

ООО «АйТи-Альянс»

<http://www.it-alnc.ru/>

ИНН 7718755096 / КПП 772501001

ОГРН 1097746108745

# **Описание функциональных характеристик экземпляра программного обеспечения ИТА ВРМ («Конструктор процессов»)**

Москва, 2026



## Оглавление

|                                                                            |    |
|----------------------------------------------------------------------------|----|
| 1.Аннотация.....                                                           | 3  |
| 2.Термины и определения.....                                               | 3  |
| 3.Основные функции программного обеспечения ИТА BPM.....                   | 4  |
| 3.1.  ИТА-DESIGNER-UI .....                                                | 4  |
| 3.1.1.  Редактор форм .....                                                | 4  |
| 3.1.2.  Множественная загрузка форм.....                                   | 24 |
| 3.1.3.  Описание объектов DataVault .....                                  | 25 |
| 3.1.4.  Редактор бизнес – объектов .....                                   | 26 |
| 3.1.5.  Редактор бизнес-процессов.....                                     | 27 |
| 3.1.6.  Редактор меню.....                                                 | 34 |
| 3.2.  ИТА-DESIGNER-EDITOR.....                                             | 36 |
| 3.3.  ИТА-DESIGNER-VIEWER.....                                             | 36 |
| 3.4.  ИТА-ADMIN-UI .....                                                   | 37 |
| 3.4.1.  Пользователи .....                                                 | 37 |
| 3.4.2.  Роли .....                                                         | 39 |
| 3.4.3.  Политики доступа .....                                             | 42 |
| 3.4.4.  Орг. Структура .....                                               | 45 |
| 3.4.5.  Бизнес-процессы .....                                              | 49 |
| 3.4.6.  Список задач.....                                                  | 55 |
| 3.4.7.  Список сервисных задач (Service Task) .....                        | 57 |
| 3.5.  Uiback.....                                                          | 59 |
| 3.6.  Tasklist.....                                                        | 59 |
| 3.6.1.  Предоставление списка задач .....                                  | 59 |
| 3.6.2.  Изменение параметров задачи.....                                   | 60 |
| 3.7.  Набор сервисов для асинхронного взаимодействия с процессом (воркеры) | 60 |
| 3.7.1.  worker-start .....                                                 | 61 |
| 3.7.2.  worker-usertask.....                                               | 61 |
| 3.7.3.  worker-usertask-complete .....                                     | 61 |
| 3.7.4.  worker-message .....                                               | 61 |
| 3.7.5.  worker-ext-task .....                                              | 61 |
| 3.7.6.  worker-ext-task .....                                              | 61 |
| 3.8.  Gateway .....                                                        | 61 |
| 3.9.  Движок ИТА-BPM версии 7.....                                         | 62 |
| 3.10.  Движок ИТА-BPM версии 8 .....                                       | 62 |

## 1. Аннотация

Настоящий документ содержит описание функциональных характеристик программного обеспечения ИТА BPM («Конструктор процессов») в части её базового функционала.

Программное обеспечение ИТА BPM («Конструктор процессов») представляет собой интеграционный сервис управления бизнес-процессами, расширяющий opensource-библиотеки Camunda BPM для работы с web-приложениями. Решение обеспечивает быструю разработку прикладных сервисов для процессов Заказчика и отказоустойчивое управление бизнес-процессами в промышленной эксплуатации, сохраняя и поддерживая сквозные бизнес-процессы организации.

## 1. Термины и определения

| Термин                                       | Определение                                                                                                                                                                                                                                      |
|----------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Программное обеспечение / ПО                 | Программа или множество программ, используемых для управления компьютером                                                                                                                                                                        |
| Модуль                                       | Составная часть программного обеспечения, обладающая самостоятельным функционалом                                                                                                                                                                |
| Приложение                                   | Программное обеспечение «Дизайнер форм и бизнес-процессов»                                                                                                                                                                                       |
| web-приложение, веб-приложение               | Прикладное программное обеспечение, которое работает на веб-сервере, в отличие от компьютерных программ, которые запускаются локально в операционной системе (ОС) устройства                                                                     |
| БД, база данных                              | Совокупность данных, хранимых в соответствии со схемой данных, манипулирование которыми выполняют в соответствии с правилами средств моделирования данных                                                                                        |
| BPMN 2.0                                     | Стандарт моделирования бизнес-процессов, поддерживаемый ИТА BPM.                                                                                                                                                                                 |
| Бизнес-процесс                               | Набор действий, выполняемых внутри организации или между организациями. В спецификации BPMN бизнес-процесс изображается в качестве диаграммы, представляющей собой совокупность действий и шлюзов, влияющих на порядок выполнения этих действий. |
| Задача пользователя (User Task)              | Задача в процессе, предназначенная для выполнения пользователем через пользовательский интерфейс.                                                                                                                                                |
| Сервисная задача (Service Task)              | Автоматическая задача, выполняемая системой без участия пользователя.                                                                                                                                                                            |
| Форма (Form)                                 | Пользовательский интерфейс для взаимодействия с задачей, создаваемый в редакторе форм.                                                                                                                                                           |
| Идентификатор взаимодействия (correlationId) | Уникальный идентификатор, используемый для связывания сообщений и задач в рамках одного процесса или скринфлоу.                                                                                                                                  |
| Скринфлоу (Screen Flow)                      | Последовательность пользовательских задач, выполняемых одним или несколькими пользователями в рамках единого интерфейсного потока.                                                                                                               |
| Обработчик событий (Worker)                  | Сервис, выполняющий фоновые операции (например, старт процесса, завершение задач, информирование о появлении новой задачи).                                                                                                                      |

| Термин        | Определение                                                                                    |
|---------------|------------------------------------------------------------------------------------------------|
| Кафка (Kafka) | Платформа потоковой передачи данных, используемая для асинхронного взаимодействия компонентов. |
| Виджет        | Вариант формы, у которой дополнительно определен собственный источник данных.                  |

## 2. Основные функции программного обеспечения ИТА ВРМ

Программное обеспечение ИТА ВРМ включает в себя следующие функции/сервисы:

- ita-designer-ui – интерфейс редактора форм и схем бизнес-процессов
- ita-designer-editor – сервис редактирования форм и схем бизнес-процессов
- ita-designer-viewer – сервис предоставления форм и схем бизнес-процессов
- ita-admin-ui – интерфейс администратора (просмотра бизнес-процессов, списков пользовательских и сервисных задач)
- uiback – сервис для взаимодействия с интерфейсом пользователя
- tasklist – сервис ведения списка задач
- набор сервисов для асинхронного взаимодействия с процессом (воркеры)
- gateway – маршрутизатор запросов
- поддержка работы на движке ИТА-ВРМ версии 7
- поддержка работы на движке ИТА-ВРМ версии 8

### 2.1. ИТА-DESIGNER-UI

Интерфейс редактора форм и схем бизнес-процессов содержит следующие разделы:

- **Редактор форм** – позволяет создавать и редактировать экранные формы разной сложности;
- **Множественная загрузка форм** – позволяет загружать в систему / выгружать из системы шаблоны форм в виде файлов .txt или .json; переносить формы и бизнес-процессы между стендами в среде заказчика.
- **Объекты DataVault** – позволяет просматривать структуру бизнес-объектов в формате DataVault (хабов, сателлитов и связей).
- **Редактор бизнес-объектов** – позволяет создавать модель данных
- **Редактор процессов** – позволяет создавать и редактировать схемы бизнес – процессов;
- **Редактор меню** – позволяет настраивать содержимое разделов меню.

#### 2.1.1. Редактор форм

Данный функционал позволяет создавать экранные формы, которые в дальнейшем можно вызывать как в бизнес – процессах – в пользовательских задачах User tasks, так и вне процессов. Drag-and-drop интерфейс редактора позволяет перетаскивать элементы дизайна на веб-странице и изменять их положение или порядок.

Базовые функции:

- **Поиск и открытие форм** – позволяет найти в списке доступных форм нужную форму, открыть найденную форму для последующего редактирования.
- **Сохранение форм** – сохраняет внесенные изменения; допустимо как сохранение основной формы, так и сохранение всей цепочки вложенных форм.
- **Действия с формой** – позволяет загрузить форму из файла для последующего редактирования, сохранить форму в файл (для последующей загрузки), скопировать форму, привязать форму к пользовательской задаче бизнес-процесса.
- **Редактор формы** – drag-and-drop интерфейс редактора, в котором выполняется настройка формы.
- **Предварительный просмотр** – позволяет оценить, как будет выглядеть форма по мере добавления на нее элементов дизайна.
- **Паттерн формы** – позволяет описать структуру данных, которые будут отображаться на форме.
- **Хендлеры формы** – позволяет настроить вызов определенных процедур в форме, которые будут выполняться при наступлении определенных событий – триггеров.
- **Параметры формы** – позволяет задать некоторые глобальные параметры формы.
- **Объекты** – позволяет создавать на форме свое собственное меню (справа) для перехода в другие формы.

#### 2.1.1.1. Функция поиска формы

Данная функциональность позволяет найти нужную форму для последующего редактирования. Поиск выполняется по уникальному коду - имени формы, либо названию - лейблу формы. Функция доступна в редакторе посредством поисковой строки.

#### 2.1.1.2. Функция открытия формы

Данная функциональность позволяет открыть на редактирование найденную в БД форму. Функция доступна в редакторе посредством кнопки «Открыть», становится активной, если заданная в поисковой строке форма найдена.

#### 2.1.1.3. Функция сохранения формы

Данная функциональность позволяет сохранять в БД изменения, выполненные на форме. Данная функциональность доступна в редакторе в виде двух кнопок:

- Сохранить основной шаблон
- Сохранить
- Кнопка «Сохранить основной шаблон» выполнит сохранение основной формы, при этом не будет сохранять изменения в шаблонах таблиц, виджетов и других контролов, использующих внутри себя формы-шаблоны.

Кнопка «Сохранить» сохранит всю цепочку вложенных форм.

#### 2.1.1.4. Действия с формой

В рамках данной функциональности поддерживаются следующие действия с формой:

- **Загрузить из файла** – позволяет загрузить в форму сохраненный ранее в виде файла .txt / .json шаблон
- **Сохранить в файл** – позволяет сохранить открытую текущую форму в виде файла txt / .json
- **Скопировать форму** – позволяет скопировать форму со всеми вложенными формами (кроме форм, открываемых хендлерами). Получившаяся копия не сохранена в БД, поэтому её нужно сохранить чтобы не потерять изменения.
- **Привязать к процессу** – позволяет связать форму с пользовательской задачей в некотором бизнес – процессе.

Данная функциональность представлена в редакторе в виде выпадающего меню.

#### 2.1.1.5. Функционал редактирования формы

Данная функциональность позволяет создавать новые формы и редактировать существующие.

Функционал доступен в закладке «Редактор» и представляет собой drag-and-drop интерфейс.

Обязательные атрибуты для создания формы:

- **Name** – уникальный код формы
- **Label** – Наименование формы

После того как форма создана, ее нужно наполнить элементами дизайна, или контролами. Наполнение формы контролами осуществляется путем перетаскивания мышкой нужного элемента из перечня доступных элементов. Для вновь добавленного элемента необходимо задать атрибуты.

Обязательными атрибутами являются:

- **Attribute name** – уникальный идентификатор поля формы; указывается путь к данным внутри HLS-данных в формате *hub.{link[x]}.hub.{sat.attr}*. Путь должен однозначно соответствовать единственному значению в наборе данных, полученному из источника данных. Пример доступа к данным в простом поле input:

*hub\_MPShowcase.link\_MPVersion\_MPShowcase[0].hub\_MPVersion.sat\_MPVersion\_Main.Description*

- **Ширина (в связке с Настройка ширины)** – ширина отображения поля на странице;
- **Control type** – тип контрола.

Другие атрибуты могут стать обязательными в сочетании с какими-то определенными полями. (Например, при создании таблицы необходимо заполнять и поле Template name).

- **Label** – отображаемое название поля формы (то, что будет отображаться на экране в качестве подписи к полю);

- **Переменная** – переменная Camunda, связанная с полем формы, которая передается в бизнес-процесс;
- **Placeholder** – текстовая подсказка-заполнитель, для полей вида input, text-area;
- **Test formulae** – для контроля text позволяет задать формулу преобразования значения (можно указать значение другого поля в соответствии с правилами построения выражений);
- **Color** – для кнопок action, позволяет выбрать цвет и рамку кнопок из списка;
- **Hint/icon** - позволяет назначать иконки некоторым полям, например, кнопкам; требуется указать код иконки; поддерживаются как коды из Angular Material Icons, так и кастомные из собственной библиотеки;
- **Items** – опции, позволяет задавать константные значения для контролов, предполагающих выбор из нескольких значений;
- **Disabled** – позволяет сделать поле неактивным / не редактируемым;
- **key field** – признак ключевого поля;
- **Template name** – указывается имя шаблона, который вызывается в экранной форме (для таблиц, массивов, аккордеонов, некоторых кастомных контролов);
- **Title** – здесь задается формула для заголовка элемента аккордеона;
- **Description** – здесь задается формула для описания элемента аккордеона;

Также у каждого контрола есть свои дополнительные параметры, расширяющие его функциональность. Дополнительные параметры определяются типом контрола.

Для контролов доступен функционал удаления и копирования. Функционал доступен с помощью соответствующих кнопок.

При копировании контрола у этой копии назначается новый attribute\_name, поскольку не может быть двух контролов на форме с одинаковыми attribute\_name, это уникальное значение.

#### 2.1.1.5.1. Типы контролов

Все контролы в редакторе форм условно сгруппированы в несколько разделов:

- Поля ввода
- Контейнеры
- Таблицы
- Ссылки и файлы
- Сложные контролы
- Другое

##### 2.1.1.5.1.1. Поля ввода

В данном меню редактора форм собраны простые поля, предназначенные как для просмотра, так и для ввода / редактирования данных

- **Input** - простое текстовое поле для ввода и отображения данных; текст отображается одной строкой
- **Input-number** - поле для ввода числовых значений, значение форматируется с разделителями разрядов и дробной части

- **Text-area** - текстовое поле для ввода и отображения данных; позволяет ввести длинный текст, который отображается в несколько строк
- **Check-box** - чек-бокс. Используется для различных признаков, подразумевающих значение «Да» или «Нет»
- **Slide-toggle** - переключатель, по смыслу это чек-бокс
- **Radio-button** - для различных признаков, вариантов. Варианты можно указать с помощью опций в поле Items, тогда если значение поля соответствует одному из вариантов в Items, то автоматически будет устанавливаться в этом варианте.
- **Solid-radio-button** - как Radio-button, но внешний вид - панель
- **Slider** - позволяет выбирать значение из диапазона с определенным шагом, перемещая ползунок
- **Select** - выбор значения из списка (список выпадающий). Список можно задать:
  - С помощью опций в атрибуте «Items»,
  - С помощью хендлера - получить данные из каких-либо справочников ITA Info,
  - С помощью параметров listPath, listKey, listValue – выбрать массив из контекста данных или из переменной бизнес-процесса.
- **Date-picker** - для отображения и задания даты без времени
- **Datetime-picker** - для отображения и задания даты со временем
- **Date-picker-range** - для задания диапазона дат

#### 2.1.1.5.1.2. Контейнеры

В данном меню редактора форм собраны контролы, подразумевающие группировку полей на форме.

- **Accordion** - аккордеон. Позволяет отображать вложенные массивы.
- **Array** – массив. Позволяет отображать на форме множественные значения данных.
- **Container** - элемент дизайна; предназначен для красивой группировки полей на форме.
- **Expansion-panel** - элемент дизайна; также предназначен для красивой группировки полей на форме, при этом группу можно свернуть или развернуть.
- **Modal** - контейнер для полей, которые будут отображены в модальном окне. Используется в связке с хендлером "Вывод модального окна" - чтобы открыть модальное окно нужно создать этот хендлер, указать соответствующий тип модального окна и указать нужное модальное окно. Далее этот хендлер можно выполнить в любом удобном месте, например - привязать к кнопке.
- **Tab** - элемент tab-панели – одна из закладок горизонтального меню
- **Tablist** - Tab-панель – это горизонтальное меню закладок

#### 2.1.1.5.1.3. Таблицы

В данном меню редактора форм собраны различные виды таблиц, для которых источником данных является паттерн формы.

- **Simple-table** - простая таблица с данными.
- **Multiple-table-select** - для множественного выбора данных, которые отображаются в виде таблицы (таблица с чек-боксом).
- **Object-table** – сложная таблица, в которой можно проваливаться в содержимое строки. Детальное содержимое строки открывается поверх таблицы с возможностью возврата назад.
- **Address-table** – специальная таблица для отображения адресов клиента

#### 2.1.1.5.1.4. Ссылки и файлы

- **Link** - позволяет отображать значение в виде ссылки с открытием другой формы в отдельном экране (модальное окно или перекрытие текущего экрана) по клику.
- **File** - контрол, позволяющий пользователю выбрать файл. Доступ к выбранному файлу осуществляется с помощью модификатора get-file: {{ my-file | get-file }}

#### 2.1.1.5.1.5. Сложные контролы

В данном меню редактора форм собраны кастомные контролы со специально разработанным сложным функционалом.

- **Ita-object-list** - контрол, предназначенный для реализации реестровых таблиц. Отбирает объекты по заданным критериям. Фильтрацию для отбора можно либо настроить в интерфейсе, либо в редакторе с помощью параметра "ita-filter". Значениями фильтров могут быть выражения. Если не заданы параметры объекта (object-template, object-boclass), то открытие формы объекта будет невозможно. В шаблоне строки таблицы отбора должно присутствовать поле с параметром role: id - значение этого поля будет использовано в качестве идентификатора объекта при двойном клике по строке. Для того чтобы по двойному клику по строке форму объекта можно было открыть, нужно добавить соответствующие параметры - object-template и object-boclass.
- **Ita-object-history-list** - таблица истории изменений. Самостоятельный реестр, которому для получения данных необходимо указать имя сателлита и идентификатор объекта. Под объектом подразумевается родительский для нужного хаб. В этом реестре отображаются состояния объекта после каждого изменения, изменённые поля подсвечены цветом #B5E61D. Таблица истории изменений не имеет отдельного шаблона - шаблон строится прямо на лету по данным от ITAObjects. По этой причине изменение шаблона в редакторе отключено.
- **Ita-object-search** - позволяет пользователю отобрать объект по заданным критериям. Фильтрацию для отбора можно либо настроить в интерфейсе, либо в редакторе с помощью параметра "ita-filter". Значениями фильтров могут быть выражения. Если не заданы параметры объекта (object-template, object-boclass), то открытие формы объекта будет невозможно. В шаблоне строки таблицы отбора должны присутствовать поле с параметром role: id и поле с параметром role: label - значения этих полей будут использованы в качестве идентификатора и имени объекта при выборе объекта соответственно

- **Ita-object-tree** - представляет собой иерархический список элементов. Обязательным параметром является object-tree-levels, который представляет собой набор параметров для каждого уровня дерева.
- На данный момент существует 2 вида узлов: Статические узлы - узлы, которые не обращаются к операциям и, соответственно, не получают данные извне. Имеют простую структуру key + value; Динамические узлы - узлы, которые при клике обращаются к операции с id выбранного узла. Каждый узел имеет полный набор данных, пришедших от операции.
- **Widget** - блок с динамическим контентом для отображения данных объекта: по заданному типу (object-boclass) и идентификатору хаба (object-bocode) загружаются данные объекта и привязываются к полям шаблона (object-template). При изменении связанных объектов виджет самостоятельно обновляет данные шаблона в соответствии с выбранной политикой обновления.
- **Ita-mass-upload** - кастомный контрол, разрабатывался для реализации функционала Массовых операций.

#### 2.1.1.5.1.6. Другое

- **Action** - кнопки, предназначено для запуска операций или выполнения специальных действий. Является триггером в хендлерах.
- **Dual-select** - для множественного выбора данных; позволяет перемещать данные из одного окошка в другое.
- **Chips-select** - для множественного выбора данных; позволяет выбирать множество данных, которые будут отображаться отдельными «чипами».
- **Divider** - задает линию-разделитель, может быть вертикальным и горизонтальным.
- **Hidden** - скрытое поле, может использоваться как в разметке, так и для задания скрытых значений для вычислений.
- **Image** - позволяет добавлять картинку в форму.
- **Input-address** - поле для задания адреса строкой (в некоторых проектах взаимодействие со справочником ФИАС),
- **Multiple-select** - для множественного выбора данных; данные выбираются из выпадающего списка и отображаются строкой через запятую.
- **Stepper** - элемент дизайна, позволяет задавать текущий шаг процесса; может быть горизонтальным и вертикальным.
- **Text** - простой текст; можно задать в поле «Text formulae» как константу, так и более сложное выражение с помощью модификаторов. По умолчанию отображается значение атрибута attribute\_name.
- **Dynamic-field** - для задания динамической таблицы или динамических полей. Сложный кастомный контрол, реализован для отображения таблиц тарифов.
- **Json-table** - сложный кастомный контрол, реализован для функционала настройки уведомлений.

#### 2.1.1.6. Поддержка выражений и модификаторов

Функционал выражений и модификаторов предназначен для более гибкого управления отображением данных на форме.

В выражении можно указать значение другого поля в соответствии с правилами построения выражений: `{{some_control}}` , где `some_control` – это `attribute_name` другого контроля на форме.

- В выражениях поддерживаются специальные ключи, которые дают доступ к данным из других источников, например:
- `SELECTED_OBJECT` - идентификатор объекта, выбранного в таблице/списке
- `user.*` - данные о текущем пользователе. Доступны такие поля: `id`, `name`, `firstName`, `middleName`, `surName`, `login`, `office`, `office_id`, `position`, `phone`, `roles`, `token`, `email`
- `ROOT_ID` – `id` корневого хаба

В выражениях разрешено использовать различные модификаторы. Модификаторы преобразуют вывод значения поля. Модификаторы указываются через символ "|" после имени поля, например: `{{some_control | number}}`. При значении `some_control = '1234567'` вывод будет такой: `'1 234 567.00'`.

#### 2.1.1.6.1. Виды модификаторов

##### 2.1.1.6.1.1. Условия

**is-not-empty** - проверяет является ли значение не пустым - если да, то вернётся 1, иначе - 0

Если `field1` не существует, то `{{ field1 | is-not-empty }}` вернёт 0; если `field2 === null`, то `{{ field2 | is-not-empty }}` вернёт 0; если `field3 === ""`, то `{{ field3 | is-not-empty }}` вернёт 0; если `field4 === "some-value"`, то `{{ field4 | is-not-empty }}` вернёт 1

**is-set** - проверяет существует ли значение у указанного контроля - если да, то вернётся 1, иначе - 0

Если `field1` не существует, то `{{ field1 | is-set }}` вернёт 0; если `field2 === null`, то `{{ field2 | is-set }}` вернёт 0; если `field3 === "some-value"`, то `{{ field3 | is-set }}` вернёт 1

**if** - простейший условный оператор: принимает на вход 3 значения - условие, значение 1 и значение 2. Если условие истинно/существует, то возвращает значение 1; иначе - возвращает значение 2. `this` - полученное модификатором значение.

Если `value === 3`, то `{{ value | if:2:"value is equal to two!":"value is not equal to two..." }}` вернёт `"value is not equal to two..."`

**or** - возвращает STATEMENT если значение falsy: Если выполняется над нулевым значением (`null`, `undefined`, `"`) или над выражением с ошибкой (типа `"x + undefined = z"`, по которому видно что из-за ошибки одно значение не удалось получить), то возвращает STATEMENT.

Если `field1` не существует, то `{{ field1 | or:'error' }}` вернёт `"error"`; если `field2 === null`, то `{{ field2 | or:'error' }}` вернёт `"error"`; если `field3 === "some-value"`, то `{{ field3 | or:'error' }}` вернёт `"some-value"`

**switch** - комплексный условный оператор: принимает на вход список кейсов в виде пар условие-значение, включая кейс `default`, представляющий ситуацию, когда ни одно из перечисленных условий не выполнилось.

Выражение `{{ x | switch:1:"uno":2:"dos":default:"extraviado" }}` вернёт "uno" если `x === 1`; "dos" если `x === 2`; в остальных случаях - "extraviado"

#### 2.1.1.6.1.2. Даты

**date** - преобразовывает дату в удобный для просмотра вид. Если передать `true`, то вернёт дату без времени.

Если `my_date === "2014-02-27T10:00:00"`, то `{{ my_date | date }}` вернёт "27.02.2014 10:00", а `{{ my_date | date:true }}` вернёт "27.02.2014"

**addPeriod** - добавляет период к дате по коду.

Если `my_date === "2014-02-27T10:00:00"`, то `{{ my_date | addPeriod:12M | date }}` вернёт "2015-02-27T10:00:00"

**moment\_get** - получает для даты количество единиц, указанных в `unit` - годы, месяцы, часы и тд.

Например, если `date === 2023-04-21T13:51:27.677`, то выражение `{{ date | moment_get:year }}` вернёт 2023, а `{{ date | moment_get:month }}` - 4.

**moment\_add** / **moment\_subtract** - добавить к дате/отнять от даты указанное количество единиц. Например, если `my_date === "2023-04-21T13:51:27.677"`, то `{{ my_date | moment_add:2:days | date }}` вернёт "23.04.2023 13:51"

#### 2.1.1.6.1.3. Строки

**displayPeriod** - функция для отображения периода в читабельном виде, принимает на вход код периода вида 20D, 3M, 12M.

Если `period_code === '12M'`, то выражение `{{ period_code | displayPeriod }}` вернёт "12 мес."

**trim** - удаляет пробелы в начале и конце строки. Если `value === " x "`, то `{{ value | trim }}` вернёт "x"

**split** - разбивает строку по разделителю. В качестве разделителя можно указать RegExp.

Если `my_string === "строка, с, запятыми"`, то `{{ my_string | split:',' }}` вернёт [ "строка", " с", " запятыми" ], а `{{ my_string | split:'\s?' }}` вернёт [ "строка", "с", "запятыми" ]

**replace** - ищет соответствия и заменяет их. Для поиска соответствия используется регулярное выражение. Например, если `x === "aclm14110"`, то выражение `{{ x | replace:1:2 }}` вернёт 'aclm24220', а выражение `{{ x | replace:\d+:2 }}` вернёт 'aclm2'. Регулярное выражение создаётся сразу с ключом `g` - global; спецсимволы должны быть экранированы.

**includes** - возвращает true/false если множества A и B пересекаются, то есть все или несколько элементов совпадают. Например, если `a = '123'`, `b = '234'` и `c = '987'`, то `{{ a | split: "" | includes:({{ b | split: "" }}) }}` вернёт true, а `{{ a | split: "" | includes:({{ c | split: "" }}) }}` вернёт false.

**number** - приводит строку представляющую число к стандартному формату отображения чисел. По умолчанию количество знаков после запятой - 2. Также принимает на вход количество знаков после запятой.

Если `x === "12345.14159127369"`, то `{{ x | number }}` вернёт `"12 345.14"`, а `{{ x | number:0 }}` вернёт `"12 345"`

**to-number** - приводит строку к числу. Умеет парсить float.

Если `x === "3.14"`, то `{{ x | to-number }}` вернёт `3`, а `{{ x | to-number:true }}` вернёт `3.14`

#### 2.1.1.6.1.4. Списки / массивы / таблицы

**selection** - возвращает выбранные в таблице строки. На данный момент актуально только для object-list и multiple-table-select.

**length** - возвращает количество элементов списка/массива. Также можно применить и к строке.

Если в таблице `my_table` 17 элементов, то `{{ my_table | length }}` вернёт `"17"`

**sumBy** - суммирует все строки списка/массива по конкретному полю.

**for-each** - выполняет выражение для каждого элемента массива. `this` - текущий элемент.

Если `array === [ '1', '2', '3' ]`, то `{{ array | for-each:(= {{ this }} =) }}` вернёт `[ '_=1=-_', '_=2=-_', '_=3=-_' ]`

**get-item** - получает из списка/массива элемент по индексу.

Пример: `{{ table | get-item:0 }}` вернёт первую строку.

**pick** - собирает все значения конкретного поля из каждого элемента списка/массива и возвращает в виде списка примитивных значений.

Если в строке таблицы `table` есть поле `id`, то `{{ table | pick:id }}` вернёт

```
[  
  {значение поля id из первой строки},  
  {значение поля id из второй строки},  
  ...  
].
```

**map** - собирает нужные значения из каждого элемента списка/массива и возвращает в виде списка объектов. Нужен для того чтобы отобрать из списка/массива конкретные поля. `this` - текущий элемент списка.

Пример: `{{ table | map:id:hub_Client.id:label:hub_Client.Name }}` вернёт

```
[  
  {  
    id: {значение поля hub_Client.id из первой строки},  
    label: {значение поля hub_Client.Name из первой строки},  
  },  
  ...  
].
```

```

{
  id: {значение поля hub_Client.id из второй строки},
  label: {значение поля hub_Client.Name из второй строки},
}
...
]

```

**find** - ищет элемент, соответствующий критерию, то есть возвращает элемент списка, для которого переданное выражение-условие вернуло true. **this** - текущий элемент списка.

Пример: если в таблице `clients` перечислены клиенты, для каждого из которых существует таблица счетов `accounts`, то например с помощью `find` можно проверить есть ли такой клиент у которого нету счетов - `{{ clients | find:(this | get-property:accounts | length | if:0:true:false) | is-set }}` вернёт 1 если в таблице `clients` есть клиенты без счетов.

**filter** - ищет элементы, соответствующие критерию, то есть возвращает элементы списка для которых переданное выражение-условие вернуло true. Аналогичен модификатору `find`, только возвращает множество элементов, а не один элемент. **this** - текущий элемент списка.

Пример: если `pin === 1000008`, то `{{ pin | split:"" | filter:(this | if:'0':true:false) | length }}` вернёт 5. То есть берём строку, делим на символы, получаем список символов, и фильтруем по критерию "если элемент равен '0', то true, иначе false", .

**join** - объединяет элементы массива в строку, между элементами будет помещен указанный пользователем разделитель.

Если `array === [ 1, 2, 3 ]`, то `{{ array | join:"|" }}` вернёт "1|2|3"

#### 2.1.1.6.1.5. Объекты

**get-property** - возвращает значение запрашиваемого атрибута объекта. Чаще всего используется в связке с `get-item`: `get-item` отбирает нужный объект из списка, а `get-property` отбирает у него нужный атрибут.

Если в первом элементе массива `arr` есть свойство `id` со значением 1234, то `{{ arr | get-item:0 | get-property:id }}` вернёт "1234"

#### 2.1.1.6.1.6. Специальные

**get-file** - возвращает объект файла, выбранный в контроле file. Файл имеет такие поля:

name - имя файла

size - размер, в байтах

type - MIME-тип файла

content - содержимое файла в виде строки

Чтобы получить, например, содержимое файла нужно написать такое выражение: `{{ my-file | get-file | get-property:content }}`

**get-policy-decision** - возвращает true/false если объект доступен/не доступен пользователю. Объекты разбиты на группы - можно запросить разрешение объекта из той или иной группы, на данный момент доступная группа только одна - forms. Например, если значение контроля test === 'form\_JL', то выражение `{{ test | get-policy-decision:forms }}` вернёт true если форма form\_JL доступна пользователю.

**get-status** - возвращает один из статусов формы. Существующие статусы:

changed

unchanged

valid

invalid

Например, если текущая форма валидна и не изменена пользователем, то `{{ this | get-status:valid }}` и `{{ this | get-status:changed }}` вернут true и false.

#### 2.1.1.6.1.7. Самостоятельные

**get-context** - возвращает текущий контекст формы: FL или UL (реализовывалось для раздельных БД физических и юридических лиц).

Пример: выражение `{{ get-context }}` вернёт текущий контекст, если он у формы задан.

#### 2.1.1.7. Предварительный просмотр формы

Данная функциональность позволяет просматривать получаемый результат по мере добавления различных элементов дизайна на форму без специального запуска формы в приложении или запуска бизнес-процесса.

При предварительном просмотре форма будет отображаться без данных, если у контролов не заданы константные значения через специальные параметры.

Функционал доступен в редакторе в закладке «Предварительный просмотр».

#### 2.1.1.8. Хендлеры

Данный функционал позволяет с помощью типовых процедур получать данные на форме или выполнять определенные действия на форме. Такие процедуры выполняются при наступлении событий, перечисленных в списке триггеров; в качестве триггера можно использовать одно из фиксированных событий или имя поля. Если триггер - имя поля, то хендлер выполнится при изменении пользователем этого поля (при условии, что значение поля установлено).

Для вызова хендлера необходимо выбрать вид хендлера из списка возможных, задать триггер (также из списка), параметры хендлера, при необходимости – условия выполнения хендлера.

Фиксированные события триггера:

- **Сборка формы** – событие сборки формы, это самое раннее событие, происходящее в процедуре обработки формы, до любого другого;
- **Инициализация формы** – событие инициализации формы, выполняется после сборки формы и подхватывает значения, заданные контролам в форме;
- **Возврат к форме** – событие возврата к форме;
- **Нажатие кнопки «Назад»** - нажали на кнопку Назад

Контролы, которые могут выступать в качестве триггера:

- Action
- Select
- Dynamic-field
- (Другие изменяемые контролы)

Параметр «Имя хендлера» для всех видов хендлеров опциональный и задается тогда, когда данный хендлер нужно вызвать из другого хендлера. Другие параметры хендлеров задаются в зависимости от вида хендлеров. Также, любой из параметров хендлера может быть выражением.

Условия выполнения – это дополнительные условия, при выполнении которых хендлер будет выполняться. Этот параметр позволяет более гибко управлять запуском хендлеров.

#### 2.1.1.8.1. Виды хендлеров

##### 2.1.1.8.1.1. Хендлеры вне групп

**Вывод сообщения** – выводит на экран сообщение. Если не задан текст сообщения, то сообщение выведено не будет, а в консоль будет выведено соответствующее предупреждение

Параметры:

Тип – тип сообщения, поддерживается 5 типов сообщений: success, error, warning, info, show

Текст – текст сообщения

Заголовок – заголовок окошка, в котором выводится текст сообщения

**Вывод модального окна** - отображает на экране модальное окно, которое в зависимости от указанного типа ждёт от пользователя того или иного действия. От действия

пользователя зависит дальнейшая обработка: если этот хендлер используется перед выполнением операции и пользователь изъявил отказ, то целевой хендлер выполнение операции выполнен не будет. Специальный тип custom позволяет использовать собственный шаблон для модального окна, для чего нужно предварительно создать контейнер modal с требуемым наполнением. В качестве заголовка модального окна используется заголовок, указанный в хендлере или заголовок указанного контрола modal. Соответственно, если нет ни того, ни другого, то у модального окна не будет заголовка. Для закрытия модального окна в ней нужно создать кнопки с соответствующими ролями - Кнопка согласия и Кнопка отказа. Клик вне области модального окна также закроет её с отрицательным решением.

Параметры:

- Тип – тип модального окна, поддерживается 3 типа: alert, confirm и custom
  - Alert – информационное окно с одной кнопкой «ОК»;
  - Confirm – окно с текстом (вопроса) и двумя кнопками «Да» и «Нет»;
  - Custom – модальное окно, которое для отображения использует контрол типа modal.
- Заголовок – заголовок окна
- Текст – текст, который будет отображаться в модальном окне (для alert и confirm)
- Модальное окно – (для custom) имя контрола modal из формы

Модификация и сохранение данных в форму для дальнейшего использования другими контролами – используется в связке с выполнением пользовательского скрипта.

Параметры:

- Имя создаваемого контрола, куда будут сложены данные
- Модификация входных значений

**Выполнение пользовательского скрипта** – позволяет выполнять скрипты, написанные пользователями, для какого-либо преобразования данных, которое невозможно реализовать простыми средствами редактора.

Параметры:

- Скрипт – окно редактирования пользовательского скрипта. Поддерживается JavaScript.
- **Прокрутка страницы** – прокручивает страницу до указанного поля
- Параметры:
  - Целевое поле – имя поля, до которого будет прокручена страница. После прокрутки это поле будет видно пользователю
  - Выравнивание по вертикали – тип выравнивания, возможно 4 типа:
    - Элемент будет расположен в центре (center)
    - Элемент будет расположен в конце страницы, в самом низу (end)
    - Докрутить до ближайшего края элемента (nearest)
    - Элемент будет расположен в начале страницы, в самом верху (start)

#### 2.1.1.8.1.2. Группа «Формы»

В данном разделе объединены хендлеры, предназначенные для взаимодействия между формами.

**Открытие формы** – открывает форму объекта по заданным параметрам. Работает с контролем action и пока не работает с object- list и link.

Параметры:

- **Идентификатор объекта** – идентификатор объекта, который будет использован для загрузки данных в форму. Поддерживает выражения. Если не задан, то шаблон не будет обогащен данными объекта.
- **Шаблон** – имя шаблона, который будет использован в качестве новой формы.
- **Способ открытия формы** – на данный момент существует три способа открытия формы:
  - В качестве самостоятельной формы (по умолчанию, также можно задать как `_form`)
  - В модальном окне (задается через `_modal`)
  - В контейнере (указывается имя контейнера, созданного в данной форме; контейнеры появятся в выпадающем списке)
- **Экспорт значений** – в открываемую форму можно перенести значения из текущей формы, для этого надо добавить нужные поля в список для экспорта и назначить им имя в целевой форме. Например, если выбрать для экспорта поле `test-field` и назначить ему имя `imported-test-field`, то при открытии целевой формы в ней появится скрытое поле `imported-test-field` с тем значением, которое было у поля `test-field` в предыдущей форме. Если при импорте поля окажется что такое поле в форме уже есть, то его значение будет перезаписано, а в консоль будет выведено оповещение.
- **Таблица-источник** – для того чтобы открыть форму объекта из таблицы, нужно указать целевую таблицу. Это нужно только для того чтобы создать линк для нового объекта - если в таблице нет линка для открываемого объекта, то он будет создан, и его можно будет впоследствии сохранить стандартными средствами.
- **Строка таблицы** - если указана таблица-источник, то также нужно указать какую именно строку открываем. В качестве значения может быть только выражение. Если строка не найдена или не указана, то при открытии формы будет создан линк, который можно будет сохранить стандартными средствами.
- **ID хаба** - для того чтобы создать линк, нужно знать идентификаторы двух хабов, но в общем случае их нельзя узнать сразу, так как один из хабов, возможно, будет собран позже. Поэтому нужен способ указать переменный ID хаба. Как раз для этого и нужен этот параметр - указывается выражение, а при сохранении второй хаб нашего линка получит тот ID который получился в результате обработки выражения.
- **Имя хендлера** - устанавливается для того, чтобы можно было сослаться на конкретный хендлер. Опционально.

**Обновление формы** – обновляет форму. На данный момент варианта всего 2 можно обновить эту форму, ту в которой создан хендлер, или можно обновить текущую форму в цепочке форм.

Параметры:

- Целевая форма – два типа целевой формы:
  - Эта форма ( `_self` )
  - Текущая форма в цепочке открытых форм ( `_current` )

**Переход на форму / страницу** – совершает переход: либо на одну из страниц, перечисленных в меню; либо на одну из предыдущих форм в текущей цепочке форм; либо закрывает активные модальные окна; либо по конкретному URL, достроенному после текущего base. Если указан переход на предыдущую форму, но указанная форма не найдена, то переход осуществлён не будет, а в консоль будет выведено соответствующее сообщение.

Параметры:

- Тип перехода – поддерживается три типа перехода:
  - Предыдущая форма (previous-form) – переход к одной из предыдущих форм из цепочки форм;
  - Заккрытие модального окна (close-modal) – при указании данного типа 1) не обязательно заполнять target и 2) закрыты будут все активные модальные окна
  - Навигация (url) – переход по заданному адресу
- Экспорт значений - в открываемую форму можно перенести значения из текущей формы, для этого надо добавить нужные поля в список для экспорта и назначить им имя в целевой форме. Например, если выбрать для экспорта поле test-field и назначить ему имя imported-test-field, то при открытии целевой формы в ней появится скрытое поле imported-test-field с тем значением, которое было у поля test-field в предыдущей форме. Если при импорте поля окажется что такое поле в форме уже есть, то его значение будет перезаписано, а в консоль будет выведено оповещение.
- URL строка (с поддержкой выражений) – можно задать путь до конкретной формы, например: 'audit-ui/audit/operations/{{Client\_Id}}'
- Целевая форма/страница – для типа перехода url значение '\_blank' будет открывать ссылку в новой вкладке, пустое - в текущей.

#### 2.1.1.8.1.3. Группа «Операции»

В данном разделе объединены хендлеры, запускающие какие-либо операции.

**Выполнение операции** – выполняет указанную в параметрах операцию

Параметры хендлера:

- **Класс** – класс BOClass объекта из модели данных ITA Objects. В выпадающем списке отображаются все доступные классы
- **Операция** – доступные для выбранного класса объекта операции.
- **Идентификатор объекта** – идентификатор объекта, над которым выполняется операция.

- **Аргументы для операции** – набор входных аргументов для операции. В качестве значения любому аргументу можно указать выражение, по которому при выполнении операции будет рассчитано значение аргумента. Если при расчёте значения аргумента произойдет ошибка (например, в случае отсутствия поля, использованного в выражении), то это поле будет исключено из аргументов операции.
- **Перед выполнением** – перед выполнением операции можно выполнить те или иные действия, представленные набором хендлеров. Ошибка или отрицательный результат при обработке этих хендлеров остановит цепочку выполнения: ни последующие хендлеры, ни сама операция не будут выполнены.
- **Критерии вывода на успех/ошибку** – после выполнения операции, её ответ не всегда представляет собой положительный результат. Иногда он может быть вполне успешным, но говорящим о том, что на процессе есть ошибка и, как пример, должен не давать возможность пропустить процесс далее. Этот параметр служит как раз регулятором того, на какой случай мы должны наткнуться «успех / ошибка / другое» в случае выполнения условия. Пример условия `{{ response | get- property: clientStateList | find:(this | get- property: error) }}` - говорит о том, что в ответе операции есть массив `clientStateList`, который ищет поле `error` в любом из его элементов. Если такое поле находится, то ивент бросается в либо успех, либо ошибку, либо другое
- **После выполнения** – после выполнения операции, в зависимости от результата, можно выполнить те или иные действия, представленные набором хендлеров. Для этого нужно задать хендлерам имена и потом указать их в поле требуемого результата «успех», «ошибка» или другого. Например, можно вывести сообщение об успешном выполнении операции и вернуться на заданную форму: для этого нужно создать два соответствующих хендлера и указать их последовательно в данном параметре.

Всем хендлерам, выполняемым в рамках этой операции, ответ операции доступен по ключу `response`. Например, если в ответе есть поле `message`, то оно доступно по ключу `response.message`.

**READER** – операция чтения данных и сборки формы. Для этого хендлера не нужны триггеры, но он выполняется при сборке формы. Если на форме в разделе «Паттерн» не задан паттерн, то будет использован паттерн текущего шаблона.

Параметры хендлера:

- **Класс** – класс `BOClass` объекта из модели данных `ita-objects`. В выпадающем списке отображаются все доступные для чтения классы
- **Операция** – список доступных операций для выбранного класса. Нужно выбирать операцию, которая возвращает в ответе `json` в структуре `HLS`
- **Идентификатор объекта** – идентификатор объекта над которым выполняется операция. Если не задан, то будет использован идентификатор, с которым была вызвана форма. Например, в случае с виджетом идентификатор указывается в параметре `object-bocode` самого поля `widget`, и при сборке формы виджета **READER** получит этот идентификатор
- **Аргументы для операции** – набор входных аргументов для операции. В качестве значения любому аргументу можно указать выражение, по которому при

выполнении операции будет рассчитано значение аргумента. Если при расчёте значения аргумента произойдет ошибка (например, в случае отсутствия поля, использованного в выражении), то это поле будет исключено из аргументов операции. Поддержка выражений – такая же, как и в хендлере Выполнение операций

- После выполнения – функционал действий в зависимости от результата выполнения взят из хендлера Выполнения операций.

**SAVER** – Операция сохранения данных и сборки формы. В случае, если параметры не заполнены, будут использованы значения по умолчанию.

Параметры хендлера:

- Класс – по умолчанию Project, но можно также выбрать класс из доступных BOClass модели
- Операция – по умолчанию DVSAver, но также можно выбрать операцию
- Идентификатор – идентификатор объекта, над которым выполняется сохранение

#### 2.1.1.8.1.4. Группа «Справочники»

В данном разделе объединены хендлеры для работы со справочниками. Под справочниками подразумеваются справочники, содержащиеся в модуле ITA Info, который является составной частью ITA FORMS.

**Наполнение селекта из справочника** – загружает для указанного поля опции из справочников. Допустимые типы поля для данного хендлера: select, multiple-select, dual-select, radio-button, solid-radio-button, chips-select, stepper

Параметры:

- **Справочник** – код справочника из справочной системы ITA Info. Можно вписать выражение, по которому будет рассчитан код справочника.
- **Лейбл** – имя атрибута справочной записи, значение которого будет использовано в качестве лейбла, отображаемого на форме. Выбор в выпадающем списке происходит из атрибутов справочников в справочной системе ITA Info. По умолчанию выбран label
- **Ключ** – имя атрибута справочной записи, значение которого будет использовано в качестве ключа (сохраняемое и передаваемое значение). Выбор в выпадающем списке происходит из атрибутов справочников в справочной системе ITA Info. По умолчанию выбран refKey
- **Фильтр** – задание условия для фильтра данных из справочника. Можно выбрать атрибут справочника и указать критерий
- **Установить значением первый элемент** – если установлен, то первая запись справочника будет автоматически выбрана в поле (если нет – то поле будет пустым, пока что-нибудь не выберем)
- **Атрибут для сортировки** – имя атрибута справочной записи, по которому будет производиться сортировка. По умолчанию выбран атрибут showOrder
- **Сортировать по убыванию** – по умолчанию, когда данный параметр не установлен, сортировка выполняется по возрастанию. Ее можно изменить, установив данный параметр в «Да».
- **Поле** – имя поля на форме, для которого будут загружены опции.

**Наполнение контрола значением из справочника** – загружает в указанное поле значение из справочника. Данный хендлер нужен, чтобы для ключевого значения (например, какого-то кода) найти читабельную расшифровку в справочнике и отобразить на форме. Для данного хендлера допустим тип поля input.

Параметры:

- Справочник – код справочника из справочной системы ITA Info. Можно вписать выражение, по которому будет рассчитан код справочника.
- Атрибут для вывода – имя атрибута справочной записи, значение которого будет выведено в целевой контрол. По умолчанию выбран label
- Идентификатор записи – идентификатор записи, то есть на самом деле ее refKey. То есть здесь указывается имя контрола (в `{{}}`), значение которого будет являться refKey в заданном справочнике, и по которому будет выполняться поиск атрибута для вывода.
- Целевое поле – имя поля, в которое будет загружено значение

#### 2.1.1.8.1.5. Таблицы

В данном разделе объединены хендлеры для манипуляций с данными в таблицах, которые невозможно выполнить стандартными способами в редакторе

**Добавление объектов** – добавляет объекты в таблицу из списка объектов. Используется в multiple-table- select, simple-table, object-table, address-table

Хендлер принимает на вход имя целевой таблицы и имя формы отбора: при выполнении этого хендлера будет отображена форма отбора, пользователь сможет выбрать объекты и нажать на кнопку отбора, после чего выбранные объекты будут добавлены в таблицу.

Параметры:

- Целевая таблица – таблица, в которую будут добавлены объекты. Важно: так как для получения данных реестра используется паттерн, то в шаблоне целевой таблицы должны быть использованы атрибуты именно этого паттерна, а не атрибуты хаба. Иначе нужно указать связь полей паттерна с полями HLS-объекта
- Форма отбора – имя шаблона, который будет использоваться в качестве формы отбора
- Маппинг полей – так как для получения данных используется паттерн, то для успешного создания HS-объекта нужно указать с каким атрибутом HLS-объекта связан атрибут паттерна: pattern -> HLS. Нужно только если поля целевой таблицы непосредственно представляют атрибуты HS-объекта.
- Источник объектов – таблица или массив, из которой будут получены объекты. Специфическая настройка, нужная для тех случаев, когда список объектов полученный из целевого ita-object-list нуждается в пользовательской обработке перед добавлением

#### 2.1.1.8.1.6. Виджеты

В данном разделе сгруппированы хендлеры для взаимодействия с виджетами.

**Обновление данных виджета** – обновляет данные виджета. Список объектов (ita-object-list) в том числе является виджетом, поэтому и его можно обновить при желании.

Также возможно обновить родительский виджет, в случае применения этой функции будет найден ближайший родитель-виджет и обновлен.

Параметры:

- Виджеты – список доступных полей-виджетов на форме. Можно выбрать из выпадающего списка, или в некоторых случаях написать вручную
- Обновить родительский виджет – при установке в «Да» происходит поиск и обновление ближайшего виджета-родителя.

**Импорт данных в файл** – выгружает данные контрола в файл. На данный момент хендлер применим только к ita-object-list.

Параметры:

- Имя контрола – имя контрола ita-object-list, данные из которого который предполагается импортировать в файл
- Формат выгрузки – на данный момент возможна только выгрузка в csv-файл

**Удаление бизнес-объекта** – проставляет бизнес-объекту все атрибуты, соответствующие удалённому состоянию. Если аргументы не заданы, то проставляет атрибуты корневому для текущей формы бизнес-объекту. Для фактического удаления после проставления атрибутов требуется сохранить объект.

Параметры:

- Класс – по умолчанию Project
- Операция – по умолчанию DVSAver
- Идентификатор объекта – идентификатор объекта, над которым выполняется операция
- Перед выполнением – перед выполнением операции можно выполнить те или иные действия, представленные набором хендлеров. Ошибка или отрицательный результат при обработке этих хендлеров остановит цепочку выполнений ни последующие хендлеры, ни сама операция выполнены не будут.

#### 2.1.1.9. Паттерн формы

Данный функционал позволяет задать источник данных для определенной формы.

Паттерн – это описание структуры данных в формате DataVault (HLS - хабов / сателлитов / линков). В такой структуре есть один главный хаб. Для работы паттерна ID этого главного хаба должен быть каким-то образом задан: можно задать константой, а можно передать динамически.

Пример паттерна с указанием hub\_id в параметре BOCode:

```
{
  "Pattern": {
    "hub_MPShowcase": {
      "sat_MPShowcase_Main": {},
      "link_MPVersion_MPShowcase": {
```

```
"hub_MPVersion": {  
    "sat_MPVersion_Main": {},  
    "link_MPField_MPVersion": {  
        "hub_MPField": {  
            "sat_MPField_Main": {}  
        }  
    }  
},  
"BOCode": "32b37788-2ff2-4431-9931-eca3c9d7879e"  
}
```

#### 2.1.1.10. Параметры формы

Данный функционал позволяет дополнительно установить для формы глобальные параметры.

В настоящий момент существуют следующие глобальные параметры:

- **Лейбл для кнопки «Назад»** - позволяет подменять название стандартной кнопки «Назад» на другое произвольное название.
- **Лейбл для breadcrumbs** – этот лейбл будет использоваться для отображения имени формы в строке пути - breadcrumbs. При его отсутствии будет использован основной лейбл шаблона формы.
- **Контекст обращений ФЛ / ЮЛ** – специальный параметр для запросов в отдельные базы данных для физических лиц или юридических лиц. Все хендлеры, виджеты, реестры добавляют в запрос этот параметр, указывающий на контекст - ФЛ или ЮЛ.
- **Использовать данные родительской формы** – используется, когда данные родительской формы еще не хранятся на сервере.
- **Использовать Id корневого объекта родительской формы** – используется, когда данные родительской формы еще не хранятся на сервере.

Добавление и удаление глобальных параметров на форме выполняется с помощью соответствующих функциональных кнопок.

#### 2.1.2. Множественная загрузка форм

Данный функционал предназначен для групповой загрузки форм на разные стенды, массовой выгрузки форм и схем бизнес – процессов для последующей загрузки на другой стенд, поиска форм.

Поддерживаются следующие возможности:

- **Глобальный поиск** – данная функциональность позволяет осуществлять поиск по всем зарегистрированным формам. Поиск осуществляется по всем атрибутам формы - по имени, лейблу, содержимому полей и т. д. Кликните по имени формы чтобы открыть ее в редакторе; по клику на имя поля форма также откроется в редакторе - поле будет подсвечено.
- **Загрузить форму / формы** – данная функциональность позволяет загружать форму или несколько форм из файлов .json или .txt

- **Скачать форму / формы** – данная функциональность позволяет выгружать форму или несколько форм в файлы .json или .txt для последующей загрузки на другом стенде.
- **Загрузить процесс / процессы с формами и скриптами** – данная функциональность позволяет загружать bpmn-схему бизнес – процесса вместе с файлами форм одним zip-архивом.
- **Скачать процесс / процессы с формами и скриптами** – данная функциональность позволяет скачивать в виде единого zip-архива bpmn-схемы бизнес – процессов вместе с файлами форм.

### 2.1.3. Описание объектов DataVault

В данном разделе предоставлена возможность просматривать бизнес-объекты системы в виде модели DataVault.

DataVault — это модель хранения данных, которая представляет собой детализированный, исторически отслеживаемый и уникально связанный набор нормализованных таблиц, поддерживающий одну или несколько функциональных областей бизнеса. DataVault организует данные вокруг бизнес-процессов и функций, а не отдельных предметных областей. Модель Data Vault содержит временные отметки размещения данных и позволяет проследить изменение хранимой информации во времени. Структура Data Vault включает следующие компоненты:

- **Хаб (Hub)** — единичная бизнес-сущность, например: клиент, счет, пакет услуг. Содержит уникальный бизнес-ключ, дату загрузки и источник записи.
- **Связь (Link)** — связь между сущностями (хабами). Обеспечивает транзакционную интеграцию между ними.
- **Сателлит (Satellite)** — описательные атрибуты сущностей (хабов) и связей. Содержит суррогатный ключ родителя, значения атрибутов, дату начала действия версии, дату загрузки и источник записи.

Поддерживаются следующие функции:

- Поиск хабов, линков, саттелитов
- Просмотр списка хабов, линков, саттелитов
- Просмотр содержимого выбранного хаба, линка, саттелита

#### 2.1.3.1. Поиск хабов, линков, саттелитов

Поиск возможен по коду хабов, линков, саттелитов (поиск происходит в той БД, что указана как связанная).

#### 2.1.3.2. Просмотр списка хабов, линков, саттелитов

При входе в раздел «Объекты DataVault» первоначально отображается первый уровень – список хабов (той БД, что указана как связанная).

Пользователь через интерфейс имеет возможность раскрыть первый уровень, «провалиться» в хаб и увидеть список саттелитов хаба, а также список линков (той БД, что указана как связанная).

#### 2.1.3.3. Просмотр содержимого выбранного хаба, линка, спутника

При активации (выборе мышкой) конкретного хаба, линка, спутника пользователю доступен просмотр содержимого (список атрибутов).

По каждому атрибуту выводится информация:

- Код
- Комментарий
- Тип
- Подтип
- Длина поля

#### 2.1.4. Редактор бизнес – объектов

В данном разделе предоставлена возможность просматривать реестр бизнес-объектов в виде дерева. При раскрытии узла пользователю становятся доступны описания модели бизнес-объекта и операции, которые можно выполнять над бизнес-объектом.

Поддерживаются следующие функции:

- Поиск бизнес-объекта
- Просмотр моделей бизнес-объектов
- Просмотр операций
- Просмотр обработчиков
- Установка темы редактора

##### 2.1.4.1.1. Просмотр моделей бизнес-объектов

Модели бизнес-объектов в дереве визуально отмечены значком 

При активации (выборе мышкой) конкретной модели в дереве (левая часть экрана) в правой части экрана выводится код модели для просмотра с возможностью прокрутки по горизонтали/вертикали.

##### 2.1.4.1.2. Просмотр операций

Операции над бизнес-объектами в дереве визуально отмечены значком 

При активации (выборе мышкой) конкретной операции в дереве (левая часть экрана) в правой части экрана выводится код операции для просмотра с возможностью прокрутки по горизонтали/вертикали.

##### 2.1.4.1.3. Просмотр обработчиков

Обработчики бизнес-объектов в дереве визуально отмечены значком 

При активации (выборе мышкой) конкретного обработчика в дереве (левая часть экрана) в правой части экрана выводится код операции для просмотра с возможностью прокрутки по горизонтали/вертикали.

#### 2.1.4.1.4. Установка темы редактора

При просмотре моделей бизнес-объектов, обработчиков и операций возможно изменить тему редактора.

Возможные варианты:

- idea
- cobalt
- eclipse
- material

#### 2.1.5. Редактор бизнес-процессов

В редакторе бизнес – процессов пользователь с помощью графического интерфейса может настроить схему бизнес – процесса в нотации BPMN, настроить свойства процесса и задач, необходимые для технического исполнения, выложить созданную схему на сервер выполнения бизнес-процессов на основе движков ita-bpm версии 7/версии 8.

В разделе Редактор бизнес-процессов доступны следующие функции:

- поиск процессов по параметрам
- просмотр списка схем процессов
- настройка списка колонок для отображения списка процессов
- поиск версий процессов по параметрам
- просмотр версий процессов
- настройка списка колонок для отображения списка версий процессов
- загрузка файла BPMN
- выгрузка списка схем процессов в Excel
- создание схемы нового процесса
- редактирование схемы процесса
- настройка свойств процесса
- настройка задач процесса
- привязка экранных форм, которые конструируются в редакторе форм/разработанных вне редактора форм, к пользовательским задачам (User Task)
- сохранение черновика процесса
- деплой процесса на сервер выполнения бизнес-процессов на основе движков ita-bpm версии 7/версии 8
- возможность скачать схему процесса как файл BPMN
- возможность скачать схему процесса как файл SVG
- вывод на экран списка «горячих» клавиш для действий при моделировании процесса
- смена цветового режима редактора
- переход в полноэкранный режим
- приближение/удаление изображения схемы процесса
- вписать схему в окно
- удаление процесса

#### 2.1.5.1. Поиск процессов по параметрам

При входе в раздел Редактор бизнес-процессов отображаются созданные схемы процессов.

Для поиска/отбора схем процессов необходимо воспользоваться панелью фильтрации. Установить значение в предустановленных фильтрах либо добавить дополнительный фильтр из списка доступных.

#### 2.1.5.2. Просмотр списка схем процессов

При входе в раздел Редактор бизнес-процессов отображаются созданные схемы процессов.

По умолчанию выводятся следующие колонки:

- Дата изменения
- Дата создания
- Ключ
- Кто изменил
- Кто создал
- Лицензия
- Название
- Описание процесса
- Хэш-сумма
- id определения процесса

Созданные схемы процессов выводятся со свернутым списком своих версий.

#### 2.1.5.3. Настройка колонок для отображения списка процессов

По умолчанию выводятся следующие колонки:

- Дата изменения
- Дата создания
- Ключ
- Кто изменил
- Кто создал
- Лицензия
- Название
- Описание процесса
- Хэш-сумма
- id определения процесса

Пользователь может выбрать только те колонки для отображения, что ему нужны.

Чтобы не выводить ту или иную колонку, необходимо нажать на иконку , где возможно снять выделение кнопкой мыши.

Дополнительно присутствуют настройки:

- Сохранять выбранную сортировку
- Сохранять выбранные колонки

- Сохранять фильтры
- Сохранять количество строк для отображения

#### 2.1.5.4. Поиск версий процессов по параметрам

Версии процесса отображаются в привязке к процессу.

Для поиска/отбора версий процессов необходимо развернуть список версий у конкретного процесса и нажать на значок воронки .

При нажатии на воронку будет выведено меню для выбора параметра для поиска и значения. При необходимости отбора по нескольким параметрам нужно нажать на текст/кнопку «Добавить условие» и повторить шаги.

#### 2.1.5.5. Просмотр версий процессов

Версии процесса отображаются в привязке к процессу.

При входе в раздел Редактор бизнес-процессов отображаются созданные схемы процессов со свернутым списком его версий. Для просмотра версий необходимо развернуть список, нажав стрелку  на строке процесса. При этом отобразятся все версии процесса.

Версия, у которой заполнен параметр «ID определения процесса» загружена на сервер выполнения бизнес-процессов на основе движков ita-bpm версии 7/версии 8 (задеплоена). Версии с незаполненным параметром являются черновиками.

#### 2.1.5.6. Настройка колонок для отображения списка версий процессов

По умолчанию выводятся следующие колонки:

- Версия
- Дата изменения
- Дата создания
- Ключ
- Кто изменил
- Кто создал
- Название
- Хэш-сумма
- id определения процесса

Пользователь может выбрать только те колонки для отображения, что ему нужны. Чтобы не выводить ту или иную колонку, необходимо снять выделение кнопкой мыши.

Дополнительно присутствуют настройки:

- Сохранять выбранную сортировку
- Сохранять выбранные колонки
- Сохранять фильтры
- Сохранять количество строк для отображения

#### 2.1.5.7. Загрузка файла BPMN

Возможно загрузить файл BPMN, используя кнопку на правой панели .

При нажатии кнопку открывается окно проводника для выбора файла для загрузки.

При успешной загрузке/при ошибке пользователю выводятся соответствующие информационные сообщения.

#### 2.1.5.8. Выгрузка списка схем процессов в Excel

Выгрузка в Excel производится при нажатии кнопки на правой панели .

Данный функционал предназначен для выгрузки в файл с расширением .xlsx. списка процессов, который был предварительно отфильтрован (т.е. в файл выгружается то, что отображается в списке процессов в момент нажатия кнопки Выгрузить в Excel).

#### 2.1.5.9. Создание схемы нового процесса

Инициация создания нового процесса происходит при нажатии кнопки  на правой панели. После указания ключа определения процесса осуществляется переход к экрану моделирования процесса.

Для моделирования процесса предлагается панель инструментов (символов), которые являются стандартными для нотации BPMN 2.0.

#### 2.1.5.10. Редактирование схемы процесса

Для перехода к редактированию необходимо:

1. Выделить строку процесса и сделать двойной клик по строке  
При этом на редактирование откроется последняя версия процесса (по номеру)
2. Выделить строку процесса и нажать кнопку редактирования на правой панели



При этом на редактирование откроется последняя версия процесса (по номеру)

3. Выделить строку с версией процесса и сделать двойной клик по строке  
При этом откроется выбранная версия процесса
4. Выделить строку с версией процесса и нажать кнопку редактирования на правой



панели

При этом откроется выбранная версия процесса

#### 2.1.5.11. Настройка свойств процесса

При моделировании процесса возможно делать следующие настройки:

1. Общие настройки
  - a. заполнение идентификатора процесса (является ключом определения процесса)
  - b. заполнение наименования процесса
  - c. тег версии
  - d. признак исполняемого процесса
  - e. настройка задач, выполняемых внешними сервисами
    - приоритет задачи
  - f. настройка задания
    - приоритет задания
  - g. настройка запуска процесса
    - группы пользователей, которым разрешено запускать процесс
    - список пользователей, которым разрешено запускать процесс

- h. настройка истории
    - срок жизни истории
  - i. настройка списка задач
    - процесс запускаемый
  - j. документирование
    - заполнение описания процесса
2. Настройки переменных
  3. Настройки листнеров (слушателей)
  4. Настройки расширений (добавление свойств)

#### 2.1.5.12. Настройка задач процесса

При моделировании процесса для пользовательских задач (User Task) возможно делать следующие настройки:

1. Общие настройки
  - a. Общие настройки
    - i. заполнение идентификатора задачи (соответствует ключу определения задачи)
    - ii. наименование задачи
  - b. Пользовательские настройки
    - i. исполнитель
    - ii. дата выполнения
    - iii. дата проверки
    - iv. приоритет
    - v. кандидаты на исполнение задачи (выбор из списка существующих пользователей)
    - vi. группы кандидатов на исполнение задачи (выбор из списка существующих групп)
  - c. Настройки асинхронности
    - i. асинхронно до
    - ii. асинхронно после
  - d. Документирование
    - i. заполнение описания задачи
2. Настройки листнеров (слушателей)
  - a. Слушатели исполнения задачи (старт, завершение)
  - b. Слушатели статуса задач (создание, назначение, выполнение, обновление, удаление, просрочка)
3. Внедрение полей
4. Входящие/исходящие параметры
5. Расширения

При моделировании процесса для сервисных задач (Service Task) возможно делать следующие настройки:

1. Общие настройки
  - a. заполнение идентификатора задачи
  - b. наименование задачи
2. Имплементация (исполнение)
3. Настройки асинхронности
  - a. асинхронно до

- b. асинхронно после
- 4. Документирование
  - a. заполнение описания задачи
- 5. Настройки листнеров (слушателей)
  - a. Слушатели исполнения задачи (старт, завершение)
- 6. Внедрение полей
- 7. Входящие/исходящие параметры
- 8. Расширения

При моделировании процесса для задач со скриптом (Script Task) возможно делать следующие настройки:

- 1. Общие настройки
  - a. заполнение идентификатора задачи
  - b. наименование задачи
- 2. Скрипт
  - a. формат скрипта
  - b. тип скрипта
  - c. результирующие переменные
- 3. Настройки асинхронности
  - a. асинхронно до
  - b. асинхронно после
- 4. Документирование
  - a. заполнение описания задачи

При моделировании процесса для задач вызова элемента (Call Activity) возможно делать следующие настройки:

- 1. Общие настройки
  - a. заполнение идентификатора задачи
  - b. наименование задачи
- 2. Вызываемый элемент
  - a. Тип вызываемого элемента
- 3. Бизнес-ключ (формат передачи ключа)
- 4. Настройки асинхронности
  - a. асинхронно до
  - b. асинхронно после
- 5. Документирование
  - a. заполнение описания задачи

#### 2.1.5.13. Привязка экранных форм

Привязка экранных форм, которые конструируются в редакторе форм/разработанных вне редактора форм, к пользовательским задачам (User Task) производится через настройку Расширения.

При создании пользовательской задачи (User Task) автоматически на вкладке Расширения созданы 3 параметра:

- **formName** (код формы)
- **usePreviousAssignee** (назначать задачу на того же исполнителя, что и предыдущая в процессе)
- **closeScreenFlowAfterComplete** (закрывать форму после выполнения задачи)

По умолчанию поле **formName** не заполнено. Варианты использования:

- если форма уже создана в редакторе форм – необходимо указать ее код
- если формы нет, но предполагается отрисовка в редакторе форм – в поле указать произвольный код, который далее будет использован для формы
- если предполагается использование внешней формы (разработана вне редактора форм) – необходимо указать ее код

По умолчанию поле **usePreviousAssignee** заполнено значением **true** (да). Необходимо скорректировать согласно логике процесса. Если на вкладке основных параметров заполнены поля «Кандидаты на исполнение задачи» и «Группы кандидаты на исполнение задачи» - значение параметра в данном случае должно быть установлено в **false**.

По умолчанию поле **closeScreenFlowAfterComplete** заполнено значением **false** (нет). Необходимо скорректировать согласно логике процесса.

#### 2.1.5.14. Сохранение черновика процесса

Для сохранения черновика процесса необходимо нажать кнопку  на правой панели. Кнопка становится активной при любом изменении процесса, открытого на редактирование/только что созданного. Черновик сохраняется с присвоением следующей версии процесса.

#### 2.1.5.15. Деплой процесса на сервер выполнения бизнес-процессов на основе движков ita-bpm версии 7/версии 8

Для деплоя процесса на сервер выполнения бизнес-процессов предназначена кнопка  на правой панели. Кнопка активна для схемы процесса, сохраненной как черновик и при этом не задеплойенной на сервер.

#### 2.1.5.16. Возможность скачать схему процесса как файл BPMN

Для сохранения схемы процесса в формате BPMN необходимо нажать кнопку  в нижней части экрана. Файл будет помещен в папку Загрузки/будет открыто окно для выбора папки сохранения – в зависимости от настроек браузера.

#### 2.1.5.17. Возможность скачать схему процесса как файл SVG

Для сохранения схемы процесса как файл SVG (в виде картинки) необходимо нажать кнопку  в нижней части экрана. Файл будет помещен в папку Загрузки/будет открыто окно для выбора папки сохранения – в зависимости от настроек браузера.

#### 2.1.5.18. Вывод на экран списка «горячих» клавиш для действий при моделировании процесса

Функционал позволяет увидеть список «горячих» клавиш для действий при моделировании процесса. Для вывода на экран информации необходимо нажать кнопку  в правой части экрана моделирования процесса.

#### 2.1.5.19. Смена цветового режима редактора

Функционал позволяет изменить цветовой режим редактора. Для изменения режима предназначена кнопка  в правой части экрана моделирования процесса.

#### 2.1.5.20. Переход в полноэкранный режим

Функционал позволяет отображать экран моделирования процесса на весь экран. Для перехода в полноэкранный режим предназначена кнопка  в правой части экрана моделирования процесса. Для возврата в обычный режим необходимо повторно нажать на эту же кнопку либо кнопку ESC на клавиатуре.

#### 2.1.5.21. Приближение/удаление схемы процесса

Функционал позволяет изменять масштаб отображения схемы процесса.   
Управляющие кнопки  расположены в правой нижней части экрана.

#### 2.1.5.22. Вписать схему в окно

Функционал позволяет помещать схему процесса полностью в экран. Управляющая кнопка расположена  в правой нижней части экрана.

#### 2.1.5.23. Удаление процесса

Функционал позволяет производить удаление:

- версии схемы процесса, которая не была загружена на сервер (не задеплоена)
- версии схемы процесса, которая была загружена на сервер (задеплоена) и не имеет запущенных экземпляров (после дополнительного подтверждения пользователем необходимости удаления). Схема процесса будет удалена и из раздела Редактор процессов и с сервера ITA-BPM.

При попытке удалить версию схемы процесса, которая была загружена на сервер (задеплоена) и имеет активные экземпляры процесса на экран будет выведено информационное сообщение о том, что удаление невозможно.

### 2.1.6. Редактор меню

В редакторе меню пользователь может управлять наполнением левого меню модулей, которые разработаны на платформе ITA::FORMS. В случае приобретения только продукта ITA-BPM, в редакторе меню присутствуют модули Администрирование, Дизайнер.

Доступен следующий функционал:

- Просмотр настроенных пунктов меню
- Настройка пунктов меню
- Добавление нового пункта меню
- Удаление пункта меню

#### 2.1.6.1. Просмотр настроенных пунктов меню

При входе в раздел Редактор меню отображаются настроенные пункты меню в разрезе модулей.

Для модуля Дизайнер это пункты:

- Редактор форм
- Множественная загрузка форм
- Объекты DataVault
- Редактор бизнес-объектов
- Редактор бизнес-процессов
- Редактор меню

Для модуля Администрирование это пункты:

- Пользователи
- Роли
- Политики Доступа
- Орг. структура
- Бизнес-процессы
- Список задач
- Список сервисных задач

#### 2.1.6.2. Настройка пунктов меню

Для настройки доступны следующие параметры:

- Лейбл (название пункта меню)
- Иконка
- Описание
- Тип (доступен для редактирования не всегда)
- URL (ссылка доступен для редактирования не всегда)
- Компонент (доступен для редактирования не всегда)

У некоторых пунктов настройки есть текстовые подсказки, они выводятся при наведении на значок .

Если Тип – Предусмотренный компонент – не редактируются поля Тип, URL и Компонент.

#### 2.1.6.3. Добавление нового пункта меню

Функционал добавления доступ при нажатии кнопки Добавить раздел в области настроенного меню нужного модуля.

При добавлении нового пункта меню необходимо заполнить следующие параметры:

- Лейбл (название пункта меню)
- Иконка
- Описание
- Тип (доступен для редактирования не всегда)
- URL (ссылка доступен для редактирования не всегда)

Иконки выбираются из списка доступных.

Возможные типы:

- Форма
- Предустановленный компонент
- Список подразделов

При выборе типа Список подразделов дополнительно необходимо добавить разделы-пункты меню

#### 2.1.6.4. Удаление пункта меню

Если раздел был добавлен пользователем (не предустановлен), его можно удалить. Функционал доступен при наведении указателя мыши на раздел и нажатия кнопки .

## 2.2.ITA-DESIGNER-EDITOR

Сервис предназначен для обеспечения работы пользовательского интерфейса ita-designer-ui при редактировании экранных форм, параметров сервисных задач и схем бизнес-процессов, в том числе:

- создание/изменение/удаление настроек сервисных задач в БД
- создание/изменение/удаление настроек экранных форм в БД
- создание/изменение/удаление черновиков схем бизнес-процессов
- публикация схем бизнес-процессов в процессном движке
- импорт из внешнего файла полного описания схемы бизнес-процесса, включая экранные формы и настройки сервисных задач

## 2.3.ITA-DESIGNER-VIEWER

Сервис предоставления форм и схем бизнес-процессов в интерфейсе редактирования ita-designer-ui и в режиме исполнения бизнес-процессов, в том числе:

- предоставление списков экранных форм
- предоставление списков схем бизнес-процессов
- предоставление схем бизнес-процессов
- предоставление экранных форм по идентификатору
- предоставление связей экранных форм и пользовательских задач в схемах бизнес-процессов
- полнотекстовый поиск по элементам экранных форм
- выгрузка экранных форм и схем бизнес-процессов в архивированном виде

## 2.4.ITA-ADMIN-UI

Интерфейс администратора содержит следующие разделы:

- **Пользователи** – позволяет сотруднику ознакомиться с текущим списком зарегистрированных пользователей и их атрибутами, а также *создать* нового пользователя или *отредактировать* уже созданного;
- **Роли**– позволяет найти и просмотреть информацию о созданных ролях и актуальной привязке пользователей к этой роли. А также создать новую или отредактировать уже существующую роль;
- **Политики доступа** – позволяет найти и просмотреть информацию об уже созданных и применяемых политиках для ролей в системе. А также, если нужно, создать новую или удалить старую;
- **Орг. структура** – позволяет найти и просмотреть информацию о уже созданных узлах структуры и добавленных в них пользователей, а также создать новые узлы дерева элементов орг. структуры и добавить в них пользователей.
- **Бизнес-процессы** - в разделе отображаются все процессы, которые были загружены (задеплоены) на сервер ITA-BPM
- **Список задач** - в разделе отображаются все активные пользовательские задачи (User Task)
- **Список сервисных задач** - в разделе отображаются все активные сервисные задачи (Service Task)

Допускается использование как внешнего (для ITA-BPM) Keycloak, так и внутреннего. Выбор указывается в переменной окружения.

### 2.4.1. Пользователи

В разделе обеспечивается выполнение следующих функций:

- просмотр списка заведенных пользователей
- поиск пользователей по параметрам
- заведение нового пользователя
- редактирование данных пользователя
- удаление пользователя
- выгрузка в Excel списка пользователей

#### 2.4.1.1. Просмотр списка заведенных пользователей

При входе в раздел Пользователи отображается список заведенных пользователей.

По умолчанию выводятся следующие колонки:

- Логин
- Наименование (фамилия + имя + отчество)
- Должность
- Орг. структура
- Подразделение тарификации
- Телефон

- Удален
- Email

На правой панели расположены кнопки действий, в верхней части экрана расположен фильтр для поиска и отбора.

#### 2.4.1.2. Настройка колонок для отображения списка процессов

Пользователь может выбрать только те колонки для отображения, что ему нужны.

Чтобы не выводить ту или иную колонку, необходимо нажать на иконку , где возможно снять выделение кнопкой мыши.

Дополнительно присутствуют настройки:

- Сохранять выбранную сортировку
- Сохранять выбранные колонки
- Сохранять фильтры
- Сохранять количество строк для отображения

#### 2.4.1.3. Поиск пользователей по параметрам

Для поиска/отбора пользователей можно воспользоваться панелью фильтров в верхней части экрана. При нажатии на текст/кнопку «Добавить условие» на экран выводится перечень фильтров для использования. После задания значений необходимо нажать кнопку «Отобразить».

Второй вариант поиска и отбора – использование значка воронки  в заголовке таблицы со списком пользователей.

При нажатии на воронку над каждым столбцом появится поле для ввода параметров отбора. При вводе данных в поля отбор производится сразу же, онлайн (без нажатия дополнительных кнопок).

#### 2.4.1.4. Заведение нового пользователя

Функционал доступен при нажатии кнопки  на правой панели. При нажатии пользователю предлагается форма для заполнения со следующим набором атрибутов:

- Логин
- Фамилия
- Имя
- Отчество
- Должность
- Орг. структура
- Подразделение тарификации
- Телефон
- Email
- Признак Удален

Также на форме есть блок со списком ролей пользователя. Назначение роли производится через соответствующий раздел меню, на данной форме отображаются назначенные роли.

На форме расположены кнопки:

- Отмена
- Удалить
- Сохранить

#### 2.4.1.5. Редактирование данных пользователя

Функционал доступен при нажатии кнопки  на правой панели либо выполнении двойного клика кнопкой мыши на строке пользователя. При нажатии пользователю открывается форма с заполненными данными пользователя со следующим набором атрибутов:

- Логин
- Фамилия
- Имя
- Отчество
- Должность
- Орг. структура
- Подразделение тарификации
- Телефон
- Email
- Признак Удален

На форме расположены кнопки:

- Отмена
- Удалить
- Сохранить

#### 2.4.1.6. Удаление пользователя

Удаление пользователя производится путем проставления соответствующего признака «Удален». Функционал доступен с формы создания/редактирования данных пользователя. При нажатии кнопки Удалить и подтверждения пользователем действия проставляется признак Удален.

#### 2.4.1.7. Выгрузка в Excel списка пользователей

Выгрузка в Excel производится при нажатии кнопки на правой панели . Второй вариант – нажатие кнопки , расположенной в верхней части заголовка таблицы со списком пользователей.

Данный функционал предназначен для выгрузки в файл с расширением .xlsx. списка пользователей, который был предварительно отфильтрован (т.е. в файл выгружается то, что отображается в списке пользователей в момент нажатия кнопки Выгрузить в Excel).

#### 2.4.2. Роли

В разделе обеспечивается выполнение следующих функций:

- просмотр списка заведенных ролей

- просмотр списка пользователей, которым назначена роль
- поиск роли по параметрам
- заведение новой роли
- редактирование данных роли
- назначение роли пользователям
- выгрузка в Excel списка ролей

#### 2.4.2.1. Просмотр списка заведенных ролей

При входе в раздел Роли отображается список заведенных ролей. Список пользователей, которым назначена та или иная роль, свернут.

По умолчанию выводятся следующие колонки:

- Код
- Наименование
- Путь

На правой панели расположены кнопки действий, в верхней части экрана расположен фильтр для поиска и отбора.

#### 2.4.2.2. Просмотр списка пользователей, которым назначена роль

Для просмотра списка ролей, которым назначена та или иная роль, необходимо нажать на иконку  на строке роли для раскрытия списка.

Список пользователей, которым назначена роль, отображается со следующим набором колонок:

- Должность
- Имя
- Имя пользователя
- Отчество
- Телефон
- Удален
- Фамилия
- Email
- Подразделение тарификации (по умолчанию не выводится, но можно поменять в настройке колонок для отображения)
- Орг. Структура (по умолчанию не выводится, но можно поменять в настройке колонок для отображения)

#### 2.4.2.3. Поиск роли по параметрам

Для поиска/отбора ролей можно воспользоваться панелью фильтров в верхней части экрана. При нажатии на текст/кнопку «Добавить условие» на экран выводится перечень фильтров для использования. После задания значений необходимо нажать кнопку «Отобразить».

Второй вариант поиска и отбора – использование значка воронки  в заголовке таблицы со списком ролей.

При нажатии на воронку над каждым столбцом появится поле для ввода параметров отбора. При вводе данных в поля отбор производится сразу же, онлайн (без нажатия дополнительных кнопок).

#### 2.4.2.4. Заведение новой роли

Функционал доступен при нажатии кнопки  на правой панели. При нажатии пользователю предлагается форма для заполнения со следующим набором атрибутов:

- Код
- Наименование
- Путь

На форме присутствуют кнопки:

- Отмена
- Сохранить

После сохранения новой роли возможно добавить пользователей, которым необходимо назначить роль.

Функционал доступен при нажатии кнопки  в двух вариантах:

- 1) Добавить нового – вариант, когда пользователь еще не заведен, первоначально будет предложено создать пользователя, а затем добавить в (назначить) роль.
- 2) Добавить существующего – выбор пользователя из списка

#### 2.4.2.5. Редактирование данных роли

Функционал доступен при выполнении двойного клика кнопкой мыши на строке роли. При нажатии пользователю открывается форма с заполненными данными роли со следующим набором атрибутов:

- Код
- Наименование
- Список пользователей, которым назначена роль

На форме присутствуют кнопки:

- Отмена
- Сохранить

#### 2.4.2.6. Назначение роли пользователям

Функционал доступен на форме создания/редактирования роли. При создании роли – после заполнения и сохранения основных полей формы.

Назначение роли возможно при нажатии кнопки  в двух вариантах:

- 1) Добавить нового – вариант, когда пользователь еще не заведен, первоначально будет предложено создать пользователя, а затем добавить в (назначить) роль.
- 2) Добавить существующего – выбор пользователя из списка

#### 2.4.2.7. Выгрузка в Excel списка ролей

Выгрузка в Excel производится при нажатии кнопки на правой панели . Вторым вариантом – нажатие кнопки , расположенной в верхней части заголовка таблицы со списком ролей.

Данный функционал предназначен для выгрузки в файл с расширением .xlsx. списка ролей, который был предварительно отфильтрован (т.е. в файл выгружается то, что отображается в списке ролей в момент нажатия кнопки Выгрузить в Excel).

В выгрузке присутствует только список самих ролей, без списка пользователей, которым назначена роль.

#### 2.4.3. Политики доступа

Раздел предназначен для работы с Политиками доступа. т.е. настройками прав пользователя на операции в системе и условия их применения.

Настройки политик доступа производятся в разрезе модулей.

По умолчанию отображается следующий список:

- Общие настройки
- Администрирование
- Дизайнер

В разделе доступны следующие функции:

- Просмотр списка политик доступа
- Массовое редактирование
- Добавление роли
- Копирование роли
- Копирование политики
- Добавление действия
- Просмотр информации действия
- Выгрузить в файл
- Загрузить из файла

##### 2.4.3.1. Просмотр списка политик доступа

После выбора модуля, на экране отображается список настроенных политик доступа.

Для отображения настроены следующие параметры:

- Класс объекта/Роль/Действие
- Приоритет
- Решение

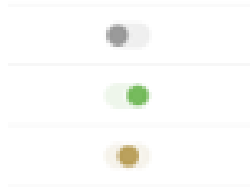
Класс объекта: на верхнем уровне дерева отображать классы объектов выбранного модуля.

Роль: при раскрытии узла верхнего уровня дерева отображаются политики для объекта.

Действие: при раскрытии узла второго уровня дерева отображаются действия доступные для роли.

Приоритет: отображается установленный приоритет действия.

Решение: используется трехпозиционный переключатель (Запрещено, Разрешено, Не определено)



#### 2.4.3.2. Массовое редактирование

Для массового редактирования политик доступа необходимо выделить один или более классов объектов, ИЛИ одну или более ролей, ИЛИ одно или более действий в роли, и нажать на «Массовое редактирование».

При подтверждении редактирования выбранное решение устанавливается для всех выбранных политик доступа.

#### 2.4.3.3. Добавление роли

Добавление роли происходит по кнопке «Добавить роль», если отмечен один класс объекта.

При нажатии на кнопку «Добавить роль» появляется модальное окно «Добавить роль» с выбором ролей из списка, чекбоксом «Добавить для всех объектов» и списком чекбоксов «Действия». После выбора параметров и нажатия кнопки «ОК» добавляется в отмеченные объекты политики доступа для выбранной роли с выбранными действиями.

Если не отмечен чекбокс «Добавить для всех объектов», добавляются политики для выбранной роли с выбранными действиями для одного выбранного объекта.

#### 2.4.3.4. Копирование роли

Копирование ролей происходит по кнопке «Копировать роль», если отмечена одна строка роли или одна строка объект. Доступен выбор целевого объекта и радиокнопки «Добавить новые и обновить существующие политики»/«Только добавить новые политики».

При подтверждении, если выбрано состояние «Только добавить новые политики», то в целевую роль добавляются недостающие действия с политиками из исходного объекта. Если выбрано состояние «Добавить новые и обновить существующие политики», то в целевой роли удаляются существующие политики и в целевую роль добавляются все действия с политиками из исходной роли.

#### 2.4.3.5. Копирование политики

Копирование политики происходит по кнопке «Копировать политики», если отмечена одна или более строк класса объекта.

При нажатии на кнопку «Копировать политики» появляется модальное окно «Копировать политики». Доступен выбор целевого объекта и радиокнопок «Добавить новые и обновить существующие политики»/«Только добавить новые политики».

При подтверждении, если выбрано состояние «Только добавить новые политики», то в целевой объект добавляются недостающие роли с политиками из исходного объекта. Если выбрано состояние «Добавить новые и обновить существующие политики», то в целевом объекте удаляются существующие политики и в целевой объект добавляются все роли с политиками из исходного объекта.

#### 2.4.3.6. Добавление действия

Добавление действий происходит по кнопке «Добавить действие», если отмечена одна или несколько ролей.

При нажатии на кнопку «Добавить действие» появляется модальное окно «Добавить действие» с чекбоксом «Добавить все действия» и списком чекбоксов «Действия» для объекта.

В поле «Выберите действие» доступен выбор:

- Утвердить
- Удалить
- Получить список
- Изменить
- Сохранить
- Просмотр

При подтверждении добавляется политика доступа для выбранного действия под выбранную роль с решением «не определено» (не отмечен чекбокс «Добавить все действия»). Если отмечен чекбокс «Добавить все действия», то добавить политики на все действия объекта под выбранную роль с решением «не определено».

#### 2.4.3.7. Просмотр информации действия

Для просмотра информации по политике необходимо двойным кликом нажать на строке с действием.

Открывается форма просмотра/редактирования политики доступа с полями:

- Создал: дата и время создания карточки политики доступа, а также пользователь, создавший политику
- Изменил: дата и время изменения карточки политики доступа, а также пользователь, изменивший политику
- ID: id карточки политики доступа

Доступные кнопки:

- Отменить – закрытие формы политики с несохранением изменений
- Сохранить – сохранение измененной политики доступа

Отображаются вкладки:

- Фильтр применимости
- Решение
- Обязательства

Вкладка «Фильтр применимости»:

Основные параметры

- Объект Класс (недоступно для редактирования)
- Действие (недоступно для редактирования)
- Роль (недоступно для редактирования)
- Приоритет (доступно для редактирования)

Вкладка «Решение»:

- Отображается параметр «Решение» с выбором значений «Запрещено» и «Разрешено»

Предикаты

- Доступно добавление предиката по кнопке

Вкладка «Обязательства»:

- Доступно добавление предиката по кнопке

#### 2.4.3.8. Выгрузить в файл

Данный функционал доступен по кнопке «Выгрузить в файл». Выгрузка производится в формате JSON отобранных политик (т.е. в файл выгружается то, что отображается в списке политик в момент нажатия кнопки).

#### 2.4.3.9. Загрузить из файла

Данный функционал доступен по кнопке «Загрузить из файла». Открывается окно проводника для выбора загружаемого файла. Возможна загрузка файла в формате JSON.

### 2.4.4. Орг. Структура

Раздел содержит просмотр дерева элементов орг. структуры с входящими в них пользователями, возможностями создания/редактирования/копирования/удаления узлов дерева элементов орг. структуры и привязки новых пользователей.

В разделе доступен следующий функционал:

- Просмотр дерева элементов орг. структуры
- Поиск элемента
- Просмотр элемента орг. структуры
- Просмотр пользователей, привязанных к элементу
- Создание корневого элемента
- Создание дочернего элемента
- Редактирование элемента
- Привязка пользователей
- Удаление привязки пользователя к элементу орг. структуры

- Копирование элемента
- Удаление элемента
- Создание/редактирование записи в календаре
- Удаление записи в календаре

#### 2.4.4.1. Просмотр дерева элементов орг. структуры

При переходе в раздел Орг. структура в левой части экранной формы отображается дерево элементов орг. структуры, состоящее из:

- 1) Области фильтрации по наименованию узла
- 2) Панели с управляющими кнопками
- 3) Фильтра по статусу записи
- 4) Списка элементов орг. структуры

#### 2.4.4.2. Поиск элемента

При задании значения в фильтре и выполнении  будет выполнена фильтрация по вхождению и показаны только те элементы (узлы любого уровня), которые соответствуют заданному фильтру.

Для отбора элементов по статусу записи необходимо установить значение в поле radio button в одно из положений:

- Все
- Неудаленные

По умолчанию radio button установлен в позицию «Неудаленные» (отображаются узлы дерева орг. структуры, для которых признак Удален = нет).

Позиция «Все» - отображаются все узлы дерева орг. структуры, независимо от значения признака Удален.

#### 2.4.4.3. Просмотр элемента орг. структуры

Функционал просмотра элемента орг. структуры доступен при выделении кнопкой мыши определенного элемента в дереве.

Возможно увидеть следующую информацию по узлу:

- Регион присутствия
- Бизнес тип узла
- Часовой пояс
- Активна
- Удалена
- Комментарий
- Список пользователей, привязанных к узлу
- Настройки календаря

#### 2.4.4.4. Просмотр пользователей, привязанных к элементу

Функционал просмотра списка пользователей, привязанных к элементу орг. структуры, доступен из карточки узла, на вкладке «Пользователи».

Список пользователей выводится с колонками:

- Логин (имя пользователя)
- Фамилия
- Имя
- Отчество
- Должность
- Телефон
- Email
- Признак Удален

Возможен поиск по списку пользователей, для этого необходимо нажать на иконку воронки , затем задать необходимые фильтры.

#### 2.4.4.5. Создание корневого узла

Создание корневого узла доступно по кнопке  .

Для заполнения открывается форма, содержащая атрибуты/разделы:

- Регион присутствия
- Бизнес тип узла
- Часовой пояс
- Активна
- Удалена
- Комментарий
- Список пользователей, привязанных к узлу
- Настройки календаря

На форма присутствуют кнопки:

- Сохранить
- Отмена

#### 2.4.4.6. Создание дочернего узла

#### 2.4.4.7. Редактирование элемента

Редактирование элемента доступно с карточки элемента орг. структуры. Для перехода к редактированию необходимо открыть карточку двойным кликом кнопки мыши, выбрав предварительно нужный в дереве элементов и нажать кнопку «Редактировать».

#### 2.4.4.8. Привязка пользователей

Функционал привязки пользователей к элементу орг. структуры, доступен из карточки узла, на вкладке «Пользователи».

Список пользователей выводится с колонками:

- Логин (имя пользователя)
- Фамилия
- Имя

- Отчество
- Должность
- Телефон
- Email
- Признак Удален

Возможен поиск по списку пользователей, для этого необходимо нажать на иконку воронки , затем задать необходимые фильтры.

Добавление пользователя доступно при нажатии кнопки  в двух вариантах:

- 1) Добавить нового – вариант, когда пользователь еще не заведен, первоначально будет предложено создать пользователя, а затем привязать его к элементу орг. структуры.
- 2) Добавить существующего – выбор пользователя из списка  
К отбору фильтруются только пользователи, для которых признак «Удален» = нет. Возможно выделить сразу нескольких пользователей в списке и привязать их к элементу.

#### 2.4.4.9. Удаление привязки пользователя к элементу орг. структуры

Функционал удаления привязки пользователя от элемента орг. структуры доступен при выделении пользователя в списке (вкладка «Пользователи» карточки элемента) и нажатии кнопки . После подтверждения операции привязка будет удалена.

#### 2.4.4.10. Копирование элемента

Функционал доступен при выделении элемента орг. структуры в дереве и нажатии кнопки . При нажатии кнопки открывается карточка создания элемента, куда наследуется только Наименование. Целевой узел будет создан на том же уровне и с той же подчиненностью, что у прототипа.

#### 2.4.4.11. Удаление элемента

Функционал доступен при выделении элемента орг. структуры в дереве и нажатии кнопки . После подтверждения операции выбранному узлу, а также его подчиненным узлам будет присвоен признак Удален = да, а также будут удалены привязки пользователей к данным узлам.

#### 2.4.4.12. Создание/редактирование записи в календаре

Вкладка «Календарь» карточки элемента орг. структуры содержит 2 табличные формы:

- Рабочая неделя
- Исключения

Данные выводятся с колонками:

- Временной период
- Название



Для создания записи необходимо нажать кнопку . После внесения данных и проверок на пересечение периодов новая запись отображается в табличной форме вкладки Календарь.

Для редактирования строки необходимо выделить ее в списке и открыть форму двойным кликом мыши.

#### 2.4.4.13. Удаление записи в календаре

Для редактирования строки необходимо выделить ее в списке и нажать кнопку



### 2.4.5. Бизнес-процессы

Раздел предназначен для мониторинга и управления бизнес-процессами ИТА ВРМ.

Раздел обеспечивает выполнение следующих функций:

- Просмотр списка исполняемых процессов
- Просмотр списка версий исполняемых процессов
- Выгрузка в Excel списка загруженных на сервер ИТА ВРМ исполняемых процессов
- Просмотр схемы процесса с отображением количества активных задач на схеме и списком запущенных экземпляров
- Просмотр параметров экземпляра процесса
- Выгрузка в Excel списка запущенных экземпляров процесса
- Запуск процесса
- Удаление процесса с сервера ИТА-ВРМ

#### 2.4.5.1. Просмотр списка исполняемых процессов

При входе в раздел Бизнес-процессы отображаются загруженные на сервер ИТА-ВРМ исполняемые процессы.

По умолчанию выводятся следующие колонки:

- Ключ
- Название
- Количество экземпляров
- Количество инцидентов

Созданные схемы процессов выводятся со свернутым списком своих версий.

#### 2.4.5.1. Выгрузка в Excel списка исполняемых процессов

Выгрузка в Excel производится при нажатии кнопки на правой панели .

Данный функционал предназначен для выгрузки в файл с расширением .xlsx. списка процессов, который был предварительно отфильтрован (т.е. в файл выгружается то, что отображается в списке процессов в момент нажатия кнопки Выгрузить в Excel).

#### 2.4.5.2. Просмотр списка версий процессов

При входе в раздел Бизнес-процессы отображаются загруженные на сервер ITA-BPM исполняемые процессов.

Для просмотра версий необходимо развернуть список, нажав стрелку  на строке процесса. При этом отобразятся все версии процесса.

По умолчанию выводятся следующие колонки:

- Версия схемы
- ID
- Количество экземпляров
- Количество инцидентов

Возможен просмотр дополнительной информации по версии – схема процесса, список экземпляров (активных, завершенных).

Для просмотра необходимо спозиционироваться на нужной версии и:

- сделать двойной клик кнопки мыши
- нажать кнопку  на правой панели

При попытке перейти к информации по версии, где нет экземпляров – об этом пользователю будет выведено информационное сообщение.

#### 2.4.5.3. Просмотр схемы версии процесса с отображением количества активных задач на схеме и списком запущенных экземпляров

При переходе к просмотру данных версии процесса доступен функционал:

- Отображение схемы выбранной версии процесса
- На схеме отображаются активные задачи экземпляров процесса с указанием количества на каждой
- Список экземпляров процесса

Список экземпляров выводится с отображением колонок:

- Время старта
- Состояние
- ID
- ID определение процесса

Возможные состояния экземпляров процессов:

- ACTIVE - активен
- COMPLETED - завершен
- SUSPENDED - приостановлен

При выделении на схеме определенной задачи (любого типа), где есть активные токены, в списке экземпляров остаются только те, которые остановились на выделенной задаче.

#### 2.4.5.4. Просмотр параметров экземпляра процесса

Для просмотра необходимо, находясь на странице версии процесса, спозиционироваться на нужной и:

- сделать двойной клик кнопки мыши
- нажать кнопку  на правой панели

Доступен следующий функционал по работе с экземпляром процесса:

- отображение таймлайна со статусом задачи и временем его изменения
- просмотр схемы процесса с отображением активных токенов
- отображение основной информации экземпляра процесса
- отображение списка инцидентов экземпляра процесса
- отображение переменных экземпляра процесса
- поиск и фильтрация списка переменных процесса
- редактирование значения переменной процесса
- добавление переменной процесса
- удаление переменной процесса
- отображение списка подпроцессов
- отображение списка вызываемых процессов
- отображение списка задач процесса
- возможность скачать схему процесса как файл BPMN
- возможность скачать схему процесса как файл SVG
- вывод на экран списка «горячих» клавиш для действий при моделировании процесса
- смена цветового режима редактора
- переход в полноэкранный режим
- приближение/удаление изображения схемы процесса
- вписать схему в окно
- приостановка исполнения экземпляра процесса
- запуск ранее приостановленного экземпляра процесса
- удаление запущенного экземпляра процесса

##### 2.4.5.4.1. Просмотр схемы процесса с отображением активных токенов

При переходе к просмотру данных экземпляра версии процесса доступен функционал:

- Отображение схемы выбранного экземпляра процесса
- На схеме отображаются активные задачи экземпляра процесса с указанием количества на каждой
- Отображение основной информации экземпляра процесса

##### 2.4.5.4.2. Отображение основной информации экземпляра процесса

На странице информации об экземпляре версии процесса отображаются следующие технические данные процесса:

- definitionId (идентификатор определения процесса)
- businessKey (бизнес-ключ)

- name (название)
- description (описание)
- version (версия)
- Id (идентификатор экземпляра)
- deploymentId (идентификатор загрузки схемы процесса на сервисе ITA-BPM)
- versionTag (тег версии)
- category (категория)
- ended (признак завершения)
- suspended (признак приостановки)

#### 2.4.5.4.3. Отображение списка инцидентов экземпляра процесса

На странице информации об экземпляре версии процесса отображаются следующие технические данные по инцидентам:

- Сообщение об ошибке
- Дата/время возникновения инцидента
- Наименование задачи, на которой возник инцидент
- Ссылка на экземпляр - причину инцидента
- Ссылка на головной экземпляр - причину инцидента
- Тип инцидента
- Действие

Возможные типы инцидентов:

- **failedJob**. Создаётся, когда автоматические попытки выполнения задачи (по таймеру или асинхронному продолжению) исчерпаны. Инцидент указывает, что выполнение застряло и не будет продолжаться автоматически, необходимо административное действие. Разрешается, когда задача выполняется вручную или когда для неё устанавливается значение повторных попыток  $> 0$ .
- **failedExternalTask**. Создаётся, когда исполнитель внешней задачи сообщает о сбое, а количество повторных попыток установлено на значение  $\leq 0$ . Инцидент указывает, что внешняя задача застряла и не может быть выполнена. Необходимо административное действие, чтобы сбросить количество повторных попыток.

Доступное действие для инцидента – повтор выполнения.

#### 2.4.5.4.4. Отображение переменных экземпляра процесса

На странице информации об экземпляре версии процесса отображаются следующие данные по переменным процесса:

- Название переменной (код)
- Тип переменной
- Значение
- Доп. информация

#### 2.4.5.4.5. Поиск и фильтрация списка переменных процесса

Для поиска/отбора переменных процесса необходимо нажать на значок воронки 

При нажатии на воронку будет выведено меню для выбора параметра для поиска и значения. При необходимости отбора по нескольким параметрам нужно нажать на текст/кнопку «Добавить условие» и повторить шаги.

#### 2.4.5.4.6. Редактирование значения переменной процесса

Функционал редактирования переменной доступен при:

- Выделении в списке переменной и нажатии кнопки 
- Выделении в списке переменной и двойном клике кнопкой мыши

Для редактирования пользователю предлагается форма, на которой расположены поля:

- Название переменной (код)
- Тип переменной
- Значение

Также на форме расположены кнопки:

- Сохранение
- Отмена

#### 2.4.5.4.7. Добавление переменной процесса

Для добавления переменной процесса необходимо нажать кнопку 

Для ввода данных пользователю предлагается форма, на которой расположены поля:

- Название переменной (код)
- Тип переменной
- Значение

Также на форме расположены кнопки:

- Сохранение
- Отмена

#### 2.4.5.4.8. Удаление переменной процесса

Для удаления переменной процесса необходимо выделить нужную в списке и нажать кнопку . После подтверждения пользователем операции переменная будет удалена.

#### 2.4.5.4.9. Отображение списка вызываемых процессов

На странице информации об экземпляре версии процесса отображаются следующие данные по вызываемым процессам:

- Статус вызова процесса
- Идентификатор экземпляра
- Описание процесса
- Название задачи

#### 2.4.5.4.10. Отображение списка задач процесса

На странице информации об экземпляре версии процесса отображаются следующие данные по активным задачам процесса (те, на которых стоят токены):

- Ссылка на задачу (название задачи)
- На кого назначена задача
- Дата создания
- Дата выполнения
- Дата проверки
- Приоритет
- Статус делегирования
- Идентификатор задачи
- Действие (добавить группу, добавить исполнителя)

#### 2.4.5.4.11. Возможность скачать схему процесса как файл BPMN

Для сохранения схемы процесса в формате BPMN необходимо нажать кнопку  в нижней части экрана. Файл будет помещен в папку Загрузки/будет открыто окно для выбора папки сохранения – в зависимости от настроек браузера.

#### 2.4.5.4.12. Возможность скачать схему процесса как файл SVG

Для сохранения схемы процесса как файл SVG (в виде картинки) необходимо нажать кнопку  в нижней части экрана. Файл будет помещен в папку Загрузки/будет открыто окно для выбора папки сохранения – в зависимости от настроек браузера.

#### 2.4.5.4.13. Вывод на экран списка «горячих» клавиш для действий при моделировании процесса

Функционал позволяет увидеть список «горячих» клавиш для действий отображения схемы процесса. Для вывода на экран информации необходимо нажать кнопку  в правой части экрана со схемой процесса.

#### 2.4.5.4.14. Смена цветового режима редактора

Функционал позволяет изменить цветовой режим редактора. Для изменения режима предназначена кнопка  в правой части экрана отображения схемы процесса.

#### 2.4.5.4.15. Переход в полноэкранный режим

Функционал позволяет отображать экран отображения схемы процесса на весь экран. Для перехода в полноэкранный режим предназначена кнопка  в правой части экрана. Для возврата в обычный режим необходимо повторно нажать на эту же кнопку либо кнопку ESC на клавиатуре.

#### 2.4.5.4.16. Приближение/удаление схемы процесса

Функционал позволяет изменять масштаб отображения схемы процесса. Управляющие кнопки   расположены в правой нижней части экрана.

#### 2.4.5.4.17. Вписать схему в окно

Функционал позволяет помещать схему процесса полностью в экран. Управляющая кнопка расположена  в правой нижней части экрана.

#### 2.4.5.4.18. Приостановка исполнения экземпляра процесса

Для приостановки исполнения экземпляра процесса необходимо нажать кнопку  в левой части экрана с отображением схемы процесса. После подтверждения пользователем выполнения операции исполнение процесса будет осуществлено (на текущей выполняемой задаче).

#### 2.4.5.4.19. Запуск ранее приостановленного экземпляра процесса

Для запуска ранее приостановленного экземпляра процесса необходимо нажать кнопку  в левой части экрана с отображением схемы процесса. После подтверждения пользователем выполнения операции исполнение процесса будет продолжено.

#### 2.4.5.4.20. Удаление запущенного экземпляра процесса

Для удаления экземпляра процесса необходимо нажать кнопку  в левой части экрана с отображением схемы процесса. После подтверждения пользователем выполнения операции экземпляр будет удален.

#### 2.4.5.5. Выгрузка в Excel списка экземпляров процесса

Выгрузка в Excel производится при нажатии кнопки на правой панели .

Данный функционал предназначен для выгрузки в файл с расширением .xlsx. списка экземпляров процессов, который был предварительно отфильтрован (т.е. в файл выгружается то, что отображается в списке процессов в момент нажатия кнопки Выгрузить в Excel).

#### 2.4.5.6. Запуск процесса

Данный функционал доступен:

- При выделении головной строки процесса и нажатии кнопки . При этом запускается последняя версия процесса, загруженная на сервер ИТА-BPM
- При выделении строки последней версии процесса и нажатии кнопки .

#### 2.4.5.7. Удаление процесса с сервера ИТА-BPM

Из списка процессов, которые загружены на сервер ИТА-BPM, доступен функционал удаления процесса со всеми его версиями и экземплярами. Для удаления необходимо выделить в списке процесс и нажать кнопку . После подтверждения пользователем выполнения операции процесс будет удален.

#### 2.4.6. Список задач

В разделе отображается активных пользовательских список задач (User Task) всех экземпляров процессов, загруженных на сервер ИТА-BPM.

По умолчанию список выводится с колонками:

- Дата выполнения задачи

- Дата создания задачи
- Исполнитель
- Исполнитель, на которого назначена задача
- Ключ определения бизнес-процесса
- Роль, на которую назначена задача
- Статус задачи
- Id взаимодействия
- Id задачи
- Id формы
- Id экземпляра бизнес-процесса

В разделе Список задач доступен следующий функционал:

- Просмотр списка активных пользовательских задач
- Поиск/отбор задач по параметрам
- Настойка колонок списка задач
- Просмотр информации по задаче/экземпляру процесса задачи
- Изменение параметров задачи
- Выгрузка в Excel списка задач

#### 2.4.6.1. Просмотр списка активных пользовательских задач (User task)

При входе в раздел Список задач для просмотра всех пользовательских задач необходимо указать необходимые значения фильтрации и нажать кнопку Отобразить (либо сразу нажать кнопку без использования фильтров).

#### 2.4.6.2. Поиск/отбор задач по параметрам

Для поиска/отбора задач необходимо воспользоваться панелью фильтрации. Установить значение в предустановленных фильтрах либо добавить дополнительный фильтр из списка доступных.

Также возможно использовать фильтр по нажатию кнопки 

При нажатии на воронку будет выведено меню для выбора параметра для поиска и значения. При необходимости отбора по нескольким параметрам нужно нажать на текст/кнопку «Добавить условие» и повторить шаги.

#### 2.4.6.3. Настойка колонок списка задач

По умолчанию список выводится с колонками:

- Дата выполнения задачи
- Дата создания задачи
- Исполнитель
- Исполнитель, на которого назначена задача
- Ключ определения бизнес-процесса
- Роль, на которую назначена задача
- Статус задачи
- Id взаимодействия
- Id задачи
- Id формы

- Id экземпляра бизнес-процесса

Пользователь может выбрать только те колонки для отображения, что ему нужны.

Чтобы не выводить ту или иную колонку, необходимо нажать на иконку , где возможно снять выделение кнопкой мыши.

Дополнительно присутствуют настройки:

- Сохранять выбранную сортировку
- Сохранять выбранные колонки
- Сохранять фильтры
- Сохранять количество строк для отображения

#### 2.4.6.4. Просмотр информации по задаче/экземпляру процесса задачи

Функционал просмотра информации по задаче/экземпляру процесса доступен при выделении строки нужной задачи в списке и нажатии кнопки  на правой панели либо при выполнении двойного клика кнопкой мыши на строке нужной задачи. При этом осуществляется переход на страницу экземпляра процесса (см. подробнее п.4.4.5.4.1).

#### 2.4.6.5. Изменение параметров задачи

Функционал просмотра информации по задаче/экземпляру процесса доступен при выделении строки нужной задачи в списке и нажатии кнопки  на правой панели. При этом пользователю отображаются следующие данные:

- Id задачи
- Текущий пользователь
- Доступно для пользователя
- Доступно для роли
- Плановая дата исполнения
- Для редактирования пользователю доступны все поля, кроме Id задачи.

#### 2.4.6.6. Выгрузка в Excel списка задач

Выгрузка в Excel производится при нажатии кнопки на правой панели .

Данный функционал предназначен для выгрузки в файл с расширением .xlsx. списка задач, который был предварительно отфильтрован (т.е. в файл выгружается то, что отображается в списке задач в момент нажатия кнопки Выгрузить в Excel).

#### 2.4.7. Список сервисных задач (Service Task)

В разделе отображаются активных сервисных задач (Service Task) всех экземпляров процессов, загруженных на сервер ИТА-ВРМ. В данном случае под активными понимаются такие, где вызов операций/сервисов был выполнен, но ответ получен не был.

По умолчанию список выводится с колонками:

- Идентификатор пространства описания процессов
- Ключ определения бизнес-процесса
- Наименование процесса

- Поставлено на паузу
- Приоритет
- Activity id (идентификатор задачи в процессе)
- Id задачи
- Id определения бизнес-процесса
- Id воркера
- Id экземпляра процесса

В разделе Список сервисных задач доступен следующий функционал:

- Просмотр списка активных сервисных задач
- Поиск/отбор задач по параметрам
- Настойка колонок списка задач
- Просмотр информации по задаче/экземпляру процесса задачи
- Выгрузка в Excel списка задач

#### 2.4.7.1. Просмотр списка активных сервисных задач (Service task)

При входе в раздел Список сервисных задач для просмотра всех задач необходимо указать необходимые значения фильтрации и нажать кнопку Отобразить (либо сразу нажать кнопку без использования фильтров).

#### 2.4.7.2. Поиск/отбор задач по параметрам

Для поиска/отбора задач необходимо воспользоваться панелью фильтрации. Установить значение в предустановленных фильтрах либо добавить дополнительный фильтр из списка доступных.

Также возможно использовать фильтр по нажатию кнопки 

При нажатии на воронку будет выведено меню для выбора параметра для поиска и значения. При необходимости отбора по нескольким параметрам нужно нажать на текст/кнопку «Добавить условие» и повторить шаги.

#### 2.4.7.3. Настойка колонок списка задач

По умолчанию список выводится с колонками:

- Идентификатор пространства описания процессов
- Ключ определения бизнес-процесса
- Наименование процесса
- Поставлено на паузу
- Приоритет
- Activity id (идентификатор задачи в процессе)
- Id задачи
- Id определения бизнес-процесса
- Id воркера
- Id экземпляра процесса

Пользователь может выбрать только те колонки для отображения, что ему нужны.

Чтобы не выводить ту или иную колонку, необходимо нажать на иконку , где возможно снять выделение кнопкой мыши.

Дополнительно присутствуют настройки:

- Сохранять выбранную сортировку
- Сохранять выбранные колонки
- Сохранять фильтры
- Сохранять количество строк для отображения

#### 2.4.7.4. Просмотр информации по задаче/экземпляру процесса задачи

Функционал просмотра информации по задаче/экземпляру процесса доступен при выделении строки нужной задачи в списке и нажатии кнопки  на правой панели либо при выполнении двойного клика кнопкой мыши на строке нужной задачи. При этом осуществляется переход на страницу экземпляра процесса (см. подробнее п.4.4.5.4.1).

#### 2.4.7.1. Выгрузка в Excel списка задач

Выгрузка в Excel производится при нажатии кнопки на правой панели .

Данный функционал предназначен для выгрузки в файл с расширением .xlsx. списка задач, который был предварительно отфильтрован (т.е. в файл выгружается то, что отображается в списке задач в момент нажатия кнопки Выгрузить в Excel).

## 2.5.Uiback

Сервис uiback предназначен для обеспечения взаимодействия пользовательского интерфейса и ITA-BPM, в том числе:

- обеспечивает взаимодействие user-ui и ITA-BPM
- сохраняет информацию о попытках запуска процесса
- передает команду от user-ui на старт процесса
- передает сообщения от user-ui в ITA-BPM
- предоставляет информацию об успехе/неуспехе старта процесса для user-ui
- передает команду от user-ui на завершение задачи

## 2.6.Tasklist

Сервис tasklist предназначен для:

- предоставления списка задач
- изменения параметров задачи

#### 2.6.1. Предоставление списка задач

Список задач может предоставляться исполнителю по задаче, в таком случае исполнитель может получить задачи:

- находящиеся у него на исполнении (статус - в работе, assignee - исполнитель)
- назначенные на исполнителя (статус - новая, assignee - null, candidate user - исполнитель)
- назначенные на роль (статус - новая, assignee - null, candidate role - роль исполнителя)

Список задач может предоставляться руководителю, в таком случае руководитель должен получить задачи:

- находящиеся у него на исполнении в подразделении (статус - в работе, assignee - исполнитель из подразделения руководителя)
- назначенные на исполнителя в подразделении (статус - новая, assignee - null, candidate user - исполнитель из подразделения руководителя)
- назначенные на роль (статус - новая, assignee - null, candidate role - роль исполнителя из подразделения руководителя )

На вход сервису tasklist поступает фильтр для отбора задач. Tasklist обращается к сервису политик доступа и получает разрешение на отображение списка и предикаты, какие задачи доступны

#### 2.6.2. Изменение параметров задачи

Исполнитель может:

- Взять задачу в работу (assignee = исполнитель)
- Отказаться от выполнения задачи (assignee = null)

Руководитель может:

- Назначить задачу на исполнителя или группу (candidate user, candidate role)
- Поменять исполнителя по задаче, которая уже в работе (assignee)
- Отменить задачу (прекратить выполнение процесса, изменить статус задачи)
- Изменить срок выполнения задачи (due date)

Для каждой задачи сервис tasklist должен вызвать сервис политик доступа и получить разрешенные действия с задачей.

С целью удовлетворения возможных требований по нагрузке, сервис должен уметь горизонтально масштабироваться. Масштабирование потребуется, если в не успевают вычитываться топики событий-завершения работы пользователем над задачей или не обрабатываются все REST-запросы от интерфейса.

В сервисе есть внутренний кэш (сохраняет результат со списком задач, полученным по одному searchId), аутентификация в keycloak не выполняется.

### 2.7.Набор сервисов для асинхронного взаимодействия с процессом (воркеры)

В ITA-BPM реализованы следующие сервисы для асинхронного взаимодействия с процессом (воркеры):

- worker-start: обработка событий запуска процессов
- worker-usertask: поиск и блокировка новых задач пользователя
- worker-usertask-complete: завершение задач пользователя
- worker-message: обработка сообщений, отправляемых в процессы
- worker-ext-task: поиск и отправка сервисных задач
- worker-ext-task-complete: завершение сервисных задач

### 2.7.1. worker-start

Сервис worker-start предназначен для обработки событий-требований запуска бизнес-процессов в ИТА-BPM. Сервис обрабатывает события, вызывает запуск бизнес-процесса в ИТА-BPM, генерирует событие успешного старта или событие ошибки при старте бизнес-процесса. При невозможности обработать требование запуска бизнес-процесса сервис повторяет сообщение в топик DEAD LETTER QUEUE для возможности повторной обработки. В случае, если ИТА-BPM недоступен и фактический старт процесс выполнить нельзя, сервис останавливает чтение из kafka до тех пор, пока ИТА-BPM не станет доступен.

### 2.7.2. worker-usertask

Сервис worker-usertask предназначен для поиска новых задач пользователя в ИТА-BPM и генерации событий, информирующих о новом задании. Сервис запрашивает по расписанию задачи в ИТА-BPM, у которых не задан конкретный исполнитель (assignee) на уровне схемы процесса, устанавливает им assignee = worker, параметры задачи в kafka. Запрос происходит пачками по N штук. При невозможности обработать задачу сервис устанавливает assignee обратно в null.

### 2.7.3. worker-usertask-complete

Сервис worker-usertask-complete предназначен для обработки событий-завершения работы пользователем над задачей. Сервис обрабатывает события, вызывает завершение задачи в ИТА-BPM. При невозможности обработать завершение задачи сервис повторяет сообщение в топик DEAD LETTER QUEUE для возможности повторной обработки.

### 2.7.4. worker-message

Сервис worker-message предназначен для обработки событий-передачи сообщений в процесс ИТА-BPM. Сервис обрабатывает события, вызывает отправку сообщения в ИТА-BPM. При невозможности обработать отправку сообщения сервис повторяет сообщение в топик DEAD LETTER QUEUE для возможности повторной обработки.

### 2.7.5. worker-ext-task

Сервис worker-ext-task предназначен для поиска новых сервисных задач в ИТА-BPM и генерации событий, информирующих о новом задании. Сервис запрашивает по расписанию сервисные задачи (External task) в ИТА-BPM, у которых не установлен lock на уровне схемы процесса и отправляет задачи в кафку. При этом в bpm устанавливается lock у этих задач. При невозможности обработать задачу сервис выполняет unlock.

### 2.7.6. worker-ext-task

Сервис worker-ext-task-complete предназначен для обработки событий-завершения работы пользователем над задачей. Сервис обрабатывает события, вызывает завершение сервисной задачи в ИТА-BPM. При невозможности обработать завершение задачи сервис повторяет сообщение в топик DEAD LETTER QUEUE для возможности повторной обработки.

## 2.8. Gateway

Единая точка входа пользователей в систему, обеспечивает проверку аутентификации, перенаправление пользователей на страницу аутентификации и маршрутизацию запросов пользователей внутри системы.

## 2.9. Движок ITA-BPM версии 7

Поддерживаются следующие возможности:

- Исполнение процессов BPMN 2.0 на ядре Camunda 7.15+
- Предоставление REST API для управления процессами
- Хранение данных процессов и истории выполнения

## 2.10. Движок ITA-BPM версии 8

Поддерживаются следующие возможности:

- Исполнение процессов BPMN 2.0 на ядре Camunda 8
- Предоставление REST API для управления процессами
- Хранение данных процессов и истории выполнения

