|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | УТВЕРЖДАЮ: | УТВЕРЖДАЮ: | | Государственное бюджетное учреждение | ООО «Систематика» | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ И.О. Фамилия | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_А.С. Куминов | | МП | МП | | «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2018 г. | «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2018 г. | |  |  | |  |  | |  |  | |  |  | |  |  | |  |
|  |  |
|  |  |
|  |  |
|  |  |

АВТОМАТИЗИРОВАННАЯ ИНФОРМАЦИОННАЯ СИСТЕМА

«ЕДИНАЯ ИНФОРМАЦИОННО-ТЕХНОЛОГИЧЕСКАЯ ПЛАТФОРМА ЭЛЕКТРОННОГО ПРАВИТЕЛЬСТВА»

РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

ПОДСИСТЕМЫ «КОНСТРУИРОВАНИЯ ИНФОРМАЦИОННЫХ СИСТЕМ»

Хабаровск

2018

**СОДЕРЖАНИЕ**

[Введение 5](#_Toc527802871)

[1 Установка подсистемы конструирования 6](#_Toc527802872)

[2 Создание Приложения и связанных сущностей 7](#_Toc527802873)

[2.1 Создание приложения 7](#_Toc527802874)

[2.2 Создание класса модели 8](#_Toc527802875)

[2.3 Удаление модели 9](#_Toc527802876)

[2.4 Создание атрибутов сущности 10](#_Toc527802877)

[2.5 Схема связи таблиц данных 15](#_Toc527802878)

[3 Динамические формы 17](#_Toc527802879)

[3.1 Задание условий для отображения полей 17](#_Toc527802880)

[3.2 Способы группировки атрибутов на форме 21](#_Toc527802881)

[3.3 Клонирование элементов 24](#_Toc527802882)

[4 Специфичные настройки отображения атрибутов 27](#_Toc527802883)

[4.1 Настройка маски в атрибутах 27](#_Toc527802884)

[4.2 Указание на Обязательность атрибута 27](#_Toc527802885)

[4.3 Условия отбора допустимых значений 27](#_Toc527802886)

[4.4 Сортировка выборки допустимых значений 27](#_Toc527802887)

[4.5 Список выбора допустимых значений 28](#_Toc527802888)

[5. Бизнес – процессы 29](#_Toc527802889)

[6. Подключение режима офлайн 34](#_Toc527802890)

[7. Экспорт/импорт приложения 35](#_Toc527802891)

**ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ**

|  |  |  |
| --- | --- | --- |
| №п/п | Сокращение /Термин | Значение |
|  | АИС | Автоматизированная информационная система |
|  | АРМ | Автоматизированное рабочее место |
|  | БД | База данных |
|  | ВИС | Ведомственная информационная система |
|  | ГИС | Государственная информационная система |
|  | ЕИТП, Система | Единая информационно-технологическая платформа электронного правительства Республики Саха (Якутия) |
|  | ИС | Информационная система |
|  | Модуль | Набор взаимосвязанных классов и отдельных объектов, подсистем, зависимостей, операций, событий и ограничений. В модуль может входить несколько подсистем |
|  | НД | Несанкционированный доступ |
|  | ОГВ | Органы государственной власти Республики Саха (Якутия) |
|  | ОМСУ | Органы местного самоуправления Республики Саха (Якутия) |
|  | ОС | Операционная система |
|  | ПМИ | Программа и методика испытаний |
|  | ПО | Программное обеспечение |
|  | РС(Я) | Республика Саха (Якутия) |
|  | РОИВ | Исполнительные органы государственной власти Республики Саха (Якутия) |
|  | РПГМУ | Портал государственных и муниципальных услуг Республики Саха (Якутия) |
|  | СМЭВ | Система межведомственного электронного взаимодействия |
|  | СУБД | Система управления базами данных |
|  | ТЗ | Техническое задание |
|  | ЦОД | Центра обработки данных электронного правительства Республики Саха (Якутия) |
|  | ЭП | Электронная подпись |
|  | ЭП-ОВ | Электронная подпись органа власти |
|  | ЭП-СП | Сертификат электронной подписи |
|  | LDAP | Протокол прикладного уровня для доступа к службе каталогов X.500, разработанный IETF как облегчённый вариант разработанного ITU-T протокола DAP |
|  | SOAP | Simple Object Access Protocol – простой протокол доступа к объектам Протокол обмена структурированными сообщениями в распределённой информационной среде |
|  | REST API | Архитектура программного обеспечения для построения распределенных масштабируемых веб-сервисов |
|  | UML | Unified Modeling Language — унифицированный язык моделирования) Язык графического описания для объектного моделирования в области разработки программного обеспечения, моделирования бизнес-процессов, системного проектирования и отображения организационных структур |
|  | UI | Пользовательский интерфейс |
|  | Web-браузер | Программное обеспечение для просмотра веб-сайтов, то есть для запроса веб-страниц, их обработки, вывода и перехода от одной страницы к другой |
|  | WSDL | (англ. Web Services Description Language) Язык описания веб-сервисов и доступа к ним, основанный на языке XML. |
|  | XML | (англ. eXtensible Markup Language) Расширяемый язык разметки. |

# Введение

Общие сведения

Данный документ представляет собой руководство пользователя по работе с автоматизированной информационной системой «Единая информационно-технологическая платформа электронного правительства» Подсистема конструирования информационных систем.

Руководство пользователя описывает последовательность и состав операций, которые выполняет оператор по работе в системе.

**Уровень подготовки пользователя**

До работы с Системой допускаются пользователи, обладающие базовыми навыками работы на компьютере и сети Интернет.

# Установка подсистемы конструирования

Для автономной работы (офлайн режим) необходимо:

1. Скачать архив «studio.zip».
2. Распаковать архив.
3. Перейти в папку «studio» и запустить командную строку.
4. Ввести команду «npm install» - она поставит ключевые зависимости, в том числе локально сборщик gulp
5. После успешного завершения предыдущего шага ввести команду сборки приложения «gulp assemble».
6. Далее запускаем Подсистему конструирования информационных систем командой «node .\www».
7. Перейти по адресу http://localhost:8888, в браузере отобразится страница Подсистемы конструирования информационных систем (см. рисунок 1)

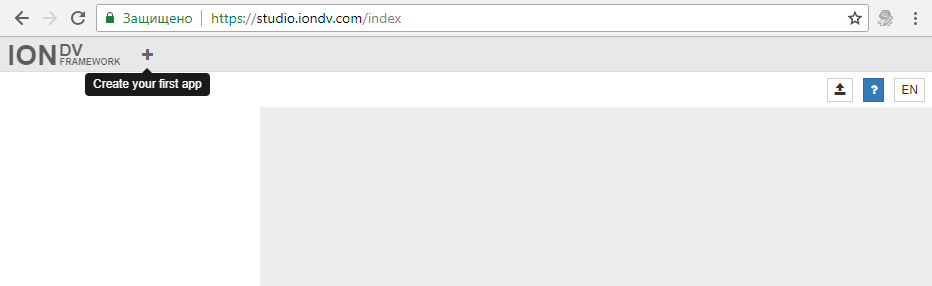


Рисунок 1 – открытие страницы Подсистемы

Для работы непосредственно в системе, необходимо перейти на адрес подсистемы конструктора информационных систем, представленный администратором системы.

# Создание Приложения и связанных сущностей

## Создание приложения

На главной странице (см. рисунок 2) в правом верхнем углу расположена кнопка переключения между языками (RU, EN), на рабочей панели доступны действия:

 - создать приложение

 - загрузить приложение из сохраненных

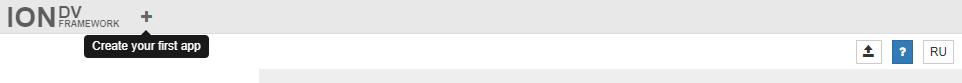


Рисунок 2 – главная страница

При выборе действия «Создать приложение», открывается форма создания приложения с полем атрибута для ввода наименования (системного имени) приложения, описания и версии (см. рисунок 3).

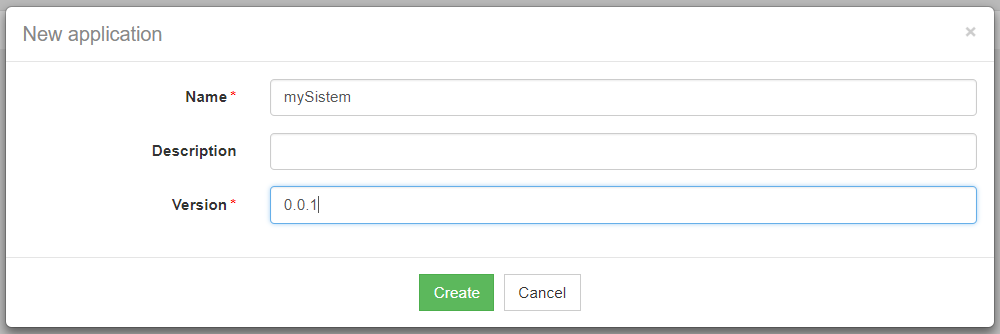


Рисунок 3 – создание приложения

Таблица 1 – описание полей для формы создания приложения

|  |  |
| --- | --- |
| **Наименование поля** | **Описание** |
| Название | Системное имя приложения. Состоит из символов латинского алфавита, без пробелов. Допускается применение символа “–“. |
| Описание | Краткое описание приложения. |
| Версия | Строка с маской xx.xx.xx. Используется для обновления версии приложения. |

Сохраняем форму кнопкой «Создать» и видим отображение приложения и его компонентов в виде иерархии в навигации, а также на рабочей панели появилась действия для редактирования и удаления приложения.

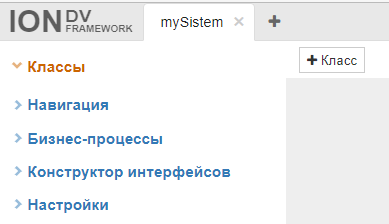


Рисунок 4 – рабочая панель Приложения

## Создание класса модели

Для создания класса модели данных нужно перейти в пункт меню «Классы» и в рабочей панели отобразятся действия:

 - создать класс

Выбираем действие «Класс», открывается форма создания класса, в ней необходимо указать системное и логическое имя, а также заполнить остальные свойства атрибутов, при необходимости (см. рисунок 5). К системному имени предъявляются требования наименования java-классов (латинские буквы, подчеркивание, цифры, запрещены системные переменные и функции). Значения атрибутов приведены в таблице 2.

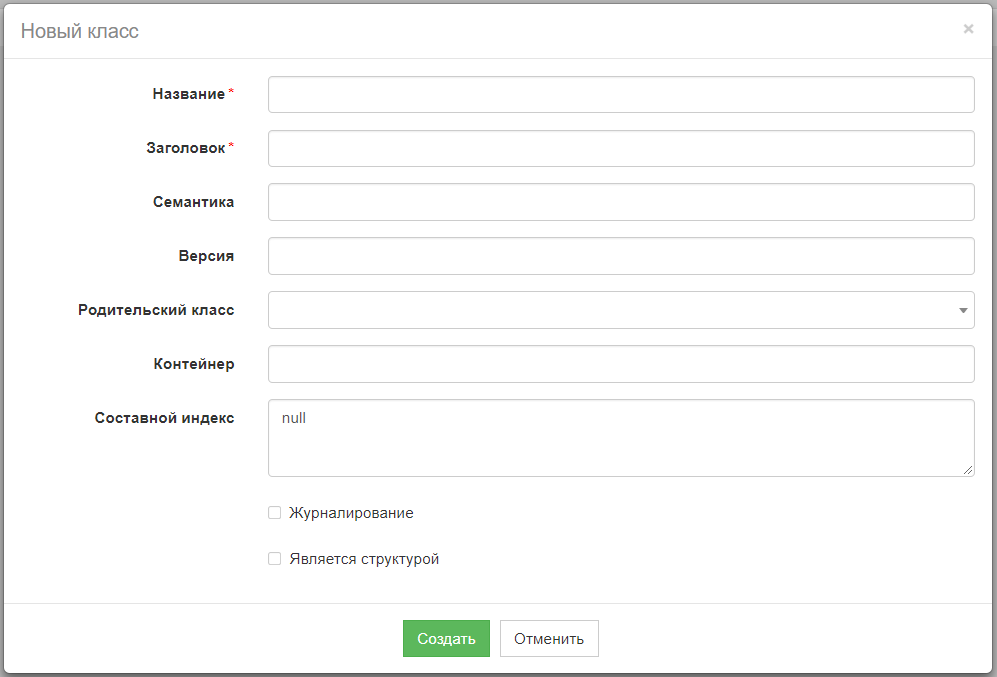


Рисунок 5 – Форма создания класса

Таблица 2 – свойства класса

|  |  |
| --- | --- |
| **Наименование атрибута** | **Назначение, характеристики** |
| Название | Предназначено для идентификации класса модели в системе. Имя должно состоять из символов латинского алфавита, без пробелов. В последующем поменять имя нельзя. |
| Заголовок | Используется для отражения класса в пользовательском интерфейсе. В последующем можно изменить. |
| Семантика | Атрибут, использующийся для приведения объекта к строке (несущий основную смысловую нагрузку) – указывается из состава атрибутов на вкладе «Атрибуты». При этом возможно использование выражений, задающих строковое представление объекта. (п. 4.1) |
| Версия | Позволяет задавать версионирвоание меты, для возможности оперирования данными созданными в разных версиях меты в рамках одной коллекции. |
| Родительский класс | Класс, от которого наследуется модель. Наследник будет содержать все атрибуты и свойства родительского класса, которые не будут переназначены |
| Контейнер | Здесь можно выбрать ссылочный атрибут, который будет использован для автоматического построения иерархической навигации. Объект, на который будет ссылаться выбранный атрибут, будет восприниматься средой как контейнер экземпляра доменного класса, и автоматически выстроит иерархию объектов. |
| Метка времени создания | Здесь можно указать атрибут типа «Дата-время» в который будет автоматически записываться момент создания объекта класса. |
| Метка времени изменения | Здесь можно указать атрибут типа «Дата-время», в который будет автоматически записываться момент последнего изменения объекта класса. |
| Журналирование (флаг) | Здесь можно указать необходимость журналирования всех изменений объекта. |
| Является структурой (флаг) | Обозначает, что класс модели является «Структурным типом», т.е. экземпляры этого класса могут существовать только как агрегаты (содержимое) какого-то объекта (контейнера). Такие типы предназначены для определения в модели устойчивых наборов атрибутов, характеризующих какой-то объект. Такие наборы атрибутов можно повторно использовать, в том числе в разных доменных моделях. |

Для создания класса-наследника, необходимо указать при создании сущности родительский класс.

## Удаление модели

Для удаления класса модели необходимо выделить либо пункт навигации в дереве проекта, либо таблицу в рабочей области с наименованием класса и в рабочей панели выбрать действие «Удалить класс», как приведено на рисунке 6.

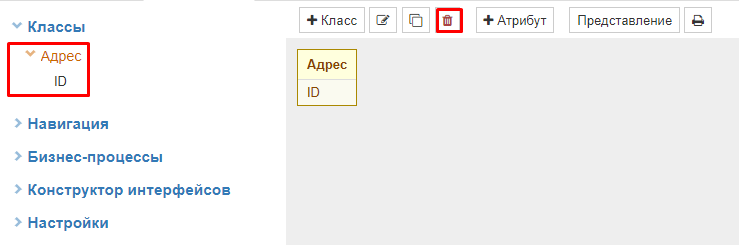


Рисунок 6 – Удаление класса

По двойному щелчку на классе модели в дереве проекта, либо с помощью действия  «Редактировать класс», отображается форма редактирования его свойств, как представлено на рисунке 7.

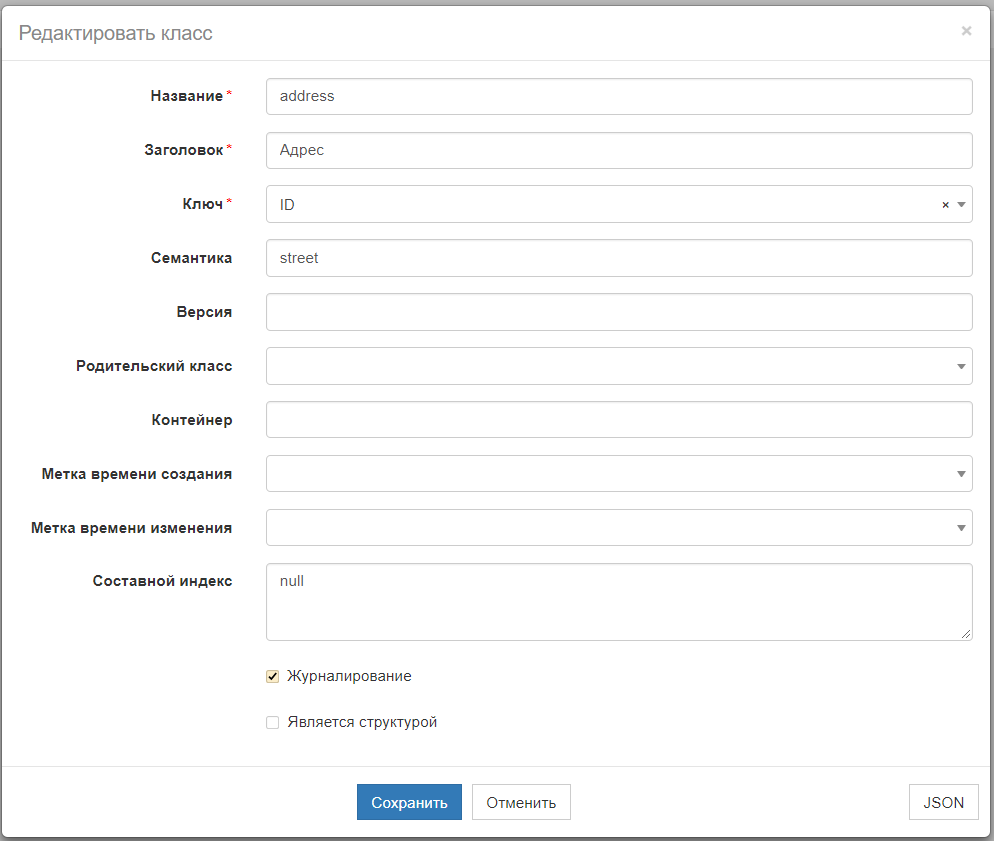


Рисунок 7 – форма редактирования класса

## Создание атрибутов сущности

При переходе на пункт навигации в дереве проекта, либо таблицы в рабочей области, с наименованием класса в рабочей панели отображается действие «Создать атрибут», как показано на рисунке 8.

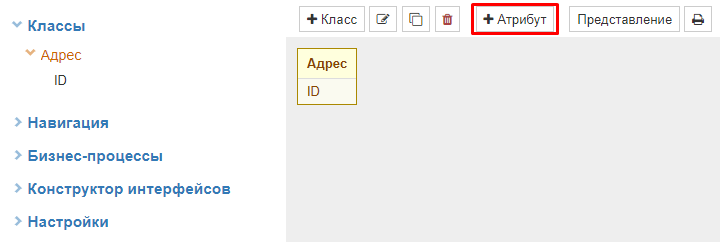


Рисунок 8 – создание атрибута

При этом вызывается окно создания атрибута модели, с полями, назначение которых представлено в таблице 3.

Таблица 3 – свойства атрибута

|  |  |
| --- | --- |
| **Наименование атрибута** | **Назначение, характеристики** |
| Название | Предназначено для идентификации атрибута. Имя должно состояться из символов латинского алфавита, без пробелов. В последующем поменять имя нельзя. |
| Заголовок | Используется для отображения атрибута в пользовательском интерфейсе. В последующем можно изменить. |
| Тип | Тип атрибута (таблица 3) |
| Размер | Максимальный размер данных атрибута. Для строковых значений — это максимальная длина строки, для числовых — разрядность. Используется при развертывании модели в хранилище. Используется по умолчанию для определения размера полей форм редактирования. Для безразмерных типов (Файл, Текст, HTML, Структура) не указывается. |
| Порядковый номер | Порядковый номер атрибута. Задает физический порядок расположения полей данных в хранилище. По умолчанию используется и для определения порядка полей в формах редактирования. Приветствуется при создании использование значений кратных 10-ти, для облегчения в последующем переназначения порядкового номера атрибута, без изменения номеров других атрибутов. |
| Размер | Максимальный размер данных атрибута. Для строковых значений — это максимальная длина строки, для числовых — разрядность. Используется при развертывании модели в хранилище. Используется по умолчанию для определения размера полей форм редактирования. Для безразмерных типов (Файл, Текст, HTML, Структура) не указывается. |
| Десятичные | Разрядность десятичных чисел (используется для типа атрибута «Десятичное») |
| Допустимые типы файлов | Расширения файлов, допустимые для значений данного атрибута (используется для типа атрибута «Файл», «Изображение») |
| Максимальное количество файлов | Задает максимальное количество файлов, которые пользователь может загрузить в атрибут типа "Коллекция файлов" |
| Подсказка | Задает сообщение, которые выведется в пользовательском интерфейсе рядом с именем атрибута. |
| Значение по умолчанию | Значение, присваиваемое атрибуту по умолчанию. Может быть пустым. |
| Ссылочный класс | Класс модели на который ссылается атрибут. Используется только для атрибутов типа «Ссылка». |
| Коллекция | Класс модели, экземпляры которого будут содержаться в коллекции. Используется только для атрибутов типа «Коллекция». |
| Семантика | Указывается для ссылочных атрибутов для того, чтобы выводить и формировать информацию из ссылочного класса, корректную для описания атрибута в объекте исходного класса. |
| Условия выборки | Позволяет ограничить выбор объектов по ссылке, допустимых для привязки в данном ссылочном атрибуте. |
| Сортировка выбора | Позволяет сортировать выбор объектов по ссылке, допустимых для привязки в данном ссылочном атрибуте. |
| Список выбора допустимых значений | Задает список выбора допустимых значений для атрибута типа «Строка», «Целое», «Десятичное», «Действительное», «Текст» |
| Жадная загрузка (флаг) | Загрузка объекта по ссылке. Если какие-то ссылочные атрибуты часто используются, либо объектов, на которые идет ссылка, много, их лучше грузить жадно, чтобы загрузка шла в одном запросе к БД. |
| Только для чтения (флаг) | Запрещает изменение значения атрибута пользователем, используется для создания полей формы по умолчанию |
| Индексируемый (флаг) | Используется при развертывании модели в хранилище данных. Указывает нужно ли индексировать значения данного атрибута для ускорения поиска. Целесообразно использовать для атрибутов, по которым предполагается атрибутивный поиск. |
| Допустимо пустое значение (флаг) | Разрешает или запрещает пустые значения атрибута |
| Уникальный (флаг) | Значение атрибута каждого экземпляра должно быть уникальным в контексте доменного класса. |
| Авто присеваемый (флаг) | Указывает, что значение данного атрибута должно быть заполнено автоматически при создании экземпляра класса. Имеет смысл только для атрибутов типа «Уникальный идентификатор», |

Таблица 4 – типы атрибутов

|  |  |
| --- | --- |
| **Тип атрибута** | **Характеристика** |
| Строка | Строковое значение |
| Целое | Целое числовое значение |
| Действительное | Действительное числовое значение (число с плавающей точкой) |
| Логическое | Логическое значение (флаг, да/нет) |
| Дата/время | Значение даты с точностью до секунд |
| Десятичное | Десятичное числовое значение (число с явным указанием количества знаков после запятой) |
| Ссылка | Ссылка на объект данных (экземпляр доменного класса) |
| Коллекция | Коллекция объектов данных (экземпляров доменного класса) |
| Файл | Идентификатор файла в файловом хранилище |
| Изображение | Идентификатор файла изображения в файловом хранилище |
| Текст | Многострочный текст |
| HTML | Текст содержащий гипертекстовую разметку либо Rich format |
| URL | URL |
| Глобальный идентификатор | Строковое значение уникальное в глобальном контексте (не повторяется нигде и никогда) |
| Пароль | Хеш пароля |
| Множество | Несколько значений из фиксированного предопределенного набора. Для использования атрибута такого типа необходимо указать список допустимых значений атрибута. |

Таким же образом создаем вторую сущность класса «Регион» [region] (необходимо для демонстрации связей между сущностями.

Для класса «Регион» выбираем действие «Создать атрибут», открывается форма создания атрибута. Заполняем обязательные поля свойств атрибутов как указано на рисунке 9, сохраняем форму.

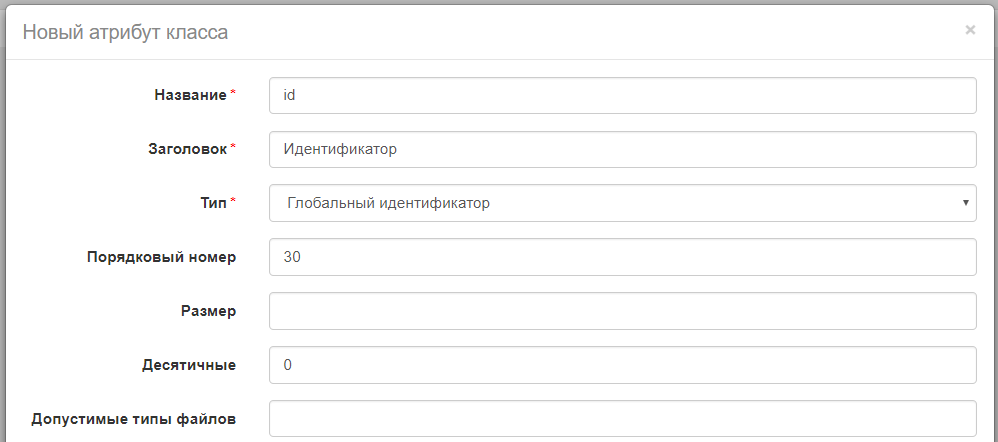


Рисунок 9 – форма создания атрибута

Аналогично создаем атрибут «Наименование» [name] с типом «Строка» для данного класса. Отображение сущности класса «Регион» в виде таблицы в рабочей области и в иерархической структуре меню представлено на рисунке 10.

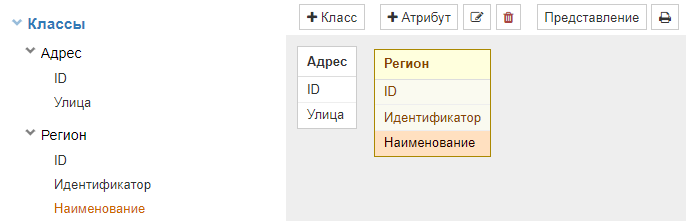


Рисунок 10 – отображение сущности в рабочей области

Далее создадим связь между сущностями приложения. Для этого выделяем пункт меню «Адрес» и выбираем действие «Создать атрибут» (действие отобразится также при выделении таблицы «Адрес» в рабочей области Подсистемы). Создаем атрибут «Идентификатор» [id] и атрибут «Регион» [region] с типом «Ссылка», для него в поле свойства «Ссылочный класс» из списка выбора допустимых значений выбираем класс «Регион».

При указании связей в мете класса - связь между таблицами (сущностями класса) настроена автоматически (см. рисунок 11)

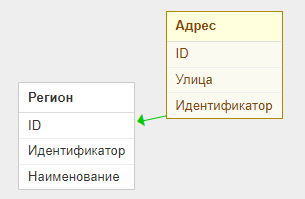


Рисунок 11 – связь между сущностями класса

## Схема связи таблиц данных

Диаграмма в рабочей области Подсистемы, процесс создания которой описан в п. 2.2 – 2.4, состоит из таблиц сущности и связей между ними (см. рисунок 12)

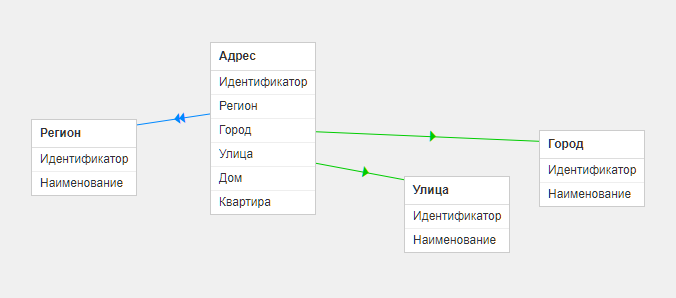


Рисунок 12 – диаграмма сущностей

Связи между таблицами на диаграмме, указывают на отношение атрибута класса к сущности и разделяются цветом по типу связи. Зеленый цвет – указывает на связь типа «один-к-одному», синий цвет связи – указывает на тип «один-ко-многим».

Действия, направленные на изменение структуры данных в схеме, осуществляются при помощи рабочей панели (см. рисунок 13)



Рисунок 13 – рабочая панель диаграммы

Определение сущности для наследования происходит с помощью создания новой сущности, используя действие на рабочей панели «Создать класс» и определение для него родительского класса способом выбора из списка существующих классов в поле «Родительский класс» на форме создания класса-наследника. После заполнения обязательных полей и сохранения формы – наблюдаем отображение нового элемента на диаграмме в рабочей области в виде связи с родительским классом.

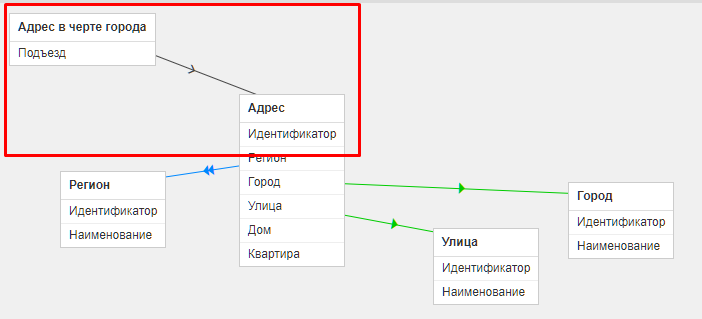


Рисунок 14 – пример наследования

# Динамические формы

## Задание условий для отображения полей

Для того чтобы задать список атрибутов, выводимых на форму, в зависимости от выбранного типа представления, нужно перейти в рабочей панели по кнопке «Представление». Кнопка доступна для сущностей, либо их атрибутов (см. рисунок 15)

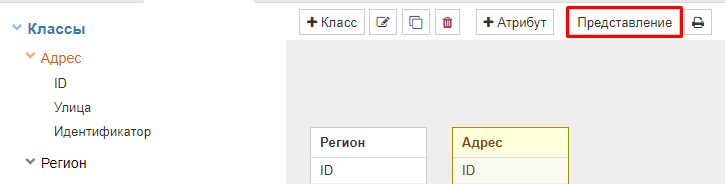


Рисунок 15 – создание представления класса

Из выпадающего списка выбора допустимых значений в поле «Представление» выбираем тип формы представления для класса, в соответствии с таблицей 5.

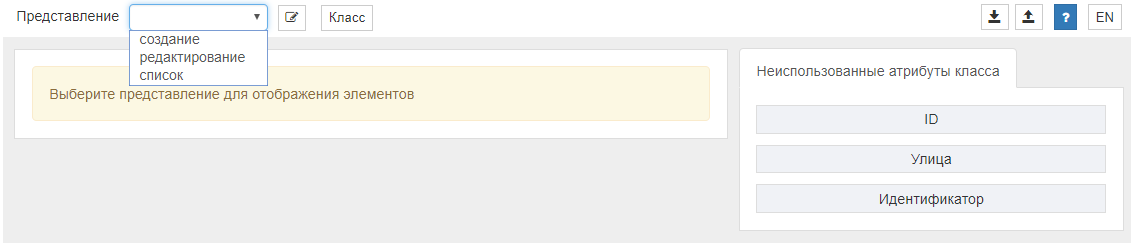


Рисунок 16 – выбор типа представления

Таблица 5 – типы представления

|  |  |
| --- | --- |
| **Тип представления** | **Назначение** |
| Список объектов | обеспечивает вывод списка объектов указанного класса |
| Форма создания | обеспечивает вывод формы создания объекта указанного класса |
| Форма изменения | обеспечивает вывод формы изменения объекта указанного класса |

Далее отображается список всех атрибутов класса и область для формирования формы представления в соответствии с выбранным типом.

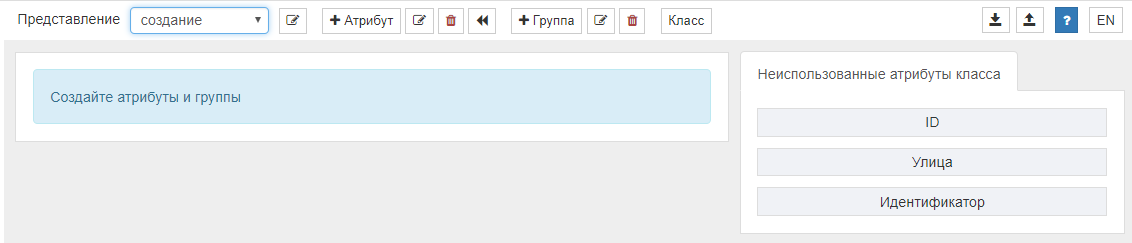


Рисунок 17 – форма представления «Создание объектов»

Путем перетаскивания переносим атрибут из правой колонки в левую область формирования формы представления. Двойным щелчком по наименованию атрибута открываем его форму редактирования, задаем для него свойства, значения которых указаны в таблице 6, сохраняем форму.

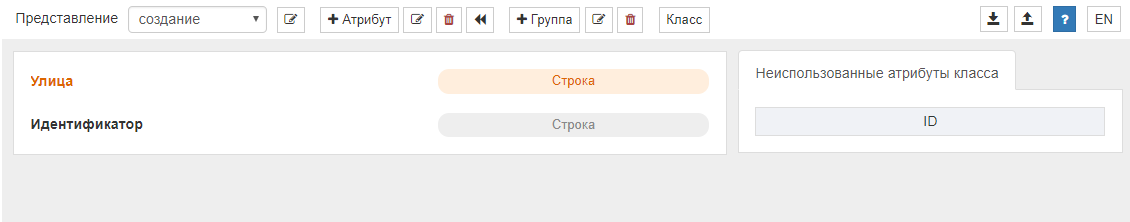


Рисунок 18 – область формирования формы представления

Таблица 6 – свойства атрибута на форме представления

|  |  |
| --- | --- |
| Наименование свойства | Назначение, характеристики |
| Название | Указывается имя атрибута, с которым будет работать система, а значит не может быть пустым (Кроме типа представления «Группа») |
| Заголовок | Заголовок поля ввода/колонки атрибута отображаемый в представлениях. |
| Группа | Группирует на форме атрибуты, задается согласно п. 3.2 |
| Тип | Задается тип поля. Доступны следующие типы полей:   * Текстовое — будет отображено простое поле для ввода строки текста * Выпадающий список — будет отображено поле со выпадающим списком для выбора значения * Флаг — будет отображено поле для указания логического значения да/нет * Выбор даты — будет отображено поле для указания. Используется только для полей типа «Коллекция» в режимах отображения «Список» и «Таблица» даты и времени * Выбор файла — будет отображено поле для загрузки файла * Выбор изображения — будет отображено поле для загрузки изображения * Многострочный текст — будет отображено поле для ввода многострочного текста * Форматированный текст — будет отображено поле для ввода текста с форматированием * Множественный выбор — будет отображено поле для выбора нескольких значений из списка * Пароль — будет отображено поле для ввода пароля * Редактор целых чисел — будет отображено поле для ввода целых чисел * Редактор вещественных чисел — будет отображено поле для ввода действительных чисел * Группа — будет отображена группа полей. Поля данного типа могут содержать другие поля. * Ссылка — будет отображено поле для выбора объекта ссылки * Коллекция — будет отображено поле для вывода коллекции объектов |
| Подсказка | Задает (или переопределяет заданное в поле «Подсказка» меты атрибута) сообщение, которые выведется в пользовательском интерфейсе рядом с именем атрибута. |
| Порядковый номер | Порядковый номер атрибута задает расположение атрибута относительно других атрибутов этого же класса в пользовательском интерфейсе. |
| Размер | Указывает условный размер поля. Реальный размер будет определяться конкретной реализацией приложения. Доступны следующие размеры:   * Малый * Меньше среднего * Средний * Больше среднего * Большой |
| Название маски | При наличии в платформе предустановленных масок - можно задать маску по внутреннему имени, указав его в данном поле |
| Маска | Позволяет ограничить допустимые значения атрибута (п. 4.2) |
| Режим | Задает режим отображения для полей типа «Ссылка» и «Коллекция».  Ссылочные поля могут отображаться в следующих режимах:   * Ссылка — адрес страницы отображения объекта * Строка — строковое представление объекта   Поля типа «Коллекция» могут отображаться в следующих режимах:   * Список — вывод списка объектов коллекции в свободной форме * Таблица — вывод таблицы списка объектов коллекции |
| Обязательный (флаг) | Определяет обязательно ли заполнение данного атрибута при создании/изменении объекта. |
| Только для чтения (флаг) | Разрешает или запрещает изменять значение атрибута в данном представлении. |
| Постраничный выбор (флаг) | Данный признак используется для атрибутов типа «Ссылка» и «Коллекция». Для атрибутов типа «Ссылка» признак означает, что количество объектов, на которые потенциально может ссылаться атрибут мало. Для атрибутов типа «Коллекция» признак означает, что в коллекции потенциально будет находится малое количество объектов. Может использоваться конкретной реализацией приложения для оптимизации работы с объектом в пользовательском интерфейсе, так как подразумевает, что список объектов для выбора или отображения не требует постраничной навигации. |
| Теги | Поле предназначено для хранения модификаторов отображения и прочей служебной информации. |

Перемещение компонентов из палитры на рабочую область можно осуществлять способом переноса. Выбираем компонент из «Списка неиспользованных атрибутов класса» и переносим вручную его в область формирования представления. Далее задаем для него свойства, открыв форму редактирования компонента. Убрать атрибут из области формирования обратно на палитру можно с помощью действия «Удалить атрибут» на рабочей панели области формирования форм представления класса.

### **Редактор кода**

Некоторые значения свойств имеют объектный формат и указываются в виде кода. При вводе значений для таких полей выбираем его на форме, после чего открывается окно для ввода значения в виде кода в формате `.json`, при вводе подсвечивается синтаксис вводимого значения (рисунок 19). При не верном формате ввода – отобразится сообщение от ошибки сохранить форму будет невозможно до исправления ошибки. После сохранении формы код отображается как значение атрибута на форме.

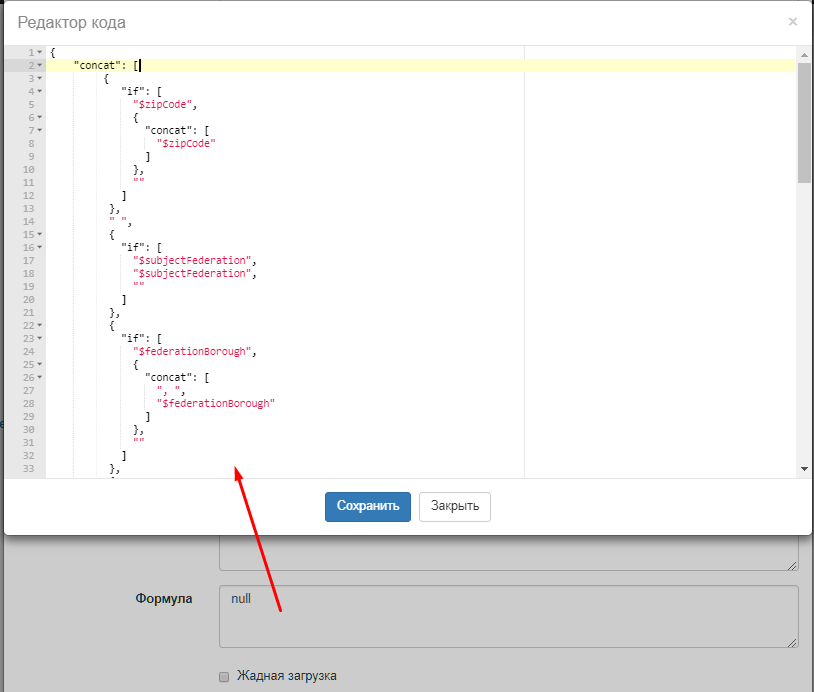


Рисунок 19 – ввод значения с подсветкой синтаксиса

## Способы группировки атрибутов на форме

Для создания группы атрибутов на форме необходимо выбрать действие «Группа»  на рабочей панели, после чего откроется форма создания группы.

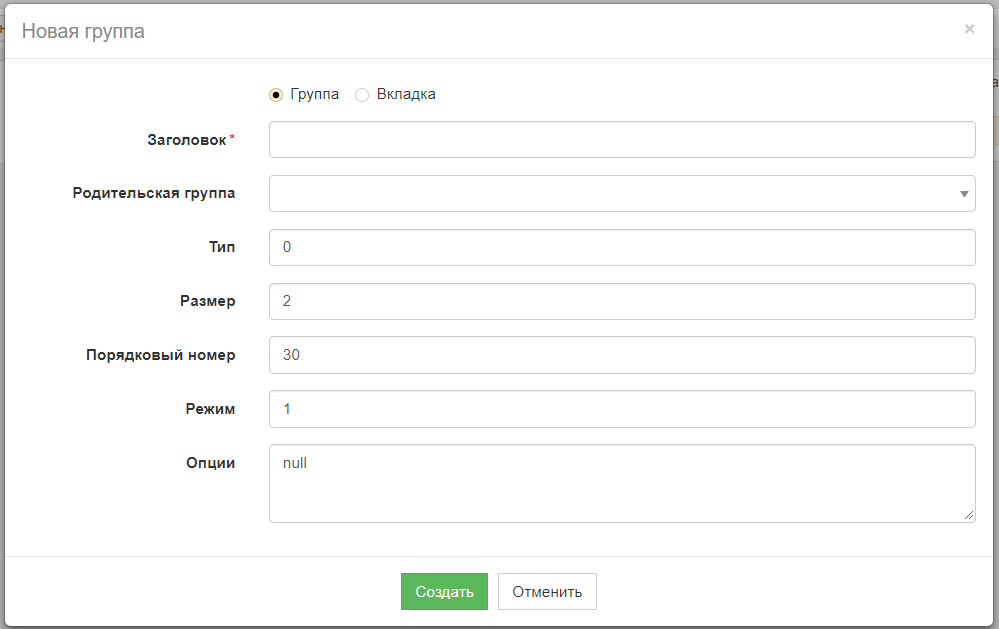


Рисунок 20 – создание группы

Для данной формы доступно создание:

* Группа – Позволяет группировать атрибуты в рамках одной формы.
* Вкладка - Задаются в представлении создания и редактирования и используются для разбиения атрибутов класса по отдельным вкладкам на форме.

Заполняем свойства группы для отображения их на форме представления атрибутов класса. Создаем группу «Общая информация» об адресе и на форме редактирования свойств атрибута «Регион» и «Город» задаем значение для свойства «Группа».

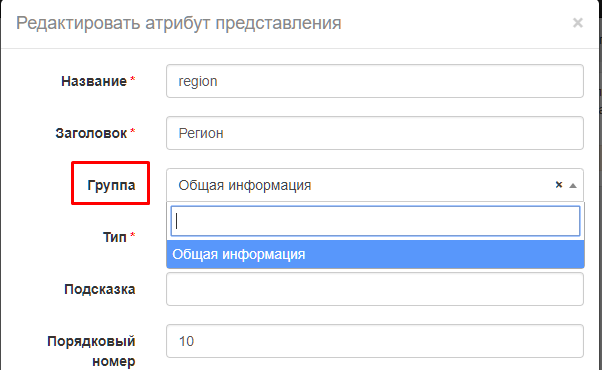


Рисунок 21 – формирование группы

Теперь, в области формирования списка атрибутов для формы представления, отобразилась группа для атрибутов, присвоенных ей, как показано на рисунке 22.

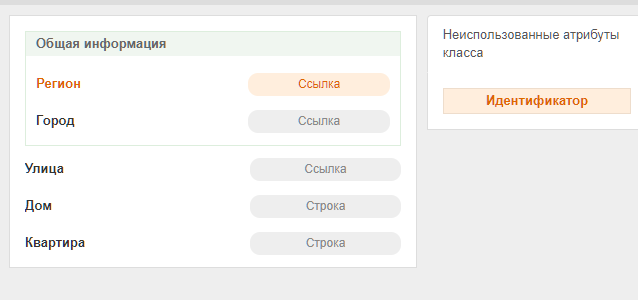


Рисунок 22 – отображение группы

Для создания вкладки выбираем тип формы представления для необходимого класса и открываем форму создания «Группы», переключаем форму на «Вкладка» и заполняем ключевые поля и прочие свойства.

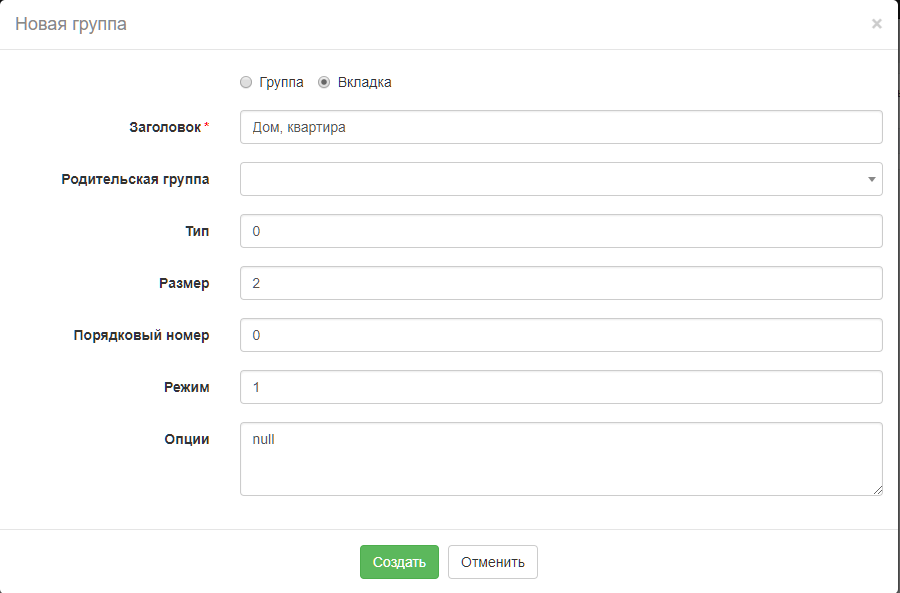


Рисунок 23 – создание вкладки

Далее открываем форму редактирования атрибутов текущего класса и для атрибутов, которые необходимо отобразить в созданной вкладке, в свойстве «Группа» из списка выбора допустимых значений выбираем необходимую вкладку по наименованию.

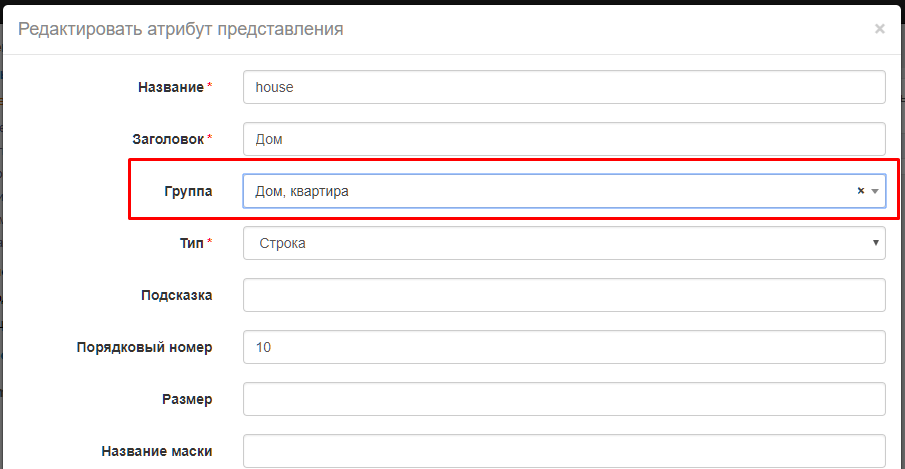


Рисунок 24 – формирование вкладки

Сохраняем форму, аналогично соотносим атрибуты класса по вкладкам и в результате видим отображение атрибутов по вкладкам в области формирования форм представления.

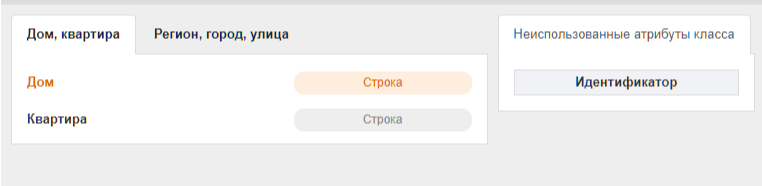


Рисунок 25 – отображение вкладки

## Клонирование элементов

При необходимости размножить компоненты или группы приложения доступен функционал клонирования, который позволяет создавать копии необходимого компонента, либо группы компонентов для дальнейшего использования.

Клонирование доступно для следующих компонентов Подсистемы:

* Сущность
* Бизнес-процесс (а также его состояния и переходы)
* Формы

Действия для клонирования компонентов системы:

* 1. Выделяем компонент для клонирования
  2. На рабочей панели выбираем действие «Клонировать» (см. рисунок 26)

1. После подтверждения действия открывается форма редактирования клонированного элемента, с добавлением префикса «clone\_» для обязательных полей формы. При необходимости, заполняем остальные поля формы, сохраняем форму.
   1. Клонируемый компонент отобразился в текущей рабочей области, с добавлением префикса «clone\_» в название (см. рисунок 27)

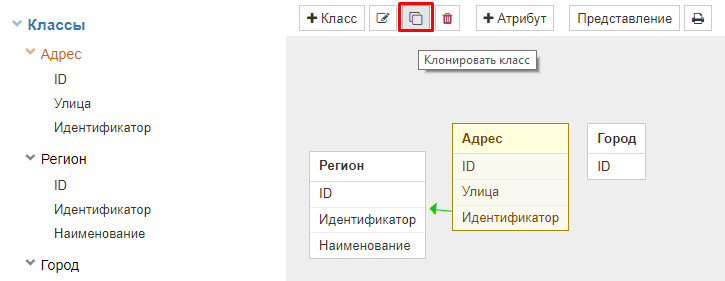


Рисунок 26 – действие «Клонировать»

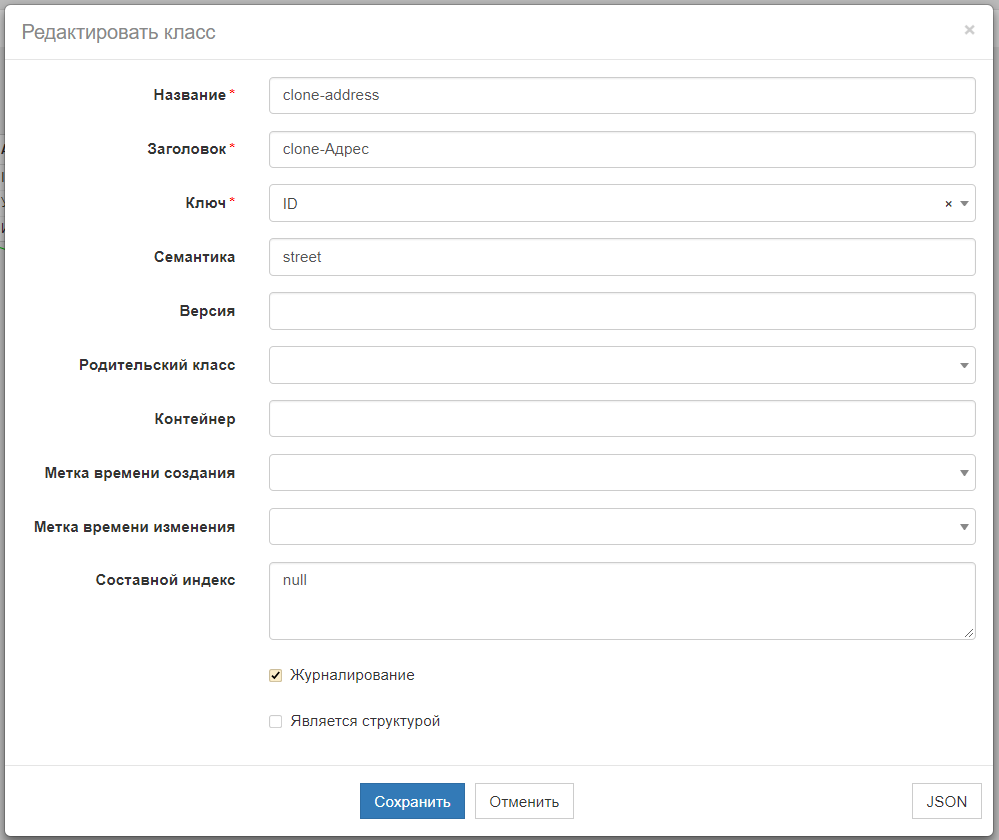


Рисунок 27 – форма редактирования клонированного компонента

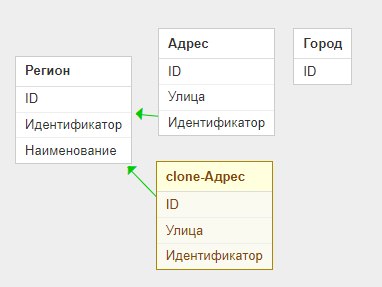


Рисунок 28 – клонированный компонент

## 3.4 Автоматическое формирование формы представления

Если на форму представления класса необходимо выводить весь атрибутивный состав класса, то можно не переносить атрибуты из палитры в область формирования атрибутивного состава формы представления, а воспользоваться действием «Добавить все атрибуты», расположенной на рабочей панели:



Данной действие переносит все атрибуты из палитры в область формирования формы представления. Так же позволяет переносить атрибуты в заранее созданную группу или вкладку в рабочей области.

# 4 Печатные формы

Выбираем класс, для которого необходимо выводить данные в печатной форме, переходим в рабочую область печатной формы при помощи действия в рабочей панели «Печатная форма».

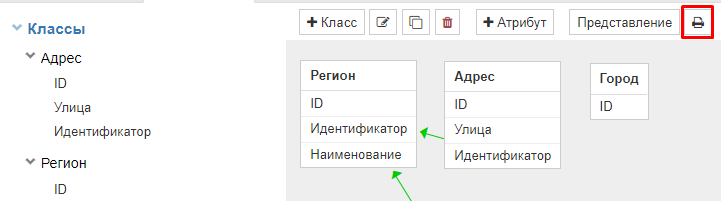


Рисунок 29 – переход на рабочую область Печатной формы

В рабочей панели доступны действия над печатной формой (создать, редактировать, удалить) и действие для перехода к рабочей области класса. Тут же отображается табличка, в которой будут хранится созданные элементы печатных форм, содержащие заданные для них свойства и сам файлик для формирования печатной формы.

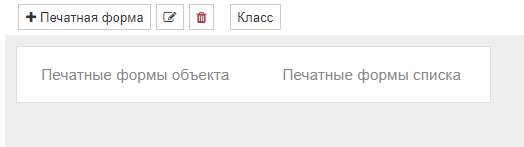


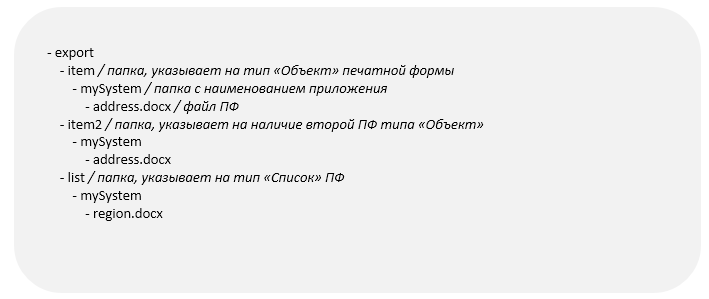
Рисунок 30 – рабочая панель печатной формы

Выбираем действие «Создать печатную форму». Открывается форма создания печатной формы, поля которой описаны в таблице 7.

Таблица 7 – свойства печатной формы

|  |  |
| --- | --- |
| **Свойство печатной формы** | **Описание свойства** |
| Тип | Объект – печатная форма определяется для представления формы изменения класса. |
| Список – печатная форма определяется для формы представления списка класса. |
| Название | Системное наименование печатной формы. |
| Заголовок | Логическое наименование печатной формы. |
| Тип MIME | Тип файла печатной формы. По умолчанию значение поля задано как «*application/vnd.openxmlformats-officedocument.wordprocessingml.document*». |
| Расширение | Расширение файла печатной формы. По умолчанию значение поля указано как «*docx*». |
| Файл | Файл, содержащий в себе настройки для формирования печатной формы. Есть два способа добавления файла в поле: перетаскиванием файла в область формы создания или выбором из структуры папок рабочего стола, с помощью кнопки «Выберите файл». |

При импорте приложения с заданными печатными формами, они хранятся в папке «export» и имеют следующую структуру:



# Навигация

Навигация – элемент интерфейса пользователя, позволяющий наглядно отобразить структуру данных приложения в иерархическом виде. Пункты меню формируются за счет сущностей, созданных в рамках приложения. Навигации состоит секций и пунктов, которые содержит секция. Также возможно создание дочерних пунктов навигации. Описание элементов навигации указано в таблице 8.

Таблица 8 – элементы навигации

|  |  |
| --- | --- |
| **Элемент навигации** | **Описание** |
| Секция | Объединяет в себе пункты навигации приложения по какому-либо общему признаку. |
| Пункт | Может использоваться как страница класса, то есть отображать структуру и объекты сущности приложения. Но также может объединять в себе по какому-либо общему признаку дочерние пункты навигации, не являясь при этом отображением страницы класса приложения. Возможности иерархической структуры навигации не ограничены. |
| Дочерний пункт | Данный пункт должен быть создан в пункте навигации, не являющемся страницей класса. В остальном подобен элементу навигации «Пункт». |

При создании навигации приложения, в первую очередь создается секция навигации. Для этого необходимо перейти в раздел «Навигация» в боковом меню Подсистемы конструирования. В рабочей области доступно действие «Создать секцию».

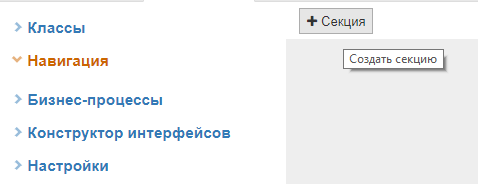


Рисунок 31 – создание секции навигации

Далее открывается форма создания секции навигации. В ней необходимо заполнить поля, как показано на рисунке 32.

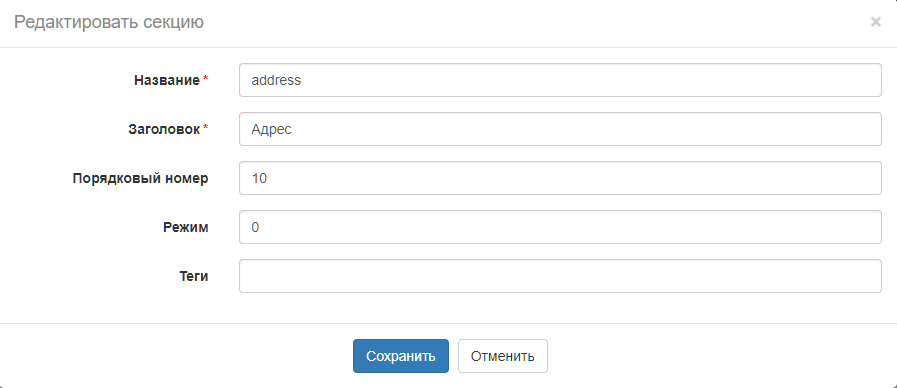


Рисунок 32 – форма создания секции навигации

После сохранения формы, на рабочей панели расширились возможности действий над секцией навигации и стало доступно действие для создания пункта навигации.

Выбираем действие «Создать пункт», после чего открывается форма создания пункта навигации. Заполняем поля на форме, в соответствии с рисунком 33.

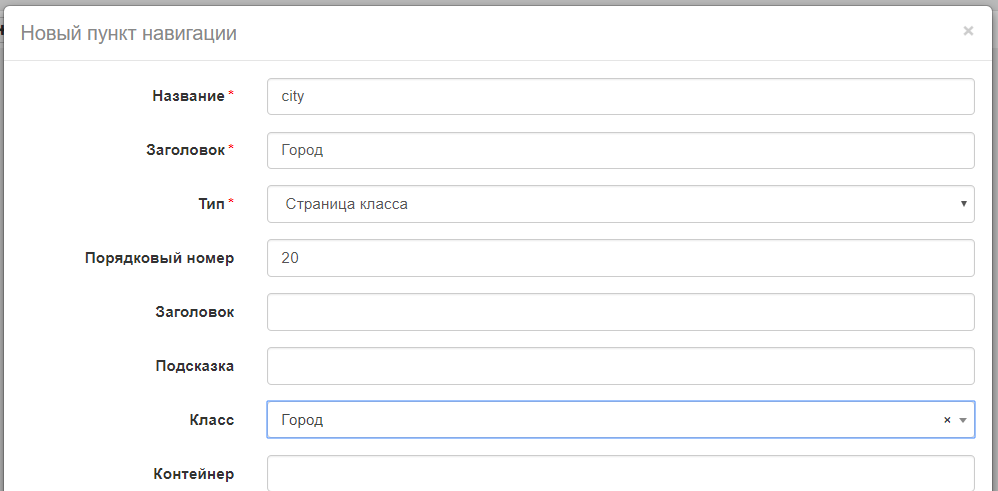


Рисунок 33 – форма создания пункта навигации

После сохранения формы создания пункта навигации, переходим по нему в боковом меню Подсистемы конструирования. В рабочей области видим отображение формы создания списка той сущности, которая была указана на форме создания данного пункта навигации в поле «Страница класса».

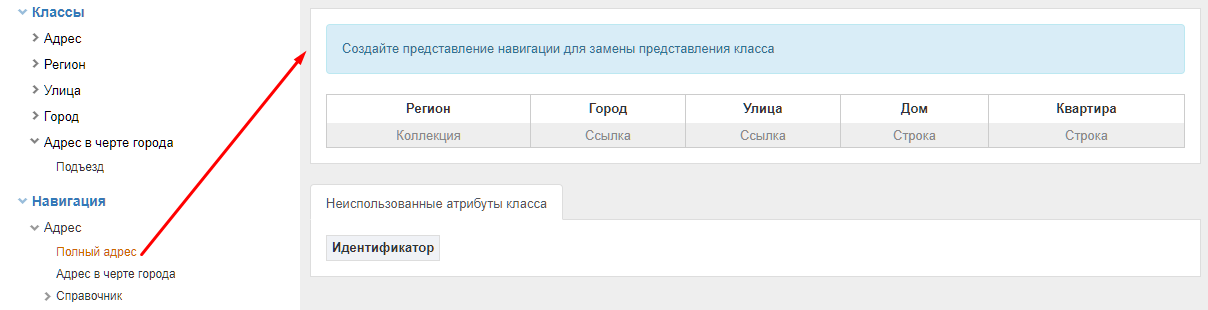


Рисунок 34 – рабочая область пункта навигации

Как видно из рисунка 34, для пункта навигации есть возможность формировать индивидуальный атрибутивный состав формы представления списка для пункта навигации, не зависимо от состава формы представления списка, заданной при помощи редактора форм. Рабочая область пункта навигации, являющегося страницей класса показана на рисунке 36. Для создания формы представления списка для пункта навигации, выбираем действие «Создать представление навигации».

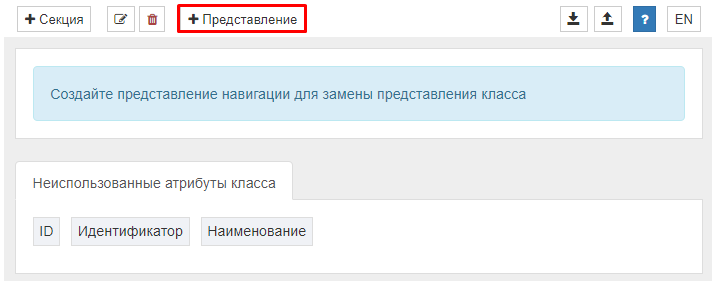


Рисунок 35 – создание представления для пункта навигации

Открывается форма создания представления списка для пункта навигации, на ней можно переопределить значения полей, заданные для формы представления списка класса. Свойства, заданные в данной форме, будут переменяться только к представлению списка для данного пункта навигации.

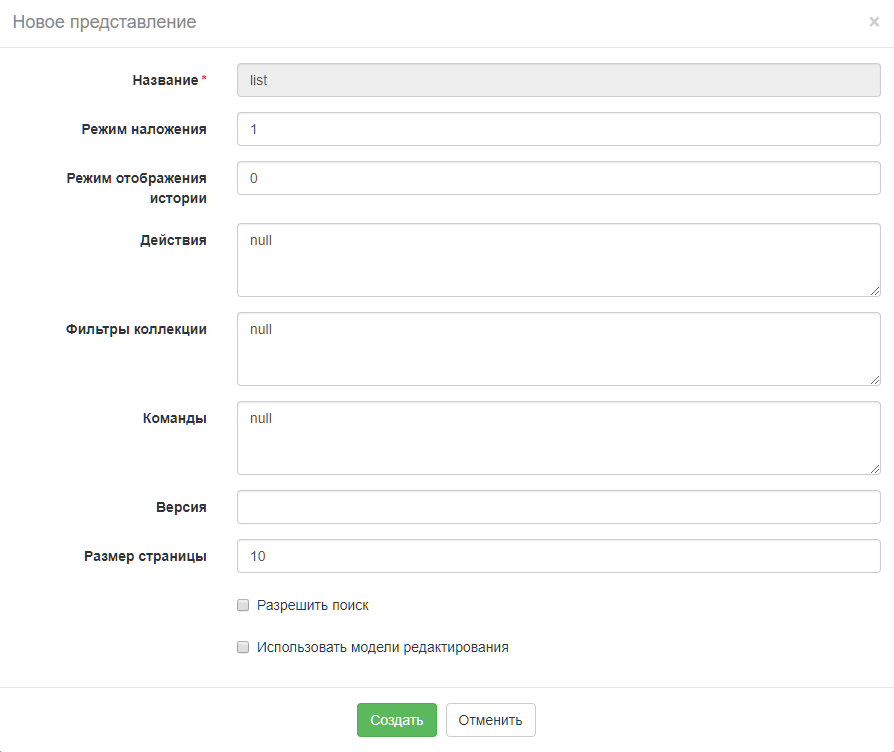


Рисунок 36 – форма создания представления списка для пункта навигации

После сохранения формы – элементы рабочей области становятся доступными для редактирования, а в рабочей панели для представления появляются действия «Редактировать представление навигации» - позволяет редактировать свойства формы представления для пункта навигации, «Удалить представление навигации» - позволяет удалить представление пункта навигации и вернуться к общему представлению, созданного для класса.

Далее формируем атрибутивный состав формы представления списка для пункта навигации. Выполняется операция аналогично процессу формирования атрибутивного состава на формах представления, описание которого выполнено в п.3 настоящего документа РП.

**Создание дочернего пункта навигации.**

Для начала необходимо создать пункт навигации, не являющийся страницей класса. Дочерние пункты навигации, сформированные по какому-либо признаку, будут находится в этом пункте. Для уже созданной нами секции навигации выбираем действие «Создать пункт» и на появившейся форме заполняем поля как показано на рисунке 37.

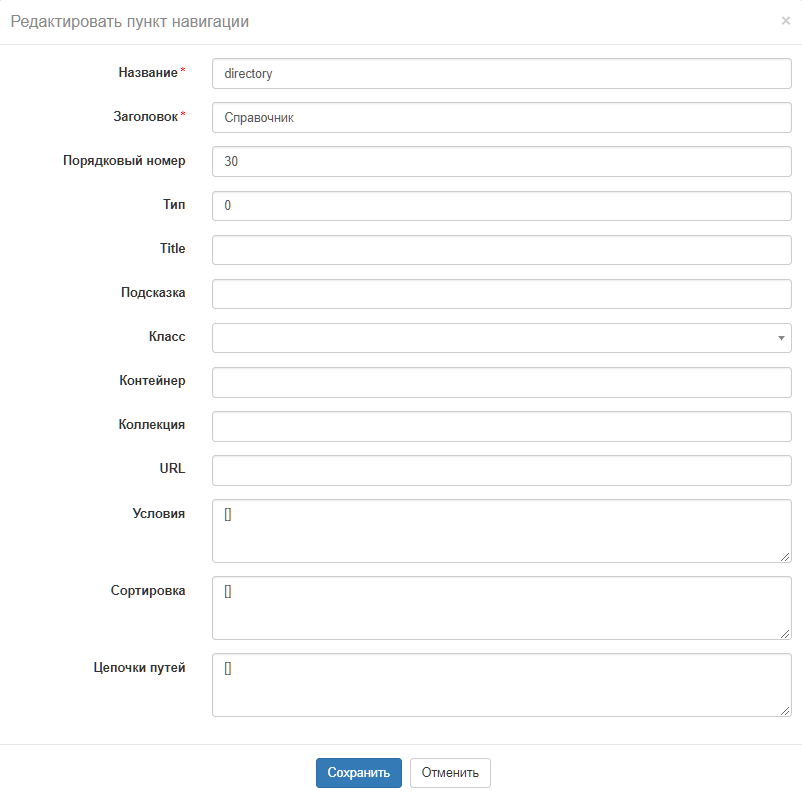


Рисунок 37 – создание пункта навигации, содержащего дочерние пункты

Отличительным признаком пункта навигации, объединяющего дочерние классы и пункта навигации, представляющего страницу класса, является наличие заполненного поля «Страница класса».

Далее переходим в созданный пункт навигации «Справочники» и на рабочей панели для данного пункта выбираем действие «Создать пункт». На отрытой форме создания заполняем поля как показано на рисунке 38.

После сохранения формы, созданный дочерний пункт навигации отобразится в иерархической структуре навигации на боковой панели Подсистемы конструирования.

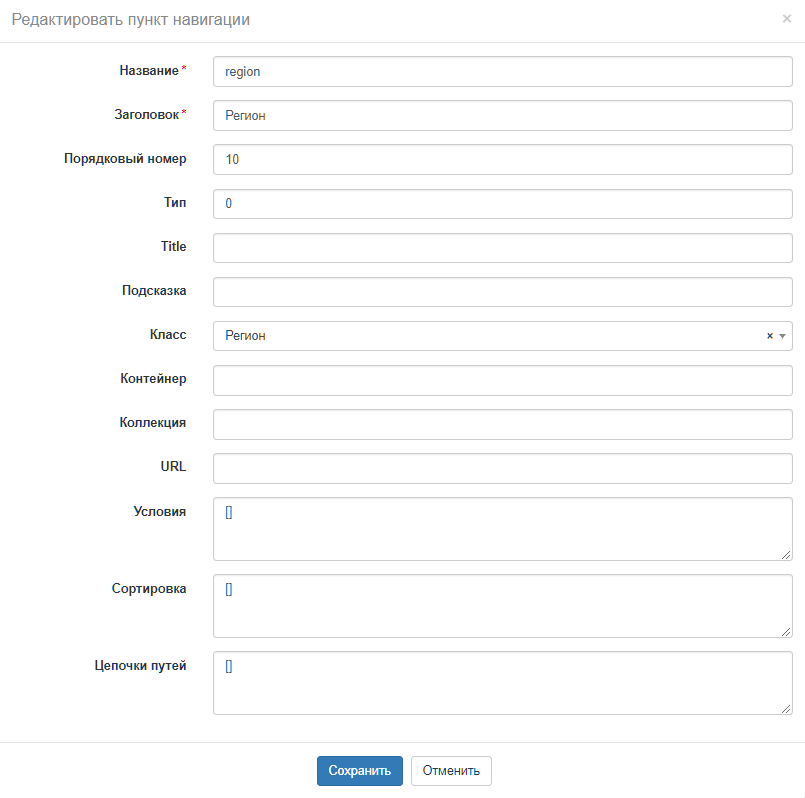


Рисунок 38 – создание дочернего пункта навигации

# Специфичные настройки отображения атрибутов

## Настройка маски в атрибутах

Маска задается для атрибута на форме представления класса. Для указания масок используется синтаксис masked-input-plugin. Описание синтаксиса можно посмотреть здесь: <http://digitalbush.com/projects/masked-input-plugin/>.

Для указания маски открываем рабочую область «Представление», выбираем класс и в нем открываем двойным щелчком форму редактирования атрибута, для которого необходимо задать маску и вводим значение для свойства атрибута «Маска», в соответствии с синтаксисом указанным на странице <http://digitalbush.com/projects/masked-input-plugin/>

А) a - Represents an alpha character (A-Z,a-z)

Б) 9 - Represents a numeric character (0-9)

В) \* - Represents an alphanumeric character (A-Z,a-z,0-9)

## Указание на Обязательность атрибута

Для того чтобы указать обязательность на атрибуте в представлении поля - открываем рабочую область «Представление», выбираем класс и в нем открываем двойным щелчком форму редактирования атрибута, для которого необходимо задать обязательность и выставляем «флаг» в соответствующем поле Обязательность.

## Условия отбора допустимых значений

Условие задается на форме создания/редактирования сущности в поле «Условие отбора» и представляет собой фильтр, который накладывает условия ограничения выборки объектов. Применяется для атрибутов типа «Ссылка» и «Коллекция».

*Доступные операции:*

* EQUAL: 0, // равно =
* NOT\_EQUAL: 1, // не равно <>
* EMPTY: 2, // пусто '' или null
* NOT\_EMPTY: 3, // не пусто !'' или !null
* LIKE: 4, // похож
* LESS: 5, // меньше <
* MORE: 6, // больше >
* LESS\_OR\_EQUAL: 7, // меньше или равно <=
* MORE\_OR\_EQUAL: 8, // больше или равно >=
* IN: 9, // похож (IN)
* CONTAINS: 10 // содержит

## Сортировка выборки допустимых значений

Условие задается на форме создания/редактирования сущности в поле «Сортировка выбора» и представляет собой фильтр, который задает способ сортировки объектов. Применяется для атрибутов типа «Ссылка».

*Доступные типы сортировки:*

* Сортировка по возрастанию
* Сортировка по убыванию

## Список выбора допустимых значений

Список формируется в виде массива объектов типа «ключ-значения» и представляет собой список выбора значения для атрибута с типом «Строка», «Действительное», «Целое», «Десятичное», «Текст».

*Синтаксис формирования списка:*

{“type”: “SIMPLE”, “list”: [{“key”: “string1”, “value”: “Строка 1”}], [{“key”: “string2”, “value”: “Строка 2”}], …}

## 

# Бизнес – процессы

Бизнес-процесс строится на основе сущностей, созданных в п. 2, 3. Переходим в дереве приложения в пункт навигации «Бизнес-процессы», в рабочей панели выбираем действие «Создать бизнес-процесс», как показано на рисунке 39.

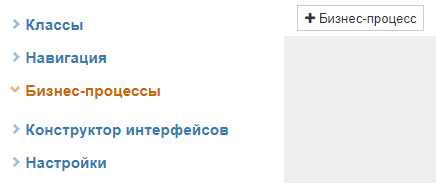


Рисунок 39 – рабочая панель бизнес-процесса

Открывается форма создания бизнес-процесса, заполняем обязательные поля.

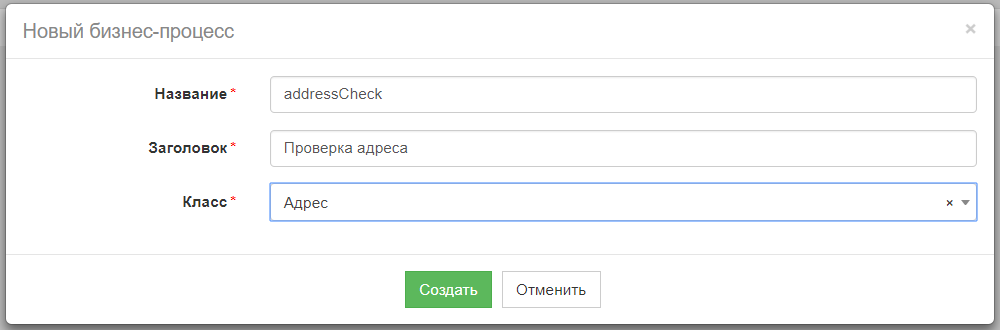


Рисунок 40 – создание бизнес-процесса

После сохранения формы и перехода по пункту меню созданного бизнес-процесса – в рабочей панели выбираем действие «Создать состояние». Заполняем данные для состояния бизнес-процесса, в соответствии с описанием его свойств в таблице 9.

Таблица 9 – Описание свойств состояния

|  |  |
| --- | --- |
| **Свойство состояния** | **Описание** |
| Наименования | Системное имя перехода |
| Заголовок | Логическое имя перехода |
| Условия | Задается условие для состояния. *Формат:*    “property” – системное имя атрибута  “operation” – операции (таблица 8)  “value” – значение атрибута  “nestedConditions” – вложенные условия |
| Объектные разрешения | Задается разрешение для объекта, находящегося в данном состоянии БП. |
| Атрибутные разрешения | Задается разрешение для атрибута |
| Выборка допустимых значений | Допустимые значения для состояния бизнес-процесса |

Форма создания состояния для бизнес-процесса показана на рисунке 41.

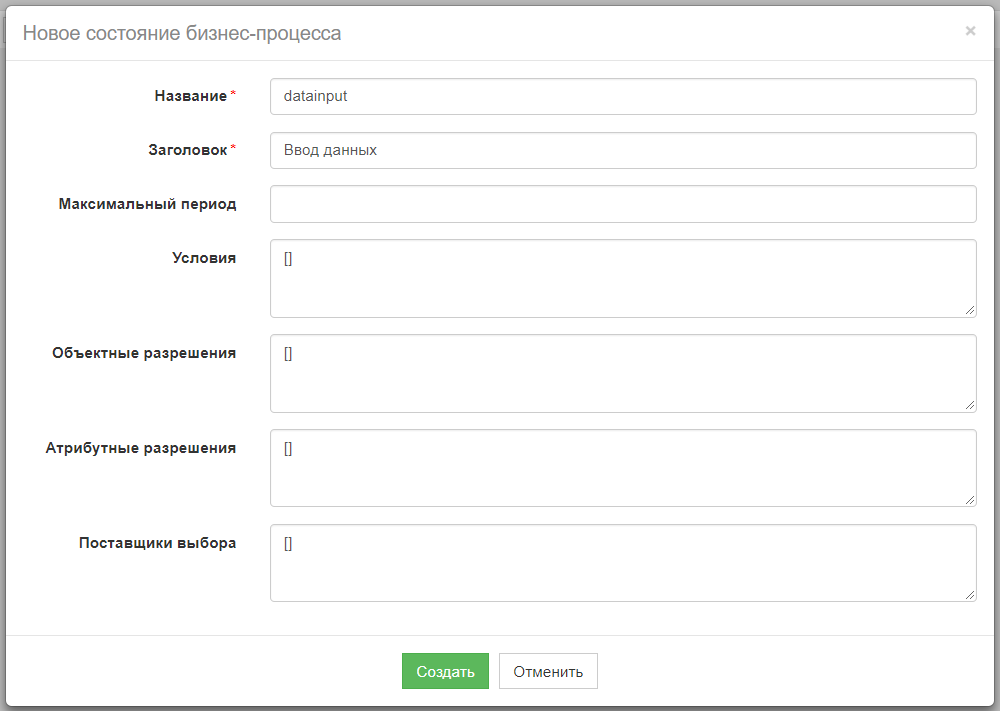


Рисунок 41 – создание состояния

Создаем еще два состояния:

* «Проверка адреса» [addressVerification]
* «Проверен» [checked]

Далее, при переходе на пункт меню с наименованием состояния, на рабочей панели становится доступно действие «Создать переход». Создаем переходы для каждого состояния в соответствии с описанием его свойств в таблице 10, согласно схеме 1.

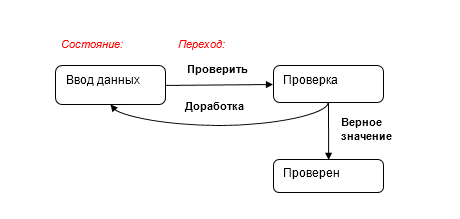


Схема 1 – Бизнес-процесс «Проверка адреса»

Таблица 10 – описание свойств перехода

|  |  |
| --- | --- |
| **Свойство перехода** | **Описание перехода** |
| Наименование | Системное имя перехода |
| Заголовок | Логическое имя перехода |
| Начальное состояние | Начальное состояние БП для совершения перехода |
| Конечное состояние | Состояние БП, присваиваемое при завершении перехода |
| Подписать до (флаг) | ЭП проставляется до начала перехода |
| Подписать после (флаг) | ЭП проставляется по завершению перехода |
| Требуется подтверждение (флаг) | Подтверждение перехода |
| Подтверждающее сообщение | Сообщение, отображаемое при подтверждении перехода |
| Присвоение | Присвоение значения атрибуту при завершении перехода. Формат:    “key” – системное имя атрибута  “value” – присваиваемое значение |
| Условия | Условие для совершения перехода. Формат:    “property” – системное имя атрибута  “operation” – операции (таблица 8)  “value” – значение атрибута  “nestedConditions” – вложенные условия |
| Роли | Роли, для которых доступен переход по бизнес-процессу |

Таблица 11 – операции для выборки

|  |  |
| --- | --- |
| Операция | Значение |
| 0 | Равно |
| 1 | Не равно |
| 2 | Пусто |
| 3 | Не пусто |
| 5 | Меньше |
| 6 | Больше |
| 7 | Меньше или равно |
| 8 | Больше или равно |
| 9 | Похож |
| 10 | Содержит |

Также с помощью переходов по бизнес-процессу можно отправлять SOAP запросы во внешнюю систему. Достаточно задать условие для возможности перехода по БП, которое проверяет готовность запроса к отправке и по завершению перехода, во время которого и произойдет отправка данных, присвоить отметку об успешной отправке запроса.

Далее возвращаемся на форму редактирования бизнес-процесса в целом (для этого переходим в пункт меню «Проверка адреса» и на рабочей панели выбираем действие «Редактировать бизнес-процесс») и задаем значение в поле свойства «Начальное состояние» = Ввод данных [dataInput]. Структура пунктов навигации созданного бизнес-процесса представлена на рисунке 42.

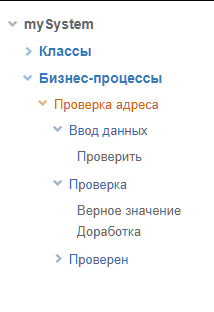


Рисунок 42 – навигация бизнес-процесса

Для отображения схемы бизнес-процесса в рабочей области Подсистемы необходимо перейти по пункту навигации «Бизнес-процессы» и выбрать бизнес-процесс для отображения. После в рабочей панели отобразиться действие «Автоматическая расстановка», при нажатии на кнопку данного действия элементы бизнес-процесса выстроят связи автоматически.

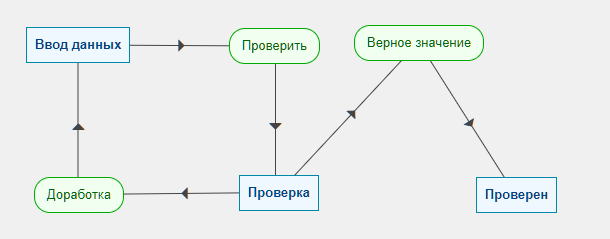


Рисунок 43 – диаграмма бизнес-процесса

На рабочей панели бизнес процесса доступно действие формирования текстового документа в формате DOCX, XSLX, PDF, или иного. Документ заполняется на основе переменных бизнес-процесса. Шаблон создаваемого документа выбирается из списка определенных ранее, либо создается индивидуально.

# Конструктор Инерфейсов

Для создания интерфейса выбираем пункт «Конструктор интерфейсов» в боковом меню Подсистемы конструирования. После чего откроется рабочая область, на которой находятся различные инструменты для создания интерфейса. На рабочей панели доступно действие «Создать интерфейс». При выборе действия, открывается форма создания интерфейса, на которой необходимо заполнить поля, в соответствии с рисунком 44.

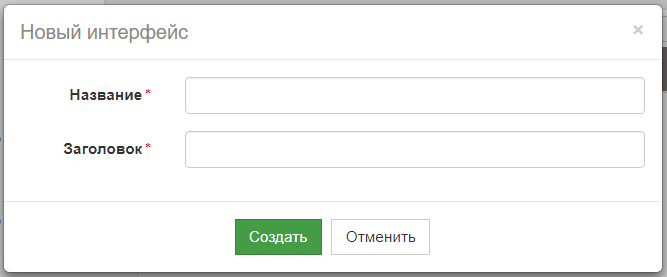


Рисунок 44 – форма создания интерфейса

После сохранения формы становятся активны инструменты для создания элементов интерфейса в рабочей области. При наведении на инструмент, отображается всплывающая подсказка, с информацией о его назначении.



Рисунок 45 – подсказка о назначении инструмента

На рабочей панели стали доступны действия «Редактировать интерфейс» - позволяет редактировать наименования интерфейса страницы и «Удалить интерфейс» - позволяет удалить интерфейс страницы.

Ниже расположена панель инструментов конструктора интерфейсов. В левом углу расположены инструменты для просмотра вариантов отображения интерфейса экрана, наглядно отображающие внешний вид создаваемой страницы, подстраивая ее содержимое под размеры различных видов экрана пользователя.

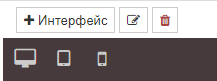


Рисунок 46 – виды экранов пользователя

В правом углу расположены инструменты для модификации блоков, создаваемых в рамках интерфейса страницы:

*  - Показать границы
*  - Предпросмотр
*  - Полный экран
*  - Экспорт
*  - Отменить действие
*  - Повторить действие
*  - Импорт
*  - Очистить созданный интерфейс

Далее в шапке правой боковой панели располагаются инструменты для создания элементов интерфейса:

*  - Менеджер стилей
*  - Настройки компонентов
*  - Слои
*  - Блоки – при выборе инструмента отображаются блоки для создания интерфейса.

## Блоки

Блоки являются основным составляющим элементом конструирования интерфейса. Для формирования интерфейса необходимо перейти в раздел блоки и добавить в рабочую область набор необходимых блоков способом перемещения их из правой боковой панели. Конструктор интерфейсов подразделяет блоки на следующие группы:

* Базовые – данная группа содержит в себе блоки в виде столбов, с несколькими вариациями количества и размера, текстового поля, вставки гиперссылки, изображения и видео (рисунок 47).

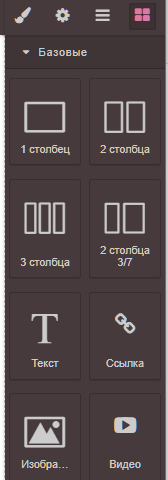
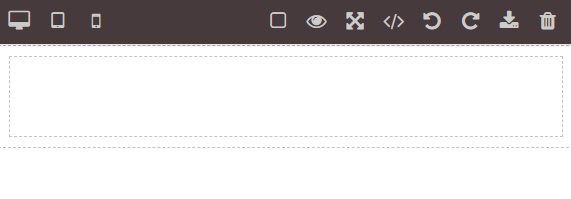
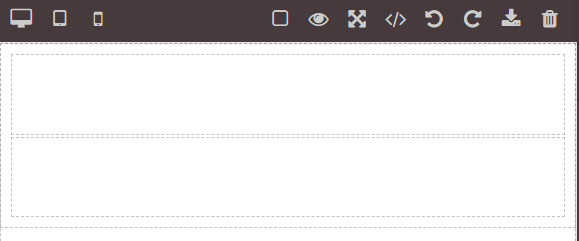


Рисунок 47 – блоки группы «Базовые»

* Блок «1 столбец»



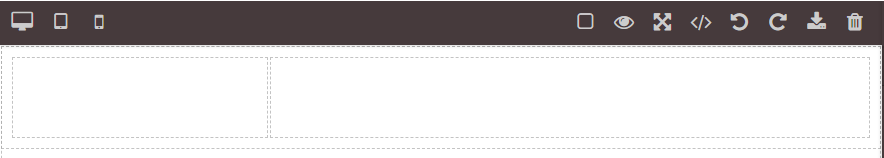
* Блок «2 столбца»



* Блок «3 столбца»



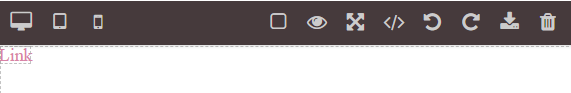
* Блок «2 столбца 3/7» – имеет возможность изменения ширины столбов



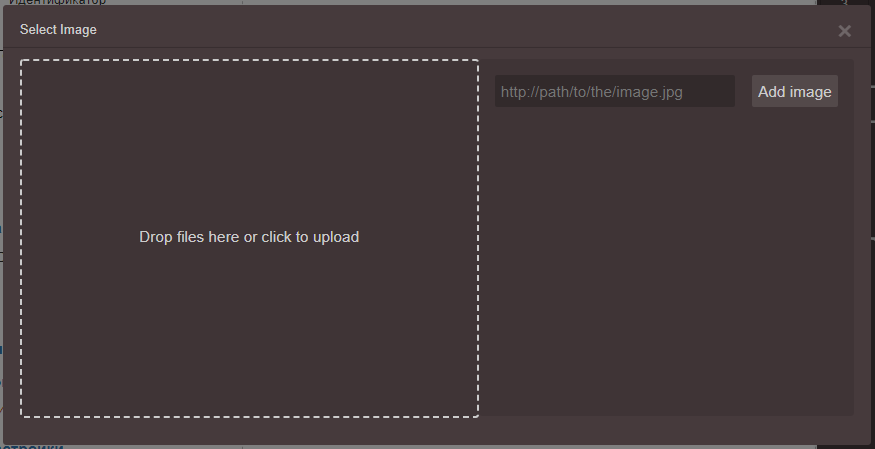
* Блок «Текст»



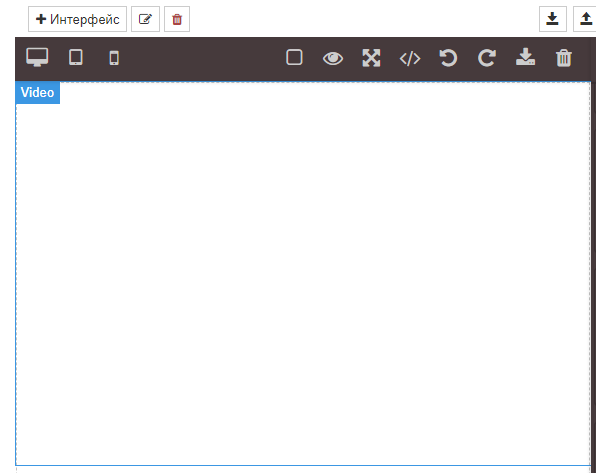
* Блок «Ссылка»



* Блок «Изображение» - при переносе блока в рабочую область появляется модальное окно для указания ссылки на добавляемое изображение.



* Блок «Видео»



* Расширения – данная группа содержит в себе блоки в виде навигации, вкладок, слайдеров, кода.

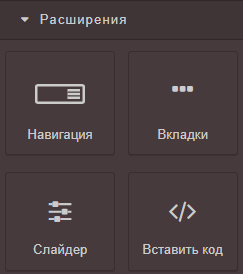
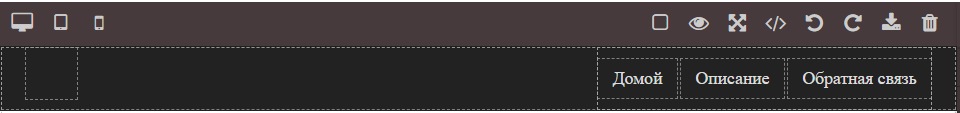
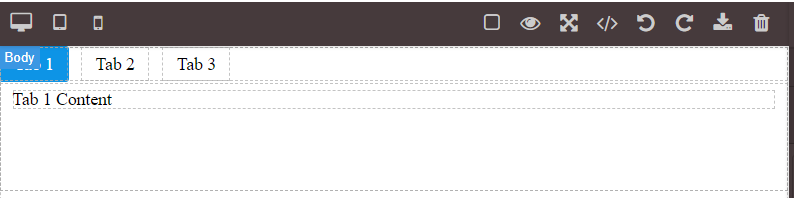


Рисунок 48 – блок группы «Расширения»

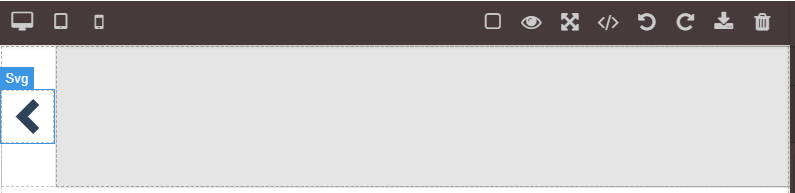
* Блок «Навигация» – позволяет отображать элемент интерфейса, применяемый в качестве навигации информационной системы, либо шапки сайта.



* Блок «Вкладки» – данный блок позволяет отобразить в интерфейсе функционал переключения между вкладками на странице



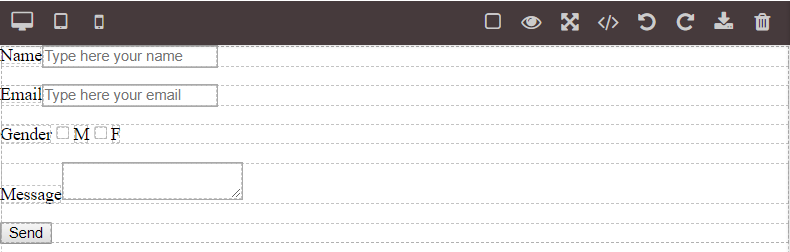
* Блок «Слайдер» – данный блок позволяет отобразить в интерфейсе функционал, позволяющий менять слайды на странице при помощи боковых стрелок



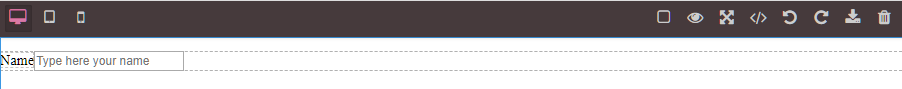
* Блок «Вставка кода»



* Формы – данная группа содержит в себе блоки в виде формы для заполнения данных, поля ввода, текстового поля, поля выбора значений из выпадающего списка, кнопки, метки, чекбокса.
* Блок «Форма»



* Блок «Поле ввода»



* Блок «Текстовое поле»
* Блок «Выбор»
* Блок «Кнопка»
* Блок «Метка»



* Блок «Чекбокс»
* Основные – данная группа содержит в себе блоки в виде отображения элемента ссылки, цитаты, текстового раздела.

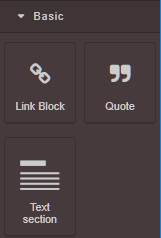
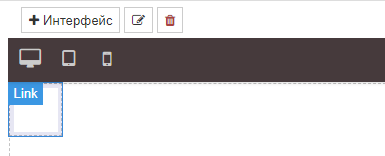
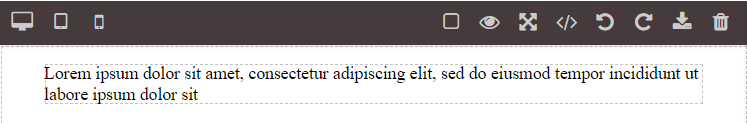


Рисунок 49 – блоки группы «Основные»

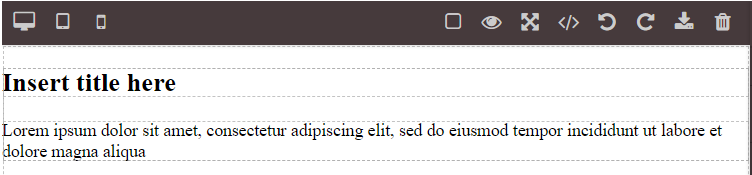
* Блок «Ссылка»



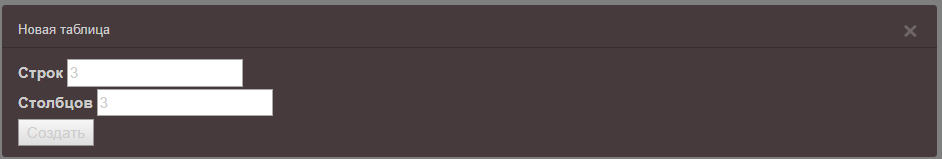
* Блок «Цитата»



* Блок «Текстовый раздел»



* Таблицы



* Блоки метаданных – данная группа содержит в себе блоки в виде бокового меню, поиска, списка, формы.

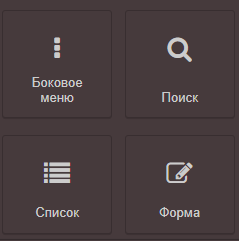
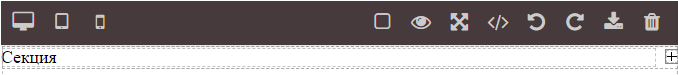
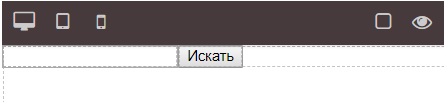


Рисунок 50 – блоки группы «Блоки метаданных»

* Блоки метаданных



* Поиск



* Список



* Форма



Каждый блок, отображаемый в рабочей области конструктора интерфейсов, имеет стандартный набор действий, увидеть которые можно выделив любой элемент блока или блок полностью (). При помощи данных действий можно осуществлять переход между элементами блока, копировать и удалять элемент блока или блок полностью.

## Менеджер стилей

При переходе в меню Менеджера стилей на правой боковой панели Конструктора интерфейсов, становится доступна возможность изменения стилей для рабочей области Конструктора интерфейсов. Например, можно изменить цвет фона, при вводе значения цвета во вкладке «Оформление», по умолчанию установлено значение «Без цвета».

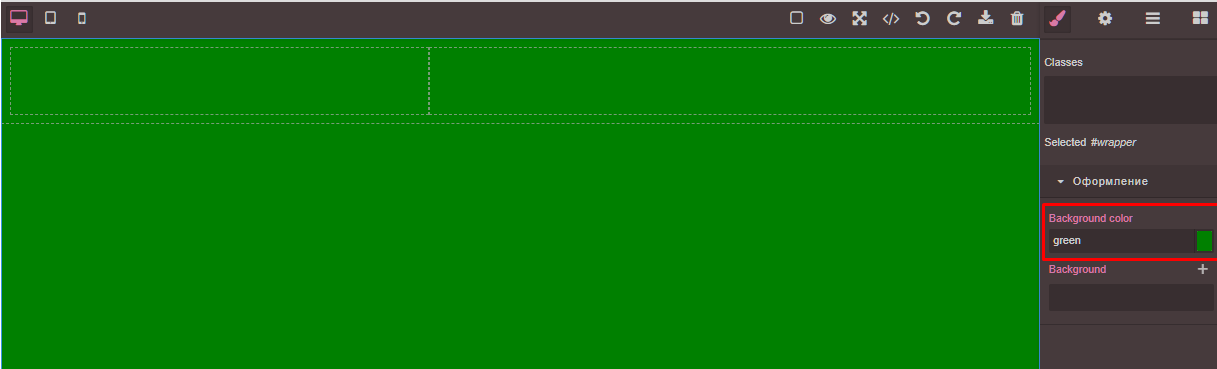


Рисунок 51 – изменение фона рабочей области

При добавлении в рабочую область блока и выделении его отдельных элементов, либо полностью, в правой боковой панели автоматически открываются элементы Менеджера стилей, предназначенные для модификации блоков или его отдельных элементов.

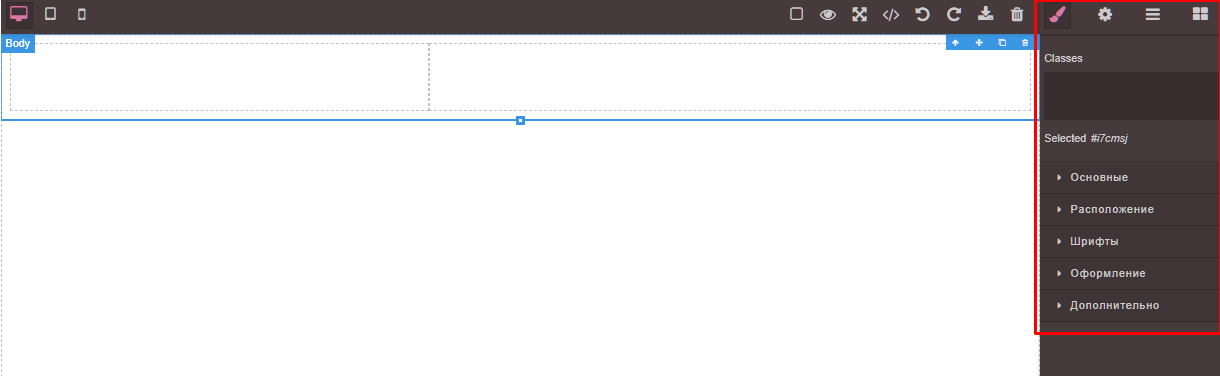


Рисунок 52 – боковая панель Менеджера стилей, применяемых к блокам

Стили блоков подразделяются на следующие группы:

* Основные
* Размещение – по всей области (значение по умолчанию), в левой области, в правой области.
* Отображение – значение выбирается из выпадающего списка.
* Расположение – задаются значения параметров расположения блока относительно рабочей области Конструктора интерфейсов.
* Шрифты – задаются значения параметров для текста, находящегося внутри блока или элемента блока, в виде: выбора шрифта, размера, цвета, расположения относительно выбранной области и др.
* Оформление – задаются значения параметров для цвета заливки и границ выбранного элемента (цвет, размер и др.).
* Дополнительно – задаются html параметры выбранного объекта.

## Слои

При переходе в меню Слои, на боковой панели отображаются все элементы рабочей области в иерархической структуре. При помощи данного функционала можно скрывать в рабочей области отдельные элементы или блоки полностью, переименовывать их и изменять порядок расположения по отношению друг к другу способом перемещения их между собой.

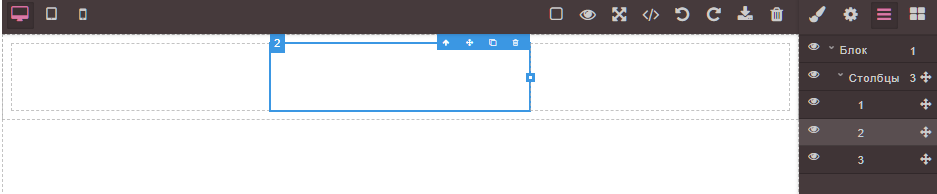


Рисунок 53 – боковая панель Слои

## Настройки компонентов

При переходе на «Настройки компонентов» на боковой панели отображаются поля «id» - указывает на идентификатор элемента и «title» - указывает на наименование заголовка элемента. Значения в полях соответствуют тому элементу, при выделении которого они были заданы. Значение в поле «title» соответствует значению, отображаемому при наведении на элемент интерфейса.

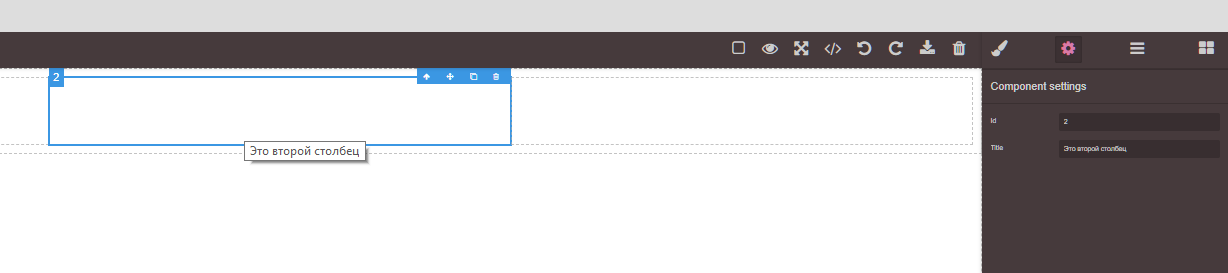


Рисунок 54 – боковая панель Настройка компонентов

# Настройки

В меню навигации Настройка содержится журнал изменений. В журнале изменений отображается информация обо всех изменениях в Подсистеме конструирования информационных систем. Данные хранятся в виде объектов в таблице, изображенной на рисунке 55.

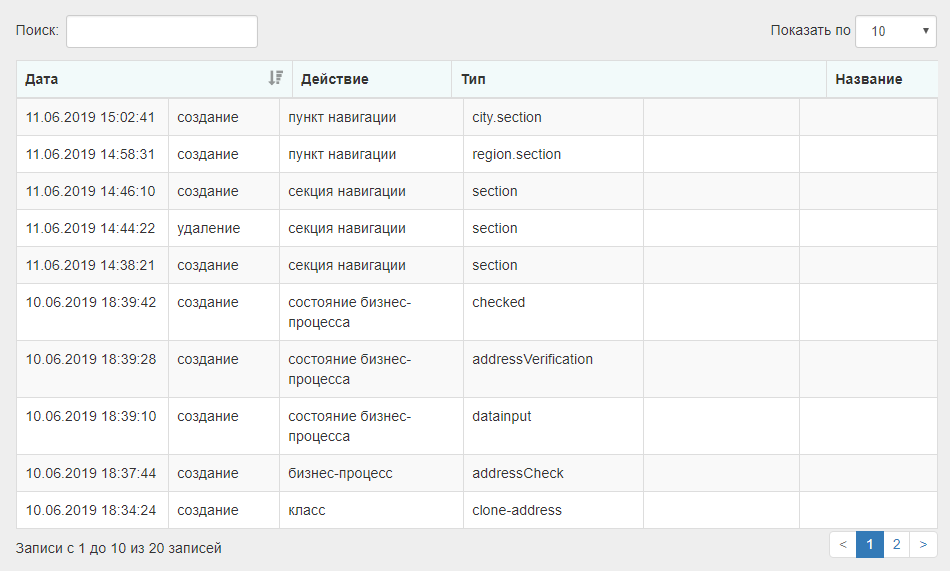


Рисунок 55 – журнал изменений

# Подключение режима офлайн

Для сборки инсталлятора нужно перейти в папку «studio» и в ней, двойным кликом мыши, запустить файл «Studio.nsi». Далее, в текущей папке, будет сформирован готовый инсталлятор под названием «Studio.exe».

Для работы с Подсистемой в режиме офлайн требуются ключи и пути, по которым будет дан доступ режиму офлайн к его серверной части и идти обмен между клиентом и сервером. Эти настройки находятся в папке «studio», в архиве «config.7z». При запуске инсталлятора необходимо, чтобы сам инсталлятор и файл «config.7z» находились в одной папке.

Запускаем инсталлятор, наблюдаем за работой Подсистемы в автономном режиме. Например, создаем класс, заполняем для него ключевые формы, сохраняем форму. Наблюдаем отображение данного класса в рабочей области и в навигации приложения, за счет синхронизации с сервером при восстановлении связи с ним.

# Экспорт/импорт приложения

Для работы с функционалом экспорта/импорта приложений в Подсистеме на рабочей панели заданы кнопки:

*  - Скачать текущее приложение
*  - Загрузить приложение в Подсистему

Для того что бы развернуть ранее созданное приложение в текущей Подсистеме – необходимо добавить в архив (формат .zip) папку с наименованием приложения, которое хотим развернуть в Подсистеме, а в ней должны находится папки, содержащие в себе файлы классов (meta), бизнес-процессов (workflow), навигации (navigation) и форм представлений (views). То есть создаем папку с названием приложения (project-management), добавляем в нее папку файлов классов, бизнес-процессов, навигаций и форм представлений, добавляем в архив папку приложения. Далее открываем Подсистему конструирования, переходим в самый верхний пункт меню и выбираем действие в рабочей панели «Загрузить приложение», откроется окно выбора архива для загрузки.

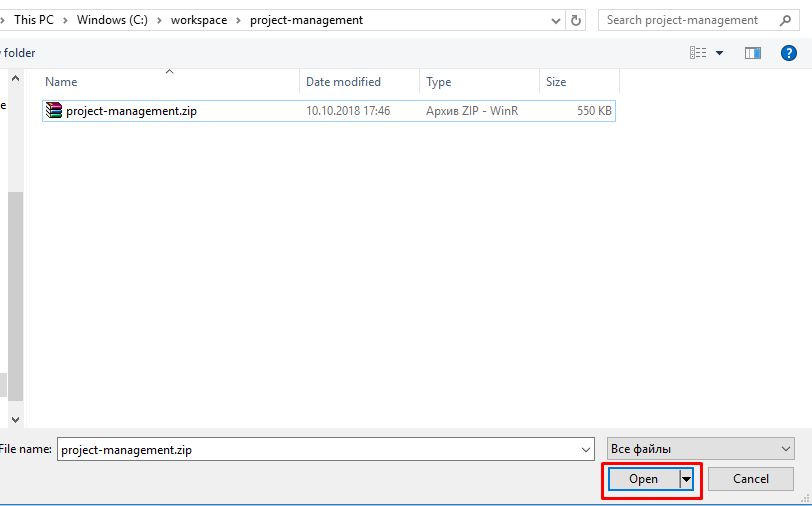


Рисунок 56 – Выбор архива с метаданными приложения для загрузки в Подсистему

Приложение отобразится в меню Подсистемы.

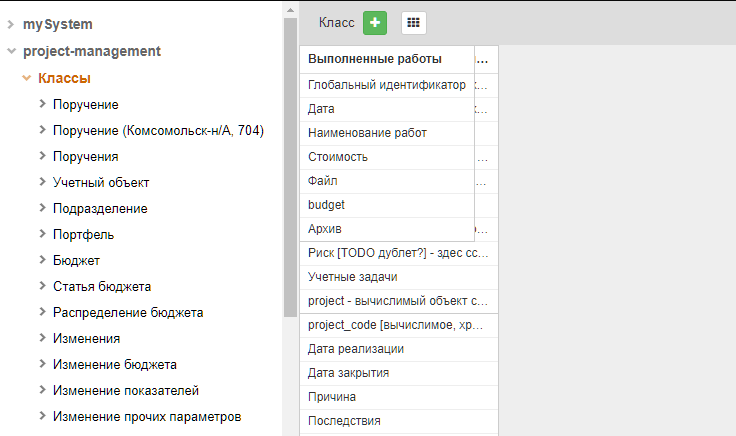


Рисунок 57 – первичное отображение структуры приложения при импорте

Что бы схема связей сущностей построилась автоматически необходимо перейти в пункт меню «Классы» в текущем приложении и в рабочей панели выбрать кнопку  «Автоматическая расстановка».

Для экспорта текущего приложения необходимо выбрать действие «Скачать приложение» на рабочей панели, при переходе из пункта навигации приложения. Приложение автоматически загружается в папку «Загрузки» в формате .zip. На рисунке 58 показана структура папок в архиве скачанного приложения.

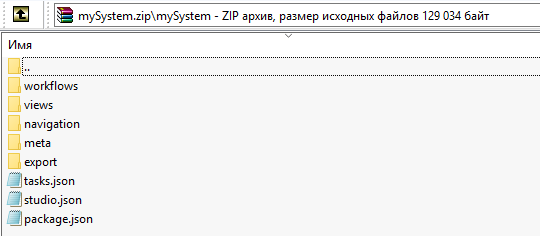


Рисунок 58 – экспорт приложения

## Импорт совпадающих элементов

При загрузке уже имеющихся метаданных проекта в Подсистему, импорт выполняется, но при наличии совпадающих сущностей – появляется информационное сообщение о совпадении загружаемых элементов, в котором содержится информация о перечне таких элементов и возможностью выбора действия для замены совпадающих элементов на более новые (выбрать сущности для замены из списка и нажать кнопку «Готово»), либо отказаться от обновления сущностей проекта, закрыв форму.

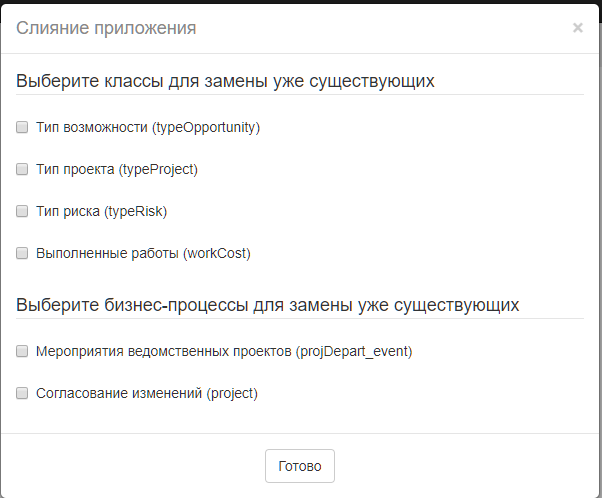


Рисунок 59 – сообщение о совпадении загружаемых элементов

## Экспорт во внешний сервис

Для того что бы выполнить экспорт приложения во внешний сервис, выбираем действие «Экспорт во внешний сервис» на рабочей панели, в корне раздела приложения. После чего появится модальное окно с параметрами экспорта, а также адресом назначения.

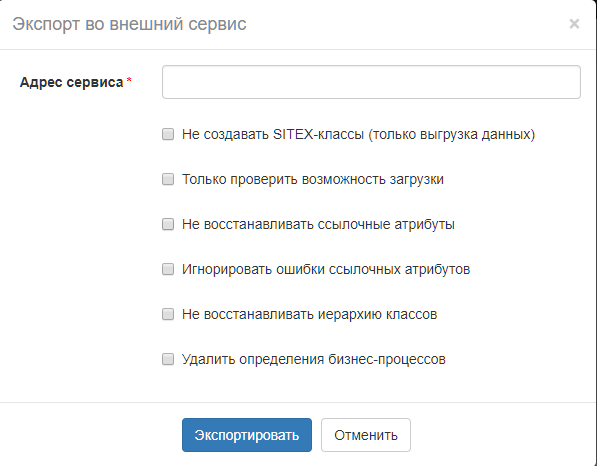


Рисунок 60 – форма настройки параметров экспорта

Адрес сервиса выбирается из выпадающего списка, но при отсутствии в нем необходимого адреса – есть возможность задать его значение вручную (например, http://10.18.1.9/util/radMetaManageUtil.sx).

Параметры экспорта задаются чек боксами, по умолчанию задано значение «false». После заполнения формы выбираем кнопку «Экспортировать». Далее получаем ответ о результате экспорта.

После выполнения процедуры экспорта приложения во внешний сервис, для него на рабочей панели становится доступным действие «Просмотреть историю экспорта», которое представляет собой журнал экспорта и содержит в себе информацию с датой, адресом и параметрами экспорта, совершаемых в рамках данного приложения.

## Формирование файла конфигурации приложения

При выполнении импорта приложения, созданного в Подсистеме конструирования, формируется системный файл, содержащий настройки конфигурации объектов, имеющий следующую структуру хранения файлов:

