

Программная платформа ReInform 2.0

РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

листов 229

Москва, 2023

АННОТАЦИЯ

Руководство пользователя представляет собой документ, предназначенный для обеспечения работы пользователей по эксплуатации и ведению информационных ресурсов Программной платформы ReInform 2.0 (далее – Платформа RIF, Система).

Руководство пользователя состоит из следующих разделов.

В разделе «Введение» приведено полное наименование Системы, область в которой применяется разработанная система, краткое описание возможностей Системы.

В разделе «Назначение и условия применения» приведен перечень функций подсистем. В разделе «Подготовка к работе» определен порядок подготовки Системы к работе, порядок загрузки данных.

В разделе «Описание операций» приведен порядок (последовательность действий пользователей) при работе с компонентами Системы.

В иллюстративных материалах приведены экранные формы и правила их заполнения.

В разделе «Аварийные ситуации» приведен порядок действий пользователей в случае возникновения аварийных ситуаций (сбоев и ошибок), контрольные примеры по освоению порядка работы с системой.

СОДЕРЖАНИЕ

- 1 ВВЕДЕНИЕ6**
 - 1.1 Наименование системы6
 - 1.2 Область применения6
 - 1.3 Краткое описание возможностей Платформы6
 - 1.4 Уровень подготовки пользователя7
 - 1.5 Перечень эксплуатационной документации, с которой необходимо ознакомиться пользователю8
- 2 НАЗНАЧЕНИЕ И УСЛОВИЯ ПРИМЕНЕНИЯ9**
 - 2.1 Назначение и виды деятельности, для автоматизации которых предназначена Платформа9
 - 2.2 Функции, для автоматизации которых предназначена Платформа9
 - 2.2.1 Подсистема портал9
 - 2.2.2 Подсистема управления бизнес-процессами9
 - 2.2.3 Подсистема хранения документов10
 - 2.2.4 Подсистема поиска11
 - 2.2.5 Подсистема интеграции с внешними системами12
 - 2.2.6 Подсистема управления справочниками и классификаторами13
 - 2.2.7 Подсистема инструментария разработки13
 - 2.2.8 Подсистема информационной безопасности16
- 3 ПОДГОТОВКА К РАБОТЕ17**
 - 3.1 Порядок загрузки данных и программ17
 - 3.2 Порядок проверки работоспособности17
- 4 ОПИСАНИЕ ОПЕРАЦИЙ17**
 - 4.1 Начало работы с Платформой17
 - 4.2 Раздел «Мои задачи»18
 - 4.3 Раздел «Мои возможности»21
 - 4.4 Создание информационного объекта в Конструкторе ИО22
 - 4.4.1 Описание данных конструктора информационных объектов22
 - 4.4.2 Описание витрины «Реестр информационных объектов»23
 - 4.4.3 Окно создания и редактирования свойств информационного объекта24
 - 4.4.4 Окно редактирования информационного объекта26
 - 4.4.5 Окно просмотра информационного объекта29
 - 4.4.6 Загрузка информационного объекта30
 - 4.5 Создание БП в конструкторе БП30
 - 4.5.1 Предварительная подготовка30
 - 4.5.2 Основные этапы разработки в конструкторе бизнес-процессов51
 - 4.5.3 Инструкция по работе пользователей с редактором бизнес-процессов58
 - 4.5.4 Руководство пользователя по работе с реестром прикладных сервисов80
 - 4.6 Создание задач на Конструкторе форм86
 - 4.6.1 Общая информация86
 - 4.6.2 Требования к XSD86
 - 4.6.3 Реестр бизнес-процессов102
 - 4.6.4 Стартовая форма104
 - 4.6.5 Общая информация о создании формы104
 - 4.6.6 Компоненты105
 - 4.6.7 Примеры js-кода, используемого в настройках форм задач177
 - 4.7 Создание карточки просмотра объекта в Конструкторе карточек178
 - 4.7.1 Реестр карточек178
 - 4.7.2 Создание новой карточки179
 - 4.7.3 Редактор карточек179
 - 4.7.4 Компоненты182
 - 4.7.5 Ссылка на карточку199

- 4.8 Создание витрины на Конструкторе витрин199
 - 4.8.1 Панель администрирования конструктора витрин199
 - 4.8.2 Список витрин199
 - 4.8.3 Редактор витрины200
 - 4.8.4 Предварительный просмотр214
 - 4.8.5 Встраивание витрины в приложение216
 - 4.8.6 Описание данных конструктора витрин216
- 4.9 Справочники223
 - 4.9.1 Справочник Systems223
- 4.10Администрирование226
- 5 АВАРИЙНЫЕ СИТУАЦИИ227**
 - 5.1 Действия в случае несоблюдения условий выполнения технологического процесса, в том числе при длительных отказах технических средств227
 - 5.2 Действия по восстановлению программ и/или данных при отказе магнитных носителей или обнаружении ошибок в данных227
 - 5.3 Действия в случаях обнаружении несанкционированного вмешательства в данные227
 - 5.4 Действия в других аварийных ситуациях227
- 6 РЕКОМЕНДАЦИИ ПО ОСВОЕНИЮ229**

ПЕРЕЧЕНЬ ТЕРМИНОВ И СОКРАЩЕНИЙ

АИС	– Автоматизированная информационная система
АРМ	– Автоматизированное рабочее место
БД	– База данных
БП	– Бизнес-процесс
ИО	– Информационные объекты
ИС	– Информационная(ые) система(ы)
ОПО	– Общее программное обеспечение
ПО	– Программное обеспечение
Платформа, Система	– Программная платформа автоматизации бизнес-процессов и производственной деятельности REINFORM
СПО	– Специальное программное обеспечение
СУБД	– Платформа управления базами данных
ЭП	– Электронная подпись
Apache SOLR	– Расширяемая поисковая платформа от Apache
API	– Программный интерфейс
BPM	– Методология управления бизнес-процессами (Business Process Management)
BPMN	– Система условных обозначений (нотация) и их описания в XML для моделирования бизнес-процессов
JSON	– Текстовый формат обмена данными, основанный на JavaScript
REST-API	– REST-интерфейса
SOLR	– Платформа полнотекстового поиска с открытым исходным кодом
XML	– англ. eXtensible Markup Language – расширяемый язык разметки
XSD	– англ. XML Schema definition – язык описания структуры XML документа
XML Scheme	– Язык метаописания сообщений

1 ВВЕДЕНИЕ

1.1 Наименование системы

Программная платформа автоматизации бизнес-процессов и производственной деятельности REINFORM (далее – Платформа, Система).

1.2 Область применения

Платформа разработана для создания, развития и эксплуатации информационных систем автоматизации бизнес-процессов, управления документами на базе готовых технологических компонентов и инструментария автоматизации бизнес-процессов с минимальным участием разработчиков (low-code).

Настоящее руководство предназначено для применения пользователями Системы при изучении ее возможностей.

1.3 Краткое описание возможностей Платформы

В состав Платформы входят следующие подсистемы:

- 1) Подсистема портал;
- 2) Подсистема управления бизнес-процессами;
- 3) Подсистема хранения документов
- 4) Подсистема поиска;
- 5) Подсистема интеграции с внешними системами;
- 6) Подсистема управления справочниками и классификаторами;
- 7) Подсистема инструментария разработки
- 8) Подсистема информационной безопасности.

Подсистема Портал обеспечивает формирование единой точки доступа пользователя ко всем возможностям предоставляемому ему Платформой и развёрнутыми на Платформе прикладными системами: единый навигатор, единые панели «Мои задачи» и «Мои возможности», интегрированную систему поиска.

Подсистема управления бизнес-процессами обеспечивает управление технологическими процессами автоматизации и управления задачами сотрудников по методологии управления бизнес-процессами (Business Process Management, BPM), в которой маршрут исполнения процесса моделируется в виде шаблона по стандарту BPMN в графической среде.

Подсистема хранения документов обеспечивает централизованное хранение электронных документов с настраиваемой моделью и структурой данных.

Подсистема поиска обеспечивает индексацию и поиск данных по различным параметрам.

Подсистема интеграции с внешними системами обеспечивает электронное взаимодействие между подсистемами.

Подсистема управления справочниками и классификаторами обеспечивает централизованное и единообразное ведение справочников и классификаторов и обеспечение их использования со стороны других подсистем при формировании данных, поиске и фильтрации данных, построении аналитической отчетности.

Подсистема инструментария разработки включает следующие компоненты:

- конструктор витрин, обеспечивает возможность конструирования в браузере витрин данных для поиска и просмотра массивов информационных объектов с заданными параметрами;
- конструктор информационных объектов обеспечивает визуальный редактор для создания в браузере иерархических бизнес-объектов сложной структуры;
- конструктор бизнес-процессов реализует в браузере визуальный редактор для создания и редактирования бизнес-процессов в нотации BPMN 2.0;
- конструктор форм обеспечивает визуальный редактор для создания в браузере форм пользовательских задач бизнес-процессов и форм ввода и редактирования данных;
- генератор печатных форм документов (генератор документов)

Подсистема информационной безопасности обеспечивает механизмы создания учётных записей пользователей, механизмы администрирования прав доступа пользователя к возможностям Платформы развёрнутых на Платформе прикладных систем, единые механизмы аутентификации и проверки прав доступа при обращении пользователя к информационным ресурсам и функциям Системы.

Для информационного обмена между компонентами Системы используются ресурсы серверного узла и локальной вычислительной сети со стандартными протоколами информационного обмена.

При организации информационного обмена между компонентами Системы использованы стандартные протоколы обмена данными, базирующиеся на стеке транспортных протоколов TCP/IP.

1.4 Уровень подготовки пользователя

К работе с Платформой допускаются Пользователи, имеющие общие навыки работы со средствами вычислительной техники. Пользователи должны уметь работать с веб-браузером, Microsoft Internet Explorer или его аналогами от других фирм, офисными приложениями MS Word и Excel и прошедшие специальную подготовку для работы с Платформой.

Специальная подготовка Пользователей включает в себя знания и навыки работы с комплексом технических средств и ОПО Платформы в объеме, необходимом для исполнения своих должностных обязанностей.

1.5 Перечень эксплуатационной документации, с которой необходимо ознакомиться пользователю

Перед началом работы с Платформой пользователь должен ознакомиться с настоящим Руководством пользователя.

2 НАЗНАЧЕНИЕ И УСЛОВИЯ ПРИМЕНЕНИЯ

2.1 Назначение и виды деятельности, для автоматизации которых предназначена Платформа

Платформа предназначена для создания, развития и эксплуатации информационных систем автоматизации бизнес-процессов, управления документами на базе готовых технологических компонентов и инструментария автоматизации бизнес-процессов с минимальным участием разработчиков (low-code).

2.2 Функции, для автоматизации которых предназначена Платформа

2.2.1 Подсистема портал

Подсистема портал реализует следующие функции:

- 1) единый навигатор для быстрого доступа пользователя к возможностям Платформы и развёрнутых на Платформе прикладных систем;
- 2) единая панель «Мои задачи», которая представляет собой интегрированный реестр назначенных на пользователя задач бизнес-процессов из всех прикладных систем;
- 3) единая панель «Мои возможности», представляющая собой интегрированный реестр всех бизнес-процессов, которые может инициировать пользователь, независимо от прикладной системы, к которой относится этот бизнес-процесс;
- 4) единая среда представления витрин данных Подсистемы поиска, независимо от информационной системы, которой принадлежат эти данные.

2.2.2 Подсистема управления бизнес-процессами

Подсистема управления бизнес-процессами (BPM) реализует следующие функции:

- 1) инициирование и исполнение процессов;
- 2) управление исполнением бизнес-процесса в соответствии с его структурой, определённой в схеме (шаблоне), учётом условий ветвления и переходов между задачами (маршрутизация задач);
- 3) автоматическое назначение процесса исполнителю исполнителями в соответствии с шаблоном процесса;
- 4) вычисление плановых сроков задач с учетом рабочего календаря;
- 5) ведение рабочего календаря на год с учетом праздничных дней, переносов рабочих дней, временного режима работы;
- 6) заполнение необходимых форм ввода данных в ходе выполнения задач процесса;
- 7) рассылка уведомлений исполнителям о назначении новых задач;
- 8) визуальный контроль выполнения процесса в табличном и графическом виде;
- 9) администрирование (назначение, изменение) исполнителей задач;
- 10) делегирование задач между сотрудниками (в случае временного отсутствия сотрудника, в случае делегирования полномочий);

11) формирование персонифицированного представления с отбором задач пользователя.

2.2.3 Подсистема хранения документов

Подсистема хранения документов обеспечивает загрузку, хранение файлов документов и их атрибутов, а также предоставление файлов документов для различных потребителей документов, хранящихся в системе.

Для обеспечения этих функций в данной подсистеме реализована атрибутная модель данных для хранения электронных документов со связями между ними, а также реализован программный интерфейс со следующими функциями:

- 1) создание нового или обновление существующего файла;
- 2) поиск файлов;
- 3) получение файла (контента), получение информации о файле/папке (получение атрибутов);
- 4) привязка файлов к папке;
- 5) отвязывание файлов от папки;
- 6) изменение атрибутов файла/папки;
- 7) переименование папки и изменение её атрибутов;
- 8) создание дубликата файла в той же папке с другим именем;
- 9) копирование файлов в другую папку;
- 10) создание/ удаление папки/ файла;
- 11) сохранение файла с электронной подписью (ЭП) в хранилище;
- 12) получение/ сохранение/проверка ЭП файла хранилища;
- 13) удаление /усовершенствование ЭП;
- 14) формирования признака «скрытая папка»;
- 15) удаление признака скрытия папки;
- 16) вычисление хэш-файла по разным алгоритмам;
- 17) получение списка родительских папок для заданной папки/файла;
- 18) получение списка объектов уровня папки (с/без скрытых объектов);
- 19) получение файла на просмотр в браузере без предварительного сохранения на диск;
- 20) создание архивов с файлами и ЭП;
- 21) формирование архива с заданной структурой каталогов в фоновом режиме.

Программный интерфейс (API) реализован в виде REST-интерфейса (REST-API).

Программный интерфейс REST-API документирован в формате интерактивной документации Swagger с возможностью ознакомления с API и онлайн-тестирования в веб-

интерфейсе.

В качестве выходных параметров функций используются стандартные коды, описание которых приведено ниже (Таблица 1).

Таблица 1 – Коды выходных параметров функций

Код ответа	Описание ответа
200	ОК (успешное выполнение функции)
400	Данные запроса не корректны
403	Недостаточно полномочий
404	Запрашиваемый ресурс не найден
500	Внутренняя ошибка сервера

В случае ответа с кодом 500 (Внутренняя ошибка сервера) функция выдает дополнительное сообщение в формате JSON, содержащее системное и человеко-читаемое сообщение об ошибке.

2.2.4 Подсистема поиска

Подсистема поиска обеспечивает возможность поиска информационных объектов в системе по их атрибутам. Подсистема поиска разработана на основе специализированной платформы полнотекстового поиска с открытым исходным кодом Apache SOLR.

Для управления параметрами настройки индексирования данных с учетом структуры хранения данных Платформы (JSON-объекты) и требований программного интерфейса (API) Apache SOLR применяются:

1) справочники: «Ядра подсистемы поиска», «Типы полей подсистемы поиска», «Поля документов подсистемы поиска», «Типы поиска по полям», «Правила разграничения прав доступа к результатам общего поиска»;

2) программный интерфейс (API), в том числе:

- функция создания ядра;
- функция запроса списка ядер и типов документов;
- функция получения списка ошибок конфигурации;
- функция экспорта справочников;
- функция запроса даты последнего обновления конфигурации;
- функция обновления конфигурации модуля;
- функция удаления документов (по ядру, идентификатору, типу);
- функция запроса всех документов для переиндексации;
- функция обновления документов;
- функция выгрузки отчета по настройкам по типу документа;

- функция пересоздания полей схемы ядра;
- функция синхронизации полей схемы ядра;
- функция поиска документов по типу и идентификатору, расширенного поиска;
- функция запроса через синтаксис подсистемы поиска.

Программный интерфейс (API) реализован в REST-интерфейсе (REST-API).

Программный интерфейс REST-API документирован в формате интерактивной документации Swagger с возможностью ознакомления с API и онлайн-тестирования в веб-интерфейсе.

В качестве выходных параметров функций используются стандартные коды, (Таблица 1).

В случае ответа с кодом 500 (Внутренняя ошибка сервера) функция выдает дополнительное сообщение в формате JSON, содержащее системное и человеко-читаемое сообщение об ошибке.

2.2.5 Подсистема интеграции с внешними системами

Подсистема интеграции с внешними системами обеспечивает обмен информацией между компонентами и подсистемами Платформы, а также для автоматизации информационного взаимодействия при интеграции с внешними информационными системами и реализует следующие функции:

- 1) обмен сообщениями в асинхронном режиме;
- 2) обмен сообщениями в синхронном режиме;
- 3) получение информации о сообщении по идентификатору;
- 4) получение информации о сообщениях, отправленных подписчику за указанный период;
- 5) получение информации о количестве сообщений, отправленных подписчику за указанный период;
- 6) хранение метаданных, в том числе:
 - системы, источники сообщений;
 - системы, получатели сообщений;
 - типы публикуемых сообщений;
 - типы получаемых сообщений;
 - полученные от источников сообщения;
 - передаваемые подписчикам сообщения;
 - журнал передачи сообщений;
 - настройки Подсистемы интеграции с внешними системами.

2.2.6 Подсистема управления справочниками и классификаторами

Подсистема управления справочниками и классификаторами обеспечивает реализацию следующих функций:

- 1) хранение и предоставление реестра справочников и классификаторов;
- 2) создание справочника / классификатора, редактирование метаданных справочника / классификатора;
- 3) хранение и предоставление элементов справочника / классификатора;
- 4) создание и редактирование элемента справочника / классификатора;
- 5) обеспечение возможности связывания справочников – использование в качестве значения атрибута элемента другого (связанного) справочника;
- 6) поддержка иерархических справочников / классификаторов;
- 7) загрузка / выгрузка справочников в формате JSON;
- 8) предоставление сервисов (API) для других подсистем по получению справочников / классификаторов.

Программный интерфейс Подсистемы управления справочниками и классификаторами реализован в виде REST-интерфейса (REST-API).

Программный интерфейс REST-API документирован в формате интерактивной документации Swagger с возможностью ознакомления с API и онлайн-тестирования в веб-интерфейсе

2.2.7 Подсистема инструментария разработки

Подсистема инструментария разработки обеспечивает инструментарий конструкторов прикладных приложений и автоматизации бизнес-процессов с минимальным участием разработчиков (low-code) для выполнения функций.

Данная подсистема включает следующие компоненты:

- 1) Конструктор информационных объектов;
- 2) Конструктор бизнес-процессов;
- 3) Конструктор форм;
- 4) Конструктор витрин;
- 5) Генератор документов.

Программный интерфейс Подсистемы инструментария разработки реализован в виде REST-интерфейса (REST-API).

2.2.7.1 Конструктор информационных объектов

Конструктор информационных объектов обеспечивает возможности создания и визуального редактирования в браузере иерархических бизнес-объектов сложной древовидной структуры и реализует следующие функции:

- 1) визуальное создание модели данных информационного объекта с сохранением данных в формате XSD;
- 2) работу с реестром моделей данных информационных объектов;
- 3) работу со справочником типов информационных объектов;
- 4) получение типов информационных объектов из реестра моделей;
- 5) импорт/экспорт схемы информационного объекта в реестр;
- 6) сохранение/изменение структуры данных для документа по его типу в реестре моделей;
- 7) удаление структуры данных из реестра моделей;
- 8) получение из реестра моделей описания структуры данных в JSON.

Конструктор информационных объектов включает:

- визуальный компонент, работающий в браузере;
- backend-компоненты, обеспечивающие хранение и использование спецификаций информационных объектов другими компонентами Платформы.

2.2.7.2 Конструктор бизнес-процессов

Конструктор бизнес-процессов включает:

- средства визуального редактора для создания и изменения в браузере иерархических бизнес-объектов древовидной структуры;
- средства создания и редактирования схемы (шаблона) бизнес процесса в графической среде с определением параметров задач и переходов и условий переходов.

Конструктор бизнес-процессов обеспечивает реализацию следующих функций:

- 1) создание, редактирование, удаление схемы бизнес-процесса в стандарте BPMN;
- 2) хранение конфигураций схем бизнес-процессов;
- 3) создание и редактирование настроек (конфигураций) схем бизнес-процессов;
- 4) создание и настройку прикладных сервисов бизнес-процессов.

2.2.7.3 Конструктор форм

Конструктор форм обеспечивает визуальное создание и редактирование в браузере форм пользовательских задач бизнес-процессов, а также форм ввода данных (карточек объектов).

2.2.7.3.1 Функция создания форм пользовательских задач

Функция создания форм пользовательских задач обеспечивает:

- 1) создание спецификации форм задач в визуальном дизайнера динамических форм;
- 2) подключение форм задач к общей витрине задач («Мои задачи»);
- 3) создание и редактирование настроек отображения форм пользовательских задач бизнес-процессов;

4) подключение процесса к меню «Мои возможности» (инициирование бизнес-процессов)

5) конфигурирование и настройку отображения форм инициирования бизнес-процессов.

2.2.7.3.2 Функция создания карточек объектов

Функция создания карточек объектов обеспечивает:

- 1) создание, изменение карточки объекта;
- 2) формирование реестра карточек объекта;
- 3) связку карточки с информационным объектом;
- 4) отображение данные полей объекта, представленного в формате JSON;
- 5) настройку правил отображения данных в карточке объекта.

2.2.7.4 Конструктор витрин

Конструктор витрин обеспечивает возможность конструирования в браузере витрин данных для поиска и просмотра массивов информационных объектов с заданными параметрами выборки (реестров объектов).

Конструктор витрин обеспечивает реализацию следующих функций:

- 1) создание, редактирование, копирование, удаление реестров объектов;
- 2) предварительный просмотр реестра объектов;
- 3) формирование списка реестров объектов;
- 4) настройка полей, фильтров, колонок, сортировки, группировки данных в реестре;
- 5) пользовательский интерфейс реестра;
- 6) сброс пользовательских фильтров и сортировок реестра при выходе из системы;
- 7) сохранение пользовательских фильтров и сортировок реестра в рамках сессии;
- 8) управление поиском, в том числе:
 - переключение представления результатов поиска;
 - поиск по строке реестра;
 - расширенный поиск (фильтрация);
 - управление колонками таблицы результатов поиска;
 - поиск по логину пользователя для нескольких полей;
 - управление количеством выводимых результатов поиска;
 - очистка поискового запроса;
 - управление сортировкой результатов поиска;
 - группировка результатов поиска;
 - постраничная навигация по результатам поиска.

2.2.7.5 Генератор документов

Генератор документов обеспечивает возможность создания в системе печатных форм документов с использованием данных и метаданных информационных объектов.

Генератор документов обеспечивает реализацию следующих функций:

- формирование, хранение шаблонов печатных форм документов;
- ведение реестра печатных форм шаблонов документов;
- поддержка версионности шаблонов печатных форм документов;
- формирование документов по шаблону в форматах pdf, word;
- создание документа путем «склейки» страниц нескольких исходных документов;
- наложение страниц поверх страниц исходного документа.

2.2.8 Подсистема информационной безопасности

Подсистема информационной безопасности обеспечивает реализацию функций, связанных с управлением пользователями и правами доступа, в том числе:

- 1) создание и изменение учётных записей пользователей в едином каталоге;
- 2) блокировка и разблокировка учётных записей пользователей;
- 3) журналирование входов пользователя в систему;
- 4) управление правами доступа пользователей пользователя к информационным ресурсам и возможностям Платформы и прикладных систем;
- 5) проверка прав доступа пользователя при его обращении к информационным ресурсам и функциям Платформы и информационных систем.

3 ПОДГОТОВКА К РАБОТЕ

Платформа относится к классу веб-приложений, дистрибутив Платформы поставляется в комплекте ПО.

Платформа размещена по закреплённому за ней веб-адресу. Автоматизированное рабочее место (далее – АРМ) Пользователя должно входить в состав сети Интранет организации. Для запуска и работы с Платформой на «Рабочем столе» АРМ Пользователя необходимо настроить ссылку на веб-адрес Платформы.

Настройку «Рабочего стола» проводит Администратор Платформы. Для входа в Платформу Пользователь может самостоятельно указать в поле адреса браузера веб-адрес Платформы и нажать на клавиатуре клавишу Enter (Ввод).

3.1 Порядок загрузки данных и программ

Данные и программы загружаются на соответствующих серверах Сети передачи данных до начала работы Пользователя с Платформой. В процессе работы с Платформой Пользователь посредством операций ввода, которые будут описаны далее, сможет дозагружать необходимые данные в Платформу.

3.2 Порядок проверки работоспособности

Работоспособность Платформы проверяется Администратором Платформы после установки соответствующей ссылки (иконки) на рабочем столе АРМ Пользователя. В случае обнаружения неработоспособности какой-либо функции в процессе реальной работы с Платформой Пользователю следует связаться со службой технической поддержки Платформы.

4 ОПИСАНИЕ ОПЕРАЦИЙ

4.1 Начало работы с Платформой

Для входа в Платформу используется ссылка (иконка) на «Рабочем столе» АРМ. Ссылка настраивается системным администратором при первичной настройке АРМ Пользователя.

В открывшейся экранной форме авторизации (Рисунок 1), Пользователю необходимо ввести свои логин и пароль, предоставленные системным администратором, и нажать клавишу [Enter] (или кнопку «Войти»). Если вы забыли пароль – нажмите ссылку «Забыли пароль» (щёлкнуть мышью по ссылке).

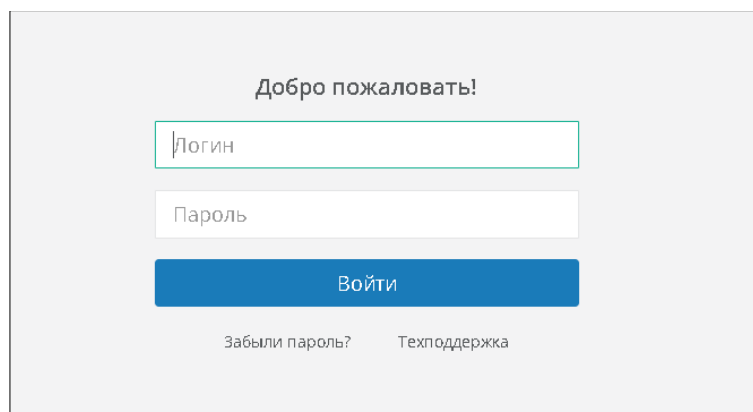


Рисунок 1 – Экранная форма авторизации

При появлении сообщения об ошибке авторизации, необходимо проверить раскладку клавиатуры (EN/RU), убедиться, что не включен режим CapsLock, и попробовать авторизоваться еще раз. При затруднении входа в Систему обратитесь в техническую поддержку по ссылке «Техподдержка».

4.2 Раздел «Мои задачи»

Раздел «Мои задачи» представляет собой персонализированный список задач пользователя в соответствии с исполняемыми бизнес-процессами (Рисунок 2).

Перечень «Мои задачи» имеет цветовую схему. На зеленом фоне указано общее количество данного типа задач, находящихся на исполнении у данного пользователя. На красном фоне указано количество данного типа задач, срок исполнения которых истек. На сером фоне – количество данного типа задач, находящихся в работе, с неистекшим сроком исполнения.

Список задач содержит следующие сведения:

- дату / время назначения задачи;
- иконки-признаки делегирования задачи, конкурентной задачи, назначенной на группу исполнителей или автоматической задачи;
- наименование задачи;
- описание объекта задачи;
- срок исполнения с индикацией просрочки.

Система позволяет фильтровать задачи по дате, искать по описанию в поле «Найти задачу».

Строка задачи является ссылкой на соответствующую форму исполнения задачи.

Зайцева Светлана Леонтьевна [Выйти](#)

Мои задачи

Все задачи 19 6 13

Назначена	Задача	Содержание	Срок
12 июля 16:54	Исполнить заявку на закупку офисного оборудования	Заявка на закупку офисного оборудования № 13 от 12.07.2023 Зайцева Светлана Леонтьевна	26 июля 17:00
12 июля 16:57	Исполнить заявку на закупку офисного оборудования	Заявка на закупку офисного оборудования № 14 от 12.07.2023 Зайцева Светлана Леонтьевна	26 июля 17:00
12 июля 16:47	Исполнить заявку на закупку офисного оборудования	Заявка на закупку офисного оборудования № 12 от 12.07.2023 Зайцева Светлана Леонтьевна	26 июля 17:00
10 июля 14:39	Исполнить заявку на закупку офисного оборудования	Заявка на закупку офисного оборудования № 7 от 10.07.2023 Зайцева Светлана Леонтьевна	24 июля 17:00
7 июля 17:05	Исполнить заявку на закупку офисного оборудования	Заявка на закупку офисного оборудования № 5 от 26.06.2023 Зайцева Светлана Леонтьевна	Завтра 15:45
10 июля 13:39	Согласовать заявку на закупку офисного оборудования	Заявка на закупку офисного оборудования № 5 от 26.06.2023 Зайцева Светлана Леонтьевна	Завтра 14:55
13 июля 11:40	Согласовать заявку на закупку офисного оборудования	Заявка на закупку офисного оборудования № 15 от 13.07.2023 Зайцева Светлана Леонтьевна	18 июля 17:00
11 июля 16:31	Согласовать заявку на закупку офисного оборудования	Заявка на закупку офисного оборудования № 5 от 26.06.2023 Зайцева Светлана Леонтьевна	14 июля 15:45

Рисунок 2 – Раздел «Мои задачи»

При нажатии на пиктограмму перечень задач пользователя по типам убирается. При повторном нажатии – перечень восстанавливается (Рисунок 3).

Все задачи 19 6 13

Назначена	Задача	Содержание	Срок
12 июля 16:54	Исполнить заявку на закупку офисного оборудования	Заявка на закупку офисного оборудования № 13 от 12.07.2023 Зайцева Светлана Леонтьевна	26 июля 17:00
12 июля 16:57	Исполнить заявку на закупку офисного оборудования	Заявка на закупку офисного оборудования № 14 от 12.07.2023 Зайцева Светлана Леонтьевна	26 июля 17:00
12 июля 16:47	Исполнить заявку на закупку офисного оборудования	Заявка на закупку офисного оборудования № 12 от 12.07.2023 Зайцева Светлана Леонтьевна	26 июля 17:00
10 июля 14:39	Исполнить заявку на закупку офисного оборудования	Заявка на закупку офисного оборудования № 7 от 10.07.2023 Зайцева Светлана Леонтьевна	24 июля 17:00
7 июля 17:05	Исполнить заявку на закупку офисного оборудования	Заявка на закупку офисного оборудования № 5 от 26.06.2023 Зайцева Светлана Леонтьевна	Завтра 15:45
10 июля 13:39	Согласовать заявку на закупку офисного оборудования	Заявка на закупку офисного оборудования № 5 от 26.06.2023 Зайцева Светлана Леонтьевна	Завтра 14:55
13 июля 11:40	Согласовать заявку на закупку офисного оборудования	Заявка на закупку офисного оборудования № 15 от 13.07.2023 Зайцева Светлана Леонтьевна	18 июля 17:00
11 июля 16:31	Согласовать заявку на закупку офисного оборудования	Заявка на закупку офисного оборудования № 5 от 26.06.2023 Зайцева Светлана Леонтьевна	14 июля 15:45

Рисунок 3 – Перечень задач пользователя по типам убирается

При нажатии на строку выбранного типа задачи в перечне «Мои задачи» – справа отображается перечень задач выбранного типа, количество которых соответствует числу, отображаемому на зеленой подложке (Рисунок 4).

Мои задачи

- 19 Все задачи (6, 13)
- 11 Закупка офисного оборудования (6, 5)
- 6 Согласовать заявку на закупку офисного оборудования (1, 5)
- 5 Исполнить заявку на закупку офисного оборудования


Согласовать заявку на закупку офисного оборудования

Найдено задач: 6, 1, 5.

Назначена	Задача	Содержание	Срок
10 июля 13:39	Согласовать заявку на закупку офисного оборудования	Заявка на закупку офисного оборудования № 5 от 26.06.2023 Зайцева Светлана Леонтьевна	Завтра 14:55
13 июля 11:40	Согласовать заявку на закупку офисного оборудования	Заявка на закупку офисного оборудования № 15 от 13.07.2023 Зайцева Светлана Леонтьевна	18 июля 17:00
11 июля 16:31	Согласовать заявку на закупку офисного оборудования	Заявка на закупку офисного оборудования № 5 от 26.06.2023 Зайцева Светлана Леонтьевна	14 июля 15:45
6 июля 15:26	Согласовать заявку на закупку офисного оборудования	Заявка на закупку офисного оборудования № 6 от 06.07.2023 Зайцева Светлана Леонтьевна	11 июля 17:00
29 июня 12:02	Согласовать заявку на закупку офисного оборудования	Заявка на закупку офисного оборудования № 5 от 26.06.2023 Зайцева Светлана Леонтьевна	4 июля 17:00
29 июня 12:19	Согласовать заявку на закупку офисного оборудования	Заявка на закупку офисного оборудования № 5 от 26.06.2023 Зайцева Светлана Леонтьевна	4 июля 17:00

Первая | Предыдущая | 1 | Следующая | Последняя | 50

Рисунок 4 – Перечень задач выбранного типа

При переходе по гиперссылке в названии задачи без значка  – открывается карточка задачи (Рисунок 5).



Исполнить заявку на закупку офисного оборудования

Дата создания: 12.07.2023 16:54 | Срок исполнения: 26.07.2023 17:00 | Исполнитель: Зайцева С.Л.

Автор заявки

Логин: zaitseva_sl
 ФИО: Зайцева Светлана Леонтьевна
 Адрес электронной почты: szaytseva@reinform-int.ru
 Должность: Системный аналитик
 Организация: Тестовая организация
 Телефон: Комитет строительства Курской области


Сведения в заявке

Номер и дата: №13 от 12.07.2023
 Категория оборудования: Мебель
 Примечание: Сведения отсутствуют
 Файл заявки:  Заявка на закупку офисного оборудования №13.pdf 58.7 Кб 12.07.2023 16:54 
 Скачать | Скачать ЭП

Состав заявки

Наименование	Количество
Стул офисный STAFF "Iso Lite CF-008"	1

Рисунок 5 – Карточка задачи

При переходе по гиперссылке в названии задачи со значком  – открывается сначала окно выбора задачи. После нажатия кнопки «Взять» пользователь-кандидат становится исполнителем задачи (Рисунок 6).

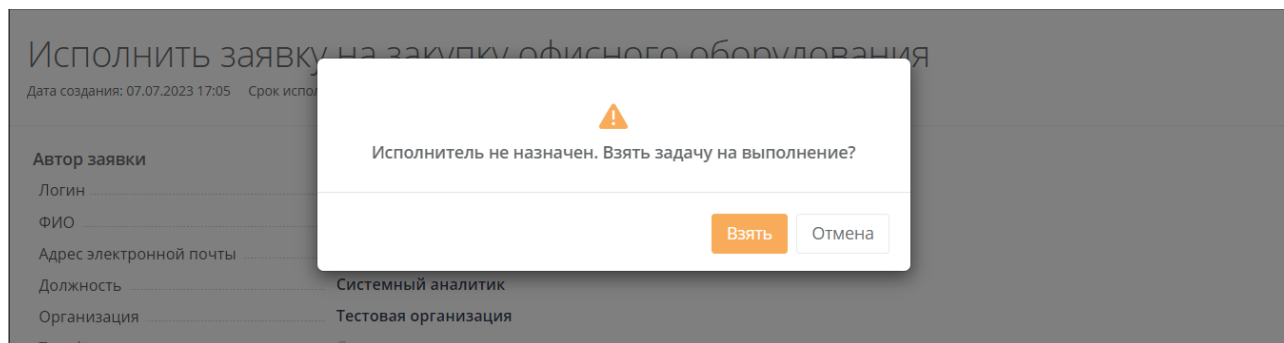


Рисунок 6 – Кнопка «Взять»

4.3 Раздел «Мои возможности»



В разделе «Мои возможности» отображается список возможных действий, доступных пользователю (например – запуск бизнес-процесса, создание документа и т.д.)


Список действий раздела «Мои возможности» (Рисунок 7) содержит:

- наименование возможности;
- функционал добавления в избранное;
- описание возможности;
- группу принадлежности.



Рисунок 7 – Кнопка «Взять»

Для того, чтобы часто используемые возможности отображались в начале списка необходимо нажать на пиктограмму , после чего пиктограмма будет отображаться в цвете как  и будет зафиксирована в начале списка возможностей. Вслед за избранными возможностями будут отображены остальные.

Чтобы отобразить только избранные возможности, достаточно нажать на пиктограмму  в шапке таблицы.

Доступен функционал фильтрации возможностей по алфавиту. Для того, чтобы применить фильтр, необходимо нажать на заголовок столбца, текст: «Наименование» либо на заголовок столбца, текст: «Группа» в зависимости от того, по какому признаку необходимо осуществить фильтрацию.

При повторном посещении раздела «Мои возможности» примененные фильтры будут сохранены.

Для быстрого доступа к конкретной возможности можно воспользоваться функционалом поиска. Для этого необходимо ввести в строку поиска наименование искомой возможности и нажать кнопку «Найти».

4.4 Создание информационного объекта в Конструкторе информационных объектов

4.4.1 Описание данных конструктора информационных объектов

Конструктор информационных объектов (ИО) предназначен для построения моделей данных информационных объектов.

Для работы конструктора в системе используются следующие справочники:

- 1) InformationObjectRegister (Таблица 2) (Реестр моделей информационных объектов);
- 2) InformationObjectTypes (Таблица 3) (Типы информационных объектов);
- 3) SubSystems (Подсистемы);
- 4) Systems (Системы);
- 5) DocumentTypes (Типы документов).

Таблица 2 – Описание справочника InformationObjectRegister

Поле	Описание	Примечание
code*	Код модели ИО	
name*	Наименование модели ИО	
type*	Тип ИО	Связь с элементом справочника InformationObjectTypes
system*	Система	Связь с элементом справочника Systems
subsystem*	Подсистема	Связь с элементом справочника SubSystems
url*	Путь к репозиторию в GIT	
filename*	Путь к файлу	
schemaLocation	Глобальный путь	
useVisualEditor*	Использовать графический редактор	

Таблица 3 – Описание справочника InformationObjectTypes

Поле	Описание	Примечание
code*	Код	
name*	Наименование	Выбор из значений: <ul style="list-style-type: none">– Бизнес-объект – предназначен для схем документов прикладных приложений– Модель данных информационного обмена – предназначена для схем информационного

Поле	Описание	Примечание
		обмена между подсистемами – Служебный информационный объект платформы – предназначен для схем документов платформы – Тип прикладной системы – предназначен для комплексных типов прикладных систем – Общеплатформенный тип - предназначен для комплексных типов платформы
typesLibrary*	Библиотека типов	–

4.4.2 Описание витрины «Реестр информационных объектов»

Для доступа к витрине «Реестр информационных объектов» в режиме просмотра используется группа доступа «Информационные объекты - Просмотр SYS_SCHEMA_VIEW», в режиме редактирования: «Информационные объекты - Редактирование SYS_SCHEMA_EDIT».

Витрина «Реестр информационных объектов» наполняется элементами справочника InformationObjectRegister (Рисунок 8).

Рисунок 8 – Наполнение справочника «Реестр информационных объектов»

Рисунок 8 – Наполнение справочника «Реестр информационных объектов»

Панель инструментов витрины «Реестр информационных объектов» содержит следующие функциональные элементы:

- контекстный поиск по содержимому витрины;
- кнопка настройки отображения/скрытия колонок таблицы;
- кнопка расширенного поиска по содержимому витрины;
- кнопка вызова окна создания нового элемента реестра.

Витрина «Реестр информационных объектов» имеет табличный вид с управляющими иконками (Таблица 4).

Таблица 4 – Иконки витрины «Реестр информационных объектов»

Иконка	Наименование	Примечание
	Использовать графический	Признак установлен или нет флаг

Иконка	Наименование	Примечание
	редактор	«Использовать графический редактор» в свойствах информационного объекта
	Редактировать свойства	Переход к окну редактирования свойств информационного объекта (Раздел 4.4.3)
	Редактировать информационный объект	Переход к окну редактирования информационного объекта (Раздел 4.4.4) Иконка не отображается если флаг «Использовать графический редактор» не установлен
	Просмотреть информационный объект	Переход к окну просмотра информационного объекта (Раздел 4.4.5) Иконка не отображается если флаг «Использовать графический редактор» не установлен
	Выгрузить информационный объект	Вызов выгрузки информационного объекта из GIT на локальный компьютер пользователя
	Загрузить информационный объект	Вызов окна загрузки информационного объекта в GIT с локального компьютера пользователя (Раздел 4.4.6)

4.4.3 Окно создания и редактирования свойств информационного объекта

При нажатии кнопки «Создать» открывается модальное окно для добавления нового элемента реестра информационных объектов (Рисунок 9).

Добавление информационного объекта
✕

Код модели ИО *

Наименование модели ИО *

Тип ИО *

Система *

Подсистема *

Типы документов

Путь к репозиторию в GIT *

Путь к файлу *

Глобальный путь

Использовать графический редактор *

Сохранить

Удалить

Отмена

Рисунок 9 – Модальное окно добавления нового объекта

Описание полей модального окна приведено ниже (Таблица 5).

Таблица 5 – Описание полей модального окна

Поле	Наименование
Код модели ИО*	Уникальный код для создаваемой модели информационного объекта Рекомендовано формирование по шаблону <i>Код системы_Код подсистемы_Код типа документа</i>
Наименование модели ИО*	Текстовое поле
Тип ИО*	Выбор из значений справочника InformationObjectTypes
Система*	Выбор из значений справочника Systems
Подсистема*	Выбор из значений справочника SubSystems
Типы документов	Выбор из значений справочника DocumentTypes
Путь к репозиторию в GIT*	Должен содержать путь к папке с приложением без учета домена
Путь к файлу*	Должен содержать путь к файлу внутри папки с приложением
Глобальный путь	Должен содержать путь к общедоступному ресурсу со схемами для подключения

Поле	Наименование
Использовать графический редактор*	Признак

Для добавления нового элемента реестра информационных объектов можно использовать функционал работы со справочником InformationObjectRegister. В этом случае связь между информационным объектом и типом документа необходимо устанавливать самостоятельно через справочник DocumentTypes.

4.4.4 Окно редактирования информационного объекта

Окно редактирования информационного объекта содержит следующие элементы:

- 1) иерархическое дерево элементов;
- 2) блок «Узел схемы»;
- 3) блок «Редактировать элемент»;
- 4) блок «Библиотека типов»;
- 5) кнопка «Проверить валидность»;
- 6) кнопка «Сохранить в GIT».

Иерархическое дерево элементов содержит следующую нотацию (Рисунок 10):

- обязательность элемента – сплошная рамка;
- опциональность элемента – пунктирная рамка;
- множественность элемента – двойная рамка;
- желтый фон – комплексный тип.

Активный элемент при выборе отмечается цветом.

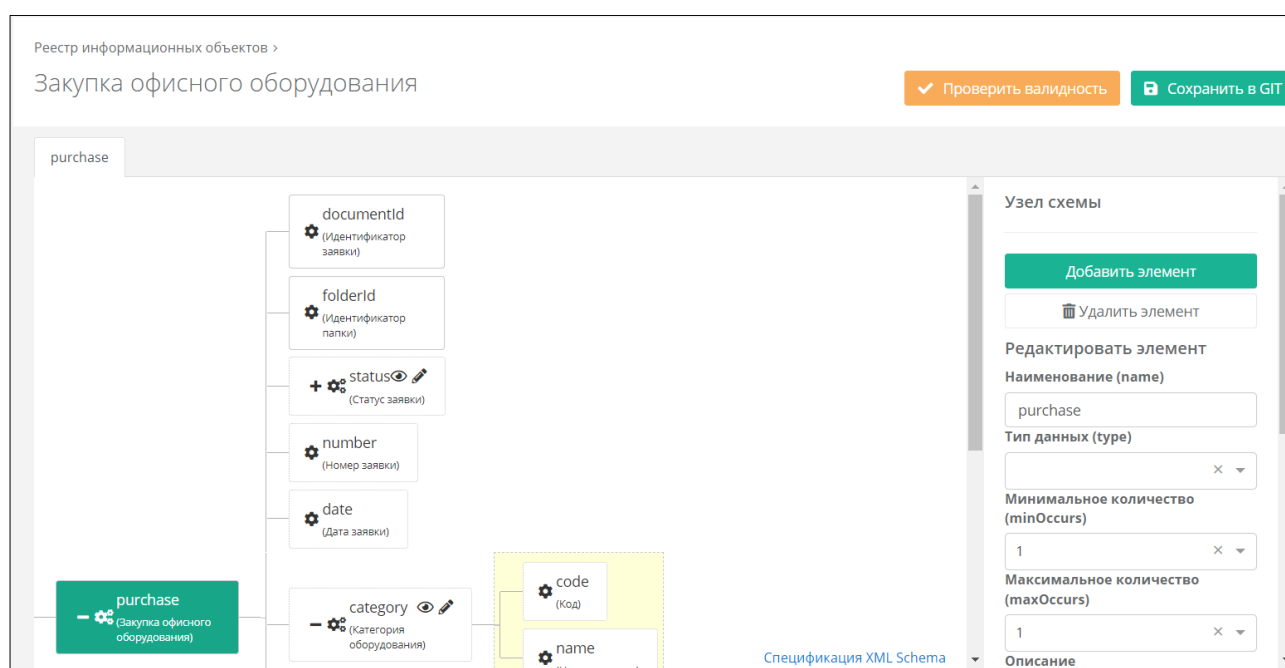


Рисунок 10 – Нотация иерархического дерева элементов

Блок «Узел схемы» содержит кнопки для добавления и удаления элементов модели.

После нажатия кнопки добавления появляется новый элемент, дочерний активному. Кнопка добавления элемента отображается только для узловых (не имеющих типа данных) элементов.

После нажатия кнопки удаления активный элемент удаляется.

Блок «Редактировать элемент» содержит следующие поля для работы с описанием активного элемента:

- Наименование;
- Тип данных;
- Минимальное количество (0 – элемент необязательный, элемент обязательный);
- Максимальное количество (1 – элемент единичный, unbounded – элемент множественный);
- Описание.

Поддерживаются следующие типы данных элементов:

- string;
- integer;
- decimal;
- date;
- dateTime;
- boolean;
- token;
- time;
- base64Binary;
- комплексные типы в рамках текущей XSD и подключенные через include;
- простые типы в рамках текущей XSD и подключенные через include.

Поля блока «Редактировать элемент» заблокированы от изменения, если в качестве активного выбран элемент комплексного типа.

Блок «Библиотека типов» содержит кнопки добавления комплексного и простого типов, список добавленных типов, а также кнопку для включения схемы.

После нажатия кнопки добавления простого или комплексного типа в список добавленных типов добавляется новый тип (Рисунок 11). Рядом с именем типа отображаются иконки просмотра и редактирования. После нажатия на такую иконку открывается новая вкладка, содержащая окно для просмотра или редактирования добавленного типа.

Работа с элементами комплексного типа аналогична работе с элементами иерархического дерева элементов.

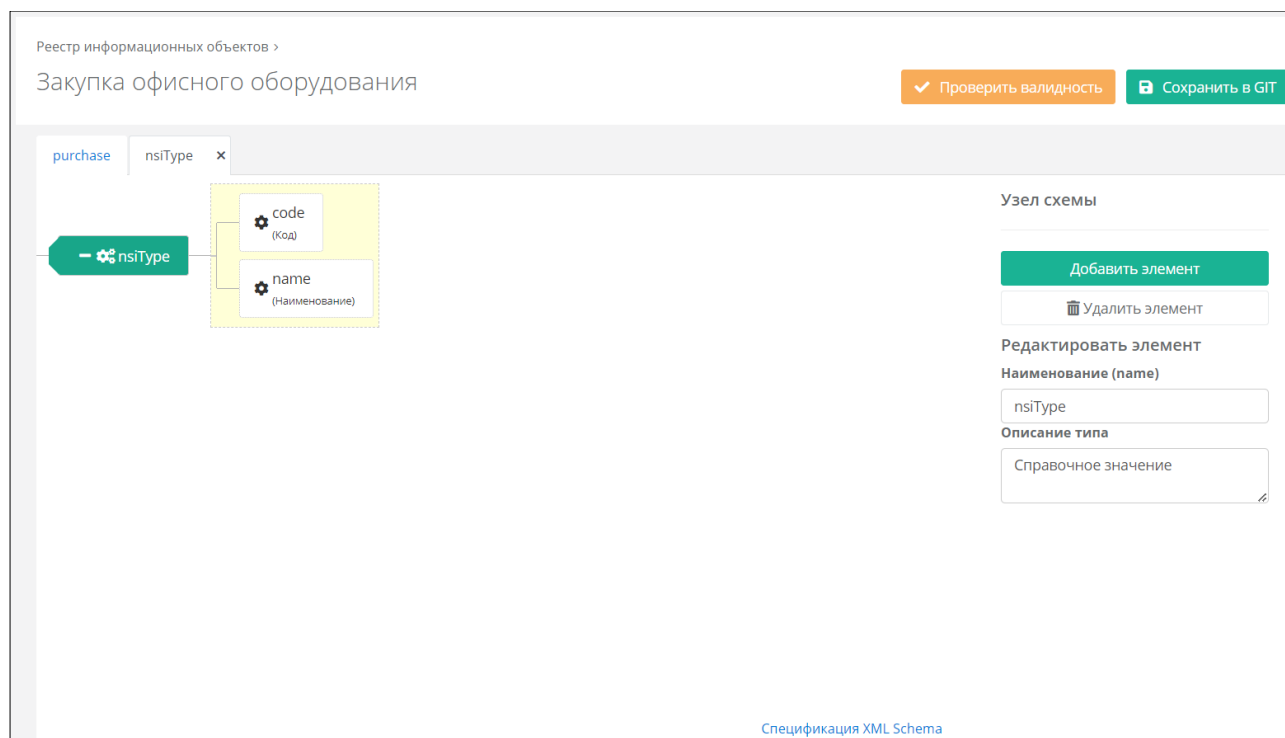


Рисунок 11 – Добавление нового типа элемента nsiType

При работе с параметрами простого типа дополнительно отображаются следующие поля:

- baseType;
- minLength – минимальное число символов;
- maxLength – минимальное число символов;
- patterns – шаблон вводимого значения;
- enumerationValues – допустимые значения.

После нажатия кнопки «Включение схемы (include)» открывается окно для включения схемы (Рисунок 12).

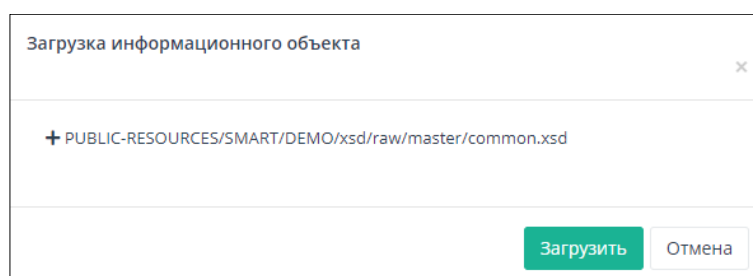


Рисунок 12 – Окно для включения схемы

Включение схемы доступно только для информационных объектов, в свойствах которого заполнено значение «Глобальный путь». В списке для загрузки отображаются только те информационные объекты, тип которых является библиотекой типов, и чья система совпадает с системой текущего информационного объекта.

После нажатия кнопки «Проверить валидность» осуществляется проверка модели информационного объекта на соответствие спецификации XML Scheme. По результатам проверки отображается уведомление о валидности или невалидности схемы.

После нажатия кнопки «Сохранить в GIT» отображается модальное окно для ввода параметров учетной записи и комментария при сохранении модели информационного объекта в GIT (Рисунок 13). Для сохранения в GIT у пользователя должна быть группа доступа «Работа с GIT SYS_GIT_MANAGER».

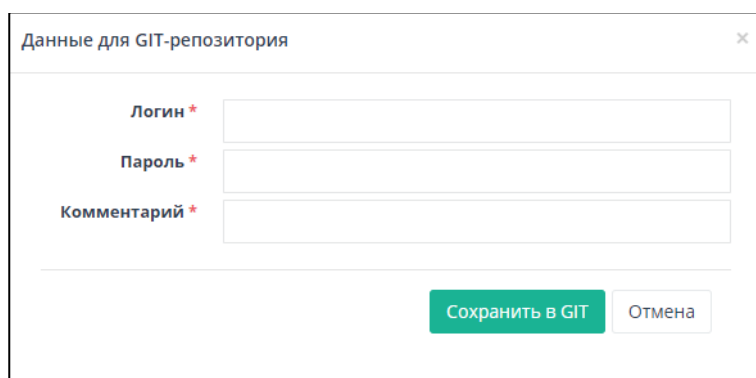


Рисунок 13 – Модальное окно для ввода данных

Примечание: если XSD содержит элементы или атрибуты, не поддерживаемые графическим редактором, просмотр и редактирование XSD в графическом редакторе невозможны. В свойствах таких схем должен быть снят флаг «Использовать графический редактор».

4.4.5 Окно просмотра информационного объекта

Окно просмотра информационного объекта содержит иерархическое дерево элементов схемы аналогичное окну редактирования (Раздел 4.4.4), однако не содержит блоков для добавления/удаления элементов и изменения их свойств (Рисунок 14).

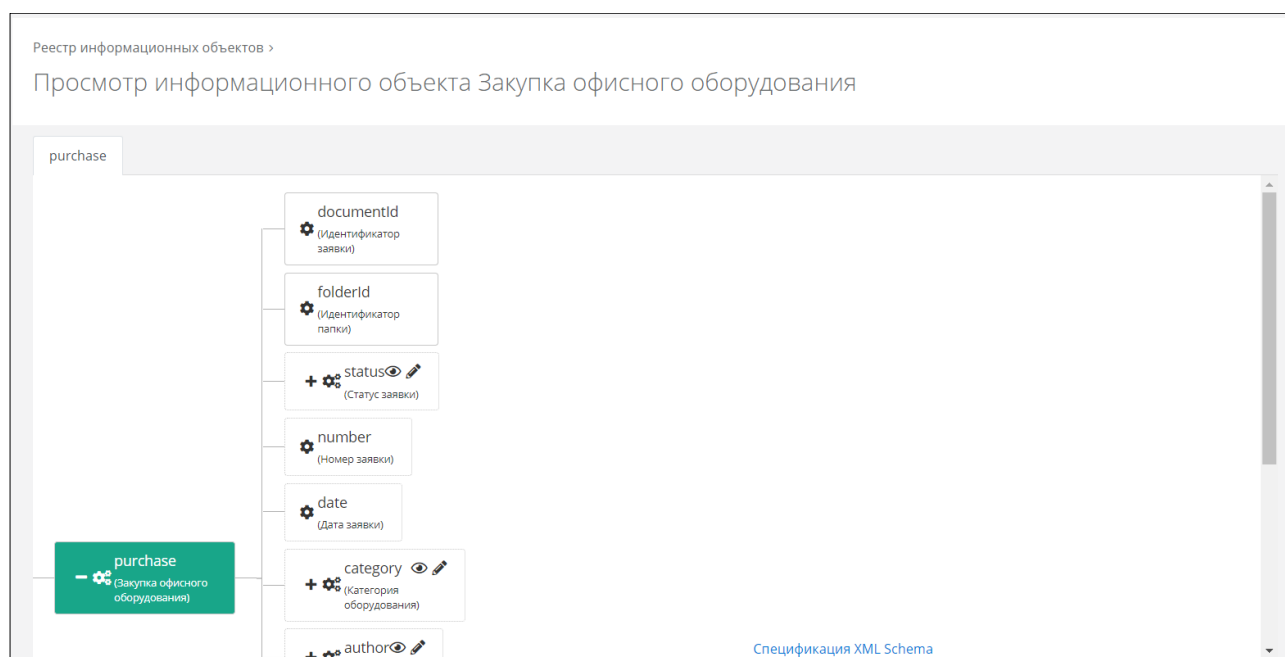


Рисунок 14 – Окно просмотра информационного объекта

4.4.6 Загрузка информационного объекта

Модальное окно «Загрузка информационного объекта» предназначено для загрузки информационного объекта в GIT с локального компьютера пользователя (Рисунок 15).

Файл для загрузки должен отвечать следующим требованиям:

- иметь расширение XSD;
- имя приложенного файла должно совпадать с именем файла из поля «Путь к файлу» в свойствах информационного объекта (без учета пути до файла).

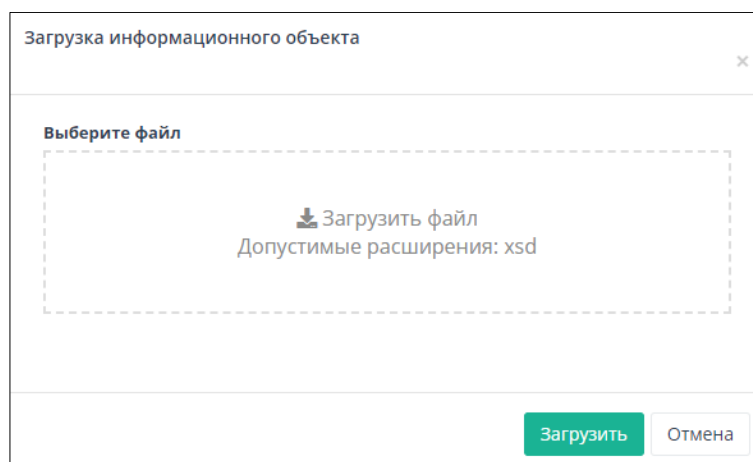


Рисунок 15 – Модальное окно «Загрузка информационного объекта»

После нажатия кнопки «Загрузить» отображается модальное окно для ввода параметров учетной записи и комментария при сохранении модели информационного объекта в GIT. Для сохранения в GIT у пользователя должна быть группа доступа «Работа с GIT SYS_GIT_MANAGER».

4.5 Создание бизнес-процессов в конструкторе БП

4.5.1 Предварительная подготовка

Платформа обеспечивает возможность разработки бизнес-процессов (БП) без установки дополнительных программных средств, с использованием конструктора бизнес-процессов, встроенного в портал платформы. Для корректной разработки и использования бизнес-процессов, необходимо выполнить ряд подготовительных шагов (Таблица 6).

Таблица 6 – Требования к развитию функций Подсистем

№	Описание шага
1	Добавление разработчика бизнес-процессов в группу пользователей SYS_BPMN_VIEW
2	Добавление новой системы в справочник Systems
3	Добавление новой подсистемы в справочник SubSystems

4	Добавление нового типа документов в справочник DocumentTypes
5	Добавление нового бизнес-процесса в справочник BusinessProcessRegistry
6	Добавление нового префикса бизнес-процессов в справочник MyTaskSettings

4.5.1.1 Доступ к конструктору бизнес-процессов

Пользователь должен обладать правами на просмотр реестра бизнес-процессов. Для этого его необходимо включить в группу пользователей: SYS_BPMN_VIEW.

4.5.1.2 Заполнение справочников

При разработке бизнес-процессов для новой системы, предварительно должны быть заполнены следующие справочники:

- Systems (Системы);
- SubSystems (Подсистемы);
- DocumentTypes (Типы документов);
- BusinessProcessRegistry (Реестр конфигураций бизнес-процессов);
- MyTasksSettings (Настройки для витрины «Мои задачи»).

4.5.1.2.1 Справочник Systems

В справочнике Systems должна быть заполнена информация о новой системе. Описание атрибутов справочника приведено ниже (Таблица 7).

Таблица 7 – Описание атрибутов справочника Systems

Наименование атрибута	Описание	Обязательный
code	Код системы	Да
name	Наименование системы	Да
Developer	Наименование организации - разработчика системы	Нет
url	URL ссылка системы	Нет
bpmUrl	URL ссылка на сервер BPM	Нет
alfrescoFolderName	Имя папки в хранилище документов Alfresco	Нет
parent	Имя системы для аналитических отчетов	Нет
description	Описание	Нет
external	Внешняя система	Нет

Пример заполнения справочника Systems приведен ниже (Рисунок 16).

Код системы * code	DEMO
Наименование системы * name	DEMO приложение
Разработчик developer	
URL системы url	/demo/
URL движка БП bpmUrl	/app/demo/bpm
Имя папки в Alfresco alfrescoFolderName	ИС Демо
Имя системы для QLIK parent	
	для аналитики
Описание description	
Внешняя система external	<input type="radio"/> Да <input type="radio"/> Нет <input checked="" type="radio"/> Отсутствует

Рисунок 16 – Пример заполнения справочника Systems

4.5.1.2.2 Справочник SubSystems

В справочнике SubSystems должна быть заполнена информация о новой подсистеме. Описание атрибутов справочника приведено ниже (Таблица 8).

Таблица 8 – Описание атрибутов справочника SubSystems

Наименование атрибута	Описание	Обязательный
code	Код подсистемы	Да
name	Наименование подсистемы	Да
appUrl	Ссылка на приложение	Нет
system	Система	Нет
alfrescoFolderName	Имя папки в хранилище документов Alfresco	Нет

Пример заполнения справочника SubSystems приведен ниже (Рисунок 17).

Код подсистемы * code	DEMO_DEMO
Наименование подсистемы * name	Демо
Ссылка на приложение appUri	/app/demo/demo
Система * system	DEMO DEMO приложение
Имя папки в Alfresco alfrescoFolderName	
Сортировочное значение элемента	

Рисунок 17 – Пример заполнения справочника SubSystems

4.5.1.2.3 Справочник DocumentTypes

В справочнике DocumentTypes должна быть заполнена информация о новом типе документов. Описание атрибутов справочника приведено ниже (Таблица 9).

Таблица 9 – Описание атрибутов справочника DocumentTypes

Наименование атрибута	Описание	Обязательный
code	Код типа документа	Да
name	Наименование типа документа	Да
solrCore	Ядро SOLR для поиска	Нет
fields	Список полей для поиска	Нет
virtTypes	Виртуальные типы документов	Нет
virtual	Виртуальный	Да
subsystem	Подсистема	Да
systemCode	Система	Да
informationObject	Модель ИО	Нет

Пример заполнения справочника DocumentTypes приведен ниже (Рисунок 18).

Код типа *
code DEMO_FORMSDEMO_CONTRACT

Название типа *
name Контракт для демонстрации

Ядро SOLR
solrCore demo_formsdemo Демо - клон динамических форм

Список полей
fields

- x demo_formsdemo_contract_contractDate contractDate
- x demo_formsdemo_contract_typeName typeName
- x demo_formsdemo_contract_siteAOrgName siteAOrgName
- x demo_formsdemo_contract_siteBOrgName siteBOrgName

Виртуальные типы документов
virtTypes

Виртуальный *
virtual Да Нет Отсутствует

Подсистема *
subsystem DEMO_FORMSDEMO Демо для конструктора форм

Система *
systemCode DEMO DEMO приложение

Модель ИО
informationObject

Сортировочное значение элемента

Рисунок 18 – Пример заполнения справочника DocumentTypes

4.5.1.2.4 Справочник BusinessProcessRegistry

В справочнике BusinessProcessRegistry должна быть заполнена информация о новом бизнес-процессе. Информация из справочника используется для формирования витрины «Реестр конфигураций бизнес-процессов», отображаемой при доступе к функционалу конструктора бизнес-процессов. Описание атрибутов справочника приведено ниже (Таблица 10).

Таблица 10 – Описание атрибутов справочника BusinessProcessRegistry

Наименование атрибута	Описание	Обязательный
code	Код бизнес-процесса. Формируется в следующем формате: (в нижнем регистре, без пробелов, латинскими буквами) префикс подпроцесса + '_' + имя подпроцесса. Префикс подпроцесса содержит: код_системы + краткий_код_системы + имя процесса(если не пустое). Символ '_' может использоваться 1 раз для отделения префикса подпроцесса от кода подпроцесса.	Да
name	Наименование бизнес-процесса	Да
docType	Тип документа, по которому разрабатывается процесс	Да

Наименование атрибута	Описание	Обязательный
subsystem	Подсистема	Да
system	Система	Да
exists	Выходы из бизнес-процесса	Нет
url	Путь к репозиторию в GIT	Нет
filename	Имя файла, должно совпадать с кодом бизнес-процесса	Нет

Пример заполнения справочника BusinessProcessRegistry приведен ниже (Рисунок 19).

The screenshot shows a web form for adding a new business process to the BusinessProcessRegistry. The fields are as follows:

- Код *** (code): demo_Demo_prc. Below the field is the formula: КОД_СИСТЕМЫ.toLowerCase() + КРАТКИЙ_КОД_ПОДСИСТЕМЫ.toLowerCase() + (ИМЯ_ПРОЦЕССА != null ? ИМЯ_ПРОЦЕССА : '') + '_' + ИМЯ_ПОДПРОЦЕССА
- Наименование *** (name): Демо-процесс
- Тип документа *** (docType): DEMO_DEMO_DEMO Демо-документ
- Подсистема *** (subsystem): DEMO_DEMO Демо
- Система *** (system): DEMO DEMO приложение
- Описание** (description): (empty)
- Выходы** (exists): (empty)
- Путь к репозиторию в GIT** (url): BPM/DEMO/DEMO/demo
- Путь к файлу** (filename): demo_Demo.bpmn. Below the field is the note: Имя файла должно совпадать с кодом (code+".bpmn")

Рисунок 19 – Пример заполнения справочника BusinessProcessRegistry

4.5.1.2.5 Справочник MyTaskSettings

В справочнике MyTaskSettings должна быть заполнена информация о новом бизнес-процессе для витрины «Мои задачи». Информация из справочника используется для отображения пользовательских задач.

Описание атрибутов справочника приведено ниже (Таблица 11).

Таблица 11 – Описание атрибутов справочника MyTaskSettings

Наименование атрибута	Описание	Обязательный
code	Код префикса бизнес-процесса	Да
name	Наименование бизнес процесса	Нет
system	Система	Да

Наименование атрибута	Описание	Обязательный
searchUrl	Базовый URL поиска	Да
tasksUrl	Базовый URL задач	Да
contentGroovy	Содержание	Нет
showTasks	Показывать список задач	Нет

Пример заполнения справочника MyTasksettings приведен ниже (Рисунок 20).

✕

Редактирование элемента

Префикс бизнес процесса * code

Стандарт заполнения - кодсистемыкраткийкодподсистемы

Наименование бизнес процесса name

Система * system

Выбор значения из справочника "Системы"

Базовый URL поиска * searchUrl

Обычно '/seach'

Базовый URL задач * tasksUrl

Обычно '/кодсистемы'

Содержание contentGroovy

item.document ? ((item.document.purchase_number ? "Заявка на закупку
офисного оборудования № " + item.document.purchase_number: "") +
(item.document.purchase_date ? " от " + item.document.purchase_date.split('T')
[0].split('-').reverse().join('.') + (item.document.purchase_authorFio ? "
" +
item.document.purchase_authorFio: "")): ""

Показывать список задач showTasks Да Нет Отсутствует

Сортировочное значение элемента

Сохранить
⚙️
🗑️
Отмена

Рисунок 20 – Пример заполнения справочника MyTaskSettings

4.5.1.3 Сервисные функции бизнес-процессов

При разработке процессов возникает необходимость вызова внешнего сервиса, организации рассылки почтовых уведомлений, записи или запроса информации из json документа. Для этого в конструкторе бизнес-процессов используются сервисные функции. Для реализации сервисных функций используется язык groovy. Сервисные функции по функционалу группируются в библиотеки (Рисунок 21).

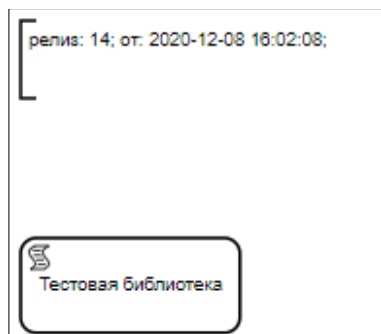


Рисунок 21 – Пример библиотеки сервисных функций

Библиотеки сервисных функций по каждому из проектов собраны в хранилища библиотек.

Существует 2 вида хранилищ сервисных groovy-функций:

1) Системное хранилище – библиотеки универсальных сервисных функций хранятся в коде сервера bpm;

2) Пользовательское хранилище – библиотеки сервисных функций хранятся в определениях бизнес-процессов специального вида: «кодсистемы_наименованиехранилища_Libs.bpmn». Хранилище должно быть общим для всех процессов системы (Рисунок 22). Такие бизнес-процессы не предназначены для запуска. Отдельные сервисные функции библиотек запускаются скриптовыми задачами (Script Task).

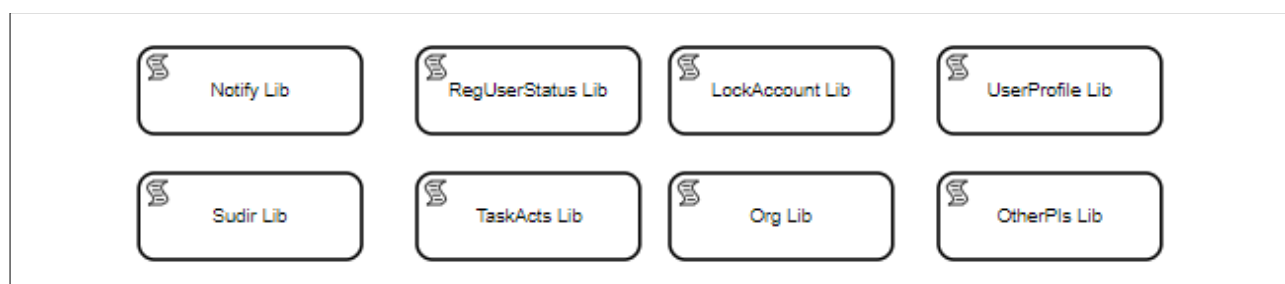


Рисунок 22 – Пример пользовательского хранилища библиотек сервисных groovy-функций

Системное хранилище сервисных функций размещено в коде сервера bpm. Системному хранилищу библиотек присвоен код: *cdp_Libs*. Библиотеки, на которые загрузчик скриптов ссылается без префикса, также запрашиваются из системного хранилища библиотек сервисных функций (если они не переопределены в пользовательском хранилище библиотек сервисных функций).

Системное хранилище содержит следующие библиотеки функций:

- ApprovalLibId – библиотека содержит сервисные функции для подпроцессов согласования;

- AssignLibId – библиотека содержит сервисные функции для расчета ответственного исполнителя (или списка кандидатов) задачи процесса, предназначенные для использования в create-событиях пользовательских задач;

- DueDateLib – библиотека содержит сервисные функции для расчета срока

исполнения задачи процесса, производимого с учетом производственного календаря и расписания рабочего времени из справочников МДМ;

– ErrLibId – библиотека содержит стандартные обработчики исключительных ситуаций, дополняющие стек ошибок информацией о текущем вызове (включающей имена и идентификаторы текущего экземпляра процесса и задачи). Обработчики логируют ошибки в стандартном для платформы формате (и со стандартным префиксом ERROR ru.reinform.script:);

– MailLibId – библиотека содержит функции рассылки почтовых сообщений из задач процесса через платформенное приложение mail (включая нотификацию пользователей о назначении исполнителем/кандидатом в исполнители пользовательской задачи процесса);

– MdmLibId – библиотека содержит функции получения элементов справочников МДМ по значению атрибута. Структура элемента справочника определяется структурой конкретного справочника;

– OtherPisLibId – библиотека содержит функции поиска параллельных экземпляров процесса (того же определения), совпадающих с текущим по значениям переменных (как правило - переменной EntityIdVar, то есть экземпляров процессов, связанных с тем же информационным объектом, что и текущий);

– StarterLibId – библиотека содержит функции запуска процесса с передачей параметров из процесса-инициатора. Запущенный экземпляр процесса не становится дочерним по отношению к текущему. Для запуска дочернего экземпляра процесса используются задачи типа Call activiti.

Перечень функций из системной библиотеки ApprovalLibId представлен ниже (Таблица 12).

Таблица 12 – Перечень функций из системной библиотеки ApprovalLibId

№	Наименование функции, параметры вызова	Описание
1	setApprovalVars(execution, jsonRoot, agreedFn, approvalUsersVarName = 'ApprovalUsersVar', approvalDurationsVarName = 'ApprovalDurationsVar')	Получение и обработка (сохранение в переменных экземпляра процесса) списка согласующих и сроков согласования agreedFn - функция, принимающая на вход jsonRoot, пример: jsonRoot -> jsonRoot.agr.stage.find { stage -> stage?.code == 'refuse' } }.preparedDocument.find { preparedDocument -> preparedDocument?.docType?.code == 'serviceRefuseLetter' } }.approval.approvalCycle.agreed
2	beforeApproval(task, approvalApprovedVarName = 'ApprovalApprovedVar')	Очищает переменную, сохраняющую результат согласования. Параметр task – текущая пользовательская задача approvalApprovedVarName имя переменной, сохраняющей результат согласования
3	afterApproval(task, approvalRegistryVarName = 'ApprovalRegistryVar', approvalUserVarName = 'ApprovalUserVar', approvalApprovedVarName = 'ApprovalApprovedVar')	Заполняет историю согласования. Параметр task - текущая пользовательская задача Параметр approvalRegistryVarName - имя переменной истории согласования Параметр approvalUserVarName - имя переменной, содержащей текущего согласующего Параметр approvalApprovedVarName - имя переменной, сохраняющей результат согласования

Перечень функций из системной библиотеки AssignLib представлен ниже (Таблица 13).

Таблица 13 – Перечень функций из системной библиотеки AssignLib

Наименование функции, параметры вызова	Описание
calcAssignees(task)	Расчет ответственного исполнителя (или списка кандидатов) задачи на основании кандидатской группы. Функция вызывается в обработчике create-события задачи процесса и предполагает наличие у задачи одной кандидатской группы. Если в ней один пользователь, то он назначается ответственным исполнителем. Если список - они формируют список кандидатов. Параметр функции – task текущая пользовательская задача

Перечень функций из системной библиотеки DueDateLib представлен ниже (Таблица 14).

Таблица 14 – Перечень функций из системной библиотеки DueDateLib

№	Наименование функции, параметры вызова	Описание
1	setDueDate(task)	<p>Расчет срока исполнения задачи, с учетом производственного календаря. Функция вызывается в обработчике create-события задачи процесса. Использует длительность, задаваемую через параметр Due date задачи: Если значение явно задано, или вычислено через выражение (начинается с \$) - то значение даты не меняется. Если длительность задана в часах (PTNH) или рабочих днях (PND) - то срок пересчитывается по алгоритму добавления рабочих часов или дней Параметр задачи task текущая пользовательская задача</p>
2	setDueDateVar(execution, shiftDuration, varName)	<p>Пересчет временного интервала (от текущего момента времени), с учетом производственного календаря. Рассчитанная длительность сохраняется в переменной с именем varName текущего экземпляра процесса. Параметр execution текущий исполняемый фрагмент процесса; Параметр shiftDuration - длительность временного интервала; Параметр varName имя переменной для сохранения результата;</p>

Перечень функций из системной библиотеки ErrLib представлен ниже (Таблица 15).

Таблица 15 – Перечень функций из системной библиотеки ErrLib

№	Наименование функции, параметры вызова	Описание
1	taskErrCalcAndLog(exception, task, fnName, extraParams = [:])	<p>Обработка исключительных ситуаций для пользовательских задач, оборачивается в универсальную исключительную ситуацию платформы, с сохранением параметров контекста (включая параметры экземпляра процесса и исполняемой задачи), логируется в log ru.geinform.script и возвращается вызывающей функции (откуда может быть проброшена вверх оператором throw). Пример вызова в блоке catch (exception) { throw taskErrCalcAndLog(exception, task, 'demoreconstructor.PlanDate.calcPlanDate') } Параметр exception обрабатываемая</p>

№	Наименование функции, параметры вызова	Описание
		<p>исключительная ситуация Параметр task текущая пользовательская задача Параметр fnName имя исполняемой функции (рекомендуется уточнять идентификаторами хранилища и библиотеки: 'demoreconstructor.PlanDate.calcPlanDate) Параметры extraParams дополнительные параметры (ассоциативный массив), включаемые в сообщение об ошибке (обычно - входные параметры функции fnName)</p>
2	<p>executionErrCalcAndLog(exception, execution, fnName, extraParams = [:])</p>	<p>Обработка и запись в лог исключительной ситуации в контексте исполняемого фрагмента процесса(execution, в т.ч. скриптовых задач) экземпляра процесса. Оборачивается в универсальную исключительную ситуацию платформы, с сохранением параметров контекста (включая параметры экземпляра процесса), логируется в лог ru.reinform.script и возвращается вызывающей функции (откуда может быть проброшена вверх оператором throw). Пример вызова в блоке catch (exception) { throw executionErrCalcAndLog(exception, execution, 'demoreconstructor.PlanDate.calcPlanDate') } Параметр exception обрабатываемая исключительная ситуация Параметр execution текущий исполняемый фрагмент Параметр fnName имя исполняемой функции (рекомендуется уточнять идентификаторами хранилища и библиотеки: 'demoreconstructor.PlanDate.calcPlanDate) Параметр extraParams дополнительные параметры , включаемые в сообщение об ошибке (обычно - входные параметры функции fnName)</p>

Перечень функций из системной библиотеки MailDateLib представлен ниже (Таблица 16).

Таблица 16 – Перечень функций из системной библиотеки MailDateLib

№	Наименование функции, параметры вызова	Описание
1	<p>getMailInfo(execution, users)</p>	<p>Получение информации (ФИО, почтовый адрес) о пользователях по массиву логинов. Информация запрашивается в mdm. Возвращаемые данные имеют следующий формат:</p>

№	Наименование функции, параметры вызова	Описание
		<p>[fio: ФИО, email: почтовый_адрес]</p> <p>Параметр execution – текущий исполняемый фрагмент</p> <p>Параметр users – список логинов пользователей</p> <p>Возвращает массив записей с ФИО и почтовыми адресами пользователей</p>
2	sendMails(execution, subject, body, to)	<p>Отправка почтового сообщения пользователям. Отправка осуществляется через платформенное приложение mail.</p> <p>Параметр execution – текущий исполняемый фрагмент</p> <p>Параметр subject – заголовок письма</p> <p>Параметр body – текст письма</p> <p>Параметр to – массив адресатов формата [fio: ФИО, email: почтовый_адрес]</p>
3	sendMail(execution, subject, body, email, fio)	<p>Отправка почтового сообщения пользователю. Отправка осуществляется через платформенное приложение mail.</p> <p>Параметр execution – текущий исполняемый фрагмент</p> <p>Параметр subject – заголовок письма</p> <p>Параметр body – текст письма</p> <p>Параметр email – почтовый адрес адресата</p> <p>Параметр fio – ФИО адресата</p>
4	notifyAssignee(task)	<p>Уведомление пользователя о назначении пользовательской задачи процесса.</p> <p>Используется в событиях assignment и create пользовательской задачи.</p> <p>Уведомление содержит наименование процесса, задачи, а также срок ее исполнения.</p> <p>Параметр task – текущая пользовательская задача</p>
5	notifyCandidates(task)	<p>Уведомление пользователей о назначении кандидатов в исполнители пользовательской задачи процесса.</p> <p>Используется в событии create пользовательской задачи.</p> <p>Уведомление содержит наименование процесса, задачи, а также срок ее исполнения.</p> <p>Параметр task – текущая пользовательская задача</p>
6	notifyAssigneeOrCandidates(task)	<p>Уведомление исполнителя или кандидатов в исполнители пользовательской задачи процесса.</p> <p>Используется в событии create пользовательской задачи, если событие assignment не обрабатывается.</p> <p>Уведомление содержит наименование процесса, задачи, а также срок ее исполнения.</p> <p>Параметр task текущая пользовательская задача</p>

Перечень функций из системной библиотеки MdmLib представлен ниже (Таблица 17).

Таблица 17 – Перечень функций из системной библиотеки MdmLib

№	Наименование функции, параметры вызова	Описание
1	findMdmElements(execution, nickDict, nickAttr, valueAttr)	Поиск массива элементов словаря МДМ по значению поискового атрибута. Результатом поиска может быть пустой массив или массив из нескольких элементов. Параметр execution текущий исполняемый фрагмент Параметр nickDict справочник Параметр nickAttr поисковый атрибут Параметр valueAttr значение поискового атрибута Возвращает массив элементов словаря МДМ
2	getMdmElement(execution, nickDict, nickAttr, valueAttr)	Получение элемента словаря МДМ по значению поискового атрибута. Элемент должен быть один и только один (прочие случаи трактуются как исключительные ситуации). Параметр execution текущий исполняемый фрагмент Параметр nickDict справочник Параметр nickAttr поисковый атрибут Параметр valueAttr значение поискового атрибута Возвращает элемент словаря МДМ

Перечень функций из системной библиотеки OtherPisLib представлен ниже (Таблица 18).

Таблица 18 – Перечень функций из системной библиотеки OtherPisLib

№	Наименование функции, параметры вызова	Описание
1	findOtherProcessInstances(execution, keyVarNames)	Поиск экземпляров процесса, совпадающих с текущим по значениям переменных и идентификатору определения процесса. Параметр execution – текущий исполняемый фрагмент Параметр keyVarNames- массив имен переменных, по значениям которых идет поиск Возвращает массив экземпляров процесса (ProcessInstance)
2	checkOtherProcessInstance(execution,	Проверка наличия экземпляров процесса,

№	Наименование функции, параметры вызова	Описание
	keyVarNames, isAnyActiveProcessVarName)	совпадающих с текущим по значениям переменных и идентификатору определения процесса Результат проверки (true/false) сохраняется в переменной текущего экземпляра процесса. Параметр execution – текущий исполняемый фрагмент Параметр keyVarNames – массив имен переменных, по значениям которых идет поиск Параметр isAnyActiveProcessVarName – имя переменной, в которую записывается результат проверки

Перечень функций из системной библиотеки StarterLib представлен ниже (Таблица 19).

Таблица 19 – Перечень функций из системной библиотеки StarterLib

Наименование функции, параметры вызова	Описание
startNextProcess(execution, nextProcessDefinitionKey, extraVars = [:])	Запуск процесса с передачей параметров из текущего процесса. Используется актуальная версия определения процесса. При запуске передаются: переменные, запрашиваемые запускаемым процессом через форму стартового события, значения которых определены в текущей задаче переменные Entity*Var (EntityIdVar, EntityDescriptionVar), значения которых определены в текущей задаче; переменные из списка extraVars (с наивысшим приоритетом, перезаписывают автоматически определяемые переменные из предыдущих пунктов) Параметр execution – текущий исполняемый фрагмент Параметр nextProcessDefinitionKey – ключ запускаемого процесса Параметр extraVars – список дополнительных переменных и их значений, передаваемых через стартовую форму запускаемого процесса

4.5.1.3.1 Добавление настроек для вызовов из процессов

Для вызова сервисов из процессов необходимо внести информацию в конфигурационный файл: *systemname_bpm.yml*, где *systemname* – это код разрабатываемой системы.

В конфигурационный файл прописывается *alias* пути до вызываемых сервисов.

4.5.1.3.2 Добавление хранилища библиотек пользовательских функций

Для добавления хранилища библиотек пользовательских функций необходимо выполнить следующую последовательность действий:

1) открыть вкладку «Конфигурации бизнес-процессов», создать новый бизнес-процесс;

2) заполнить атрибуты формы создания нового бизнес-процесса (Рисунок 23);

Создание бизнес-процесса

Код процесса: demo_Libs

Наименование процесса: Библиотеки демо-приложений

Система: ДЕМО приложение

Подсистема: Демо

Тип документа: Демо-документ


Путь к репозиторию в GIT

Путь к файлу

Имя файла должно совпадать с кодом (code+'.bpmn')

Сохранить Отмена

Рисунок 23 – Пример заполнения для Демо-приложения

3) открыть добавленный процесс на редактирование, затем добавить новую скриптовую задачу (Рисунок 24). Для этого в меню инструментов выберем блок создания новой задачи: , нажмем левой кнопкой мыши на добавленный блок, в открывшемся меню выберем «Изменить тип», затем – «Задача-сценарий»;

4) в правой области экрана изменим значение атрибута «Идентификатор» на TestLibId. Первую часть идентификатора (Test) нужно заменить на подходящую по теме, идентификатор функции должен быть уникальным в рамках проекта. Заполним атрибуты «Имя», язык скрипта (groovy). Если функция должна вызываться из пользовательской задачи, то в параметры вызова добавим «task», если из контекста процесса (отдельной скриптовой задачи), то в параметры вызова функции добавим «execution». В поле «Скрипт» добавим функцию, например, для чтения json документа.

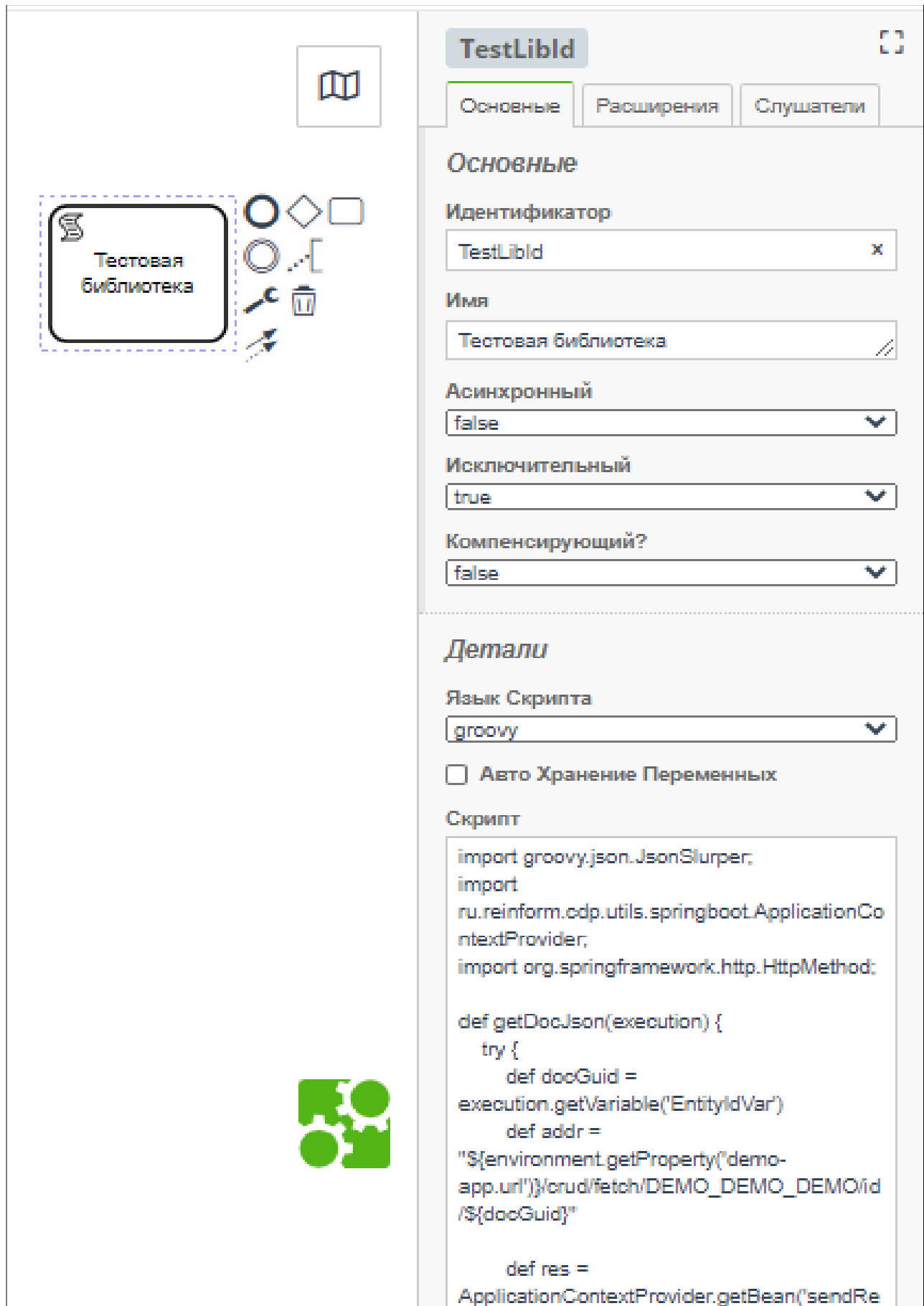


Рисунок 24 – Добавление новой скриптовой задачи

В примере выше функция будет вызываться из отдельной скриптовой задачи, в параметрах вызова прописан «execution». Пример реализации функции чтения json заявки

представлен ниже:

```
import groovy.json.JsonSlurper;
import ru.reinform.cdp.utils.springboot.ApplicationContextProvider;
import org.springframework.http.HttpMethod;
def getDocJson(execution) {
    try {
        def docGuid = execution.getVariable('EntityIdVar')
        def addr = "${environment.getProperty('demo-
app.url')}/crud/fetch/DEMO_DEMO_DEMO/id/${docGuid}"
        def res =
        ApplicationContextProvider.getBean('sendRestUtils').sendJsonRequestServiceAuth(addr.to
String(), HttpMethod.GET, null, String.class)
        def slurper = new JsonSlurper()
        def docJson = slurper.parseText(res)
        return docJson;
    } catch (exception) {
        throw executionErrCalcAndLog(exception, execution, 'Test.getDocJson')
    }
}
```

При использовании метода `ApplicationContextProvider.getBean('sendRestUtils').sendJsonRequestServiceAuth` для отправки REST запросов необходимо импортировать библиотеки:

- `ru.reinform.cdp.utils.springboot.ApplicationContextProvider;`
- `org.springframework.http.HttpMethod.`

Для отправки запроса GET нужно заранее заполнить конфигурацию атрибут конфигурации `demo-app.url` (см. раздел «Добавление настроек для вызовов из процессов»).

Добавим новую функцию для вызова из пользовательских задач, например, переопределим функцию вычисления планового срока выполнения задач.

Для этого добавим новую «Задачу-сценарий». Идентификатор функции будет совпадать с системной функцией `DueDate – DueDateLibId`:

```
import java.time.*
import java.time.format.DateTimeFormatter
import ru.reinform.cdp.utils.springboot.ApplicationContextProvider
import org.springframework.http.HttpMethod
def setDueDate(task) {
    def execution = task.getExecution()
    try {
        def localDateTime =
        DateTimeFormatter.ISO_LOCAL_DATE_TIME.format(LocalDateTime.now())
        def shiftDurationOrExactDueDate =
        task.getTaskDefinition().getDueDateExpression().toString()
        if (shiftDurationOrExactDueDate.startsWith('$') {
            shiftDurationOrExactDueDate = (new GroovyShell(this.binding)).evaluate('"' +
            shiftDurationOrExactDueDate + '"')
        }
        Date dueDate
```

```

    if (shiftDurationOrExactDueDate.startsWith('P')) {//duration
def addr
    if (shiftDurationOrExactDueDate.startsWith('PT')) {
        addr =
"$environment.getProperty('mdm.url')}/api/v1/businessCalendar/addWorkHours/${localDateTime}
}/${shiftDurationOrExactDueDate}"
    } else {
        def shiftDays = Duration.parse(shiftDurationOrExactDueDate).toDays()
        addr =
"$environment.getProperty('mdm.url')}/api/v1/businessCalendar/addWorkDays/${localDateTime}
}/${shiftDays}"
    }
    def res =
ApplicationContextProvider.getBean('sendRestServiceAuthUtils').sendJsonRequest(addr.toString(),
HttpMethod.GET, null, String.class)
    LocalDateTime localDueDate = LocalDateTime.parse(res)
    ZonedDateTime zonedDueDate = localDueDate.atZone(ZoneId.systemDefault())
    dueDate = Date.from(zonedDueDate.toInstant())
    } else {//exact date
dueDate = Date.from(OffsetDateTime.parse(shiftDurationOrExactDueDate).toInstant())
    }
    task.setDueDate(dueDate)
    } catch (exception) {
        throw taskErrCalcAndLog(exception, task, 'DueDate.setDueDate')
    }
}
}

```

В параметре функции определен контекст вызова «task» – пользовательская задача. Вызов обработчика исключительных ситуаций, в данном случае рассчитан на работу с пользовательскими задачами (taskErrCalcAndLog(exception, task, 'DueDate.setDueDate')), в данном случае так же подключен для работы.

Функция вызывается в create-событии задачи и вычисляет срок выполнения задачи, в зависимости от параметра «Максимальный срок» (Рисунок 25).

Если введенное значение начинается с «P», то вычисление производится по правилу вычисления промежутка времени, в днях (начинается с «P», вызывается сервис расчета рабочих дней) или в часах (начинается с «PT», вызывается сервис расчета рабочих часов). В противном случае, срок передан в формате дата/время, и он подставляется без дополнительных расчетов.

Скриншот панели настроек с вкладками «Основные», «Свойства Формы» и «Слушатель». Вкладка «Основные» содержит следующие параметры:

- Идентификатор: ExecuteTest_ID
- Имя: Выполнить тест
- Асинхронный: false
- Исключительный: true
- Компенсирующий?: false

Вкладка «Детали» содержит следующие параметры:

- Исполнитель: (пустое поле)
- Кандидаты В Пользователи: (пустое поле)
- Группы Кандидатов: DEMO_CREATE
- Ключ Формы: ExecuteTest
- Максимальный Срок: PT4H
- Приоритет: (пустое поле)

Рисунок 25 – Параметр «Максимальный срок»

Для завершения необходимо нажать кнопку «Действия», затем – «Обновить в системе».

4.5.1.3.3 Вызов пользовательских функций из процессов

Функцию можно вызвать из скриптовой или пользовательской задачи, синтаксис вызова, при этом отличается.

Для вызова пользовательских функций из скриптовых задач откроем процесс на редактирование. Добавим новую скриптовую задачу, как это делалось выше. Вместо реализации функции, добавим вызов из библиотеки: `evaluate(ru.reinform.activiti.utils.ScriptUtils.calc(execution,'Err,Test',"getDocJson(execution);"))`.

В левой части вызова указаны библиотеки функций, которые используются в функции чтения документа: «Test» и «Err» (для обработки исключительных ситуаций). В правой части вызова указана функция и ее параметр (`execution` – фрагмент процесса) (Рисунок 26).

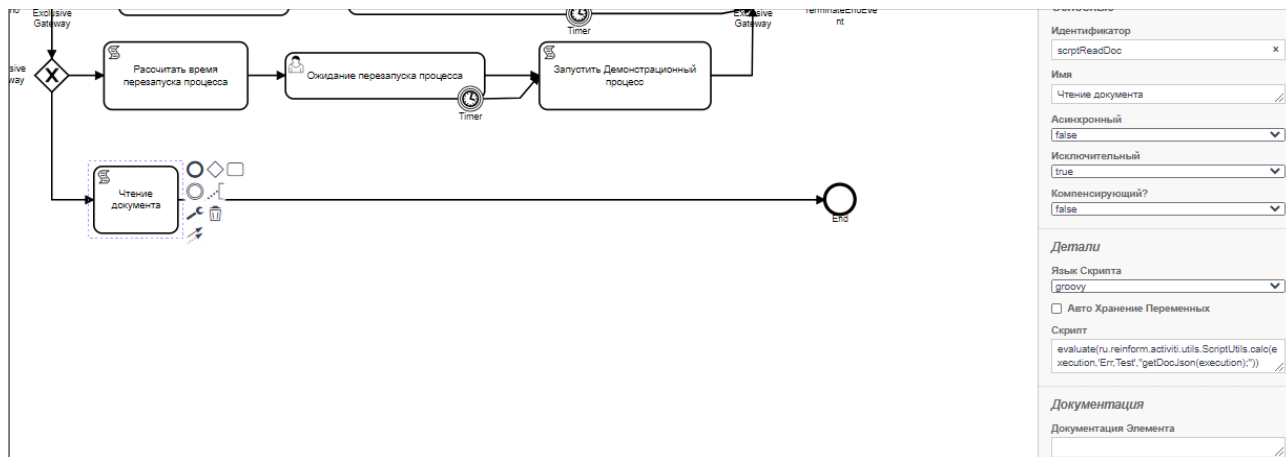


Рисунок 26 – Скрипт вызова функции

Для вызова пользовательских функций из пользовательской задачи, нажмем левой кнопкой мыши на задачу. Откроем вкладку «Слушатели», добавим нового слушателя (Рисунок 27).

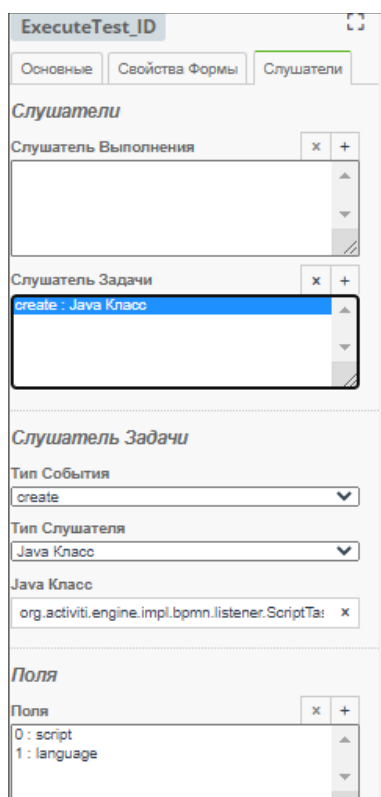


Рисунок 27 – Добавление нового слушателя

Заполним следующие атрибуты слушателя:

- Тип события: «create»;
- Тип слушателя: «Java Класс»;
- Java Класс: org.activiti.engine.impl.bpmn.listener.ScriptTaskListener.

Добавим два новых поля и заполним их атрибуты:

- 1) поле «script»:
 - Имя: «script»;

- Тип: «Выражение»;
- Значение: `evaluate(ru.reinform.activiti.utils.ScriptUtils.calc(task,'Err,DueDate','setDueDate(task);'))`;

2) поле «language»:

- Имя: «language»;
- Тип: «строка»;
- Значение: «groovy».

В вышеприведенном примере будет вызываться функция расчета планового срока завершения задачи. В параметре вызова функции указан атрибут «task» – значит, функция будет выполняться в контексте пользовательской задачи.

4.5.2 Основные этапы разработки в конструкторе бизнес-процессов

Основные этапы разработки в конструкторе процессов приведены ниже (Таблица 20).

Таблица 20 – Основные этапы разработки

№	Описание шага	Комментарии
1	Разработка диаграммы бизнес-процесса в графическом редакторе	Предварительная проработка требований, выявление, определение ролевой модели бизнес-процесса, необходимых справочников
2	Определение ролевой модели, необходимой для реализации бизнес-процесса	Подготовка списка групп пользователей
3	Создание необходимых групп пользователей в Idap, в соответствии с ролевой моделью. Добавление пользователей в группы	Создание и заполнение групп пользователей, в соответствии с ролевой моделью
4	При необходимости, добавление дополнительных справочников	Добавление новых справочников, например справочника статусов подготовки документов
5	Проектирование бизнес-процесса в конструкторе бизнес-процессов	Создание и публикация спецификаций бизнес-процессов, на базе диаграмм, разработанных в шаге 1

4.5.2.1 Разработка диаграммы бизнес-процесса в графическом редакторе

Диаграммы бизнес-процессов создаются в графическом редакторе, и являются постановкой на создание ролевой модели и разработку бизнес-процессов (Рисунок 28).

Проектирование осуществляется в Microsoft Visio (допускается и другое).

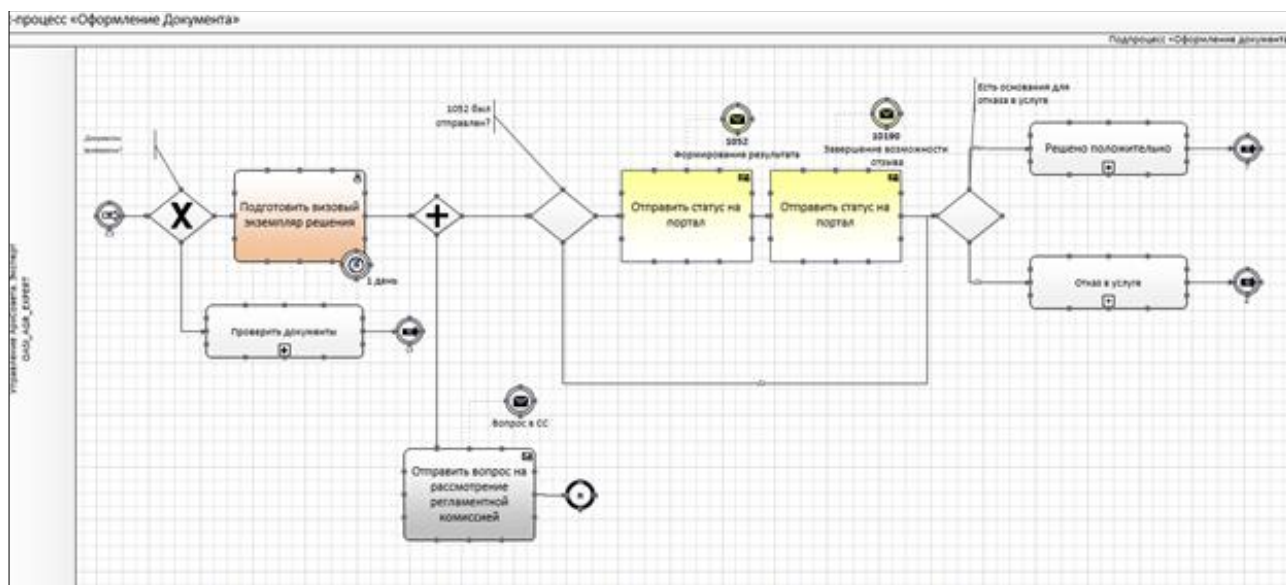


Рисунок 28 – Пример постановки

4.5.2.2 Определение ролевой модели

В диаграмме бизнес-процесса пользовательские задачи, размещаются в BPMN элементах типа pool/lane. В левой части элемента указывается группа пользователей, или исполнитель, на которого должны быть назначены задачи, находящиеся в этом пуле.

Группы пользователей-участников бизнес-процессов должны быть заведены в LDAP, в следующем порядке:

- 1) создаем новые ou (organisation unit) для Системы и подсистемы;
- 2) в ou для системы/подсистемы добавляем новый ou для группы пользователей (на русском языке);
- 3) в ou для группы пользователей добавляется cn (common name) для кода группы (на латинском языке) (Рисунок 29);

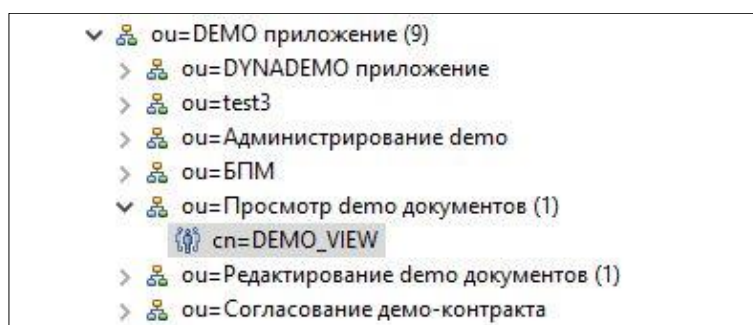


Рисунок 29 – Пример параметра cn

- 4) в cn добавляем одного или более пользователей, которых необходимо включить в группу. В cn должен быть добавлен хотя бы один member (пользователь). Для каждой подсистемы в новую группу добавляем сервисного пользователя (serviceuser_subsystemcode) (Рисунок 30).

cn	DEMO_VIEW
member	uid=ivanov_ii,ou=Префектура Центрального Административного Округа,ou=Организация,ou=Пользо...
member	uid=serviceuser,ou=Технологические аккаунты,dc=cdp,dc=local

Рисунок 30 – Пример параметра serviceuser

4.5.2.3 Проектирование бизнес-процесса в конструкторе бизнес-процессов

Подробная инструкция по работе пользователей с конструктором бизнес-процессов, добавлению пользовательских задач и прочих элементов, представлена в разделе «Инструкция по работе пользователей с редактором бизнес-процессов».

Рассмотрим создание демонстрационного бизнес-процесса (Рисунок 31).

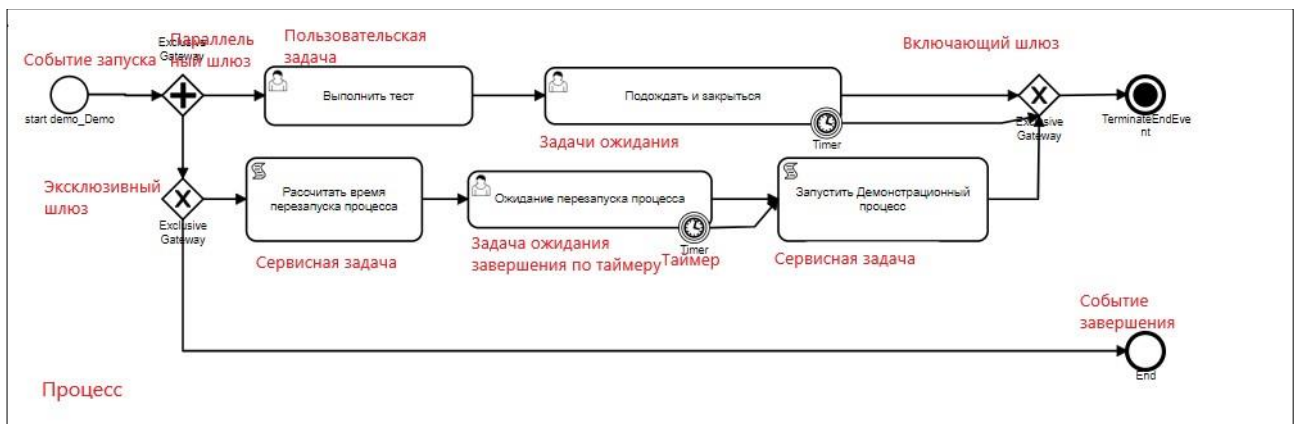


Рисунок 31 – Диаграмма демонстрационного процесса

4.5.2.3.1 Настройка атрибутов процесса

Заполним атрибуты процесса Идентификатор, Имя и Документация Элемента. Идентификатор должен начинаться с префикса системы (Рисунок 32).

demo_Demo

Основные | Расширения

Основные

Идентификатор: demo_Demo

Имя: Демонстрационный процесс

Асинхронный: true

Исключительный: true

Компенсирующий?: true

Детали

Запускать Могут Пользователи:

Разделены запятыми

Запускать Могут Группы:

Разделены запятыми

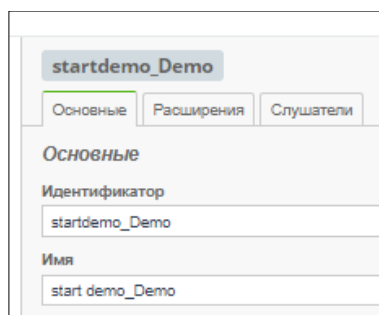
Документация

Документация Элемента: Демонстрационный процесс

Рисунок 32 – Атрибуты процесса: Идентификатор, Имя и Документация Элемента

4.5.2.3.2 Событие запуска процесса

Заполним атрибуты события запуска процесса Идентификатор и Имя (Рисунок 33).

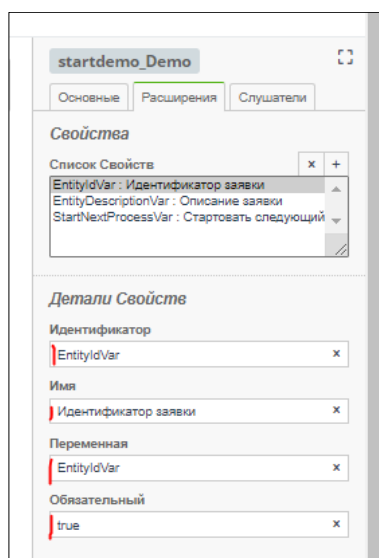


The screenshot shows a configuration window for 'startdemo_Demo'. It has three tabs: 'Основные', 'Расширения', and 'Слушатели'. The 'Основные' tab is active. Under the heading 'ОСНОВНЫЕ', there are two input fields: 'Идентификатор' with the value 'startdemo_Demo' and 'Имя' with the value 'start demo_Demo'.

Рисунок 33 – Атрибуты события запуска процесса Идентификатор и Имя

. Добавим атрибуты процесса, для этого откроем вкладку «Расширения». Обязательный параметр EntityIdVar, при запуске процесса в него сохраняем идентификатор документа, связанного с процессом.

При добавлении атрибута необходимо заполнить поля Идентификатор, Имя, Переменная, Обязательный (Рисунок 34).



The screenshot shows the 'startdemo_Demo' configuration window with the 'Расширения' tab active. A 'Свойства' section contains a list of properties: 'EntityIdVar: Идентификатор заявки', 'EntityDescriptionVar: Описание заявки', and 'StartNextProcessVar: Стартовать следующий'. Below this, the 'Детали Свойств' section shows four fields: 'Идентификатор' (EntityIdVar), 'Имя' (Идентификатор заявки), 'Переменная' (EntityIdVar), and 'Обязательный' (true).

Рисунок 34 – Поля Идентификатор, Имя, Переменная, Обязательный

4.5.2.3.3 Параллельный шлюз

Задачи в ветках, исходящих из параллельного шлюза, будут назначаться параллельно.

4.5.2.3.4 Пользовательская задача

Заполним атрибуты Идентификатор, Имя, Ключ формы, Максимальный срок (PT4H = 4 часа). Укажем группу, на которую будет назначена задача (DEMO_CREATE) (Рисунок 35).

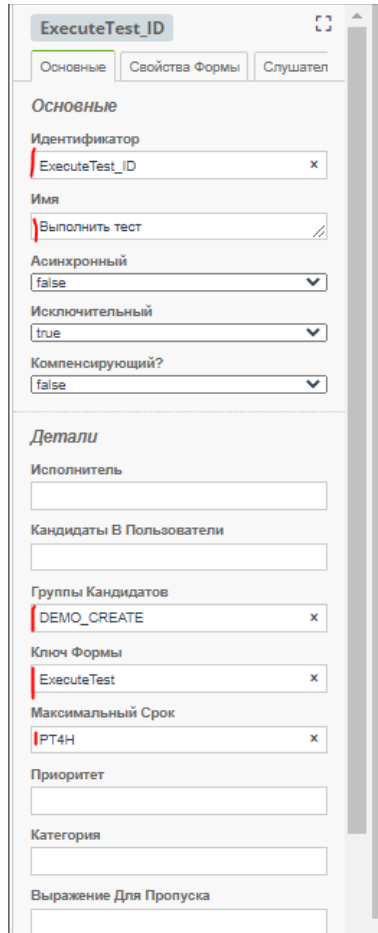


Рисунок 35 – Атрибуты Идентификатор, Имя, Ключ формы, Максимальный срок

Добавим расчет кандидатов выполнения задачи и срока завершения задачи, для этого откроем вкладку Слушатели. Добавим параметр «Слушатель Задачи» с атрибутами «Тип события» (create), «Тип слушателя» (Java Класс), «Java Класс» (org.activiti.engine.impl.bpmn.listener.ScriptTaskListener) (Рисунок 36).

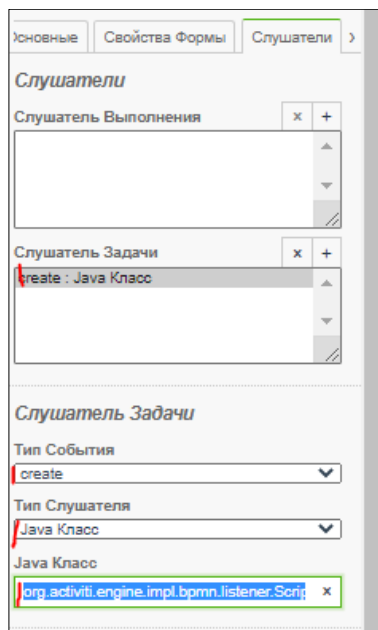


Рисунок 36 – Параметр «Слушатель Задачи» с атрибутами
Добавим поля «script» и «language» с учетом следующего:

1) Имя: script, Тип: выражение, Значение: evaluate(ru.reinform.activiti.utils.ScriptUtils.calc(task,'Err,DueDate,Assign','calcAssignees(task);setDueDate(task);')) (Рисунок 37);

2) Имя: language, Тип: строка, Значение: groovy.

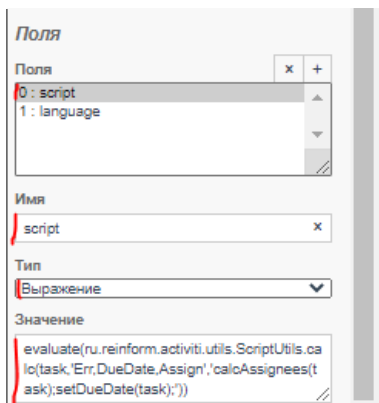


Рисунок 37 – Параметры поля «script»

4.5.2.3.5 Сервисная задача

В сервисной задаче вызовом сервис расчета времени завершения задачи ожидания. Сервис реализован в системном хранилище библиотек. Заполним следующие атрибуты сервисной задачи:

- Идентификатор: scrptCalcProcessRestart;
- Имя: Рассчитать время перезапуска процесса;
- Язык скрипта: groovy;
- скрипт: def duration имеет значение 'PT15M'

evaluate(ru.reinform.activiti.utils.ScriptUtils.calc(execution,'Err,DueDate','setDueDateVar(execution,'\${duration}','TimerDateVar');')).

Переменная TimerDateVar будет использована далее в задаче ожидания.

4.5.2.3.6 Задача ожидания

Целью добавления задач ожидания является отображение текущего шага процесса в витринах. Задача ожидания реализуется без формы. Заполним следующие атрибуты задачи ожидания (Рисунок 38):

- Идентификатор: scrptCalcProcessRestart;
- Имя (необходимо рассчитать время перезапуска процесса);
- Максимальный срок: подставляется рассчитанное значение из переменной TimerDateVar (\${TimerDateVar}).

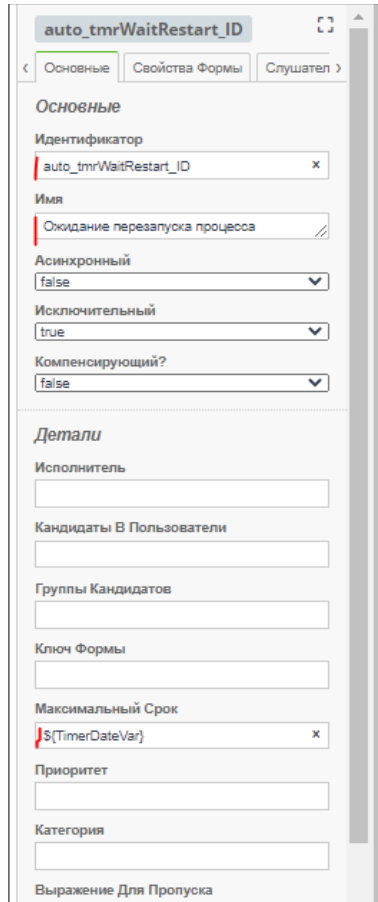


Рисунок 38 – Атрибуты задачи ожидания

4.5.2.3.7 Таймер

Для настройки таймера в поле «Тип Определения Таймера» выберем значение «Дата», в поле «Определение Таймера» подставим рассчитанное значение из переменной TimerDateVar ($\{TimerDateVar\}$) (Рисунок 39).

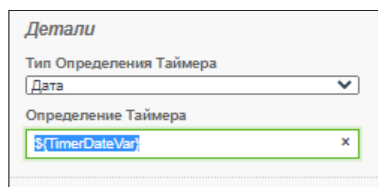


Рисунок 39 – Настройки таймера

4.5.2.3.8 Сервисная задача запуска процесса

Для сервисной задачи заполним атрибуты: Идентификатор (scrpt_StartDemoProcess), Имя (Запустить Демонстрационный процесс), Язык Скрипта (groovy), Скрипт:

```
def processDefinitionKey = 'demo_Demo_prc'
evaluate(ru.reinform.activiti.utils.ScriptUtils.calc(execution,'Err,Starter',"startNextProcess(
execution,'${processDefinitionKey}',[StartNextProcessVar:false]);"))
```

Задача запустит новый экземпляр процесса *demo_Demo_prc* с входными параметрами, как у текущего процесса (Идентификатор заявки: EntityIdVar).

4.5.2.3.9 Включающий шлюз

При прохождении любой из веток процесса через шлюз остальные ветки текущего

экземпляра процесса завершаются при помощи *TerminateEndEvent*. Если *Terminate End Event* заменить на *End Event*, то процесс не завершится, пока остальные ветки не будут завершены.

4.5.2.3.10 Событие завершения

Служит для завершения процесса. В конце каждой из веток процесса добавляем отдельный параметр: *End Event*.

4.5.3 Инструкция по работе пользователей с редактором бизнес-процессов

В инструкции приведены примеры использования функционала редактора бизнес-процессов на площадке разработки. Для работы с другими площадками необходимо изменить адрес ссылки для входа в систему.

4.5.3.1 Порядок входа в редактор бизнес-процессов

В левой панели портала выбрать раздел «Настройки» (⚙️), далее нажать на «Конструкторы» - «Бизнес-процессы».

Откроется «Реестр конфигураций бизнес-процессов» (Рисунок 40).

Код процесса *	Наименование процесса *	Система *	Подсистема *	Тип документа *	Файл
aktsign_dinBezSafelinkpr	Связь между документом и PDF (обычно используется в архиве)	ОКСИ	КТР	Заявление КТР	
aktsign_dokument	Создание и печать	Получение ДС	ДС: Получение	Получение ДС	
aktsign_ukolnogrampr	Исполнение подумов	Получение ДС	ДС: Получение	Получение ДС	
aktsign_ukolnogramprnd	Тестовый процесс с нуля	Получение ДС	ДС: Получение	Получение ДС	aktsign_ukolnogramprnd.bpm
aktsign_ukolnogramprvskidprkazprkazSafelinkpr	Создание документа	Получение ДС	ДС: Получение	Получение ДС	aktsign_ukolnogramprvskidprkazprkazSafelinkpr.bpm
aktsign_dinpr	Демо-процесс	DEMO получение	Демо	Демо-документ	aktsign_dinpr.bpm
aktsign_dinprDocument	Подготовка документа	DEMO получение	Демо для конфигураций	Демо-документ	aktsign_dinprDocument.bpm
aktsign_dinprDocument	Демонстрация функциональности элементов редактора BPM	DEMO получение	Демо для конфигураций	Счет для демонстрации	aktsign_dinprDocument.bpm
aktsign_dinprDocument	Подготовка счета для демонстрации	DEMO получение	Демо для конфигураций	Счет для демонстрации	aktsign_dinprDocument.bpm
aktsign_dinprDocument	Подготовка и подписание контракта	DEMO получение	Демо для конфигураций	Контракт для демонстрации	aktsign_dinprDocument.bpm

Рисунок 40 – Реестр конфигураций бизнес-процессов

4.5.3.2 Создание новой конфигурации бизнес-процесса

В реестре конфигураций БП можно создать новую конфигурацию процесса или отредактировать ранее созданную.

Для создания нового БП нажмем на кнопку «Создать». Откроется модальное окно «Создание бизнес-процесса» (Рисунок 41).

Создание бизнес-процесса

Код процесса *

Наименование процесса *

Система *

Подсистема *

Тип документа *

Путь к репозиторию в GIT

Путь к файлу

Имя файла должно совпадать с кодом (code*.bpm)

Сохранить Отмена

Рисунок 41 – Модальное окно «Создание бизнес-процесса»


Заполним следующие основные атрибуты нового бизнес-процесса:

- Код процесса – формируем по правилу: ‘код системы’ + ‘краткий код подсистемы’ + ‘имя БП’+’_’+’имя подпроцесса’. Имя процесса может быть пропущено. Код процесса до нижнего подчеркивания (‘_’) должен быть заполнен латинскими буквами в нижнем регистре. Для имени подпроцесса можно использовать латинские символы как в нижнем, так и в верхнем регистре;
- Наименование процесса – заполняем на русском языке в соответствии с постановкой;
- Система – в выпадающем списке выбираем систему, к которой относится новый процесс;
- Подсистема – в выпадающем списке выбираем подсистему в соответствии с выбранной системой;
- Тип документа – в выпадающем списке выбираем тип документа, по которому должен быть разработан БП.

Остальные атрибуты не обязательны к заполнению.

После нажатия кнопки «Сохранить» атрибуты нового бизнес-процесса будут сохранены в реестре конфигураций БП.

4.5.3.3 Разработка/редактирование диаграммы бизнес-процесса

Для редактирования диаграммы бизнес-процесса нажмем кнопку «Редактировать процесс» (). Откроется интерфейс web-редактора бизнес-процессов (Рисунок 42).

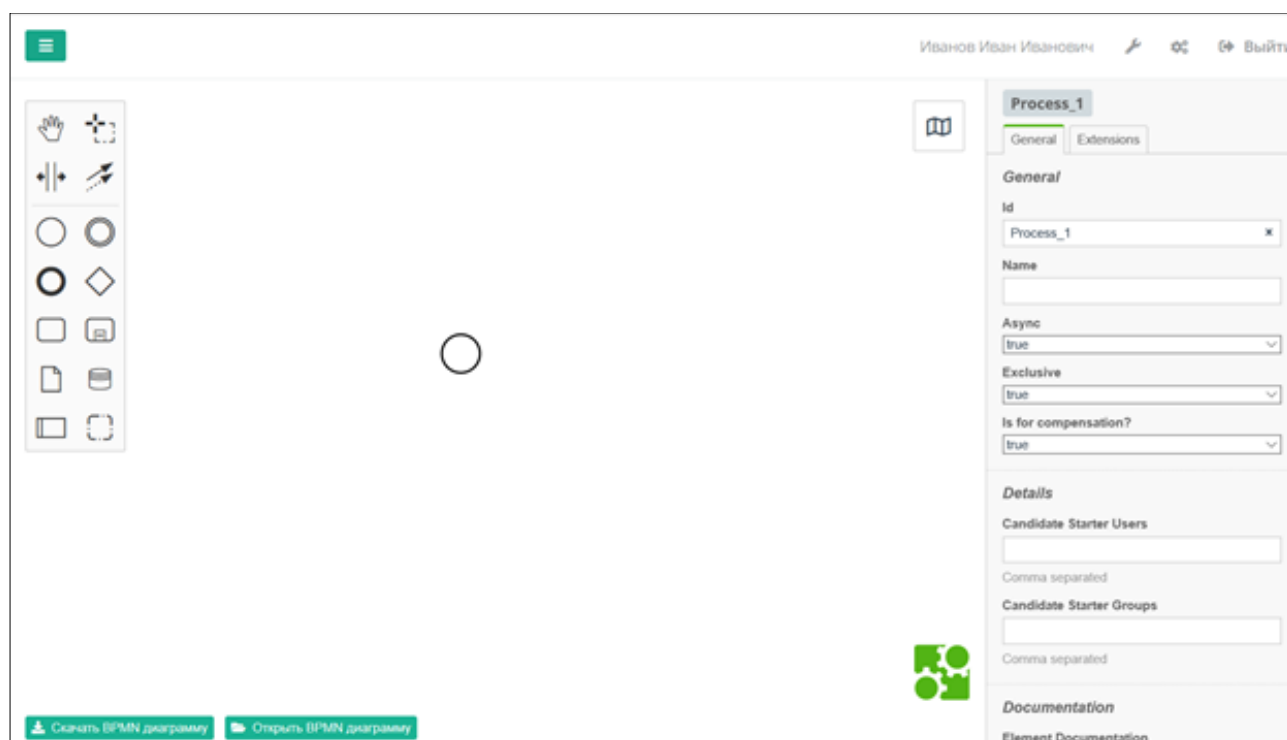



Рисунок 42 – Интерфейс web-редактора бизнес-процессов

Редактор бизнес-процессов обеспечивает выполнение следующих функций:

- 1) разработка нового бизнес-процесса;
- 2) редактирование спецификаций бизнес-процессов, загруженных в редактор процессов из BPMN-файлов (кнопка «Открыть BPMN-диаграмму»);
- 3) сохранение разработанного/отредактированного бизнес-процесса в BPMN-файле на локальном компьютере (кнопка «Скачать BPMN диаграмму»);
- 4) сохранение разработанного/отредактированного бизнес-процесса в репозитории бизнес-процессов (кнопка «Сохранить BPMN диаграмму»);
- 5) обновление спецификации бизнес-процесса в системе – сохранение и публикация спецификации бизнес-процесса на сервере бизнес-процессов (кнопка «Обновить в системе»).

Вышеперечисленные функции доступны по нажатию кнопки «Действия» ().

4.5.3.3.1 Сохранение спецификации бизнес-процесса на локальном компьютере

Для сохранения спецификации бизнес-процесса на локальном компьютере, нажмем кнопку «Действия». В выпадающем списке выберем «Скачать BPMN диаграмму» (Рисунок 43).

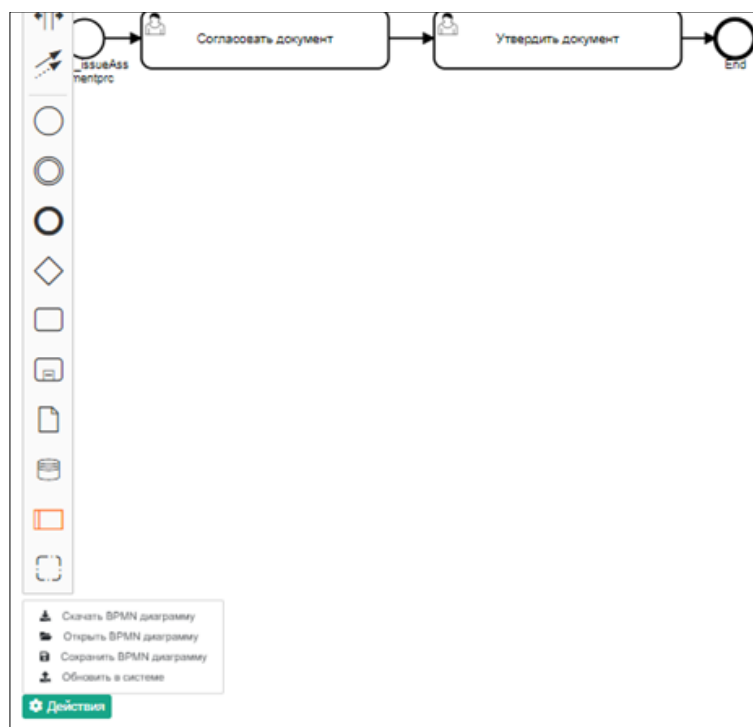


Рисунок 43 – Выпадающий список

После завершения редактирования бизнес-процесса нажимаем кнопку «Скачать BPMN-диаграмму» под схемой бизнес-процесса (Рисунок 44).

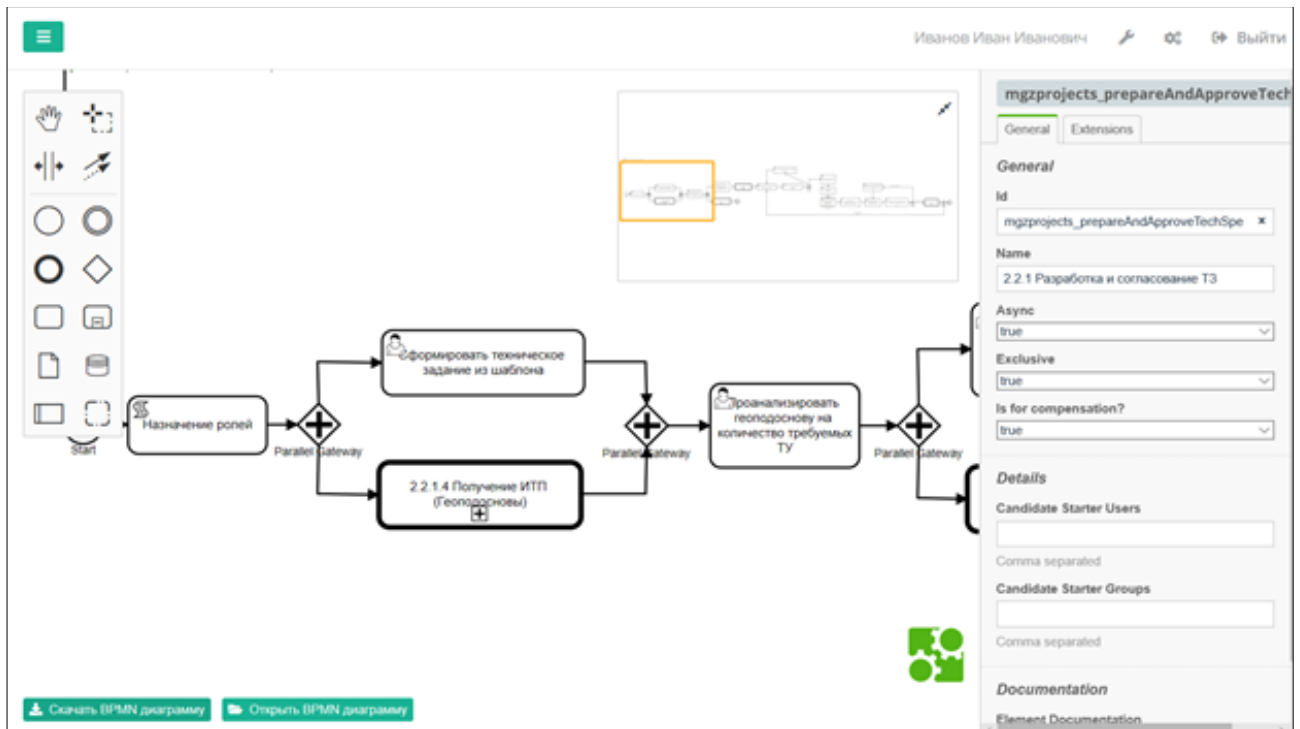


Рисунок 44 – Кнопка «Скачать BPMN-диаграмму»

В модальном окне будет предложено заполнить атрибут пространство имен «Namespace» (менять необязательно), ввести имя скачиваемого файла и нажать кнопку «Скачать» (Рисунок 45).

The dialog box titled 'Скачать диаграмму' contains two input fields. The first is labeled 'Namespace' and contains the text 'http://bpmn.io/schema/bpmn'. The second is labeled 'File Name' and contains 'demo_demoProcess|bpmn'. At the bottom right, there are two buttons: 'Скачать' and 'Отмена'.

Рисунок 45 – Кнопка «Скачать»

4.5.3.3.2 Открытие спецификации бизнес-процесса с локального компьютера

Для того, чтобы открыть спецификацию бизнес-процесса, в web-редакторе нажмем кнопку «Действия». В выпадающем списке выберем «Открыть BPMN диаграмму» (Рисунок 46).

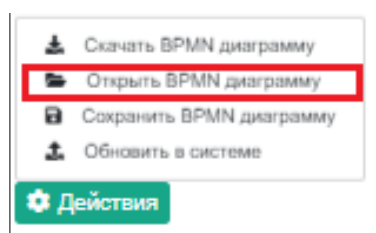


Рисунок 46 – Кнопка «Скачать»

Откроется модальное окно выбора файла с локального компьютера (Рисунок 47).

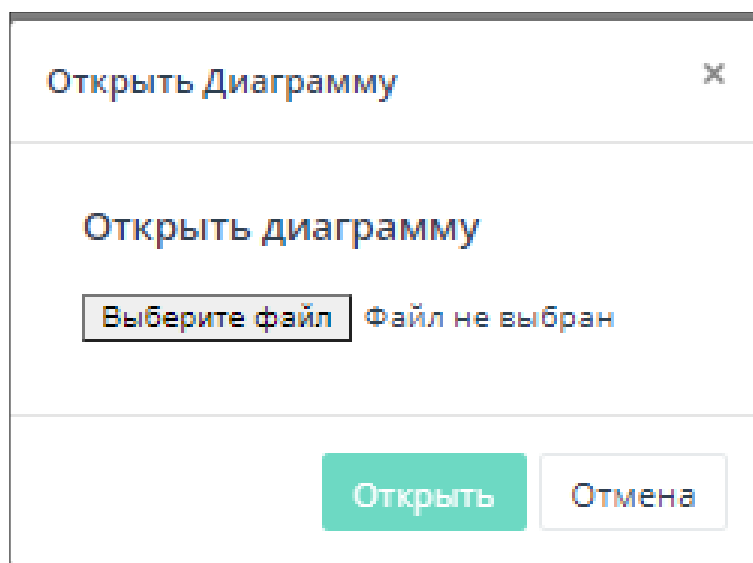


Рисунок 47 – Модальное окно выбора файла

После выбора файла необходимо нажать кнопку «Открыть», и выбранная диаграмма будет загружена в web-редактор.

4.5.3.3.3 Сохранение спецификации бизнес-процесса

Для того, чтобы сохранить изменения в спецификации бизнес-процесса, нажмем кнопку «Действия», в выпадающем списке выберем «Сохранить BPMN диаграмму» (Рисунок 48).

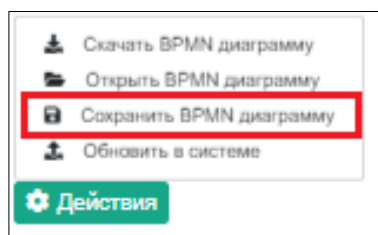


Рисунок 48 – Выпадающий список

Изменения диаграммы будут сохранены в репозитории процессов.

4.5.3.3.4 Обновление спецификации бизнес-процесса

Для того, чтобы обновить текущую версию бизнес-процесса, нажмем кнопку «Действия», в выпадающем списке выберем «Обновить в системе» (Рисунок 49).

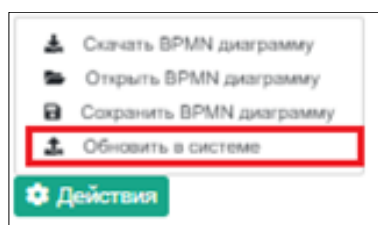


Рисунок 49 – Выпадающий список

Бизнес процесс будет сохранен в репозитории процессов, и обновлен в системе.

4.5.3.4 Инструменты разработки/редактирования бизнес-процессов

В левой части экрана расположена панель элементов бизнес-процессов. В верхней части панели находятся основные инструменты разработки процессов в составе:

- «Activate the hand tool» – ручной режим работы с диаграммой, предназначен для упрощенной навигации по диаграмме процесса, без изменения местоположения элементов диаграммы, которые попадают под курсор мыши;
- «Activate the lasso tool» – режим лассо предназначен для выбора нескольких элементов диаграммы, которые впоследствии можно переместить в другую часть диаграммы, или удалить;
- «Activate the create/remove space tool» – режим добавления/удаления лишнего пространства диаграммы, позволяет добавить или удалить дополнительное пространство для размещения элементов диаграммы по горизонтали или вертикали;
- «Activate the global connect tool» – режим добавления связей между элементами диаграмм. Для добавления связи необходимо выбрать данный режим, затем удерживая левую кнопку мыши провести стрелку между двумя элементами диаграммы.

4.5.3.5 Элементы диаграммы бизнес-процесса

Элементы диаграммы бизнес-процесса находятся ниже инструментов разработки. Элементы перетаскиваются в область редактирования мышкой, посредством drag&drop.

Основные элементы диаграммы процесса включают:


- Событие запуска БП «StartEvent» – предназначено для добавления событий запуска бизнес-процесса;
- Промежуточное событие «Intermediate/Boundary Event» – предназначено для добавления промежуточных событий бизнес-процесса, например, таймеров;
- Событие завершения «EndEvent» – предназначено для добавления событий завершения бизнес-процесса;
- Шлюз «Gateway» – предназначен для добавления параллельных, инклюзивных и эксклюзивных шлюзов, а также шлюзов на основе событий и сложных шлюзов;
- Задача «Task» – предназначена для добавления пользовательских, сервисных и скриптовых задач, запуска подпроцессов;
- Развернутый подпроцесс «expanded SubProcess» – предназначен для добавления подпроцессов в спецификацию бизнес-процесса;
- Ссылка на объект данных «DataObjectReference» – предназначен для добавления ссылок на объект данных;
- Ссылка на хранилище данных «DataStoreReference» – предназначен для

добавления ссылок на хранилище сохраненной информации;

– Пул/Ответственный «Pool/Participant» – предназначен для визуализации границ бизнес-процесса, отображения групп и пользователей, выполняющих задачи бизнес-процесса;

– Группа «Group» – предназначена для группировки графических элементов, принадлежащих одной и той же категории, для повышения простоты восприятия спецификации бизнес-процесса.

4.5.3.5.1 Событие запуска БП

Событие запуска БП служит для обозначения начала процесса. Для добавления события запуска нажмем на элемент «Create StartEvent» () и перетащим его в область редактирования процесса.

У события можно заполнить значения атрибутов «Id» и «Name» закладки «General» (Рисунок 50).

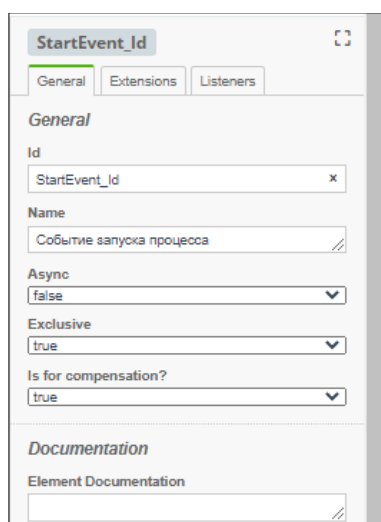


Рисунок 50 – Закладка «General»

В событии «StartEvent» чаще всего определяется сквозная переменная процесса «EntityIdVar», по которой осуществляется связь между процессом и документом, по которому был запущен процесс.

Для добавления переменной в событие запуска откроем вкладку «Extensions»

Добавим в событие запуска атрибут «EntityIdVar», тип данных «string». Для этого нажмем левой кнопкой мыши на значок добавить «+» в поле «Properties List». Выберем добавленный атрибут, при этом, в блоке «Property Details» отобразятся поля, которые нужно заполнить.

Значения полей «Id» и «Variable» могут совпадать и должны быть заполнены на латинском языке. В поле «Name» введем наименование поля на русском языке, которое будет понятно конечным пользователям. Атрибут «Required» – логический, введем значение

«true» (поле обязательно для заполнения) (Рисунок 51).

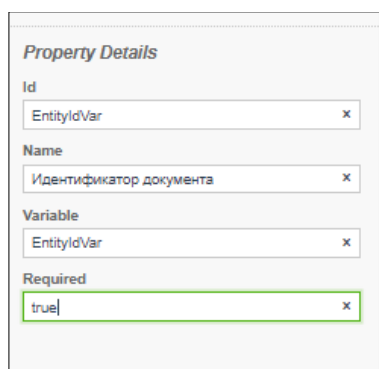



Рисунок 51 – Значение «true»

При необходимости, можно в событие запуска добавить дополнительные атрибуты.

4.5.3.5.2 Промежуточное событие

Событие предназначено для добавления промежуточных событий бизнес-процесса, например, таймеров

Для добавления промежуточного события, нажмем на элемент «Create Intermediate/Boundary Event» () и перетащим его в область редактирования процесса.

Для выбора типа промежуточного события нажмем левой кнопкой мыши на элемент диаграммы, затем на значок «Change Type» (Рисунок 52).



Рисунок 52 – Значок «Change Type»

Выберем тип события «Timer Intermediate Catch Event». Заполним атрибуты добавленного события «Id» и «Name» на закладке свойств «General». Выберем тип определения таймера («Timer Definition Type») – Date, Duration или Cycle. При выборе даты, в срок действия таймера надо передать дату, например, через переменную процесса. Для этого переменная должна быть ранее определена и должна быть заполнена значением (Рисунок 53).

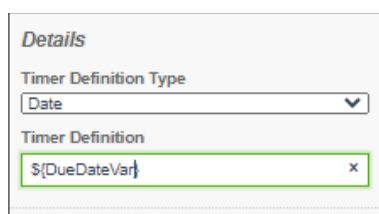


Рисунок 53 – Значение переменной процесса

4.5.3.5.3 Событие завершения

Событие предназначено для обозначения завершения бизнес-процесса. Для

добавления события завершения, нужно перетащить элемент «Create End Event» (○) в область редактирования процесса и соединить его с ранее добавленными элементами процесса. У события завершения можно заполнить значения атрибутов «Id» и «Name» в закладке «General» (Рисунок 54). Остальные атрибуты заполнять не нужно.

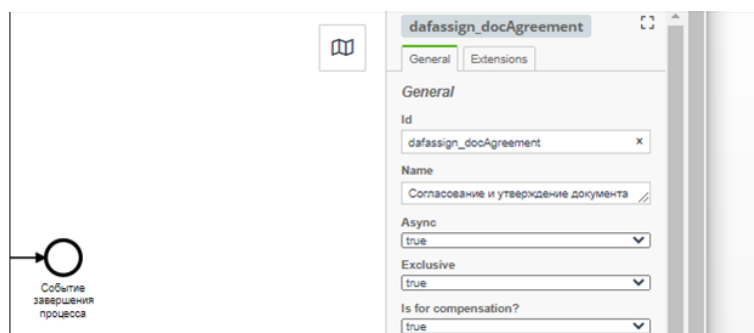


Рисунок 54 – Закладка «General»

4.5.3.5.4 Шлюз ◊

Шлюз предназначен для добавления параллельных, инклюзивных и эксклюзивных развилок в процессы, а также шлюзов на основе событий и сложных шлюзов.

Для добавления шлюза, нужно перетащить элемент «Create Gateway» (◊) в область редактирования процесса и соединить его с ранее добавленными элементами процесса.

Для выбора типа шлюза нужно нажать левой кнопкой мыши на элемент диаграммы, затем, на кнопку «Change Type». Для выбора доступны следующие виды шлюзов:

- Параллельный «Parallel Gateway» – используется для создания параллельных потоков задач без условий для сходящихся потоков и синхронизации параллельных веток при выполнении процесса;
- Инклюзивный «Inclusive Gateway» – используется для создания альтернативных потоков задач или сходящихся задач, при необходимости объединения нескольких потоков в один. При использовании шлюза для создания альтернативных потоков задач на схеме процесса необходимо прописать условие, по которому будут осуществляться переходы по различным веткам шлюза (Рисунок 55);

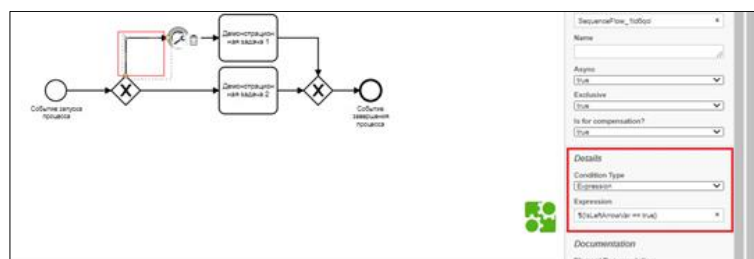


Рисунок 55 – Описание условия для перехода

- Сложный «Complex Gateway» – используется для более сложных условий перехода, которые могут объединять различные факторы анализа. В результате переход должен быть осуществлен хотя бы по одной ветке, исходящей из шлюза;

– Основанный на событии «Event based Gateway» – используется для перехода по веткам процесса, в зависимости от полученных сообщений или сигналов.

4.5.3.5.5 Задача

Элемент «Задача» предназначен для добавления пользовательских, сервисных и скриптовых задач, запуска подпроцессов.

Для добавления задачи нужно перетащить элемент «Create Task» () в область редактирования процесса и соединить его с ранее добавленными элементами процесса.

Для выбора типа задачи нажмем левой кнопкой мыши на элемент диаграммы, затем, на значок «Change Type». Для выбора доступны следующие виды задач:

- «Send Task» – Задача отправки сообщения;
- «Receive Task» – Задача получения сообщения;
- «User Task» – Пользовательская задача;
- «Manual Task» – Ручная задача;
- «Business Rule Task» – Задача бизнес-правил;
- «Service Task» – Сервисная задача;
- «Script Task» – Скриптовая задача;
- «Call Activity» – Запуск активности;
- «Sub Process (expanded)» – Подпроцесс развернутый.

4.5.3.5.5.1 Пользовательская задача

Пользовательская задача используется, когда исполнителем задачи является человек или группа людей.

Основными атрибутами задачи являются:

- «Id» – уникальный идентификатор задачи, в рамках спецификации процесса;
- «Name» – наименование задачи, должно быть понятно конечным пользователям;
- «Assignee» – исполнитель задачи, если он один (передаем логин пользователя, правильнее через переменную процесса);
- «Candidate Users» – кандидаты на выполнение задачи, список логинов кандидатов через запятую;
- «Candidate Groups» – группы, в которых должны быть кандидаты на выполнение задачи, список через запятую;
- «Form Key» – ключ формы, используется для связи задачи процесса и формы пользовательского интерфейса, предназначенной для выполнения этой задачи;
- «Due Date» – срок завершения задачи, передается в виде штампа дата/время (ISO-

8601) или продолжительности выполнения задачи (например, PT4H, 4 часа, ISO-8601);

– «Skip Expression» – регулярное выражение, при котором задача может быть пропущена.

При использовании пользовательских задач возникает необходимость подключения дополнительных функций, например, расчета исполнителей по группе или вычисления срока выполнения задачи с учетом рабочего календаря. Для этого в задачу добавляется слушатель («Listener»).

Для добавления слушателя откроем вкладку «Listeners» и нажмем кнопку «Добавить», напротив поля «Task Listener» (Рисунок 56).

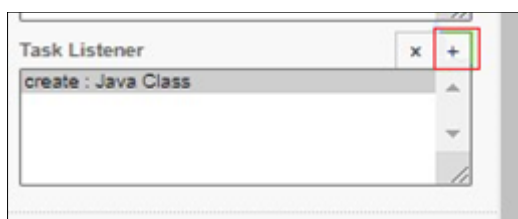


Рисунок 56 – Кнопка «Добавить»

Заполним основные атрибуты слушателя, в том числе:

- в поле «Event Type» – create;
- в поле «Listener Type» – Java Class;
- в поле «Java Class» – org.activiti.engine.impl.bpmn.listener.ScriptTaskListener;
- в поле «Fields» – скрипт, который должен запускаться в задаче;
- в поле «Name» – script;
- в поле «Type» – string
- в поле «Value» – evaluate(ru.reinform.activiti.utils.ScriptUtils.calc(task,'Err,DueDate', "calcDueDate(task);")) (Рисунок 57).

В данном примере будет запущен расчет срока исполнения задачи, с учетом рабочего календаря.

The image shows a configuration window for a 'Task Listener'. It is divided into several sections:

- Task Listener:** A list containing one item: 'create : Java Class'.
- Task Listener:** Configuration fields:
 - Event Type: create
 - Listener Type: Java Class
 - Java Class: org.activiti.engine.impl.bpmn.listener.Script
- Fields:** A list containing one item: '0 : script'.
- Fields:** Configuration for the selected field:
 - Name: script
 - Type: String
 - Value: evaluate(ru.reinform.activiti.utils.ScriptUtils.calcDueDate(task, "Err, DueDate", "calcDueDate(task);"))

Рисунок 57 – Поле «Value»

Часто, бывает необходимость создать [1..n] экземпляров задачи, последовательно или параллельно. Для этого, нажмем левой кнопкой мыши на задачу, нажмем значок «Change Type», затем, выберем тип множественной задачи – параллельный (☐) или последовательный (☐) (Рисунок 58).

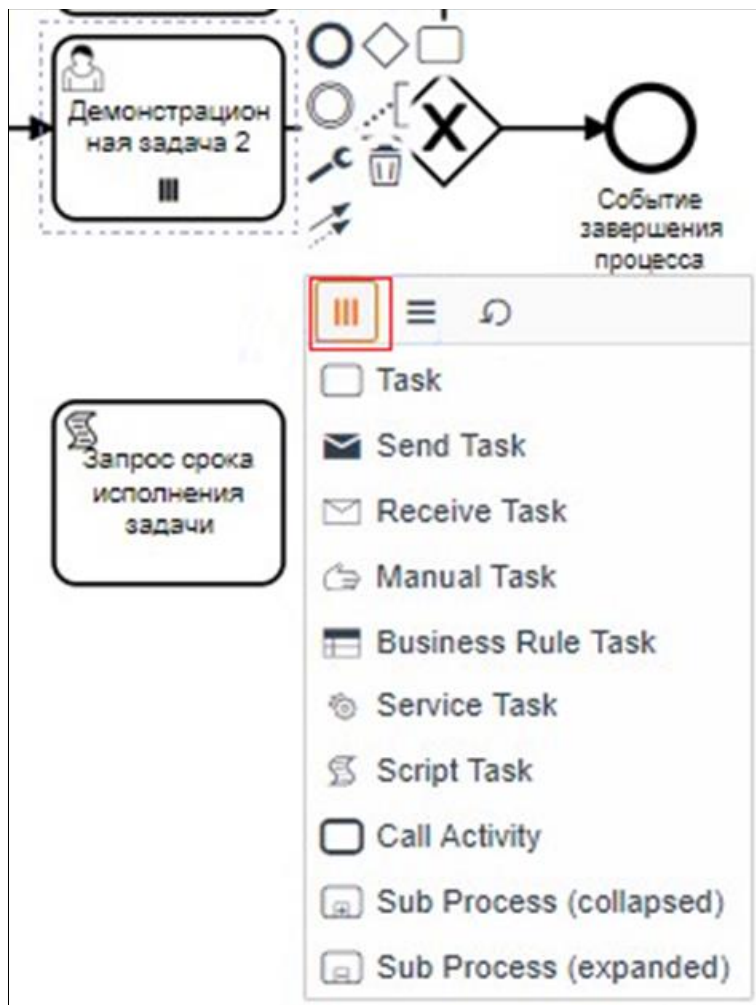


Рисунок 58 – Тип множественной задачи – параллельный

У атрибутов задачи появится секция «Multi Instance». Допустим, что в переменной процесса «CollectionVar» хранится список пользователей, на которых необходимо назначить задачу. Заполним следующие атрибуты этой секции (Рисунок 59):

- «Collection» – `${CollectionVar}`;
- «Element Variable» – AssigneeVar.

Рисунок 59 – Секция «Multi Instance»

В атрибут задачи «Assignee» добавим переменную, из которой будет подставляться значение для каждого экземпляра задачи (Рисунок 60).

Рисунок 60 – Атрибут задачи «Assignee»

4.5.3.5.5.2 Скриптовая задача

Скриптовая задача используется, когда в бизнес-процессе должен быть выполнен какой-либо скрипт.

Основные атрибуты задачи включают (Рисунок 61):

- «Id» – уникальный идентификатор задачи, в рамках спецификации процесса;
- «Name» – наименование задачи, должно быть понятно конечным пользователям;
- «Script Language» – язык скрипта, для выбора доступны javascript и groovy. При разработке используем «groovy»;
- «Script» – в поле добавляем вызов скрипта, например: `evaluate(ru.reinform.activiti.utils.ScriptUtils.calc(execution,'Err,Doc',"getTaskDueDate(execution);"))`.

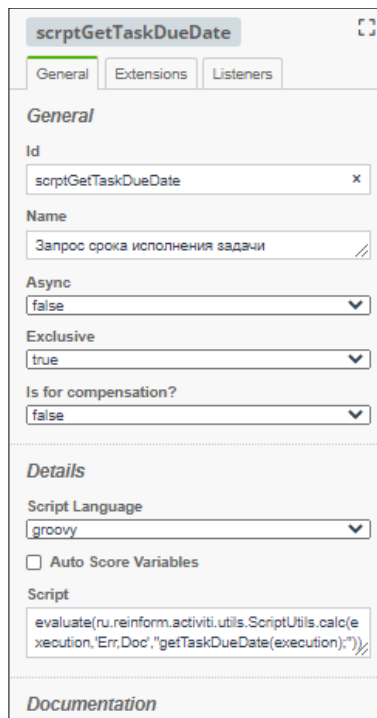


Рисунок 61 – Атрибуты скриптовой задачи

4.5.3.5.5.3 Запуск активности

Запуск активности используется, при необходимости запуска подпроцессов, например, множественных процессов согласования. Для запуска основных процессов используется функционал скриптовых задач.

Добавим активность на диаграмму процесса, заполним значение атрибута «Name» (Рисунок 62).



Рисунок 62 – Атрибут «Name»

Основные атрибуты для заполнения находятся на вкладке «Main Config», с том числе (Рисунок 63):

- «Called Element» – вводим ключ подпроцесса, который должен вызываться;
- «In Parameter» – список входных параметров, которые необходимо передать в подпроцесс, включает следующие:
 - «Source» – параметр источник из основного процесса;
 - «Source expression» – выражение, рассчитанное в основном процессе;
 - «Target» – параметр подпроцесса, в который передается значение из основного процесса;
 - «Target Expression» – выражение подпроцесса, в который передается значение из основного процесса.

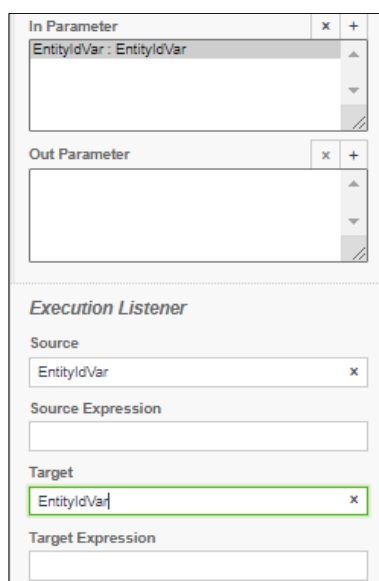


Рисунок 63 – Атрибуты вкладки «Main Config»

4.5.3.5.6 Развернутый подпроцесс

Развернутый подпроцесс предназначен для добавления подпроцессов в спецификацию бизнес-процесса. Этот элемент используется редко, ввиду сложности восприятия итоговых диаграмм спецификаций процессов;

Для добавления развернутого подпроцесса необходимо перетащить элемент «expanded SubProcess» в область редактирования диаграммы. Работа с этим элементом аналогична работе с обычным процессом.

4.5.3.5.7 Ссылка на объект данных

Ссылка на объект данных предназначена для добавления ссылок на объекты данных, используемые в бизнес-процессе. Используя данный элемент, можно отметить объекты, с которыми осуществляется работа в разных частях процесса (Рисунок 64).

Атрибут «Name» обязателен для заполнения.

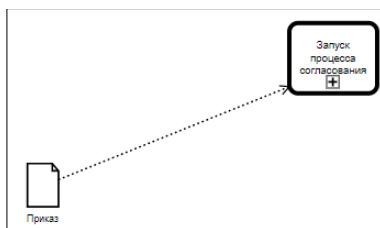


Рисунок 64 – Пример ссылки на объект данных

4.5.3.5.8 Ссылка на хранилище данных

Ссылка на хранилище данных предназначена для визуализации ссылок на хранилище сохраненной информации. Использование этого элемента носит визуальный характер. С его помощью можно указать с каким хранилищем работаем в текущем процессе.

4.5.3.5.9 Пул/Ответственный

Элемент Пул/Ответственный предназначен для визуализации разграничения задач по типам исполнителей, например, системных и пользовательских.

4.5.3.5.10 Группа

Элемент Группа предназначен для группировки графических элементов, принадлежащих к одной категории, для повышения простоты восприятия спецификации бизнес-процесса.

4.5.3.5.11 Пример реализации процесса «Работа по заявке на установку компьютера»

Диаграмма процесса «Работа по заявке на установку компьютера» представлена ниже (Рисунок 65).

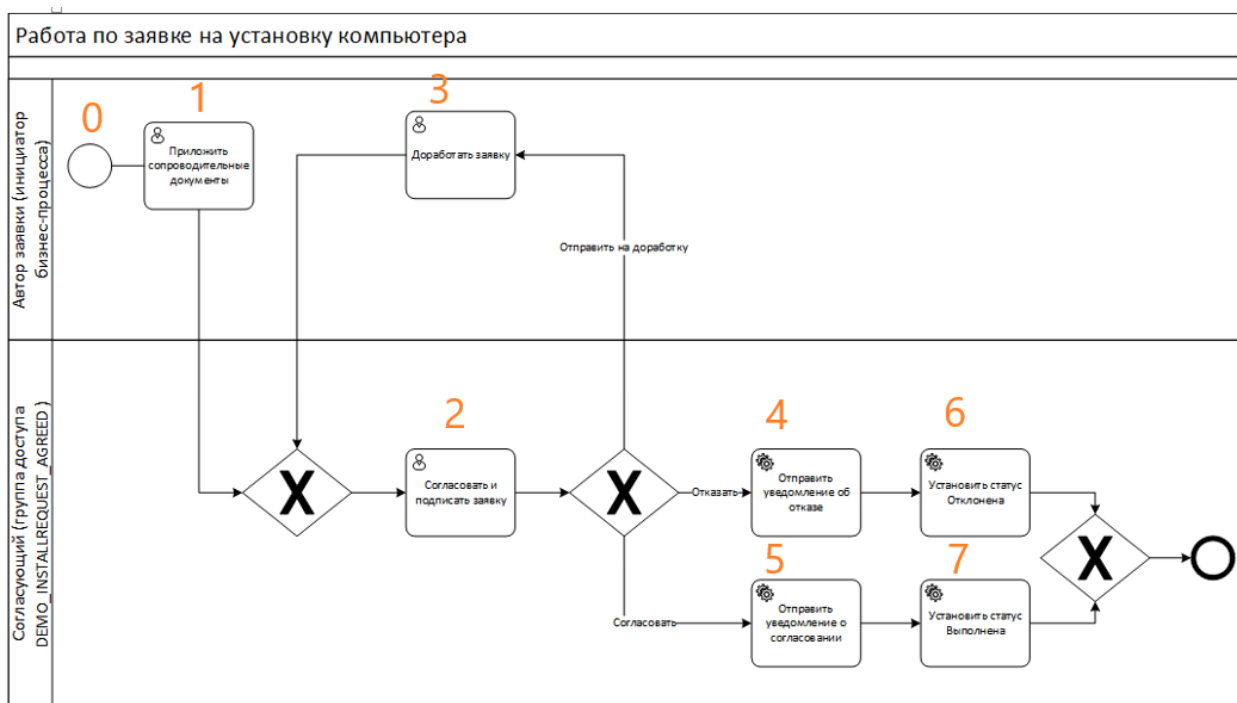


Рисунок 65 – Диаграмма процесса «Работа по заявке на установку компьютера»

4.5.3.5.11.1 Событие запуска бизнес-процесса

4.5.3.5.11.2 Заполнение атрибутов пользовательской задачи

В событие запуска бизнес-процесса необходимо добавить переменную: EntityIdVar/Идентификатор заявки/String.

Для добавления переменной откроем вкладку «Расширения».

Заполним следующие атрибуты переменной «EntityIdVar»:

- Идентификатор: «EntityIdVar»;
- Имя: «Идентификатор заявки»;
- Переменная: «EntityIdVar»;
- Обязательный: «true».

4.5.3.5.11.3 Пользовательская задача «Приложить сопроводительные документы»

Для добавления пользовательской задачи в настройках задачи выберем тип: «Задача, выполняемая пользователем».

Заполним следующие атрибуты задачи (Рисунок 66):

- Идентификатор: «AttachCorrespondingDocs»;
- Имя: «Приложить сопроводительные документы»;
- Ключ формы: «AttachCorrespondingDocs»;
- Максимальный срок: «P3D».

The screenshot shows the configuration window for a task named "AttachCorrespondingDocs". The window has three tabs: "Основные" (Basic), "Свойства Формы" (Form Properties), and "Слушатели" (Listeners). The "Основные" tab is active. Under the "Основные" section, the following fields are visible: "Идентификатор" (Identifier) with the value "AttachCorrespondingDocs"; "Имя" (Name) with the value "Приложить сопроводительные документы"; "Асинхронный" (Asynchronous) with the value "false"; "Исключительный" (Exclusive) with the value "true"; and "Компенсирующий?" (Compensating?) with the value "false". Under the "Детали" (Details) section, the following fields are visible: "Исполнитель" (Executor) is empty; "Кандидаты В Пользователи" (Candidates in Users) is empty; "Группы Кандидатов" (Candidate Groups) is empty; "Ключ Формы" (Form Key) with the value "AttachCorrespondingDocs"; and "Максимальный Срок" (Maximum Duration) with the value "P3D".

Рисунок 66 – Атрибуты задачи «Приложить сопроводительные документы»

4.5.3.5.11.4 Вызов типовых функций расчета планового срока выполнения задачи и назначения исполнителя задачи

Для добавления расчета плановой даты завершения задачи и кандидатов на выполнение задачи, откроем вкладку «Слушатели» и добавим «Слушателя задачи».

Заполним следующие атрибуты слушателя:

- Тип события: «create»;
- Тип слушателя: «Java Класс»;
- Java Класс: «org.activiti.engine.impl.bpmn.listener.ScriptTaskListener»;
- поле «script», значения:
 - Имя: «script»;
 - Тип: «Выражение»;
 - Значение: «evaluate(ru.reinform.activiti.utils.ScriptUtils.calc(task,'Err,Doc,DueDate','setTaskAssignee(task);setDueDate(task);'))»;
- поле «language», значения:
 - Имя: «language»;
 - Тип: «Строка»;
 - Значение: «groovy».

4.5.3.5.11.5 Пользовательская задача «Согласовать и подписать заявку»

Для добавления пользовательской задачи в настройках задачи выберем тип: «Задача, выполняемая пользователем».

Заполним следующие атрибуты пользовательской задачи (Рисунок 67):

- Идентификатор: «ApproveAndSignRequest»;
- Имя: «Согласовать и подписать заявку»;
- Группа кандидатов: «DEMO_INSTALL_AGREED»;
- Ключ формы: «ApproveAndSignRequest»;
- Максимальный срок: «P3D».

Рисунок 67 – Атрибуты задачи «Согласовать и подписать заявку»

4.5.3.5.11.6 Добавление свойства пользовательской задачи.

Для добавления свойства пользовательской задачи, откроем вкладку «Свойства формы», затем, ждем кнопку добавления нового свойства.

Заполняем следующие атрибуты нового свойства:

- Id: «ApprovalResult»;
- Name: «Результат согласования»;
- Type: «enum»;
- Variable: «ApprovalResult»;
- Обязательный: «true»;
- Значения форм: добавляем допустимые значения для каждого результата завершения задачи, в том числе:
 - Идентификатор: «accept», Имя: «Принять»;
 - Идентификатор: «reject», Имя: «Отклонить»;
 - Идентификатор: «remake», Имя: «Доработать».

4.5.3.5.11.7 Вызов типовых функций расчета планового срока и кандидатов на выполнение задачи

Для добавления расчета плановой даты завершения задачи и кандидатов на выполнение задачи, откроем вкладку «Слушатели» и добавим «Слушателя задачи».

Заполняем следующие атрибуты слушателя:

- Тип события: «create»;
- Тип слушателя: «Java Класс»;
- Java Класс: «org.activiti.engine.impl.bpmn.listener.ScriptTaskListener»;
- поле «script», значения:
 - Имя: «script»;
 - Тип: «Выражение»;
 - Значение: «evaluate(ru.reinform.activiti.utils.ScriptUtils.calc(task,'Err,Doc,DueDate', 'setTaskAssignee(task);setDueDate(task);'))»;
- поле «language», значения:
 - Имя: «language»;
 - Тип: «Строка»;
 - Значение: «groovy».

4.5.3.5.11.8 Пользовательская задача «Доработать заявку»

Для добавления пользовательской задачи в настройках задачи выберем тип: «Задача, выполняемая пользователем».

Заполним следующие атрибуты пользовательской задачи:

- Идентификатор: «ReworkRequest»;
- Имя: «Доработать заявку»;
- Ключ формы: «ReworkRequest»;
- Максимальный срок: «P3D».

4.5.3.5.11.9 Вызов типовой функции расчета планового срока выполнения задачи

Для добавления расчета плановой даты завершения задачи, откроем вкладку «Слушатели» и добавим «Слушателя задачи».

Заполняем следующие атрибуты слушателя:

- Тип события: «create»;
- Тип слушателя: «Java Класс»;
- Java Класс: «org.activiti.engine.impl.bpmn.listener.ScriptTaskListener»;
- поле «script», значения:
 - Имя: «script»;
 - Тип: «Выражение»;
 - Значение: «evaluate(ru.reinform.activiti.utils.ScriptUtils.calc(task,'Err,Doc,DueDate', 'setTaskAssignee(task);setDueDate(task);'))»;

- поле «language», значения:
 - Имя: «language»;
 - Тип: «Строка»;
 - Значение: «groovy».

4.5.3.5.11.10 Задача-сценарий «Отправить уведомление об отказе»

Для добавления задачи-сценария в настройках задачи выберем тип: «Задача - сценарий».

Заполним следующие атрибуты задачи:

- Идентификатор: «sendRejectNotification»;
- Имя: «Отправить уведомление об отказе»;
- Язык скрипта: «groovy»;
- Скрипт: «evaluate(ru.reinform.activiti.utils.ScriptUtils.calc(execution,'Err,Mail',»notify Reject(execution);»))».

4.5.3.5.11.11 Задача-сценарий «Отправить уведомление о согласовании»

Для добавления задачи-сценария в настройках задачи выберем тип: «Задача - сценарий».

Заполним следующие атрибуты задачи:

- Идентификатор: «sendApproveNotification»;
- Имя: «Отправить уведомление о согласовании»;
- Язык скрипта: «groovy»;
- Скрипт: «evaluate(ru.reinform.activiti.utils.ScriptUtils.calc(execution,'Err,Mail',»notify Approve(execution);»))».

4.5.3.5.11.12 Задача-сценарий «Установить статус Отклонена»

Для добавления задачи-сценария в настройках задачи выберем тип: «Задача - сценарий».

Заполним следующие атрибуты задачи:

- Идентификатор: setStatusReject;
- Имя: Установить статус Отклонена;
- Язык скрипта: groovy;
- Скрипт:

```
def status = 'reject'
evaluate(ru.reinform.activiti.utils.ScriptUtils.calc(execution,'Err,Status,Mdm',"setStatus(execution,'${status}');"))
```

4.5.3.5.11.13 Задача-сценарий «Установить статус Выполнена»

Для добавления задачи-сценария в настройках задачи выберем тип: «Задача - сценарий».

Заполним следующие атрибуты задачи:

- Идентификатор: "setStatusDone";
- Имя: "Установить статус Выполнена";
- Язык скрипта: "groovy";
- Скрипт:

```
"def status = 'done'  
evaluate(ru.reinform.activiti.utils.ScriptUtils.calc(execution, 'Err, Status, Mdm', "setStatus(execution, '${status}');"))"
```

4.5.3.5.11.14 Условие перехода на доработку

Для добавления условия перехода на доработку выберем переход с атрибутами:

- Тип условия: «Выражение»;
- Выражение: « $\{\text{ApprovalResult} == \text{'remake'}\}$ ».

4.5.3.5.11.15 Условие перехода на отказ

Для добавления условия перехода на отказ выберем переход с атрибутами:

- Тип условия: «Выражение»;
- Выражение: « $\{\text{ApprovalResult} == \text{'reject'}\}$ ».

4.5.3.5.11.16 Условие перехода на согласовано

Для добавления условия перехода на согласовано выберем переход с атрибутами:

- Тип условия: «Выражение»;
- Выражение: « $\{\text{ApprovalResult} == \text{'accept'}\}$ ».

4.5.3.5.11.17 Публикация бизнес-процесса

Для публикации бизнес-процесса нажмем кнопку «Действия», а в выпадающем списке выберем действие «Обновить в системе» (Рисунок 68).

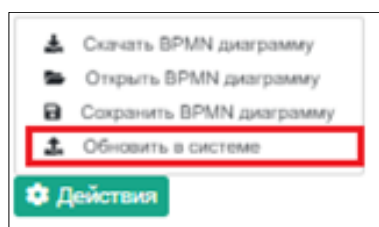


Рисунок 68 – Действие «Обновить в системе»

В открывшемся окне заполняем логин и пароль от GIT, комментарий к коммиту и нажимаем кнопку «Обновить» (Рисунок 69).

Рисунок 69 – Кнопка «Обновить»

Бизнес процесс будет сохранен в репозитории процессов GIT и обновлен в Системе.

4.5.4 Руководство пользователя по работе с реестром прикладных сервисов

4.5.4.1 Описание данных реестра прикладных сервисов

Реестр прикладных сервисов предназначен для хранения библиотек сервисных функций, используемых в бизнес-процессах.

Для работы реестра в системе необходимы следующие справочники:

- BPMFnLibsStorages (Реестр хранилищ библиотек сервисных функций);
- BPMComplexProcesses (Реестр комплексных бизнес-процессов);
- SubSystems (Подсистемы);
- Systems (Системы).

Справочники BPMFnLibsStorages (Таблица 21) и BPMComplexProcesses (Таблица 22) являются общими и обновляются разработчиками прикладных приложений.

Таблица 21 – Описание справочника BPMFnLibsStorages

Поле	Описание	Примечание
code*	Код	
name*	Наименование	
subsystem*	Подсистема	Связь с элементом справочника SubSystems
system*	Система	Связь с элементом справочника Systems
complexProcesses	Комплексный ВР	Связь с элементом справочника BPMComplexProcesses
description	Описание	
url*	Путь к репозиторию в GIT	

Поле	Описание	Примечание
filename*	Путь к файлу	

Таблица 22 – Описание справочника BPMComplexProcesses

Поле	Описание	Примечание
code*	Код	
name*	Наименование	
subsystem*	Подсистема	Связь с элементом справочника SubSystems
system*	Система	Связь с элементом справочника Systems
description	Описание	

4.5.4.2 Описание витрины «Реестр конфигураций прикладных сервисов, используемых в бизнес-процессах»

Для доступа к витрине «Реестр конфигураций прикладных сервисов, используемых в бизнес-процессах» в режиме просмотра предназначена группа доступа «Реестр бизнес-процессов - Просмотр SYS_BPMN_VIEW», в режиме редактирования – «Разработчик приложений SYS_APP_DEVELOPER».

Витрина «Реестр конфигураций прикладных сервисов, используемых в бизнес-процессах, наполняется системными хранилищами библиотек, а также элементами справочника BPMFnLibsStorages (Рисунок 70).

Код	Наименование	Система	Подсистема	Комплексный процесс
cdr_libs	Системное хранилище библиотек: Учет данных по аэрофотосъемке	Учет данных по аэрофотосъемке		
cdr_libs	Системное хранилище библиотек: Алгоритмическая проверка градостроительных решений	Алгоритмическая проверка градостроительных решений		
cdr_libs	Системное хранилище библиотек: Троицк и Щербинка	Троицк и Щербинка		
cdr_libs	Системное хранилище библиотек: Поручения ДС	Поручения ДС		
cdr_libs	Системное хранилище библиотек: Мой район	Мой район		
cdr_libs	Системное хранилище библиотек: Платформа	Платформа	Общие компоненты	
cdr_libs	Системное хранилище библиотек: ОАСИ	ОАСИ		ОАСИ_COMMON
cdr_libs	Системное хранилище библиотек: Расширения платформы	Расширения платформы		
cdr_libs	Системное хранилище библиотек: DEMO приложение	DEMO приложение		DEMO_COMMON
dafassign_libs	Библиотеки Поручений	Поручения ДС		ДИТ. Поручения




Рисунок 70 – Витрина «Реестр конфигураций прикладных сервисов, используемых в бизнес-процессах»

Панель инструментов витрины «Реестр конфигураций прикладных сервисов, используемых в бизнес-процессах» имеет табличный вид и содержит следующие элементы:

- контекстный поиск по содержимому витрины;
- кнопка настройки отображения / скрытия колонок таблицы;
- кнопка расширенного поиска по содержимому витрины;
- кнопка вызова окна создания нового элемента реестра.

Витрина «Реестр конфигураций прикладных сервисов, используемых в бизнес-процессах» содержит функциональные иконки (Таблица 23).

Таблица 23 – Иконки витрины «Реестр конфигураций прикладных сервисов, используемых в бизнес-процессах»

Иконка	Наименование	Примечание
	Редактировать свойства	Переход к окну редактирования свойств прикладного сервиса (Раздел 4.5.4.3) Иконка не отображается для системных хранилищ библиотек
	Редактировать прикладные сервисы	Переход к окну редактирования прикладного сервиса (Раздел 4.5.4.4) Иконка не отображается для системных хранилищ библиотек
	Просмотреть прикладные сервисы	Переход к окну просмотра прикладного сервиса (Раздел 4.5.4.5) Иконка не отображается для системных хранилищ библиотек

4.5.4.3 Окно создания и редактирования свойств прикладного сервиса

После нажатия кнопки «Создать» открывается модальное окно для добавления нового элемента реестра (Рисунок 71).

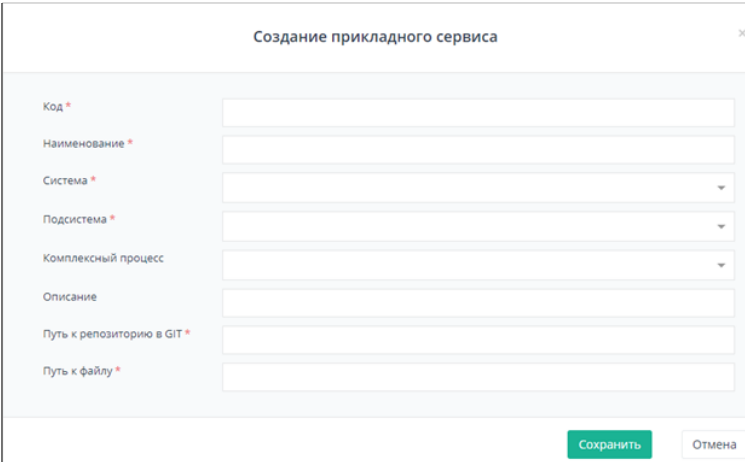


Рисунок 71 – Модальное окно для добавления нового элемента реестра
Описание полей модального окна приведено ниже (Таблица 24).

Таблица 24 – Описание полей модального окна

Иконка	Наименование
Код*	Уникальный код для создаваемой модели информационного объекта Рекомендовано формирование по шаблону Код системыКод хранилища_Libs
Наименование*	–
Система*	Выбор из значений справочника Systems
Подсистема*	Выбор из значений справочника SubSystems
Комплексный процесс	–
Описание	–
Путь к репозиторию в GIT*	Должен содержать путь к папке с приложением без учета домена
Путь к файлу*	Должен содержать путь к файлу внутри папки с приложением

Для добавления нового элемента реестра информационных объектов можно использовать функционал работы со справочником BPMFnLibsStorages.

4.5.4.4 Окно редактирования прикладного сервиса

Работа с прикладным сервисом в режиме редактирования аналогична работе с бизнес-процессом в режиме редактирования. Окно редактирования прикладного сервиса приведено ниже (Рисунок 72).

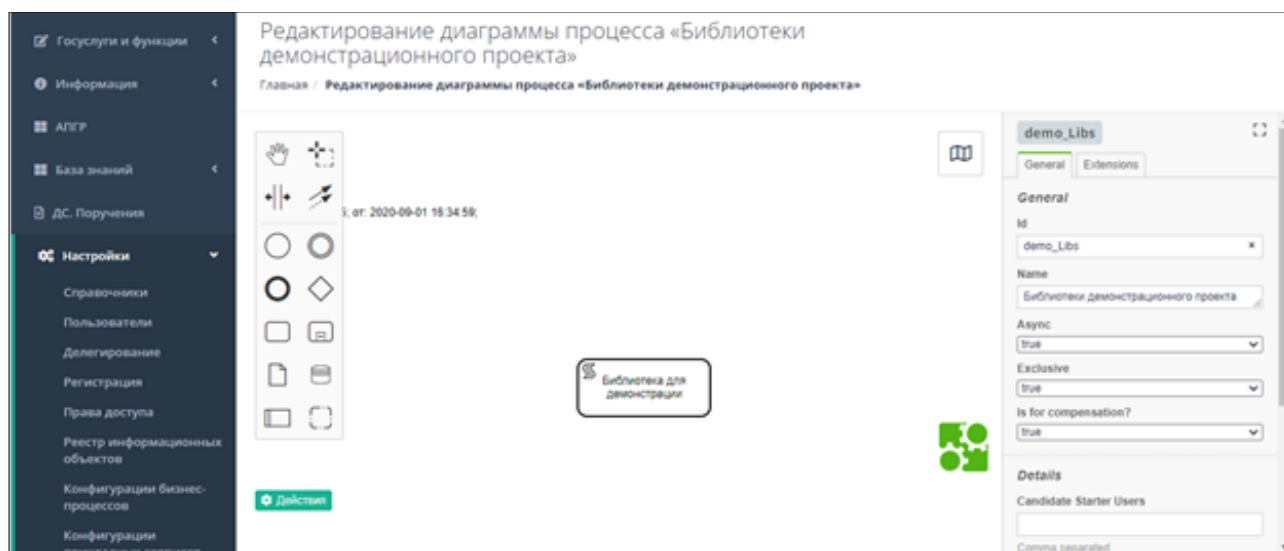


Рисунок 72 – Окно редактирования прикладного сервиса

4.5.4.5 Окно просмотра прикладного сервиса

Работа с прикладным сервисом в режиме просмотра аналогична работе с бизнес-процессом в режиме просмотра. Окно просмотра прикладного сервиса приведено ниже (Рисунок 73).

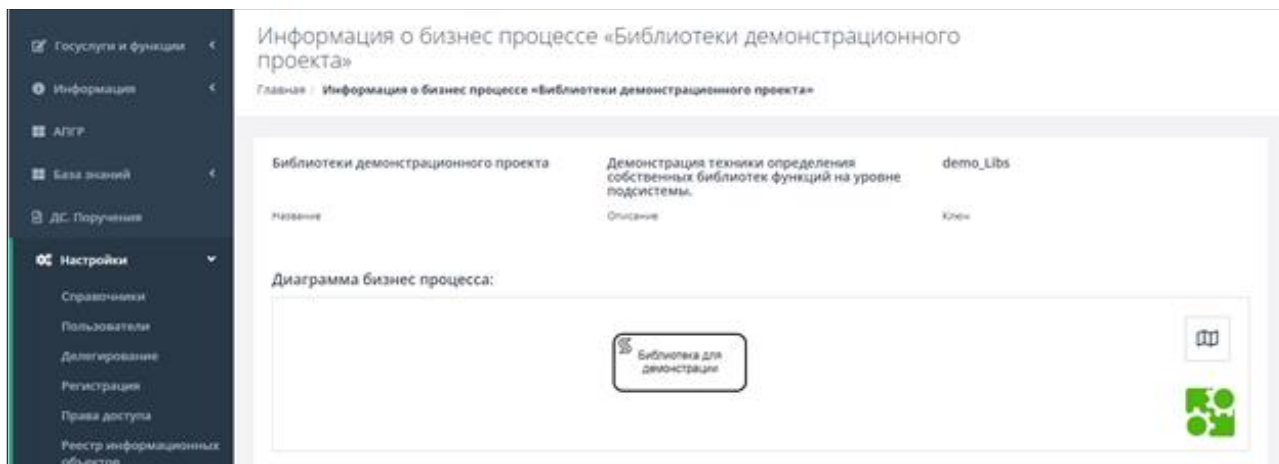


Рисунок 73 – Окно просмотра прикладного сервиса

4.5.4.6 Описание витрины со списком библиотек прикладных сервисов, используемых в бизнес-процессах

После нажатия на строку таблицы в витрине «Реестр конфигураций прикладных сервисов, используемых в бизнес-процессах» осуществляется переход к витрине со списком библиотек прикладных сервисов, используемых в бизнес-процессах (Рисунок 74).

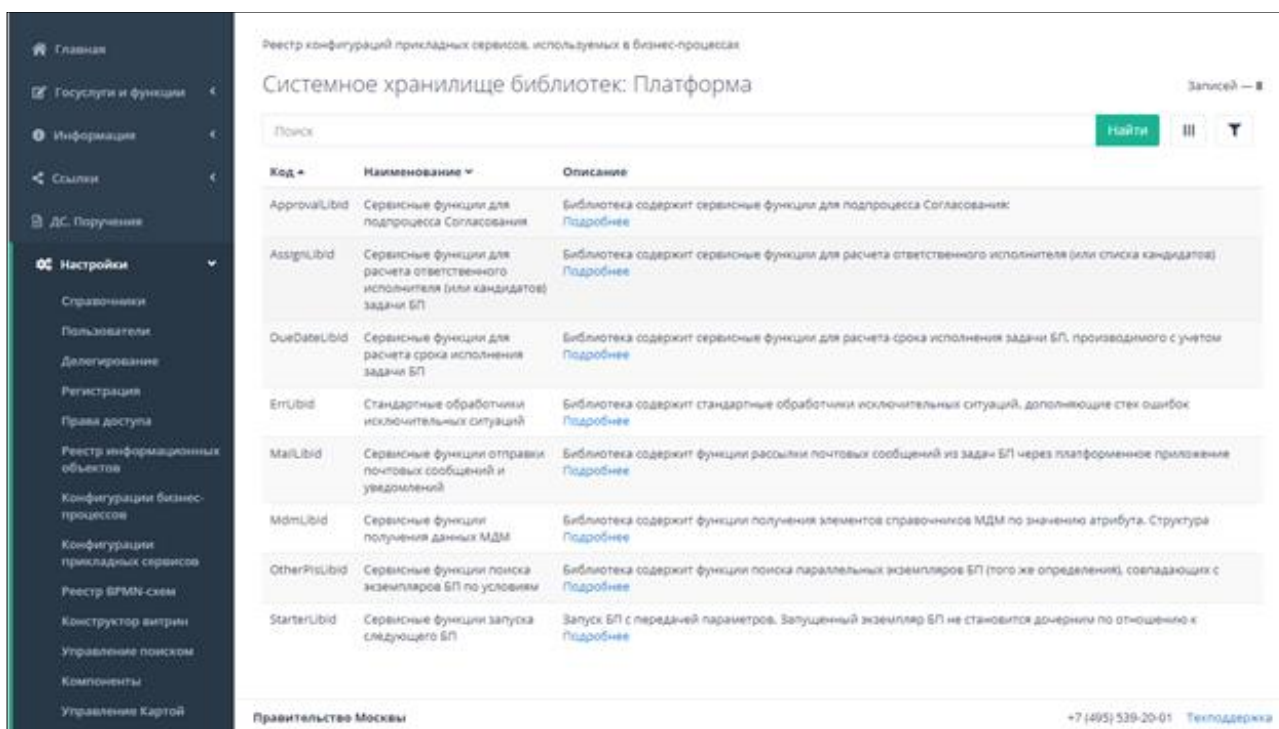


Рисунок 74 – Витрина со списком библиотек прикладных сервисов, используемых в бизнес-процессах

Витрина со списком библиотек прикладных сервисов, используемых в бизнес-процессах, содержит следующие элементы:

- контекстный поиск по содержанию витрины;
- кнопка настройки отображения / скрытия колонок таблицы;
- кнопка расширенного поиска по содержанию витрины.

4.5.4.7 Описание карточки просмотра библиотеки прикладных сервисов, используемых в бизнес-процессах

После нажатия на строку таблицы в витрине со списком библиотек прикладных сервисов, используемых в бизнес-процессах, осуществляется переход к карточке просмотра библиотеки прикладных сервисов, используемых в бизнес-процессах (Рисунок 75).

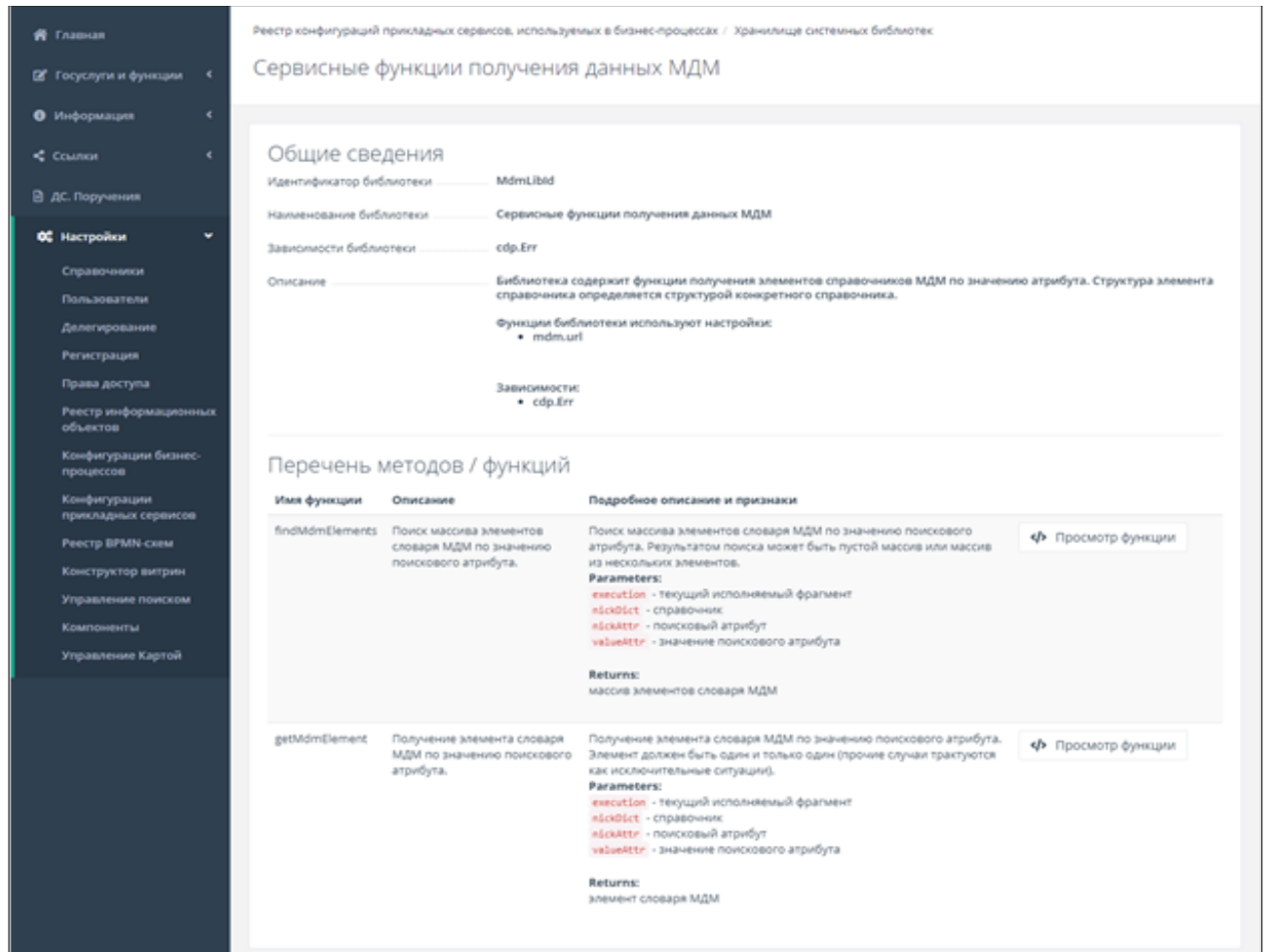


Рисунок 75 – Пример карточки просмотра библиотеки прикладных сервисов

После нажатия кнопки «Просмотр функции» открывается модальное окно с параметрами методов / функций прикладного сервиса бизнес-процесса (Рисунок 76).

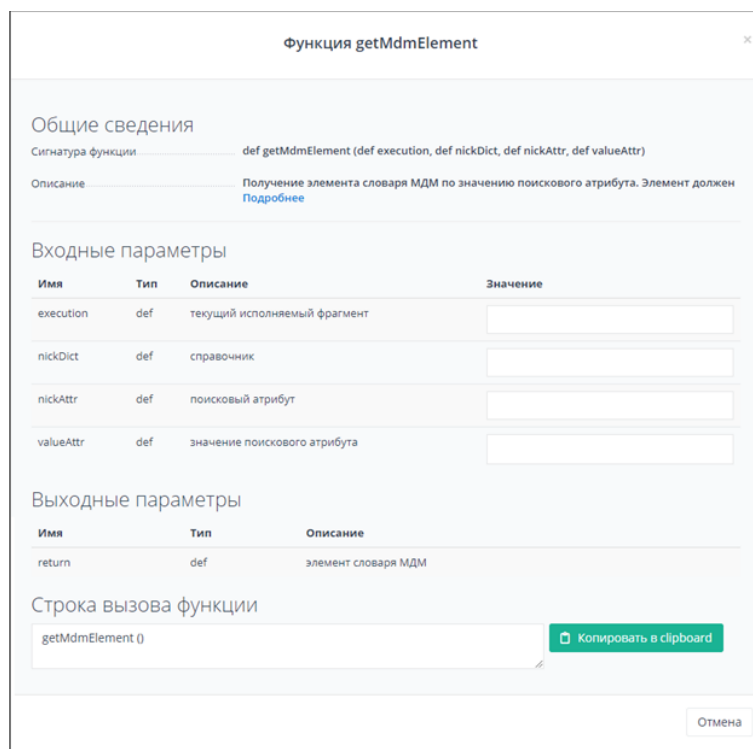


Рисунок 76 – Модальное окно с параметрами методов / функций прикладного сервиса

Значение входных параметров необходимо указывать в одинарных кавычках (').

Строка вызова функции содержит конкатенацию имени функции и указанных значений входных параметров, а также кнопку для копирования содержимого строки в буфер обмена (Рисунок 77).

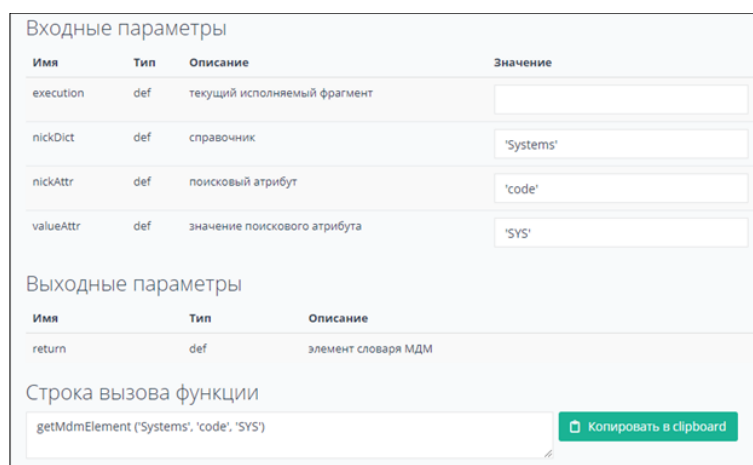


Рисунок 77 – Модальное окно. Строка вызова функции

4.6 Создание задач на Конструкторе форм

4.6.1 Общая информация

Документация по стандартным возможностям создания форм доступна по ссылке: <https://help.form.io/userguide/forms/>.

4.6.2 Требования к XSD

4.6.2.1 Именованые неймспейсов

Именованые неймспейсов должно выполняться с учетом следующего (Рисунок 78):

'http://' + НАИМЕНОВАНИЕ_КОМПАНИИ_РАЗРАБОТЧИКА.toLowerCase() + '/' + КОД_СИСТЕМЫ.toLowerCase() + '/' + КРАТКИЙ_КОД_ПОДСИСТЕМЫ.toLowerCase()+ '/model/' + КРАТКИЙ_КОД_ТИПА_ДОКУМЕНТА.toLowerCase() (или КОД_СЕМЕЙСТВА_ОБЩИХ_ТИПОВ.toLowerCase() для библиотек типов, разделяемых в пределах подсистемы).
 Пример: http://reinform.ru/demo/reconstructor/model/purchase.

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <!-- edited with XMLSpy v2011 rel. 2 (http://www.altova.com) by TeaM DJiNN (TeaM DJiNN) -->
3 <xs:schema xmlns="http://reinform.ru/demo/reconstructor/model/purchase" xmlns:xs="http://www.w3.org/2001/XMLSchema" targetNamespace="
4 http://reinform.ru/demo/reconstructor/model/purchase" elementFormDefault="qualified" attributeFormDefault="unqualified">
5   <xs:element name="purchase">
6     <xs:annotation>
7       <xs:documentation>Закупка офисного оборудования</xs:documentation>
8     </xs:annotation>
9     <xs:complexType>
10      <xs:sequence>
11        <xs:element name="documentId" type="xs:string">
12          <xs:annotation>
13            <xs:documentation>Идентификатор заявки</xs:documentation>
14          </xs:annotation>
15          </xs:element>
16          <xs:element name="folderId" type="xs:string">
17            <xs:annotation>
18              <xs:documentation>Идентификатор папки</xs:documentation>
19            </xs:annotation>
20            </xs:element>
21            <xs:element name="status" type="nsiType" minOccurs="0">
22              <xs:annotation>
23                <xs:documentation>Статус заявки</xs:documentation>
24              </xs:annotation>
25              </xs:element>
26              <xs:element name="number" type="xs:long" minOccurs="0">
27                <xs:annotation>
28                  <xs:documentation>Номер заявки</xs:documentation>
29                </xs:annotation>
30                </xs:element>
31                <xs:element name="date" type="xs:date" minOccurs="0">
32                  <xs:annotation>

```

Рисунок 78 – Именованние неймспейсов

4.6.2.2 Использование функций import и include

В настоящий момент использование функций import и include для внешних схем не предполагается. Все используемые элементы должны быть скопированы в схему.

По этой же причине использование префиксов в пространстве имен не имеет смысла и не рекомендуется.

4.6.2.3 Именованние элементов

При именовании элементов используем стиль написания camelCase: несколько слов пишутся слитно без пробелов, при этом каждое слово внутри фразы, кроме первого, пишется с большой буквы (<https://textedit.tools/camelcase> – онлайн конвертер).

При именовании элементов запрещено использовать ключевые слова Java и ключевые слова JavaScript, а также не рекомендуется использование слов «object» и «type» (список может быть расширен).

Рекомендованные наименования простых типов приведены ниже (Таблица 25).

Таблица 25 – Рекомендованные наименования простых типов

Наименование	Описание
documentId	Идентификатор документа

Наименование	Описание
folderId	Идентификатор папки документа

Рекомендованные наименования комплексных типов приведены ниже (Таблица 26).

Таблица 26 – Рекомендованные наименования комплексных типов

Наименование	Описание
nsiType	Справочное значение
userType	Пользователь
responsibleType	Ответственный исполнитель
fileType	Описание файла
approveeType	Согласующий по листу согласования
approvalCycleType	Согласование по листу согласования
folderType	Проектная библиотека
versionType	Версионность

Для полей со связями рекомендуется формировать имя поля со ссылкой на связанный документ по правилу «краткий код типа документа» + «Id», если связь единичная или + "Ids", если связь множественная.

Например, «objectId» – для указания связи с объектом или «documentIds» – для связи с документами.

Необходимо избегать случайного использования кириллицы вместо латиницы в именах полей при работе со схемами. Для проверки на наличие кириллицы в именах полей перед публикацией схемы необходимо скопировать схему из вкладки «Text Altova XMLSpy» и проверить на любом онлайн-ресурсе (например: [Сервис для поиска кириллицы в латинице](#)), что из кириллицы в схеме только описания полей.

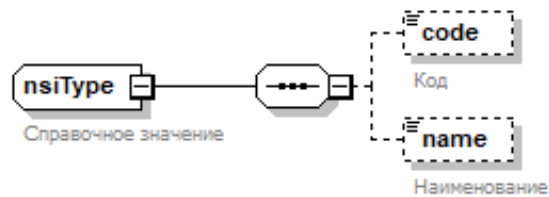
4.6.2.4 Требования к структуре комплексных типов

4.6.2.4.1 nsiType Справочное значение

Описание (Таблица 27) и структура (Рисунок 79) типа «Справочное значение» приведены ниже.

Таблица 27 – Описание типа «Справочное значение»

Наименование	Описание	Тип
code	Код	string
name	Наименование	string



Generated by XMLSpy

www.altova.com

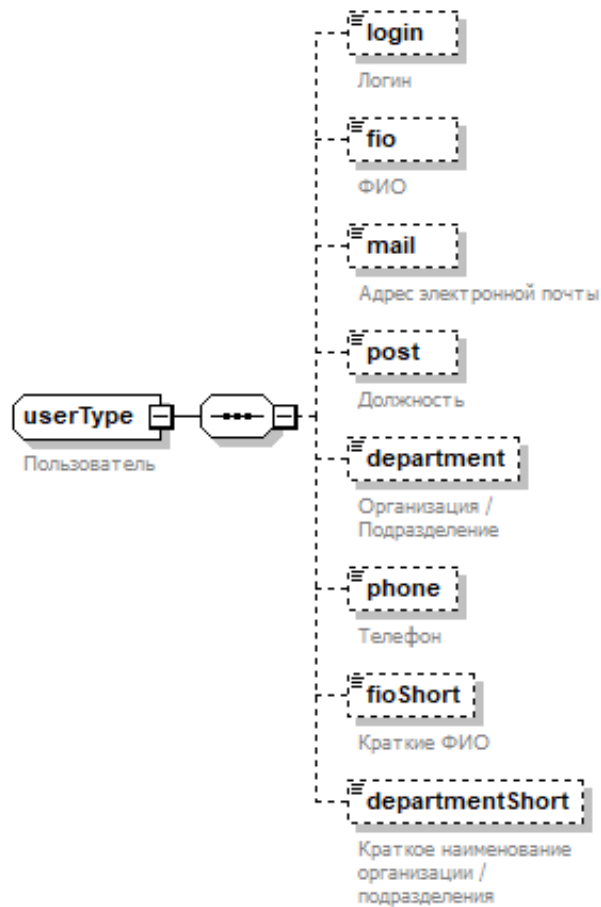
Рисунок 79 – Структура nsiType Справочное значение

4.6.2.4.2 userType Пользователь

Описание (Таблица 28) и структура (Рисунок 80) типа «Пользователь» приведены ниже.

Таблица 28 – Описание типа «Пользователь»

Наименование	Описание	Тип
login	Логин	string
fio	ФИО	string
mail	Адрес электронной почты	string
post	Должность	string
department	Организация / Подразделение	string
phone	Телефон	string
fioShort	Краткие ФИО	string
departmentShort	Краткое наименование организации / подразделения	string



Generated by XMLSpy

www.altova.com

Рисунок 80 – Структура userType Пользователь

4.6.2.4.3 responsibleType Ответственный исполнитель

Описание (Таблица 29) и структура (Рисунок 81) типа «Ответственный исполнитель» приведены ниже.

Таблица 29 – Описание типа «Ответственный исполнитель»

Наименование	Описание	Тип
responsibleRole	Роль ответственного исполнителя	nsiType
login	Логин пользователя	string
fioFull	ФИО исполнителя (полное)	string
fioShort	ФИО исполнителя (краткое)	string
post	Должность исполнителя	string
department	Организация / подразделение	
departmentFull	Название полное	string
departmentCode	Код	string
departmentShort	Название краткое	string

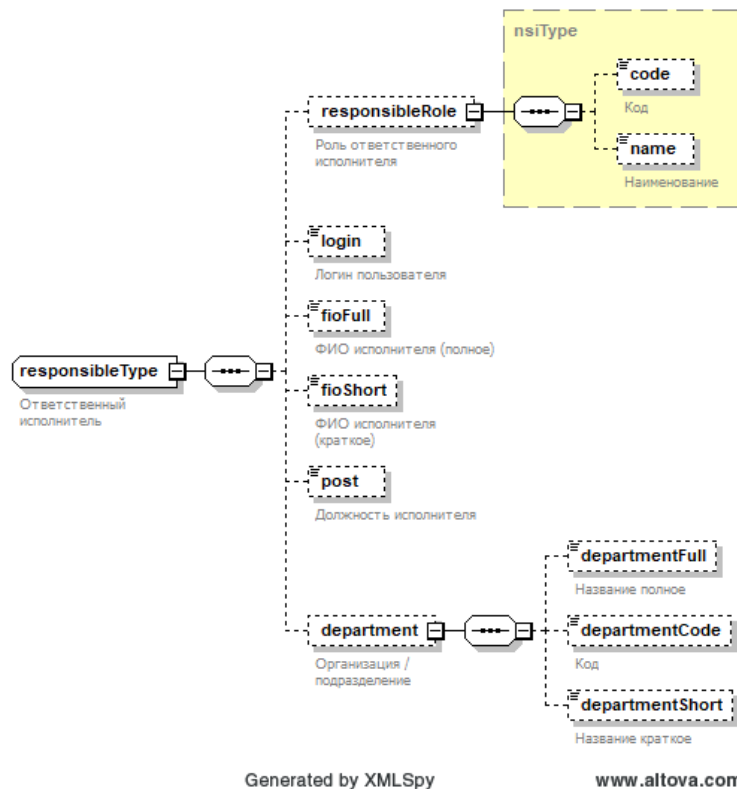


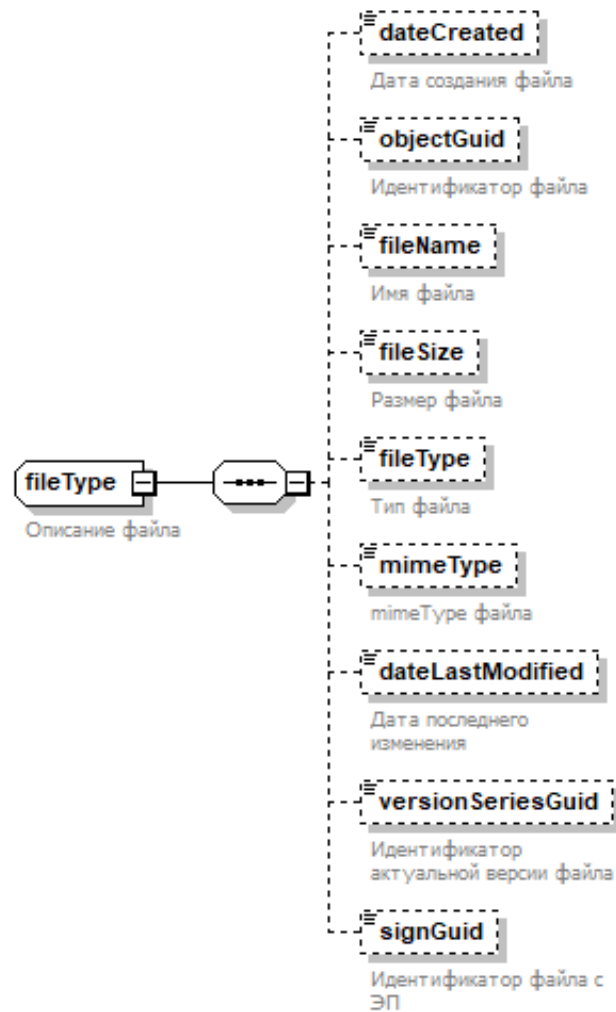
Рисунок 81 – Структура responsibleType Ответственный исполнитель

4.6.2.4.4 fileType Описание файла

Описание (Таблица 30) и структура (Рисунок 82) типа «Описание файла» приведены ниже.

Таблица 30 – Описание типа «Описание файла»

Наименование	Описание	Тип
dateCreated	Дата создания файла	dateTime
objectGuid	Идентификатор файла	string
fileName	Имя файла	string
fileSize	Размер файла	integer
fileType	Тип файла	string
mimeType	mimeType файла	string
dateLastModified	Дата последнего изменения	dateTime
versionSeriesGuid	Идентификатор актуальной версии файла	string
signGuid	Идентификатор файла с ЭП	string



Generated by XMLSpy

www.altova.com

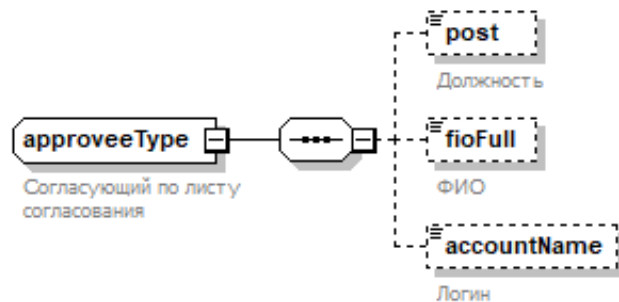
Рисунок 82 – Структура fileType Описание файла

4.6.2.4.5 approveType Согласующий по листу согласования

Описание (Таблица 31) и структура (Рисунок 83) типа «Согласующий по листу согласования» приведены ниже.

Таблица 31 – Описание типа «Согласующий по листу согласования»

Наименование	Описание	Тип
post	Должность	string
fioFull	ФИО	string
accountName	Логин	string



Generated by XMLSpy

www.altova.com

Рисунок 83 – Структура approveeType Согласующий по листу согласования

4.6.2.4.6 approvalCycleType Согласование по листу согласования

Указанная структура не поддерживает согласование с добавлением дополнительных согласующих.

Для согласования по листу согласования с сокращенной структурой хранения файла и датой в качестве даты начала цикла (Вариант 1) описание (Таблица 32) и структура (Рисунок 84) данного типа приведены ниже.

Таблица 32 – Описание типа «Согласование по листу согласования» с сокращенной структурой хранения файла

Наименование	Описание	Тип
approvalCycleNum	Номер цикла согласования	string
approvalCycleDate	Дата начала цикла согласования	date
approvalCycleFile	Документы согласования	string
agreed	Согласовано	
approvalNum	Номер согласования	string
approvalTime	Продолжительность согласования	string
agreedBy	Согласовал	approveeType
approvalFactDate	Фактическая дата согласования	dateTime
approvalResult	Результат согласования	string
approvalType	Вид согласования	string
approvalTypeCode	Код вида согласования	string
approvalNote	Комментарий	string
fileApproval	Файл согласования	string

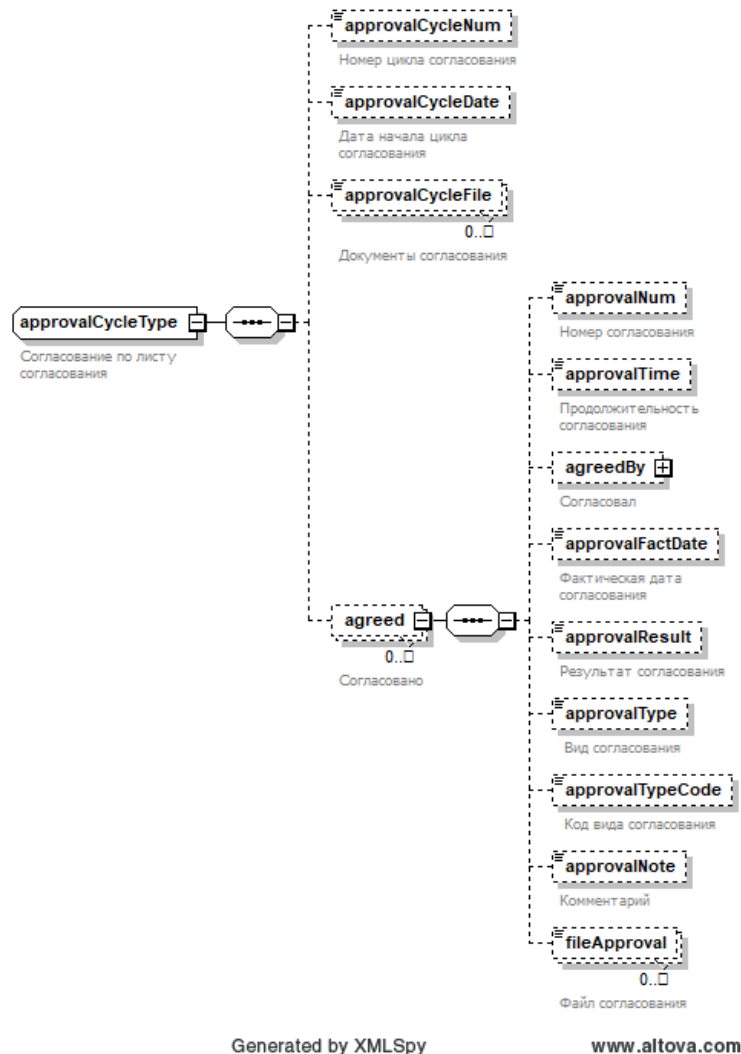


Рисунок 84 – Структура approvalCycleType Согласование по листу согласования. Вариант 1

Для согласования по листу согласования с расширенной структурой хранения файла и датой в качестве даты начала цикла (Вариант 2) описание (Таблица 33) и структура (Рисунок 85) данного типа приведены ниже.

Таблица 33 – Описание типа «Согласование по листу согласования» с расширенной структурой хранения файла

Наименование	Описание	Тип
approvalCycleNum	Номер цикла согласования	string
approvalCycleDate	Дата начала цикла согласования	date
approvalCycleFile	Документы согласования	fileType
agreed	Согласовано	
approvalNum	Номер согласования	string
approvalTime	Продолжительность согласования	string
agreedBy	Согласовал	approveeType
approvalFactDate	Фактическая дата согласования	dateTime

Наименование	Описание	Тип
approvalResult	Результат согласования	string
approvalType	Вид согласования	string
approvalTypeCode	Код вида согласования	string
approvalNote	Комментарий	string
fileApproval	Файл согласования	fileType

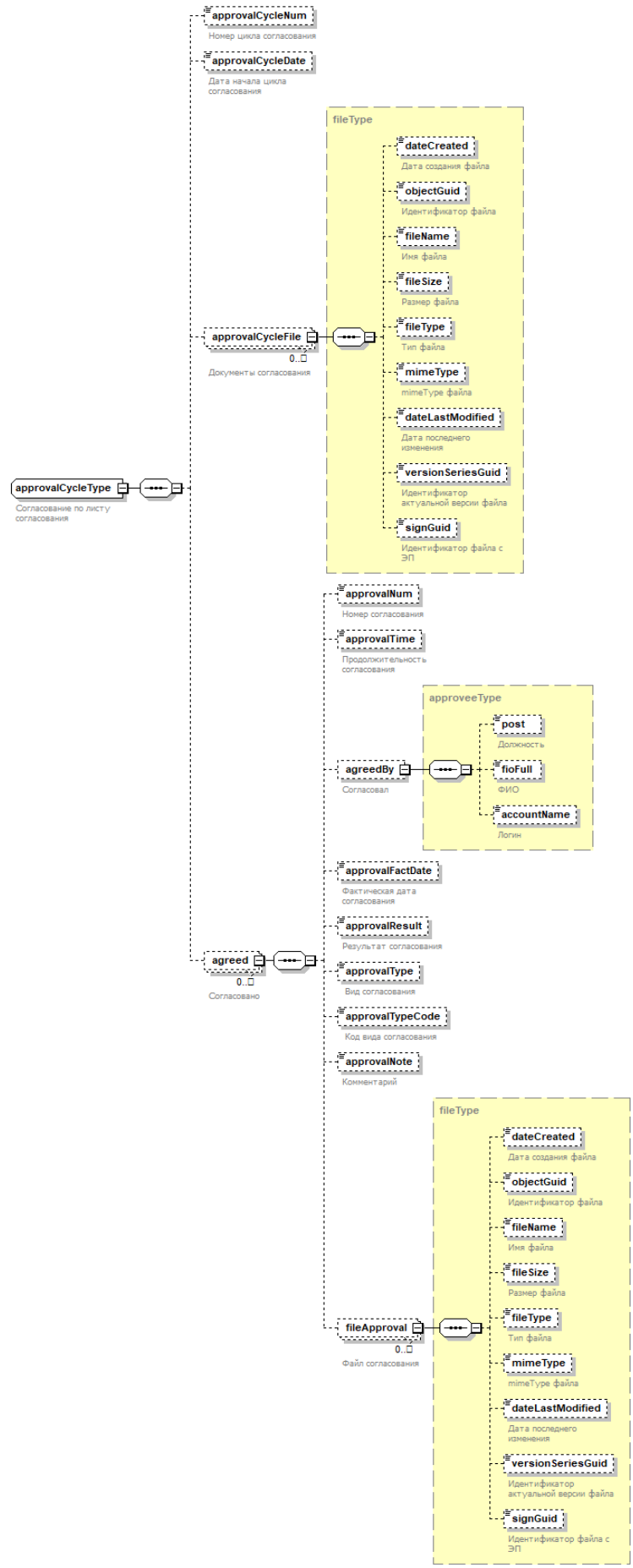


Рисунок 85 – Структура approvalCycleType Согласование по листу согласования. Вариант 2
 Для согласования по листу согласования с сокращенной структурой хранения файла и

датой и временем в качестве даты начала цикла (Вариант 3) описание (Таблица 34) и структура (Рисунок 86) данного типа приведены ниже.

Таблица 34 – Описание типа «Согласование по листу согласования» с расширенной структурой хранения файла

Наименование	Описание	Тип
approvalCycleNum	Номер цикла согласования	string
approvalCycleDate	Дата начала цикла согласования	dateTime
approvalCycleFile	Документы согласования	string
agreed	Согласовано	
approvalNum	Номер согласования	string
approvalTime	Продолжительность согласования	string
agreedBy	Согласовал	approveeType
approvalFactDate	Фактическая дата согласования	dateTime
approvalResult	Результат согласования	string
approvalType	Вид согласования	string
approvalTypeCode	Код вида согласования	string
approvalNote	Комментарий	string
fileApproval	Файл согласования	string

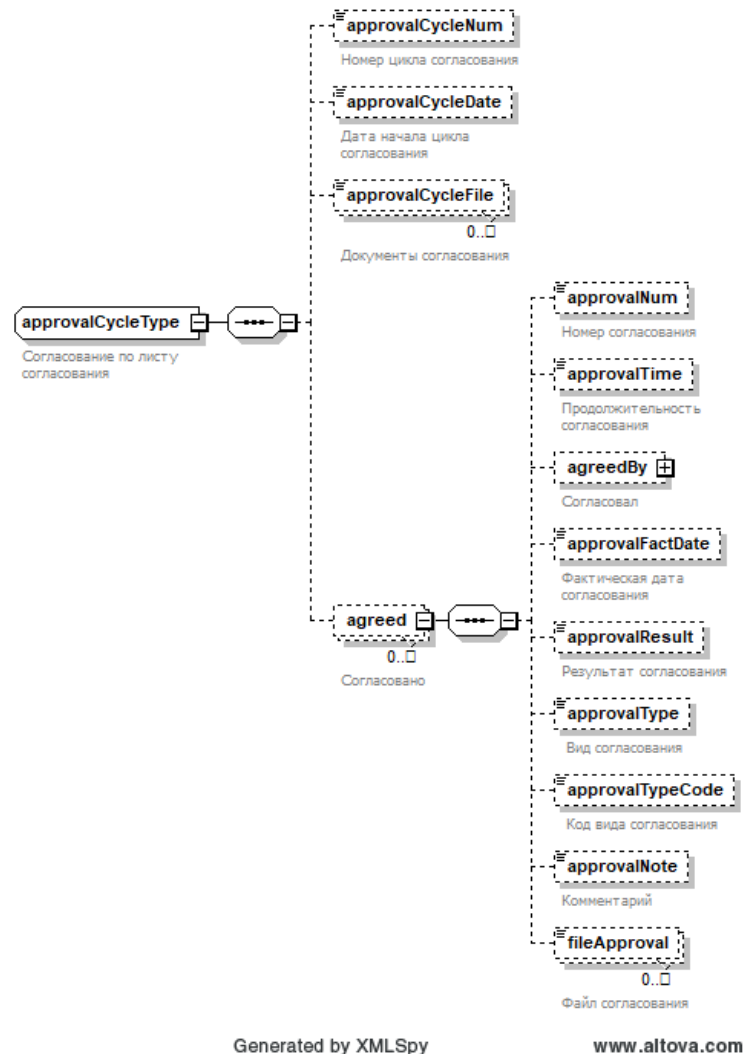


Рисунок 86 – Структура approvalCycleType Соглашение по листу согласования. Вариант 3

Для согласования по листу согласования с расширенной структурой хранения файла и датой и временем в качестве даты начала цикла (Вариант 4) описание (Таблица 35) и структура (Рисунок 87) данного типа приведены ниже.

Таблица 35 – Описание типа «Согласование по листу согласования» с расширенной структурой хранения файла

Наименование	Описание	Тип
approvalCycleNum	Номер цикла согласования	string
approvalCycleDate	Дата начала цикла согласования	dateTime
approvalCycleFile	Документы согласования	fileType
agreed	Согласовано	
approvalNum	Номер согласования	string
approvalTime	Продолжительность согласования	string
agreedBy	Согласовал	approveeType
approvalFactDate	Фактическая дата согласования	dateTime

Наименование	Описание	Тип
approvalResult	Результат согласования	string
approvalType	Вид согласования	string
approvalTypeCode	Код вида согласования	string
approvalNote	Комментарий	string
fileApproval	Файл согласования	fileType

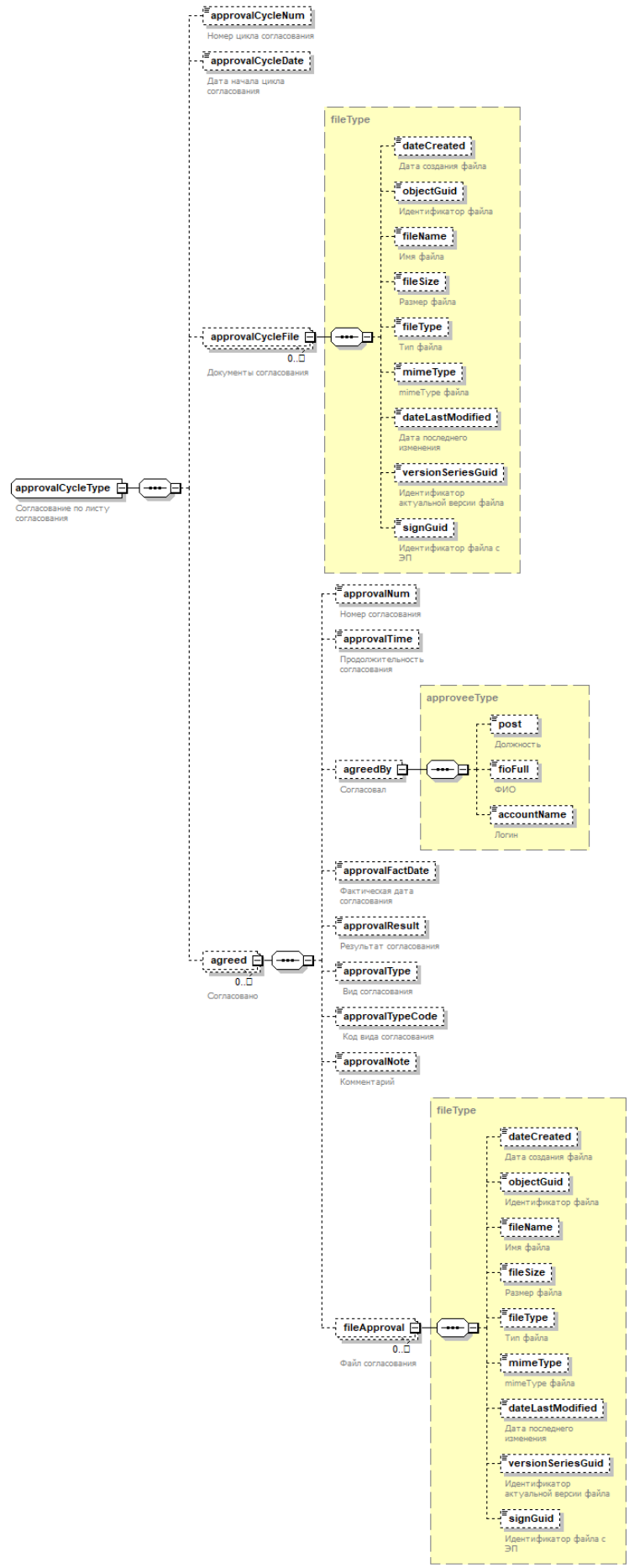


Рисунок 87 – Структура approvalCycleType Согласование по листу согласования. Вариант 4

4.6.2.4.7 folderType Проектная библиотека

Описание (Таблица 36) и структура (Рисунок 88) типа «Проектная библиотека» приведены ниже.

Таблица 36 – Описание типа «Проектная библиотека»

Наименование	Описание	Тип
folderName	Название папки	string
filesID	Идентификаторы файлов	string
folder	Папка	folderType

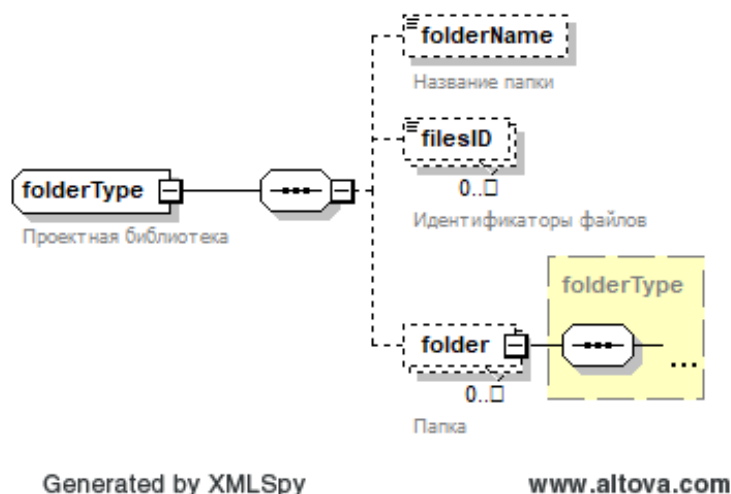


Рисунок 88 – Структура folderType Проектная библиотека

4.6.2.4.8 versionType Версионность

В качестве «seriesId» должен быть указан идентификатор первой версии документа. Для документов в версии 1 атрибуты «documentId» и «seriesId» будут совпадать.

Описание (Таблица 37) и структура (Рисунок 89) типа «Версионность» приведены ниже.

Таблица 37 – Описание типа «Версионность»

Наименование	Описание	Тип
seriesId	Идентификатор серии	string
number	Номер версии	integer
createDate	Дата и время создания версии	dateTime
author	Автор версии	userType
base	Основание создания версии	string

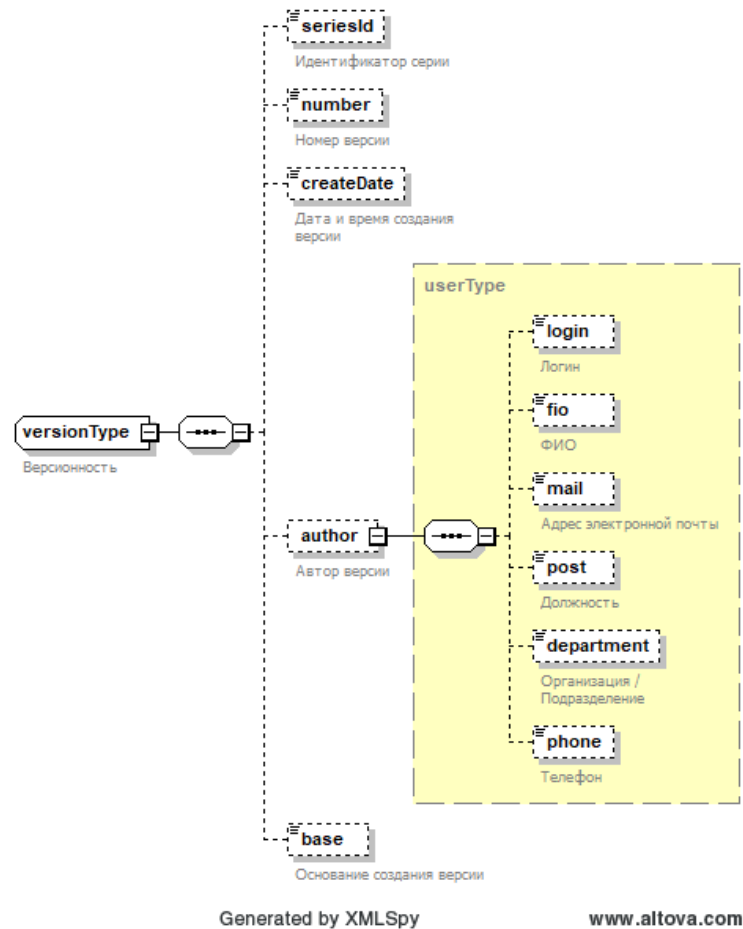


Рисунок 89 – Структура versionType Версионность

4.6.3 Реестр бизнес-процессов

Пример реестра бизнес-процессов приведен ниже (Рисунок 90).

Код процесса	Наименование процесса	Подсистема	Действия
demopropusk_issuePermit	Выдача пропусков	DEMO_DYNADEMO	🔍 📄 📄
demodemo_Demo	Демо-процесс	DEMO_DEMO	🔍 📄 📄
demoobject_demoProcess	Демо-процесс	DEMO_DYNADEMO	🔍 📄 📄
demoformsdemo_bpmUsecases	Демонстрация функциональных возможностей редактора BPM	DEMO_FORMSDEMO	🔍 📄 📄
demodemo_supertestnew	Новый тестовый демо документ	DEMO_DEMO	🔍 📄 📄
democontract_Signing	Подготовка и подписание контракта	DEMO_DYNADEMO	🔍 📄 📄
demoformsdemo_install	Работа по заявке на установку компьютера	DEMO_FORMSDEMO	🔍 📄 📄

Рисунок 90 – Пример реестра бизнес-процессов

Список систем в поле «Система» содержит выпадающий список только с системами, к которым у пользователя есть доступ и для которых настроены флексыформы.

Реестр бизнес-процессов содержит функциональные иконки, описание которых приведено ниже (Таблица 38).

Таблица 38 – Описание функциональных иконок реестра бизнес-процессов

Иконка	Действие
★	Добавление бизнес-процесса в избранное
🔍	Просмотр диаграммы бизнес-процесса
⬇️	Выгрузка специфического JSON примера (используется в режиме предпросмотра формы)
⬆️	Загрузка специфического JSON примера (используется в режиме предпросмотра формы)

В случае если специфический тестовый JSON пример не загружен, используется типовой, формируемый методом генерации json-примера по XSD схеме.

Для работы типового JSON примера в прикладном приложении должна быть подключена библиотека schema-lib.

4.6.3.1 Список задач бизнес-процесса

При переходе по гиперссылке в строке реестра бизнес-процессов открывается форма описания бизнес-процесса. Форма содержит схему и перечень форм задач бизнес-процесса. Перечень задач бизнес-процесса имеет табличный вид с колонками: Наличие, Код формы, Наименование формы, Действия (Рисунок 91).

▲ Схема бизнес-процесса

релиз: 3 от: 2019-12-20 10:45:25






Формы задач бизнес-процесса:

Наличие	Код формы	Наименование формы	Действия
Отсутствует	Отсутствует	Отсутствует	✎ ⬆️
Имеется	dyna_formFillContractData	Сформировать данные контракта	✎ 🗑️ ⬇️ ⬆️ 🔗
Имеется	dyna_formNegotiateContract_A	Согласование контракта - сторона А	✎ 🗑️ ⬇️ ⬆️ 🔗
Отсутствует	dyna_formPrintContract	Отсутствует	✎ ⬆️
Имеется	dyna_formNegotiateContract_B	Согласование контракта - сторона В	✎ 🗑️ ⬇️ ⬆️ 🔗
Отсутствует	dyna_formEditContractData	Отсутствует	✎ ⬆️

Рисунок 91 – Форма описания бизнес-процесса

Колонка «Действия» перечня форм задач бизнес-процесса содержит функциональные иконки, описание которых приведено ниже (Таблица 38).

Таблица 39 – Описание функциональных иконок колонки «Действия»

Иконка	Действие
	Переход к созданию / редактированию формы
	Копирование формы в другую задачу того же бизнес-процесса
	Выгрузка JSON с настройками формы
	Загрузка JSON с настройками формы
	Установление связи с композитным объектом

4.6.4 Стартовая форма

Ссылка на стартовую форму формируется по следующему алгоритму:

`/sys/flex/#/app/flexforms/start/{ключ_бизнес_процесса}?systemCode={код_системы}`.

4.6.5 Общая информация о создании формы

Для создания флексформы необходимо выполнить следующие действия:

- выбрать типы раскладки элементов на форме (при необходимости);
- разместить и настроить элементы на форме;
- связать элементы с атрибутами бизнес-объекта, определёнными ранее в XSD схеме или в композитном объекте.

Для связывания элементов форм с атрибутами бизнес-объекта в редакторе флексформ обеспечена возможность просмотра структуры XSD схемы документа или композитного объекта (при установлении связи с ним) и копирования в буфер обмена пути в JSON от корня объекта до выбранного элемента (JSON Path) (Рисунок 92).

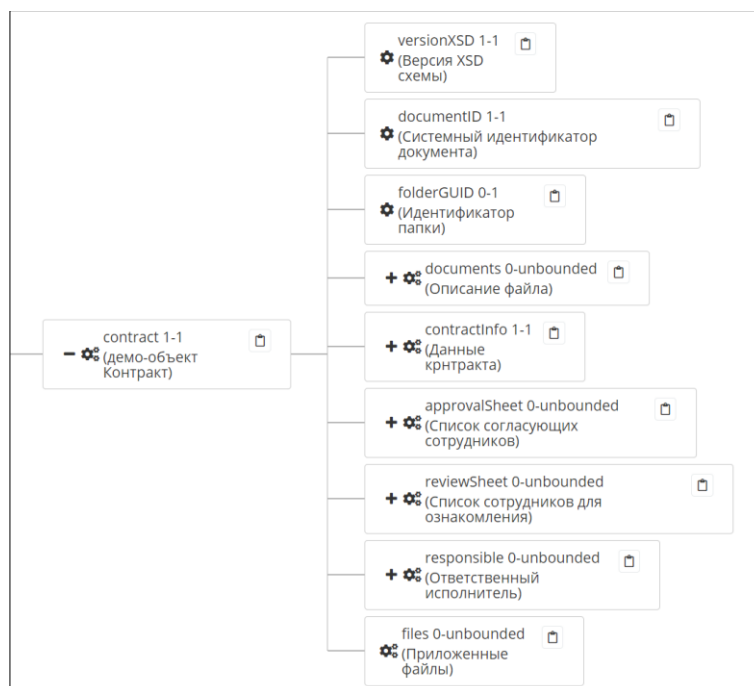



Рисунок 92 – Просмотр структуры XSD схемы

Иконка  обеспечивает копирование пути до выбранного атрибута.

Скопированный в буфер обмена путь до элемента данных необходимо вставить в поле «Property Name» на вкладке API (имеются исключения: подробности в описании компонентов) редактируемого элемента формы.

При первоначальном заполнении лейбла элемента формы Form.io формирует значение по умолчанию поля «Property Name» на вкладке API. Если метка содержит буквы русского алфавита, то поле будет заполнено значением с некорректным синтаксисом, что не даст сохранить элемент на форме. Рекомендуется перед первым редактированием элемента формы скопировать необходимый JSON Path в буфер обмена и вставить его в поле «Property Name» на вкладке API. Если Вы забыли скопировать в clipboard JSON Path заранее, то чтобы иметь возможность сохранить элемент на форме можно временно заполнить поле «Property Name» на вкладке API любыми буквами латинского алфавита, а позже не забыть заменить их на корректный путь в структуре данных.

4.6.6 Компоненты

4.6.6.1 Basic Components

Описание компонента Basic Components доступно по ссылке: <https://help.form.io/userguide/form-building/form-components>.

4.6.6.2 Advanced

Описание компонента Advanced доступно по ссылке: <https://help.form.io/userguide/form-building/advanced-components>.

4.6.6.2.1 Date Time Component. Календарь

Данный компонент предназначен для добавления на форму компонента для выбора даты или даты/времени из календаря (Рисунок 93).

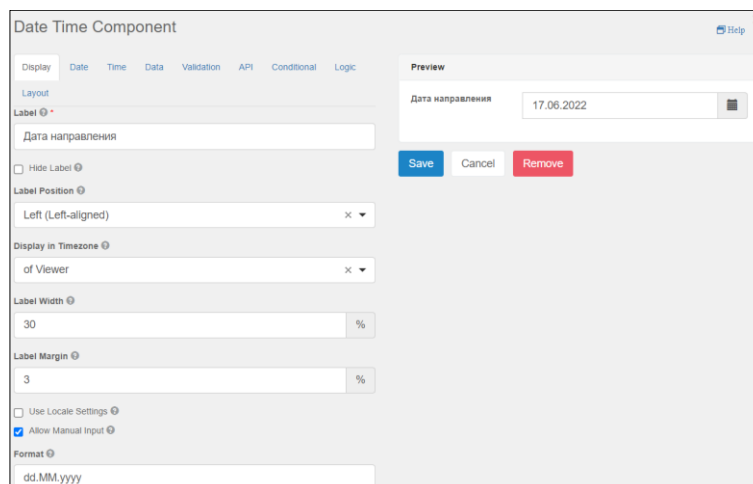


Рисунок 93 – Date Time Component (Календарь)

Для корректной работы компонента при использовании предзаполнения на вкладке Display обязательно должны быть заполнены следующие поля:

- Format: отвечает за формат отображения даты в пользовательском интерфейсе (обычно dd.MM.yyyy);
- Output (JSON) Date Format: отвечает за формат сохранения даты в JSON (обычно yyyy-MM-dd для даты и yyyy-MM-ddThh:mm:ss для даты и времени).

Для автоматического заполнения поля с датой текущей датой в формате yyyy-MM-dd необходимо добавить следующий код в поле «Custom Default Value» на вкладке «Data»:

```
var d = new Date();  
value = d.toISOString().substring(0,10).
```

Если при выборе значения из календаря не закрывается окно выбора, необходимо снять флаг «Enable Time Input» на вкладке «Time».

4.6.6.2.2 HTML Element. Использование html-разметки

Данный компонент предназначен для добавления на форму заголовков, строк-разделителей и прочего.

Настройка компонента для добавления строки-разделителя приведена ниже (Рисунок 94).

Рисунок 94 – Настройка компонента для добавления строки-разделителя
 Настройка компонента для добавления заголовка в поле «Content» приведена ниже
 (Рисунок 95).

Рисунок 95 – Настройка компонента для добавления заголовка

Настройка компонента для добавления заголовка с отступом от предыдущего элемента формы в поле «Content» (`<div style="margin-top:20px;"> Автор заявки </div>`) приведена ниже (Рисунок 96).

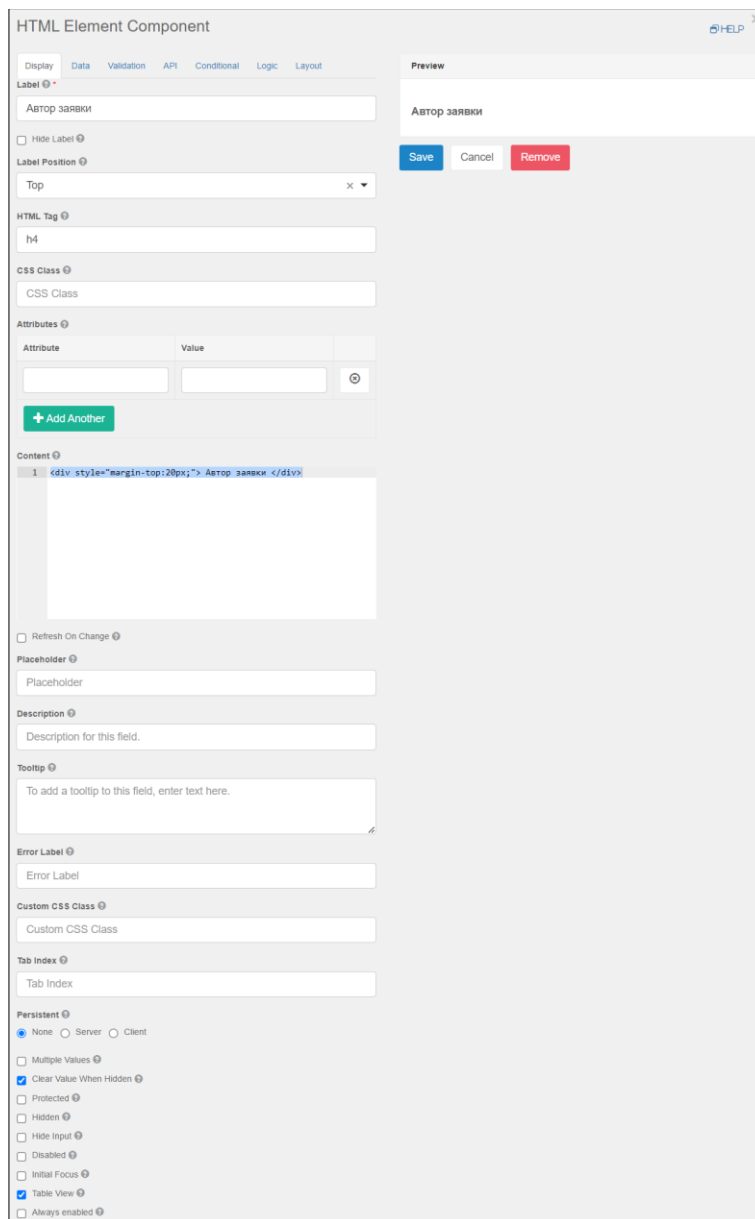


Рисунок 96 – Настройка компонента для добавления заголовка

4.6.6.3 Layout

Описание компонента Layout доступно по ссылке: <https://help.form.io/userguide/form-building/layout-components>

4.6.6.4 Data

Описание компонента Data доступно по ссылке: <https://help.form.io/userguide/form-building/data-components>

4.6.6.5 CDP Components

4.6.6.5.1 FileManager. Компонент для сохранения файлов в хранилище

Компонент предназначен для добавления на форму задачи поля для прикладывания файлов.

Для корректной работы компонента хранение файлов в JSON документа должно соответствовать одной из двух структур, указанных ниже:

1) при установленном на вкладке «Data» флаге «Only ID in JSON» на вкладке API необходимо указывать путь до поля с массива, предназначенного для хранения идентификатора сохраненного файла (Рисунок 97);

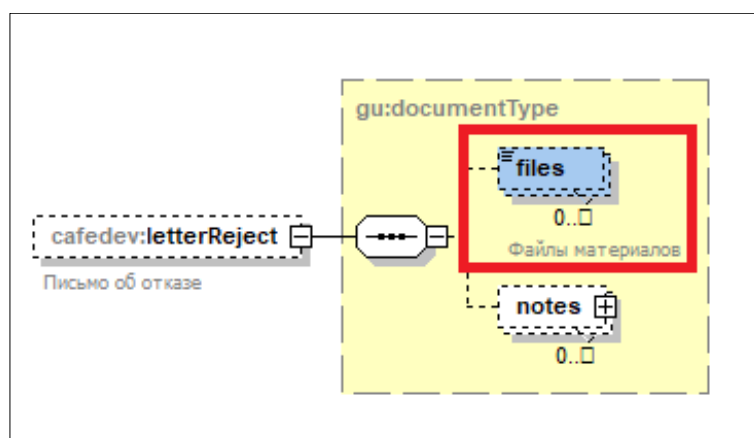


Рисунок 97 – Массив хранения идентификатора сохраненного файл

2) при неустановленном на вкладке «Data» флаге «Only ID in JSON» на вкладке API (Рисунок 98) необходимо указывать путь до массива, содержащего расширенную информацию о файле (Рисунок 99):

- dateCreated;
- dateLastModified;
- fileName;
- fileSize;
- fileType;
- mimeType;
- objectGuid;
- signGuid;
- versionSeriesGuid.

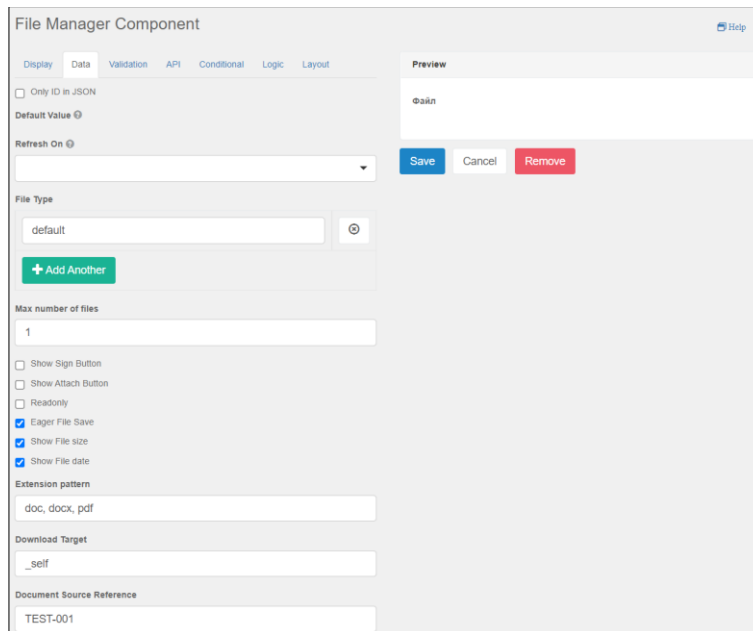


Рисунок 98 – Вкладка «Data», флаг «Only ID in JSON» не установлен

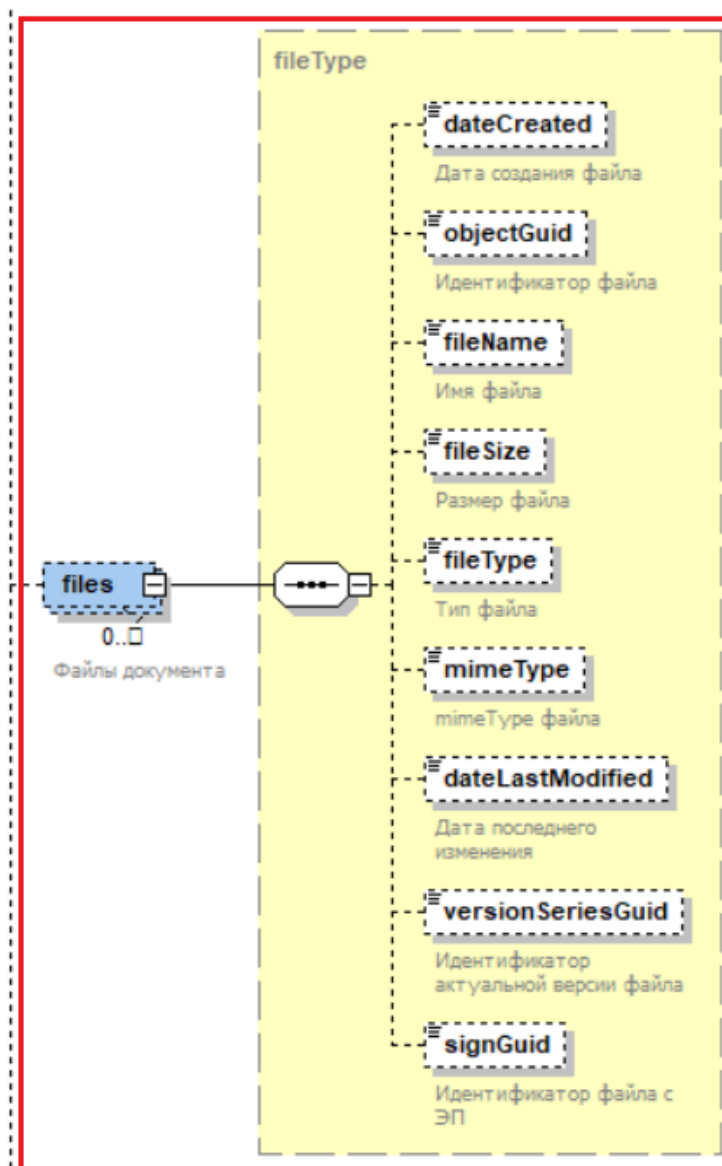


Рисунок 99 – Массив, содержащий расширенную информацию о файле
 На вкладке «Data» могут быть настроены параметры загружаемых файлов

(Таблица 40).

Таблица 40 – Параметры загружаемых файлов на вкладке «Data»

Наименование	Описание
Only ID in JSON	Структура хранения данных (см. подробности выше) По умолчанию признак не установлен
File Type	Тип документа для сохранения в хранилище Alfresco По умолчанию default Тип документа - это не расширение файла. Это дополнительное обязательное свойства файла в хранилище. Чтобы загруженный файл был виден на следующих формах задач или в карточке, File Type указанный при загрузке должен обязательно совпадать с File Type , указанным при отображении
Max number of files	Максимальное количество файлов для загрузки По умолчанию 1
Show Sign Button	Отображение ссылки на подписание файла По умолчанию признак установлен
Show Attach Button	Отображение ссылки на прикладывания электронной подписи По умолчанию признак установлен
Eager File Save	Сохранение JSON с данными, не дожидаясь стандартного асинхронного сохранения перед закрытием формы По умолчанию признак установлен
Show File size	Отображение размера загруженного файла
Show File date	Отображение даты загрузки файла
Extension pattern	Допустимые расширения файлов По умолчанию doc, docx, pdf

4.6.6.5.2 Complete Button. Кнопка завершения задачи

Компонент предназначен для добавления на форму задачи кнопки завершения задачи процесса.

При нажатии кнопки завершения задачи выполняются следующие действия:

- сохранение документа процесса;
- вызов внешнего REST-API (опционально);
- подписание файла (опционально);
- визуализация электронной подписи (опционально);
- дополнение данных печатной формы при подписании (опционально);
- завершение задачи процесса.

Для предотвращения завершения задачи до заполнения обязательных полей или с ошибками валидации необходимо в настройках компонента на вкладке «Display» установить флаг «Disable on Form Invalid». При установленном флаге кнопка будет заблокирована от

нажатия при наличии на форме ошибок заполнения (Рисунок 100).

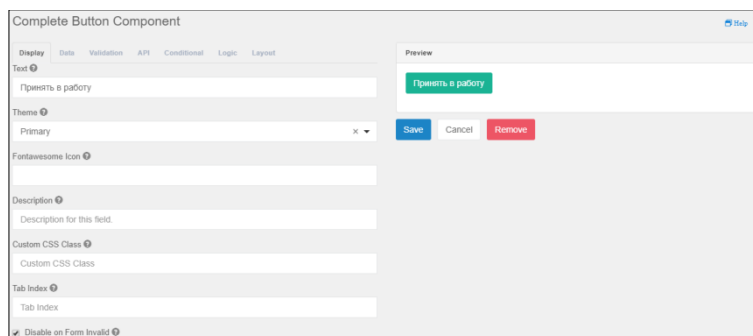


Рисунок 100 – Флаг «Disable on Form Invalid»

При наличии в настройках процесса обязательной переменной (или переменных), которая должна передаваться при завершении задачи процесса (например, переменная определяющая дальнейшие правила маршрутизации процесса), такая переменная автоматически занесется в поле «Variable Name» на вкладке «Data». Значение, указанное в поле «Value», будет передано при завершении задачи.

При наличии двух и более переменных их значения должны быть разделены запятой. В качестве значения переменной может быть использовано значение поля из документа. В таком случае путь к полю должен быть указан в фигурных скобках (Рисунок 101).

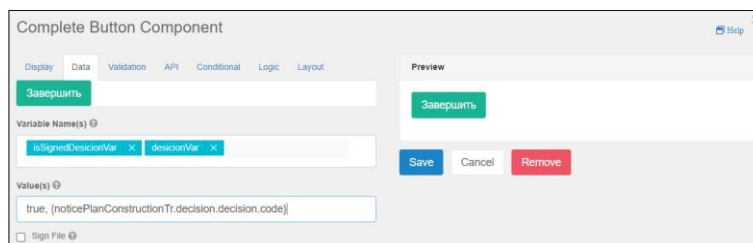


Рисунок 101 – Путь к полю указан в фигурных скобках

В указанном примере после нажатия кнопки «Завершить» при завершении задачи процесса будут переданы следующие переменные:

- isSignedDesicionVar = true;
- desicionVar = значению поля noticePlanConstructionTr.decision.decision.code.

Для вызова функционала подписания файла при нажатии кнопки необходимо установить флаг «Sign File» на вкладке «Data» и указать путь к подписываемому файлу в отобразившемся поле «SignFileId» (Рисунок 102).

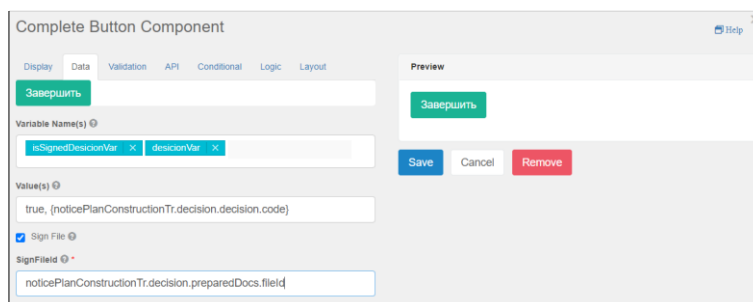


Рисунок 102 – Флаг «Sign File» на вкладке «Data»

4.6.6.5.3 Визуализация электронной подписи

Визуализация электронной подписи возможна только для файлов формата doc и docx.

При установке флага «Visualize Sign» будут выполнены следующие действия:

- заполнение в шаблоне данных в операторных скобках `sn` на основании данных в электронной подписи (Рисунок 103, Рисунок 104);
- формирование pdf-файла из обновленного файла формата doc и docx;
- запись идентификатора pdf-файла в узел, где ранее был сохранен файл формата doc и docx.

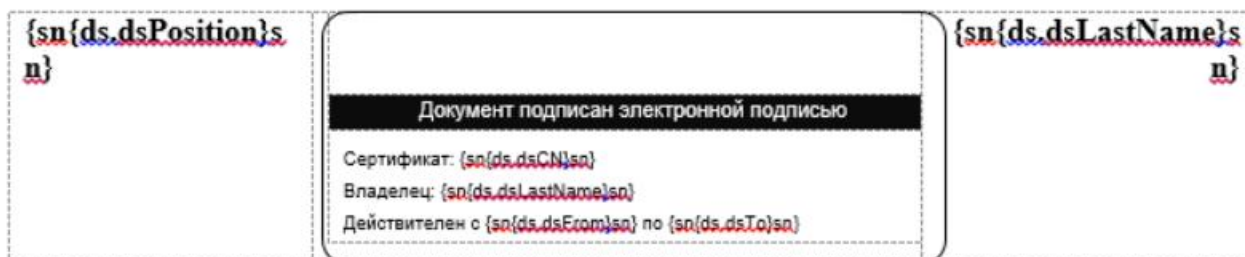


Рисунок 103 – Пример заполнения операторных скобок `sn`

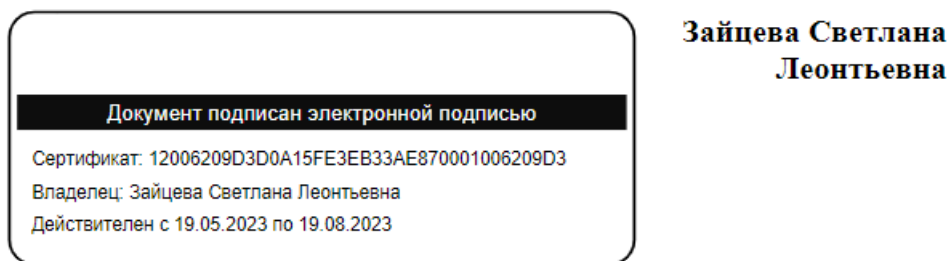


Рисунок 104 – Пример результата визуализации

4.6.6.5.4 Дополнение данных печатной формы при подписании

Дополнение данных печатной формы при подписании только для файлов формата doc и docx.

При установке флага «Report» и заполнении «Report FileId» одновременно с подписанием будет выполнено заполнение в шаблоне данных в операторных скобках `ap` (внутри операторных скобок `sn`) на основании данных в «Report FileId» (Рисунок 105, Рисунок 106).

Пример заполнения данных в поле «Report FileId»:

```
"date" : "{{visit.interactionCP.regDateNotice}}", "regNum":  
"{{visit.interactionCP.regNumNotice}}".
```

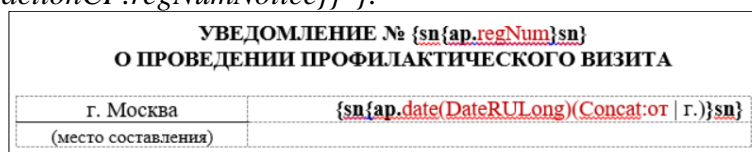


Рисунок 105 – Пример заполнения данных шаблона в операторных скобках `ap`

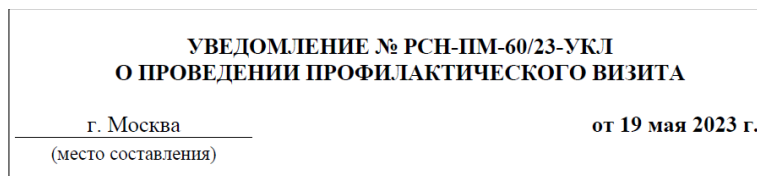


Рисунок 106 – Пример результата дополнения данных

4.6.6.5.5 Save Button. Кнопка сохранения данных формы без завершения

Компонент предназначен для добавления на форму задачи кнопки сохранения данных документа без завершения задачи процесса (Рисунок 107).

При нажатии кнопки выполняется сохранение документа процесса без закрытия формы и завершения текущей задачи процесса.

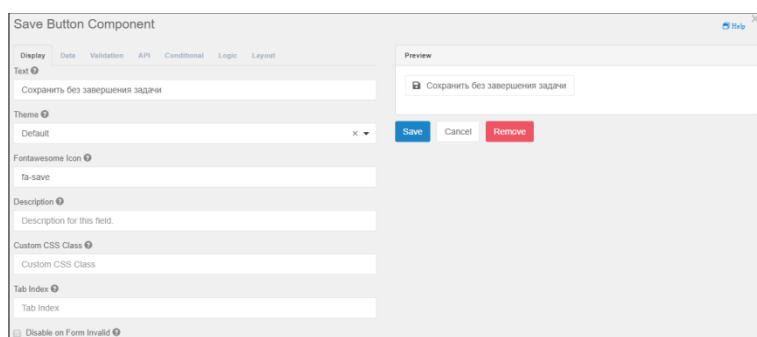


Рисунок 107 – Save Button компонент

По умолчанию после сохранения изменений форма не закрывается. Если необходимо вернуться в витрину с задачами после нажатия кнопки, следует установить флаг «Close after saving» на вкладке «Data».

4.6.6.5.6 Launch Process Button. Кнопка запуска бизнес-процесса

Компонент предназначен для добавления кнопки только на стартовую форму бизнес-процесса (Рисунок 108).

При нажатии кнопки выполняется создание документа заданного типа и запуск выбранного бизнес-процесса по созданному документу.

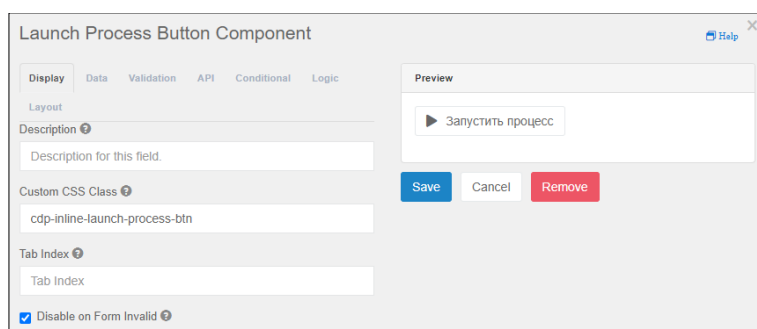


Рисунок 108 – Launch Process Button компонент

4.6.6.5.7 Close Button. Кнопка закрытия задачи без завершения

Компонент предназначен для добавления на форму задачи кнопки закрытия формы без завершения задачи процесса (Рисунок 109).

При нажатии кнопки выполняется возврат пользователя к витрине с задачами без

сохранения документа и завершения текущей задачи процесса.

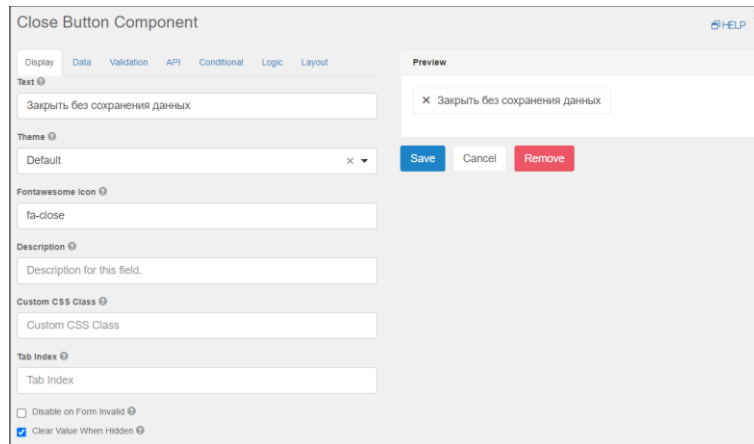


Рисунок 109 – Close Button компонент

4.6.6.5.8 Assign. Назначение ответственного исполнителя

Компонент предназначен для работы с функционалом назначения ответственного исполнителя.

Для корректной работы компонента Assign хранение данных ответственного исполнителя в JSON документа должно соответствовать структуре, приведенной ниже (Рисунок 110).

Добавленный узел с типом «responsibleType» обязательно должен быть массивом.

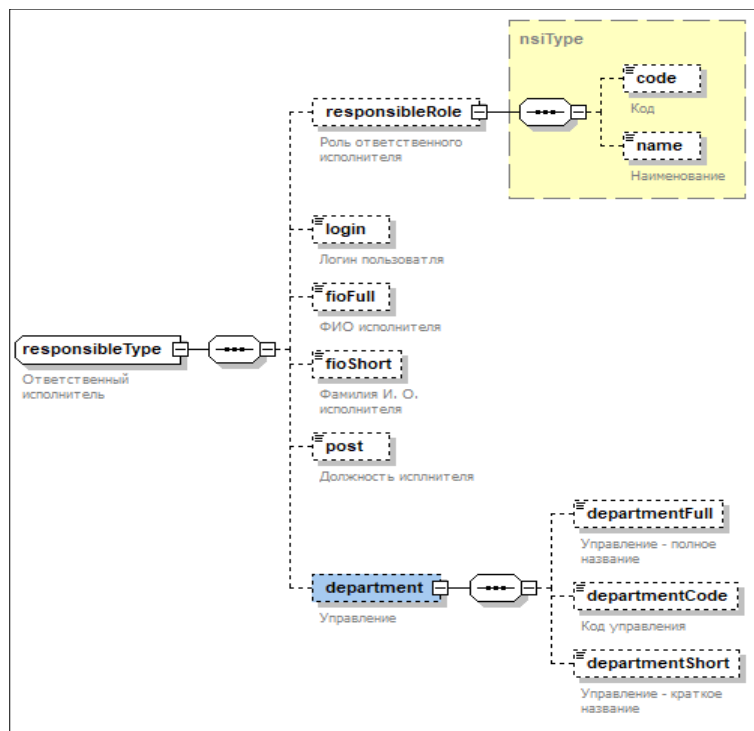


Рисунок 110 – Узел с типом «responsibleType»

На вкладке «Display» добавлены специфические поля для работы данного компонента (Рисунок 111).

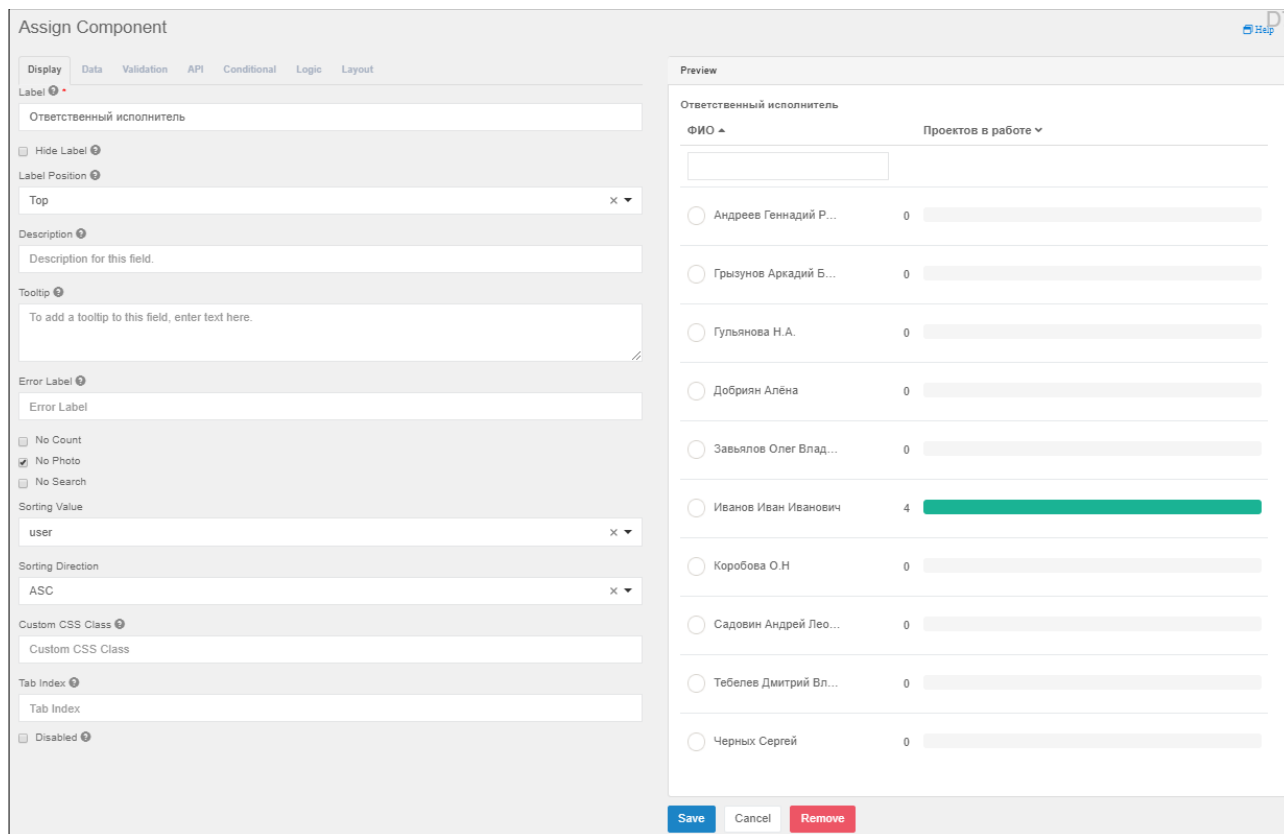


Рисунок 111 – Вкладка «Display»

Таблица 41 – Специфические поля вкладки «Display»

Наименование поля	Обязательность	Описание
No Count	Нет	Чекбокс Предназначено для скрытия/отображения информации о количестве проектов пользователей
No Photo	Нет	Чекбокс Предназначено для скрытия/отображения фотографий пользователей
No Search	Нет	Чекбокс Предназначено для скрытия/отображения поисковой строки
Sorting Value	Нет	Список с единичным выбором Предназначено для указания элемента списка, по которому осуществляется сортировка значений Наполняется значениями user и count
Sorting Direction	Нет	Список с единичным выбором Предназначено для указания направления сортировки значений списка по полю, выбранному в Sorting Value Наполняется значениями ASC и DESC

На вкладке «Data» добавлены специфические поля для работы данного компонента (Таблица 42).

Таблица 42 – Специфические поля вкладки «Data»

Наименование поля	Обязательность	Описание
REST	Да	<p>Поле ввода</p> <p>Предназначено для указания относительной ссылки на REST-сервис, возвращающий информацию о пользователях, которые могут быть назначены в качестве ответственных исполнителей</p> <p>Может быть использован стандартный сервис: /app/sys/flexlogic/users/responsiblelist-grp</p>
JSON	Да	<p>Расширяемый текстовый блок</p> <p>Предназначено для указания JSON-запроса к REST-сервису.</p> <p>При использовании стандартного сервиса JSON-файл должен содержать следующие параметры:</p> <p>group (массив) – перечень групп доступа, в которые включены пользователи (поиск по AND, если не указано иное в operation)</p> <p>operation - способ объединения перечня групп доступа (OR или AND)</p> <p>orgstruct - перечень узлов ОИШС (reinformNodeID), в которые включены пользователи (поиск по OR, значение из operation не применяется)</p> <p>query – запрос к search, включая обязательный параметр {userLogin}, куда автоматически подставляется пользователи, включенные в группы доступа из параметра group</p> <p>types – тип документа search</p> <pre> { "group": ["DEMO_RECONSTRUCTOR_RESPONSIBLE", "DEMO_RECONSTRUCTOR_EXPERT"], "query": "(status:\\"В работе\") AND (agrExpertLogin:\\"{userLogin}\")", "types": ["DEMO_RECONSTRUCTOR_PURCHASE"] } </pre> <p>В случае если возврат данных по количеству проектов не требуется, то query и types указывать не следует</p> <pre> { "group": ["UKS_PROD_SPEC", "UKS_TEH_SPEC"], "operation": "OR" } </pre> <p>Пример использования данных из ОИШС</p> <pre> { "orgstruct": [</pre>

Наименование поля	Обязательность	Описание
		"42d95cdc-310e-41eb-a531-f1f05c5be161"] }
Role Code	Да	Поле вводаПредназначено для указания кода роли ответственного исполнителя
Role Name	Да	Поле вводаПредназначено для указания наименования роли ответственного исполнителя

Требования к REST-сервису, возвращающему информацию о пользователях (в случае использования нестандартного сервиса):

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XMLSpy v2011 rel. 2 (http://www.altova.com) by TeaM DJiNN (TeaM DJiNN) -->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
  <xs:element name="responsible">
    <xs:annotation>
      <xs:documentation>Назначение исполнителя</xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:sequence>
        <xs:element name="user">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="photoURL"
type="xs:string" minOccurs="0">
                <xs:annotation>
                  <xs:documentation>Фотография пользователя</xs:documentation>
                </xs:annotation>
              </xs:element>
              <xs:element name="login" type="xs:string">
                <xs:annotation>
                  <xs:documentation>Логин
пользователя</xs:documentation>
                </xs:annotation>
              </xs:element>
              <xs:element name="fioFull"
type="xs:string">
                <xs:annotation>
                  <xs:documentation>ФИО
исполнителя</xs:documentation>
                </xs:annotation>
              </xs:element>
              <xs:element name="fioShort"
type="xs:string">
                <xs:annotation>
                  <xs:documentation>Фамилия
И. О. исполнителя</xs:documentation>
```

```

        </xs:annotation>
    </xs:element>
    <xs:element name="post" type="xs:string"
minOccurs="0">
        <xs:annotation>
            <xs:documentation>Должность исполнителя</xs:documentation>
        </xs:annotation>
    </xs:element>
    <xs:element name="department"
minOccurs="0">
        <xs:annotation>
            <xs:documentation>Управление</xs:documentation>
        </xs:annotation>
        <xs:complexType>
            <xs:sequence>
                <xs:element
name="departmentFull" type="xs:string" minOccurs="0">
                    <xs:annotation>
                        <xs:documentation>Управление - полное название</xs:documentation>
                    </xs:annotation>
                </xs:element>
                <xs:element
name="departmentCode" type="xs:string" minOccurs="0">
                    <xs:annotation>
                        <xs:documentation>Код управления</xs:documentation>
                    </xs:annotation>
                </xs:element>
                <xs:element
name="departmentShort" type="xs:string" minOccurs="0">
                    <xs:annotation>
                        <xs:documentation>Управление - краткое название</xs:documentation>
                    </xs:annotation>
                </xs:element>
            </xs:sequence>
        </xs:complexType>
        </xs:element>
        <xs:element name="count" type="xs:integer" default="0">
            <xs:annotation>
                <xs:documentation>Количество
проектов</xs:documentation>
            </xs:annotation>
        </xs:element>
    </xs:sequence>

```



```
        </xs:complexType>
    </xs:element>
</xs:schema>
```

Ответ REST-сервиса должен содержать JSON-файл с массивом элементов (каждый элемент формируется по приведенной выше схеме), где:

- responsible/user/photoURL – относительная ссылка на фотографию пользователя;
- responsible/user/login – логин учетной записи пользователя;
- responsible/user/fioFull – полное имя пользователя;
- responsible/user/fioShort – краткое имя пользователя;
- responsible/user/post – должность пользователя;
- responsible/user/department/departmentFull – полное наименование организации пользователя;
- responsible/user/department/departmentCode – код организации пользователя;
- responsible/user/department/departmentShort – краткое наименование организации пользователя;
- responsible/count – количество проектов пользователя.

При завершении задачи выполняются следующие шаги:

1) шаг 1 – проверка на наличие элемента массива, путь к которому указывается на вкладке API, где путь_с_вкладки_API.responsibleRole.code – значение указанное в поле Role Code на вкладке Data:

- если элемент найден, то перейти к шагу 2;
- если элемент не найден, то перейти к шагу 3;

2) шаг 2 – удалить элементы массива, где: где путь_с_вкладки_API.responsibleRole.code – значение указанное в поле Role Code на вкладке «Data»;

3) шаг 3 – создать новый элемент массива, путь к которому указывается на вкладке API в режиме редактирования компонента, где:

- путь_с_вкладки_API.responsibleRole.code – значение указанное в поле Role Code на вкладке Data;
- путь_с_вкладки_API.responsibleRole.name – значение указанное в поле Role Name на вкладке Data;
- путь_с_вкладки_API.login – значение поля responsible/user/login JSON, полученного в ответе REST-сервиса;
- путь_с_вкладки_API.fioFull – значение поля responsible/user/fioFull JSON,

полученного в ответе REST-сервиса;

– путь_с_вкладки_API.fioShort – значение поля responsible/user/fioShort JSON, полученного в ответе REST-сервиса;

– путь_с_вкладки_API.post – значение поля responsible/user/post JSON, полученного в ответе REST-сервиса;

– путь_с_вкладки_API.department.departmentFull – значение поля responsible/user/department/departmentFull JSON, полученного в ответе REST-сервиса;

– путь_с_вкладки_API.department.departmentCode – значение поля responsible/user/department/departmentCode JSON, полученного в ответе REST-сервиса;

– путь_с_вкладки_API.department.departmentShort – значение поля responsible/user/department/departmentShort JSON, полученного в ответе REST-сервиса.

4.6.6.5.9 Comment. Добавление комментария

Компонент предназначен для сохранения комментария пользователя и расширенной информации о нем (Рисунок 112).



Рисунок 112 – Компонент Comment

Для корректной работы компонента Comment хранение данных комментария в JSON документа должно соответствовать структуре, указанной ниже (Рисунок 113).

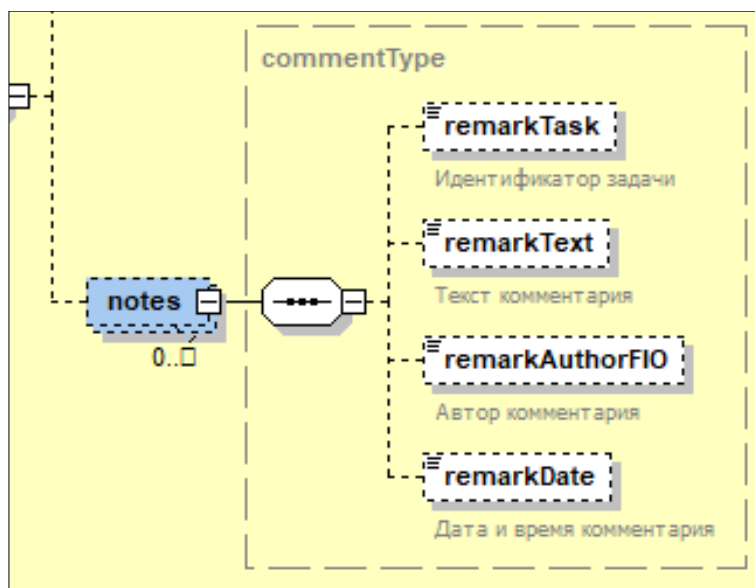


Рисунок 113 – Структура компонента Comment

Специфические поля в настройках компонента отсутствуют.

Для корректной работы компонента в поле «Property Name» на вкладке API должен быть указан путь к массиву, содержащему вложенные элементы.

При завершении задачи в JSON документа будет создан новый элемент массива, путь

к которому указывается на вкладке API в режиме редактирования компонента, где:

- путь_с_вкладки_API.remarkTask – id текущей задачи бизнес-процесса;
- путь_с_вкладки_API.remarkText – комментарий, введенный на форме;
- путь_с_вкладки_API.remarkAuthorFIO – displayName текущего пользователя;
- путь_с_вкладки_API.remarkDate – текущая дата и время.

4.6.6.5.10 Info Field. Отображение ранее сохраненных полей

Компонент предназначен для отображения ранее сохраненных полей JSON в режиме просмотра (Рисунок 114).

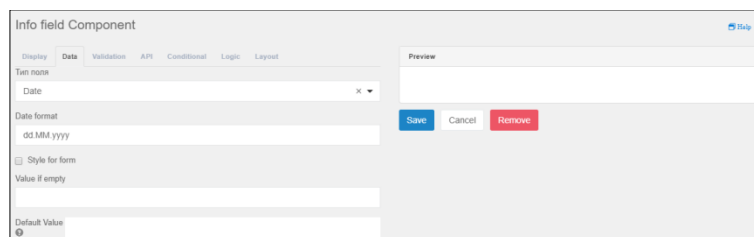


Рисунок 114 – Info Field компонент

На вкладке «Data» добавлено специфическое поле «Тип поля» для работы указанного компонента (Таблица 43).

Таблица 43 – Описание поля «Тип поля»

Наименование поля	Обязательность	Описание
Тип поля	Да	<p>Список с единичным выбором</p> <p>Предназначено для определения формата отображаемых данных</p> <p>Доступен выбор из значений:</p> <p>Link Ссылка</p> <p>Date Дата</p> <p>Boolean Логическое</p> <p>Array Массив</p> <p>File Файл</p> <p>Class inline Класс в строку</p> <p>Text Текст</p> <p>Solr Поиск в Solr</p>

Состав полей вкладки «Data» и правила их заполнения определяются значением, выбранным в специфическом поле «Тип поля».

Чтобы обеспечить возможность добавления нескольких ссылок на форму (когда в качестве пути к элементу указывается корневой элемент схемы) для указания пути к элементу JSON вместо поля «Property Name» на вкладке API предназначено поле «Значение поля (API)» на вкладке «Data». Правила заполнения нового поля аналогичны полю «Property Name». Для всех ранее созданных форм перенастройка компонента не требуется. При этом

важно обеспечить, чтобы в поле «Property Name» на вкладке API всегда указывалось уникальное значение.

4.6.6.5.10.1 Link Ссылка

При выбранном типе поля «Link» (Рисунок 115) на вкладке «Data» отображаются дополнительно специфические поля (Таблица 44).

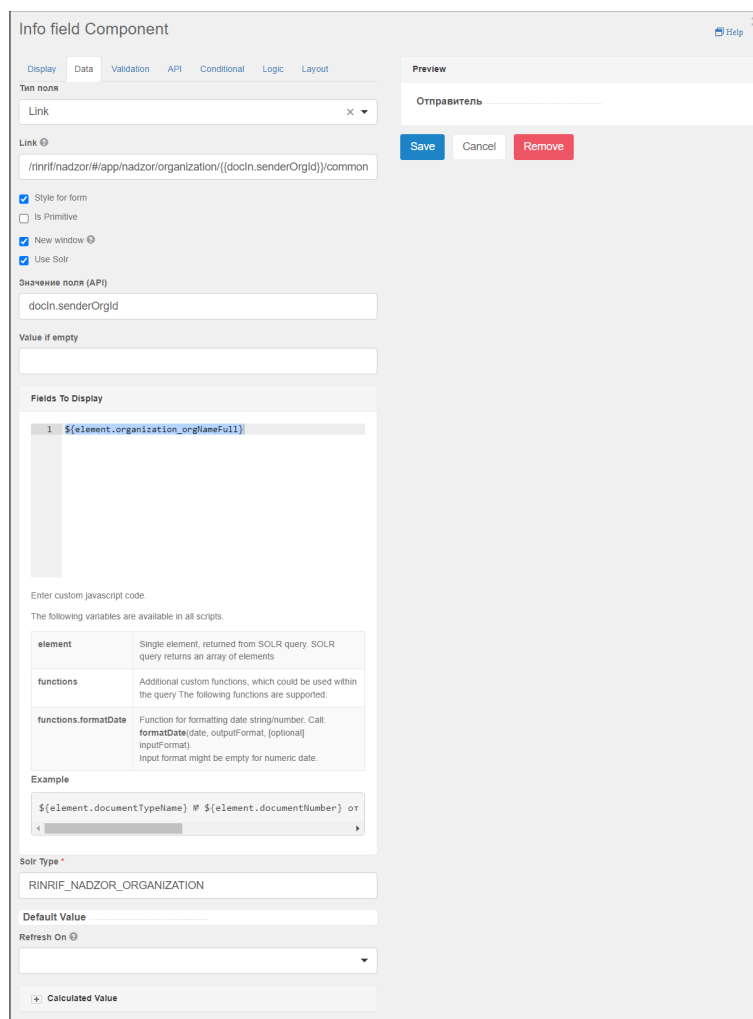


Рисунок 115 – Link ссылка

Таблица 44 – Специфические поля вкладки «Data»

Наименование поля	Обязательность	Описание
Link	Да	Поле ввода Предназначено для указания относительной ссылки Может включать параметр из JSON документа Указанный параметр должен содержать JsonPath элемента
New window	Нет	Чекбокс При установленном флаге ссылка открывается в новом окне/вкладке (в зависимости от настроек браузера пользователя)
Use Solr	Нет	Чекбокс При установленном флаге отображается набор

Наименование поля	Обязательность	Описание
		полей для реализации поиска идентификатора, указанного в поле "Значение поля (API)" среди документов Solr
Fields To Display	–	Отображается и обязательно для заполнения только при установленном флаге Use Solr Предназначен для указания поля или набора полей, которые будут отображаться в качестве значения поля вместо идентификатора связанного документа
Solr Type	–	Отображается и обязательно для заполнения только при установленном флаге Use Solr Предназначен для указания типа документа, среди которых будет осуществлен поиск документа, идентификатор которого указан в поле "Значение поля (API)"

Настройка текста для отображаемой ссылки осуществляется на вкладке «API».

Описанная ниже настройка актуальна, если не установлен флаг «Use Solr», в противном случае текст определяется настройками свойств Solr, описанными выше (Рисунок 116).

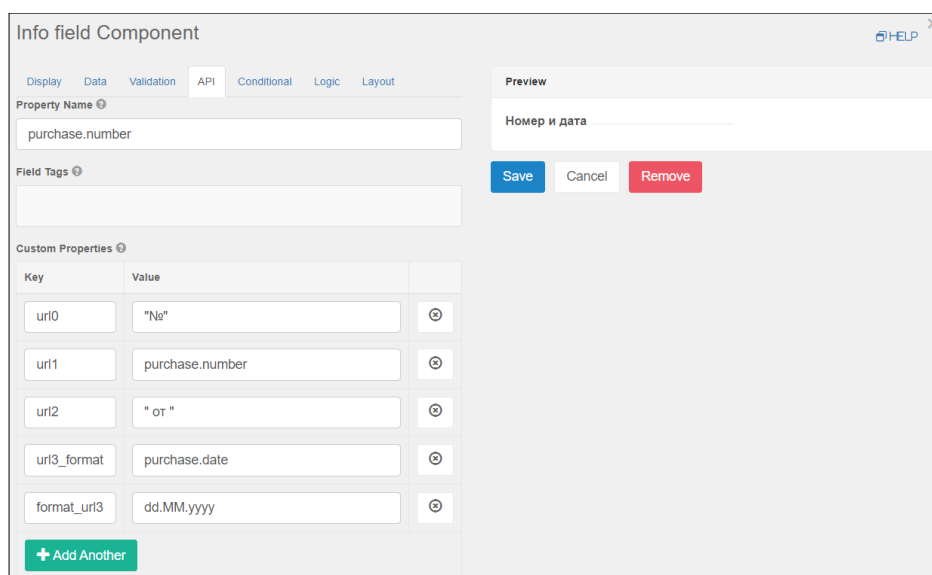


Рисунок 116 – Настройка компонента на вкладке «API»

В группе полей «Custom Properties» необходимо указать правила формирования ссылки (Таблица 45).

Таблица 45 – Правила формирования ссылки

Key	Value
urlN, где N – порядковый номер, начиная с 0	Путь до элемента JSON или константа указанная в двойных кавычках ("от")

Key	Value
urlN_format, где N – порядковый номер, начиная с 0	Путь до элемента JSON, который должен быть преобразован по определенному правилу
format_urlN, где N – порядковый номер, начиная с 0	Правила преобразования элемента с аналогичным порядковым номером

4.6.6.5.10.2 Date Дата

При выбранном типе поля «Date» (Дата) на вкладке «Data» отображаются дополнительно специфические поля (Рисунок 117, Таблица 46).

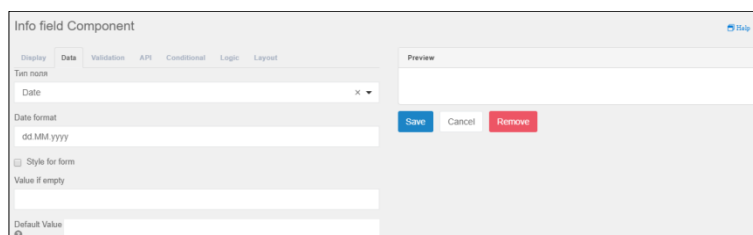


Рисунок 117 – Тип поля Date

Таблица 46 – Специфические поля вкладки «Data»

Наименование поля	Обязательность	Описание
Date format	Нет	Поле ввода Предназначено для ввода формата отображения даты
Style for form	Нет	Чекбокс При установленном флаге лейблу присваивается стиль заголовка При установленном флаге лейбл поля на вкладке Display должен быть скрыт
Value if empty	Нет	Поле ввода Предназначено для ввода константы, которая будет отображаться при отсутствии в JSON документа значения поля, указанного на вкладке API

4.6.6.5.10.3 Boolean Логическое

При выбранном типе поля «Boolean» (Логическое) (Рисунок 118) на вкладке «Data» отображаются дополнительно специфические поля (Таблица 47).

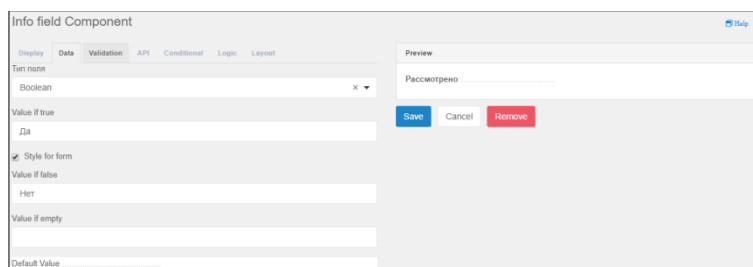


Рисунок 118 – Тип поля Boolean

Таблица 47 – Специфические поля вкладки «Data»

Наименование поля	Обязательность	Описание
Value if true	Нет	Поле ввода Предназначено для ввода значения, которое отображается вместо true, полученного в поле, указанном на вкладке API
Value if false	Нет	Поле ввода Предназначено для ввода значения, которое отображается вместо false, полученного в поле, указанном на вкладке API
Style for form	Нет	Чекбокс При установленном флаге лейблу присваивается стиль заголовка При установленном флаге лейбл поля на вкладке Display должен быть скрыт
Value if empty	Нет	Поле ввода Предназначено для ввода константы, которая будет отображаться при отсутствии в JSON документа значения поля, указанного на вкладке API

4.6.6.5.10.4 Array Массив

При выбранном типе поля «Array» (Массив) (Рисунок 119) на вкладке «Data» отображаются дополнительно специфические поля (Таблица 48).

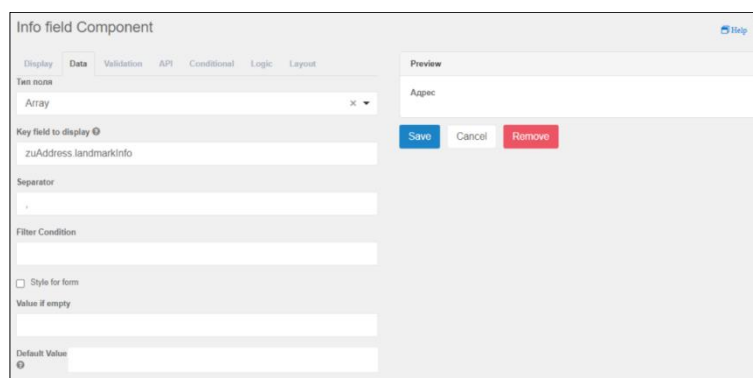
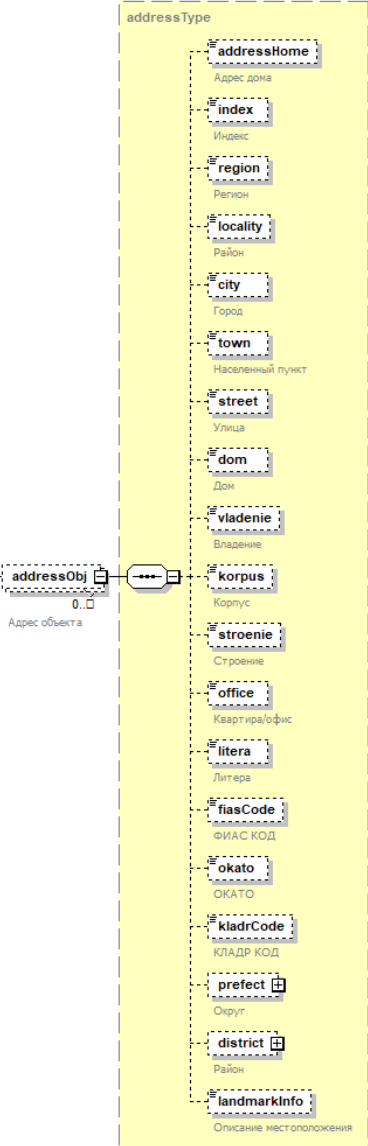


Рисунок 119 – Тип поля Array

Таблица 48 – Специфические поля вкладки «Data»

Наименование поля	Обязательность	Описание
Is Primitive	Нет	Чекбокс Устанавливается при выводе данных из простого массива (не содержащего в себе других элементов)

Наименование поля	Обязательность	Описание
		 <p>0..□ Наименование работ, подлежащих проверке</p>
Key field to display	Только с неустановленным флагом Is Primitive	<p>Поле ввода Предназначено для указания поля для отображения в рамках узла, указанного в поле "Значение поля (API)" на вкладке Data для сложных массивов (содержащих в себе другие элементы)</p>  <p>addressType</p> <ul style="list-style-type: none"> addressHome: Адрес дома index: Индекс region: Регион locality: Район city: Город town: Населенный пункт street: Улица dom: Дом vladenie: Владение korpus: Корпус stroenie: Строение office: Квартира/офис litera: Литера fiasCode: ФИАС КОД okato: ОКАТО kladrCode: КЛАДР КОД prefect: Округ district: Район landmarkInfo: Описание местоположения <p>addressObj: Адрес объекта</p>
Separator	Да	Поле ввода Предназначено для указания символа-разделителя для элементов массива (для переноса на другую строку использовать разделитель)
Style for form	Нет	Чекбокс При установленном флаге лейблу присваивается

Наименование поля	Обязательность	Описание
		стиль заголовка При установленном флаге лейбл поля на вкладке Display должен быть скрыт
Filter Condition	Нет	Поле ввода Предназначено для вывода конкретного элемента массива (например, v.responsibleRole.code == "responsibleExecutor")
Value if empty	Нет	Поле ввода Предназначено для ввода константы, которая будет отображаться при отсутствии в JSON документа значения поля, указанного на вкладке API

4.6.6.5.10.5 File Файл

При выбранном типе поля «File» (Файл) вкладке «Data» отображаются дополнительно специфические поля (Таблица 49).

Таблица 49 – Специфические поля вкладки «Data» поля «File»

Наименование поля	Обязательность	Описание
Style for form	Нет	Чекбокс При установленном флаге лейблу присваивается стиль заголовка При установленном флаге лейбл поля на вкладке Display должен быть скрыт
Value if empty	Нет	Поле ввода Предназначено для ввода константы, которая будет отображаться при отсутствии в JSON документа значения поля, указанного на вкладке API
File Type	Да	Поле ввода Предназначено для ввода типа файла, с которым он был сохранен в файловое хранилище
Only ID	Нет	Чекбокс При установленном на вкладке Data флаге Only ID в поле "Значение поля (API)" на вкладке Data необходимо указывать путь до поля с массива, предназначенного для хранения идентификатора сохраненного файла, при неустановленном - путь до узла, содержащего расширенную информацию о файле (см. описание компонента File Manager)
Show Sign Button	Нет	Чекбокс При установленном флаге отображается ссылка для подписания файла электронной подписью

Sign this file	Нет	Чекбокс При установленном флаге при завершении задачи по кнопке Complete Button будет осуществлено подписание выбранного файла
----------------	-----	---

4.6.6.5.10.6 Text Текст

При выбранном типе поля «Text» (Текст) (Рисунок 120) на вкладке «Data» отображаются дополнительно специфические поля (Таблица 50).

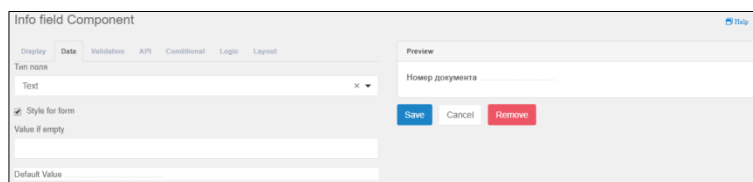


Рисунок 120 – Тип поля Text

Таблица 50 – Специфические поля вкладки «Data» поля «Text»

Наименование поля	Обязательность	Описание
Style for form	Нет	Чекбокс При установленном флаге лейблу присваивается стиль заголовка При установленном флаге лейбл поля на вкладке Display должен быть скрыт
Value if empty	Нет	Поле ввода Предназначено для ввода константы, которая будет отображаться при отсутствии в JSON документа значения поля, указанного на вкладке API

Для использования конкатенации нескольких полей в группе полей «Custom Properties» необходимо указать правила формирования конкатенации в соответствии со следующими условиями.

В поле «Property Name» должен быть указан корневой элемент JSON документа (Таблица 51).

Таблица 51 – Корневые элементы JSON документа

Key	Value
textN, где N – порядковый номер, начиная с 0	Путь до элемента JSON (<i>pmt.typePmt.name</i>) или константа указанная в кавычках (" <i>om</i> ")
textN_format, где N – порядковый номер, начиная с 0	Путь до элемента JSON, который должен быть преобразован по определенному правилу (<i>pmt.regDate</i>)
format_textN, где N – порядковый номер, начиная с 0	Правила преобразования элемента с аналогичным порядковым номером (<i>dd.MM.yyyy</i>)

4.6.6.5.10.7 Solr Поиск в Solr

Тип поля «Solr» предназначен для поиска в Solr сохраненного идентификатора связанного документа и отображения атрибутов найденного документа.

При выбранном типе поля «Solr» (Рисунок 121) на вкладке «Data» отображаются дополнительно специфические поля (Таблица 52).

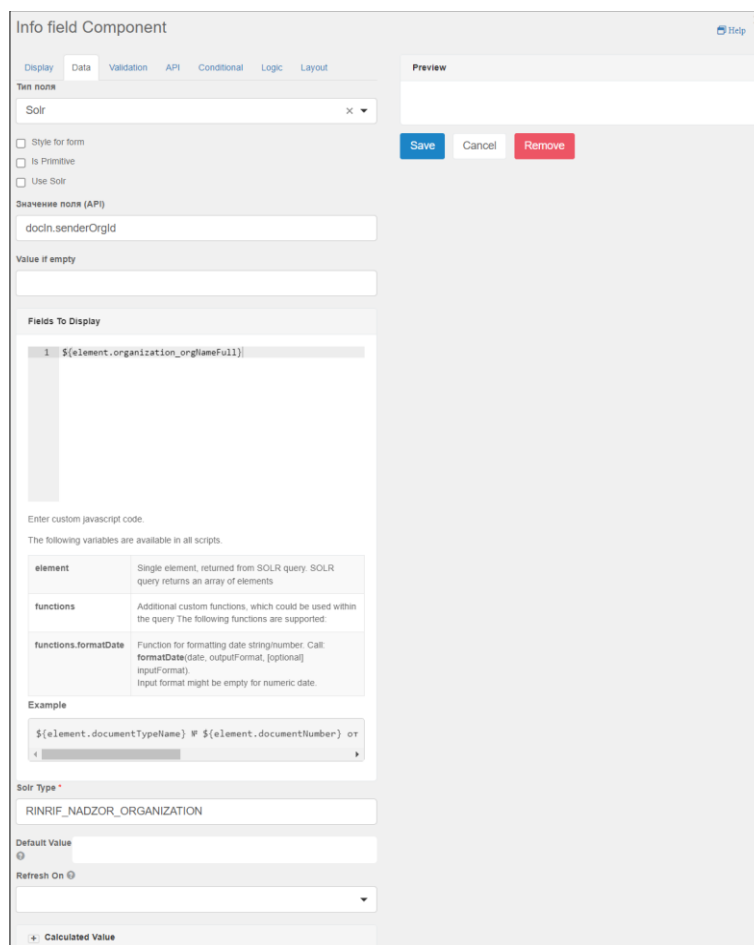


Рисунок 121 – Тип поля Solr

Таблица 52 – Специфические поля вкладки «Data» поля «Solr»

Наименование поля	Обязательность	Описание
Fields To Display	Нет	Отображается и обязательно для заполнения только при установленном флаге Use Solr. Предназначен для указания поля или набора полей, которые будут отображаться в качестве значения поля вместо идентификатора связанного документа.
Solr Type	Да	Отображается и обязательно для заполнения только при установленном флаге Use Solr. Предназначен для указания типа документа, среди которых будет осуществлен поиск документа, идентификатор которого указан в поле "Значение поля (API)".

4.6.6.5.11 Approval List View. Просмотр листа согласования

Компонент предназначен для просмотра листа согласования (Рисунок 122).

№ пп	ФИО и должность	Срок согласования	Результат согласования
▶ 1	Иванов Иван Иванович Специалист	12.12.2019	<input type="button" value="Согласовано"/> 11.12.2019 00:00
▶ 2	Иванов Иван Иванович Специалист		<input type="button" value="Утверждено"/>
▶ 1	Иванов Иван Иванович Специалист		<input type="button" value="Согласовано"/>
▶ 1	Петров Константин Романович		<input type="button" value="Согласовано"/>
▶ 5	Петров Константин Романович		<input type="button" value="Утверждено"/>
▶ Предыдущий цикл согласования			
▶ Предыдущий цикл согласования			
▶ Предыдущий цикл согласования			

Рисунок 122 – Просмотр листа согласования

Для корректной работы компонента Approval List View структура данных для листа согласования должна соответствовать одной из структур, требования к которым указаны в п. 4.6.6.5.8.

На вкладках «Display» и «Data» добавлены специфические поля для работы данного компонента (Рисунок 123, Таблица 53).

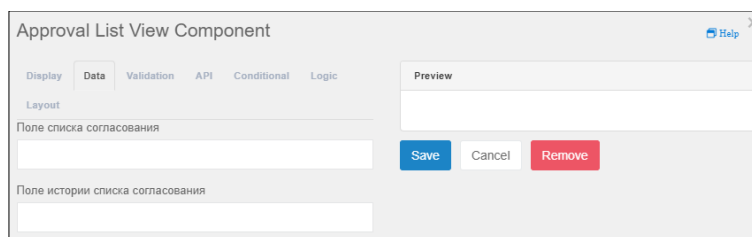


Рисунок 123 – Специфические поля вкладки «Data»

Таблица 53 – Специфические поля вкладок «Display» и «Data»

Наименование поля	Обязательность	Описание
Expand Current Cycle	Нет	Флаг Отображать текущий цикла согласования сразу развернутым
Поле списка согласования	Да	Поле ввода Предназначено для указания пути к узлу JSON документа с комплексным типом approvalCycleType, содержащем сведения о текущем цикле согласования
Поле истории списка согласования	Да	Поле ввода Предназначено для указания пути к узлу JSON документа с комплексным типом approvalCycleType, содержащем сведения о прошедших циклах согласования (массив)
Only ID in JSON	Нет	Флаг Устанавливается при использовании сокращенной структуры хранения файла

Заполнение поля «Property Name» на вкладке «API» не требуется.

4.6.6.5.12 Approval List Edit. Предзаполнение листа согласования

Компонент предназначен для формирования предзаполненного листа согласования по документу (Рисунок 124).

	№ пп	Вид согласования	Согласующий
- ↓	1	Согласование	Иванов Иван Иванович Эксперт
- ↑ ↓	2	Согласование	Петров Петр Петрович Эксперт
- ↑	3	Вид согласования	Сидоров Иван Иванович Эксперт
+			

Рисунок 124 – Предзаполненный лист согласования по документу

Для корректной работы компонента Approval List View структура данных для листа согласования должна соответствовать одной из структур, требования к которым указаны в п. 4.6.6.5.8.

На вкладке «Data» добавлены специфические поля для работы данного компонента (Рисунок 125, Таблица 54).

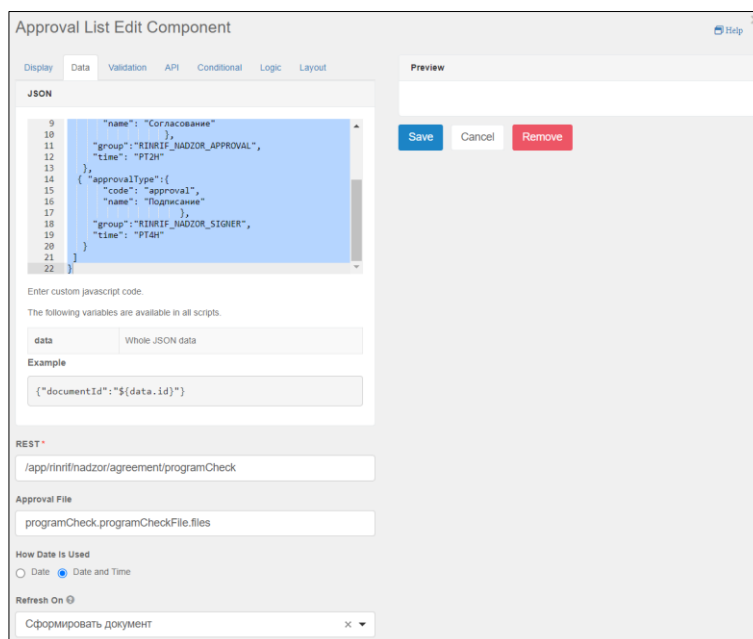


Рисунок 125 – Специфические поля вкладки «Data»

Таблица 54 – Специфические поля вкладки «Data»

Наименование поля	Обязательность	Описание
JSON	Да	Расширяемый текстовый блок Предназначено для указания JSON-запроса к REST-сервису В качестве переменной к запросу может передаваться параметр из JSON документа {"documentId": "\${data.id}"}
REST	Да	Поле ввода Предназначено для указания относительной ссылки на

Наименование поля	Обязательность	Описание
		REST-сервис, возвращающий сведения для предзаполнения листа согласования Требования к сервису описаны ниже
Approval File	Нет	Путь к файлу, который должен быть скопирован (создается дубликат) в цикл согласования Структура хранения копируемого файла должна быть идентичной структуре хранения данных поля approvalCycleFile в листе согласования Либо оба хранятся в расширенной структуре, либо оба хранятся в сокращенной
How Date Is Used	Да	Тип поля, указанный в XSD схеме для approvalCycleDate:Date - если xs:dateDate and Time - если xs:dateTime

Требования к REST-сервису (схема «approval»), возвращающему сведения для предзаполнения листа согласования:

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XMLSpy v2011 rel. 2 (http://www.altova.com) by TeaM DJiNN (TeaM DJiNN) -->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
  <xs:element name="approval">
    <xs:annotation>
      <xs:documentation>Лист согласования</xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:sequence>
        <xs:element name="move" type="xs:boolean">
          <xs:annotation>
            <xs:documentation>Можно менять порядок
элементов списка?</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="mainApproval" minOccurs="0"
maxOccurs="unbounded">
          <xs:annotation>
            <xs:documentation>Список основных
согласующих</xs:documentation>
          </xs:annotation>
          <xs:complexType>
            <xs:sequence>
              <xs:element name="type" type="nsiType"
minOccurs="0" maxOccurs="unbounded">
                <xs:annotation>
                  <xs:documentation>Вид
согласования</xs:documentation>
                </xs:annotation>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

```

minOccurs="0">
    <xs:element name="login" type="xs:string"
        <xs:annotation>
            <xs:documentation>Логин
согласующего</xs:documentation>
        </xs:annotation>
    </xs:element>
    <xs:element name="fioFull" type="xs:string"
        <xs:annotation>
            <xs:documentation>ФИО
согласующего</xs:documentation>
        </xs:annotation>
    </xs:element>
    <xs:element name="post" type="xs:string"
        <xs:annotation>
            <xs:documentation>Должность согласующего</xs:documentation>
        </xs:annotation>
    </xs:element>
    <xs:element name="delete"
        <xs:annotation>
            <xs:documentation>Может
быть удален из списка?</xs:documentation>
        </xs:annotation>
    </xs:element>
    <xs:element name="time" type="xs:string"
        <xs:annotation>
            <xs:documentation>Продолжительность согласования</xs:documentation>
        </xs:annotation>
    </xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="addApproval" minOccurs="0"
maxOccurs="unbounded">
    <xs:annotation>
        <xs:documentation>Список дополнительных
согласующих</xs:documentation>
    </xs:annotation>
    <xs:complexType>
        <xs:sequence>
            <xs:element name="type" type="nsiType"
minOccurs="0" maxOccurs="unbounded">
                <xs:annotation>
                    <xs:documentation>Вид
согласования</xs:documentation>
                </xs:annotation>
            </xs:element>

```

```

minOccurs="0">
    <xs:element name="login" type="xs:string"
        <xs:annotation>
            <xs:documentation>Логин
согласующего</xs:documentation>
        </xs:annotation>
    </xs:element>
    <xs:element name="fioFull" type="xs:string"
        <xs:annotation>
            <xs:documentation>ФИО
согласующего</xs:documentation>
        </xs:annotation>
    </xs:element>
    <xs:element name="post" type="xs:string"
        <xs:annotation>
            <xs:documentation>Должность согласующего</xs:documentation>
        </xs:annotation>
    </xs:element>
    <xs:element name="time" type="xs:string"
        <xs:annotation>
            <xs:documentation>Продолжительность согласования</xs:documentation>
        </xs:annotation>
    </xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:complexType name="nsiType">
    <xs:annotation>
        <xs:documentation>Справочное значение</xs:documentation>
    </xs:annotation>
    <xs:sequence>
        <xs:element name="code" type="xs:string" minOccurs="0">
            <xs:annotation>
                <xs:documentation>Код</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element name="name" type="xs:string" minOccurs="0">
            <xs:annotation>
                <xs:documentation>Наименование</xs:documentation>
            </xs:annotation>
        </xs:element>
    </xs:sequence>
</xs:complexType>
</xs:schema>

```

Структура данных для листа согласования приведена ниже (Рисунок 126,

Таблица 55).

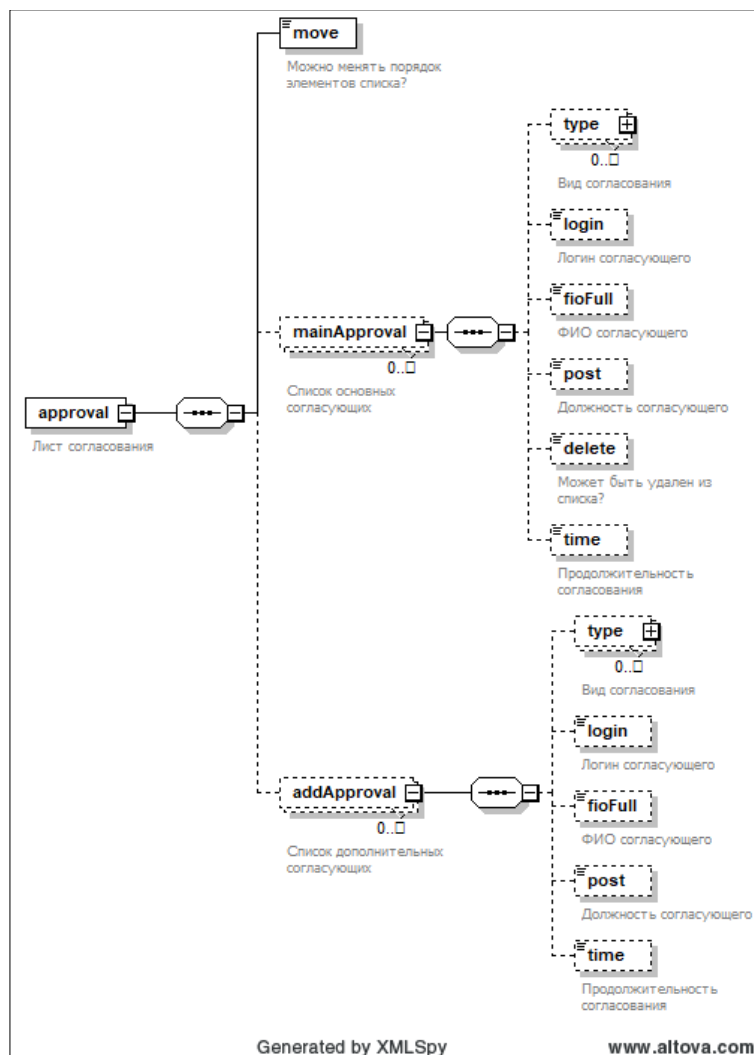


Рисунок 126 – Структура данных для листа согласования

Ответ должен представлять собой JSON документ по схеме «approval».

Таблица 55 – Описание структуры данных для листа согласования

Путь	Описание
approval/move	Если = true, то стрелки для изменения порядка записей в таблице отображаются, если = false, то нет
approval/mainApproval	Определяет наличие в таблице предзаполненных строк 1 элемент массива = 1 строке таблицы Номер для столбца № pp определяется порядком элементов массива
approval/mainApproval/type/name	Отображается в столбце "Вид согласования", при наличии >1 элемента массива approval/mainApproval/type отображать значение в столбце "Вид согласования" в виде списка с единичным выбором, в противном случае в виде константы
approval/mainApproval/fioFull	Отображается в столбце "Согласующий", всегда константа
approval/mainApproval/post	Отображается в столбце "Согласующий" под ФИО, всегда

Путь	Описание
	константа
approval/mainApproval/delete	Если = true, то отображается минус для удаления пользователя из списка согласующих
approval/addApproval	Определяет наличие в таблице плюса для добавления дополнительных согласующих Номер для столбца № пп определяется как порядковый
approval/addApproval/type/name	Отображается в столбце "Вид согласования", при наличии >1 элемента массива approval/addApproval/type отображать значение в столбце "Вид согласования" в виде списка с единичным выбором, в противном случае в виде константы
approval/addApproval/fioFull	Отображается в виде списка с единичным выбором, наполняется значениями всех элементов approval/addApproval/fioFull
approval/addApproval/post	Отображается в столбце "Согласующий" под ФИО после выбора значения в списке, всегда константа

Пример листа согласования, состоящего только из основных согласующих без возможности изменения (Рисунок 127).

Лист согласования		
№ пп	Вид согласования	Согласующий
1	Согласование	Маркова Евгения Александровна Начальник управления бюджетных инвестиций и государственных программ
2	Согласование	Дубяшевский Анатолий Викторович Первый заместитель председателя комитета
3	Утверждение	Афонин Александр Викторович Председатель комитета

Рисунок 127 – Лист согласования, состоящий только из основных согласующих без возможности изменения

Код листа согласования, состоящего только из основных согласующих без возможности изменения, имеет вид:

```
{
  "approval": {
    "move": false,
    "mainApproval": [
      {
        "type": [
          {
            "code": "agreed",
            "name": "Согласование"
          }
        ]
      },
    ],
    "login": "markova_ea",
    "fioFull": "Маркова Евгения Александровна",
    "post": "Начальник управления бюджетных инвестиций и государственных программ",
    "delete": false,
    "time": "PT8H"
  },
}
```

```

{
  "type": [
    {
      "code": "agreed",
      "name": "Согласование"
    }
  ],
  "login": "dubashevskiy_av",
  "fioFull": "Дубашевский Анатолий Викторович",
  "post": "Первый заместитель председателя комитета",
  "delete": false,
  "time": "PT8H"
},
{
  "type": [
    {
      "code": "signer",
      "name": "Утверждение"
    }
  ],
  "login": "afonin_av",
  "fioFull": "Афонин Александр Викторович",
  "post": "Председатель комитета",
  "delete": false,
  "time": "PT8H"
}
]
}
}

```

Пример листа согласования, состоящего только из дополнительных согласующих (Рисунок 128).

Лист согласования			
	№ пп	Вид согласования	Согласующий
+			

Рисунок 128 – Лист согласования, состоящий только из дополнительных согласующих я

Код листа согласования, состоящего только из дополнительных согласующих, имеет вид:

```

{
  "approval": {
    "move": true,
    "addApproval": [
      {
        "type": [
          {
            "code": "agreed",
            "name": "Согласование"
          }
        ],
        "login": "afonin_av",
        "fioFull": "Афонин Александр Викторович",
        "post": "Председатель комитета",
        "time": "PT8H"
      }
    ]
  }
}

```

```

},
{
  "type": [
    {
      "code": "agreed",
      "name": "Согласование"
    }
  ],
  "login": "ivanov_ki",
  "fioFull": "Иванов Иван Иванович",
  "post": "Специалист 3 категории",
  "time": "PT8H"
},
{
  "type": [
    {
      "code": "agreed",
      "name": "Согласование"
    }
  ],
  "login": "petrov_pp",
  "fioFull": "Петров Петр Петрович",
  "post": "Специалист",
  "time": "PT8H"
}
]
}
}

```

На вкладке «Validation» при установленном признаке «Required» осуществляется проверка на заполненность полей добавленных строк, а именно:

- при наличии альтернативы в выборе вида согласования проверяется заполненность всех строк столбца «Вид согласования»;
- при наличии альтернативы в выборе согласующего проверяется заполненность всех строк столбца «Согласующий».

После завершения задачи в JSON документа создаются новые элементы массива в узле, путь к которому указывается на вкладке API в режиме редактирования компонента, где:

- путь_с_вкладки_API.approvalCycleNum – количество элементов массива путь_с_вкладки_API.approvalHistory.approvalCycle + 1;
- путь_с_вкладки_API.approvalCycleDate – текущая дата;
- путь_с_вкладки_API.approvalCycleFile – скопированное значение из поля, путь к которому указан в поле Approval File на вкладке Data;
- путь_с_вкладки_API.agreed.approvalNum – значение в столбце «№ пп» для данного согласующего;

- путь_с_вкладки_API.agreed.approvalTime – значение в поле approval/mainApproval/time или approval/addApproval/time для данного согласующего;
- путь_с_вкладки_API.agreed.approvalType – значение выбранное или установленное по умолчанию в столбце «Вид согласования»;
- путь_с_вкладки_API.agreed.approvalTypeCode – соответствует путь_с_вкладки_API.agreed.approvalType значению approval/mainApproval/type/code или approval/addApproval/type/code;
- путь_с_вкладки_API.agreed.agreedBy.post – approval/mainApproval/post или approval/addApproval/post для данного согласующего;
- путь_с_вкладки_API.agreed.agreedBy.fioFull – approval/mainApproval/fioFull или approval/addApproval/fioFull для данного согласующего;
- путь_с_вкладки_API.agreed.agreedBy.accountName – approval/mainApproval/login или approval/addApproval/login для данного согласующего.

4.6.6.5.13 Save Cur User. Сохранение данных текущего пользователя

Компонент предназначен для автоматического заполнения поля данными текущего пользователя из ответа метода /auth/user.

На вкладке «Data» добавлено специфическое поле «User Field» для работы данного компонента (Таблица 56, Рисунок 129).

Таблица 56 – Специфическое поле для работы данного компонента

Наименование поля	Обязательность	Описание
User Field	Да	Поле ввода Предназначено для указания атрибута ответа сервиса /auth/user <pre>{ "accountName": "string", "authenticated": true, "departmentCode": "string", "departmentFullName": "string", "displayName": "string", "distinguishedName": "string", "fioShort": "string", "iofShort": "string", "lock": true, "lockCode": "string", "mail": "string", "name": "string", "otherPhone": "string", "ou": "string", "permissions": ["string"], }</pre>

Наименование поля	Обязательность	Описание
		<pre>"post": "string", "principal": { "name": "string" }, "roomNumber": "string", "snils": "string", "snilsDate": "2021-01-12T06:48:30.537Z", "telephoneNumber": "string", "userSessionsNumber": 0 }</pre>

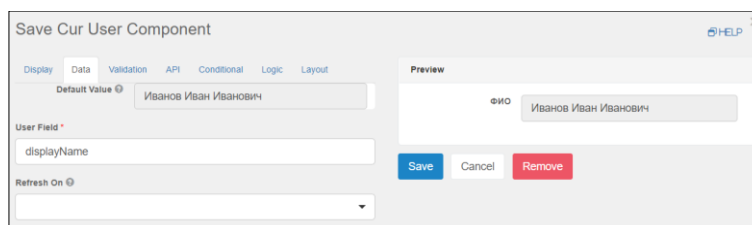


Рисунок 129 – Специфическое поле «User Field»

Для каждого сохраняемого атрибута (ФИО, логин, должность и так далее) должен быть добавлен отдельный компонент.

4.6.6.5.14 Approval Component. Согласование по листу согласования

Компонент предназначен для осуществления согласования по листу согласования (Рисунок 130).

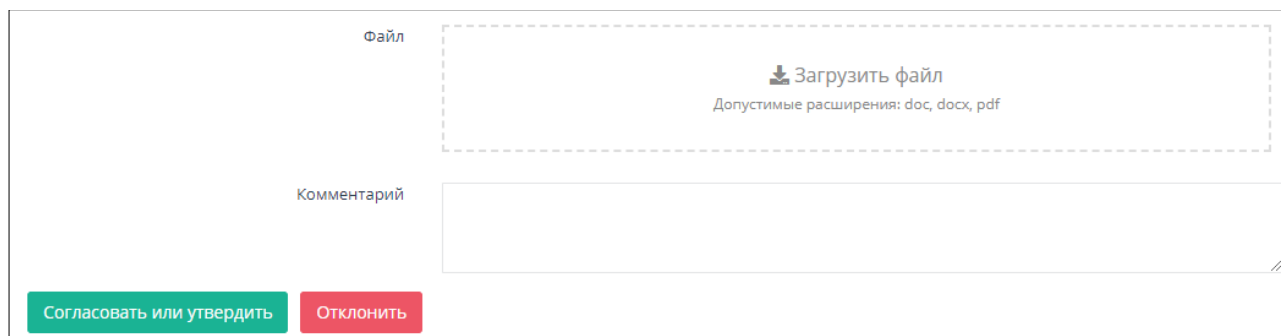


Рисунок 130 – Компонент согласования по листу согласования

Для корректной работы компонента значения для согласования JSON документа должен содержать узел, содержащий узлы approvalCycle (единичный) и approvalCycleHistory (массив).

Узлы approvalCycle и approvalCycleHistory должны соответствовать одной из структур, указанных в п.4.6.6.5.8.

В режиме редактирования компонент отображается как: «Ошибка при отрисовке formio-cdr-approval-comp-by-approval-list». При настройке необходимо ориентироваться на отображение в режиме предпросмотра с корректным тестовым документом. Корректный

тестовый документ – это документ, который содержит узлы для записи результатов согласования, но при этом не содержат результатов согласования.

Пример кода корректного тестового документа:

```
{
  "document": {
    "approvalCycle": {
      "approvalCycleNum": "3",
      "approvalCycleDate": "2022-08-11",
      "approvalCycleFile": [
        "3b3416a8-c6f8-4039-80ac-621053284ee3"
      ],
      "agreed": [
        {
          "approvalNum": "1",
          "approvalTime": "PT8H",
          "agreedBy": {
            "post": "Специалист",
            "fioFull": "Иванов Иван Иванович",
            "accountName": "ivanov_ii"
          },
          "approvalType": "Согласование",
          "approvalTypeCode": "agreed"
        },
        {
          "approvalNum": "2",
          "approvalTime": "PT8H",
          "agreedBy": {
            "post": "Специалист",
            "fioFull": "Петров Петр Петрович",
            "accountName": "petrov_pp"
          },
          "approvalType": "Подписание",
          "approvalTypeCode": "approval"
        }
      ]
    },
    "approvalCycleHistory": [
      {
        "approvalCycleNum": "1",
        "approvalCycleDate": "2022-08-11",
        "approvalCycleFile": [
          "e08692a3-40e0-43b8-8d1e-53ce37397c7d"
        ],
        "agreed": [
          {
            "approvalNum": "1",
            "approvalTime": "PT8H",
            "agreedBy": {
              "post": "Специалист",
              "fioFull": "Иванов Иван Иванович",
              "accountName": "ivanov_ii"
            }
          }
        ]
      }
    ]
  }
}
```

```

    "approvalFactDate": "2022-08-11T13:26:31.285",
    "approvalResult": "Не согласовано",
    "approvalType": "Согласование",
    "approvalTypeCode": "agreed"
  },
  {
    "approvalNum": "2",
    "approvalTime": "PT8H",
    "agreedBy": {
      "post": "Специалист",
      "fioFull": "Петров Петр Петрович",
      "accountName": "petrov_pp"
    },
    "approvalType": "Подписание",
    "approvalTypeCode": "approval"
  }
]
},
{
  "approvalCycleNum": "2",
  "approvalCycleDate": "2022-08-11",
  "approvalCycleFile": [
    "9a555e35-04b0-4c77-8520-42ee77679708"
  ],
  "agreed": [
    {
      "approvalNum": "1",
      "approvalTime": "PT8H",
      "agreedBy": {
        "post": "Специалист",
        "fioFull": "Иванов Иван Иванович",
        "accountName": "ivanov_ii"
      },
      "approvalFactDate": "2022-08-11T13:27:12.221",
      "approvalResult": "Не согласовано",
      "approvalType": "Согласование",
      "approvalTypeCode": "agreed"
    },
    {
      "approvalNum": "2",
      "approvalTime": "PT8H",
      "agreedBy": {
        "post": "Специалист",
        "fioFull": "Петров Петр Петрович",
        "accountName": "petrov_pp"
      },
      "approvalType": "Подписание",
      "approvalTypeCode": "approval"
    }
  ]
}
}
}
}
}

```


На вкладке «Display» добавлены специфические поля для работы данного компонента (Таблица 57).

Таблица 57 – Специфические поля вкладки «Display» для работы данного компонента

Наименование поля	Обязательность	Описание
No Comment	Нет	Поле ввода Предназначено для скрытия поля для указания комментария
No File	Нет	Поле ввода Предназначено для скрытия поля для прикладывания файла

На вкладке «Data» добавлены специфические поля для работы данного компонента (Таблица 58).

Таблица 58 – Специфические поля вкладки «Data» для работы данного компонента

Наименование поля	Обязательность	Описание
Dictionary Name	Да	Список с единичным выбором Предназначено для выбора справочника, содержащего информацию типах и результатах согласований
Link Code	Да	Поле ввода Предназначено для указания кода поля справочника, содержащего код типа согласования
True Code	Да	Поле ввода Предназначено для указания кода поля справочника, содержащего значение для положительного результата согласования
False Code	Да	Поле ввода Предназначено для указания кода поля справочника, содержащего значение для отрицательного результата согласования

Пример структуры и содержимого справочника для поля «Dictionary Name» (Рисунок 131, Рисунок 132):

- Link Code = code;
- True Code = true;
- False Code = false.

Это значит, что для согласования с типом «agreed» (Согласование) в случае положительного результата будет записано значение «Согласовано», а в случае отрицательного – «Не согласовано».

Атрибуты справочника									
Ник *	Наименование *	Тип данных *	Ссылка	Обязательный	Уникальный	Множественный	Удален	Комментарий	
code	Код	string		Да	Да	Нет	Нет		
name	Наименован	string		Да	Нет	Нет	Нет		
true	Положители	string		Нет	Нет	Нет	Нет		
false	Отрицатель	string		Нет	Нет	Нет	Нет		

Рисунок 131 – Пример структуры справочника для поля «Dictionary Name»







Код ▲	Наименование ▼	Положительный результат ▼	Отрицательный результат ▼	Статус ▼	
Код	Наименование	Положительный резуль'	Отрицательный резуль'	Используется	
agreed	Согласование	Согласовано	Не согласовано	Используется	 
approve	Утверждение	Утверждено	Не утверждено	Используется	 
assent	Одобрение	Одобрено	Не одобрено	Используется	 

Рисунок 132 – Пример содержимого справочника для поля «Dictionary Name»

В поле «Property Name» на вкладке «API» должен быть указан путь до узла, содержащего внутри себя узлы approvalCycle и approvalCycleHistory.

Для работы компонента добавлены специфические вкладки:

- Text Area - Display;
- Text Area - Data;
- Text Area - Validation;
- File Manager - Display;
- File Manager - Data;
- File Manager - Validation;
- Approve Button - Display;
- Approve Button - Data;
- Reject Button - Display;
- Reject Button – Data.

Text Area – для настройки поля с комментарием, File Manager – для настройки поля для прикладывания файла, Approve Button – для настройки кнопки с положительным результатом согласования, Reject Button – для настройки кнопки с отрицательным результатом согласования.

Вкладки «Approve Button – Display» и «Approve Button – Data» предназначены для настройки внешнего вида и поведения по нажатию на кнопку успешного согласования (утверждения) файла.

Правила настройки данных на вкладках «Approve Button – Display» и «Approve Button – Data» аналогичны настройкам для компонента Complete Button.

При завершении задачи нажатием кнопки «Approve Button» сохраняются следующие настройки:

- путь_с_вкладки_API_компонента.approvalCycle.agreed.approvalFactDate – текущие дата и время;
- путь_с_вкладки_API_компонента.approvalCycle.agreed.approvalResult – значение поля, код которого указан в поле «True Code» на вкладке «Data» справочника, выбранного в поле «Dictionary Name», где значение в поле, код которого указан в Link Code, равно: путь_с_вкладки_API_компонента.approvalCycle.agreed.approvalTypeCode, того элемента массива, куда осуществляется сохранение.

Вкладки «Reject Button – Display» и «Reject Button – Data» предназначены для настройки внешнего вида и поведения при нажатии кнопки отклонения согласования (утверждения) файла.

Правила настройки данных на вкладках «Reject Button – Display» и «Reject Button – Data» аналогичны настройкам для компонента Complete Button.

После завершения задачи нажатием кнопки «Reject Button» сохраняются следующие настройки:

- путь_с_вкладки_API_компонента.approvalCycle.agreed.approvalFactDate – текущие дата и время;
- путь_с_вкладки_API_компонента.approvalCycle.agreed.approvalResult – значение поля, код которого указан в поле «False Code» на вкладке «Data» справочника, выбранного в поле «Dictionary Name», где значение в поле, код которого указан в «Link Code» равно: путь_с_вкладки_API_компонента.approvalCycle.agreed.approvalTypeCode, того элемента массива, куда осуществляется сохранение.

При нажатии кнопки «Reject Butto»n осуществляются следующие действия:

- копируется узел путь_с_вкладки_API_компонента.approvalCycle;
- создается новый элемент массива: путь_с_вкладки_API_компонента.approvalCycleHistory, куда вставляются скопированные ранее данные;
- очищается узел: путь_с_вкладки_API_компонента.approvalCycle.

4.6.6.5.15 Vr-request. Запросы в БР

Компонент предназначен для отображения таблицы со списком запросов в Базовый регистр (Рисунок 133).

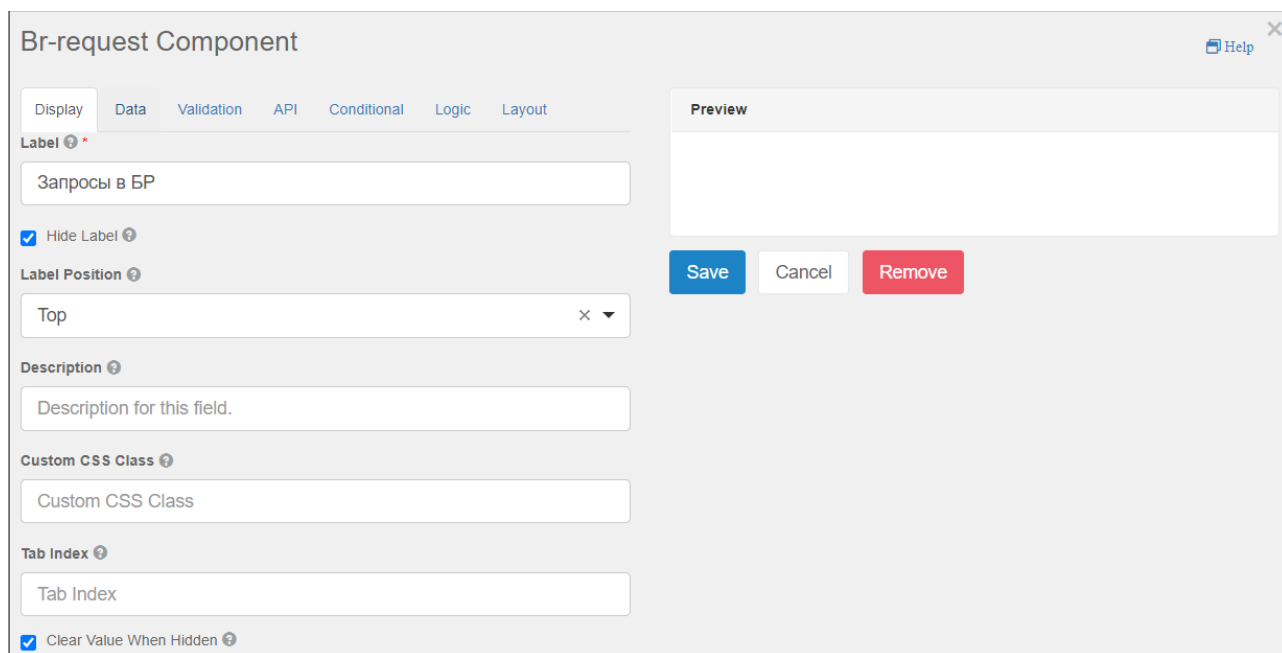


Рисунок 133 – Br-request компонент

Для отображения всех запросов в Базовый регистр необходимо установить флаг «Showcase» на вкладке «Data».

Для отображения запросов только по текущему документу необходимо в поле «Service Number» указать путь к полю с параметром «Service Number» в JSON документа (Рисунок 134).

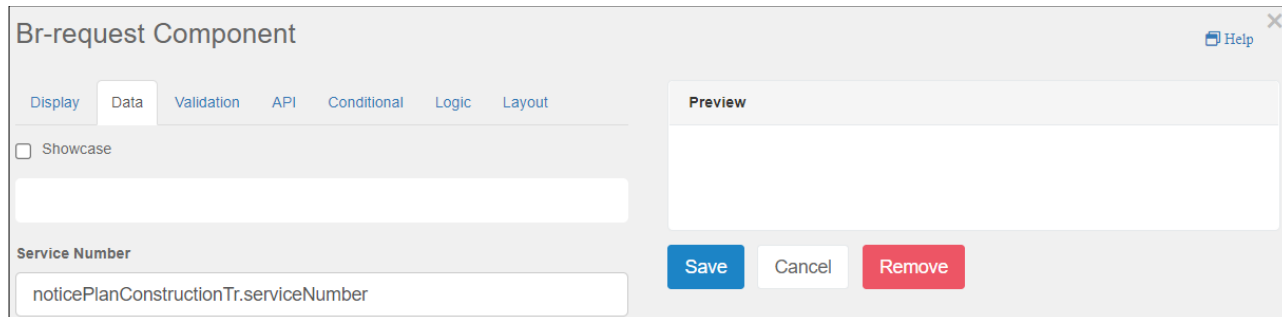


Рисунок 134 – Параметр «Service Number» в JSON документа

4.6.6.5.16 Link Button. Кнопка со ссылкой

Компонент предназначен для добавления на форму кнопок с абсолютными и относительными ссылками.

Внешний вид кнопки (цвет, иконка, всплывающая подсказка) настраиваются на вкладке «Display» (Рисунок 135).

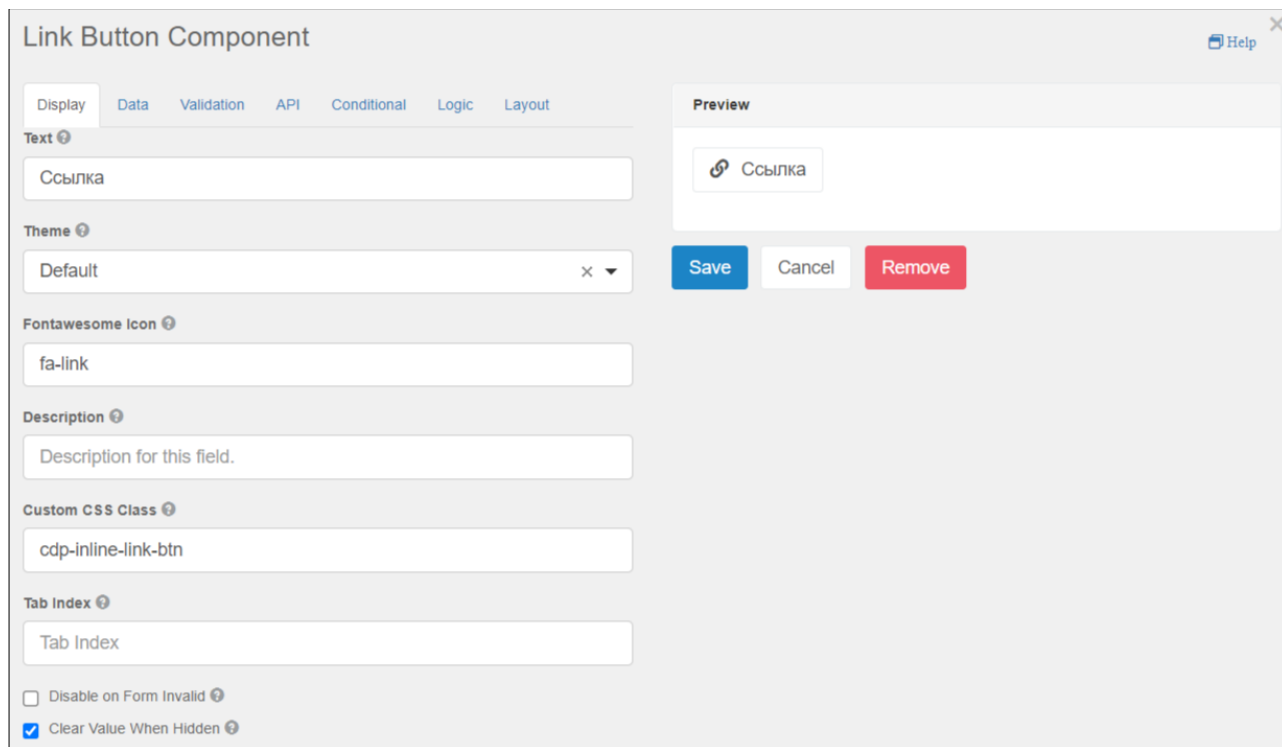


Рисунок 135 – Вкладка «Display». Настройки вида кнопки

Ссылка, открываемая по кнопке, указывается на вкладке «Data» (Рисунок 136).

Для использования в ссылке переменной из JSON путь к переменной необходимо указывать в двойных фигурных скобках.

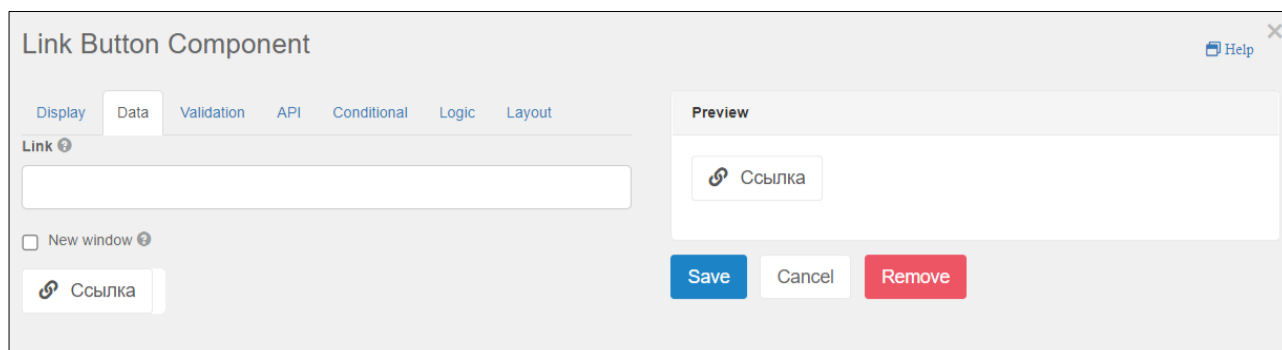


Рисунок 136 – Вкладка «Data». Настройки ссылки

4.6.6.5.17 Status badge. Статус в цветной плашке

Компонент предназначен для добавления в карточку значения поля в цветной плашке (Рисунок 137).

Цвет плашки должен быть определен кодом в справочнике, значение поля в JSON должно совпадать со значением поля в справочнике.

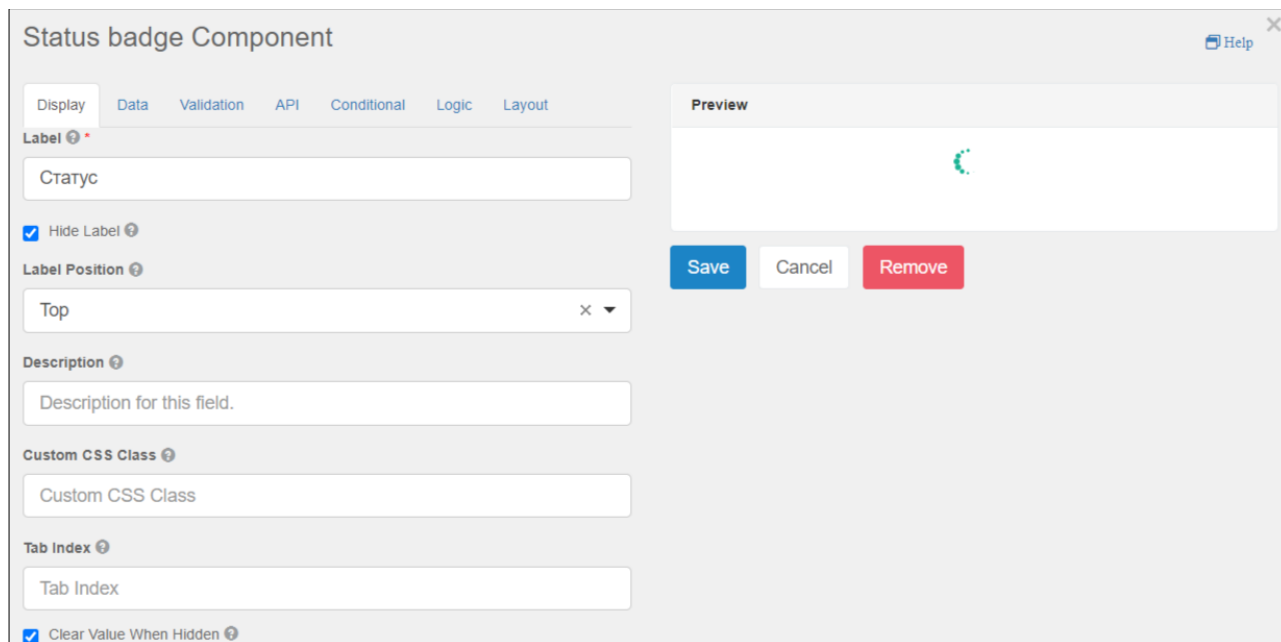


Рисунок 137 – Status badge компонент

В поле «Dictionary Name» вкладки «Data» необходимо выбрать справочник, содержащий список статусов и соответствующих им цветов (выбор осуществляется из платформенных справочников и справочников системы, к которой принадлежит бизнес-процесс).

В поле «Value Field» необходимо указать поле справочника, с которым сравнивается значение поля JSON с вкладки «API».

В поле «Color Field» необходимо указать поле справочника, в котором указан код цвета соответствующий статусу (Рисунок 138)

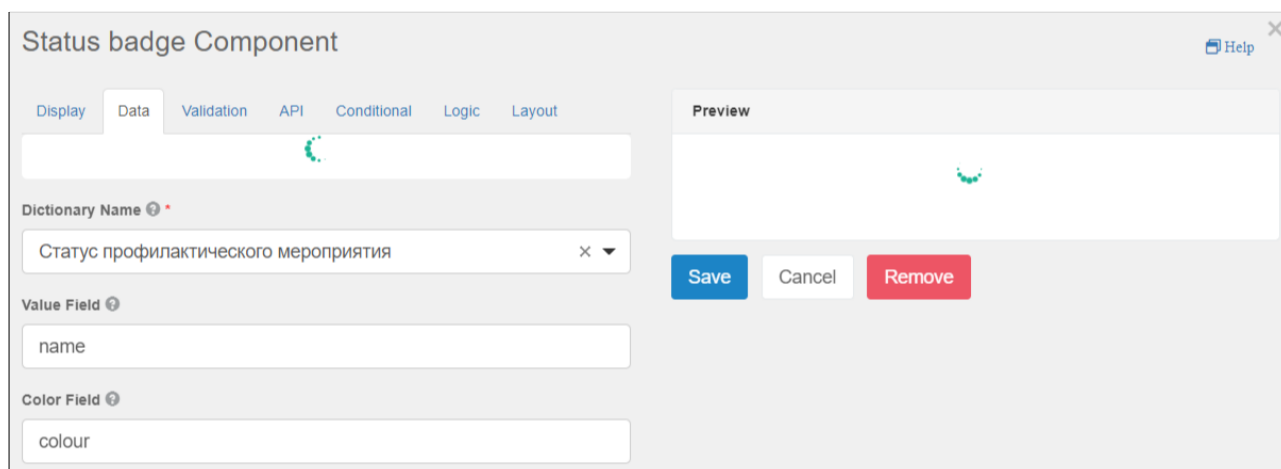


Рисунок 138 – Вкладка «Data». Настройки полей

4.6.6.5.18 Project-PD. Загрузка проектной и рабочей документации

Компонент предназначен для отображения блока загрузки проектной и рабочей документации (Рисунок 139).

Для корректной работы структура узла для сохранения данных должна соответствовать комплексному типу «folderType» (см. п. 4.6.6.5.8).

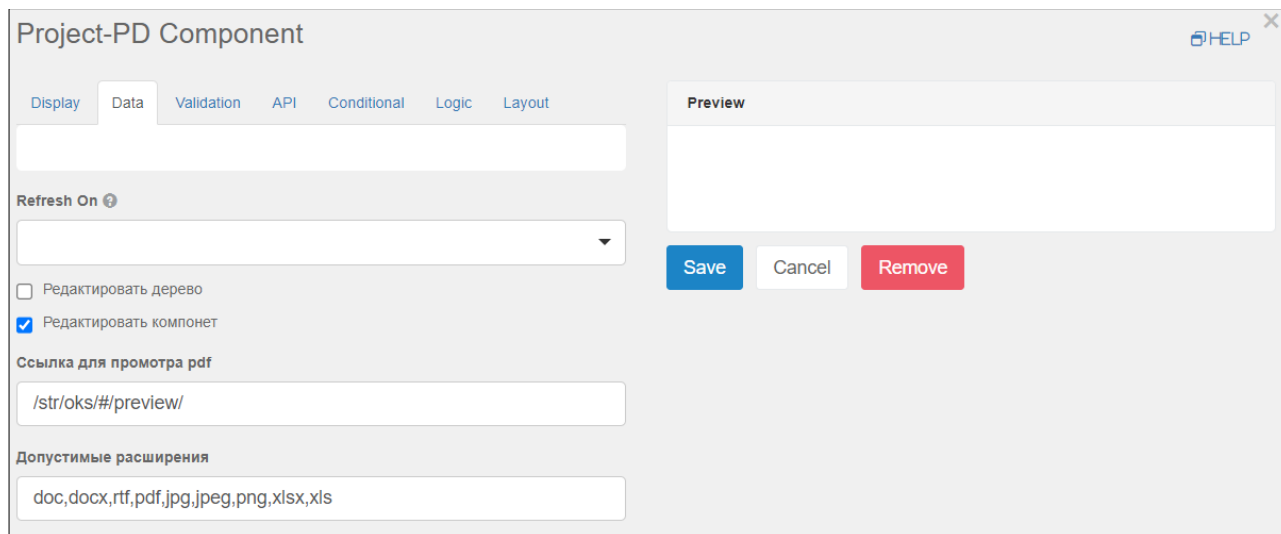


Рисунок 139 – Project-PD компонент. Установлен флаг «Редактировать компонент»
 На вкладку «Data» компонента добавлены специфические настройки (Таблица 59).

Таблица 59 – Специфические настройки вкладку «Data»

Поле	Описание
Редактировать дерево	При установленном флаге доступно изменение дерева папок
Редактировать компонент	При установленном флаге доступна загрузка / удаление файлов
Ссылка для просмотра pdf	–
Допустимые расширения	Список допустимых для загрузки расширений файлов с разделителем запятой

Пример внешнего вида компонента с установленными флагами «Редактировать дерево» и «Редактировать компонент» приведен ниже (Рисунок 140).

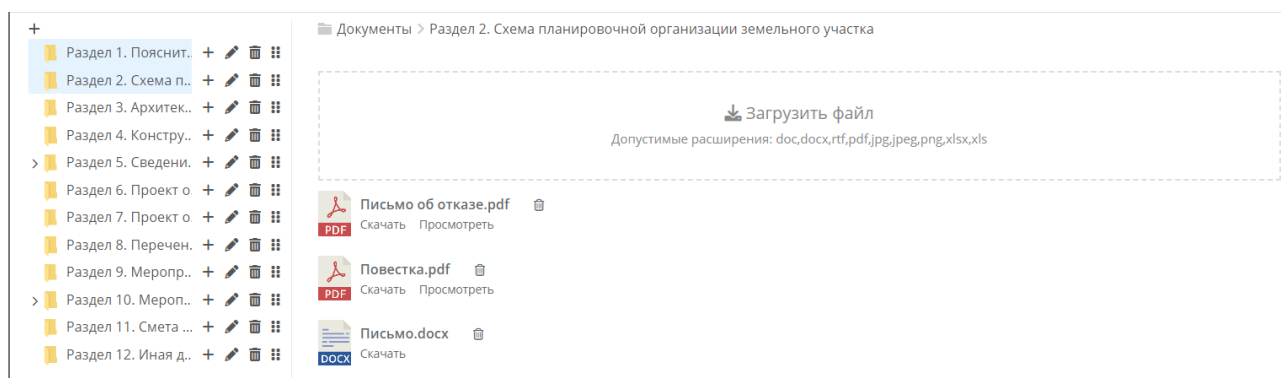


Рисунок 140 – Пример внешнего вида компонента

4.6.6.5.19 Pgu Events Table. Отображение статусов взаимодействия с ПГУ

Компонент предназначен для отображения статусов взаимодействия с ПГУ (Рисунок 141).

Извещение о начале СМР

№ regNumber_80 от 28.02.2023 Услуга оказана. Решение положительное

Основная информация | Документы | Результат | Внешние системы | Процессы | История изменений | </>

Статусы ПГУ | Запросы в БР

Дата	Тип события	Статус	Код статуса	Комментарий
23.03.2023 19:18:04	Отправка статуса	Услуга оказана. Решение положительное	1075	По Вашему заявлению подготовлено и оформлено Разрешение на строительство № 09-3-1805/22-(0)-0 от 26.10.2022
02.03.2023 10:25:11	Отправка статуса	Формирование результата	1052	
02.03.2023 10:25:11	Отправка статуса		10190	(Окончание периода, в течение которого заявитель имеет возможность отозвать заявление) направляется до статуса 1052
02.03.2023 10:23:00	Отправка статуса	Формирование результата	1052	
02.03.2023 10:23:00	Отправка статуса		10190	(Окончание периода, в течение которого заявитель имеет возможность отозвать заявление) направляется до статуса 1052

Рисунок 141 – Пример отображения статусов взаимодействия с ПГУ

Для настройки компонента необходимо выбрать справочник со статусами на вкладке «Data» и указать путь к полю, где хранится «serviceNumber» в поле «Property Name» на вкладке «API».

4.6.6.5.20 Sign Button. Кнопка подписания файла

Компонент предназначен для добавления на форму задачи кнопки подписания файла (Рисунок 142).

При нажатии кнопки подписания файла выполняются следующие действия:

- сохранение документа процесса;
- вызов внешнего REST-API (опционально);
- подписание файла;
- визуализация электронной подписи (опционально);
- дополнение данных печатной формы при подписании (опционально).

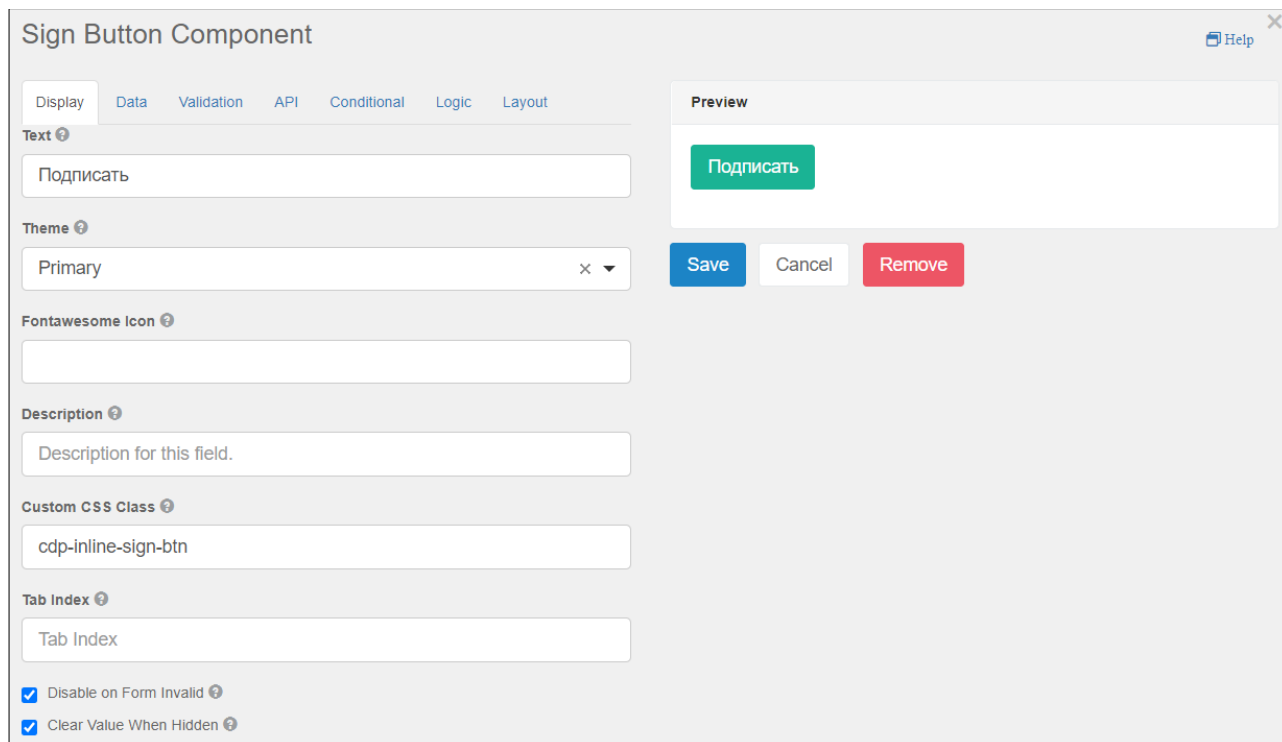


Рисунок 142 – Sign Button компонент. Параметры отображения
В закладке «Data» указываются параметры подписания (Рисунок 143).

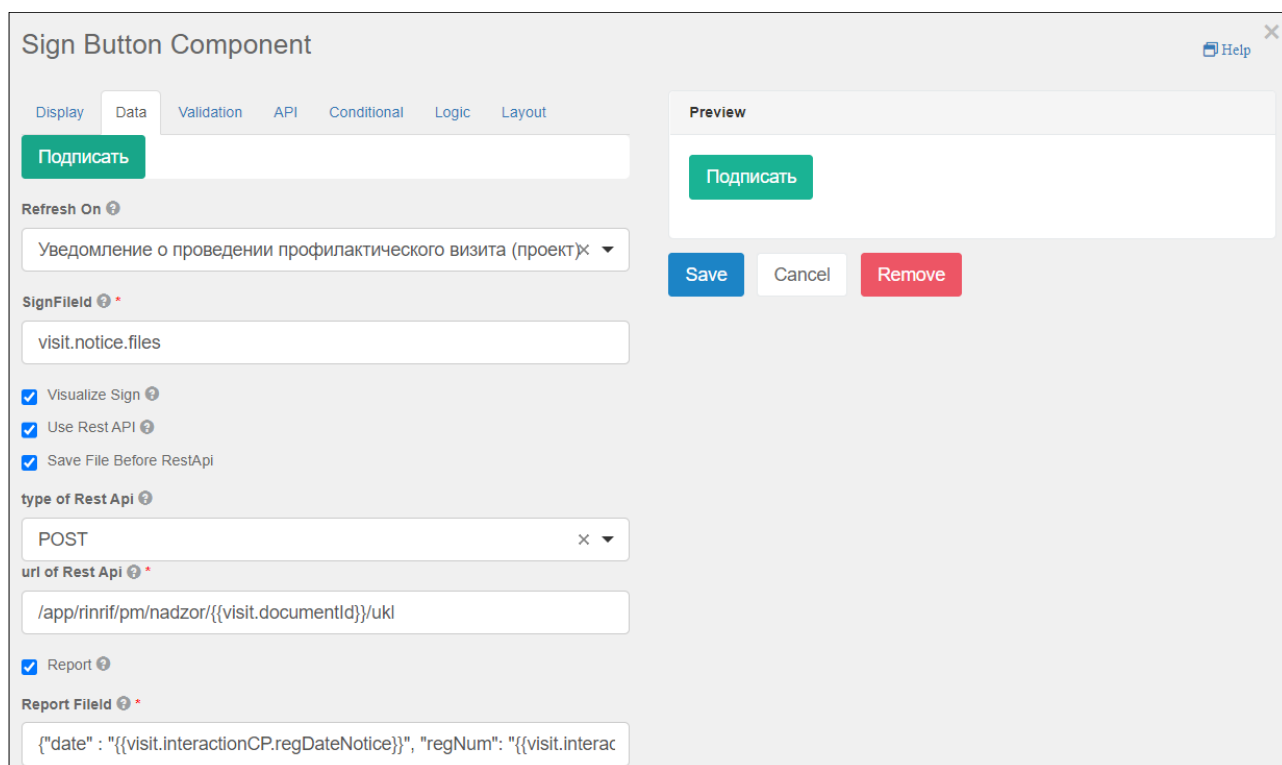


Рисунок 143 – Sign Button компонент. Параметры подписания

4.6.6.5.20.1 Визуализация электронной подписи

Визуализация электронной подписи возможна только для файлов формата doc и docx

При установке флага «Visualize Sign» выполняются следующие действия:

- заполнение в шаблоне данных в операторных скобках `sn` на основании данных в электронной подписи (Рисунок 144);

- формирование pdf-файла из обновленного файла формата doc и docx;
- запись идентификатора pdf-файла в узел, где ранее был сохранен файл формата doc и docx.

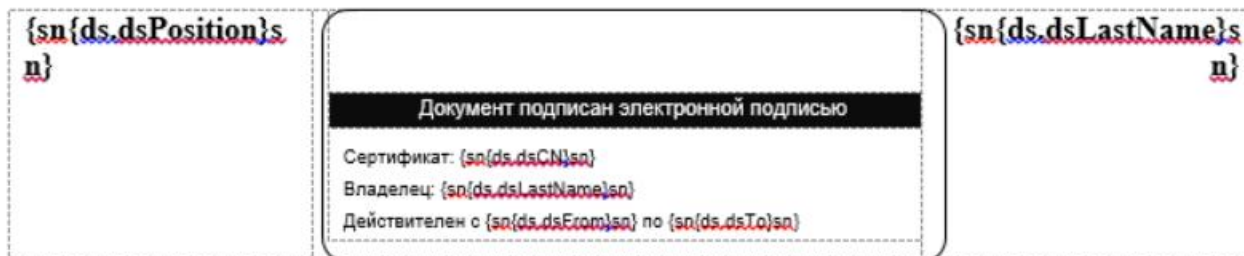


Рисунок 144 – Заполнение в шаблоне данных в операторных скобках sn
Пример визуализации электронной подписи приведен ниже (Рисунок 145).

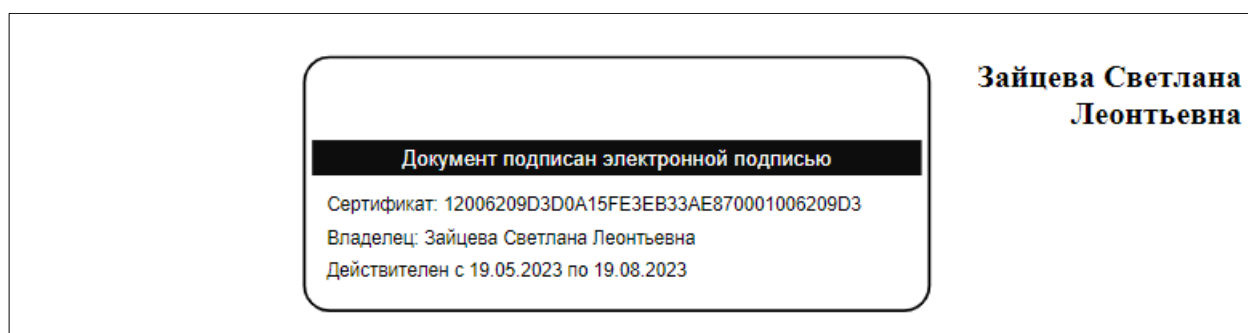


Рисунок 145 – Пример визуализации электронной подписи

4.6.6.5.20.2 Дополнение данных печатной формы при подписании

Дополнение данных печатной формы при подписании выполняется только для файлов формата doc и docx.

При установке флага «Report» и заполнении «Report FileId» одновременно с подписанием будет выполнено заполнение в шаблоне данных в операторных скобках ar (внутри операторных скобок sn) на основании данных в «Report FileId» (Рисунок 146).

Пример заполнения данных в поле «Report FileId»:

```
"date" : "{{visit.interactionCP.regDateNotice}}", "regNum":  
"{{visit.interactionCP.regNumNotice}}".
```

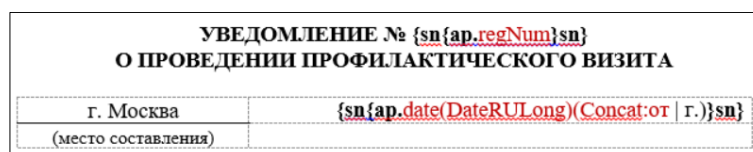


Рисунок 146 – Пример настройки шаблона

Пример дополнения данных печатной формы приведен ниже (Рисунок 147).

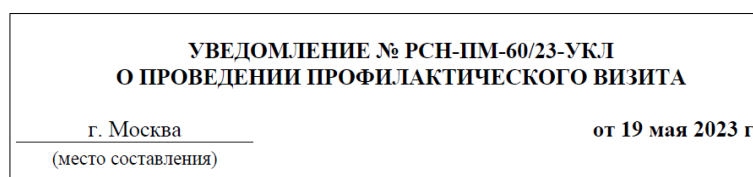


Рисунок 147 – Пример дополнения данных печатной формы

4.6.6.5.21 Dict. Выбор значения из плоского справочника

Компонент предназначен для работы с «плоскими» (не иерархическими) справочниками и представляет собой раскрывающийся список с единственным или множественным выбором (Рисунок 148).

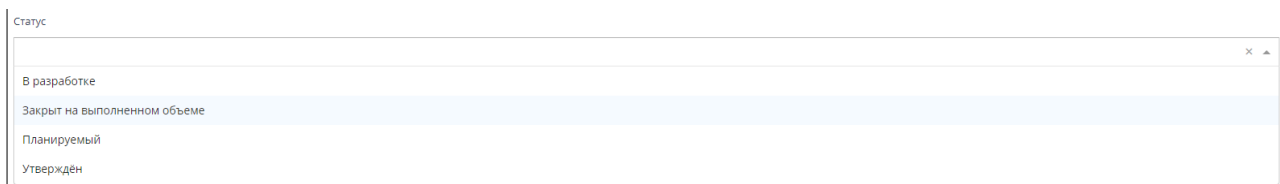


Рисунок 148 – Пример раскрывающегося списка

Для корректной работы компонента «Dict» хранение справочных значений в JSON документа должно соответствовать структуре, указанной ниже (Рисунок 149).

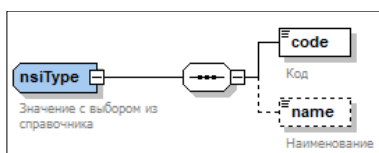


Рисунок 149 – Структура хранения справочных значений

Примеры структуры XSD схем для хранения единичного справочного значения и массива приведен ниже (Рисунок 150).

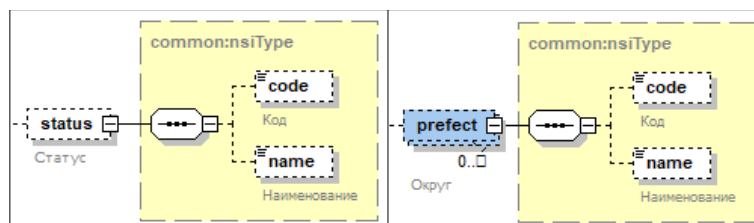


Рисунок 150 – Структура XSD схем для хранения единичного справочного значения и массива

Для справочников с очень длинным значением на вкладке «Display» необходимо указать параметр «Custom CSS Class» – prevent-sel-elt-to-expand-page-hor, чтобы избежать «разрывания» окна браузера при выборе значения (Рисунок 151).

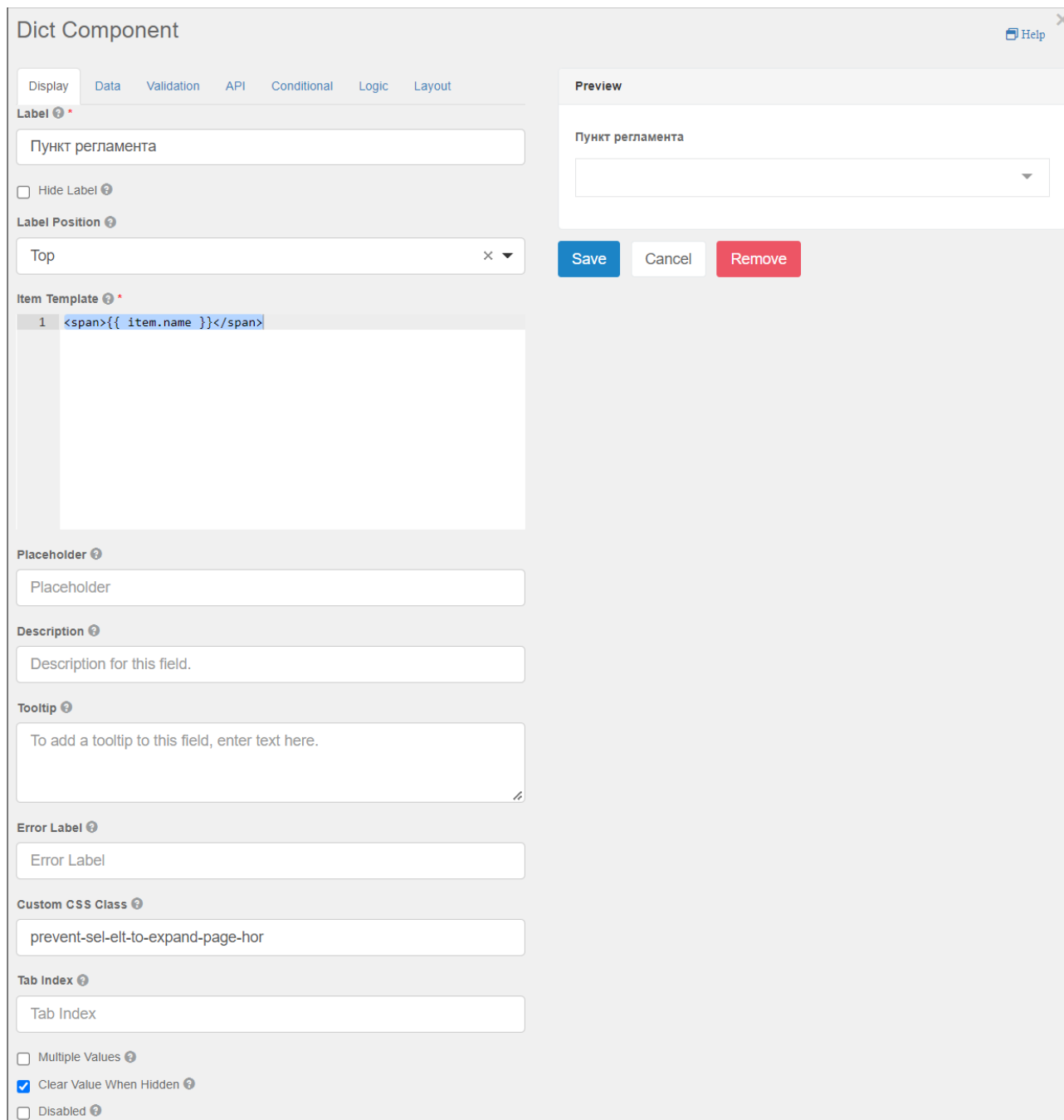


Рисунок 151 – Параметр «Custom CSS Class»

На вкладке «Data» добавлены специфические поля для работы данного компонента (Рисунок 152, Таблица 60).

Dict Component [HELP] [X]

Display | **Data** | Validation | API | Conditional | Logic | Layout

Default Value [?] * Default Value

Dictionary Name [?] * Цели проверок x

Sorting Value * name

Sorting Direction * ASC x

Code Field * code

Name Field * name

Refresh On [?] *

Relation

Make these values unique within DataGrid

Filter Params

```

1  {
2    "comparison": "EQUAL",
3    "nickAttr": "code",
4    "values": [
5      "hiddenWorks", "inspection"
6    ]
7  }

```

Preview

Предмет проверки

Save Cancel Remove

Рисунок 152 – Вкладка «Data». Специфические поля

Таблица 60 – Специфические поля вкладки «Data»

Наименование поля	Обязательность	Описание
Dictionary Name	Да	Список с единичным выбором Предназначено для установления связи со справочником Наполняется «плоскими» справочниками, где наименование системы = или SYSTEM или COMMON или коду системы, в рамках которой разрабатывается бизнес-процесс
Sorting Value	Да	Поле ввода Предназначено для указания кода элемента справочника, по которому осуществляется сортировка значений списка
Sorting Direction	Да	Список с единичным выбором Предназначено для указания направления сортировки значений списка по полю, выбранному в Sorting Value

Наименование поля	Обязательность	Описание
		Наполняется значениями ASC и DESC
Code Field	Да	Поле ввода Предназначено для указания кода элемента справочника, принимаемого как код элемента для сохранения в JSON документа
Name Field	Да	Поле ввода Предназначено для указания кода элемента справочника, принимаемого как наименование элемента для отображения в списке и сохранения в JSON документа
Relation	Нет	Флаг Позволяет настроить зависимость значений текущего справочника от значений ссылочного
Linked Field	Да, при установленном Relation	Поле ввода Код поля текущего справочника, в котором установлена ссылка на связанный
Related Dict	Да, при установленном Relation	Список с единичным выбором Связанный справочник
Related Dict Related Field	Да, при установленном Relation	Поле ввода Код поля связанного справочника, по которому установлена ссылка (по умолчанию id)
Related Dict Code Field	Да, при установленном Relation	Поле ввода Код поля связанного справочника, в котором хранится код элемента
Related Dict Value Path	Да, при установленном Relation	Поле ввода Путь до значения связанного справочника в JSON документа (путь к узлу, содержащему в себе вложенные элементы code и name)
Make these values unique within DataGrid	Нет	Флаг При использовании компонента внутри Data Grid проверяет уникальность значения и не позволяет выбрать один и тот же элемент справочника в разных строках таблицы
Filter Params	Нет	Расширяемый текстовый блок Для добавления фильтрации значений по правилам поисковых запросов к справочникам { "comparison": "LIKE", "nickAttr": "code", "values": ["ProjectDocumentation"] }

Настройки зависимости справочника «Районы» от выбранного значения справочника

«Округа» приведены ниже (Рисунок 153).

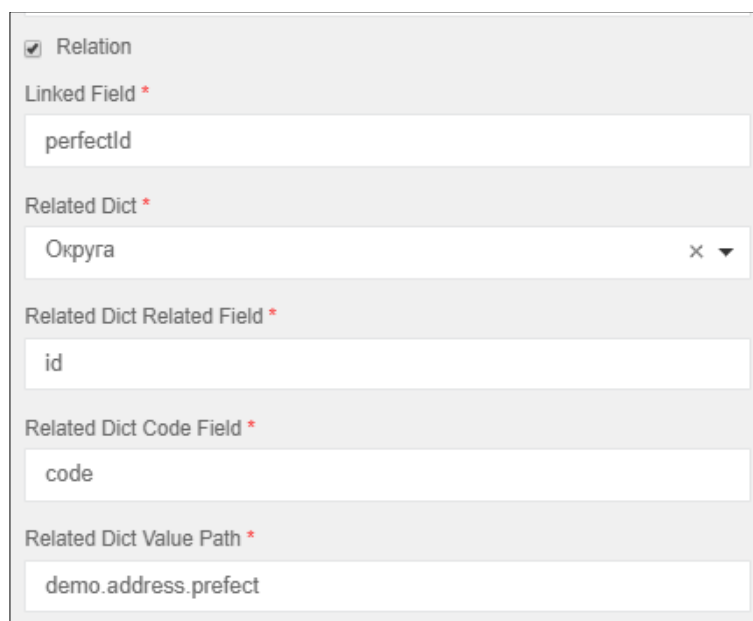


Рисунок 153 – Настройки зависимости справочника «Районы»

На вкладке «API» должен быть указан путь к узлу, содержащему в себе вложенные элементы code и name (Рисунок 154).

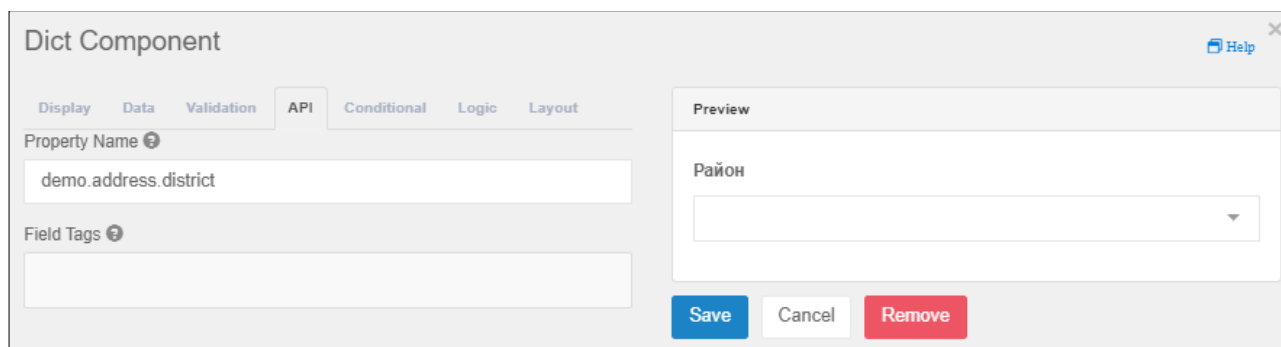


Рисунок 154 – Настройки вкладки «API»

При завершении задачи и не установленном на вкладке «Display» флаге «Multiple» в JSON документа сохраняются значения в узле, путь к которому указан на вкладке «API», где:

- путь_с_вкладки_API.code – значение поля code справочника;
- путь_с_вкладки_API.name – значение поля справочника, указанное в поле «Name Field» на вкладке «Data».

При завершении задачи и установленном на вкладке «Display» флаге «Multiple» в JSON документа должен быть очищен узел, путь к которому указан на вкладке «API» и созданы новые элементы массива, где:

- путь_с_вкладки_API.code – значение поле code справочника;
- путь_с_вкладки_API.name – значение поля справочника, указанное в поле «Name Field» на вкладке «Data».

4.6.6.5.22 List Hierarchic Dict. Выбор значения из иерархического справочника (раскрывающийся список)

Компонент предназначен для работы с иерархическими справочниками и представляет собой раскрывающийся список с единичным или множественным выбором (Рисунок 155).

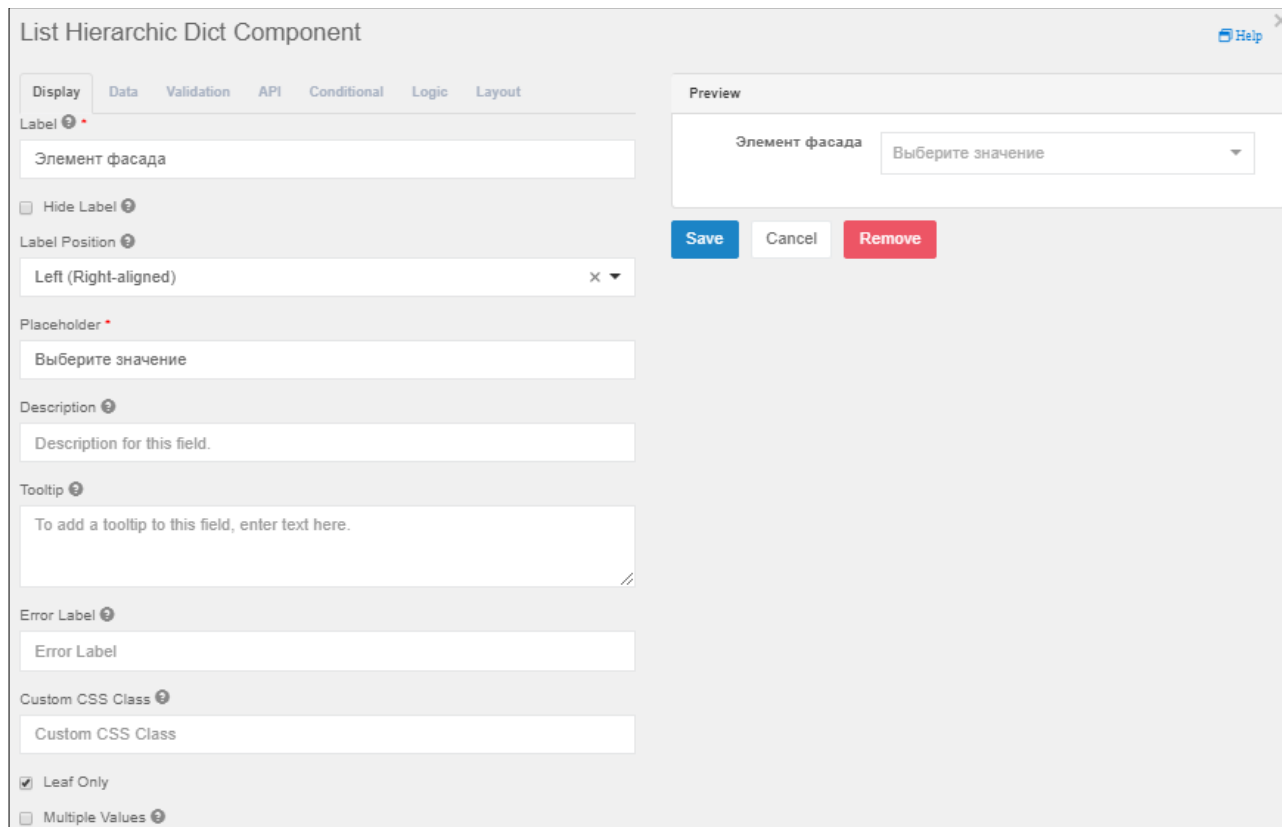


Рисунок 155 – List Hierarchic Dict компонент

Для корректной работы компонента List Hierarchic Dict хранение справочных значений в JSON документа должно соответствовать структуре, указанной ниже (Рисунок 156).

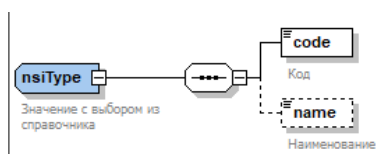


Рисунок 156 – Структура хранения справочных значений

Примеры структуры XSD схем для хранения единичного справочного значения и массива приведены ниже (Рисунок 157).

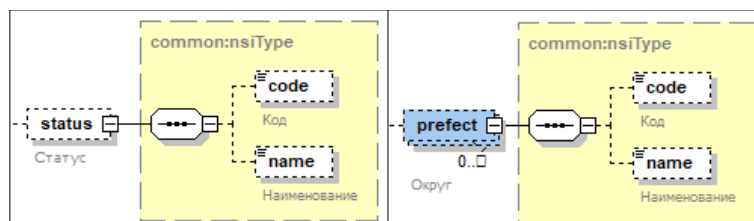


Рисунок 157 – Примеры структуры XSD схем единичного справочного значения и массива

На вкладке «Display» добавлено специфическое поле «Leaf Only» для работы данного

компонента (Таблица 61).

Таблица 61 – Описание специфического поля «Leaf Only»

Наименование поля	Обязательность	Описание
Leaf Only	Нет	Чекбокс При установленном флаге Leaf Only доступны для выбора только элементы самого низкого уровня иерархии ("листья" справочника)

На вкладке «Data» добавлены специфические поля для работы данного компонента (Рисунок 158, Таблица 62).

Рисунок 158 – Специфические поля вкладки «Data»

Таблица 62 – Специфические поля вкладки «Data»

Наименование поля	Обязательность	Описание
Dictionary Name	Да	Список с единичным выбором Предназначено для установления связи со справочником Наполняется иерархическими справочниками, где наименование системы = или SYSTEM или COMMON или коду системы, в рамках которой разрабатывается бизнес-процесс
Sorting Value	Да	Поле ввода Предназначено для указания кода элемента справочника, по которому осуществляется сортировка значений списка
Sorting Direction	Да	Список с единичным выбором

Наименование поля	Обязательность	Описание
		Предназначено для указания направления сортировки значений списка по полю, выбранному в Sorting Value Наполняется значениями ASC и DESC
Code Field	Да	Поле ввода Предназначено для указания кода элемента справочника, принимаемого как код элемента для сохранения в JSON документа
Name Field	Да	Поле ввода Предназначено для указания кода элемента справочника, принимаемого как наименование элемента для отображения в списке и сохранения в JSON документа
Relation	Нет	Флаг Позволяет настроить зависимость значений текущего справочника от значений ссылочного
Linked Field	Да, при установленном Relation	Поле ввода Код поля текущего справочника, в котором установлена ссылка на связанный
Related Dict	Да, при установленном Relation	Список с единичным выбором Связанный справочник
Related Dict Related Field	Да, при установленном Relation	Поле ввода Код поля связанного справочника, по которому установлена ссылка (по умолчанию id)
Related Dict Code Field	Да, при установленном Relation	Поле ввода Код поля связанного связанного справочника, в котором хранится код элемента
Related Dict Value Path	Да, при установленном Relation	Поле ввода Путь до значения связанного справочника в JSON документа (путь к узлу, содержащему в себе вложенные элементы code и name)
Make these values unique within DataGrid	Нет	Флаг При использовании компонента внутри Data Grid проверяет уникальность значения и не позволяет выбрать один и тот же элемент справочника в разных строках таблицы
Filter Params	Нет	Расширяемый текстовый блок Для добавления фильтрации значений по правилам поисковых запросов к справочникам { "comparison": "LIKE", "nickAttr": "code", "values": [

Наименование поля	Обязательность	Описание
		"ProjectDocumentation"] }

На вкладке «API» должен быть указан путь к узлу, содержащему вложенные элементы code и name.

При завершении задачи и неустановленном на вкладке «Display» флаге «Multiple» в JSON документа сохраняются значения в узле, путь к которому указан на вкладке «API», где:

- путь_с_вкладки_API.code – значение поля code справочника;
- путь_с_вкладки_API.name = значение поля справочника, указанное в поле «Name Field» на вкладке «Data».

При завершении задачи и установленном на вкладке «Display» флаге «Multiple» в JSON документа должен быть очищен узел, путь к которому указан на вкладке «API» и созданы новые элементы массива, где:

- путь_с_вкладки_API.code – значение поле code справочника;
- путь_с_вкладки_API.name – значение поля справочника, указанное в поле «Name Field» на вкладке «Data».

4.6.6.5.23 Modal Hierarchic Dict. Выбор значения из иерархического справочника (модальное окно)

Компонент предназначен для работы с иерархическими справочниками и представляет собой модальное окно со встроенным справочником (Рисунок 159).

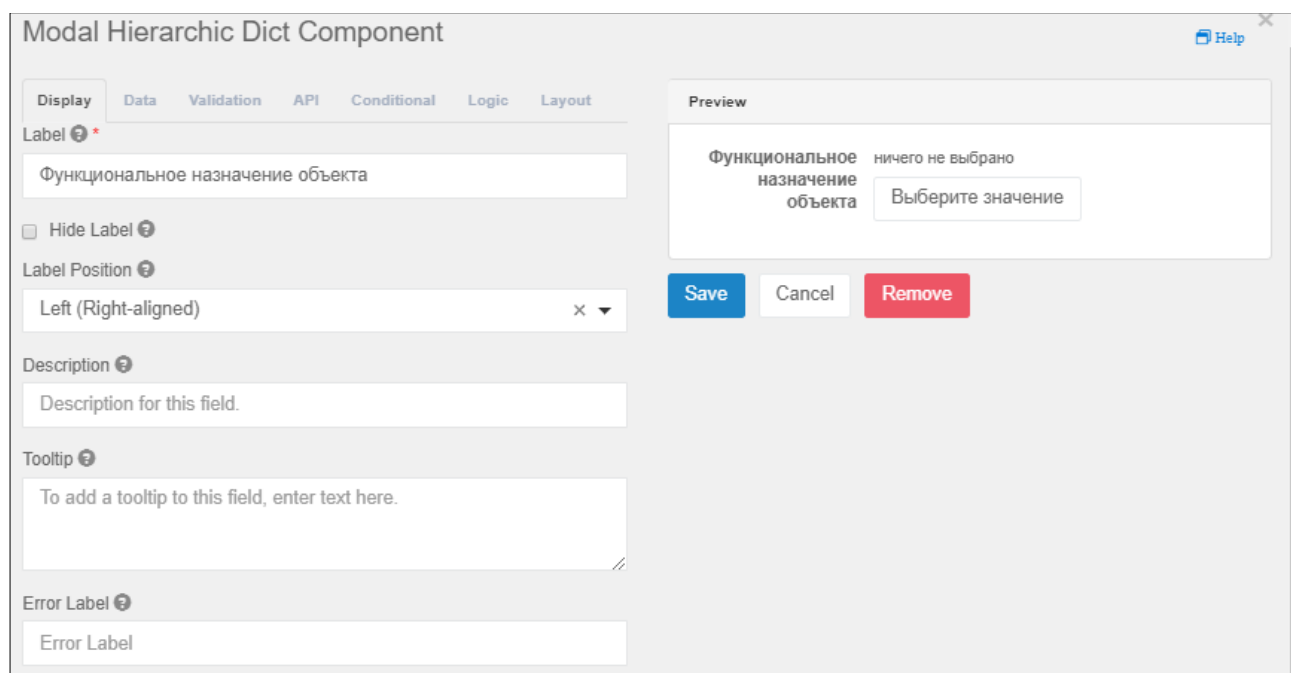


Рисунок 159 – Modal Hierarchic Dict компонент

Для корректной работы компонента Modal Hierarchic Dict хранение справочных значений в JSON документа должно соответствовать структуре комплексного типа nsType (см. п. 4.6.6.5.8).

Примеры структуры XSD схем для хранения единичного справочного значения и массива приведены ниже (Рисунок 160).

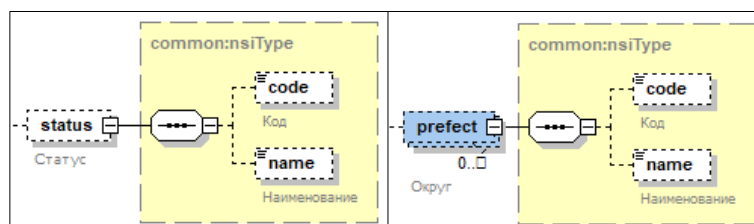


Рисунок 160 – Структуры XSD схем справочного значения и массива

Параметры вкладки «Display» компонента Modal Hierarchic Dict приведены ниже (Таблица 63).

Таблица 63 – Параметры вкладки «Display» компонента Modal Hierarchic Dict

Наименование поля	Обязательность	Описание
Button Name	Да	Поле ввода Предназначено для указания имени кнопки, по нажатию на которую должно открываться модальное окно
Modal Name	Да	Поле ввода Предназначено для указания имени открывающегося модального окна
Leaf Only	Нет	Чекбокс При установленном флаге Leaf Only доступны для выбора только элементы самого низкого уровня иерархии ("листья" справочника)

На вкладке «Data» добавлены специфические поля для работы данного компонента (Рисунок 161, Таблица 64).

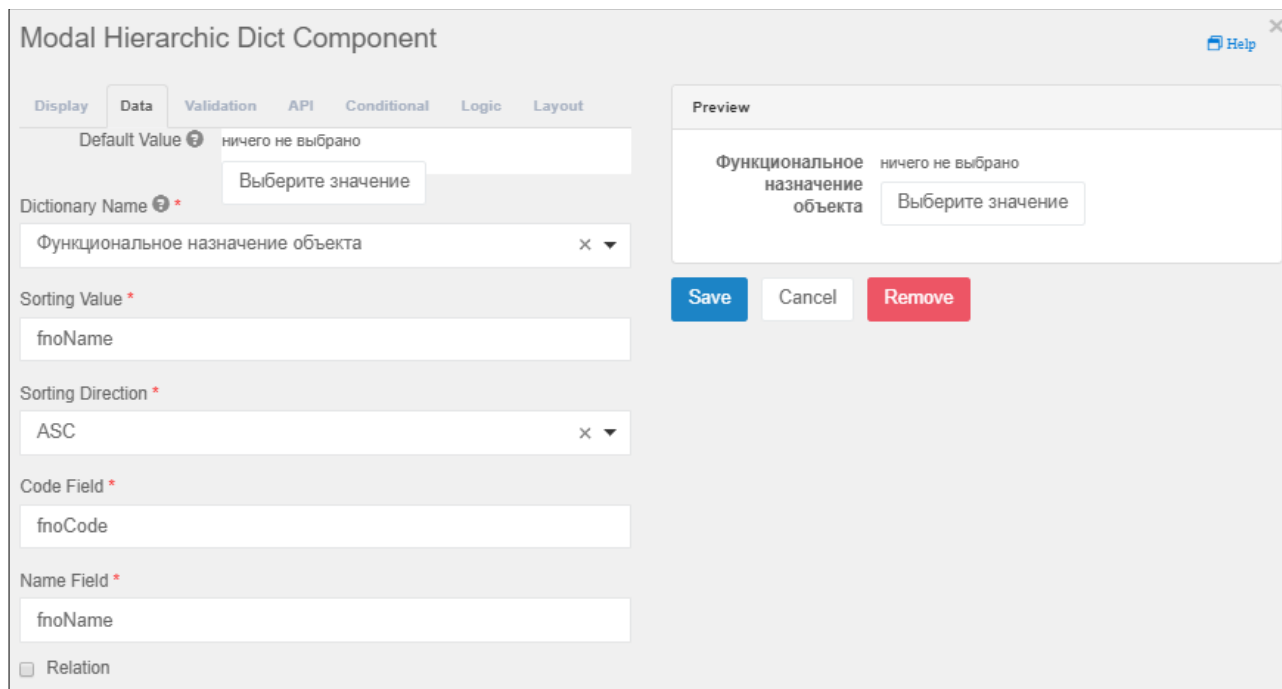


Рисунок 161 – Специфические поля вкладки «Data»

Таблица 64 – Специфические поля вкладки «Data»

Наименование поля	Обязательность	Описание
Dictionary Name	Да	Список с единичным выбором Предназначено для установления связи со справочником Наполняется иерархическими справочниками, где наименование системы = или SYSTEM или COMMON или коду системы, в рамках которой разрабатывается бизнес-процесс
Sorting Value	Да	Поле ввода Предназначено для указания кода элемента справочника, по которому осуществляется сортировка значений списка
Sorting Direction	Да	Список с единичным выбором Предназначено для указания направления сортировки значений списка по полю, выбранному в Sorting Value Наполняется значениями ASC и DESC
Code Field	Да	Поле ввода Предназначено для указания кода элемента справочника, принимаемого как код элемента для сохранения в JSON документа
Name Field	Да	Поле ввода Предназначено для указания кода элемента справочника, принимаемого как наименование элемента для отображения в списке и сохранения в JSON документа

Наименование поля	Обязательность	Описание
Relation	Нет	Флаг Позволяет настроить зависимость значений текущего справочника от значений ссылочного
Linked Field	Да, при установленном Relation	Поле ввода Код поля текущего справочника, в котором установлена ссылка на связанный
Related Dict	Да, при установленном Relation	Список с единичным выбором Связанный справочник
Related Dict Related Field	Да, при установленном Relation	Поле ввода Код поля связанного справочника, по которому установлена ссылка (по умолчанию id)
Related Dict Code Field	Да, при установленном Relation	Поле ввода Код поля связанного справочника, в котором хранится код элемента
Related Dict Value Path	Да, при установленном Relation	Поле ввода Путь до значения связанного справочника в JSON документа (путь к узлу, содержащему в себе вложенные элементы code и name)
Make these values unique within DataGrid	Нет	Флаг При использовании компонента внутри Data Grid проверяет уникальность значения и не позволяет выбрать один и тот же элемент справочника в разных строках таблицы
Filter Params	Нет	Расширяемый текстовый блок Для добавления фильтрации значений по правилам поисковых запросов к справочникам <pre>{ "comparison": "LIKE", "nickAttr": "code", "values": ["ProjectDocumentation"] }</pre>

При завершении задачи при установленном на вкладке «Display» флаге «Multiple» в JSON документа сохраняются значения в узле, путь к которому указан на вкладке «API», где:

- путь_с_вкладки_API.code – значение поля code справочника;
- путь_с_вкладки_API.name – значение поля справочника, указанное в поле «Name Field» на вкладке «Data».

При завершении задачи при установленном на вкладке «Display» флаге «Multiple» в

JSON документа должен быть очищен узел, путь к которому указан на вкладке «API» и созданы новые элементы массива, где:

- путь_с_вкладки_API.code – значение поле code справочника;
- путь_с_вкладки_API.name – значение поля справочника, указанное в поле «Name Field» на вкладке «Data».

4.6.6.5.24 Solr Select. Установление связи с другой сущностью

Компонент предназначен для выбора связанной сущности из SOLR и сохранение идентификатора связанной сущности в JSON документа бизнес-процесса (Рисунок 162).

Solr Select Component
HELP

Display
Data
Validation
API
Conditional
Logic
Layout

REST*

Fields To Display

1	{element.objectName}
---	----------------------

Enter custom javascript code.

The following variables are available in all scripts.

element	Single element, returned from SOLR query. SOLR query returns an array of elements.
functions	Additional custom functions, which could be used within the query. The following functions are supported:
functions.formatDate	Function for formatting date string/number. Call <code>FormatDate(date, outputFormat, [optional inputFormat])</code> . input format might be empty for numeric date.

Example

```
{(element.documentTypeName) * {(element.documentNumber) or
```

Value Field*

Solr Types*

Solr Sort*

Solr Query

1	{(objectName:*{search}*) OR {objectAddress:*{search}*}
---	--

Enter custom javascript code.

The following variables are available in all scripts.

search	Search string
data	Whole JSON data

Example

```
{{(documentTypeCode: setOffProjectDocumentation) OR {document
```

Включить поиск типа отправки формы (cpInstantSearchMode)

Автоматически подставлять выбранные значения (cpSubstituteSelectData)

Refresh On

Object

Prefill Select From Value

The following variables are available in all scripts.

form	The complete form JSON object
submission	The complete submission object
data	The complete submission data object
row	Contextual "row" data, used within DataGrid, EditGrid, and Container components
component	The current component JSON
instance	The current component instance
value	The current value of the component
moment	The moment.js library for date manipulation
_	An instance of Lodash
utils	An instance of the FormUtils object
util	An alias for "utils"

JavaScript

1	value = row.objectId
---	----------------------

Enter custom javascript code.

Example:

```
value = data.firstName + " " + data.lastName;
```

JSONLogic

Prefill Solr Query (used by Prefill Select From Value)

1	{(sys_documentId:*{search})}
---	------------------------------

Enter custom javascript code.

The following variables are available in all scripts.

search	Search string
data	Whole JSON data

Example

```
{{(documentTypeCode: setOffProjectDocumentation) OR {document
```

Preview

Наименование

Для начала поиска введите не менее трех символов...

Save Cancel Remove

Рисунок 162 – Solr Select компонент

На вкладку «Data» компонента Solr Select добавлены специфические поля, представленные ниже (Таблица 65).

Таблица 65 – Специфические поля вкладки «Data» компонента Solr Select

Наименование	Описание	Пример
REST	Путь к методу query в приложении search	/search/v1/solr/query
Fields To Display	Список полей связанной сущности, которые будут отображаться в списке в результате поиска Оборачиваются спецсимволами \${element.кодполя}	\${element.fullName}
Value FieldId	Сохраняемое поле SOLR По умолчанию устанавливается sys_documentId, сохранение других атрибутов возможно только при использовании предзаполнения других полей	sys_documentId
Solr Sort	Список полей и направление сортировки	fullName asc
Solr Query	Структура запроса, передаваемого в search Переменные значения (вводимые пользователем в строке) оборачиваются спецсимволами \${search}	inn:[* TO *] AND (fullName:*\${search}*) OR (shortName:*\${search}*)
Выполнять поиск при открытии формы	При установленном флаге обращение к SOLR идет сразу при открытии формы и не требует ввода трех символов Рекомендовано использовать только с небольшим набором данных	—
Автоматически подставлять выбранные значения	Работает только совместно с флагом "Выполнять поиск при открытии формы" Предназначено для автозаполнения значения, например, для скрытых полей	—
Prefill Select From Value	Предзаполнять значение из другого поля Определяет откуда выбрать значение для предзаполнения	value = data.document.objectId
Prefill Solr Query (used by Prefill Select From Value)	Предзаполнять значение из другого поля Определяет куда передать значение для предзаполнения	(sys_documentId:\${search})

4.6.6.5.25 User Select. Выбор пользователя из списка

Для корректной работы компонента User Select для сохранения данных должен использоваться типовой userType (см. 4.6.6.5.8).

Компонент User Select предназначен для выбора пользователя из списка и сохранения его данных в JSON документа (Рисунок 163).

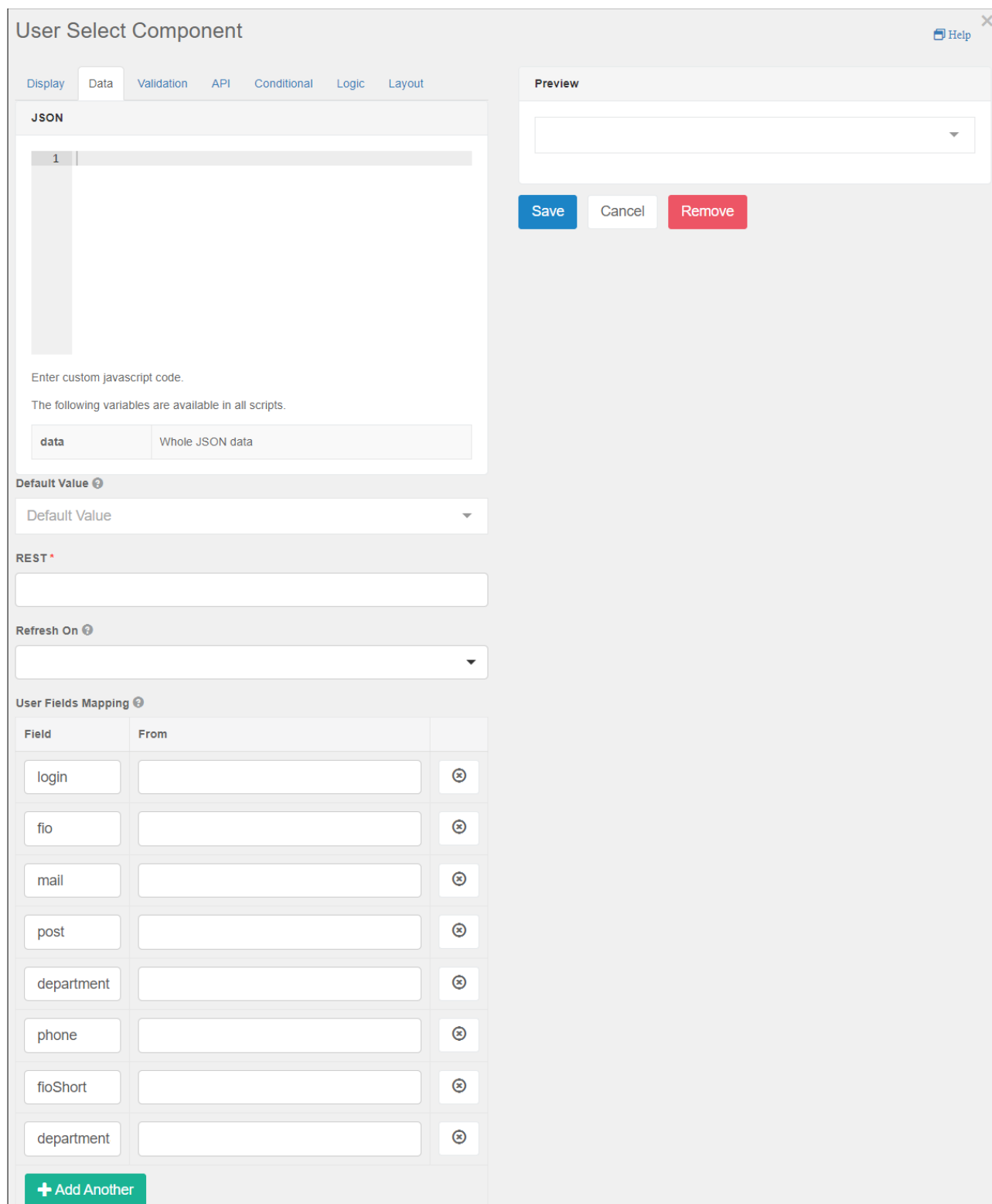


Рисунок 163 – User Select компонент

На вкладку «Data» компонента User Select добавлены специфические поля, представленные в таблице ниже (Таблица 66).

Таблица 66 – Специфические поля вкладки «Data» компонента User Select

Наименование	Описание	Пример																		
JSON	Запрос в формате JSON, который отфильтрует данные пользователей	<pre>{ "group": ["DEMO_RECONSTRUCTOR_SIGNER"] }</pre>																		
REST	Путь к методу POST, возвращающему данные пользователей, по переданным на вход критериям в формате JSON. Метод обязательно должен возвращать среди полей поле fio, так как именно по нему осуществляется сортировка и отображение данных в списке	/app/sys/mdm/api/v1/users																		
User Fields Mapping	Маппинг ответа метода на поля со сведениями о пользователе, входящими в узел, указанный в поле Property Name на вкладке API	<table border="1"> <thead> <tr> <th>Field</th> <th>From</th> </tr> </thead> <tbody> <tr> <td>login</td> <td>accountName</td> </tr> <tr> <td>fio</td> <td>displayName</td> </tr> <tr> <td>mail</td> <td>mail</td> </tr> <tr> <td>post</td> <td>post</td> </tr> <tr> <td>department</td> <td>departmentFullName</td> </tr> <tr> <td>phone</td> <td>telephoneNumber</td> </tr> <tr> <td>fioShort</td> <td>fioShort</td> </tr> <tr> <td>departmentShort</td> <td></td> </tr> </tbody> </table>	Field	From	login	accountName	fio	displayName	mail	mail	post	post	department	departmentFullName	phone	telephoneNumber	fioShort	fioShort	departmentShort	
Field	From																			
login	accountName																			
fio	displayName																			
mail	mail																			
post	post																			
department	departmentFullName																			
phone	telephoneNumber																			
fioShort	fioShort																			
departmentShort																				

4.6.6.6 CDP Doc Generator

4.6.6.6.1 MDM Document Generator. Формирование печатной формы по шаблону из справочника

Компонент MDM Document Generator предназначен для формирования печатной формы документа по шаблону из справочника (Рисунок 164).

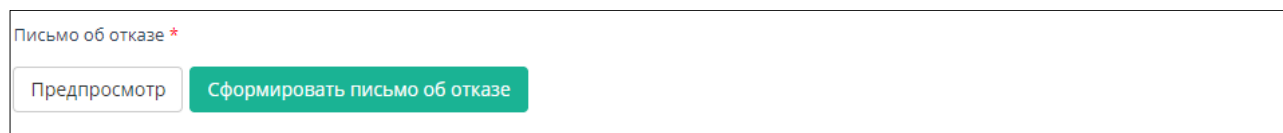


Рисунок 164 – Пример реализации компонента

Для использования компонента MDM Document Generator шаблон должен быть опубликован в справочнике «ReportTemplates» Шаблоны отчетов.

Для корректной работы компонента MDM Document Generator хранение

сформированной печатной формы в JSON документа должно соответствовать одной из двух структур, указанных в описании компонента File Manager (п. 4.6.6.5.1).

На вкладке «Display» добавлены специфические поля для работы данного компонента (Рисунок 165, Таблица 67).

Рисунок 165 – Специфические поля вкладки «Display»

Таблица 67 – Специфические поля вкладки «Display» компонента MDM Document Generator

Наименование поля	Обязательность	Описание
showErrorMessage	Нет	Чекбокс При установленном флаге в интерфейсе компонента выводится текст ошибки формирования печатной формы
showFileAttributes	Нет	Чекбокс При установленном флаге в интерфейсе компонента отображаются атрибуты сформированного файла
showSignButton	Нет	Чекбокс При установленном флаге в интерфейсе компонента отображается ссылка для подписания файла электронной подписью
showAttachSign	Нет	Чекбокс При установленном флаге в интерфейсе компонента отображается ссылка для прикладывания электронной подписи к сформированному файлу
generateButtonName	Нет	Поле ввода Предназначено для указания имени кнопки формирующей и сохраняющей печатную форму По умолчанию Сгенерировать отчет
generateButtonClass	Нет	Поле ввода Предназначено для указания кода цвета кнопки формирующей и сохраняющей печатную форму (например, btn btn-primary)

Наименование поля	Обязательность	Описание
showPreviewButton	Нет	Флаг Предназначен для скрытия кнопки предпросмотра сформированной печатной формы Рекомендован для установки при вставке изображения в печатную форму
previewButtonName	Нет	Поле ввода Предназначено для указания имени кнопки формирующей печатную кнопку для предпросмотра По умолчанию Предпросмотр
previewButtonClass	Нет	Поле ввода Предназначено для указания кода цвета кнопки формирующей печатную кнопку для предпросмотра (например, btn btn-default)

На вкладке «Data» добавлены специфические поля для работы данного компонента (Рисунок 166, Таблица 68).

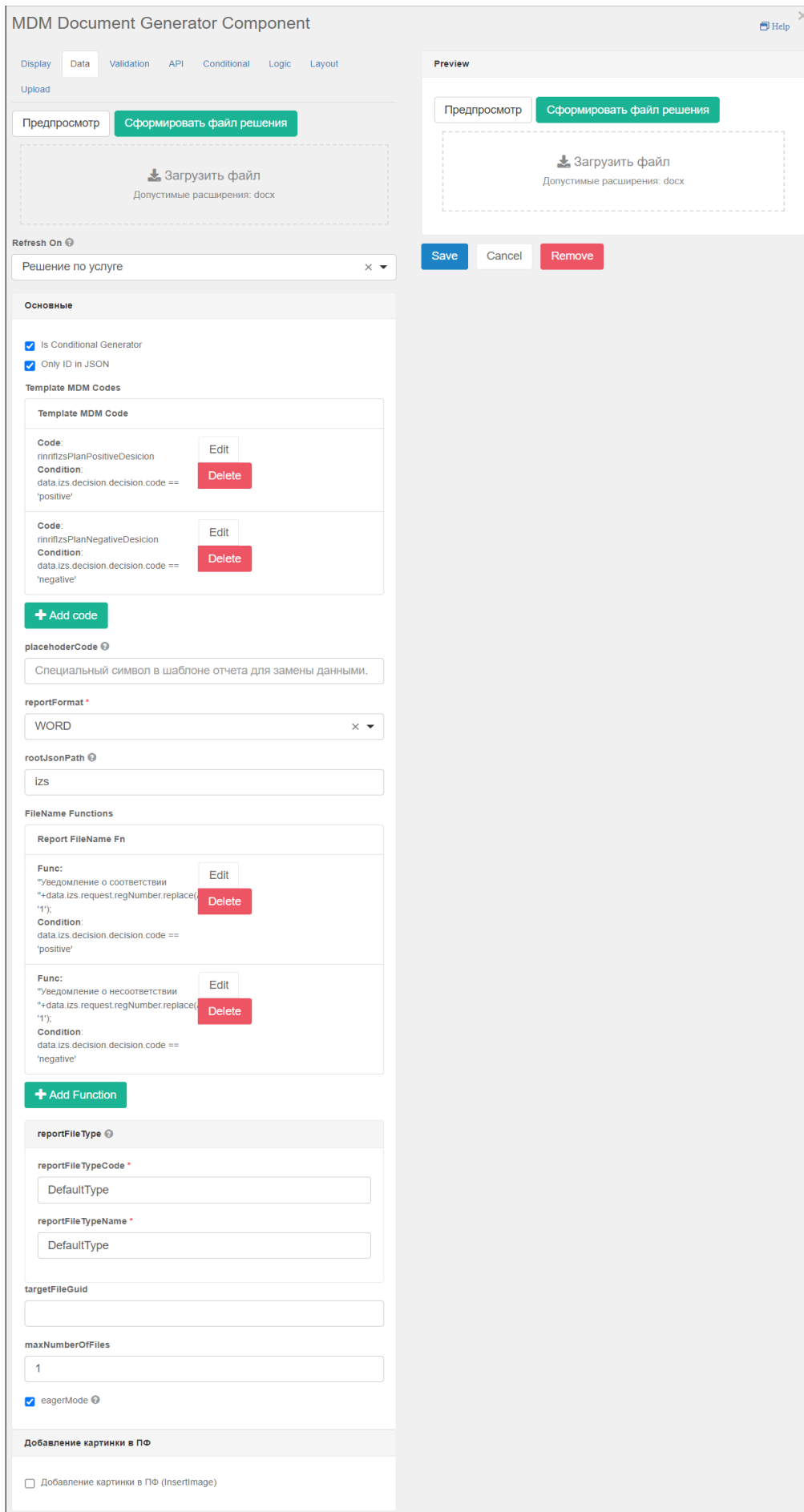



Рисунок 166 – Специфические поля вкладки «Data»

Таблица 68 – Специфические поля вкладки «Data» компонента MDM Document Generator

Наименование поля	Обязательность	Описание
Is Conditional Generator	Нет	Флаг Устанавливается в случае необходимости генерации разных печатных форм по условию из JSON документа
Only ID in JSON	Нет	Флаг Предназначен для сохранения в JSON документа только идентификатора сформированного файла
templateMdmCode	Да	Поле ввода Предназначено для указания кода шаблона печатной формы в справочнике ReportTemplates
Condition (Expression)	Да, при установленном флаге Is Conditional Generator	Поле ввода Предназначено для указания условия вызова указанного шаблона печатной формы Отображается при установленном флаге Is Conditional Generator
placeholderCode	Нет	Поле ввода Предназначено для указания специального символа в шаблоне отчета для замены данными
reportFormat	Да	Список с единичным выбором Наполняется значениями WORD и PDF Предназначено для определения формата сформированного файла
rootJsonPath	Нет	Поле ввода Предназначено для указания корневого узла в объекте данных jsonDataPath в спецификации json path (по умолчанию \$)
reportFileNameFn	Да	Текстовый блок Предназначено для указания js-кода для формирования имени отчета Постоянное значение необходимо указывать в кавычках, а переменную часть из JSON документа без кавычек с добавлением префикса data <i>"Письмо об отказе "+data.signboard.request.regNumber</i> Если сформированная печатная форма используется в листе согласования, необходимо каждый раз формировать документ с разным именем, чтобы обеспечить сохранность предыдущей версии для каждого цикла согласования (при использовании одинакового имени новая версия файла затирает предыдущую), в данном случае рекомендовано подставлять в имя файла дату и время формирования <i>"Приказ об утверждении ПД "+ new Date().toLocaleDateString()+ " " +new</i>

Наименование поля	Обязательность	Описание
		<i>Date().toLocaleTimeString()</i>
reportFileType	Да	Поле ввода Пара значений code и name для описания кода и наименования типа отчета в хранилище
eagerMode	Да	Чекбокс Флаг определяет выполнение сохранения JSON с данными, не дожидаясь стандартного асинхронного сохранения перед закрытием формы Флаг должен быть обязательно установлен для корректного сохранения сформированного файла
Добавление картинки в ПФ (InsertImage)	Нет	Чекбокс Предназначен для инициации вызова функции вставки изображения в печатную форму 
imagePath	Да (при установленном флаге InsertImage)	Поле ввода Предназначено для указания пути к полю JSON, в котором сохранен идентификатор изображения
maxHeight	Да (при установленном флаге InsertImage)	Поле ввода Предназначено для ввода числа с размером изображения
placeholder	Да (при установленном флаге InsertImage)	Поле ввода Предназначено для ввода текста, который должен быть заменен на форме при вставке изображения

На вкладке «API» в поле «Property Name» должен быть указан путь к полю (или узлу), в котором должна быть сохранена информация о сформированном файле с учетом следующего:

- при установленном на вкладке «Data» флаге «Only ID in JSON» указывается путь до конечного поля, куда будет сохранен идентификатор;

– при снятом на вкладке «Data» флаге «Only ID in JSON» указывается путь до узла, содержащего набор полей для хранения файла (см. п. 4.6.6.5.1)

4.6.7 Примеры js-кода, используемого в настройках форм задач

4.6.7.1 Скрытие / отображение поля по значению поля

Указывается в поле «Advanced Conditions» на вкладке «Conditional» любого компонента.

Необходимо обязательно указывать префикс «data» перед «JSONpath» элемента, если используется единичное значение, и префикс «row», если устанавливается зависимость от элемента того же массива.

Например, отображать поле только, если значение поля – true:

```
show = (!data.document.specific.KS2 || data.document.specific.KS2.inspectionRequired == true).
```

Например, отображать поле только, если значение поля не совпадает с указанным:

```
show = (!data.inspection.purposeInspection || data.inspection.purposeInspection.code !== 'qualityMonitoring').
```

Например, отображать только, если значение поля совпадает с одним из указанных;

```
show = (!data.protocolAP.status || data.protocolAP.status.name == 'В МГЧН') || (!data.protocolAP.status || data.protocolAP.status.name == 'В суде').
```

4.6.7.2 Скрытие таблицы с комментариями до момента появления значений во вложенных полях

Код указывается в поле «Advanced Conditions» на вкладке «Conditional» компонента «Data Grid».

Необходимо обязательно указывать префикс «data» перед «JSONpath» элемента.

Например, отображать таблицу только при наличии комментариев в массиве *light.requestStages.analysis.analysis.notes*:

```
if (data.light.requestStages.analysis.analysis.notes) {  
  show =  
  (JSON.stringify(data.light.requestStages.analysis.analysis.notes).indexOf("remarkDate") >  
  0)  
} else {  
  show = false  
}
```

4.6.7.3 Вывод времени без секунд и миллисекунд

Код указывается в поле «Calculated Value» на вкладке «Data».

Необходимо обязательно указывать префикс «data» или «row» (в случае если используется данное условие для нескольких полей) перед JSONpath элемента, например:

```
value = row.caution.decision.startTime.substr(0, 5).
```

4.6.7.4 Скрытие поля, если оно пустое, отображение если не пустое

Код указывается в поле «Advanced Conditions» на вкладке «Conditional» любого

компонента.

Необходимо обязательно указывать префикс «data» перед «JSONpath» элемента, например:

```
if(!data.visit.isvSmrID || 0 === data.visit.isvSmrID.length){  
  show = false;  
} else {  
  show = true;  
}
```

Если поле имеет глубокую вложенность, то отдельно прописываем каждый элемент в пути, например:

```
if(!data.visit.motiv ||!data.visit.motiv.checkOrg ||!data.visit.motiv.checkOrg.documentID || 0  
=== data.visit.motiv.checkOrg.documentID.length){  
  show = false;  
} else {  
  show = true;  
}
```

4.7 Создание карточки просмотра объекта в Конструкторе карточек

4.7.1 Реестр карточек

Работа с реестром карточек осуществляется только на площадке разработки.

Размещение раздела с реестром карточек: в разделе «Настройки» - «Конструкторы» - «Карточки».

Для вывода настроенных карточек на другие площадки необходимо использовать функционал частичной выгрузки справочников: Имя – справочник FlexCards.

Если приложение для системы не развернуто, то вместо таблицы со списком карточек, отобразится информация «Ошибка при инициализации приложения» (Рисунок 167).

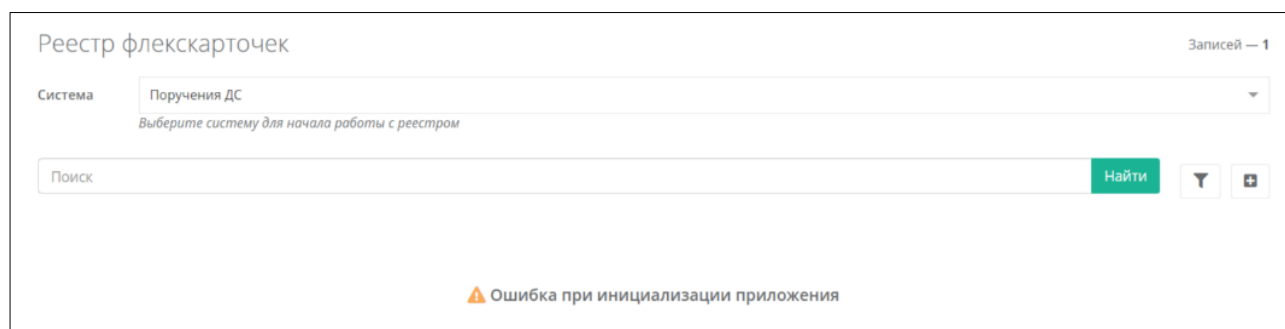


Рисунок 167 – Информация об ошибке

При развернутом приложении отобразится таблица со списком карточек с возможностью создания новой карточки (Рисунок 168).

Код карточки	Наименование карточки	Подсистема	Тип документа	Действие
SYS_MANAGEMENT_CONTRACTOR_CARD	Заявка на регистрацию представителя подрядной организации (для администраторов)	SYS_MANAGEMENT	SYS_MANAGEMENT_CONTRACTOR	
SYS_MANAGEMENT_CONTRACTOR_MYCARD	Заявка на регистрацию представителя подрядной организации (для подрядчиков)	SYS_MANAGEMENT	SYS_MANAGEMENT_CONTRACTOR	

Рисунок 168 – Таблица со списком карточек

Описание функциональных элементов списка карточек приведено ниже (Таблица 69).

Таблица 69 – Описание функциональных элементов списка карточек

Иконка	Действие
	Переход к редактированию карточки
	Установление связи с композитным объектом
	Выгрузка JSON с настройками карточки
	Загрузка JSON с настройками карточки
	Загрузка JSON с тестовыми данными для настройки карточки (используется в режиме предпросмотра карточки)

4.7.2 Создание новой карточки

Для создания новой карточки необходимо нажать на значок . Откроется форма создания новой карточки (Рисунок 169)

Рисунок 169 – Форма создания новой карточки

Код карточки должен быть уникален среди других карточек, рекомендуется именован по маске:

кодсистемы_краткийкодподсистемы_краткийкодтипадокумента_уникальныйкодкарточки.

4.7.3 Редактор карточек

Создание карточек в редакторе карточек (Рисунок 170) включает следующие действия:

- выбор типа раскладки элементов в карточке;
- размещение элементов отображения данных в карточке;
- связывание элементов отображения с атрибутами документа.

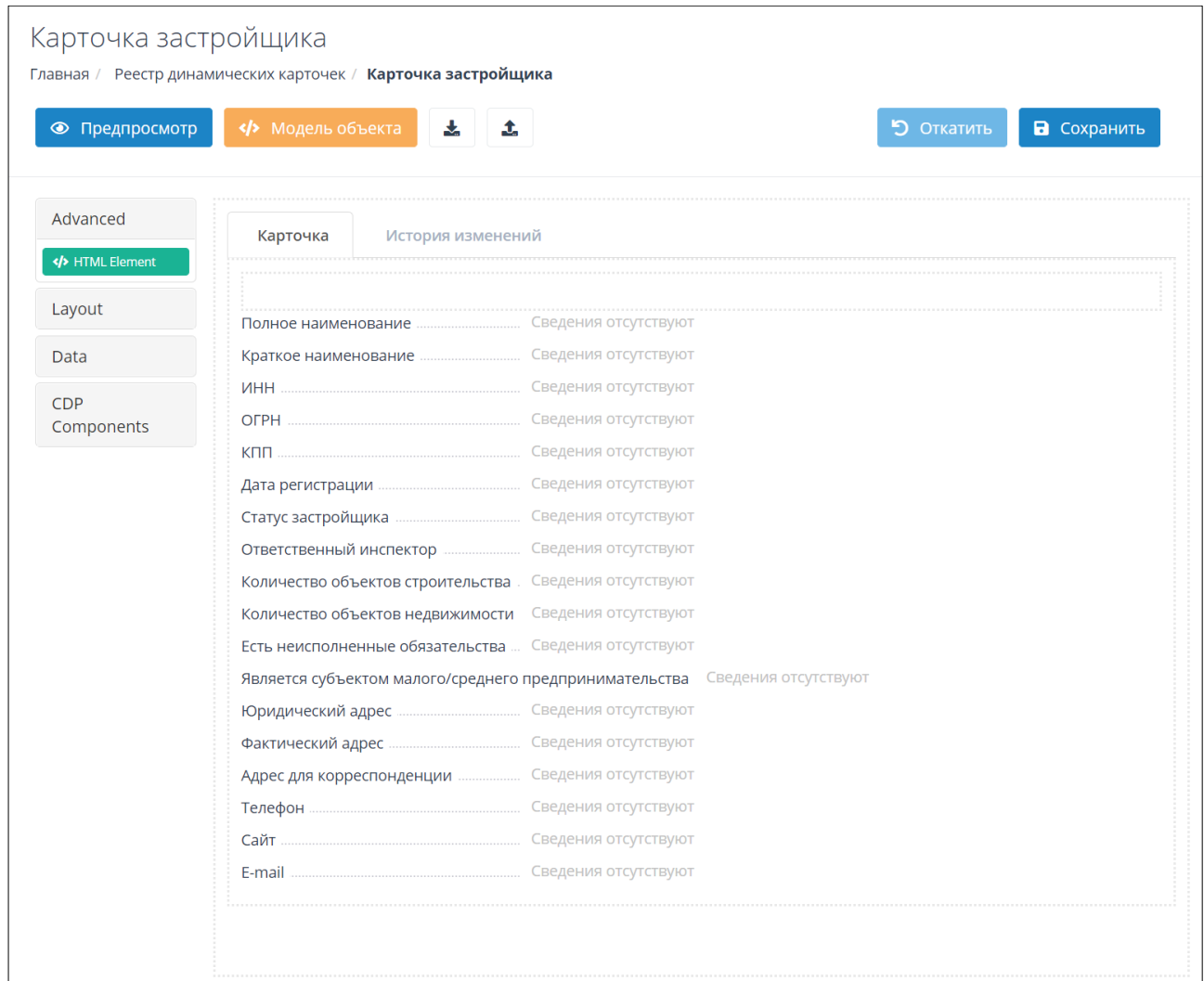


Рисунок 170 – Редактор карточек

Для установления связи в редакторе карточек обеспечивается возможность просмотра модели объекта (Рисунок 171) и копирования в буфер обмена пути в JSON от корня объекта до выбранного элемента. Скопированный в буфер обмена путь до элемента необходимо вставить в поле «Property Name» на вкладке «API» редактируемого элемента карточки.

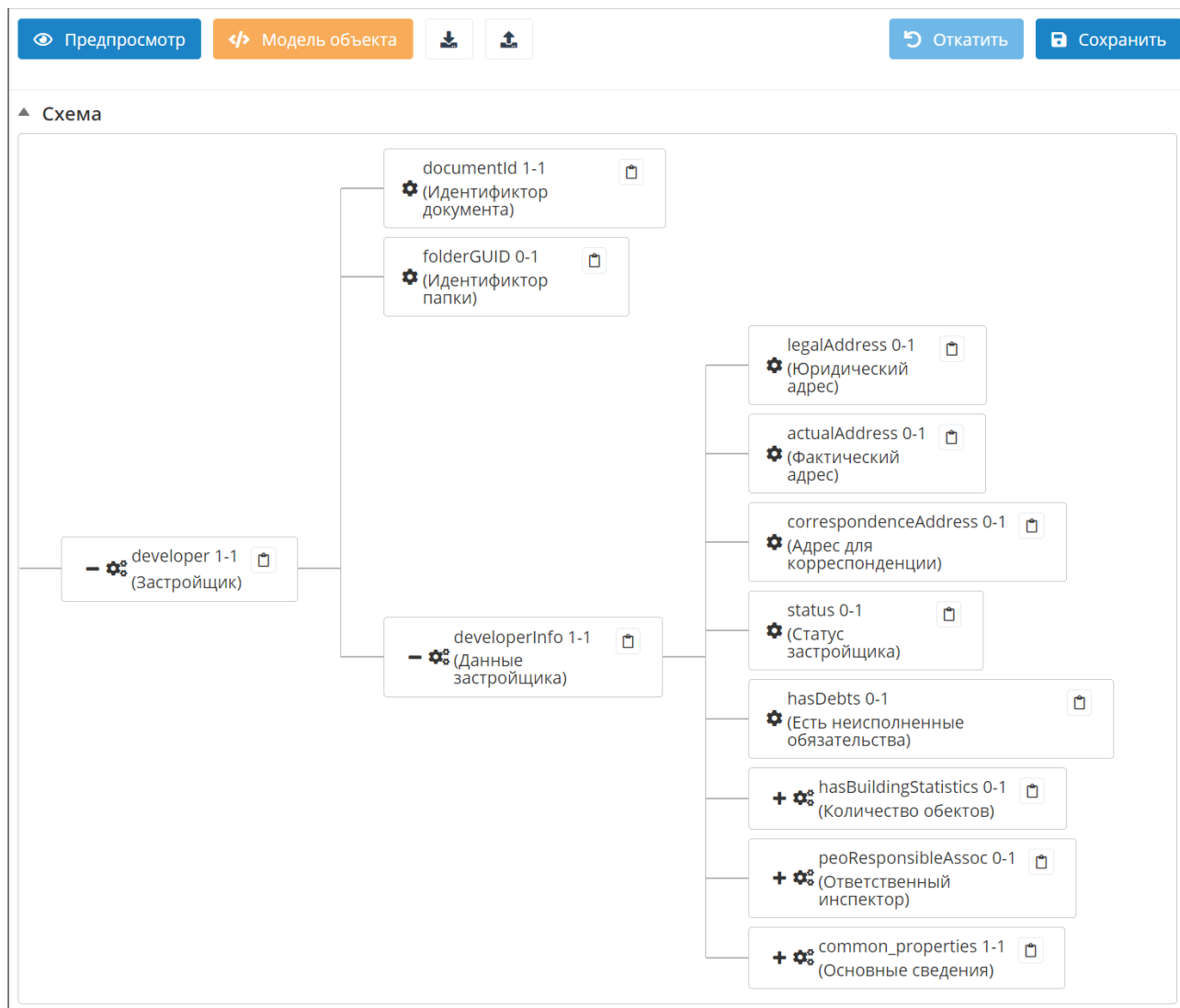


Рисунок 171 – Просмотр модели объекта

В процессе разработки карточки можно включить режим предварительного просмотра, в котором в полях карточки будут отображаться данные из JSON с тестовыми данными для настройки карточки (Рисунок 172). Отредактировать и загрузить новый JSON с данными для тестирования можно в реестре карточек.

Предварительный просмотр карточки «Карточка застройщика»

Карточка История изменений

Полное наименование ООО Рога и Копыта
 Краткое наименование ООО РИК
 ИНН 2263490398
 ОГРН 1090396498125
 КПП 822245747
 Дата регистрации 03.11.2020
 Статус застройщика Действующий
 Ответственный инспектор ООО Ромашка, Иванов Иван Иванович (Ведущий специалист, ivanov@pochta.ru)
 Количество объектов строительства . 100
 Количество объектов недвижимости 20
 Есть неисполненные обязательства ... Да
 Является субъектом малого/среднего предпринимательства Да
 Юридический адрес Москва, улица Широкая, дом 7
 Фактический адрес Москва, улица Большая, дом 1
 Адрес для корреспонденции Москва, Просторный проезд, дом 8, строение 1
 Телефон 8-495-123-45-67
 Сайт test.ru
 E-mail test@test.ru

← Назад к редактированию

Рисунок 172 – Предварительный просмотр карточки

4.7.4 Компоненты

4.7.4.1 Advanced

Описание компонента Advanced доступно по ссылке:

<https://help.form.io/userguide/form-building/advanced-components>.

4.7.4.2 HTML Element. Использование html-разметки

Компонент HTML Element предназначен для добавления на форму заголовков, строк-разделителей и прочего контента.

Настройка компонента для добавления строки-разделителя приведена ниже (Рисунок 173).

HTML Element Component
HELP

Display
Data
Validation
API
Conditional
Logic
Layout

Label ⊗

Hide Label ⊗

Label Position ⊗

HTML Tag ⊗

CSS Class ⊗

Attributes ⊗

Attribute	Value	
<input type="text"/>	<input type="text"/>	⊗

+ Add Another

Content ⊗

Refresh On Change ⊗

Placeholder ⊗

Description ⊗

Tooltip ⊗

Error Label ⊗

Custom CSS Class ⊗

Tab Index ⊗

Persistent ⊗

None
 Server
 Client

Multiple Values ⊗

Clear Value When Hidden ⊗

Protected ⊗

Hidden ⊗

Hide Input ⊗

Disabled ⊗

Initial Focus ⊗

Table View ⊗

Always enabled ⊗

Preview

Save
Cancel
Remove

Рисунок 173 – Настройка компонента HTML Element

Настройка компонента для добавления заголовка приведена ниже (Рисунок 174).

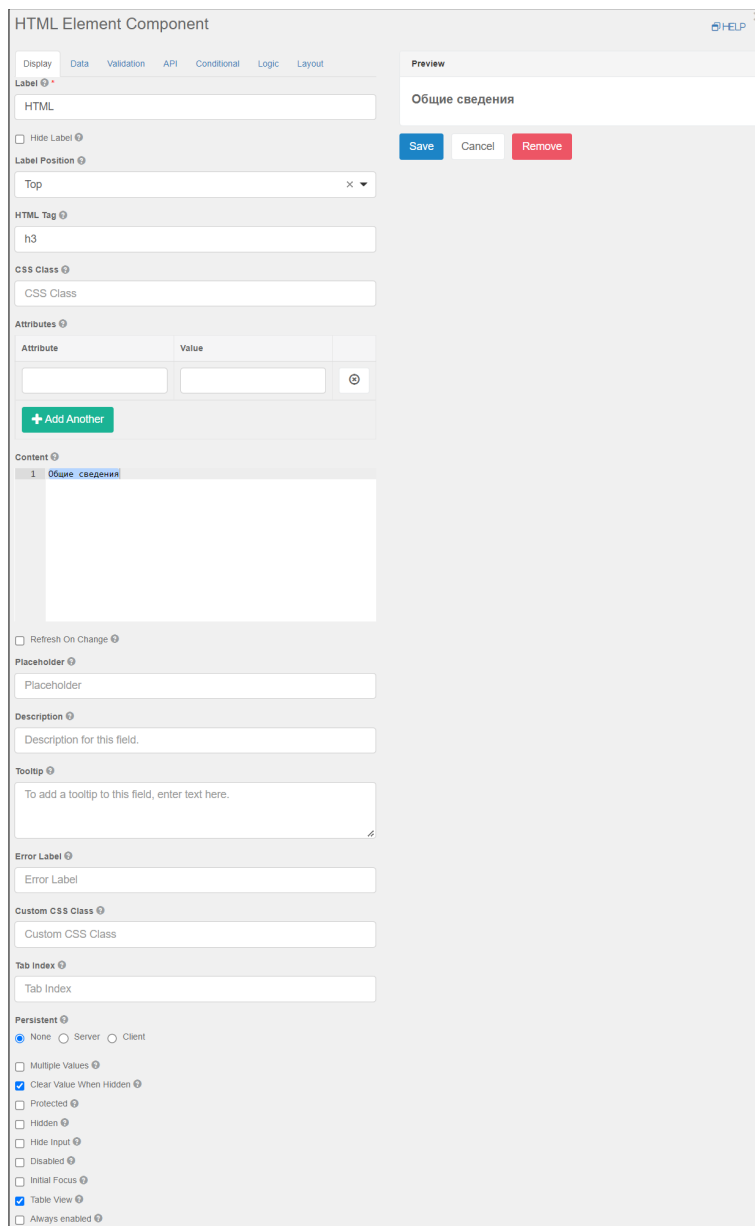


Рисунок 174 – Настройка компонента для добавления заголовка

4.7.4.3 Layout

Описание компонента Advanced доступно по ссылке:

<https://help.form.io/userguide/form-building/layout-components>

4.7.4.4 Breadcrumbs. Хлебные крошки

Компонент предназначен для отображения навигационной цепочки («хлебных крошек») (Рисунок 175).

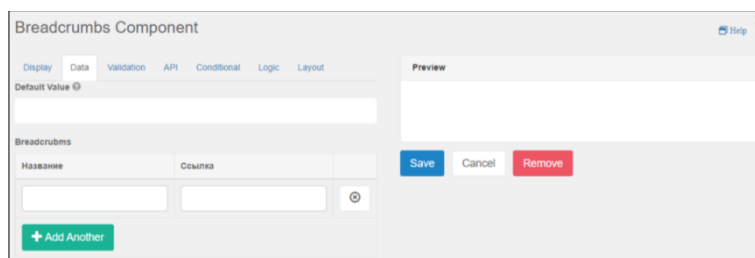


Рисунок 175 – Breadcrumbs компонент

На вкладке «Data» добавлен блок специфических полей компонента Breadcrumbs для указания элементов навигации.

4.7.4.5 Data

Описание компонента Advanced доступно по ссылке: <https://help.form.io/userguide/form-building/data-components>

4.7.4.6 CDP Components

4.7.4.6.1 Info Field. Отображение ранее сохраненных полей

Компонент Info Field предназначен для отображения ранее сохраненных полей JSON в режиме просмотра (Рисунок 176).

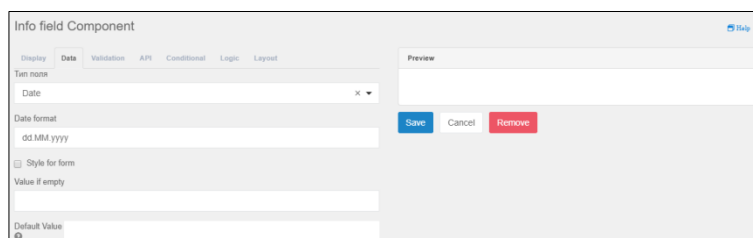


Рисунок 176 – Info Field компонент

На вкладке «Data» добавлено специфическое поле «Тип поля» для работы указанного компонента (Таблица 70).

Таблица 70 – Специфическое поле «Тип поля»

Наименование поля	Обязательность	Описание
Тип поля	Да	Список с единичным выбором Предназначено для определения формата отображаемых данных Доступен выбор из значений: Link Ссылка Date Дата Boolean Логическое Array Массив File Файл Class inline Класс в строку Text Текст Solr Поиск в Solr

Состав полей вкладки «Data» и правила их заполнения определяются значением, выбранным в специфическом поле «Тип поля».

Чтобы обеспечить возможность добавления нескольких ссылок на форму (когда в качестве пути к элементу указывается корневой элемент схемы) для указания пути к элементу JSON вместо поля «Property Name» на вкладке «API» предназначено поле «Значение поля (API)» на вкладке «Data». Правила заполнения нового поля аналогичны

полю «Property Name». Для всех ранее созданных форм перенастройка компонента не требуется.

При этом важно обеспечить, чтобы в поле «Property Name» на вкладке «API» всегда указывалось уникальное значение.

4.7.4.6.1.1 Link Ссылка

При выбранном типе поля «Link» (Рисунок 177) на вкладке «Data» отображаются дополнительно специфические поля (Таблица 71).

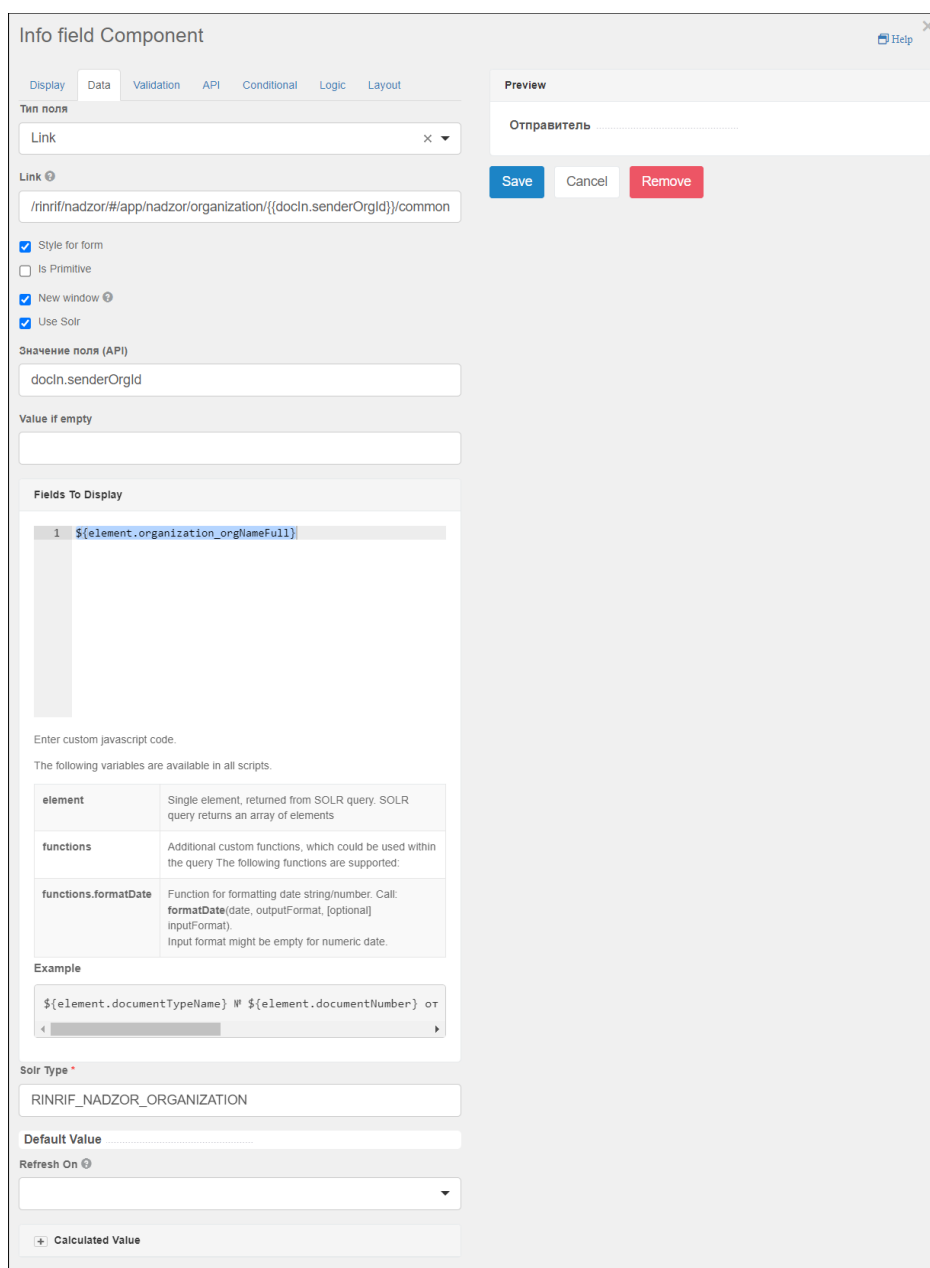


Рисунок 177 – Link ссылка

Таблица 71 – Специфические поля вкладки «Data»

Наименование поля	Обязательность	Описание
Link	Да	Поле ввода Предназначено для указания относительной ссылки

Наименование поля	Обязательность	Описание
		Может включать параметр из JSON документа Указанный параметр должен содержать JsonPath элемента
New window	Нет	Чекбокс При установленном флаге ссылка открывается в новом окне/вкладке (в зависимости от настроек браузера пользователя)
Use Solr	Нет	Чекбокс При установленном флаге отображается набор полей для реализации поиска идентификатора, указанного в поле "Значение поля (API)" среди документов Solr
Fields To Display	—	Отображается и обязательно для заполнения только при установленном флаге Use Solr Предназначен для указания поля или набора полей, которые будут отображаться в качестве значения поля вместо идентификатора связанного документа
Solr Type	—	Отображается и обязательно для заполнения только при установленном флаге Use Solr Предназначен для указания типа документа, среди которых будет осуществлен поиск документа, идентификатор которого указан в поле "Значение поля (API)"

Настройка текста для отображаемой ссылки осуществляется на вкладке «API».

Описанная ниже настройка актуальна, если не установлен флаг «Use Solr», в противном случае текст определяется настройками свойств Solr, описанными выше (Рисунок 178).

The screenshot shows the 'Info field Component' configuration interface. The 'API' tab is selected, displaying the 'Property Name' as 'purchase.number' and a 'Preview' of the field content: 'Номер и дата'. The 'Custom Properties' section is visible, containing a table with the following entries:

Key	Value
url0	"№"
url1	purchase.number
url2	" от "
url3_format	purchase.date
format_url3	dd.MM.yyyy

Рисунок 178 – Link ссылка

В группе полей «Custom Properties» необходимо указать правила формирования ссылки (Таблица 72).

Таблица 72 – Правила формирования ссылки в группе «Custom Properties»

Key	Value
urlN, где N – порядковый номер, начиная с 0	Путь до элемента JSON (<i>pmt.typePmt.name</i>) или константа указанная в кавычках (" <i>om</i> ")
urlN_format, где N – порядковый номер, начиная с 0	Путь до элемента JSON, который должен быть преобразован по определенному правилу (<i>pmt.regDate</i>)
format_urlN, где N – порядковый номер, начиная с 0	Правила преобразования элемента с аналогичным порядковым номером (<i>dd.MM.yyyy</i>)

4.7.4.6.1.2 Date Дата

При выбранном типе поля «Date» (Дата) на вкладке «Data» отображаются дополнительно специфические поля (Рисунок 179, Таблица 73).

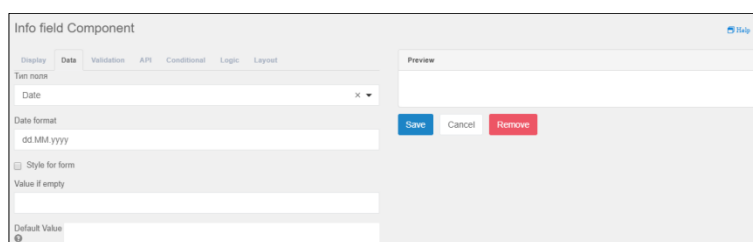


Рисунок 179 – Тип поля «Date»

Таблица 73 – Специфические поля вкладки «Data»

Наименование поля	Обязательность	Описание
Date format	Нет	Поле ввода Предназначено для ввода формата отображения даты
Style for form	Нет	Чекбокс При установленном флаге лейблу присваивается стиль заголовка При установленном флаге лейбл поля на вкладке Display должен быть скрыт
Value if empty	Нет	Поле ввода Предназначено для ввода константы, которая будет отображаться при отсутствии в JSON документа значения поля, указанного на вкладке API

4.7.4.6.1.3 Boolean Логическое

При выбранном типе поля «Boolean» (Логическое) (Рисунок 180) на вкладке «Data» отображаются дополнительно специфические поля (Таблица 74).

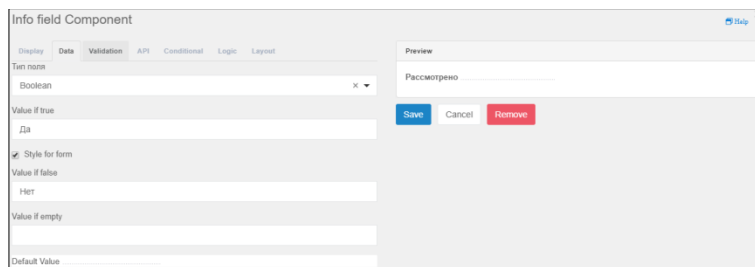


Рисунок 180 – Тип поля Boolean

Таблица 74 – Специфические поля вкладки «Data»

Наименование поля	Обязательность	Описание
Value if true	Нет	Поле ввода Предназначено для ввода значения, которое отображается вместо true, полученного в поле, указанном на вкладке API
Value if false	Нет	Поле ввода Предназначено для ввода значения, которое отображается вместо false, полученного в поле, указанном на вкладке API
Style for form	Нет	Чекбокс При установленном флаге лейблу присваивается стиль заголовка При установленном флаге лейбл поля на вкладке Display должен быть скрыт
Value if empty	Нет	Поле ввода Предназначено для ввода константы, которая будет отображаться при отсутствии в JSON документа значения поля, указанного на вкладке API

4.7.4.6.1.4 Array Массив

При выбранном типе поля «Array» (Массив) (Рисунок 181) на вкладке «Data» отображаются дополнительно специфические поля (Таблица 75).

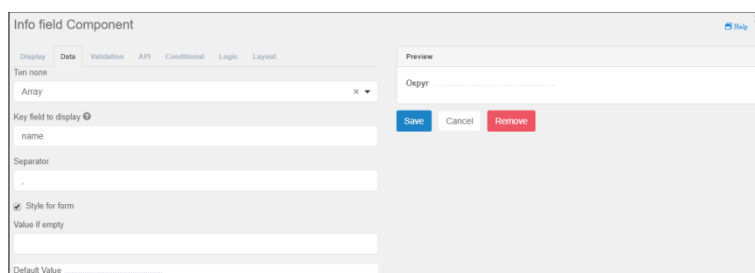


Рисунок 181 – Тип поля Array

Таблица 75 – Специфические поля вкладки «Data»

Наименование поля	Обязательность	Описание
Key field to display	Да	Поле ввода Предназначено для указания поля для отображения в

Наименование поля	Обязательность	Описание
		рамках узла, указанного на вкладке API
Separator	Да	Поле ввода Предназначено для указания символа-разделителя для элементов массива (для переноса на другую строку использовать разделитель)
Style for form	Нет	Чекбокс При установленном флаге лейблу присваивается стиль заголовка При установленном флаге лейбл поля на вкладке Display должен быть скрыт
Value if empty	Нет	Поле ввода Предназначено для ввода константы, которая будет отображаться при отсутствии в JSON документа значения поля, указанного на вкладке API

Для корректной работы компонента в поле «Property Name» на вкладке «API» должен быть указан путь к узлу, содержащему в себе вложенные элементы code и name.

4.7.4.6.1.5 File Файл

При выбранном типе поля «File» (Файл) вкладке «Data» отображаются дополнительно специфические поля (Таблица 76).

Таблица 76 – Специфические поля вкладки «Data» поля «File»

Наименование поля	Обязательность	Описание
Style for form	Нет	Чекбокс При установленном флаге лейблу присваивается стиль заголовка При установленном флаге лейбл поля на вкладке Display должен быть скрыт
Value if empty	Нет	Поле ввода Предназначено для ввода константы, которая будет отображаться при отсутствии в JSON документа значения поля, указанного на вкладке API
File Type	Да	Поле ввода Предназначено для ввода типа файла, с которым он был сохранен в файловое хранилище
Only ID	Нет	Чекбокс При установленном на вкладке Data флаге Only ID на вкладке API необходимо указывать путь до поля с массива, предназначенного для хранения идентификатора сохраненного файла, при неустановленном - путь до узла, содержащего

Наименование поля	Обязательность	Описание
		расширенную информацию о файле (см. описание компонента File Manager)
Show Sign Button	Нет	Чекбокс При установленном флаге отображается ссылка для подписания файла электронной подписью
Sign this file	Нет	Чекбокс При установленном флаге при завершении задачи по кнопке Complete Button будет осуществлено подписание выбранного файла

4.7.4.6.1.6 Text Текст

При выбранном типе поля «Text» (Текст) (Рисунок 182) на вкладке «Data» отображаются дополнительно специфические поля (Таблица 77).

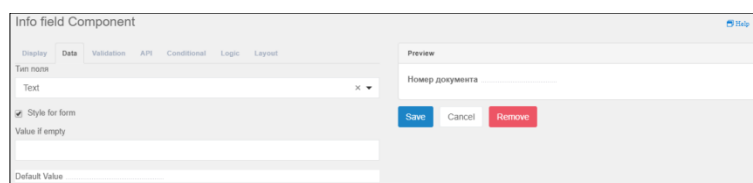


Рисунок 182 – Тип поля Text

Таблица 77 – Специфические поля вкладки «Data» поля «Text»

Наименование поля	Обязательность	Описание
Style for form	Нет	Чекбокс При установленном флаге лейблу присваивается стиль заголовка При установленном флаге лейбл поля на вкладке Display должен быть скрыт
Value if empty	Нет	Поле ввода Предназначено для ввода константы, которая будет отображаться при отсутствии в JSON документа значения поля, указанного на вкладке API

Для использования конкатенации нескольких полей в группе полей «Custom Properties» необходимо указать правила формирования конкатенации.

В поле «Property Name» должен быть указан корневой элемент JSON документа (Таблица 78).

Таблица 78 – Корневые элементы JSON документа

Key	Value
textN, где N – порядковый номер, начиная с 0	Путь до элемента JSON (<i>pmt.typePmt.name</i>) или константа указанная в кавычках (" <i>om</i> ")
textN_format, где N – порядковый номер, начиная с 0	Путь до элемента JSON, который должен быть преобразован по определенному правилу (<i>pmt.regDate</i>)
format_textN, где N – порядковый номер, начиная с 0	Правила преобразования элемента с аналогичным порядковым номером (<i>dd.MM.yyy</i>)

4.7.4.6.1.7 Solr Поиск в Solr

Тип поля «Solr» предназначен для поиска в Solr сохраненного идентификатора связанного документа и отображения атрибутов найденного документа.

При выбранном типе поля «Solr» (Рисунок 183) на вкладке «Data» отображаются дополнительно специфические поля (Таблица 79).

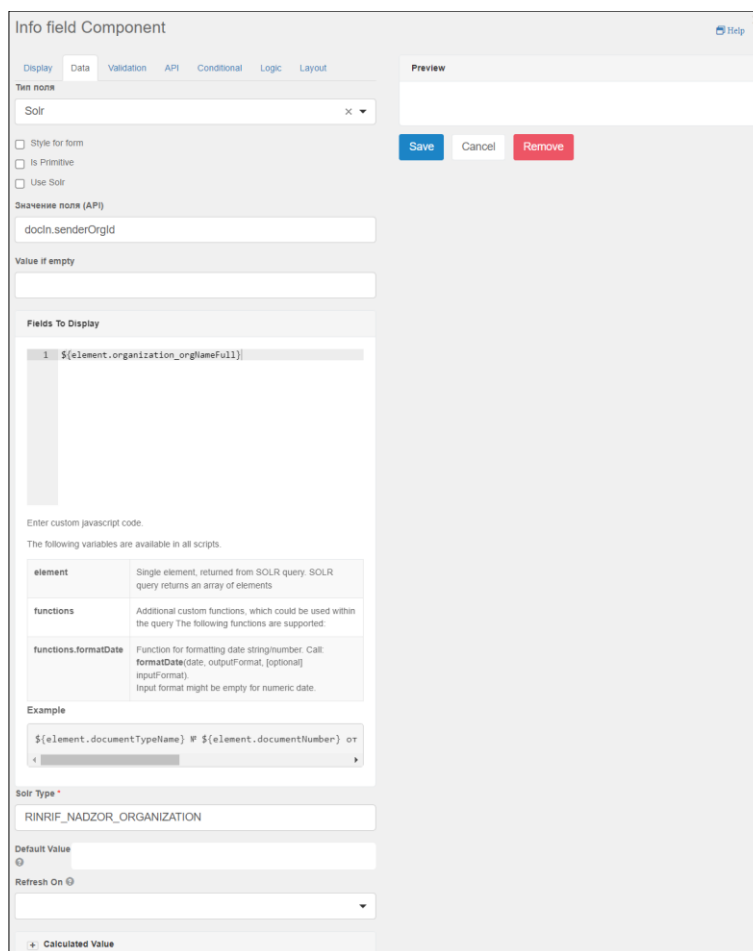


Рисунок 183 – Тип поля Solr

Таблица 79 – Специфические поля вкладки «Data» поля «Solr»

Наименование поля	Обязательность	Описание
Fields To Display	—	Отображается и обязательно для заполнения только при установленном флаге Use Solr

Наименование поля	Обязательность	Описание
		Предназначен для указания поля или набора полей, которые будут отображаться в качестве значения поля вместо идентификатора связанного документа
Solr Type	—	Отображается и обязательно для заполнения только при установленном флаге Use Solr Предназначен для указания типа документа, среди которых будет осуществлен поиск документа, идентификатор которого указан в поле "Значение поля (API)"

4.7.4.6.2 Approval List View. Просмотр листа согласования

Компонент предназначен для просмотра листа согласования (Рисунок 184).

№ пп	ФИО и должность	Срок согласования	Результат согласования
▶ 1	Иванов Иван Иванович Специалист	12.12.2019	Согласовано 11.12.2019 00:00
▶ 2	Иванов Иван Иванович Специалист		Утверждено
▶ 1	Иванов Иван Иванович Специалист		Согласовано
▶ 1	Петров Константин Романович		Согласовано
▶ 5	Петров Константин Романович		Утверждено
▶	Предыдущий цикл согласования		
▶	Предыдущий цикл согласования		
▶	Предыдущий цикл согласования		

Рисунок 184 – Просмотр листа согласования

Для корректной работы компонента Approval List View структура данных для листа согласования должна соответствовать одной из структур, требования к которым указаны в п. 4.6.6.5.8.

На вкладках «Display» и «Data» добавлены специфические поля для работы данного компонента (Рисунок 185, Таблица 80).

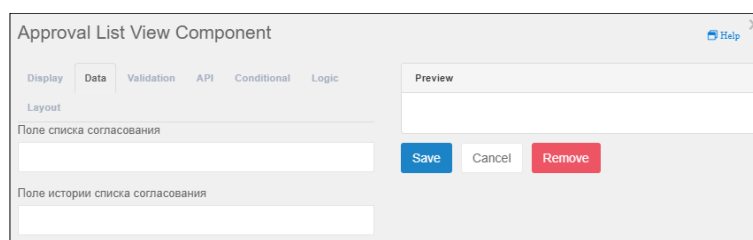


Рисунок 185 – Специфические поля вкладки «Data»

Таблица 80 – Специфические поля вкладок «Display» и «Data»

Наименование поля	Обязательность	Описание
Expand Current Cycle	Нет	Флаг Отображать текущий цикла согласования сразу развернутым

Наименование поля	Обязательность	Описание
Поле списка согласования	Да	Поле ввода Предназначено для указания пути к узлу JSON документа с комплексным типом approvalCicleType, содержащем сведения о текущем цикле согласования
Поле истории списка согласования	Да	Поле ввода Предназначено для указания пути к узлу JSON документа с комплексным типом approvalCicleType, содержащем сведения о прошедших циклах согласования (массив)
Only ID in JSON	Нет	Флаг Устанавливается при использовании сокращенной структуры хранения файла

Заполнение поля «Property Name» на вкладке «API» не требуется.

4.7.4.6.3 JSON Edit. Редактор и просмотр JSON

Компонент предназначен для отображения стандартного компонента просмотра и редактирования JSON документа (Рисунок 186).

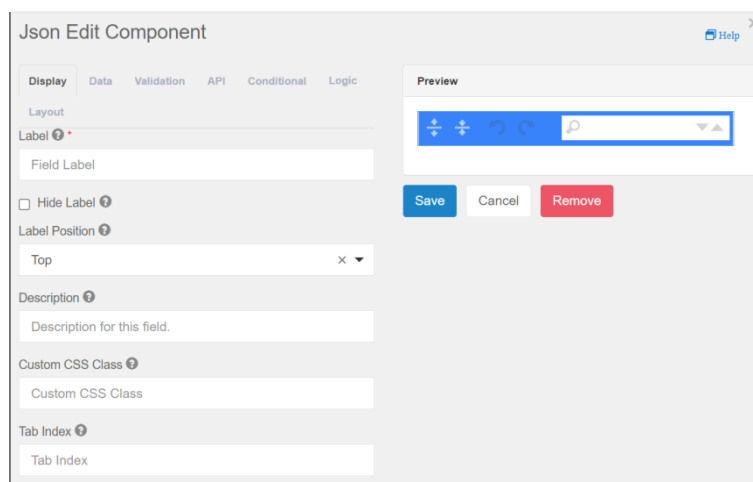


Рисунок 186 – JSON Edit компонент

На вкладке «API» в поле «Property Name» должен быть указан корневой элемент документа.

Для возможности сохранения изменений, внесенных в JSON, необходимо установить флаг «Показывать кнопку Сохранить» на вкладке «Data».

4.7.4.6.4 Document Log. История изменений

Компонент предназначен для отображения стандартного компонента с историей изменений по документу (Рисунок 187).

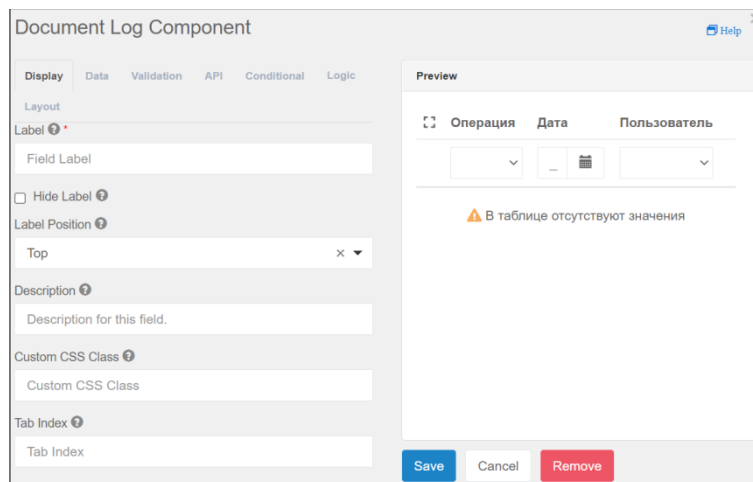


Рисунок 187 – Document Log компонент

4.7.4.6.5 Process History. История бизнес-процесса

Компонент предназначен для отображения стандартного компонента с историей бизнес-процессов (Рисунок 188).

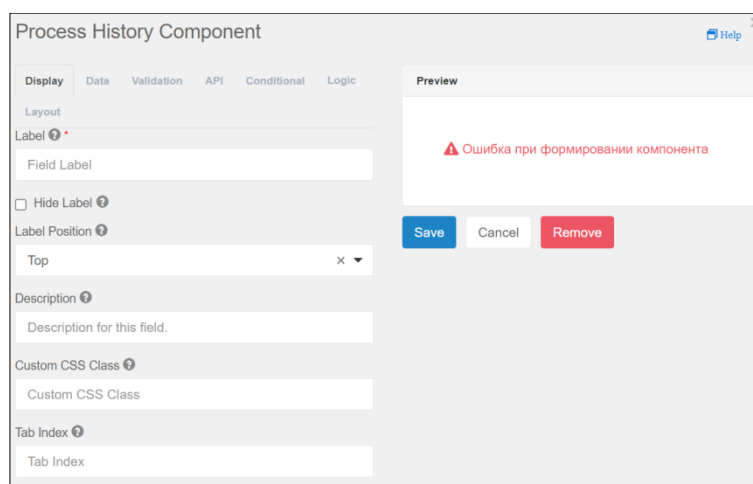


Рисунок 188 – Process History компонент

На вкладке «Data» задаются следующие специфические параметры:

- Process Prefix – поле ввода для выбора префикса процесса, который должен отображаться на вкладке;
- ID Variable Name – поле ввода для указания ключа переменной с идентификатором (по умолчанию: «EntityIdVar»).

4.7.4.6.6 Link Button. Кнопка со ссылкой

Компонент предназначен для добавления на форму кнопок с абсолютными и относительными ссылками.

Внешний вид кнопки (цвет, иконка, всплывающая подсказка) настраиваются на вкладке «Display» (Рисунок 189).

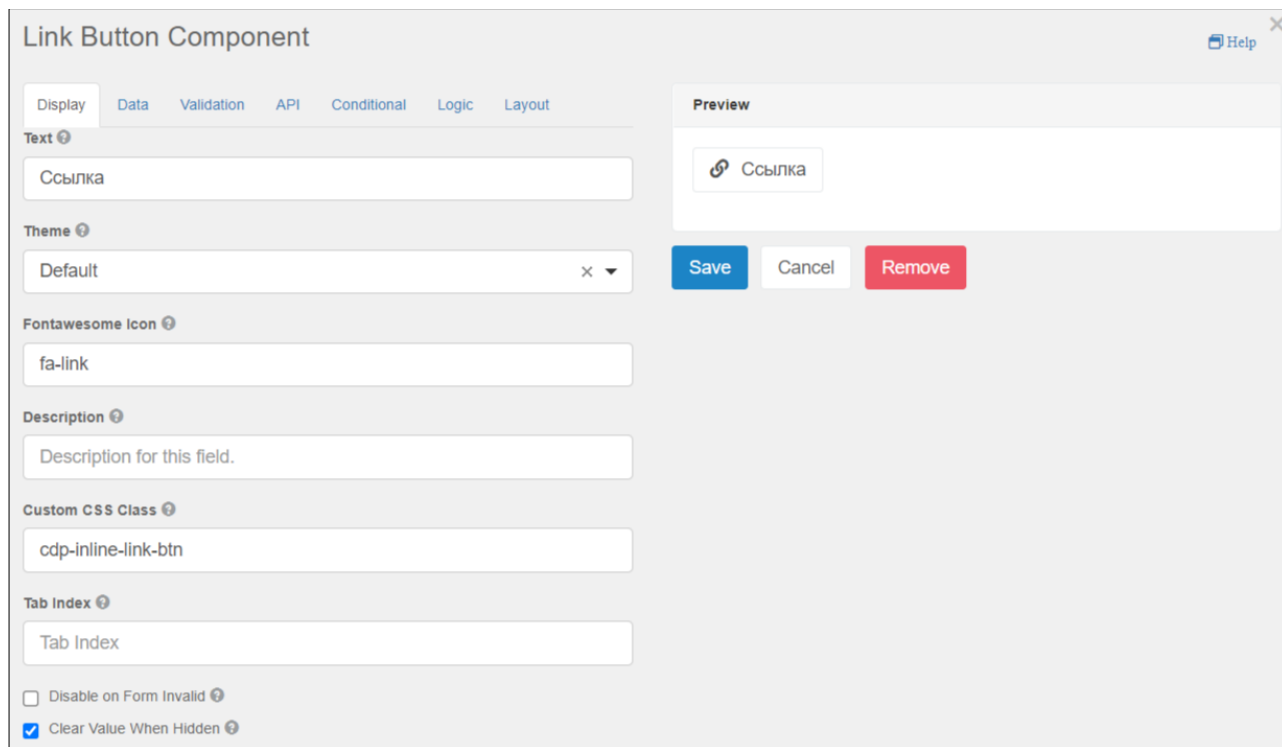


Рисунок 189 – Вкладка «Display». Настройки вида кнопки

Ссылка, открываемая по кнопке, указывается на вкладке «Data» (Рисунок 190).

Для использования в ссылке переменной из JSON путь к переменной необходимо указывать в двойных фигурных скобках.

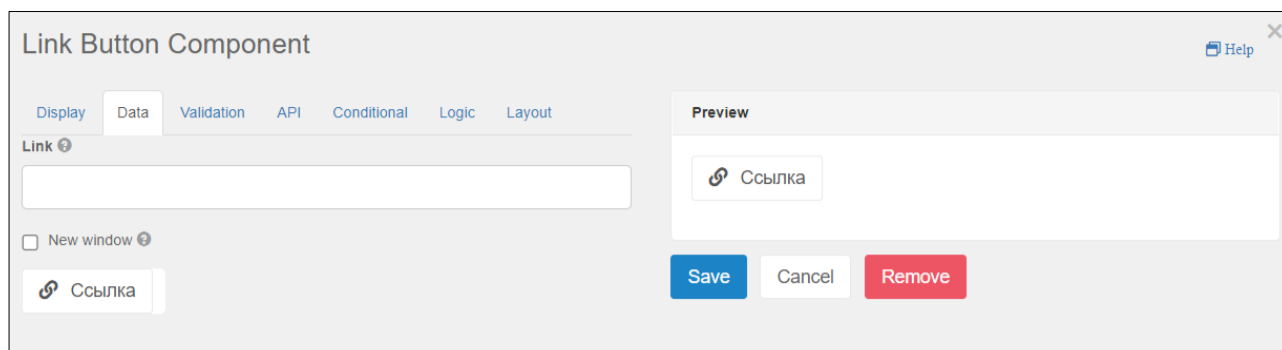


Рисунок 190 – Вкладка «Data». Настройки ссылки

4.7.4.6.7 Status badge. Статус в цветной плашке

Компонент предназначен для добавления в карточку значения поля в цветной плашке (Рисунок 191).

Цвет плашки должен быть определен кодом в справочнике, значение поля в JSON должно совпадать со значением поля в справочнике.

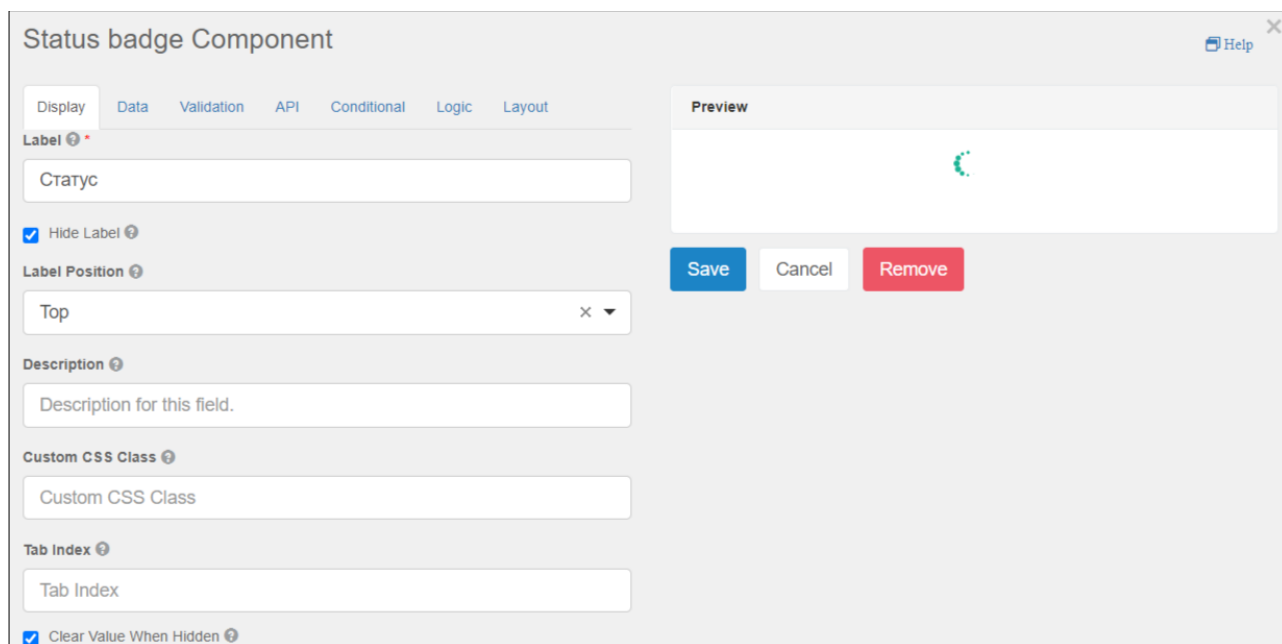


Рисунок 191 – Status badge компонент

В поле «Dictionary Name» вкладки «Data» необходимо выбрать справочник, содержащий список статусов и соответствующих им цветов (выбор осуществляется из платформенных справочников и справочников системы, к которой принадлежит бизнес-процесс).

В поле «Value Field» необходимо указать поле справочника, с которым сравнивается значение поля JSON с вкладки «API».

В поле «Color Field» необходимо указать поле справочника, в котором указан код цвета, соответствующий статусу (Рисунок 192)

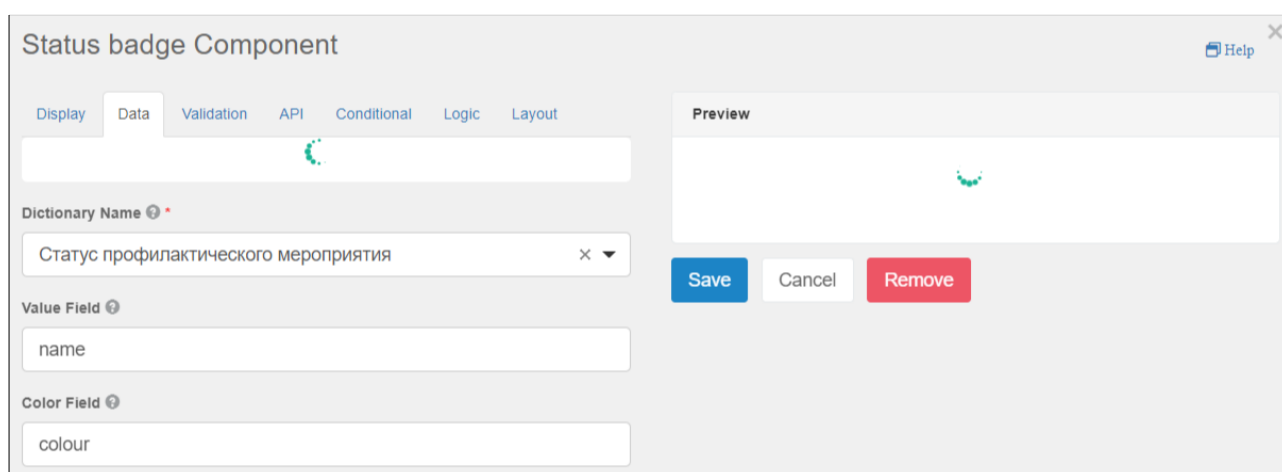


Рисунок 192 – Вкладка «Data». Настройки полей

4.7.4.6.8 Project-PD. Загрузка проектной и рабочей документации

Компонент предназначен для отображения блока загрузки проектной и рабочей документации (Рисунок 193).

Для корректной работы структура узла для сохранения данных должна соответствовать комплексному типу «folderType» (см. п. 4.6.6.5.8).

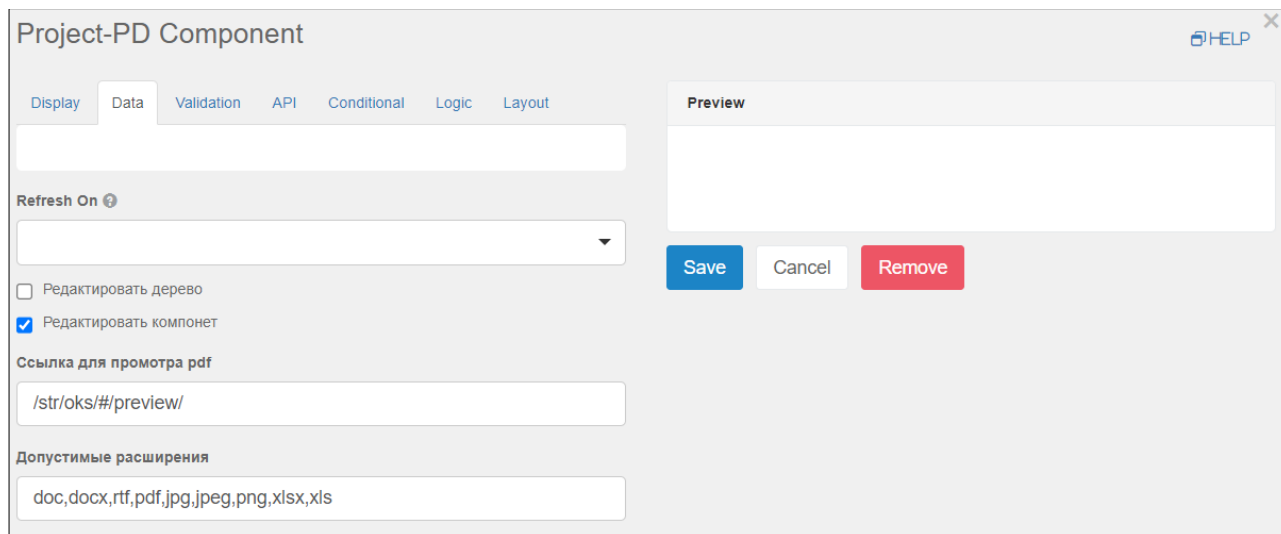


Рисунок 193 – Project-PD компонент. Установлен флаг «Редактировать компонент»
 На вкладку «Data» компонента добавлены специфические настройки (Таблица 81).

Таблица 81 – Специфические настройки вкладку «Data»

Поле	Описание	Примечание
Редактировать дерево	При установленном флаге доступно изменение дерева папок	Не применяется для карточек
Редактировать компонент	При установленном флаге доступна загрузка / удаление файлов	Не применяется для карточек
Ссылка для просмотра pdf	—	—
Допустимые расширения	Список допустимых для загрузки расширений файлов с разделителем запятой	Не применяется для карточек

4.7.4.6.9 Pgu Events Table. Отображение статусов взаимодействия с ПГУ

Компонент предназначен для отображения статусов взаимодействия с ПГУ (Рисунок 194).

Извещение о начале СМР

№ regNumber_80 от 28.02.2023

Услуга оказана. Решение положительное

Основная информация | Документы | Результат | Внешние системы | Процессы | История изменений | </>

Статусы ПГУ | Запросы в БР

Дата	Тип события	Статус	Код статуса	Комментарий
23.03.2023 19:18:04	Отправка статуса	Услуга оказана. Решение положительное	1075	По Вашему заявлению подготовлено и оформлено Разрешение на строительство № 09-3-1805/22-(0)-0 от 26.10.2022
02.03.2023 10:25:11	Отправка статуса	Формирование результата	1052	
02.03.2023 10:25:11	Отправка статуса		10190	(Окончание периода, в течение которого заявитель имеет возможность отозвать заявление) направляется до статуса 1052
02.03.2023 10:23:00	Отправка статуса	Формирование результата	1052	
02.03.2023 10:23:00	Отправка статуса		10190	(Окончание периода, в течение которого заявитель имеет возможность отозвать заявление) направляется до статуса 1052

Рисунок 194 – Пример отображения статусов взаимодействия с ПГУ

Для настройки компонента необходимо выбрать справочник со статусами на вкладке «Data» и указать путь к полю, где хранится «serviceNumber» в поле «Property Name» на вкладке «API».

4.7.5 Ссылка на карточку

Ссылка на карточку для использования в реестрах формируется по следующему алгоритму:

`/код_приложения}/flex/#/app/flexcards/{полный_код_типа_документа}/{код_карточк и}/{идентификатор_документа}/view?systemCode=код_системы`

4.8 Создание витрины на Конструкторе витрин

До настройки витрины должна быть выполнена индексация данных в SOLR через приложение *search*.

4.8.1 Панель администрирования конструктора витрин

Панель администрирования конструктора витрин содержит следующие компоненты:

- поиск по содержимому панели администрирования;
- список витрин с возможностью удаления, редактирования, копирования и создания новой витрины;
- редактор витрин (обеспечивает заполнение данных витрины при создании новой витрины, копировании, редактировании, использует справочник «ShowcaseBuilder»);
- предварительный просмотр витрины.

4.8.2 Список витрин

Список витрин включает все витрины, используемые в системе (Рисунок 195).

Конструктор витрин

Название витрины, документа, системы или подсистемы				Найти
Система	Подсистема	Документ	Витрина	+
1 Платформа	Регистрация пользователей	Заявка на регистрацию представителя подрядной организации	Заявки на регистрацию представителей подрядных организаций (мои заявки)	
2 Платформа	Регистрация пользователей	Заявка на регистрацию представителя подрядной организации	Заявки на регистрацию представителей подрядных организаций (все заявки)	
3 Геоинформационная платформа ReGIS	REGIS META	Источник геоданных	Источник геоданных	
4 DEMO приложение	ReConstructor	Закупка офисного оборудования	Закупка офисного оборудования (мои заявки)	
5 Система строительства	Административное управление	Шаблон совещания	Шаблоны совещаний	
6 DEMO приложение	ReConstructor	Закупка офисного оборудования	Закупка офисного оборудования (все заявки)	

Рисунок 195 – Пример списка витрин

Функциональные кнопки списка витрин приведены ниже (Таблица 82).

Таблица 82 – Функциональные кнопки списка витрин

Кнопка	Действие
	Создание новой витрины
	Редактирование, копирование, удаление витрины

4.8.3 Редактор витрины

Редактор витрины содержит набор вкладок для заполнения данных витрины (Рисунок 196).

Конструктор витрин >

Редактирование витрины Закупка офисного оборудования (мои заявки)






Главная Настройки Поля (5) Фильтры (5) Колонки таблицы (4) Карточка Сортировки Группировки Ссылочность словарей

Код системы	DEMO
Наименование системы	DEMO приложение
Код подсистемы	DEMO_RECONSTRUCTOR
Наименование подсистемы	ReConstructor
Код документа	DEMO_RECONSTRUCTOR_PURCHASE
Наименование документа	Закупка офисного оборудования
Код витрины	demo_reconstructor_purchase_my
Наименование витрины	Закупка офисного оборудования (мои заявки)
Код	demo_reconstructor_purchase_my

Рисунок 196 – Пример редактора витрин

Функциональные кнопки редактора витрин приведены ниже (Таблица 83).

Таблица 83 – Функциональные кнопки списка витрин

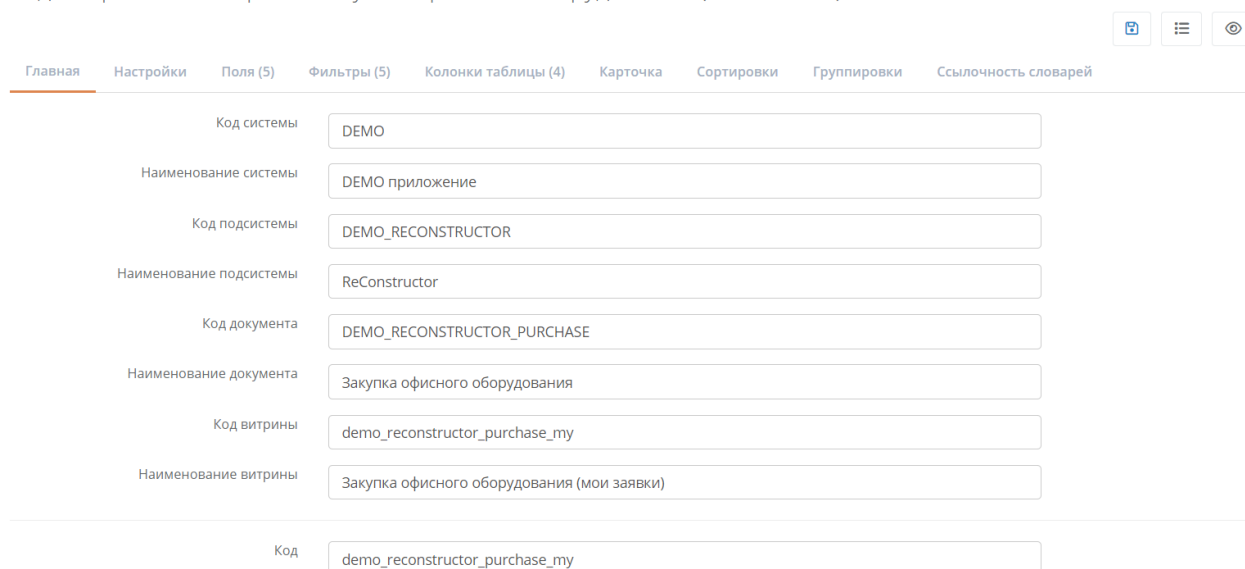
Кнопка	Действие
	Сохранение изменений
	К списку витрин
	Предварительный просмотр




4.8.3.1 Вкладка «Главная»

Вкладка «Главная» используется для заполнения параметров системы: кода и наименования системы, кода и наименования подсистемы, кода и наименования документа, кода и наименования витрины (Рисунок 197).

Конструктор витрин >

Редактирование витрины Закупка офисного оборудования (мои заявки)



Кнопки:   

Навигация: Главная | Настройки | Поля (5) | Фильтры (5) | Колонки таблицы (4) | Карточка | Сортировки | Группировки | Ссылочность словарей

Код системы: DEMO

Наименование системы: DEMO приложение

Код подсистемы: DEMO_RECONSTRUCTOR

Наименование подсистемы: ReConstructor

Код документа: DEMO_RECONSTRUCTOR_PURCHASE

Наименование документа: Закупка офисного оборудования

Код витрины: demo_reconstructor_purchase_my

Наименование витрины: Закупка офисного оборудования (мои заявки)

Код: demo_reconstructor_purchase_my

Рисунок 197 – Вкладка «Главная»

4.8.3.2 Вкладка «Настройки»

Вкладка «Настройки» используется для заполнения основных параметров витрины (Рисунок 198).

ГлавнаяНастройкиПоля (5)Фильтры (5)Колонки таблицы (4)КарточкаСортировкиГруппировкиСсылочность словарей

Тема green aero blue gray orange pink

Путь к данным витрины

Заголовок

Html-шаблон заголовка

Показывать заголовок

Html-шаблон кнопок

Представление результатов поиска по умолчанию Таблица Карточки

Табличная сортировка

Автономная сортировка

Табличные фильтры

Автономные фильтры

Возможность выбора колонок

Возможность переключения результатов поиска

Возможность переключения типов сортировки

Возможность переключения типов фильтрации

Группировка

Нумерация колонок

Нумерация строк

Отключение позиционирования заголовка по центру

Отключение общего поиска

Заполнитель для общего поиска

Выгрузка в excel

Количество элементов на странице 10 20 50 100

Возможность менять количество элементов на странице

Фильтры переданные при инициализации витрины всегда будут применяться

Новый solr

Отключение выбора настроек

Дополнительные типы документов

Рисунок 198 – Вкладка «Настройки»

Заполнение основных параметров витрины осуществляется с учетом следующего:

- Тема – цветовое оформление витрины;
- Путь к данным витрины – url к данным solr. В случае, если в качестве пути к данным витрины указан /search/v1/solr/query необходимо установить флаг в поле «Новый solr» для корректной работы поиска;
- Заголовок – заголовок витрины;

- Html-шаблон заголовка – шаблон заголовка витрины;
- Показывать заголовок – логический признак отображения заголовка у витрины;
- Html-шаблон кнопок – шаблон кнопок (если на панели инструментов помимо стандартных должны отображаться дополнительные кнопки). Пример html-шаблона дополнительных кнопок:

```
<a class="btn btn-default m-r-sm" href="/test/#/app/document/new">
<i class="fa fa-plus"></i>
</a>
<a class="btn btn-default m-r-sm" href="/map/#/map;onMode3D=true">
<i class="fa fa-map"></i>
</a>
```

- Представление результатов поиска по умолчанию – результаты поиска в виде табличного представления или списка карточек по умолчанию;
- Табличная сортировка – логический признак включения табличной сортировки по умолчанию;
- Автономная сортировка – логический признак включения автономной сортировки по умолчанию;
- Табличные фильтры – логический признак включения табличной фильтрации по умолчанию;
- Автономные фильтры – логический признак включения автономных фильтров по умолчанию;
- Возможность выбора колонок – логический признак отображения компонента, позволяющего манипулировать видимостью колонок при табличном представлении результатов поиска;
- Возможность переключения результатов поиска – логический признак того отображения компонента, позволяющего переключать представление результатов поиска;
- Возможность переключения типа сортировки – логический признак возможности переключения типов сортировки;
- Возможность переключения типа фильтрации – логический признак возможности переключения типов фильтрации;
- Группировка – логический признак отображения компонента, позволяющего группировать результаты поиска
- Отключение общего поиска – возможность отключать общий поиск в витрине;
- Заполнитель для общего поиска – текст-подсказка, отображающийся в поле ввода общего поиска;
- Количество элементов на странице – выбор количества элементов на странице

результатов поиска;

– Возможность менять количество элементов на странице – логический признак отображения компонента, позволяющего менять количество элементов на странице;

– Фильтры, переданные при инициализации витрины, всегда будут применяться – логический признак того, будут ли фильтры, указанные при инициализации витрины (см. Таблица 94), всегда применяться к витрине;

– Новый solr – какой SOLR используется для данного типа документов (обязательно для корректного поиска при использовании /search/v1/solr/query);

– Отключение выбора настроек – при данной функции нет возможности выбрать в клиентской части различные виды для отображения;

– Дополнительные типы документов – позволяет добавить типы документов, которые будут также отображаться в витрине.

При использовании дополнительных типов документов необходимо учесть, что все используемые типы документов должны находиться в одном ядре SOLR. В кодах общих полей в таком случае должен использоваться не тип документа, а параметр «common».

4.8.3.3 Вкладка «Поля»

Вкладка «Поля» используется для заполнения информации о полях документа, индексируемых в solr (Рисунок 199).

Конструктор витрин >

Редактирование витрины Закупка офисного оборудования (мои заявки)

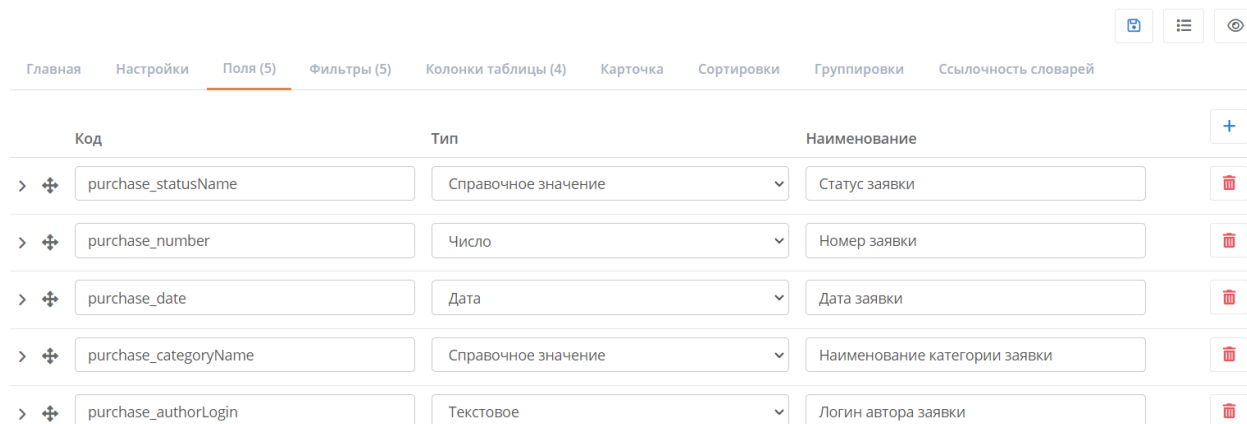





Рисунок 199 – Вкладка «Поля»

Функциональные кнопки вкладки «Поля» приведены ниже (Таблица 84).

Таблица 84 – Функциональные кнопки вкладки «Поля»

Кнопка	Действие
	Добавление нового поля

Кнопка	Действие
	Удаление поля
	Изменение последовательности полей
	Открывает доступ к дополнительной информации о поле, доступной для заполнения

Список полей вкладки «Поля», которые необходимо заполнять:

- Код – код поля в подсистеме поиска;
- Тип – тип поля (Текстовое, Логическое значение, Дата, Пользователь, Справочное значение, Вычисляемое);
- Наименование – наименование поля;
- Массив – логический признак является ли данное поле документа массивом. Отображается только для полей типа: ‘Текстовое’, ‘Пользователь’, ‘Справочное значение’;
- Сортируемое – логический признак того, может ли данное поле участвовать в сортировке. Отображается для всех полей не являющихся массивами, кроме вычисляемого поля;
- Фильтруемое – логический признак того, может ли данное поле участвовать в фильтрации. Отображается для всех полей, кроме вычисляемых;
- Сортировка по умолчанию – указывает как будут отсортированы значения поля при первом открытии витрины. Доступные значения: ‘По возрастанию’, ‘По убыванию’, ‘Отсутствует’. Допускается указать сортировку только для одного поля из списка. Отображается для всех полей, не являющихся массивами кроме вычисляемых полей;
- Разрешения – список разрешений, под которыми поле будет видно в витрине. Если не указано, то отображается для всех;
- Участвует в общем поиске – логический признак того, может ли данное поле участвовать в общем поиске. Отображается для всех полей, кроме вычисляемых;
- Справочник – справочник, который служит источником данных для данного поля документа. Отображается только для полей типа ‘Справочное значение’;
- Код справочника – поле из справочника, которое хранится в данном поле документа;
- Текст справочника – поле из справочника, которое будет отображаться в витрине;
- Иерархический справочник – заполняется автоматически из метаданных

выбранного справочника;

– Значения только на листьях справочника – только для иерархических справочников. Имеется в виду, что в данное поле документа вводятся значения, находящиеся только на листьях справочника.

Если для поля установлен флаг «Участвует в общем поиске» для него обязательно должны быть настроены условия поиска на вкладке «Фильтры».

4.8.3.4 Вкладка «Фильтры»

Вкладка «Поля» используется для заполнения информации о фильтрах витрины (Рисунок 200).

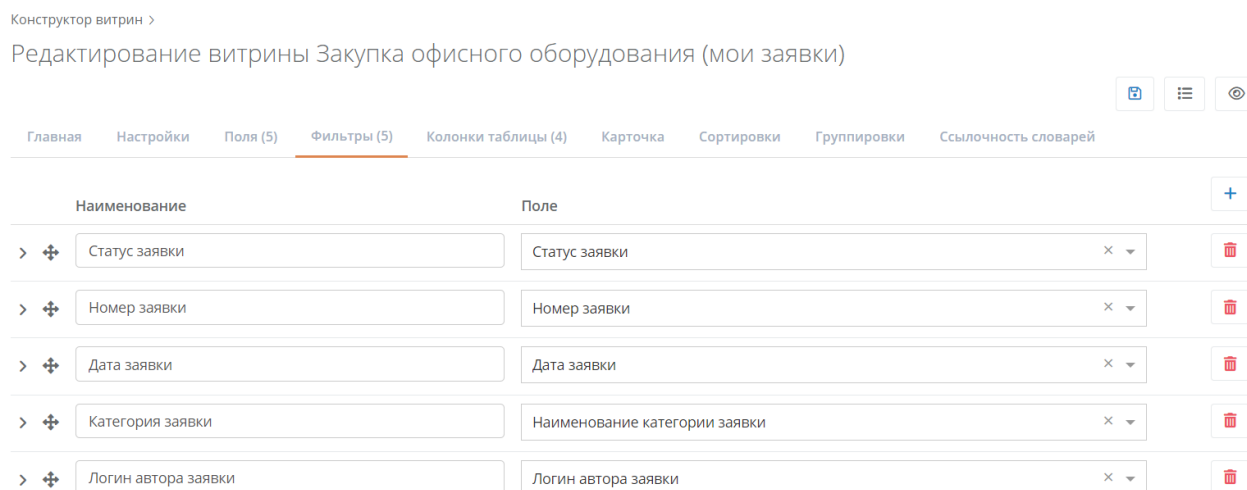


Рисунок 200 – Вкладка «Фильтры»

Функциональные кнопки вкладки «Фильтры» приведены ниже (Таблица 85).

Таблица 85 – Функциональные кнопки вкладки «Фильтры»

Кнопка	Действие
	Добавление нового фильтра
	Удаление фильтра
	Изменение последовательности фильтров в расширенном поиске
	Открывает доступ к дополнительной информации о фильтре доступной для заполнения

Список полей вкладки «Фильтры», которые необходимо заполнять:

- Наименование – имя фильтра, которое будет отображаться в расширенном поиске;
- Поле – ссылка на поле, по которому будет осуществляться фильтрация;

- Множественный – возможность фильтровать по нескольким значениям. Только для полей типа справочное значение и пользователь;
- html-шаблон – позволяет указывать шаблон для полей типа справочное значение или логическое значение;
- Заполнитель – позволяет указывать текст подсказку в поле ввода фильтра. Только для полей типа текстовое значение, пользователь или справочное значение;
- Поле для фильтрации по логину – возможность установить первоначальный фильтр записей по логину пользователя в системе. Если установлено несколько полей с данным свойством, то будут искаться записи, в которых хотя бы одно поле удовлетворяет условию (т.е. с OR , а не AND);
- Отображать только не пустые значения – позволяет установить фильтр, при котором будут отображаться только записи, в которых заданное поле не пустое;
- Учитывать ссылочность справочников – позволяет учитывать ссылочность справочников при фильтрации. Например, справочники округов и районов имеют ссылочную связь, поэтому при фильтрации по округу фильтр районов будет предзаполняться только районами, относящимися к данному округу. При этом должна быть запись на вкладке «Ссылочность словарей», где указаны параметры связи;
- Предустановленный фильтр – предустановленный фильтр на данном поле. Если поле заполнено, то в самой витрине данное поле не отображается. Может быть массив значений, разделитель «-». Если массив, то поиск – по условию ИЛИ;
- Фильтр по профилю пользователя – предустановленный фильтр на данном поле. Значение – путь до необходимого поля, динамическое, берется из json профиля пользователя с помощью метода /userprofile/bylogin/{login}. Если массив, то поиск – по условию ИЛИ;
- Строгий поиск – поиск без звездочек и кавычек: (поле: искомое значение);
- Поиск по части текста – включающий поиск (поле: *искомое значение*).

Пример заполнения поля Html-шаблон для проставления названий для логических полей:

```
<div *ngIf="item"> Удалено</div>
<div *ngIf!="item"> Активно</div>
```

На формирование запроса, отправляемого к SOLR при установке фильтра, влияют как тип, установленный для поля на вкладке «Поля», так и значения полей «Тип поля» (type) и «Тип поиска по полю» (searchRule), установленные для поля в справочнике «Поля документов подсистемы поиска» (SearchFields).

Для поиска в витрине по номерам документов, состоящим из цифр и символов (например: 2048-9300003-880145-011208/22), рекомендовано установить в справочнике

«Поля документов подсистемы поиска» (SearchFields) «Тип поля» (type) – STRING, а в настройках витрины на вкладке «Поля» тип – Номер документа.

4.8.3.5 Вкладка «Колонки таблицы»

Вкладка «Колонки таблицы» используется для заполнения информации о колонках в табличном отображении результатов поиска (Рисунок 201).

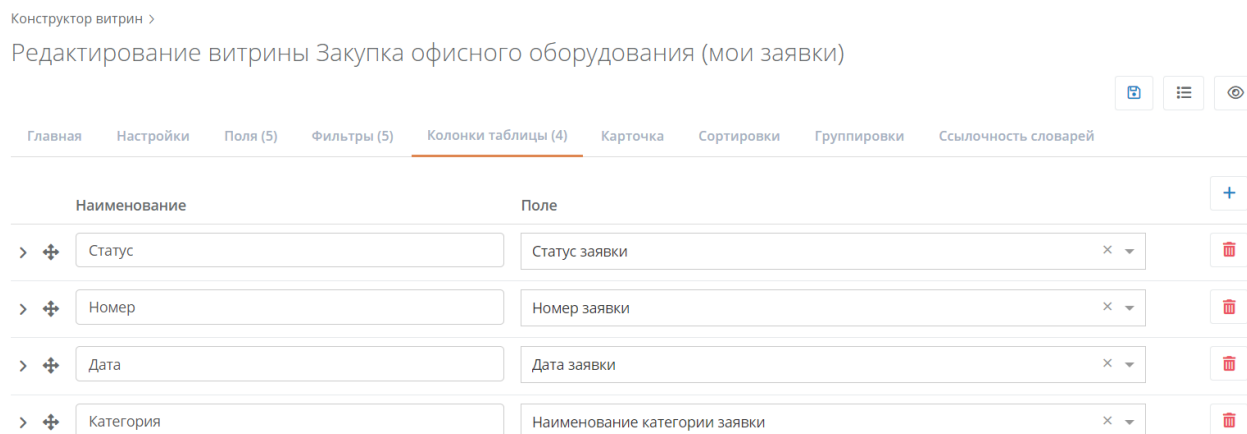






Рисунок 201 – Вкладка «Колонки таблицы»

Функциональные кнопки вкладки «Колонки таблицы» приведены ниже (Таблица 86).

Таблица 86 – Функциональные кнопки вкладки «Колонки таблицы»

Кнопка	Действие
	Добавление новой колонки
	Удаление колонки
	Изменение последовательности колонок в табличном отображении результатов поиска
	Открывает доступ к дополнительной информации о колонке доступной для заполнения

Список полей вкладки «Колонки таблицы», которые необходимо заполнять:

- Наименование – имя колонки, которое будет отображаться в заголовке колонки таблицы;
- Поле – связь колонки с полем. Заполнять, если для данной колонки нужна возможность табличной сортировки или фильтрации;
- Ширина – ширина колонки (указываем в процентах – % после числа, либо в пикселях – значок px после числа);
- Html-шаблон – html-шаблон колонки таблицы;

- Сортируемое – возможность использовать табличную сортировку. Доступно в зависимости от выбранного поля;
- Фильтруемое – возможность использовать табличную фильтрацию. Доступно в зависимости от выбранного поля;
- Видимое – видимость колонки при первом открытии витрины.

Пример заполнения поля Html-шаблон для поля с типом «Логическое значение»:

```
<a style="color: #676a6c;">
<span *ngIf="item.isDeletedAddress=== true">Да</span>
<span *ngIf="item.isDeletedAddress=== false">Нет</span>.
```

Пример заполнения поля Html-шаблон для поля с типом «Дата»;

```
<a style="color: #676a6c;" href="/extras/#/app/requests/permissions-
card/{{item.documentId}}/main">
<div *ngIf="item.documentDate">{{item.documentDate|date:'dd.MM.yyyy'}}</div>
</a>.
```

Пример заполнения поля Html-шаблон для поля с типом «Справочное значение» с отображением значения в цветной плашке в соответствии с данными справочника:

```
<a style="color: #676a6c;" href="/extras/#/app/requests/permissions-
card/{{item.documentId}}/main">
<div *ngIf="item.statusName">
<span [ngClass]="label label-' +
item.statusName.color">{{item.statusName.name}}</span>
</div>
</a>.
```

Пример заполнения поля Html-шаблон для поля с типом «Справочное значение» содержащего массив значений:

```
<a style="color: #676a6c;"
href="/rayon/#/app/object/card/{{item.sys_documentId}}/view/main" target="_blank">
<div *ngIf="item.prefectsNameObject">
<span *ngFor="let p of item.prefectsNameObject; let i = index;">{{i ? ', ' : ''}}<span
*ngIf="p && p.name">{{p.name}}</span></span>
</div>
</a>.
```

Пример заполнения поля Html-шаблон для поля, содержащего идентификатор файла в файловом хранилище, для формирования ссылки на файл:

```
<a style="color: #676a6c;">
<div *ngIf="item.fileId"><a
href="/filestore/v1/files/{{item.fileId}}?systemCode={{systemCode}}>Ссылка на
документ</a></div>
</a>.
```

Пример отображения разных блоков данных по условию:

```
<div *ngIf="item.isSet; then docset else document"></div>
<ng-template #docset><a style="color: #676a6c;"
href="/str/oks/#/app/docset/card/{{item.sys_documentId}}/main">
<div *ngIf="item.documentVersionNumber">{{item.documentVersionNumber}}</div>
</a></ng-template>
```

```

<ng-template #document><a style="color: #676abc;"
href="/str/oks/#/app/document/card/{{item.sys_documentId}}">
<div *ngIf="item.documentVersionNumber">{{item.documentVersionNumber}}</div>
</a></ng-template>.

```

Пример заполнения Html-шаблона для отображения иконок файлов в витрине (единичный экземпляр):

```

<a *ngIf="item.fileProtocolVersionSeriesGuid" style="color: #2589c5;"
href="/filestore/v1/files/{{item.fileProtocolVersionSeriesGuid}}?systemCode=STR"
target="_blank">
  
</a>.

```

Пример заполнения Html-шаблона для отображения иконок файлов в витрине - множественное поле, показываем все файлы:

```

<div *ngFor="let fileGuid of item.resultAcceptFileVersionSeriesGuid; index as i">
  <a style="color: #2589c5"
href="/filestore/v1/files/{{fileGuid}}?systemCode=STR" target="_blank">
  
  </a>
</div>.


```

Пример заполнения Html-шаблона для отображения иконок файлов в витрине - множественное поле, показываем только последний файл (Флаг isLast):

```

<div *ngFor="let fileGuid of item.resultAcceptFileVersionSeriesGuid; index as i; last as
isLast">
  <a *ngIf="isLast" style="color: #2589c5"
href="/filestore/v1/files/{{fileGuid}}?systemCode=STR" target="_blank">
  
  </a>
</div>.

```

Пример заполнения Html-шаблона для отображения иконок файлов в витрине - множественное поле, показываем только первый файл ((достаточно опереться на индекс  - если он равен нулю – его отрицание (!i) будет равно true)):

```

<div *ngFor="let fileGuid of item.resultAcceptFileVersionSeriesGuid; index as i">
  <a *ngIf="!i" style="color: #2589c5"
href="/filestore/v1/files/{{fileGuid}}?systemCode=STR" target="_blank">
  
  </a>
</div>.

```

4.8.3.6 Вкладка «Карточка»

Используется для заполнения html-шаблона карточки при отображении результатов поиска в виде списка карточек (Рисунок 202).

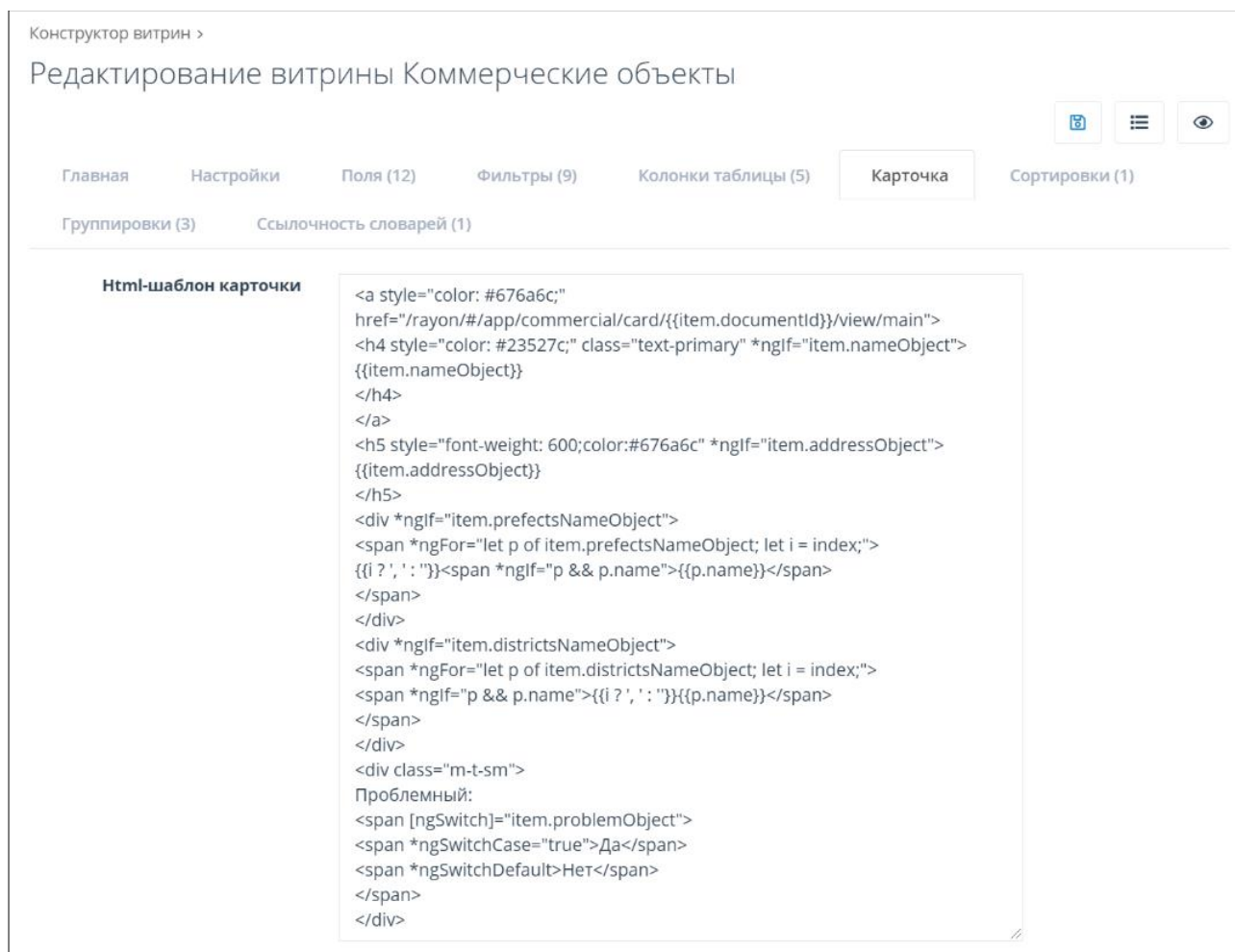


Рисунок 202 – Вкладка «Карточка»

4.8.3.7 Вкладка «Сортировки»

Используется для заполнения информации о сортировках (при автономной сортировке) (Рисунок 203).

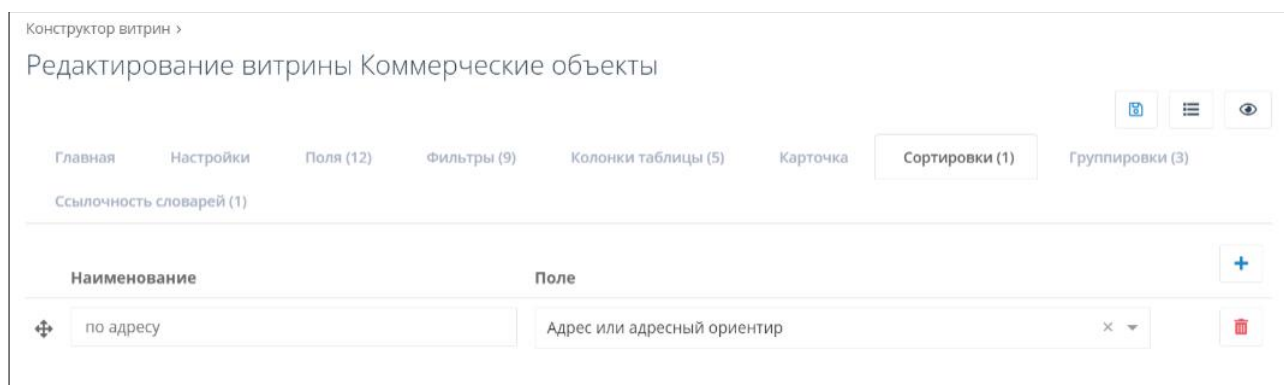



Рисунок 203 – Вкладка «Сортировки»

Функциональные кнопки вкладки «Сортировки» приведены ниже (Таблица 87).

Таблица 87 – Функциональные кнопки вкладки «Сортировки»

Кнопка	Действие
	Добавление новой сортировки

Кнопка	Действие
	Удаление сортировки

Список полей вкладки «Сортировки», которые необходимо заполнять:

- Наименование – имя, которое будет отображаться в компоненте автономной сортировки;
- Поле – связь сортировки с полем.

4.8.3.8 Вкладка «Группировки»

Вкладка «Группировки» используется для заполнения информации о группировках (Рисунок 204).

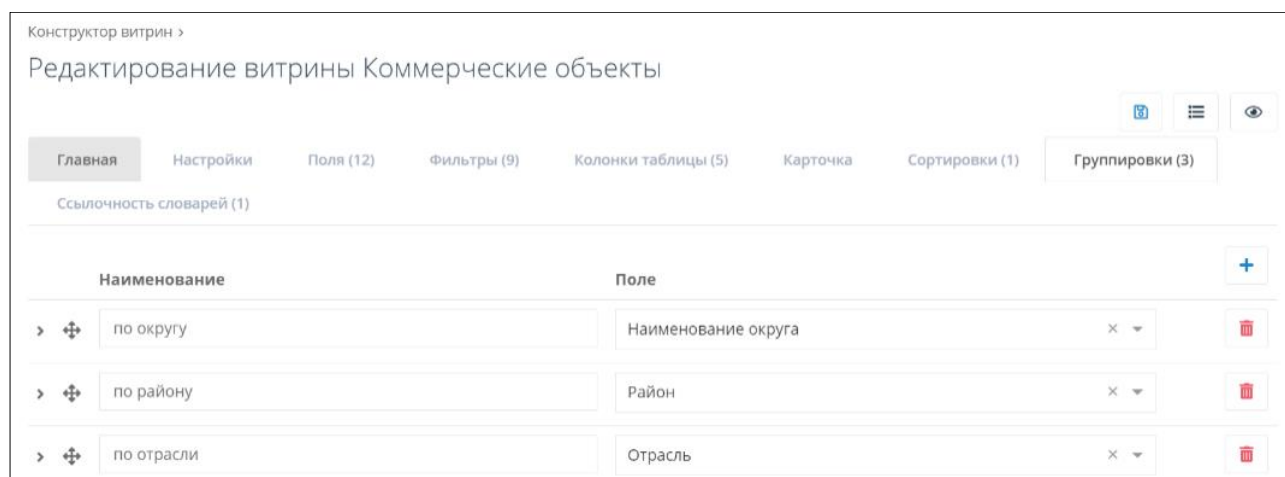






Рисунок 204 – Вкладка «Группировки»

Функциональные кнопки вкладки «Группировки» приведены ниже (Таблица 88).

Таблица 88 – Функциональные кнопки вкладки «Группировки»

Кнопка	Действие
	Добавление новой группировки
	Удаление группировки
	Изменение последовательности группировок в соответствующем компоненте
	Открывает доступ к дополнительной информации о группировки доступной для заполнения

Список полей вкладки «Группировки», которые необходимо заполнять:

- Наименование – имя, которое будет отображаться в компоненте, позволяющем выбирать группировку;

- Поле – связь группировки с полем;
- Html-шаблон – шаблон группировки;
- Учитывать ссылочность справочников – позволяет учитывать ссылочность справочников при группировке. Например, справочники округов и районов имеют ссылочную связь, поэтому при группировке по округу в рамках округа выполняется группировка по районам, относящимся к данному округу. При этом должна быть запись на вкладке «Ссылочность словарей», где указаны параметры связи;
- Скрыт – позволяет скрывать группировку из списка в компоненте выбора группировки. Например, если группируем по округам и учитываем ссылочность со справочником районов, то отдельно группировку по районам можно скрыть.

4.8.3.9 Вкладка «Ссылочность словарей»

Вкладка «Ссылочность словарей» используется для заполнения информации о связях двух справочников (Рисунок 205).

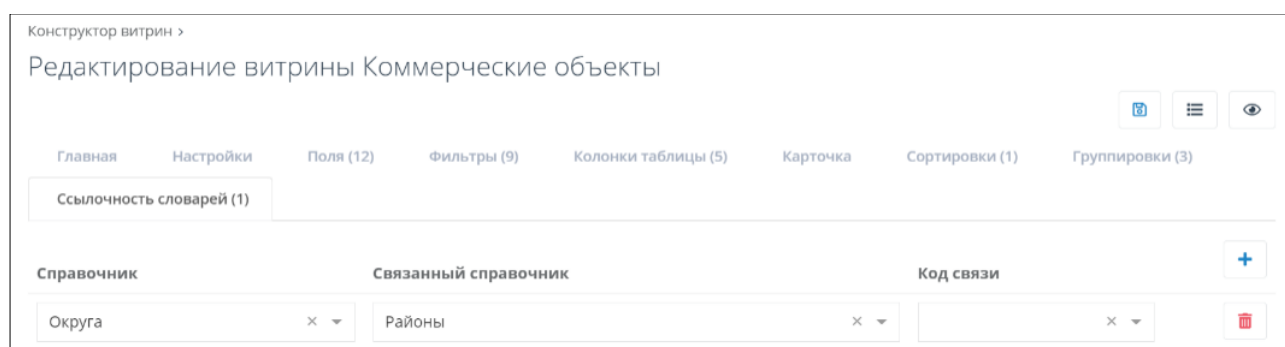


Рисунок 205 – Вкладка «Ссылочность словарей»

Функциональные кнопки вкладки «Ссылочность словарей» приведены ниже (Таблица 89).


Таблица 89 – Функциональные кнопки вкладки «Ссылочность словарей»

Кнопка	Действие
	Добавление новой связи
	Удаление связи

Список полей вкладки «Ссылочность словарей», которые необходимо заполнять:

- Справочник – справочник;
- Связанный справочник – связанный справочник;
- Код связи – поле, связывающее два справочника.

4.8.4 Предварительный просмотр

При нажатии на значок  открывается витрина предварительного просмотра (Рисунок 206).

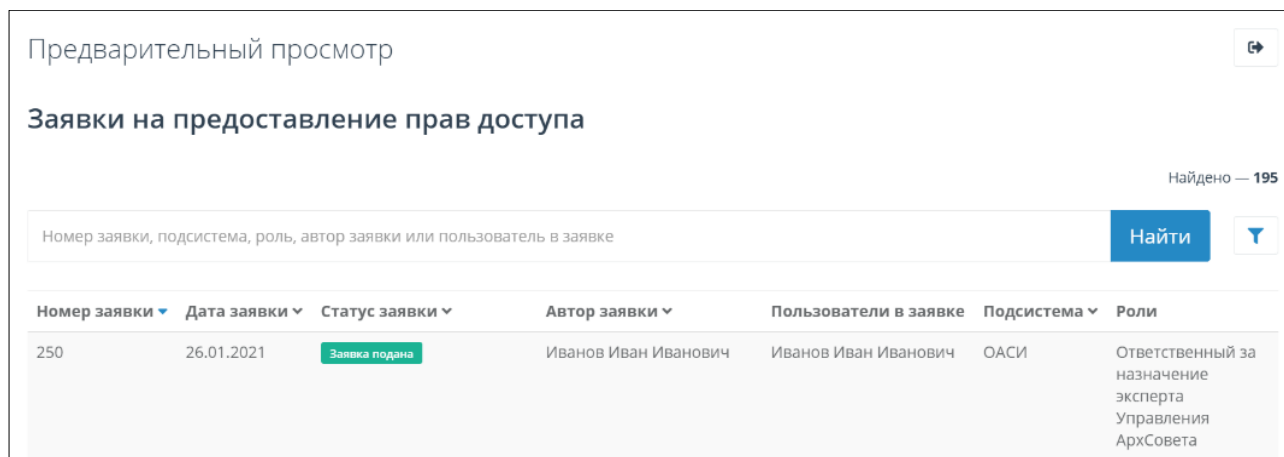






Рисунок 206 – Витрина предварительного просмотра

Выход из режима предварительного просмотра осуществляется нажатием на значок .

Витрина предварительного просмотра содержит следующие элементы:

- заголовок витрины. Пример: Заявки на предоставление прав доступа ;
- общий поиск. Пример:  ;
- количество найденных документов. Пример:  ;
- расширенный поиск в свернутом виде. Пример:  . Цифра в скобочках – количество примененных фильтров. При нажатии на значок происходит скрытие/отображение полей поиска;
- расширенный поиск в развернутом виде (Рисунок 207);

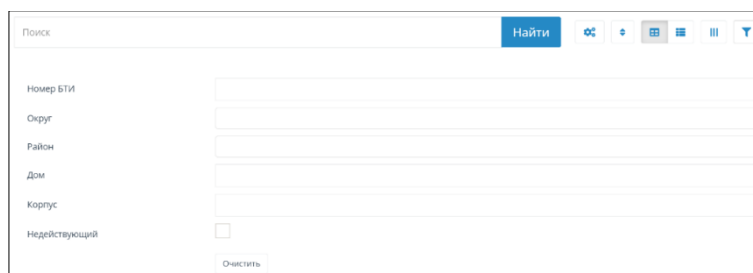


Рисунок 207 – Расширенный поиск

- витрина в режиме группировки (Рисунок 208);

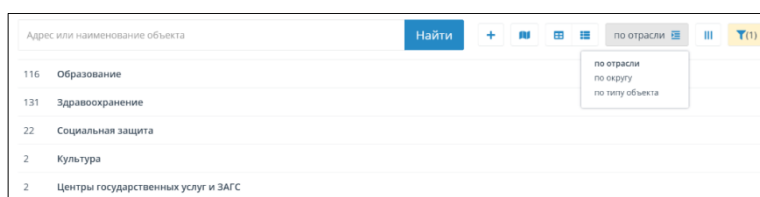



Рисунок 208 – Витрина в режиме группировки

– переключатель отображения результатов поиска (между табличным и списком карточек). Пример:  ;

– витрина в режиме отображения результатов поиска списком карточек (Рисунок 209);



Рисунок 209 – Витрина в режиме отображения результатов списком

– витрина в режиме отображения результатов поиска в виде таблицы (Рисунок 210);

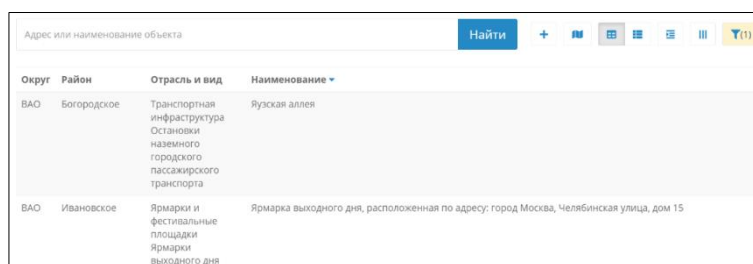


Рисунок 210 – Витрина в режиме отображения результатов таблицей

– настройка колонок табличного представления результатов поиска (Рисунок 211);

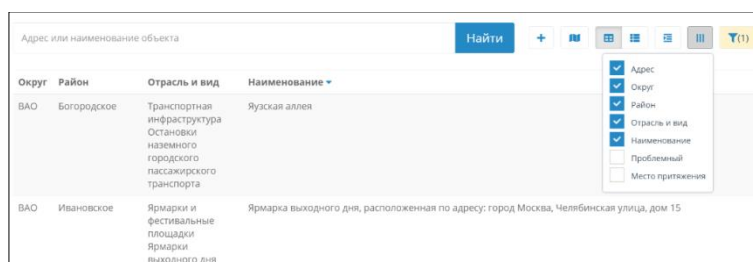


Рисунок 211 – Настройка колонок табличного представления

– витрина с компонентом автономной сортировки (Рисунок 212);

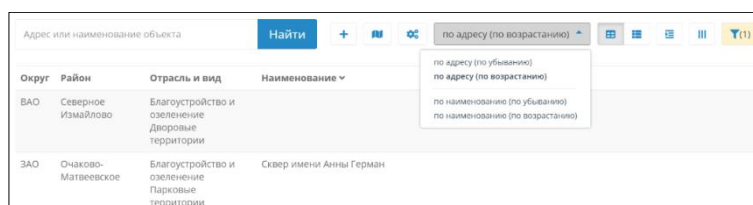



Рисунок 212 – Витрина с компонентом автономной сортировки

– компонент постраничной навигации (с включенной возможностью смены количества строк на странице). Пример:  ;

– компонент с дополнительными настройками витрины (Рисунок 213);

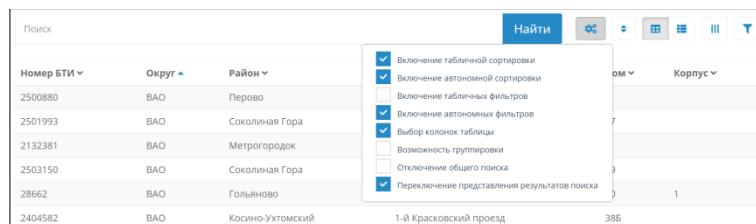


Рисунок 213 – Компонент с дополнительными настройками витрины

Дополнительные настройки витрины выполняют следующие функции:

- Включение табличной сортировки – управляет доступностью табличной сортировки;
- Включение автономной сортировки – управляет видимостью компонента автономной сортировки;
- Включение табличных фильтров – управляет доступностью табличных фильтров;
- Включение автономных фильтров – управляет видимостью компонента расширенного поиска;
- Выбор колонок таблицы – управляет видимостью компонента выбора колонок;
- Возможность группировки – управляет видимостью компонента позволяющего группировать результаты поиска;
- Отключение общего поиска – позволяет скрывать общий поиск из витрины;
- Переключение представления результатов поиска – управляет видимостью переключателя представлений результатов поиска.

4.8.5 Встраивание витрины в приложение

Встраивание витрины в приложение выполняется в следующем порядке:

1) в package.json в секцию dependencies добавить:

```
"@reinform-cdp/showcase-builder": "^1.0.2",
"ngx-autosize": "1.4.1",
"ngx-bootstrap": "3.2.0";
```

2) в styles.scss подключить: @import '~ngx-bootstrap/daterangepicker/bs-daterangepicker.css';

3) в app.module.ts добавить:

```
import {ShowcaseBuilderModule} from '@reinform-cdp/showcase-builder';
и подключить модуль в angular;
```

4) добавить витрину (формат параметра «config» см. Таблица 98):

```
<cdp-showcase-builder [config]="config"></cdp-showcase-builder>.
```

4.8.6 Описание данных конструктора витрин

Конструктор витрин предназначен для построения поисковых витрин на основе данных о документах индексируемых в SOLR (платформа полнотекстового поиска). Для работы конструктора в системе используются следующие справочники:

- Permissions – разрешения;
- ShowcaseBuilder– справочник с настройками витрин.

Создание и обновление элементов справочника ShowcaseBuilder осуществляется автоматически при сохранении настроек витрины через панель администрирования.

Описание справочника ShowcaseBuilder приведено ниже (Таблица 90).

Таблица 90 – Описание справочника ShowcaseBuilder

Поле	Описание
systemCode*	Код системы
systemName*	Наименование системы
subsystemCode*	Код подсистемы
subsystemName*	Наименование подсистемы
documentCode*	Код документа
documentName*	Наименование документа
showcaseCode*	Код витрины
showcaseName*	Наименование витрины
settings	Настройки витрины (см. Таблица 91)
code*	Код элемента справочника

Поле «code», а также комбинация полей полей «systemCode», «subsystemCode», «documentCode», «showcaseCode» должны быть уникальными.

Описание поля «settings» справочника ShowcaseBuilder приведено ниже (Таблица 91).

Таблица 91 – Описание поля «settings» справочника ShowcaseBuilder

Поле	Описание	Значение по умолчанию
theme	Тема оформления (доступно шесть тем: green, aero, blue, gray, orange, pink)	blue
url*	Путь к данным solr (например /search/v1/solr/query)	
title*	Заголовок витрины	
titleTemplate	Шаблон заголовка (для случая если в заголовке помимо текста должны отображаться дополнительные элементы, например, ссылки или кнопки перехода к созданию нового документа и др.)	
showTitle	Отображать или скрывать заголовок витрины (true – отображается, false – скрыт)	true

Поле	Описание	Значение по умолчанию
customBtnTemplate	Шаблон кнопок (для случая если на панели инструментов помимо стандартных должны отображаться дополнительные кнопки)	
defaultView	Представление отображения результатов поиска (card – список карточек, table – табличное представление)	table
tableSorting	Включение или отключение табличной сортировки (true – включена, false – отключена)	true
separateSorting	Включение или отключение автономной сортировки (true – включена, false – отключена)	false
tableFilter	Включение или отключение табличной сортировки (true – включена, false – отключена)	false
separateFilter	Включение или отключение автономных фильтров (true – включены, false – отключены)	true
changeTableColumns	Возможность манипулировать отображением колонок в таблице (true – включена, false – отключена)	false
changeViewType	Возможность переключения представлений результатов поиска (true – включена, false – отключена)	false
changeSortingType	Возможность переключения типов сортировки (true – включена, false – отключена)	false
changeFilterType	Возможность переключения типов фильтрации (true – включена, false – отключена)	false
grouping	Включение или отключение возможности группировки (true – включена, false – отключена)	false
disableCommonSearch	Возможность отключения общего поиска (true – отключен, false – включен)	false
placeholderForCommonSearch	Заполнитель для общего поиска	
numberOfElementPerPage	Количество элементов на странице (10,20,50,100)	20
changeNumberOfElementsPerPage	Возможность менять количество элементов на странице (true – включена, false – отключена)	false
urlFilterAlwaysApplied	Фильтры переданные переданные при инициализации витрины всегда будут применяться	false

Поле	Описание	Значение по умолчанию
	(true – включено, false – отключено)	
isNewSearch	Признак использования новой версии приложения search (true – включен, false – отключен)	true
hideSettings	Возможность отключения выбора настроек (true – отключен, false – включен)	true
fields*	Список полей витрины (см. Таблица 92)	
tableColumns	Список колонок табличного представления результатов поиска (см. Таблица 93)	
cardTemplate	Шаблон карточки при отображении результатов поиска в виде списка карточек	
filters	Список фильтров витрины (см. Таблица 94)	
sortings	Список сортировок витрины (см. Таблица 95)	
groupings	Список группировок витрины (см. Таблица 96)	
dictionaryLinks	Список ссылочности словарей (см. Таблица 97)	

Описание поля витрины приведено ниже (Таблица 92).

Таблица 92 – Описание поля витрины

Поле	Описание	Значение по умолчанию
code*	Код поля	
type	Тип поля ('text' – текстовое, 'boolean' – логическое, 'date' - дата, 'user' – пользователь, 'dictionary' - словарь, 'calculated' -вычисляемое)	text
name*	Наименование поля	
isArray	Массив (только для полей типа 'text', 'user', 'dictionary')	false
dictionary	Словарь (только для полей типа 'dictionary')	
dictionaryCode	Поле справочника которое храниться в поле solr (только для полей типа 'dictionary')	
dictionaryText	Поле справочника которое используется для отображения в витрине (только для полей типа 'dictionary')	
dictionaryAttrs	Поля справочника, которые используются для работы витрины (только для полей типа "dictionary")	
dictionaryLinksDict	Справочник, связанный с основным	

Поле	Описание	Значение по умолчанию
dictionaryLinksDictAttrs	Поля справочника, связанного с основным, которые используются для работы витрины (только для полей типа "dictionary")	
hierarchical	Признак иерархического справочника (только для полей типа `dictionary`)	заполняется автоматически
valueOnLeaves	Значения только на листьях иерархических справочников (только для полей типа `dictionary` и иерархических справочников) Имеется ввиду что в данное поле документа пишутся значения находящиеся только на листьях справочника	true
sortable	Признак того что данное поле может участвовать в сортировке (все кроме полей типа `calculated`)	false
filterable	Признак того что данное поле может участвовать в фильтрации (все кроме полей типа `calculated`)	false
defaultSorting	Сортировка по умолчанию (`ASC` - по возрастанию, `DESC` - по убыванию, `NONE` - без сортировки)	NONE
isSearchable	Признак того что данное поле может участвовать в общем поиске.	false
permissions	Разрешения (указываются из справочника Permissions) Влияют на отображение данного поля в результатах поиска в зависимости от прав доступа пользователя	если не указаны то поле отображается всегда

Описание колонки табличного представления результатов поиска приведено ниже (Таблица 93).

Таблица 93 – Описание колонки табличного представления результатов поиска

Поле	Описание	Значение по умолчанию
name*	Наименование колонки	
fieldCode	Ссылка на поле (см. Таблица 92)	
width	Ширина колонки (указываем в процентах - % после числа, либо в пикселях - значек px после числа)	
template	Шаблон	
sortable	Сортируемый столбец (только для сортируемых полей)	false
filterable	Фильтруемый столбец (только для фильтруемых полей)	false
visible	Видимость колонки	true

Описание фильтра витрины приведено ниже (Таблица 94).

Таблица 94 – Описание фильтра витрины

Поле	Описание	Значение по умолчанию
name*	Наименование фильтра отображаемое в расширенном поиске	
fieldCode*	Ссылка на поле (см. Таблица 92)	
multiple	Возможность фильтровать по нескольким значениям	false
template	Шаблон	
placeholder	Заполнитель	
loginFromStorage	Возможность добавления фильтра по умолчанию для логина пользователя в системе	false
notEmpty	Отображение записей только с не пустым полем	false
usingDictionaryLinks	Использовать ссылочность словарей. Например справочники округов и районов имеют ссылочную связь, поэтому при фильтрации по округу логично фильтр районов предзаполнять только районами относящимися к данному округу)	false
presetFilter	Предустановленный фильтр. Может быть массивом. Разделитель ";"	
userprofileFilter	Предустановленный фильтр из значения JSON, полученного из метода /userprofile/bylogin/{login} приложения app/sys/userprofile	
strictMode	"Строгий поиск" поле: искомое (без кавычек)	
searchPartOfText	Поиск включающий – поле: *искомое* (в звездочках)	

Описание сортировки витрины (для автономной сортировки) приведено ниже (Таблица 95).

Таблица 95 – Описание сортировки витрины (для автономной сортировки)

Поле	Описание	Значение по умолчанию
name*	Наименование сортировки	
fieldCode*	Ссылка на поле (см. Таблица 92)	

Описание группировки витрины приведено ниже (Таблица 96).

Таблица 96 – Описание группировки витрины

Поле	Описание	Значение по умолчанию
name*	Наименование группировки	
fieldCode*	Ссылка на поле (см. Таблица 92). Только для полей типа 'dictionary'	

Описание ссылочности справочников приведено ниже (Таблица 97).

Таблица 97 – Описание ссылочности справочников

Поле	Описание	Значение по умолчанию
dictionary*	Справочник	
linkedDictionary*	Связанный справочник	
code*	Код связи	

Описание параметров, передаваемых на вход конструктора при инициализации приведено ниже (Таблица 98).

Таблица 98 – Описание параметров, передаваемых на вход конструктора при инициализации

Поле	Описание
system*	Код системы
subsystem*	Код подсистемы
document*	Код документа
showcase*	Код витрины
calculatedFields	Список вычисляемых полей (см. Таблица 99)
filters	Список предустановленных фильтров (см. Таблица 100)
tabs	Список вкладок (вкладки представляют собой одну из разновидностей фильтров накладываемых на данные возвращаемые SOLR см. Таблица 101)

Описание вычисляемого поля приведено ниже (Таблица 99).

Таблица 99 – Описание вычисляемого поля

Поле	Описание
code*	Код поля
func*	Функция обратного вызова (на вход получает значение индексированных данных для одного документа из SOLR, на выходе значение вычисляемого поля)

Описание предустановленного фильтра приведено ниже (Таблица 100).

Таблица 100 – Описание предустановленного фильтра

Поле	Описание
code*	Код поля
value*	Значение фильтра

Описание вкладок приведено ниже (Таблица 101).

Таблица 101 – Описание вкладок

Поле	Описание
code*	Код вкладки
name*	Наименование вкладки
query*	Фильтр в виде query-запроса, накладывающий ограничение на данные, возвращаемые подсистемой поиска
active	true – указывается для одной из вкладок которая будет активной при первом открытии витрины. Для остальных вкладок необходимо указывать значение false.

4.9 Справочники

При разработке бизнес-процессов для Системы, предварительно заполняются следующие справочники:

- Системы (Systems);
- Подсистемы (SubSystems);
- Типы документов (DocumentTypes).

4.9.1 Справочник Systems

В справочнике Systems заполняются обязательные атрибуты с информацией о Системе (Таблица 102).

Таблица 102 – Перечень обязательных атрибутов

Наименование атрибута	Описание	Обязательный
code	Код системы	Да
name	Наименование системы	Да
Developer	Наименование организации - разработчика системы	Нет
url	URL ссылка системы	Нет
bpmUrl	URL ссылка на сервер BPM	Нет
alfrescoFolderName	Имя папки в хранилище документов Alfresco	Нет

Наименование атрибута	Описание	Обязательный
parent	Имя системы для аналитических отчетов	Нет
description	Описание	Нет
external	Внешняя система	Нет

Пример заполнения справочника Systems приведен ниже (Рисунок 214).

The screenshot shows a form for creating or editing a system entry. The fields are as follows:

- Код системы *** (code): DEMO
- Наименование системы *** (name): DEMO приложение
- Разработчик** (developer):
- URL системы** (url): /demo/
- URL движка БП** (bpmUrl): /app/demo/bpm
- Имя папки в Alfresco** (alfrescoFolderName): ИС Демо
- Имя системы для QLIK** (parent):
- Описание** (description): для аналитики
- Внешняя система** (external): Да Нет Отсутствует

Рисунок 214 – Пример заполнения справочника Systems

4.9.1.1 Справочник SubSystems

В справочнике SubSystems заполняются обязательные атрибуты с информацией о Системе (Таблица 103).

Таблица 103 – Перечень обязательных атрибутов

Наименование атрибута	Описание	Обязательный
code	Код подсистемы	Да
name	Наименование подсистемы	Да
appUrl	Ссылка на приложение	Нет
system	Система	Нет
alfrescoFolderName	Имя папки в хранилище документов Alfresco	Нет

Пример заполнения справочника SubSystems приведен ниже (Рисунок 215).

Код подсистемы * code	DEMO_DEMO
Наименование подсистемы * name	Демо
Ссылка на приложение appUri	/app/demo/demo
Система * system	DEMO DEMO приложение
Имя папки в Alfresco alfrescoFolderName	
Сортировочное значение элемента	

Рисунок 215 – Пример заполнения справочника SubSystems

4.9.1.2 Справочник DocumentTypes

В справочнике DocumentTypes заполняются обязательные атрибуты с информацией о типе документов (Таблица 104).

Таблица 104 – Перечень обязательных атрибутов

Наименование атрибута	Описание	Обязательный
code	Код типа документа	Да
name	Наименование типа документа	Да
solrCore	Ядро SOLR для поиска	Нет
fields	Список полей для поиска	Нет
virtTypes	Виртуальные типы документов	Нет
virtual	Виртуальный	Да
subsystem	Подсистема	Да
systemCode	Система	Да
informationObject	Модель ИО	Нет

Пример заполнения справочника DocumentTypes приведен ниже (Рисунок 216).

Код типа * code	DEMO_FORMSDEMO_CONTRACT
Название типа * name	Контракт для демонстрации
Ядро SOLR solrCore	demo_formsdemo Демо - клон динамических форм
Список полей fields	<ul style="list-style-type: none"> x demo_formsdemo_contract_contractDate contractDate x demo_formsdemo_contract_typeName typeName x demo_formsdemo_contract_siteAOrgName siteAOrgName x demo_formsdemo_contract_siteBOrgName siteBOrgName
Виртуальные типы документов virtTypes	<input type="text"/> <input type="button" value="X"/> <input radio"="" type="button" value="+</input></td> </tr> <tr> <td>Виртуальный *
virtual</td> <td><input type="/> Да <input checked="" type="radio"/> Нет <input type="radio"/> Отсутствует
Подсистема * subsystem	DEMO_FORMSDEMO Демо для конструктора форм
Система * systemCode	DEMO DEMO приложение
Модель ИО informationObject	<input type="text"/>
Сортировочное значение элемента	<input type="text"/>

Рисунок 216 – Пример заполнения справочника DocumentTypes

4.10 Администрирование

Описание управления пользователями и правами доступа, справочниками и классификаторами, настройками и параметрами автоматизируемых процессов приведено в документе «Руководство администратора».

5 АВАРИЙНЫЕ СИТУАЦИИ


5.1 Действия в случае несоблюдения условий выполнения технологического процесса, в том числе при длительных отказах технических средств

При сбое, выходе из строя или при длительных отказах одного или нескольких основных средств или обеспечивающих средств технического обеспечения Платформы, необходимо обратиться к Администратору Платформы или службу технической поддержки по почте: rif-support@reinform.ru.

5.2 Действия по восстановлению программ и/или данных при отказе магнитных носителей или обнаружении ошибок в данных

Для восстановления программ и/или данных при отказе магнитных носителей в составе основных средств или обнаружении ошибок в данных необходимо обратиться к Администратору Платформы.

5.3 Действия в случаях обнаружении несанкционированного вмешательства в данные

Незамедлительно обратиться в службу технической поддержки Платформы, используя значок  рядом со значком выхода из Платформы, для отправки сообщения о несанкционированном вмешательстве в данные.

5.4 Действия в других аварийных ситуациях

Действия в других аварийных ситуациях, повлекших частичную или полную потерю данных, описаны ниже (Таблица 105).

Таблица 105 – Действия в аварийных ситуациях, повлекших частичную или полную потерю данных

№ п/п	Аварийная ситуация	Возможные потери информации	Действие	Исполнитель
1.	Сбой АРМ Администратора Платформы, повлекший аварийный останов клиентского приложения Подсистемы (Платформы), включая отключение питания, сбой операционной системы и т.д.	Несохраненные пользователем данные	Перезапустить АРМ, запустить клиентское приложение Подсистемы (Платформы), повторить действия по ведению / редактированию данных	Администратор Платформы
2.	Сбой при передаче данных	Несохраненные пользователем данные	Перезапустить клиентское приложение Подсистемы (Платформы), повторить действия по ведению /	Администратор Платформы

№ п/п	Аварийная ситуация	Возможные потери информации	Действие	Исполнитель
			редактированию данных	
3.	Сбой ППО клиентского приложения Подсистемы (Платформы)	Несохраненные пользователем данные	Перезапустить клиентское приложение Подсистемы (Платформы), повторить действия по ведению / редактированию данных При повторных сбоях обратиться к разработчику Платформы для устранения недостатков в рамках его гарантийных обязательств	Администратор Платформы
4.	Сбой основных средств или ОПО Платформы, повлекший частичную или полную потерю данных, включая сбой операционной системы, СУБД	Вся информация, поступившая в подсистему с момента окончания последнего резервного копирования данных.	Выполнить загрузку данных и программ; Выполнить восстановление данных из последней резервной копии; Выполнить проверку работоспособности ППО Платформы.	Администратор Платформы

6 РЕКОМЕНДАЦИИ ПО ОСВОЕНИЮ

Для освоения работы с Платформой рекомендуется ознакомиться с Эксплуатационной документацией на Платформу и следовать рекомендациям производителя ОПО Платформы, правилам их запуска и выполнения, указанным в эксплуатационной документации на соответствующие компоненты Платформы.