное имя элемента в рамках настраиваемой формы редактирования, используемое для настройки динамических

правил на формы редактирования. Может содержать только буквы латинского алфавита, цифры и знак «_».

- Название поля – отображаемая на интерфейсе пользователя подпись к полю.
- Путь к атрибуту – показывает полный путь к атрибуту. Если на форму вынесен атрибут текущего шаблона записи, тогда в текущей характеристике отображается системное имя атрибута. Если атрибут выбран из связанной сущности, то характеристика покажет полный путь до атрибута, например, вынесение описания получателя товара по выбранному адресу поставки:

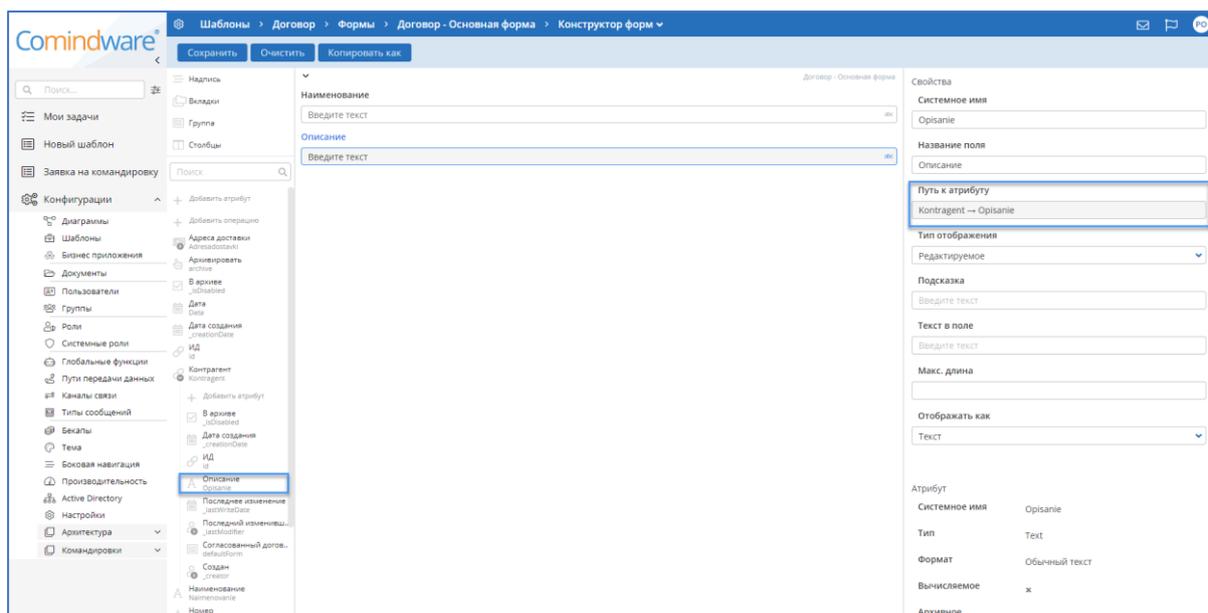


Рис. 71 Внешний вид сложного пути к атрибуту

- Тип отображения – показывает, каким образом данное поле должно быть доступно пользователю на форме редактирования. Имеет 4 состояния:
 - Обязательное – поле доступно для редактирования и обязательно для заполнения пользователем. Неактуально для вычисляемых и системных атрибутов.
 - Редактируемое – поле доступно для редактирования. Неактуально для вычисляемых и системных атрибутов.
 - Только для чтения – поле недоступно для редактирования и необязательно.
 - Скрытое – поле скрыто для пользователя на форме редактирования.
- Подсказка – текстовая информация, которая отображается пользователю, при щелчке на знак вопроса, расположенный рядом с подписью поля, в котором определена подсказка.
- Минимум – указывается минимальное числовое значение, которое пользователь сможет ввести в поле на форме редактирования.
- Максимум – указывается максимальное числовое значение, которое пользователь сможет ввести в поле на форме редактирования.

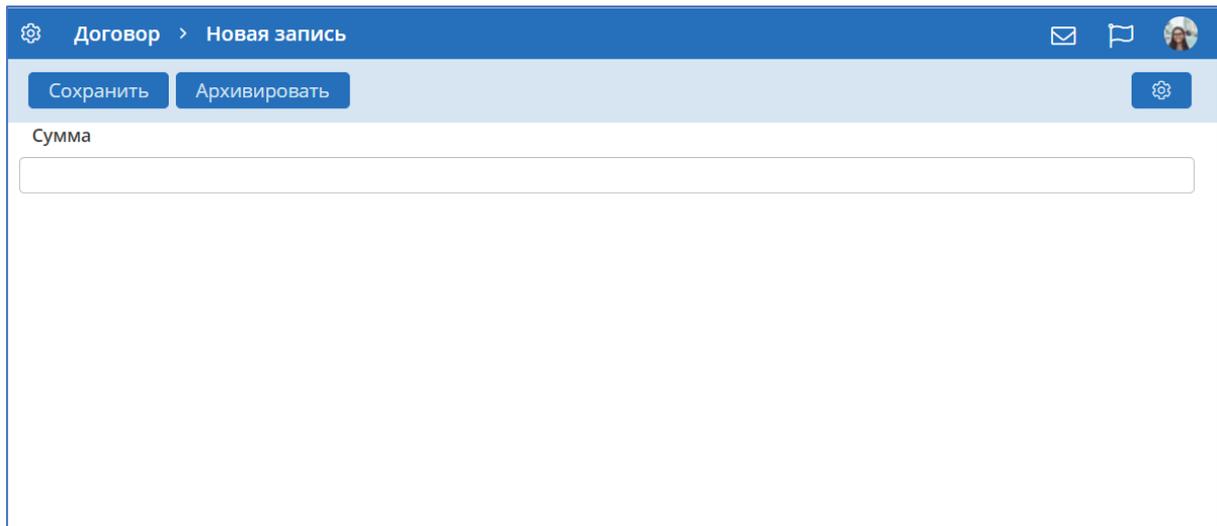


Рис. 72 Внешний вид поля атрибута типа «Число» на форме редактирования

Поле с атрибутом типа «Число» отображается на форме редактирования пользователя в зависимости от формата отображения.

- Если выбран формат отображения «Целое», то пользователю невозможно будет ввести числа после запятой.
- Если формат отображения указан «Валюта», то после запятой возможно ввести два знака.
- Если формат отображения «Десятичное», то после запятой возможно ввести три знака.

Длительность

Внешний вид в конструкторе форм.

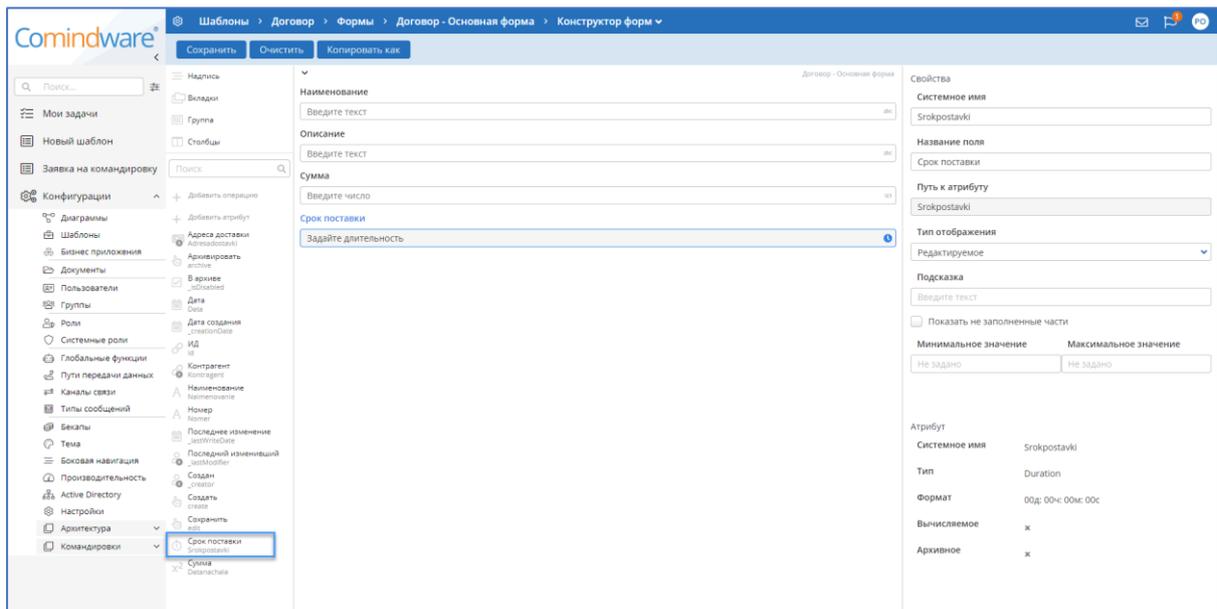


Рис. 73 Внешний вид атрибута типа «Длительность» в конструкторе форм

Панель настроек атрибута типа «Длительность» имеет следующие настройки:

- Системное имя – уникальное текстовое системное имя элемента в рамках настраиваемой формы редактирования, используемое для настройки динамических правил на формы редактирования. Может содержать только буквы латинского алфавита, цифры и знак «_».
- Название поля – отображаемая на интерфейсе пользователя подпись к полю.
- Путь к атрибуту – показывает полный путь к атрибуту. Если на форму вынесен атрибут текущего шаблона записи, тогда в текущей характеристике отображается системное имя атрибута. Если атрибут выбран из связанной сущности, то характеристика покажет полный путь до атрибута, например, вынесение описания получателя товара по выбранному адресу поставки:

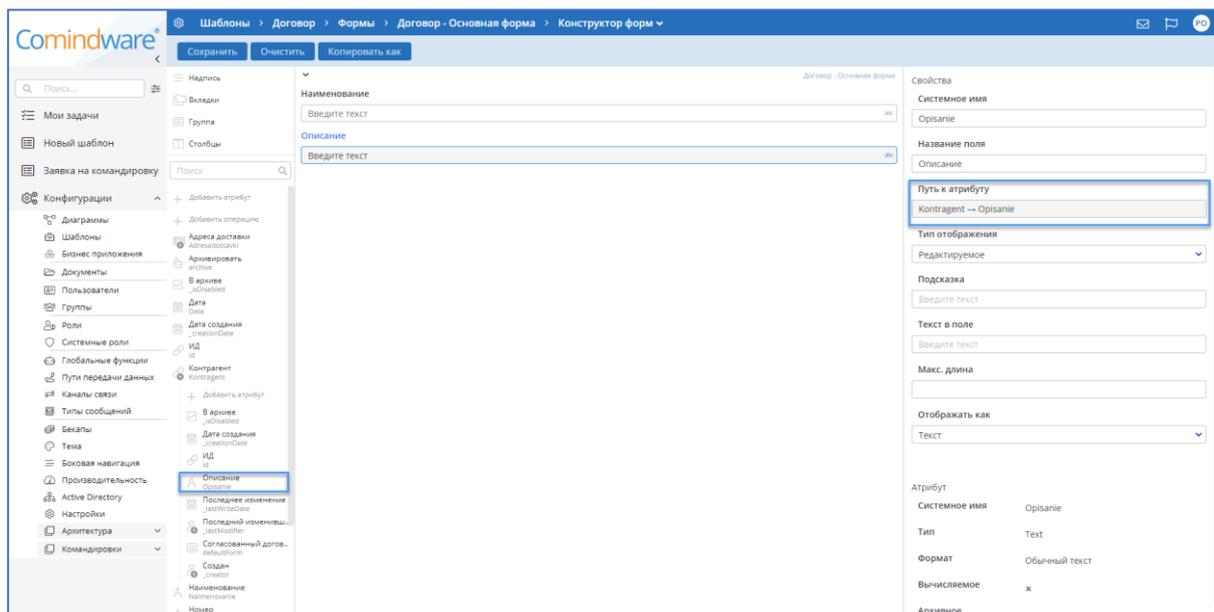


Рис. 74 Внешний вид сложного пути к атрибуту

- Тип отображения – показывает, каким образом данное поле должно быть доступно пользователю на форме редактирования. Имеет 4 состояния:
 - Обязательное – поле доступно для редактирования и обязательно для заполнения пользователем. Неактуально для вычисляемых и системных атрибутов.
 - Редактируемое – поле доступно для редактирования. Неактуально для вычисляемых и системных атрибутов.
 - Только для чтения – поле недоступно для редактирования и необязательно.
 - Скрытое – поле скрыто для пользователя на форме редактирования.
- Подсказка – текстовая информация, которая отображается пользователю, при щелчке на знак вопроса, расположенный рядом с подписью поля, в котором определена подсказка.
- Минимальное значение – минимальное значение длительности, которое можно ввести на форме пользователем.
- Максимальное значение – максимальное значение длительности, которое можно будет ввести на форме пользователем.

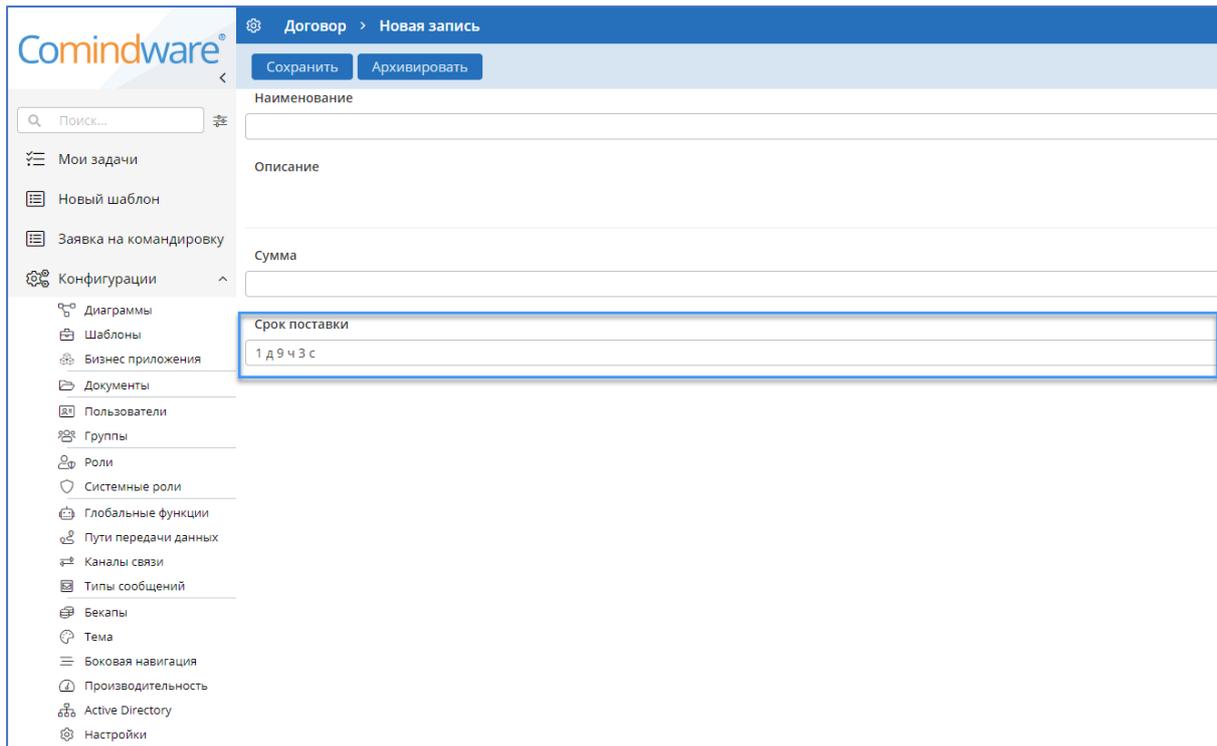


Рис. 75 Внешний вид поля атрибута типа «Длительность» на форме редактирования

Поле с атрибутом типа «Длительность» отображается на форме редактирования пользователя в зависимости от формата отображения. В зависимости от формата отображения пользователю будут доступны соответствующие разряды в поле атрибута «Длительность».

Дата / Время

Внешний вид в конструкторе форм.

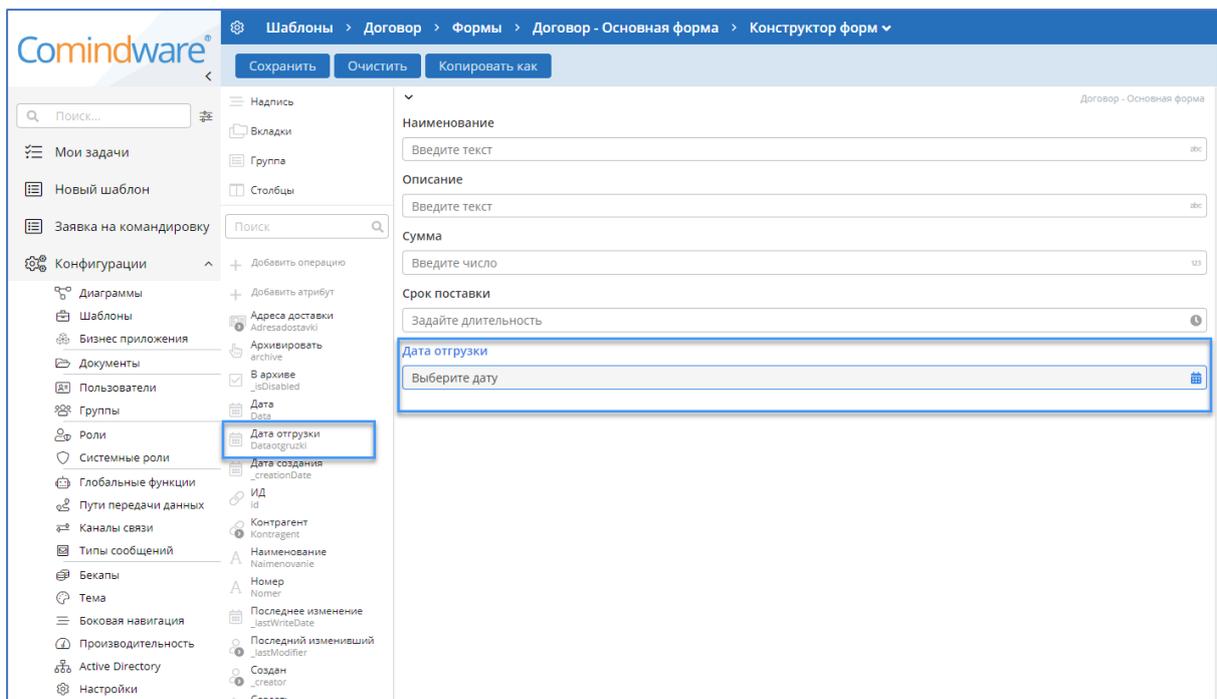


Рис. 76 Внешний вид атрибута типа «Дата / Время» в конструкторе форм

Панель настроек атрибута типа «Дата / время» имеет следующие настройки:

- Системное имя – уникальное текстовое системное имя элемента в рамках настраиваемой формы редактирования, используемое для настройки динамических правил на формы редактирования. Может содержать только буквы латинского алфавита, цифры и знак «_».
- Название поля – отображаемая на интерфейсе пользователя подпись к полю.
- Путь к атрибуту – показывает полный путь к атрибуту. Если на форму вынесен атрибут текущего шаблона записи, тогда в текущей характеристике отображается системное имя атрибута. Если атрибут выбран из связанной сущности, то характеристика покажет полный путь до атрибута, например, вынесение описания получателя товара по выбранному адресу поставки:

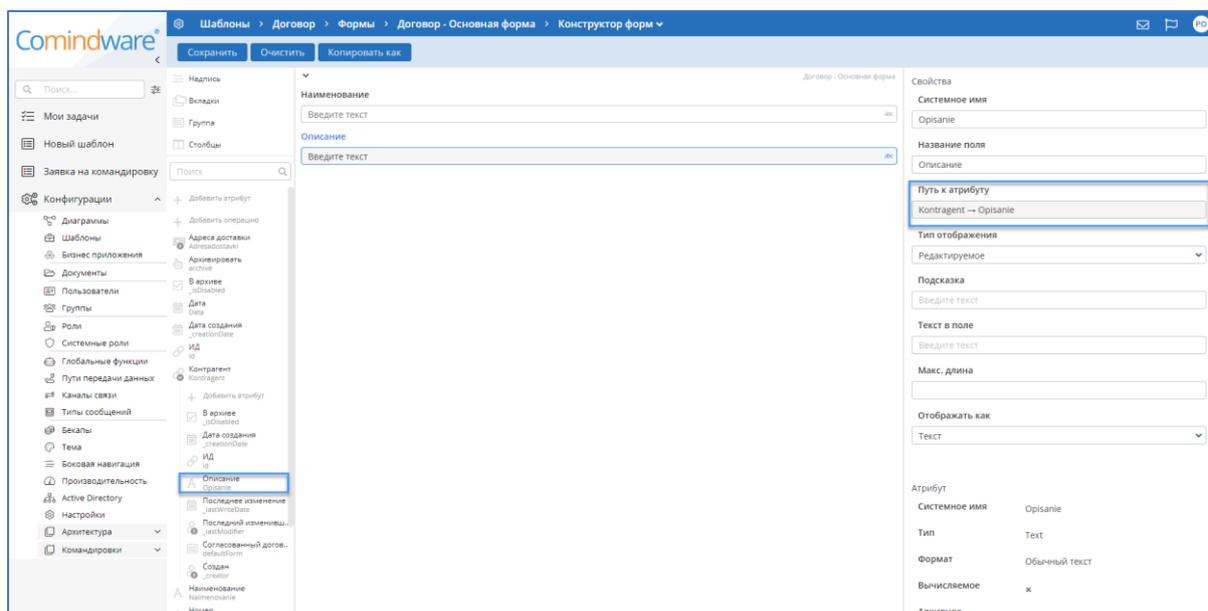


Рис. 77 Внешний вид сложного пути к атрибуту

- Тип отображения – показывает, каким образом данное поле должно быть доступно пользователю на форме редактирования. Имеет 4 состояния:
 - Обязательное – поле доступно для редактирования и обязательно для заполнения пользователем. Неактуально для вычисляемых и системных атрибутов.
 - Редактируемое – поле доступно для редактирования. Неактуально для вычисляемых и системных атрибутов.
 - Только для чтения – поле недоступно для редактирования и необязательно.
 - Скрытое – поле скрыто для пользователя на форме редактирования.
- Подсказка – текстовая информация, которая отображается пользователю, при щелчке на знак вопроса, расположенный рядом с подписью поля, в котором определена подсказка.

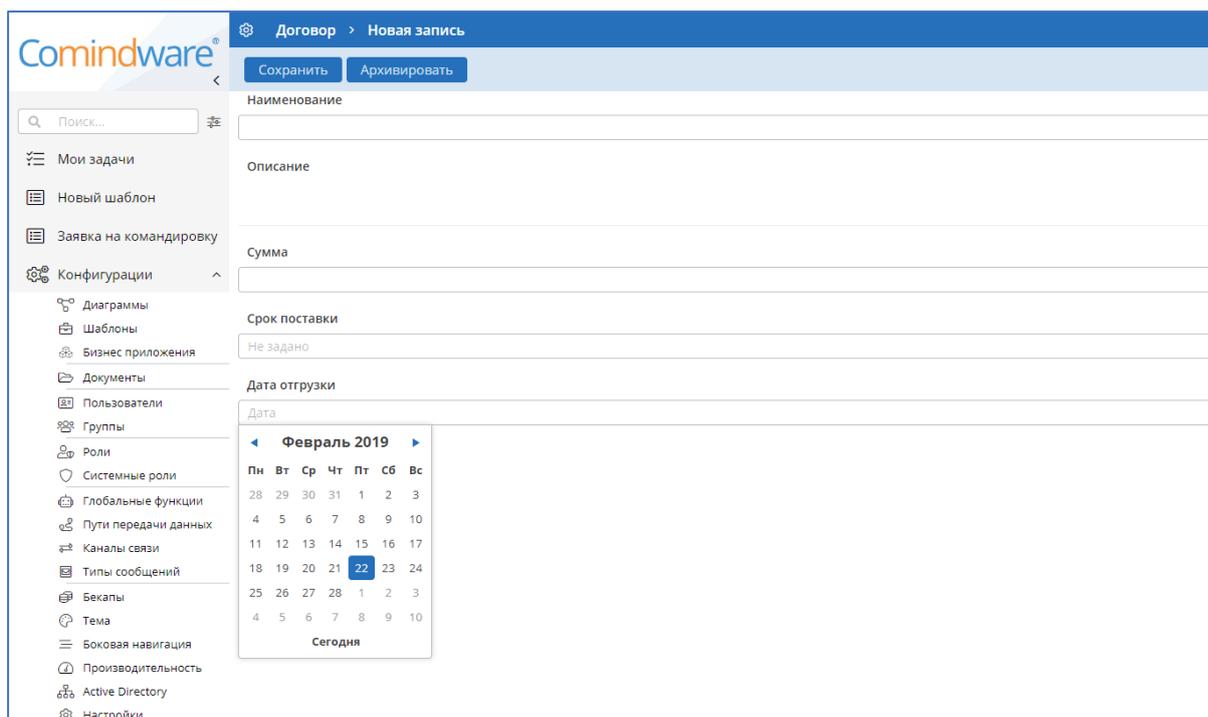


Рис. 78 Внешний вид поля атрибута типа «Дата / Время» на форме редактирования

Поле с атрибутом типа «Дата / время» отображается на форме редактирования пользователя в зависимости от формата отображения. В зависимости от формата отображения пользователю будут доступны выбор даты и времени.

Логический

Внешний вид в конструкторе форм.

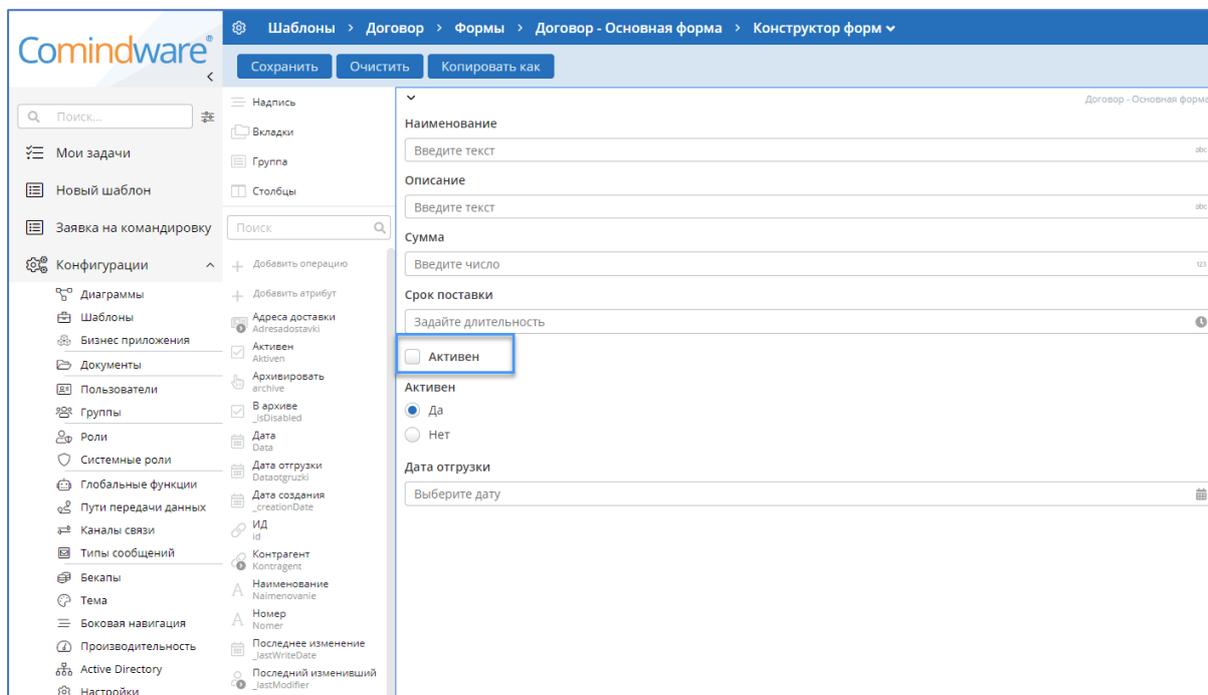


Рис. 79 Внешний вид атрибута типа «Логический» в конструкторе форм (Флажок)

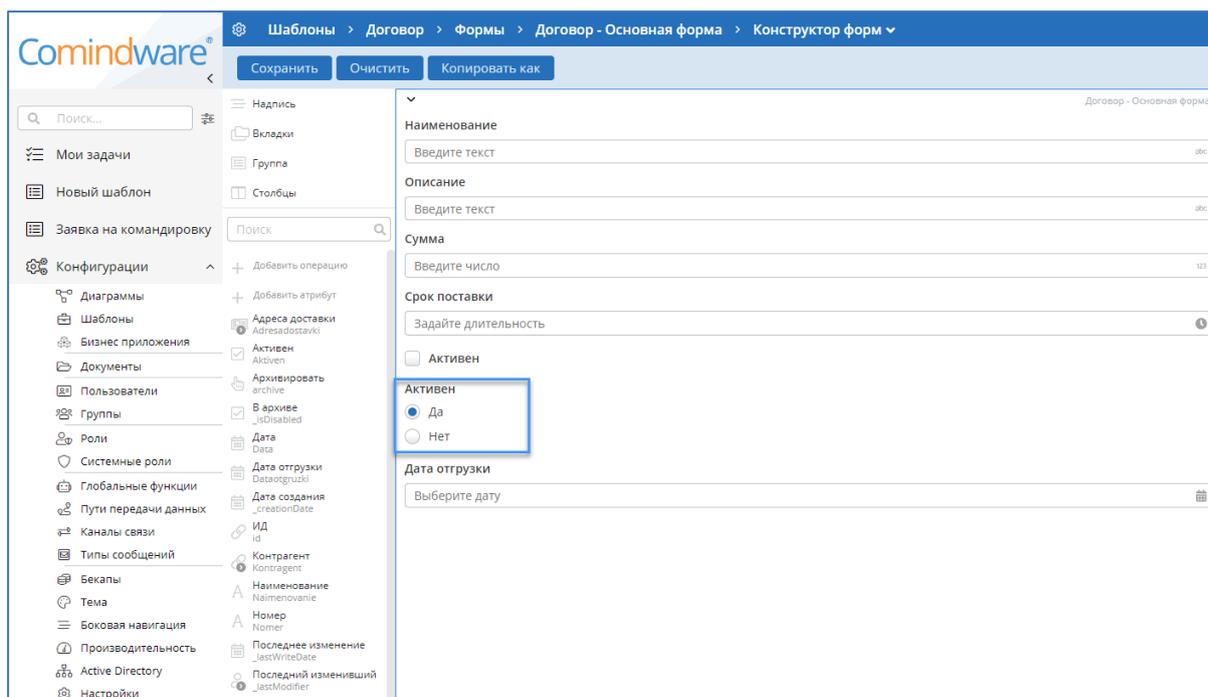


Рис. 80 Внешний вид атрибута типа «Логический» в конструкторе форм («Переключатель»)

Панель настроек атрибута типа «Логический» имеет следующие настройки:

- Системное имя – уникальное текстовое системное имя элемента в рамках настраиваемой формы редактирования, используемое для настройки динамических правил на формы редактирования. Может содержать только буквы латинского алфавита, цифры и знак «_».
- Название поля – отображаемая на интерфейсе пользователя подпись к полю.
- Путь к атрибуту – показывает полный путь к атрибуту. Если на форму вынесен атрибут текущего шаблона записи, тогда в текущей характеристике отображается системное имя атрибута. Если атрибут выбран из связанной сущности, то характеристика покажет полный путь до атрибута, например, вынесение описания получателя товара по выбранному адресу поставки:

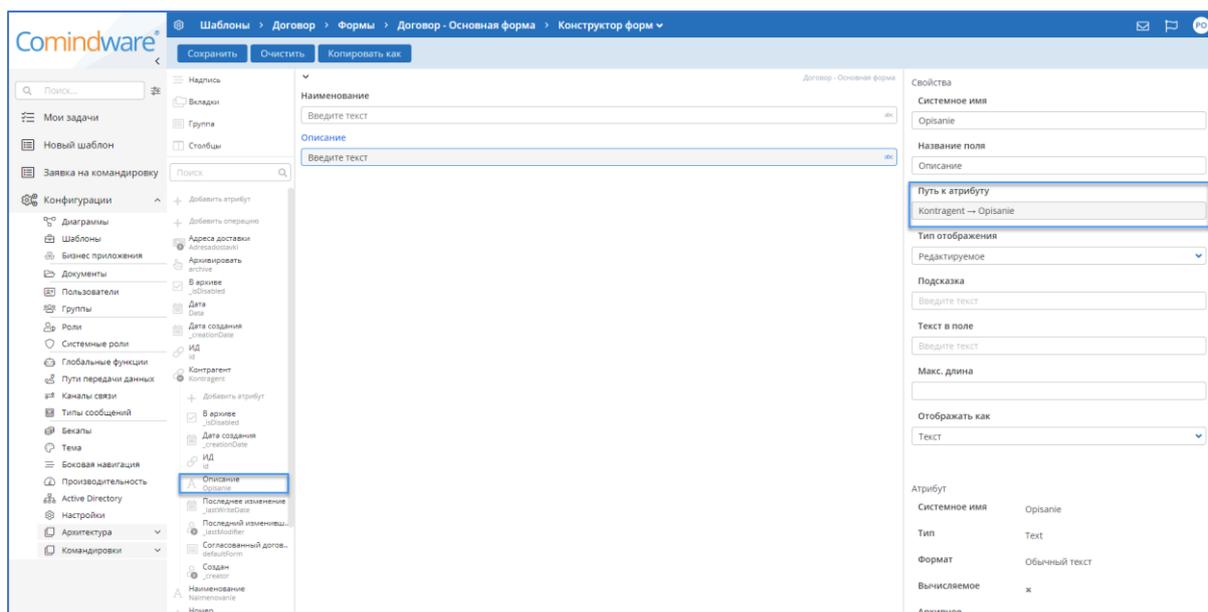


Рис. 81 Внешний вид сложного пути к атрибуту

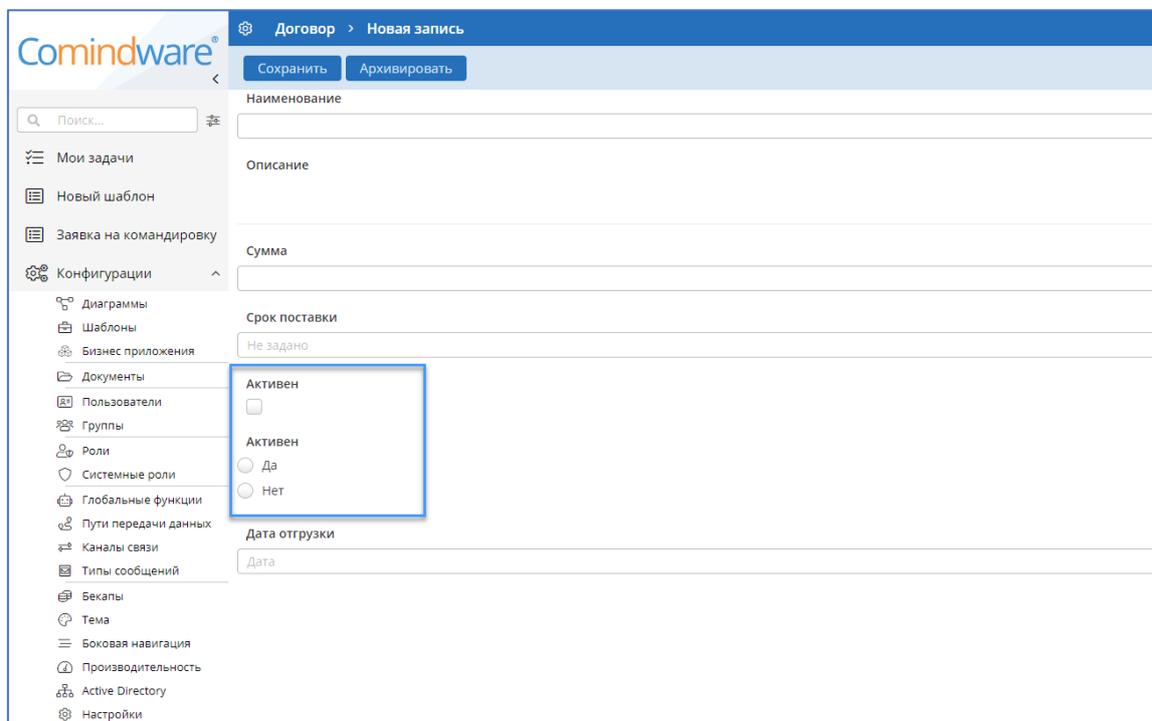
- Тип отображения – показывает, каким образом данное поле должно быть доступно пользователю на форме редактирования. Имеет 4 состояния:
 - Обязательное – поле доступно для редактирования и обязательно для заполнения пользователем. Неактуально для вычисляемых и системных атрибутов.
 - Редактируемое – поле доступно для редактирования. Неактуально для вычисляемых и системных атрибутов.
 - Только для чтения – поле недоступно для редактирования и необязательно.
 - Скрытое – поле скрыто для пользователя на форме редактирования.
- Подсказка – текстовая информация, которая отображается пользователю, при щелчке на знак вопроса, расположенный рядом с подписью поля, в котором определена подсказка.
- Отображать как – показывает, каким образом отображать текстовое поле.
 - Флажок – поле отображается в виде флага, который может быть в двух состояниях (активен и неактивен).
 - Переключатель – отображается как переключатель между двух значений.
- Изменить названия вариантов – флаг, который позволяет изменить подписи у двух значений, если выбран режим отображения «Переключатель».

Данная характеристика доступна для атрибута типа «Логический», если в характеристике «Отображать как» выбрано значение «Переключатель».

- Истина – текстовая подпись для положительного значения атрибута типа «Логический».
- Ложь – текстовая подпись для отрицательного значения атрибута типа «Логический».

Данные характеристики доступны для атрибута типа «Логический», если в флаг «Изменить названия вариантов» активен. Независимо от отображаемых значений, атрибут типа «Логический» всегда содержит одно значение из двух: Истина (true) или Ложь (false).

Внешний вид на форме редактирования пользователя.

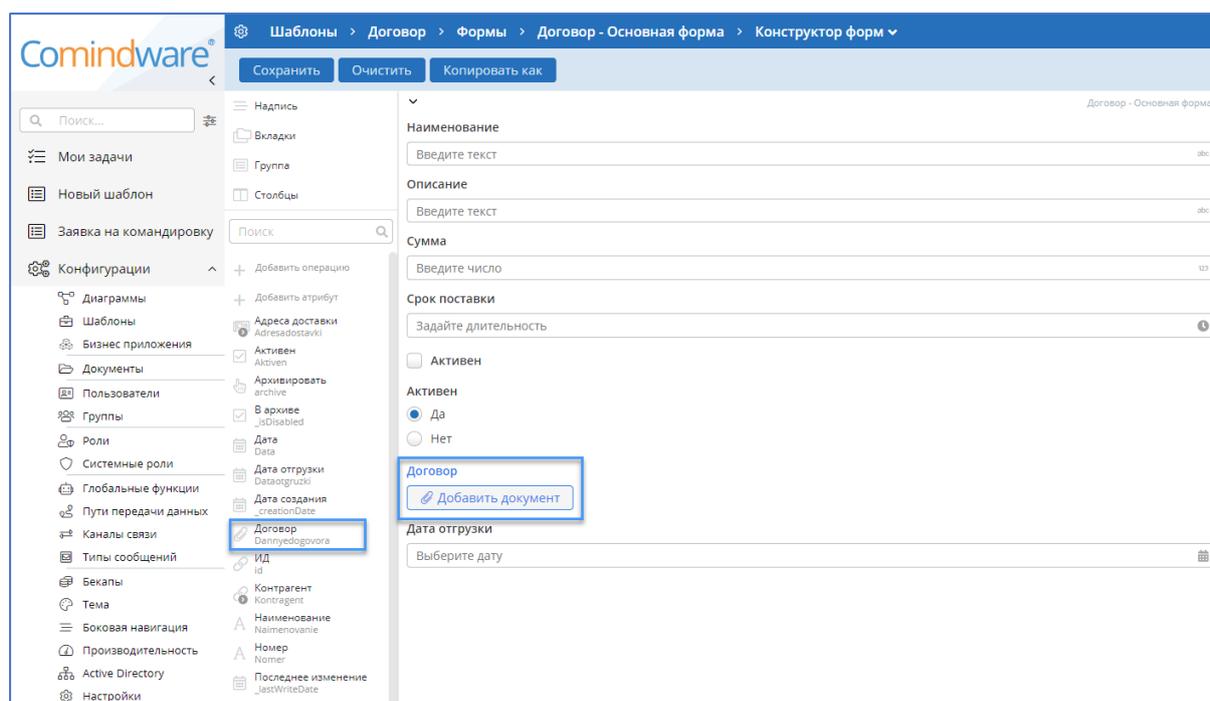


The screenshot shows the 'Comindware' user editing interface. The breadcrumb path is 'Договор > Новая запись'. The main form has fields for 'Наименование', 'Описание', 'Сумма', 'Срок поставки' (with value 'Не задано'), and 'Дата отгрузки' (with value 'Дата'). A dropdown menu is open for the 'Активен' field, showing options: 'Активен' (unchecked), 'Активен' (radio 'Да'), and 'Активен' (radio 'Нет'). The left sidebar contains navigation items like 'Мои задачи', 'Новый шаблон', 'Заявка на командировку', and 'Конфигурации'.

Рис. 82 Внешний вид поля атрибута типа «Логический» на форме редактирования

Документ

Внешний вид в конструкторе форм.



The screenshot shows the 'Comindware' form builder interface. The breadcrumb path is 'Шаблоны > Договор > Формы > Договор - Основная форма > Конструктор форм'. The main form has fields for 'Наименование', 'Описание', 'Сумма', 'Срок поставки' (with value 'Задайте длительность'), and 'Дата отгрузки' (with value 'Выберите дату'). A dropdown menu is open for the 'Активен' field, showing options: 'Активен' (unchecked), 'Активен' (radio 'Да'), and 'Активен' (radio 'Нет'). The left sidebar contains navigation items like 'Надпись', 'Вкладки', 'Группа', 'Столбцы', and 'Конфигурации'. A list of attributes is visible, with 'Договор' (Data: Договор) selected and highlighted.

Рис. 83 Внешний вид атрибута типа «Документ» в конструкторе форм

Панель настроек атрибута типа «Документ» имеет следующие настройки:

- Системное имя – уникальное текстовое системное имя элемента в рамках настраиваемой формы редактирования, используемое для настройки динамических правил на формы редактирования. Может содержать только буквы латинского алфавита, цифры и знак «_».
- Название поля – отображаемая на интерфейсе пользователя подпись к полю.
- Путь к атрибуту – показывает полный путь к атрибуту. Если на форму вынесен атрибут текущего шаблона записи, тогда в текущей характеристике отображается системное имя атрибута. Если атрибут выбран из связанной сущности, то характеристика покажет полный путь до атрибута, например, вынесение описания получателя товара по выбранному адресу поставки:

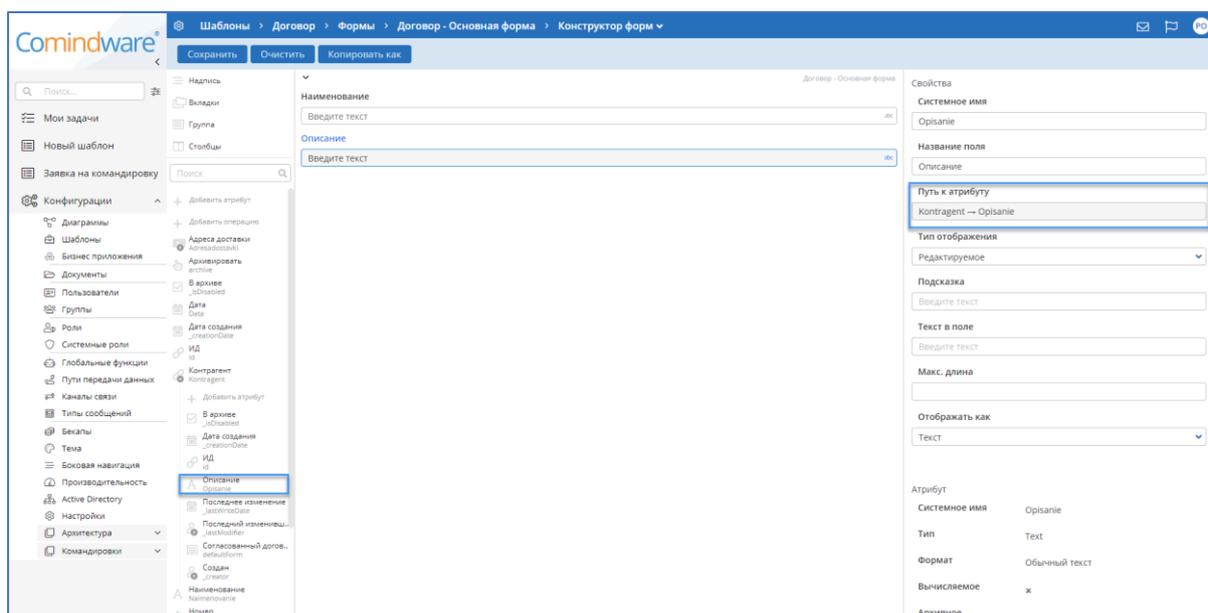


Рис. 84 Внешний вид сложного пути к атрибуту

- Тип отображения – показывает, каким образом данное поле должно быть доступно пользователю на форме редактирования. Имеет 4 состояния:
 - Обязательное – поле доступно для редактирования и обязательно для заполнения пользователем. Неактуально для вычисляемых и системных атрибутов.
 - Редактируемое – поле доступно для редактирования. Неактуально для вычисляемых и системных атрибутов.
 - Только для чтения – поле недоступно для редактирования и необязательно.
 - Скрытое – поле скрыто для пользователя на форме редактирования.
- Подсказка – текстовая информация, которая отображается пользователю, при щелчке на знак вопроса, расположенный рядом с подписью поля, в котором определена подсказка.
- Отображать как – показывает, каким образом отображать текстовое поле.
 - Флажок – поле отображается в виде флага, который может быть в двух состояниях (активен и неактивен).

- Переключатель – отображается как переключатель между двух значений.
- Несколько документов – флаг включает режим, при котором пользователь сможет прикладывать неограниченное количество файлов к атрибуту.
- Разрешить удалять – флаг включает режим работы поля, при котором пользователю будет разрешено удалять документ, если «тип отображения» принимает одно из значений «Обязательное» или «Редактируемое».

Внешний вид на форме редактирования пользователя.

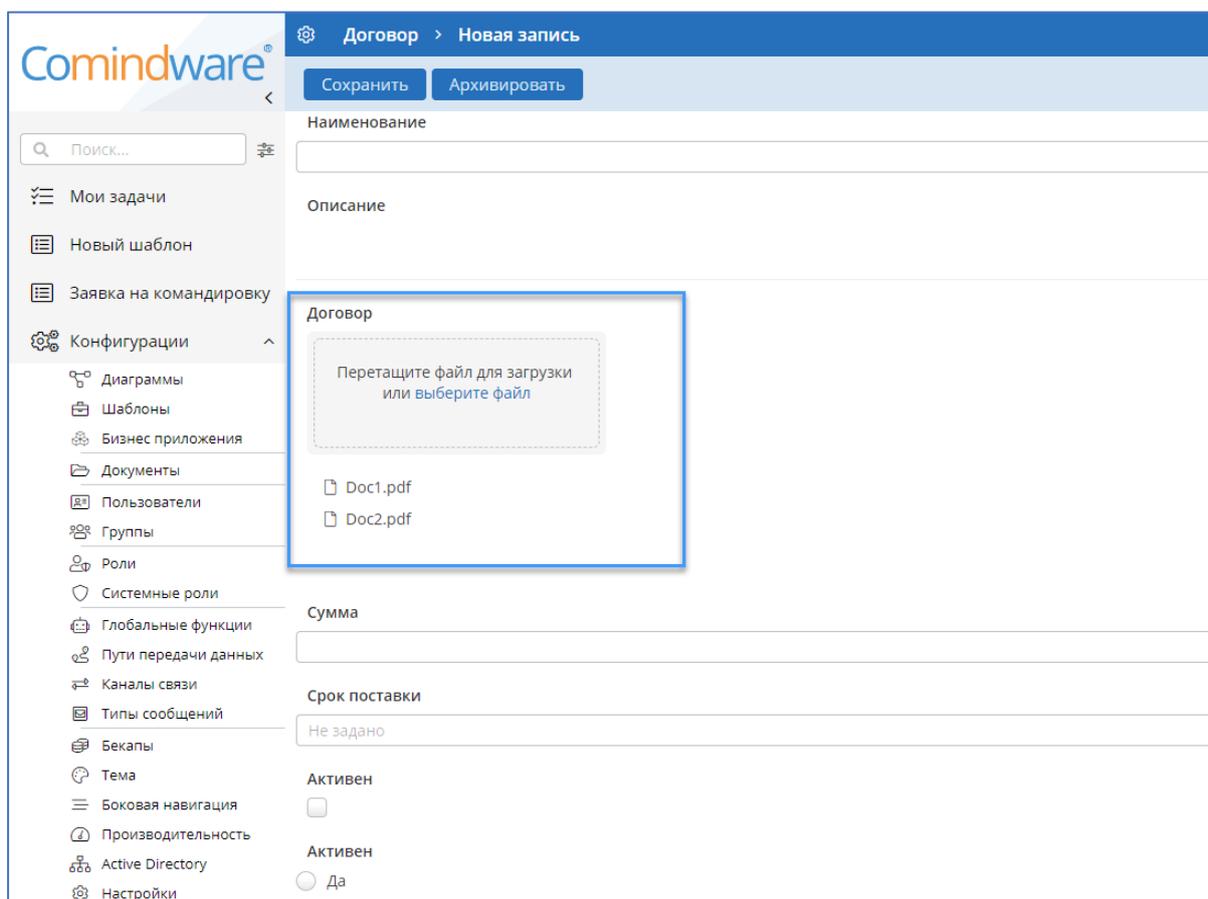


Рис. 85 Внешний вид поля атрибута типа «Документ» на форме редактирования

Пользователь

Внешний вид в конструкторе форм.

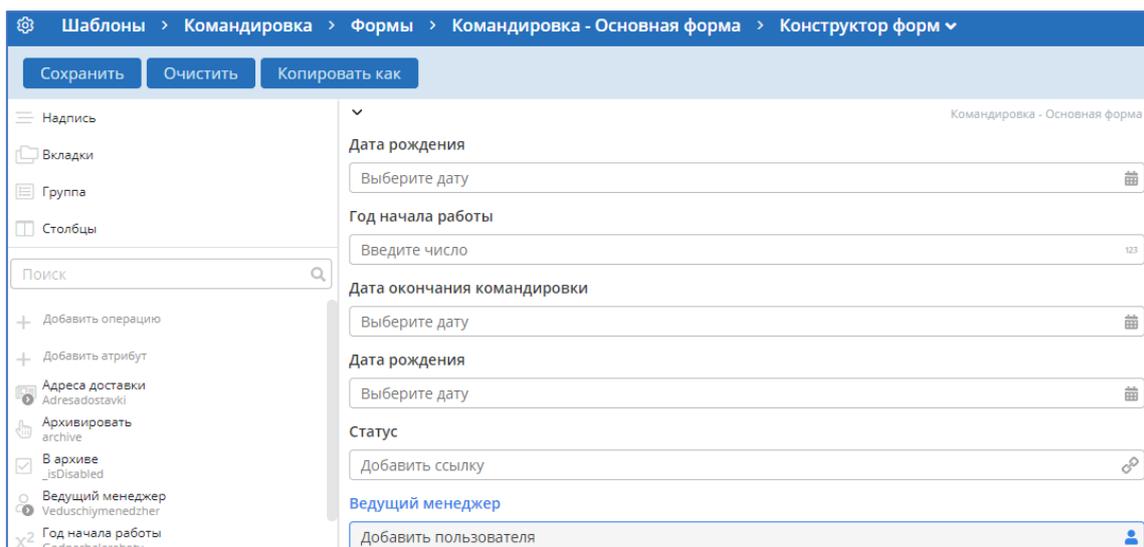


Рис. 86 Внешний вид атрибута типа «Пользователь» в конструкторе форм

Платформа позволяет добавлять на форму атрибуты связанных по ссылке сущностей неограниченного уровня вложенности. Каждый атрибут типа «Пользователь» в панели элементов конструктора форм имеет возможность раскрыть содержимое, путем нажатия левой кнопки мыши на стрелку, находящуюся на иконке слева от атрибута ссылки.

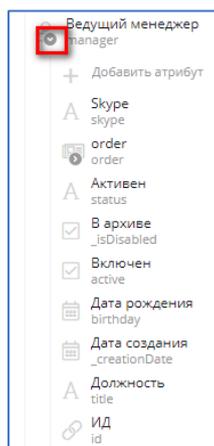


Рис. 87 Элемент раскрытия содержимого атрибута типа «Пользователь»

Содержимое атрибута типа «Пользователь» является перечнем системных атрибутов системной сущности Пользователи и пользовательских атрибутов шаблона пользователя, с которым связан текущий атрибут. Кроме этого, платформа позволяет добавить новый атрибут в связанный шаблон пользователя (если он указан) без необходимости перехода на страницу атрибутов связанного шаблона пользователя.

Таки образом платформа позволяет добавлять на форму любые атрибуты всех связанных между собой шаблонов пользователей на неограниченную глубину связей.

Панель настроек атрибута типа «Пользователь» имеет следующие настройки:

- Системное имя – уникальное текстовое системное имя элемента в рамках настраиваемой формы редактирования, используемое для настройки динамических правил на формы редактирования. Может содержать только буквы латинского алфавита, цифры и знак «_».
- Название поля – отображаемая на интерфейсе пользователя подпись к полю.

- Путь к атрибуту – показывает полный путь к атрибуту. Если на форму вынесен атрибут текущего шаблона записи, тогда в текущей характеристике отображается системное имя атрибута. Если атрибут выбран из связанной сущности, то характеристика покажет полный путь до атрибута, например, вынесение описания получателя товара по выбранному адресу поставки:

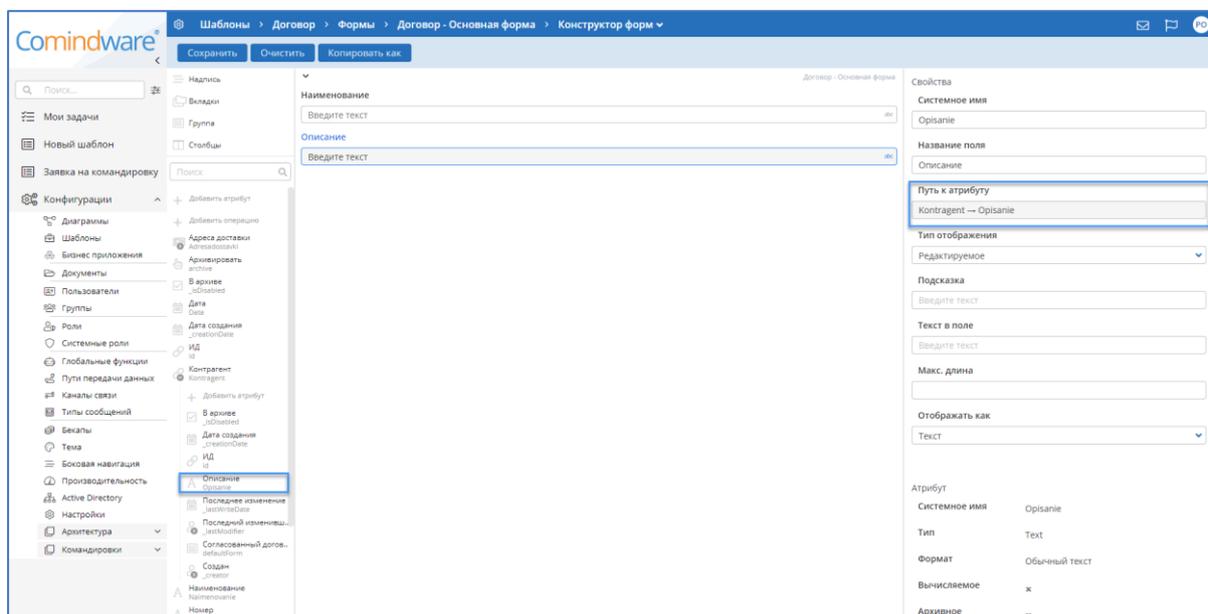


Рис. 88 Внешний вид сложного пути к атрибуту

- Тип отображения – показывает, каким образом данное поле должно быть доступно пользователю на форме редактирования. Имеет 4 состояния:
 - Обязательное – поле доступно для редактирования и обязательно для заполнения пользователем. Неактуально для вычисляемых и системных атрибутов.
 - Редактируемое – поле доступно для редактирования. Неактуально для вычисляемых и системных атрибутов.
 - Только для чтения – поле недоступно для редактирования и необязательно.
 - Скрытое – поле скрыто для пользователя на форме редактирования.
- Подсказка – текстовая информация, которая отображается пользователю, при щелчке на знак вопроса, расположенный рядом с подписью поля, в котором определена подсказка.
- Скрыть строку поиска – флаг, который на форме редактирования пользователя не дает возможности осуществлять поиск контрагента по вводимым символам.
- Несколько пользователей – флаг включает режим, при котором пользователь сможет выбирать и сохранять в атрибуте несколько значений пользователей.
- Макс. пользователей – указывается максимальное количество пользователей, которые можно будет выбрать на форме редактирования. Указывается только в случае, если включен флаг Несколько пользователей.

- Фильтр – запрос, который позволяет ограничить перечень значений пользователей, доступных для выбора пользователю на форме редактирования. Фильтр должен возвращать перечень id пользователей. Выражение пишется либо на языке выражений платформы, либо на языке запросов RDF N3 ⁽²²³⁾.

Внешний вид на форме редактирования пользователя.

Подразделения > подразделение для тестирования задачи

Сохранить Архивировать

Наименование
подразделение для тестирования задачи

Согласующий
Введите имя...

- Administrator1
- Aleksandr Oleynikov
- Alexey Nevgin
- Amazon Mailer
- Andrey Volodin
- assistant
- Backup
- comindware-awesome-bot
- Daria Bogdanova
- demo
- dmarc

Отображены не все значения, уточните поиск

Рис. 89 Внешний вид поля атрибута типа «Пользователь» на форме редактирования

Поле атрибута типа «Пользователь» показывается на форме в виде выпадающего списка. По умолчанию в список выводятся первые сто пользователей. Если перечень хранимых значений пользователей превышает сто записей, тогда платформа выдает предупреждение «Отображены не все значения, уточните поиск».

[Ссылка](#)

Внешний вид в конструкторе форм.

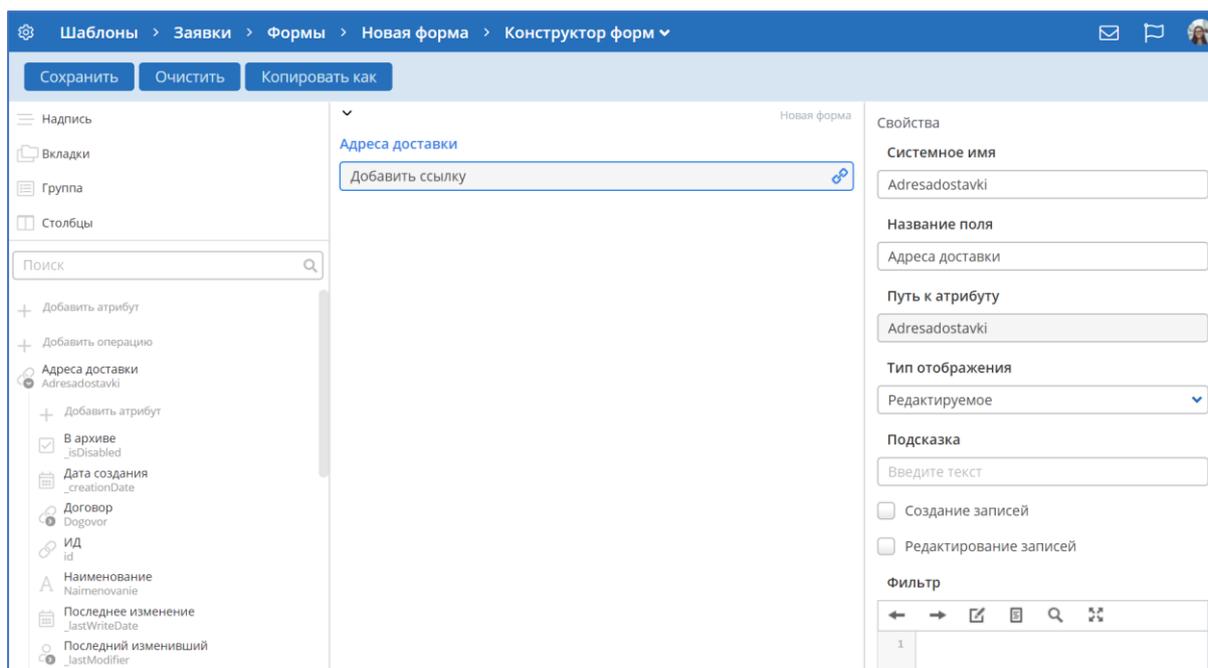


Рис. 90 Внешний вид атрибута типа «Ссылка» в конструкторе форм

Платформа позволяет добавлять на форму атрибуты связанных по ссылке сущностей неограниченного уровня вложенности. Каждый атрибут типа «Ссылка» в панели элементов конструктора форм имеет возможность раскрыть содержимое, путем нажатия левой кнопки мыши на стрелку, находящуюся на иконке слева от атрибута ссылки.

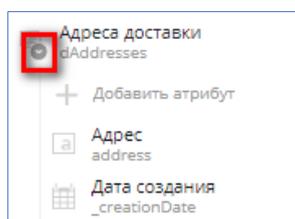


Рис. 91 Элемент раскрытия содержимого атрибута типа «Ссылки»

Содержимое атрибута типа «Ссылка» является перечнем атрибутов шаблона записи, на который ссылается атрибут типа «Ссылка». Кроме этого, платформа позволяет добавить новый атрибут в связанный шаблон записи без необходимости перехода на страницу атрибутов связанного шаблона записи.

Таким образом платформа позволяет добавлять на форму любые атрибуты всех связанных между собой шаблонов записей на неограниченную глубину связей.

Панель настроек атрибута типа «Ссылка» имеет следующие настройки:

- Системное имя – уникальное текстовое системное имя элемента в рамках настраиваемой формы редактирования, используемое для настройки динамических правил на формы редактирования. Может содержать только буквы латинского алфавита, цифры и знак «_».
- Название поля – отображаемая на интерфейсе пользователя подпись к полю.
- Путь к атрибуту – показывает полный путь к атрибуту. Если на форму вынесен атрибут текущего шаблона записи, тогда в текущей характеристике отображается системное имя атрибута. Если атрибут выбран из связанной сущности, то характеристика

покажет полный путь до атрибута, например, вынесение описания получателя товара по выбранному адресу поставки:

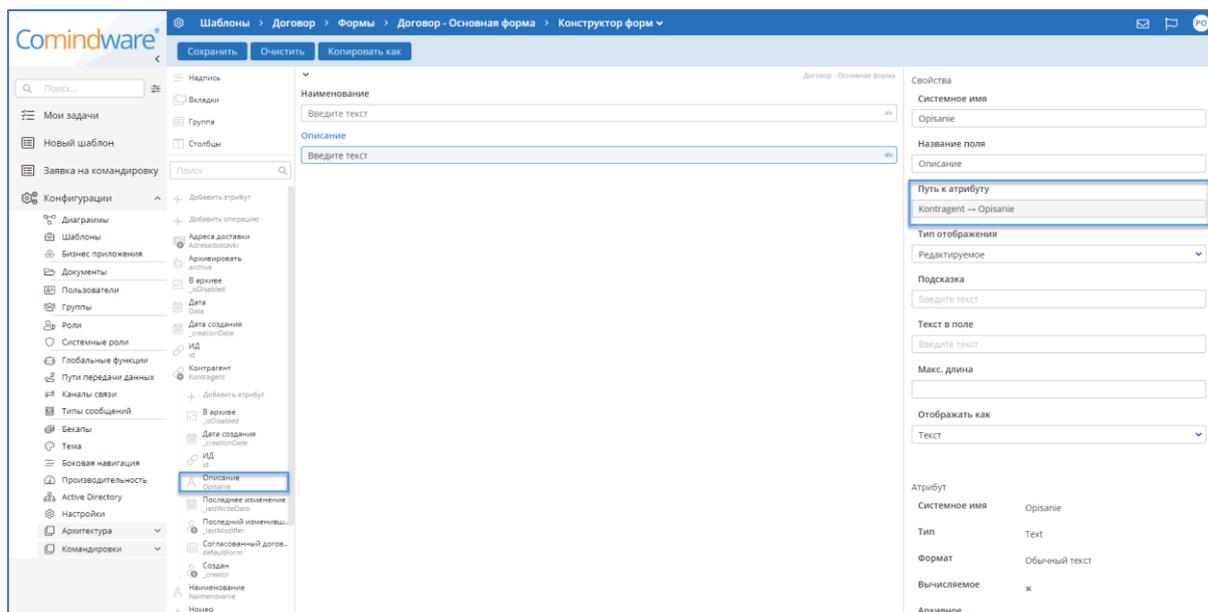
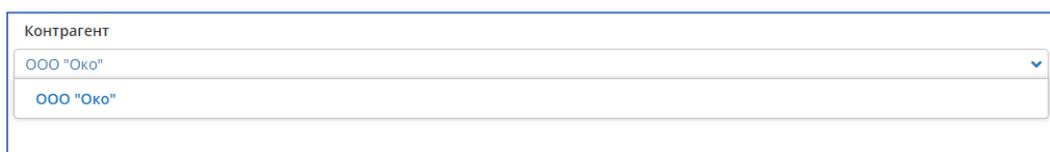


Рис. 92 Внешний вид сложного пути к атрибуту

- Тип отображения – показывает, каким образом данное поле должно быть доступно пользователю на форме редактирования. Имеет 4 состояния:
 - Обязательное – поле доступно для редактирования и обязательно для заполнения пользователем. Неактуально для вычисляемых и системных атрибутов.
 - Редактируемое – поле доступно для редактирования. Неактуально для вычисляемых и системных атрибутов.
 - Только для чтения – поле недоступно для редактирования и необязательно.
 - Скрытое – поле скрыто для пользователя на форме редактирования.
- Подсказка – текстовая информация, которая отображается пользователю, при щелчке на знак вопроса, расположенный рядом с подписью поля, в котором определена подсказка.
- Создание записей – флаг, который на форме редактирования пользователя не дает возможности осуществлять поиск контрагента по вводимым символам.
- Редактирование записей – флаг включает режим возможности перехода к форме редактирования записи.
- Форма во всплывающем окне – выпадающий список, содержащий перечень форм связанного шаблона записи, на который указывает ссылка. Выбор формы доступен, если выбран хоть один флаг «Создание записей» или «Редактирование записей». Позволяет определить произвольную форму, с помощью которой необходимо создавать или редактировать элементы с данной ссылкой. Доступны для выбора публичные и внутренние формы.
- Фильтр – запрос, который позволяет ограничить перечень записей, доступных для выбора пользователю на форме редактирования в выпадающем списке. Фильтр

должен возвращать перечень id записей. Выражение пишется либо на языке выражений платформы, либо на языке запросов RDF N3⁽²²³⁾.

Внешний вид на форме редактирования пользователя.



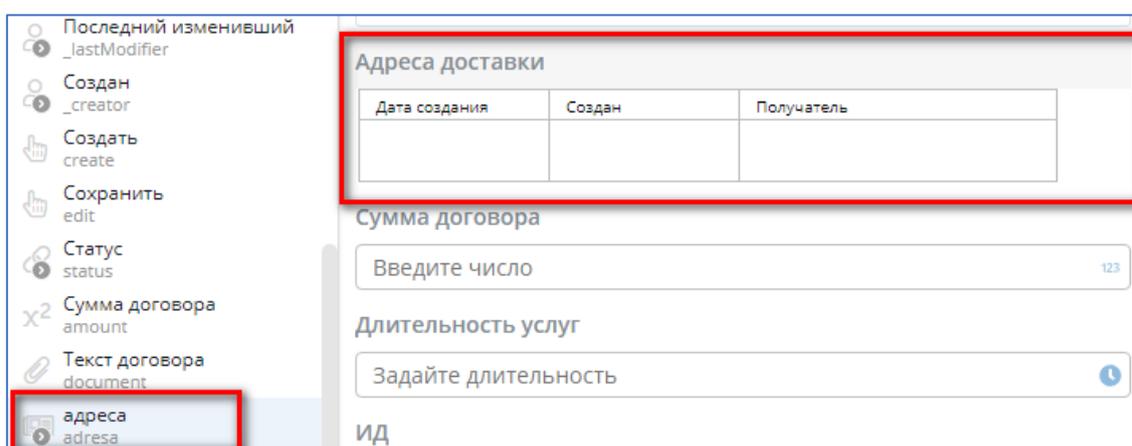
Контрагент
ООО "Око"
ООО "Око"

Рис. 93 Внешний вид поля атрибута типа «Ссылка» на форме редактирования

Поле атрибута типа «Ссылка» показывается на форме в виде выпадающего списка. По умолчанию в список выводятся первые сто пользователей. Если перечень хранимых значений пользователей превышает сто записей, тогда платформа выдает предупреждение «Отображены не все значения, уточните поиск».

Коллекция

Внешний вид в конструкторе форм.



Последний изменивший
_lastModifier
Создан
_creator
Создать
create
Сохранить
edit
Статус
status
Сумма договора
amount
Текст договора
document
адреса
adresa

Адреса доставки

Дата создания	Создан	Получатель

Сумма договора

Введите число 123

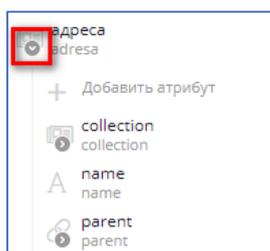
Длительность услуг

Задайте длительность

ИД

Рис. 94 Внешний вид атрибута типа «Коллекция» в конструкторе форм

Платформа позволяет добавлять на форму атрибуты связанных коллекцией сущностей неограниченного уровня вложенности. Каждый атрибут типа «Коллекция» в панели элементов конструктора форм имеет возможность раскрыть содержимое, путем нажатия левой кнопки мыши на стрелку, находящуюся на иконке слева от атрибута ссылки.



адреса
adresa

+ Добавить атрибут

- collection
collection
- A name
name
- parent
parent

Рис. 95 Элемент раскрытия содержимого атрибута типа «Коллекция»

Содержимое атрибута типа «Коллекция» является перечнем атрибутов шаблона записи, с которым связан атрибут. Кроме этого, платформа позволяет добавить новый атрибут в связанный шаблон записи без необходимости перехода на страницу атрибутов связанного шаблона записи.

Таким образом платформа позволяет добавлять на форму любые атрибуты всех связанных между собой шаблонов записей на неограниченную глубину связей.

Панель настроек атрибута типа «Коллекция» имеет следующие настройки:

- Системное имя – уникальное текстовое системное имя элемента в рамках настраиваемой формы редактирования, используемое для настройки динамических правил на формы редактирования. Может содержать только буквы латинского алфавита, цифры и знак «_».
- Заголовок – текст, отображаемый в качестве подписи к таблице на пользовательском интерфейсе. Если необходимо подпись оставить не заполненной, то рекомендуется ставить в качестве значения заголовка «пробел».
- Путь к атрибуту – показывает полный путь к атрибуту. Если на форму вынесен атрибут текущего шаблона записи, тогда в текущей характеристике отображается системное имя атрибута. Если атрибут выбран из связанной сущности, то характеристика покажет полный путь до атрибута, например, вынесение перечня адресов доставки товара по выбранному контрагенту в договоре:

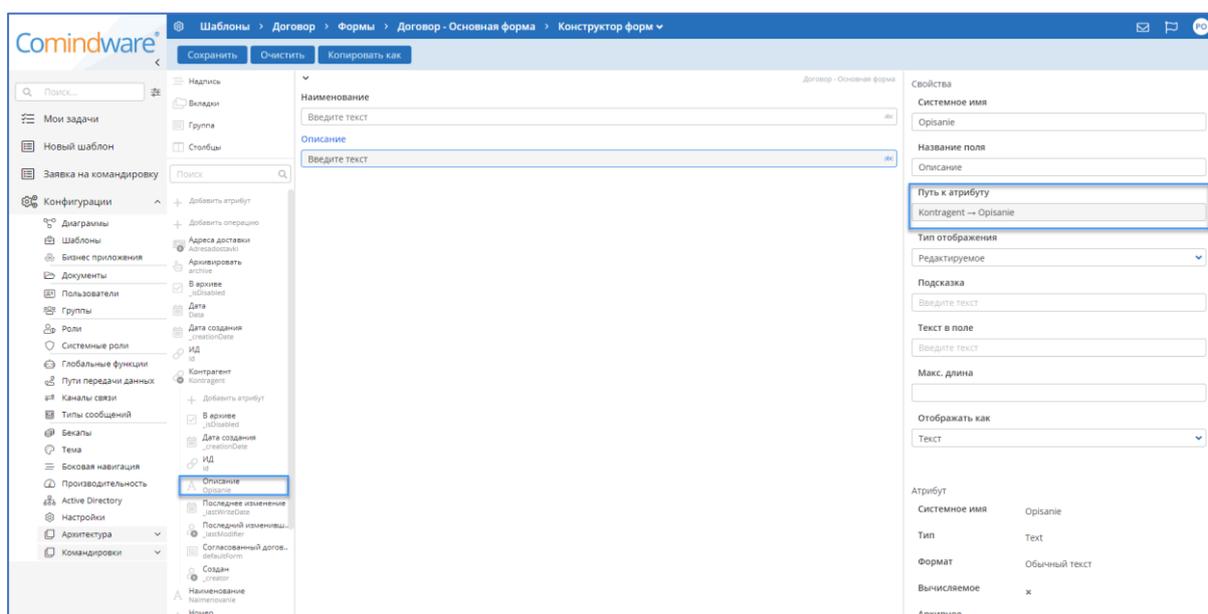


Рис. 96 Внешний вид сложного пути к атрибуту

- Подсказка – текстовая информация, которая отображается пользователю, при щелчке на знак вопроса, расположенный рядом с подписью поля, в котором определена подсказка.
- Тип отображения (таблица или шевроны) – показывает какой внешний вид на интерфейсе пользователя будет принимать поле «Коллекция».
 - Таблица – «Коллекция» будет отображена в табличном виде, колонки которой определяются также в дизайнера.
 - Шевроны – записи в «Коллекции» будут отображены на интерфейсе в виде горизонтально расположенных в заданной последовательности шевронов.

Для улучшения взаимодействия с коллекцией на форме пользователю доступны горячие клавиши:

- Tab - переход на следующую ячейку;
- Shift+Tab - переход на предыдущую ячейку;
- Стрелки - навигация по ячейкам коллекции;
- Page up - перемещение на один экран коллекции вверх;
- Page down - переход на один экран вниз;
- Space - выбор строки;
- F2 - начало редактирования ячейки.

<input type="checkbox"/>	№ п/п	Вид	...	Плановая ...	Плановая ...	Ответственный	Исполните
<input type="checkbox"/>	4.1	Общая	...	14.01.2019	29.03.2019	* БТ	* ВИ
<input type="checkbox"/>	4.2	Общая	...	14.01.2019	31.12.2019	* БЕ	* СЯ
<input type="checkbox"/>	4.3	Общая	...	14.01.2019	31.12.2019	* АВ	* СС
<input type="checkbox"/>	4.4	Общая	...	14.01.2019	29.03.2019	* БМ	* СС
<input type="checkbox"/>	4.5	Общая	...	01.04.2019	28.06.2019	* ФА	* СС
<input type="checkbox"/>	4.6	Общая	...	01.04.2019	28.06.2019	* ФА	* СС

Рис. 97 Внешний вид «Коллекции» с типом отображения «Таблица» на пользовательском интерфейсе

Ниже приведены свойства для поля «Коллекция», в настройках которого выбран тип отображения «Таблица».

- Тип отображения (вид доступа) – показывает, каким образом данное поле должно быть доступно пользователю на форме редактирования. Имеет 4 состояния:
 - Обязательное – поле доступно для редактирования и обязательно для заполнения пользователем. Неактуально для вычисляемых и системных атрибутов. Для коллекции это значит, что в коллекции должна присутствовать хоть одна запись.
 - Редактируемое – поле доступно для редактирования. Неактуально для вычисляемых и системных атрибутов. Коллекция доступна для редактирования.
 - Только для чтения – поле недоступно для редактирования и необязательно.
 - Скрытое – поле скрыто для пользователя на форме редактирования.
- Дочерний атрибут и Атрибут предшественника – используются для настройки отображения иерархии элементов в коллекции. В качестве дочернего атрибута указывается коллекция, которая хранит перечень дочерних элементов дерева. В качестве атрибута предшественника указывается атрибут «Ссылка», который хранит ссылку на родительский элемент в иерархии.
- Порядковый атрибут – указывается атрибут типа «Число», который при добавлении элементов в коллекцию будет хранить порядковый номер. Порядковый номер записи пересчитывается при ручном изменении порядка элементов коллекции.

-
- Атрибут автонумерации – указывается текстовый атрибут, который будет хранить нумерацию элементов коллекции с учетом их иерархии (1, 1.1, 1.2, 1.2.1 и т.д.). Нумерация записи пересчитывается при ручном изменении порядка элементов коллекции.
 - Сортировать по и Направление сортировки – указывается атрибут и порядок, согласно которому необходимо сортировать элементы коллекции. Для древовидной коллекции сортировка будет работать только в ручном режиме.
 - Фильтр для отображения – запрос, который позволяет ограничить перечень записей коллекции, отображаемых в таблице на интерфейсе пользователя. Фильтр должен возвращать перечень id записей, отображаемых в коллекции. Выражение пишется либо на языке выражений платформы, либо на языке запросов RDF N3⁽²²³⁾.
 - Фильтр для добавления – запрос, который позволяет ограничить перечень существующих записей, которые возможно включить в коллекцию. Используется только при включенном режиме «Добавление существующих записей». Фильтр должен возвращать перечень id записей, возможных для добавления в коллекцию. Выражение пишется либо на языке выражений платформы, либо на языке запросов RDF N3⁽²²³⁾.
 - Добавление существующих записей – флаг позволяет включить возможность выбора и добавления существующих записей с учетом настроенного фильтра для добавления. При включении флага на интерфейсе пользователя в заголовке коллекции появляется кнопка для выбора существующих записей из связанного с коллекцией шаблона записи.
 - Редактирование записей – флаг включает режим возможности перехода к форме редактирования записи. При включении флага на интерфейсе пользователя в заголовке коллекции появляется кнопка редактирования выбранной в коллекции записи.
 - Переход на связанный объект – флаг, который позволяет переходить на форму редактирования записи коллекции с покиданием формы редактирования текущей записи. При включении флага на интерфейсе пользователя в заголовке коллекции появляется кнопка, позволяющая перейти в отдельном окне на форму редактирования выбранной записи коллекции.
 - **Внимание.** Отличие режима перехода на связанный объект от редактирования записи заключается в том, что при редактировании объекта в режиме редактирования записей сохранение данных будет произведено только после сохранения информации на основной записи. В случае перехода на связанный объект сохранение данных связанной записи будет произведено в момент нажатия на кнопку «Сохранить» на связанном объекте.
 - Показывать архивные записи – флаг, позволяющий отобразить на интерфейсе пользователя в списке записей коллекции записи, которые были архивированы.
 - Способ создания сущностей – выпадающий список, который позволяет настроить порядок создания записей в коллекции:
 - Не доступно – создание записей связанной сущности в коллекции невозможно.

-
- Форма – создание записей связанного шаблона записи производится с использованием формы редактирования. При выборе данной опции на интерфейсе пользователя в заголовке коллекции появляется кнопка добавления записи, при нажатии на которую будет открываться форма редактирования связанной записи, добавляемой в коллекцию. После сохранения формы, запись будет добавлена в коллекцию. Однако, окончательное сохранение данных произойдет только после сохранения основной записи.
 - Встроенное – создание записей связанного шаблона записи производится непосредственно в коллекции. При выборе данной опции на интерфейсе пользователя в заголовке коллекции появляется кнопка добавления записи, при нажатии на которую будет добавляться строка в коллекции, в которой будет производиться редактирование атрибутов записи.
 - Удаление записей – выпадающий список, позволяющий настроить каким образом будет производиться удаление записей из коллекции:
 - Отключено – удаление записей из коллекции не допускается.
 - Убрать из коллекции – опция позволяет на интерфейсе пользователя в заголовке коллекции включить кнопку удаления выбранных записей коллекции путем отсоединения их от коллекции, т.е. очистки значений ссылок связанной сущности на текущую запись.
 - Архивировать – опция позволяет на интерфейсе пользователя в заголовке коллекции включить кнопку удаления выбранных записей коллекции путем архивирования записей. При этом записи не отсоединяются от коллекции. Таким образом, если на коллекции включен режим отображения архивных записей, фактического удаления из коллекции не произойдет.
 - Удалить – опция позволяет на интерфейсе пользователя в заголовке коллекции включить кнопку удаления выбранных записей коллекции путем физического удаления связанной записи из базы данных.
 - Форма во всплывающем окне – указывается форма редактирования связанного шаблона записи, которая должна открываться при двойном щелчке левой кнопкой мыши по записи коллекции, нажатию на кнопку редактирования или добавления новой записи (при условии режима добавления записи «Форма»).

Коллекция, представленная в виде древовидной структуры

Коллекция, представленная в виде древовидной структуры, позволяет сформировать в шаблоне записи наглядную иерархию записей из коллекции, которая формируются при помощи автонумерации (1, 1.1, 1.2 итд.) и порядкового номера:

- Автонумерация – текстовый атрибут, который хранит нумерацию элементов коллекции с учетом их иерархии (1, 1.1, 1.2, 1.2.1 и т.д.). Нумерация записи пересчитывается при ручном изменении порядка элементов коллекции.
- Порядковый номер - указывается атрибут типа «Число», который при добавлении элементов в коллекцию хранит порядковый номер. Порядковый номер позволяет вручную менять последовательность записей без добавления и удаления записей, а также необходим для присвоения порядкового номера дочернему атрибуту.

Для формирования Коллекция, представленная в виде дерева, необходимо добавить «Дочерний атрибут» и «Атрибут предшественника» в конструкторе форм (описание атрибутов см. выше в данном разделе).

- Дочерний атрибут – это атрибут из выбранной коллекции, который также имеют тип данных коллекция и целевой шаблон, которого совпадает с целевым шаблоном выбранной коллекции.
- Атрибут предшественника — это атрибут из выбранной коллекции, с типом данных - ссылка и целевой шаблон, которого совпадает с целевым шаблоном выбранной коллекции. Для наглядного представления см. таблицу 1.

Ниже представлена таблица доступности типов действий для элементов формы различных типов.

Выбранная коллекция			
	Тип данных	Целевой шаблон	Ссылка
Дочерний атрибут	Коллекция	Целевой шаблон текущей коллекции	Атрибут предшественника
Атрибут предшественника	Ссылка	Целевой шаблон текущей коллекции	-

Таблица 1. Параметры дочернего атрибута и атрибута предшественника.

Например, в шаблон записи «**Контрагенты**» необходимо добавить коллекцию договоров, где записи имеют иерархию: общих договоров и дополнительных соглашений.

Выбранная коллекция- Договоры			
	Тип данных	Целевой шаблон	Ссылка
Дочерний атрибут – Дополнительные соглашения	Коллекция	Целевой шаблон текущей коллекции - Договоры	Атрибут предшественника- Общие договоры
Атрибут предшественника- Общие договоры	Ссылка	Целевой шаблон текущей коллекции - Договоры	-

Таблица 2. Пример заполнения параметров дочернего атрибута и атрибута предшественника.

В результате на форме отображается дерево иерархии договоров:

Договоры

<input type="checkbox"/>	▼ Автономумерация ↓	ИД	name
1 <input type="checkbox"/>	▼ 1	30	договор 30.05
2 <input type="checkbox"/>	1.1	84	доп согл 1
3 <input type="checkbox"/>	1.2	85	доп согл 2

Рис. 98 Внешний вид атрибута типа «Коллекция» с настроенным деревом иерархии.

После настройки дерева иерархии, в коллекции появляется возможность добавлять/создавать записи определенного уровня или перемещать запись, на нужный уровень. Если добавление или создание записей ограничены настройками коллекции, то данные функции недоступны.

Для создания/добавления записи на уровень ниже необходимо выбрать запись из коллекции и нажать кнопку добавления записи - . После этого в коллекции отображается новая запись, атрибут автономумерации, которой на уровень ниже выбранной записи. В случае, если на конструкторе форм был настроен способ создания сущностей в форме, то после добавления записи отобразится форма создания для дочерней записи (см. п. Коллекция). Другой способ создания записей на уровень ниже – выбрать запись правой кнопкой и во всплывающем меню нажать «Создать».

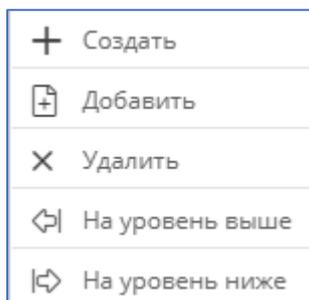


Рис. 99 Всплывающее окно дерева иерархии.

Для редактирования уровня иерархии записей коллекции необходимо кликнуть правой кнопкой мыши на запись из коллекции. Далее из всплывающего окна выбрать соответствующую функцию (на уровень выше/на уровень ниже), после чего атрибут автономумерации выбранной записи меняет значение на соответствующее присвоенному уровню. Записи высшего уровня (1, 2, итд.) можно только понизить, а записи низшего уровня - только повысить.



Рис. 100 Панель инструментов доступная при выделении записи.

Создание/добавление и редактирование уровней также доступно из панели инструментов над коллекцией. Необходимо выделить запись из списка галочкой - слева, после чего сверху - в панели инструментов появятся доступные функции для данной записи.

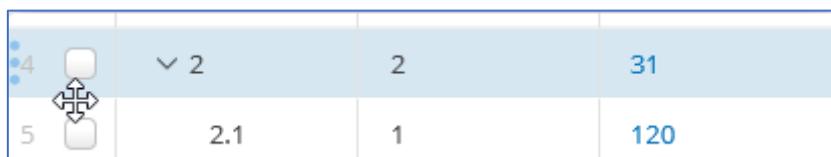


Рис. 101 Значок перемещения записей в коллекции.

Для перемещения записи в коллекции необходимо навести курсор мыши на данную запись - слева, до отображения синего значка, после чего нажать левой кнопкой мыши запись и перетащить ее на новое место в коллекции. Если переместить дочерний атрибут в область другого атрибута предшественника, то он присвоится новому атрибуту предшественнику.

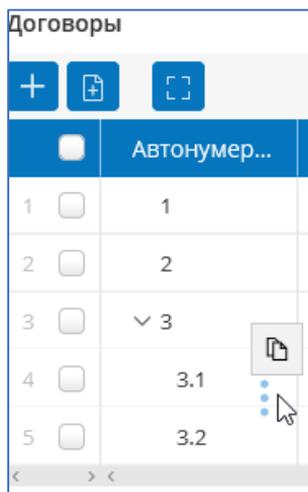


Рис. 102 Перемещение записи в коллекции.

Ниже приведены свойства для поля «Коллекция», в настройках которого выбран тип отображения «Шевроны».

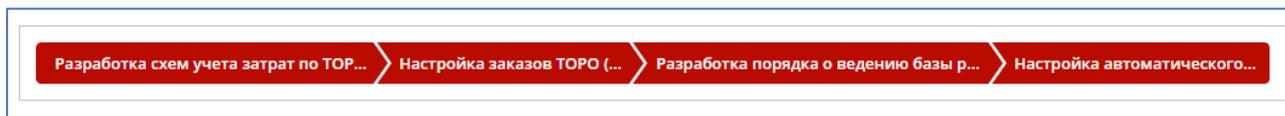


Рис. 103 Внешний вид «Коллекции» с типом отображения «Таблица» на пользовательском интерфейсе

- Порядковый атрибут – определяет числовой атрибут, согласно которому будет производиться выстраивание последовательности записей коллекции в виде шевронов.

- Фильтр – запрос, который позволяет ограничить перечень записей, выводимых в виде шевронов. Фильтр должен возвращать перечень id записей, связанных с текущей записью. Выражение пишется либо на языке выражений платформы, либо на языке запросов RDF N3⁽²²³⁾.
- Форма во всплывающем окне – указывается форма редактирования связанного шаблона записи, которая должна открываться при двойном щелчке левой кнопкой мыши по шеврону.
- Правила окраски записей – определяют правила окраски шевронов на интерфейсе пользователя.

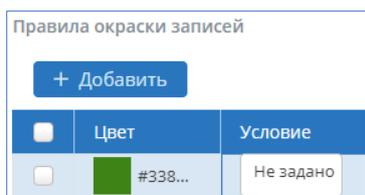


Рис. 104 Внешний вид блока настройки правил окраски данных

Для настройки правил необходимо:

1. Нажать на кнопку Добавить.
2. Выбрать цвет, в который необходимо окрасить шеврон

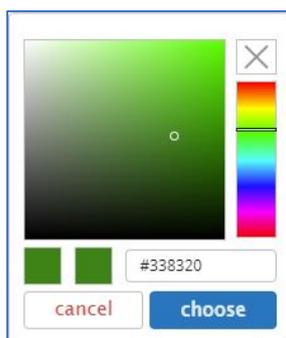


Рис. 105 Внешний вид окна выбора цвета

- Определить правило, согласно которому должен возвращаться выбранный цвет шеврона. Правило должно возвращать true для записей, которые должны быть окрашены в выбранный цвет. Выражение пишется либо на языке выражений платформы, либо на языке запросов RDF N3. Если настройкой определено несколько правил, которые для одной записи одновременно вернули разный цвет, то берется цвет, который был последним в последовательности описанных правил.

Операция

Подробное описание настройки операций можно найти в разделе ⁽⁹⁷⁾.

Панель элементов содержит набор операций, настроенных на текущем шаблоне записи.

Ограничение. Платформа не позволяет добавить на форму операции связанных сущностей, так как контекст запуска операции в общем случае будет неопределенным. Данное ограничение можно обойти путем добавления кнопки на форму связанной сущности и добавления формы из связанной сущности на текущую форму редактирования.

При добавлении операции на рабочую область конструктора визуальный элемент представлен в виде кнопки.

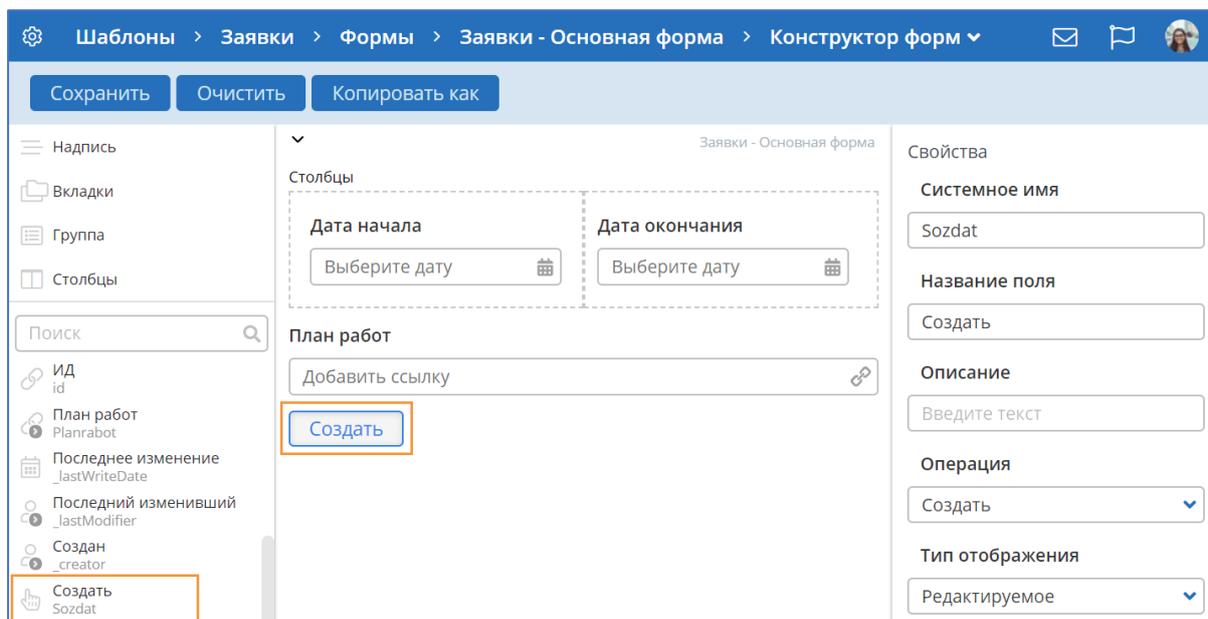


Рис. 106 Внешний вид операции в конструкторе форм

Панель настроек операции имеет следующие настройки:

- Системное имя – уникальное текстовое системное имя элемента в рамках настраиваемой формы редактирования, используемое для настройки динамических правил на формы редактирования. Может содержать только буквы латинского алфавита, цифры и знак «_».
- Название поля – отображаемая на интерфейсе пользователя подпись на кнопке.
- Операция – выпадающий список настроенных на текущем шаблоне записи операций.
- Тип отображения – показывает, каким образом кнопка должна быть доступна пользователю на форме редактирования. Имеет 3 состояния:
 - Редактируемое – кнопка доступна для запуска на форме редактирования.
 - Только для чтения – кнопка не доступна для запуска на форме редактирования.
 - Скрытое – кнопка скрыта для пользователя на форме редактирования.
- Подсказка – текстовая информация, которая отображается пользователю, при наведении курсора мыши на операцию.

Внешний вид на форме редактирования пользователя.

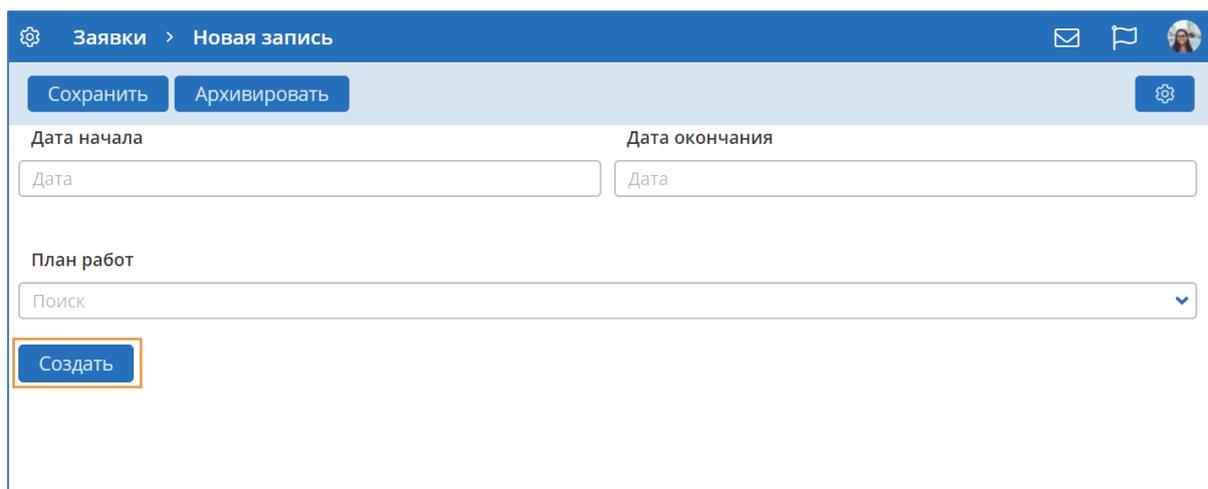


Рис. 107 Внешний вид операции на форме редактирования

На форме редактирования операция отображается в виде кнопки, которая запускается с контекстом текущей записи.

Форма

Платформа позволяет добавить на форму редактирования настроенную форму в виде вложенной формы текущего шаблона записи или из связанной сущности. Для этих целей на панели элементов конструктора форм существует возможность выбирать и добавлять настроенные формы редактирования в рабочую область конструктора форм.

При добавлении формы на рабочую область конструктора форм визуальный элемент представлен в следующем виде:

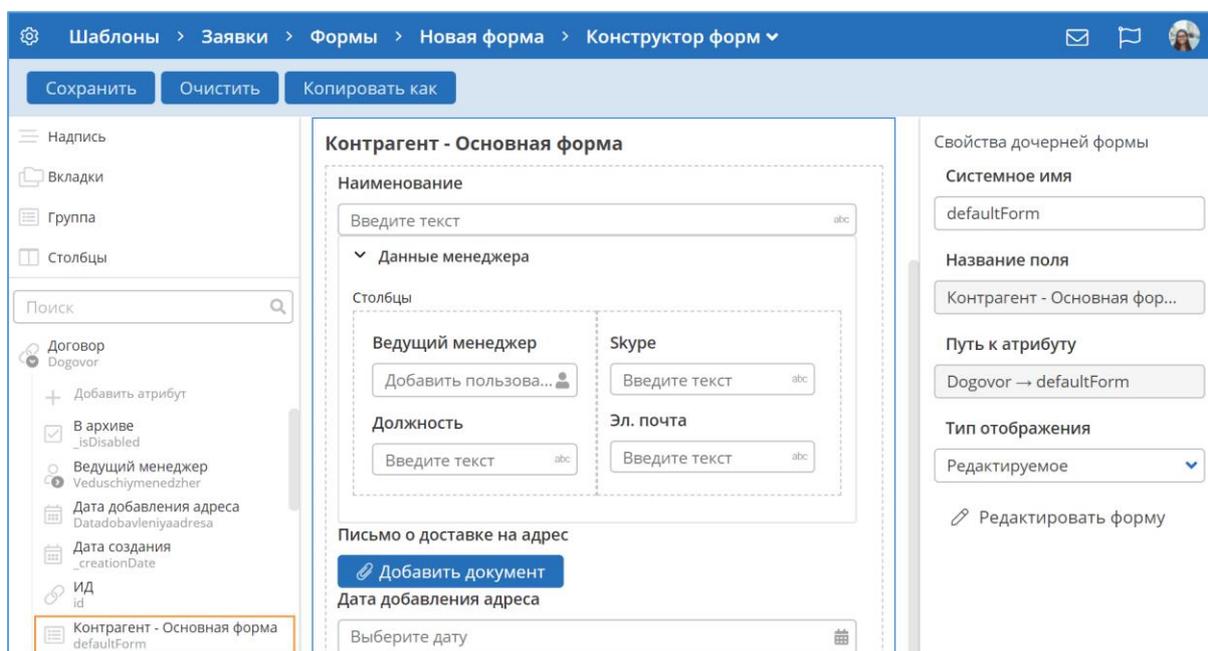


Рис. 108 Внешний вид вложенной формы в конструкторе форм

Панель настроек формы имеет следующие настройки:

- Системное имя – уникальное текстовое системное имя элемента в рамках настраиваемой формы редактирования, используемое для настройки динамических

правил на формы редактирования. Может содержать только буквы латинского алфавита, цифры и знак «_».

- Название поля – отображаемая на интерфейсе пользователя подпись к форме. Служит в информационных целях и не отображается на форме редактирования пользователя.
- Путь к атрибуту – показывает полный путь к форме. Если на форму вынесена форма текущего шаблона записи, тогда в текущей характеристике отображается системное имя формы.
- Тип отображения – показывает, каким образом кнопка должна быть доступна пользователю на форме редактирования. Имеет 3 состояния:
 - Редактируемое – вложенная форма доступна для редактирования на форме редактирования.
 - Только для чтения – вложенная форма не доступна для редактирования на форме редактирования.
 - Скрытое – вложенная форма скрыта для пользователя на форме редактирования.

Платформа позволяет добавлять вложенные формы неограниченного уровня вложенности. Если вложенная форма уже содержит в своем описании другую вложенную форму, то на форме редактирования пользователя будут отображены форма с вложенностью форм в соответствии с заданными настройками.

Платформа запрещает создавать бесконечные циклы из вложенных форм между собой. Например, цикл из двух форм, который заключается в том, что если на форму №1 добавлена форма №2 в качестве вложенной, то зайдя на конструктор формы №2 форма №1 будет недоступна в панели элементов конструктора формы №2. Аналогичное правило действует при любом количестве форм, участвующих в цикле.

Выполнение операций, вынесенных на вложенную форму, производится с контекстом записи, отображенной на вложенной форме.

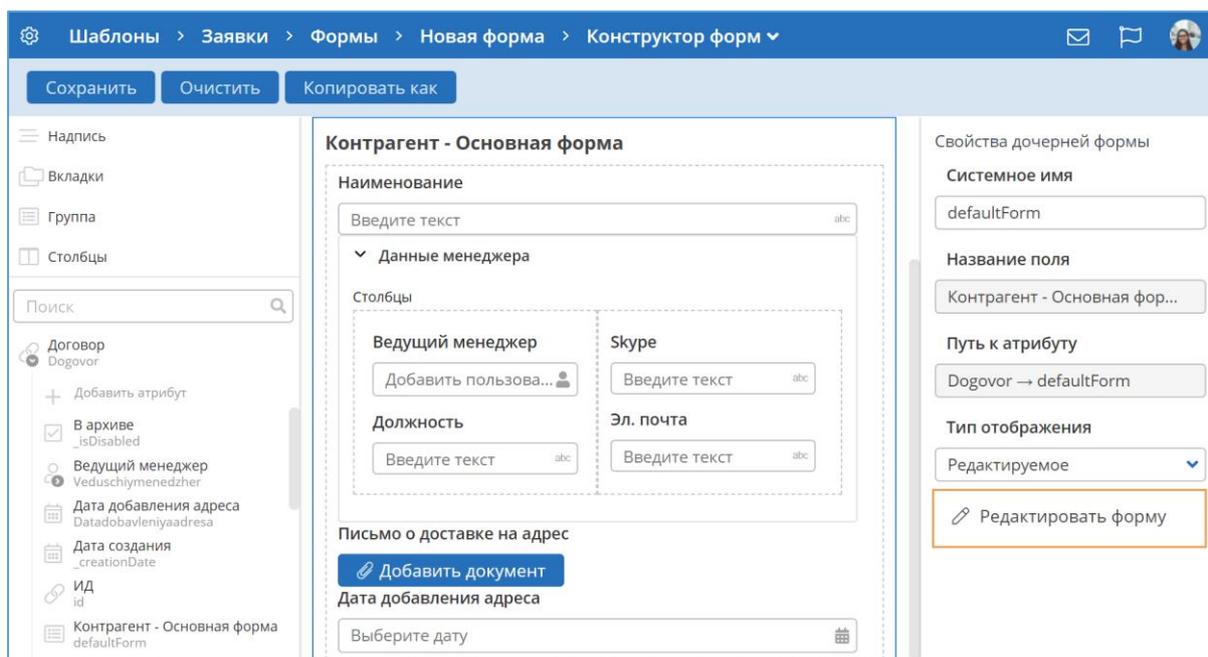


Рис. 109 Внешний вид панели настроек вложенной формы

Платформа позволяет перейти к редактированию вложенной формы путем нажатия на кнопку «Редактировать форму» на панели настроек вложенной формы.

2.4.3 Удаление формы

Для удаления формы шаблона записи необходимо:

1. Перейти в раздел **Формы**. Открывается список настроенных форм шаблона записи.

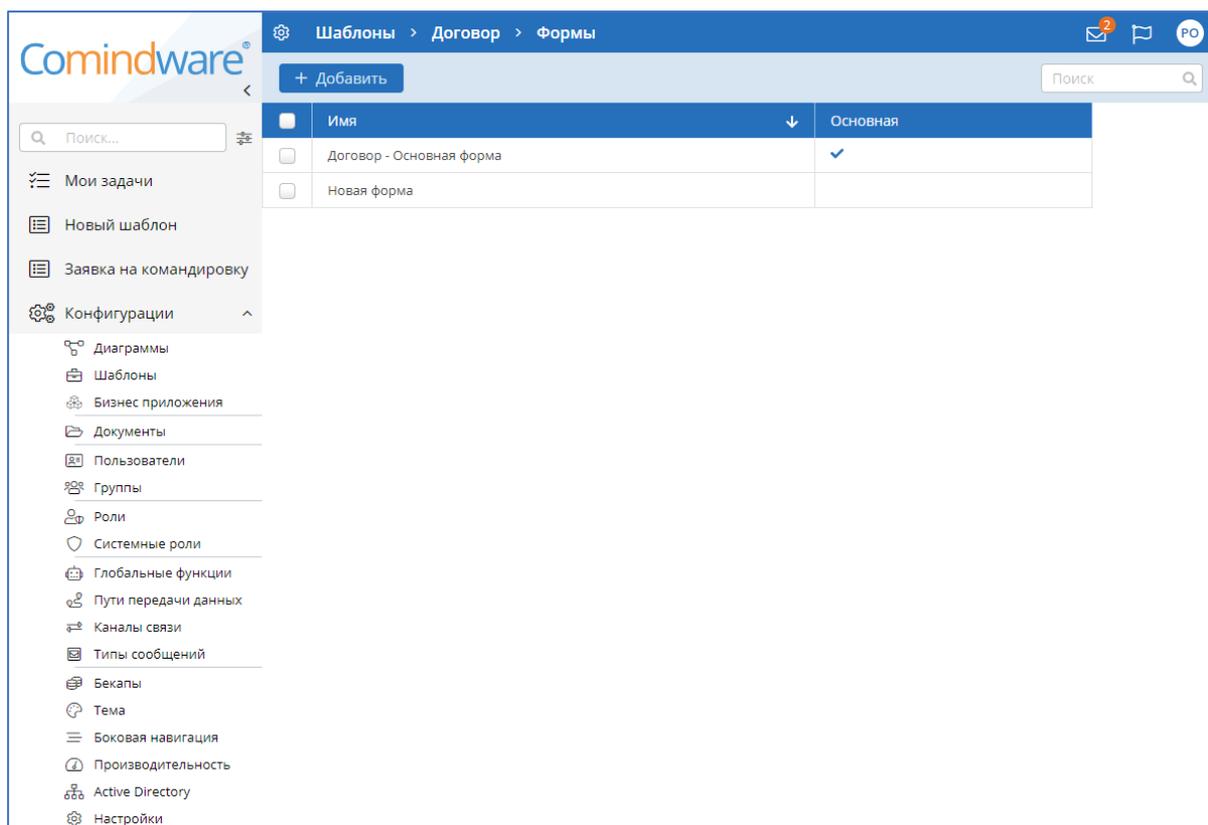


Рис. 110 Список форм шаблона записи

2. Выбрать нужную форму, отметив флаг слева от списка.

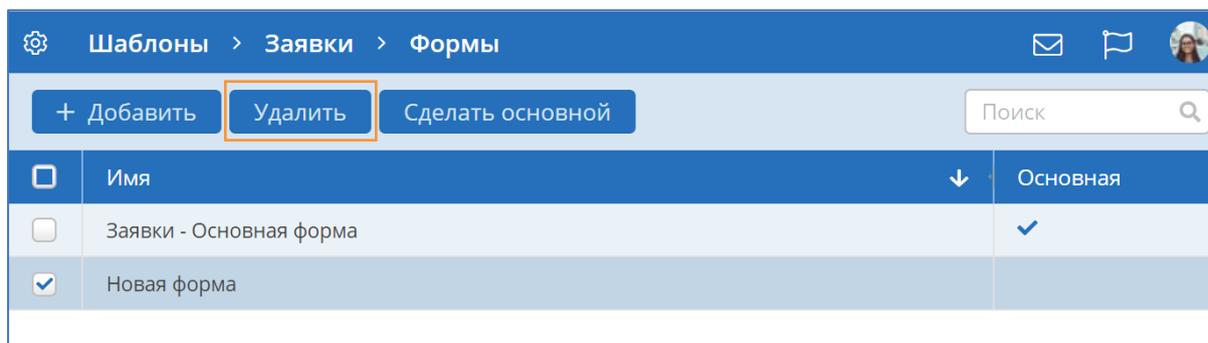


Рис. 111 Расположение кнопки «Удалить»

На панели инструментов появится кнопка «Удалить».

3. Нажать на кнопку «Удалить».

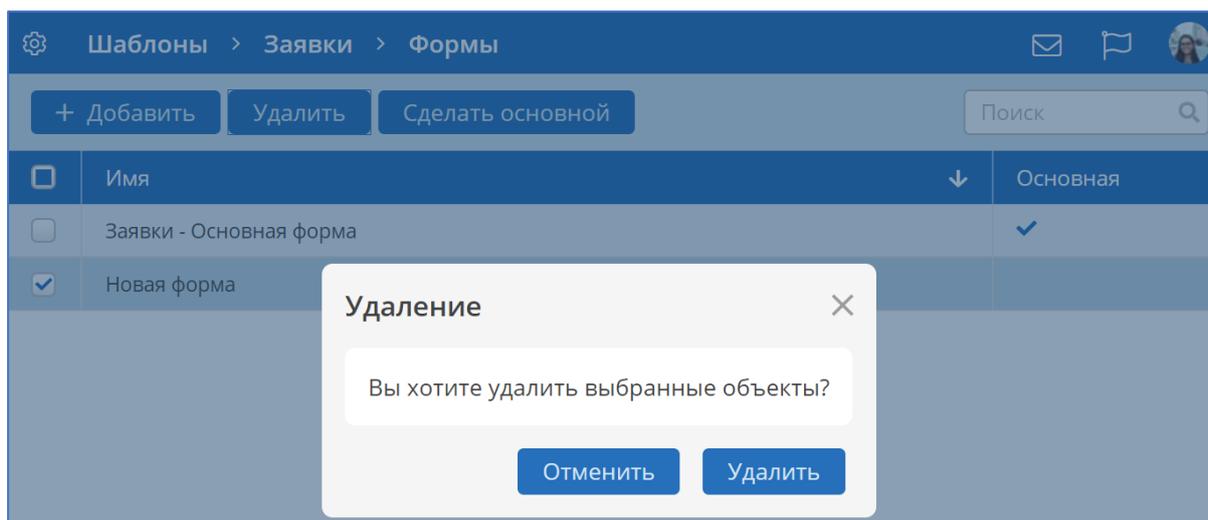


Рис. 112 Внешний вид диалогового окна подтверждения удаления формы

На экране открывается окно подтверждения удаления формы.

4. Для удаления формы нажать кнопку «Удалить». Для отмены удаления – «Отменить».

2.4.4 Правила для формы

Правила для формы предназначены для выполнения трех основных сценариев работы пользователя с платформой:

- Автоматическое условное и безусловное заполнение данными на форме (значения по умолчанию, зависимости полей и т.д.).
- Динамическое изменение внешнего вида формы (изменение доступности и видимости полей и блоков формы).
- Логический контроль введенных значений в поля формы.

Добавление правил

Для добавления правил для формы необходимо:

1. Находясь в разделе конструктора форм, перейти в раздел «Правила для формы».

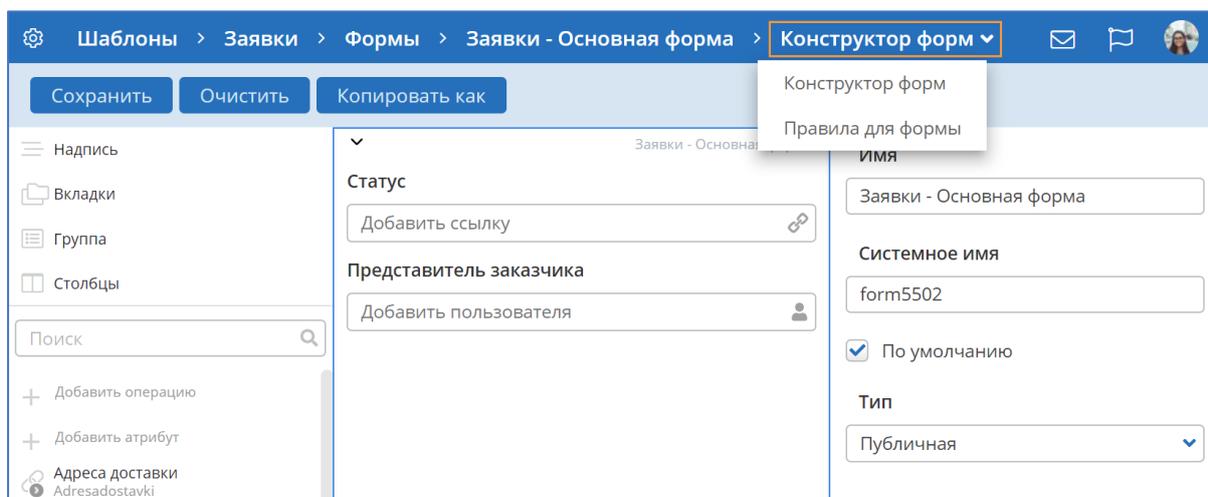


Рис. 113 Переход к разделу «Правила для формы» с конструктора форм

Открывается редактор правил для формы.

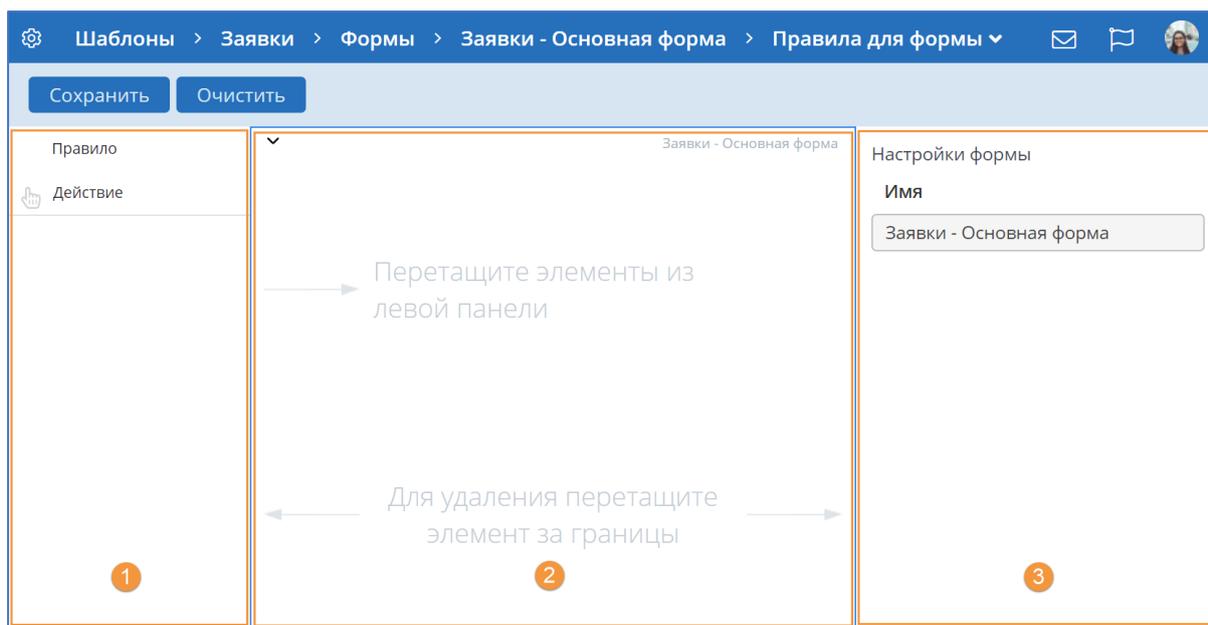


Рис. 114 Внешний вид редактора правил формы

Редактор правил имеет три области:

- (1) - Панель элементов – содержит элементы, доступные для добавления на рабочую область – Правило и Действие:
 - Правило – определяет базовое условие, которое должно выполняться, чтобы запустить выполнение действий.
 - Действие – определяют характер действий, которые необходимо выполнить при выполнении условий, заданных в правиле.
 - (2) - Рабочая область – перечень активных правил и действий формы.
 - (3) - Панель настроек – позволяет определить характеристики правил и действий формы.
2. Добавить правила и действия, необходимые для выполнения бизнес-задач с помощью перетаскивания элементов с панели элементов в рабочую область. Добавление действий возможно только в границы вынесенных правил. Подробное описание приведено в разделе «Настройка правил» (стр. 87).
 3. Для сохранения произведенных настроек нажать на кнопку «Сохранить».

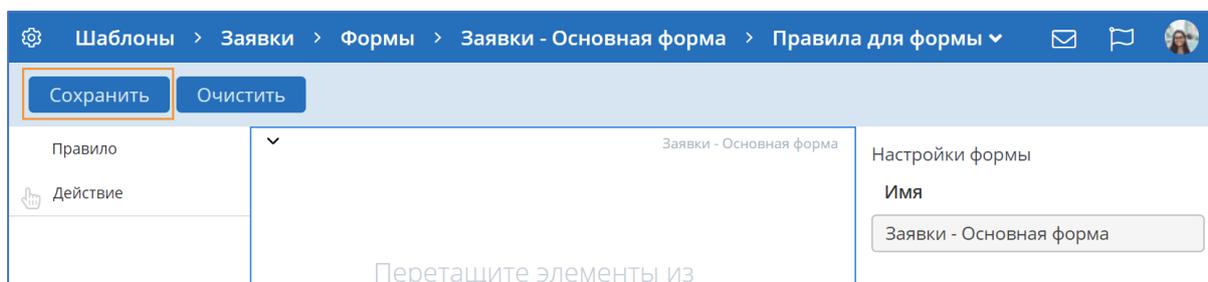


Рис. 115 Расположение кнопки «Сохранить»

Настройка правил

Настройка правил для формы выполняется путем вынесения соответствующих элементов с панели элементов в рабочую область редактора правил, а также настройки характеристик вынесенных в рабочую область элементов.

Платформа поддерживает два типа элемента, с помощью которых описываются правила формы:

- Правило – определяет базовое условие, которое должно выполняться, чтобы запустить выполнение действий.
- Действие – определяют характер действий, которые необходимо выполнить при выполнении условий, заданных в правиле.

Действие всегда включается в состав какого-то правила.

Рассмотрим каждый элемент более подробно.

Правило

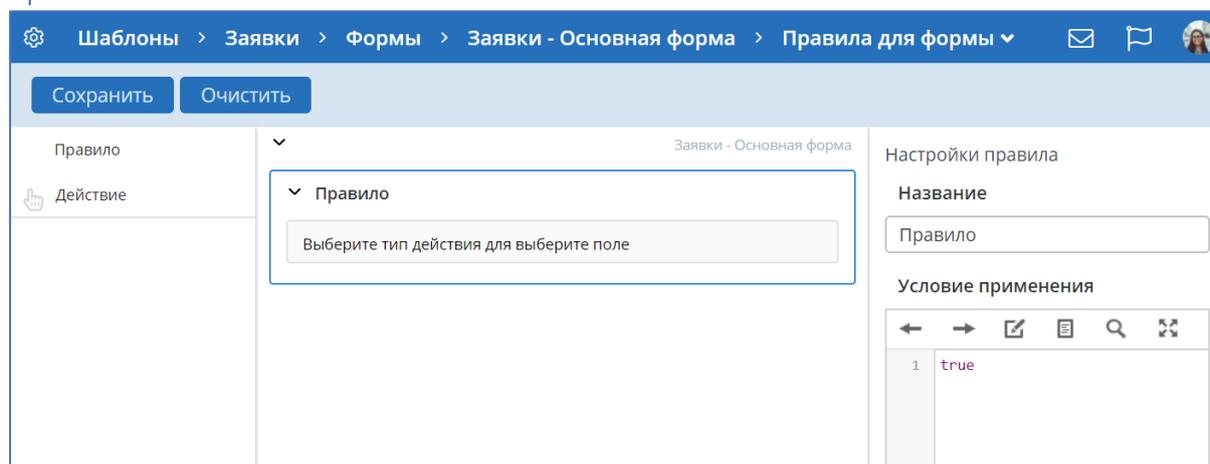


Рис. 116 Внешний вид добавленного правила в рабочую область и его характеристик

Правило является контейнером для включения в него действий.

Для описания правила, существуют следующие характеристики:

- Название – указывается логическое название правила для простоты поиска разработчиком системы.
- Условие применения – определяется условие с помощью языка выражений или запроса RDF N3. Если выражение возвращает «false», то правило не запускает выполнение действий, добавленных внутри правила. Если выражение возвращает «true», то правило запускает выполнение действий внутри правила.

Платформа позволяет добавлять неограниченное количество правил и определять порядок их расположения друг за другом путем перемещения правил с помощью указателя мыши.

Запуск проверки условия применения правил выполняется после завершения внесения изменений в любое поле на форме или при открытии формы.

Для примера, пользователь открыл запись на редактирование в настроенной форме с описанными правилами. При открытии формы, платформа производит проверку условий применения всех правил и по очереди запускает все действия (запускаемые при открытии) внутри правил, выражения для условий применения которых вернули значение «true». Если

пользователь выполнил изменение любого поля, платформа повторяет процедуру проверки применения правил и запуска действий.

Все правила выполняется независимо друг от друга в точном порядке, определенном в рабочей области редактора правил - сверху вниз. Это очень важно, так как, если в разных правилах присутствуют действия, изменяющие один и тот же атрибут и при этом условия применения этих правил вернули значение true, в результате в атрибуте будет значение, которое установит правило, находящееся ниже в конструкторе, т.е. которое выполнится последним.

Условие применения получает на вход контекст текущей записи, для которой открывается форма.

Если форма, для которой настроены правила используется в качестве вложенной формы в другую форму, то контекстом для выполнения условий применения правил вложенной формы будет запись, которая выводится в данной форме. Если вложенная форма относится к текущей записи, то контекстом будет являться текущая запись, если вложенная форма относится к ссылочному атрибуту, то контекстом для вложенной формы будет запись, указанная в атрибуте типа ссылка.

Действие

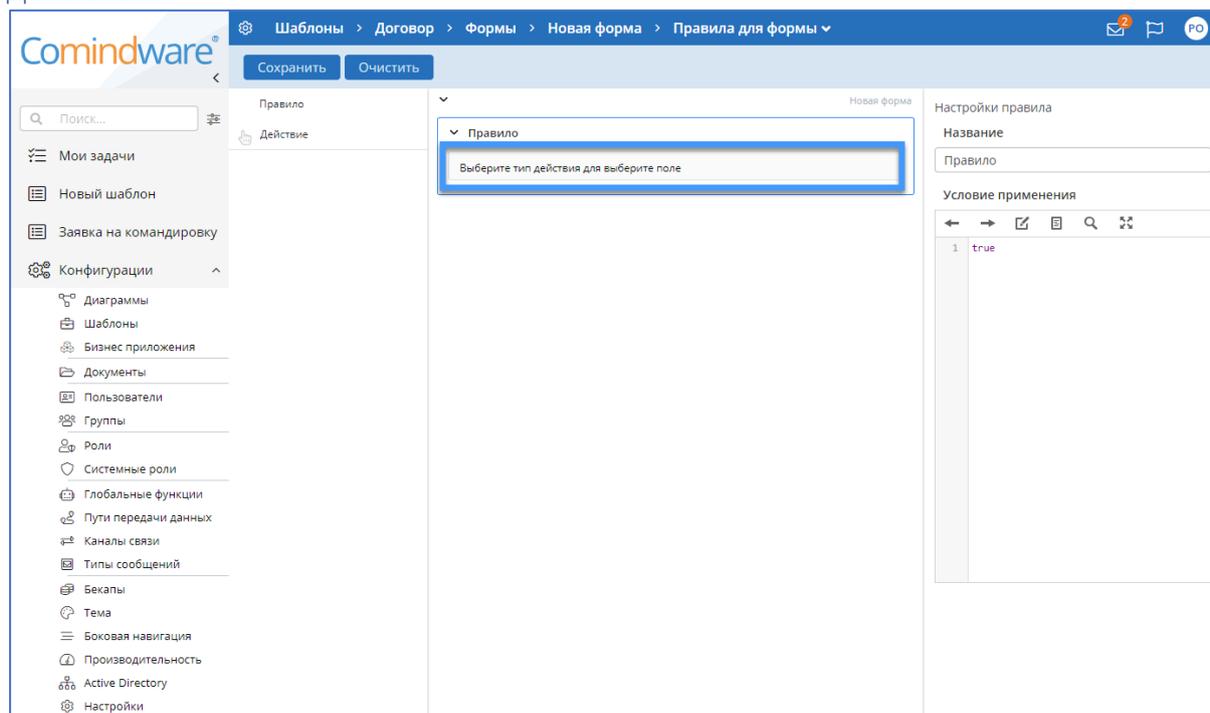


Рис. 117 Внешний вид добавленного действия в существующее правило

Действие определяет, изменение, которое необходимо произвести на форме (с полями формы или с данными формы) при наступлении условия применения действия, описанного в правиле, в которое добавленное действие вложено.

Каждое действие имеет следующие параметры для настройки:

- Поле – элемент формы (наименование и системное имя поля), над которым необходимо выполнить действие.

- Тип действия – характер действия, которое необходимо выполнить для выбранного поля (элемента формы). Появляется после выбора элемента формы.
- Условие применения – дополнительное условие применения действия, которое применяется дополнительно к условию применения, заданному в элементе «Правило», описанному в разделе «Правило». Более подробно о характеристике будет рассказано в описании каждого типа действия ниже.
- Описание дополнительных параметров в зависимости от выбранного типа действия будут в соответствующих разделах с описанием каждого типа действия ниже.

Существует 3 типа действий, которые применимы к различным элементам формы в зависимости от их типа:

- Изменить тип доступа – позволяет определить доступность элемента формы для заполнения.
- Установить значение – позволяет для полей-атрибутов рассчитать и записать новое значение в поле.
- Показать ошибку – позволяет вывести ошибку на интерфейс пользователя, привязанную к указанному элементу.

Ниже представлена таблица доступности типов действий для элементов формы различных типов.

Тип элемента формы	Изменить тип доступа	Установить значение	Показать ошибку
Статические элементы	+	-	-
Атрибуты	+	+	+
Операции	+	-	-
Вложенные формы	+	-	-

Рассмотрим каждый тип действия более подробно.

Изменить тип доступа

Действие предназначено для переопределения доступности и видимости элемента на форме редактирования на интерфейсе пользователя.

Действие имеет единственный дополнительный параметр **Новое значение**, который предназначен для выбора нового типа доступа, который будет установлен в рамках текущего действия.

Существуют следующие типы доступа, которые определяются в параметре Новое значение:

- Обязательное – на пользовательском интерфейсе формы редактирования элемент должен быть обязательно заполнен на момент сохранения данных в форме.
- Редактирование – на пользовательском интерфейсе формы редактирования элемент доступен для редактирования, запуска.

- Только для чтения – на пользовательском интерфейсе формы редактирования элемент отображается, но не доступен для редактирования или запуска.
- Скрытое – элемент скрыт с пользовательского интерфейса формы редактирования.

В следующей таблице представлены доступные вариации типов доступа в зависимости от типа элемента формы.

Тип элемента формы	Обязательное	Редактирование	Только для чтения	Скрытое
Статические элементы	-	+ Имеет одинаковое назначение – элемент видим на форме		+
Атрибуты	+	+	+	+
Атрибуты вычисляемые	+ Не имеет эффекта, так как атрибут доступен только для чтения	+	+	+
Операции	-	+ Операция доступна для запуска	+ Операция не доступна для запуска	+
Вложенные формы	-	+ Имеет одинаковое назначение – элемент видим на форме		+

Действие выполняется при открытии формы и после окончания внесения изменений в любое поле формы при положительном результате условия применения правила, в которое включено действие.

Новое значение типа доступа в соответствии с настроенным действием присваивается элементу формы только до следующего изменения данных на форме. Со следующим изменением данных на форме редактирования пользователем типа доступа элемента формы возвращается в исходное состояние, заданного в конструкторе форм. Все заданные правила с действиями применяются заново. Если, в соответствии с новым изменением данных, условие применения больше не выполняется и, как следствие, действие внутри правила не выполняется, тип доступа восстанавливается в исходное состояние, которое определено настройками формы.

Условие применения для типа действия «Изменить тип доступа» действует в дополнение к условию применения Правила, внутри которого находится действие. Действие выполняется, если условие применения возвращает значение «true».

Контекст условия применения зависит от того, к какому элементу формы применяется действие:

- Если элемент формы, к которому применяется действие, является элементом текущей формы и не относится к полю таблицы атрибута типа **Коллекция**, то

в качестве контекста условия выступает текущая запись, для которой открыта форма редактирования.

- Если элемент формы, к которому применяется действие является полем таблицы атрибута типа **Коллекция**, в качестве контекста выступает запись, которая относится к строке таблицы атрибута **Коллекция**.

В следующей таблице приведены некоторые бизнес-примеры по использованию действий с типом «Изменить тип доступа».

Условие	Правило	Поле	Тип действия	Новое значение
Сделать обязательным поле «Договор», если тип счета «Оплата поставки по договору»	Тип счета = «Оплата поставки по договору»	Договор	Изменить тип доступа	Обязательное
Скрыть поле «Количество вагонов», если тип отгрузки не равен «Ж/д транспорт»	Тип отгрузки != «Ж/д транспорт»	Количество вагонов	Изменить тип доступа	Скрытое
Скрыть поле «Сумма договора», если пользователь не является Менеджером по договору	Текущий пользователь = указанный Менеджер по договору»	Сумма договора	Изменить тип доступа	Только для чтения

Установить значение

Действие применимо только к элементам формы атрибутам и предназначено для изменения данных, отображаемых в полях формы редактирования.

Действие имеет единственный дополнительный параметр **Новое значение**, состоящий из двух полей: выбора типа значения и указания значения, которым необходимо обновить поле.

При настройке действия по установке значения доступны следующие варианты указания типов значений:

- Значение – означает, что в поле указания значения необходимо ввести или выбрать фиксированное значение поля. Указание значения выполняется в соответствии с типом атрибута, привязанного к полю формы, для которого указывается поле:
- Текст – отображается строка ввода текста.
- Число – отображается строка ввода числа.
- Длительность – отображается поле для ввода длительности.
- Дата / Время – отображается поле для выбора даты и времени.
- Логический – отображается как выпадающий список из двух значений.
- Документ – недоступен для изменения значения.

- Пользователь – отображается как выпадающий список с перечнем пользователей.
- Ссылка – отображается как выпадающий список с перечнем элементов шаблона записи, на который указывает ссылка.
- Коллекция – недоступна для изменения значения.
- Выражение – в качестве указанного значения будет определено расчетное выражение поля, которое будет вычислено при запуске действия и результат его выполнения возвращен в качестве значения в поле. Поле указания значения в этом случае преобразуется в поле ввода выражения на языке выражений Comindware или RDF N3 ⁽²²³⁾.

Действие выполняется при открытии формы и после окончания внесения изменений в любое поле формы при положительном результате условия применения правила, в которое включено действие.

Если в поле указания значения ничего не введено или в результате выполнения вычисления возвращено пустое значение, то по результатам выполнения действия поле очищается.

Измененное действием значение поля действует до момента изменения значения этого поля текущим (другим) действием или ручным изменением значения пользователем.

Условие применения для типа действия «Установить значение» действует в дополнение к условию применения Правила, внутри которого находится действие. Действие выполняется, если условие применения возвращает значение «true».

Контекст условия применения зависит от того, к какому элементу формы применяется действие:

- Если элемент формы, к которому применяется действие, является элементом текущей формы и не относится к полю таблицы атрибута типа **Коллекция**, то в качестве контекста условия выступает текущая запись, для которой открыта форма редактирования.
- Если элемент формы, к которому применяется действие является полем таблицы атрибута типа **Коллекция**, в качестве контекста выступает запись, которая относится к строке таблицы атрибута **Коллекция**.

В следующей таблице приведены некоторые бизнес-примеры по использованию действий с типом «Установить значение».

Условие	Правило	Поле	Тип действия	Новое значение
Указать флаг «Крупная сделка» включенным, если сумма по договору больше 1 000 000 рублей	Сумма по договору > 1 000 000	Крупная сделка	Установить значение	Значение = Да
Заполнить поле «Наименование контрагента» из выбранного в поле	Контрагент не пустое	Наименование контрагента	Установить значение	Выражение = \$Контрагент ->

«Контрагент» (ссылка) конкретное значение				наименован ие контрагента
--	--	--	--	---------------------------------

[Показать ошибку](#)

Действие применимо только к элементам формы атрибутам и предназначено для проверки и отображения сообщения пользователю о наступлении какой-либо исключительной ситуации с введенными данными.

Действие имеет следующие дополнительные параметры

- Тип сообщения – указывается тип выводимого сообщения, один из двух:
- Предупреждение – показывает предупреждение на выбранном поле с возможностью сохранения данных.
- Ошибка – показывает ошибку на выбранном поле без возможности сохранения данных, пока ошибка не будет устранена.
- Сообщение – состоящий из двух полей: выбора типа значения сообщения и указания сообщения, которое необходимо вывести в сообщении об ошибке или в текст предупреждения.

При настройке действия по установке значения сообщения доступны следующие варианты указания типов значений:

- Значение – означает, что в поле указания сообщения необходимо ввести фиксированный текст, который необходимо выводить в предупреждение или сообщение об ошибке.
- Выражение – в качестве указанного сообщения будет определено расчетное выражение поля, которое будет вычислено при запуске действия и результат его выполнения возвращен в качестве сообщения в предупреждении или ошибке. Поле указания сообщения в этом случае преобразуется в поле ввода выражения на языке выражений Comindware или RDF N3⁽²²³⁾.

Действие выполняется только при попытке сохранения формы редактирования.

Условие применения для типа действия «Установить значение» действует в дополнение к условию применения Правила, внутри которого находится действие. Действие выполняется, если условие применения на действии возвращает значение «false».

Контекст условия применения зависит от того, к какому элементу формы применяется действие:

- Если элемент формы, к которому применяется действие, является элементом текущей формы и не относится к полю таблицы атрибута типа **Коллекция**, то в качестве контекста условия выступает текущая запись, для которой открыта форма редактирования.
- Если элемент формы, к которому применяется действие является полем таблицы атрибута типа **Коллекция**, в качестве контекста выступает запись, которая относится к строке таблицы атрибута **Коллекция**.
- В следующей таблице приведены некоторые бизнес-примеры по использованию действий с типом «Показать ошибку».

Условие	Правило	Поле	Тип действия	Сообщение
Показать ошибку, если дата начала работы указана раньше даты начала проекта	Дата начала работы < Дата начала проекта	Дата начала работы	Показать ошибку	Значение = Дата начала работы не может быть раньше даты начала проекта
Показать предупреждение если выбранный контрагент имеет кредиторскую задолженность	\$Контрагент -> кредиторская задолженность <= 0	Контрагент	Показать предупреждение	Выражение = У выбранного \$Контрагент имеется кредиторская задолженность

На интерфейсе сообщение об ошибке выглядит следующим образом.

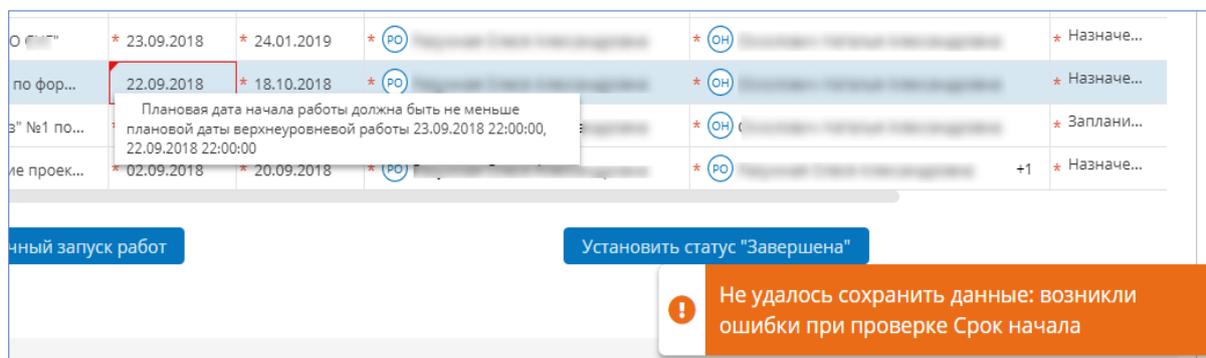


Рис. 118 Внешний вид сообщения об ошибке, настроенной на поле

Удаление правил

Для удаления правил и действий форм, необходимо:

- Находясь в разделе **Правила для формы** необходимо выбрать правило или действие и перетащить его на панель элементов. При перетаскивании, у правила или действия в виде прямоугольника должен измениться цвет контура на красный.
- Для очистки всех настроенных правил и действий можно воспользоваться кнопкой «Очистить», которая производит удаление всех настроенных правил и действий с рабочей области редактора.

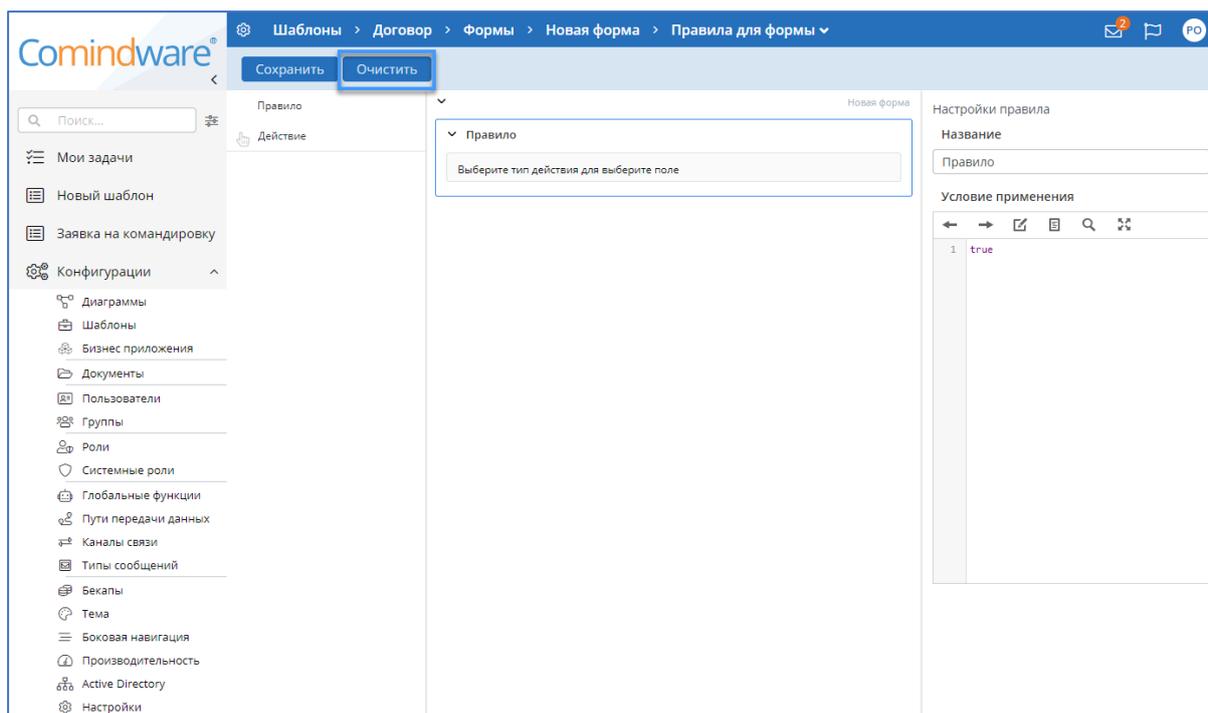


Рис. 119 Расположение кнопки «Очистить»

6. Для сохранения произведенных настроек нажать на кнопку «Сохранить».

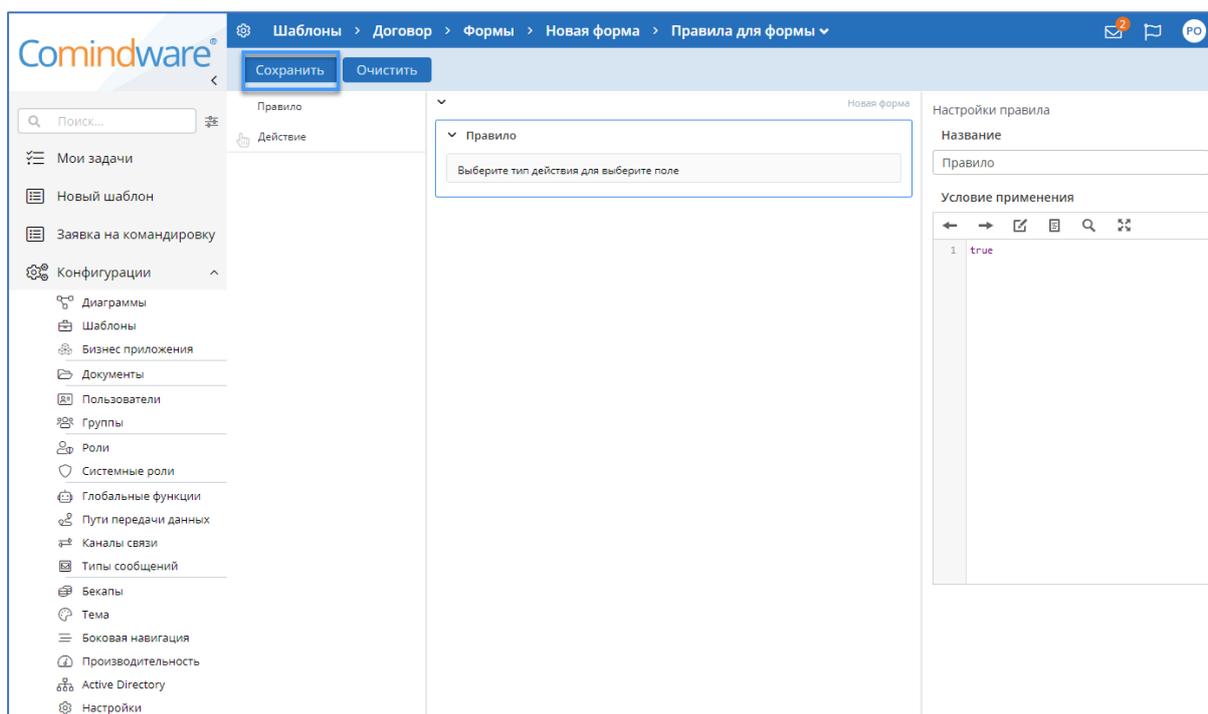


Рис. 120 Расположение кнопки «Сохранить»

2.5 Операции

Операции – это настраиваемые активные элементы платформы, которые могут выполнять предустановленные системные функции, а также произвольные функции, определенные разработчиком в виде скрипта. Операция, вынесенная на интерфейс пользователя, отображается в виде кнопки.

2.5.1 Добавление операции

Для добавления операций:

1. Перейдите в раздел **Операции** шаблона записи. Для этого кликните по иконке пользователя в правом верхнем углу и выберите пункт **Глобальные настройки**, затем перейдите в раздел **Шаблоны**, выберите нужный шаблон и перейдите в раздел **Операции**.

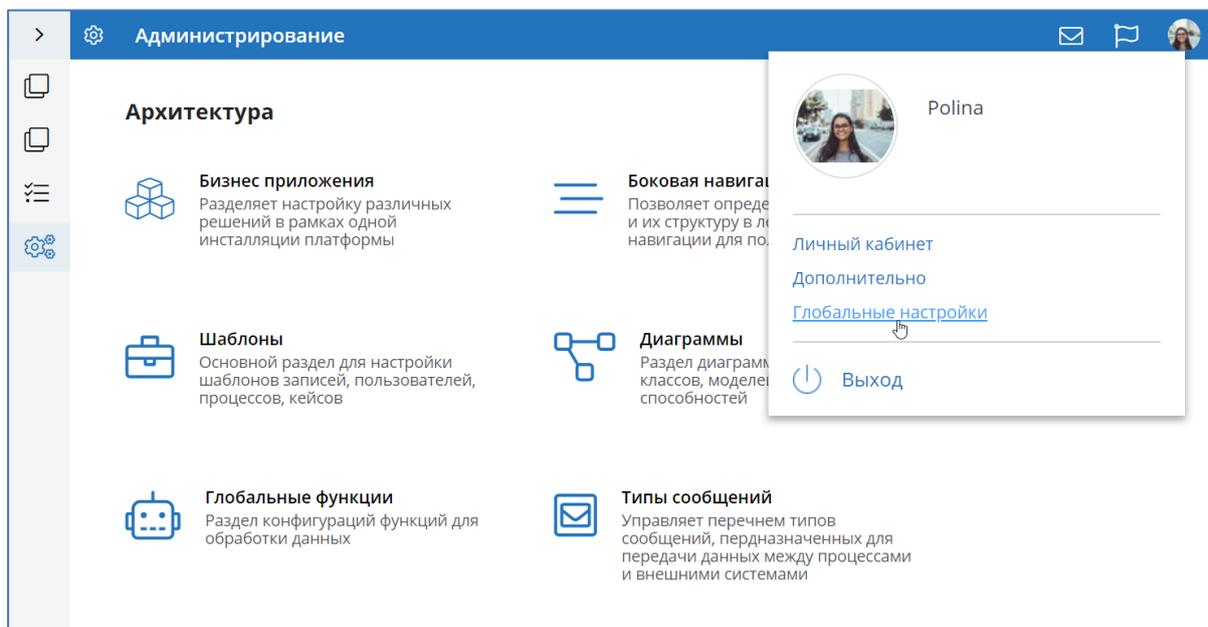


Рис. 121 Переход в раздел **Шаблоны** через глобальные настройки

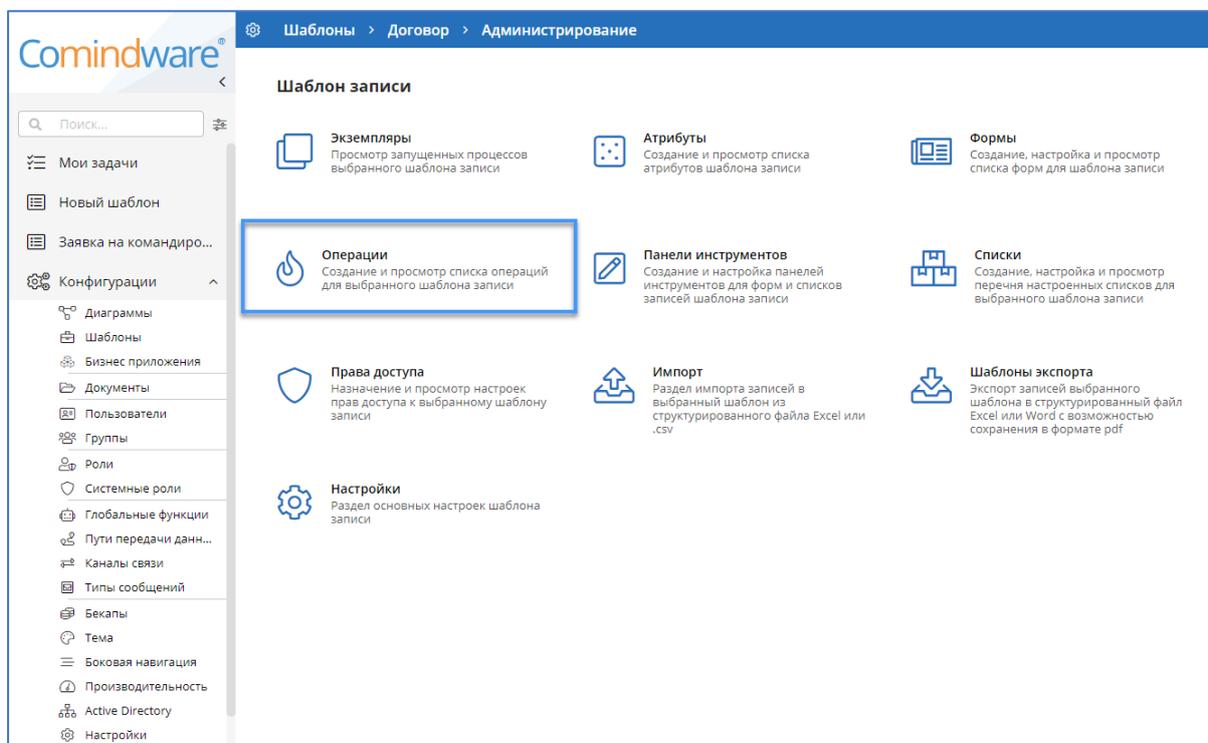


Рис. 122 Переход в раздел **Операции** выбранного шаблона записи

	Ид	Системное и...	Имя	Тип операции	Результат вы...	Сохранение после выполнения	Валидация	Требуется подтверждение...	Условие
<input type="checkbox"/>	event.23	include	Добавить в шаблон	Добавить в ш...	Обновить дан...	✓	✓		
<input type="checkbox"/>	event.22	create	Создать	Создать	Обновить дан...	✓	✓		
<input type="checkbox"/>	event.24	edit	Сохранить	Сохранить	Обновить дан...	✓	✓		
<input type="checkbox"/>	event.25	deleteAccount	Удалить	Удалить	Обновить дан...	✓	✓		
<input type="checkbox"/>	event.26	excludeAccount	Удалить из шаблона	Удалить из ша...	Обновить дан...	✓	✓		

Рис. 123 Список операций шаблона записи

Открывается список настроенных операций для шаблона записи. По умолчанию в новом шаблоне записи присутствуют следующие операции:

- Создать.
- Сохранить.
- Архивировать.

2. Для добавления новой операции нажмите кнопку «Добавить».

Рис. 124 Внешний вид формы редактирования операции

Открывается форма редактирования процесса. Более подробно описание настройки операции приведено в разделе Настройка операции (97)

3. После настройки операции нажмите кнопку «Сохранить».

2.5.2 Настройка операции

Настройка операций включает заполнение следующих данных:

- Вкладка «Общие» – общие данные об операции.
- Вкладка «Скрипт» – C# скрипт операции.

- Вкладка «Подтверждение» – информация для подтверждения выполнения операции пользователем.
- Вкладка «Условие» – условие на выполнение операции.

Для удобства работы платформа позволяет открыть окно настройки операции на полный экран.

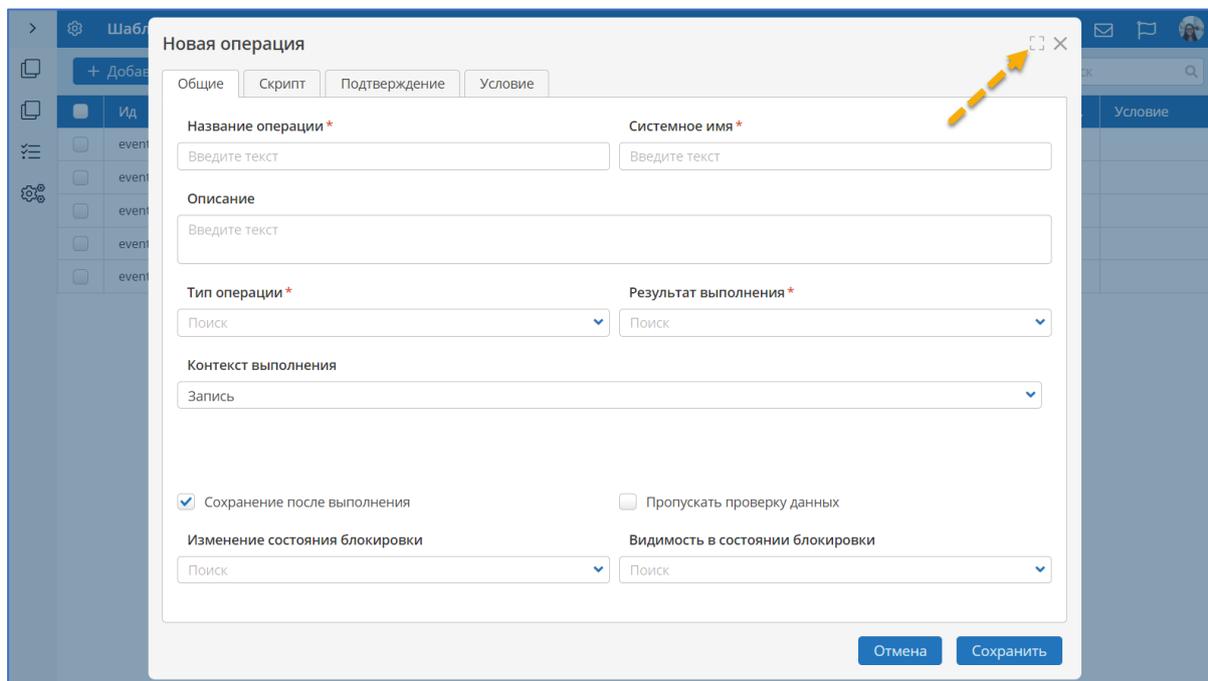


Рис. 125 Расположение кнопки раскрытия окна на весь экран

Кнопка ESC закрывает окно без подтверждения сохранения.

Общие

Вкладка Общие содержит основные характеристики операции:

- Название операции – наименование создаваемой операции.
- Системное имя – присвоенное уникальное имя операции на латинице.
- Описание – произвольное дополнительное описание о целях, назначении и функциях операции.
- Тип операции – определяет действие, которое должно запуститься при выполнении операции (нажатии на кнопку). Платформа поддерживает следующие типы операций:
 - С# скрипт – произвольный скрипт на языке С#, которые пишется разработчиком и может выполнять любые действия в пределах разрешенных библиотек в платформе.
 - Запустить процесс – производит запуск выбранного процесса. Выбор запускаемого процесса становится доступным в отдельном поле «Шаблон».
 - Запустить связанный процесс – производит запуск процесса, ассоциированный шаблон записи которого связан с текущим шаблоном записи с помощью атрибута типа «Ссылка». Отличие от типа операции «Запустить процесс» заключается в том, что после запуска процесса, ссылка

на текущую сущность в ассоциированной с запущенным процессом записью будет заполнена. Выбор запускаемого процесса становится доступным в отдельном поле «Шаблон», который ограничивает перечень связанных процессов для запуска наличием ссылки на текущий шаблон и в котором также указывается атрибут, который ссылается на текущую сущность. Например, есть шаблон записи Договор и есть процесс создания дополнительного соглашения, у которого есть ассоциированный шаблон записи Дополнительные соглашения. Если в шаблоне записи Дополнительное соглашение создать атрибут-ссылку на Договор, то в договоре можно будет создать операцию запуска связанного процесса создания дополнительного соглашения.

- Запустить кейс – производит запуск выбранного кейса. Выбор запускаемого кейса становится доступным в отдельном поле «Шаблон».
- Запустить связанный кейс – производит запуск кейса, ассоциированный шаблон записи которого связан с текущим шаблоном записи с помощью атрибута типа «Ссылка». Отличие от типа операции «Запустить кейс» заключается в том, что после запуска кейса, ссылка на текущую сущность в ассоциированной с запущенным экземпляром кейса записью будет заполнена. Выбор запускаемого кейса становится доступным в отдельном поле «Шаблон», который ограничивает перечень доступных кейсов для запуска наличием ссылки на текущий шаблон и в котором также указывается атрибут, который ссылается на текущую сущность. Например, есть шаблон записи Протокол и есть кейс создания пункта протокола, у которого есть ассоциированный шаблон записи Пункт протокола. Если в шаблоне записи Пункт протокола создать атрибут-ссылку на Протокол, то в шаблоне записи Протокол можно будет создать операцию запуска связанного кейса создания пункта протокола.
- Экспорт записи – операция, которая позволяет осуществлять создание формы записи в виде файла MS Word, MS Excel или PDF и скачать. Более подробно о создании шаблона будет рассказано в разделе «Шаблоны экспорта» (2.9).
- Экспорт списка – операция, которая позволяет выгрузить в файл MS Excel данные текущего открытого списка записей с учетом примененных параметров фильтрации, сортировки, группировки и агрегации.
- Архивировать – архивирует выбранные записи (устанавливает значение системного атрибута `_isDisabled` в значение `true`).
- Восстановить из архива – восстанавливает архивные записи (устанавливает значение системного атрибута `_isDisabled` в значение `false`).
- Создать – операция создает запись выбранного шаблона записи. Выбор создаваемого шаблона записи становится доступным в отдельном поле «Создать запись». После выбора шаблона записи в вышеназванном поле, пользователю становится доступно поле выбора формы создания записи.
- Создать связанную запись – производит создание записи, шаблон которой связан с текущим шаблоном записи с помощью атрибута типа «Ссылка». Отличие от типа операции «Создать» заключается в том, что после создания записи, ссылка на текущую запись будет заполнена в созданной записи.

Выбор запускаемого шаблона записи становится доступным в отдельном поле «Шаблон», который ограничивает перечень доступных шаблонов записи для создания. Например, есть шаблон записи Накладная и есть шаблон записи Строка накладной. Если в шаблоне записи Строка накладной создать атрибут-ссылку на накладную, то в шаблоне записи Накладная можно будет создать операцию создания строки накладной с автоматическим заполнением ссылки на накладную, из которой создается строка.

- Добавить в шаблон – не используется для шаблона записи.
- Удалить из шаблона – не используется для шаблона записи.
- Сохранить – осуществляет сохранение данных формы шаблона записи.
- Удалить – удаляет запись без возможности восстановления. Если на запись присутствовали ссылки, то платформа обрабатывает их в соответствии с настроенным правилом удаления.
- Завершить задачу – не используется для шаблона записи.
- Открыть заново – не используется для шаблона записи.
- Отложить выполнение – не используется для шаблона записи.
- Принять – не используется для шаблона записи.
- Подписать – не используется для шаблона записи.
- Отписаться – не используется для шаблона записи.
- Результат выполнения – выбирается тип результата, который система выдает пользователю после завершения выполнения операции.
 - Обновить данные – после завершения операции выполняется обновление данных на форме или списке.
 - Сообщение – используется для C# операций для вывода соответствующего сообщения пользователю.
 - Навигация – после завершения операции осуществляется переход на выбранный объект навигации. Выбор объекта навигации осуществляется в поле «Объект навигации». Для операции типа «C# скрипт» объект навигации определяется в тексте скрипта в выходном параметре.
 - Текущая запись – платформа остается на текущем месте.
 - Связанная запись – платформа открывает созданную запись.
 - Экземпляр процесса – платформа открывает карточку запущенного процесса (актуально для операций запуска процессов и связанных процессов).
 - Задача запущенного процесса – открывается карточка пользовательской задачи запущенного процесса (актуально для операций запуска процессов и связанных процессов). Если процесс не передал управление пользовательской задаче, то осуществляется перевод в список задач.

-
- Следующая задача в списке – платформа открывает следующую задачу из списка задач на выполнение (актуально для операций управления задачами процесса).
 - Список – открывается список записи, с которой осуществлялся вызов операции.
 - Список задач – открывается список задач пользователя.
 - Скачать документ – в результате выполнения операции пользователю выгружается результирующий файл (актуально для операций Экспорт записи, Экспорт списка и С# скрипт). Для типа операции «С# скрипт» файл, скачиваемый в интернет обозревателе определяется возвращаемым параметром С# скрипта.
 - Контекст выполнения – определяет контекст, который будет доступен в ходе выполнения операции. Для шаблона записи актуален только контекст выполнения Запись.
 - «Сохранение после выполнения» – флаг, определяющий, выполнять ли запись транзакции по изменению данных, после выполнения операции. Если флаг отключен, то сохранения данных при покидании формы (например, при результате выполнения «Навигация») не будет произведено. Аналогично, при запуске операции со списка, с незаполненным флагом, изменение данных применено не будет.
 - «Пропускать проверку данных» – данный флаг необходим для выполнения операций на форме редактирования без применения проверок обязательности полей и проверочных правил формы с типом действия «Показать ошибку».
 - «Изменение состояния блокировки» - изменение режима отображения формы на один из следующих типов:
 - «Чтение»;
 - «Редактирование»;
 - «Не изменять».
 - «Видимость в состоянии блокировки» - определение видимости операции на форме в состоянии блокировки (например, когда настроен запрет на редактирование формы несколькими пользователями). Доступны следующие режимы отображения:
 - «Чтение»;
 - «Редактирование»;
 - «Всегда».

Скрипт

Скрипт – это раздел **настройки** (вкладка), позволяющая пользователю определить произвольную процедуру на языке С#, которая может выполнять любые действия по контролю, чтению, изменению, удалению данных внутри системы, а также вызову внешних сервисов в ограниченном пространстве подключаемых библиотек.

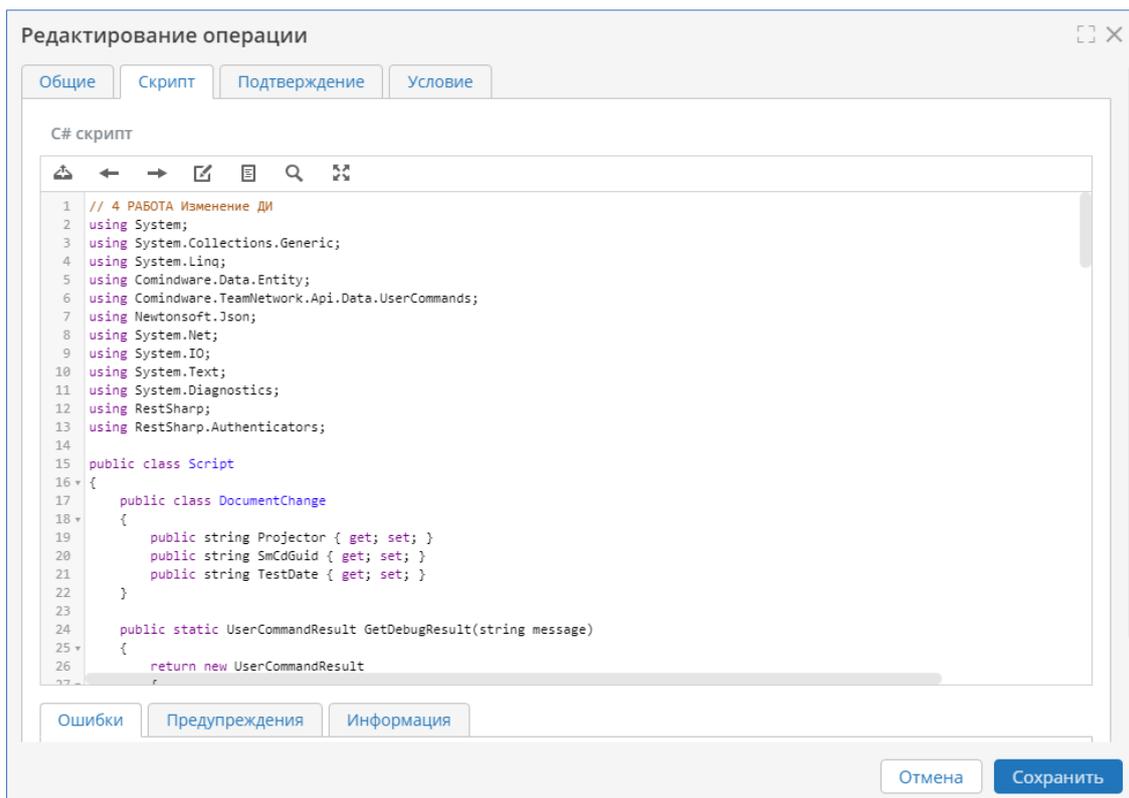


Рис. 126 Внешний вид раздела разработки C# скрипта

C# скрипт может так же использоваться для типа операции «Экспорт записи» с целью подготовки данных для вывода на шаблон экспорта.

Раздел **Скрипт** содержит блок ввода C# скрипта.

- **Ограничение.** Блок с вкладками «Ошибки», «Предупреждения» и «Информация» является перспективной разработкой по развитию функционала отладки скриптов и не отображает какой-либо информации в настоящее время.

Скрипт получает на вход в качестве контекста:

- массив одной или нескольких ID записей, выбранных на списке, а также форма редактирования которой открыта;
- массив выбранных записей в коллекции на форме редактирования;
- глобальный контекст «Entity» для доступа к модели данных.

Выходной параметр должен содержать следующие параметры:

- процедура закончилась успехом или нет;
- сохранять результаты работы процедуры или нет;
- тип результата;
- сообщение, которое нужно вывести пользователю;
- контент, который необходимо отдать на скачивание;
- параметры навигации для открытия раздела.

Более подробно о конструкции C# скрипта, описании входящих и исходящих параметров описано в руководстве разработчика.

Подтверждение

Подтверждение операции – это механизм, позволяющий запросить у пользователя подтверждение выполнения операции. Настроенное подтверждение является своеобразной защитой от случайного нажатия на кнопку на интерфейсе пользователя.

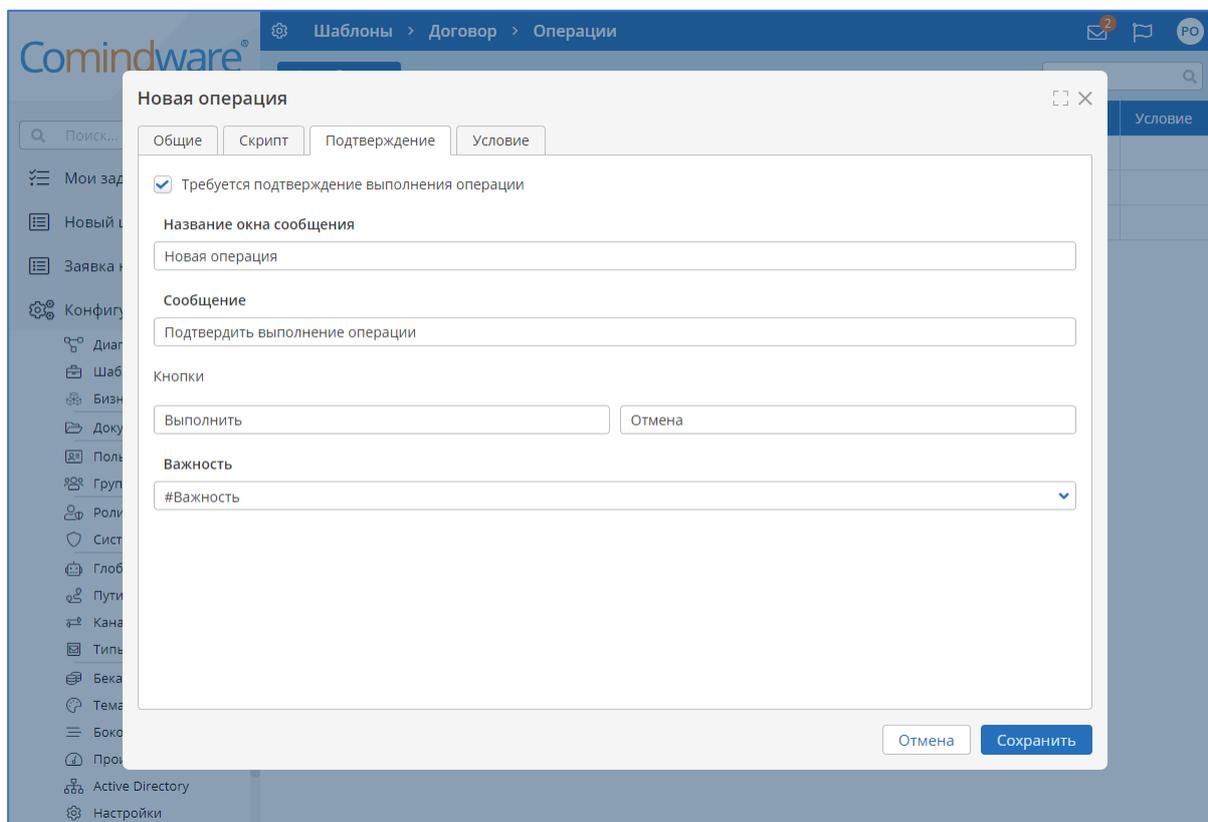


Рис. 127 Внешний вид раздела настройки подтверждения операции

Для настройки подтверждения, необходимо заполнить следующие данные:

- «Требуется подтверждение выполнения операции» – флаг, включающий и отключающий механизм подтверждения.
- Название окна сообщения – текст показывается на интерфейсе пользователя в качестве заголовка окна подтверждения.
- Сообщение – основной текст окна подтверждения.
- Кнопки – определяется текст двух кнопок подтверждения и отмены выполнения операции. По умолчанию заполнено «Выполнить» и «Отмена».
- Важность – будущая разработка по настраиваемой цветовой гамме и расширению возможностей подтверждений. В настоящий момент не используется.

Внешний вид окна подтверждения, которое отображается на интерфейсе пользователю выглядит следующим образом:

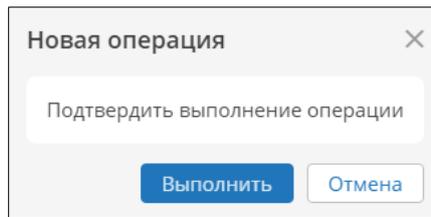


Рис. 128 Внешний вид окна подтверждения выполнения операции для пользователя

Условие

Позволяет определить условие видимости кнопки, относящейся к операции на интерфейсе пользователя.

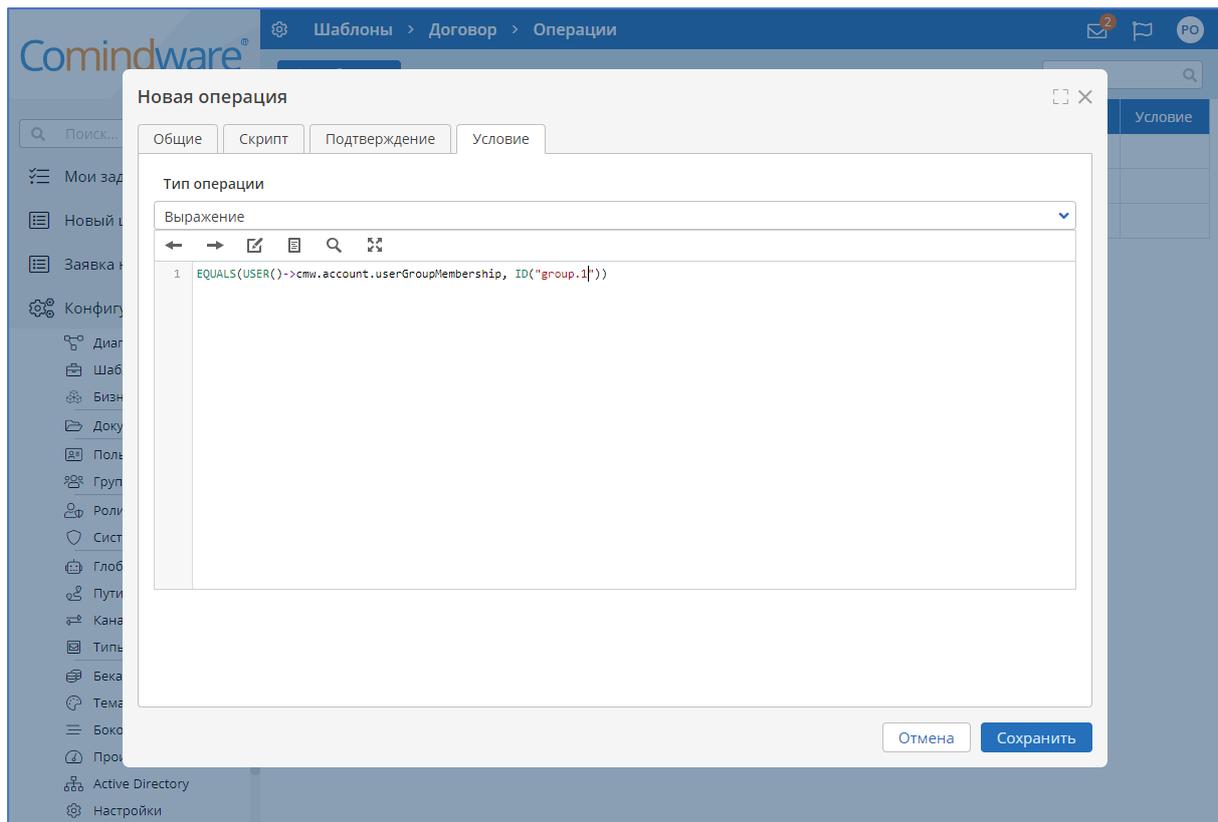


Рис. 129 Внешний вид раздела настройки условия видимости операции

Настройка условия отображения операции выполняется двумя характеристиками:

- Тип операции – определяет, каким образом выполняется проверка условия:
- Не задано – условие не определено, и кнопка отображается безусловно.

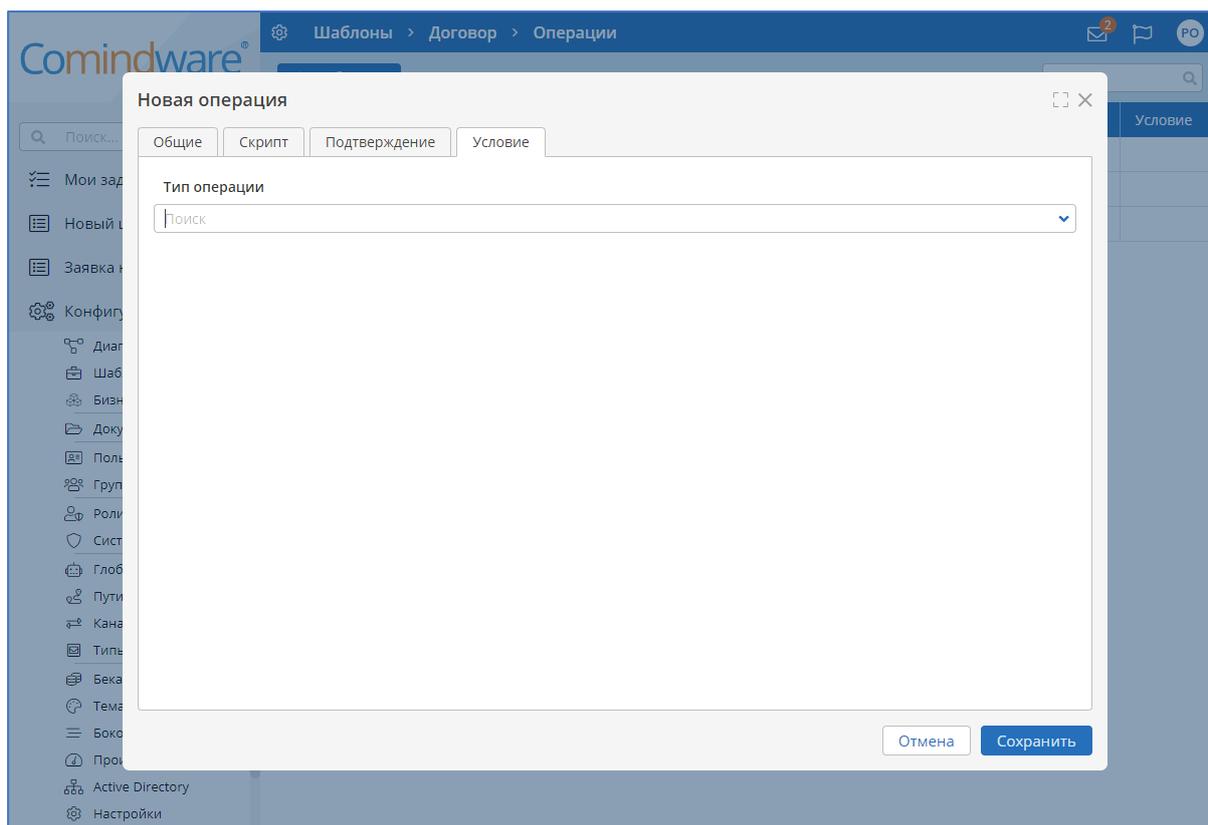


Рис. 130 Внешний вид незаданного условия отображения операции

- **Выражение** – определяет условие отображения с использованием языка выражений Comindware. Более подробно язык выражений Comindware описан в разделе Язык выражений Comindware Business Application Platform (стр. 223). Выражение должно возвращать «true», если операция должна быть видна на интерфейсе пользователя.

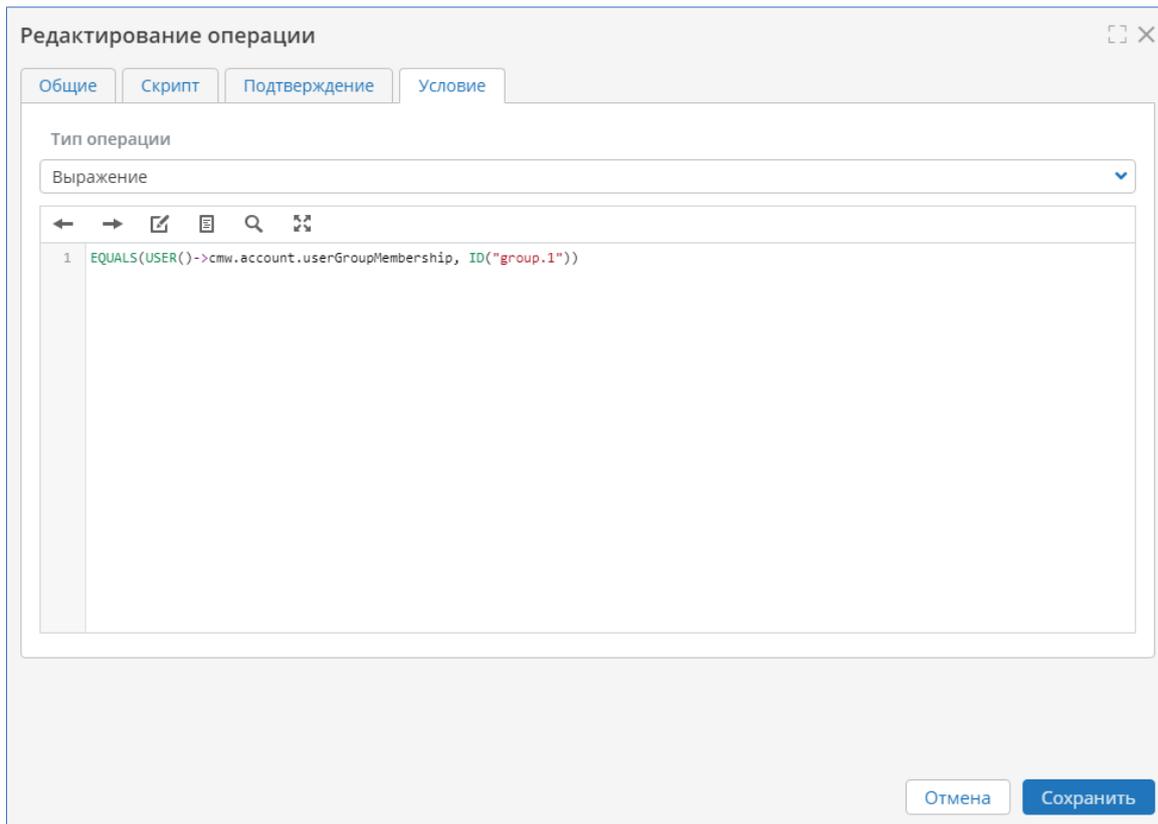


Рис. 131 Внешний вид окна настройки выражения условия видимости операции

- C# скрипт – условие отображения определяется с использованием написанного скрипта на языке C#. Более подробно о разработке скрипта условия отображения кнопки можно познакомиться в руководстве разработчика.

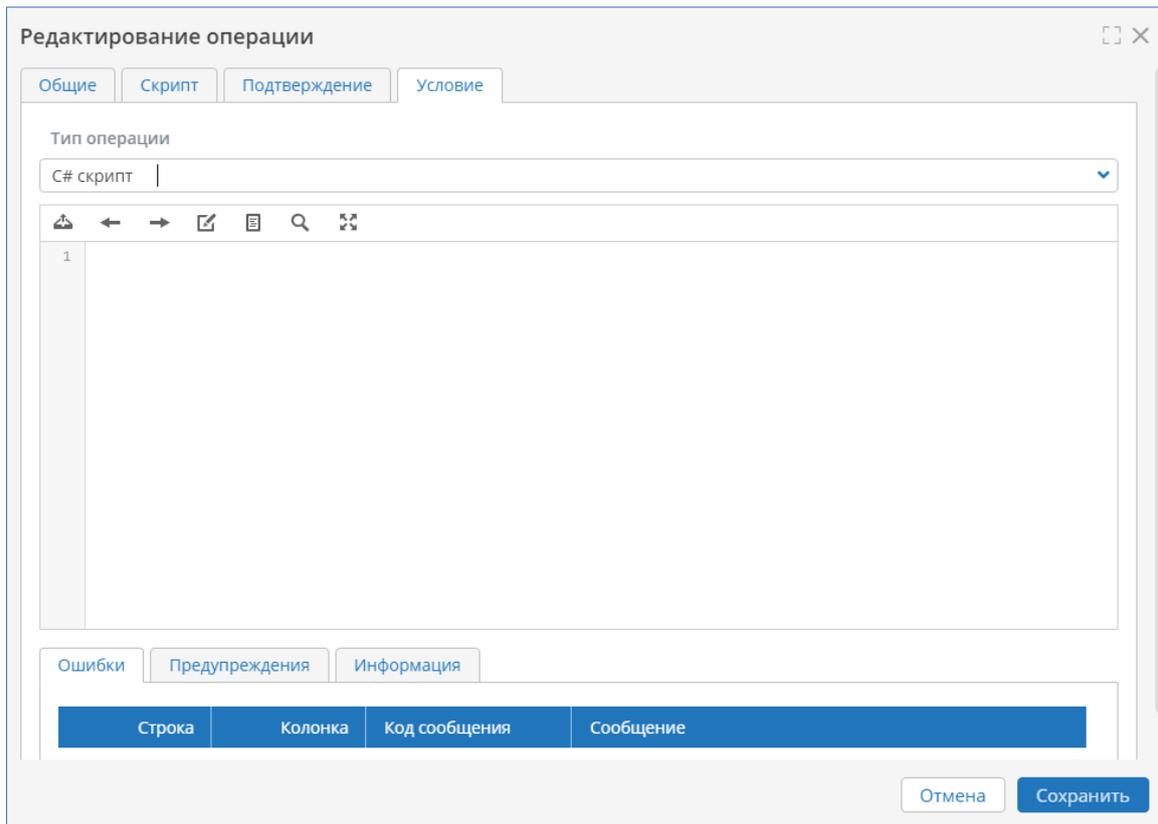


Рис. 132

- Ограничение. Блок с вкладками «Ошибки», «Предупреждения» и «Информация» является перспективной разработкой по развитию функционала отладки скриптов и не отображает какой-либо информации в настоящее время.
- Блок для задания выражения или скрипта – окно ввода текста выражения или скрипта. Выражение или скрипт должны возвращать «true», если операция должна быть видна на интерфейсе пользователя.

Если для операции определено условие отображения кнопки, это значит, что для кнопки существует контекст выбранной записи. В этом случае кнопка не будет отображаться на списке записей, если на нем не выбран ни один элемент.

Условие обрабатывает для каждого объекта независимо, поэтому если хоть один в списке выбранных объектов не удовлетворяет условию, то кнопка не отображается вообще.

2.5.3 Удаление операции

Для удаления операции необходимо выполнить следующие действия:

1. Перейти в список операций (2.5.1) и выбрать нужную операцию для удаления.

В панели инструментов появится кнопка «Удалить».

2. Нажать на кнопку **Удалить**.

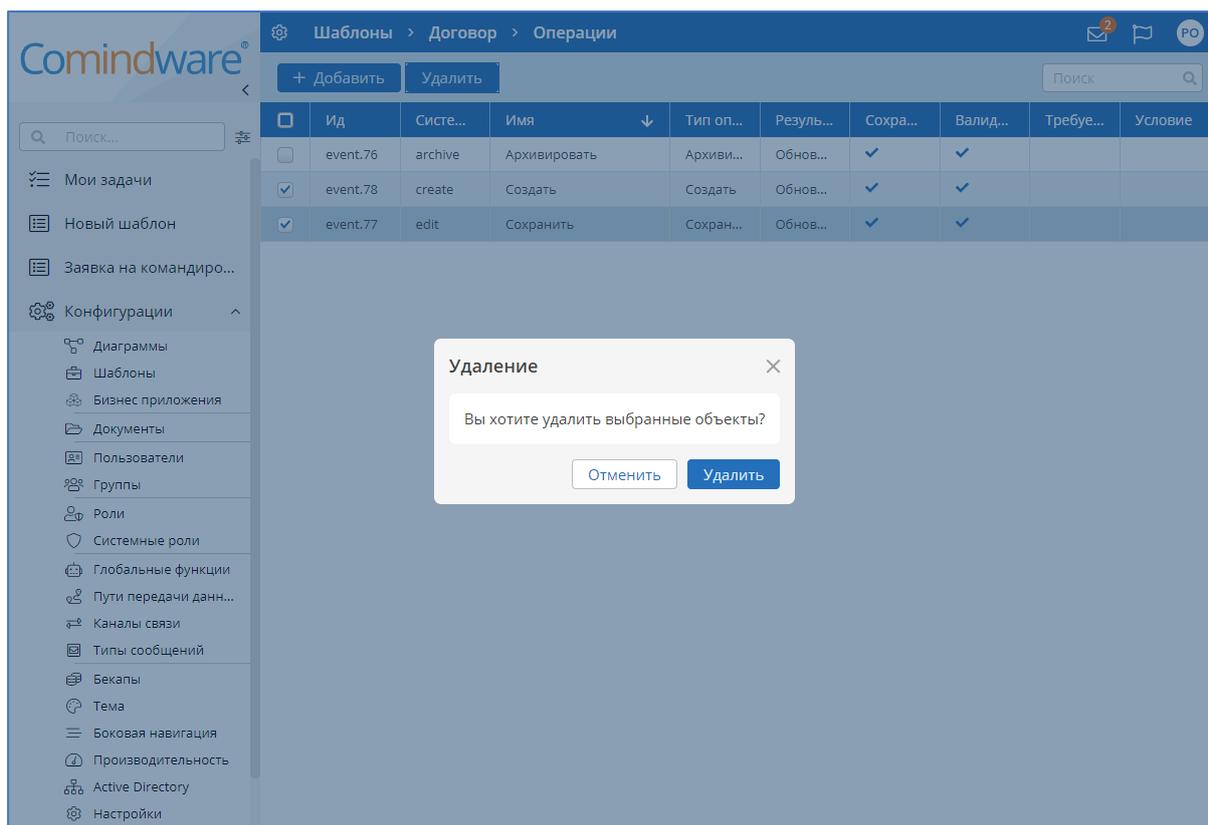


Рис. 133 Окно подтверждение удаления выбранных операций

Открывается окно подтверждения удаления операций.

- Для подтверждения удаления необходимо нажать на кнопку «Удалить». Для отмены удаления нажать на кнопку «Отменить».

2.6 Панель инструментов

Раздел «Панель инструментов» позволяет настроить отображаемые кнопки на панели инструментов списка или формы редактирования шаблона записи.

Настройка панели инструментов выполняется в визуальном дизайнера и позволяет мышью сформировать произвольную панель инструментов, удобную для конечного пользователя.

По умолчанию платформа для каждого нового шаблона записи создает две панели инструментов, одна для отображения на списках, вторая для отображения на формах.

Шаблон записи может содержать несколько настроенных панелей инструментов, однако на пользовательском интерфейсе отображаются только панели инструментов, назначенные по умолчанию.

2.6.1 Добавление панели инструментов

Для добавления новой панели инструментов необходимо выполнить следующие действия:

- Войти в раздел «Панели инструментов».

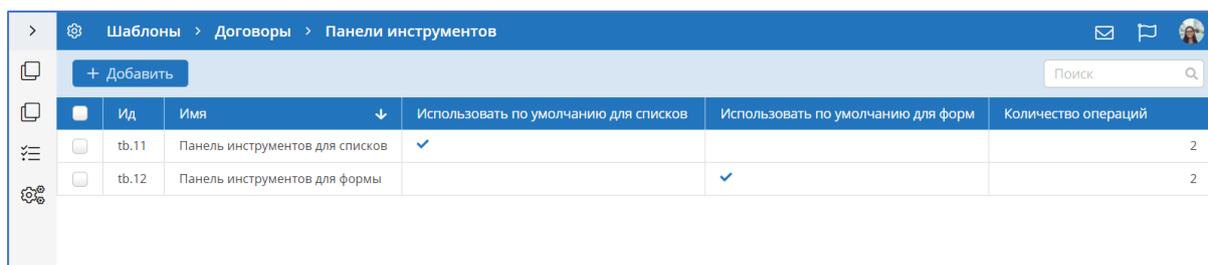


Рис. 134 Расположение кнопки **Добавить**

2. Для добавления новой панели инструментов нажать на кнопку **Добавить**.

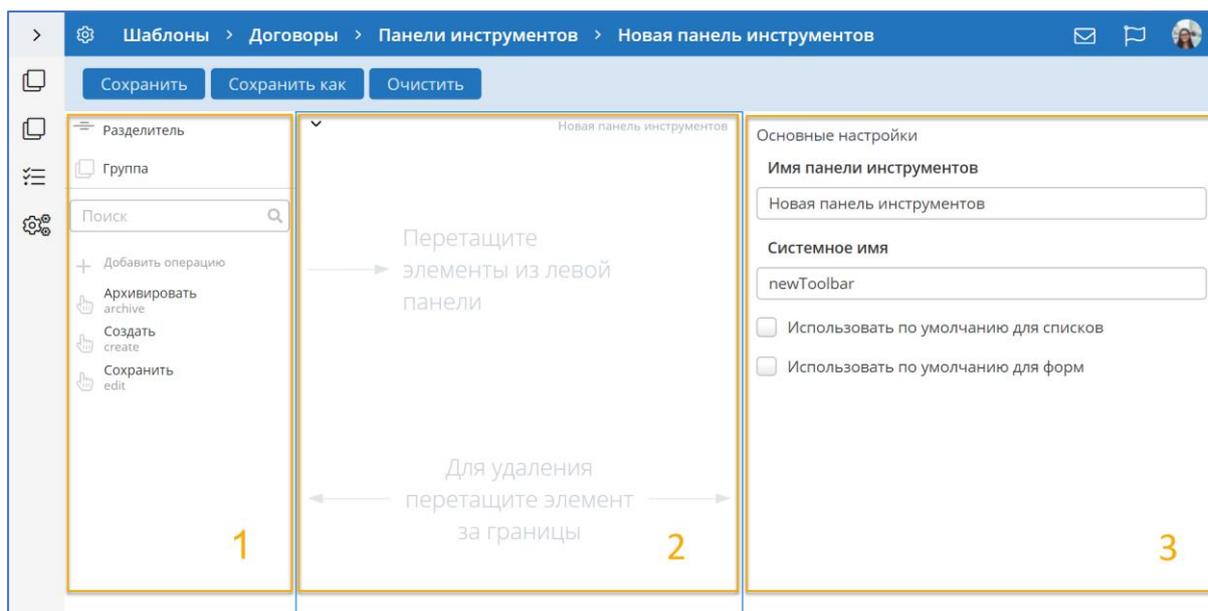


Рис. 135 Внешний вид конструктора панелей инструментов

На пользовательском интерфейсе открывается конструктор панели инструментов.

Конструктор панели инструментов разбит на 3 области:

- **1.** Панель элементов – отображается перечень доступных элементов для добавления в рабочую область конструктора панели инструментов.
- **2.** Рабочая область – основная визуальная область конструктора форм, где производится визуальное наполнение и конструирование внешнего вида панели инструментов.
- **3.** Панель настроек – отображается набор характеристик выбранного элемента в рабочей области. Для выбора доступны сама форма, а также любой элемент, помещаемый в рабочую область. По умолчанию, когда нет ни одного элемента выбрана и, следовательно, отображаются характеристики формы.

Подробное описание конструктора панели инструментов приведено в разделе «Настройка панели инструментов» (стр. 110).

3. После настройки панели сохраните, нажав на кнопку «Сохранить».

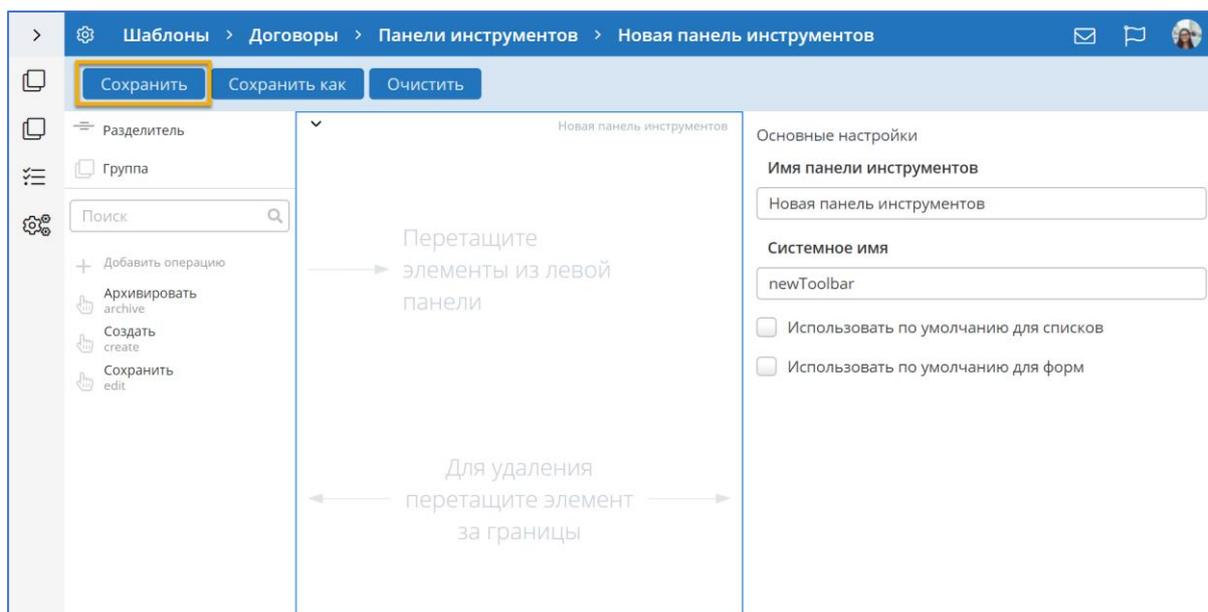


Рис. 136 Расположение кнопки «Сохранить»

4. Для сохранения панели как копии под другим именем нажмите на кнопку «Сохранить как».

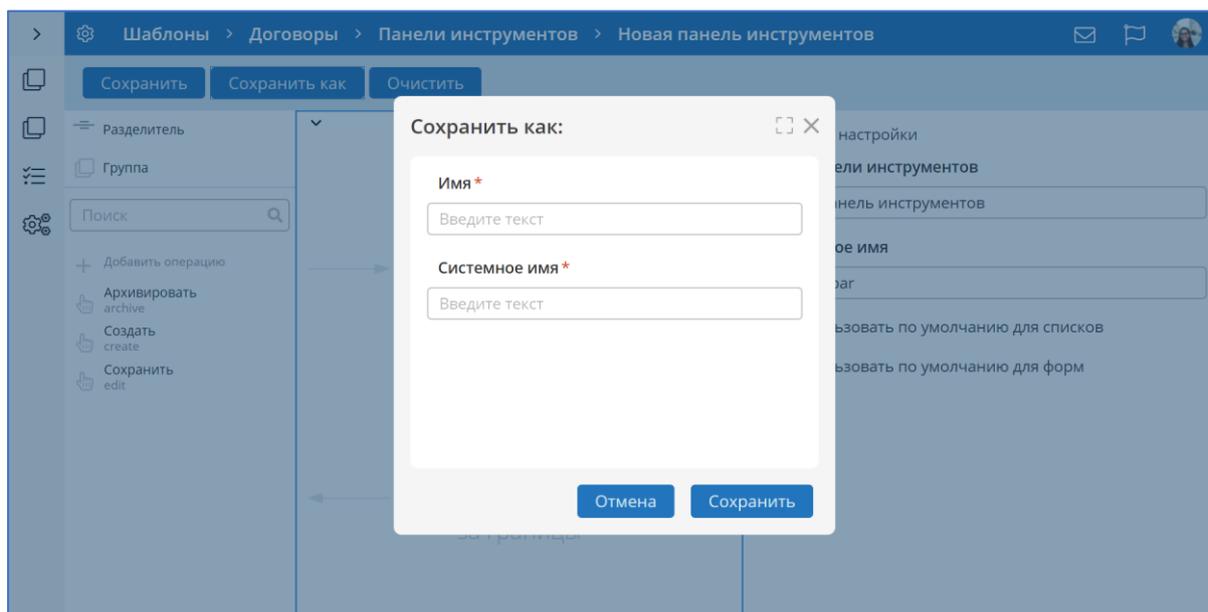


Рис. 137 Форма ввода реквизитов новой панели инструментов

На открывшемся окне необходимо заполнить наименование и системное имя панели инструментов.

5. Для сохранения новой панели инструментов нажмите на кнопку «Сохранить». Для отмены сохранения новой формы, необходимо нажать на кнопку «Отмена».

2.6.2 Настройка панели инструментов

Основная работа по настройке будущей панели инструментов выполняется в визуальном конструкторе. Разработка панели производится путем вынесения из панели элементов на рабочую область необходимых элементов управления и настройки их характеристик.

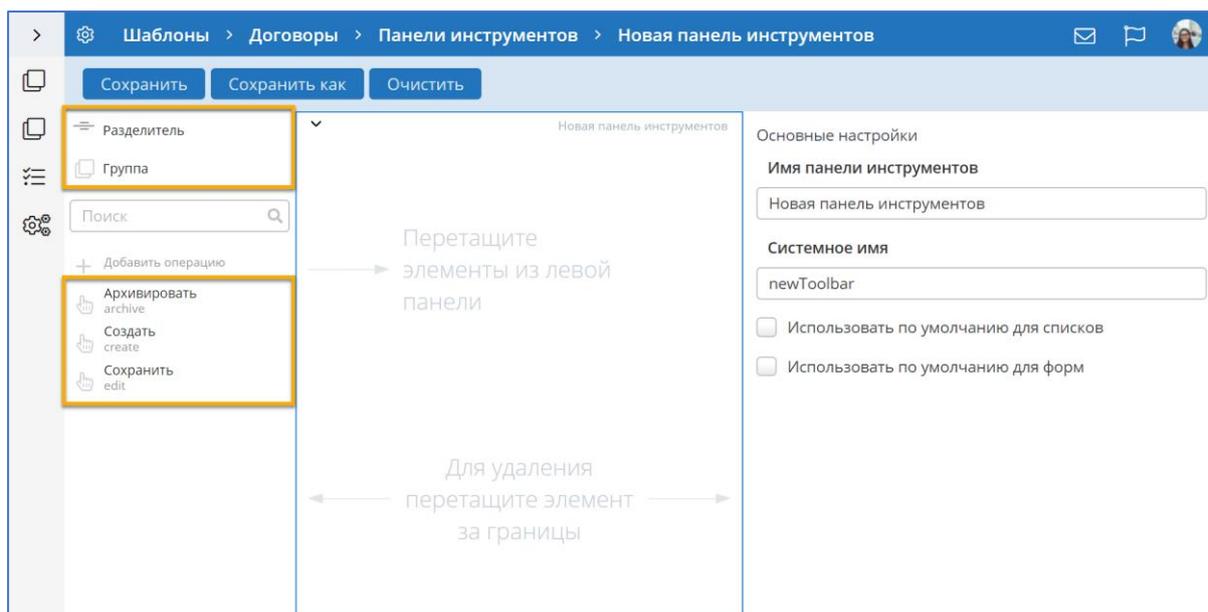


Рис. 138 Фиксированные и динамические элементы панели инструментов

Для панели инструментов могут быть определены следующие характеристики:

- Имя панели инструментов – наименование панели, которое отображается в списке и несет смысловую нагрузку цели созданной панели.
- Системное имя – уникальное текстовое наименование панели в рамках шаблона записи. Может содержать только буквы латинского алфавита, цифры и знак «_».
- «Использовать по умолчанию для списка» – флаг, который показывает, что данная панель инструментов будет отображаться на списках. Флаг может быть указан только у одной панели инструментов в рамках одного шаблона записи. Если флаг указывается на новой панели, он снимается с другой панели, на которой данный флаг установлен.
- «Использовать по умолчанию для форм» – флаг, который говорит, что данная панель инструментов отображается на всех формах в рамках шаблона записи. Флаг может быть указан только у одной панели инструментов в рамках шаблона записи. Если флаг устанавливается пользователем на одной панели, то существующий на другой панели снимается в автоматическом режиме.

Панель элементов конструктора содержит следующие элементы (Рис. 138):

- Фиксированные элементы – это набор визуальных элементов, которые не зависят от шаблона записи, на котором настраивается панель инструментов и позволяют визуально улучшить размещение операций (кнопок) на панели инструментов.
- Динамические элементы – это набор операций, настроенных на текущем шаблоне записи, позволяющих вынести кнопки на форму редактирования.

Добавление элементов на панель инструментов производится путем «перетаскивания» элементов с панели элементов конструктора в рабочую область и определения индивидуальных характеристик визуальных элементов с помощью панели настройки.

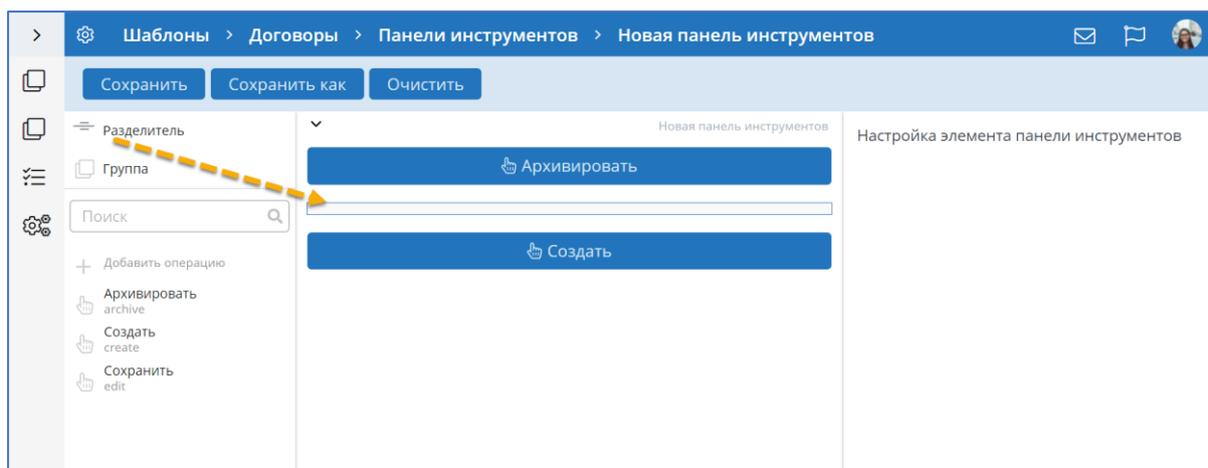


Рис. 139 Механизм добавления визуальных элементов на форму

Выбор визуального элемента, который необходимо вынести на форму производится с помощью указателя мыши нажатия и удерживания левой кнопки мыши на элементе. Удерживая левую кнопку мыши, необходимо начать движение мышью и в рабочей области синими зонами подсвечиваются доступные для переноса области. Чтобы завершить перенос, необходимо отпустить удерживаемую левую кнопку мыши элемент в доступной синей зоне (при этом она окрасится в темно-синий цвет). Таким образом визуальный элемент будет размещен в нужном месте рабочей области, который затем аналогично отобразится в виде кнопки на панели инструментов списка или формы.

Чтобы изменить положение уже добавленного визуального элемента в рабочей области, необходимо аналогичным образом осуществить перенос элемента в нужное место.

Для описания характеристик визуальных элементов, необходимо сделать щелчок на нужном элементе в рабочей области конструктора панелей инструментов. Выбранный элемент выделяется цветом и в панели настройки появляется перечень доступных характеристик, которые доступны проектировщику для редактирования.

Также имеется возможность посмотреть характеристики текущей панели инструментов, нажав левую кнопку мыши в любом свободном от добавленных элементов месте рабочей области.

Динамические визуальные элементы на панели элементов конструктора содержат строку поиска по наименованию и системному имени операции.

Ниже представлено описание визуальных элементов панели элементов конструктора панели инструментов.

Разделитель

Предназначен для визуального разделения кнопок на панели инструментов. Элемент не имеет характеристик.

На панели инструментов в пользовательском интерфейсе элемент **Разделитель** выглядит следующим образом.

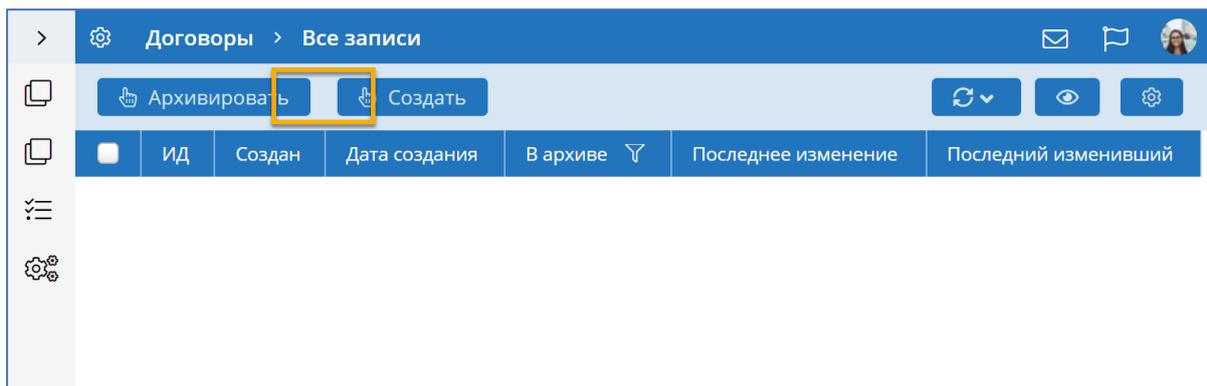


Рис. 140 Элемент «Разделитель» на пользовательском интерфейсе

Группа

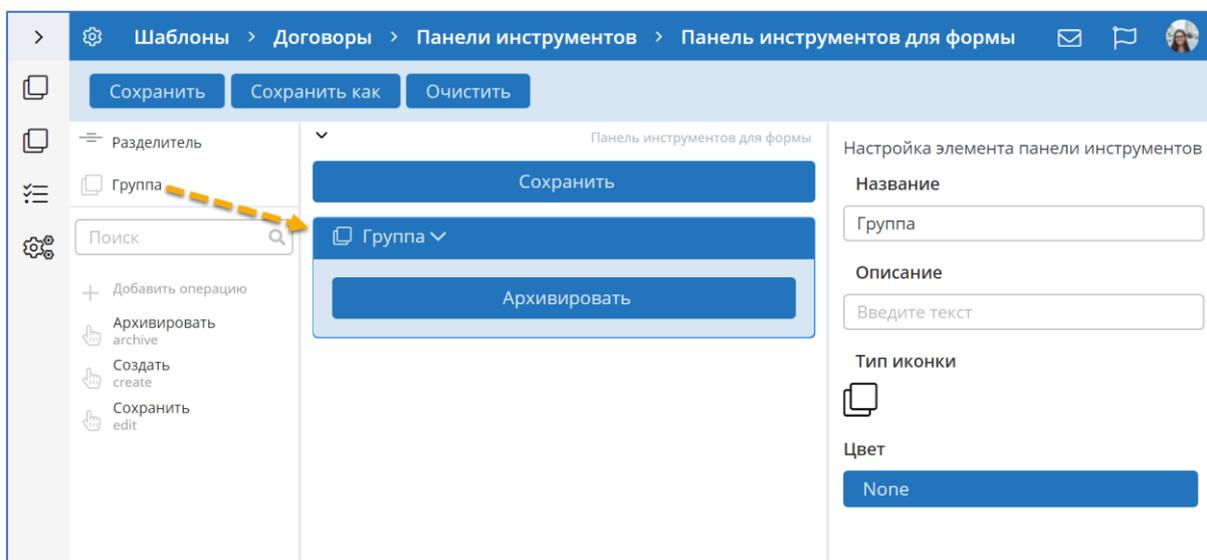


Рис. 141 Элемент «Группа»

Предназначена для формирования папки с выпадающим списком, объединяющую в себе группу операций.

Группа имеет следующие характеристики для настройки:

- Названия – отображаемое на интерфейсе пользователя наименование папки.
- Описание – детальное описание папки. Выводится на интерфейсе пользователя как подсказка при наведении курсора мыши на папку.
- Тип иконки – выбирается иконка из фиксированной библиотеки иконок платформы. Будет отображена на папке на интерфейсе пользователя.
- Цвет – указывается цвет из приведенного набора цветов, которым будет окрашена папка на интерфейсе пользователя.

На панели инструментов в пользовательском интерфейсе элемент **Группа** отображается следующим образом.

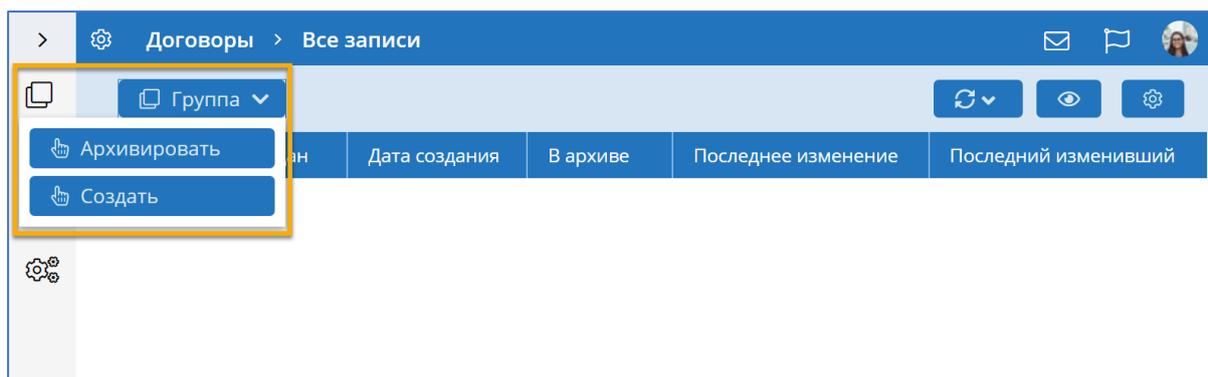


Рис. 142 Элемент «Группа» на пользовательском интерфейсе.

Папка не будет выведена на интерфейс пользователя, если в ней отсутствуют кнопки, доступные для отображения, в том числе не отображаемые согласно заданному условию.

Операции

Динамические элементы панели элементов наполняются только перечнем созданных операций в границах текущего шаблона записи.

Подробное описание настройки операций приведено в разделе «Операции» (стр. 87). При двойном щелчке мышью по элементу в панели элементов открывается окно настройки операции.

При добавлении операции в рабочую область конструктора панели инструментов на панели настройки выводится общая информация об операции, которая позволяет идентифицировать, какая операция была вынесена.

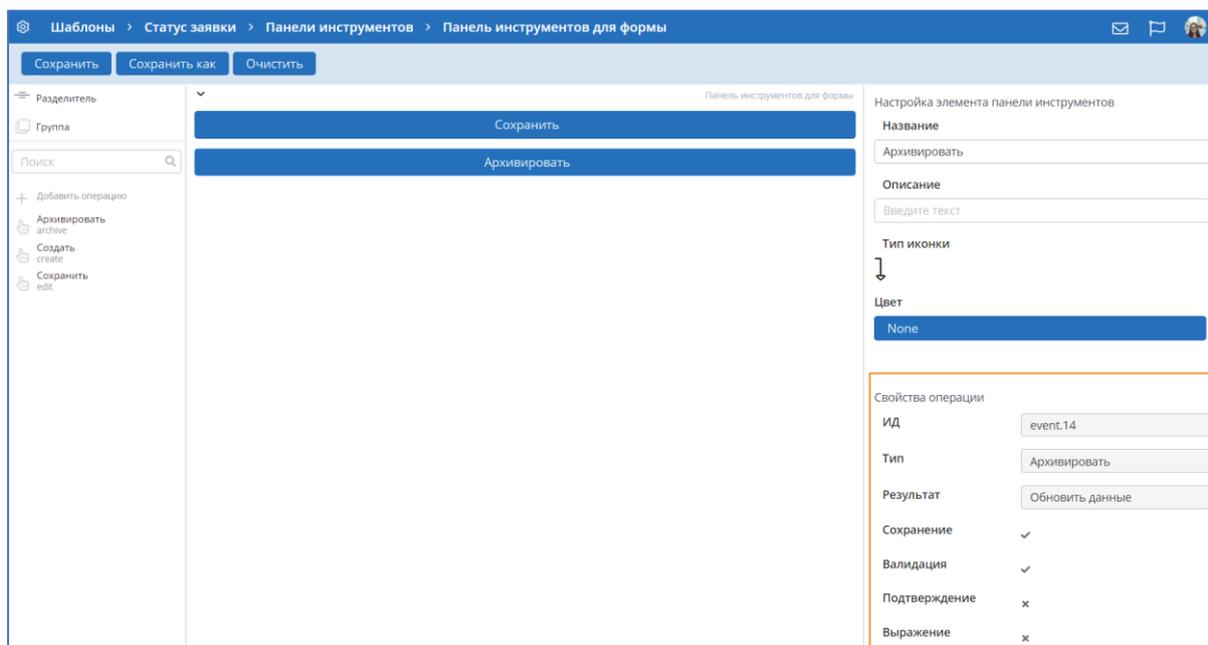


Рис. 143 Информация об операции на панели настройки

Проектировщик имеет возможность переопределить следующие характеристики кнопки с помощью панели настройки:

- Название – отображаемая на интерфейсе пользователя текстовая надпись, наносимая на кнопку.

- Описание – дополнительное описание кнопки, которое выводится в качестве подсказки, при наведении на кнопку курсором мыши.
- Тип иконки – выбирается иконка из фиксированной библиотеки иконок платформы. Будет отображена на кнопке на интерфейсе пользователя.
- Цвет – указывается цвет из приведенного набора цветов, которым будет окрашена кнопка на интерфейсе пользователя.

На панели инструментов кнопка будет отображена только, если она должна отображаться согласно условию отображения.

2.6.3 Удаление панели инструментов

Для удаления панели инструментов необходимо выполнить следующие действия:

1. Перейти в список панелей инструментов и выбрать нужную панель для удаления.



Рис. 144 Список панелей инструментов с выбранными на удаление

В панели инструментов должна появиться кнопка «Удалить».

2. Нажать на кнопку «Удалить».

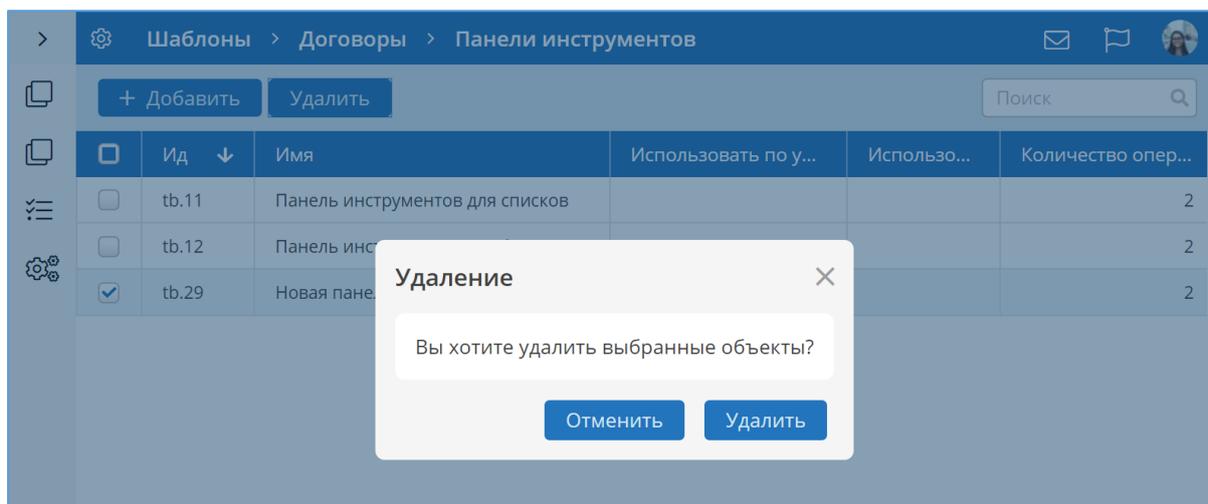


Рис. 145 Окно подтверждение удаления выбранных операций

Открывается окно подтверждения удаления панели инструментов.

3. Для подтверждения удаления необходимо нажать на кнопку «Удалить». Для отмены удаления нажать на кнопку «Отменить».

2.7 Списки

Раздел **Списки** позволяет настроить произвольное количество списочных представлений для отображения пользователям.

Настройка списков выполняется в визуальном дизайнера и позволяет мышью сформировать произвольный список, требуемый конечному пользователю.

По умолчанию платформа для каждого нового шаблона записи создает список «Все записи», который содержит следующие колонки, привязанные к одноименным системным атрибутам (более подробно о системных атрибутах описано в разделе «Настройка атрибутов» на стр. 22):

- ИД.
- Создан.
- Дата создания.
- В архиве.
- Последнее изменение.
- Последний изменивший.

Для просмотра списка перейдите в созданный шаблон и выберите раздел **Экземпляры**.

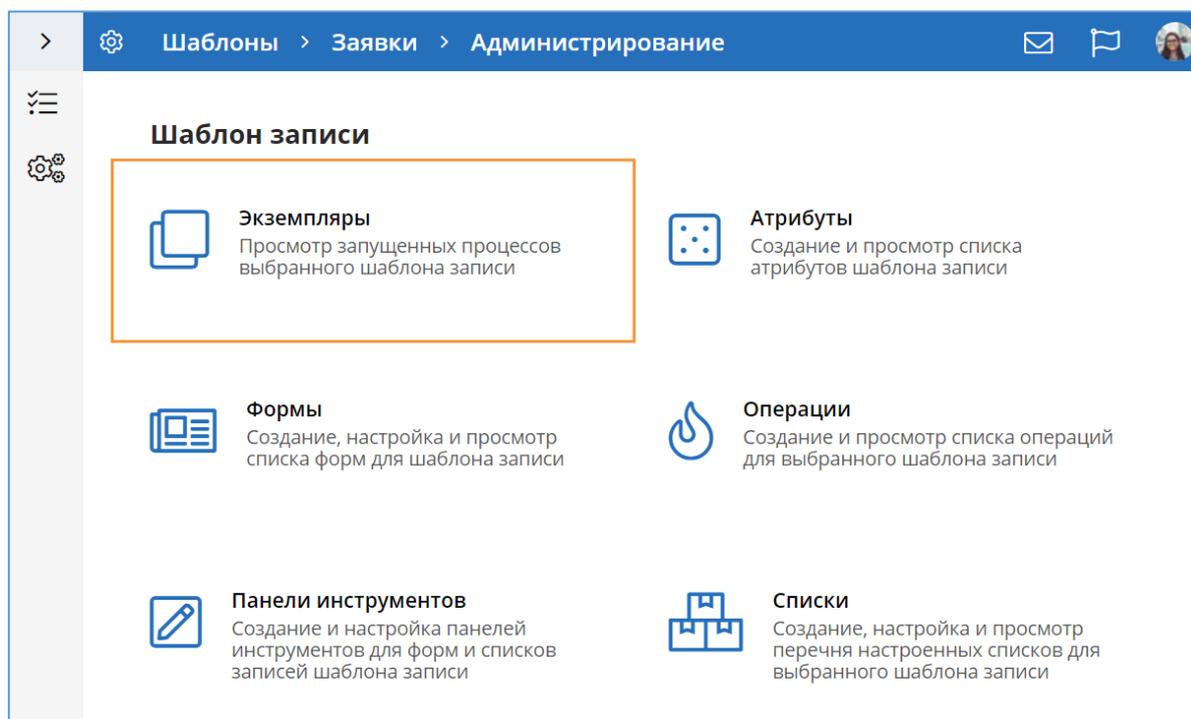


Рис. 146 Переход к просмотру списка записей шаблона

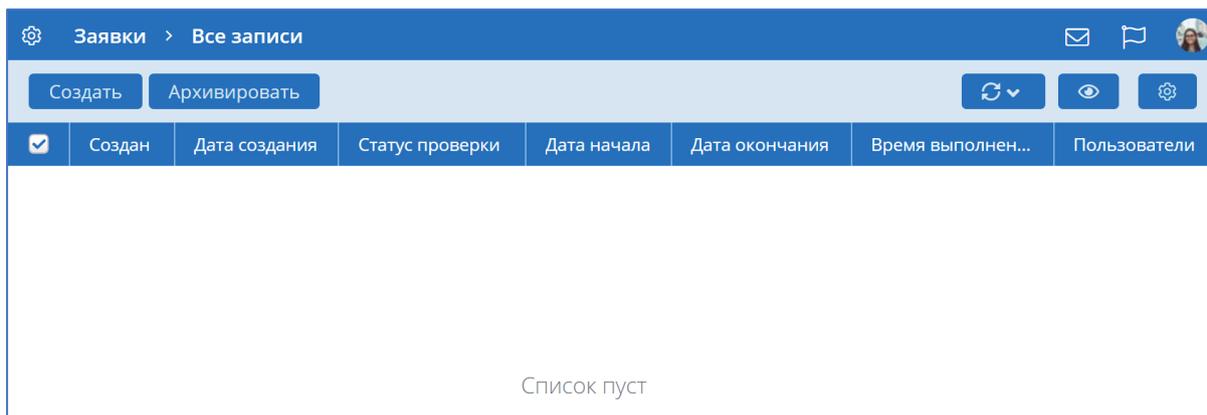


Рис. 147 Список нового шаблона записи

Шаблон записи может содержать несколько настроенных списков, которые могут быть доступны пользователю согласно настроенным правам доступа.

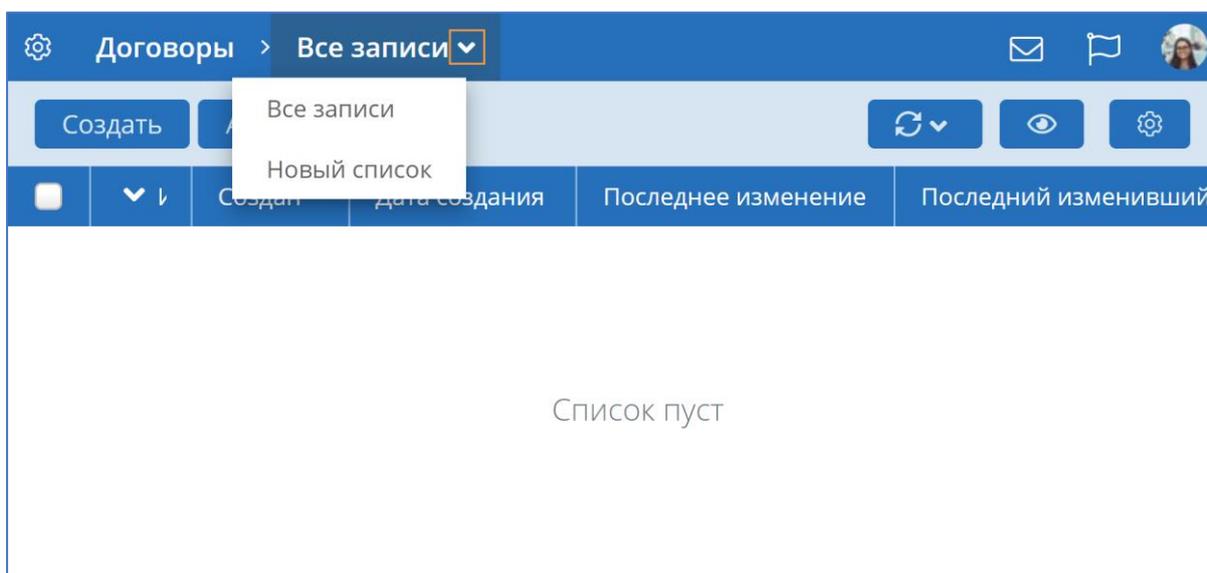


Рис. 148 Отображение нескольких настроенных списков записей для шаблона

2.7.1 Добавление списка

Для добавления нового списка необходимо выполнить следующие действия:

1. Войти в раздел **Списки**.

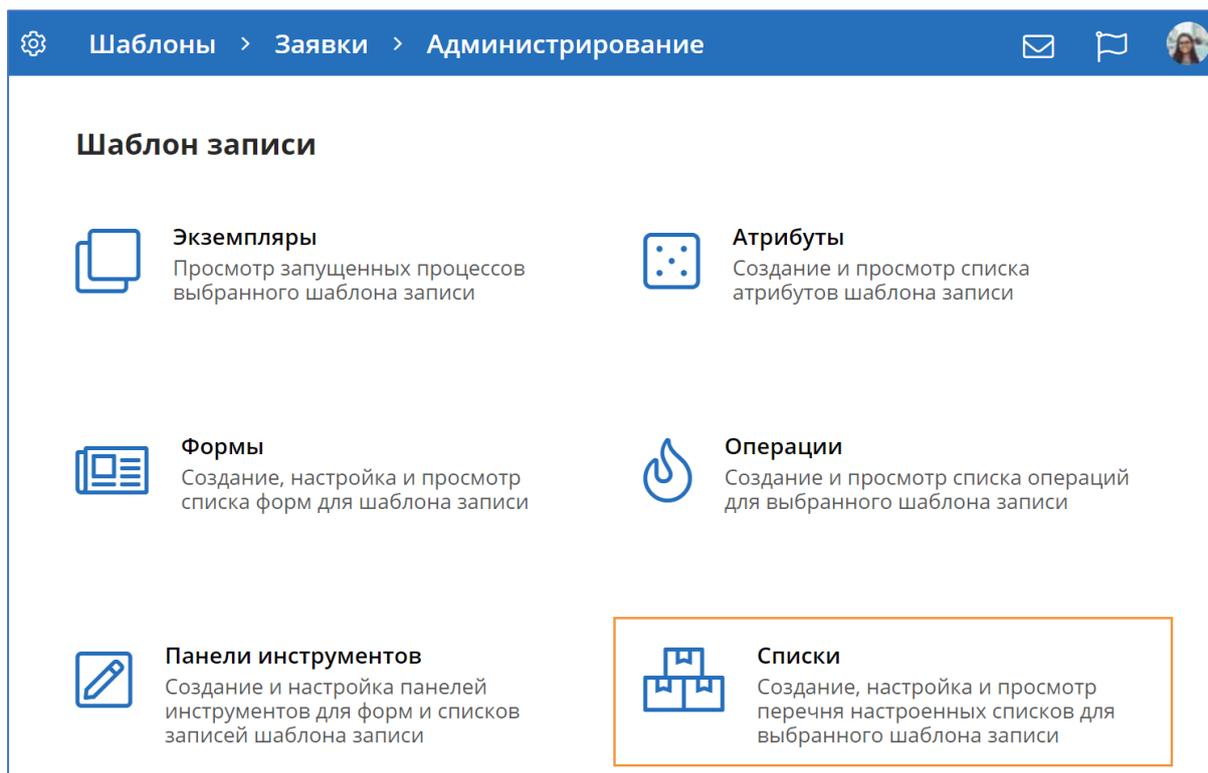


Рис. 149 Расположение раздела **Списки**

2. Для добавления нового списка нажать на кнопку «Добавить».

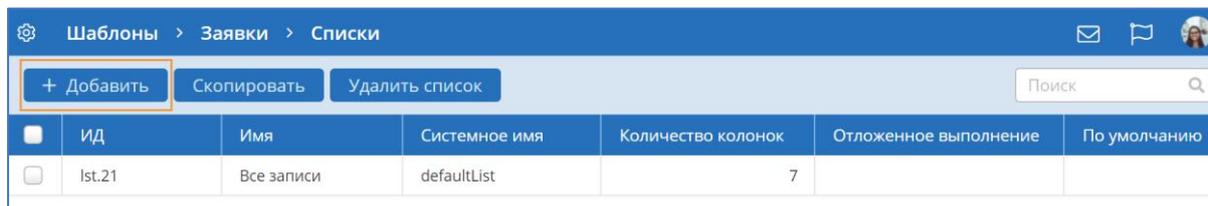


Рис. 150 Расположение кнопки «Добавить»

На пользовательском интерфейсе открывается конструктор списка. Конструктор списка разбит на 3 области:

- 1 - В левой части страницы отображается перечень доступных элементов для добавления в рабочую область.
- 2 - Центральная форма является Рабочей областью - основная визуальная область конструктора списка, где производится визуальное наполнение и формирование последовательности элементов в списке.
- 3 - В правой части отображается Панель настроек – набор характеристик выбранного элемента в рабочей области. Для выбора доступен сам список, а также любой элемент, помещаемый в рабочую область. По умолчанию, когда не выбрано ни одного элемента, отображаются характеристики списка.

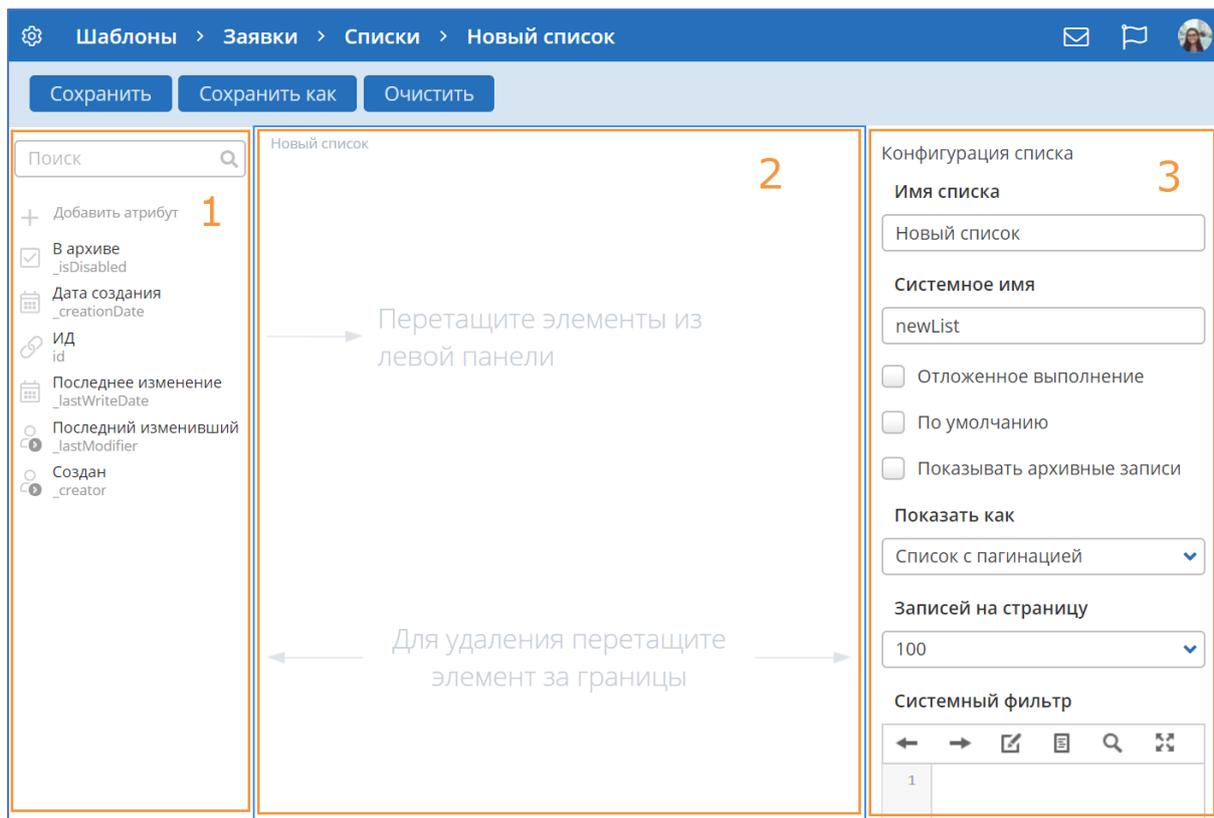


Рис. 151 Внешний вид конструктора списков

3. После настройки списка сохраните, нажав на кнопку **Сохранить**.

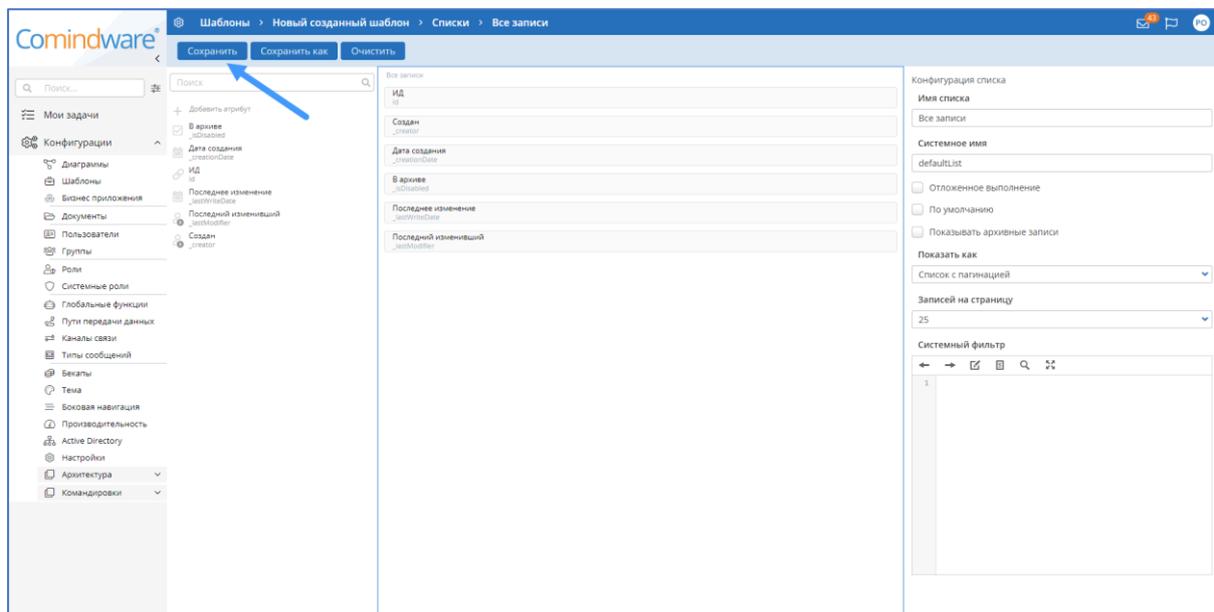


Рис. 152 Расположение кнопки «Сохранить»

4. Для сохранения панели как копии под другим именем нажмите на кнопку «Сохранить как».

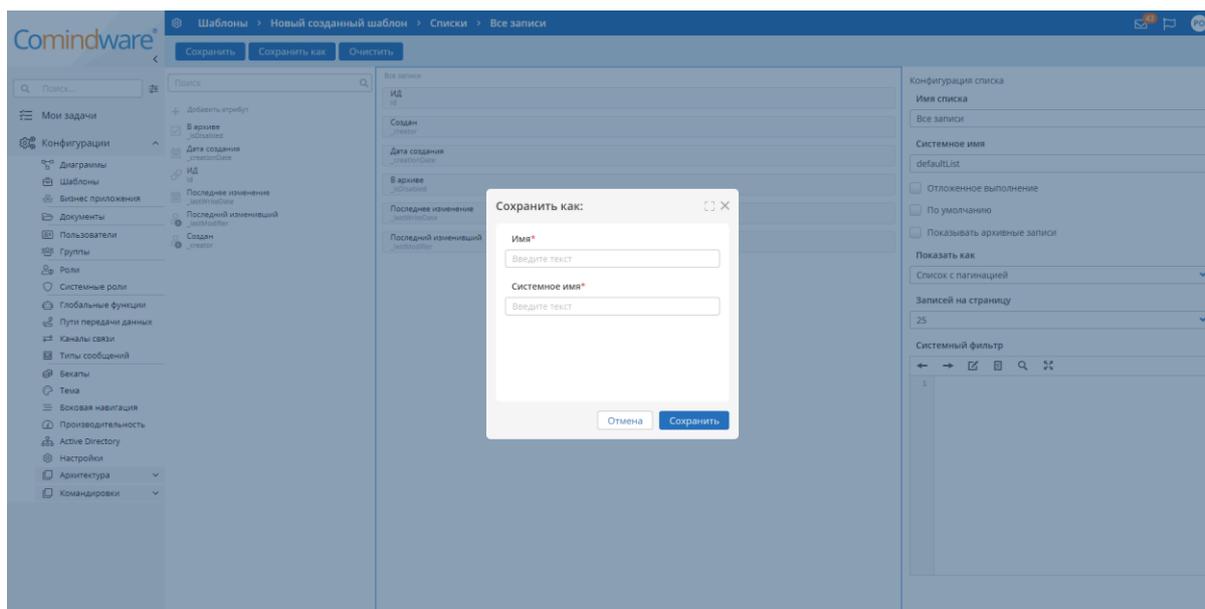


Рис. 153 Форма ввода реквизитов нового списка

В всплывающем окне необходимо заполнить наименование и системное имя списка.

- Для сохранения нового списка нажмите на кнопку **Сохранить**. Для отмены сохранения новой формы, необходимо нажать на кнопку **Отмена**.

2.7.2 Настройка списка в конструкторе списка

Настройка списка производится на конструкторе списка (рис. 119) путем перетаскивания на рабочую область конструктора нужных элементов – колонок списка.

Панель настроек списка

Для списков могут быть определены следующие характеристики:

- Имя списка – наименование списка, которое отображается в перечне списков и несет смысловую нагрузку цели создания списка.
- Системное имя – уникальное текстовое наименование списка в рамках шаблона записи²⁻². Может содержать только буквы латинского алфавита, цифры и знак «_».
- Отложенное выполнение - флаг, который создает пустой список. Позволяет предварительно настроить список. Загрузка данных происходит при нажатии на кнопку «Загрузить данные».

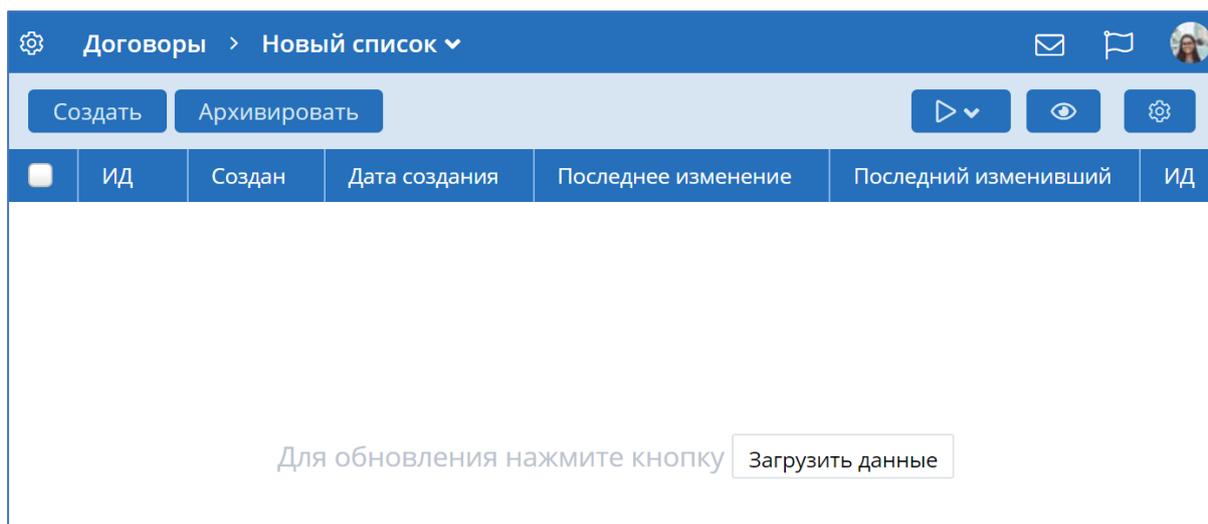


Рис. 154 Расположение кнопки «Загрузить данные» на пользовательской панели

- «По умолчанию» – флаг, который означает, что данный список будет отображаться по умолчанию при переходе пользователя к списку.
- «Показывать архивные записи» - флаг, который означает, что в перечне будут отображаться архивные записи, которые скрыты по умолчанию.
- «Показывать как» – определяет тип визуального отображения списка в пользовательском интерфейсе, платформа предлагает 2 типа:
 - 1) «Список с пагинацией» – записи отображаются в виде строк, а их атрибуты в виде столбцов единой таблицы. Этот тип используется по умолчанию.
 - 2) «Плитка» – записи отображаются в виде отдельных таблиц, где атрибуты отражены в виде строк.

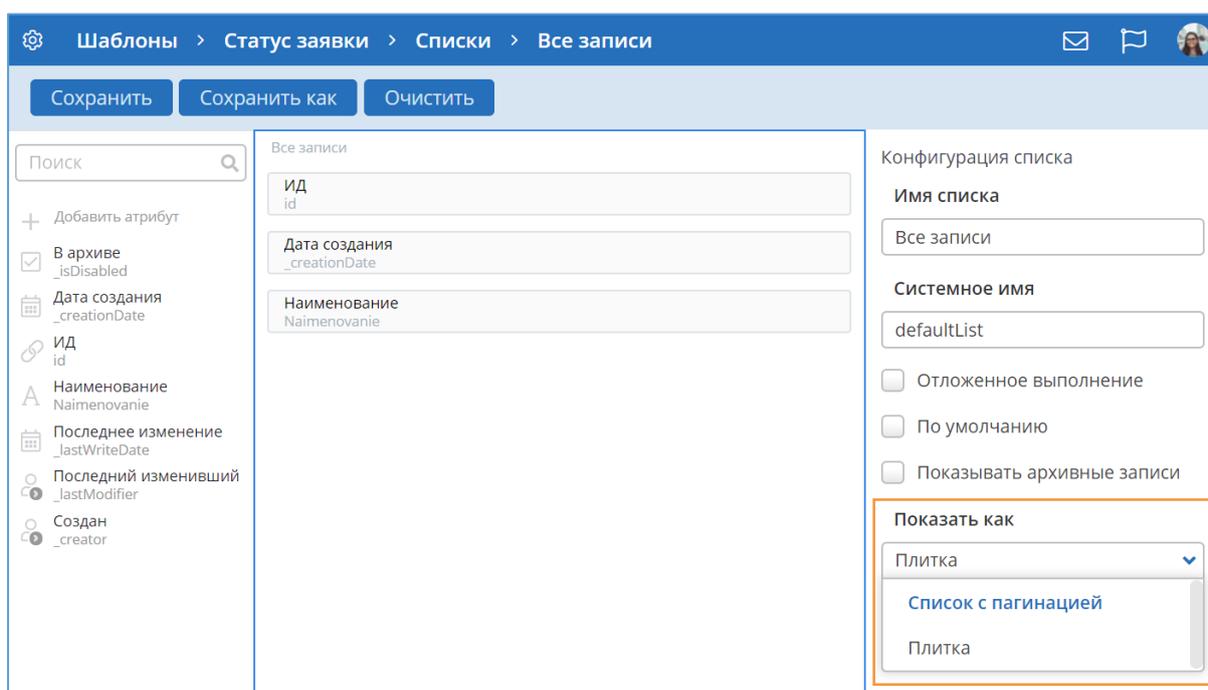


Рис. 155 Расположение настройки типа отображения списка

	Название	ИД	Дата создания
<input type="checkbox"/>	В ожидании согласования	159	06.02.2019 12:57
<input type="checkbox"/>	Согласована	156	06.02.2019 12:57
<input type="checkbox"/>	Отменена	153	06.02.2019 12:57
<input type="checkbox"/>	Новая	150	06.02.2019 12:57

Рис. 156 Внешний вид списка типа «Список с пагинацией»

Рис. 157 Внешний вид списка типа «Плитка»

- Записей на страницу – атрибут, позволяющий определить количество отображаемых записей на одной странице списка. По умолчанию присвоено значение - 25 записей.
- Системный фильтр - запрос, который позволяет ограничить перечень записей в списке. Фильтр должен возвращать перечень id записей. Выражение пишется либо на языке выражений платформы, либо на языке запросов RDF N3.

Панель элементов конструктора списков

Панель элементов конструктора содержит динамические элементы:

- Динамические элементы – это набор атрибутов, настроенных на текущем шаблоне записи, позволяющих вынести атрибуты в рабочую область для формирования списка.

Добавление элементов на панель инструментов производится путем «перетаскивания» элементов с панели элементов конструктора в рабочую область и определения индивидуальных характеристик визуальных элементов, используя Панель настроек списка.

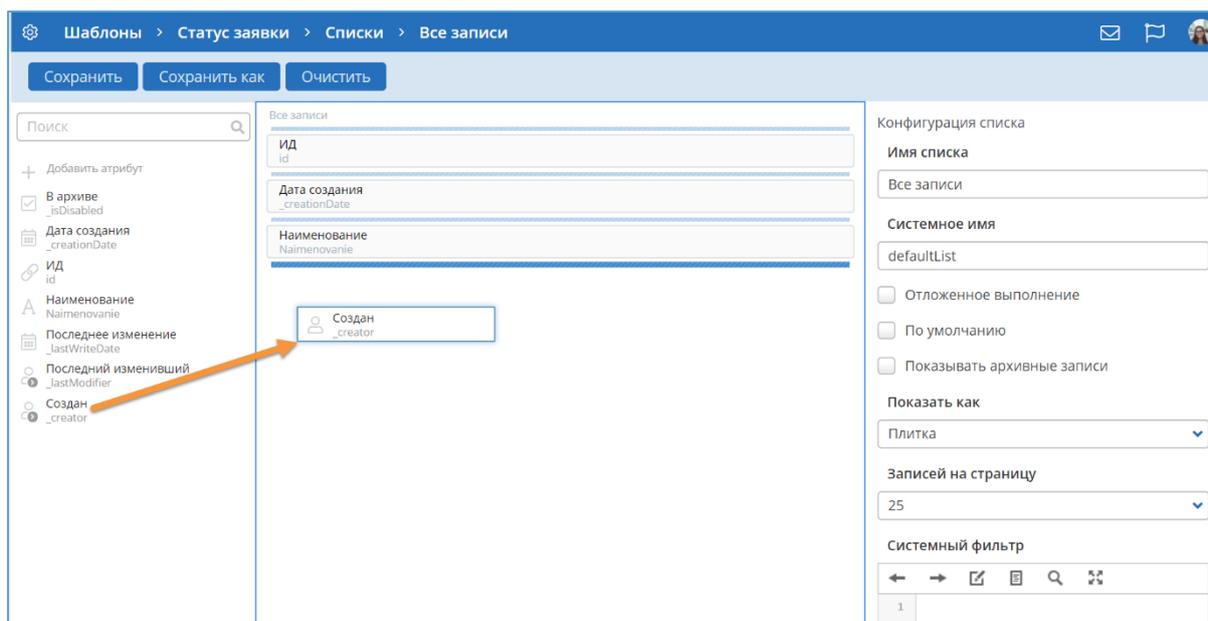


Рис. 158 Механизм добавления визуальных элементов в список

Чтобы перенести элемент на рабочую область, выберите элемент из списка на левой панели и, удерживая левую кнопку мыши, перенесите на рабочее поле. Доступные для переноса области подсвечиваются синими линиями.

Таким образом визуальный элемент будет размещен в нужном месте рабочей области, который затем аналогично в заданной последовательности отобразится в виде столбца списка.

Последовательность определяется сверху вниз, т.е. элемент, который расположен вверху рабочей области – отобразится первым в списке записей.

Для просмотра свойств колонок, необходимо выбрать колонку в рабочей области конструктора списка - в панели свойств появится перечень характеристик, которые доступны для редактирования.

Также имеется возможность посмотреть характеристики текущего списка, нажав левую кнопку мыши в любом свободном от добавленных элементов месте рабочей области.

Динамические визуальные элементы на панели элементов конструктора содержат строку поиска по наименованию и системному имени операции.

Атрибуты

Панель элементов содержит только перечень созданных атрибутов, которые содержит текущий шаблон записи.

При двойном щелчке мышью по элементу в панели элементов - открывается окно настройки атрибута.

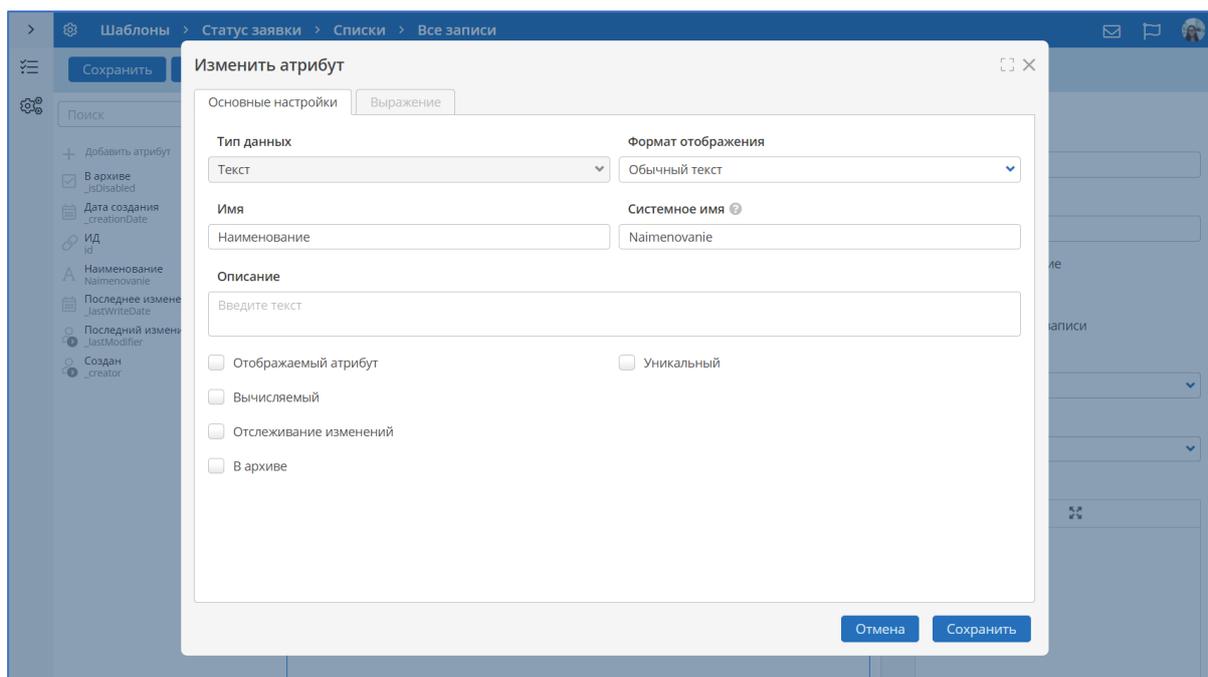


Рис. 159 Окно настройки атрибута

При добавлении атрибута в рабочую область конструктора списка на панели настройки выводится общая информация об атрибуте.

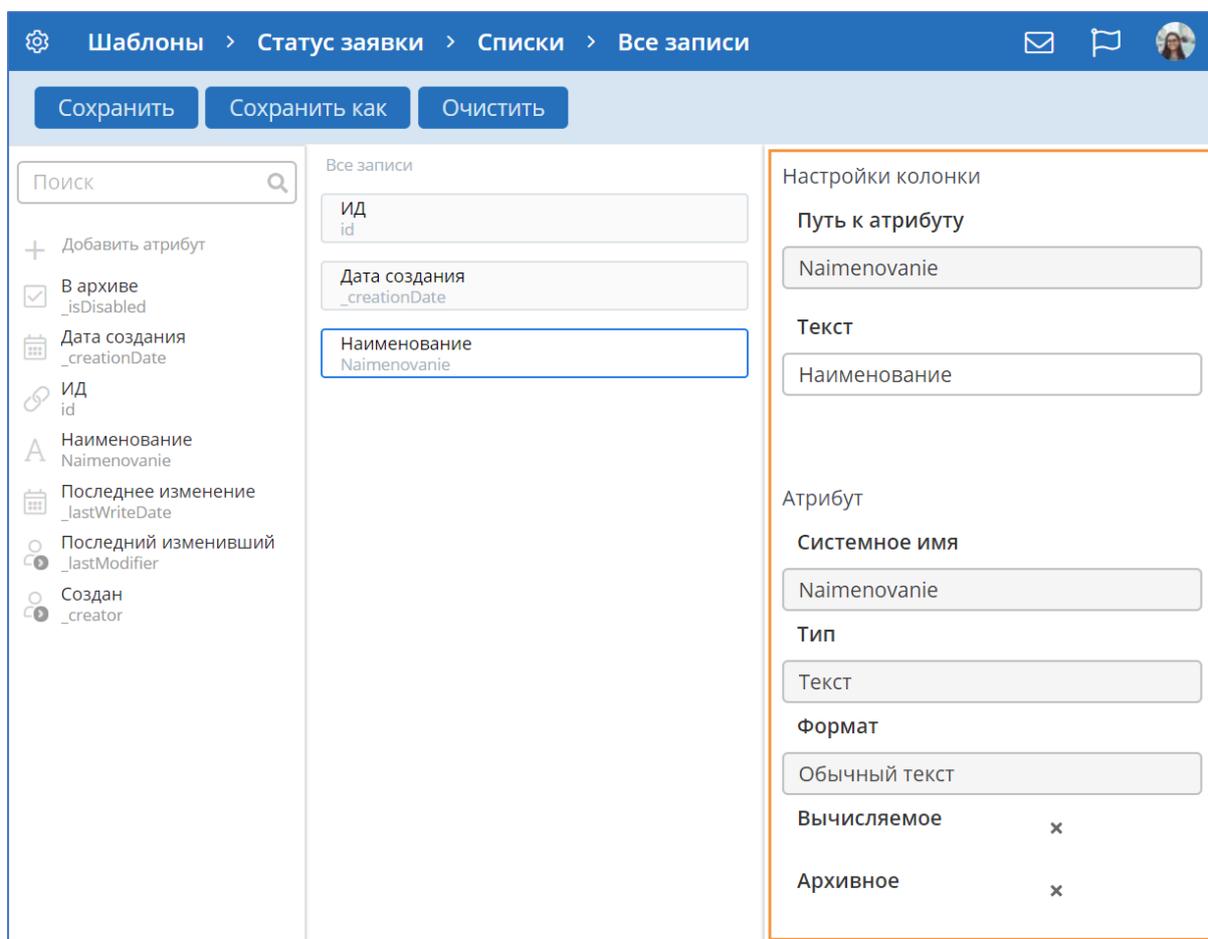


Рис. 160 Панель настроек с характеристиками выбранного элемента

Проектировщик имеет возможность переопределить следующие характеристики атрибута с помощью панели настройки:

- Текст – отображаемая на интерфейсе пользователя текстовая надпись, которая расположена в столбце соответствующего атрибута.

Для просмотра доступны следующие характеристики, выбранного атрибута:

- Путь к атрибуту – отображает полный путь к атрибуту и недоступен для редактирования. Подробное описание см. [выше в данном разделе](#).
- Системное имя – отображает системное имя атрибута и недоступно для редактирования. Подробное описание см. [выше в данном разделе](#).
- Тип – тип данных, выбранного атрибута. Подробное описание см. [выше в данном разделе](#).
- Формат - отображает тип данных атрибута и недоступен для редактирования. Подробное описание см. [выше в данном разделе](#).
- Вычисляемое – флаг, обозначающий является ли данный элемент вычисляемым атрибутом. Подробное описание см. [выше в данном разделе](#).
- Архивное - флаг, обозначающий является ли данный элемент архивным атрибутом. Подробное описание см. [выше в данном разделе](#).

Для быстрой очистки рабочей области на панели инструментов есть кнопка «Очистить», которая удаляет все элементы из настроенного списка.

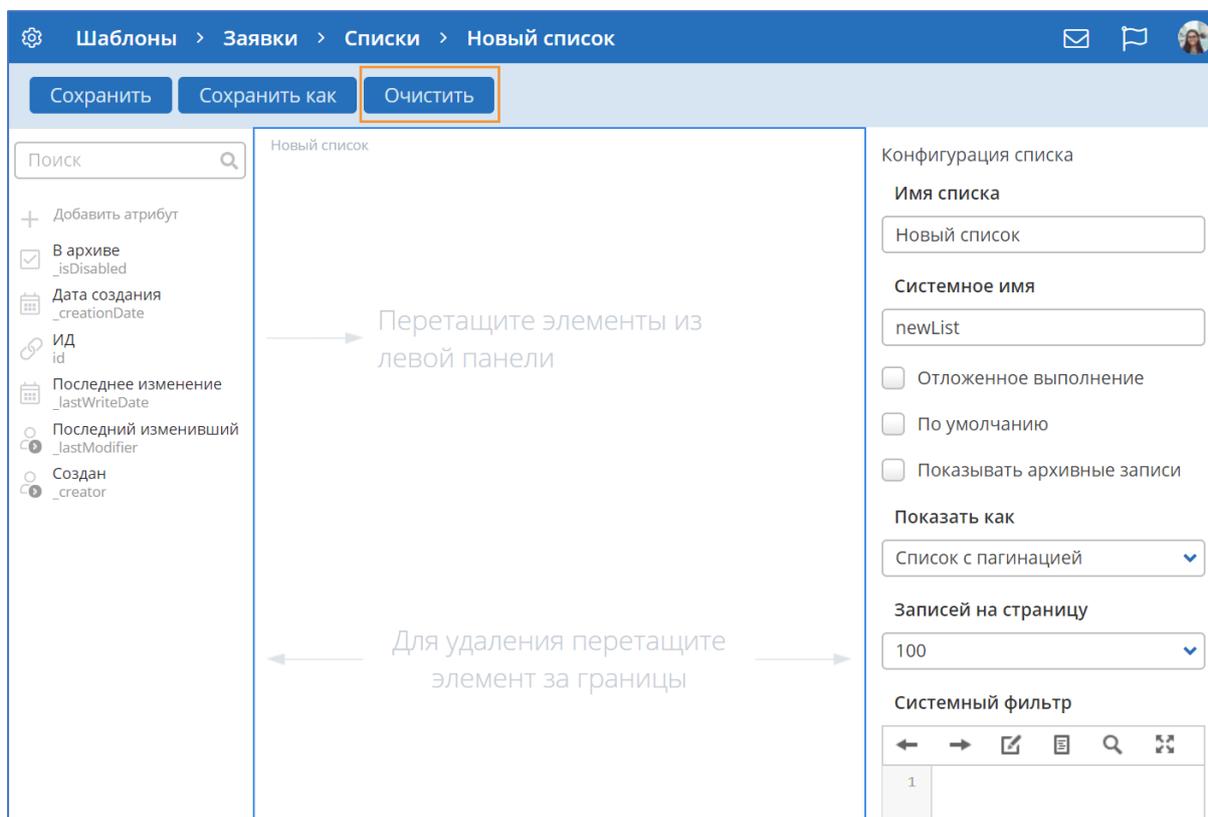


Рис. 161 Расположение кнопки «Очистить» в конструкторе списка

2.7.3 Удаление списка

Для удаления списка необходимо выполнить следующие действия:

1. Перейти в раздел **Списки** с перечнем списков и выбрать список для удаления.

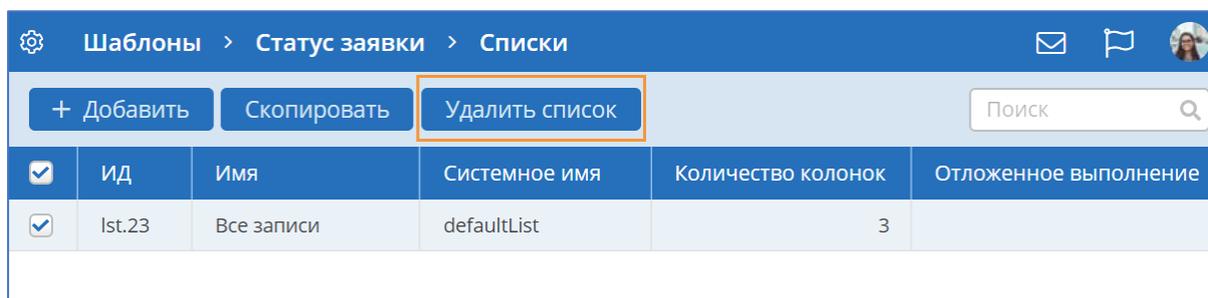


Рис. 162 Расположение кнопки «Удалить список»

В панели инструментов отображается кнопка «**Удалить список**».

2. Нажать на кнопку Удалить список.

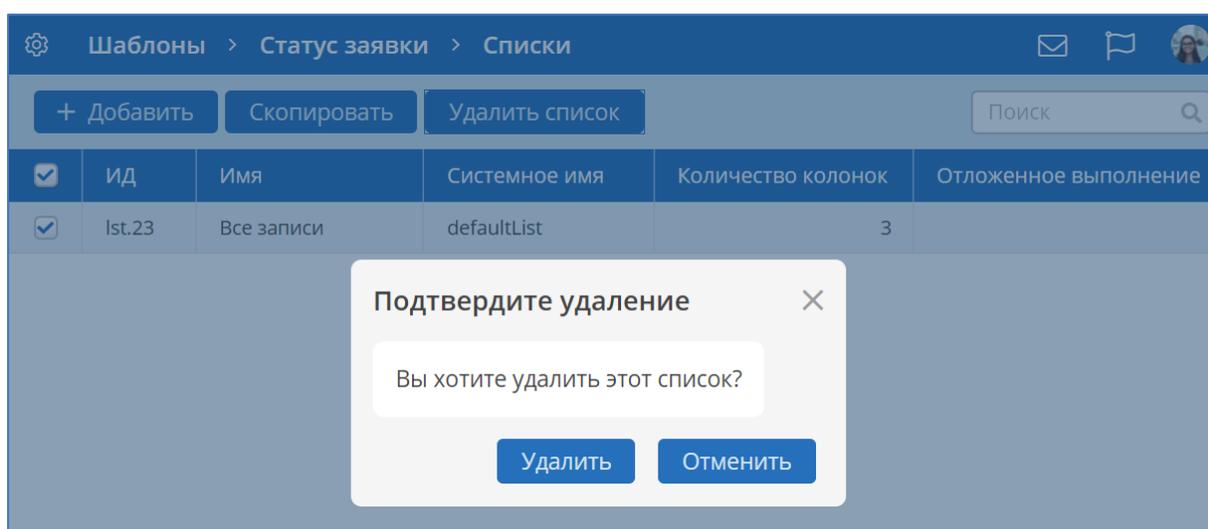


Рис. 163 Окно подтверждение удаления выбранных списков

Открывается диалоговое окно подтверждения удаления списков.

3. Для подтверждения удаления необходимо нажать на кнопку **Удалить**. Для отмены удаления нажать на кнопку **Отменить**.

2.7.4 Копирование списка

Для копирования списка необходимо выполнить следующие действия:

1. Перейти в раздел **Списки** с перечнем списков и выбрать список для копирования.

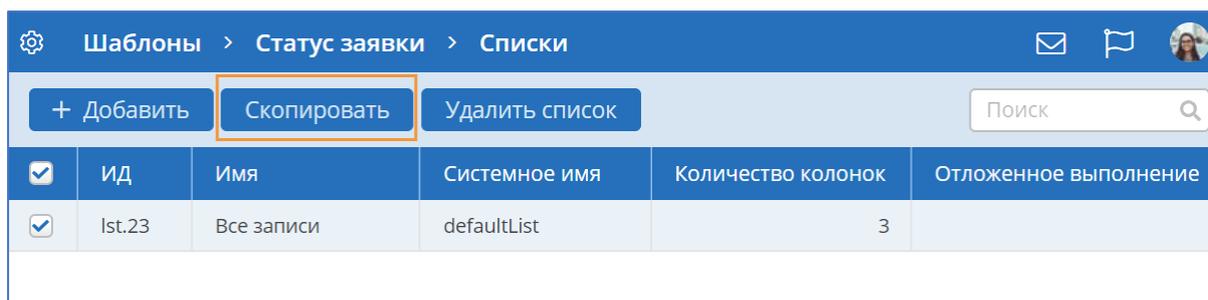


Рис. 164 Расположение кнопки «Скопировать»

В панели инструментов отображается кнопка «Скопировать».

2. Нажать на кнопку **Скопировать**.

Открывается новый конструктор списка с настройками, идентичными скопированному списку.

2.7.5 Работа со списком в пользовательском интерфейсе

Настройка списка может осуществляться в конструкторе списка, а также напрямую в пользовательском интерфейсе.

Для сортировки/фильтрации/группировки/агрегации списка необходимо навести курсор мыши на верхнюю строку списка, где отображается соответствующий атрибут, в рамках которого осуществляется сортировка. На экране отображается всплывающее окно с вкладками сортировки/фильтрации/группировки/агрегации.

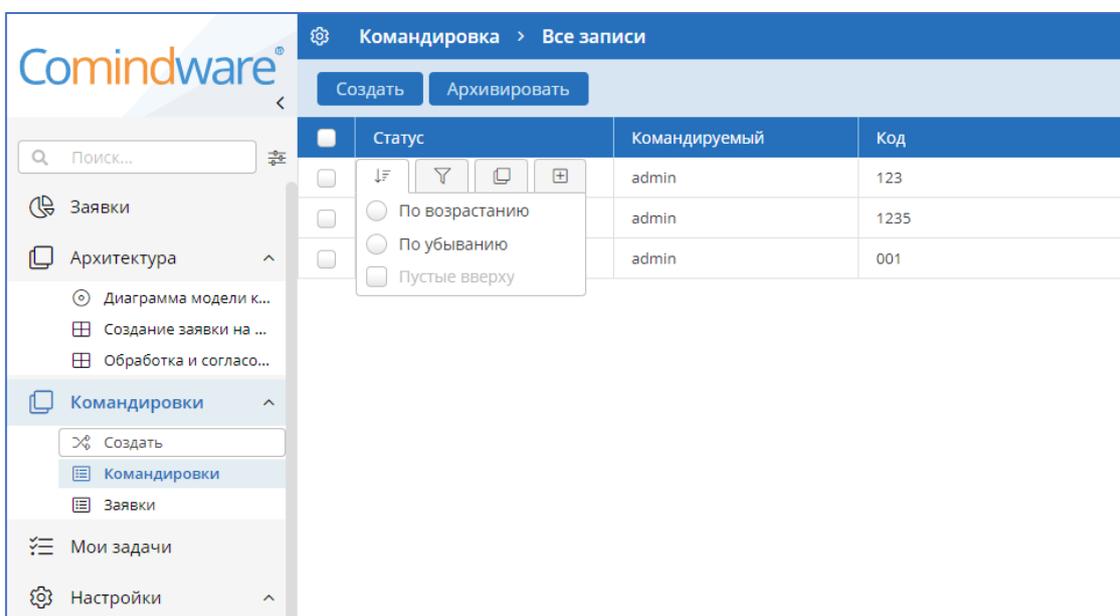


Рис. 165 Всплывающее окно настройки списка со вкладкой сортировки списка

1. Для сортировки списка в всплывающем окне настройки списка нужно выбрать первую вкладку, в которой отображаются логические атрибуты:

- По возрастанию – флаг, который позволяет сортировать список в порядке возрастания. При этом данная сортировка возможна для числовых, текстовых (в алфавитном порядке), а также временных атрибутов.

- По убыванию - флаг, который позволяет сортировать список в порядке убывания. При этом данная сортировка возможна для числовых, текстовых (в алфавитном порядке), а также временных атрибутов.
 - Пустые вверху – флаг, который позволяет при сортировке значений по возрастанию или убыванию вывести записи с пустыми значениями данного атрибута в первую очередь.
2. Для фильтрации значений списка в всплывающем окне настройки списка нужно выбрать вторую вкладку, в которой отображается строка выбора значений для установки фильтра. А также кнопки для удобства настройки фильтра:
- Кнопка «Добавление фильтра» - если необходимо выбрать множество значений, то можно добавить несколько строк фильтрации, для этого во вкладке есть кнопка добавления фильтра
 - Кнопка «Сохранение» - для применения фильтра к списку нужно нажать кнопку «Сохранение».

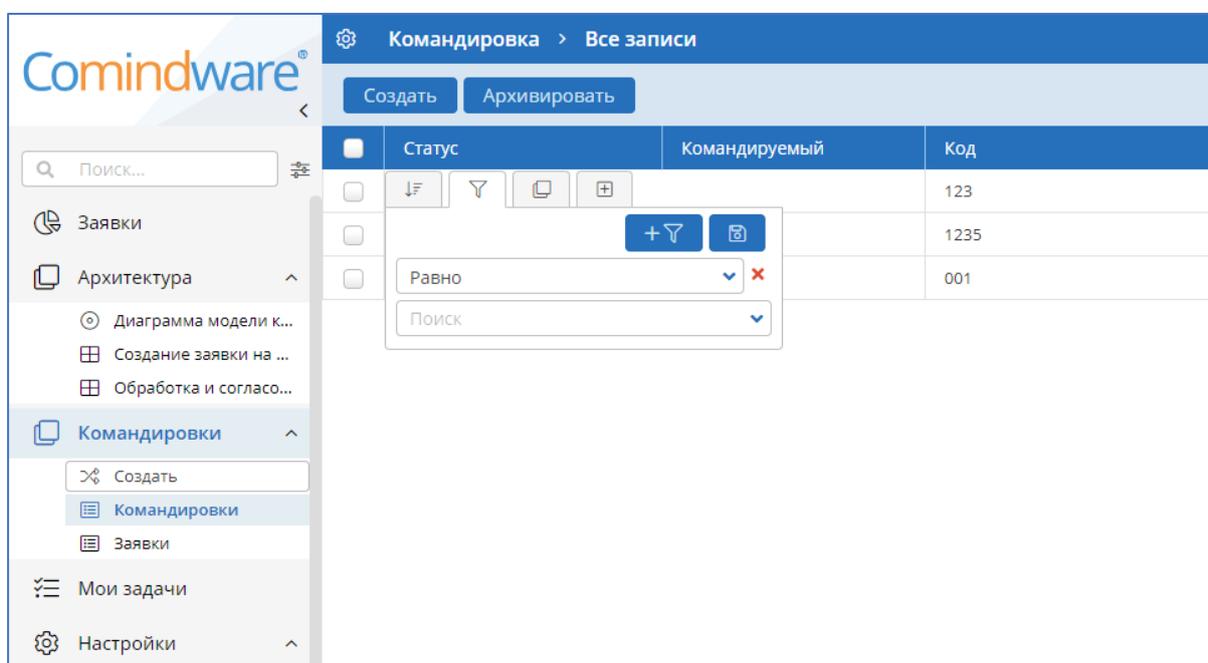


Рис. 166 Всплывающее окно настройки списка со вкладкой фильтрации списка

3. Для группировки значений списка в всплывающем окне настройки списка нужно выбрать третью вкладку и указать флаг «Группировать».

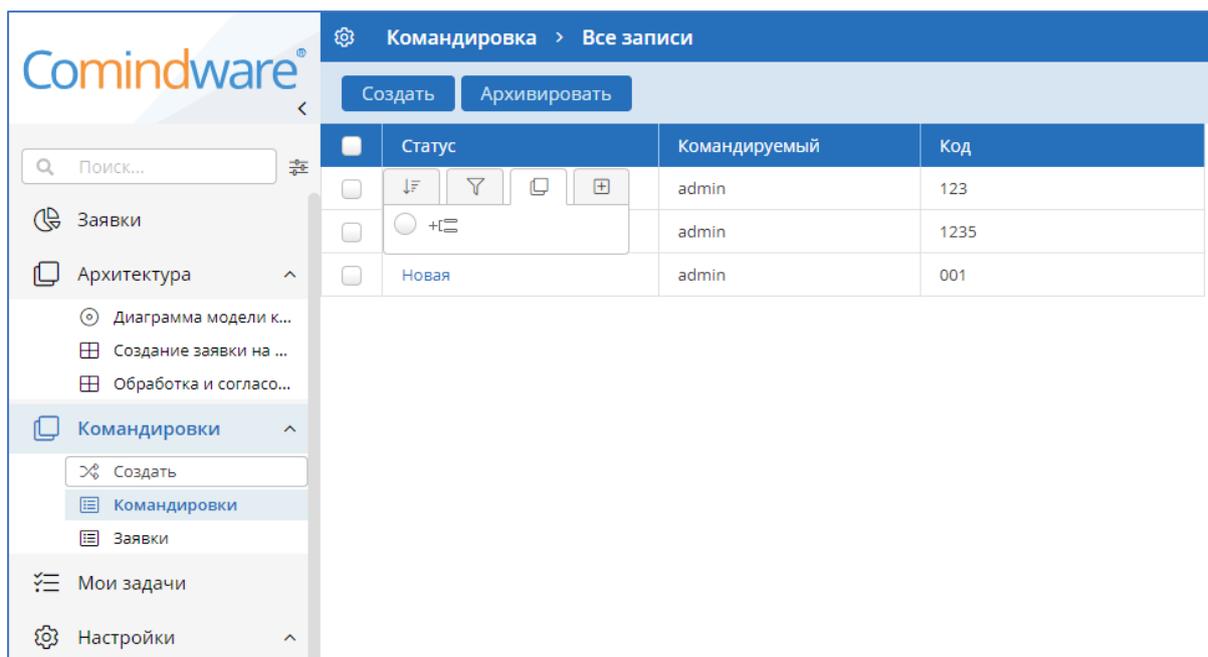


Рис. 167 Всплывающее окно настройки списка со вкладкой группировки списка

После установки флага группировки строки автоматически сгруппируются в соответствии со значениями данного столбца, одинаковые значения будут располагаться в одной группе, которая обозначается отдельной строкой в списке.

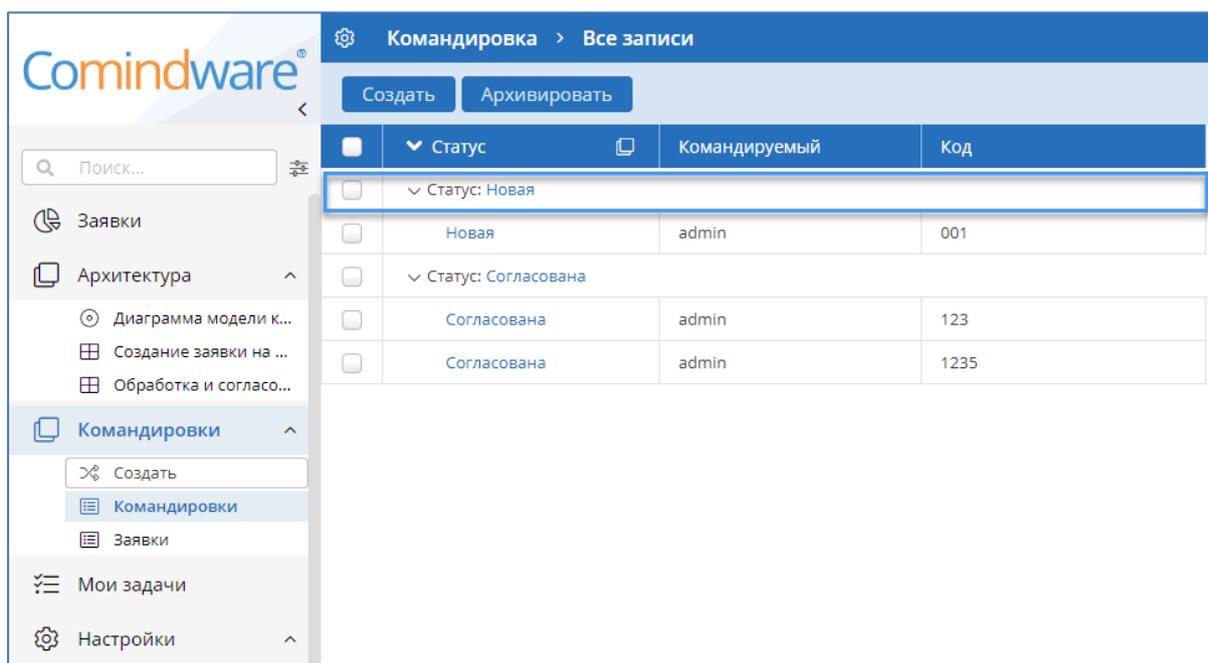


Рис. 168 Отдельная строка группировки списка.

После группировки по одной колонке станет доступно сделать подгруппу по другой колонке. Для этого нужно выбрать флаг «Установить как подгруппу» в третьей вкладке всплывающей настройки списка.

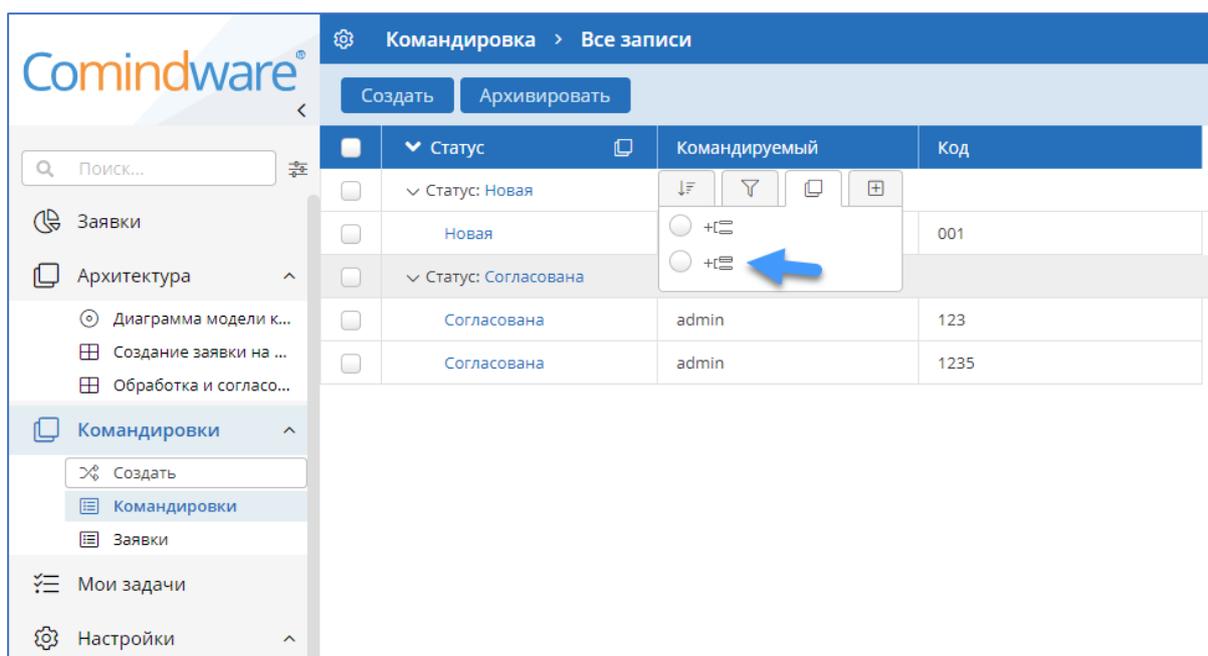


Рис. 169 Флаг «Группировать как подгруппу»

После этого список будет сгруппирован по двум колонкам.

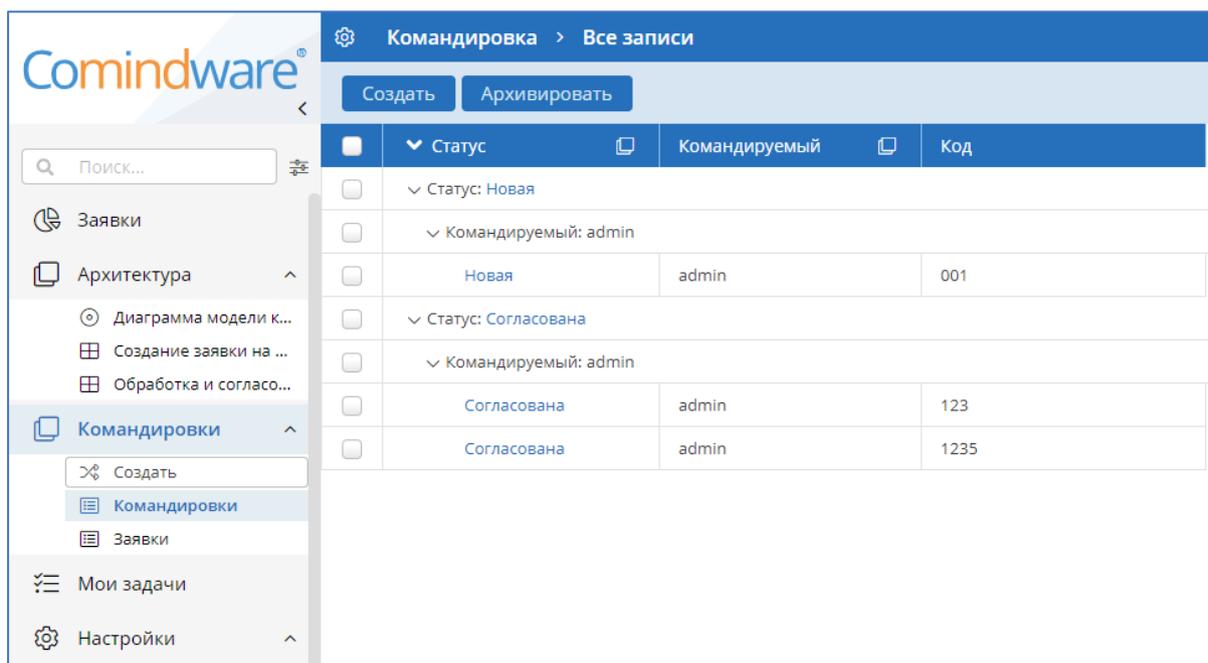


Рис. 170 Список с настроенной группировкой по двум колонкам

- Для агрегации значений списка в всплывающем окне настройки списка нужно выбрать четвертую вкладку и указать флаг «Количество». Функция агрегации выполняется после группировки полей списка и отображает количество записей в каждой группе.

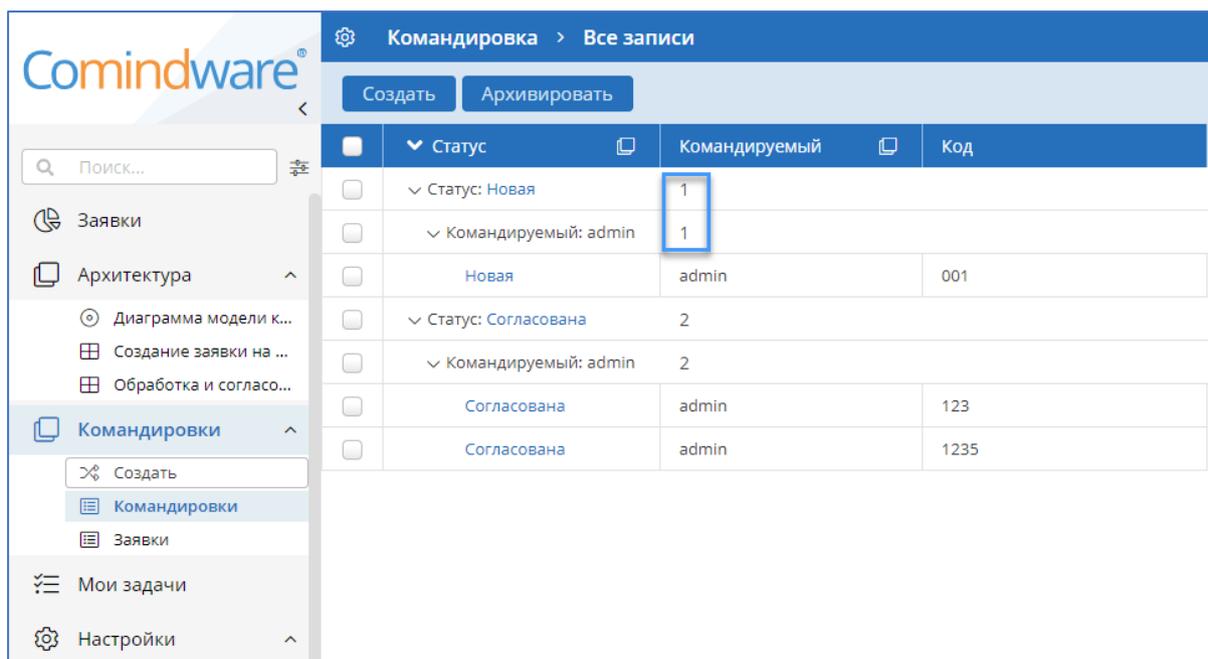


Рис. 171 Агрегация сгруппированных значений списка.

Для удаления настроек списка справа в строке наименования столбца нужно нажать на значок примененных настроек.

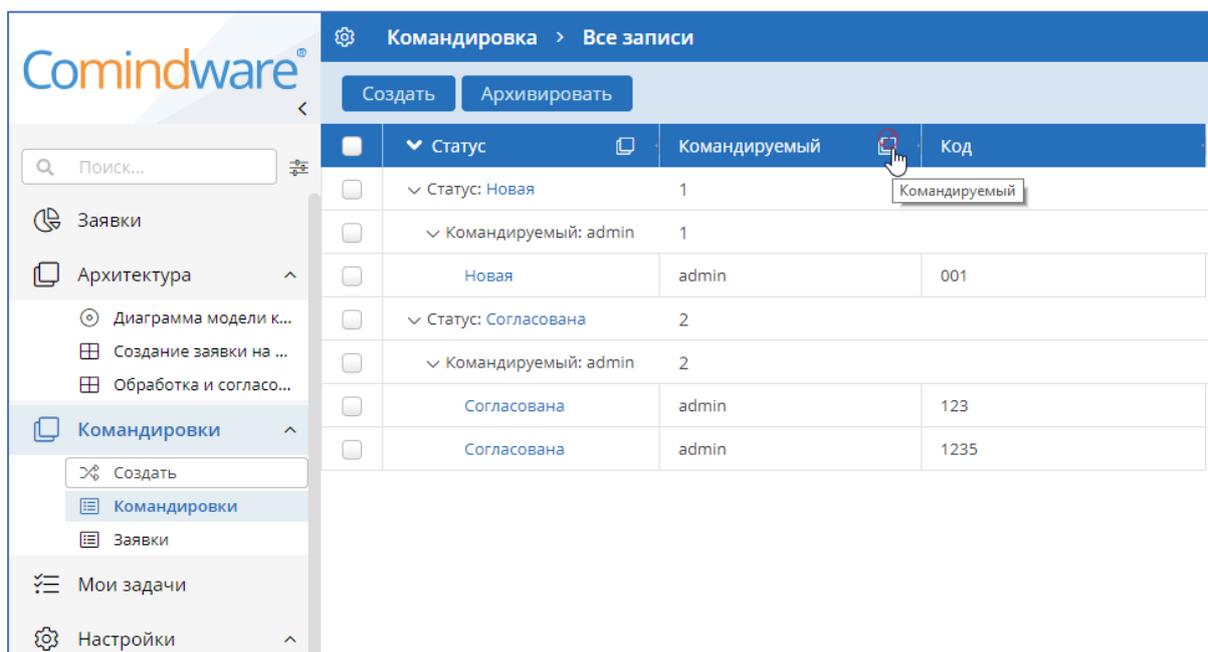


Рис. 172 Значок отмены применённых настроек списка.

В панели инструментов пользовательского интерфейса справа есть функциональные кнопки для настройки списков «Отображаемые столбцы» и «Конфигурация»

Командируемый	Код
admin	123
admin	1235
admin	001

Рис. 173 Расположение кнопок «Отображаемые столбцы» и «Конфигурация» на пользовательской панели инструментов.

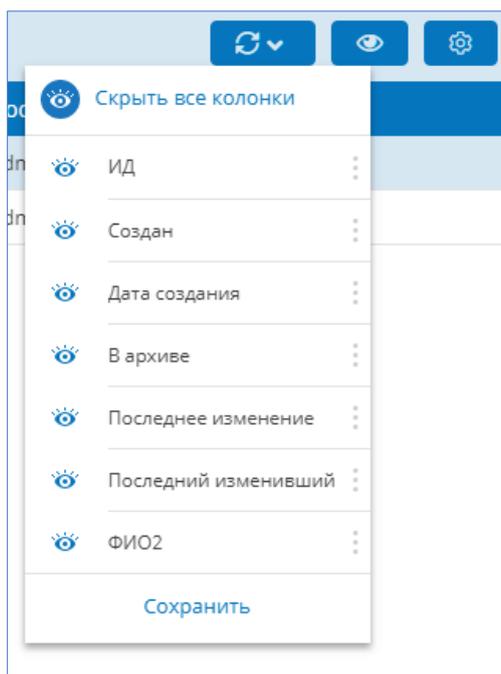


Рис. 174 Раскрывающийся список кнопки «Отображаемые столбцы»

- Отображаемые столбцы – кнопка, которая позволяет настраивать видимость столбца. По умолчанию все атрибуты списка, настроенные на конструкторе списка^{2.7.2}, отображаются. Чтобы скрыть какой-либо столбец, нужно нажать на кнопку «Отображаемые столбцы» и из раскрывающегося списка атрибутов кликнуть один раз по синему значку слева от атрибута, после чего он изменит цвет на серый. Для применения настройки внизу раскрывающегося списка необходимо нажать

кнопку «Сохранить». Отмеченный атрибут больше не отображается в списке записей. Для отображения скрытого атрибута нужно проделать аналогичные действия и вернуть значку напротив соответствующего атрибута синий цвет. Вверху раскрывающегося списка есть строка, позволяющая скрыть/показать сразу все атрибуты списка.

- Конфигурация - кнопка, которая позволяет сохранить настройки, произведенные в пользовательском интерфейсе для данного списка записей. Конфигурация также позволяет сбросить произведенные настройки, при необходимости.

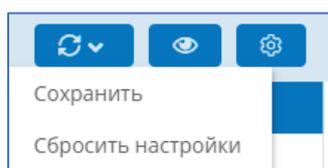


Рис. 175 Функции кнопки «Конфигурация»

2.8 Права доступа

Раздел **Права доступа** позволяет настроить ограничения прав доступа к шаблону записи для ролей, политика безопасности которых предполагает доступ к данному шаблону.

Настройка прав доступа выполняется в списке ролей и позволяет мышью сформировать уровень доступа к шаблону для каждой роли.

2.8.1 Добавление роли

Добавление ролей в список настройки прав доступа подробно описано в документе «Ролевая модель».

2.8.2 Настройка прав доступа в пользовательском интерфейсе

Для перехода к настройке прав доступа перейдите в шаблон, для которого требуется настроить доступ. Для этого кликните по иконке пользователя в правом верхнем углу и перейдите в раздел **Шаблоны**.

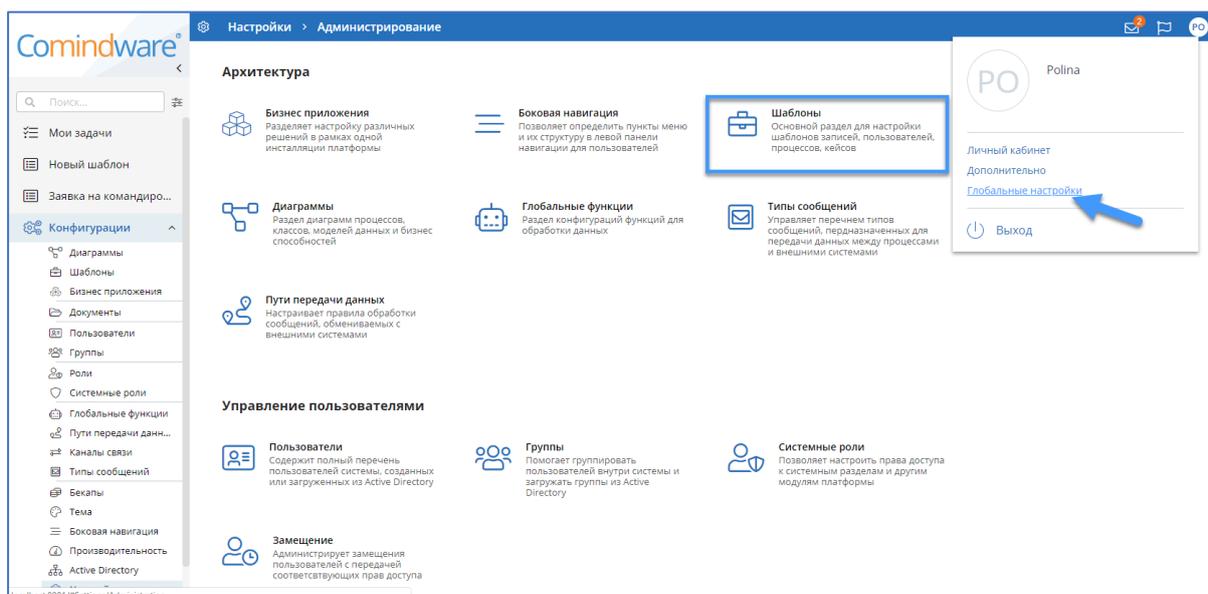


Рис. 176 Переход к списку шаблонов

После этого выберите шаблон из списка и перейдите в раздел **Права доступа**.

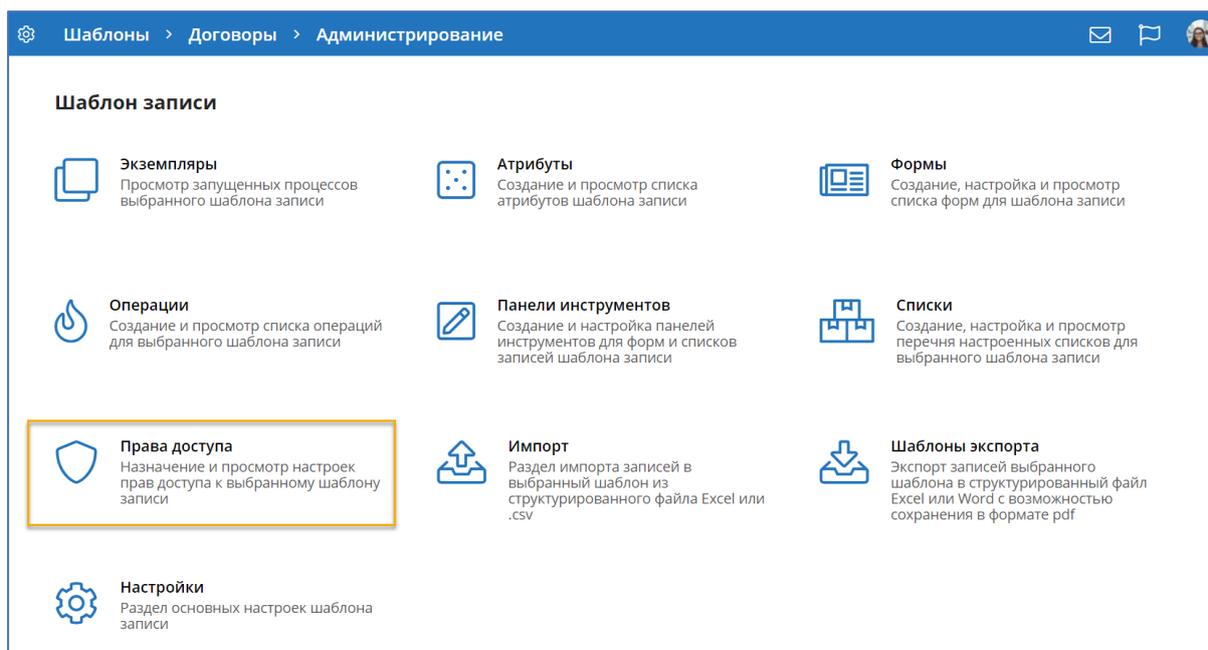


Рис. 177 Переход в раздел **Права доступа**

Роль	Пользователи	Имя	Тип ресурса	Полный доступ	Создание	Чтение	Редактирование	Удаление	Выполнение
Руководитель	1 человек 0 групп	Договоры	Шаблон записи	<input checked="" type="checkbox"/>					
Директор	1 человек 0 групп	Договоры	Шаблон записи	<input checked="" type="checkbox"/>					

Рис. 178 Список ролей раздела «Права доступа»

По умолчанию, платформа для каждого шаблона записи создает список «Ролей», которые имеют доступ к данному шаблону. Список «Ролей» содержит следующие колонки, привязанные к соответствующим ролям.

- роль – ссылочное поле, которые ссылается на соответствующую роль, для которой настраиваются права доступа и политика безопасности которой предполагает доступ к данному шаблону.
- пользователи – ссылочное поле, которое отражает всех пользователей, относящихся к данной системной роли.
- имя – текстовое поле, которое отображает наименование шаблона, к которому настраиваются права доступа.
- тип ресурса - текстовое поле, которое отображает тип шаблона, к которому настраиваются права доступа (например, шаблон записи).

Для настройки прав доступа ролей, отображаемых в списке, напротив каждой отображаются соответствующие чек-боксы:

- полный доступ – чек-брок, который определяет возможность пользователя, с соответствующей ролью выполнять функции администратора.
- создание – чек-брок, который определяет возможность пользователя, с соответствующей ролью создавать записи в шаблоне.

-
- Чтение – чек-бокс, который определяет возможность пользователя, с соответствующей ролью создавать записи в шаблоне.
 - Редактирование - чек-бокс, который определяет возможность пользователя, с соответствующей ролью редактировать записи в шаблоне.
 - Удаление - чек-бокс, который определяет возможность пользователя, с соответствующей ролью удалять записи в шаблоне.
 - Выполнение - чек-бокс, который определяет возможность пользователя, с соответствующей ролью выполнять операции над записями в шаблоне.
 - Выражение – возвращает набор пользователей, для которых будет определен доступ к ресурсу. Записывается на языке выражений или на языке N3.
 - Условие - возвращает «true» или «false» для пользователей с доступом к ресурсу.

2.9 Шаблоны экспорта

2.9.1 Простой шаблон

Создание шаблона экспорта

Шаблон документа в формате .xls

1. Создайте новую книгу excel.
2. Создайте список с данными по типу ниже, где:
 - {#Price} – заполняется значение атрибута Price, отображается в виде ссылки.
 - {Manager.phone} – заполняется значение атрибута phone из связанного шаблона записи.
 - {foreach:Passengers}{#Name} {Weight}{end:Passengers} – заполняются значения атрибутов name, weight по коллекции, то есть создается количество строк соответствующее количеству строк в коллекции.

	A	B	C	D	E	F
1	Color	{Color}				
2	IsNew	{IsNew}				
3	ReceivedAt	{ReceivedAt}				
4	ServiceDuration	{ServiceDuration}				
5	Price	{#Price}				
6	ProductionYear	{ProductionYear}				
7	PriceInEuro	{PriceInEuro}				
8	Manager	{Manager}				
9	Manager Department	{Manager.phone}				
10	Instruction	{Instruction}				
11	Passengers	{Passengers}				
12	Passengers weight	{Passengers.Weight}				
13	Two properties in the same cell	{ProductionYear}; {PriceInEuro}				
14	Unknown property	{Unknown}				
15						
16						
17	Collection1:					
18	Color={Color}; {foreach:Passengers}{#Name} {Weight}{end:Passengers} Price={Price}					
19						
20	Collection2:					
21	Color={Color}; {foreach:Passengers}{Name} - {#Weight}{end:Passengers} Price={Price}					
22						
23	Collection3:					
24	Color={#Color}; {foreach:Passengers} {Name}	{Weight}	{end:Pass	IsNew={IsNew}		
25						
26						

Рис. 179 Пример документа формата .xls

Шаблон документа в формате .doc

3. Создайте новую книгу word.
4. Создайте шаблон с данными по типу ниже:

Color	{Color}
IsNew	{IsNew}
ReceivedAt	{ReceivedAt}
ServiceDuration	{ServiceDuration}
Price	{#Price}
ProductionYear	{ProductionYear}
PriceInEuro	{PriceInEuro}
Manager	{Manager}
Manager Department	{Manager_phone}
Instruction	{Instruction}
Passengers	{Passengers}
Passengers weight	{Passengers.Weight}
Two properties in the same cell	{ProductionYear}: {PriceInEuro}
Unknown property	{Unknown}
Collection1:	
{Color}: {foreach:Passengers}{#Name}	{Weight}{end:Passengers} {Price}
Collection2:	
{Color}: {foreach:Passengers}{Name} - {#Weight}{end:Passengers} {Price}	
Collections3:	
{#Color}: {foreach:Passengers}	{Name} {Weight} {end:Passengers} {IsNew} {Price}

Рис. 180 Пример документа формата .doc

Загрузка шаблона экспорта

Для создания шаблона экспорта перейдите к шаблону, для которого нужно настроить шаблон экспорта. Для этого кликните по иконке пользователя в правом верхнем углу и перейдите в раздел **Шаблоны**.

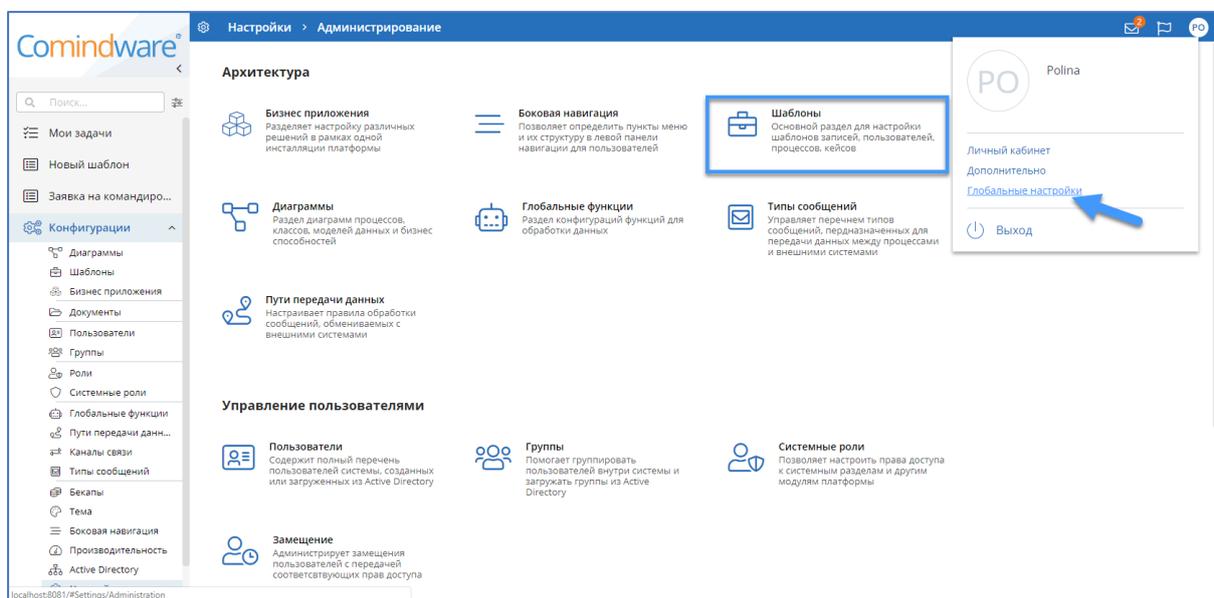


Рис. 181 Переход к списку шаблонов

Затем выберите нужный шаблон и перейдите в раздел **Шаблоны экспорта**.

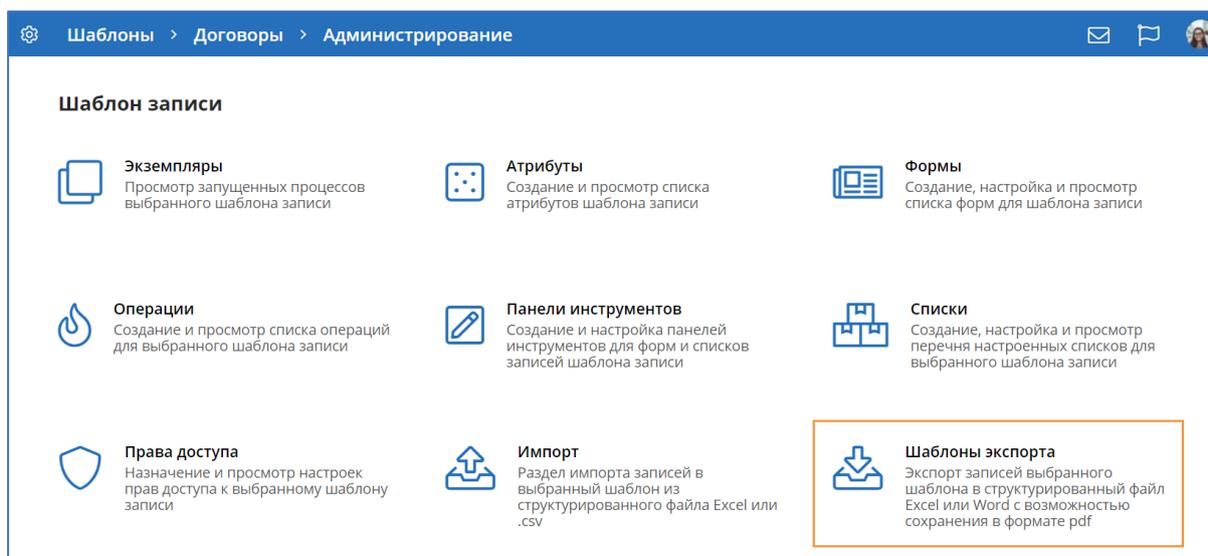


Рис. 182 Переход в раздел **Шаблоны экспорта**

Для добавления шаблона экспорта нажмите «+Добавить».

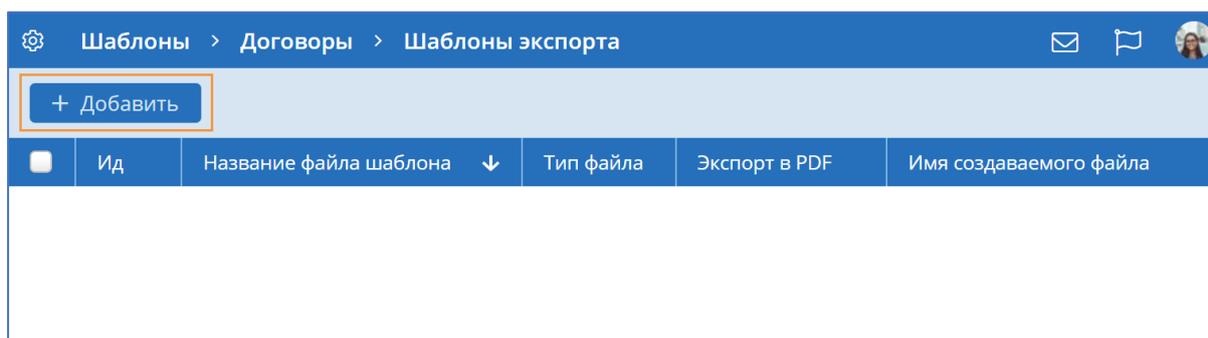


Рис. 183 Кнопка «Добавить» на списке шаблонов экспорта

После этого отобразится окно для загрузки шаблона.

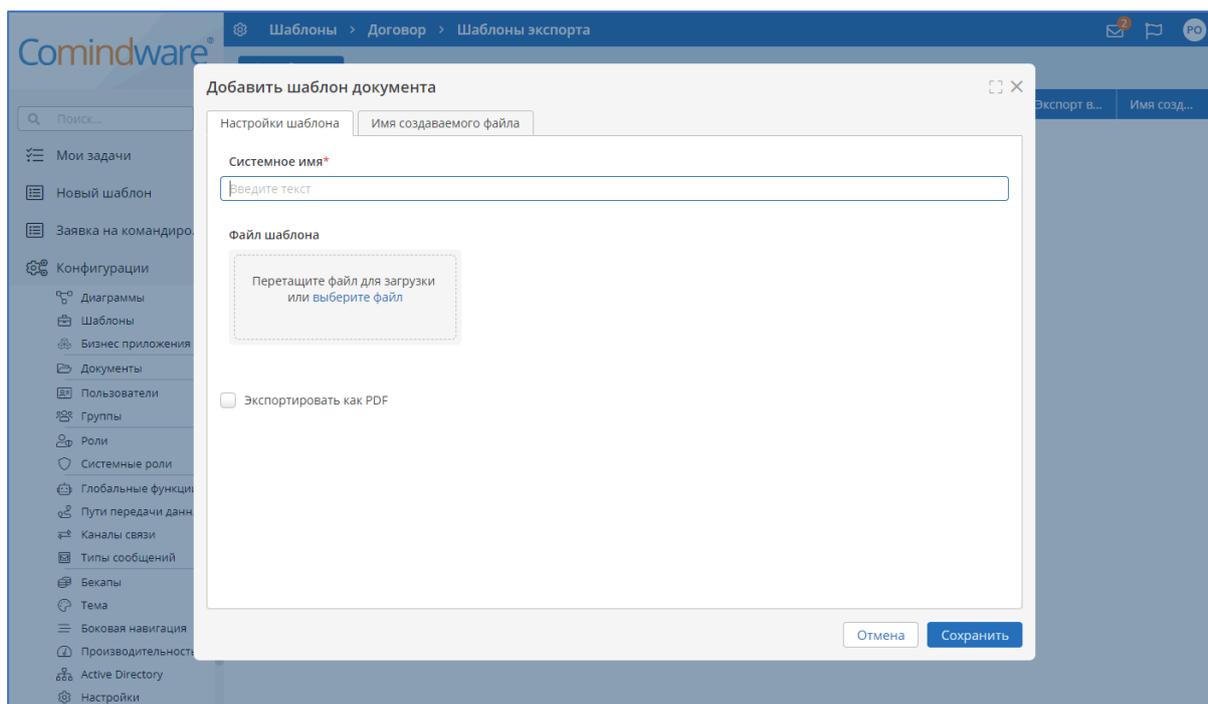


Рис. 184 Окно для загрузки шаблона экспорта

1. В окне настройки шаблона документа прикрепите файл с созданным ранее шаблоном в формате .doc или .xls

После создания шаблона документа автоматически создается операция с одноименным названием шаблона. Для последующей настройки перейдите в раздел «Операции»⁸⁷ тип «Экспорт записи». При отмеченном флаге «Экспортировать как PDF» документ преобразуется в формат PDF.

2. Задайте имя шаблона: напрямую, с помощью языка выражений платформы²²³ или скрипта C#.

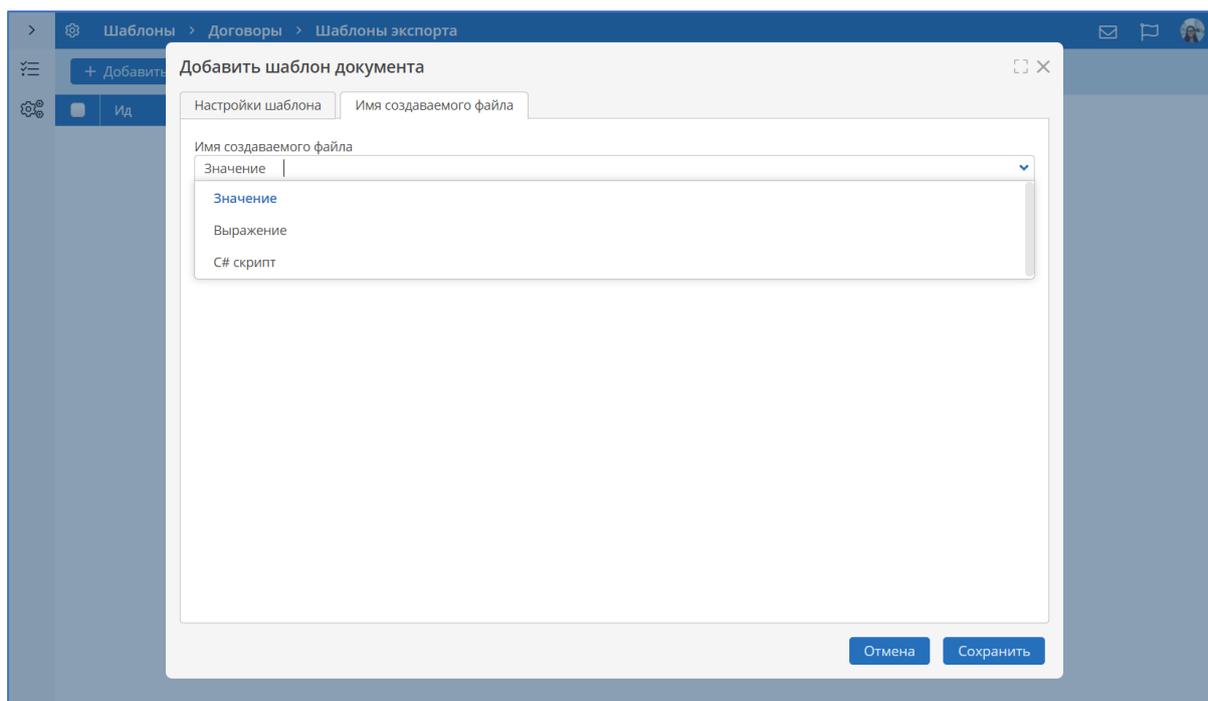


Рис. 185 Настройка имени файла

2.9.2 Шаблон с использованием C#

Создание шаблона экспорта

Шаблон документа в формате .xls

1. Создайте новую книгу excel.
2. Создайте список с данными по типу представленном ниже:

Клиент	Вид отгрузки	Номер распоряжения
&=data.client	&=data.ship_type	&=data.ship_num

Рис. 186 Пример документа формата .xls

Где &=data.свойство_класса

Свойства класса – структура подготовки данных, которые определяются в C# скрипте.

Шаблон документа в формате .doc

3. Создайте новую книгу word.
4. Создайте шаблон с данными по типу представленном ниже:

Client	Contract Price
<<[!Any()]>>No data	
<<[else]>><<foreach [in contracts]>><<[Client.Name]>>	<<[Price]>><</foreach>>
Total:	<<[Sum(c => c.Price)]>><</if>>

Рис. 187 Пример документа формата .doc

Более детальное описание смотрите по ссылке:

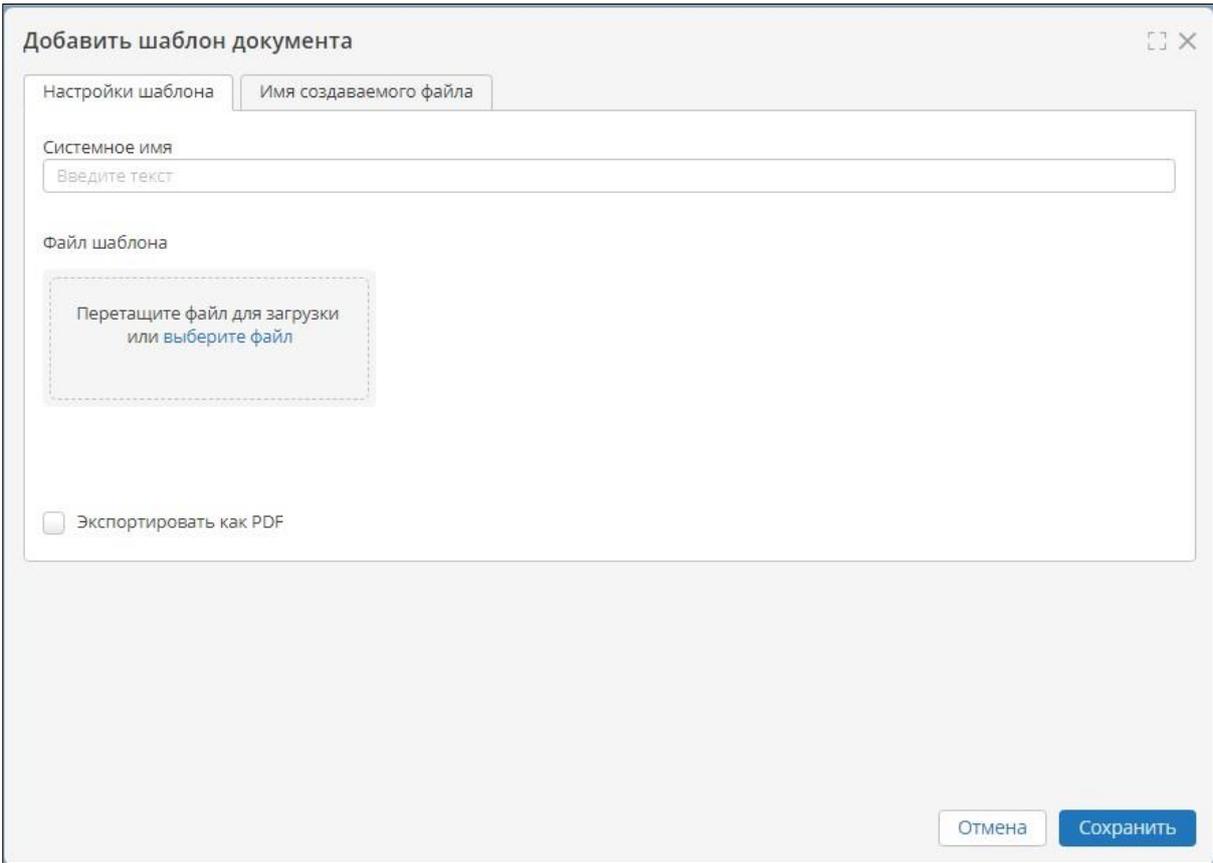
<https://docs.aspose.com/display/wordnet/Typical+Templates>

Загрузка шаблона документа

Для загрузки шаблона перейдите в раздел «**Шаблоны**» – добавить.

1. В окне настройки шаблона документа прикрепите файл с созданным ранее шаблоном в формате .doc или .xls.

После создания шаблона документа автоматически создается операция с одноименным названием шаблона. Для последующей настройки перейдите в раздел [операции](#)⁸⁷ тип «Экспорт записи». При отмеченном флаге «Экспортировать как PDF» документ преобразуется в формат PDF.



Добавить шаблон документа

Настройки шаблона Имя создаваемого файла

Системное имя
Введите текст

Файл шаблона

Перетащите файл для загрузки
или выберите файл

Экспортировать как PDF

Отмена Сохранить

Рис. 188 Загрузка шаблона документа

2. Задайте имя шаблона: напрямую, с помощью языка выражений платформы²²³ или скрипта C#.

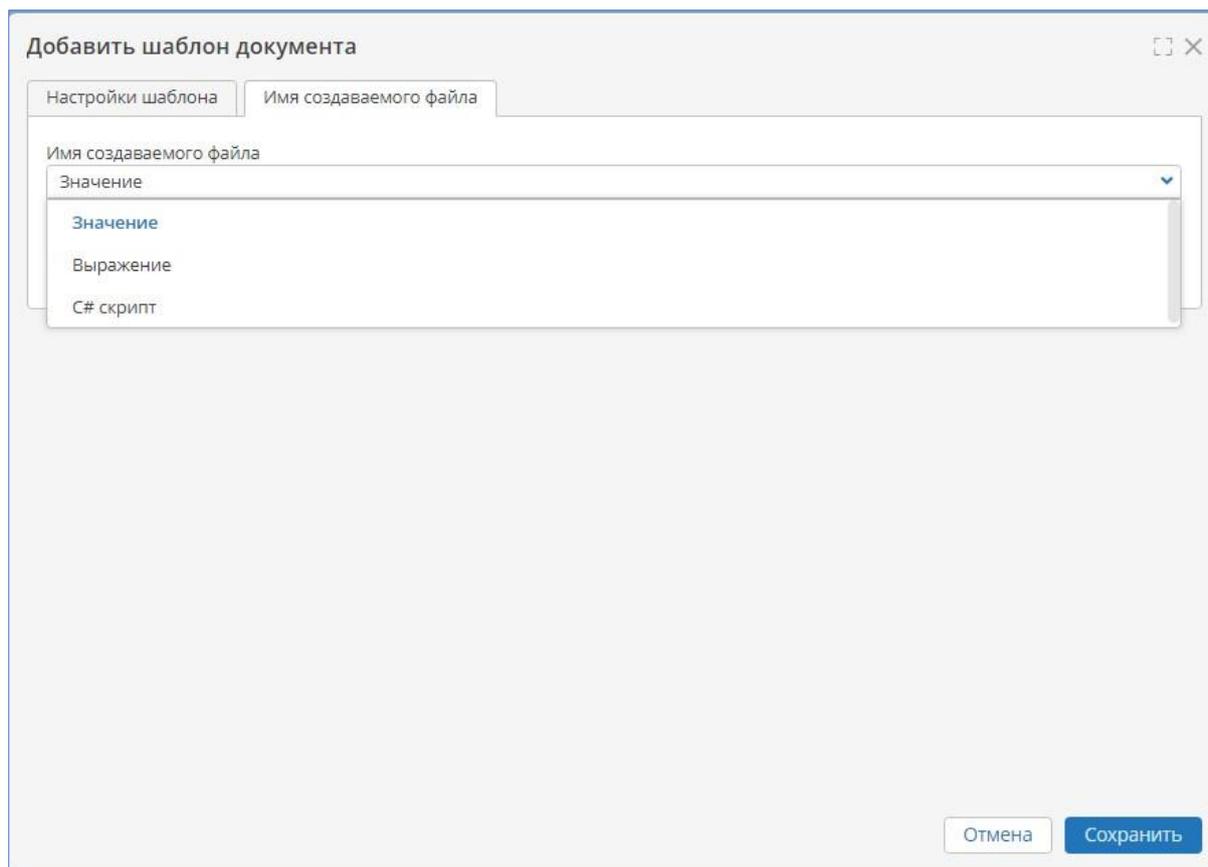


Рис. 189 Настройка наименования файла

Подготовка данных производится путем заполнения структуры данных в C# скрипте операции.⁸⁷ Пример структуры приведен ниже:

```
[Serializable]
public class Position
{
public string prod_name
public string client
public string ship_type
public DateTime? ship_date
public string ship_decade
public string ship_station
public decimal? quantity
}

var
res=Api.TeamNetwork.ObjectAppExportService.ExecuteWordExportTemplate(userCommandContext.DocumentTemplateId,dataToExport,true
);

    var result = new UserCommandResult
    {
        Success = true,
        Committed = true,
        File=new UserCommandFileResult()
        {
            Content=res,
            Name = userCommandContext.ExportAsPdf ?userCommandContext.FileName + ".pdf" : userCommandContext.FileName +
".docx",
            Type = userCommandContext.ExportAsPdf ? "PDF" : "Word"
        },
        ResultType = UserCommandResultType.Notificate,
        Messages = new[]
    };
```

2.10 Импорт данных записей

Для того, чтобы импортировать записи в формате CSV/XLS перейдите в раздел «Импорт» шаблона, для которого нужно настроить шаблон экспорта. Для этого кликните по иконке пользователя в правом верхнем углу и перейдите в раздел **Шаблоны**.

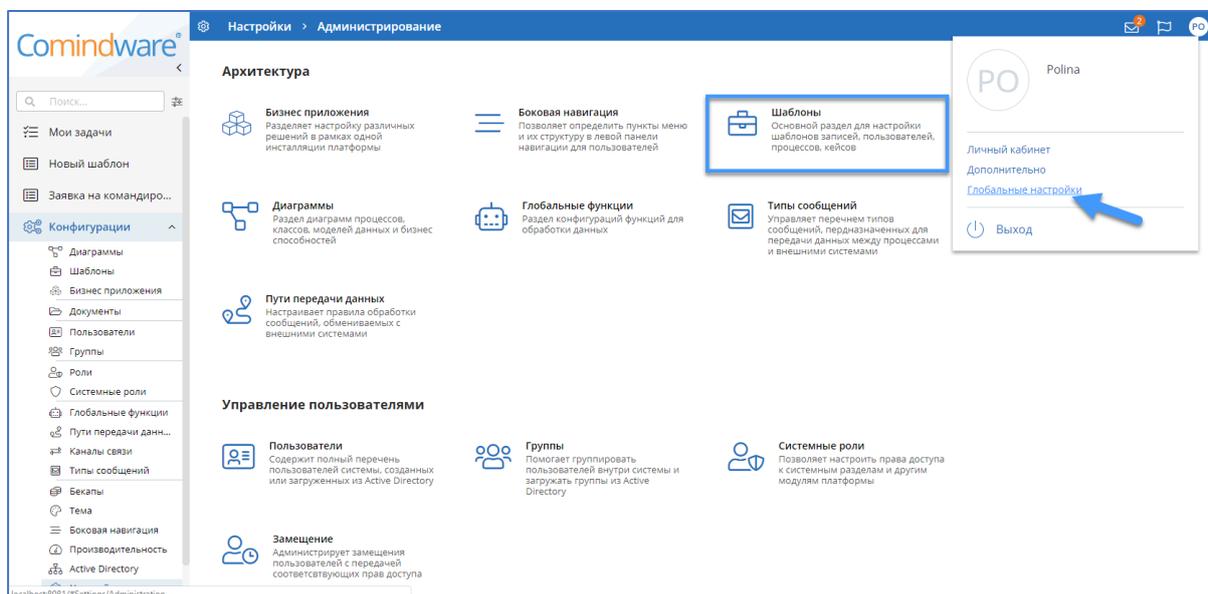


Рис. 190 Переход к списку шаблонов

Затем выберите нужный шаблон и перейдите в раздел **Импорт**.

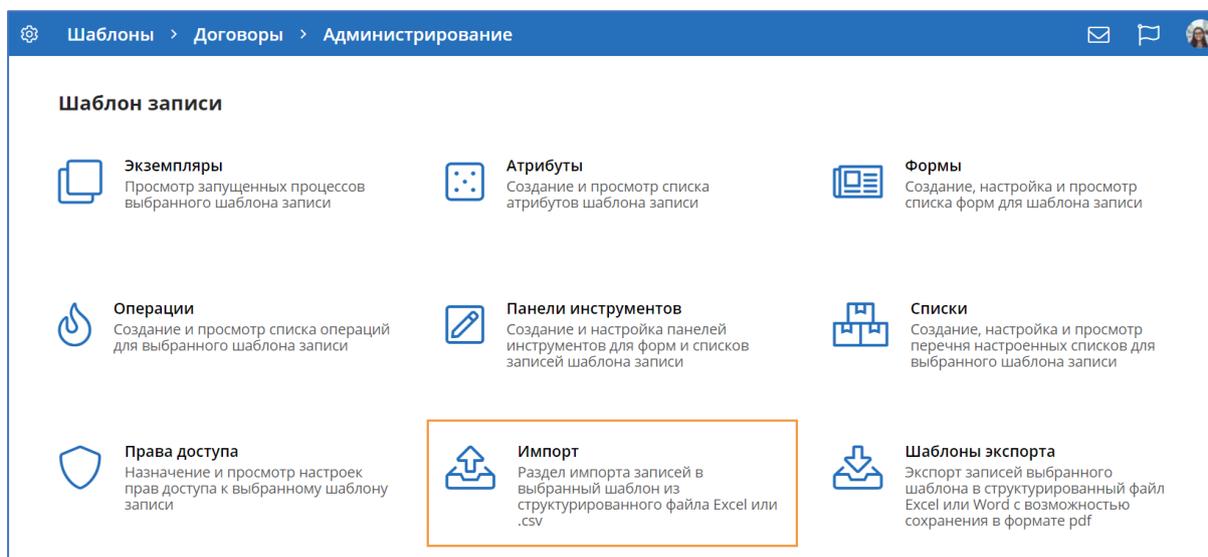


Рис. 191 Переход в раздел Шаблоны экспорта

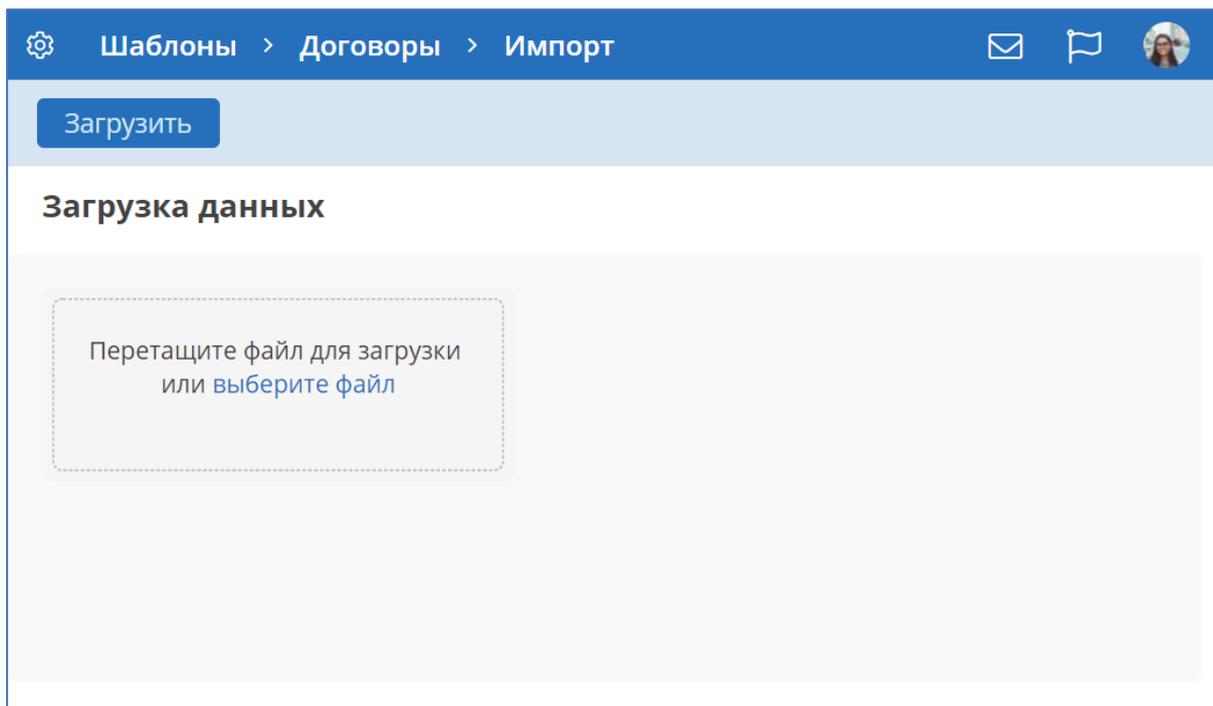


Рис. 192 Раздел **Импорт**

Перетащите файл для загрузки в поле «Загрузка данных».

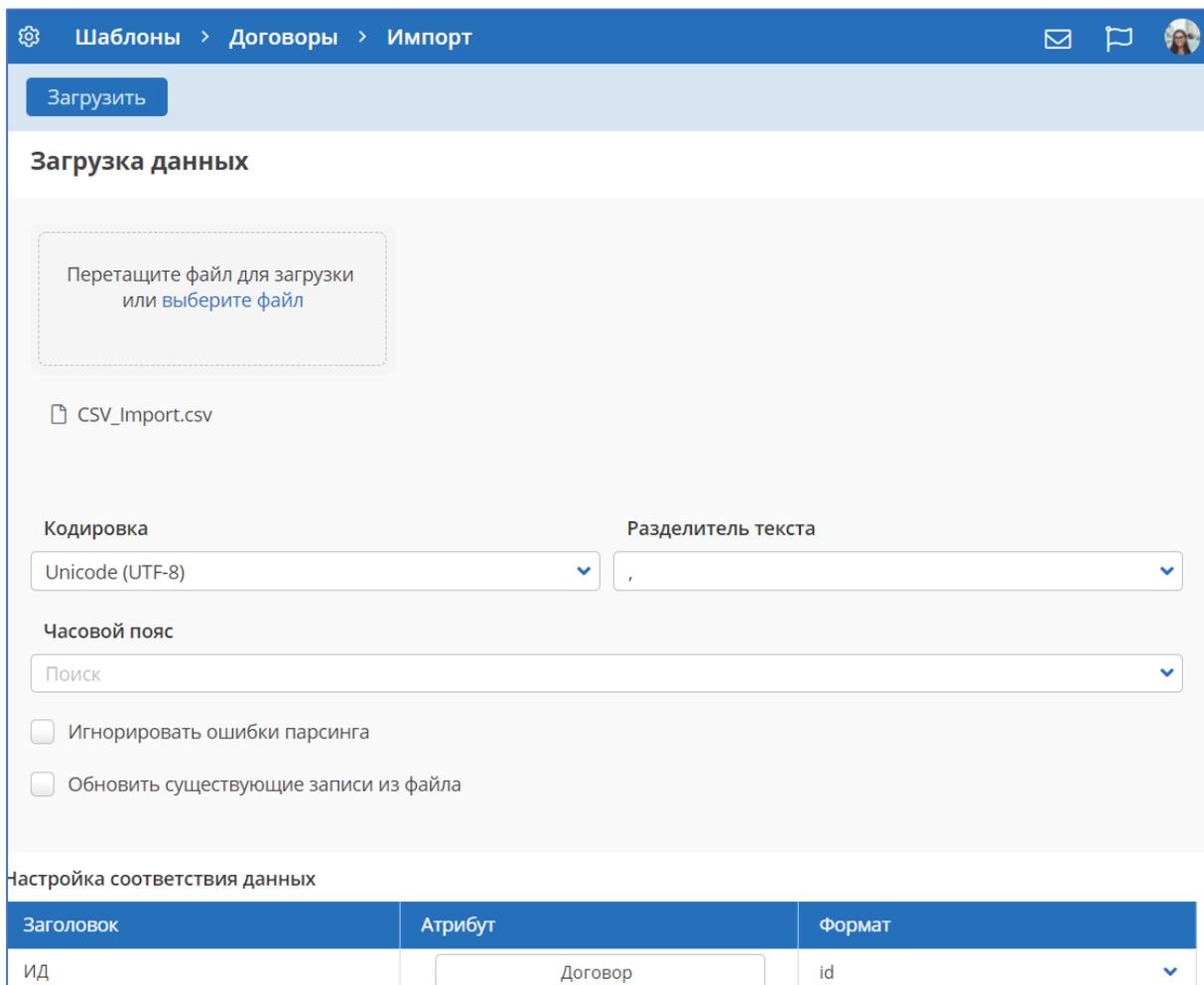


Рис. 193 Раздел **Импорт** с загруженным файлом

Платформа позволяет выбрать:

- Кодировка – выпадающий список выбора кодировки файла импорта.
- Разделитель текста – выпадающий список выбора разделителя текста в файле импорта.
- Часовой пояс – выпадающий список выбора часового пояса.
- Игнорировать ошибки парсинга – логическое поле, которое выбирается выставлением флага игнорирования ошибок парсинга.
- Обновить существующие записи – логическое поле, которое выбирается выставлением флага обновления существующих записей.

Настройте соответствие данных файла импорта и атрибутов шаблона записи.

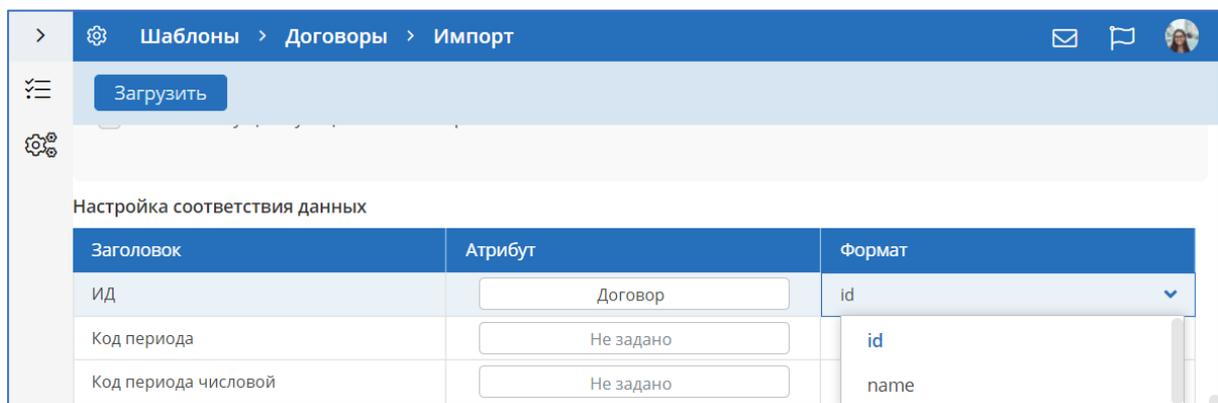


Рис. 194 Настройка соответствия данных

После настройки нажмите на кнопку **Загрузить**.

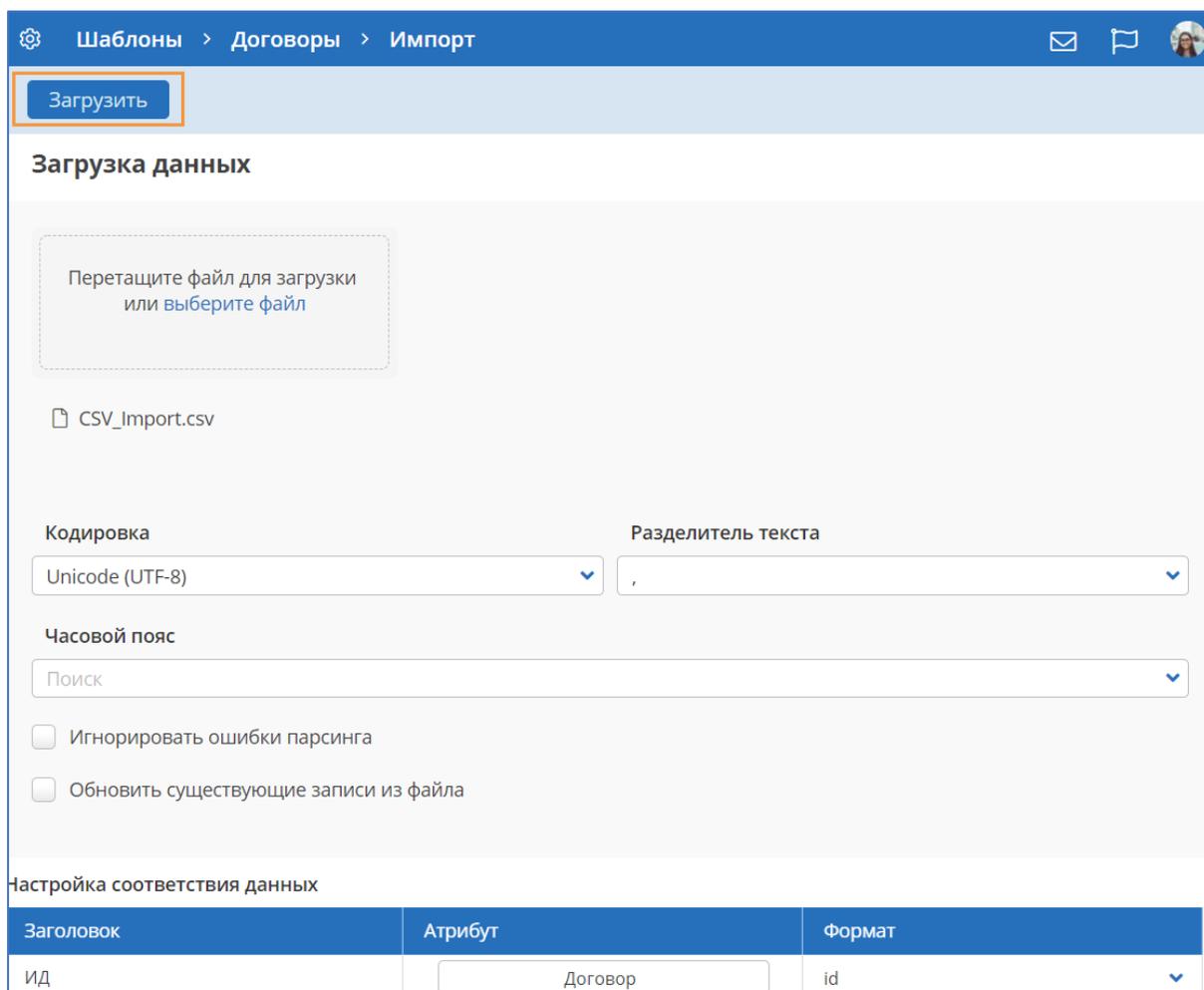


Рис. 195 Расположение кнопки «Загрузить».

2.11 Страницы

Для просмотра статистики за выбранный период в Comindware Business Application Platform есть инструмент для создания дашбордов. Дашборды позволяют представить данные таблиц и статистику процессов в графическом виде.

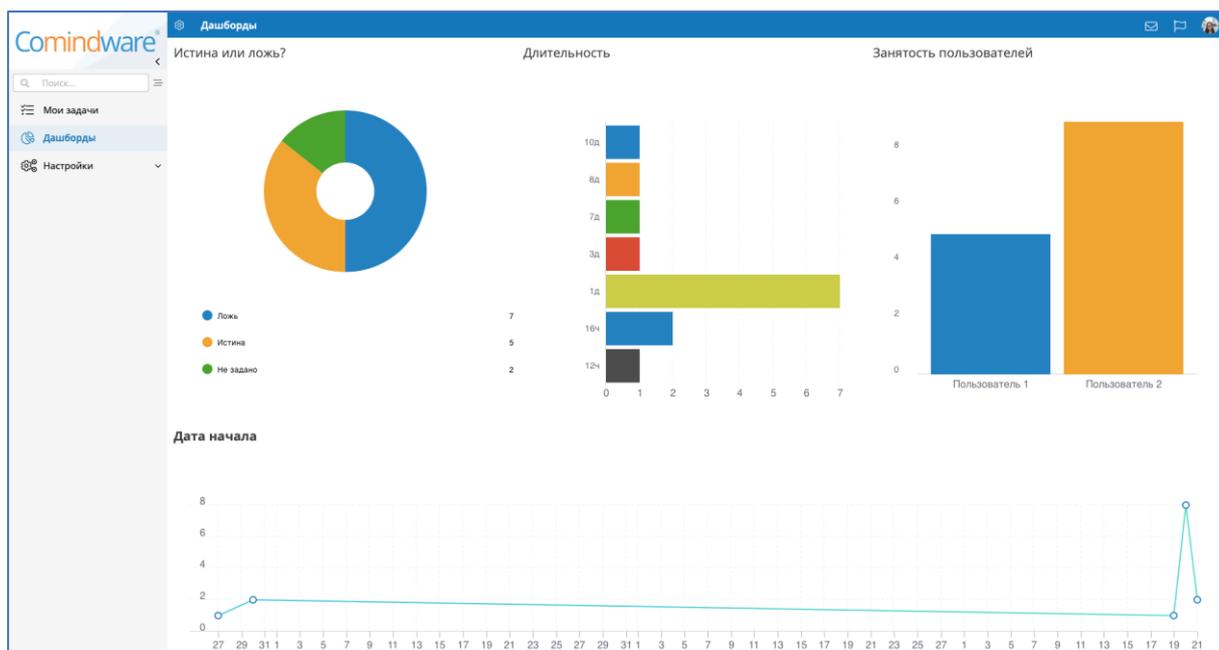


Рис. 196 Пример настроенных дашбордов

Для создания и настройки страницы в бизнес-приложении должно быть не менее одного созданного шаблона записи и созданные экземпляры шаблона записи. На списке должны быть вынесены колонки с данными, которые будут представлены на Страницах.

Ниже приведен пример создания Страницы для настроенного списка.

Создан	Дата создания	Истина или ложь?	Дата начала	Дата окончания	Длительность	Число	Пользователи	DateLink
<input type="checkbox"/>	Администратор	20.03.2019 18:25		30 дек. 2018 г.		1д		20 мар. 2019 г.
<input type="checkbox"/>	Администратор	20.03.2019 18:22	✓	30 дек. 2018 г.	12 мар. 2019 г.	1д		20 мар. 2019 г.
<input type="checkbox"/>	Администратор	20.03.2019 18:21		27 дек. 2018 г.	6 мар. 2019 г.	8д	7	20 мар. 2019 г.
<input type="checkbox"/>	Администратор	20.03.2019 18:17	✗	19 мар. 2019 г.	28 мар. 2019 г.	1д	10	20 мар. 2019 г.
<input type="checkbox"/>	Администратор	20.03.2019 18:12	✓	20 мар. 2019 г.	26 мар. 2019 г.	16ч	6	20 мар. 2019 г.
<input type="checkbox"/>	Администратор	20.03.2019 13:04	✗	20 мар. 2019 г.	26 мар. 2019 г.	3д	10	20 мар. 2019 г.
<input type="checkbox"/>	Администратор	20.03.2019 13:04	✗	20 мар. 2019 г.	28 мар. 2019 г.	10д	10	20 мар. 2019 г.
<input type="checkbox"/>	Администратор	20.03.2019 13:04	✓	21 мар. 2019 г.	26 мар. 2019 г.	1д	7	20 мар. 2019 г.
<input type="checkbox"/>	Администратор	20.03.2019 13:04	✓	21 мар. 2019 г.	26 мар. 2019 г.	16ч	7	20 мар. 2019 г.
<input type="checkbox"/>	Администратор	20.03.2019 13:04	✗	20 мар. 2019 г.	27 мар. 2019 г.	7д	7	20 мар. 2019 г.
<input type="checkbox"/>	Администратор	20.03.2019 13:04	✗	20 мар. 2019 г.	27 мар. 2019 г.	12ч	10	20 мар. 2019 г.
<input type="checkbox"/>	Администратор	20.03.2019 13:02	✓	20 мар. 2019 г.	26 мар. 2019 г.	1д	7	20 мар. 2019 г.
<input type="checkbox"/>	Администратор	20.03.2019 13:02	✗	20 мар. 2019 г.	26 мар. 2019 г.	1д	10	20 мар. 2019 г.
<input type="checkbox"/>	Администратор	20.03.2019 13:01	✗	20 мар. 2019 г.	27 мар. 2019 г.	1д	10	20 мар. 2019 г.

Рис. 197 Настроенный список экземпляров шаблона записи

Чтобы создать страницу:

1. Выберите бизнес-приложение, для которого будут настраиваться Страницы и перейдите в раздел **Страницы**.

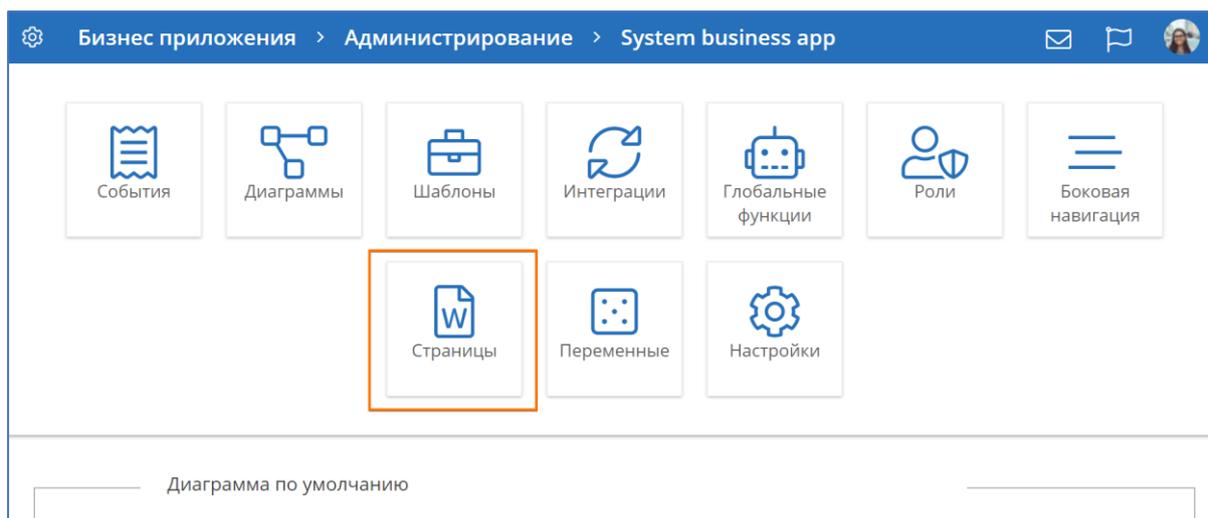


Рис. 198 Переход в раздел Страницы

Отобразится список с настроенными Страницами.

2. Нажмите «+Добавить».

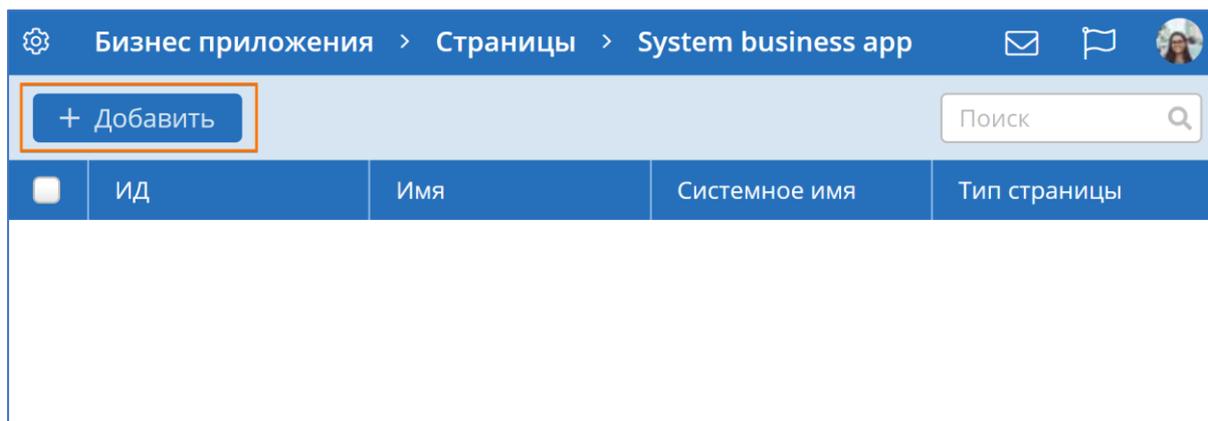


Рис. 199 Добавление новой Страницы

3. Укажите название и нажмите «Сохранить».

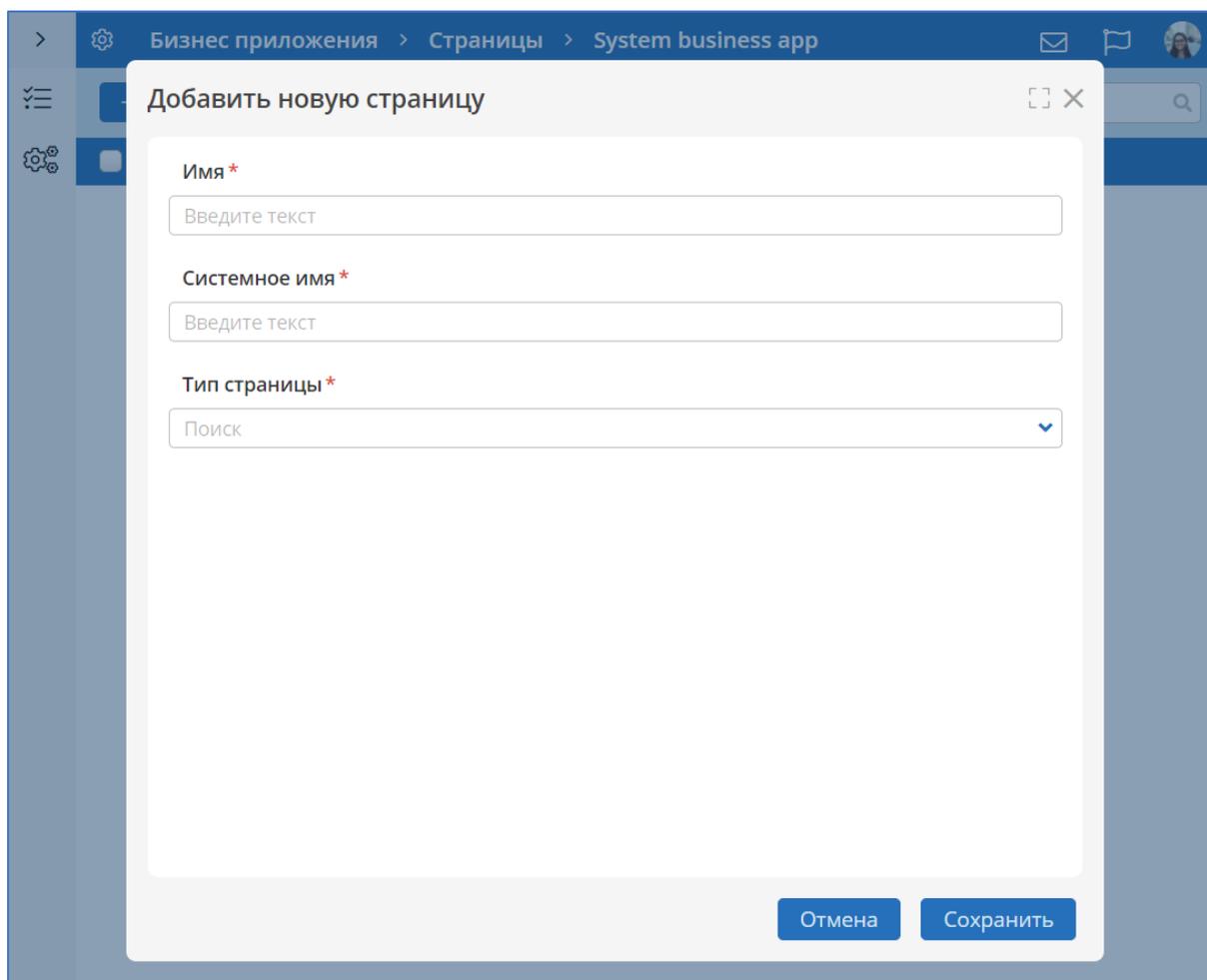


Рис. 200 Окно создания новой страницы

Отобразится конструктор страницы.

4. На конструкторе нажмите кнопку «Создать диаграмму» - отобразится окно для создания диаграммы.

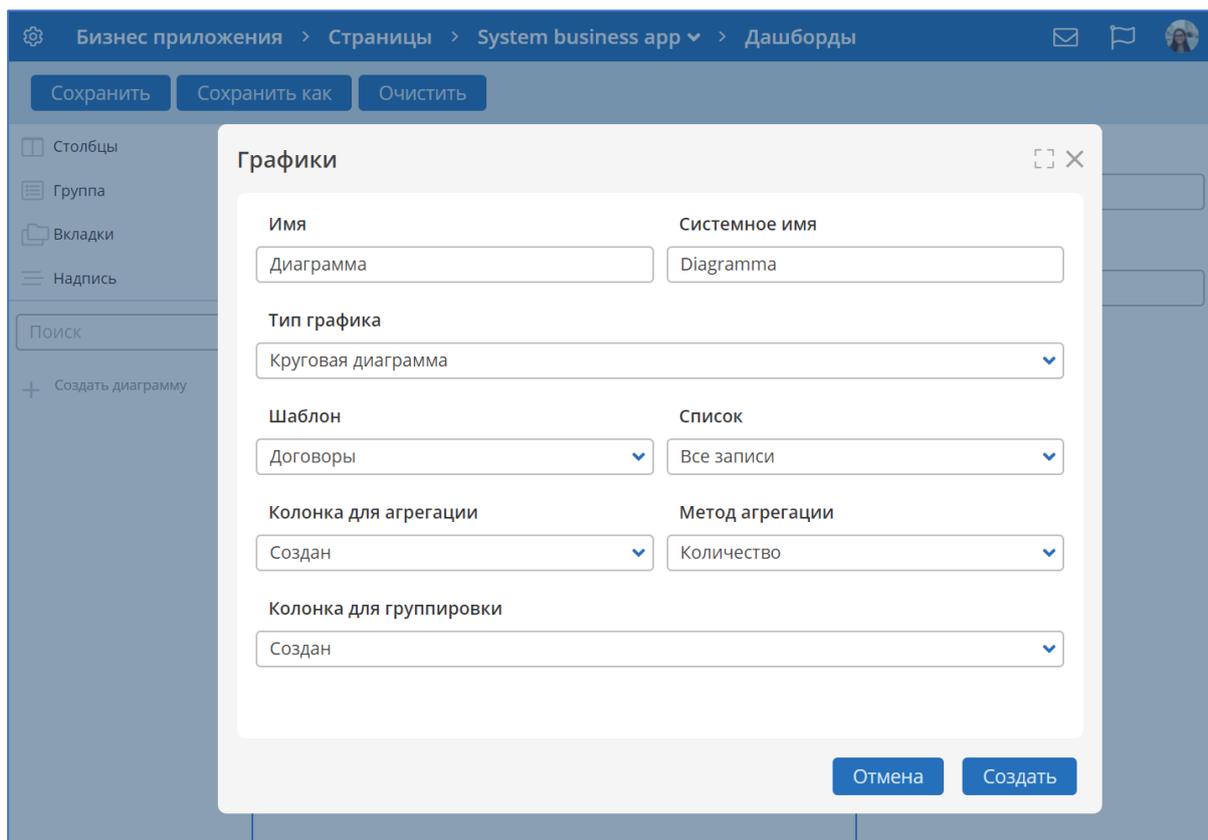


Рис. 201 Создание диаграммы

5. Заполните следующие данные:

- Тип графика – вид диаграммы, в котором будут представлены данные.
- Шаблон – шаблон записи, данные которого будут представлены в выбранном виде графика.
- Датасет - список, по данным которого будет формироваться диаграмма.
- Метод агрегации:
 - Количество – диаграмма будет представлена по количеству созданных экземпляров шаблона записи с выбранными значениями;
 - Сумма – диаграмма будет представлена по сумме созданных экземпляров шаблона записи с выбранными значениями.

6. После этого нажмите «Создать».

Созданная диаграмма отобразится на левой панели конструктора Страниц.

7. Перенесите созданную диаграмму с левой панели на рабочее поле конструктора Страниц.

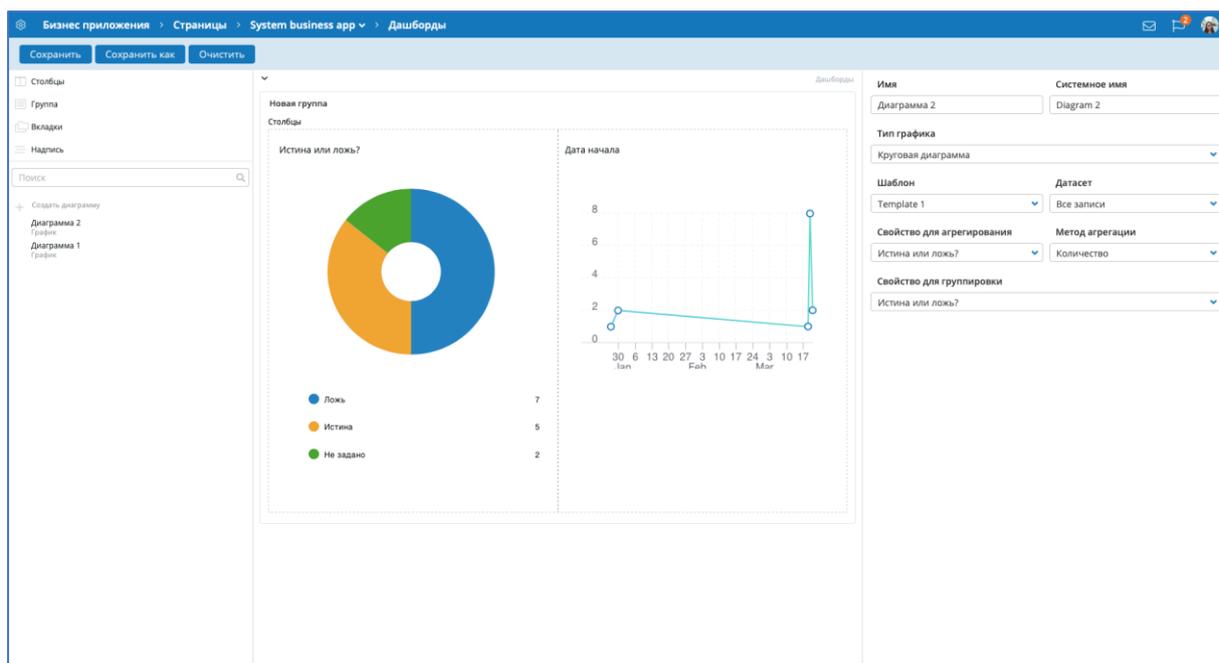


Рис. 202 Созданная диаграмма на рабочем поле конструктора дашбордов

На правой панели конструктора отображаются свойства диаграммы, доступные для редактирования.

Для сохранения Страницы нажмите «Сохранить» на панели инструментов конструктора Страниц.

Для удобства, настроенную Страницу можно вынести на боковую панель.

3. Управление процессами и кейсами

В этой главе:



[Введение в управление процессами](#) ^{3.1}

Основы управления процессами в Comindware Business Application Platform.



[Элементы процессной схемы](#) ^{3.3}

Подробная информация о задачах, событиях, подпроцессах, шлюзах и других элементах процессной схемы.



[Проверка и публикация схемы процесса](#) ^{3.5}

Информация о том, как проверить и опубликовать схему процесса.



[Настройка формы процесса](#) ^{3.7}

Задание формы, которая будет открываться при запуске процесса.



[Создание процесса](#) ^{3.9}

Информация о том, как запустить процесс.



[Импорт и экспорт данных процесса](#)

Информация о том, как импортировать и экспортировать шаблоны процессов.



[Создание шаблона кейса](#) ^{3.12}

Общая инструкция о том, как создать и настроить шаблон кейса.



[Создание шаблона процесса](#) ^{3.2}

Общая инструкция о том, как создать и настроить шаблон процесса.



[Настройка бизнес-правил](#) ^{3.4}

Настройка действий, которые должны быть автоматически выполнены до начала или после завершения того или иного этапа процесса.



[Управление версиями схемы процесса](#) ^{3.6}

Информация о том, как работать с версиями процессной схемы.



[Межпроцессное взаимодействие](#) ^{3.8}

Настройка взаимодействия между процессами.



[Списки](#) ^{3.10}

Создание и настройка списков записей.



[Введение в управление кейсами](#)

Основы управления кейсами в Comindware Business Application Platform



[Настройка доступа к процессам и кейсам](#) ^{3.13}

Настройка доступа пользователей к шаблонам и экземплярам процессов и кейсов.

3.1 Введение в управление процессами

Процесс представляет собой последовательность связанных между собой и структурированных активностей (шагов или задач процесса), выполнение каждой из которых служит определенной цели.

- Comindware Process поддерживает последнюю версию стандарта управления бизнес-процессами BPMN 2.0. Бизнес-процессы настраиваются в дизайнера процессов, а управляются в виде запускаемых экземпляров процессов.
- Специфические данные процесса хранятся в ассоциированном с процессом шаблоне записи.
- Процессы могут использовать данные, хранящиеся как в ассоциированном шаблоне записи, так и в других шаблонах записей.
- Процессы могут взаимодействовать друг с другом, а также с внешними ресурсами, такими как серверы электронной почты и веб-сервисы.
- Процесс может включать в себя кейсы — важные для компании бизнес-действия, которые на данном этапе не могут быть формализованы в виде этапов процесса.
- Процесс может быть запущен из записей, из списков записей, из других процессов, а также вручную в разделе **Экземпляры** выбранного шаблона процесса.
- При выполнении процесса возможен вызов глобальных функций (выполнение скрипта C# или вызов веб-службы).

3.2 Создание шаблона процесса

Для того чтобы создать шаблон процесса:

1. На боковой панели навигации открыть раздел **Шаблоны**, затем нажать **«+Добавить»** для создания **Нового шаблона процесса**.

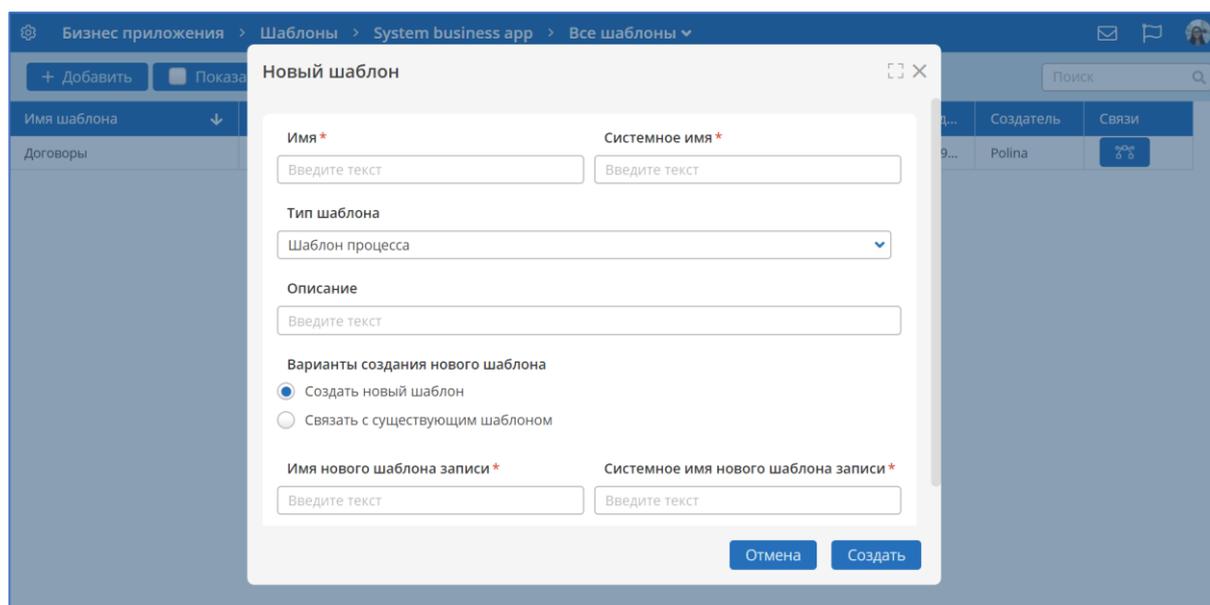


Рис. 203 Окно создания нового шаблона процесса.

2. Выбрать вариант создания нового шаблона процесса: создать новый шаблон записи или связать с существующим шаблоном записи. При выборе пункта «Связать с существующим

шаблоном» в выпадающем списке нужно выбрать созданный шаблон записи для создания связи.

Новый шаблон

Введите текст

Введите текст

Тип шаблона*

Шаблон процесса

Бизнес-приложение*

System business app

Описание

Введите текст

Варианты создания нового шаблона

Создать новый шаблон

Связать с существующим шаблоном

Связанный шаблон*

Поиск

Адрес доставки

Договор

Контрагент

В

Рис. 204 Варианты создания нового шаблона

3. Выбрать «Создать новый шаблон» для создания шаблона и указать ему имя.
В списке шаблонов появится созданный шаблон процесса.
4. Выбрать шаблон процесса из списка и перейти в раздел **Схема**.

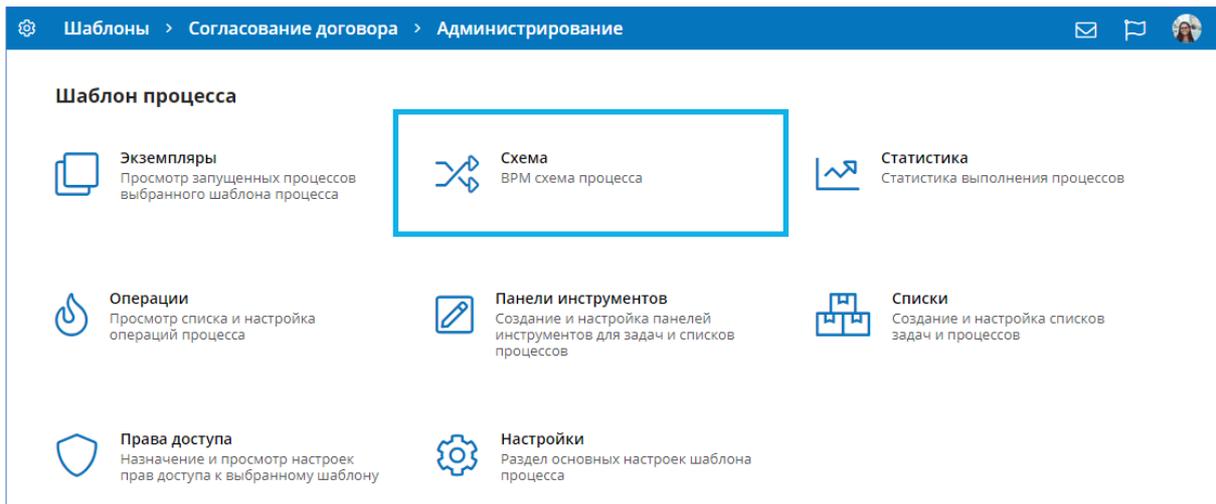
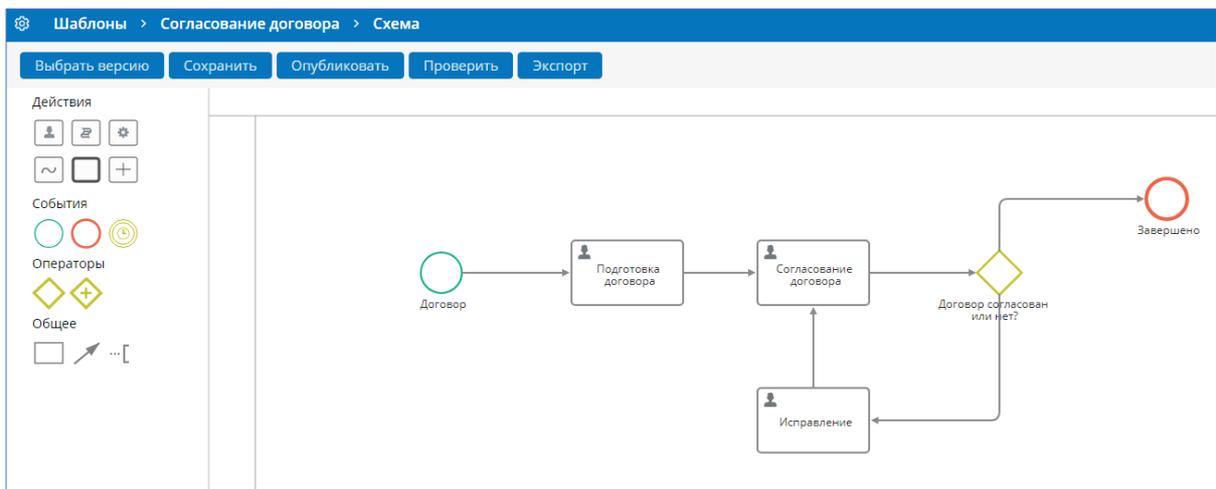


Рис. 205 Переход в раздел **Схема** выбранного шаблона процесса

5. Постройте схему процесса, перетаскивая на форму элементы, расположенные на панели слева.



Для пользовательских задач укажите Исполнителей:

6. Выберите пользовательскую задачу, в появившемся меню нажмите кнопку «Настройки».
7. В Свойствах задачи перейдите во вкладку **Дополнительные свойства**.
8. Выберите Исполнителя.
9. Проверьте и опубликуйте схему процесса.

3.3 Элементы процессной схемы

Comindware Business Application Platform предоставляет широкий выбор BPMN элементов, с использованием которых схемы ваших процессов будут наиболее полно отвечать потребностям вашего бизнеса.

В этом разделе:



[Задачи](#) ^{3.3.1}

Пользовательская задача, скрипт-задача, кейс, глобальная функция.



[Шлюзы](#) ^{3.3.4}

Шлюз исключающего «или» и параллельный шлюз.



[События](#) ^{3.3.2}

Различные типы стартовых, промежуточных и конечных событий. Вы также узнаете, что представляют собой отправляющее и принимающее событие.



[Другие элементы](#) ^{3.3.5}

Поток, дорожка, текстовая ассоциация.



[Подпроцессы](#) ^{3.3.3}

Внешние и вложенные подпроцессы.

3.3.1 Задачи

Задача служит для описания действий, которые должны быть выполнены в ходе выполнения задачи.

В процессе могут присутствовать задачи следующих типов:

Тип задачи	Описание
Пользовательская задача	Представляет собой задачу, выполняемую человеком. При переходе процесса на данный шаг задача автоматически назначается на исполнителя или становится доступной для нескольких исполнителей. Завершение задачи выполняется либо вручную пользователем, либо по настроенному событию. По завершении работы пользовательская задача завершается.
Задача на выполнение сценария	При переходе процесса на данный шаг автоматически выполняется скрипт. По завершении выполнения скрипта скрипт-задача завершается. В случае ошибки статус задачи и процесса изменяется на Ошибка.
Задача на вызов сервиса	Используется для вызова веб-службы или скрипта С#. По завершении выполнения глобальной функции задача завершается.
Кейс	Представляет нестандартную задачу, курируемую ответственным лицом или группой лиц. Кейсы используются, когда нужно быстро организовать в рамках процесса неструктурированную или слабоструктурированную активность. Кейс завершается, когда работа, которую нужно было сделать, выполнена.

Пользовательская задача

Пользовательская задача представляет собой задачу, выполняемую человеком. При переходе процесса на данный шаг задача автоматически назначается на исполнителя или становится доступной для нескольких исполнителей.

При выборе задачи на схеме процесса откроется панель, которая имеет следующие разделы:

- **Действия** - В данном разделе расположены: Настройки задачи, Правила для данных ^{3.4}.
- доступ к **Форме** ³⁶ и правилам для формы ⁸¹ и кнопка для удаления задачи.
- **Изменить тип** – Изменение типа задачи с пользовательского на выполнение сценария, вызов сервиса или встроенный кейс.
- **Быстрое создание** – быстрое добавление связанного элемента на форму.

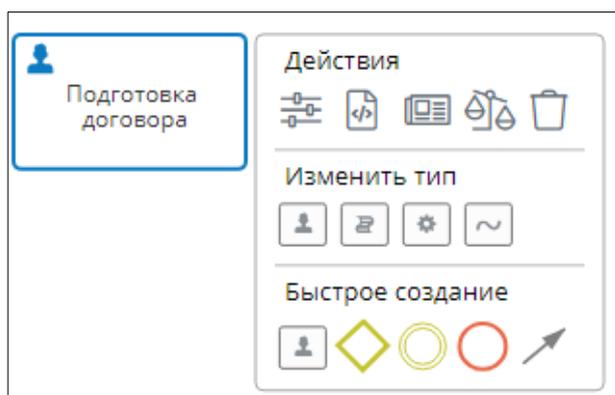


Рис. 206 Свойства пользовательской задачи

При выборе «Настройки»  на панели действий пользовательской задачи откроются основные и дополнительные Свойства задачи. Основные свойства задачи включают в себя системное и отображаемое на схеме имя задачи, а также её описание.

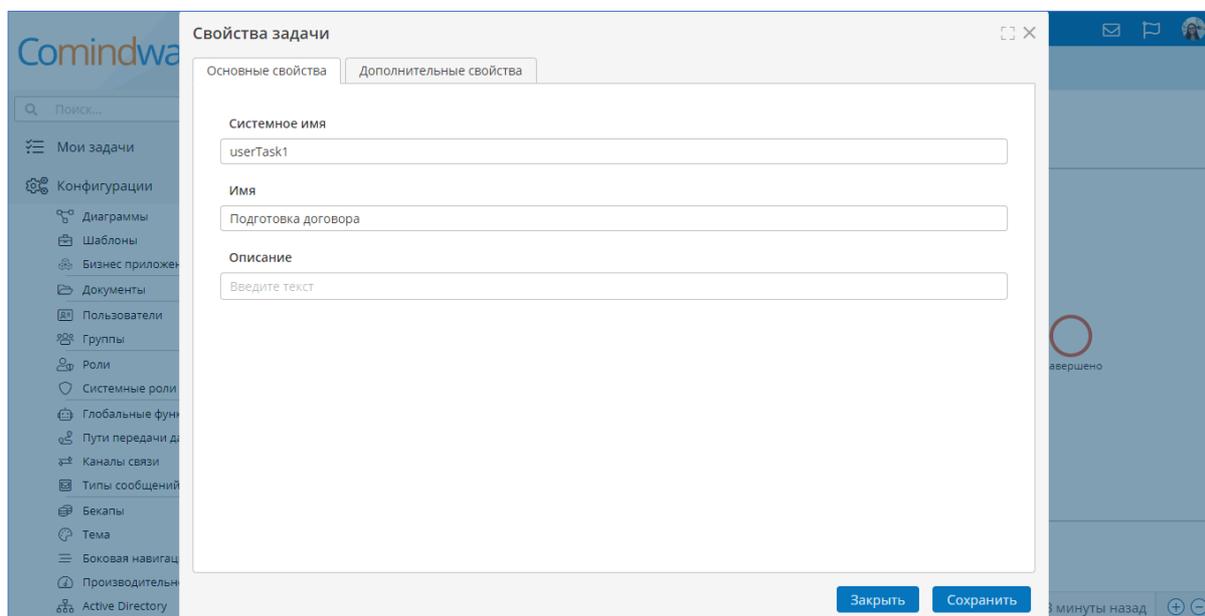


Рис. 207 Настройка Пользовательской Задачи

Дополнительные свойства включают в себя:

- Заголовок задачи, который может быть задан напрямую, с использованием атрибута, значение, языка выражений ²²³ или C# скрипта.
- Исполнители, которые могут быть заданы напрямую, с использованием атрибута, значения, языка выражений ²²³ (например, назначьте исполнителя задачи ²⁶⁴) или с помощью скрипта C#. Если указано несколько потенциальных исполнителей, но не заданы правила назначения задачи, задача будет отображаться у каждого из потенциальных исполнителей в списке **Мои задачи**.
- Флаг возможности переназначения задачи.
- Продолжительность выполнения задачи, которая может быть задана напрямую, с использованием атрибута, значение, языка выражений ²²³ или C# скрипта.

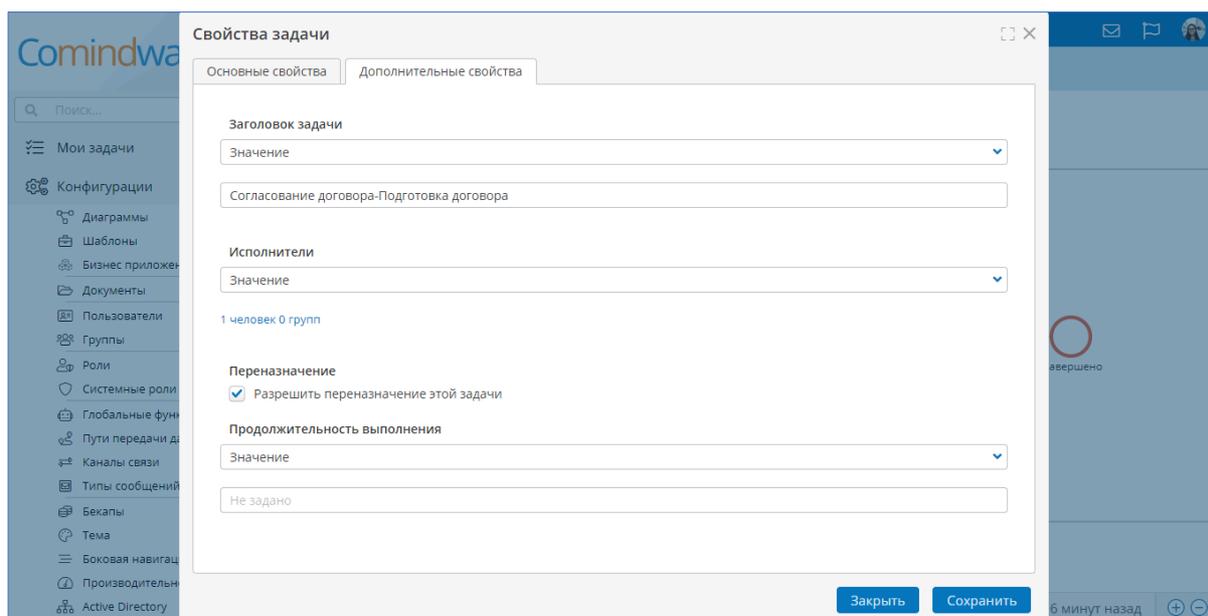


Рис. 208 Настройка Пользовательской Задачи

Задача на выполнение сценария

При активации скрипт-задачи вызывается связанный скрипт. По завершении выполнения скрипта скрипт-задача завершается.

Для того чтобы добавить скрипт-задачу в шаблон процесса:

Перетаскиванием добавьте на схему иконку задачи на выполнение сценария из группы Задачи.

При выборе задачи на схеме процесса откроется панель, которая имеет следующие разделы:

- **Действия** - В данном разделе расположены: Настройки сценария, Правила для данных ,
и кнопка для удаления сценарий.
- **Изменить тип** – Изменение типа с сценария на выполнение пользовательскую задачу, вызов сервиса или встроенный кейс.
- **Быстрое создание** – быстрое добавление связанного элемента на форму.

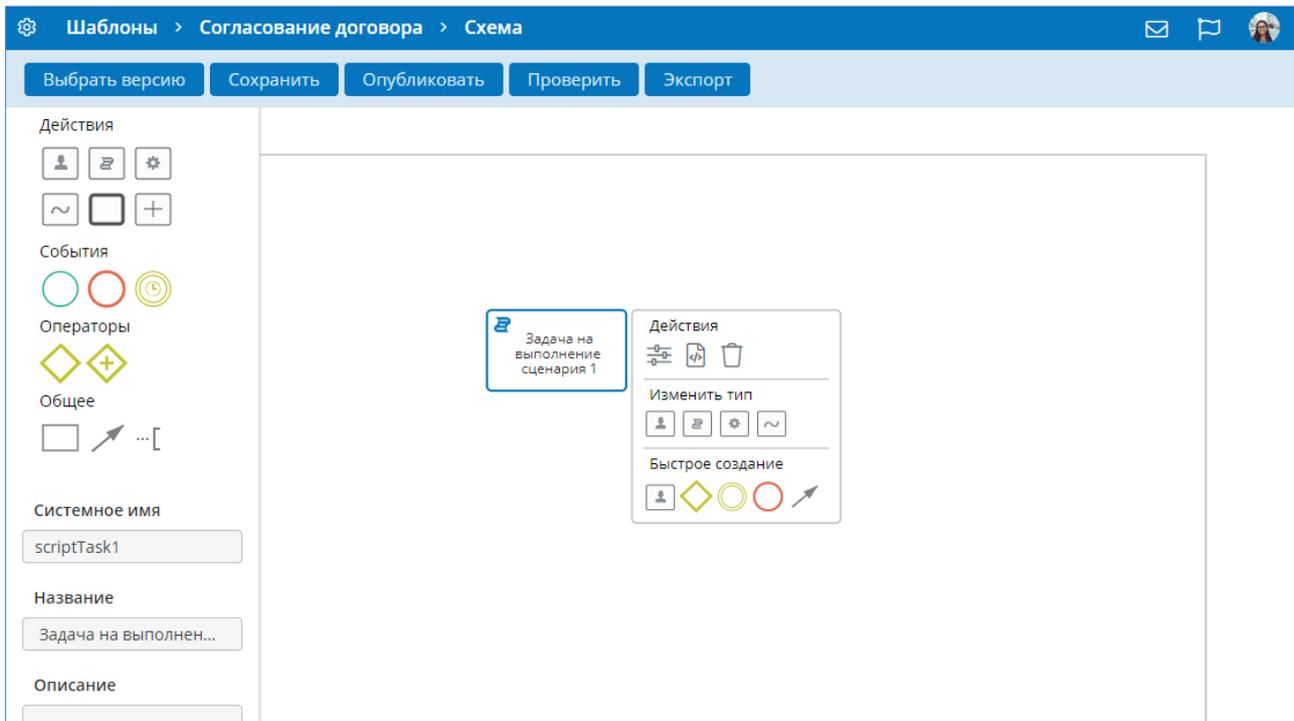


Рис. 209 Панель действий с задачей на выполнение сценария

В разделе настроек задачи на выполнение сценария задаются основные и дополнительные свойства. Основные свойства задачи включают в себя системное и отображаемое на схеме имя задачи, а также её описание. Дополнительные свойства включают в себя поле, в которое необходимо ввести C# скрипт, который будет выполняться при активации задачи.

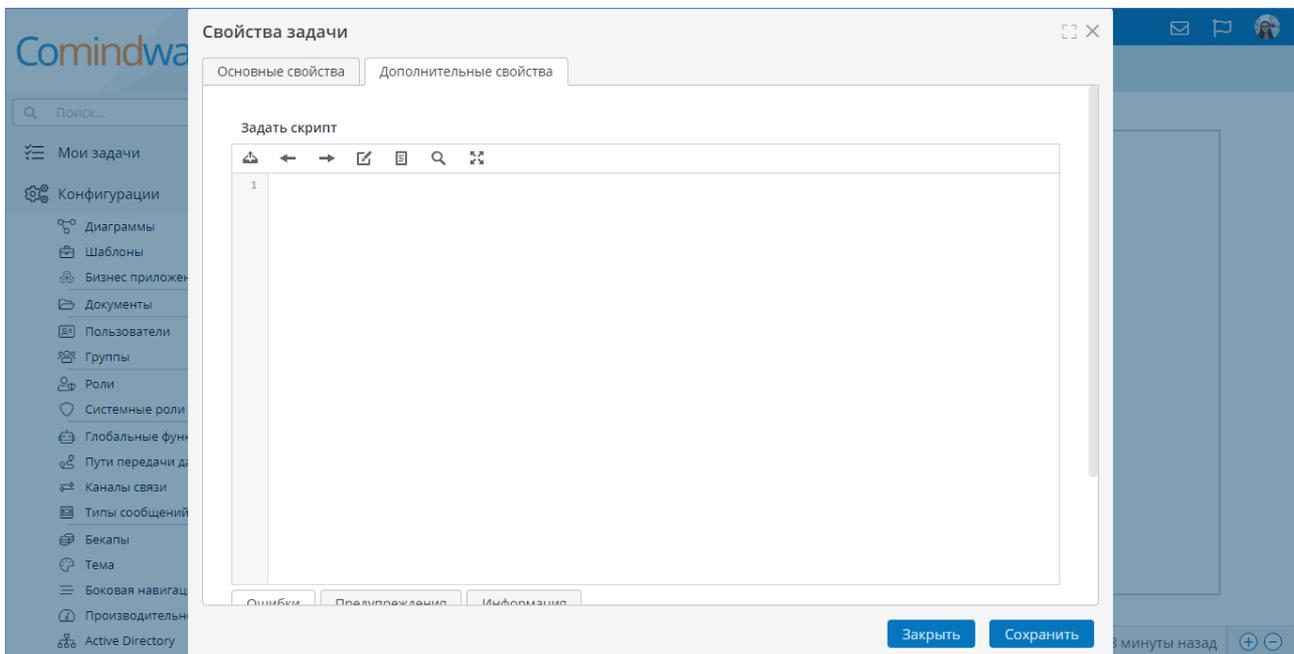


Рис. 210 Дополнительные свойства задачи на выполнение сценария

Задача на вызов сервиса

Задача на вызов сервиса представляет собой вызов глобальной функции в ходе процесса. По завершении выполнения глобальной функции задача завершается.

В платформе выделяют два типа глобальных функций:

- **Скрипт C#:** платформа выполняет заданный скрипт C#.
- **Вызов веб-службы:** платформа выполняет API запрос к внешней веб-службе (например, к сервису тех. поддержки, базе знаний и т. д.).

Для того чтобы добавить задачу-функцию в шаблон процесса:

Создайте сигнатуру глобальной функции.

Перетаскиванием добавьте на схему иконку задачи на вызов сервиса из группы Действия.

В разделе действия задачи отображены настройки, правила для данных и кнопка для удаления сервиса.

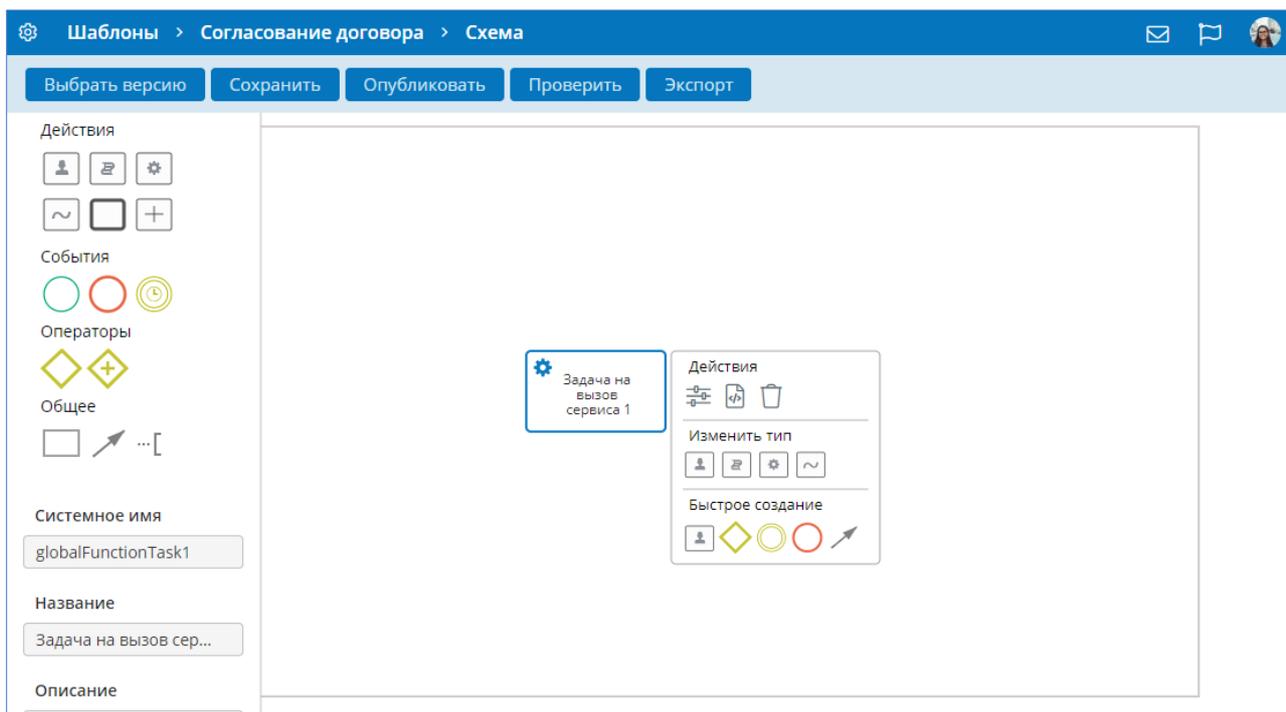


Рис. 211 Панель действий с задачей на вызов сервиса

В разделе настроек задачи на вызов сервиса задаются основные, дополнительные свойства, настройка данных на входе и настройка данных на выходе. Основные свойства задачи включают в себя системное и отображаемое на схеме имя задачи, а также её описание. В разделе дополнительные свойства указывается глобальная функция. Из раскрывающегося списка в поле **Глобальная функция** выберите созданную ранее функцию. В результате появятся ее входные и выходные данные. Настройте соответствие между входными и выходными атрибутами функции и данными процесса. Входные атрибуты задаются значением напрямую, с помощью языка выражений платформы²²³ или C# скрипта, выбором атрибута текущего процесса из выпадающего списка. Выходные атрибуты задаются только выбором атрибута процесса из выпадающего списка.

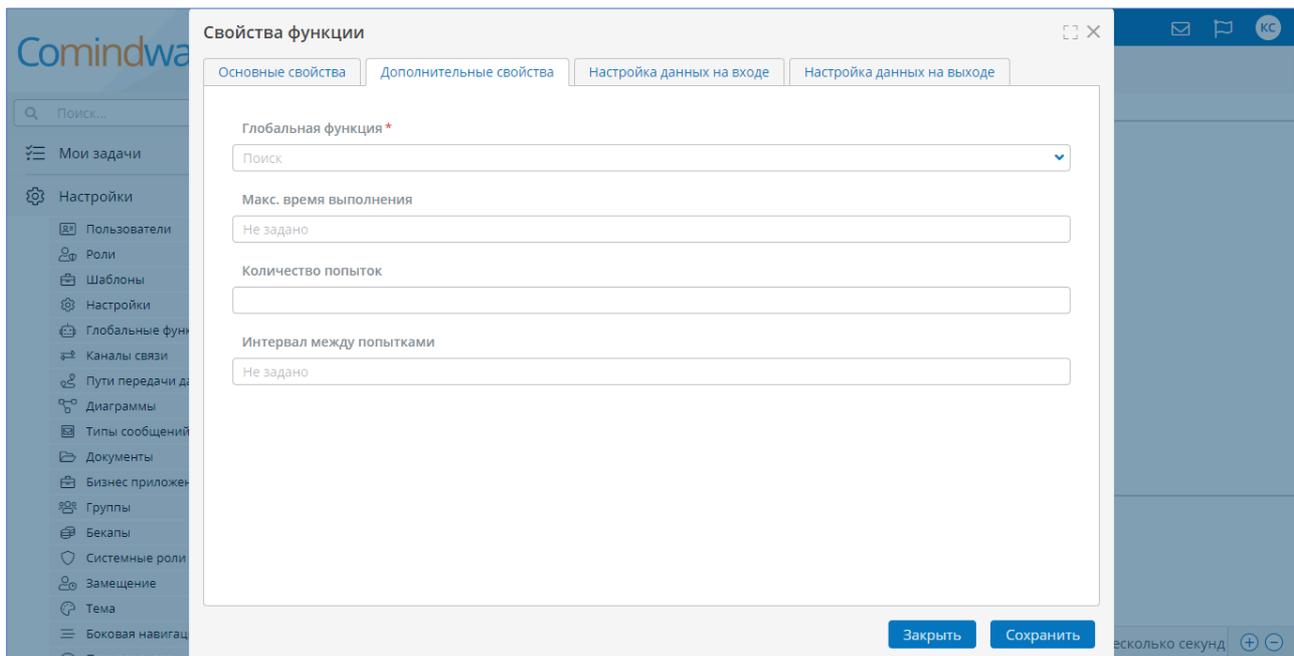


Рис. 212 Дополнительные свойства задачи на вызов сервиса

Далее необходимо задать определение **Глобальной функции**.

Кейс

Кейс представляет собой нестандартную задачу, курируемую ответственным лицом или группой лиц. Кейсы используются, когда нужно быстро организовать неструктурированную или слабоструктурированную активность в рамках процесса.

Для того чтобы добавить кейс в шаблон процесса перетаскиванием добавьте на схему иконку кейса из группы Действия.

В разделе **действия задачи** отображены настройки, правила для данных и кнопка для удаления кейса.

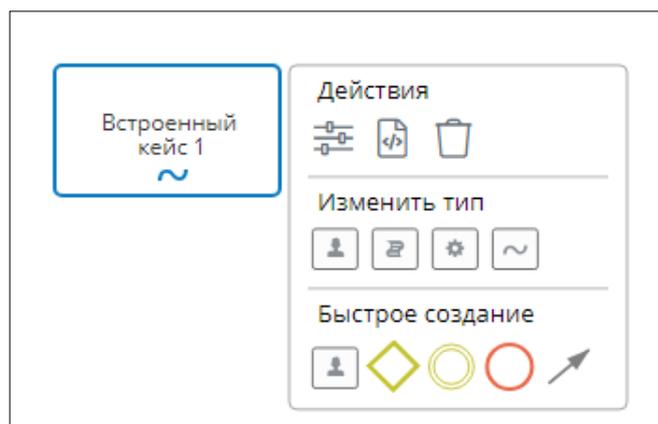


Рис. 213 Панель действий с кейсом

В разделе **настроек кейса** задаются основные и дополнительные свойства. Основные свойства задачи включают в себя системное и отображаемое на схеме имя задачи, а также её описание. Дополнительные свойства включают в себя следующие поля:

- Заголовок кейса

- Исполнители – пользователи, которые могут быть исполнителями кейса. Существует 3 способа это сделать: напрямую, с помощью языка выражений платформы^[223] (например, назначьте исполнителя задачи^[264]) или скрипта C#. Если указано несколько потенциальных исполнителей, но не заданы правила назначения кейса, кейс будет отображаться у каждого из них в списке **Доступные**.
- Продолжительность выполнения кейса: напрямую, с помощью языка выражений платформы²²³ или скрипта C#.
- Шаблон кейса, созданный ранее.
- Ссылка на ассоциированный шаблон записи – ссылка, которая связывает шаблон кейса и шаблон записи.

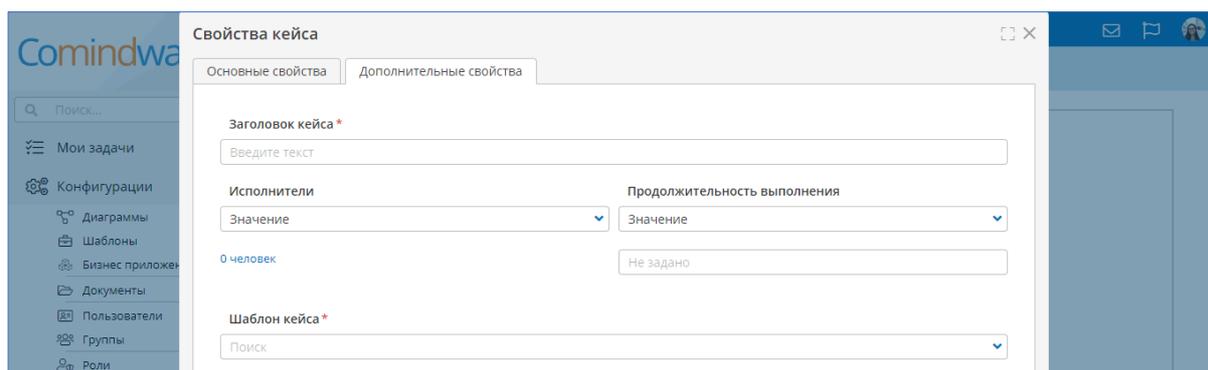


Рис. 214 Дополнительные настройки кейса

3.3.2 События

Событие служит для описания того, что должно случиться (в отличие от задачи, когда что-то должно быть сделано).

События можно классифицировать несколькими способами:

- Как **Стартовое**, **Промежуточное** и **Конечное**, в зависимости от положения события на схеме процесса.
- По типу события:

Тип события	Описание
Простое событие	Представляет не типизированное событие.
Событие-сообщение	Показывает отправку или получение сообщения.
Событие-таймер	Используется для моделирования регулярных событий. Также таймер может использоваться для моделирования моментов времени, временных промежутков и превышения лимита времени.

- Как **Принимающее событие** (например, когда получение сообщения инициирует процесс) и **Отправляющее событие** (например, когда завершение процесса сопровождается отправкой сообщения). Данная классификация в основном используется, когда речь заходит о межпроцессном взаимодействии, а также о взаимодействии с внешними ресурсами.

Стартовое событие

Для того чтобы добавить стартовое событие в шаблон процесса:

Перетаскиванием добавьте на схему процесса иконку стартового события из группы События.

Стартовое событие имеет следующие типы:

Тип события	Описание	Иконка
Простое стартовое событие	Процесс инициируется не типизированным событием.	
Старт по сообщению	Получение сообщения (например, от другого процесса или внешнего ресурса) инициирует процесс.	
Старт по таймеру	Процесс запускается в определенный момент времени или по расписанию.	

По умолчанию, только что добавленное на схему стартовое событие относится к простому типу. Для того чтобы изменить тип события, щелкните по иконке события и в разделе типов выберите тип таймера или сообщения.

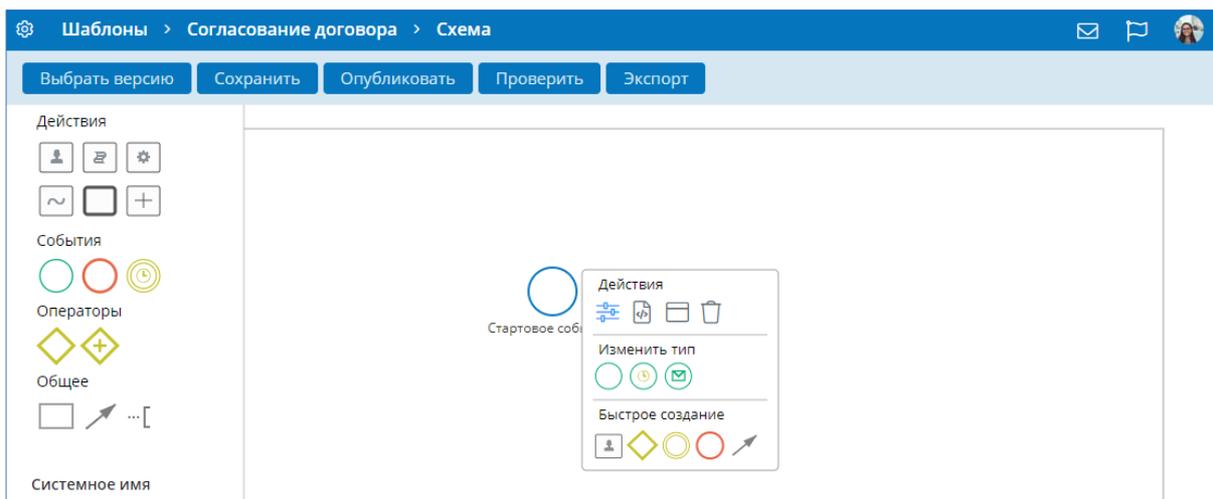


Рис. 215 Настройка стартового события

В разделе **настроек события** задаются основные свойства. Основные свойства события включают в себя системное и отображаемое на схеме имя события, а также её описание.

Событие-сообщение

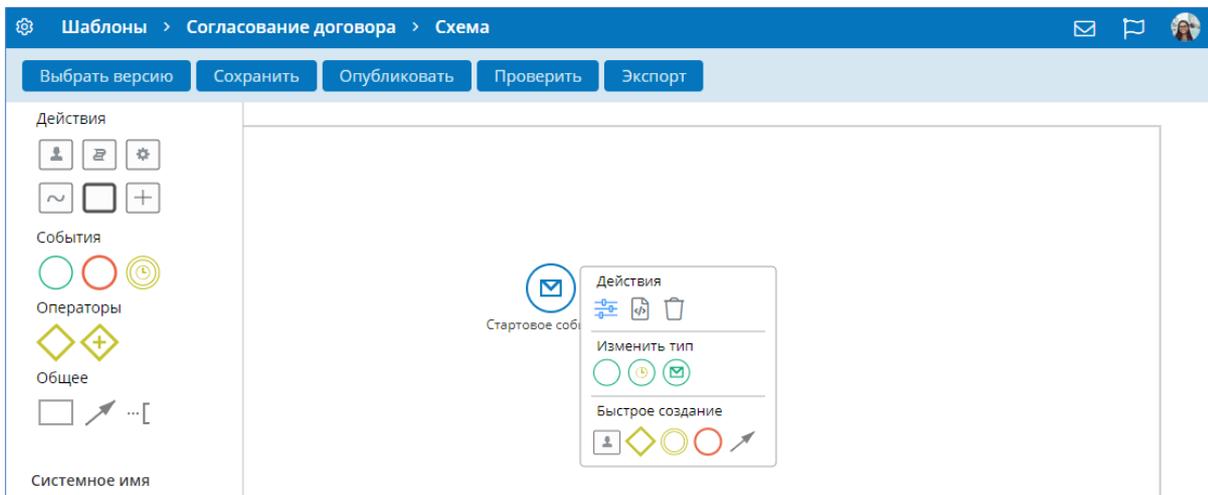


Рис. 216 Событие-получение сообщения

Основные свойства события-сообщения включают в себя настройки сообщения. В поле **Имя сообщения** указывается имя сообщения, отправленного связанным событием "Отправка сообщения" или внешним ресурсом. Событие Отправка сообщения (или путь передачи данных, обеспечивающий взаимодействие процесса с внешним ресурсом) должно быть уже создано и настроено.

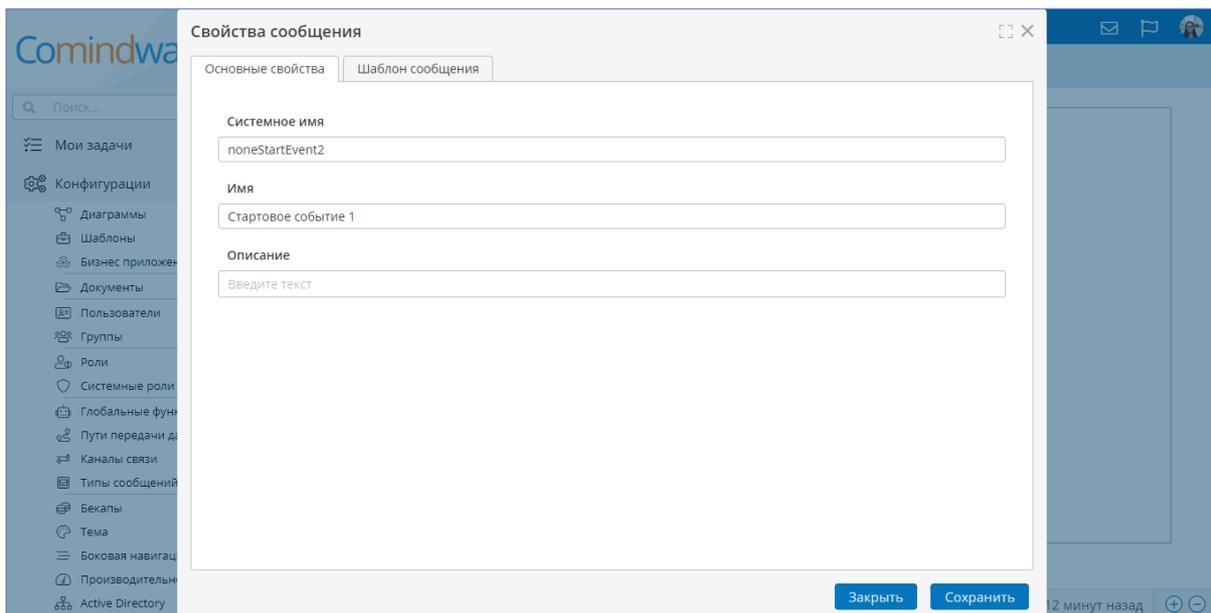


Рис. 217 Настройка события-сообщения

Из раскрывающегося списка **Шаблон сообщения** выбирается необходимый шаблон сообщения (предварительно сконфигурированный в связанном событии "Отправка сообщения" или в связанном пути передачи данных).

Далее настраивается соответствие между атрибутами сообщения и процессными атрибутами.

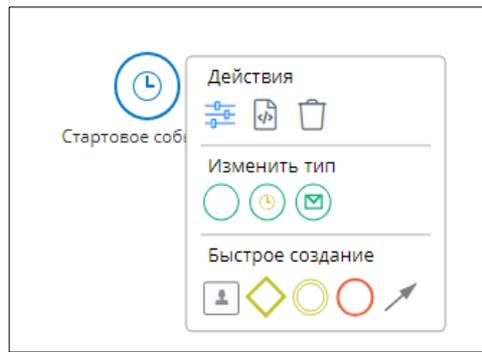


Рис. 218 Событие-таймер

Вкладка «Дополнительные свойства» включает в себя настройки таймера:

- Время старта - Указывается время старта из выпадающего списка или вручную.
- Тип таймера - выбирается интервал срабатывания таймера. В зависимости от выбранного типа доступны следующие настройки:
 - По минутам:
 - Выбор Времени старта;
 - Выбор Времени завершения. Например, каждую 15 минут каждого 8 часа каждого 1 дня.

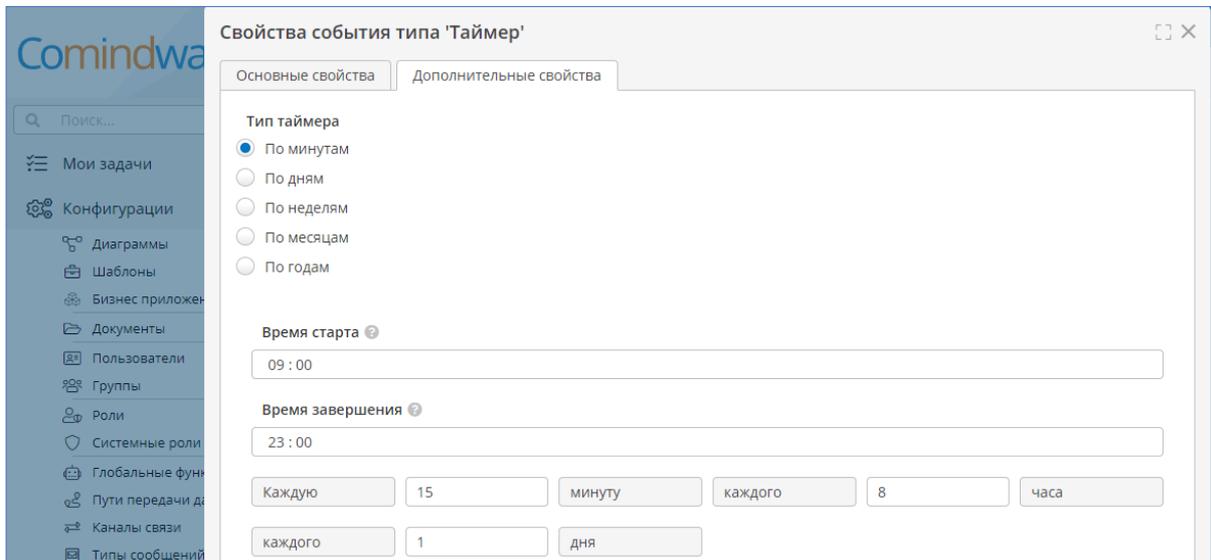


Рис. 219 Тип таймера «по минутам»

- По дням – укажите порядок старта числом (например, каждый 1 – то есть через день) или отметьте галочкой старт «каждый рабочий день».

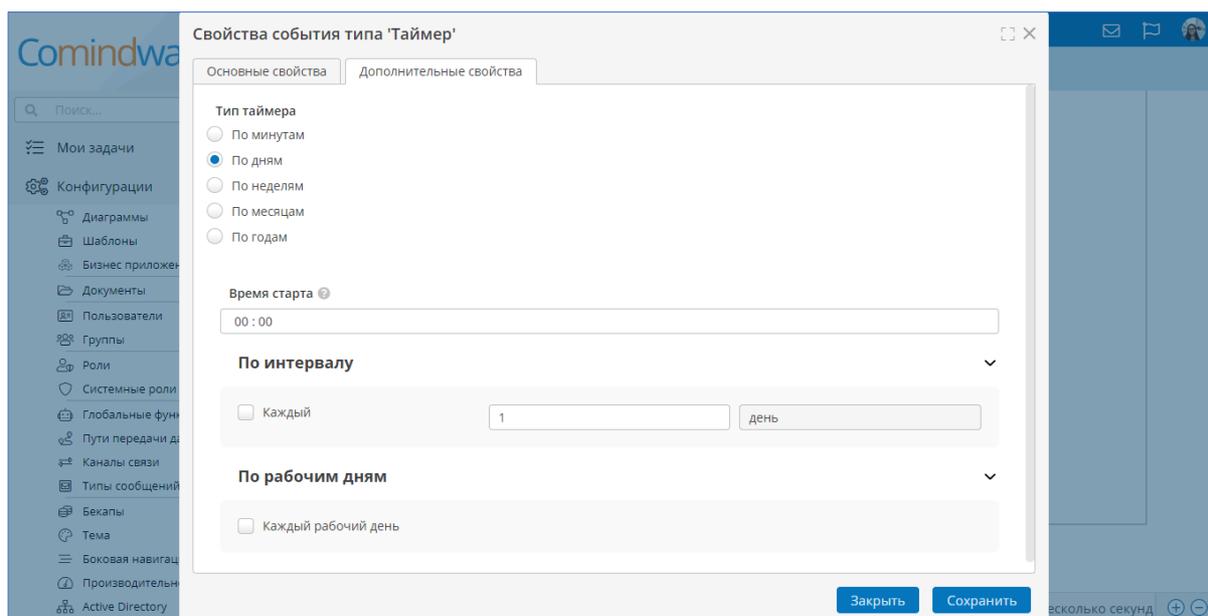


Рис. 220 Тип таймера «по дням»

Рис. 221

- По неделям – укажите время старта, день/дни и номер недели, когда будет запускаться событие.

Время старта ?

00 : 00

Каждую указанную неделю *

1

Понедельн...
 Вторник
 Среда
 Четверг
 Пятница
 Суббота
 Воскресенье

Рис. 222 Тип таймера «по неделям»

- По месяцам – укажите время старта, точное число и периодичность месяцев или укажите порядок старта, выбрав номер недели, день и периодичность месяцев (например, 26– каждый месяц)

Время старта ⓘ

00 : 00

По интервалу ▾

День 26 каждого 1 месяца

В определенный день месяца ▾

Каждый 4 Среда каждого 1 месяца

Рис. 223 Тип таймера «по месяцам»

- По годам – укажите точную дату или выберите периодичность повторения (например, каждый 4 среда декабрь)

Время старта ⓘ

00 : 00

В определенный день ▾

Дата Декабрь 26

В определенный день недели и месяца ▾

 4 Среда месяца Декабрь

Рис. 224 Тип таймера «по годам»

Промежуточное событие

Для того чтобы добавить промежуточное событие в шаблон процесса:

Перетаскиванием добавьте на схему процесса иконку промежуточного события из группы События.

Существует два способа использования Промежуточного события:

- Промежуточное событие, включенное в состав Стандартного потока операций какого-либо Процесса;
- Промежуточное событие, присоединенное к границе Действия.



Рис. 225 Промежуточное событие, включенное в состав Стандартного потока операций какого-либо Процесса

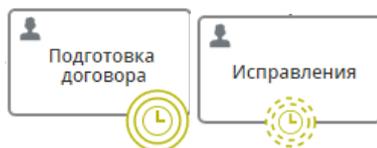


Рис. 226 Промежуточное событие, присоединенное к границе Действия

Промежуточные события имеют следующие типы:

Тип события	Описание	Иконка
Событие-таймер	Приостанавливает процесс или инициирует активность в определенный момент времени.	
Отправка сообщения	Отправляет сообщение, которое передает данные или инициирует активность.	
Получение сообщения	Получает и обрабатывает сообщение.	
Простое промежуточное событие	Не типизированное событие используется для управления временем процесса (в качестве вехи или для обнуления времени жизни процесса)	

По умолчанию, только что добавленное на схему промежуточное событие относится к типу "таймер". Для того чтобы изменить тип события, щелкните по иконке события и выберите тип отправки или получения сообщения, или простого события.

Событие-таймер

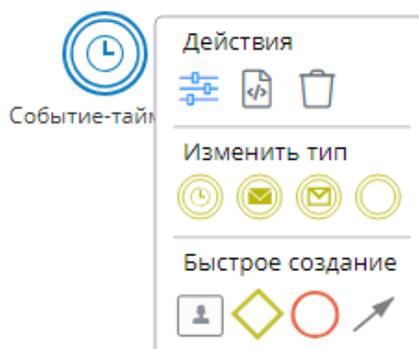


Рис. 227 Настройка промежуточного события

В разделе **настроек события** задаются основные и дополнительные свойства. Основные свойства события включают в себя системное и отображаемое на схеме имя события, а также её описание.

Дополнительные свойства события-таймера включают в себя настройки таймера. Интервал срабатывания таймера задается: напрямую, с помощью значение, языка выражений платформы 223 или C# скрипта.

Если событие присоединено к границе действия выберите, будет ли событие прерывающим выполнение действия или непрерывающим – отметив галочкой соответствующее поле. Граничное непрерывающее событие изображается пунктирными линиями.

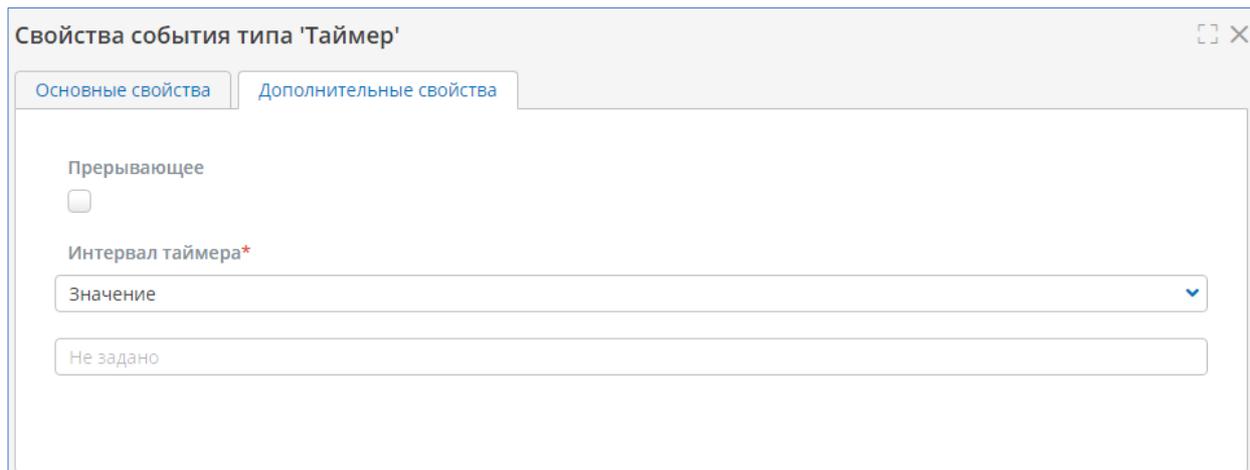


Рис. 228 Настройка событие-таймер

Отправка и получение сообщения

Дополнительные свойства события отправки сообщения включают в себя настройки сообщения. В поле **Имя сообщения** указывается имя отправляемого сообщения. Имейте в виду, что имена отправляемого и получаемого сообщений в связанных событиях должны совпадать.

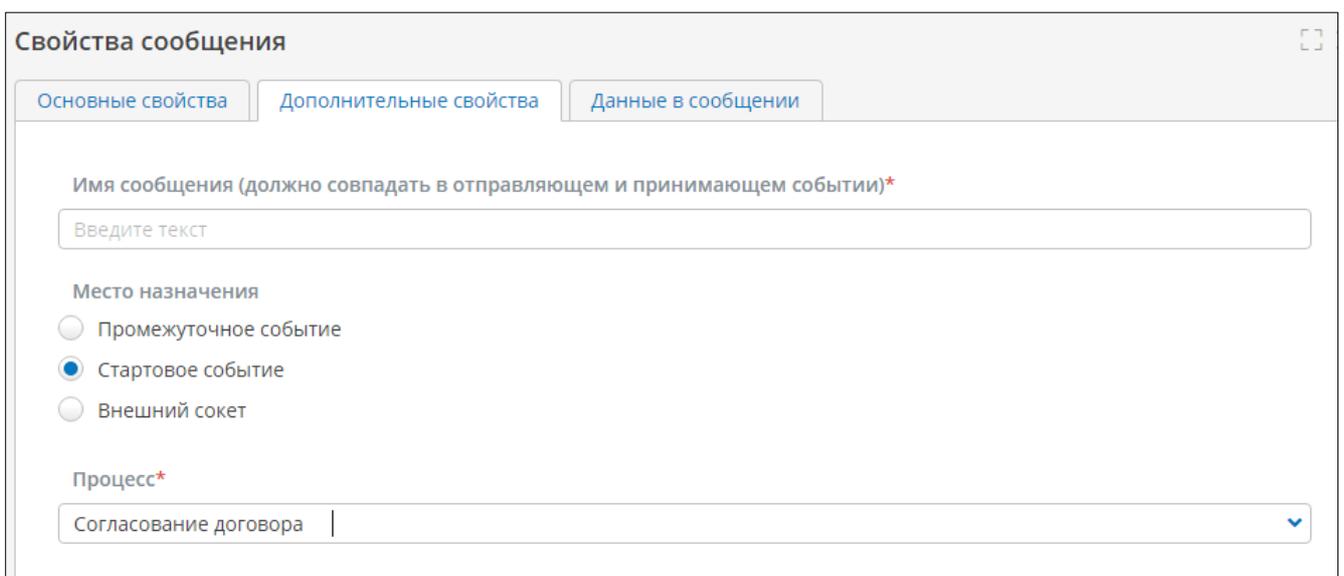


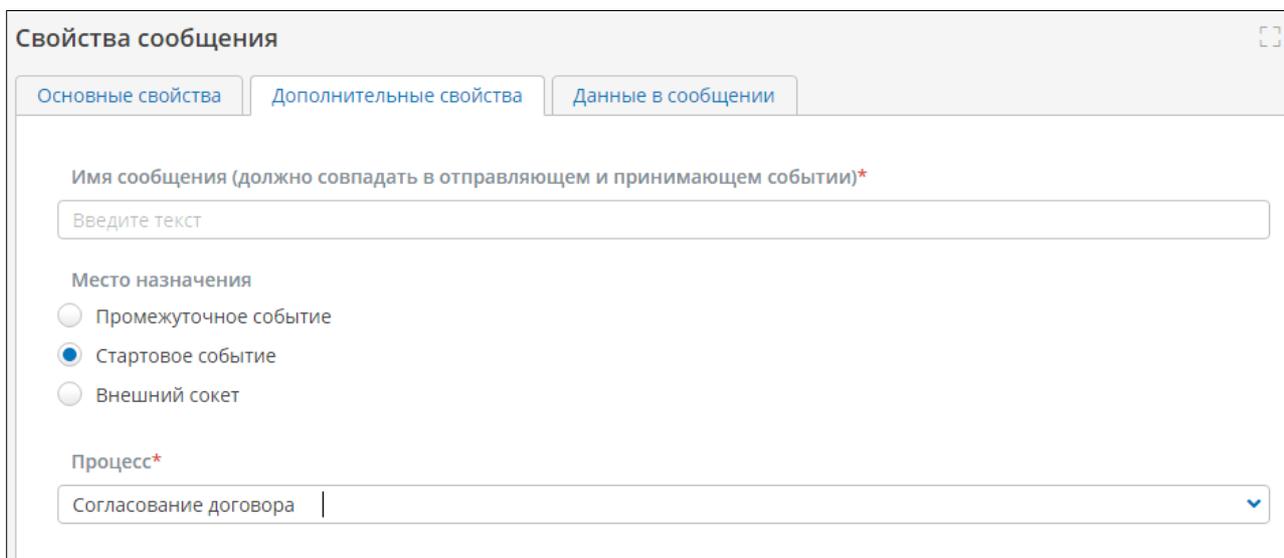
Рис. 229 Дополнительные свойства события отправки сообщения

Место назначения, куда будет отправлено сообщение может быть следующим: промежуточное событие, стартовое событие или внешний ресурс.

Если сообщение адресовано промежуточному событию, указывается экземпляр процесса, который получит сообщение, напрямую, с помощью языка выражений платформы ²²³ или C# скрипта.

Имейте в виду, что при выборе опции **Значение** вам потребуется указать ID активного на данный момент экземпляра процесса.

Если сообщение адресовано стартовому событию, выберите процесс, который будет таким образом запущен.



Свойства сообщения

Основные свойства | **Дополнительные свойства** | Данные в сообщении

Имя сообщения (должно совпадать в отправляющем и принимающем событии)*

Введите текст

Место назначения

Промежуточное событие

Стартовое событие

Внешний сокет

Процесс*

Согласование договора

Рис. 230 Выбор процесса

В разделе **Данные в сообщении** создается или выбирается шаблон сообщения (4.6) и настраивается соответствие между атрибутами сообщения и отправляемыми данными. Системные имена атрибутов в сообщении – имена, которые отображаются в платформе.

Значения атрибутов сообщения задаются: напрямую, с помощью языка выражений платформы ²²³ или C# скрипта, или атрибута. Имейте в виду, что при выборе опции **Атрибут** вам потребуется настроить соответствие между атрибутом сообщения и атрибутом процесса.

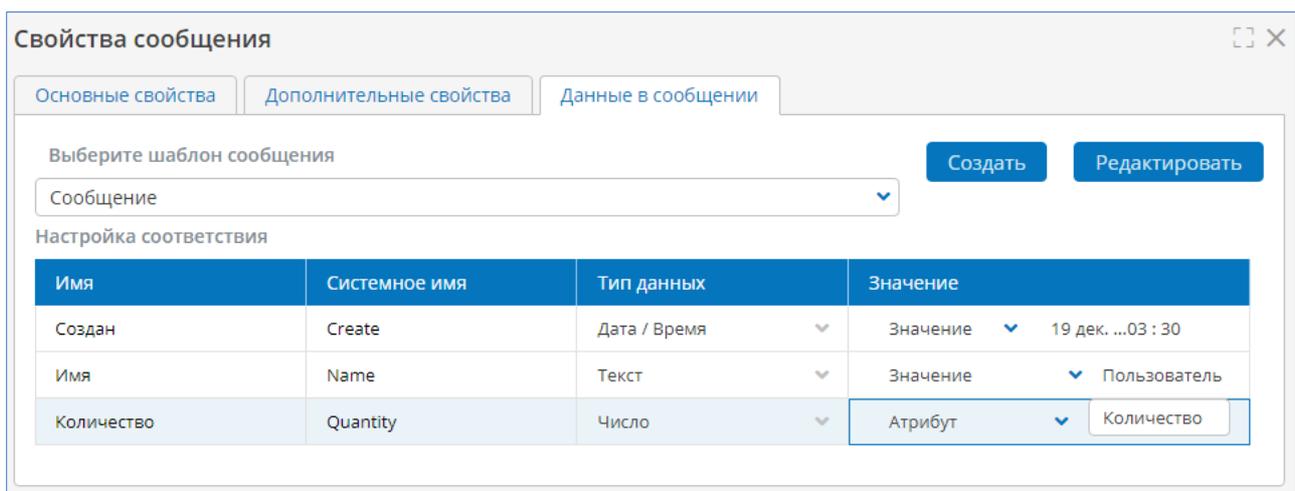
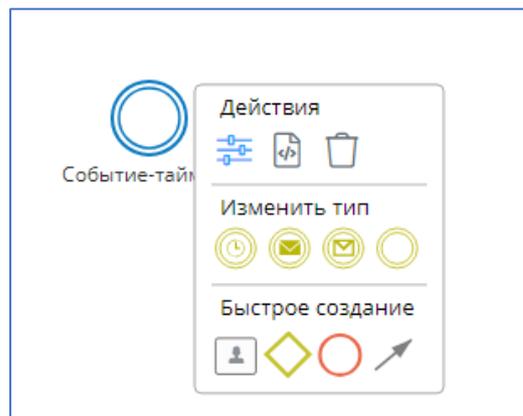


Рис. 231 Настройка атрибутов сообщения

Простое событие



Настройка простого промежуточного события имеет отношение исключительно к времени жизни экземпляра процесса и вехам процесса. Время жизни и вехи играют важную роль при построении отчетов.

В поле **Временная веха** указывается, какой момент времени, считая от начала процесса, является критерием того, что процесс завершен в установленные сроки. Если на данном этапе время жизни экземпляра процесса не превышает время вехи, считается, что процесс будет завершен "в срок". В противном случае, процесс выполняется с "риском задержки".

Временная веха задается напрямую значением, с помощью языка выражений платформы ²²³ или скрипта C#.

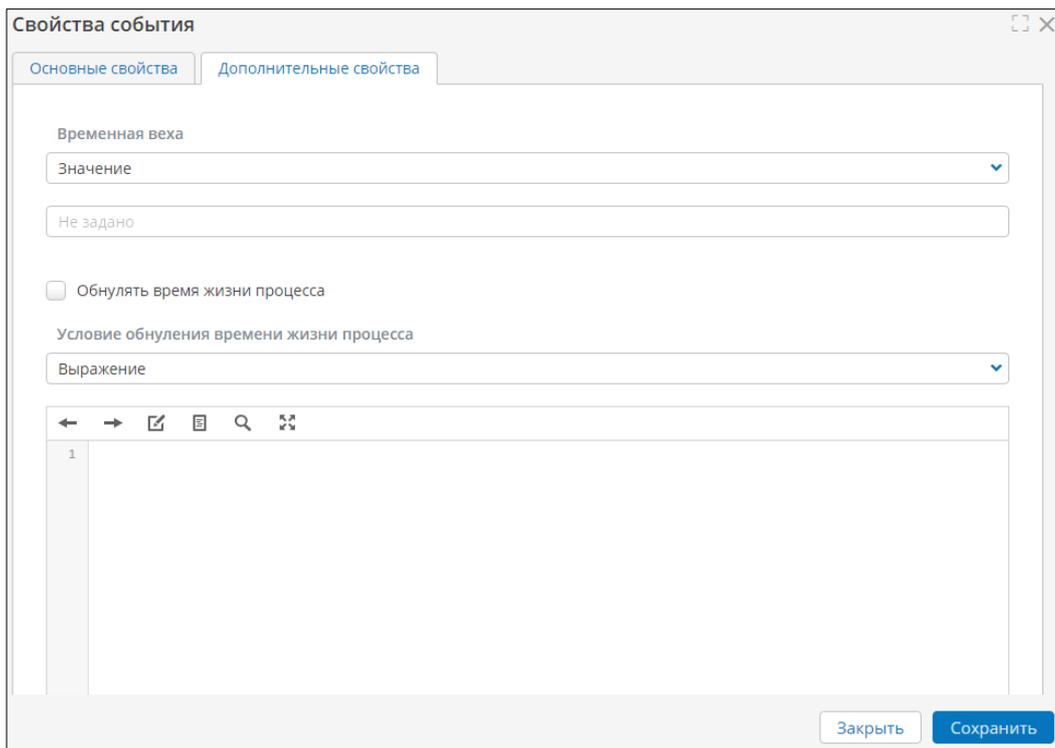


Рис. 232 Дополнительные свойства простого события

Если событие предшествует не запланированной бизнес-активности (например, кейсу), на этапе события может потребоваться обнуление времени жизни процесса. Для этого отметьте **«Обнулить время жизни процесса»**.

Конечное событие

Для того чтобы добавить конечное событие в шаблон процесса:

Перетаскиванием добавьте на схему процесса иконку конечного события из группы События.

Конечное событие имеет следующие типы

Тип события	Описание	Иконка
Простое конечное событие	Указывает на то, что процесс завершен и что с окончанием процесса не связано никакой другой активности.	
Отправка сообщения конечного	В дополнение к завершению процесса производится отправка сообщения.	
Событие-останов	Приводит к немедленному завершению всего бизнес-процесса.	

По умолчанию, только что добавленное на схему конечное событие относится к простому типу. Для того чтобы изменить тип события, щелкните по иконке события и выберите тип сообщения.

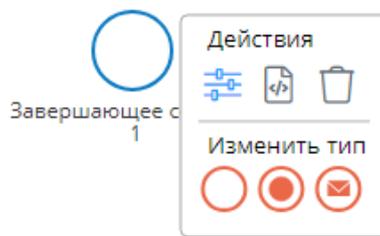


Рис. 233 Типы конечного события

Простое событие

В разделе **настроек события** задаются основные и дополнительные свойства. Основные свойства события включают в себя системное и отображаемое на схеме имя события, а также её описание.

Дополнительные свойства события включают в себя настройку продолжительности выполнения процесса. Продолжительность задается: напрямую, с значением, помощью языка выражений платформы²²³ или C# скрипта.

Событие-сообщение

Дополнительные свойства события-сообщения включают в себя настройки сообщения. В поле **Имя сообщения** указывается имя отправляемого сообщения. Имейте в виду, что имена отправляемого и получаемого сообщений в связанных событиях должны совпадать.

В области место назначения указывается, куда будет отправлено сообщение: промежуточное событие, стартовое событие или внешний ресурс.

Если сообщение адресовано промежуточному событию, укажите экземпляр процесса, который получит сообщение и продолжительность выполнения процесса.

Имейте в виду, что при выборе опции **Значение** вам потребуется указать ID активного на данный момент экземпляра процесса.

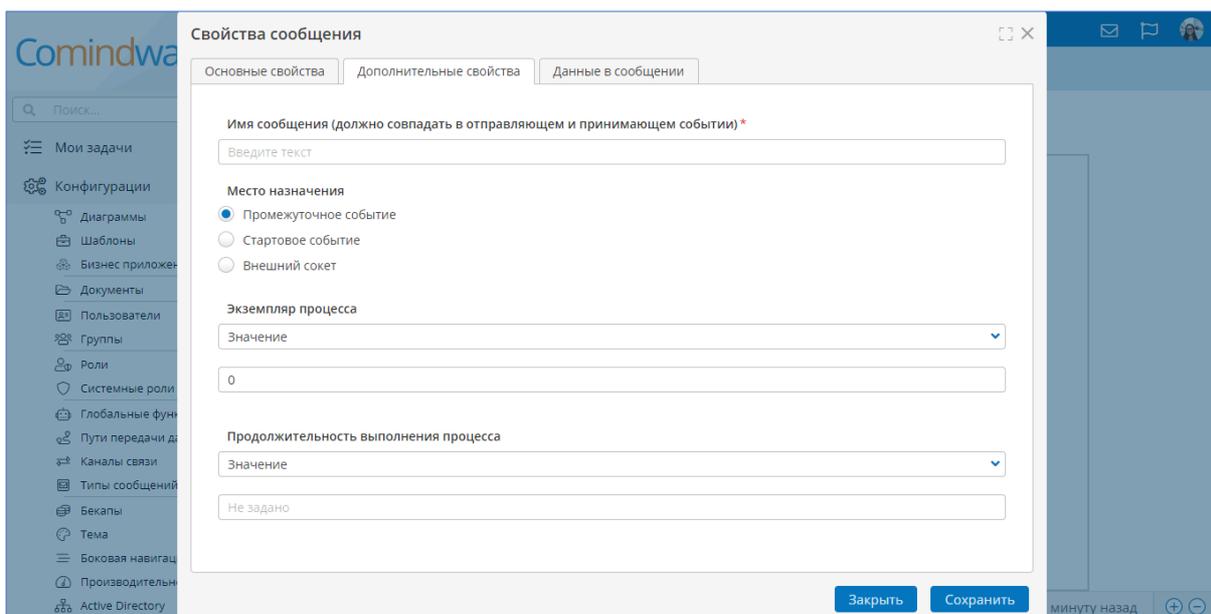


Рис. 234 Конечное событие сообщения

Если сообщение адресовано **стартовому событию**, выберите процесс, который будет таким образом запущен и его продолжительность.

Продолжительность процесса будет учтена при построении отчета. В качестве критерия запаздывания процесса. Если время жизни экземпляра процесса превышает максимальную длительность, считается, что процесс выполняется "с опозданием". В противном случае, процесс выполняется "в срок" либо с "риском задержки".

В разделе **данные в сообщении** создается или выбирается шаблон сообщения и настраивается соответствие между атрибутами сообщения и отправляемыми данными. Системные имена атрибутов в сообщении – имена, которые отображаются в платформе.

Значения атрибутов сообщения задаются: напрямую, с помощью языка выражений платформы²²³ или C# скрипта, или атрибута. Имейте в виду, что при выборе опции **Атрибут** вам потребуется настроить соответствие между атрибутом сообщения и атрибутом процесса.

Свойства сообщения

Основные свойства | **Дополнительные свойства** | Данные в сообщении

Выберите шаблон сообщения

Сообщение

Создать | Редактировать

Настройка соответствия

Имя	Системное имя	Тип данных	Значение
Создан	Create	Дата / Время	Значение 19 дек. ...03 : 30
Имя	Name	Текст	Значение Пользователь
Количество	Quantity	Число	Атрибут Количество

Рис. 235 Настройка атрибутов сообщения

3.3.3 Подпроцессы

В платформе используются подпроцессы двух типов:

Тип подпроцесса	Описание
Встроенный	Представляет собой последовательность активностей, которая может быть выделена в отдельный процесс внутри текущего процесса. Используется для упрощения и структуризации основной процессной схемы.
Внешний	Представляет собой вызов стороннего процесса внутри текущего. Настройка внешнего подпроцесса имеет смысл, если определенная последовательность действий входит сразу в несколько процессов.

Встроенный подпроцесс

Встроенный подпроцесс представляет собой последовательность активностей, которая может быть выделена в отдельный процесс внутри текущего процесса. Используется для упрощения и структуризации основной процессной схемы.

Чтобы добавить вложенный подпроцесс в шаблон процесса, перетаскиванием добавьте на схему процесса иконку встроенного подпроцесса из группы Действия.

В разделе **Действия** расположены настройки подпроцесса, доступ к правилам для данных и схеме.

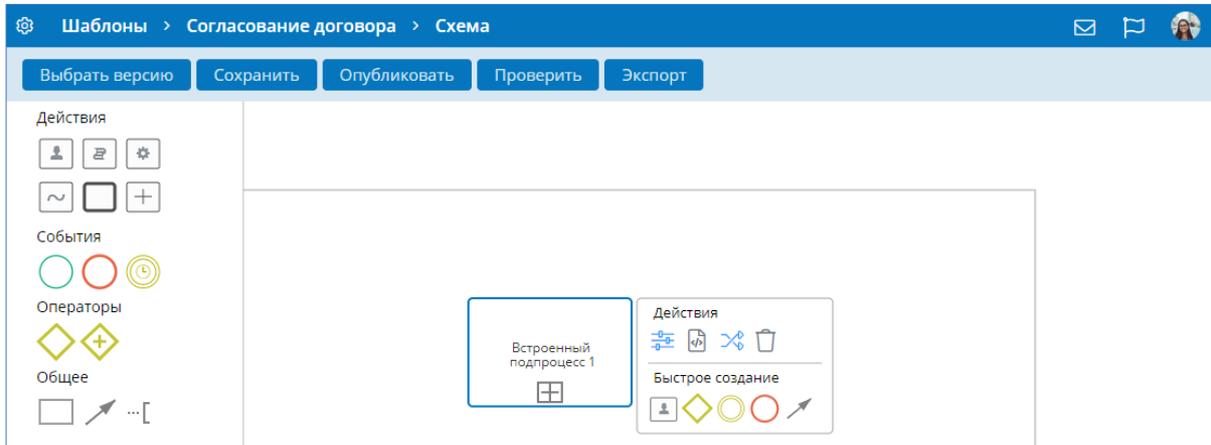


Рис. 236 Встроенный подпроцесс

В разделе настроек подпроцесса задаются основные свойства. Основные свойства события включают в себя системное и отображаемое на схеме имя подпроцесса, а также его описание.

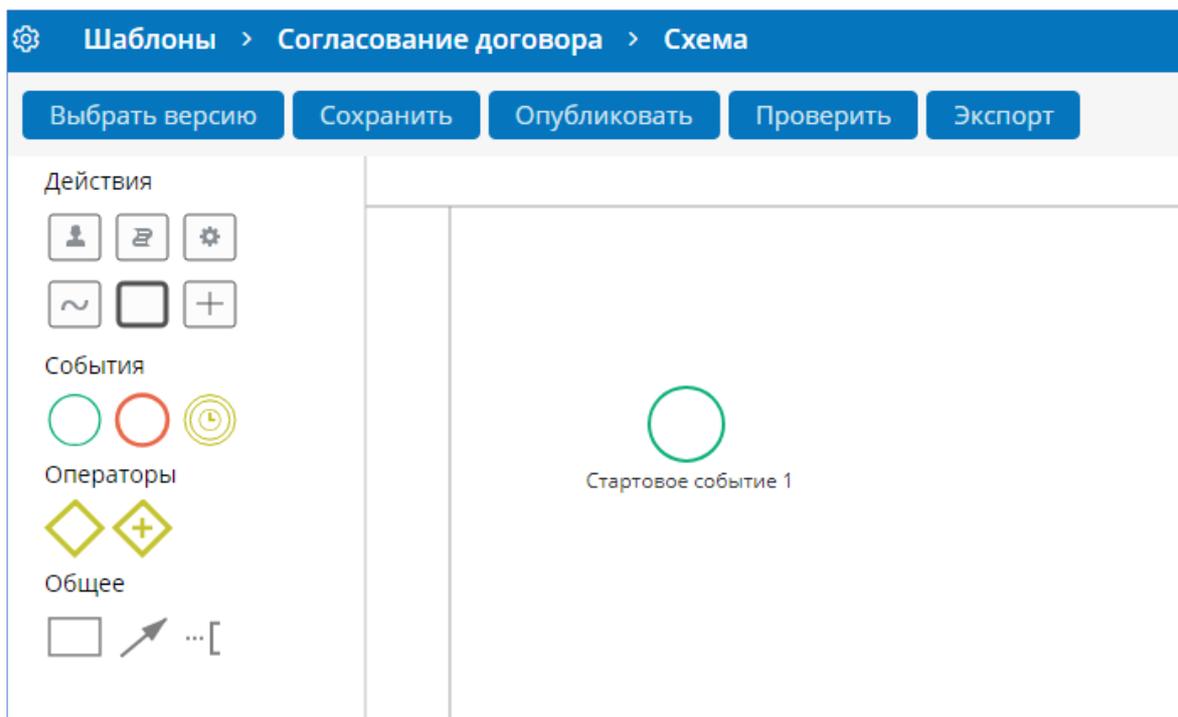


Рис. 237 Схема подпроцесса

Имейте в виду, что в отличие от внешнего подпроцесса, встроенный подпроцесс является частью основного процесса и оперирует тем же набором атрибутов.

Внешний подпроцесс

Внешний подпроцесс представляет собой вызов стороннего процесса внутри текущего. Настройка внешнего подпроцесса имеет смысл, если определенная последовательность действий входит сразу в несколько процессов.

Чтобы добавить внешний подпроцесс в шаблон процесса, перетаскиванием добавьте на схему процесса иконку повторно используемого подпроцесса из группы Действия.

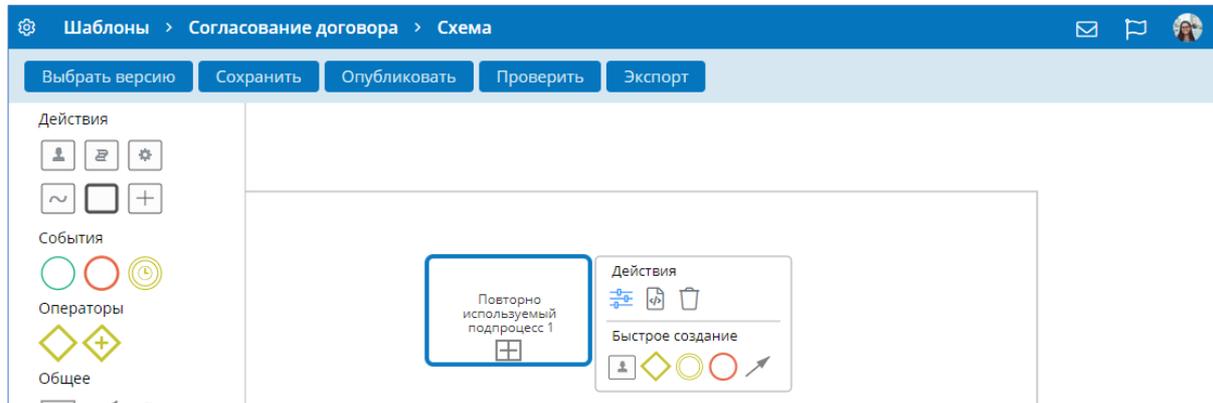


Рис. 238 Настройка повторно используемого подпроцесса

В разделе настроек подпроцесса задаются основные свойства. Основные свойства подпроцесса включают в себя системное и отображаемое на схеме имя подпроцесса, а также его описание.

Дополнительные свойства подпроцесса включают в себя следующие поля:

- Использовать процесс – из раскрывающегося списка необходимо выбрать процесс, вызываемый в качестве подпроцесса.
- Использовать схему процесса – версия процессной схемы.
- Тип цикла – если при выполнении основного процесса подпроцесс должен быть выполнен несколько раз, необходимо выбрать тип повтора: параллельный или последовательный цикл.

Далее необходимо выбрать, выполняется ли цикл определенное количество итераций или по коллекции. Для последовательного цикла, выполняемого определенное количество итераций, вы можете задать:

- либо количество итераций (напрямую, с помощью значений, языка выражений платформы²²³ или C# скрипта),
- либо условие выполнения цикла (с помощью языка выражений платформы²²³ или C# скрипта),
- либо и то, и другое.

В последнем случае число итераций имеет приоритет над условием выполнения цикла. При необходимости отметьте флаг **Проверка условия перед итерацией**.

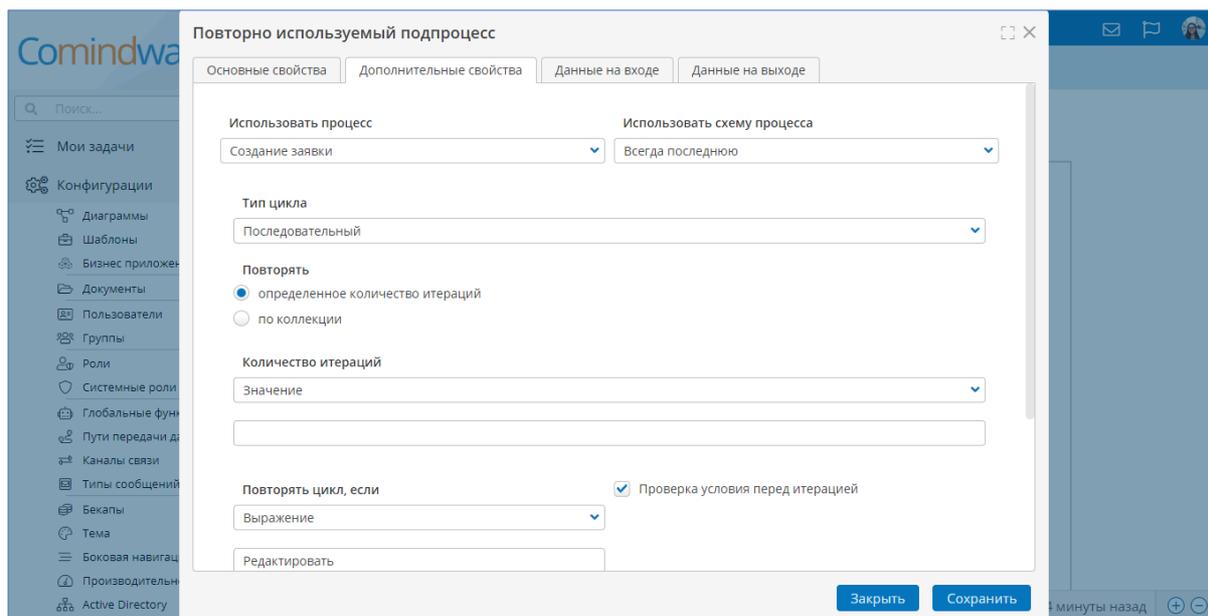


Рис. 239 Дополнительные свойства повторно используемого подпроцесса

Для цикла, выполняемого **по коллекции**, вы можете:

- либо указать коллекцию основного процесса (опция **Атрибут**)
- либо создать новую коллекцию "на лету" (опция "**Выражение**" или "**С#**").

В случае если коллекция создается "на лету", укажите атрибут основного процесса, который будет использован в качестве источника данных для создаваемой коллекции (поле **передавать данные из атрибута родительского процесса**).

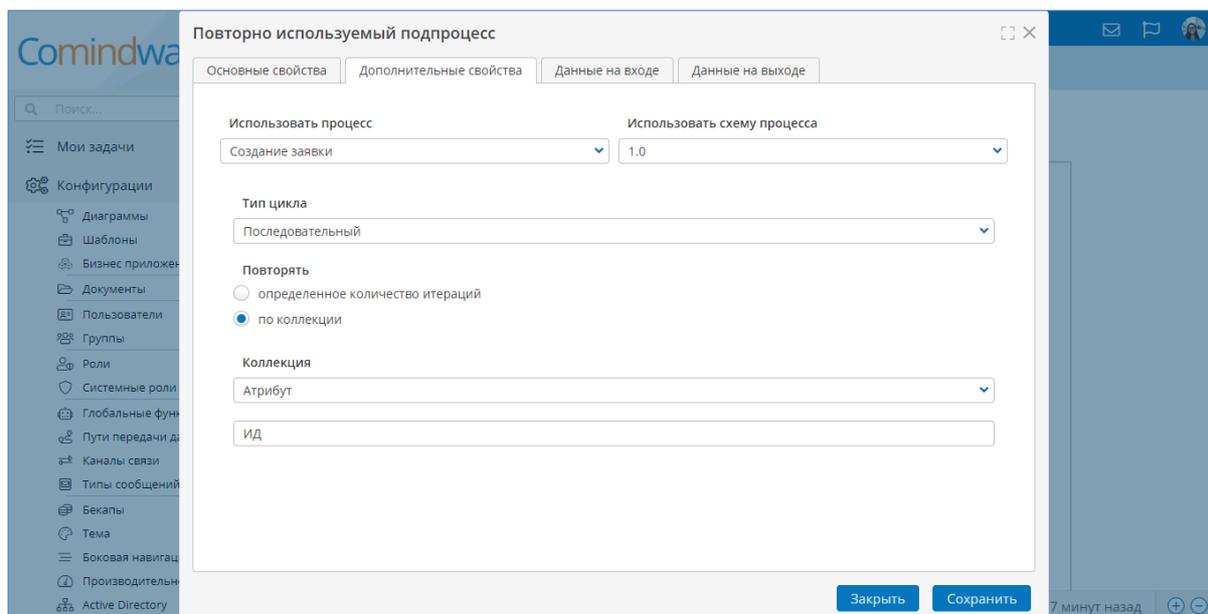


Рис. 240 Дополнительные свойства повторно используемого подпроцесса

Подпроцесс будет создавать записи, по одной на каждую итерацию цикла. При необходимости передайте ссылки на эти записи в коллекцию. Эта коллекция должна быть предварительно создана в связанном с основным процессом шаблоне записи.

Внешний подпроцесс оперирует собственными атрибутами. В разделе **Данные на входе** указываются их значения внутри основного процесса.

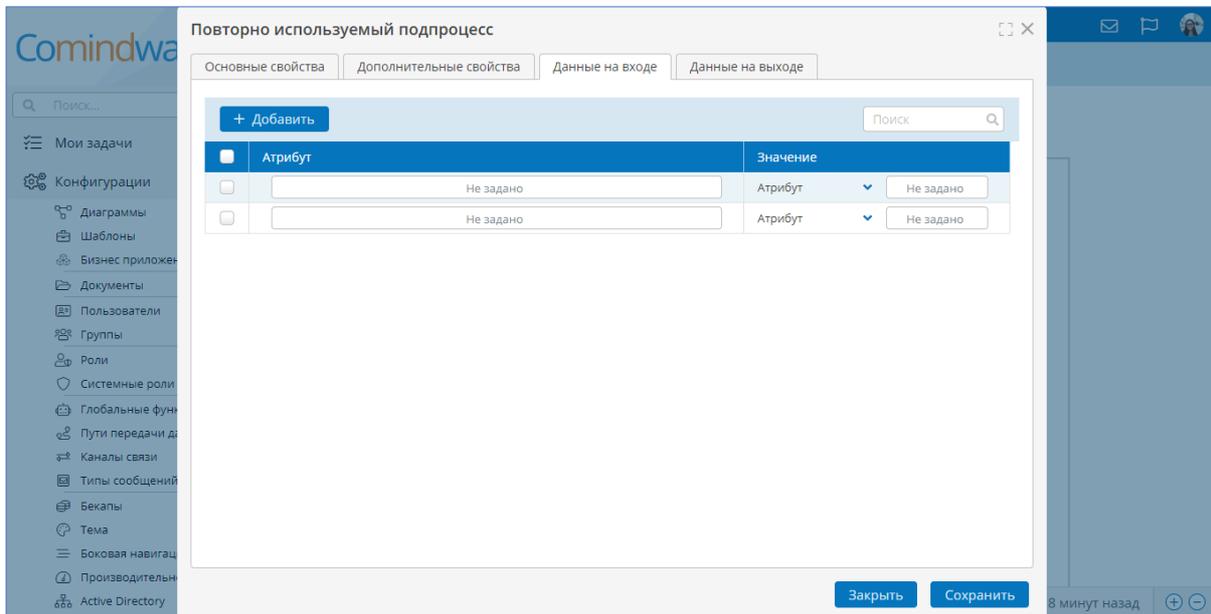


Рис. 241 Свойства повторно используемого подпроцесса

Имейте в виду, что при выборе опции **Атрибут** вам потребуется настроить соответствие между атрибутом подпроцесса и атрибутом основного процесса.

Аналогичным образом укажите, какие данные подпроцесс вернет в основной процесс после завершения.

3.3.4 Шлюзы

Шлюз определяет ветвление и слияние потоков.

В платформе используются шлюзы двух типов:

Шлюз исключаящего «или»

Используется для создания альтернативных потоков (путей) процесса. Только один из путей может быть выбран. На рисунке ниже отображена схема процесса: после выполнения пользовательской задачи дальнейшее выполнение процесса продолжится только по одному потоку:

- Если верно условие 1, то продолжится выполнение задачи 2
- Если верно условие 2, то продолжится выполнение задачи 3
- Если неверно ни одно условие, то продолжится выполнение задачи 1 (поток по умолчанию)

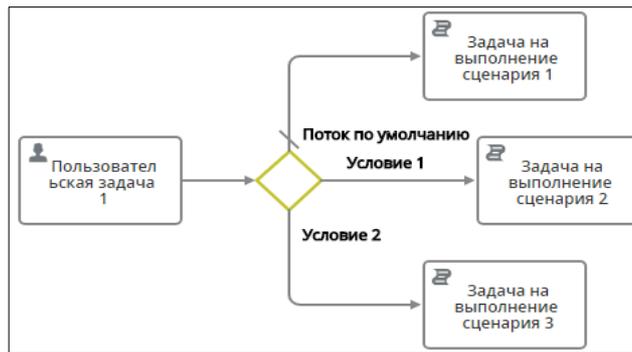


Рис. 242 Шлюз исключаящего «или»

Используется для сходящихся потоков управления. В данном случае шлюз пропускает все потоки через себя без синхронизации. Таким образом задача на выполнение сценария будет выполнена два раза: после выполнения пользовательской задачи 1 и после выполнения пользовательской задачи 2.

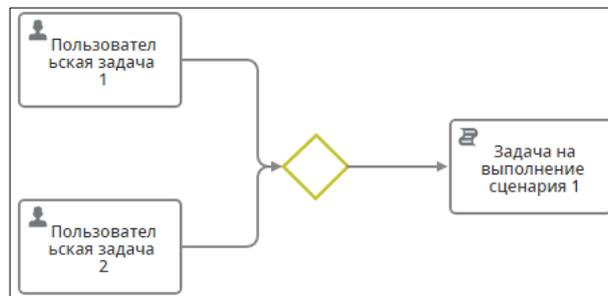


Рис. 243 Шлюз исключаящего «или»

Параллельный

Используется для создания параллельных путей без оценки какого бы то ни было условия. Таким образом после выполнения пользовательской задачи запустится выполнение кейса 1 и кейса 2.

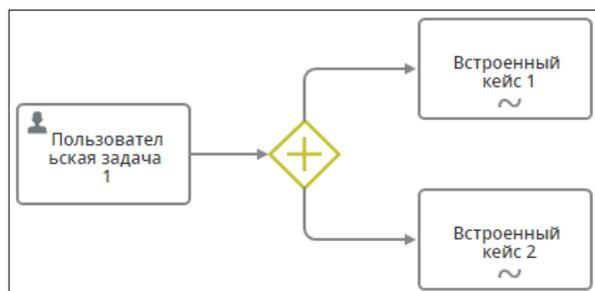


Рис. 244 Параллельный шлюз исключаящего «или»

Используется для сходящихся потоков и синхронизации параллельных веток выполнения процесса. Выполнение кейса начнется только тогда, когда будут выполнены пользовательские задачи 1 и 2.

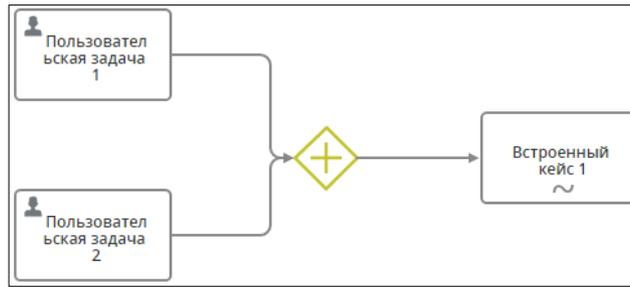


Рис. 245 Параллельный шлюз исключаяющего «или»

Для того чтобы добавить шлюз в шаблон процесса:

Перетаскиванием добавьте на поле иконку шлюза исключаяющего "или" или иконку параллельного шлюза из группы Операторы.

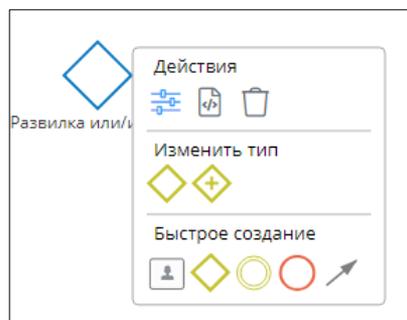


Рис. 246 Настройка наименования шлюза

В разделе настроек шлюза задаются основные свойства. Основные свойства шлюза включают в себя системное и отображаемое на схеме имя шлюза, а также его описание.

Дополнительные свойства шлюза включают в себя настройки исходящих потоков.

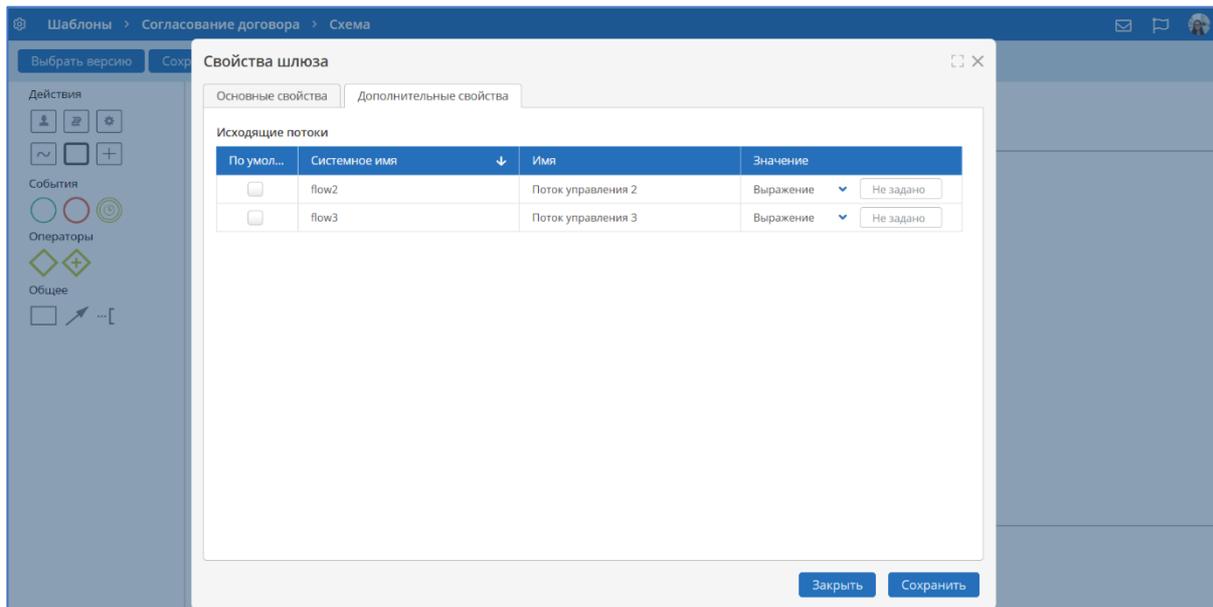


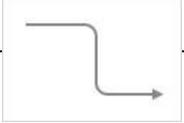
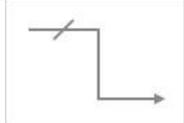
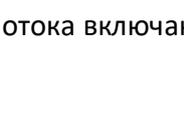
Рис. 247 Настройка исходящих потоков шлюза

3.3.5 Другие элементы

Поток

Поток представляет собой переход между двумя последовательными этапами (активностями) процесса.

В платформе используются потоки следующих типов:

Тип потока	Описание	Иконка	Изображение потока на схеме
Поток управления	Соединяет этапы (активности) процесса.		
Условный	Вызывается при определенном условии. В противном случае вызывается поток по умолчанию. Может исходить только из шлюза, исключаяющего «или».		
Поток по умолчанию	Вызывается, если условие, заданное в условном потоке, не выполнилось. Может исходить только из шлюза, исключаяющего «или».		

Перетаскиванием добавьте на схему процесса иконку потока из группы Общее.

В разделе «**настроек потока**» задаются основные свойства. Основные свойства потока включают в себя системное и отображаемое на схеме имя потока, а также его описание.



Рис. 248 Настройка потока

Дорожка

Дорожка используется для организации и классификации активностей внутри пула в соответствии с их функциями и ролями пользователей в платформе.

Перетаскиванием добавьте на схему процесса иконку дорожки из группы Общее. Добавьте на дорожку активности.

Текстовая ассоциация



Текстовая ассоциация используется, когда нужно привязать текст к тому или иному элементу процессной схемы (например, к событию отправки сообщения).

Перетаскиванием добавьте на схему процесса иконку ассоциации из группы Общее.

В разделе настроек ассоциации задаются основные свойства. Основные свойства включают в себя системное и отображаемое на схеме имя ассоциации, а также его описание.

3.4 Настройка бизнес-правил

Вы можете настроить автоматическое выполнение тех или иных действий до начала или после завершения этапа процесса. Для каждого этапа процесса можно создать неограниченное количество правил.

Для того чтобы создать правило необходимо в разделе «**Действия**» открыть «**Правила для данных**».

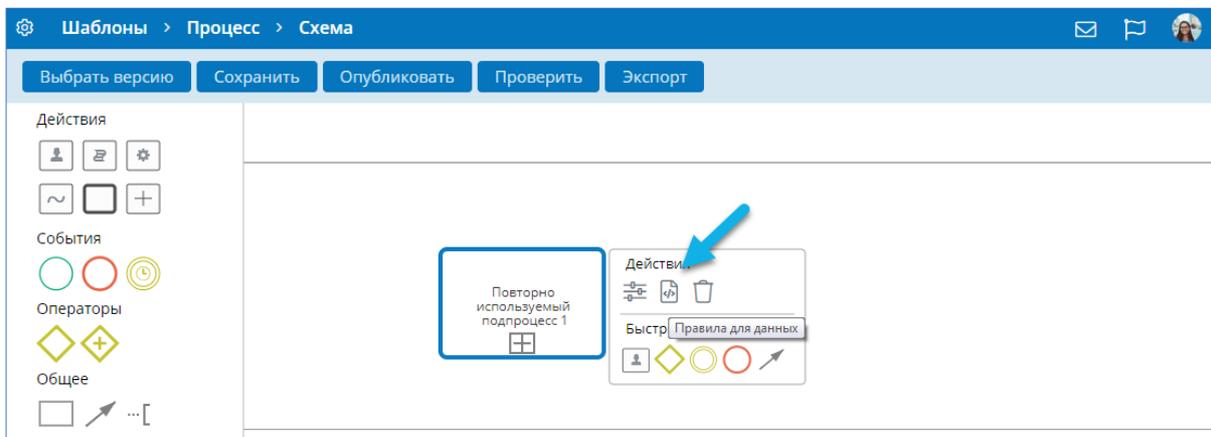


Рис. 249 Настройка правил для данных

Правила на входе и выходе из активности настраиваются одинаково. Для создания правила перетащите элемент **Изменить значение** или **Проверить значение** в область правила.

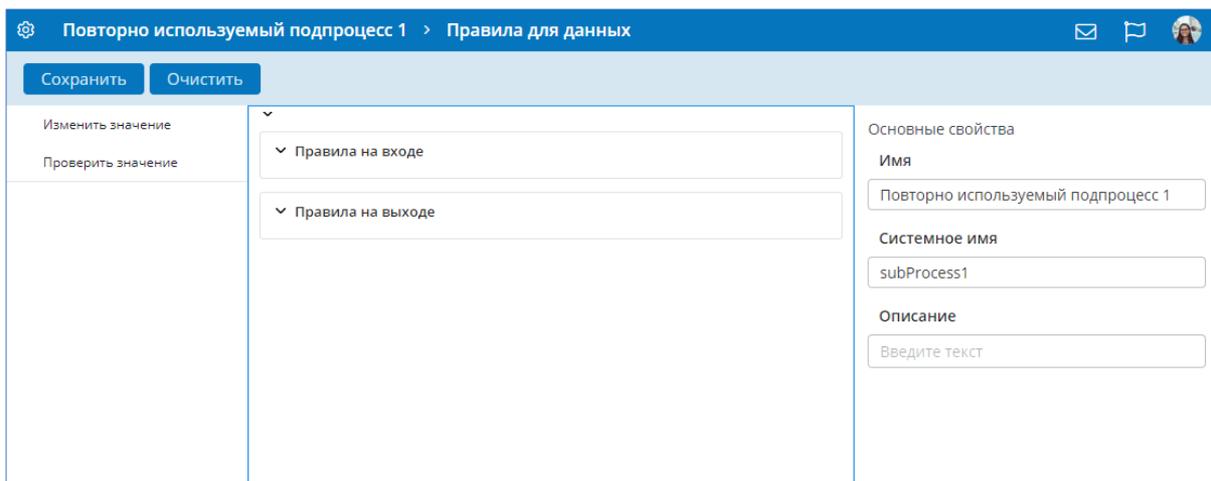


Рис. 250 Редактор правила для данных

- Если выбрано правило изменения, укажите изменяемый атрибут. Затем укажите способ изменения атрибута. Вы можете изменить атрибут напрямую, используя значение, язык

выражений платформы ²²³ или C# скрипт, а также подставив значение другого атрибута. Данное правило позволяет на входе/выходе задачи или события изменить параметры.

- Если выбрано правило проверки, укажите проверяемые параметры и критерии проверки. Данное правило позволяет приостановить процесс на входе/выходе задачи или события для проверки параметров. Введите сообщение, которое будет показано пользователю, если критерии проверки не были удовлетворены. Условие задается: напрямую значением, используя язык выражений платформы ²²³, C# скрипт, а также выбрав значение другого атрибута. Если условие возвращает Истина процесс останавливается с выводом ошибки. Если условие возвращает Ложь – процесс не останавливается.

Рис. 251 Настройка правила

3.5 Проверка и публикация схемы процесса

Как только схема процесса будет готова, нажмите «Проверить», чтобы проверить правильность построения схемы. Если проверка прошла успешно, нажмите «Опубликовать». Процессные схемы рекомендуется публиковать каждый раз при внесении изменений. Для экспорта схемы процесса в формате .svg нажмите «Экспорт».

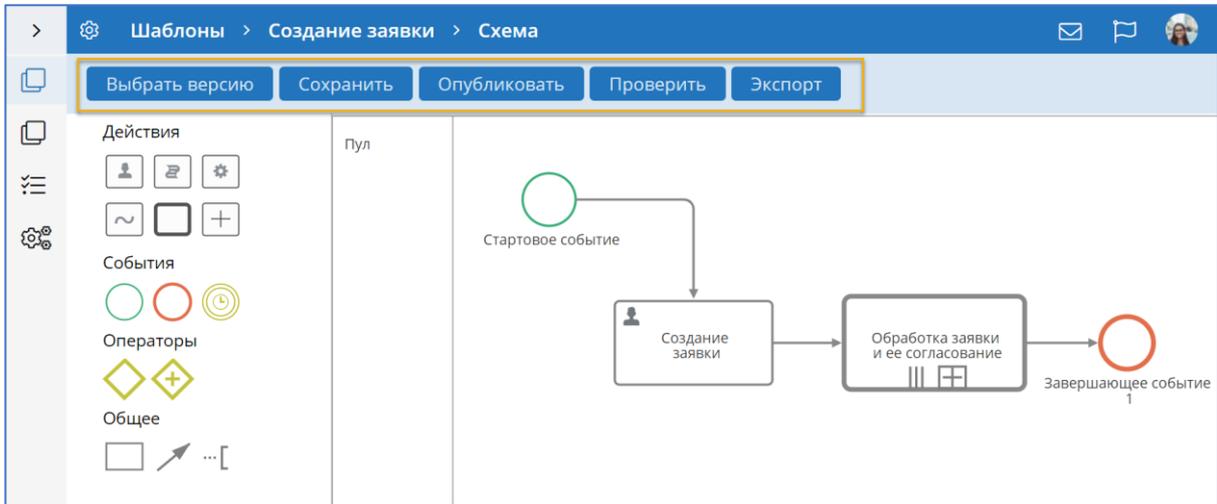


Рис. 252 Панель инструментов

При возникновении ошибок элемент, в котором присутствует ошибка, подсвечивается красным, и на верхней панели навигации отображается флаг с количеством ошибок, при нажатии на который открывается окно с описанием ошибок.

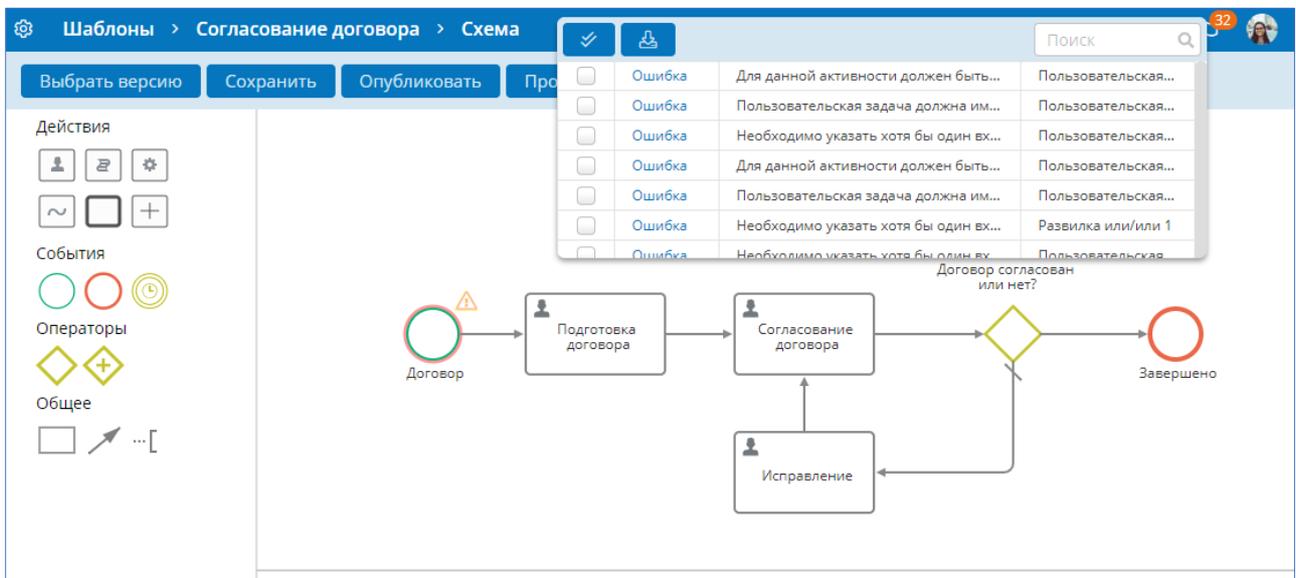


Рис. 253 Схема процесса

3.5.1 Типовые ошибки диаграммы процесса

При работе с диаграммой возникает ряд ошибок, которые не позволяют корректно опубликовать схему и продолжить работать с ней. В данном разделе описаны типовые ошибки. Они приведены в таблице с описанием ошибки и способом ее устранения.

Ошибка	Элементы процессной схемы	Описание	Способ устранения
Схема пуста	Схема	Схема не содержит ни одного объекта	Добавить на схему объекты диаграммы
Схема должна включать хотя бы одно стартовое событие	Схема	На схеме отсутствует стартовое событие	Добавить на схему стартовое событие
Для данной активности должен быть задан один исходящий поток	Стартовое событие	У стартового события отсутствует исходящий поток	Добавить исходящий поток стартовому событию
На схеме не может быть более одного простого (не типизированного) стартового события	Стартовое событие	На схеме находится более одного стартового события	Удаление всех стартовых событий кроме одного либо типизирование событий
Требуется хотя бы один входящий поток	Завершающее событие	У завершающего события отсутствует входящий поток	Добавить входящий поток завершающему событию
Требуется хотя бы один входящий поток	Пользовательская задача	У пользовательской задачи отсутствует входящий поток	Добавить входящий поток пользовательской задаче
Для данной активности должен быть задан один исходящий поток	Пользовательская задача	У пользовательской задачи отсутствует исходящий поток	Добавить исходящий поток пользовательской задаче
Активность не имеет потоков управления или отсутствует возможность перейти из стартового события к данной активности	Пользовательская задача	У пользовательской задачи отсутствует входящий и исходящий потоки	Добавить исходящий и входящий поток пользовательской задаче
Пользовательская задача должна иметь	Пользовательская задача	В настройках пользовательской задачи не указан	Зайти в настройки пользовательской задачи. В разделе « Дополнительные

Ошибка	Элементы процессной схемы	Описание	Способ устранения
минимум одного ответственного		исполнитель данной задачи	свойства» задать исполнителя
Выражение не задано	Пользовательская задача	В настройках пользовательской задачи в разделе «Дополнительные свойства» значение поля «Исполнитель» выбрано <i>«Выражение»</i>	Зайти В Настройки Пользовательской Задачи. В Разделе «Дополнительные Свойства» В Поле «Исполнитель» Выбрать <i>«Выражение»</i> И Задать Корректное Выражение
Код сценария отсутствует	Пользовательская задача	В настройках пользовательской задачи в разделе «Дополнительные свойства» значение поля «Исполнитель» выбрано <i>«С# скрипт»</i>	Зайти в настройки пользовательской задачи. В разделе «Дополнительные свойства» в поле «Исполнитель» выбрать <i>«С# скрипт»</i> и задать корректный скрипт
Требуется исходящий поток по умолчанию	Развилка или/или	В настройках развилки или/или не указан исходящий поток по умолчанию	В настройках развилки необходимо указать один поток по умолчанию, в зависимости от пользовательского процесса. Зайти в настройки развилки или/или. В разделе «Дополнительные свойства» указать один из потоков как поток по умолчанию.
Активность не имеет потоков управления или отсутствует возможность переместиться из стартового события к данной активности	Развилка или/или	У развилки или/или отсутствует входящий поток	Добавить входящий поток развилке или/или

Ошибка	Элементы процессной схемы	Описание	Способ устранения
Требуется хотя бы один входящий поток	Развилка или/или	У развилки или/или отсутствует входящий поток	Добавить входящий поток развилке или/или
Требуется хотя бы один исходящий поток	Развилка или/или	У развилки или/или отсутствует исходящий поток	Добавить исходящий поток развилке или/или
Не допустимое определение потока операции	Развилка или/или	Развилка или/или имеет более одного исходящего потока. Один поток является потоком по умолчанию. У потоков не определено условия их выполнения.	<p>1. Зайти в настройки развилки или/или, выбрать меню «Дополнительные свойства». Записать выражение либо скрипт в поле «Значение» потоков (кроме потока по умолчанию).</p> <p>2. Зайти в настройки потока в меню «Дополнительные свойства». Записать выражение либо скрипт (кроме потока по умолчанию).</p>
Счетчик не настроен в событии счетчика	Промежуточное событие	В настройках события-таймер не задан интервал времени	Зайти в настройки события таймер. В раздел «Дополнительные свойства» задать значение интервала времени
Не задано выражение	Промежуточное событие	В настройках события-таймер не задано выражение или C# скрипт	Зайти в настройки события таймер. В раздел «Дополнительные свойства» задать выражение либо скрипт
Требуется хотя бы один входящий поток	Промежуточное событие	У промежуточного события отсутствует входящий поток	Добавить входящий поток

Ошибка	Элементы процессной схемы	Описание	Способ устранения
			промежуточному событию
Для данной активности должен быть задан один исходящий поток	Промежуточное событие	У промежуточного события отсутствует исходящий поток.	Добавить исходящий поток промежуточному событию.
Неверные данные события-таймера	Событие таймер	В настройках события таймер не указан интервал таймера.	Выполнить настройку события таймер в соответствии с разделом «Промежуточные события» .

3.6 Управление версиями схемы процесса

Новая версия схемы процесса создается следующим образом:

- Автоматически при публикации схемы.
- Вручную, когда вы собираете «Сохранить текущую схему как новую версию» из раскрывающегося списка .

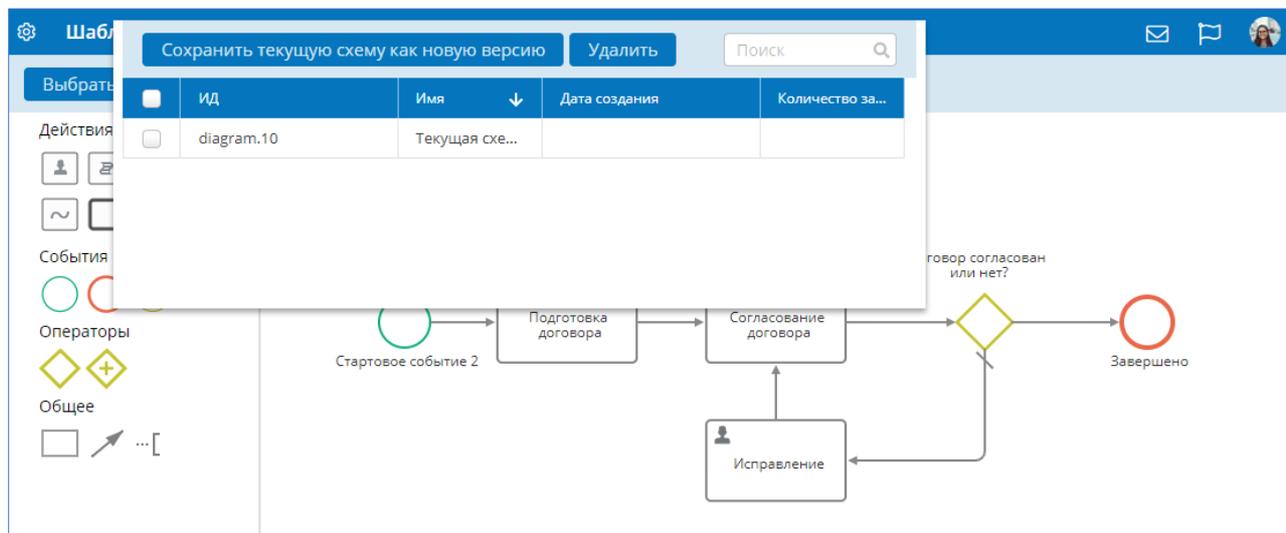


Рис. 254 Сохранение схемы

Все предыдущие версии схемы процесса доступны из раскрывающегося списка “Выбрать версию”.

3.7 Настройка формы процесса

Некоторые процессы начинаются с ввода данных в форму. Например, первым этапом исправления ошибки в программе является ввод ее описания в платформы отслеживания ошибок.

Форма бывает двух типов: стартовая форма события, пользовательская форма.

Для настройки формы задачи:

1. Кликните по элементу (событию/процессу) на процессной схеме и из выпадающего списка выберите **Стартовая форма**, если это стартовое событие, или **Форма**, если это пользовательская задача.
2. Существует несколько способов конструирования формы задачи:
 - Настройка формы задачи по типу формы шаблона. ³⁶
 - Добавление формы связанной записи

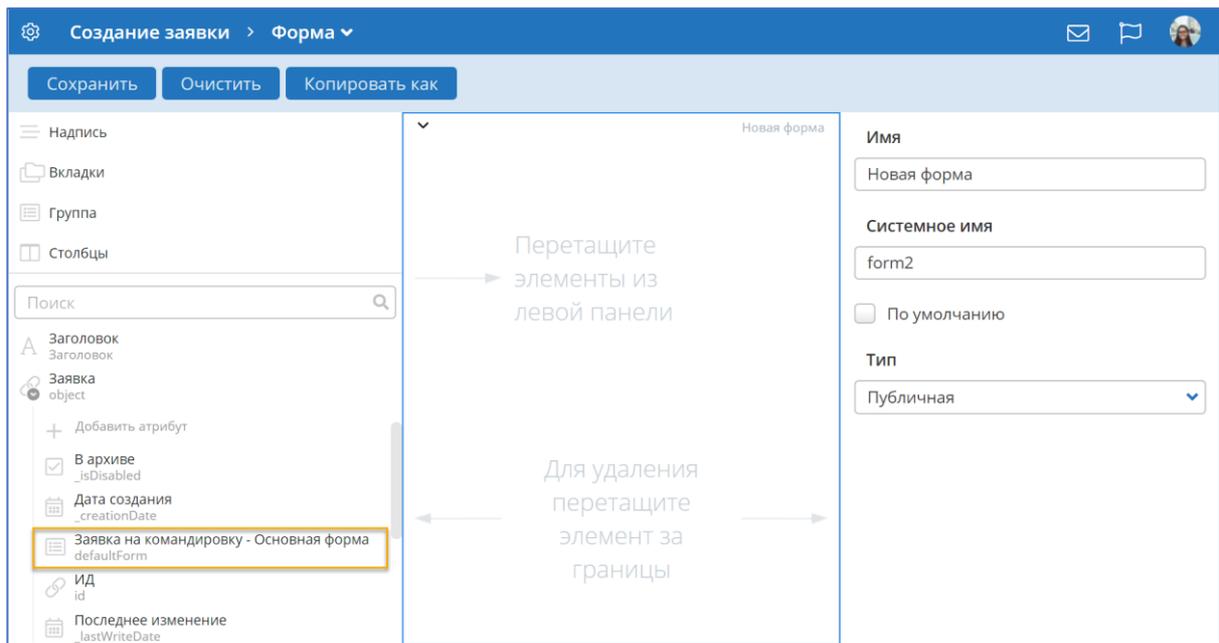


Рис. 255 Настройка формы записи

Добавленная форма записи недоступна для редактирования. Для того чтобы изменить поля, необходимо обратиться к кнопке «Редактировать форму» в правой панели.

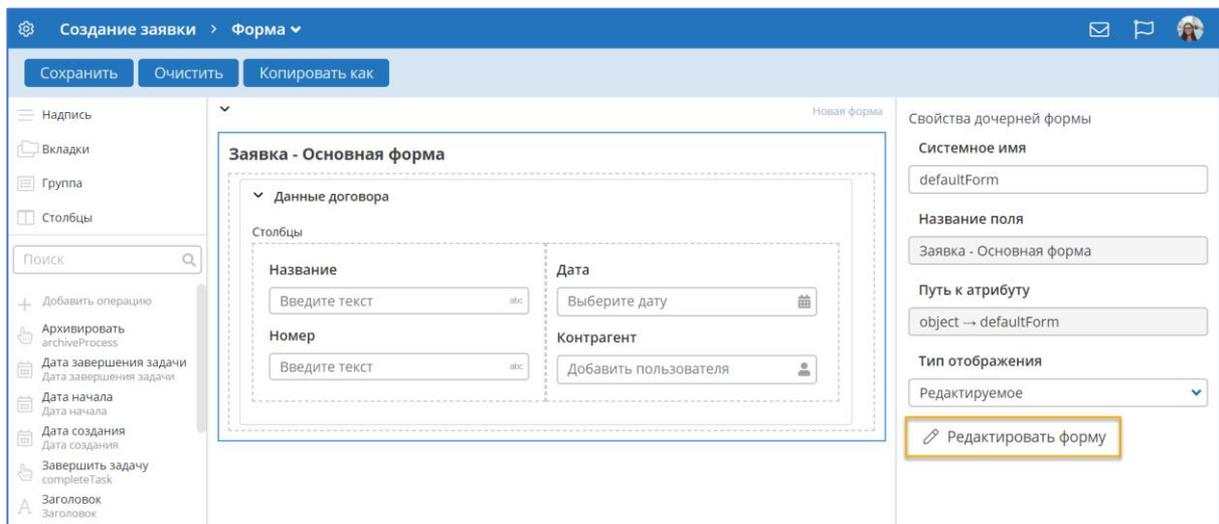


Рис. 256 Свойства дочерней формы

Откроются настройки записи, в которых изменяется или создается форма связанной записи.

- Копирование формы другой задачи

Скопировать форму задачи возможно только, если на процессной схеме, помимо стартового события, существуют пользовательские задачи. Для этого необходимо нажать на соответствующую кнопку.

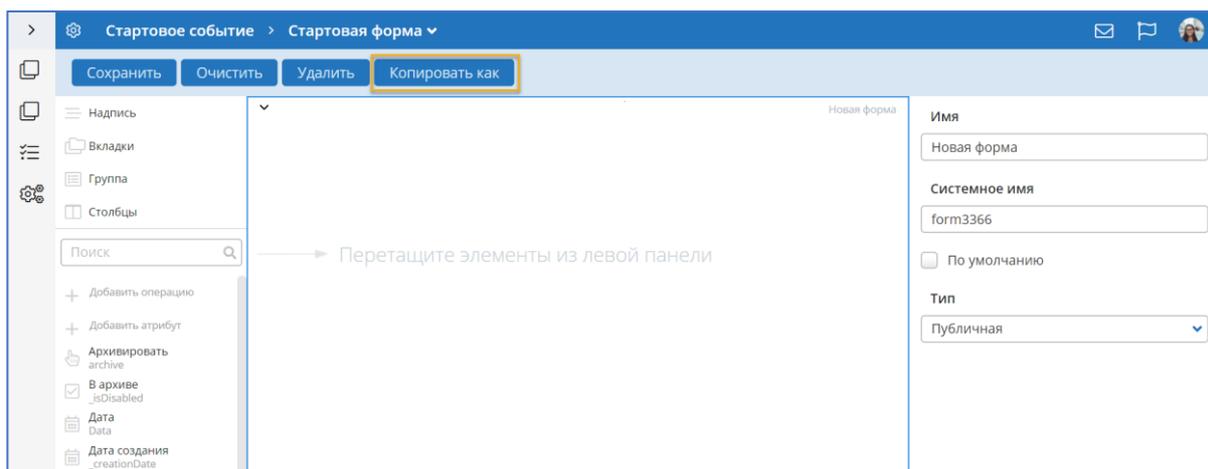


Рис. 257 Кнопка «Копировать как»

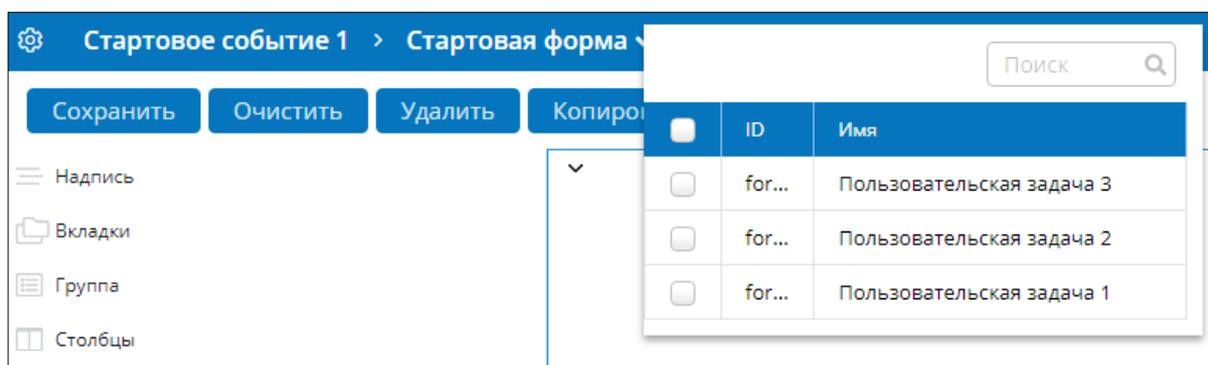


Рис. 258 Копирование формы задачи

3.8 Межпроцессное взаимодействие

Для того чтобы настроить межпроцессное взаимодействие, используйте отправляющее и принимающее события. Отправляющее событие отправляет сообщение в принимающее событие, передавая таким образом данные или иницируя запуск процесса.

3.8.1 Отправка сообщения

Отправляющее событие отправляет сообщение в принимающее событие, передавая таким образом данные или иницируя запуск процесса. После того как сообщение отправлено, процесс автоматически переходит на следующий шаг.

Для того чтобы настроить отправку сообщения:

Добавьте на схему процесса промежуточное или конечное событие-сообщение.

Перейдите в настройки события.

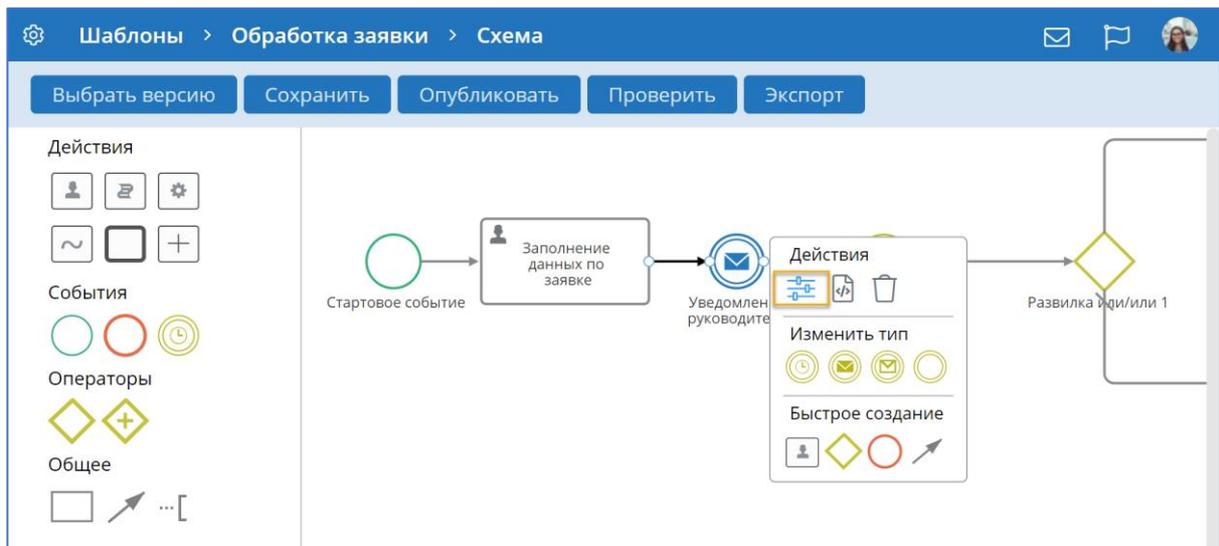


Рис. 259 Настройки события

Дополнительные свойства события-сообщения включают в себя настройки сообщения. В поле Имя сообщения указывается имя отправляемого сообщения. Имейте в виду, что имена отправляемого и получаемого сообщений в связанных событиях должны совпадать.

В области место назначения указывается, куда будет отправлено сообщение: промежуточное событие, стартовое событие или внешний ресурс.

Если сообщение адресовано промежуточному событию, укажите экземпляр процесса, который получит сообщение и продолжительность выполнения процесса.

Имейте в виду, что при выборе опции Значение вам потребуется указать ID активного на данный момент экземпляра процесса.

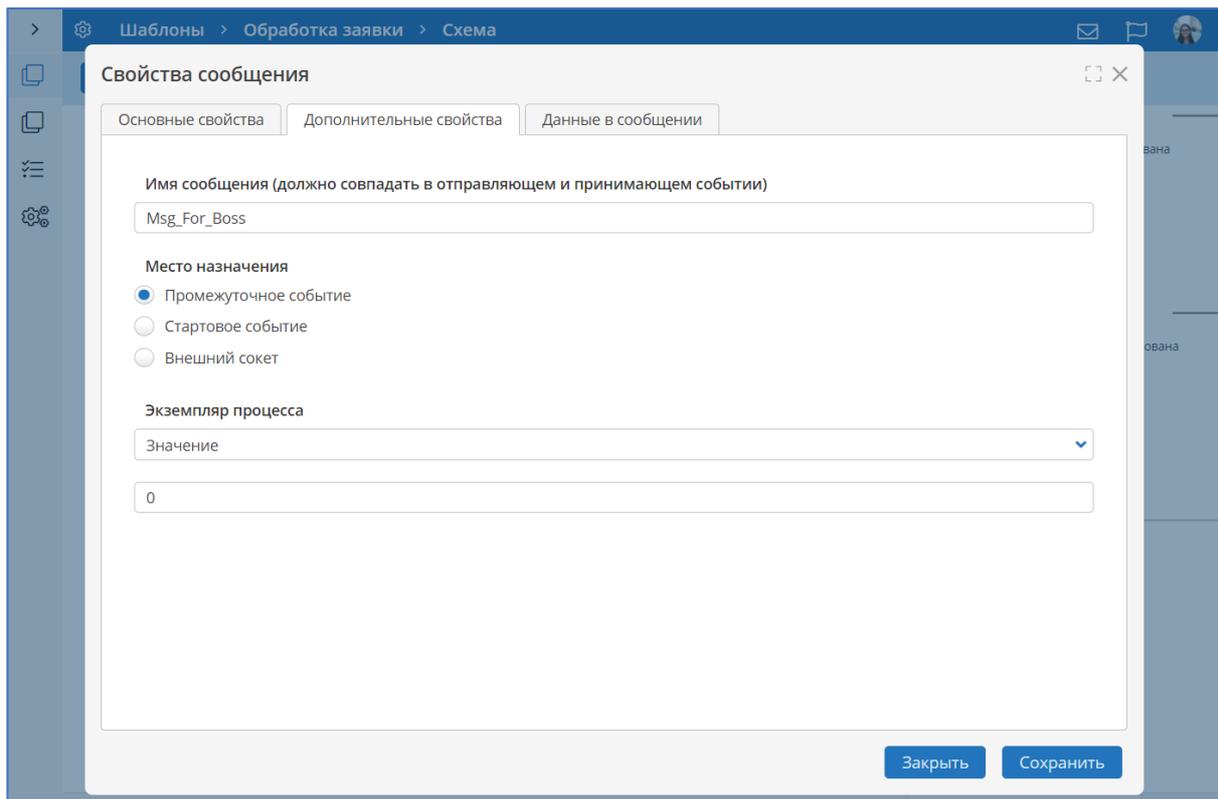


Рис. 260 Дополнительные свойства события-сообщения

Если сообщение адресовано **стартовому событию**, выберите процесс, который будет таким образом запущен и его продолжительность.

Продолжительность процесса (активна только в конечном событии-сообщении) будет учтена при построении отчета в качестве критерия запаздывания процесса. Если время жизни экземпляра процесса превышает максимальную длительность, считается, что процесс выполняется "с опозданием". В противном случае, процесс выполняется "в срок" либо с "риском задержки".

В разделе **«Данные в сообщении»** создается или выбирается **«Шаблон сообщения»** и настраивается соответствие между атрибутами сообщения и отправляемыми данными. Системные имена атрибутов в сообщении – имена, которые отображаются в платформе.

Значения атрибутов сообщения задаются: напрямую, с помощью языка выражений платформы²²³ или C# скрипта, или атрибута. Имейте в виду, что при выборе опции **Атрибут** вам потребуется настроить соответствие между атрибутом сообщения и атрибутом процесса.

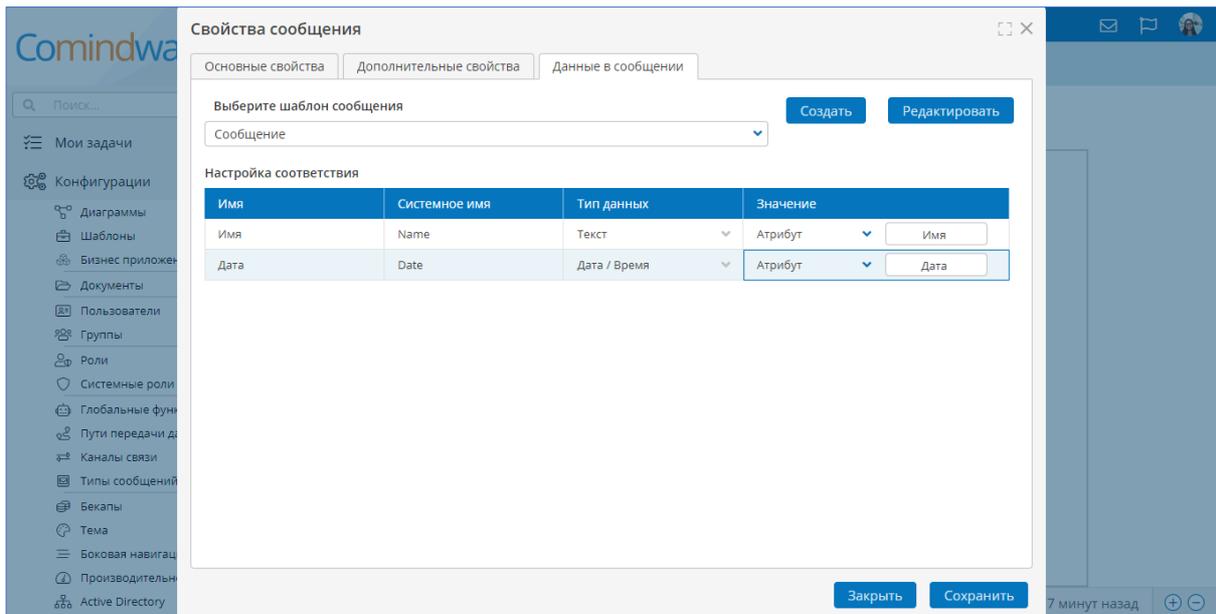


Рис. 261 Настройка атрибутов сообщения

3.8.2 Получение сообщения

Принимающее событие получает сообщение, отправленное событием "Отправка сообщения" или внешним ресурсом. После того как сообщение получено, процесс автоматически переходит на следующий шаг.

Для того чтобы настроить получение сообщения:

Добавьте на схему процесса стартовое или промежуточное событие-сообщение.

Перейдите в настройки события.



Рис. 262 Настройки события

Дополнительные свойства события-сообщения включают в себя настройки сообщения. В поле **Имя сообщения** указывается имя сообщения, отправленного связанным событием "Отправка сообщения" или внешним ресурсом. Событие Отправка сообщения (или путь передачи данных, обеспечивающий взаимодействие процесса с внешним ресурсом) должно быть уже создано и настроено.

Имя	Системное имя	Тип данных	Значение
id	id	Текст	Не задано
Approved	Approved	Логический	Статус согласования

Рис. 263 Настройка события-сообщения

Из раскрывающегося списка **Шаблон сообщения** выбирается необходимый шаблон сообщения (предварительно сконфигурированный в связанном событии "Отправка сообщения" или в связанном пути передачи данных).

Далее настраивается соответствие между атрибутами сообщения и процессными атрибутами.

3.9 Создание процесса

Создание процесса доступно в разделе **Экземпляры** шаблона процесса. Чтобы запустить процесс, нужно:

1. Перейти в раздел **Шаблоны** и выбрать шаблон процесса.
2. Перейти в раздел **Экземпляры**.



Рис. 264 Переход в раздел **Экземпляры** выбранного шаблона процесса

3. Нажать «Создать» на панели инструментов:

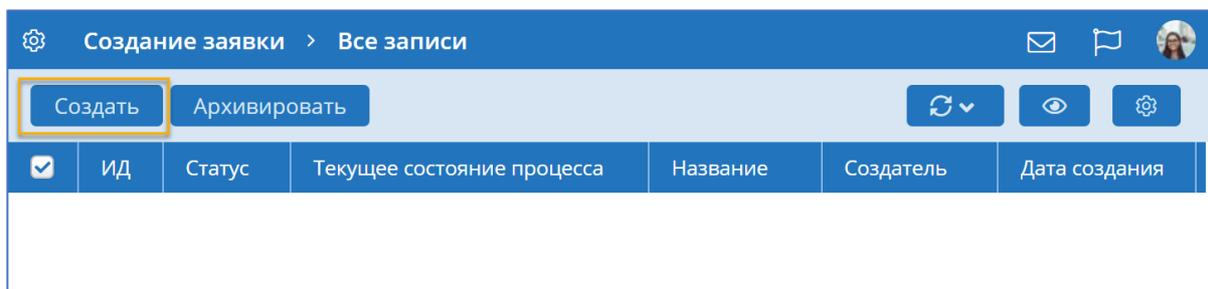


Рис. 265 Создание процесса

В системных задачах процесс запускается по условию (стартовому событию). В пользовательских задачах – пользователю приходит сообщение-уведомление. Чтобы открыть список задач необходимо в левой боковой панели перейти в раздел **«Мои задачи»**.

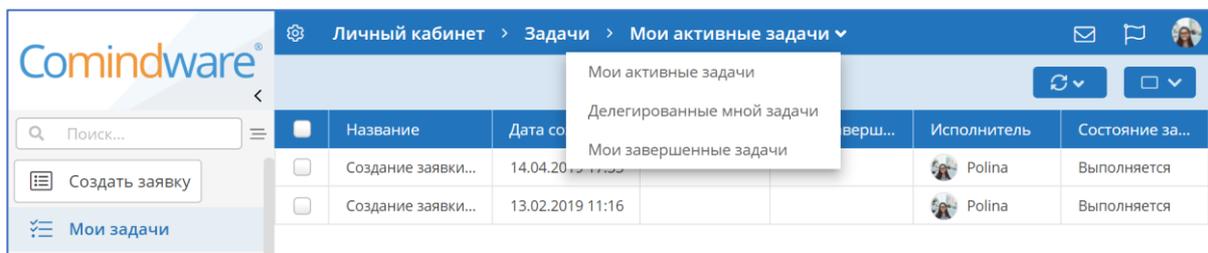


Рис. 266 Мои активные задачи

При выборе задачи из списка справа отобразится форма задачи, выбирая задачи из списка можно переключаться между формами. При этом доступны операции и поля формы. Для перехода к полноэкранной форме задачи нужно выбрать задачу двойным щелчком мыши.

Для отслеживания статуса и истории процесса нужно:

1. Перейти в раздел **Экземпляры** шаблона процесса.
2. Открыть запущенный процесс двойным кликом из списка.

Раздел «История» демонстрирует последовательность выполнения процесса.

На правой панели отображается Информация об объекте и Информация о процессе.

Информация об объекте содержит описание экземпляра запущенного процесса: версию схемы, по которой был запущен процесс, связанный шаблон записи и название связанной записи.

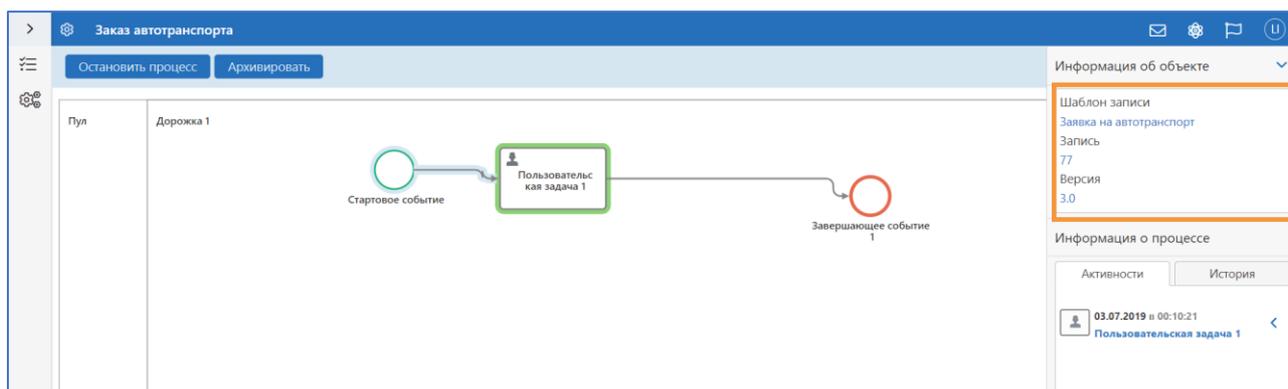


Рис. 267 Информация об объекте

По щелчку на элемент процесса на правой панели в разделе «Информация о процессе» выводится информация об активностях:

- Вкладка «Активности» содержит:
 - информацию об этапе, который процесс проходит в данный момент (для процессов со статусом «Завершен» вкладка содержит надпись «Список пуст»);
 - кнопку для перехода на карточку задачи;
 - кнопку для перезапуска активности;
 - кнопку для удаления активности.
- Вкладка «История» содержит подробную информацию обо всех этапах процесса:
 - дату начала;
 - дату завершения,
 - исполнителя задачи (фактический исполнитель);
 - назначенного исполнителя (указанный в настройках задачи);
 - срок выполнения задачи и др.

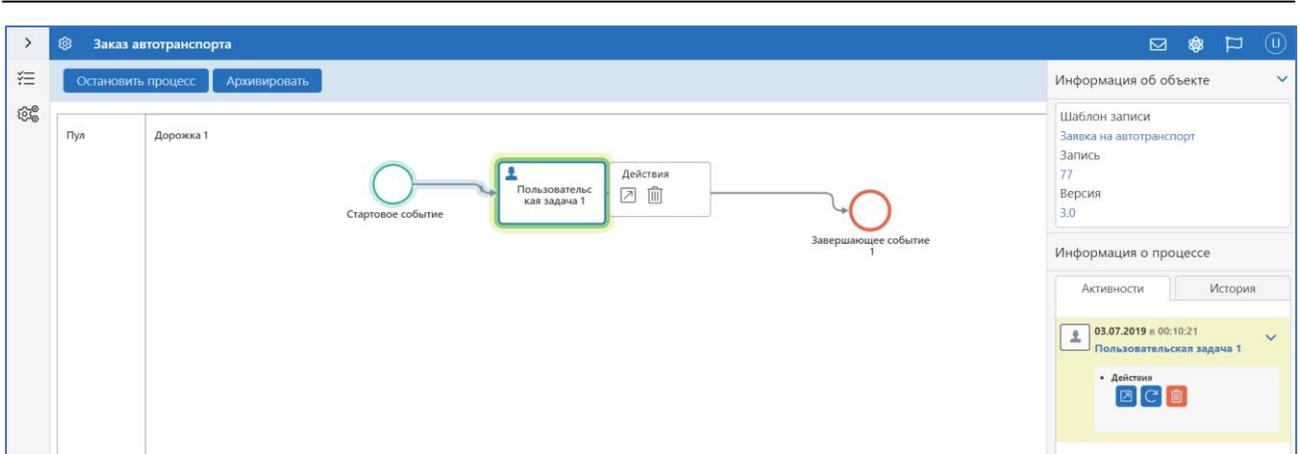


Рис. 268 Информация о процессе

3.10 Настройка списков записей

Для того чтобы настроить списки записей:

1. Откройте шаблон процесса и перейдите в раздел «Списки».

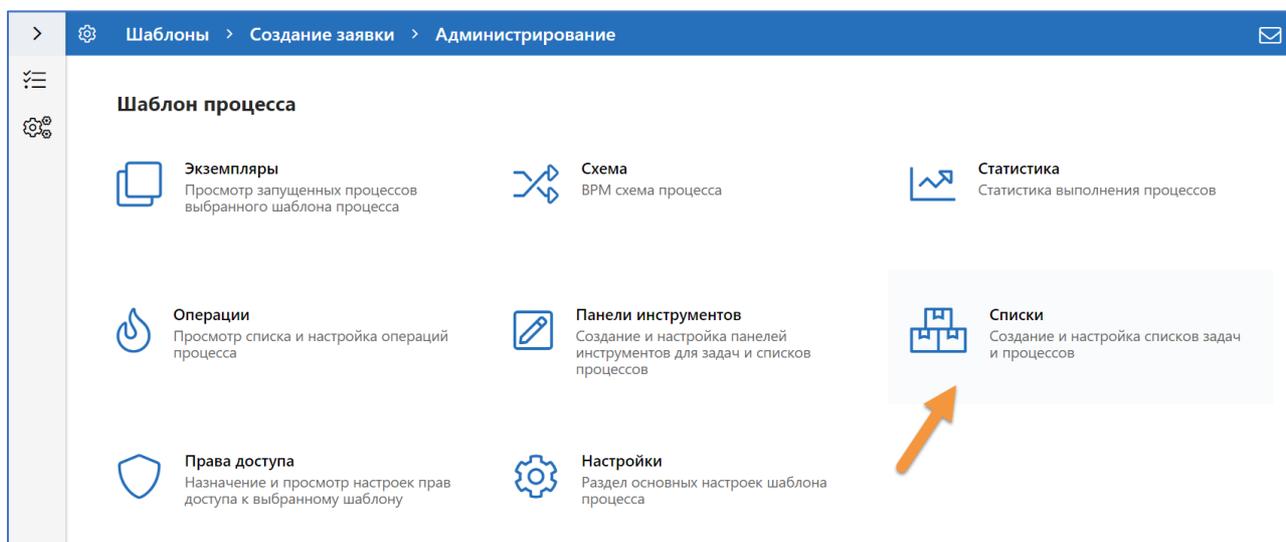


Рис. 269 Меню шаблона процесса

2. Откройте или создайте новый список записей.

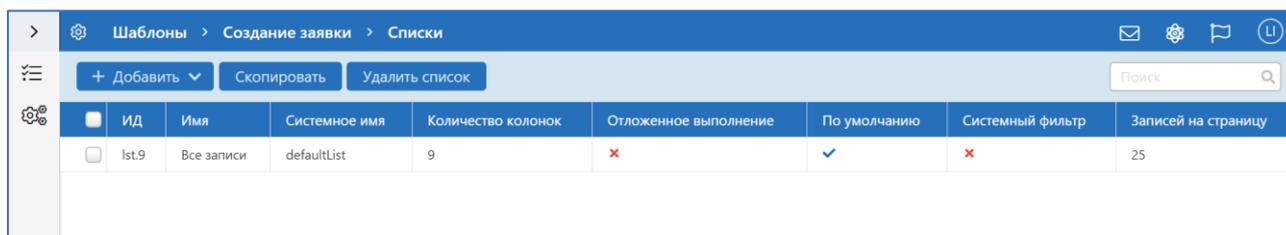


Рис. 270 Список записей

3. В левой части страницы отображается список атрибутов. Центральная форма является конструктором списка. В правой части страницы отображается конфигурация списка и его элементов. В платформе доступно сохранение списка как нового. Операция **Очистить** удаляет все элементы из настроенного списка.

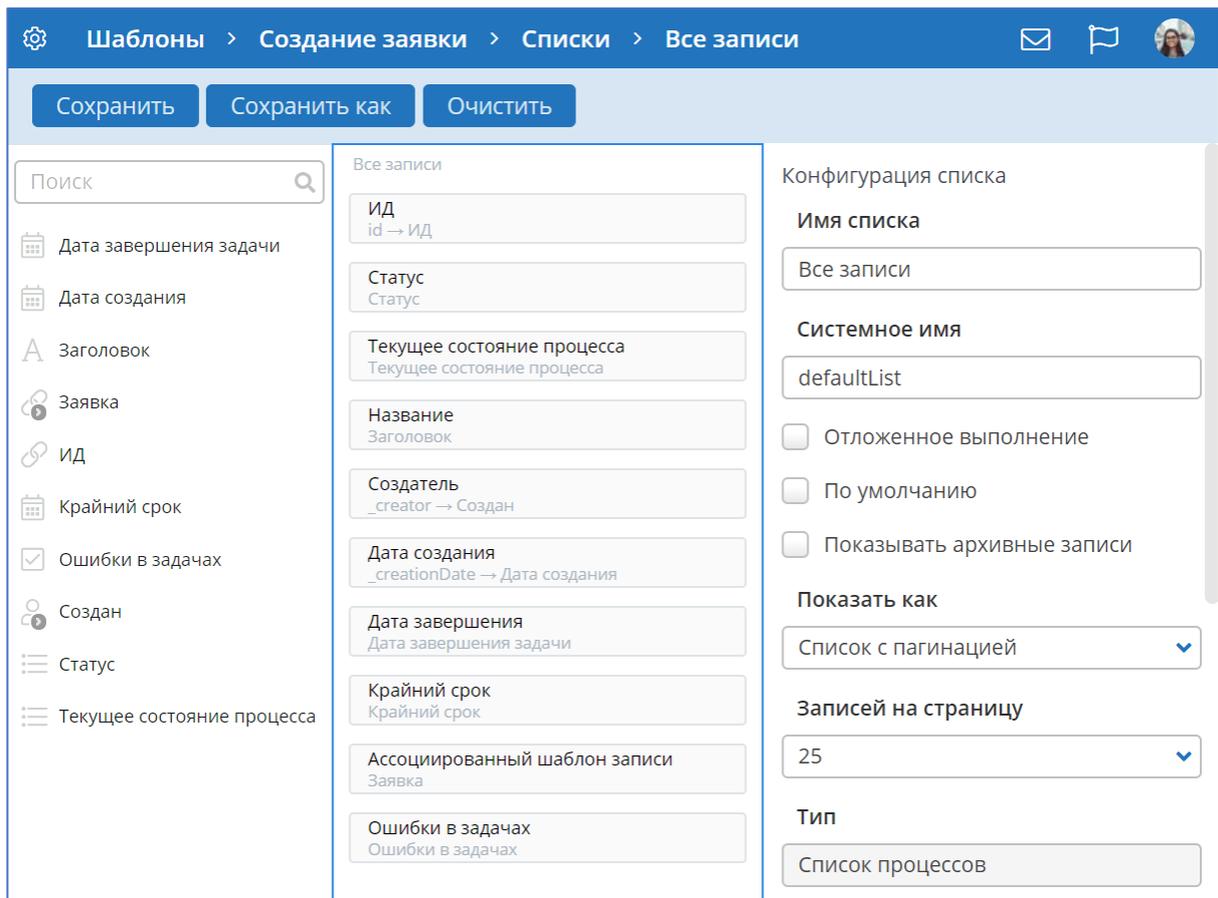


Рис. 271 Редактор списка

4. Настройте конфигурацию нового списка. При отметке флага «Отложенное выполнение» создается пустой список. Суть операции заключается в предварительной настройке колонок. Загрузка данных происходит при нажатии на кнопку «Загрузить данные». Данные отображаются согласно настройкам списка. Укажите отображение списка записей плиткой или списком с пагинацией. В поле системного фильтра с помощью языка запросов N3 задайте условие отображения данных.
5. Настройте колонки, путем перетаскивания их в центральную форму. Колонки отображаются в порядке очередности на форме.

Настроить отображение данных можно напрямую в списках. Для этого пользователь может навести на колонку и настроить фильтрацию/сортировку/группировку/агрегацию или выбрать колонки для скрытия и отображения:

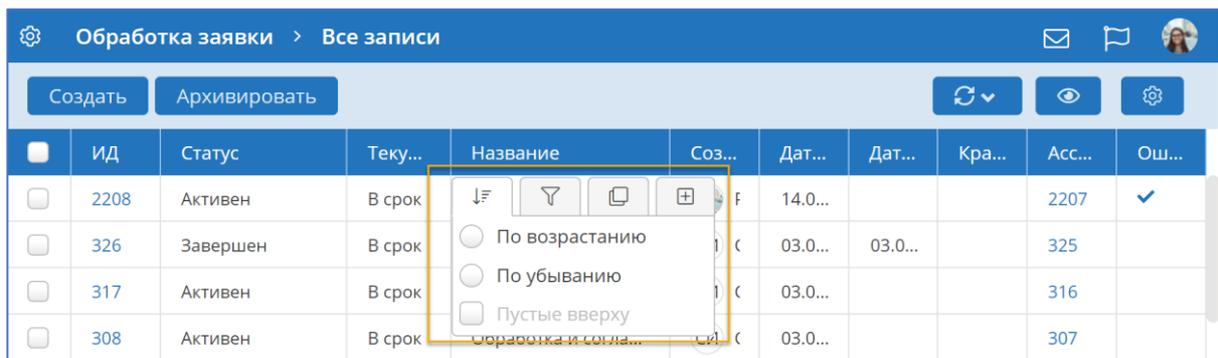


Рис. 272 Настройка фильтрации колонок

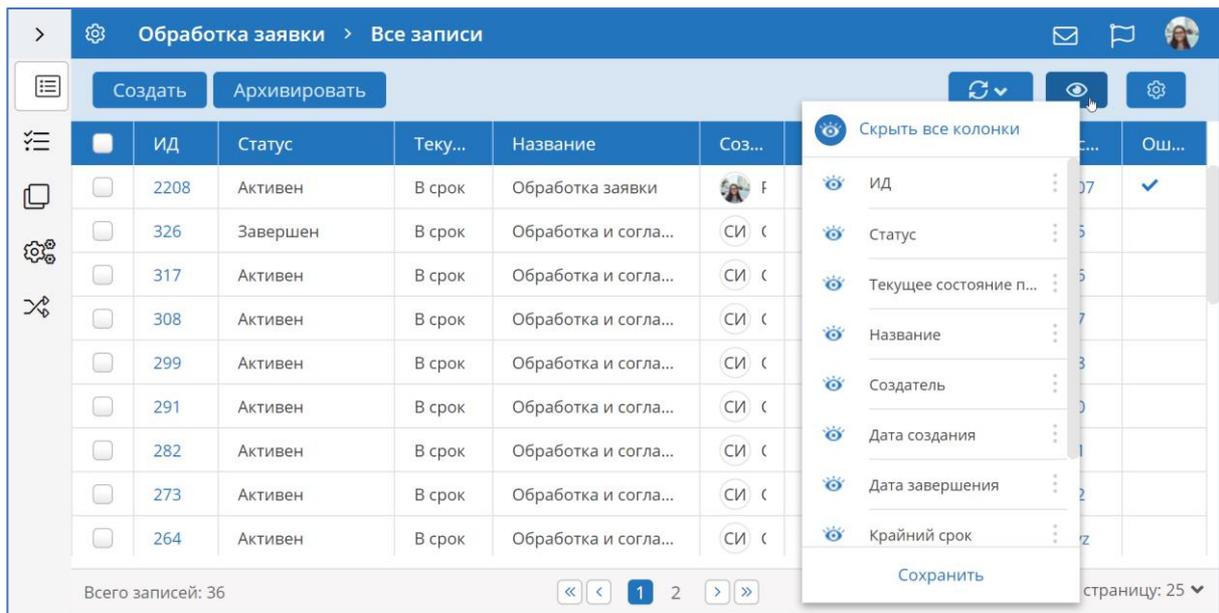


Рис. 273 Настройка скрытия и отображения колонок на списке

3.11 Введение в управление кейсами

Кейс представляет собой нестандартную бизнес-задачу, курируемую ответственным лицом или группой лиц. Например, кейсом является развертывание новой бизнес-модели, поскольку деятельность подобного рода не является совокупностью предписанных рутинных действий.

- Кейсы настраиваются и управляются в форме шаблонов и экземпляров кейсов.
- Кейс может быть запущен в рамках процесса, в разделе «Процессы», а также в обсуждении или комментарии к задаче.
- Кейс может служить контейнером для задач с любой степенью вложенности. Разница между задачей и кейсом заключается в том, что кейс имеет связанный шаблон записи, а задача нет.

3.12 Создание и настройка шаблона кейса

Для того чтобы создать шаблон кейса:

1. На боковой панели навигации перейдите в раздел «Шаблоны». Добавьте **Новый шаблон кейса**.

Новый шаблон

Имя* Системное имя*

Введите текст Введите текст

Тип шаблона*

Шаблон кейса

Описание

Введите текст

Варианты создания нового шаблона

Создать новый шаблон

Связать с существующим шаблоном

Имя нового шаблона записи* Системное имя нового шаблона записи*

Введите текст Введите текст

Отмена Создать

Рис. 274 Создание шаблона кейса

2. В форме **Новый шаблон кейса** укажите имя шаблона. Помимо шаблона кейса, платформа автоматически создает новый шаблон записи, в котором хранятся специфические сведения о кейсе (атрибуты кейса, данные экземпляров кейса и т. д.). Введите имя для нового шаблона записи или свяжите шаблон кейса с существующим шаблоном записи.

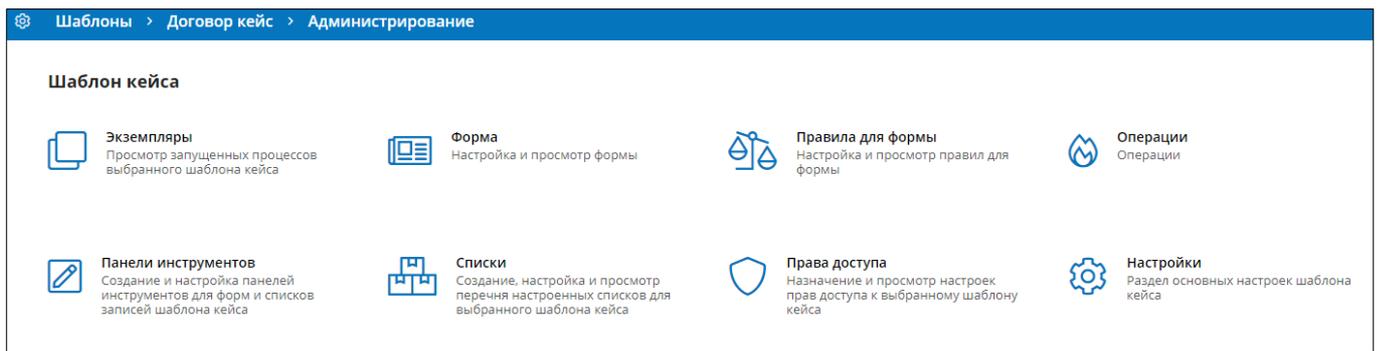


Рис. 275 Меню кейса

3. В разделе «**Форма**» создайте для кейса одну или несколько форм по типу формы шаблона записи. Создайте, при необходимости, правила для формы.³⁶
4. Настройте списки записей в разделе «**Списки**».

3.13 Настройка доступа к процессам и кейсам

Для того чтобы настроить доступ к шаблону и экземплярам процесса или кейса:

1. Откройте шаблон. Перейдите в раздел «**Безопасность**».

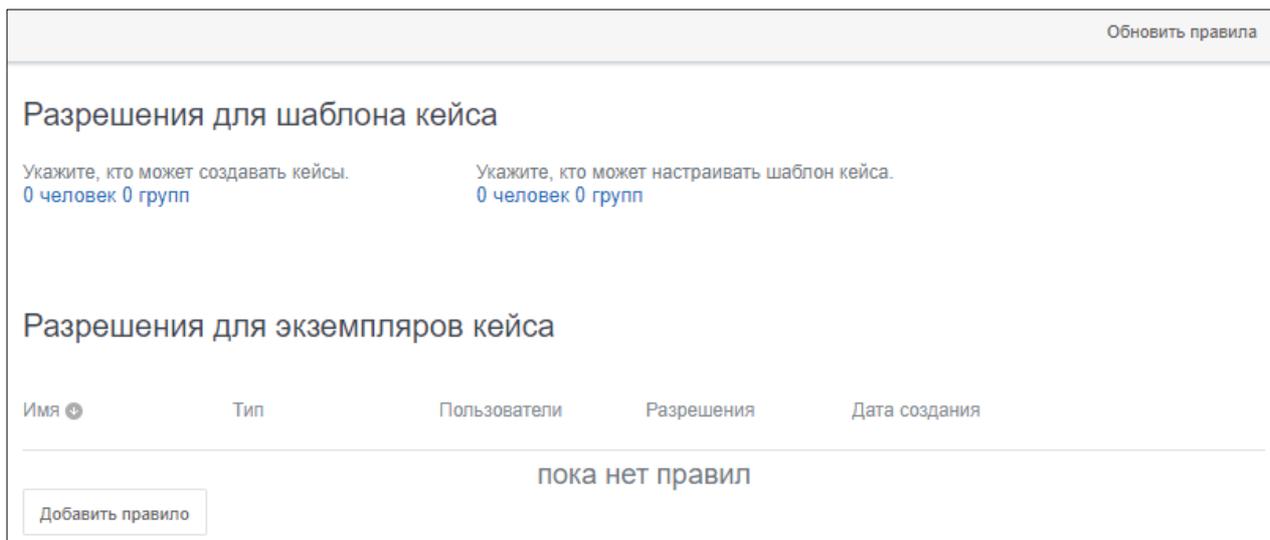


Рис. 276 Меню безопасность

2. Укажите пользователей, которые могут создавать и/или настраивать экземпляры процесса/кейса.
3. Добавьте особое правило доступа к экземплярам процесса/кейса.
4. Заполните данные для правила.
 - Задайте имя и тип правила.
 - Задать права пользователям можно двумя способами: непосредственно значением – открывается новое окно со списком всех пользователей и групп пользователей; выражением – открывается новое окно для записи условия, при выполнении которого пользователям задаются права.

- Укажите, какие действия регламентирует правило: **Чтение**, **Запись** или **Удаление**. Данные действия в зависимости от типа правила будут или разрешены, или запрещены.
- В поле **Область применения правила** задайте с помощью языка выражений платформы²²³ записи, к которым будет применено правило (например, настройка правил²⁶¹)

Создать Отмена

Введите текст

* Укажите, кому при активации правила будут выданы права на кейсы.
Значение

0 человек 0 групп

Выберите разрешения

Чтение Удаление
 Запись

Область применения правила

← → **B** *I* 🔍 🔗

1

Рис. 277 Настройка данных правила

4. Взаимодействие с внешними ресурсами

В этой главе:



[Основные сведения о взаимодействии с внешними ресурсами](#)^[4.1]

Информация о том, с какими внешними ресурсами могут взаимодействовать процессы и как выполнить настройку.



[Путь передачи электронной почты](#)^[4.4]

Настройка пути передачи данных через канал электронной почты.



[Канал электронной почты](#)^[4.0]

Настройка канала отправки и получения электронных сообщений через внешний сервер электронной почты.



[Путь передачи веб-запроса](#)^[4.5]

Настройка пути передачи данных через веб-канал.



[Веб-канал для приема запросов](#)^[4.3]

Настройка канала получения запросов от внешней веб-службы.

4.1 Основные сведения о взаимодействии с внешними ресурсами

В Comindware Business Application Platform процессы могут взаимодействовать со следующими внешними ресурсами:

- **Сервер эл. почты:** событие процесса получает или отправляет электронные сообщения через внешний сервер электронной почты.
- **Веб-служба:** событие процесса получает запросы от внешней веб-службы.

Реализация взаимодействия процесса с внешним ресурсом состоит из трех этапов:

№	Место в платформе	Действие
1	Боковая навигация -> Каналы связи	Настройте канал электронной почты или веб-канал
2	Боковая навигация -> Пути передачи данных	Создайте путь передачи данных через канал эл. почты или веб-канал. Настройте сообщение, которое будет передаваться по настроенному пути и посредством которого процесс будет взаимодействовать с внешним ресурсом.
3	Боковая навигация -> Шаблоны -> шаблон процесса -> Схема	Добавьте на схему процесса событие, принимающее или отправляющее сообщения. Имейте в виду, что имя сообщения, указанное при настройке пути передачи данных, должно совпадать с именем сообщения в событии процесса. Настройте соответствие между атрибутами процесса и атрибутами сообщения.

При настройке сообщений на процессе должен быть выбран (создан) шаблон сообщения (4.6). После этого создается канал связи (4.2, 4.3) и настраивается путь передачи данных (4.4, 4.5) для этого шаблона сообщения, где указан используемый канал связи.

4.2 Канал электронной почты

Для настройки канала связи с внешним сервером электронной почты:

1. Необходимо открыть раздел **Каналы связи** в настройках платформы.

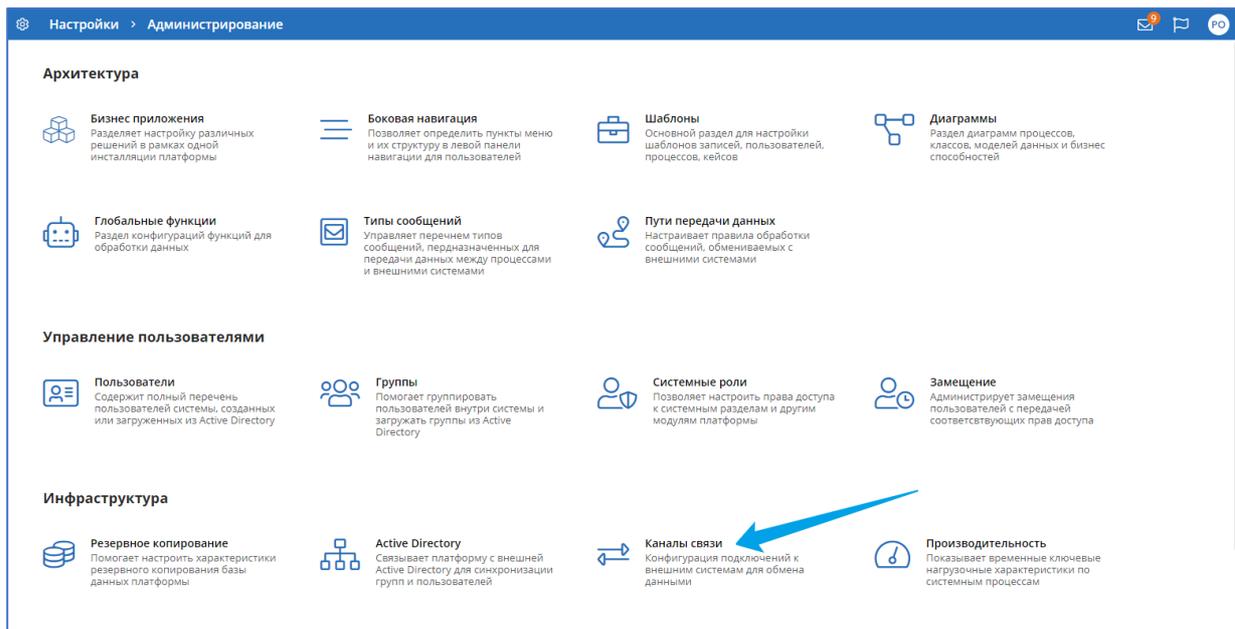


Рис. 278 Расположение раздела «Каналы связи»

2. Настройки платформы
3. Добавьте канал для обмена эл. сообщениями.



Рис. 279 Раздел «Каналы связи»

4. Введите имя канала, при необходимости сделайте канал каналом по умолчанию.

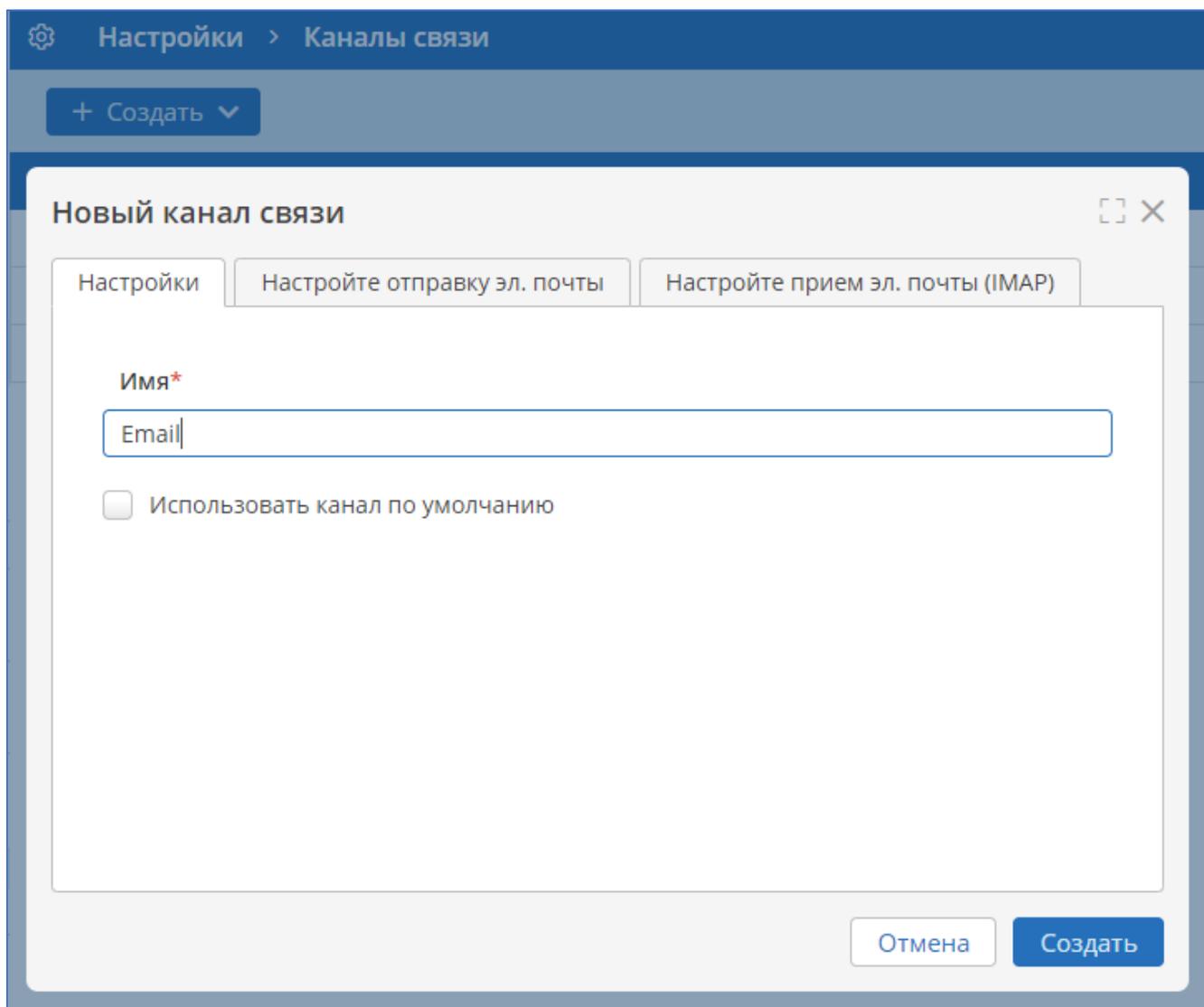


Рис. 280 Настройка канала связи

5. Настройте отправку электронной почты:

Новый канал связи

Настройки Настройте отправку эл. почты Настройте прием эл. почты (IMAP)

Включено

Протокол* URL почтового сервера*

SMTP Введите текст

Порт* Защита данных

25 SSL

Имя пользователя Пароль

admin ..

Адрес отправителя* Имя отправителя

Введите текст Введите текст

Отмена Создать

Рисунок 1

Рис. 281 Настройка отправки эл. почты

- Выберите протокол: Microsoft Exchange или SMTP;
- Введите имя хоста и порт;
- Выберите способ защиты данных;
- Укажите адрес отправителя;

Для того, чтобы проверить настройку отправки эл. почты введите адрес получателя и отправьте тестовое сообщение.

6. Настройте получение электронной почты (IMAP):

Рисунок 2

Рис. 282 Настройка приема эл. почты

Для того, чтобы проверить настройку приема эл. почты нажмите на кнопку «Проверить входящее соединение».

4.3 Веб-канал для приема запросов

Для настройки канала связи с внешней веб-службой:

1. Необходимо открыть раздел **Каналы связи** в настройках платформы.
2. Добавьте канал для приема веб-запросов.

Канал для обмена эл. сообщениями	Тип
Канал для приема веб-запросов	Url
Канал для синхронизации с веб-сервисом	Mail
Odata channel78	WebService

Рис. 283 Раздел «Каналы связи»

3. Введите имя канала и укажите URL веб-службы.

Набирая в поле URL веб-службы, вы увидите, что ниже автоматически отобразится URL обработки запросов. Используйте URL обработки запросов при настройке взаимодействия с Comindware Business Application Platform на стороне веб-службы.

Новый канал связи ☐ ×

Имя*

URL приема внешних запросов*

URL обслуживания запросов

Заглавие сообщения

Тело сообщения

Рис. 284 Создание нового канала связи для приема веб - запросов

4.4 Путь передачи электронной почты

Путь передачи данных создается для каждого события процесса типа «Сообщение» (3.3.2), которое используется для взаимодействия процесса с внешним ресурсом (сервером эл. почты или веб-службой).

Для создания пути передачи данных через канал электронной почты:

Необходимо открыть раздел **Пути передачи данных** в настройках платформы. Создайте входящий и/или исходящий путь.

Имя	Тип	Бизнес-приложение	Имя канала
флаб	Incoming	sln.5	Email
Approval	Incoming	sln.3	Approval Channel
Approve	Outgoing	sln.3	Email

Рис. 285 Пути передачи данных

Входящий путь

Выберите ранее настроенный Канал связи (0).

Укажите имя электронного сообщения, которое будет приниматься по настраиваемому пути передачи данных. Имейте в виду, что данное имя должно совпадать с именем шаблона сообщения в связанном событии процесса, принимающем сообщения (3.8.2).

Новый путь передачи данных

Основные настройки | Шаблон сообщения

Имя* Путь активен

Канал связи* Имя сообщения*

Бизнес-приложение*

Процесс*

Назначение*

Отмена Создать

Рис. 286 Настройки входящего пути передачи данных

При необходимости настройте данные, которые будет содержать в себе эл. сообщение. На вкладке **Шаблон сообщения** выберите существующий или добавьте новый шаблон: укажите имя и добавьте атрибуты.

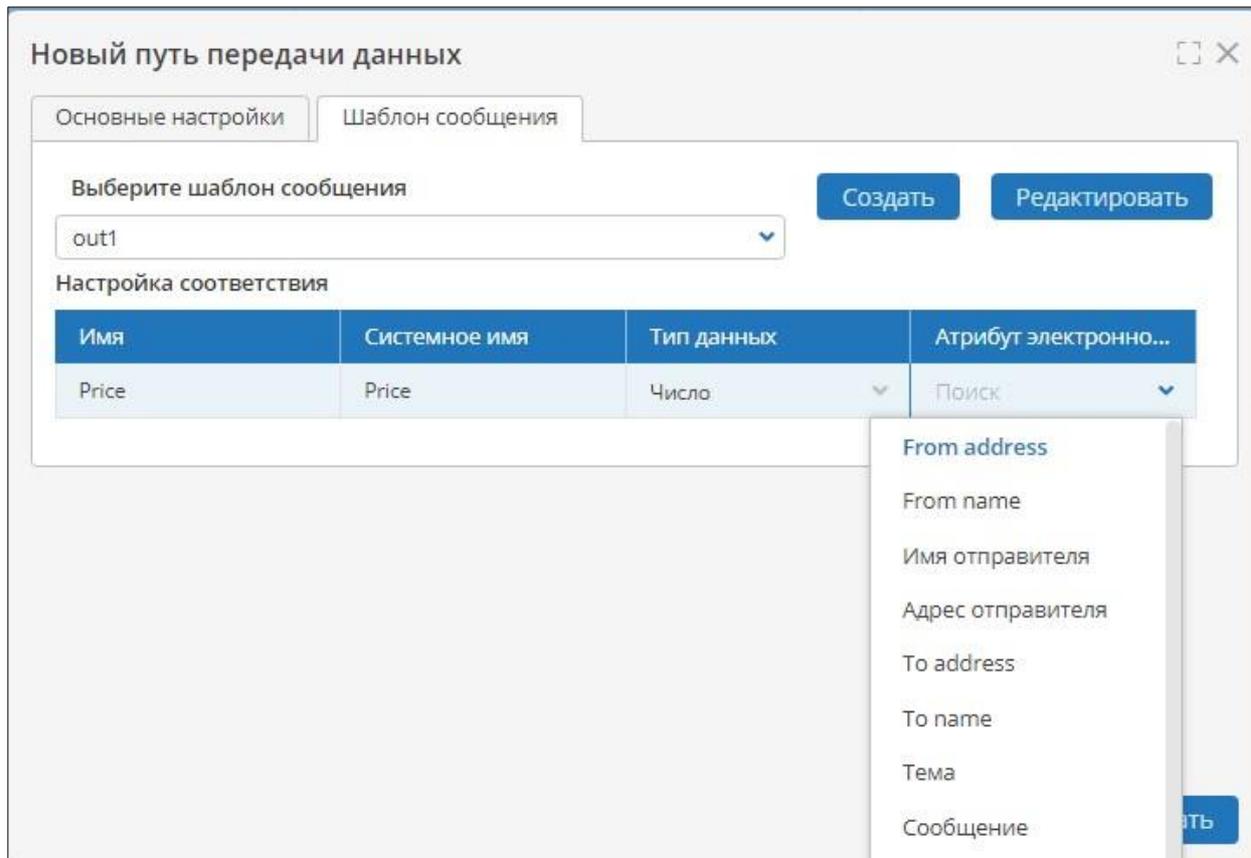


Рис. 287 Настройки шаблона сообщений

Исходящий путь

Выберите ранее настроенный Канал связи.

Укажите имя электронного сообщения, которое будет отправляться по настраиваемому пути передачи данных. Имейте в виду, что данное имя должно совпадать с именем сообщения в связанном событии процесса, отправляющем сообщения (3.8.1).

Настройки > Путь передачи данных

Добавить входящий путь Добавить исходящий путь

Имя ↓ Тип

Новый путь передачи данных

Основные настройки Шаблон сообщения Настройте электронное сообщение

Имя* Путь активен

Введите текст

Канал связи*

Поиск

Использовать персональный канал связи

Имя сообщения* **Бизнес-приложение***

Введите текст Поиск

Отмена Создать

Рис. 288 Настройки исходящего пути передачи данных

При необходимости настройте данные, которые будут содержать в себе эл. сообщение. На вкладке «Шаблон сообщения» выберите существующий или добавьте новый шаблон: укажите имя и добавьте атрибуты.

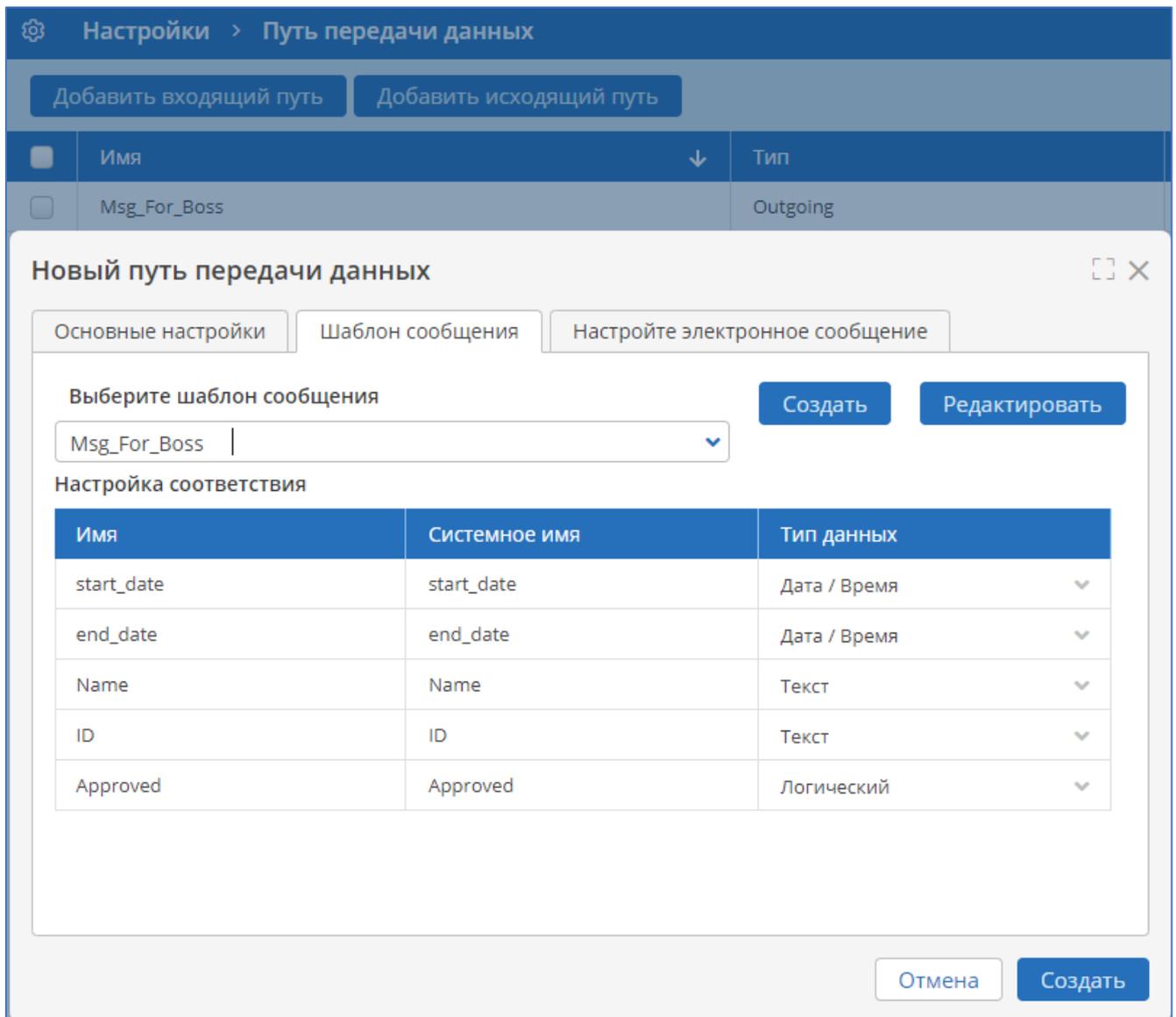


Рис. 289 Настройки шаблона сообщений

Атрибуты используемого шаблона сообщения будут доступны для настройки соответствия стандартным атрибутам электронного сообщения. Для этого на вкладке «Настройте электронное сообщение» укажите атрибуты шаблона сообщения в фигурных скобках в соответствующих полях.

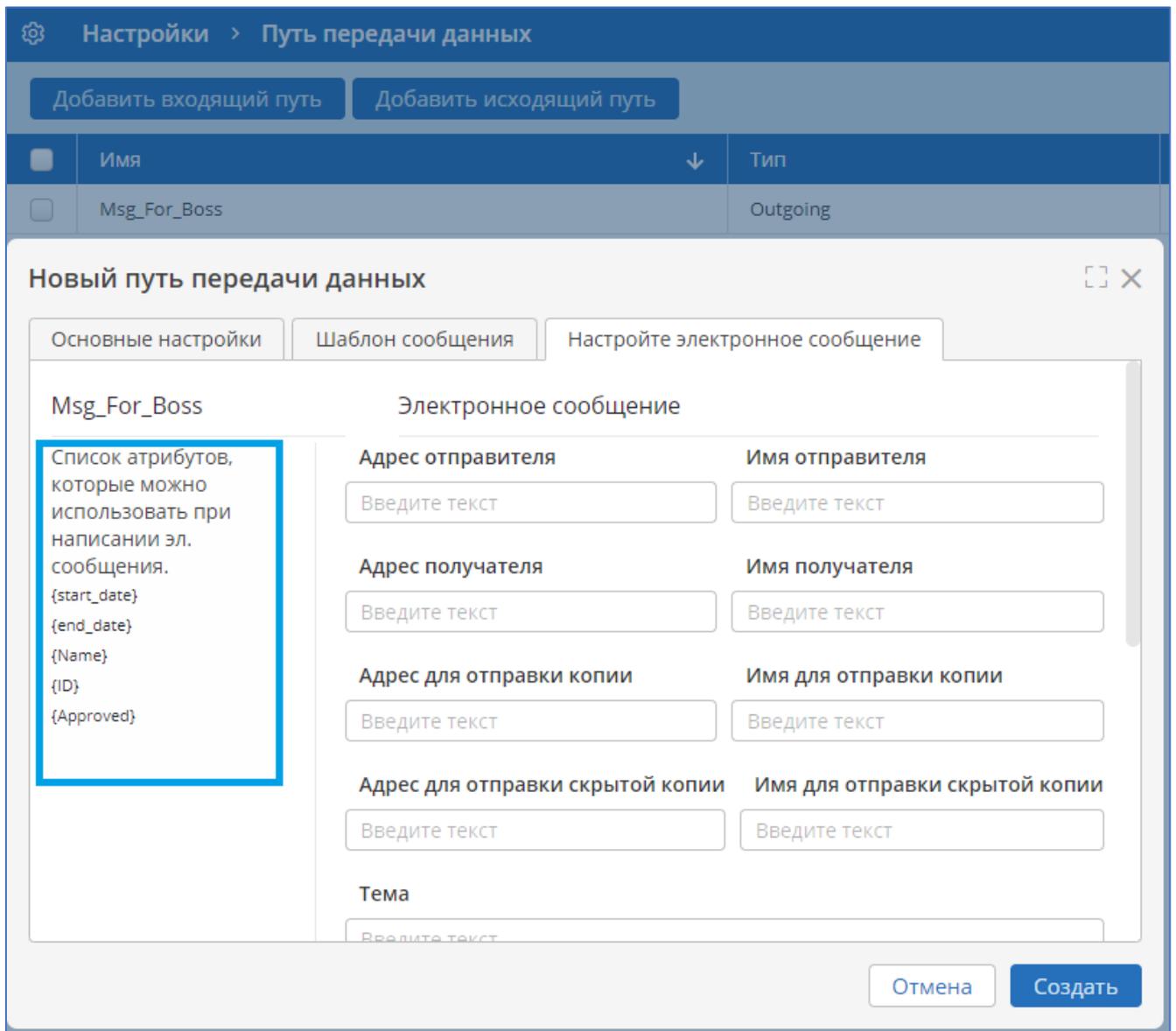


Рис. 290 Список атрибутов шаблона сообщения, доступных для настройки исходящего пути передачи данных

4.5 Путь передачи веб-запросов

Путь передачи данных создается для каждого события процесса типа «Сообщение», которое используется для взаимодействия процесса с внешним ресурсом (сервером эл. почты или веб-службой).

Передача веб-запроса происходит по следующей схеме:



Для создания пути передачи данных через веб-канал:

1. На боковой панели навигации откройте **Пути передачи данных**. Создайте входящий путь.
2. Из раскрывающегося списка **Канал** выберите веб-канал. Канал должен быть предварительно настроен.
3. Укажите имя сообщения от внешней веб-службы, которое будет приниматься по настраиваемому пути передачи данных. Имейте в виду, что данное имя должно совпадать с именем сообщения в связанном событии процесса, принимающем сообщения.
4. Из раскрывающегося списка **Процесс** выберите шаблон процесса, в записи которого будут отправляться сообщения от веб-службы.

Рис. 291 Настройки входящего пути передачи данных

Из раскрывающегося списка **Метод парсинга** выберите метод парсинга, который будет применяться к сообщению от веб-службы: **Как есть**, **XML**, или **JSON**.

При необходимости настройте данные, которые будут содержать в себе сообщение от веб-службы. В группе **Шаблон сообщения** выберите существующий или добавьте новый шаблон: укажите имя и добавьте атрибуты

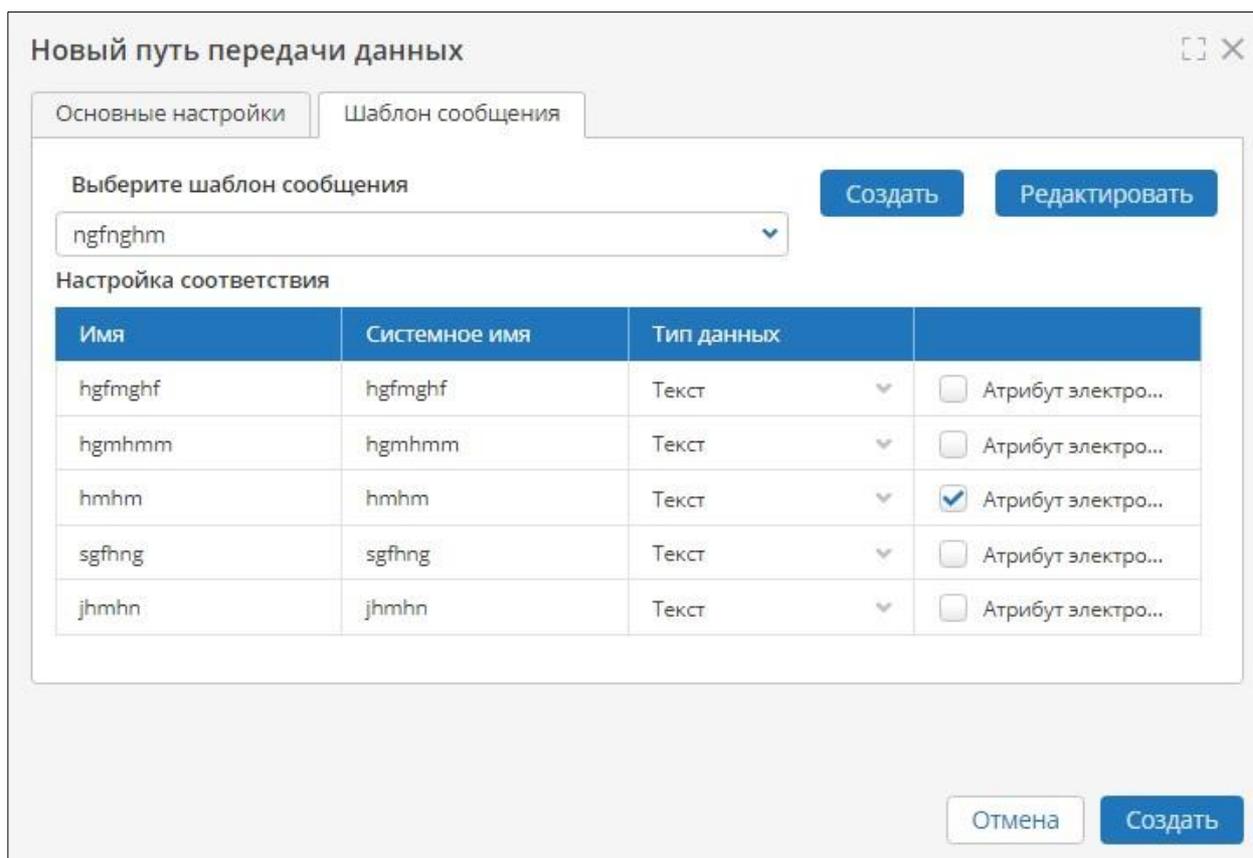


Рис. 292 Настройки шаблона сообщения

4.6 Типы сообщений

Для того, чтобы настроить шаблон сообщения необходимо перейти в раздел типы сообщений.

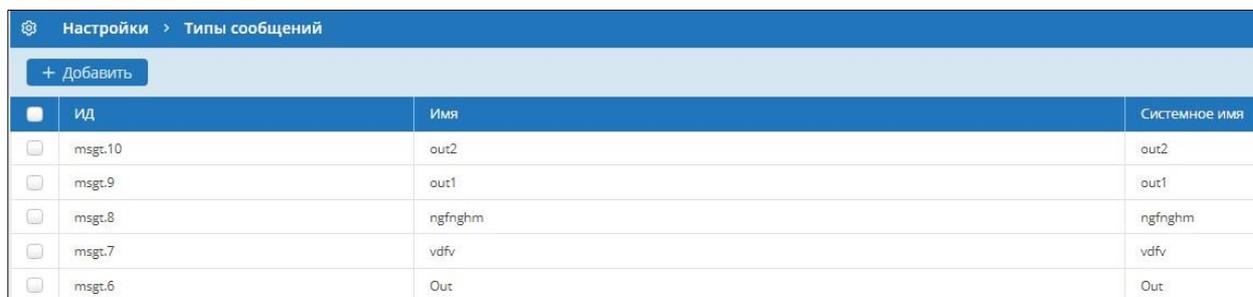


Рис. 293 Раздел «Типы сообщений»

1. Нажмите на кнопку «Добавить».
2. Необходимо задать имя и системное имя шаблона сообщений
3. Необходимо задать имя, системное имя и тип данных для настройки соответствия атрибутов из шаблона записи и приходящих из внешних ресурсов.

Новый шаблон сообщения ☐ X

Имя* Системное имя*

Атрибуты шаблона сообщения

+ Добавить

<input type="checkbox"/>	Имя	Системное имя ?	Тип данных
<input type="checkbox"/>	* уца	* utsa	* Текст ▼
<input type="checkbox"/>	* мвфа	* mvfa	* Число ▼
<input type="checkbox"/>	* ывм	* yvm	* Дата / Время ▼
<input type="checkbox"/>	* уцамфц	* utsamfts	* Поиск ▼

Текст

Число

Целое число

Длительность

Готово

Рис. 294 Окно настройки шаблона сообщений

5. Язык выражений Comindware Business Application Platform

В этой главе:



[Введение в язык выражений Comindware Business Application Platform](#)²²⁴

Информация о том, как использование языка выражений Comindware Business Application Platform может улучшить ваши бизнес процессы с точки зрения удобства и эффективности.



[Основные сведения](#)²²³

Синтаксис выражений Comindware Business Application Platform: объекты, идентификаторы и системные имена, типы данных, функции, литералы и операторы.

5.1 Введение в язык выражений Comindware Business Application Platform

Настройка рабочей среды и пользовательского интерфейса Comindware Business Application Platform производится простым перетаскиванием элементов и не требует программирования. В случае если ваш бизнес требует нетривиального подхода, воспользуйтесь возможностями языка выражений Comindware Business Application Platform, легкого, но эффективного языка программирования, встроенного в платформу.

С помощью выражений вы сможете неограниченно масштабировать бизнес-процессы в согласии с нуждами вашего бизнеса.

Возможности языка выражений Comindware Business Application Platform:

- Настройка значений атрибутов на основании данных, хранящихся в различных шаблонах записи.
- Проверка и изменение процессных данных.
- Задание временных рамок выполнения тех или иных процессных действий, например, разрешенные запросы в тех. поддержку могут закрываться автоматически после двух недель в неактивном состоянии.
- Назначение исполнителей задачи на основании того или иного условия (например, в зависимости от категории запроса).
- Настройка межпроцессного взаимодействия.
- Настройка вызова стороннего процесса внутри текущего.
- Настройка условий ветвления процесса.
- Защита платформы от негативного влияния человеческого фактора.

5.2 Основные сведения

В этом разделе:



[Идентификаторы и системные имена](#)²²⁶

Назначение уникальных имен объектам и свойствам объектов Comindware Business Application Platform.



[Функции](#)²²⁷

Синтаксис и использование функций в языке выражений Comindware Business Application Platform.



[Типы данных](#)²²⁷

Типы данных, используемые в выражениях Comindware Business Application Platform.



[Операторы](#)²⁵³

Синтаксис и использование операторов в языке выражений Comindware Business Application Platform.



[Вызов связанных данных](#)²⁵⁸

Информация о том, как вызвать данные связанного шаблона записи, используя ссылку на этот шаблон.



[Литералы](#)²⁵⁴

Синтаксис и использование литералов в языке выражений Comindware Business Application Platform.

5.2.1 Объекты

Большинство рабочих элементов, с которыми вы имеете дело в Comindware Business Application Platform, представлены частными случаями типа данных **Объект**:

- Учетная запись пользователя
- Группа пользователей
- Пользовательский объект

5.2.2 Идентификаторы и системные имена

В целях идентификации объектам и свойствам объектов Comindware Business Application Platform должны быть присвоены уникальные имена. В платформе существуют два типа уникальных имен:

- Идентификатор
- Системное имя

Идентификатор представляет собой уникальное имя, которое задается платформой для идентификации следующих объектов Comindware Business Application Platform:

- учетная запись пользователя
- группа пользователей
- задача
- статус

Системное имя представляет собой уникальное имя, которое задается создателем и служит для идентификации пользовательского объекта (например, шаблона записи).

Уникальные идентификаторы и системные имена также используются для идентификации **свойств объектов**, например, атрибутов шаблона записи.

Как идентификаторы, так и системные имена позволяют администратору, архитектору и специалистам по управлению данными и процессами обращаться к объектам и их свойствам в выражениях Comindware Business Application Platform.

Работая с именами в Comindware Process, придерживайтесь следующих правил:

1. В выражениях используйте только идентификаторы и системные имена, а не имена, отображаемые в платформе.
2. Идентификаторы и системные имена чувствительны к регистру.
3. Системные имена должны начинаться с буквы или подчеркивания (_). В самом системном имени можно использовать латинские буквы (A-Z, a-z), числа (0-9) и подчеркивания (_).
4. Системные имена должны быть уникальны только в рамках того или иного шаблона записи. Некоторые атрибуты шаблонов записи создаются автоматически платформой под формализованными именами, такими как id, _creationDate, _creator, _lastWriteDate и др.
5. Для того чтобы обратиться к тому или иному свойству текущего объекта, используйте оператор `$->Property` (краткая форма `$Property`), где символ `$` представляет текущий объект, а `Property` - идентификатор или системное имя свойства объекта.

5.2.3 Типы данных

В таблице ниже приведены типы данных, используемые в выражениях Comindware Business Application Platform.

Тип данных	Описание
Объект	Тип данных Объект используется для объявления объекта, например, задачи, шаблона записи или учетной записи пользователя.
Число	Используется для объявления чисел с плавающей точкой.
Строка	Данный тип используется для объявления последовательностей символов.
Логический	Переменные данного типа могут принимать только для значения: истина/ложь.
Список	Используется для представления упорядоченного набора элементов одного типа данных.
Дата/время	Данный тип используется для объявления переменных, содержащих значения даты и времени.
Длительность	Данный тип используется для представления временных интервалов. Значение временного интервала хранится в секундах в ISO формате.

Вы можете складывать и вычитать значения Дата/Время (DateTime) и Длительность между собой, руководствуясь следующими правилами:

Действие	Результат
Длительность + длительность	Длительность
Дата/время + длительность	Дата/время
Длительность - длительность	Длительность
Дата/время - длительность	Дата/время
Дата/время – дата/время	Длительность

5.2.4 Функции

В данном разделе приведен синтаксис и использование функций в языке выражений Comindware.

ADD ()	
Описание	Суммирует два значения. Конкатенация строк не поддерживается.
Синтаксис функции	ADD(argument1, argument2)
Аргументы	argument1: число, argument2: число argument1: длительность, argument2: длительность argument1: дата и время, argument2: длительность

ADD ()	
Возвращает	число длительность дата/время

ALL ()	
Описание	Берет список логических переменных и возвращает значение Истина, если все логические переменные принимают значение Истина.
Синтаксис функции	ALL(argument1)
Аргументы	argument1: список логических переменных, или список выражений, результатом которых являются значения логического типа.
Возвращает	значение логического типа

AND ()	
Описание	Берет два значения логического типа или два выражения, результатом которых являются значения логического типа, и возвращает результат логического умножения этих значений.
Синтаксис функции	AND(argument1, argument2)
Аргументы	argument1, argument2: значение логического типа или выражение, результатом которого является значение логического типа.
Возвращает	значение логического типа

ANY (), проверка на непустую последовательность	
Описание	Проверяет список на не пустоту. По условию выборки формирует последовательность значений любого типа и проверяет её на не пустоту.
Синтаксис функции	ANY(list), ANY((sequence)) Для задания последовательности используйте стандартный запрос ^[25] from where select. Пример: ANY(from l in db->Tickets where i->product = 'Data Monsoon' select i->name). Данное выражение вернет значение Истина, если хотя бы один элемент имеет значение предиката product равным «Data Monsoon».

ANY (), проверка на непустую последовательность	
Аргументы	list: список значений любого типа sequence: условие выборки
Возвращает	значение логического типа

ANY (), проверка предиката	
Описание	Формирует выборку элементов и проверяет, что в ней есть хотя бы один элемент с истинным значением заданного предиката.
Синтаксис функции	ANY([(argument1), argument2, argument3]) Пример: ANY([(from l in db->Tickets where i->product=='Data Monsoon' select i), i, i->NeedsQAHelp]) Данное выражение вернет значение Истина, если поступил хотя бы один запрос по продукту Data Monsoon, при обработке которого потребуются участие отдела QA.
Аргументы	argument1: запрос типа from where select, который возвращает последовательность элементов. argument2: локальная переменная, используемая в запросе. argument3: логическое утверждение об элементе последовательности, результатом которого является значение логического типа.
Возвращает	значение логического типа

AVERAGE ()	
Описание	Возвращает среднее арифметическое аргументов.
Синтаксис функции	AVERAGE(argument1, argument2)
Аргументы	argument1, argument2: число
Возвращает	число

CEILING ()	
Описание	Округляет число в большую сторону.
Синтаксис функции	CEILING(argument1)

CEILING ()	
Аргументы	argument1: число
Возвращает	число

CENTURY ()

Описание Использует в качестве аргумента значение типа дата/время и возвращает числовое значение, соответствующее веку указанной даты в местном часовом поясе.

Синтаксис функции CENTURY(argument1)

Аргументы argument1: дата/время

Возвращает число

CONCAT ()

Описание Берет список строк и объединяет их в новую строку. Возвращает данную строку.

Синтаксис функции CONCAT(LIST(string1, string2, ..., string N))

Аргументы LIST(string1, string2, ..., string N): список строк.

Возвращает строка.

CONTAINS ()

Описание Берет два аргумента одного типа: список строк и значение. Возвращает Истина, если аргумент 2 является частью списка аргумента 1, иначе возвращает Ложь.

Синтаксис функции CONTAINS(argument1, argument2)

Аргументы argument1: список строк

argument2: значение

Возвращает значение логического типа

COUNT ()

Описание Возвращает количество элементов в списке.

Синтаксис функции COUNT(argument1)

Аргументы argument1: список строк

COUNT ()

Возвращает	число
------------	-------

DATETIMEFROMSECONDS ()

Описание	Конвертирует заданное количество секунд в дату.
----------	---

Синтаксис функции	DATETIMEFROMSECONDS(argument1)
-------------------	--------------------------------

Аргументы	argument1: число
-----------	------------------

Возвращает	дата/время
------------	------------

DATETIMEINSECONDS ()

Описание	Конвертирует заданную в качестве аргумента дату в количество секунд, прошедшее с 00:00:00 1 января 1970 года (Unix-время).
----------	--

Синтаксис функции	DATETIMEINSECONDS(argument1)
-------------------	------------------------------

Аргументы	argument1: дата/время
-----------	-----------------------

Возвращает	число
------------	-------

DAY ()

Описание	Использует в качестве аргумента значение типа дата/время и возвращает числовое значение, соответствующее дню месяца указанной даты в местном часовом поясе.
----------	---

Синтаксис функции	DAY(argument1)
-------------------	----------------

Аргументы	argument1: дата/время
-----------	-----------------------

Возвращает	число
------------	-------

DAYTZ ()

Описание	Использует в качестве аргумента значение типа дата/время и возвращает числовое значение, соответствующее дню месяца указанной даты в заданном часовом поясе. Идентификатор часового пояса должен быть задан в формате базы данных часовых поясов IANA.
----------	--

Синтаксис функции	DAYTZ(argument1, argument2)
-------------------	-----------------------------

DAYTZ ()

Аргументы argument1: дата/время
 argument2: строка

Возвращает число

DIVIDE ()

Описание Берет в качестве аргументов два значения числового типа и возвращает результат деления одного на другое.

Синтаксис функции DIVIDE(argument1, argument2)

Аргументы argument1 (делимое): число
 argument2 (делитель): число

Возвращает число

DURATIONFROMSECONDS ()

Описание Конвертирует заданное количество секунд в длительность.

Синтаксис функции DURATIONFROMSECONDS(argument1)

Аргументы argument1: число

Возвращает длительность

DURATIONINSECONDS ()

Описание Конвертирует заданную длительность в количество секунд.

Синтаксис функции DURATIONINSECONDS(argument1)

Аргументы argument1: длительность

Возвращает число

EMPTY ()

Описание Берет в качестве аргумента выражение, результатом которого является некоторое значение, и проверяет данное значение на пустоту.

Синтаксис функции EMPTY(argument1)

EMPTY ()

Аргументы argument1: значение, являющееся результатом выражения, заданного в качестве аргумента, и имеющее один с ним тип.

Возвращает значение логического типа

ENDOFDAY ()

Описание Использует в качестве аргумента значение типа дата/время и возвращает дату и время окончания дня, в пределах которого находится значение аргумента.

Синтаксис функции ENDOFDAY(argument1)

Аргументы argument1: дата/время

Возвращает дата/время

ENDOFMONTH ()

Описание Использует в качестве аргумента значение типа дата/время и возвращает дату и время окончания месяца, в пределах которого находится значение аргумента.

Синтаксис функции ENDOFMONTH(argument1)

Аргументы argument1: дата/время

Возвращает дата/время

ENDOFQUARTER ()

Описание Использует в качестве аргумента значение типа дата/время и возвращает дату и время окончания квартала, в пределах которого находится значение аргумента.

Синтаксис функции ENDOFQUARTER(argument1)

Аргументы argument1: дата/время

Возвращает дата/время

ENDOFWEEK ()

Описание Использует в качестве аргумента значение типа дата/время и возвращает дату и время окончания недели, в пределах которого находится значение аргумента.

Синтаксис функции ENDOFWEEK(argument1)

ENDOFWEEK ()

Аргументы argument1: дата/время

Возвращает дата/время

ENDOFYEAR ()

Описание Использует в качестве аргумента значение типа дата/время и возвращает дату и время окончания года, в пределах которого находится значение аргумента.

Синтаксис функции ENDOFYEAR(argument1)

Аргументы argument1: дата/время

Возвращает дата/время

ENDSWITH ()

Описание Сравнивает два аргумента и возвращает значение Истина, если значение аргумента1 оканчивается на значение аргумента2, в противном случае возвращает Ложь.

Синтаксис функции ENDSWITH(argument1, argument2)

Аргументы argument1, argument2: строка

Возвращает значение логического типа

EQUALS ()

Описание Сравнивает два аргумента и возвращает значение Истина, если аргумент1 равен аргументу2, в противном случае возвращает результат Ложь.

Синтаксис функции EQUALS()

Аргументы argument1, argument2: значение любого типа

Возвращает значение логического типа

FIRSTORDEFAULT ()

Описание Возвращает либо первый элемент последовательности, либо значение по умолчанию для этого типа данных, если первый элемент последовательности не был определён.

FIRSTORDEFAULT ()

Синтаксис функции	<p><code>FIRSTORDEFAULT(sequence)</code></p> <p>Для задания последовательности используйте стандартный запрос ²⁵⁵ <code>from where select</code>.</p> <p>Например, <code>FIRSTORDEFAULT((from i in db->music where i->album == 'Whale Meditation' select i->song))</code>. Данное выражение вернет первую песню из альбома Песни китов.</p>
Аргументы	<p><code>sequence</code>: последовательность однородных элементов любого типа</p>
Возвращает	<p>Тип возвращаемого значения совпадает с типом элементов последовательности.</p>

FLOOR ()

Описание	<p>Округляет число в меньшую сторону.</p>
Синтаксис функции	<p><code>FLOOR(argument1)</code></p>
Аргументы	<p><code>argument1</code>: число</p>
Возвращает	<p>число</p>

FORMAT ()

Описание	<p>Выполняет форматирование аргументов в соответствии с заданным шаблоном и возвращает отформатированную строку.</p>
Синтаксис функции	<p><code>FORMAT(строка задания формата, LIST(argument0, argument1, ..., argument N))</code></p> <p>Формат для <code>argument0</code> задается частью строки, обозначенной как <code>{0}</code>. Формат для <code>argument1</code> определяется частью строки <code>{1}</code> и так далее.</p> <p>Аргументы должны быть записаны в следующем виде: <code>LIST(argument0, argument1, ..., argument N)</code>.</p> <p>Также возможно использование функции <code>FORMAT()</code> для конкатенации строк. Например, <code>FORMAT("{0}, {1}", LIST(\$title,\$assignee))</code>.</p>
Аргументы	<p><code>строка задания формата</code>: строка</p> <p><code>argument0, argument1, ..., argument N</code>: значения любого типа.</p>
Возвращает	<p>строка</p>

GREATER ()

Описание Возвращает логическое значение Истина, если аргумент1 больше аргумента2, в противном случае возвращает результат Ложь.

Синтаксис функции GREATER(argument1, argument2)

Аргументы argument1, argument2: значение логического типа, типа число, строка или дата/время.

Возвращает значение логического типа

GREATEREQ ()

Описание Возвращает логическое значение Истина, если аргумент1 больше либо равен аргументу2, в противном случае возвращает результат Ложь.

Синтаксис функции GREATEREQ(argument1, argument2)

Аргументы argument1, argument2: значение логического типа, типа число, строка или дата/время.

Возвращает значение логического типа

HOUR ()

Описание Использует в качестве аргумента значение типа дата/время и возвращает числовое значение, соответствующее часу указанной даты в местном часовом поясе.

Синтаксис функции HOUR(argument1)

Аргументы argument1: дата и время

Возвращает число

HOURTZ ()

Описание Использует в качестве аргумента значение типа дата/время и возвращает числовое значение, соответствующее часу указанной даты в заданном часовом поясе. Идентификатор часового пояса должен быть задан в формате базы данных часовых поясов IANA.

Синтаксис функции HOURTZ(argument1, argument2)

HOURTZ ()

Аргументы argument1: дата и время
 argument2: строка

Возвращает число

IF ()

Описание Использует в качестве аргументов значения или выражения, результатом которых является некоторое значение. Возвращает аргумент2, если аргумент1 является Истиной, в противном случае возвращает аргумент3.

Синтаксис функции IF(argument1, argumen2, argument3)

Аргументы argument1: значение логического типа или выражение, результатом которого является значение логического типа.
 argument2, argument3: значение любого типа или выражение, результатом которого является значение любого типа.

Возвращает Тип возвращаемого значения тот же, что и у аргументов (argument1, argument2)

INDEXOF ()

Описание Возвращает номер (отсчитываемый от нуля) позиции, на которой располагается искомая строка в исходной строке. Номер позиции может отсчитываться от произвольного положения (опционально).

Синтаксис функции INDEXOF(argument1, argument2, [опциональный argument3])

Аргументы argument1: исходная строка,
 argument2: искомая строка,
 argument3 (опционально): номер позиции, от которой производится отсчет

Возвращает число

INSECONDS ()

Описание Конвертирует значение длительности в секунды и наоборот.

Синтаксис функции INSECONDS(argument1)

INSECONDS ()

Аргументы	argument1: длительность argument 1: число
Возвращает	число, если в качестве аргумента используется значение длительности длительность, если в качестве аргумента используется число

ISOFORMAT ()

Описание	Конвертирует значение типа дата/время в формат ISO.
Синтаксис функции	ISOFORMAT(argument1)
Аргументы	argument1: дата и время
Возвращает	строка

ISTODAY ()

Описание	Возвращает значение Истина, если задана сегодняшняя дата. В противном случае возвращает результат Ложь.
Синтаксис функции	ISTODAY(аргумент1)
Аргументы	аргумент1: дата и время
Возвращает	значение логического типа

JOIN ()

Описание	Объединяет строки с использованием заданного разделителя.
Синтаксис функции	JOIN(argument1, list(argument2, argument3,.. argument N))
Аргументы	argument1: разделитель объединяемых строк, argument2, argument3, ..., argument N (в функции list): объединяемые строки
Возвращает	строка

LENGTH ()

Описание Возвращает количество символов в строке.

Синтаксис функции LENGTH(argument1)

Аргументы argument1: строка

Возвращает число

LESS ()

Описание Возвращает логическое значение Истина, если аргумент1 меньше аргумента2, в противном случае возвращает результат Ложь.

Синтаксис функции LESS(argument1, argument2)

Аргументы argument1, argument2: значение логического типа, число, строка или дата/время.

Возвращает значение логического типа

LESSEQ ()

Описание Возвращает логическое значение Истина, если аргумент1 меньше либо равен аргументу2, в противном случае возвращает результат Ложь.

Синтаксис функции LESSEQ(argument1, argument2)

Аргументы argument1, argument2: значение логического типа, число, строка или дата/время..

Возвращает значение логического типа

LIST ()

Описание Определяет список. Берет перечисление значений одного типа, разделяет их запятыми и возвращает список.

Синтаксис функции LIST(argument1, ..., argument N)

Аргументы argument1, ..., argument N: значения одного типа

Возвращает список

MAKEDATETIME ()

Описание Возвращает дату и время в местном часовом поясе, соответствующие указанным значениям года, месяца, дня, часа, минут и секунд.

Синтаксис функции MAKEDATETIME(argument1, argument2, argument3, argument4, argument5, argument6)

Аргументы

- argument1: числовое значение, соответствующее году.
- argument2: числовое значение, соответствующее месяцу.
- argument3: числовое значение, соответствующее дню.
- argument4: числовое значение, соответствующее часу.
- argument5: числовое значение, соответствующее минутам.
- argument6: числовое значение, соответствующее секундам.

Возвращает дата/время

MAKEDATETIMEZ ()

Описание Возвращает дату и время в заданном часовом поясе, соответствующие указанным значениям года, месяца, дня, часа, минут и секунд. Идентификатор часового пояса должен быть задан в формате базы данных часовых поясов IANA

Синтаксис функции MAKEDATETIMEZ(argument1, argument2, argument3, argument4, argument5, argument6, argument7)

Аргументы

- аргумент1: числовое значение, соответствующее году.
- аргумент2: числовое значение, соответствующее месяцу.
- аргумент3: числовое значение, соответствующее дню.
- аргумент4: числовое значение, соответствующее часу.
- аргумент5: числовое значение, соответствующее минутам.
- аргумент6: числовое значение, соответствующее секундам.
- аргумент7: строка, соответствующая идентификатору часового пояса

Возвращает дата/время

MATCHES ()

Описание Использует в качестве аргументов строку и регулярное выражение. Возвращает логическое значение Истина, если какая-либо часть строки или строка полностью соответствует

MATCHES ()

условию, заданному регулярным выражением. В противном случае возвращает результат Ложь.

Синтаксис функции MATCHES(argument1, argument2)

Аргументы argument1: строка
argument2: строка (регулярное выражение)

Возвращает значение логического типа

MAX ()

Описание Определяет максимальное значение при помощи функции GREATER среди всех элементов списка, заданного в качестве аргумента. Элементы списка могут быть значениями или выражениями, результатом которых является значение. Все элементы списка должны быть одного типа.

Синтаксис функции MAX(argument1)

Аргументы argument1: список, состоящий из значений типа число, длительность или дата/время.

Возвращает Тип возвращаемого значения тот же, что и у элементов списка. Являющегося аргументом.

MIN ()

Описание Определяет минимальное значение при помощи функции LESS среди всех элементов списка, заданного в качестве аргумента. Элементы списка могут быть значениями или выражениями, результатом которых является значение. Все элементы списка должны быть одного типа.

Синтаксис функции MIN(argument1)

Аргументы argument1: список, состоящий из значений типа число, длительность или дата/время.

Возвращает Тип возвращаемого значения тот же, что и у элементов списка, являющегося аргументом.

MINUTE ()

Описание Использует в качестве аргумента значение типа дата/время и возвращает числовое значение, соответствующее минуте указанной даты в местном часовом поясе.

MINUTE ()

Синтаксис функции MINUTE(argument1)

Аргументы argument1: дата/время

Возвращает число

MINUTETZ ()

Описание

Использует в качестве аргумента значение типа дата/время и возвращает числовое значение, соответствующее минуте указанной даты в заданном часовом поясе. Идентификатор часового пояса должен быть задан в формате базы данных часовых поясов IANA.

Синтаксис функции MINUTETZ(argument1, argument2)

Аргументы argument1: дата/время
 argument2: строка

Возвращает число

MONTH ()

Описание

Использует в качестве аргумента значение типа дата/время и возвращает числовое значение, соответствующее месяцу указанной даты в местном часовом поясе.

Синтаксис функции MONTH(argument1)

Аргументы argument1: дата/время

Возвращает число

MONTHTZ ()

Описание

Использует в качестве аргумента значение типа дата/время и возвращает числовое значение, соответствующее месяцу указанной даты в заданном часовом поясе. Идентификатор часового пояса должен быть задан в формате базы данных часовых поясов IANA.

Синтаксис функции MONTHTZ(argument1, argument2)

Аргументы argument1: дата/время
 argument2: строка

MONTHTZ ()

Возвращает число

NOT ()

Описание

Использует в качестве аргумента значение логического типа или выражение, результатом которого является значение логического типа. Выполняет операцию логического отрицания над аргументом.

Синтаксис функции NOT(argument1)

Аргументы argument1: логический

Возвращает значение логического типа

NOTEQUALS ()

Описание

Сравнивает два аргумента и возвращает логическое значение Истина, если аргумент1 не равен аргументу2, в противном случае возвращает результат Ложь.

Синтаксис функции NOTEQUALS(argument1, argument2)

Аргументы argument1, argument2: значение любого типа

Возвращает значение логического типа

NOTMATCHES ()

Описание

Использует в качестве аргументов строку и регулярное выражение. Возвращает логическое значение Истина, если часть строки или строка полностью не соответствуют условию, заданному регулярным выражением. В противном случае возвращает результат Ложь.

Синтаксис функции NOTMATCHES(argument1, argument2)

Аргументы argument1: строка
argument2: строка (регулярное выражение)

Возвращает значение логического типа

NOW ()

Описание Возвращает текущую дату и время

NOW ()

Синтаксис функции NOW()

Аргументы аргументы не требуются

Возвращает текущую дату и время

OR ()

Описание

Берет два значения логического типа или два выражения, результатом которых являются значения логического типа, и возвращает результат логического сложения этих значений.

Синтаксис функции OR(argument1, argument2)

Аргументы

argument1, argument2: значение логического типа или выражение, результатом которого является значение логического типа.

Возвращает значение логического типа

PRODUCT ()

Описание

Берет в качестве аргументов два значения числового типа и возвращает результат умножения одного на другое.

Синтаксис функции PRODUCT(argument1, argument2)

Аргументы

argument1: число
argument2: число

Возвращает число

QUOTIENT ()

Описание

Использует в качестве аргументов два значения числового типа и возвращает целочисленный результат деления одного на другое.

Синтаксис функции QUOTIENT(argument1, argument2)

Аргументы

argument1(делимое): число
argument2(делитель): число

Возвращает целое число

REMAINDER ()

Описание	Использует в качестве аргументов два значения числового типа и возвращает остаток от деления одного значения на другое.
Синтаксис функции	REMAINDER(argument1, argument2)
Аргументы	argument1(делимое): число argument2(делитель): число
Возвращает	число

REPLACE ()

Описание	Заменяет все экземпляры заданной строки в исходной строке на новое значение.
Синтаксис функции	REPLACE(argument1, argument2, argument3)
Аргументы	argument1: исходная строка argument2: заменяемая строка argument3: новая строка
Возвращает	строка

ROUND ()

Описание	Округляет число к ближайшему целому числу.
Синтаксис функции	ROUND(argument1)
Аргументы	argument1: число
Возвращает	число

SECOND ()

Описание	Использует в качестве аргумента значение типа дата/время и возвращает числовое значение, соответствующее секундам указанной даты в местном часовом поясе.
Синтаксис функции	SECOND(argument1)
Аргументы	argument1: дата/время
Возвращает	число

SECONDTZ ()

Описание	Использует в качестве аргумента значение типа дата/время и возвращает числовое значение, соответствующее секундам указанной даты в заданном часовом поясе. Идентификатор часового пояса должен быть задан в формате базы данных часовых поясов IANA.
Синтаксис функции	SECONDTZ(argument1, argument2)
Аргументы	argument1: дата/время argument2: строка
Возвращает	число

STARTOFDAY ()

Описание	Использует в качестве аргумента значение типа дата/время и возвращает дату и время начала дня, в пределах которого находится значение аргумента.
Синтаксис функции	STARTOFDAY(argument1)
Аргументы	argument1: дата/время
Возвращает	дата/время

STARTOFMONTH ()

Описание	Использует в качестве аргумента значение типа дата/время и возвращает дату и время начала месяца, в пределах которого находится значение аргумента.
Синтаксис функции	STARTOFMONTH(argument1)
Аргументы	argument1: дата/время
Возвращает	дата/время

STARTOFQUARTER ()

Описание	Использует в качестве аргумента значение типа дата/время и возвращает дату и время начала квартала, в пределах которого находится значение аргумента.
Синтаксис функции	STARTOFQUARTER(argument1)

STARTOFQUARTER ()

Аргументы argument1: дата/время

Возвращает дата/время

STARTOFWEEK ()

Описание Использует в качестве аргумента значение типа дата/время и возвращает дату и время начала недели, в пределах которой находится значение аргумента.

Синтаксис функции STARTOFWEEK(argument1)

Аргументы argument1: дата/время

Возвращает дата/время

STARTOFYEAR ()

Описание Использует в качестве аргумента значение типа дата/время и возвращает дату и время начала года, в пределах которой находится значение аргумента.

Синтаксис функции STARTOFYEAR(argument1)

Аргументы argument1: дата/время

Возвращает дата/время

STARTSWITH ()

Описание Сравнивает два аргумента и возвращает значение Истина, если значение аргумента1 начинается с значения аргумента2, в противном случае возвращает Ложь.

Синтаксис функции STARTWITH(argument1, argument2)

Аргументы argument1, argument2: строка

Возвращает значение логического типа

SUBSTRING ()

Описание Извлекает из строки подстроку определенной длины начиная с заданной позиции (задание длины опционально).

SUBSTRING ()

Синтаксис функции	SUBSTRING(argument1, argument2, [опциональный argument3])
Аргументы	argument1: строка argument2: номер позиции подстроки (отсчитываемый от нуля) argument 3(опционально): количество символов в подстроке
Возвращает	строка

SUBTRACT ()

Описание Вычитает значение аргумента2 из значения аргумента1.

Синтаксис функции SUBTRACT(argument1, argument2)

Аргументы
argument1: число; argument2: число
argument1: длительность; argument2: длительность
argument1: дата и время; argument2: длительность.
argument1: дата и время; argument2: дата и время

Возвращает
число
длительность
дата/время
длительность

SUM ()

Описание Суммирует все значения списка, используемого в качестве аргумента. Конкатенация строк не поддерживается.

Синтаксис функции SUM(argument1)

Аргументы
argument1: список, состоящий из значений типа число или длительность

Возвращает
Тип возвращаемого значения тот же, что и у элементов списка, являющегося аргументом.

TOLOWER ()

Описание Переводит строку в нижний регистр (к прописным буквам).

Синтаксис функции TOLOWER(argument1)

TOLOWER ()

Аргументы argument1: строка

Возвращает строка

TOUPPER ()

Описание Переводит строку в верхний регистр (к заглавным буквам).

Синтаксис функции TOUPPER(argument1)

Аргументы argument1: строка

Возвращает строка

USER ()

Описание Возвращает текущего пользователя или идентификатор текущего пользователя, в зависимости от того, к какому типу данных относится поле. Если вычисляемый атрибут принадлежит типу данных Пользователь, функция возвращает текущего пользователя. Если вычисляемый атрибут текстового типа, функция возвращает идентификатор текущего пользователя.

Синтаксис функции USER()

Аргументы аргументы не требуются

Возвращает пользователь или строка

WORKDAYS ()

Описание Возвращает дату на основании количества рабочих дней с определенной даты, с поправкой на праздники и фактические выходные дни.

Если временная зона рабочего офиса отличается от зоны сервера Comindware, используйте функцию WORKDAYSTZ().

Синтаксис функции WORKDAYS(argument1, argument2, [опционально list (argument3, argument4, ..., argument M)])

WORKDAYS ()

Аргументы	argument1: дата/время начала работы, argument2: количество рабочих дней, argument3, argument4, ..., argument M (в опциональной функции list): праздники и исключения из выходных дней, т. е. если в списке указан рабочий день, он считается праздником, и наоборот, если в списке выходной день, он считается рабочим.
Возвращает	дата/время

WORKDAYSTZ ()

Описание	Возвращает дату на основании количества рабочих дней с определенной даты, с поправкой на праздники и фактические выходные дни, а также на временную зону рабочего офиса.
Синтаксис функции	WORKDAYSTZ(argument1, argument2, list (argument3, argument4, .., argument M), argument N)
Аргументы	argument1: дата и время начала работы argument2: количество рабочих дней argument3, argument4, ..., argument M (в функции list): праздники и исключения из выходных дней, т. е. если в списке указан рабочий день, он считается праздником, и наоборот, если в списке выходной день, он считается рабочим argument N: временная зона рабочего офиса в формате Olson db
Возвращает	дата/время

WORKDAYSDURATION ()

Описание	Возвращает продолжительность работы между двумя датами, на основании предположения, что рабочий день начинается в 00:00 и длится 24 часа. Если временная зона рабочего офиса отличается от зоны сервера Comindware, используйте функцию WORKDAYSDURATIONTZ ().
Синтаксис функции	WORKDAYSDURATION(argument1, argument 2, [опционально list (argument3, argument4, ..., argument M)])

WORKDAYSDURATION ()

Аргументы	argument1: дата и время начала работы, argument2: дата и время окончания работы, argument3, argument4, ..., argument M (в опциональной функции list): праздники и исключения из выходных дней, т. е. если в списке указан рабочий день, он считается праздником, и наоборот, если в списке выходной день, он считается рабочим.
Возвращает	длительность

WORKDAYSDURATIONTZ ()

Описание	Возвращает продолжительность работы между двумя датами с поправкой на временную зону рабочего офиса, на основании предположения, что рабочий день начинается в 00:00 и длится 24 часа.
Синтаксис функции	WORKDAYSDURATIONTZ(argument1, argument2, list (argument3, Argument4, .., argument M), argument N)
Аргументы	argument1: дата и время начала работы, argument2: дата и время окончания работы, argument3, argument4, ..., argument M (в функции list): праздники и исключения из выходных дней, т. е. если в списке указан рабочий день, он считается праздником, и наоборот, если в списке выходной день, он считается рабочим, argument N: временная зона рабочего офиса в формате Olson db.
Возвращает	длительность

WORKHOURS ()

Описание	Возвращает дату и время на основании количества рабочих часов с определенного момента времени, с поправкой на праздники и фактические выходные дни. Если временная зона рабочего офиса отличается от зоны сервера Comindware, используйте функцию WORKHOURSTZ ().
Синтаксис функции	WORKHOURS(argument1, argument2, argument3, argument 4, [опционально list (argument5, argument6, .., argument M)])

WORKHOURS ()

Аргументы	<p>argument1: дата/время начала работы, argument2: количество рабочих часов, argument3: время начала рабочего дня, argument4: длительность рабочего дня (должна быть меньше разницы между 24 ч и временем начала рабочего дня), argument5, argument6, ... argument M (в опциональной функции list): праздники и исключения из выходных дней, т. е. если в списке указан рабочий день, он считается праздником, и наоборот, если в списке выходной день, он считается рабочим.</p>
Возвращает	дата/время

WORKHOURSURATION ()

Описание	<p>Возвращает продолжительность работы между двумя датами при условии, что рабочий день начинается в заданное время и имеет определенную длительность.</p> <p>Если временная зона рабочего офиса отличается от зоны сервера Comindware, используйте функцию WORKHOURSURATIONTZ ().</p>
Синтаксис функции	<p>WORKHOURSURATION(argument1, argument2, argument3, argument 4, [опционально list (argument5, argument6, ..., argument M)])</p>
Аргументы	<p>argument1: дата/время начала работы argument2: дата/время окончания работы argument3: время начала рабочего дня argument4: длительность рабочего дня (должна быть меньше разницы между 24 ч и временем начала рабочего дня) argument5, argument6, .., argument M (в опциональной функции list): праздники и исключения из выходных дней, т. е. если в списке указан рабочий день, он считается праздником, и наоборот, если в списке выходной день, он считается рабочим.</p>
Возвращает	длительность

WORKHOURSURATIONTZ

()

Описание	Возвращает продолжительность работы между двумя датами с поправкой на временную зону рабочего офиса, при условии, что рабочий день начинается в заданное время и имеет определенную длительность.
Синтаксис функции	WORKHOURSURATIONTZ(argument1, argument2, argument3, argument4, list (argument5, argument6, .., argument M), argument N)
Аргументы	argument1: дата/время начала работы argument2: дата/время окончания работы argument3: время начала рабочего дня argument4: длительность рабочего дня (должна быть меньше разницы между 24 ч и временем начала рабочего дня) argument5, argument6, .., argument M (в функции list): праздники и исключения из выходных дней, т. е. если в списке указан рабочий день, он считается праздником, и наоборот, если в списке выходной день, он считается рабочим argument N: временная зона рабочего офиса в формате Olson DB.
Возвращает	длительность

WORKHOURSTZ ()

Описание	Возвращает дату и время на основании количества рабочих часов с определенного момента времени, с поправкой на праздники и фактические выходные дни, а также на временную зону рабочего офиса.
Синтаксис функции	WORKHOURSTZ(argument1, argument2, argument3, argument4, list(argument5, argument6, .., argument M), argument N)
Аргументы	argument1: дата/время начала работы, argument2: количество рабочих часов, argument3: время начала рабочего дня, argument4: длительность рабочего дня (должна быть меньше разницы между 24 ч и временем начала рабочего дня), argument5, argument6, argument M (в функции list): праздники и исключения из выходных дней, т. е. если в

WORKHOURSTZ ()

списке указан рабочий день, он считается праздником, и наоборот, если в списке выходной день, он считается рабочим,

argument N: временная зона рабочего офиса в формате Olson DB.

Возвращает дата/время

YEAR ()

Описание

Использует в качестве аргумента значение типа дата/время и возвращает числовое значение, соответствующее году указанной даты в местном часовом поясе.

Синтаксис функции YEAR(argument1)

Аргументы argument1: дата/время

Возвращает число

YEARTZ ()

Описание

Использует в качестве аргумента значение типа дата/время и возвращает числовое значение, соответствующее году указанной даты в заданном часовом поясе.

Идентификатор часового пояса должен быть задан в формате базы данных часовых поясов IANA.

Синтаксис функции YEARTZ(argument1, argument2)

Аргументы argument: дата/время

argument: строка

Возвращает число

5.2.5 Литералы

DATE ()

Описание

Объявляет дату в ISO формате.

Значения даты и времени расположены в порядке от наиболее значимых к наименее значимым: год, месяц (или неделя), день, час, минута, секунда, доля секунды.

Формат

Дата: ГГГГ-ММ-ДД

DATE ()

Время: чч:мм:сс

Синтаксис DATE(argument1)

Аргументы argument1: строка

Возвращает дата/время

DURATION ()

Описание Объявляет длительность в ISO формате.

Формат P[n]Y[n]M[n]DT[n]H[n]M[n]S
Заглавные буквы P, Y, M, W, D, T, H, M, S обозначают элементы даты и времени.

Синтаксис DURATION(argument1)

Аргументы argument1: строка

Возвращает длительность

ID ()

Описание Объявляет объект.

Синтаксис ID(argument1)

Аргументы argument1: строка

Возвращает идентификатор объекта

5.2.6 Операторы

В данном разделе приведен синтаксис и использование операторов в языке выражений Comindware Business Application Platform.

from

Описание Выражение запроса должно начинаться с оператора from. Выражение запроса также может содержать начинающиеся с from подзапросы. Оператор from состоит из следующих компонентов:

from

Источник данных, по которому выполняется запрос (например, шаблон записи, база данных пользователей и т.д.)

Локальная переменная, которая представляет каждый элемент источника данных (например, запись в шаблоне или учетную запись пользователя)

Синтаксис `from переменная in источник_данных`

Аргументы
переменная: локальная переменная
источник_данных: шаблон записи или база данных пользователей

where

Описание
Оператор `where` используется в выражении запроса и определяет, какие элементы из источника данных будут возвращены. Он применяет логическое условие (предикат) к каждому элементу источника (представленному локальной переменной) и возвращает те элементы, в отношении которых условие выполняется. Одно выражение запроса может содержать несколько операторов `where`, а один оператор `where` может включать в себя несколько предикатов.

Синтаксис `Where предикат`

Аргументы
Предикат: значение логического типа или выражение, результатом которого является значение логического типа.

select

Описание
В выражении запроса оператор `select` определяет, какое свойство объекта будет результатом выполнения запроса. В конечном результате учитывается как работа предыдущих операторов, так и любые выражения в самом операторе `select`.

Синтаксис `Select свойство_объекта`

Аргументы
Свойство_объекта: атрибут шаблона записи, поле в базе данных и т.д.

Следующие слова не могут использоваться в операторах в качестве локальных переменных:

- `item`
- `source`
- `db`
- `group`
- любое слово, начинающееся с подчеркивания (`_`)



5.2.7 Вызов связанных данных

Для того чтобы вызвать данные связанного шаблона записи, используя ссылку на этот шаблон, присоедините постфикс Ref к системному имени ссылки, добавьте символ -> и системное имя вызываемого атрибута связанного шаблона записи.

В приведенном ниже примере показано, как использовать постфикс Ref:

Шаблон записи	Атрибуты
Car (Машина)	Driver (Водитель) – ссылка на шаблон записи Staff (Персонал)
Staff (персонал)	Name (Имя) – имя водителя

Для того чтобы вызвать имя водителя в записи «Машина», используйте следующее выражение: `$DriverRef->Name`. Количество переходов по ссылкам неограничено.

Имейте в виду, что выражение `$Driver` возвращает ИД водителя, а выражение `$Driver->Name` неработоспособно.

5.3 Примеры использования

5.3.1 Задание вычисляемого атрибута

В данном примере настраивается автоматическое назначение клиента для текущего пользователя. Существует база данных клиентов и менеджеры компании. Каждому менеджеру привязывается клиент.

Системные имена шаблонов записи и атрибутов, задействованных в примере, приведены в таблице:

Шаблон записи	Атрибуты
Request (Заявка)	ClientForCurrentUser – клиент
Ср (Контактные лица)	Ср – деловой партнер User – пользователь
Clients (Деловые партнеры)	Name – наименование делового партнера (ФИО)

Атрибут **ClientForCurrentUser** шаблона записи **Request** вычисляется на основании данных, хранящихся в шаблоне записи **Ср**.

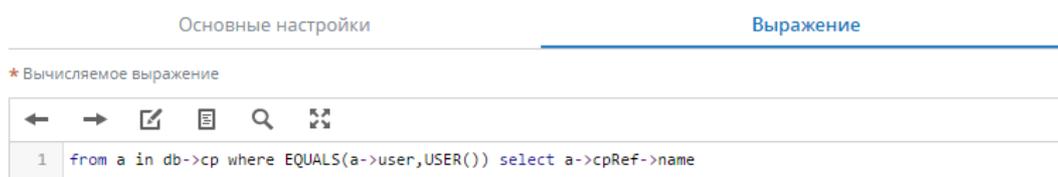
В выражении будут использованы следующие Функции и Операторы:

- функция **EQUALS()** для выбора записи, которая соответствует текущему пользователю,
- функция **USER()** для определения текущего пользователя,
- оператор **from** для перечисления всех записей из шаблона записи **Ср**,
- оператор **where** для выбора записей, которые соответствуют текущему пользователю,
- оператор **select** для выбора значений атрибута **Name**.

Пошаговая инструкция:

1. Создайте и настройте шаблоны записей ¹⁹ и атрибуты ²²
2. В шаблоне записи **Request** откройте созданный атрибут **ClientForCurrentUser**.
3. Отметьте флаг **Вычисляемый** и перейдите на вкладку **Выражение**
4. Напишите выражение

Изменить атрибут



5. Запись выражения атрибута

Синтаксис

Элемент	Значение
from a in db->cp	Объявление локальной переменной a. Объявление шаблона записи cp в качестве источника данных: db->cp. Перечисление всех записей из шаблона записи cp.
where EQUALS(a->user,USER())	Выбор записи, значение User в которой соответствует текущему пользователю. Функция User() возвращает текущего пользователя, если вычисляемый атрибут принадлежит типу данных Пользователь.
select a->cpRef->Name	Выбор значений атрибута Name в соответствующих записях и передача этих значений в качестве аргумента. Постфикс Ref в атрибуте cp используется для того, чтобы вызвать данные связанного шаблона записи Name.

5.3.2 Задание фильтра

В данном примере в форму Заявки будет добавлена ссылка на контракт.

Системные имена шаблонов записи и атрибутов, задействованных в примере, приведены в таблице:

Шаблон записи	Атрибуты
Request (Заявка)	contracts – контракты client_code – клиент
Contracts (Контракты)	client – клиент

В выражении будут использованы следующие Функции и Операторы:

- функция **EQUALS()** для выбора записи, которая соответствует клиенту в заявке,
- оператор **from** для перечисления всех записей из шаблона записи **contracts**,
- оператор **where** для выбора записей, которые соответствуют условию,
- оператор **select** для выбора значений атрибута.

Пошаговая инструкция:

1. Создайте и настройте шаблоны записей ¹⁹ и атрибуты ²²
2. В шаблоне записи **Request** откройте **Формы**.
3. Добавьте атрибут **contracts** на форму.
4. Добавьте фильтр – введите выражение



```
1 from a in db->contracts where EQUALS($client_code,a->client) select a->id
```

Синтаксис

Элемент	Значение
from a in db->contracts	Объявление локальной переменной a. Объявление шаблона записи contract в качестве источника данных: db->contracts. Перечисление всех записей из шаблона записи contract.
where EQUALS(\$client_code, a->client)	Выбор записи, в которой значение Client_code текущей записи соответствует значению атрибута client в справочнике Контрактов.
select a->id	Выбор идентификаторов необходимых учетных записей.

5.3.3 Настройка правил

В данном примере настраивается правило для записи – права доступа.

Системные имена шаблонов записи и атрибутов, задействованных в примере, приведены в таблице:

Шаблон записи	Атрибуты
Ship_raspor (Распоряжение на отправку)	CI_SAP – клиент
Client (Деловые партнеры)	Ср – контактные лица
СР (Контактные лица)	User – пользователь

В выражении будут использованы следующие Функции и Операторы:

- функция **NOT()** для выполнения логического отрицания,
- функция **EMPTY()** для проверки значения на пустоту,
- функция **IF()** проверяет выполнение условия в аргументе,
- оператор **from** для перечисления всех записей из шаблона записи **СР**,
- оператор **where** для выбора записей, которые соответствуют условию,
- оператор **select** для выбора значений атрибута **User**.

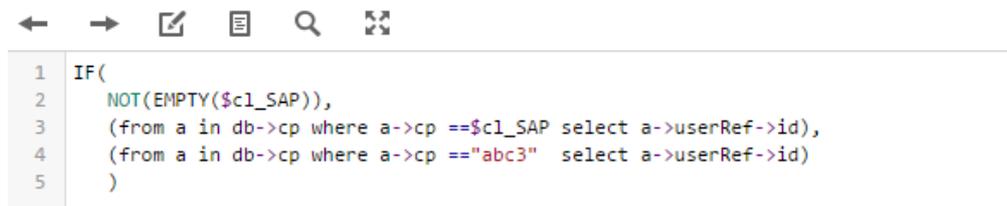
Пошаговая инструкция:

1. Создайте и настройте шаблоны записей ¹⁹ и атрибуты ²²

2. В шаблоне записи **Ship_raspor** перейдите в раздел «Права доступа».

3. Добавьте правило

4. Напишите выражение.



```
1 IF(
2   NOT(EMPTY($cl_SAP)),
3   (from a in db->cp where a->cp ==$cl_SAP select a->userRef->id),
4   (from a in db->cp where a->cp =="abc3" select a->userRef->id)
5  )
```

5. Запись выражения атрибута

Синтаксис

Элемент	Значение
IF(NOT(EMPTY(\$cl_SAP))	<ul style="list-style-type: none">• Проверка условия, что атрибут cl_SAP не пустой• Если условие возвращает Истина, то аргумент принимает значение, которое является результатом выполнения выражения 1, иначе аргумент принимает значение, которое является результатом выполнения выражения 2• \$ перед атрибутом указывает на то, что атрибут находится в текущей записи
Выражение 1: from a in db->cp	<ul style="list-style-type: none">• Объявление локальной переменной a.• Объявление шаблона записи cp в качестве источника данных: db->cp.• Перечисление всех записей из шаблона записи cp.
where a->cp ==\$cl_SAP	Выбор записи, в которой значение атрибута CP равно значению атрибута cl_SAP
select a->userRef->id	<ul style="list-style-type: none">• Выбор идентификаторов необходимых учетных записей.• Постфикс Ref в атрибуте user используется для того, чтобы вызвать данные связанного шаблона записи ID.
Выражение 2: from a in db->cp where a->cp =="abc3" select a->userRef->id	Выбор записи, в которой значение атрибута CP равно abc3

В данном примере настраивается правило для формы.

Системные имена шаблонов записи и атрибутов, задействованных в примере, приведены в таблице:

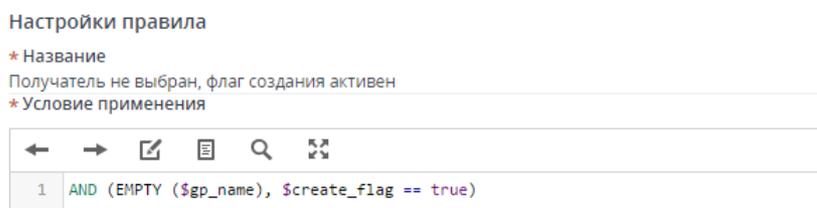
Шаблон записи	Атрибуты
Request (Заявка)	gp_name – получатель create_flag – флаг «ввести данные нового клиента»

В выражении будут использованы следующие Функции:

- функция **AND()** для выполнения логического сложения,
- функция **EMPTY()** для проверки значения на пустоту.

Пошаговая инструкция:

1. Создайте и настройте шаблоны записей ¹⁹ и атрибуты ²²
2. В шаблоне записи **Ship_raspor** перейдите в раздел «**Формы**».
3. Откройте форму и перейдите в раздел «**Правила для формы**».
4. Добавьте правило.
5. Напишите выражение.



6. Запись выражения для правила

Синтаксис

Элемент	Значение
AND (EMPTY (\$gp_name), \$create_flag == true)	Логическое сложение – при отсутствии получателя флаг «ввести данные нового клиента» активен \$ перед атрибутом указывает на то, что атрибут находится в текущей записи

5.3.4 Задание условия

В данном примере добавим условие на выполнение операции. В отчете «Заявка» существуют позиции заявки. Необходимо добавить операцию «Добавить новую позицию». Позиция добавляется только при условии, что заявка находится в статусе «Новая», либо «В корзине».

Системные имена шаблонов записи и атрибутов, задействованных в примере, приведены в таблице:

Шаблон записи	Атрибуты
Request (Заявка)	Status – статус заявки
Request_status (Статусы заявок)	Status_code – код статуса

Применение правила

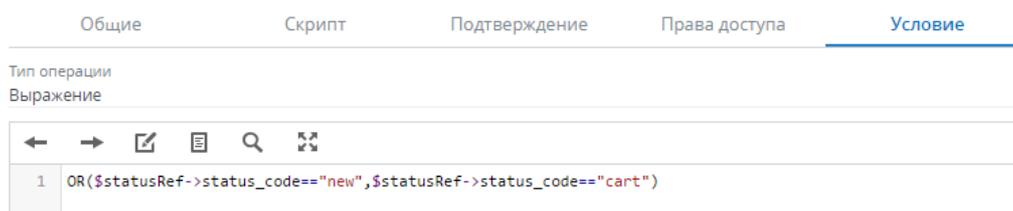
В выражении будет использована следующая Функция:

- функция **OR()** для выполнения логического сложения аргументов.

Пошаговая инструкция:

1. Создайте и настройте шаблоны записей ¹⁹ и атрибуты ²²
2. В шаблоне записи **Request** перейдите в раздел «**Операции**».
3. Создайте операцию.⁸⁷
4. Перейдите на вкладку **Условие** и напишите выражение.

Редактирование операции



5. Запись выражения для операции

Синтаксис

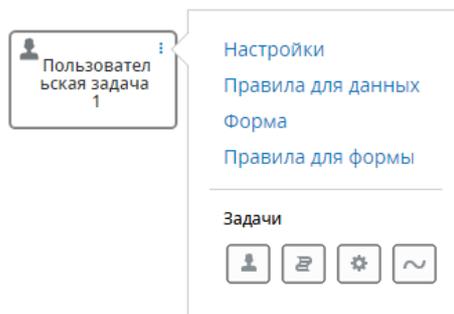
Элемент	Значение
OR(\$statusRef->status_code=="new", \$statusRef->status_code=="cart")	OR() – условие выполняется при значении аргумента равном либо "new", либо "cart" \$ перед атрибутом указывает на то, что атрибут находится в текущей записи Постфикс Ref в атрибуте status используется для того, чтобы вызвать данные связанного шаблона записи status_code.

5.3.5 Назначение исполнителя задачи

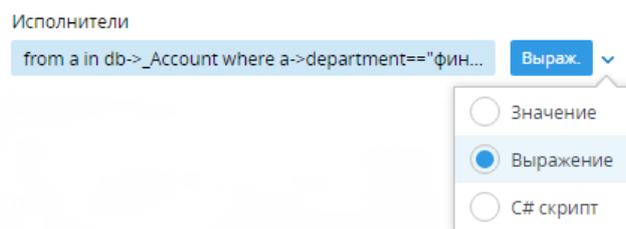
В данном примере с помощью простого запроса "from in where select" настраивается доступ к пользовательской задаче для всех пользователей финансового департамента.

Пошаговая инструкция

1. Добавьте пользователей в финансовый департамент.
2. Создайте и настройте простой процесс, включающий в себя пользовательскую задачу.
3. Перейдите в раздел «**Настройки** задачи».



4. Настройки пользовательской задачи
5. В поле Исполнители выберите **Выражение**



6. Запись выражения в поле исполнители
7. Введите следующее выражение

```
from a in db->_Account where a->department=="финансовый департамент" select a->id
```

8. Пример выражения

Синтаксис

Элемент	Значение
	Объявление локальной переменной a.
from a in db->cp	Объявление системной базы данных _Account в качестве источника данных: db->_Account. Перечисление всех записей из базы данных _Account.
where a->department=="финансовый департамент"	Отбор учетных записей со строкой «финансовый департамент» в поле департамент.
select a->id	Выбор идентификаторов необходимых учетных записей.

6. Интеграция через API

В этой главе:



[Введение в Web API](#) ²⁶⁷

Основы API-интеграции Comindware Business Application Platform с веб-службами.



RESTful Web Api (публичные методы) ²⁶⁷

Использование публичных методов Web API для управления данными.



System Core Api (системные методы) ²⁶⁹

Информация о системных методах Web API.



Solution Api (пользовательские методы) ²⁷³

Использование пользовательских методов Web API для управления данными.

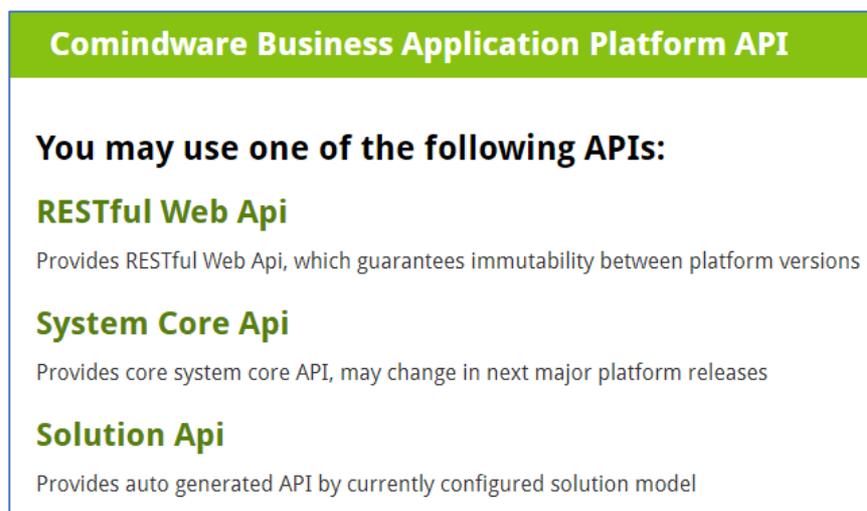
6.1 Введение в Web API

Помимо встроенных инструментов, обеспечивающих взаимодействие процессов с внешними веб-службами через настраиваемые каналы связи, Comindware Business Application Platform может взаимодействовать с внешними ресурсами через Web API.

- Web API реализован в интерфейсе [Swagger](#).
- Все методы Web API получают и отправляют данные в формате JSON.
- Все методы Web API поддерживают базовую проверку подлинности.
- Существуют публичные, системные и пользовательские методы Web API. Пользовательские методы уникальны для каждого шаблона записи.

Перейдите по адресу <http://your-host/docs>, где your-host – адрес вашего домена.

Comindware Business Application Platform API включает в себя следующие методы:



The image shows a screenshot of the Comindware Business Application Platform API documentation. It features a green header with the text "Comindware Business Application Platform API". Below the header, the text reads "You may use one of the following APIs:". There are three entries listed:

- RESTful Web Api**: Provides RESTful Web Api, which guarantees immutability between platform versions
- System Core Api**: Provides core system core API, may change in next major platform releases
- Solution Api**: Provides auto generated API by currently configured solution model

Рис. 295 Методы Comindware Business Application Platform API

6.2 RESTful Web Api (публичные методы)

Публичные методы универсальны и неизменны между версиями Comindware Business Application Platform.

Каждый метод справа в строке имеет собственное описание, применимое к группе методов, в которой он состоит.

Comindware Platform WEB API v1.0		
Account		
GET	/webapi/Account	Get all accounts
POST	/webapi/Account	Create account
PUT	/webapi/Account	Edit account
DELETE	/webapi/Account/{accountId}	Delete account
GET	/webapi/Account/{accountId}	Get account by id
Attribute		
Backup		
CaseTemplate		
Dataset		
Document		
DocumentExportTemplate		
Form		
ProcessTemplate		
Record		
Records		
RecordTemplate		
Toolbar		
Transfer		
UserCommand		

Рис. 296 Методы Comindware Business Application Platform API

Рассмотрим примеры на один из методов в некоторых группах методов:

- Account – шаблон записей Люди. Для создания пользователя в платформе отправьте запрос POST на адрес <http://your-host/webapi/Account>. В теле запроса укажите названия атрибутов и их значения в формате json.
- Attribute – атрибуты шаблона записи. Для того чтобы отправить атрибут в архив отправьте запрос PUT на адрес <http://your-host/webapi/Attribute/Category/title/Disable>, где «Category» - системное имя шаблона записи, «title» - системное имя атрибута.
- Backup – резервное копирование. Для того чтобы удалить созданные сессии необходимо узнать id сессий. Для этого отправьте запрос GET на адрес <http://your-host/webapi/Backup/Session/>. Тело запроса должно содержать атрибут {"id": "backupSession.1"}. Для удаления сессии отправьте запрос DELETE на адрес <http://your-host/webapi/Backup/Session/>, в теле запроса укажите массив id сессий, например ["backupSession.1"].
- CaseTemplate – шаблон кейса.
- Dataset – список шаблона записи.

- Document – прикрепленный документ в шаблоне записи. Для того чтобы узнать id документа необходимо первоначально получить все данные по экземпляру записи. Отправьте запрос GET на адрес <http://your-host/webapi/Record/1>, где «1» — id экземпляра записи. Тело ("Response body") запроса должно содержать атрибут op.10, где «op.10» – системное имя атрибута типа документ, и его значение: {"op.10": ["document.2"]}. Для того чтобы скачать документ отправьте запрос GET на адрес <http://your-host/webapi/Document/document.2/Content>, где «document.2» - id прикрепленного файла. Тело ("Response body") запроса должно содержать ссылку на скачивание файла.
- DocumentExportTemplate – свойство шаблона записи (раздел **Шаблоны**).
- Form – свойство шаблона записи (раздел **Формы**).
- ProcessTemplate – шаблон процесса.
- Record – экземпляр записи. Запросы, связанные с одним экземпляром записи.
- Records – экземпляры записи. Запросы, связанные с несколькими экземплярами записи
- RecordTemplate – шаблон записи.
- Toolbar – панель инструментов.
- Transfer – операция импорта шаблона записи.
- UserCommand – операции определенного шаблона записи.

6.3 System Core Api (системные методы)

Системные методы универсальны и неизменны в рамках одной и той же версии Comindware Business Application Platform.

- Список системных методов можно посмотреть по адресу <http://your-host/Docs/SystemApi>

Comindware Business Application Platform API	
System Core API	
Object	Show/Hide List Operations Expand Operations
ObjectApp	Show/Hide List Operations Expand Operations
POST	/TeamNetwork/ObjectAppService/List
POST	/TeamNetwork/ObjectAppService/ListAll
POST	/TeamNetwork/ObjectAppService/ListAppsForAccount
POST	/TeamNetwork/ObjectAppService/Get
POST	/TeamNetwork/ObjectAppService/GetByAlias
POST	/TeamNetwork/ObjectAppService/GetByObject

Рис. 297 Системные методы

- Для доступа к Swagger-описанию системных методов перейдите по ссылке <http://your-host/api/public/system/>

Для вызова системного метода отправьте запрос HTTP-POST на адрес метода (приведен в swagger-описании). Например, для создания шаблона записи нужно отправить запрос HTTP-POST на адрес <http://your-host/api/public/system/TeamNetwork/ObjectAppService/Create>, указав системное имя шаблона записи в теле ("body") запроса (например, "Car"). В ответ на запрос платформа отправит id созданного шаблона записи (например, "oa.1").

6.3.1 Настройка связи «многие ко многим» в коллекции

Пример настройки связи N-N:

1. Создайте атрибут-коллекция, со ссылкой на другой шаблон записи.

Шаблоны > Заявка на командировку > Атрибуты

+ Добавить

Новый атрибут

Основные настройки | Выражение

Тип данных
Коллекция

Имя | **Системное имя** ?
Командировки | Komandirovki

Описание
Введите текст

Целевой шаблон
Командировка

Имя для новой ссылки | **Системное имя для новой ссылки**
Заявка на командировку | Zayavkanakomandirovku

Удалять связанный объект

Отслеживание изменений

Отмена | Сохранить

Рис. 298 Создание коллекции

Шаблоны > Заявка на командировку > Атрибуты

+ Добавить

Ид	Системное имя ↓	Имя	Тип данных	Формат отображ...	Вычисляет
<input type="checkbox"/> оп.101	Komandirovki	Командировки	Коллекция	Не предусмотрен	

Рис. 299 ID атрибута коллекции

2. Откройте в другой вкладке браузер следующую страницу:

<http://your-host/Docs/SystemApi/#!/ObjectApp/TeamNetworkObjectAppServiceBindProperties>

Comindware Platform API

localhost:8082/Docs/SystemApi/#!/ObjectApp/TeamNetworkObjectAppServiceBindProperties

POST /TeamNetwork/ObjectAppService/CreateProperty

POST /TeamNetwork/ObjectAppService/CreateProperty1

POST /TeamNetwork/ObjectAppService/CreateAccountProperty

POST /TeamNetwork/ObjectAppService/CreateForwardProperty

POST /TeamNetwork/ObjectAppService/CreateBackwardProperty

POST /TeamNetwork/ObjectAppService/BindProperties

Response Class (Status 200)
string

Response Content Type application/json

Parameters

Parameter	Value	Description	Parameter Type	Data Type
parameters	<input type="text"/>		body	Model Example Value

```
{
  "op1": "string",
  "op1MultiValue": true,
  "op2": "string",
  "op2MultiValue": true
}
```

Parameter content type: application/json

Try it out!

POST /TeamNetwork/ObjectAppService/UnbindProperties

POST /TeamNetwork/ObjectAppService/DisableProperty

POST /TeamNetwork/ObjectAppService/EnableProperty

POST /TeamNetwork/ObjectAppService/ListProperties

3. Вставьте следующий код:

```

{
  "op1": "string",
  "op1MultiValue": true,
  "op2": "string",
  "op2MultiValue": true
}

```

4. Замените "string" на ID соответствующих атрибутов. Для этого:

- перейдите к списку атрибутов первого шаблона записи, скопируйте ID коллекции и вставьте его в первую строку кода;
- затем перейдите к списку атрибутов второго шаблона, скопируйте ID коллекции и вставьте его в 4 строку кода.

<input type="checkbox"/>	Ид	Системное имя ↓	Имя
<input type="checkbox"/>	op.101	Komandirovki	Командировки

Рис. 300 Скопируйте ID атрибута коллекции

<input type="checkbox"/>	Ид	Системное имя ↓	Имя	Тип данных
<input type="checkbox"/>	op.100	Zayavkanakomandi...	Заявка на команд...	Ссылка

Рис. 301 Скопируйте ID атрибута ссылки

5. Нажмите кнопку «Try it out»

Response Class (Status 200)
string

Response Content Type

Parameters

Parameter	Value	Description	Param Type
parameters	<pre>{ "op1": "op.101", "op1MultiValue": true, "op2": "op.100", "op2MultiValue": true }</pre>		body

Parameter content type:

[Try it out!](#) [Hide Response](#)

Рис. 302 Пример настройки связи N-N

Если все настроено верно, то между шаблонами появится связь «многие-ко-многим».

6.4 Solution Api (пользовательские методы)

В данном разделе находятся все шаблоны записи, которые есть в базе данных.

Для каждого шаблона записи в платформе автоматически генерируются четыре пользовательских метода Web API, предназначенных для выполнения следующих операций:

- создание записи

```

techwritingpc:8083/api/public/system/swagger

"swagger": "2.0",
"info": {
  "title": "Comindware Process API",
  "version": "1.0"
},
"basePath": "/api/public/system",
"paths": {
  "/TeamNetwork/ObjectService/Create": {
    "post": {
      "operationId": "TeamNetworkObjectServiceCreate",
      "consumes": [
        "application/json"
      ],
      "produces": [
        "application/json"
      ],
      "parameters": [
        {
          "name": "parameters",
          "in": "body",
          "schema": { "$ref": "#/definitions/TeamNetworkObjectServiceCreateParameters" }
        }
      ],
      "responses": {
        "200": {
          "name": "OK",
          "schema": { "type": "string" }
        }
      }
    }
  }
}

```

Рис. 303 Запись

- редактирование записи
- получение данных из определенной записи
- получение данных из всех записей

Список пользовательских методов можно посмотреть по адресу <http://your-host/Docs/SolutionApi>

Comindware Business Application Platform API	
Solution API	
build	Show/Hide List Operations Expand Operations
DELETE /build/{id}	
GET /build/{id}	
POST /build/{id}	
GET /build	
POST /build	
POST /build/Filter	
jobEvaluation	Show/Hide List Operations Expand Operations
Department	Show/Hide List Operations Expand Operations

Рис. 304 Пользовательские методы

В адресах пользовательских методов указывается системное имя шаблона записи.

Использование пользовательских методов будет рассмотрено на примере. Создайте шаблон записи с системным именем "Car" ("Машина") и присвойте ему текстовый атрибут "Color" ("Цвет"). Используйте пользовательские методы следующим образом:

- Для создания новой записи "Машина" отправьте запрос POST на адрес `http://yourhost/api/public/solution/Car`. Тело ("body") запроса должно содержать атрибут Color и его значение в формате JSON, например, `{"Color":"Red"}` для создания красной машины. В ответ на запрос платформа отправит id созданной записи (например, "1").
- Для редактирования записи "Машина", отправьте запрос POST на адрес `http://yourhost/api/public/solution/Car/1`, где "1" — id экземпляра записи. Тело ("body") запроса должно содержать атрибут Color и его новое значение: `{"Color":"Silver"}`.
- Для получения данных из записи, отправьте запрос GET на адрес `http://your-host/api/public/solution/Car/1`, где "1" — id записи. В ответ на запрос платформа отправит атрибут и его значение в формате JSON, например: `{"Color":"Silver"}`.
- Для получения данных из всех записей отправьте запрос GET на адрес `http://yourhost/api/public/solution/Car`. Платформа отправит массив, содержащий данные из всех записей в формате JSON.

7. Системное администрирование

Более полное руководство по системному администрированию содержится в документе «Руководство системного администратора».

В этой главе:



[Глобальные настройки](#)^[277]

Глобальные платформы и уведомления и обслуживание скриптов.

[Глобальные функции](#)^[7.2]

Глобальные функции и этапы их настройки в Comindware Business Application Platform.

7.1 Глобальные настройки

В данном разделе расположены настройки Comindware Business Application Platform, связанные с глобальными уведомлениями платформы, обслуживанием скриптов и серверной частью.

Необходимо открыть раздел **Глобальные настройки** в настройках платформы:

1. Кликните по иконке пользователя в правом верхнем углу.
2. Выберите пункт «Глобальные настройки».
3. Перейдите в раздел «Глобальные настройки».

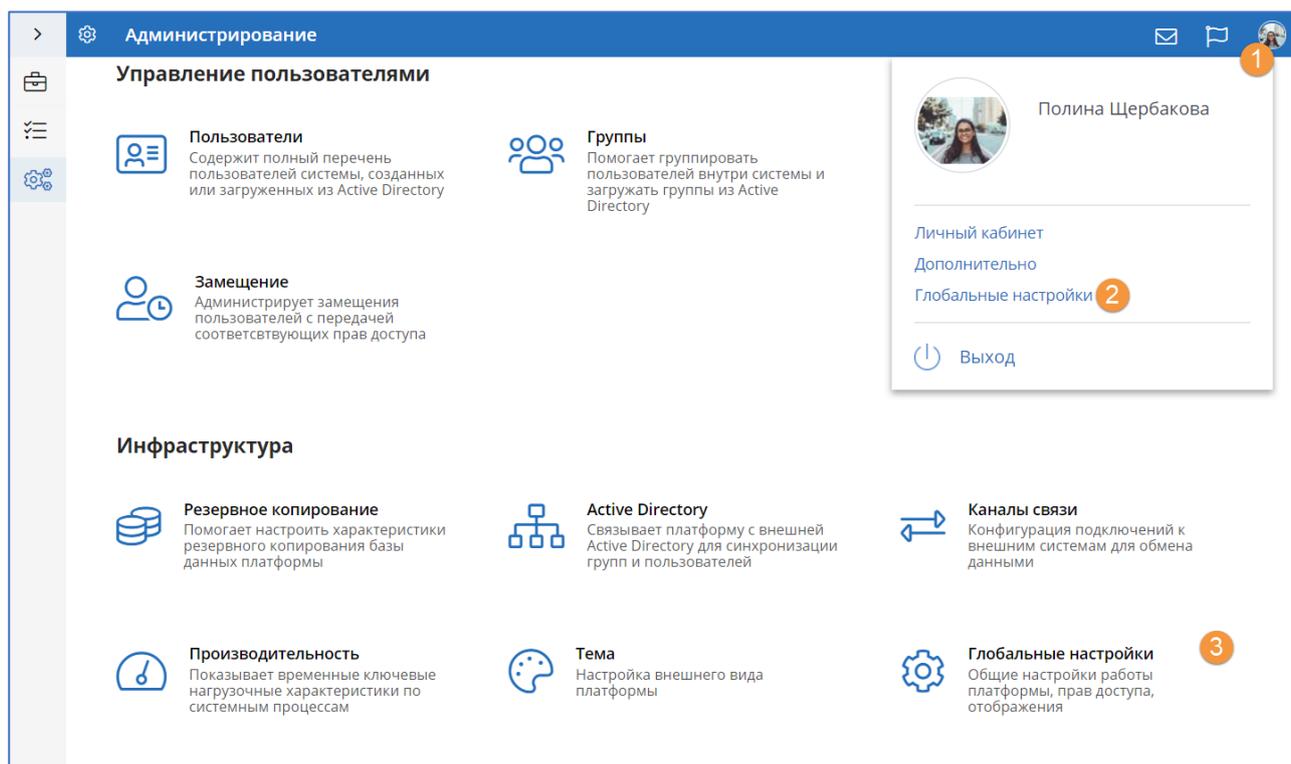


Рис. 305 Раздел **Администрирование**

В разделе **«Основные настройки»** отображены следующие операции:

- Флаг «Включить отслеживание производительности» позволяет отслеживать производительность в соответствующем разделе настроек платформы.
- Флаг «Включить внешние уведомления» позволяет отправлять уведомления о назначенных задачах на внешний сокет с адреса, указанного в поле «Внешний адрес сервера», а так же включать внутренние уведомления исполнителей задач.
- Внешней адрес сервера.
- Настройки сервера – часовой пояс и язык.
- Действие при редактировании страницы несколькими пользователями.

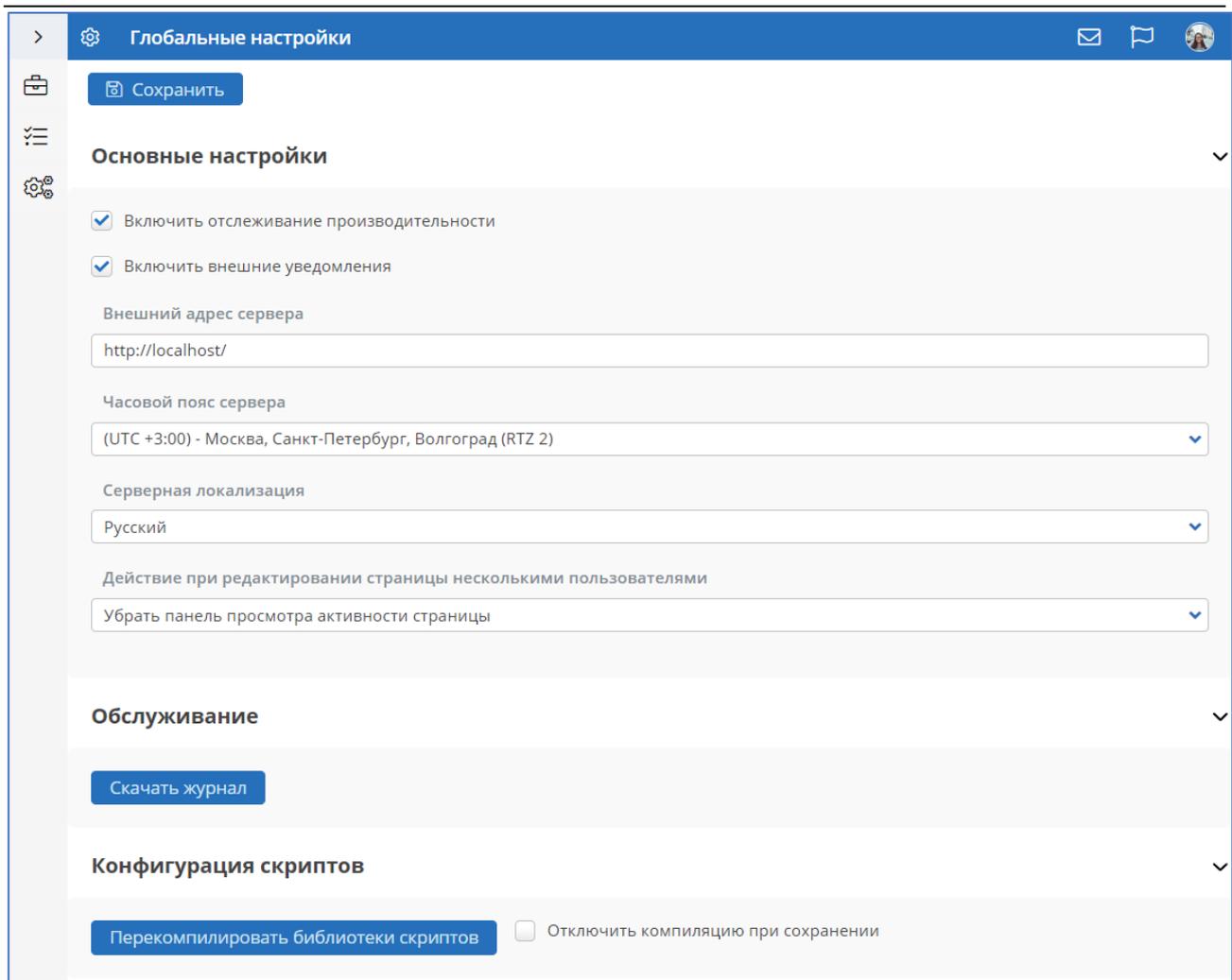


Рис. 306 Глобальные настройки

Раздел **«Обслуживание»** содержит кнопку для скачивания журнала (логов).

В разделе **«Конфигурация скриптов»** отображена системная операция перекомпилирования библиотеки скриптов – автоматическая компиляция всех скриптов платформы, за исключением процессов.

7.2 Глобальные функции

В этой главе:



[Введение в глобальные функции](#) (7.2.1)

Подробная информация о том, что представляет собой глобальная функция, и об этапах ее настройки в Comindware Business Application Platform.



[Определение глобальной функции](#) (7.2.3)

Определение глобальной функции нужного типа: скрипт C# или вызов веб-службы.



[Сигнатура глобальной функции](#) (7.2.2)

Создание функции и настройка ее сигнатуры.

7.2.1 Введение в глобальные функции

Глобальная функция — это функция, которая вызывается при выполнении процесса, когда это необходимо.

В платформе выделяют два типа глобальных функций:

- **Скрипт С#:** выполняется скрипт С#.
- **Вызов веб-службы:** платформа выполняет API запрос к внешней веб-службе (например, к сервису тех. поддержки, базе знаний и т. д.).

Каждая функция характеризуется двумя шаблонами сообщений — входным и выходным, соответствие которых параметрам того или иного процесса настраивается отдельно в каждом конкретном процессе.

Настройка глобальной функции состоит из трех этапов:

№	Место в платформе	Действие
1	Боковая навигация -> Глобальные функции -> Сигнатура	Создайте_глобальную функцию. Настройте сигнатуру глобальной функции.
2	Боковая навигация -> Шаблоны -> Шаблон процесса -> Схема	Добавьте на схему процесса задачу на вызов сервиса. В настройках задачи настройте соответствие между сигнатурой функции и атрибутами, и параметрами процесса.
3	Боковая навигация -> Глобальные функции -> Настройки	Настройте определение функции.

7.2.2 Сигнатура глобальной функции

Для того чтобы создать глобальную функцию и настроить ее сигнатуру:

1. Необходимо открыть раздел **Глобальные функции** в настройках платформы и создать функцию.

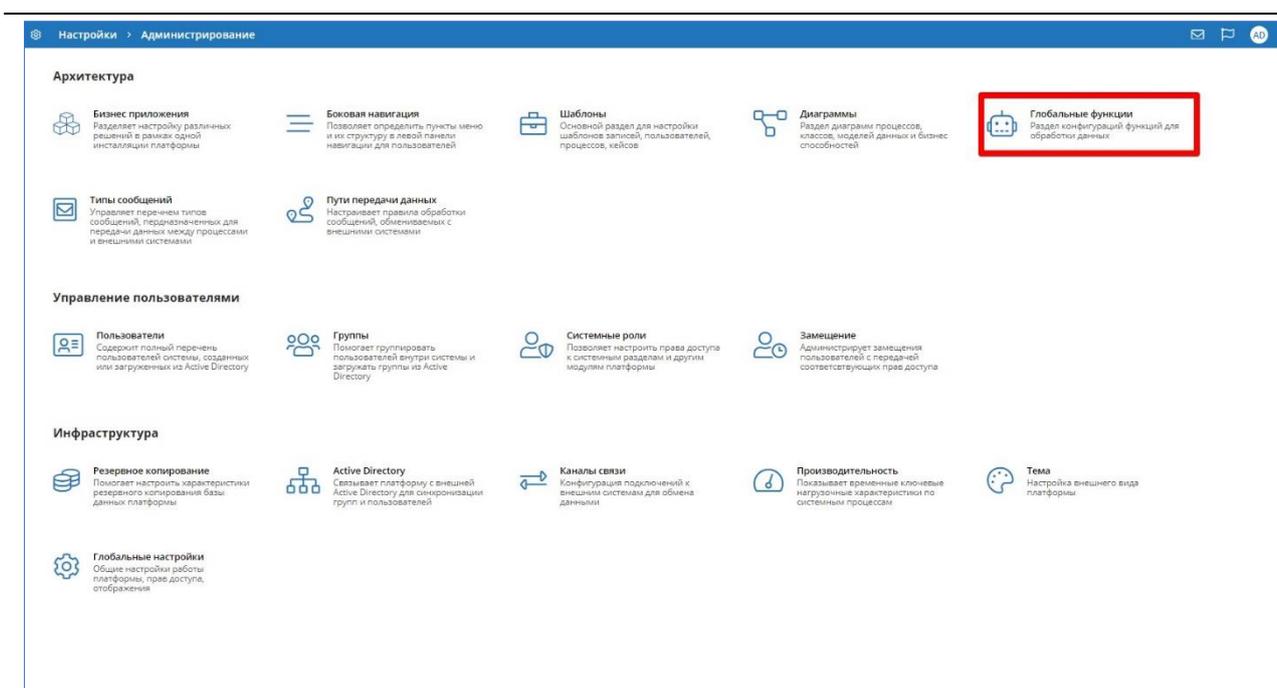


Рис. 307 Раздел Глобальные функции

2. Укажите отображаемое и системное имена функции и бизнес-приложение. При необходимости добавьте описание функции.
3. Выберите тип функции: **Скрипт C#** для вызова скрипта C# или **Вызов веб-службы** для настройки API запросов к внешней веб-службе (например, к сервису тех. поддержки, к базе знаний и т. д.), **Абстрактная функция** используется для отладки сигнатуры.

Создание новой глобальной функции

Основные настройки | Сигнатура | Описание

Имя*

Системное имя*

Бизнес-приложение*

Описание

Тип глобальной функции

Абстрактная функция

Скрипт C#

Вызов веб-службы

Отмена Сохранить

Рис. 308 Создание глобальной функции

4. Настройка сигнатуры функции заключается в создании входного и выходного шаблона сообщений:
 1. Создайте или выберите уже существующий Шаблон сообщения на входе.

Создание новой глобальной функции

Основные настройки | Сигнатура | Описание

Шаблон сообщения на входе

Создать Редактировать

Настройка соответствия

Имя	Системное имя	Тип данных

Шаблон сообщения на выходе

Создать Редактировать

Настройка соответствия

Имя	Системное имя	Тип данных

Отмена Сохранить

Рис. 309 Настройка сигнатуры функции

2. В форме Новый шаблон сообщения укажите имя нового шаблона сообщения.

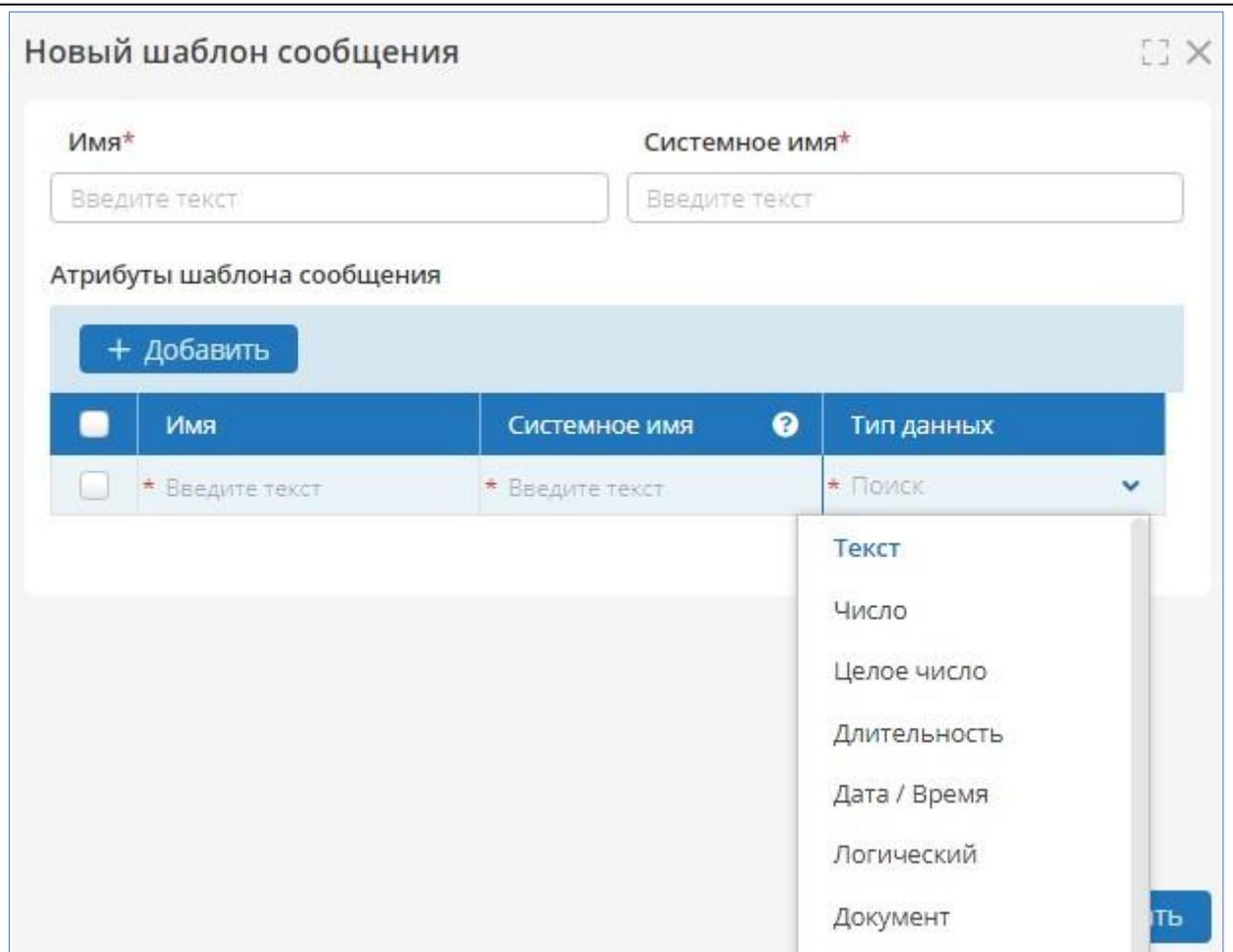


Рис. 310 Создание нового шаблона сообщения

3. Создайте атрибуты, в которых будут храниться данные сообщения. Укажите системные имена атрибутов, имена, которые будут отображаться в платформе, а также выберите тип данных.

Новый шаблон сообщения ☐ ✕

Имя* Системное имя*

Атрибуты шаблона сообщения

+ Добавить

<input type="checkbox"/>	Имя	Системное имя ?	Тип данных
<input type="checkbox"/>	* уца	* utsa	* Текст ▼
<input type="checkbox"/>	* мвфа	* mvfa	* Число ▼
<input type="checkbox"/>	* ывм	* yvm	* Дата / Время ▼
<input type="checkbox"/>	* уцамфц	* utsamfts	* Поиск ▼

Текст

Число

Целое число

Длительность

Добавить

Рис. 311 Создание атрибутов

4. Аналогичным образом создайте шаблон сообщения на выходе из функции.

7.2.3 Определение глобальной функции

Для того чтобы настроить определение глобальной функции:

1. Перейдите в раздел «**Описание**».
2. В зависимости от типа функции, настройте её определение:

C# скрипт

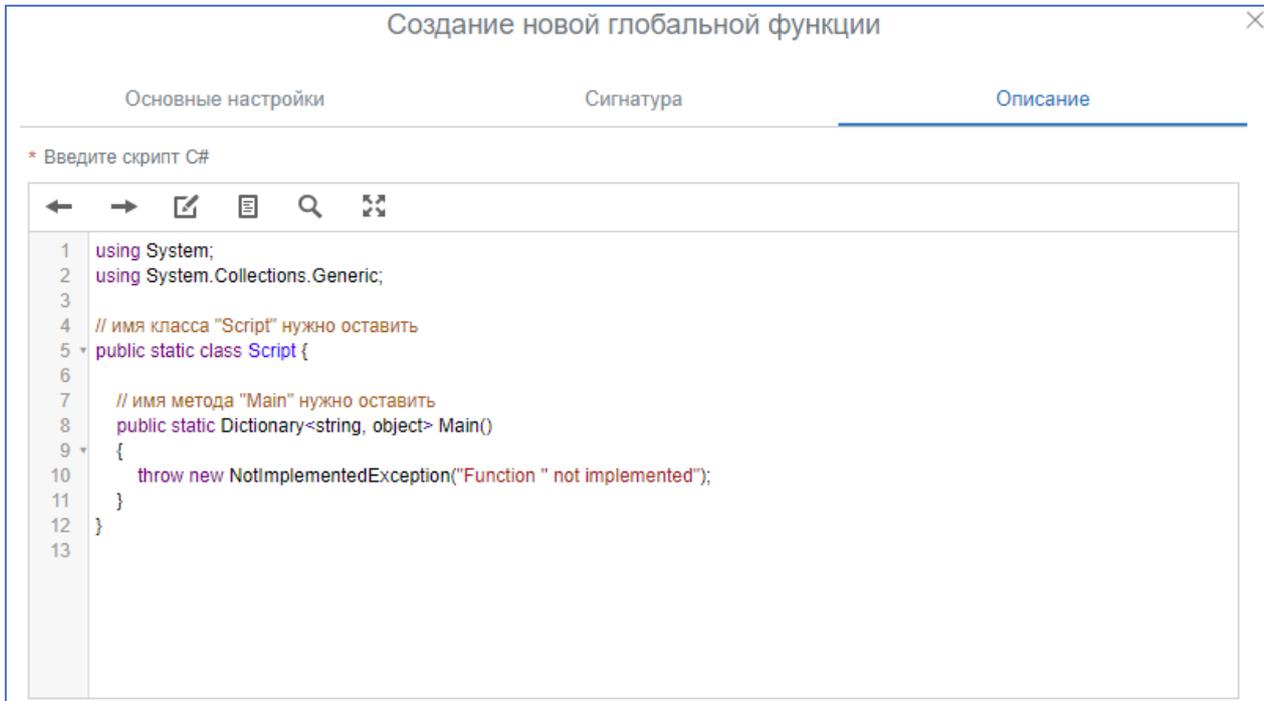


Рис. 312 Параметры C# скрипта

Напишите скрипт. Входные параметры являются параметрами функции. Используйте класс `Dictionary<TKey,TValue>` для передачи параметров ключа и значения. Ключевые параметры соответствуют выходным параметрам сигнатуры.

Настройте соответствие между результатом выполнения скрипта и атрибутами выходного шаблона сообщения. Для этого используйте ключи скрипта. Отметьте флаг **Использовать для вывода результата скрипта**, если скрипт возвращает единственное значение без ключа, и вы хотите записать это значение в определенный выходной атрибут.

Вызов веб-службы

1. Зарегистрируйте приложение Comindware Business Application Platform во внешней веб-службе в качестве клиента.
2. Настройте аутентификацию приложения Comindware Business Application Platform во внешней веб-службе.
3. Приготовьтесь к вводу параметров аутентификации, предоставленных веб-службой приложению Comindware Business Application Platform. Возможно, вам сначала потребуется изучить документацию веб-службы на соответствующую тему, чтобы узнать, каким образом сторонние приложения (в данном случае Comindware Business Application Platform) проходят аутентификацию при подключении к веб-службе. Используйте эти знания при задании настроек.

Настройка	Действие
Аутентификация	Выберите метод аутентификации: базовая аутентификация или без аутентификации.
Логин (базовая аутентификация)	Введите логин.
Пароль (базовая аутентификация)	Введите пароль.
Заголовки	<p>Нажмите кнопку Добавить заголовок, чтобы добавить заголовки данных, которые будут использоваться в API запросах к веб-службе. Синтаксис заголовков предоставляется веб-службой.</p> <p>Заголовок Content-Type является обязательным и может принимать значения "application/json" или "application/xml".</p>
Метод запроса	<p>Выберите метод запроса: GET, POST, PUT или DELETE, в зависимости от цели запроса.</p> <p>При выборе одного из методов «отправки» («POST» или «PUT») появится поле Текст запроса. Введите в это поле данные, которые должны быть отправлены в веб-службу.</p>
Адрес URL веб-службы	<p>Укажите полный адрес URI (Uniform Resource Identifier) веб-службы (см. документацию веб-службы). В адресе обязательно используйте "http" или "https".</p> <p>Пример: https://comindware.dreamsupport.ru.</p>
Метод парсинга	<p>Выберите метод парсинга данных, содержащихся в ответе веб-службы:</p> <p>"Как есть", "Xml", или "JSON".</p>
Настройка соответствия	<p>Настройте соответствие между атрибутами шаблона сообщения на выходе из функции и данными из ответа веб-службы.</p>

Создание новой глобальной функции

Основные настройки | Сигнатура | Описание

Адрес URI веб-службы*

Введите текст

Аутентификация

Без аутентификации

Заголовки

Content-Type application/json

Добавить заголовок

Метод запроса

GET

POST

PUT

DELETE

Метод парсинга

Как есть

XML

JSON

Отмена Сохранить

Рис. 313 Параметры вызова веб-службы

7.3 Переменные

Переменные – это атрибуты, привязанные к бизнес-приложению. Из можно вызывать в любом шаблоне данного бизнес-приложения и использовать в скриптах и выражениях.

1. Перейдите в окно настройки бизнес-приложения – выберите бизнес-приложение двойным кликом.

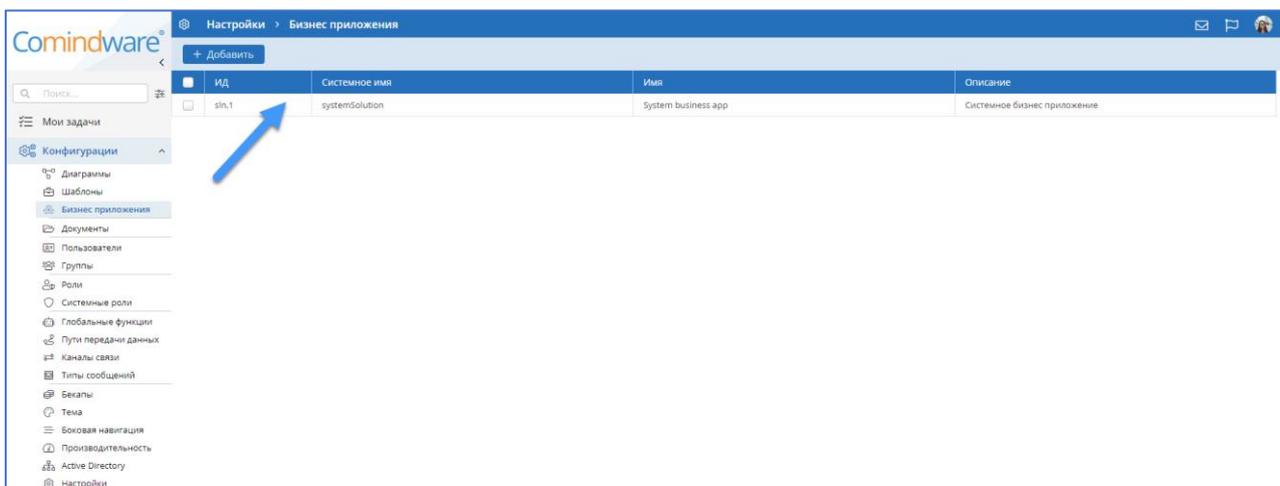


Рис. 314 Список бизнес-приложений

2. Перейдите в раздел **Переменные**.

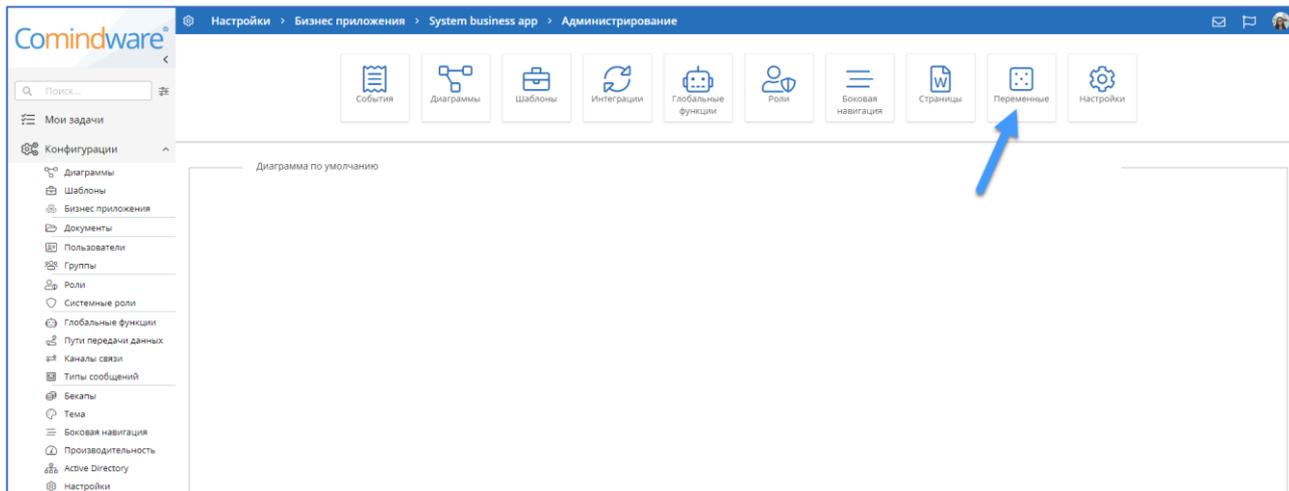


Рис. 315 Расположение раздела переменные

3. Нажмите на кнопку «Добавить».

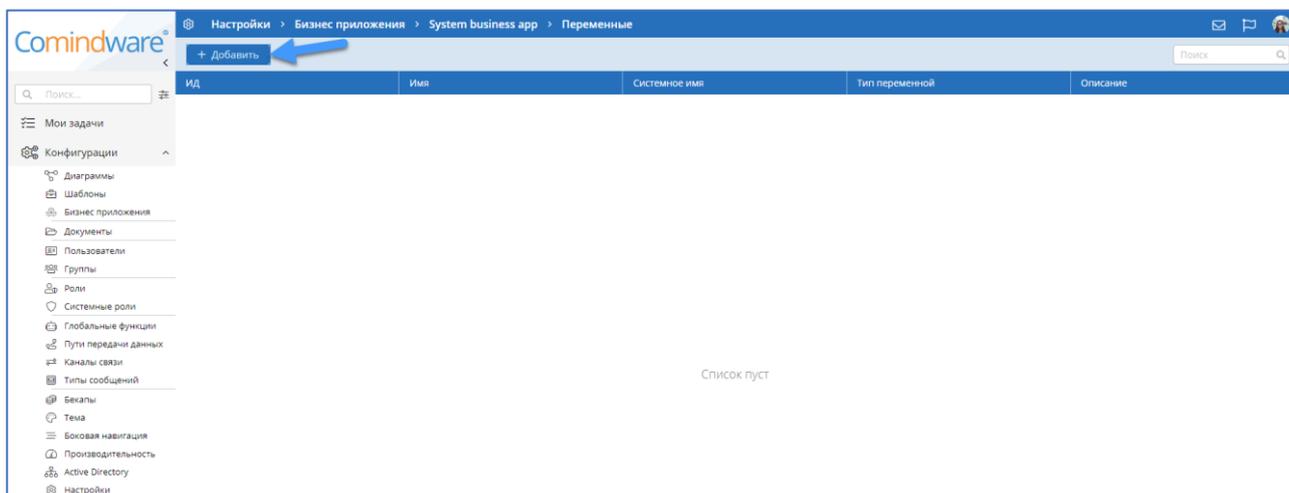


Рис. 316 Расположение кнопки «Добавить».

4. Откроется окно настройки переменной.

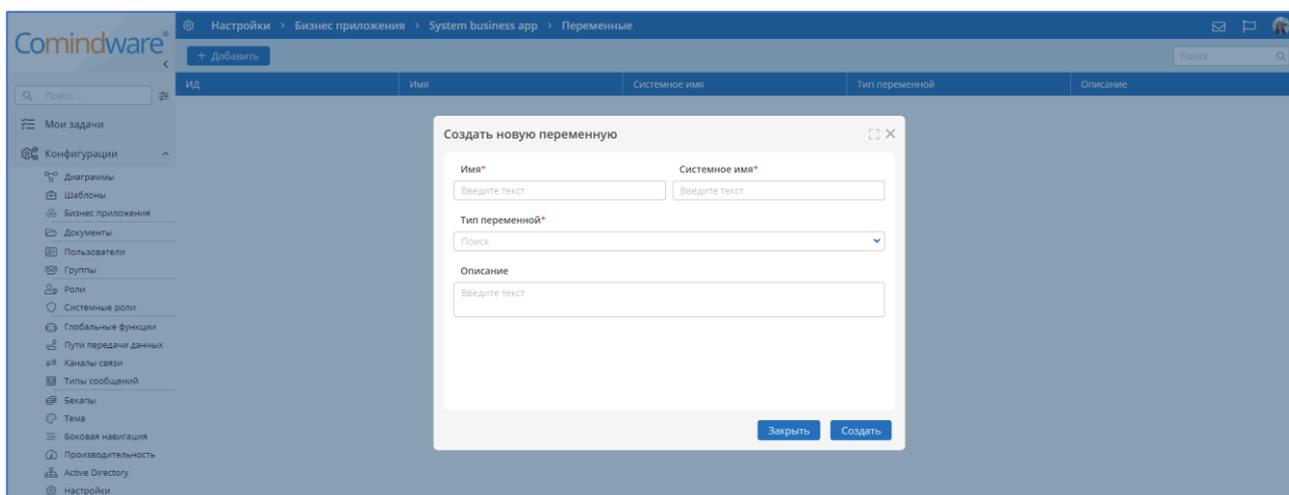
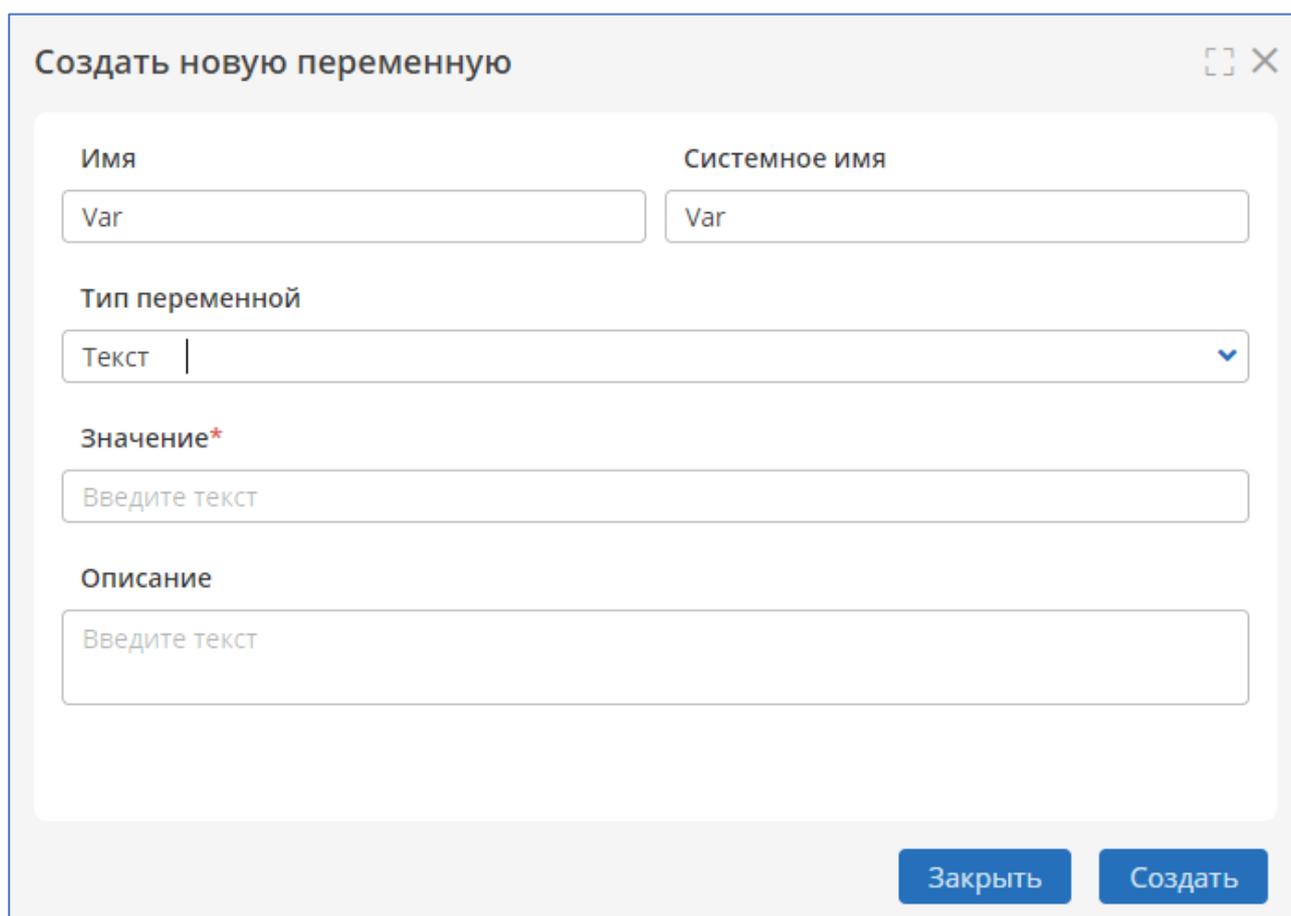


Рис. 317 Окно настройки переменной.

5. Окно настройки переменной содержит следующие поля:

- Имя – название переменной.
- Системное имя - уникальное текстовое системное имя элемента в рамках шаблона записи. Может содержать только буквы латинского алфавита, цифры и знак «_».
- Тип переменной – формат переменной, который определяет какую информацию может хранить переменная:
 - Текст – хранит строковую информацию.
 - Число – хранит числовую информацию.
 - Длительность – хранит информацию формата длительность.
 - Дата / время – хранит информацию формата дата/время.
 - Логический – хранит логическую информацию true или false.
- Описание – информация о цели использования переменной.

6. Заполните поля окна настройки переменной.



Создать новую переменную

Имя: Var

Системное имя: Var

Тип переменной: Текст

Значение*: Введите текст

Описание: Введите текст

Закреть Создать

Рис. 318 Заполненные поля окна настройки переменной.

7. Появится поле значение для ввода начального значения переменной.
8. Заполните поле значение в зависимости от выбранного типа переменной.
9. Нажмите на кнопку «Создать».

